Elastic Load Balance

FAQs

Issue	01
Date	2022-09-30

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

HUAWEI and other Huawei trademarks are trademarks 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 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

- **How Can I Transfer the IP Address of a Client?**
- **How Do I Troubleshoot an Unhealthy Backend Server?**
- **How Does ELB Perform UDP Health Checks? What Are the Precautions for UDP Health Checks?**
- **What Types of Sticky Sessions Does ELB Support?**
- **Can I Modify the Bandwidth of a Load Balancer?**
- **How Is WebSocket Used?**
- **How Do I Check If Sticky Sessions Failed to Take Effect?**
- **What Are the Relationships Between Load Balancing Algorithms and Sticky Session Types?**
- **How Does ELB Distribute Traffic?**
- **What Is Quota?**

2 ELB Use

2.1 Service Abnormality

2.1.1 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

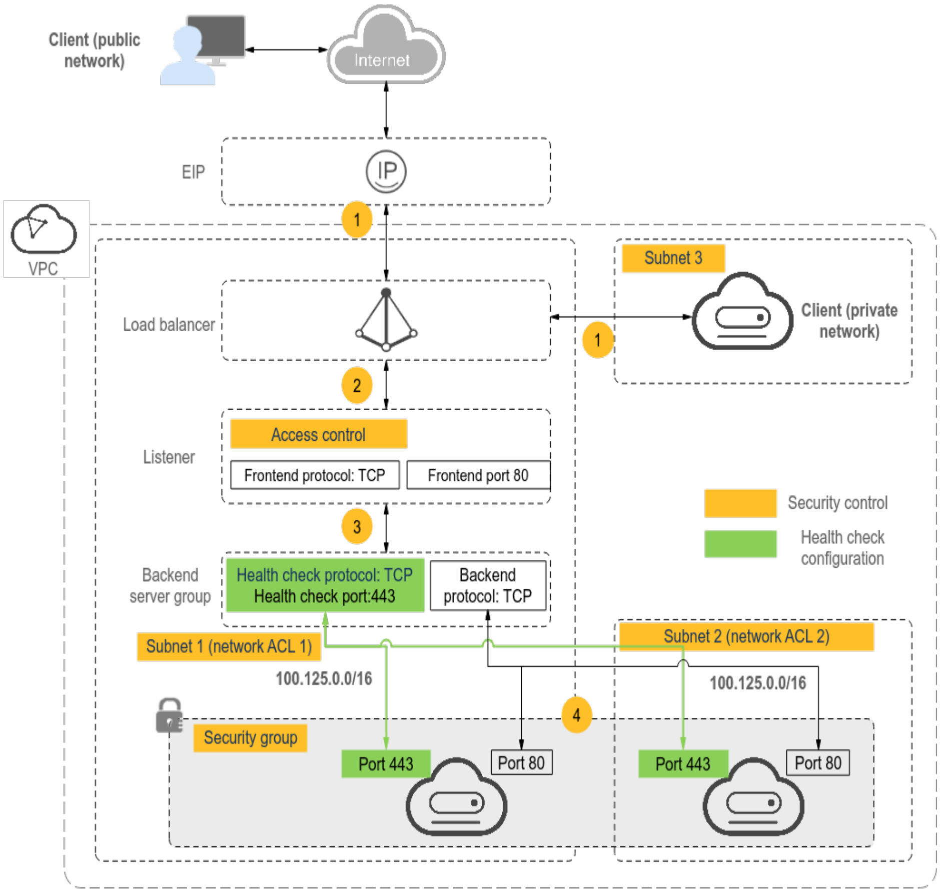
Figure 2-1 shows how clients access backend servers through a load balancer.

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).

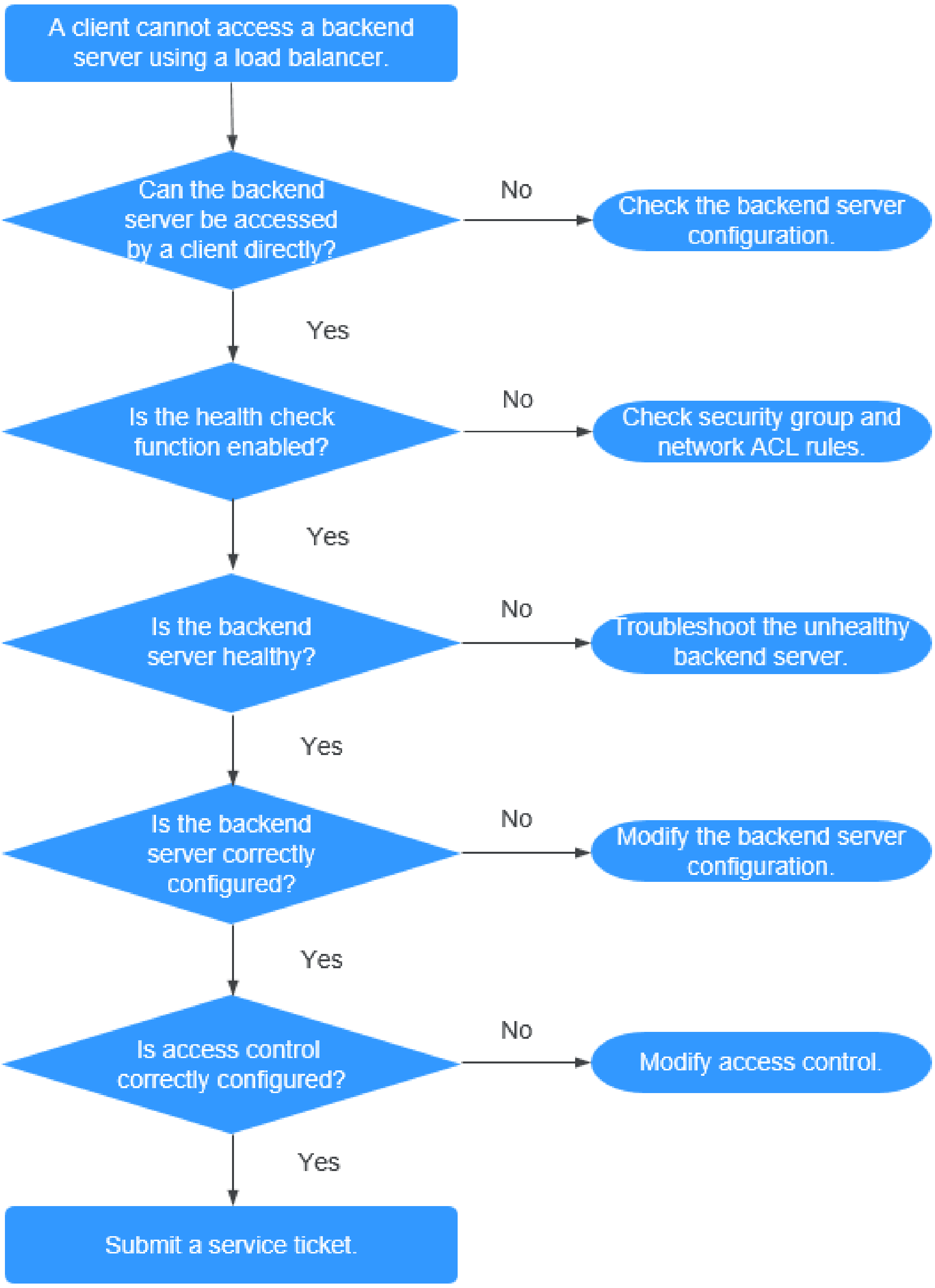
Troubleshooting should start with backend servers, then move on to the load balancer, and finally to the clients.

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



Troubleshooting Process

Figure 2-2 Troubleshooting process


A client cannot access a backend server using a load balancer.
Can the backend server be accessed by a client directly?
No
Check the backend server configuration.
Yes
Is the health check function enabled?
No
Check security group and network ACL rules.
Yes
Is the backend server healthy?
No
Troubleshoot the unhealthy backend server.
Yes
Is the backend server correctly configured?
No
Modify the backend server configuration.
Yes
Is access control correctly configured?
No
Modify access control.
Yes
Submit a service ticket.

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 network communication between the client and backend server are enabled.

- 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. Hover on ☰ 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 **Backend Server Groups** tab page, 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

- 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. Locate the listener, click the **Backend Server Groups** tab on the right, and view the health check result of the backend server.

- 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. Locate the listener, click the **Backend Server Groups** tab on the right, and view the backend server parameters. Note the following 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 page, 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.
     - Ensure that the security group allows traffic from the backend subnet where the load balancer resides 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 **Basic Information** tab page 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**.

2.1.2 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.
   - 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.

   ⚠ CAUTION

   - 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 a backend server and a client at the same time.
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.

2.1.3 How Can I Handle Error Codes?

Common error codes include 400, 403, 502, and 504. 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 2-1**. If the fault persists, contact customer service.

Table 2-1 Common error codes

Error 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 error code is returned to the client by the backend server.)
403	Forbidden	The request was intercepted by the backend server. (This error 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 error 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	URI Too Long	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 error code is recorded only in access logs.
500	Internal Server Error	There was an internal error. (This error code is returned to the client by the backend server.)

Error Code	Description	Possible Causes
501	Not Implemented	The load balancer failed to identify the request. The value of the **Transfer-Encoding** header field is not **chunked** or **identity**.
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 is incorrectly configured, for example, incorrect routes or network ACL.
503	Service Unavailable	The application or backend server was unavailable. Generally, this error code is returned by the backend server.
504	Gateway Timeout	● During the first connection, the load balancer fails 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 of the subnet did not allow the load balancer to access backend servers in the subnet.

2.2 ELB Functionality

2.2.1 Can ELB Be Used Separately?

ELB cannot be used alone.

ELB distributes incoming traffic to multiple backend servers based on the forwarding policy to balance workloads. So, it can expand external service capabilities of your applications and eliminate single points of failure (SPOFs) to improve service availability. To use a load balancer, you must associated backend servers (such as ECSs) with it.

2.2.2 Does ELB Support Persistent Connections?

Yes.

The connections between the client and load balancer are persistent connections. After a TCP persistent connection is established, the client continuously sends HTTP requests to the load balancer until the connection times out. The reuse of TCP connections reduces the costs for a large number of short connections.

2.2.3 Does ELB Support FTP on Backend Servers?

ELB does not support File Transfer Protocol (FTP), but supports Secure File Transfer Protocol (SFTP) on backend servers.

2.2.4 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 in the HUAWEI CLOUD, other clouds, and on-premises data centers.

2.2.5 Is an EIP Assigned Exclusively to a Load Balancer?

During the lifecycle of a load balancer, whether the assigned EIP is exclusive depends on the type of the load balancer.

- Load balancers: The EIP can be unbound from the load balancer. If the EIP is unbound, the load balancer becomes a private network load balancer, and the EIP can be bound to other resources.

