Elastic Load Balance

FAQs

Issue 01

Date 2025-02-28

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Popular Questions

2 Service Abnormality

2.1 What Can I Do If There Is Packet Loss?

Scenarios

This FAQ provides guidance for you to troubleshoot packet loss.

Troubleshooting

Table 2-1 Check items

Check Item	What to Do
Access Control Blacklist or Whitelist	Check whether client requests are blocked by access control policies.
The Maximum Number of Concurrent Connections That IP as Backend Servers Can Handle	Check whether the maximum number of concurrent connections that IP as backend servers can handle is less than the number defined by the load balancer specifications.
Backend Server Connectivity	Check whether backend servers are available by running the curl command instead of the ping command.

Access Control Blacklist or Whitelist

- Symptom: Some specific IP addresses are blocked.
- Analysis: The IP addresses may be blocked by a blacklist or whitelist.
  - A blacklist: Packets from the IP addresses in the blacklisted IP address group will be denied to access the load balancer.

  - A whitelist: Packets from the IP addresses not in the whitelisted IP address group will be denied to access the load balancer.
- Solution: You can view the monitoring metrics **Blocked Packets** and **Blocked Traffic** to check whether there are packets blocked by blacklist or whitelist. For details about the metrics, see **Monitoring ELB Resources**. If there are blocked packets, check whether the clients are blocked by the blacklist or whitelist.

The Maximum Number of Concurrent Connections That IP as Backend Servers Can Handle

- Symptom: When ELB is used to route traffic across IP as backend servers, the number of concurrent connections does not reach the upper limit, but there are still access failures or dropped packets.
- Analysis: An IP as backend server can handle no more than 100,000 concurrent connections. If there is only a small number of IP as backend servers, the maximum number of concurrent connections that these servers can handle may be less than the number defined by the load balancer specifications.
- Solution: You can use a formula to calculate the maximum number of concurrent connections that can be handled by the IP as backend servers associated with a load balancer.

  Assume that your load balancer is running in two AZs and has ten IP as backend servers. You can use either of the following formulas to calculate the maximum number of concurrent connections:

  - A public network load balancer: 100,000 concurrent connections × Number of IP as backend servers × Number of AZs

    In this example, the maximum number of concurrent connections is calculated as: 100,000 × 10 × 2 = 2,000,000.
  - A private network load balancer that is in the same AZ as the clients:

    100,000 concurrent connections × Number of IP as backend servers

    In this example, the maximum number of concurrent connections is calculated as: 100,000 × 10= 1,000,000.

  If the maximum number of concurrent connections allowed is less than the number defined by the load balancer specifications, you can add more IP as backend servers.

Backend Server Connectivity

- Symptom: The IP address of a load balancer can be pinged, but there are still access failures or dropped packets.
- Analysis: If the load balancer is working normally, it returns ICMP Echo to the ping command. However, ping packets are not forwarded to backend servers, so the ping command output does not reflect the actual status of backend servers.
- Solution: Check whether backend servers are available by running the curl command.

2.2 Why Can't I Access My Backend Servers Through a Load Balancer?

Symptom

This FAQ provides guidance for you to troubleshoot the following problems:

- Backend servers cannot be accessed through a load balancer.
- You can access the load balancer from a private IP address, but not from a public IP address.
- Backend servers are considered unhealthy.

Background

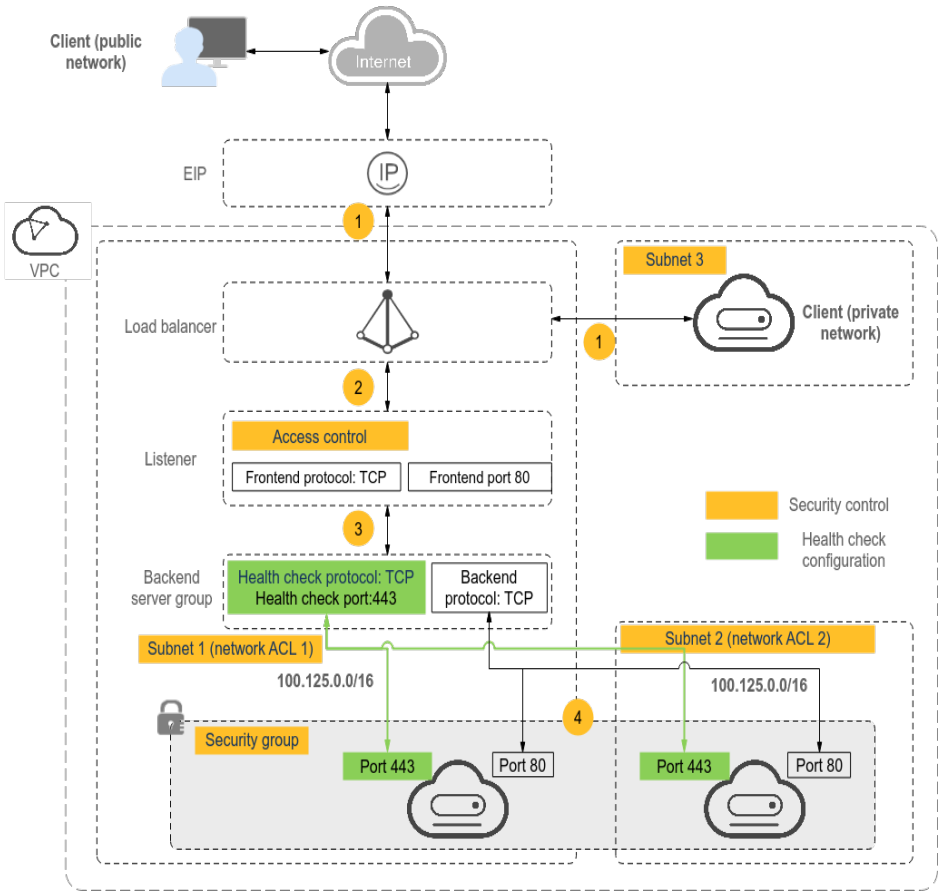
Figure 2-1 shows how clients access backend servers through a load balancer.

1. The public network load balancer uses an EIP to receive traffic over the Internet, while the private network load balancer receives traffic from within the VPC.
2. The load balancer receives incoming traffic using the frontend protocol and port configured for the listener.
3. The listener checks the health of backend servers. Only healthy backend servers can receive traffic from the listener.
4. The listener forwards the traffic to backend servers based on their weights and the listening rules.

Generally, the problem is probably caused by an access control issue (the parts in yellow) or a health check setting (the green parts).

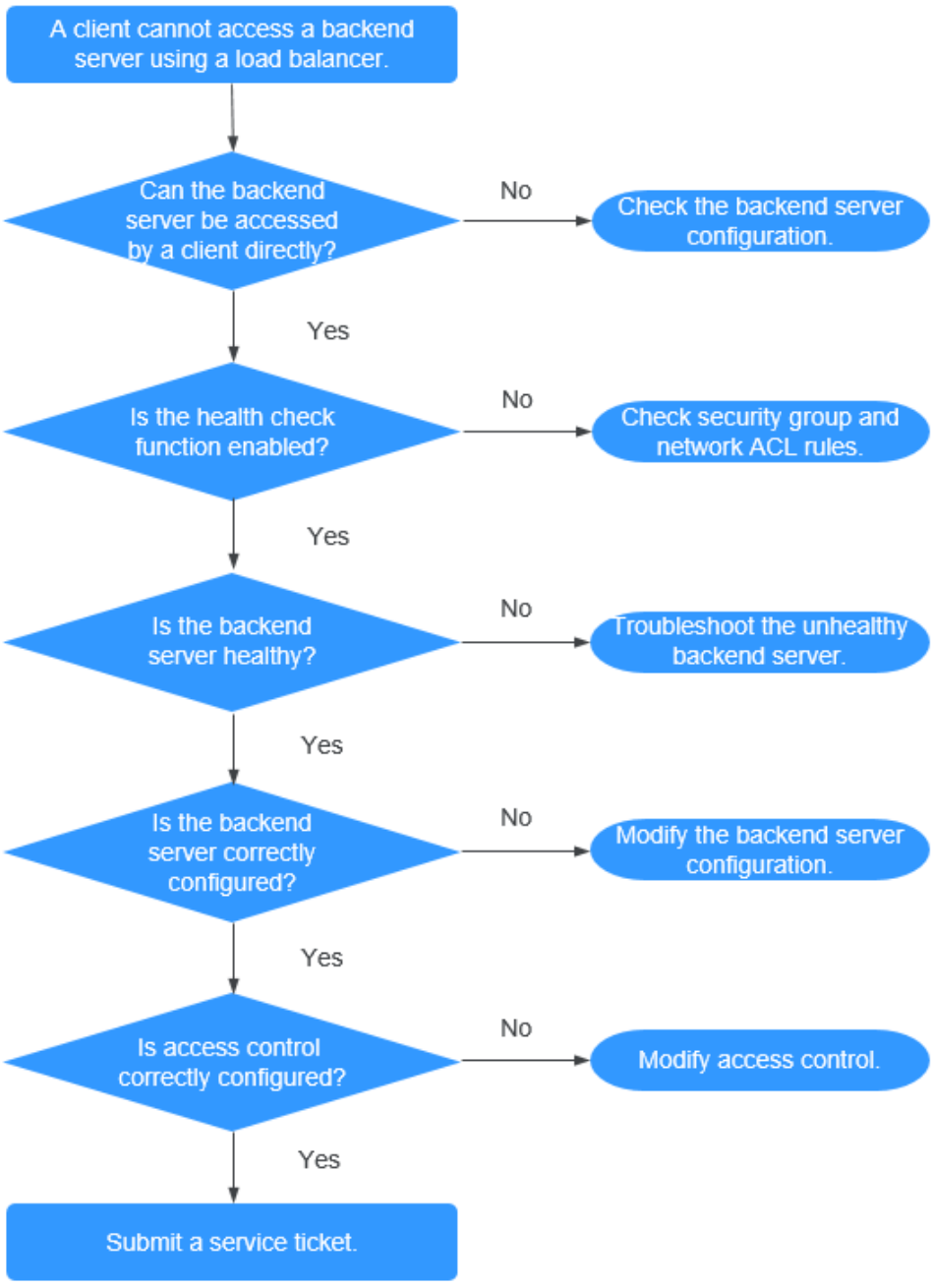
Troubleshooting should start with backend servers, then move on to the load balancer, and finally to the clients.

Figure 2-1 How clients access backend servers through a load balancer



Troubleshooting Process

Figure 2-2 Troubleshooting process



1. **Check whether the backend server can be accessed directly.** Use the client to access the backend server and verify that the backend server configuration and application configuration are correct.
2. **Check whether the health check is enabled on the console.**
3. **Check whether the health check result of the backend server on the console.** If the backend server is unhealthy, the load balancer will not route traffic to it.

4. **Check whether the weight and port of the backend server are correctly configured on the console.**
5. **Check whether access control is enabled and the IP address of the client is allowed to access the listener on the console.**

Step 1: Check Whether the Backend Server Can Be Accessed Directly

Use a client to access the backend server to determine whether the fault is caused by the load balancer or backend server. To do so, ensure that the network ACL rules allow communications between the client and backend server.

- Clients on the public network: Bind an EIP to the backend server. After the verification is complete, release the EIP.
- Clients on the private network: No EIP is required. If the client is in another VPC, set up a VPC peering connection.

If the fault persists, go to **Step 2: Check Whether the Health Check Is Enabled**.

Step 2: Check Whether the Health Check Is Enabled

If the client can access the backend server directly, check whether the health check is enabled. If the health check is enabled but the backend server is detected unhealthy, the load balancer will not route traffic to it.

1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click  in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. Click the name of the load balancer.
5. On the **Listeners** tab, check whether the health check is enabled.
   - If the health check is enabled, go to **Step 3: Check Whether the Backend Server Is Healthy**.
   - If the health check is not enabled:
     - Shared load balancers: Check whether the security group rules of the backend servers and network ACL rules allow traffic from 100.125.0.0/16.
     - Dedicated load balancers: Check whether the backend security group rules allow access from the VPC CIDR block where the ELB backend subnet works.

     This CIDR block is used by ELB to access backend servers and has no security risks. If traffic is allowed but the fault persists, go to **Step 4: Check Whether the Backend Server Configuration Is Correct**.

CAUTION

- Shared load balancers: If **Transfer Client IP Address** is enabled for a TCP or UDP listener, there is no need to configure security group rules and Network ACL rules to allow traffic from 100.125.0.0/16 and client IP addresses to backend servers.
- Dedicated load balancers: If **IP as a Backend** is not enabled for a load balancer that has a TCP or UDP listener, there is no need to configure security group rules and Network ACL rules to allow traffic from the backend subnet where the load balancer is deployed to the backend servers.

Step 3: Check Whether the Backend Server Is Healthy

If the health check is enabled but the backend server is detected unhealthy, the load balancer will not route traffic to it.

- If the backend server is unhealthy, rectify the fault by referring to **How Do I Troubleshoot an Unhealthy Backend Server?**
- If the backend server is healthy, go to **Step 4: Check Whether the Backend Server Configuration Is Correct**.

If the fault persists, go to **Step 4: Check Whether the Backend Server Configuration Is Correct**.

Step 4: Check Whether the Backend Server Configuration Is Correct

1. Choose **Backend Server Groups** > **Backend Servers** to view the backend server parameters:
   - **Weight**: If the weight is set to 0, traffic will not be forwarded to the server.
   - **Backend port**: It must be the same as the port used by the backend server.
2. On the **Listeners** tab, locate the TCP or UDP listener and check whether **Transfer Client IP Address** is enabled.
   - If this function is enabled, the load balancer uses the IP address of the client to access the backend server. In this case, configure security group and network ACL rules to allow access from this IP address.

     In addition, if this function is enabled, a server cannot be used as both the client and the backend server. This is because the backend server determines that the packet is sent by a local host based on the source IP address and will not return the response packet to the load balancer.
   - If this function is disabled, verify that the security group allows traffic from the corresponding IP address range to the backend server.
     - Dedicated load balancers: Ensure that the security group allows traffic from the backend subnet where the dedicated load balancer resides to the backend server.
     - Shared load balancers: Ensure that the security group allows traffic from 100.125.0.0/16 to the backend server.

If the fault persists, go to **Step 5: Check Whether Access Control Is Enabled**.

Step 5: Check Whether Access Control Is Enabled

On the **Summary** tab of the listener, check whether access control is enabled and the client is allowed to access the listener.

Submit a Service Ticket

If the problem persists, **submit a service ticket**.

2.3 Why Does a Server Occasionally Time Out When a Client Access It Through Different Load Balancers or Different Listeners of a Load Balancer?

Symptom

Table 2-2 Symptom description

Scenarios	• A client accesses the same backend server through different Layer 4 load balancers. • A client accesses the same backend server through different Layer 4 listeners of a load balancer.
Symptom	The client fails to receive responses from the backend server within the timeout duration.
Cause	The backend server received packets with the same 5-tuple.

5-tuple Conflicts

There will be 5-tuple conflicts if the following happens:

- The source port is hard coded as a fixed one, for example, 80, when the client accesses different load balancers.
- The client OS automatically selects the same source port when the client accesses different load balancers (different IP addresses).

