

Elastic Load Balance

Best Practices

Issue 01
Date 2024-11-12



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

Contents

1 Using IP as a Backend to Route Traffic Across Backend Servers.....	1
1.1 Overview.....	1
1.2 Routing Traffic to Backend Servers in a Different VPC from the Load Balancer.....	3
1.3 Routing Traffic to Backend Servers in the Same VPC as the Load Balancer.....	8
2 Using Advanced Forwarding for Application Iteration.....	14
3 Integrating WAF with ELB to Protect Your Websites.....	23
4 Configuring HTTPS Mutual Authentication to Improve Service Security.....	30
5 Using ELB to Redirect HTTP Requests to an HTTPS Listener for Higher Service Security.....	38

1 Using IP as a Backend to Route Traffic Across Backend Servers

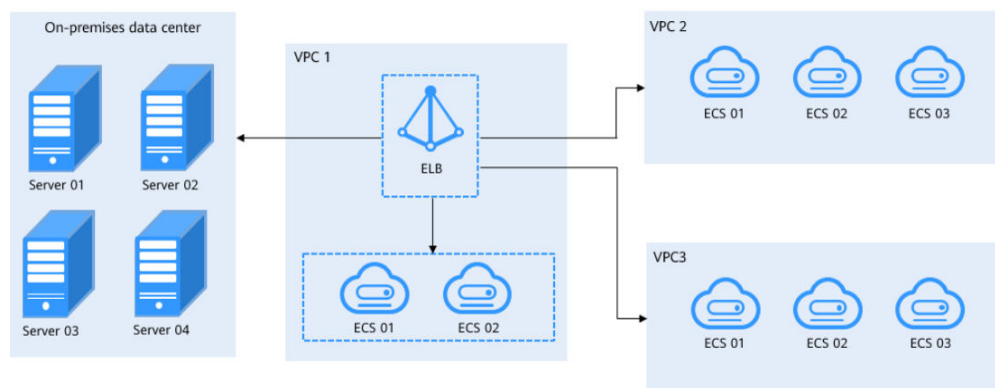
1.1 Overview

Scenarios

You have servers both in VPCs and your on-premises data center and want to use load balancers to distribute incoming traffic across these servers.

This section describes how you can use a dedicated load balancer to route incoming traffic across cloud and on-premises servers.

Figure 1-1 Routing traffic across cloud and on-premises servers



Solution

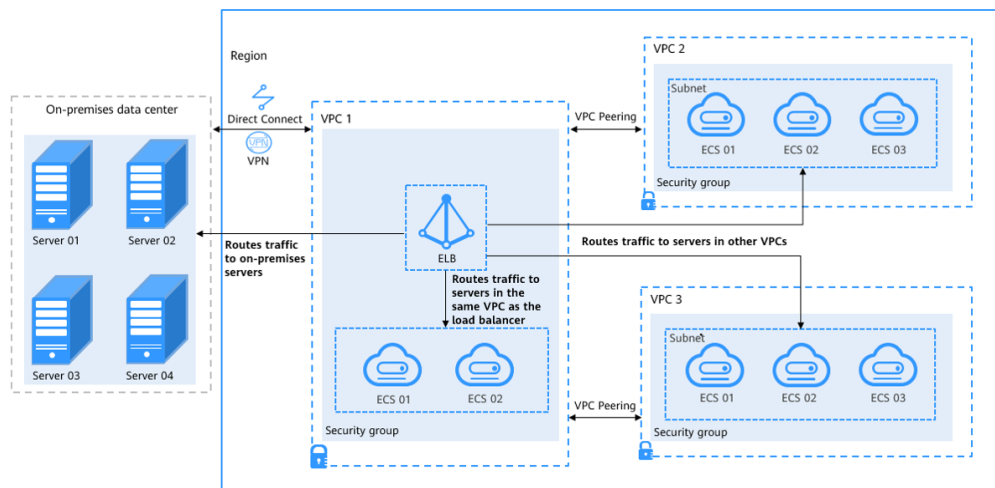
You can enable **IP as a Backend** when creating a dedicated load balancer and associate cloud and on-premises servers with this dedicated load balancer using their IP addresses.

As shown in [Figure 1-2](#), ELB can realize hybrid load balancing.

- You can associate the servers in the same VPC as the load balancer no matter whether you enable **IP as a Backend**.

- If you enable **IP as a Backend**:
 - You can associate on-premises servers with the load balancer after the on-premises data center is connected to the cloud through Direct Connect or VPN.
 - You can also associate the servers in other VPCs different from the load balancer after the VPCs are connected to the VPC where the load balancer is running over VPC peering connections.
 - You can associate the servers in the same VPC as the load balancer.

Figure 1-2 Associating servers with the load balancer



Advantages

You can add servers in the VPC where the load balancer is created, in a different VPC, or in an on-premises data center, by using private IP addresses of the servers to the backend server group of the load balancer. In this way, incoming traffic can be flexibly distributed to cloud servers and on-premises servers for hybrid load balancing.

- You can add backend servers in the same VPC as the load balancer.
- You can add servers in other VPCs different from the load balancer by establishing a VPC peering connection between the two VPCs.
- You can add on-premises servers by connecting your on-premises data center to the cloud through Direct Connect or VPN.

Notes and Constraints

When you add IP as backend servers, note the following:

- Enable **IP as a Backend** first on the **Summary** page of the load balancer.
- IP as backend servers must use IPv4 addresses.
- To ensure requests are properly routed, IP as backend servers cannot use public IP addresses.
- To add IP as backend servers, the subnet where the load balancer works must have at least 16 available IP addresses. You can add more subnets for more IP addresses on the **Summary** page of the load balancer.

- To ensure normal health checks, security group rules configured for IP as backend servers must allow traffic from the backend subnet of the load balancer.
- **IP as a Backend** cannot be disabled after it is enabled.

1.2 Routing Traffic to Backend Servers in a Different VPC from the Load Balancer

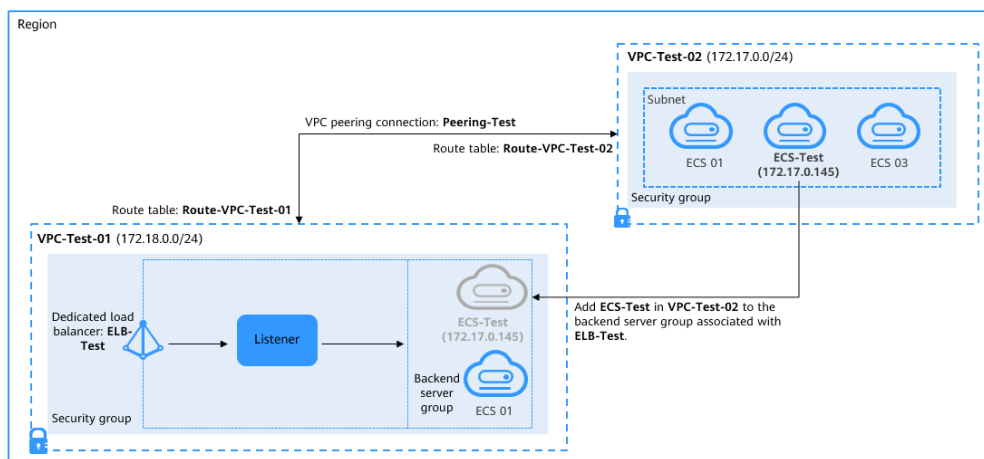
Scenarios

You can use ELB to route traffic to backend servers in a VPC that is different from where the load balancer works.

Solution

- Dedicated load balancer **ELB-Test** is running in a VPC named **VPC-Test-01** (172.18.0.0/24).
- An ECS named **ECS-Test** is running in **VPC-Test-02** (172.17.0.0/24).
- **IP as a Backend** is enabled for **ELB-Test**, and **ECS-Test** in **VPC-Test-02** (172.17.0.0/24) is added to the backend server group associated with **ELB-Test** in **VPC-Test-01**.

Figure 1-3 Topology



Advantages

You can enable **IP as a Backend** for a dedicated load balancer to route incoming traffic to servers in different VPCs from the load balancer.

