Elastic Load Balance User Guide

User Guide

Issue 30

Date 2024-06-20

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

Notice

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

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

Contents

1 User Guide for Dedicated Load Balancers

1.1 Permissions Management

1.1.1 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 Huawei Cloud account or cloud service to perform efficient O&M on your ELB resources.

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

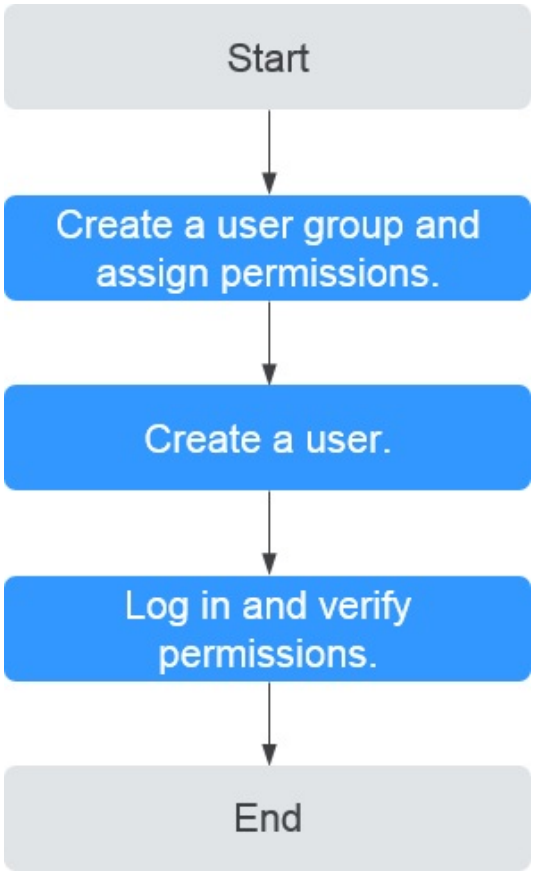
This following describes the procedure for granting permissions.

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 1-1 Process for granting ELB permissions



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 ReadOnlyAccessELB Viewer** 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.

1.1.2 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: Edit JSON policies from scratch or based on an existing policy.