Figure 2-3 5-tuple conflicts

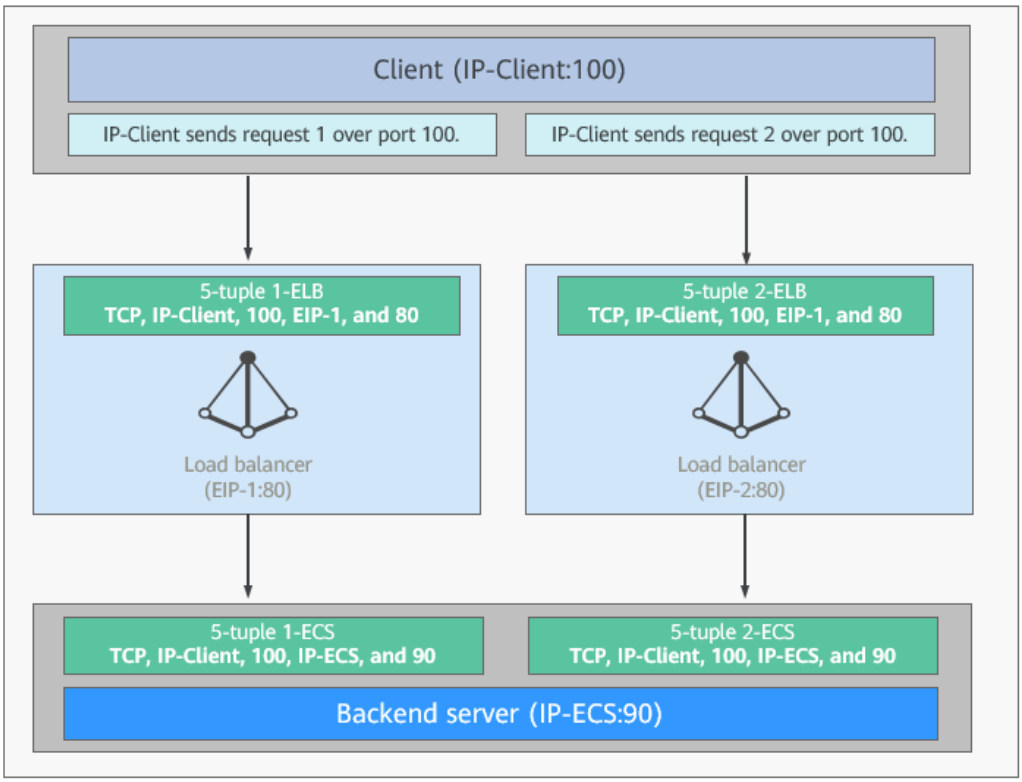


Figure 2-3 shows how two load balancers (with **Transfer Client IP Address** enabled) forward client requests to the same backend server over the same source port.

1. The client (IP-Client) uses the same source port (port 100) to send TCP request 1 to EIP1 (12.*xx.xx.xx*) and TCP request 2 to EIP2 (13.*xx.xx.xx*) bound to two load balancers.
2. The two TCP listeners forward client requests over the same port (port 80), and the two load balancers receive the 5-tuples as follows:

   Table 2-3 5-tuples received by the two load balancers

Name	Protocol	Source IP Address	Source Port	Destination IP Address	Destination Port
5-tuple 1-ELB	TCP	IP-Client	100	12.xx.xx.xx	80
5-tuple 2-ELB	TCP	IP-Client	100	13.xx.xx.xx	80

3. Each load balancer forwards the client requests to the same backend server (IP-ECS) over port 90.
4. **Transfer Client IP Address** has been enabled for the two TCP listeners, so the source IP address and source port are not changed. In this case, IP-ECS receives the 5-tuples from load balancers as follows:

Table 2-4 5-tuples received by IP-ECS

Name	Protocol	Source IP Address	Source Port	Destination IP Address	Destination Port
5-tuple 1-ECS	TCP	IP-Client	100	IP-ECS	90
5-tuple 2-ECS	TCP	IP-Client	100	IP-ECS	90

5. IP-ECS receives requests 1 and 2 with the same 5-tuple, causing connection failures.

Solution

You can solve the problem by taking the following measures:

- If the source port is hard coded as a fixed port, change the configuration so that the client can access different load balancers over different fixed ports.
- If the source port is automatically selected by the OS:
  - Allow the client to resend requests using different source ports to load balancers.
  - Add IP addresses as backend servers if your service is sensitive to reset connections. If **IP as a Backend** is enabled, ELB uses FullNAT to forward requests and **Transfer Client IP Address** will not take effect. In this case, the source IP address is changed to an IP address in the backend subnet of the load balancer. If you want to transfer the source IP address of the client, **Configuring the TOA Plug-in**.
- If you use different listeners to forward client requests, configure listeners to route requests to different backend servers.

2.4 What Can I Do If ELB Can't Be Accessed or Traffic Routing is Interrupted?

1. Check the health of backend servers. If a backend server is unhealthy, traffic will be routed to other healthy servers. Rectify the health check fault and access ELB again.
2. Check whether the security group rules allow access from the corresponding IP address range.
   - Dedicated load balancers: Check whether the security group containing the backend server has inbound rules to allow traffic from the backend subnet where the load balancer is deployed.
   - Shared load balancers: Check whether the security group containing the backend server has inbound rules to allow traffic from 100.125.0.0/16.

CAUTION

- Shared load balancers: If **Transfer Client IP Address** is enabled for a TCP or UDP listener, there is no need to configure security group rules and Network ACL rules to allow traffic from 100.125.0.0/16 and client IP addresses to backend servers.
- Dedicated load balancers: If **IP as a Backend** is not enabled for a load balancer that has a TCP or UDP listener, there is no need to configure security group rules and Network ACL rules to allow traffic from the backend subnet where the load balancer is deployed to the backend servers.

3. Check whether a TCP connection is established between the load balancer and the client. The timeout duration for a TCP connection is 300s and cannot be changed. If the duration exceeds 300s, the load balancer sends an RST message to the client and the backend server to disconnect the connection.
4. Check whether sticky sessions are enabled and the sticky session type is set to source IP address. If yes, check whether the request IP address changes before the request reaches the load balancer.

   For example, if ELB is combined with Content Delivery Network (CDN) or Web Application Firewall (WAF), the IP address of the request changes when it passes through CDN or WAF. The IP address change causes session stickiness to fail. If you want to use CDN or WAF, it is recommended that you add an HTTP or HTTPS listener and configure cookie-based sticky sessions.
5. Check whether the listener is an HTTP or HTTPS listener and sticky sessions are enabled. If yes, check whether the request contains a cookie. Sticky sessions at Layer 7 are based on cookies. If the request contains a cookie, check whether the cookie value changes.
6. Check the stickiness duration configured for the backend server group. If sticky sessions are enabled, the default stickiness duration of the backend server group at Layer 4 and Layer 7 is 20 minutes. After the stickiness duration times out, the connection will be disconnected.
7. Check whether the servers you access are associated with a load balancer.

   If **Transfer Client IP Address** is enabled for TCP or UDP listeners, a cloud server cannot be used as both a backend server and a client.
8. Check whether you have added a backend server in a VPC that is different from the one where the load balancer is running, by using the server's IP address. If yes, check whether a VPC peering connection has been established between the two VPCs.
9. Check whether your account is in arrears. If your account is in arrears, resources such as EIPs will be frozen and cannot be used.

2.5 How Can I Handle Abnormal Status Codes?

Common abnormal status codes include 400, 403, 502, 504, and more. If any of these codes is returned, it is recommended that you access the backend server to check if it can respond properly.

If the backend server responds properly, rectify the fault by referring to **Table 2-5**. If the fault persists, contact the customer service.

Table 2-5 Common abnormal status codes

Status Code	Description	Possible Causes
400	Bad Request	• The client sent a malformed request that does not comply with the HTTP specification. • An HTTP request was sent to the HTTPS port. • The size of the request header exceeded 64 KB.
401	Unauthorized	Authentication on the backend server failed. (This status code is returned to the client by the backend server.)
403	Forbidden	The request was intercepted by the backend server. (This status code is returned to the client by the backend server.)
404	Not Found	• The backend server is abnormal or the application does not exist. (This status code is returned to the client by the backend server.) • The forwarding policy was incorrectly configured, and the request was not routed to the right backend server.
408	Request Timeout	The client did not send the request within the time that the server was configured to wait, which is 60s by default. Sending a TCP keepalive packet does not prevent this timeout.
413	Payload Too Large	The size of the request body sent by the client exceeded 10 GB.
414	Too long URI	The request URL or query string parameter sent by the client was too long.
499	Client Closed Request	The client disconnected from the load balancer before receiving a response from the load balancer. This status code is recorded only in access logs.
500	Internal Server Error	There was an internal error. (This status code is returned to the client by the backend server.)
501	Not Implemented	The load balancer failed to identify the request. The value of the **Transfer-Encoding** header field is not **chunked** or **identity**.

Status Code	Description	Possible Causes
502	Bad Gateway	• The port used by the backend server was incorrectly configured. • The load balancer received a TCP RST packet from the backend server when attempting to establish a connection with or sending data to the backend server. • The format of the response from the backend server was incorrect, or the response contained an invalid HTTP response header. • The backend server was incorrectly configured, for example, incorrect routes or network ACLs.
503	Service Unavailable	The application or backend server was unavailable. Generally, this status code is returned by the backend server.
504	Gateway Timeout	• During the first connection, the load balancer failed to connect to the backend server before the connection times out. (The default timeout is 5 seconds). • The load balancer established a connection with the backend server, but did not respond before the response timeout (which is 300s by default) elapsed. • The network ACL rules of the subnet did not allow the load balancer to access backend servers in the subnet.

2.6 How Do I Handle Abnormal Request Headers?

In addition to abnormal status codes, ELB notifies clients of typical exceptions through the **x-router-code** header. Common header values are **40000**, **40001**, **50000**, and **50001**. You can handle these headers based on the following table.

Table 2-6 Abnormal headers

Header Value	Meaning	Possible Causes
40000	The default backend server group has no backend servers.	• The client request does not match any forwarding policy. • No backend server is configured for the backend server groups.

Header Value	Meaning	Possible Causes
40001	The backend server group of the matched forwarding policy has no backend servers.	• An unexpected forwarding policy is matched by accident. • The backend server group of the matched forwarding policy has no backend servers.
50000	All backend servers are unavailable.	• The health check is not correctly configured. As a result, ELB considers that all backend servers are unhealthy. • All backend servers themselves are unavailable.
50001	The configuration is loading.	The configuration is not complete. Wait and try again later.

2.7 How Do I Check for Traffic Inconsistencies?

Check if there are failed requests on the clients, especially when *4xx* status codes are returned. One possible cause is that the requests are not being routed to backend servers because ELB considers these requests abnormal.

2.8 Why Does ELB Fail to Distribute Traffic Evenly?

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

NOTE

Generally, in addition to the load balancing algorithm, factors that affect load balancing include connection type, session stickiness, and server weights.

2.9 How Do I Check If There Is Excessive Access Delay?

1. Bind an EIP to a backend server to make the applications accessible from the Internet and then check the access delay. In this way, you can determine whether the problem is caused by the client, load balancer, or applications.
2. Check the incoming traffic. If the incoming traffic exceeds the EIP bandwidth, there may be congestion and packet loss.

   NOTE

   If the incoming traffic exceeds the available bandwidth, it does not mean that the bandwidth is fully used. In this case, you need perform further operations to locate the fault or increase the bandwidth.
3. Check the load and security policies of backend servers. If backend servers are heavily loaded or they have security policies configured, they cannot quickly respond to requests from the associated load balancer.
4. Check the **Unhealthy Servers** metric to view the health statuses of backend servers. If the applications are unstable and connections to the backend server time out, the retry mechanism will route the requests to another backend server. As a result, access to the applications will be successful but there will be more access delay.
5. If the problem persists, contact customer service.

2.10 What Do I Do If a Load Balancer Fails a Stress Test?

1. Check the load of backend servers. If their vCPU usage reaches 100%, applications may have performance bottlenecks.
2. Check the incoming traffic. If burst traffic exceeds the bandwidth set for the EIP, a large number of packets will be discarded and requests will not be responded to, thereby affecting the load balancer's performance.

   NOTE

   If burst traffic exceeds the available bandwidth, it does not mean that the bandwidth is fully used. In this case, you need perform further operations to locate the fault or increase the bandwidth.
3. Check the number of short connections in the **time_wait** state on the clients. One possible cause is that there are insufficient client ports.
4. The listening queue backlog of the backend servers may be full. If this happens, the backend server will not respond to SYN ACK packets. You can increase the maximum allowed of the backlog by adjusting parameter **net.core.somaxconn**.

2.11 How Do I Check If Sticky Sessions Failed to Take Effect?

1. Check whether sticky sessions are enabled for the backend server group. If sticky sessions are enabled, go to the next step.
2. Check the health check result of the backend server. If the health check result is **Unhealthy**, traffic is routed to other backend servers and sticky sessions become invalid.
3. If you select the source IP hash algorithm, check whether the IP address of the request changes before the load balancer receives the request.
4. If sticky sessions are enabled for an HTTP or HTTPS listener, check whether the request carries a cookie. If they are, check whether the cookie value changed (because load balancing at Layer 7 uses cookies to maintain sessions).

2.12 How Do I Check SSL/TLS Authentication Errors?

When you use an HTTPS or TLS listener, there may be errors in every step of the SSL/TLS authentication negotiation. Check the potential causes described below one by one.

This section uses Java as an example to describe how to identify the cause.

Potential Cause 1: No Valid Certificates

- Error message:
  ```
  Exception in thread "main" javax.net.ssl.SSLHandshakeException: Received fatal alert: handshake_failure
  ```
- Cause: The load balancer does not have a valid certificate for authenticating SSL/TLS handshake requests.
- Solutions:
  - Check whether the certificate configured for the listener is valid.
  - Check whether the cipher suite used by the TLS security policy of the listener meets the client requirements.

Potential Cause 2: Certificate Verification Failed

- Error message:
  ```
  Exception in thread "main" javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target
  ```
- Cause: The certificate chain may be incomplete or the certificate authority (CA) is not trusted.
- Solution: Replace the listener certificate with a valid one issued by a trusted CA.

Potential Cause 3: Mismatches Between the Returned and Requested Host Names

- Error message:
  ```
  Exception in thread "main" javax.net.ssl.SSLHandshakeException:
  java.security.cert.CertificateException: No name matching localhost found
  ```
- Cause: This error is commonly seen in two-way authentication scenarios. If the host name in the server certificate is different from the requested host name, the local host name verification fails.
- Solution: Check whether the client has a certificate that contains the local host name.

Potential Cause 4: Incorrect TLS Security Policy Version

- Error message:
  ```
  Exception in thread "main" javax.net.ssl.SSLHandshakeException: No appropriate protocol (protocol is disabled or cipher suites are inappropriate)
  ```
- Cause: The client and server cannot agree on a supported SSL/TLS protocol version or cipher suite.
- Solution: Check whether the TLS protocol version and cipher suite version of the TLS security policy used by the client match those used by the listener.

3 Health Checks

3.1 How Do I Troubleshoot an Unhealthy Backend Server of a Dedicated Load Balancer?

Symptom

If a client cannot access a backend server through a load balancer, the backend server is declared unhealthy. You can view the health check results for a backend server on the ELB console.