Resource and Cost Planning

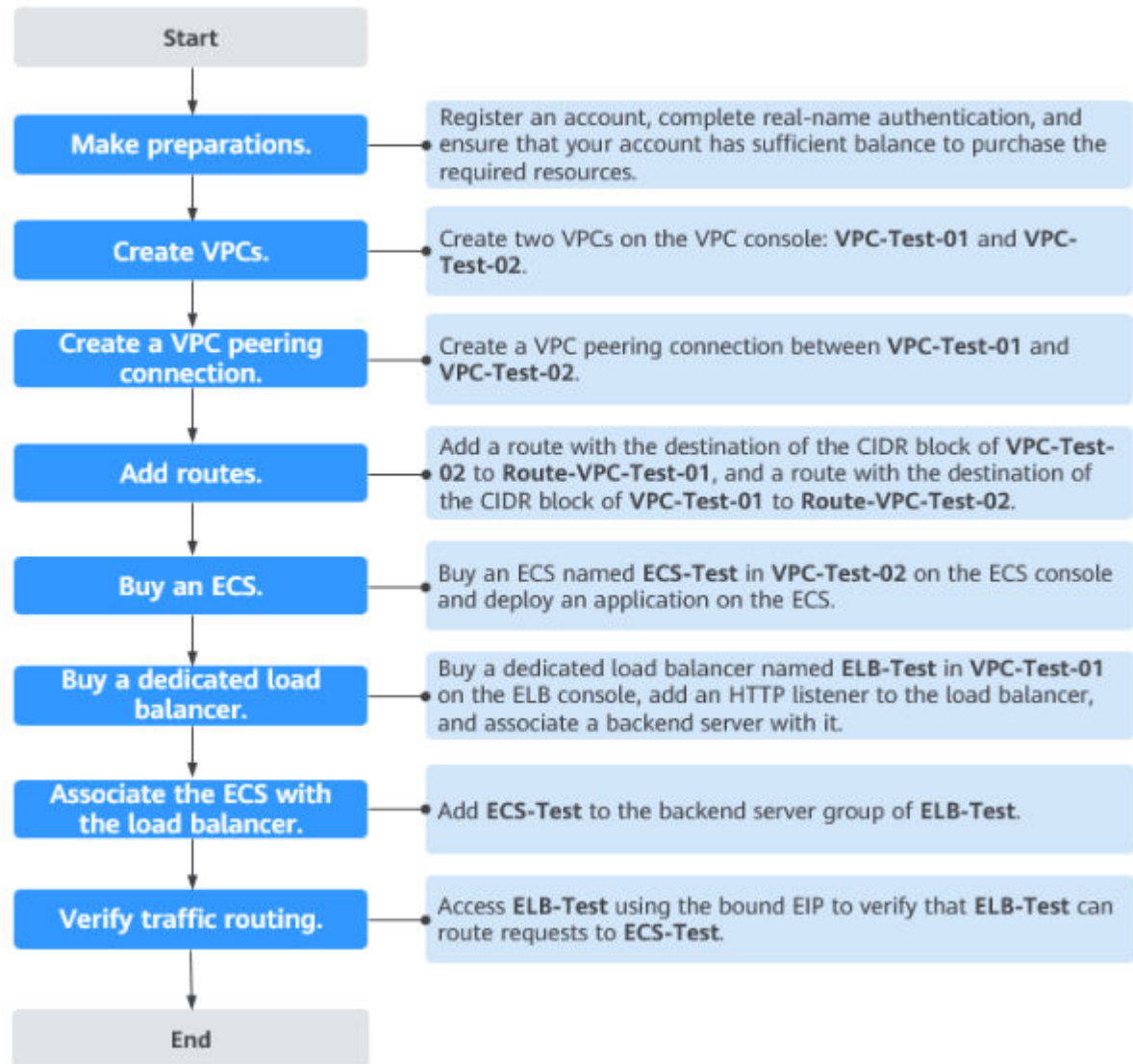
The actual cost shown on the Huawei Cloud console is used.

Table 1-1 Resource planning

Resource Type	Resource Name	Description	Quantity
VPC	VPC-Test-01	The VPC where ELB-Test is running: 172.18.0.0/24	1
	VPC-Test-02	The VPC where ECS-Test is running: 172.17.0.0/24	1
VPC peering connection	Peering-Test	The connection that connects the VPC where ELB-Test is running and the VPC where ECS-Test is running. Local VPC: 172.18.0.0/24 Peer VPC: 172.17.0.0/24	1
Route table	Route-VPC-Test-01	The route table of VPC-Test-01 . Destination: 172.17.0.0/24	1
	Route-VPC-Test-02	The route table of VPC-Test-02 . Destination: 172.18.0.0/24	1
ELB	ELB-Test	The dedicated load balancer to distribute incoming traffic.	1
EIP	EIP-Test	The EIP bound to ELB-Test : 119.3.233.52	1
ECS	ECS-Test	The ECS that is running in VPC-Test-02 . Private IP address: 172.17.0.145	1

Operation Process

Figure 1-4 Process of associating servers in a VPC that is different from the dedicated load balancer



Step 1: Create VPCs

1. Log in to the management console.
2. Choose **Networking > Virtual Private Cloud**. On the displayed page, click **Create VPC**.
3. Configure the parameters as described in [Table 1-1](#) and click **Create Now**. For details on how to create a VPC, see the [Virtual Private Cloud User Guide](#).
 - **Name:** **VPC-Test-01**
 - **IPv4 CIDR Block:** **172.18.0.0/24**
 - Configure other parameters as required.
4. Create the other VPC.

- **Name: VPC-Test-02**
- **IPv4 CIDR Block: 172.17.0.0/24**
- Configure other parameters as required.

Figure 1-5 Viewing the two VPCs

Name	IPv4 CIDR Block	Status	Subnets	Route Ta...	Servers	Enterprise Project	Operation
VPC-Test-01	172.18.0.0/24 (Primary CIDR)	Available	1	1	0	longterm-EPSTes...	Edit CIDR Block Delete
VPC-Test-02	172.17.0.0/24 (Primary CIDR)	Available	1	1	1	longterm-EPSTes...	Edit CIDR Block Delete

Step 2: Create a VPC Peering Connection

1. In the navigation pane on the left, click **VPC Peering Connections**.
2. In the upper right corner, click **Create VPC Peering Connection**.
3. Configure the parameters as follows and click **Create Now**. For details on how to create a VPC peering connection, see the [Virtual Private Cloud User Guide](#).
 - **Name: Peering-Test**
 - **Local VPC: VPC-Test-01**
 - **Peer VPC: VPC-Test-02**
 - Configure other parameters as required.

Step 3: Add Routes for Peering-Test

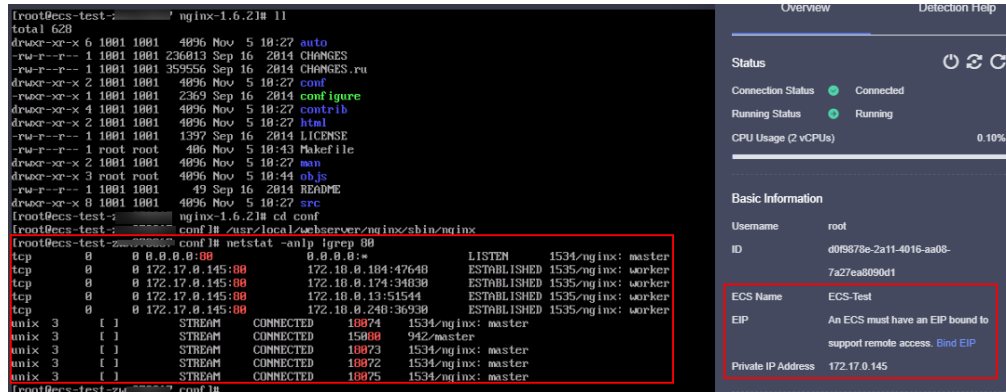
1. In the navigation pane on the left, click **Route Tables**.
2. In the upper right corner, click **Create Route Table**.
3. Configure the parameters as described in [Table 1-1](#) and click **OK**. For details on how to create a route table, see the [Virtual Private Cloud User Guide](#).
 - **Name: Route-VPC-Test-01**
 - **VPC: VPC-Test-01**
 - **Destination: 172.17.0.0/24**
 - **Next Hop Type: VPC peering connection**
 - **Next Hop: Peering-Test**
4. Repeat the preceding steps to create the other route table.
 - **Name: Route-VPC-Test-02**
 - **VPC: VPC-Test-02**
 - **Destination: 172.18.0.0/24**
 - **Next Hop Type: VPC peering connection**
 - **Next Hop: Peering-Test**

Step 4: Create an ECS

1. Under **Compute**, click **Elastic Cloud Server**.
2. In the upper right corner, click **Buy ECS**.

3. Select **VPC-Test-02** as the VPC and set **ECS Name** to **ECS-Test**. Configure other parameters as required. For details, see the [Elastic Cloud Server User Guide](#).
4. Deploy Nginx on the **ECS-Test**.

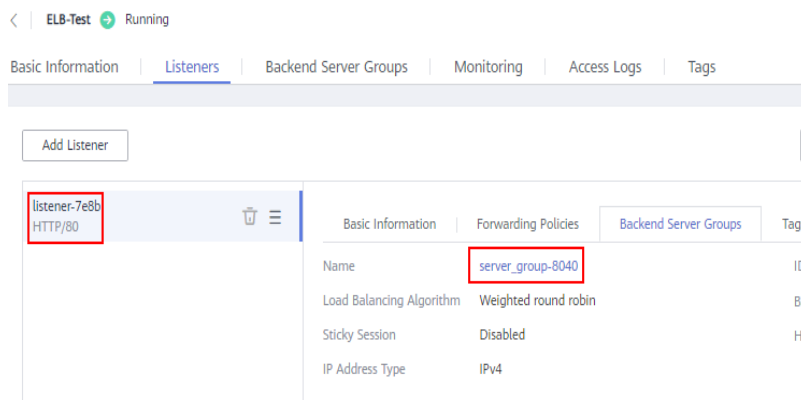
Figure 1-6 Deploying Nginx on ECS-Test



Step 5: Create a Dedicated Load Balancer with an HTTP Listener and Associate a Backend Server Group

1. On the management console, choose **Networking > Elastic Load Balance**.
2. In the upper right corner, click **Buy Elastic Load Balancer**.
3. Configure the parameters as follows. For details, see [Elastic Load Balance User Guide](#).
 - **Type:** Dedicated load balancer
 - **VPC:** VPC-Test-01
 - **Name:** ELB-Test
 - **IP as a Backend:** Enable it.
 - Configure other parameters as required.
4. Add an HTTP listener to **ELB-Test** and associate a backend server group with it.

