ALIBABA CLOUD

Alibaba Cloud

Apsara Stack Enterprise

Server Load Balancer

User Guide

Product Version: v3.16.2

Document Version: 20220913

Alibaba Cloud

Legal disclaimer

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

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

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

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

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

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

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

Document conventions

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

Table of Contents

1.What is SLB?

This topic provides an overview of Server Load Balancer (SLB). SLB distributes network traffic across backend Elastic Compute Service (ECS) instances based on forwarding rules. SLB extends the service capability of applications and enhances their availability.

Overview

After you add ECS instances that are deployed in the same region to an SLB instance, SLB uses virtual IP addresses (VIPs) to virtualize the ECS instances into backend servers in a high-performance server pool that ensures high availability. Client requests are distributed to the ECS instances based on forwarding rules.

SLB checks the health status of the ECS instances and automatically removes unhealthy ECS instances from the pool to eliminate single points of failure. This improves the availability of your applications.

Components

SLB consists of the following components:

- SLB instances
  An SLB instance is a running entity of the SLB service that receives and distributes traffic to backend servers. To get started with SLB, you must create an SLB instance and add at least one listener and two ECS instances to the SLB instance.
- Listeners
  A listener checks client requests and forwards them to backend servers. A listener also performs health checks on backend servers.
- Backend servers
  ECS instances are used as backend servers in SLB to receive and process client requests. You can add ECS instances to the default server group of an SLB instance. You can also add multiple ECS servers to vServer groups or primary/secondary server groups after the corresponding groups are created.

Benefits

- High availability
  SLB is designed with full redundancy to eliminate single points of failure and support zone-disaster recovery. You can use SLB with services such as Apsara Stack DNS to implement geo-disaster recovery with 99.95% service uptime.
  SLB can be scaled based on application loads to ensure business continuity during traffic fluctuations.
- Scalability
  You can add or remove backend servers based on your business requirements to improve the availability of your service.
- Cost-effectiveness
  SLB can reduce 60% of load balancing costs compared with traditional hardware-based solutions.
- Security
  You can integrate SLB with Apsara Stack Security to defend your applications against DDoS attacks of up to 5 Gbit/s.
- High concurrency
  An SLB cluster supports hundreds of millions of concurrent connections and a single SLB instance supports tens of millions of concurrent connections.

2.Log on to the SLB console

This topic describes how to navigate to the Server Load Balancer (SLB) console after you log on to the Apsara Uni-manager Management Console by using Google Chrome.

Prerequisites

- Before you log on to the Apsara Uni-manager Management Console, you must obtain the endpoint of the console from the deployment personnel.
- We recommend that you use Google Chrome.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 8 to 20 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Login**.
4. In the top navigation bar, choose **Products > Networking > Server Load Balancer**.

3.Quick start

3.1. Overview

This topic describes how to create an Internet-facing Server Load Balancer (SLB) instance and distribute client requests to two backend Elastic Compute Service (ECS) instances.

> **Note** Before you create an SLB instance, you must decide the region, type, and billing method of the SLB instance. For more information, see Preparations.

This topic includes the following operations:

1. Create an SLB instance
   Create an SLB instance An SLB instance is a running entity of the SLB service.
2. Add listeners and backend servers
   After you create an SLB instance, you must add listeners and backend servers to the SLB instance.
3. Release an SLB instance
   You can release an SLB instance that is no longer needed to reduce costs.

3.2. Preparations

Before you create a Server Load Balancer (SLB) instance, you must decide the listener type and network type of the SLB instance.

Select a region

When you select a region, take note of the following items:

- To reduce the network latency and increase the transmission speed, we recommend that you select the region that is closest to your customers.
- To provide more stable and reliable load balancing services, SLB supports primary/secondary zone deployment in most regions. You can select more than one zone to implement zone-disaster recovery. We recommend that you select a region that supports primary/secondary zone deployment.
- SLB does not support cross-region deployment. Therefore, you must deploy the SLB instance and backend Elastic Compute Service (ECS) instances in the same region.

Select a network type (Internet-facing or internal-facing)

SLB provides load balancing services for both Internet-facing and internal applications:

- If you want to use SLB to distribute requests from the Internet, create an Internet-facing SLB instance.
  An Internet-facing SLB instance is assigned a public IP address to receive requests from the Internet.
- If you want to use SLB to distribute requests from the internal network, create an internal-facing SLB instance.
  An internal-facing SLB instance is assigned a private IP address and is accessible only within the internal network.

Select an instance type

Shared-resource SLB instances share resources with each other. The performance of shared-resource SLB instances is not guaranteed.

Apsara Stack also released high-performance SLB instances. You can choose high-performance SLB instances that use exclusive resources if you require higher service availability. The following table describes the supported SLB instance types.

Instance type	Specification	Maximum number of connections	CPS	QPS	Purchase method
Type 1	Small I (slb.s1.small)	5,000	3,000	1,000	Available for purchase from the official website of Apsara Stack.
Type 2	Standard I (slb.s2.small)	50,000	5,000	5,000	Available for purchase from the official website of Apsara Stack.
Type 3	Standard II (slb.s2.medium)	100,000	10,000	10,000	Available for purchase from the official website of Apsara Stack.
Type 4	Higher I (slb.s3.small)	200,000	20,000	20,000	Available for purchase from the official website of Apsara Stack.

Select a listener protocol

SLB supports Layer 4 (TCP and UDP) and Layer 7 (HTTP and HTTPS) load balancing:

- A Layer 4 listener distributes requests to backend servers without modifying packet headers. After a client request reaches a Layer 4 listener, SLB establishes a TCP connection to the ECS instance port specified in the listener configuration.
- A Layer 7 listener functions as a reverse proxy. After a client request reaches a Layer 7 listener, SLB establishes a new TCP connection over HTTP with a backend server, instead of directly forwarding the request to the backend ECS instance.
  Compared with Layer 4 listeners, Layer 7 listeners require an additional step of Tengine processing. Therefore, Layer 4 listeners provide higher performance than Layer 7 listeners. In addition, the performance of Layer 7 listeners may be affected by factors such as insufficient client ports or excessive backend server connections. We recommend that you use Layer 4 listeners if you require high performance.

Add backend servers

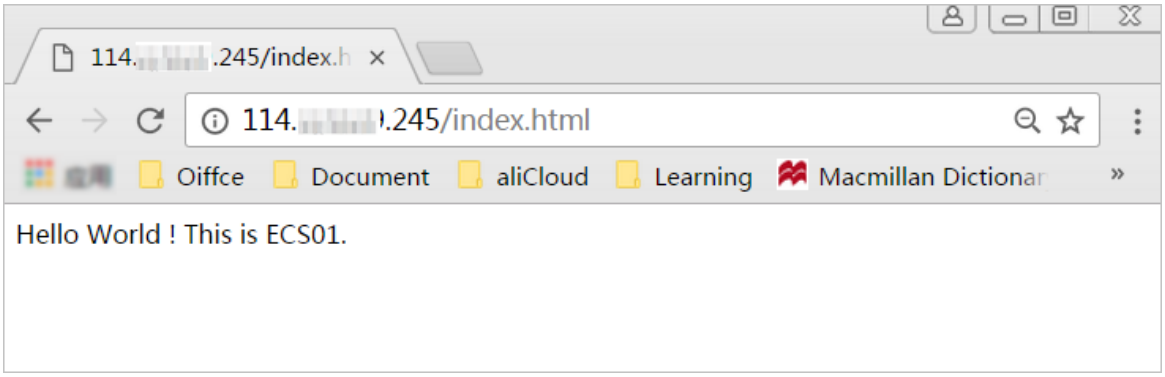
Before you use the SLB service, you must create ECS instances, deploy applications on the ECS instances, and then associate the ECS instances with your SLB instance to process client requests.

When you create and configure an ECS instance, take note of the following issues:

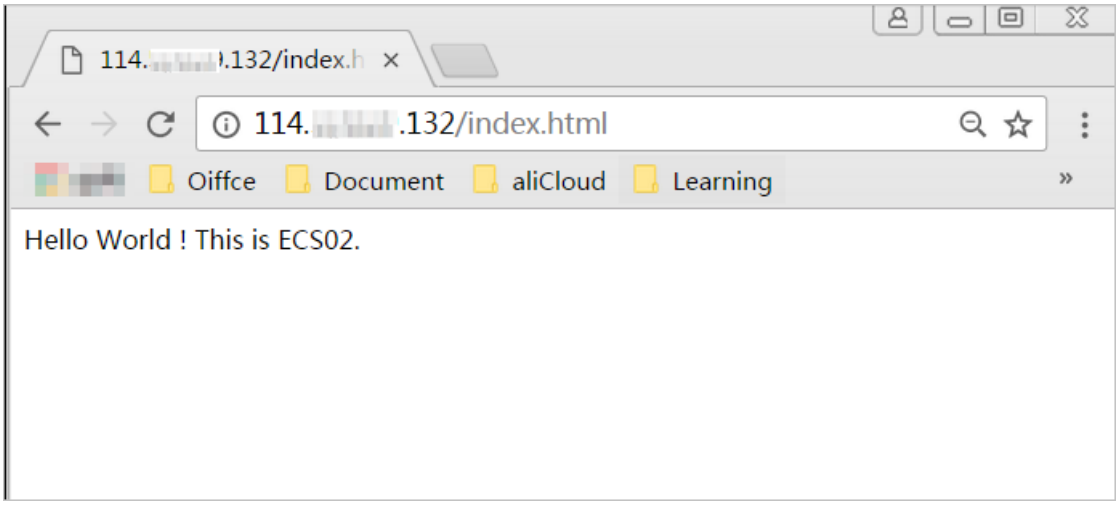
- Region and zone of the ECS instance

Make sure that the ECS instance is deployed in the same region as the SLB instance.
In this example, two ECS instances named ECS01 and ECS02 are created in Region 1.

- Configurations
  In this example, two static web pages are hosted on ECS01 and ECS02 by using Apache.
  - Enter the elastic IP address (EIP) associated with ECS01 in the address bar of your browser.



  - Enter the EIP associated with ECS02 in the address bar of your browser.



No additional configurations are required after you deploy the applications on the ECS instances. However, if you want to use a Layer 4 (TCP or UDP) listener and the ECS instances run Linux, make sure that the following parameters in the *net.ipv4.conf* file located in */etc/sysctl.conf* are set to 0:

```
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.rp_filter = 0
net.ipv4.conf.eth0.rp_filter = 0
```

3.3. Create an SLB instance

This topic describes how to create a Server Load Balancer (SLB) instance. An SLB instance is a running entity of the SLB service. You can add multiple listeners and backend servers to an SLB instance.

Prerequisites

- Elastic Compute Service (ECS) instances are created and applications are deployed on the ECS instances.
- The ECS instances and the SLB instance belong to the same organization. In addition, the security group rules configured for the ECS instances allow HTTP requests from port 80 and HTTPS requests

from port 443 (HTTPS).

Procedure

1. Log on to the SLB console.
2. In the left-side navigation pane, choose **Instances > Instances**.
3. On the **Instances** page, click **Create Instance**.
4. Configure the SLB instance and click **Submit**.

Parameter	Description
Organization	Select the organization to which the SLB instance belongs. **Note** Make sure that the SLB instance and its backend servers belong to the same organization.
Resource Set	Select the resource group to which the SLB instance belongs.
Region	Select the resource group to which the SLB instance belongs.
Zone	Select the zone where you want to deploy the SLB instance.
Quantity	Select the number of SLB instances that you want to purchase.
Instance Name	Enter a name for the SLB instance. If you set Quantity to a value greater than 1, the system automatically assigns a name to the SLB instance.
Instance Edition	Select one of the following options: ○ **Shared-resource**: Shared-resource SLB instances share resources with each other. The performance of shared-resource SLB instances is not guaranteed. ○ **High-performance**: High-performance SLB instances use exclusive resources. The performance of high-performance SLB instances depends on the specification.
Instance Type	Select the type of network traffic that you want to distribute. Valid values: Internal Network and Internet. Internal Network is selected in this example.
Network Type	Select the network type of the SLB instance. Valid values: Classic Network and VPC. VPC is selected in this example.
IP Version	Select an IP version.
VPC	Select a virtual private cloud (VPC).
vSwitch	Select a vSwitch.
Billing Method	Set the billing method.

What's next

Configure an SLB instance

3.4. Configure an SLB instance

This topic describes how to configure a Server Load Balancer (SLB) instance. Before an SLB instance can forward traffic, you must add at least one listener and one group of backend servers to the SLB instance. The following example shows how to add a TCP listener and Elastic Compute Service (ECS) instances to an SLB instance. The ECS instances are ECS 01 and ECS 02. These ECS instances function as backend servers that host static web pages.

Procedure

1. Log on to the SLB console.
2. On the **Instances** page, find the SLB instance that you want to manage and click **Configure Listener** in the Actions column.
3. On the **Protocol and Listener** wizard page, set the following parameters to configure the listener. Use the default settings for other parameters. Click **Next**.
   - **Select Listener Protocol**: Select a listener protocol. **TCP** is selected in this example.
   - **Listening Port**: Specify a frontend port to receive and distribute requests to backend servers. In this example, the port is set to **80**.
     The SLB instance uses this port to provide external services. In most cases, port 80 is set for HTTP listeners and port 443 is set for HTTPS listeners.

   Advanced:

   - **Enable Peak Bandwidth Limit**: Applications that run on backend ECS instances provide external services. You can set bandwidth caps to limit service capabilities of applications.
   - **Scheduling Algorithm**: SLB supports the following scheduling algorithms. **Round-Robin (RR)** is selected in this example.
     - **Weighted Round-Robin (WRR)**: Requests are distributed to backend servers in sequence. Backend servers with higher weights receive more requests.
     - **Round-Robin (RR)**: Requests are distributed to backend servers in sequence.
     - **Consistent Hash (CH)**: Only high-performance SLB instances support the CH algorithm.
       - **Source IP**: specifies consistent hashing that is based on source IP addresses. Requests from the same source IP address are distributed to the same backend server.
       - **Tuple**: specifies consistent hashing that is based on four factors: source IP address, destination IP address, source port, and destination port. Requests that contain the same information based on the four factors are distributed to the same backend server.
4. On the **Backend Servers** wizard page, select **Default Server Group** and click **Add More** to add backend servers.
   i. In the **My Servers** panel, select ECS 01 and ECS 02 and click **Next**.
   ii. A backend server with a higher weight receives more requests. The default value is 100. We recommend that you keep the default value.
   iii. Click **Add**.

iv. On the **Default Server Group** tab, specify backend ports that are available to receive requests. The ports are used by backend ECS instances to receive requests. You can specify the same port for multiple backend servers that are added to the same SLB instance. In this example, the port is set to 80.

