



Elastic Load Balance

User Guide

Date 2020-07-30

Contents

1 Service Overview.....	1
1.1 What Is Elastic Load Balance?.....	1
1.2 Product Advantages.....	2
1.3 Application Scenarios.....	3
1.4 How ELB Works.....	5
1.5 Network Type.....	6
1.6 ELB and Other Services.....	8
1.7 Product Concepts.....	8
1.7.1 Basic Concepts.....	8
1.7.2 Region and AZ.....	10
2 Load Balancer.....	12
2.1 Network Type.....	12
2.2 Preparing for Creation.....	13
2.3 Creating a Load Balancer.....	15
2.4 Changing Load Balancer Settings.....	17
2.5 Binding or Unbinding an EIP.....	18
2.6 Deleting a Load Balancer.....	18
3 Listener.....	20
3.1 Overview.....	20
3.2 Protocols and Ports.....	21
3.3 Adding a Listener.....	22
3.4 Load Balancing Algorithms.....	27
3.5 Sticky Session.....	28
3.6 Access Control.....	29
3.7 Modifying or Deleting a Listener.....	31
3.8 Advanced Settings for HTTP or HTTPS Listeners.....	31
3.8.1 Forwarding Policy.....	31
3.8.2 Mutual Authentication.....	35
3.8.3 HTTP/2.....	40
3.8.4 HTTP Redirection to HTTPS.....	41
3.8.5 SNI.....	43
4 Backend Server.....	44

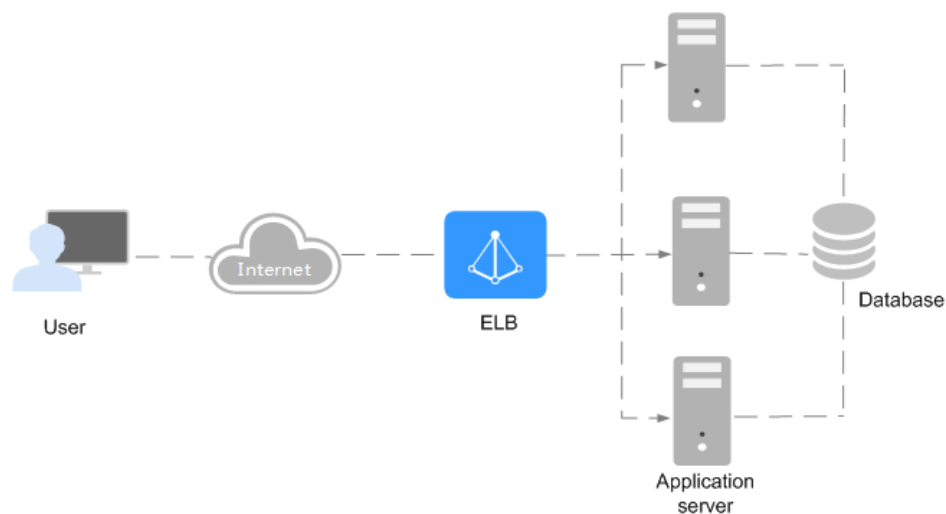
4.1 Overview.....	44
4.2 Configuring Security Group Rules.....	45
4.3 Adding Backend Servers to or Removing Backend Servers.....	47
5 Health Check.....	53
5.1 Configuring a Health Check.....	53
5.2 Disabling the Health Check Function.....	55
6 Certificate.....	56
6.1 Certificate and Private Key Format.....	56
6.2 Converting Certificate Formats.....	57
6.3 Creating a Certificate.....	58
7 Access Logging.....	61
8 Monitoring.....	66
8.1 Monitoring Metrics.....	66
8.2 Setting an Alarm Rule.....	71
8.2.1 Adding an Alarm Rule.....	71
8.2.2 Modifying an Alarm Rule.....	71
8.3 Viewing Metrics.....	72
9 Auditing.....	73
9.1 Key Operations Recorded by CTS.....	73
9.2 Viewing Traces.....	74
10 FAQs.....	76
10.1 Questions Summary.....	76
10.2 ELB Usage.....	76
10.2.1 Service Abnormality.....	76
10.2.1.1 How Can I Check ELB Unavailability or Routing Interruption?.....	76
10.2.2 ELB Functionality.....	77
10.2.2.1 Can ELB Be Used Separately?.....	77
10.2.2.2 Is the EIP Assigned to a Load Balancer Exclusive?.....	77
10.2.2.3 How Many Load Balancers and Listeners Can I Have?.....	77
10.2.2.4 Can I Adjust the Number of Backend Servers When a Load Balancer is Running?.....	77
10.2.2.5 Can Backend Servers Run Different OSs?.....	77
10.2.3 Performance and Workloads.....	78
10.2.3.1 How Can I Check Traffic Inconsistency?.....	78
10.2.3.2 How Can I Check that Traffic Is Unbalanced?.....	78
10.2.3.3 How Can I Check High Access Delay of a Load Balancer?.....	78
10.2.3.4 What Should I Do If a Load Balancer's Performance Fails the Stress Test?.....	78
10.3 Load Balancer.....	79
10.3.1 How Does ELB Distribute Traffic?.....	79
10.3.2 How Can I Configure a Public or Private Network Load Balancer?.....	79
10.4 Listener.....	80

10.4.1 What Are the Relationships Between Load Balancing Algorithms and Sticky Session Types?.....	80
10.4.2 How Can ELB Support Multiple Certificates?.....	80
10.4.3 How Can I Use WebSocket?.....	80
10.5 Backend Server.....	80
10.5.1 Why Is the Interval at Which Backend Servers Receive Health Check Packets Is Different from the Configured Health Check Interval?.....	81
10.5.2 Can Backend Servers Access the Public Network After They Are Associated with a Load Balancer?	81
10.5.3 How Can I Check the Network Conditions of a Backend Server?.....	81
10.5.4 How Can I Check the Network Configuration of a Backend Server?.....	81
10.5.5 How Can I Check the Status of a Backend Server?.....	82
10.5.6 When Is a Backend Server Considered Healthy?.....	82
10.6 Health Check.....	82
10.6.1 What Should I Do If a Backend Server Is Unhealthy?.....	83
10.6.2 What Are the Precautions of Using UDP for Health Checks?.....	88
10.6.3 Why Does ELB Frequently Send Requests to Backend Servers During Health Checks?.....	90
10.7 Obtaining Source IP Addresses.....	90
10.7.1 How Can I Obtain the IP Address of a Client?.....	90
10.8 HTTP/HTTPS Listeners.....	97
10.8.1 Why Is the Security Warning Still Displayed After a Certificate Is Configured?.....	97
10.9 Sticky Session.....	97
10.9.1 What Should I Do If Sticky Sessions Fail to Take Effect?.....	97
10.9.2 What Types of Sticky Sessions Does ELB Support?.....	98
11 Appendix.....	99
11.1 Configuring the TOA Plug-in.....	99
12 Change History.....	106

1 Service Overview

1.1 What Is Elastic Load Balance?

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



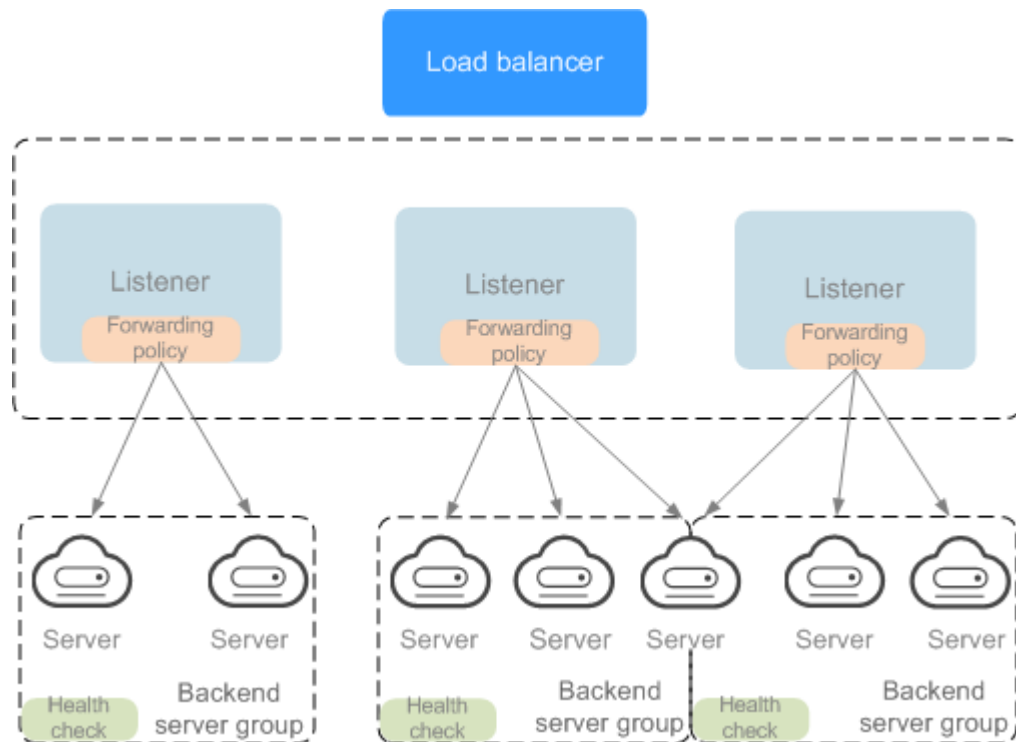
Components

ELB consists of the following components:

- Load balancer: distributes incoming traffic across backend servers in one or more availability zones (AZs).
- Listener: uses the protocol and port you specify to check for requests from clients and route the requests to associated backend servers based on the listening rules you define. You can add one or more listeners to a load balancer.
- Backend server group: routes requests from the load balancer to one or more backend servers. At least one backend server must be added to a backend server group.

You can set a weight for each backend server based on their performance. You can also configure health checks for a backend server group to check the health of each backend server. When the health check result of a backend server is **Unhealthy**, the load balancer automatically stops routing new requests to this server until it recovers.

Figure 1-1 ELB components



Access to ELB

You can use either of the following methods to access ELB:

- Management console
A graphical user interface has been provided for you. To access the service, log in to the management console and choose **Network > Elastic Load Balance** on the homepage.
- APIs
You can call APIs to access ELB. For details, see the *Elastic Load Balance API Reference*.

1.2 Product Advantages

ELB has the following advantages:

- High performance
ELB can establish hundreds of millions of concurrent connections to handle huge numbers of concurrent requests.
- High availability

ELB is deployed in cluster mode and ensures that your services are uninterrupted. If your servers in one AZ are unhealthy, ELB automatically routes traffic to healthy servers in other AZs.

- Flexible scalability
ELB makes sure that your applications always have enough capacity for varying levels of workloads. It works with Auto Scaling to flexibly adjust the number of backend servers and intelligently distribute incoming traffic across servers.
- Easy to use
A diverse set of protocols and algorithms enable you to configure traffic routing policies to suit your needs while keeping deployments simple.

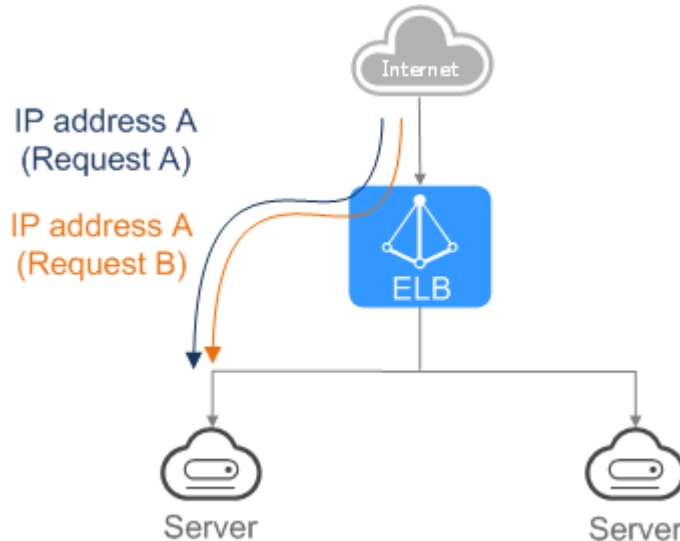
1.3 Application Scenarios

Heavy-Traffic Applications

For an application with heavy traffic, such as a large portal or mobile app store, ELB evenly distributes incoming traffic to multiple backend servers, balancing the load while ensuring steady performance.

Sticky sessions ensure that requests from one client are always forwarded to the same backend server, improving access efficiency.

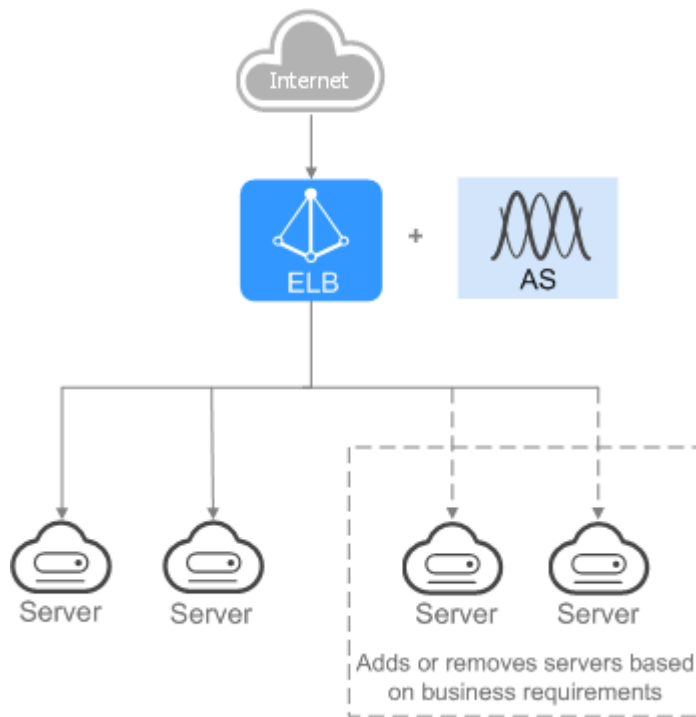
Figure 1-2 Session stickiness



Applications with Predictable Peaks and Troughs in Traffic

For an application that have predictable peaks and troughs in traffic volumes, ELB works with AS to allow backend servers to be added or removed to keep up with changing demands, improving resource utilization. One example is flash sale, which usually lasts only a few days or even several hours and during which demand on your applications increases rapidly in a short period. By combining ELB with AS, you can always run desired number of backend servers that matches to the load of your application.

Figure 1-3 Flexible scalability

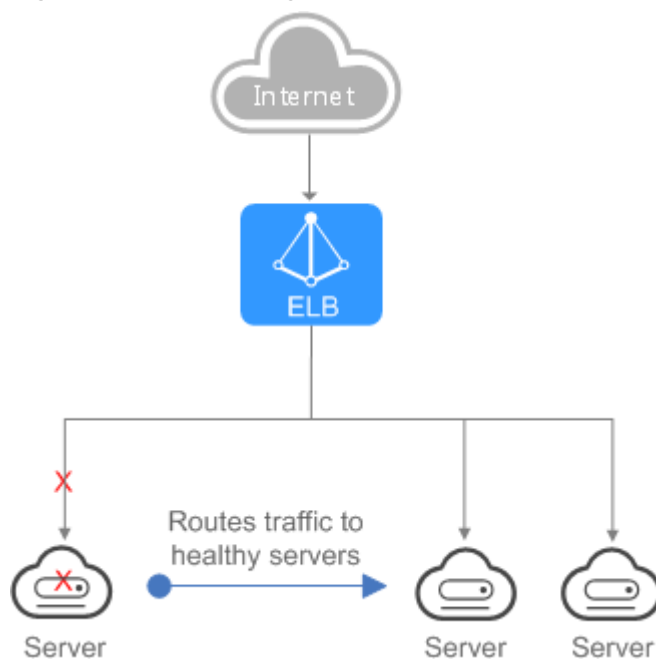


Zero SPOFs

ELB routinely performs health checks on backend servers to monitor their healthy state. If a backend server is detected unhealthy, ELB will not route requests to this server until it recovers.

This makes ELB a good choice for services that require high reliability, such as official websites and web services.

Figure 1-4 Eliminating SPOFs

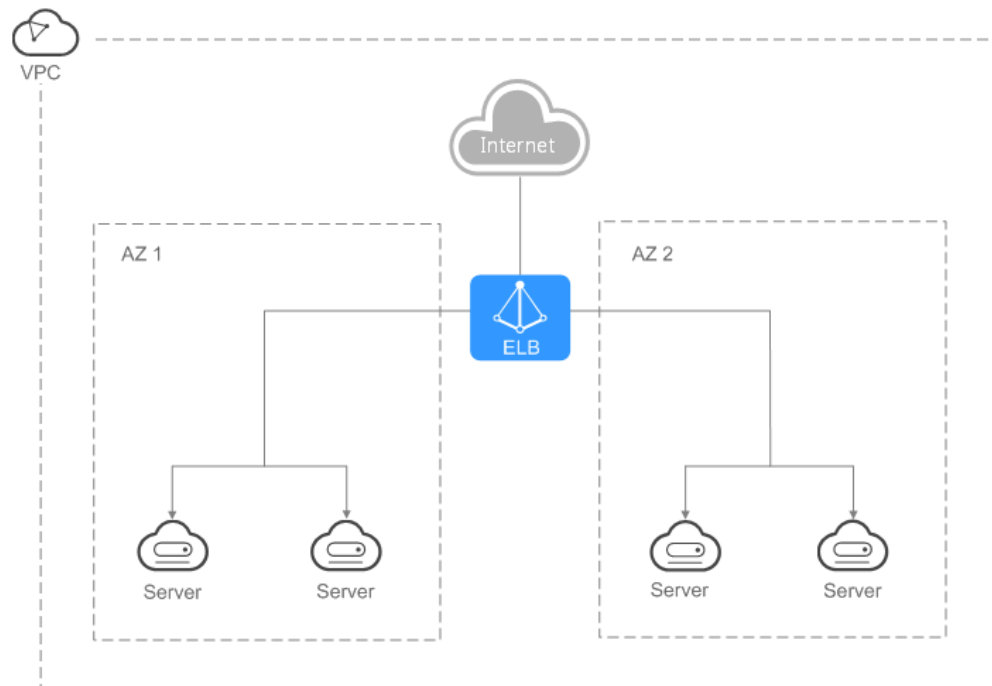


Cross-AZ Load Balancing

ELB can distribute traffic across AZs. When an AZ becomes faulty, ELB distributes traffic to backend servers in other AZs.

Banking, policing, and large application systems can use ELB to ensure high availability.

Figure 1-5 Traffic distribution to servers in one or more AZs



1.4 How ELB Works

To balance the load of your applications, you need to create a load balancer to receive requests from clients and route the requests to backend servers in one or more AZs. After the load balancer is created, you must add a listener to it and associate at least one backend server with it. The load balancing algorithm you select when you add the listener determines how requests are distributed.