Figure 1-7 Viewing the HTTP listener and backend server group



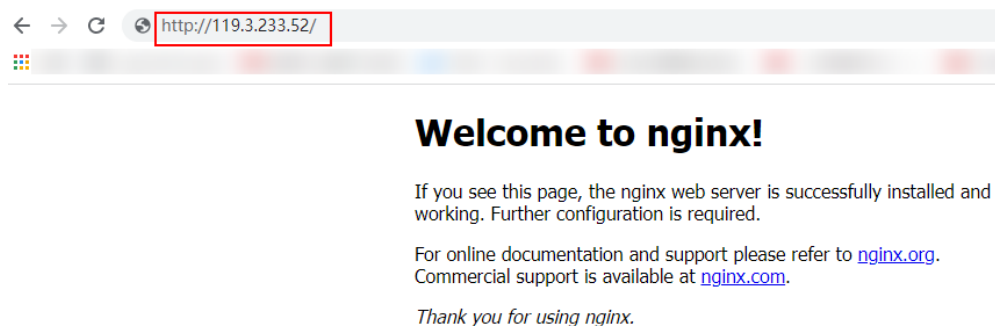
Step 6: Add ECS-Test to the Backend Server Group

1. Locate **ELB-Test** and click its name.
2. On the **Listeners** tab, locate the HTTP listener added to **ELB-Test** and click its name.
3. On the **Default Backend Server Group** area of the **Summary** tab, click **View/Add Backend Server** on the right.
4. The page for adding backend servers is displayed.
5. Click **IP as Backend Servers** on the lower part of the page. Click **Add** on the right, set parameters as required, and click **OK**. For details, see *Elastic Load Balance User Guide*.
 - **IP Address:** Set it to the private IP address of **ECS-Test** (172.17.0.145).
 - **Backend Port:** Set it as required.
 - **Weight:** Set it as required.
6. Click **OK**.

Step 7: Verify Traffic Routing

1. Locate **ELB-Test** and click **More** in the **Operation** column.
2. Select **Bind IPv4 EIP** to bind an EIP (**EIP-Test**: 119.3.233.52) to **ELB-Test**.
3. Enter **http://119.3.233.52/** in the address box of your browser to access **ELB-Test**. If the following page is displayed, **ELB-Test** routes the request to **ECS-Test**, which processes the request and returns the requested page.

Figure 1-8 Verifying that the request is routed to ECS-Test



1.3 Routing Traffic to Backend Servers in the Same VPC as the Load Balancer

Scenarios

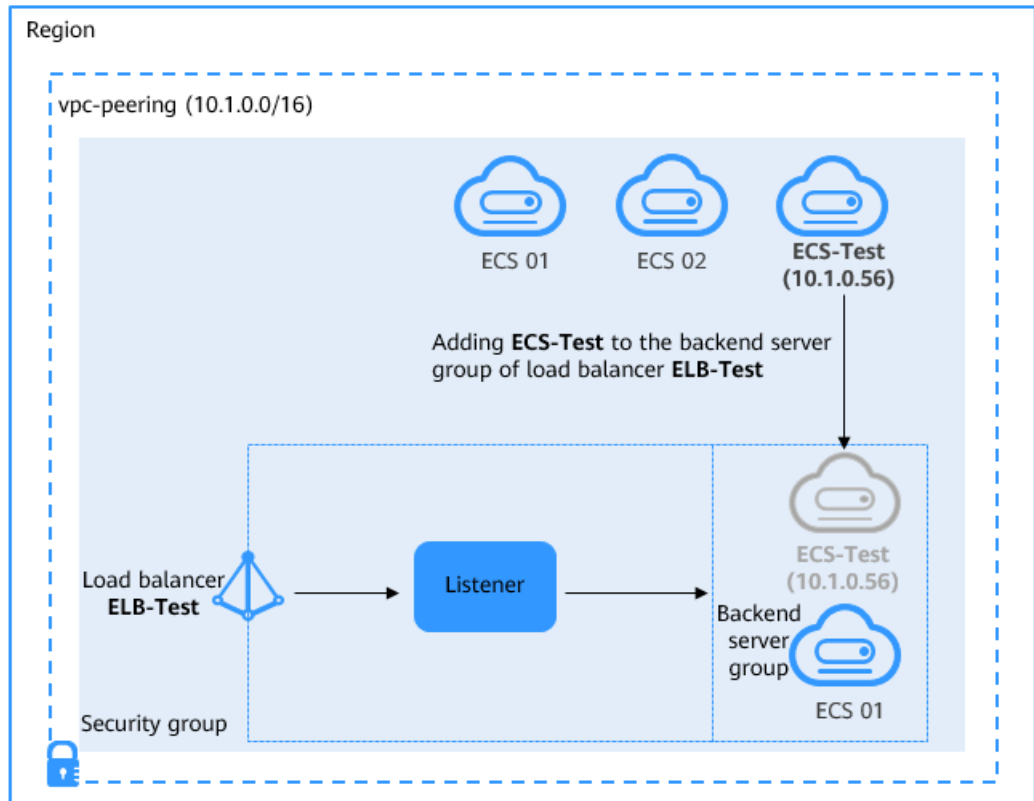
You can use ELB to route traffic to backend servers in the same VPC as the load balancer.

Solution

- Dedicated load balancer **ELB-Test** is running in a VPC named **vpc-peering** (10.1.0.0/16).

- An ECS named **ECS-Test** is also running in **vpc-peering** (10.1.0.0/16).
- **IP as a Backend** is enabled for **ELB-Test**, and **ECS-Test** in **vpc-peering** (10.1.0.0/16) is added to the backend server group associated with **ELB-Test** in **vpc-peering**.

Figure 1-9 Topology



Advantages

You can enable **IP as a Backend** for a dedicated load balancer to route traffic to backend servers in the same VPC as the load balancer.

Resource and Cost Planning

The actual cost shown on the Huawei Cloud console is used.

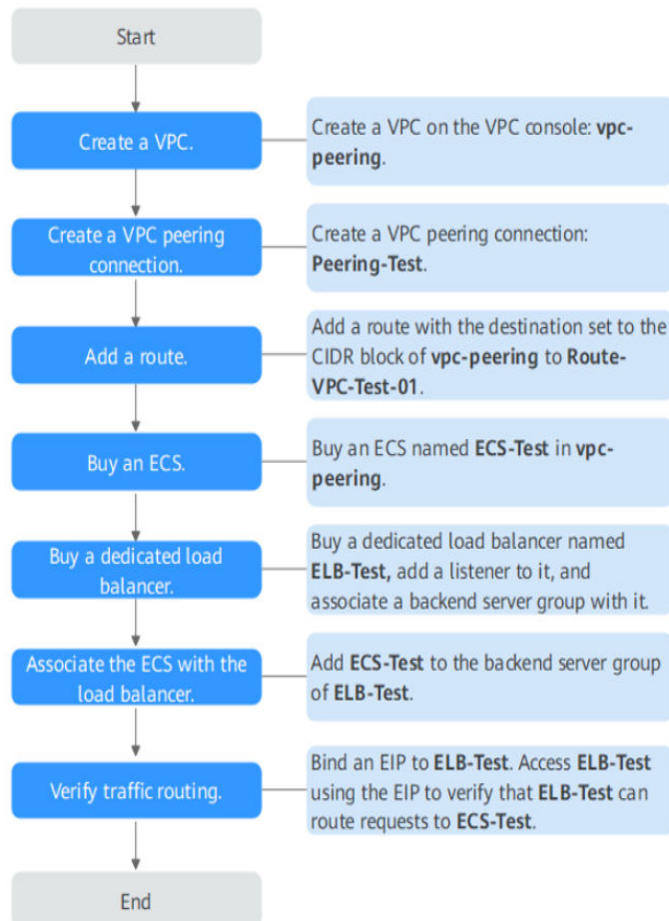
Table 1-2 Resource planning

Resource Type	Resource Name	Description	Quantity
VPC	vpc-peering	The VPC where ELB-Test and ECS-Test are running: 10.1.0.0/16	1

Resource Type	Resource Name	Description	Quantity
VPC peering connection	Peering-Test	The connection that connects the VPC where ELB-Test is running and other VPCs. Local VPC: 10.1.0.0/16 Peer VPC: any VPC	1
Route table	Route-VPC-Test-01	The route table of VPC-Test-01 . Destination: 10.1.0.0/16	1
ELB	ELB-Test	The dedicated load balancer to distribute incoming traffic. Private IP address: 10.1.0.9	1
EIP	EIP-Test	The EIP bound to ELB-Test : 120.46.131.153	1
ECS	ECS-Test	The ECS that is running in vpc-peering . Private IP address: 10.1.0.56	1

Operation Process

Figure 1-10 Process for adding backend servers in the same VPC as the load balancer



Step 1: Create a VPC

1. Log in to the management console.
2. Choose **Networking > Virtual Private Cloud**. On the displayed page, click **Create VPC**.
3. Configure the parameters as follows and click **Create Now**. For details on how to create a VPC, see the [Virtual Private Cloud User Guide](#).
 - **Name: vpc-peering**
 - **IPv4 CIDR Block: 10.1.0.0/16**
 - Configure other parameters as required.