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"
            ]
```



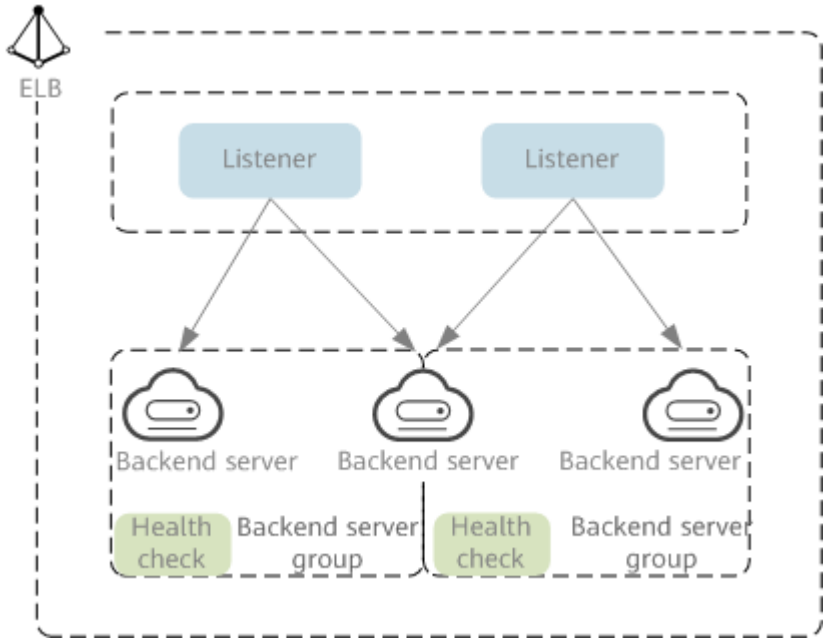
```
    }
  ]
}
```

1.2 Load Balancer

1.2.1 Dedicated 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 1-2 ELB components



Region

- You are advised to select a region that is closest to your users to reduce network latency and improve the download speed.
- 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 **IP Addresses as Backend Servers**.
- To add backend servers in different regions, you can use Cloud Connect to connect the VPCs across regions. For details, see the **Cloud Connect User Guide**.

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 1-1 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 1-2 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.

Source	Traffic Distribution
A VPC that is different from where the load balancer works	If requests are in a VPC that is different from where the load balancer works, the load balancer in the AZ where the original VPC 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

Dedicated load balancers provide a wide range of specifications to meet your requirements.

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 **Specifications of Dedicated Load Balancers**.

You can view the monitoring metrics on the Cloud Eye console to analyze the peak traffic and usage trends to select the specifications as needed.

For details, see **Table 1-3**.

Table 1-3 Guide for selecting a specification

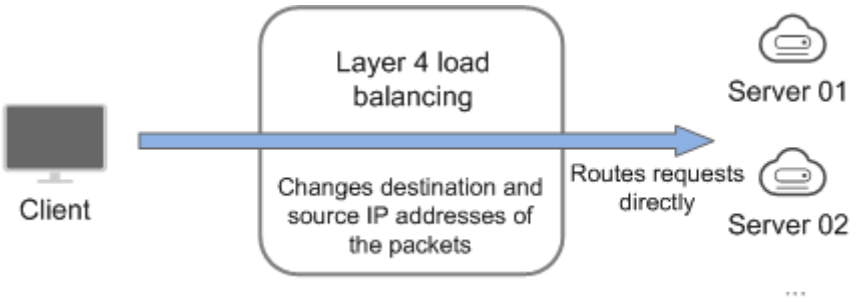
Specifications	Description
Network load balancing (TCP/UDP)	Pay attention to the maximum number of concurrent connections and consider maximum concurrent connections as a key metric. Estimate the maximum number of concurrent connections that a load balancer needs to handle and select the corresponding specification.
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 1-3 Layer 4 load balancing



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

Figure 1-4 Layer 7 load balancing

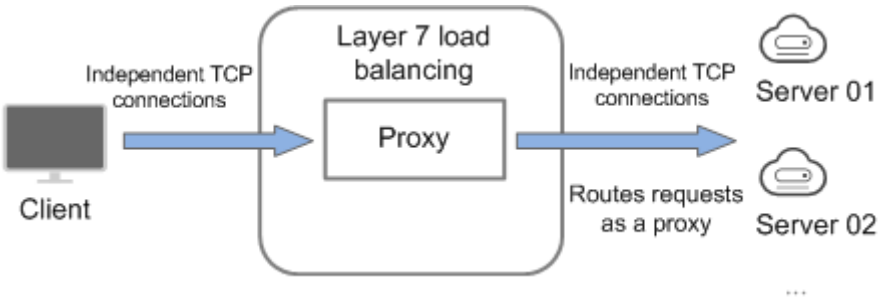


Table 1-4 Protocols

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 1-5 ELB network types

Network Type	Note	Application Scenarios
Load balancing on a public network	You need to assign an EIP or bind an existing 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.
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 must be in the same region as the load balancer.
- Cloud servers that run the same OS are recommended so that you can manage them more easily.
- ELB does not support File Transfer Protocol (FTP), but supports Secure File Transfer Protocol (SFTP) on backend servers.

1.2.2 Creating a Dedicated Load Balancer

Scenario

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

Constraints

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

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click  in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, click **Buy Elastic Load Balancer**.

   Complete the basic configurations based on **Table 1-6**.

   Table 1-6 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. For details about the differences, see **Differences Between Dedicated and Shared Load Balancers**.
Billing Mode	Specifies the billing mode of the dedicated load balancer. You are charged for how long you use each load balancer.
Region	Specifies the desired region. Resources in different regions cannot communicate with each other over internal networks. For lower network latency and faster access to resources, select the nearest region.

Parameter	Description
AZ	Specifies the AZ of the load balancer. An AZ is a part of a region and has its own independent power supplies and networks. AZs are physically isolated but interconnected through an internal network. 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 can route 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 the number of 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.
Specifications	Select **Elastic** or **Fixed** if pay-per-use is chosen as the billing mode. • 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. Select either **Application load balancing (HTTP/HTTPS)** or **Network load balancing (TCP/UDP)** or both, and then select the desired specification. You can select only one specification for **Application load balancing (HTTP/HTTPS)** and **Network load balancing (TCP/UDP)**, respectively. Select the desired specifications based on your service size by referring to **Specifications of Dedicated Load Balancers**.
Name	Specifies the load balancer name.
Enterprise Project	Specifies an enterprise project by which cloud resources and members are centrally managed.
Description	Provides supplementary information about the load balancer.
Tag	Identifies 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 1-7**. A maximum of 20 tags can be added.

Table 1-7 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. • Only letters, digits, underscores (_), hyphens (-), at signs (@), and Chinese characters are allowed.
Tag value	• Can contain a maximum of 43 characters. • Only letters, digits, underscores (_), hyphens (-), at signs (@), and Chinese characters are allowed.

5. Configure the network parameters based on **Table 1-8**.

Table 1-8 Parameters for network configurations

Parameter	Description
IP as a Backend	Specifies whether to associate backend servers that are not in the VPC of the load balancer. After this function is enabled, you can associate the backend servers with the load balancer by using their IP addresses. **NOTE** • To use this function, you need to configure correct VPC routes to ensure requests can be routed to backend servers. • If you enable this option, more IP addresses in the backend subnet will be reserved for the load balancer to communicate with backend servers. Ensure that the selected subnet has sufficient IP addresses. After you select a subnet, you can view the number of IP addresses required by the load balancer in the infotip.
Network Type	Specifies the network where the load balancer works. You can select one or more network types. • **Public IPv4 network**: The load balancer routes IPv4 requests from the clients to backend servers over the Internet. • **Private IPv4 network**: The load balancer routes IPv4 requests from the clients to backend servers in a VPC. • **IPv6 network**: An IPv6 address will be assigned to the load balancer to route requests from IPv6 clients. **NOTE** If you do not select any of the options, the load balancer cannot communicate with the clients after it is created. When you are using ELB or testing network connectivity, ensure that the load balancer has a public or private IP address bound.

Parameter	Description
VPC	Specifies the VPC where the dedicated load balancer will work. You cannot change the VPC after the load balancer is created. Plan the VPC as required. Select an existing VPC, or click **View VPCs** and create a desired one. You can select a VPC and subnet shared by another account for improved resource management and reduced O&M costs. For more information about VPC sharing, see **VPC Sharing** in the **Virtual Private Cloud User Guide** *Virtual Private Cloud User Guide*.
Frontend Subnet	Specifies the frontend subnet from which an IP address will be assigned to the load balancer to receive client requests. After a load balancer is created, you can unbind the IP address from it and assign an IP address from a new frontend subnet to the load balancer. The system assigns IP addresses in this subnet to load balancers for receiving requests based on the configured network type. • **IPv4 private network**: assigns IPv4 private addresses. • **IPv6 network**: assigns IPv6 private or public addresses. **NOTE** If you select **IPv6 network** for **Network Type** and the selected VPC does not have any subnet that supports IPv6, enable IPv6 for the subnets or create a subnet that supports IPv6. For details, see the **Virtual Private Cloud User Guide**.
Backend Subnet	Specifies the backend subnet from which an IP address will be assigned to the load balancer to forward requests to backend servers. • **Subnet of the load balancer** is selected by default. • Select an existing subnet in the VPC where the load balancer works. • Create a new subnet. **NOTE** • The number of IP addresses required depend on the specifications, number of AZs, and IP as a backend function you have configured when you create the load balancer. See the number of occupied IP addresses on the console. • An application load balancer requires 8 to 30 additional IP addresses in the backend subnet for traffic forwarding. The actual number of required IP addresses depends on the ELB cluster size. If load balancers are deployed in the same cluster and work in the same backend subnet, they share the same IP addresses to save resources.

Parameter	Description
Private IPv4 network configuration	
IPv4 Address	Specifies how you want the IPv4 address to be assigned. • **Automatically assign IP address**: The system automatically assigns an IPv4 address to the load balancer. • **Manually specify IP address**: You need to manually specify an IPv4 address for the load balancer. **NOTE** Network ACL rules configured for the backend subnet of the load balancer will not restrict the traffic from the clients to the load balancer. If network ACL rules are configured, the clients can directly access the load balancer. To control access to the load balancer, configure access control for all listeners added to the load balancer. For details, see **What Is Access Control?**
IPv6 network configuration	
IPv6 Address	Specifies how you want the IPv6 address to be assigned. • **Automatically assign IP address**: The system automatically assigns an IPv6 address to the load balancer. • **Manually specify IP address**: You need to manually specify an IPv6 address for the load balancer. **NOTE** Network ACL rules configured for the backend subnet of the load balancer will not restrict the traffic from the clients to the load balancer. If network ACL rules are configured, the clients can directly access the load balancer. To control access to the load balancer, configure access control for all listeners added to the load balancer. For details, see **What Is Access Control?**
Shared bandwidth	Specifies the shared bandwidth that the IPv6 address will be added to. You can choose not to select a shared bandwidth, select an existing shared bandwidth, or assign a shared bandwidth.
Public IPv4 network configuration	
EIP	Specifies the public IP address that will be bound to the load balancer for receiving and forwarding requests over the Internet. This parameter is mandatory when **Network Type** is set to **IPv4 public network**. • **New EIP**: The system will assign a new EIP to the load balancer. • **Use existing**: Select an existing EIP.

Parameter	Description
EIP Type	Specifies the link type (BGP) when a new EIP is used. • **Dynamic BGP**: When changes occur on a network using dynamic BGP, routing protocols provide automatic, real-time optimization of network configurations, ensuring network stability and optimal user experience. • **Static BGP**: When changes occur on a network using static BGP, carriers cannot adjust network configurations in real time to ensure optimal user experience.
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 a maximum bandwidth and pay for the outbound traffic you use. • **Shared Bandwidth**: The bandwidth is suitable for staggered traffic.
Bandwidth	Specifies the maximum bandwidth.

6. Click **Next**.
7. Confirm the configuration and submit your request.

Exporting the Load Balancer List

After a load balancer is created, you can export the information about all load balancers under your account to a local directory as an Excel file.

This file records the name, ID, status, type, and specifications of the load balancers.

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** > **Elastic Load Balance**.
4. In the upper left corner of the load balancer list, click **Export**.

   The system will automatically export information about all of your load balancers as an Excel file to a local directory.

1.2.3 Creating a Dedicated Load Balancer in a Shared Subnet

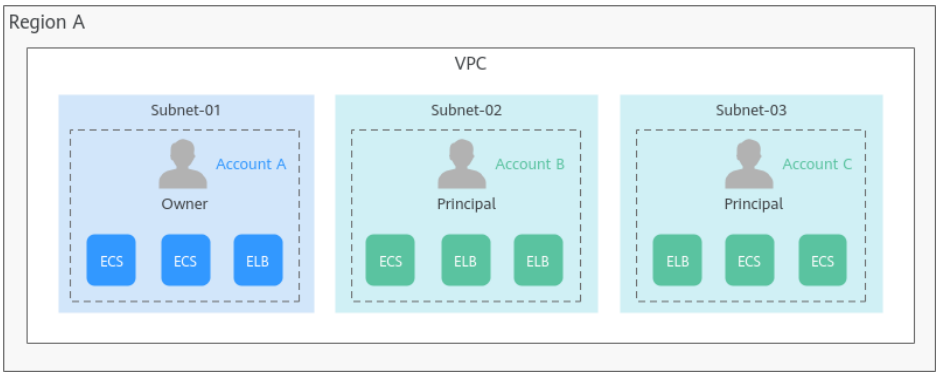
Scenarios

With VPC sharing, you can create a dedicated load balancer in a subnet shared by another account. This helps you manage resources easily and reduce O&M costs.

The following describes how you can share subnets among several accounts. Let us call them accounts A, B, and C:

- Account A: the IT management account and the owner of the VPC and subnets. Account A creates a VPC and three subnets and shares these subnets with accounts B and C. Account A can create resources in **Subnet-01**.
- Account B: a service account and a principal of the shared subnet **Subnet-02**. Account B can create dedicated load balancers in **Subnet-02**.
- Account C: a service account and a principal of the shared subnet **Subnet-03**. Account C can create dedicated load balancers in **Subnet-03**.

Figure 1-5 Service planning



For more information about VPC subnet sharing, see **VPC Sharing** in the *Virtual Private Cloud User Guide*.

Notes and Constraints

- A principal can receive a maximum of 100 subnet shares.
- VPC sharing is free. You only need to pay for the resources (such as ECSs and dedicated load balancers) you create in the shared subnets.

Prerequisites

Account A, as the resource owner, has created a VPC and subnets, and specified account B as the principal. For details, see **Creating a Resource Share**.

Procedure

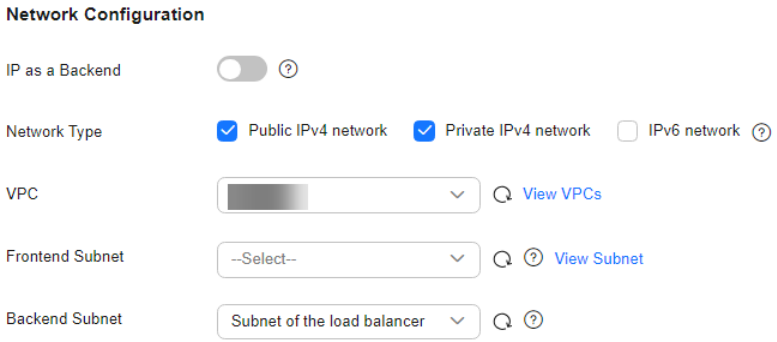
1. Log in to the **Resource Access Manager** console using account B and accept the resource share.

   For details, see **Responding to a Resource Sharing Invitation**.
2. Go to the ELB console.

3. Create a dedicated load balancer and configure basic and network parameters.

   Select the VPC and subnet shared by account A.

   Figure 1-6 Configuring network parameters

   For other parameters, see **Creating a Dedicated Load Balancer**.

1.2.4 Enabling or Disabling Modification Protection for Dedicated Load Balancers

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

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. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. On the **Summary** tab, click **Configure** next to **Modification Protection**.
6. In the **Configure Modification Protection** dialog box, enable or disable **Modification Protection**.

   Fill in the reason if needed.
7. Click **OK**.

NOTE

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

1.2.5 Modifying the Basic Configurations of a Dedicated Load Balancer

After a dedicated load balancer is created, you can change its specifications and AZ as required.

Modifying Specifications

You can change the specifications of a dedicated load balancer on the console:

- Change the elastic specifications to fixed specifications, or the other way round.
- Change the 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 specifications, for example, upgrade small I to medium I, or downgrade large I to medium I.

Change options vary by billing mode. To find out what changes you can make, see **Table 1-9**.

NOTE

- Upgrading specifications does not interrupt your services.
- Downgrading specifications will temporarily disconnect services.
  - Network load balancing (TCP/UDP): New connections may not be able to be established.
  - Application load balancing (HTTP/HTTPS): New connections may not be able to be established and some persistent connections may be interrupted.

Pay-per-Use

Table 1-9 Supported change options for a pay-per-use load balancer

Billing Mode	Specifications	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. 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** > **Elastic Load Balance**.

4. On the **Load Balancers** page, locate the load balancer, click **More** in the **Operation** column, and select **Change Specifications**.
5. Select the new specifications and click **Next**.
6. Confirm the information and click **Submit**.

Changing an AZ

You can only deploy a load balancer in an additional AZ but cannot remove it from an AZ.

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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer, click **More** in the **Operation** column, and select **Change AZs**.
5. Select one or more new AZs and click **Next**.
6. Confirm the information and click **Submit**.

CAUTION

Changing AZs will temporarily affect services. New connections may not be able to be established and some persistent connections may be interrupted. If this happens, try again to restore the connections.

Popular Questions

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.

1.2.6 Modifying the Network Configurations of a Dedicated Load Balancer

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

Binding or Unbinding an IP Address

You can bind or unbind an IPv4 EIP, a private IPv4 address, or an IPv6 address, to or from a dedicated load balancer as required.

NOTE

- Load balancers without IPv4 EIPs cannot route requests over the public IPv4 network.
- Load balancers without private IPv4 addresses cannot route requests over the private IPv4 network.
- Load balancers without IPv6 addresses cannot route requests over the IPv6 network.

Binding or Unbinding an IPv4 EIP

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** 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. 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** > **Elastic Load Balance**.
4. On the **Load Balancers** 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. 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** > **Elastic Load Balance**.
4. On the **Load Balancers** 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. 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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and choose **More** > **Change Private IPv4 Address** in the **Operation** column.
5. In the **Change Private IPv4 Address** dialog box, select the subnet where the IP address resides and specify 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.

6. Click **OK**.

Changing an IPv6 Address

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and choose **More** > **Change IPv6 Address** in the **Operation** column.
5. 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.
6. 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.

NOTE

- 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. 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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click **More** in the **Operation** column.
5. Click **Modify IPv4 Bandwidth** or **Modify IPv6 Bandwidth**.
6. In the **New Configuration** area, modify the billing option and 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.
7. 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 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 requests within a VPC.

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** > **Elastic Load Balance**.
4. On the **Load Balancers** 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 bandwidths are 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.
5. Click **OK**.

1.2.7 Deleting a Dedicated Load Balancer

Scenarios

You can delete a load balancer if you on longer need it.

CAUTION

Back up the data if necessary. The data will be deleted immediately after you delete or unsubscribe from the load balancers and cannot be restored.

Prerequisites

To delete a load balancer, first delete the resources associated with it in the following order:

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

Deleting a Pay-per-Use Load Balancer

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

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** > **Elastic Load Balance**.
4. Locate the target load balancer and choose **More** > **Delete** in the **Operation** column.

   A confirmation dialog box is displayed.
5. In the displayed dialog box, enter **DELETE**.
6. Click **OK**.

1.3 Listener

1.3.1 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 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 1-10 Protocols supported by ELB

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

NOTE

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 1-11 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 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 port numbers: • 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 1-12 Backend protocols and ports

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

Listen to All Ports

Listen to All Ports 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 range you specify and routes them to the corresponding ports on the backend servers.

Timeout Durations

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

Figure 1-7 Timeout durations at Layer 4

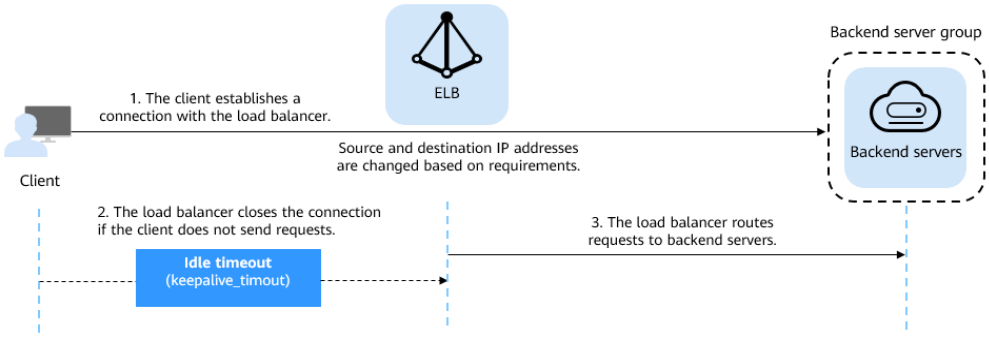


Figure 1-8 Timeout durations at Layer 7

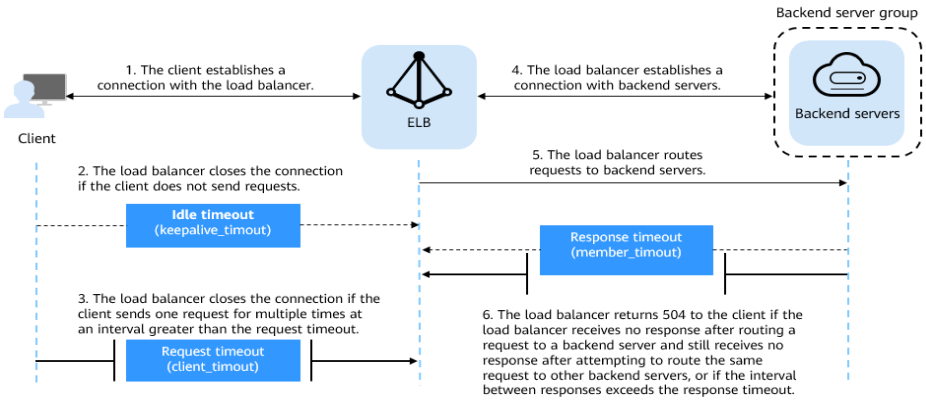


Table 1-13 Timeout durations

Protocol	Type	Description	Value Range	Default Timeout Duration
TCP	Idle Timeout	Duration for a connection to keep alive. If no request is received within this period, the load balancer closes the connection and establishes a new one with the client when the next request arrives.	10–4000s	300s
UDP	Idle Timeout		10–4000s	300s
HTTP/ HTTPS	Idle Timeout		0–4000s	60s

Protocol	Type	Description	Value Range	Default Timeout Duration
	Request Timeout	Duration 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.	1–300s	60s
	Response Timeout	Duration after which the load balancer sends a 504 Gateway Timeout error to the client if the load balancer receives no response after routing a request to a backend server and receives no response after attempting to route the same request to other backend servers. **NOTE** If 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.	1–300s	60s

1.3.2 Listeners at Layer 4

1.3.2.1 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 application load balancing (HTTP/HTTPS) for your dedicated load balancer, you cannot add a TCP listener to this load balancer.

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. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**. Configure the parameters based on **Table 1-14**.

Table 1-14 Parameters for configuring a TCP listener

Parameter	Description
Name	Specifies the listener name.
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients. Select **TCP**.
Listen to All Ports	This option is available only for TCP or UDP listeners of a dedicated load balancer. It cannot be disabled after it is enabled. If this option is enabled, the listener listens to requests from all ports in the port range you specify and routes the requests to the corresponding ports on the backend servers.
Frontend Port	Specifies the port that will be used by the load balancer to receive requests from clients. The port number ranges from 1 to 65535. **NOTE** If you enable **Listen to All Ports**, you need to enter a start and end port number as the port range.
Access Control	Specifies how access to the listener is controlled. For details, see **What Is Access Control?** The following options are available: • All IP addresses • Blacklist • Whitelist
IP Address Group	Specifies the IP address group associated with a whitelist or blacklist. If there is no IP address group, create one first. For more information, see **IP Address Group**.

Parameter	Description
Transfer Client IP Address	Specifies whether to transmit IP addresses of the clients to backend servers. This function is enabled for dedicated load balancers by default and cannot be disabled.
Advanced Settings	
Idle Timeout	Specifies the length of time for a connection to keep alive, in seconds. If no request is received within this period, the load balancer closes the connection and establishes a new one with the client when the next request arrives. The idle timeout duration ranges from **10** to **4000**.
Description	Provides supplementary information about the listener. You can enter a maximum of 255 characters.

6. Click **Next: Configure Request Routing Policy**.
   a. You are advised to select an existing backend server group.
   b. You can also click **Create new** to create a backend server group.
      i. Configure the backend server group based on **Table 1-44**.
      ii. Click **Next: Add Backend Server**. Add backend servers and configure health check for the backend server group.
      For details about how to add backend servers, see **Backend Server Overview**. For the parameters required for configuring a health check, see **Table 1-45**.
7. Click **Next: Confirm**.
8. Confirm the configurations and click **Submit**.

1.3.2.2 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 quotations.

Constraints

- UDP listeners do not support fragmentation.
- The port of UDP listeners cannot be 4789.
- UDP packets can have any size less than 1,500 bytes. The packets will be discarded if they are bigger than 1,500 bytes. To avoid this, you need to modify the configuration files of the applications based on the maximum transmission unit (MTU) value.

- Dedicated load balancers: The backend protocol can be UDP or QUIC if the frontend protocol is UDP.
- If you only select application load balancing (HTTP/HTTPS) for your dedicated load balancer, you cannot add a UDP listener to this load balancer.

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. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**. Configure the parameters based on **Table 1-15**.

Table 1-15 Parameters for configuring a UDP listener

Parameter	Description
Name	Specifies the listener name.
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients. Select **UDP**.
Listen to All Ports	This option is available only for TCP or UDP listeners of a dedicated load balancer. It cannot be disabled after it is enabled. If this option is enabled, the listener listens to requests from all ports in the port range you specify and routes the requests to the corresponding ports on the backend servers.
Frontend Port	Specifies the port that will be used by the load balancer to receive requests from clients. The port number ranges from 1 to 65535. **NOTE** If you enable **Listen to All Ports**, you need to enter a start and end port number as the port range.
Access Control	Specifies how access to the listener is controlled. For details, see **What Is Access Control?** The following options are available: • All IP addresses • Blacklist • Whitelist

Parameter	Description
IP Address Group	Specifies the IP address group associated with a whitelist or blacklist. If there is no IP address group, create one first. For more information, see **IP Address Group**.
Transfer Client IP Address	Specifies whether to transmit IP addresses of the clients to backend servers. This function is enabled for dedicated load balancers by default and cannot be disabled.
Advanced Settings	
Idle Timeout	Specifies the length of time for a connection to keep alive, in seconds. If no request is received within this period, the load balancer closes the connection and establishes a new one with the client when the next request arrives. The idle timeout duration ranges from **10** to **4000**.
Description	Provides supplementary information about the listener. You can enter a maximum of 255 characters.

6. Click **Next: Configure Request Routing Policy**.
   a. You are advised to select an existing backend server group.
   b. You can also click **Create new** to create a backend server group.
      i. Configure the backend server group based on **Table 1-44**.
      ii. Click **Next: Add Backend Server**. Add backend servers and configure health check for the backend server group.
      For details about how to add backend servers, see **Backend Server Overview**. For the parameters required for configuring a health check, see **Table 1-45**.
7. Click **Next: Confirm**.
8. Confirm the configurations and click **Submit**.

1.3.2.3 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 a mobile network.

NOTE

- QUIC versions include Q043, Q046, and Q050.
- UDP listeners using QUIC as backend protocol do not support fragmentation.

Constraints

- Only dedicated load balancers support the QUIC protocol.
- You can add only UDP listeners if you want to use QUIC as the backend protocol.

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. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.

   Ensure that **Network load balancing (TCP/UDP)** has been selected for the load balancer.
5. Under **Listeners**, click **Add Listener**.
6. In the **Configure Listener** step, set **Frontend Protocol** to **UDP**, configure other parameters as required, and click **Next: Configure Request Routing Policy**.
7. In the **Configure Routing Policy** step, set **Backend Protocol** to **QUIC** and configure other parameters as required.
8. 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**.

1.3.3 Listeners at Layer 7

1.3.3.1 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

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

- If you only select network load balancing (TCP/UDP) for your dedicated load balancer, you cannot add an HTTP listener to this load balancer.

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. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**. Configure the parameters based on **Table 1-16**.

Table 1-16 Parameters for configuring an HTTP listener

Parameter	Description
Name	Specifies the listener name.
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients. Select **HTTP**.
Frontend Port	Specifies the port that will be used by the load balancer to receive requests from clients. The port number ranges from 1 to 65535.
Redirect	Specifies whether to enable redirection. If you have both HTTPS and HTTP listeners, you can use this function to redirect the requests from the HTTP listener to the HTTPS listener to ensure security. If you create a redirect for an HTTP listener, the backend server will return HTTP 301 Move Permanently to the clients.
Redirected To	Specifies the HTTPS listener to which requests are redirected if **Redirect** is enabled.
Access Control	Specifies how access to the listener is controlled. For details, see **What Is Access Control?** The following options are available: • All IP addresses • Blacklist • Whitelist

Parameter	Description
IP Address Group	Specifies the IP address group associated with a whitelist or blacklist. If there is no IP address group, create one first. For more information, see **IP Address Group**.
Transfer Client IP Address	Specifies whether to transmit IP addresses of the clients to backend servers. This function is enabled for dedicated load balancers by default and cannot be disabled.
Advanced Forwarding	Specifies whether to enable the advanced forwarding policy. You can add advanced forwarding policies to HTTP or HTTPS listeners to forward requests to different backend server groups. For more information, see **Advanced Forwarding**.
Advanced Settings	
HTTP Headers	You can enable the following options as needed. • Transfer headers: – **Transfer Load Balancer EIP**: transmits the EIP bound to the load balancer to backend servers through the X-Forwarded-ELB-IP header. – **Transfer Listener Port Number**: transmits the port number used by the listener to backend servers through the X-Forwarded-Port header. – **Transfer Port Number in the Request**: transmits the port number used by the client to backend servers through the X-Forwarded-For-Port header. – **Transfer Load Balancer ID**: transmits the load balancer ID to backend servers through the X-Forwarded-ELB-ID header. • Rewrite headers: – **Rewrite X-Forwarded-Host**: rewrites the Host header of the client into the X-Forwarded-Host header and transmits it to the backend servers. – **Rewrite X-Forwarded-Proto**: rewrites the listener protocol into the X-Forwarded-Proto header and transmits it to the backend servers. – **Rewrite X-Real-IP**: rewrites the source IP address of the client into the X-Real-IP header and transmits it to the backend servers. For details, see **HTTP Headers**.

Parameter	Description
Data Compression	Specifies whether to enable the data compression option. If you do not enable this option, files will not be compressed. • Brotli can compress all files. • Gzip can be configured to compress the following content types: text/xml text/plain text/css application/javascript application/x-javascript application/rss+xml application/atom+xml application/xml application/json.
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 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.

6. Click **Next: Configure Request Routing Policy**.
   a. You are advised to select an existing backend server group.
   b. You can also click **Create new** to create a backend server group.
      i. Configure the backend server group based on **Table 1-44**.
      ii. Click **Next: Add Backend Server**. Add backend servers and configure health check for the backend server group.

For details about how to add backend servers, see **Backend Server Overview**. For the parameters required for configuring a health check, see **Table 1-45**.

7. Click **Next: Confirm**.
8. Confirm the configurations and click **Submit**.

1.3.3.2 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.

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, ensure that ACL rules are not configured for this subnet. If rules are configured, request packets may not be allowed.

Constraints

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

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. Click  in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**. Configure the parameters based on **Table 1-17**.

Table 1-17 Parameters for configuring an HTTPS listener

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

Parameter	Description
Frontend Port	Specifies the port that will be used by the load balancer to receive requests from clients. The port number ranges from 1 to 65535.
SSL Authentication	Specifies how you want the clients and backend servers to be authenticated. There are two options: **One-way authentication** or **Mutual authentication**. • If only server authentication is required, select **One-way authentication**. • If you want the clients and the load balancer to authenticate each other, select **Mutual authentication**. Only authenticated clients will be allowed to access the load balancer.
CA Certificate	Specifies the certificate that will be used by the backend server to authenticate the client when **SSL Authentication** is set to **Mutual authentication**. A CA certificate is issued by a certificate authority (CA) and used to verify the certificate issuer. If HTTPS mutual authentication is required, HTTPS connections can be established only when the client provides a certificate issued by a specific CA. For details, see **Adding a Certificate**.
Server Certificate	Specifies the certificate that will be used by the backend server to authenticate the client when HTTPS is used as the frontend protocol. The server certificate is used for SSL handshake negotiation to authenticate clients and ensure encrypted transmission. For details, see **Adding a Certificate**.
Enable SNI	Specifies whether to enable SNI when HTTPS is used as the frontend protocol. SNI can be used when a server uses multiple domain names and certificates. This allows the client to submit the domain name information while sending an SSL handshake request. After the load balancer receives the request, the load balancer queries the corresponding certificate based on the domain name and returns it to the client. If no certificate is found, the load balancer will return the default certificate. For details, see **SNI Certificate**.

Parameter	Description
SNI Certificate	Specifies the certificate associated with the domain name when the frontend protocol is HTTPS and SNI is enabled. Select an existing certificate or create one. For details, see **Adding a Certificate**.
Access Control	Specifies how access to the listener is controlled. For details, see **What Is Access Control?** The following options are available • All IP addresses • Blacklist • Whitelist
IP Address Group	Specifies the IP address group associated with a whitelist or blacklist. If there is no IP address group, create one first. For more information, see **IP Address Group**.
Transfer Client IP Address	Specifies whether to transmit IP addresses of the clients to backend servers. This function is enabled for dedicated load balancers by default and cannot be disabled.
Advanced Forwarding	Specifies whether to enable the advanced forwarding policy. You can add advanced forwarding policies to HTTP or HTTPS listeners to forward requests to different backend server groups. For more information, see **Advanced Forwarding**.
Advanced Settings	
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 **HTTP/2**.

Parameter	Description
HTTP Headers	You can enable the following options as needed. • Transfer headers: – **Transfer Load Balancer EIP**: transmits the EIP bound to the load balancer to backend servers through the X-Forwarded-ELB-IP header. – **Transfer Listener Port Number**: transmits the port number used by the listener to backend servers through the X-Forwarded-Port header. – **Transfer Port Number in the Request**: transmits the port number used by the client to backend servers through the X-Forwarded-For-Port header. – **Transfer Load Balancer ID**: transmits the load balancer ID to backend servers through the X-Forwarded-ELB-ID header. • Rewrite headers: – **Rewrite X-Forwarded-Host**: rewrites the Host header of the client into the X-Forwarded-Host header and transmits it to the backend servers. – **Rewrite X-Forwarded-Proto**: rewrites the listener protocol into the X-Forwarded-Proto header and transmits it to the backend servers. – **Rewrite X-Real-IP**: rewrites the source IP address of the client into the X-Real-IP header and transmits it to the backend servers. For details, see **HTTP Headers**.
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 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.

6. Click **Next: Configure Request Routing Policy**.
   a. You are advised to select an existing backend server group.
   b. You can also click **Create new** to create a backend server group.
      i. For details about how to configure a backend server group, see **Table 1-44**.
      ii. Click **Next: Add Backend Server**. Add backend servers and configure health check for the backend server group.
      For details about how to add backend servers, see **Backend Server Overview**. For the parameters required for configuring a health check, see **Table 1-45**.
7. Click **Next: Confirm**.
8. Confirm the configurations and click **Submit**.

1.3.3.3 Forwarding Policy

Overview

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

A forwarding policy consists of two parts: forwarding rule and action. For details, see **Table 1-18**.

Table 1-18 Rules and actions supported by a forwarding policy

Policy Type	Forwarding Rules	Actions
Forwarding policy	**Domain name** and **URL**	**Forward to another backend server group** and **Redirect to another listener** (only for HTTP listeners)
Advanced forwarding policy	**Domain name**, **URL**, **HTTP request method**, **HTTP header**, **Query string**, and **CIDR block**	**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 you add a forwarding policy, the load balancer forwards requests based on the specified domain name or URL:
  - If the domain name or URL in a request matches what is specified in the forwarding policy, the request is forwarded to the backend server group you select or create when you add the forwarding policy.
  - If the domain name or URL in a request does not match what is specified in the forwarding policy, the request is forwarded to the default backend server group of the listener.
- Matching priority:
  - Forwarding policy priorities are independent of each other regardless of domain names. If a forwarding rule uses both domain names and URLs, requests are matched based on domain names first.
  - If the forwarding rule is a URL, the priorities follow the order of exact match, prefix match, and regular expression match. If the matching types are the same, the longer the URL length, the higher the priority.

Table 1-19 Example forwarding policies

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

NOTE

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

Constraints

- Forwarding policies can be added only to HTTP and HTTPS listeners.
- Forwarding policies must be unique.
- A maximum of 100 forwarding policies can be configured for a listener. If the number of forwarding policies exceeds the quota, the excess forwarding policies will not be applied.
- When you add a forwarding policy, note the following:
  - Each URL path must exist on the backend server. 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 matching ends when any rule is successfully matched. Matching rules cannot overlap with each other.
  - A URL path cannot be configured for two forwarding policies.
  - A domain name cannot exceed 100 characters.

Adding a Forwarding Policy

1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click  in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. Locate the load balancer you want to add forwarding policy to and click its name.
5. 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 **Forwarding Policies** tab.
6. Click **Add Forwarding Policy**. Configure the parameters based on **Table 1-20**.

Table 1-20 Forwarding policy parameters

Parameter	Type	Description	Example Value
Forwarding Rule	Domain name	Specifies the domain name used for forwarding requests. The domain name in the request must exactly match that in the forwarding policy. You need to specify either a domain name or URL.	www.test.com

Parameter	Type	Description	Example Value
	URL	Specifies the URL used for forwarding requests. There are three URL matching rules: • Exact match: The request URL must exactly match what is specified in the forwarding policy. • Prefix match: The requested URL starts with the specified URL string. • Regular expression match: The URLs are matched using a regular expression.	/login.php
Action	Forward to a backend server group	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

7. Click **Save**.

1.3.3.4 Advanced Forwarding

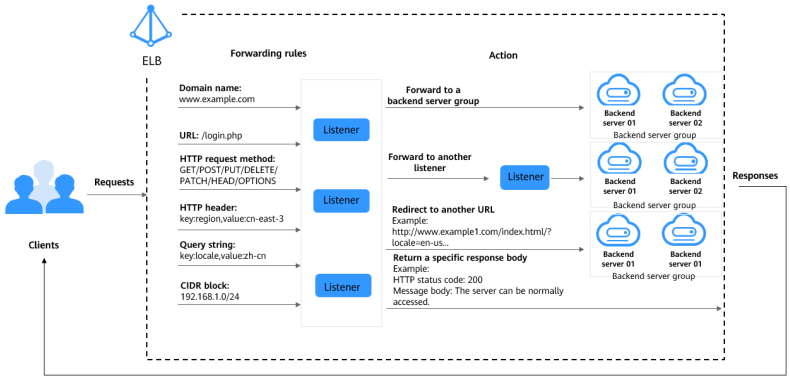
1.3.3.4.1 Advanced Forwarding

Overview

Advanced forwarding policies are available only for dedicated load balancers. If you have enabled **Advanced Forwarding**, you can add advanced forwarding policies to HTTP and HTTPS listeners of dedicated load balancers.

You can add advanced forwarding policies to HTTP or HTTPS listeners to forward requests to different backend server groups based on HTTP request method, HTTP header, query string, or CIDR block in addition to domain names and URLs. **Table 1-21** describes the rules and actions that you can configure for request forwarding.

Figure 1-9 How advanced forwarding works



The following describes how an advanced forwarding policy works:

Step 1 The client sends a request to the load balancer.

Step 2 The load balancer matches the request based on the forwarding rule you configure.

Step 3 The load balancer forwards the request to the corresponding backend server or returns a fixed response to the client based on the action you configure.

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

----End

Table 1-21 Rules and actions supported by an advanced forwarding policy

Forwarding Policy	Description
Forwarding rule	There are six types of forwarding rules: domain name, URL, HTTP request method, HTTP header, query string, and CIDR block For details, see **Forwarding Rule**.
Action	The following actions are supported: forward to a backend server group, redirect to another listener, redirect to another URL, and return a specific response body. For details, see **Action Types**.

How Requests Are Matched

After you add an HTTP or HTTPS listener to a load balancer, a default forwarding policy is generated. This policy uses the protocol and port specified for the listener to match requests and forward the requests to the backend server group you specified when adding the listener.

The default forwarding policy has the lowest priority and is not included when you sort forwarding policies. It can be edited but cannot be deleted.

Each request is matched based on the forwarding policy priority (a smaller value indicates a higher priority). Once a forwarding policy is matched, the request is forwarded based on this forwarding policy.

- 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 Rule

Advanced forwarding policies support the following types of forwarding rules: domain name, URL, HTTP request method, HTTP header, query string, and CIDR block (source IP addresses).

Table 1-22 Forwarding rules

Forwarding Rule	Description
Domain name	• Description Route requests based on the domain name. – You can configure multiple domain names in a forwarding policy. Each domain name contains at least two labels separated by periods (.). Max total: 100 characters. Max label: 63 characters. – Each label can contain letters, digits, hyphens (-), periods (.), and asterisks (*). A label must start with a letter, digit, or asterisk (*) and cannot end with a hyphen (-). An asterisk (*) must be used as the leftmost label if you want to configure a wildcard domain name. • Matching rules Exact match domains and wildcard domains are supported. Example Request URL: https://**www.example.com**/login.php?locale=en-us=#videos Domain name in the forwarding rule: **www.example.com**

Forwarding Rule	Description
URL	• Description Route requests based on URLs. You can configure multiple URLs in a forwarding policy. A URL can contain letters, digits, and special characters _~';@^-%#$.*+?,=!:|\/()[]{}. If the URL contains special characters such as question marks (?) or pound keys (#), escape the special characters before configuring the forwarding rule. • Matching rules – Exact match: The request URL must exactly match that specified in the forwarding policy. The URL must start with a slash (/) and can use asterisks (*) and question marks (?) as wildcards. – Prefix match: The requested URL starts with the specified URL string. The URL must start with a slash (/) and can use asterisks (*) and question marks (?) as wildcard characters. – Regular expression match: The URLs are matched using a regular expression. For more information about URL matching rules, see **URL Matching**. Example Request URL: https://www.example.com/**login.php**?locale=en-us#videos URL in the forwarding rule: **/login.php**
Query string	Route requests based on the query string. A query string consists of a key and one or more values. You need to set the key and values separately. • The key can contain only letters, digits, and special characters !$'()*+,./:;=?@^-_' • A key can have one or more values. 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

Forwarding Rule	Description
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 (-). • A key can have one or more values. 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.

Table 1-23 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** If you select **Redirect to another listener** and create a redirect for the listener, it will redirect the requests to the specified HTTPS listener, but access control configured for the listener will still take 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.

Action	Description
Redirect to another URL	Requests are redirected to the configured URL. When clients access website A, the load balancer returns 302 or any other 3xx status code and automatically redirects the clients to website B. You can custom the redirection URL that will be returned to the clients. 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, digit, or asterisk (*), 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 **URL Matching Based on Regular Expressions**. • **Query String**: A query string can contain only letters, digits, and special characters !$'()*+,./:;=?@&^-_'. Ampersand (&) 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

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 is a string of 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.

URL Matching

Table 1-24 shows how URLs configured in the forwarding policies match the URLs in the requests.

Table 1-24 URL matching examples

Request URL	Forwarding Policy	URL in the Forwarding Policy	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 URL	Forwarding Policy	URL in the Forwarding Policy	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 URL 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 URL 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 URL is /mpl/index.html, it matches forwarding policy 05 exactly, and requests are forwarded to backend server group 05.

URL 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. URL 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 1-25 URL matching based on regular expressions

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

1.3.3.4.2 Managing an Advanced Forwarding Policy

Scenarios

You can add advanced forwarding policies to 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.

- Dedicated load balancers support the following types of forwarding rules: domain name, URL, HTTP request method, HTTP header, query string, and CIDR block (source IP addresses). For details, see **Forwarding Rule**.
- The following actions are supported: forward to a backend server group, redirect to another listener, redirect to another URL, and return a specific response body. 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. 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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Click the **Listeners** tab and click the target listener.
6. On the **Summary** tab, click **Enable** next to **Advanced Forwarding**.
7. Click **OK**.

Adding an Advanced Forwarding Policy

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. On the **Listeners** tab, add a forwarding policy in either of the following ways:
   - Click **Add/Edit Forwarding Policy** in the **Forwarding Policies** column.
   - Locate the target listener, click its name, and click **Forwarding Policies**.
6. Click **Add Forwarding Policy** and configure the parameters based on **Table 1-22** and **Table 1-23**.
7. Click **Save**.

Sorting Forwarding Policies

Multiple forwarding policies can be sorted to set their priorities.

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** > **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. On the **Forwarding Policies** tab, click **Sort**.
7. Drag the forwarding policies to adjust their priorities.
8. Click **Save**.

Modifying a Forwarding Policy

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner to display **Service List** and choose **Networking** > **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. On the **Forwarding Policies** tab, select the forwarding policy, and click **Edit**.