5. Click **Next** to configure health checks. In this example, the default health check settings are used.

   After you enable health checks for the SLB instance, the SLB instance periodically checks whether the backend ECS instances are healthy. When the SLB instance detects an unhealthy ECS instance, the SLB instance distributes requests to other healthy ECS instances. When the unhealthy ECS instance recovers, the SLB instance starts to distribute requests to the ECS instance again.

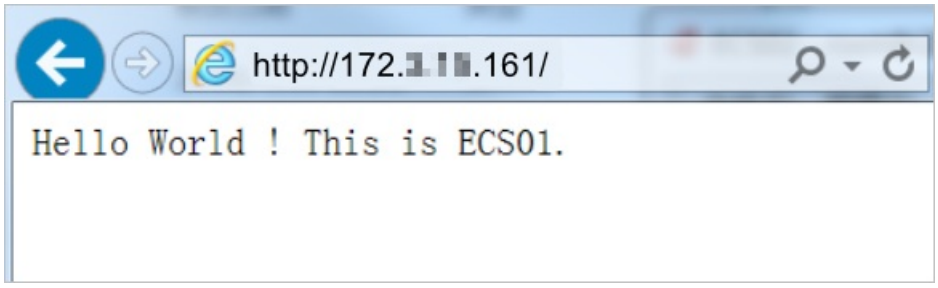
6. Click **Next**. On the **Confirm** wizard page, check the configurations and click **Submit**.

7. Click **OK** to return to the **Instances** page. Then, click ↻ to refresh the page.

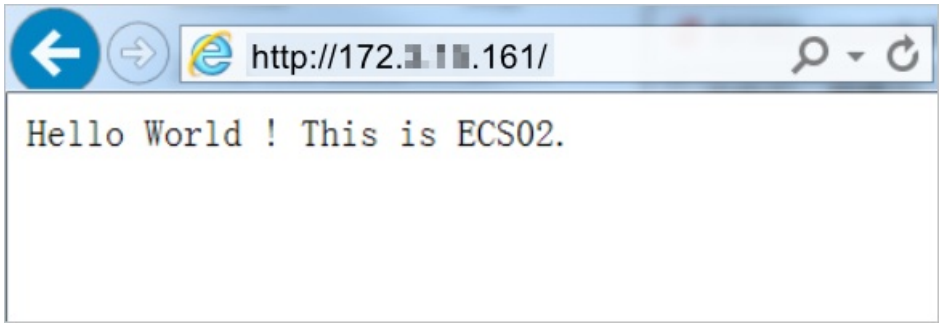
   If the health check states of the backend ECS instances are **Normal**, the backend ECS instances run as normal and can process requests distributed from the SLB instance.

8. Enter the IP address of the SLB instance into the address bar of a browser to test load balancing services of the SLB instance.

   ECS01

   ECS02

3.5. Release an SLB instance

You can release SLB instances that are no longer needed to reduce costs. The backend Elastic Compute Service (ECS) instances are not deleted or affected after you release an SLB instance.

Procedure

1. Log on to the SLB console.
2. On the **Instances** page, find the SLB instance that you want to release and choose ⋮ > **Release** in the Actions column, or select the SLB instance and click **Release** at the lower part of the page.
3. In the **Release** panel, select **Release Now**.

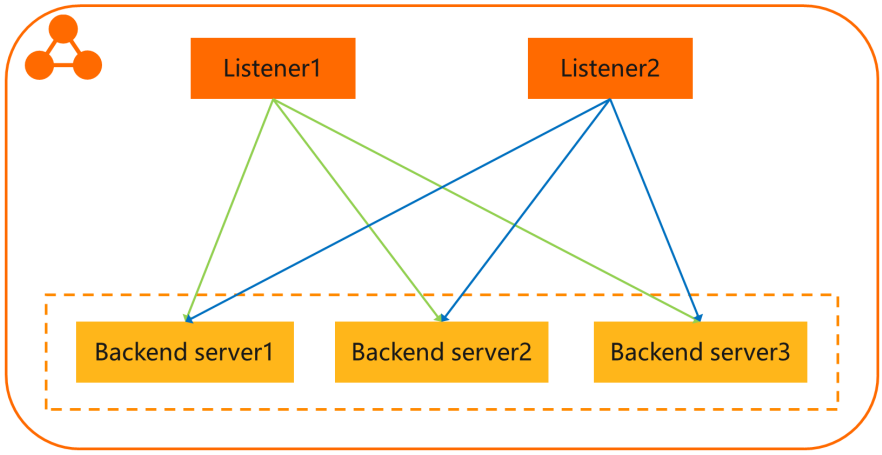
> **Note** The system releases the SLB instances at 30-minute and hour marks. However, the system stops billing the SLB instance at the specified release time.

4. Click **Next**.
5. Click **OK** to release the SLB instance.

4.SLB instances

4.1. Overview

This topic provides an overview of Server Load Balancer (SLB) instances. An SLB instance is a running entity of the SLB service. To use the SLB service, you must create an SLB instance and add listeners and backend servers to the instance.



Instance types

Alibaba Cloud provides Internet-facing and internal-facing SLB instances.

When you create an Internet-facing SLB instance, the system allocates a public IP address to the instance. You can associate a domain name with the public IP address. This way, the SLB instance can receive requests from the Internet and distribute requests to backend servers based on the forwarding rules that you have configured for the listener.

Features of Internet-facing SLB instances when they serve Internet-facing applications:

- You cannot disassociate the public IP address allocated by the system from an Internet-facing SLB instance.
- Subscription Internet-facing SLB instances support only the pay-by-bandwidth metering method. Pay-as-you-go Internet-facing SLB instances support the pay-by-data-transfer and pay-by-bandwidth metering methods.

Internet-facing SLB instance

Internal-facing SLB instances distribute client requests within the same virtual private cloud (VPC) to the private IP addresses of backend servers based on the specified load balancing policies.

Internal-facing SLB instances that are associated with elastic IP addresses (EIPs) can process requests from the Internet. Features of Internal-facing SLB instances when they serve Internet-facing applications:

- Public IP addresses are allocated to EIPs. You can disassociate EIPs from internal-facing SLB instances based on your requirements.
- An EIP that is associated with an EIP bandwidth plan supports the pay-by-95th-percentile-bandwidth billing method in addition to the subscription and pay-as-you-go billing methods.

The network types supported by an internal-facing SLB instance vary based on the billing method of the internal-facing SLB instance.

- Subscription internal-facing SLB instances support classic network and VPC.
  - VPC
    If you choose VPC for an internal-facing SLB instance, the IP address of the SLB instance is allocated from the CIDR block of the vSwitch that is attached to the VPC. This internal-facing SLB instance can be accessed only by Elastic Compute Service (ECS) instances in the VPC.
  - Classic network
    If you choose classic network for an internal-facing SLB instance, the IP address of the SLB instance is allocated and maintained by Alibaba Cloud. This internal-facing SLB instance can be accessed only by ECS instances in the classic network.

    Notice Internal-facing SLB instances of the classic network type are no longer available for purchase.
- Pay-as-you-go internal-facing SLB instances support only VPC.
  In addition to basic features, these SLB instances also support PrivateLink. Internal-facing SLB instances can receive requests from other VPCs through PrivateLink connections and distribute the requests to backend servers based on the specified load balancing policies. For more information, see What is PrivateLink?.

Internet-facing SLB instance

Internal-facing SLB instance

Instance types and specifications

SLB provides high-performance SLB instances and shared-resource SLB instances

- High-performance SLB instances
  The following section describes three key metrics of high-performance SLB instances:
  - Maximum number of connections
    The maximum number of concurrent connections that a CLB instance supports. When the number of existing concurrent connections reaches the upper limit, new connection requests are dropped.
  - Connections per second (CPS)
    The number of new connections that are established per second. When the CPS value reaches the upper limit, new connection requests are dropped.
  - Queries per second (QPS)
    The number of HTTP or HTTPS queries (requests) that can be processed per second. This metric is specific to Layer 7 listeners. When the QPS value reaches the upper limit, new connection requests are dropped.

  Alibaba Cloud provides the following specifications for high-performance SLB instances.

  Note If you require a higher QPS specification, you can purchase ALB instances. For more information, see ALB.

Type	Specification	Maximum number of connections	CPS	QPS	Purchase method
Type 1	Small I (slb.s1.small)	5,000	3,000	1,000	Available for purchase from the official website of Apsara Stack.
Type 2	Standard I (slb.s2.small)	50,000	5,000	5,000	Available for purchase from the official website of Apsara Stack.
Type 3	Standard II (slb.s2.medium)	100,000	10,000	10,000	Available for purchase from the official website of Apsara Stack.
Type 4	Higher I (slb.s3.small)	200,000	20,000	20,000	Available for purchase from the official website of Apsara Stack.

- Shared-resource SLB instances
  Shared-resource SLB instances share resources with each other. The performance of shared-resource SLB instances is not guaranteed.

  > **Notice** Shared-performance SLB instances are no longer available for purchase.

The following table describes the differences between shared-resource SLB instances and high-performance SLB instances.

Feature	High-performance SLB instance	Shared-resource SLB instance
Resource allocation	Exclusive resources	Shared resources
Service uptime guaranteed by the service-level agreement (SLA)	99.95%	Not supported
IPv6	√	-
Server Name Indication (SNI)	√	-
Blacklists and whitelists	√	-

Feature	High-performance SLB instance	Shared-resource SLB instance
Elastic network interface (ENI) mounting	√	-
Assigning secondary IP addresses to ENIs that are bound to ECS instances	√	-
HTTP-to-HTTPS redirection	√	-
Consistent hashing	√	-
TLS security policies	√	-
HTTP2	√	-
WebSocket or WebSocket Secure	√	-

Note In the preceding table, "√" indicates that the feature is supported, and "-" indicates that the feature is not supported.

4.2. Create an SLB instance

This topic describes how to create a Server Load Balancer (SLB) instance. An SLB instance is a running entity of the SLB service. You can add multiple listeners and backend servers to an SLB instance.

Prerequisites

- Elastic Compute Service (ECS) instances are created and applications are deployed on the ECS instances.
- The ECS instances and the SLB instance belong to the same organization. In addition, the security group rules configured for the ECS instances accept HTTP requests received on port 80 and HTTPS requests received on port 443.

Procedure

1. Log on to the SLB console.
2. In the left-side navigation pane, choose **Instances > Instances**.
3. On the **Instances** page, click **Create Instance**.
   - **Organization**: Select an organization for the SLB instance from the drop-down list.
   - **Resource Set**: Select a resource set for the SLB instance from the drop-down list.
   - **Region**: Select the region where you want to deploy the SLB instance.
   - **Zone**: Select a zone for the SLB instance from the drop-down list.
   - **Instance Name**: Enter a name for the SLB instance in the Instance Name field.
     The name must be 2 to 128 characters in length, and can contain letters, digits, hyphens (-), colons (:), periods (.), and underscores (_). Line breaks and spaces are supported. It must start with a letter and cannot start with `http://` or `https://`.
   - **Instance Edition**: Select an instance type: shared-resource or high-performance. Shared-

resource SLB instances share resources with each other. The performance of shared-resource SLB instances is not guaranteed. The performance of high-performance SLB instances varies by specification.

- **Instance Type**: Select the type of network traffic that you want to distribute. Valid values: Internal Network and Internet.
- **Network Type**: Select the network type of the SLB instance. Valid values: Classic Network and VPC.
- **IP Version**: Select an IP version.
- **IP Address**: Enter a service IP address for the SLB instance. Make sure that the service IP address is not in use. Otherwise, the SLB instance cannot be created. If you do not set this parameter, the system automatically allocates an IP address to the SLB instance.

> **Note** When you configure an internal-facing SLB instance, the service IP address of the SLB instance must belong to the CIDR block of the vSwitch.

4. Click **Submit**.

4.3. Start and stop an SLB instance

This topic describes how to start or stop a Server Load Balancer (SLB) instance. You can start or stop SLB instances at any time. A stopped SLB instance does not receive or forward client requests.

Procedure

1. Log on to the SLB console
2. In the left-side navigation pane, choose **Instances > Instances**.
3. Find the SLB instance that you want to manage and choose > **Start** or > **Stop** in the **Actions** column.
4. To start or stop multiple SLB instances at a time, select the SLB instances and click **Start** or **Stop** at the lower part of the page.

4.4. Release an SLB instance

This topic describes how to release a Server Load Balancer (SLB) instance. You can release SLB instances based on your needs.

Procedure

1. Log on to the SLB console.
2. In the left-side navigation pane, choose **Instances > Instances**.
3. Find the SLB instance that you want to release and choose > **Release** in the Actions column.
4. In the **Release** panel, click **Release Now**.
5. Click **Next**.
6. Click **OK**. Enter the verification code that you obtained to release the SLB instance.

5.Listeners

5.1. Listener overview

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

SLB provides TCP and UDP listeners for Layer 4 load balancing, and HTTP and HTTPS listeners for Layer 7 load balancing. The following table describes the features and use scenarios of these listeners.

Protocol	Description	Use scenario
TCP	• A connection-oriented protocol that requires a logical connection to be established before data can be transmitted. • Session persistence is based on source IP addresses. • Source IP addresses are visible at the network layer. • Data is transmitted at a fast rate.	• Applicable to scenarios that require high reliability and data accuracy but can withstand a low transmission speed. These scenarios include file transmission, email sending and receiving, and remote logons. • Web applications that do not have custom requirements.
UDP	• A connectionless protocol. UDP transmits data packets directly instead of making a three-way handshake with the other party before UDP sends data. UDP does not provide error recovery or data re-transmission. • Fast data transmission but relatively low reliability.	Applicable to scenarios where real-time transmission outweighs reliability, such as video conferencing and real-time quote services.
HTTP	• An application-layer protocol that is used to package data. • Cookie-based session persistence. • Retrieve client IP addresses by using the X-Forward-For header.	Intended for applications that identify data content, such as web applications and mobile games.
HTTPS	• Encrypted data transmission that prevents unauthorized access. • Centralized certificate management service. You can upload certificates to SLB. Then, data decryption is offloaded from backend servers to SLB.	Intended for applications that require encrypted data transmission.