Step 2: Create a VPC Peering Connection

1. In the navigation pane on the left, click **VPC Peering Connections**.
2. In the upper right corner, click **Create VPC Peering Connection**.
3. Configure the parameters as follows and click **Create Now**. For details on how to create a VPC peering connection, see the [Virtual Private Cloud User Guide](#).

- **Name: Peering-Test**
- **Local VPC: vpc-peering**
- **Peer VPC: any VPC**
- Configure other parameters as required.

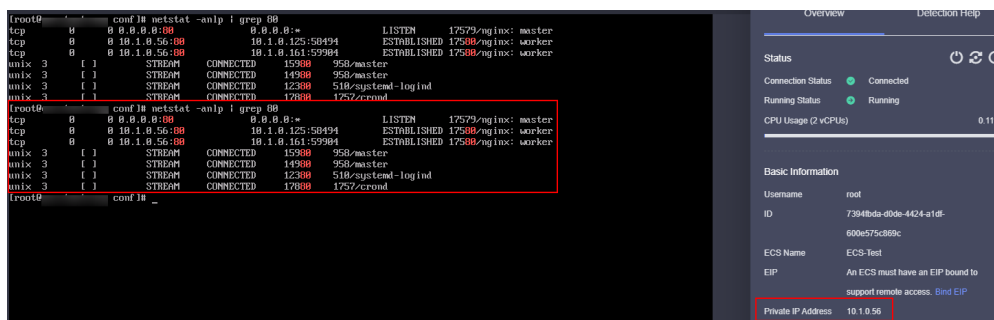
Step 3: Add Routes for Peering-Test

1. In the navigation pane on the left, click **Route Tables**.
2. In the upper right corner, click **Create Route Table**.
3. Configure the parameters as follows and click **OK**. For details on how to create a route table, see the [Virtual Private Cloud User Guide](#).
 - **Name: Route-VPC-Test-01**
 - **VPC: vpc-peering**
 - **Destination: 10.1.0.0/16**
 - **Next Hop Type: VPC peering connection**
 - **Next Hop: Peering-Test**

Step 4: Create an ECS

1. Under **Compute**, click **Elastic Cloud Server**.
2. In the upper right corner, click **Buy ECS**.
3. Configure the parameters as required. For details, see the [Elastic Cloud Server User Guide](#).
Select **vpc-peering** as the VPC and set **Name** to **ECS-Test**.
4. Deploy Nginx on the **ECS-Test**.

Figure 1-11 Deploying Nginx on ECS-Test



Step 5: Create a Dedicated Load Balancer with an HTTP Listener and Associate a Backend Server Group

1. On the management console, choose **Networking > Elastic Load Balance**.
2. In the upper right corner, click **Buy Elastic Load Balancer**.
3. Configure the parameters as follows. For details, see [Elastic Load Balance User Guide](#).
 - **Type: Dedicated**
 - **VPC: vpc-peering**

- **Name: ELB-Test**
 - **IP as a Backend:** Enable it.
 - Configure other parameters as required.
4. Add an HTTP listener to **ELB-Test** and associate a backend server group with it.

Step 6: Add ECS-Test to the Backend Server Group

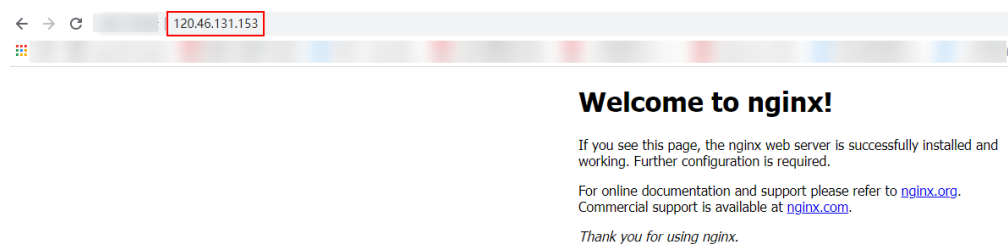
Locate **ELB-Test** and click its name.

1. On the **Default Backend Server Group** area of the **Summary** tab, click **View/Add Backend Server** on the right.
2. The page for adding backend servers is displayed.
3. Click **IP as Backend Servers** on the lower part of the page. Click **Add** on the right, set parameters as required, and click **OK**. For details, see *Elastic Load Balance User Guide*.
 - **IP Address:** Set it to the private IP address of **ECS-Test** (10.1.0.56).
 - **Backend Port:** Set it as required.
 - **Weight:** Set it as required.

Step 7: Verify Traffic Routing

1. Locate **ELB-Test** and click **More** in the **Operation** column.
2. Select **Bind IPv4 EIP** to bind an EIP (120.46.131.153) to **ELB-Test**.
3. Enter **http://120.46.131.153/** in the address box of your browser to access **ELB-Test**. If the following page is displayed, **ELB-Test** routes the request to **ECS-Test**, which processes the request and returns the requested page.

Figure 1-12 Verifying that the request is routed to ECS-Test



2 Using Advanced Forwarding for Application Iteration

Scenarios

As the business grows, you may need to upgrade your application based on user feedback. In this process, you can use advanced forwarding to redirect requests from users to both the new and old version first. When the application of the new version runs stably, direct all the requests to the new version.

Prerequisites

Six ECSs are available, with three having the application of the old version deployed and the other three having the new version deployed.

Process for Configuring Advanced Forwarding

Figure 2-1 Flowchart

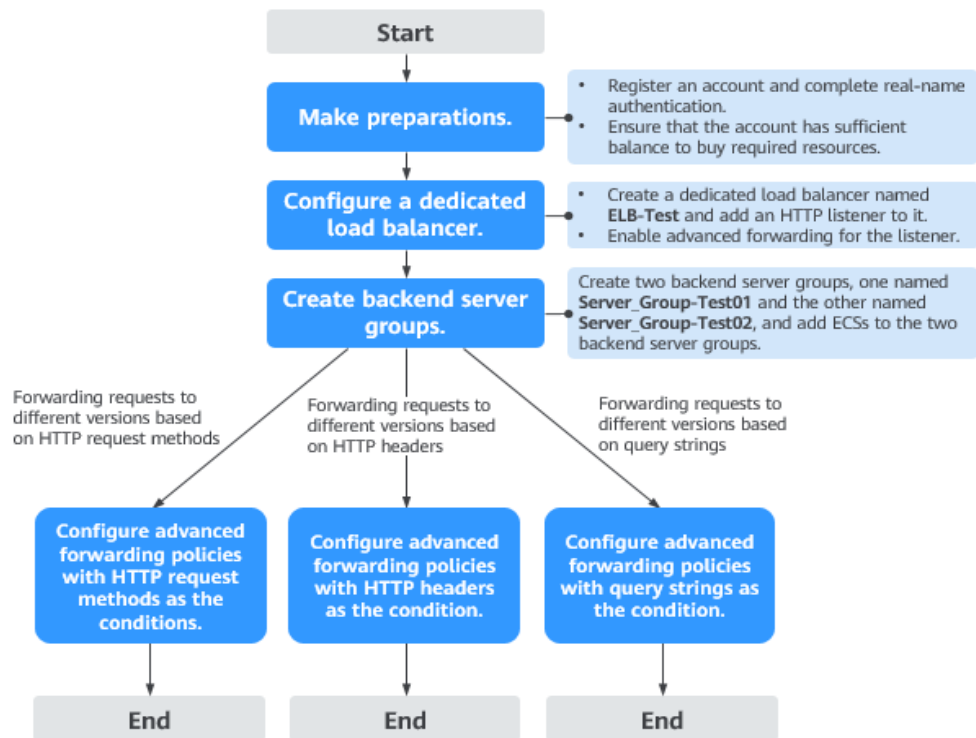


Table 2-1 Resource planning

Resource Name	Resource Type	Description
ELB-Test	Dedicated load balancer	Only dedicated load balancers support advanced forwarding.
Server_Group-Test01	Backend server group	Used to manage the ECSs where the application of the old version is deployed.
Server_Group-Test02	Backend server group	Used to manage the ECSs where the application of the new version is deployed.
ECS01	ECS	Used to deploy the application of the old version and added to Server_Group-Test01 .
ECS02	ECS	Used to deploy the application of the old version and added to Server_Group-Test01 .
ECS03	ECS	Used to deploy the application of the old version and added to Server_Group-Test01 .

Resource Name	Resource Type	Description
ECS04	ECS	Used to deploy the application of the new version and added to Server_Group-Test02 .
ECS05	ECS	Used to deploy the application of the new version and added to Server_Group-Test02 .
ECS06	ECS	Used to deploy the application of the new version and added to Server_Group-Test02 .