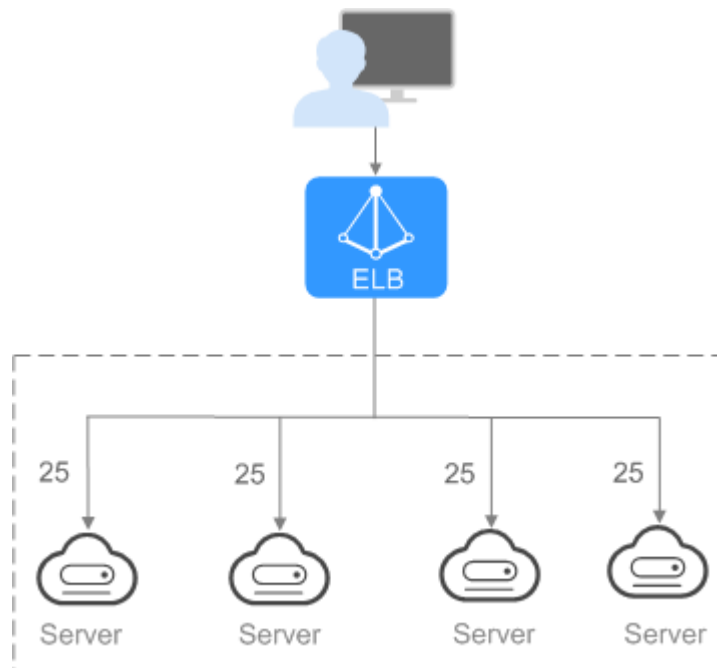
ELB supports the following load balancing algorithms:

- **Weighted round robin:** Requests are distributed across 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. This algorithm is often used for short connections, such as HTTP connections.
- **Weighted least connections:** This algorithm is designed based on the least connections algorithm that uses the number of active connections to each backend server to make its load balancing decision. In addition to the number of connections, each server is assigned a weight based on its capacity. Requests are routed to the server with the lowest connections-to-weight ratio. This algorithm is often used for persistent connections, such as database connections.

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

The following figure shows an example of how requests are distributed using the weighted round robin algorithm. Four backend servers have the same weight, and each server receives the same proportion of requests.

Figure 1-6 Traffic distribution using the weighted round robin algorithm

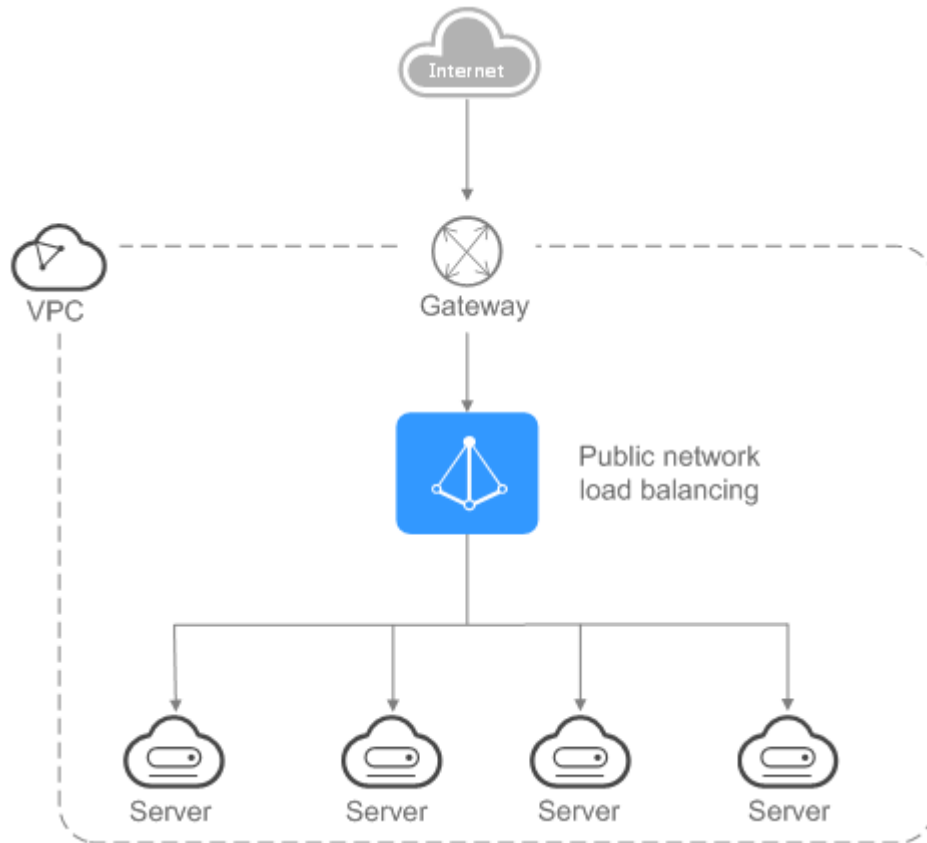


1.5 Network Type

Public Network Load Balancer

A public network load balancer provides load balancing services through a public IP address and routes requests from clients to servers over the Internet.

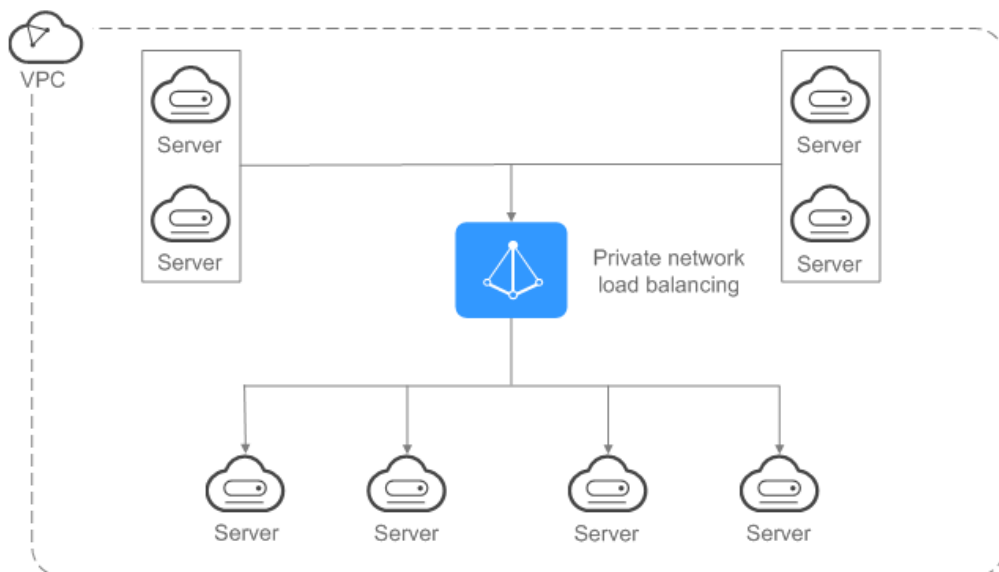
Figure 1-7 Public network load balancer



Private Network Load Balancer

A private network load balancer uses a private IP address to route requests from clients across servers in the same VPC.

Figure 1-8 Private network load balancer



1.6 ELB and Other Services

- Virtual Private Cloud (VPC)
Provides IP addresses and bandwidth for load balancers.
- Auto Scaling (AS)
Works with ELB to automatically scale the number of backend servers for better traffic distribution.
- Identity and Access Management (IAM)
Provides authentication for ELB.
- Cloud Trace Service (CTS)
Records the operations performed on ELB resources.
- Cloud Eye

1.7 Product Concepts

1.7.1 Basic Concepts

Table 1-1 Some concepts about ELB

Term	Definition
Load balancer	A load balancer distributes incoming traffic across backend servers.
Listener	A listener listens on requests from clients and routes the requests to backend servers based on the settings that you configure when adding the listener.
Backend server	A backend server is cloud server added to a backend server group associated with a load balancer. When adding a listener to a load balancer, you create or select a backend server group to receive requests from the load balancer using the port and protocol you specify for the backend server group and the load balancing algorithm you select.
Backend server group	A backend server group is a collection of cloud servers that have same features. When you add a listener, you select a load balancing algorithm and create or select a backend server group. When the listener settings are met, traffic is routed to the corresponding backend server group.

Term	Definition
Health check	ELB periodically sends requests to associated backend servers to check their health results. This process is called health check, through which the ELB system decides whether backend servers are able to process requests. If a backend server is detected unhealthy, the load balancer stops routing requests to it, ensuring service reliability. After the backend server recovers, the load balancer resumes routing requests to it.
Redirect	HTTPS is an extension of HTTP. HTTPS encrypts data between a web server and a browser. Redirection allows requests to be redirected from HTTP to HTTPS.
Sticky session	Sticky sessions are a mechanism that ensures that requests from a client always get routed to the same server before a session elapses.
WebSocket	WebSocket is a new HTML5 protocol that provides full-duplex communication between the browser and the server. WebSocket saves server resources and bandwidth, and enables real-time communication. Both WebSocket and HTTP depend on TCP to transmit data. A handshake connection is required between the browser and server, so that they can communicate with each other only after the connection is established. However, as a bidirectional communication protocol, WebSocket is different from HTTP. After the handshake succeeds, both the server and browser (or client agent) can actively send or receive data to or from each other, which is similar to Socket.
SNI	If an application provides multiple domain names and each domain name uses a different certificate, you can enable SNI when adding an HTTPS listener. SNI is an extension to TLS. It allows a server to present multiple certificates on the same IP address and TCP port number and hence allows multiple secure (HTTPS) websites (or any other service over TLS) to be served by the same IP address without requiring all those sites to use the same certificate. Before SNI, one server can use only one certificate. SNI allows the client to submit the domain name information while sending an SSL handshake request. Once receiving the request, the load balancer queries the right certificate based on the domain name and returns it to the client. If no certificate is found, the load balancer will return a default certificate.
Persistent connection	A persistent connection allows multiple data packets to be sent continuously over a TCP connection. If no data packet is sent during the connection, the client and server send link detection packets to each other to maintain the connection.
Short connection	A short connection is a connection established when data is exchanged between the client and server and immediately closed after the data is sent.

Term	Definition
Concurrent connection	Concurrent connections are total TCP connections initiated by clients and routed to backend servers by a load balancer per second.

1.7.2 Region and AZ

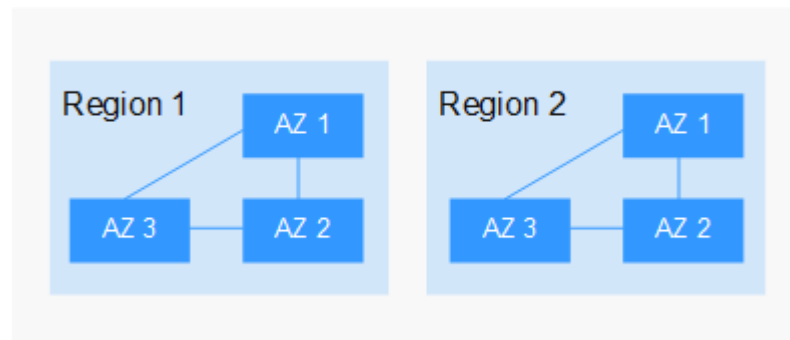
Concept

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

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

Figure 1-9 shows the relationship between regions and AZs.

Figure 1-9 Regions and AZs



Selecting a Region

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

Selecting an AZ

When deploying resources, consider your applications' requirements on disaster recovery (DR) and network latency.

- For high DR capability, deploy resources in different AZs within the same region.
- For low network latency, deploy resources in the same AZ.

Regions and Endpoints

Before you use an API to call resources, specify its region and endpoint. For more details, see [Regions and Endpoints](#).

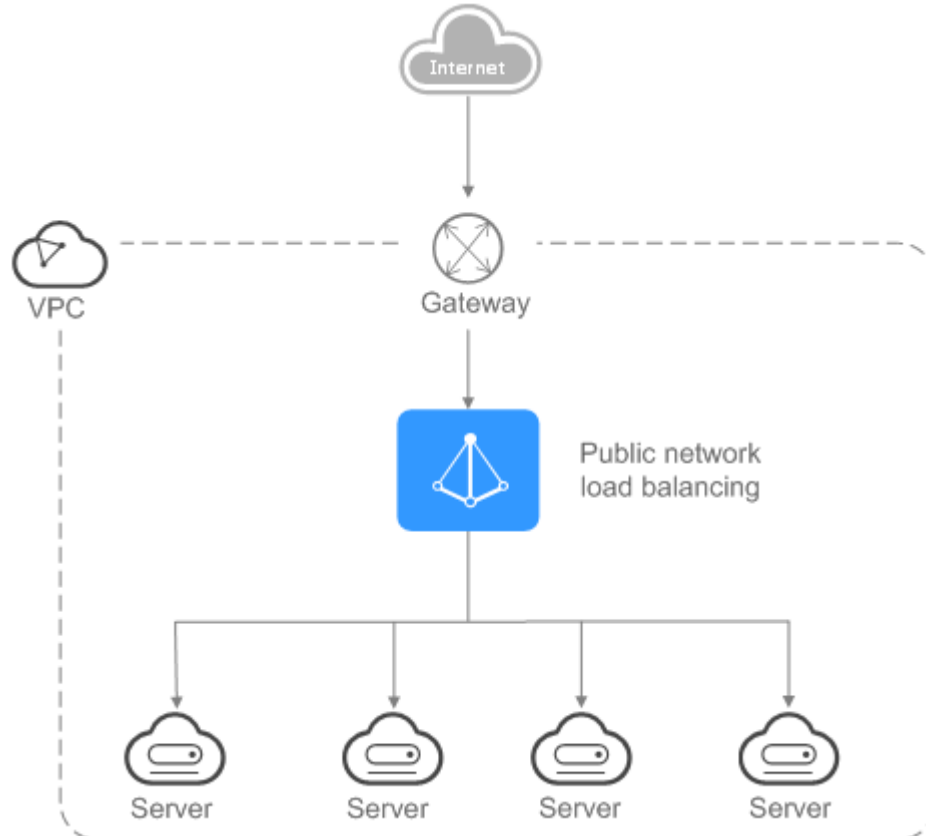
2 Load Balancer

2.1 Network Type

Public Network Load Balancer

A public network load balancer provides load balancing services through a public IP address and routes requests from clients to servers over the Internet.

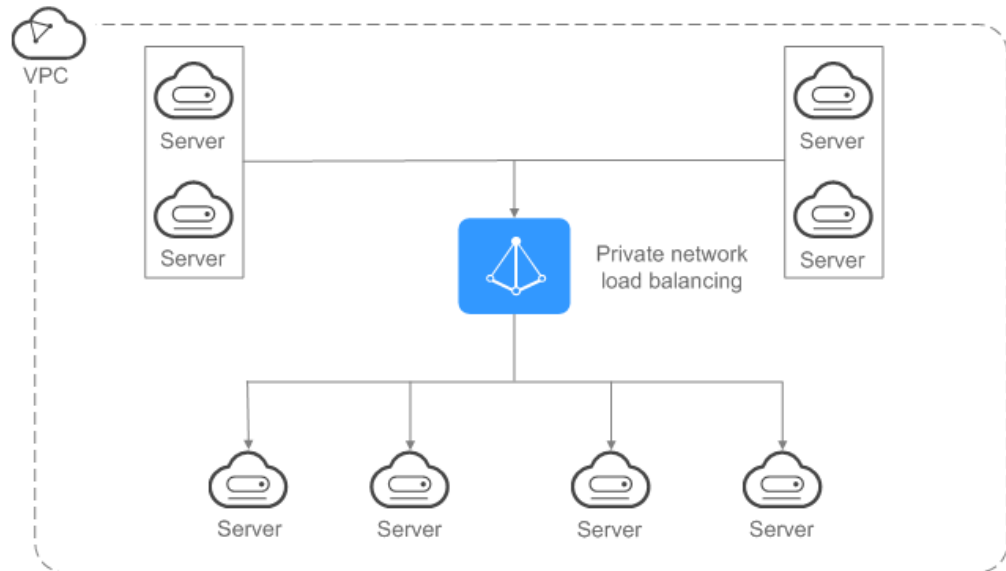
Figure 2-1 Public network load balancer



Private Network Load Balancer

A private network load balancer uses a private IP address to route requests from clients across servers in the same VPC.

Figure 2-2 Private network load balancer



2.2 Preparing for Creation

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

Region

When you select a region, pay attention to the following:

- The region must be close to your customers to reduce network latency and improve the download speed.
- The region must be the same as that of backend servers. Currently, ELB cannot be deployed across regions.

Network Type

Load balancers are classified as public network load balancers or private network load balancers by network type.

- To distribute requests over the Internet, you need to create a public network load balancer.

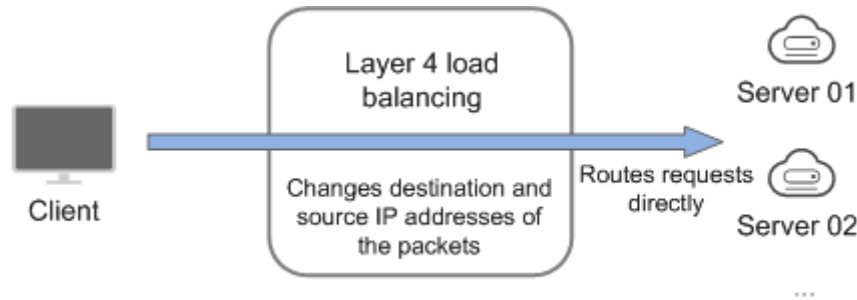
Each public network load balancer has an EIP bound to receive requests from the Internet.

- If you want to distribute requests within a VPC, create a private network load balancer.

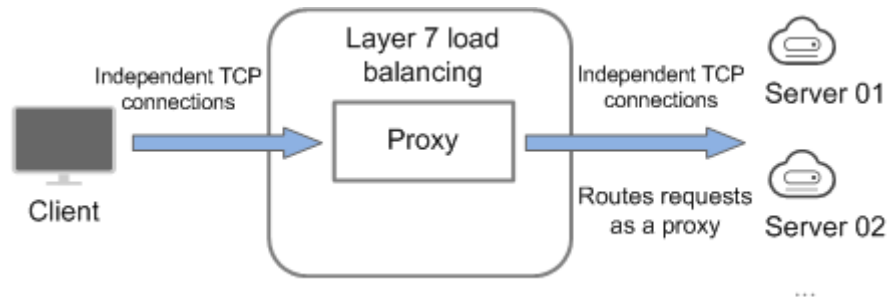
Protocol

ELB provides load balancing at both Layer 4 and Layer 7. Choose an appropriate protocol when adding a listener.

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



- Load balancing at Layer 7 is also called "content exchange". After receiving a request, the load balancer identifies and forwards the data based on the fields in the HTTP/HTTPS request header. The load balancer works as a proxy of backend servers to establish a connection (three-way handshake) with the client and receive requests from the client. After receiving a request, the load balancer determines to which backend server the request is to be routed based on the fields in the packets and the load balancing algorithm you selected when adding the listener. In this process, the load balancer functions as a proxy server that connects to both the client and backend server.



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 creating ECSs or BMSs, observe the following rules:

- The region of ECSs or BMSs must be the same as that of the load balancer.
- ECSs or BMSs that run the same OS are recommended so that you can manage them more easily.

2.3 Creating a Load Balancer

Prerequisites

You have prepared everything required for creating a load balancer. For details, see [2.2 Preparing for Creation](#).

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

Create 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. On the **Load Balancers** page, click **Create Elastic Load Balancer**. Set the parameters by referring to [Table 2-1](#).

