Cloud Load Balance

Operation Guide

Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

Operation Guide
CLB instance
Directions for Upgrading to Domain Name-Based CLB

Last updated: 2023-09-04 19:19:02

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.

Comparison Before and After the Upgrade

Comparison Item	After Upgrade	Before Upgrade
SLA	99.99%	99.95%
Domain name supported	Supported	Not required
Automatic VIP scaling supported	This feature is supported.	Unavailable
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

Description

- Classic networks do not support the upgrade of instances. Please complete the migration first. For more information, refer to Migration Guide.
- Classic Cloud Load Balancer does not support upgrade. Upgrade classic CLB instances to CLB instances as instructed in Classic Instance Upgrade.
- Container-created Cloud Load Balancer instances are not currently supported for direct upgrade through the console. If you have upgrade requirements, please seek online support.
- Anti-DDoS Pro does not currently support protection for domain name-based Cloud Load Balancer instances. Upgrading to a domain name-based CLB instance may cause the loss of Anti-DDoS Pro protection, which could severely impact the security of your services. It is not recommended for users with public network CLB instances bound to Anti-DDoS Pro or those with Anti-DDoS requirements to upgrade to domain name-based CLB instances. For other issues, please seek online support.

Preparations

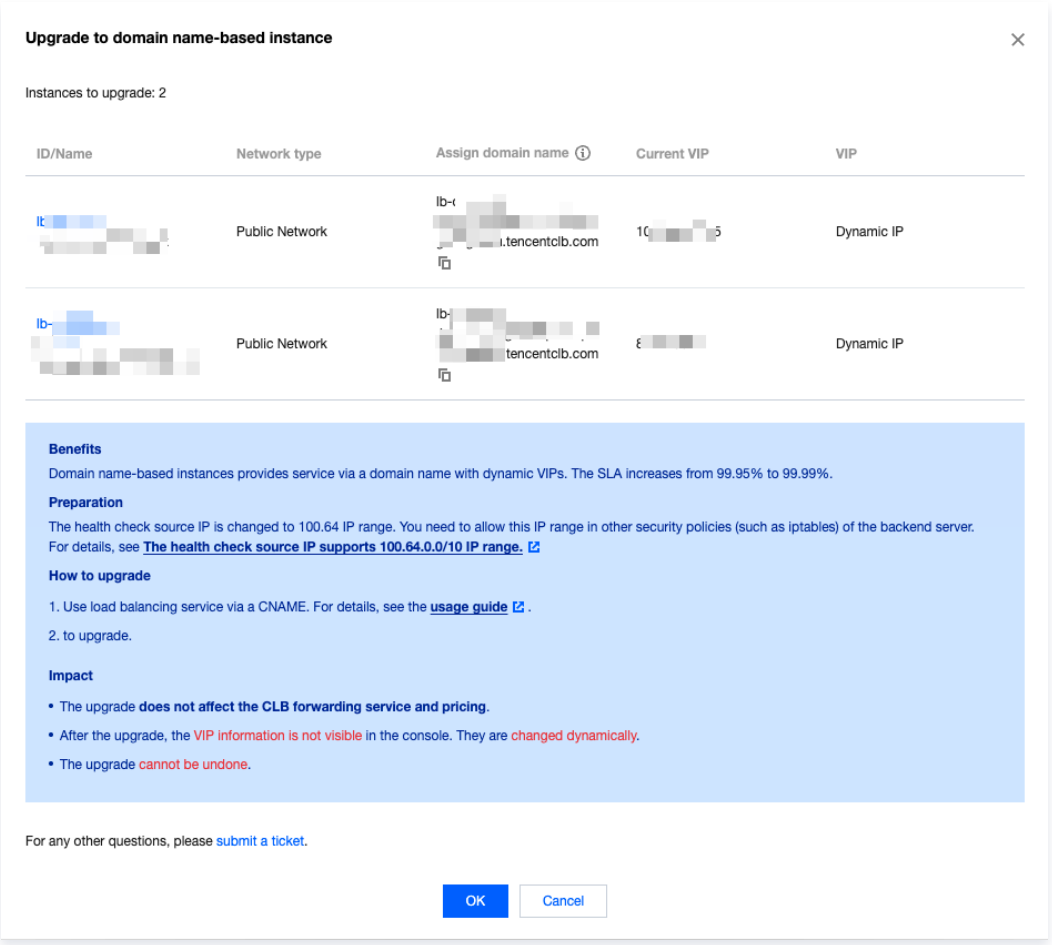
1. Customers provide external access using CNAME domain name resolution. For more information, refer to User Guide.
2. The health check source IP has been changed to the 100.64.0.0/10 IP range. For more information, see Health Check Source IP Diagnostic Assistant.

Instructions

Method 1: Upgrading a specific CLB instance

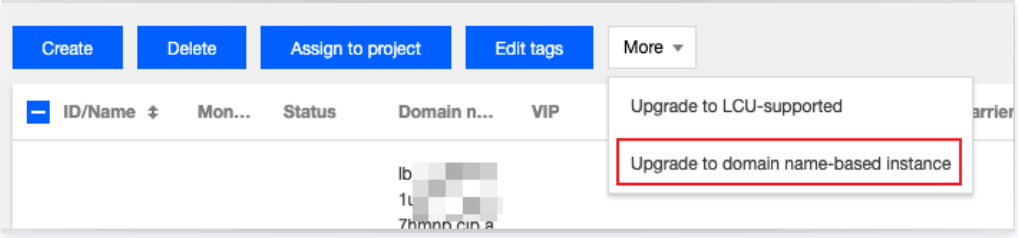
1. Log in to the Cloud Load Balancer console.
2. (Optional) Use the Health Check Source IP Diagnostic Assistant to confirm that the health check source IP of the instance to be upgraded is in the 100.64.0.0/10 IP range. For more information, see Health Check Source IP Diagnostic Assistant.
3. In the upper left corner of the **Instance Management** page, select the region. Locate the target instance in the instance list, then click on **More** > **Upgrade to Domain Name Instance** in the operation column on the right.
4. Record the **Assigned Domain Name** in the upgrade pop-up window, and CNAME your own domain name to the assigned domain name.

5. Click **Confirm** in the **Upgrade to Domain Name Instance** pop-up window to complete the upgrade.

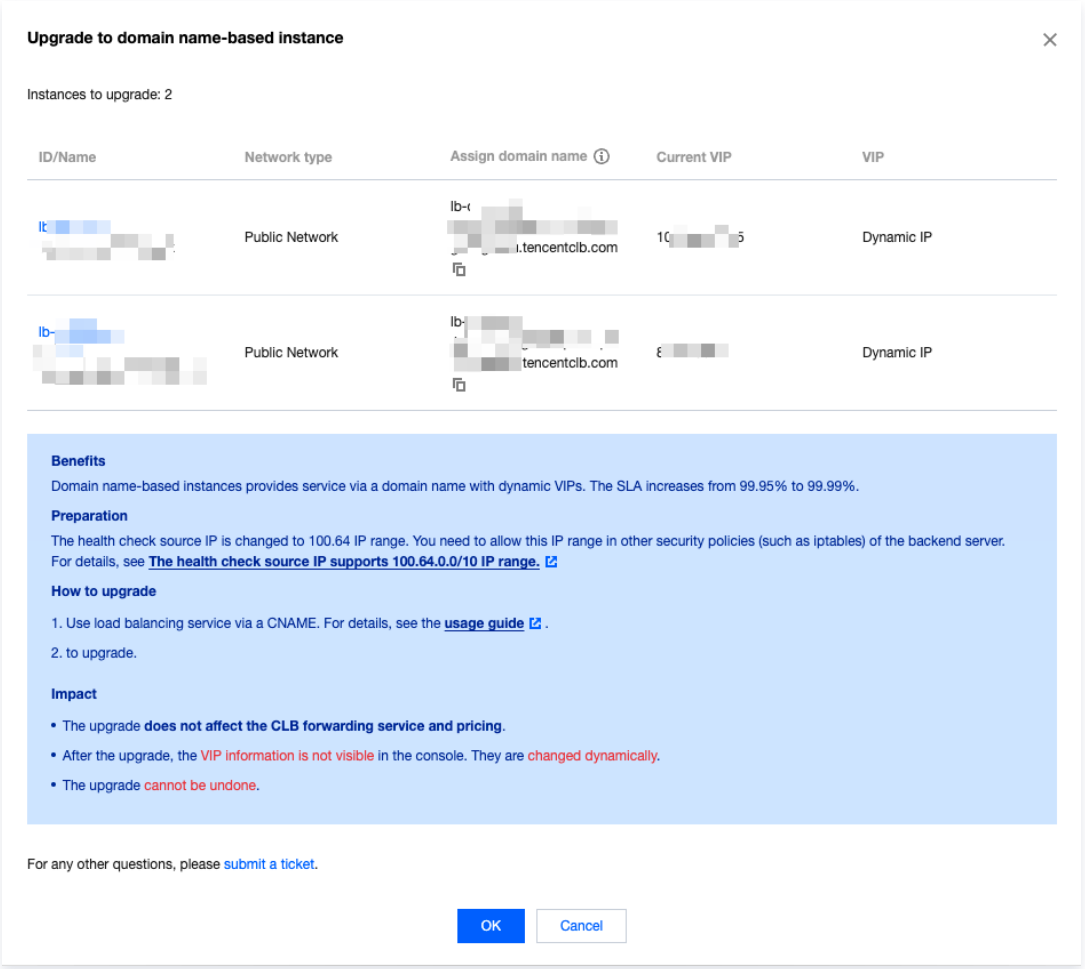


Method 2: Upgrading instances in batches

1. Log in to the Cloud Load Balancer console.
2. (Optional) Use the Health Check Source IP Diagnostic Assistant to confirm that the health check source IP of the instance to be upgraded is in the 100.64.0.0/10 IP range. For more information, see Health Check Source IP Diagnostic Assistant.
3. In the top left corner of the **Instance Management** page, select the region and check the boxes for the Cloud Load Balancer instances that have not been upgraded.
4. Above the instance list, select **More Actions** > **Upgrade to Domain Name-based Instance**.
5. Record the **Assigned Domain Name** in the upgrade pop-up window, and CNAME your own domain name to the assigned domain name.
6. Click **Confirm** in the **Upgrade to Domain Name Instance** pop-up window to complete the upgrade.



7. Click **OK** in the **Upgrade to Domain Name-based Instance** pop-up window.



Creating CLB Instances

Last updated: 2023-09-04 19:56:49

Tencent Cloud offers two methods for purchasing Cloud Load Balancer: through the official website and via API. This section will provide a detailed explanation of both purchasing options.

Purchasing a CLB instance on the official purchase page

All users can purchase a Cloud Load Balancer on the Tencent Cloud official website. Tencent Cloud accounts are divided into standard and traditional account types. Accounts registered after 00:00:00 on June 17, 2020, are of the standard account type. For accounts registered before this time, please check your account type in the console as instructed in Determining Account Type.

1. Log in to the Tencent Cloud console and go to the Cloud Load Balancer purchase page.
2. Select the following CLB configuration items as needed:

Standard Account Type

Category	Note
Billing	Both monthly subscription and pay-as-you-go billing modes are supported.
Regions	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	Supports two network types: public network and private network. For more information, see Network Types. • Public network: CLB is used to distribute requests from the 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 the monthly subscription mode, only the public network type is supported. • In pay-as-you-go billing mode, both the public and private network types are supported.
Elastic IP (EIP)	• If EIP is not selected, Tencent Cloud will assign you a public network CLB instance whose public IP address cannot be changed. (By default, only non-EIP selection is supported for public network CLB instances with monthly or yearly subscription plans.) • 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 testing. To participate, please submit a beta application. For usage restrictions, see Usage 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	CLB supports two types of networks: Virtual Private Cloud (VPC) and Classic Network. • By contrast, the VPC is a logically isolated network space in Tencent Cloud. In a VPC, you can customize IP ranges, IP addresses, and routing policies. • The classic network is a public network resource pool shared by all Tencent Cloud users. The private IPs of all CVM instances are assigned by Tencent Cloud. You cannot customize IP ranges or IP addresses. Compared to the classic network, a Virtual Private Cloud (VPC) is more suitable for scenarios requiring custom network configurations. Additionally, the classic network products will be officially discontinued on December 31, 2022. For more information, please refer to Classic Network End-of-Support Notice. We recommend choosing a VPC.

ISP Type	Supports the following ISP types: BGP (multi-line), China Mobile, China Telecom, and China Unicom. • In the pay-as-you-go mode, only the multi-line BGP ISP type is supported, and this configuration item is not displayed by default. • 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.
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. Currently, only IPv4 CLB instances in the Guangzhou, Shanghai, Nanjing, Beijing, Hong Kong (China), and Seoul regions support primary/secondary AZs.
Instance Specification	Shared and LCU-supported instances are supported. • Shared instances provide performance guarantees based on their specifications. A single shared instance can support up to 50,000 concurrent connections, 5,000 new connections per second, and 5,000 queries per second (QPS). • Performance-guaranteed instances provide performance assurance based on their specifications. A single instance can support up to 1 million concurrent connections, 100,000 new connections per second, and 50,000 queries per second (QPS). If you require a larger specification, you can submit a ticket to request it.
Network Billing Mode	The following network billing modes are supported: bill-by-bandwidth (monthly subscription and hourly bandwidth), bill-by-traffic, and bandwidth package. • For instances with a Monthly Subscription billing mode, only the bandwidth-based billing (monthly bandwidth) network billing mode is supported. • A pay-as-you-go instance supports three network billing modes: bill-by-bandwidth (hourly bandwidth), bill-by-traffic, and bandwidth package. Currently, the bandwidth package billing mode is in beta. To use it, submit a beta application.
Bandwidth cap	• The bandwidth cap for shared public network CLB instances is 2 Gbps, while the recommended bandwidth cap for shared private network CLB instances should not exceed 5 Gbps. • The bandwidth cap for LCU-supported CLB instances depends on the selected specifications. For more information, please refer to Instance Specification Comparison.
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

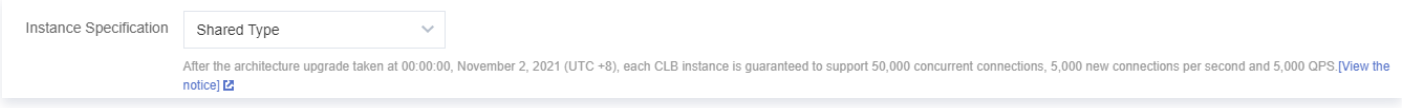
Category	Note
Billing	Supports pay-as-you-go billing only.
Regions	Select a region. For more information on the regions supported by CLB, see Region List.
Instance Type	Supports the CLB instance type only. Starting from October 20, 2021, classic CLB instances can no longer be purchased. For more information, see Classic CLB End-of-Sale Notice.
Network	Supports two network types: public network and private network. For more information, see Network Types. • Public network: CLB is used to distribute requests from the 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 two types of networks: Virtual Private Cloud (VPC) and Classic Network. • By contrast, the VPC is a logically isolated network space in Tencent Cloud. In a VPC, you can customize IP ranges, IP addresses, and routing policies. • The classic network is a public network resource pool shared by all Tencent Cloud users. The private IPs of all CVM instances are assigned by Tencent Cloud. You cannot customize IP ranges or IP addresses. Compared to the classic network, a Virtual Private Cloud (VPC) is more suitable for scenarios requiring custom network configurations. Additionally, the classic network products will be officially discontinued on December 31, 2022. For more information, please refer to Classic Network End-of-Support Notice. We recommend choosing a VPC.
ISP Type	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 testing. To try it out, please submit a beta application. 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.
Instance Specification	Shared and LCU-supported instances are supported. • Shared instances provide performance guarantees based on their specifications. A single shared instance can support up to 50,000 concurrent connections, 5,000 new connections per second, and 5,000 queries per second (QPS). • Performance-guaranteed instances provide performance assurance based on their specifications. A single instance can support up to 1 million concurrent connections, 100,000 new connections per second, and 50,000 queries per second (QPS). If you require a larger specification, you can submit a ticket to request it.
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 purchase duration (supported only for the monthly/yearly subscription mode), quantity, and fees, then click **Buy Now**.
   - For the Monthly Subscription Plan: Proceed to the confirmation page, and if you have a product voucher, select the option to use it. Verify the information is correct and click **Submit Order**. Choose your preferred payment method on the payment page and complete the transaction.
   - For the Pay-as-you-go model: In the pop-up "Load Balancer Order Confirmation" dialog box, click **Confirm Order**.
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 Cloud Load Balancer purchase page.
2. Refer to the above Official Website Purchase operation steps, select the shared load balancing instance configuration as needed, and choose **Shared** under "Instance Specifications".

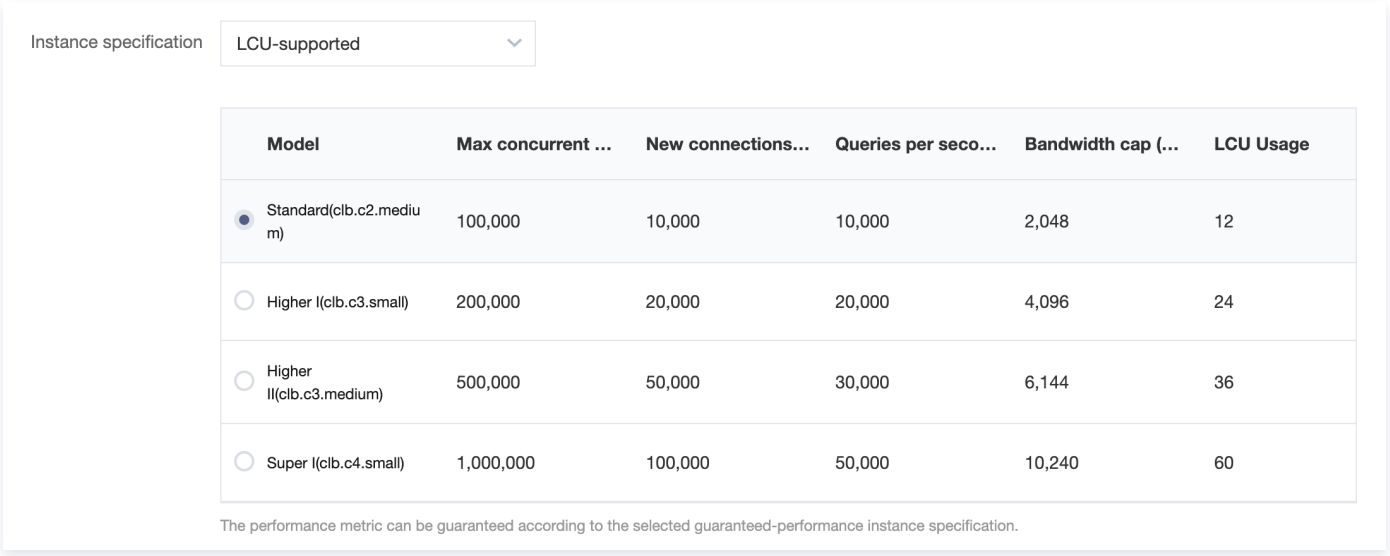


3. Complete the subsequent operations by referring to the steps in Purchasing a CLB Instance on the Official Purchase Page.

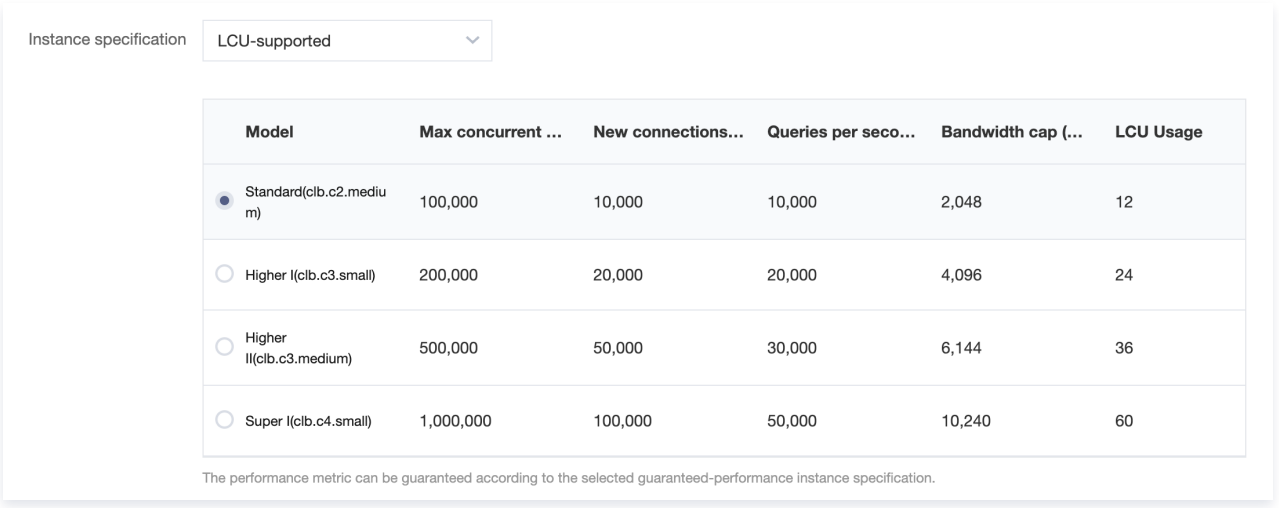
Purchasing an LCU-supported instance

1. Log in to the Tencent Cloud console and go to the Cloud Load Balancer purchase page.
2. Refer to the steps in Purchasing a CLB Instance on the Official Purchase Page, select the LCU-supported instance configuration items as needed, and choose **LCU-supported** for "Instance specification".

Monthly subscribed LCU-supported instances



Pay-as-you-go LCU-supported instances



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 Cloud Load Balancer instance via an API, see CreateLoadBalancer.

See Also

- To create a listener for a Cloud Load Balancer, please refer to CLB Listener Overview.
- To bind a CLB listener to a backend service, please refer to Backend Server Overview.

Documentation

Product Attribute Selection

Creating an IPv6 CLB Instance

Last updated: 2023-09-04 20:40:51

Note:

- The IPv6 Cloud Load Balancer is in beta testing. To use it, please submit a ticket.
- Currently, IPv6 Cloud Load Balancer is supported only in the following regions: Guangzhou, Shenzhen Finance, Shanghai, Shanghai Finance, Nanjing, Beijing, Beijing Finance, Chengdu, Chongqing, Hong Kong (China), Singapore, and Virginia. For the compliance zones tailored to the financial industry regulatory requirements in Shenzhen Finance and Shanghai Finance regions, you need to submit a ticket to apply for using the dedicated zone.
- IPv6 CLB does not support classic CLB.
- IPv6 CLB supports obtaining the client's IPv6 source address, which can be directly obtained by layer-4 IPv6 CLB or through the X-Forwarded-For header of HTTP layer-7 IPv6 CLB.
- Currently, IPv6 CLB balances the load completely over the public network. Clients in the same VPC cannot access IPv6 CLB over the private network.
- IPv6 implementations are still at the preliminary stage across the internet. In case of access failure, please submit a ticket for feedback.

Overview

IPv6 CLB is implemented based on the IPv6 single-stack technology and can collaborate with IPv4 CLB to enable IPv6/IPv4 dual-stack communication. An IPv6 CLB instance is bound to the IPv6 address of a Cloud Virtual Machine and provides an external IPv6 VIP address.

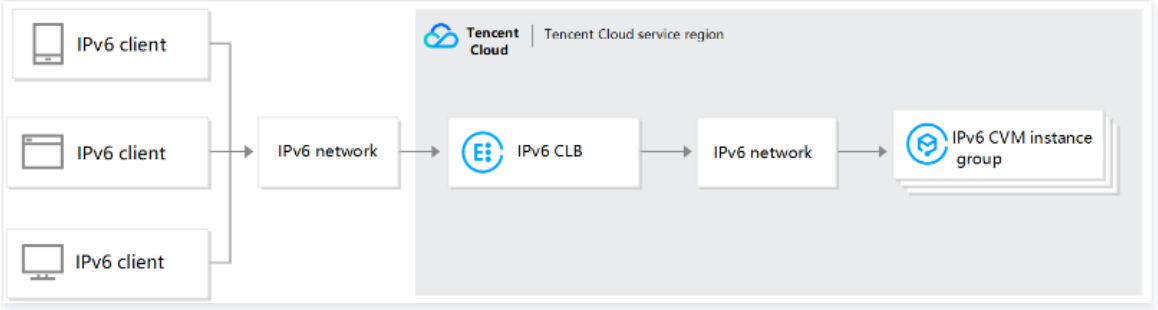
IPv6 CLB Advantages

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:

Standard Account Type

Parameter	Description
Billing Method	Supports both monthly subscription and pay-as-you-go billing modes. 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	Supports two network types: public network and private network. For more information, see Network Types. Select the public network type for IPv6 CLB.
Elastic IP (EIP)	Don't select an EIP.
IP Version	Select the IPv6 version.
Network	Select an existing VPC or subnet. If the current networks are unsuitable, you can create a new VPC or create a new subnet.
ISP Type	Select **Multi-line BGP**.
Metric quota	Shared and LCU-supported instances are supported. • Shared instances provide performance guarantees based on their specifications. A single shared instance can support up to 50,000 concurrent connections, 5,000 new connections per second, and 5,000 queries per second (QPS). • Performance-guaranteed instances provide performance assurance based on their specifications. A single instance can support up to 1 million concurrent connections, 100,000 new connections per second, and 50,000 queries per second (QPS). If you require a larger specification, you can submit a ticket to request it.
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.
Bandwidth cap	• The bandwidth cap for shared public network CLB instances is 2 Gbps, while the recommended bandwidth cap for shared private network CLB instances should not exceed 5 Gbps. • The bandwidth cap for LCU-supported CLB instances depends on the selected specifications. For more information, please refer to Instance Specification Comparison.
Projects	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 Protocol	I've read and agreed to Tencent Cloud Terms of Service and CLB Service Level Agreement.

Bill-by-CVM account

Parameter	Description
Billing Method	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	Supports two network types: public network and private network. For more information, see Network Types. • Public network: CLB is used to distribute requests from the 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	Select IPv6. For more information on the use limits, see IP Versions.
Network	CLB supports classic network and VPC. • The classic network is a public network resource pool shared by all Tencent Cloud users. The private IPs of all CVMs are assigned by Tencent Cloud. You cannot customize IP ranges or IP addresses. • By contrast, the VPC is a logically isolated network space in Tencent Cloud. In a VPC, you can customize IP ranges, IP addresses, and routing policies. Compared to the classic network, a Virtual Private Cloud (VPC) is more suitable for scenarios requiring custom network configurations. Additionally, the classic network products will be officially discontinued on December 31, 2022. For more information, please refer to Classic Network End-of-Support Notice. We recommend choosing a VPC.
ISP Type	Select **Multi-line BGP**.
Metric quota	Shared and LCU-supported instances are supported. • Multiple shared instances share resources, and a single instance does not provide guaranteed performance. By default, all instances are shared instances. • An LCU-supported instance guarantees the performance and does not preempt resources like a shared instance. Its forwarding performance is not affected by other instances. A single instance can sustain up to 1 million concurrent connections, 100,000 new connections per second, and 50,000 queries per second.
Network Billing Mode	Bill by bandwidth.
Bandwidth cap	1-1024Mbps。
Projects	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 Protocol	I've read and agreed to Tencent Cloud Terms of Service and CLB Service Level Agreement.

3. After selecting the desired configurations on the purchase page, click **Buy Now**. In the "Cloud Load Balance Order Confirmation" pop-up window, click **Confirm Order**. Return to the Cloud Load Balance Instance List page to view the purchased IPv6 Cloud Load Balance.

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

3. Configure the name, listening protocol port, and load balancing method in "Basic Settings," then click **Next**.
4. Configure health check and click **Next**.
5. Configure session persistence and click **Submit**.
6. After the listener is created, select it and click **Bind** on the right.

> **Note:**
> Before binding the listener to a CVM instance, please make sure that the CVM instance has obtained an IPv6 address.

7. In the pop-up window, select the target IPv6 Cloud Virtual Machine, configure the service ports and their weights, and click **OK**.

More Operations

Binding IPv6 CLB with both IPv6 and IPv4 real servers

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

1. Enable dual-stack binding.
   - Enable dual-stack binding when purchasing IPv6 CLB on the purchase page.
   - Enable dual-stack binding on the IPv6 CLB instance details page.
2. Create a layer-7 HTTP or HTTPS listener.
3. Bind the listener with a IPv6 or IPv4 backend server.

Relevant Document

Setting up IPv6 Virtual Private Cloud

Creating IPv6 NAT64 CLB Instances

Last updated: 2023-09-04 20:44:23

Note:
- IPv6 NAT64 CLB can only be created in three regions: Beijing, Shanghai, and Guangzhou.
- IPv6 NAT64 CLB does not support classic CLB.
- 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.

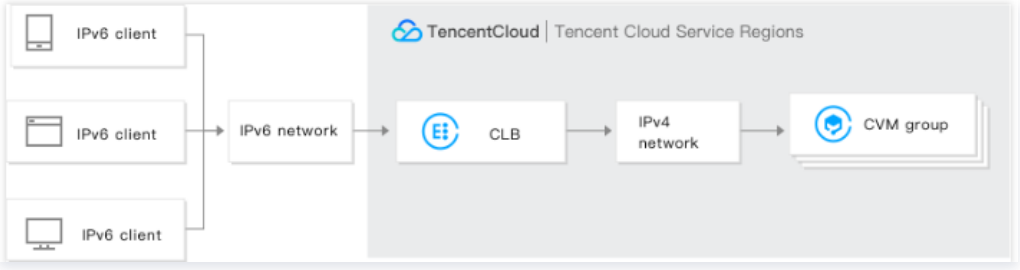
CLB supports creating IPv6 NAT64 CLB instances. Tencent Cloud will assign an IPv6 public IP address, i.e., VIP of the IPv6 edition, to an instance, and the VIP will forward requests from IPv6 clients to the real IPv4 CVM instance.

What is an IPv6 NAT64 Cloud Load Balancer?

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 seamlessly integrate IPv6, businesses only need to modify the client without altering backend services. IPv6 NAT64 CLB supports access from IPv6 clients and converts IPv6 packets into IPv4 packets. Applications on real CVM instances remain unaware of IPv6 and continue to operate in their original form.
- **Easy to use:** IPv6 NAT64 CLB is compatible with the original IPv4 CLB workflow, offering a seamless experience with no learning curve and low entry barriers.

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: Both monthly subscription and pay-as-you-go billing methods are supported.
   - Region: Only Beijing, Shanghai, and Guangzhou are supported.
   - Instance Type: Cloud Load Balancer.
   - Network Type: Public network.
   - IP Version: IPv6 NAT64.
   - Network: Virtual Private Cloud (VPC).
   - Other configurations are the same as general instance configurations.

3. After selecting the desired configurations, click **Buy Now** and return to the CLB Instance Management page to view the purchased IPv6 NAT64 CLB.

Using IPv6 NAT64 CLB

Log in to the CLB Console, click on the instance ID to access the details page, and configure listeners, forwarding rules, and bind CVMs in the "Listener Management" page. For more information, please refer to CLB Quick Start Guide.

Relevant Document

Obtaining Real Client IPs via TOA in Hybrid Cloud Deployment

Creating an Anycast CLB Instance

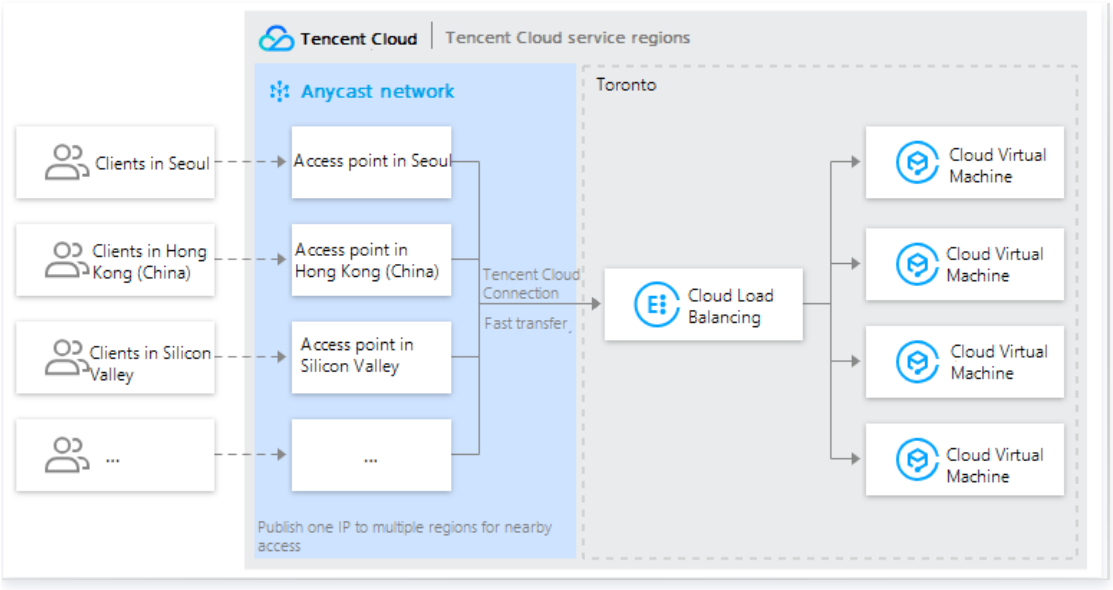
Last updated: 2023-09-04 20:50:28

Anycast CLB is a load balancing service that supports multi-region dynamic acceleration. An Anycast CLB VIP is published in multiple regions. Each client accesses the nearest POP, and the access traffic to different POPs is forwarded to CVM instances through the high-speed internet of Tencent Cloud IDC.
Anycast CLB can optimize network transfer and achieve multi-entry nearby access, reducing network jitter and packet loss. It improves the service quality of applications on the cloud, expands the service scope, and streamlines backend deployment.

> **Note:**
> This feature is currently in beta. To try it out, please submit a beta application.

Anycast CLB architecture

Anycast CLB VIPs are published in multiple regions, and clients connect to the nearest POP access point. The access traffic is rapidly forwarded to Cloud Virtual Machines through Tencent Cloud's internal network. For supported regions, please refer to Anycast Supported Regions.



Limits

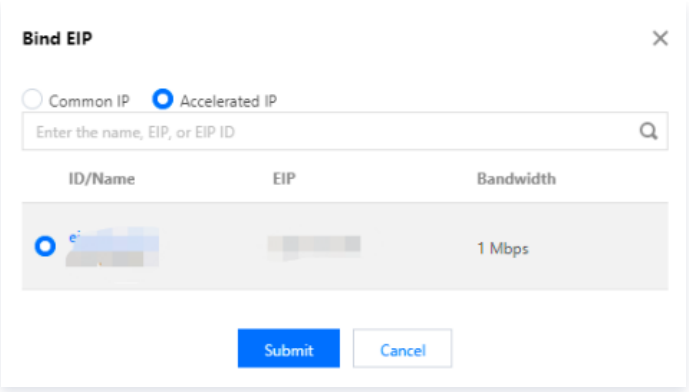
Anycast CLB is provided by binding an Anycast EIP to a private network Cloud Load Balancer, offering Anycast capabilities. For limitations, please refer to Restrictions on Binding EIP to Private Network CLB.

Prerequisites

This feature is in beta testing. Before proceeding, please ensure that your beta application has been approved.

Operation Guide

1. Log in to the Public IP Console, select the region in the upper left corner of the "Public IP" page, and click **Apply**.
2. In the pop-up "Apply for EIP" dialog box, select **Acceleration IP** for the IP address type, set the bandwidth cap, check "**I agree to the Tencent Cloud EIP Service Agreement and Arrears Rules**", and click **OK**.
3. Log in to the Cloud Load Balance console, select a region in the top-left corner of the "Instance Management" page, choose the target private network Cloud Load Balance instance from the instance list, and click **More** > **Bind EIP** in the operation column.
4. In the pop-up "Bind EIP" dialog box, select the acceleration IP created earlier and click **Submit**.



5. After binding an Anycast Accelerated IP to a private network Cloud Load Balancer, it can provide Anycast Cloud Load Balancing services. For more information on Cloud Load Balancer configuration, please refer to Cloud Load Balancer Listener Overview.

Relevant Document

- Anycast Internet Acceleration
- Binding Private Network Cloud Load Balancer to EIP

Configure the forwarding domain name for the load balancer

Last updated: 2023-09-04 21:03:53

When a client initiates a request, the Cloud Load Balancer forwards the request to the real servers based on the configured listener forwarding rules. The domain name in the listener forwarding rules corresponds to the domain name used by your backend services. This document explains how to configure the domain name.

Instructions

Step 1: Register a domain name

Domain name registration is the prerequisite for building a service on the Internet.

- If you have already registered a domain name with another registrar, you can transfer it to Tencent Cloud domain service. For more information, see Domain Transfer In.
- If you do not have a domain name, you must register a domain name first. For more information, see Domain Registration.

Step 2: Add a domain name resolution

After registering a domain name, you can add a CNAME record for the domain name so that the domain name can be used to access your website.

1. Log in to the DNSPod console. In the **Domain Name List** page, click **DNS** in the **Operation** column of the target domain name. This document uses the `example.com` domain as an example.
2. On the **Record Management** tab, click **Add Record**.
3. In the **Add Record** section, set the following parameters:
   3.1 Fill in the host record as needed. The host record is the domain name prefix. For more information, see Subdomain Explanation and Wildcard Resolution Explanation. Common use cases include:
   - www: The resolved domain name is `www.example.com`.
   - @: Directly resolve the primary domain name `example.com`.
   - *: Wildcard resolution, matching all other domain names, such as `*.example.com`.

   3.2 Select **Record Type**, it is recommended to choose `CNAME`.
   3.3 Line Type: Choose the "Default" type, otherwise, it may cause resolution issues for some users. For example, if you want to direct China Unicom users to `2.com` and all non-China Unicom users to `1.com`, you can achieve this by adding two CNAME records with the line type set to Default and record value as `1.com`, and another with the line type set to China Unicom and record value as `2.com`.
   3.4 **Record Value**: You can enter the domain name allocated by CLB.
   3.5 Retain the default values for other parameters and click **Save**.
4. After adding the record, you can view the record in the record list on the **Record Management** tab.

Step 3: Verify the resolution result

> **Note:**
> It takes approximately ten minutes for the DNS resolution to take effect.

After completing the above steps, you can enter the CNAME domain name with added domain resolution (such as `www.example.com` in this example) in your browser to test if the domain name resolution is working properly.

Configuring CLB Security Group

Last updated: 2023-09-05 20:06:06

After creating a Cloud Load Balancer (CLB), you can configure its security group to isolate public network traffic. This article will explain how to configure security groups for different CLB modes.

Limit

- Each CLB can be bound to up to 5 security groups. If you need to increase the quota, please go to Quota Management and submit a quota application.
- Each security group for CLB can have up to 512 rules.
- Classic private network Cloud Load Balancer and private network Cloud Load Balancer in the basic network do not support binding security groups. When a private network Cloud Load Balancer is bound to an Anycast EIP, the security group bound to the private network Cloud Load Balancer is temporarily ineffective.
- Classic private network Cloud Load Balancer and Cloud Load Balancer in the basic network do not support the security group default pass-through feature. Bare Metal Cloud Virtual Machine currently does not support the security group default pass-through capability.

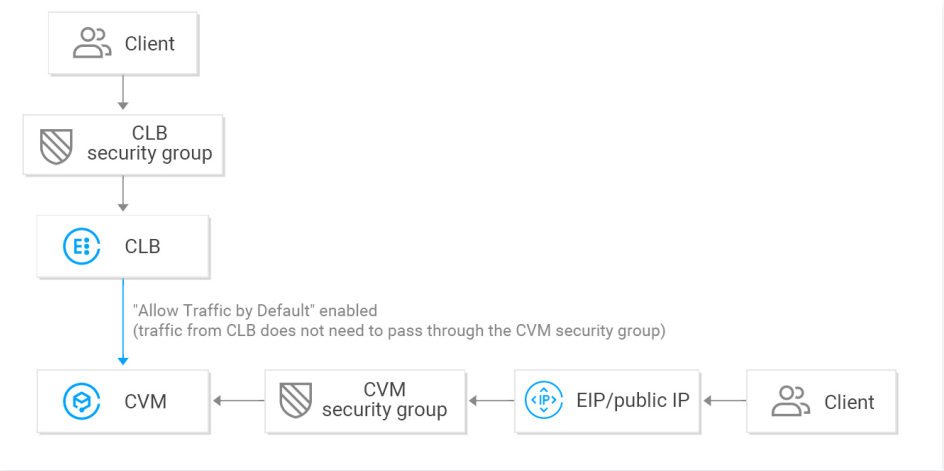
Background Information

A security group is a virtual firewall with stateful data packet filtering capabilities, controlling the inbound and outbound traffic at the instance level. For more information, please refer to Security Group Overview.
CLB security groups are bound to CLB instances, while CVM security groups are bound to CVM instances, with different objects being restricted. There are mainly two modes for configuring CLB security groups:

- Enable Default Security Group Allowance
- Disable Allow by Default in Security Group

> **Note:**
> - For IPv4 CLB security groups, **Allow by Default** is disabled by default, you can enable it in the console.
> - For IPv6 CLB security groups, **Allow by Default** is enabled by default and you cannot disable it.

CLB Security Group Configuration

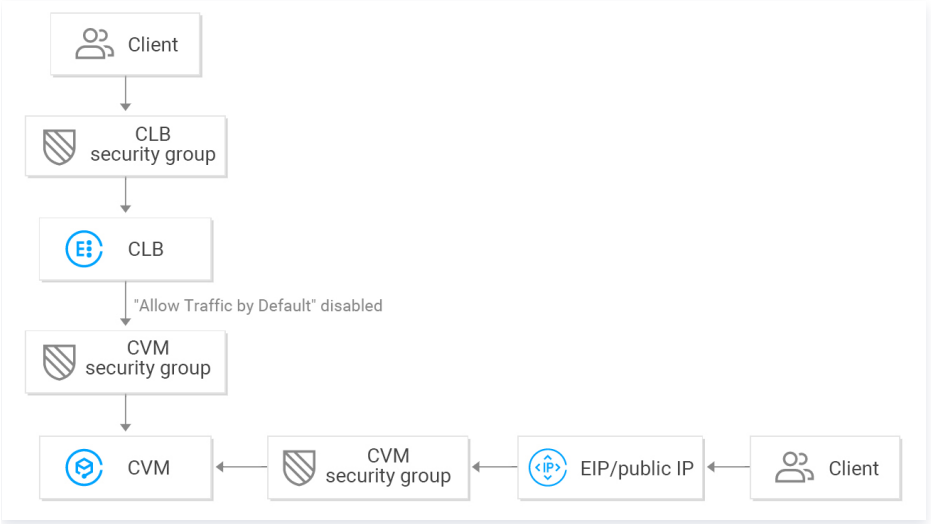


When **Allow by Default** is enabled:

- If you wish to allow access from a specific fixed client IP, the CLB security group needs to allow the client IP and listening port, while the backend CVM security group does not need to allow the client IP and service port. Access traffic from the CLB only needs to pass through the CLB security group, as the backend Cloud Virtual Machine will allow traffic from the CLB by default, without the need to expose any ports externally.
- 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.

Disable "Allow Traffic by Default" in Security Group



When **Allow by Default** is disabled:

- If you want to allow access from a specific fixed client IP, the CLB security group needs to allow the client IP and listening port, and the backend CVM security group also needs to allow the client IP and service port. In other words, the traffic passing through the CLB will undergo a double check by both the CLB security group and the 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 **Allow by Default** is disabled, the CVM security group should be configured as follows to ensure effective health check:

1. Configuring Public Network Cloud Load Balancer

   You need to open the CLB VIP to the internet on the backend CVM security group, so that CLB can use the VIP to detect the backend CVM health status.
2. Configuring Private Network Cloud Load Balancer
   - For private network Cloud Load Balancer (formerly "Application-based private network Cloud Load Balancer"), if your CLB belongs to a VPC network, you need to allow the CLB VIP (used for health checks) on the backend CVM security group; if your CLB belongs to the basic network, there is no need to configure the backend CVM security group, as the health check IP is allowed by default.
   - For classic private network Cloud Load Balancer, if the instance was created before December 5, 2016, and the network type is VPC, you need to allow the CLB VIP (used for health checks) on the backend CVM security group. For other types of classic private network CLB, there is no need to configure the backend CVM security group, 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:**
> In this example, a public network CLB is used, and the CLB VIP needs to be allowed on the backend CVM security group for health checks. Currently, `0.0.0.0/0` represents any IP, which already includes the CLB VIP.

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

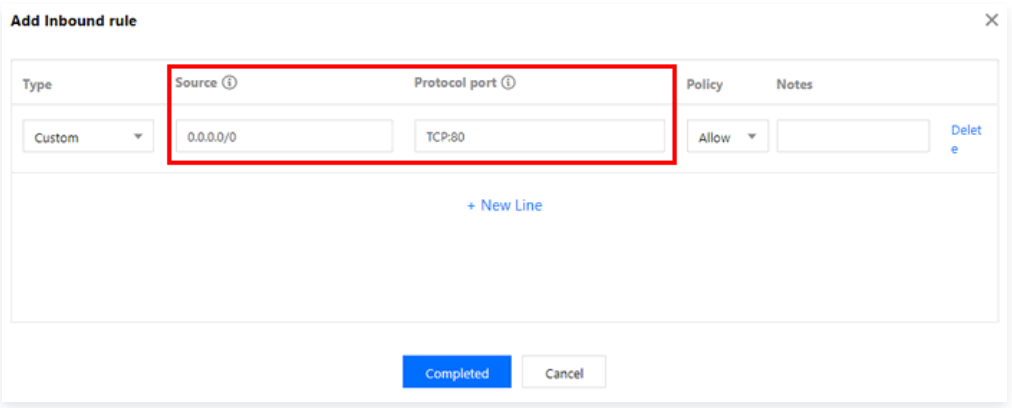
For more information, see Getting Started with Cloud Load Balancer. In this example, an HTTP:80 listener is created and a backend CVM is bound, with the service port of the backend CVM set to 8080.

Step 2. Configure a CLB security group

1. Configuring Cloud Load Balancer Security Group Rules

   Configure security group rules on the Security Group Console. In the inbound rules, allow all IPs (i.e., `0.0.0.0/0`) for port 80 and deny traffic on other ports.

> **Note**
> - Security group rules take effect sequentially from top to bottom. Once a previously set allow rule is applied, other rules are rejected by default. Please pay attention to the configuration order. For more information, see Security Group Rules.
> - Security groups have inbound and outbound rules. The above configuration limits apply to inbound traffic, so all configurations are for **inbound rules**. No special configuration is required for outbound rules.