If a backend server is unhealthy, ELB will not route traffic to it until it is declared healthy again.

Background

To check the health of backend servers, dedicated load balancers use the IP addresses in the backend subnet to send heartbeat requests to the backend servers. For details about how a health check works, see **Health Check**.

- If health checks are disabled, the load balancer will consider the backend servers healthy by default and still route requests to them.
- When the weight of a backend server is set to 0:
  - Dedicated load balancer: Requests are not routed to this backend server, even if the backend server is healthy.
  - Shared load balancer: Requests are not routed to this backend server, and the health check result is unhealthy.

Troubleshooting

You can use ELB self-service troubleshooting to locate and fix unhealthy backend servers. If the faults persist, you can perform further checks based on **Table 3-1**.

Changes to the health check configuration will not be applied immediately, and the required time depends on the health check interval and timeout duration.

You can view the health check results in the backend server list of the load balancer.

Table 3-1 Troubleshooting

Method	Items
Self-service troubleshooting	**Checking Security Group Rules**
	Checking Network ACL Rules
	Checking the Health Check Configuration
Other check items	**Checking Whether the Backend Server Group Is Associated with a Listener**
	Checking Whether an EIP or Private IP Address Is Bound to the Load Balancer
	Checking the Backend Server
	Checking the Firewall on the Backend Server
	Checking the Backend Server Route
	Checking the Backend Server Load
	Checking the hosts.deny File

Checking Security Group Rules

The security group rules for backend servers must allow traffic from the backend subnet where the load balancer resides to the backend server through the health check protocol and port.

You can view the health check settings on the summary page of the backend server group and configure health checks on the summary page of the listener.

You can use ELB self-service troubleshooting to check the security group rules configured for backend servers based on **Table 3-2**.

Table 3-2 Checking security group rules

Check Item	Solution
The source configured for the inbound rule	Ensure that the inbound rules of the security group allow traffic from the backend subnet where the load balancer resides to the backend server through the health check protocol and port. For details, see **configuring security group rules (dedicated load balancers)**.
The port configured for the inbound rule	
The protocol configured for the inbound rule	

Check Item	Solution
The destination configured for the outbound rule	The default outbound security group rules allow all outbound traffic. If you configure outbound rules, ensure that the rules allow traffic from the backend server to the backend subnet where the load balancer resides through the health check protocol and port. For details, see **configuring security group rules (dedicated load balancers)**.
The port configured for the outbound rule	
The protocol configured for the outbound rule	

NOTE

If the load balancer has Layer 4 listeners and **IP as a Backend** is disabled, network ACL rules and security group rules will not work. You can use access control to specify which IP addresses are allowed to access the listeners. Learn how to configure access control.

Checking Network ACL Rules

Network ACL rules are optional for subnets. If network ACL rules are configured for the subnets where the backend servers are deployed, the network ACL rules must allow traffic from the backend subnet of the dedicated load balancer to the backend server subnets through the health check protocol and port.

Default network ACL rules deny all inbound and outbound traffic. You can configure an inbound rule to allow traffic from the backend subnet of the load balancer through the port of the backend server.

You can use ELB self-service troubleshooting to check the network ACL rules configured for backend server subnets based on **Table 3-3**.

Table 3-3 Checking network ACL rules

Check Item	Solution
The protocol configured for the inbound rule	Ensure that the inbound rules of the network ACL allow traffic from the backend subnet of the dedicated load balancer to the backend server subnets through the health check protocol and port. For details, see **Configuring Network ACL Rules**.
The source configured for the inbound rule	
The source port configured for the inbound rule	
The destination configured for the inbound rule	
The destination port configured for the inbound rule	

Check Item	Solution
The protocol configured for the outbound rule	Ensure that the outbound rules of the network ACL allow traffic from the backend server subnets to the backend subnet of the dedicated load balancer through the health check protocol and port. For details, see **Configuring Network ACL Rules**.
The source configured for the outbound rule	
The source port configured for the outbound rule	
The destination configured for the outbound rule	
The destination port configured for the outbound rule	

NOTE

If the load balancer has Layer 4 listeners and **IP as a Backend** is disabled, network ACL rules and security group rules will not work. You can use access control to specify which IP addresses are allowed to access the listeners. Learn how to configure access control.

Checking the Health Check Configuration

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
5. On the **Backend Server Groups** page, locate the backend server group and click its name.
6. On the **Summary** page, click **Health Check** on the right.

   On the **Configure Health Check** page, view the parameters in the following table. For details about how to set health check parameters, see **Modifying Health Check Settings**.

Table 3-4 Parameters for configuring a health check

Parameter	Description
Domain Name	If you use HTTP for health checks and the backend server is configured to verify the Host header, enter the domain name configured for the backend server.

Parameter	Description
Protocol	The outbound rules of the network ACL do not allow traffic over the health check protocol.
Port	Configure the backend port as the health check port by referring to **Modifying Health Check Settings**.
Path	If HTTP is used for health checks, you must check this parameter. A simple static HTML file is recommended.

NOTE

- If the health check protocol is HTTP and the health check port is normal, change the path or change the health check protocol to TCP.
- Enter an absolute path. The following are example paths:
  - If the URL is **http://www.example.com** or **http://192.168.63.187:9096**, enter **/** as the health check path.
  - If the URL is **http://www.example.com/chat/try/**, enter **/chat/try/** as the health check path.
  - If the URL is **http://192.168.63.187:9096/chat/index.html**, enter **/chat/index.html** as the health check path.

Checking Whether the Backend Server Group Is Associated with a Listener

Check whether the backend server group that has unhealthy backend servers is associated with a listener.

If the backend server group is not associated with a listener, health checks will fail.

If the backend server group has been associated with a listener, proceed with the following possible reasons.

Checking Whether an EIP or Private IP Address Is Bound to the Load Balancer

NOTE

Check this only when you add a TCP or UDP listener to the load balancer.

If you add a TCP or UDP listener to the load balancer, check whether the load balancer has a private IP address or an EIP bound.

When you create a load balancer for the first time, if no EIP or private IP address is bound to the load balancer, the health check result of backend servers associated with a TCP or UDP listener is **Unhealthy**.

Checking the Backend Server

NOTE

If the backend server runs on Windows, use a browser to access **https://***{Backend server IP address}*:*{Health check port}*. If a 2xx or 3xx code is returned, the backend server is running normally.

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

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

If the health check port and **LISTEN** are displayed, the health check port is in the listening state. As shown in **Figure 3-1**, TCP port 880 is listened on.

If you do not specify a health check port, backend ports are used as the health check port by default.

Figure 3-1 Backend server port listened on

Figure 3-2 Backend server port not listened on

If the health check port is not in the listening state, the backend server is not listened on. You need to start the application on the backend server and check whether the health check port is listened on.

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

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

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

TCP listeners: 200

200 for HTTP/HTTPS health checks

Figure 3-3 Unhealthy backend server

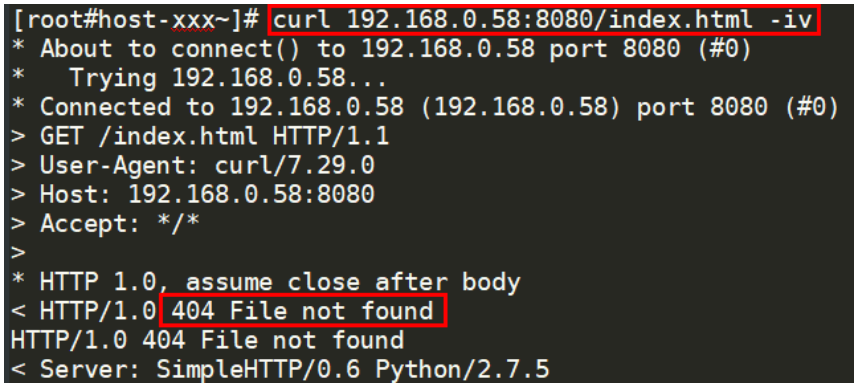
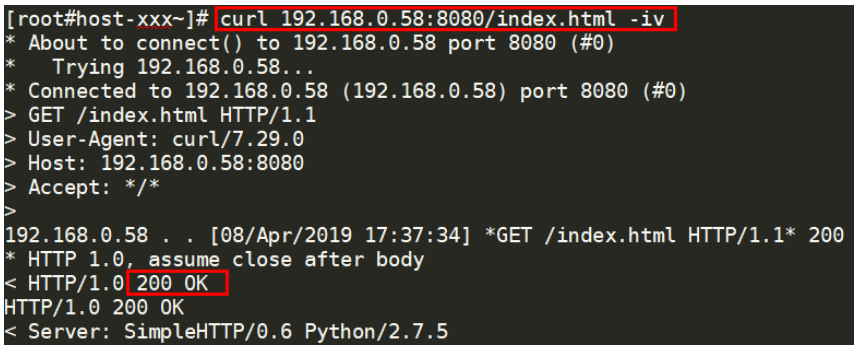


Figure 3-4 Healthy backend server



- If HTTP is used for health checks and the backend server is detected unhealthy, take the following step to configure a TCP health check:

  On the **Listeners** tab, modify the target listener, select the backend server group for which TCP health check has been configured, or add a backend server group and select TCP as the health check protocol. After you complete the configuration, wait for a while and check the health check result.

Checking the Firewall on the Backend Server

If the firewall or other security software is enabled for the backend server, the software may block the IP addresses from the backend subnet where the load balancer works.

Configure inbound firewall rules to allow traffic from the backend subnet where the load balancer works to backend servers.

Checking the Backend Server Route

Check whether the default route configured for the primary NIC (for example, eth0) has been manually modified. If the default route is changed, health check packets may fail to reach the backend server.

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

```
ip route
```

Alternatively, run the following command:

```
route -n
```

Figure 3-5 shows the command output when the backend server route is normal.

Figure 3-5 Example default route pointing to the gateway

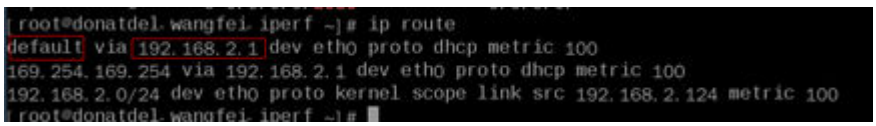
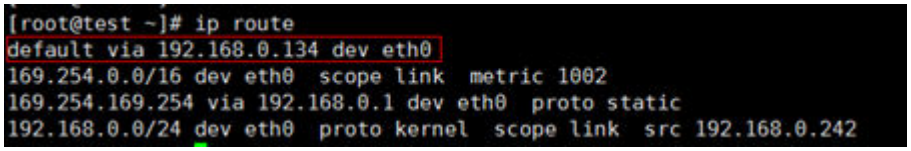


Figure 3-6 Example default route not pointing to the gateway



If the command output does not contain the first route, or the route does not point to the gateway, configure or modify the default route to point to the gateway.

Checking the Backend Server Load

View the vCPU usage, memory usage, network connections of the backend server on the Cloud Eye console to check whether the backend server is overloaded.

If the workload is high, connections or requests for health checks may time out.

Checking the hosts.deny File

Verify that IP addresses in backend subnet where the load balancer works are not written to the **/etc/hosts.deny** file on the backend servers.

Submitting a Service Ticket

If the problem persists, **submit a service ticket**.

3.2 How Do I Troubleshoot an Unhealthy Backend Server of a Shared Load Balancer?

Symptom

If a client cannot access a backend server through a load balancer, the backend server is declared unhealthy. You can view the health check results for a backend server on the ELB console.

If a backend server is unhealthy, ELB will not route traffic to it until it is declared healthy again.

Background

To check the health of backend servers, shared load balancers use the IP addresses in the backend subnet to send heartbeat requests to the backend servers. For details about how a health check works, see **Health Check**.

- If health checks are disabled, the load balancer will consider the backend servers healthy by default and still route requests to them.
- When the weight of a backend server is set to 0:
  - Dedicated load balancer: Requests are not routed to this backend server, even if the backend server is healthy.
  - Shared load balancer: Requests are not routed to this backend server, and the health check result is unhealthy.

Troubleshooting

You can use ELB self-service troubleshooting to locate and fix unhealthy backend servers. If the faults persist, you can perform further checks based on **Table 3-5**.

Changes to the health check configuration will not be applied immediately, and the required time depends on the health check interval and timeout duration.

You can view the health check results in the backend server list of the load balancer.

Table 3-5 Troubleshooting

Method	Items
Self-service troubleshooting	**Checking Security Group Rules**
	Checking Network ACL Rules
	Checking the Health Check Configuration
Other check items	**Checking Whether the Backend Server Group Is Associated with a Listener**
	Checking Whether an EIP or Private IP Address Is Bound to the Load Balancer
	Checking the Backend Server
	Checking the Firewall on the Backend Server
	Checking the Backend Server Route
	Checking the Backend Server Load
	Checking the hosts.deny File

Checking Security Group Rules

The security group rules for backend servers must allow traffic from the IP addresses in 100.125.0.0/16 to the backend server through the health check protocol and port.

You can configure and view the health check on the summary page of a backend server group. Shared load balancers use the IP addresses in 100.125.0.0/16 for health checks.

You can use ELB self-service troubleshooting to check the security group rules configured for backend servers based on **Table 3-6**.

Table 3-6 Checking security group rules

Check Item	Solution
The source configured for the inbound rule	Ensure that the inbound rules of the security group allow traffic from the IP addresses in 100.125.0.0/16 to the backend server through the health check protocol and port. For details, see **configuring security group rules (shared load balancers)**.
The port configured for the inbound rule	
The protocol configured for the inbound rule	
The destination configured for the outbound rule	The default outbound security group rules allow all outbound traffic. Ensure that the outbound rules of the security group allow traffic from the backend server to the IP addresses in 100.125.0.0/16 through the health check protocol and port. For details, see **configuring security group rules (shared load balancers)**.
The port configured for the outbound rule	
The protocol configured for the outbound rule	

NOTE

If **Transfer Client IP Address** is enabled for Layer 4 listeners, network ACL rules and security group rules will not work. You can use access control to specify which IP addresses are allowed to access the listeners. Learn how to configure access control.

Checking Network ACL Rules

Network ACL rules are optional for subnets. If network ACL rules are configured for the subnets where the backend servers are deployed, the network ACL rules must allow traffic from the backend subnet of the dedicated load balancer to the backend server subnets through the health check protocol and port.

Default network ACL rules deny all inbound and outbound traffic. You can configure an inbound rule to allow traffic from the backend subnet of the load balancer through the port of the backend server.

You can use ELB self-service troubleshooting to check the network ACL rules configured for backend server subnets based on **Table 3-7**.

