ALIBABA CLOUD

Alibaba Cloud

Server Load Balancer
Listeners

Document Version: 20220628

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
⚠ Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	⚠ **Danger:** Resetting will result in the loss of user configuration data.
🔔 **Warning**	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	🔔 **Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
🔈 **Notice**	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	🔈 **Notice:** If the weight is set to 0, the server no longer receives new requests.
ⓘ **Note**	A note indicates supplemental instructions, best practices, tips, and other content.	ⓘ **Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings> Network> Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Listener overview

This topic provides an overview of listeners. After you create a Classic Load Balancer (CLB) instance, you must configure one or more listeners for it. A listener checks for connection requests and then distributes the requests to backend servers based on the forwarding rules that are defined by a specified scheduling algorithm.

CLB provides Layer 4 (TCP or UDP) and Layer 7 (HTTP or HTTPS) listeners. The following table lists the features and use cases of these listeners.

Protocol	Description	Scenario
TCP	• A connection-oriented protocol. A reliable connection must be established before data can be sent and received. • Session persistence is based on source IP addresses. • Source IP addresses are visible at the network layer. • Data is transmitted at a fast rate.	• Applicable to scenarios that require high reliability and data accuracy but can tolerate low speeds, such as file transmission, sending or receiving emails, and remote logons. • Web applications that do not have custom requirements. For more information, see Add a TCP listener.
UDP	• A connectionless protocol. UDP transmits data packets directly instead of making a three-way handshake with the other party before UDP sends data. UDP does not provide error recovery or data re-transmission. • Fast data transmission but relatively low reliability.	Applicable to scenarios where real-time transmission is more important than reliability, such as video chats and real-time financial market pushes. For more information, see Add a UDP listener.
HTTP	• An application-layer protocol that is used to package data. • Cookie-based session persistence. • Use the X-Forward-For header to obtain the real IP addresses of clients.	Applicable to scenarios that require data content identification, such as web applications and small mobile games. For more information, see Add an HTTP listener.
HTTPS	• Encrypted data transmission that prevents unauthorized access. • Centralized certificate management service. You can upload certificates to CLB. The decryption operations are completed directly on CLB.	Applicable to scenarios that require encrypted transmission. For more information, see Add an HTTPS listener.

2.Add a TCP listener

This topic describes how to add a TCP listener to a Classic Load Balancer (CLB) instance. TCP allows you to transmit data in a reliable and accurate manner but at relatively low speeds. Therefore, you can use TCP to transfer files, send or receive emails, and perform remote logons. You can add a TCP listener to forward TCP requests.

Prerequisites

A instance is created. For more information, see Create a CLB instance.

Step 1: Configure a TCP listener

1.
2. In the top navigation bar, select the region where the instance is deployed.
3. Use one of the following methods to open the listener configuration wizard:
   - On the **Instances** page, find the CLB instance and click **Configure Listener** in the **Actions** column.
   - On the **Instances** page, find the CLB instance that you want to manage and click the ID of the instance. On the **Listener** tab, click **Add Listener**.
4. Set the following parameters and click **Next**.

Parameter	Description
Select Listener Protocol	Select **TCP**.
Listening Port	Set the listening port that is used to receive requests and forward them to backend servers. Valid values: 1 to 65535.
Listener Name	Specify a name for the listener.
Advanced	Click **Modify** to configure advanced settings.

Parameter	Description
Scheduling Algorithm	Select a scheduling algorithm. ○ **Weighted Round-Robin (WRR)**: Backend servers that have higher weights receive more requests than backend servers that have lower weights. ○ **Round-Robin (RR)**: Requests are distributed to backend servers in sequence. ○ **Consistent Hash (CH)**: ■ **Tuple**: specifies consistent hashing that is based on four factors: source IP address, destination IP address, source port, and destination port. Requests that contain the same information based on the four factors are distributed to the same backend server. ■ **Source IP**: specifies consistent hashing that is based on source IP addresses. Requests from the same source IP address are distributed to the same backend server. **Note** Only high-performance CLB instances support the CH algorithm.
Enable Session Persistence	Specify whether to enable session persistence. After session persistence is enabled, CLB forwards all requests from a client to the same backend server. For TCP listeners, session persistence is implemented based on IP addresses. Requests from the same IP address are forwarded to the same backend server.
Enable Access Control	Specify whether to enable access control. Select an access control method after you enable access control. Then, select an access control list (ACL) that is used as the whitelist or blacklist of the listener. **Note** IPv6 instances can be associated only with IPv6 ACLs, while IPv4 instances can be associated only with IPv4 ACLs. For more information, see Create an access control list.

Parameter	Description
Enable Peak Bandwidth Limit	Specify whether to set the bandwidth limit of the listener. If a CLB instance is billed based on bandwidth usage, you can set different maximum bandwidth values for different listeners. This limits the amount of traffic that flows through each listener. The sum of the maximum bandwidth values of all listeners that are added to a CLB instance cannot exceed the maximum bandwidth value of the CLB instance. By default, this feature is disabled and all listeners share the bandwidth of the CLB instance. ? **Note** If a CLB instance is billed based on data transfer, the bandwidth of its listeners is not limited by default.
Idle Timeout	Specify the timeout period of idle TCP connections. Unit: seconds. Valid values: 10 to 900.
Proxy Protocol	Use the proxy protocol to pass client IP addresses to backend servers. ? **Note** You cannot enable this feature in scenarios where PrivateLink is used.
Obtain Client Source IP Address	Specify whether to retrieve the real IP addresses of clients. Only Layer 4 listeners support this feature. By default, this feature is enabled.
Automatically Enable Listener After Creation	Specify whether to immediately enable the listener after it is created. By default, listeners are enabled after they are created.

Step 2: Add backend servers

After you configure the listener, you must add backend servers to process client requests. You can use the default server group that is configured for the CLB instance. You can also configure a vServer group or a primary/secondary server group, or enable the primary/secondary mode for the listener. For more information, see Backend server overview.

1. On the **Backend Servers** wizard page, select the type of the server group to which requests are forwarded. The default server group is used in this example.

   Select **Default Server Group** and click **Add More**.

2. In the **My Servers** panel, select the ECS instances that you want to add as backend servers and click **Next**.

3. On the **Configure Ports and Weights** wizard page, specify the weights of the backend servers that you added. A backend server with a higher weight receives more requests.

   > ? **Note** If the weight of a backend server is set to 0, no request is distributed to the backend server.

4. Click **Add**. On the Default Server Group tab, specify the ports that you want to open on the backend servers to receive requests. The backend servers are the ECS instances that you selected.

Valid values: 1 to 65535.

You can specify the same port on different backend servers that are added to a CLB instance.

5. Click **Next**.

Step 3: Configure health checks

CLB performs health checks to check the availability of the ECS instances that serve as backend servers. The health check feature improves overall service availability and reduces the impact of backend server failures.

On the **Health Check** wizard page, click **Modify** to modify the health check configurations. For more information, see Configure health checks.

Step 4: Submit the configurations

1. On the **Confirm** wizard page, check the configurations. You can click **Modify** to modify the configurations.
2. After you confirm the configurations, click **Submit**.
3. When Configuration Successful appears, click **OK**.

   After you configure the listener, you can view the listener on the Listener tab.

3.Add a UDP listener

This topic describes how to add a UDP listener to a Classic Load Balancer (CLB) instance. UDP applies to services that prioritize real-time content delivery over reliability, such as video conferencing and real-time quote services. You can add a UDP listener to forward UDP packets.

Context

Before you configure a UDP listener, take note of the following items:

- You cannot specify ports 250, 4789, or 4790 for UDP listeners. They are system reserved ports.
- Fragmentation is not supported.
- You cannot view source IP addresses by using the UDP listeners of a CLB instance in the classic network.
- The following operations take effect 5 minutes after they are performed on a UDP listener:
  - Remove backend servers.
  - Set the weight of a backend server to 0 after it is detected unhealthy.
- IPv6 packets have longer IP headers than IPv4 packets. When an IPv6 instance uses a UDP listener, make sure that the following requirement is met: The maximum transmission unit (MTU) supported by the network interface controller (NIC) that each backend server uses to communicate with does not exceed 1,200 bytes. Otherwise, oversized packets may be discarded. You must modify the MTU setting in the configuration files of some applications accordingly.

  TCP supports Maximum Segment Size (MSS) announcement. Therefore, when you use a TCP, HTTP, or HTTPS listener, you do not need to perform additional configurations.

Prerequisites

A CLB instance is created. For more information, see Create an SLB instance.

Step 1: Configure a UDP listener

1.
2. In the top navigation bar, select the region where the instance is deployed.
3. Use one of the following methods to open the listener configuration wizard:
   - On the **Instances** page, find the CLB instance and click **Configure Listener** in the **Actions** column.
   - On the **Instances** page, find the CLB instance that you want to manage and click the ID of the instance. On the **Listeners** tab, click **Add Listener**.
4. Set the following parameters and click **Next**.

Parameter	Description
Select Listener Protocol	Select **UDP**.

Parameter	Description
Listening Port	Set the listening port that is used to receive requests and forward them to backend servers. Valid values: 1 to 65535. You can set a TCP or UDP listener to listen on all ports within a specified port range. **Note** This feature is available for only users in a whitelist. To use this feature, .
Listener Name	Specify a name for the listener.
Advanced	Click **Modify** to configure advanced settings.
Scheduling Algorithm	Select a scheduling algorithm. ○ **Weighted Round-Robin (WRR)**: Backend servers that have higher weights receive more requests than backend servers that have lower weights. ○ **Round-Robin (RR)**: Requests are distributed to backend servers in sequence. ○ **Consistent Hash (CH)**: ■ **Source IP**: specifies consistent hashing that is based on source IP addresses. Requests from the same source IP address are distributed to the same backend server. ■ **Tuple**: specifies consistent hashing that is based on four factors: source IP address, destination IP address, source port, and destination port. Requests that contain the same information based on the four factors are distributed to the same backend server. ■ **QUIC ID**: specifies consistent hashing that is based on Quick UDP Internet Connections (QUIC) IDs. Requests that contain the same QUIC ID are distributed to the same backend server. **Notice** QUIC is implemented based on draft-ietf-quic-transport-10 and is rapidly evolving. Therefore, compatibility is not guaranteed for all QUIC versions. We recommend that you perform tests before you apply the protocol to a production environment.
Enable Session Persistence	Specify whether to enable session persistence. After session persistence is enabled, the listener forwards all requests from the same client to the same backend server.

