

Cloud Load Balancer

Operation Guide

Product Documentation



Copyright Notice

©2013-2025 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Operation Guide

CLB Instance

Directions for Upgrading to Domain Name-Based CLB

Creating CLB Instances

Creating an IPv6 CLB Instance

Creating IPv6 NAT64 CLB Instances

Configure CLB Forwarding Domain Name

Configuring CLB Security Group

Binding Private Network CLB to EIP

Starting or Stopping a CLB Instance

Cloning CLB Instances

Exporting CLB Instances

Upgrade to a LCU-supported instances

Adjusting Specification of LCU-supported CLBs

Deleting CLB Instances

Releasing Idle CLB Instances

Configuring Deletion Protection

Adjusting Instance Public Network Configurations

CLB Listener

CLB Listener Overview

Configuring TCP Listener

Configuring a UDP Listener

Configuring TCP SSL Listener

Configuring a QUIC Listener

Configuring an HTTP Listener

Configuring HTTPS Listener

Load Balancing Methods

Session Persistence

Layer-7 Redirection Configuration

Layer-7 Custom Configuration

Layer-7 Domain Name Forwarding and URL Rules

Using QUIC Protocol on CLB

SNI Support for Binding Multiple Certificates to a CLB Instance

Configuring gRPC Support for Layer-7 Protocols

Real Server

Real Server Overview

Managing Real Servers

Binding an ENI

Binding with SCF

Cross-Region Binding 2.0 (New)

Hybrid Cloud Deployment

Configuring CVM Security Groups

Health Check

Health Check Overview

Configuring Health Check

Setting 100.64.0.0/10 IP Range as the Health Check IP

Verifying Health Check Source IPs

Certificate Management

Managing Certificates

Certificate Requirements and Certificate Format Conversion

SSL One-way Authentication and Mutual Authentication

Log Management

Access Log Overview

Viewing Operation Logs

Configuring Access Logs

Sampling Logs

Configuring Health Check Logs

Accessing Log Dashboard

Monitoring and Alarm

Obtaining Monitoring Data

Monitoring Metrics

Configuring Alarm Policy

Alarming Metric Descriptions

Cloud Access Management

Overview

Authorization Definition

Policy Examples

Classic CLB

Classic CLB Overview

Configuring Classic CLB

Managing Real Servers of Classic CLB Instances

Operation Guide

CLB Instance

Directions for Upgrading to Domain Name-Based CLB

Last updated : 2024-10-17 18:05:02

You can upgrade your existing public network Cloud Load Balancer (CLB) instances to domain name-based CLB instances. After the upgrade, the CLB service will be delivered through domain names, and VIPs may change dynamically with business requests and will no longer be displayed in the console.

CLB Service Comparison Before and After the Upgrade

Item	After Upgrade	Before Upgrade
SLA	99.99%	99.95%
Domain names supported	Yes	No
Automatic VIP scaling supported	Yes	No
VIP changes	VIPs may change dynamically with business requests and will no longer be displayed in the console.	VIPs are fixed.
Health check source IP	100.64.0.0/10 IP range by default, helping prevent IP conflicts	CLB instance VIP by default, which can be switched to the 100.64.0.0/10 IP range

Restrictions

Classic networks do not support the upgrade of CLB instances. If your CLB instance is in a classic network, migrate the instance first. For more information, see [Ending Support for Classic Network](#). Classic CLB does not support upgrade. Upgrade classic CLB instances to CLB instances as instructed in [Classic CLB Upgrade Prompt](#).

CLB instances created by containers currently do not support direct upgrade through the console. If upgrade is needed, please [submit a ticket](#) for assistance.

Anti-DDoS Pro currently does not support protection for domain name-based CLB. After upgrading to domain name-based CLB, Anti-DDoS Pro will fail and seriously affect business security. For users with public network CLB instances already bound to Anti-DDoS Pro, or users with Anti-DDoS needs for public network CLB instances, upgrading to domain name-based CLB instances is not recommended. For other issues, please [submit a ticket](#) for assistance.

Prerequisites

1. Your business service is accessible through CNAME resolution.
2. The health check source IP has been changed to the 100.64.0.0/10 IP range. For more information, see [Changing Health Check Source IP](#).







Directions

Method 1: Upgrading a specific CLB instance

1. Log in to the [CLB Console](#).
2. Select a region in the top-left corner of the **Instance management** page. In the instance list, locate the target instance, click **More** in the **Operation** column of the instance, and select **Upgrade to domain name-based instance**.
3. In the **Upgrade to domain name-based instance** pop-up window, click **OK**.

Upgrade to domain name-based instance

Instances to upgrade: 1

ID/Name	Network type	Assign domain name ⓘ	Current VIP	VIP
lb-  	Public Network	lb-1   .tencentclb.com 	1  .47	Dynamic IP

Benefits

Domain name-based instances provides service via a domain name with dynamic VIPs. The SLA increases from 99.95% to 99.99%.

Preparation

The health check source IP is changed to 100.64 IP range. You need to allow this IP range in other security policies (such as iptables) of the backend server. For details, see [The health check source IP supports 100.64.0.0/10 IP range.](#)

How to upgrade

1. Use load balancing service via a CNAME. For details, see the [usage guide](#).
2. to upgrade.

Impact

- The upgrade **does not affect the CLB forwarding service and pricing.**
- After the upgrade, the **VIP information is not visible** in the console. They are **changed dynamically.**
- The upgrade **cannot be undone.**

For any other questions, please [submit a ticket](#).

OK

Cancel

Method 2: Upgrading instances in batches

1. Log in to the [CLB Console](#).
2. Select a region in the top-left corner of the **Instance management** page and select the instances to upgrade in the instance list.
3. Click **More** above the instance list, and select **Upgrade to domain name-based instance**.

4. In the **Upgrade to domain name-based instance** pop-up window, click **OK**.

Page 8 of 262

Creating CLB Instances

Last updated : 2025-02-13 18:38:24

Tencent Cloud allows you to purchase a CLB instance on the official purchase page or via API. The two methods are detailed below:

Purchasing a CLB Instance on the Official Purchase Page

You can purchase a CLB instance on the [Tencent Cloud official website](#). There are two types of Tencent Cloud accounts, namely the bill-by-IP accounts and bill-by-CVM accounts. The accounts created after 00:00:00 on June 17, 2020 (Beijing time) are of the bill-by-IP type. If you have created your account before that time, you can check your account type in the console as instructed in [Checking Account Type](#).

1. Log in to the Tencent Cloud console and go to the [CLB purchase page](#).
2. Select the following CLB configuration items as needed:

Bill-by-IP account

Parameter	Description
Billing mode	Supports pay-as-you-go billing.
Region	Select a region. For more information on the regions supported by CLB, see Region List .
Instance type	Supports the CLB instance type only. Starting from October 20, 2021, classic CLB instances can no longer be purchased. For more information, see Classic CLB End-of-Sale Notice .
Network type	Supports two network types: public network and private network. For more information, see Network Types . Public network: CLB is used to distribute requests from a public network. Private network: CLB is used to distribute requests from the Tencent Cloud private network. A private network instance does not support the following configuration items, and therefore they are not displayed by default: EIP, IP version, ISP, instance specification, network billing mode, and bandwidth cap. The supported network types vary by billing mode: In pay-as-you-go billing mode, both the public and private network types are supported.
EIP	If EIP is not selected, Tencent Cloud will assign you a public network CLB instance whose public IP address cannot be changed. If EIP is selected, Tencent Cloud will assign you an EIP and a private network CLB instance, which has the similar features of public network CLB. (Only pay-as-you-go public network CLB instances allow you to select an EIP.)

	This feature is currently in beta. To use it, submit a ticket . For the use limits, see Use Limits .
IP version	Supports the following CLB IP versions: IPv4, IPv6, and IPv6 NAT64. Only pay-as-you-go instances support the IPv6 version. For more information about other restrictions, see IP Versions . IPv6 CLB is currently in beta. To use it, submit a ticket .
Network	<p>CLB supports classic network and VPC.</p> <p>The classic network is a public network resource pool for all Tencent Cloud users. The private IPs of all CVMs are assigned by Tencent Cloud. You cannot customize IP ranges or IP addresses.</p> <p>A VPC is a logically isolated network space in Tencent Cloud. In a VPC, you can customize IP ranges, IP addresses, and routing policies.</p> <p>A VPC is more suitable for use cases requiring custom configurations. Besides, the overall classic network products were officially discontinued on December 31, 2022. For details, see Ending Support for Classic Network. We recommend you choose a VPC.</p>
ISP	<p>Supports the following ISP types: BGP (multi-line), China Mobile, China Telecom, and China Unicom.</p> <p>In pay-as-you-go billing mode, all of the above four options are supported. Currently, the static single-line IP is supported only in Guangzhou, Shanghai, Nanjing, Jinan, Hangzhou, Fuzhou, Beijing, Shijiazhuang, Wuhan, Changsha, Chengdu, and Chongqing. For the support information in other regions, see the console. If you want to try it out, contact the sales rep for application. Once your application is approved, you can select an ISP (China Mobile, China Unicom, or China Telecom) on the purchase page.</p>
Primary/Secondary availability zone	The primary availability zone (AZ) is an AZ that currently sustains the traffic. The secondary AZ does not sustain traffic by default and will be used only when the primary AZ is unavailable.
Instance specification	<p>Supports shared and LCU-supported instances.</p> <p>Shared instances guarantee performance according to their specifications. A single instance can sustain up to 50,000 concurrent connections, 5,000 new connections per second, and 5,000 queries per second.</p> <p>An LCU-supported instances provide guaranteed performance according to specifications. A single instance can sustain up to Ten million concurrent connections, 1 million new connections per second, and 300,000 queries per second.</p>
Network billing mode	<p>Supports the following network billing modes: bill-by-bandwidth (monthly subscription and hourly bandwidth), bill-by-traffic, and bandwidth package.</p> <p>A pay-as-you-go instance supports three network billing modes: bill-by-bandwidth (hourly bandwidth), bill-by-traffic, and bandwidth package.</p>
Bandwidth cap	<p>The public network bandwidth cap of a shared CLB is 2 Gbps. For the private network bandwidth cap, it's suggested to set a value within 5 Gbps.</p> <p>The bandwidth cap of an LCU-supported CLB instance depends on the selected specification. For details, see Instance Specifications Comparison.</p>

Project	Select a project.
Tag	Select a tag key and value. You can also create a tag as instructed in Creating Tags and Binding Resources .
Instance name	The name can contain up to 60 characters, including letters, numbers, hyphens, underscores, and dots. If it is not specified, a name will be automatically generated by default.

Bill-by-CVM account

Parameter	Description
Billing mode	Supports pay-as-you-go billing only.
Region	Select a region. For more information on the regions supported by CLB, see Region List .
Instance type	Supports the CLB instance type only. Starting from October 20, 2021, classic CLB instances can no longer be purchased. For more information, see Classic CLB End-of-Sale Notice .
Network type	Supports two network types: public network and private network. For more information, see Network Types . Public network: CLB is used to distribute requests from a public network. Private network: CLB is used to distribute requests from the Tencent Cloud private network. A private network instance does not support the following configuration items, and therefore they are not displayed by default: IP version, ISP, and instance specification.
IP version	Supports the following CLB IP versions: IPv4, IPv6, and IPv6 NAT64. For more information about use limits, see IP Versions . IPv6 CLB is currently in beta. To use it, submit a ticket .
Network	CLB supports classic network and VPC. The classic network is a public network resource pool for all Tencent Cloud users. The private IPs of all CVMs are assigned by Tencent Cloud. You cannot customize IP ranges or IP addresses. A VPC is a logically isolated network space in Tencent Cloud. In a VPC, you can customize IP ranges, IP addresses, and routing policies. A VPC is more suitable for use cases requiring custom configurations. Besides, the overall classic network products were officially discontinued on December 31, 2022. For details, see Ending Support for Classic Network . We recommend you choose a VPC.
ISP	Supports the following ISP types: BGP (multi-line), China Mobile, China Telecom, and China Unicom. Currently, the static single-line IP is supported only in Guangzhou, Shanghai, Nanjing, Jinan, Hangzhou, Fuzhou, Beijing, Shijiazhuang, Wuhan, Changsha, Chengdu, and Chongqing. This feature is in beta. To use it, submit a ticket . For the support information in other regions, see the console. If you want to try this feature, contact the sales rep for application. Once your

	application is approved, you can select an ISP (China Mobile, China Unicom, or China Telecom) on the purchase page.
Instance specification	Supports shared and LCU-supported instances. Shared instances guarantee performance according to their specifications. A single instance can sustain up to 50,000 concurrent connections, 5,000 new connections per second, and 5,000 queries per second. An LCU-supported instances provide guaranteed performance according to specifications. A single instance can sustain up to Ten million concurrent connections, 1 million new connections per second, and 300,000 queries per second.
Project	Select a project.
Tag	Select a tag key and value. You can also create a tag as instructed in Creating Tags and Binding Resources .
Instance name	The name can contain up to 60 characters, including letters, numbers, hyphens, underscores, and dots. If it is not specified, a name will be automatically generated by default.

3. After completing the above configuration, confirm the quantity and fees and click **Buy now**.

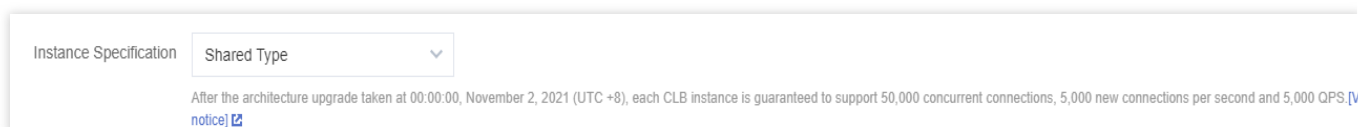
Pay-as-you-go billing mode: In the **Confirm** pop-up window, click **OK**.

4. After successful purchase, CLB will be activated and you can configure and use the CLB instance.

Purchasing a shared instance

1. Log in to the Tencent Cloud console and go to the [CLB purchase page](#).

2. Set the shared instance configuration items by referring to the steps in [Purchasing a CLB Instance on the Official Purchase Page](#) and select **Shared** for **Instance specification**.



3. Complete the subsequent operations by referring to the steps in [Purchasing a CLB Instance on the Official Purchase Page](#).

Purchasing an LCU-supported instance

1. Log in to the Tencent Cloud console and go to the [CLB purchase page](#).

2. Set the LCU-supported instance configuration items by referring to the steps in [Purchasing a CLB Instance on the Official Purchase Page](#) and select **LCU-supported** for **Instance specification**.

Instance specification

LCU-supported

Provides the guaranteed forwarding performance for each instance. Each instance supports up to 10,000,000 concurrent connections, 1,000,000 new connections, and 300,000 QPS.[Billing description](#)

Model	Max concurrent con...	New connections pe...	Queries per second	Bandwidth cap (Mbps)
<input checked="" type="radio"/> Standard(c1b.c2.medium)	100,000	10,000	10,000	2,048
<input type="radio"/> Higher I(c1b.c3.small)	200,000	20,000	20,000	4,096
<input type="radio"/> Higher II(c1b.c3.medium)	500,000	50,000	30,000	6,144
<input type="radio"/> Super I(c1b.c4.small)	1,000,000	100,000	50,000	10,240
<input type="radio"/> Super II(c1b.c4.medium)	2,000,000	200,000	100,000	20,480
<input type="radio"/> Super III(c1b.c4.large)	5,000,000	500,000	200,000	40,960
<input type="radio"/> Super IV(c1b.c4.xlarge)	10,000,000	1,000,000	300,000	61,440

For the pay-as-you-go mode, the selected specification refers to the upper limit. The LCU price is the same. For details, see [Billing description](#)

3. Complete the subsequent operations by referring to the steps in [Purchasing a CLB Instance on the Official Purchase Page](#).

Purchasing a CLB Instance via an API

To purchase a CLB instance via an API, see [CreateLoadBalancer](#).

Related Operations

You can create a listener for a CLB instance as instructed in [CLB Listener Overview](#).

You can bind a CLB listener to a real server as instructed in [Real Server Overview](#).

References

[Product Attribution Selection](#)

Creating an IPv6 CLB Instance

Last updated : 2024-10-09 16:16:34

Note:

The IPv6 CLB is in beta test. To try it out, [submit a ticket](#).

Currently, IPv6 CLBs are available in the following regions: Guangzhou, Shenzhen Finance, Shanghai, Shanghai Finance, Nanjing, Beijing, Beijing Finance, Chengdu, Chongqing, Hong Kong (China), Singapore, Virginia, and São Paulo. Due to compliance requirements, you need to [submit a ticket](#) to open Shenzhen Finance and Shanghai Finance regions. IPv6 CLB does not support classic CLB.

IPv6 load balancing does not support conventional load balancing.

IPv6 load balancing facilitates the acquisition of the client's IPv6 source address. Both four-layer and seven-layer protocols inherently transmit the client's source address from the IPv6 load balancer. Additionally, the seven-layer IPv6 load balancer supports the retrieval of the client's IPv6 source address via the HTTP's `X-Forwarded-For` header field.

Currently, IPv6 CLB balances the load completely over a public network. Clients in the same VPC cannot access IPv6 CLB over a private network.

IPv6 implementations are still at the preliminary stage across the internet. In case of access failure, [submit a ticket](#). SLA is not guaranteed during the beta test period.

Overview

IPv6 load balancing is implemented based on the IPv6 single stack technology. It can collaborate with IPv4 CLB to implement IPv6/IPv4 dual-stack communication. An IPv6 CLB instance is bound to an IPv6 address of a CVM instance and provides an IPv6 VIP address.

IPv6 CLB Advantages

Tencent Cloud IPv6 CLB has the following advantages when helping your business quickly connect to IPv6:

Quick connection: CLB enables connection to IPv6 within seconds and is available upon purchase.

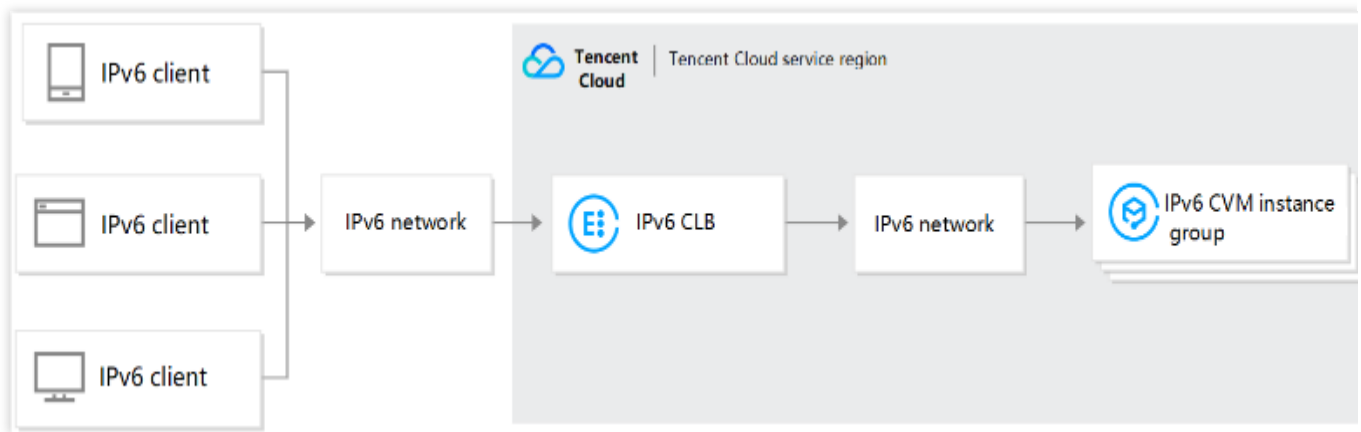
Ease of use: IPv6 CLB is compatible with IPv4 CLB flowchart and easy to use with no additional learning costs incurred.

End-to-end IPv6 communication: IPv6 CLB instances communicate with CVM instances over IPv6, which helps applications deployed on the CVM instances quickly upgrade to IPv6 and implement end-to-end IPv6 communication.

IPv6 CLB Architecture

CLB supports creating IPv6 CLB instances. Tencent Cloud will assign an IPv6 public IP address, i.e., VIP of the IPv6 edition, to an IPv6 CLB instance, and the VIP will forward requests from IPv6 clients to the real IPv6 CVM instance.

An IPv6 CLB instance can support quick access of users from IPv6 public network and communicate with real servers over IPv6, which helps in-cloud applications quickly upgrade to IPv6 and implement end-to-end IPv6 communication. The IPv6 CLB architecture is as shown below:



Step 1. Create an IPv6 CLB instance

1. Log in to the Tencent Cloud console and go to the [CLB purchase page](#).
2. Select the following CLB configuration items as needed:

Bill-by-IP account

Parameter	Description
Billing mode	Supports pay-as-you-go billing. IPv6 CLB is supported only in pay-as-you-go mode. For other restrictions, see IP Versions .
Region	Select a region. For more information on the regions supported by CLB, see Region List .
Instance type	Supports the CLB instance type only. Starting from October 20, 2021, classic CLB instances can no longer be purchased. For more information, see Classic CLB End-of-Sale Notice .
Network type	Supports two network types: public network and private network. For more information, see Network Types . Select the public network type for IPv6 CLB.
EIP	Don't select an EIP.
IP version	Select the IPv6 version.
Network	Select an existing VPC or subnet. If the existing networks are inapplicable, you can create a VPC or create a subnet as required.
ISP	Select Multi-line BGP .

Instance specification	<p>Supports shared and LCU-supported instances.</p> <p>Shared instances guarantee performance according to their specifications. A single instance can sustain up to 50,000 concurrent connections, 5,000 new connections per second, and 5,000 queries per second.</p> <p>An LCU-supported instances provide guaranteed performance according to specifications. A single instance can sustain up to Ten million concurrent connections, 1 million new connections per second, and 300,000 queries per second.</p>
Dual-stack binding	After this feature is enabled, the layer-7 listener can be bound with both IPv4 and IPv6 backend servers. But layer-4 listeners only support binding of IPv6 backend server.
Network billing mode	Supports bill by traffic and bill by bandwidth package.
Bandwidth cap	<p>The public network bandwidth cap of a shared CLB is 2 Gbps. For the private network bandwidth cap, it's suggested to set a value within 5 Gbps.</p> <p>The bandwidth cap of an LCU-supported CLB instance depends on the selected specification. For details, see Instance Specifications Comparison.</p>
Project	Select a project.
Tag	Select a tag key and value. You can also create a tag as instructed in Creating Tags and Binding Resources .
Instance name	The name can contain up to 60 characters, including letters, numbers, hyphens, underscores, and dots. If it is not specified, a name will be automatically generated by default.
Service agreement	I've read and agreed to Tencent Cloud Terms of Service and CLB Service Level Agreement .

Bill-by-CVM account

Parameter	Description
Billing mode	Supports pay-as-you-go billing only.
Region	Select a region. For more information on the regions supported by CLB, see Region List .
Instance type	Supports the CLB instance type only. Starting from October 20, 2021, classic CLB instances can no longer be purchased. For more information, see Classic CLB End-of-Sale Notice .
Network type	<p>Supports two network types: public network and private network. For more information, see Network Types.</p> <p>Public network: CLB is used to distribute requests from a public network.</p> <p>Private network: CLB is used to distribute requests from the Tencent Cloud private network. A private network instance does not support the following configuration items, and therefore they are not displayed by default: IP version, ISP, and instance specification.</p>

IP version	Select IPv6. For more information on the use limits, see IP Versions .
Network	<p>CLB supports classic network and VPC.</p> <p>The classic network is a public network resource pool for all Tencent Cloud users. The private IPs of all CVMs are assigned by Tencent Cloud. You cannot customize IP ranges or IP addresses.</p> <p>A VPC is a logically isolated network space in Tencent Cloud. In a VPC, you can customize IP ranges, IP addresses, and routing policies.</p> <p>A VPC is more suitable for use cases requiring custom configurations. Besides, the overall classic network products were officially discontinued on December 31, 2022. For details, see Ending Support for Classic Network. We recommend you choose a VPC.</p>
ISP	Select Multi-line BGP .
Instance specification	<p>Supports shared and LCU-supported instances.</p> <p>Shared instances guarantee performance according to their specifications. A single instance can sustain up to 50,000 concurrent connections, 5,000 new connections per second, and 5,000 queries per second.</p> <p>An LCU-supported instances provide guaranteed performance according to specifications. A single instance can sustain up to Ten million concurrent connections, 1 million new connections per second, and 300,000 queries per second.</p>
Network billing mode	Select Bandwidth package .
Bandwidth cap	Value range: 1-1024 Mbps.
Project	Select a project.
Tag	Select a tag key and value. You can also create a tag as instructed in Creating Tags and Binding Resources .
Instance name	The name can contain up to 60 characters, including letters, numbers, hyphens, underscores, and dots. If it is not specified, a name will be automatically generated by default.
Service agreement	I've read and agreed to Tencent Cloud Terms of Service and CLB Service Level Agreement .

3. After configuring the above items, click **Buy now**. In the "CLB order confirmation" pop-up window, click **Confirm order**. Then, return to the [Instance Management](#) page where you can view the IPv6 CLB instance you just purchased.

Step 2. Create an IPv6 CLB listener

1. Log in to the [CLB console](#) and click the IPv6 CLB instance ID to go to the details page.
2. Select the **Listener management** tab and click **Create**. For example, create a TCP listener.

Note:

CLB supports creating layer-4 (TCP/UDP/TCP SSL) and layer-7 (HTTP/HTTPS) IPv6 CLB listeners. For more information, see [CLB Listener Overview](#).

3. In **Basic configuration**, configure the name, listening protocol and port, and balancing method, and click **Next**.

CreateListener

1 Basic Configuration

2 Health Check

3 Session Persistence

Name

ipv6-ssh

Listen Protocol Ports

TCP

:

22

Balance Method

Weighted Round Robin

If you set a same weighted value for all CVMs, requests will be distributed by a simple pooling policy.

Close

Next

4. Configure health check and click **Next**.

CreateListener

1 Basic Configuration

2 **Health Check**

3 Session Persistence

Health Check ⓘ ☒

Show advanced options ▼

BackNext

5. Configure session persistence and click **Submit**.

CreateListener

1 Basic Configuration

2 Health Check

3 **Session Persistence**

Session Persistence ⓘ ☒

Hold Time ⓘ

III

30 Seconds3600 Seconds

-54+ Second

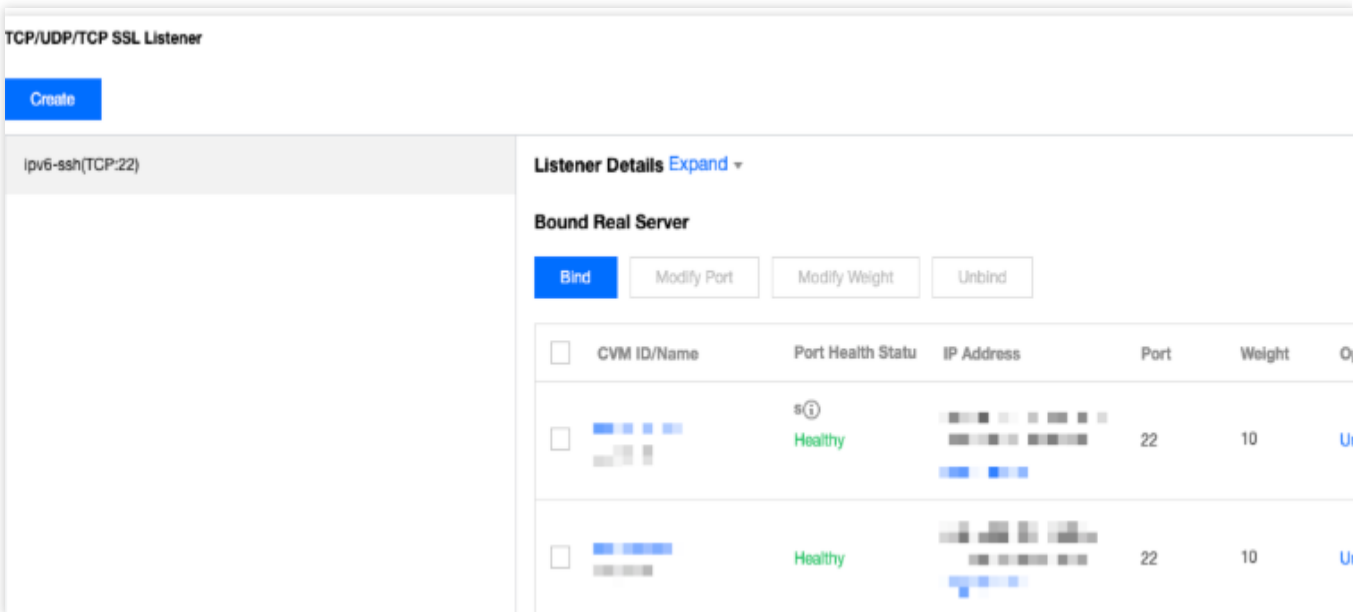
Session persistence based on the source IP

BackSubmit

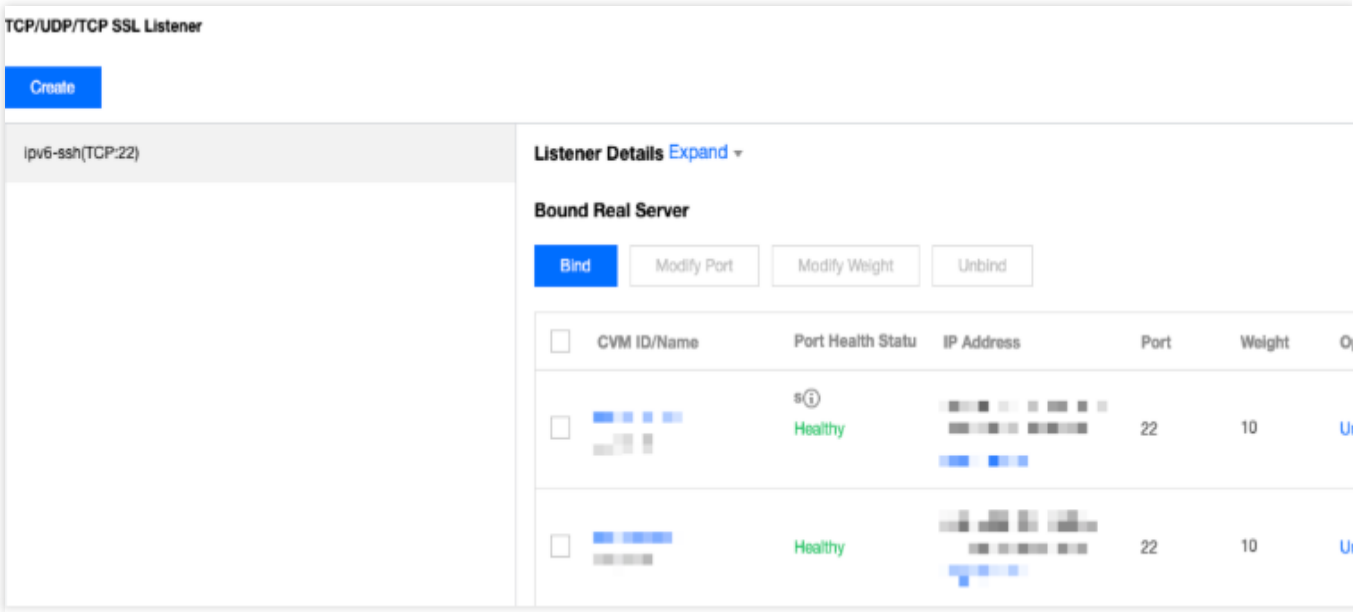
6. After the listener is created, select it and click **Bind** on the right.

Note:

Before binding the listener to a CVM instance, make sure that the CVM instance has obtained an IPv6 address.



7. In the pop-up window, select the target IPv6 CVM instance, configure the server ports and their weights, and click OK.



More Operations

Binding IPv6 CLB with both IPv6 and IPv4 real servers

After enabling the dual-stack binding, the IPv6 CLB layer-7 listener can be bound with both IPv4 and IPv6 backend servers, and can obtain the source IP via XFF. But layer-4 listeners only support binding of IPv6 backend server.

1. Enable dual-stack binding.

Enable dual-stack binding when purchasing IPv6 CLB on the purchase page.

Dual-stack binding ☒ Enable Dual-stack Binding

If it is enabled, the layer-7 listener of the CLB instance can be bound with both IPv4 and IPv6 real servers. But its layer-4 listener can only be bound with IPv6 real server.

Enable dual-stack binding on the IPv6 CLB instance details page.

← lb- (lb-)

Basic information

Listener management

Redirection configurations

Monitoring

Security group

Basic information

Name

lb- ()

ID

lb- ()

Status

Normal

Domain

()

VIP

(IPv6)

Dual-stack binding

Enabled ()

Instance type

Public network

Region

Guangzhou

Availability zone

Guangzhou Zone 3

ISP

Multi-line BGP

Network

Default-VPC(172.16.0.0/16 |)

Subnet

Default-Subnet(172.16.48.0/20 |)

VIP features

-

Support obtaining client IP ⓘ

Supported

Project

Default Project

Tags

aprilzxyang:2 ()

Instance Deletion Protection

Not enabled [Enable instance deletion protection](#)

Configuration Change Protection

Not enabled [Enable configuration change protection](#)

- 2. Create a layer-7 HTTP or HTTPS listener.
- 3. Bind the listener with a IPv6 or IPv4 real server.

Bind with backend service

IP version ⓘ
☐ IPv6 ☒ IPv4

Target type ⓘ
☒ Instance ☐ IP type

Network
Default-VPC

Select an instance.

CVM

ENI

Container instance

Default port

Default weight

CVM name	Search by CVM name, and	
<input type="checkbox"/> Instance ID/name		
<input type="checkbox"/> ins-43-)		
<input type="checkbox"/> ins-43- (private)		

Press the Shift key to select more.

Selected (0)

Instance ID/name	Port	Weight ⓘ
No data yet		



Confirm

Cancel

Creating IPv6 NAT64 CLB Instances

Last updated : 2024-10-09 16:30:26

Note :

IPv6 NAT64 CLB can only be created in three regions: Beijing, Shanghai, and Guangzhou.

IPv6 NAT64 CLB does not support classic CLB.

The security groups bound to IPv6 NAT64 CLB support IPv4 addresses but do not currently support IPv6 addresses. The IPv6 NAT64 CLB facilitates access for both IPv6 and IPv4 clients. If there is a requirement for IPv4 client access, please [submit a ticket](#) to obtain an IPv4 VIP.

IPv6 implementations are still at the preliminary stage across the internet. In case of access failure, please [submit a ticket](#). SLA is not guaranteed during the beta test period.

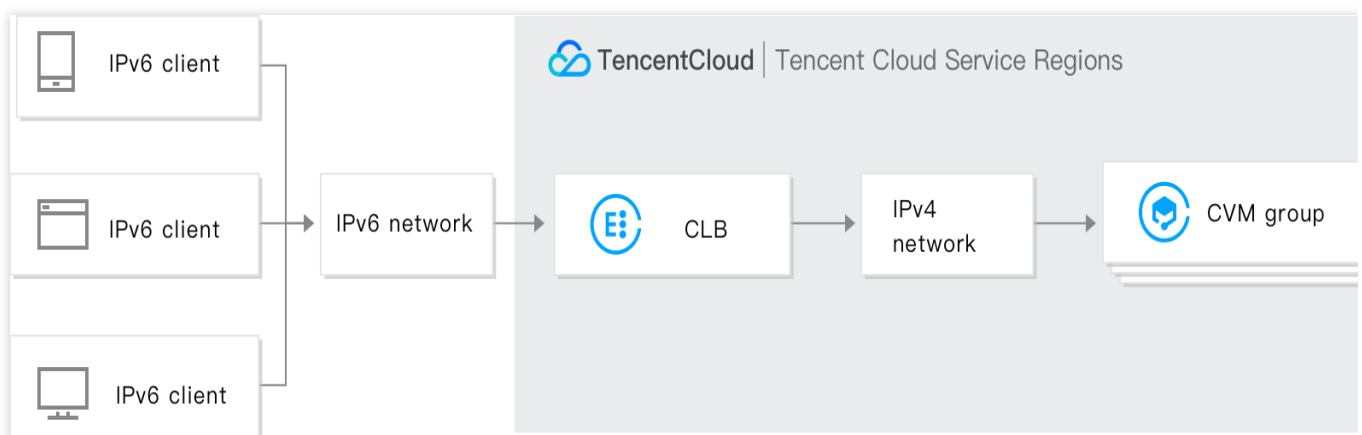
CLB supports creating IPv6 NAT64 CLB instances. Tencent Cloud will assign an IPv6 public IP address, i.e., VIP of the IPv6 edition, to an instance, and the VIP will forward requests from IPv6 clients to the real IPv4 CVM instance.

What Is an IPv6 NAT64 CLB Instance?

An IPv6 NAT64 CLB instance is a load balancer implemented based on the IPv6 NAT64 transitional technology. Through an IPv6 NAT64 CLB instance, real servers can be quickly accessed by IPv6 users without any IPv6 modification required.

IPv6 NAT64 CLB Architecture

The IPv6 NAT64 CLB architecture is as shown below.



When IPv6 NAT64 CLB is accessed from an IPv6 network, CLB can smoothly convert IPv6 addresses to IPv4 addresses to adapt to existing services.

IPv6 NAT64 CLB Advantages

Tencent Cloud IPv6 NAT64 CLB has the following advantages when helping your business quickly connect to IPv6:

Quick connection: CLB enables connection to IPv6 in a matter of seconds and is available upon purchase.

Smooth business transition: In order to smoothly transit your business to IPv6, you only need to transform the client with no modifications required for real servers. IPv6 NAT64 CLB supports access from IPv6 clients and converts IPv6 messages into IPv4 messages. IPv6 transition is imperceptible to applications on real servers, which still work in their original way.

Ease of use: IPv6 NAT64 CLB is compatible with IPv4 CLB flowchart and easy to use with no additional learning costs incurred.

Operation Guide

Creating an IPv6 NAT64 CLB instance

1. Log in to the Tencent Cloud console and go to the [CLB purchase page](#).

2. Select options for the following parameters correctly:

Billing Mode: Supports pay-as-you-go billing.

Region: Only Beijing, Shanghai, and Guangzhou are supported.

Instance Type: CLB.

Network Type: Public network.

IP version: IPv6 NAT64.

Network: VPC.

Other configurations are the same as general instance configurations.

3. After configuring the above items, click **Buy now**, and return to [Instance Management](#) page where you can view the IPv6 CLB instance you just purchased.

Using IPv6 NAT64 CLB

Log in to the [CLB Console](#) and click an instance ID to enter the details page. On the **Listener Management** tab, you can configure listeners and forwarding rules, and bind CVM instances. For more information, see [Getting Started with CLB](#).

Instance Management

Guangzhou(8)

Shanghai

Nanjing

Beijing

Chengdu

Chongqing

Taipei, China

Hong Kong, China

Singapore

Bangkok

Mumbai

Seoul

Tokyo

Silicon Valley

Virginia

Toronto

Frankfurt

Moscow

Cloud Load Balancer(7)

Classic Cloud Load Balancer(1)

Create


Delete

Change Project

Edit Tags

Project: All Projects

Use " " to split more than one keyword: Q

<input type="checkbox"/>	ID/Name *	Monito...	Status	VIP	Networ... ▾	Network	Health Status	Project ▾	Tag	Operati...
<input type="checkbox"/>			Normal	<div>72 (IPv6 NAT64)</div>	Public Network		Health check not enabled (Configuration)	DEFAULT PROJECT	-	<div>Configure More</div>

References

[Obtaining Real Client IPs via TOA in Hybrid Cloud Deployment](#)

Configure CLB Forwarding Domain Name

Last updated : 2024-04-02 09:41:48

When a client initiates a request, the Cloud Load Balancer forwards the request to the backend server according to the configured listener forwarding rules. The domain name in the listener forwarding rules is the domain name used by your backend service. This document explains how to configure a domain name.

Directions

Step 1: Register a Domain Name

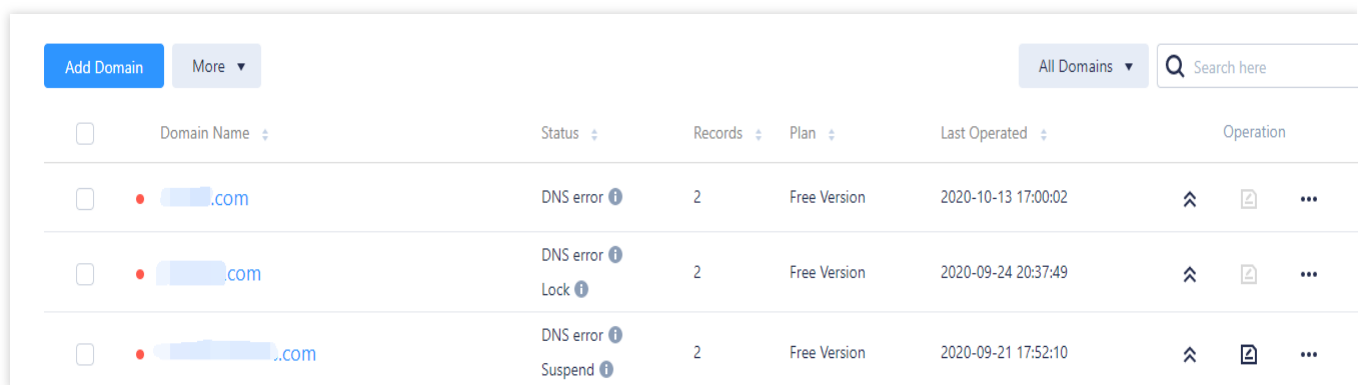
Domain registration is the foundation of establishing services on the Internet.

If you already have your own domain name with another registrar, you can transfer the domain name to Tencent Cloud Domain Services. For details, please see [Domain Transfer In](#).

If you do not have a domain name yet, you need to register a domain name. For details, please see [Domain Registration](#).

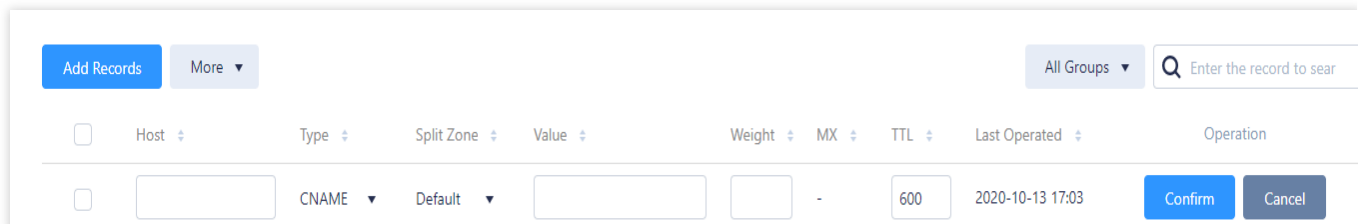
Step 2: Add Domain Name Resolution

1. Log in to the [DNSPod Console](#).
2. In **My Domains**, click the domain name that requires CNAME record forwarding to enter its **Record Management** page. As shown below:



<input type="checkbox"/>	Domain Name	Status	Records	Plan	Last Operated	Operation
<input type="checkbox"/>	.com	DNS error ⓘ	2	Free Version	2020-10-13 17:00:02	⬆️ 📄 ...
<input type="checkbox"/>	.com	DNS error ⓘ Lock ⓘ	2	Free Version	2020-09-24 20:37:49	⬆️ 📄 ...
<input type="checkbox"/>	.com	DNS error ⓘ Suspend ⓘ	2	Free Version	2020-09-21 17:52:10	⬆️ 📄 ...

3. Click **Add Records** and fill in the following record information. As shown below:



Host: fill in a subdomain. For example, when adding a record for `www.dnspod.com`, you can simply select **www** in the **Host** field. If you only want to add a record for `dnspod.com`, select **@** in the **Host** field. The **@** CNAME record will affect the normal resolution of MX records, please add it with caution.

Type: select **CNAME**.

Split Zone: select **Default**, otherwise some users may not be able to resolve it.

For example, if you want to point China Unicom users to `2.com` and all other users to `1.com`. You can implement this by adding two CNAME records, one with the split zone **Default** and record value `1.com`, and the other with the split zone **China Unicom** and record value `2.com`.

Value: you can only fill in a domain name pointed to by the CNAME.

Weight: a split zone with identical hosts. It can be set for different record values, and the resolved content will be returned according to the set weight ratio during DNS resolution. Enter an integer between 0 and 100.

MX: leave it empty.

TTL: cache time. The smaller the value, the faster the modification record will take effect in various regions. Default: 600s.

4. Click **Confirm**.

Step 3: Verify the Resolution Result

Note:

Resolution will take effect in about 10 minutes.

After completing the above steps, you can enter the CNAME domain (such as `www.example.com`) in the browser to test if the domain resolves correctly.

Configuring CLB Security Group

Last updated : 2024-09-02 10:45:55

After a CLB instance is created, you can configure a CLB security group to isolate public network traffic. This document describes how to configure CLB security groups in different modes.

Use Limits

Each CLB can be bound with up to 5 security groups. To increase the quota, go to [Increase Quota](#) to submit a request.

A single CLB security group supports up to 512 rules, including outbound rules, inbound rules, and backend parameter templates (ipm/ipmg/ppm/ppmg) fully expanded.

Cross-Region Binding 2.0 and Hybrid Cloud Deployment do not support Bypass Security Group. You should allow the client IP and service port on the real server.

After a private network CLB is bound with an EIP, the security groups of new CLB instances will take effect for the traffic from both the EIP and the private network CLB, while the security groups of existing CLB instances will take effect only for the traffic from the private network CLB. If the existing instances require the security groups to take effect for the traffic from the EIP, you can [submit a ticket](#) for request.

Classic private network CLBs and classic network-based private network CLBs do not support binding security groups

Classic private network CLBs and classic network-based CLBs do not support the Bypass Security Group feature. CPM 2.0 currently does not support the Bypass Backend Security Group feature for security groups.

Background

A security group is a virtual firewall that can filter stateful data packets and control outbound and inbound traffic at the instance level. For more information, please see [Security Group](#).

A CLB security group is bound to a CLB instance, while a CVM security group is bound to a CVM instance. They target at different objects. For a CLB security group, you can choose to:

[Enable Bypass Backend Security Group](#)

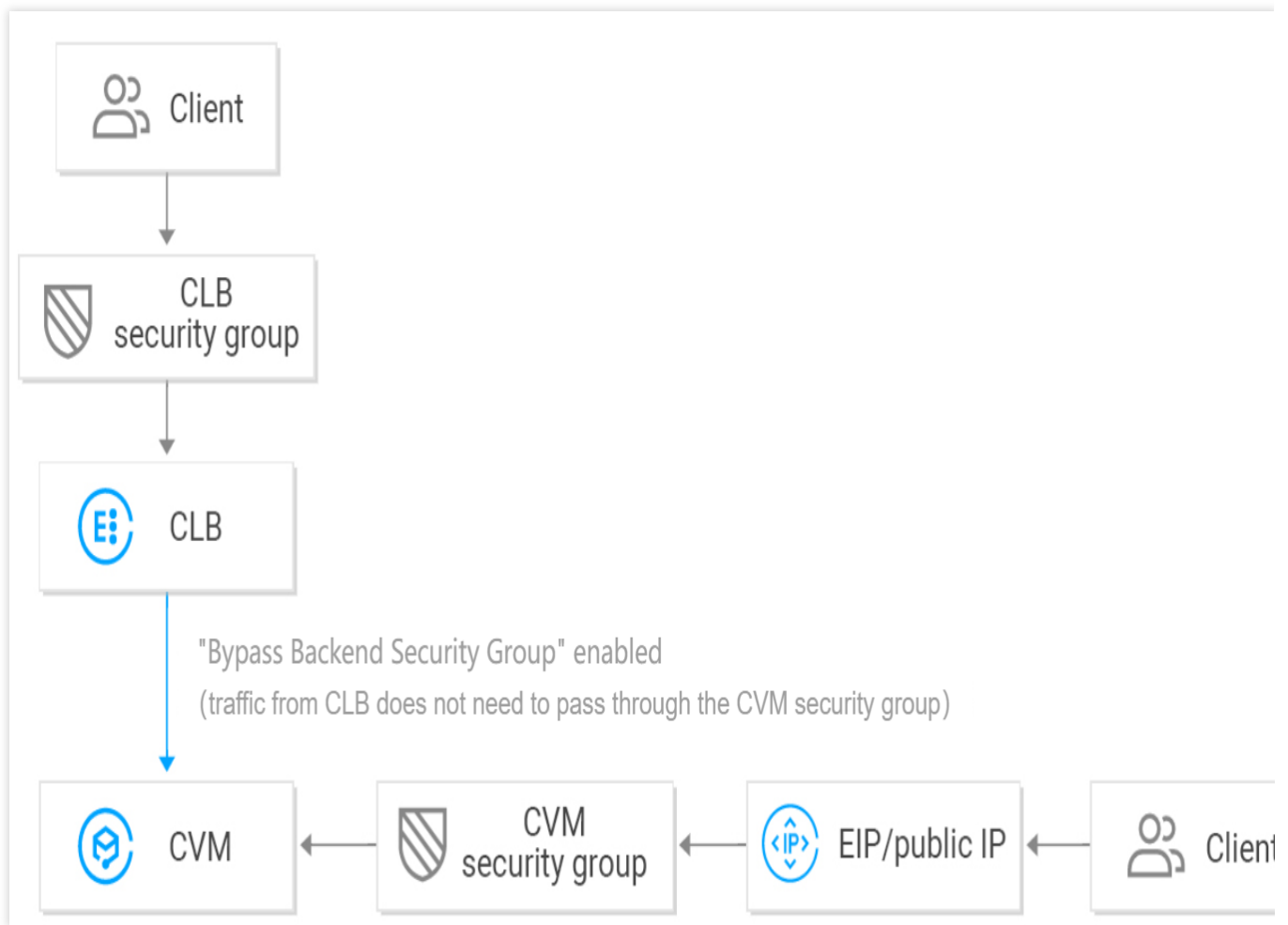
[Disable Bypass Backend Security Group](#)

Note:

For IPv4 CLB security groups, **Bypass Backend Security Group** is disabled by default, you can enable it in the console.

For IPv6 CLB security groups, **Bypass Backend Security Group** is enabled by default and you cannot disable it.