Table 3-7 Checking network ACL rules

Check Item	Solution
The protocol configured for the inbound rule	Ensure that the inbound rules of the network ACL allow traffic from IP addresses in 100.125.0.0/16 to the backend server subnets through the health check protocol and port. For details, see **Configuring Network ACL Rules (Shared Load Balancer)**.
The source configured for the inbound rule	

Check Item	Solution
The source port configured for the inbound rule	
The destination configured for the inbound rule	
The destination port configured for the inbound rule	
The protocol configured for the outbound rule	Ensure that the outbound rules of the network ACL allow traffic from the backend server subnets to the IP addresses in 100.125.0.0/16 through the health check protocol and port. For details, see **Configuring Network ACL Rules (Shared Load Balancer)**.
The source configured for the outbound rule	
The source port configured for the outbound rule	
The destination configured for the outbound rule	
The destination port configured for the outbound rule	

NOTE

If **Transfer Client IP Address** is enabled for Layer 4 listeners, network ACL rules and security group rules will not work. You can use access control to specify which IP addresses are allowed to access the listeners. Learn how to configure access control.

Checking the Health Check Configuration

1. In the navigation pane on the left, choose **Elastic Load Balance** > **Backend Server Groups**.
2. On the **Backend Server Groups** page, locate the backend server group and click its name.
3. On the **Summary** page, click **Health Check** on the right.

   On the **Configure Health Check** page, view the parameters in the following table. For details about how to set health check parameters, see **Modifying Health Check Settings**.

Table 3-8 Parameters for configuring a health check

Parameter	Description
Domain Name	If you use HTTP for health checks and the backend server is configured to verify the Host header, enter the domain name configured for the backend server.
Protocol	The outbound rules of the network ACL do not allow traffic over the health check protocol.
Port	Use the backend port as the health check port.
Path	If HTTP is used for health checks, you must check this parameter. A simple static HTML file is recommended.

NOTE

- If the health check protocol is HTTP and the health check port is normal, change the path or change the health check protocol to TCP.
- Enter an absolute path. The following are example paths:
  - If the URL is **http://www.example.com** or **http://192.168.63.187:9096**, enter **/** as the health check path.
  - If the URL is **http://www.example.com/chat/try/**, enter **/chat/try/** as the health check path.
  - If the URL is **http://192.168.63.187:9096/chat/index.html**, enter **/chat/index.html** as the health check path.

Checking Whether the Backend Server Group Is Associated with a Listener

Check whether the backend server group that has unhealthy backend servers is associated with a listener.

If the backend server group is not associated with a listener, health checks will fail.

If the backend server group has been associated with a listener, proceed with the following possible reasons.

Checking Whether an EIP or Private IP Address Is Bound to the Load Balancer

NOTE

- Check this only when you add a TCP or UDP listener to the load balancer.

If you add a TCP or UDP listener to the load balancer, check whether the load balancer has a private IP address or an EIP bound.

When you create a load balancer for the first time, if no EIP or private IP address is bound to the load balancer, the health check result of backend servers associated with a TCP or UDP listener is **Unhealthy**.

Checking the Backend Server

NOTE

If the backend server runs on Windows, use a browser to access **https://***{Backend server IP address}*:*{Health check port}*. If a 2xx or 3xx code is returned, the backend server is running normally.

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

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

If the health check port and **LISTEN** are displayed, the health check port is in the listening state. As shown in **Figure 3-7**, TCP port 880 is listened on.

If you do not specify a health check port, backend ports are used as the health check port by default.

Figure 3-7 Backend server port listened on

Figure 3-8 Backend server port not listened on

If the health check port is not in the listening state, the backend server is not listened on. You need to start the application on the backend server and check whether the health check port is listened on.

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

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

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

TCP listeners: 200

200, 202, or 401 for HTTP health checks

Figure 3-9 Unhealthy backend server

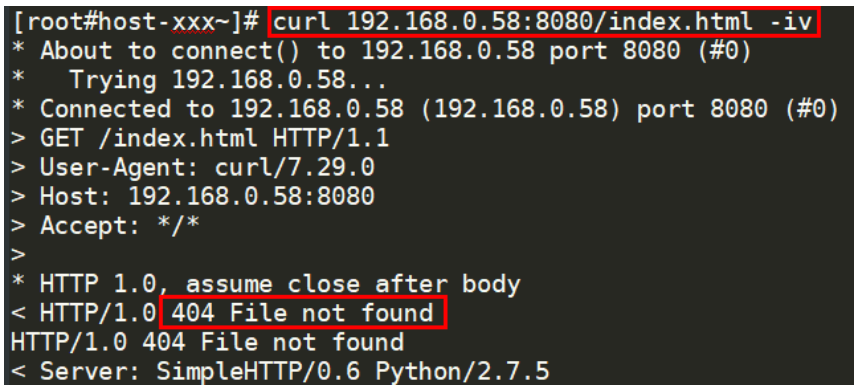


Figure 3-10 Healthy backend server

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

- If the HTTP health check is abnormal, configure a TCP health check. The procedure is described as follows:

  On the **Listeners** tab, modify the target listener, select the backend server group for which TCP health check has been configured, or add a backend server group and select TCP as the health check protocol. After you complete the configuration, wait for a while and check the health check result.

Checking the Firewall on the Backend Server

If the firewall or other security software is enabled for the backend server, the software may block the IP addresses from the backend subnet where the load balancer works.

Configure inbound firewall rules to allow traffic from IP addresses in 100.125.0.0/16 to backend servers.

Checking the Backend Server Route

Check whether the default route configured for the primary NIC (for example, eth0) has been manually modified. If the default route is changed, health check packets may fail to reach the backend server.

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

```
ip route
```

Alternatively, run the following command:

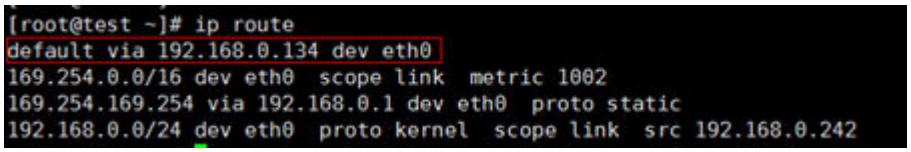
```
route -n
```

Figure 3-11 shows the command output when the backend server route is normal.

Figure 3-11 Example default route pointing to the gateway

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


Figure 3-12 Example default route not pointing to the gateway



If the command output does not contain the first route, or the route does not point to the gateway, configure or modify the default route to point to the gateway.

Checking the Backend Server Load

View the vCPU usage, memory usage, network connections of the backend server on the Cloud Eye console to check whether the backend server is overloaded.

If the workload is high, connections or requests for health checks may time out.

Checking the hosts.deny File

Verify that IP addresses in 100.125.0.0/16 are not written to the **/etc/hosts.deny** file on the backend servers.

Submitting a Service Ticket

If the problem persists, **submit a service ticket**.

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

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

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

3.4 How Does ELB Perform UDP Health Checks? What Are the Precautions for UDP Health Checks?

How UDP Health Checks Work

UDP is a connectionless protocol. A UDP health check is implemented as follows:

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

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

When you use UDP for health checks, retain default parameter settings.

Troubleshooting

If the backend server is unhealthy, use either of the following methods to locate the fault:

- Check whether the timeout duration is too short.

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

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

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

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

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

The default rate limit is **1000**.

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

The default rate mask is **6168**.

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

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

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

```
man 7 icmp
```

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

NOTE

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

Precautions

Note the following when you configure UDP health checks:

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

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

  cat /proc/sys/net/ipv4/icmp_echo_ignore_all

  - If the returned value is **1**, ICMP is disabled.
  - If the returned value is **0**, ICMP is enabled.
- The health check result may be different from the actual health of the backend server.

  If the backend server runs Linux, the rate of ICMP packets may be limited due to Linux's defense against ping flood attacks when there is a large number of concurrent requests. In this case, if a service exception occurs, the load balancer will not receive error message **port XX unreachable** and will consider the health check to be successful. As a result, there is an inconsistency between the health check result and the actual server health.

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

ELB is deployed in clusters, and all nodes for request forwarding in the cluster send requests to backend servers at the same time. If the health check interval is too short, health checks are performed once every few seconds, and a large number of packets are sent to backend servers. To control the frequency of access to backend servers, change the health check interval by referring to *Configuring a Health Check*.

3.6 When Does a Health Check Start?

After a backend server is added to a backend server group, the health check is performed at a random time during the first interval and then at the specified interval.

3.7 What Do I Do If a Lot of Access Logs Are Generated During Health Checks?

1. You can increase the health check interval by referring to Changing the Health Check Configurations.

   Risk: After the health check interval is prolonged, the time for the load balancer to detect unhealthy servers will increase.
2. You can disable the health check by referring to Disabling a Health Check.

   Risk: After health checks are disabled, the load balancer will not check the backend servers. If a backend server becomes faulty, the load balancer will still route requests to this server.

3.8 What Status Codes Will Be Returned If Backend Servers Are Identified as Healthy?

Table 3-9 Health check status codes

Load Balancer Type	Health Check Protocol	Status Code
Dedicated load balancers	HTTP	200
	HTTPS	200
Shared load balancers	HTTP	• 200 • 202 • 401

4 ELB Functionality

4.1 Can Load Balancers Be Used Separately?

No, load balancers must be used together with other necessary components: listeners and backend server groups.

ELB automatically distributes incoming traffic across multiple backend servers based on the listening rules you configure. It expands the service capabilities of your applications and improves their availability by eliminating single points of failure (SPOFs). To use a load balancer to distribute traffic, you must add at least one listener to and associate backend servers (such as ECSs) with this load balancer.

4.2 Can ELB Block DDoS Attacks and Secure Web Code?

- ELB does not provide security functions such as blocking DDoS attacks.
- Anti-DDoS is enabled for cloud services by default, and all incoming traffic on the public network is protected.

NOTE

You can also use Advanced Anti-DDoS (AAD), an advanced version of Anti-DDoS. AAD provides high-defense IP addresses to hide the origin server IP addresses, so that your applications can weather larger and more sophisticated DDoS attacks, ensuring service continuity. You can configure a DNS record to map the origin server IP addresses to high-defense addresses for diverting malicious attack traffic, protecting the origin servers against attacks and preventing interruptions to your workloads. This service can be deployed on hosts used on Huawei Cloud, other clouds, and on-premises data centers.

4.3 What Types of APIs Does ELB Provide? What Are Permissions of ELB?

ELB supports the following policies:

Table 4-1 ELB policies

Policy Type	Policy Name	Description
RBAC policy	ELB Administrator	Has all permissions on ELB Before assigning the RBAC policy to a user group, check whether the user group has a dependent policy. If yes, set the dependent permission to make the RBAC policy take effect.
Fine-grained policy	ELB FullAccess	Has all permissions on ELB. If this function is not enabled, you cannot assign a fine-grained policy to a user group.
	ELB ReadOnlyAccess	Has the read-only permission on ELB.

Table 4-2 Common operations supported by system-defined policies

Operation	ELB FullAccess	ELB ReadOnlyAccess	ELB Administrator
Creating a load balancer	√	×	√
Querying a load balancer	√	√	√
Querying a load balancer and associated resources	√	√	√
Querying load balancers	√	√	√
Modifying a load balancer	√	×	√
Deleting a load balancer	√	×	√
Adding a listener	√	×	√
Querying a listener	√	√	√
Modifying a listener	√	×	√
Deleting a listener	√	×	√

Operation	ELB FullAccess	ELB ReadOnlyAccess	ELB Administrator
Creating a backend server group	√	×	√
Querying a backend server group	√	√	√
Modifying a backend server group	√	×	√
Deleting a backend server group	√	×	√
Adding a backend server	√	×	√
Querying a backend server	√	√	√
Modifying a backend server	√	×	√
Removing a backend server	√	×	√
Configuring a health check	√	×	√
Querying a health check	√	√	√
Modifying a health check	√	×	√
Disabling a health check	√	×	√
Assigning an EIP	×	×	√
Binding an EIP to a load balancer	×	×	√
Querying an EIP	√	√	√
Unbinding an EIP from a load balancer	×	×	√

Operation	ELB FullAccess	ELB ReadOnlyAccess	ELB Administrator
Viewing metrics	×	×	√
Viewing access logs	×	×	√

NOTE

- To unbind an EIP, you also need to configure the **vpc:bandwidths:update** and **vpc:publicIps:update** permissions for the VPC service. For details, see the *Virtual Private Cloud API Reference*.
- To view metrics, obtain the **CES ReadOnlyAccess** permissions. For details, see the *Cloud Eye API Reference*.
- To view access logs, you also need to configure the **LTS ReadOnlyAccess** permission. For details, see the *Log Tank Service API Reference*.

For details about fine-grained permissions, see the *Elastic Load Balance API Reference*.

4.4 Can Backend Servers of a Load Balancer Run Different OSs?

Yes, you can associate backend servers that run different OSs with a load balancer. ELB does not restrict OSs of backend servers as long as applications on these servers are the same and the data is consistent.

To make management easier, it is recommended that you install the same OS on backend servers.

4.5 Can ELB Be Used Across Accounts or VPCs?

- Your shared load balancers cannot be used by another account, and you cannot associate backend servers whose VPCs are not the same as the load balancers.
- For dedicated load balancers, you can add servers in a VPC connected using a VPC peering connection, in a VPC in another region and connected through a cloud connection, or in an on-premises data center at the other end of a Direct Connect or VPN connection, by using their IP addresses.

4.6 Can a Backend Server Access Its Load Balancer?

Generally, you are not advised to use a server as both a backend server and a client.

If you have special requirements, select a solution with caution.

- Public network communication: If a backend server has a public IP address, it can access the public IP address of the associated load balancer.
- Private network communication:
  - The backend server can access the associated load balancer at Layer 7.
  - If **Transfer Client IP Address** is enabled for Layer 4 listeners, the backend server cannot serve as a client and access the associated load balancer.
  - If **Transfer Client IP Address** is not enabled for Layer 4 listeners, the backend server can serve as a client and access the associated load balancer.

    NOTE

    If **IP as a Backend** is enabled for the load balancer, **Transfer Client IP Address** settings will not be applied. In this case, the backend server can serve as a client and access the associated load balancer.

4.7 Can Both the Listener and Backend Server Group Use HTTPS?

Dedicated load balancers support this function.

You can select HTTPS as the protocol of both the listener and the backend server group.

To use HTTPS at the frontend only, you can create a dedicated load balancer, add an HTTPS listener to the load balancer, and set the backend protocol to HTTP.

NOTE

Using HTTPS at both the frontend and backend only allows you to enable mutual authentication on the load balancer and backend servers.

4.8 Does ELB Support IPv6 Networks?

Shared load balancers support only IPv4 networks. Dedicated load balancer support both IPv4 and IPv6 networks.

When a client communicates with a dedicated network load balancer using an IPv6 address, the load balancer must communicate with backend servers using an IPv6 address. When a client communicates with an application dedicated load balancer using an IPv6 address, the load balancer must communicate with backend servers using an IPv4 address.

NOTE

- If you do not enable IPv6 for the specified backend subnet when you create a dedicated load balancer, the load balancer cannot use IPv6 addresses to route requests.
- If you need IPv6 networks, you must select a backend subnet with IPv6 enabled for your dedicated load balancer.

4.9 How Do I Determine the Server Response Time Based on Monitoring Data and Logs?

For HTTP and HTTPS load balancing, you can view the average server response time through monitoring metric and view the response time of each request from access logs.

1. On the ELB console, click the name of the load balancer.
2. On the **Monitoring** tab page, select an HTTP or HTTPS listener added to the load balancer.
3. Check the **Average Server Response Time** metric to view the average time that backend servers respond to requests routed by the load balancer.