2.2.6 How Many Load Balancers and Listeners Can I Have?

By default, each account can have up to 50 load balancers and 100 listeners. If you need more load balancers or listeners, apply to increase your quotas.

All load balancers in your account share the same quota for listeners.

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

ELB supports the following policies:

Table 2-2 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 2-3 Common operations supported by each system policy

Operation	ELB FullAccessAdmin	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	√	×	√
Adding a backend server group	√	×	√

Operation	ELB FullAccessAdmin	ELB ReadOnlyAccess	ELB Administrator
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	√	×	√
Deleting 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	×	×	√
Viewing metrics	×	×	√
Viewing access logs	×	×	√

NOTE

- To unbind an EIP, you also need to configure the **vpc:bandwidths:update** and **vpc:publicIps:update** permission of the VPC service. For details, see the *Virtual Private Cloud API Reference*.
- To view monitoring metrics, you also need to configure the **CES ReadOnlyAccess** permission. 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*.

2.2.8 Can I Adjust the Number of Backend Servers When a Load Balancer is Running?

You can adjust the number of backend servers associated with a load balancer at any time. You can also change the type of backend servers according to your service needs. To ensure service stability, ensure that health checks are normal and that at least one healthy backend server is associated with the load balancer.

2.2.9 Can Backend Servers Run Different OSs?

Yes.

ELB does not restrict OSs of backend servers as long as applications on these servers are the same and the data is consistent. However, it is recommended that you install the same OS on backend servers to simplify management.

2.2.10 Can I Configure Different Backend Ports for a Load Balancer?

Yes. You can configure different backend ports for backend servers associated with a load balancer.

2.2.11 Are There Any Restrictions on the Frequency of Access from an IP Address?

No. ELB does not limit the access frequency.

A blacklist denies access from specified IP addresses, and a whitelist allows access from specified IP addresses.

2.2.12 Can ELB Be Used Across Accounts or VPCs?

- 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. For details, see **Configuring Hybrid Load Balancing**.

2.2.13 Can Backend Servers Access the Ports of a Load Balancer?

No. Backend servers cannot access the ports of the load balancer they are associated with.

2.2.14 Can I Bind a Public IP Address Purchased from a Third-Party Cloud Provider to My Load Balancer?

No.

You can bind only an EIP purchased from HUAWEI CLOUD to your load balancer.

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

Dedicated load balancers support this function.

You can select HTTPS as the listener's and the backend server group's protocol. For details about how to add a listener, see **Adding an HTTPS Listener**.

2.2.16 Can I Change the VPC and Subnet for My Load Balancer?

You can change the subnet but not the VPC for your dedicated load balancers.

2.2.17 Does ELB Support IPv6 Networks?

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

At Layer 4, when a client communicates with a dedicated load balancer using an IPv6 address, the load balancer must communicate with backend servers using an IPv6 address. At Layer 7, when a client communicates with a dedicated load balancer using an IPv6 address, the load balancer must communicate with backend servers using an IPv4 address.

Figure 2-3 Network types supported by dedicated load balancers at Layer 4

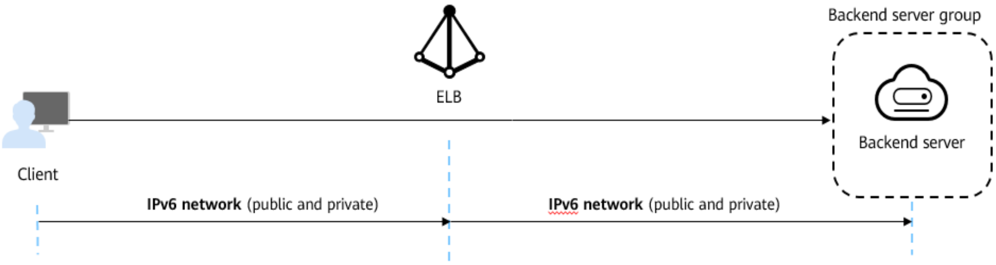
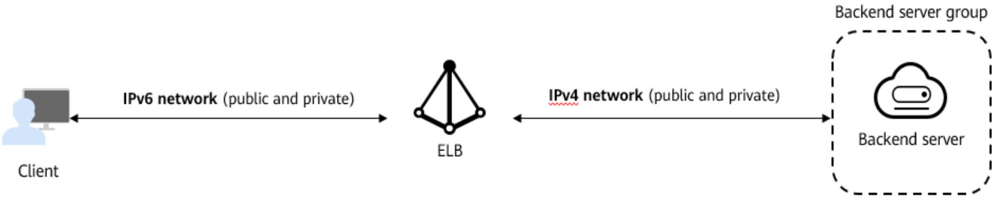

ELB
Backend server group
Backend server
Client
IPv6 network (public and private)
IPv6 network (public and private)

Figure 2-4 Network types supported by dedicated load balancers at Layer 7


Backend server group
IPv6 network (public and private)
IPv4 network (public and private)
ELB
Backend server
Client

2.3 Load Balancing Performance

2.3.1 How Do I Check for Traffic Inconsistencies?

Check for 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.

2.3.2 How Do I Check If Traffic Is Being Evenly Distributed?

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.
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.

2.3.3 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 health of backend servers based on the **Unhealthy Servers** metric. 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.

2.3.4 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 lost 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, and the client will time out. You can increase the maximum allowed of the backlog by adjusting the **net.core.somaxconn** parameter.

3 Load Balancers

3.1 What Is Quota?

What Is Quota?

Quotas can limit the number or amount of resources available to users, such as the maximum number of ECSs or EVS disks that can be created.

If the existing resource quota cannot meet your service requirements, you can apply for a higher quota.

How Do I View My Quotas?

1. Log in to the management console.
2. Click 📍 in the upper left corner and select the desired region and project.
3. In the upper right corner of the page, choose **Resources** > **My Quotas**.

   The **Service Quota** page is displayed.
4. View the used and total quota of each type of resources on the displayed page.

   If a quota cannot meet service requirements, apply for a higher quota.

How Do I Apply for a Higher Quota?

1. Log in to the management console.
2. In the upper right corner of the page, choose **Resources** > **My Quotas**.

   The **Service Quota** page is displayed.
3. Click **Increase Quota** in the upper right corner of the page.
4. On the **Create Service Ticket** page, configure parameters as required.

   In the **Problem Description** area, fill in the content and reason for adjustment.
5. After all necessary parameters are configured, select **I have read and agree to the Ticket Service Protocol and Privacy Statement.** and click **Submit**.

3.2 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 3-1 Load balancing at Layer 4

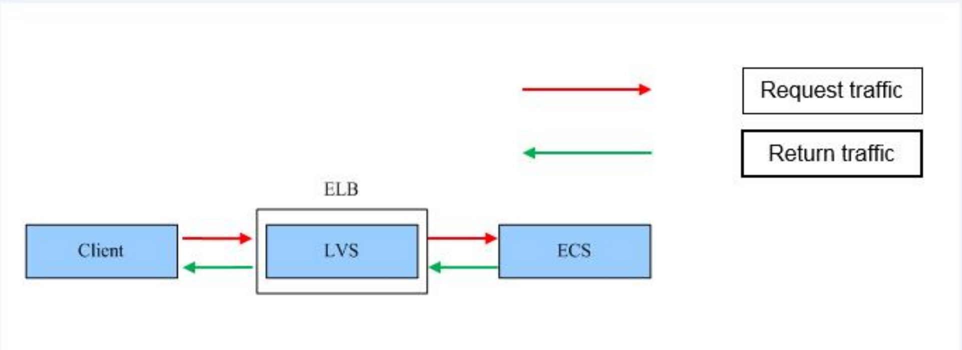
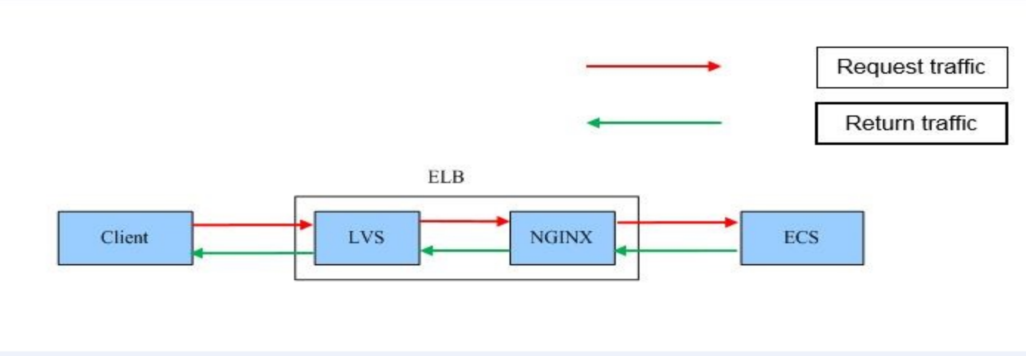


Figure 3-2 Load balancing at Layer 7



3.3 How Can I Access a Load Balancer Across VPCs?

VPC Peering can help you achieve this. For example, if another user has created load balancer ELB01 in VPC01, and you are in VPC02 and want to access ELB01, you just need to set up a VPC peering connection between VPC01 and VPC02 and add a route for the connection.

3.4 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**.

3.5 Why Can't I Delete My Load Balancer?

There may be resources associated with the load balancer. Delete these resources first.

Delete the resources configured for the load balancer in the following sequence:

1. Delete all the forwarding policies added to HTTP and HTTPS listeners of the load balancer.
2. Delete the redirect created for each HTTP listener of the load balancer.
3. Remove all the backend servers from the backend server groups associated with each listener of the load balancer.
4. Delete all the listeners added to the load balancer.
5. Delete all backend server groups associated with each listener of the load balancer.

3.6 Do I Need to Configure Bandwidth for My Load Balancers?

If you use a load balancer on a private network, you do not need to configure bandwidth. You only need to bind an EIP to the load balancer and configure bandwidth if you are using the load balancer on the Internet.

3.7 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*.

3.8 Why Multiple IP Addresses Are Required When I Create or Enable a Load Balancer?

These IP addresses are used by underlying resources.

Generally, 2 IP addresses are required for creating a load balancer in a single 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.

3.9 Why Are Requests from the Same IP Address Routed to Different Backend Servers When the Load Balancing Algorithm Is Source IP Hash?

One possible cause is that the backend server receiving requests from the client has become unhealthy. The source IP hash algorithm uses the source IP address of each request as a hashing key to route traffic from a particular client to the same backend server, as long as it is available. This allows requests from different clients to be routed based on their source IP addresses and ensures that a given client is always directed to the same backend server.

However, if a backend server become unhealthy and then recovers, ELB will generate a new hash key based on the source IP address of the request and numbers the backend server. As a result, requests from the same IP address are routed to different backend servers.

3.10 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.