NOTE

In this practice, the dedicated load balancer is in the same VPC as the ECSs. You can also add servers in a different VPC or in an on-premises data center as needed. For details, see [Using IP as a Backend to Route Traffic Across Backend Servers](#).

Step 1: Configure a Dedicated Load Balancer



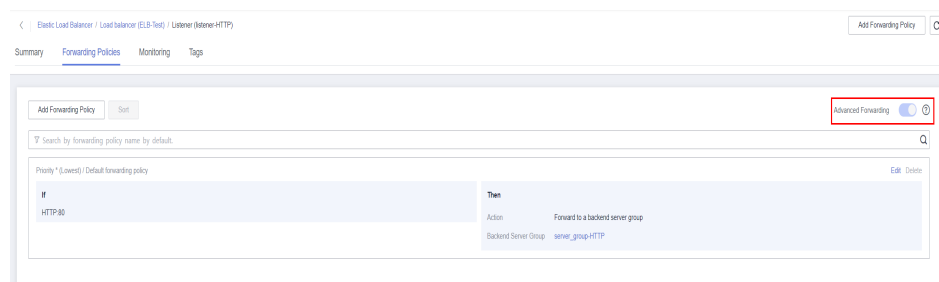


1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click  in the upper left corner to display **Service List** and choose **Networking > Elastic Load Balance**.
4. In the upper right corner, click **Buy Elastic Load Balancer**.
5. Create a dedicated load balancer and configure the parameters as follows.
 - **Type: Dedicated**
 - **Name: ELB-Test**
 - Set other parameters as required. For details, see [Creating a Dedicated Load Balancer](#).
6. Add an HTTP listener to **ELB-Test**. For details, see [Adding a Listener](#).
7. Enable advanced forwarding. For details, see [Advanced Forwarding Policy](#).

Figure 2-2 Enabling advanced forwarding



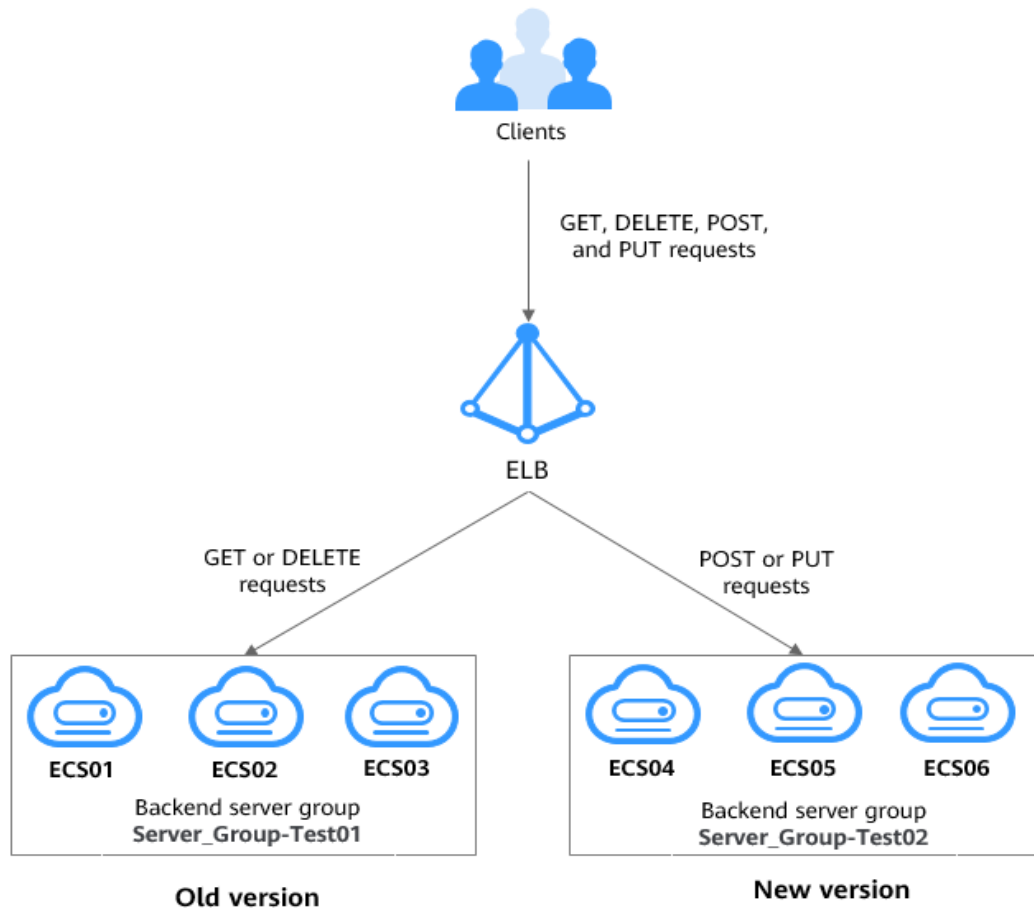
Step 2: Create Two Backend Server Groups and Add Backend Servers to Them

1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click  in the upper left corner to display **Service List** and choose **Networking > Elastic Load Balance**.
4. In the navigation pane on the left, choose **Elastic Load Balance > Backend Server Groups**.
5. Click **Create Backend Server Group** in the upper right corner.
 - Name: **Server_Group-Test01**
 - Load Balancer: Select **ELB-Test**.
 - **Backend Protocol: HTTP**
 - Configure other parameters as required.
6. Repeat [Step 5](#) to create backend server group **Server_Group-Test02**.
7. Add **ECS01**, **ECS02**, and **ECS03** to backend server group **Server_Group-Test01**.
8. Add **ECS04**, **ECS05**, and **ECS06** to backend server group **Server_Group-Test02**.

Forwarding Requests to Different Versions of the Application based on HTTP Request Methods

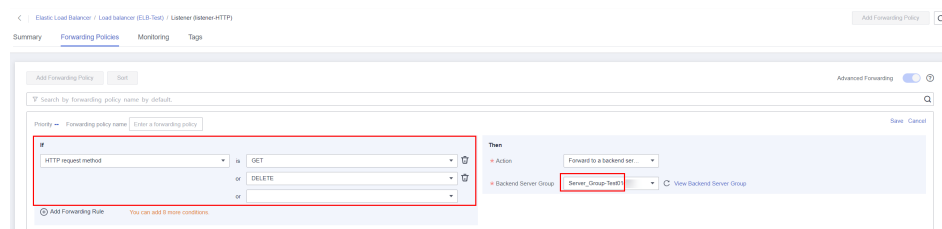
Configure two advanced forwarding policies with the HTTP request method as the condition to route GET and DELETE requests to the application of the old version and POST and PUT requests to the application of the new version. When the application of the new version runs stably, direct all the requests to the new version.

Figure 2-3 Forwarding requests based on HTTP request methods



1. Locate the dedicated load balancer and click its name **ELB-Test**.
2. On the **Listeners** tab, locate the HTTP listener added to the dedicated load balancer and click its name.
3. Switch to the **Forwarding Policies** tab on the right, and click **Add Forwarding Policy** to forward requests to application of the old version. Select **GET** and **DELETE** from the **HTTP request method** drop-down list, select **Forward to a backend server group** for **Action**, and select **Server_Group-Test01** from the drop-down list.

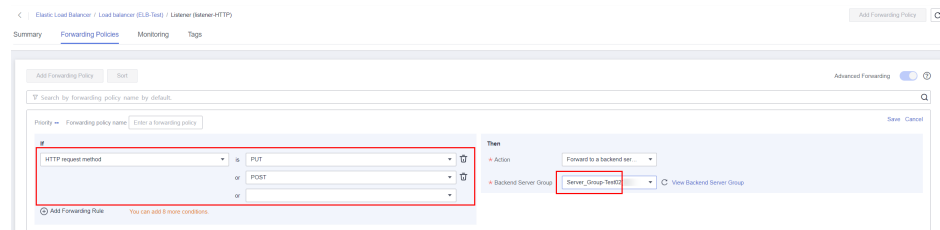
Figure 2-4 Forwarding GET and DELETE requests to the application of the old version



4. Click **Save**.
5. Repeat the preceding steps to add a forwarding policy to forward PUT and POST requests to the application of the new version.

Select **PUT** and **POST** from the **HTTP request method** drop-down list, select **Forward to a backend server group** for **Action**, and select **Server_Group-Test02** from the drop-down list.