5.2. Add a TCP listener

This topic describes how to add a TCP listener to a Server Load Balancer (SLB) instance. TCP provides reliable and accurate data delivery at relatively low connection speeds. Therefore, TCP applies to file transmission, email sending or receiving, and remote logons. You can add a TCP listener to forward TCP requests.

Step 1: Open the listener configuration wizard

To open the listener configuration wizard, perform the following operations:

1. Log on to the SLB console.
2. In the left-side navigation pane, choose **Instances > Instances**.
3. Use one of the following methods to open the listener configuration wizard:
   - On the **Instances** page, find the SLB instance and click **Configure Listener** in the **Actions** column.
   - On the **Instances** page, click the ID of the SLB instance. On the **Listener** tab, click **Add Listener**.

Step 2: Configure the TCP listener

To configure the TCP listener, perform the following operations:

1. Set the following parameters and click **Next**.

Parameter	Description
Select Listener Protocol	Select the protocol of the listener. In this example, **TCP** is selected.
Listening Port	Specify the listening port that is used to receive requests and forward them to backend servers. Valid values: 1 to 65535.
Advanced	

Parameter	Description
Scheduling Algorithm	SLB supports the following scheduling algorithms: ○ **Weighted Round-Robin (WRR)**: Backend servers that have higher weights receive more requests than backend servers that have lower weights. ○ **Round-Robin (RR)**: Requests are distributed to backend servers in sequence. ○ **Consistent Hash (CH)**: ■ **Source IP**: specifies consistent hashing that is based on source IP addresses. Requests from the same source IP address are distributed to the same backend server. ■ **Tuple**: specifies consistent hashing that is based on four factors: source IP address, destination IP address, source port, and destination port. Requests that contain the same information based on the four factors are distributed to the same backend server. **Note** Only high-performance SLB instances support the CH algorithm.
Enable Session Persistence	Specify whether to enable session persistence. After session persistence is enabled, SLB forwards all requests from a client to the same backend server. SLB persists TCP sessions based on IP addresses. Requests from the same IP address are forwarded to the same backend server.
Enable Peak Bandwidth Limit	Specify whether to set the bandwidth limit of the listener. If an SLB instance is billed based on bandwidth usage, you can specify different bandwidth limits for different listeners. This limits the amount of network traffic that flows through each listener. The sum of the bandwidth limits of all listeners that are added to an SLB instance cannot exceed the bandwidth limit of the SLB instance. By default, this feature is disabled and all listeners share the bandwidth of the SLB instance.
Idle Timeout	Specify the timeout of idle TCP connections. Unit: seconds. Valid values: 10 to 900.
Obtain Client Source IP Address	Backend servers associated with Layer 4 listeners can retrieve client IP addresses without additional configurations.
Automatically Enable Listener After Creation	Specify whether to immediately enable the listener after it is created. By default, listeners are enabled after they are created.

Step 3: Add backend servers

After you configure the listener, you must add backend servers to process client requests. You can use the default server group that is configured for the SLB instance, or create a vServer group or a primary/secondary server group.

1. On the **Backend Servers** wizard page, select **Default Server Group** and click **Add More**.

2. In the **My Servers** panel, select the Elastic Compute Service (ECS) instances that you want to add as backend servers and click **Next**.
3. On the **Configure Ports and Weights** wizard page, specify the weights of the backend servers that you want to add. A backend server with a higher weight receives more requests.

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

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

   You can specify the same port on different backend servers that are added to an SLB instance.
5. Click **Next**.

Step 4: Configure health checks

SLB performs health checks to check the availability of the ECS instances that serve as backend servers. The health check feature improves the overall availability of frontend services and prevents service interruptions caused by backend server anomalies. Click **Modify** to configure advanced health check settings and click Next. For more information, see Health check overview.

Step 5: Submit the configurations

To submit the configurations, perform the following operations:

1. On the **Confirm** wizard page, check the configurations. You can click **Modify** to modify the configurations.
2. Click **Submit**.
3. In the **Configuration Successful** message, click **OK**.

   You can check the created listener on the Listener tab.

5.3. Add a UDP listener

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

Context

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

- You are not allowed to specify ports 250, 4789, or 4790 for UDP listeners. They are system reserved ports.
- Fragmentation is not supported.
- If you add a UDP listener to an SLB instance deployed in a classic network, the UDP listener cannot pass client IP addresses to backend servers.
- The following operations take effect 5 minutes after they are performed on a UDP listener:
  - Remove backend servers.
  - Set the weight of a backend server to 0 after it is detected unhealthy.

Step 2: Configure the UDP listener

To configure the listener, perform the following operations:

1. On the **Protocol and Listener** wizard page, set the following parameters and click **Next**.

Parameter	Description
Select Listener Protocol	Select the protocol of the listener. In this example, **UDP** is selected.
Listening Port	Set the listening port that is used to receive requests and forward them to backend servers. Valid values: 1 to 65535.
Advanced	
Scheduling Algorithm	SLB supports the following scheduling algorithms: ○ **Weighted Round-Robin (WRR)**: Backend servers that have higher weights receive more requests than those that have lower weights. ○ **Round-Robin (RR)**: Requests are distributed to backend servers in sequence. ○ **Consistent Hash (CH)**: ■ **Source IP**: specifies consistent hashing that is based on source IP addresses. Requests from the same source IP address are distributed to the same backend server. ■ **Tuple**: specifies consistent hashing that is based on four factors: the source IP address, destination IP address, source port, and destination port. Requests that contain the same information based on the four factors are distributed to the same backend server. ■ **QUIC ID**: specifies consistent hashing that is based on Quick UDP Internet Connections (QUIC) IDs. Requests that contain the same QUIC ID are distributed to the same backend server. **Notice** QUIC is implemented based on draft-ietf-quic-transport-10 and iterates rapidly. Therefore, compatibility is not guaranteed for all QUIC versions. We recommend that you perform tests before you apply the protocol to a production environment.
Enable Session Persistence	Specify whether to enable session persistence. SLB maintains the persistence of UDP sessions by using consistent hashing that is based on source IP addresses.
Enable Peak Bandwidth Limit	Specify whether to set a bandwidth limit for the listener. If an SLB instance is billed based on bandwidth usage, you can specify different bandwidth limits for different listeners. This limits the amount of network traffic that flows through each listener. The sum of the bandwidth limits of all listeners that are added to an SLB instance cannot exceed the bandwidth limit of the SLB instance. By default, this feature is disabled and all listeners share the bandwidth of the SLB instance.

Parameter	Description
Obtain Client Source IP Address	Backend servers associated with UDP listeners can obtain client IP addresses without additional configurations. **Note** If you add a UDP listener to an SLB instance deployed in a classic network, the UDP listener cannot pass client IP addresses to backend servers.
Automatically Enable Listener After Creation	Specify whether to enable the listener after it is created. By default, listeners are automatically enabled after they are created.

Step 3: Add backend servers

After you configure the listener, you must add backend servers to process client requests. You can use the default server group that is configured for the SLB instance, or create a vServer group or a primary/secondary server group.

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

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

   An ECS instance with a higher weight receives more requests.

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

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

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

5. Click **Next**.

Step 3: Configure health checks

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

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

Step 4: Submit the configurations

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

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

5.4. Add an HTTP listener

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

Step 1: Configure an HTTP listener

1. Log on to the SLB console.
2. Use one of the following methods to open the listener configuration wizard:
   - On the **Instances** page, find the SLB instance that you want to manage and click **Configure Listener** in the **Actions** column.
   - On the **Instances** page, click the ID of the SLB instance that you want to manage. On the **Listener** tab, click **Add Listener**.
3. Set the following parameters to configure the listener.

Parameter	Description
Select Listener Protocol	Select the protocol of the listener. In this example, **HTTP** is selected.
Listening Port	Set the listening port that is used to receive requests and forward them to backend servers. Valid values: 1 to 65535.
Advanced	Click **Modify** to configure advanced settings.
Scheduling Algorithm	Select a scheduling algorithm. ○ **Weighted Round-Robin (WRR)**: Backend servers that have higher weights receive more requests than backend servers that have lower weights. ○ **Round-Robin (RR)**: Requests are distributed to backend servers in sequence.
Redirection	Specify whether to redirect traffic from the HTTP listener to an HTTPS listener. **Note** Before you enable redirection, make sure that an HTTPS listener is created.

Parameter	Description
Enable Session Persistence	Specify whether to enable session persistence. After session persistence is enabled, SLB forwards all requests from a client to the same backend server. SLB persists HTTP sessions based on cookies. SLB allows you to use the following methods to process cookies: ◦ **Insert cookie**: If you select this option, you need only to specify the timeout period of the cookie. SLB inserts a cookie (SERVERID) into the first HTTP or HTTPS response that is sent to a client. The next request from the client will contain this cookie, and the listener will forward this request to the recorded backend server. ◦ **Rewrite cookie**: If you select this option, you can specify the cookie that you want to insert into an HTTP or HTTPS response. You must specify the timeout period and the lifetime of a cookie on a backend server. After you specify a cookie, SLB overwrites the original cookie with the specified cookie. The next time SLB receives a client request that carries the specified cookie, the listener distributes the request to the recorded backend server.
Enable Peak Bandwidth Limit	Specify whether to set a bandwidth limit for the listener. Unit: Mbit/s. Valid values: 0 to 5120. If an SLB instance is billed based on bandwidth usage, you can set different bandwidth limits for different listeners. This limits the amount of traffic that flows through each listener. The sum of the bandwidth limit values of all listeners that are added to an SLB instance cannot exceed the bandwidth limit of this SLB instance. By default, this feature is disabled and all listeners share the bandwidth of the SLB instance.
Idle Timeout	Specify the timeout period of idle connections. Unit: seconds. Valid values: 1 to 60. If no request is received within the specified timeout period, SLB closes the connection. SLB recreates the connection when a new connection request is received.
Request Timeout	Specify the request timeout period. Unit: seconds. Valid values: 1 to 180. If no response is received from the backend server within the request timeout period, SLB returns an HTTP 504 error code to the client.
Enable Gzip Compression	Specify whether to enable Gzip compression to compress specific types of files. Gzip supports the following file types: text/xml, text/plain, text/css, application/javascript, application/x-javascript, application/rss+xml, application/atom+xml, and application/xml.

Parameter	Description
Add HTTP Header Fields	You can add the following HTTP headers: ○ Use the `X-Forwarded-For` header to retrieve client IP addresses. ○ Use the `SLB-ID` header to retrieve the ID of the SLB instance. ○ Use the `SLB-IP` header to retrieve the public IP address of the SLB instance. ○ Use the `X-Forwarded-Proto` header to retrieve the listener protocol used by the SLB instance.
Obtain Client Source IP Address	Specify whether to retrieve the client IP address. By default, this feature is enabled.
Automatically Enable Listener After Creation	Specify whether to enable the listener after it is created. By default, listeners are automatically enabled after they are created.

4. Click **Next**.

Step 2: Add backend servers

After you configure the listener, you must add backend servers to process client requests. You can use the default server group that is configured for the SLB instance. You can also create a vServer group or a primary/secondary server group.

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

1. On the **Backend Servers** wizard page, select **Default Server Group**. Then, click **Add More**.
2. In the **My Servers** panel, select the Elastic Compute Service (ECS) instances that you want to add as backend servers and click **Next**.
3. On the **Configure Ports and Weights** wizard page, specify the weights of the backend servers that you want to add. A backend server with a higher weight receives more requests.

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

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

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

5. Click **Next**.

Step 3: Configure health checks

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

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

Step 4: Submit the configurations

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

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

5.5. Add an HTTPS listener

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

Step 1: Configure an HTTPS listener

1. Log on to the SLB console
2. Use one of the following methods to open the listener configuration wizard:
   - On the **Instances** page, find the SLB instance that you want to manage and click **Configure Listener** in the Actions column.
   - On the **Instances** page, click the ID of the SLB instance that you want to manage. On the **Listener** tab, click **Add Listener**.
3. Set the following parameters and click **Next**.

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

Parameter	Description
Enable Session Persistence	Specify whether to enable session persistence. After session persistence is enabled, SLB forwards all requests from a client to the same backend server. SLB persists HTTP sessions based on cookies. SLB allows you to use the following methods to process cookies: ◦ **Insert cookie**: If you select this option, you only need to specify the timeout period of the cookie. SLB inserts a cookie (SERVERID) into the first HTTP or HTTPS response that is sent to a client. The next request from the client will contain this cookie, and the listener will forward this request to the recorded backend server. ◦ **Rewrite cookie**: If you select this option, you can specify the cookie that you want to insert into an HTTP or HTTPS response. You must specify the timeout period and the lifetime of a cookie on a backend server. After you specify a cookie, SLB overwrites the original cookie with the specified cookie. The next time SLB receives a client request that carries the specified cookie, the listener distributes the request to the recorded backend server.
Enable HTTP/2	Select whether to enable HTTP/2.0 for the frontend protocol of the SLB instance.
Enable Peak Bandwidth Limit	Specify whether to set the bandwidth limit of the listener. If an SLB instance is billed based on bandwidth usage, you can specify different bandwidth limits for different listeners. This limits the amount of network traffic that flows through each listener. The sum of the bandwidth limits of all listeners that are added to an SLB instance cannot exceed the bandwidth limit of the SLB instance. By default, this feature is disabled and all listeners share the bandwidth of the SLB instance.
Idle Timeout	Specify the timeout period of idle connections. Unit: seconds. Valid values: 1 to 60. If no request is received within the specified timeout period, SLB closes the connection. SLB recreates the connection when a new connection request is received.
Request Timeout	Specify the request timeout period. Unit: seconds. Valid values: 1 to 180. If no response is received from the backend server within the request timeout period, SLB returns an HTTP 504 error code to the client.
Enable Gzip Compression	Specify whether to enable Gzip compression to compress specific types of files. Gzip supports the following file types: text/xml, text/plain, text/css, application/javascript, application/x-javascript, application/rss+xml, application/atom+xml, and application/xml.