2. Bind the security group to the CLB instance

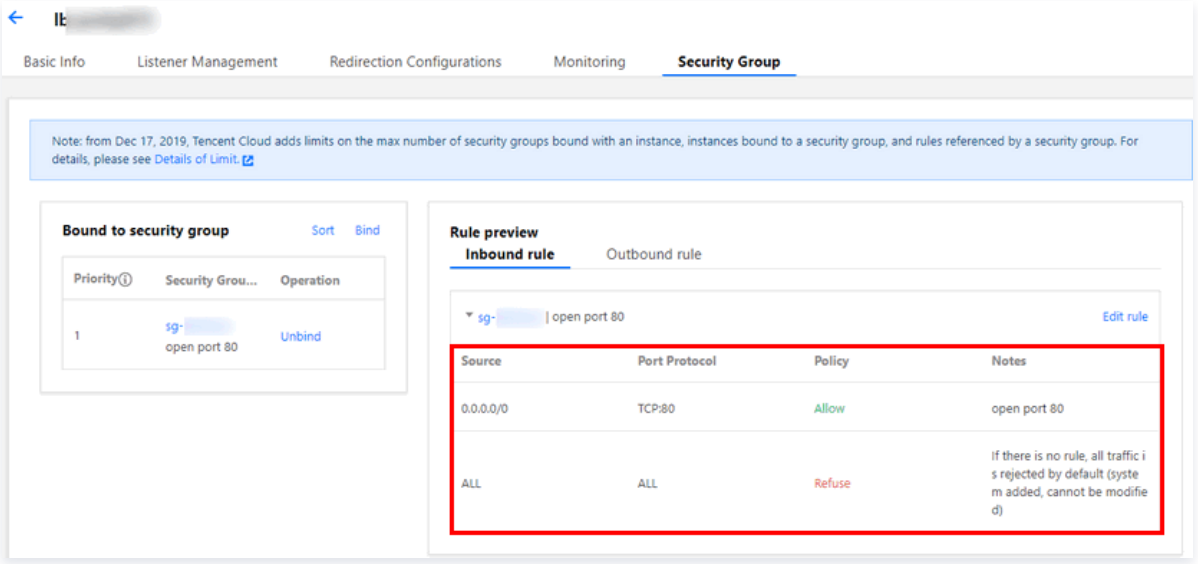
   2.1 Log in to the Cloud Load Balancer console.

   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 in the "Bound Security Groups" module, click **Bind**.

   2.4 In the "Configure Security Group" pop-up window, select the corresponding security group to be bound to the CLB and click **Confirm**.

   The CLB security group configuration is complete, and only traffic on port 80 is allowed to access the CLB.



Step 3. Configure Allow by Default

You can choose to enable or disable **Allow by Default** with different configurations as follows:

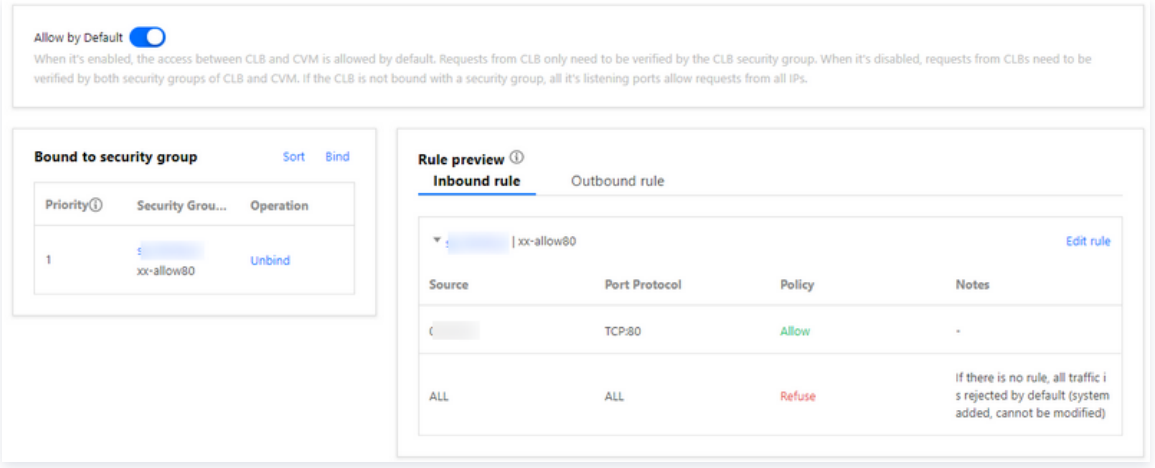
- Method 1: Enable **Allow by Default** for the security group, so the backend Cloud Virtual Machine does not need to expose ports to the Internet.

> **Note:**
> This feature is not supported for classic private network CLB and CLB in the classic network.

- Method 2: Disable **Allow by Default**, and open Client IP (in this case, 0.0.0.0/0) in the CVM security group as well.

Method 1. Enable Allow by Default

1. Log in to the Cloud Load Balancer 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" page, click to enable "Allow by Default."
5. After enabling the default pass-through function, only the security group rules in the **Preview Rules** will be verified.



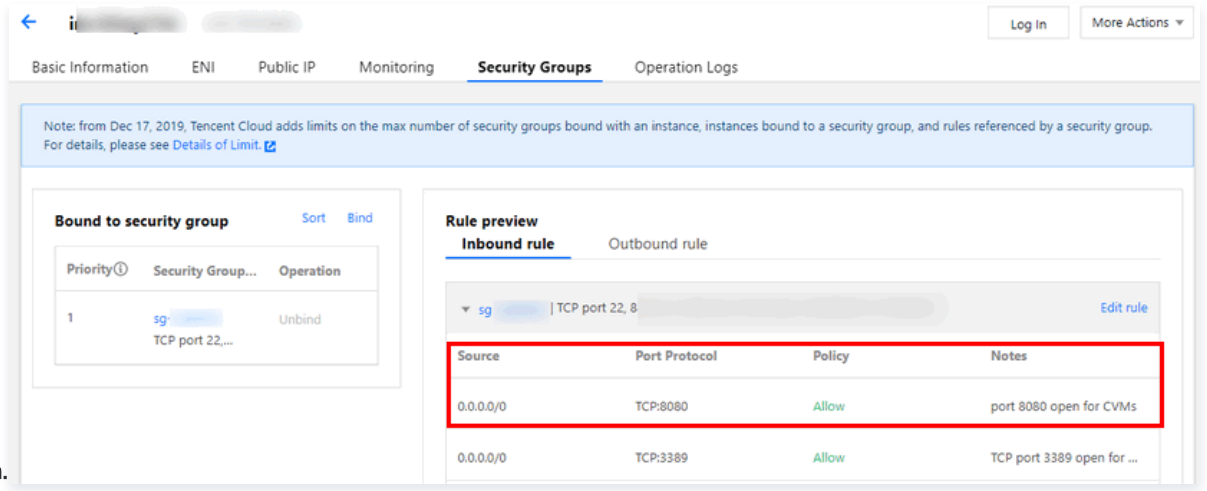
Method 2. Disable Allow by Default

When "Allow Traffic by Default" is disabled, you also need to allow the client IP in the CVM security group. For business traffic accessing CVM through CLB, only allow inbound traffic from CLB's port 80, and the service is provided via CVM's port 8080.

> **Note:**
> To allow traffic from a specified client IP, you need o 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 for the backend CVM to only allow traffic to access a specified service port.
   Configure the security group policy on the Security Group Console. Open port 8080 for all IPs in the inbound rules. To ensure remote login to the host and Ping service, open ports 22, 3389, and ICMP service in the security group.
2. Bind the security group to the CVM instance
   2.1 On the Cloud Virtual Machine Console, click the ID of the CVM bound to the CLB 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" pop-up window, select the corresponding security group bound to the CVM, and click **Confirm**.



Binding Private Network CLB to EIP

Last updated: 2023-09-04 21:23:58

Private Network Cloud Load Balancer is utilized for distributing requests originating from Tencent Cloud's private network, lacking a public IP and incapable of communicating with the public network. If you require using a Private Network Cloud Load Balancer and establish communication with the public network, you may opt to bind the Private Network Cloud Load Balancer to an Elastic Public IP, enabling public network access via the Elastic Public IP.

Note:
The feature of binding an Elastic Public IP to a Private Network CLB is in beta testing. To use it, please submit a beta request.

Limit

- **Region**
  - 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.
- **Functional limitations**
  - 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 instance is bound to an EIP, the security group of the instance takes effect for traffic from the instance but not from the EIP.
  - After a private network CLB instance is bound to an EIP and the "Allow by Default" feature is enabled in the security group of a backend CVM instance, the security group allows traffic from both the EIP and the CLB instance by default. That is, the security group does not take effect for both types of traffic. Therefore, we recommend not enabling the "Allow by Default" feature.

Directions

Method 1: Selecting an EIP when purchasing a CLB instance

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

Parameter	Description
Billing Method	Select the **Pay-as-You-Go** mode.
Region	Select a region. For more information on the regions supported by CLB, see Region List.
Instance type.	Only CLB instance type is supported.
Network	Select the **Public Network** type.
Elastic IP (EIP)	Select an Elastic Public IP, and Tencent Cloud will allocate an Elastic Public IP and a Private Network CLB for you. Supported Elastic Public IP 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 Cloud Load Balancer console and click Instance management in the left sidebar.
2. In the upper left corner of the "Instance Management" page, select the region, then choose the target Private Network CLB instance from the instance list. In the "Actions" column on the right, select **More** > **Bind Elastic Public IP**.
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 currently in beta testing. To use them, please submit Anycast Internet Acceleration IP beta request and static single-line IP beta request.

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.

Relevant Document

- AssociateAddress API Documentation
- Purchase Method
- Product Attribute Selection

Enabling or Disabling a CLB Instance

Last updated: 2023-09-04 21:26:06

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:**
> This feature is in beta testing. To try it out, please contact online support.

Application Scenarios

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. Once all listeners of an instance are started, the entire instance will be activated.
- After an instance is stopped, if any of its listeners are started, the instance will be started and receive and forward traffic normally with the started listener, while other listeners will remain stopped.

Limits

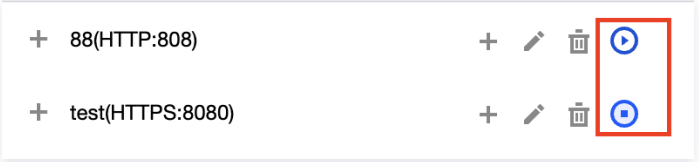
- Classic Cloud Load Balance types are not supported.
- This feature is supported only by VPC but not by classic networks.
- This feature is not supported for TLS 1.3 and earlier.

Prerequisites

- You have created a Cloud Load Balance 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 Cloud Load Balancer console.
2. In the upper left corner of the **Instance Management** page, select the region, locate the target instance in the list, and click **More** > **Start** or **More** > **Stop** in the operation column on the right.
3. (Optional) On the **Listener Management** tab, find the target listener and click **Start listener** or **Stop listener**.



Cloning CLB instances

Last updated: 2023-09-04 21:26:14

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.

> **Note:**
> The cloning feature is currently in beta testing. To try it out, please submit a request for beta access.

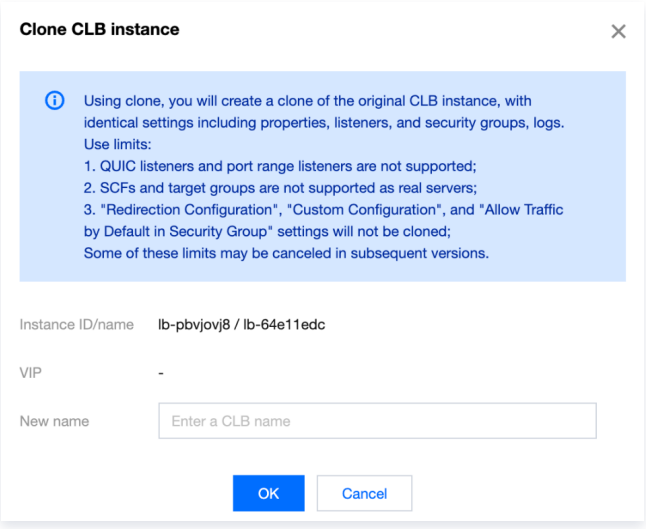
Limits

- **Instance attribute restrictions**
  - Only pay-as-you-go instances can be cloned; prepaid instances are not supported.
  - 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.
  - Cloning IPv6 and IPv6 NAT64 instances as well as the instances binding the IPv4 and IPv6 simultaneously is not supported.
  - The following settings will not be cloned and require reconfiguration: **Custom configuration**, **Redirection configurations**, and **Allow Traffic by Default** in security groups.
  - 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**
  - Cloning the instances with QUIC and port listeners is not supported.
  - Private network CLB instances with TCP_SSL listeners cannot be cloned.
  - Instances with layer-7 listeners that have no forwarding rules cannot be cloned.
  - Cloning is not supported when the number of listeners for an instance exceeds 50.
- **Backend service restrictions**

  Cloning the instances with the target group and SCF as the real server type is not supported.

Cloning Instances in Console

1. Log in to the Cloud Load Balancer console and click **Instance Management** in the left sidebar.
2. In the upper left corner of the "Instance Management" page, select the region. Locate the instance to be cloned in the instance list, and click **Actions** on the right side, then **More** > **Clone**.
3. In the **Clone Cloud Load Balance** pop-up window, enter the name for the cloned instance and click **OK**.



Cloning Instances via API

For more information, see CloneCloudLoadBalancer.

Deleting CLB Instances

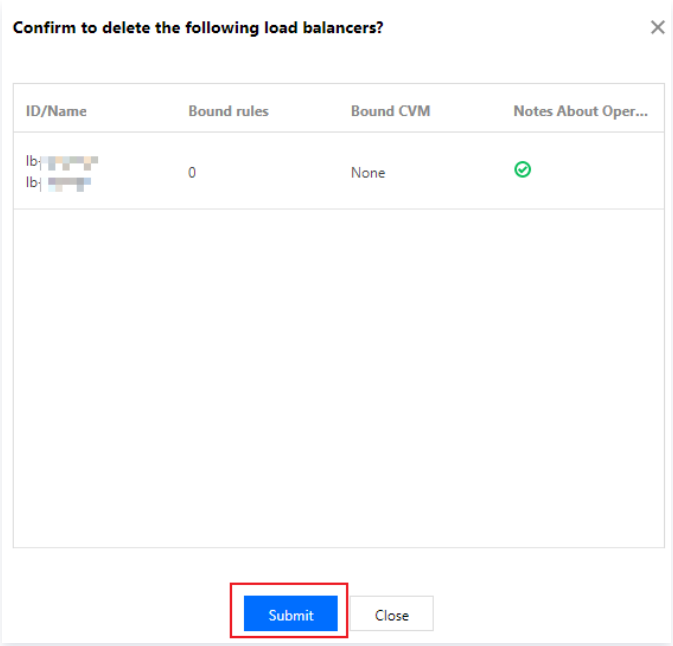
Last updated: 2023-09-04 21:38:54

> **Note:**
> The monthly-subscribed instances cannot be deleted, but you can stop renewing them upon expiration.

After confirming that the Cloud Load Balance instance has no traffic and is no longer needed, you can delete it via the Cloud Load Balance console or API.
Once deleted, the instance will be permanently terminated and cannot be recovered. We strongly recommend unbinding all backend servers and observing for a period before proceeding with the deletion.

Deleting a CLB instance via the console

1. Log in to the Cloud Load Balancer console.
2. Locate the Cloud Load Balance instance you wish to delete, and click **More** > **Delete** in the rightmost action column.
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.



Deleting a CLB instance via API

For more information, see Delete Cloud Load Balancer.

CLB listener
CLB Listener Overview

Last updated: 2023-12-14 15:08:11

After creating a Cloud Load Balancer instance, you need to configure a listener for it. The listener is responsible for monitoring requests on the Cloud Load Balancer instance and distributing traffic to backend 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 of the Cloud Load Balancer, also known as the frontend port, is used to receive requests and forward them to the backend servers.
2. Listening policies, such as load balancing policies, session persistence, and so on.
3. Health check policies.
4. Bind with backend service. Select the backend server's IP address and port. The service port, also known as the backend port, is used by the backend service to receive requests.

Supported Protocol Types

CLB listeners can monitor layer-4 and layer-7 requests on the Cloud Load Balancer instance and distribute these requests to backend servers for processing. The main difference between layer-4 and layer-7 CLB lies in whether the traffic is forwarded based on layer-4 or layer-7 protocols when load balancing user requests. For example, layer-4 CLB is used for TCP, UDP, and other layer-4 protocol requests, while layer-7 CLB is used for HTTP, HTTPS, and other layer-7 protocol 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:**
> Traditional private network Cloud Load Balancer instances do not support configuring TCP SSL listeners.

Protocol Type	Agreements	Description	Application Scenarios
Layer-4 protocol	TCP	Connection-oriented and reliable transport layer protocol: • The source and destination ends must perform a three-way handshake to establish a connection before data transfer. • Session persistence based on the client IP address (source IP address) is supported. • The client IP address can be found at the network layer. • The server can directly obtain the client IP address.	TCP is suitable for scenarios that require high reliability and data accuracy but have lower demands for transfer speed, such as file transfers, email sending and receiving, and remote logins. For more information, see Configuring a TCP Listener.
	UDP	Connection-less transport layer protocol:	UDP is suitable for scenarios that have high requirements for transfer efficiency

		• The source and destination ends do not establish a connection, nor maintain the connection status. • Each UDP connection is point-to-point. • One-to-one, one-to-many, many-to-one, and many-to-many communications are supported. • Session persistence based on the client IP address (source IP address) is supported. • The server can directly obtain the client IP address.	but relatively low requirements for accuracy, such as instant messaging and online videos. For more information, see Configuring a UDP Listener.
	TCP SSL	Secure TCP: • TCP SSL listeners support configuring certificates to block unauthorized access. • Unified certificate management is supported for CLB to implement decryption. • One-way authentication and mutual authentication are supported • The server can directly obtain the client IP address.	TCP SSL is suitable for scenarios with high security requirements under the TCP protocol and supports TCP-based custom protocols. For more information, see Configuring a TCP SSL Listener.
	QUIC	UDP-based multiplexing concurrent transport layer protocol: • QUIC implements reliable data transmission, security and HTTP2 over UDP, and is comparable to the combination of TCP, TLS, and HTTP2. • In a QIUC connection, no matter what happens to the IP address or port, the connection will not be interrupted, enabling seamless connection migration.	Suitable for scenarios such as audio/video services and game services, where smooth connection migration without interruption is required when the network changes, for example, frequent switches between mobile and Wi-Fi networks. For more information, please refer to Configuring 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.	Applications that require identifying the content of requests, such as web applications and app services. 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	For HTTP applications requiring encrypted transmission, see Configuring an HTTPS Listener.

Port Configuration

Port type	Description	Specifications and Limits
Listening Port (frontend port)	The listening port is used by a CLB instance to receive and forward requests to real servers for load balancing. You can configure CLB for the port range 1-65535, such as 21 (FTP), 25 (SMTP), 80 (HTTP), and 443 (HTTPS).	On one CLB instance: • UDP protocol ports can be the same as TCP protocol ports. For example, you can create both TCP:80 and UDP:80 listeners. • Listening ports cannot be duplicated within the same protocol category. TCP, TCP SSL, HTTP, and HTTPS all belong to the TCP category. For example,

		you cannot create both a TCP:80 listener and an HTTP:80 listener simultaneously.
Service Port (backend port)	The service port is the port through which backend servers provide services, receiving and processing traffic from the Cloud Load Balancer. In a single Cloud Load Balancer instance, the same frontend listening port can forward traffic to multiple ports on multiple backend servers.	On one CLB instance: • Service ports can be reused for different listening protocols. For example, HTTP:80 and HTTPS:443 listeners can be bound to the same port on the same backend 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.

Relevant Document

Usage restrictions

Configuring TCP Listener

Last updated: 2023-09-04 21:51:51

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 with high requirements for reliability and data accuracy, and lower demands for transmission speed, such as file transfers, email sending and receiving, and remote logins. Real servers bound to the TCP listener can directly obtain the real client IP address.

Prerequisites

You need to create a Cloud Load Balancer instance.

Directions

Step 1. Configure a listener

1. Log in to the Cloud Load Balancer console and click **Instance management** in the left sidebar.
2. In the upper left corner of the CLB instance list page, select the region, and in the operation column on the right side of the instance list, click **Configure Listener**.
3. Under the TCP/UDP/TCP SSL/QUIC listener section, click **New** and configure a TCP listener in the "Create Listener" dialog box that appears.

3.1 **Basic configuration**

Configuration Item	Description	Deleting spectators
Item	Listener name.	test-tcp-80
Listener Protocol and Ports	• Listening protocol: In this case, select `TCP`. • Listening port: The port used to receive requests and forward them to the real server. The port number ranges from 1 to 65535. • The 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). • Weighted Round Robin (WRR) Algorithm: Requests are distributed to backend servers in sequence based on their weights. This algorithm performs scheduling based on the number of new connections. Servers with higher weights have a higher probability of being polled, 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. **Note**: If WLC is selected, the listener does not support session persistence.	WRR
Two-Way RST	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. If you want to use this feature, please submit a ticket.	Selected

3.2 Health check

For more information about health check, see TCP Health Check.

3.3 Session persistence

Session Persistence Configuration	Description	Deleting spectators
Session Persistence Switch	• 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
Session persistence duration.	Session persistence duration. • Session persistence is terminated if there are no new requests in the connection within the specified duration. • Value range: 30–3600 seconds	30s

Step 2. Bind a backend server

1. On the "Listener Management" page, click the listener you just created, such as the `TCP:80` listener mentioned above, to view the bound backend services on the right side 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 Function: First, enter the "Default Port," and then select the backend servers. The port for each backend server will be set to the default port.

Step 3. Configure a security group (optional)

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 an existing listener, go to the "Listener Management" page, click on the completed listener, and click the icon to modify or the icon to delete.

Configuring a UDP Listener

Last updated: 2023-09-04 21:52:00

You can create a UDP listener to a Cloud Load Balancer (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.

Limits

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

Prerequisites

You need to create a Cloud Load Balancer instance.

Directions

Step 1. Configure a listener

1. Log in to the Cloud Load Balancer console and click **Instance management** in the left sidebar.
2. In the upper left corner of the CLB Instances list page, select the region, and in the Actions column on the right side of the instance list, click **Configure Listener**.
3. Under TCP/UDP/TCP SSL/QUIC listeners, click **New**, and configure the UDP listener in the **Create Listener** dialog box that appears.

3.1 Basic configuration

Configuration Item	Description	Deleting spectators
Item	Listener name.	test-udp-8000
Listener Protocol and Ports	• Listening protocol: In this case, select `UDP`. • 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. • Within the same Cloud Load Balancer instance, listening ports must be unique.	UDP:8000
Balancing Method	CLB supports two scheduling algorithms for UDP listeners: weighted round robin (WRR) and weighted least connections (WLC). • Weighted Round Robin (WRR) Algorithm: Requests are distributed to backend servers in sequence based on their weights. This algorithm performs scheduling based on the number of new connections. Servers with higher weights have a higher probability of being polled, and servers with the same weight process the same number of connections. • Weighted Least Connections (WLC): Server loads are estimated based on the number of active connections. WLC scheduling is performed considering both server loads and weights. When weights are equal, backend servers with fewer connections have a higher probability of being polled. **Note**: If WLC is selected, the listener does not support session persistence.	WRR
Scheduling by QUIC ID	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).	Enabled

3.2 Health check

For more information about health check, see UDP Health Check.

3.3 Session persistence

Session Persistence Configuration	Description	Deleting spectators
Session Persistence Switch	• 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
Session persistence duration.	Session persistence duration. • Session persistence is terminated if there are no new requests in the connection within the specified duration. • Value range: 30–3600 seconds	30s

Step 2. Bind a backend 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:**
> Default port functionality: First, enter the "default port" and then select the backend servers. The port for each backend server will be set to the default port.

Step 3. Configure a security group (optional)

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 created listener, go to the "Listener Management" page, click on the completed listener, and click the icon to modify or the icon to delete.

Configuring TCP SSL Listener

Last updated: 2023-09-04 21:52:09

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 listener is currently supported only for CLB but not classic CLB.

Scenarios

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.

Preparations

You need to create a Cloud Load Balancer instance.

Instructions

Step 1. Configure a listener

1. Log in to the Cloud Load Balancer console and click **Instance management** in the left sidebar.
2. Select the region from the top left corner of the CLB instance list page, and click **Configure Listener** in the operation column on the right side of the instance list.
3. Under TCP/UDP/TCP SSL/QUIC listeners, click **Create** and configure the TCP SSL listener in the **Create Listener** dialog box that appears.

   3.1 Basic Configuration

Configuration Item	Note	Sample
Name	Listener name.	test-tcpssl-9000
Listener Protocol and Ports	• Listening protocol: In this case, select `TCP SSL`. • Listening port: The port used to receive requests and forward them to the real server. The port number ranges from 1 to 65535. • The listening port must be unique in the same CLB instance.	TCP SSL:9000
SSL parsing method	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 new certificate.	Existing certificate
Balancing Method	CLB supports two scheduling algorithms for TCP SSL listeners: weighted round robin (WRR) and weighted least connections (WLC). • Weighted Round Robin (WRR) Algorithm: Requests are distributed to backend servers in sequence based on their weights. This algorithm performs scheduling based on the number of new connections. Servers with higher weights have a higher probability of being polled, 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	WRR

	server loads and weights. For servers with the same weight, those have less loads are more likely to be scheduled.	

3.2 Health check

For more information on health checks, see TCP SSL Health Check.

3.3 Session persistence (not supported currently)

TCP SSL listeners don't support session persistence currently.

Step 2. Bind a backend 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:**
> 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 (optional)

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 an existing listener, navigate to the "Listener Management" page, select the completed listener, and click on the icon to modify or the icon to delete.

Configuring a QUIC Listener

Last updated: 2023-09-04 21:54:05

You can create a QUIC listener to a Cloud Load Balancer (CLB) instance to forward encrypted QUIC requests from the client. For QUIC listeners, the real server can directly obtain the real client IP address.
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 multiplexing 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.

Description

- QUIC listeners are only supported by Cloud Load Balancer (CLB) instances and not available for traditional CLB instances.
- QUIC listeners are only supported for Cloud Load Balancer instances deployed on VPC networks, and not available for instances on the classic network.
- Only IPv4 and IPv6 NAT64 Cloud Load Balancer (CLB) instances support the QUIC listener; the IPv6 version is not supported.

Preparations

You need to create a Cloud Load Balancer instance.

Instructions

Step 1. Configure a listener

1. Log in to the Cloud Load Balancer console and click **Instance management** in the left sidebar.
2. In the upper left corner of the CLB instance list page, select the region. In the operation column on the right side of the instance list, click on 'Configure Listener'.
3. Under TCP/UDP/TCP SSL/QUIC listeners, click on **Create** to configure a QUIC listener in the **Create Listener** dialog box that appears.

   3.1 Basic configuration

Configuration Item	Note	Sample
Name	Listener name.	test-quic-443
Listener Protocol and Ports	• Listener protocol: Select QUIC. CLB can receive QUIC requests made by clients, but TCP is still used between CLB and real server. • Listening port: The port used to receive requests and forward them to the real server. The port number ranges from 1 to 65535. • The listening port must be unique in the same CLB instance.	QUIC:443
SSL parsing method	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 new certificate.	Existing certificate

Balancing Method	CLB supports two scheduling algorithms for QUIC listeners: weighted round robin (WRR) and weighted least connections (WLC). • Weighted Round Robin (WRR) Algorithm: Requests are distributed to backend servers in sequence based on their weights. This algorithm performs scheduling based on the number of new connections. Servers with higher weights have a higher probability of being polled, 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 Health check

For more information on health checks, see TCP SSL 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 choose 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)

Should you need to modify or delete an existing listener, navigate to the "Listener Management" page, select the desired listener, and click on the icon to modify or the icon to delete.

Documentation

CLB Supports QUIC Protocol

Configuring an HTTP Listener

Last updated: 2023-09-04 21:58:06

You can create an HTTP listener on a Cloud Load Balancer (CLB) instance to forward HTTP protocol requests from the client. HTTP is suitable for applications that require content recognition of requests, such as web applications and app services.

Preparations

You need to create a Cloud Load Balancer instance.

Instructions

Step 1. Configure a listener

1. Log in to the Cloud Load Balancer console and click **Instance management** in the left sidebar.
2. In the upper left corner of the CLB instance list page, select the region. In the operation column on the right side of the instance list, click **Configure Listener**.
3. Under HTTP/HTTPS Listener, click **New** and configure the HTTP listener in the pop-up "Create Listener" dialog box.

3.1 Create a listener

Configuration Item	Note	Sample
Name	Listener name.	test-http-80
Listener Protocol and Ports	• Listening protocol: In this case, select `HTTP`. • Listening port: The port used to receive requests and forward them to the real server. The port number ranges from 1 to 65535. • The listening port must be unique in the same CLB instance.	HTTP:80
Enable Persistent Connection	Once enabled, the CLB instance and backend services will use persistent connections, and the CLB instance will no longer pass through the source IP address, which can be obtained from XFF. To ensure normal forwarding, 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 enabled, the number of connections between the CLB and backend services will fluctuate within the range of [QPS, QPS*60], depending on the connection reuse rate. If there is a limit on the maximum number of connections for backend services, enable this feature with caution. This feature is currently in beta testing. If you wish to use it, please submit a beta test application. • The health check source IP range 100.64.0.0/10 has been allowed by default, and IPs within this range do not need to be allowed again.	Disabled

3.2 Forwarding rule creation

Forwarding Rule Configuration	Note	Sample
Domain name	Forwarding domain name: • Length: 1 to 80 characters. • It cannot begin with underscores (_). • Exact and wildcard domain names are supported. • Regular expressions are supported.	www.example.com

	• For detailed configuration rules, see Layer-7 Domain Name Forwarding and URL Rules.	
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 detailed configuration rules, see URL Path Forwarding Rules.	/index
Balancing Method	For HTTP listeners, CLB supports three scheduling algorithms: weighted round robin (WRR), weighted least connections (WLC), and IP Hash. • Weighted Round Robin (WRR) Algorithm: Requests are distributed to backend servers in sequence based on their weights. This algorithm performs scheduling based on the number of new connections. Servers with higher weights have a higher probability of being polled, 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
Getting Client IP	Enabled by default.	Enabled
Gzip Compression	Enabled by default.	Enabled

3.3 Health check

For more information, see HTTP Health Check.

3.4 Session persistence

Session Persistence Configuration	Note	Sample
Session Persistence Switch	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
Session persistence duration.	Session persistence is terminated if there are no new requests in the connection within the specified duration. Value range: 30-3600 seconds	30s

Step 2. Bind a backend server

1. On the "Listener Management" page, click the listener you just created, such as the `HTTP:80` listener. Click the + icon on the left to expand the domain names and URL paths. Select the specific URL path to view the backend services bound to this path on the right side 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 (optional)

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 an existing listener, navigate to the "Listener Management" page, click on the completed listener, and then click on the icon to modify or the icon to delete.

Configuring HTTPS Listener

Last updated: 2023-09-04 21:58:16

You can create an HTTPS listener on a Cloud Load Balancer (CLB) instance to forward HTTPS protocol requests from the client. HTTPS is suitable for HTTP applications that require encrypted transmission.

Preparations

You need to create a Cloud Load Balancer instance.

Instructions

Step 1. Configure a listener

1. Log in to the Cloud Load Balancer console and click **Instance management** in the left sidebar.
2. In the upper left corner of the CLB instance list page, select the region. In the operation column on the right side of the instance list, click **Configure Listener**.
3. Under HTTP/HTTPS Listener, click **New** and configure the HTTPS listener in the pop-up "Create Listener" dialog box.

3.1 **Create a listener**

Configuration Item	Note	Sample
Name	Listener name.	test-https-443
Listener Protocol and Ports	• Listening protocol: HTTPS is used in this example. • Listening port: The port used to receive requests and forward them to the real server. The port number ranges from 1 to 65535. • The listening port must be unique in the same CLB instance.	HTTPS:443
Enable Persistent Connection	Once activated, a persistent connection is established between the CLB and the backend service, and the CLB no longer forwards the source IP. Please retrieve the source IP from the XFF. To ensure normal forwarding, please either enable the default security group on the CLB or allow 100.127.0.0/16 on the CVM's security group. **Note:** • Once enabled, the number of connections between the CLB and backend services will fluctuate within the range of [QPS, QPS*60], depending on the connection reuse rate. If there is a limit on the maximum number of connections for backend services, enable this feature with caution. This feature is currently in beta testing. To participate, please submit a beta test application. • The health check source IP range 100.64.0.0/10 has been allowed by default, and IPs within this range do not need to be allowed 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 method	One-way authentication and mutual authentication are supported. Cloud Load Balancer handles the SSL encryption and decryption overhead, ensuring secure access.	One-Way authentication
Server Certificate	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. **Note:** Configuring dual certificates is only supported for Cloud Load Balancer, not for traditional Cloud Load Balancer. After configuring dual certificates, QUIC functionality	Select an existing certificate.

	cannot be enabled.	
CA certificate	You can select an existing certificate in the SSL Certificate Service console or upload a certificate.	Select an existing certificate.

3.2 Forwarding rule creation

Forwarding Rule Configuration	Note	Sample
Domain name	Forwarding domain name: • Length: 1 to 80 characters. • It cannot begin 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 only one default domain name.	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 Path	Forwarding URL: • Length: 1-200 characters. • Regular expressions are supported. • For detailed configuration rules, see URL Path Forwarding Rules.	/index
Balancing Method	For HTTPS listeners, CLB supports three scheduling algorithms: weighted round robin (WRR), weighted least connections (WLC), and IP Hash. • Weighted Round Robin Algorithm: Requests are distributed to backend servers in sequence based on their weights. The weighted round robin algorithm schedules based on the **number of new connections**. Servers with higher weights have a higher probability of being polled, 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. • When gRPC is selected as the backend protocol, the gRPC service must be deployed on the real server. The backend forwarding protocol supports gRPC only when HTTP2.0 is enabled and QUIC is disabled.	HTTP
Getting Client IP	Enabled by default.	Enabled

Gzip Compression	Enabled by default.	Enabled

3.3 Health check

For more information on health checks, see HTTPS Health Check.

3.4 Session persistence

Session Persistence Configuration	Note	Sample
Session Persistence Switch	• 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
Session persistence duration.	• Session persistence is terminated if there are no new requests in the connection within the specified duration. • Value range: 30–3600 seconds	30s

Step 2. Bind a backend server

1. On the "Listener Management" page, click on the listener you just created, such as the aforementioned `HTTPS:443` listener. Click on the + on the left to expand the domain name and URL path. Select the specific URL path to view the backend services bound to this path on the right side 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 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 an existing listener, navigate to the "Listener Management" page, click on the completed listener, and then click on the icon to modify or the icon to delete.

Load Balancing Methods

Last updated: 2023-09-05 10:02:36

The load balancing method refers to the algorithm used by Cloud Load Balancer to distribute traffic to real servers, achieving various balancing effects based on the chosen method.

Weighted round-robin algorithm

The Weighted Round-Robin Scheduling algorithm distributes requests to different servers in a sequential manner. This scheduling method addresses performance disparities among servers by assigning appropriate weights to represent their processing capabilities. Requests are allocated to servers based on their weights and round-robin order. The algorithm schedules based on the number of new connections, with higher-weighted servers receiving connections first. Servers with higher weights have a greater probability of being polled, and those with equal weights process the same number of connections.

- **Advantages**: Simple and practical, no need to track the status of all current connections, making it a stateless scheduling method.
- **Disadvantages**: Relatively simple, and in cases where the request service time varies greatly or each request consumes different amounts of time, it can easily lead to imbalanced server workloads.
- **Applicable scenarios**: Performs best when the backend time consumed by each request is roughly equal. Commonly used for short-lived connections, such as HTTP.
- **User Recommendation**: Weighted round-robin scheduling is recommended when it is known that each request occupies a similar amount of backend time, and the backend servers process the same or similar types of requests. It is also recommended when the difference in request times is small, as this implementation consumes less resources, does not require traversal, and offers higher efficiency.

Load Balancing Methods

In real-world scenarios, the time a client's request spends on a server can vary significantly. As the working time extends, using simple round-robin or random balancing algorithms may result in a significant difference in the number of connection processes on each server, leading to an inability to achieve true Cloud Load Balancing.

Least-connection scheduling is a dynamic scheduling algorithm that estimates server workloads based on the number of active connections to the servers. The scheduler needs to track the number of established connections for each server. When a request is scheduled to a server, its connection count increases by one; when the connection terminates or times out, the count decreases by one.

The Weighted Least-Connection Scheduling algorithm is an improvement on the least-connection scheduling algorithm. It assigns different weights to servers based on their processing capabilities, allowing them to accept a corresponding number of service requests, thus enhancing the basic least-connection scheduling algorithm.

> **Note:**
> Suppose the weights of each real server are denoted as wi, and their current number of connections as ci. Calculate ci/wi for each server in sequence, and the server instance with the smallest value will be the next one to receive a new request. If there are multiple server instances with the same ci/wi value, use the weighted round-robin method for scheduling.

- **Advantages**: This algorithm is suitable for long-duration request services, such as FTP and other applications.
- **Disadvantages**: Due to interface limitations, the least connections and session persistence features cannot be enabled simultaneously.
- **Applicable Scenarios**: Situations where the backend time occupied by each request varies significantly. Commonly used for long-lived connection services.
- **User Recommendation**: If users need to handle various requests with significantly different backend processing times, such as a difference of 3ms and 3s, it is recommended to use the Weighted Least-Connection Scheduling algorithm to implement Cloud Load Balancer.

Source hashing scheduling algorithm

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.

- **Advantages**: Allows requests from a specific client to be consistently mapped to the same real server through a hash table. In scenarios where session persistence is not supported, ip_hash can be used to implement simple session persistence.
- **User Recommendation**: Perform hash calculations on the source address of the request, and combine it with the backend server weights you set to distribute requests to a matching server, ensuring that requests from the same client IP are always sent to a specific server. This method is suitable for protocols without Cookie functionality.

Load Balancing Algorithm Selection and Weight Configuration

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 there are 3 backend servers with identical configurations (CPU/memory), and due to their consistent performance, the weights for all backend servers can be set to 10.
  1.2 Currently, each real server has established 100 TCP connections with the client, and an additional real server has been 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 Assuming you are new to cloud services and have a short website setup time with low site load, it is recommended to purchase real servers with the same configuration. In this case, all real servers serve as indistinguishable access layer servers.
  1.2 In this scenario, you can set the weights of all backend servers to the default value of 10 and use the weighted round-robin load balancing method for traffic distribution.
- **Scenario 3:**
  1.1 Suppose you have 5 servers for hosting simple static website access, and the ratio of their compute capabilities is 9:3:3:3:1 (calculated based on CPU and memory).
  1.2 In this scenario, the backend server weights can be set sequentially to 90, 30, 30, 30, and 10. As most static website visits involve short connection requests, the weighted round-robin load balancing method can be employed, allowing the Cloud Load Balancer instance to distribute requests according to the performance ratios of the backend servers.
- **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, it is recommended to set appropriate weights based on the performance of the backend servers, assigning smaller weights to those with higher workloads. Additionally, adopting the least-connection Cloud Load Balancer method can help distribute requests to backend servers with fewer active connections, thus addressing the issue of excessive load on a particular server.
- **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 In this situation, the highest-performing server handles a larger number of requests, and you do not want to overload it. Instead, you aim to allocate new requests to idle servers.
  1.3 In this scenario, the least-connection scheduling method can be adopted, along with appropriately reducing the weight of busy servers. This facilitates Cloud Load Balancer to allocate requests to real servers with fewer active connections, achieving effective load balancing.
- **Scenario 6:**
  1.1 Suppose you want subsequent client requests to be allocated to the same server. In this case, using the weighted round-robin or weighted least connections method cannot guarantee that requests from the same client will be assigned to a fixed server.
  1.2 To accommodate the needs of specific application servers and ensure that client sessions have "stickiness" or "persistence," the ip_hash load balancing method can be employed for traffic distribution. This approach guarantees that

requests from the same client are always directed to the same real server (except when the number of servers changes or the server becomes unavailable).

Session persistence

Last updated: 2023-09-05 14:24:33

Session persistence enables requests from the same IP to be forwarded to a single backend server. By default, Cloud Load Balance routes each request to different backend server instances for load distribution. However, you can utilize session persistence to direct requests from specific users to the same backend server instance, allowing applications that require session maintenance (such as shopping carts) to function 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

The current HTTP connection timeout (keepalive_timeout) is set to 75 seconds by default. If adjustment is required, please enable custom configuration. If the time threshold is exceeded and there is no data transmission in the session, the connection will be terminated.

The timeout for the current TCP connection is not adjustable at the moment and is set to 900 seconds by default. If the time threshold is exceeded and there is no data transmission in the session, the connection will be terminated.

Configuring Session Persistence

1. Log in to the Cloud Load Balance console and click the ID of the Cloud Load Balance instance to be configured with session persistence to enter its details page.
2. Select the **Listener management** tab.
3. Click **Modify** next to the Cloud Load Balance 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 persistent connection, see Configuring an HTTP Listener and Configuring an HTTPS Listener.

Scenario 1: HTTP Layer-7 business

Assuming the client-side access is via HTTP/1.1 protocol, with the header information set to Connection:keep-alive. If the access to the backend server is through CLB without session persistence enabled, will the subsequent access be directed to the same server?
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: Layer-4 TCP 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.
Firstly, according to the Layer-4 implementation mechanism, when TCP enables persistent connections, if the connection remains unbroken, the same server can be accessed in both the initial and subsequent requests. However, if the first connection is released due to other reasons (network restart, connection timeout) during the second request, it may be scheduled to another backend server. The default global timeout for persistent connections is 900 seconds, meaning that if there are no new requests, the connection will be released.

Layer-7 Redirection Configuration

Last updated: 2023-09-05 14:24:42

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

> **Note:**
> - Session Persistence: If a client accesses `example.com/bbs/test/123.html` and the backend CVM has session persistence enabled, the original session persistence mechanism will be invalidated when traffic is redirected to `example.com/bbs/test/456.html` after enabling redirection.
> - TCP/UDP redirection: redirection at IP + port level is not supported currently but will be available in subsequent versions.

Redirection Overview

- Automatic Redirection
  - Introduction
    The system automatically creates an HTTP listener for the existing `HTTPS:443` listener, using port 80 by default. After successful creation, the `HTTP:80` address can automatically redirect to the `HTTPS:443` address for access.
  - Use Case:
    Forced HTTPS redirection, i.e., HTTP to HTTPS conversion. When a PC or mobile browser accesses a web service with an HTTP request, CLB redirects all `HTTP:80` requests to `HTTPS:443` for forwarding.
  - Solution 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
  - Introduction
    You can configure one-to-one redirection, such as redirecting `Listener1 / Domain1 / URL1` to `Listener2 / Domain2 / URL2` within a specific CLB instance.

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

  - Use Case:
    Single-path redirection, such as when a web service needs to be temporarily taken offline (e.g., due to product sellout, page maintenance, or updates and upgrades). In this case, the original page should be redirected to a new page. Without redirection, the old address in users' favorites and search engine databases would only return a `404/503` error message page, resulting in a degraded user experience and wasted traffic.

Automatic Redirection

Tencent Cloud CLB supports one-click HTTP to HTTPS conversion. Suppose you need to configure the website `https://www.example.com`. You want users to access the site securely via HTTPS protocol, regardless of whether they enter an HTTP request (`http://www.example.com`) or an HTTPS request (`https://www.example.com`) in the browser.

Usage Limits

Redirection configuration includes protocol/port, domain, and path settings. To avoid loops, please be aware of the following restrictions:

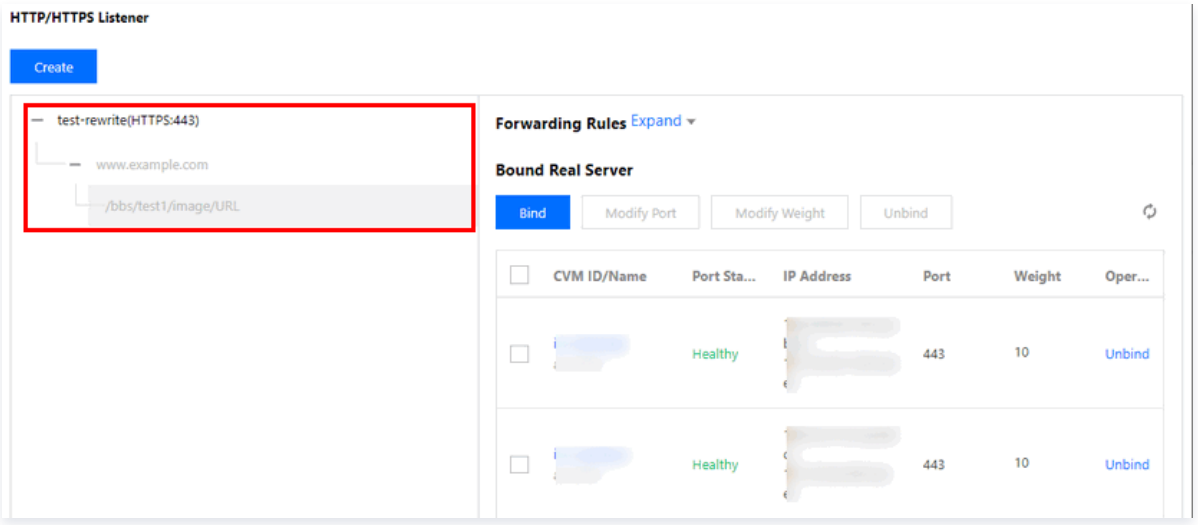
- Redirection configuration is not allowed if the original access path and the redirected access path are identical.
- If the original access path has already been configured with a redirection policy (including the original access path and the redirected access path), further configuration is not allowed.
- Redirection configuration is not allowed if the configured redirection access path is the original access path of another redirection policy.