7. 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. 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** > **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. On the **Forwarding Policies** tab, select the forwarding policy and click **Delete** on the top right.
7. In the displayed dialog box, click **OK**.

1.3.3.5 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 HTTP listeners.

Table 1-26 Transfer headers

Header	Feature	Description	Dedicated Load Balancers
X-Forwarded-ELB-IP	Transfer Load Balancer EIP	If this option is enabled, the EIP bound to the load balancer will be transmitted to backend servers through the X-Forwarded-ELB-IP header. The format is as follows (XX.XXX.XX.XXX indicates the EIP of the load balancer): X-Forwarded-ELB-IP: *XX.XXX.XX.XXX*	Supported
X-Forwarded-Port	Transfer Listener Port Number	If this option is enabled, the port number used by the listener will be transmitted to backend servers through the X-Forwarded-Port header.	Supported
X-Forwarded-For-Port	Transfer Port Number in the Request	If this option is enabled, the port number used by the client will be transmitted to backend servers through the X-Forwarded-For-Port header.	Supported

Table 1-27 Rewrite headers

Header	Feature	Description	Dedicated Load Balancers
X-Forwarded-Host	Rewrite X-Forwarded-Host	• If this option is enabled, the Host header of the client request will be rewritten into the X-Forwarded-Host header and transmitted to the backend servers. • If this option is disabled, the X-Forwarded-Host header of the client will be transmitted to the backend servers.	Supported

NOTE

- More HTTP headers are coming soon. See the available HTTP headers on the management console.
- √ indicates the load balancer supports the header, whereas × indicates the load balancer does not support the header.

Enabling HTTP/HTTPS Headers

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** > **Elastic Load Balance**.
4. You can enable these header features in either of the following ways:
   - On the **Load Balancers** page, locate the load balancer and click its name. Under **Listeners**, click **Add Listener**.
   - On the **Load Balancers** page, locate the load balancer and click **Add Listener** in the **Operation** column.
5. On the **Configure Listener** page, expand **Advanced Settings** and enable the features as needed.
6. Configure the listener as prompted.
7. Confirm the configuration and click **Submit**.

Modifying HTTP Header Features

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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Click the **Listeners** tab, locate the target listener and click **Edit** in **Operation** column.
6. On the displayed page, expand **Advanced Settings** and enable or disable the features.
7. Click **OK**.

1.3.4 Modifying 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 load balancer by referring to **Creating a Dedicated Load Balancer**.
- 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. 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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Click **Listeners** tab, locate the listener, and click its name.
6. On the **Summary** tab, click **Configure** next to **Modification Protection**.
7. 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. 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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. 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.
6. 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. 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** > **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 the name of the listener.
6. On the **Summary** tab, click **Edit** on the top right.
7. In the **Edit** dialog box, expand **Advanced Settings**.
8. Configure **Idle Timeout (s)**, **Request Timeout (s)**, or **Response Timeout (s)** as you need.
9. Click **OK**.

Changing the Backend Server Group of a Listener

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

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

   a. Click the name of the backend server group or enter the name in the search box to search for the target group.
   b. Click **Create Backend Server Group**. After the backend server group is created, click the refresh icon.

   NOTE

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

Deleting a Listener

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

4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Click the **Listeners** tab, locate the listener, and click **Delete** in the **Operation** column.
6. In the displayed dialog box, enter **DELETE**.
7. Click **OK**.

1.4 Backend Server Group

1.4.1 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. A backend server can be an ECS, BMS, supplementary network interface, or IP address.

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

Table 1-28 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 an ECS, BMS, supplementary network interface, or IP address 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.

Figure 1-10 Backend server group architecture

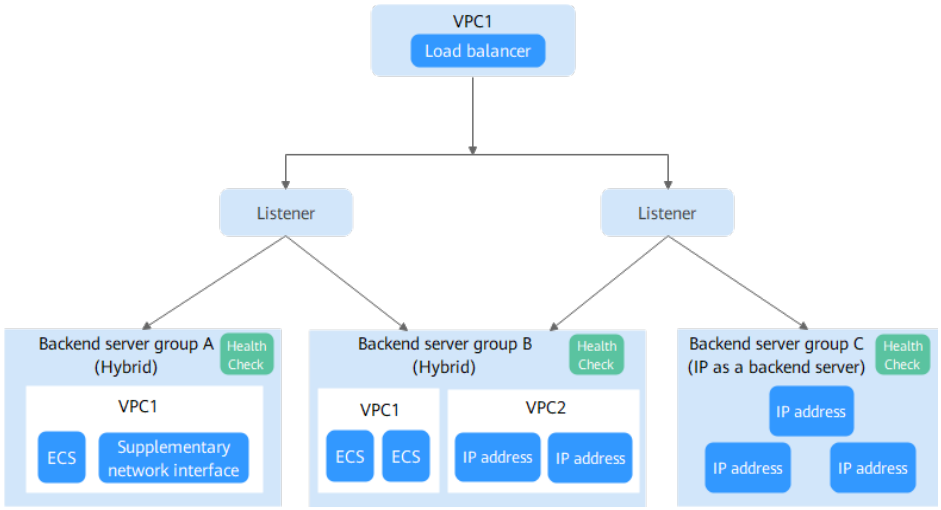


Table 1-29 Backend server group types

Backend Server Group Type	Backend Server Type	Example	Reference
Hybrid	• ECSs, BMSs, or 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 servers or supplementary network interfaces in VPC1. • In backend server group B, you can add IP addresses in VPC2 as backend servers.	• **Adding a Cloud Server** • **Adding Supplementary Network Interfaces** • **Adding IP Addresses as Backend Servers**
IP as a backend server	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**, IP addresses can be added to backend server group C as backend servers.	**Adding IP Addresses 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**: Traffic is routed only to healthy backend servers in the backend server group.

Key Functions

You can configure the key functions listed in **Table 1-30** for each backend server group to ensure service stability.

Table 1-30 Key functions

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

Precautions for Creating a Backend Server Group

The backend protocol of the new backend server group must match the frontend protocol of the listener as described in **Table 1-31**.

You can create a backend server group by referring to **Table 1-32**.

Table 1-31 The frontend and backend protocol

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

Table 1-32 Creating a backend server group

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

1.4.2 Key Features

1.4.2.1 Health Check

ELB periodically sends requests to backend servers to check whether they can process requests. This process is called health check.

If a backend server is detected unhealthy, the load balancer will stop route requests to it. After the backend server recovers, the load balancer will resume routing requests to it.

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

Health Check Protocol

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

If you want to modify health check settings, see details in **Enabling or Disabling Health Check**.

Select a health check protocol that matches the backend protocol as described in **Table 1-33**.

Table 1-33 The backend protocol and health check protocols (dedicated load balancers)

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

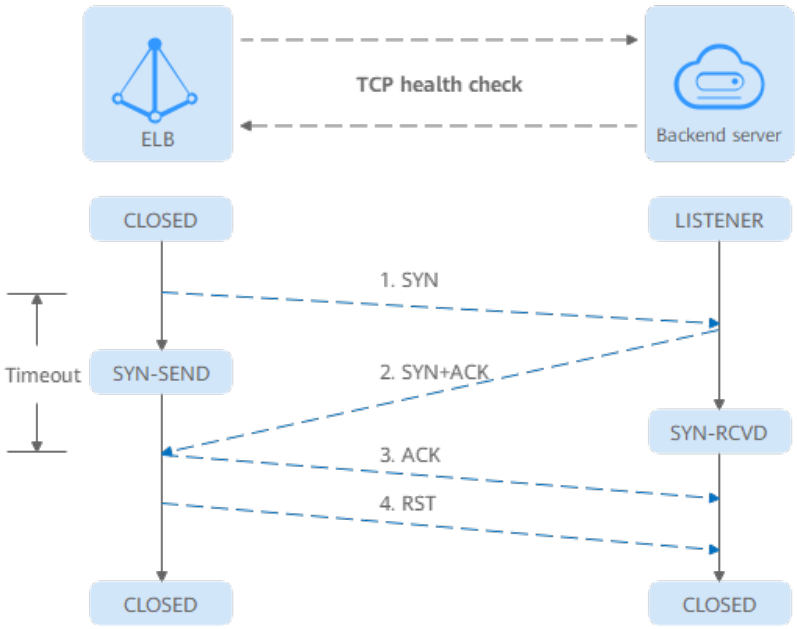
Health Check Source IP Address

A dedicated load balancer uses the IP addresses in its backend subnet to send requests to backend servers and verify their health status. To perform health checks, 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

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

Figure 1-11 TCP health check



The TCP health check process is as follows:

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

NOTICE

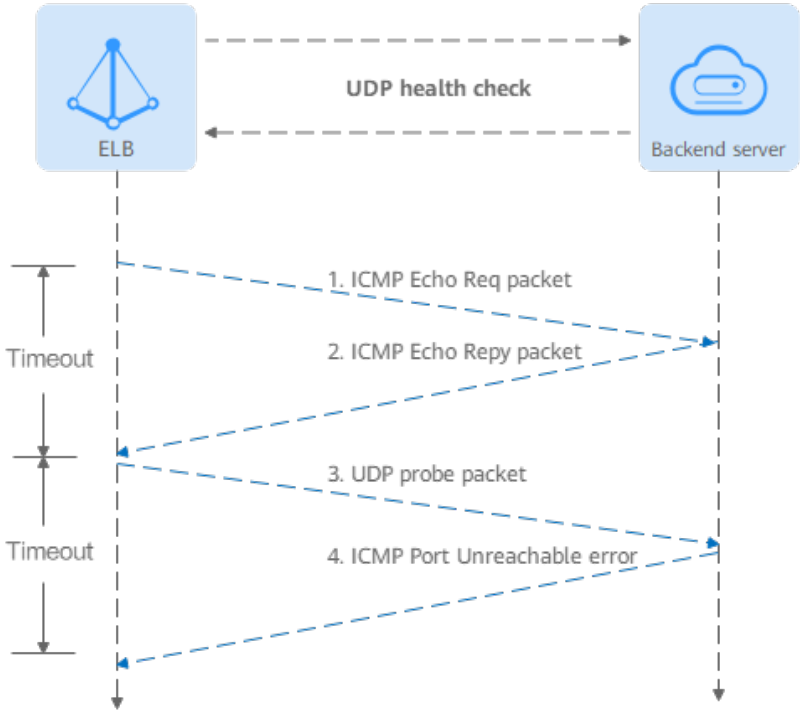
After a successful TCP three-way handshake, an RST packet will be sent to close the TCP connection. The application on the backend server may consider this packet a connection error and reply with a message, for example, "Connection reset by peer". To avoid this issue, take either of the following actions:

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

UDP Health Check

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

Figure 1-12 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.

NOTE

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. This causes the health check result to be different from the actual health of the backend server.

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 1-13** shows how an HTTP health check works.

Figure 1-13 HTTP health check

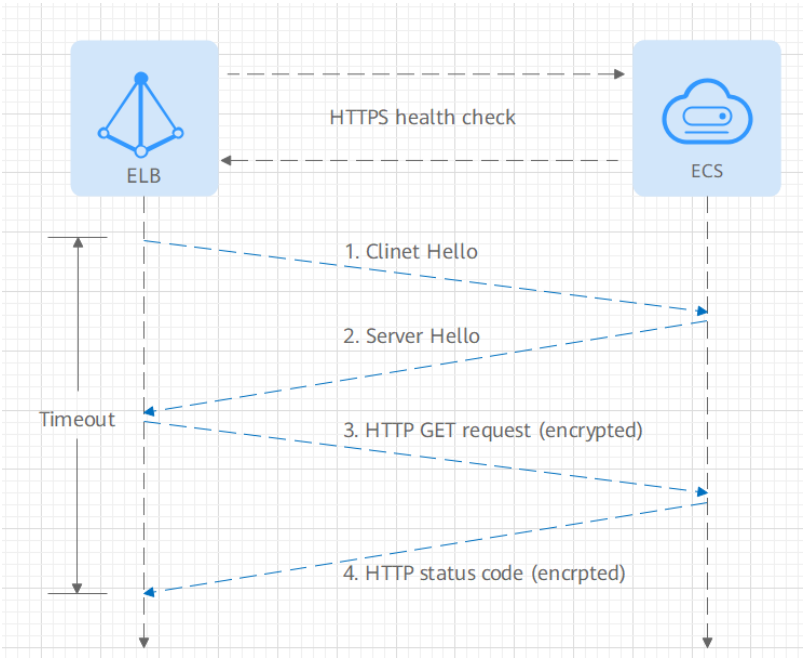
The HTTPS health check process is as follows:

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

HTTPS Health Check

For TCP, HTTP, and HTTPS backend protocols, you can use HTTPS to establish an SSL connection over TLS handshakes to obtain the statuses of backend servers. **Figure 1-14** shows how an HTTPS health check works.

Figure 1-14 HTTPS health check



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.

gRPC Health Check

Figure 1-15 gRPC health check

The gRPC health check process is as follows:

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

Health Check Time Window

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

The health check time window is determined by the factors in **Table 1-34**.

Table 1-34 Factors affecting the health check time window

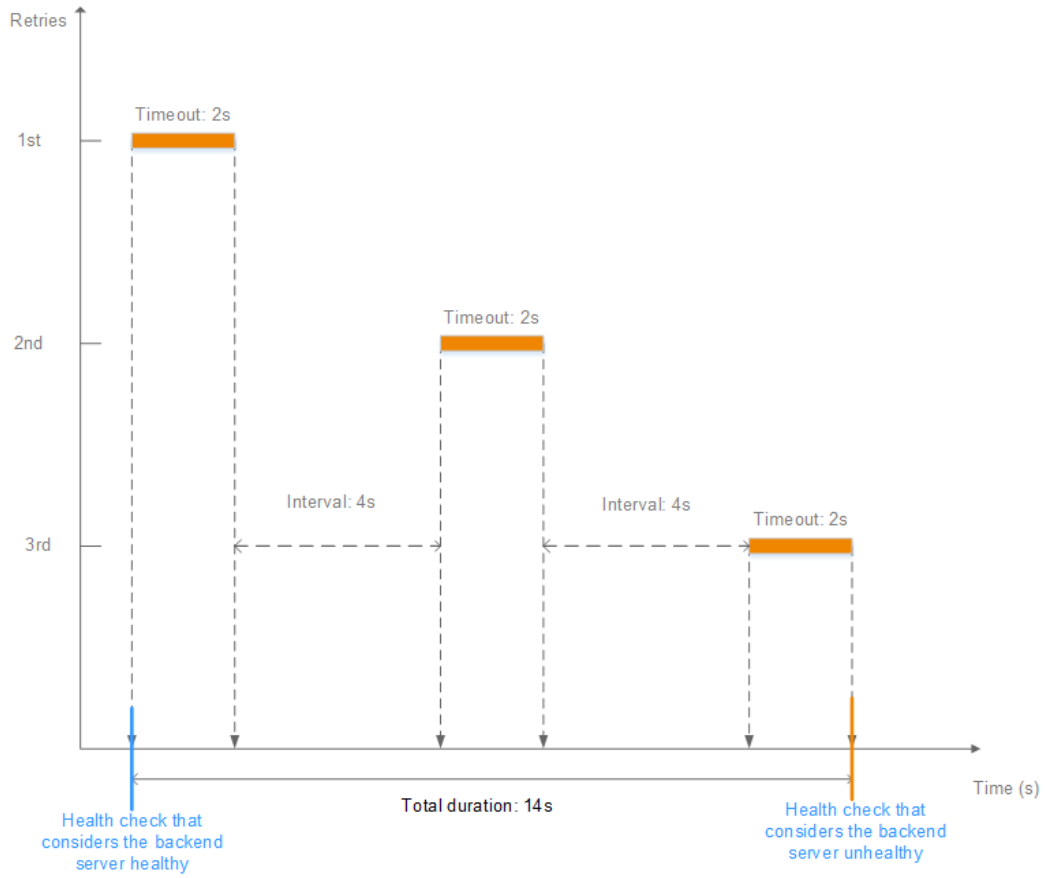
Factor	Description
Check Interval	How often health checks are performed.
Timeout Duration	How long the load balancer waits for the response from the backend server.
Health Check Threshold	The number of consecutive successful or failed health checks required for determining whether the backend server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

As shown in **Figure 1-16**, if the health check interval is 4s, the health check timeout duration is 2s, and unhealthy threshold is 3, the time window for a backend server to be considered unhealthy is calculated as follows: 2 x 3 + 4 x (3 – 1) = 14s.

Figure 1-16 Health check timeout duration



Rectifying an Unhealthy Backend Server

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

1.4.2.2 Load Balancing Algorithms

Overview

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

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

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

Table 1-35 Load balancing algorithms

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

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 1-17 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 1-17 Traffic distribution using the weighted round robin algorithm

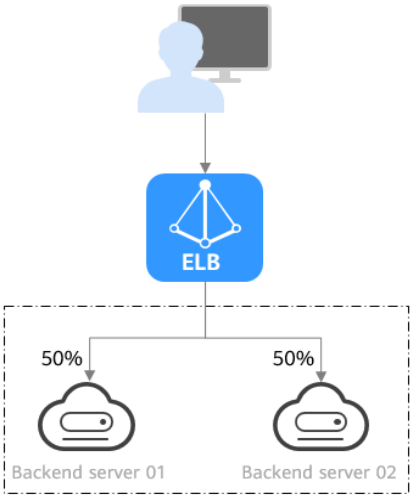


Table 1-36 Weighted round robin

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 1-18 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 1-18 Traffic distribution using the weighted least connections algorithm

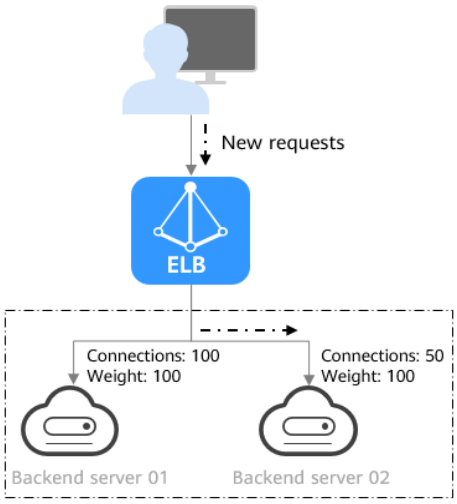


Table 1-37 Weighted least connections

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 1-19 shows an example of how requests are distributed using the source IP hash algorithm. Two backend servers are in the same AZ and have the same weight. If backend server 01 has processed a request from IP address A, the load balancer will route new requests from IP address A to backend server 01.

Figure 1-19 Traffic distribution using the source IP hash algorithm

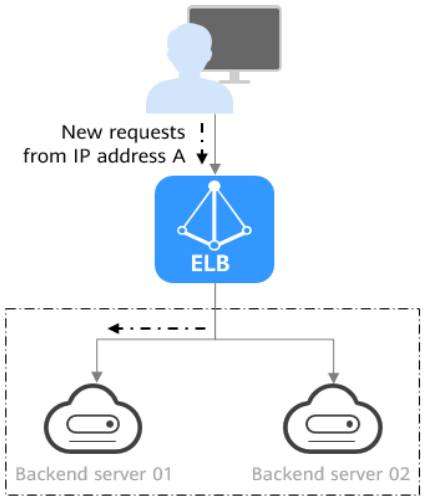


Table 1-38 Source IP hash

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 1-20 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 1-20 Traffic distribution using the connection ID algorithm

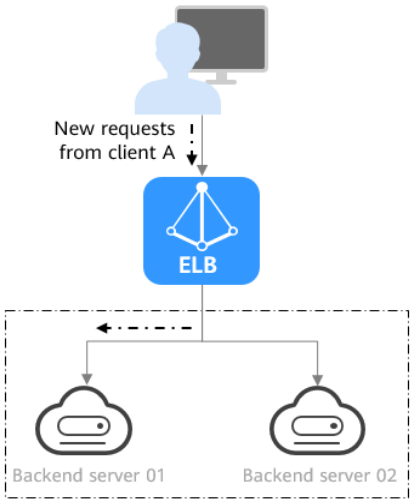


Table 1-39 Connection ID

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

1.4.2.3 Sticky Session

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

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

Differences Between Sticky Sessions at Layer 4 and Layer 7

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

Table 1-40 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 are 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.
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.	• 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**.

Constraints and Limitations

- If you use **Cloud Connect connection**, **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.

1.4.2.4 Forwarding Mode (Dedicated Load Balancers)

The load balancer routes the traffic across backend servers based on the forwarding mode. There are two options: **Load balancing** and **Active/Standby**.

NOTE

- This feature is only available for backend server groups that are bound to dedicated load balancers.

Table 1-41 Forwarding modes

Forwarding Mode	Description	When to Use
Load balancing	You can add multiple backend servers to a backend server group. And then the load balancer distributes requests across these backend servers based on the load balancing algorithm configured for this backend server group.	You want your load balancer to forward requests based on the forwarding policies configured for the listener.

Forwarding Mode	Description	When to Use
Active/Standby	You must add two backend servers to the backend server group, one acting as the active server and the other as the standby server. Active/Standby forwarding requires at least one healthy backend server. The load balancer routes the traffic to the active server if it works normally. If the active server becomes unhealthy, the load balancer then routes the traffic to the standby server.	You need higher service availability.

1.4.2.5 Slow Start (Dedicated Load Balancers)

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

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

NOTE

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

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

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

Constraints

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

1.4.3 Creating a Backend Server Group

Scenario

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

You can create a backend server group for a load balancer in any of the ways described in **Table 1-42**.

Table 1-42 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 protocol of the new backend server group must match the frontend protocol of the listener as described in **Table 1-43**.

Table 1-43 The frontend and backend protocol

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

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click  in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. Click **Create Backend Server Group** in the upper right corner.
6. Configure the routing policy based on **Table 1-44**.

Table 1-44 Parameters required for configuring a routing policy

Parameter	Description
Load Balancing Type	Specifies the type of load balancers that can use the backend server group. Select **Dedicated**.
Load Balancer	Specifies whether to associate a load balancer. You can associate an existing dedicated load balancer when you create a backend server group or associate one later. • Associate later • Associate existing
Forwarding Mode	Specifies the forwarding mode to distribute traffic. There are two options: **Load balancing** and **Active/Standby**. • **Load balancing**: You can add one or more backend servers to the backend server group. • **Active/Standby**: You can add only two backend servers to the backend server group, one acting as the active server and the other as the standby server. If the active server is faulty, traffic is forwarded to the standby server, improving service reliability.

Parameter	Description
Backend Server Group Type	Specifies the type of the backend server group. • **Hybrid**: You can add ECSs and supplementary network interfaces as backend servers, or add IP addresses as servers when **IP as a Backend** is enabled. When you create a hybrid backend server group, you must specify a VPC and associate the backend server group with a load balancer in this VPC. • **IP as a backend server**: You can add IP addresses as backend servers only when you enable **IP as a Backend**.
Backend Server Group Name	Specifies the name of the backend server group.
VPC	Specifies the VPC where the backend server group works. You can associate the backend server group with a load balancer in this VPC. This parameter is mandatory if you select **Hybrid** for **Backend Server Group Type**. You can select an existing VPC or create a new one. For more information about VPC, see the ***Virtual Private Cloud User Guide***.
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: • **Load balancing**: HTTP, HTTPS, gRPC, TCP, UDP, and QUIC • **Active/Standby**: TCP, UDP, and QUIC
Forward to Same Port	If this option is enabled, you do not need to specify a backend port when you add a backend server. The listener routes the requests to the backend server over the same port as the frontend port. This option cannot be disabled after being enabled. **NOTE** This option is available only for TCP, UDP, or QUIC backend server groups associated with a dedicated load balancer.

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 different connection IDs to be routed to different backend servers and ensures that requests with the same connection ID are routed to the same backend server. 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 **Sticky Session**. **NOTE** TLS backend server groups do not support sticky sessions.

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 type: • **Source IP address**: The source IP address of each request is calculated using the consistent hashing algorithm to obtain a unique hashing key, and all backend servers are numbered. The system allocates the client to a particular server based on the generated key. This allows requests from the same IP address are forwarded to the same backend server. • **Load balancer cookie**: The load balancer generates a cookie after receiving a request from the client. All subsequent requests with the cookie are routed to the same backend server. **NOTE** • **Source IP address** is available when you have selected **TCP**, **QUIC,** or **UDP** for **Backend Protocol**. • **Load balancer cookie** is available when you have selected **HTTP** or **HTTPS** for **Backend Protocol**.
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 (Dedicated Load Balancers)**.
Slow Start Duration (s)	Specifies how long the slow start will last, in seconds. This parameter is mandatory if **Slow Start** is enabled.

Parameter	Description
Description	Provides supplementary information about the backend server group.

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

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

   Configure health check for the backend server group based on **Table 1-45**. For more information about health checks, see **Health Check**.

   Table 1-45 Parameters required for configuring a health check

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

Parameter	Description
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 (?), number signs (#), percent signs (%), ampersands (&), and extended character sets _;~!. () *[]@$^:',+
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**.
Status Code	Specifies the status codes that will be returned to the load balancer to indicate the health of backend servers. This parameter is available only when you set the health check protocol to HTTP or HTTPS. You can enter a unique number or a positive number range within the status code range, for example, 0-10 and 200-300. A maximum of five HTTP status codes are supported. If there is more than one status code, press **Enter** to separate them. ● If the health check protocol is HTTP or HTTPS, the status code ranges from 200 to 599. ● When the gRPC protocol is used, the status code ranges from 0 to 99. **NOTE** This feature will be available in more regions. See details on the management console.

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

Related Operations

You can associate the backend server group with the listener of a dedicated load balancer in either ways listed in **Table 1-42**.

1.4.4 Modifying a Backend Server Group

1.4.4.1 Overview

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

Health Check

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

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

For details about how to modify health check settings, see **Enabling or Disabling Health Check**.

Basic Information

You can modify the basic information of a backend server group listed in **Table 1-46**.

Table 1-46 Basic information that can be modified

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

Parameter	Description
Description	Change the description of the backend server group by performing the operations in **Changing the Load Balancing Algorithm**.

1.4.4.2 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 and Notes

- The health check protocol can be different from the backend protocol.
- To reduce the vCPU usage of the backend servers, it is recommended that you use TCP for health checks. If you want to use HTTP for health checks, you can use static files to return the health check results.
- If health check is enabled, security group rules must allow traffic from the health check port to the backend servers over the health check protocol. 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. 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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** tab, locate the backend server group.
6. On the **Summary** page, click **Health Check** on the right.

7. In the **Configure Health Check** dialog box, configure the parameters based on **Table 1-47**.

Table 1-47 Parameters required for configuring health check

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

Parameter	Description	Example Value
Path	Specifies the health check URL, which is the destination on backend servers for health checks. This parameter is mandatory if the health check protocol is HTTP or HTTPS. The path can contain 1 to 80 characters and must start with a slash (/). 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
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

Parameter	Description	Example Value
Status Code	Specifies the status codes that will be returned to the load balancer to indicate the health of backend servers. This parameter is available only when you set the health check protocol to HTTP, gRPC, or HTTPS. You can enter a unique number or a positive number range within the status code range, for example, 0-10 and 200-300. A maximum of five HTTP status codes are supported. If there is more than one status code, press **Enter** to separate them. ● If the check protocol is HTTP or HTTPS, the status code ranges from 200 to 599. ● When the gRPC protocol is used, the status code ranges from 0 to 99. **NOTE** This feature will be available in more regions. See details on the management console.	200

8. Click **OK**.

Disabling Health Check

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

1.4.4.3 Changing the Load Balancing Algorithm

Scenario

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

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

Procedure

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

NOTE

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

1.4.4.4 Modifying Sticky Session Settings

Scenario

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

NOTE

You can also configure sticky sessions when adding a listener or creating a backend server group.

Enabling Sticky Session

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

5. On the **Backend Server Groups** page, locate the backend server group and click **Edit** in the **Operation** column.
6. In the **Modify Backend Server Group** dialog box, enable sticky session, select the sticky session type, and set the session stickiness duration.
7. Click **OK**.

Disabling Sticky Session

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

1.4.4.5 Modifying Slow Start Settings

Scenario

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

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

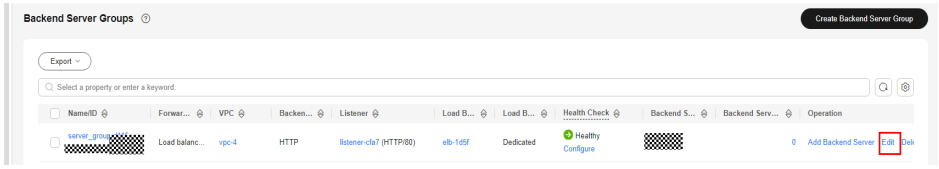
NOTE

You can also configure slow start when adding a listener or creating a backend server group.

Enabling Slow Start

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

Figure 1-21 Modifying a backend server group



6. In the **Modify Backend Server Group** dialog box, enable slow start and set the slow start duration.

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

7. Click **OK**.

Disabling slow start

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

1.4.5 Changing a Backend Server Group

Scenario

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

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

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

Constraints and Limitations

- The backend server group cannot be changed if redirection is enabled.
- The backend protocol of the backend server group must match the frontend protocol of the listener. For details, see **Table 1-31**.

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. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the target load balancer and click its name.
5. On the **Listeners** tab, locate the target listener and click its name.
6. On the **Summary** tab, click **Change Backend Server Group** on the right of **Default Backend Server Group** area.
7. In the displayed dialog box, click the server group name box.

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

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

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

   NOTE

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

1.4.6 Viewing a Backend Server Group

Scenario

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

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

Procedure

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

a. On the **Summary** tab, view the basic information and health check settings.
b. On the **Backend Servers** tab, view the servers that have been added to the backend server group.
c. On the **Associated Resources** tab, view the resources associated with the backend server group.

1.4.7 Deleting a Backend Server Group

Scenario

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

Constraints and Limitations

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

Procedure

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

1.5 Backend Server

1.5.1 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 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.

Different types of backend servers can be added to different types of backend server groups as described in **Table 1-48**.

Table 1-48 Backend server group and backend server types

Backend Server Group Type	Backend Server Types	Reference
Hybrid	• Cloud servers or supplementary network interfaces that are in the same VPC as the load balancer, if **IP as a Backend** is disabled • IP addresses of servers in other VPCs or in your on-premises data center, if **IP as a Backend** is enabled **NOTE** When you create a hybrid backend server group, you must specify a VPC and associate the backend server group with a load balancer in this VPC.	• **Adding a Cloud Server** • **Adding Supplementary Network Interfaces** • **Adding IP Addresses as Backend Servers**
IP as a backend server	IP addresses of cloud or on-premises servers **NOTE** **IP as a Backend** must have been enabled for the load balancer.	**Adding IP Addresses as Backend 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.
- 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.

Constraints and Limitations

- A maximum of 500 backend servers can be added to a backend server group.
- Inbound security group rules must be configured to allow traffic over the port of each backend server and health check port. For details, see **Security Group 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 1-49 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.

1.5.2 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 must allow traffic from the backend subnet where the load balancer resides to the backend servers. (By default, the backend subnet of a load balancer is the same as the subnet where the load balancer resides.) For details about how to configure security group rules, see **Configuring Security Group Rules**.
- Network ACL rules are optional for subnets. If network ACL rules are configured for the backend subnet of the load balancer, the rules must allow traffic from the backend subnet of the load balancer to the backend servers. For details about how to configure network ACL rules, see **Configuring Network ACL Rules**.

NOTE

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

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 and Limitations

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

Configuring Security Group Rules

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

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner to display **Service List** and choose **Compute** > **Elastic Cloud Server**.
4. In the ECS list, click the name of the ECS whose security group rules you want to modify.

   The ECS details page is displayed.
5. Click **Security Groups**, locate the security group, and view security group rules.

6. Click the ID of a security group rule or **Modify Security Group Rule**. The security group details page is displayed.
7. On the **Inbound Rules** tab, click **Add Rule**. Configure an inbound rule based on **Table 1-50**.

Table 1-50 Security group rules

Backend Protocol	Policy	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	
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 is still need to be allowed to backend servers.
- Traffic from the new backend subnet is also need to be allowed to backend servers.

8. Click **OK**.

Configuring Network ACL Rules

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

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

- 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 an inbound or outbound rule.
   - **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 from all IP addresses is permitted.
   - **Destination Port Range**: Select a port range.
   - (Optional) **Description**: Describe the network ACL rule.
7. Click **OK**.

1.5.3 Cloud Servers

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.

Constraints

- Cloud servers can only be added to a hybrid backend server group.
- The cloud servers must be in the same VPC as the backend server group.

Adding a Cloud Server

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

6. Switch to the **Backend Servers** tab and click **Add** above the cloud server list.
7. Search for cloud servers using keywords.
   Select the cloud servers you want to add and click **Next**.
8. Specify the weights and ports for the cloud servers and click **Finish**.
   You can set backend ports in batches.

Modifying Cloud Server Ports/Weights

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**.

NOTE

Only certain regions support backend port modification. See the details on the management 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 **Networking** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the target backend server group.
6. On the **Backend Servers** tab, click **Backend Servers**.
7. Select the cloud servers and click **Modify Port/Weight** up above the cloud server list.
8. 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.

9. 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.

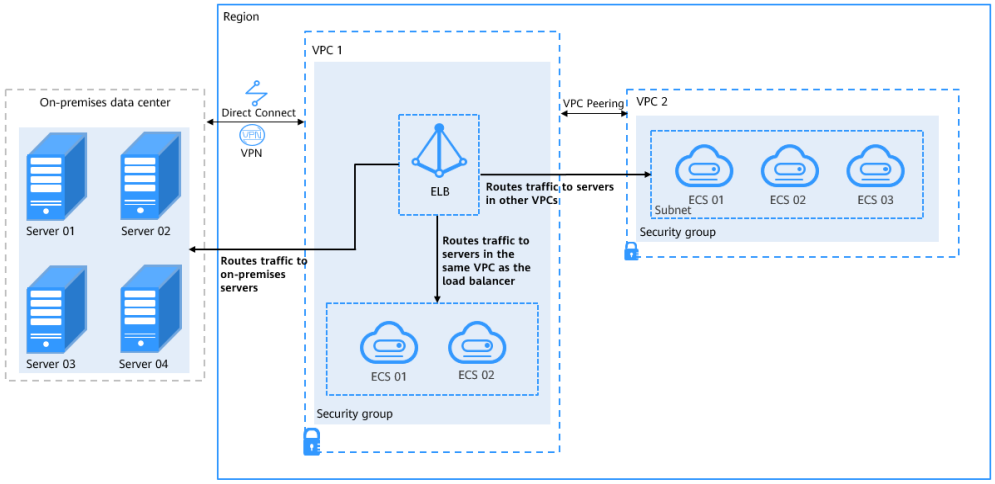
1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click  in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the target backend server group.
6. Switch to the **Backend Servers** tab and click **Backend Servers**.
7. Select the cloud servers you want to remove and click **Remove** above the cloud server list.
8. In the displayed dialog box, click **OK**.

1.5.4 IP Addresses as Backend Servers

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

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

Figure 1-22 Routing requests to cloud and on-premises servers



Constraints and Limitations

- The cross-VPC backend function of the IP address type cannot be disabled after being enabled.
- Only private IPv4 addresses can be added as backend servers.
- **IP as a Backend** cannot be disabled after it is enabled.
- 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.

Scenarios

After you enable **IP as a Backend**, you can add backend servers by using their IP addresses. You need to get prepared for different scenarios as shown in **Table 1-51**.

Table 1-51 Adding IP addresses as backend servers

Where Servers Are Running	Preparations
In a different VPC from the load balancer	Set up a VPC peering connection between the VPC where the load balancer is running and the VPC where the servers are running. For details about how to set up a VPC peering connection, see the **Virtual Private Cloud User Guide**.
In the same VPC as the load balancer	Set up a VPC peering connection for the VPC where the load balancer and the servers are running, and then add routes for the VPC peering connection. For details, see **Routing Traffic to Backend Servers in the Same VPC as the Load Balancer**.

Where Servers Are Running	Preparations
In on-premises data centers	Connect the on-premises data center to the VPC where the load balancer is running through Direct Connect or VPN. For details about how to connect on-premises data centers to the cloud, see the **Direct Connect User Guide** or **Virtual Private Network User Guide**.

Enabling IP as a Backend

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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. On the **Summary** tab, click **Enable** next to **IP as a Backend**.
6. Click **OK**.

Adding IP Addresses as Backend Servers

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

Modifying the Ports/Weights of IP Addresses 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**.

NOTE

Only certain regions support backend port modification. See the details on the management 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 **Networking** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the target backend server group.
6. Switch to the **Backend Servers** tab and click **IP as Backend Servers**.
7. Select the servers and click **Modify Port/Weight** up the server list.
8. In the displayed dialog box, modify ports/weights as you need.
   - Modifying ports:
     - Modifying the port of an IP address as backend server: Set the port in the **Backend Port** column.
     - Modifying the ports of multiple IP address as backend servers: Set the port next to **Batch Modify Ports**, and click OK.
   - Modifying weights:
     - Modifying the weight of an IP address as backend server: Set the weight in the **Weight** column.
     - Modifying the weights of multiple IP address 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.
9. Click **OK**.

Removing IP Addresses 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. 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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the target backend server group.
6. Switch to the **Backend Servers** tab and click **IP as Backend Servers**.
7. Select the IP addresses as backend servers to be removed and click **Remove** above the server list.
8. In the displayed dialog box, click **OK**.

1.5.5 Supplementary Network Interfaces

In addition to cloud servers and IP addresses as servers, you can also add supplementary network interfaces to a backend server group.

Supplementary network interfaces allow you to configure more NICs than a cloud server would normally support. They can be attached to VLAN subinterfaces of elastic network interfaces.

For details about supplementary network interfaces, see the *Virtual Private Cloud User Guide*.

Constraints and Limitations

Supplementary network interfaces can only be added to a hybrid backend server group.

Adding Supplementary Network Interfaces

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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the backend server group that you want to add supplementary network interfaces to.
6. Switch to the **Backend Servers** tab and click **Add** above the supplementary network interface list.

   Select the supplementary network interfaces and click **Next**. You can search for supplementary network interfaces by ID, private IP address, network interface private IP address, subnet name, or subnet ID.
7. Specify the weights and ports for the supplementary network interfaces and click **Finish**.

Changing the Ports/Weights of Supplementary Network Interfaces

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. 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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the backend server group.
6. Switch to the **Backend Servers** tab and click **Supplementary Network Interfaces**.
7. Select the supplementary network interfaces and click **Modify Port/Weight** above the supplementary network interface list.
8. In the displayed dialog box, modify weights and ports as you need.
   - Modifying ports:
     - Modifying the port of a supplementary network interface: Set the port in the **Backend Port** column.
     - Modifying the ports of multiple supplementary network interfaces: Set the port next to **Batch Modify Ports** and click **OK**.
   - Modifying weights:
     - Changing the weight of a supplementary network interface: Set the weight in the **Weight** column.
     - Modifying the weights of multiple supplementary network interfaces: Set the weight next to **Batch Modify Weights** and click **OK**.

     NOTE

     You can set the weights of multiple supplementary network interfaces to **0** to block them from receiving requests routed by each load balancer.
9. Click **OK**.

Removing Supplementary Network Interfaces

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. 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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the backend server group.
6. Switch to the **Backend Servers** tab and click **Supplementary Network Interfaces**.
7. Select the supplementary network interfaces and click **Remove** above the list.
8. In the displayed dialog box, click **OK**.

1.6 Security

1.6.1 Transfer Client IP Address

Overview

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

Table 1-52 lists whether you can enable or disable this feature.

Table 1-52 Transfer client IP address support

Listener Type	Enabling Transfer Client IP Address	Disabling Transfer Client IP Address
TCP and UDP	Enabled by default	Not supported
HTTP and HTTPS	Enabled by default	Not supported

Constraints

- 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 **Transfer Client IP Address** is enabled, traffic, such as unidirectional data transmission or push traffic, may be interrupted when backend servers are being migrated. After backend servers are migrated, retransmit the packets to restore the traffic.

- If you add IP addresses as backend servers, the source IP addresses of the clients cannot be passed to these servers. Install the **TOA module** to obtain source IP addresses.

Alternatives for Obtaining the IP Address of a Client

You can obtain the IP address of a client in the ways listed in **Table 1-53**.

Table 1-53 Alternatives

Listener Type	Alternatives
TCP	**Configuring the TOA Module**
HTTP and HTTPS	**Layer 7 Load Balancing**

1.6.2 HTTP/2

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.

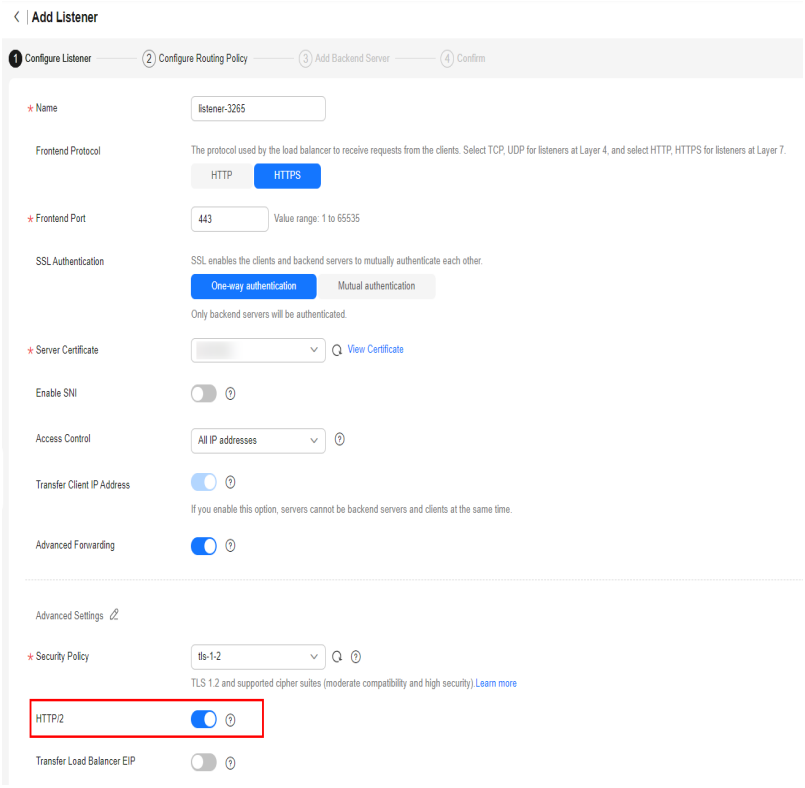
Enabling HTTP/2 When Adding a Listener

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

1. 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** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**.
6. In the **Add Listener** dialog box, set **Frontend Protocol** to **HTTPS**.

7. Expand **Advanced Settings** and enable HTTP/2.
8. Confirm the configurations and go to the next step.

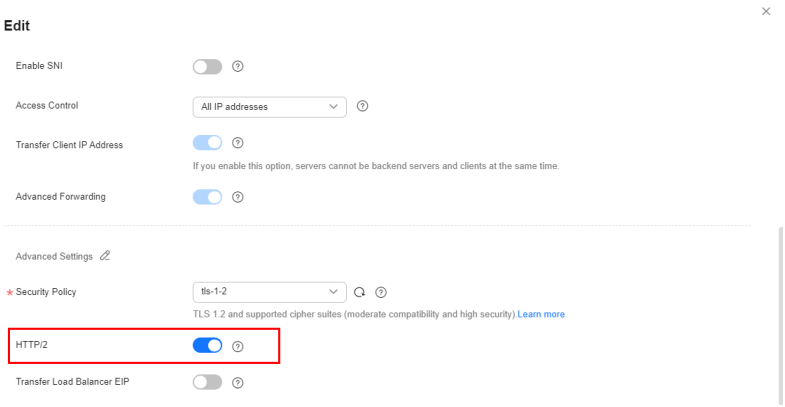
Figure 1-23 Enabling HTTP/2



Enabling or Disabling HTTP/2 for an Existing Listener

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

Figure 1-24 Disabling or enabling HTTP/2



1.6.3 TLS Security Policy

HTTPS encryption is commonly used for applications that require secure transmission of data, 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 to improve security.

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

Default Security Policy

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 1-54 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

Security Policy	TLS Versions	Cipher Suites
hybrid-policy-1-0	TLS 1.2 TLS 1.1	• ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • AES128-GCM-SHA256 • AES256-GCM-SHA384 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • AES128-SHA256 • AES256-SHA256 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA • AES128-SHA • AES256-SHA • ECC-SM4-SM3 • ECDHE-SM4-SM3
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 by ELB and clients are used, and the cipher suites supported by ELB take precedence.

Differences Among Default Security Policies

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

Table 1-55 Differences among default 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	hybrid-policy-1-0
TLS version									
Protocol-TLS 1.3	×	×	×	×	×	√	√	√	×
Protocol-TLS 1.2	√	√	√	√	√	√	√	√	√
Protocol-TLS 1.1	√	√	×	√	×	√	×	×	√
Protocol-TLS 1.0	√	×	×	√	×	√	×	×	×
Cipher suite									
EDHE-RSA-AES128-GCM-SHA256	√	√	√	×	√	×	×	×	×
ECDHE-RSA-AES256-GCM-SHA384	√	√	√	√	√	√	√	√	√
ECDHE-RSA-AES128-SHA256	√	√	√	√	√	√	√	√	√
ECDHE-RSA-AES256-SHA384	√	√	√	√	√	√	√	√	√
AES128-GCM-SHA256	√	√	√	√	√	√	×	×	√
AES256-GCM-SHA384	√	√	√	√	√	√	×	×	√
AES128-SHA256	√	√	√	√	√	√	×	×	√
AES256-SHA256	√	√	√	√	√	√	×	×	√
ECDHE-RSA-AES128-SHA	√	√	√	√	×	√	×	×	√
ECDHE-RSA-AES256-SHA	√	√	√	√	×	√	×	×	√
AES128-SHA	√	√	√	√	×	√	×	×	√

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	hybrid-policy-1-0
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	×	×	×	×	×	√	√	√	×
TLS_CHACHA20_POLY1305_SHA256	×	×	×	×	×	√	√	√	×
TLS_AES_128_GCM_SHA256	×	×	×	×	×	√	√	√	×
TLS_AES_128_CCM_8_SHA256	×	×	×	×	×	√	√	√	×
TLS_AES_128_CCM_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	hybrid-policy-1-0
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	×	×	×	√	×	×	×	×	×
ECC-SM4-SM3	×	×	×	×	×	×	×	×	√
ECDHE-SM4-SM3	×	×	×	×	×	×	×	×	√

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. 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** > **Elastic Load Balance**.

4. In the navigation pane on the left, choose **TLS Security Policies**.
5. On the displayed page, click **Create Custom Security Policy** in the upper right corner.
6. Configure the parameters based on **Table 1-56**.

Table 1-56 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	Provides supplementary information about the custom security policy.

7. 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 version, cipher suite, and description of a custom security policy as required.

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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **TLS Security Policies**.
5. On the **TLS Security Policies** page, click **Custom Security Policies**, locate the custom security policy, and click **Modify** in the **Operation** column.
6. In displayed dialog box, modify the custom security policy as described in **Table 1-56**.
7. 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. Disassociate the security policy from the listener first.

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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **TLS Security Policies**.
5. On the **TLS Security Policies** page, click **Custom Security Policies**, locate the custom security policy, and click **Delete** in the **Operation** column.
6. In the displayed dialog box, click **OK**.

Selecting a Security Policy for an HTTPS Listener

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

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

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

Changing a Security Policy for an HTTPS Listener

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner to display **Service List** and choose **Networking** > **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. On the **Summary** tab, click **Edit** on the top right.
7. In the **Edit** dialog box, expand **Advanced Settings** and change the security policy.
8. Click **OK**.

1.6.4 Access Control

1.6.4.1 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 backend servers 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

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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Configure access control for a listener in either of the following ways:
   - On the **Listeners** page, locate the listener and click **Configure** in the **Access Control** column.

- Click the name of the target listener. On the **Summary** page, click **Configure** on the right of **Access Control**.

6. In the displayed **Configure Access Control** dialog box, configure parameters as described in **Table 1-57**.

Table 1-57 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.

7. Click **OK**.

1.6.4.2 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. 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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
5. On the displayed page, click **Create IP Address Group**.
6. Configure the parameters based on **Table 1-58**.

Table 1-58 IP address group parameters

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 IP addresses or 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

7. 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 1-58**.

Adding IP Addresses

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

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
5. On the **IP Address Groups** page, locate the target IP address group and click its name.
6. In the lower part of the displayed page, choose **IP Addresses** tab and click **Add IP Addresses**. On the **Add IP Addresses** page, add IP addresses.
7. Click **OK**.

Changing IP Addresses

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

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

5. On the **IP Address Groups** page, you can:
   a. Modify the basic information and change IP addresses of an IP address group:
      i. Locate the target address group, click **Modify** in the **Operation** column. 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 Address**, 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. 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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
5. On the **IP Address Groups** page, locate the target IP address group and click its name.
6. In the IP address list, locate the IP address you want to delete and click **Delete** in the **Operation** column.
7. 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. 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** > **Elastic Load Balance**.

4. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
5. On the **IP Address Groups** page, locate the target IP address group and click its name.
6. 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. 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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
5. On the **IP Address Groups** page, locate the IP address group and click **Delete** in the **Operation** column.
6. Click **OK**.

1.6.5 SNI Certificate

Scenarios

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

SNI, an extension to Transport Layer Security (TLS), enables a server to present multiple certificates on the same IP address and port number. After you enable SNI, the client can submit the requested domain name at the start of the SSL handshake. After receiving the request, the load balancer searches for the certificate based on the domain name. If the certificate is found, the load balancer will return it to the client. If the certificate is not found, the load balancer will return the default certificate.

Constraints

- SNI can be only enabled for HTTPS listeners.
- 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

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

Prerequisites

- You have created a load balancer by referring to **Creating a Dedicated Load Balancer**.
- You have created an SNI certificate by referring to **Adding a Certificate**.
- You have added an HTTPS listener to the load balancer by referring to **Adding an HTTPS Listener**.

Restrictions

- You must specify at least one domain name for each certificate. The domain name must be the same as that in the certificate.
- A domain name can be used by both an ECC certificate and an RSA certificate. If there are two SNI certificates that use the same domain name, the ECC certificate is displayed preferentially.
- Domain names in an SNI certificate are matched as follows:

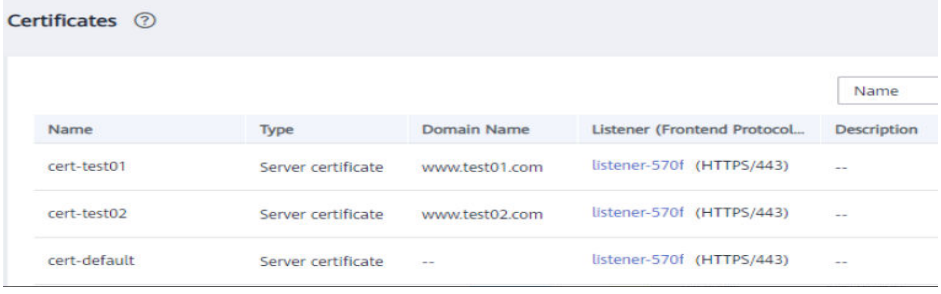
  If the domain name of the certificate is *.test.com, a.test.com and b.test.com are supported, but a.b.test.com and c.d.test.com are not supported.

  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.
- As shown in **Figure 1-25**, **cer-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 certificate will be used for authentication.

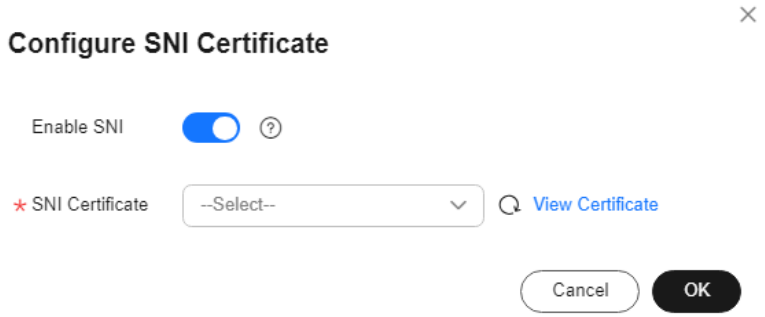
Figure 1-25 Configuring certificates



Enabling SNI for an HTTPS Listener

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner to display **Service List** and choose **Networking** > **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. On the **Summary** tab, click **Configure** on the right of SNI.
7. Enable SNI and select an SNI certificate.

   Figure 1-26 Configuring an SNI certificate



8. Click **OK**.

1.6.6 Certificate

1.6.6.1 Certificate Overview

ELB supports two types of certificates. If 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.

- **Server certificate**: used for SSL handshake negotiations if an HTTPS listener is used. Both the certificate content and private key are required.
- **CA certificate**: issued by a certificate authority (CA) and used to verify the certificate issuer. If HTTPS mutual authentication is required, HTTPS connections can be established only when the client provides a certificate issued by a specific CA.

NOTE

SSL Certificate Manager (SCM) allows you to purchase a certificate from Huawei Cloud or upload your own certificates for easier management.

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 supports certificates only in PEM format. If you have a certificate in any other format, you must convert it to a PEM-encoded certificate.
- If a certificate has expired, you need to manually replace or delete it.

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

The following is an example:

```
-----BEGIN CERTIFICATE-----
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 must contain 64 characters except the last row.

The following is an example:

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

1.6.6.2 Converting Certificate Formats

Scenarios

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

From DER to PEM

The DER format is usually used on a Java platform.

Run the following command to convert the certificate format:

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

Run the following command to convert the private key format:

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

From P7B to PEM

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

Run the following command to convert the certificate format:

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

From PFX to PEM

The PFX format is usually used by Windows Server.

Run the following command to convert the certificate format:

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

Run the following command to convert the private key format:

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

1.6.6.3 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: You can purchase a certificate from SSL Certificate Manager (SCM) or upload your own certificates.
- CA certificate: You can only upload your own CA certificates.
- Server SM certificates: You can purchase a certificate from SSL Certificate Manager (SCM) or upload your own 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 **Networking** > **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 1-59**.

Table 1-59 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. • **CA certificate**: issued by a certificate authority (CA) and used to verify the certificate issuer. If HTTPS mutual authentication is required, HTTPS connections can be established only when the client provides a certificate issued by a specific CA.
Certificate Name	Specifies the name of your certificate. This parameter is only available for your certificates.
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-----
Domain Name	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 **Networking** > **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 1-60**.

Table 1-60 CA certificate parameters

Parameter	Description
Certificate Type	Specifies the certificate type. Select **CA certificate**. • **Server certificate**: used for SSL handshake negotiations if an HTTPS listener is used. Both the certificate content and private key are required. • **CA certificate**: issued by a certificate authority (CA) and used to verify the certificate issuer. If HTTPS mutual authentication is required, HTTPS connections can be established only when the client provides a certificate issued by a specific CA.
Certificate Name	Specifies the name of the CA certificate.
Enterprise Project	Specifies an enterprise project by which cloud resources and members are centrally managed.
Certificate Content	Specifies the content of the CA certificate. The certificate must be a PEM file. 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**.

1.6.6.4 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. 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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Certificates**.
5. In the certificate list, click the listener name in the **Listener (Frontend Protocol/Port)** column to view its details.

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

Modifying a Certificate

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

Deleting a Certificate

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Certificates**.
5. Locate the certificate and click **Delete** in the **Operation** column.
6. In the displayed dialog box, click **OK**.

1.6.6.5 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. 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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Click the **Listeners** tab, locate the listener, and click **Edit** in **Operation** column.
6. On the displayed dialog box, select a server certificate or CA certificate.
7. Click **OK** in the **Edit** dialog box.

1.6.6.6 Replacing the Certificate Bound to Different Listeners

Scenario

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

NOTE

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

Constraints

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

Modifying a Certificate

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

1.6.7 Protection for Mission-Critical Operations

Scenarios

ELB supports sensitive operation protection. When you perform sensitive operations on the management console, you need to enter a credential that can prove your identity. You can perform corresponding operations only after your identity is authenticated. It is recommended that you enable operation protection to secure your account.

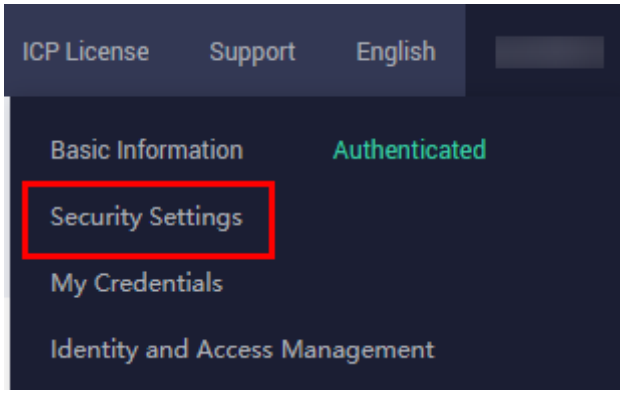
This function can be configured only by the administrator and takes effect for the resources in your account and the resources of users under your account. Common users have only the view permissions. To modify the permissions, contact the administrator.

Enabling Operation Protection

Operation protection is disabled by default. Perform the following operations to enable it:

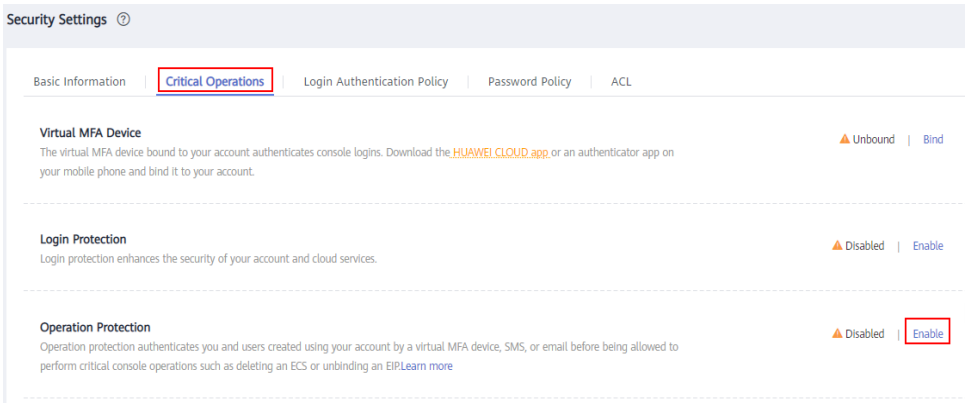
1. Log in to the management console.
2. Move the cursor to the username in the upper right corner of the page and select **Security Settings** from the list.

Figure 1-27 Security settings



3. On the **Security Settings** page, choose **Critical Operations** > **Operation Protection** > **Enable**.

   Figure 1-28 Critical operations



4. On the **Operation Protection** page, select **Enable**.

   If operation protection is enabled, you and IAM users created using your account need to enter a verification code when performing a critical operation, such as deleting an ECS resource.

   NOTE

   - When performing a critical operation, you will be asked to choose a verification method from email, SMS, and virtual MFA device.
     - If you have bound only a mobile number, only SMS verification is available.
     - If you have bound only an email address, only email verification is available.
     - If you have not bound an email address, mobile number, or virtual MFA device, bind one to perform critical operations.
   - You can change the mobile number, email address, and virtual MFA device on the **Basic Information** page.

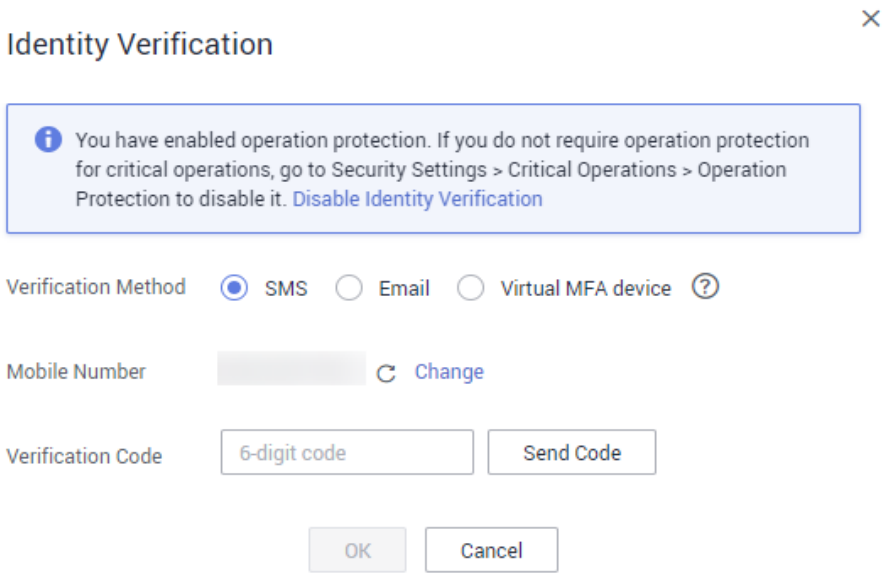
Verifying Operation Protection

After operation protection is enabled, when you perform a mission-critical operation, the system will verify your identity.

- If you have bound an email address, enter the email verification code.
- If you have bound a mobile number, enter the SMS verification code.
- If you have bound a virtual MFA device, enter a 6-digit dynamic verification code of the MFA device.

When you attempt to delete a load balancer, the following dialog box is displayed, and you need to select a verification method:

Figure 1-29 Identity verification

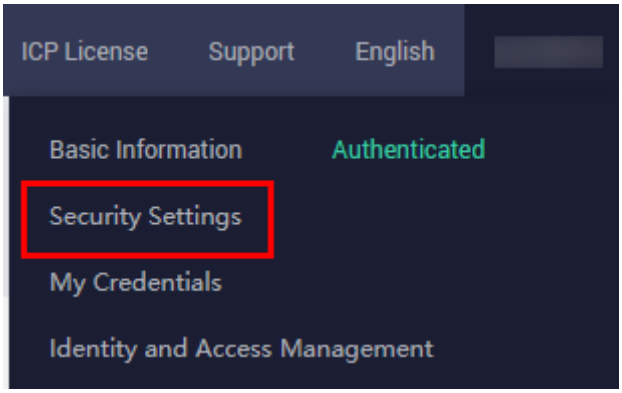


Disabling Operation Protection

Perform the following operations to disable operation protection:

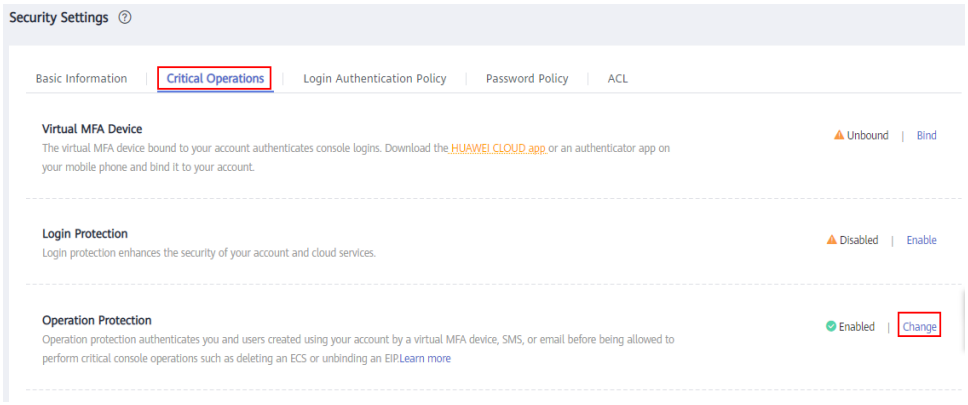
1. Log in to the management console.
2. Move the cursor to the username in the upper right corner of the page and select **Security Settings** from the list.

   Figure 1-30 Security settings

3. On the **Security Settings** page, choose **Critical Operations** > **Operation Protection** > **Change**.

Figure 1-31 Modifying operation protection settings



4. On the **Operation Protection** page, select **Disable** and click **OK**.

References

- **How Do I Bind a Virtual MFA Device?**
- **How Do I Obtain an MFA Verification Code?**

1.7 Access Logging

Scenarios

ELB logs HTTP and HTTPS requests received by load balancers, including the time when the request was sent, client IP address, request path, and server response.

With Log Tank Service (LTS), you can view logs of requests to load balancers at Layer 7 and analyze response status codes to quickly locate unhealthy backend servers.

NOTE

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

Notes and Constraints

- Access logs can be configured only for application (Layer 7) load balancers.
- 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. For details, see **Creating a Dedicated Load Balancer**.
- You have enabled LTS. For details, see **Accessing 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 add an HTTP or HTTPS listener to the load balancer. For details, see **Adding an HTTP Listener** or **Adding an HTTPS Listener**.

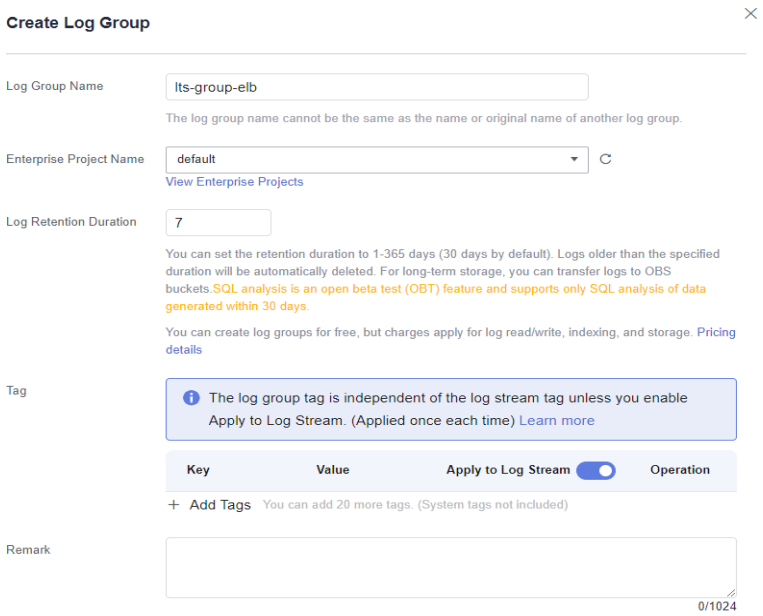
Flowchart

Figure 1-32 Process for locating an unhealthy backend server

Creating a Log Group

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. In the upper left corner of the page, click and select **Log Tank Service** under **Management & Governance**.
4. In the navigation pane on the left, choose **Log Management**.
5. On the lower part of the displayed page, click **Create Log Group**. In the displayed dialog box, enter a name for the log group.

   Figure 1-33 Creating a log group

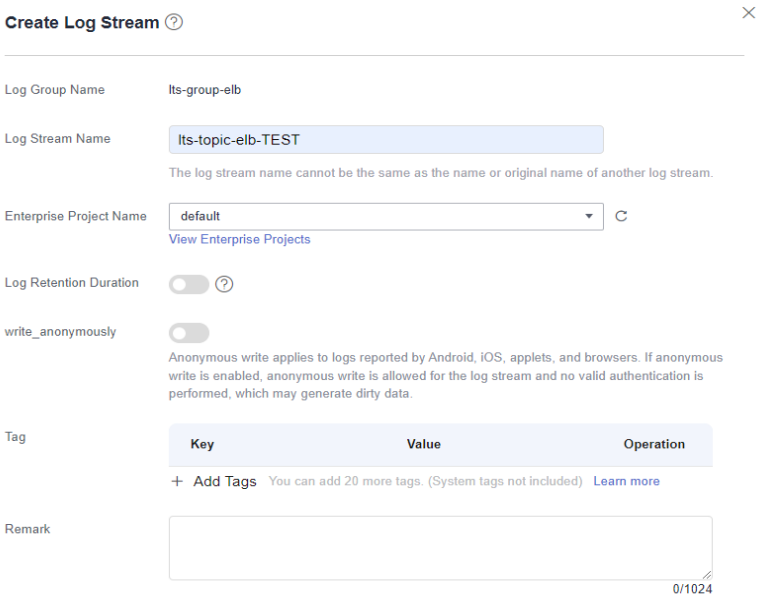


6. Confirm the settings and click **OK**.

Creating a Log Stream

1. On the LTS console, click on the left of the target log group.
2. Click **Create Log Stream**. In the displayed dialog box, enter a name for the log stream.

Figure 1-34 Creating a log stream

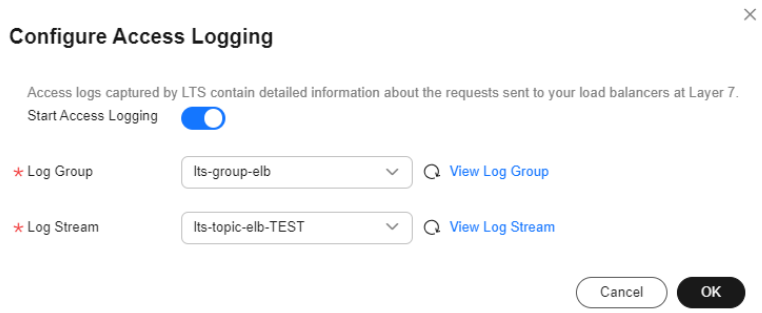


3. Confirm the settings and click **OK**.

Configuring Access Logging

1. Click ☰ in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
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.

Figure 1-35 Configuring access logging



5. Click **OK**.

NOTICE

Ensure that the log group is in the same region as the load balancer.

Viewing Access Logs

There are two ways for you to view access logs.

- On the ELB console, click the name of the target load balancer and click **Access Logs** to view logs.
- (Recommended) On the 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" "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 -
```

Table 1-61 describes the fields in the log.

Table 1-61 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

Parameter	Description	Value Description	Example Value
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
body_bytes_sent	Number of bytes sent to the client (excluding the response header).	Integer	3

Parameter	Description	Value Description	Example Value
request_time	Request processing time in seconds from the time when the load balancer receives the first request packet from the client to the time when the load balancer sends the response packet.	Floating-point data	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"

Parameter	Description	Value Description	Example Value
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"
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"

Parameter	Description	Value Description	Example Value
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"
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	"-"

Parameter	Description	Value Description	Example Value
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	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 the value can be **member_id**) or **-**.	String	"-"
tenant_id	Tenant ID.	String	f2bc165ad9b4483a9b17762da851bbbb

Parameter	Description	Value Description	Example Value
eip_address:eip_port	EIP of the load balancer and frontend port that were set when the listener was added.	EIP of the load balancer and frontend port that were set when the listener was added.	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	-
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

Parameter	Description	Value Description	Example Value
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	-

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.

Locating an Unhealthy Backend Server

The following is a log that records an exception:

```
1554944564.344 - [2019-04-11T09:02:44+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, 2019, 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 result

The backend server was unhealthy and failed to respond to the request.

NOTE

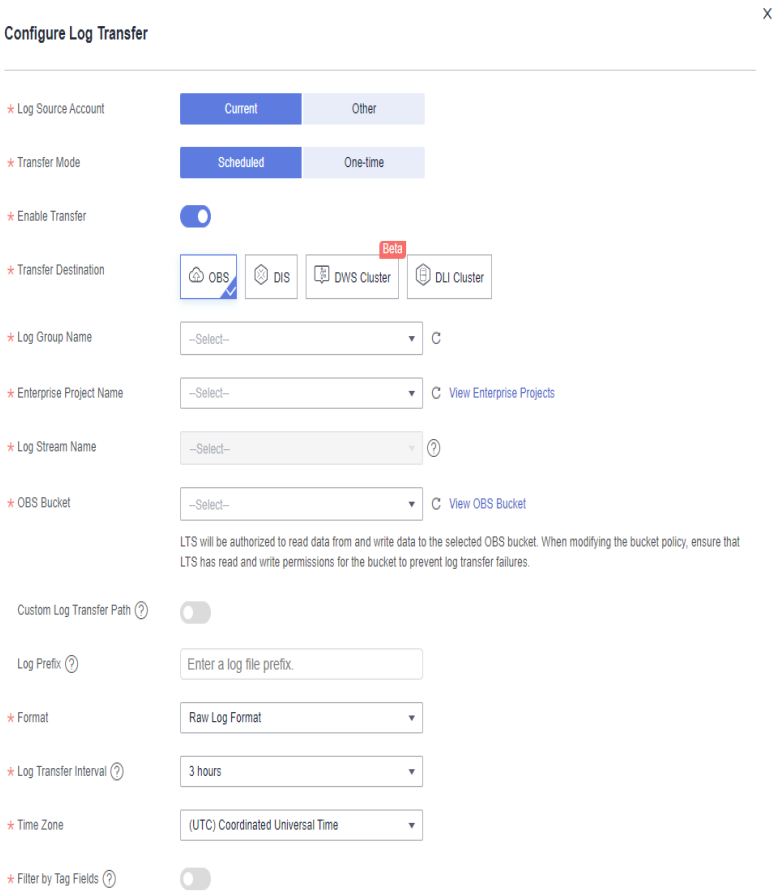
172.17.0.82:3000 is the private IP address of the backend server.

Configuring Log Transfer

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

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

Figure 1-36 Configuring log transfer



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

1.8 Tags and Quotas

1.8.1 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.

  For details about the procedure and parameters, see **Creating a Dedicated 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 **Networking** > **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**.

  NOTE

  - A maximum of 20 tags can be added to a load balancer.
  - Each tag is a key-value pair, and the tag key is unique.

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 **Networking** > **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**.

   NOTE

   - A maximum of 20 tags can be added to a listener.
   - Each tag is a key-value pair, and the tag key is unique.

Modifying a Tag

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** > **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 modified.
6. 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

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** > **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.

1.8.2 Quotas

What Is Quota?

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

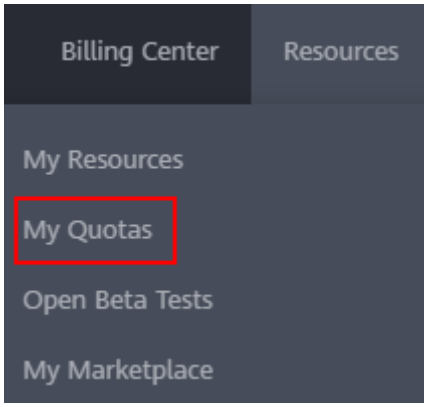
If the existing resource quota cannot meet your service requirements, you can apply for a higher quota.

How Do I View My Quotas?

1. Log in to the management console.
2. 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.

   Figure 1-37 My Quotas

4. View the used and total quota of each type of resources on the displayed page.

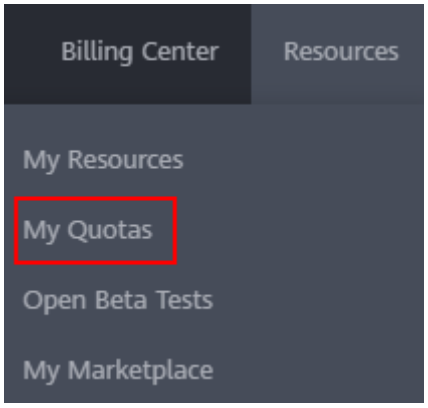
   If a quota cannot meet service requirements, apply for a higher quota.

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**.

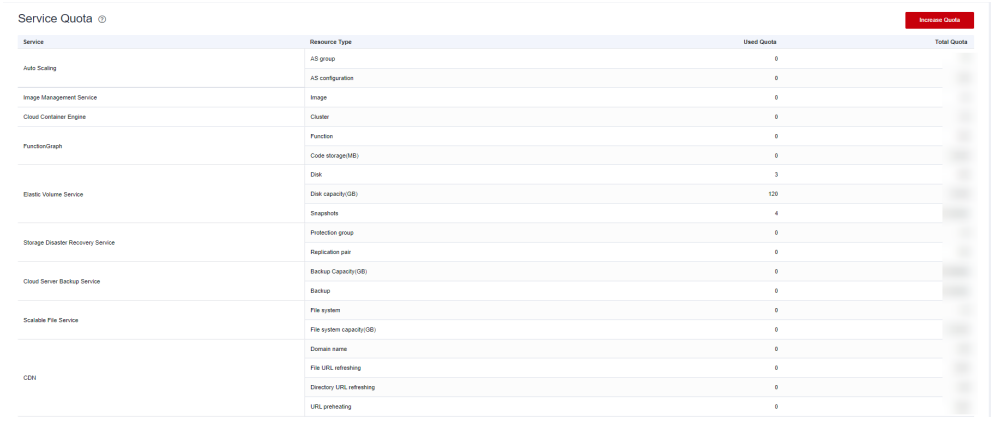
   The **Quotas** page is displayed.

   Figure 1-38 My quotas

3. Click **Increase Quota** in the upper right corner of the page.

Figure 1-39 Increasing quota



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**.

1.9 Monitoring

1.9.1 Monitoring Metrics

Overview

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

Namespace

SYS.ELB

Metrics

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 1-62 Metrics supported by each dedicated load balancer

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m1_cps	Concurrent Connections	Load balancing at Layer 4: total number of TCP and UDP connections from the monitored object to backend servers. Load balancing at Layer 7: total number of TCP connections from the clients to the monitored object. Unit: Count	≥0	Dedicated load balancer	1 minute
m2_act_conn	Active Connections	Number of TCP and UDP connections in the **ESTABLISHED** state between the monitored object and backend servers. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an` Unit: Count	≥0	Dedicated load balancer	1 minute
m3_inact_conn	Inactive Connections	Number of TCP connections between the monitored object and backend servers except those in the **ESTABLISHED** state. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an` Unit: Count	≥0	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m4_ncps	New Connections	Number of connections established between clients and the monitored object per second. Unit: packets/s	≥0/s	Dedicated load balancer	1 minute
m5_in_pps	Incoming Packets	Number of packets received by the monitored object per second. Unit: packets/s	≥0/s	Dedicated load balancer	1 minute
m6_out_pps	Outgoing Packets	Number of packets sent from the monitored object per second. Unit: packets/s	≥0/s	Dedicated load balancer	1 minute
m7_in_Bps	Inbound Rate	Traffic used for accessing the monitored object from the Internet. Unit: bytes/s	≥0 bytes/s	Dedicated load balancer	1 minute
m8_out_Bps	Outbound Rate	Traffic used by the monitored object to access the Internet per second. Unit: bytes/s	≥0 bytes/s	Dedicated load balancer	1 minute
m9_abnormal_servers	Unhealthy Servers	Number of unhealthy backend servers associated with the monitored object. Unit: Count	≥0	Dedicated load balancer	1 minute
ma_normal_servers	Healthy Servers	Number of healthy backend servers associated with the monitored object. Unit: Count	≥0	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m22_in_bandwidth	Inbound Bandwidth	Bandwidth used for accessing the monitored object from the Internet. Unit: bits/s	≥0 bits/s	Dedicated load balancer	1 minute
m23_out_bandwidth	Outbound Bandwidth	Bandwidth used by the monitored object to access the Internet. Unit: bits/s	≥0 bits/s	Dedicated load balancer	1 minute
m26_in_bandwidth_ipv6	IPv6 Inbound Bandwidth	IPv6 network bandwidth used for accessing the monitored object from the Internet. Unit: bits/s	≥0 bits/s	Dedicated load balancer	1 minute
m27_out_bandwidth_ipv6	IPv6 Outbound Bandwidth	IPv6 network bandwidth used by the monitored object to access the Internet. Unit: bits/s	≥0 bits/s	Dedicated load balancer	1 minute
m1e_server_rps	Reset Packets from Backend Servers	Number of reset packets sent from backend servers to clients. These reset packets are generated by the backend servers and then forwarded by the load balancer. This metric is available only for TCP listeners. Unit: Count/s	≥0/s	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m21_client_rps	Reset Packets from Clients	Number of reset packets sent by clients to backend servers. These reset packets are generated by clients and then forwarded by the load balancer. This metric is available only for TCP listeners. Unit: Count/s	≥0/s	Dedicated load balancer	1 minute
m1f_lvs_rps	Reset Packets from Load Balancers	Number of reset packets generated by the load balancer. This metric is available only for TCP listeners. Unit: Count/s	≥0/s	Dedicated load balancer	1 minute
mb_l7_qps	Layer-7 Query Rate	Number of requests the monitored object receives per second. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer	1 minute
mc_l7_http_2xx	Layer-7 2xx Status Codes	Number of 2xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
md_l7_http_3xx	Layer-7 3xx Status Codes	Number of 3xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer	1 minute
me_l7_http_4xx	Layer-7 4xx Status Codes	Number of 4xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer	1 minute
mf_l7_http_5xx	Layer-7 5xx Status Codes	Number of 5xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer	1 minute
m10_l7_http_other_status	Layer-7 Other Status Codes	Number of status codes returned by the load balancer and backend servers except 2xx, 3xx, 4xx, and 5xx status codes. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m11_l7_http_404	Layer-7 404 Not Found	Number of 404 Not Found status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer	1 minute
m12_l7_http_499	Layer-7 499 Client Closed Request	Number of 499 Client Closed Request status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer	1 minute
m13_l7_http_502	Layer-7 502 Bad Gateway	Number of 502 Bad Gateway status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m14_l7_rt	Average Layer-7 Response Time	Average response time of the monitored object. This metric is available only when the frontend protocol is HTTP or HTTPS. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Unit: ms **NOTE** The average response time it takes to establish a WebSocket connection may be very high. This metric cannot be used as a reference.	≥0 ms	Dedicated load balancer	1 minute
m15_l7_upstream_4xx	4xx Status Codes Backend	Number of 4xx status codes returned by the backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer	1 minute
m16_l7_upstream_5xx	5xx Status Codes Backend	Number of 5xx status codes returned by the backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m17_l7_upstream_rt	Average Server Response Time	Average response time of backend servers. 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. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: ms **NOTE** The average response time it takes to establish a WebSocket connection may be very high. This metric cannot be used as a reference.	≥0 ms	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m1a_l7_upstream_rt_max	Maximum Server Response Time	Maximum response time of backend servers. 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. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: ms	≥0 ms	Dedicated load balancer	1 minute
m1b_l7_upstream_rt_min	Minimum Server Response Time	Minimum response time of backend servers. 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. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: ms	≥0 ms	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m1c_l7_rt_max	Maximum Layer-7 Response Time	Maximum response time of the monitored object. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: ms	≥0 ms	Dedicated load balancer	1 minute
m1d_l7_rt_min	Minimum Layer-7 Response Time	Minimum response time of the monitored object. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: ms	≥0 ms	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m25_l7_resp_Bps	Backend Server Response Bandwidth	The bandwidth that the monitored object uses to return response to clients. Unit: bits/s **NOTE** When HTTP/2 is enabled for a listener, this metric cannot be used as a reference.	≥ 0bit/s	Dedicated load balancer	1 minute
m24_l7_req_Bps	Backend Server Request Bandwidth	The bandwidth that the monitored object uses to receive requests from clients. Unit: bits/s **NOTE** When HTTP/2 is enabled for a listener, this metric cannot be used as a reference.	≥ 0bit/s	Dedicated load balancer	1 minute
l7_con_usage	Layer-7 Concurrent Connection Usage	Ratio of HTTP and HTTPS connections established between the monitored object and backend servers per second, to the maximum number of concurrent connections allowed per second. Unit: percentage (%)	≥0%	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
l7_in_bps_usage	Layer-7 Inbound Bandwidth Usage	Ratio of the bandwidth that the monitored object uses to return response to clients over HTTP and HTTPS, to the maximum outbound bandwidth allowed. Unit: percentage (%) **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%	Dedicated load balancer	1 minute
l7_out_bps_usage	Layer-7 Outbound Bandwidth Usage	Ratio of the bandwidth that the monitored object uses to return response to clients over HTTP and HTTPS, to the maximum outbound bandwidth allowed. Unit: percentage (%) **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%	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
l7_ncps_usage	Layer-7 New Connection Usage	Ratio of HTTP and HTTPS connections established between clients and the monitored object per second, to the maximum number of new connections allowed per second. Unit: percentage (%)	≥0%	Dedicated load balancer	1 minute
l7_qps_usage	Layer 7 QPS Usage	Ratio of HTTP and HTTPS queries per second on the monitored object, to the maximum number of queries allowed per second. Unit: percentage (%)	≥0%	Dedicated load balancer	1 minute
l4_con_usage	Layer-4 Concurrent Connection Usage	Ratio of TCP and UDP connections established between the monitored object and backend servers per second, to the maximum number of concurrent connections allowed per second. Unit: percentage (%)	≥0%	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
l4_in_bps_usage	Layer-4 Inbound Bandwidth Usage	Ratio of the bandwidth that the monitored object uses to receive requests from clients over TCP and UDP, to the maximum inbound bandwidth allowed. Unit: percentage (%) **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%	Dedicated load balancer	1 minute
l4_out_bps_usage	Layer-4 Outbound Bandwidth Usage	Ratio of the bandwidth that the monitored object uses to return response to clients over TCP and UDP, to the maximum outbound bandwidth allowed. Unit: percentage (%) **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%	Dedicated load balancer	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
l4_ncps_usage	Layer-4 New Connection Usage	Ratio of TCP and UDP connections established between clients and the monitored object per second, to the maximum number of new connections allowed per second. Unit: percentage (%)	≥0%	Dedicated load balancer	1