Parameter	Description
Enable Access Control	Specify whether to enable access control. Select an access control method after you enable access control. Then, select an access control list (ACL) that is used as the whitelist or blacklist of the listener. ⓘ **Note** IPv6 instances can be associated only with IPv6 ACLs, while IPv4 instances can be associated only with IPv4 ACLs. For more information, see Create an access control list.
Enable Connection Draining	After connection draining is enabled, connections to backend servers can function as expected during the specified timeout period after the backend servers are removed or fail health checks. ⓘ **Note** This feature is available for only users in a whitelist. To use this feature, .
Enable Peak Bandwidth Limit	Specify whether to set a maximum bandwidth value for the listener. If a CLB instance is billed based on bandwidth usage, you can set different maximum bandwidth values for different listeners. This limits the amount of traffic that flows through each listener. The sum of the maximum bandwidth values of all listeners that are added to a CLB instance cannot exceed the maximum bandwidth value of the CLB instance. By default, this feature is disabled and all listeners share the bandwidth of the CLB instance. ⓘ **Note** If a CLB instance is billed based on data transfer, the bandwidth of its listeners is not limited by default.
Proxy Protocol	Use the proxy protocol to pass client IP addresses to backend servers. ⓘ **Note** You cannot enable this feature in scenarios where PrivateLink is used.
Obtain Client Source IP Address	Specify whether to reserve the real IP addresses of clients. Only Layer 4 listeners support this feature. By default, this feature is enabled. ⓘ **Note** You cannot view source IP addresses by using the UDP listeners of a CLB instance in the classic network. To obtain source IP addresses, enable **Proxy Protocol**.

Parameter	Description
Automatically Enable Listener After Creation	Specify whether to immediately enable the listener after it is created. By default, this feature is enabled.

Step 2: Add backend servers

After you configure the listener, you must add backend servers to process client requests. You can use the default server group that is configured for the CLB instance. You can also configure a vServer group or a primary/secondary server group, or enable the primary/secondary mode for the listener. For more information, see Backend server overview.

1. On the **Backend Servers** wizard page, select the type of the server group to which requests are forwarded. The default server group is used in this example.

   Select **Default Server Group** and click **Add More**.

2. In the **My Servers** panel, select the ECS instances that you want to add as backend servers and click **Next**.

3. On the **Configure Ports and Weights** wizard page, specify the weights of the backend servers that you added. A backend server with a higher weight receives more requests.

   > **Note** If the weight of a backend server is set to 0, no request is distributed to the backend server.

4. Click **Add**. On the Default Server Group tab, specify the ports that you want to open on the backend servers to receive requests. The backend servers are the ECS instances that you selected. Valid values: 1 to 65535.

   You can specify the same port on different backend servers that are added to a CLB instance.

5. Click **Next**.

Step 3: Configure health checks

CLB performs health checks to check the availability of the ECS instances that serve as backend servers. The health check feature improves overall service availability and reduces the impact of backend server failures.

On the **Health Check** wizard page, click **Modify** to modify the health check configurations. For more information, see Configure health checks.

Step 4: Submit the configurations

1. On the **Confirm** wizard page, check the configurations. You can click **Modify** to modify the configurations.

2. After you confirm the configurations, click **Submit**.

3. When Configuration Successful appears, click **OK**.

   After you configure the listener, you can view the listener on the Listener tab.

Related information

References

- Configure health checks

- Add a default backend server
- Create a vServer group
- Create a primary/secondary server group
- Overview
- CreateLoadBalancerUDPListener

4.Add an HTTP listener

This topic describes how to add an HTTP listener to a Classic Load Balancer (CLB) instance. HTTP is applicable to applications that must identify data from different sources, such as web applications and mobile games. You can add HTTP listeners to forward HTTP requests.

Prerequisites

A instance is created. For more information, see Create a CLB instance.

Step 1: Configure an HTTP listener

1.
2. Select the region where the CLB instance is deployed.
3. Use one of the following methods to start the listener configuration wizard:
   - On the **Instances** page, find the CLB instance and click **Configure Listener** in the **Actions** column.
   - On the **Instances** page, click the ID of the CLB instance that you want to manage. On the **Listener** tab, click **Add Listener**.
4. Set the following parameters and click **Next**.

Parameter	Description
Select Listener Protocol	Select the protocol of the listener. **HTTP** is selected in this example.
Backend Protocol	In this topic, **HTTP** is used, and **Backend Protocol** is set to **HTTP**.
Listening Port	Specify the listener port that is used to receive requests and forward them to backend servers. Valid values: 1 to 65535. HTTP uses port 80.
Listener Name	Specify a name for the custom listener. The description must be 1 to 256 characters in length, and can contain letters, digits, hyphens (-), forward slashes (/), periods (.),and underscores (_).
Advanced	Click **Modify** to configure advanced settings.
Scheduling Algorithm	Select a scheduling algorithm. ○ **Weighted Round-Robin (WRR)**: Backend servers that have higher weights receive more requests than backend servers that have lower weights. ○ **Round-Robin (RR)**: Requests are distributed to backend servers in sequence.

Parameter	Description
Redirection	Specify whether to redirect traffic from the HTTP listener to an HTTPS listener. **Note** Before you enable redirection, make sure that an HTTPS listener is created.
Enable Session Persistence	Specify whether to enable session persistence. After session persistence is enabled, CLB forwards all requests from the same client to the same backend server. CLB persists HTTP sessions based on cookies. CLB allows you to use the following methods to process cookies: ○ **Insert cookie**: If you select this option, you only need to specify the timeout period of the cookie. CLB inserts a cookie (SERVERID) into the first HTTP or HTTPS response that is sent to a client. The next request from the client will contain this cookie, and the listener will forward this request to the recorded backend server. ○ **Rewrite cookie**: If you select this option, you can specify the cookie that you want to insert into an HTTP or HTTPS response. You must specify the timeout period and the lifetime of a cookie on a backend server. After you specify a cookie, CLB overwrites the original cookie with the specified cookie. The next time CLB receives a client request that carries the specified cookie, the listener distributes the request to the recorded backend server.
Enable Access Control	Specify whether to enable access control. Select an access control method after you enable access control. Then, select an access control list (ACL) that is used as the whitelist or blacklist of the listener. **Note** IPv6 instances can be associated only with IPv6 network ACLs, and IPv4 instances can be associated only with IPv4 network ACLs. For more information, see Create an access control list.

Parameter	Description
Enable Peak Bandwidth Limit	Specify whether to set a bandwidth limit for the listener. Unit: Mbit/s. Valid values: 0 to 5120. If a CLB instance is billed based on bandwidth usage, you can set different maximum bandwidth values for different listeners. This limits the amount of traffic that flows through each listener. The sum of the maximum bandwidth values of all listeners that are added to a CLB instance cannot exceed the maximum bandwidth value of the CLB instance. By default, this feature is disabled and all listeners share the bandwidth of the CLB instance. **Note** If a CLB instance is billed based on data transfer, the bandwidth of its listeners is not limited by default.
Idle Timeout	Specify the timeout period of idle connections. Unit: seconds. Valid values: 1 to 60. If no request is received within the specified timeout period, CLB closes the current connection. CLB creates a new connection when a new connection request is received.
Request Timeout	Specify the request timeout period. Unit: seconds. Valid values: 1 to 180. If no response is received from the backend server within the request timeout period, CLB returns an HTTP 504 error to the client.
Enable Gzip Compression	If you enable Gzip compression, files of specific types are compressed. If you disable Gzip compression, no file is compressed. Gzip supports the following file types: `text/xml` , `text/plain` , `text/css` , `application/javascript` , `application/x-javascript` , `application/rss+xml` , `application/atom+xml` , and `application/xml` .

Parameter	Description
Add HTTP Header Fields	You can add the following HTTP header fields: ◦ `X-Forwarded-For` : Add the header field to retrieve the real IP address of the client. ◦ `SLB-ID` : Add the header field to retrieve the ID of the CLB instance. ◦ `SLB-IP` : Add the header field to retrieve the IP address of the CLB instance. ◦ `X-Forwarded-Proto` : Add the header field to retrieve the listener protocol used by the CLB instance. ◦ `X-Forwarded-Port` : Add the header field to retrieve the listener ports of the CLB instance. ◦ `X-Forwarded-Client-srcport` : Add the header field to retrieve the port over which a client communicates with the CLB instance.
Obtain Client Source IP Address	Specify whether to retrieve the client IP address. By default, this feature is enabled.
Automatically Enable Listener After Creation	Specify whether to immediately enable the listener after it is created. By default, listeners are enabled after they are created.
WAF Protection	Specify whether to enable Web Application Firewall (WAF) protection.

Step 2: Add backend servers

After you configure the listener, you must add backend servers to process client requests. You can add backend servers to the default server group, or create a vServer group or primary/secondary server group, and then add servers to the server group. For more information, see Backend server overview.

Backend servers are added to the default server group in this example.

1. On the **Backend Servers** wizard page, select **Default Server Group**. Then, click **Add More**.
2. In the **My Servers** panel, select the Elastic Compute Service (ECS) instances that you want to add as backend servers and click **Next**.
3. Set weights for the selected ECS instances in the **Weight** column.

> Note
>
> - An ECS instance with a higher weight receives more requests. The default weight is 100. You can click **Reset** to set the **weight** to the default value.
> - If you set the weight of a server to 0, the server does not receive requests.

4. Click **Add**. On the Default Server Group tab, specify the ports that you want to open on the backend servers to receive requests. Valid values: 1 to 65535. Click **Next**.

   You can specify the same port on different backend servers that are added to a CLB instance.

Step 3: Configure health checks

CLB performs health checks to check the availability of the ECS instances that serve as backend servers. The health check feature improves overall service availability and reduces the impact of backend server failures.

On the **Health Check** wizard page, click **Modify** to modify the health check configurations. For more information, see Configure health checks.

Step 4: Submit the configurations

1. On the **Confirm** wizard page, check the configurations. You can click **Modify** to modify the configurations.
2. After you confirm the configurations, click **Submit**.
3. When Configuration Successful appears, click **OK**.

   After you configure the listener, you can view the listener on the Listener tab.

Related information

References

- Add a default backend server
- Configure health checks
- Create a vServer group
- Create a primary/secondary server group
- Overview
- Forward requests based on domain names or URLs
- Manage a domain name extension
- CreateLoadBalancerHTTPSListener: creates an HTTPS listener.