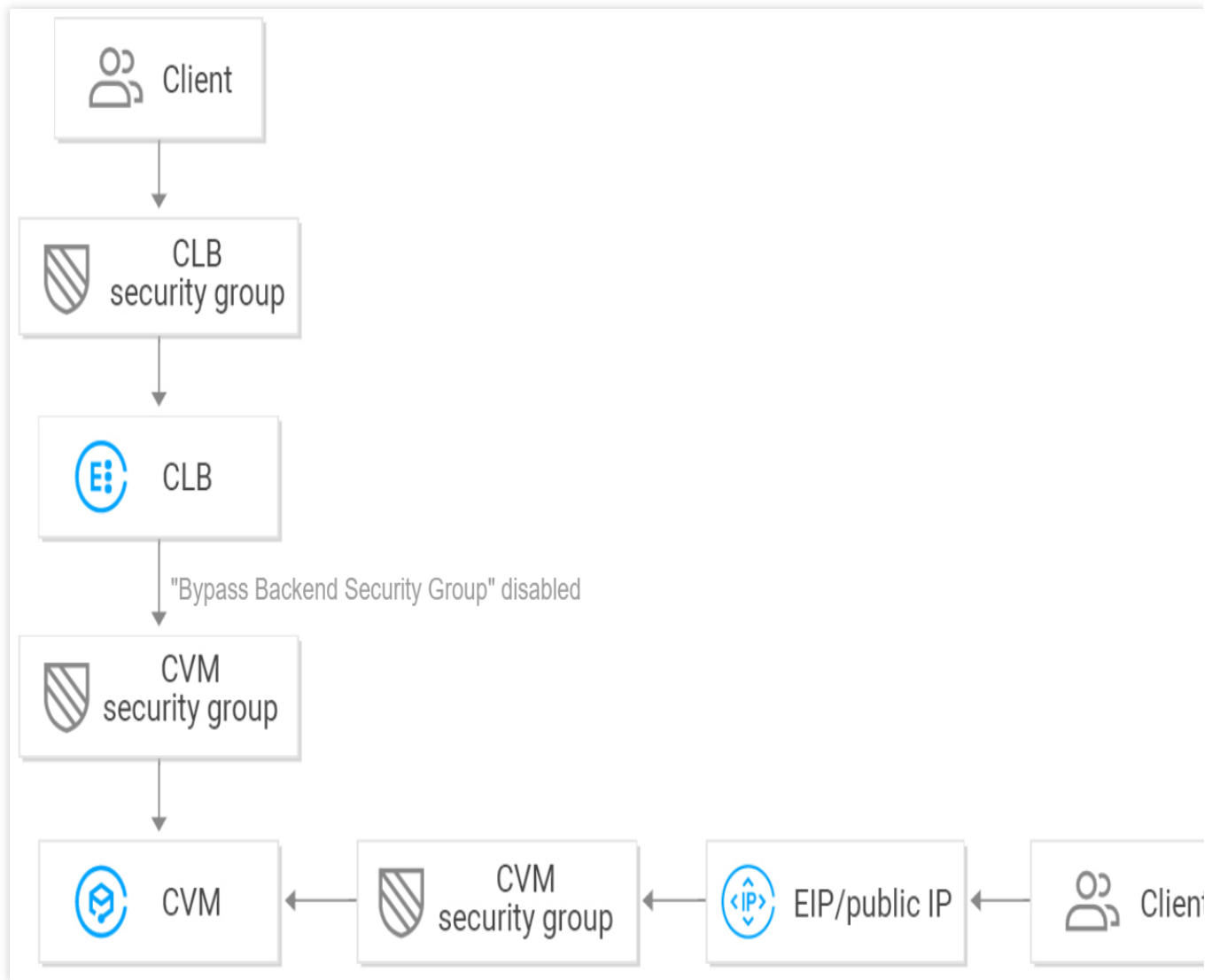
Enabling Bypass Backend Security Group



When Bypass Backend Security Group is enabled:

If you want to allow access only from a specified client IP, you need to allow it and the listening port in the CLB security group, however you don't need to allow the client IP and service port in the backend CVM security group. Access traffic from the CLB only pass through the CLB security group, as the real server allows traffic from CLB by default. Traffic from public IPs (including general public IPs and EIPs) still needs to pass through the CVM security group. If a CLB instance has no security group configured, all traffic will be allowed, and only ports configured with listeners on the VIP of the CLB instance can be accessed; therefore, the listening port will allow traffic from all IPs. To reject traffic from a specified client IP, you need to configure in the CLB security group. Rejecting a client IP in the CVM security group takes effect only for traffic from public IPs (including general public IPs and EIPs) but not for traffic from CLB.

Disabling Bypass Backend Security Group



When Bypass Backend Security Group is disabled:

If you want to only allow access from the specified client IP, you need to allow the client IP and listening port in the CLB security group and also allow the client IP and service port in the CVM security group; therefore, business traffic passing through CLB will be double checked by both the CLB security group and CVM security group.

Traffic from public IPs (including general public IPs and EIPs) still needs to pass through the CVM security group.

If a CLB instance has no security group configured, only traffic passing through the CVM security group will be allowed.

You can reject access either the CLB security group or the CVM security group to reject traffic from a specified client IP.

When Bypass Backend Security Group is disabled, the CVM security group should be configured as follows to ensure effective health check:

1. Configure public network CLB

You need to allow the CLB VIP on the backend CVM security group, so that CLB can use the VIP to detect the backend CVM health status.

2. Configure private network CLB

For private network CLB (formerly "private network application CLB"), if your CLB instance is in a VPC, the CLB VIP needs to be allowed in the backend CVM security group for health check; if your CLB instance is in the classic network, no additional configuration is needed as the health check IP is allowed by default.

For private network classic CLB, if your CLB instance was created before December 5, 2016 and is in a VPC, the CLB VIP needs to be allowed (for health check) in the backend CVM security group; otherwise, no additional configuration is needed as the health check IP is allowed by default.

Directions

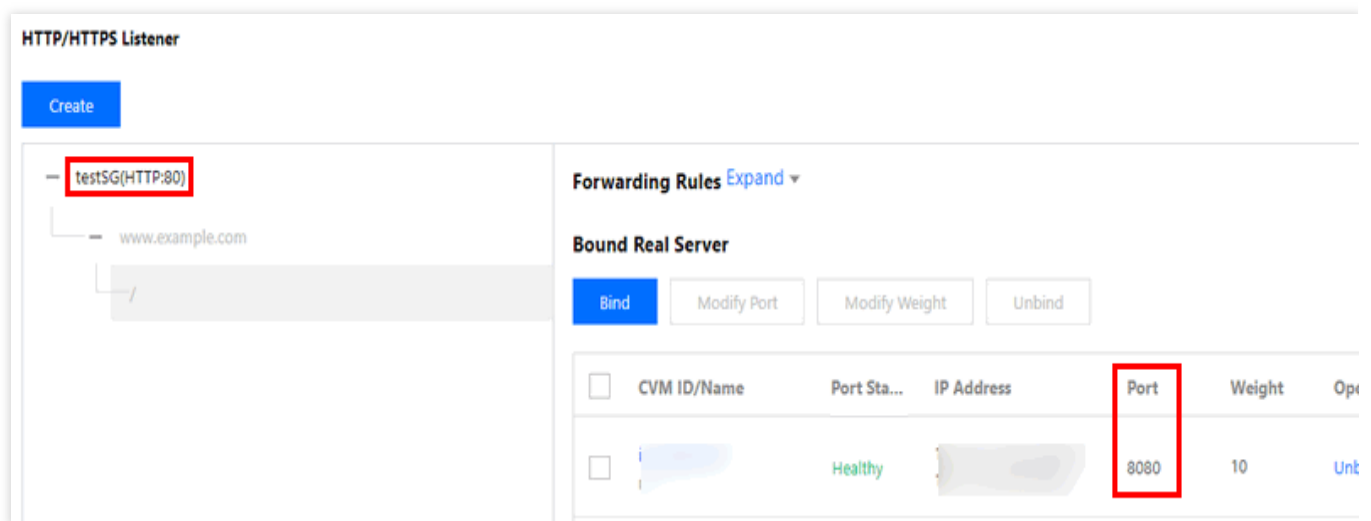
In the following example, the security group is configured to only allow inbound traffic to the CLB from port 80, and the service is provided via CVM port 8080. There is no limit upon the client IPs.

Note:

For the public network CLB instance used in this example, the CLB VIP needs to be allowed in the backend CVM security group for health check. The current IP is set to `0.0.0.0/0`, which means all IPs are allowed.

Step 1. Create a CLB instance and listener, and bind them to a CVM

For more information, please see [Getting Started with CLB](#). An HTTP:80 listener is created and bound to a backend CVM instance whose service port is 8080 in this example.



Step 2. Configure a CLB security group

1. Configure a CLB security group rule

Log in to the [Security Group Console](#) to configure a security group rule. In the inbound rule, allow requests from port 80 of all IPs (i.e., `0.0.0.0/0`) and reject traffic from other ports.

Note:

Security group rules are screened to take effect from top to bottom. If the new rule is put into effect, other rules will be denied by default; therefore, pay attention to their order. For more information, see [Security Group Overview](#). A security group has inbound and outbound rules. The above configuration is intended to restrict inbound traffic and is therefore an **inbound rule**, while the outbound rule does not need to be specially configured.

Type	Source ⓘ	Protocol port ⓘ	Policy	Notes
Custom ▼	0.0.0.0/0	TCP:80	Allow ▼	

+ New Line

Completed Cancel

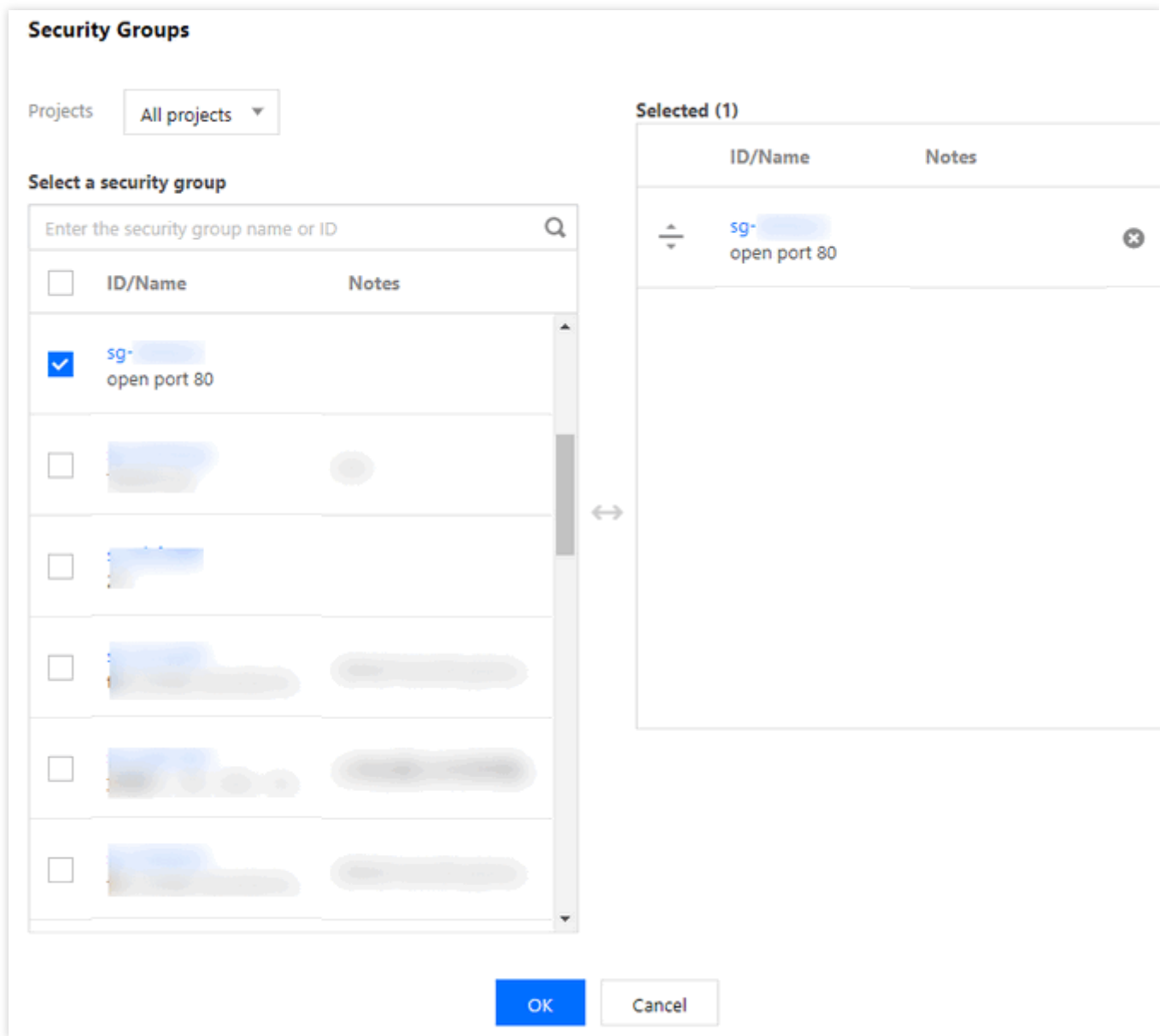
2. Bind the security group to the CLB instance

2.1 Log in to the [CLB Console](#).

2.2 On the "Instance Management" page, click the ID of the target CLB instance.

2.3 On the instance details page, click the **Security Group** tab and click **Bind** in the **Bound Security Groups** module.

2.4 In the **Configure Security Group** window that pops up, select the security group bound to the CLB instance and click **OK**.



The CLB security group configuration is complete, which only allows access to CLB from port 80.

Step 3. Configure Bypass Backend Security Group

You can choose to enable or disable **Bypass Backend Security Group** with different configurations as follows:

Method 1. Enable **Bypass Backend Security Group**, so that the real server does not need to allow the port.

Note:

This feature is not supported for classic private network CLB and CLB in the classic network.

Method 2. Disable **Bypass Backend Security Group**, and you also need to allow the client IP (0.0.0.0/0 in this example) in the CVM security group.

Method 1. Enable Bypass Backend Security Group

1. Log in to the [CLB Console](#).
2. On the **Instance Management** page, click the ID of the target CLB instance.

3. On the instance details page, click the **Security Group** tab.
4. On the **Security Group** tab, click



to enable **Bypass Backend Security Group**.

5. When **Bypass Backend Security Group** is enabled, only security group rules in the **rule preview** as shown below need to be verified.

Basic information Listener management Redirection configurations Monitoring **Security groups**

Enable Bypass Backend Security Group ⓘ ☒

Once enabled, traffic is allowed by default between CLB and CVM, and traffic from CLB only needs to be checked by the security group on CLB; if not enabled, traffic from CLB needs to be checked by the security groups on both CLB and CVM. When CLB is not bound to a security group, its listening port will allow traffic from all IPs by default. See [the documentation](#).

CPM 2.0 currently does not support the Bypass Backend Security Group feature for security groups.

Bound security groups			Sort: Configuration
Priority ⓘ	Security group ID/name	Operation	
No security groups configured			

Rule preview
Inbound rules Outbound rules
No inbound rules now

Method 2. Disable Bypass Backend Security Group

If **Bypass Backend Security Group** is disabled, you need to allow the client IP in the CVM security group. Business traffic is allowed to access CVM only from CLB port 80 and use services provided by CVM port 8080.

Note:

To allow traffic from a specified client IP, you need to allow the IP in both the CLB security group and CVM security group. If the CLB does not have a security group, please allow the IP in the CVM security group.

1. Configure a CVM security group rule

A CVM security group can be configured to only allow access from service ports for traffic accessing the backend CVM instance.

Go to the [Security Group Console](#) to configure a security group policy. In the inbound rule, all port 8080 of all IPs. To ensure smooth remote CVM login and ping services, open 22, 3389, and ICMP services in the security group.

2. Bind the security group to the CVM instance

2.1 In the [CVM Console](#), click the ID of CVM instance bound to the CLB instance to enter the details page.

2.2 Select the **Security Group** tab and click **Bind** in the **Bound Security Groups** module.

2.3 In the **Configure Security Group** window that pops up, select the security group bound to the CVM instance and click **OK**.

Binding Private Network CLB to EIP

Last updated : 2024-10-09 16:50:51

Private network CLB is used to distribute requests from Tencent Cloud's private network. It doesn't have a public IP and cannot communicate with the public network. If you need to use a private network CLB instance and want it to communicate with the public network, you can bind it to an EIP for public network access.

Note:

The feature of binding an EIP to a private network CLB instance is in beta testing. To try it out, please [submit a ticket](#).

Use Limits

Region limits

Private network CLB instances are unavailable in Ji'nan, Fuzhou, Shijiazhuang, Wuhan, and Changsha regions. Therefore, this feature is not supported in these regions.

Product attribute limits

This feature is supported only for bill-by-IP accounts but not bill-by-CVM accounts.

This feature is supported only for CLB instances but not classic CLB instances.

This feature is supported only for private network CLB instances in VPCs but not in the classic network.

Feature limits

Currently, private network CLB instances do not support port ranges.

A private network CLB instance can be bound to only an EIP that is in the same region as the CLB instance and not bound to other resources.

Each private network CLB instance can be bound to only one EIP.

After a private network CLB instance is bound to an EIP, its features will be similar to those of a public network CLB instance, but public network CLB cannot be split into private network CLB and EIP.

Security group limits

After a private network CLB is bound with an EIP, the security groups of new CLB instances will take effect for the traffic from both the EIP and the private network CLB, while the security groups of existing CLB instances will take effect only for the traffic from the private network CLB. If the existing instances require the security groups to take effect for the traffic from the EIP, you can [submit a ticket](#) for request.

After a private network CLB is bound with an EIP and enables Bypass Security Group, the traffic from the CLB only needs to be verified by the CLB security group. The security groups of a CVM bypass the traffic from its CLBs by default.

Directions

Method 1: Selecting an EIP when purchasing a CLB instance

1. Log in to the Tencent Cloud console and go to the CLB purchase page.
2. Specify the following CLB configuration items as needed. For more information about the configuration, see

Purchase Methods.

Parameter	Description
Billing Mode	Select the Pay-as-You-Go mode.
Region	Select a region. For more information about the regions supported by CLB, see Region List .
Instance Type	Only CLB instance type is supported.
Network Type	Select the Public network type.
EIP	Select EIP . Tencent Cloud will assign you an EIP and a private network CLB instance. Supported EIP types include general IP, accelerated IP, and static single-line IP.

Method 2: Binding a private network CLB instance to an EIP

1. Log in to the [CLB console](#) and click **Instance management** in the left sidebar.
2. Select a region in the top-left corner of the **Instance management** page, select the target private network CLB instance in the instance list, and choose **More > Bind EIP** in the **Operation** column on the right.
3. In the pop-up window, select the EIP to be bound and click **Submit** to bind the EIP to the private network CLB instance.

Note:

The accelerated IPs and static single-line IPs are in beta testing. To try them out, please [submit a ticket](#).

4. (Optional) Select the target private network CLB instance in the instance list and choose **More > Unbind EIP** in the **Operation** column on the right to unbind the instance from the EIP.

References

[AssociateAddress](#)

[Purchase Methods](#)

[Product Attribute Selection](#)

Starting or Stopping a CLB Instance

Last updated : 2025-05-16 10:18:09

You can start or stop instances. After an instance is stopped, it will no longer receive or forward traffic, perform health checks, or allow ping.

Note:

The feature is in a beta test. To try it out, please [submit a ticket](#).

Use Cases

If you have configured a large number of CLB instances, and some of them are temporarily unused for business considerations but cannot be deleted, you can choose to stop them.

After an instance is stopped, all its listeners will also be stopped, and it will no longer receive or forward traffic.

After an instance is started, all its listeners will also be started, and it will receive and forward traffic normally.

After a listener is stopped, it will no longer receive or forward traffic. After all listeners of an instance are stopped, the instance will be stopped.

After a listener is started, it will receive and forward traffic normally. After all listeners of an instance are started, the instance will be started.

After an instance is stopped, if any of its listeners is started, the instance will be started and receive and forward traffic normally by using the started listener, while other listeners will remain stopped.

Restrictions

This feature is not supported for classic CLB.

This feature is supported only by VPC but not by classic networks.

This feature is not supported for TLS 1.3 and earlier.

Prerequisites

You have created a CLB instance. For more information, see [Creating CLB Instances](#).

You have created a listener. For more information, see [Creating a Listener](#).

Directions

1. Log in to the [CLB Console](#).
2. Select a region in the top-left corner of the **Instance Management** page, find the target instance in the instance list, and choose **More > Start** or **More > Stop** in the **Operation** column on the right to start or stop the instance.
3. (Optional) On the **Listener Management** tab, find the target listener and click **Start listener** or **Stop listener**.

Cloning CLB Instances

Last updated : 2024-12-19 20:47:49

CLB supports instance cloning. This feature allows you to easily copy the configuration of existing CLB instances, including instance attributes, listeners, security groups, and logs.

Restrictions

Instance attribute restrictions

CLB instances without any billable items cannot be cloned.

Classic CLB instances and CLB with Anti-DDoS Pro cannot be cloned.

Classic network-based instances cannot be cloned.

Anycast instances cannot be cloned.

IPv6 NAT64 instances cannot be cloned.

Blocked or frozen instances cannot be cloned.

Before cloning an instance, make sure all certificates used on the instance are valid. Cloning will fail if there are any expired certificates.

Listener restrictions

A CLB instance associated with over 50 listeners cannot be cloned.

Shared instances with the public network bandwidth cap exceeding 2 Gbps cannot be cloned.

Cloning Instances in the Console

1. Log in to the [CLB console](#) and click **Instance management** in the left sidebar.
2. Select a region in the top-left corner of the **Instance management** page, find the instance to be cloned in the instance list, and choose **More > Clone** in the **Operation** column of the instance.
3. In the pop-up window, enter the name of the target instance and click **OK**.

Cloning Instances via API

For more information, see [CloneLoadBalancer](#).

Exporting CLB Instances

Last updated : 2024-10-09 17:01:05

You can export a list of CLB instances containing the configuration and resource usage details by specifying the region or other conditions.

Directions

1. Log in to the [CLB Console](#) and select a region in the top left corner of the **Instance Management** page.
2. In the instance list, select an instance and click



in the top right corner.

3. In the pop-up window, select the fields and scopes to export and click **Confirm** to download the instance list locally.

Export instances

Exported files:

☒ Export All

Instance field:

☒ ID

☒ Name

☒ Status

☒ VIP

☒ Network type

☒ Network

☒ ISP

☒ Instance Specification

☒ Billing Mode

☒ Bandwidth Cap

☒ Project

☒ Tags

☒ VIP features

☒ Bind with Custom

☒ Creation Time

Rule field:

☒ Listener ID, listener protocol, listener port, forwarding rule ID, forwarding domain, forwarding URL, CVM ID, RS IP, RS port, RS weight

Backend service type:

☒ Non-target group

☐ Target Group

In case some of the CLB's listeners are bound with the target group and the rest listeners don't, you need you export them separately.

Exported range:

☐ All Instances

☐ Only search results

☒ Only selected instances

Confirm

Cancel

Parameter	Description
Field	<div>The following fields can be exported:</div> <div>Instance field</div> <div>Rule field</div> <div>The "RS health status" of the rule field is visible only when the rule field is checked and the export scope is "Only selected instances".</div>
Scope	<div>The following scopes can be exported:</div> <div>All instances</div> <div>Only search results</div> <div>Only selected instances</div> <div>The field "Only selected instances" will be grayed out if no instance is selected.</div>

Upgrade to a LCU-supported instances

Last updated : 2024-10-09 17:02:24

CLB provides two types of instance specifications, shared instances and LCU-supported instances. By default, all CLB instances are shared instances when they are created. You can upgrade them to LCU-supported CLB instances.

Advantages of LCU-supported CLBs

A single shared instance can sustain up to 50,000 concurrent connections, 5,000 new connections per second, and 5,000 queries per second.

An LCU-supported instance supports up to Ten million concurrent connections, 1,000,000 new connections, and 300,000 QPS. You can [submit a ticket](#) to request for even higher capabilities.

Upgrade impact

Rate limiting

During the upgrade, The default bandwidth for intranet performance capacity instances corresponds to the upper limit of the specified configuration, which can be adjusted in the console after upgrading. The default bandwidth for public network performance capacity instances remains consistent with the pre-upgrade state, and can be modified in the console post-upgrade.

Upon upgrading, speed will be regulated according to the instance specifications. Exceeding these specifications will result in speed limitations and packet loss. For performance capacity type speed limitations, please refer to the following monitoring indicators. For more details, see [Monitoring Metrics](#).

ClientConcurConn (Client-to-CLB concurrent connections)

ClientNewConn (Client-to-CLB new connections)

TotalReq (Requests per second)

ClientOuttraffic (Client-to-CLB bandwidth out)

ClientIntraffic (Client-to-CLB bandwidth in)

Existing connections will not be affected if the maximum capability of upgraded instances is not exceeded.

Billing

The billing mode remains unchanged.

The LCU fee will be billed based on the number of LCUs used per hour. For more details, see [LCU Pricing](#).

Network connection

The upgrade does not interrupt network connections and can be completed within 1 minute.

Rollback

Degrading to shared CLB instances is not allowed.

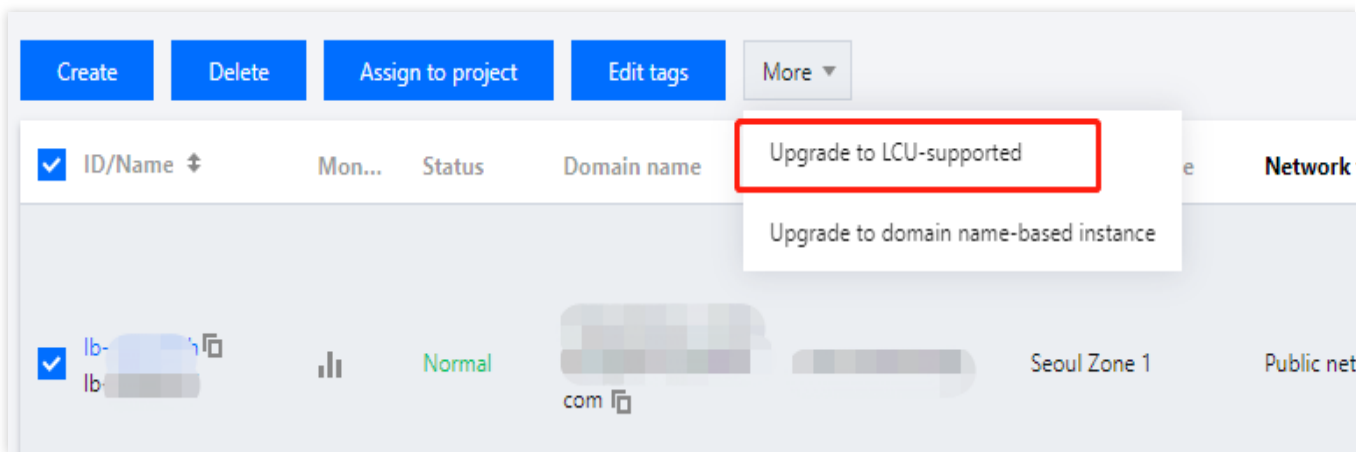
Limitations

Upgrading multiple pay-as-you-go shared CLB instances is supported.

Upgrading classic CLB instances is not allowed.

How to Upgrade

1. Log in to the [CLB console](#) and click **Instance management** in the left sidebar.
2. Select a shared CLB instance from the list and click **Upgrade to LCU-supported**.



3. Click **OK** in the **Upgrading to LCU-supported** pop-up window.

See also

[LCU Pricing](#)

Adjusting Specification of LCU-supported CLBs

Last updated : 2025-03-14 17:44:08

This document describes how to adjust the specification of LCU-supported CLBs. For more information about specifications, see [Instance Specifications Comparison](#).

Note:

If the selected specification is lower than the existing specification, the new specification may not meet the business requirements, which may bring packet loss due to rate limiting.

For pay-as-you-go LCU-supported CLBs, the selected specification is the maximum speed limit, and the LCU unit price remains unchanged.

Directions

1. Log in to the [CLB console](#) and click **Instance management** in the left sidebar.
2. On the **Instance management** page, select a region in the top-left corner. Locate the target instance in the instance list, and choose **More > Adjust specification** under **Operation**.



Click an LCU-supported CLB to open the **Basic information** page, and select **Billing information > Instance specification > Adjust specification**.

Billing information

Instance billing mode	Pay
Network billing mode	Bill by traffic
Bandwidth cap	0Mbps Adjust network configuration
Instance specification	Standard ⓘ Adjust specification
Creation time	2023-08-23 11:03:42

3. In the **Adjust specification** window, set the target specification, and click **Confirm**.

Note :

1. If the bandwidth limit of the selected target specification is lower than the current bandwidth limit, adjust the current bandwidth limit first, or select another specification.
2. If the selected target specification is lower than the existing one, there is a risk that the downgraded specification cannot meet business needs well. This may cause packet loss due to speed limitation and affect your business. Please confirm the adjustment operation once again.

Adjust specification

Current specifications Standard

Model	Max concurrent co...	New connections ...	Queries per second	Bandwidth cap (M...
<input type="radio"/> Higher I	200,000	20,000	20,000	4,096
<input checked="" type="radio"/> Higher II	500,000	50,000	30,000	6,144
<input type="radio"/> Super I	1,000,000	100,000	50,000	10,240
<input type="radio"/> Super II	2,000,000	200,000	100,000	20,480
<input type="radio"/> Super III	5,000,000	500,000	200,000	40,960
<input type="radio"/> Super IV	10,000,000	1,000,000	300,000	61,440

For the pay-as-you-go mode, the selected specification refers to the upper limit. The LCU price is the same. For details, see [Billing description](#)

Confirm

Cancel

Deleting CLB Instances

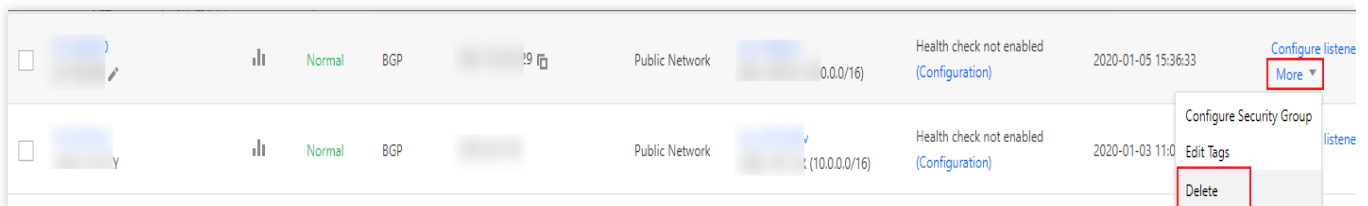
Last updated : 2024-10-09 17:07:18





After you confirm that a CLB instance has no traffic and is no longer needed, you can delete it via the CLB console or API.

Once deleted, the CLB instance will be completely terminated and cannot be recovered. We strongly recommend unbinding all real servers and observing for a while before deleting any instance.

Deleting CLB Instances in the Console

1. Log in to the [CLB console](#).
2. Find the CLB instance you want to delete and choose **More > Delete** in the **Operation** column of the instance.



<input type="checkbox"/>			Normal	BGP	9	Public Network	0.0.0/16	Health check not enabled (Configuration)	2020-01-05 15:36:33	Configure listen More
<input type="checkbox"/>			Normal	BGP		Public Network	(10.0.0.0/16)	Health check not enabled (Configuration)	2020-01-03 11:0	Configure Security Group Edit Tags Delete

3. In the pop-up window, click **Submit** to confirm the deletion after you read the operation security prompt.

The pop-up window is as shown below. We recommend that you delete the instance only after confirming that there are **0** bound rules, **none** bound real servers, and a green tick in the **Note** column.

Confirm to delete the following load balancers?

ID/Name	Bound rules	Bound CVM	Notes About Oper...
lb-2jrl6dv0 lb-162309	0	None	✓

Submit

Close

Deleting CLB Instances via API

For more information, see [DeleteLoadBalancer](#).

Releasing Idle CLB Instances

Last updated : 2024-01-04 09:44:16

When a pay-as-you-go instance is not configured with a listener or bound to a real server seven days after the creation, it's considered as an idle instance. To reduce unnecessary charges, please release idle instances in time.

Restrictions

CLB supports batch release of idle instances in the same region only.




Directions

Note:

The idle instance data can be cached for one day. Make sure that the instance to be released is not in use to prevent a release error.

1. Log in to the [CLB Console](#) and click **Idle Instance** on the left sidebar.
2. Select a region in the upper left corner of the idle instance page, find the target instance in the idle instance list, and click **Delete** in the **Operation** column on the right.
3. (Optional) Select all instances on the left side of the idle instance list, and click **Delete** at the top of the page.
4. In the pop-up window, confirm the instance information and click **OK**.

Are you sure you want to delete the following load balancers? ×

ID/Name	Bound rules	Bound CVM	Notes About Op...
lb-  lb- 	0	None	

Note: After it's deleted, CLB configurations including VIP, listener and forwarding rules will be removed permanently.

SubmitCancel

Configuring Deletion Protection

Last updated : 2024-10-09 17:09:48

This feature prevents an instance from being deleted and released by mistake.

Restrictions

When a CLB instance is terminated due to overdue payment, the instance will be released automatically even if deletion protection is enabled.

Directions

1. Log in to the [CLB console](#) and select your region in the top left corner of the **Instance management** page.
2. In the instance list, click the ID of the target instance.
3. On the page that appears, click **Enable Deletion Protection** on the **Basic configuration** tab.



Adjusting Instance Public Network Configurations

Last updated : 2024-10-09 17:11:53

You can adjust the bandwidth or the billing mode of public network CLB instances as needed in real time.

Restrictions

IPv4 CLB instances: network configuration adjustment is only supported for bill-by-IP accounts but not for bill-by-CVM accounts.

IPv6 CLB instances: network configuration adjustment is supported for both bill-by-IP and bill-by-CVM accounts.

For more information on checking your account type, please refer to [Checking Account Type](#).

Bandwidth Cap

Instance Billing Mode	Network Billing Mode	Bandwidth Cap Range (in Mbps)
Pay-as-you-go	Bill-by-bandwidth (hourly)	0 - 2048 (inclusive)
	Bill-by-traffic	
	Bandwidth package	

Note:

If you need to set a higher bandwidth cap, please [submit a ticket](#) or contact your Tencent Cloud sales rep.

Adjusting Bandwidth

1. Log in to the [CLB console](#).
2. On the **Instance Management** page, select a region, and click **More** -> **Adjust Bandwidth** on the right of a public network CLB instance.
3. In the dialog box, set the bandwidth cap and click **Submit**.

Changing Billing Mode

1. Log in to the [CLB console](#).
2. On the **Instance Management** page, select a region, click **More** on the right of a public network CLB instance and continue to adjust the network billing mode.

Instance Billing Mode	Network Billing Mode	Adjustment
Pay-as-you-go	Bill-by-bandwidth (hourly)	Add IP to a bandwidth package: instance billing mode remains the same; network billing mode is switched to using a bandwidth package; each instance can have its billing mode switched once only.
	Bill-by-traffic	Switch to monthly subscription: instance billing mode is switched to monthly subscription; network billing mode is switched to bill-by-bandwidth (monthly); each instance can have its billing mode switched once only. Add IP to a bandwidth package: instance billing mode remains the same; network billing mode is switched to using a bandwidth package; each instance has unlimited chances for switching billing modes.
	Bandwidth package	Remove IP from bandwidth package: instance billing mode remains the same; network billing mode is switched to bill-by-traffic; each instance has unlimited chances for switching billing modes.

3. Click **Submit** in the pop-up dialog box.

CLB Listener

CLB Listener Overview

Last updated : 2024-10-10 10:33:09

After creating a CLB instance, you need to configure a listener for it. The listener listens to requests on the instance and distributes traffic to real servers based on the load balancing policy.

You need to configure a CLB listener with the following items:

1. Listening protocol and port. The listening port, or frontend port, is used to receive and forward requests to real servers.
2. Listening policies, such as the load balancing policy and [session persistence](#).
3. [Health check](#) policies.
4. Real server. Bind a real server by selecting its IP address and port. A service port, or backend port, is used by the real server to receive requests.

Supported Protocol Types

A CLB listener can listen to layer-4 and layer-7 requests on a CLB instance and route them to real servers for processing. The main difference between layer-4 CLB and layer-7 CLB is whether layer-4 protocol (such as TCP or UDP) or layer-7 protocol (such as HTTP or HTTPS) is used to forward traffic for load balancing of user requests.

Layer-4 protocols: Transport layer protocols that receive requests and forward traffic to the real server mainly via VIP and port.

Layer-7 protocols: Application layer protocols that distribute traffic based on application layer information such as URL and HTTP header.

If you use a layer-4 listener (i.e., layer-4 protocol forwarding), the CLB instance will establish a connection with the real server on the listening port, and directly forward requests to the real server. This process does not modify any data packets (in pass-through mode) and has high forwarding efficiency.

Tencent Cloud CLB supports request forwarding over the following protocols:

- TCP (transport layer)
- UDP (transport layer)
- TCP SSL (transport layer)
- QUIC (transport layer)
- HTTP (application layer)
- HTTPS (application layer)

Note:

TCP SSL listeners currently not support classic CLB instances.

--	--	--	--

Protocol Type	Protocol	Description	Use Case
Layer-4 protocol	TCP	<p>Connection-oriented and reliable transport layer protocol:</p> <p>The source and destination ends must perform a three-way handshake to establish a connection before data transfer.</p> <p>Session persistence based on the client IP address (source IP address) is supported.</p> <p>The client IP address can be found at the network layer.</p> <p>The server can directly obtain the client IP address.</p>	<p>TCP is suitable for scenarios that have high requirements for reliability and data accuracy but relatively low requirements for transfer speed, such as file transfer, receiving and sending emails, and remote login. For more information, see Configuring a TCP Listener.</p>
	UDP	<p>Connection-less transport layer protocol:</p> <p>The source and destination ends do not establish a connection, nor maintain the connection status.</p> <p>Each UDP connection is point-to-point. One-to-one, one-to-many, many-to-one, and many-to-many communications are supported.</p> <p>Session persistence based on the client IP address (source IP address) is supported.</p> <p>The server can directly obtain the client IP address.</p>	<p>UDP is suitable for scenarios that have high requirements for transfer efficiency but relatively low requirements for accuracy, such as instant messaging and online videos. For more information, see Configuring a UDP Listener.</p>
	TCP SSL	<p>Secure TCP:</p> <p>TCP SSL listeners support configuring certificates to block unauthorized access. Unified certificate management is supported for CLB to implement decryption.</p> <p>One-way authentication and mutual authentication are supported.</p> <p>The server can directly obtain the client IP address.</p>	<p>TCP SSL is suitable for scenarios that have high requirements for security when TCP is used and supports TCP-based custom protocols. For more information, see Configuring a TCP SSL Listener.</p>
	QUIC	<p>UDP-based multiplexing concurrent transport layer protocol:</p> <p>QUIC implements reliable data transmission, security and HTTP2 over</p>	<p>QUIC is suitable for audio and video services, game services, etc. When the network is unstable, such as frequent switching between 4G network and Wi-Fi</p>

		UDP, and is comparable to the combination of TCP, TLS, and HTTP2. In a QUIC connection, no matter what happens to the IP address or port, the connection will not be interrupted, enabling seamless connection migration.	network, it can smoothly migrate and connect services without interruption. For more information, see Configuring a QUIC Listener .
Layer-7 protocol	HTTP	Application layer protocol: Forwarding based on the request domain name and URL is supported. Cookie-based session persistence is supported.	HTTP is suitable for applications that need to identify request content, such as web applications and mobile applications. For more information, see Configuring an HTTP Listener .
	HTTPS	Encrypted application layer protocol: Forwarding based on the request domain name and URL is supported. Cookie-based session persistence is supported. Unified certificate management is supported for CLB to implement decryption. One-way authentication and mutual authentication are supported.	HTTPS is suitable for HTTP applications that require encrypted transmission. For more information, see Configuring an HTTPS Listener .

Port Configuration

Port Type	Description	Restrictions
Listening port (frontend port)	Listening ports are used by CLB instances to receive and forward requests to real servers. You can configure CLB instances for ports 1 to 65535, such as port 21 (FTP), port 25 (SMTP), port 80 (HTTP), and port 443 (HTTPS).	On one CLB instance: Listening ports of UDP can be used for TCP. For example, a <code>TCP:80</code> listener and a <code>UDP:80</code> listener can coexist. Listening ports must be unique for the same type of protocol. TCP, TCP SSL, HTTP, and HTTPS are all TCP protocols, so a <code>TCP:80</code> listener and an <code>HTTP:80</code> listener cannot coexist.
Service port (backend port)	Service ports are used by real servers to provide services, receive and process traffic from CLB instances. On one CLB instance, one listening port can forward traffic to ports of multiple real servers.	On one CLB instance: Listeners using different protocols can be bound to the same service port. For example, listener <code>HTTP:80</code> and listener

HTTPS : 443 can be bound to the same port of a real server.

When using the same listening protocol, each real server port can be bound to only one listener, that is, the quadruple (VIP, listening protocol, private IP address of the real server, and real server port) must be unique.

References

[Use Limits](#)

Configuring TCP Listener

Last updated : 2025-04-25 11:46:09

You can create a TCP listener for a Cloud Load Balancer (CLB) instance to forward TCP requests from the client. TCP is suitable for scenarios that have high requirements for reliability and data accuracy but relatively low requirements for transmission speed, such as file transfer, email messaging, and remote login. Real servers bound to the TCP listener can directly obtain the real client IP address.

Prerequisites

You have created a CLB instance. For more information, see [Creating CLB Instances](#).

Directions

Step 1. Configure a listener

1. Log in to the [CLB console](#) and click **Instance management** in the left sidebar.
2. Select your region in the top-left corner of the CLB instance list and click **Configure listener** in the **Operation** column of the target instance.

3. Under **TCP/UDP/TCP SSL/QUIC listener**, click **Create** and configure the TCP listener in the pop-up window.

3.1 Configure basic parameters

Parameter	Description	Example
Name	Listener name.	test-tcp-80
Listening protocol and port	Listening protocol: In this case, select <code>TCP</code> . Listening port: The port used to receive requests and forward them to the real server. The port number ranges from 1 to 65535. A listening port must be unique in the same CLB instance.	TCP:80
Balancing method	CLB supports two scheduling algorithms for TCP listeners: weighted round robin (WRR) and weighted least connections (WLC). WRR: Requests are distributed to real servers in sequence based on their weights. This algorithm performs scheduling based on the number of new connections. Servers with higher weights are more likely to be scheduled and servers with the same weight process the same number of connections.	WRR

	<p>WLC: Loads of servers are estimated based on the number of active connections to the servers. This algorithm performs scheduling based on server loads and weights. For servers with the same weight, those with less loads are more likely to be scheduled.</p> <p>Note: If WLC is selected, the listener does not support session persistence.</p>	
Configure ProxyProtocol	<p>After checking, you can enable the ProxyProtocol configuration. It supports carrying the client's source address to the real server through the ProxyProtocol protocol. Currently, the ProxyProtocol configuration feature is in beta test. If needed, submit a ticket application.</p>	Selected
Two-way RST	<p>If this option is selected, corresponding operations will send RST packets to both ends (client and server) to close the connection; otherwise, two-way RST packets will not be sent, and the persistent connection will exist until it times out.</p>	Selected

3.2 Configure health check

For more information about health check, see [TCP Listener](#).

3.3 Configure session persistence

Parameter	Description	Example
Session persistence	<p>After session persistence is enabled, a CLB listener will distribute access requests from the same client to the same real server.</p> <p>TCP session persistence is implemented based on the client IP address. The access requests from the same IP address are forwarded to the same real server. Session persistence can be enabled for WRR scheduling but not WLC scheduling.</p>	Enabled
Hold Time	<p>Session persistence duration.</p> <p>Session persistence is terminated if there are no new requests in the connection within the specified duration.</p> <p>Value range: 30-3600 seconds</p>	30 seconds

Step 2. Bind a real server

1. On the **Listener management** page, click the created listener `TCP : 80` to view the bound real servers on the right of the listener.
2. Click **Bind**, select the target real server, and configure the server port and weight in the pop-up window.

Note:

If you set **Default port** first and then select real servers, the port of every real server is the default port.

Step 3. Configure a security group (optional)

You can configure a CLB security group to isolate public network traffic. For more information, see [Configuring a CLB Security Group](#).

Step 4. Modify or delete a listener (optional)

If you need to modify or delete a created listener, click the listener on the **Listener management** page and click

for modification or

for deletion.

Configuring a UDP Listener

Last updated : 2025-04-25 11:46:09

You can create a UDP listener to a CLB instance to forward UDP requests from the client. UDP is suitable for scenarios that have high requirements for transfer speed but relatively low requirements for accuracy, such as instant messaging and online videos. For UDP listeners, the real server can directly get the real client IP address.

Restrictions

Port 4789 of the UDP listener is a system reserved port and unavailable yet.

Prerequisites

You have created a CLB instance. For more information, see [Creating CLB Instances](#).

Directions

Step 1. Configure a listener

1. Log in to the [CLB console](#) and click **Instance management** in the left sidebar.
2. Select your region in the top-left corner of the CLB instance list and click **Configure listener** in the **Operation** column of the target instance.

3. Under **TCP/UDP/TCP SSL/QUIC listener**, click **Create** and configure the UDP listener in the pop-up window.

3.1 Configure basic parameters

Parameter	Description	Example
Name	Listener name.	test-udp-8000
Listening protocol and port	Listening protocol: In this case, select <code>UDP</code> . Listening port: The port used to receive requests and forward them to real servers. The port number ranges from 1 to 65535. Port 4789 is reserved for the system and unavailable yet. A listening port must be unique in the same CLB instance.	UDP:8000

Balancing method	<p>CLB supports two scheduling algorithms for UDP listeners: weighted round robin (WRR) and weighted least connections (WLC).</p> <p>WRR: Requests are distributed to real servers in sequence based on their weights. This algorithm performs scheduling based on the number of new connections. Servers with higher weights are more likely to be scheduled and servers with the same weight process the same number of connections.</p> <p>WLC: Loads of servers are estimated based on the number of active connections to the servers. This algorithm performs scheduling based on server loads and weights. For servers with the same weight, those have less loads are more likely to be scheduled.</p> <p>Note: If WLC is selected, the listener does not support session persistence.</p>	WRR
Schedule by QUIC ID	<p>Once this feature is enabled, CLB will schedule client requests by QUIC ID, so requests with the same QUIC Connection ID will be scheduled to the same real server. If a request doesn't have a QUIC Connection ID, it will be downgraded to normal WRR scheduling, i.e., scheduling according to the quadruple (source IP address + destination IP address + source port + destination port).</p>	Enabled
Configure ProxyProtocol	<p>After checking, you can enable the ProxyProtocol configuration. It supports carrying the client's source address to the real server through the ProxyProtocol protocol.</p>	Check and use

3.2 Configure health check

For more information about health check, see [UDP Listener](#).

3.3 Configure session persistence

Parameter	Description	Example
Session persistence	<p>After session persistence is enabled, a CLB listener will distribute access requests from the same client to the same real server.</p> <p>UDP session persistence is implemented based on the client IP address. The access requests from the same IP address are forwarded to the same real server.</p> <p>Session persistence can be enabled for WRR scheduling but not WLC scheduling.</p>	Enabled
Hold Time	<p>Session persistence duration.</p> <p>Session persistence is terminated if there are no new requests in the connection within the specified duration.</p> <p>Value range: 30-3600 seconds</p>	30 seconds

Step 2. Bind a real server

1. On the **Listener management** page, click the created listener `UDP:8000` to view the bound real servers on the right of the listener.
2. Click **Bind**, select the target real server, and configure the server port and weight in the pop-up window.

Note:

If you set **Default port** first and then select real servers, the port of every real server is the default port.

Step 3. Configure a security group (optional)

You can configure a CLB security group to isolate public network traffic. For more information, see [Configuring a CLB Security Group](#).

Step 4. Modify or delete a listener (optional)

If you need to modify or delete a created listener, click the listener on the **Listener management** page and click

for modification or

for deletion.

Configuring TCP SSL Listener

Last updated : 2024-10-10 10:42:30

You can create a TCP SSL listener for a Cloud Load Balancer (CLB) instance to forward encrypted TCP requests from the client. TCP SSL is applicable to scenarios where ultra-high performance and large-scale TLS offloading are required. Real servers bound to the TCP SSL listener can directly obtain the real client IP address.

Note:

TCP SSL listeners currently support CLB instances but not classic CLB instances.

Use Cases

TCP SSL is suitable for scenarios that have high requirements for security when the TCP protocol is used:

TCP SSL listeners support configuration of certificates to block unauthorized access.

Unified certificate management is supported for CLB to implement decryption.

One-way authentication and mutual authentication are supported.

A real server can directly obtain the client IP address.

Prerequisites

You have created a CLB instance. For more information, see [Creating CLB Instances](#).

Directions

Step 1. Configure a listener

1. Log in to the [CLB console](#) and click **Instance management** in the left sidebar.
2. Select your region in the top-left corner of the CLB instance list and click **Configure listener** in the **Operation** column of the target instance.

ID/Name	Mon...	Status	VIP	Availability Z...	Network ...	Carrier	Instance Spe...	Health Status	Billing Mode	Tag	Custom Con...	Operation
<input type="checkbox"/> lb-...		Normal		Beijing Zone 4	Public Network	BGP	Dedicated	Health check not enabled Configuration	Pay-as-you-go — bandwidth Created at 2022-01-11 11:32		-	Configure Listen More

3. Under **TCP/UDP/TCP SSL/QUIC listener**, click **Create** and configure the TCP SSL listener in the pop-up window.

3.1 Configure basic parameters

Parameter	Description	Example
Name	Listener name.	test-tcpssl-9000
Listening protocol and port	Listening protocol: In this case, select <code>TCP SSL</code> . Listening port: The port used to receive requests and forward them to the real server. The port number ranges from 1 to 65535. A listening port must be unique in the same CLB instance.	TCP SSL:9000
SSL parsing	One-way authentication and mutual authentication are supported.	One-way authentication
Server certificate	You can select an existing certificate in the SSL Certificate Service console or create a certificate.	Select an existing certificate.
Balancing method	CLB supports two scheduling algorithms for TCP SSL listeners: weighted round robin (WRR) and weighted least connections (WLC). WRR: Requests are distributed to real servers in sequence based on their weights. This algorithm performs scheduling based on the number of new connections. Servers with higher weights are more likely to be scheduled and servers with the same weight process the same number of connections. WLC: Loads of servers are estimated based on the number of active connections to the servers. This algorithm performs scheduling based on server loads and weights. For servers with the same weight, those have less loads are more likely to be scheduled.	WRR

3.2 Configure health check

For more information, see [TCP SSL Listener](#).

3.3 Configure session persistence

TCP SSL listeners don't support session persistence currently.

Step 2. Bind a real server

1. On the **Listener management** page, click the created listener `TCP SSL: 9000` to view the bound real servers on the right of the listener.
2. Click **Bind**, select the target real server, and configure the server port and weight in the pop-up window.

Note:

If you set **Default port** first and then select real servers, the port of every real server is the default port.

Step 3. Configure a security group (optional)

You can configure a CLB security group to isolate public network traffic. For more information, see [Configuring a CLB Security Group](#).

Step 4. Modify or delete a listener (optional)

If you need to modify or delete a created listener, click the listener on the **Listener management** page and click



for modification or



for deletion.

Configuring a QUIC Listener

Last updated : 2024-10-10 15:29:20

You can create a QUIC listener to a CLB instance to forward encrypted QUIC requests from the client. For QUIC listeners, the real server can directly get the real client IP.

QUIC (Quick UDP Internet Connection) is a transport layer network protocol designed by Google, multiplexing concurrent data streams using UDP. Compared with the popular TCP+TLS+HTTP2 protocol, QUIC has the following advantages:

- Establish a connection faster

- Improve congestion control.

- Adopt multiplex to avoid head-of-line (HOL) blocking.

- Support connection migration.

Use Cases

A QUIC listener supports connection migration. When your network changes, such as frequent switches between mobile and Wi-Fi networks, it can smoothly migrate the connections without interruption. This is suitable for audio/video services, game services, etc.

Restrictions

- QUIC listeners are not available for classic CLB.

- QUIC listeners are not available for CLBs deployed on the classic network.

- Only IPv4 and IPv6 NAT64 CLB instances support the QUIC listener.

Prerequisites

Create a CLB instance as instructed in [Creating CLB Instances](#).

Directions

Step 1. Configure a listener

1. Log in to the [CLB console](#) and click **Instance Management** on the left sidebar.

2. Select a region in the top-left corner of the CLB instance list page and click **Configure Listener** in the **Operation** column on the right.

ID/Name	Mon...	Status	VIP	Availability Z...	Network ...	Carrier	Instance Spe...	Health Status	Billing Mode	Tag	Custom Con...	Operation
lb- lb-		Normal		Beijing Zone 4	Public Network	BGP	Dedicated	Health check not enabled Configuration	Pay-as-you-go — bandwidth Created at 2022-01-11 11:32			Configure Listen More

3. Under **TCP/UDP/TCP SSL/QUIC Listener**, click **Create** and configure the QUIC listener in the **Create Listener** pop-up window.

3.1 Basic Configuration

Configuration Item	Description	Example
Name	Listener name.	test-quic-443
Listener Protocol and Ports	<p>Listener protocol: Select QUIC. CLB can receive QUIC requests made by clients, but TCP is still used between CLB and real server.</p> <p>Listener port: It's used to receive requests and forward them to the real server. Port range: 1-65535. The listener port must be unique in the same CLB instance.</p>	QUIC:443
SSL parsing	One-way authentication and mutual authentication are supported.	One-way authentication
Server certificate	You can select an existing certificate in the SSL certificate console or create a certificate.	Existing certificate
Balancing method	<p>For QUIC listeners, CLB supports two scheduling algorithms: weighted round robin (WRR) and weighted least connections (WLC).</p> <p>WRR: Requests are sequentially delivered to different real servers according to their weights. Scheduling is done based on the number of new connections, where servers with higher weights will undergo more polls (i.e., a higher probability), while servers with the same weight process the same number of connections.</p> <p>WLC: Loads of servers are estimated according to the number of active connections to the servers. Scheduling is done based on server loads and weights. If their weights are the same, servers with fewer active connections will undergo more polls (i.e., a higher probability).</p>	WRR

3.2 Health check

For details of health check, see [Configuring Health Check](#).