Table 2-1 Parameter description

Parameter	Description	Example Value
Region	Specifies the region of the load balancer. 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.	N/A
Network Type	Specifies the network type of a load balancer. There are two options: <ul style="list-style-type: none">● Public network: A public network load balancer routes requests from the clients to backend servers over the Internet.● Private network: A private network load balancer routes requests from the clients to backend servers in a VPC.	Private network
VPC	Specifies the VPC where the load balancer works. Select an existing VPC or create one. For more information about VPC, see the <i>Virtual Private Cloud User Guide</i> .	-
Subnet	Specifies the subnet that the load balancer belongs to.	N/A

Parameter	Description	Example Value
Private IP Address	Specifies the IP address that will be bound to the load balancer. Enter an IP address if you do not select Automatic IP address allocation when selecting a subnet.	192.168.0.2
EIP	Specifies the public IP address that will be bound to the load balancer for receiving and forwarding requests over the Internet. The following options are available: <ul style="list-style-type: none">• New EIP: The system will automatically assign an EIP.• Use existing: Select an existing EIP.	New EIP
EIP Type	Specifies the link type (BGP) when a new EIP is used. Dynamic BGP: When changes occur on a network using dynamic BGP, routing protocols provide automatic, real-time optimization of network configurations, ensuring network stability and optimal user experience.	Dynamic BGP
Bandwidth	Specifies the bandwidth when a new EIP is used, in the unit of Mbit/s.	10 Mbit/s
Name	Specifies the load balancer name.	elb-ys0
Enterprise Project	Provides a cloud resource management mode, in which cloud resources and members are centrally managed by project.	default
Description	Provides supplementary information about the load balancer.	N/A
Tag	Identifies load balancers so that they can be easily categorized and quickly searched. A tag consists of a tag key and a tag value. That is, you can distinguish cloud resources from two dimensions. The tag key marks a tag, and the tag value specifies specific tag content. For details about the naming specifications, see Table 2-2 .	<ul style="list-style-type: none">• Key: elb_key1• Value: elb-01

Table 2-2 Naming rules of load balancer tags

Item	Requirement	Example Value
Tag key	<ul style="list-style-type: none">• Cannot be empty.• Must be unique for the same load balancer.• Can contain a maximum of 36 characters.• Cannot contain asterisks (*), angle brackets (< and >), backslashes (\), equal signs (=), commas (,), vertical bars (), or slashes (/).• Can contain letters, digits, underscores (_), hyphens (-), and Chinese characters.	elb_key1
Tag value	<ul style="list-style-type: none">• Can contain a maximum of 43 characters.• Cannot contain asterisks (*), angle brackets (< and >), backslashes (\), equal signs (=), commas (,), vertical bars (), or slashes (/).• Can contain letters, digits, underscores (_), dots (.), hyphens (-), and Chinese characters.	elb-01


5. Click **Create Now**.
6. Confirm the configuration and click **Submit**.

2.4 Changing Load Balancer Settings

Scenarios

For public network load balancers, you can change their bandwidth as required.

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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click **Modify Bandwidth** in the **Operation** column.
5. In the **New Configuration** area, change the bandwidth and click **Next**.
6. Click **Submit**.

2.5 Binding or Unbinding an EIP

Scenarios


You can bind an EIP to a private network load balancer. After the EIP is bound, the load balancer can receive requests over the Internet. You can also unbind the EIP from a public network load balancer. After the EIP is unbound, the load balancer can no longer receive requests over the Internet.

Bind an 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click **Bind EIP** in the **Operation** column.
5. In the **Bind EIP** dialog box, select the EIP to be bound and click **OK**.

Alternatively, go to the basic information page of the load balancer and click **Bind** beside **EIP**.

Unbind an 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click **More > Unbind EIP** in the **Operation** column.
5. Click **Yes**.

Alternatively, go to the basic information page of the load balancer and click **Unbind** beside **EIP**.

2.6 Deleting a Load Balancer

Scenarios

When a load balancer is not used any longer, you can delete it at any time. A deleted load balancer cannot be recovered. Exercise caution when performing this operation.


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

Prerequisites

Before you delete a load balancer, ensure that the following resources have been deleted or removed:



- Listeners
- Backend server groups
- Backend servers

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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click **Delete** in the **Operation** column.
5. Click **Yes**.

Export Load Balancer Information

You can also export the load balancer list as a local backup.

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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. In the upper right corner of the load balancer list, click .

3 Listener

3.1 Overview

At least one listener must be added to a load balancer. A listener receives requests from clients and routes requests to backend servers using the protocol, port, and load balancing algorithm you select.

Supported Protocols

ELB provides load balancing at both Layer 4 and Layer 7. Select a protocol that meets your needs in a specific scenario.

Table 3-1 Protocols supported by ELB

Protocol		Description	Application Scenario
Layer 4	TCP	<ul style="list-style-type: none">• Source IP address-based sticky sessions• Fast data transfer	<ul style="list-style-type: none">• Scenarios that require high reliability and data accuracy, such as file transfer, email sending and receiving, and remote login• Web applications with a number of concurrent connections or requiring high performance
Layer 4	UDP	<ul style="list-style-type: none">• Low reliability• Fast data transfer	Scenarios that focus more on timeliness than on reliability, such as video chats, games, and real-time financial quotations
Layer 7	HTTP	<ul style="list-style-type: none">• Cookie-based sticky sessions• X-Forward-For request header	Applications where data content needs to be identified, such as web applications and mobile games

Protocol		Description	Application Scenario
Layer 7	HTTPS	<ul style="list-style-type: none"> An extension of HTTP for encrypted data transmission to prevent unauthorized access Encryption and decryption performed on load balancers to reduce the workload of backend servers. 	Applications that require encrypted transmission

3.2 Protocols and Ports

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 needs.

Protocol	Port
TCP	<p>For one load balancer:</p> <ul style="list-style-type: none"> The port numbers of different protocols must be unique except UDP. Specifically, the port numbers used by UDP can be the same as those of other protocols. For example, if you have a UDP listener that uses port 88, you can add a TCP, HTTP, or HTTPS listener that also uses port 88. However, if you already have an HTTP listener that uses port 443, you cannot add an HTTPS or TCP listener with the same port. The port numbers of the same protocol must be unique. For example, if you have a TCP listener that uses port 80, you cannot add another listener with the same port. <p>The port numbers range from 1 to 65535. The following are some commonly-used protocols and port numbers: TCP/80 HTTPS/443</p>
UDP	
HTTP	
HTTPS	

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 3-2 Backend protocols and ports

Protocol	Port
TCP	Backend ports of the same load balancer can also be the same. The port numbers range from 1 to 65535. The following are some commonly-used protocols and port numbers: TCP/80 HTTP/443
UDP	
HTTP	

3.3 Adding a Listener

Scenarios

After creating a load balancer, you need to add at least one listener to the load balancer. A listener is a process that checks for requests using the protocol and port you configure for connections from clients to the load balancer, and the protocol and port from the load balancer to backend servers.

A listener also defines the health check configuration, based on which the load balancer continually checks the running statuses of backend servers. If a backend server is detected unhealthy, the load balancer routes traffic to these healthy ones. Traffic forwarding to this server resumes once it recovers.

Add 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Under **Listeners**, click **Add Listener**. Set the parameters by referring to [Table 3-3](#), [Table 3-4](#), and [Table 3-5](#).

Table 3-3 Parameters for configuring a listener

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

Parameter	Description	Example Value
Frontend Protocol/Port	<p>Specifies the protocol and port used by the load balancer to receive requests from clients and forward the requests to backend servers.</p> <p>The port numbers range from 1 to 65535, and the following protocols are supported:</p> <ul style="list-style-type: none">• HTTP• TCP• HTTPS• UDP	HTTP/80
Redirect	<p>Redirects requests to an HTTPS listener when HTTP is used as the frontend protocol. If you have both HTTPS and HTTP listeners, you can use this feature to redirect the requests from the HTTP listener to the HTTPS listener to ensure security.</p> <p>After an HTTP listener is redirected, backend servers return HTTP 301 Move Permanently to the clients.</p>	N/A
Redirected To	Specifies the HTTPS listener to which requests are redirected.	N/A
Server Certificate	Specifies the certificate used by the server to authenticate the client when HTTPS is used as the frontend protocol.	N/A
Advanced Settings		
HTTP/2	Specifies whether HTTP/2 is supported when you select HTTPS for Frontend Protocol .	N/A
Mutual Authentication	Specifies whether to enable mutual authentication between the server and client. To enable mutual authentication, both a server certificate and CA certificate are required. This feature can be enabled when Frontend Protocol is set to HTTPS .	N/A
CA Certificate	Specifies the certificate used by the server to authenticate the client when HTTPS is used as the frontend protocol. This parameter is mandatory when Frontend Protocol is set to HTTPS and mutual authentication is enabled.	N/A

Parameter	Description	Example Value
Description	Provides supplementary information about the listener.	N/A

Table 3-4 Parameters for adding a backend server group

Parameter	Description	Example Value
Backend Server Group	Specifies a group of servers with the same features that receive requests from the load balancer. Select Create new or Use existing .	Create new
Name	Specifies the name of the backend server group.	server_group-sq4v
Backend Protocol	Specifies the protocol used by backend servers to receive requests.	HTTP
Load Balancing Algorithm	<p>Specifies the algorithm used by the load balancer to distribute traffic.</p> <ul style="list-style-type: none"> • Weighted round robin: Requests are routed to different servers based on their weights, which indicate server processing performance. Backend servers with higher weights receive proportionately more requests, whereas equal-weighted servers receive the same number of requests. • Weighted least connections: In addition to the weight assigned to each server, the number of connections processed by each backend server is also considered. Requests are routed to the server with the lowest connections-to-weight ratio. • Source IP hash: The source IP address of the request is input into a hash algorithm, and the resulting hash is used to identify a server in the static fragment table. <p>NOTE Choose an appropriate algorithm based on your business needs for better traffic distribution.</p>	Weighted round robin

Parameter	Description	Example Value
Sticky Session	<p>Specifies whether to enable sticky sessions. After this feature is enabled, all requests from a client during one session are sent to the same backend server.</p> <p>NOTE For HTTP and HTTPS listeners, enabling or disabling sticky sessions may cause few seconds of service interruption.</p>	N/A
Sticky Session Type	<p>Specifies the sticky session type. The following options are available:</p> <ul style="list-style-type: none"> • Source IP address: The hash of the source IP address of the request is used to identify a server in the static fragment table. • 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 for processing. • Application cookie: The application deployed on the backend server generates a cookie after receiving the first request from the client. All requests with the cookie are routed to this backend server. <p>NOTE</p> <ul style="list-style-type: none"> • Source IP address is the only choice available when TCP is used as the frontend protocol. If HTTP or HTTPS is selected as the frontend protocol, the sticky session type can be Load balancer cookie or Application cookie. Choose an appropriate sticky session type to better distribute requests and improve load balancing. • Sticky sessions at Layer 4 are maintained for one minute, while sticky sessions at Layer 7 are maintained for 24 hours. 	Source IP address
Cookie Name	<p>Specifies the cookie name. When Application cookie is selected, you need to enter a cookie name.</p>	cookieName-qsp
Stickiness Duration (min)	<p>Specifies the duration that sticky sessions are maintained in minutes. The value ranges from 1 to 60.</p>	20

Parameter	Description	Example Value
Description	Provides supplementary information about the backend server group.	N/A

Table 3-5 Parameters for configuring a health check

Parameter	Description	Example Value
Enable Health Check	Specifies whether to enable the health check function.	N/A
Protocol	<ul style="list-style-type: none"> Specifies the protocol used by the load balancer to perform health checks on backend servers. You can use either TCP or HTTP. A selected protocol cannot be changed. If the frontend protocol is UDP, the health check protocol is UDP by default. 	HTTP
Domain Name	Specifies the domain name in the health check request. The domain name consists of digits, letters, hyphens (-), and periods (.), and must start with a digit or letter. This parameter is left blank by default and needs to be set only when the health check protocol is HTTP.	www.elb.com
Port	<p>Specifies the port used by the load balancer to perform health checks on backend servers. The port numbers range from 1 to 65535.</p> <p>NOTE This parameter is optional. If no health check port is specified, the port of each backend server is used for health checks by default.</p>	80
Advanced Settings	Provides some advanced features. Two options are available, Default and Custom .	Default
Interval (s)	Specifies the maximum time between health checks in the unit of second. The value ranges from 1 to 50 .	5
Timeout (s)	Specifies the maximum time required for waiting for a response from the health check in the unit of second. The value ranges from 1 to 50 .	10

Parameter	Description	Example Value
Check Path	Specifies the health check URL, which is the destination on backend servers for health checks. This parameter is available only when Protocol is set to HTTP . The value can contain 1 to 80 characters and must start with a slash (/).	/index.html
Maximum Retries	Specifies the maximum number of health check retries. The value ranges from 1 to 10 .	3

6. Click **Finish**.
7. Click **OK**.

3.4 Load Balancing Algorithms

ELB supports the following load balancing algorithms:

- **Weighted round robin:** Requests are distributed across 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. This algorithm is often used for short connections, such as HTTP connections.
- **Weighted least connections:** In addition to the number of active connections to each backend server, a weight is assigned to each backend server based on their processing capability. This algorithm is often used for persistent connections, such as database connections.
- **Source IP hash:** The source IP address of each request is calculated using the hash algorithm to obtain a unique hash key, and all backend servers are numbered. The generated key allocates the client to a particular server. This allows requests from different clients to be routed based on source IP addresses and ensures that a client is directed to the same server that it was using previously. This algorithm works well for TCP connections of load balancers that do not use cookies.

Server Weight


Each backend server can be given a numeral value from 0 to 100 to indicate the proportion of requests to receive. Requests will not be routed to the backend server whose weight is 0, when its health check result is meaningless. The following algorithms allow you to set the server weight:

- **Weighted round robin:** If none of the servers have a weight of 0, the load balancer routes requests to these servers using the round robin algorithm based on their weights. When the weights of backend servers are the same, the weights do not take effect, and the load balancer distributes requests using the round robin algorithm.
- **Weighted least connections:** If none of the servers have a weight of 0, the load balancer calculates each server's workload using the formula: Overhead

= Number of current connections/Server weight. The load balancer routes requests to the backend server with the lowest overhead in each request distribution.

- Source IP hash: The server weight does not take effect. Requests from same IP address are scheduled to the same backend server.

To set the server weight, 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Backend Server Groups**, locate the target backend server group and then the target server, and click the number in the **Weight** column to set the server weight.
6. Click **OK**.

3.5 Sticky Session

Sticky sessions are a mechanism that ensures that requests from a client always get routed to the same server before a session elapses.

Here we use an example to explain what sticky sessions can do. Suppose 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 just need to log in once.

Sticky Sessions at Layer 4

Sticky sessions at Layer 4 use source IP addresses to maintain sessions. Requests from the same IP address are routed to the same backend server during the session.

Sticky sessions at Layer 4 will be invalid in the following scenarios:

- Source IP addresses of the clients change.
- Requests from the clients exceed the session stickiness duration.

NOTE

- You can set the stickiness duration only when you select **Weighted round robin** as the load balancing algorithm.
- The default stickiness duration is 20 minutes, and the maximum duration is 60 minutes (that is, 1 hour).

Sticky Sessions at Layer 7

Sticky sessions at Layer 7 allow you to use load balancer cookies or application cookies to maintain sessions. Choose an appropriate sticky session type to better distribute requests across backend servers.

- **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 for processing.
- **Application cookie:** The application deployed on the backend server generates a cookie after receiving the first request from the client. All requests with the cookie are routed to this backend server.

Sticky sessions at Layer 7 will be invalid in the following scenarios:



- If requests sent by the clients do not contain a cookie, sticky sessions do not take effect.
- Requests from the clients exceed the session stickiness duration.

 **NOTE**

- You can set the stickiness duration only when you select **Weighted round robin** as the load balancing algorithm.
- The default stickiness duration is 20 minutes, and the maximum duration is 1440 minutes (that is, 24 hours).

ELB supports three types of sticky session, including **Source IP address**, **Load balancer cookie**, and **Application cookie**.

Enable Sticky Sessions

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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Backend Server Groups**, locate the target backend server group, and click  on the right of its name.
6. Enable the sticky session feature, select the sticky session type, and set the session stickiness duration.
7. Click **OK**.

3.6 Access Control

Access control allows you to add a whitelist to specify IP addresses that can access a listener.

NOTICE

- You can add whitelists only to listeners. Adding whitelists may cause service risks. Once a whitelist is added, only IP addresses in the whitelist can access the listener.
- If access control is enabled but no whitelist is added, the listener cannot be accessed.
- Access control does not conflict with inbound security group rules. Whitelists define the IP addresses or CIDR blocks from which the load balancer receives traffic, whereas inbound security group rules specify the protocol, ports, and IP addresses that allow traffic to backend servers.

Add a Whitelist


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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Listeners**, locate the target listener, and click its name. In the **Basic Information** area, click **Configure** beside **Access Control**.

Table 3-6 Parameter description

Parameter	Description	Example Value
Access Control	<p>Enabled</p> <ul style="list-style-type: none"> • If access control is enabled and no whitelist is set, no IP address can access the listener. • If access control function is enabled and a whitelist is set, only IP addresses in the whitelist can access the listener. <p>Disabled</p> <ul style="list-style-type: none"> • If access control is disabled, the listener can be accessed from any IP address. 	N/A
Whitelist	<p>Lists the IP addresses or CIDR blocks that can access the listener.</p> <p>NOTE</p> <ul style="list-style-type: none"> • A maximum of 300 IP addresses or CIDR blocks are supported. A comma (,) is used to separate every two entries. • The whitelist does not support IPv6 addresses. 	10.168.2.24,1 0.168.16.0/24



6. Click **OK**.

3.7 Modifying or Deleting a Listener


Scenarios

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


Modify 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Listeners**, locate the target listener, and click  on the right of its name.
6. Click **OK**.

Delete 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.

NOTE

- If the listener has backend servers associated, disassociate the backend servers before deleting the listener.
 - If HTTP requests are redirected to HTTPS, delete the redirect before deleting the HTTPS listener.
 - After a listener is deleted, the associated backend server group is also deleted.
5. Click **Listeners**, locate the target listener, and click  on the right of its name.
 6. Click **Yes**.

3.8 Advanced Settings for HTTP or HTTPS Listeners

3.8.1 Forwarding Policy

Scenarios

ELB allows you to add forwarding policies to forward requests based on domain names or URLs. This function is only supported for HTTP or HTTPS listeners.

A maximum of 500 forwarding policies can be added to a listener. With forwarding policies, requests for videos, images, audio, or text are forwarded to different backend server groups, making it easy to allocate resources.


When you add a forwarding policy, pay attention to the following:

- Each URL path can be found on the backend servers in the backend server group to which requests are forwarded. Otherwise, the backend servers will return 404.
- The same path cannot be configured for two forwarding policies.
- In regular expression match, sequential matching is used, and matching ends when any rule is successfully matched. Therefore, matching rules cannot overlap with each other if you select **Regular expression match** for **URL Matching Rule**.

After a forwarding policy is added, the load balancer forwards requests based on the specified domain name or URL:

- If the domain name or URL in a request matches the forwarding policy, the request is forwarded to the backend server group you configured when adding the forwarding policy.
- If the domain name or URL in a request does not match the forwarding policy, the request is forwarded to the default backend server group associated with the listener.