Preparations

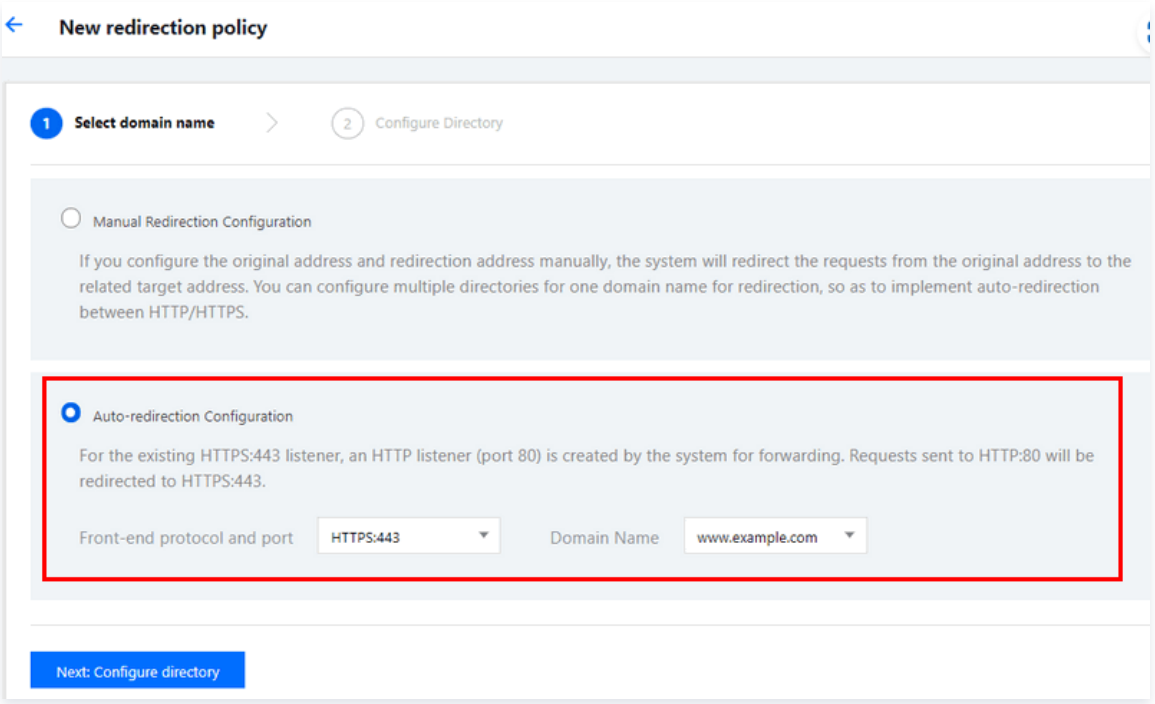
An `HTTPS:443` listener has been configured.

Instructions

1. Please complete the configuration of the CLB HTTPS listener in the Tencent Cloud Load Balance Console and set up the web environment for `https://example.com`. For more information, refer to Configuring HTTPS Listener.
2. The result of the HTTPS listener configuration is as shown below:



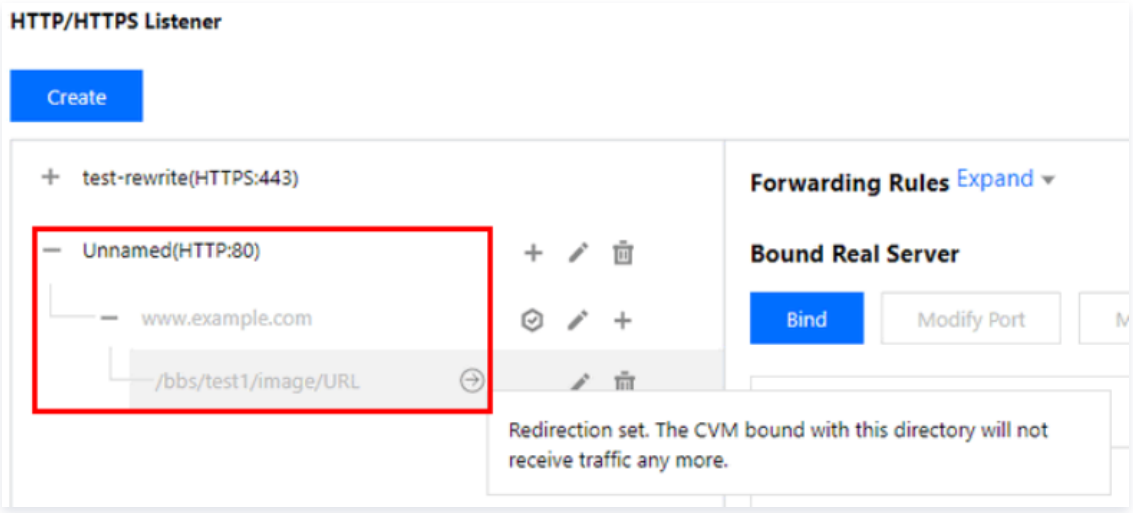
3. In the "Redirection Configuration" tab of the CLB instance details, click **Create Redirection Configuration**.
4. Select **Automatic Redirection Configuration**, choose the configured HTTPS listener and domain name, then select the redirection status code in "Domain Configuration" and click **Submit** to complete the configuration.



Note:

- The "Domain Configuration" feature in redirection is currently in beta testing. To use it, please contact Online Support.
- Status codes 301 (Moved Permanently), 302 (Move Temporarily), and 307 (Temporary Redirect) are detailed in the HTTP/1.1 standard (RFC 7231).

5. Upon completing the redirection configuration, as shown in the image below, an `HTTP:80` listener is automatically configured for the `HTTPS:443` listener, and all HTTP traffic will be automatically redirected to HTTPS.



Manual Redirection

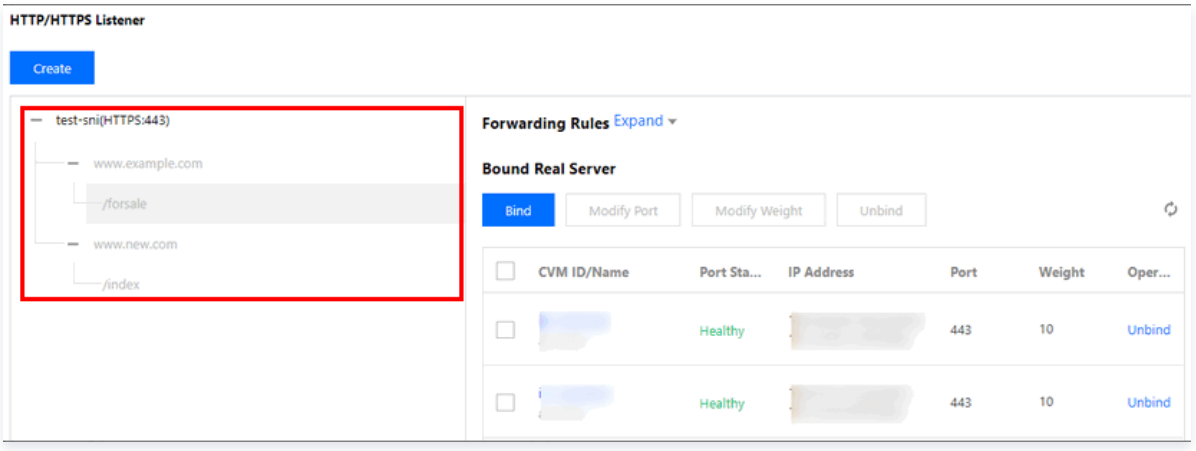
Tencent Cloud CLB supports configuring one-to-one redirection. For example, if a business uses the "forsale" page for promotional activities and the event has ended, the event page `https://www.example.com/forsale` needs to be redirected to the new homepage `https://www.new.com/index`.

Preparations

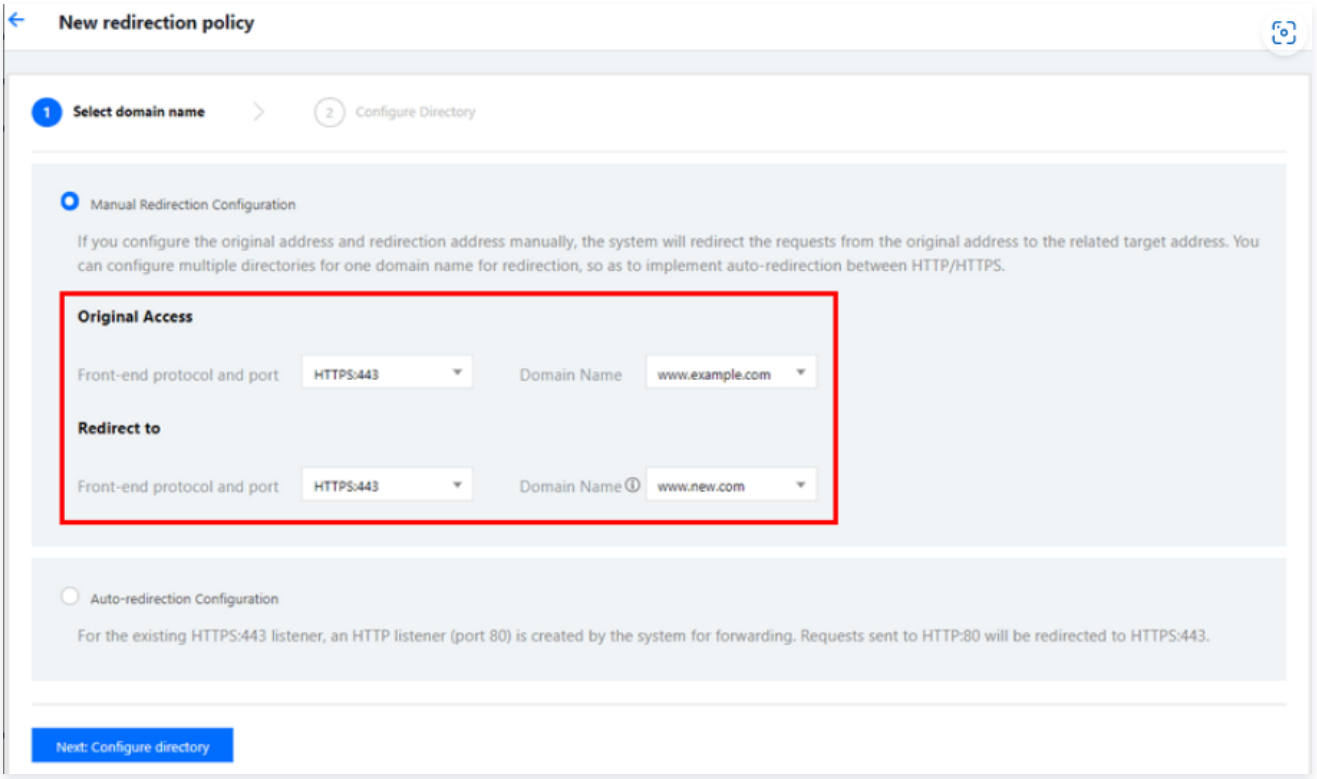
- An HTTPS listener has been configured.
- Configured forwarding domain: `https://www.example.com/forsale`.
- Configured forwarding domain and path: `https://www.new.com/index`.

Instructions

1. Please complete the configuration of the CLB HTTPS listener in the Tencent Cloud Load Balance Console and set up the web environment for `https://example.com`. For more information, refer to Configuring HTTPS Listener.
2. The result of the HTTPS configuration is as shown below:



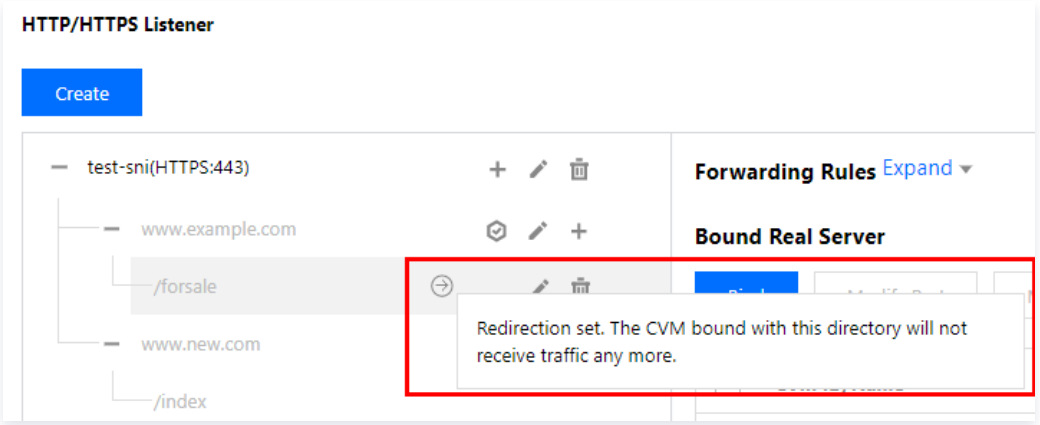
3. In the "Redirection Configuration" tab of the CLB instance details, click **Create Redirection Configuration**.
4. Select **Manual Redirection Configuration**, choose the original frontend protocol port, domain, and path, as well as the redirected frontend protocol port, domain, and path. In "Domain Configuration", select the redirection status code, choose to retain or not retain the URL, and click **Submit** to complete the configuration.



Note:

- The "Domain Configuration" feature in redirection is currently in beta testing. To use it, please contact Online Support.
- Status codes 301 (Moved Permanently), 302 (Move Temporarily), and 307 (Temporary Redirect) are detailed in the HTTP/1.1 standard (RFC 7231).

5. Upon completing the redirection configuration, the result is as shown below. You can see that in the `HTTP:443` listener, `https://www.example.com/forsale` has been redirected to `https://www.new.com/index`.



Layer-7 Custom Configuration

Last updated: 2025-07-30 10:10:40

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.
- Each instance can be bound to only one custom configuration, while a single custom configuration can be associated with multiple instances.
- 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 configuraitons:

Configuration Field	Default Value/Recommended Value	Valid Values	Note
ssl_protocols	• 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.
ssl_ciphers	ssl_ciphers default value	ssl_ciphers value range	Cipher suite.
client_header_timeout	60s	[30-120]s	Timeout period of obtaining client request headers. Status code 408 is returned in case of timeout.
client_header_buffer_size	4k	[1-256]k	Size of the default buffer where client request headers are stored.
client_body_timeout	60s	[30-120]s	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	60M	[1-10240]M	• If you set this field to a value in the range of 1-256 MB, there are no other requirements. • The maximum supported size is 10240M, or 10G. When the configuration range of `client_max_body_size` exceeds 256M, the value of proxy_request_buffering must be set to 'off'.
keepalive_timeout	75s	[0-900]s	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 `add_header xxx yyy`.

			For example, you can set it to `add_header Access-Control-Allow-Methods 'POST, OPTIONS'; add_header Access-Control-Allow-Origin *;` for cross-region scenarios.
more_set_headers	Custom	-	Headers returned to the client. Set this field in the format of `more_set_headers "A:B"`.
proxy_connect_timeout	4s	[4-120]s	Timeout period of connecting to a real server.
proxy_read_timeout	60s	[30-3600]s	Timeout period of reading a real server response.
proxy_send_timeout	60s	[30-3600]s	Timeout period of sending a request to a real server.
server_tokens	on	on，off	• `on`: displays version information. • `off`: hides version information.
keepalive_requests	100	[1-10000]	Maximum number of requests that can be sent over the client-server persistent connection.
proxy_buffer_size	16k	[1-32]k	Size of server response headers, which is the size of a single buffer set in `proxy_buffer` by default. To use `proxy_buffer_size`, `proxy_buffers` must be set at the same time.
proxy_buffers	4 16k	[3-8] [4-16]k	Buffer quantity and size.
proxy_request_buffering	off	on，off	• `on`: 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. • `off`: 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.
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	• `X-Real-Port $remote_port`: client port. • `X-clb-lbid $lbid`: CLB LBID, which is the identifier of a CLB instance. • `Stgw-request-id $stgw_request_id`: request ID (used in CLB only). • `X-Forwarded-Port`: CLB listener port. • `X-Method`: client request method. • `X-Uri`: client request URI.
send_timeout	60s	[1-3600]s	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), `proxy_redirect` will reset `http`

			to `https` in the HTTP header `Location` or `Refresh` for safe redirection.
ssl_early_data	off	on，off	Enable or disable TLS 1.3 0-RTT. The `ssl_early_data` will only take effect when the `ssl_protocols` field includes TLSv1.3. **Enabling `ssl_early_data` carries the risk of replay attacks, so proceed with caution.**
http2_max_field_size	4k	[1-256]k	Maximum size of request headers after HPACK compression.
error_page	-	error_page code [= [response]] uri	Upon encountering a specific error code, a predefined URI is displayed, with the default response code set to 302. The URI must begin with a `/` path.
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	The TOA feature toggle enables the TOA functionality by default, adding the client source IP and client source port from TOA to `$remote_addr` and `$remote_port`, respectively. This means that the IP information from TOA is already passed through in the `X-Forwarded-For` and `X-Real-IP` headers. Note: This parameter is only applicable to IPv4 CLB instance configurations.
l7_toa_proxy_transparent	off	on，off	• When this configuration is disabled, the CLB instance, upon establishing a new connection with the real server, will by default encapsulate the source IP address of the received four-tuple as the client source IP and forward it to the backend. • When this configuration is enabled, it means that the client source IP packet in TOA will be sent to the backend real server. If persistent connections are enabled, IPs within the `100.127.0.0/16` network segment will be used. Note: This parameter is only applicable to IPv4 CLB instance configurations.

Note:
The values of `proxy_buffer_size` and `proxy_buffers` must satisfy the constraint: 2 *max(`proxy_buffer_size`, `proxy_buffers.size`)* ⩽ *(`proxy_buffers.num` − 1)* `proxy_buffers.size`. For example, if `proxy_buffer_size` is set to 24 KB and `proxy_buffers` is set to 8 8 KB, then 2 *24 KB = 48 KB, and (8 − 1)* 8 KB = 56 KB. In this case, 48 KB ⩽ 56 KB, so the configuration will not result in an error; otherwise, an error will occur.

ssl_ciphers Configuration Instructions

When configuring the `ssl_ciphers` encryption suite, the format must be consistent with the one used by OpenSSL. The algorithm list consists of one or more `<cipher strings>`, with multiple algorithms separated by colons. "ALL" represents all algorithms, "!" disables the specified algorithm, and "+" moves the algorithm to the last position.
The default forcibly disabled encryption algorithms are: `!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!DHE`.
Default Value:

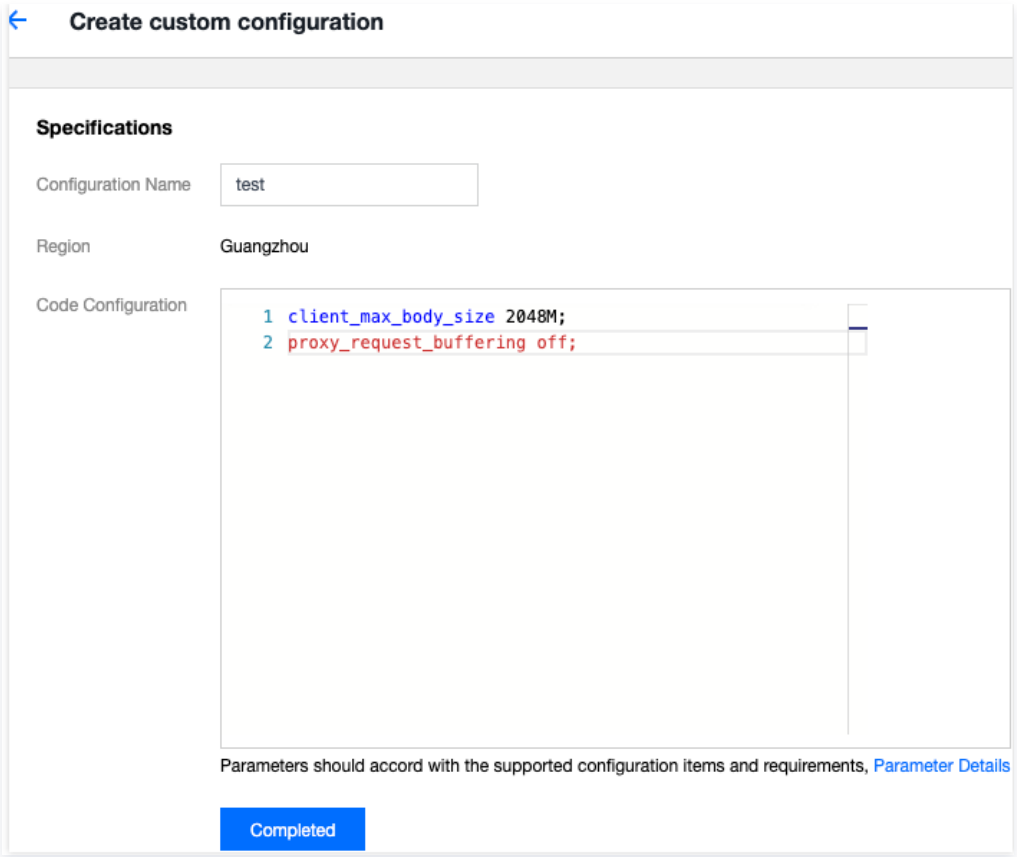
```
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-CHACHA20-POLY1305:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128:AES256:AES:HIGH:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!DHE:3DES;
```

Value Range:

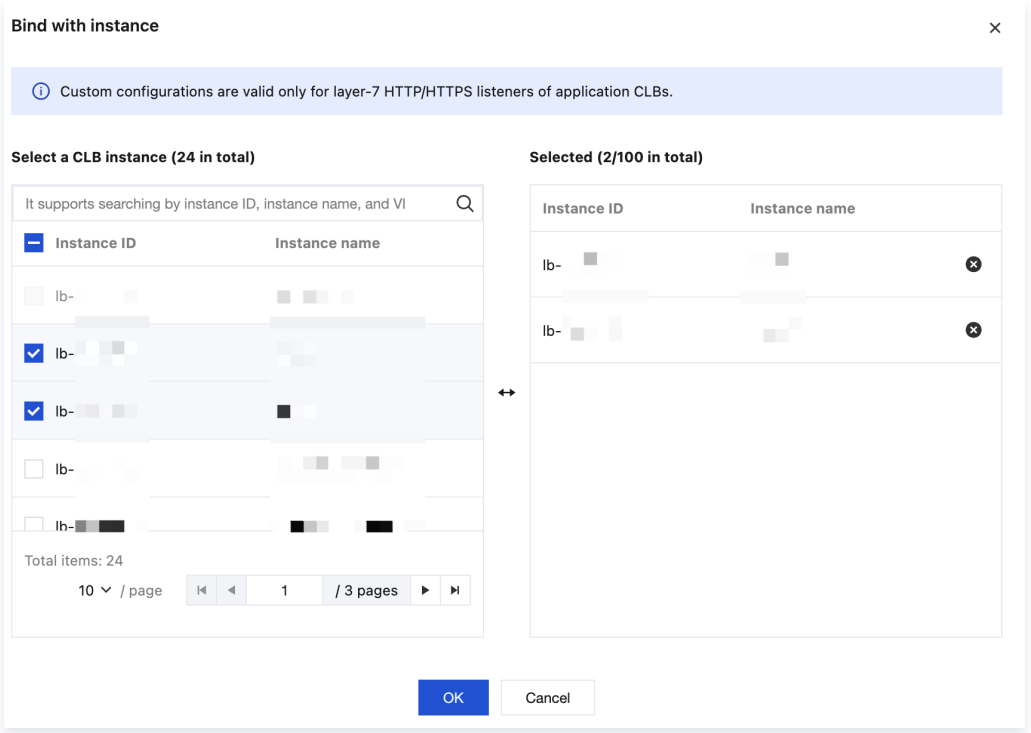
```
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-
GCM-SHA384:ECDHE-RSA-CHACHA20-POLY1305:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-
SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-
SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:ECDH-ECDSA-AES128-SHA256:ECDH-RSA-AES256-SHA:ECDH-
ECDSA-AES256-SHA:SRP-DSS-AES-256-CBC-SHA:SRP-AES-128-CBC-SHA:ECDH-RSA-AES128-SHA256:DH-RSA-AES128-
SHA256:DH-RSA-CAMELLIA128-SHA:DH-DSS-AES256-GCM-SHA384:DH-RSA-AES256-SHA256:AES256-SHA256:SEED-
SHA:CAMELLIA256-SHA:ECDH-RSA-AES256-SHA384:ECDH-ECDSA-AES128-GCM-SHA256:DH-RSA-AES128-SHA:DH-RSA-AES128-
GCM-SHA256:DH-DSS-AES128-SHA:ECDH-RSA-AES128-SHA:DH-DSS-CAMELLIA256-SHA:SRP-AES-256-CBC-SHA:DH-DSS-AES128-
SHA256:SRP-RSA-AES-256-CBC-SHA:ECDH-ECDSA-AES256-GCM-SHA384:ECDH-RSA-AES256-GCM-SHA384:DH-DSS-AES256-
SHA256:ECDH-ECDSA-AES256-SHA384:AES128-SHA:DH-DSS-AES128-GCM-SHA256:AES128-SHA256:DH-RSA-SEED-SHA:ECDH-
ECDSA-AES128-SHA:IDEA-CBC-SHA:AES128-GCM-SHA256:DH-RSA-CAMELLIA256-SHA:CAMELLIA128-SHA:DH-RSA-AES256-GCM-
SHA384:SRP-RSA-AES-128-CBC-SHA:SRP-DSS-AES-128-CBC-SHA:ECDH-RSA-AES128-GCM-SHA256:DH-DSS-CAMELLIA128-
SHA:DH-DSS-SEED-SHA:AES256-SHA:DH-RSA-AES256-SHA:kEDH+AESGCM:AES256-GCM-SHA384:DH-DSS-AES256-
SHA:HIGH:AES128:AES256:AES:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!DHE
```

CLB Custom Configuration Examples

1. Log in to the Cloud Load Balance console and click **Custom Configuration** in the left sidebar.
2. At the top of the "Custom Configuration" page, select the region and click on **Create**.
3. On the "Create Custom Configuration" page, fill in the configuration name and code configuration items, with each code configuration item ending with a `;` . Once the configuration is complete, click **Finish**.



4. Return to the "Custom Configuration" page and click on **Bind to Instance** under the operations column on the right.
5. In the pop-up "Bind to Instance" dialog, select the Cloud Load Balance instance to bind, and click **Submit**.



6. After binding an instance, click on the custom configuration ID you just set on the "Custom Configuration" page to access the details page. Click on the **Bind Instance** tab to view the load balancing instance you just bound.
7. (Optional) You can now view the corresponding custom configuration information on the instance list page.

> **Note:**
> If the "Bind Custom Configurations" column is not displayed on the list page, click the ⚙ icon in the top-right corner. In the pop-up "Customize List Field" dialog box, select "Bind Custom Configurations" and click **OK**. The "Bind Custom Configurations" column will then be displayed on the list page.

The default configuration sample code is as follows. When copying the code, please ensure there are no blank lines at the end to guarantee successful configuration:

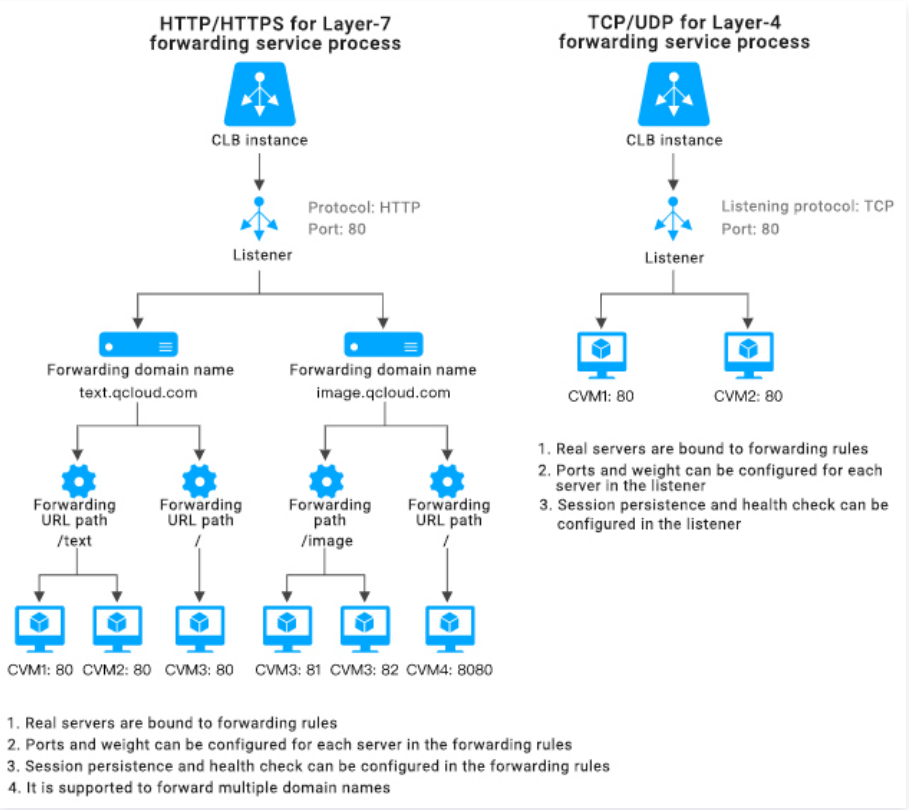
```
ssl_protocols   TLSv1 TLSv1.1 TLSv1.2;
client_header_timeout   60s;
client_header_buffer_size   4k;
client_body_timeout    60s;
client_max_body_size   60M;
keepalive_timeout    75s;
add_header     xxx yyy;
more_set_headers      "A:B";
proxy_connect_timeout    4s;
proxy_read_timeout    60s;
proxy_send_timeout    60s;
```

Layer-7 Domain Name Forwarding and URL Rules

Last updated: 2023-09-05 14:40:48

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.

- When only one forwarding rule is established, accessing VIP + URL corresponds to the respective forwarding rule, allowing normal access to the service.
- When multiple forwarding rules are established, accessing VIP + URL does not guarantee access to a specific domain name + URL. Users need to directly access the domain name + URL to ensure the specific forwarding rule takes effect. That is, when configuring multiple forwarding rules, the same VIP corresponds to multiple domain names. In this case, it is not recommended to access the service via VIP + URL; instead, access the service through the specific domain name + URL.

Layer-7 Forwarding Configurations

Domain forwarding configurations

Layer-7 CLB can forward requests from different domain names and URLs to different servers for processing. A layer-7 listener can be configured with multiple domain names, and each domain name can have multiple forwarding paths. For configuring forwarding domain names, please refer to Configuring CLB Forwarding Domain Names.

- Domain name length limit: 1 to 80 characters.
- It cannot begin with `_`.
- Supports precise domain names, such as `www.example.com`.
- Wildcard domain names are supported, currently only in the form of `*.example.com` or `www.example.*`, with `*` appearing at the beginning or the end, and only one `*` occurrence in a single domain name.
- For non-regular expression forwarding domain names, the supported character set includes: `a-z` `0-9` `.` `-` `_`.
- Forwarding domain names support regular expressions. The domain names for regular expressions are:
  - Supported character set includes: `a-z` `0-9` `.` `-` `?` `=` `~` `_` `-` `+` `\` `^` `*` `!` `$` `&` `|` `(` `)` `[` `]`.

- Must start with `~` and `~` can only appear once.
- Examples of regular domain names supported by Cloud Load Balance: `~^www\d+\.example\.com$`.

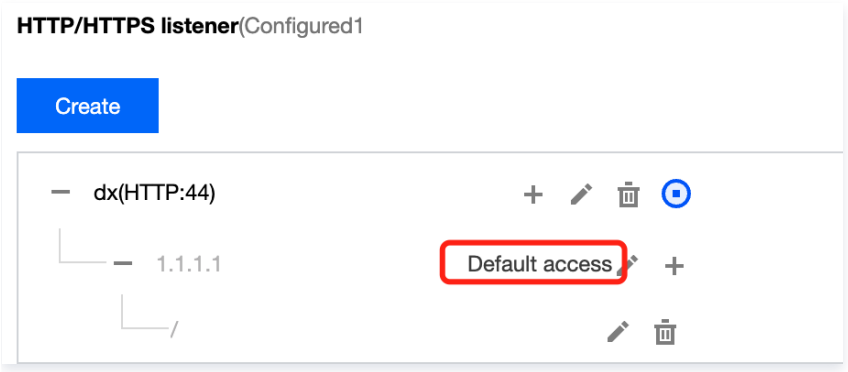
Forwarded domain name matching

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 for access.
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 for access.
3. Configure a wildcard domain name in the forwarding rule and multiple URLs in the forwarding group, accessing through matching the request domain name + URL. When users want different domain names to point to the same URL address, this configuration method can be followed. Taking `example.qcould.com` as an example, the format is as follows:
   - Exact domain name `example.qcloud.com` precisely matches the `example.qcloud.com` domain.
   - The prefix wildcard domain `*.qcloud.com` matches all domain names ending with `qcloud.com`.
   - Suffix wildcard domain `example.qcloud.*` matches all domain names starting with `example.qcloud`.
   - Domain name with regex matching: `~^www\d+\.example\.com$` matches based on the regular expression.
   - Matching priority: Exact domain name > Prefix wildcard domain name > Suffix wildcard domain name > Regular expression domain name. If multiple domain names at the same level are matched simultaneously, the matching order cannot be guaranteed. It is recommended to use more precise domain names to avoid multiple rules being matched at the same time.
4. Configure a domain name in the forwarding rule and a fuzzy-matching URL in the forwarding group. Use prefix matching and add a wildcard $ at the end for a complete match.
   For example, by configuring forwarding group `URL ~*.(gif|jpg|bmp)$`, the user aims to match any file ending with `gif`, `jpg`, or `bmp`.

Layer-7 Domain Name Forwarding and URL Rules

When a client request does not match any domain name of the listener, CLB forwards the request to the default domain name (Default Server), making the default rule controllable. Only one default domain name can be configured per listener.
For example, two domain names are configured under the `HTTP:80` listener of CLB1: `www.test1.com` and `www.test2.com`, with `www.test1.com` being the default domain name. When a user accesses `www.example.com`, since no domain name is matched, CLB forwards 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 the default domain name.
  - If your Layer-7 listener does not have a default domain name configured, unmatched client requests will be forwarded to the first domain name loaded by the CLB. Since the loading order may not be consistent with the console configuration order, it may not necessarily be the first one configured in the console.
- Starting from May 18, 2020:
  - All newly created layer-7 listeners must have a default domain name configured: The first rule of a layer-7 listener will always enable the default domain name. When calling the API to create a layer-7 rule, CLB will automatically set

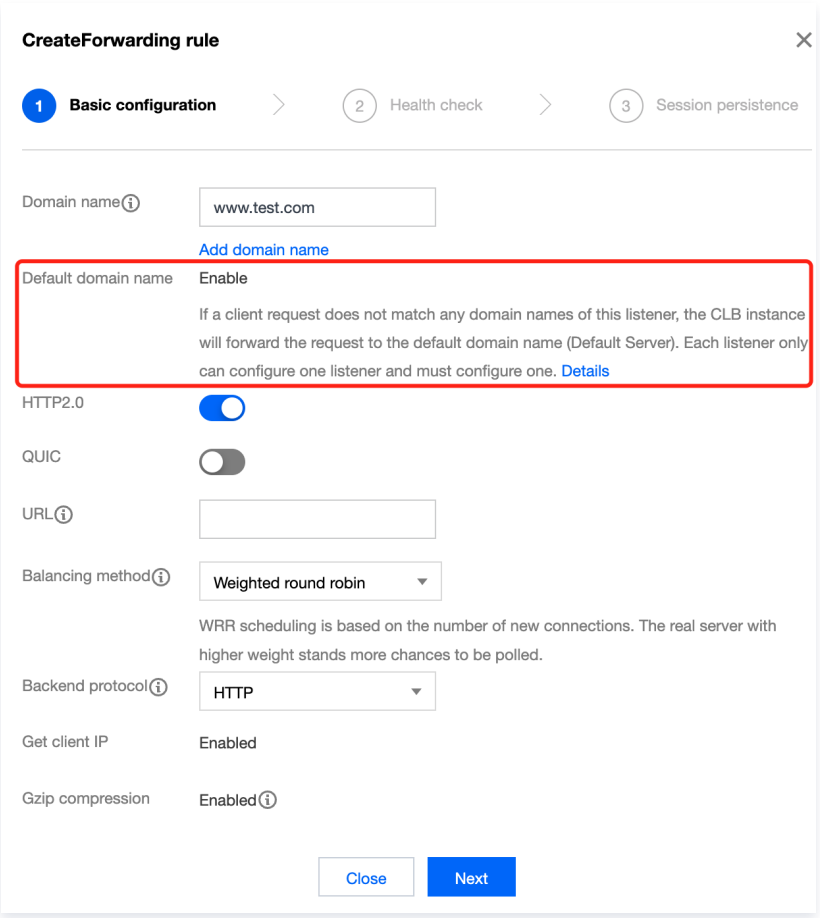
the DefaultServer field to true.

- For all listeners with configured default domain names, when modifying or deleting the default domain name, a new default domain name must be specified: When using the console, you need to specify a new default domain name; when calling the API, if a new default domain name is not set, CLB will automatically set the earliest created domain name among the remaining domain names as the new default domain name.
- For existing rules without a default domain name configured: You can directly configure the default domain name according to your business requirements, as described in "Step 4". If you do not configure it, Tencent Cloud will set the first domain name loaded by the CLB as the default domain name, and the existing listeners will be processed by June 19, 2020.

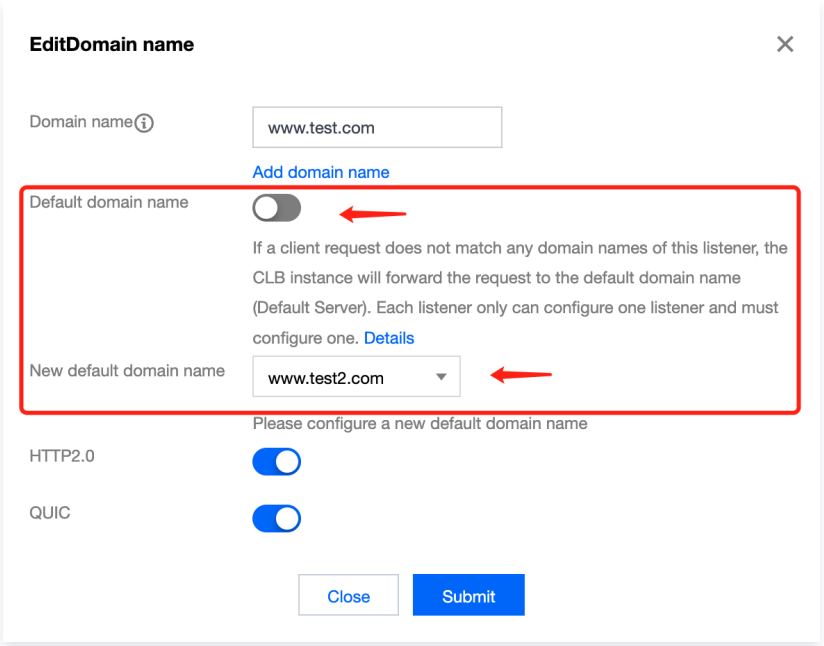
The aforementioned policy has been gradually implemented since May 18, 2020, with slight variations in the effective dates for different instances. Starting from June 20, 2020, all layer-7 listeners with non-empty forwarding domain names will have a default domain name.

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

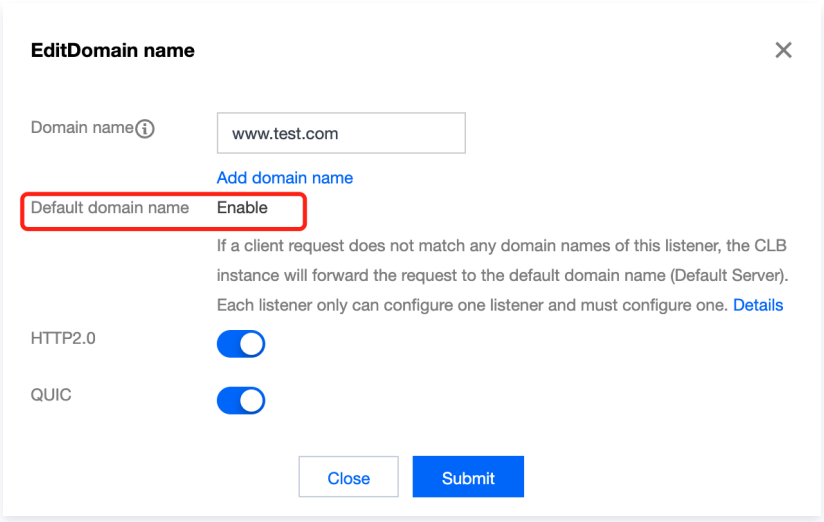
- **Operation 1**: When configuring the first forwarding rule for a layer-7 listener, the default domain name must be enabled.



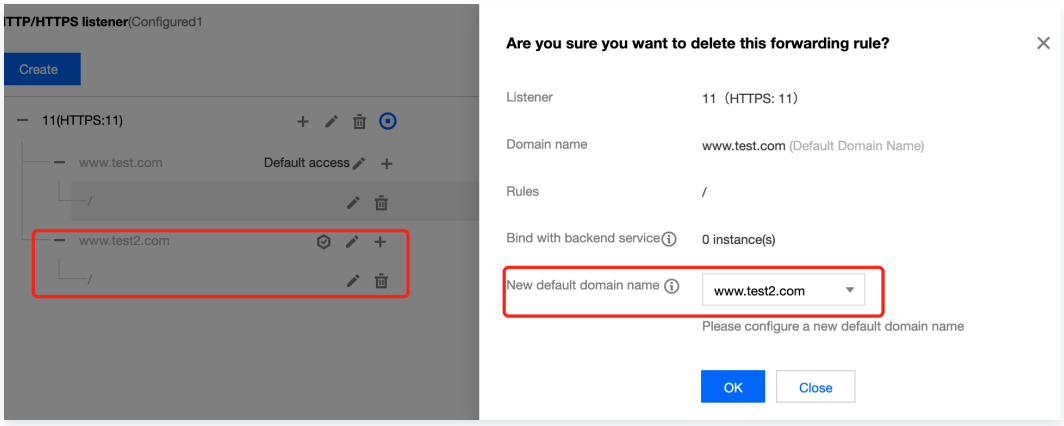
- **Step 2**: Disable the current default domain name.
  - When multiple domain names exist under a listener and the current default domain name is disabled, a new default domain name must be specified.



- When a listener has only one domain name, and that domain name is the default domain name, disabling the default domain name is not allowed.

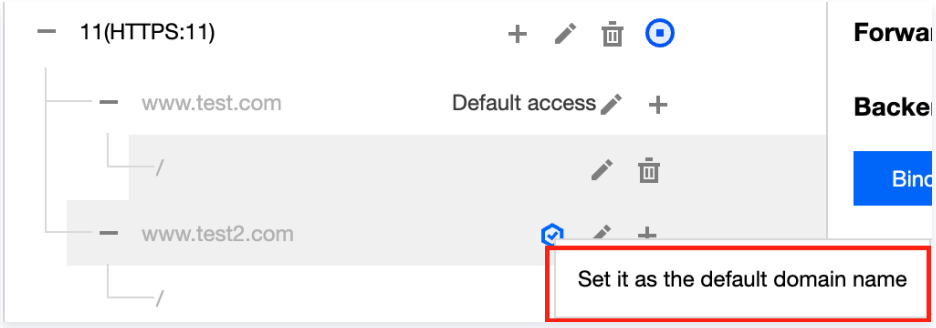


- **Step 3**: Delete the default domain name.
  - To delete rules under the default domain name for a listener with multiple domain names:
    - If this rule is not the last one for the default domain, it can be deleted directly.
    - If this rule is the last one for the default domain name, a new default domain name must be set.



- When a listener has only one domain name, all rules can be deleted directly without the need to set a new default domain name.

- **Step 4:** Modify the default domain name. You can quickly modify the default domain name in the listener list.

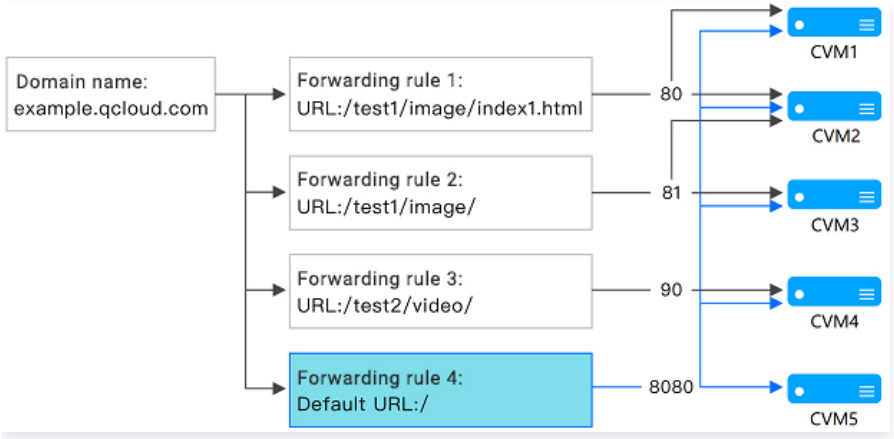


Forwarded URL path configuration rules

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.

- Forwarding URL length: 1 to 200 characters.
- Non-regular expression forwarding URLs must start with `/`, are case-sensitive, and support the following character sets: `a-z` `A-Z` `0-9` `.` `-` `_` `/` `=` `?` `:`.
- Forwarding URL supports regular expressions:
  - The URL of the regular expression must start with `~`, and `~` can only appear once.
  - The supported character set for regular expression URLs includes: `a-z` `A-Z` `0-9` `.` `-` `_` `/` `=` `?` `~` `^` `*` `$` `:` `(` `)` `[` `]` `+` `|`.
  - An example of a regular expression URL is as follows: `~* .png$`.
- URL forwarding matching rules are as follows:
  - `=` Indicates an exact match.
  - `^~` at the beginning indicates that the URL starts with a regular string, not a regex match.
  - `~` at the beginning indicates case-sensitive regular expression matching.
  - `~*` at the beginning indicates case-insensitive regular expression matching.
  - `/` Universal match, if no other matches are found, any request will be matched.

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 `example.qloud.com/test1/image/index1.html` precisely matches the URL rule set in forwarding rule 1, so the request will be forwarded to the backend Cloud Virtual Machines associated with forwarding rule 1, which are the 80 ports of CVM1 and CVM2 in the diagram.

1.2 `example.qloud.com/test1/image/hello.html` has no exact match, and by the longest prefix, it will match forwarding rule 2. Therefore, the request will be forwarded to the backend Cloud Virtual Machines associated with forwarding rule 2, which are ports 81 of CVM2 and CVM3 in the diagram.

1.3 `example.qloud.com/test2/video/mp4/` has no exact match, and by the longest prefix, it will match forwarding rule 3. Therefore, the request will be forwarded to the backend Cloud Virtual Machine associated with forwarding rule 3, which is port 90 of CVM4 in the diagram.

1.4 `example.qloud.com/test3/hello/index.html` has no exact match and will be matched to the root directory Default URL: `example.qloud.com/` based on the longest prefix. At this point, Nginx forwards the request to the backend application server, such as FastCGI (php) or Tomcat (jsp), with Nginx acting as a reverse proxy server.

1.5 `example.qloud.com/test2/` has no exact match, and will be matched to the root directory Default URL: `example.qloud.com/` based on the longest prefix.