   Table 4-3 Average response time

Metric	Definition
Average Server Response Time	Average time that backend servers respond to requests from the load balancer (This metric is available only when the frontend protocol is HTTP or HTTPS.) The process starts when the load balancer routes the requests to backend servers and ends when it receives responses from backend servers. Unit: ms

4. Check **access logs** to view the response time of each request.

The **request_time**, **upstream_connect_time**, **upstream_header_time**, or **upstream_response_time** fields in the access log reflect the time required for a load balancer to route a request to the corresponding backend server.

Table 4-4 Parameter description

Field	Description
request_time	Request processing time in seconds, that is, the duration from the time when the load balancer receives the first request packet from the client to the time when the load balancer sends the response packet
upstream_connect_time	Time taken to establish a connection with the server, in seconds with a milliseconds resolution When the load balancer attempts to retry a request, there will be multiple connection times. If the request is not correctly routed to the backend server, a hyphen (-) is displayed as a null value for this field.

Field	Description
upstream_header_time	Time taken to receive the response header from the server, in seconds with a milliseconds resolution When the load balancer attempts to retry a request, there will be multiple response times. If the request is not correctly routed to the backend server, a hyphen (-) is displayed as a null value for this field.
upstream_response_time	Time taken to receive the response from the server, in seconds with a milliseconds resolution When the load balancer attempts to retry a request, there will be multiple response times. If the request is not correctly routed to the backend server, a hyphen (-) is displayed as a null value for this field.

4.10 How Can I Transfer the IP Address of a Client?

When you use ELB to route requests to backend servers, IP addresses of the clients will be translated by the ELB. This FAQ guides you to obtain the IP addresses of the clients.

- Load balancing at Layer 7 (HTTP or HTTPS listeners): Configure the application server and obtain the IP address of a client from the HTTP header.

  For details, see **Layer 7 Load Balancing**.
- Load balancing at Layer 4 (TCP or UDP listeners): Use either of the following methods to obtain the real IP address of a client.
  - Method 1: Enable **Transfer Client IP Address** for the listeners.
  - Method 2: Configure the TOA plug-in.

  For details, see **Layer 4 Load Balancing**.

Notes and Constraints

- If Network Address Translation (NAT) is used, you cannot obtain the IP addresses of the clients.
- If the client is a container, you can obtain only the IP address of the node where the container is located, but cannot obtain the IP address of the container.
- If **Transfer Client IP Address** is enabled for TCP or UDP listeners, a cloud server cannot be used as a backend server and a client at the same time.
- By default, the **Transfer Client IP Address** function is enabled for TCP and UDP listeners of dedicated load balancers and cannot be disabled.

NOTE

If both WAF and ELB are used, you can also obtain the IP addresses of the clients through WAF. For details, see **Web Application Firewall User Guide**.

Layer 7 Load Balancing

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

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

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

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

Apache Server

1. Install Apache 2.4.

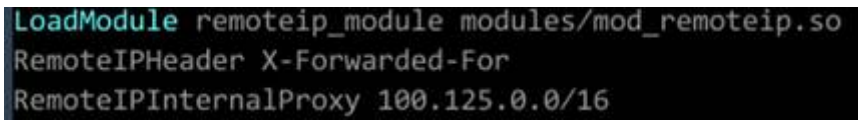
   For example, if CentOS 7.5 is used as the OS, run the following command to install the software:

   ```
   yum install httpd
   ```

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

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

   Figure 4-1 Content to be added

   NOTE

   Add the IP address range of the proxy server after **RemoteIPInternalProxy**.

   - Shared load balancers: 100.125.0.0/16 and the IP address range used by the AAD service. Load balancers use IP addresses in 100.125.0.0/16 to communicate with backend servers, and there are no security risks. Use commas (,) to separate multiple entries.
   - Dedicated load balancers: CIDR block of the subnet where the load balancer resides

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

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

4. Restart Apache.

   ```
   systemctl restart httpd
   ```

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

Nginx Server

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

1. Run the following commands to install http_realip_module:

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

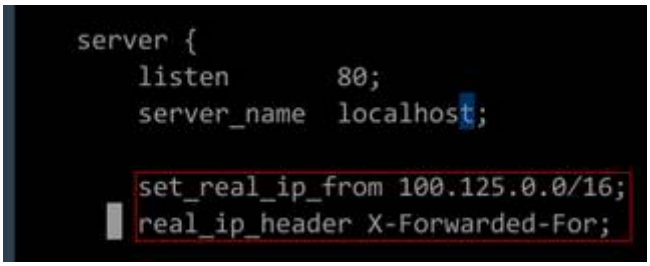
```
http_ssl_module --with-http_realip_module
make
make install
```

2. Run the following command to open the **nginx.conf** file:
   ```
   vi /path/server/nginx/conf/nginx.conf
   ```
3. Add new fields and information to the end of the following configuration information:

   Add the following information under **http** or **server**:

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

   Figure 4-2 Adding information

   NOTE

   Add the IP address range of the proxy server after **RemoteIPInternalProxy**.

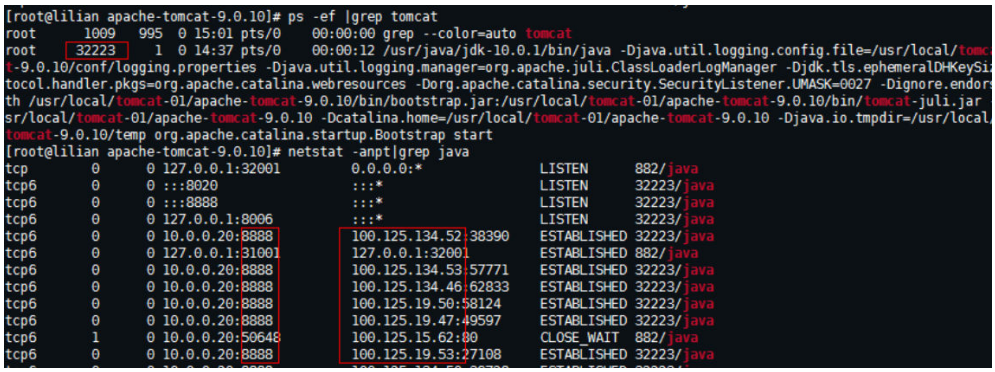
   - Shared load balancers: 100.125.0.0/16 and the IP address range used by the AAD service. Load balancers use IP addresses in 100.125.0.0/16 to communicate with backend servers, and there are no security risks. Use commas (,) to separate multiple entries.
   - Dedicated load balancers: CIDR block of the subnet where the load balancer resides
4. Start Nginx.
   ```
   /path/server/nginx/sbin/nginx
   ```
5. Obtain the actual IP address of the client from the Nginx access logs.
   ```
   cat /path/server/nginx/logs/access.log
   ```

Tomcat Servers

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

1. Log in to a server on which Tomcat is installed.
2. Check whether Tomcat is running properly.
   ```
   ps -ef|grep tomcat
   netstat -anpt|grep java
   ```

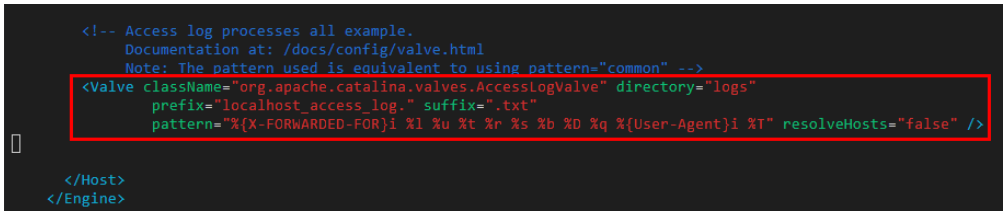

Figure 4-3 Tomcat running properly



3. Modify **className="org.apache.catalina.valves.AccessLogValve"** in the **server.xml** file as follows:

```
vim /usr/tomcat/tomcat8/conf/server.xml
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
prefix="localhost_access_log." suffix=".txt"
pattern="%{X-FORWARDED-FOR}i %l %u %t %r %s %b %D %q %{User-Agent}i %T"
resolveHosts="false" />
```

Figure 4-4 Example configuration

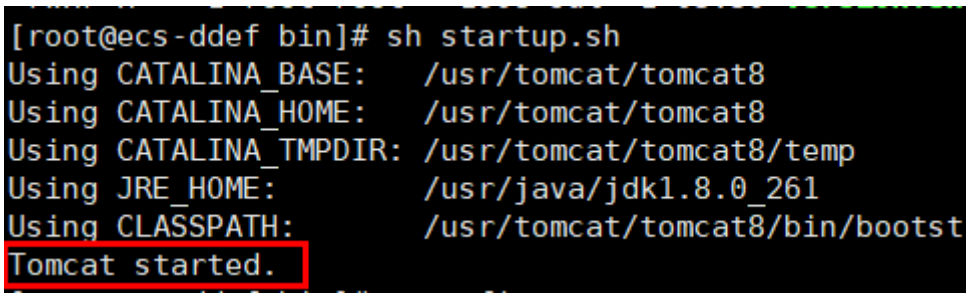


4. Restart the Tomcat service.

```
cd /usr/tomcat/tomcat8/bin && sh shutdown.sh && sh startup.sh
```

/usr/tomcat/tomcat8/ is where Tomcat is installed. Change it based on site requirements.

Figure 4-5 Restarting the Tomcat service



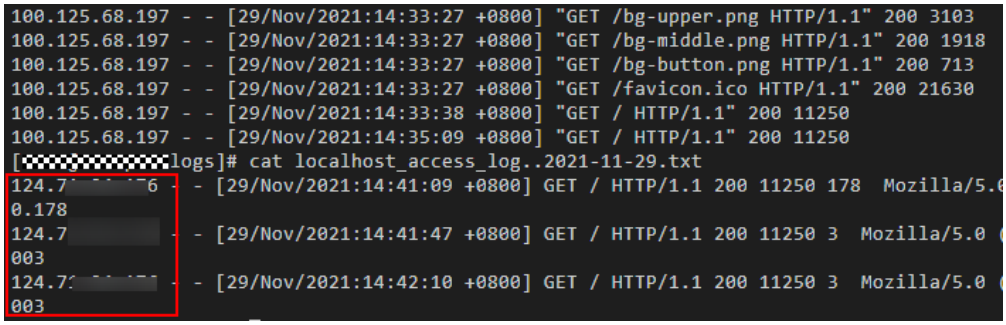
5. View the latest logs.

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

```
cd /usr/tomcat/tomcat8/logs/
cat localhost_access_log..2021-11-29.txt
```

   In this command, **localhost_access_log..2021-11-29.txt** indicates the log path of the current day. Change it based on site requirements.

Figure 4-6 Querying the source IP address



Windows Server with IIS Deployed

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

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

Figure 4-7 Selecting modules

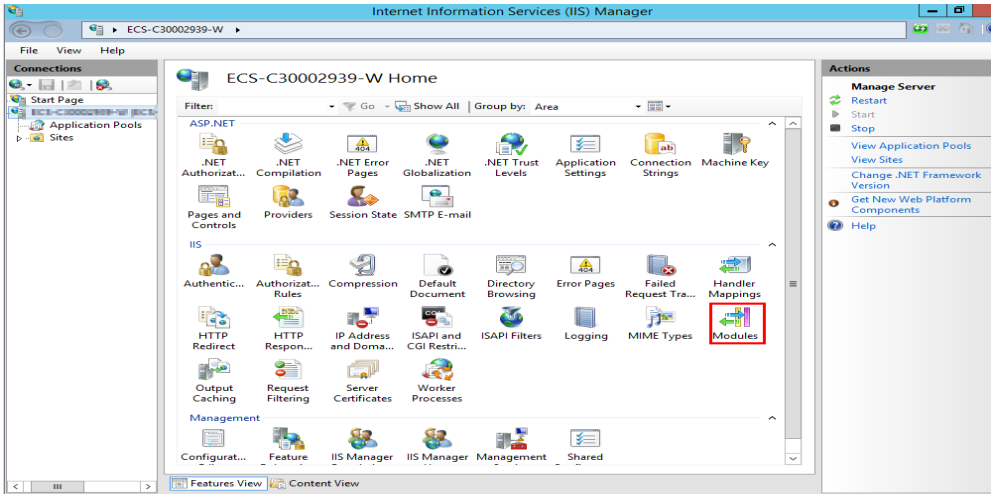
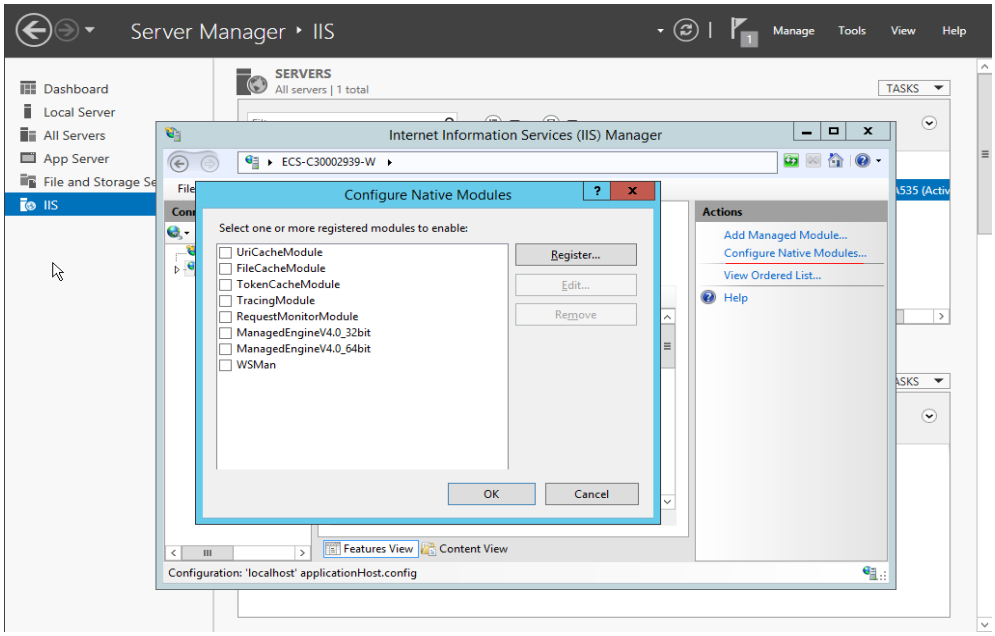
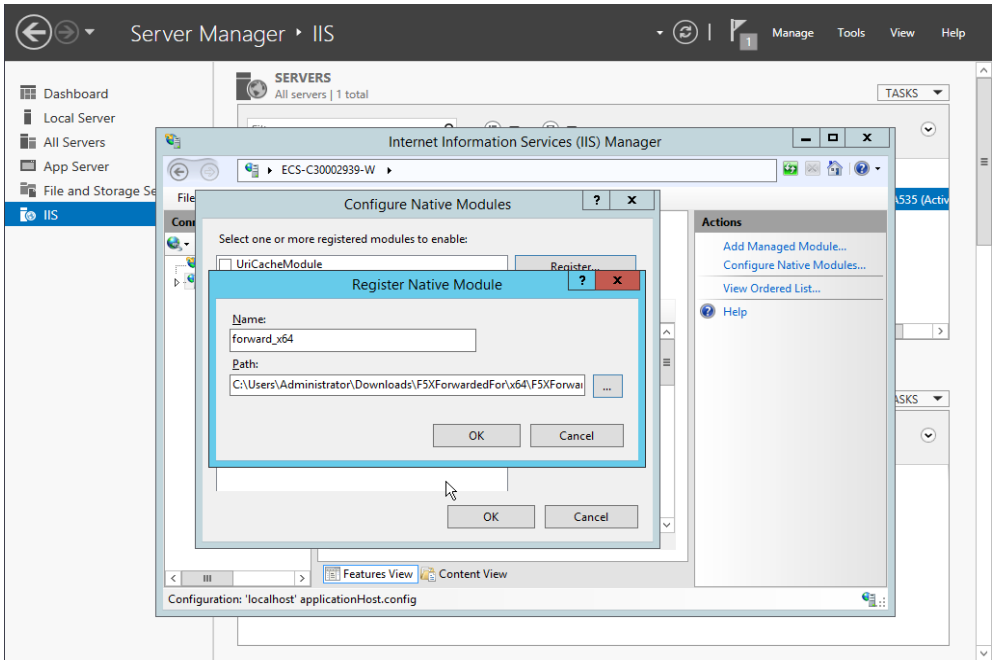