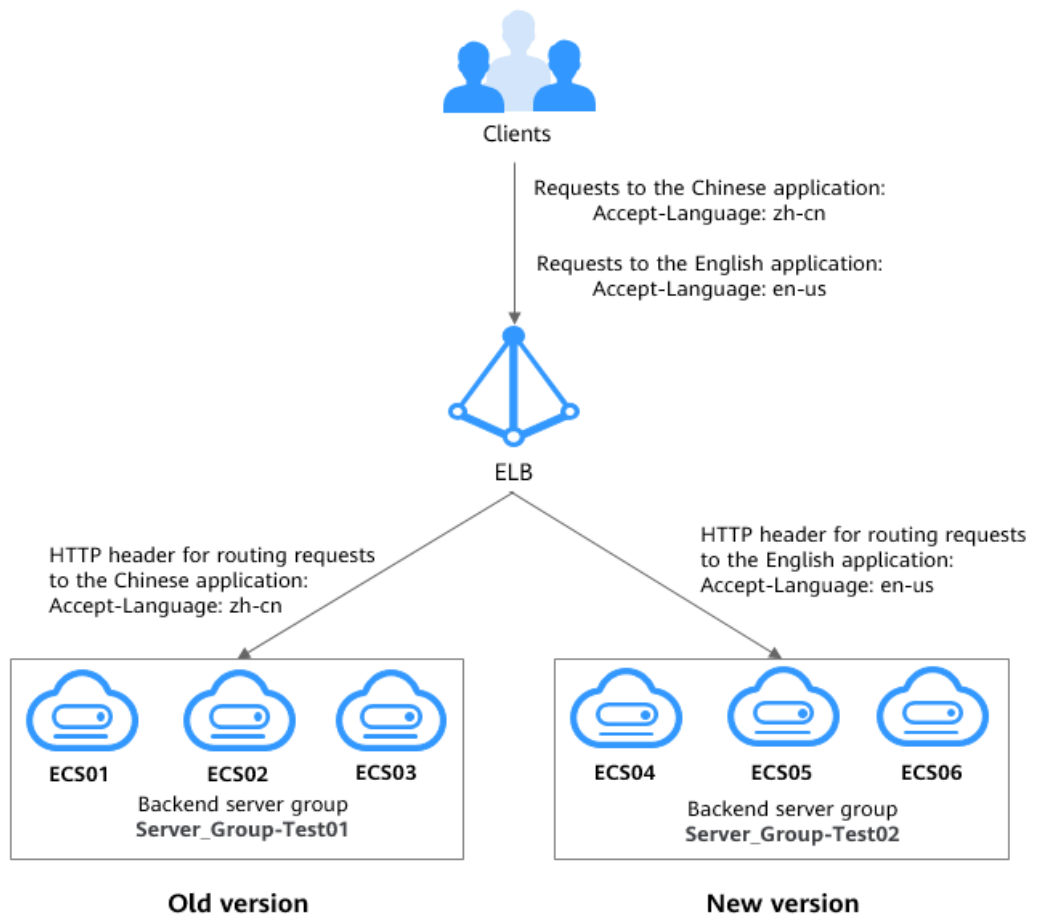
Figure 2-5 Forwarding PUT and POST requests to the application of the new version



Forwarding Requests to Different Versions of the Application based on HTTP Headers

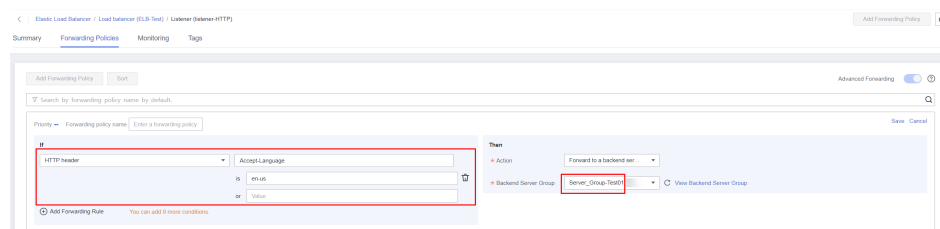
If the old version supports both Chinese and English, but the new version only supports English because the Chinese version is still under development, you can configure two advanced forwarding policies with the HTTP header as the condition to route requests to the Chinese application to the old version and requests to the English application to the new version. When the application of the new version supports the Chinese, direct all the requests to the new version.

Figure 2-6 Smooth application transition between the old and new versions based on the HTTP request header



1. Locate the dedicated load balancer and click its name **ELB-Test**.
2. On the **Listeners** tab, locate the HTTP listener added to the dedicated load balancer and click its name.
3. Switch to the **Forwarding Policies** tab on the right, and click **Add Forwarding Policy** to forward requests to application of the old version. Select **HTTP header** from the drop-down list, set the key to **Accept-Language** and value to **en-us**, set the action to **Forward to a backend server group**, and select **Server_Group-Test01** as the backend server group.

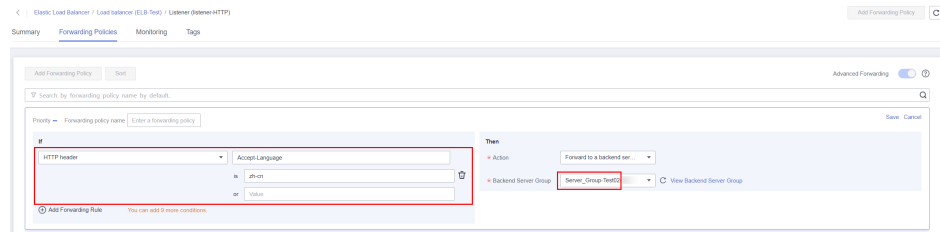
Figure 2-7 Forwarding requests to the application of the old version



4. Click **Save**.
5. Repeat the preceding steps to add a forwarding policy to forward requests to the application of the new version.

Select **HTTP header** from the drop-down list, set the key to **Accept-Language** and value to **zh-cn**, set the action to **Forward to a backend server group**, and select **Server_Group-Test02** as the backend server group.

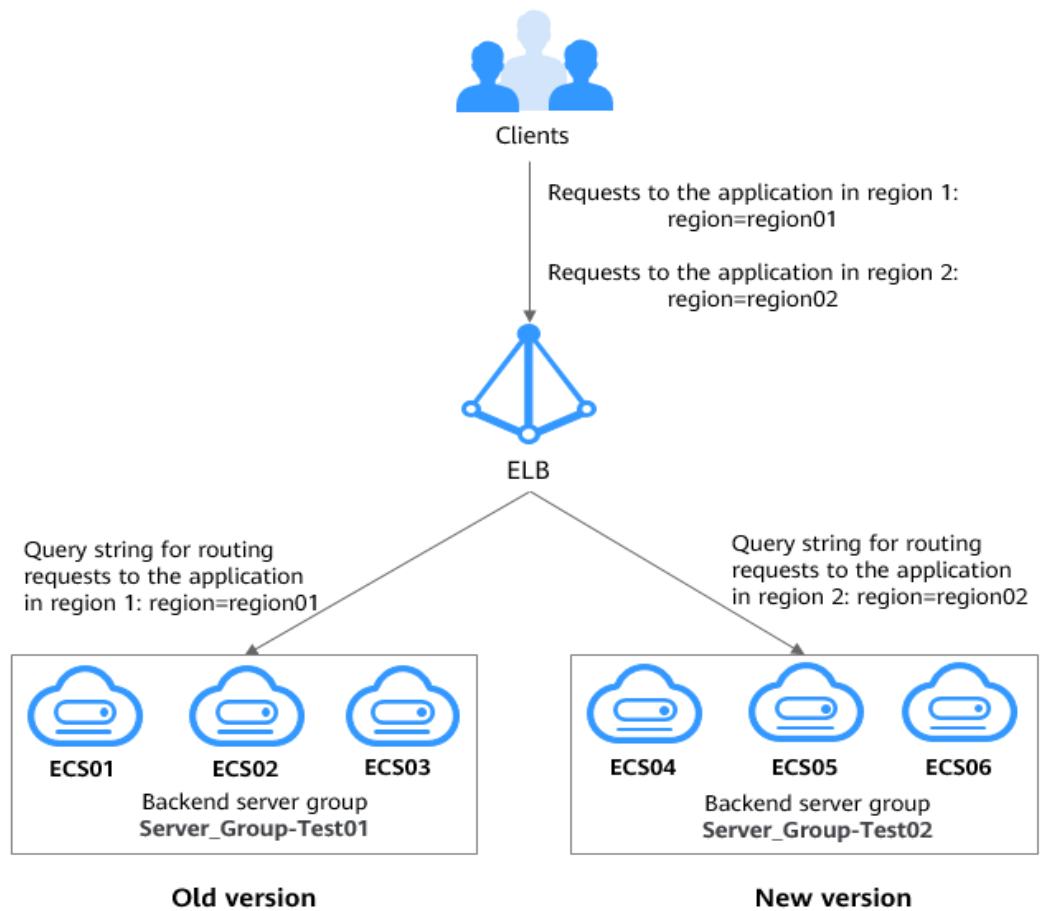
Figure 2-8 Forwarding requests to the application of the new version



Forwarding Requests to Different Versions of the Application based on Query Strings

If the application is deployed across regions, you can configure two advanced forwarding policies with query string as the condition to forward requests to the application in region 1 to the old version and requests to the application in region 2 to the new version. When the application of the new version runs stably, direct all the requests to the new version.

Figure 2-9 Forwarding requests based on query strings

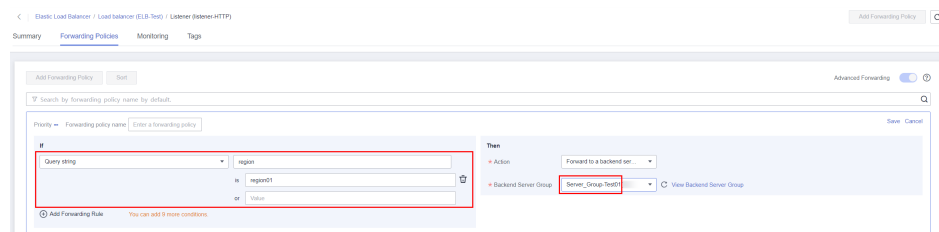


NOTE

- Dedicated load balancers can distribute traffic across regions or VPCs.
- In this example, you need to use Cloud Connect to connect the VPCs in two regions and then use a dedicated load balancer to route traffic to backend servers in the two regions.