2. If the service cannot run properly with the URL rules set by the user, the request will not be redirected to another page after a successful match.
   For example, if the client request for example.qloud.com/test1/image/index1.html matches forwarding rule 1, but the backend server of forwarding rule 1 is experiencing an issue and returns a 404 error page, the user's access to the page will display a 404 error and will not be redirected to another page.
3. It is recommended that users set a Default URL, pointing it to a stable service page (such as a static page or homepage) and binding it to all backend Cloud Virtual Machines. In this case, if none of the rules match successfully, the system will direct the request to the page where the Default URL is located, otherwise, a 404 error may occur.
4. If default URL is not configured and none of the forwarding rules match, a 404 error will be returned when you access the service.
5. Explanation of the trailing slash in Layer-7 URL paths: When the user sets a URL ending with `/`, but the client does not include `/` when accessing, the request will be redirected to the rule ending with `/` (301 redirect).
   For example, under the `HTTP:80` listener, the configured domain name is `www.test.com`.
   5.1 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 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.

- Health check domain name length limit: 1 - 80 characters.
- The default health check domain is the forwarding domain.
- Health check domain names do not support regular expressions. When 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.
- The supported character set for health check domain names includes: `a-z` `0-9` `.` `-` `_`, for example, `www.example.qcould.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.

- Health check path length limit: 1 to 200 characters.
- The default health check path is `/` and must start with `/`.
- Health check paths do not support regular expressions. It is recommended to specify a fixed URL path (static page) for health checks.
- The supported character set for health check paths includes: `a-z` `A-Z` `0-9` `.` `-` `_` `/` `=` `?` `:`, for example, `/index`.

Using QUIC Protocol on CLB

Last updated: 2023-09-05 14:42:53

QUIC protocol can significantly enhance your App's access speed, enabling multiplexing without reconnection in scenarios such as weak networks and frequent Wi-Fi to 4G switches. This document will guide you on how to configure the QUIC protocol in the Cloud Load Balance console.

QUIC Overview

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 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. However, the CLB instance and the real server still use the HTTP1.x protocol.

Usage Limits

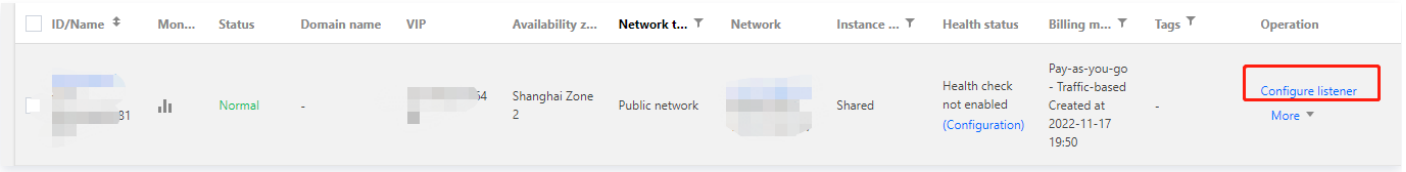
- Only CLB instances, excluding classic CLB instances, support the QUIC protocol.
- Only layer-7 HTTPS listeners support the QUIC protocol.
- Currently, CLB supports the following QUIC versions: Q050, Q046, Q043, h3-29, and h3-27.

Instructions

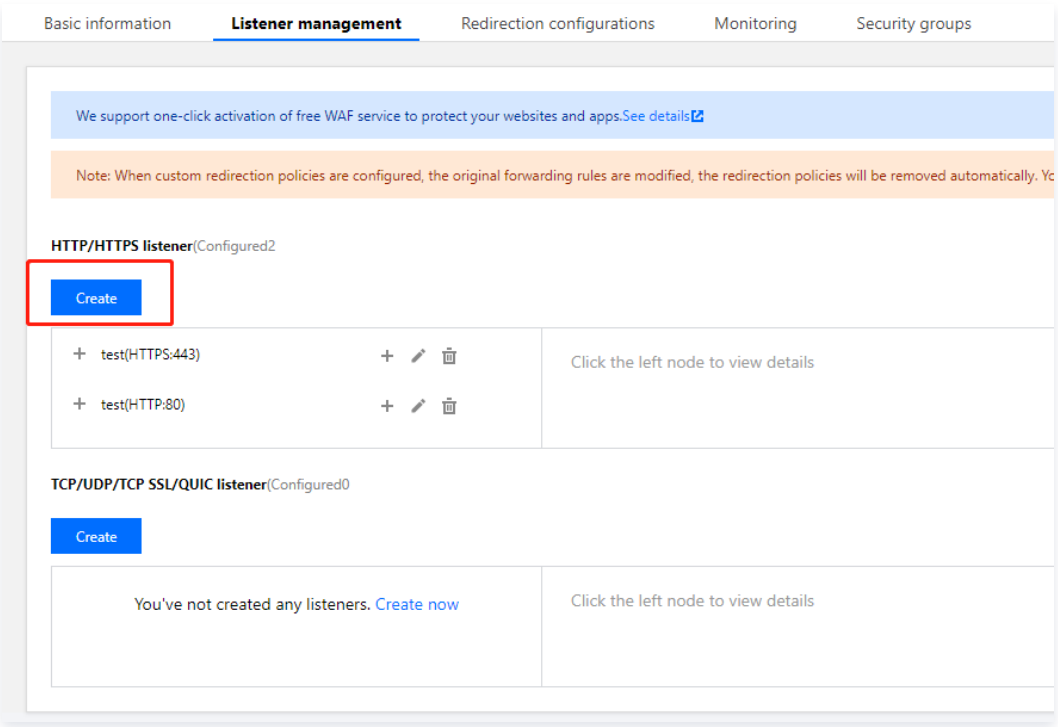
1. Create a Cloud Load Balance instance based on your requirements. For more information, see Creating Cloud Load Balance Instances.

   > **Note:**
   > When creating a CLB instance, select "Beijing", "Shanghai" or "Mumbai" for **Region**, and "Public Network" for **Network Type**.

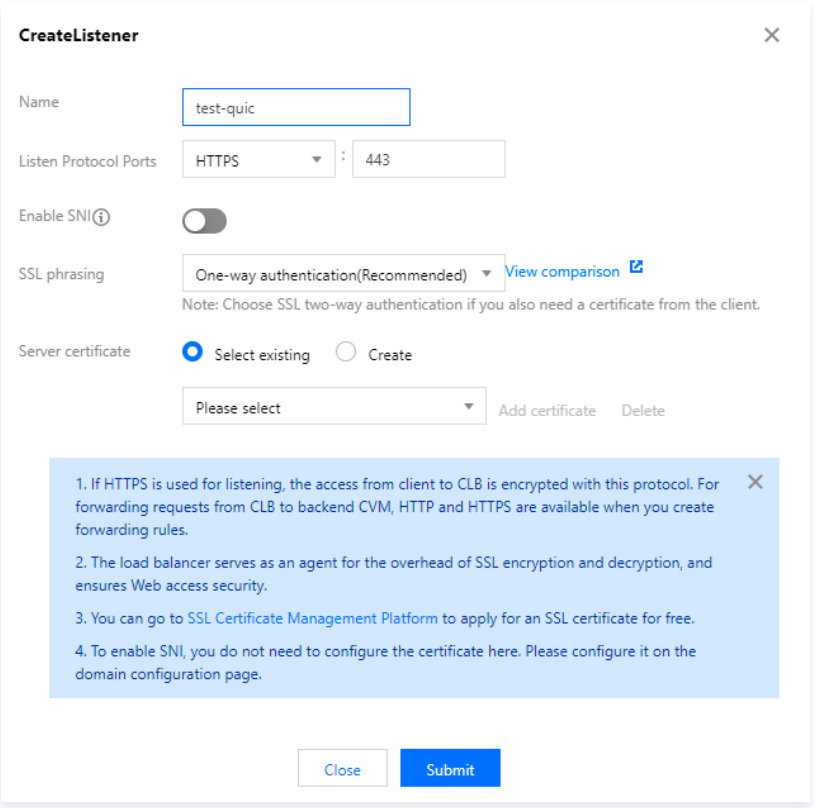
2. Log in to the Cloud Load Balance console and click **Instance management** in the left sidebar.
3. On the "Instance Management" page, click the **Cloud Load Balance** tab.
4. In the "Cloud Load Balance" tab, locate the public network Cloud Load Balance instance created in Beijing, Shanghai, or Mumbai region, and click **Configure listener** in the operation column on the right.



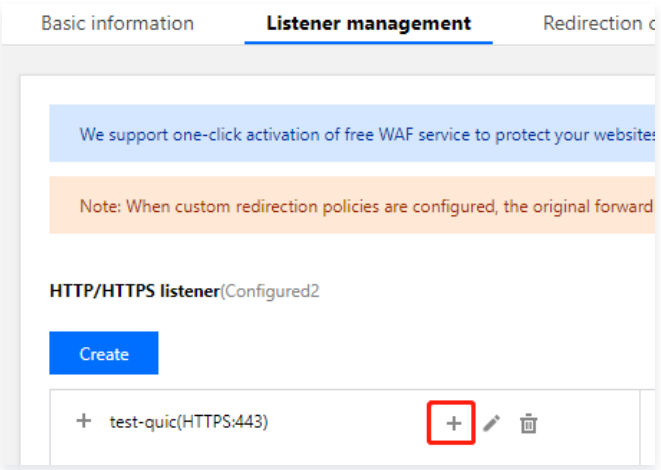
5. Under "HTTP/HTTPS Listener" on the "Listener Management" page, click **New**.



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

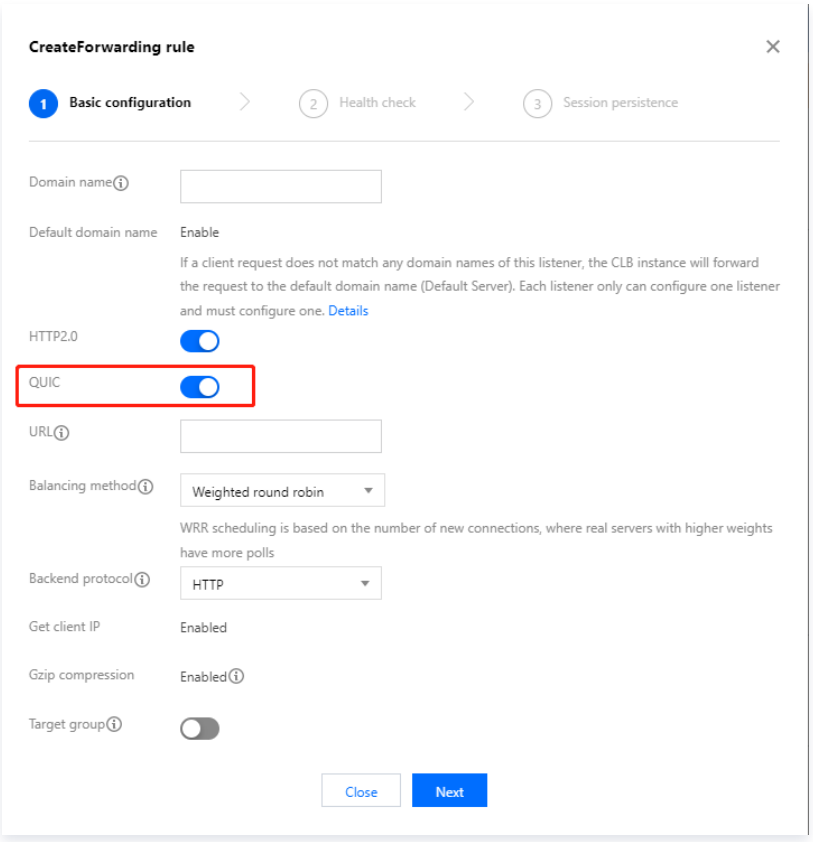


7. On the **Listener Management** tab, click the + symbol for the new listener.



8. On the "Create Forwarding Rule" page, enable the QUIC protocol, create a Layer 7 rule, fill in the 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.
> - QUIC uses the UDP protocol, occupying the UDP ports of the CLB. When the HTTPS listener enables the QUIC protocol, it automatically occupies the corresponding UDP and TCP ports. For example, when the HTTPS:443 listener enables the QUIC protocol, the rule occupies both TCP:443 and UDP:443 ports, preventing you from creating additional TCP:443 and UDP:443 listeners.



See Also

After completing the basic configuration, you can proceed with the health check and session persistence settings.

SNI Support for Binding Multiple Certificates to a CLB Instance

Last updated: 2023-09-05 14:48:55

Server Name Indication (SNI) is a technique used to enhance SSL/TLS communication between servers and clients, primarily addressing the issue of a single server being limited to using one certificate. If a server supports SNI, it can bind multiple certificates. When a client uses SNI, it must specify the domain name it wants to connect to before establishing an SSL/TLS connection with the server. The server will then return an appropriate certificate based on the specified domain name.

Use Cases

Tencent Cloud CLB's Layer 7 HTTPS listener supports SNI, allowing for the binding of multiple certificates, with different domain names in the listener rules using different certificates. For example, within the same CLB's `HTTPS:443` listener, `*.test.com` uses Certificate 1, forwarding requests from this domain to a group of servers; `*.example.com` uses Certificate 2, forwarding requests from this domain to another group of servers.

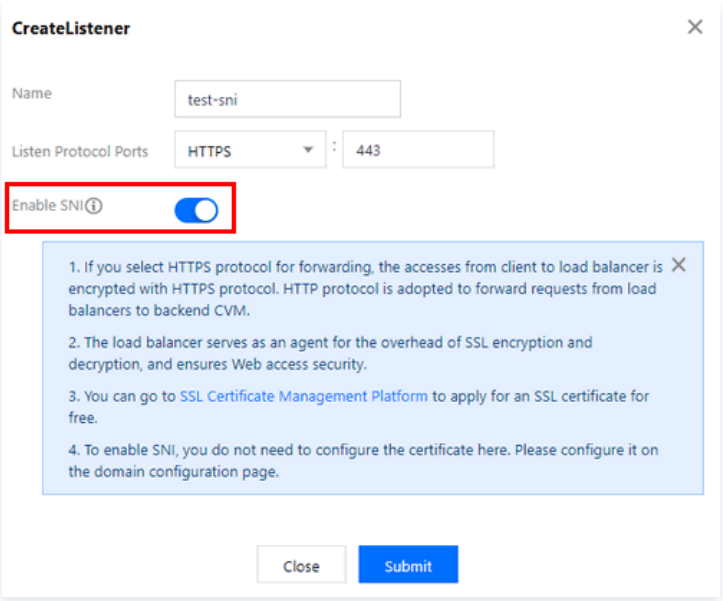
Preparations

You have purchased a Cloud Load Balance instance.

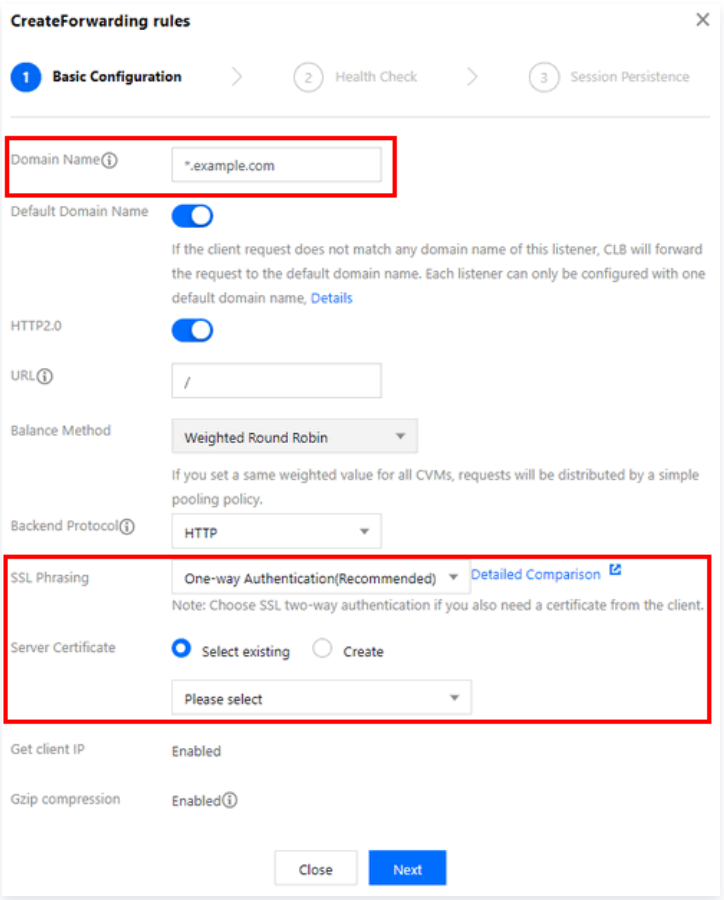
> **Note:**
> Classic Cloud Load Balance does not support forwarding based on domain name and URL; therefore, it does not support SNI.

Instructions

1. Log in to the Cloud Load Balancer console.
2. Refer to Configuring a Listener for the steps to configure a listener, and enable SNI when configuring an HTTPS listener.



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.



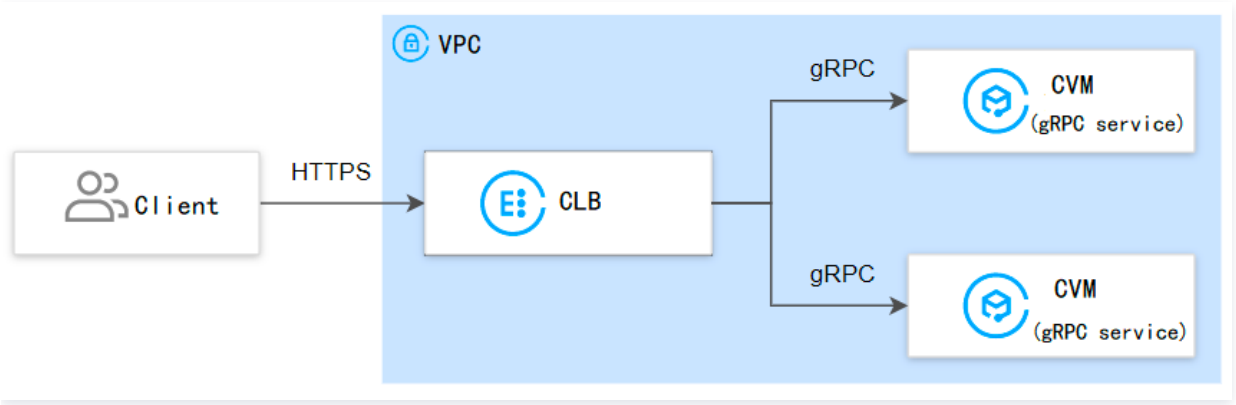
Configuring gRPC Support for Layer-7 Protocols

Last updated: 2023-09-05 14:51:58

gRPC is a high-performance open-source software framework released by Google, based on the HTTP 2.0 transport layer protocol, offering support for multiple programming languages and methods for configuring and managing network devices. This guide demonstrates how to configure a gRPC health check for an HTTPS listener, forwarding client gRPC requests through a CLB instance to backend services using the gRPC protocol.

Use Cases

When clients access backend services with the gRPC protocol via HTTPS requests, you can implement this by using an HTTPS listener on a CLB instance that supports the gRPC protocol.



Preparations

- You have created a VPC. For more information, see Creating a Virtual Private Cloud.
- You have created a CVM instance 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 Cloud Load Balance Instances.

Usage 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 supported only by VPC but not by classic networks.
- Real servers do not support SCF. (Support for the gRPC protocol within the SCF target is required.)

Instructions

Step 1. Configure a listener

1. Log in to the Cloud Load Balancer console and click **Instance management** in 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.
3. Under **HTTP/HTTPS listener**, click **Create** and configure the HTTPS listener in the pop-up window.

   3.1 Create a listener

Configuration Item	Note	Sample
Name	Listener name.	test-https-443
Listener Protocol and Ports	• Listening protocol: HTTPS is used in this example. • Listening port: The port used to receive requests and forward them to the real server. The port number ranges from 1 to 65535.	HTTPS:443

	• The listening port must be unique in the same CLB instance.	
Enable Persistent Connection	Once enabled, a persistent connection is established between the CLB and the backend service, with the CLB no longer transmitting the source IP. Please retrieve the source IP from XFF. To ensure normal forwarding, please either open the default security group on the CLB or allow 100.127.0.0/16 on the CVM's security group. **Note:** • Once enabled, the number of connections between the CLB and backend services will fluctuate within the range of [QPS, QPS*60], depending on the connection reuse rate. If there is a limit on the maximum number of connections for backend services, enable this feature with caution. This feature is currently in beta testing. If you wish to use it, please submit a beta test application. • The health check source IP range 100.64.0.0/10 has been allowed by default, and IPs within this range do not need to be allowed 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 method	One-way authentication and mutual authentication are supported. Cloud Load Balancer handles the SSL encryption and decryption overhead, ensuring secure access.	One-Way authentication
Server Certificate	You can select an existing certificate in the SSL Certificate Service console or create a new certificate.	Select an existing certificate.

3.2 Forwarding rule creation

Forwarding Rule Configuration	Note	Sample
Domain name	Forwarding domain name: • Length: 1 to 80 characters. • It cannot begin 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 only one default domain name.	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, 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. However, the CLB instance and the real server still use the HTTP1.x protocol. For more information, see CLB Supports QUIC Protocol.	Enabled
URL	Forwarding URL: • Length: 1-200 characters. • Regular expressions are supported.	/index

	• For detailed configuration rules, see URL Path Forwarding Rules.	
Balancing Method	For HTTPS listeners, CLB supports three scheduling algorithms: weighted round robin (WRR), weighted least connections (WLC), and IP Hash. • Weighted Round Robin (WRR) Algorithm: Requests are distributed to backend servers in sequence based on their weights. This algorithm performs scheduling based on the number of new connections. Servers with higher weights have a higher probability of being polled, 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. • When gRPC is selected as the backend protocol, the gRPC service must be deployed on the real server. The backend forwarding protocol supports gRPC only when HTTP2.0 is enabled and QUIC is disabled.	gRPC
Getting Client IP	Enabled by default.	Enabled
Gzip Compression	Enabled by default.	Enabled

3.3 HTTPS Health Check.

3.4 Session persistence

Session Persistence Configuration	Note	Sample
Session Persistence Switch	• 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
Session persistence duration.	• Session persistence is terminated if there are no new requests in the connection within the specified duration. • Value range: 30–3600 seconds	30s

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:**
> Default port: Enter the **Default Port** first and then select the CVM instance. The port of every CVM instance 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 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: 2023-09-05 14:53:40

Real servers are created and bound to the Cloud Load Balancer (CLB) instance to process corresponding forwarded requests. When configuring a CLB listener, it is necessary to bind real servers. The CLB forwards requests to real servers using various polling methods, which are then processed by the real servers to ensure stable and reliable application performance.

Supported Real Server Types

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

- Instance types include Cloud Virtual Machine (CVM), Elastic Network Interface (ENI), and Elastic Kubernetes Service (EKS) instances.
- Real servers of the IP type are mainly used to bind private IP addresses across multiple VPCs and private IP addresses in IDCs.

Supports and Limits

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

- Enable session persistence to allow CLB to maintain a longer TCP connection for reuse by multiple requests, thereby reducing load on the web server and improving CLB throughput.
- Ensure that the security group for the real server has inbound rules for CLB listener ports and health check ports. For more information, see Configuring Security Groups for Backend Cloud Virtual Machines.

Documentation

- Managing Real Servers
- Binding an ENI
- Binding Container Instances
- Hybrid Cloud Deployment
- Binding with Serverless Cloud Function (SCF)

Managing Real Servers

Last updated: 2023-09-05 14:57:06

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.

Preparations

You must have created a CLB instance and configured a listener. For more information, see CLB Quick Start Guide.

Instructions

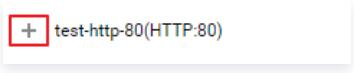
Adding a Real Server to a CLB Instance

Note:

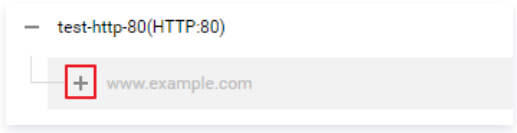
- If a CLB instance is associated with an Auto Scaling group, the CVM instances in that group will be automatically added to the real servers of the CLB. If a CVM instance is removed from the Auto Scaling group, it will be automatically deleted from the real servers of the CLB.
- To add a real server to a CLB instance using the API, see the Bind Real Server to CLB API documentation.
- If your account type is a traditional account and the ISP of the instance is China Mobile, China Telecom, or China Unicom, you can only bind CVMs with traffic-based billing or bandwidth package billing. For more information on account types, see Determining Account Type. For more information on ISP types, see ISP Types.

1. Log in to the Cloud Load Balancer console.
2. On the "Instance Management" page, under the "Cloud Load Balancer" tab, click **Configure Listener** in the operation column to the right of the target CLB instance.
3. On the listener configuration page, select a listener to bind to the backend CVM.
   - **HTTP/HTTPS Listener**

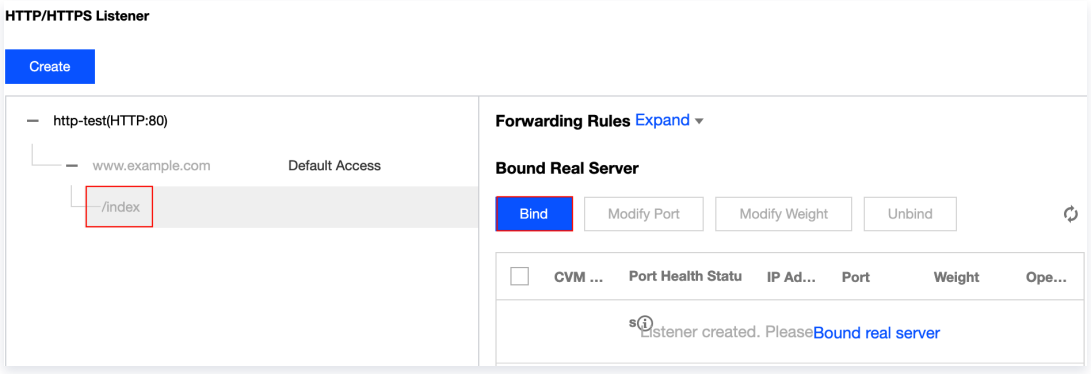
     3.1.1 In the **HTTP/HTTPS Listener** section, click + on the left of the listener you select.

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

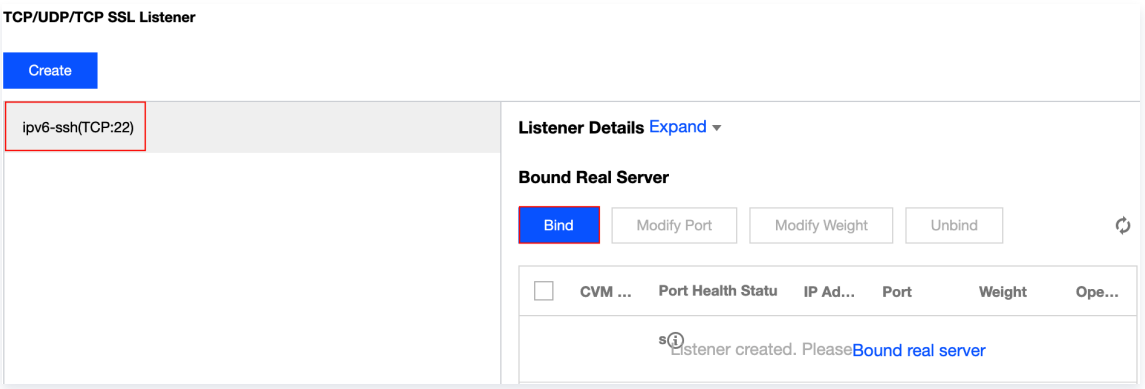
     3.1.3 Select the expanded URL path and click **Bind**.

   - **TCP/UDP/TCP SSL Listener**

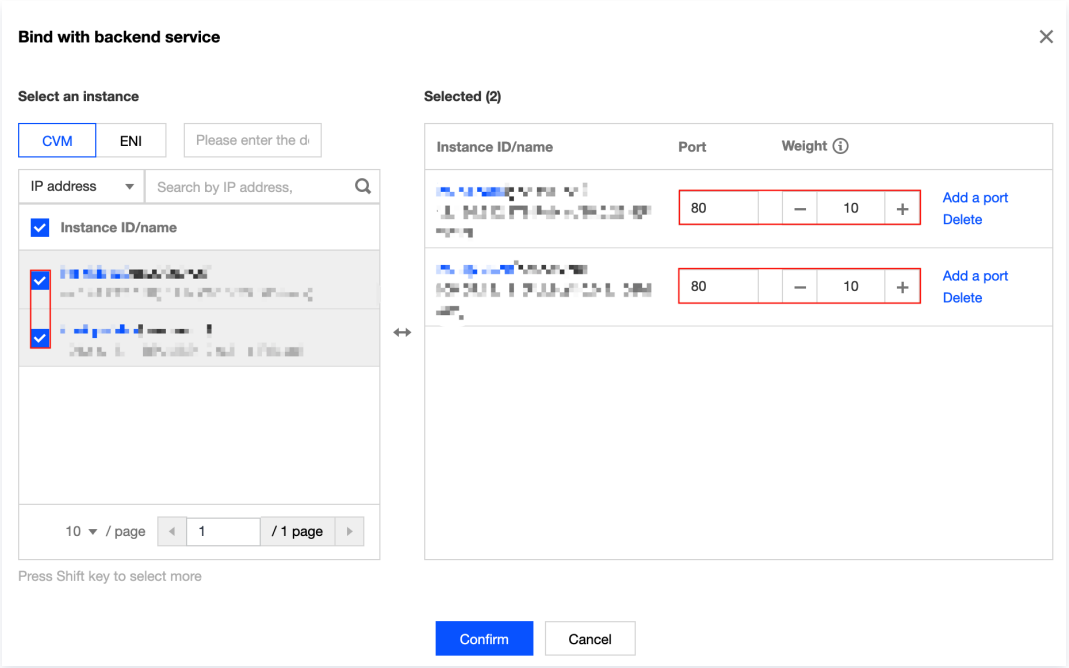
     In the left list of the TCP/UDP/TCP SSL listener module, select the listener that needs to bind the backend Cloud Virtual

Machine and click **Bind**.



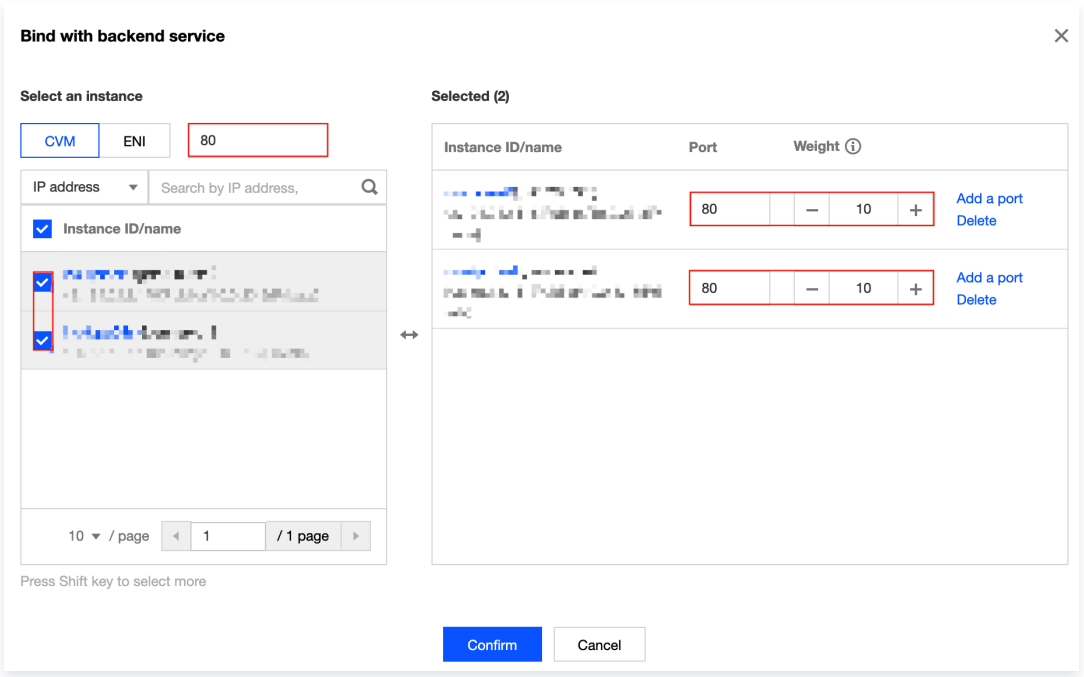
4. Bind a real server to the CLB instance.
   - **Method 1**: In the "Bind Real Server" pop-up window, click **Cloud Virtual Machine**, select the Cloud Virtual Machine(s) to be associated (multiple selections allowed), fill in the required forwarding ports and weights for the selected Cloud Virtual Machine(s). For more information on common server ports, see Server Common Ports. Click **Confirm**.

> **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 greater the weight, the more requests are forwarded. Default is 10, with a configurable range of 0 – 100. When the weight is set to 0, the server will no longer accept new requests. Enabling session persistence may result in uneven distribution of requests among backend servers. For more information, see Load Balancing Algorithm Selection and Weight Configuration Examples.



   - **Method 2**: To bind multiple servers with the same default port value, click **Cloud Virtual Machine** in the "Bind Real Server" pop-up window, enter the default port value (for port selection, see Common Server Ports), select the relevant servers and

set their weights, and click **Confirm**.

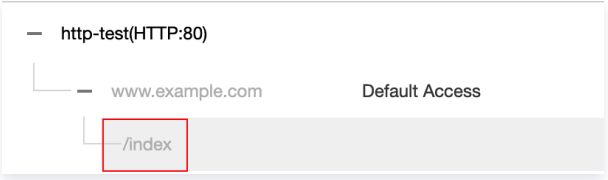


Modifying Real Server Weight for a CLB Instance

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:**
> - To modify the weight of a real server in a CLB instance using an API, see ModifyLoadBalancerBackendServersWeight API documentation.
> - For more information about the weight of real servers in CLB, see CLB Load Balancing Algorithms.

1. Log in to the Cloud Load Balancer console.
2. On the "Instance Management" page, under the "Cloud Load Balancer" tab, click **Configure Listener** in the operation column to the right of the target 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.

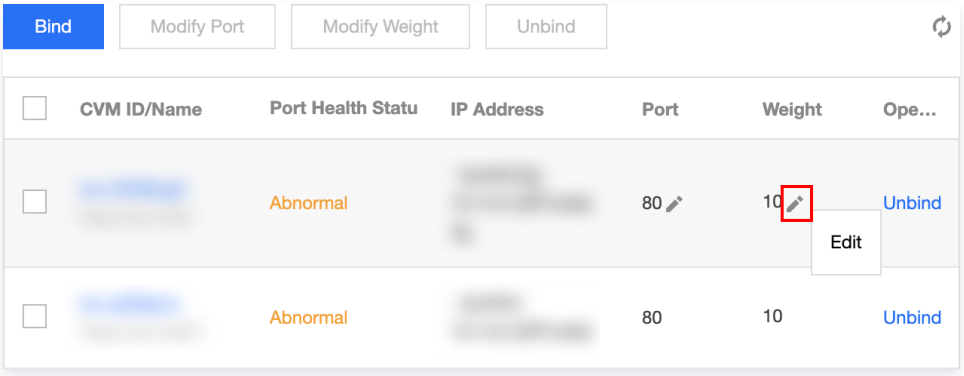


4. In the server list on the right of the HTTP/HTTPS listener section, modify the corresponding server weight.

> **Note:**
> The larger the weight, the more requests are forwarded. The default value is 10, with a configurable range of 0 – 100. When the weight is set to 0, the server will no longer accept new requests. Enabling session persistence may result in uneven distribution of requests among backend servers. For more details, see Load Balancing Algorithm Selection and Weight Configuration Examples.

- **Method 1**: Modify the weight of a specific server individually.

  4.1.1 Locate the server for which you want to modify the weight. Hover your mouse over the corresponding weight and click the button.

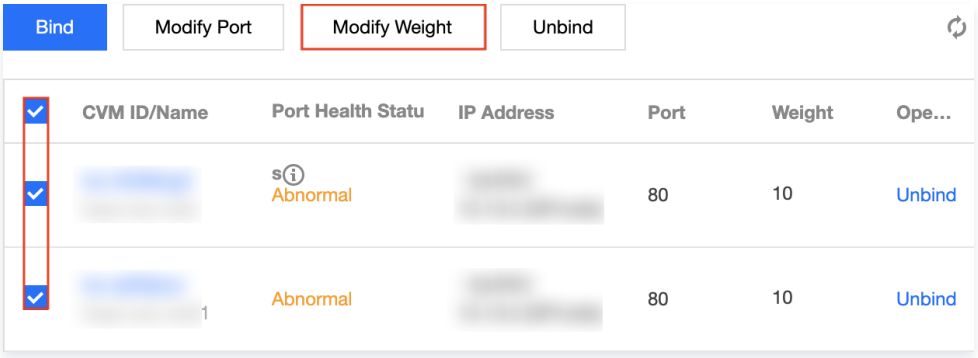


4.1.2 In the "Modify Weight" pop-up window, enter the updated weight value and click **Submit**.

- **Method 2**: Batch modify the weight of certain servers.

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

4.1.1 Click the checkbox in front of the server to select multiple servers, and then click **Modify Weight** at the top of the list



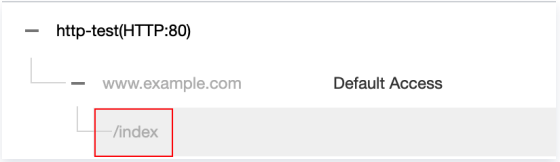
4.1.2 In the "Modify Weight" pop-up window, enter the updated weight value and click **Submit**.

Modifying Real Server Port for a CLB Instance

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).

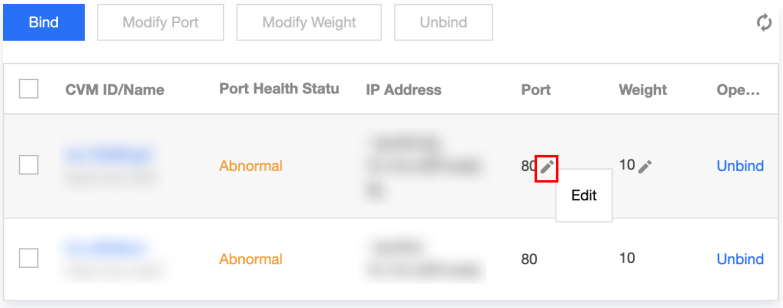
> **Note:**
> To modify the real server port of a CLB listener using the API, please refer to the Modify Listener's Real Server Port API documentation.

1. Log in to the Cloud Load Balancer console.
2. On the "Instance Management" page, under the "Cloud Load Balancer" tab, click **Configure Listener** in the operation column to the right of the target 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 server list on the right of the HTTP/HTTPS listener section, modify the corresponding server port. For port selection, please refer to Common Server Ports.
   - **Method 1**: Modify the port of a specific server individually.

4.1.1 Locate the server whose port you want to modify, hover your mouse over the corresponding port, and click the Edit button.

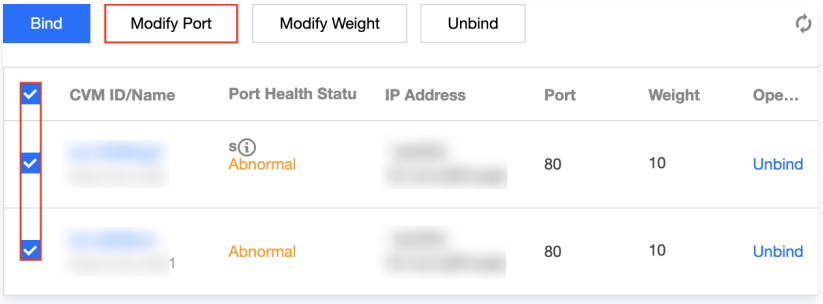


4.1.2 In the "Modify Port" pop-up window, enter the new port value and click **Submit**.

- **Method 2**: Batch modify ports for specific servers.

> **Note:**
> After you perform batch modification, the backend CVMs will use the same port.

4.1.1 Click the checkbox in front of the server to select multiple servers, and then click **Modify Port** at the top of the list.



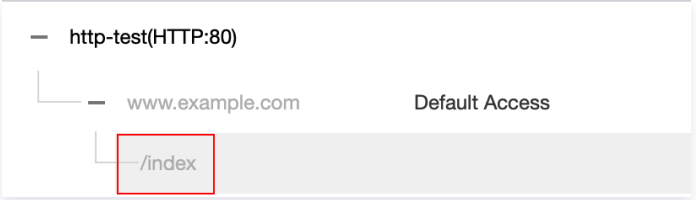
4.1.2 In the "Modify Port" pop-up window, enter the new port value and click **Submit**.

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).

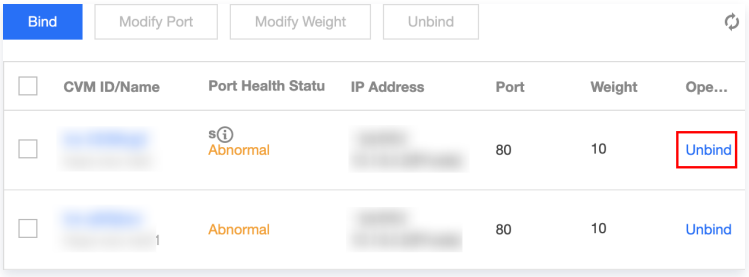
> **Note:**
> - Unbinding a real server will unbind the CLB instance from the CVM instance, and CLB will immediately stop forwarding requests to it.
> - Unbinding a real server will not affect the lifecycle of your CVM instance, which can also be added to the real server cluster again.
> - To unbind a real server from a CLB listener using the API, please refer to the Unbinding Real Servers from CLB Listeners API documentation.

1. Log in to the Cloud Load Balancer console.
2. On the "Instance Management" page, under the "Cloud Load Balancer" tab, click **Configure Listener** in the operation column to the right of the target 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.



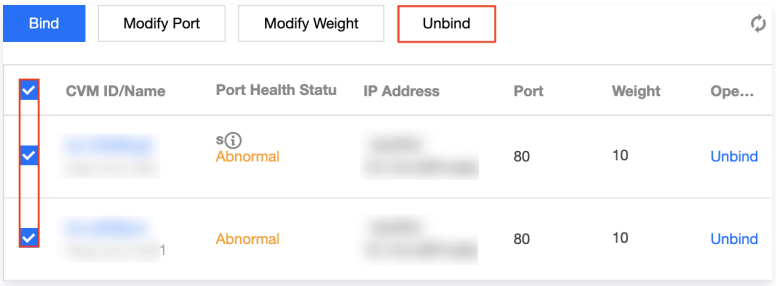
4. In the list on right of the HTTP/HTTPS listener section, unbind the bound real server.
   - **Method 1**: Unbind a specific server individually.

     4.1.1 Locate the server you want to unbind, and click **Unbind** in the operation column on the right.



     4.1.2 In the "Unbind" pop-up window, confirm the service to be unbound and click **Submit**.
   - **Method 2**: Unbind multiple servers in bulk.

     4.1.1 Click the checkbox in front of the server(s) to select multiple servers, and then click **Unbind** at the top of the list.



     4.1.2 In the "Unbind" pop-up window, confirm the service to be unbound and click **Submit**.

Binding an ENI

Last updated: 2023-09-05 14:58:09

ENI Overview

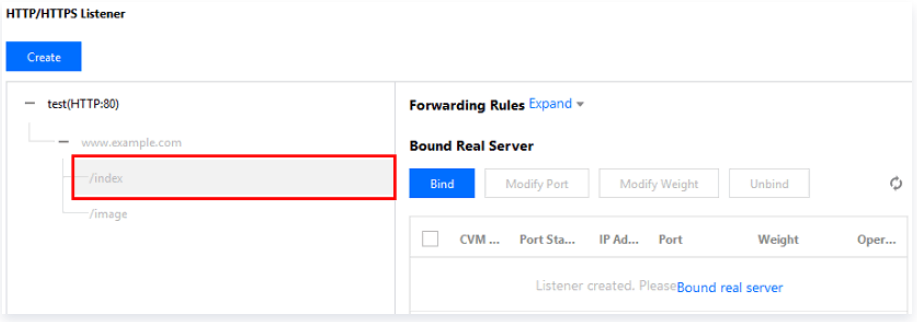
Elastic Network Interface (ENI) is a virtual network card that can be bound to CVM instances within a Virtual Private Cloud. ENIs can be freely migrated between CVM instances in the same Virtual Private Cloud and availability zone, enabling the construction of high-availability clusters, low-cost failover, and fine-grained network management.
CLB backend services support both CVM and ENI, meaning that CLB can bind to both CVM and ENI instances. CLB communicates with backend services via the private network, and when CLB is bound to multiple CVM and ENI instances, the traffic is forwarded to the private IPs of the CVM and ENI instances.