Parameter	Description
Add HTTP Header Fields	You can add the following HTTP headers: ○ Use the `X-Forwarded-For` header to retrieve client IP addresses. ○ Use the `SLB-ID` header to retrieve the ID of the SLB instance. ○ Use the `SLB-IP` header to retrieve the public IP address of the SLB instance. ○ Use the `X-Forwarded-Proto` header to retrieve the listener protocols used by the SLB instance.
Obtain Client Source IP Address	Specify whether to retrieve the client IP address. By default, this feature is enabled.
Automatically Enable Listener After Creation	Specify whether to immediately enable the listener after it is created. By default, listeners are enabled after they are created.

Step 2: Configure an SSL certificate

To add an HTTPS listener, you must upload a server certificate or CA certificate. The following table describes the two types of certificates.

Certificate	Description	Required for one-way authentication	Required for mutual authentication
Server certificate	A server certificate is used to authenticate the identity of a server. A browser authenticates the identity of a server by checking whether the certificate sent by the server is issued by a trusted certification authority (CA).	Yes You must upload the server certificate to the certificate management system of SLB.	Yes You must upload the server certificate to the certificate management system of SLB.
Client certificate	A client certificate is used to authenticate the identity of a client. A server authenticates the identity of a client by verifying the certificate sent by the client. You can sign a client certificate with a self-signed CA certificate.	No	Yes You must install the client certificate on the client.
CA certificate	A CA certificate is used by a server to verify the signature on a client certificate. If the signature is invalid, the connection request is denied.	No	Yes You must upload the CA certificate to the certificate management system of SLB.

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

- SLB supports the following public key algorithms: RSA 1024, RSA 2048, RSA 4096, ECDSA P-256, ECDSA P-384, and ECDSA P-521.
- The certificate that you want to upload must be in the PEM format.

- After you upload a certificate to SLB, SLB can manage the certificate. You do not need to bind the certificate to backend servers.
- It may take a few minutes to upload, load, and verify the certificate. Therefore, an HTTPS listener does not take effect immediately after it is created. It requires about 1 to 3 minutes to enable an HTTPS listener.
- The ECDHE cipher suite used by HTTPS listeners supports forward secrecy. It does not support the security enhancement parameters that are required by the DHE cipher suite. Therefore, you cannot upload certificates (PEM files) that contain the `BEGIN DH PARAMETERS` field.
- HTTPS listeners do not support Server Name Indication (SNI). You can choose TCP listeners and configure SNI on backend servers.
- By default, the timeout period of session tickets for HTTPS listeners is 300 seconds.
- The actual amount of data transfer on an HTTPS listener is larger than the billed amount because a portion of data is used for handshaking.
- Therefore, the amount of data transfer greatly increases when a large number of connections are established.

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

Step 3: Add backend servers

After you configure the listener, you must add backend servers to process client requests. You can use the default server group that is configured for the SLB instance. You can also create a vServer group or a primary/secondary server group.

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

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

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

4. Click **Add**. Specify the ports for backend servers to receive requests. Valid values: 1 to 65535. Click **Next**.

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

Step 4: Configure health checks

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

Step 5: Submit the configurations

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

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

5.6. Configure forwarding rules

This topic describes how to configure forwarding rules for a Server Load Balancer (SLB) instance. You can configure domain name-based or URL-based forwarding rules for an SLB instance that uses Layer 7 listeners. Layer 7 listeners distribute requests destined for different domain names or URLs to different Elastic Compute Service (ECS) instances.

Context

You can add multiple forwarding rules to a listener. Each forwarding rule is associated with a unique server group. Each server group contains one or more ECS instances. For example, you can configure a listener to forward read requests to one server group and write requests to another server group. This allows you to balance the loads of the backend servers.

SLB forwards requests based on the following rules:

- If a request matches a domain name-based or URL-based forwarding rule of a listener, the listener forwards the request to the server group specified in the matching forwarding rule.
- If a request does not match a domain name-based or URL-based forwarding rule but the listener is associated with a server group, the listener forwards the request to the associated server group.
- If none of the preceding conditions are met, requests are forwarded to the ECS instances in the default server group of the SLB instance.

Procedure

1. Log on to the SLB console.
2. Click the ID of the SLB instance that you want to manage. On the **Listener** tab, find the listener that you want to manage.

   You can configure domain name-based or URL-based forwarding rules only for HTTP and HTTPS listeners.
3. Click **Set Forwarding Rule** in the **Actions** column.
4. Configure forwarding rules based on the following information:
   - Configure a domain name-based forwarding rule
     - When you configure a domain name-based forwarding rule, leave the URL field empty. You do not need to enter a forward slash (/) in this field. The domain name can contain only letters, digits, hyphens (-), and periods (.).

- Domain name-based forwarding rules support both exact matching and wildcard matching. For example, www.aliyun.com is an exact domain name, whereas *.aliyun.com and *.market.aliyun.com are wildcard domain names. When a request matches multiple domain name-based forwarding rules, an exact match prevails over wildcard matches, as described in the following table.

  Domain name matching rule

Type	Request URL	Domain name matching rule (√ indicates that the domain name is matched whereas x indicates that the domain name is not matched.)		
		www.aliyun.com	*.aliyun.com	*.market.aliyun.com
Exact matching	www.aliyun.com	√	x	x
Wildcard matching	market.aliyun.com	x	x	x
	info.market.aliyun.com	x	x	√

- Configure a URL-based forwarding rule
  - When you configure a URL-based forwarding rule, leave the Domain Name field empty.
  - The URL can contain only letters, digits, and the following characters: `-/%?#&`
  - The URL must start with a forward slash (/).

    > **Note** If you enter only a forward slash (/) in the URL field, the URL-based forwarding rule is invalid.

  - URL-based forwarding rules support string matching and adopt sequential matching. Examples: `/admin`, `/bbs_`, and `/ino_test`.
- Configure both domain name-based and URL-based forwarding rules
  You can configure both domain name-based and URL-based forwarding rules to forward traffic destined for different URLs of the same domain name. We recommend that you configure a default forwarding rule with the URL field left empty in case errors are returned when the URLs of requests are not matched.
  For example, the domain name of a website is `www.example.com`. You are required to forward requests destined for `www.example.com/index.html` to Server Group 1 and forward requests destined for other URLs of the domain name to Server Group 2. To meet the preceding requirements, you must configure two forwarding rules, as shown in the following figure. Otherwise, a 404 error code is returned when a request destined for the `www.example.com` domain name does not match a forwarding rule.

5. Click **Save**.

5.7. Enable access control

This topic describes how to enable access control for a listener. You can enable access control for each listener of a Server Load Balancer (SLB) instance. You can set whitelists for different listeners.

Procedure

1. Log on to the SLB console.
2. Click the ID of the SLB instance for which you want to enable access control.
3. Click the **Listener** tab, find the listener that you want to manage, and then choose ⋮ > **Set Access Control** in the **Actions** column.
4. Set the following parameters and click **OK**.

Parameter	Description
Enable Access Control	Enable access control.
Access Control Method	**Whitelist**: After you set a whitelist for a listener, the listener forwards only requests from IP addresses or CIDR blocks that are added to the whitelist. Your business may be adversely affected if the whitelist is not set properly. After the whitelist is set, only requests from IP addresses that are added to the whitelist are forwarded by the listener. If the whitelist does not contain IP addresses, the SLB listener forwards all requests.
Access Control List	Select a network access control list (ACL). IPv6 instances can be associated only with IPv6 network ACLs, and IPv4 instances can be associated only with IPv4 network ACLs. **Note** Separate multiple IP entries with commas (,). You can add at most 300 IP entries to each network ACL. IP entries must be unique within each network ACL.

5.8. Disable access control

This topic describes how to disable access control for a listener.

Procedure

1. Log on to the SLB console.
2. Click the ID of the Server Load Balancer (SLB) instance for which you want to disable access control.
3. Click the **Listener** tab next to the **Instance Details** tab.
4. Find the listener for which you want to disable access control and choose > **Set Access Control** in the Actions column.
5. In the **Access Control Settings** panel, disable access control and click **OK**.

6.Backend servers

6.1. Overview

Before you use a Server Load Balancer (SLB) instance, you must specify Elastic Compute Service (ECS) instances as the backend servers of the SLB instance to receive requests forwarded by the listeners of the SLB instance.

Introduction

SLB allows you to create a server group, add multiple ECS instances in the same region to the server group, and then set a virtual IP address for the server group. This improves the performance and availability of your application deployed on the ECS instances. You can also use vServer groups to manage backend servers. You can associate listeners with different server groups. Then, the SLB instance can use these listeners to distribute requests to different backend server ports.

> **Note** If you associate a listener with a vServer group, the listener distributes requests to the ECS instances in the vServer group instead of the ECS instances in the default server group.

Limits

You can add ECS instances to or remove ECS instances from an SLB instance anytime, or switch network traffic between different ECS instances. You must make sure that the health check feature is enabled and at least one ECS instance is healthy to avoid service interruptions from the preceding operations.

Before you add backend ECS instances to an SLB instance, take note of the following limits:

- SLB does not have limits on the types of operating systems that backend ECS instances use. However, the ECS instances attached to the same SLB instance must maintain the same application and application data. To facilitate management and maintenance, we recommend that you add ECS instances that run the same operating system to an SLB instance.
- You must configure a listener for each application deployed on backend ECS instances. Each SLB instance supports at most 50 listeners. Each listening port of an SLB instance specifies a port used by an application on a backend ECS instance to provide services.
- You can specify a weight for each ECS instance in a server group. An ECS instance with a higher weight receives more requests.
- If session persistence is enabled, requests may not be evenly distributed to backend servers. To resolve this issue, we recommend that you disable session persistence and check whether the problem persists.
  If requests are still not evenly distributed, troubleshoot the issue by using the following methods:
  i. Count the numbers of access log entries generated on the backend ECS instances within a specified time period.
  ii. Check whether the numbers of access log entries generated on the backend ECS instances conform to the SLB configurations.If session persistence is enabled, you must exclude access log entries that contain the same IP address. If backend ECS instances have different weights, you must check whether the numbers of access log entries generated on the ECS instances are also weighted.
- When an ECS instance performs hot migration, persistent connections to SLB may be closed. Make sure that your application is configured with the automatic reconnection mechanism.

Default server groups

A default server group contains ECS instances that are used to receive requests. If a listener is not associated with a vServer group or a primary/secondary server group, the listener forwards requests to the ECS instances in the default server group.

Before an SLB instance can process requests, you must add at least one backend server to the default server group to receive requests. For more information, see Add backend servers to the default server group.

vServer groups

To distribute requests to different backend servers, you can specify the backend servers in different vServer groups. To allow an SLB instance to distribute requests based on domain names and URLs, you can specify vServer groups in domain name-based forwarding rules and URL-based forwarding rules. For more information, see Add ECS instances to a vServer group.

Primary/secondary server groups

A primary/secondary server group contains only two ECS instances. One ECS instance serves as the primary server and the other ECS instance serves as the secondary server. SLB does not perform health checks on the secondary server in a primary/secondary server group. If the primary server is declared unhealthy, network traffic is automatically forwarded to the secondary server. When the primary server recovers, traffic is forwarded to the primary server again. For more information, see Add ECS instances to a primary/secondary server group.

> **Note** You can add primary/secondary server groups only to TCP and UDP listeners.

6.2. Default server groups

6.2.1. Add backend servers to the default server group

Before a Server Load Balancer (SLB) instance can process requests, you must add at least one backend server to the default server group to receive requests.

Prerequisites

Before you add an Elastic Compute Service (ECS) instance to the default server group, make sure that the following requirements are met:

- An SLB instance is created. For more information, see Create an SLB instance.
- ECS instances are created and applications are deployed on the ECS instances to receive requests.

Procedure

1. Log on to the SLB console.
2. Find the SLB instance that you want to manage and click the instance ID.
3. Click the **Default Server Group** tab.
4. Click **Add**.
5. In the **Servers** panel, select one or more ECS instances that you want to add to the default server group in the **Select Servers** step.

6. Click **Next**.
7. In the **Configure Ports and Weights** step, specify the weight of each ECS instance.

   An ECS instance with a higher weight receives more requests.

   > Notice
   > - Valid values of weights: 0 to 100. If you set the weight of a backend server to 0, the server does not receive requests.
   > - If session persistence is enabled, requests may not be evenly distributed to backend servers.

8. Click **Add**.
9. Select the ECS instances and click **OK**.

6.2.2. Add on-premises servers to the default server group

This topic describes how to add on-premises servers to the default server group. Before you use the Server Load Balancer (SLB) service, you must add at least one backend server to the default server group to receive requests.

Prerequisites

Applications are deployed on the on-premises servers, and the servers are ready to receive requests.

Procedure

1. Log on to the SLB console.
2. Find the SLB instance that you want to manage and click the instance ID.
3. Click the **Default Server Group** tab.
4. Click **Add IDC Server**.
5. In the **My Servers** panel, click **Add**.
6. Select a virtual private cloud (VPC) from the VPC Connected to IDC drop-down list, enter a name for the on-premises server that you want to add, and then specify the IP address of the server.
7. Click **Next**.
8. On the **Configure Ports and Weights** wizard page, specify the weight of each on-premises server.

   A server with a higher weight receives more requests.

   You can change the weight of a server and then move the pointer over the  icon to synchronize the change to other servers:

   - If you click **Replicate to Below**, the weights of all servers below the current server are set to the weight of the current server.
   - If you click **Replicate to Above**, the weights of all servers above the current server are set to the weight of the current server.
   - If you click **Replicate to All**, the weights of all servers in the default server group are set to the

weight of the current server.

- If you click **Reset** after you clear the weight of the current server, the weights of all the other servers in the default server group are also cleared.

> **Notice** If you set the weight of a server to 0, the server does not receive requests.

9. Click **Add**.
10. Click **OK**.

6.2.3. Change the weight of a backend server

This topic describes how to change the weight of a backend server to adjust the proportion of requests sent to the backend server.

Procedure

1. Log on to the SLB console.
2. Find the Server Load Balancer (SLB) instance that you want to manage and click the instance ID.
3. Click the **Default Server Group** tab.
4. Find the backend server that you want to manage, move the pointer over the weight specified for the backend sever, and then click the  icon that appears.
5. Change the weight and click **OK**.