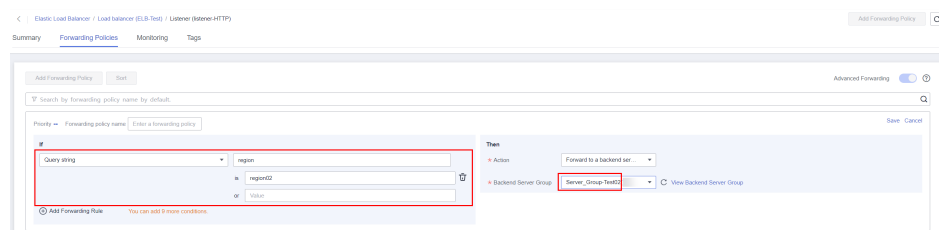
1. Locate **ELB-Test** and click its name.
2. On the **Listeners** tab, locate the HTTP listener added to the dedicated load balancer and click its name.
3. Switch to the **Forwarding Policies** tab on the right, and click **Add Forwarding Policy** to forward requests to application of the old version.

Select **Query string** from the drop-down list, set the key to **region** and value to **region01**, set **Action** to **Forward to a backend server group**, and select **Server_Group-Test01** as the backend server group.

Figure 2-10 Forwarding requests to the old version

4. Click **Save**.
5. Repeat the preceding steps to add a forwarding policy to forward requests to the application of the new version.

Select **Query string** from the drop-down list, set the key to **region** and value to **region02**, set **Action** to **Forward to a backend server group**, and select **Server_Group-Test02** as the backend server group.

Figure 2-11 Forwarding requests to the new version

3 Integrating WAF with ELB to Protect Your Websites

Scenarios

If your service servers are deployed on Huawei Cloud, you can purchase dedicated WAF instances to protect important domain names or websites that only use IP addresses to provide services.

In this way, ELB routes HTTP or HTTPS requests first to dedicated WAF instances for filtering out malicious traffic and the latter then directs normal traffic to backend servers.

This document describes how you can add dedicated WAF instances to the backend server group of your load balancer to protect your websites.

Notes and Constraints

- The security group rules configured for backend servers must allow traffic from the backend subnet where the load balancer resides to the backend servers over the backend port. For details, see [Configuring Security Group Rules for Backend Servers](#).
- The security group rules configured for WAF instances must allow traffic over the specified port. For details, see [Adding a Security Group Rule](#).

Traffic Path

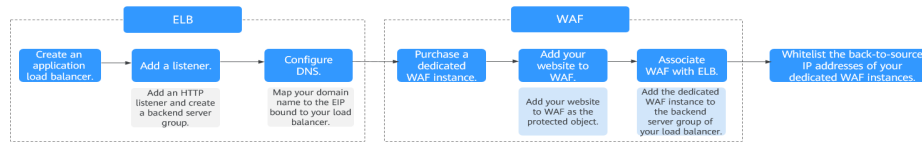
After WAF is integrated with ELB, the traffic flow is as illustrated in [Figure 3-1](#).

Figure 3-1 Traffic path



Procedure

Figure 3-2 Process for associating a dedicated WAF instance with an application load balancer

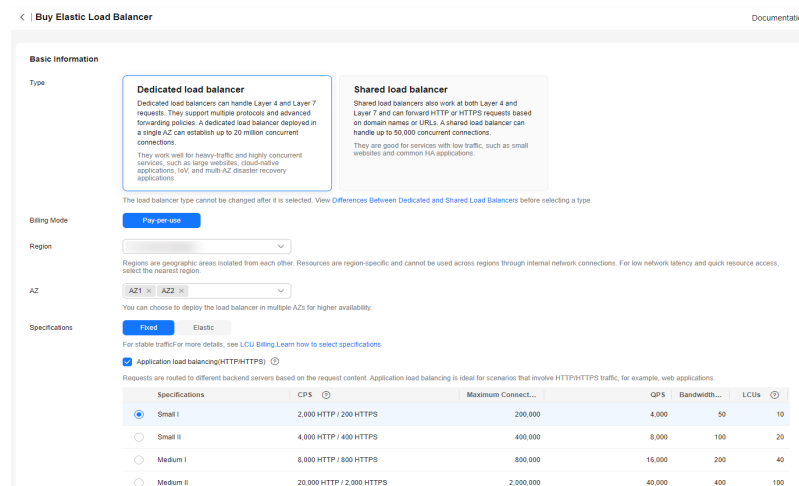


Creating an Application Load Balancer

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner to display **Service List** and choose **Networking > Elastic Load Balance**.
4. On the **Load Balancers** page, click **Buy Elastic Load Balancer**. For details, see [Creating a Dedicated Load Balancer](#).

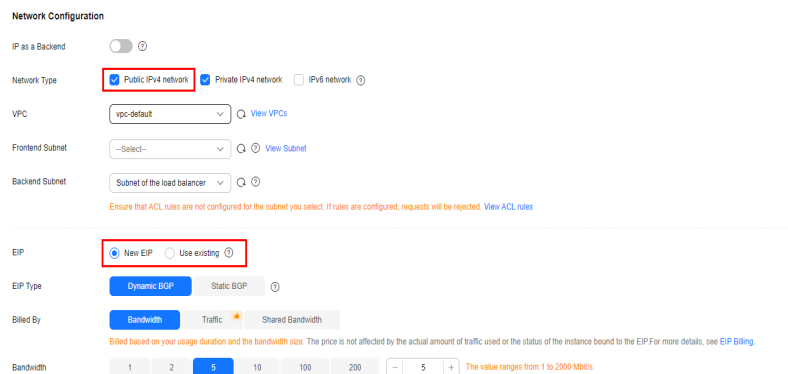
Complete the basic configuration of the load balancer as prompted. For example, select **Application load balancing (HTTP/HTTPS)** for **Specifications**.

Figure 3-3 Creating an application load balancer (Dedicated)





5. Configure the network as prompted.
Choose **Public IPv4 network** for **Network Type** and select an existing EIP for or assign a new EIP to the load balancer to receive requests from public networks.

Figure 3-4 Selecting an EIP for the load balancer



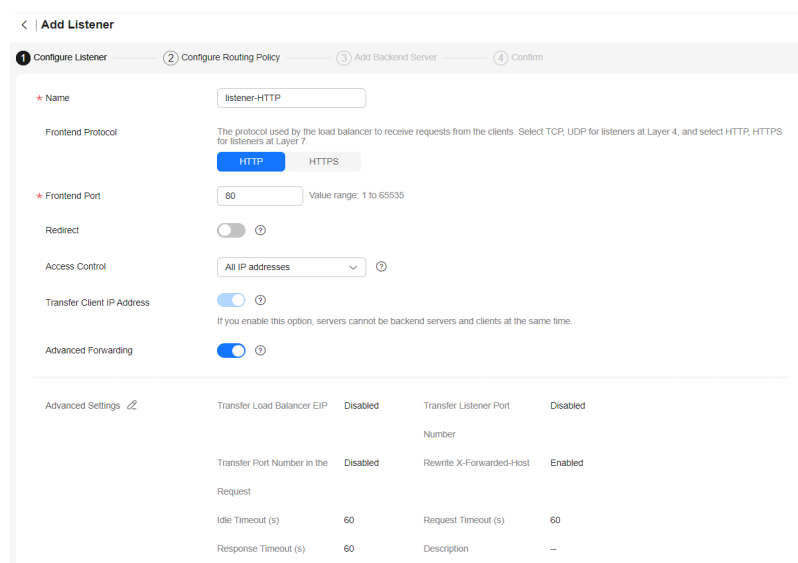
6. Confirm the information, click **Next**, and submit your request.

Adding an HTTP Listener and Creating a Backend Server Group

1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click  in the upper left corner to display **Service List** and choose **Networking > Elastic Load Balance**.
4. On the **Load Balancers** page, locate the created load balancer and click its name.
5. Under **Listeners**, click **Add Listener**. Add an HTTP listener and specify a frontend port for it.

For details, see .

Figure 3-5 Adding an HTTP listener



6. Click **Next: Configure Request Routing Policy**. On the **Configure Routing Policy** page, select **Create new** for **Backend Server Group**.

Figure 3-6 Creating a backend server group

7. Click **Next: Add Backend Server**. Add backend servers and configure health check for the backend server group.
8. Click **Next: Confirm**, confirm the settings, and click **Submit**.

Translating Domain Names into the EIP of the Load Balancer

Use Huawei Cloud DNS to translate your domain name, such as `www.example.com`, into the EIP bound to your load balancer.

For details about how to configure DNS, see [Routing Internet Traffic to a Website](#).