Table 1-63 Metrics supported by each listener

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m1_cps	Concurrent Connections	Load balancing at Layer 4: total number of TCP and UDP connections from the monitored object to backend servers. Load balancing at Layer 7: total number of TCP connections from the clients to the monitored object. Unit: Count	≥0	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m2_act_conn	Active Connections	Number of TCP and UDP connections in the **ESTABLISHED** state between the monitored object and backend servers. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an` Unit: Count	≥0	Dedicated load balancer - listener	1 minute
m3_inact_conn	Inactive Connections	Number of TCP connections between the monitored object and backend servers except those in the **ESTABLISHED** state. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an` Unit: Count	≥0	Dedicated load balancer - listener	1 minute
m4_ncps	New Connections	Number of connections established between clients and the monitored object per second. Unit: packets/s	≥0/s	Dedicated load balancer - listener	1 minute
m5_in_pps	Incoming Packets	Number of packets received by the monitored object per second. Unit: Count/s	≥0/s	Dedicated load balancer - listener	1

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m6_out_pps	Outgoing Packets	Number of packets sent from the monitored object per second. Unit: Count/s	≥0/s	Dedicated load balancer - listener	1 minute
m7_in_Bps	Inbound Rate	Traffic used for accessing the monitored object from the Internet. Unit: bytes/s	≥0 bytes/s	Dedicated load balancer - listener	1 minute
m8_out_Bps	Outbound Rate	Traffic used by the monitored object to access the Internet per second. Unit: bytes/s	≥0 bytes/s	Dedicated load balancer - listener	1 minute
m9_abnormal_servers	Unhealthy Servers	Number of unhealthy backend servers associated with the monitored object. Unit: Count	≥0	Dedicated load balancer - listener	1 minute
ma_normal_servers	Healthy Servers	Number of healthy backend servers associated with the monitored object. Unit: Count	≥0	Dedicated load balancer - listener	1 minute
m22_in_bandwidth	Inbound Bandwidth	Bandwidth used for accessing the monitored object from the Internet. Unit: bits/s	≥ 0bit/s	Dedicated load balancer - listener	1 minute
m23_out_bandwidth	Outbound Bandwidth	Bandwidth used by the monitored object to access the Internet. Unit: bits/s	≥ 0bit/s	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m1e_server_rps	Reset Packets from Backend Servers	Number of reset packets sent from backend servers to clients. These reset packets are generated by the backend servers and then forwarded by the load balancer. This metric is available only for TCP listeners. Unit: Count/s	≥0/s	Dedicated load balancer - listener	1 minute
m21_client_rps	Reset Packets from Clients	Number of reset packets sent by clients to backend servers. These reset packets are generated by clients and then forwarded by the load balancer. This metric is available only for TCP listeners. Unit: Count/s	≥0/s	Dedicated load balancer - listener	1 minute
m1f_lvs_rps	Reset Packets from Load Balancers	Number of reset packets generated by the load balancer. This metric is available only for TCP listeners. Unit: Count/s	≥0/s	Dedicated load balancer - listener	1 minute
mb_l7_qps	Layer-7 Query Rate	Number of requests the monitored object receives per second. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
mc_l7_http_2xx	Layer-7 2xx Status Codes	Number of 2xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer - listener	1 minute
md_l7_http_3xx	Layer-7 3xx Status Codes	Number of 3xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer - listener	1 minute
me_l7_http_4xx	Layer-7 4xx Status Codes	Number of 4xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer - listener	1 minute
mf_l7_http_5xx	Layer-7 5xx Status Codes	Number of 5xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m10_l7_http_other_status	Layer-7 Other Status Codes	Number of status codes returned by the monitored object except 2xx, 3xx, 4xx, and 5xx status codes. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer - listener	1 minute
m11_l7_http_404	Layer-7 404 Not Found	Number of 404 Not Found status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer - listener	1 minute
m12_l7_http_499	Layer-7 499 Client Closed Request	Number of 499 Client Closed Request status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m13_l7_http_502	Layer-7 502 Bad Gateway	Number of 502 Bad Gateway status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer - listener	1 minute
m14_l7_rt	Average Layer-7 Response Time	Average response time of the monitored object. This metric is available only when the frontend protocol is HTTP or HTTPS. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Unit: ms **NOTE** The average response time it takes to establish a WebSocket connection may be very high. This metric cannot be used as a reference.	≥0 ms	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m15_l7_upstream_4xx	4xx Status Codes Backend	Number of 4xx status codes returned by the backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer - listener	1 minute
m16_l7_upstream_5xx	5xx Status Codes Backend	Number of 5xx status codes returned by the monitored object. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer - listener	1 minute
m17_l7_upstream_rt	Average Server Response Time	Average response time of backend servers. This metric is available only when the frontend protocol is HTTP or 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. Unit: ms **NOTE** The average response time it takes to establish a WebSocket connection may be very high. This metric cannot be used as a reference.	≥0 ms	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m1a_l7_upstream_rt_max	Maximum Server Response Time	Maximum response time of backend servers. This metric is available only when the frontend protocol is HTTP or 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. Unit: ms	≥0 ms	Dedicated load balancer - listener	1
m1b_l7_upstream_rt_min	Minimum Server Response Time	Minimum response time of backend servers. This metric is available only when the frontend protocol is HTTP or 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. Unit: ms	≥0 ms	Dedicated load balancer - listener	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m1c_l7_rt_max	Maximum Layer-7 Response Time	Maximum response time of the monitored object. This metric is available only when the frontend protocol is HTTP or HTTPS. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Unit: ms	≥0 ms	Dedicated load balancer - listener	1 minute
m1d_l7_rt_min	Minimum Layer-7 Response Time	Minimum response time of the monitored object. This metric is available only when the frontend protocol is HTTP or HTTPS. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Unit: ms	≥0 ms	Dedicated load balancer - listener	1 minute

Table 1-64 Metrics supported by each backend server group

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m9_abnormal_servers	Unhealthy Servers	Number of unhealthy backend servers associated with the monitored object. Unit: Count	≥0	Dedicated load balancer - backend server group	1 minute
ma_normal_servers	Healthy Servers	Number of healthy backend servers associated with the monitored object. Unit: Count	≥0	Dedicated load balancer - backend server group	1 minute
mb_l7_qps	Layer-7 Query Rate	Number of requests the monitored object receives per second. Unit: Count/s	≥0/s	Dedicated load balancer - backend server group	1 minute
m17_l7_upstream_rt	Average Server Response Time	Average response time of backend servers. This metric is available only when the frontend protocol is HTTP or 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. Unit: ms **NOTE** The average response time it takes to establish a WebSocket connection may be very high. This metric cannot be used as a reference.	≥ 0ms	Dedicated load balancer - backend server group	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m1a_l7_upstream_rt_max	Maximum Server Response Time	Maximum response time of backend servers. This metric is available only when the frontend protocol is HTTP or 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. Unit: ms	≥ 0ms	Dedicated load balancer - backend server group	1 minute
m1b_l7_upstream_rt_min	Minimum Server Response Time	Minimum response time of backend servers. This metric is available only when the frontend protocol is HTTP or 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. Unit: ms	≥ 0ms	Dedicated load balancer - backend server group	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m18_l7_upstream_2xx	2xx Status Codes Backend	Number of 2xx status codes returned by the monitored object. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer - backend server group	1 minute
m19_l7_upstream_3xx	3xx Status Codes Backend	Number of 3xx status codes returned by the monitored object. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer - backend server group	1 minute
m15_l7_upstream_4xx	4xx Status Codes Backend	Number of 4xx status codes returned by the monitored object. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer - backend server group	1 minute
m16_l7_upstream_5xx	5xx Status Codes Backend	Number of 5xx status codes returned by the monitored object. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Dedicated load balancer - backend server group	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m25_l7_resp_Bps	Backend Server Response Bandwidth	The bandwidth that the monitored object uses to return response to clients. Unit: bits/s **NOTE** When HTTP/2 is enabled for a listener, this metric cannot be used as a reference.	≥ 0bit/s	Dedicated load balancer - backend server group	1 minute
m24_l7_req_Bps	Backend Server Request Bandwidth	The bandwidth that the monitored object uses to receive requests from clients. Unit: bits/s **NOTE** When HTTP/2 is enabled for a listener, this metric cannot be used as a reference.	≥ 0bit/s	Dedicated load balancer - backend server group	1 minute

Table 1-65 Metrics supported by AZ

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m1_cps	Concurrent Connections	Load balancing at Layer 4: total number of TCP and UDP connections from the monitored object to backend servers. Load balancing at Layer 7: total number of TCP connections from the clients to the monitored object. Unit: Count	≥0	AZ	1 minute
m2_act_conn	Active Connections	Number of TCP and UDP connections in the **ESTABLISHED** state between the monitored object and backend servers. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an` Unit: Count	≥0	AZ	1 minute
m3_inact_conn	Inactive Connections	Number of TCP connections between the monitored object and backend servers except those in the **ESTABLISHED** state. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an` Unit: Count	≥0	AZ	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m4_ncps	New Connections	Number of connections established between clients and the monitored object per second. Unit: packets/s	≥0/s	AZ	1 minute
m5_in_pps	Incoming Packets	Number of packets received by the monitored object per second. Unit: Count/s	≥0/s	AZ	1 minute
m6_out_pps	Outgoing Packets	Number of packets sent from the monitored object per second. Unit: Count/s	≥0/s	AZ	1 minute
m7_in_Bps	Inbound Rate	Traffic used for accessing the monitored object from the Internet. Unit: bytes/s	≥ 0bytes/s	AZ	1 minute
m8_out_Bps	Outbound Rate	Traffic used by the monitored object to access the Internet per second. Unit: bytes/s	≥ 0bytes/s	AZ	1 minute
m26_in_bandwidth_ipv6	IPv6 Inbound Bandwidth	IPv6 network bandwidth used for accessing the monitored object from the Internet. Unit: bits/s	≥ 0bit/s	AZ	1 minute
m27_out_bandwidth_ipv6	IPv6 Outbound Bandwidth	IPv6 network bandwidth used by the monitored object to access the Internet. Unit: bits/s	≥ 0bit/s	AZ	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m1e_server_rps	Reset Packets from Backend Servers	Number of reset packets sent from backend servers to clients. These reset packets are generated by the backend servers and then forwarded by the load balancer. This metric is available only for TCP listeners. Unit: Count/s	≥0/s	AZ	1 minute
m21_client_rps	Reset Packets from Clients	Number of reset packets sent by clients to backend servers. These reset packets are generated by clients and then forwarded by the load balancer. This metric is available only for TCP listeners. Unit: Count/s	≥0/s	AZ	1 minute
m1f_lvs_rps	Reset Packets from Load Balancers	Number of reset packets generated by the load balancer. This metric is available only for TCP listeners. Unit: Count/s	≥0/s	AZ	1 minute
mb_l7_qps	Layer-7 Query Rate	Number of requests the monitored object receives per second. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	AZ	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
mc_l7_http_2xx	Layer-7 2xx Status Codes	Number of 2xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	AZ	1 minute
md_l7_http_3xx	Layer-7 3xx Status Codes	Number of 3xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	AZ	1 minute
me_l7_http_4xx	Layer-7 4xx Status Codes	Number of 4xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	AZ	1 minute
mf_l7_http_5xx	Layer-7 5xx Status Codes	Number of 5xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	AZ	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m10_l7_http_other_status	Layer-7 Other Status Codes	Number of status codes returned by the load balancer and backend servers except 2xx, 3xx, 4xx, and 5xx status codes. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	AZ	1 minute
m11_l7_http_404	Layer-7 404 Not Found	Number of 404 Not Found status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	AZ	1 minute
m12_l7_http_499	Layer-7 499 Client Closed Request	Number of 499 Client Closed Request status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	AZ	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m13_l7_http_502	Layer-7 502 Bad Gateway	Number of 502 Bad Gateway status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	AZ	1 minute
m14_l7_rt	Average Layer-7 Response Time	Average response time of the monitored object. This metric is available only when the frontend protocol is HTTP or HTTPS. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Unit: ms **NOTE** The average response time it takes to establish a WebSocket connection may be very high. This metric cannot be used as a reference.	≥ 0ms	AZ	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m15_l7_upstream_4xx	4xx Status Codes Backend	Number of 4xx status codes returned by the backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	AZ	1 minute
m16_l7_upstream_5xx	5xx Status Codes Backend	Number of 5xx status codes returned by the backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	AZ	1 minute
m17_l7_upstream_rt	Average Server Response Time	Average response time of backend servers. 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. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: ms **NOTE** The average response time it takes to establish a WebSocket connection may be very high. This metric cannot be used as a reference.	≥ 0ms	AZ	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m1a_l7_upstream_rt_max	Maximum Server Response Time	Maximum response time of backend servers. 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. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: ms	≥ 0ms	AZ	1 minute
m1b_l7_upstream_rt_min	Minimum Server Response Time	Minimum response time of backend servers. 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. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: ms	≥ 0ms	AZ	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m1c_l7_rt_max	Maximum Layer-7 Response Time	Maximum response time of the monitored object. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: ms	≥ 0ms	AZ	1 minute
m1d_l7_rt_min	Minimum Layer-7 Response Time	Minimum response time of the monitored object. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: ms	≥ 0ms	AZ	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
l4_con_usage	Layer-4 Concurrent Connection Usage	Ratio of TCP and UDP connections established between the monitored object and backend servers per second, to the maximum number of concurrent connections allowed per second. Unit: percentage (%)	≥ 0%	AZ	1 minute
l4_in_bps_usage	Layer-4 Inbound Bandwidth Usage	Ratio of the bandwidth that the monitored object uses to receive requests from clients over TCP and UDP, to the maximum inbound bandwidth allowed. Unit: percentage (%) **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%	AZ	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
l4_out_bps_usage	Layer-4 Outbound Bandwidth Usage	Ratio of the bandwidth that the monitored object uses to return response to clients over TCP and UDP, to the maximum outbound bandwidth allowed. Unit: percentage (%) **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%	AZ	1 minute
l4_ncps_usage	Layer-4 New Connection Usage	Ratio of TCP and UDP connections established between clients and the monitored object per second, to the maximum number of new connections allowed per second. Unit: percentage (%)	≥ 0%	AZ	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
l7_in_bps_usage	Layer-7 Inbound Bandwidth Usage	Ratio of the bandwidth that the monitored object uses to return response to clients over HTTP and HTTPS, to the maximum outbound bandwidth allowed. Unit: percentage (%) **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%	AZ	1 minute
l7_out_bps_usage	Layer-7 Outbound Bandwidth Usage	Ratio of the bandwidth that the monitored object uses to return response to clients over HTTP and HTTPS, to the maximum outbound bandwidth allowed. Unit: percentage (%) **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%	AZ	1 minute

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
l7_con_usage	Layer-7 Concurrent Connection Usage	Ratio of HTTP and HTTPS connections established between the monitored object and backend servers per second, to the maximum number of concurrent connections allowed per second. Unit: percentage (%)	≥ 0%	AZ	1 minute
l7_ncps_usage	Layer-7 New Connection Usage	Ratio of HTTP and HTTPS connections established between clients and the monitored object per second, to the maximum number of new connections allowed per second. Unit: percentage (%)	≥ 0%	AZ	1 minute
l7_qps_usage	Layer 7 QPS Usage	Ratio of HTTP and HTTPS queries per second on the monitored object, to the maximum number of queries allowed per second. Unit: percentage (%)	≥ 0%	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.

Key	Value
available_zone	AZ where a dedicated load balancer works.

1.9.2 Setting an Alarm Rule

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

1.9.2.1 Creating an Alarm Rule

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

   Configure parameters based on **Table 1-66**.

   Table 1-66 Parameters for creating an alarm rule

Parameter	Setting
Resource Type	Select **Elastic Load Balance**.
Dimension	Select from the following options: • **Elastic Load Balancers** • **Elastic Load Balancers - Listeners** • **Elastic Load Balancers - Backend Server Group**
Other Parameters	Set them as required.

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

   NOTE

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

1.9.2.2 Modifying an Alarm Rule

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner and choose **Management & Governance** > **Cloud Eye**.
4. In the navigation pane on the left, choose **Alarm Management** > **Alarm Rules**.
5. On the **Alarm Rules** page, locate the alarm rule and click **Modify** in the **Operation** column.
   a. On the **Modify Alarm Rule** page, modify the parameters.
   b. Set other parameters as required and then click **Modify**.

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

      NOTE

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

1.9.3 Viewing Metrics

Scenarios

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

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

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

Prerequisites

- The load balancer is running properly.

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

  NOTE

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

- You have interconnected ELB with Cloud Eye and configured an alarm rule for the load balancer on the Cloud Eye console.

  Without alarm rules, there is no monitoring data. For details, see **Setting an Alarm Rule**.

- If an IAM user wants to view the ELB monitoring data on the Cloud Eye console, the IAM user must be granted the **ELB Administrator** permission. Otherwise, the IAM user cannot view all monitoring data.

Viewing Monitoring Metrics on the 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 **Networking** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. View the metrics of each load balancer and listener.
   a. Load balancer: Click **Monitoring** tab and select **Load balancer** for **Dimension**.
   b. Listener (two ways):
      i. Click **Monitoring** tab, select **Load listener** for **Dimension**, locate the target listener, and view the monitoring metrics.
      ii. Click the name of the target listener, 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** > **Elastic Load Balance**.
5. On the **Cloud Service Monitoring** page, click the name of the load balancer. Alternatively, locate the load balancer and click **View Metric** in the **Operation** column.
6. Select the time period during which you want to view metrics. You can select a system-defined time period (for example, last 1 hour) or specify a time period.
7. Click **Select Metric** in the upper right corner and select the metrics to be viewed.

NOTE

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

1.9.4 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.

The associated backend servers are running normally and are not deleted or in the stopped or faulty state.

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 the service list and choose **Networking** > **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 1-40 EIP traffic usage

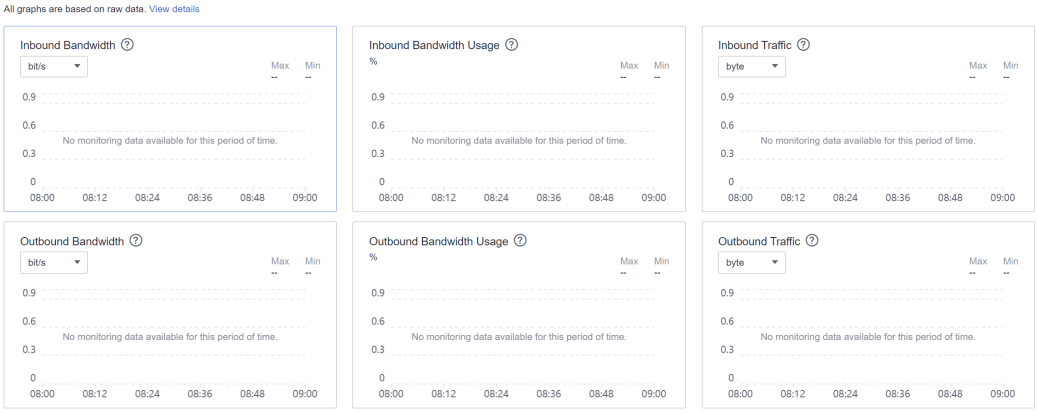


Table 1-67 EIP and bandwidth metrics

Metric	Meaning	Value Range	Monitored Object	Monitoring Period (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. 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 the service list and choose **Networking** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. 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. For details, see **ELB Metrics**.

1.10 Auditing

1.10.1 Key Operations Recorded by CTS

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

Table 1-68 lists the operations recorded by CTS.

Table 1-68 ELB operations recorded by CTS

Action	Resource Type	Trace
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

Action	Resource Type	Trace
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
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

1.10.2 Viewing Traces

Scenarios

CTS records the operations performed on ELB and allows you to view the operation records of the last seven days on the CTS console. To query these records, perform the following operations.

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**

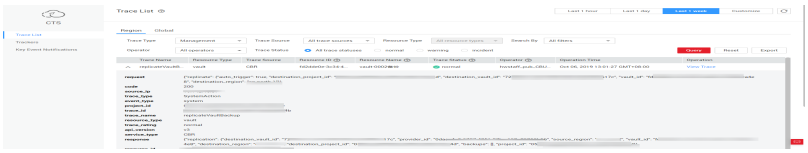
     Select a filter from the drop-down list.

     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 1-41 Expanding trace details



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

Figure 1-42 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": "
    "tracker_name": "system",
    "time": "1569321775225",
    "resource_name": "9646e73b-338c-4d27-a17c-219be532812c",
    "record_time": "1569321775903",
    "user": {
        "domain": {
            "name": "
            "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 Huawei Cloud account or cloud service to perform efficient O&M on your ELB resources.

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

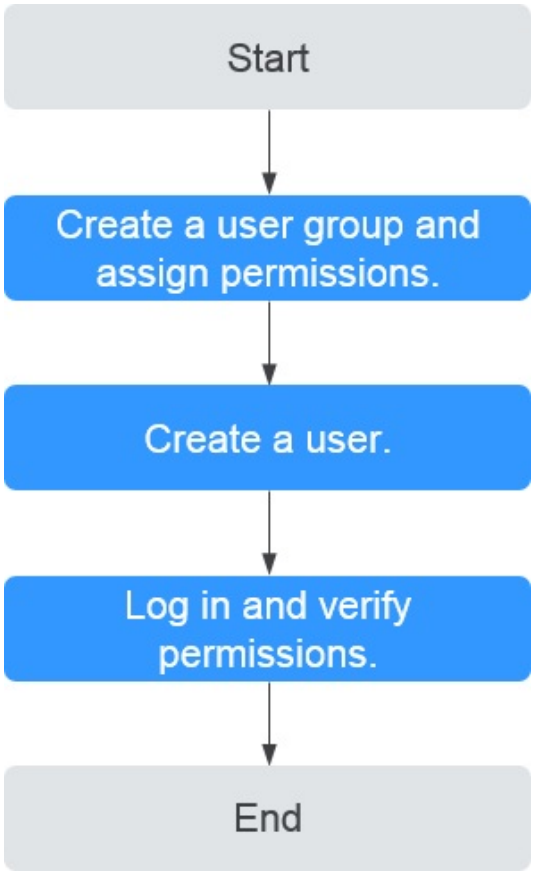
This following describes the procedure for granting permissions.

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



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 ReadOnlyAccessELB Viewer** 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: Edit JSON policies from scratch or based on an existing policy.