3.11 Will Traffic Routing Be Interrupted If the Load Balancing Algorithm Is Changed?

No. If the load balancing algorithm is changed, established connections will not be affected. Therefore, traffic routing will not be interrupted.

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

The bandwidth included in the specifications of dedicated load balancers is the maximum value of the total inbound and outbound traffic of the load balancer. The bandwidth of the EIP bound to the load balancer is the limit for traffic required by the clients to access the load balancer.

3.13 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**.

4 Listeners

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

Sticky sessions ensure that requests from the same client are routed to the same backend server. **Table 4-1** lists the types of sticky sessions.

Table 4-1 Sticky sessions supported by 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

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.

4.2 Can I Bind Multiple Certificates to a Listener?

You can configure multiple certificates for an HTTPS listener by enabling SNI so that different certificates can be used for authentication based on the domain names of the requests.

For details, see **SNI Certificate (for HTTPS Listeners)**.

4.3 Do HTTP and HTTP Listeners Support the X-Forwarded-Host Header?

Yes. **Table 4-2** describes the HTTP header fields supported by HTTP and HTTP Listeners.

Table 4-2 Supported header fields

Field	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.

4.4 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.

4.5 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.

4.6 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.

4.7 How Is WebSocket Used?

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

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

Table 4-3 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 4-1 Timeout durations at Layer 7

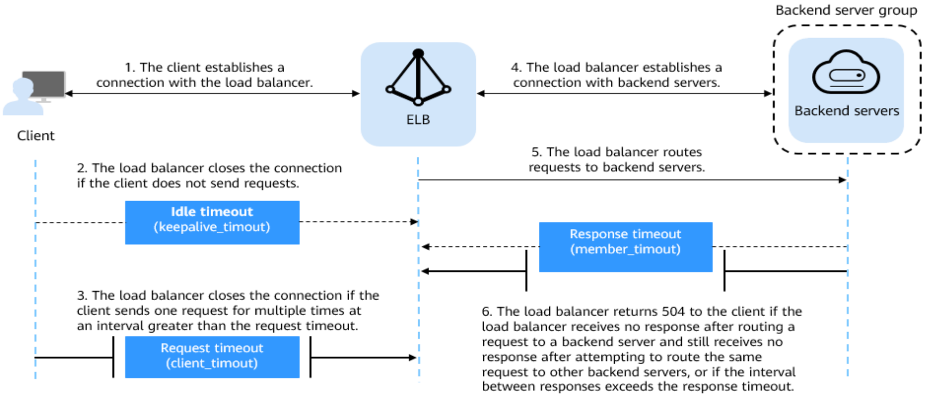

Backend server group
1. The client establishes a connection with the load balancer.
ELB
4. The load balancer establishes a connection with backend servers.
Backend servers
Client
5. The load balancer routes requests to backend servers.
2. The load balancer closes the connection if the client does not send requests.
Idle timeout (keepalive_timout)
Response timeout (member_timout)
3. The load balancer closes the connection if the client sends one request for multiple times at an interval greater than the request timeout.
Request timeout (client_timout)
6. The load balancer returns 504 to the client if the load balancer receives no response after routing a request to a backend server and still receives no response after attempting to route the same request to other backend servers, or if the interval between responses exceeds the response timeout.

Figure 4-2 Timeout durations at Layer 4

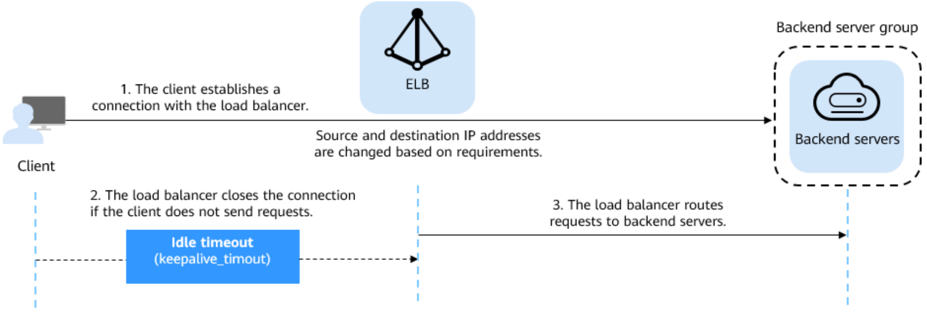

ELB
Backend server group
1. The client establishes a connection with the load balancer.
Backend servers
Source and destination IP addresses are changed based on requirements.
Client
2. The load balancer closes the connection if the client does not send requests.
3. The load balancer routes requests to backend servers.
Idle timeout (keepalive_timout)

Table 4-3 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	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

Protocol	Type	Description	Value Range	Default Timeout Duration
	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

4.9 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's protocol (frontend protocol). There are some constraints on the backend protocol when you associate a backend server group with a listener.

Table 4-4 Frontend and backend protocols of dedicated load balancers

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

Table 4-5 Frontend and backend protocols of shared load balancers

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

4.10 Why Cannot I Add a Listener to a Dedicated Load Balancer?

If you select either network load balancing (TCP/UDP) or application load balancing (HTTP/HTTPS) when creating the load balancer, you can only add listeners of the matched protocol.

The load balancing type cannot be changed after being selected. For example, if you have selected network load balancing during load balancer creation, you cannot change it to application load balancing and you cannot add HTTP or HTTPS listeners.

Table 4-6 Protocols and load balancing types

Load Balancing Type	Protocol	Listener Types
Network load balancing	TCP/UDP	TCP and UDP listeners
Application load balancing	HTTP/HTTPS	HTTP and HTTPS listeners

5 Backend Servers

5.1 Why Is the Interval at Which Backend Servers Receive Health Check Packets Different from What I Have Configured?

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

During this period, backend servers receive multiple detection packets from LVS and Nginx nodes. This makes it seem like backend servers are receiving packets at intervals shorter than the specified health check interval.

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

Yes. Backend servers can access the Internet whether or not they are associated with a load balancer.

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

- 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 Servers**.
- Database instances cannot be used as backend servers.
- Backend servers cannot work in active/standby mode.

5.4 Can ELB Route Traffic Across Regions?

- 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 **Configuring Hybrid Load Balancing**

5.5 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.

5.6 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.

5.7 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.

CAUTION

- 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 ACLs 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.

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

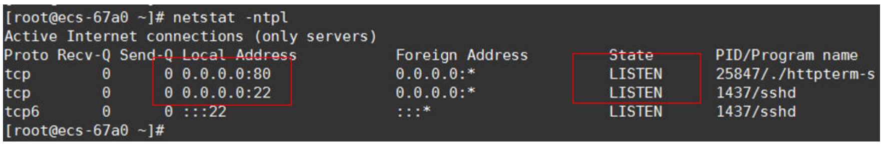
1. Verify that the applications on the backend server are enabled.
   a. Log in to the backend server. (An ECS is used as an example here.)
   b. Check the port status.

   netstat -ntpl

   NOTE

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

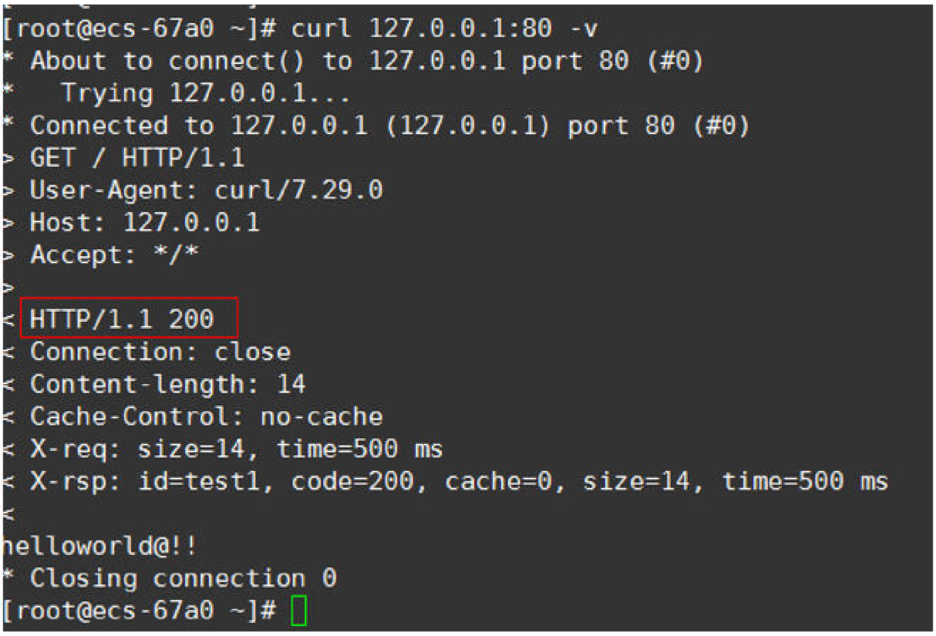
   Figure 5-1 Port status



```
[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.