   An Elastic Compute Service (ECS) instance or on-premises server with a higher weight receives more requests.

> **Notice** The value of the weight ranges from 0 to 100. If the weight of a backend server is set to 0, the server does not receive requests.

6.2.4. Remove a backend server

This topic describes how to remove a backend server that is no longer needed.

Procedure

1. Log on to the SLB console.
2. Find the Server Load Balancer (SLB) instance that you want to manage and click the instance ID.
3. Click the **Default Server Group** tab.
4. Find the backend server that you want to remove and click **Remove** in the **Actions** column.
5. In the message that appears, click **OK**.

6.3. vServer groups

6.3.1. Add ECS instances to a vServer group

This topic describes how to create a vServer group and then add Elastic Compute Service (ECS) instances to the vServer group. If you associate a vServer group with a listener, the listener distributes requests only to backend servers in the vServer group.

Prerequisites

- Create an SLB instance.
- ECS instances are created and applications are deployed on the ECS instances to receive requests.

Context

Before you create a vServer group, take note of the following items:

- An ECS instance can be added to multiple vServer groups.
- A vServer group can be associated with multiple listeners of a Server Load Balancer (SLB) instance.
- A vServer group consists of ECS instances and application service ports.

Procedure

1. Log on to the SLB console.
2. Find the SLB instance that you want to manage and click the instance ID.
3. Click the **VServer Groups** tab.
4. On the **VServer Groups** tab, click **Create VServer Group**.
5. On the **Create VServer Group** page, configure the vServer group.
   i. In the **VServer Group Name** field, enter a name for the vServer group.
   ii. Click **Add**. In the **My Servers** panel, select the ECS instances that you want to add.
   iii. Click **Next**.

iv. Specify a port and a weight for each ECS instance and then click **Add**.

Set the Port and Weight parameters based on the following information:

- **Port**: The backend port opened on an ECS instance to receive requests.
  You can set the same port for multiple backend servers of the same SLB instance. In addition, you can click **Add Port** to add multiple ports for a backend server.
- **Weight**: An ECS instance with a higher weight receives more requests.

> **Notice** If you set the weight of a server to 0, the server does not receive requests.

You can change the weight of a server and then move the pointer over the icon to synchronize the change to other servers:

- If you click **Replicate to Below**, the weights of all servers below the current server are set to the weight of the current server.
- If you click **Replicate to Above**, the weights of all servers above the current server are set to the weight of the current server.
- If you click **Replicate to All**, the weights of all servers in the default server group are set to the weight of the current server.
- If you click **Reset** after you clear the weight of the current server, the weights of all the other servers in the default server group are also cleared.

> **Notice** If you set the weight of a server to 0, the server does not receive requests.

v. Click **Add**.

6. Click **Create**.

6.3.2. Add on-premises servers to a vServer group

This topic describes how to create a vServer group and then add on-premises servers to the vServer group. A vServer group is a group of Elastic Compute Service (ECS) instances or on-premises servers. If you associate a vServer group with a listener, the listener distributes requests only to backend servers in the vServer group.

Prerequisites

Before you create a vServer group, make sure that applications are deployed on the on-premises servers and the servers are ready to receive requests.

Context

Before you create a vServer group, take note of the following items:

- An on-premises server can be added to multiple vServer groups.
- A vServer group can be associated with multiple listeners of a Server Load Balancer (SLB) instance.
- A vServer group consists of on-premises servers and application service ports.

Procedure

1. Log on to the SLB console.
2. Find the SLB instance that you want to manage and click the instance ID.

3. Click the **VServer Groups** tab.
4. On the **VServer Groups** tab, click **Create VServer Group**.
5. On the **Create VServer Group** page, configure the vServer group.
   i. In the **VServer Group Name** field, enter a name for the vServer group.
   ii. Click **Add IDC Server**.
   iii. In the **My Servers** panel, click **Add**.
   iv. Select a virtual private cloud (VPC) from the VPC Connected to IDC drop-down list, enter a name for the on-premises server that you want to add, and then specify the IP address of the server.

   The on-premises server must can use the specified IP address to communicate with the VPC.
   v. Click **Next**.
   vi. Specify a port and weight for each on-premises server, and then click **Add**.

   Set the Port and Weight parameters based on the following information:

   - **Port**: The backend port opened on an on-premises server to receive requests. Multiple ports can be opened on an on-premises server.
     You can set the same port for multiple backend servers of the same SLB instance.
   - **Weight**: A server with a higher weight receives more requests.

     > **Notice** If you set the weight of a server to 0, the server does not receive requests.

   You can change the weight of a server and then move the pointer over the icon to synchronize the change to other servers:

   - If you click **Replicate to Below**, the weights of all servers below the current server are set to the weight of the current server.
   - If you click **Replicate to Above**, the weights of all servers above the current server are set to the weight of the current server.
   - If you click **Replicate to All**, the weights of all servers in the default server group are set to the weight of the current server.
   - If you click **Reset** after you clear the weight of the current server, the weights of all the other servers in the default server group are also cleared.

   > **Notice** If you set the weight of a server to 0, the server does not receive requests.

   vii. Click **Add**.
6. Click **Create**.

6.3.3. Modify a vServer group

This topic describes how to modify the settings of Elastic Compute Service (ECS) instances or on-premises servers in a vServer group.

Procedure

1. Log on to the SLB console.

2. Find the Server Load Balance (SLB) instance that you want to manage and click the instance ID.
3. Click the **VServer Groups** tab.
4. Find the vServer group that you want to modify and click **Edit** in the Actions column.
5. Modify the ports and weights of the ECS instances or on-premises servers, and click **Save**.

6.3.4. Delete a vServer group

This topic describes how to delete a vServer group. If a vServer group is no longer needed to receive traffic, you can delete the vServer group.

Procedure

1. Log on to the SLB console.
2. Find the Server Load Balance (SLB) instance that you want to manage and click the instance ID.
3. Click the **VServer Groups** tab.
4. Find the vServer group that you want to delete and click **Delete** in the Actions column.
5. In the message that appears, click **OK**.

6.4. Primary/secondary server groups

6.4.1. Add ECS instances to a primary/secondary server group

This topic describes how to create a primary/secondary server group and then add Elastic Compute Service (ECS) instances to the primary/secondary server group. A primary/secondary server group contains a primary server and a secondary server that can fail over to prevent service interruptions. By default, the primary server receives all requests that are distributed by a Server Load Balancer (SLB) instance. When the primary server fails, requests are redirected to the secondary server.

Prerequisites

Before you create a primary/secondary server group, make sure that the following requirements are met:

- An SLB instance is created. For more information, see Create an SLB instance.
- ECS instances are created and applications are deployed on the ECS instances to receive requests.

Procedure

1. Log on to the SLB console.
2. Find the SLB instance that you want to manage and click the instance ID.
3. Click the **Primary/Secondary Server Groups** tab.
4. On the **Primary/Secondary Server Groups** tab, click **Create Primary/Secondary Server Group**.
5. On the **Create Primary/Secondary Server Group** page, configure the primary/secondary server group.
    i. In the **Primary/Secondary Server Group Name** field, enter a name for the primary/secondary server group.

ii. Click **Add**. In the **My Servers** panel, select the ECS instances that you want to add on the **Select Servers** wizard page.

You can add only two ECS instances to a primary/secondary server group.

iii. Click **Next**.

iv. Configure the backend port opened on each ECS instance to receive requests, and then click **Add**.

You can set multiple ports for an ECS instance.

v. Set an ECS instance as the primary server.

vi. Click **Create**.

6.4.2. Add on-premises servers to a primary/secondary server group

This topic describes how to create a primary/secondary server group and then add on-premises servers to the primary/secondary server group. A primary/secondary server group contains a primary server and a secondary server that can fail over to prevent service interruption. By default, the primary server receives all requests that are distributed by a Server Load Balancer (SLB) instance. When the primary server fails, requests are redirected to the secondary server.

Prerequisites

Applications are deployed on the on-premises servers, and the servers are ready to receive requests.

Procedure

1. Log on to the SLB console.
2. Find the SLB instance that you want to manage and click the instance ID.
3. Click the **Primary/Secondary Server Groups** tab.
4. On the **Primary/Secondary Server Groups** tab, click **Create Primary/Secondary Server Group**.
5. On the **Create Primary/Secondary Server Group** page, configure the primary/secondary server group.
   i. In the **Primary/Secondary Server Group Name** field, enter a name for the primary/secondary server group.
   ii. Click **Add IDC Server**. In the **My Servers** panel, select the servers that you want to add on the **Select Servers** wizard page.

   You can add only two on-premises servers to a primary/secondary server group.

   iii. In the **My Servers** panel, click **Add**.
   iv. Select a virtual private cloud (VPC) from the VPC Connected to IDC drop-down list, enter a name for the on-premises server that you want to add, and then specify the IP address of the server.

   The on-premises server must can use the specified IP address to communicate with the VPC.

   v. Click **Next**.
   vi. Configure the backend port opened on each on-premises server to receive requests, and then click **Add**.

   You can set multiple ports for an on-premises server.

vii. Set a server as the primary server.

viii. Click **Create**.

6.4.3. Delete a primary/secondary server group

This topic describes how to delete a primary/secondary server group of a Server Load Balancer (SLB) instance. If a primary/secondary server group is no longer needed to receive traffic, you can delete the primary/secondary server group.

Procedure

1. Log on to the SLB console.
2. Find the SLB instance that you want to manage and click the instance ID.
3. Click the **Primary/Secondary Server Groups** tab.
4. On the **Primary/Secondary Server Groups** tab, find the primary/secondary server group that you want to delete and click **Delete** in the **Actions** column.
5. In the message that appears, click **OK**.

6.5. Add backend servers by specifying their ENIs

An elastic network interface (ENI) is a virtual network interface that can be attached to an Elastic Compute Service (ECS) instance in a virtual private cloud (VPC). ENIs can improve the availability of clusters and offer cost savings in service failovers and fine-grained network management. You can add a backend server to an SLB instance by specifying the primary or secondary IP address of the ENI used by the backend server.

Context

If multiple ENIs are attached to the ECS instance that you want to add to an SLB instance, you can specify both the primary and secondary ENIs of the ECS instance.

> **Note**
>
> - You can specify the primary and secondary ENIs of ECS instances only for high-performance SLB instances.
> - You can specify only secondary ENIs in the console. To specify primary ENIs, make an API request.

Procedure

1. Log on to the SLB console.
2. In the top navigation bar, select the region where the SLB instance is deployed.
3. On the **Instances** page, click the ID of the SLB instance for which you want to create a server group.
4. Click the **vServer Groups**, **Default Server Group**, or **Primary/Secondary Server Groups** tab.

> **Note** You can add the primary and secondary ENIs of ECS instances to the default server group, a vServer group, or a primary/secondary server group. A vServer group is used in this example.

5. On the **vServer Groups** tab, click **Create vServer Group**.
6. On the **Create vServer Group** page, click **Add**.
7. In the **Select Servers** step, select **ECS Instance Name** from the drop-down list and turn on **Advanced Mode**.
8. Select the backend servers that you want to add to the vServer group and click **Next**.
9. In the **Configure Ports and Weights** step, specify the ports and weights of the backend servers and click **Add**.

   After you add backend servers to the server group of the listener, you can view the server group that contains the primary and secondary ENIs on the **Instances** page.

   - indicates an ECS instance.
   - indicates a primary or secondary ENI.

7.Health checks

7.1. Health check overview

This topic describes the health check feature of Server Load Balancer (SLB). SLB performs health checks to verify the availability of Elastic Compute Service (ECS) instances that serve as backend servers. The health check feature improves the overall availability of frontend services and prevents service interruptions caused by backend server anomalies.

After you enable the health check feature for an SLB instance, the SLB instance periodically checks whether the backend ECS instances are healthy. When the SLB instance detects an unhealthy ECS instance, the SLB instance distributes new requests to other healthy ECS instances. When the unhealthy ECS instance recovers, the SLB instance distributes requests to the recovered ECS instance again.

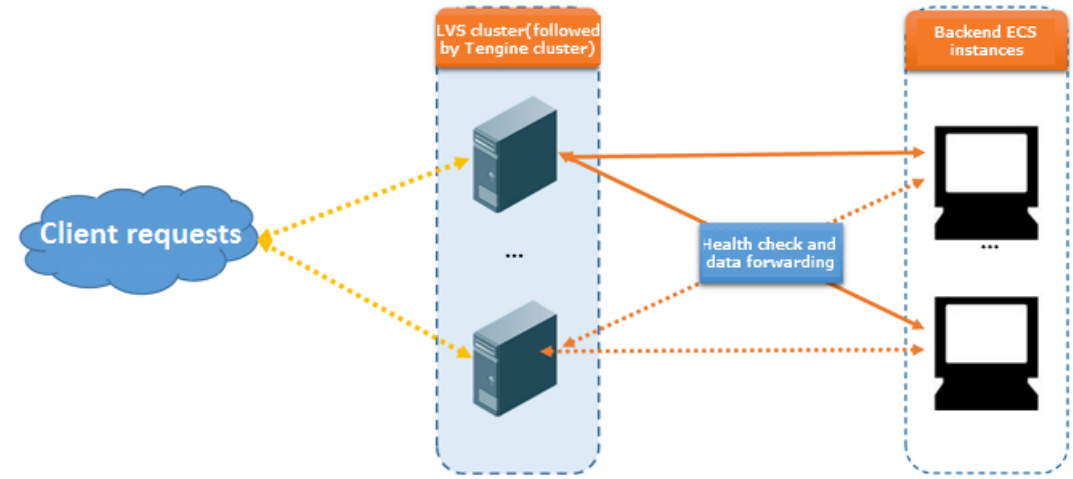
If your business is highly sensitive to traffic fluctuations, frequent health checks may affect the availability of your business. To reduce the impacts of health checks on your business, you can reduce the health check frequency, increase the health check interval, or change Layer 7 health checks to Layer 4 health checks. To ensure business continuity, we recommend that you enable the health check feature.

Health check workflow

SLB uses a cluster architecture. Nodes in an LVS or Tengine cluster are used to forward data and perform health checks.