5.Add an HTTPS listener

This topic describes how to add an HTTPS listener to a Classic Load Balancer (CLB) instance. HTTPS is intended for applications that require encrypted data transmission. You can add an HTTPS listener to forward HTTPS requests.

Prerequisites

A instance is created. For more information, see Create a CLB instance.

Step 1: Configure a UDP listener

1.
2. Select the region where the CLB instance is deployed.
3. Use one of the following methods to open the listener configuration wizard:
   - On the **Instances** page, find the CLB instance that you want to manage and click **Configure Listener** in the **Actions** column.
   - On the **Instances** page, click the ID of the CLB instance that you want to manage. On the **Listener** tab, click **Add Listener**.
4. Set the following parameters and click **Next**.

Parameter	Description
Select Listener Protocol	Select the protocol type of the listener. In this example, **HTTPS** is selected.
Listening Port	Set the listening port that is used to receive requests and forward them to backend servers. Valid values: 1 to 65535.
Listener Name	Enter a name for the listener.
Advanced	Click **Modify** to configure advanced settings.
Scheduling Algorithm	Select a scheduling algorithm. ○ **Weighted Round-Robin (WRR)**: Backend servers that have higher weights receive more requests than backend servers that have lower weights. ○ **Round-Robin (RR)**: Requests are distributed to backend servers in sequence.

Parameter	Description
Enable Session Persistence	Specify whether to enable session persistence. After session persistence is enabled, CLB forwards all requests from the same client to the same backend server. HTTP session persistence is implemented through cookies. CLB allows you to use the following methods to process cookies: ○ **Insert cookie**: If you select this option, you only need to specify the timeout period of the cookie. CLB inserts a cookie (SERVERID) into the first HTTP or HTTPS response that is sent to a client. The next request from the client will contain this cookie, and the listener will forward this request to the recorded backend server. ○ **Rewrite cookie**: If you select this option, you can specify the cookie that you want to insert into an HTTP or HTTPS response. You must specify the timeout period and the lifetime of a cookie on a backend server. After you specify a cookie, CLB overwrites the original cookie with the specified cookie. The next time CLB receives a client request that carries the specified cookie, the listener distributes the request to the recorded backend server.
Enable HTTP/2	Select whether to enable HTTP/2 for the frontend protocol of the CLB instance.
Enable Access Control	Specify whether to enable access control. Select an access control method after you enable access control. Then, select an access control list (ACL) that is used as the whitelist or blacklist of the listener. ○ **Whitelist: Only requests from the IP addresses or CIDR blocks in the specified ACL are forwarded.** Whitelists apply to scenarios in which you want to allow only specific IP addresses to access an application. Your business may be adversely affected if the whitelist is not set properly. After a whitelist is configured, only IP addresses in the whitelist can access the CLB listener. If you enable a whitelist but the whitelist does not contain an IP address, the listener forwards all requests. ○ **Blacklist**: Requests from the IP addresses or CIDR blocks in the specified ACL are blocked. You can choose this option if you want to block requests from specified IP addresses. If you enable a blacklist but the blacklist does not contain an IP address, the CLB listener forwards all requests. **Note** IPv6 instances can be associated with only IPv6 ACLs, while IPv4 instances can be associated only with IPv4 ACLs. For more information, see Create an access control list.

Parameter	Description
Enable Peak Bandwidth Limit	Specify whether to set a maximum bandwidth value for the listener. If a CLB instance is billed based on bandwidth usage, you can set different maximum bandwidth values for different listeners. This limits the amount of traffic that flows through each listener. The sum of the maximum bandwidth values of all listeners that are added to a CLB instance cannot exceed the maximum bandwidth value of the CLB instance. By default, this feature is disabled and all listeners share the bandwidth of the CLB instance. **Note** If a CLB instance is billed based on data transfer, the bandwidth of its listeners is not limited by default.
Idle Timeout	Specify the timeout period of idle connections. Unit: seconds. Valid values: 1 to 60. If no request is received within the specified timeout period, CLB closes the current connection. CLB creates a new connection when a new connection request is received.
Request Timeout	Specify the request timeout period. Unit: seconds. Valid values: 1 to 180. If no response is received from the backend server within the request timeout period, CLB returns an HTTP 504 error to the client.
Enable Gzip Compression	Specify whether to enable Gzip compression to compress specific types of files. Gzip supports the following file types: text/xml, text/plain, text/css, application/javascript, application/x-javascript, application/rss+xml, application/atom+xml, and application/xml.
Add HTTP Header Fields	Select the custom HTTP header field that you want to add.
Obtain Client Source IP Address	Specify whether to retrieve the real IP addresses of clients. By default, this feature is enabled.
Automatically Enable Listener After Creation	Specify whether to immediately enable the listener after it is created. By default, this feature is enabled.

Step 2: Configure an SSL certificate

When you add an HTTPS listener, you must upload a server certificate or certification authority (CA) certificate and select a TLS security policy, as shown in the following table.

Certificate	Description	Required for one-way authentication	Required for mutual authentication
Server certificate	The certificate that is used to identify the server. Your browser uses the server certificate to check whether the certificate sent by the server is signed and issued by a trusted CA.	Yes You must upload the server certificate to the certificate management system of CLB.	Yes You must upload the server certificate to the certificate management system of CLB.
Client certificate	The certificate that is used to identify the client. The server identifies the client by checking the certificate sent by the client. You can sign a client certificate with a self-signed CA certificate.	No	Yes You must install the client certificate on the client.
CA certificate	The server uses a CA certificate to verify the signature on the client certificate. If the signature is invalid, the connection request is denied.	No	Yes You must upload the CA certificate to the certificate management system of CLB.
TLS security policy	TLS security policies are supported only by high-performance CLB instances. A TLS security policy contains TLS protocol versions and cipher suites that are available for HTTPS. For more information, see Manage TLS security policies.	Yes	Yes

Before you upload a certificate, take note of the following rules:

- CLB supports the following public key algorithms: RSA 1024, RSA 2048, RSA 4096, ECDSA P-256, ECDSA P-384, and ECDSA P-521.
- The certificate that you want to upload must be in the PEM format.
- After you upload a certificate to CLB, CLB can manage the certificate. You do not need to bind the certificate to backend servers.
- It may take a few minutes to upload, load, and verify the certificate. Therefore, an HTTPS listener is not enabled immediately after it is created. It takes about 1 to 3 minutes to enable an HTTPS listener.
- The ECDHE cipher suite used by HTTPS listeners supports forward secrecy. It does not support the security enhancement parameters that are required by the DHE cipher suite. Therefore, you cannot upload certificates (PEM files) that contain the `BEGIN DH PARAMETERS` field. For more information, see Certificate requirements.
- HTTPS listeners do not support Server Name Indication (SNI). You can choose TCP listeners and configure SNI on backend servers.
- By default, the timeout period of session tickets for HTTPS listeners is 300 seconds.

- The actual amount of data transfer on an HTTPS listener is larger than the billed amount because a portion of data is used for handshaking.
- If a large number of connections are established, a large amount of data is used for handshaking.

1. On the **SSL Certificates** wizard page, select the server certificate that you uploaded. You can also click **Create Server Certificate** to upload a server certificate.
2. To enable mutual authentication or configure a TLS security policy, click **Modify** next to **Advanced**.
3. Enable mutual authentication, and select an uploaded CA certificate. You can also create a CA certificate.
4. For more information about TLS security policies, see Manage TLS security policies.

Step 3: Add backend servers

After configuring the listener, you must add backend servers to process client requests. You can use the default server group that is configured for the CLB instance. You can also create a vServer group or a primary/secondary server group. For more information, see Backend server overview.

The default server group is selected in this example.

Step 4: Configure health checks

CLB performs health checks to check the availability of the ECS instances that serve as backend servers. The health check feature improves overall service availability and reduces the impact of backend server failures.

Step 5: Submit the configurations

1. On the **Confirm** wizard page, check the configurations. You can click **Modify** to modify the configurations.
2. After you confirm the configurations, click **Submit**.
3. When Configuration Successful appears, click **OK**.

   After you configure the listener, you can view the listener on the Listener tab.

Related information

References

- Add a default backend server
- Create a vServer group
- Create a primary/secondary server group
- Overview
- Forward requests based on domain names or URLs
- Add an additional certificate
- CreateLoadBalancerHTTPSListener

6.Domain name extensions

6.1. Manage a domain name extension

HTTPS listeners of guaranteed-performance Server Load Balancer (SLB) instances support configuring multiple certificates, allowing you to forward requests with different domain names to different backend servers.

Introduction to SNI

Server Name Indication (SNI) is an extension to the SSL/TLS protocol, allowing a server to install multiple SSL certificates on the same IP address. When a client accesses SLB, the certificate configured for the domain name is used by default. If no certificate is configured for the domain name, the certificate configured for the HTTPS listener is used.

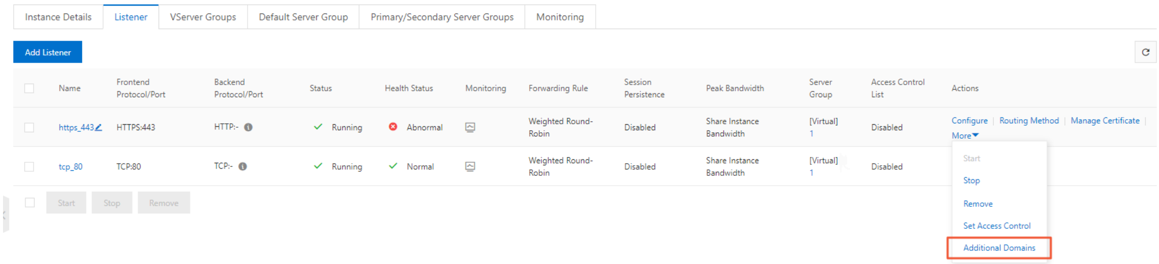
> **Note** Only guaranteed-performance SLB instances support SNI.

If you want to resolve multiple domain names to the IP address of an SLB instance, distribute requests from different domains to different backend servers, and at the same time use HTTPS encrypted access, you can use the domain name extension function.

The domain name extension function is available in all regions.

Add a domain name extension

1. Log on to the SLB console.
2. Select the region of the target SLB instance.
3. Find the target SLB instance and click the instance ID.
4. Click the **Listeners** tab.
5. On the **Listeners** tab page, find the target HTTPS listener, and choose **More > Additional Domains** in the **Actions** column.