```
[root@ecs-67a0 ~]# curl 127.0.0.1:80 -v
* About to connect() to 127.0.0.1 port 80 (#0)
*   Trying 127.0.0.1...
* Connected to 127.0.0.1 (127.0.0.1) port 80 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.29.0
> Host: 127.0.0.1
> Accept: */*
>
< HTTP/1.1 200
< Connection: close
< Content-length: 14
< Cache-Control: no-cache
< X-req: size=14, time=500 ms
< X-rsp: id=test1, code=200, cache=0, size=14, time=500 ms
<
helloworld@!!
* Closing connection 0
[root@ecs-67a0 ~]#
```

5.9 How Long Is the Timeout Duration of Connections Between a Load Balancer and Backend Servers?

The timeout duration varies depending on the protocol used by the listeners added to the load balancer. The default timeout durations are as follows:

- TCP listeners: 300s
- UDP listeners: 10s
- HTTP listeners: 60s

5.10 When Is a Backend Server Considered Healthy?

When a backend server is associated with a load balancer for the first time, the backend server is considered healthy after one health check. After this, the server is considered healthy only after the maximum number of health checks has been attempted.

5.11 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.

5.12 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.

5.13 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 ACLs or security group rules.

5.14 When Will Modified Weights Take Effect?

The new weights for backend servers take effect immediately after the weights are configured. Connections that have been established with 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.

5.15 How Much Time Is Required for a Load Balancer to Disconnect from Backend Servers After The Servers Are Removed?

The load balancer disconnects from the backend servers if connections have been established with these backend servers but no requests are sent over the connections before the timeout elapses.

NOTE

The timeout duration varies depending on the protocol used by the listeners added to the load balancer. The default timeout durations are as follows:

- TCP listeners: 300s
- UDP listeners: 10s
- HTTP listeners: 60s

5.16 Why Must the Subnet Where the Load Balancer Resides Have at Least 16 Available IP Addresses for Enabling IP as a Backend?

These IP addresses are used by the ELB system. Generally, two IP addresses are required for creating a dedicated load balancer in a single AZ, and six IP addresses are required for creating a dedicated load balancer with IP as a backend enabled. If you create a dedicated load balancer in multiple AZs, more IP addresses will be required. There is an algorithm to calculate how many.

6 Health Checks

6.1 How Do I Troubleshoot an Unhealthy Backend Server?

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.

- Dedicated load balancers

  On the **Load Balancers** page, click the name of the load balancer to view its details. Click **Backend Server Groups** and locate the server group. You can find the health check results for backend servers in the **Basic Information** area.

Background

Load balancers use the IP addresses from the backend subnet where the load balancers work to send heartbeat requests to backend servers. To ensure that health checks can be performed normally, you need to ensure that the IP addresses from the backend subnet where the load balancers work are allowed to access the backend servers.

⚠ CAUTION

- - Load balancers: Ensure that security group rules allow access from IP addresses in the VPC where the backend server resides. For details about how to configure security groups for backend servers associated with load balancers, see **Configuring Security Group Rules for Backend Servers**.
- Dedicated load balancer: 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 VPC where the backend subnet of the load balancer works to the backend servers associated with TCP or UDP listener.

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

NOTE

- When a backend server is detected as unhealthy, the load balancer will stop routing requests to this server.
- If health checks are disabled, the load balancer will consider the backend server healthy by default and still route requests to it.
- If **Transfer Client IP Address** is enabled for TCP and UDP listeners of both dedicated and shared load balancers, client IP addresses instead of IP addresses in 100.125.0.0/16 are used to communicate with the backend server.
- Traffic will not be routed to a backend server with a weight of 0, so the health check result for this backend server is not relevant.

Troubleshooting

Possible causes are described here in order of how likely they are to occur.

Check these causes one by one until you find the cause of this issue.

NOTE

If you need to change the health check configuration, it takes a while for the changes to be applied. The required time depends on the health check interval and timeout duration. You can find the health check results in the backend server list of the load balancer.

Figure 6-1 Troubleshooting process

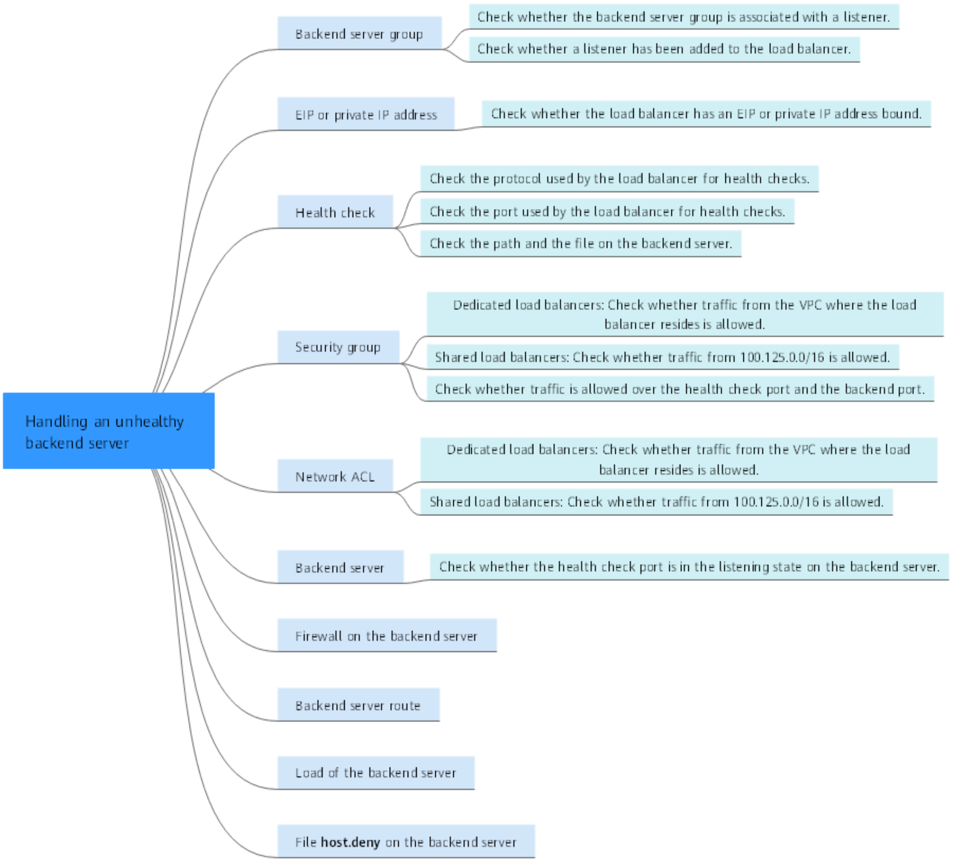

Handling an unhealthy backend server
Backend server group
Check whether the backend server group is associated with a listener.
Check whether a listener has been added to the load balancer.
EIP or private IP address
Check whether the load balancer has an EIP or private IP address bound.
Health check
Check the protocol used by the load balancer for health checks.
Check the port used by the load balancer for health checks.
Check the path and the file on the backend server.
Security group
Dedicated load balancers: Check whether traffic from the VPC where the load balancer resides is allowed.
Shared load balancers: Check whether traffic from 100.125.0.0/16 is allowed.
Check whether traffic is allowed over the health check port and the backend port.
Network ACL
Dedicated load balancers: Check whether traffic from the VPC where the load balancer resides is allowed.
Shared load balancers: Check whether traffic from 100.125.0.0/16 is allowed.
Backend server
Check whether the health check port is in the listening state on the backend server.
Firewall on the backend server
Backend server route
Load of the backend server
File host.deny on the backend server

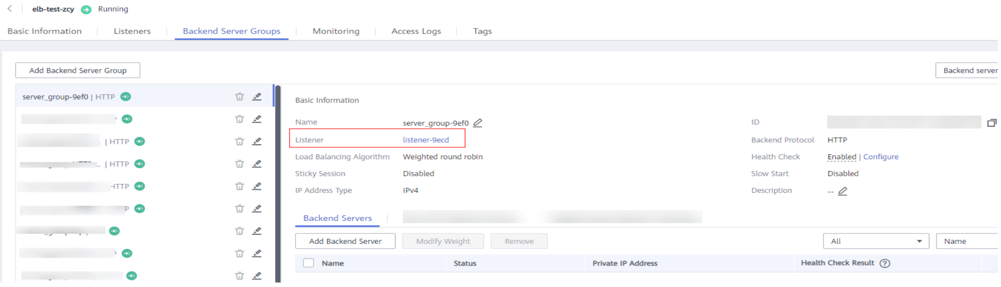
Table 6-1 Troubleshooting process

Possible Cause	Solution
Backend server group	**Checking Whether the Backend Server Group Is Associated with a Listener**
EIP or private IP address	**Checking Whether an EIP or a Private IP Address Is Bound to the Load Balancer**
Health check configuration	**Checking the Health Check Configuration**
Security group rules	**Checking Security Group Rules**
Network ACL rules	**Checking Network ACL Rules**
Backend server listening configuration	**Checking the Backend Server**
Backend server firewall configuration	**Checking the Firewall on the Backend Server**
Backend server route configuration	**Checking the Backend Server Route**
Backend server load	**Checking the Backend Server Load**
Backend server **host.deny** file	**Checking the host.deny File**

Checking Whether the Backend Server Group Is Associated with a Listener

Check whether the backend server group that the unhealthy backend server belongs to is associated with a listener.

Figure 6-2 Checking the listener with the backend server group associated



elb-test-zcy Running

Basic Information

Name server_group-9ef0

Listener listener-9ecd

Load Balancing Algorithm Weighted round robin

Sticky Session Disabled

IP Address Type IPv4

ID [redacted]

Backend Protocol HTTP

Health Check Enabled | Configure

Slow Start Disabled

Description --

Name	Status	Private IP Address	Health Check Result

- If the backend server group is not associated with a listener, check whether a listener has been added to the load balancer.
  - If there is a listener, associate the backend server group with the listener.
  - If there are no listeners, add a listener. Select **Use existing** and then select the backend server group when you add the listener.
- If the backend server group has been associated with a listener, proceed with the following operations.

Checking Whether an EIP or a 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 an HTTP or HTTPS listener to the load balancer, health checks will not be affected no matter whether an EIP or private IP address is bound to the load balancer.

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

If the load balancer has no EIP or private IP address bound, bind one.

NOTE

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**. After you bind an EIP or private IP address to the load balancer, the health check result becomes **Healthy**. If you unbind the EIP or private IP address from the load balancer, the health check result is still **Healthy**.

Checking the Health Check Configuration

Click the name of the load balancer to view its details. On the **Backend Server Group** tab page, click the name of the backend server group. In the **Basic Information** area, to the right of **Health Check**, click **Configure** and then check the following parameters:

- **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 protocol used for health checks.
- **Port** The port must be the one used on the backend server, and it cannot be changed. Check whether the health check port is in the listening state on the backend server. If the health check port is not in the listening state on the backend server, the backend server will be identified as unhealthy.
- **Check 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, the port and the path are used for health checks.
- If the health check protocol is TCP, only the port is used for health checks.
- If 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.

  For example:

  If the URL is **http://www.example.com** or **http://192.168.63.187:9096**, the health check path is **/**.

  If the URL is **http://www.example.com/chat/try/**, the health check path is **/chat/try/**.

  If the URL is **http://192.168.63.187:9096/chat/index.html**, the health check path is **/chat/index.html**.

Checking Security Group Rules

- **TCP, HTTP, or HTTPS listeners**: Verify that the inbound security group rule allows TCP traffic from the VPC where the load balancer works to the backend server over the health check port.

  ⚠ CAUTION

  - Dedicated load balancer: 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 VPC where the backend subnet of the load balancer works to the backend servers associated with TCP or UDP listener.

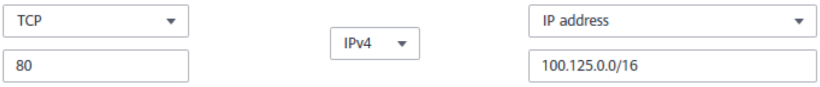
  – If the health check port is the same as the backend port, the inbound rule must allow traffic over the backend port, for example, port 80.

  – If the port (port 80 as an example) for health check is different from that used by the backend server (port 443 as an example), inbound security group rules must allow traffic over the both ports.

    NOTE

    You can check the protocol and port in the **Basic Information** area of the backend server group.

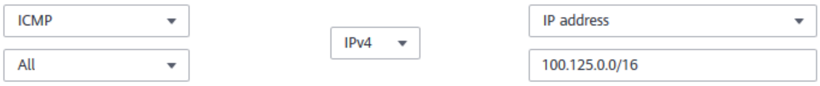
  Figure 6-3 Example inbound rule

  TCP
  80
  IPv4
  IP address
  100.125.0.0/16

- **UDP listeners**: Verify that the inbound security group rule allows traffic from the VPC where the load balancer works to the backend server using the health check protocol and over the health check port. In addition, inbound ICMP traffic must be allowed.

  Figure 6-4 Example inbound rule that allows ICMP traffic

  ICMP
  All
  IPv4
  IP address
  100.125.0.0/16

NOTE

- If you are not sure about the security group rules, change the protocol and port range to **All** for testing.
- For UDP listeners, see **How Does ELB Perform UDP Health Checks? What Are the Precautions for UDP Health Checks?**

Checking Network ACL Rules

⚠ CAUTION

- Dedicated load balancer: 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 VPC where the backend subnet of the load balancer works to the backend servers associated with TCP or UDP listener.

- **Load balancers**

  To control traffic in and out of a subnet, you can associate a network ACL with the subnet. Similar to security groups, network ACLs control access to subnets and add an additional layer of defense to your subnets. Default network ACL rules reject all inbound and outbound traffic. If the subnet of a load balancer or associated backend servers has a network ACL associated, the load balancer cannot receive traffic from the Internet or route traffic to backend servers, and backend servers cannot receive traffic from and respond to the load balancer.

  Configure an inbound network ACL rule to allow traffic from the VPC where the load balancer works to backend servers.

  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 ☰ in the upper left corner of the page and choose **Networking** > **Virtual Private Cloud**.

  d. In the navigation pane on the left, choose **Access Control** > **Network ACL**.

  e. In the network ACL list, click the name of the network ACL to switch to the page showing its details.

  f. On the **Inbound Rules** or **Outbound Rules** tab page, click **Add Rule** to add a rule.

     ▪ **Action**: Select **Allow**.

     ▪ **Protocol**: The protocol must be the same as the one you selected for the listener.

     ▪ **Source**: Set it to the VPC CIDR block.

     ▪ **Source Port Range**: Select a port range.

     ▪ **Destination**: If you keep the default value, **0.0.0.0/0**, traffic will be allowed for all destination IP addresses.

     ▪ **Destination Port Range**: Select a port range.

     ▪ (Optional) **Description**: Describe the network ACL rule.

g. Click **OK**.

Checking the Backend Server

NOTE

If the backend server runs a Windows OS, 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.

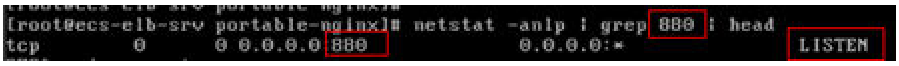
- Run the following command on the backend server to check whether the health check port is listened on:

```
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 6-5**, TCP port 880 is listened on.

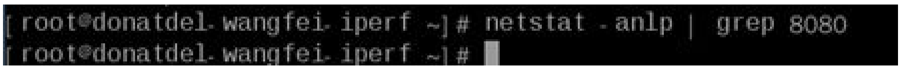
If you do not specify a health check port, backend ports are used by default.

Figure 6-5 Backend server port listened on



```
[root@ecs-elb-srv portable-nginx]# netstat -anlp | grep 880 | head
tcp        0      0 0.0.0.0:880              0.0.0.0:*               LISTEN
```

Figure 6-6 Backend server port not listened on



```
[root@donatdel-wangfei-iperf ~]# netstat -anlp | grep 8080
[root@donatdel-wangfei-iperf ~]#
```

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:

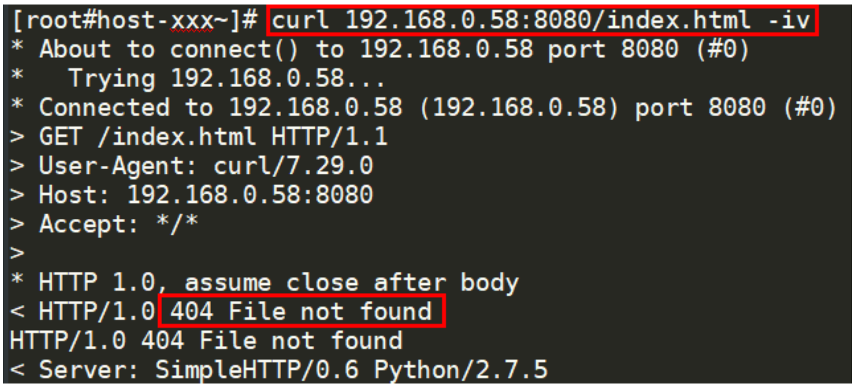
```
curl {Private IP address of the backend server}:{Health check port}/{Health check path} -iv
```

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

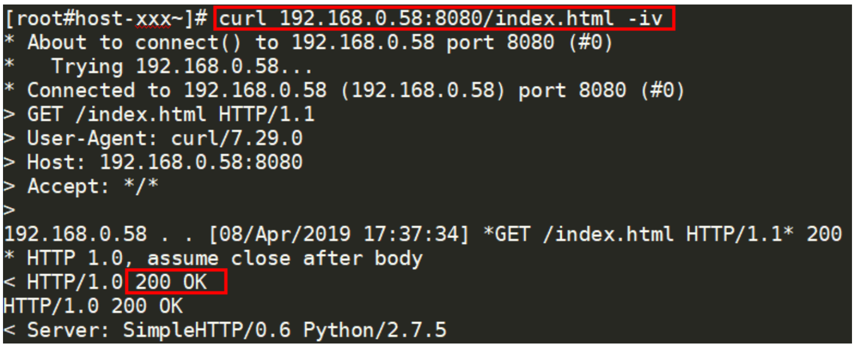
The status code is 200, 202, or 401 if the backend server is healthy.

Figure 6-7 Unhealthy 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: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 404 File not found
HTTP/1.0 404 File not found
< Server: SimpleHTTP/0.6 Python/2.7.5
```

Figure 6-8 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 HTTP is used for health checks and the backend server is detected unhealthy, perform the following steps to configure a TCP health check:

  On the **Listeners** tab page, modify the 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 in the backend server, the software may block the IP addresses from the VPC where the load balancers work. Configure inbound firewall rules to allow traffic from the VPC CIDR block to the backend servers.

Checking the Backend Server Route

Check whether the default route configured for the primary NIC 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):

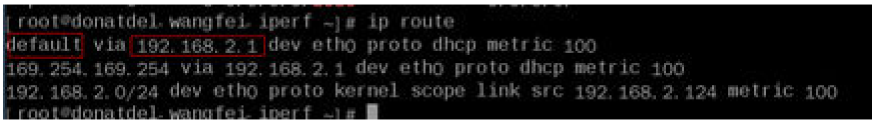
```
ip route
```

Alternatively, run the following command:

```
route -n
```

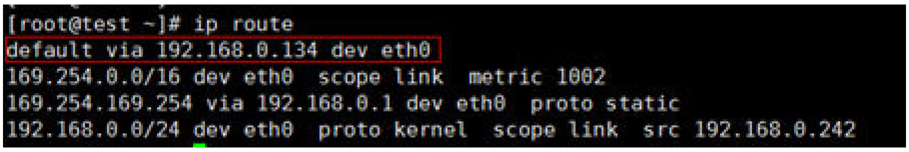
Figure 6-9 shows the command output when the backend server route is normal.

Figure 6-9 Example default route pointing to the gateway



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

Figure 6-10 Example default route not pointing to the gateway



```
[root@test ~]# ip route
default via 192.168.0.134 dev eth0
169.254.0.0/16 dev eth0  scope link  metric 1002
169.254.169.254 via 192.168.0.1 dev eth0  proto static
192.168.0.0/24 dev eth0  proto kernel  scope link  src 192.168.0.242
```

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 load is high, connections or requests for health checks may time out.

Checking the host.deny File

Verify that IP addresses from the VPC where the load balancers work are not written into the **/etc/hosts.deny** file.

Submitting a Service Ticket

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

6.2 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.

6.3 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.

6.4 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**.

6.5 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.

6.6 Do Maximum Retries Include Health Checks That Consider Backend Servers Unhealthy?

Yes. Maximum retries are the maximum number of health checks after which a backend server is detected healthy or the maximum number of health checks after which the same backend server is detected unhealthy.

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

1. You can increase the health check interval as follows:

   Log in to the management console.

   Locate the load balancer and click its name.

   On the **Backend Server Groups** tab page, locate the backend server group.

   In the **Basic Information** area, click **Configure** next to **Health Check**.

Change the health check interval to a larger value.

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 function as follows:

   Log in to the management console.

   Locate the load balancer and click its name.

   On the **Backend Server Groups** tab page, locate the backend server group.

   In the **Basic Information** area, click **Configure** next to **Health Check**.

   Disable the 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.

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

Table 6-2 Status Code

Load Balancer Type	Health Check Protocol	Status Code
Load balancers	HTTP	200
	HTTPS	200

7 Obtaining Source IP Addresses

7.1 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**.

Constraints and Limitations

- 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.

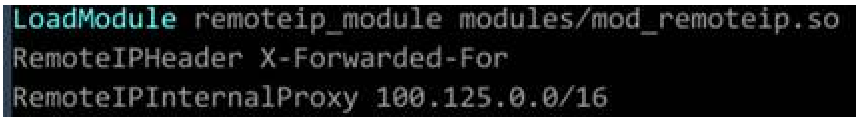
   For example, if CentOS 7.5 is used as the OS, run the following command to install the software:

   ```
   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 7-1 Content to be added

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

   NOTE

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

   - 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:

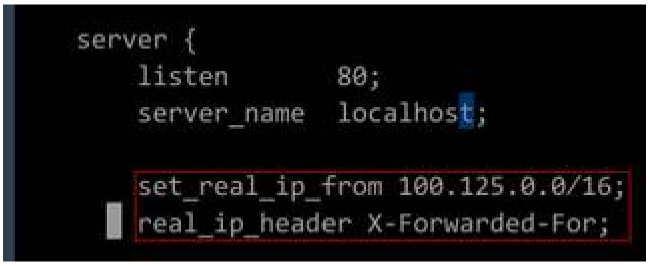
   ```
   vi /path/server/nginx/conf/nginx.conf
   ```

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 7-2 Adding information

   ```
   server {
       listen        80;
       server_name   localhost;

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

   NOTE

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

   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
   ```