Preparations

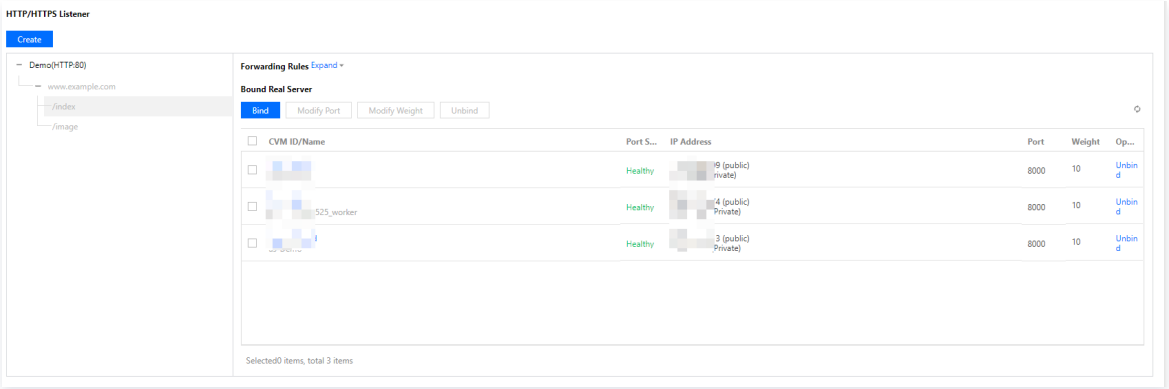
An ENI must be bound to a CVM instance first before it can be bound to a CLB instance. As a CLB instance only forwards traffic as a load balancer but does not process the business logic, the CVM instance, as a computing resource, is needed to process user requests. Please log in to the ENI Console to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

Instructions

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** to select the desired backend server and configure the server port and weight in the pop-up window. You can choose either "Cloud Virtual Machine" or "Elastic Network Interface" when binding backend services:
   - 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.
4. The specific configuration after binding is as shown below:



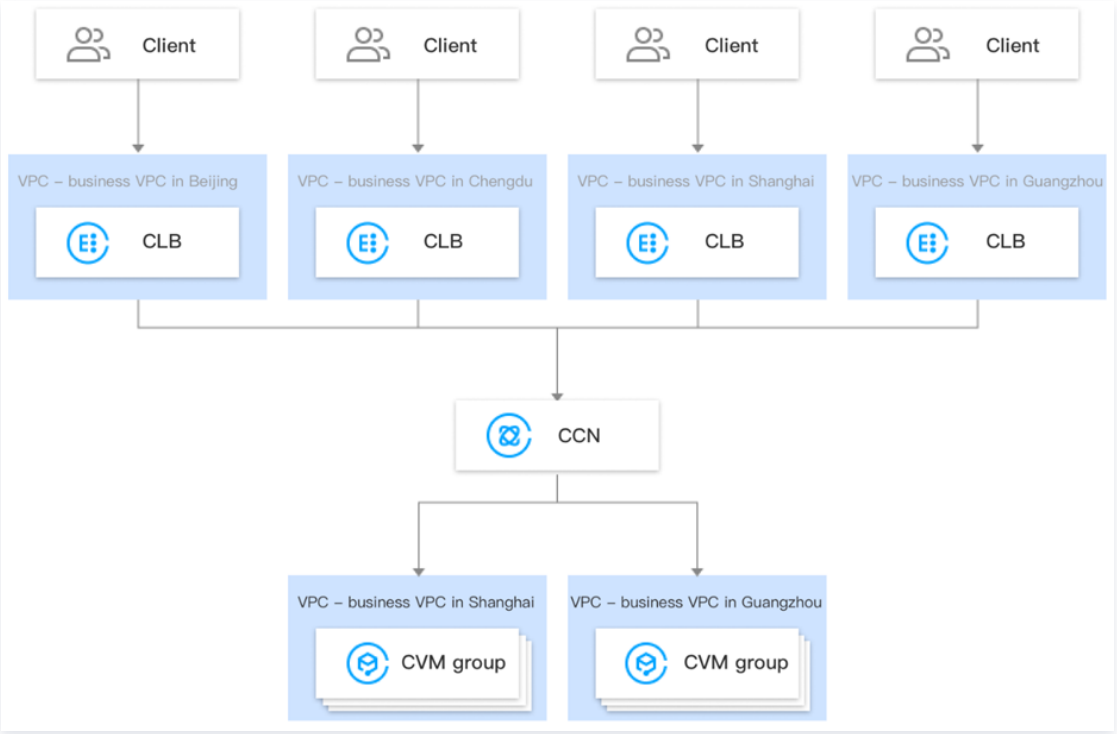
Cross-Region Binding 2.0 (New)

Last updated: 2023-09-05 15:25:11

Cloud Load Balancer (CLB) supports binding real servers across regions through Cloud Connect Network (CCN), allowing customers to select real servers in various regions and bind them across VPCs and regions.
Currently, this feature is in beta testing. If you wish to experience this functionality, please submit a beta test application.

Scenarios

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. In financial transaction and order payment scenarios, the transfer quality and consistency of key business data can be effectively guaranteed.



Differences from Legacy Cross-region Binding

Comparison Item	Cross-Region Binding 2.0 (New)	Cross-region Binding 1.0 (Legacy)
Whether binding to services in multiple regions at the same time is supported	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 and intra-region	Yes:On the new version, the original intra-region binding can be switched back after cross-region binding is used.	No:On the legacy version, after the real server region attribute is modified for cross-region binding, if the new region is different from that of the CLB instance, you cannot change it

		back to the original intra-region binding.
Supported CLB types	Public network CLB instances and private network CLB instances	Public network CLB.
Whether CLB is automatically unbound when a CVM instance is released	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. 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.	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. 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.
Whether the price is favorable	Billed in CCN. The costs are controlled in a fine-grained manner, which leads to lower prices.	Billed by daily 95th percentile.

Limits

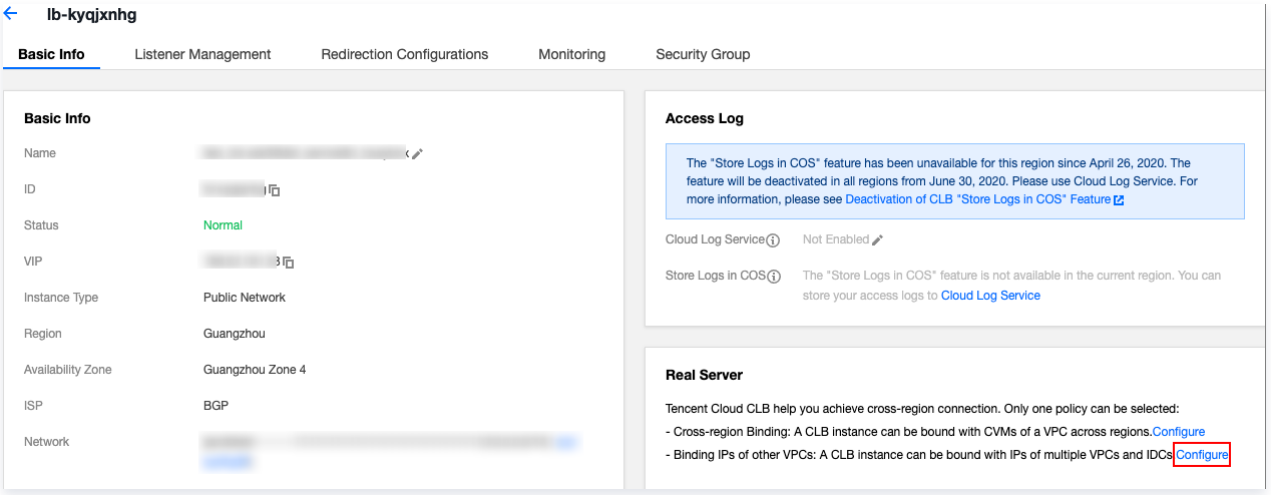
- Cross-network real server binding is currently unavailable for classic CLB instances.
- This feature is available only to bill-by-IP accounts. To check your account type, see Checking Account Type.
- This feature is only supported by VPC but not by classic networks.
- Binding will fail if the VPC IP range of the Cloud Load Balancer instance overlaps with the VPC IP range of the backend service.
- Both IPv4 and IPv6 NAT64 versions of Cloud Load Balancer instances support this feature. For IPv6 instances, dual-stack binding must be enabled. Once enabled, the layer-7 listener can simultaneously bind IPv4 and IPv6 backend servers. When the layer-7 listener binds IPv4 IPs, cross-region binding 2.0 and hybrid cloud deployment are supported. However, when IPv6 instances bind IPv6 backend servers, cross-region binding 2.0 and hybrid cloud deployment are not supported.
- 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) Cloud Load Balancer (CLB) services support obtaining 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.

Preparations

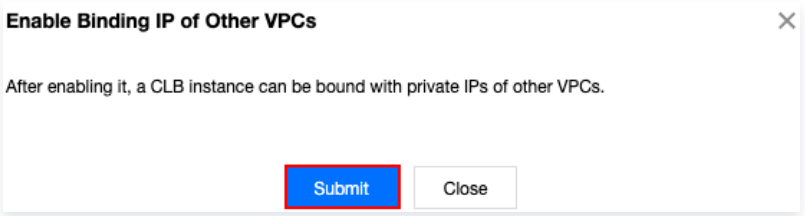
1. You have submitted the application for beta test eligibility. For cross-region binding in Chinese mainland, submit a ticket for application. For cross-region binding outside Chinese mainland, contact your Tencent Cloud rep.
2. You have created a Cloud Load Balancer instance. For more information, see Creating Cloud Load Balancer Instances.
3. You have created a CCN instance. For more information, see Creating a CCN Instance.
4. Associate the target VPC with the created CCN instance. For more information, see Associating Network Instances.

Instructions

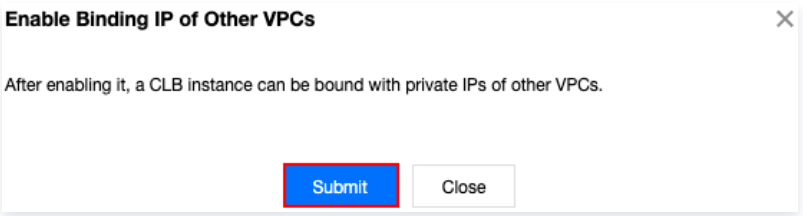
1. Log in to the Cloud Load Balancer console.
2. Select the region from the top left corner of the **Instance Management** page, find the target instance in the list, and click on the **Instance ID**.
3. In the "Backend Services" section of the "Basic Information" page, click **Configure** to bind an internal IP that is not in the current VPC.



4. Click **Submit** in the pop-up dialog box for enabling non-local VPC IPs.



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, click the "Listener Management" tab, and in the listener configuration section, bind backend services to the Cloud Load Balancer instance. For more information, see Adding Cloud Load Balancer Backend Cloud Virtual Machines.
7. In the pop-up "Bind Backend Service" dialog box, select "Other VPC", click **Cloud Virtual Machine**, choose the Cloud Virtual Machines to be associated (multiple selections allowed), and fill in the required forwarding ports and weights for the selected Cloud Virtual Machines. For more information, see Common Server Ports. Click **Confirm** to proceed.
8. Now in the **Bound Real Servers** section, you can view the bound CVM instances of other regions.

Documentation

Cross-region Binding Billing Details

CLB Instance Cross-Region Binding

Last updated: 2023-09-05 15:33:19

At present, public Cloud Load Balancer (CLB) supports binding Cloud Virtual Machines (CVMs) across regions, allowing you to select CVMs from other regions and bind them across VPCs and regions. To experience this feature, for cross-region binding within the Chinese mainland, submit a ticket; for cross-region binding outside the Chinese mainland, contact us.

Note:
- Private network CLB and classic CLB currently are not available for cross-region CVM instance binding.
- For traditional account types, only CLB instances with shared bandwidth package as their network billing mode support cross-region binding 1.0 feature.

Scenarios

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. In financial transaction and order payment scenarios, the transfer quality and consistency of key business data can be effectively guaranteed.

Instructions

1. Log in to the Cloud Load Balancer console.
2. On the **Instance Management** page, click the ID of the target CLB instance.
3. On the **Basic Info** page, in the **Real Server Module**, click **Cross-region Binding 1.0** followed by **Configure** to modify the region and network attributes of the backend Cloud Virtual Machines.

   Note:
   If a public Cloud Load Balancer is already bound to a Cloud Virtual Machine in the same region, you need to unbind the CVM before switching regions. For more information, please refer to Unbinding a CLB Real Server.

4. In the **Modify Backend Service Configuration** pop-up window, select the desired region and network from the **Backend Service Region** and **Backend Service Network** lists, then click **Submit**.

   Note:
   - Currently, a Cloud Load Balancer (CLB) instance can only be bound to Cloud Virtual Machines (CVMs) in a single 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 both the Beijing and Shanghai regions simultaneously.
   - After modifying the backend instance service attributes, if the new region is different from the CLB region, you will be unable to revert to the original intra-region binding.
   - Currently, binding CLB and CVM instances across VPCs within the same region is not allowed.
   - Supports scenarios across Classic Network and VPC.
   - The bandwidth fees incurred by cross-region binding will be settled daily using peak bandwidth tiered billing. For more information, see Cross-region Binding Billing Description.

Billing description

The cross-region binding feature is implemented based on the principle of cross-region peering connections. For billing details, please refer to Billing Description.

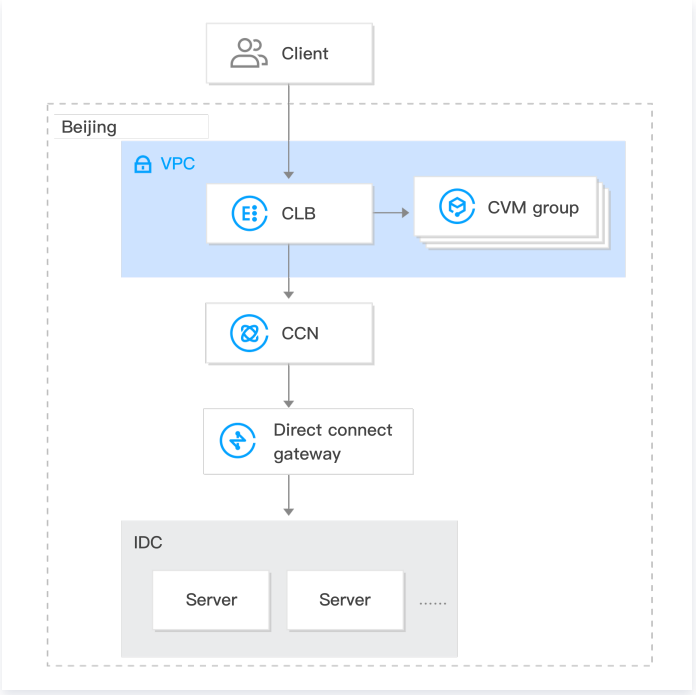
Hybrid Cloud Deployment

Last updated: 2025-05-07 18:18:36

In a hybrid-cloud deployment scenario, Cloud Load Balancer can be directly bound to IPs within local data centers (IDC), facilitating the binding of backend servers across VPCs and IDCs.
Currently, this feature is in the beta testing phase. If you wish to experience this functionality, please apply for beta testing for cross-region binding within the country, and submit a business application for cross-region binding outside the country.

Solution strengths

- Rapidly construct a hybrid cloud with seamless connections between cloud and on-premises resources. Cloud Load Balancer can simultaneously forward requests to servers within VPCs on the cloud and Cloud Virtual Machines in local IDCs.
- Utilize the high-quality public network access capabilities of Tencent Cloud.
- Leverage the rich feature set of Tencent Cloud's Cloud Load Balancer, such as layer-4/7 access, health checks, 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 unavailable 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 only supported by VPC but not by classic networks.
- Both IPv4 and IPv6 NAT64 versions of Cloud Load Balancer instances support this feature. For IPv6 instances, dual-stack binding must be enabled. Once enabled, the layer-7 listener can simultaneously bind IPv4 and IPv6 backend servers. When the layer-7 listener binds IPv4 IPs, cross-region binding 2.0 and hybrid cloud deployment are supported. However, when IPv6 instances bind IPv6 backend servers, cross-region binding 2.0 and hybrid cloud deployment are not supported.
- 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.
- 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.

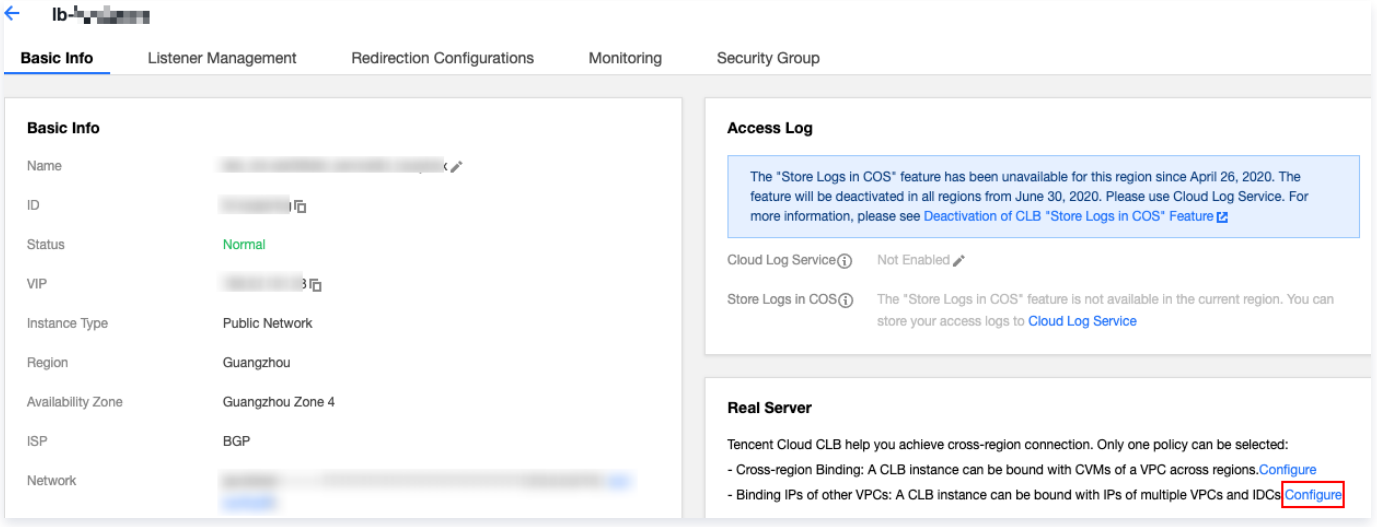
- TCP and TCP SSL listeners need to use TOA on the real server to get the source IP. For more information, see Obtaining Real Client IPs via TOA in Hybrid Cloud Deployment.
- HTTP and HTTPS listeners need to use X-Forwarded-For (XFF) to obtain the source IP.
- UDP listeners cannot get the source IP.

Preparations

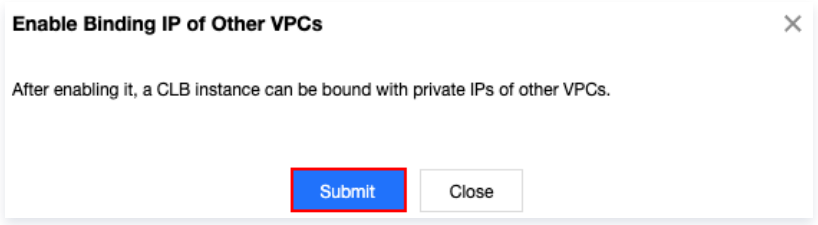
1. You have submitted the application for beta test eligibility. For cross-region binding in Chinese mainland, submit a ticket for application. For cross-region binding outside Chinese mainland, contact your Tencent Cloud rep.
2. You have created a Cloud Load Balancer instance. For more information, see Creating Cloud Load Balancer Instances.
3. You have created 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.

Instructions

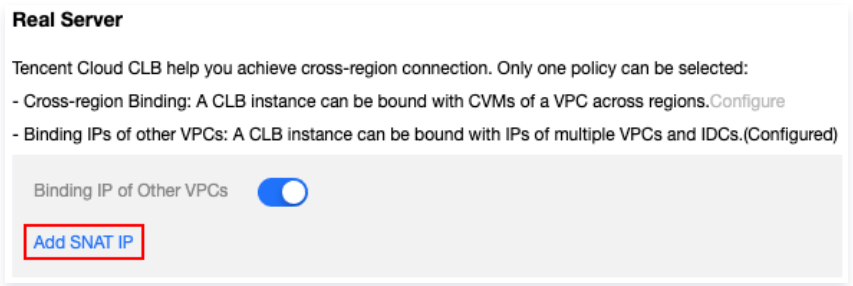
1. Log in to the Cloud Load Balancer console.
2. On the **Instance Management** page, click the ID of the target CLB instance.
3. In the "Backend Services" section of the "Basic Information" page, click **Configure** to bind an internal IP that is not from the current VPC.



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



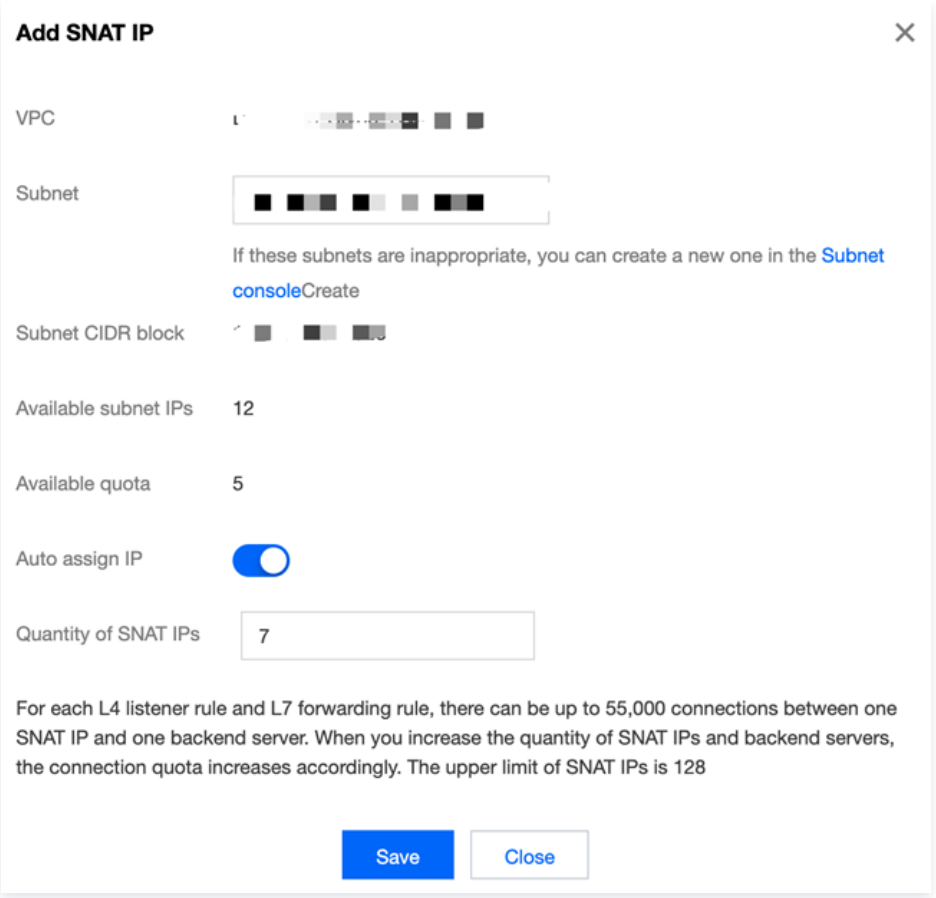
5. On the "Basic Information" page, in the "Backend Services" section, click **Add SNAT IP**.



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

Note:

- A SNAT IP is primarily used in hybrid cloud deployment scenarios for forwarding requests to servers within the IDC. When using Cloud Load Balancer to bind an interconnected IDC IP through Cloud Connect Network, a SNAT IP must be assigned. The SNAT IP serves as the private IP of your VPC.
- A maximum of 10 SNAT IPs can be configured for each CLB instance.
- For a single CLB instance with one rule configured with one SNAT IP and one backend service, the maximum number of connections is 55,000. If you increase the number of SNAT IPs or backend services, the number of connections increases proportionally. For example, if a CLB instance has two SNAT IPs configured and ten backend ports bound, the total number of connections for that instance would be: 2 x 10 x 55,000 = 1,100,000. You can assess the allocation of SNAT IPs based on the number of connections.
- Note that deleting a SNAT IP disconnects all connections on the IP. Please exercise caution.



7. On the instance details page, click the "Listener Management" tab, and in the listener configuration section, bind backend services to the Cloud Load Balancer instance. For more information, see Adding Cloud Load Balancer Backend Cloud Virtual Machines.
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.

Documentation

Cross-Region Binding 2.0 (New)

Configuring CVM Security Groups

Last updated: 2023-09-05 15:43:22

Overview of CVM Security Group

Cloud Load Balancer's backend Cloud Virtual Machine instances can be access-controlled through Security Groups, serving as a firewall.
You can associate one or more security groups with backend Cloud Virtual Machines and add one or multiple rules to each security group to control traffic access permissions for different servers. You can modify the rules of a security group at any time, and the new rules will automatically apply to all instances associated with that security group. For more information, please refer to the Security Group Operation Guide. In a Virtual Private Cloud environment, you can also use Network ACLs 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 open the CLB VIP to the internet on the backend CVM security group, so that CLB can use the VIP to detect the backend CVM health status.
2. Private network CLB:
   - For private network Cloud Load Balancer (formerly "Application-based private network Cloud Load Balancer"), if your CLB belongs to a VPC network, you need to allow the CLB VIP (used for health checks) on the backend CVM security group; if your CLB belongs to the basic network, there is no need to configure the backend CVM security group, as the health check IP is allowed by default.
   - For classic private network CLB instances created before December 5, 2016, and with a VPC network type, you need to open the CLB VIP (used for health checks) on the backend CVM security group. For other types of classic private network CLB instances, there is no need to configure the backend CVM security group, as the health check IP is allowed by default.

Configuration Sample

The following example demonstrates the configuration of a CVM security group when accessing a CVM through a CLB. If you have also configured a security group on the CLB, please refer to Configuring Cloud Load Balancer Security Group to set up the security group rules on the CLB.

- **Scenario 1:**
  Public network CLB, with a TCP:80 listener configured, and the backend service port set to 8080. If you want to allow only specific client IPs (ClientA IP and ClientB IP) to access the CLB, the inbound rules for the backend server security group should be configured as follows:

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

- **Use Case 2:**
  For a public network CLB with an HTTP:80 listener and a backend service port of 8080, if you want to allow normal access for all client IPs, the inbound rules for the backend server security group should be configured as follows:

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

- **Scenario 3:**
  For private network Cloud Load Balancer (formerly "Application-based Private Network Cloud Load Balancer"), with a VPC network type, you need to allow the CLB VIP in the CVM security group for health checks. Configure a TCP:80 listener for the CLB, with a backend service port of 8080, and allow only Client IP (ClientA IP and ClientB IP) to access the CLB VIP. You also want to restrict the Client IP to access only the backend hosts bound to this CLB.
  a. The backend server security group inbound rules are configured 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
```

- **Scenario 4:**
For traditional private network Cloud Load Balancers purchased after December 5, 2016, in a VPC network, the CVM security group only needs to allow the Client IP (no need to allow the CLB VIP, as the health check IP is allowed by default). Configure a TCP:80 listener for the CLB, with the backend service port set to 8080. You want to allow only Client IP (ClientA IP and ClientB IP) to access the Cloud Load Balancer's VIP and restrict the Client IP to access only the backend hosts bound to this CLB.
a. The backend server security group inbound rules are configured 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
```

- **Use Case 5: Blacklist**
If you need to set up a blacklist for certain client IPs and deny their access, you can achieve this by configuring the security group associated with the cloud service. The security group rules should 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. **Order matters**.
> - Security groups are stateful, so the above configurations are for **inbound rules** only, and no special configuration is required for outbound rules.

Operation Guide of CVM Security Groups

Managing Backend CVM Security Groups Using the Console

1. Log in to the Cloud Load Balancer console and click the corresponding Cloud Load Balancer instance ID to enter the Cloud Load Balancer 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 to bind or unbind security groups.

Managing backend CVM security groups using Tencent Cloud API

Please refer to Bind Security Group API and Unbind Security Group API.

Health check
Configuring Health Check

Last updated: 2023-09-05 16:18:21

You can enable health check when configuring listeners to determine the availability of real servers. For more information on health checks, see Health Check Overview.

Description

- TCP listeners for IPv6 CLB instances do not support HTTP health checks and custom health checks.
- UDP listeners for IPv6 Cloud Load Balancer (CLB) instances do not support custom health checks.

Preparations

1. You have created a Cloud Load Balancer (CLB) instance. For more information, see Creating CLB Instances.
2. Create 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.

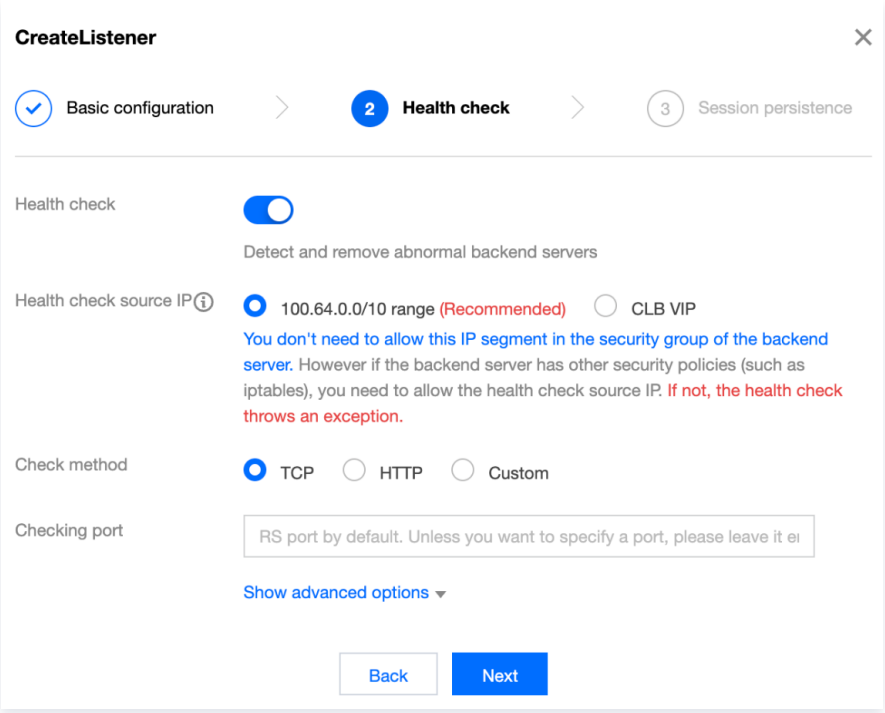
TCP Listener

Layer-4 TCP listeners support three types of health checks: Layer-4 TCP, Layer-7 HTTP, and custom health checks.

- 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. Refer to Preparations and navigate to the "Health Check" tab.
2. On the **Health Check** tab, select **TCP** as the check method.

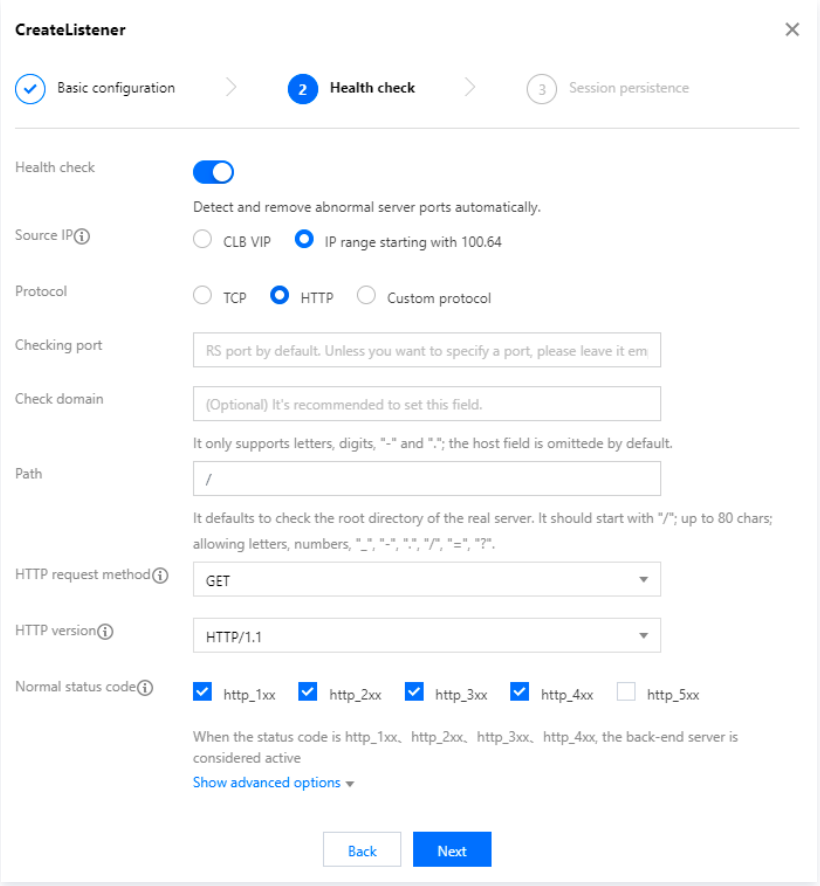


Category	Note
Health check	You can enable or disable health check. We recommend that you enable health check for automatic checks on real servers and removal of abnormal ports.

Health check source IP	The source IP address of health check packets defaults to the 100.64.0.0/10 IP range, which effectively prevents address conflicts. Existing users can choose the VIP of the load balancer as the source IP for health checks, or switch to the 100.64.0.0/10 IP range with a single click using the self-service tool. For more details, please refer to Health Check Source IP Diagnostic Assistant.
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 details, refer to Advanced Options.

Configuring HTTP health check

1. Refer to Preparations and navigate to the "Health Check" tab.
2. On the "Health Check" tab, select "HTTP" as the check method.

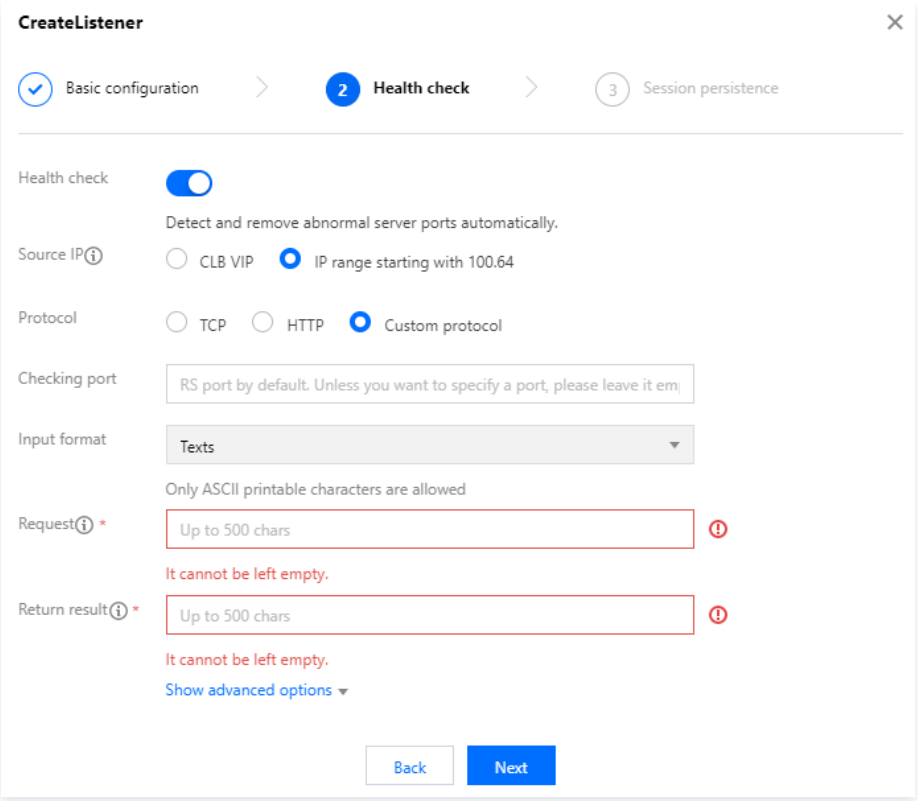


Category	Note
Health check	You can enable or disable health check. We recommend that you enable health check for automatic checks on real servers and removal of abnormal ports.
Health check source IP	The source IP address of health check packets defaults to the 100.64.0.0/10 IP range, which effectively prevents address conflicts. Existing users can choose the VIP of the load balancer as the source IP for health checks, or switch to the 100.64.0.0/10 IP range with a single click using the self-service tool. For more details, please refer to Health Check Source IP Diagnostic Assistant.
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.

The domains to check	Limits on a health check domain name: • Length: 1 to 80 characters. • It is the forwarding domain name by default. • Regular expressions are not supported. If your forwarding domain name is a wildcard one, 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: 1-200 characters. • `/` is the default value and should be the first character. • Regular expressions are not supported. We recommend specifying a fixed URL (static webpage) for the health check. • 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, which can reduce backend overheads and improve request efficiency. The real server must support HEAD. • If GET is selected, the real server must support GET.
HTTP version	HTTP version of the real server. • If the version supported by the real server is HTTP 1.0, then the host field of the request does not need authentication, that is, the check domain does not need to be configured. • If the version supported by the real server is HTTP 1.1, the Host field of the request needs to be verified, that is, the check domain needs to be configured. **Note:** When selecting HTTP/1.1 version, if the check domain name is not configured, the real server will return a 400 error code according to the HTTP standard protocol, indicating a health check exception. It is recommended to select http_4xx as the 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 details, refer to Advanced Options.

Configuring Custom Health Check

1. Refer to Preparations and navigate to the "Health Check" tab.
2. In the "Health Check" tab, select the "Custom" check method.



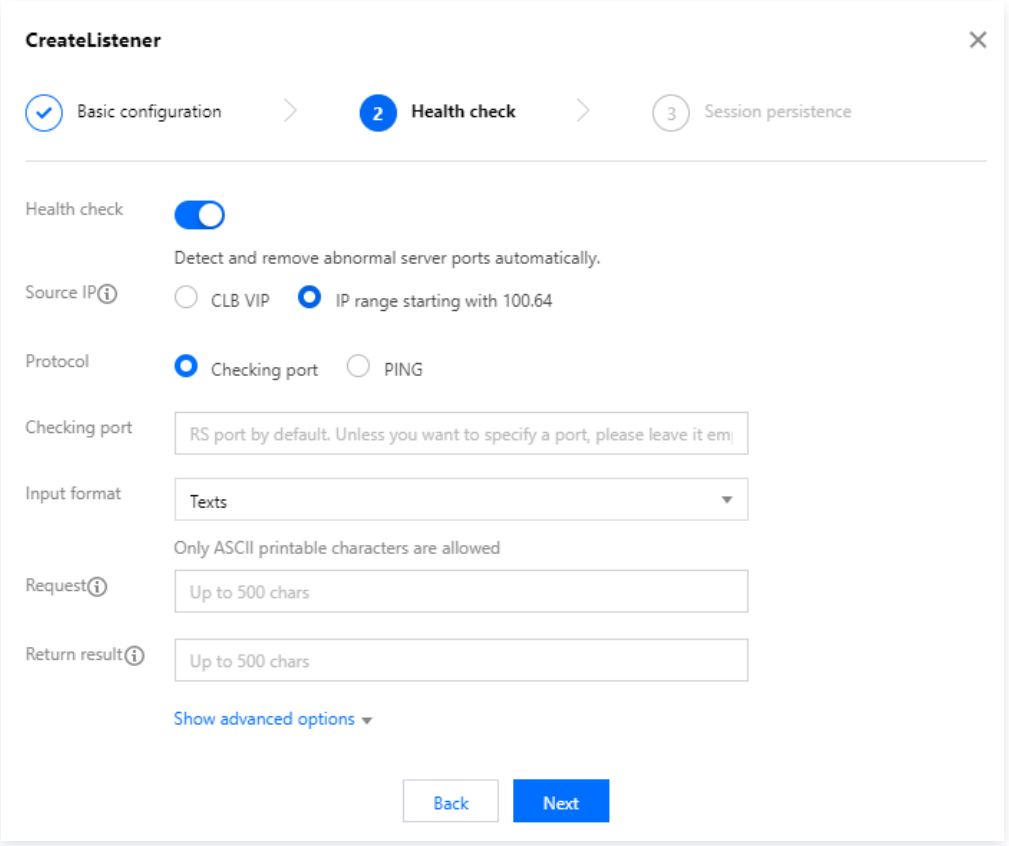
Category	Note
Health check	You can enable or disable health check. We recommend that you enable health check for automatic checks on real servers and removal of abnormal ports.
Health check source IP	The source IP address of health check packets defaults to the 100.64.0.0/10 IP range, which effectively prevents address conflicts. Existing users can choose the VIP of the load balancer as the source IP for health checks, or switch to the 100.64.0.0/10 IP range with a single click using the self-service tool. For more details, please refer to Health Check Source IP Diagnostic Assistant.
Check method	Selecting "Custom" indicates the configuration of custom protocol health checks.
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.
Input formats	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	Custom health check request content is mandatory. For instance, a sample request for probing a DNS service could be: F13E0100000100000000000003777777047465737403636F6D0774656E63656E7403636F6D0000010001.
Return result	When customizing a health check request, it is mandatory to provide the health check response. For instance, the response for a DNS service check could be: F13E.
Show advanced options	For more details, refer to Advanced Options.

UDP Listener

UDP listeners support UDP health checks, which can be conducted by checking ports and running the Ping command.

Configuring Custom Health Check

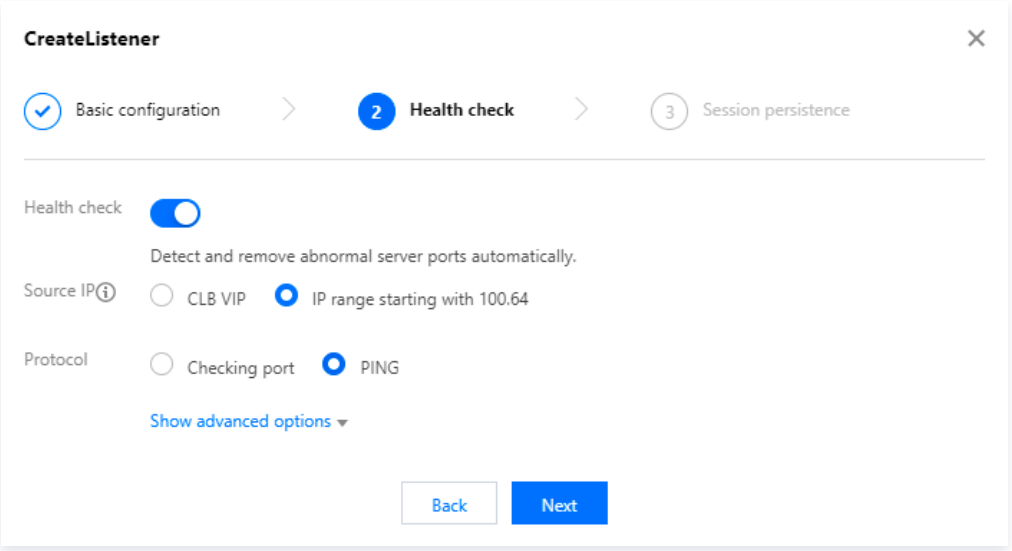
1. Refer to Preparations and navigate to the "Health Check" tab.
2. In the "Health Check" tab, select the "Custom" check method.



Category	Note
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 defaults to the 100.64.0.0/10 IP range, which effectively prevents address conflicts. Existing users can choose the VIP of the load balancer as the source IP for health checks, or switch to the 100.64.0.0/10 IP range with a single click using the self-service tool. For more details, please refer to Health Check Source IP Diagnostic Assistant.
Check method	If **Custom** is selected, UDP detection packets are sent from the health check source IP address to a real server to obtain the status of the real server.
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.
Input formats	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	Customize the health check request content. For example, the check request sample for probing DNS service is: F13E010000010000000000000377777704746573740363 6F6D0774656E63656E7403636F6D0000010001.
Return result	When customizing health check requests, it is essential to configure the health check response. For instance, the response for a DNS service check might be: F13E.
Show advanced options	For more details, refer to Advanced Options.

Configuring PING health check

1. Refer to Preparations and navigate to the "Health Check" tab.
2. On the **Health Check** tab, select **PING** as the check method.

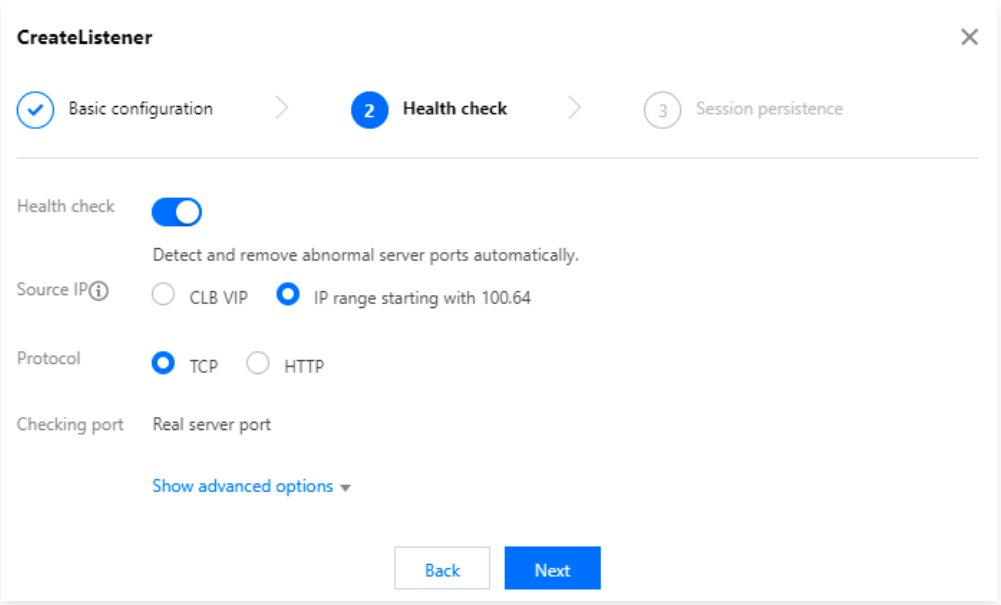


Category	Note
Health check	You can enable or disable health check. We recommend that you enable health check for automatic checks on real servers and removal of abnormal ports.
Health check source IP	The source IP address of health check packets defaults to the 100.64.0.0/10 IP range, which effectively prevents address conflicts. Existing users can choose the VIP of the load balancer as the source IP for health checks, or switch to the 100.64.0.0/10 IP range with a single click using the self-service tool. For more details, please refer to Health Check Source IP Diagnostic Assistant.
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 details, refer to Advanced Options.

TCP SSL Listener

Configuring TCP health check

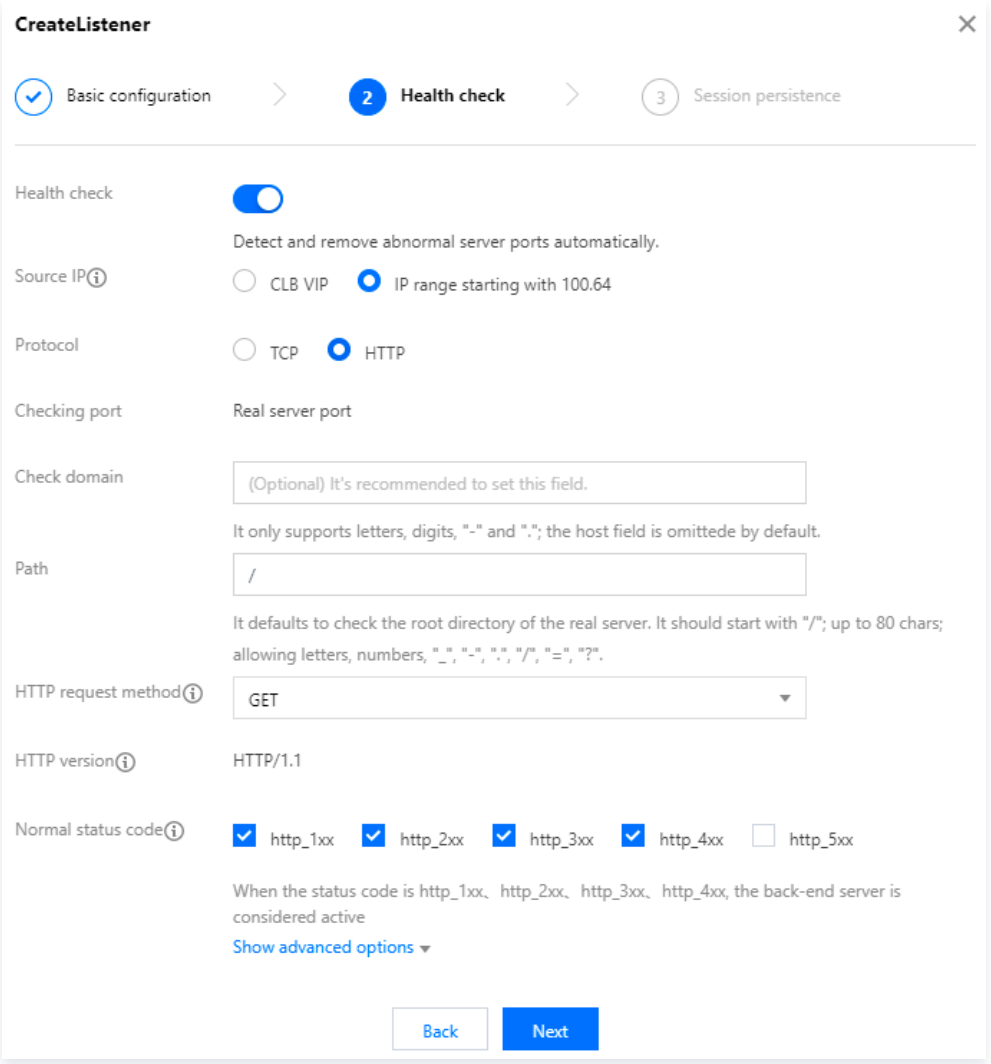
1. Refer to Preparations and navigate to the "Health Check" tab.
2. On the **Health Check** tab, select **TCP** as the check method.