□	Name	Frontend Protocol/Port	Backend Protocol/Port	Status	Health Status	Monitoring	Forwarding Rule	Session Persistence	Peak Bandwidth	Server Group	Access Control List	Actions
□	https_443	HTTPS:443	HTTP:-	✓ Running	☒ Abnormal		Weighted Round-Robin	Disabled	Share Instance Bandwidth	[Virtual] 1	Disabled	Configure | Routing Method | Manage Certificate | More
□	tcp_80	TCP:80	TCP:-	✓ Running	✓ Normal		Weighted Round-Robin	Disabled	Share Instance Bandwidth	[Virtual] 1	Disabled	

□ Start Stop Remove

6. Click **Add Additional Domain** and configure the domain name:

i. Enter a domain name. The domain name can only contain letters, numbers, hyphens (-), and periods (.), and must start with a letter or a number. To check if the domain name you enter is valid, you can use the Alibaba Cloud domain name check tool.

Domain name-based forwarding rules support exact match and wildcard match.

- Exact domain name: www.aliyun.com
- Wildcard domain name (generic domain name): *.aliyun.com, *.market.aliyun.com

  When a request matches multiple forwarding rules, exact match takes precedence over small-scale wildcard match and small-scale wildcard match takes precedence over large-scale wildcard match, as shown in the following table.

Type	Request URL	Request URL		
		www.aliyun.com	*.aliyun.com	*.market.aliyun.com
Exact match	www.aliyun.com	✓	×	×
Wildcard match	market.aliyun.com	×	✓	×
Wildcard match	info.market.aliyun.com	×	×	✓

ii. Select the certificate associated with the domain name.

> ⓘ **Note** The domain name in the certificate must be the same as the added domain name extension.

iii. Click **OK**.

7. On the **Listeners** page, find the target HTTPS listener and click **Add Forwarding Rules** in the **Actions** column.
8. On the **Add Forwarding Rules** page, configure the forwarding rule and click **Add Forwarding Rules**.
9. For more information, see Forward requests based on domain names or URLs.

> ⓘ **Note** Make sure that the domain name configured in the forwarding rule is the same as the added domain name extension.

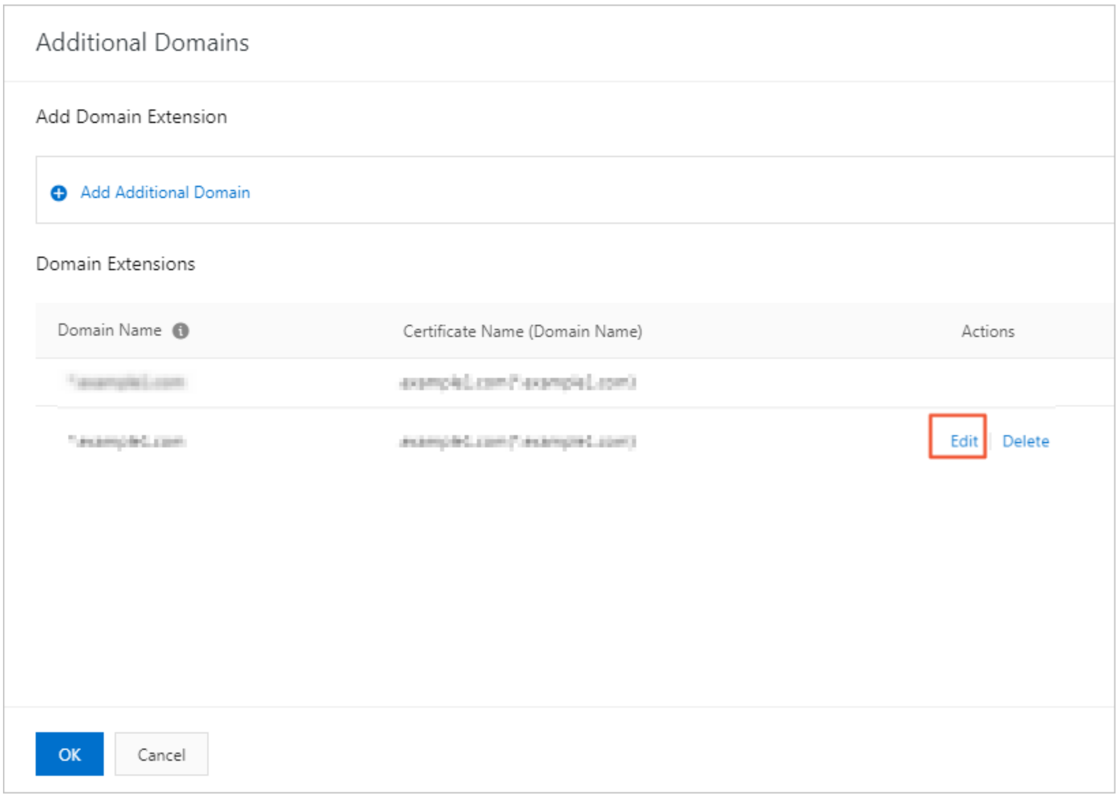
Edit a domain name extension

You can replace the certificate used by an added domain name extension.

To edit a domain name extension, follow these steps:

1. Log on to the SLB console.
2. Select the region of the target SLB instance.
3. Find the target SLB instance and click the instance ID.
4. Click the **Listeners** tab.
5. On the **Listeners** page, find the created HTTPS listener, and then choose **More > Additional Domains** in the **Actions** column.

6. Find the target domain name extension and then click **Edit**.
7. In the **Edit Additional Domain** dialog box, select a new certificate and then click **OK**.



Additional Domains

Add Domain Extension

Add Additional Domain

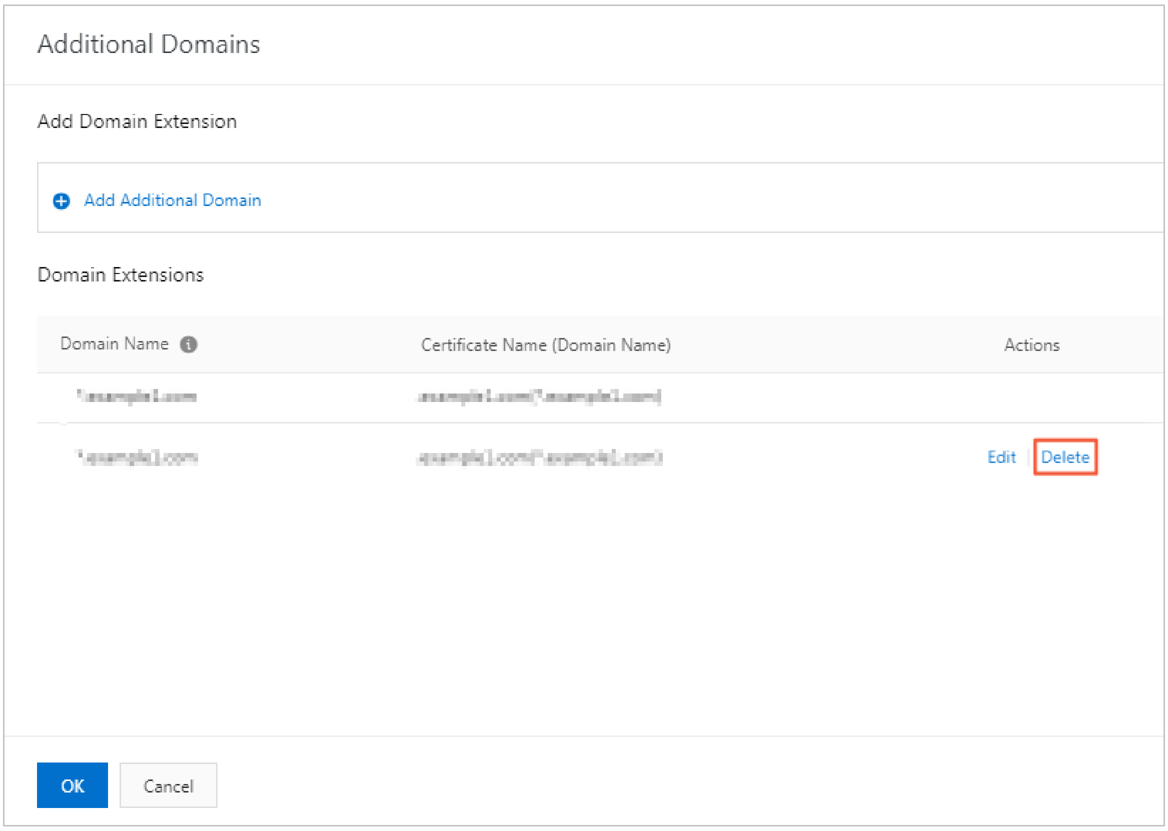
Domain Extensions

Domain Name	Certificate Name (Domain Name)	Actions
[redacted]	[redacted]	
[redacted]	[redacted]	Edit Delete

Delete a domain name extension

To delete a domain name extension, follow these steps:

1. Log on to the SLB console.
2. Select the region of the target SLB instance.
3. Find the target SLB instance and click the instance ID.
4. Click the **Listeners** tab.
5. On the **Listeners** page, find the created HTTPS listener, and then choose **More > Additional Domains** in the **Actions** column.
6. Find the target domain name extension and click **Delete**.



Additional Domains

Add Domain Extension

Domain Extensions

Domain Name	Certificate Name (Domain Name)	Actions
*.example1.com	example1.com(*.example1.com)	
*.example1.com	example1.com(*.example1.com)	Edit | Delete

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

6.2. Add an additional certificate

This topic describes how to add an additional certificate to a listener of a Classic Load Balancer (CLB) instance.

Procedure

1.
2. On the **Instances** page, click the ID of the CLB instance that you want to manage.
3. On the **Listener** tab, find the HTTPS listener that you created and choose ⋮ > **Manage Additional Certificate** in the **Actions** column.
4. In the **Manage Additional Certificate** panel, click **Add Additional Certificate**.

i. Enter a domain name. The domain name can contain only letters, digits, hyphens (-), and periods (.). It must start with a letter or digit.

   Domain name-based forwarding rules support exact matching and wildcard matching.

   - You can specify a specific domain name such as www.aliyun.com in a forwarding rule.
   - You can also specify a wildcard domain name such as *.aliyun.com or *.market.aliyun.com in a forwarding rule.