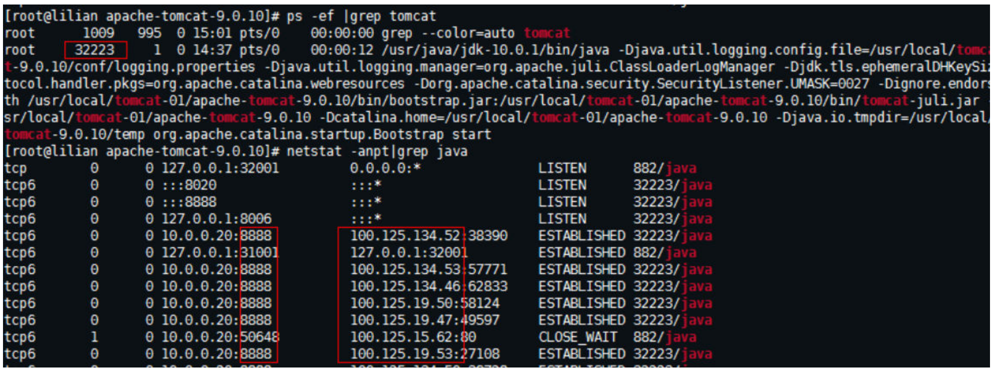
Tomcat Servers

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 7-3 Tomcat running properly

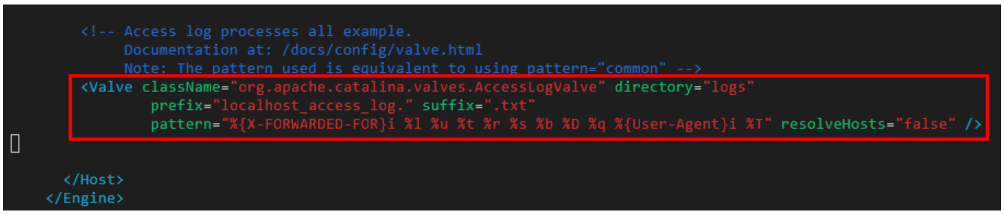
   

   ```
   [root@lilian apache-tomcat-9.0.10]# ps -ef |grep tomcat
   root      1009   995  0 15:01 pts/0    00:00:00 grep --color=auto tomcat
   root     32223     1  0 14:37 pts/0    00:00:12 /usr/java/jdk-10.0.1/bin/java -Djava.util.logging.config.file=/usr/local/tomc
   t-9.0.10/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySiz
   tocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Dignore.endors
   th /usr/local/tomcat-01/apache-tomcat-9.0.10/bin/bootstrap.jar:/usr/local/tomcat-01/apache-tomcat-9.0.10/bin/tomcat-juli.jar
   sr/local/tomcat-01/apache-tomcat-9.0.10 -Dcatalina.home=/usr/local/tomcat-01/apache-tomcat-9.0.10 -Djava.io.tmpdir=/usr/local/
   tomcat-9.0.10/temp org.apache.catalina.startup.Bootstrap start
   [root@lilian apache-tomcat-9.0.10]# netstat -anpt|grep java
   tcp        0      0 127.0.0.1:32001         0.0.0.0:*               LISTEN      882/java
   tcp6       0      0 :::8020                 :::*                    LISTEN      32223/java
   tcp6       0      0 :::8888                 :::*                    LISTEN      32223/java
   tcp6       0      0 127.0.0.1:8006          :::*                    LISTEN      32223/java
   tcp6       0      0 10.0.0.20:8888          100.125.134.52:38390    ESTABLISHED 32223/java
   tcp6       0      0 127.0.0.1:31001         127.0.0.1:32001         ESTABLISHED 882/java
   tcp6       0      0 10.0.0.20:8888          100.125.134.53:57771    ESTABLISHED 32223/java
   tcp6       0      0 10.0.0.20:8888          100.125.134.46:62833    ESTABLISHED 32223/java
   tcp6       0      0 10.0.0.20:8888          100.125.19.50:58124     ESTABLISHED 32223/java
   tcp6       0      0 10.0.0.20:8888          100.125.19.47:49597     ESTABLISHED 32223/java
   tcp6       1      0 10.0.0.20:50648         100.125.15.62:80        CLOSE_WAIT  882/java
   tcp6       0      0 10.0.0.20:8888          100.125.19.53:27108     ESTABLISHED 32223/java
   ```

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 7-4 Example configuration



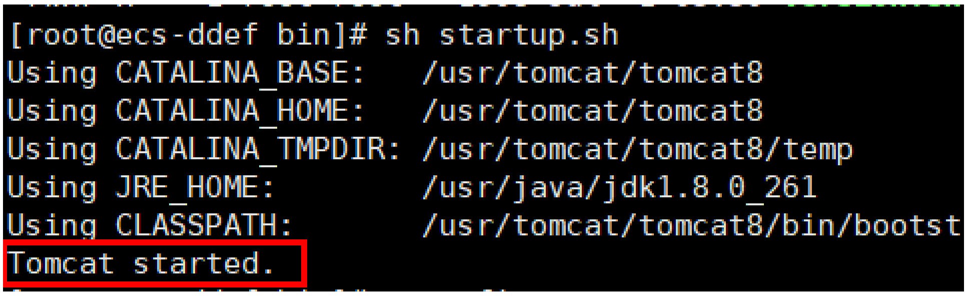
```
        <!-- Access log processes all example.
             Documentation at: /docs/config/valve.html
             Note: The pattern used is equivalent to using pattern="common" -->
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
               prefix="localhost_access_log." suffix=".txt"
               pattern="%{X-FORWARDED-FOR}i %l %u %t %r %s %b %D %q %{User-Agent}i %T" resolveHosts="false" />

      </Host>
    </Engine>
```

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 7-5 Restarting the Tomcat service



```
[root@ecs-ddef bin]# sh startup.sh
Using CATALINA_BASE:   /usr/tomcat/tomcat8
Using CATALINA_HOME:   /usr/tomcat/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat/tomcat8/temp
Using JRE_HOME:        /usr/java/jdk1.8.0_261
Using CLASSPATH:       /usr/tomcat/tomcat8/bin/bootst
Tomcat started.
```

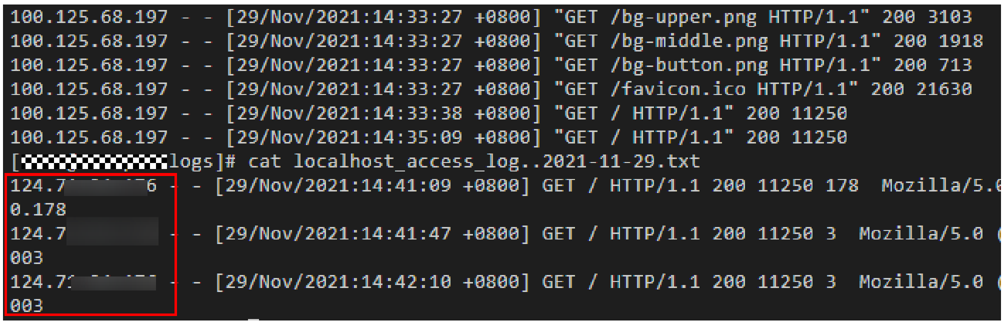
5. View the latest logs.

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 7-6 Querying the source IP address



```
100.125.68.197 - - [29/Nov/2021:14:33:27 +0800] "GET /bg-upper.png HTTP/1.1" 200 3103
100.125.68.197 - - [29/Nov/2021:14:33:27 +0800] "GET /bg-middle.png HTTP/1.1" 200 1918
100.125.68.197 - - [29/Nov/2021:14:33:27 +0800] "GET /bg-button.png HTTP/1.1" 200 713
100.125.68.197 - - [29/Nov/2021:14:33:27 +0800] "GET /favicon.ico HTTP/1.1" 200 21630
100.125.68.197 - - [29/Nov/2021:14:33:38 +0800] "GET / HTTP/1.1" 200 11250
100.125.68.197 - - [29/Nov/2021:14:35:09 +0800] "GET / HTTP/1.1" 200 11250
[[redacted]logs]# cat localhost_access_log..2021-11-29.txt
124.7[redacted] - [29/Nov/2021:14:41:09 +0800] GET / HTTP/1.1 200 11250 178  Mozilla/5.
0.178
124.7[redacted] - [29/Nov/2021:14:41:47 +0800] GET / HTTP/1.1 200 11250 3  Mozilla/5.0
003
124.7[redacted] - [29/Nov/2021:14:42:10 +0800] GET / HTTP/1.1 200 11250 3  Mozilla/5.0
003
```

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 7-7 Selecting modules

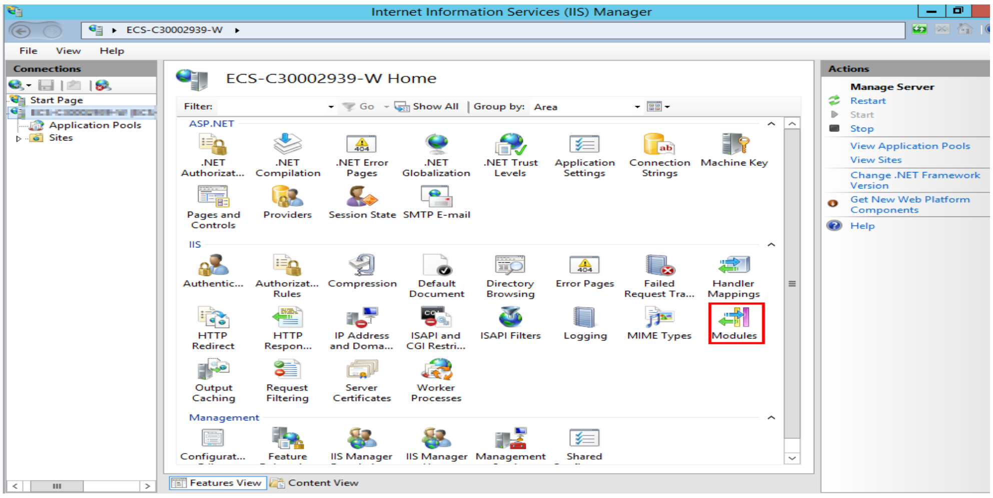
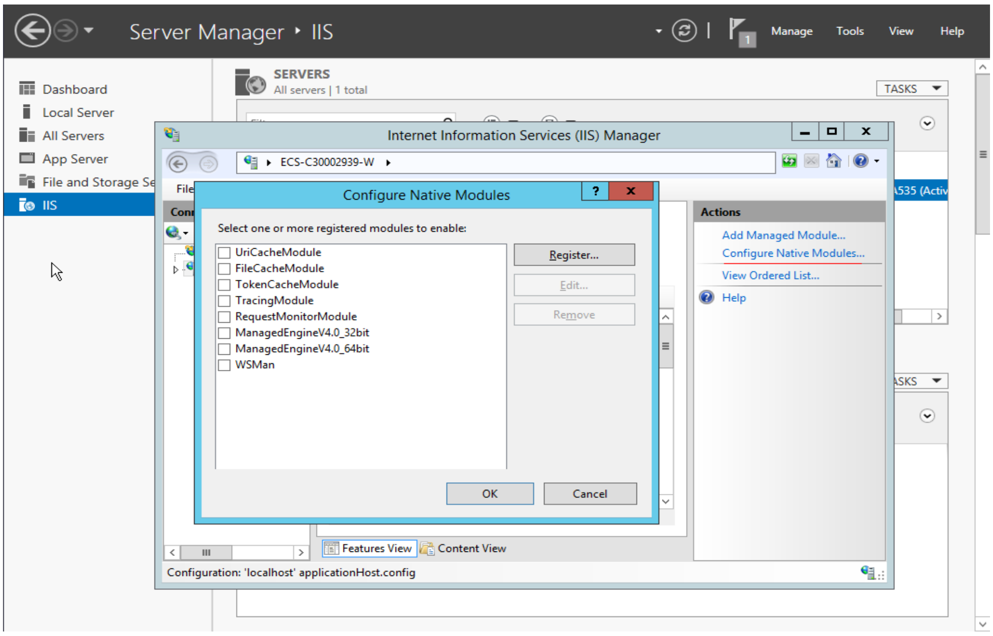