Category	Note
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 defaults to the 100.64.0.0/10 IP range, which effectively prevents address conflicts. Existing users can choose the VIP of the load balancer as the source IP for health checks, or switch to the 100.64.0.0/10 IP range with a single click using the self-service tool. For more details, please refer to Health Check Source IP Diagnostic Assistant.
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 details, refer to Advanced Options.

Configuring HTTP health check

1. Refer to Preparations and navigate to the "Health Check" tab.
2. On the "Health Check" tab, select "HTTP" as the check method.



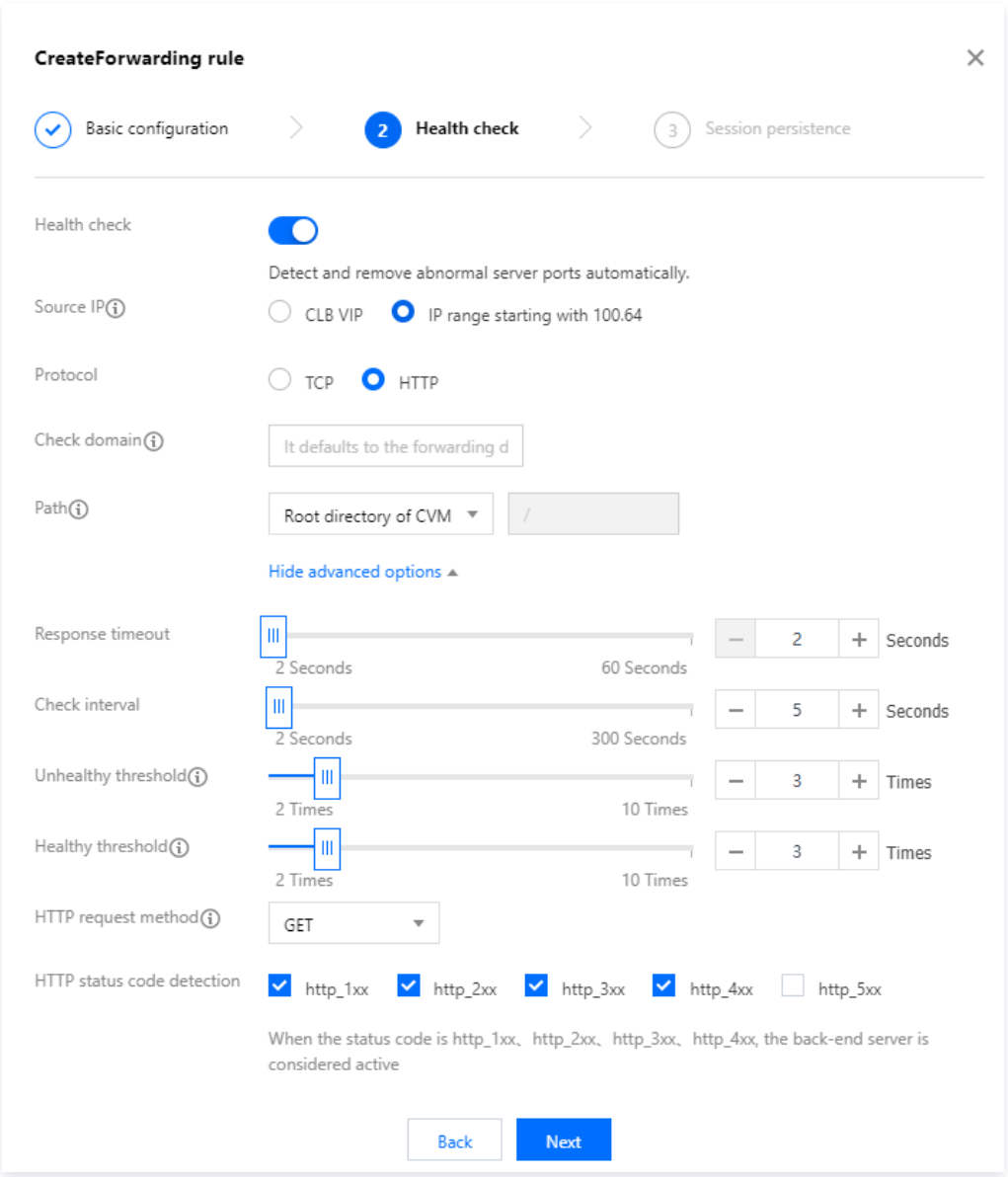
Category	Note
Health check	You can enable or disable health check. We recommend that you enable health check for automatic checks on real servers and removal of abnormal ports.
Health check source IP	The source IP address of health check packets defaults to the 100.64.0.0/10 IP range, which effectively prevents address conflicts. Existing users can choose the VIP of the load balancer as the source IP for health checks, or switch to the 100.64.0.0/10 IP range with a single click using the self-service tool. For more details, please refer to Health Check Source IP Diagnostic Assistant.
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.
The domains to check.	Limits on a health check domain name: • Length: 1 to 80 characters. • It is the forwarding domain name by default. • Regular expressions are not supported. If your forwarding domain name is a wildcard one, 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: 1-200 characters. • `/` is the default value and should be the first character. • Regular expressions are not supported. We recommend specifying a fixed URL (static webpage) for the health check.

	• 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, which can reduce backend overheads and improve request efficiency. The real server must support HEAD. • If GET is selected, the real server must support GET.
HTTP version	The HTTP version of the backend service only supports HTTP1.1. The backend service needs to verify the Host field of the request, which means a check domain needs to be configured. **Note:** If the check domain is not configured, the backend server will return a 400 error code according to the HTTP standard protocol, indicating a health check anomaly. It is recommended to select the normal status code http_4xx.
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 details, refer to Advanced Options.

HTTP Listener

Configuring HTTP health check

1. Refer to Preparations and navigate to the "Health Check" tab.
2. On the "Health Check" tab, select "HTTP" as the check method.

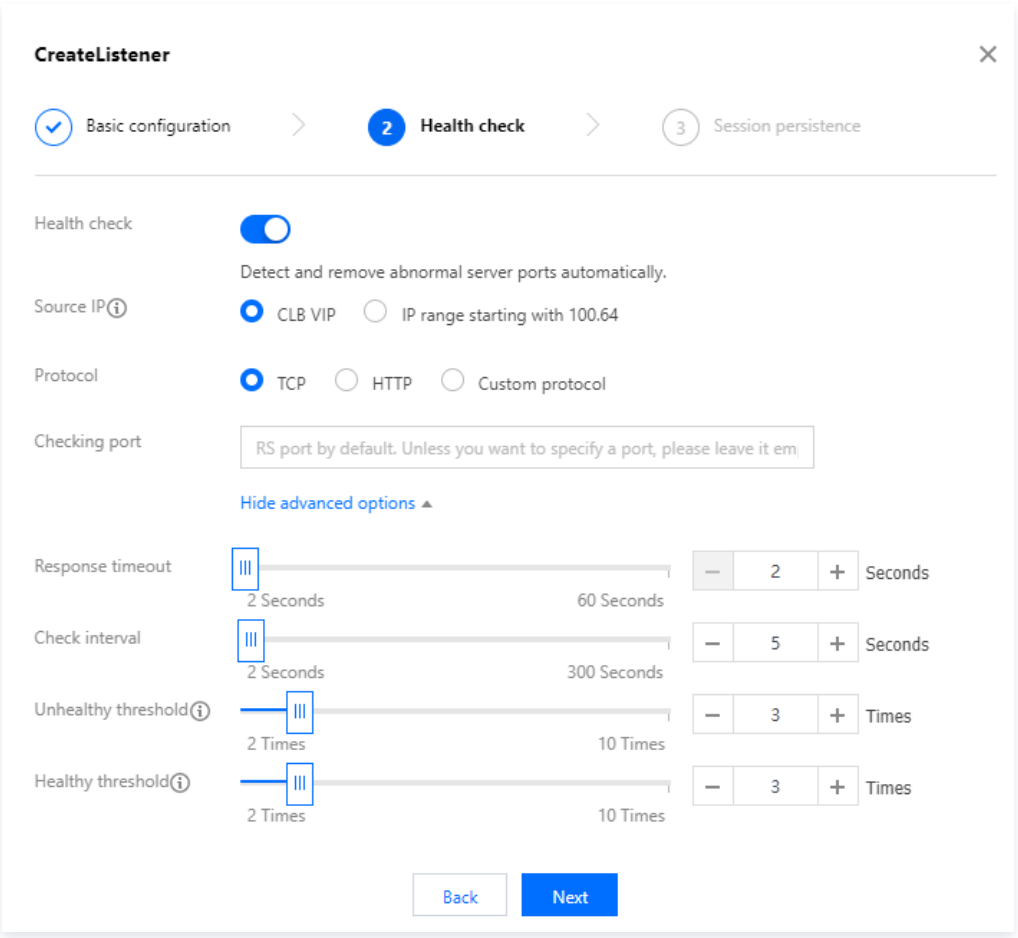


Category	Note
Health check	You can enable or disable health check. We recommend that you enable health check for automatic checks on real servers and removal of abnormal ports.
Health check source IP	The source IP address of health check packets defaults to the 100.64.0.0/10 IP range, which effectively prevents address conflicts. Existing users can choose the VIP of the load balancer as the source IP for health checks, or switch to the 100.64.0.0/10 IP range with a single click using the self-service tool. For more details, please refer to Health Check Source IP Diagnostic Assistant.
The domains to check	Limits on a health check domain name: • Length: 1 to 80 characters. • It is the forwarding domain name by default. • Regular expressions are not supported. If your forwarding domain name is a wildcard one, 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: 1-200 characters. • `/` is the default value and should be the first character.

	• Regular expressions are not supported. We recommend specifying a fixed URL (static webpage) for the health check. • 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, which can reduce backend overheads and improve request efficiency. The real server must support HEAD. • If GET is selected, the real server must support GET.
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.

Configuring TCP health check

1. Refer to Preparations and navigate to the "Health Check" tab.
2. On the **Health Check** tab, select **TCP** as the check method.



Category	Note
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 defaults to the 100.64.0.0/10 IP range, which effectively prevents address conflicts. Existing users can choose the VIP of the load balancer as the source IP for health checks, or switch to the 100.64.0.0/10 IP range with a single click using the self-service tool. For more details, please refer to Health Check Source IP Diagnostic Assistant.
Check method	TCP health checks are conducted if **TCP** is selected.
Show advanced options	For more details, refer to 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 configuring health checks for HTTPS listeners, refer to the HTTP Listener health check mentioned above.

Advanced Options

Health Check Configuration	Note	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
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

Documentation

- Health Check Overview
- Configure Alarm Policy

Setting 100.64.0.0/10 IP 69Range as the Health Check IP

Last updated: 2023-09-05 16:21:47

This document illustrates how to configure the health check source IP address of a CLB instance from the CLB VIP to the 100.64.0.0/10 IP range, using a TCP listener as an example.

Use Cases

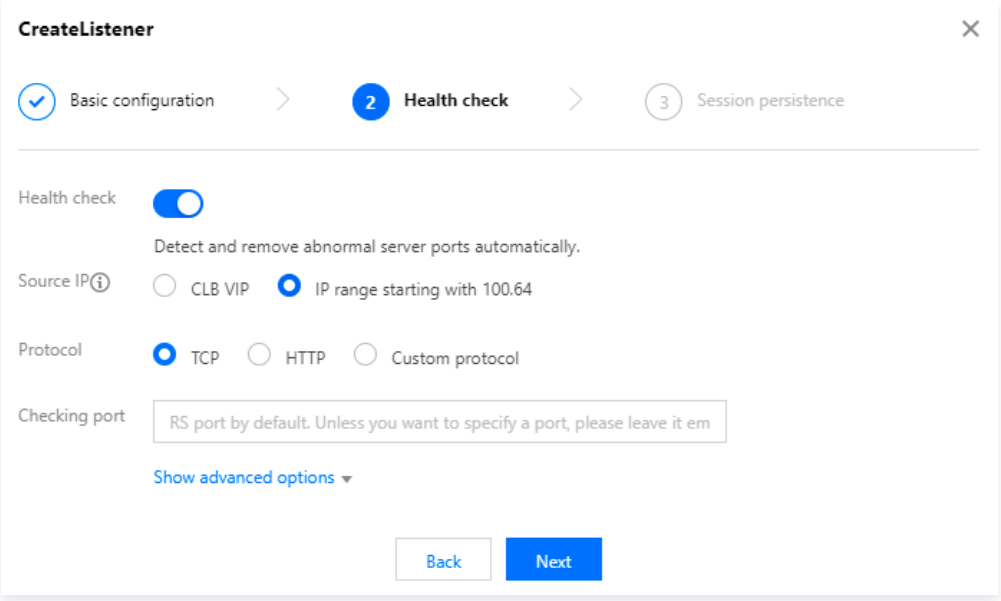
1. **Aggregating backend server security groups**
   The health check source IP is aggregated into the 100.64.0.0/10 IP range.
2. **Resolving the loopback issue in self-built Kubernetes clusters**
   K8s services need to be exposed both within and outside the cluster. Within the cluster, this is achieved through the internal Cloud Load Balance (IPVS), while outside the cluster, it is implemented through the private network Cloud Load Balance (CLB). IPVS binds the private network CLB's IP address to a local interface, so when accessing the private network CLB within the cluster, the IPVS Cloud Load Balance is used.

   In Tencent Kubernetes Engine (TKE), the private network CLB uses the CLB VIP address as the health check source IP, which conflicts with the IPVS-bound address in the native K8s implementation, leading to health check failures for the private network CLB.

   By setting the health check source IP to the 100.64.0.0/10 IP range, address conflicts can be avoided, and health check failures can be resolved.

Instructions

1. Log in to the Cloud Load Balancer 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**.



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.

Documentation

- Configuring Health Checks
- Health Check Probe Identifier

Health Check Source IP Diagnosis Assistant

Last updated: 2023-09-05 16:21:55

This document explains how to quickly diagnose whether the health check source IP of CLB instances under an account is using the 100.64.0.0/10 IP range. With this self-help tool, you can also easily switch the health check source IP from the CLB VIP to the 100.64.0.0/10 IP range with a single click.

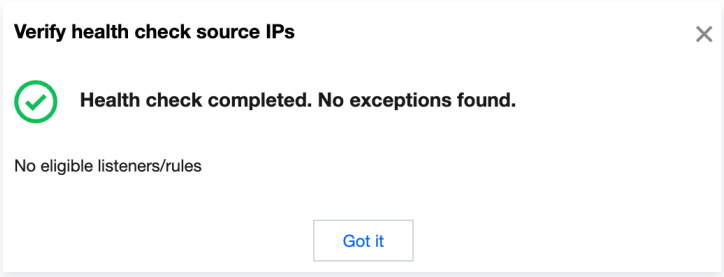
Use Cases

Diagnose the health check source IP to quickly determine whether the health check source IP of CLB instances under an account is using the 100.64.0.0/10 IP range and support one-click switching.

> **Note:**
> It is recommended to first select 1-2 instances for diagnosis and switching to verify functionality before proceeding with batch operations.

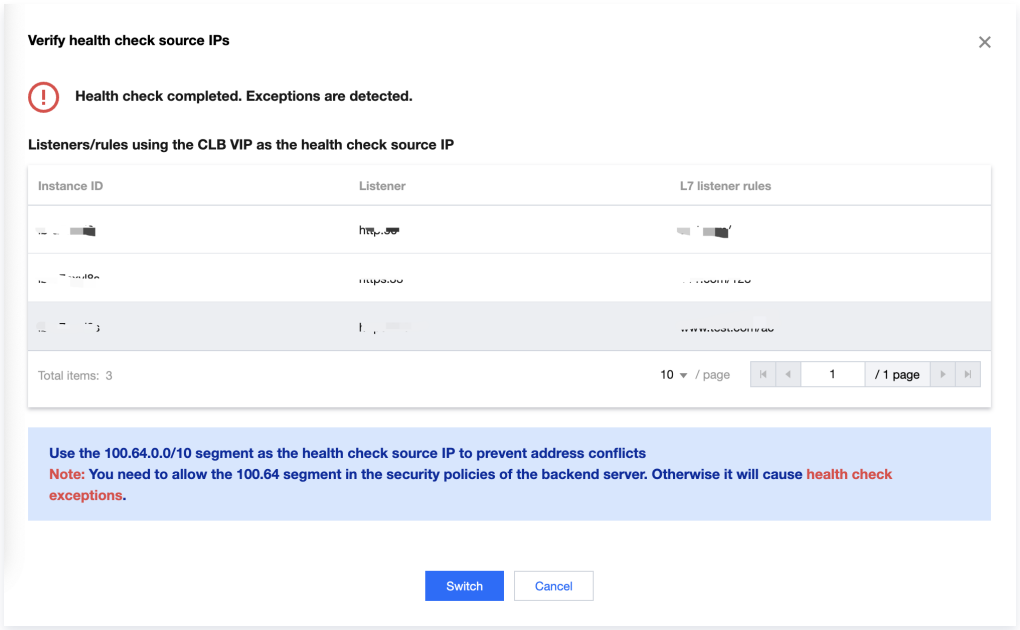
Instructions

1. Log in to the Cloud Load Balancer console.
2. In the left sidebar, select **Self-Help Assistant** > **Health Check Source IP Diagnosis**.
3. Select a region at the top of the page, and click **Start Diagnosis**.
4. On the **Health Check Source IP Diagnosis** page, select the target instance and click **Start Diagnosis**.
5. Upon completion of the diagnosis, the results will be displayed.
   - If the target instance does not use the CLB VIP as the health check source IP, it will be displayed as shown in the following image.



   - If the target instance uses the CLB VIP as the health check source IP, it will be displayed as shown in the image below. At this point, clicking **One-Click Switch** will quickly complete the process of changing the health check source IP from the CLB

VIP to the 100.64.0.0/10 IP range.



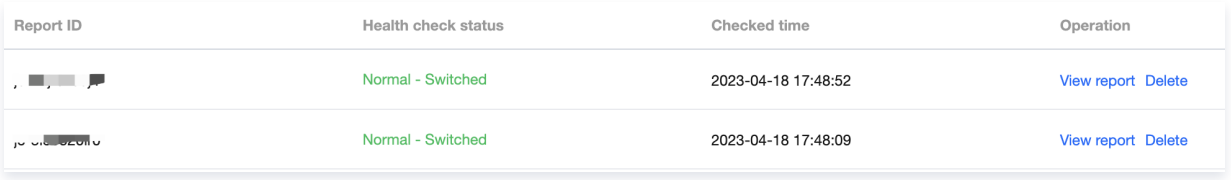
Other operations

Diagnosis Report Status Description

Status information	Description
Normal – No switching required	In this diagnosis, none of the CLB instances have their health check source IP set as the virtual service address (VIP) of the instance.
Normal – Switch completed	In this diagnosis, some CLB instances have their health check source IP set as the instance's virtual service address (VIP), and the operation to switch the health check source IP to the 100.64.0.0/10 IP range has been completed.
Alert – Not switched	In this diagnosis, some CLB instances have their health check source IP set as the instance's virtual service address (VIP) and have not yet switched. It is recommended to take action promptly.

View Diagnosis Report

1. Log in to the Cloud Load Balancer console.
2. In the left sidebar, select **Self-Help Assistant** > **Health Check Source IP Diagnosis**.
3. After selecting a region at the top of the page, you can view the diagnostic report information for the corresponding region.
4. Click **View Report** in the **Action** column to view detailed information in the historical diagnosis report.



Deleting Diagnostic Report

1. Log in to the Cloud Load Balancer console.
2. In the left sidebar, select **Self-Help Assistant** > **Health Check Source IP Diagnosis**.
3. After selecting a region at the top of the page, you can view the diagnostic report information for the corresponding region.
4. Click **Delete** in the **Operation** column, and in the pop-up confirmation dialog box, click OK to delete the historical diagnosis report.

Documentation

- Configure Health Check
- Health Check Probe Identifier
- Changing Health Check Source IP

Certificate Management

Managing Certificates

Last updated: 2023-09-05 16:30:42

When configuring an HTTPS listener for a Cloud Load Balancer, you can directly use a certificate from the SSL Certificate Service or upload the required third-party server certificate and SSL certificate to the Cloud Load Balancer.

Certificate Requirements

Cloud Load Balancer supports only certificates in PEM format. Before uploading a certificate, ensure that your certificate, certificate chain, and private key meet the format requirements. For information about the certificate requirements, see Certificate Requirements and Certificate Format Conversion.

Certificate Encryption Algorithms

Cloud Load Balancer 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.

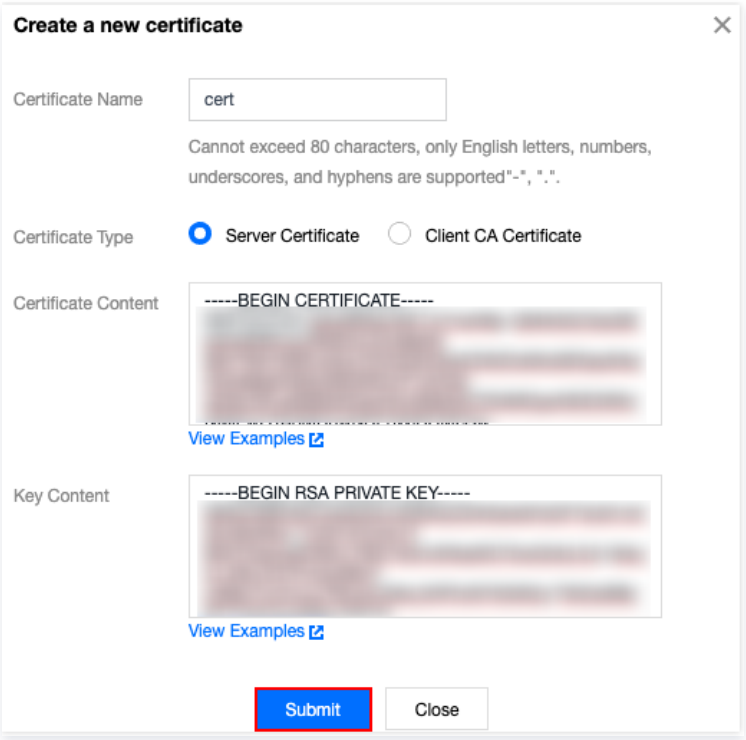
Updating Certificates

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.

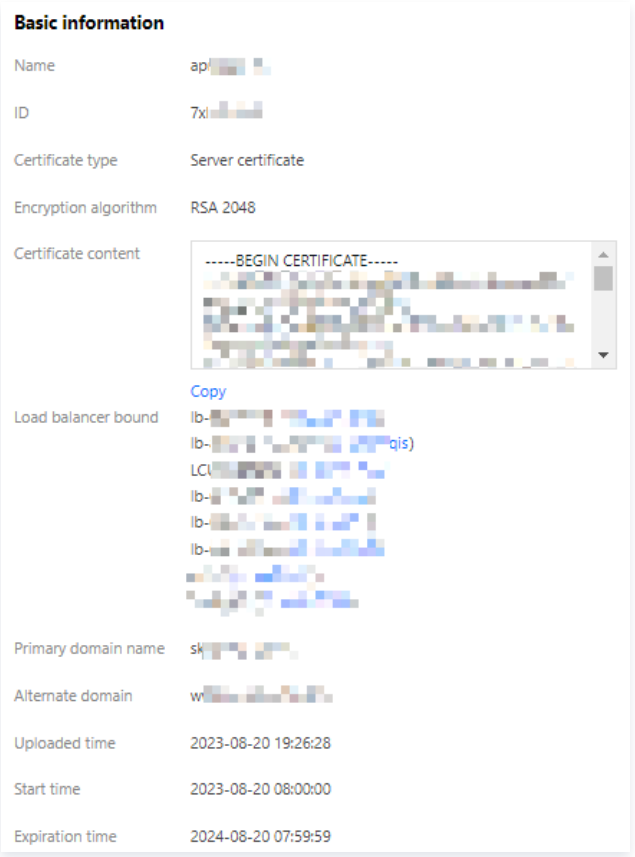
1. Log in to the Cloud Load Balancer console.
2. Click **Certificate Management** in the left sidebar.
3. In the **Certificate Management** page's certificate list, click **Update** in the **Action** column to the right of the target certificate.

4. In the pop-up window, enter the content and key of the new certificate and click **Submit**.



Viewing CLB Instances Associated with a Certificate

1. Log in to the Cloud Load Balancer console.
2. Click **Certificate Management** in the left sidebar.
3. In the **Certificate Management** page's certificate list, click the target certificate ID.
4. On the **Basic Information** page, view the Cloud Load Balancer instances associated with the certificate.



Apply for certificate

Last updated: 2023-09-05 16:30:49

Registering an account

To apply for a certificate on Tencent Cloud, first you need to register a Tencent Cloud account and complete identity verification.

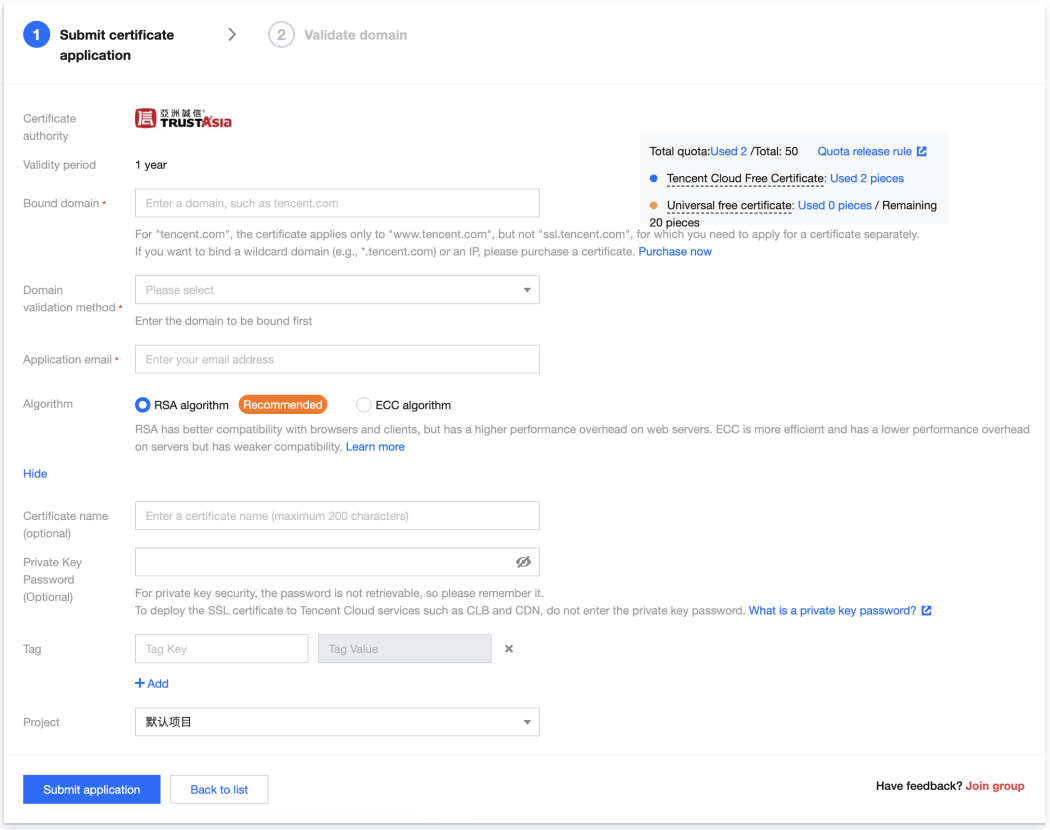
1. New users, please click Tencent Cloud's official website and select **Sign Up for Free** in the top right corner to access the registration page.
2. Please register a Tencent Cloud account to access the Tencent Cloud Console.
3. Complete identity verification before proceeding with the certificate application.

Applying for a Free Certificate

Note

- Free certificates are only available for second-level domain names and their subdomains, and do not support IP or wildcard domain applications. For example, `dnspod.cn` and `docs.dnspod.cn`.
- Within the scope of TrustAsia (not necessarily applied through Tencent Cloud), a maximum of 20 free certificates can be applied for the same primary domain. When applying, please check whether the domain has TrustAsia certificates on other service provider platforms to avoid reaching the application limit. For more details, please refer to Free Certificate Quota Related Questions.
- If you wish to continue using a free certificate after its expiration, please reapply and install it.

1. Log in to the Tencent Cloud Console, navigate to the "My Certificates" management page, and click on **Apply for a Free Certificate**.
2. Fill out the certificate application form as shown below:



- **Binding the certificate to a domain**: Enter a single domain, such as `tencent.com` or `ssl.tencent.com`.
- **Domain Validation Methods:**

  Note

  - Automatic DNS Verification: For the verification method, please refer to Automatic DNS Addition.

    If the domain name applied for is successfully hosted on the DNSPod DNS Console, automatic DNS addition is supported.
  - Manual DNS verification: For the verification method, please refer to DNS Validation.

- File Verification: For the verification method, see File Verification.

- **Email Address**: Please enter your email address.
- **Algorithm Selection**: Choose the desired encryption algorithm for your certificate. For more information about the algorithms, refer to What are the differences between RSA and ECC encryption algorithms?
- **Certificate Name**: Optional, please enter a remark for the certificate, not exceeding 200 characters.
- **Private Key Password**: Optional. To ensure the security of your private key, **password recovery is NOT supported**, so please remember the password.

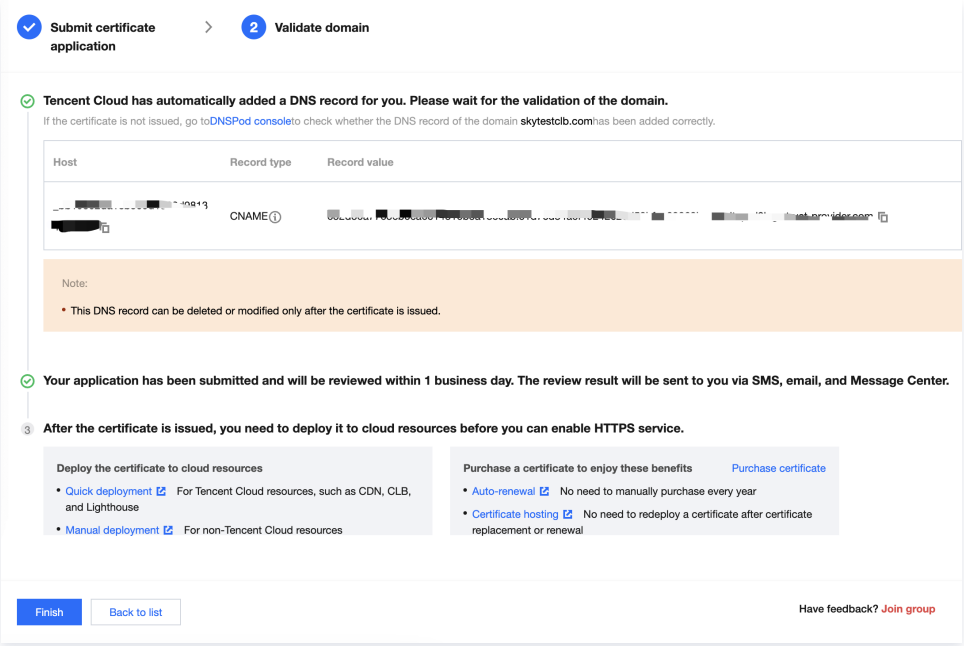
> **Note**
> If you need to deploy Tencent Cloud services such as Cloud Load Balance or CDN, do not enter the private key password.

- **Tag**: Select your tag key and tag value to better manage existing Tencent Cloud resources by category.

> **Note**
> To add tags, please refer to Managing Tags.

- **Project**: Please select the project to which your certificate belongs, making it convenient for you to manage your certificates through the project.

3. Follow the **verification instructions** to complete domain identity validation, and click **Finish**. As shown in the image below:



4. Once the domain verification is approved, the CA will issue the certificate within 24 hours. Please be patient.

> **Note**
> The submitted domain failed the security review by the certificate authority. For specific reasons, please refer to Reasons for Security Review Failure.

Download and Deployment

After completing the domain verification, you can click **Download** to obtain the issued certificate for local installation and deployment, or deploy it to relevant Tencent Cloud services. For related operations, please refer to How to choose the SSL certificate installation and deployment type?

FAQs

- Free SSL Certificate Quota Related Questions

- Can the TXT records for domain name resolution configured in the SSL certificate be deleted?
- What should I do if I forgot my private key password?
- What to do if the free SSL certificate remains unverified?

Certificate Requirements and Certificate Format Conversion

Last updated: 2023-09-05 16:30:58

This document introduces the requirements on SSL certificates and describes how to convert certificate formats.

Certificate Application Process

1. Use the OpenSSL tool to generate a private key file locally, where `privateKey.pem` is your private key file. Please keep it safe and secure.

```
openssl genrsa -out privateKey.pem 2048
```

2. Use the OpenSSL tool to generate a certificate request file, where `server.csr` is your certificate request file, which can be used to apply for a certificate.

```
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

- Users need to apply for a PEM format certificate in a Linux environment. Cloud Load Balance does not support certificates in other formats. For information on converting other formats to PEM, please refer to the Instructions for Converting Certificates to PEM Format section below.
- If the certificate is issued by a root CA, you will receive a unique certificate that does not require additional certificates. The configured site will be considered trustworthy by browsers and other access devices.
- If the certificate is issued by an intermediate CA, the certificate file you receive will contain multiple certificates. You need to manually combine the server certificate and the intermediate certificate before uploading.
- When your certificate has a certificate chain, please convert the certificate chain content into PEM format and merge it with the certificate content for uploading.
- The concatenation rule is: place the server certificate first, followed by 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.

Example of Certificate Format and Certificate Chain Format
Below are examples of certificate format and certificate chain format. Please ensure the formats are correct before uploading:

1. Root CA-issued certificate: The certificate format is PEM format for Linux environments. Example as follows:

   Certificate rules are:

   - Upload the content starting with [-----BEGIN CERTIFICATE-----] and ending with [-----END CERTIFICATE-----].
   - Each line contains 64 characters, with the last line not exceeding 64 characters.

2. Certificate chain from an intermediate CA:

   -----BEGIN CERTIFICATE-----
   -----END CERTIFICATE-----
   -----BEGIN CERTIFICATE-----
   -----END CERTIFICATE-----
   -----BEGIN CERTIFICATE-----
   -----END CERTIFICATE-----

   Certificate chain rules:

   - There must be no blank lines between certificates.
   - Each certificate must adhere to the Certificate Format Requirements mentioned earlier.

RSA Private Key Format Requirements

An example is as follows:

RSA private keys can include all private keys (RSA and DSA), public keys (RSA and DSA), and (x509) certificates. They store data in Base64-encoded DER format, surrounded by ASCII headers, making them suitable for text-mode transmission between systems.
RSA private key rules:

- Please upload the content starting with [-----BEGIN RSA PRIVATE KEY-----] and ending with [-----END RSA PRIVATE KEY-----].
- Each line contains 64 characters, and the last line may have fewer 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 the `new_server_key.pem` content together with the certificate.

Converting Certificates to PEM Format

Currently, Cloud Load Balance supports only PEM format certificates. Certificates in other formats need to be converted to PEM format before they can be uploaded to Cloud Load Balance. It is recommended to use the OpenSSL tool for conversion. Below are some popular methods for converting certificate formats to PEM format.

Converting DER 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
```

Converting P7B to 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.