     If a request matches multiple domain-based forwarding rules, exact matching prevails over wildcard matching. If multiple wildcard domain name are matched, the higher-level wildcard domain name prevails over the lower-level wildcard domain name. The following table describes the priorities of domain name-based forwarding rules.

Type	Request URL	Domain name-based forwarding rule		
		www.aliyundoc.com	*.aliyundoc.com	*.market.aliyundoc.com
Exact matching	www.aliyun.com	√	×	×
Wildcard matching	market.aliyun.com	×	√	×
Wildcard matching	info.market.aliyun.com	×	×	√

ii. Select the certificate that is associated with the domain name.

> ? Note
>
> - The domain name of the certificate must be the same as that of the additional certificate.
> - If multiple wildcard certificates are added, only the first wildcard certificate can be matched.

iii. Click **OK**.

   After you add an additional certificate, you must create a forwarding rule based on the domain name associated with the additional certificate. Otherwise, the additional certificate is invalid.

5. (Optional)To create a forwarding rule, perform the following operations:

   i. On the **Note** page, click **Configure Rule**. You can also click the ID of the CLB instance on the **Instances** page and click the **Listeners** tab.

   ii. Find the HTTPS listener that you want to manage and click **Set Forwarding Rule** in the Actions column.

   iii. In the **Add Forwarding Rules** panel, click **Add Forwarding Rules**.

iv. Set the parameters of the forwarding rule.

For more information, see Forward requests based on domain names or URLs.

> **Note** The domain name that you specify in the forwarding rule must be the same as the domain name associated with the additional certificate.

Related information

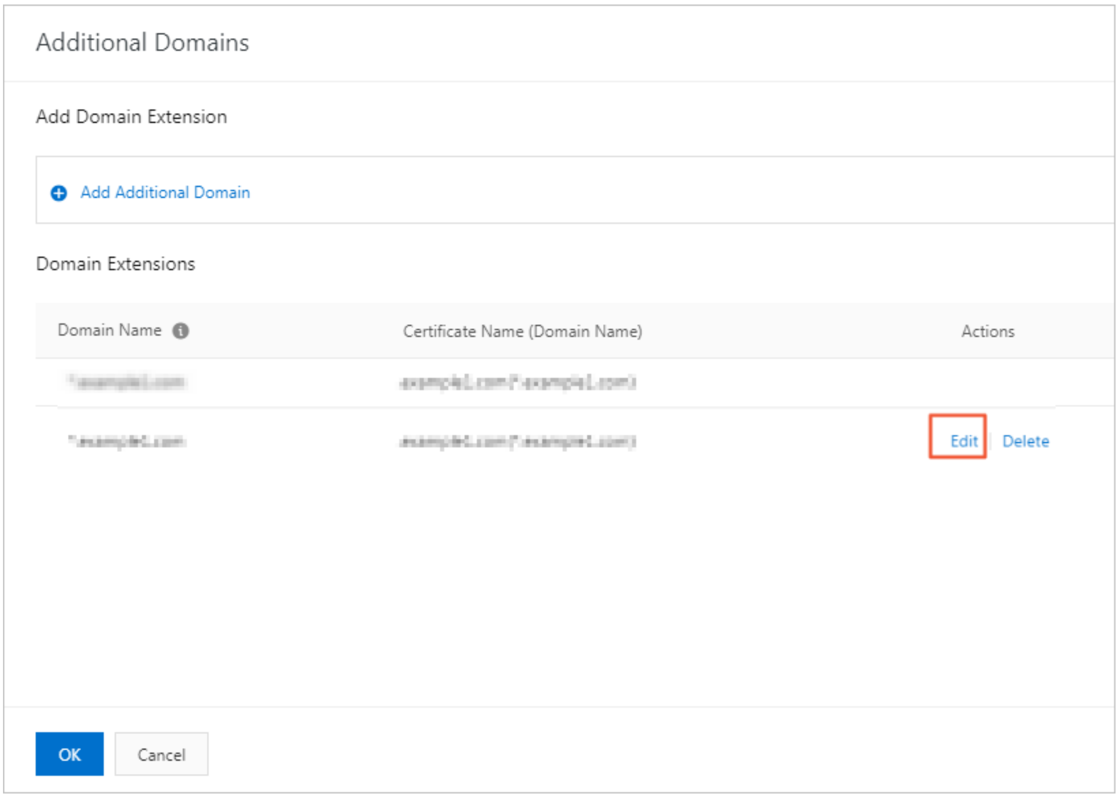
- CreateDomainExtension

6.3. Replace an additional certificate

This topic describes how to replace an existing additional certificate.

Procedure

1.
2.
3. In the left-side navigation pane, choose **Instances > Instances**.
4. On the **Instances** page, find the Classic Load Balancer (CLB) instance that you want to manage and click its instance ID.
5. On the **Listener** tab, find the HTTPS listener that you created and choose ⋮ > **Manage Additional Certificate** in the **Actions** column.
6. Find an additional certificate, and then click **Edit** in the **Actions** column.



Additional Domains

Add Domain Extension

Domain Extensions

Domain Name	Certificate Name (Domain Name)	Actions
[redacted]	[redacted]	
[redacted]	[redacted]	Edit Delete

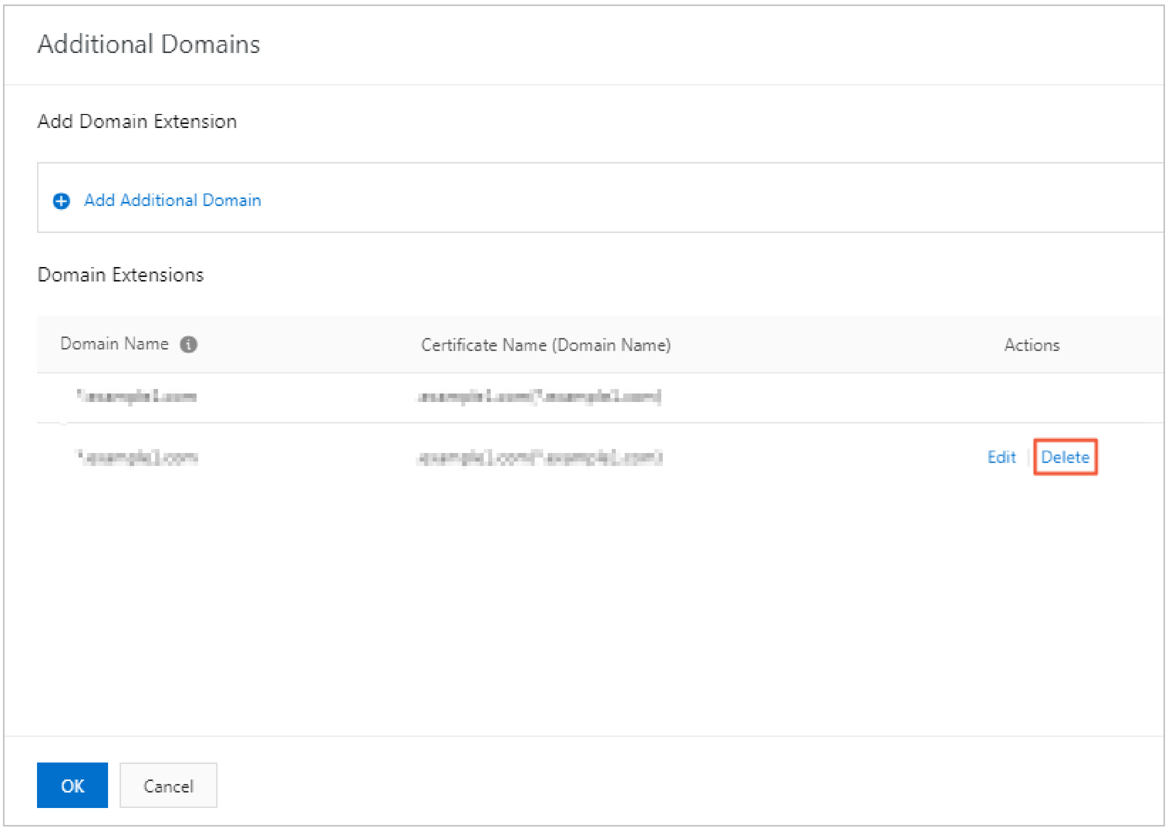
7. In the **Modify Additional Certificate** dialog box, select a new certificate and then click **OK**.

6.4. Delete an additional certificate

You can delete additional certificates that are no longer needed.

Procedure

1.
2.
3. In the left-side navigation pane, choose **Instances > Instances**.
4. On the **Instances** page, find the SLB instance from which you want to delete an additional certificate and click its instance ID.
5. On the **Listener** tab, find the HTTPS listener, and then choose ⋮ > **Manage Additional Certificate** in the Actions column.
6. In the **Manage Additional Certificate** panel, click **Delete** in the Actions column corresponding to an additional certificate.



Additional Domains

Add Domain Extension

⊕ Add Additional Domain

Domain Extensions

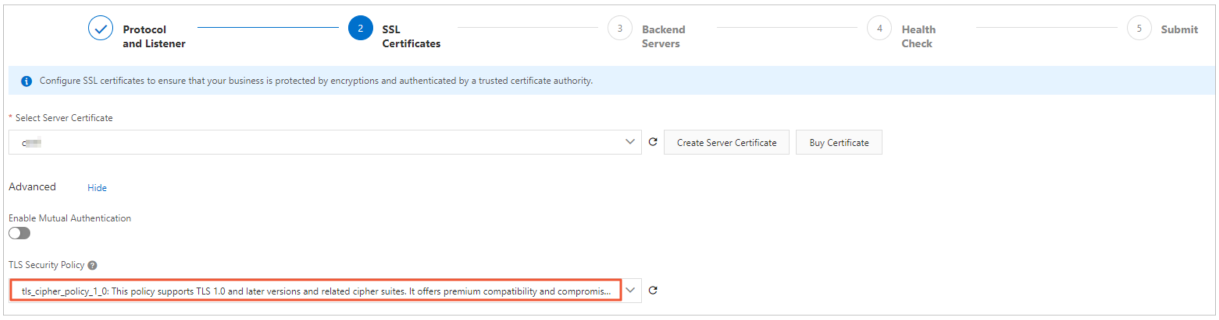
Domain Name ⓘ	Certificate Name (Domain Name)	Actions
[redacted]	[redacted]	
[redacted]	[redacted]	Edit | Delete

7. In the dialog box that appears, click **OK**.

7.Manage TLS security policies

When you create or configure an HTTPS listener for a guaranteed-performance server load balancer (SLB) instance, you can select from a variety of TLS security policies.