Nodes in an LVS cluster forward data and perform health checks independently and in parallel based on the configured load balancing policies. If a node in an LVS cluster detects an unhealthy backend ECS instance, the node stops sending client requests to the unhealthy ECS instance. This operation is synchronized across all of the nodes in the LVS cluster.

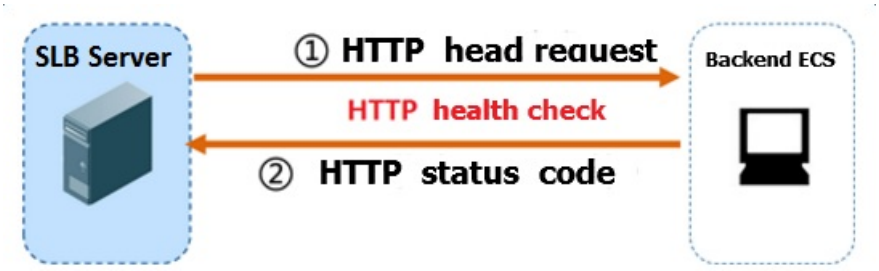
SLB uses the 100.64.0.0/10 CIDR block for health checks. Make sure that the backend ECS instances do not block this CIDR block. You do not need to configure a security group rule to allow access from the CIDR block 100.64.0.0/10 unless you have configured security rules such as iptables. Permitting 100.64.0.0/10 does not increase potential risks because the CIDR block is reserved by Alibaba Cloud. IP addresses within the CIDR block are not allocated to users.



Layer 7 health checks (HTTP or HTTPS listeners)

If you add Layer 7 HTTP or HTTPS listeners to your SLB instance, SLB checks the status of backend ECS instances by sending HTTP HEAD requests, as shown in the following figure.

For HTTPS listeners, certificates are managed in SLB. To improve system performance, HTTPS is not used to exchange data, including health check data and business data, between the SLB instance and backend ECS instances.

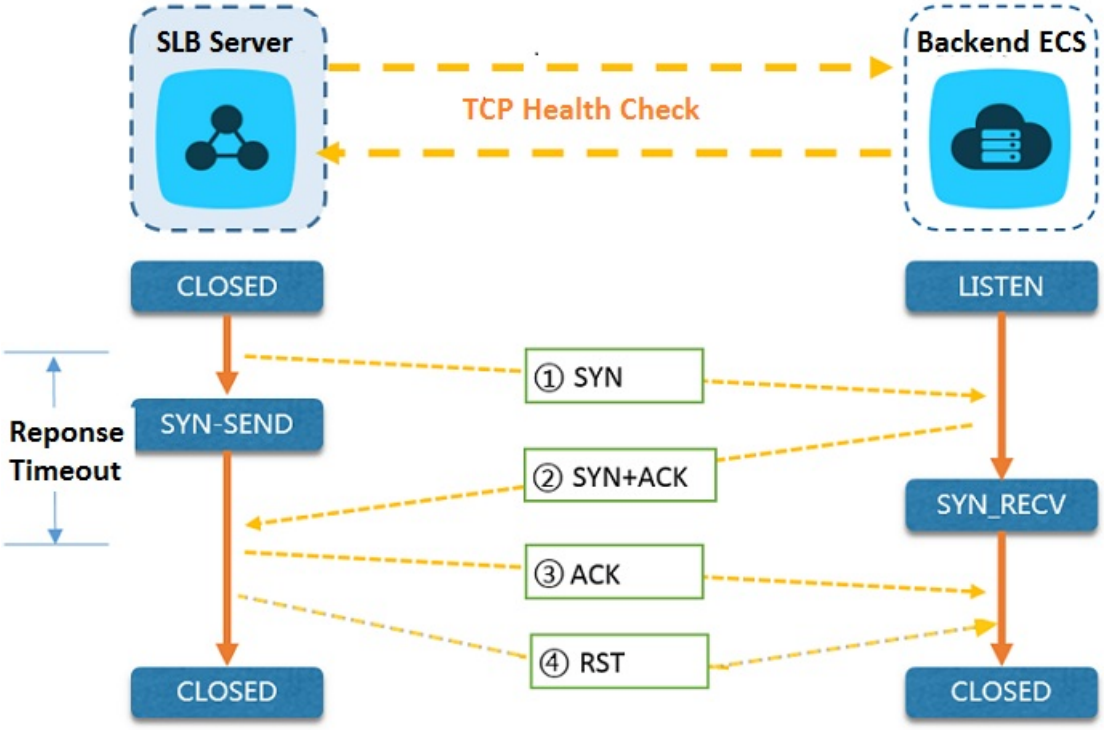


How SLB performs Layer 7 health checks:

1. A Tengine node sends an HTTP HEAD request to a backend ECS instance based on the health check configurations of the Layer 7 listener. The HTTP HEAD request is sent to an address in the following format: ECS instance private IP address + Health check port + Health check path. The HTTP HEAD request also carries the domain name specified in the health check configurations.
2. After the backend ECS instance receives the request, the ECS instance checks the status of the application and returns a relevant HTTP status code.
3. If the Tengine node does not receive a response from the backend ECS instance within the specified timeout period, the backend server is considered unhealthy.
4. If the Tengine node receives a response from the backend ECS instance within the specified timeout period, the node matches the returned status code against the status codes specified in the health check configurations. If the returned status code matches one of the specified status codes, the backend server is considered healthy. Otherwise, the backend server is considered unhealthy.

Layer 4 health checks (TCP listeners)

If you add TCP listeners to your SLB instance, SLB performs Layer 4 health checks by establishing TCP sessions, as shown in the following figure. This improves the health check efficiency.



How SLB performs Layer 4 health checks when TCP listeners are used:

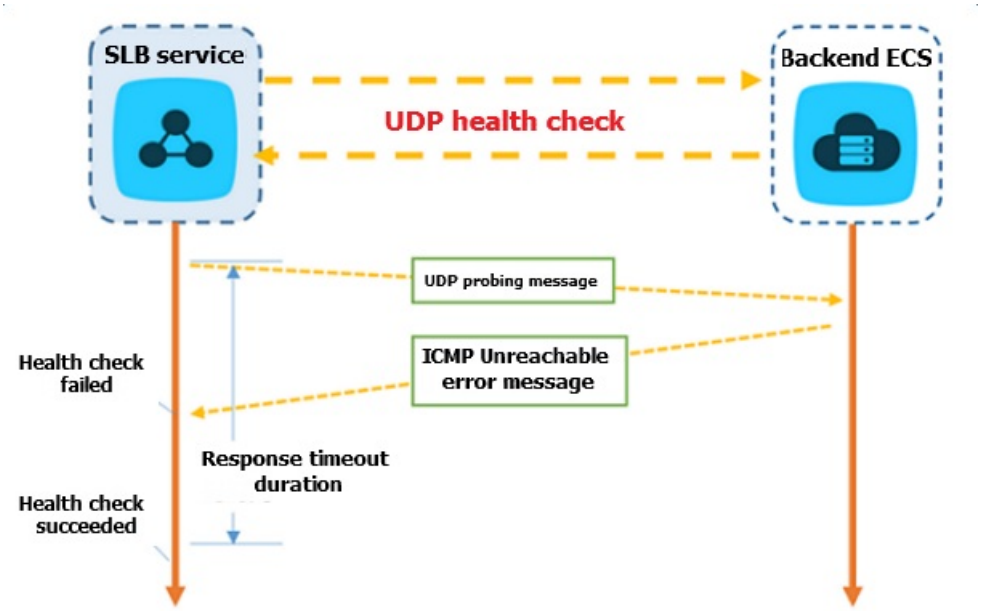
1. An LVS node sends a TCP SYN packet to a backend ECS instance based on the health check configurations of the TCP listener. The TCP SYN packet is sent to an address in the following format: ECS instance private IP address + Health check port.
2. After the backend ECS instance receives the request, the ECS instance returns an SYN-ACK packet if the ECS port on which SLB listens is normal.
3. If the LVS node does not receive a response from the backend ECS instance within the specified timeout period, the backend ECS instance is considered unhealthy. Then, the node sends an RST packet to the backend ECS instance to terminate the TCP session.
4. If the LVS node receives a response from the backend ECS instance within the specified timeout period, the node considers the ECS instance healthy and then sends an RST packet to the backend ECS instance to terminate the TCP session.

Note A TCP three-way handshake is performed to establish TCP sessions. After the LVS node receives the SYN-ACK packet from the backend ECS instance, the node sends an ACK packet, and then immediately sends an RST packet to terminate the TCP session.
Due to the three-way handshake mechanism, the backend ECS instances may take TCP session terminations as connectivity errors. Then, the ECS instances may record an error, such as `Connection reset by peer`, in the application log, such as the Java connection pool log.
Solution:

- Configure HTTP health checks in the TCP listener.
- Enable client IP address preservation on backend ECS instances. This enables the ECS instances to ignore connectivity errors that are caused by access from the SLB CIDR block.

Layer 4 health checks (UDP listeners)

If you add UDP listeners to your SLB instance, SLB checks the status of backend ECS instances by sending UDP packets, as shown in the following figure.



How SLB performs Layer 4 health checks when UDP listeners are used:

1. An LVS node sends a UDP packet to a backend ECS instance based on the health check configurations of the UDP listener. The UDP packet is sent to an address in the following format: ECS instance private IP address + Health check port.
2. If the ECS port on which SLB listens is abnormal, an Internet Control Message Protocol (ICMP) error message, such as `port XX unreachable`, is returned. If the ECS port is normal, no message is returned.
3. If the LVS node receives an ICMP error message within the timeout period, the backend ECS instance is considered unhealthy.
4. If the LVS node does not receive ICMP error messages from the backend ECS instance within the timeout period, the ECS instance is considered healthy.

Note UDP health check results may not reflect the actual status of the application on a backend ECS instance in the following situation:
If a backend ECS instance runs the Linux operating system, the rate at which ICMP messages are sent in high concurrency scenarios is limited due to the ICMP attack prevention mechanism of Linux. In this case, even if an application error occurs, SLB may consider the backend ECS instance healthy because SLB has not received the error message `port XX unreachable`. As a result, the health check result is different from the actual application status.
Solution:
You can set SLB to send a given string to a backend server. The backend server is considered healthy only if it returns a given response to SLB. However, the application on the backend server must be configured accordingly to return responses.

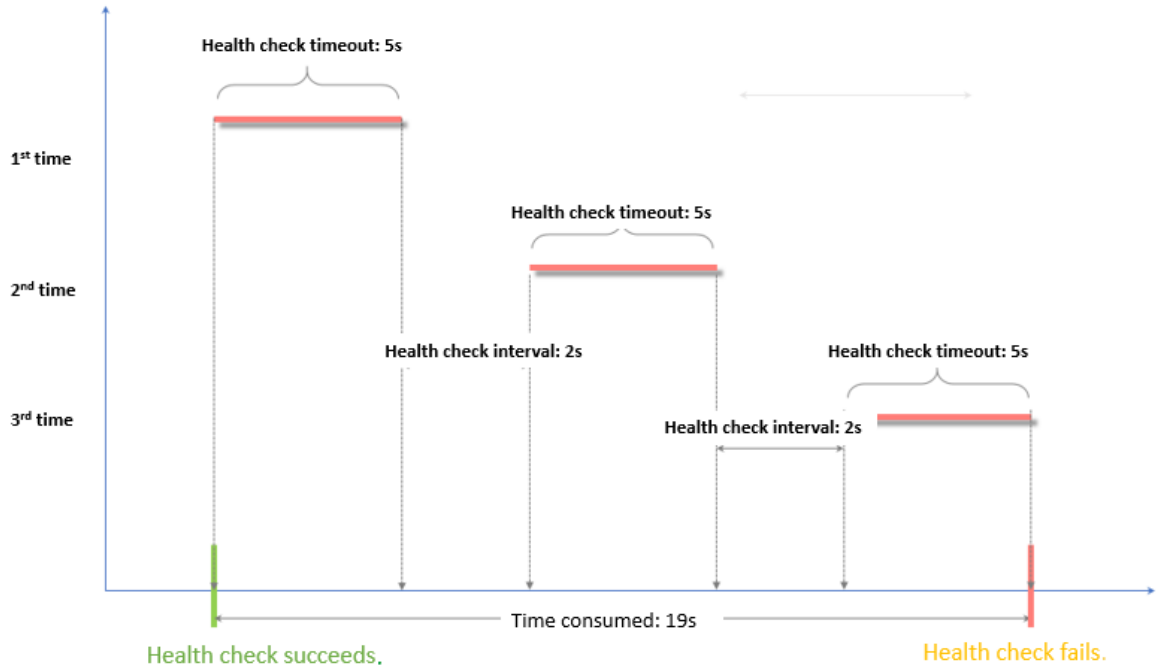
Health check time window

The health check feature improves the availability of your services. However, frequent failovers caused by unhealthy backend servers may affect system availability. Health check time windows are introduced to control failovers. A failover is performed only a backend server consecutively pass or fail a certain number of health checks within a time window. The health check time window is determined by the following factors:

- Health check interval: how often health checks are performed.
- Response timeout: the time to wait for a response.
- Health check threshold: the number of consecutive successes or failures of health checks.