3.3 Session persistence

QUIC listeners don't support session persistence currently.

Step 2. Bind a backend server

1. On the **Listener Management** page, click the created listener `QUIC:443` to view the bound real servers on the right of the listener.
2. Click **Bind**, select the target real server, and configure the server port and weight in the pop-up window.

Note:

Default port: If you select the **Default Port** first and then select the real servers, all real servers use the default port.

Step 3. Configure a security group

You need to configure a CLB security group to isolate public network traffic. For more information, see [Configuring CLB Security Group](#).

Step 4. Modify or delete a listener (optional)

If you need to modify or delete a listener, click the listener on the **Listener Management** page and click



for modification or



for deletion.

References

[Using QUIC Protocol on CLB](#)

Configuring an HTTP Listener

Last updated : 2024-10-10 15:38:26

You can create an HTTP listener to a CLB instance to forward HTTP requests from the client. HTTP is suitable for applications where request contents need to be identified, such as web applications and mobile apps.

Prerequisites

You have created a CLB instance. For more information, see [Creating CLB Instances](#).

Directions

Step 1. Configure a listener

1. Log in to the [CLB console](#) and click **Instance management** in the left sidebar.
2. Select your region in the top-left corner of the CLB instance list and click **Configure listener** in the **Operation** column of the target instance.

<input type="checkbox"/>	ID/Name	Mon...	Status	VIP	Availability Z...	Network ...	Carrier	Instance Spe...	Health Status	Billing Mode	Tag	Custom Con...	Operation
<input type="checkbox"/>	lb- lb-		Normal		Beijing Zone 4	Public Network	BGP	Dedicated	Health check not enabled Configuration	Pay-as-you-go — bandwidth Created at 2022-01-11 11:32		-	Configure Listen More

3. Under **HTTP/HTTPS listener**, click **Create** and configure the HTTP listener in the pop-up window.

3.1 Create a listener

Parameter	Description	Example
Name	Listener name.	test-http-80
Listening protocol and port	Listening protocol: In this case, select <code>HTTP</code> . Listening port: The port used to receive requests and forward them to a real server. Port range: 1-65535. A listening port must be unique in the same CLB instance.	HTTP:80
Enable persistent connection	Once this feature is enabled, persistent connections will be used between a CLB instance and real servers, and the CLB instance will no longer pass through the source IP address that can be obtained from XFF. To ensure normal forwarding,	Disabled

enable the "Allow Traffic by Default" feature in the CLB security group or allow `100.127.0.0/16` in the CVM security group.

Note:

Once this feature is enabled, the number of the connections between a CLB instance and real servers will fluctuate in the range of [QPS,QPS*60], subject to the connection reuse rate. If there is a limit on the maximum number of connections, we recommend you be cautious when enabling this feature. This feature is currently in beta test. To try it out, [submit a ticket](#).

The IP range 100.64.0.0/10 is already allowed as the health check source IP. You don't need to allow IPs within this range again.

3.2 Create a forwarding rule

Parameter	Description	Example
Domain name	Forwarding domain name: Length: 1-80 characters. It cannot begin with underscores (_). Exact and wildcard domain names are supported. Regular expressions are supported. For more information, see Layer-7 Domain Name Forwarding and URL Rules .	www.example.com
Default Domain	If all domain names of a listener are not matched, the system distributes requests to the default domain name, making default access controllable. Each listener can be configured with only one default domain name.	Enabled by default
URL	Forwarding URL: Length: 1-200 characters. Regular expressions are supported. For more information, see Layer-7 Domain Name Forwarding and URL Rules .	/index
Balancing method	For HTTP listeners, CLB supports three scheduling algorithms: weighted round robin (WRR), weighted least connections (WLC), and IP Hash. WRR: Requests are distributed to real servers in sequence based on their weights. This algorithm performs scheduling based on the number of new connections. Servers with higher weights are more likely to be scheduled and servers with the same weight process the same number of connections. WLC: Loads of servers are estimated based on the number of active connections to the servers. This algorithm performs scheduling based on server loads and weights. For servers with the same weight, those have less loads are more likely to be scheduled.	WRR

	IP Hash: This algorithm uses a request source IP address as the Hash key to locate the corresponding server in the static hash table. If a server is available and not overloaded, requests will be distributed to it; otherwise, a null value will be returned.	
Get client IP	Enabled by default	Enabled
Gzip compression	Enabled by default	Enabled

3.3 Configure health check

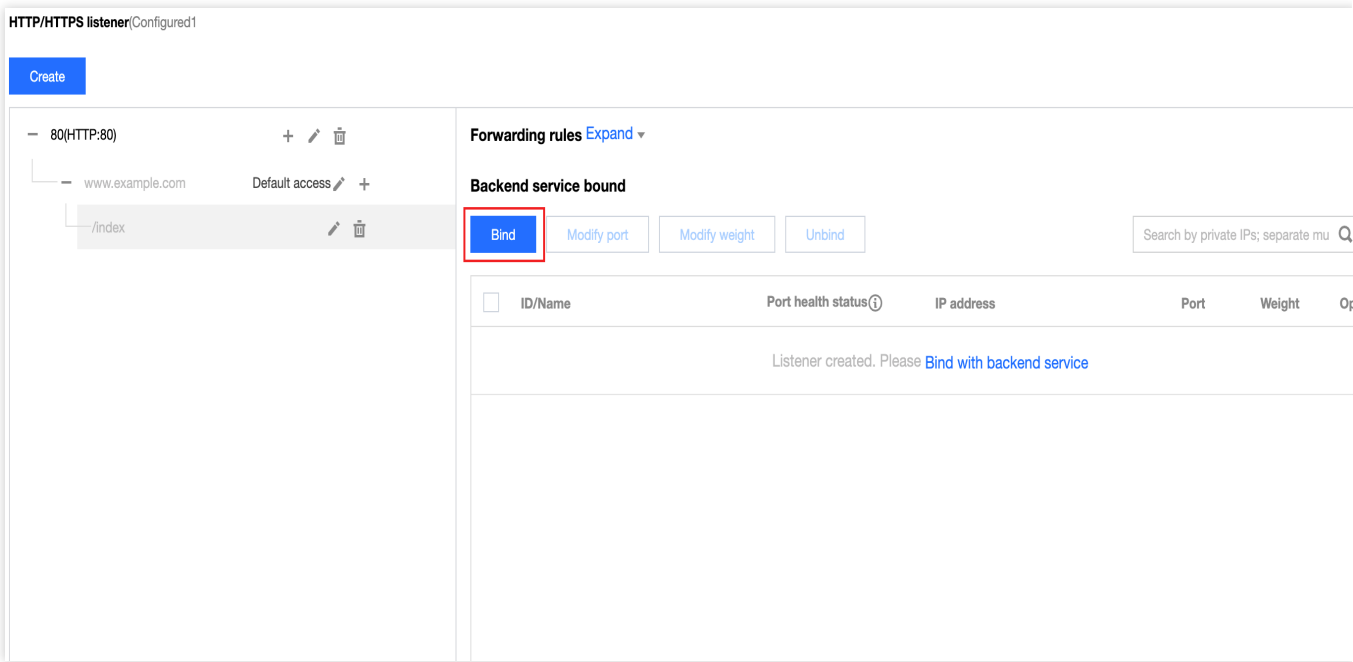
For more information, see [HTTP Listener](#).

3.4 Configure session persistence

Parameter	Description	Example
Session persistence	After session persistence is enabled, a CLB listener will distribute access requests from the same client to the same real server. TCP session persistence is implemented based on the client IP address. The access requests from the same IP address are forwarded to the same real server. Session persistence can be enabled for WRR scheduling but not WLC scheduling.	Enabled
Hold Time	Session persistence is terminated if there are no new requests in the connection within the specified duration. Value range: 30-3600 seconds	30 seconds

Step 2. Bind a real server

1. On the **Listener management** page, select the created listener `HTTP:80`. Click **+** on the left to expand the domain names and URL paths, select the desired URL path, and view the real servers bound to the path on the right of the listener.
2. Click **Bind**, select the target real server, and configure the server port and weight in the pop-up window.



Note:
If you set **Default port** first and then select real servers, the port of every real server is the default port.

Step 3. Configure a security group (optional)

You can configure a CLB security group to isolate public network traffic. For more information, see [Configuring a CLB Security Group](#).

Step 4. Modify or delete a listener (optional)

If you need to modify or delete a created listener, click the listener on the **Listener management** page and click

 for modification or

 for deletion.

Configuring HTTPS Listener

Last updated : 2024-10-10 15:46:47

You can create an HTTPS listener for a CLB instance to forward HTTPS requests from the client. HTTPS is suitable for HTTP applications where data transfer needs to be encrypted.

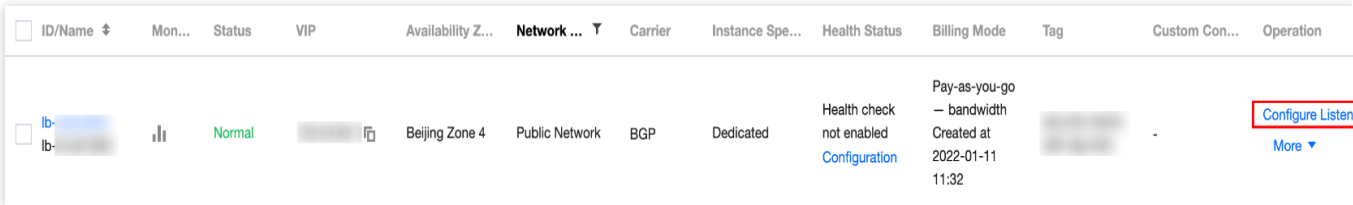
Prerequisites

You have created a CLB instance. For more information, see [Creating CLB Instances](#).

Directions

Step 1. Configure a listener

1. Log in to the [CLB console](#) and click **Instance management** in the left sidebar.
2. Select your region in the top-left corner of the CLB instance list page and click **Configure listener** in the **Operation** column of the target instance.



3. Under **HTTP/HTTPS listener**, click **Create** and configure the HTTPS listener in the pop-up window.

3.1 Create a listener

Parameter	Description	Example
Name	Listener name.	test-https-443
Listening protocol and port	Listening protocol: In this case, select <code>HTTPS</code> . Listening port: The port used to receive requests and forward them to a real server. Port range: 1-65535. A listening port must be unique in the same CLB instance.	HTTPS:443
Enable persistent connection	Once this feature is enabled, persistent connections will be used between a CLB instance and real servers, and the CLB instance will no longer pass through the source IP address that can be obtained from XFF. To ensure normal forwarding, enable the "Allow Traffic by Default" feature in the CLB security group or allow <code>100.127.0.0/16</code> in the CVM security group. Note:	Disabled

	<p>Once this feature is enabled, the number of the connections between a CLB instance and real servers will fluctuate in the range of [QPS,QPS*60], subject to the connection reuse rate. If there is a limit on the maximum number of connections, we recommend you be cautious when enabling this feature. This feature is currently in beta test. To try it out, please submit a ticket.</p> <p>The IP range 100.64.0.0/10 is already allowed as the health check source IP. You don't need to allow IPs within this range again.</p>	
Enable SNI	If SNI is enabled, multiple domain names of a listener can be configured with different certificates; if it is disabled, multiple domain names of a listener can be configured with one certificate only.	Disabled
SSL parsing	One-way authentication and mutual authentication are supported. CLB takes over the overheads of SSL encryption and decryption to guarantee the access security.	One-way authentication
Server certificate	<p>You can select an existing certificate in the SSL Certificate Service console or upload a certificate. You can configure two certificates that use different encryption algorithms.</p> <p>Note: You can configure two certificates only for CLB but not classic CLB. After two certificates are configured, you cannot enable QUIC.</p>	Select an existing certificate.
CA certificate	You can select an existing certificate in the SSL Certificate Service console or upload a certificate.	Select an existing certificate.

3.2 Create a forwarding rule

Parameter	Description	Example
Domain name	<p>Forwarding domain name: Length: 1 to 80 characters. A domain name cannot start with underscores (_). Exact and wildcard domain names are supported. Regular expressions are supported. For detailed configuration rules, see Layer-7 Domain Name Forwarding and URL Rules.</p>	www.example.com
Default Domain	<p>If all domain names of a listener are not matched, the system distributes requests to the default domain name, making default access controllable. Each listener can be configured with only one default domain name.</p>	Enabled
HTTP 2.0	After HTTP 2.0 is enabled, CLB instances can receive HTTP 2.0 requests. CLB instances access real servers over HTTP 1.1 no matter what HTTP version the client uses to access CLB instances.	Enabled

URL	Forwarding URL: Length: 1 to 200 characters. Regular expressions are supported. For detailed configuration rules, see Layer-7 Domain Name Forwarding and URL Rules .	/index
Balancing method	For HTTP listeners, CLB supports three scheduling algorithms: weighted round robin (WRR), weighted least connections (WLC), and IP Hash. WRR: Requests are distributed to real servers in sequence based on their weights. This algorithm performs scheduling based on the number of new connections . Servers with higher weights are more likely to be scheduled and servers with the same weight process the same number of connections. WLC: Loads of servers are estimated based on the number of active connections to the servers. This algorithm performs scheduling based on server loads and weights. For servers with the same weight, those have less loads are more likely to be scheduled. IP Hash: This algorithm uses a request source IP address as the Hash key to locate the corresponding server in the static hash table. If a server is available and not overloaded, requests will be distributed to it; otherwise, a null value will be returned.	WRR
Backend Protocol	Backend protocol is used between a CLB instance and a real server: If HTTP is selected as the backend protocol, the HTTP service must be deployed on the real server. If HTTPS is selected as the backend protocol, the HTTPS service must be deployed on the real server. In this case, the encryption and decryption of the HTTPS service will consume more resources on the real server. If gRPC is selected as the backend protocol, the gRPC service must be deployed on the real server. You can select gRPC as the backend forwarding protocol only when HTTP2.0 is enabled and QUIC is disabled.	HTTP
Get client IP	Enabled by default.	Enabled
Gzip compression	Enabled by default.	Enabled

3.3 Configure HTTPS health check

For more information, see [HTTPS Health Check Overview](#).

3.4 Configure session persistence

Parameter	Description	Example

Session persistence	After session persistence is enabled, a CLB listener will distribute access requests from the same client to the same real server. TCP session persistence is implemented based on client IP address. The access requests from the same IP address are forwarded to the same real server. Session persistence can be enabled for WRR scheduling but not WLC scheduling.	Enabled
Hold Time	Session persistence is terminated if there are no new requests in the connection within the specified duration. Value range: 30-3600 seconds	30 seconds

Step 2. Bind a real server

1. On the **Listener management** page, select the created listener `HTTPS : 443` . Click **+** on the left to expand the domain names and URL paths, select the desired URL path, and view the real servers bound to the path on the right of the listener.
2. Click **Bind**, select the target real server, and configure the server port and weight in the pop-up window.

Note:

If you set **Default port** first and then select real servers, the port of every real server is the default port.

Step 3. Configure a security group (optional)

You can configure a CLB security group to isolate public network traffic. For more information, see [Configuring a CLB Security Group](#).

Step 4. Modify or delete a listener (optional)

If you need to modify or delete a created listener, click the listener on the **Listener management** page and click



for modification or



for deletion.

Load Balancing Methods

Last updated : 2024-10-10 15:50:34

A load balancing method is an algorithm that allocates traffic to [real servers](#). Each method produces different load balancing effects.

Weighted Round-Robin Scheduling

The weighted round-robin scheduling algorithm is to schedule requests to different servers based on polling. It can solve problems with imbalanced performance of different servers. It uses weight to represent the processing performance of a server and schedules requests to different servers by weight in a polling manner. It schedules servers based on the number of new connections, where servers with a higher weight receive connections earlier and have a higher chance to be polled. Servers with the same weight will process the same number of connections.

Advantage: this algorithm features simplicity and high practicability. It does not need to record the status of all connections and is therefore a stateless scheduling algorithm.

Disadvantage: this algorithm is relatively simple, so it is unsuitable for situations where the service time of a request changes significantly, or each request needs to consume different amounts of time. In these cases, it will cause imbalanced load distribution among servers.

Applicable scenario: this algorithm is suitable for scenarios where each request consumes basically the same amount of time on the backend with the best loading performance. It is usually used in non-persistent connection services such as HTTP service.

Recommendation: if you know that each request consumes basically the same amount of time on the backend (for example, requests processed by a real server are of the same type or similar types), you are recommended to use weighted round-robin scheduling. If the time difference between each request is small, this algorithm is also recommended as it has low consumption and high efficiency with no need of traversal.

Weighted Least-Connection Scheduling

In actual situations, the time requests from the client spend staying on the server may vary greatly. As the working time gets longer, if a simple round-robin or random load balancing algorithm is used, the number of connection processes on each server may vary hugely, which cannot achieve load balancing effect.

Contrary to round-robin scheduling, least-connection scheduling is a dynamic scheduling algorithm that estimates the load of a server by its active connection quantity. The scheduler needs to record the number of current established connections on each server. If a request is scheduled to a server, the number of connections will be increased by 1. If a connection stops or times out, the number of connections will be decreased by 1.

In the weighted least-connection scheduling algorithm that is based on least-connection scheduling, different weights are allocated to servers according to their processing capability. In this way, a server can receive a corresponding number of requests according to its weight, which is an improvement on least-connection scheduling.

Note:

Suppose that the weight of a real server is w_i , and the current number of connections is c_i . The c_i/w_i values of each server are calculated in sequence. The real server with the smallest c_i/w_i value will be the next server that receives a new request. If there are real servers with the same c_i/w_i value, they will be scheduled based on weighted round-robin scheduling.

Advantage: this algorithm is suitable for requests requiring long-time processing, such as FTP.

Disadvantage: due to API restrictions, least-connection and session persistence cannot be enabled at the same time.

Applicable scenario: this algorithm is suitable for scenarios where the time used by each request on the backend varies greatly. It is usually used in persistent connection services.

Recommendation: if you need to process different requests and the service time needed by them on the backend varies greatly (such as 3 milliseconds and 3 seconds), you are recommended to use weighted least-connection scheduling to achieve load balancing.

Source Hashing Scheduling

The source hashing scheduling algorithm (`ip_hash`) uses the source IP address of the request as the hash key and finds the corresponding server from the statically assigned hash table. The request will be sent to this server if it is available and not overloaded; otherwise, null will be returned.

Advantage: `ip_hash` can map requests from a client to the same real server through the hash table. Therefore, in scenarios where session persistence is not supported, it can be used to achieve simple session persistence effect.

Recommendation: this algorithm calculates the hash value of the source address of a request and distributes the request to the matched real server based on its weight. In this way, all requests from the same client IP can be distributed to the same server. This algorithm is suitable for the protocols that do not support cookie.

Choosing Load Balancing Algorithm and Configuring Weight

In order to allow real server clusters to undertake business in a stable manner in different scenarios, some cases regarding how to choose the load balancing algorithm and configure weight are provided below for your reference.

Scenario 1:

1.1 Suppose that there are 3 real servers with the same configuration (CPU and memory) and you set all their weights to 10 as they have the same performance.

1.2 100 TCP connections have been established between each real server and the client, and a new real server is added.

1.3 In this scenario, you are recommended to use the least-connection scheduling algorithm, which can quickly increase the load of the 4th real server and reduce the pressure on the other 3 ones.

Scenario 2:

1.1 Suppose that you use Tencent Cloud services for the first time and your website was just built with low load. You are recommended to purchase real servers of the same configuration since they are all equivalent access-layer servers.

1.2 In this scenario, you can set the weights of all real servers to the default value of 10 and use the weighted round-robin scheduling algorithm to distribute the traffic.

Scenario 3:

1.1 Suppose that you have 5 real servers that undertake simple access requests to static pages, and the ratio of computing power (calculated by CPU and memory) of these servers is 9:3:3:3:1.

1.2 In this scenario, you can set the weight of the real servers to 90, 30, 30, 30, and 10, respectively. As most access requests to static web pages are of non-persistent connection type, you can use the weighted round-robin scheduling algorithm, so that the CLB instance can allocate requests based on the servers' performance ratio.

Scenario 4:

1.1 Suppose that you have 10 real servers to undertake massive amounts of web access requests and do not want to purchase more servers as that will increase the expenditure, and one of the servers often restarts due to overload.

1.2 In this scenario, you are recommended to set the weights of existing servers based on their performance and set a relatively small weight to servers with high load. In addition, you can use the least-connection scheduling algorithm to allocate requests to real servers with fewer active connections so as to avoid server overload.

Scenario 5:

1.1 Suppose that you have 3 real servers for processing some persistent connections, the ratio of computing power (calculated by CPU and memory) of these servers is 3:1:1.

1.2 The server with the best performance processes more requests, but you do not want it to be overloaded and want to allocate new requests to idle servers.

1.3 In this scenario, you can use the least-connection scheduling algorithm and appropriately reduce the weight of the busy server, so that the CLB instance can allocate requests to real servers with fewer active connections, thereby achieving load balancing.

Scenario 6:

1.1 Suppose that you want subsequent requests from the client to be allocated to the same server. As weighted round-robin or weighted least-connection scheduling cannot ensure that requests from the same client are allocated to the same server,

1.2 To satisfy the requirements of your specific application server and maintain the "stickiness" (or "continuity") of the client sessions, you can use ip_hash to distribute the traffic. This algorithm can ensure that all requests from the same client will be distributed to the same real server, unless the number of servers changes or the server becomes unavailable.

Session Persistence

Last updated : 2024-10-10 15:56:09

Session persistence can forward requests from the same IP to a single real server. By default, a CLB instance will route requests to different real server instances for load balancing. However, you can use session persistence to route requests from a specified user to the same real server instance. This enables applications that require session persistence (such as shopping cart) to run properly.

Layer-4 Session Persistence

Layer-4 protocols (TCP/UDP) support source IP address-based session persistence. The session persistence duration can be set to any integer (in seconds) between 30 and 3,600. If the time threshold is exceeded and the session has no new requests, session persistence will end. Session persistence is subject to the load balancing mode: In the mode of **Weighted round robin** where requests are distributed based on the weight of real servers, source IP address-based session persistence is supported.

In the mode of **Weighted least connections** where scheduling is performed based on server load and weight, session persistence is not supported.

Layer-7 Session Persistence

Layer-7 protocols (HTTP/HTTPS) support session persistence based on cookie insertion (CLB inserts the cookie into the client). The session persistence duration can be set to a value (in seconds) between 30 and 3,600. Session persistence is subject to the load balancing mode:

In the mode of **Weighted round robin** where requests are distributed based on the weight of real servers, session persistence based on cookie insertion is supported.

In the mode of **Weighted least connections** where scheduling is performed based on server load and weight, session persistence is not supported.

The mode of **IP Hash** supports session persistence based on source IP addresses, but not on cookie insertion.

Connection Timeout Period

Currently, HTTP connection timeout period (`keepalive_timeout`) is 75s by default. To adjust it, enable [custom configuration](#). If the threshold is exceeded and the session has no data transmission, the connection will end.

Currently, TCP connection timeout period is 900s by default and cannot be customized. If the threshold is exceeded and the session has no data transmission, the connection will end.

Configuring Session Persistence

1. Log in to the [CLB console](#) and click the ID of the CLB instance to be configured with session persistence to enter its details page.
2. Select the **Listener management** tab.
3. Click **Modify** next to the CLB listener to be configured with session persistence.
4. Choose whether to enable the session persistence feature. Click the button to enable it, enter the persistence duration, and click **submit**.

Relationship Between Persistent Connection and Session Persistence

For information about how to enable long connection, see [Configuring an HTTP Listener](#) and [Configuring an HTTPS Listener](#).

Scenario 1: HTTP layer-7 business

Assume a client uses HTTP/1.1 as the request protocol and includes the `Connection:keep-alive` header in requests. If the client accesses a real server via CLB without session persistence enabled, can the client access the same real server next time?

A: No.

First, HTTP keep-alive indicates TCP connection remains connected after a request is sent, so the browser can send requests via the same connection. Persistent connection reduces the time required for establishing a new connection for each request and lowers bandwidth consumption. The default timeout period of a CLB cluster is 75s (if there is no new request within 75s, TCP will be disconnected by default).

The HTTP keep-alive connection is established between the client and a CLB instance. If cookie session persistence is disabled, the CLB instance will randomly select a real server according to the round-robin policy for your access next time. The previous persistent connection is no longer valid.

Therefore, we recommend you enable session persistence.

If the cookie session persistence duration is configured as 1,000s, the client will initiate a request again. Because the interval between the two requests exceeds 75s, TCP connection needs to be established again. The application layer identifies the cookie and finds the real server the client accessed last time so it will be assessed again this time.

Scenario 2: TCP layer-4 business

Assume a client initiates access, TCP is the transport-layer protocol, persistent connection is enabled, but session persistence based on source IP address is disabled. Can the same client access the same server in the next access request?

A: Not necessarily.

First, according to layer-4 implementation mechanism, when persistent connection is enabled for TCP and not closed, and the same connection is accessed in two requests, then the same client can access the same server. If the connection is closed for some reasons (such as network restart or connection timeout) during the second access request, the request may be scheduled to another real server. The default global timeout period for a persistent connection is 900s, that is, the persistent connection will be released if there is no new request in 900s.

For information about how to enable persistent connection, see [Configuring an HTTP Listener](#) and [Configuring an HTTPS Listener](#).

Layer-7 Redirection Configuration

Last updated : 2024-10-10 16:01:16

CLB supports layer-7 redirection, so that you can configure redirection on layer-7 HTTP/HTTPS listeners.

Note:

Session persistence: If the client accesses `example.com/bbs/test/123.html` and session persistence has been enabled on the backend CVM instance, after redirection is enabled to forward traffic to

`example.com/bbs/test/456.html`, the original session persistence mechanism will not take effect.

TCP/UDP redirection: Redirection based on an IP and port is not supported currently but will be available in later versions.

Redirection Overview

Automatic redirection

Overview

For an existing `HTTPS:443` listener, an HTTP listener (port 80) will be created automatically by the system for forwarding. Requests sent to `HTTP:80` will be automatically redirected to `HTTPS:443`.

Use case

Forced HTTPS redirection, i.e., redirecting HTTP requests to HTTPS. When a user accesses a web service in a PC or mobile browser over HTTP, CLB will redirect all requests sent to `HTTP:80` to `HTTPS:443` for forwarding.

Strengths

Set-and-forget configuration: Forced HTTPS redirection can be implemented for a domain name with only one configuration operation needed.

Convenient update: If the number of URLs of the HTTPS service changes, you only need to use this feature again in the console for refreshing.

Manual redirection

Overview

You can configure 1-to-1 redirection. For example, in a CLB instance, you can configure redirection of `listener 1 / domain name 1 / URL 1` to `listener 2 / domain name 2 / URL 2`.

Note:

If the domain name has been configured with automatic redirection, you cannot configure manual redirection for it.

Use case

Single-path redirection. For example, if you want to temporarily deactivate your web business in cases such as product sellout, page maintenance, or update and upgrade, the original page needs to be redirected to a new page. If no redirection is performed, the old address in a visitor's favorites and search engine database will return a `404/503` error message page, degrading the user experience and resulting in traffic waste.

Automatic Redirection

CLB supports one-click forced redirection from HTTP to HTTPS.

Assume that you need to configure the website `https://www.example.com` , so that end users can visit it securely over HTTPS no matter whether they send HTTP requests (`http://www.example.com`) or HTTPS requests (`https://www.example.com`) in the browser.

Use limits

The redirection configuration includes the protocol/port, domain name and destination directory. Note the following limitations to avoid redirect loops.

If the source directory is the same as the destination directory, the direction configuration is not allowed.

If the source directory is already configured with a redirection policy (including the source and redirection directory), the configuration is not allowed.

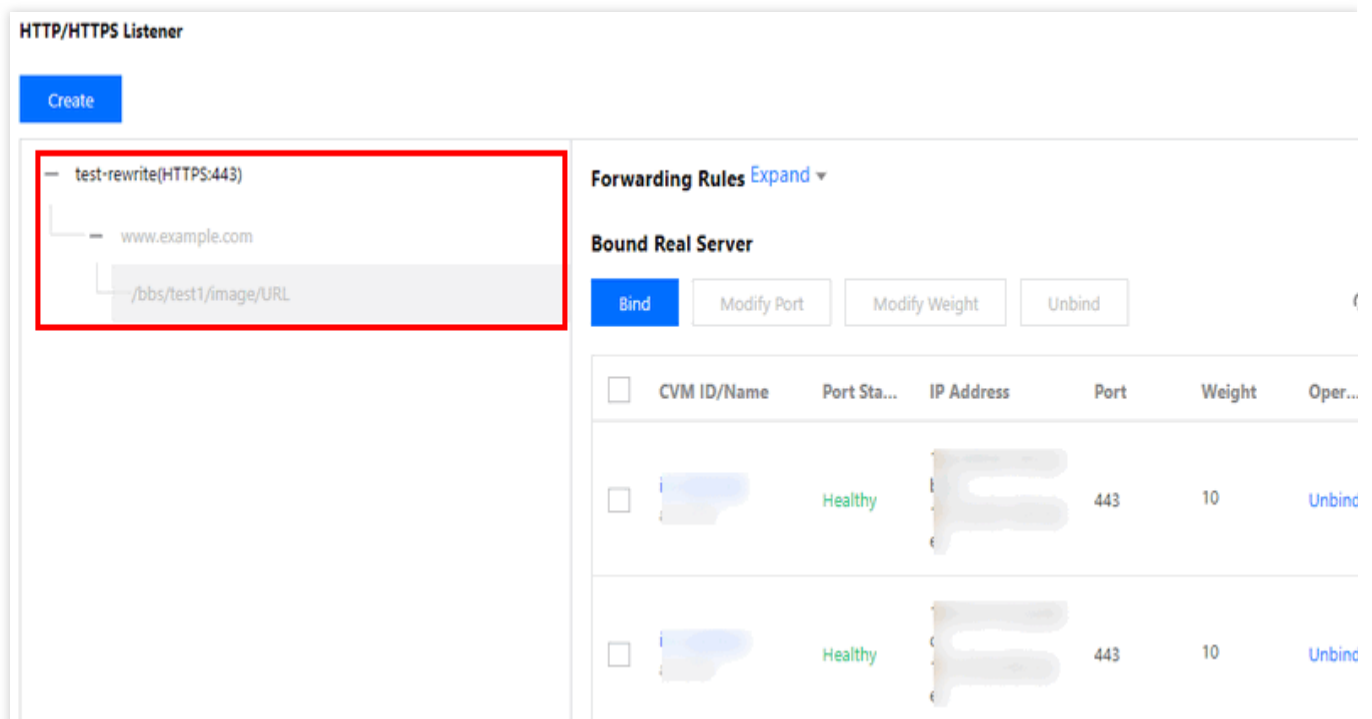
If the destination directory is used as the source directory of another redirection policy, the configuration is not allowed.

Prerequisites

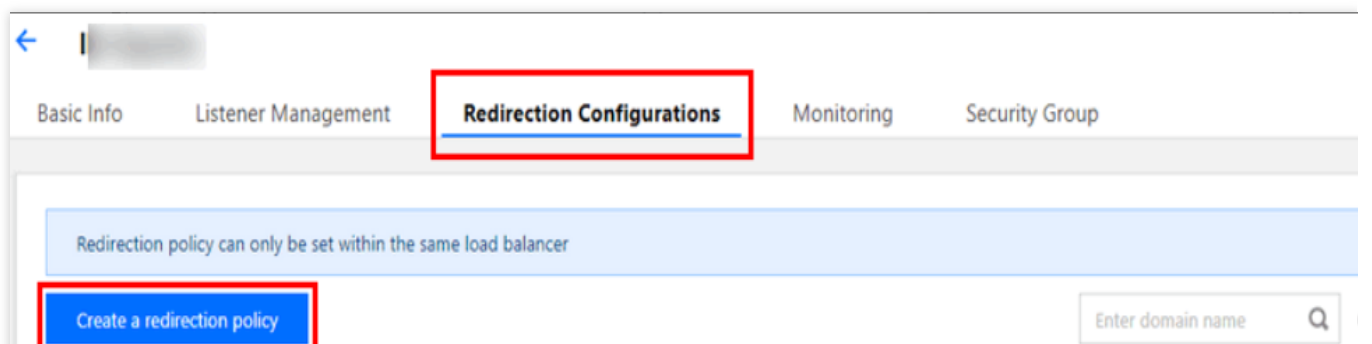
The `HTTPS:443` listener has been configured.

Directions

1. Configure the CLB HTTPS listener in the [CLB console](#) and set up the web environment of `https://example.com` . For more information, see [Configuring an HTTPS Listener](#).
2. View the result of the HTTPS listener configuration, as shown below:



3. On the **Redirection configurations** tab of the CLB instance details page, click **Create redirection policy**.



4. Select **Auto-redirection configuration**, select the configured HTTPS listener and domain name, and select the redirection status code under **Domain configuration**. Click **Submit**.

[←](#) **New redirection policy**

1 **Select domain name** >

2 **Configure Directory**

☐ Manual Redirection Configuration

If you configure the original address and redirection address manually, the system will redirect the requests from the original address to related target address. You can configure multiple directories for one domain name for redirection, so as to implement auto-redirection between HTTP/HTTPS.

☒ Auto-redirection Configuration

For the existing HTTPS:443 listener, an HTTP listener (port 80) is created by the system for forwarding. Requests sent to HTTP:80 will be redirected to HTTPS:443.

Front-end protocol and port Domain Name

[Next: Configure directory](#)

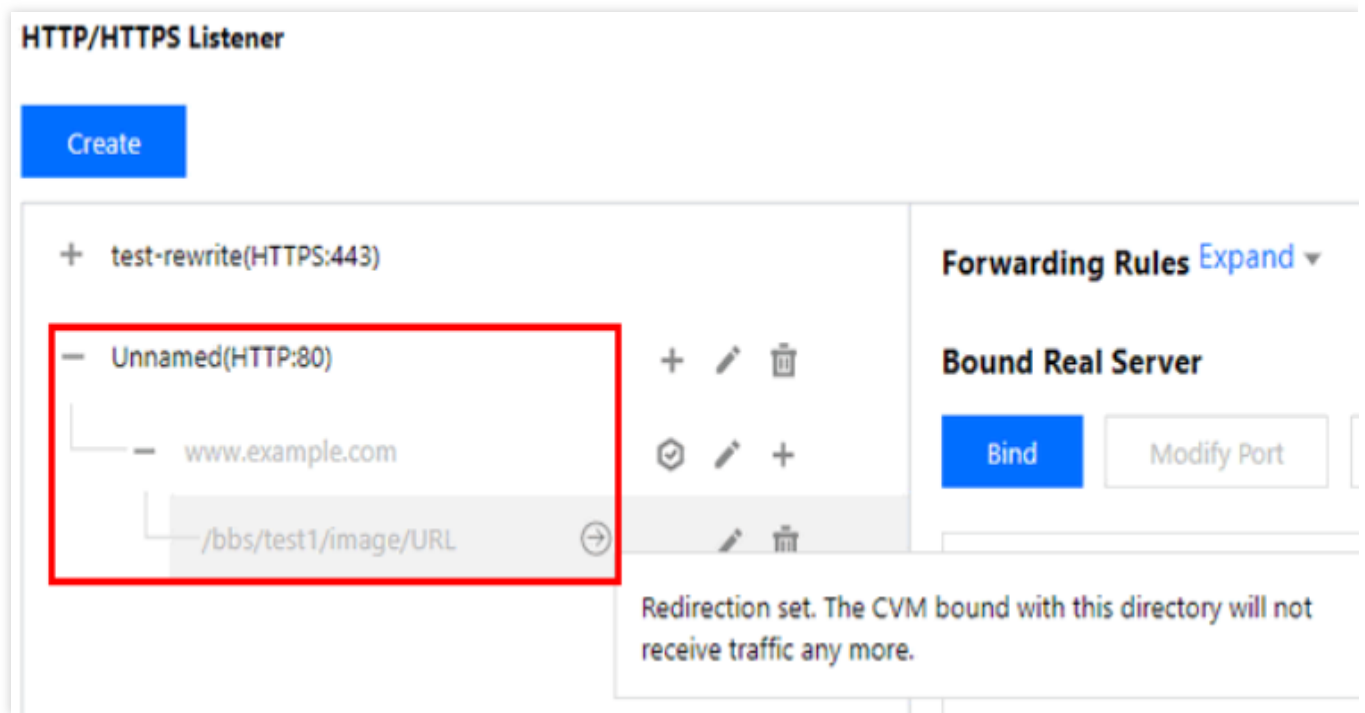
Note:

Domain configuration for redirection is in beta testing. To try it out, please [submit a ticket](#).

The status codes are as follows: 301 (Moved Permanently), 302 (Move Temporarily), and 307 (Temporary Redirect).

For more information, see [HTTP/1.1 \(RFC 7231\)](#).

5. View the result after redirection is configured, as shown below. As you can see, the `HTTP:80` listener has been automatically configured for the `HTTPS:443` listener, and all HTTP traffic will be automatically redirected to HTTPS.



Manual Redirection

CLB supports configuring 1-to-1 redirection.

For example, your business uses a `forsale` page for a promotional campaign and needs to redirect the campaign page `https://www.example.com/forsale` to the new homepage `https://www.new.com/index` after the campaign ends.

Prerequisites

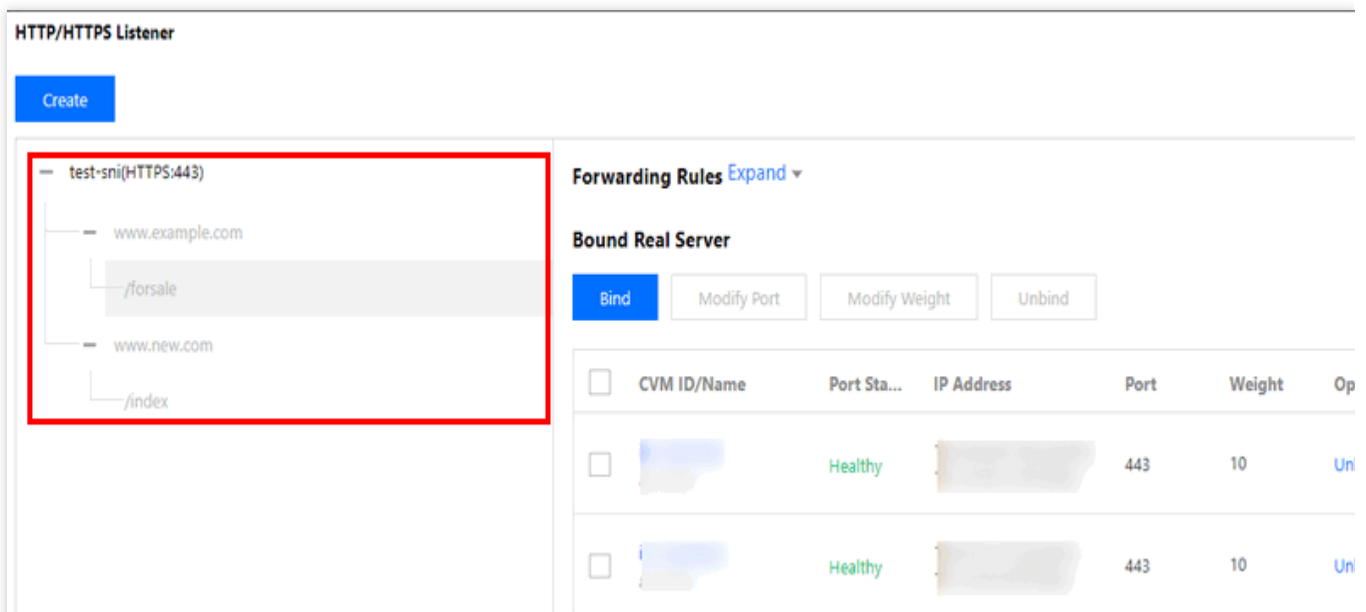
An HTTPS listener has been configured.

The forwarding domain name `https://www.example.com/forsale` has been configured.

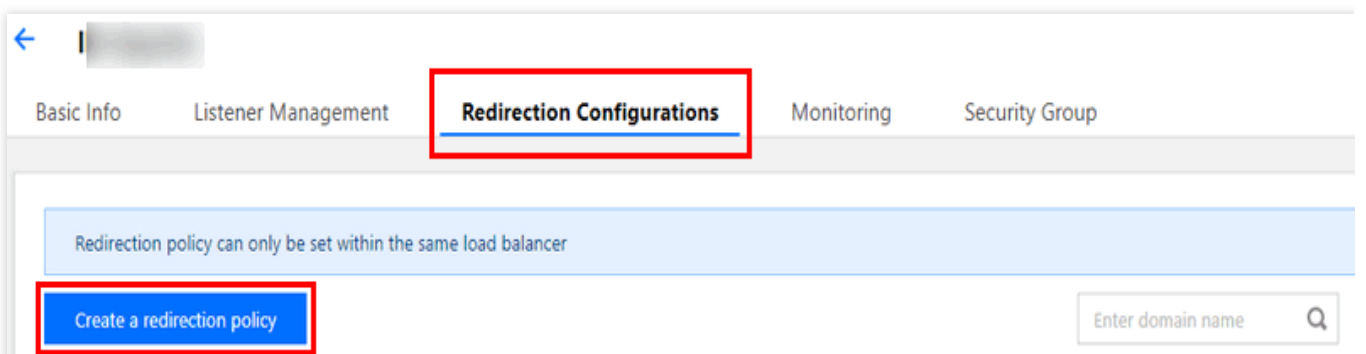
The forwarding domain name and path `https://www.new.com/index` has been configured.

Directions

1. Configure the CLB HTTPS listener in the [CLB console](#) and set up the web environment of `https://example.com`. For more information, see [Configuring HTTPS Listener](#).
2. View the result of the HTTPS configuration, as shown below:



3. On the **Redirection configurations** tab of the CLB instance details page, click **Create redirection policy**.



4. Select **Manual Redirection Configuration**, and then select the following: the originally accessed frontend protocol port, domain name, and path and the frontend protocol port, domain name, and path after redirection. Under **Domain configuration**, select the redirection status code, and select or clear the **URL Reservation** check box. Then click **Submit**.

←

New redirection policy

1

Select domain name

>

2

Configure Directory

Manual Redirection Configuration

If you configure the original address and redirection address manually, the system will redirect the requests from the original address to the related target address. You can configure multiple directories for one domain name for redirection, so as to implement auto-redirection between HTTP/HTTPS.

Original Access

Front-end protocol and port

HTTPS:443

Domain Name

www.example.com

Redirect to

Front-end protocol and port

HTTPS:443

Domain Name ⓘ

www.new.com

Auto-redirection Configuration

For the existing HTTPS:443 listener, an HTTP listener (port 80) is created by the system for forwarding. Requests sent to HTTP:80 will be redirected to HTTPS:443.

Next: Configure directory

Note:

Domain configuration for redirection is in beta testing. To try it out, please [submit a ticket](#).

The status codes are as follows: 301 (Moved Permanently), 302 (Move Temporarily), and 307 (Temporary Redirect).

For more information, see [HTTP/1.1 \(RFC 7231\)](#).

5. View the result of the redirection configuration, as shown below. As you can see, in the `HTTPS:443` listener, `https://www.example.com/forsale` is redirected to `https://www.new.com/index`.

©2013-2024 Tencent Cloud. All rights reserved.

Page 90 of 262

HTTP/HTTPS Listener

Create

test-sni(HTTPS:443)

www.example.com

/forsale

www.new.com

/index

+ ✎ 🗑

🛡 ✎ +

Forwarding Rules [Expand](#) ▼

Bound Real Server

➔ ✎ 🗑

Bind

...

Redirection set. The CVM bound with this directory will not receive traffic any more.

Layer-7 Custom Configuration

Last updated : 2025-05-19 16:56:16

CLB supports custom configurations, allowing you to set the configuration parameters for a single CLB instance, such as `client_max_body_size` and `ssl_protocols` , so as to meet your unique needs.

Note:

Each region can have up to 200 entries of custom configurations.

Custom configurations are limited to 64 KB.

Each instance can be bound to only one entry of custom configuration.

Custom configurations are valid only for layer-7 HTTP/HTTPS CLB (former Application CLB) listeners.

CLB Custom Configuration Parameters

CLB custom configuration supports the following configurations:

Configuration Field	Default Value/Recommended Value	Value Range	Description
<code>ssl_protocols</code>	Default value: TLSv1, TLSv1.1, TLSv1.2 Recommend value: TLSv1.2, TLSv1.3	TLSv1, TLSv1.1, TLSv1.2, TLSv1.3	Version of the TLS protocol used.
<code>ssl_ciphers</code>	ssl_ciphers default value	ssl_ciphers value range	Cipher suite.
<code>client_header_timeout</code>	60 seconds	30-120 seconds	Timeout period of obtaining client request headers. Status code 408 is returned in case of timeout.
<code>client_header_buffer_size</code>	4 KB	1-256 KB	Size of the default buffer where client request headers are stored.
<code>client_body_timeout</code>	60 seconds	30-120 seconds	Timeout period of obtaining a client request body, which is not the time for obtaining the entire body but refers to the idle period without data transmission.

			Status code 408 is returned in case of timeout.
client_max_body_size	60 MB	1-10240 MB	If you set this field to a value in the range of 1-256 MB, there are no other requirements. The maximum value of this field is 10240 MB (or 10 GB). If you set this field to a value greater than 256 MB, you must set proxy_request_buffering to <code>off</code> .
keepalive_timeout	75 seconds	0-900 seconds	Hold time of the client-server persistent connection. If this field is set to 0, persistent connection is prohibited. If you want to set this parameter to over 900, submit a ticket . The maximum value allowed is 3600.
add_header	Custom	-	Headers returned to the client. Set this field in the format of <code>add_header xxx yyy</code> . For example, you can set it to <code>add_header Access-Control-Allow-Methods 'POST, OPTIONS';</code> <code>add_header Access-Control-Allow-Origin *</code> ; for cross-region scenarios.
more_set_headers	Custom	-	Headers returned to the client. Set this field in the format of <code>more_set_headers "A:B"</code> .
proxy_connect_timeout	4 seconds	4-120 seconds	Timeout period of connecting to a real server.
proxy_read_timeout	60 seconds	30-3600 seconds	Timeout period of reading a real server response.
proxy_send_timeout	60 seconds	30-3600 seconds	Timeout period of sending a request to a real server.

server_tokens	on	on, off	<p><code>on</code> : displays version information.</p> <p><code>off</code> : hides version information.</p>
keepalive_requests	100	1-10000	Maximum number of requests that can be sent over the client-server persistent connection.
proxy_buffer_size	4 KB	1-32 KB	Size of server response headers, which is the size of a single buffer set in <code>proxy_buffer</code> by default. To use <code>proxy_buffer_size</code> , <code>proxy_buffers</code> must be set at the same time.
proxy_buffers	Quantity: 8; size: 4 KB	Quantity: 3-8; size: 4-16 KB	Buffer quantity and size.
proxy_request_buffering	off	on, off	<p><code>on</code> : caches the client request body. The CLB instance caches the request and forwards it to the backend CVM instance in multiple parts after the request is completely received.</p> <p><code>off</code> : does not cache the client request body. After receiving a request, the CLB instance directly forwards it to the backend CVM instance, which increases pressure on the performance of the backend CVM instance.</p>
proxy_set_header	X-Real-Port \$remote_port	X-Real-Port \$remote_port X-clb-lbid \$lbid Stgw-request-id \$stgw_request_id X-Forwarded-Port \$vport X-Method \$request_method X-Uri \$uri	<p><code>X-Real-Port \$remote_port</code> : client port.</p> <p><code>X-clb-lbid \$lbid</code> : CLB LBID, which is the identifier of a CLB instance.</p> <p><code>Stgw-request-id \$stgw_request_id</code> : request ID (used in CLB only).</p> <p><code>X-Forwarded-Port</code> : CLB listener port.</p>

			<code>X-Method</code> : client request method. <code>X-Uri</code> : client request URI.
send_timeout	60 seconds	1-3600 seconds	Timeout period of data transfer from the server to the client, which is the time interval between two consecutive data transfer actions, not the entire request transfer period.
ssl_verify_depth	1	[1,10]	Verification depth of the client certificate chain.
proxy_redirect	http:// https://	http:// https://	If the real server returns a redirect or refresh request (status code 301 or 302), <code>proxy_redirect</code> will reset <code>http</code> to <code>https</code> in the HTTP header <code>Location</code> or <code>Refresh</code> for safe redirection.
ssl_early_data	off	on, off	Enables or disables TLS 1.3 0-RTT. Only when the value of <code>ssl_protocols</code> contains <code>TLSv1.3</code> , <code>ssl_early_data</code> can take effect. You shall consider the risk of replay attacks before enabling <code>ssl_early_data</code> .
http2_max_field_size	4 KB	1-256 KB	Maximum size of request headers after HPACK compression.
proxy_intercept_errors	off	on, off	When configuring <code>error_page</code> , <code>proxy_intercept_errors</code> must be set to on in advance.
error_page	-	error_page code [= [response]] uri	A predefined URI is shown for the specific error code. The default response code is 302. The URI must start with <code>/</code> .
proxy_ignore_client_abort	off	on, off	Whether to disconnect the CLB

			instance from the real server when the client terminates its connection with the CLB instance without waiting for a response.
l7_toa	off	on, off	Switch of TOA After TOA is enabled, the client source IP and port in the TOA are added to \$remote_addr and \$remote_port separately. In this case, the IP information of TOA is passed through to X-Forwarded-For and X-Real-IP. Note: This parameter is only available for IPv4 CLB instances.
l7_toa_proxy_transparent	off	on, off	When it is off, when a new connection is set up between a CLB and real server, the 4-tuple source IP address received is encapsulated as the client source IP and sent to the real server. When it is on, the client source IP in TOA is encapsulated as the client source IP and sent to the real server. If long connection is enabled, IPs within the 100.127.0.0/16 range are used. Note: This parameter is only available for IPv4 CLB instances.

Note:

Requirement on the value of `proxy_buffer_size` and `proxy_buffers` : $2 * \max(\text{proxy_buffer_size}, \text{proxy_buffers.size}) \leq (\text{proxy_buffers.num} - 1) * \text{proxy_buffers.size}$; For example, if `proxy_buffer_size` is 24 KB and `proxy_buffers` is 8 8 KB , then $2 * 24 \text{ KB} = 48 \text{ KB} \leq (8 - 1) * 8 \text{ KB} = 56 \text{ KB}$, meeting the requirement. Therefore, there will be no configuration error.

ssl_ciphers Configuration Instructions

The `ssl_ciphers` encryption suite being configured must be in the same format as that used by OpenSSL. The algorithm list is one or more `<cipher strings>`; multiple algorithms should be separated with ":"; "!" indicates not to enable an algorithm, and "+" indicates to move an algorithm to the last place.

The encryption algorithm for default forced disabling is:

```
!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!DHE .
```

Default value:

```
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA3
```

Value range:

```
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA3
```

CLB Custom Configuration Examples

1. Log in to the [CLB console](#) and click **Custom Configuration** in the left sidebar.
2. Select a region at the top of the **Custom Configuration** page, and click **Create**.
3. On the **Create custom configuration** page, enter the configuration name and code configuration items, each item ending with a semicolon (;). After filling in all the information, click **Completed**.

4. Return to the **Custom Configuration** page. Click **Bind to Instance** on the right.
5. In the pop-up window, select a CLB instance to bind, and click **Submit**.

6. On the **Custom Configuration** page, click the configured ID to go to its details page. You can check the bound instance on the **Bind Instance** tab.

7. (Optional) You can now view the corresponding custom configuration information on the instance list page.

Note:

If **Bind Custom Configurations** is not displayed on the instance list, click

in the top-right corner. In the pop-up **Customize List Field** dialog box, select **Bind Custom Configurations**, and click **OK**. You should see the column displayed.

Check below for the sample codes of default configuration. When you try to copy the codes, make sure there is no blank line at the end.

```
ssl_protocols      TLSv1 TLSv1.1 TLSv1.2;
```

```
client_header_timeout    60s;  
client_header_buffer_size 4k;  
client_body_timeout      60s;  
client_max_body_size     60M;  
keepalive_timeout        75s;  
add_header               xxx yyy;  
more_set_headers          "A:B";  
proxy_connect_timeout     4s;  
proxy_read_timeout        60s;  
proxy_send_timeout        60s;
```

Layer-7 Domain Name Forwarding and URL Rules

Last updated : 2025-03-26 15:11:46

Process Flows

The process flows of layer-7 and layer-4 CLB (formerly application CLB) are shown below:

Using Layer-7 CLB to forward an HTTP/HTTPS protocol, you can add a corresponding domain name when creating a forwarding rule in a CLB instance listener.