Figure 4-8 Configure Native Modules



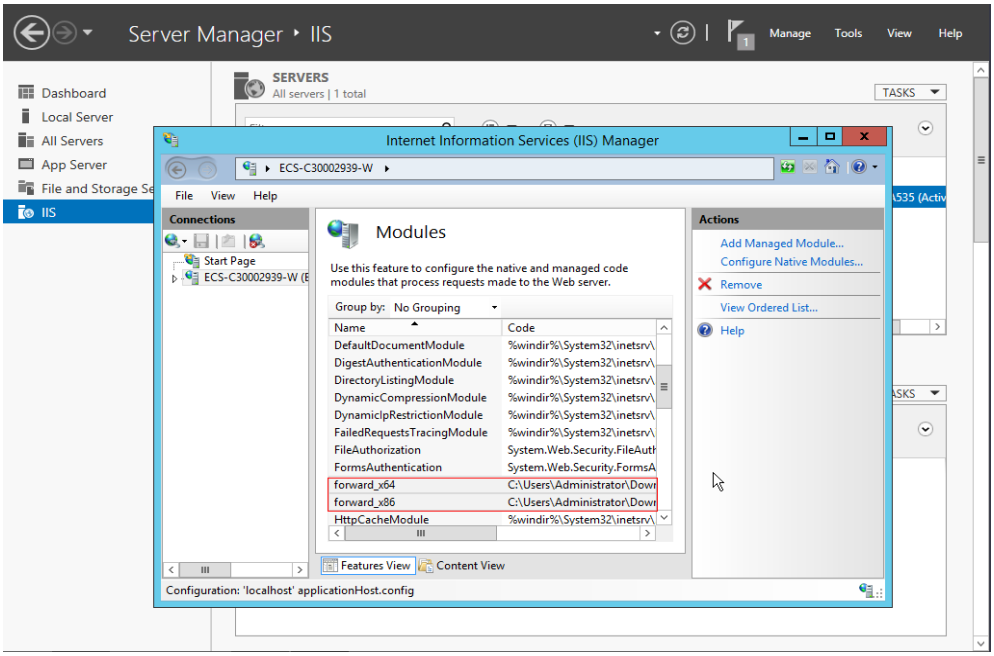
4. Click **Register** to register the x86 and x64 plug-ins.

Figure 4-9 Registering plug-ins



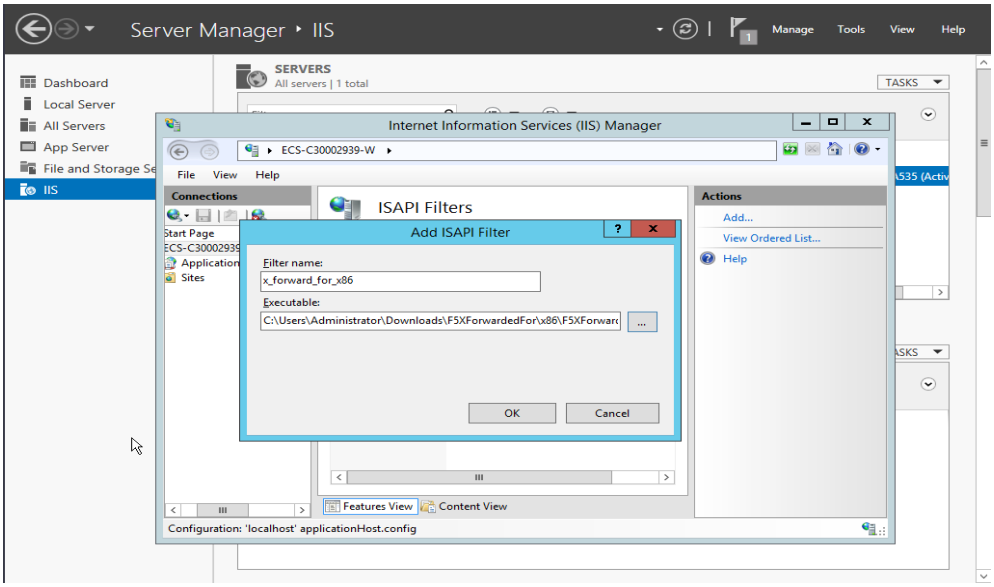
5. In the **Modules** dialog box, verify that the registered plug-ins are displayed in the list.

Figure 4-10 Confirming the registration



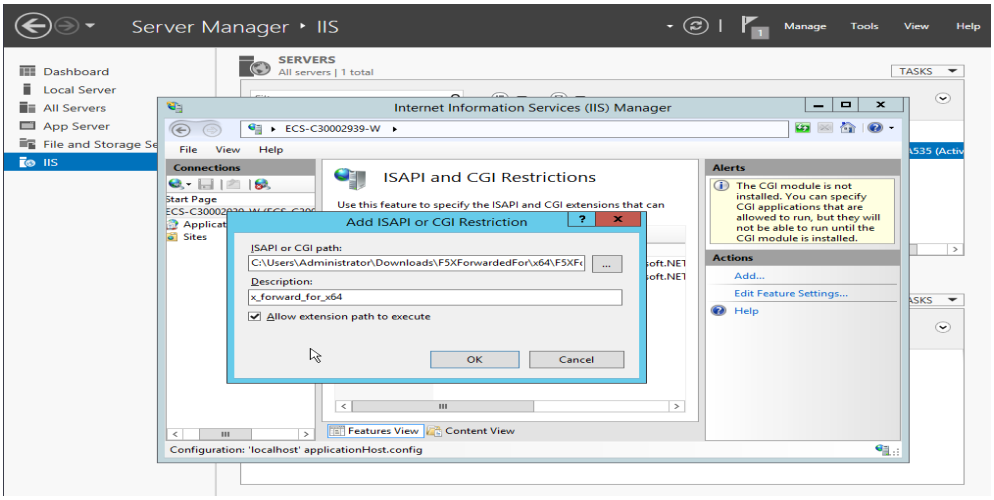
6. Select **ISAPI Filters** on the Server Manager homepage and authorize two plug-ins to run ISAPI and CGI extensions.

Figure 4-11 Adding authorization



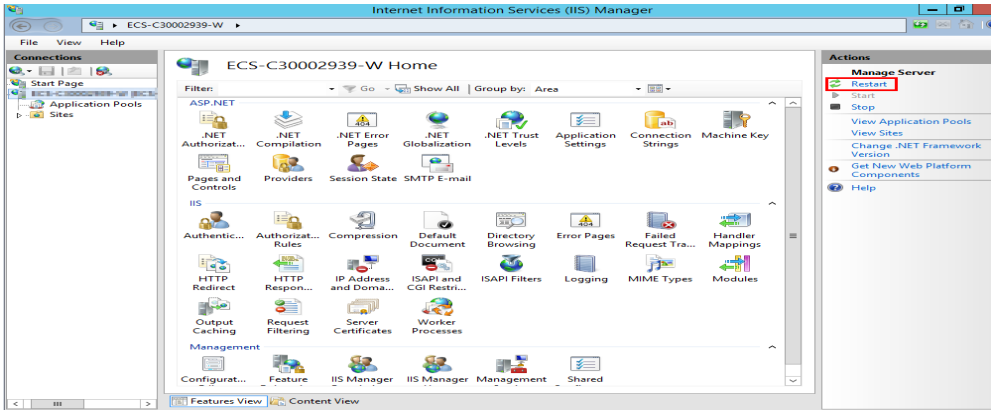
7. Select **ISAPI and CGI Restriction** to set the execution permission for the two plug-ins.

Figure 4-12 Allowing the plug-ins to execute



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

Figure 4-13 Restarting IIS



Layer 4 Load Balancing

For load balancing at Layer 4 (TCP or UDP listeners), use either of the following methods to obtain the real IP address of a client:

- **Method 1 (for TCP or UDP listeners)**: Enable **Transfer Client IP Address**.

CAUTION

- After this function is enabled, traffic, such as unidirectional download or push traffic, may be interrupted when backend servers are being migrated during the migration of the associated classic load balancer. After backend servers are migrated, retransmit the packets to restore the traffic.
- After this function is enabled, the associated backend servers cannot be used as clients to access the listener.
- If a backend server has been associated with the listener and health checks are enabled, enabling this function will check the health of the backend server, and traffic to this server will be interrupted for one or two health check intervals.

a. Perform the following steps to enable the function:
   i. Log in to the management console.
   ii. In the upper left corner of the page, click and select the desired region and project.
   iii. Click in the upper left corner to display **Service List** and choose **Networking** > **Elastic Load Balance**.
   iv. In the load balancer list, click the name of the load balancer.
   v. Click **Listeners**.
      - To add a listener, click **Add Listener**.
      - To modify a listener, locate the listener, click on the right of its name, and click **Modify Listener**. In the **Modify Listener** dialog box, modify the parameters as needed.
   vi. Enable **Transfer Client IP Address**.
b. Configure security groups, network ACLs, and OS and software security policies so that IP addresses of the clients can access these backend servers.

NOTE

If you enable this function, a server cannot serve as both a backend server and a client. If the client and the backend server use the same server and the **Transfer Client IP Address** option is enabled, the backend server will think the packet from the client is sent by itself and will not return a response packet to the load balancer. As a result, the return traffic will be interrupted.

- **Method 2 (for TCP listeners)**: Configure the TOA plug-in.

  TCP listeners require the TOA plug-in to obtain real IP addresses. For details, see **Configuring the TOA Plug-in**.

4.11 What Are the Differences Between Persistent Connections and Sticky Sessions?

Persistent connections are not necessarily related to sticky sessions.

A persistent connection allows multiple data packets to be sent continuously over a TCP connection. If no data packets are sent over the connection, the client and the server need to send link detection packets to each other. Sticky sessions enable all requests from the same client during one session to be sent to the same backend server.

4.12 How Do I Test Sticky Sessions Using Linux Curl Commands?

1. Prepare required resources.
   a. Buy three ECSs, one as the client and the other two as backend servers.
   b. Create a load balancer and add an HTTP listener to the load balancer. Enable sticky sessions when you add the listener.
2. Start the HTTP service of the two backend servers.

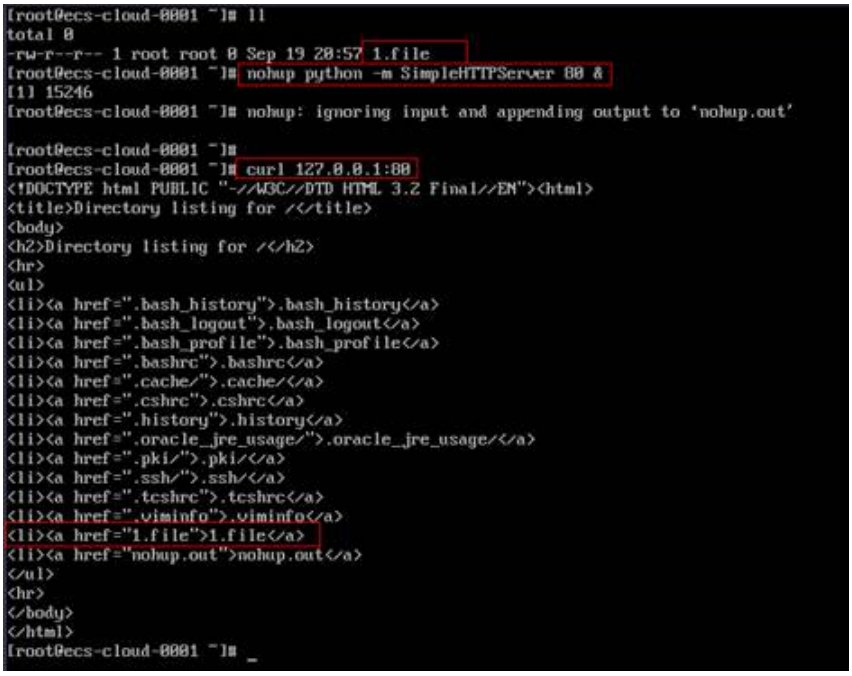
   Log in to a backend server and create a file named **1.file** in the current directory to mark this server.

   Run the following command in the current directory to start the HTTP service:

   nohup python –m SimpleHTTPServer 80 &

   Run the following command to check whether the HTTP service is normal:

   curl http://127.0.0.1:80



   Log in to the other backend server and create a file named **2.file** in the current directory.