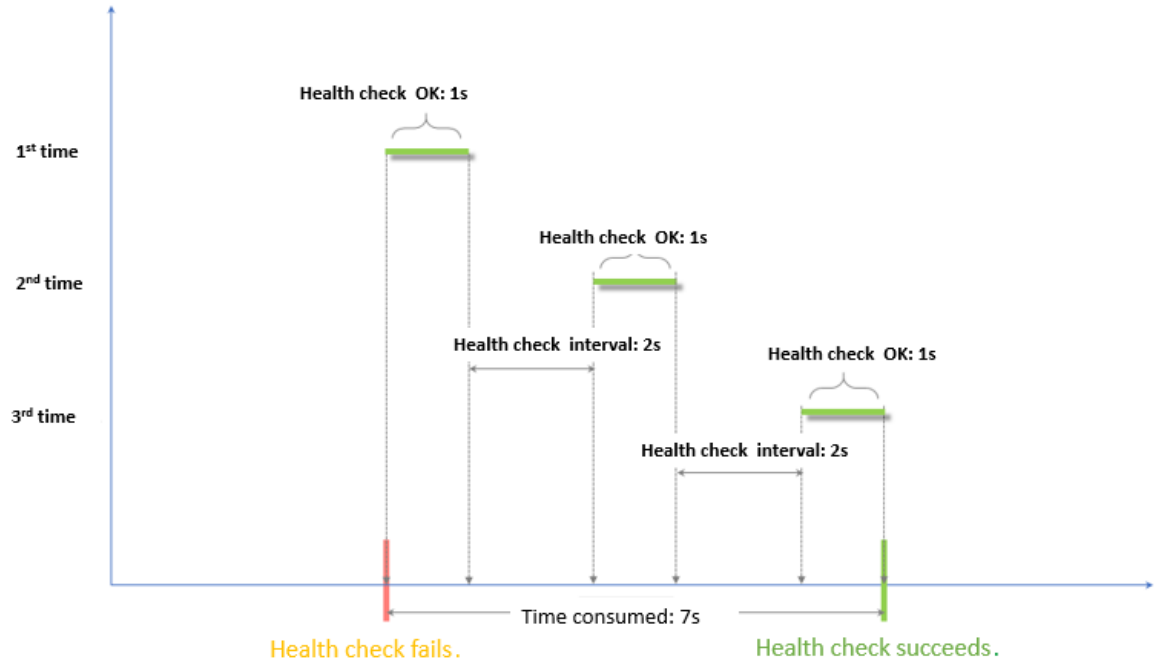
The health check time window is calculated based on the following formula:

- Time window for health check failures = Response timeout × Unhealthy threshold + Health check interval × (Unhealthy threshold - 1)



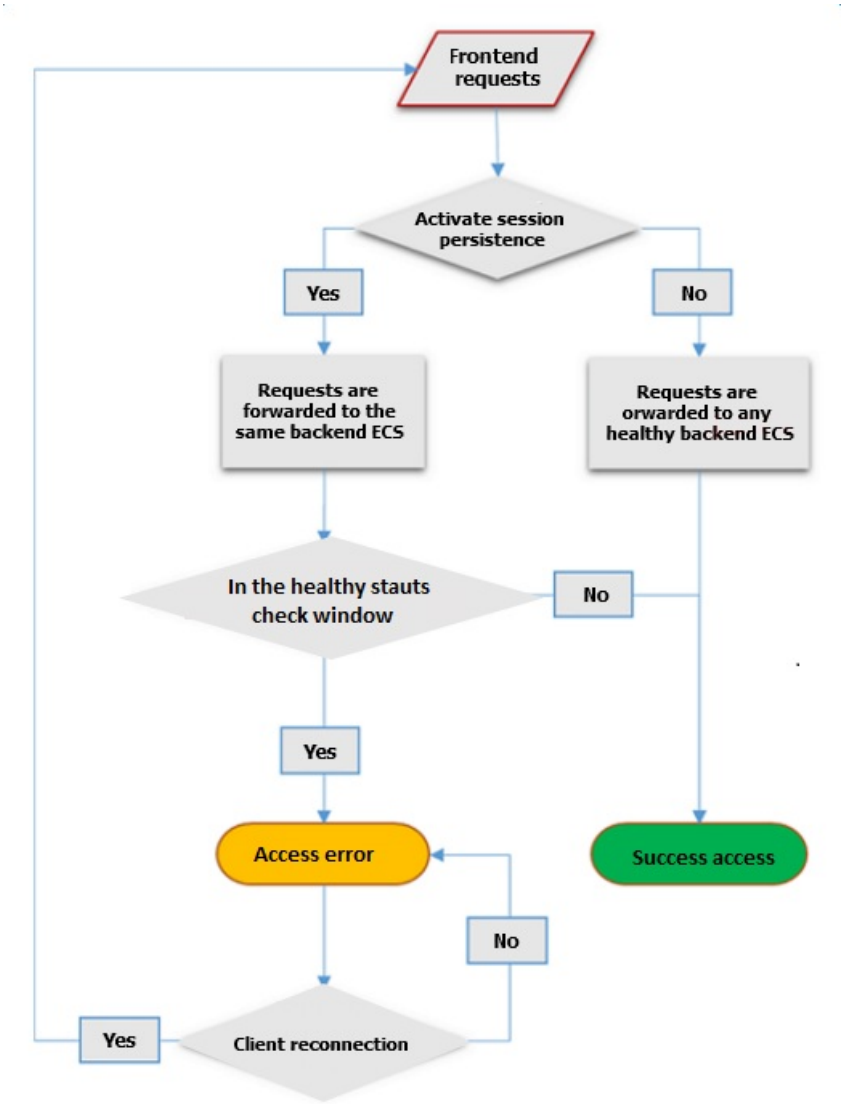
- Time window for health check successes = Response time of a successful health check × Healthy threshold + Health check interval × (Healthy threshold - 1)

> **Note** The response time of a successful health check is the duration from the time when the health check request is sent to the time when the response is received. When TCP health checks are configured, the response time is short and almost negligible because the only check item is whether the probed port is alive. When HTTP health checks are configured, the response time depends on the performance and load of the application server and is typically within a few seconds.



The health check result has the following impacts on request forwarding:

- If the backend ECS instance fails the health check, new requests are distributed to other backend ECS instances. The client can still access the application as normal.
- If the backend ECS instance passes the health check, new requests are distributed to the ECS instance. The client can access the application as normal.
- If an exception occurs on the backend ECS instance and a request arrives during a time window for health check failures, the request is still sent to the unhealthy backend ECS instance. This is because the number of failed health checks has not reached the unhealthy threshold (three times by default). In this case, the client fails to access the application.



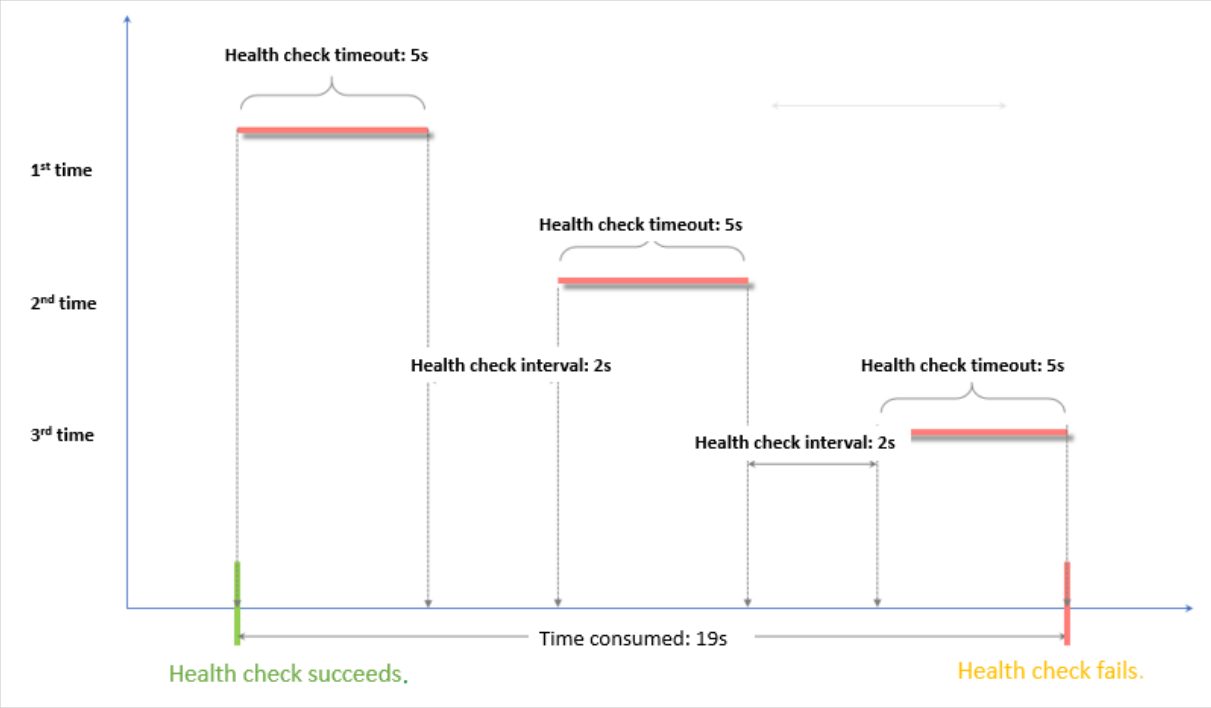
Examples of health check response timeout and health check interval

The following health check settings are used:

- **Response Timeout Period: 5 Seconds**
- **Health Check Interval: 2 Seconds**
- **Healthy Threshold: 3 Times**
- **Unhealthy Threshold: 3 Times**

Time window for health check failures = Response timeout × Unhealthy threshold + Health check interval × (Unhealthy threshold - 1). In this example, the time window is 19 seconds based on the formula 5 × 3 + 2 × (3 - 1). If the response time of a health check exceeds 19 seconds, the health check fails.

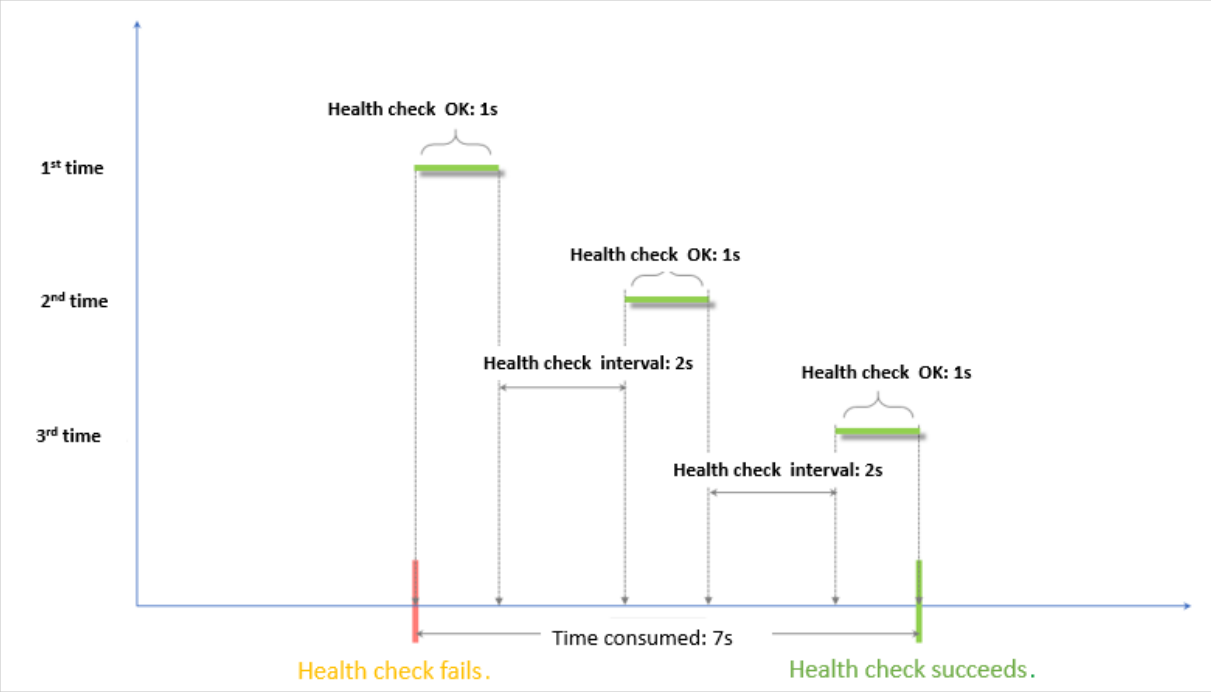
The following figure shows the time window from a healthy status to an unhealthy status.



Time window for health check successes = Response time of a successful health check × Healthy threshold + Health check interval × (Healthy threshold - 1). In this example, the time window is 7 seconds based on the formula (1 × 3) + 2 × (3 – 1). If the response time of a successful health check is less than 7 seconds, the health check succeeds.

> **Note** The response time of a successful health check is the duration from the time when the health check request is sent to the time when the response is received. When TCP health checks are configured, the response time is short and almost negligible because the only check item is whether the probed port is alive. When HTTP health checks are configured, the response time depends on the performance and load of the application server and is typically within a few seconds.

The following figure shows the time window from an unhealthy status to a healthy status. In the following figure, the time that is required for the server to respond to a health check request is 1 second.



Domain name setting in HTTP health checks

You can specify a domain name for HTTP health checks. This setting is optional. Some application servers must verify the host field in requests before the application servers can accept the requests. In this case, the request header must carry the host field. If a domain name is configured in health check settings, SLB adds this domain name to the host field when SLB forwards a health check request to one of the preceding application servers. If no domain name is configured, SLB does not include the host field in the request. As a result, the request is rejected by the application server and the health check fails. If your application server verifies the host field in requests, you must configure a domain name in health check settings to ensure that the health check feature works as expected.

7.2. Configure health checks

This topic describes how to configure health checks. You can configure health checks when you add a listener. The default health check settings can meet your requirements in most cases.

Procedure

1. Log on to the SLB console.
2. Find the Server Load Balancer (SLB) instance that you want to manage and click the instance ID.
3. On the page that appears, click the **Listener** tab.
4. Click **Add Listener**, or find the listener that you want to manage and click **Modify Listener** in the **Actions** column.
5. Click **Next** to go to the **Health Check** step and configure the health check.

   We recommend that you use the default settings when you configure health checks.

   ### Health check configuration

Parameter	Description
Health Check Protocol	Select the protocol that the SLB instance uses when it performs health checks. For TCP listeners, both TCP health checks and HTTP health checks are supported. ○ A TCP health check probes whether a server port is healthy at the network layer by sending SYN packets to the port. ○ An HTTP health check probes whether a backend server is healthy by simulating HTTP requests with the HEAD or GET method.
Health Check Method (for HTTP and HTTPS health checks only)	Health checks of Layer 7 (HTTP or HTTPS) listeners support both the HEAD and GET methods. The HEAD method is used by default. If your backend server does not support the HEAD method or if the HEAD method is disabled, the health check may fail. To resolve this issue, you can use the GET method instead. If the GET method is used and the response size exceeds 8 KB, the response is truncated. However, you can still identify the health check result based on the response.
Health Check Path and **Health Check Domain Name (Optional)** (for HTTP health checks only)	By default, when SLB performs HTTP health checks, it tests the default homepage configured on a backend Elastic Compute Service (ECS) instance by sending HTTP HEAD requests to the private IP address of the ECS instance. If you do not want to use the default homepage for health checks, you can manually specify a URL. Some application servers must verify the host field in requests before the application servers can accept the requests. In this case, the request header must carry the host field. If a domain name is configured in health check settings, SLB adds this domain name to the host field when SLB forwards a health check request to one of the preceding application servers. If no domain name is configured, SLB does not include the host field in the request. As a result, the request is rejected by the application server and the health check fails. If your application server needs to verify the host field in requests, you must configure a domain name in health check settings to ensure that the health check feature functions as normal.
Normal Status Code (for HTTP health checks only)	Select the HTTP status code that indicates successful health checks. By default, http_2xx and http_3xx are selected.
Health Check Port	The backend server port that is probed for health checks. By default, the backend server port configured on the listener is probed for health checks. **Note** If a vServer group or a primary/secondary server group is associated with the listener, and the ECS instances in the server group use different ports, leave this parameter empty. SLB automatically probes the port of each ECS instance to perform health checks.