If only one forwarding rule is created, you can access the corresponding forwarding rule and the service through VIP+URL.

If multiple forwarding rules are created, the use of VIP+URL does not guarantee access to a specified domain name+URL. You should access a domain name+URL directly to make sure a forwarding rule has taken effect. In other words, when you configure multiple forwarding rules, a VIP may correspond to multiple domain names. In this case, we recommend you access the service via specified domain name+URL instead of VIP+URL.

Layer-7 Forwarding Configurations

Domain forwarding configurations

Layer-7 CLB can forward requests from different domain names and URLs to different servers. A layer-7 listener can be configured with multiple domain names, each of which can be configured with multiple forwarding paths.

Length limit for the forwarded domain name: 1 to 80 characters.

It cannot begin with `_`.

An exact domain is supported, such as `www.example.com`.

Wildcard domain names are supported, but currently only those in the form of `*.example.com` or `www.example.*`, that is, wildcard domain names begin or end with `*` which appears only once.

For non-regex forwarded domain names, valid character sets include `a-z`, `0-9`, `.`, `-` and `_`.

The forwarded domain name supports regex. Regex domain names:

Supported character sets include `a-z`, `0-9`, `.`, `-`, `?`, `=`, `~`, `_`, `-`, `+`, `\\`, `<code>\\\\`, `</code>`, `^`, `*`, `!`, `$`, `&`, `|`, `(`, `)`, `[`, and `]`.

Must begin with `~` which can appear only once.

An example of regex domain name supported by CLB may be `~^www\\d+\\.example\\.com$`.

Forwarded domain name matching

General matching policies

1. If you enter an IP address instead of a domain name in the forwarding rule and configure multiple URLs in the forwarding group, VIP+URL will be used to access the service.
2. If you configure a full domain name in the forwarding rule and multiple URLs in the forwarding group, domain name+URL will be used to access the service.
3. If you configure a wildcard domain name in the forwarding rule and multiple URLs in the forwarding group, you will access the service through the matching of requested domain name and URLs. To have different domain names point to the same URL, you can use this method for configuration. Taking `example.qcloud.com` as an example, the format is as follows:

Exact match: matches the domain name which is completely matched with the entered domain.

Prefix wildcard: matches all domain names with the specified second- and top-level domain, such as

```
*.qcloud.com .
```

Suffix wildcard: matches all domain names with the specified third- and second-level domain, such as

```
example.qcloud.* .
```

Regex match: `~^www\\d+\\.example\\.com$`

Matching priority: Exact path > prefix path (non-regular expression) > regular expression path (~). There is no priority difference between regular expressions. If a domain name hits multiple regular expression rules, the specific order for rules to take effect depends on the underlying configuration order. If the customer demands for exact forwarding, it is recommended to distinguish domain names by exact path matching and prefix path matching.

4. If you configure a domain name in the forwarding rule and a URL for fuzzy matching in the forwarding group, you can initiate full matching by using prefix matches and adding a suffixed wildcard `$`.

For example, if you enter `URL ~*. (gif|jpg|bmp) $`, it will match all .gif, .jpg and .bmp files.

Default domain name policy

When the requested domain name does not match any rule, CLB will forward the request to the default domain name (Default Server). One listener can have only one default domain name.

For example, the `HTTP:80` listener of CLB instance 1 is configured with two domain names: `www.test1.com`

and `www.test2.com`, where `www.test1.com` is the default domain name. When a user visits

`www.example.com`, since no domain name is matched, CLB will forward the request to the default domain name

`www.test1.com`.

Note:

Before May 18, 2020, the default domain name is optional for layer-7 listeners.

If your layer-7 listener has a default domain name configured, client requests that do not match other rules will be forwarded to it.

If your layer-7 listener has no default domain name configured, client requests that do not match other rules will be forwarded to the first domain name loaded by CLB (its loading order may be different from that configured in the console; therefore, it may not be the first one configured in the console).

Starting from May 18, 2020:

All new layer-7 listeners must have a default domain name: the first rule of a layer-7 listener will be set as the default domain name. When you create a layer-7 rule via API, the `DefaultServer` field is set to `true`.

For all listeners that have a default domain name configured, you need to specify a new default domain name when modifying or deleting the existing default domain name: when you perform the operation in the console, you need to specify a new default domain name; when you perform the operation by calling an API, if you do not set a new default domain name, CLB will set the earliest-created one among the remaining domain names as the new default domain name.

For existing rules without a default domain name, you can directly configure a default domain name based on your business needs as instructed in [operation 4](#) below. If you don't do so, Tencent Cloud will set the first domain name loaded by CLB as the default domain name. Existing listeners will be all processed before June 19, 2020.

The above policy will be implemented gradually starting from May 18, 2020, and the effective date for each instance may vary slightly. As of June 20, 2020, all layer-7 listeners that have a forwarded domain name will have a default domain name.

The following four operations can be performed on the default domain name:

Operation 1: when configuring the first forwarding rule for the layer-7 listener, the default domain name must be in "enabled" status.

Operation 2: disable the current default domain name.

If there are multiple domain names under a listener, when disabling the current default domain name, you need to specify a new default domain name.

If a listener has only one domain name and the domain name is the default domain name, it cannot be disabled.

Operation 3: delete the default domain name.

If there are multiple domain names under a listener, when you delete a rule under the default domain name:

If the rule is not the last rule of the default domain name, you can delete it directly.

If the rule is the last rule of the default domain name, you need to set a new default domain name.

If there is only one domain name under a listener, you can directly delete all rules without setting a new default domain name.

Operation 4

: you can quickly modify the default domain name in the listener list.

Forwarded URL path configuration rules

Layer-7 CLB can forward requests from different URLs to different servers for processing, and multiple forwarded URL paths can be configured for a single domain name.

Length limit of forwarded URL: 1–200 characters.

A non-regex forwarded URL is case-sensitive and must start with `/`, with valid character sets including `a-z`, `A-Z`, `0-9`, `.`, `-`, `_`, `/`, `=`, `?`, and `:`.

Forwarded URL supports regex:

A regex URL must begin with `~` which can appear only once.

For a regex URL, the valid character sets include `a-z`, `A-Z`, `0-9`, `.`, `-`, `_`, `/`, `=`, `?`, `~`, `^`, `*`, `$`, `:`, `(`, `)`, `[`, `]`, `+`, and `|`.

An example of regex URL may be `~*.png$`.

The matching rules for a forwarded URL are as follows:

Beginning with `=` indicates exact match.

Beginning with `^~` indicates that the URL starts with a regular string and is not for regex match.

Beginning with `~` indicates case-sensitive regex match.

Beginning with `~*` indicates case-insensitive regex match.

`/` indicates generic match, where any requests will be matched if there are no other matches.

Forwarded URL path matching description

1. Matching rules: based on longest prefix match, exact match is performed first followed by fuzzy match.

For example, after you configure the forwarding rules and forwarding groups as shown above, the following requests will be matched into different forwarding rules in sequence.

1.1 Because `example.qcloud.com/test1/image/index1.html` exactly matches the URL rule configured by forwarding group 1, the request will be forwarded to the real server associated with forwarding group 1, i.e., port 80 of CVM1 and CVM2 in the figure.

1.2 Because `example.qcloud.com/test1/image/hello.html` has no exact match, it will match forwarding rule 2 based on longest prefix match; therefore, the request will be forwarded to the real servers associated with forwarding rule 2, i.e., port 81 of CVM2 and CVM3 in the figure.

1.3 Because `example.qcloud.com/test2/video/mp4/` has no exact match, it will match forwarding rule 3 based on longest prefix match; therefore, the request will be forwarded to the real server associated with forwarding rule 3, i.e., port 90 of CVM4 in the figure.

1.4 Because `example.qcloud.com/test3/hello/index.html` has no exact match, it will match the root directory's default URL `example.qcloud.com/` by longest prefix match. In this case, Nginx will forward the request to the real server such as FastCGI (php) or Tomcat (jsp), while Nginx will exist as a reverse proxy server.

1.5 Because `example.qcloud.com/test2/` has no exact match, it will match the root directory's default URL `example.qcloud.com/` by longest prefix match.

2. If the service does not work properly in the set URL rules, it will not be redirected to other pages after successful match.

For example, the client requests `example.qcloud.com/test1/image/index1.html` and matches it with forwarding group 1. However, the real server of forwarding group 1 has an exception and a 404 error page appears. You will see the 404 error page, but not being redirected to other pages.

3. You are recommended to point the default URL to a stable page (such as a static page or homepage) and bind it to all real servers. If none of the rules match, the system will point the request to the default URL page; otherwise, a 404 error may occur.

4. If you do not set the default URL, and none of the forwarding rules match, a 404 error will be returned when you access the service.

5. Note on the slash at the end of the layer-7 URL path: if the URL you set ends with `/`, but the access request from the client does not contain `/`, then the request will be redirected to a rule ending with `/` (301 redirect).

For example, under the `HTTP:80` listener, the configured domain name is `www.test.com`:

5.1 If the URL set under this domain name is `/abc/`:

When the client accesses `www.test.com/abc`, it will be redirected to `www.test.com/abc/`.

When the client accesses `www.test.com/abc/`, it will match `www.test.com/abc/`.

5.2 If the URL set under this domain name is `/abc`:

When the client accesses `www.test.com/abc`, it will match `www.test.com/abc`.

When the client accesses `www.test.com/abc/`, it will also match `www.test.com/abc`.

Layer-7 Health Check Configuration Description

Health check domain name configuration rules

A health check domain name is the domain name used by layer-7 CLB to detect the health status of a real server.

Length limit: 1-80 characters.

Default: forwarded domain name.

Regex is not supported. If your forwarded domain name is a wildcard domain name, you should specify a fixed one (non-regex).

Valid character sets include `a-z`, `0-9`, `.`, `-`, and `_`. For example, `www.example.qcloud.com`.

Health check path configuration rules

A health check path is the URL path used by layer-7 CLB to detect the health status of a real server.

Length limit: 1-200 characters.

Default: `/`. You can enter a custom path starting with `/`.

Regex is not supported. You are recommended to specify a fixed URL (static page) for health check.

Valid character sets include `a-z` , `A-Z` , `0-9` , `.` , `-` , `_` , `/` , `=` , `?` , and `:` . For example, `/index` .

Using QUIC Protocol on CLB

Last updated : 2024-10-10 16:41:14

The [Quick UDP Internet Connection \(QUIC\)](#) protocol helps you access applications faster and achieve multiplexing with no reconnection required in scenarios such as weak network or frequent switch between Wi-Fi and 4G. This document describes how to configure the QUIC protocol in the CLB console.

QUIC Overview

QUIC is a transport layer network protocol designed by Google, multiplexing concurrent data streams using UDP.

Compared with the popular TCP+TLS+HTTP2 protocol, QUIC has the following advantages:

Establish a connection faster.

Improve congestion control.

Adopt multiplexing to avoid head-of-line (HOL) blocking.

Support connection migration.

After QUIC is enabled, the client can establish a QUIC connection with a CLB instance. If the QUIC connection fails due to negotiation between the client and the CLB instance, HTTPS or HTTP/2 will be used. Upon enabling QUIC, the backend protocol can solely utilize HTTP1.x protocol.

Use Limits

Only CLB instances, excluding classic CLB instances, support the QUIC protocol.

Only IPv4 and IPv6 NAT64 CLB instances support the QUIC protocol.

Only layer-7 HTTPS listeners support the QUIC protocol.

Currently, CLB supports the following QUIC versions: Q050, Q046, Q043, h3-29, and h3-27.

Directions

1. Create a CLB instance as needed. For more information, see [Creating CLB Instances](https://www.tencentcloud.com/document/product/214/614918d55ae663895d87ef8c66952b1635f3c)(<https://www.tencentcloud.com/document/product/214/614918d55ae663895d87ef8c66952b1635f3c>).
2. Log in to the [CLB console](#) and click **Instance management** in the left sidebar.
3. On the page that appears, click the **Cloud Load Balancer** tab.
4. Create a Load Balancer instance and in the right-hand operation column, click **Configure Listener**.

ID/Name	Mon...	Status	Domain name	VIP	Availability z...	Network t...	Network	Instance ...	Health status	Billing m...	Tags	Operation
31		Normal	-	4	Shanghai Zone 2	Public network		Shared	Health check not enabled (Configuration)	Pay-as-you-go - Traffic-based Created at 2022-11-17 19:50	-	Configure listener More

5. On the **Listener management** page, click **Create** under **HTTP/HTTPS Listener**.

Basic information

Listener management

Redirection configurations

Monitoring

Security groups

We support one-click activation of free WAF service to protect your websites and apps.[See details](#)

Note: When custom redirection policies are configured, the original forwarding rules are modified, the redirection policies will be removed automatically. You can click [View details](#) to learn more.

HTTP/HTTPS listener(Configured2)

Create

+ test(HTTPS:443)

+ test(HTTP:80)

Click the left node to view details

TCP/UDP/TCP SSL/QUIC listener(Configured0)

Create

You've not created any listeners. [Create now](#)

Click the left node to view details

6. On the page that appears, select “HTTPS” as the protocol of the listening protocol port. Complete other configurations, and click **Submit**.

CreateListener

Name

test-quic

Listen Protocol Ports

HTTPS

:

443

Enable SNI

SSL phrasing

One-way authentication(Recommended)

[View comparison](#)

Note: Choose SSL two-way authentication if you also need a certificate from the client.

Server certificate

☒ Select existing

☐ Create

Please select

Add certificate

Delete

1. If HTTPS is used for listening, the access from client to CLB is encrypted with this protocol. For forwarding requests from CLB to backend CVM, HTTP and HTTPS are available when you create forwarding rules.

2. The load balancer serves as an agent for the overhead of SSL encryption and decryption, and ensures Web access security.

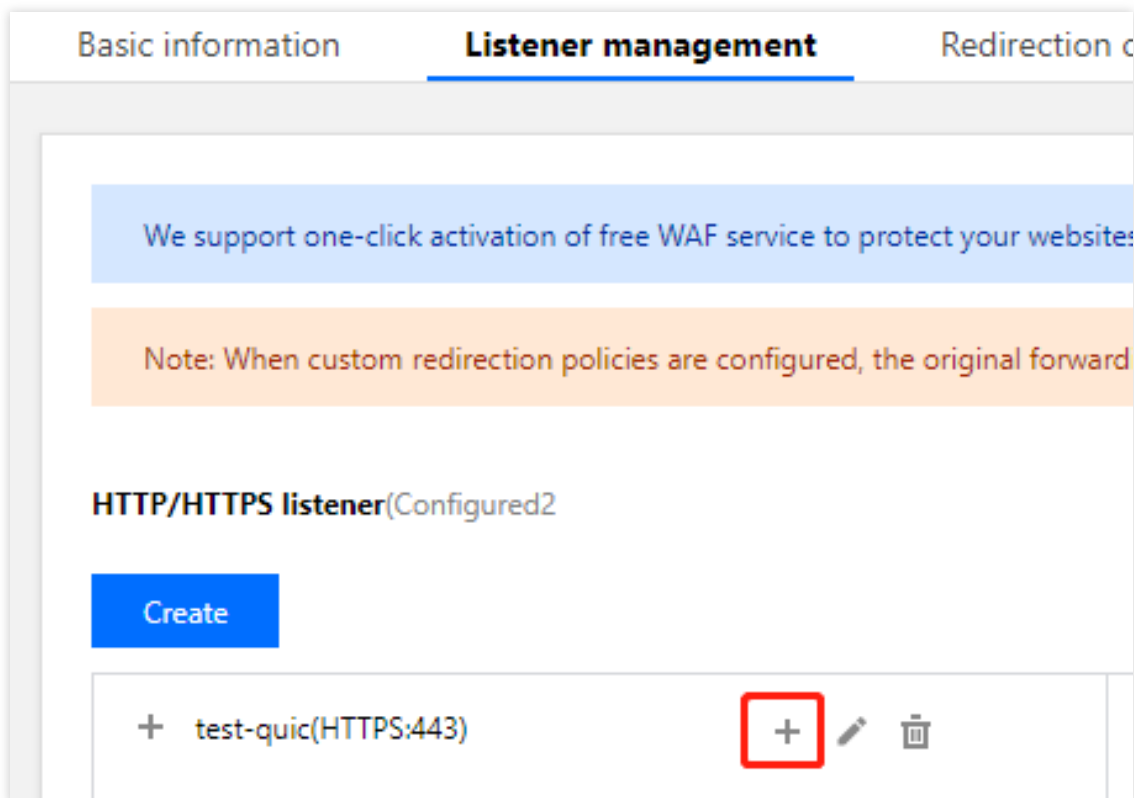
3. You can go to [SSL Certificate Management Platform](#) to apply for an SSL certificate for free.

4. To enable SNI, you do not need to configure the certificate here. Please configure it on the domain configuration page.

Close

Submit

7. On the **Listener management** tab, click + on the right of the created listener.



8. On the **Create forwarding rules** page, enable **QUIC** and create a layer-7 rule. Fill in relevant fields and click **Next** to complete the basic configuration.

Note:

After you create an HTTPS forwarding rule, you can enable or disable the QUIC protocol as needed under the domain name of the rule.

Based on the UDP protocol, QUIC will use the UDP port of a CLB instance. If you enable QUIC for a HTTPS listener, UDP and TCP ports will be used. For example, if you enable QUIC for the HTTPS:443 listener, both TCP:443 and UDP:443 ports are used, and you cannot create the TCP:443 or UDP:443 listener.

Create Forwarding rule ✕

1 Basic configuration

2 Health check

3 Session persistence

Domain name ⓘ

Default domain name

Enable

If a client request does not match any domain names of this listener, the CLB instance will forward the request to the default domain name (Default Server). Each listener only can configure one listener and must configure one. [Details](#)

HTTP2.0

☒

QUIC

☒

URL ⓘ

Balancing method ⓘ

Weighted round robin ▼

WRR scheduling is based on the number of new connections, where real servers with higher weights have more polls

Backend protocol ⓘ

HTTP ▼

Get client IP

Enabled

Gzip compression

Enabled ⓘ

Target group ⓘ

☐

Close

Next

Related Operations

After the basic configuration is completed, you can configure [health check](#) and [session persistence](#).

SNI Support for Binding Multiple Certificates to a CLB Instance

Last updated : 2024-10-10 16:46:13

Server Name Indication (SNI) is designed to solve the problem that one server can only use one certificate so as to improve SSL/TLS extensions of the server and the client. If a server supports SNI, it means that the server can be bound to multiple certificates. To use SNI for the client, the domain name to connect to should be specified before SSL/TLS connections to the server are established, and then the server will return an appropriate certificate based on the domain name.

Use Cases

A layer-7 HTTPS CLB listener supports SNI, i.e., binding multiple certificates, which can be used by different domain names in the listening rules. For example, in the same `HTTPS:443` listener of a CLB instance, you can use certificate 1 and certificate 2 for `*.test.com` and `*.example.com` respectively to forward requests from these domain names to two different sets of servers.

Prerequisites

You have [purchased a CLB instance](#).

Note:

Classic CLB does not support forwarding based on domain name and URL; therefore, it does not support SNI.

Directions

1. Log in to the [CLB console](#).
2. [Configure an HTTPS listener](#) and enable SNI.

CreateListener

Name

test-sni

Listen Protocol Ports

HTTPS

:

443

Enable SNI ⓘ

☒

1. If you select HTTPS protocol for forwarding, the accesses from client to load balancer is encrypted with HTTPS protocol. HTTP protocol is adopted to forward requests from load balancers to backend CVM.

2. The load balancer serves as an agent for the overhead of SSL encryption and decryption, and ensures Web access security.

3. You can go to [SSL Certificate Management Platform](#) to apply for an SSL certificate for free.

4. To enable SNI, you do not need to configure the certificate here. Please configure it on the domain configuration page.

Close

Submit

3. When adding a forwarding rule to the listener, configure different server certificates for different domain names. Then, click **Next** and configure health check and session persistence.

©2013-2024 Tencent Cloud. All rights reserved.

Page 111 of 262

Create Forwarding rules

1 Basic Configuration

2 Health Check

3 Session Persistence

Domain Name ⓘ

*.example.com

Default Domain Name

☒

If the client request does not match any domain name of this listener, CLB will forward the request to the default domain name. Each listener can only be configured with one default domain name, [Details](#)

HTTP2.0

☒

URL ⓘ

/

Balance Method

Weighted Round Robin

If you set a same weighted value for all CVMs, requests will be distributed by a simple pooling policy.

Backend Protocol ⓘ

HTTP

SSL Phrasing

One-way Authentication(Recommended)

[Detailed Comparison](#)

Note: Choose SSL two-way authentication if you also need a certificate from the client.

Server Certificate

☒ Select existing ☐ Create

Please select

Get client IP

Enabled

Gzip compression

Enabled ⓘ

Close

Next

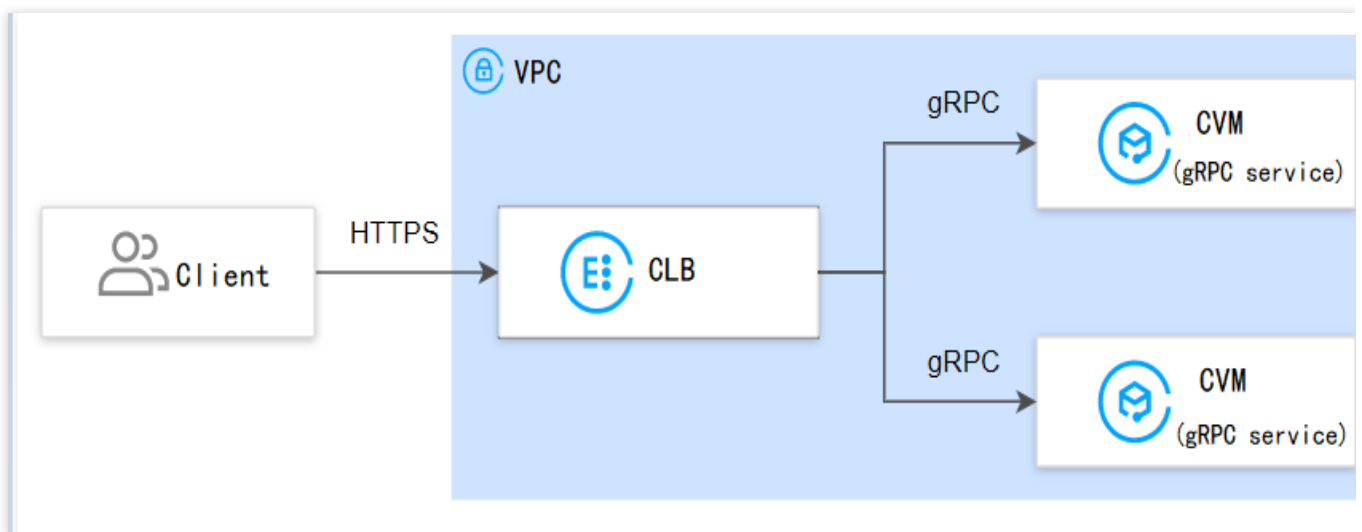
Configuring gRPC Support for Layer-7 Protocols

Last updated : 2024-10-10 16:49:41

[gRPC](#) is a high-performance, open-source software framework developed by Google based on the HTTP 2.0 transport layer protocol. The framework provides methods for configuring and managing network devices in multiple programming languages. This document describes how to configure gRPC health check for the HTTPS listener of a CLB instance to forward client gRPC requests to real servers that use the gRPC protocol.

Use Cases

When a client sends HTTPS requests to access real servers that use the gRPC protocol, you can configure gRPC health check for the HTTPS listener of the CLB instance to implement the access.



Prerequisites

You have created a VPC. For more information, see [Creating VPC](#).

You have created a CVM instance (used as a real server) in the VPC, and deployed a gRPC service on the instance.

For more information, see [Creating Instances via Images](#).

You have purchased a CLB instance. For more information, see [Creating CLB Instances](#).

Use Limits

This feature is supported only by CLB but not classic CLB.

This feature is not supported by CLB for IPv6 and CLB for IPv6 with layer-7 mixed binding enabled.

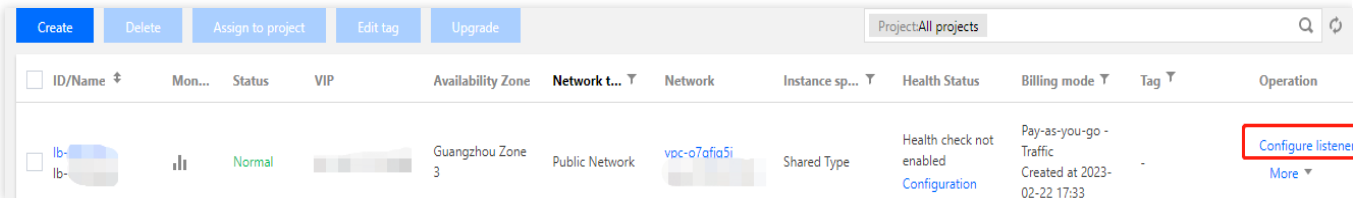
This feature is only supported by VPC but not classic networks.

Real servers do not support SCF. (Support for the gRPC protocol within the SCF target is required.)

Directions

Step 1. Configure a listener

1. Log in to the [CLB console](#) and click **Instance management** in the left sidebar.
2. Select your region in the top-left corner of the **Instance management** page and click **Configure listener** in the **Operation** column of your CLB instance.



3. Under **HTTP/HTTPS listener**, click **Create** and configure the HTTPS listener in the pop-up window.

3.1 Create a listener

Parameter	Description	Example
Name	Listener name.	test-https-443
Listening protocol and port	Listening protocol: HTTPS is used in this example. Listening port: The port used to receive requests and forward them to a real server. Port range: 1-65535. The listening port must be unique in the same CLB instance.	HTTPS:443
Enable persistent connection	Once this feature is enabled, persistent connections will be used between a CLB instance and real servers, and the CLB instance will no longer pass through the source IP address that can be obtained from XFF. To ensure normal forwarding, enable the "Allow Traffic by Default" feature in the CLB security group or allow <code>100.127.0.0/16</code> in the CVM security group. Note: Once this feature is enabled, the number of the connections between a CLB instance and real servers will fluctuate in the range of [QPS,QPS*60], subject to the connection reuse rate. If there is a limit on the maximum number of connections, we recommend you be cautious when enabling this feature. This feature is currently in beta test. To try it out, submit a ticket . The IP range 100.64.0.0/10 is already allowed as the health check source IP. You don't need to allow IPs within this range again.	Disabled

Enable SNI	If SNI is enabled, multiple domain names of a listener can be configured with different certificates; if it is disabled, multiple domain names of a listener can be configured with one certificate only.	Disabled
SSL parsing	One-way authentication and mutual authentication are supported. CLB takes over the overheads of SSL encryption and decryption to guarantee the access security.	One-way authentication
Server certificate	You can select an existing certificate in the SSL Certificate Service console or upload a certificate.	Select an existing certificate.

3.2 Create a forwarding rule

Parameter	Description	Example
Domain name	Forwarding domain name: Length: 1 to 80 characters. A domain name cannot start with underscores (_). Exact and wildcard domain names are supported. Regular expressions are supported. For detailed configuration rules, see Layer-7 Domain Name Forwarding and URL Rules .	www.example.com
Default Domain	If all domain names of a listener are not matched, the system distributes requests to the default domain name, making default access controllable. Each listener can be configured with one default domain name only.	Enabled
HTTP 2.0	After HTTP 2.0 is enabled, CLB instances can receive HTTP 2.0 requests. CLB instances access real servers over HTTP 1.1 no matter what HTTP version the client uses to access CLB instances.	Enabled
QUIC	After QUIC is enabled, a client can establish a QUIC connection with a CLB instance. If the QUIC connection fails due to negotiation between the client and the CLB instance, HTTPS or HTTP/2 will be used. However, the CLB instance and the real server still use the HTTP 1.x protocol. For more information, see Using QUIC Protocol on CLB .	Enabled
URL	Forwarding URL: Length: 1 to 200 characters. Regular expressions are supported. For detailed configuration rules, see Layer-7 Domain Name Forwarding and URL Rules .	/index
Balancing method	For HTTPS listeners, CLB supports three scheduling algorithms: weighted round robin (WRR), weighted least connections (WLC), and	WRR

	<p>IP Hash.</p> <p>WRR: Requests are distributed to real servers in sequence based on their weights. This algorithm performs scheduling based on the number of new connections. Servers with higher weights are more likely to be scheduled and servers with the same weight process the same number of connections.</p> <p>WLC: Loads of servers are estimated based on the number of active connections to the servers. This algorithm performs scheduling based on server loads and weights. For servers with the same weight, those have less loads are more likely to be scheduled.</p> <p>IP Hash: This algorithm uses a request source IP address as the Hash key to locate the corresponding server in the static hash table. If a server is available and not overloaded, requests will be distributed to it; otherwise, a null value will be returned.</p>	
Backend Protocol	<p>Backend protocol is used between a CLB instance and a real server:</p> <p>If HTTP is selected as the backend protocol, the HTTP service must be deployed on the real server.</p> <p>If HTTPS is selected as the backend protocol, the HTTPS service must be deployed on the real server. In this case, encryption and decryption of the HTTPS service will consume more resources on the real server.</p> <p>If gRPC is selected as the backend protocol, the gRPC service must be deployed on the real server. You can select gRPC as the backend forwarding protocol only when HTTP2.0 is enabled and QUIC is disabled.</p>	gRPC
Get client IP	Enabled by default.	Enabled
Gzip compression	Enabled by default.	Enabled

3.3 Configure HTTPS health check (see [HTTPS Health Check Overview](#))

3.4 Configure session persistence

Parameter	Description	Example
Session persistence	<p>After session persistence is enabled, a CLB listener will distribute access requests from the same client to the same real server.</p> <p>TCP session persistence is implemented based on client IP address. The access requests from the same IP address are forwarded to the same real server.</p> <p>Session persistence can be enabled for WRR scheduling but not WLC scheduling.</p>	Enabled
Hold Time	<p>Session persistence is terminated if there are no new requests in the connection within the specified duration.</p> <p>Value range: 30-3600 seconds</p>	30 seconds

Step 2. Bind a real server

1. On the **Listener management** page, select the created listener `HTTPS : 443` . Click **+** on the left to expand the domain names and URL paths, select the desired URL path, and view the real servers bound to the path on the right of the listener.
2. Click **Bind**, select the target real server, and configure the server port and weight in the pop-up window.

Note:

If you set **Default port** first and then select real servers, the port of every real server is the default port.

Step 3. Configure a security group (optional)

You can configure a CLB security group to isolate public network traffic. For more information, see [Configuring a CLB Security Group](#).

Step 4. Modify or delete a listener (optional)

If you need to modify or delete a created listener, click the listener on the **Listener management** page and click



for modification or



for deletion.

Real Server

Real Server Overview

Last updated : 2024-11-29 14:53:01

A real server is a server that is bound to the created CLB instance to process requests forwarded by the CLB instance. When [configuring a CLB listener](#), you need to bind real servers to the listener. Through [different polling methods](#), the CLB instance forwards requests to the real server for processing to ensure application stability and reliability.

Supported Real Server Types

CLB supports the following real server types: instance, IP address, and [Serverless Cloud Function \(SCF\)](#).

Real servers of the instance type include [Cloud Virtual Machine \(CVM\)](#), [Elastic Network Interface \(ENI\)](#), and Elastic Kubernetes Service (EKS) instances.

Real servers of the IP type mainly refer to the private IP addresses of VPCs and IDCs.

Reminders

When adding a real server, you are advised to do the following:

Enable [session persistence](#), so that CLB can maintain a longer TCP connection for reuse by multiple requests, thereby reducing load on the web server and improving CLB throughput.

Make sure that the security group of the real server has inbound rules for CLB listener ports and health check ports.

For more information, see [Configuring CVM Security Groups](#).

References

[Managing Real Servers](#)

[Binding an ENI](#)

[Hybrid Cloud Deployment](#)

[Binding with SCF](#)

Managing Real Servers

Last updated : 2024-10-10 16:57:38

CLB routes requests to real server instances that are running normally. This document describes how to add or delete real servers as needed or when you use CLB for the first time.

Prerequisites

You have created a CLB instance and configured a listener. For more information, see [Getting Started with CLB](#).

Directions

Adding a real server to CLB

Note:

If a CLB instance is associated with an auto scaling group, CVMs in the group will be automatically added to the real servers of CLB. When a CVM instance is removed from the auto scaling group, it will be automatically deleted from the real servers of CLB.

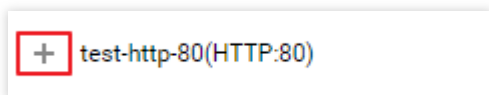
For details on how to add real servers using the API, see [RegisterInstancesWithLoadBalancer](#).

If you have a bill-by-CVM account and select a non-BGP ISP (China Mobile/China Unicom/China Telecom), you can only bind bill-by-traffic and bill-by-bandwidth package CVM instances. For more information about account and ISP types, see [Checking Account Type](#) and [Product Attribute Selection](#).

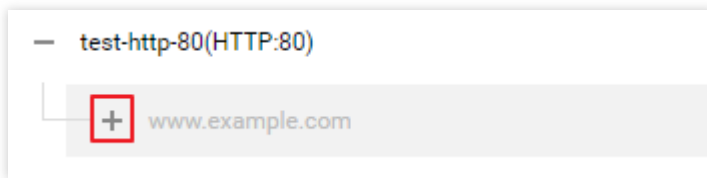
1. Log in to the [CLB Console](#).
2. On the **Instance Management** page, click **Configure Listener** on the right of a CLB instance.
3. On the listener configuration page, select a listener to bind to the backend CVM.

HTTP/HTTPS listener

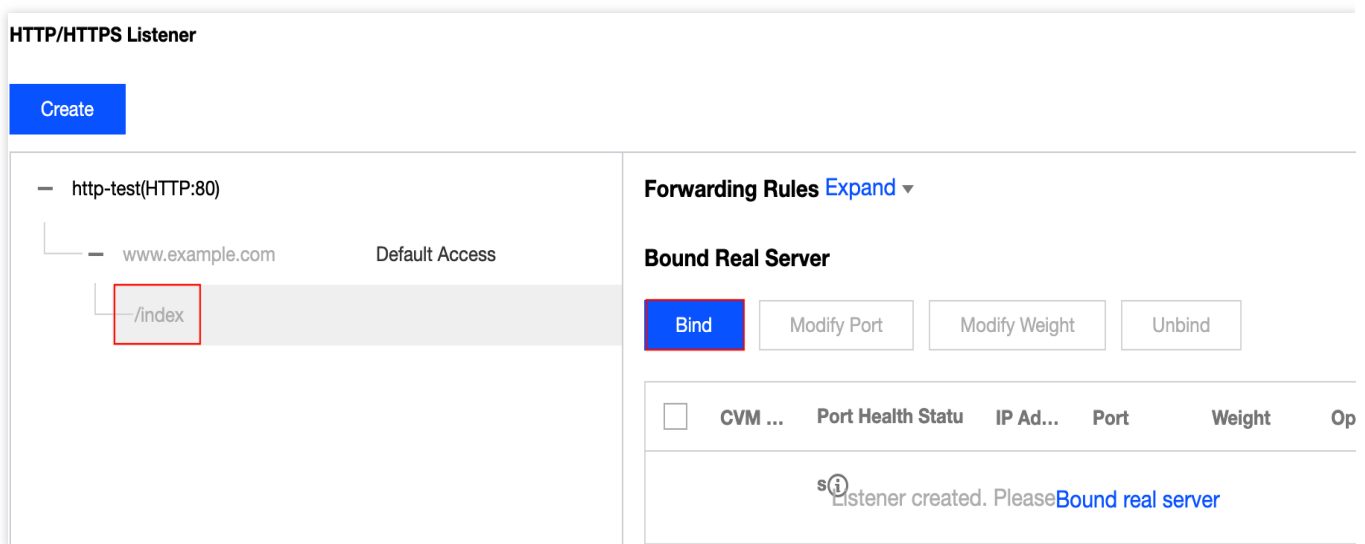
- 3.1.1 In the **HTTP/HTTPS Listener** section, click **+** on the left of the listener you select.



- 3.1.2 Click **+** on the left of the domain name displayed.

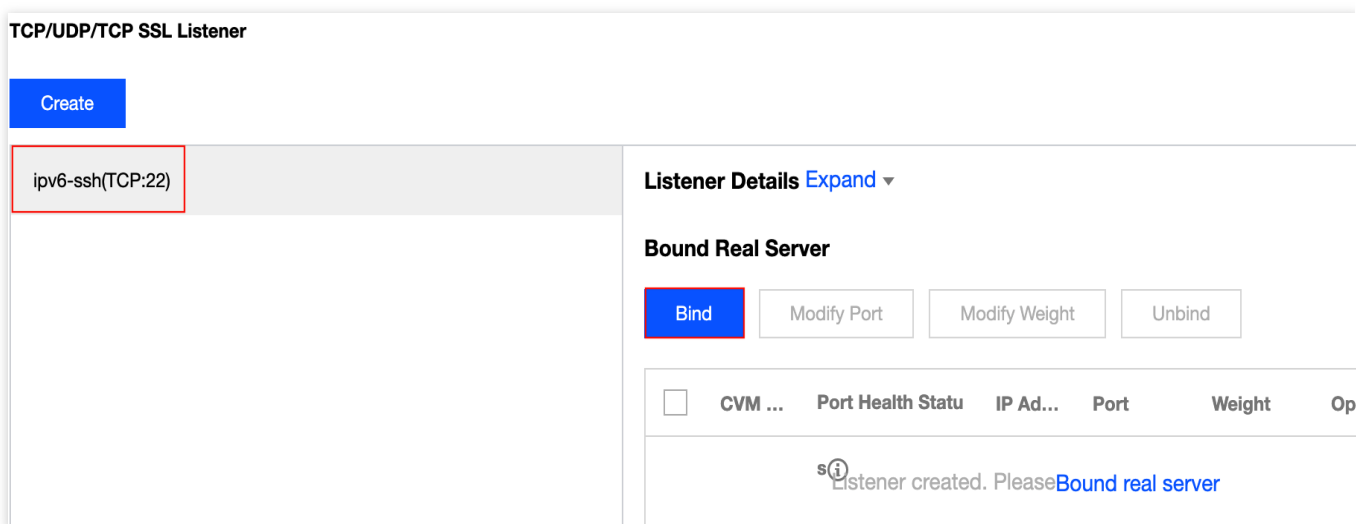


3.1.3 Select the URL displayed and click **Bind**.



TCP/UDP/TCP SSL listener

In the list on the left of the TCP/UDP/TCP SSL listener section, select a listener to bind with the backend CVM.



4. Bind the CLB instance with a real server.

Method 1: In the **Bind with backend service** dialog, click **CVM**, select one or more CVM instances, enter the forwarding port and weight, and click **Confirm**. For more information on ports, please see [Server Common Port](#).

Note:

The pop-up window only displays available CVMs that are not isolated nor expired, in the same region, in the same network environment, and have peak bandwidth greater than 0.

When the CLB instance is bound with multiple real servers, it use the hash algorithm to forward traffic.

The larger the weight is, the more the requests are forwarded. Value range is 0-100 (default: 10). If it is set to 0, the real server stops receiving new requests. If session persistence is enabled, the requests from the real server may be not evenly distributed. For more details, see [Algorithms and Weight Configuration](#).

Bind with backend service

Select an instance

CVM

ENI

Please enter the d

IP address

Search by IP address,

Instance ID/name

10 / page

1

/ 1 page

Press Shift key to select more

Selected (2)

Instance ID/name	Port	Weight	
	80	10	<div><div></div><div></div><div></div><div></div></div> <div><div>Add a port</div><div>Delete</div></div>
	80	10	<div><div></div><div></div><div></div><div></div></div> <div><div>Add a port</div><div>Delete</div></div>

Confirm

Cancel

Method 2: If the CVM instances that need to be bound in batches have the same preset port value, in the pop-up dialog click **CVM**, enter the port value, select the corresponding CVM instances, set the weights, and click **OK** to bind them in batches. For more details on ports, see [Server Common Port](#).

©2013-2024 Tencent Cloud. All rights reserved.

Page 121 of 262

Bind with backend service

Select an instance

CVM

ENI

80

IP address

Search by IP address,

Instance ID/name

10 / page

1

/ 1 page

Press Shift key to select more

Selected (2)

Instance ID/name	Port	Weight ⓘ	
	80	10	<div>Add a port</div> <div>Delete</div>
	80	10	<div>Add a port</div> <div>Delete</div>

Confirm

Cancel

Modifying real server weights for CLB

The real server weight determines the number of CVM requests to be forwarded. When binding a real server, you need to preset its weight. The following shows an example of how to change the real server weight when a HTTP/HTTPS listener is used (which is also applied to a TCP/UDP/TCP SSL listener).

Note:

For details on how to modify the real server weight with the API, see [ModifyLoadBalancerBackends](#).
For more information on weight of the load balancing real server, refer to [Cloud Load Balancer Round-robin Method](#).

1. Log in to the [CLB Console](#).
2. On the **Instance Management** page, click **Configure Listener** on the right of a CLB instance.
3. In the list on the left of the HTTP/HTTPS listener section, click the expand icon to show the instance and listener rules, and select a URL.

©2013-2024 Tencent Cloud. All rights reserved.

Page 122 of 262



4. In the server list on the right of the HTTP/HTTPS listener section, modify the corresponding server weight.

Note:
The larger the weight is, the more the requests are forwarded. Value range is 0-100 (default: 10). If it is set to 0, the real server stops receiving new requests. If session persistence is enabled, the requests from the real server may be not evenly distributed. For more details, see [Algorithms and Weight Configuration](#).

Method 1: Modify weight of a single backend CVM

4.1.1 Locate the CVM instance you want to edit and click the



icon on the left of the weight.

<div>BindModify PortModify WeightUnbind</div>						
<input type="checkbox"/>	CVM ID/Name	Port Health Statu	IP Address	Port	Weight	Ope.
<input type="checkbox"/>		Abnormal		80	10	Unbi
					Edit	
<input type="checkbox"/>		Abnormal		80	10	Unbi

4.1.2 In the pop-up window, enter a new weight and click **Submit**.

Method 2: Modify weights of multiple backend CVMs

Note:
After you perform batch modification, the backend CVMs will use the same weight.

4.1.1 Click checkboxes in front of the CVM instances you want to edit, and click **Modify Weight**.

Bind	Modify Port	Modify Weight	Unbind			
<input checked="" type="checkbox"/>	CVM ID/Name	Port Health Statu	IP Address	Port	Weight	Ope.
<input checked="" type="checkbox"/>		si Abnormal		80	10	Unbi
<input checked="" type="checkbox"/>		Abnormal		80	10	Unbi

4.1.2 In the pop-up window, enter a new weight and click **Submit**.

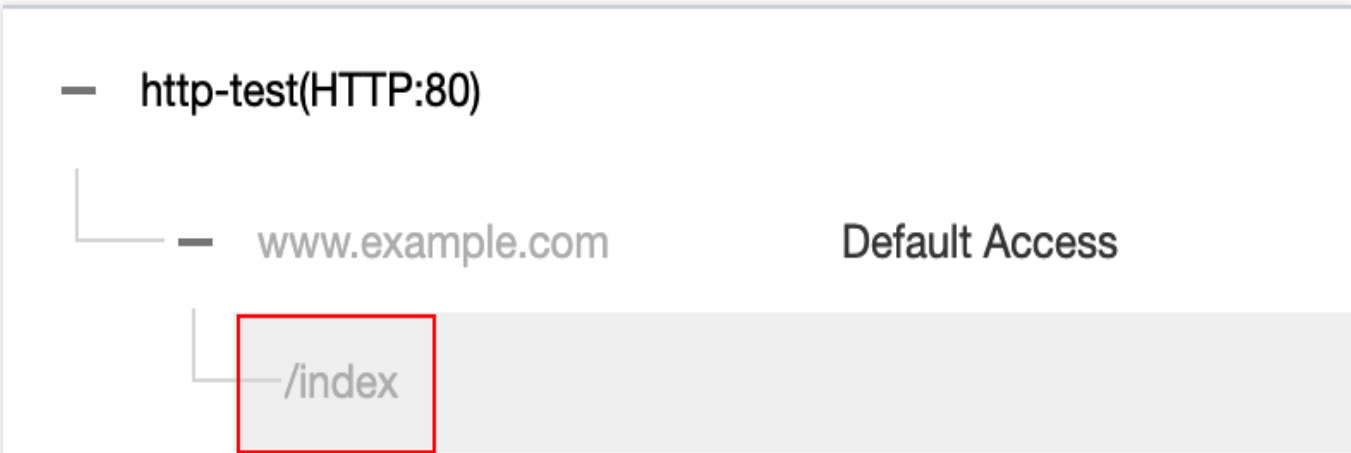
Modifying real server ports for CLB

You can modify real server ports in the CLB console. The following shows an example of how to change the real server port when a HTTP/HTTPS listener is used (which is also applied to a TCP/UDP/TCP SSL listener).

Note:

For details on how to modify the real server port with the API, see [ModifyTargetPort](#).


1. Log in to the [CLB Console](#).
2. On the **Instance Management** page, click **Configure Listener** on the right of a CLB instance.
3. In the list on the left of the HTTP/HTTPS listener section, click the expand icon to show the instance and listener rules, and select a URL.

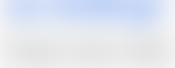







4. In the list on the right of the HTTP/HTTPS listener section, modify the corresponding server port. For more details on how to select ports, see [Server Common Port](#).

Method 1: Modify port of a single backend CVM

4.1.1 Locate the CVM instance you want to edit and click the

 icon on the left of the port.

<div>BindModify PortModify WeightUnbind</div>						
<input type="checkbox"/>	CVM ID/Name	Port Health Statu	IP Address	Port	Weight	Ope.
<input type="checkbox"/>		Abnormal		80 	10 	Unbi
<input type="checkbox"/>		Abnormal		80	10	Unbi






4.1.2 In the pop-up window, enter a new port value and click **Submit**.

Method 2: Modify ports of multiple backend CVMs

Note:

After you perform batch modification, the backend CVMs will use the same port.

4.1 Click checkboxes in front of the CVM instances you want to edit, and click **Modify Port**.

<div>BindModify PortModify WeightUnbind</div>						
<input checked="" type="checkbox"/>	CVM ID/Name	Port Health Statu	IP Address	Port	Weight	Ope.
<input checked="" type="checkbox"/>		 Abnormal		80	10	Unbi
<input checked="" type="checkbox"/>		Abnormal		80	10	Unbi

4.2 In the pop-up window, enter a new port value and click **Submit**.

Unbinding real servers from CLB

You can unbind bound real servers in the CLB console. The following shows an example of how to unbind the real server when a HTTP/HTTPS listener is used (which is also applied to a TCP/UDP/TCP SSL listener).

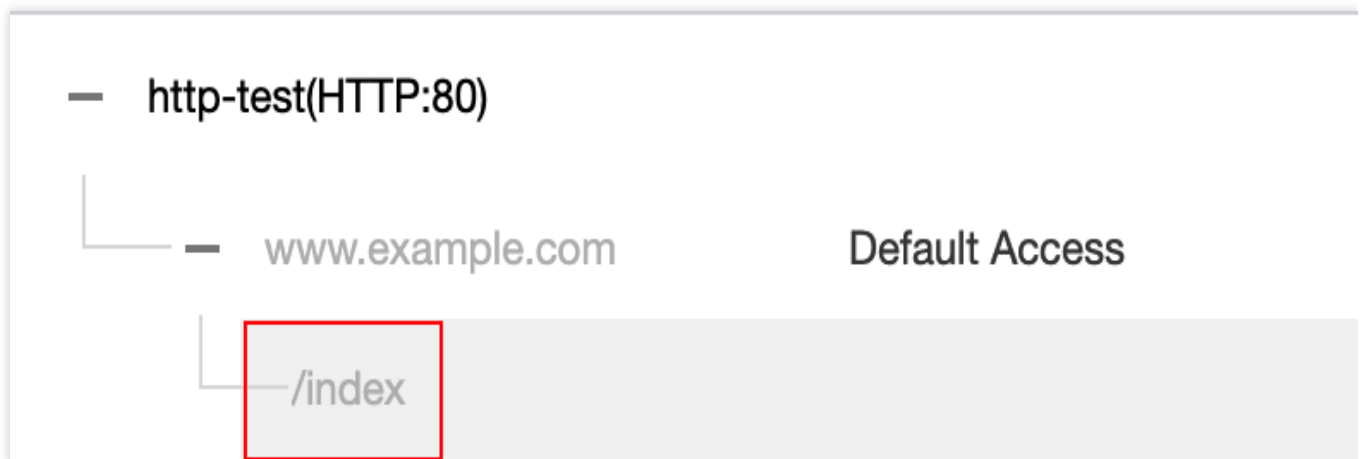
Note:

Unbinding a real server will unbind the CLB instance from the CVM instance, and CLB will immediately stop forwarding requests to it.

Unbinding a real server will not affect the lifecycle of your CVM instance, which can also be added to the real server cluster again.

For details on how to unbind real servers with the API, see [DeregisterTargets](#).





1. Log in to the [CLB Console](#).
2. On the **Instance Management** page, click **Configure Listener** on the right of a CLB instance.
3. In the list on the left of the HTTP/HTTPS listener section, click the expand icon to show the instance and listener rules, and select a URL.



4. In the list on right of the HTTP/HTTPS listener section, unbind the bound real server.

Method 1: Unbind a single backend CVM





- 4.1.1 Select the target CVM instance and click **Unbind**.

Bind	Modify Port	Modify Weight	Unbind			
<input type="checkbox"/>	CVM ID/Name	Port Health Statu	IP Address	Port	Weight	Ope.
<input type="checkbox"/>		<div><div>s</div><div>i</div><div>Abnormal</div></div>		80	10	Unbi
<input type="checkbox"/>		Abnormal		80	10	Unbi

4.1.2 In the pop-up window, check the CVM instance you select and click **Submit**.

Method 2: Unbind multiple backend CVMs

4.1.1 Click checkboxes in front of the CVM instances you want to unbind, and click **Unbind**.

Bind	Modify Port	Modify Weight	Unbind			
<input checked="" type="checkbox"/>	CVM ID/Name	Port Health Statu	IP Address	Port	Weight	Ope.
<input checked="" type="checkbox"/>		<div><div>s</div><div>i</div><div>Abnormal</div></div>		80	10	Unbi
<input checked="" type="checkbox"/>		Abnormal		80	10	Unbi

4.1.2 In the pop-up window, check the CVM instances you select and click **Submit**.

Binding an ENI

Last updated : 2024-10-10 17:01:53

ENI Overview

[Elastic Network Interface \(ENI\)](#) refers to the type of virtual network interfaces that can be bound to CVM instances in VPCs. An ENI can be migrated freely between CVM instances within the same VPC and AZ, helping you easily build high-availability clusters, implement low-cost failover, and manage networks in a more refined manner.

CLB real servers can be bound to both CVM and ENI. Specifically, a CLB instance communicates with the real server over the private network, and if multiple CVM instances and ENIs are bound to the CLB instance, access traffic will be forwarded to the private IPs of the CVM instances and ENIs.