Add 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Listeners**, locate the target listener, and click its name.
6. Click **Add** on the right of **Forwarding Policies**.
7. In the **Add Forwarding Policy** dialog box, specify the parameters by referring to [Table 3-7](#).
8. Click **OK**.

Alternatively, locate the target load balancer in the load balancer list and click the name of the target listener in the **Listener** column. In the **Listeners** area, click **Add** on the right of **Forwarding Policies** and then add a forwarding policy.

Table 3-7 Forwarding policy parameters

Item	Parameter	Description	Example Value
Configure Forwarding Policy	Name	Specifies the forwarding policy name.	forwarding_policy-q582

Item	Parameter	Description	Example Value
	Domain Name	Specifies the domain name for triggering the forwarding policy. The specified domain name will be exactly matched. Note that either a domain name or URL must be specified.	www.test.com
	URL	Specifies the URL for triggering the forwarding policy.	/login.php
	URL Matching Rule	<ul style="list-style-type: none"> • Exact match The request URL is identical to the preset URL. • 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. <p>NOTE Exact match enjoys the highest priority, followed by Prefix match. Regular expression match is the last matching rule that will be used.</p>	Exact match
	Description	Provides supplementary information about the forwarding policy.	N/A
Add Backend Server Group	Backend Server Group	<p>Specifies whether a new or existing backend server group will be used. Select Create new or Use existing. If you select Create new, set parameters by referring to Table 4-1 and Table 4-2.</p> <p>NOTE The backend protocol can only be HTTP.</p>	Create new

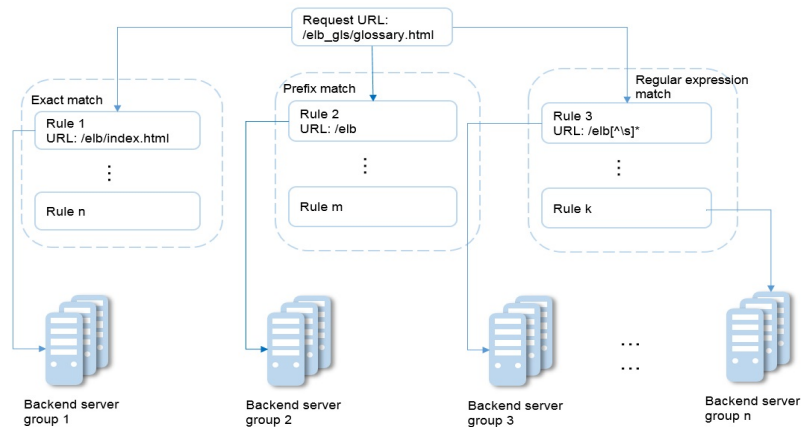
URL Matching Example

The following table lists how a URL is matched, and [Figure 3-1](#) shows how a request is forwarded to a backend server group.

Table 3-8 URL matching


URL Matching Rule	URL	Preset URL			
		/elb/index.html	/elb	/elb[^\s]*	/index.html
-	-	/elb/index.html	/elb	/elb[^\s]*	/index.html
Exact match	/elb/index.html	√	-	-	-
Prefix match		√	√	-	-
Regular expression match		√	-	√	-


Figure 3-1 Request forwarding





In this figure, the system first searches the requested URL (/elb_gls/glossary.html) using the **Exact match** rule. If no exactly matched URL is found, the **Prefix match** rule is used. If the start string of the requested URL matches that of specified URL, the request is forwarded to backend server group 2. Even if the requested URL also matches rule 3 (**Regular expression match**), the request is forwarded to backend server group 2 because **Prefix match** takes effect in priority.

Modify 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Listeners**, locate the target listener, and click its name.
6. Click **Forwarding Policies**.

7. Locate the target forwarding policy and click  on the right of its name.
8. In the **Modify Forwarding Policy** dialog box, modify the parameters and click **OK**.

Delete 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Listeners**, locate the target listener, and click its name.
6. Click **Forwarding Policies**.
7. Locate the target forwarding policy and click  on the right of its name.
8. Click **Yes**.

3.8.2 Mutual Authentication

Scenarios

In common HTTPS service scenarios, only the server certificate is required for authentication. For some mission-critical services, such as bank payment, the identities of both communication parties need to be authenticated, for which mutual authentication is required to ensure service security.

In this case, you need to deploy both the server certificate and client certificate.

Self-signed certificates are used here to describe how to configure mutual authentication. Self-signed certificates do not provide all of the security properties that certificates signed by a CA aim to provide. You are advised to purchase certificates from other authorities.

Create a CA Certificate Using OpenSSL

1. Log in to a Linux server with OpenSSL installed.
2. Create the **server** directory and enter the directory:
mkdir ca
cd ca
3. Create the certificate configuration file **ca_cert.conf**. The file content is as follows:

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

[ req_distinguished_name ]
O = ELB
```
4. Create the CA certificate private key **ca.key**.
openssl genrsa -out ca.key 2048

Figure 3-2 Private key of the CA certificate

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

5. Create the certificate signing request (CSR) file **ca.csr** for the CA certificate.
openssl req -out ca.csr -key ca.key -new -config ./ca_cert.conf
6. Create the self-signed CA certificate **ca.crt**.
openssl x509 -req -in ca.csr -out ca.crt -sha1 -days 5000 -signkey ca.key

Figure 3-3 Creating a self-signed CA certificate

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

Issue a Server Certificate Using the CA Certificate

The server certificate can be a CA signed certificate or a self-signed one. The following steps use a self-signed certificate as an example to describe how to create a server certificate.

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

```
mkdir server
```

```
cd server
```

3. Create the certificate configuration file **server_cert.conf**. The file content is as follows:

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

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

NOTE

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

4. Create the server certificate private key **server.key**.
openssl genrsa -out server.key 2048
5. Create the CSR file **server.csr** for the server certificate.
openssl req -out server.csr -key server.key -new -config ./server_cert.conf
6. Use the CA certificate to issue the server certificate **server.crt**.
openssl x509 -req -in server.csr -out server.crt -sha1 -CAcreateserial -days 5000 -CA ../ca/ca.crt -CAkey ../ca/ca.key

Figure 3-4 Issuing a server certificate

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

Issue a Client Certificate Using the CA Certificate

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

```
mkdir client
```

```
cd client
```

3. Create the certificate configuration file **client_cert.conf**. The file content is as follows:

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

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

NOTE

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

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

```
openssl genrsa -out client.key 2048
```

Figure 3-5 Creating a client certificate private key

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

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

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

Figure 3-6 Creating a client certificate CSR file

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

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

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

Figure 3-7 Issuing a client certificate

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

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

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

 **NOTE**

A password is required during command execution. Save this password, which is required when the certificate is imported to the browser.

Configure the Server Certificate and Private Key

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

 **NOTE**

Delete the last newline character to avoid an error when you copy the content.

 **NOTE**

The content of the certificate and private key must be PEM-encoded.

Configure the CA Certificate

Step 1 Log in to the management console.

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

Step 3 Click **Create Certificate**. In the **Create Certificate** dialog box, select **CA certificate**, copy the content of CA certificate **ca.crt** created in [Issue a Server Certificate Using the CA Certificate](#) to the **Certificate Content** area, and click **OK**.

 **NOTE**

Delete the last newline character to avoid an error when you copy the content.

 **NOTE**

The certificate content must be PEM-encoded.

----End

Configure Mutual Authentication

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

Add Backend Servers

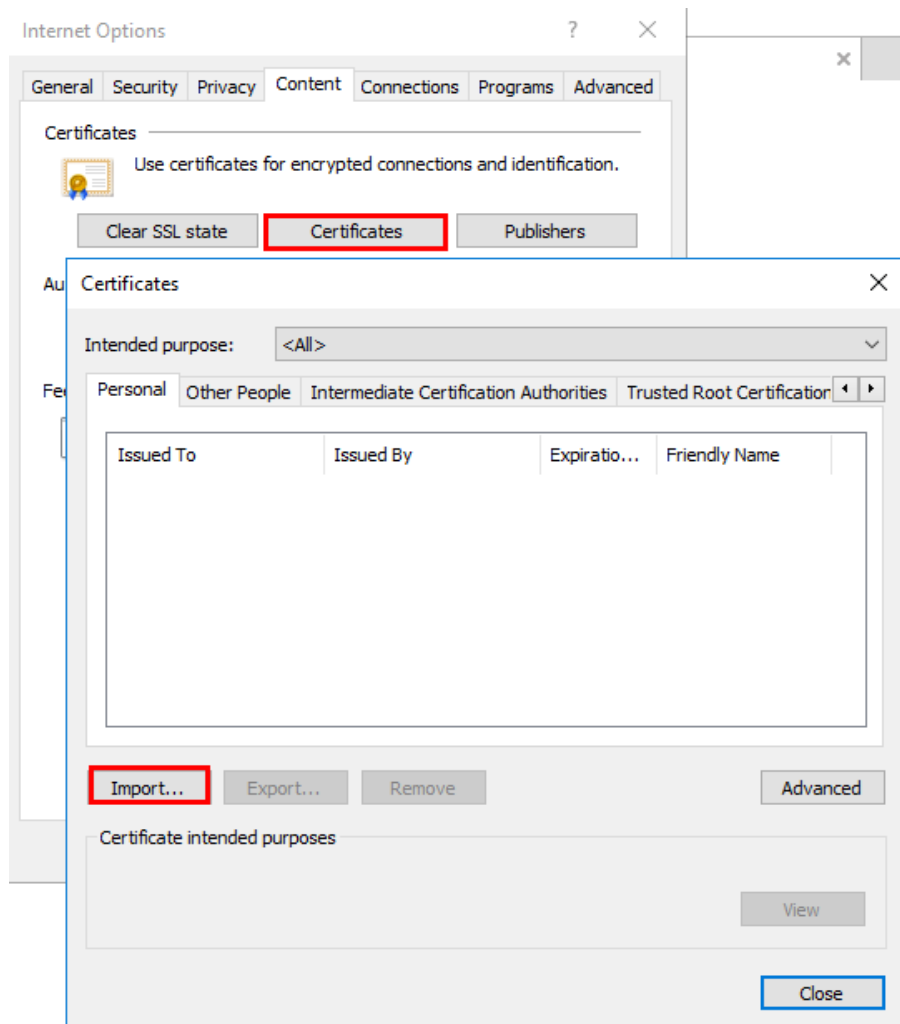
For detailed operations, see [Add Backend Servers](#).

Import and Test the Client Certificate

Method 1: Using a Browser

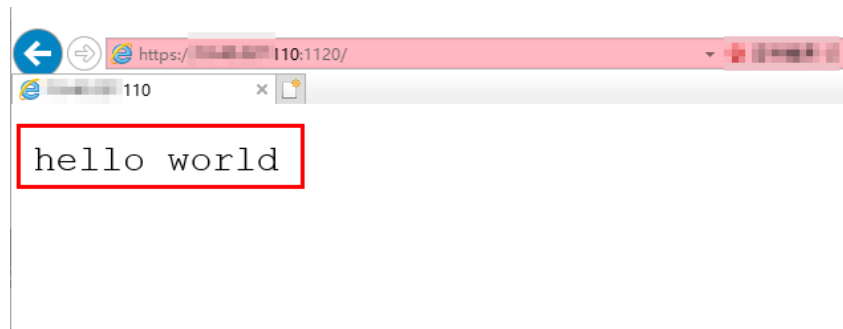
1. Import the client certificate using a browser (Internet Explorer 11 is used as an example).
 - a. Export **client.p12** from the Linux server.
 - b. Open the browser, choose **Settings > Internet Options** and click **Content**.
 - c. Click **Certificates** and then **Import** to import the **client.p12**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

Figure 3-8 Importing the **client.p12** certificate



2. Verify the import.

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

Figure 3-9 Accessing the website**Method 2: Using cURL**

1. Import the client certificate.

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

2. Verify the import.

On the Shell screen, run the following command:

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

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

Figure 3-10 Example of a correct response code


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

3.8.3 HTTP/2

Scenarios

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

Enable HTTP/2


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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Under **Listeners**, click **Add Listener**.

6. In the **Add Listener** dialog box, expand **Advanced Settings** and enable this option.
7. Click **OK**.

 **NOTE**

This option can be configured only for HTTPS listeners.

Disable HTTP/2

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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Listeners**, locate the target listener, and click **Modify**.
6. In the **Modify Listener** dialog box, expand **Advanced Settings** and disable this option.
7. Click **OK**.

3.8.4 HTTP Redirection to HTTPS

Scenarios

HTTPS is an extension of HTTP. HTTPS encrypts data between a web server and a browser. Redirection allows requests to be redirected from HTTP to HTTPS.

After redirection is enabled, all HTTP requests to access your website are transmitted over HTTPS connections to improve service security.

Prerequisites

- An HTTPS listener has been added.
- An HTTP listener has been added.

Create a Redirect


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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Listeners**, locate the target HTTP listener, and click its name.
6. Click **Redirects** and then **Create** on the right.

Table 3-9 Parameters for configuring redirection


Parameter	Description	Example Value
Name	Specifies the redirect name.	redirect-g8h9
Redirected To	Specifies the HTTPS listener to which requests are redirected.	N/A
Description	Provides supplementary information about the redirect.	N/A

7. Click **OK**.


 **NOTE**

- If requests to an HTTP listener are redirected, its configuration becomes invalid except for access control.
- After an HTTP listener is redirected, backend servers return HTTP 301 Move Permanently to the clients.

Modify a Redirect

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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Listeners**, locate the target HTTP listener, and click its name.
6. Click **Redirects**, locate the target redirect, and click **Modify** in the **Operation** column.
7. In the **Modify Redirect** dialog box, modify the redirect name or description, or select another listener, and click **OK**.

Delete a Redirect

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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Listeners**, locate the target listener, and click its name.
6. Click **Redirects**, locate the target redirect, and click **Delete** in the **Operation** column.
7. In the **Delete Redirect** dialog box, click **Yes**.

3.8.5 SNI

Scenarios

If an application provides multiple domain names and each domain name uses a different certificate, you can enable SNI when adding an HTTPS listener. SNI is an extension to TLS. It allows a server to present multiple certificates on the same IP address and TCP port number and hence allows multiple secure (HTTPS) websites (or any other service over TLS) to be served by the same IP address without requiring all those sites to use the same certificate. Before SNI, one server can use only one certificate. SNI allows the client to submit the domain name information while sending an SSL handshake request. Once receiving the request, the load balancer queries the right certificate based on the domain name and returns it to the client. If no certificate is found, the load balancer will return a default certificate.


Prerequisites

A certificate has been created. For details, see [6.3 Creating a Certificate](#).

 **NOTE**

Specify the domain name for the SNI certificate. Only one domain name can be specified for each certificate.

Procedure

1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Listeners** and locate the target listener. In the **Basic Information** area, click **Configure** on the right of **SNI**.
6. Enable SNI and select the SNI certificate to be used.
7. Click **OK**.

4 Backend Server

4.1 Overview

A backend server is cloud server added to a backend server group associated with a load balancer. When adding a listener to a load balancer, you create or select a backend server group to receive requests from the load balancer using the port and protocol you specify for the backend server group and the load balancing algorithm you select.

After a new server is added to the associated backend server group for which health check is configured, the load balancer will check its running status. If the backend server responds properly, it is declared healthy. If the backend server does not respond properly, the load balancer periodically checks its health for several times. Once it is declared healthy, it can receive requests from the load balancer.

You can adjust the number of backend servers to ensure stable and reliable service based on your budget. Load balancers can distribute requests across backend servers in different AZs to prevent SPOFs. You must ensure that at least one backend server is working properly in each AZ.

Precautions

When adding backend servers, pay attention to the following:

- Backend servers must be in the same VPC as the load balancer.
- It is recommended that backend servers run the same OS for ease of management and maintenance.
- You can set a weight for each server in the backend server group. The higher the weight is, the more requests the backend server receives.
- If you enable the sticky session feature, the proportions of requests processed by backend servers may become unbalanced. In this case, disable the sticky session feature and wait until each backend server receives almost the equal proportion of requests.


4.2 Configuring Security Group Rules

Scenarios

Before adding servers to a backend server group, ensure that their security groups have inbound rules that allow traffic from 100.125.0.0/16, and specify the health check protocol and port. Otherwise, health checks cannot be conducted for the added servers. If UDP is used for health checks, inbound security group rules must allow the ICMP traffic in addition to allowing access from 100.125.0.0/16.

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, you need to configure inbound rules for security groups containing these servers.

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 **Computing**, click **Elastic Cloud Server**.
4. In the ECS list, locate the target ECS and click its name.
The ECS details page is displayed.
5. Click **Security Groups**, locate the target security group, and view security group rules.
6. Click the security group rule ID or **Modify Security Group Rule** in the right corner.
The security group details page is displayed.
7. Under **Inbound Rules**, click **Add Rule**.
TCP, HTTP, or HTTPS listeners:
 - If the health check port is different from the ports of backend servers, the inbound rules must allow TCP traffic from the health check port and backend server ports.
 - If no health check port is specified, the inbound rules must allow TCP traffic from the ports of backend servers.
 - In addition, the inbound rules must allow access from 100.125.0.0/16. Otherwise, health checks may fail.UDP listeners:
 - If the health check port is different from the ports of backend servers, the inbound rules must allow UDP traffic from the health check port and backend server ports.
 - If no health check port is specified, the inbound rules must allow UDP traffic from the ports of backend servers.
 - The inbound rules must allow access from 100.125.0.0/16. Otherwise, health checks may fail.


- The inbound rules must allow ICMP traffic.
8. Click **OK**.

Network ACL Rule

A network ACL is an optional subnet-class security configuration. You can associate one or more subnets with a network ACL for controlling traffic in and out of the subnets. Similar to security groups, network ACLs provide access control functions, but add an additional layer of defense to your VPC. Default network ACL rules reject all inbound and outbound traffic. If a network ACL and load balancer reside in the same subnet, or the network ACL and backend servers associated with the load balancer reside in the same subnet, the load balancer cannot receive traffic from the public or private network, or backend servers become unhealthy.

You can configure an inbound network ACL rule to permit access from 100.125.0.0/16.

ELB translates public IP addresses that access backend servers into IP addresses in 100.125.0.0/16. Therefore, you cannot configure network ACL rules to prevent public IP addresses from accessing 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. Under **Network**, click **Virtual Private Cloud**.
4. In the navigation pane on the left, choose **Network ACLs**.
5. Locate the target network ACL, and click the network ACL name to switch to the network ACL details page.
6. On the **Inbound Rules** or **Outbound Rules** tab page, click **Add Rule** to add an inbound or outbound rule.
 - **Action:** Select **Allow**.
 - **Protocol:** The protocol must be the same as the frontend protocol set when the listener is added.
 - **Source:** Set the value to **100.125.0.0/16**.
 - **Source Port Range:** Select the port range of the service.
 - **Destination:** Enter default value **0.0.0.0/0**, which indicates that traffic from all IP addresses is permitted.
 - **Destination Port Range:** Select the port range of the service.
 - **Description:** provides supplementary information about the network ACL rule. This parameter is optional.
7. Click **OK**.

4.3 Adding Backend Servers to or Removing Backend Servers


Scenarios

When using ELB, ensure that at least a healthy backend server is in the backend server group associated with your load balancer. If the number of requests increases, you need to add more backend servers.

After a backend server is removed, it cannot receive requests from the load balancer. You can add it back to the backend server group when the traffic goes up again.

If a load balancer is associated with an AS group, instances in the AS group are automatically added to the backend server group of the load balancer. If instances are removed from the AS group, they will be automatically removed from the backend server group.

Add 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Backend Server Groups**, locate the target backend server group, and click its name.
6. In the **Basic Information** area, click **Add** in the upper left corner of the server list. Select the subnet where the backend servers reside, select the backend servers to be added, and click **Next**.

NOTE

If a backend server has multiple NICs, you can only select the subnet where the primary NIC resides and use the primary NIC to add the backend server.


7. Set the weights and backend ports of backend server and click **Finish**.

NOTE

In the **Backend Port** text box, enter the port of the backend server. If the ports of multiple backend servers are the same, you can enter the ports in the **Batch Add Port** text box and then click **OK**.

8. Click **OK**.

Remove 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Backend Server Groups**, locate the target backend server group, and click its name.
6. In the **Basic Information** area, click **Remove** in the **Operation** column to remove a backend server. To remove multiple backend servers, select all the backend servers to be removed and click **Remove** above the server list.
7. Click **Yes**.

Add a Backend Server 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Under **Backend Server Groups**, click **Add Backend Server Group**.
6. In the **Add Backend Server Group** dialog box, set the parameters.
For details about the parameters, see [Table 4-1](#) and [Table 4-2](#).

Table 4-1 Parameters for adding a backend server group

Parameter	Description	Example Value
Name	Specifies the name of the backend server group.	server_group-sq4v
Backend Protocol	Specifies the protocol used by backend servers to receive requests.	HTTP

Parameter	Description	Example Value
Load Balancing Algorithm	<p>Specifies the algorithm used by the load balancer to distribute traffic.</p> <ul style="list-style-type: none">• Weighted round robin: Requests are routed to different servers based on their weights, which indicate server processing performance. Backend servers with higher weights receive proportionately more requests, whereas equal-weighted servers receive the same number of requests.• Weighted least connections: The Least connections algorithm uses the number of active connections to each backend server to make its load balancing decision. Building on Least connections, the Weighted least connections algorithm assigns a weight to each server based on their processing capability.• Source IP hash: The source IP address of the request is input into a hash algorithm, and the resulting hash is used to identify a server in the static fragment table. <p>NOTE Choose an appropriate algorithm based on your business needs for better traffic distribution.</p>	Weighted round robin
Sticky Session	<p>If the sticky session feature is enabled, all requests from the same client during one session are sent to the same backend server.</p> <p>NOTE For HTTP and HTTPS listeners, enabling or disabling sticky sessions may cause few seconds of service interruption.</p>	N/A

Parameter	Description	Example Value
Sticky Session Type	<p>Specifies the sticky session type. The following options are available:</p> <ul style="list-style-type: none">• Source IP address: Requests with the same source IP address are routed to the same backend server for processing.• 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 for processing.• Application cookie: This method relies on backend applications. The application on the first backend server that receives the request generates a cookie. All subsequent requests that contain the cookie will be processed by the same backend server. <p>NOTE</p> <ul style="list-style-type: none">• Source IP address is the only choice available when TCP is used as the frontend protocol. If HTTP or HTTPS is selected as the frontend protocol, the sticky session type can be Load balancer cookie or Application cookie. Choose an appropriate sticky session type to better distribute requests and improve load balancing.• Sticky sessions at Layer 4 are maintained for one minute, while sticky sessions at Layer 7 are maintained for 24 hours.	Application cookie
Cookie Name	Specifies the cookie name. When Application cookie is selected, you need to enter a cookie name.	cookieName-qsps
Stickiness Duration (min)	Specifies the duration that sticky sessions are maintained in minutes. The value ranges from 1 to 60 .	20
Description	Provides supplementary information about the backend server group.	N/A



Table 4-2 Parameters for configuring a health check

Parameter	Description	Example Value
Enable Health Check	Specifies whether to enable the health check function.	N/A
Protocol	Specifies the protocol used by the load balancer to perform health checks on backend servers. <ul style="list-style-type: none">• If the frontend protocol is TCP, HTTP or HTTPS, the health check protocol can be TCP or HTTP. The health check protocol cannot be changed once it is set.• If the frontend protocol is UDP, the health check protocol is UDP by default.	HTTP
Domain Name	Specifies the domain name in the health check request. The domain name consists of digits, letters, hyphens (-), and periods (.), and must start with a digit or letter. This parameter is left blank by default and needs to be set only when the health check protocol is HTTP.	www.elb.com
Port	Specifies the port used by the load balancer to perform health checks on backend servers. This is an optional parameter. The port numbers range from 1 to 65535. NOTE If no port is specified, the port of each backend server is used for health checks by default.	80
Advanced Settings	Provides some advanced features. Two options are available, Default and Custom .	Default
Interval (s)	Specifies the maximum time between health checks in the unit of second. The value ranges from 1 to 50 .	5
Timeout (s)	Specifies the maximum time required for waiting for a response from the health check in the unit of second. The value ranges from 1 to 50 .	10



Parameter	Description	Example Value
Check Path	Specifies the health check URL, which is the destination on backend servers for health checks. This parameter is available only when Protocol is set to HTTP . The value can contain 1 to 80 characters and must start with a slash (/).	/index.html
Maximum Retries	Specifies the maximum number of health check retries. The value ranges from 1 to 10 .	3

7. Click **OK**.

Modify a Backend Server 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Backend Server Groups**, locate the target backend server group, and click  on the right of its name.
6. Modify the parameters as needed and click **OK**.

Delete a Backend Server 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Backend Server Groups**, locate the target backend server group, and click  on the right of its name.
6. Click **Yes**.

5 Health Check

5.1 Configuring a Health Check

Scenarios

You can configure a health check when you add a listener. If you have no special requirements, retain the default settings. You can also disable the health check function or change the health check settings.

Function Description

- The health check protocol and backend protocol are independent of each other. They can be the same or different from each other.
- To reduce the CPU usage of backend servers, it is recommended that you use TCP for health checks. If you want to use HTTP as the health check protocol, you are advised to use static files to return the health check results.
- You can increase the health check Interval to reduce the health check frequency.

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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Backend Server Groups**, locate the target backend server group, and click its name.
6. In the **Basic Information** area, click **Configure** beside **Health Check**.
7. In the **Configure Health Check** dialog box, enable or disable the health check function. Set the parameters by referring to [Table 5-1](#).

Table 5-1 Parameters for configuring a health check

Parameter	Description	Example Value
Enable Health Check	Specifies whether to enable the health check function.	N/A
Protocol	<ul style="list-style-type: none"> Specifies the protocol used by the load balancer to perform health checks on backend servers. You can use either TCP or HTTP. A selected protocol cannot be changed. If the frontend protocol is UDP, the health check protocol is UDP by default. 	HTTP
Domain Name	Specifies the domain name in the health check request. The domain name consists of digits, letters, hyphens (-), and periods (.), and must start with a digit or letter. This parameter is left blank by default and needs to be set only when the health check protocol is HTTP.	www.elb.com
Port	<p>Specifies the port used by the load balancer to perform health checks on backend servers. The port numbers range from 1 to 65535.</p> <p>NOTE This parameter is optional. If no health check port is specified, the port of each backend server is used for health checks by default.</p>	80
Interval (s)	<p>Specifies the maximum time between health checks in the unit of second.</p> <p>The value ranges from 1 to 50.</p>	5
Timeout (s)	<p>Specifies the maximum time required for waiting for a response from the health check in the unit of second.</p> <p>The value ranges from 1 to 50.</p> <p>NOTE The timeout must be less than or equal to the interval. Otherwise, the value set for the interval will be used as the timeout.</p>	3

Parameter	Description	Example Value
Check Path	Specifies the health check URL, which is the destination on backend servers for health checks. This parameter is available only when Protocol is set to HTTP . The value can contain 1 to 80 characters and must start with a slash (/).	/index.html
Maximum Retries	Specifies the maximum number of health check retries. The value ranges from 1 to 10 .	3

8. Click **OK**.


5.2 Disabling the Health Check Function

Scenarios

You can disable the health check function if you do not need health checks. If you have already configured a health check, you can also disable the health check function to stop checking server health results.

After the health check function is disabled, backend servers will not be detected, and the load balancer will consider the backend servers healthy. If a backend server becomes faulty or is working improperly, the load balancer will still route requests to this server. As a result, applications on this server are inaccessible. If this happens, you need to ensure that the ports of backend servers are normal. It is recommended that do not disable the health check function unless necessary.

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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. Locate the target load balancer and click its name.
5. Click **Backend Server Groups**, locate the target backend server group, and click its name.
6. In the **Basic Information** area, click **Configure** beside **Health Check**.
7. In the **Configure Health Check** dialog box, disable the health check function.
8. Click **OK**.

6 Certificate

6.1 Certificate and Private Key Format

Certificate Format

When creating a certificate, you can copy and paste the certificate content or directly upload the certificate.

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

The certificate content must meet the following requirements:

- The content starts with -----BEGIN CERTIFICATE----- and ends with -----END CERTIFICATE-----.
- Each row contains 64 characters except last row, which contains fewer than 64 characters.
- There are no empty rows.

The following is an example:

```
-----BEGIN CERTIFICATE-----
MIIDljCCAougAwlBAglJALV96mEtVF4EMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNV
BAYTAnh4MQswCQYDVQQIEwJ4eDELMAkGA1UEBxMCEHgxGjAJBgNVBAoTAnh4MQsw
CQYDVQQLEwJ4eDELMAkGA1UEAxMCEHgxGjAJBgkqhkiG9w0BCQEWc3h4eEAXNjMu
Y29tMB4XDTE3MTEyMzYyMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYyMjYy
eHgxGjAJBgNVBAgTAnh4MQswCQYDVQQHEwJ4eDELMAkGA1UEChMCEHgxGjAJBgNV
BAhTAnh4MQswCQYDVQQDEwJ4eDEaMBGCSqGSIb3DQEJARYLeHh4QDE2My5jb20w
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMU832iM+d3FILgTWmpZBUoYcIwV
cAAE7F5Z9LNEROyjJpyi256oypdBvG59JAUBN5WaFk81UQx29wAyNixX+bKa0DB
WpUDqr84V1f9vdQc75v9WoujcnlKszpV6qePPC7igJjpu4QOI362BrWzJCYQbg4
Uzo1KYBhLfxl0TovAgMBAAAgjgc8wgcvwwHQYDVR0OBBYEFMbtvDyvE2KsRy9zPq/J
WOjovG+WMIGcBgNVHSMegZQwgZGAFMbtvDyvE2KsRy9zPq/JWOjovG+WoW6kbDBq
MQswCQYDVQQGEwJ4eDELMAkGA1UECBMCEHgxGjAJBgNVBACtAnh4MQswCQYDVQQK
EwJ4eDELMAkGA1UECXMCEHgxGjAJBgNVBAMTAnh4MRowGAYJKoZIhvcNAQkBFgt4
eHhAMTYzLmNvbYIJALV96mEtVF4EMAWGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEF
BQADgYEAA5Kc/1iwiALa2RU3YCxqZFEEsZZvQxikrDkDbFeoa6Tk49Fnb1f7FCW6
PTtY3HPWL5ygsMsSy0Fi3xp3jmulwzJhcQ3tck5gC99HWp6Kw37RL8WoB8GWFUOQ
4tHLOjBixkZROPRhH+zMlRqUexv6fsb3NWKhnlfh1Mj5wQE4Ldo=
-----END CERTIFICATE-----
```

Private Key Format

When creating a server certificate, you also need a private key. 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 their content must meet the following requirements:

- The content must start with -----BEGIN RSA PRIVATE KEY----- and end with -----END RSA PRIVATE KEY-----.
- There are no empty rows. Each row must contain 64 characters except the last one, which contains fewer than 64 characters.

The following is an example:

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQDFPN9ojPndxSC4E1pqWQVKGHCFIXAAGBOxbGfSzXqzsoyacotu
eqMqXQbXrPSQFATeVmhzPNVEMdvcAMjYsV/mymtAwVqVA6q/OFdX/b3UHO+b/VqL
o3J5SrM86VeqnjzWu4oCSabuEDiN+tga1syQmEG4OFM6NSmAYSxcZdE6LwIDAQAB
AoGBAJvLzjCylsCjCKHWL6onbSutDtyFwPViD1QrVAtQYabF14g8CGUZG/9fgheu
TXPtTDcvu7czdUArvgYW3I9F9IBb2lmF3a44xfiAkDhZr4DK/vQhvHPuuTeZA41
r2zp8Cu+Bp40pSxmoAOK3B0/peZAka01Ju7c7ZChDWrXleHZAKEA/6dcaWHotfGS
eW5YLBsms3f0m0GH38nRL7oxyCW6yMIDkFHURVMBKW1OhrCuGo8u0nTMI5IH9gRg
5bH8XcujlQJBAMWBQgzCHyoSeryD3TFieXIFzqDBw6Ve5hyMjUtjvgdVKoxRPvpO
kclC39QHP6Dm2wrXXHEej+9RILxBZCVCQNbMCQQC42i+Ut0nHvPuXN/UkXzomDHde
h1ySsOAO4H+8Y6OSI87l3HURByCQ7stX1z3L0HofjHqV9Koy9emGTFLEzSdAkB7
Ei6cUKKmtkYe3rr+RcATEmwAw3tEJOHmrW5ErApVZKr2TzLMQZ7WZpIPzQRCYnY
2ZZLDuZWFFG3vW+wKKktAkAaQ5GNzbwkRLpXF1FZFuNF7erxypzstbUmU/31b7tS
i5LmxTGKL/xRYtZEHjya4lkkkg40q1MrUsgIYbFYMf2
-----END RSA PRIVATE KEY-----
```

6.2 Converting Certificate Formats

Scenarios

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

6.3 Creating a Certificate


Scenarios

To enable authentication for securing data transmission over HTTPS, ELB allows you to deploy certificates on load balancers.

NOTE

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

Create 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Certificates**.
5. Click **Create Certificate**. In the **Create Certificate** dialog box, configure the following parameters:
 - **Certificate Name**
 - **Certificate Type**
 - **Server certificate**: used for SSL handshake negotiations when an HTTPS listener is added. 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 Content**: 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.
 - **Private Key**
Click **Upload** and select the private key to be uploaded. Ensure that your browser is of the latest version.
The private key must be an unencrypted one, and its format is as follows:

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

 **NOTE**

If a certificate chain is used, you need to configure the content and private keys of all certificates in sequence, starting from the sub-certificate, and ensure that the certificate content is configured in the same sequence as private keys. For example, if you have three certificates: sub-certificate, intermediate certificate, and root certificate, the first one to be configured is the sub-certificate, followed by the intermediate certificate, and the last one is the root certificate.

– **Domain Name**


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

– **Description**


6. Click **OK**.

Delete a Certificate


Only certificates that are not in use can be deleted.

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

Modify 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 **Service List**. Under **Network**, click **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Certificates**.
5. Locate the target certificate and click **Modify** in the **Operation** column.
6. In the **Modify Certificate** dialog box, modify the parameters.
7. Click **OK**.

Bind 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 **Service List**. Under **Network**, click **Elastic Load Balance**.


4. Locate the target load balancer and click its name.
5. Under **Listeners**, click **Add Listener**.
6. In the **Add Listener** dialog box, set the parameters. When **Frontend Protocol** is set to **HTTPS**, a server certificate must be bound to the listener.
7. Click **OK**.

7 Access Logging

Scenarios

Access logs record HTTP and HTTPS requests to your load balancer in detail, such as request time, client IP address, request path, and server response. To enable access logging, you need to interconnect ELB with LTS and create a log group and log stream on the LTS console. After access logging is enabled, requests in new connections are recorded as logs, which are then uploaded to an AOM log bucket.

Configure Access Logging

1. Create a log group.
 - a. Log in to the management console.
 - b. In the upper left corner of the page, click  and select the desired region and project.
 - c. Click **Service List**. Under **Management & Deployment**, click **Log Tank Service**.
 - d. In the navigation pane on the left, choose **Log Management**.
 - e. Click **Create Log Group**. In the displayed dialog box, enter a name for the log group.
 - f. Click **OK**.
2. Create a log stream.
 - a. Locate the newly created log group and click its name.
 - b. Click **Create Log Stream**. In the displayed dialog box, enter a name for the log stream.
 - c. Click **OK**.
3. To display access logs in real time, you need to configure the search function for the log stream.
 - a. Locate the newly created log stream and click **Search** in the **Operation** column.
 - b. On the displayed page, enter the search criteria as prompted.
 - c. Click the search icon.

4. Configure access logging.
 - a. Click **Service List**. Under **Network**, click **Elastic Load Balance**.
 - b. Locate the target load balancer and click its name.
 - c. Under **Access Logs**, click **Configure Access Log**.
 - d. Enable access logging and select the created log group and log stream.
 - e. Click **OK**.

View Access Logs

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

There are two ways for you to view access logs.

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

The following is an example log. For details about the fields in the log, see [Table 7-1](#).

```
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
```

Table 7-1 Parameter description

Parameter	Description
msec	Time in seconds with a milliseconds resolution
access_log_topic_id	Log stream ID
time_iso8601	Local time in the ISO 8601 standard format
log_ver	Log format version
remote_addr: remote_port	IP address and port number of the client
status	HTTP status code
request_method scheme:// host router_request_uri server_protocol	Request method Request scheme://Hostname:URI Protocol (with version)
request_length	Length of the request received from the client, including the header and body
bytes_sent	Number of bytes sent to the client

Parameter	Description
body_bytes_sent	Number of bytes sent to the client (excluding the response header)
request_time	Request processing time in seconds, that is, the duration 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
upstream_status	Response status code returned by the backend server <ul style="list-style-type: none"> 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.
upstream_connect_time	Time taken to receive the response header from the backend server, in seconds with a millisecond resolution <ul style="list-style-type: none"> 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.
upstream_header_time	Time taken to receive the response header from the backend server, in seconds, with a millisecond resolution <ul style="list-style-type: none"> 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.
upstream_response_time	Time taken to receive the response header from the backend server, in seconds, with a millisecond resolution <ul style="list-style-type: none"> 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.
upstream_addr	Internal IP address and port number of the backend server This field can be ignored.

Parameter	Description
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
http_referer	http_referer content in the request header received by the load balancer, indicating the page link of the request
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
lb_name	Load balancer name in the format of loadbalancer_Load balancer ID
listener_name	Listener name in the format of listener_Listener ID
listener_id	Listener ID (This field can be ignored.)
pool_name	Backend server group name in the format of pool_backend server group ID
member_name	Backend server name in the format of member_server ID This field is not supported yet.
tenant_id	Tenant ID
eip_address:eip_port	EIP of the load balancer and frontend port set when the listener is added
upstream_addr_priv	IP address and port number of the backend server
certificate_id	[HTTPS listener] Certificate ID used for establishing an SSL connection This field is not supported yet.
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.
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.
sni_domain_name	[HTTPS listener] SNI domain name provided by the client during SSL handshake For a non-HTTPS listener, a hyphen (-) is displayed as a null value for this field.