When you create or configure an HTTPS listener, you can modify the advanced settings in the **SSL Certificates** step and select a TLS security policy. For more information, see Add an HTTPS listener.



Configure SSL certificates to ensure that your business is protected by encryptions and authenticated by a trusted certificate authority.

* Select Server Certificate: c[redacted]

Advanced

Enable Mutual Authentication

TLS Security Policy: tls_cipher_policy_1_0: This policy supports TLS 1.0 and later versions and related cipher suites. It offers premium compatibility and compromis...

A TLS security policy contains available TLS protocols and supported cipher suites.

TLS security policies

Security policy	Feature	Supported TLS version	Supported cipher suite
tls_cipher_policy_1_0	Provides optimal compatibility and basic security	TLSv1.0, TLSv1.1, and TLSv1.2	ECDHE-ECDSA-AES128-GCM-SHA256, ECDHE-ECDSA-AES256-GCM-SHA384, ECDHE-ECDSA-AES128-SHA256, ECDHE-ECDSA-AES256-SHA384, ECDHE-RSA-AES128-GCM-SHA256, ECDHE-RSA-AES256-GCM-SHA384, ECDHE-RSA-AES128-SHA256, ECDHE-RSA-AES256-SHA384, AES128-GCM-SHA256, AES256-GCM-SHA384, AES128-SHA256, AES256-SHA256, ECDHE-ECDSA-AES128-SHA, ECDHE-ECDSA-AES256-SHA, ECDHE-RSA-AES128-SHA, ECDHE-RSA-AES256-SHA, AES128-SHA, AES256-SHA, and DES-CBC3-SHA
tls_cipher_policy_1_1	Provides high compatibility and standard security	TLSv1.1 and TLSv1.2	ECDHE-ECDSA-AES128-GCM-SHA256, ECDHE-ECDSA-AES256-GCM-SHA384, ECDHE-ECDSA-AES128-SHA256, ECDHE-ECDSA-AES256-SHA384, ECDHE-RSA-AES128-GCM-SHA256, ECDHE-RSA-AES256-GCM-SHA384, ECDHE-RSA-AES128-SHA256, ECDHE-RSA-AES256-SHA384, AES128-GCM-SHA256, AES256-GCM-SHA384, AES128-SHA256, AES256-SHA256, ECDHE-ECDSA-AES128-SHA, ECDHE-ECDSA-AES256-SHA, ECDHE-RSA-AES128-SHA, ECDHE-RSA-AES256-SHA, AES128-SHA, AES256-SHA, and DES-CBC3-SHA

Security policy	Feature	Supported TLS version	Supported cipher suite
tls_cipher_policy_1_2	Provides high compatibility and advanced security	TLSv1.2	ECDHE-ECDSA-AES128-GCM-SHA256, ECDHE-ECDSA-AES256-GCM-SHA384, ECDHE-ECDSA-AES128-SHA256, ECDHE-ECDSA-AES256-SHA384, ECDHE-RSA-AES128-GCM-SHA256, ECDHE-RSA-AES256-GCM-SHA384, ECDHE-RSA-AES128-SHA256, ECDHE-RSA-AES256-SHA384, AES128-GCM-SHA256, AES256-GCM-SHA384, AES128-SHA256,AES256-SHA256, ECDHE-ECDSA-AES128-SHA, ECDHE-ECDSA-AES256-SHA, ECDHE-RSA-AES128-SHA, ECDHE-RSA-AES256-SHA, AES128-SHA, AES256-SHA, and DES-CBC3-SHA
tls_cipher_policy_1_2_strict	Supports only perfect forward secrecy (PFS) cipher suites and provides premium security.	TLSv1.2	ECDHE-ECDSA-AES128-GCM-SHA256, ECDHE-ECDSA-AES256-GCM-SHA384, ECDHE-ECDSA-AES128-SHA256, ECDHE-ECDSA-AES256-SHA384, ECDHE-RSA-AES128-GCM-SHA256, ECDHE-RSA-AES256-GCM-SHA384, ECDHE-RSA-AES128-SHA256, ECDHE-RSA-AES256-SHA384, ECDHE-ECDSA-AES128-SHA, ECDHE-ECDSA-AES256-SHA, ECDHE-RSA-AES128-SHA, and ECDHE-RSA-AES256-SHA

Security policy	Feature	Supported TLS version	Supported cipher suite
tls_cipher_policy_1_2_strict_with_1_3 Note TLSv1.3 is supported in the following regions: • UK (London) • China (Qingdao) • China (Hohhot) • China (Chengdu) • Singapore • Japan (Tokyo) • India (Mumbai) • Australia (Sydney) • Malaysia (Kuala Lumpur) • US (Silicon Valley) • US (Virginia) • Germany (Frankfurt) • UAE (Dubai)	Supports only perfect forward secrecy (PFS) cipher suites and provides premium security.	TLSv1.2 and TLSv1.3	TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384, TLS_CHACHA20_POLY1305_SHA256, TLS_AES_128_CCM_SHA256, TLS_AES_128_CCM_8_SHA256, ECDHE-ECDSA-AES128-GCM-SHA256, ECDHE-ECDSA-AES256-GCM-SHA384, ECDHE-ECDSA-AES128-SHA256, ECDHE-ECDSA-AES256-SHA384, ECDHE-RSA-AES128-GCM-SHA256, ECDHE-RSA-AES256-GCM-SHA384, ECDHE-RSA-AES128-SHA256, ECDHE-RSA-AES256-SHA384, ECDHE-ECDSA-AES128-SHA, ECDHE-ECDSA-AES256-SHA, ECDHE-RSA-AES128-SHA, and ECDHE-RSA-AES256-SHA

Cipher suites supported by different TLS security policies

Security policy		tls_cipher_policy_1_0	tls_cipher_policy_1_1	tls_cipher_policy_1_2	tls_cipher_policy_1_2_strict	tls_cipher_policy_1_2_strict_with_1_3
TLS	-	1.2/1.1/1.0	1.2/1.1	1.2	1.2	1.2 and 1.3
	ECDHE-RSA-AES128-GCM-SHA256	✔	✔	✔	✔	✔
	ECDHE-RSA-AES256-GCM-SHA384	✔	✔	✔	✔	✔
	ECDHE-RSA-AES128-SHA256	✔	✔	✔	✔	✔

Security policy		tls_cipher_policy_1_0	tls_cipher_policy_1_1	tls_cipher_policy_1_2	tls_cipher_policy_1_2_strict	tls_cipher_policy_1_2_strict_with_1_3
CIPHER	ECDHE-RSA-AES256-SHA384	✔	✔	✔	✔	✔
	AES128-GCM-SHA256	✔	✔	✔	-	-
	AES256-GCM-SHA384	✔	✔	✔	-	-
	AES128-SHA256	✔	✔	✔	-	-
	AES256-SHA256	✔	✔	✔	-	-
	ECDHE-RSA-AES128-SHA	✔	✔	✔	✔	✔
	ECDHE-RSA-AES256-SHA	✔	✔	✔	✔	✔
	AES128-SHA	✔	✔	✔	-	-
	AES256-SHA	✔	✔	✔	-	-
	DES-CBC3-SHA	✔	✔	✔	-	-
	TLS_AES_128_GCM_SHA256	-	-	-	-	✔
	TLS_AES_256_GCM_SHA384	-	-	-	-	✔
	TLS_CHACHA20_POLY1305_SHA256	-	-	-	-	✔
	TLS_AES_128_CCM_SHA256	-	-	-	-	✔
	TLS_AES_128_CCM_8_SHA256	-	-	-	-	✔
	ECDHE-ECDSA-AES128-GCM-SHA256	-	-	-	-	✔
	ECDHE-ECDSA-AES256-GCM-SHA384	-	-	-	-	✔

Security policy		tls_cipher_policy_1_0	tls_cipher_policy_1_1	tls_cipher_policy_1_2	tls_cipher_policy_1_2_strict	tls_cipher_policy_1_2_strict_with_1_3
	ECDHE-ECDSA-AES128-SHA256	-	-	-	-	✔
	ECDHE-ECDSA-AES256-SHA384	-	-	-	-	✔
	ECDHE-ECDSA-AES128-SHA	-	-	-	-	✔
	ECDHE-ECDSA-AES256-SHA	-	-	-	-	✔

8.Redirect HTTP requests to HTTPS

This topic describes how to change configurations to redirect HTTP requests to HTTPS. HTTPS is the secure version of HTTP. Server Load Balancer (SLB) allows you to redirect HTTP requests to HTTPS, which facilitates whole-site HTTPS deployment. You can redirect HTTP requests to HTTPS in all regions, and customize the status code for redirection in the UK (London) region.

Prerequisites

An HTTPS listener is created. For more information, see Add an HTTPS listener.

Context

This feature is available only when you create an HTTP listener in the new SLB console.

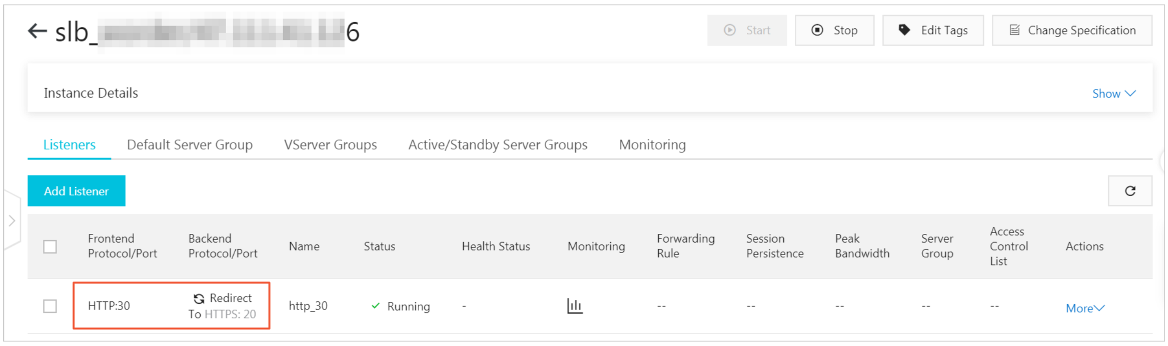
Procedure

1. Log on to the SLB console.
2. In the left-side navigation pane, choose **Instances > Instances**.
3. On the **Instances** page, click the ID of an SLB instance.
4. On the **Listener** tab. Click **Add Listener**.
5. In the **Protocol and Listener** step, set the listener protocol to **HTTP** and specify a listening port.
6. In the **Advanced** section, click Modify. Turn on **Redirection** and select a listener to which you want to redirect requests.