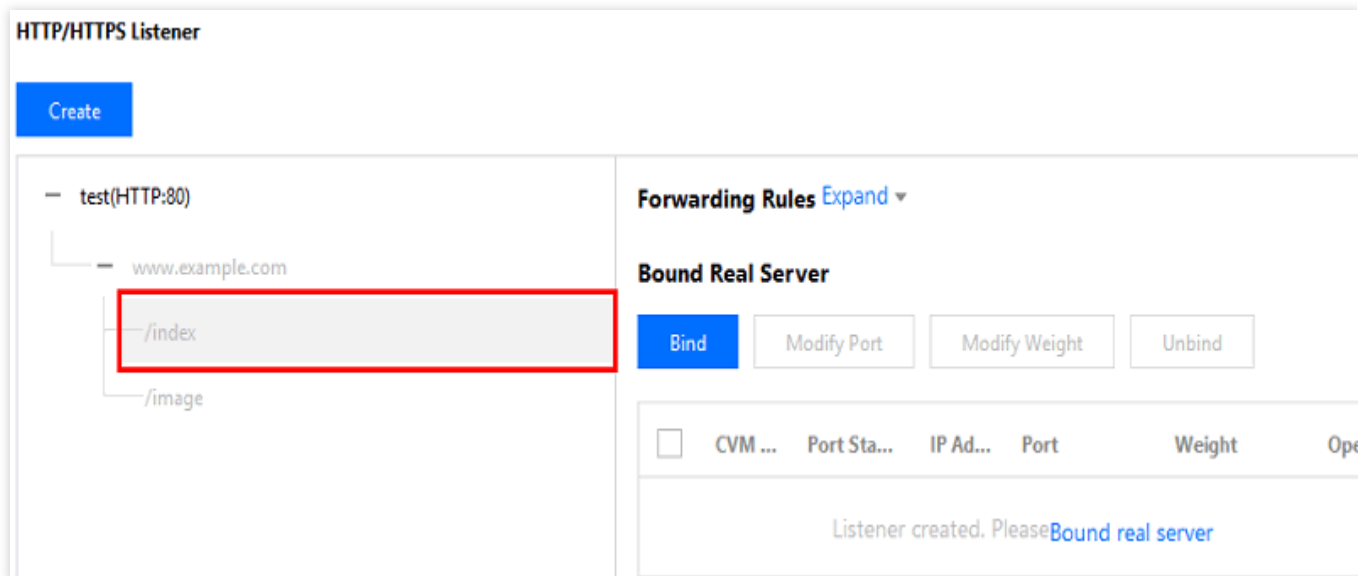
Prerequisites

An ENI must be bound to a CVM instance first before it can be bound to a CLB instance. As a CLB instance only forwards traffic as a load balancer but does not process the business logic, the CVM instance, as a computing resource, is needed to process user requests. Please log in to the [ENI Console](#) to bind the required ENI to the CVM instance first.

ENI South China (Guangzhou) All VPCs Help o						
<div>+ New</div> <div>Use ' ' to split more than one keywords, and press Enter to split tags</div>						
ID/Name	ENI Parameters	Network	Subnet	Bind CVM	Private IP	Operation
	Secondary ENI			-	1	Bind CVM Edit Tags
	Secondary ENI			-	1	Bind CVM Edit Tags
	Secondary ENI			-	1	Bind CVM Edit Tags
	Secondary ENI				1	Unbind CVM Edit Tags etc

Directions

1. You need to configure a CLB listener first. For more information, please see [CLB Listener Overview](#).
2. Click + on the left of the created listener to expand the domain names and URL paths, select the desired URL path, and view the existing real server bound on the right of the listener.



3. Click **Bind** and select the real server to be bound and configure the server port and weight in the pop-up window. You can select "CVM" or "ENI" as the real server.

CVM: you can bind the primary private IPs of primary ENIs of all CVM instances in the same VPC as the CLB instance.

ENI: you can bind all ENI IPs in the same VPC as the CLB instance except the primary private IPs of primary ENIs of CVM instances, such as secondary private IPs of primary ENIs and private IPs of secondary ENIs. For more information on the types of ENI IPs, please see [ENI - Key Concepts](#).

Bound real server

IP

Enter the IP; Separate each on

<input checked="" type="checkbox"/>	ID/Name
<input checked="" type="checkbox"/>	named (Public)/10.20...
<input checked="" type="checkbox"/>	'tke_cls-9cj31525_worker (Public)/10.20...
<input checked="" type="checkbox"/>	d/as-Demo (Public)/10.20...

Selected (3)

Default Port

ID/Name	Port	Weight①	
named (Public)/10.20...	8000	- 10 +	Add Port Delete
'tke_cls-9cj315... (Public)/10.20...	8000	- 10 +	Add Port Delete
d/as-Demo (Public)/10.20...	8000	- 10 +	Add Port Delete

Note: To ensure the forwarding works properly, please set the public network bandwidth to more than 0 MB for the CVM associated with public CLB.

OK

Cancel

4. The specific configuration after binding is as shown below:

HTTP/HTTPS Listener

Create

- Demo(HTTP:80)
- www.example.com
- /index
- /image

Forwarding Rules Expand

Bound Real Server

Bind

Modify Port

Modify Weight

Unbind

<input type="checkbox"/>	CVM ID/Name	Port S...	IP Address	Port	Weight	0
<input type="checkbox"/>		Healthy	9 (public Private)	8000	10	U d
<input type="checkbox"/>	525_worker	Healthy	4 (public Private)	8000	10	U d
<input type="checkbox"/>		Healthy	3 (public Private)	8000	10	U d

Selected0 items, total 3 items

Binding with SCF

Last updated : 2025-04-30 16:08:24

You can implement backend web services by writing SCF functions and bind them with CLB instances to provide services.

Background

Tencent Cloud [Serverless Cloud Function \(SCF\)](#) is a serverless execution environment that enables you to build and run applications without having to purchase and manage servers. After creating a function, you can create a CLB trigger to bind the function and event. The CLB trigger will pass the request content as parameters to the function and return the result from the function back to the requester as the response.

Overview

HTTP/HTTPS general access

Applicable to apps for ecommerce, social media and other services, and web applications for personal blogging, event pages and more. The workflow is as follows:

1. HTTP/HTTPS requests initiated by apps, browsers, H5 pages, or Mini Programs access the SCF function through the CLB instance.
2. After the CLB instance completes the certificate uninstallation, SCF only needs to provide HTTP services.
3. The request is then transferred to the SCF function for subsequent processing, such as writing to the cloud database and calling other APIs.

Switching between CVM and SCF

Applicable to migrating HTTP/HTTPS services from CVM to SCF, especially in the event of failover. The workflow is as follows:

1. The app, browser, H5, or Wechat Mini Program initiates an HTTP/HTTPS request.
2. The request is then resolved to two CLB instances' VIPs by the DNS.
3. One CLB instance forwards the request to the CVM and the other forwards it to the SCF.
4. The switch from CVM to SCF on the backend is complete without affecting the client side.

CVM/SCF business diversion

Applicable to using SCF to handle highly elastic services and CVM to handle daily business in scenarios such as flash sales and snap-up purchase.

1. Through DNS resolution, domain name A is resolved to one CLB instance's VIP and domain name B is resolved to the other CLB instance's VIP.
2. One CLB instance forwards the request to the CVM and the other forwards it to the SCF.

Restrictions

Binding with SCF is only available in Guangzhou, Shanghai, Beijing, Chengdu, Hong Kong (China), Singapore, Mumbai, Tokyo, and Silicon Valley.

SCF functions can only be bound with CLB instances of bill-by-IP accounts but not with bill-by-CVM accounts. If you are using a bill-by-CVM account, we recommend upgrading it to a bill-by-IP account. For more information, please see [Checking Account Type](#).

SCF functions cannot be bound with classic CLB instances.

SCF functions cannot be bound with classic network-based CLB instances.

SCF functions in the same region can be bound with CLB instances. SCF functions can only be bound across VPCs but not regions.

SCF functions can only be bound with IPv4 and IPv6 NAT64 CLB instances, but currently not with IPv6 CLB instances.

SCF functions can only be bound with layer-7 HTTP and HTTPS listeners, but not with layer-7 QUIC listeners or layer-4 (TCP, UDP, and TCP SSL) listeners.

Only SCF event functions can be bound with CLB instances.

One CLB rule can be bound to only one SCF function and does not support binding to other types of real servers at the same time.

The maximum size of the response body for binding a function to a CLB rule cannot exceed 128 KB.

Prerequisites

1. You have created a [CLB instance](#).
2. You have configured an [HTTP](#) or an [HTTPS](#) listener.

Directions

Step 1. Create a function

1. Log in to the [SCF Console](#) and click **Function Service** on the left sidebar.
2. On the **Function Service** page, click **Create**.

3. On the **Create** page, select **Custom** for the creation mode, and enter a function name. Then select the same region that you selected for your CLB instance and **Python3.6** for the runtime environment, enter the following code in the input box (Hello CLB is used for illustration), and click **Complete**.

Note:

When you bind your CLB instance to the SCF function, content needs to be returned in the specific response integration format. For more information, see [CLB Trigger](#).

```
# -*- coding: utf8 -*-
import json
def main_handler(event, context):

    return {
        "isBase64Encoded": False,
        "statusCode": 200,
        "headers": {"Content-Type": "text/html"},
        "body": "<html><body><h1>Hello CLB</h1></body></html>"
    }
```

Step 2. Deploy the function

1. On the "Functions" list page, click the name of the function you created.
2. On the **Function Management** page, select the **Function Codes** tab and click **Deploy** at the bottom.

Step 3. Bind the function

1. Log in to the [CLB Console](#) and click **Instance Management** on the left sidebar.
2. On the **Instance Management** page, click **Configure Listener** on the right of an instance.
3. In the **HTTP/HTTPS Listener** section, select the listener to be bound with an SCF function. Click the **+** icon on the left of the listener and the domain name under it, select the URL path displayed, and click **Bind**.
4. In the pop-up window, select SCF as the target type, set the configuration items, and click **Confirm**.
5. On the **Listener Management** tab, you should see the function bound to the CLB instance in the **Forwarding Rules** section, indicating the CLB trigger is created.

Note:

1. You can also create a CLB trigger in the SCF console to bind the CLB instance with an SCF function. For more information, please see [Creating Triggers](#).
2. During binding to an SCF function, the response body size cannot exceed 128 KB. If this limit is exceeded, CLB will return the error code 403 to the client.

Result Validation

1. If you bind a public network CLB with a cloud function, and set the IP mode to **Static IP**, access the cloud function with the VIP and port of the CLB instance. **Hello CLB** indicates that the cloud function is deployed successfully.
2. If you bind a public network CLB with a cloud function, and set the IP mode to **Dynamic IP**, access the cloud function with the domain name and port of the CLB instance. **Hello CLB** indicates that the cloud function is deployed successfully.
3. If you bind a private network CLB with a cloud function, access the cloud function via a CVM in the same VPC as the CLB instance. **Hello CLB** indicates that the cloud function is deployed successfully.

References

[Creating functions using the console](#)

Cross-Region Binding 2.0 (New)

Last updated : 2025-04-25 11:46:09

Cloud Load Balancer (CLB) supports binding real servers across regions through Cloud Connect Network (CCN). This allows you to select real servers in different regions and bind real servers across VPCs or regions. Currently, the feature is in beta. If you need to experience the feature, [submit a ticket](#).

Note:

Cross-region binding 2.0 is unavailable for classic CLBs.

This feature is available only to bill-by-IP accounts. To check your account type, see [Checking Account Type](#).

Cross-region binding 2.0 and hybrid cloud deployment do not support [Allow by Default in security groups](#). You need to allow the client IP address and service port on the real server.

CLB instances cannot be bound with each other in cross-region binding 2.0 and hybrid cloud deployment scenarios.

Use Cases

1. The cross-region binding feature meets the needs in P2P gaming scenarios where the same server is shared by players from different regions. For example, if your real server cluster is deployed in Guangzhou, you can create CLB instances in Shanghai and Beijing and bind the instances to the same real server cluster in Guangzhou to achieve game acceleration and traffic convergence, ensuring the data transfer quality and reducing the latency.
2. This feature ensures the transfer quality and data consistency in key business transactions, meeting the stringent requirements of the financial industry and payment scenarios.

Differences from Legacy Cross-region Binding

Item	Cross-Region Binding 2.0 (New)	Cross-Region CVM Binding (Legacy)
Binding to services in multiple regions	Supported. In the new version, a CLB instance can be bound to CVM instances in multiple regions at the same time. For example, a CLB instance in the Beijing region can be bound to CVM instances in the Beijing and Shanghai regions at the same time.	Unsupported. In the legacy version, a CLB instance can be bound to CVM instances in only one region. For example, a CLB instance in the Beijing region can be bound to CVM instances in the Shanghai region, but cannot be bound to those in the Beijing and Shanghai regions at the same time.

Switching between cross-region binding and intra-region binding	Supported. In the new version, the original intra-region binding can be switched back after cross-region binding is used.	Unsupported. In the legacy version, after the region attribute of the real server is changed, if the new region is different from that of the CLB instance, you cannot switch back to the original intra-region binding.
Supported CLB types	Public network CLB instances and private network CLB instances	Public network CLB instances
Unbinding CLB automatically when the bound CVM is released	<p>Intra-region binding: If a CLB instance is bound to a CVM instance in the same region, when the CVM instance is released, the CLB instance will be automatically unbound from the CVM instance.</p> <p>Cross-region binding: If a CLB instance is bound to a CVM instance in another region, when the CVM instance is released, the CLB instance will not be automatically unbound from the CVM instance, and you need to manually unbind them.</p>	<p>Intra-region binding: If a CLB instance is bound to a CVM instance in the same region, when the CVM instance is released, the CLB instance will be automatically unbound from the CVM instance.</p> <p>Cross-region binding: If a CLB instance is bound to a CVM instance in another region, when the CVM instance is released, the CLB instance will be automatically unbound from the CVM instance.</p>
Price	Billed in CCN . The costs are controlled in a fine-grained manner, which leads to lower prices.	Billed by daily 95th percentile

Limits

Cross-network real server binding is currently unavailable for classic CLB instances.

This feature is available only to bill-by-IP accounts. To check your account type, see [Checking Account Type](#).

This feature is only supported by VPC but not by classic networks.

This feature is supported on IPv4 and IPv6 NAT64 CLB instances. Dual-stack binding needs to be enabled for IPv6 CLB instances, so that the layer-7 listener can be bound to both IPv4 and IPv6 real servers. Cross-region binding 2.0 and hybrid cloud deployment are supported when the layer-7 listener is bound to an IPv4 IP address, but are not supported when the IPv6 CLB instance is bound to an IPv6 real server.

Cross-region binding 2.0 and hybrid cloud deployment do not support [Allow by Default in security groups](#). You need to allow the client IP address and service port on the real server.

CLB instances cannot be bound with each other in cross-region binding 2.0 and hybrid cloud deployment scenarios.

Both layer-4 and layer-7 (HTTP/HTTPS) CLB services support getting a client IP. For layer-4 CLB, the source IP address obtained on the backend CVM instance is the client IP address. For layer-7 CLB, you can use the X-Forwarded-For or remote_addr field to directly get the client IP address. For more information, see [Obtaining Real Client IPs Over IPv4 CLBs](#).

Prerequisites

1. Application approved. For cross-region binding, please [submit a ticket to apply](#). For cross-border binding, [contact your Tencent Cloud rep](#).
2. You have created a CLB instance. For more information, see [Creating CLB Instances](#).
3. You have created a CCN instance. For more information, see [Creating a CCN Instance](#).
4. You have associated the target VPC with the created CCN instance. For more information, see [Associating Network Instances](#).

Directions

1. Log in to the [CLB Console](#).
2. On the **Instance Management** page, select a region from the top-left corner, and click the instance ID.
3. In the **Real Server** section on the **Basic Info** tab, click **Configure** to bind a private IP address of another VPC.
4. Click **Submit** in the pop-up window.
5. In the **Real Server** section on the **Basic Info** tab, you can see that **Binding IP of Other VPCs** is enabled, which indicates that you can bind in-cloud IP addresses.
6. On the instance details page, open the **Listener Management** tab, and bind a real server to the CLB instance in the listener configuration section. For more information, see [Managing Real Servers](#).
7. In the pop-up window, select **Other VPC** for **Network type**, click **CVM**, select one or multiple CVM instances to be associated with, enter the forwarding port and weight, and click **Confirm**. For more information about ports, see [Server Common Port](#).
8. Now in the **Real Servers Bound** section, you can view the bound CVM instances of other regions.

Hybrid Cloud Deployment

Last updated : 2025-05-07 18:22:09

In a hybrid-cloud environment, you can bind CLB instances to IPs of servers in the local IDC, so as to connect real servers across VPCs and IDCs.

This feature is in beta test. To try it out, for cross-region binding in the Chinese mainland, [submit a ticket](#); for cross-region binding outside the Chinese mainland, [contact us](#).

Benefits

Forward requests to both on-cloud servers and local IDC servers at the same time.

Utilize the public network access capabilities of Tencent Cloud.

Support features of CLB, such as layer-4/7 access, health check, and session persistence.

Set up connections between private networks by using [Cloud Connect Network](#), which supports fine-grained routing and tiered pricing.

Limitations

Cross-region binding 2.0 is not available for classic CLBs.

This feature is available only to bill-by-IP accounts. To check your account type, see [Checking Account Type](#).

This feature is not available for CLBs based on the classic network.

This feature is supported on IPv4 and IPv6 NAT64 CLB instances. The IPv6 CLB instance needs to enable the dual-stack mixing binding, which allows the layer-7 listener to bind both IPv4 and IPv6 CVM instances. On this basis, the CLB instance supports cross-region binding 2.0 and hybrid cloud deployment.

Cross-region binding 2.0 and hybrid cloud deployment do not support [Allow Traffic by Default in security groups](#), for which you need to allow the client IP and service port on the real server.

CLB instances cannot be bound with each other in cross-region binding 2.0 and hybrid cloud deployment scenarios.

This feature is only available in Guangzhou, Shanghai, Jinan, Hangzhou, Hefei, Beijing, Tianjin, Chengdu, Chongqing, Nanjing, Wuhan, Beijing Finance, Shanghai Finance, Hong Kong (China), Singapore, Silicon Valley, Frankfurt, São Paulo.

TCP and TCP SSL listeners need to use TOA on the real server to get the source IP. For more information, see [Obtaining Real Client IPs via TOA in Hybrid Cloud Deployment](#).

HTTP and HTTPS listeners need to use `X-Forwarded-For` (XFF) to get the source IP.

UDP listeners cannot get the source IP.

Prerequisites

1. Submit the application to join the beta. For cross-region binding in the Chinese mainland, [submit a ticket](#) for application. For cross-region binding outside the Chinese mainland, [contact your Tencent Cloud rep.](#)
2. Create a CLB instance. For more information, see [Creating CLB Instances](#).
3. Create a CCN instance. For more information, see [Creating a CCN Instance](#).
4. Bind the direct connect gateway associated with the IDC and the target VPC to the created CCN instance. For more information, see [Associating Network Instances](#).

Directions

1. Log in to the [CLB Console](#).
2. On the **Instance Management** page, click the ID of the target CLB instance.
3. On the **Basic Info** tab of the **Real Server** section, click **Configure** to bind a private IP of another VPC.
4. Click **Submit** in the pop-up dialog box.

5. On the **Basic Info** tab of the **Real Server** section, click **Add SNAT IP**.

6. In the pop-up dialog box, select **Subnet**, click **Add** to assign an IP, and click **Save**.

Note :

A SNAT IP is mainly used in hybrid cloud deployment where requests are forwarded to IDC servers. It must be assigned when you bind a CLB instance to an IP in the IDC that is interconnected with CNN, and serves as the private IP of your VPC.

A maximum of 10 SNAT IPs can be configured for each CLB instance.

Each CLB instance configures one SNAT IP in one forwarding rule, and supports 55,000 max connections after being bound to one real server. If you configure more SNAT IPs or real servers, the number of connections increases proportionally. Assume that you configure 2 SNAT IPs for the CLB instance and bind 10 ports to the real server, resulting in a maximum of 1.1 million connections (2 x 10 x 55,000). You can calculate how many SNAT IPs to assign based on the number of connections.

Note that deleting a SNAT IP disconnects all connections on the IP.

7. On the instance details page, open the **Listener Management** tab, and bind a real server to the CLB instance in the listener configuration section. For more information, see [Managing Real Servers](#).
8. In the pop-up dialog box, select **Other Private IP**, click **Add a private IP**, and enter the target IDC private IP, port, and weight. Then click **Confirm**. For more information on ports, see [Server Common Port](#).

9. Now you can view the bound IDC private IP in the **Bound Real Servers** section.

References

[Cross-Region Binding 2.0 \(New\)](#)

Configuring CVM Security Groups

Last updated : 2024-10-10 17:37:00

Overview of CVM Security Group

The backend CVM instances of CLB can perform access control through [Security Group](#), which acts as a firewall. You can associate one or more security groups with a backend CVM, and add one or more rules to each security group to control the traffic access permissions of different servers. You can modify the rules for a security group at any time, and the new rules are automatically applied to all instances associated with that security group. For more information, see the [Security Group](#). In a VPC as instructed in [Overview](#), you can also use a network ACL as instructed in [Rule Overview](#) for access control.

Configuration of CVM Security Group

You need to allow the client IP and open the service port in the CVM security group.

If you want to use a CLB instance to forward business traffic to your CVM instance, the CVM security group should be configured as follows to ensure effective health checks:

1. Public network CLB: You need to allow the CLB VIP in the security group of the backend CVM, so that the CLB instance can use the VIP to check the health status of the backend CVM.

2. Private network CLB:

For private network CLB (formerly called the application private network CLB), if your CLB instance is in a VPC, the CLB VIP needs to be allowed in the security group of the backend CVM for health checks; if your CLB instance is in a basic network, no additional configuration is needed as the health check IP is allowed by default.

For classic private network CLB, if your CLB instance was created before December 5, 2016 and is in a VPC, the CLB VIP needs to be allowed in the security group of the backend CVM for health checks; otherwise, no additional configuration is needed as the health check IP is allowed by default.

Configuration Sample

This example shows a sample of configuring CVM security groups when accessing a CVM through the CLB. To configure the rules of CLB security groups, please see [Configuring CLB Security Group](#).

Application Scenario 1:

For a public network CLB configured with a TCP:80 listener and a backend service port 8080, if you want to allow only

the ClientA IP and ClientB IP to access the CLB, you need to configure the inbound rules of the backend CVM security group as follows:

```
ClientA IP + 8080 allow
ClientB IP + 8080 allow
CLB VIP    + 8080 allow
0.0.0.0/0  + 8080 drop
```

Application Scenario 2:

For a public network CLB configured with a HTTP:80 listener and a backend service port 8080, if you want to allow all Client IPs to access the CLB, you need to configure the inbound rules of the backend CVM security group as follows:

```
0.0.0.0/0 + 8080 allow
```

Application Scenario 3:

Allow the CLB VIP on the CVM security group to perform health check. For a private network CLB (formerly "application private network CLB") using a VPC and configured with TCP:80 listener and real server port 8080, if you want to only allow the Client IPs (ClientA IP and ClientB IP) to access the CLB VIP, and to restrict Client IP to only access backend CVMs bound with the CLB,

a. Configure the security group inbound rules for the real server as follows:

```
ClientA IP + 8080 allow
ClientB IP + 8080 allow
CLB VIP    + 8080 allow
0.0.0.0/0  + 8080 drop
```

b. Configure the security group outbound rules for the server used as Client as follows:

```
CLB VIP    + 8080 allow
0.0.0.0/0  + 8080 drop
```

Application Scenario 4:

After December 5, 2016, for a newly purchased classic private network CLB using a VPC, you need to allow the Client IP only for the CVM security group. It is not necessary to allow the CLB VIP, and the health check IP is allowed by default. Configure this CLB with TCP:80 listener and real server port 8080. If you want to only allow the Client IPs (ClientA IP and ClientB IP) to access the CLB VIP, and to restrict Client IPs to only access backend CVMs bound with the CLB,

a. Configure the security group inbound rules for the real server as follows:

```
ClientA IP + 8080 allow
ClientB IP + 8080 allow
0.0.0.0/0  + 8080 drop
```

b. Configure the security group outbound rules for the server used as Client as follows:

```
CLB VIP      + 8080 allow
0.0.0.0/0    + 8080 drop
```

Application Scenario 5: Blocklist

If you need to configure a blocklist for some client IPs to deny their access requests, you can configure the security group associated with the cloud services. The security group rules need to be configured as follows:

Add the Client IP and port to be rejected into the security group, and select the option in the policy column to reject access from this IP.

Add another security group rule after completing the above configuration to allow access requests to the port from all IPs by default.

When the configuration completes, the security group rules are as follows:

```
clientA IP + port drop
clientB IP + port drop
0.0.0.0/0  + port accept
```

Note:

Follow the steps above **strictly in the given order**, otherwise the blocklist configuration may fail.

The security group is stateful. The above configurations are all configurations of **inbound rules**.

Operation Guide of CVM Security Groups

Managing Backend CVM Security Groups Using the Console

1. Log into the [CLB Console](#) and click the corresponding CLB instance ID to enter the CLB details page.
2. On the page of CVMs bound to the CLB, click the target backend CVM ID to enter the CVM details page.
3. Click the **Security Group** tab. On the tab, bind/unbind a security group.

Managing backend CVM security groups using Tencent Cloud API

See [AssociateSecurityGroups](#) and [DisassociateSecurityGroups](#).

Health Check

Health Check Overview

Last updated : 2024-11-27 14:54:51

CLB instances determine the availability of real servers through health checks, preventing frontend businesses from being affected by real server exceptions and improving the overall availability of businesses.

After health check is enabled, regardless of the weights of real servers (including 0), the CLB instance will always perform a health check. You can check the health check status in the **Health status** column on the instance list or on the listener's bound real server details page.

If a real server instance is abnormal, the CLB instance will automatically forward new requests to other normal real servers.

Once the abnormal instance is recovered, it will be used in the CLB service again and will receive new requests.

If all real servers are found abnormal, requests will be forwarded to all real servers.

If health check is disabled, the CLB instance will forward traffic to all real servers including those abnormal ones.

Therefore, we strongly recommend enabling health check for the CLB instance to automatically check real servers and remove abnormal ones.

By default, passive health check is enabled (and cannot be disabled) for layer-4 TCP SSL listeners and layer-7 HTTP/HTTPS listeners. The CLB instance forwards traffic to a real server and records the health status of the real server. If forwarding fails, the CLB instance will try to forward traffic to other real servers and adds 1 to the failure count of the failed real server. If the failure count reaches 3, the real server is blocked for 10 seconds. After the blocking period ends, the CLB instance resumes forwarding traffic to the real server and monitoring the health status of the real server.

Health Check Status

Description of Health Check Status of a Single Listener

The health check status of real servers is described as follows:

Status	Description	Whether to Forward Traffic
Detecting	The status of a new real server during the period of check interval × healthy threshold. For example, assume the check interval is 2 seconds and the healthy threshold is 3 times, the real server remains in this status for 6 seconds.	No.
Healthy	The real server is normal.	Yes.

Abnormal	The real server is abnormal.	No. For a layer-4 listener or layer-7 URL-based rule, if a CLB instance detects that all real servers are unhealthy, it will forward requests to all real servers.
Disabled	Health check is disabled.	Yes.

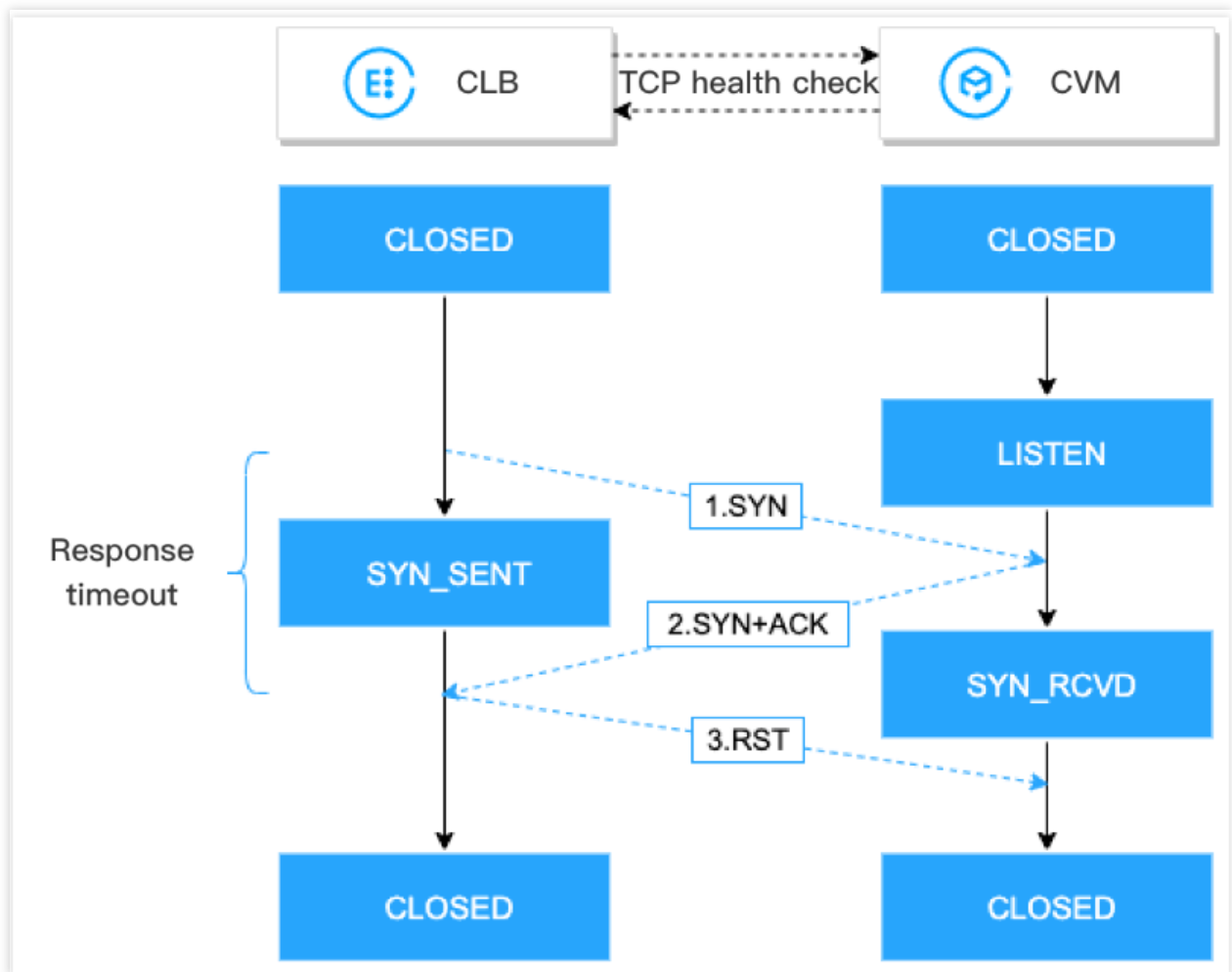
Description of Health Check Status of the List Page

It is displayed based on the health check conditions of all listeners under the instance:

Status	Description
Normal	The real servers of all listeners under this instance are normal. The health checks of all listeners under this instance are not enabled.
Abnormal	If any listener under this instance is abnormal, it will be displayed as abnormal.
Not configured	No listener/rule is configured for this instance. No listener under this instance is bound to the real server and no listener is abnormal.

TCP Health Check

For layer-4 TCP listeners, you can configure TCP health check to obtain the status of real servers through SYN packets, i.e., TCP three-way handshake. Also, to this end, you can customize the request and return content of the protocol.

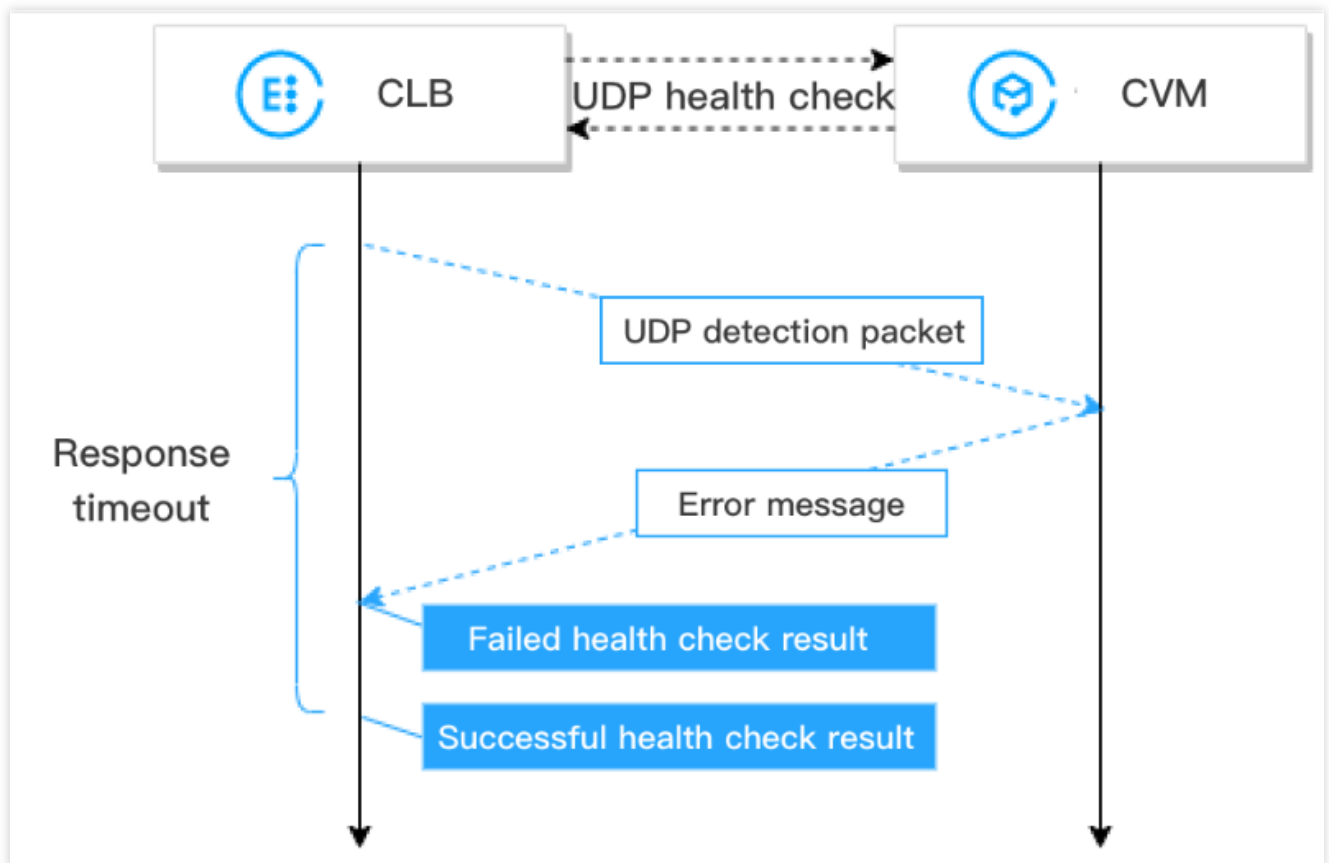


TCP health check mechanism is as follows:

1. A CLB instance sends a SYN connection request packet to a real server (private IP address and health check port).
2. After receiving the SYN request packet, the real server will return a SYN-ACK response packet if the port is listening normally.
3. If the CLB instance receives the returned SYN-ACK response packet within the response timeout period, it indicates that the real server is normal and the health check is successful. Then the CLB instance will send the real server a TCP Reset (RST) packet to cut the TCP connection.
4. If the CLB instance does not receive the returned SYN-ACK response packet within the response timeout period, it indicates that the real server is abnormal and the health check failed. Then the CLB instance will send the real server a TCP Reset (RST) packet to cut the TCP connection.

UDP Health Check

For layer-4 UDP listeners, you can configure UDP health check to obtain the status of a real server by running the `Ping` command and sending UDP detection packets to the health check port. Also, to this end, you can customize the request and return content of the protocol.



UDP health check mechanism is as follows:

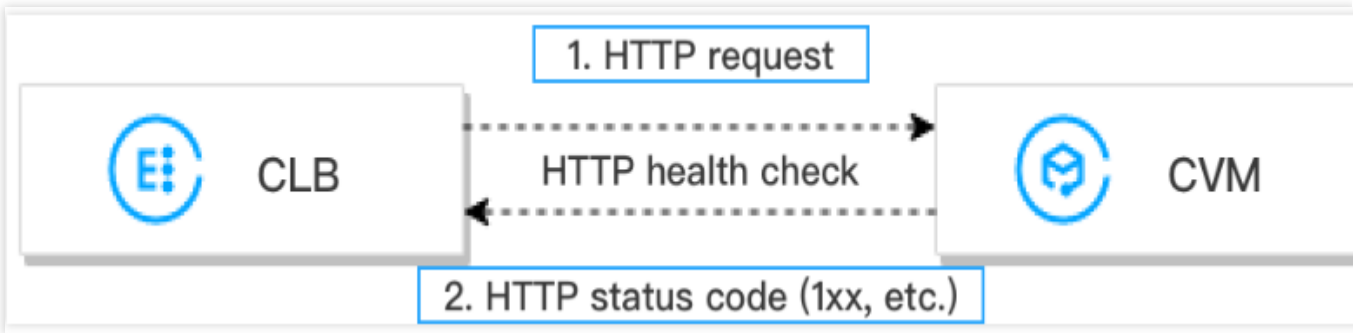
1. A CLB instance sends a `Ping` command to the private IP address of a real server.
2. Then the CLB instance sends a UDP detection packet to the real server (private IP address and health check port).
3. If the `Ping` command succeeds and the real server does not return the error `port XX unreachable` within the response timeout period, it indicates that the real server is normal and the health check is successful.
4. If the `Ping` command fails or the real server returns the error `port XX unreachable` within the response timeout period, it indicates that the real server is abnormal and the health check failed.

Note:

1. UDP health checks are based on ICMP, therefore, real servers need to be allowed to reply ICMP packets (i.e., `Ping` command is supported) and ICMP "port unreachable" packets (i.e., the port can be detected).
2. If a Linux server is used as a real server, the speed of the server to send ICMP packets will be limited during high concurrency as the Linux server has a mechanism of defending itself from ICMP attacks. In this case, although the real server is abnormal, it cannot return the error `port XX unreachable` to the CLB instance. Then the CLB instance will determine that the health check is successful, so the actual status of the real server cannot be returned. Solution: You can configure the UDP health check with custom input and output strings. So in a health check, the custom input string will be sent to the real server, and the result will be determined as successful only after the CLB instance receives the custom response string. This method is based on the real server, which needs to process the health check input string and return the custom output string.

HTTP Health Check

For layer-4 TCP listeners and layer-7 HTTP/HTTPS listeners, you can configure HTTP health check to obtain the status of real servers by sending HTTP requests.



HTTP health check mechanism is as follows:

1. According to the health check configuration, a CLB instance can send HTTP requests (with the target domain name specified) to (the private IP address, health check port, and check path of) a real server.
2. After receiving the request, the real server will return the corresponding HTTP status code.
3. If the CLB instance receives the returned HTTP status code within the response timeout period and the HTTP status code matches the set one, it indicates that the health check is successful, otherwise, failed.
4. If the CLB instance does not receive the response from the real server within the response timeout period, it indicates that the health check failed.

Note:

For layer-7 HTTPS listeners, if HTTP is selected as the backend protocol of the HTTPS listener's forwarding rules, HTTP health check will be conducted; if HTTPS is selected, HTTPS health check will be conducted. HTTPS health checks are basically the same as [HTTP health checks](#). The difference is that in HTTPS health checks, HTTPS requests are sent and the status of a real server is determined based on the returned HTTPS status code.

Health Check Time Window

CLB health check mechanism improves business availability, but frequent health check failures can cause unnecessary server switches, compromising system availability. Therefore, health check status can be switched between healthy and abnormal only if the results are being the same in a health check time window for several times. The health check time window is based on the factors below:

Health Check Factor	Description	Default Value
Response timeout	Maximum response timeout period for a health check.	2 seconds

	If a real server fails to respond within the timeout period, the real server is considered as abnormal. Value range: 2-60 seconds.	
Check interval	Interval between two health checks. Value range: 5-300 seconds.	5 seconds
Unhealthy threshold	If a real server failed the health check for n (a customizable value) consecutive times, the real server is considered unhealthy, and Abnormal is displayed in the console. Value range of n: 2-10.	3 times
Healthy threshold	If a real server passed the health check for n (a customizable value) consecutive times, the real server is considered healthy, and Healthy is displayed in the console. Value range of n: 2-10.	3 times

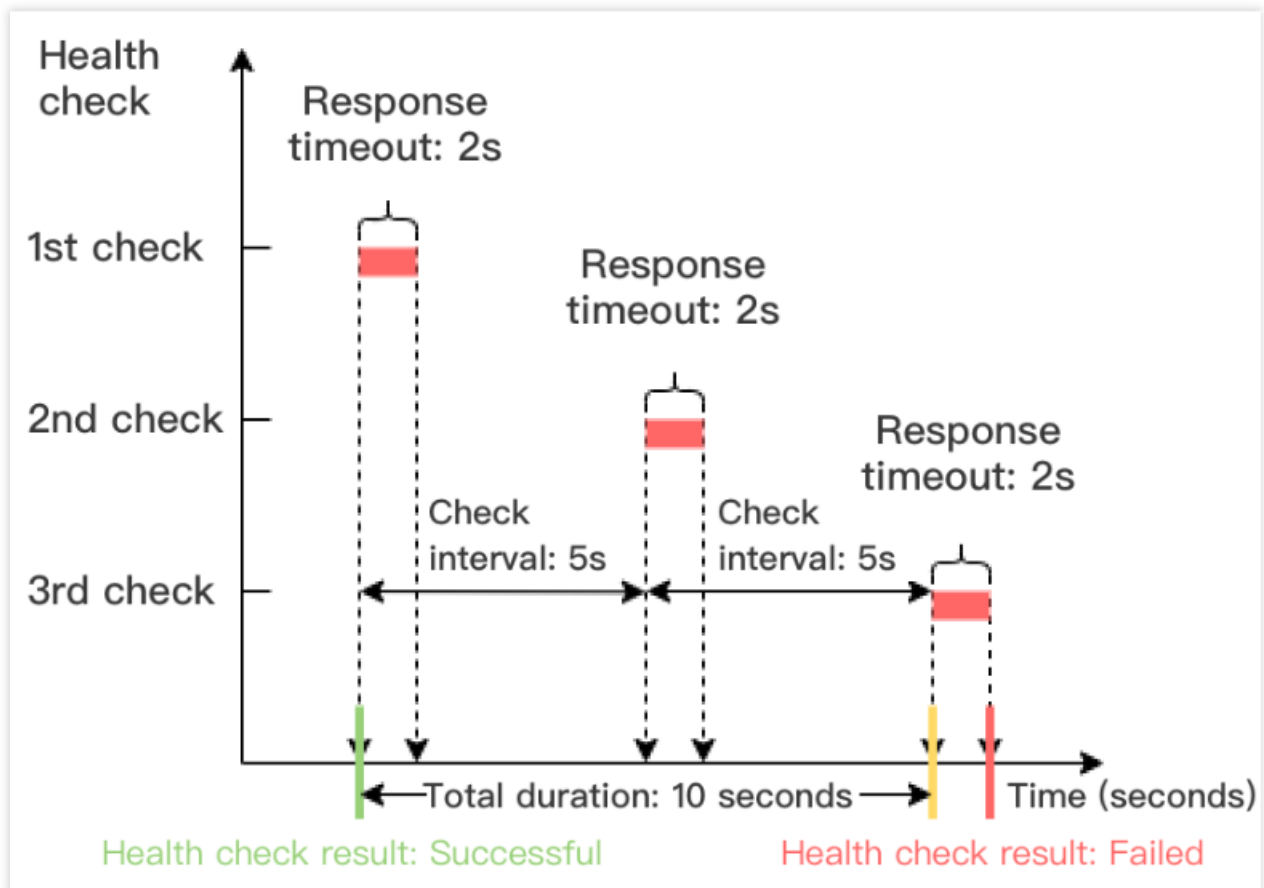
The calculations of **layer-4 health check time window** are as follows:

Note:

Layer-4 health check, namely the TCP health check or UDP health check, the time interval between two checks is the set value, no matter the result is successful or whether the response times out.

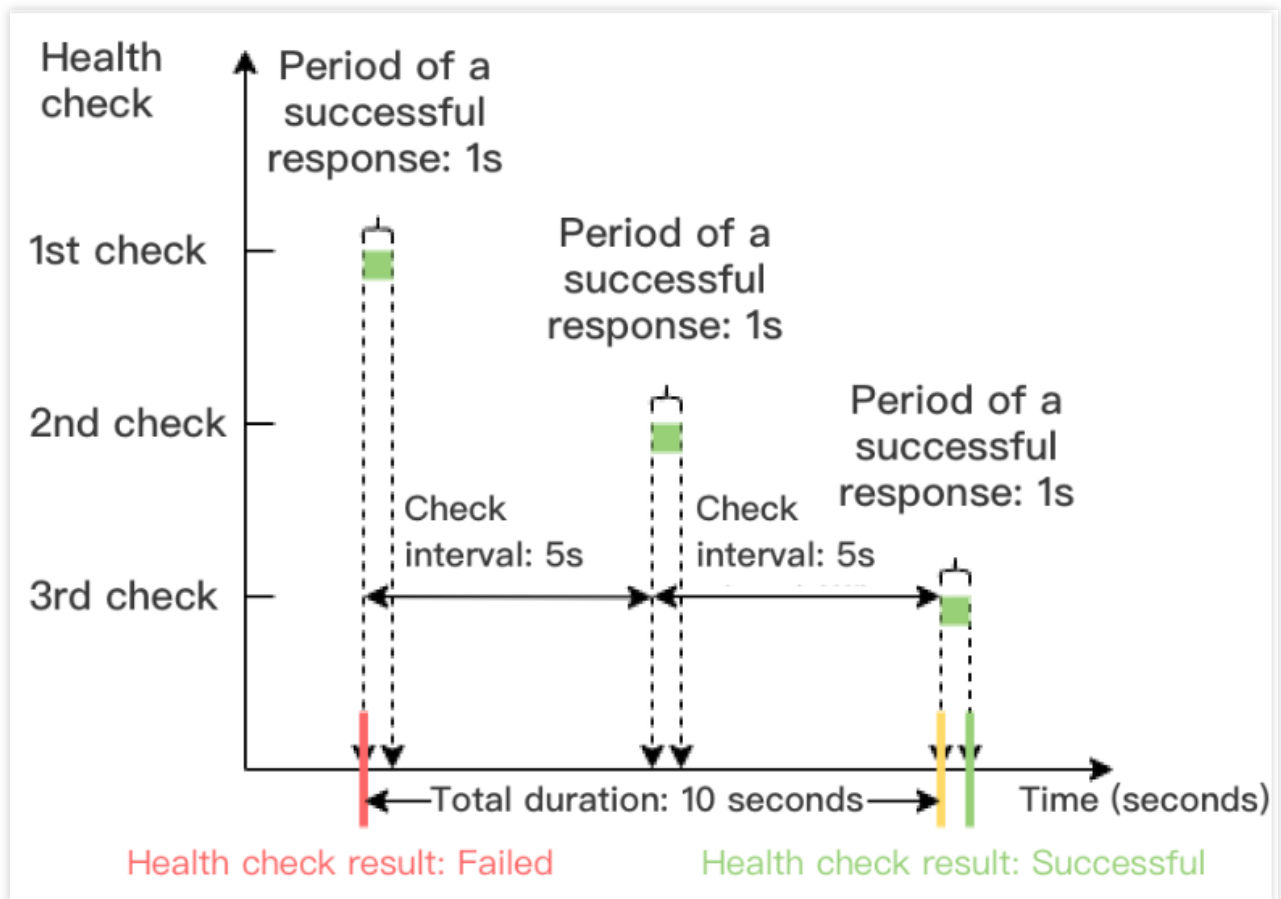
Time window of a health check with a failed result = Check interval × (Unhealthy threshold - 1)

In the example below, the health check response timeout period is 2 seconds, check interval is 5 seconds, and the unhealthy threshold is 3 times, so the time window of a health check with a failed result = 5 × (3 - 1) = 10 seconds.



Time window of a health check with a successful result = Check interval \times (Healthy threshold - 1)

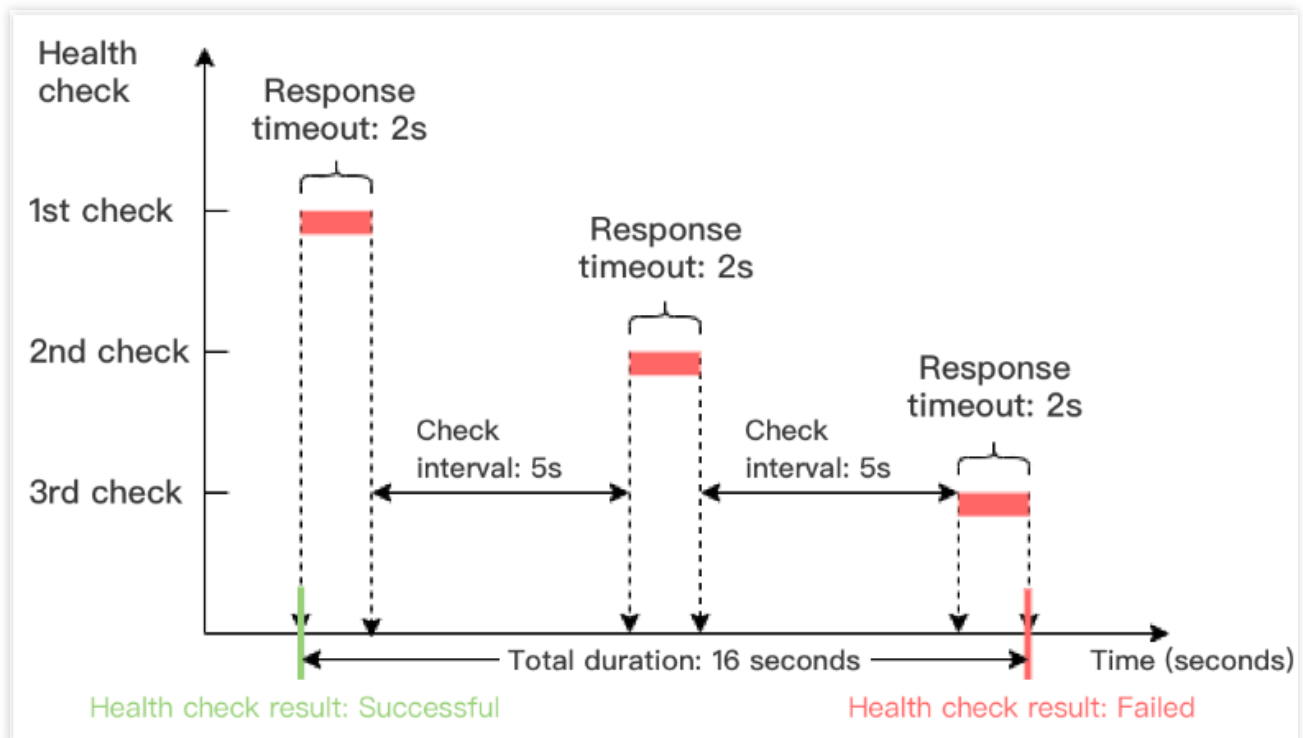
In the example below, the period of a successful health check response is 1 second, check interval is 5 seconds, and the healthy threshold is 3 times, so the time window of a health check with a successful result = $5 \times (3 - 1) = 10$ seconds.