Figure 7-8 Configure Native Modules

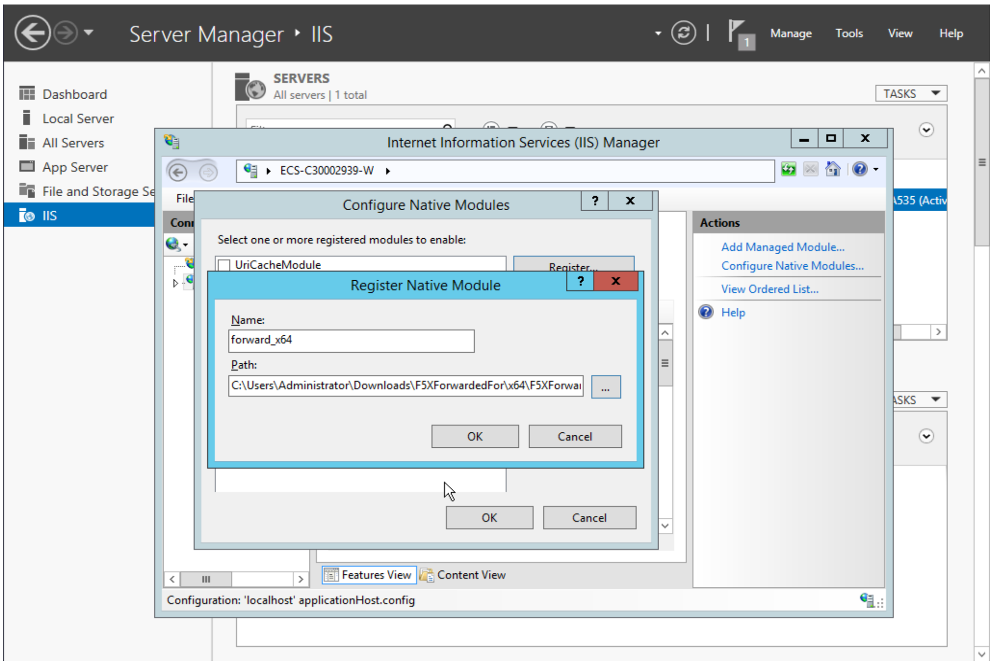


Select one or more registered modules to enable:

- □ UriCacheModule
- □ FileCacheModule
- □ TokenCacheModule
- □ TracingModule
- □ RequestMonitorModule
- □ ManagedEngineV4.0_32bit
- □ ManagedEngineV4.0_64bit
- □ WSMan

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

Figure 7-9 Registering plug-ins



Select one or more registered modules to enable:

□ UriCacheModule

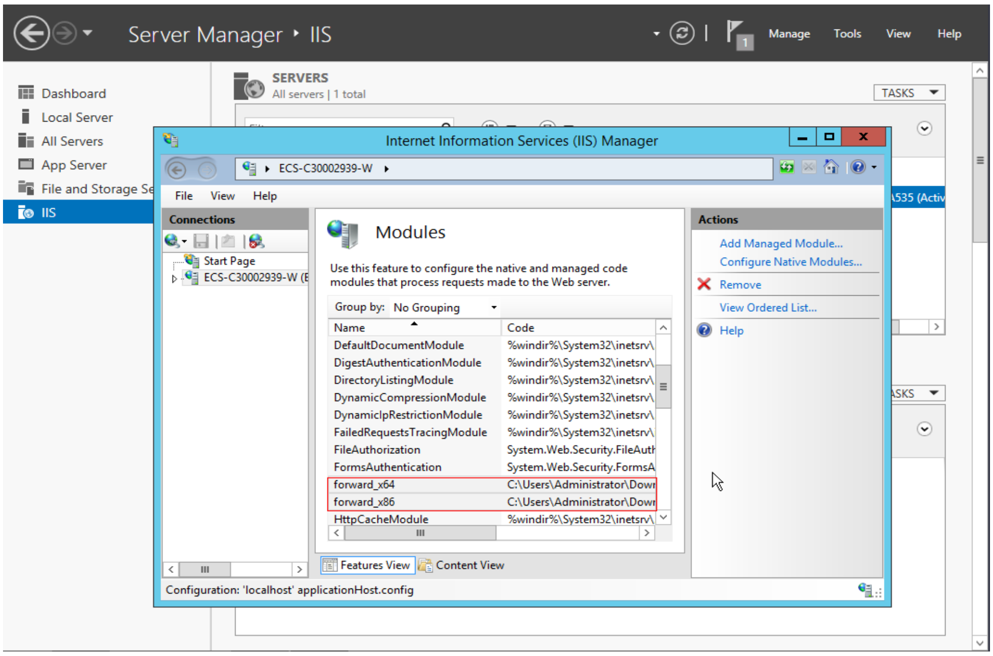
Name:
forward_x64

Path:
C:\Users\Administrator\Downloads\F5XForwardedFor\x64\F5XForwar

Configuration: 'localhost' applicationHost.config

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

Figure 7-10 Confirming the registration



Modules

Use this feature to configure the native and managed code modules that process requests made to the Web server.

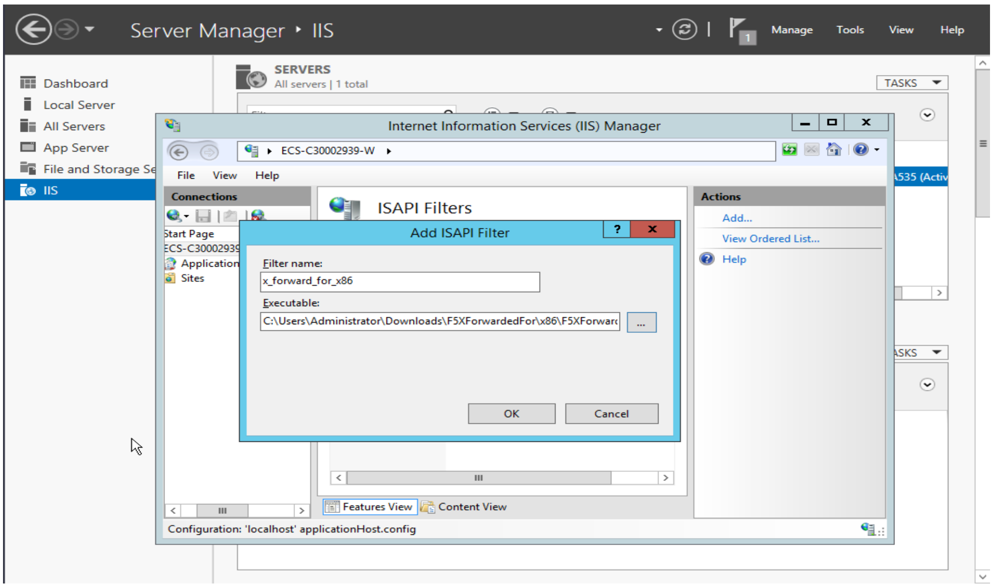
Group by: No Grouping

Name	Code
DefaultDocumentModule	%windir%\System32\inetsrv\
DigestAuthenticationModule	%windir%\System32\inetsrv\
DirectoryListingModule	%windir%\System32\inetsrv\
DynamicCompressionModule	%windir%\System32\inetsrv\
DynamicIpRestrictionModule	%windir%\System32\inetsrv\
FailedRequestsTracingModule	%windir%\System32\inetsrv\
FileAuthorization	System.Web.Security.FileAuth
FormsAuthentication	System.Web.Security.FormsA
forward_x64	C:\Users\Administrator\Dowr
forward_x86	C:\Users\Administrator\Dowr
HttpCacheModule	%windir%\System32\inetsrv\

Configuration: 'localhost' applicationHost.config

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

Figure 7-11 Adding authorization



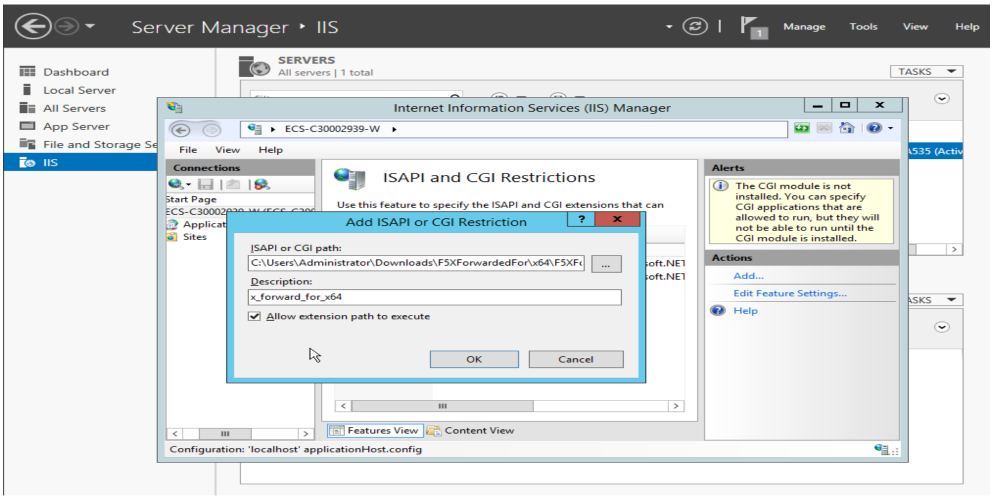
Add ISAPI Filter

Filter name: x_forward_for_x86

Executable: C:\Users\Administrator\Downloads\F5XForwardedFor\x86\F5XForwarc

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

Figure 7-12 Allowing the plug-ins to execute



Use this feature to specify the ISAPI and CGI extensions that can

The CGI module is not installed. You can specify CGI applications that are allowed to run, but they will not be able to run until the CGI module is installed.