   The listener can be an HTTPS listener with any port in the SLB instance.

   You can set the status code for redirection in the UK (London) region.
7. Click **Next**.
8. Check the configurations and click **Submit**.
9. Click **OK**.

   After the redirection feature is enabled, all HTTP requests are redirected to the selected HTTPS listener and distributed based on the configurations of the HTTPS listener.



← slb_[redacted]6

□	Frontend Protocol/Port	Backend Protocol/Port	Name	Status	Health Status	Monitoring	Forwarding Rule	Session Persistence	Peak Bandwidth	Server Group	Access Control List	Actions
□	HTTP:30	Redirect To HTTPS: 20	http_30	✓ Running	-		--	--	--	--	--	More

9.FAQ

9.1. FAQ about CLB

This topic provides answers to some frequently asked questions about Classic Load Balancer (CLB).

- Does CLB support port forwarding?
- Will my service be interrupted after I disable a public network interface controller (NIC)?
- Why do connections fail to reach the maximum bandwidth?
- How long is the connection timeout period for each listener?
- Why does a CLB connection time out?
- Why does session persistence fail?
- How do I view a cookie?
- How do I verify session persistence by using the Linux curl command?
- If a connection is closed on the client side before the client receives a response, does CLB close the connection on the server side?
- If a request already carries a TCP Option Address (TOA) header before the request is sent to CLB, can CLB preserve the client IP address in the header?

Does CLB support port forwarding?

Yes, CLB supports port forwarding.

For more information, see Redirect HTTP requests to HTTPS.

Will my service be interrupted after I disable a public network interface controller (NIC)?

If an Elastic Compute Service (ECS) instance is assigned a public IP address, your service will be interrupted after you disable the public NIC.

By default, if an ECS instance is assigned a public IP address, the ECS instance uses the public IP address to communicate with the Internet. In this case, network traffic is transferred through the public NIC of the ECS instance. If you disable the public NIC, the ECS instance cannot send responses to the Internet. We recommend that you keep the public NIC enabled. If you want to disable the public NIC, you must change the destination of the default route to a private IP address. Before you modify the default route, make sure that the ECS instance does not need to access resources such as ApsaraDB RDS over the Internet.

Why do connections fail to reach the maximum bandwidth?

CLB provides load balancing services by evenly distributing network traffic across groups of backend servers. Therefore, bandwidth is evenly allocated to each connection.

The maximum bandwidth of each connection is calculated based on the following formula: `Maximum bandwidth per connection = Maximum bandwidth of the CLB instance/(N - 1)` . N represents the number of server groups. In this example, N is 4. For example, if you set the maximum bandwidth of a CLB instance to 10 Mbit/s, the maximum bandwidth of each connection = `10/(4 - 1) = 3.33 Mbit/s` .

To prevent service interruptions, we recommend that you set the maximum bandwidth of each listener to a proper value.

How long is the connection timeout period for each listener?

- TCP listener: 900 seconds
- UDP listener: 90 seconds
- HTTP listener: 60 seconds
- HTTPS listener: 60 seconds

Why does a CLB connection time out?

Possible causes:

- The IP address of CLB is blocked for security reasons.

  The IP address of CLB may be blocked due to traffic blackholing and scrubbing or Web Application Firewall (WAF) protection. WAF offers protection by sending Reset (RST) packets to the client and the backend server after a connection is established.

- Client port exhaustion.

  Client port exhaustion can cause connection errors, especially in stress tests. By default, CLB removes the timestamps of TCP connections. As a result, the tw_reuse setting does not take effect and connections in the time_wait state cannot be reused. These accumulated connections exhaust all ports.

  Solution: Set clients to establish long-lived connections instead of short-lived connections. Set the so_linger socket option to close connections by sending RST packets instead of FIN packets.

- The accept queue of a backend server is full.

  If the accept queue on the backend server is full, the backend server can no longer return syn_ack packets. As a result, the client times out.

  Solution: The default value of the net.core.somaxconn parameter is 128. Run `sysctl -w net.core.somaxconn=1024` to change the value of this parameter to 1024 and then restart the application on the backend server.

- Layer 4 CLB is accessed from backend servers.

  If you access Layer 4 CLB from a backend server, the connection fails. For example, a request that reaches a backend server is redirected to the application on another backend server.

- RST packets are not correctly responded.

  If no data is transmitted within 900 seconds after a TCP connection is established on CLB, CLB sends RST packets to the client and the backend server to close the connection. If the application on the backend server does not correctly respond to the RST packet, it may send data packets after the connection is closed. As a result, a CLB connection timeout occurs.

Why does session persistence fail?

- Check whether session persistence is enabled for the listener.
- HTTP and HTTPS listeners cannot persist sessions by inserting cookies into responses that carry 4xx status codes.

Solution: Use TCP listeners instead of HTTP or HTTPS listeners. TCP listeners persist sessions based on client IP addresses. Backend servers can also insert or even validate cookies to ensure that sessions are persisted.

- HTTP 302 redirects change the SERVERID string for persisting a session.

  When CLB inserts a cookie into a response that carries the HTTP status code 302, the SERVERID string is changed. As a result, the session cannot be persisted.

  To verify the cause, check the requests and responses by using your browser or packet capture software. Then, check whether a 302 status code is included in the packets and whether the SERVERID string in the cookie is changed.

  Solution: Use TCP listeners instead of HTTP or HTTPS listeners. TCP listeners persist sessions based on client IP addresses. Backend servers can also insert or even validate cookies to ensure that sessions are persisted.

- The timeout period is set to a small value. You can set the timeout period to a greater value.

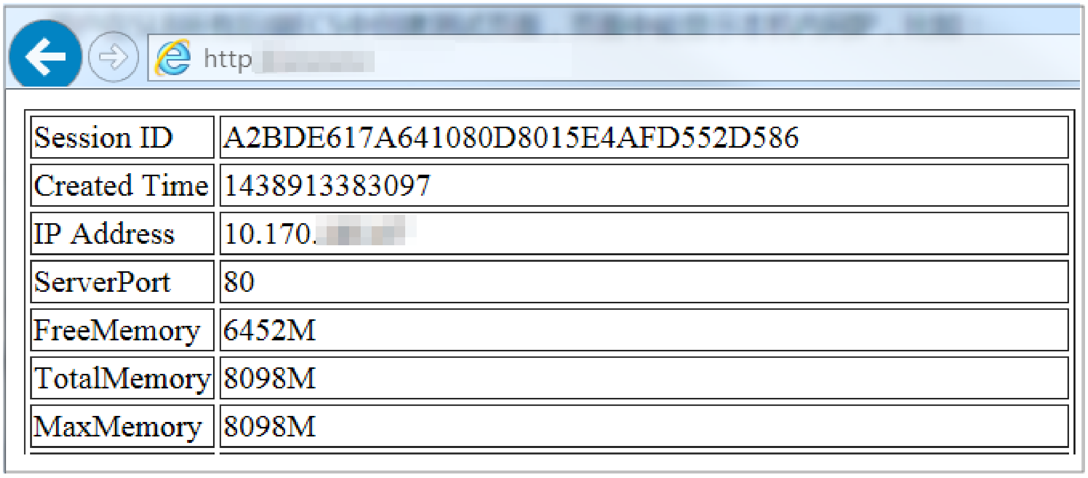
How do I view a cookie?

Open the browser and press F12 to check whether SERVERID or a user-defined cookie is inserted in the response. You can also run the `curl www.example.com -c /tmp/cookie123` command to save a cookie and then run the `curl www.example.com -b /tmp/cookie123` command to view the cookie.

How do I verify session persistence by using the Linux curl command?

1. Create a test page.

   Create a test page on each backend server. You can view the private IP address of the backend server on the test page. The following figure shows an example of a test page. The private IP address indicates the backend server to which requests are distributed. The private IP address is used to check whether CLB can persist sessions.

Session ID	A2BDE617A641080D8015E4AFD552D586
Created Time	1438913383097
IP Address	10.170.[redacted]
ServerPort	80
FreeMemory	6452M
TotalMemory	8098M
MaxMemory	8098M

2. Run the curl command in Linux.

   In this example, the IP address of the CLB instance that runs Linux is 10.170.XX.XX and the URL of the created page is `http://10.170.XX.XX/check.jsp`.

   i. Log on to a server that runs Linux.

ii. Run the following command to query the cookie inserted by the backend server:

```
curl -c test.cookie http://10.170.XX.XX/check.jsp
```

> ⓘ **Note** By default, CLB persists sessions by inserting cookies. However, curl does not send or save cookies. Therefore, you must save a cookie before you perform the test. Otherwise, the curl test result may show that session persistence is invalid.

iii. After you save the cookie, run the following command:

```
for ((a=1;a<=30;a++));
    do curl  -b test.cookie http://10.170.XX.XX/check.jsp  | grep '10.170.XX.XX';
    sleep 1;
done
```

> ⓘ **Note** a<=30 indicates the number of tests to be performed. You can set this value based on your business requirements. Set the IP address in `grep '10.170.XX.XX'` to the private IP address of your ECS instance.

iv. Check the IP addresses returned in the preceding tests. If the same IP address is returned, it indicates that CLB can persist sessions.

If a connection is closed on the client side before the client receives a response, does CLB close the connection on the server side?

No, CLB does not close the connection on the server side in this case.

If a request already carries a TCP Option Address (TOA) header before the request is sent to CLB, can CLB preserve the client IP address in the header?

No, CLB cannot preserve the client IP address in this case. By default, if the TOA module is installed, CLB automatically adds TOA headers to preserve client IP addresses. When CLB receives a request that carries a TOA header, CLB cannot preserve the client IP address in the header.

You can use the following methods to preserve the client IP address:

- Enable Proxy Protocol for a Layer 4 listener to retrieve client IP addresses
- Preserve client IP addresses when Layer 7 listeners are used

9.2. FAQ about Layer 7 listeners that use HTTPS or HTTP

The following questions about Layer 7 listeners that use HTTPS or HTTP are frequently asked:

- Why are some response header parameters deleted after requests are forwarded by Layer 7 listeners?
- Why is an additional header, Transfer-Encoding: chunked, added to an HTTP request?
- Why do the style sheets fail to load when I open a website over an HTTPS listener?
- Which port number do HTTPS listeners use?
- What types of certificates does SLB support?