The calculations of **layer-7 health check time window** are as follows:

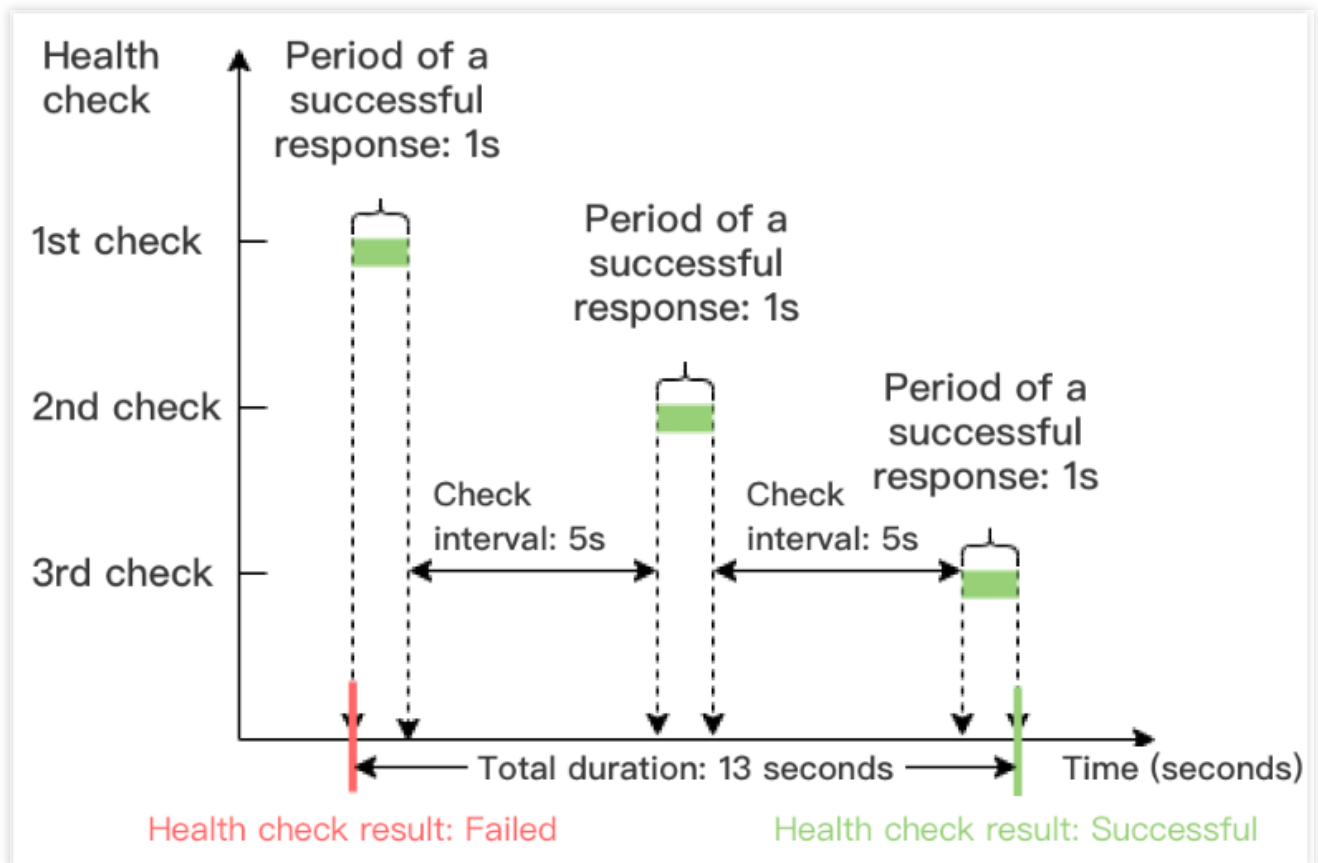
Time window of a health check with a failed result = Response timeout period × Unhealthy threshold + Check interval × (Unhealthy threshold - 1)

In the example below, the health check response timeout period is 2 seconds, check interval is 5 seconds, and the unhealthy threshold is 3 times, so the time window of a health check with a failed result = $2 \times 3 + 5 \times (3 - 1) = 16$ seconds.



Time window of a health check with a successful result = Period of a successful health check response × Healthy threshold + Check interval × (Healthy threshold - 1)

In the example below, the period of a successful health check response is 1 second, check interval is 5 seconds, and the healthy threshold is 3 times, so the time window of a health check with a successful result = $1 \times 3 + 5 \times (3 - 1) = 13$ seconds.



Health Check Identifiers

After CLB health checks start, the real server will receive health check requests in addition to normal business requests. A health check request may have the following properties:

The health check source IP address is the CLB VIP or the 100.64.0.0/10 IP range.

A health check request from layer-4 listeners (TCP, UDP, and TCP SSL) will be marked with "HEALTH CHECK".

For a health check request from layer-7 listeners (HTTP and HTTPS), the value of the `User-Agent` header is `clb-healthcheck`.

Note:

For a health check request from private network classic CLB instances, the health check source IP address is the `169.254.128.0/17` IP range.

For a health check request from classic network CLB instances, the health check source IP address is the physical IP address.

References

[Configuring Health Check](#)

[Configuring Health Check Logs](#)

Configuring Alarm Policy

Configuring Health Check

Last updated : 2024-01-04 20:51:00

When configuring listeners, you can enable health check to obtain the availability information of real servers. For more information about health check, see [Health Check Overview](#).

Restrictions

TCP listeners for IPv6 Cloud Load Balancer (CLB) instances do not support HTTP health checks or custom protocol health checks.

UDP listeners for IPv6 CLB instances do not support port-based health checks.

Prerequisites

1. You have created a CLB instance. For more information, see [Creating CLB Instances](#).
2. You have created a CLB listener.

To create a TCP listener, see [Configuring a TCP Listener](#).

To create a UDP listener, see [Configuring a UDP Listener](#).

To create a TCP SSL listener, see [Configuring a TCP SSL Listener](#).

To create an HTTP listener, see [Configuring an HTTP Listener](#).

To create an HTTPS listener, see [Configuring an HTTPS Listener](#).

TCP Listener

Layer-4 TCP listeners support three types of health checks, namely the layer-4 TCP health check, layer-7 HTTP health check, and custom protocol health check.

TCP health checks are conducted with SYN packets, that is, TCP three-way handshakes are initiated to obtain the status information of real servers.

HTTP health checks are conducted by sending HTTP requests to obtain the status information of real servers.

Custom protocol health checks are conducted by customizing the input and output content of the application layer protocol to obtain the status information of real servers.

Configuring TCP health check

1. Configure a listener to the step of **Health check** as instructed in [Prerequisites](#).
2. On the **Health check** tab, select **TCP** for **Check method**.

CreateListener

Basic configuration

2Health check

3Session persistence

Health check

Detect and remove abnormal server ports automatically.

Source IP ⓘ

CLB VIP

IP range starting with 100.64

Protocol

TCP

HTTP

Custom protocol

Checking port

RS port by default. Unless you want to specify a port, please leave it em

Show advanced options ▼

Back

Next

Parameter	Description
Health check	You can enable or disable health check. We recommend that you enable health check for automatic checks on real servers and removal of abnormal ports.
Health check source IP	The source IP address of health check packets. The default value is the 100.64.0.0/10 IP range. By using this IP range, you can solve the container loopback problem in TKE scenarios. Existing users can choose to use a CLB VIP as the health care source IP address.
Check method	TCP health checks are conducted if TCP is selected.
Port	This parameter is optional. We recommend not specifying the port unless you need to check a specific port. The real server port will be checked if the port is not specified here.
Show advanced options	For more information, see Advanced Options .

Configuring HTTP health check

1. Configure a listener to the step of **Health check** as instructed in [Prerequisites](#).
2. On the **Health Check** tab, select **HTTP** as the protocol.

CreateListener

✓ Basic configuration

2 Health check

3 Session persistence

Health check

Detect and remove abnormal server ports automatically.

Source IP ⓘ

CLB VIP IP range starting with 100.64

Protocol

TCP HTTP Custom protocol

Checking port

RS port by default. Unless you want to specify a port, please leave it empty.

Check domain

(Optional) It's recommended to set this field.

It only supports letters, digits, "-" and "."; the host field is omitted by default.

Path

/

It defaults to check the root directory of the real server. It should start with "/"; up to 80 chars; allowing letters, numbers, "_", "-", ".", "/", "=", "?".

HTTP request method ⓘ

GET

HTTP version ⓘ

HTTP/1.1

Normal status code ⓘ

http_1xx http_2xx http_3xx http_4xx http_5xx

When the status code is http_1xx, http_2xx, http_3xx, http_4xx, the back-end server is considered active

Show advanced options ▾

Back

Next

Parameter	Description
Health check	You can enable or disable health check. We recommend that you enable health check for automatic checks on real servers and removal of abnormal ports.
Health check source IP	The source IP address of health check packets. The default value is the 100.64.0.0/10 IP range. By using this IP range, you can solve the container loopback problem in TKE scenarios.

	Existing users can choose to use a CLB VIP as the health care source IP address.
Check method	HTTP health checks are conducted if HTTP is selected.
Port	This parameter is optional. We recommend not specifying the port unless you need to check a specific port. The real server port will be checked if the port is not specified here.
Check domain	<p>Limits on a health check domain name:</p> <p>Length: 1 to 80 characters.</p> <p>The default value is the forwarding domain name.</p> <p>Regular expressions are not supported. If your forwarding domain name is a wildcard domain name, you need to specify a fixed (non-regular) domain name as the health check domain name.</p> <p>Supported characters: lowercase letters (a to z), digits (0 to 9), decimal points (.), and hyphens (-).</p>
Path	<p>Limits on a health check path:</p> <p>Length: 1 to 200 characters.</p> <p>A health check path must start with /. The default value is /.</p> <p>Regular expressions are not supported. You are advised to specify a fixed URL (static webpage) for health checks.</p> <p>Supported characters: lowercase letters (a to z), uppercase letters (A to Z), digits (0 to 9), decimal points (.), hyphens (-), underscores (_), forward slashes (/), equal signs (=), and question marks (?).</p>
HTTP request method	<p>HTTP request method of health checks. Valid values: GET (default) and HEAD.</p> <p>If HEAD is selected, the server will only return the HTTP header information. This reduces backend overheads and improves request efficiency. The real server must support the HEAD method.</p> <p>If GET is selected, the real server must support the GET method.</p>
HTTP version	<p>HTTP version of the real server.</p> <p>If the real server supports HTTP 1.0, then the Host field of a request does not need authentication, that is, the health check domain name does not need to be configured.</p> <p>If the real server supports HTTP 1.1, the Host field of a request needs authentication, that is, the health check domain name needs to be configured.</p> <p>Note: If you select HTTP/1.1 but do not specify the health check domain name, the real server returns the 400 error code, which indicates a health check exception. You are advised to select http_4xx for Normal status code.</p>
Normal status code	If the status code is the selected one, the real server is considered as alive (healthy). Valid values: http_1xx, http_2xx, http_3xx, http_4xx, and http_5xx.
Show advanced options	For more information, see Advanced Options .

Configuring custom protocol health check

- 1. Configure a listener to the step of **Health check** as instructed in [Prerequisites](#).
- 2. On the **Health check** tab, select **Custom** for **Check method**.

CreateListener

✓ Basic configuration

2 Health check

3 Session persistence

Health checkDetect and remove abnormal server ports automatically.

Source IP

CLB VIP

IP range starting with 100.64

Protocol

TCP

HTTP

Custom protocol

Checking port

RS port by default. Unless you want to specify a port, please leave it empty

Input format

Texts

Request

Up to 500 chars

It cannot be left empty.

Return result

Up to 500 chars

It cannot be left empty.

Show advanced options

Back

Next

Parameter	Description
Health check	You can enable or disable health check. We recommend that you enable health check for automatic c servers and removal of abnormal ports.
Health check source IP	The source IP address of health check packets. The default value is the 100.64.0.0/10 IP range. By u you can solve the container loopback problem in TKE scenarios. Existing users can choose to use a C health care source IP address.

Check method	Custom protocol health checks are conducted if Custom is selected. This is applicable to non-HTTP
Port	This parameter is optional. We recommend not specifying the port unless you need to check a specific server port will be checked if the port is not specified here.
Format	Text and hexadecimal strings are supported. If Text is selected, the text will be converted into a binary string for sending requests and comparing returned results. If Hexadecimal is selected, the hexadecimal string will be converted into a binary string for sending requests and comparing returned results.
Request	This parameter is the custom health check request content, which is required, such as <code>F13E0100000100000000000003777777047465737403636F6D0774656E63656E7403636E</code> for the DNS service health check.
Return result	When customizing the health check request, you must enter the returned health check result, such as service health check.
Show advanced options	For more information, see Advanced Options .

UDP Listener

UDP listeners support UDP health checks, which can be conducted by checking ports and running the Ping command.

Configuring UDP health check - port check

1. Configure a listener to the step of **Health check** as instructed in [Prerequisites](#).
2. On the **Health check** tab, select **Custom** for **Check method**.

CreateListener

- ✓

Basic configuration
- >
- 2

Health check
- >
- 3

Session persistence

Health check ☒

Detect and remove abnormal server ports automatically.

Source IP ⓘ ☐ CLB VIP ☒ IP range starting with 100.64

Protocol ☒ Checking port ☐ PING

Checking port

Input format

Only ASCII printable characters are allowed

Request ⓘ

Return result ⓘ

[Show advanced options](#) ▼

Back

Next

Parameter	Description
Health check	You can enable or disable health check. We recommend that you enable health check for automatic c servers and removal of abnormal ports.
Health check source IP	The source IP address of health check packets. The default value is the 100.64.0.0/10 IP range. By u you can solve the container loopback problem in TKE scenarios. Existing users can choose to use a C health care source IP address.
Check method	If Custom is selected, UDP detection packets are sent from the health check source IP address to a obtain the status of the real server.
Port	This parameter is optional. We recommend not specifying the port unless you need to check a specifi server port will be checked if the port is not specified here.
Format	Text and hexadecimal strings are supported.

	If Text is selected, the text will be converted into a binary string for sending requests and comparing r If Hexadecimal is selected, the hexadecimal string will be converted into a binary string for sending r comparing returned results.
Request	This is the custom health check request content, such as F13E01000001000000000000003777777047465737403636F6D0774656E63656E7403636E for the DNS service health check.
Return result	When customizing the health check request, you must configure the returned health check result, suc the DNS service health check.
Show advanced options	For more information, see Advanced Options .

Configuring UDP health check - Ping command

1. Configure a listener to the step of **Health check** as instructed in [Prerequisites](#).
2. On the **Health check** tab, select **PING** for **Check method**.

CreateListener

Basic configuration

2 Health check

3 Session persistence

Health check

Detect and remove abnormal server ports automatically.

Source IP ⓘ

CLB VIP

IP range starting with 100.64

Protocol

Checking port

PING

Show advanced options

Back

Next

Parameter	Description
Health check	You can enable or disable health check. We recommend that you enable health check for automatic checks on real servers and removal of abnormal ports.
Health	The source IP address of health check packets. The default value is the 100.64.0.0/10 IP range.

check source IP	By using this IP range, you can solve the container loopback problem in TKE scenarios. Existing users can choose to use a CLB VIP as the health care source IP address.
Check method	If you select PING , the IP address of the real server will be pinged to obtain the status of the real server.
Show advanced options	For more information, see Advanced Options .

TCP SSL Listener

Configuring TCP health check

1. Configure a listener to the step of **Health check** as instructed in [Prerequisites](#).
2. On the **Health check** tab, select **TCP** for **Check method**.

CreateListener

Basic configuration

2 Health check

3 Session persistence

Health check

Detect and remove abnormal server ports automatically.

Source IP ⓘ

CLB VIP

IP range starting with 100.64

Protocol

TCP

HTTP

Checking port

Real server port

Show advanced options

Back

Next

Parameter	Description
Health check	You can enable or disable health check. We recommend that you enable health check for automatic checks on real servers and removal of abnormal ports.

Health check source IP	The source IP address of health check packets. The default value is the 100.64.0.0/10 IP range. By using this IP range, you can solve the container loopback problem in TKE scenarios. Existing users can choose to use a CLB VIP as the health care source IP address.
Check method	TCP health checks are conducted if TCP is selected.
Port	The health check port and listening port of a TCP SSL listener are the same.
Show advanced options	For more information, see Advanced Options .

Configuring HTTP health check

1. Configure a listener to the step of **Health check** as instructed in [Prerequisites](#).
2. On the **Health check** tab, select **HTTP** for **Check method**.

CreateListener

✓

Basic configuration

2Health check

3Session persistence

Health check

Detect and remove abnormal server ports automatically.

Source IP ⓘ

CLB VIP

IP range starting with 100.64

Protocol

TCP

HTTP

Checking port

Real server port

Check domain

(Optional) It's recommended to set this field.

It only supports letters, digits, "-" and "."; the host field is omitted by default.

Path

/

It defaults to check the root directory of the real server. It should start with "/"; up to 80 chars; allowing letters, numbers, "_", "-", ".", "/", "=", "?".

HTTP request method ⓘ

GET

HTTP version ⓘ

HTTP/1.1

Normal status code ⓘ

✓

http_1xx

✓

http_2xx

✓

http_3xx

✓

http_4xx

http_5xx

When the status code is http_1xx, http_2xx, http_3xx, http_4xx, the back-end server is considered active

Show advanced options ▾

Back

Next

Parameter	Description
Health check	You can enable or disable health check. We recommend that you enable health check for automatic checks on real servers and removal of abnormal ports.
Health check source IP	The source IP address of health check packets. The default value is the 100.64.0.0/10 IP range. By using this IP range, you can solve the container loopback problem in TKE scenarios. Existing users can choose to use a CLB VIP as the health care source IP address.

Check method	HTTP health checks are conducted if HTTP is selected.
Port	The health check port and listening port of a TCP SSL listener are the same.
Check domain	Limits on a health check domain name: Length: 1 to 80 characters. The default value is the forwarding domain name. Regular expressions are not supported. If your forwarding domain name is a wildcard domain name, you need to specify a fixed (non-regular) domain name as the health check domain name. Supported characters: lowercase letters (a to z), digits (0 to 9), decimal points (.), and hyphens (-).
Path	Limits on a health check path: Length: 1 to 200 characters. A health check path must start with /. The default value is /. Regular expressions are not supported. You are advised to specify a fixed URL (static webpage) for health checks. Supported characters: lowercase letters (a to z), uppercase letters (A to Z), digits (0 to 9), decimal points (.), hyphens (-), underscores (_), forward slashes (/), equal signs (=), and question marks (?).
HTTP request method	HTTP request method of health checks. Valid values: GET (default) and HEAD. If HEAD is selected, the server will only return the HTTP header information. This reduces backend overheads and improves request efficiency. The real server must support the HEAD method. If GET is selected, the real server must support the GET method.
HTTP version	HTTP version of the real server. Only HTTP 1.1 is supported. The real server needs to verify the Host field of the request, that is, the check domain name needs to be configured. Note: If you do not specify the check domain name, the real server returns the 400 error code, which indicates a health check exception. We recommend that you select http_4xx for Normal status code .
Normal status code	If the status code is the selected one, the real server is considered as alive (healthy). Valid values: http_1xx, http_2xx, http_3xx, http_4xx, and http_5xx.
Show advanced options	For more information, see Advanced Options .

HTTP Listener

Configuring HTTP health check

1. Configure a listener to the step of **Health check** as instructed in [Prerequisites](#).

CreateForwarding rule

Basic configuration

2Health check

3Session persistence

Health check

Detect and remove abnormal server ports automatically.

Source IP

CLB VIP

IP range starting with 100.64

Protocol

TCP

HTTP

Check domain

It defaults to the forwarding d

Path

Root directory of CVM

/

Hide advanced options

Response timeout

2 Seconds

60 Seconds

2

Seconds

Check interval

2 Seconds

300 Seconds

5

Seconds

Unhealthy threshold

2 Times

10 Times

3

Times

Healthy threshold

2 Times

10 Times

3

Times

HTTP request method

GET

HTTP status code detection

http_1xx

http_2xx

http_3xx

http_4xx

http_5xx

When the status code is http_1xx, http_2xx, http_3xx, http_4xx, the back-end server is considered active

Back

Next

Parameter	Description
Health	You can enable or disable health check. We recommend that you enable health check for

check	automatic checks on real servers and removal of abnormal ports.
Health check source IP	The source IP address of health check packets. The default value is the 100.64.0.0/10 IP range. By using this IP range, you can solve the container loopback problem in TKE scenarios. Existing users can choose to use a CLB VIP as the health care source IP address.
Check domain	Limits on a health check domain name: Length: 1 to 80 characters. The default value is the forwarding domain name. Regular expressions are not supported. If your forwarding domain name is a wildcard domain name, you need to specify a fixed (non-regular) domain name as the health check domain name. Supported characters: lowercase letters (a to z), digits (0 to 9), decimal points (.), and hyphens (-).
Path	The health check path can be set to the root directory of the real server or a specified URL. Limits on a health check path are as follows: Length: 1 to 200 characters. A health check path must start with /. The default value is /. Regular expressions are not supported. You are advised to specify a fixed URL (static webpage) for health checks. Supported characters: lowercase letters (a to z), uppercase letters (A to Z), digits (0 to 9), decimal points (.), hyphens (-), underscores (_), forward slashes (/), equal signs (=), and question marks (?).
Response timeout	Maximum response timeout period for a health check. If a real server fails to respond within the timeout period, the real server is considered as abnormal. Value range: 2-60 seconds.
Check interval	Interval between two health checks. Value range: 2-300 seconds.
Unhealthy threshold	If a real server has failed the health check for n (a customizable value) consecutive times, the real server is considered unhealthy, and Abnormal is displayed in the console. Value range of n: 2-10.
Healthy threshold	If a real server has passed the health check for n (a customizable value) consecutive times, the real server is considered healthy, and Healthy is displayed in the console. Value range of n: 2-10.
HTTP request method	HTTP request method of health checks. Valid values: GET (default) and HEAD. If HEAD is selected, the server will only return the HTTP header information. This reduces backend overheads and improves request efficiency. The real server must support the HEAD method. If GET is selected, the real server must support the GET method.
Normal status	If the status code is the selected one, the real server is considered as alive (healthy). Valid values: http_1xx, http_2xx, http_3xx, http_4xx, and http_5xx.

code

Configuring TCP health check

- 1. Configure a listener to the step of **Health check** as instructed in [Prerequisites](#).
- 2. On the **Health check** tab, select **TCP** for **Check method**.

CreateListener

Basic configuration

2Health check

3Session persistence

Health check

Detect and remove abnormal server ports automatically.

Source IP ⓘ

CLB VIP

IP range starting with 100.64

Protocol

TCP

HTTP

Custom protocol

Checking port

RS port by default. Unless you want to specify a port, please leave it empty.

Hide advanced options ▲

Response timeout

2 Seconds

60 Seconds

–

2

+

Seconds

Check interval

2 Seconds

300 Seconds

–

5

+

Seconds

Unhealthy threshold ⓘ

2 Times

10 Times

–

3

+

Times

Healthy threshold ⓘ

2 Times

10 Times

–

3

+

Times

Back

Next

Parameter	Description
Health check	You can enable or disable health check. We recommend that you enable health check for automatic checks on real servers and removal of abnormal ports.
Health check source IP	The source IP address of health check packets. The default value is the 100.64.0.0/10 IP range. By using this IP range, you can solve the container

	loopback problem in TKE scenarios. Existing users can choose to use a CLB VIP as the health care source IP address.
Check method	TCP health checks are conducted if TCP is selected.
Show advanced options	For more information, see Advanced Options .

HTTPS Listener

Note:

If HTTP is selected as the backend protocol of the HTTPS listener's forwarding rules, HTTP health checks will be conducted; if HTTPS is selected, HTTPS health checks will be conducted.

For the health check configuration of HTTPS listeners, see [HTTP Listener](#).

Advanced Options

Option	Description	Default Value
Response timeout	Maximum response timeout period for a health check. If a real server fails to respond within the timeout period, the real server is considered as abnormal. Value range: 2-60 seconds.	2 seconds
Check interval	Interval between two health checks. Value range: 2-300 seconds.	5 seconds
Unhealthy threshold	If a real server has failed the health check for n (a customizable value) consecutive times, the real server is considered unhealthy, and Abnormal is displayed in the console. Value range of n: 2-10.	3 times
Healthy threshold	If a real server has passed the health check for n (a customizable value) consecutive times, the real server is considered healthy, and Healthy is displayed in the console. Value range of n: 2-10.	3 times

References

[Health Check Overview](#)

Configuring Alarm Policy

Setting 100.64.0.0/10 IP Range as the Health Check IP

Last updated : 2024-01-04 14:34:05

This document takes a TCP listener as an example to describe how to change the health check source IP address of a CLB instance from the CLB [VIP](#) to the `100.64.0.0/10` IP range.

Use Cases

1. Aggregating real server security groups

The health check source IP is aggregated into the `100.64.0.0/10` IP range.

2. Solving the problem of private network loopback in self-built Kubernetes cluster

The K8s service needs to be exposed both inside and outside the cluster. The former is implemented through the cluster's internal load balancing (IPVS), and the latter is implemented through private network CLB. IPVS will bind the IP address of the private network CLB instance to a local interface, so that access to the instance address in the cluster is actually to use the IPVS load balancing in the cluster.

In the TKE service, the private network CLB uses the CLB VIP as the health check source IP, which conflicts with the address bound to the IPVS in the native K8s implementation, resulting in the failure of private network CLB health check.

Setting the health check source IP to the `100.64.0.0/10` IP range can avoid address conflicts and solve the problem of health check failures.

Troubleshooting the Issue

1. Log in to the [CLB console](#).
2. Select your region in the top-left corner of the **Instance management** page, find the target instance in the instance list, and click **Configure listener** in the **Operation** column.
3. On the **Listener management** tab, find the target listener, and click the



icon on the right to edit the listener.

4. In the **Edit listener** pop-up window, click **Next** to go to the **Health check** tab.
5. On the **Health check** tab, select `100.64.0.0/10` **IP range** as the health check source IP address, click **Next**, and click **Submit**.

CreateListener

Basic configuration

2 Health check

3 Session persistence

Health check

☒ ☐

Detect and remove abnormal server ports automatically.

Source IP ⓘ

☐ CLB VIP ☒ IP range starting with 100.64

Protocol

☒ TCP ☐ HTTP ☐ Custom protocol

Checking port

RS port by default. Unless you want to specify a port, please leave it em

Show advanced options ▼

Back

Next

FAQs

What are the advantages of using the 100.64.0.0/10 IP range as the health check source IP address?

For CLB instances whose health check source IP address falls into the 100.64.0.0/10 IP range, you do not need to add this IP range to the allowlist of the security group of the associated real servers. If the real servers are configured with other security policies (such as iptables), this IP range must be added to the allowlist. Otherwise, health check failures may be caused.

The security policy for real servers is aggregated to the 100.64.0.0/10 IP range.

This IP range can prevent IP conflicts because it is a private IP range of Tencent Cloud and will not be allocated to users.

Will a fixed IP address be used when I select the 100.64.0.0/10 IP range as the health source IP address?

No. An IP address in the 100.64.0.0/10 IP range, instead of a fixed IP address, is used as the health check source IP address.

References

[Configuring Health Check](#)

[Health Check Identifiers](#)

Verifying Health Check Source IPs

Last updated : 2024-01-04 14:34:05

This document describes how to check whether the health check source IPs of a CLB are within the 100.64.0.0/10 IP range. You can also switch the health check source IPs from the CLB VIP to the IP range 100.64.0.0/10 directly.

Use cases

Check the health check source IPs of a CLB. If CLB VIPs are used as health check source IPs, switch them to the IP range 100.64.0.0/10 directly.

Note:

It is recommended to check one or two CLB instances as a test first.

Directions

1. Log in to the [CLB console](#).
2. Select **Tools > Verifying health check source IPs** in the left sidebar.
3. Select a region and click **Check now**.
4. On the **Verify health check source IPs** page, select the target instance and click **Check now**.
5. After the diagnosis is completed, the diagnosis result is displayed.

If the target CLB instances do not use CLB VIPs as the health check source IPs, the following figure is shown.
If the target CLB instance uses CLB VIPs as the health check source IPs, the following figure is shown. Click **Switch** to switch the health check source IPs from the CLB VIPs to the IP range 100.64.0.0/10.

Other operations

Report status

Status	Description
Normal - Don't need to switch	The CLB instances are not using CLB VIPs as the health check source IPs.
Normal - Switched	The CLB instances used CLB VIPs as the health check source IPs, and the VIPs have been switched to the 100.64.0.0/10 IP range.
Alert - Not	The CLB instances are using CLB VIPs as the health check source IPs. Please switch IPs

switched	in time.
----------	----------

Viewing diagnostic reports

1. Log in to the [CLB console](#).
2. Select **Tools > Verifying health check source IPs** in the left sidebar.
3. Select a region at the top of the page to list all diagnostic reports of the corresponding region.
4. Under **Operations**, click **View report** to check details of history reports.

Deleting reports

1. Log in to the [CLB console](#).
2. Select **Tools > Verifying health check source IPs** in the left sidebar.
3. Select a region at the top of the page to list all diagnostic reports of the corresponding region.
4. Under **Operations**, click **Delete**, and confirm the deletion.

See also

[Configuring Health Checks](#)

[Health Check Identifiers](#)

[Adding 100.64.0.0/10 for Health Check Source IPs](#)

Certificate Management

Managing Certificates

Last updated : 2024-01-04 14:34:05

When configuring an HTTPS listener of a CLB instance, you can directly use a certificate in SSL Certificate Service or upload the third-party server certificate and [SSL certificate](#) that you require to the CLB console.

Certificate Requirements

CLB supports only certificates in PEM format. Before uploading a certificate, make sure that your certificate, certificate chain, and private key meet the format requirement. For information about the certificate requirements, see [Certificate Requirements and Certificate Format Conversion](#).

Certificate Encryption Algorithms

CLB supports the following algorithms for certificate encryption: ECC and RSA. For more information about the algorithms, see [What are the differences between RSA and ECC?](#).

Note:

You can configure two certificates that use different algorithms in SSL parsing for HTTPS listeners. For more information, see [Configuring an HTTPS Listener](#).

Listener Type	Supported Encryption Algorithm When Configuring One Certificate	Supported Encryption Algorithms When Configuring Two Certificates
HTTPS	RSA or ECC	RSA and ECC
TCP_SSL, QUIC	RSA or ECC	Does not support configuring two certificates that use different encryption algorithms.
TCP, UDP, HTTP	Does not support configuring certificates.	Does not support configuring certificates.

Configuring Certificates

There are two types of certificate configuration for an HTTPS listener:

Listener-level certificate configuration: If SNI is not enabled, the same certificate is configured for all domain names under the listener. For more information, see [Configuring an HTTPS Listener](#).

Domain name-level certificate configuration: If SNI is enabled, different certificates can be configured for different domain names under the listener. For more information, see [SNI Support for Binding Multiple Certificates to a CLB Instance](#).

Updating Certificates

To prevent certificate expiration from affecting your service, please update your certificate before it expires.

Note:

After a certificate is updated, the system does not delete the legacy certificate but generates a new one. The certificate will be automatically updated for all CLB instances that use it.

1. Log in to the [CLB console](#).
2. Click **Certificate management** in the left sidebar.
3. In the certificate list, click **Update** in the **Operation** column of the target certificate.
4. In the pop-up window, enter the content and key of the new certificate and click **Submit**.

Create a new certificate

Certificate Name

cert

Cannot exceed 80 characters, only English letters, numbers, underscores, and hyphens are supported "-", ".", ".".

Certificate Type

☒ Server Certificate ☐ Client CA Certificate

Certificate Content

-----BEGIN CERTIFICATE-----
[Blurred content]
-----END CERTIFICATE-----
[View Examples](#)

Key Content

-----BEGIN RSA PRIVATE KEY-----
[Blurred content]
-----END RSA PRIVATE KEY-----
[View Examples](#)

Submit

Close

Viewing CLB Instances Associated with a Certificate

1. Log in to the [CLB console](#).
2. Click **Certificate management** in the left sidebar.
3. In the certificate list, click the ID of the target certificate.
4. On the **Basic information** page, view the CLB instances associated with the certificate.

Basic Info

Namemanuel-test

IDha2qQzkD

Certificate TypeServer Certificate

Certificate Content

-----BEGIN CERTIFICATE-----

[Blurred certificate content]

Copy

Load Balancer Bound

[Blurred content]

Primary Domain Name

Alternate Domain-

Upload Time2020-10-29 12:06:20

Start Time2020-07-03 18:05:58

Expiry Time2021-07-03 18:05:58

Certificate Requirements and Certificate Format Conversion

Last updated : 2024-01-04 14:34:05

This document introduces the requirements on SSL certificates and describes how to convert certificate formats.

Certificate Application Process

1. Use the OpenSSL to generate a private key file locally, i.e., `privateKey.pem` . Please keep it private.

```
openssl genrsa -out privateKey.pem 2048
```

2. Use the OpenSSL to generate a certificate request file, i.e., `server.csr` . It can be used for certificate application.

```
openssl req -new -key privateKey.pem -out server.csr
```

3. Obtain the content of the certificate request file and visit CA sites to apply for a certificate.

Certificate Format Requirements

The certificate that needs to be applied for should be in PEM format on Linux. CLB does not support certificates in other formats. For more information, see [Converting Certificates to PEM format](#).

If your certificate is issued by a root CA, the certificate is unique, and the configured website will be considered trustworthy by browsers and other accessing devices with no additional certificates required.

If your certificate is issued by an intermediate CA, your certificate file will consist of multiple certificates. In this case, you need to manually combine the server certificate and intermediate certificate for upload.

If your certificate has a certificate chain, please convert it to PEM format and merge with the certificate content for upload.

The concatenation rule is as follows: put the server certificate before the intermediate certificate with no blank lines in between.

Note:

You can check for applicable rules or instructions provided by the CA when issuing the certificate.

Certificate format and certificate chain format

Below are examples of certificate and certificate chain formats. Please confirm the format before upload:

1. Certificate issued by a root CA: PEM format on Linux, as shown below:

```

-----BEGIN CERTIFICATE-----
MIIE+TCCA+GgAwIBAgIQU306HIX4KsioTW1s2A2krTANBgkqhkiG9w0BAQUFADCB
tTELMAkGA1UEBhMCVVMxZzAVBgNVBAsTD1Zlcm1TdWduLmNvbmMuMR8wHQYDVQQL
ExZWZlZjU2LmN1bGU2Z3RvbjEQAQA4GA1UEBxQHU2VhdHRsZTEYMBYGA1UEChQPQW1hem9uLmNv
bSBzBjbmMuMRowGAYDVQQDFBFBYw0uYW1hem9uYXNzLmNvbTCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwGyKCGYEA3Xb0EGea2d88QGEUwLcEppwGawEkUdLZmGL1rQJZdeeN
3vaF+ZTm8QwSAdk2Gr/RwYXtpx04xvQXmNm+9YmksHmCZdruCrW1eN/P9wbFqMMZ
X964CjVov3NrF5AuxU8jgtw0yu//C3hWn0uIVGdg76626gg0oJSaj48R2n0MnVcC
AwEAAaOCAdEwggHnMAkGA1UdEwQCAAwCwYDVR0PBAQDAgWgMEUGA1UdHwQ+MDww
OqA4oDaGNGh0dHA6Ly9TVlJTZW1cmUtdRzItY3JsLnZlcm1zaWduLmNvbS9TVlJT
ZW1cmVHMi5jcmwwRAYDVVR0gBD0wOzA5BgtghkgBhvFAQcXAZAqMCgGCCsGAQUF
BwIBFhxodHRwczovL3d3dy52ZXJpc2lnbi5jb20vcnBhMB0GA1UdJQQWMBQGCCsG
AQUFBwMBBggrBgEFBQcDAjAFBgNVHSMEGDAWgBS17wsRzsBBA6NKZZBIshzgVy19
RzB2BggrBgEFBQcBAQRqMGgwJAYIKwYBBQUHMAAGGh0dHA6Ly9vY3NwLnZlcm1za
WduLmNvbTBABggrBgEFBQcQAoY0aHR0cDovL1NWU1N1Y3VyZS1HMi1haWEudmVy
aXNpZ2Z4U29tL1NWU1N1Y3VyZUcyLmN1c2lnbi5jb20vcnBhMB0GA1UdJQYJKoZI
WDBWFglpbWFnZS9naWYwITAFMACGBSs0AwIaBBRLa7kolgYMu9BS0JsprEsHieF
GDAmFiRodHRwOi8vbG9nb352ZXJpc2lnbi5jb20vcnBhMB0GA1UdJQYJKoZI
hvcNAQEFBQADggEBALpFBXeG782QsTtGwEE9zBcVCuKjrsL3dWK1dFiq30P4y/Bi
ZBYEyw8t8zNuYFUE25Ub/zmvmp7p0G76tmQ8bRp/4qkJoISesHJvFgJ1mksr3IQ
3gaE1a2BSUIHxGLn9N4F09hYwweEzaCxfGbiLdEiodNwzcvGJ+2LLDWGJOGnNI
NM856xjqhJCPxYzk9buuCl1B4Kzu0CTbexz/iEgYV+DiuTxcFA4uhwMDSe0nynbn
1qiWk450mCOnqH4ly4P41Xo02t4A/DI1I8ZNct/QfL69a2Lf6vc9rF7BELT0e5Y
R7CKx7Fc5xRaeQdyGj/dJevm9BF/mSdncLS5vas=
-----END CERTIFICATE-----

```

Certificate rules are:

Your certificate should start with "-----BEGIN CERTIFICATE-----" and end with "-----END CERTIFICATE-----", which should be uploaded together.

Each line should contain 64 characters, while the last line can contain less than 64 characters.

2. Certificate chain from an intermediate CA:

```

-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----

```

Certificate chain rules:

No blank lines between certificates.

All certificates should meet the requirements as above.

RSA Private Key Format Requirements

Below is an example:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAzSSSCHH67bmT8mFykAxQ1tKCYukwBiWZwkOStFEbTWHy8K
tTHSFD1u9TL6qycrHEG7cjYD4DK+kVIHU/Of/pUWj9LLnrE3W34DaVzQdKA00I3A
Xw9SgrqFJMjclva2khNKA1+tNPSCPJoo9DDrP7wx7cQx7LbMb0dfZ8858KIoluzJ
/fD0XXyuWoqaIePZtK9Qnjn957ZEPHjtUpVZuhS3409DDM/tJ3Tl8aaNYWHRPBc0
jNcz0Z6XQGf1rZG/Ve520GX6rb5dUYpdcfXzN5NM6xYg8a1L7UHDHHPi4AYsatdG
z5TMPnmEf8yZPUYudTLxgMVAovJr09Dq+5Dm3QIDAQABAoIBAGL68Z/nnFyRHRFi
laF6+Wen8ZvNqkm0hAMQwIjh1Vp1fL74//8Qyea/EvUtuJHyB6T/2PZQoNVhxe35
cgQ93Tx424WGPcwUshSfxewFbAYGf3ur8W0xq0uU07BAxaKHnCMNG7dGyo1UowRu
S+yXLRpVzH1YkuH8TT53udd6TeTWi77r8dkGi9KSAZ0pRa19B7t+CHKIzm6ybs/2
06W/zH24YAxwkTYLKGHjoieYs111ahIAJvICVgTc3+LzG2pIpM7I+K0nHCSeswvM
i5x9h/OT/ujZsyX9P0PaAyE2bqy0t080tGexM076Ssv0KVhKFvWjLUnhf6WcqFCD
xqhnxkECgYEA+PftNb6eyXl+/Y/U8NM2fg3+rSCms0j9Bg+9+yZzF5GhgqHu0edU
ZXIHrJ9u6B1XE1arpijVs/WHmFhYSTm6DbdD7S1tLy0BY4cPTRhziFTKt8AkIXMK
605u0UiWsq0Z8hn1X141ox2cW9ZQa/HC9udeyQotP4NsMJWgpBV7tC0CgYEAwvNf
0f+/jUjt0HoyxCh4SIAqk4U0o4+hBCQbWcXv5qCz4mRyTaWzFEG8/AR3Md2rhmiZi
GnJ5dfde7uY+JsQXf2Q5JjwTadlBW4led0Sa/uKRa04UzVgnYp2aJKxtuWfFvVbU
+kf728ZJRA6azSLvGmA8hu/GL6bgfU3fkSkw03ECgYBpYK7TT7JvvnAERmtJf2yS
ICRkQaB3gPSe/lCgzy1nhtaFOUbNxeuowLAZR0wrz7X3TZqHEDcYoJ7mK346of
QhGLITyoeHkbYkAUtq038Y04EKH6S/IzMzB0frXiPKg9s8UKQzku+GSE7ootli+a
R8Xzu835EwxI6BwNN1abpQKBgQC8TialClq1FteXQyGcNdcReLMncUHKIKcP/+xn
R3kVl06MZCfAdqirAjiQwApkh9Bxbp2eHCrB8lMFAWLRS1ok79b/jVmtZMC3upd
EJ/iSWjZKPbw7hCFAeRtPhxyNTJ5idEiu9U8EQid811giPgn0p3sE0HpDI89qZX
aaiMEQKBgQDK2bsnZE9y0ZWGTeu94vziKmFrSkJMGH8pLaTiliw1iRhRYWJysZ9
B0IDxnrmwiPa9bCtEpK80zq28dq7qxpCs9CavQRcv0Bh5Hx0yy23m9hFRzfDeQ7z
NTKh193HHF1joNM81LHFyGRFEWWrroW5gfbudR6USRnR/6iQ11xZXw==
-----END RSA PRIVATE KEY-----
```

RSA private key can include all private keys (RSA and DSA), public keys (RSA and DSA), and (X.509) certificates. It stores data in Base64-encoded DER format and is wrapped by ASCII headers, making it suitable for transmission in text mode between systems.

RSA private key rules:

Your certificate should start with "-----BEGIN RSA PRIVATE KEY-----" and end with "-----END RSA PRIVATE KEY-----", which should be uploaded together.

Each line should contain 64 characters, while the last line can contain less than 64 characters.

If your private key does not start with "-----BEGIN PRIVATE KEY-----" and end with "-----END PRIVATE KEY-----", you can convert it in the following way:

```
openssl rsa -in old_server_key.pem -out new_server_key.pem
```

You can then upload `new_server_key.pem` content together with the certificate.

Converting Certificates to PEM Format

Currently, CLB only supports certificates in PEM format. Certificates in other formats need to be converted to PEM format first before uploading to CLB. We recommend you use OpenSSL. The following shows how to convert several common formats to PEM.

DER to PEM

P7B to PEM

PFX to PEM

CER/CRT to PEM

DER format is generally used on Java platforms.

Certificate conversion:

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

Private key conversion:

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

P7B format is generally used on Windows Server and Tomcat.

Certificate conversion:

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

You need to get the content between "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----" in `outcertificat.cer` to upload as certificate.

Private key conversion: private keys can generally be exported on IIS servers.

PFX format is generally used on Windows Server.

Certificate conversion:

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

Private key conversion:

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

You can convert certificates in CER/CRT formats to PEM by directly modifying their file extension names. For example, you can directly rename the `servertest.crt` certificate file as `servertest.pem`.

# SSL One-way Authentication and Mutual Authentication

Last updated : 2024-01-04 14:35:48

Secure Sockets Layer (SSL) is a security protocol designed to ensure security and data integrity for Internet communications. This document introduces SSL one-way authentication and mutual authentication.

## Note:

When creating a TCP SSL listener or an HTTPS listener for a CLB instance, you can select one-way authentication or mutual authentication as the SSL parsing method. For more information, please see [Configuring a TCP SSL Listener](#) and [Configuring an HTTPS Listener](#).

## Differences Between SSL One-way Authentication and Mutual Authentication

For [SSL one-way authentication](#), certificates are only required on the server but not the client; while for [SSL mutual authentication](#), certificates are required both on the server and the client.

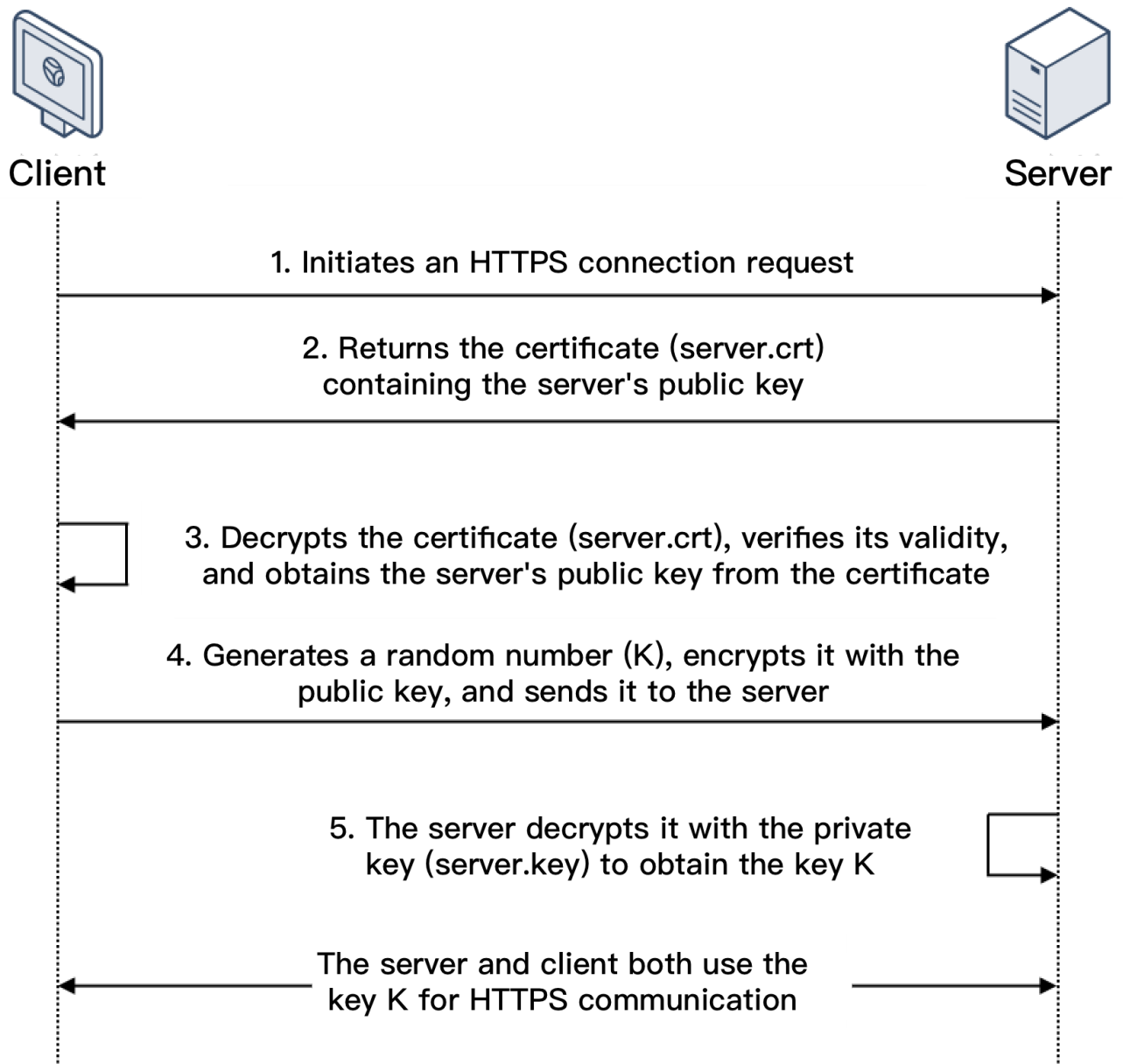
Compared to SSL mutual authentication, the one-way authentication does not involve the client certificate verification and encryption scheme negotiation on the server. Although the encryption scheme the server sent to the client is not encrypted, the security of SSL authentication is not compromised.

Web applications generally have a large number of users and user identity verification is not necessary on the communication layer, for which the SSL one-way authentication can be used. However, identity verification may be required for clients connecting to financial applications, SSL mutual authentication should be used.

## SSL One-way Authentication

In SSL one-way authentication, only the server identity but not the client identity needs to be verified. The process of SSL one-way authentication is as shown below:





1. A client initiates an HTTPS connection request to the server together with the supported SSL protocol versions, encryption algorithms, generated random numbers, and other information.

2. The server returns an SSL protocol version, encryption algorithm, generated random number, server certificate (server.crt), and other information to the client.

3. The client verifies the validity of the certificate (server.crt) for the factors below and obtains the server's public key from the certificate.

Whether the certificate is expired.

Whether the certificate is revoked.

Whether the certificate is trusted.

Whether the domain name requested is the same as the domain name in the certificate received.

4. After the certificate is verified, the client will generate a random number (the key K; which is used as the symmetric encryption key for the communication), encrypt it with the public key obtained from the server certificate, and then

send it to the server.

5. After receiving the encrypted information, the server will use its private key (server.key) to decrypt it to obtain the symmetric encryption key (the key K).

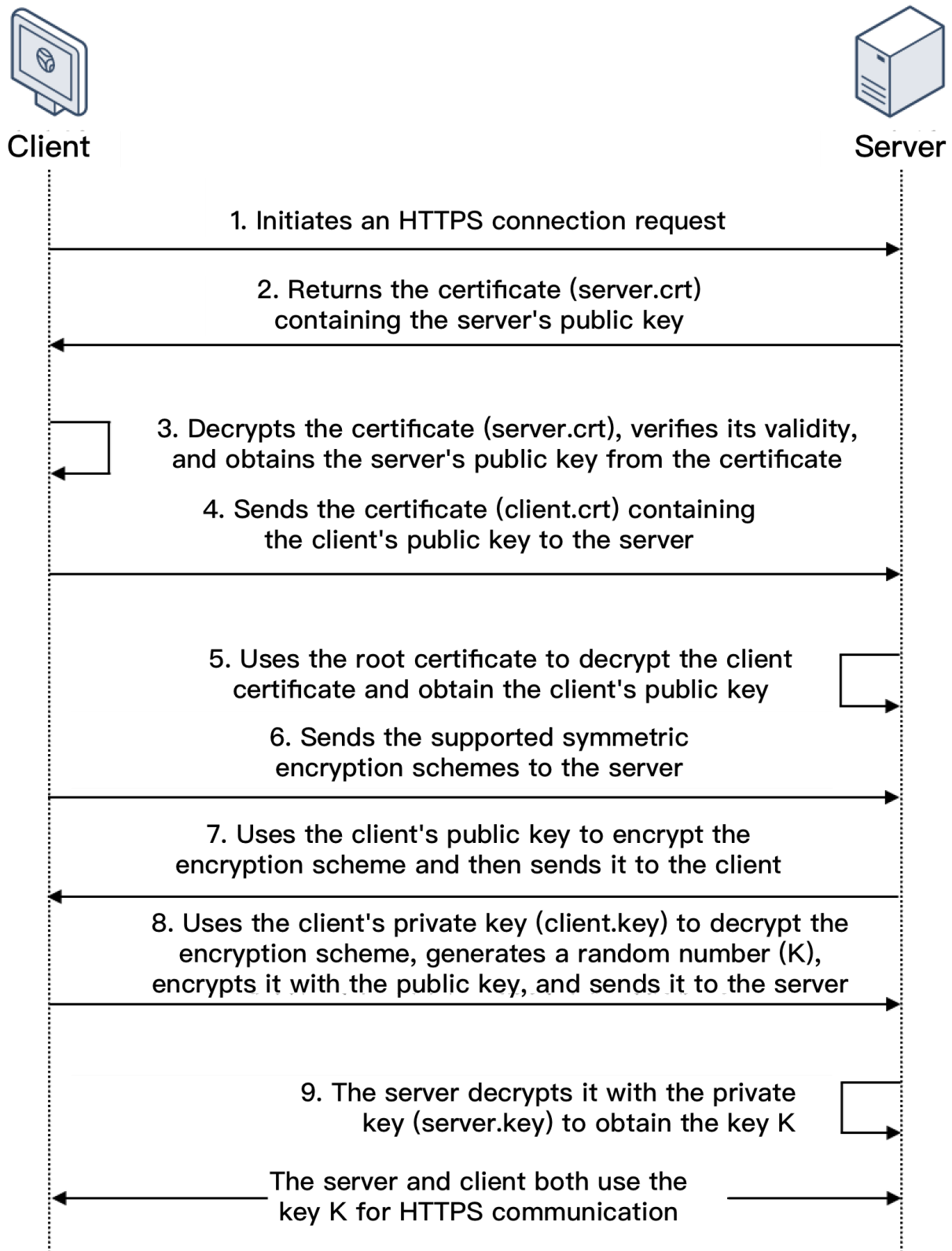
The sy

mmetric encry

ption key (the key K) will be used by the server and client for communication to guarantee information security.

## SSL Mutual Authentication

In SSL mutual authentication, the server identity and the client identity both need to be verified. The process of SSL mutual authentication is as shown below:



1. A client initiates an HTTPS connection request to the server together with the supported SSL protocol versions, encryption algorithms, generated random numbers, and other information.

2. The server returns an SSL protocol version, encryption algorithm, generated random number, server certificate (server.crt), and other information to the client.



3. The client verifies the validity of the certificate (server.crt) for the factors below and obtains the server's public key from the certificate.

Whether the certificate is expired.

Whether the certificate is revoked.

Whether the certificate is trusted.

Whether the domain name requested is the same as the domain name in the certificate received.

4. The server requires the client to send the client certificate (client.crt), and the client does so as required.

5. The server verifies the client certificate (client.crt). After it is verified, the server will use the root certificate to decrypt the client certificate and obtain the client's public key.

6. The client sends the supported symmetric encryption schemes to the server.

7. The server selects the encryption scheme with the highest encryption level from the schemes sent from the client, uses the client's public key to encrypt it, and returns it to the client.

8. The client uses its private key (client.key) to decrypt the encryption scheme and generate a random number (the key K; which is used as the symmetric encryption key for the communication), encrypts it with the public key obtained from the server certificate, and then sends it to the server.

9. After receiving the encrypted information, the server will use its private key (server.key) to decrypt it to obtain the symmetric encryption key (the key K).

The symmetric encryption key (the key K) will be used by the server and client for communication to guarantee information security.

## Relevant Document

[Certificate Requirements and Certificate Format Conversion](#)

# Log Management

## Access Log Overview

Last updated : 2024-01-04 14:34:05

CLB supports configuring access logs to collect and record the details of each client request, such as the request time, request path, client IP and port, return code, and response time. This feature can help you better understand client requests, troubleshoot issues, and analyze user behaviors.

**Note:**

Only Layer-7 CLB supports configuring access logs.

This feature is only available in regions listed below.

## Storage Methods

CLB access logs can be stored in [Cloud Log Service \(CLS\)](#): CLS is a one-stop log service platform that provides a variety of log services including log collection, storage, search, analysis, real-time export, and shipping. It assists you in implementing business operations, security monitoring, log audit, and log analysis.

Item	Storing Access Logs in CLS
Time granularity for log obtainment	Minute
Online search	Supported
Search syntax	Full-text search, key-value search, fuzzy keyword search, etc. For more information, please see <a href="#">Legacy CLS Search Syntax</a> .
Supported regions	For more details on CLS available regions, see <a href="#">Available Regions</a> .
Supported CLB type	Public network/private network CLB
Upstream and downstream links	CLS logs can be shipped to COS, and exported to CKafka for further processing.
Log retention	Tencent Cloud does not store access logs by default. The storage feature can be configured as needed.

## Relevant Operations

[Storing Access Logs in CLS](#)

# Viewing Operation Logs

Last updated : 2024-01-04 14:34:05

You can query and download the operation history of CLB in the [CloudAudit console](#).

[CloudAudit](#) enables you to perform supervision, compliance check, operational review, and risk review for your Tencent Cloud account. It provides event history of your Tencent Cloud account activities, including operations performed through Tencent Cloud Console, APIs, command line tools, and other Tencent Cloud services, which simplifies security analysis, resource change tracking, and troubleshooting.

## Directions

1. Log in to the [CloudAudit console](#).
2. Click **Operation Record** on the left sidebar to enter the **Operation Record** page. You can also log in to the [CLB Console](#) and click **CloudAudit** in the top-right corner.
3. On the operation history page, query the operations by username, resource type, resource name, event source, event ID, etc. By default, only partial data will be displayed, and you can click **View More** at the bottom of the page to get more results.

EventName

CreateListener

Nearly 7 days

2020-02-09 00:00:00 ~ 2020-03-09 23:59:59

Event time	User name	Event name	Resource type	Resource name
▶ 2020-02-27 11:51:28		CreateListener	clb	clb/
▶ 2020-02-11 20:28:03		CreateListener	clb	clb/

4. Click

▶ on the left of an operation to view its details such as access key, error code, and event ID. To view the details of an event, click **View Event**.

Event time	User name	Event name	Resource type	Resource name
▼ 2020-02-27 11:51:28	roleUser	CreateListener	clb	
access key		CAM Error Code	0	
Event ID	f	Event Region	ap-guangzhou	
Event name	CreateListener	Event source	c	
Event time	2020-02-27 11:51:28	Request ID		
Source IP address		User name		
Resource Region	gz			
<a href="#">View event</a>				

# Configuring Access Logs

Last updated : 2024-01-04 14:34:05

CLB supports configuring layer-7 (HTTP/HTTPS) access logs that can help you better understand client requests, troubleshoot issues, and analyze user behaviors. Currently, access logs can be stored in CLS, reported at a minute granularity, and searched online by multiple rules.

Access logs of CLB are mainly used to quickly locate and troubleshoot issues. The access logging feature includes log reporting, storage, and search:

Log reporting: provides best-effort services. In other words, service forwarding has a higher priority than log reporting.

Log storage and query: SLA is guaranteed based on the storage service currently in use.

## Note:

Currently, access logs can be stored in CLS only for layer-7 protocols (HTTP/HTTPS) but not layer-4 protocols (TCP/UDP/TCP SSL).

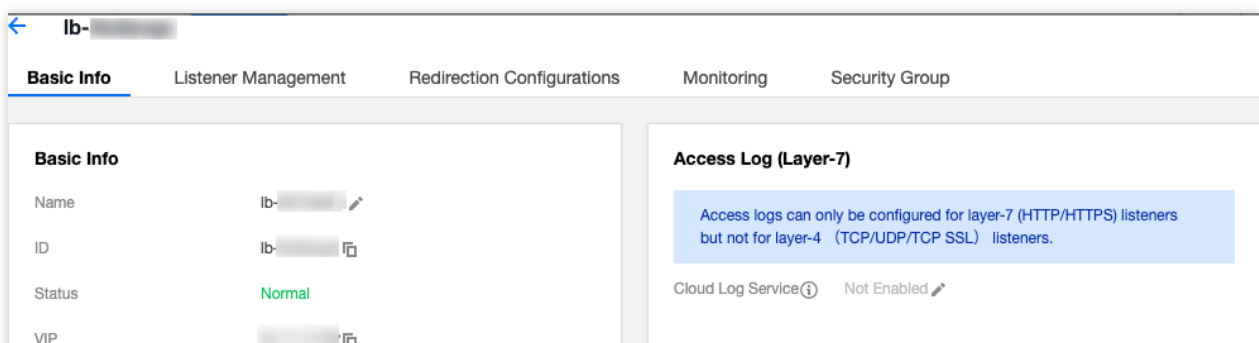
Storing CLB access logs to CLS is now free of charge. You only need to pay for the CLS service.

This feature is supported only in certain regions as displayed in the console.

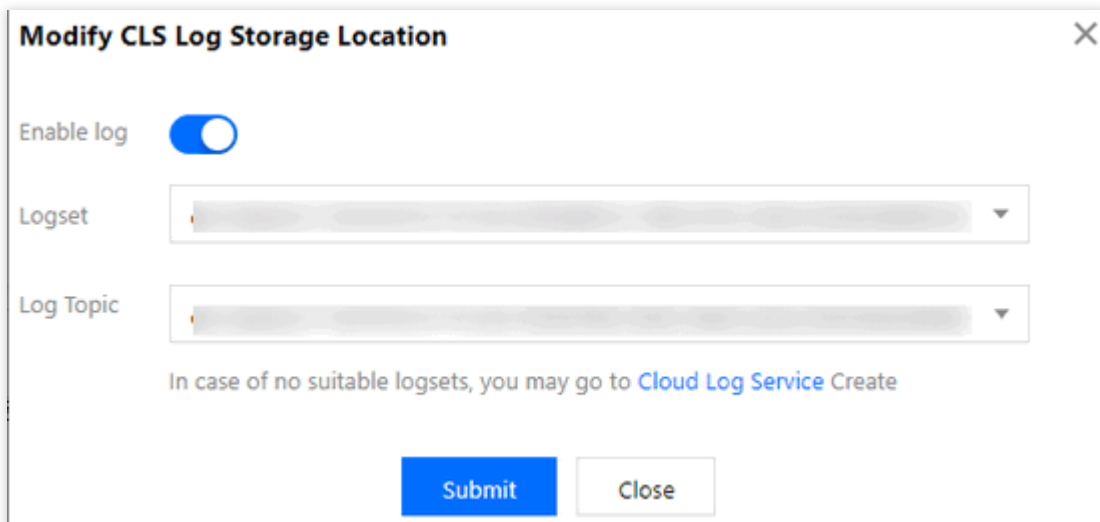
## Method 1: Single-Instance Access Logging

### Step 1. Enable access log storage in CLS

1. Log in to the [CLB console](#), and click **Instance management** in the left sidebar.
2. On the **Instance management** page, click the ID of the target CLB instance.
3. Click the pencil icon in the **Access Log (Layer-7)** panel on the **Basic Info** tab.



4. In the pop-up **Modify CLS Log Storage Location** window, enable logging and select the destination logset and log topic for access log storage, and then click **Submit**. If you have not created a logset or log topic, [create one](#) and then select it as the storage location.

**Note:**

We recommend that you use a log topic marked with **CLB** in the clb\_logset logset. The differences between a log topic marked with **CLB** and a common log topic are as follows:

CLB log topics can automatically create an index, while a common log topic requires manual index creation.

A dashboard is provided for CLB log topics by default, but needs to be manually configured for a common log topic.

5. Click the logset or log topic to go to the **Search Analysis** page in the CLS console.

6. (Optional) To disable access logging, click the pencil icon. In the **Modify CLS Log Storage Location** window, disable it and click **Submit**.

## Step 2. Configure log topic indexes

**Note:**

If access logging is configured for a single instance, you must configure the index for the log topic. Otherwise, no logs can be found.

The recommended indexes are as follows:

Key-value Index	Field Type	Delimiter
server_addr	text	Not required
server_name	text	Not required
http_host	text	Not required
status	long	-
vip_vpcid	long	-

The steps are as follows:

1. Log in to the [CLS console](#), and click **Log Topic** in the left sidebar.
2. On the **Log Topic** page, click the ID of the target log topic.

3. On the log topic details page, click the **Index Configuration** tab, and click **Edit** in the top-right corner to add indexes. For more information about index configuration, see [Configuring Index](#).

Basic Info   Collection Configuration   **Index Configuration**   Shipping Configuration   Kafka Consumption

1. The modified index configuration is only effective for newly written data, and have no impact on the index of the existed data.  
2. Delimiter cannot be letters, numbers or Chinese characters. For whitespace characters, such as "t" "n" "r", escaping is required. For other characters, escaping is not required.

**Index Configuration**

Index Status ☒

Full-Text Index ☒ ☒ Case-sensitive

Full-text delimiter

Key-Value Index ☒ ☐ Case-sensitive

Key-Value Index	Field Type	Delimiter	Operation
<input type="text" value="remote_addr"/>	<input type="text" value="text"/>	<input type="text" value="!@#%^&amp;*()_-=, &lt;&gt;/?\:\n\t\r"/>	<a href="#">Delete</a>
<input type="text" value="remote_port"/>	<input type="text" value="text"/>	<input type="text" value="!@#%^&amp;*()_-=, &lt;&gt;/?\:\n\t\r"/>	<a href="#">Delete</a>
<input type="text" value="status"/>	<input type="text" value="long"/>	-	<a href="#">Delete</a>

4. The index configuration is as shown below:

**Index Configuration** [Edit](#)

Index Status **Enabled**

Full-Text Index **Enabled** ☒ Case-sensitive

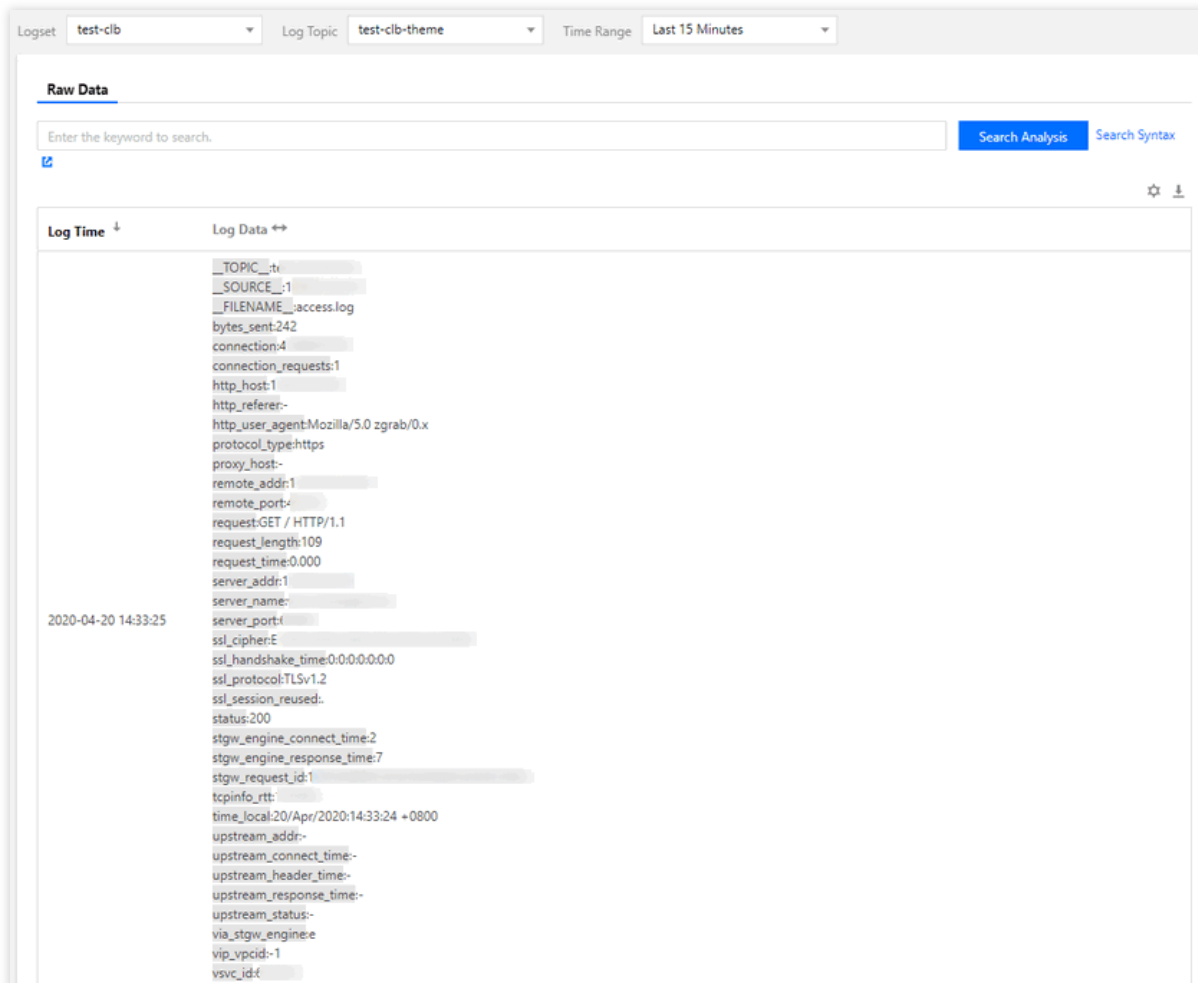
Full-text delimiter

Key-Value Index **Enabled**

Key-Value Index	Field Type	Delimiter
remote_addr	text	!@#%^&*()_-=, <>/?\:\n\t\r
remote_port	text	!@#%^&*()_-=, <>/?\:\n\t\r
status	long	None
server_addr	text	!@#%^&*()_-=, <>/?\:\n\t\r
server_name	text	!@#%^&*()_-=, <>/?\:\n\t\r
http_host	text	!@#%^&*()_-=, <>/?\:\n\t\r
request_time	double	None

### Step 3. View access logs

1. Log in to the [CLS console](#), and click **Search Analysis** in the left sidebar.
2. On the **Search Analysis** page, select a logset, log topic, and time range, and click **Search Analysis** to search for the access logs reported by CLB to CLS. For more information about the search syntax, see [Legacy CLS Search Syntax](#).



## Method 2: Batch Configure Access Logging

### Step 1: Create a logset and log topic.

To configure access logs in CLS, you need to first create a logset and log topic.

If you have created a logset and log topic, skip to [Step 2](#).

1. Log in to the [CLB console](#) and click **Access Logs** in the left sidebar.
2. On the **Access Logs** page, select a region for the logset, and then click **Create Logset** in the **Logset information** section.
3. In the pop-up **Create Logset** window, set the retention period and click **Save**.

#### Note:

You can create only a single logset named "clb\_logset" in each region.

4. Click **Create Log Topic** in the **Log Topic** section of the **Access Logs** page.
5. In the pop-up window, specify the storage type and log retention period, select a CLB instance in the list on the left and add it to the list on the right, and then click **Save**.

#### Note:



Supported storage types: STANDARD storage and IA storage. For more information, see [Storage Class Overview](#). Logs can be retained permanently or for a specified period of time.

When you create a log topic, you can add a CLB instance as needed. To add a CLB instance after a log topic is created, click **Manage** in the **Operation** column of the log topic in the list. Each CLB instance can be added to only one log topic.

A logset can contain multiple log topics. You can categorize CLB logs into various log topics which will be marked with **CLB** by default.

6. (Optional) To disable access logging, click **Disable**.

### Step 2. View access logs

Without any manual configurations, CLB has been automatically configured with index search by access log valuable. You can directly query access logs through search and analysis.

1. Log in to the [CLB console](#) and click **Access Logs** in the left sidebar.
2. Click **Search** in the **Operation** column of the topic log topic to go to the **Search Analysis** page in the [CLS console](#).
3. On the **Search Analysis** page, enter the search syntax in the input box, select a time range, and then click **Search Analysis** to search for access logs reported by CLB to CLS.

**Note:**

For more information about the search syntax, see [Syntax Rules](#).

## Log Format and Variable Description

### Log format

```
[$stgw_request_id] [$time_local] [$protocol_type] [$server_addr:$server_port]
[$server_name] [$remote_addr:$remote_port] [$status] [$upstream_addr]
[$upstream_status] [$proxy_host] [$request] [$request_length] [$bytes_sent]
[$http_host] [$http_user_agent] [$http_referer] [$request_time]
[$upstream_response_time] [$upstream_connect_time] [$upstream_header_time]
[$tcpinfo_rtt] [$connection] [$connection_requests] [$ssl_handshake_time]
[$ssl_cipher] [$ssl_protocol] [$vip_vpcid] [$uri] [$server_protocol]
```

### Field type

Currently, CLS supports the following three field types:

Name	Description
text	Text type.
long	Integer type (Int 64).

double	Floating point type (64 bit).
--------	-------------------------------

## Log variable description

Variable Name	Description	Field Type
stgw_request_id	Request ID.	text
time_local	Access time and time zone. Example: <code>01/Jul/2019:11:11:00+0800</code> , where <code>+0800</code> represents UTC+8.	text
protocol_type	Protocol type. Supported protocols: HTTP, HTTPS, SPDY, HTTP2, WS, and WSS.	text
server_addr	VIP of the CLB instance.	text
server_port	CLB VPort, that is, the listening port.	long
server_name	<code>server_name</code> value of a rule, that is, the domain name configured in a CLB listener.	text
remote_addr	Client IP address.	text
remote_port	Client port.	long
status	Status code returned by the CLB instance to the client.	long
upstream_addr	Address of the real server (RS).	text
upstream_status	Status code returned by the RS to the CLB instance.	text
proxy_host	Stream ID.	text
request	Request line.	text
request_length	Number of bytes of the request received from the client.	long
bytes_sent	Number of bytes sent to the client.	long
http_host	Request domain name, which is the value of the <b>Host</b> field in the HTTP header.	text
http_user_agent	<code>user_agent</code> field in the HTTP header.	text
http_referer	Source of the HTTP request.	text
http_x_forward_for	Content of <code>x-forward-for</code> header in the HTTP request.	text

request_time	Request processing time, which is duration from when the first byte is received from the client to when the last byte is sent to the client, that is, the total time consumed by the whole process in which the client request reaches the CLB instance, the CLB instance forwards the request to an RS, the RS responds and sends data to the CLB instance, and finally the CLB instance forwards the data to the client. Unit: seconds.	double
upstream_response_time	The time that an entire backend request process takes, starting from when the CLB instance connects with an RS to when the RS receives the request and responds. Unit: seconds.	double
upstream_connect_time	The time taken to establish a TCP connection with an RS, starting from when the CLB instance connects with the RS to when the CLB instance sends an HTTP request.	double
upstream_header_time	The time taken to receive an HTTP header from the RS, starting from when the CLB instance connects with the RS to when the HTTP response header is received from the RS.	double
tcpinfo_rtt	The round-trip time (RTT) of the TCP connection.	long
connection	Connection ID.	long
connection_requests	Number of requests in the connection	long
ssl_handshake_time	<p>Time in microseconds taken by SSL handshake phases, in the format of <code>x:x:x:x:x:x:x</code>, with the time strings of different phases separated by colons (:). If the time of a phase is less than 1 ms, <code>0</code> is displayed.</p> <p>The first field indicates whether the SSL session is reused.</p> <p>The second field indicates the time taken by the entire handshake process.</p> <p>The third to seventh fields indicate the time taken by each SSL handshake phase.</p> <p>The third field indicates the time from when the CLB instance receives <code>client hello</code> to when the CLB instance sends <code>server hello done</code>.</p> <p>The fourth field indicates the time from when the CLB instance starts sending the server certificate to when the CLB instance finishes sending the server certificate.</p> <p>The fifth field indicates the time from when the CLB instance calculates the signature to when the CLB instance finishes sending <code>server key exchange</code>.</p> <p>The sixth field indicates the time from when the CLB instance starts receiving <code>client key exchange</code> to when the CLB instance</p>	text

	finishes receiving <code>client key exchange</code> . The seventh field indicates the time from when the CLB instance receives <code>client key exchange</code> to when the CLB instance sends <code>server finished</code> .	
ssl_cipher	SSL cipher suite.	text
ssl_protocol	SSL protocol version.	text
vip_vpcid	ID of the VPC instance to which the CLB instance belongs. The <code>vip_vpcid</code> value of a public network CLB instance is <code>-1</code> .	long
request_method	Request method. Only POST and GET requests are supported.	text
uri	Uniform resource identifier.	text
server_protocol	Protocol used for CLB.	text

## Default search log valuable

The following fields can be found in logsets with "CLB" by default:

Index Field	Description	Field Type
time_local	Access time and time zone. Example: <code>01/Jul/2019:11:11:00+0800</code> , where <code>+0800</code> represents UTC+8.	text
protocol_type	Protocol type. Supported protocols: HTTP, HTTPS, SPDY, HTTP2, WS, and WSS.	text
server_addr	VIP of the CLB instance.	text
server_name	<code>server_name</code> value of a rule, that is, the domain name configured in a CLB listener.	text
remote_addr	Client IP address.	text
status	Status code returned by the CLB instance to the client.	long
upstream_addr	Address of the RS.	text
upstream_status	Status code returned by the RS to the CLB instance.	text
request_length	Number of bytes of the request received from the client.	long
bytes_sent	Number of bytes sent to the client.	long

http_host	Request domain name, which is the value of the <b>Host</b> field in the HTTP header.	text
request_time	Request processing time, which is duration from when the first byte is received from the client to when the last byte is sent to the client, that is, the total time consumed by the whole process in which the client request reaches the CLB instance, the CLB instance forwards the request to an RS, the RS responds and sends data to the CLB instance, and finally the CLB instance forwards the data to the client.Unit: seconds.	double
upstream_response_time	The time that an entire backend request process takes, starting from when the CLB instance connects with an RS to when the RS receives the request and responds.Unit: seconds.	double

# Sampling Logs

Last updated : 2024-01-04 14:34:05

After you enable layer-7 access logging or health check logging, if the log volume is large, full log reporting may result in high log costs. Tencent Cloud Load Balancer (CLB) supports log collection through sampling to reduce the amount of reported data and reduce log costs.

**Note:**

You can configure to store CLB access logs and health check logs in Tencent Cloud Log Service (CLS) to achieve log search, analysis, visualization, and alarming. CLS is separately billed. For more information about the billing of CLS, see [Product Pricing](#).

## Prerequisites

You have created the logset and log topics for access logs. For more information, see [Configuring Access Logs](#).  
You have created the logset and log topics for health check logs. For more information, see [Configuring Health Check Logs](#).

## Sampling Layer-7 Access Logs

1. Log in to the [CLB console](#) and choose **Access logs > Log list** in the left sidebar.
2. In the top-left corner of the **Access logs** page, select your region. Find the target log topic in the log topic list and choose **More > Sample** in the **Operation** column.
3. In the **Sample CLB logs** pop-up window, turn on the sampling switch and configure the parameters as needed.

Parameter	Description
Sample	If the switch is turned on, log sampling is enabled. If the switch is turned off, full logs are collected.
Default ratio	If you configured the log sampling rule, logs that do not match the sampling rule are sampled based on the default sampling ratio. You can enter an integer from 1 to 100.
Sampling field	The <b>status</b> field is currently supported.
Sampling rule	Sampling rules support regular expressions. For example, if you want to sample logs whose status code is 400 or 500, you can set the sampling rule as to <b>400 500</b> .
Sampling ratio	Sampling ratio. You can enter an integer from 1 to 100.

Operation	You can delete the sampling rule.
Add	If the existing sampling rules do not meet your needs, you can add more sampling rules. At most five sampling rules can be configured for each log topic.

Sample CLB logs

Sample

Default ratio ⓘ

10

%

Logs are sampled based on the sampling rule and sampling ratio. The sampling rule supports regular expressions, and the sampling ratio is an integer between 1-100. [Learn more](#)


Sampling field	Sampling rule	Sampling ratio	Operation
<div>status ▾</div>	<div>400 500</div>	<div>20</div> <div>%</div>	Delete

Add

Submit

Cancel

4. Click **Submit** to return to the log topic list page. If log sampling is enabled for a log topic, the word **Sampling** is displayed next to the topic name.

test	<div>Sampling</div>	Shipping	30 
------	---------------------	----------	------------------------------------------------------------------------------------------

## Sampling Health Check Logs

1. Log in to the [CLB console](#) and click **Health Check Logs** in the left sidebar.
2. Other steps are the same as those described in the [Sampling Layer-7 Access Logs](#) section.

## References

---

[Configuring Access Logs](#)

[Configuring Health Check Logs](#)



# Configuring Health Check Logs

Last updated : 2025-01-22 14:15:17

CLB supports storing health check logs to CLS where you can view the logs, reporting at a minute granularity and querying online by multiple rules, helping you identify the causes of health check failures.

## Note:

The feature is in a beta test. To try it out, [submit a ticket](#).

Health check logging includes log reporting, storage and query:

Log reporting: Service forwarding has a higher priority than log reporting.

Log storage and query: SLA is guaranteed based on the storage service currently in use.

## Restrictions

Health check is a transition log, and health check logs are generated only when the health status of the real server changes.

CLB layer-4 and layer-7 protocols can be used for storing health check logs to CLS.

Storing CLB health check logs to CLS is now free of charge. You only need to pay for the CLS service.

This feature is available only to CLB (formerly known as application CLB) instances.

This feature is supported only in certain regions as displayed in the console.

## Step 1: Add a Role Permission

To add a role permission, make sure you have activated the CLS service.

1. Log in to the [CLB console](#) and click **Health Check Logs** in the left sidebar.
2. On the **Health Check Logs** page, click **Activate now**, and then click **Authorize and Activate** in the pop-up window.
3. Switch to the **Role Management** page in the [CAM console](#), and click **Grant**.

## Step 2: Create a Logset and Log Topic

To store health check logs to CLS, you need to first create a logset and log topic.

If you have created a logset and log topic, skip to [Step 3](#).

1. Log in to the [CLB console](#) and click **Health Check Logs** in the left sidebar.
2. On the **Health Check Logs** page, select a region for the logset, and then click **Create Logset** in the **Logset information** section.

3. In the pop-up **Create Logset** window, set the retention period and click **Save**.
4. Click **Create Log Topic** in the **Log Topic** section of the **Health Check Logs** page.
5. In the pop-up window, specify the storage type and log retention period, select a CLB instance in the list on the left and add it to the list on the right, and then click **Save**.

**Note:**

Supported storage types: STANDARD storage and IA storage. For more information, see [Storage Class Overview](#). Logs can be retained permanently or for a specified period of time.

When you create a log topic, you can add a CLB instance as needed. To add a CLB instance after a log topic is created, click **Manage** in the **Operation** column of the log topic in the list. Each CLB instance can be added to only one log topic.

A logset can contain multiple log topics. You can categorize CLB logs into various log topics which will be marked with "CLB" by default.

6. (Optional) To disable health check logging, click **Disable**.

### Step 3. View Health Check Logs

- Without any manual configurations, CLB has been automatically configured with index search by health check log valuable. You can directly query health check logs through search and analysis.
1. Log in to the [CLB console](#) and click **Health Check Logs** in the left sidebar.
  2. On the **Health Check Logs** page, select the region of the logset you want to view. In the **Log Topic** section, click **Search** in the **Operation** column of the log topic you select to go to the [CLS Console](#).
  3. In the CLS console, click **Search Analysis** in the left sidebar.
  4. On the **Search Analysis** page, enter the search syntax in the input box, select a time range, and then click **Search Analysis** to search for health check logs reported by CLB to CLS.
- Note:**
- For more information about the search syntax, see [Syntax Rules](#).

### Health Check Log Format and Variable

#### Log format