Add ISAPI or CGI Restriction

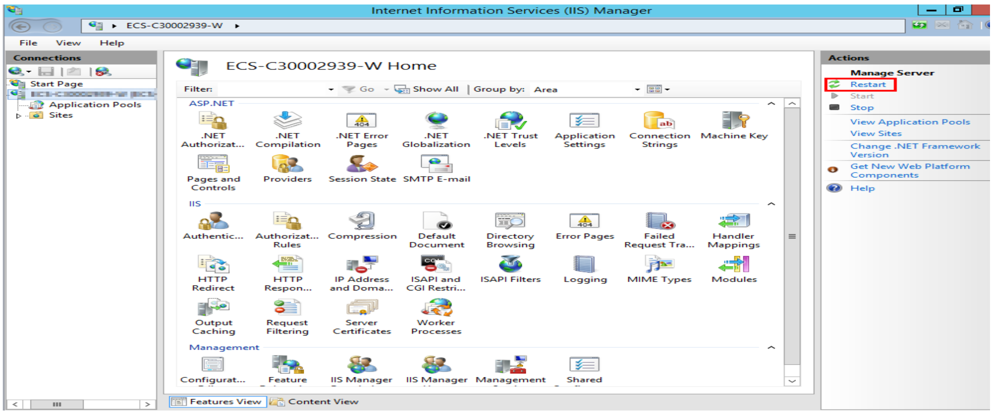
ISAPI or CGI path: C:\Users\Administrator\Downloads\F5XForwardedFor\x64\F5XF(

Description: x_forward_for_x64

☑ Allow extension path to execute

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

Figure 7-13 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 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. Hover on 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 be used as both the client and the backend server. 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 plug-in to obtain real IP addresses. For details, see **Configuring the TOA Plug-in**.

7.2 How Can an EIP Bound to a Load Balancer Be Transmitted to Backend Servers?

When you add an HTTPS or HTTP listener, enable **Obtain Load Balancer EIP** to store the EIP bound to the load balancer in the HTTP header and transmit it to backend servers. For details, see *Obtaining the Load Balancer EIP*.

8 HTTP/HTTPS Listeners

8.1 Which Protocol Should I Select for the Backend Server Group When Adding an HTTPS Listener?

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

To use HTTPS at the frontend or backend, you can create a load balancer, and set the backend protocol to HTTP.

8.2 Why Is There a Security Warning After a Certificate Is Configured?

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.

8.3 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.

8.4 Why Can't I Add a Forwarding Policy to a Listener?

Check the listener's protocol.

Forwarding policies can only be added to HTTP and HTTPS listeners.

8.5 Why Cannot I Select an Existing Backend Server Group When Adding a Forwarding Policy?

This is because the backend server group has been used by another forwarding policy. A backend server group can be used by only one forwarding policy.

9 Sticky Sessions

9.1 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.

9.2 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).

9.3 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.

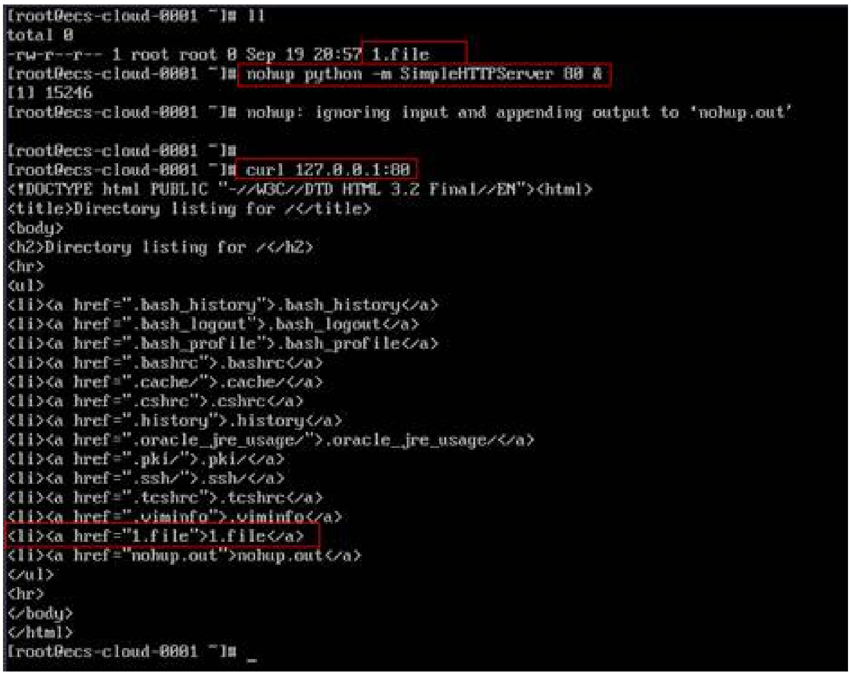
Log in to a backend server and create a file named **1.file** in the current directory to mark this server.

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



```
[root@ecs-cloud-0001 ~]# ll
total 0
-rw-r--r-- 1 root root 0 Sep 19 20:57 1.file
[root@ecs-cloud-0001 ~]# nohup python -m SimpleHTTPServer 80 &
[1] 15246
[root@ecs-cloud-0001 ~]# nohup: ignoring input and appending output to 'nohup.out'

[root@ecs-cloud-0001 ~]#
[root@ecs-cloud-0001 ~]# 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="1.file">1.file</a>
<li><a href="nohup.out">nohup.out</a>
</ul>
<hr>
</body>
</html>
[root@ecs-cloud-0001 ~]# _
```

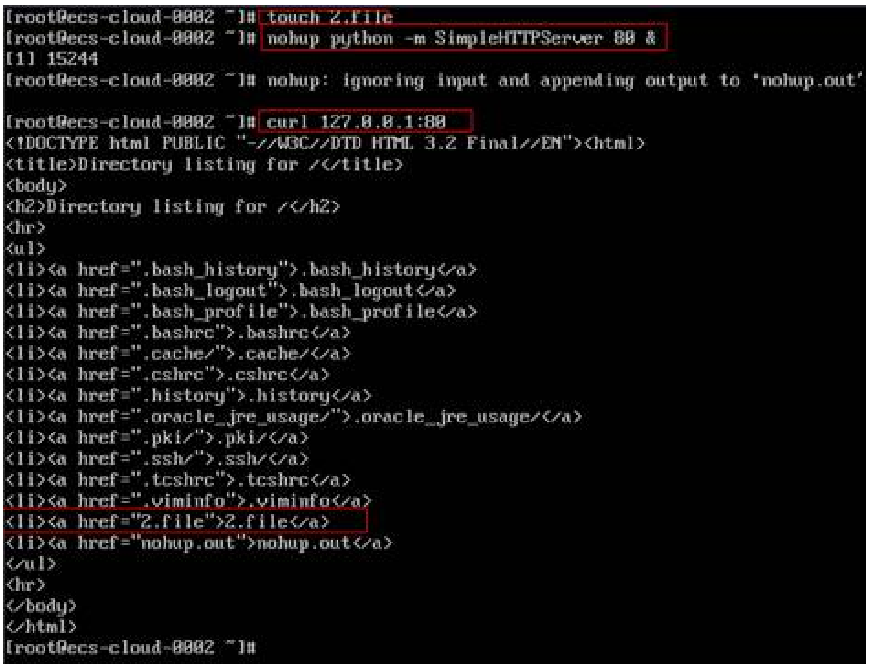
Log in to the other backend server and create a file named **2.file** in the current directory.

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

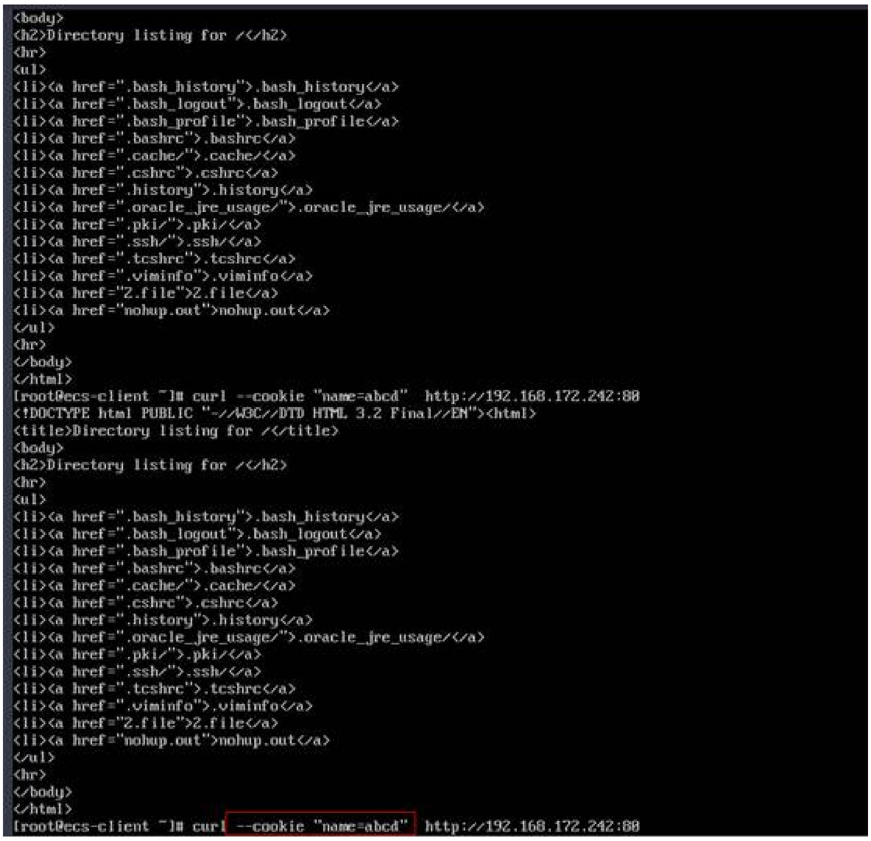


```
[root@ecs-cloud-0002 ~]# touch 2.file
[root@ecs-cloud-0002 ~]# nohup python -m SimpleHTTPServer 80 &
[1] 15244
[root@ecs-cloud-0002 ~]# nohup: ignoring input and appending output to 'nohup.out'

[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
```

9.4 What Types of Sticky Sessions Does ELB Support?

ELB supports sticky sessions source IP address and load balancer cookie.

10 Certificates

10.1 How Can I Create Server Certificates and CA Certificates?

Refer to **Mutual Authentication** to create server certificates and CA certificates. Generally, only backend servers need to be authenticated. You only need to configure server certificates.

10.2 Does ELB Support Wildcard Certificates?

Yes. If the domain name of a wildcard certificate is *.test.com, domain names a.test.com, b.test.com, a.b.test.com, and c.d.test.com are supported.

10.3 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.

10.4 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.

11 Monitoring

11.1 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.

11.2 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.

12 Billing

12.1 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.

12.2 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.

12.3 Can I Modify the Bandwidth of a Load Balancer?

Yes. For details, see **Modifying the Bandwidth**.

12.4 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.