   Run the following command in the current directory to start the HTTP service:

   nohup python –m SimpleHTTPServer 80 &

   Run the following command to check whether the HTTP service is normal:

   curl http://127.0.0.1:80


```
[root@ecs-cloud-0002 ~]# touch 2.file
[root@ecs-cloud-0002 ~]# nohup python -m SimpleHTTPServer 80 &
[1] 15244
[root@ecs-cloud-0002 ~]# nohup: ignoring input and appending output to 'nohup.out'

[root@ecs-cloud-0002 ~]# curl 127.0.0.1:80
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html>
<title>Directory listing for /</title>
<body>
<h2>Directory listing for /</h2>
<hr>
<ul>
<li><a href=".bash_history">.bash_history</a>
<li><a href=".bash_logout">.bash_logout</a>
<li><a href=".bash_profile">.bash_profile</a>
<li><a href=".bashrc">.bashrc</a>
<li><a href=".cache/">.cache/</a>
<li><a href=".cshrc">.cshrc</a>
<li><a href=".history">.history</a>
<li><a href=".oracle_jre_usage/">.oracle_jre_usage/</a>
<li><a href=".pki/">.pki/</a>
<li><a href=".ssh/">.ssh/</a>
<li><a href=".tcshrc">.tcshrc</a>
<li><a href=".viminfo">.viminfo</a>
<li><a href="2.file">2.file</a>
<li><a href="nohup.out">nohup.out</a>
</ul>
<hr>
</body>
</html>
[root@ecs-cloud-0002 ~]#
```

3. Access the load balancer from the client and specify the cookie value.

   The following is an example command. Change the parameters as needed. Ensure that the returned file names of each request are the same.

   curl --cookie "name=abcd" http://ELB_IP:Port

```
<body>
<h2>Directory listing for /</h2>
<hr>
<ul>
<li><a href=".bash_history">.bash_history</a>
<li><a href=".bash_logout">.bash_logout</a>
<li><a href=".bash_profile">.bash_profile</a>
<li><a href=".bashrc">.bashrc</a>
<li><a href=".cache/">.cache/</a>
<li><a href=".cshrc">.cshrc</a>
<li><a href=".history">.history</a>
<li><a href=".oracle_jre_usage/">.oracle_jre_usage/</a>
<li><a href=".pki/">.pki/</a>
<li><a href=".ssh/">.ssh/</a>
<li><a href=".tcshrc">.tcshrc</a>
<li><a href=".viminfo">.viminfo</a>
<li><a href="2.file">2.file</a>
<li><a href="nohup.out">nohup.out</a>
</ul>
<hr>
</body>
</html>
[root@ecs-client ~]# curl --cookie "name=abcd"  http://192.168.172.242:80
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html>
<title>Directory listing for /</title>
<body>
<h2>Directory listing for /</h2>
<hr>
<ul>
<li><a href=".bash_history">.bash_history</a>
<li><a href=".bash_logout">.bash_logout</a>
<li><a href=".bash_profile">.bash_profile</a>
<li><a href=".bashrc">.bashrc</a>
<li><a href=".cache/">.cache/</a>
<li><a href=".cshrc">.cshrc</a>
<li><a href=".history">.history</a>
<li><a href=".oracle_jre_usage/">.oracle_jre_usage/</a>
<li><a href=".pki/">.pki/</a>
<li><a href=".ssh/">.ssh/</a>
<li><a href=".tcshrc">.tcshrc</a>
<li><a href=".viminfo">.viminfo</a>
<li><a href="2.file">2.file</a>
<li><a href="nohup.out">nohup.out</a>
</ul>
<hr>
</body>
</html>
[root@ecs-client ~]# curl --cookie "name=abcd" http://192.168.172.242:80
```


5 Load Balancers

5.1 How Does ELB Distribute Traffic?

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

NOTE

In FullNAT, LVS translates source IP addresses and destination IP addresses of the clients.

Figure 5-1 Load balancing at Layer 4

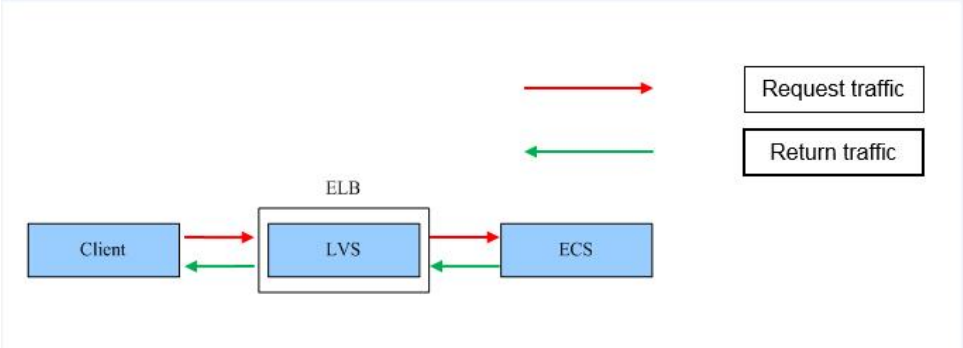
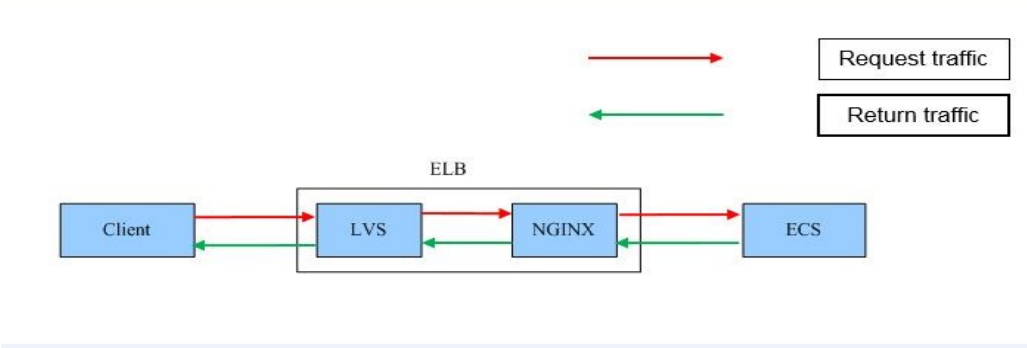


Figure 5-2 Load balancing at Layer 7



5.2 How Can I Configure Load Balancing for Containerized Applications?

You can configure load balancing using either of the following:

- Management console
- kubectl commands

For details, see **LoadBalancer**.

5.3 Can I Bind Multiple EIPs to a Load Balancer?

No.

- If you want to use the load balancer on a public network, you can only bind one EIP to the load balancer to receive requests from the Internet.
- If you want to use the load balancer in a VPC, bind a private IP address. To route requests from a different VPC, you need to create a VPC peering connection between the VPC where the load balancer works and the other VPC. For details, see section "Creating a VPC Peering Connection with Another VPC in Your Account" in the *Virtual Private Cloud User Guide*.

5.4 Why Multiple IP Addresses Are Required When I Create a Dedicated Load Balancer?

These IP addresses are used by underlying resources.

Generally, 2 IP addresses are required for creating a load balancer in an AZ, and 6 IP addresses are required for creating a load balancer with **IP as a Backend** enabled. If you create a load balancer in multiple AZs, more IP addresses will be required. There is an algorithm to determine how many IP addresses are required.

5.5 Can Backend Servers Access the Internet Using the EIP of the Load Balancer?

No.

The load balancer uses the EIP to receive requests from the Internet and routes the requests to backend servers over a private network.

If you want the backend servers to access the Internet or provide Internet-accessible services directly, you can bind an EIP to each backend server. You can also configure a NAT gateway for the backend servers so that they can share an EIP to access the Internet.

5.6 Do Shared Load Balancers Have Specifications?

No.

Shared load balancers share underlying resources, and the performance of one load balancer is affected by other load balancers. Only dedicated load balancers have exclusive use of their underlying resources. The performance of a dedicated load balancer is not affected by other dedicated load balancers on the Internet.

5.7 What Is the Difference Between the Bandwidth Defined in Each Specification of a Dedicated Load Balancer and the Bandwidth of an EIP?

If you use a load balancer with fixed specifications to forward requests over the Internet, check whether the bandwidth defined in each specification and the bandwidth of the EIP meet your service requirements.

- Bandwidth (Mbit/s) in each specification defines the maximum amount of inbound and outbound data transmitted over a dedicated load balancer.
- The bandwidth of the EIP bound to a load balancer defines the maximum bandwidth for traffic required by the clients to access the load balancer.

5.8 How Do I Combine ELB and WAF?

After you connect your website to Web Application Firewall (WAF), you can configure access control on ELB to allow only traffic from the WAF-back-to-source IP addresses to origin servers. This prevents hackers from obtaining your origin server IP addresses and then bypassing WAF to attack origin servers. For details, see **Web Application Firewall User Guide**.

6 Listeners

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

ELB supports three types of sticky sessions that can send requests from the same client to the same backend server. The following tables list the types of sticky sessions corresponding to each load balancing algorithm.

Table 6-1 Sticky sessions supported by dedicated load balancers

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

Table 6-2 Sticky sessions supported by shared load balancers

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

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

6.2 What HTTP Headers Can I Configure for an HTTP and HTTP Listener?

Table 6-3 describes the HTTP headers supported by HTTP and HTTP Listeners.

Table 6-3 Supported headers

Headers	Description
X-Forwarded-ELB-IP	The EIP bound to the load balancer is transmitted to backend servers through the HTTP header.
X-Forwarded-Host	The Host field in the request from the client is placed in X-Forwarded-Host and sent to backend servers.
X-Forwarded-Port	The protocol used by the listener is transmitted to backend servers through the HTTP header.
X-Forwarded-Proto	The protocol type (HTTP or HTTPS) of the request is transmitted to backend servers through the HTTP header.
X-Forwarded-For	Source IP addresses and proxy IP addresses of the clients are transmitted to backend servers through the HTTP header.
X-Real-IP	Source IP addresses of the clients are transmitted to backend servers through the HTTP header.

6.3 Will ELB Stop Distributing Traffic Immediately After a Listener Is Deleted?

- If a TCP or UDP listener is deleted, the load balancer immediately stops routing traffic because the client uses short connections to communicate with the load balancer.
- If an HTTP or HTTPS listener is deleted, persistent connections that have been established between the client and the load balancer will be kept alive until they time out, and therefore request routing is not affected. After the connections time out, the client stops sending requests over these connections. The default timeout duration is 300s.

  NOTE

  The duration for which persistent connections are kept alive is called idle timeout, and this takes effect only for persistent connections established between the client and load balancer.

6.4 Does ELB Have Restrictions on the File Upload Speed and Size?

- ELB has no restrictions on the file upload speed on the clients. However, the bandwidth may limit the upload speed.
- For HTTP or HTTPS listeners, the maximum file size is 10 GB. However, TCP or UDP listeners have no limit on the file size.

6.5 Can Multiple Load Balancers Route Requests to One Backend Server?

Yes. This is supported as long as the load balancers are in the same subnet as the backend server.

6.6 How Is WebSocket Used?

For HTTP listeners, unencrypted WebSocket (ws://) is supported by default. For HTTPS listeners, encrypted WebSocket (wss://) is supported by default.

6.7 What Are the Three Timeouts of a Listener and What Are the Default Durations?

Table 6-4 lists the timeout durations of listeners at both Layer 4 and Layer 7.

NOTE

- For shared load balancers, you can configure and modify the timeout durations for TCP, HTTP, and HTTPS listeners.
- For dedicated load balancers, you can configure and modify the timeout durations for TCP, UDP, HTTP, and HTTPS listeners.

Figure 6-1 Timeout durations at Layer 7

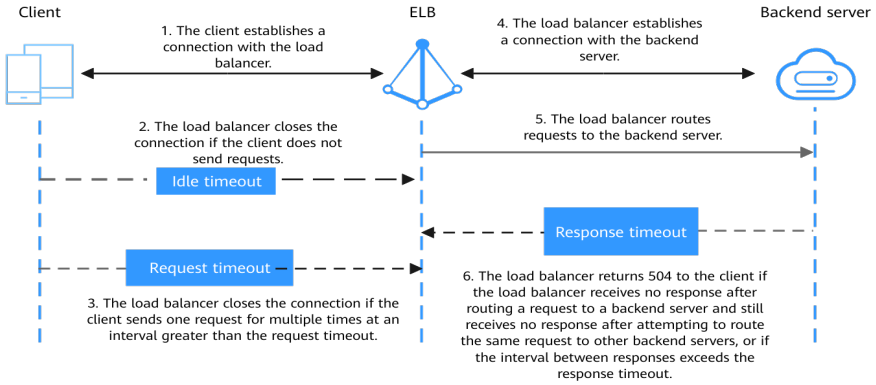


Figure 6-2 Timeout durations at Layer 4

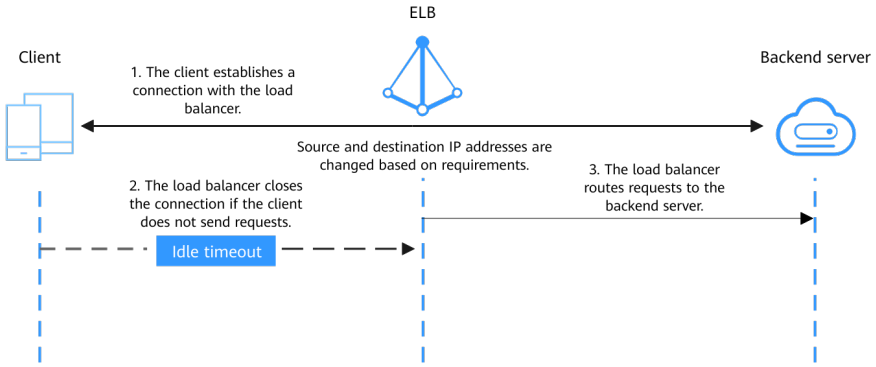


Table 6-4 Timeout durations

Protocol	Type	Description	Value Range	Default Timeout Duration
TCP	Idle Timeout (**keepalive_timeout**)	Duration for a connection to keep alive. If no request is received within this period, the load balancer closes the connection and establishes a new one with the client when the next request arrives.	10s to 4000s	300s
UDP	Idle Timeout (**keepalive_timeout**)		10s to 4000s	Shared load balancers: 10s Dedicated load balancers: 300s
HTTP/ HTTPS	Idle Timeout (**keepalive_timeout**)		0s to 4000s	60s
	Request Timeout (**client_timeout**)	Duration after which the load balancer closes the connection with the client if the load balancer does not receive a request from the client.	1s to 300s	60s
	Response Timeout (**member_timeout**)	Duration after which the load balancer sends a 504 Gateway Timeout error to the client if the load balancer receives no response after routing a request to a backend server and receives no response after attempting to route the same request to other backend servers **NOTE** If you have enabled sticky sessions and the backend server does not respond within the response timeout duration, the load balancer returns 504 Gateway Timeout to the clients.	1s to 300s	60s

6.8 Why Can't I Select the Target Backend Server Group When Adding or Modifying a Listener?

The backend server group's protocol (backend protocol) you want to select is not supported by the listener protocol (frontend protocol). There are some constraints on the backend protocol when you associate a backend server group with a listener.

Table 6-5 Frontend and backend protocols of dedicated load balancers

Frontend Protocol	Backend Protocol
TCP	TCP
UDP	UDP/QUIC
HTTP	HTTP
HTTPS	HTTP/HTTPS

Table 6-6 Frontend and backend protocols of shared load balancers

Frontend Protocol	Backend Protocol
TCP	TCP
UDP	UDP
HTTP	HTTP
HTTPS	HTTP

6.9 Why Is There a Security Warning After a Certificate Is Configured for an HTTPS Listener?

The following may cause the Not Secure warning even after a certificate is configured:

- The domain name used by the certificate is different from the domain name accessed by users. (If this is the case, check the domain name used the certificate to ensure that the domain names are the same or create a self-signed certificate.)
- SNI is configured, but the specified domain name is different from the one used by the certificate.
- The domain name level is inconsistent with the certificate level.

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

6.10 Why Is a Forwarding Policy in the Faulty State?

A possible cause is that you added a forwarding policy that is the same as an existing one. Even if you delete the existing forwarding policy, the newly-added forwarding policy is still faulty.

To resolve this issue, delete the newly-added forwarding policy and add a different one.

7 Backend Servers

7.1 Can Backend Servers Access the Internet After They Are Associated with a Load Balancer?

Yes. Backend servers can access the Internet whether they are associated with a load balancer.

7.2 Can ELB Distribute Traffic Across Servers That Are Not Provided by Huawei Cloud?

- Shared load balancers: Backend servers can only be cloud servers from Huawei Cloud. For more information, see *Backend Server Overview*.
- You can add servers in a VPC connected using a VPC peering connection, in a VPC in another region and connected through a cloud connection, or in an on-premises data center at the other end of a Direct Connect or VPN connection, by using their IP addresses. For more information, see Backend Server Overview.
- Database instances cannot be used as backend servers.