```
[$protocol] [$rsport] [$rs_vpcid] [$vport] [$vpcid] [$time] [$vip] [$rsip] [$status]
[$domain] [$url]
```

#### Log variable description

Variable	Description	Field
----------	-------------	-------

Name		Type
protocol	Protocol type. Supported protocols: HTTP, HTTPS, SPDY, HTTP2, WS, and WSS.	text
rsport	Port of the real server.	long
rs_vpcid	VPC ID of the real server. The <code>vip_vpcid</code> value of a public network CLB instance is <code>-1</code> .	long
vport	CLB VPort, that is, the listening port.	long
vpcId	VPC ID of the VIP of the CLB instance. The <code>vip_vpcid</code> value of a public network CLB instance is <code>-1</code> .	long
time	Access time and time zone. Example: <code>01/Jul/2019:11:11:00 +0800</code> , where <code>+0800</code> represents UTC+8.	text
vip	VIP of the CLB instance.	text
rsip	IP address of the real server.	text
status	Health status. Valid values: <code>true</code> : healthy <code>false</code> : unhealthy	text
domain	Domain name to be checked. This parameter is left empty if a layer-4 listener is used.	text
url	URL to be checked. This parameter is left empty if a layer-4 listener is used.	text

## References

[Getting Started in Five Minutes](#)

# Accessing Log Dashboard

Last updated : 2024-01-04 14:34:05

By connecting CLB access logs to Cloud Log Service, you can check the access logs in a dashboard. The dashboard provides charts of multiple metrics, giving you a full picture of the load balancer.

## Dashboard

Each log topic has its own dashboard, which contains data of following metrics.

PV

UV

Outgoing request message traffic

Incoming response traffic

Average request time

Average response time

Backend status code distribution

Overall status code distribution

PV/UV trend

Outgoing/Incoming traffic trend

Average requests/responses per minute

P99, P95, P90, P50 access duration

Top requested instances

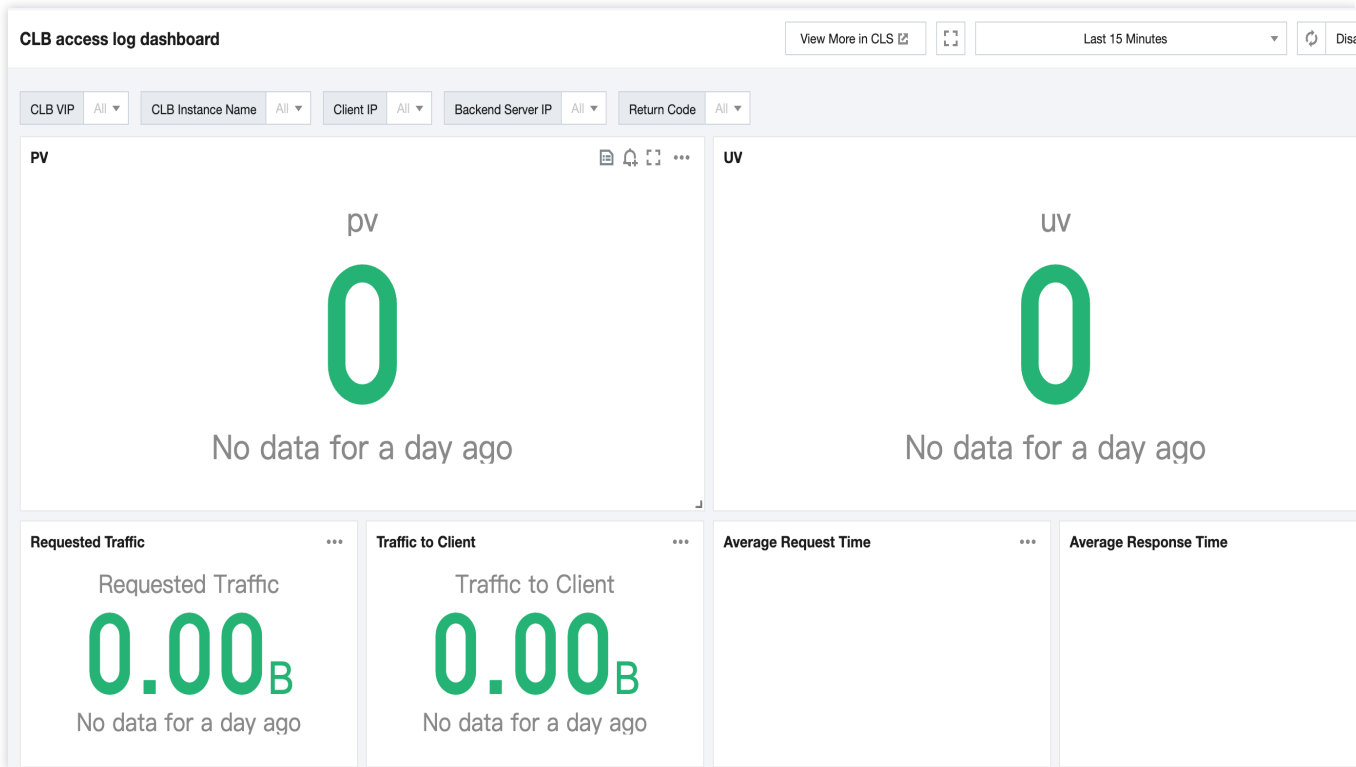
Top requested domain names

## Preparations

Create logsets for CLB. See [Creating Logsets and Log Topics](#).

## Directions

1. Log in to the [CLB console](#) and select **Access Logs** on the left sidebar.
2. In the **Access Log Dashboard** page, select the region and log topic to see the dashboard of this log topic.



3. (Optional) In the upper corner of the **Access Log Dashboard** page, filter logs by the CLB VIP, client IP, backend server IP and status code.

## See Also

For more information, see [Configuring Access Logs](#).

# Monitoring and Alarm

## Obtaining Monitoring Data

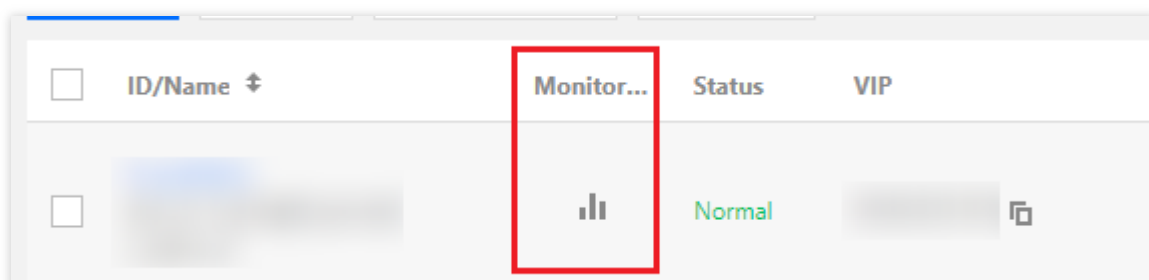
Last updated : 2024-01-04 14:34:05

Tencent Cloud Observability Platform collects and displays data for the CLB instance and the real server, helping you obtain CLB statistics, verify whether the system is running normally, and create alarms. For more information about Tencent Cloud Observability Platform, see the [Tencent Cloud Observability Platform](#) documentation.

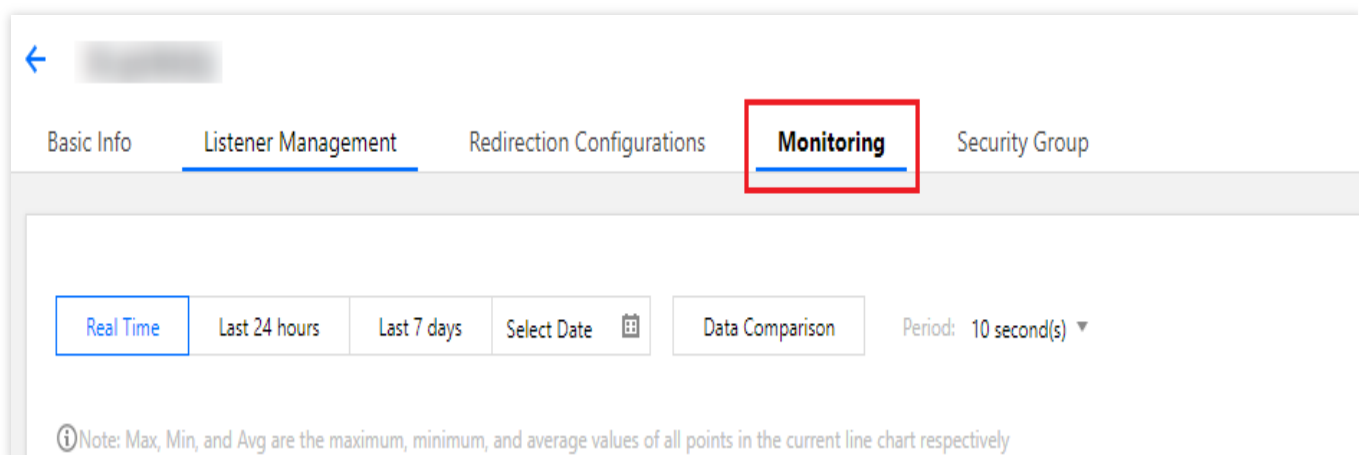
Tencent Cloud provides the Tencent Cloud Observability Platform feature for all users by default and does not require manual activation. You can use Tencent Cloud Observability Platform to collect the monitoring data of your CLB instances and view the data using the following methods.

### CLB Console Method

1. Log in to the [CLB console](#), click the monitoring icon next to the CLB instance ID, and then browse the performance data of the instance in the floating window.



2. Click the ID/Name of the CLB instance to access its details page. Click **Monitoring** to view its monitoring data.



### Tencent Cloud Observability Platform Console Method

Log in to the [Tencent Cloud Observability Platform console](#) to view CLB monitoring data. Click **Cloud Load Balancer** on the left sidebar, and then click the ID/name of the CLB instance to access its monitoring details page. You can view the monitoring data of the CLB instance and expand its drop-down list to view the listener and real server monitoring information.

## API Method

Use the `GetMonitorData` API to get the monitoring data of all products. For more information, see [GetMonitorData](#), [Public Network CLB Monitoring Metrics](#), [Private Network CLB](#).

# Monitoring Metrics

Last updated : 2024-11-27 15:00:34

Tencent Cloud Observability Platform (TCOP) collects raw data from the running CLB instances and displays the data entries in intuitive graphs. Statistics will be kept for one month by default. You can observe the operations of instances in the month to stay informed of the status of application services.

You can log in to the [TCOP console](#) to view CLB monitoring data. Choose **Cloud Product Monitoring > Cloud Load Balancer** and then click a CLB instance ID to enter the monitoring details page. You can view monitoring data of the CLB instance, and expand it to view the listener and real server monitoring information.

**Note:**

For now, the data of concurrent connection utilization and new connection utilization of LCU-supported CLB instances are reported. These data of shared CLB instances are not reported.

## CLB Instance Level

Metric Name	Metric Meaning	Metric Description	Unit	Statistical Period (Sec)
ClientConnum	Client-to-CLB active connections	Number of active connections from clients to a CLB instance or listener at a certain time point in the statistical period.	-	10/60/300
ClientInactiveConn	Client-to-CLB inactive connections	Number of inactive connections from clients to a CLB instance or listener at a certain time point in the statistical period.	-	10/60/300
ClientConcurConn	Client-to-CLB concurrent connections	Number of concurrent connections from clients to a CLB instance or listener at a certain time point in the statistical period.	-	10/60/300
ConcurConnVipRatio	Concurrent connection utilization	The utilization of the concurrent connections from clients to a CLB at a certain time point in the statistical period, compared to the	%	10/60/300



		performance upper limit of the concurrent connections for the performance capacity specification. This metric is only supported by the performance capacity instance and the speed-limited shared instance.		
ClientNewConn	Client-to-CLB new connections	Number of new connections from clients to a CLB instance or listener in the statistical period.	Count/s	10/60/300
NewConnVipRatio	New connection utilization	The utilization of the new connections from clients to a CLB at a certain time point in the statistical period, compared to the performance upper limit of the new connections for the performance capacity specification. This metric is only supported by the performance capacity instance and the speed-limited shared instance.	%	10/60/300
ClientInpkg	Client-to-CLB packets in	Number of data packets sent per second from clients to a CLB instance in the statistical period.	Count/s	10/60/300
ClientOutpkg	Client-to-CLB packets out	Number of data packets sent per second from a CLB instance to clients in the statistical period.	Count/s	10/60/300
ClientAccIntraffic	Client-to-CLB traffic in	Traffic from clients to a CLB instance in the statistical period.	MB	10/60/300
ClientAccOuttraffic	Client-to-CLB traffic out	Traffic from a CLB instance to clients in the statistical period.	MB	10/60/300
ClientOuttraffic	Client-to-CLB	Bandwidth used by traffic	Mbps	10/60/300

	bandwidth out	from a CLB instance to clients in the statistical period.		
ClientInTraffic	Client-to-CLB bandwidth in	Bandwidth used by traffic from clients to a CLB instance in the statistical period.	Mbps	10/60/300
OutTraffic	CLB-to-real server bandwidth out	Bandwidth used by traffic from real servers to a CLB instance in the statistical period.	Mbps	60/300
InTraffic	CLB-to-real server bandwidth in	Bandwidth used by traffic from a CLB instance to real servers in the statistical period.	Mbps	60/300
AccOuttraffic	CLB-to-real server traffic out	Traffic from real servers to a CLB instance in the statistical period. This metric is supported only by public network CLB instances.	MB	10/60/300/3600
DropTotalConns	Dropped connections	Number of connections dropped by a CLB instance or listener in the statistical period. This metric is supported only by bill-by-IP accounts. For more information about how to determine the account type, see <a href="#">Checking Account Type</a> .	-	10/60/300
InDropBits	Dropped bandwidth in	Bandwidth dropped by clients when accessing a CLB instance through a public network in the statistical period. This metric is supported only by bill-by-IP accounts. For more information about how to determine the account	Bits	10/60/300

		type, see <a href="#">Checking Account Type</a> .		
OutDropBits	Dropped bandwidth out	Bandwidth dropped by a CLB instance when accessing a public network in the statistical period. This metric is supported only by bill-by-IP accounts. For more information about how to determine the account type, see <a href="#">Checking Account Type</a> .	Bits	10/60/300
InDropPkts	Dropped packets in	Number of packets dropped by clients when accessing a CLB instance through a public network in the statistical period. This metric is supported only by bill-by-IP accounts. For more information about how to determine the account type, see <a href="#">Checking Account Type</a> .	Count/s	10/60/300
OutDropPkts	Dropped packets out	Number of packets dropped by a CLB instance when accessing a public network in the statistical period. This metric is supported only by bill-by-IP accounts. For more information about how to determine the account type, see <a href="#">Checking Account Type</a> .	Count/s	10/60/300
DropQps	Dropped QPS	Number of requests dropped by a CLB instance or listener in the statistical period. This metric is dedicated to layer-7 listeners and is supported only by bill-by-IP accounts. For more information about how to	-	60/300

		determine the account type, see <a href="#">Checking Account Type</a> .		
IntrafficVipRatio	Inbound bandwidth utilization	<p>Utilization of bandwidth used by clients to access a CLB instance through a public network in the statistical period.</p> <p>This metric is supported only by bill-by-IP accounts. For more information about how to determine the account type, see <a href="#">Checking Account Type</a>. This metric is currently in beta. To try it out, <a href="#">submit a ticket</a>.</p> <p>This metric is only supported by the performance capacity instance and the speed-limited shared instance.</p> <p>The numerator is the current inbound bandwidth and the denominator is the upper limit of the bandwidth of LCU-supported instances or the upper limit of rate-limited shared instances.</p>	%	10/60/300
OuttrafficVipRatio	Outbound bandwidth utilization	<p>Utilization of bandwidth used by a CLB instance to access a public network in the statistical period.</p> <p>This metric is supported only by bill-by-IP accounts. For more information about how to determine the account type, see <a href="#">Checking Account Type</a>. This metric is currently in beta. To try it out, <a href="#">submit a ticket</a>.</p> <p>This metric is only supported by the performance capacity instance and the speed-limited shared instance.</p> <p>The numerator is the current outbound bandwidth and the</p>	%	10/60/300

		denominator is the upper limit of the bandwidth of LCU-supported instances or the upper limit of the rate-limited shared instances.		
ReqAvg	Average request time	Average request time of a CLB instance in the statistical period. This metric is dedicated to layer-7 listeners.	Millisecond	60/300
ReqMax	Maximum request time	Maximum request time of a CLB instance in the statistical period. This metric is dedicated to layer-7 listeners.	Millisecond	60/300
RspAvg	Average response time	Average response time of a CLB instance in the statistical period. This metric is dedicated to layer-7 listeners.	Millisecond	60/300
RspMax	Maximum response time	Maximum response time of a CLB instance in the statistical period. This metric is dedicated to layer-7 listeners.	Millisecond	60/300
RspTimeout	Timed-out responses	Number of CLB timed-out responses in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
SuccReq	Successful requests per minute	Number of successful requests per minute of a CLB instance in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
TotalReq	Requests per second	Number of requests per second of a CLB instance in the statistical period.	-	60/300

		This metric is dedicated to layer-7 listeners.		
QpsVipRatio	QPS Utilization Rate	Within a specific moment in the statistical granularity, the utilization rate of the QPS performance upper limit of the load balancing is compared to the QPS performance of the performance capacity type specification. This metric is solely supported by performance capacity instances, and is not compatible with shared instances.	%	60、300
ClbHttp2xx	2xx status codes returned by CLB	Number of 2xx status codes of a CLB instance in the statistical period. (Status codes returned by both the CLB instance and real servers are counted.) This metric is dedicated to layer-7 listeners.	Count/min	60/300
ClbHttp3xx	3xx status codes returned by CLB	Number of 3xx status codes of a CLB instance in the statistical period. (Status codes returned by both the CLB instance and real servers are counted.) This metric is dedicated to layer-7 listeners.	Count/min	60/300
ClbHttp4xx	4xx status codes returned by CLB	Number of 4xx status codes of a CLB instance in the statistical period. (Status codes returned by both the CLB instance and real servers are counted.) This metric is dedicated to layer-7 listeners.	Count/min	60/300
ClbHttp5xx	5xx status	Number of 5xx status codes	Count/min	60/300

	codes returned by CLB	of a CLB instance in the statistical period. (Status codes returned by both the CLB instance and real servers are counted.) This metric is dedicated to layer-7 listeners.		
ClbHttp404	404 status codes returned by CLB	Number of 404 status codes of a CLB instance in the statistical period. (Status codes returned by both the CLB instance and real servers are counted.) This metric is dedicated to layer-7 listeners.	Count/min	60/300
ClbHttp499	499 status codes returned by CLB	Number of 499 status codes of a CLB instance in the statistical period. (Status codes returned by both the CLB instance and real servers are counted.) This metric is dedicated to layer-7 listeners.	Count/min	60/300
ClbHttp502	502 status codes returned by CLB	Number of 502 status codes of a CLB instance in the statistical period. (Status codes returned by both the CLB instance and real servers are counted.) This metric is dedicated to layer-7 listeners.	Count/min	60/300
ClbHttp503	503 status codes returned by CLB	Number of 503 status codes of a CLB instance in the statistical period. (Status codes returned by both the CLB instance and real servers are counted.) This metric is dedicated to layer-7 listeners.	Count/min	60/300
ClbHttp504	504 status codes	Number of 504 status codes of a CLB instance in the	Count/min	60/300

	returned by CLB	statistical period. (Status codes returned by both the CLB instance and real servers are counted.) This metric is dedicated to layer-7 listeners.		
ClbOther	Other status codes returned from CLB	Number of other status codes returned by a CLB within the statistical period (the sum of the status codes returned by the CLB and the backend servers). The number of these other status codes is equivalent to the sum of the 1xx status codes and the 2xx status codes returned by the CLB. This metric is dedicated to layer-7 listeners.	Count/min	60/300
Http2xx	2xx status codes	Number of 2xx status codes returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
Http3xx	3xx status codes	Number of 3xx status codes returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
Http4xx	4xx status codes	Number of 4xx status codes returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
Http5xx	5xx status codes	Number of 5xx status codes returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
Http404	404 status	Number of 404 status codes	Count/min	60/300



	codes	returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.		
Http499	499 status codes	Number of 499 status codes returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
Http502	502 status codes	Number of 502 status codes returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
Http503	503 status codes	Number of 503 status codes returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
Http504	504 status codes	Number of 504 status codes returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
OverloadCurConn	Concurrent SNAT connections	Number of concurrent connections per minute to the SNAT IP addresses of a CLB instance in the statistical period. This metric is still in beta. To try it out, <a href="#">submit a ticket</a> .	Count/min	60
ConnRatio	SNAT port utilization	Port utilization of the SNAT IP addresses of a CLB instance in the statistical period. Port utilization = Number of concurrent SNAT connections/(Number of SNAT IP addresses x 55000 x Number of servers)	%	60

		This metric is still in beta. To try it out, <a href="#">submit a ticket</a> .		
SnatFail	Failed SNAT connections	Number of failed connections per minute between the SNAT IP addresses of a CLB instance and real servers in the statistical period. This metric is still in beta. To try it out, <a href="#">submit a ticket</a> .	Count/min	60
HealthRsCount	Number of those passing the health check	During the statistical period, the number of CLBs passing the health check	-	60/300
UnhealthRsCount	Abnormal health checks	Number of abnormal health checks of a CLB instance in the statistical period.	-	60/300

## Layer-4 Listener (TCP/UDP) Level

Layer-4 listeners allow you to view the monitoring metrics at three levels:

Listener level

Real server level

Real server port level

Metric Name	Metric Meaning	Metric Description	Unit	Statistical Period (Sec)
ClientConnum	Client-to-CLB active connections	Number of active connections from clients to a CLB instance or listener at a certain time point in the statistical period.	-	10/60/300
ClientNewConn	Client-to-CLB new connections	Number of new connections from clients to a CLB instance or listener in the statistical period.	Count/s	10/60/300
ClientInpkg	Client-to-CLB packets in	Number of data packets sent per second from clients to a	Count/s	10/60/300

		CLB instance in the statistical period.		
ClientOutpkg	Client-to-CLB packets out	Number of data packets sent per second from a CLB instance to clients in the statistical period.	Count/s	10/60/300
ClientAccIntraffic	Client-to-CLB traffic in	Traffic from clients to a CLB instance in the statistical period.	MB	10/60/300
ClientAccOuttraffic	Client-to-CLB traffic out	Traffic from a CLB instance to clients in the statistical period.	MB	10/60/300
ClientOuttraffic	Client-to-CLB bandwidth out	Bandwidth used by traffic from a CLB instance to clients in the statistical period.	Mbps	10/60/300
ClientIntraffic	Client-to-CLB bandwidth in	Traffic from clients to a CLB instance in the statistical period.	Mbps	10/60/300
OutTraffic	CLB-to-real server bandwidth out	Bandwidth used by traffic from real servers to a CLB instance in the statistical period.	Mbps	60/300
InTraffic	CLB-to-real server bandwidth in	Bandwidth used by traffic from a CLB instance to real servers in the statistical period.	Mbps	60/300
OutPkg	CLB-to-real server packets out	Number of data packets sent per second from real servers to a CLB instance in the statistical period.	Count/s	60/300
InPkg	CLB-to-real server packets in	Number of data packets sent per second from a CLB instance to real servers in the statistical period.	Count/s	60/300
AccOuttraffic	CLB-to-real server traffic out	Traffic from real servers to a CLB instance in the statistical period. This metric is supported only by public network CLB instances.	MB	10/60/300/3600

ConNum	CLB-to-real server connections	Number of connections from a CLB instance to real servers in the statistical period.	-	60/300
NewConn	CLB-to-real server new connections	Number of new connections from a CLB instance to real servers in the statistical period.	Count/min	60/300
HealthRsCount	Number of those passing the health check	During the statistical period, the number of CLBs passing the health check	-	60/300
UnhealthRsCount	Abnormal health checks	Number of abnormal health checks of a CLB instance in the statistical period.	-	60/300

## Layer-7 Listener (HTTP/HTTPS) Level

Layer-7 listeners allow you to view the monitoring metrics at three levels:

Listener level

Real server level

Real server port level

Metric Name	Metric Meaning	Metric Description	Unit	Statistical Period (Sec)
ClientConnum	Client-to-CLB active connections	Number of active connections from clients to a CLB instance or listener at a certain time point in the statistical period.	-	10/60/300
ClientNewConn	Client-to-CLB new connections	Number of new connections from clients to a CLB instance or listener in the statistical period.	Count/s	10/60/300
ClientInpkg	Client-to-CLB packets in	Number of data packets sent per second from clients to a CLB instance in the statistical period.	Count/s	10/60/300
ClientOutpkg	Client-to-CLB packets out	Number of data packets sent per second from a CLB	Count/s	10/60/300

		instance to clients in the statistical period.		
ClientAccIntraffic	Client-to-CLB traffic in	Traffic from clients to a CLB instance in the statistical period.	MB	10/60/300
ClientAccOuttraffic	Client-to-CLB traffic out	Traffic from a CLB instance to clients in the statistical period.	MB	10/60/300
ClientOuttraffic	Client-to-CLB bandwidth out	Bandwidth used by traffic from a CLB instance to clients in the statistical period.	Mbps	10/60/300
ClientIntraffic	Client-to-CLB bandwidth in	Bandwidth used by traffic from clients to a CLB instance in the statistical period.	Mbps	10/60/300
OutTraffic	CLB-to-real server bandwidth out	Bandwidth used by traffic from real servers to a CLB instance in the statistical period.	Mbps	60/300
InTraffic	CLB-to-real server bandwidth in	Bandwidth used by traffic from a CLB instance to real servers in the statistical period.	Mbps	60/300
OutPkg	CLB-to-real server packets out	Number of data packets sent per second from real servers to a CLB instance in the statistical period.	Count/s	60/300
InPkg	CLB-to-real server packets in	Number of data packets sent per second from a CLB instance to real servers in the statistical period.	Count/s	60/300
AccOuttraffic	CLB-to-real server traffic out	Traffic from real servers to a CLB instance in the statistical period. This metric is supported only by public network CLB instances.	MB	10/60/300/3600
ConNum	CLB-to-real server connections	Number of connections from a CLB instance to real servers in the statistical period.	-	60/300

NewConn	CLB-to-real server new connections	Number of new connections from a CLB instance to real servers in the statistical period.	Count/min	60/300
ReqAvg	Average request time	Average request time of a CLB instance in the statistical period. This metric is dedicated to layer-7 listeners.	Millisecond	60/300
ReqMax	Maximum request time	Maximum request time of a CLB instance in the statistical period. This metric is dedicated to layer-7 listeners.	Millisecond	60/300
RspAvg	Average response time	Average response time of a CLB instance in the statistical period. This metric is dedicated to layer-7 listeners.	Millisecond	60/300
RspMax	Maximum response time	Maximum response time of a CLB instance in the statistical period. This metric is dedicated to layer-7 listeners.	Millisecond	60/300
RspTimeout	Timed-out responses	Number of CLB timed-out responses in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
SuccReq	Successful requests per minute	Number of successful requests per minute of a CLB instance in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
TotalReq	Requests per second	Number of requests per second of a CLB instance in the statistical period. This metric is dedicated to layer-7 listeners.	-	60/300
QpsVipRatio	QPS Utilization	Within a specific moment in the	%	60、300

	Rate	<p>statistical granularity, the utilization rate of the QPS performance upper limit of the load balancing is compared to the QPS performance of the performance capacity type specification.</p> <p>This metric is solely supported by performance capacity instances, and is not compatible with shared instances.</p>		
DropQps	Discard QPS	<p>Within the statistical granularity, the number of requests dropped on the load balancer or listener. This metric is unique to the seven-layer listener and is only supported by standard account types. Traditional account types do not support it. For the method of determining the account type, please refer to <a href="#">Checking Account Type</a>.</p>	-	60、300
ClbHttp2xx	2xx status codes returned by CLB	<p>Number of 2xx status codes of a CLB instance in the statistical period. (Status codes returned by both the CLB instance and real servers are counted.)</p> <p>This metric is dedicated to layer-7 listeners.</p>	Count/min	60、300
ClbHttp3xx	3xx status codes returned by CLB	<p>Number of 3xx status codes of a CLB instance in the statistical period. (Status codes returned by both the CLB instance and real servers are counted.)</p> <p>This metric is dedicated to layer-7 listeners.</p>	Count/min	60/300
ClbHttp4xx	4xx status codes returned by CLB	<p>Number of 4xx status codes of a CLB instance in the statistical period. (Status codes returned</p>	Count/min	60/300

		by both the CLB instance and real servers are counted.) This metric is dedicated to layer-7 listeners.		
ClbHttp5xx	5xx status codes returned by CLB	Number of 5xx status codes of a CLB instance in the statistical period. (Status codes returned by both the CLB instance and real servers are counted.) This metric is dedicated to layer-7 listeners.	Count/min	60/300
ClbHttp404	404 status codes returned by CLB	Number of 404 status codes of a CLB instance in the statistical period. (Status codes returned by both the CLB instance and real servers are counted.) This metric is dedicated to layer-7 listeners.	Count/min	60/300
ClbHttp499	499 status codes returned by CLB	Number of 499 status codes of a CLB instance in the statistical period. (Status codes returned by both the CLB instance and real servers are counted.) This metric is dedicated to layer-7 listeners.	Count/min	60/300
ClbHttp502	502 status codes returned by CLB	Number of 502 status codes of a CLB instance in the statistical period. (Status codes returned by both the CLB instance and real servers are counted.) This metric is dedicated to layer-7 listeners.	Count/min	60/300
ClbHttp503	503 status codes returned by CLB	Number of 503 status codes of a CLB instance in the statistical period. (Status codes returned by both the CLB instance and real servers are counted.) This metric is dedicated to layer-7 listeners.	Count/min	60/300
ClbHttp504	504 status codes	Number of 504 status codes of	Count/min	60/300



	returned by CLB	a CLB instance in the statistical period. (Status codes returned by both the CLB instance and real servers are counted.) This metric is dedicated to layer-7 listeners.		
ClbOther	Other status codes returned by CLB	Number of other status codes returned by a CLB within the statistical period (the sum of the status codes returned by the CLB and the backend servers). The number of these other status codes is equivalent to the sum of the 1xx status codes and the 2xx status codes returned by the CLB. This metric is dedicated to layer-7 listeners.	Count/min	60/300
Http2xx	2xx status codes	Number of 2xx status codes returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
Http3xx	3xx status codes	Number of 3xx status codes returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
Http4xx	4xx status codes	Number of 4xx status codes returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
Http5xx	5xx status codes	Number of 5xx status codes returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
Http404	404 status codes	Number of 404 status codes	Count/min	60/300

		returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.		
Http499	499 status codes	Number of 499 status codes returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
Http502	502 status codes	Number of 502 status codes returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
Http503	503 status codes	Number of 503 status codes returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
Http504	504 status codes	Number of 504 status codes returned by real servers in the statistical period. This metric is dedicated to layer-7 listeners.	Count/min	60/300
HealthRsCount	Number of those passing the health check	During the statistical period, the number of CLBs passing the health check	-	60/300
UnhealthRsCount	Abnormal health checks	Number of abnormal health checks of a CLB instance in the statistical period.	-	60/300

**Note:**

If you want to view the monitoring data of a real server under a listener, log in to the [CLB console](#), click the monitoring icon next to the CLB instance ID, and then browse the performance data of the instance in the floating window.

## References

[Public Network CLB](#)

# Configuring Alarm Policy

Last updated : 2024-01-04 14:34:05

This document describes how to create an alarm policy.

## Use Cases

You can set threshold alarms for the performance consumption metrics of the monitor types supported by Tencent Cloud Observability Platform. You can also set event alarms for the service status of Tencent Cloud service instances or the underlying platform infrastructure. This way, when an exception occurs, you will promptly receive notifications, which will allow you to take appropriate measures. An alarm policy consists of five required parameters: name, policy type, alarm trigger condition, alarm object, and alarm notification template. You can create alarm policies by following the directions below:

## Concepts

Term	Definition
Alarm policy	It consists of alarm name, alarm policy type, alarm trigger condition, alarm object, and alarm notification template
Alarm policy type	Alarm policy type identifies policy category and corresponds to specific Tencent Cloud products. For example, if you choose the CVM policy, you can customize metric alarms for CPU utilization, disk utilization, and more
Alarm trigger condition	An alarm trigger condition is a semantic condition consisting of metric, comparison, threshold, statistical period, and N consecutive monitoring data points
Monitor type	Types include Tencent Cloud service monitoring, application performance monitoring, frontend performance monitoring, and cloud automated testing
Notification template	A notification template can be quickly reused for multiple policies, making it suitable for alarm receipt in various use cases. For more information, see <a href="#">Creating Alarm Notification Template</a>

## Directions

1. Log in to the [Tencent Cloud Observability Platform](#).

2. Click **Alarm Configuration > Alarm Policy** to enter the alarm policy configuration page.

3. Click **Add** and configure a new alarm policy as shown below:

Configuration Type	Configuration Item	Description
Basic info	Policy name	Custom policy name
	Remarks	Custom policy remarks
	Monitor type	Types include Tencent Cloud service monitoring, application performance monitoring, frontend performance monitoring, and cloud automated testing
	Policy type	Select the desired policy type for monitoring Tencent Cloud services.
	Projects	This configuration item has two functions: Manage alarm policies. Alarm policies of a project can be quickly located in the alarm policy list. It manages instances. Choose a project based on your needs. Then, in "Alarm Object", you can quickly select instances under the project. You can assign Tencent Cloud services to each project based on your business types. If you want to create a project, see <a href="#">Project Management</a> . After creating a project, you can use the console of each Tencent Cloud service to assign projects to resources. Some Tencent Cloud services such as TencentDB for MySQL do not support project assignment. In that case, you can refer to <a href="#">Specifying Project for Instance</a> to assign projects to the corresponding instances. If you do not have project permissions, see <a href="#">Cloud Access Management (CAM)</a> to get permissions.
Configure alarm rule	Alarm object	If you select "instance ID", the alarm policy will be associated with the selected instance. If you select "instance group", the alarm policy will be associated with the selected instance group. All Objects: associate the policy with all instances under the current account (permission required)
	Trigger condition(choosing Manual Configuration)	An alarm trigger condition is a semantic condition consisting of metric, comparison, threshold, measurement period, and N monitoring data points. You can set an alarm threshold according to the trend of metric change in the chart. For example, if the metric is CPU utilization, the comparison is `>`, the threshold is `80%`, the measurement period is `5 minutes`, and the consecutive monitoring data points is `2 data points`, then data on the CPU utilization of a CVM instance will be

		<p>collected once every 5 minutes, and an alarm will be triggered if the CPU utilization exceeds 80% for two consecutive periods.</p> <p>Alarm frequency: you can set a repeated notification policy for each alarm rule. This way, an alarm notification will be sent repeatedly at a specified frequency when an alarm is triggered.</p> <p>Frequency options: do not repeat, once every 5 minutes, once every 10 minutes, at an exponentially increasing interval, and other frequency options.</p> <p>An exponentially increasing interval means that a notification is sent when an alarm is triggered the first time, second time, fourth time, eighth time, and so on. In other words, the alarm notification will be sent less and less frequently as time goes on to reduce the disturbance caused by repeated notifications.</p> <p>Default logic for repeated alarm notifications: the alarm notification will be sent to you at the configured frequency within 24 hours after the alarm is triggered. After 24 hours, the alarm notification will be sent once a day by default.</p>
	Configure manually (event alarm)	You can create event alarms so that when the Tencent Cloud service resources or the underlying infrastructure services encounter any errors, you will promptly receive notifications and can then take measures accordingly.
	Select template	Click Select template and select a configured template from the drop-down list. For detailed configurations, see <a href="#">Configuring Trigger Condition Template</a> . If a newly created template is not displayed, click Refresh on the right.
Configure alarm notification	Alarm notification	You can select a preset or custom notification template. Each alarm policy can be bound to three notification templates at most. For more information, see <a href="#">Notification Template</a> .
Advanced configuration	Auto scaling	After this option is enabled and configured successfully, an auto scaling policy will be triggered for scaling when the alarm condition is met.

4. After configuring the above information, click **Save**.

**Note:**

CVM alarms can be sent normally only after the monitoring [Agent](#) has been installed on CVM instances and reports monitoring metric data. On the Cloud Monitor page, you can view CVM instances that do not have Agent installed and download the IP address list.

# Alarming Metric Descriptions

Last updated : 2024-01-04 14:34:05

## Alert Policies

Set up metric-based alert policies to notify target recipients when the monitored metric reaches the threshold.

You can create alert policies on the following levels:

Public network listener

Private network listener

Real server

Listener (Not available for private-network Classic CLBs)

Server port

Layer-7 protocol

## Public/Private Network Listeners

Supported network listener metrics:

Metric	Unit	Description
Inbound bandwidth	Mbps	Bandwidth consumed when data is coming into the CLB over the public network within a sampling period.
Outbound bandwidth	Mbps	Bandwidth consumed by the CLB to access the public network within a sampling period.
Inbound packets	Packets/s	Number of request data packets received by the CLB per second within a sampling period.
Outbound packets	Packets/s	Number of data packets sent by the CLB per second within a sampling period.

## Real Server

CLB supports alert policies on the listener and server port level.

### 1. Listener level

Set up an policy to trigger alerts when the number of abnormal ports of real servers bound to the listener reaches the threshold. In the example below, the number of abnormal ports of all real servers under the selected listener is

collected once every minute. If the number is greater than 10 per second for two consecutive sampling period, an alert is triggered. Only one alert is sent per day.

**Note :**

To activate listener-level alert, [submit a ticket](#).

Configure the alert source:

Alarm Object

☐ All Objects

☒ Select some objects(2 selected)

☐ Select instance group [Create instance group](#)

Region: Guangzhou Project: DEFAULT PROJECT

Region: Guangzhou Project: DEFAULT PROJECT

http(http:12) ☒

tcp(tcp:1222) ☐

http1(http:121) ☐

http2(http:1211) ☐

8(http:80) ☐

7(tcp:7) ☒

ID	VIP	Listener
	1	TCP:7
	1	HTTP:12

Configure trigger conditions:

Trigger Condition

☐ Trigger Condition Template [Add Trigger Condition Template](#)

☒ Configure trigger conditions

☒ Indicator alarm

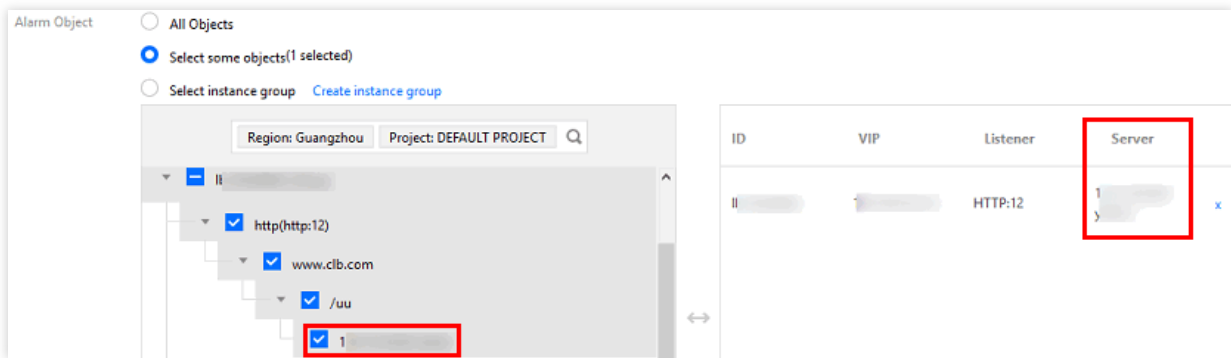
RS\_UNHEALTH\_NUM Measurement Period > 10 Continuous Alarm occurs every 1

Add

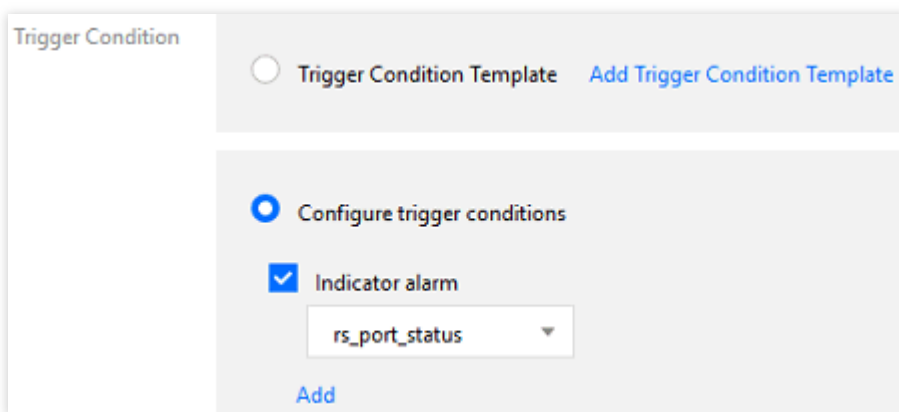
## 2. Server port level

You can configure a policy to receive alerts whenever a specified port of a real server bound to a listener is abnormal.

Configure the alert source:



Configure trigger conditions:



#### Note :

Real server port exception: The port of the real server is unavailable. Network jitter can also trigger port exceptions. A listener-level policy is suggested as it covers ports of all bound real servers and is not affected by network jitter.

## Layer-7 Protocol Monitoring

You can configure metri-based alert policies for layer-7 (HTTP/HTTPS) listeners. The specific metrics are as follows:

Metric	Unit	Description
Inbound bandwidth	Mbps	Bandwidth used by the client to access CLB over the public network within a sampling period.
Outbound bandwidth	Mbps	Bandwidth used by CLB to access the public network within a sampling period.
Inbound packets	Packets/s	Number of request data packets received by CLB per second within a sampling period.
Outbound packets	Packets/s	Number of data packets sent by CLB per second within a sampling period.



New connections	-	Number of new connections established per minute within a sampling period.
Active connections	-	Number of active connections per minute within a sampling period.
Average response time	ms	Average response time of CLB within a sampling period.
Longest response time	ms	Longest response time of CLB within a sampling period.
Real server - 2xx status codes	-	Number of 2xx status codes returned by the real server within a sampling period.
Real server - 3xx status codes	-	Number of 3xx status codes returned by the real server within a sampling period.
Real server - 4xx status codes	-	Number of 4xx status codes returned by the real server within a sampling period.
Real server - 5xx status codes	-	Number of 5xx status codes returned by the real server within a sampling period.
Real server - 404 status codes	-	Number of 404 status codes returned by the real server within a sampling period.
Real server - 502 status codes	-	Number of 502 status codes returned by the real server within a sampling period.
CLB - 3xx status codes	-	Number of 3xx status codes returned by CLB within a sampling period.
CLB - 4xx status codes	-	Number of 4xx status codes returned by CLB within a sampling period.
CLB - 5xx status codes	-	Number of 5xx status codes returned by CLB within a sampling period.
CLB - 404 status codes	-	Number of 404 status codes returned by CLB within a sampling period.
CLB - 502 status codes	-	Number of 502 status codes returned by CLB within a sampling period.

# Cloud Access Management Overview

Last updated : 2024-01-04 14:34:05

If you use multiple Tencent Cloud services such as CLB, CVM, and TencentDB that are managed by different users sharing your Tencent Cloud account key, you may face the following problems:

Your key is shared by multiple users, leading to high risk of compromise.

You cannot limit the access permissions of other users, which poses a security risk due to potential faulty operations.

[Cloud Access Management \(CAM\)](#) is used to manage the access permissions to your Tencent Cloud resources. With CAM, you can use the identity management and policy management features to control which Tencent Cloud resources can be accessed by which sub-accounts.

For example, if you have multiple CLB instances under your account that are deployed in different projects, to manage access permissions and authorize resources, you can bind the admin of project A with an authorization policy, which states that only this admin can use the CLB resources under project A.

If you do not need to manage the access permission to CLB resources for sub-accounts, you can skip this chapter.

This will not affect your understanding and usage of other parts in the documentation.

## Basic Concepts in CAM

The root account authorizes sub-accounts by binding policies. The policy setting can be specific to the level of **API, Resource, User/User Group, Allow/Deny, and Condition**.

### 1. Account

#### Root account

As the fundamental owner of Tencent Cloud resources, a root account acts as the basis for resource usage fee calculation and billing, and can be used to log in to Tencent Cloud services.

#### Sub-account

A sub-account is created by the root account, and it has a specific ID and identity credential that can be used to log in to the Tencent Cloud Console. A root account can create multiple sub-accounts (users). **A sub-account does not own any resources by default; instead, such resources should be authorized by its root account.**

#### Identity credential

This includes login credentials and access certificates. **Login credential** refers to the username and password.

**Access certificate** refers to the TencentCloud API keys (SecretId and SecretKey).

### 2. Resources and permissions

#### Resource

A resource is an object that is operated in Tencent Cloud service, such as a CVM instance and a VPC instance.

### Permission

Permission is an authorization to allow or forbid certain users to perform certain operations. By default, **a root account has full access to all the resources under it**, while **a sub-account does not have access to any resources under its root account**.

### Policy

Policy is the syntax rule used to define and describe one or multiple permissions. **A root account** performs authorization by **associating policies** with users/user groups.

For more information, please see [CAM Overview](#).

## Related Documents

Document Description	Link
Relationship between policy and user	<a href="#">Policy</a>
Basic policy structure	<a href="#">Element Reference</a>
More products that support CAM	<a href="#">CAM-enabled Cloud Services</a>

# Authorization Definition

Last updated : 2024-01-04 14:34:05

## Types of CLB Resources That Can Be Authorized in CAM

Resource Type	Resource Description in Authorization Policy
CLB instance	<code>qcs::clb:\$region::clb/\$loadbalancerid</code>
CLB real server	<code>qcs::cvm:\$region:\$account:instance/\$cvminstanceid</code>

In the description:

- `$region` must be the ID of a region and can be empty.
- `$account` must be the `AccountId` of the resource owner or `*`.
- `$loadbalancerid` must be the ID of a CLB instance or `*`.

And so on...

## APIs for CLB Authorization in CAM

You can authorize the following actions for a CLB resource in CAM.

### Instance

API Operation	Resource Description	API Description
<code>DescribeLoadBalancers</code>	Queries the CLB instance list.	<code>*</code> , which indicates to authenticate only the API.
<code>CreateLoadBalancer</code>	Purchases a CLB instance.	<code>qcs::\$projectid:clb:\$region:\$account:clb/*</code>
<code>DeleteLoadBalancers</code>	Deletes CLB instances.	<code>qcs::clb:\$region:\$account:clb/\$loadbalancerid</code>
<code>ModifyLoadBalancerAttributes</code>	Modifies	<code>qcs::clb:\$region:\$account:clb/\$loadbalancerid</code>

	the attributes of a CLB instance.	
ModifyForwardLBName	Modifies the name of a CLB instance.	<code>qcs::clb:\$region:\$account:clb/\$loadbalanceri</code>
SetLoadBalancerSecurityGroups	Configures security groups for a CLB instance.	<code>qcs::clb:\$region:\$account:clb/\$loadbalanceri</code>

Listener

API Operation	Resource Description	API Description
DeleteLoadBalancerListeners	Deletes CLB listeners.	<code>qcs::clb:\$region:\$account:clb/\$loadbal</code>
DescribeLoadBalancerListeners	Gets the CLB listener list.	<code>qcs::clb:\$region:\$account:clb/\$loadbal</code>
ModifyLoadBalancerListener	Modifies the attributes of a CLB listener.	<code>qcs::clb:\$region:\$account:clb/\$loadbal</code>
CreateLoadBalancerListeners	Creates CLB listeners.	<code>qcs::clb:\$region:\$account:clb/\$loadbal</code>
DeleteForwardLBListener	Deletes a layer-4 or layer-7 CLB listener.	<code>qcs::clb:\$region:\$account:clb/\$loadbal</code>
ModifyForwardLBSeventhListener	Modifies	<code>qcs::clb:\$region:\$account:clb/\$loadbal</code>

	the attributes of a layer-7 CLB listener.	
ModifyForwardLBFourthListener	Modifies the attributes of a layer-4 CLB listener.	<code>qcs::clb:\$region:\$account:clb/\$loadbal</code>
DescribeForwardLBLEaders	Queries the CLB listener list.	<code>qcs::clb:\$region:\$account:clb/\$loadbal</code>
CreateForwardLBSeventhLayerListeners	Creates layer-7 CLB listeners.	<code>qcs::clb:\$region:\$account:clb/\$loadbal</code>
CreateForwardLBFourthLayerListeners	Creates layer-4 CLB listeners.	<code>qcs::clb:\$region:\$account:clb/\$loadbal</code>

## CLB domain name and URL

API Operation	Resource Description	API Description
ModifyForwardLBRulesDomain	Modifies the domain name of a forwarding rule for a CLB listener.	<code>qcs::clb:\$region:\$account:clb/\$loadbalancer</code>
CreateForwardLBListenerRules	Creates forwarding rules for a CLB listener.	<code>qcs::clb:\$region:\$account:clb/\$loadbalancer</code>
DeleteForwardLBListenerRules	Deletes the forwarding	<code>qcs::clb:\$region:\$account:clb/\$loadbalancer</code>

	rules of a layer-7 CLB listener.	
DeleteRewrite	Deletes the redirection relationship between CLB forwarding rules.	<code>qcs::clb:\$region:\$account:clb/\$loadbalancer</code>
ManualRewrite	Manually creates a redirection relationship between CLB forwarding rules.	<code>qcs::clb:\$region:\$account:clb/\$loadbalancer</code>
AutoRewrite	Automatically generates a redirection relationship between CLB forwarding rules.	<code>qcs::clb:\$region:\$account:clb/\$loadbalancer</code>

Real server

API Operation	Resource Description	API Description
ModifyLoadBalancerBackends	Modifies real server weights for a CLB instance.	<code>qcs::clb:\$region:\$account:clb/</code>
DescribeLoadBalancerBackends	Gets the list of real servers bound to a	<code>qcs::clb:\$region:\$account:clb/</code>

	CLB instance.	
DeregisterInstancesFromLoadBalancer	Unbinds real servers.	<code>qcs::clb:\$region:\$account:clb/</code>
RegisterInstancesWithLoadBalancer	Binds real servers to a CLB instance.	<code>qcs::clb:\$region:\$account:clb/</code>
DescribeLBHealthStatus	Queries the health status of a CLB instance.	<code>qcs::clb:\$region:\$account:clb/</code>
ModifyForwardFourthBackendsPort	Modifies the CVM instance port in a forwarding rule of a layer-4 listener.	<code>qcs::clb:\$region:\$account:clb/</code>
ModifyForwardFourthBackendsWeight	Modifies the weight of CVM instances in a forwarding rule of a layer-4 listener.	<code>qcs::clb:\$region:\$account:clb/</code>
RegisterInstancesWithForwardLBSeventhListener	Binds CVM instances to the forwarding rules of a layer-7 CLB listener.	<code>qcs::clb:\$region:\$account:clb/</code>
RegisterInstancesWithForwardLBFourthListener	Binds CVM instances to the forwarding rules of a	<code>qcs::clb:\$region:\$account:clb/</code>



	layer-4 CLB listener.	
DeregisterInstancesFromForwardLBFourthListener	Unbinds CVM instances from the forwarding rules of a layer-4 CLB listener.	<code>qcs::clb:\$region:\$account:clb/</code>
DeregisterInstancesFromForwardLB	Unbinds CVM instances from the forwarding rules of a layer-7 CLB listener.	<code>qcs::clb:\$region:\$account:clb/</code>
ModifyForwardSeventhBackends	Modifies the weight of CVM instances in the forwarding rules of a layer-7 listener.	<code>qcs::clb:\$region:\$account:clb/</code>
ModifyForwardSeventhBackendsPort	Modifies the port of CVM instances in the forwarding rules of a layer-7 listener.	<code>qcs::clb:\$region:\$account:clb/</code>
DescribeForwardLBBackends	Queries the list of CVM instances bound to a CLB instance.	<code>qcs::clb:\$region:\$account:clb/</code>

DescribeForwardLBHealthStatus	Queries the health check status of a CLB instance.	<code>qcs::clb:\$region:\$account:clb/</code>
ModifyLoadBalancerRulesProbe	Modifies the health check configuration and forwarding path in a forwarding rule of a CLB listener.	<code>qcs::clb:\$region:\$account:clb/</code>

# Policy Examples

Last updated : 2024-01-04 14:34:05

## Full Access Policy for All CLB Instances

Grant a sub-account full access to the CLB service (creating, managing, etc.).

Policy name: CLBResourceFullAccess

```
{
 "version": "2.0",
 "statement": [{
 "action": [
 "name/clb:*"
],
 "resource": "*",
 "effect": "allow"
 }]
}
```

## Read-Only Policy for All CLB Instances

Grant a sub-account read-only access to CLB (i.e., the permission to view but not to create, update, or delete all CLB resources). In the console, the prerequisite to manipulate a resource is the ability to view the resource; therefore, you are recommended to grant the sub-account full read access to CLB.

Policy name: CLBResourceReadOnlyAccess

```
{
 "version": "2.0",
 "statement": [{
 "action": [
 "name/clb:Describe*"
],
 "resource": "*",
 "effect": "allow"
 }]
}
```

## Full Access Policy for CLB Service Under a Specified Tag

Grant a sub-account full access to the CLB service (creating instances, managing listeners, etc.) under a specified tag (tag key: tagkey; tag value: tagvalue).

CLB instances supports configuring tags and using tags for authentication.

```
{
 "version": "2.0",
 "statement": [
 {
 "effect": "allow",
 "action": "*",
 "resource": "*",
 "condition": {
 "for_any_value:string_equal": {
 "qcs:tag": [
 "tagkey&tagvalue"
]
 }
 }
 }
]
}
```

# Classic CLB

## Classic CLB Overview

Last updated : 2024-01-04 14:34:05

### Overview

Classic CLB is easy to configure and supports simple load balancing scenarios:

**Public network** classic CLB: supports TCP/UDP/HTTP/HTTPS protocols.

**Private network** classic CLB: supports TCP/UDP protocols.

CLB instances can be classified into two types: CLB (formerly "application CLB") and classic CLB.

CLB includes all features of classic CLB. Based on their features and performance, we recommend using CLB. For detailed comparison, see [Instance Types](#).

Note :

Currently, there are two types of Tencent Cloud accounts: bill-by-EIP/CLB and bill-by-CVM. All Tencent Cloud accounts registered after June 17, 2020 00:00:00 are bill-by-EIP/CLB accounts. For Tencent Cloud accounts registered before June 17, 2020, [check your account types](#) in the console. Bill-by-EIP/CLB accounts no longer support classic CLB. You can now only purchase a CLB instance.

This document introduces classic CLB instances. After creating an instance, you need to configure a listener for it. The listener listens to requests on the CLB instance and distributes traffic to the real server based on the load balancing policy.

### Listener Configurations

You need to configure a CLB listener as follows:

1. Listener protocol and listening port. The listening port, or frontend port, is used to receive and forward requests to real servers.
2. Backend port. It is the port through which the CVM instance provides services, receives and processes traffic from the CLB instance.
3. Listening policy, such as load balancing policy and session persistence.
4. Health check policy.
5. Real server can be bound by selecting its IP.

Note :

If you configure multiple listeners to a classic CLB instance and bind multiple real servers, each listener will forward requests to all real servers based on its configuration.

### Supported protocol types

A CLB listener can listen to Layer-4 and Layer-7 requests on a CLB instance and distribute them to real servers for processing. The main difference between Layer-4 CLB and Layer-7 CLB is which protocol is used to forward traffic when load balancing user requests.

Layer-4 protocols: transport layer protocols, including TCP and UDP.

Layer-7 protocols: application layer protocols, including HTTP and HTTPS.

Note :

1. A classic CLB instance receives requests and forwards traffic to the real server via VIP and port. Layer-7 protocols do not support forwarding based on domain name and URL.
2. A private network classic CLB instance only supports Layer-4 protocols, not Layer-7 protocols.
3. If you need aforementioned advanced features, we recommend choosing CLB over classic CLB. For more information, see [Instance Types](#).

### Port Configuration

Listening Port (frontend port)	Service Port (backend port)	Description
<p>The listening port is used by a CLB instance to receive and forward requests to real servers for load balancing.</p> <p>You can configure CLB for the port range 1-65535, such as 21 (FTP), 25 (SMTP), 80 (HTTP), and 443 (HTTPS).</p>	<p>A service port is used by the CVM to provide services, receives and processes traffic from the CLB instance.</p> <p>On a CLB instance, one listening port can forward traffic to ports of multiple CVM instances.</p>	<p>On a CLB instance, a listening port must be unique. For example, TCP:80 and HTTP:80 listeners cannot be created at the same time. Only TCP and UDP ports can be the same. For example, you can create both TCP:80 and UDP:80 listeners.</p> <p>The same service ports can be used on a CLB instance. For example, HTTP:80 and HTTPS:443 listeners can be bound to the same port of a CVM instance.</p>

# Configuring Classic CLB

Last updated : 2024-01-04 14:34:05

After creating a classic CLB instance, you need to configure a listener for it. The listener listens to requests on the instance and distributes traffic to real servers based on the load balancing policy.

## Prerequisites

You need to [create a CLB instance](#) first and select "Classic CLB" for **Instance type**.

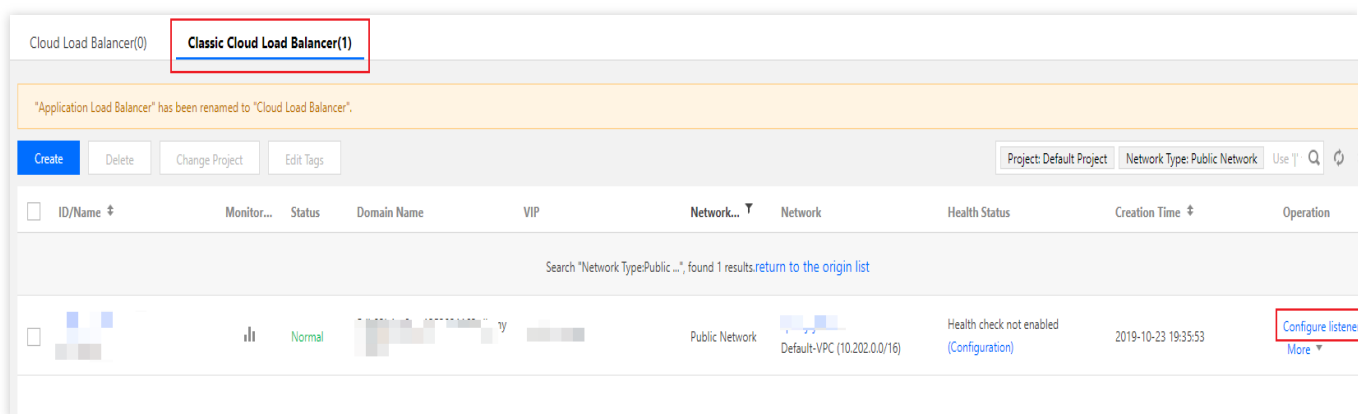
Note :

Currently, there are two types of Tencent Cloud accounts: bill-by-EIP/CLB and bill-by-CVM. All Tencent Cloud accounts registered after June 17, 2020 00:00:00 are bill-by-EIP/CLB accounts. For Tencent Cloud accounts registered before June 17, 2020, [check your account types](#) in the console. Bill-by-EIP/CLB accounts no longer support classic CLB. You can now only purchase a CLB instance.

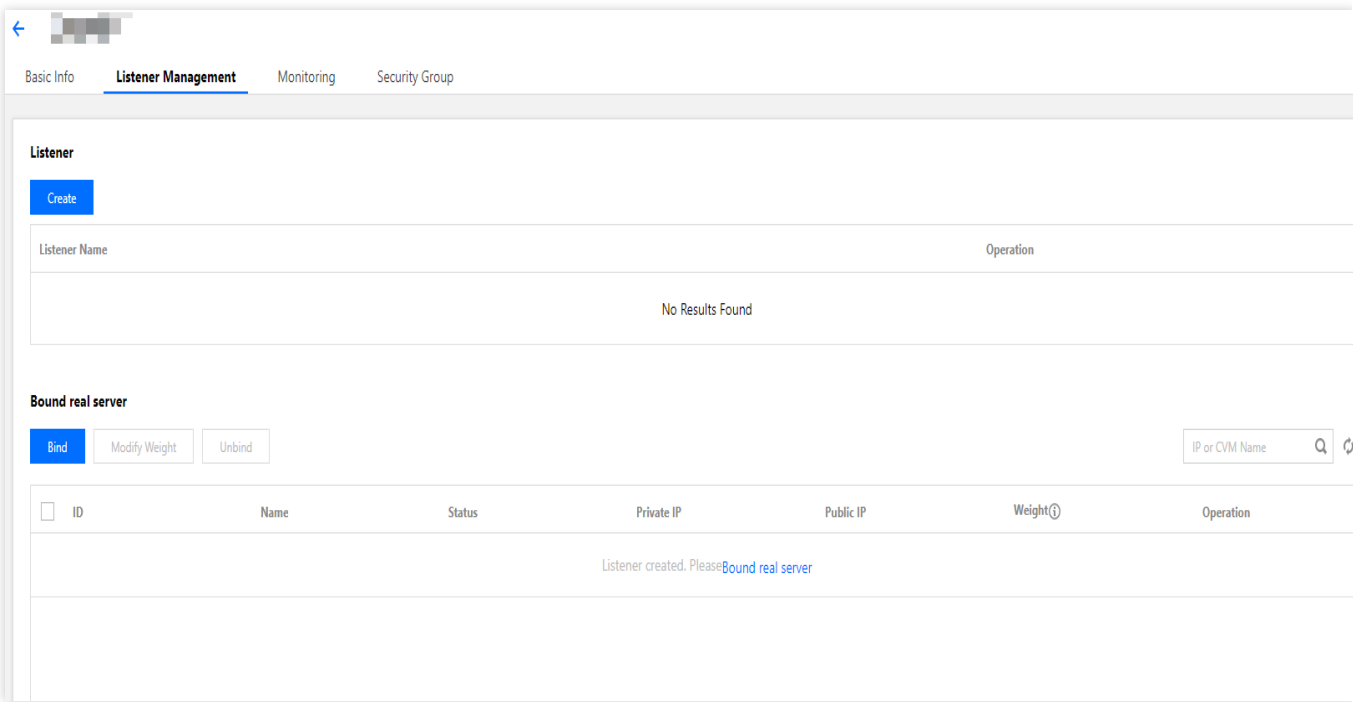
## Configuring the Listener

### Step 1. Open the Listener Management page

1. Log in to the [CLB Console](#).
2. Select **CLB Instance List** on the left sidebar.
3. On the **Instance Management** page, click the ID/Name of the instance to be configured to enter the instance details page.
4. Select the **Listener Management** tab, or click **Configure listener** under the **Operation** column on the **Instance Management** page.



5. The **Listener Management** page is as shown below.



Step 2. Configure a listener

Click **Create** under **Listener Management** and configure a TCP listener in the pop-up window.

1. Basic configuration

Configuration Item	Description	Example
Name	Listener name.	test-tcp-80
Listener Protocol Ports	Listener protocol and listening port Listener protocol: CLB supports protocols such as TCP, UDP, HTTP, and HTTPS. This example uses TCP. Listening port: used to receive and forward requests to real servers. The port range is 1-65535. The listening port must be unique in the same CLB instance.	TCP:80
Backend Port	The port through which the CVM instance provides services, receives and processes traffic from a CLB instance.	80

To create a TCP listener, complete the basic configuration as shown below:



## CreateListener

1

Basic Configuration

2

Advanced Configuration

3

Health Check

Name

test-tcp-80

Listen Protocol Ports ⓘ

TCP

:

80

Backend Port

80

Close

Next

## 2. Advanced configuration

Configuration Item	Description	Example
Balance Method	<p>For TCP listeners, CLB supports two scheduling algorithms: weighted round robin (WRR) and weighted least-connection (WLC).</p> <p>WRR: requests are forwarded to different real servers sequentially according to their weights. Scheduling is based on the <b>number of new connections</b>, where servers with higher weights have more polls (i.e., a higher probability) and servers with the same weight process the same number of connections.</p> <p>WLC: loads on servers are estimated according to their number of active connections. Scheduling is based on server loads and weights. If their weights are the same, real servers with fewer active connections will have more polls (i.e., a higher probability).</p>	WRR
Session Persistence	<p>Whether to enable or disable session persistence.</p> <p>After session persistence is enabled, CLB listener will distribute access requests from the same client to the same real server.</p> <p>TCP session persistence is implemented based on client IP address. The access requests from the same IP address are forwarded to the same real server.</p> <p>Session persistence can be enabled for WRR scheduling but not WLC scheduling.</p>	Enabled
Hold Time	<p>Session persistence time.</p> <p>If there is no new request in the connection within the session persistence time, session persistence will be automatically disconnected.</p> <p>Value range: 30-3600 seconds.</p>	30s

Complete the configuration as shown below:

CreateListener

1 Basic Configuration

2 **Advanced Configuration**

3 Health Check

Balance Method

Weighted Round Robin

If you set a same weighted value for all CVMs, requests will be distributed by a simple pooling policy.

Session Persistence ⓘ

☒

Hold Time ⓘ

III

30 Seconds

3600 Seconds

–

30

+

Seconds

Session persistence based on the source IP

Back

Next

### 3. Health check

Configuration Item	Description	Example
Health Check	Whether to enable or disable health check. In TCP listeners, CLB instances send SYN packets to specified server ports to perform health checks.	Enabled
Check Protocol	To be added.	To be added
Check Port	To be added.	To be added
Response Timeout	Maximum response timeout period for health check. If a real server fails to respond within the timeout period, it is considered as unhealthy. Value range: 2-60 seconds. Default value: 2s.	2s
Check Interval	Interval between two health checks. Value range: 5-300 seconds. Default value: 5s.	5s
Unhealthy Threshold	If the health check returns <code>failure</code> for n consecutive times (n is user-defined), the real server is unhealthy and the <b>unhealthy</b> status is displayed in the console. Value range: 2-10 times. Default value: 3 times	3 times

Healthy Threshold	If the health check returns <code>success</code> for n consecutive times (n is user-defined), the real server is healthy and the <b>healthy</b> status is displayed in the console. Value range: 2-10 times. Default value: 3 times.	3 times
-------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------

Complete the health check configuration as shown below:

CreateListener

✓ Basic Configuration

✓ Advanced Configuration

3 Health Check

Health Check ⓘ

Hide Advanced Options ▲

Response Timeout

2 Seconds

60 Seconds

—

2

+

Seconds

Check Interval

5 Seconds

300 Seconds

—

5

+

Seconds

Unhealthy Threshold ⓘ

2 Times

10 Times

—

3

+

Times

Healthy Threshold ⓘ

2 Times

10 Times

—

3

+

Times

Back

Submit

Step 3. Bind a real server

Click **Bind** on the **Listener Management** page and select the real server to be bound in the pop-up window, as shown below:

Bind CVM

Note: To ensure the forwarding works properly, please set the public network bandwidth to more than 0 MB for the CVM associated with public CLB.

Select CVM

IP or CVM Name

1 selected

Cloud Virtual Machine	Weight ⓘ	
<div></div>	10 <div>↑</div> <div>↓</div>	✕

Hold Shift to select multiple items

OK

Cancel

The configuration is as shown below:

Basic Info

Listener Management

Monitoring

Security Group

Listener

Create

Listener Name

Operation

> test-tcp-80 (TCP:80)ModifyDelete

Bound real server

Bind

Modify Weight

Unbind

IP or CVM Name

ID

Name

Status

Private IP

Public IP

Weight①

Operation

ins-hg0utoiv

Unnamed

Running

10.202.0.8

162.62.14.209

10

Unbind

Note :

If you configure multiple listeners to a classic CLB instance and bind multiple real servers, each listener will forward requests to all real servers based on its configuration.

Step 4. Security group (optional)

You can configure a CLB security group to isolate public network traffic. For more information, see [Configuring a CLB Security Group](#).

Step 5. Modify or delete a listener (optional)

If you need to modify or delete an existing listener, select the listener on the **Listener Management** page and click **Modify** or **Delete**.

Basic Info

Listener Management

Monitoring

Security Group

Listener

Create

Listener Name

Operation

> test-tcp-80 (TCP:80)ModifyDelete

# Managing Real Servers of Classic CLB Instances

Last updated : 2024-01-04 14:34:05

Classic CLB routes requests to real server instances that are running normally. This document describes how to add or delete real servers as needed or when you use Classic CLB for the first time.

## Prerequisites

You have created a Classic CLB instance and configured a listener. For more information, please see [Getting Started with Classic CLB](#).

## Directions

### Adding real server to Classic CLB instance

Note :

If a Classic CLB instance is associated with an auto scaling group, CVM instances in the group will be automatically added to the real servers of the Classic CLB instance. When a CVM instance is removed from the auto scaling group, it will be automatically deleted from the real servers of the Classic CLB instance.

If you need to use API to add real servers, please see the [RegisterTargetsWithClassicalLB](#) API.

1. Log in to the [CLB Console](#).
2. On the "Instance Management" page, select the **Classic Cloud Load Balancer** tab.
3. Click **Configure Listener** in the "Operation" column on the right of the target Classic CLB instance.
4. In the listener configuration module, click **Create**.
5. In the "Create Listener" pop-up window, enter the "backend port" (for more information on port selection, please see [Common Server Ports](#)) and other related fields and click **Next** to complete the configuration. For more information, please see [Configuring Classic CLB](#).

Note :

You need to specify the real server port for Classic CLB during listener creation.

### CreateListener

1 Basic Configuration

2 Advanced Configuration

3 Health Check

Name

test

Listen Protocol Ports*i*

TCP

:

22

Backend Port

8080

Close

Next

6. After the listener is created, click **Bind** in the real server binding module.

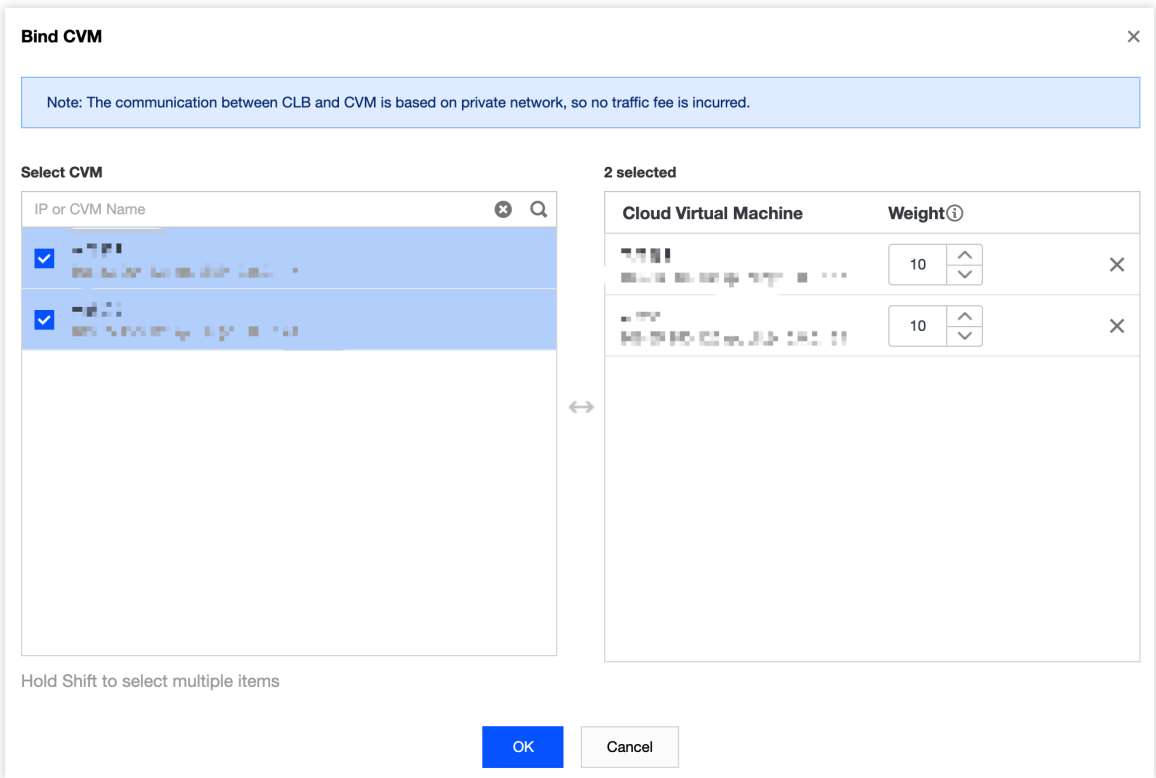
7. In the **Bind CVM** pop-up window, select the CVM instance to be bound, enter the weight, and click **OK**.

Note :

The pop-up window only displays available CVM instances in the same region and same network environment that are not isolated and have not expired with peak bandwidth greater than 0.

When multiple real servers are bound, CLB will forward traffic according to the hash algorithm to balance the load.

The greater the weight of a server, the more the requests forwarded to it. The default value is 10, and the configurable value range is 0–100. If the weight is set to 0, the server will not accept new requests. If session persistence is enabled, it may cause uneven request distribution among real servers. For more information, please see [Algorithms and Weight Configuration](#).



## Modifying real server weight for Classic CLB instance

Note :


Currently, real server weight cannot be modified through APIs for Classic CLB.

1. Log in to the [CLB Console](#).
2. On the "Instance Management" page, select the **Classic Cloud Load Balancer** tab.
3. Click **Configure Listener** in the "Operation" column on the right of the target Classic CLB instance.
4. In the real server binding module, modify the relevant server weight.

Note :

The greater the weight of a server, the more the requests forwarded to it. The default value is 10, and the configurable value range is 0–100. If the weight is set to 0, the server will not accept new requests. If session persistence is enabled, it may cause uneven request distribution among real servers. For more information, please see [Algorithms and Weight Configuration](#).

**Method 1.** Modify the weight of one single server.

- 4.1.1 Find the server whose weight needs to be modified, hover over the corresponding weight, and click .



Bind	Modify Weight	Unbind	IP or CVM Name				
<input type="checkbox"/>	ID	Name	Status	Private IP	Public IP	Weight ⓘ	Operation
<input type="checkbox"/>			Running			10	Unbind
<input type="checkbox"/>			Running			10	Unbind

4.1.2 In the "Modify Weight" pop-up window, enter the new weight value and click **Submit**.

**Method 2.** Modify the weight of multiple servers in batches.

Note :

After the batch modification, the servers will have the same weight.

4.1.3 Click the checkbox in front of the servers, select multiple servers, and click **Modify Weight** at the top of the list.

Bind	Modify Weight	Unbind	IP or CVM Name				
<input checked="" type="checkbox"/>	ID	Name	Status	Private IP	Public IP	Weight ⓘ	Operation
<input checked="" type="checkbox"/>			Running			10	Unbind
<input checked="" type="checkbox"/>			Running			10	Unbind

4.1.4 In the "Modify Weight" pop-up window, enter the new weight value and click **Submit**.

### Unbinding real server from Classic CLB instance

Note :

Unbinding a real server will unbind the Classic CLB instance from the CVM instance, and Classic CLB will stop forwarding requests to it immediately.

Unbinding a real server will not affect the lifecycle of your CVM instance, which can be added to the real server cluster again when necessary.

If you need to use API to unbind real servers, please see the [DeregisterTargetsFromClassicalLB](#) API.

1. Log in to the [CLB Console](#).
2. On the "Instance Management" page, select the **Classic Cloud Load Balancer** tab.
3. Click **Configure Listener** in the "Operation" column on the right of the target Classic CLB instance.
4. In the real server binding module, unbind the bound server.

**Method 1.** Unbind one single server.

4.1.1 Find the server that needs to be unbound and click **Unbind** in the **Operation** column on the right.

<div>BindModify WeightUnbind</div>							IP or CVM Name	<div>Q</div>	<div></div>
<input type="checkbox"/>	ID	Name	Status	Private IP	Public IP	Weight <i>i</i>	Operation		
<input type="checkbox"/>			Running			10	Unbind		
<input type="checkbox"/>			Running			10	Unbind		

4.1.2 In the "Unbind Real Server" pop-up window, confirm the server to be unbound and click **Submit**.

**Method 2.** Unbind multiple servers in batches.

4.1.3 Click the checkbox in front of the servers, select multiple servers, and click **Unbind** at the top of the list.

<div>BindModify WeightUnbind</div>							IP or CVM Name	<div>Q</div>	<div></div>
<input checked="" type="checkbox"/>	ID	Name	Status	Private IP	Public IP	Weight <i>i</i>	Operation		
<input checked="" type="checkbox"/>			Running			10	Unbind		
<input checked="" type="checkbox"/>			Running			10	Unbind		

4.1.4 In the "Unbind Real Server" pop-up window, confirm the servers to be unbound and click **Submit**.