Parameter	Description
tcpinfo_rtt	TCP Round Trip Time (RTT) between the load balancer and client in microseconds

Configure Log Transfer

If you want to perform secondary analysis on access logs, you can refer to this section to transfer logs to OBS or Data Ingestion Service (DIS) for storage.

1. Click **Service List**. Under **Management & Deployment**, click **Log Tank Service**.
2. Click **Log Transfer**.

✕

Create Log Transfer

* Log Group Name C

* Enterprise Project Name C view Enterprise

* Log Stream Name

* Transfer Mode OBS DIS

* OBS Bucket C View OBS Bucket

LTS will be authorized with read and write permissions for the OBS bucket. If the bucket policy is modified, ensure that the read and write permissions are not changed.

Custom Log Transfer Path

Log File Prefix ?

* Format

* Whether to Enable Transfer

* Transfer Period ?

3. Set the parameters based on site requirements. For details, see the Log Tank Service User Guide.

8 Monitoring

8.1 Monitoring Metrics

Overview

This section describes the metrics that can be monitored by Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the metrics of a monitored object and the generated alarms.

Namespace

SYS.ELB

Metrics

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m1_cps	Concurrent Connections	Number of TCP and UDP connections between the monitored object and backend servers Unit: Count	≥ 1	Load balancer or listener	1 minute

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

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m7_in_Bps	Inbound Rate	Traffic used for accessing the monitored object from the Internet Unit: byte/s	≥ 1 byte/s		
m8_out_Bps	Outbound Rate	Traffic used by the monitored object to access the Internet Unit: byte/s	≥ 1 byte/s		
m9_abnormal_servers	Unhealthy Servers	Number of unhealthy backend servers associated with the monitored object Unit: Count	≥ 1	Load balancer	1 minute
ma_normal_servers	Healthy Servers	Number of healthy backend servers associated with the monitored object Unit: Count	≥ 1		
Layer 7 (HTTP/HTTPS) metrics					
mb_l7_queries	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: Query/s	≥ 1/s	Load balancer or listener	1 minute
mc_l7_http_2xx	2xx Status Codes	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	≥ 1/s		

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
md_l7_http_3xx	3xx Status Codes	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	≥ 1/s		
me_l7_http_4xx	4xx Status Codes	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	≥ 1/s		
mf_l7_http_5xx	5xx Status Codes	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	≥ 1/s		
m10_l7_http_other_status	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	≥ 1/s		
m11_l7_http_404	404 Not Found	Number of 404 Not Found status codes returned by the monitored object (This metric is available only when the frontend protocol is HTTP or HTTPS.) Unit: Count/s	≥ 1/s		

Metric ID	Name	Description	Value	Monitored Object	Monitoring Period (Raw Data)
m12_l7_http_499	499 Client Closed Request	Number of 499 Client Closed Request status codes returned by the monitored object (This metric is available only when the frontend protocol is HTTP or HTTPS.) Unit: Count/s	≥ 1/s		
m13_l7_http_502	502 Bad Gateway	Number of 502 Bad Gateway status codes returned by the monitored object (This metric is available only when the frontend protocol is HTTP or HTTPS.) Unit: Count/s	≥ 1/s		
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	≥ 1 ms		

a: If a service has multiple dimensions, all dimensions are mandatory when you use APIs to query the metrics.

- Example of querying a single metric from both dimensions: `dim.0=lbaas_instance_id,223e9eed-2b02-4ed2-a126-7e806a6fee1f&dim.1=lbaas_listener_id,3baa7335-8886-4867-8481-7cbba967a917`
- Example of querying metrics in batches from both dimensions:

```
"dimensions": [
  {
    "name": "lbaas_instance_id",
    "value": "223e9eed-2b02-4ed2-a126-7e806a6fee1f"
```

```
}  
{  
  "name": "lbaas_listener_id",  
  "value": "3baa7335-8886-4867-8481-7cbba967a917"  
}  
],
```

Dimension

Key	Value
lbaas_instance_id	Specifies the load balancer ID.
lbaas_listener_id	Specifies the listener ID.

8.2 Setting an Alarm Rule

8.2.1 Adding an Alarm Rule

1. Log in to the management console.
2. Under **Management & Deployment**, click **Cloud Eye**.
3. In the navigation pane on the left, choose **Alarm Management > Alarm Rules**.
4. On the **Alarm Rules** page, click **Create Alarm Rule**.

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

- a. Select **Elastic Load Balance** for **Resource Type**.
- b. Select **Enhanced Load Balancers** for **Dimension**. You can also select **Listeners** if you want to monitor a listener.
- c. Set other parameters as required and then click **Create**.

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

NOTE

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

8.2.2 Modifying an Alarm Rule

1. Log in to the management console.
2. Under **Management & Deployment**, click **Cloud Eye**.
3. In the navigation pane on the left, choose **Alarm Management > Alarm Rules**.
4. On the **Alarm Rules** page, locate the target alarm rule. In the **Operation** column, click **More > Modify**.
 - a. Click the name of the target alarm rule.
 - b. In the upper right corner of the displayed page, click **Modify**.

- c. On the **Modify Alarm Rule** page, set parameters as prompted.
- d. 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 that complies with the alarm rule is generated.

 **NOTE**

8.3 Viewing Metrics

Scenarios

Cloud Eye is a monitoring service that which allows you to monitor your resources, including load balancers.

There is a short time delay between transmission and display of monitoring data, so the status of each load balancer displayed on the Cloud Eye dashboard at any given time is not its real-time status. For a newly created load balancer, you need to wait for about 5 minutes to 10 minutes before you can view its metrics.

Prerequisites

- The load balancer to be monitored 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 by configuring an alarm rule for the load balancer to be monitored on the Cloud Eye console.
Without alarm rules configured, there is no monitoring data. For details, see [8.2 Setting an Alarm Rule](#).

Procedure

1. Log in to the management console.
2. Under **Management & Deployment**, click **Cloud Eye**.
3. In the navigation pane on the left, choose **Cloud Service Monitoring > Elastic Load Balance**.
4. Locate the target load balancer and click **View Metric** in the **Operation** column.

9 Auditing

9.1 Key Operations Recorded by CTS

You can use CTS to record operations associated with ELB for query, auditing, and backtracking later.

Table 9-1 lists the operations that can be recorded by CTS.

Table 9-1 ELB operations that can be recorded by CTS

Action	Resource Type	Trace
Configuring access logs	accesslog	create access log
Deleting access logs	accesslog	delete access log
Creating a certificate	certificate	create certificate
Modifying a certificate	certificate	update certificate
Deleting a certificate	certificate	delete certificate
Creating a health check	healthmonitor	create healthmonitor
Modifying a health check	healthmonitor	update healthmonitor
Deleting a health check	healthmonitor	delete healthmonitor
Adding a forwarding policy	l7policy	create forwarding policy
Modifying a forwarding policy	l7policy	update forwarding policy
Deleting a forwarding policy	l7policy	delete forwarding policy
Adding a forwarding rule	l7rule	create forwarding rule


Action	Resource Type	Trace
Modifying a forwarding rule	l7rule	update forwarding rule
Deleting a forwarding rule	l7rule	delete forwarding rule
Adding a listener	listener	create listener
Modifying a listener	listener	update listener
Deleting a listener	listener	delete listener
Creating a load balancer	loadbalancer	create loadbalancer
Modifying a load balancer	loadbalancer	update loadbalancer
Deleting a load balancer	loadbalancer	delete loadbalancer
Adding a backend server	member	add backend ecs
Modifying a backend server	member	update backend ecs
Removing a backend server	member	remove backend ecs
Creating a backend server group	pool	create backend member group
Modifying a backend server group	pool	update backend member group
Deleting a backend server group	pool	delete backend member group

9.2 Viewing Traces

Scenarios

CTS records the operations performed on cloud service resources in the form of traces and allows you to view the operation records of the last seven days on the management console. This topic describes how to query these records.

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.


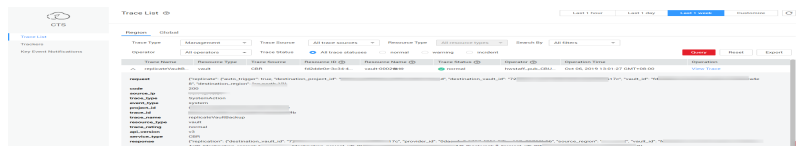
3. Under **Management & Deployment**, click **Cloud Trace Service**.
4. In the navigation pane on the left, choose **Trace List**.
5. Specify the filters used for querying traces. The following four filters are available:
 - **Trace Type, Trace Source, Resource Type, and Search By**
Select a filter from 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**, you need to select or enter a specific resource ID.
If you select **Resource name** for **Search By**, you need to 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 9-1 Expanding trace details

7. Click **View Trace** in the **Operation** column. In the **View Trace** dialog box, view details of the trace.

Figure 9-2 View Trace

```
"context": {
  "code": "204",
  "source_ip": "10.45.152.59",
  "trace_type": "ApiCall",
  "event_type": "system",
  "project_id": "0503dda89700fed2f78c00909158a4d",
  "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\": \"0503dda89700fed2f78c00909158a4d\", \"name\": \"9646e73b-338c-4d27-a17c-219be532812c\"}},",
  "resource_id": "9646e73b-338c-4d27-a17c-219be532812c",
  "tracker_name": "system",
  "time": "1569321775225",
  "resource_name": "9646e73b-338c-4d27-a17c-219be532812c",
  "record_time": "1569321775903",
  "user": {
    "domain": {
      "name": "0503dda87800fed0f75c0096d70a960",
      "id": "0503dda87800fed0f75c0096d70a960"
    }
  }
},
```

For details about key fields in the trace, see the [Cloud Trace Service User Guide](#).

10 FAQs

10.1 Questions Summary

- [How Can I Obtain the Real IP Address of a Client?](#)
- [How Do I Troubleshoot an Unhealthy Backend Server?](#)
- [What Are the Precautions of Using UDP?](#)
- [What Types of Sticky Sessions Does ELB Support?](#)
- [How Can I Use WebSocket?](#)
- [How Can I Rectify the Issue that Sticky Sessions Fail to Take Effect?](#)
- [What Are the Relationships Between the Load Balancing Algorithms and Sticky Session Types?](#)
- [How Does ELB Distribute Traffic?](#)

10.2 ELB Usage

10.2.1 Service Abnormality

10.2.1.1 How Can I Check ELB Unavailability or Routing Interruption?

1. Check the health of the backend server. If the backend server is unhealthy, traffic will be routed to those healthy ones.
2. Check whether security policies of the backend server allow access from 100.125.0.0/16.
3. Check the timeout duration of TCP connections between the client and the load balancer. By default, the timeout duration is 300s, which cannot be modified. If the timeout duration exceeds 300s, the load balancer sends an RST message to the client and backend server and disconnects the connection.
4. Check the source IP address before the request reaches the load balancer if the **Source IP hash** algorithm is used.

For example, if ELB works with the Content Delivery Network (CDN) or Web Application Firewall (WAF) service, the IP address of the request is changed after the request passes through CDN or WAF. As a result, the IP address is changed, and the session stickiness fails. If you want to use CDN or WAF, it is recommended that you add an HTTP or HTTPS listener and configure cookie-based sticky sessions.

5. Check the cookie value if sticky sessions are enabled for an HTTP or HTTPS listener. If the cookie value changes, traffic is routed to other backend servers.
6. Check the stickiness duration set for the backend server group. The default stickiness duration is 1 minute for TCP or UDP listeners, and 1440 minutes (24 hours) for HTTP or HTTPS listeners. If the stickiness duration times out, ELB becomes unavailable.

10.2.2 ELB Functionality

10.2.2.1 Can ELB Be Used Separately?

No. ELB cannot be used alone.

ELB is a service that distributes incoming traffic across servers and must be used with the ECS or BMS service.

10.2.2.2 Is the EIP Assigned to a Load Balancer Exclusive?

The EIP is not exclusive to and can be unbound from the load balancer. After you unbind the EIP, the load balancer can no longer receive requests over the Internet, and the EIP can be used by other resources.

10.2.2.3 How Many Load Balancers and Listeners Can I Have?

By default, an account can have a maximum of 50 load balancers and 100 listeners. If you need more, apply for a higher quota.

All load balancers in your account share the quota of listeners. The number of listeners that can be added to a load balancer is the remaining listener quota.

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

You can associate more backend servers with the load balancer or disassociate backend servers from the load balancer at any time. In addition, you can change the type of backend servers based on your business needs. To ensure service stability, ensure that the health check function is normal and that at least a healthy backend server has been associated with the load balancer.

10.2.2.5 Can Backend Servers Run Different OSs?

Yes.

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

10.2.3 Performance and Workloads

10.2.3.1 How Can I Check Traffic Inconsistency?

Check whether there are requests failed to be processed, especially requests with *4xx* status code. A possible cause is that requests are rejected by ELB and are not routed to backend servers because they are considered abnormal.

10.2.3.2 How Can I Check that Traffic Is Unbalanced?

1. Check whether sticky sessions are enabled. If sticky sessions are enabled and there are few clients, imbalance may occur.
2. Check the health state of backend servers, especially those whose health state changes over time. If the health check result is **Unhealthy** or switches between **Healthy** and **Unhealthy**, traffic is unbalanced.
3. Check whether the **Source IP hash** algorithm is used. If this algorithm, requests sent from the same IP address are routed to the same backend server, resulting in unbalanced traffic.
4. Check whether applications on the backend server use Keepalived to maintain TCP persistent connections. If yes, traffic may be unbalanced because the number of requests on persistent connections is different.
5. Check whether different weights are assigned to backend servers. The traffic varies according to the weights.

10.2.3.3 How Can I Check High Access Delay of a Load Balancer?

1. Bind an EIP to a backend server to make the applications accessible from the Internet and then check the access delay. By doing so, you can determine whether the problem is caused by the client, load balancer, or applications.
2. Check the incoming traffic. If the incoming traffic exceeds the maximum bandwidth set for the EIP, the access delay increases.
3. Check application workloads and security policies if there is high delay when the applications are accessed from the Internet.
4. Check the health of backend servers based on the **Unhealthy Servers** metric. If the applications are unstable and connections to the backend server time out, the retry mechanism will route the requests to another backend server. As a result, access to the applications is successful but the access delay increases.
5. If the problem persists, contact customer service.

10.2.3.4 What Should I Do If a Load Balancer's Performance Fails the Stress Test?

1. Check the workloads of backend servers. If the CPU usage reaches 100%, applications may have performance bottlenecks.
2. Check the incoming traffic. If the incoming traffic exceeds the maximum bandwidth set for the EIP, a large number of packets are lost and requests fail to be responded, affecting the ELB's performance.

3. Check the number of connections in the **time_wait** state on the clients if short connections are established for the test. A possible cause is that there are insufficient client ports.
4. The listening queue backlog of the backend server is full. As a result, the backend server does not respond to SYN ACK packets and connections to the client time out. You can increase the upper limit of the backlog by adjusting the **net.core.somaxconn** parameter.

10.3 Load Balancer

10.3.1 How Does ELB Distribute Traffic?

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

Figure 10-1 Load balancing at Layer 4

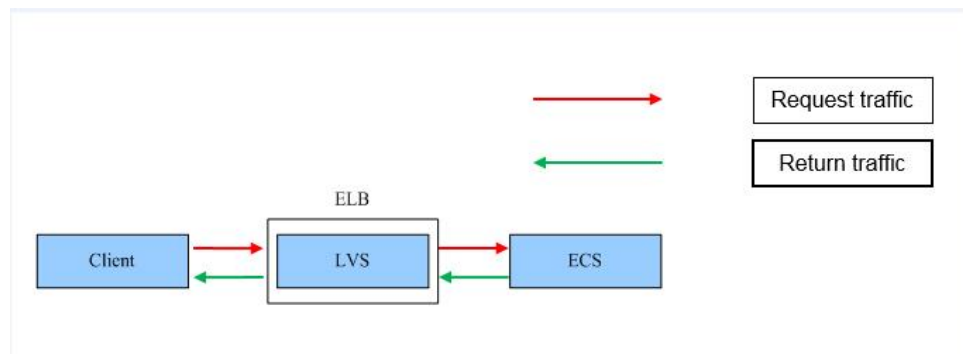
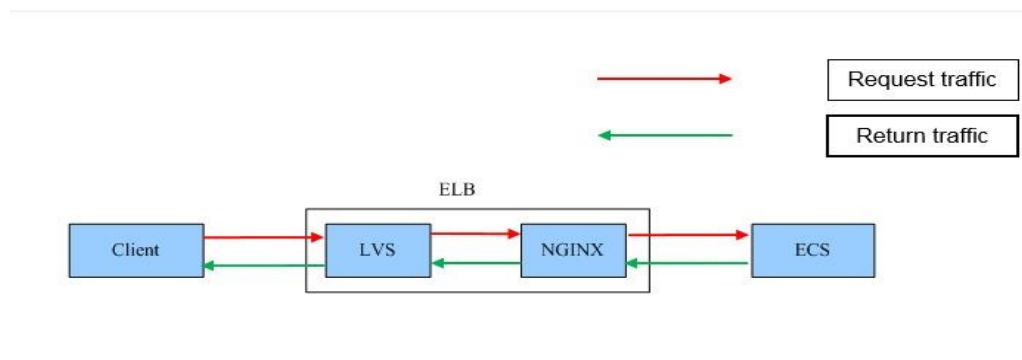


Figure 10-2 Load balancing at Layer 7



10.3.2 How Can I Configure a Public or Private Network Load Balancer?

ELB allows you to create both public and private network load balancers.

When creating a load balancer, you can set its network type. If you select the private network, a private IP address will be assigned to the load balancer, and a

private network load balancer is created by default. If you select the public network, you need to bind an EIP to the load balancer, through which the load balancer can receive requests over the Internet.

10.4 Listener

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

Table 10-1 Sticky sessions supported by load balancers

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

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

10.4.2 How Can ELB Support Multiple Certificates?

Each listener supports only one certificate or certificate chain. If you have multiple certificates or certificate chains, you need to add more listeners.

10.4.3 How Can I Use WebSocket?

No configuration is required. For HTTP listeners, unencrypted WebSocket (`ws://`) is supported by default. For HTTPS listeners, encrypted WebSocket (`wss://`) is supported by default.

10.5 Backend Server

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

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

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

10.5.2 Can Backend Servers Access the Public Network After They Are Associated with a Load Balancer?

Yes. Whether backend servers can access the public network is irrelevant to ELB. If a backend server can access the public network, it can still access the public network after it is associated with a load balancer.

10.5.3 How Can I Check the Network Conditions of a Backend Server?

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

NOTE

For Windows ECSs, run **ipconfig** on the CLI to view their IP addresses.

2. Ping the gateway of the subnet where the ECS resides to check basic network communication.
 - a. On the VPC details page, locate the target subnet and view the gateway address in the **Gateway** column. Generally, the gateway address ends with **.1**.
 - b. Ping the gateway from the ECS. If the gateway cannot be pinged, check the networks at Layer 2 and Layer 3.

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

1. Check whether the security group of the server is correctly configured.
 - a. On the server details page, view the security group.
 - b. Check whether the security group allows access from IP addresses in 100.125.0.0/16. If access is not allowed, add inbound rules for 100.125.0.0/16.
2. Check whether network ACLs of the subnet where the server resides does not intercept the traffic.