- Does SLB support keytool-created certificates?
- Can I use certificates in the PKCS#12 (PFX) format?
- Why does a KeyEncryption error occur when I upload certificates?
- What SSL protocol versions are supported by the HTTPS Server Load Balancer service?
- What is the lifetime of an HTTPS session ticket?
- Can I upload a certificate that contains DH PARAMETERS?
- Do HTTPS listeners support SNI?
- Which version of HTTP is used by HTTP and HTTPS listeners to access the backend servers?
- Can backend servers obtain the protocol version used by the client to access the HTTP or HTTPS listener?
- What are the timeout values specified for HTTP and HTTPS listeners?

Why are some response header parameters deleted after requests are forwarded by Layer 7 listeners?

Symptom: SLB modifies the values of the Date, Server, X-Pad, and X-Accel-Redirect parameters in the response headers to implement session persistence.

Solution:

- Add a prefix to the custom header, such as xl-server or xl-date.
- Change Layer 7 HTTP listeners to Layer 4 TCP listeners.

Why is an additional header, Transfer-Encoding: chunked, added to an HTTP request?

Symptom: After a domain name is resolved to the service address of Layer 7 SLB instance, a Transfer-Encoding: chunked field is added in the HTTP request header when you access the domain name from a local host. However, this field is not found in the request when you access backend servers directly from the local host.

Cause: Layer 7 SLB is based on the Tengine reverse proxy. The Transfer-Encoding field indicates how the web server encodes the response message body. For example, Transfer-Encoding: chunked indicates that chunked transfer encoding is used.

> **Note** This header is not added in the requests forwarded by Layer 4 listeners, because Layer 4 listeners only distribute traffic.

Why do the style sheets fail to load when I open a website over an HTTPS listener?

Symptom:

An HTTP listener and an HTTPS listener are created, and they use the same backend servers. When you access the website over the HTTP listener with the specified port number, the website is displayed normally. However, the website layout is messy when you access the website over the HTTPS listener.

Cause:

By default, SLB does not block loading and transferring of JavaScript files. This problem may be caused by the following reasons:

- The certificate is not compatible with the security level of the web browser.
- The certificate is an unqualified third-party certificate. In this case, contact the certificate issuer to check the certificate.

Solution:

1. When you open the website, load scripts as prompted by the browser.
2. Add the required certificate to the browser.

Which port number do HTTPS listeners use?

HTTPS listeners have no special requirements on ports. However, we recommend that you use 443 as the port number for HTTPS listeners.

What types of certificates does SLB support?

SLB supports server certificates and CA certificates in the PEM format.

For the server certificates, you must upload both the certificate content and the private key. For the CA certificates, you need to upload only the certificate content.

Does SLB support keytool-created certificates?

Yes.

You must convert the certificate format to PEM before you upload the certificates to SLB. For more information, see Convert the certificate format.

Can I use certificates in the PKCS#12 (PFX) format?

Yes.

You must convert the certificate format to PEM before you upload the certificates to SLB. For more information, see Convert the certificate format.

Why does a KeyEncryption error occur when I upload certificates?

The private key contains incorrect contents. For more information about the private key format, see Certificate requirements.

What SSL protocol versions are supported by the HTTPS Server Load Balancer service?

TLSv1, TLSv1.1, and TLSv1.2.

What is the lifetime of an HTTPS session ticket?

The lifetime of an HTTPS session ticket is set to 300 seconds.

Can I upload a certificate that contains DH PARAMETERS?

The ECDHE cipher suites used by HTTPS listeners support forward secrecy but do not support the security enhancement parameters required by DHE cipher suites. As a result, strings that contain the BEGIN DH PARAMETERS field in a PEM certificate file cannot be uploaded.

Do HTTPS listeners support SNI?

Yes. Server Name Indication (SNI) is an extension to the SSL or TLS protocol so that a server can use multiple domain names and certificates. HTTPS listeners support the SNI feature. For more information, see Add an additional certificate.

Which version of HTTP is used by HTTP and HTTPS listeners to access the backend servers?

- When the protocol used by client requests is HTTP 1.1 or HTTP 2.0, Layer 7 listeners use HTTP 1.1 to access backend servers.
- When the protocol used by client requests is not HTTP 1.1 or HTTP 2.0, Layer 7 listeners use HTTP 1.0 to access backend servers.

Can backend servers obtain the protocol version used by the client to access the HTTP or HTTPS listener?

Yes, backend servers can obtain the protocol version used by the client to access the HTTP or HTTPS listener.

What are the timeout values specified for HTTP and HTTPS listeners?

- A maximum of 100 requests can be sent continuously in an HTTP persistent connection. The connection is closed when the limit is reached.
- The timeout period between two HTTP or HTTPS requests in an HTTP persistent connection can be set to a value ranging from 1 to 60 seconds. The TCP connection is closed when the timeout period exceeds the specified value. If you want to use the HTTP persistent connection, try to send heartbeat requests within 13 seconds.
- The timeout period for the TCP three-way handshake between SLB and a backend ECS instance is 5 seconds. After the handshake times out, SLB selects the next ECS instance. You can find the timeout record by checking the upstream response time in the access logs.
- The time that SLB waits for the response from an ECS instance can be set to a value ranging from 1 to 180 seconds. If the wait time exceeds the specified timeout period, a 504 or 408 status code is sent to the client. You can find the timeout record by checking the upstream response time in the access logs.
- After 300 seconds, the HTTPS session reuse times out. Then, the client must perform the complete SSL handshake process again.

9.3. WebSocket and WebSocket Secure support FAQ

WebSocket is a new HTML5 protocol for full-duplex communication between clients and servers. It supports real-time communication and reduces the consumption of server resources and bandwidth. WebSocket Secure is the encrypted version of WebSocket.

Introduction to WebSocket and WebSocket Secure

WebSocket is a new HTML5 protocol for full-duplex communication between clients and servers. It supports real-time communication and reduces the consumption of server resources and bandwidth. Similar to HTTP, WebSocket uses an established TCP connection to transmit data. However, it is different from HTTP.

One major difference between WebSocket and HTTP is that WebSocket is a two-way communication protocol. After a connection is established, a WebSocket server and client can send or receive data between each other similar to a socket. The WebSocket server and client must complete a handshake to establish a WebSocket connection.

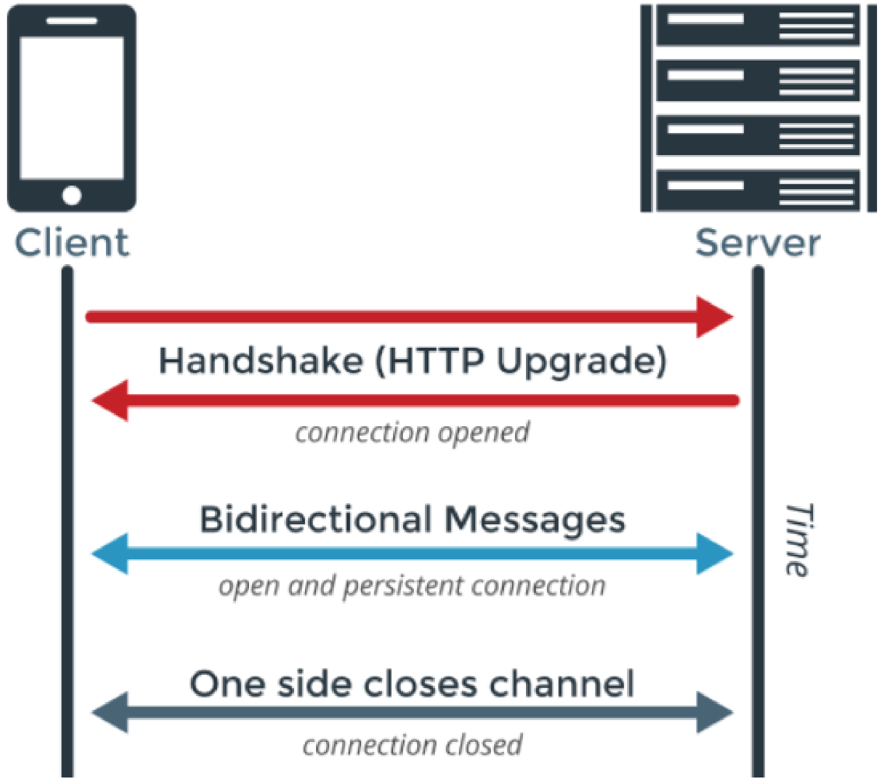
WebSocket Secure is the encrypted version of WebSocket.

Background information about WebSocket and WebSocket Secure

New web applications emerge as the Internet develops. These applications, such as live video streaming and online chat rooms, require that servers have real-time pushing capabilities. To implement pushing, a large number of websites used the polling technique. Based on the polling technique, the browser sends HTTP requests to the server at specific intervals, such as every second. Then, the server returns the most recent data to the browser of the client. One disadvantage of this technique is that bandwidth resources are wasted. The browser must constantly send requests to the server, and the HTTP request header is long and does not contain a large amount of valid data.

To solve this problem, HTML5 defines WebSocket, which helps save server and bandwidth resources and facilitates real-time communication. WebSocket supports full-duplex communication between clients and servers and allows a server to send requests to a client.

The following figure shows how a client interacts with a server by using WebSocket.



How to enable WebSocket and WebSocket Secure for an SLB instance

No configuration is required. HTTP listeners support WebSocket and HTTPS listeners support WebSocket Secure by default.

Note You must upgrade the SLB instance to a guaranteed-performance instance. For more information, see FAQ about high-performance CLB instances.

Supported regions

WebSocket and WebSocket Secure are available in all regions.

Limits

The following section describes the limits for WebSocket and WebSocket Secure:

- SLB is connected to backend ECS instances by using HTTP/1.1. We recommend that you use web servers that support HTTP/1.1 for backend ECS instances.
- If no message interactions exist between an SLB instance and backend ECS instances within 60 seconds, the connection is terminated. If you need to maintain the connection, enable Keepalive to ensure a message interaction at the frequency of once every 60 seconds.

Billing

WebSocket and WebSocket Secure are free of charge.