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"
            ]
```



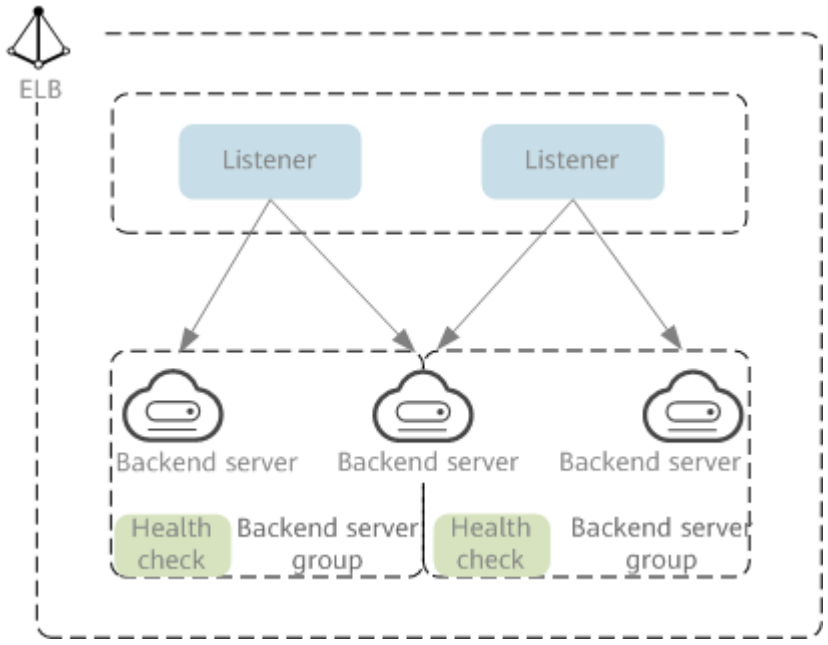
```
    }
  ]
}
```

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.

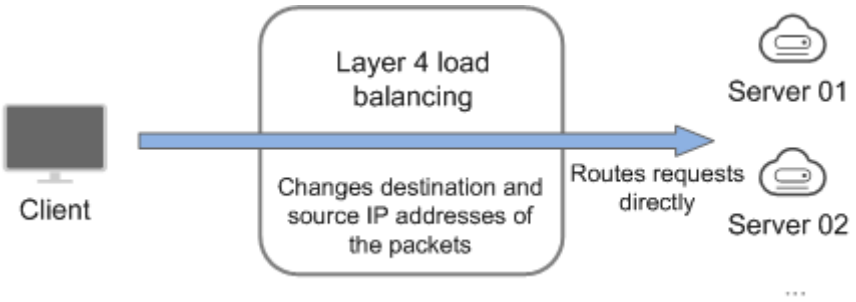
- To distribute requests over the Internet, you need to assign an EIP or bind an existing EIP to a load balancer so that it can route requests from the Internet to backend servers.
- If you want to distribute requests within a VPC, create a private network load balancer. This type of load balancers has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

Protocol

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

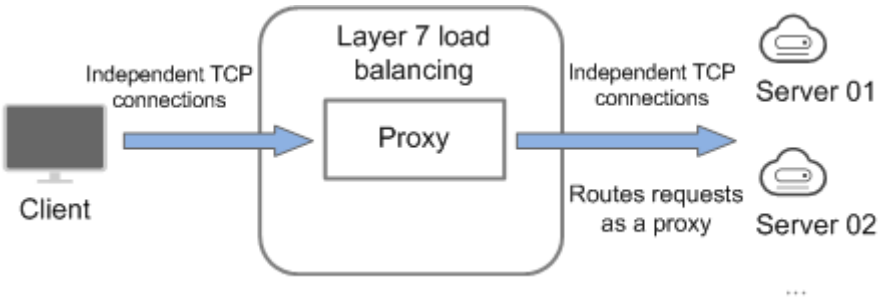
- If you choose TCP or UDP, the load balancer routes requests directly to backend servers. In this process, the destination IP address in a packet is changed to the IP address of the backend server, and the source IP address to the private IP address of the load balancer. A connection is established 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

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

Shared load balancers receive requests from clients and route them to backend servers, which answer to these requests over the private network.

Constraints

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

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click  in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, click **Create Elastic Load Balancer**. Configure the parameters based on **Table 2-1**.

Table 2-1 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. For details about the differences, see **Differences Between Dedicated and Shared Load Balancers**.
Billing Mode	Specifies the billing mode of the shared load balancer. You are charged for how long you use each load balancer.
Region	Specifies the desired region. Resources in different regions cannot communicate with each other over internal networks. For lower network latency and quicker resource access, select the nearest region.

Parameter	Description
Name	Specifies the name of the shared load balancer.
Enterprise Project	Specifies an enterprise project by which cloud resources and members are centrally managed.
Description	Provides supplementary information about the load balancer.
Tag	Identifies 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 2-2**. A maximum of 20 tags can be added.

Table 2-2 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. • Only letters, digits, underscores (_), hyphens (-), at signs (@), and Chinese characters are allowed.
Tag value	• Can contain a maximum of 43 characters. • Only letters, digits, underscores (_), hyphens (-), at signs (@), and Chinese characters are allowed.

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

Table 2-3 Parameters for network configurations

Parameter	Description
Network Type	Specifies the network type of a load balancer. • **Public IPv4 network**: The load balancer routes IPv4 requests from the clients to backend servers over the Internet. • **Private IPv4 network**: The load balancer routes IPv4 requests from the clients to backend servers in the same VPC as the load balancer.

Parameter	Description
VPC	Specifies the VPC where the shared load balancer will work. You cannot change the VPC after the load balancer is created. Plan the VPC as required. Select an existing VPC or create a new one. For more information about VPC, see the ***Virtual Private Cloud User Guide***.
Frontend Subnet	Specifies the subnet where the load balancer will work. Shared load balancers support private IPv4 network by default. The system assigns IPv4 private addresses in this subnet to load balancers.
IPv4 Address	Specifies how you want the IPv4 address to be assigned. • **Automatically assign IP address**: The system automatically assigns an IPv4 address to the load balancer. • **Manually specify IP address**: You need to manually specify an IPv4 address to the load balancer. **NOTE** Network ACL rules configured for the backend subnet of the load balancer will not restrict the traffic from the clients to the load balancer. If network ACL rules are configured, the clients can directly access the load balancer. To control access to the load balancer, configure access control for all listeners added to the load balancer. For details, see **What Is Access Control?**
Guaranteed Performance	Specifies whether to enable the guaranteed performance option. This function allows your load balancers to handle up to 50,000 concurrent connections, 5,000 new connections per second, and 5,000 queries per second. This function is enabled by default and cannot be disabled.
EIP	Specifies the public IP address that will be bound to the load balancer for receiving and forwarding requests over the Internet. You can use an existing EIP or assign a new one. The following options are available: • **New EIP**: The system will automatically assign an EIP. • **Use existing**: Select an existing EIP.

Parameter	Description
EIP Type	Specifies the link type (BGP) when a new EIP is used. • **Static BGP**: When changes occur on a network using static BGP, carriers cannot adjust network configurations in real time to ensure optimal user experience. • **Dynamic BGP**: When changes occur on a network using dynamic BGP, routing protocols provide automatic, real-time optimization of network configurations, ensuring network stability and optimal user experience.
Billed By	Specifies how the bandwidth will be billed. • **Bandwidth**: You specify the maximum bandwidth and pay for the amount of time you use the bandwidth. • **Traffic**: You specify a maximum bandwidth and pay for the outbound traffic you use. • **Shared Bandwidth**: Only the shared bandwidth will be billed. There will be no additional bandwidth or traffic costs for EIPs added to the shared bandwidth.
Bandwidth	Specifies the maximum bandwidth when a new EIP is used, in Mbit/s.

6. Click **Next**.
7. Confirm the configuration and submit your request.

Exporting the Load Balancer List

After a load balancer is created, you can export the information about all load balancers under your account to a local directory as an Excel file.

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** > **Elastic Load Balance**.
4. In the upper left corner of the load balancer list, click **Export**.

2.2.3 Configuring Modification Protection for Shared Load Balancers

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

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. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. On the **Summary** tab, click **Configure** next to **Modification Protection**.
6. In the **Configure Modification Protection** dialog box, enable or disable **Modification Protection**.

   Fill in the reason if needed.
7. Click **OK**.

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 and Unbinding an 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. 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** > **Elastic Load Balance**.
4. On the **Load Balancers** 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. 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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click **More** in the **Operation** column.
5. Click **Modify IPv4 Bandwidth**.
6. In the **New Configuration** area, modify the billing option and 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.
7. 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 Deleting a Shared Load Balancer

Scenarios

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

CAUTION

A deleted load balancer cannot be recovered.

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

Prerequisites

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

1. Delete all the forwarding policies added to HTTP and HTTPS listeners of the load balancer.

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

Deleting a Load Balancer

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

   A confirmation dialog box is displayed.
5. Click **OK**.

2.2.6 Enabling Guaranteed Performance for a Shared Load Balancer

Scenarios

Guaranteed performance allows shared load balancers to handle up to 50,000 concurrent connections, 5,000 new connections per second, and 5,000 queries per second. It provides you with more stable and reliable load balancing capabilities in case of traffic surge.

If your shared load balancers were created after February 10, 2023, guaranteed performance will be enabled for them by default.

If your shared load balancers were created before February 10, 2023, perform the following operations to enable guaranteed performance.

Notes

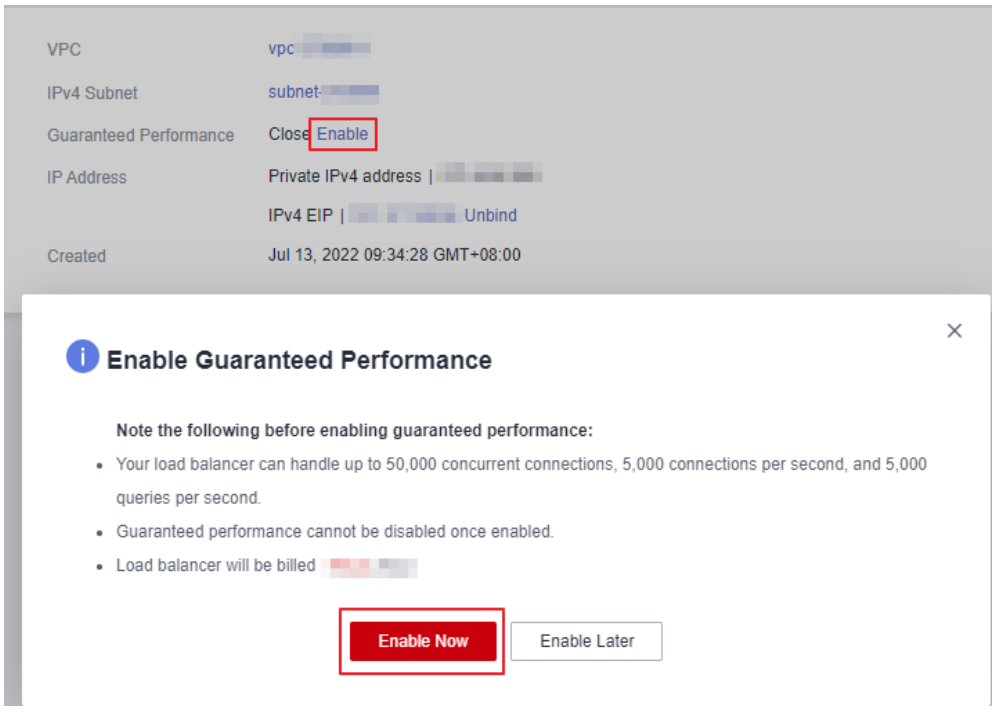
- Guaranteed performance cannot be disabled once enabled.
- After guaranteed performance is enabled, shared load balancers will be billed on a pay-per-use basis. For details about product prices, see **Product Pricing Details**.

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. Click ☰ in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. Locate the target shared load balancer and click its name to enter the **Summary** page.
5. Click **Enable**.
6. Click **Enable Now**.

Figure 2-5 Enabling guaranteed performance



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 2-4 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 quotations
Layer 7	HTTP	• Cookie-based sticky sessions • X-Forward-For request header	Web applications where data content needs to be identified, such as mobile games
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.

NOTE

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 2-5 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 port numbers: 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 2-6 Backend protocols and ports

Backend Protocol	TCP, UDP, and HTTP
Backend Port	Backend servers can use the same ports. The port number ranges from 1 to 65535. The following are some commonly-used protocols and port numbers: 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. 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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**. Configure the parameters based on **Table 2-7**.

Table 2-7 Parameters for configuring a TCP listener

Parameter	Description
Name	Specifies the listener name.
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients. Select **TCP**.
Frontend Port	Specifies the port that will be used by the load balancer to receive requests from clients. The port number ranges from 1 to 65535.
Access Control	Specifies how access to the listener is controlled. For details, see **What Is Access Control?** The following options are available: • All IP addresses • Blacklist • Whitelist
IP Address Group	Specifies the IP address group associated with a whitelist or blacklist. If there is no IP address group, create one first. For more information, see **IP Address Group**.
Transfer Client IP Address	Specifies whether to transmit IP addresses of the clients to backend servers.
Advanced Settings	
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.

6. Click **Next: Configure Request Routing Policy**.
   a. You are advised to select an existing backend server group.
   b. You can also click **Create new** to create a backend server group.
      i. Configure the backend server group based on **Table 2-27**.
      ii. Click **Next: Add Backend Server**. Add backend servers and configure health check for the backend server group.
      For details about how to add backend servers, see **Backend Server Overview**. For the parameters required for configuring a health check, see **Table 2-28**.
7. Click **Next: Confirm**.
8. 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 quotations.

Constraints

- UDP listeners do not support fragmentation.
- The port of UDP listeners cannot be 4789.
- UDP packets can have any size less than 1,500 bytes. The packets will be discarded if they are bigger than 1,500 bytes. To avoid this, you need to modify the configuration files of the applications based on the maximum transmission unit (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. 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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**. Configure the parameters based on **Table 2-8**.

Table 2-8 Parameters for configuring a UDP listener

Parameter	Description
Name	Specifies the listener name.

Parameter	Description
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients. Select **UDP**.
Frontend Port	Specifies the port that will be used by the load balancer to receive requests from clients. The port number ranges from 1 to 65535.
Access Control	Specifies how access to the listener is controlled. For details, see **What Is Access Control?** The following options are available: • All IP addresses • Blacklist • Whitelist
IP Address Group	Specifies the IP address group associated with a whitelist or blacklist. If there is no IP address group, create one first. For more information, see **IP Address Group**.
Transfer Client IP Address	Specifies whether to transmit IP addresses of the clients to backend servers.
Advanced Settings	
Description	Provides supplementary information about the listener. You can enter a maximum of 255 characters.

6. Click **Next: Configure Request Routing Policy**.
   a. You are advised to select an existing backend server group.
   b. You can also click **Create new** to create a backend server group.
      i. Configure the backend server group based on **Table 2-27**.
      ii. Click **Next: Add Backend Server**. Add backend servers and configure health check for the backend server group.
      For details about how to add backend servers, see **Backend Server Overview**. For the parameters required for configuring a health check, see **Table 2-28**.
7. Click **Next: Confirm**.
8. 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. 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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**. Configure the parameters based on **Table 2-9**.

Table 2-9 Parameters for configuring an HTTP listener

Parameter	Description
Name	Specifies the listener name.
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients. Select **HTTP**.
Frontend Port	Specifies the port that will be used by the load balancer to receive requests from clients. The port number ranges from 1 to 65535.
Redirect	Specifies whether to enable redirection. If you have both HTTPS and HTTP listeners, you can use this function to redirect the requests from the HTTP listener to the HTTPS listener to ensure security. If you create a redirect for an HTTP listener, the backend server will return HTTP 301 Move Permanently to the clients.
Redirected To	Select the HTTPS listener to which requests are redirected.
Access Control	Specifies how access to the listener is controlled. For details, see **What Is Access Control?** The following options are available: • All IP addresses • Blacklist • Whitelist

Parameter	Description
IP Address Group	Specifies the IP address group associated with a whitelist or blacklist. If there is no IP address group, create one first. For more information, see **IP Address Group**.
Transfer Client IP Address	Specifies whether to transmit IP addresses of the clients to backend servers.
Advanced Settings	
Transfer Load Balancer EIP	Specifies whether to store the EIP bound to the load balancer in the X-Forwarded-ELB-IP header field and pass this field to backend servers. Enable this option if you want to transparently transmit the EIP of the load balancer to backend servers.
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**.
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.

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

a. You are advised to select an existing backend server group.
b. You can also click **Create new** to create a backend server group.
   i. Configure the backend server group based on **Table 2-27**.
   ii. Click **Next: Add Backend Server**. Add backend servers and configure health check for the backend server group.

   For details about how to add backend servers, see **Backend Server Overview**. For the parameters required for configuring a health check, see **Table 2-28**.

7. Click **Next: Confirm**.
8. 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, ensure that ACL rules are not configured for this subnet. If rules are configured, request packets may not be allowed.

Constraints

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

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. Click  in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**. Configure the parameters based on **Table 2-10**.

Table 2-10 Parameters for configuring an HTTPS listener

Parameter	Description
Name	Specifies the listener name.

Parameter	Description
Frontend Protocol	Specifies the protocol that will be used by the load balancer to receive requests from clients. Select **HTTPS**.
Frontend Port	Specifies the port that will be used by the load balancer to receive requests from clients. The port number ranges from 1 to 65535.
SSL Authentication	Specifies how you want the clients and backend servers to be authenticated. There are two options: **One-way authentication** or **Mutual authentication**. • If only server authentication is required, select **One-way authentication**. • If you want the clients and the load balancer to authenticate each other, select **Mutual authentication**. Only authenticated clients will be allowed to access the load balancer.
CA Certificate	Specifies the certificate that allows the clients and backend servers to mutually authenticate each other. For details, see **Adding a Certificate**.
Server Certificate	Specifies the certificate that will be used by the backend server to authenticate the client when HTTPS is used as the frontend protocol. Both the certificate and private key are required. For details, see **Adding a Certificate**.
Enable SNI	Specifies whether to enable SNI when HTTPS is used as the frontend protocol. SNI is an extension to TLS and is used when a server uses multiple domain names and certificates. This allows the client to submit the domain name information while sending an SSL handshake request. After the load balancer receives the request, the load balancer queries the corresponding certificate based on the domain name and returns it to the client. If no certificate is found, the load balancer will return the default certificate. For details, see **SNI Certificate**.
SNI Certificate	Specifies the certificate associated with the domain name when the frontend protocol is HTTPS and SNI is enabled. Select an existing certificate or create one. For details, see **Adding a Certificate**.

Parameter	Description
Access Control	Specifies how access to the listener is controlled. For details, see **What Is Access Control?** The following options are available: • All IP addresses • Blacklist • Whitelist
IP Address Group	Specifies the IP address group associated with a whitelist or blacklist. If there is no IP address group, create one first. For more information, see **IP Address Group**.
Transfer Client IP Address	Specifies whether to transmit IP addresses of the clients to backend servers.
Advanced Settings	
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 **HTTP/2**.
Transfer Load Balancer EIP	Specifies whether to store the EIP bound to the load balancer in the X-Forwarded-ELB-IP header field and pass this field to backend servers. Enable this option if you want to transparently transmit the EIP of the load balancer to backend servers.
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.

6. Click **Next: Configure Request Routing Policy**.
   a. You are advised to select an existing backend server group.
   b. You can also click **Create new** to create a backend server group.
      i. Configure the backend server group based on **Table 2-27**.
      ii. Click **Next: Add Backend Server**. Add backend servers and configure health check for the backend server group.
      For details about how to add backend servers, see **Backend Server Overview**. For the parameters required for configuring a health check, see **Table 2-28**.
7. Click **Next: Confirm**.
8. Confirm the configurations and click **Submit**.

2.3.6 Forwarding Policy

Scenarios

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

This is suited for applications that are deployed on multiple backend servers and provide multiple types of services such as videos, images, audios, and texts.

A forwarding policy consists of a forwarding rule and an action.

- There are two types of forwarding rules: domain name and URL.
- HTTP listeners can forward requests to a backend server group and redirect requests to another listener.
- HTTPS listeners can forward requests to a backend server group.

How Requests Are Matched

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

Table 2-11 Example forwarding policies

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

NOTE

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

Constraints

- Forwarding policies can be added only to HTTP and HTTPS listeners.
- Forwarding policies must be unique.
- A maximum of 100 forwarding policies can be configured for a listener. If the number of forwarding policies exceeds the quota, the excess forwarding policies will not be applied.
- When you add a forwarding policy, note the following:
  - Each URL path must exist on the backend server. 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 matching ends when any rule is successfully matched. Matching rules cannot overlap with each other.
  - A URL path cannot be configured for two forwarding policies.
  - A domain name cannot exceed 100 characters.

CAUTION

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

Adding a Forwarding Policy

1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click  in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. On the **Listeners** tab, add a forwarding policy in either of the following ways:
   - On the **Listeners** page, locate the listener, and click **Add/Edit Forwarding Policy** in the **Forwarding Policies** column.
   - Locate the target listener, click its name, and click **Forwarding Policies**.
6. Click **Add Forwarding Policy**. Configure the parameters based on **Table 2-12**.
7. After the configuration is complete, click **Save**.

Table 2-12 Forwarding policy parameters

Parameter		Description	Example Value
Forwarding Rule	Domain name	Specifies the domain name used for forwarding requests. The domain name in the request must exactly match that in the forwarding policy. You need to specify either a domain name or URL.	www.test.com

Parameter		Description	Example Value
	URL	Specifies the URL used for forwarding requests. There are three URL matching rules: • Exact match The request URL must exactly match that specified in the forwarding policy. • Prefix match The requested URL starts with the specified URL string. • Regular expression match The requested URL matches the specified URL string based on the regular expression.	/login.php
Action	Forward to a backend server group	If the request matches the configured forwarding rule, the request is forwarded to the specified backend server group.	Forward to a backend server group
	Redirect to another listener	If the request matches the configured forwarding rule, the request is redirected to the specified HTTPS listener. This action can be configured only for HTTP listeners. **NOTE** If you select **Redirect to another listener** and create a redirect for the current listener, this listener will redirect the requests to the specified HTTPS listener, but access control configured for the listener will still take 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

Parameter	Description	Example Value
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

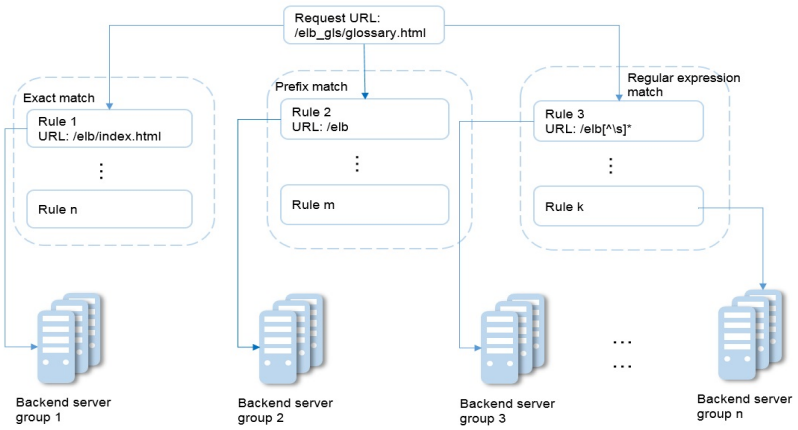
URL Matching Example

The following table lists how a URL is matched, and **Figure 2-6** shows how a request is forwarded to a backend server group.

Table 2-13 URL matching

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

Figure 2-6 Request forwarding



In this figure, the system first searches for an exact match of the requested 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. 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** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Click **Listeners**, locate the listener, and click its name.
6. On the **Forwarding Policies** tab, select the forwarding policy, and click **Edit**.
7. Modify the parameters and click **Save**.

Deleting a Forwarding Policy

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

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

2.3.7 Modifying 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 load balancer by referring to **Creating a Shared Load Balancer**.
- 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. 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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Click **Listeners** tab, locate the listener, and click its name.
6. On the **Summary** tab, click **Configure** next to **Modification Protection**.
7. 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. 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** > **Elastic Load Balance**.

4. On the **Load Balancers** page, locate the load balancer and click its name.
5. 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.
6. 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. 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** > **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 the name of the listener.
6. On the **Summary** tab, click **Edit** on the top right.
7. In the **Edit** dialog box, expand **Advanced Settings**.
8. Configure **Idle Timeout (s)**, **Request Timeout (s)**, or **Response Timeout (s)** as you need.
9. Click **OK**.

Changing the Backend Server Group of a Listener

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

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

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

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

NOTE

The backend protocol of the new backend server group must match the frontend protocol of the listener.

8. Click **OK**.

2.3.8 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.

Constraints

If modification protection is enabled for a listener, the listener cannot be deleted or modified.

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. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Click the **Listeners** tab, locate the listener, and click **Delete** in the **Operation** column.
6. In the displayed dialog box, enter **DELETE**.
7. 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 2-14 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 2-15 Adding backend servers

Backend Server Type	Description	Reference
Cloud servers	You can add ECSs or BMSs 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**: Traffic is routed only to healthy backend servers in the backend server group.

Key Functions

You can configure the key functions listed in **Table 2-16** for each backend server group to ensure service stability.

Table 2-16 Key functions

Key Function	Description	Detail
Health Check	Specifies whether to enable the health check option. Health checks determine whether backend servers are healthy. If a backend server is detected unhealthy, it will not receive requests from the associated load balancer, improving your service reliability.	**Health Check**
Load Balancing Algorithm	The load balancer distributes traffic based on the load balancing algorithm you have configured for the backend server group.	**Load Balancing Algorithms**
Sticky Session	Specifies whether to enable the sticky session option. If you enable this option, all requests from a client during one session are sent to the same backend server.	**Sticky Session**

Precautions for Creating a Backend Server Group

The backend protocol of the new backend server group must match the frontend protocol of the listener as described in **Table 2-17**.

You can create a backend server group by referring to **Table 2-18**.

Table 2-17 The frontend and backend protocol

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

Table 2-18 Creating a backend server group

Load Balancer Type	Constraints	Reference
Shared	A backend server group can be associated with only one shared load balancer and used by only one listener.	**Creating a Backend Server Group**

2.4.2 Key Features

2.4.2.1 Health Check

ELB periodically sends requests to backend servers to check whether they can process requests. This process is called health check.

If a backend server is detected unhealthy, the load balancer will stop route requests to it. After the backend server recovers, the load balancer will resume routing requests to it.

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

Health Check Protocol

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

If you want to modify health check settings, see details in **Enabling or Disabling Health Check**.

Select a health check protocol that matches the backend protocol as described in **Table 2-19**.

Table 2-19 The backend protocol 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

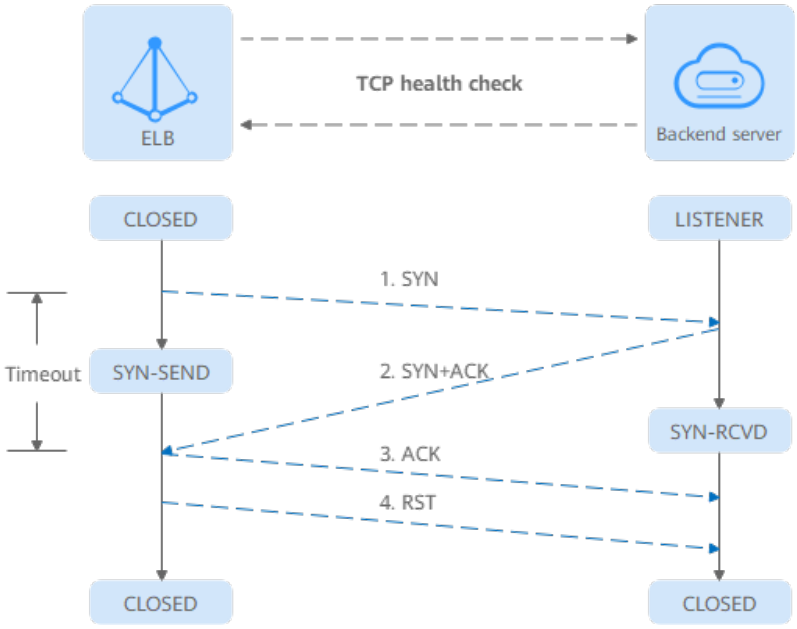
Health Check Source IP Address

A shared load balancer uses an IP address in 100.125.0.0/16 to send requests to backend servers and verify their health status. 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 2-7 TCP health check



The TCP health check process is as follows:

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

NOTICE

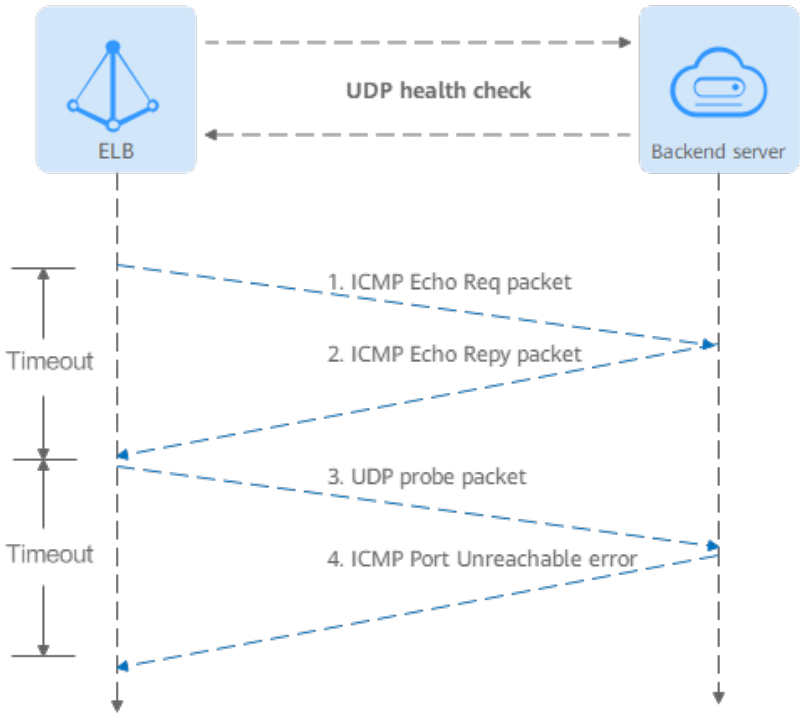
After a successful TCP three-way handshake, an RST packet will be sent to close the TCP connection. The application on the backend server may consider this packet a connection error and reply with a message, for example, "Connection reset by peer". To avoid this issue, take either of the following actions:

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

UDP Health Check

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

Figure 2-8 UDP health check



The UDP health check process is as follows:

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

HTTP Health Check

You can also configure HTTP health checks to obtain server statuses through HTTP GET requests if you select TCP, HTTP, or HTTPS as the backend protocol. **Figure 2-9** shows how an HTTP health check works.

Figure 2-9 HTTP health check

The HTTPS health check process is as follows:

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

NOTE

If HTTP health check is selected for the TCP listener of a shared 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. Otherwise, the health check may fail.

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 2-20**.

Table 2-20 Factors affecting the health check time window

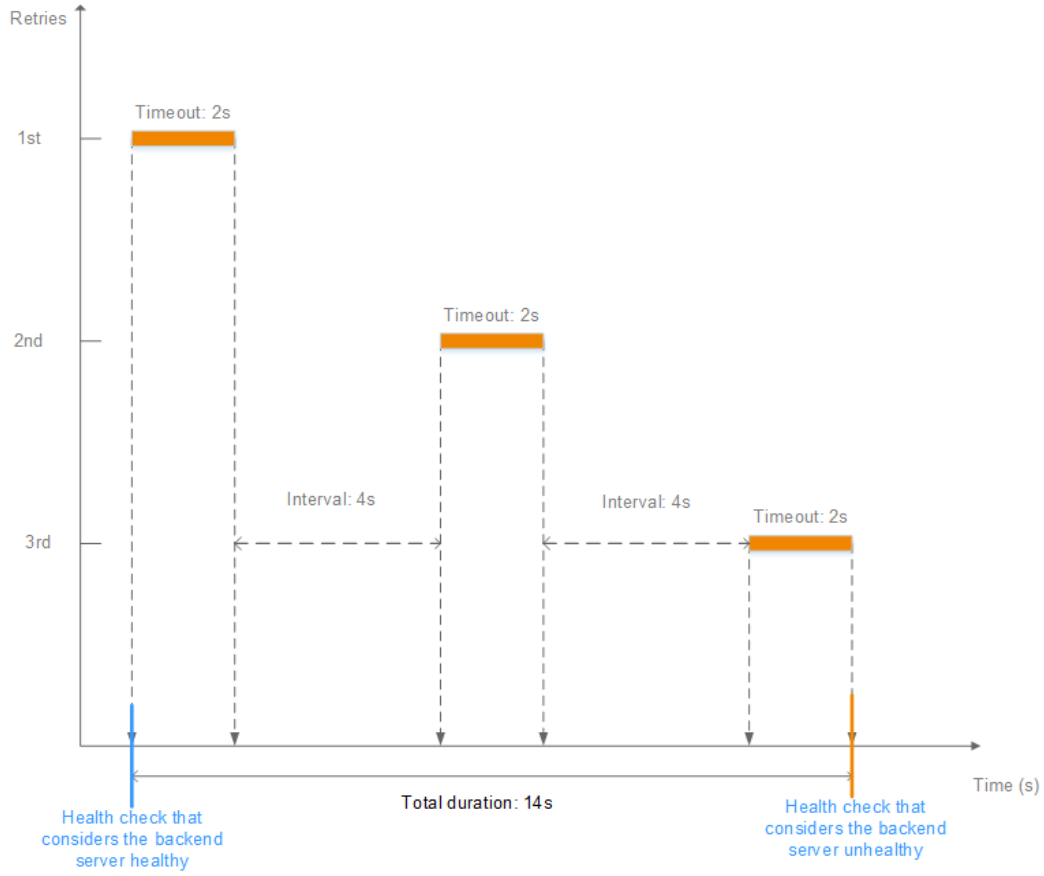
Factor	Description
Check Interval	How often health checks are performed.
Timeout Duration	How long the load balancer waits for the response from the backend server.
Health Check Threshold	The number of consecutive successful or failed health checks required for determining whether the backend server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

As shown in **Figure 2-10**, if the health check interval is 4s, the health check timeout duration is 2s, and unhealthy threshold is 3, the time window for a backend server to be considered unhealthy is calculated as follows: 2 x 3 + 4 x (3 – 1) = 14s.

Figure 2-10 Health check timeout duration



Rectifying an Unhealthy Backend Server

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

2.4.2.2 Load Balancing Algorithms

Overview

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

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 2-21 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 2-11 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 2-11 Traffic distribution using the weighted round robin algorithm

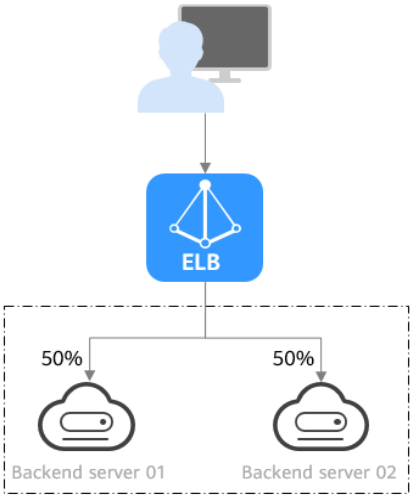


Table 2-22 Weighted round robin

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 2-12 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 have been established with backend server 02. New requests are preferentially routed to backend server 02.

Figure 2-12 Traffic distribution using the weighted least connections algorithm

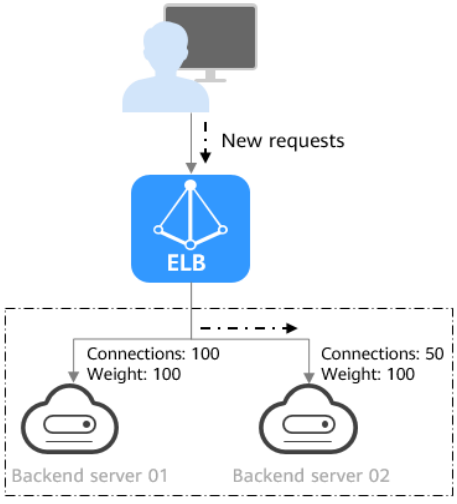


Table 2-23 Weighted least connections

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 2-13 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 IP address A to backend server 01.

Figure 2-13 Traffic distribution using the source IP hash algorithm

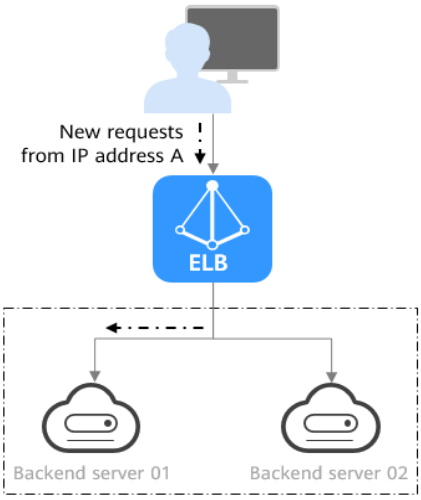


Table 2-24 Source IP hash

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.

2.4.2.3 Sticky Session

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

Here is an example that describes how sticky session works. Assume that you have logged in to a server. After a while, you send another request. If sticky sessions are

not enabled, the request may be routed to another server, and you will be asked to log in again. If sticky sessions are enabled, all your requests are processed by the same server, and you do not need to repeatedly log in.

Differences Between Sticky Sessions at Layer 4 and Layer 7

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

Table 2-25 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 are 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 same 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**.

Constraints and Limitations

- If you use **Cloud Connect connection**, **Direct Connect** or **VPN** to access ELB, you must select **Source IP hash** as the load balancing algorithm and disable sticky sessions for ELB.

- 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.

2.4.3 Creating a Backend Server Group

Scenario

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

NOTE

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

You can create a backend server group in the ways listed in **Table 2-26**.

Table 2-26 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. 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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. Click **Create Backend Server Group** in the upper right corner.
6. Configure the routing policy based on **Table 2-27**.

Table 2-27 Parameters required for configuring a routing policy

Parameter	Description
Load Balancing Type	Specifies the type of load balancers 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 protocol varies depending on the forwarding mode: 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**.

Parameter	Description
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 **Sticky Session**.
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 same 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.

7. Click **Next** to add backend servers and configure health check based on **Table 2-28**. For more information about health checks, see **Health Check**.

Table 2-28 Parameters required for configuring a health check

Parameter	Description
Health Check	Specifies whether to enable health checks. If the health check is enabled, click  next to **Advanced Settings** to set health check parameters.
Health Check Protocol	• 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 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 (/). 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**.

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

2.4.4 Modifying a Backend Server Group

2.4.4.1 Change Scenarios

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

Health Check

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

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

For details about how to modify health check settings, see **Enabling or Disabling Health Check**.

Basic Information

You can modify the basic information of a backend server group listed in **Table 2-29**.

Table 2-29 Basic information that can be modified

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

2.4.4.2 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.

Notes and 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. 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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** tab, locate the backend server group.
6. On the **Summary** page, click **Health Check** on the right.
7. In the **Configure Health Check** dialog box, configure the parameters based on **Table 2-30**.

Table 2-30 Parameters required for configuring health check

Parameter	Description
Health Check	Specifies whether to enable health checks.

Parameter	Description
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**.
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**.

8. Click **OK**.

Disabling Health Check

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

2.4.4.3 Changing the Load Balancing Algorithm

Scenario

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

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

Procedure

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

NOTE

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

2.4.4.4 Modifying Sticky Session Settings

Scenario

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

NOTE

You can also configure sticky sessions when adding a listener or creating a backend server group.

Enabling Sticky Session

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

Disabling Sticky Session

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

2.4.5 Changing a Backend Server Group

Scenario

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

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

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

Constraints and Limitations

- The backend server group cannot be changed if redirection is enabled.
- The backend protocol of the backend server group must match the frontend protocol of the listener. For details, see **Table 2-17**.
- You can only associate a backend server group that is not used by any listener with a shared load balancer.

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. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the target load balancer and click its name.
5. On the **Listeners** tab, locate the target listener and click its name.
6. On the **Summary** tab, click **Change Backend Server Group** on the right of **Default Backend Server Group** area.
7. In the displayed dialog box, click the server group name box.

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

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

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

   NOTE

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

2.4.6 Viewing a Backend Server Group

Scenario

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

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

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the backend server group.
6. Click different tabs to view the required information.
   a. On the **Summary** tab, view the basic information and health check settings.
   b. On the **Backend Servers** tab, view the servers that have been added to the backend server group.

2.4.7 Deleting a Backend Server Group

Scenario

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

Constraints and Limitations

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

Procedure

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

Constraints and Limitations

- A maximum of 500 backend servers can be added to a backend server group.
- Inbound security group rules must be configured to allow traffic over the port of each backend server and health check port. For details, see **Security Group 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 2-31**. For more information about load balancing algorithms, see **Load Balancing Algorithms**.

Table 2-31 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.

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 must allow traffic from the 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 backend subnet of the load balancer, the rules must allow traffic from the backend subnet of the load balancer to 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 the TCP or UDP 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 and Limitations

- If health check is enabled for a backend server group, security group rules must allow traffic from the health check port over the health check protocol.
- If UDP is used for health check, there must be a rule that allows ICMP traffic. 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 only communications among the servers in the VPC. To ensure that the load balancer can communicate with these servers over both the frontend port and health check port, configure inbound rules for security groups containing these servers.

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

   The page providing details about the ECS is displayed.
5. Click **Security Groups**, locate the security group, and view security group rules.
6. Click the ID of a security group rule or **Modify Security Group Rule**. The security group details page is displayed.
7. On the **Inbound Rules** tab, click **Add Rule**. Configure an inbound rule based on **Table 2-32**.

Table 2-32 Security group rules

Backend Protocol	Policy	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

8. Click **OK**.

Configuring Network ACL Rules

To control traffic in and out of a subnet, you can associate a network ACL with the subnet. Network ACL rules control access to subnets and add an additional layer of defense to your subnets. 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 used to access backend servers into private IP addresses in 100.125.0.0/16. You cannot configure network ACL rules 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 from 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 and Limitations

Only servers in the same VPC as the load balancer can be added.

Adding a Cloud Server

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
1. On the **Backend Server Groups** page, click the name of the backend server group.
2. Switch to the **Backend Servers** tab and click **Add** on the right.
3. Search for backend servers using specified keywords.
4. 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. 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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, click the name of the target backend server group.
6. On the **Backend Servers** tab, click **Backend Servers**.
7. Select the cloud servers and click **Modify Weight** up above the cloud server list.
8. In the displayed dialog box, modify ports/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.

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

2.6 Security

2.6.1 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

Listener Type	Enabling Transfer Client IP Address	Disabling Transfer Client IP Address
HTTP and HTTPS	Enabled by default	Not supported

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

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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. 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.
6. In the displayed dialog box, enable **Transfer Client IP Address**.
7. Confirm the configurations and click **OK**.

NOTE

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.

Disabling Transfer Client IP Address

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. 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.
6. In the displayed dialog box, disable **Transfer Client IP Address**.
7. 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**

2.6.2 HTTP/2

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

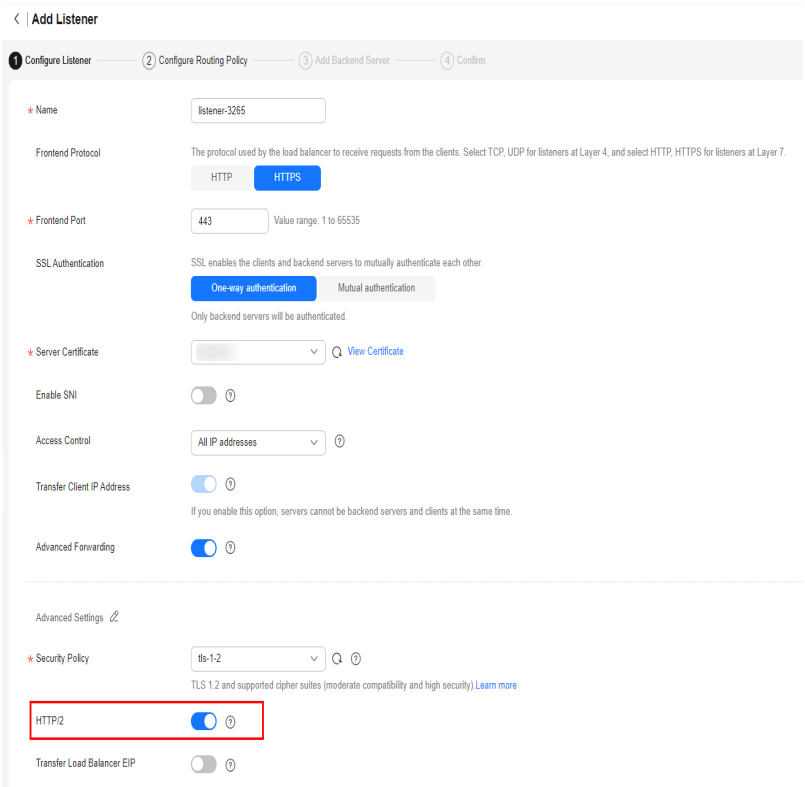
You can enable HTTP/2 when you add an HTTPS listener. You can enable or disable HTTP/2 for an existing HTTPS listener.

Enabling HTTP/2 When Adding a Listener

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

1. 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** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**.
6. In the **Add Listener** dialog box, set **Frontend Protocol** to **HTTPS**.
7. Expand **Advanced Settings** and enable HTTP/2.
8. Confirm the configurations and go to the next step.

Figure 2-14 Enabling HTTP/2

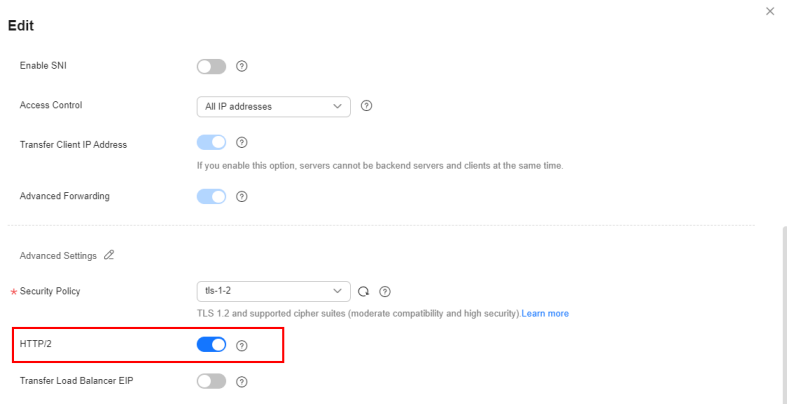


Enabling or Disabling HTTP/2 for an Existing Listener

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

4. Locate the load balancer and click its name.
5. Click **Listeners**, locate the listener, and click its name.
6. On the **Summary** tab, click **Edit** on the top right.
7. In the **Edit** dialog box, expand **Advanced Settings** and enable or disable HTTP/2.
8. Click **OK**.

Figure 2-15 Disabling or enabling HTTP/2



2.6.3 SNI Certificate

Scenarios

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

SNI, an extension to Transport Layer Security (TLS), enables a server to present multiple certificates on the same IP address and port number. After you enable SNI, the client can submit the requested domain name at the start of the SSL handshake. After receiving the request, the load balancer searches for the certificate based on the domain name. If the certificate is found, the load balancer will return it to the client. If the certificate is not found, the load balancer will return the default certificate.

Constraints

- 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.

Prerequisites

- You have created a load balancer by referring to **Creating a Shared Load Balancer**.
- You have created an SNI certificate by referring to **Adding a Certificate**.
- You have added an HTTPS listener to the load balancer by referring to **Adding an HTTPS Listener**.

Restrictions

- If a certificate has expired, you need to manually replace or delete it by following the instructions in **Binding or Replacing a Certificate**.
- You must specify at least one domain name for each certificate. The domain name must be the same as that in the certificate.
- A domain name can be used by both an ECC certificate and an RSA certificate. If there are two SNI certificates that use the same domain name, the ECC certificate is displayed preferentially.
- Domain names in an SNI certificate are matched as follows:

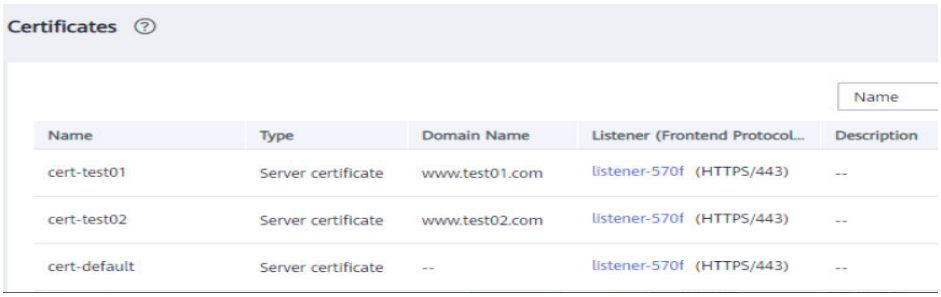
  If the domain name of the certificate is *.test.com, a.test.com and b.test.com are supported, but a.b.test.com and c.d.test.com are not supported.

  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.
- As shown in **Figure 2-16**, **cer-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 certificate will be used for authentication.

Figure 2-16 Configuring certificates



Enabling SNI for an HTTPS Listener

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner to display **Service List** and choose **Networking** > **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. On the **Summary** tab, click **Configure** on the right of SNI.
7. Enable SNI and select an SNI certificate.
8. Click **OK**.

2.6.4 TLS Security Policy

Scenarios

HTTPS encryption is commonly used for applications that require secure transmission of data, 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 Default Security Policy

1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click  in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Under **Listeners**, click **Add Listener**.
6. On the **Add Listener** page, set **Frontend Protocol** to **HTTPS**.
7. Expand **Advanced Settings** 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 ELB and clients are used, and the cipher suites supported by ELB take precedence.

8. Confirm the configurations and go to the next step.

Differences Among Security Policies

Table 2-36 Differences Among 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	hybrid-policy-1-0
TLS version									
Protocol-TLS 1.3	N/A	N/A	N/A	N/A	N/A	Supported	Supported	Supported	N/A
Protocol-TLS 1.2	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
Protocol-TLS 1.1	Supported	Supported	N/A	Supported	N/A	Supported	N/A	N/A	Supported
Protocol-TLS 1.0	Supported	N/A	N/A	Supported	N/A	Supported	N/A	N/A	N/A
Cipher suite									
EDHE-RSA-AES128-GCM-SHA256	Supported	Supported	Supported	N/A	Supported	N/A	N/A	N/A	N/A
ECDHE-RSA-AES256-GCM-SHA384	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
ECDHE-RSA-AES128-SHA256	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
ECDHE-RSA-AES256-SHA384	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
AES128-GCM-SHA256	Supported	Supported	Supported	Supported	Supported	Supported	N/A	N/A	Supported
AES256-GCM-SHA384	Supported	Supported	Supported	Supported	Supported	Supported	N/A	N/A	Supported
AES128-SHA256	Supported	Supported	Supported	Supported	Supported	Supported	N/A	N/A	Supported

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	hybrid-policy-1-0
AES256-SHA256	Supported	Supported	Supported	Supported	Supported	Supported	N/A	N/A	Supported
ECDHE-RSA-AES128-SHA	Supported	Supported	Supported	Supported	N/A	Supported	N/A	N/A	Supported
ECDHE-RSA-AES256-SHA	Supported	Supported	Supported	Supported	N/A	Supported	N/A	N/A	Supported
AES128-SHA	Supported	Supported	Supported	Supported	N/A	Supported	N/A	N/A	Supported
AES256-SHA	Supported	Supported	Supported	Supported	N/A	Supported	N/A	N/A	Supported
ECDHE-ECDSA-AES128-GCM-SHA256	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
ECDHE-ECDSA-AES128-SHA256	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
ECDHE-ECDSA-AES128-SHA	Supported	Supported	Supported	Supported	N/A	Supported	N/A	N/A	Supported
ECDHE-ECDSA-AES256-GCM-SHA384	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
ECDHE-ECDSA-AES256-SHA384	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported	Supported
ECDHE-ECDSA-AES256-SHA	Supported	Supported	Supported	Supported	N/A	Supported	N/A	N/A	Supported
ECDHE-RSA-AES128-GCM-SHA256	N/A	N/A	N/A	Supported	N/A	Supported	Supported	Supported	Supported

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	hybrid-policy-1-0
TLS_AES_256_GCM_SHA384	N/A	N/A	N/A	N/A	N/A	Supported	Supported	Supported	N/A
TLS_CHACHA20_POLY1305_SHA256	N/A	N/A	N/A	N/A	N/A	Supported	Supported	Supported	N/A
TLS_AES_128_GCM_SHA256	N/A	N/A	N/A	N/A	N/A	Supported	Supported	Supported	N/A
TLS_AES_128_CCM_8_SHA256	N/A	N/A	N/A	N/A	N/A	Supported	Supported	Supported	N/A
TLS_AES_128_CCM_SHA256	N/A	N/A	N/A	N/A	N/A	Supported	Supported	Supported	N/A
DHE-RSA-AES128-SHA	N/A	N/A	N/A	Supported	N/A	N/A	N/A	N/A	N/A
DHE-DSS-AES128-SHA	N/A	N/A	N/A	Supported	N/A	N/A	N/A	N/A	N/A
CAMELLIA128-SHA	N/A	N/A	N/A	Supported	N/A	N/A	N/A	N/A	N/A
EDH-RSA-DES-CBC3-SHA	N/A	N/A	N/A	Supported	N/A	N/A	N/A	N/A	N/A
DES-CBC3-SHA	N/A	N/A	N/A	Supported	N/A	N/A	N/A	N/A	N/A
ECDHE-RSA-RC4-SHA	N/A	N/A	N/A	Supported	N/A	N/A	N/A	N/A	N/A
RC4-SHA	N/A	N/A	N/A	Supported	N/A	N/A	N/A	N/A	N/A
DHE-RSA-AES256-SHA	N/A	N/A	N/A	Supported	N/A	N/A	N/A	N/A	N/A

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	hybrid-policy-1-0
DHE-DSS-AES256-SHA	N/A	N/A	N/A	Supported	N/A	N/A	N/A	N/A	N/A
DHE-RSA-CAMELLIA256-SHA	N/A	N/A	N/A	Supported	N/A	N/A	N/A	N/A	N/A
ECC-SM4-SM3	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Supported
ECDHE-SM4-SM3	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Supported

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. 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** > **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. On the **Summary** tab, click **Edit** on the top right.
7. In the **Edit** dialog box, expand **Advanced Settings** and change the security policy.
8. Click **OK**.

2.6.5 Access Control

2.6.5.1 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.

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 backend servers 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. 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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Configure access control for a listener in either of the following ways:
   - On the **Listeners** page, locate the listener and click **Configure** in the **Access Control** column.
   - Click the name of the target listener. On the **Summary** page, click **Configure** on the right of **Access Control**.
6. In the displayed **Configure Access Control** dialog box, configure parameters as 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.

7. Click **OK**.

2.6.5.2 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.

- **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. 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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
5. On the displayed page, click **Create IP Address Group**.
6. Configure the parameters based on **Table 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 IP addresses or CIDR blocks in each IP address group.	• Without remarks: 10.168.2.24 • With remarks: 10.168.16.0/24 | ECS01
Description	Provides supplementary information about the IP address group.	N/A

7. 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 IP addresses or CIDR blocks to each IP address group.

Adding IP Addresses

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

1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click  in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
5. On the **IP Address Groups** page, locate the target IP address group and click its name.
6. In the lower part of the displayed page, choose **IP Addresses** tab and click **Add IP Addresses**. On the **Add IP Addresses** page, add IP addresses.
7. Click **OK**.

Changing IP Addresses

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

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

5. On the **IP Address Groups** page, you can:
   a. Modify the basic information and change IP addresses of an IP address group:
      i. Locate the target address group, click **Modify** in the **Operation** column. 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 Address**, 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. 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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
5. On the **IP Address Groups** page, locate the target IP address group and click its name.
6. In the IP address list, locate the IP address you want to delete and click **Delete** in the **Operation** column.
7. 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. 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** > **Elastic Load Balance**.

4. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
5. On the **IP Address Groups** page, locate the target IP address group and click its name.
6. 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. 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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **IP Address Groups**.
5. On the **IP Address Groups** page, locate the IP address group and click **Delete** in the **Operation** column.
6. Click **OK**.

2.6.6 Certificate

2.6.6.1 Certificate Overview

ELB supports two types of certificates. If 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.

- **Server certificate**: used for SSL handshake negotiations if an HTTPS listener is used. Both the certificate content and private key are required.
- **CA certificate**: issued by a certificate authority (CA) and used to verify the certificate issuer. If HTTPS mutual authentication is required, HTTPS connections can be established only when the client provides a certificate issued by a specific CA.

NOTE

SSL Certificate Manager (SCM) allows you to purchase a certificate from Huawei Cloud or upload your own certificates for easier management.

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 supports certificates only in PEM format. If you have a certificate in any other format, you must convert it to a PEM-encoded certificate.
- If a certificate has expired, you need to manually replace or delete it.

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

The following is an example:

```
-----BEGIN CERTIFICATE-----
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 must contain 64 characters except the last row.

The following is an example:

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

2.6.6.2 Converting Certificate Formats

Scenarios

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

From DER to PEM

The DER format is usually used on a Java platform.

Run the following command to convert the certificate format:

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

Run the following command to convert the private key format:

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

From P7B to PEM

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

Run the following command to convert the certificate format:

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

From PFX to PEM

The PFX format is usually used by Windows Server.

Run the following command to convert the certificate format:

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

Run the following command to convert the private key format:

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

2.6.6.3 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: You can purchase a certificate from SSL Certificate Manager (SCM) or upload your own certificates.
- CA certificate: You can only upload your own CA certificates.
- Server SM certificate: You can purchase a certificate from SSL Certificate Manager (SCM) or upload your own 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 **Networking** > **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 2-39**.

Table 2-39 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. • **CA certificate**: issued by a certificate authority (CA) and used to verify the certificate issuer. If HTTPS mutual authentication is required, HTTPS connections can be established only when the client provides a certificate issued by a specific CA.
Certificate Name	Specifies the name of your certificate. This parameter is only available for your certificates.
Enterprise Project	Specifies an enterprise project by which cloud resources and members are centrally managed.

Parameter	Description
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-----
Domain Name	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 **Networking** > **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 2-40**.

Table 2-40 CA certificate parameters

Parameter	Description
Certificate Type	Specifies the certificate type. Select **CA certificate**. • **Server certificate**: used for SSL handshake negotiations if an HTTPS listener is used. Both the certificate content and private key are required. • **CA certificate**: issued by a certificate authority (CA) and used to verify the certificate issuer. If HTTPS mutual authentication is required, HTTPS connections can be established only when the client provides a certificate issued by a specific CA.
Certificate Name	Specifies the name of the CA certificate.
Enterprise Project	Specifies an enterprise project by which cloud resources and members are centrally managed.
Certificate Content	The content must be in PEM format. Click **Upload** and select the certificate to be uploaded. Ensure that your browser is 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**.

2.6.6.4 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. 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** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Certificates**.
5. In the certificate list, click the listener name in the **Listener (Frontend Protocol/Port)** column to view its details.

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

Modifying a Certificate

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

Deleting a Certificate

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click ☰ in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Certificates**.
5. Locate the certificate and click **Delete** in the **Operation** column.
6. In the displayed dialog box, click **OK**.

2.6.6.5 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. 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** > **Elastic Load Balance**.
4. On the **Load Balancers** page, locate the load balancer and click its name.
5. Click the **Listeners** tab, locate the listener, and click **Edit** in **Operation** column.
6. On the displayed dialog box, select a server certificate or CA certificate.
7. Click **OK** in the **Edit** dialog box.

2.6.6.6 Replacing the Certificate Bound to Different Listeners

Scenario

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

NOTE

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

Constraints

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

Modifying a Certificate

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

2.6.7 Protection for Mission-Critical Operations

Scenarios

ELB supports sensitive operation protection. When you perform sensitive operations on the management console, you need to enter a credential that can prove your identity. You can perform corresponding operations only after your identity is authenticated. It is recommended that you enable operation protection to secure your account.

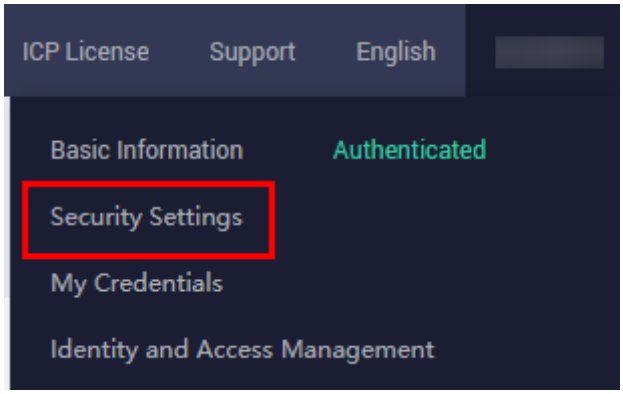
This function can be configured only by the administrator and takes effect for the resources in your account and the resources of users under your account. Common users have only the view permissions. To modify the permissions, contact the administrator.

Enabling Operation Protection

Operation protection is disabled by default. Perform the following operations to enable it:

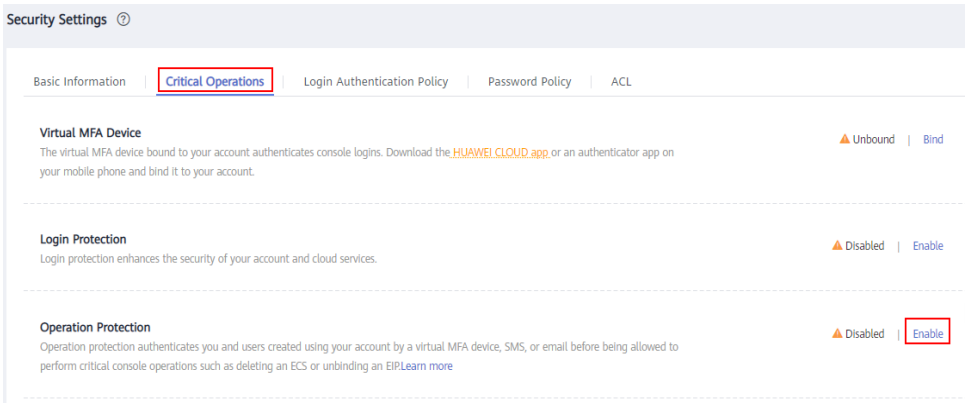
1. Log in to the management console.
2. Move the cursor to the username in the upper right corner of the page and select **Security Settings** from the list.

Figure 2-17 Security settings



3. On the **Security Settings** page, choose **Critical Operations** > **Operation Protection** > **Enable**.

   Figure 2-18 Critical operations



4. On the **Operation Protection** page, select **Enable**.

   If operation protection is enabled, you and IAM users created using your account need to enter a verification code when performing a critical operation, such as deleting an ECS resource.

   NOTE

   - When performing a critical operation, you will be asked to choose a verification method from email, SMS, and virtual MFA device.
     - If you have bound only a mobile number, only SMS verification is available.
     - If you have bound only an email address, only email verification is available.
     - If you have not bound an email address, mobile number, or virtual MFA device, bind one to perform critical operations.
   - You can change the mobile number, email address, and virtual MFA device on the **Basic Information** page.

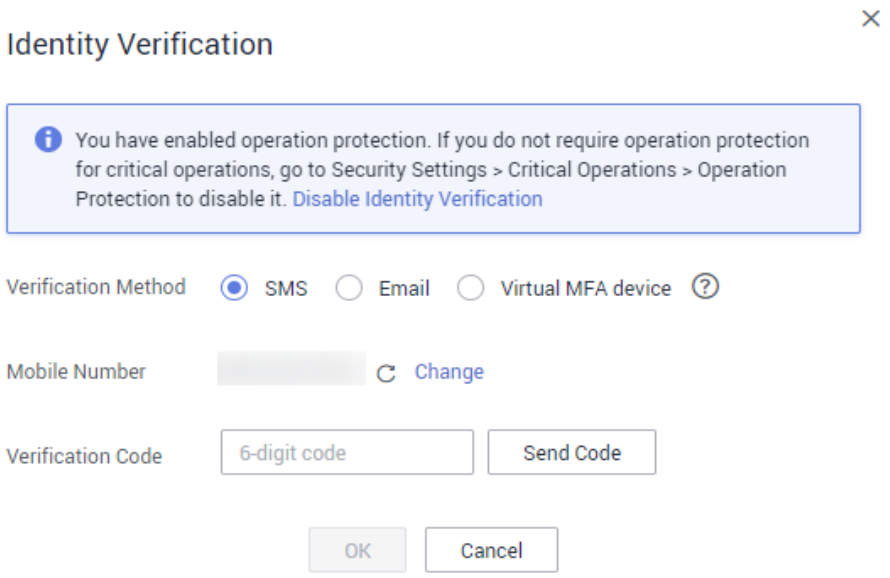
Verifying Operation Protection

After operation protection is enabled, when you perform a mission-critical operation, the system will verify your identity.

- If you have bound an email address, enter the email verification code.
- If you have bound a mobile number, enter the SMS verification code.
- If you have bound a virtual MFA device, enter a 6-digit dynamic verification code of the MFA device.

When you attempt to delete a load balancer, the following dialog box is displayed, and you need to select a verification method:

Figure 2-19 Identity verification

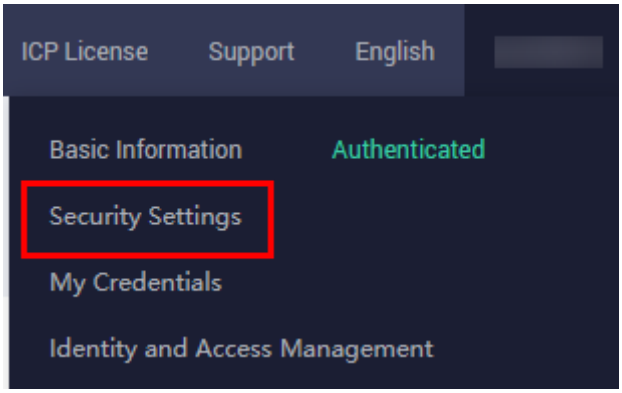


Disabling Operation Protection

Perform the following operations to disable operation protection:

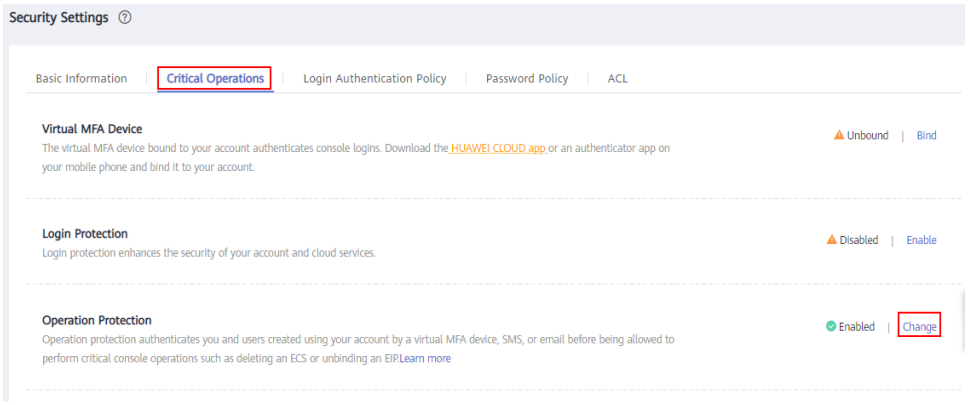
1. Log in to the management console.
2. Move the cursor to the username in the upper right corner of the page and select **Security Settings** from the list.

   Figure 2-20 Security settings

3. On the **Security Settings** page, choose **Critical Operations** > **Operation Protection** > **Change**.

Figure 2-21 Modifying operation protection settings



4. On the **Operation Protection** page, select **Disable** and click **OK**.

References

- **How Do I Bind a Virtual MFA Device?**
- **How Do I Obtain an MFA Verification Code?**

2.7 Access Logging

Scenarios

ELB logs HTTP and HTTPS requests received by load balancers, including the time when the request was sent, client IP address, request path, and server response.

With Log Tank Service (LTS), you can view logs of requests to load balancers at Layer 7 and analyze response status codes to quickly locate unhealthy backend servers.

NOTE

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

Notes and Constraints

- Access logs can be configured only for application (Layer 7) load balancers.
- 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. For details, see **Creating a Shared Load Balancer**.
- You have enabled LTS. For details, see **Accessing 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 add an HTTP or HTTPS listener to the load balancer. For details, see **Adding an HTTP Listener** or **Adding an HTTPS Listener**.

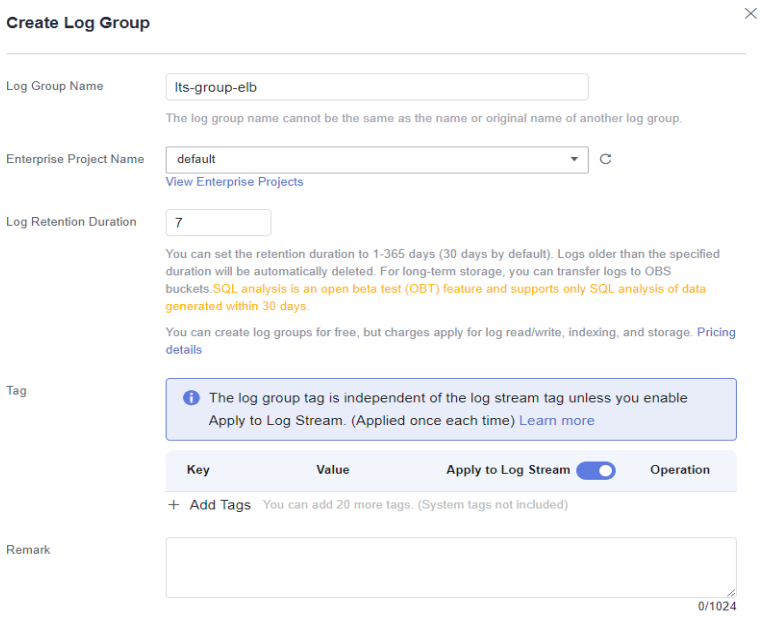
Flowchart

Figure 2-22 Process for locating an unhealthy backend server

Creating a Log Group

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. In the upper left corner of the page, click and select **Log Tank Service** under **Management & Governance**.
4. In the navigation pane on the left, choose **Log Management**.
5. On the lower part of the displayed page, click **Create Log Group**. In the displayed dialog box, enter a name for the log group.

   Figure 2-23 Creating a log group

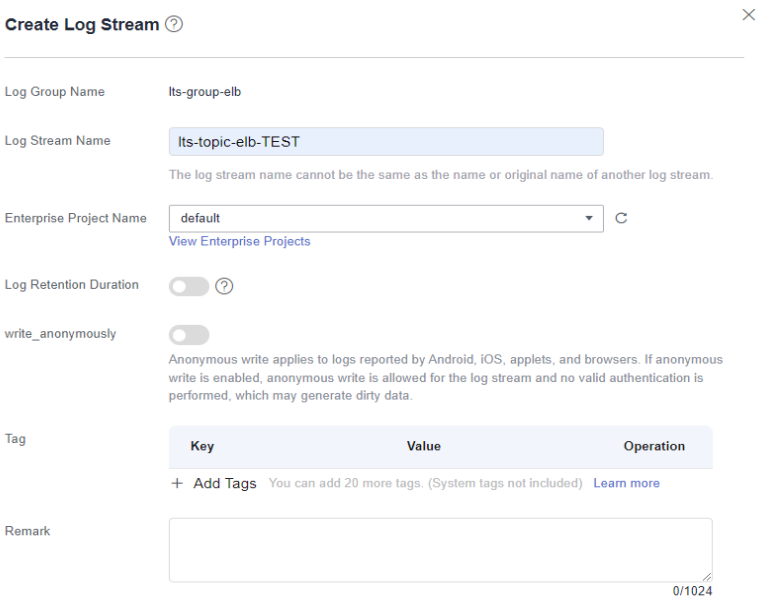


6. Confirm the settings and click **OK**.

Creating a Log Stream

1. On the LTS console, click on the left of the target log group.
2. Click **Create Log Stream**. In the displayed dialog box, enter a name for the log stream.

Figure 2-24 Creating a log stream

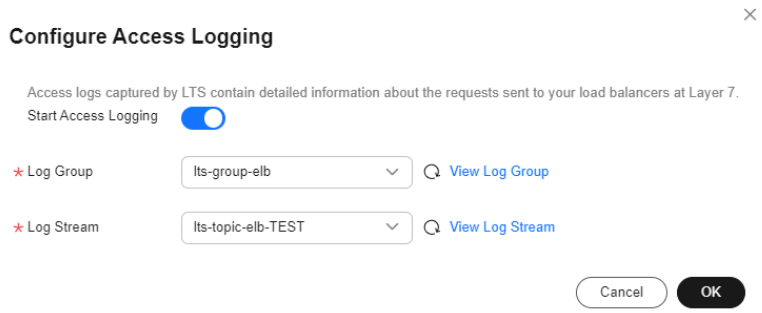


3. Confirm the settings and click **OK**.

Configuring Access Logging

1. Click ☰ in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
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.

Figure 2-25 Configuring access logging



5. Click **OK**.

NOTICE

Ensure that the log group is in the same region as the load balancer.

Viewing Access Logs

There are two ways for you to view access logs.

- On the ELB console, click the name of the load balancer and click **Access Logs** to view logs.
- (Recommended) On the LTS console, 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 2-41 describes the fields in the log.