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

10.5.5 How Can I Check the Status of a Backend Server?

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

netstat -ntpl

 NOTE

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

Figure 10-3 Port status

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

2. Check the network communication of the ECS.

For example, if the ECS uses port 80, run the **curl** command to check whether the communication is normal.

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

10.5.6 When Is a Backend Server Considered Healthy?

If a backend server is associated with a load balancer for the first time, the backend server is considered healthy after one health check. After this, the server is detected healthy after the maximum retries.

10.6 Health Check

10.6.1 What Should I Do If a Backend Server Is Unhealthy?

Symptom

If a client fails to access a backend server through a load balancer, the backend server is declared unhealthy.

Background

The load balancer uses IP addresses in 100.125.0.0/16 server to send heartbeats to backend servers and check their health. To ensure that health checks can be performed normally, IP addresses in 100.125.0.0/16 must be allowed to access the backend servers.

If a backend server is detected unhealthy, the load balancer will remove this server from the backend server group and stop forwarding traffic to it, until it is declared healthy again.

NOTE

- When a backend server is detected unhealthy, the load balancer will stop routing requests to this server.
- When the health check function is disabled, the load balancer will consider the backend server healthy by default and still route requests to it.
- ELB uses IP addresses in 100.125.0.0/16 to perform health checks and route requests to backend servers.
- Traffic is not routed to a backend server with a weight of 0, and the health check result is meaningless.

Troubleshooting Procedure

Possible causes are sequenced based on their occurrence probability.

If the fault persists after you have ruled out a cause, check other causes.

NOTE

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

Figure 10-4 Troubleshooting process

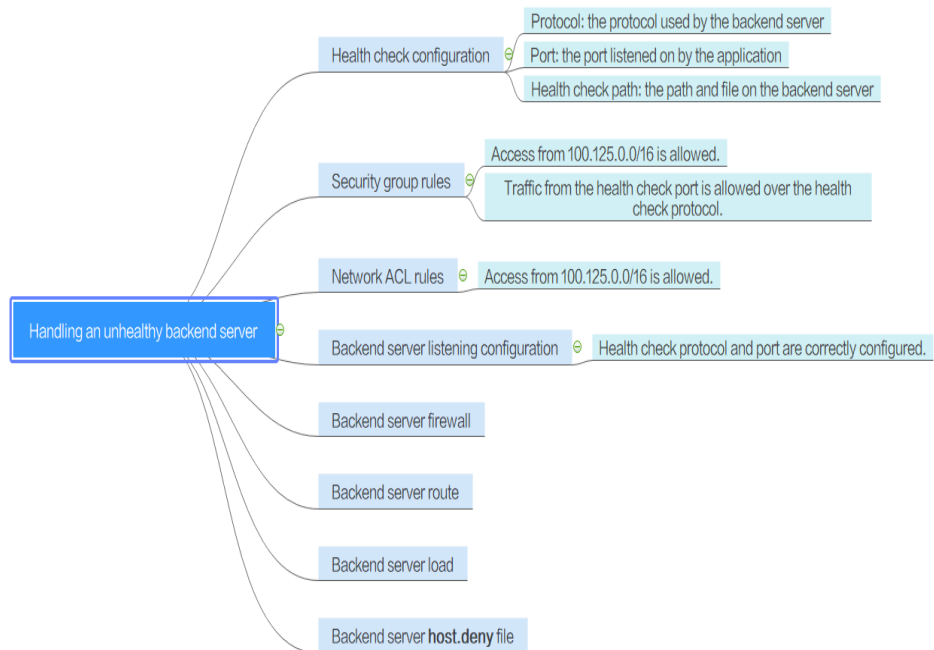


Table 10-2 Troubleshooting process

Possible Cause	Solution
Health check configuration	See Check the Health Check Configuration .
Security group rules	See Check Security Group Rules .
Network ACL rules	See Check Network ACL Rules .
Backend server listening configuration	See Check the Backend Server .
Backend server firewall configuration	See Check the Backend Server Firewall .
Backend server route configuration	See Check the Backend Server Route .
Backend server load	See Check the Backend Server Load .
Backend server host.deny file	See Check the Backend Server host.deny File .

Check the Health Check Configuration

Click the name of the target load balancer to view its details. On the **Backend Server Group** tab page, click the name of the target backend server group. In the

Basic Information area, click **Configure** on the right of **Health Check** and then check the following parameters:

- **Protocol**
- **Port**
- **Check Path** If HTTP is used for health checks, you must check this parameter. A simple static HTML file is recommended.

Check Security Group Rules

- **TCP, HTTP, or HTTPS listeners:** Verify that the inbound rule of the security group containing the backend server allows access from 100.125.0.0/16 and allows the TCP traffic from the health check port.
 - If the health check port is the same as the backend port, the inbound rule must allow traffic from the backend port, for example, 80.
 - If the health check port is different from the backend port, the inbound rule must allow traffic from both the health check port and backend port, for example, 443 and 80.

NOTE

You can check the protocol and port in the basic information area of the backend server group.

Figure 10-5 Example inbound rule

Inbound	IPv4	TCP	Any	100.125.0.0/16
---------	------	-----	-----	----------------

- **UDP listeners:** Verify that the inbound rule of the security group allows traffic from the health check protocol, health check port, and 100.125.0.0/16. In addition, the ICMP traffic must be allowed in the inbound direction.

Figure 10-6 Example inbound rule that allows ICMP traffic

Inbound	IPv4	ICMP	Any	100.125.0.0/16
---------	------	------	-----	----------------

NOTE


- Access to the backend server from IP addresses in 100.125.0.0/16 must be allowed. Load balancers communicate with backend servers using these IP addresses. After traffic is routed to backend servers, source IP addresses are converted to IP addresses starting with 100.125. Besides that, the IP address of the health check node is allocated from 100.125.0.0/16.
- If you are not sure about the security group rules, change the protocol and port range to **All** for testing.
- For UDP listeners, see [10.6.2 What Are the Precautions of Using UDP for Health Checks?](#)

Check Network ACL Rules

A network ACL is an optional subnet-class security configuration. You can associate one or more subnets with a network ACL for controlling traffic in and out of the subnets. Similar to security groups, network ACLs provide access control functions, but add an additional layer of defense to your VPC. Default network ACL rules reject all inbound and outbound traffic. If a network ACL and load

balancer reside in the same subnet, or the network ACL and backend servers associated with the load balancer reside in the same subnet, the load balancer cannot receive traffic from the public or private network, or backend servers become unhealthy.

You can configure an inbound network ACL rule to permit access from 100.125.0.0/16.

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 **Network**, click **Virtual Private Cloud**.
4. In the navigation pane on the left, choose **Network ACLs**.
5. Locate the target network ACL, and click the network ACL name to switch to the network ACL details page.
6. On the **Inbound Rules** or **Outbound Rules** tab page, click **Add Rule** to add an inbound or outbound rule.
 - **Action**: Select **Allow**.
 - **Protocol**: The protocol must be the same as the frontend protocol set when the listener is added.
 - **Source**: Set the value to **100.125.0.0/16**.
 - **Source Port Range**: Select the port range of the service.
 - **Destination**: Enter default value **0.0.0.0/0**, which indicates that traffic from all IP addresses is permitted.
 - **Destination Port Range**: Select the port range of the service.
 - **Description**: provides supplementary information about the network ACL rule. This parameter is optional.
7. Click **OK**.

Check the Backend Server

NOTE

If the backend server runs a Windows OS, use a browser to access **https://Backend server IP address:Health check port**. If a 2xx or 3xx code is returned, the backend server is working properly.

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

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

If the health check port and **LISTEN** are displayed, the backend port is in the listening state. As shown in [Figure 10-7](#), TCP port 880 is listened on.

If no health check port is specified, backend ports are used by default.

Figure 10-7 Backend server port listened on

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

Figure 10-8 Backend server port not listened on

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

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

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

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

TCP listeners: 200

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

Figure 10-9 Unhealthy backend server

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

Figure 10-10 Healthy backend server

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

- If HTTP is used for health checks and the backend server is detected unhealthy, perform the following steps to configure a TCP health check:
On the **Listeners** tab page, modify the target listener, select the backend server group for which TCP health check has been configured, or add a backend server group and select TCP as the health check protocol. After the configuration is complete, wait for a while and check the health check result.

Check the Backend Server Firewall

The firewall or other security protection software on the backend server may mask IP addresses in 100.125.0.0/16. Ensure that access from 100.125.0.0/16 is allowed in the security group containing the backend server.

Check the Backend Server Route

Check whether the default route configured for the primary NIC is manually changed. If the default route is changed, health check packets may fail to reach the backend server.

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

```
ip route
```

Alternatively, run the following command:

```
route -n
```

If the command output does not contain the highlighted route or the IP address to which the route points is not the gateway address of the VPC subnet, change the route to the default one.

Figure 10-11 Example default route pointing to the gateway

```
[root@donatdel.wangfei.iperf ~]# ip route
default via 192.168.2.1 dev eth0 proto dhcp metric 100
169.254.169.254 via 192.168.2.1 dev eth0 proto dhcp metric 100
192.168.2.0/24 dev eth0 proto kernel scope link src 192.168.2.124 metric 100
[root@donatdel.wangfei.iperf ~]#
```

Figure 10-12 Example default route not pointing to the gateway

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

Check the Backend Server Load

Check the workloads of the backend server. If the workloads are high, connections or requests for health checks may time out.

Check the Backend Server `hosts.deny` File

Verify that IP addresses in 100.125.0.0/16 cannot be written to the `/etc/hosts.deny` file on the backend server.

10.6.2 What Are the Precautions of Using UDP for Health Checks?

How UDP Health Checks Work

UDP is a connectionless protocol, which does not establish a three-way handshake before sending data. A UDP health check is implemented as follows:

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

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

When you use UDP for health checks, you are advised to retain the default settings of the parameters.

Troubleshooting Procedure

Use either of the following methods to locate the fault:

1. Check whether the timeout duration is too short.

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

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

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

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

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

The default rate limit is **1000**.

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

The default rate mask is **6168**.

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

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

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

```
man 7 icmp
```

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

NOTE

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

Precautions

Pay attention to the following when you configure UDP health checks:

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

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

```
cat /proc/sys/net/ipv4/icmp_echo_ignore_all
```

- If the returned value is **1**, ICMP is disabled.
- If the returned value is **0**, ICMP is enabled.

- The health check result may be different from the actual health of the backend server.

If the backend server runs a Linux OS, the rate of ICMP packets is limited due to protection against ICMP floods of Linux when there is a large number of concurrent requests. In this case, if a service exception occurs, the load balancer will not receive error message **port XX unreachable** and will still determine that the health check is successful. As a result, there is an inconsistency between the health check result and the actual server health.

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

ELB is deployed in cluster mode, and all nodes for request forwarding in the cluster send requests to backend servers at the same time. If the health check interval is too short, health checks are performed once every few seconds, and a large number of packets are sent to backend servers. To control the frequency of access to backend servers, refer to [5.1 Configuring a Health Check](#).

10.7 Obtaining Source IP Addresses

10.7.1 How Can I Obtain the IP Address of a Client?

Background

- If Network Address Translation (NAT) or Web Application Firewall (WAF) is used, you cannot obtain the IP addresses of the clients.
- If the client is a container, you can obtain only the IP address of the node where the container is located, but cannot obtain the IP address of the container.
- If the function of obtaining client IP addresses is enabled for TCP or UDP listeners, a cloud server cannot be used as a backend server and a client at the same time.

Layer 7 Load Balancing

Configure the application server and obtain the IP address of a client from the HTTP header.

The real IP address is placed in the X-Forwarded-For header by the load balancer in the following format:

```
X-Forwarded-For: IP address of the client,Proxy server 1-IP address,Proxy server 2-IP address,...
```

If you use this method, the first IP address obtained is the IP address of the client.

Apache Server

1. Install Apache 2.4.

For example, if CentOS 7.5 is used as the OS, run the following command to install the software:

```
yum install httpd
```

2. Add the following content to the end of Apache configuration file `/etc/httpd/conf/httpd.conf`:

```
LoadModule remoteip_module modules/mod_remoteip.so
RemoteIPHeader X-Forwarded-For
RemoteIPInternalProxy 100.125.0.0/16
```

Figure 10-13 Content to be added

```
LoadModule remoteip_module modules/mod_remoteip.so
RemoteIPHeader X-Forwarded-For
RemoteIPInternalProxy 100.125.0.0/16
```

NOTE

Set the value of **RemoteIPInternalProxy** to the IP address ranges of the proxy servers, for example, the IP address range used by the AAD service and 100.125.0.0/16 used by ELB. Use a comma (,) to separate every two entries.

3. Change the log output format in the Apache configuration file to the following (**%a** indicates the source IP address):

```
LogFormat "%a %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" combined
```

4. Restart Apache.

```
systemctl restart httpd
```

5. Obtain the actual IP address of the client from the httpd access logs.

Nginx Server

For example, if CentOS 7.5 is used as the OS, run the following command to install the software:

1. Run the following commands to install `http_realip_module`:

```
yum -y install gcc pcre pcre-devel zlib zlib-devel openssl openssl-devel
wget http://nginx.org/download/nginx-1.17.0.tar.gz
tar zxvf nginx-1.17.0.tar.gz
cd nginx-1.17.0
./configure --prefix=/path/server/nginx --with-http_stub_status_module --without-http-cache --with-http_ssl_module --with-http_realip_module
make
make install
```

2. Run the following command to open the `nginx.conf` file:

```
vi /path/server/nginx/conf/nginx.conf
```

3. Add the following content under **http** or **server**:

```
;100.125.0.0/16set_real_ip_from
real_ip_header X-Forwarded-For;
```

Figure 10-14 Content to be added

```
server {
    listen      80;
    server_name localhost;

    set_real_ip_from 100.125.0.0/16;
    real_ip_header X-Forwarded-For;
}
```

NOTE

Set the value of **set_real_ip_from** to the IP address ranges of the proxy servers, for example, the IP address range used by the AAD service and 100.125.0.0/16 used by ELB. Use a comma (,) to separate every two entries.

4. Start Nginx.
`/path/server/nginx/sbin/nginx`
5. Obtain the actual IP address of the client from the Nginx access logs.
`cat /path/server/nginx/logs/access.log`

Tomcat Servers

In the following operations, the Tomcat installation path is **/usr/tomcat/tomcat8/**.

1. Log in to a server on which Tomcat is installed.
2. Check whether Tomcat is running properly.

```
ps -ef|grep tomcat
netstat -anpt|grep java
```

Figure 10-15 Tomcat running properly

```
[root@lilian apache-tomcat-9.0.10]# ps -ef |grep tomcat
root      1009   995  0 15:01 pts/0    00:00:00 grep  --color=auto tomcat
root      32223   1  0 14:37 pts/0    00:00:12 /usr/java/jdk-10.0.1/bin/java -Djava.util.logging.config.file=/usr/local/tomcat-9.0.10/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySize=1024 -Djava.io.tmpdir=/usr/local/tomcat-9.0.10/temp org.apache.catalina.startup.Bootstrap start
[root@lilian apache-tomcat-9.0.10]# netstat -anpt|grep java
tcp        0      0 127.0.0.1:32001      0.0.0.0:*           LISTEN      882/java
tcp6       0      0 :::8020             :::*                LISTEN      32223/java
tcp6       0      0 :::8888             :::*                LISTEN      32223/java
tcp6       0      0 127.0.0.1:8006     :::*                LISTEN      32223/java
tcp6       0      0 10.0.0.20:8888     100.125.134.52:38390 ESTABLISHED 32223/java
tcp6       0      0 127.0.0.1:31001    127.0.0.1:32001     ESTABLISHED 882/java
tcp6       0      0 10.0.0.20:8888     100.125.134.53:57771 ESTABLISHED 32223/java
tcp6       0      0 10.0.0.20:8888     100.125.134.46:62833 ESTABLISHED 32223/java
tcp6       0      0 10.0.0.20:8888     100.125.19.50:59124  ESTABLISHED 32223/java
tcp6       0      0 10.0.0.20:8888     100.125.19.47:49597  ESTABLISHED 32223/java
tcp6       1      0 10.0.0.20:50648    100.125.15.62:80     CLOSE_WAIT  882/java
tcp6       0      0 10.0.0.20:8888     100.125.19.53:27108  ESTABLISHED 32223/java
```

3. Add the following configuration items to the **server.xml** file:
`<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" prefix="localhost_access_log." suffix=".txt" pattern="%{X-FORWARDED-FOR}i %l %u %t %r %s %b %D %q %{User-Agent}i %T" resolveHosts="false" />`

Figure 10-16 Example configuration

```
<!-- Access log processes all example.
Documentation at: /docs/config/valve.html
pattern="{X-Forwarded-For}i %l %u %t &quot;%r&quot; %s %b" />
Note: The pattern used is equivalent to using pattern="common" -->
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
prefix="localhost_access_log." suffix=".txt"
pattern="%h %l %u %t &quot;%r&quot; [%{postdata}r] %s %{Referer}i %{User-Agent}i %T %b" />
</Host>
</Engine>
</Service>
</Server>
```

- Restart the Tomcat service.
cd /usr/tomcat/tomcat8/bin && sh startup.sh

In this command, **/usr/tomcat/tomcat8/** is the Tomcat installation path. Change it based on site requirements.

Figure 10-17 Restarting the Tomcat service

```
[root@ecs-ddef bin]# sh startup.sh
Using CATALINA_BASE:   /usr/tomcat/tomcat8
Using CATALINA_HOME:   /usr/tomcat/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat/tomcat8/temp
Using JRE_HOME:        /usr/java/jdk1.8.0_261
Using CLASSPATH:       /usr/tomcat/tomcat8/bin/bootst
Tomcat started.
```

- View the latest logs.
As highlighted in the following figure, IP addresses that are not in the IP address range starting with 100.125 are the source IP addresses.

```
cat localhost_access_log..2020-09-10.txt
```

In this command, **localhost_access_log..2020-09-10.txt** indicates the log path of the current day. Change it based on site requirements.

Figure 10-18 Querying the source IP address

```
[root@ecs-ddef logs]# cat localhost_access_log..2020-09-10.txt
100.125.24.44 - - [10/Sep/2020:20:35:18 +0800] "GET / HTTP/1.1" [-]
100.125.24.43 - - [10/Sep/2020:20:35:18 +0800] "GET / HTTP/1.1" [-]
100.125.24.42 - - [10/Sep/2020:20:35:23 +0800] "GET / HTTP/1.1" [-]
100.125.24.44 - - [10/Sep/2020:20:35:23 +0800] "GET / HTTP/1.1" [-]
100.125.24.43 - - [10/Sep/2020:20:35:23 +0800] "GET / HTTP/1.1" [-]
10. . . .94 - - [10/Sep/2020:20:50:54 +0800] "GET / HTTP/1.1" [-]
10. . . .94 - - [10/Sep/2020:20:54:46 +0800] "GET / HTTP/1.1" [-]
10. . . .94 - - [10/Sep/2020:21:10:43 +0800] "GET / HTTP/1.1" [-]
10. . . .94 - - [10/Sep/2020:21:12:17 +0800] "GET / HTTP/1.1" [-]
```

Windows Server with IIS Deployed

The following uses Windows Server 2012 with IIS7 as an example to describe how to obtain the source IP address.