Conversion from PFX to PEM

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
`
```

Converting CER/CRT to PEM

For CER/CRT format certificates, you can convert them by directly changing the certificate file extension. For example, rename the "servertest.crt" certificate file to "servertest.pem".

SSL One-way Authentication and Mutual Authentication

Last updated: 2023-09-05 16:32:08

SSL (Secure Sockets Layer) is a security protocol designed to ensure the safety and data integrity of network communications. This article primarily discusses SSL one-way authentication and two-way authentication.

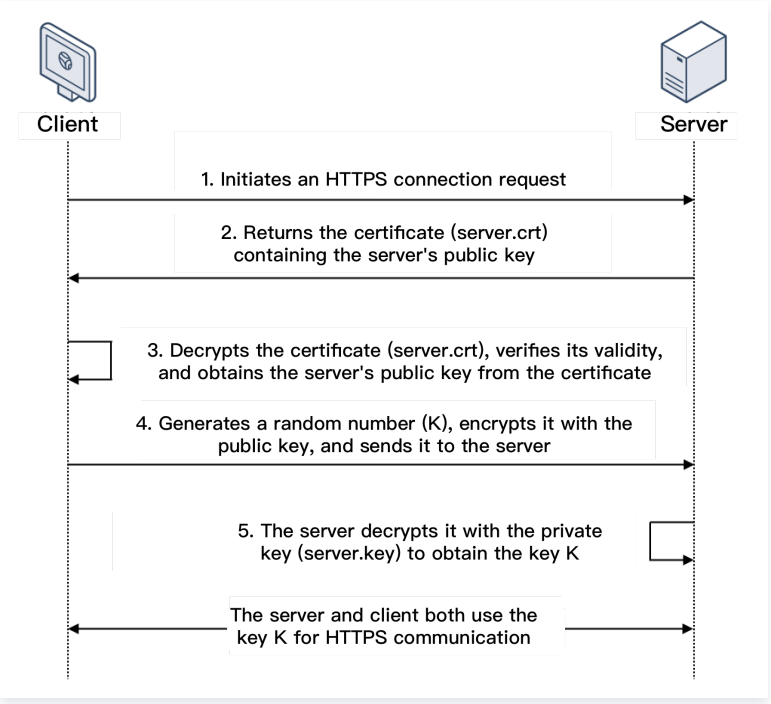
> **Note**
> With Cloud Load Balance (CLB), you can choose between one-way authentication and two-way authentication for SSL decryption when creating a TCP SSL listener or an HTTPS listener. For more information, please refer to Configuring a TCP SSL Listener and Configuring an HTTPS Listener.

Differences Between SSL One-way Authentication and Mutual Authentication

- SSL One-way Authentication requires only the server to have a certificate, while SSL Mutual Authentication necessitates both the client and server to possess certificates.
- In the SSL one-way authentication process, compared to SSL mutual authentication, there is no need for the server to verify the client's certificate or negotiate an encryption scheme. The server sends an unencrypted password scheme to the client (which does not affect the security of the SSL authentication process).
- For most web applications with a large user base, there is no need for user identity verification at the communication layer, so SSL one-way authentication is sufficient. However, some financial industry users may require client identity verification for their application integration, in which case SSL mutual authentication is necessary.

SSL One-way Authentication

SSL one-way authentication only requires the verification of the server's identity, without the need to verify the client's identity. The process of SSL one-way authentication is illustrated in the following diagram.



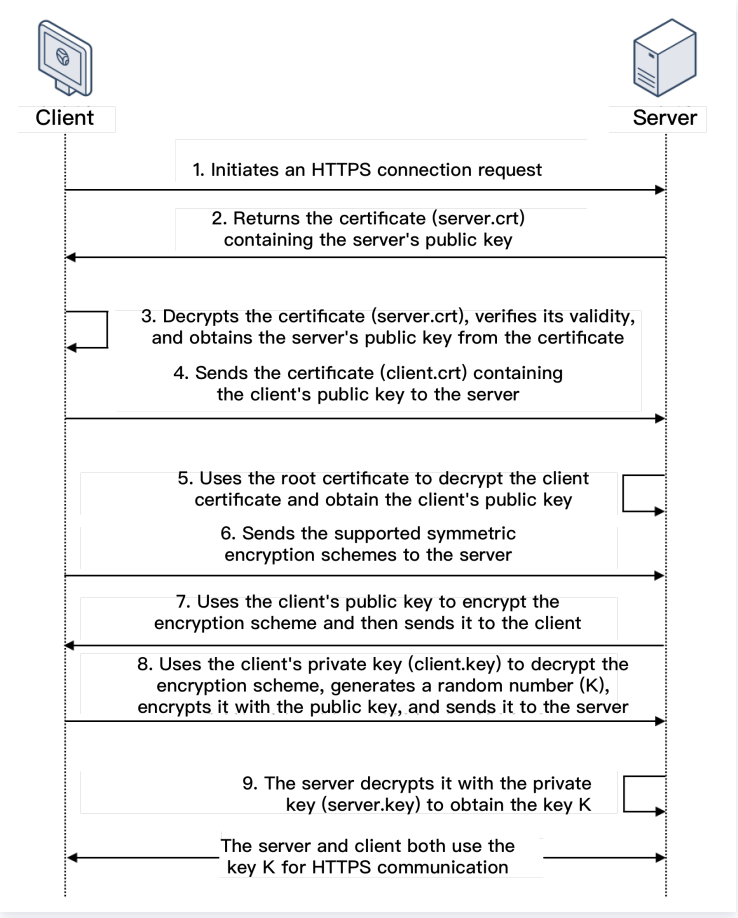
1. A client initiates an HTTPS connection request to the server together with the supported SSL protocol versions, encryption algorithms, generated random numbers, and other information.
2. The server returns an SSL protocol version, encryption algorithm, generated random number, server certificate (server.crt), 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.

- Verify if the certificate has expired.
- Verify if the certificate has been revoked.
- Verify if the certificate is trustworthy.
- Verify that the domain name in the received certificate matches the requested domain name.

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 symmetric encryption key (the key K) will be used by the server and client for communication to guarantee information security.

SSL Mutual Authentication

SSL Mutual Authentication requires verification of both client and server identities. The process of SSL Mutual Authentication is illustrated in the following diagram.



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.
   - Verify if the certificate has expired.
   - Verify if the certificate has been revoked.
   - Verify if the certificate is trustworthy.
   - Verify that the domain name in the received certificate matches the requested domain name.
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.

Documentation

Certificate Requirements and Certificate Format Conversion

Log Management
Access Log Overview

Last updated: 2023-09-05 16:33:31

Cloud Load Balancer's access logs capture detailed information for each client request, including request time, request path, client IP and port, response code, and response time. Access logs can help you better understand client requests, troubleshoot issues, and analyze user behaviors.

Note:
- Only Layer-7 CLB supports configuring access logs.
- Access log configuration is currently supported in select regions only. For details, please refer to the Available Regions of CLS.

Storage method

Cloud Load Balancer's access logs support Cloud Log Service (CLS): CLS is a one-stop log service platform that provides a range of services, including log collection, storage, retrieval and analysis, real-time consumption, and log delivery. It helps users address various issues related to business operations, security monitoring, log auditing, and log analysis through logs.

Item	Storing Access Logs in CLS
Time granularity for log obtainment	Minute-level.
Online search	Supported.
Search syntax	Full-text search, key-value search, and fuzzy keyword search are available. For more information, please refer to Search Rules.
Supported regions	For more information on supported regions, please refer to the Available Regions of CLS.
Supported Type	Public and private network Cloud Load Balancer instances are supported.
Upstream and downstream links	CLS logs can be shipped to COS, and exported to CKafka for further processing.
Log retention	Tencent Cloud stores no access log by default. You need to store the access log in CLS as needed.

Related Actions

Storing Access Logs in CLS

Viewing Operation Logs

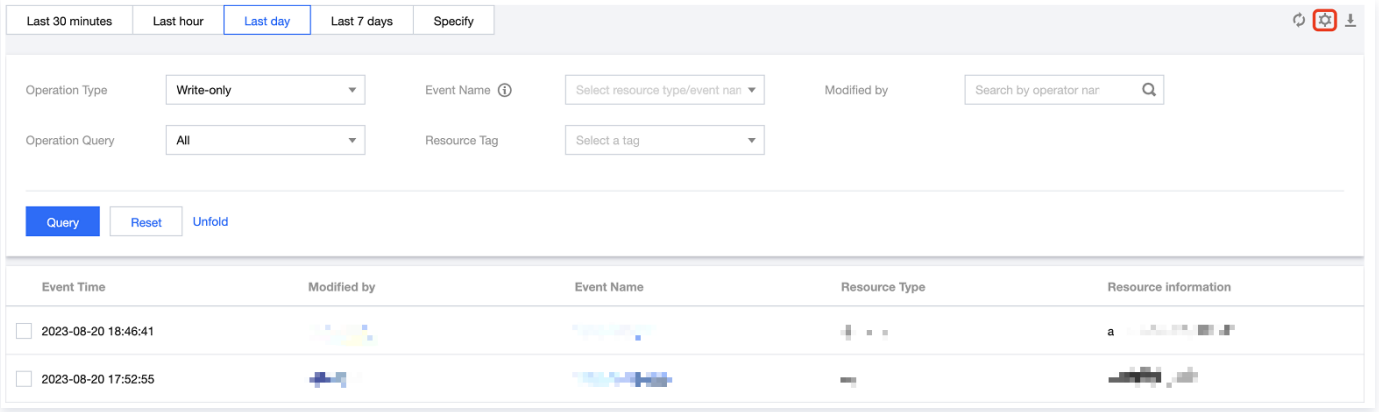
Last updated: 2023-09-05 16:37:09

You can query and download Cloud Load Balance operation records in the CloudAudit console.
CloudAudit is a service that supports supervision, compliance checks, operation reviews, and risk assessments for your Tencent Cloud account. CloudAudit provides a history of events for activities in your Tencent Cloud account, including operations performed through the Tencent Cloud Management Console, API services, command-line tools, and other Tencent Cloud services. This event history simplifies security analysis, resource change tracking, and troubleshooting tasks.

Instructions

View Operation Records

1. Log in to the CloudAudit Console.
2. Click **Operation Records** in the left sidebar to enter the "Operation Records" page. Alternatively, you can log in to the Cloud Load Balance Console and select **CloudAudit** in the upper right corner of the page for quick access to the operation records page.
3. On the Operation Records page, you can search for operation records based on username, resource type, resource name, event source, and event ID. By default, only partial data is displayed. You can click on the settings on the right side of the page to access more list fields.



The filter conditions are described as follows:

- **Time Range**: You can filter logs within a 30-day range in the last 90 days.
- **Operation Type**: Supports filtering by All, Read, and Write.
- **Event Name**: You can search and filter logs by using the interface names found in each product's API documentation. For example, CVM - RunInstances (create instance). Up to 10 events can be queried at a time.

  > **Note**
  > If you cannot find the event name of the required product in the list, please provide feedback by submitting a ticket through online customer service, and we will investigate and resolve the issue as soon as possible.

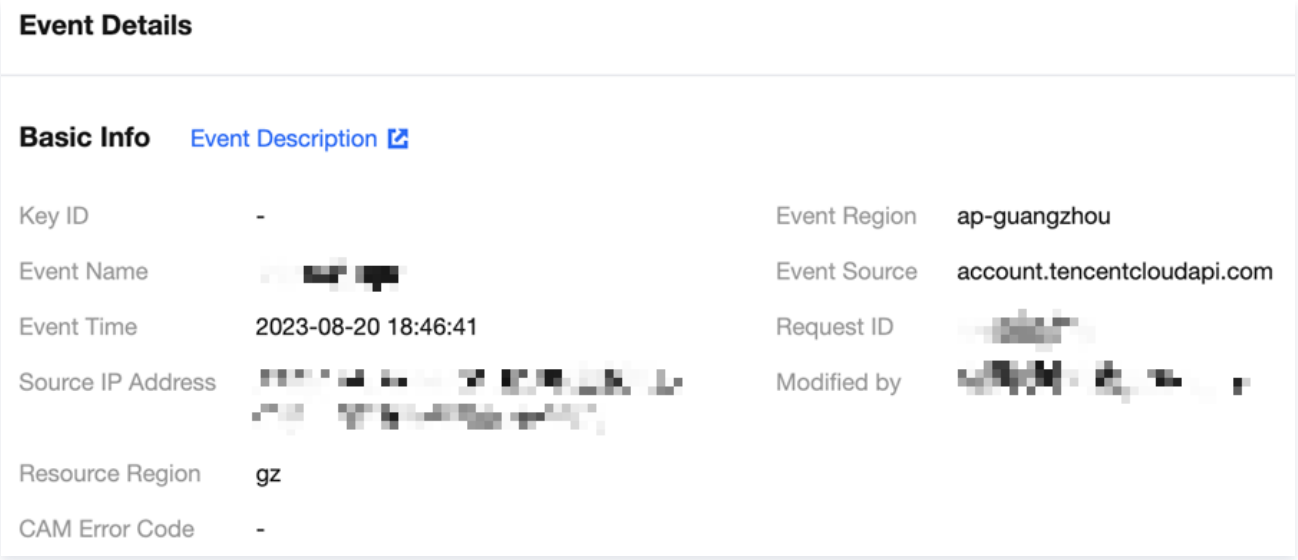
- **Operator**: Operators can be divided into the following types:
  - **Root Account Operation**: The username is displayed as root.
  - **Operation by a sub-user**: The sub-user name is displayed as the operator. If the sub-user has been deleted, the sub-user ID will be displayed instead.
  - **Role Operation**: The role name is displayed as the operator. If the role has been deleted, the role ID will be displayed instead.
    You can click on the operator to go to the "User List" page to view more information about the user.
- **Sensitive Operation Filtering**: Supports filtering of all sensitive and non-sensitive operations. Sensitive operations are events that may involve critical operations on cloud resources, as defined by the platform. If you need to include certain operations as sensitive operations, please submit a ticket through Online Customer Service, and we will handle it promptly.
- **Resource Tag**: You can filter logs by resource tag. For more information on tags, see Tag Overview.
- **Resource ID**: You can search by entering a resource ID, such as `ins-fi8oxxxx`.

- **Key ID**: Supports searching by entering the Key ID. For example, `AKIDZ0GSXSG2nT5c6Xxxxxxxxxxxxxxxx`.
- **Request ID**: Supports searching by entering a request ID. For example, `a7da0568-7580-4798-88c8-xxxxxxxxxx`.
- **API Error Code**: You can enter an API error code as listed in the corresponding API documentation for search.

4. Click **Query** to get the filtered operation records.

Viewing event details

1. If you need to view the details of a specific event, click on the **Event Name** in the list. In the expanded module, you can view the **Event Basic Information, Related Resources, and Event Records**.



> **Note**
> You can determine whether an event was successfully executed by checking the "CAM Error Code" field. If the CAM Error Code is empty, the event was executed successfully. If the CAM Error Code is not empty, the event execution failed. For specific error reasons, please view the `errorCode` and `errorMessage` fields in the event details.

2. You can view the event details in the **Event Records** module. For more information on field descriptions, see Appendix.

Configuring Access Logs

Last updated: 2023-09-05 17:07:13

Cloud Load Balancer supports configuring layer-7 (HTTP/HTTPS) access logs, which can assist in understanding client requests, troubleshooting issues, and analyzing user behavior. Currently, access logs can be stored in CLS, reported at a minute granularity, and searched online using multiple rules.
Access logs of Cloud Load Balancer are primarily used for troubleshooting and assisting businesses in quickly identifying issues. The access logging feature includes log reporting, log storage, and querying:

- 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:

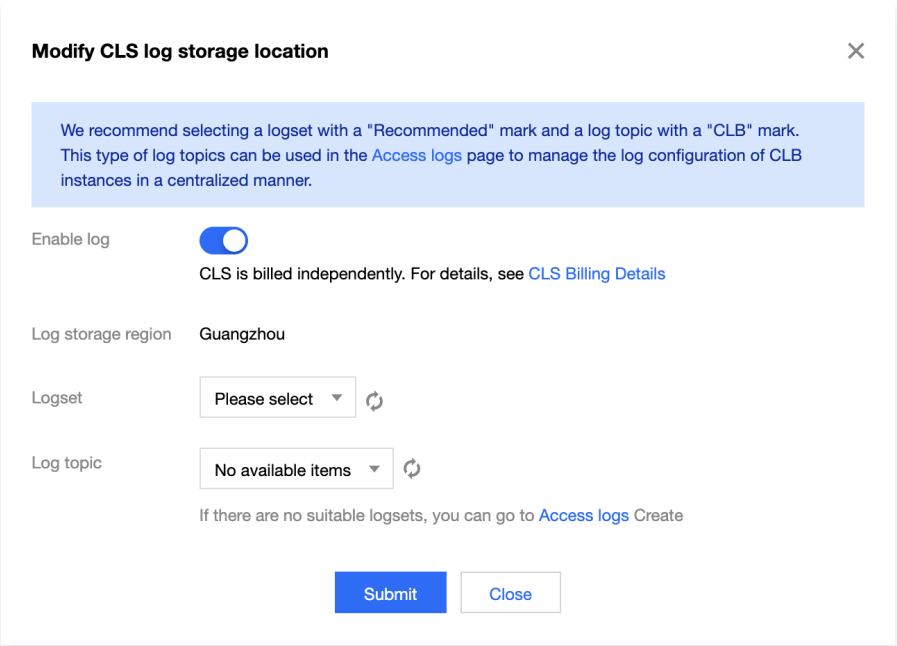
- Currently, access logs can be stored in CLS only for layer-7 protocols (HTTP/HTTPS) but not layer-4 protocols (TCP/UDP/TCP SSL).
- Storing CLB access logs to CLS is now free of charge. You only need to pay for the CLS service.
- Currently, this feature is supported only in certain regions as displayed in the console.

Method 1: Single-Instance Access Logging

Step 1. Enable access log storage in CLS

1. Log in to the Cloud Load Balancer console and click **Instance Management** in the left sidebar.
2. On the **Instance Management** page, click the ID of the target Cloud Load Balancer.
3. On the **Basic Information** page, click on the pencil icon in the "Access Log (Layer 7)" module.

4. In the **Modify CLS Log Storage Location** dialog box, enable **Log**, select the logset and log topic for storing access logs, and click **Submit**. If you have not created a logset or log topic, please Create Related Resources first, and then select the specific storage location.



Note:
We recommend selecting 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:
- Log topics marked with **CLB** automatically create indexes by default; for regular log topics, indexes must be created manually, otherwise, searching is not supported.
- Log topics marked with **CLB** support dashboards by default; for regular log topics, dashboards must be configured manually.

5. Click the logset or log topic to redirect to the log search page in CLS.
6. (Optional) If you wish to disable the access logs, you can click on the pencil icon again, and in the pop-up **Modify CLS Log Storage Location** dialog box, you can disable and submit it.

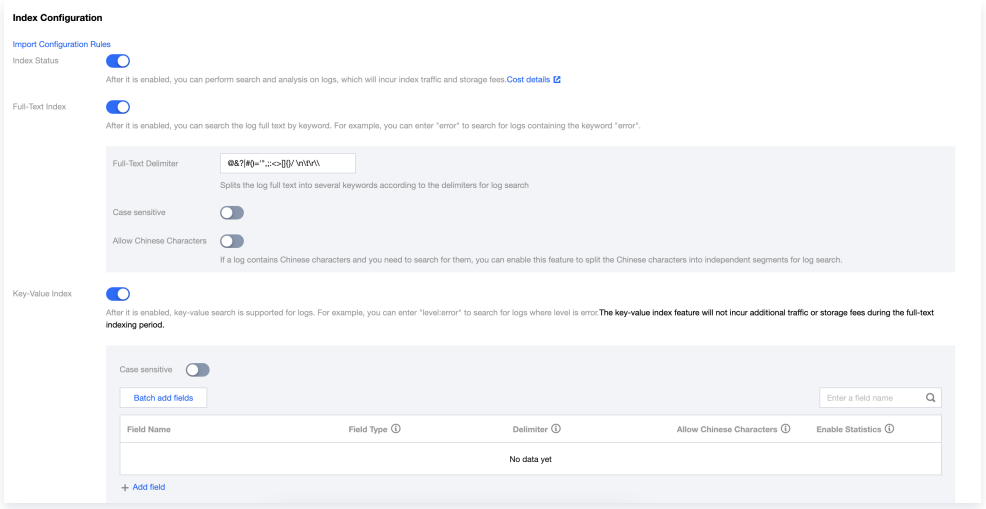
Step 2. Configure log topic indexes

Note:
If the access log is configured for a single instance, you must configure the index for the log topic; otherwise, no logs can be found.

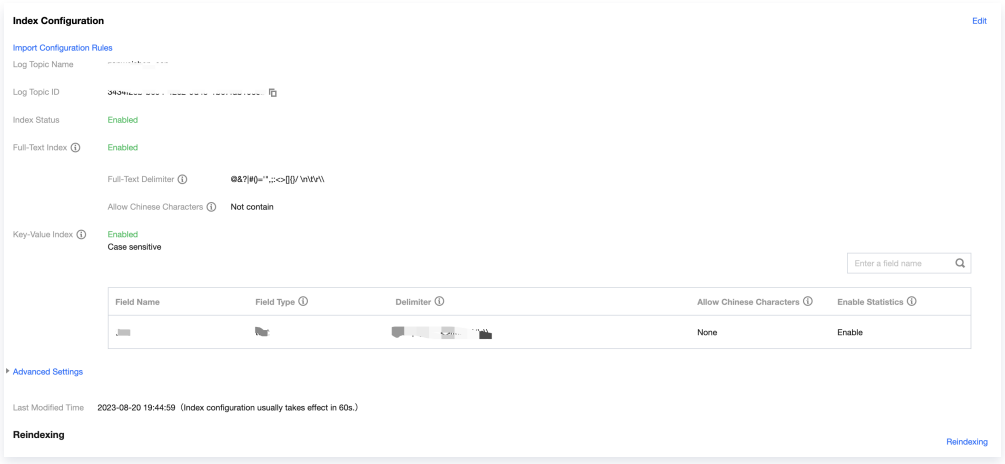
The recommended indexes are as follows:

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 Cloud Log Service console and click **Log Topics** 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 upper right corner to add an index. After adding, click **Confirm** at the bottom of the page. For information on index field configuration, see Enable Index.

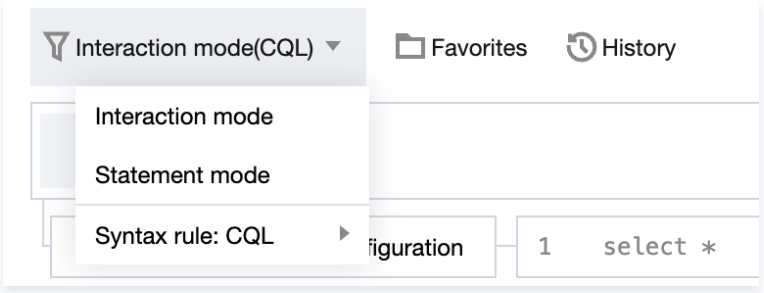


4. The result of index configuration is as shown below:

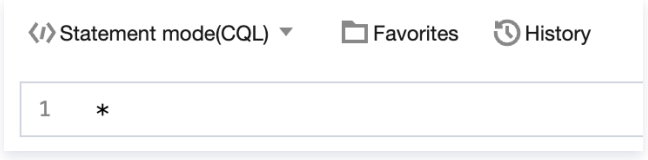


Step 3. View access logs

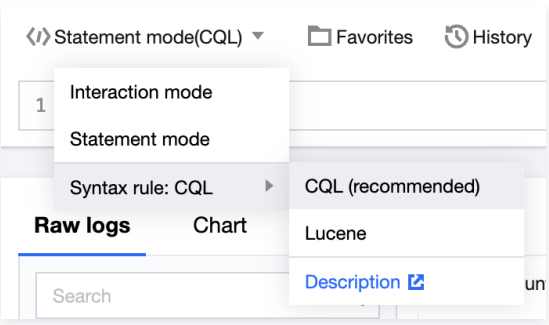
1. Log in to the Cloud Log Service console.
2. In the left sidebar, click **Search and Analysis**.
3. Select the desired log topic at the top.
4. Choose the search syntax input mode. Cloud Log Service offers two modes for entering search syntax. For more information, see Syntax Rules.
   4.1 Interactive mode: Generates search and analysis statements by specifying search conditions and statistical analysis rules through mouse clicks, offering high usability.



   4.2 Statement mode: Directly input search and analysis statements, which must comply with syntax rules, offering high flexibility.



5. To switch between syntax rules, you can do so here. It is recommended to use CQL.



6. After entering the search analysis statement, select the **Time Range** on the right, and then click the **Search** button to execute the search analysis.
   6.1 When the search and analysis statement only contains search conditions: you can view logs that match the search conditions in **Raw Logs**, sorted by log time in descending order by default.
   6.2 When the search and analysis statement includes an SQL statement, you can view the analysis results in the **Statistical Charts** and also view the logs that meet the search criteria in the **Raw Logs** for easy comparison between the statistical results and the original logs.

Method 2: Batch configure access logging

Step 1: Create a logset and log topic.

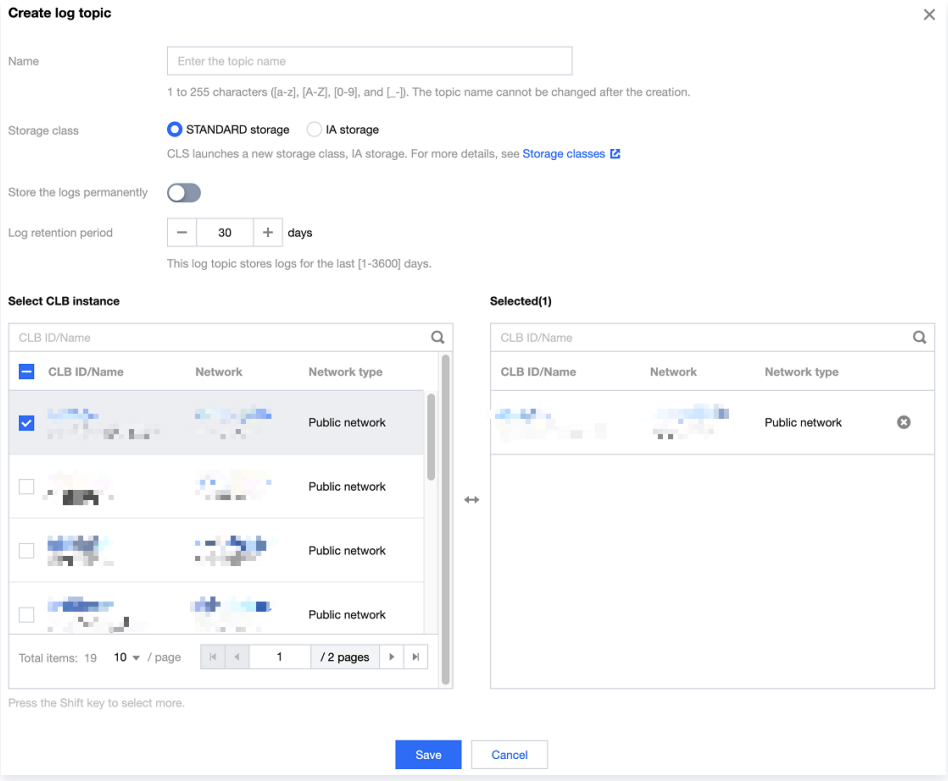
To store access logs in Cloud Log Service (CLS), you need to first create a logset and log topic.
If you have created a logset and log topic, skip to Step 2 to proceed.

1. Log in to the Cloud Load Balancer console and click **Access Logs** > **Log List** 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 only create 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 **Add Log Topic** window, after selecting the storage type and log retention period, choose the Cloud Load Balancer instances from the left side and add them to the right-side list. Then, click **Save**.

> **Note:**
> - Storage classes are divided into STANDARD storage and IA storage. For more information, see Storage Class Overview.
> - Logs can be retained permanently or for a specified time period.
> - When creating a log topic, you can choose to add or not to add a Cloud Load Balancer instance. In the **Operation** column on the right side of the log topic list, click **Manage** to add a Cloud Load Balancer instance again. Each Cloud Load Balancer instance can only be added to 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, in the **Operation** column on the right side of the log topic list, click **Stop** to cease log delivery.

Step 2. View access logs

1. Log in to the Cloud Log Service console.
2. In the left sidebar, click **Search and Analysis**.
3. Select the desired log topic at the top.
4. Choose the search syntax input mode. Cloud Log Service offers two modes for entering search syntax. For more information, see Syntax Rules.
   4.1 Interactive mode: Generates search and analysis statements by specifying search conditions and statistical analysis rules through mouse clicks, offering high usability.
   4.2 Statement mode: Directly input search and analysis statements, which must comply with syntax rules, offering high flexibility.
5. To switch between syntax rules, you can do so here. It is recommended to use CQL.
6. After entering the search analysis statement, select the **Time Range** on the right, and then click the **Search** button to execute the search analysis.
   6.1 When the search and analysis statement only contains search conditions: you can view logs that match the search conditions in **Raw Logs**, sorted by log time in descending order by default.
   6.2 When the search and analysis statement includes an SQL statement, you can view the analysis results in the **Statistical Charts** and also view the logs that meet the search criteria in the **Raw Logs** for easy comparison between the statistical results and the original logs.

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	Note	Field type
stgw_request_id	Request ID	text
time_local	Access time and time zone. Example: `01/Jul/2019:11:11:00 +0800`, where `+0800` represents UTC+8.	text
protocol_type	Protocol type. Supported protocols: HTTP, HTTPS, SPDY, HTTP2, WS, and WSS.	text
lb_id	CLB Instance ID: The unique identifier of a CLB instance.	text
server_addr	For CLB VIPs, only non-domain-based instances are supported, while domain-based instances will have an empty value.	text
server_port	CLB VPort, that is, the listening port.	long
server_name	`server_name` 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 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 **Host** field in the HTTP header.	text
http_user_agent	`user_agent` field in the HTTP header.	text
http_referer	Source of the HTTP request.	text
http_x_forward_for	Content of `x-forward-for 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: second	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: second	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	Time in microseconds taken by SSL handshake phases, in the format of x:x:x:x:x:x:x, with the time strings of different phases separated by colons (:). If the time of a phase is less than 1 ms, 0 is displayed. • The first field indicates whether the SSL session is reused. • The second field indicates the entire handshake time. • The third to seventh fields indicate the time taken by each SSL handshake phase. • The third field indicates the time from when CLB receives "client hello" to when it sends "server hello done". • The fourth field indicates the time from when CLB starts sending the server certificate to when it finishes sending the server certificate. • The fifth field indicates the time from when CLB calculates the signature to when it finishes sending "server key exchange". • The sixth field indicates the time from when CLB starts receiving "client key exchange" to when it finishes receiving "client key exchange". • The seventh field indicates the time from when the CLB instance receives `client key exchange` to when the CLB instance sends `server finished`.	text
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 `vip_vpcid` value of a public network CLB instance is `-1`.	long
request_method	Supported request methods: POST and GET.	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	Note	Field type
time_local	Access time and time zone. Example: `01/Jul/2019:11:11:00 +0800`, where `+0800` 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	`server_name` 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 **Host** 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: second	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: second	double

Sampling Logs

Last updated: 2023-09-05 17:15:25

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:**
> Cloud Load Balancer (CLB) supports configuring access logs and health check logs to be stored in Tencent Cloud Log Service (CLS), enabling log data search, analysis, visualization, and alerting services. Tencent Cloud Log Service (CLS) is billed separately. For more information on CLS billing, please refer to CLS Pricing Details.

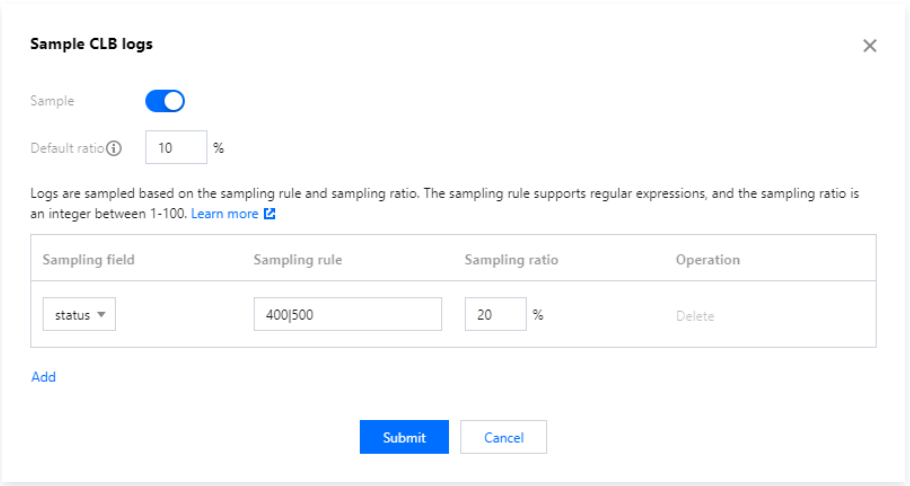
Preparations

- 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.

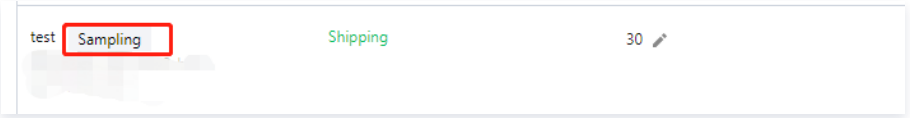
Sampling Layer-7 Access Logs

1. Log in to the Cloud Load Balancer 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.

Category	Note
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 **status** 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: 400|500.
Sampling ratio	Sampling ratio. You can enter an integer from 1 to 100.
Action	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.



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.



Sampling Health Check Logs

1. Log in to the Cloud Load Balancer 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.

Documentation

- Configuring Access Logs
- Configuring Health Check Logs

Configuring Health Check Logs

Last updated: 2023-09-05 17:12:32

If you wish to view health check logs, you must first store them in the Cloud Log Service (CLS) and then access them within CLS. Cloud Load Balancer (CLB) supports configuring health check logs to be stored in CLS, enabling minute-level granularity log reporting and multi-rule online querying, assisting you in troubleshooting health check anomalies and swiftly pinpointing issues.

> **Note:**
> The health check logging feature is currently in beta testing. To use it, please submit a beta application.

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.

Description

- 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.
- Currently, 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 Cloud Load Balancer console and click **Health Check Logs** in the left sidebar.
2. On the "Health Check Logs" page, click **Activate Now**, and in the pop-up dialog box, click **Authorize and Activate**.
3. Navigate to the Cloud Access Management console, and on the "Role Management" page, click **Grant**.

Step 2: Create a Logset and Log Topic

To store health check logs in Cloud Log Service (CLS), you need to first create a logset and log topic.
If you have already created a logset and log topic, proceed directly to Step 3 to begin.

1. Log in to the Cloud Load Balancer 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 **Add Log Topic** window, after selecting the storage type and log retention period, choose the Cloud Load Balancer instances from the left side and add them to the right-side list. Then, click **Save**.

   > **Note:**
   > - Storage classes are divided into STANDARD storage and IA storage. For more information, see Storage Class Overview.
   > - Logs can be retained permanently or for a specified time period.
   > - When creating a log topic, you can choose to add or not to add a Cloud Load Balancer instance. In the **Operation** column on the right side of the log topic list, click **Manage** to add a Cloud Load Balancer instance again. Each Cloud Load Balancer instance can only be added to one log topic.
   > - Each region supports the creation of a single logset, within which multiple log topics can be created. You can categorize different CLB logs into various log topics, which will be marked with "CLB" by default.

6. (Optional) To disable health check logging, click **Stop** in the **Operation** column on the right side of the log topic list to stop log delivery.

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 Cloud Load Balancer console and click **Health Check Logs** in the left sidebar.
2. In the upper left corner of the "Health Check Logs" page, select the relevant region. In the "Log Topic" area, click on **Search** in the right-hand "Actions" column to be redirected to the Log Service Console.
3. In the CLS console, click **Search Analysis** in the left sidebar.
4. On the **Search Analysis** page, input the search analysis statement into the input box, select a time range, and click **Search** to retrieve the 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	Note	Field type
protocol	Protocol type (HTTP/HTTPS/SPDY/HTTP2/WS/WSS).	text
rsport	Port of the real server.	long
rs_vpcid	VPC ID of the real server. The `vip_vpcid` value of a public network CLB instance is `-1`.	long
vport	CLB VPort, that is, the listening port.	long
vpcid	VPC ID of the VIP of the CLB instance. The `vip_vpcid` value of a public network CLB instance is `-1`.	long
time	Access time and time zone. Example: "01/Jul/2019:11:11:00 +0800", where "+0800" represents 8 hours ahead of UTC, which is Beijing time.	text
vip	VIP of the CLB instance.	text
rsip	IP address of the real server.	text
status	Health status. Valid values: • `true`: healthy • `false`: 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

Documentation

Getting Started with Cloud Log Service (CLS)

Accessing Log Dashboard

Last updated: 2023-09-05 17:21:38

Cloud Load Balancer offers an out-of-the-box access log dashboard. Once you configure access logs to be stored in Cloud Log Service (CLS), Cloud Load Balancer will automatically set up a dashboard for analyzing access logs in a graphical format, enabling comprehensive observation, analysis, and troubleshooting capabilities within the Cloud Load Balancer console.

Dashboard

Each log topic has its own dashboard, which contains data of following metrics.

- PV
- UV
- Requested Traffic (KB)
- Response Traffic to Client (KB)
- Average request time
- Average response time
- Distribution of Status Codes Returned by Backend Services
- Overall status codes
- PV/UV Trend
- Request/Response Traffic Trend
- Average Request/Response Time per Minute
- P99, P95, P90, P50 Access Time
- Top Instances by Request Count
- Top Domains by Request Count

Preparations

You have successfully created the log topic. For the procedure, see Creating Logsets and Log Topics.

Instructions

1. Log in to the Cloud Load Balancer console and choose **Access Logs** > **Dashboard** in the left sidebar.
2. On the **Access Log Dashboard** page, select the region and log topic, and the corresponding dashboard for the selected log topic will be displayed automatically.
3. (Optional) In the top-left corner of the **Access Log Dashboard** page, you can set filters for Cloud Load Balancer VIP, client IP, backend server IP, and response code to filter and display access logs.

Documentation

- Configuring Access Logs
- IP Function

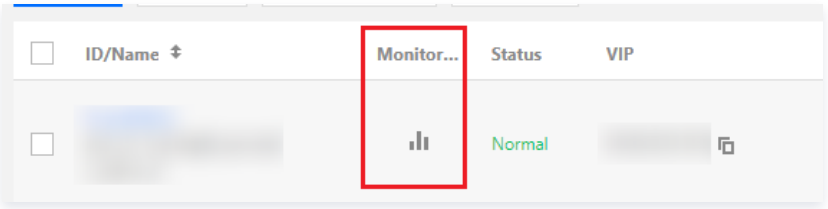
Monitoring and Alarming
Obtaining Monitoring Data

Last updated: 2023-09-05 17:25:25

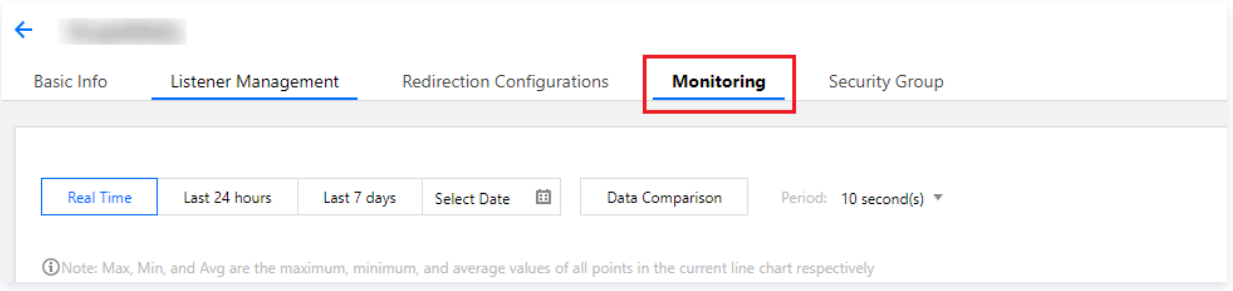
Basic Cloud Monitor provides data collection and display features for Cloud Load Balancer and backend instances. With Basic Cloud Monitor, you can view statistical data for Cloud Load Balancer, verify if the system is operating normally, and create corresponding alerts. For more information about Basic Cloud Monitor, please refer to the Basic Cloud Monitor product documentation.
Tencent Cloud provides the Basic Cloud Monitor service for all users by default; therefore, you do not need to manually activate it. As long as you use Cloud Load Balancer, Basic Cloud Monitor will help you collect relevant monitoring data. You can view the monitoring data of Cloud Load Balancer through the following methods:

CLB Console Method

1. Log in to the Cloud Load Balancer Console and click the monitoring icon next to the Cloud Load Balancer instance ID. You can then quickly browse the performance data of each instance through the monitoring floating window.



2. Click the Cloud Load Balancer instance ID to enter the Cloud Load Balancer details page. Click the **Monitoring** tab to view the monitoring data for the current Cloud Load Balancer instance.



Basic Cloud Monitor Console Method

Log in to the Basic Cloud Monitor Console, click **Cloud Load Balancer-CLB** under the "Cloud Product Monitoring" module in the left navigation bar. Click the Cloud Load Balancer instance ID to enter the monitoring details page, where you can view the monitoring data for that Cloud Load Balancer instance. Expand the instance to view the listener, backend server, and other monitoring information.

API Method

You can use the GetMonitorData API to obtain monitoring data for all products. For more information, please refer to Pull Metric Monitoring Data. For Cloud Load Balancer's namespace, please refer to Public Cloud Load Balancer Monitoring Metrics and Private Cloud Load Balancer Monitoring Metrics.

Descriptions of monitoring metrics

Last updated: 2023-09-05 17:35:19

Basic Cloud Monitor collects raw data from running Cloud Load Balancer instances and displays the data in an easily readable graphical format. By default, statistical data is stored for one month, allowing you to observe the performance of instances over a month and gain a better understanding of your application services' operational status.
We recommend that you view the load balancer monitoring through the Tencent Cloud Observability Platform Console. Select **Cloud Product Monitoring** > **Load Balancer-CLB**, click on the Load Balancer instance ID to enter the monitoring details page. Here, you can view the monitoring data for this load balancer instance. Expanding the instance will reveal monitoring information for listeners, backend servers, and more.

> **Note:**
> Currently, only the concurrent connection utilization and new connection utilization metrics of Performance Capacity Cloud Load Balancer instances report data once enabled, while Shared Cloud Load Balancer instances do not report data for the time being.

CLB Instance Level

Metric	Meaning	MetricsDescription	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.	Connections	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.	Connections	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.	Connections	10、60、300
ConcurConnVipRatio	Concurrent connection quota utilization (%)	Within the statistical period, the utilization rate of concurrent connections from clients to a Cloud Load Balancer compared to the performance upper limit of concurrent connections for Performance Capacity specifications. This metric is supported only by Performance Capacity instances and not by Shared instances.	%	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.	keys/second	10、60、300
NewConnVipRatio	New Connection Utilization	At a certain time point within the statistical period, the utilization rate of new connections from clients to the Cloud Load Balancer compared to the performance upper limit of new connections for Performance Capacity specifications. This metric is supported only by Performance Capacity instances and not by Shared instances.	%	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.	keys/second	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.	keys/second	10、60、300
ClientAccIntraffic	Client-to-CLB traffic in	Bandwidth used by 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
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	Discarded 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 Checking Account Type.	Connections	10、60、300
InDropBits	Discarded inbound bandwidth	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 type, see Checking Account Type.	Byte	10、60、300
OutDropBits	Discarded outbound bandwidth	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 Checking Account Type.	Byte	10、60、300
InDropPkts	Discarded inbound data packets	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 Checking Account Type.	keys/second	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 Checking Account Type.	keys/second	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 determine the account type, see Checking Account Type.	Connections	60、300
IntrafficVipRatio	Inbound bandwidth utilization	Utilization of bandwidth used by clients to access 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 Checking Account Type. This metric is currently in beta. To try it out, submit a ticket.	%	10、60、300
OuttrafficVipRatio	Outbound bandwidth utilization	Utilization of bandwidth used by a CLB instance to access 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 Checking Account Type. This metric is currently in beta. To try it out, submit a ticket.	%	10、60、300
ReqAvg	Average request time	Average request time of a CLB instance in the statistical period. This metric only available for layer-7 listeners.	Millisecond (ms)	60、300
ReqMax	Maximum request time	Maximum request time of a CLB instance in the statistical period. This metric only available for layer-7 listeners.	Millisecond (ms)	60、300
RspAvg	Average response time	Average response time of a CLB instance in the statistical period. This metric only available for layer-7 listeners.	Millisecond (ms)	60、300
RspMax	Maximum response time	Maximum response time of a CLB instance in the statistical period. This metric only available for layer-7 listeners.	Millisecond (ms)	60、300
RspTimeout	Timed-out responses	Number of CLB timed-out responses in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
SuccReq	Successful requests per minute	Number of successful requests per minute of a CLB instance in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
TotalReq	Requests per second	Number of requests per second of a CLB instance in the statistical period. This metric only available for layer-7 listeners.	Connections	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 only available for layer-7 listeners.	pieces/minute	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 only available for layer-7 listeners.	pieces/minute	60、300

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 only available for layer-7 listeners.	pieces/minute	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 only available for layer-7 listeners.	pieces/minute	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 only available for layer-7 listeners.	pieces/minute	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 only available for layer-7 listeners.	pieces/minute	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 only available for layer-7 listeners.	pieces/minute	60、300
ClbHttp504	504 status codes returned by CLB	Number of 504 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 only available for layer-7 listeners.	pieces/minute	60、300
Http2xx	2xx status codes	Number of 2xx status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
Http3xx	3xx status codes	Number of 3xx status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
Http4xx	4xx status codes	Number of 4xx status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
Http5xx	5xx status codes	Number of 5xx status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
Http404	404 status codes	Number of 404 status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
Http499	499 status codes	Number of 499 status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
Http502	502 status codes	Number of 502 status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300

Http503	503 status codes	Number of 503 status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
Http504	504 status codes	Number of 504 status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	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 in beta testing. To try it out, please submit a beta application.	pieces/minute	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) This metric is in beta testing. To try it out, please submit a beta application.	%	60
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 in beta testing. To try it out, please submit a beta application.	pieces/minute	60
UnhealthRsCount	Abnormal health checks	Number of abnormal health checks of a CLB instance in the statistical period.	Connections	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	Meaning	MetricsDescription	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.	Connections	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.	keys/second	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.	keys/second	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.	keys/second	10、60、300
ClientAccIntraffic	Client-to-CLB traffic in	Bandwidth used by 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 from a CLB instance	Mbps	10、60、

	bandwidth out	to clients in the statistical period.		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.	keys/second	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.	keys/second	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.	Connections	60、300
NewConn	CLB-to-real server new connections	Number of new connections from a CLB instance to real servers in the statistical period.	pieces/minute	60、300
UnhealthRsCount	Abnormal health checks	Number of abnormal health checks of a CLB instance in the statistical period.	Connections	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	Meaning	MetricsDescription	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.	Connections	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.	keys/second	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.	keys/second	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.	keys/second	10、60、300
ClientAccIntraffic	Client-to-CLB traffic in	Bandwidth used by 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 from a CLB instance	Mbps	10、60、

	bandwidth out	to clients in the statistical period.		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.	keys/second	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.	keys/second	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.	Connections	60、300
NewConn	CLB-to-real server new connections	Number of new connections from a CLB instance to real servers in the statistical period.	pieces/minute	60、300
ReqAvg	Average request time	Average request time of a CLB instance in the statistical period. This metric only available for layer-7 listeners.	Millisecond (ms)	60、300
ReqMax	Maximum request time	Maximum request time of a CLB instance in the statistical period. This metric only available for layer-7 listeners.	Millisecond (ms)	60、300
RspAvg	Average response time	Average response time of a CLB instance in the statistical period. This metric only available for layer-7 listeners.	Millisecond (ms)	60、300
RspMax	Maximum response time	Maximum response time of a CLB instance in the statistical period. This metric only available for layer-7 listeners.	Millisecond (ms)	60、300
RspTimeout	Timed-out responses	Number of CLB timed-out responses in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
SuccReq	Successful requests per minute	Number of successful requests per minute of a CLB instance in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
TotalReq	Requests per second	Number of requests per second of a CLB instance in the statistical period. This metric only available for layer-7 listeners.	Connections	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 only available for layer-7 listeners.	pieces/minute	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	pieces/minute	60、300

		by both the CLB instance and real servers are counted.) This metric only available for 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 only available for layer-7 listeners.	pieces/minute	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 only available for layer-7 listeners.	pieces/minute	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 only available for layer-7 listeners.	pieces/minute	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 only available for layer-7 listeners.	pieces/minute	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 only available for layer-7 listeners.	pieces/minute	60、300
ClbHttp504	504 status codes returned by CLB	Number of 504 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 only available for layer-7 listeners.	pieces/minute	60、300
Http2xx	2xx status codes	Number of 2xx status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
Http3xx	3xx status codes	Number of 3xx status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
Http4xx	4xx status codes	Number of 4xx status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
Http5xx	5xx status codes	Number of 5xx status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
Http404	404 status codes	Number of 404 status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
Http499	499 status codes	Number of 499 status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
Http502	502 status codes	Number of 502 status codes returned by real servers in the statistical period.	pieces/minute	60、300

		This metric only available for layer-7 listeners.		
Http503	503 status codes	Number of 503 status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
Http504	504 status codes	Number of 504 status codes returned by real servers in the statistical period. This metric only available for layer-7 listeners.	pieces/minute	60、300
UnhealthRsCount	Abnormal health checks	Number of abnormal health checks of a CLB instance in the statistical period.	Connections	60、300

Note:
If you need to view the monitoring data for a backend server under a specific listener, please log in to the Cloud Load Balancer Console, click the monitoring icon next to the Cloud Load Balancer instance ID, and quickly browse the performance data of each instance through the monitoring floating window.

Documentation

Public Network Cloud Load Balancer Monitoring Metrics

Configuring Alarm Policy

Last updated: 2023-09-05 17:30:58

This document describes how to create an alarm policy.

Scenarios

You can set threshold alarms for performance consumption metrics supported by Basic Cloud Monitor to promptly notify you to take action in case of anomalies. An alarm policy consists of five essential components: name, policy type, alarm trigger conditions, alarm objects, and alarm notification templates. You can create an alarm policy following the guidelines below.

Concepts

Term	Description
Alarm policy	Composed of alarm name, alarm policy type, alarm trigger conditions, alarm objects, and alarm notification templates.
Alarm policy type	Alarm policy types are used to identify policy categories, and each type corresponds to a Tencent Cloud service. For example, when you choose the Cloud Virtual Machine policy, you can customize alarms for metrics such as CPU usage and disk usage.
Alarm trigger condition	A semantic condition consisting of metric, comparison relationship, threshold, statistical granularity, and N consecutive monitoring data points.
Test mode	Includes cloud product monitoring, application performance observation, frontend performance monitoring, and cloud probing.
Notification Template	Reuse templates with one click for multiple policies, suitable for receiving alarm notifications in various scenarios. For more information, please refer to Creating a New Alarm Notification Template.

Instructions

1. Log in to the Basic Cloud Monitor.
2. Click **Alarm Management** > **Policy Management** to enter the alarm policy configuration page.
3. Click **Create Policy** and configure the alarm policy with the following instructions:

Configuration Specifications	Configuration items	Note
Basic Information	Rule Name	The custom policy name.
	Remarks	Custom policy remarks.
	Test mode	Supports cloud product monitoring, application performance observation, frontend performance monitoring, and cloud probing.
	Policy types	Select the desired policy type of the Tencent Cloud service to be monitored.
	Project	This configuration item is useful in the following two aspects: • Manage alarm policies. After setting the associated project, you can quickly filter the alarm policies under that project in the alarm policy list. • Manage instances by selecting a project based on your needs, and quickly choose instances under that project in the alarm objects. You can allocate cloud products to different projects according to your business type. To create a project, refer to Project Management. After creating a project, you can assign resources to projects in various cloud product consoles, although some cloud products do not support project allocation. (For example, for TencentDB for MySQL, refer to Assigning Instance to Project to allocate instances to corresponding projects.) If you do not have project permissions, refer to Cloud Access Management to grant permissions.

Configuring Alarm Rules	Alert object	• Instance ID: associate the policy with the specified CVM instance chosen by the user. • If you select "Instance group", the alarm policy will be associated with the selected instance group. • If you select **All Objects**, the alarm policy will be associated with all instances under the current account.
	Manual Configuration (Metric alarm)	• Alarm Trigger Condition: a semantic condition consisting of metric, comparison relationship, threshold, statistical granularity, and N consecutive monitoring data points. You can set alarm thresholds based on the metric trends in the chart. For example, with a metric of CPU utilization, comparison relationship of ">", threshold of 80%, statistical granularity of 5 minutes, and 2 consecutive monitoring data points. This means: CPU utilization data is collected every 5 minutes, and if a Cloud Virtual Machine's CPU utilization exceeds 80% for two consecutive times, an alarm will be triggered. • Alarm Frequency: You can set a repeat notification policy for each of your alarm rules. That is, when an alarm is generated, you can define the frequency at which the alarm is repeatedly notified. Frequency options: do not repeat, once every 5 minutes, once every 10 minutes, and other exponentially increased frequencies. ○ The meaning of exponential increase in periods is that when the alarm is triggered for the 1st, 2nd, 4th, 8th... 2 to the power of N times, an alarm notification will be sent to you. The significance is that the interval between alarm notifications will become longer, reducing the disturbance caused by repeated alarms to some extent. ○ Default logic for repeated alarms: Within 24 hours after an alarm is generated, alarm notifications will be sent to you repeatedly at the frequency you set. After the alarm has been active for 24 hours, alarm notifications will be sent once a day.
	Trigger condition (choosing Select template)	Click the Select Template button and choose a configured template from the dropdown list. For specific configuration, please refer to Configure Trigger Condition Template. If the newly created template is not displayed, click "Refresh" on the right to refresh the alarm template selection list.
	Alarm Notification	Alarm notifications support the selection of both system preset notification templates and user-defined notification templates. Each alarm policy can be bound to a maximum of three notification templates. For more information, please refer to Notification Templates.
Advanced Configurations	Elastic 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 **Complete** to successfully create the alarm policy.

> **Note:**
> Cloud Virtual Machine alarms require the Cloud Virtual Machine instance to install monitoring components and report monitoring metric data before alarms can be sent properly. You can view Cloud Virtual Machines without monitoring agents installed on the cloud product monitoring page and download the IP list.

Alarming Metric Descriptions

Last updated: 2023-09-05 17:34:57

Alert Policies

You can create alert policies for instance metrics you care about, allowing Cloud Load Balance to promptly send alerts to the concerned user groups when the running status meets certain conditions. This ensures that you can quickly detect anomalies and take appropriate measures to maintain system stability and reliability. For more information, please refer to Alert Overview.
The alert policies for Cloud Load Balance include the following types:

- Public Network Listener
- Private Network Listener
- Real Server
  - Listener level
  - Server port level
- Server Port (Traditional Private Network)
- Layer-7 Protocol Monitoring

Public/Private Network Listeners

Supported network listener metrics:

Metrics	Unit	Note
Bandwidth in	Mbps	Bandwidth consumed when clients access the Cloud Load Balance through the public network within a sampling period.
Outbound Bandwidth	Mbps	Bandwidth consumed by the Cloud Load Balance to access the public network within a sampling period.
Inbound Packets	count/s	Number of request data packets received by the Cloud Load Balance per second within the statistical period.
Outbound Packets	count/s	Number of data packets sent per second by Cloud Load Balance within a sampling period.

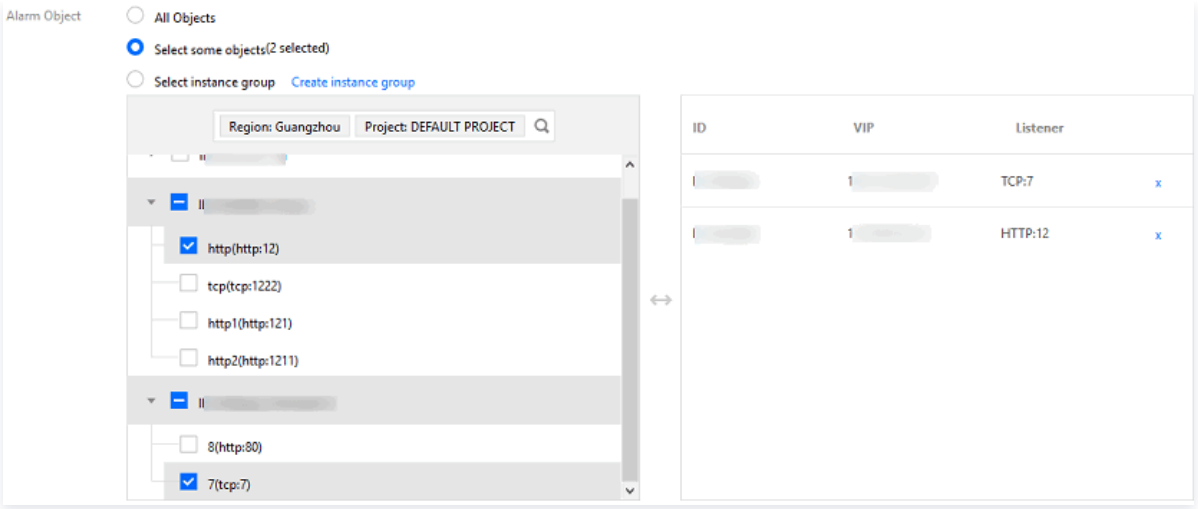
Real Server

CLB supports alert policies on the listener and server port level.

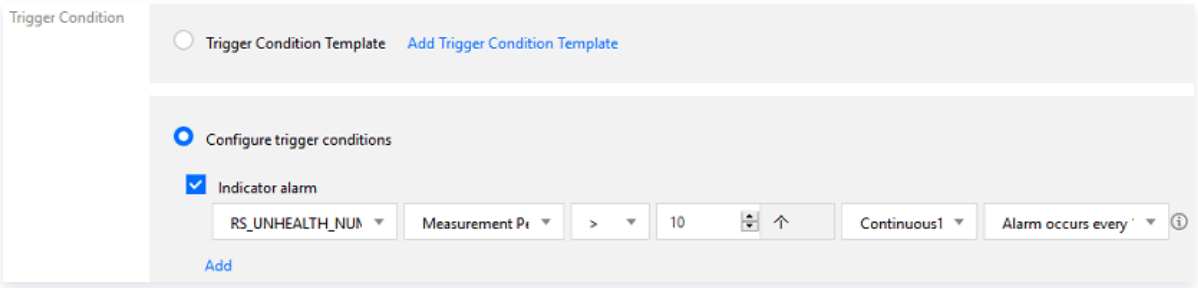
1. Listener level
   Set up a policy to trigger alerts when the number of abnormal ports of real servers bound to the listener reaches the threshold. In the example below, the number of abnormal ports of all real servers under the selected listener is collected once every minute. If the number is greater than 10 per second for two consecutive sampling periods, an alert is triggered. Only one alert is sent per day.

   > **Note**
   > To activate listener-level alert, submit a ticket.

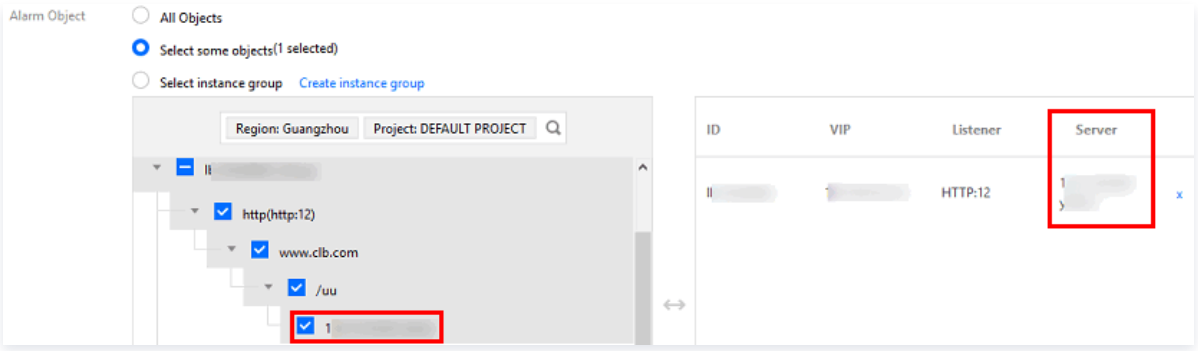
   - Policy Type Selection:



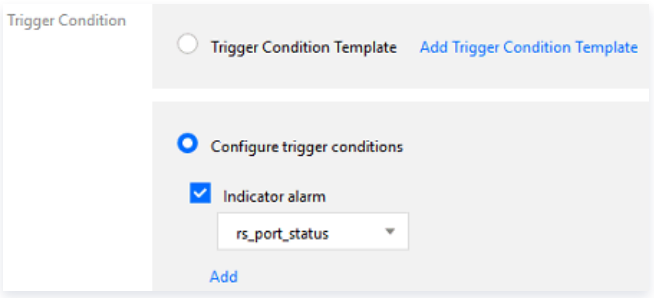
- Configure Trigger Conditions:



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.
   - Policy Type Selection:



   - 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.

Server Port (Traditional Private Network)

You can configure server port exception alarms for private network Classic CLB as instructed in "Server Port (Other) > 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.

Layer-7 Protocol Monitoring

For all Layer-7 listeners (HTTP/HTTPS), you can configure alert policies specific to Layer-7 monitoring metrics. The detailed metrics are as follows:

Metrics	<p>	Note
Bandwidth in	Mbps	Bandwidth consumed when clients access the Cloud Load Balance through the public network within a sampling period.
Outbound Bandwidth	Mbps	Bandwidth consumed by the Cloud Load Balance to access the public network within a sampling period.
Inbound Packets	count/s	Number of request data packets received by Cloud Load Balance per second within a sampling period.
Outbound Packets	count/s	Number of data packets sent per second by Cloud Load Balance within a sampling period.
New connections	Connections	Number of new connections established per minute within a sampling period.
Number of active connections	Connections	Number of active connections per minute within a sampling period.
Average response time	ms	Average response time of CLB within a sampling period.
Maximum response time	ms	Longest response time of CLB within a sampling period.
2xx status codes	Connections	Number of 2xx status codes returned by the real server within a sampling period.
3xx status codes	Connections	Number of 3xx status codes returned by the real server within a sampling period.
4xx status codes	Connections	Number of 4xx status codes returned by the real server within a sampling period.
5xx status codes	Connections	Number of 5xx status codes returned by the real server within a sampling period.
404 status codes	Connections	Number of 404 status codes returned by the real server within a sampling period.
502 status codes	Connections	Number of 502 status codes returned by real servers in the statistical period.
3xx status codes returned by CLB	Connections	Number of 3xx status codes returned by Cloud Load Balance (CLB) within a sampling period.
4xx status codes returned by CLB	Connections	Number of 4xx status codes returned by Cloud Load Balance (CLB) within a sampling period.
5xx status codes returned by CLB	Connections	Number of 5xx status codes returned by Cloud Load Balance (CLB) within a sampling period.
404 status codes returned by CLB	Connections	Number of 404 status codes returned by Cloud Load Balance (CLB) within a sampling period.
502 status codes returned by CLB	Connections	Number of 502 status codes returned by Cloud Load Balance (CLB) within a sampling period.

Cloud Access Management Overview

Last updated: 2023-09-05 17:36:07

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 will be easily compromised because it is shared by several users.
- Your users might introduce security risks from maloperations due to the lack of user access control.

Cloud Access Management (CAM) is used to manage access permissions for resources under Tencent Cloud accounts. With CAM, you can control which sub-accounts have access to specific resources through identity management and policy management. 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 sub-account access permissions via CLB, you can skip this section. Doing so does not affect your understanding and use of the rest of the documentation.

Basic CAM Concepts

The root account authorizes sub-accounts by associating policies. The policy setting can be specific to the level of **[API, Resource, User/User Group, Allow/Deny, and Condition]**.

1. Account
   - **Root account**
     The primary entity responsible for Tencent Cloud resource ownership, usage measurement, and billing, which can log in to Tencent Cloud services.
   - **Sub-account**
     Created by the root account, a sub-account has a specific identity ID and credentials, and can log in to the Tencent Cloud console. The root account can create multiple sub-accounts (users). **By default, sub-accounts do not have access to resources and must be authorized by their associated root account.**
   - **Identity credential**
     includes login credentials and access certificates. **Login credential** refers to a user's login name and password. **Access certificate** refers to Tencent Cloud API keys (SecretId and SecretKey).
2. Resource and permission
   - **Resource**
     A resource is an object that is operated in cloud services, such as a Cloud Virtual Machine instance, a VPC instance, and so on.
   - **Permission**
     Permission refers to the authorization that allows or denies certain users to perform specific operations. By default, the **root account has full access to all resources under its name**, while a **sub-account does not have access to any resources under the root account.**
   - **Policy**
     A policy is a syntax rule that defines and describes one or more permissions. The **root account** performs authorization by **associating policies** with users/user groups.

For more information, please refer to CAM Overview.

Documentation

Content	Document
Understand the relationship between policies and users	Policy Management
Understand the basic structure of policies	Policy Syntax
Check CAM-enabled products	List of CAM-Supported Cloud Services

Authorization Definition

Last updated: 2023-09-05 17:38:08

Types of CLB Resources That Can Be Authorized in CAM

ResourceType	Resource Description Method in Authorization Policy
CLB instance	`qcs::clb:$region::clb/$loadbalancerid`
CLB real server	`qcs::cvm:$region:$account:instance/$cvminstanceid`

In the description:

- All `$region` should be an ID of a specific region and can be left empty.
- All `$account` should be the resource owner's AccountId, or "*".
- All `$loadbalancerid` should be the ID of a specific load balancer, or "*".

And so on…

APIs for CLB Authorization in CAM

You can authorize the following actions for a CLB resource in CAM.

Instance

Configuring using API	Description	Format
DescribeLoadBalancers	DescribeLoadBalancers	`*` Authenticate only the API
CreateLoadBalancer	CreateLoadBalancer	`qcs:$projectid:clb:$region:$account:clb/*`
DeleteLoadBalancers	Deletes CLB instances.	`qcs::clb:$region:$account:clb/$loadbalancerid`
ModifyLoadBalancerAttributes	ModifyLoadBalancerAttributes	`qcs::clb:$region:$account:clb/$loadbalancerid`
ModifyForwardLBName	Modifies the name of a CLB instance.	`qcs::clb:$region:$account:clb/$loadbalancerid`
SetLoadBalancerSecurityGroups	Configures security groups for CLB instance	`qcs::clb:$region:$account:clb/$loadbalancerid`

Listener

Configuring using API	Description	Format
DeleteLoadBalancerListeners	DeleteLoadBalancerListeners	`qcs::clb:$region:$account:clb/$loadbalancerid`
DescribeLoadBalancerListeners	DescribeLoadBalancerListeners	`qcs::clb:$region:$account:clb/$loadbalancerid`
ModifyLoadBalancerListener	ModifyLoadBalancerListener	`qcs::clb:$region:$account:clb/$loadbalancerid`
CreateLoadBalancerListeners	CreateLoadBalancerListeners	`qcs::clb:$region:$account:clb/$loadbalancerid`
DeleteForwardLBListener	Deletes a layer-4 or layer-7 CLB listener.	`qcs::clb:$region:$account:clb/$loadbalancerid`

ModifyForwardLBSeventhListener	Modifies the attributes of a layer-7 CLB listener.	qcs::clb:$region:$account:clb/$loadbalancerid
ModifyForwardLBFourthListener	ModifyForwardLBFourthListener	qcs::clb:$region:$account:clb/$loadbalancerid
DescribeForwardLBListeners	DescribeForwardLBListeners	qcs::clb:$region:$account:clb/$loadbalancerid
CreateForwardLBSeventhLayerListeners	CreateForwardLBSeventhLayerListeners	qcs::clb:$region:$account:clb/$loadbalancerid
CreateForwardLBFourthLayerListeners	CreateForwardLBFourthLayerListeners	qcs::clb:$region:$account:clb/$loadbalancerid

CLB domain name and URL

Configuring using API	Description	Format
ModifyForwardLBRulesDomain	ModifyForwardLBRulesDomain	qcs::clb:$region:$account:clb/$loadbalancerid
CreateForwardLBListenerRules	CreateForwardLBListenerRules	qcs::clb:$region:$account:clb/$loadbalancerid
DeleteForwardLBListenerRules	DeleteForwardLBListenerRules	qcs::clb:$region:$account:clb/$loadbalancerid
DeleteRewrite	Deletes the redirection relationship between CLB forwarding rules.	qcs::clb:$region:$account:clb/$loadbalancerid
ManualRewrite	Manually creates a redirection relationship between CLB forwarding rules.	qcs::clb:$region:$account:clb/$loadbalancerid
AutoRewrite	Automatically generates a redirection relationship between CLB forwarding rules.	qcs::clb:$region:$account:clb/$loadbalancerid

Real server

Configuring using API	Description	Format
ModifyLoadBalancerBackends	Modifies real server weights for a CLB instance.	qcs::clb:$region:$account:clb/$loadbalancerid
DescribeLoadBalancerBackends	DescribeLoadBalancerBackends	qcs::clb:$region:$account:clb/$loadbalancerid
DeregisterInstancesFromLoadBalancer	Unbinds real servers.	qcs::clb:$region:$account:clb/$loadbalancerid qcs::cvm:$region:$account:instance/$cvminstanceid
RegisterInstancesWithLoadBalancer	RegisterInstancesWithLoadBalancer	qcs::clb:$region:$account:clb/$loadbalancerid qcs::cvm:$region:$account:instance/$cvminstanceid
DescribeLBHealthStatus	Queries the health status of a CLB instance.	qcs::clb:$region:$account:clb/$loadbalancerid

ModifyForwardFourthBackendsPort	ModifyForwardFourthBackendsPort	qcs::clb:$region:$account:clb/$loadbalancerid qcs::cvm:$region:$account:instance/$cvminstanceid
ModifyForwardFourthBackendsWeight	ModifyForwardFourthBackendsWeight	qcs::clb:$region:$account:clb/$loadbalancerid qcs::cvm:$region:$account:instance/$cvminstanceid
RegisterInstancesWithForwardLBSeventhListener	RegisterInstancesWithForwardLBSeventhListener	qcs::clb:$region:$account:clb/$loadbalancerid qcs::cvm:$region:$account:instance/$cvminstanceid
RegisterInstancesWithForwardLBFourthListener	RegisterInstancesWithForwardLBFourthListener	qcs::clb:$region:$account:clb/$loadbalancerid qcs::cvm:$region:$account:instance/$cvminstanceid
DeregisterInstancesFromForwardLBFourthListener	DeregisterInstancesFromForwardLBFourthListener	qcs::clb:$region:$account:clb/$loadbalancerid qcs::cvm:$region:$account:instance/$cvminstanceid
DeregisterInstancesFromForwardLB	DeregisterInstancesFromForwardLB	qcs::clb:$region:$account:clb/$loadbalancerid qcs::cvm:$region:$account:instance/$cvminstanceid
ModifyForwardSeventhBackends	ModifyForwardSeventhBackends	qcs::clb:$region:$account:clb/$loadbalancerid qcs::cvm:$region:$account:instance/$cvminstanceid
ModifyForwardSeventhBackendsPort	ModifyForwardSeventhBackendsPort	qcs::clb:$region:$account:clb/$loadbalancerid qcs::cvm:$region:$account:instance/$cvminstanceid
DescribeForwardLBBackends	DescribeForwardLBBackends	qcs::clb:$region:$account:clb/$loadbalancerid qcs::cvm:$region:$account:instance/*
DescribeForwardLBHealthStatus	DescribeLBHealthStatus	qcs::clb:$region:$account:clb/*
ModifyLoadBalancerRulesProbe	ModifyLoadBalancerRulesProbe	qcs::clb:$region:$account:clb/$loadbalancerid

Policy Examples

Last updated: 2023-09-05 17:38:49

Full Access Policy for All CLBs

- Grant a sub-account full access (including resource creation and management) to the CLB service.
- Policy Name: CLBResourceFullAccess