Table 2-41 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.	-	[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

Parameter	Description	Value Description	Example Value
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
body_bytes_sent	Number of bytes sent to the client (excluding the response header).	Integer	3

Parameter	Description	Value Description	Example Value
request_time	Request processing time in seconds from the time when the load balancer receives the first request packet from the client to the time when the load balancer sends the response packet.	Floating-point data	0.011
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"

Parameter	Description	Value Description	Example Value
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"
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"

Parameter	Description	Value Description	Example Value
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	"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	"-"

Parameter	Description	Value Description	Example Value
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	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 the value can be **member_id**) or **-**.	String	"-"
tenant_id	Tenant ID.	String	f2bc165ad9b4483a9b17762da851bbbb

Parameter	Description	Value Description	Example Value
eip_address:eip_port	EIP of the load balancer and frontend port that were set when the listener was added.	EIP of the load balancer and frontend port that were set when the listener was added.	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"
certificate_id	[HTTPS listener] Certificate ID used for establishing an SSL connection. This field is not supported yet.	String	-
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

Parameter	Description	Value Description	Example Value
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	-

Log analysis

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

Analysis results:

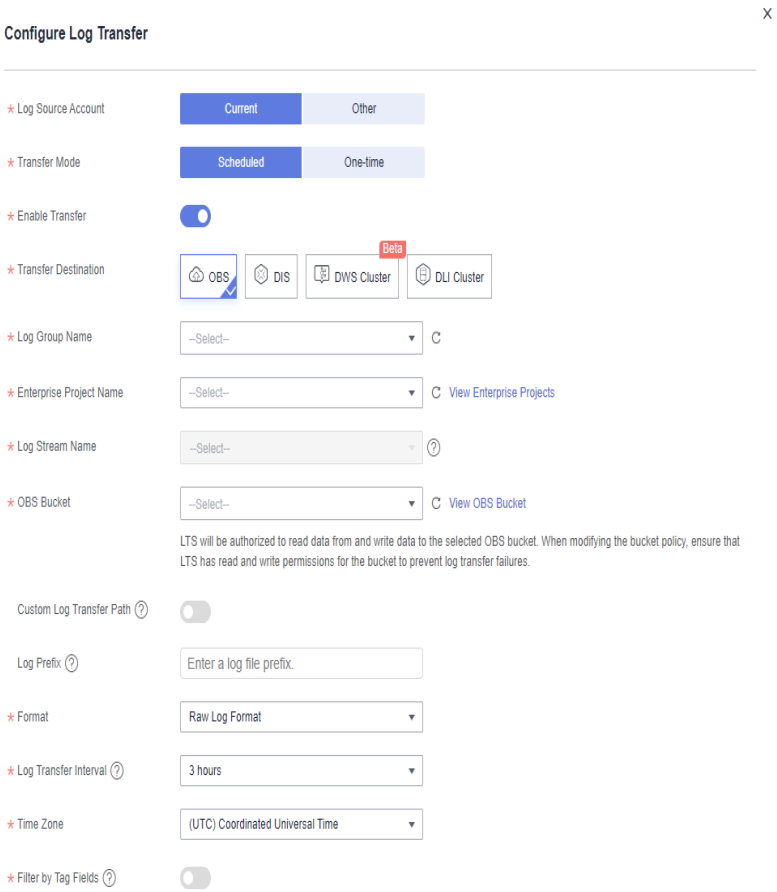
The backend server responds to the request normally.

Configuring Log Transfer

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

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

Figure 2-26 Configuring log transfer



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

2.8 Tags and Quotas

2.8.1 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.

  For detailed operations and parameters, see **Creating a Shared 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 **Networking** > **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**.

NOTE

- A maximum of 20 tags can be added to a load balancer.
- Each tag is a key-value pair, and the tag key is unique.

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 **Networking** > **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**.

NOTE

- A maximum of 20 tags can be added to a listener.
- Each tag is a key-value pair, and the tag key is unique.

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 **Networking** > **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 **Networking** > **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.

2.8.2 Quotas

What Is Quota?

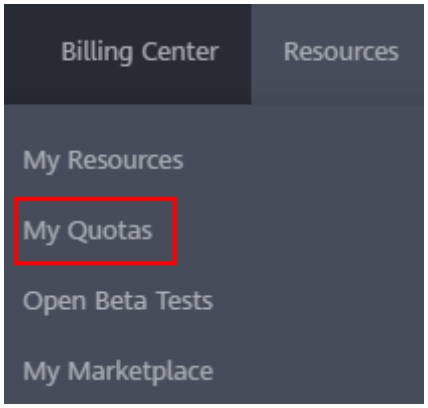
Quotas can limit the number or amount of resources available to users, such as the maximum number of ECSs or EVS disks that can be created.

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

How Do I View My Quotas?

1. Log in to the management console.
2. 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.

Figure 2-27 My Quotas



4. View the used and total quota of each type of resources on the displayed page.

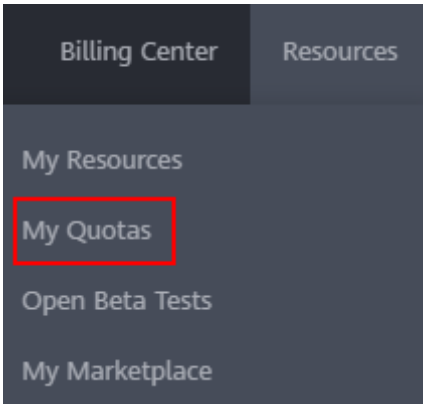
   If a quota cannot meet service requirements, apply for a higher quota.

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**.

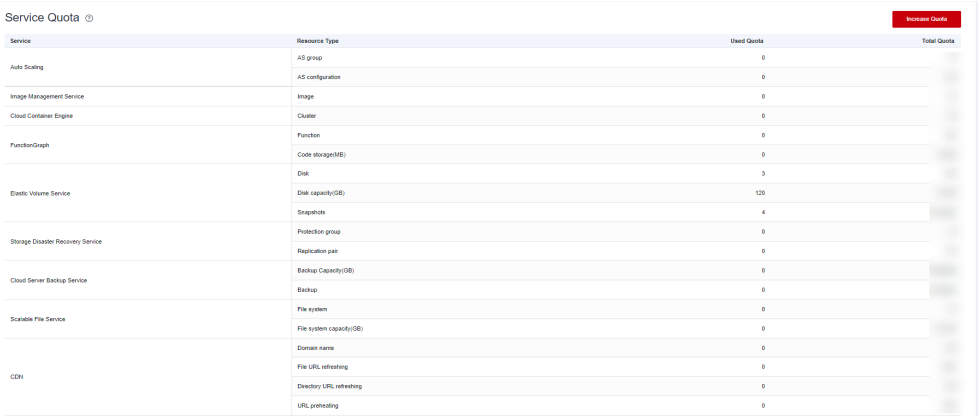
   The **Quotas** page is displayed.

   Figure 2-28 My quotas

3. Click **Increase Quota** in the upper right corner of the page.

   Figure 2-29 Increasing quota

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**.

2.9 Monitoring

2.9.1 Monitoring Metrics

Overview

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

Namespace

SYS.ELB

Metrics

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

Table 2-42 Metrics supported by each shared load balancer

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m1_cps	Concurrent Connections	Load balancing at Layer 4: total number of TCP and UDP connections from the monitored object to backend servers. Load balancing at Layer 7: total number of TCP connections from the clients to the monitored object. Unit: Count	≥0	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m2_act_conn	Active Connections	Number of TCP and UDP connections in the **ESTABLISHED** state between the monitored object and backend servers. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an` Unit: Count	≥0	Shared load balancer	1 minute
m3_inact_conn	Inactive Connections	Number of TCP connections between the monitored object and backend servers except those in the **ESTABLISHED** state. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an` Unit: Count	≥0	Shared load balancer	1 minute
m4_ncps	New Connections	Number of connections established between clients and the monitored object per second. Unit: Count/s	≥0/s	Shared load balancer	1 minute
m5_in_pps	Incoming Packets	Number of packets received by the monitored object per second. Unit: Count/s	≥0/s	Shared load balancer	1 minute
m6_out_pps	Outgoing Packets	Number of packets sent from the monitored object per second. Unit: Count/s	≥0/s	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m7_in_Bps	Inbound Rate	Traffic used for accessing the monitored object from the Internet. Unit: bytes/s	≥0 bytes/s	Shared load balancer	1 minute
m8_out_Bps	Outbound Rate	Traffic used by the monitored object to access the Internet per second. Unit: bytes/s	≥0 bytes/s	Shared load balancer	1 minute
m9_abnormal_servers	Unhealthy Servers	Number of unhealthy backend servers associated with the monitored object. Unit: Count	≥0	Shared load balancer	1 minute
ma_normal_servers	Healthy Servers	Number of healthy backend servers associated with the monitored object. Unit: Count	≥0	Shared load balancer	1 minute
m22_in_bandwidth	Inbound Bandwidth	Bandwidth used for accessing the monitored object from the Internet. Unit: bits/s	≥0 bits/s	Shared load balancer	1 minute
m23_out_bandwidth	Outbound Bandwidth	Bandwidth used by the monitored object to access the Internet. Unit: bits/s	≥0 bits/s	Shared load balancer	1 minute
m1e_server_rps	Reset Packets from Backend Servers	Number of reset packets sent from backend servers to clients. These reset packets are generated by the backend servers and then forwarded by the load balancer. This metric is available only for TCP listeners. Unit: Count/s	≥0/s	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m21_client_rps	Reset Packets from Clients	Number of reset packets sent by clients to backend servers. These reset packets are generated by clients and then forwarded by the load balancer. This metric is available only for TCP listeners. Unit: Count/s	≥0/s	Shared load balancer	1 minute
m1f_lvs_rps	Reset Packets from Load Balancers	Number of reset packets generated by the load balancer. This metric is available only for TCP listeners. Unit: Count/s	≥0/s	Shared load balancer	1 minute
mb_l7_qps	Layer-7 Query Rate	Number of requests the monitored object receives per second. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer	1 minute
mc_l7_http_2xx	Layer-7 2xx Status Codes	Number of 2xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
md_l7_http_3xx	Layer-7 3xx Status Codes	Number of 3xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer	1 minute
me_l7_http_4xx	Layer-7 4xx Status Codes	Number of 4xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer	1 minute
mf_l7_http_5xx	Layer-7 5xx Status Codes	Number of 5xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer	1 minute
m10_l7_http_other_status	Layer-7 Other Status Codes	Number of status codes returned by the load balancer and backend servers except 2xx, 3xx, 4xx, and 5xx status codes. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m11_l7_http_404	Layer-7 404 Not Found	Number of 404 Not Found status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer	1 minute
m12_l7_http_499	Layer-7 499 Client Closed Request	Number of 499 Client Closed Request status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer	1 minute
m13_l7_http_502	Layer-7 502 Bad Gateway	Number of 502 Bad Gateway status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m14_l7_rt	Average Layer-7 Response Time	Average response time of the monitored object. This metric is available only when the frontend protocol is HTTP or HTTPS. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Unit: ms **NOTE** The average response time it takes to establish a WebSocket connection may be very high. This metric cannot be used as a reference.	≥0 ms	Shared load balancer	1 minute
m15_l7_upstream_4xx	4xx Status Codes Backend	Number of 4xx status codes returned by the backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer	1 minute
m16_l7_upstream_5xx	5xx Status Codes Backend	Number of 5xx status codes returned by the backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m17_l7_upstream_rt	Average Server Response Time	Average response time of backend servers. 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. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: ms **NOTE** The average response time it takes to establish a WebSocket connection may be very high. This metric cannot be used as a reference.	≥0 ms	Shared load balancer	1 minute
m1a_l7_upstream_rt_max	Maximum Server Response Time	Maximum response time of backend servers. 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. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: ms	≥ 0ms	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m1b_l7_upstream_rt_min	Minimum Server Response Time	Minimum response time of backend servers. 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. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: ms	≥ 0ms	Shared load balancer	1 minute
m1c_l7_rt_max	Maximum Layer-7 Response Time	Maximum response time of the monitored object. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: ms	≥ 0ms	Shared load balancer	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m1d_l7_rt_min	Minimum Layer-7 Response Time	Minimum response time of the monitored object. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: ms	≥ 0ms	Shared load balancer	1 minute
m25_l7_resp_Bps	Backend Server Response Bandwidth	The bandwidth that the monitored object uses to return response to clients. Unit: bits/s **NOTE** When HTTP/2 is enabled for a listener, this metric cannot be used as a reference.	≥ 0bit/s	Shared load balancer	1 minute
m24_l7_req_Bps	Backend Server Request Bandwidth	The bandwidth that the monitored object uses to receive requests from clients. Unit: bits/s **NOTE** When HTTP/2 is enabled for a listener, this metric cannot be used as a reference.	≥ 0bit/s	Shared load balancer	1 minute

Table 2-43 Metrics supported by each listener

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m1_cps	Concurrent Connections	Load balancing at Layer 4: total number of TCP and UDP connections from the monitored object to backend servers. Load balancing at Layer 7: total number of TCP connections from the clients to the monitored object. Unit: Count	≥0	Shared load balancer - listener	1 minute
m2_act_conn	Active Connections	Number of TCP and UDP connections in the **ESTABLISHED** state between the monitored object and backend servers. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an` Unit: Count	≥0	Shared load balancer - listener	1 minute
m3_inact_conn	Inactive Connections	Number of TCP connections between the monitored object and backend servers except those in the **ESTABLISHED** state. You can run the following command to view the connections (both Windows and Linux servers): `netstat -an` Unit: Count	≥0	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m4_ncps	New Connections	Number of connections established between clients and the monitored object per second. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute
m5_in_pps	Incoming Packets	Number of packets received by the monitored object per second. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute
m6_out_pps	Outgoing Packets	Number of packets sent from the monitored object per second. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute
m7_in_Bps	Inbound Rate	Traffic used for accessing the monitored object from the Internet. Unit: bytes/s	≥0 bytes/s	Shared load balancer - listener	1 minute
m8_out_Bps	Outbound Rate	Traffic used by the monitored object to access the Internet per second. Unit: bytes/s	≥0 bytes/s	Shared load balancer - listener	1 minute
m9_abnormal_servers	Unhealthy Servers	Number of unhealthy backend servers associated with the monitored object. Unit: Count	≥0	Shared load balancer - listener	1 minute
ma_normal_servers	Healthy Servers	Number of healthy backend servers associated with the monitored object. Unit: Count	≥0	Shared load balancer - listener	1 minute
m22_in_bandwidth	Inbound Bandwidth	Bandwidth used for accessing the monitored object from the Internet. Unit: bits/s	≥0 bits/s	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m23_out_bandwidth	Outbound Bandwidth	Bandwidth used by the monitored object to access the Internet. Unit: bits/s	≥0 bits/s	Shared load balancer - listener	1 minute
m1e_server_rps	Reset Packets from Backend Servers	Number of reset packets sent from backend servers to clients. These reset packets are generated by the backend servers and then forwarded by the load balancer. This metric is available only for TCP listeners. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute
m21_client_rps	Reset Packets from Clients	Number of reset packets sent by clients to backend servers. These reset packets are generated by clients and then forwarded by the load balancer. This metric is available only for TCP listeners. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute
m1f_lvs_rps	Reset Packets from Load Balancers	Number of reset packets generated by the load balancer. This metric is available only for TCP listeners. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
mb_l7_qps	Layer-7 Query Rate	Number of requests the monitored object receives per second. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute
mc_l7_http_2xx	Layer-7 2xx Status Codes	Number of 2xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute
md_l7_http_3xx	Layer-7 3xx Status Codes	Number of 3xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute
me_l7_http_4xx	Layer-7 4xx Status Codes	Number of 4xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
mf_l7_http_5xx	Layer-7 5xx Status Codes	Number of 5xx status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute
m10_l7_http_other_status	Layer-7 Other Status Codes	Number of status codes returned by the load balancer and backend servers except 2xx, 3xx, 4xx, and 5xx status codes. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute
m11_l7_http_404	Layer-7 404 Not Found	Number of 404 Not Found status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute
m12_l7_http_499	Layer-7 499 Client Closed Request	Number of 499 Client Closed Request status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m13_l7_http_502	Layer-7 502 Bad Gateway	Number of 502 Bad Gateway status codes returned by the load balancer and backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute
m14_l7_rt	Average Layer-7 Response Time	Average response time of the monitored object. This metric is available only when the frontend protocol is HTTP or HTTPS. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Unit: ms **NOTE** The average response time it takes to establish a WebSocket connection may be very high. This metric cannot be used as a reference.	≥0 ms	Shared load balancer - listener	1 minute
m15_l7_upstream_4xx	4xx Status Codes Backend	Number of 4xx status codes returned by the backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m16_l7_upstream_5xx	5xx Status Codes Backend	Number of 5xx status codes returned by the backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer - listener	1 minute
m17_l7_upstream_rt	Average Server Response Time	Average response time of backend servers. This metric is available only when the frontend protocol is HTTP or 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. Unit: ms **NOTE** The average response time it takes to establish a WebSocket connection may be very high. This metric cannot be used as a reference.	≥0 ms	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m1a_l7_upstream_rt_max	Maximum Server Response Time	Maximum response time of backend servers. This metric is available only when the frontend protocol is HTTP or 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. Unit: ms	≥0 ms	Shared load balancer - listener	1 minute
m1b_l7_upstream_rt_min	Minimum Server Response Time	Minimum response time of backend servers. This metric is available only when the frontend protocol is HTTP or 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. Unit: ms	≥0 ms	Shared load balancer - listener	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m1c_l7_rt_max	Maximum Layer-7 Response Time	Maximum response time of the monitored object. This metric is available only when the frontend protocol is HTTP or HTTPS. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Unit: ms	≥0 ms	Shared load balancer - listener	1 minute
m1d_l7_rt_min	Minimum Layer-7 Response Time	Minimum response time of the monitored object. This metric is available only when the frontend protocol is HTTP or HTTPS. The response time starts when the monitored object receives requests from the clients and ends when it returns all responses to the clients. Unit: ms	≥0 ms	Shared load balancer - listener	1 minute

Table 2-44 Metrics supported by each backend server group

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m9_abnormal_servers	Unhealthy Servers	Number of unhealthy backend servers associated with the monitored object. Unit: Count	≥0	Shared load balancer - backend server group	1 minute
ma_normal_servers	Healthy Servers	Number of healthy backend servers associated with the monitored object. Unit: Count	≥0	Shared load balancer - backend server group	1 minute
m17_l7_upstream_rt	Average Server Response Time	Average response time of backend servers. This metric is available only when the frontend protocol is HTTP or 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. Unit: ms **NOTE** The average response time it takes to establish a WebSocket connection may be very high. This metric cannot be used as a reference.	≥ 0ms	Shared load balancer - backend server group	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m1a_l7_upstream_rt_max	Maximum Server Response Time	Maximum response time of backend servers. This metric is available only when the frontend protocol is HTTP or 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. Unit: ms	≥ 0ms	Shared load balancer - backend server group	1 minute
m1b_l7_upstream_rt_min	Minimum Server Response Time	Minimum response time of backend servers. This metric is available only when the frontend protocol is HTTP or 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. Unit: ms	≥ 0ms	Shared load balancer - backend server group	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
mb_l7_qps	Layer-7 Query Rate	Number of requests the monitored object receives per second. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer - backend server group	1 minute
m18_l7_upstream_2xx	2xx Status Codes Backend	Number of 2xx status codes returned by the backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer - backend server group	1 minute
m19_l7_upstream_3xx	3xx Status Codes Backend	Number of 3xx status codes returned by the backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer - backend server group	1 minute
m15_l7_upstream_4xx	4xx Status Codes Backend	Number of 4xx status codes returned by the backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer - backend server group	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
m16_l7_upstream_5xx	5xx Status Codes Backend	Number of 5xx status codes returned by the backend servers. This metric is available only when the frontend protocol is HTTP or HTTPS. Unit: Count/s	≥0/s	Shared load balancer - backend server group	1 minute
m25_l7_resp_Bps	Backend Server Response Bandwidth	The bandwidth that the monitored object uses to return response to clients. Unit: bits/s **NOTE** When HTTP/2 is enabled for a listener, this metric cannot be used as a reference.	≥ 0bit/s	Shared load balancer - backend server group	1 minute
m24_l7_req_Bps	Backend Server Request Bandwidth	The bandwidth that the monitored object uses to receive requests from clients. Unit: bits/s **NOTE** When HTTP/2 is enabled for a listener, this metric cannot be used as a reference.	≥ 0bit/s	Shared load balancer - backend server group	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
lbaas_pool_id	ID of a backend server group

2.9.2 Setting an Alarm Rule

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

2.9.2.1 Creating an Alarm Rule

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

   Configure parameters based on **Table 2-45**.

   Table 2-45 Parameters for creating an alarm rule

Parameter	Setting
Resource Type	Select **Elastic Load Balance**.
Dimension	Select from the following options: • **Elastic Load Balancers** • **Elastic Load Balancers - Backend Server Group** **NOTE** For a shared load balancer, **Elastic Load Balancers - Listeners** cannot be selected as a dimension.
Other Parameters	Set them as required.

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

   NOTE

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

2.9.2.2 Modifying an Alarm Rule

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

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

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

      NOTE

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

2.9.3 Viewing Metrics

Scenarios

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

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

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

Prerequisites

- The load balancer is running properly.

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

  NOTE

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

- You have interconnected ELB with Cloud Eye and configured an alarm rule for the load balancer on the Cloud Eye console.

  Without alarm rules, there is no monitoring data. For details, see **Setting an Alarm Rule**.

- If an IAM user wants to view the ELB monitoring data on the Cloud Eye console, the IAM user must be granted the **ELB Administrator** permission. Otherwise, the IAM user cannot view all monitoring data.

Viewing Monitoring Metrics on the 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 **Networking** > **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. View the metrics of each load balancer and listener.
   a. Load balancer: Click **Monitoring** tab and select **Load balancer** for **Dimension**.
   b. Listener (two ways):
      i. Click **Monitoring** tab, select **Load listener** for **Dimension**, locate the target listener, and view the monitoring metrics.
      ii. Click the name of the target listener, 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** > **Elastic Load Balance**.
5. On the **Cloud Service Monitoring** page, click the name of the load balancer. Alternatively, locate the load balancer and click **View Metric** in the **Operation** column.
6. Select the time period during which you want to view metrics. You can select a system-defined time period (for example, last 1 hour) or specify a time period.
7. Click **Select Metric** in the upper right corner and select the metrics to be viewed.

NOTE

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

2.10 Auditing

2.10.1 Key Operations Recorded by CTS

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

Table 2-46 lists the operations recorded by CTS.

Table 2-46 ELB operations recorded by CTS

Action	Resource Type	Trace
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
Adding a backend server	member	createMember
Modifying a backend server	member	updateMember

Action	Resource Type	Trace
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

2.10.2 Viewing Traces

Scenarios

CTS records the operations performed on ELB and allows you to view the operation records of the last seven days on the CTS console. To query these records, perform the following operations.

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**

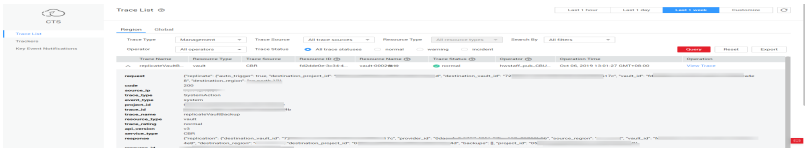
     Select a filter from the drop-down list.

     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 2-30 Expanding trace details



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

Figure 2-31 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": "
    "tracker_name": "system",
    "time": "1569321775225",
    "resource_name": "9646e73b-338c-4d27-a17c-219be532812c",
    "record_time": "1569321775903",
    "user": {
        "domain": {
            "name": "
            "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 Dedicated Load Balancer**
- **Creating a Backend Server Group**
- **Adding a TCP Listener**
- **Enabling or Disabling 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. 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** > **Elastic Load Balance**.
4. In the navigation pane on the left, click **Self-service Troubleshooting**.
5. On the **Elastic Load Balance** tab, click **Unhealthy backend servers**.
6. Select the load balancer that has unhealthy backend servers.
7. Select the unhealthy backend server you want to troubleshoot.
8. 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 IP address configured for the inbound rule	The inbound rules of the security group do not allow traffic from the source IP address to the backend server.	

Health Check Category	Health Check Item	Reason	Suggestion
	The port configured for the inbound rule	The inbound rules of the security group do not allow traffic over the health check port.	
	The protocol configured for the outbound rule	The outbound rules of the security group do not allow traffic over the health check protocol.	
	The source IP address configured for the outbound rule	The outbound rules of the security group do not allow traffic from the source IP address to the backend server.	
	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 IP address configured for the inbound rule	The inbound rules of the network ACL do not allow traffic from the source 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.	

Health Check Category	Health Check Item	Reason	Suggestion
	The destination address configured for the inbound rule	The inbound rules of the network ACL do not allow traffic to the destination address.	
	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 source IP address configured for the outbound rule	The outbound rules of the network ACL do not allow traffic from the source 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.	

Health Check Category	Health Check Item	Reason	Suggestion
	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 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 **Enabling or Disabling Health Check**.

NOTE

- If all the check items are reported as normal, perform further checks as guided by **How Do I Troubleshoot an Unhealthy Backend Server?**
- If the troubleshooting fails, click **Troubleshoot Again** or perform further checks as guided by **How Do I Troubleshoot an Unhealthy Backend Server?**

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 (Dedicated Load Balancers)** • **Billing (Shared Load Balancers)**
Billing modes	

Scenario	Reference
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**
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's source 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

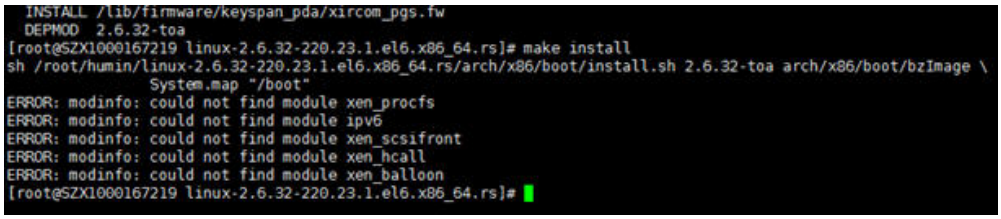
```
INSTALL /lib/firmware/kaweth/trigger_code_fix.bin
INSTALL /lib/firmware/ti_3410.fw
INSTALL /lib/firmware/ti_5052.fw
INSTALL /lib/firmware/mts_cdma.fw
INSTALL /lib/firmware/mts_gsm.fw
INSTALL /lib/firmware/mts_edge.fw
INSTALL /lib/firmware/edgeport/boot.fw
INSTALL /lib/firmware/edgeport/boot2.fw
INSTALL /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

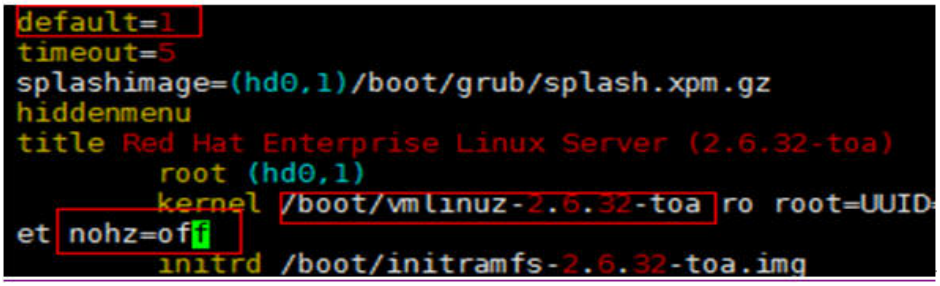
   

7. Open the **/boot/grub/grub.conf** file and configure the kernel to start up when the system starts.

   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

   c. Save the modification and exit. Restart the OS.

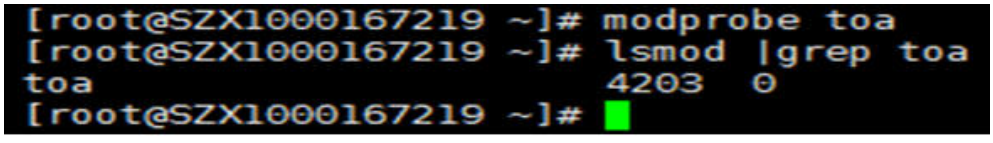
      During the restart, the system will load the **vmlinuz-2.6.32-toa** kernel.

8. After the restart, run the following command to load the module:

   modprobe toa

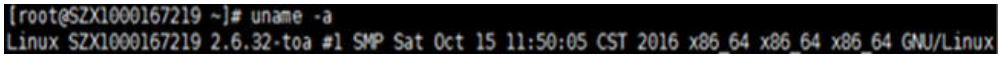
   Add the **modprobe toa** command to both the startup script and the system scheduled monitoring script.

   Figure 4-4 Adding the **modprobe toa** command

   After the module is loaded, query the kernel information.

   Figure 4-5 Querying the kernel

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.

5 Change History

Released On	Description
2024-06-20	This issue is the thirtieth official release. Added the following sections: • **User Guide for Dedicated Load Balancers** • **User Guide for Shared Load Balancers** • Added the **Listen to All Ports** and **Forward to Same Port** options. • Supported gRPC as the backend protocol.
2024-03-15	This issue is the twenty-ninth official release. Added the descriptions about tag policies in **Creating a Dedicated Load Balancer** and **Tag**.
2024-01-05	This issue is the twenty-eighth official release. Added **Self-service Troubleshooting**.
2023-09-07	This issue is the twenty-seventh official release. Updated the following sections: • **Creating a Dedicated Load Balancer** • **Creating a Shared Load Balancer** • **IP Address Group**
2023-07-14	This issue is the twenty-sixth official release. • Updated **Transfer Client IP Address**. • Added section "Transfer Client IP Address (Shared Load Balancers)".
2023-05-18	This issue is the twenty-fifth official release. Added the following sections: • **Backend Server Group** • **Backend Server** • **Backend Server**

Released On	Description
2023-02-03	This is the twenty-fourth official release. Added section "Enabling Guaranteed Performance for a Shared Load Balancer".
2022-12-30	This issue is the twenty-third official release. Modified the following sections: *Changing the Specifications of a Dedicated Load Balancer*
2022-09-07	This issue is the twenty-second official release. Modified the following section: Added restrictions on ping verification for load balancers in sections *Creating a Dedicated Load Balancer*, *Creating a Shared Load Balancer*, and **What Is Access Control?**
2022-06-30	This is the twenty-first official release. Added the following section: • **Adding a UDP Listener**, UDP listeners do not support fragmentation. • **Adding a UDP Listener (with a QUIC Backend Server Group Associated)**, UDP listeners using QUIC protocol do not support fragmentation.
2022-05-30	This issue is the twentieth official release. Added the following section: **Routing Traffic to Backend Servers in the Same VPC as the Load Balancer**
2022-03-30	This issue is the nineteenth official release. Added the following section: **Using Advanced Forwarding for Application Iteration**
2022-03-18	This issue is the eighteenth official release. Deleted the FAQ "What Is the Maximum Size of Files that Can Be Transferred Using HTTP or HTTPS?"
2022-03-07	This issue is the seventeenth official release, which incorporates the following changes: Added **Does ELB Support IPv6 Networks?**
2022-02-14	This issue is the sixteenth official release, which incorporates the following changes: Updated **Viewing Metrics**

Released On	Description
2022-01-10	This issue is the fifteenth official release. Added the following sections: **Routing Traffic Across Cloud Servers and On-Premises Servers** Section "Transfer Client IP Address".
2022-01-04	This issue is the fourteenth official release. Added the following sections: • **Forwarding Policy** • **Advanced Forwarding**
2021-12-29	This issue is the thirteenth official release. Added the following sections: • **HTTP/2** • **Certificate Overview** • **Protection for Mission-Critical Operations**
2021-12-14	This issue is the twelfth official release, which incorporates the following changes: Added **What Functions Will Become Unavailable If a Dedicated Load Balancer Is Frozen?**
2021-12-09	This issue is the eleventh official release, which incorporates the following changes: Added the diagram of timeout durations at layer 4.
2021-10-28	This issue is the tenth official release, which incorporates the following changes: Added **Permissions Management**.
2021-10-21	This issue is the ninth official release, which incorporates the following changes: • **Can Both the Listener and Backend Server Group Use HTTPS?** • **Do Shared Load Balancers Have Specifications?**

Released On	Description
2021-09-02	This issue is the eighth official release, which incorporates the following changes: • Optimized **Differences Between Dedicated and Shared Load Balancers**. • Added the following sections: **Can Backend Servers Access the Ports of a Load Balancer?** **Can I Bind a Public IP Address Purchased from a Third-Party Cloud Provider to My Load Balancer?** **Do I Need to Configure Bandwidth for My Load Balancers?** **Can I Bind Multiple EIPs to a Load Balancer?** **Why Does a Dedicated Load Balancer Need Multiple IP Addresses?** **Why Are Requests from the Same IP Address Routed to Different Backend Servers When the Load Balancing Algorithm Is Source IP Hash?** **Can Backend Servers Access the Internet Using the EIP of the Associated Load Balancer?** **Why Can't I Select the Target Backend Server Group When Adding or Modifying a Listener?** **Why Must the Subnet Where the Load Balancer Resides Have at Least 16 Available IP Addresses When I Enable the IP as a Backend Function?** **Why Is a Forwarding Policy in the Faulty State?** **How Can I Add a Forwarding Policy to a Listener?** **What Are Status Codes for Normal Health Checks?**
2021-07-16	This issue is the seventh official release, which incorporates the following changes: Changed **Management & Deployment** to **Management & Governance** and **Computing** to **Compute** based on the latest console product catalog.
2021-06-18	This issue is the sixth official release, which incorporates the following changes: Deleted all descriptions and operations related to classic load balancers.

Released On	Description
2021-02-28	This issue is the fifth official release, which incorporates the following changes: ● Optimized the meaning of **Concurrent Connections** in section "Monitoring Metrics". ● Added section "Configuring Security Group Rules for Backend Servers (Dedicated Load Balancers)". ● Added information about dedicated load balancers in FAQ "How Do I Troubleshoot an Unhealthy Backend Server?"
2020-05-30	This issue is the fourth official release, which incorporates the following changes: Changed the name of enhanced load balancers to shared load balancers.
2019-03-30	This issue is the third official release, which incorporates the following changes: Added the content related to enterprise project management.
2018-12-30	This issue is the second official release, which incorporates the following changes: Modified the content and changed some figures in the document based on the latest console.
2018-10-31	This issue is the first official release.