Parameter	Description
Response Timeout	Specify the timeout period for a health check response. If the backend ECS instance does not send an expected response within the specified period of time, the ECS instance is considered unhealthy. Valid values: 1 to 300. Unit: seconds. Default value for UDP listeners: 10. Default value for HTTP, HTTPS, and TCP listeners: 5.
Health Check Interval	The time interval between consecutive health checks. All nodes in the LVS cluster perform health checks independently and in parallel on backend ECS instances at the specified interval. However, the frequency at which a single ECS instance is probed does not conform to the health check interval because the nodes probe the ECS instance at different times. Valid values: 1 to 50. Unit: seconds. Default value for UDP listeners: 5. Default value for HTTP, HTTPS, and TCP listeners: 2.
Unhealthy Threshold	The number of health checks that a backend ECS instance must consecutively fail before the ECS instance is considered unhealthy. Valid values: 2 to 10. Default value: 3.

6. Click **Next**.

7.3. Disable the health check feature

This topic describes how to disable the health check feature for a Server Load Balancer (SLB) instance. If you disable the health check feature, requests may be distributed to unhealthy backend Elastic Compute Service (ECS) instances. This causes service interruptions. We recommend that you enable the health check feature.

Context

> **Note** You can disable the health check feature only for HTTP and HTTPS listeners. The health check feature for UDP and TCP listeners cannot be disabled.

Procedure

1. Log on to the SLB console.
2. On the **Instances** page, find the SLB instance that you want to manage and click its ID.
3. On the **Listeners** tab, find the listener for which you want to disable the health check feature and click **Modify Listener** in the **Actions** column.
4. On the **Configure Listener** page, click **Next** to proceed to the **Health Check** wizard page.
5. Turn off the health check switch and click **Next**. On the wizard page that appears, click **Submit** and click **OK**.

8.Certificate management

8.1. Overview of certificates

This topic provides an overview of certificates. To create an HTTPS listener, you must first upload the required third-party server certificate and certificate issued by a certificate authority (CA) to Server Load Balancer (SLB). You do not need to install certificates on backend servers after you upload the certificates to SLB.
SLB supports third-party certificates. To upload a third-party certificate, you must have the files that contain the public key and private key of the certificate.

HTTPS server certificates and client CA certificates are supported.

You can create at most 100 certificates with each account.

8.2. Certificate requirements

Server Load Balancer (SLB) supports only certificates in the PEM format. Before you upload a certificate, make sure that the certificate content, certificate chain, and private key meet the corresponding format requirements.

Certificates issued by a Root CA

If your certificate is issued by a Root certification authority (CA), the received certificate file contains only one certificate. You do not need other certificates to prove that the matching domain name is trusted.

The certificate must meet the following format requirements:

- The certificate must start with `-----BEGIN CERTIFICATE-----` and end with -----END CERTIFICATE-----.
- Each line (except the last line) must contain 64 characters. The last line can contain 64 or fewer characters.
- The certificate content cannot contain spaces.

Certificates issued by an Intermediate CA

If your certificate is issued by an Intermediate CA, the received certificate file contains multiple certificates. You must upload both the server certificate and the required intermediate certificates to SLB.

The format of the certificate chain must meet the following requirements:

- The server certificate must be put first and the content of the required intermediate certificates must be put underneath without blank lines between the certificates.
- The certificate content cannot contain spaces.
- Blank lines are not allowed between the certificates. Each line must contain 64 characters. For more information, see RFC1421.
- Certificates must meet the corresponding format requirements. In most cases, the Intermediate CA provides instructions about the certificate format when certificates are issued. The certificates must meet the format requirements.

The following section provides a sample certificate chain:

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

Public keys of certificates

SLB supports the following public key algorithms:

- RSA 1024
- RSA 2048
- RSA 4096
- ECDSA P-256
- ECDSA P-384
- ECDSA P-521

RSA private key formats

When you upload a server certificate, you must upload the private key of the certificate.

An RSA private key must meet the following format requirements:

- The private key must start with `-----BEGIN RSA PRIVATE KEY-----` and end with -----END RSA PRIVATE KEY-----, and these parts must also be uploaded.
- Blank lines are not allowed in the content. Each line (except the last line) must contain 64 characters. The last line can contain 64 or fewer characters. For more information, see RFC1421.

You may use an encrypted private key. For example, the private key starts with `----BEGIN PRIVATE KEY-----` and ends with -----END PRIVATE KEY-----, or starts with `-----BEGIN ENCRYPTED PRIVATE KEY-----` and ends with -----END ENCRYPTED PRIVATE KEY-----. The private key may also contain `Proc-Type: 4,ENCRYPTED`. In this case, you must first run the following command to convert the private key:

```
openssl rsa -in old_server_key.pem -out new_server_key.pem
```

8.3. Upload certificates

This topic describes how to create and upload certificates to Server Load Balancer (SLB). Before you create an HTTPS listener, you must upload the required server certificate and CA certificate to SLB. You do not need to configure certificates on backend servers after you upload the certificates to SLB.

Prerequisites

- A server certificate is purchased.
- A CA certificate and a client certificate are generated.

Context

You can create at most 100 certificates with each account.

Procedure

1. In the left-side navigation pane, click **Certificates**.
2. On the Certificates page, click **Create Certificate**.
3. In the **Create Certificate** panel, set the following parameters and click **Create**.

Parameter	Description
Certificate Name	Enter a name for the certificate. The name must be 1 to 80 characters in length, and can contain only letters, digits, hyphens (-), forward slashes (/), periods (.), underscores (_), and asterisks (*).
Organization	Select the organization to which the certificate belongs.
Resource Group	Select the resource set to which the certificate belongs.
Certificate Standard	Select the type of certificate. You can select **International Standard Certificate** or **National Standard Certificate**.
Public Key Certificate	The content of the server certificate. Paste the content into the editor. Click **Example** to view the valid certificate formats. For more information, see Certificate requirements.
Private Key	The private key of the server certificate. Paste the private key into the editor. Click **Example** to view the valid certificate formats. For more information, see Certificate requirements. **Notice** A private key is required only when you upload a server certificate.
Region	Select the region where you want to deploy the certificate.

8.4. Generate a CA certificate

When you configure an HTTPS listener, you can use a self-signed CA certificate. You can also use the CA certificate to sign a client certificate.

Generate a CA certificate by using OpenSSL

1. Run the following commands to create a *ca* folder under the */root* directory and then create four subfolders under the *ca* folder:

```
$ sudo mkdir ca
$ cd ca
$ sudo mkdir newcerts private conf server
```

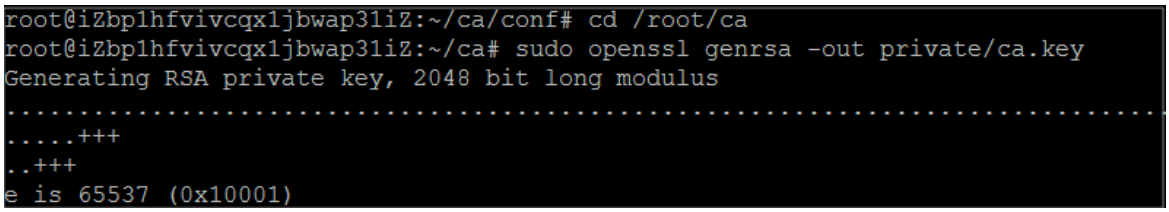
   - *newcerts* is used to store the digital certificate signed by the CA certificate.
   - *private* is used to store the private key of the CA certificate.
   - *conf* is used to store the configuration files used for simplifying parameters.
   - *server* is used to store the server certificate.
2. Create an *openssl.conf* file that contains the following information in the *conf* directory:

```
[ ca ]
default_ca = foo
[ foo ]
dir = /root/ca
database = /root/ca/index.txt
new_certs_dir = /root/ca/newcerts
certificate = /root/ca/private/ca.crt
serial = /root/ca/serial
private_key = /root/ca/private/ca.key
RANDFILE = /root/ca/private/.rand
default_days = 365
default_crl_days= 30
default_md = md5
unique_subject = no
policy = policy_any
[ policy_any ]
countryName = match
stateOrProvinceName = match
organizationName = match
organizationalUnitName = match
localityName = optional
commonName      = supplied
emailAddress    = optional
```

3. Run the following commands to generate a private key:

```
$ cd /root/ca
 $ sudo openssl genrsa -out private/ca.key
```

The following figure shows the command output.



4. Run the following command, enter the required information as prompted, and then press Enter to generate a *.csr* file.

```
$ sudo openssl req -new -key private/ca.key -out private/ca.csr
```

> **Note** Common Name specifies the domain name of the SLB instance.

```
root@iZbp1hfvivcqx1jbwap31iZ:~/ca# sudo openssl req -new -key private/ca.key -ou
t private/ca.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CN
State or Province Name (full name) [Some-State]:ZheJiang
Locality Name (eg, city) []:HangZhou
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Alibaba
Organizational Unit Name (eg, section) []:Test
Common Name (e.g. server FQDN or YOUR name) []:mydomain
Email Address []:a@alibaba.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
root@iZbp1hfvivcqx1jbwap31iZ:~/ca#
```

5. Run the following command to generate a *.crt* file:

```
$ sudo openssl x509 -req -days 365 -in private/ca.csr -signkey private/ca.key -out priv
ate/ca.crt
```

6. Run the following command to set the initial sequence number of the CA key. The key can be any four characters:

```
$ sudo echo FACE > serial
```

7. Run the following command to create a CA key library:

```
$ sudo touch index.txt
```

8. Run the following command to create a certificate revocation list for removing the client certificate:

```
$ sudo openssl ca -gencrl -out /root/ca/private/ca.crl -crldays 7 -config "/root/ca/con
f/openssl.conf"
```

Output:

```
Using configuration from /root/ca/conf/openssl.conf
```

Sign the client certificate

1. Run the following command to create the *users* directory under the *ca* directory to store client keys:

```
$ sudo mkdir users
```

2. Run the following command to create a client key:

```
$ sudo openssl genrsa -des3 -out /root/ca/users/client.key 1024
```

> **Note** When you create the key, enter a passphrase as the key password to prevent unauthorized use if the key leaks. Enter the same password twice.

3. Run the following command to create a *.csr* file for the client key:

```
$ sudo openssl req -new -key /root/ca/users/client.key -out /root/ca/users/client.csr
```

Enter the passphrase in the previous step and other required information as prompted.

> **Note** A challenge password is the password of the client certificate. Note that the challenge password is not the password of the client key.

4. Run the following command to use the CA key to sign the client key:

```
$ sudo openssl ca -in /root/ca/users/client.csr -cert /root/ca/private/ca.crt -keyfile
/root/ca/private/ca.key -out /root/ca/users/client.crt -config "/root/ca/conf/openssl.c
onf"
```

Enter *y* when you are prompted to confirm the following two operations.

```
root@iZbp1hfvivcqx1jbwap31iZ:~/ca# sudo openssl ca -in /root/ca/users/client.csr
 -cert /root/ca/private/ca.crt -keyfile /root/ca/private/ca.key -out /root/ca/us
ers/client.crt -config "/root/ca/conf/openssl.conf"
Using configuration from /root/ca/conf/openssl.conf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'CN'
stateOrProvinceName   :ASN.1 12:'ZheJiang'
localityName          :ASN.1 12:'HangZhou'
organizationName      :ASN.1 12:'Alibaba'
organizationalUnitName:ASN.1 12:'Test'
commonName            :ASN.1 12:'mydomain'
emailAddress          :IA5STRING:'a@alibaba.com'
Certificate is to be certified until Jun  4 15:28:55 2018 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
root@iZbp1hfvivcqx1jbwap31iZ:~/ca#
```

5. Run the following command to convert the certificate to a *PKCS12* file.

```
$ sudo openssl pkcs12 -export -clcerts -in /root/ca/users/client.crt -inkey /root/ca/us
ers/client.key -out /root/ca/users/client.p12
```

Enter the passphrase of the client key as prompted and press Enter. Then, enter the password used to export the client certificate. This password is used to protect the client certificate and is required when the client certificate is installed.

6. Run the following commands to view the generated client certificate:

```
cd users
ls
```

8.5. Convert certificate formats

Server Load Balancer (SLB) supports only certificates in the PEM format. Certificates in other formats must be converted to the PEM format before they can be uploaded to SLB. We recommend that you use OpenSSL to convert certificate formats.

Convert a certificate from DER to PEM

DER: This format is usually used on the Java platform. In most cases, the certificate file suffix is *.der*, *.cer*, or *.crt*.

- Run the following command to convert the certificate format:

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

- Run the following command to convert the private key:

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

Convert a certificate from P7B to PEM

P7B: This format is usually used in Windows Server and Tomcat.

Run the following command to convert the certificate format:

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

Convert a certificate from PFX to PEM

PFX: This format is usually used in Windows Server.

- Run the following command to extract the certificate:

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

- Run the following command to extract the private key:

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

8.6. Replace a certificate

This topic describes how to replace a certificate with a new certificate. We recommend that you replace certificates before they expire to avoid impacts on your service.

Procedure

1. Create and upload a new certificate.

   For more information, see Upload certificates.
2. Configure the new certificate for an HTTPS listener

   For more information, see Add an HTTPS listener.
3. On the **Certificates** page, find the expired certificate and click **Delete**.
4. In the message that appears, click **OK**.