7.3 Why Are Backend Servers Frequently Accessed by IP Addresses in 100.125.0.0/16 or 214.0.0.0/8?

Shared load balancers use IP addresses in 100.125.0.0/16 as source addresses to route traffic to backend servers and to check the health of backend servers, if you have configured a health check.

To ensure that your shared load balancer can provide services properly, ensure that the security groups configured for the backend servers allow traffic from 100.125.0.0/16.

NOTE

If 100.125.0.0/16 has no available IP addresses in the region where the shared load balancer works, an IP address 214.0.0.0/8 will be used as a source IP address to forward traffic to backend servers and to check the health of backend servers.

7.4 Can ELB Route Traffic Across Regions?

- Shared load balancers cannot distribute traffic across regions.
- Dedicated load balancers can distribute traffic across regions or VPCs.
  - To add backend servers in different regions, you can use Cloud Connect to connect the VPCs across regions. For details, see the **Cloud Connect User Guide**.
  - To add backend servers in a different VPC or an on-premises data center, you need to enable **IP as a Backend** for the load balancer. For details, see *the Elastic Load Balance User Guide*.

7.5 Does Each Backend Server Need an EIP to Receive Requests from a Public Network Load Balancer?

No. There is no need to bind an EIP to each backend server because the load balancer routes requests through the private network.

7.6 How Do I Check the Network Conditions of a Backend Server?

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

      NOTE

      For Windows ECSs, use **ipconfig** on the CLI.
2. Ping the gateway of the subnet where the ECS resides to check for network connectivity.
   a. On the VPC details page, locate the subnet and view the gateway address in the **Gateway** column. Generally, the gateway address ends with **.1**.
   b. Ping the gateway from the ECS. If the gateway cannot be pinged, check the networks at Layer 2 and Layer 3.

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

1. Check whether the security group of the server is correctly configured.
   a. On the server details page, view the security group.

b. Check whether the security group rules allow access from the corresponding IP address range.

- Dedicated load balancers: Check whether the security group of the backend server has inbound rules to allow traffic from the VPC where the load balancer works. If traffic is not allowed, add an inbound rule to allow traffic from the VPC to the backend server.
- Shared load balancers: Check whether the security group of the backend server has inbound rules to allow traffic from the 100.125.0.0/16. If traffic is not allowed, add an inbound rule to allow traffic from 100.125.0.0/16 to the backend server.

CAUTION

- Shared load balancers: If **Transfer Client IP Address** is enabled for a TCP or UDP listener, there is no need to configure security group rules and network ACL rules to allow traffic from 100.125.0.0/16 and client IP addresses to backend servers.
- Dedicated load balancers: If **IP as a Backend** is not enabled for a load balancer that has a TCP or UDP listener, there is no need to configure security group rules and network ACL rules to allow traffic from the backend subnet where the load balancer is deployed to the backend servers.

2. Ensure that the network ACLfirewalls of the subnet where the server resides does not intercept the traffic.

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

7.8 How Do I Check the Status of a Backend Server?

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

      netstat -ntpl

      NOTE

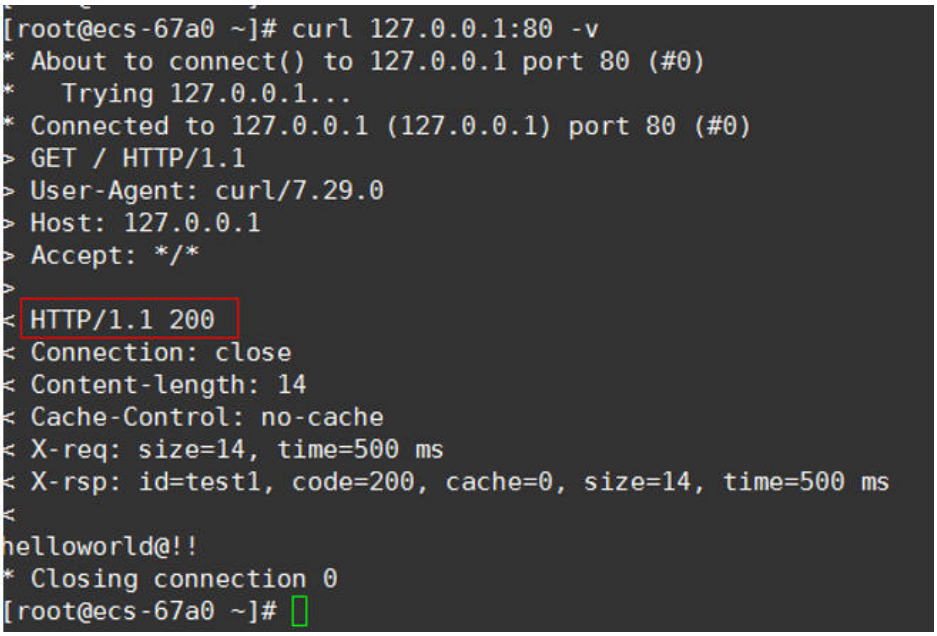
      For Windows ECSs, use **netstat -ano** on the CLI to view the port status or server software status.

      Figure 7-1 Port status

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

2. Check the network communication of the ECS.

   For example, if the ECS uses port 80, use **curl** to check whether network connectivity is normal.



7.9 How Do I Check Whether a Backend Server Can Be Accessed Through an EIP?

1. Bind an EIP to the backend Server.
   a. Log in to the management console.
   b. In the upper left corner of the page, click and select the desired region and project.
   c. Click , and choose **Computing** > **Elastic Cloud Server**.
   d. Locate the ECS and click its name.
   e. Under **EIPs**, click **Bind EIP**.
   f. Select the EIP to be bound and click **OK**.
2. Verify that the ECS can be accessed through the EIP.

For Linux ECSs, use **curl**. For Windows ECSs, use a browser.

7.10 Why Is the Number of Active Connections Monitored by Cloud Eye Different from the Number of Connections Established with the Backend Servers?

The number of active connections collected by Cloud Eye refers to the number of active connections between clients and the load balancer.

For a TCP or UDP listener, the load balancer transparently transmits client requests. The number of active connections is equal to the number of connections that the load balancer establishes with backend servers.

For an HTTP or HTTPS listener, the clients connect to the load balancer, which then connects to backend servers. The number of active connections is not related to the number of connections established with backend servers.

7.11 Why Can I Access Backend Servers After a Whitelist Is Configured?

The whitelist controls only access to a listener. Only IP addresses in the whitelist can access the listener. To control access to backend servers, you can configure network ACL rules or security group rules.

7.12 When Will the Changes to Server Weights Be Applied?

The changes are applied within 5 seconds.

- TCP and UDP listeners forward requests over new connections based on the new weights. However, connections that have been established with backend servers are still routed based on the previous weights.
- HTTP and HTTPS listeners forward requests based on the new weights. However, requests that have been forwarded to backend servers will not be affected.

NOTE

If the weight of a backend server is changed to 0, the new weight does not take effect immediately, and requests are still routed to this backend server. This is because a persistent connection is established between the load balancer and the backend server and requests are routed to this server until the connection times out.

- TCP and UDP listeners: Persistent connections are disconnected after the idle timeout duration expires.
- HTTP and HTTPS listeners: Persistent connections are disconnected after the response timeout duration expires.

8 Certificates

8.1 Does ELB Support Wildcard Certificates?

Yes.

Shared load balancers support the longest suffix match by default.

Dedicated load balancers using a SNI certificate support wildcard match by default. Only the subdomain names of the same level can be matched. You can change wildcard match to longest suffix match by changing the value of **sni_match_algo**. For details, see ***Elastic Load Balance API Reference***.

Table 8-1 Examples of wildcard-domain matching rules

Domain Name	Wildcard Match	Longest Suffix Match
*.example.com	Domain names, such as abc.example.com, sport.example.com, and good.example.com	Domain names, such as abc.example.com and mycalc.good.example.com

8.2 Why Is Access to Backend Servers Still Abnormal Even If I Have Created a Certificate?

The following are possible causes:

- You have created a certificate on the ELB console, but you do not have an HTTPS listener.

  To solve this problem, perform the following steps:

  - Continue using the current listener and install the certificate on the backend server.
  - Delete the current listener, add an HTTPS listener, and bind a certificate to the HTTPS listener.

- You have created a certificate on the **Certificates** page and are using an HTTPS listener, but you have not bound the certificate to the listener.
- Your certificate has expired.
- The domain name is different from the one specified when you create the certificate.
- A certificate chain is used, but its format is incorrect.
- You have bound a certificate to the HTTPS listener and also configure a certificate on the backend servers. Because you bind a certificate to the listener, ELB decrypts HTTPS requests from clients and sends decrypted requests to backend servers, and the certificate on backend servers decrypts these decrypted requests again. (Shared load balancers have this restriction, while dedicated load balancers do not have this restriction.)

  You can use either of the following methods to solve the problem:

  – Configure a certificate on the backend servers and use a TCP listener to transparently transmit HTTPS traffic to the backend servers.
  – Use an HTTPS listener and bind a certificate to the HTTPS listener. Do not configure the certificate on the backend servers.

8.3 Will the Network or Load Balancing Be Interrupted When a Certificate Is Being Replaced?

No.

The new certificate takes effect immediately after the replacement. The old certificate is used for established connections, and the new one is used for new connections.

NOTE

When the certificate expires, the system displays a message indicating that the connection is insecure. However, you can ignore the warning and continue accessing the website.

9 Access Logging

9.1 Why Are Access Logs Not Displayed for My Load Balancer?

- Ensure that LTS has been enabled and that a log group and a log stream have been created. For details, see **Access Logging**.
- Ensure that the load balancer can be accessed.
- Ensure that the load balancer supports access logging.

  Access logging can be enabled for HTTP or HTTPS listeners of shared load balancers.

9.2 What Information Can I Provide to Assist O&M Personnel?

Contact customer service if ELB still fails to respond after you have performed the operations provided in sections **How Do I Check the Network Conditions of a Backend Server?** to **How Do I Check Whether a Backend Server Can Be Accessed Through an EIP?**

Provide customer service with the following information.

Item	Fill in the Details
Load balancer ID	-
VPC ID	-
Load balancer IP address	-
Listener ID	-
Frontend protocol and port	-
Health check protocol and port	-

Item	Fill in the Details
Health check result	-
ID of ECS 1	-
ID of ECS 2	-

10 Monitoring

10.1 Why Is the Outgoing Rate on the ELB Console Inconsistent with the Bandwidth Usage Statistics on the Cloud Eye Console?

In the following scenarios, outgoing rate monitored by ELB is inconsistent with EIP bandwidth usage statistics on Cloud Eye:

- If the traffic does not exceed the bandwidth set for the EIP, the bandwidth is not limited and Cloud Eye collects statistics on the public network while ELB collects data on the private network.
- If the traffic exceeds the bandwidth set for the EIP, the bandwidth is limited. Traffic to the ELB system passes through a path that is different from the path in which traffic passes to the EIP.

10.2 What Are the Differences Between Layer-7 Status Codes and Backend Status Codes in ELB Metrics?

HTTP or HTTPS listeners terminate TCP connections. In other words, there are two TCP connections between the client and a backend server, one between the client and load balancer, and the other between the load balancer and backend server. The communication between the client and the backend server is divided into two parts. After receiving an HTTP request, the load balancer parses the request and routes the parsed request to the backend server for processing. The backend server returns a response to the load balancer after receiving the request. The load balancer then parses the response and returns the parsed response to the client. Therefore, there are two types of status codes: backend status codes returned by the backend server to the load balancer and Layer-7 status codes returned by the load balancer to the client.

You may encounter the following situations:

- The backend server returns a status code, and the load balancer directly transmits the status code to the client. In this case, the Layer-7 status code is the same as the backend status code.

- If the connection between the load balancer and backend server is abnormal or times out, the load balancer returns HTTP 502 or 504 to the client.
- If the listener configuration or the request format or content is incorrect, the load balancer directly returns an HTTP 4xx status code or 502 to the client, and does not route the request to the backend server. In this case, there will be only a Layer-7 status code, but no backend status code.

10.3 Why Is There a Large Number of HTTP 499 Errors?

When you are seeing the HTTP 499 status code, the client has closed the connection while the server is still processing the request.

The possible causes are as follows:

- The request timeout may not be long enough for the client to send HTTP requests before a connection is closed. Check the **request_time** field in the access log to view the total time for processing requests and set an appropriate request timeout.
- Your load balancer may be overloaded with traffic, causing packet loss due to bandwidth limit. Check the outbound bandwidth usage of your load balancer on the Cloud Eye console. For more information, see **Monitoring Metrics**.
- The network that connects the client and your load balancer may be unstable, causing long round-trip delay or packet loss. Check the **request_time** and **tcpinfo_rtt** fields in the access log or capture packets to check whether the network is normal.
- The backend server may take a longer time than the request timeout interval to process requests. Check whether the CPU, memory, and network of the backend server have performance bottlenecks.
- The client closes the connection before receiving a response from the server due to some unknown reasons. Check whether the client closes the connection before an HTTP request is complete.

11 Billing

11.1 When Do I Need Public Bandwidth for ELB?

To access a load balancer over the Internet, you need to buy an EIP, set a bandwidth for the EIP, and bind the EIP to the load balancer. If you access the load balancer within a VPC, no EIP and bandwidth are required.

If you access backend servers through their EIPs, the EIP and bandwidth of the load balancer are not used.

11.2 Will I Be Billed for Both the Bandwidth Used by the Load Balancer and the Bandwidth Used by Backend Servers?

This depends on your services. If backend servers are only accessed from within a VPC, you do not need to bind an EIP to each backend server and assign bandwidth because requests from the clients are received and routed to backend servers by the load balancer. You only need to bind an EIP to each backend server if they need to provide services accessible from the Internet. In that case, you need to pay for the bandwidth used by your load balancer and also the bandwidth used by the backend servers.

11.3 Do I Need to Adjust the Bandwidth of Shared Load Balancers Based on the Bandwidth Used by Backend Servers?

- If a public network load balancer is used, the bandwidth used by its EIP depends on the incoming traffic. It is not determined by the bandwidth used by backend servers. However, you may need to adjust its bandwidth if there is a surge in incoming traffic, which will cause the load balancer to automatically scale up.

- If the load balancer is used on a private network, there is no need to adjust.

11.4 Can I Modify the Bandwidth of a Load Balancer?

Yes. You can modify the bandwidth of a load balancer by referring to **Modifying the Bandwidth**.

11.5 What Functions Will Become Unavailable If a Load Balancer Is Frozen?

A load balancer may be frozen for either of the following reasons:

- Insufficient account balance
- Public security

When a dedicated load balancer is frozen, the following functions will be affected:

1. The load balancer will no longer distribute incoming traffic.
2. The health check function will be stopped. Health check results of backend servers displayed on the console are the results that were obtained before the load balancer was frozen.
3. The load balancer will stop reporting monitoring data to Cloud Eye.
4. The following operations cannot be performed through API calls:
   a. Modifying load balancer parameters except **Name** and **Tag**
   b. Deleting a load balancer if it is frozen due to violations of public security regulations

      In this case, resources associated with the load balancer, such as listeners, backend server groups, backend servers, health checks, forwarding policies, and forwarding rules, cannot be created, deleted, or modified.