- Download and install the Java Runtime Environment (JRE).

<https://www.microsoft.com/en-us/download/details.aspx?id=2299>

2. Download the **F5XForwardedFor.dll** plug-in and copy the plug-ins in the **x86** and **x64** directories to a directory on which IIS has the access permission, for example, **C:\F5XForwardedFor2008**.
3. Open the Server Manager and choose **Modules > Configure Native Modules**.

Figure 10-19 Selecting modules

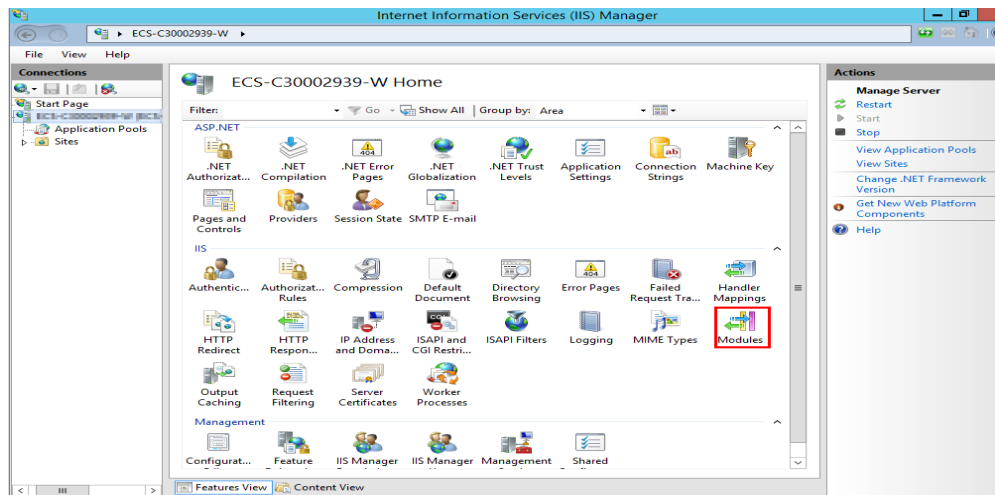
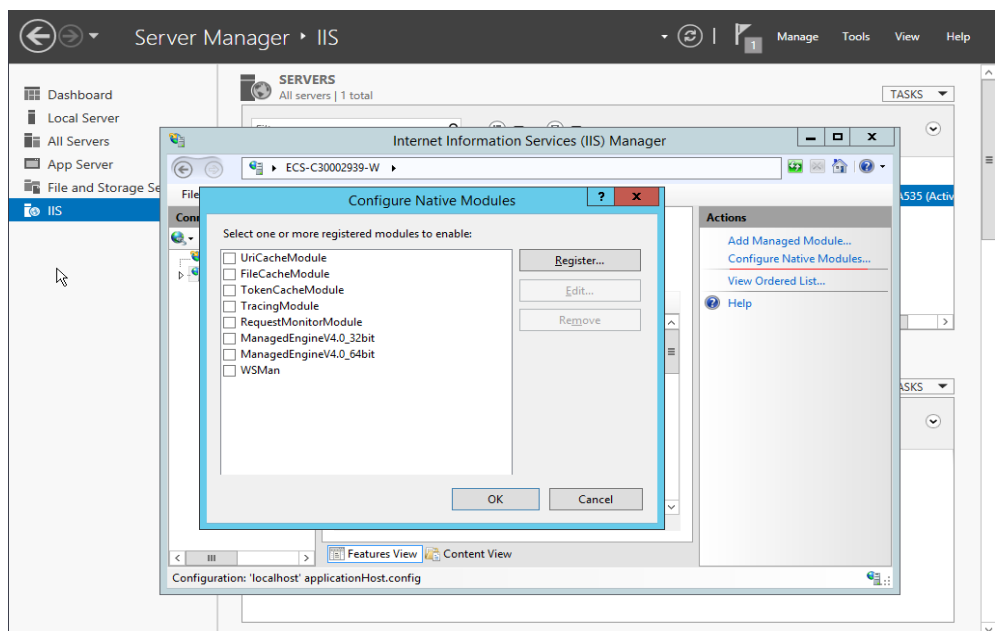
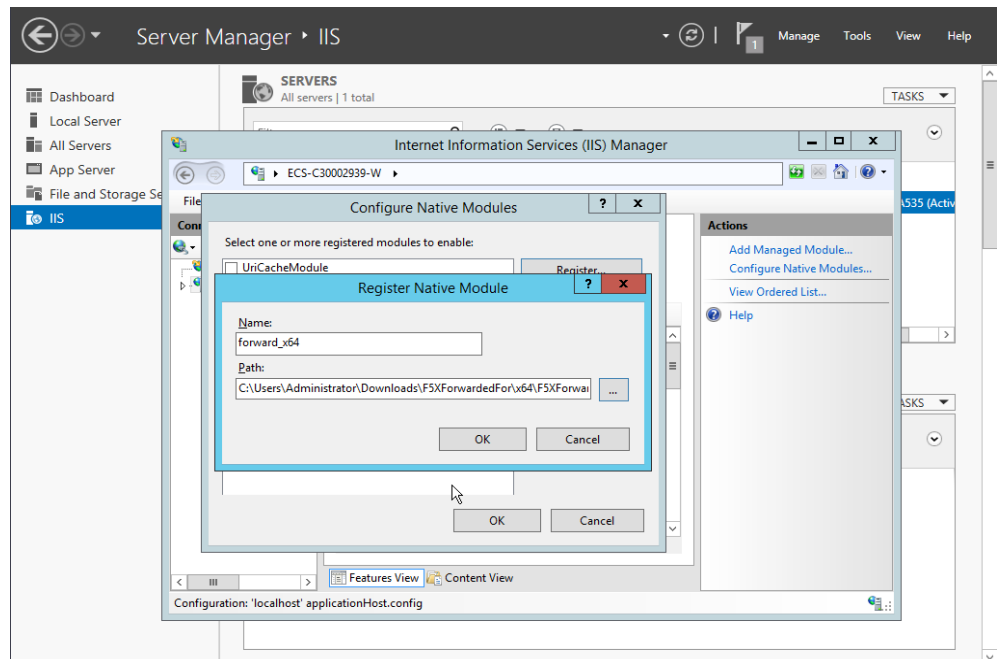


Figure 10-20 Configure Native Modules



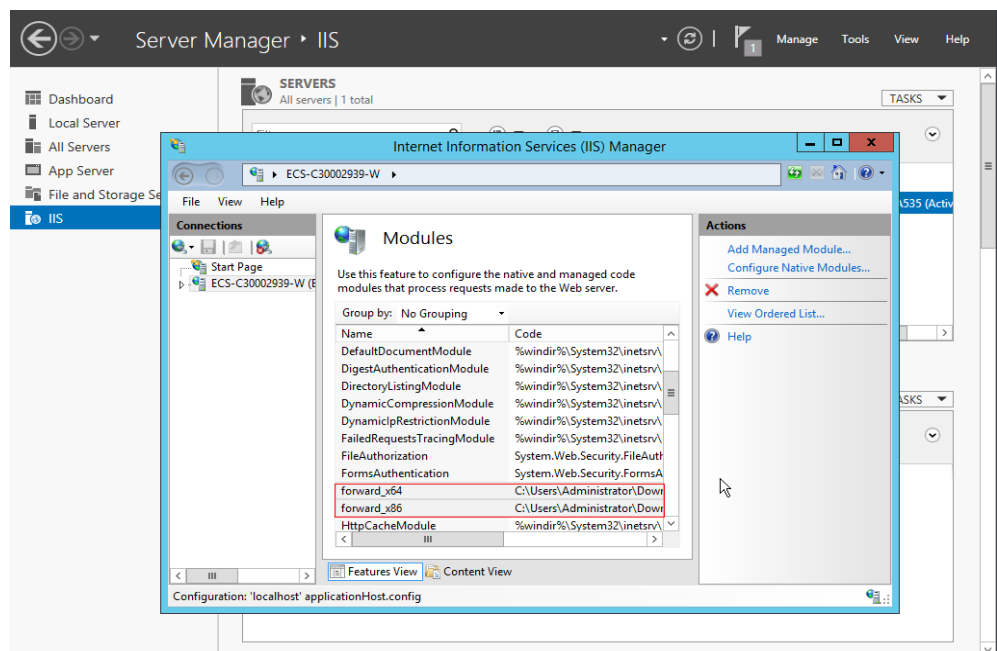
4. Click **Register** to register the x86 and x64 plug-ins.

Figure 10-21 Registering plug-ins



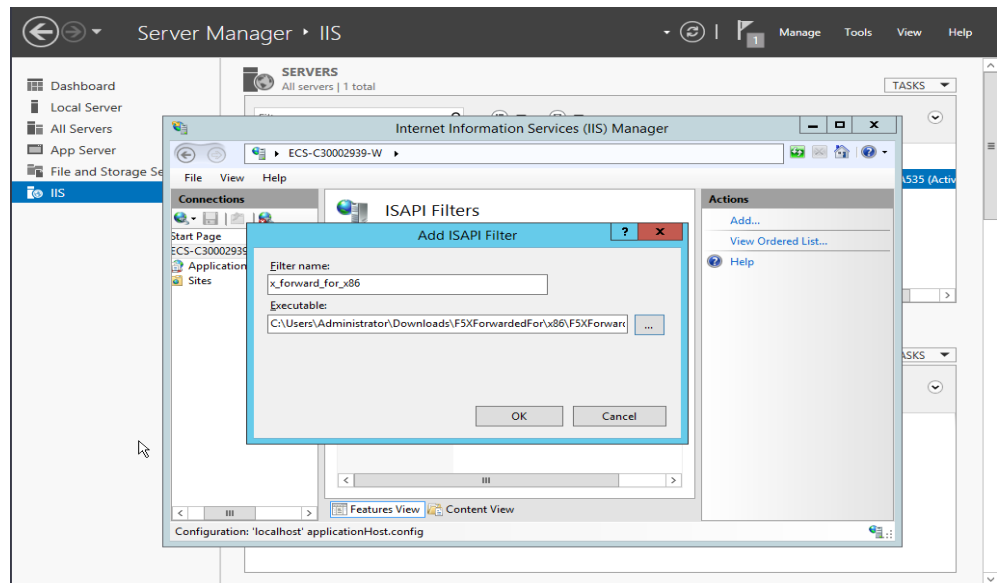
5. In the **Modules** dialog box, verify that the registered plug-ins are displayed in the list.

Figure 10-22 Confirming the registration



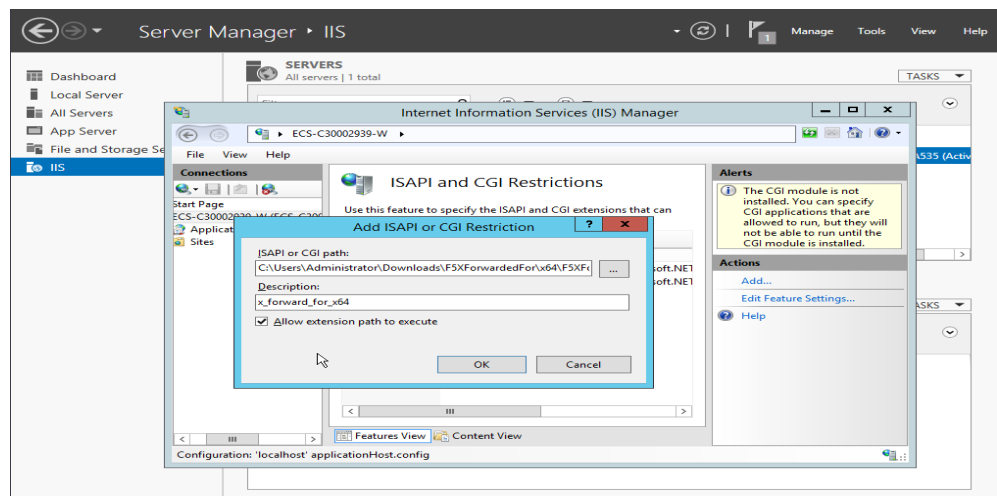
6. Select **ISAPI Filters** on the Server Manager homepage and authorize two plug-ins to run ISAPI and CGI extensions.

Figure 10-23 Adding authorization



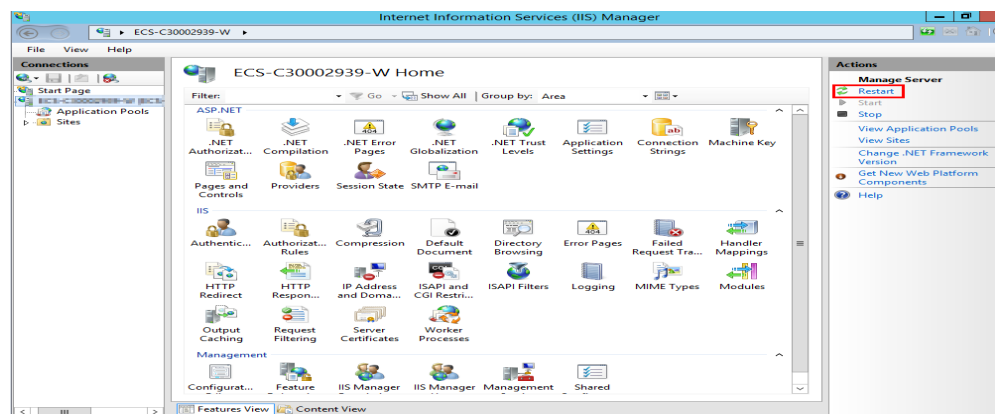
7. Select **ISAPI and CGI Restriction** to set the execution permission for the two plug-ins.

Figure 10-24 Allowing the plug-ins to execute



8. Click **Restart** on the homepage to restart IIS. The configuration takes effect after the restart.

Figure 10-25 Restarting IIS



Layer 4 Load Balancing

TCP listeners require the TOA plug-in to obtain real IP addresses. For details, see [11.1 Configuring the TOA Plug-in](#).

10.8 HTTP/HTTPS Listeners

10.8.1 Why Is the Security Warning Still Displayed After a Certificate Is Configured?

The following may cause the system to display a message indicating that a certificate is insecure:

- The domain name in the certificate is different from the domain name accessed by users. If it is the case, check the domain name in the certificate or create a self-signed certificate.
- SNI is configured, but the specified domain name is different from the one in the certificate.
- The domain name level is inconsistent with the certificate level.

If the problem persists, run the `curl Domain name` command to locate the fault based on the error information returned by the system.

10.9 Sticky Session

10.9.1 What Should I Do If Sticky Sessions Fail to Take Effect?

1. Check whether the sticky session feature is enabled for the backend server group.
2. Check the health check result of the backend server. If the health check result is **Unhealthy**, traffic is routed to other backend servers. As a result, sticky sessions become invalid.
3. If the **Source IP hash** algorithm is selected, check whether the IP address of the request changes before the load balancer receives the request.

4. If an HTTP or HTTPS listener is configured with the sticky session feature enabled, check whether the request carries a cookie. If yes, check whether the cookie value changes (because load balancing at Layer 7 uses cookies to maintain sessions).

10.9.2 What Types of Sticky Sessions Does ELB Support?

ELB supports three types of sticky sessions: source IP address, load balancer cookie, and application cookie.

11 Appendix

11.1 Configuring the TOA Plug-in

Scenarios

ELB provides customized strategies for managing service access. Before customizing these strategies, ELB needs to obtain the client's IP address contained in the access request. To obtain the IP addresses, you can install a 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 7/7.2 (kernel version 3.10.0)
 - Ubuntu 16.04.3 (kernel version 4.4.0)
 - Ubuntu 18.04 (Kernel version 4.15.0)
 - OpenSUSE 42.2 (kernel version 4.4.36)
 - CoreOS 10.10.5 (kernel version 4.9.16)
 - 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.
- VMs 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, you need to download the required module development package from the Internet if it cannot be found in the source.

The following are operations for compiling the module in different Linux OSs. Choose appropriate operations as needed.

- 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 development package header and module library must have the same version as the kernel):

```
sudo yum install kernel-devel-`uname -r`
```

NOTE

During the installation, you need to 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
```

- 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 development 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
- CoreOS
For CoreOS, the module will be compiled in a container, and it must be started before the module is compiled.
For detailed operations, see the CoreOS documentation. Obtain the documentation from the following link:
<https://coreos.com/os/docs/latest/kernel-modules.html>
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 too early. 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. Therefore, you need to compile the module again.

5. Install the module on multiple nodes.

To load the module in the same OSs, copy the **toa.ko** file to VMs 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 node 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 a simple HTTP service 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* can be set to 4.

5. Run the following command to install the module:
make modules_install

The following information is displayed.

Figure 11-1 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 11-2 Installing the kernel

```
INSTALL /lib/firmware/keyspan_pda/xircom_pgs.fw
DEPMOD 2.6.32-toa
[root@SZX1000167219 linux-2.6.32-220.23.1.el6.x86_64.rs]# make install
sh /root/humin/linux-2.6.32-220.23.1.el6.x86_64.rs/arch/x86/boot/install.sh 2.6.32-toa arch/x86/boot/bzImage \
System.map "/boot"
ERROR: modinfo: could not find module xen_procfs
ERROR: modinfo: could not find module ipv6
ERROR: modinfo: could not find module xen_scsi front
ERROR: modinfo: could not find module xen_hcall
ERROR: modinfo: could not find module xen_balloon
[root@SZX1000167219 linux-2.6.32-220.23.1.el6.x86_64.rs]#
```

7. Open the **/boot/grub/grub.conf** file and configure the kernel to start up when the system starts.
 - a. Change the default startup kernel from the first kernel to the zeroth kernel by changing **default=1** to **default=0**.
 - b. Add the **nohz=off** parameter to the end of the line containing the **vmlinuz-2.6.32-toa** kernel. If **nohz** is not disabled, the CPU0 usage may be high, causing uneven stress.

Figure 11-3 Configuration file

```
default=1
timeout=5
splashimage=(hd0,1)/boot/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.32-toa)
    root (hd0,1)
    kernel /boot/vmlinuz-2.6.32-toa ro root=UUID:
et nohz=off
    initrd /boot/initramfs-2.6.32-toa.img
```

- c. Save the modification and exit. Restart the OS.
During the restart, the system will load the **vmlinuz-2.6.32-toa** kernel.
8. After the restart, run the following command to load the module:
modprobe toa

You are advised to add the **modprobe toa** command to both the startup script and the system scheduled monitoring script.

Figure 11-4 Adding the **modprobe toa** command

```
[root@SZX1000167219 ~]# modprobe toa
[root@SZX1000167219 ~]# lsmod |grep toa
toa                4203  0
[root@SZX1000167219 ~]#
```

After the module is loaded, query the kernel information.

Figure 11-5 Querying the kernel

```
[root@SZX1000167219 ~]# uname -a
Linux SZX1000167219 2.6.32-toa #1 SMP Sat Oct 15 11:50:05 CST 2016 x86_64 x86_64 x86_64 GNU/Linux
```

9. Verify the module.

After the module is successfully installed, the source address can be directly obtained. The following provides an example for verification.

Run the following command to start a simple HTTP service 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.

12 Change History

Released On	Description
2020-07-30	This issue is the first official release.