```
{
  "version":"2.0",
  "statement": [{
      "action": [
          "name/clb:*"
      ],
      "resource":"*",
      "effect": "allow"
  }]
}
```

Read-Only Policy for All CLBs

- Grant a sub-account read-only access to CLB resources (i.e., permission to view all resources under all CLBs), but not the ability to create, update, or delete them. In the console, the prerequisite for operating a resource is being able to view it, so it is recommended to grant full read access to CLB for sub-accounts.
- Policy Name: CLBResourceReadOnlyAccess

```
{
  "version":"2.0",
  "statement": [{
      "action": [
          "name/clb:Describe*"
      ],
      "resource":"*",
      "effect": "allow"
  }]
}
```

Full Access Policy for CLBs under a Specific Tag

- Grant a sub-account full access to CLBs with a specific tag (tag key as tagkey, tag value as tagvalue), including all operations such as managing instances and listeners. For more information about tags, see Managing Project Resources Based on Tags.
- CLB instances support configuring tags and utilizing tag-based 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: 2023-09-05 17:41:16

Overview

Classic CLB is easy to configure and supports simple load balancing scenarios:
- Classic **Public** CLB: Supports TCP/UDP/HTTP/HTTPS protocols.
- Classic **Private** 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, Tencent Cloud accounts are divided into standard and traditional account types. All accounts registered after June 17, 2020, 00:00:00 are standard account types. For accounts registered before this time, please check your account type in the console and refer to Determining Account Type. Standard account types no longer support traditional CLB, and all purchased instances are CLB.

This article introduces the classic CLB instance. 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 real servers based on the load balancing policy.

Listener Configuration Guide

You need to configure a CLB listener with the following items:
1. Listener protocol and listening port. A listening port, aka frontend port, is used to receive and route requests to the real server.
2. Backend port. It is the port through which the CVM instance provides services and receives and processes the traffic from the CLB instance.
3. Listening policies, such as balancing policy and session persistence.
4. Health check policy.
5. Real server which can be bound by selecting its IP.

> **Note:**
> If you configure multiple listeners on a classic CLB instance and bind multiple real servers, each listener will route requests to all real servers according to its own configuration.

Supported Protocol Types

CLB listeners can monitor layer-4 and layer-7 requests on CLB instances and distribute them to real servers for processing. The main difference between layer-4 and layer-7 CLB lies in whether the traffic is forwarded based on layer-4 or layer-7 protocols 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 require the aforementioned advanced capabilities, please use CLB instead of classic CLB. For more information, see Instance Types.

Port Configuration

Listening Port (frontend port)	Service Port (backend port)	Note
The listening port is used by a CLB instance to receive and forward requests to real servers for load balancing. You can configure CLB for the port range 1–65535, such as 21 (FTP), 25 (SMTP), 80 (HTTP), and 443 (HTTPS).	The service port is the port through which the CVM instance provides services, receives, and processes traffic from the CLB instance. In a single CLB instance, the same listening port can forward traffic to multiple ports on multiple CVM instances.	Within the same 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. 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 Cloud Virtual Machine instance simultaneously.

Configuring Classic CLB

Last updated: 2023-09-05 18:18:00

After creating a classic Cloud Load Balancer (CLB) instance, you need to configure a listener for it. The listener is responsible for monitoring requests on the CLB instance and distributing traffic to backend servers based on the load balancing policy.

Preparations

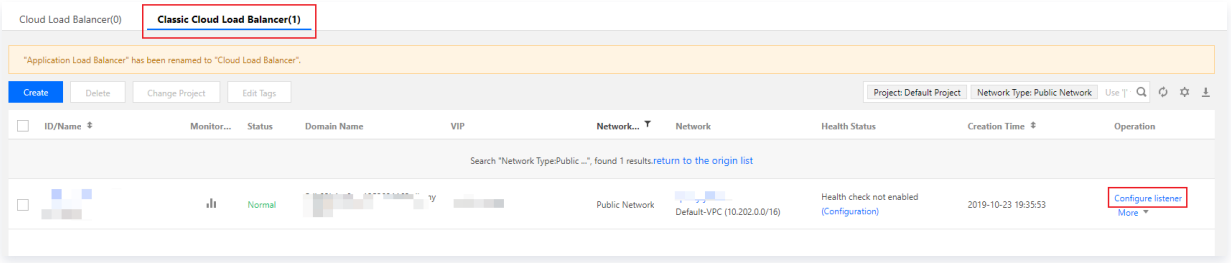
You need to create a Cloud Load Balancer instance, selecting "Classic Cloud Load Balancer" as the instance type.

> **Note:**
> Currently, Tencent Cloud accounts are divided into standard and traditional account types. All accounts registered after June 17, 2020, 00:00:00 are standard account types. For accounts registered before this time, please check your account type in the console and refer to Determining Account Type. Standard account types no longer support traditional CLB, and all purchased instances are CLB.

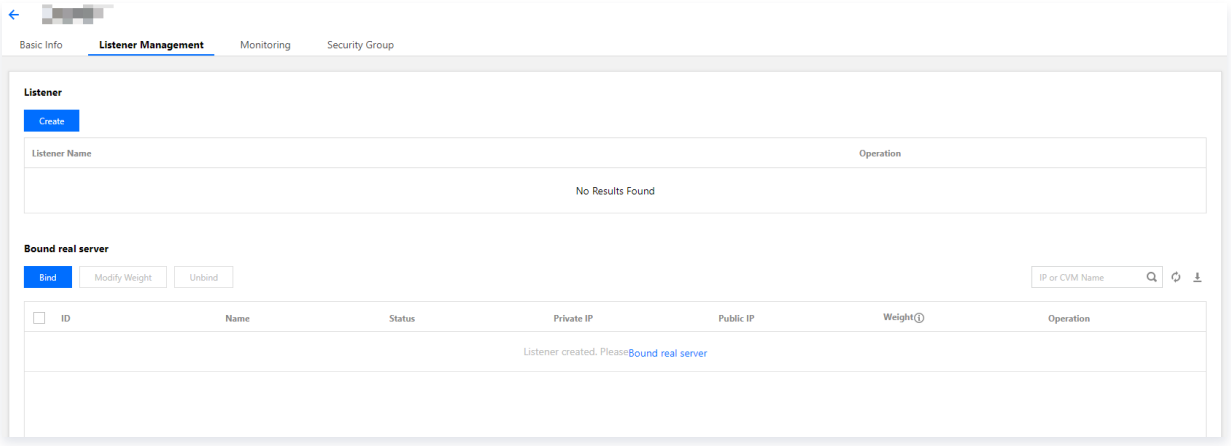
Configure listener

Step 1. Open the "Listener Management" page

1. Log in to the Cloud Load Balancer 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. Click on the **Listener Management** tab, or you can also click on **Configure Listener** in the operation column on the list page.



5. The **Listener Management** page is as shown below.



Step 2. Configure a listener

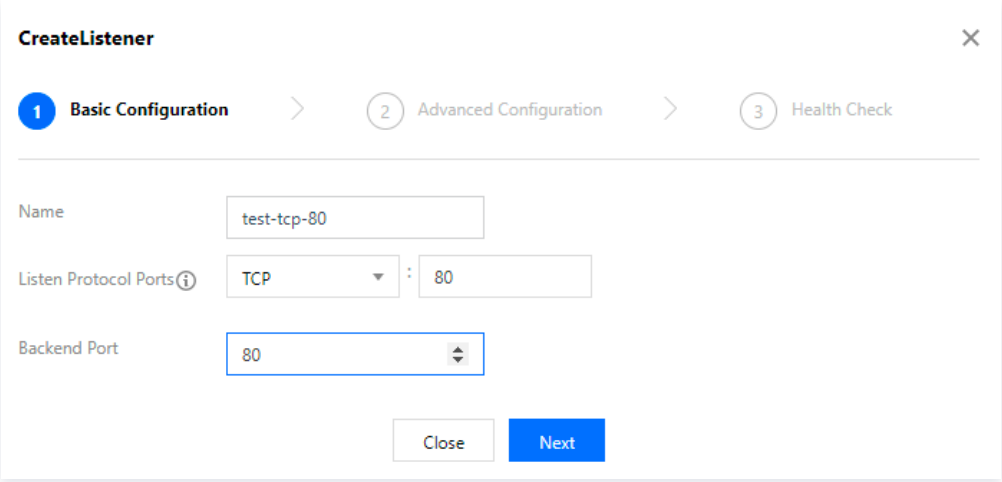
Click **Create** under the **Listener** module, and configure a TCP listener in the pop-up window.

1. Basic Configuration

Configuration Item	Note	Sample

Name	Listener name.	test-tcp-80
Listener Protocol and Ports	Listener protocol and listening port Listener protocol: CLB supports protocols such as TCP, UDP, HTTP, and HTTPS. This example uses TCP. Listening port: The port used to receive requests and forward them to the real server. The port number ranges from 1 to 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

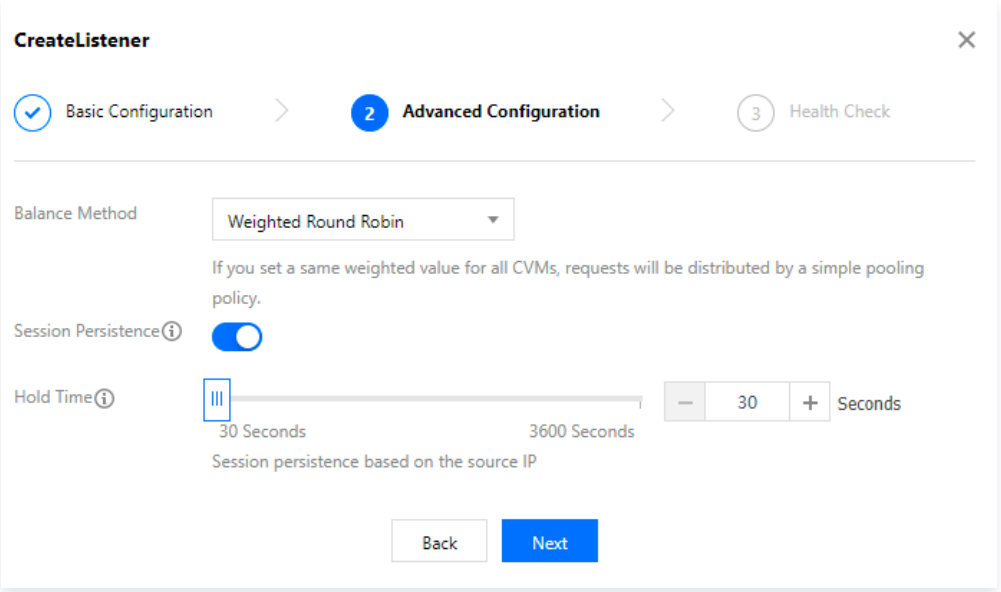
To create a TCP listener, complete the basic configuration as shown below:



2. Advanced Configuration

Advanced Configurations	Note	Sample
Balancing Method	CLB supports two scheduling algorithms for TCP listeners: weighted round robin (WRR) and weighted least connections (WLC). • Weighted Round Robin Algorithm: Requests are distributed to backend servers in sequence based on their weights. The weighted round robin algorithm schedules based on the **number of new connections**. Servers with higher weights have a higher probability of being polled, and servers with the same weight process the same number of connections. • Weighted Least Connections (WLC): Server loads 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 fewer current connections have a higher probability of being polled.	WRR
Session persistence	Enable or Disable 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
Session persistence duration.	Session persistence duration. • Session persistence is terminated if there are no new requests in the connection within the specified duration. • Value range: 30–3600 seconds.	30s

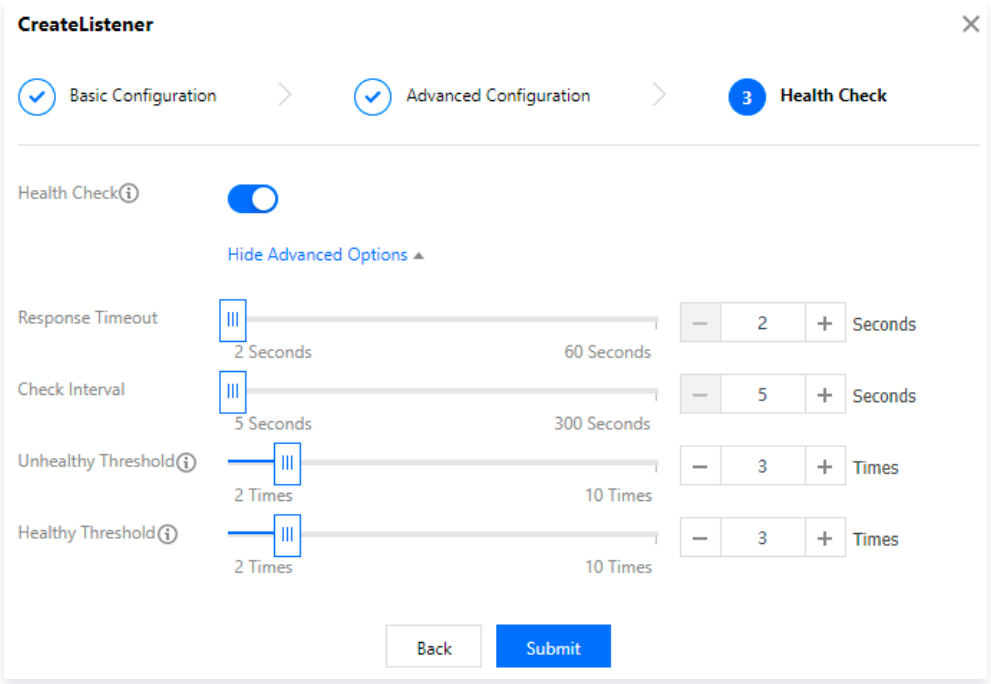
Complete the configuration as shown below:



3. Health check

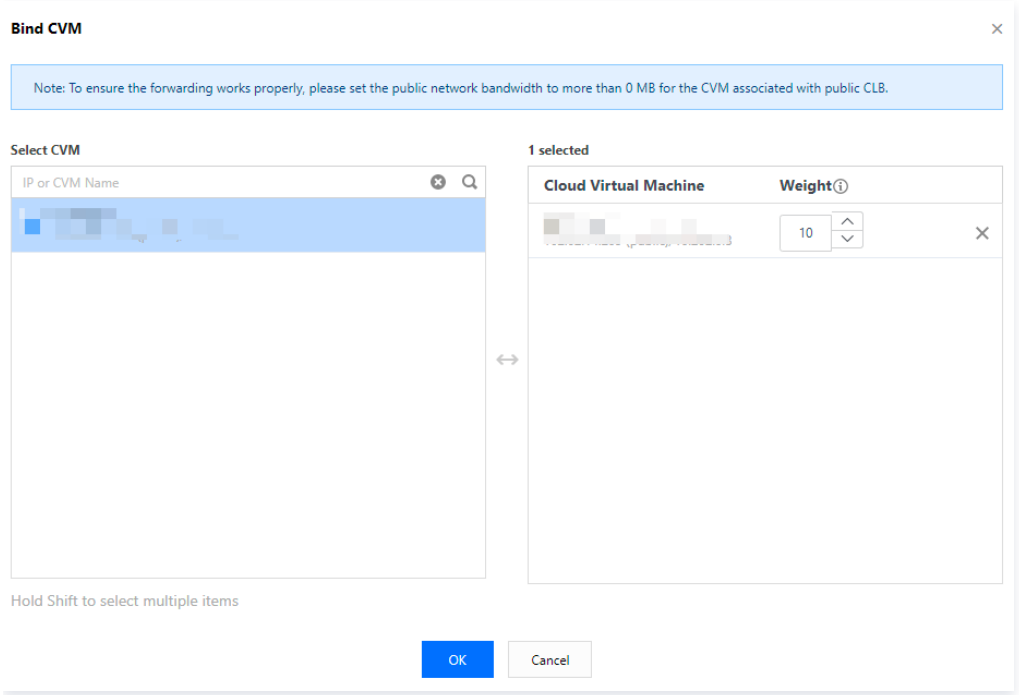
Health Check Configuration	Note	Sample
Health Check Status	Enable or disable health checks. In TCP listeners, the Cloud Load Balancer instance sends SYN packets to the specified server port to perform health checks.	Enabled
Check method	To be added	To be added
Port	To be added	To be added
Response timeout	• 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. Default value: 2s.	2s
Check interval	• Interval between two health checks. • Value range: 5–300 seconds. Default value: 5s.	5s
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: 2–10 times. Default value: 3 times.	3
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: 2–10 times. Default value: 3 times.	3

Complete the health check configuration as shown below:

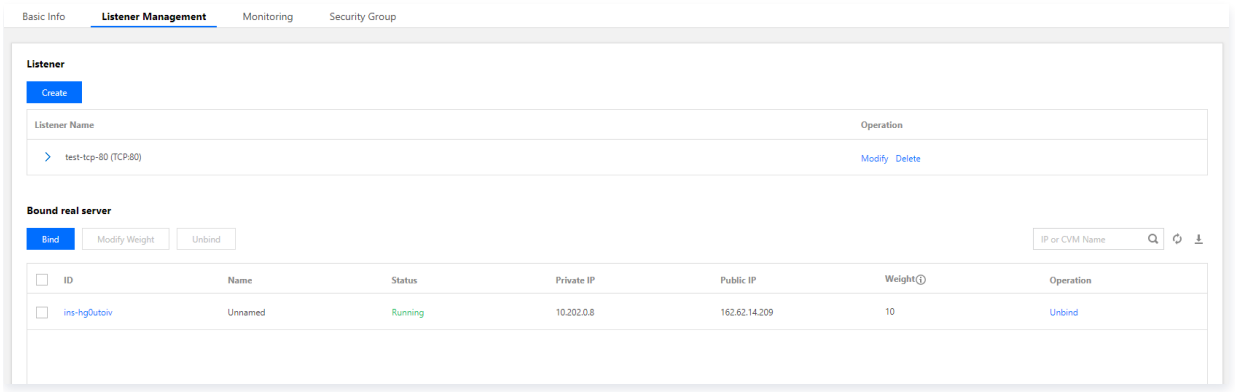


Step 3. Bind a CVM instance

On the "Listener Management" page, click the **Bind** button, and in the pop-up window, select the backend Cloud Virtual Machine (CVM) to be bound. The binding details are as follows:



The screenshot of the completed configuration is shown below:



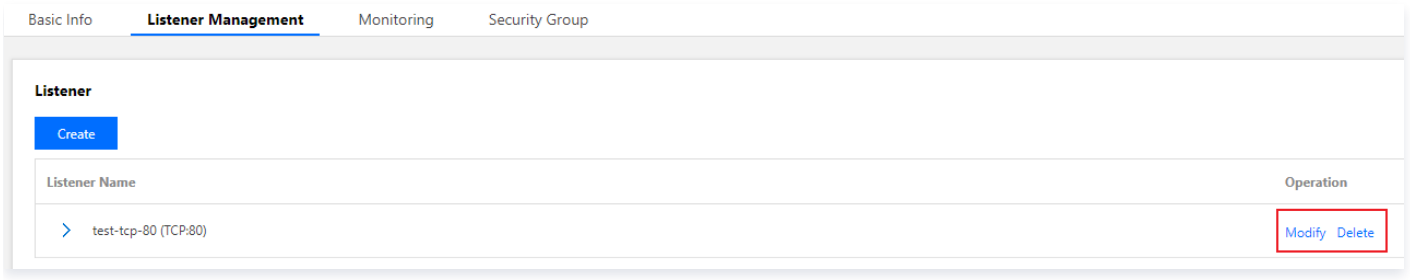
Note:
If you configure multiple listeners on a classic CLB instance and bind multiple real servers, each listener will route requests to all real servers according to its own configuration.

Step 4. Security group (optional)

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

Step 5. Modify/delete a listener (optional)

If you need to modify or delete an existing listener, please navigate to the "Listener Management" page, select the fully created listener, and choose **Modify** or **Delete** to complete the operation.



Managing Real Servers of Classic CLB Instances

Last updated: 2023-09-05 18:23:36

Traditional Cloud Load Balancer routes requests to healthy backend Cloud Virtual Machine instances. When using a traditional Cloud Load Balancer for the first time or when needing to add or remove backend servers based on business requirements, follow the guidance provided in this document.

Preparations

You must have created a traditional Cloud Load Balancer instance and configured a listener. For more information, please see Traditional Cloud Load Balancer Quick Start Guide.

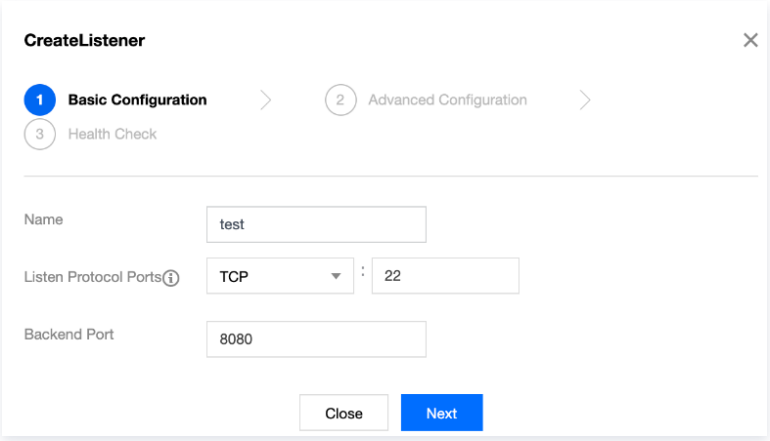
Instructions

Adding Backend Servers to a Traditional Cloud Load Balancer

> **Note**
> - If a traditional Cloud Load Balancer instance is associated with an Auto Scaling group, the Cloud Virtual Machines in that group will be automatically added to the backend Cloud Virtual Machines of the traditional Cloud Load Balancer. Cloud Virtual Machine instances removed from the Auto Scaling group will be automatically deleted from the backend Cloud Virtual Machines of the traditional Cloud Load Balancer.
> - To add backend servers using API, please refer to the Bind Backend Services to Traditional Cloud Load Balancer API documentation.

1. Log in to the Cloud Load Balancer console.
2. On the "Instance Management" page, click **Traditional Cloud Load Balancer**.
3. In the operation column on the right of the target traditional Cloud Load Balancer instance, click **Configure Listener**.
4. In the Listener Configuration module, click **Create**.
5. In the "Create Listener" dialog, fill in the "Backend Port" (for port selection, please refer to Common Server Ports) and other related fields. Click **Next** to continue completing the configuration. For more information, see Configuring Traditional Cloud Load Balancer.

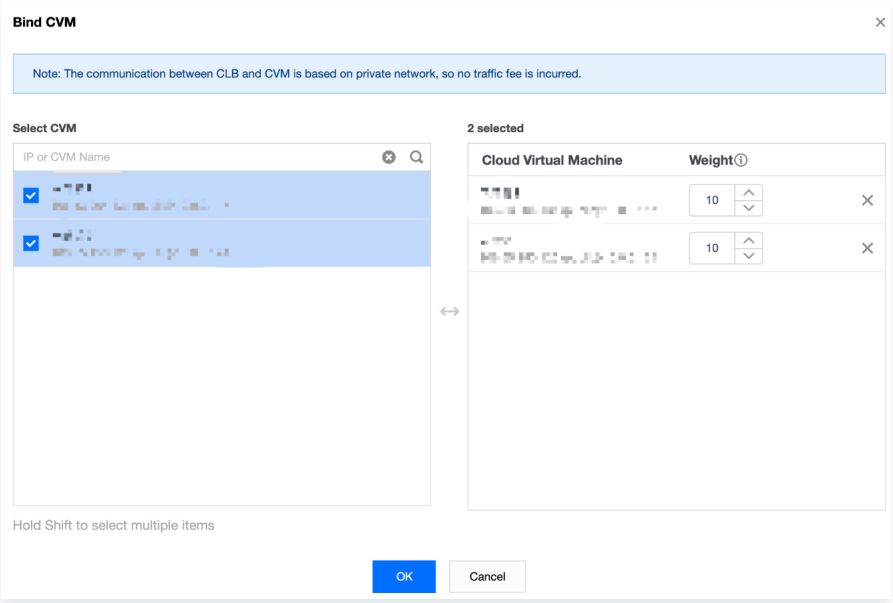
> **Note:**
> In a traditional Cloud Load Balancer, the backend server port must be specified during the **listener creation phase**.



6. After the listener is created, in the "Bind Backend Service" module, click **Bind**.
7. In the "Bind Cloud Virtual Machine" pop-up window, select the Cloud Virtual Machine instances to be bound, enter the weight information in the "Weight" field, and click **OK**.

> **Note:**

- The pop-up window only displays available Cloud Virtual Machines that are in the same region, in the same network environment, not isolated, not expired, and have a 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, the more requests are forwarded. The default value is 10, with a configurable range of 0–100. If the weight is set to 0, the server will no longer accept new requests. Enabling session persistence may result in uneven request distribution among backend servers. For more details, please see Load Balancing Algorithm Selection and Weight Configuration Examples.



Modifying Backend Server Weight for a Traditional Cloud Load Balancer

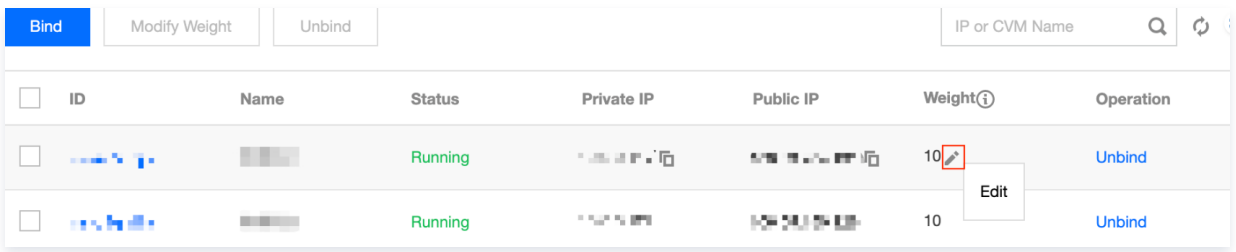
Note:
Modifying backend server weights using API is currently not supported for traditional Cloud Load Balancer.

1. Log in to the Cloud Load Balancer console.
2. On the "Instance Management" page, click **Traditional Cloud Load Balancer**.
3. In the operation column on the right of the target traditional Cloud Load Balancer instance, click **Configure Listener**.
4. In the Bind with Backend Service section, modify the corresponding server weight.

Note
The larger the weight, the more requests are forwarded. The default value is 10, with a configurable range of 0–100. If the weight is set to 0, the server will no longer accept new requests. Enabling session persistence may result in uneven request distribution among backend servers. For more details, please see Load Balancing Algorithm Selection and Weight Configuration Examples.

- **Method 1**: Modify the weight of a specific server individually.

  4.1.1 Locate the server for which you need to modify the weight, hover your mouse over the corresponding weight, and click the button.

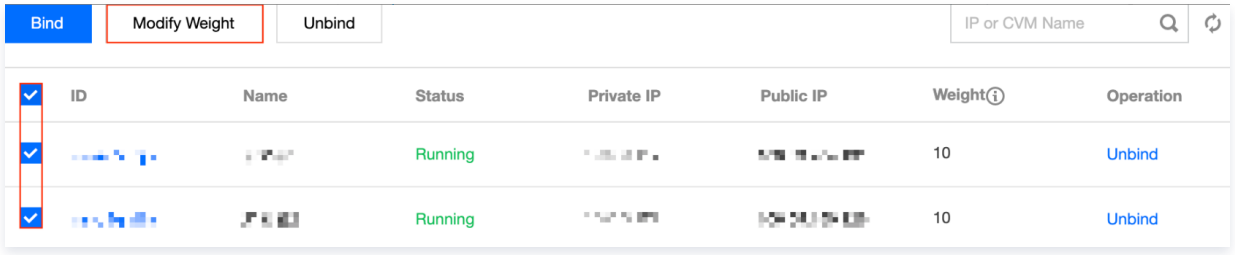


  4.1.2 In the "Modify Weight" pop-up window, enter the updated weight value and click **Submit**.

- **Method 2**: Batch modify the weight of certain servers.

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

4.1.1 Click the checkbox in front of the servers to select multiple servers, and then click **Modify Weight** at the top of the list.



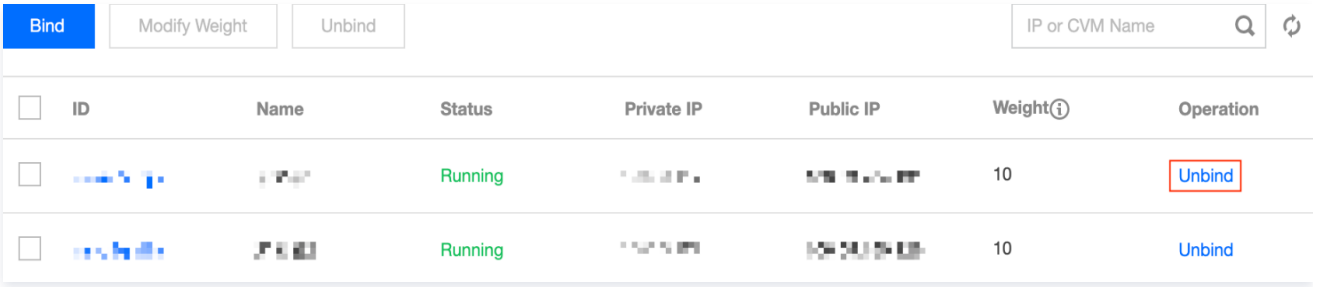
4.1.2 In the "Modify Weight" pop-up window, enter the updated weight value and click **Submit**.

Unbinding Backend Servers from a Traditional Cloud Load Balancer

> **Note:**
>
> Unbinding a backend server will dissociate the traditional Cloud Load Balancer instance from the Cloud Virtual Machine instance, and the traditional Cloud Load Balancer 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.
> To unbind backend servers using the API, please refer to the Unbinding Backend Servers from a Traditional Cloud Load Balancer API documentation.

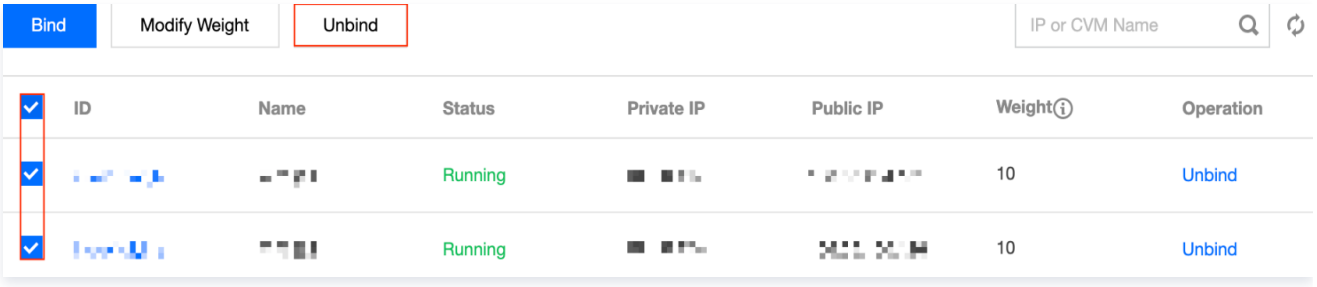
1. Log in to the Cloud Load Balancer console.
2. On the "Instance Management" page, click **Traditional Cloud Load Balancer**.
3. In the operation column on the right of the target traditional Cloud Load Balancer instance, click **Configure Listener**.
4. In the Bind with Backend Service module, unbind the already bound servers.
   - **Method 1**: Unbind a specific server individually.

     4.1.1 Locate the server you wish to unbind, and click **Unbind** in the operation column on the right.



     4.1.2 In the "Unbind Backend Service" pop-up window, confirm the service to be unbound and click **Submit**.
   - **Method 2**: Unbind multiple servers in bulk.

     4.1.1 Click the checkbox in front of the servers to select multiple servers, and then click **Unbind** at the top of the list.



4.1.2 In the "Unbind Backend Service" pop-up window, confirm the service to be unbound and click **Submit**.