Buying a Dedicated WAF Instance



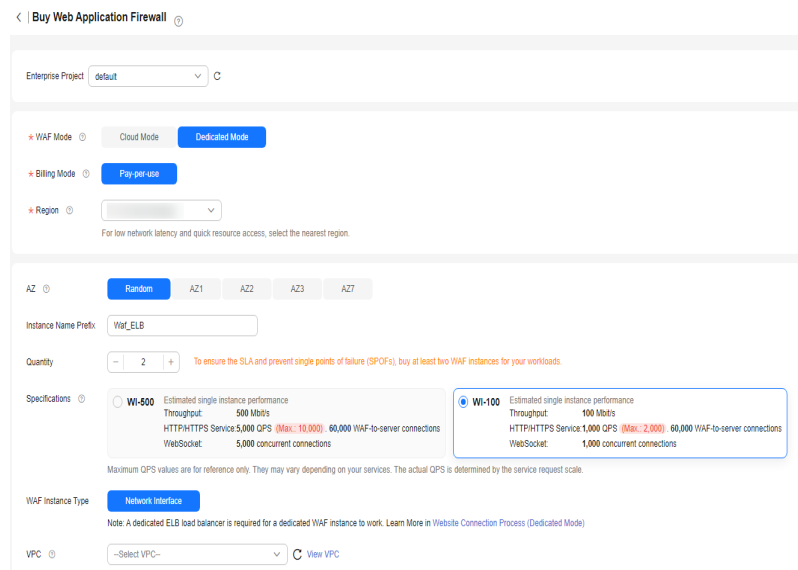
1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click  in the upper left corner and choose **Web Application Firewall** under **Security & Compliance**.
4. In the upper right corner of the page, click **Buy WAF**. Configure the parameters as prompted. Select **Dedicated Mode** for **WAF Mode**.

Figure 3-7 Buying a dedicated WAF instance**NOTE**

Dedicated WAF instances cannot be purchased. Only existing dedicated WAF instances can be used.

5. Confirm the settings and go through the subsequent steps to complete the purchase.

Adding Your Website to WAF

You can add your website (domain name: **www.example.com**) to WAF by following the below steps. For more details, see [Adding a Website to WAF \(Dedicated Mode\)](#).



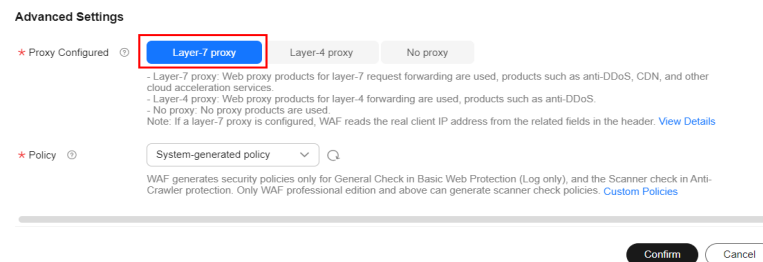
1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click  in the upper left corner and choose **Web Application Firewall** under **Security & Compliance**.
4. In the navigation pane on the left, choose **Website Settings**.
5. In the upper left corner of the website list, click **Add Website**.
In the displayed dialog box, select **Dedicated mode** and click **OK**.

Figure 3-8 Adding a domain name to WAF



6. Confirm the advanced settings. Set **Proxy Configured to Layer-7 proxy**.

Figure 3-9 Confirming advanced settings



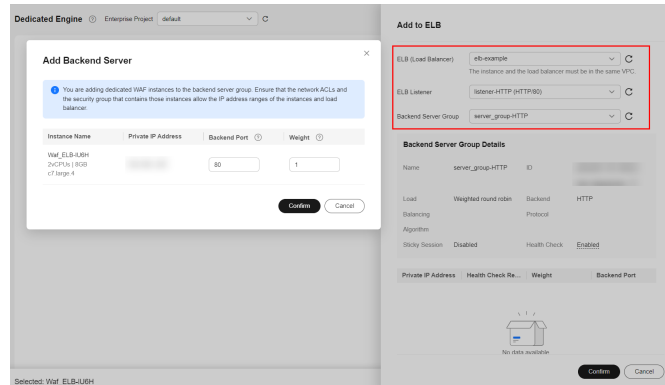
Associating WAF with Your Load Balancer

You can add THE dedicated WAF instance to the backend server group. Ensure that the network ACL and the security group that contains the WAF instance allow traffic from IP address ranges where the WAF instance and load balancer are deployed.

1. Log in to the management console.
2. In the upper left corner of the page, click and select the desired region and project.
3. Click in the upper left corner and choose **Web Application Firewall** under **Security & Compliance**.
4. In the navigation pane on the left, choose **Instance Management > Dedicated Engine** to go to the dedicated WAF instance page.
5. Locate the WAF instance created in [Buying a Dedicated WAF Instance](#) and choose **More > Add to ELB** in the **Operation** column.

6. In the **Add to ELB** dialog box, specify **ELB (Load Balancer)**, **ELB Listener**, and **Backend Server Group**.

Figure 3-10 Adding the WAF instance to a load balancer



7. Click **Confirm**. Set **Backend Port** to the one configured for **Protected Port** in [Adding Your Website to WAF](#).
8. Click **Confirm**.

Whitelisting the Back-to-Source IP Addresses of Your Dedicated WAF Instance

In dedicated mode, website traffic is directed to the load balancer and then to dedicated WAF instances. The latter filters out malicious traffic and routes only normal traffic to the origin server.

In this way, the origin server only communicates with WAF back-to-source IP addresses. By doing so, WAF protects the origin server from being attacked. In dedicated mode, the WAF back-to-source IP addresses are the subnet IP addresses of the dedicated WAF instances.

The security software on the origin server may most likely regard WAF back-to-source IP addresses as malicious and block them. Once they are blocked, the origin server will deny all WAF requests. As a result, your website may become unavailable or respond very slowly. Therefore, ACL rules must be configured on the origin server to trust only the subnet IP addresses of your dedicated WAF instances.

For details, see [Pointing Traffic to a Load Balancer](#).

4 Configuring HTTPS Mutual Authentication to Improve Service Security

Scenarios

In common HTTPS service scenarios, only the server certificate is required for authentication. For some mission-critical services, you need to deploy both the server certificate and the client certificate for mutual authentication.

Self-signed certificates are used as an example to describe how to configure mutual authentication. Self-signed certificates do not provide all the security properties provided by certificates signed by a CA. It is recommended that you purchase certificates from [SSL Certificate Manager \(SCM\)](#) or CAs.

Procedure

Figure 4-1 Procedure for configuring mutual authentication



Step 1: Add a CA Certificate Using OpenSSL

1. Log in to a Linux server with OpenSSL installed.
2. Create the **server** directory and switch to the directory:
mkdir ca
cd ca
3. Create the certificate configuration file **ca_cert.conf**. The file content is as follows:

```
[ req ]
distinguished_name = req_distinguished_name
prompt = no

[ req_distinguished_name ]
O = ELB
```
4. Create the CA certificate private key **ca.key**.
openssl genrsa -out ca.key 2048

Figure 4-2 Private key of the CA certificate

```
[root@elbv30003 ca]# openssl genrsa -out ca.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
[root@elbv30003 ca]#
```

5. Create the certificate signing request (CSR) file **ca.csr** for the CA certificate.
openssl req -out ca.csr -key ca.key -new -config ./ca_cert.conf
6. Create the self-signed CA certificate **ca.crt**.
openssl x509 -req -in ca.csr -out ca.crt -sha1 -days 5000 -signkey ca.key

Figure 4-3 Creating a self-signed CA certificate

```
[root@elbv30003 ca]# openssl x509 -req -in ca.csr -out ca.crt -sha1 -days 5000 -signkey ca.key
Signature ok
subject=O = ELB
Getting Private key
[root@elbv30003 ca]#
```

Step 2: Issue a Server Certificate Using the CA Certificate

The server certificate can be a CA signed certificate or a self-signed one. In the following steps, a self-signed certificate is used as an example to describe how to create a server certificate.

1. Log in to the server where the CA certificate is generated.
2. Create a directory at the same level as the directory of the CA certificate and switch to the directory.

```
mkdir server
```

```
cd server
```

3. Create the certificate configuration file **server_cert.conf**. The file content is as follows:

```
[ req ]
distinguished_name = req_distinguished_name
prompt = no

[ req_distinguished_name ]
O = ELB
CN = www.test.com
```

NOTE

Set the **CN** field to the domain name or IP address of the Linux server.

4. Create the server certificate private key **server.key**.
openssl genrsa -out server.key 2048
5. Create the CSR file **server.csr** for the server certificate.
openssl req -out server.csr -key server.key -new -config ./server_cert.conf
6. Use the CA certificate to issue the server certificate **server.crt**.
openssl x509 -req -in server.csr -out server.crt -sha1 -CAcreateserial -days 5000 -CA ../ca/ca.crt -CAkey ../ca/ca.key

Figure 4-4 Issuing a server certificate

```
[root@elbv30003 server]# openssl x509 -req -in server.csr -out server.crt -sha1 -CAcreateserial -days 5000 -CA ../ca/ca.crt -CAkey ../ca/ca.key
Signature ok
subject=O = ELB, CN = www.test.com
Getting CA Private Key
[root@elbv30003 server]#
```

Step 3: Issue a Client Certificate Using the CA Certificate

1. Log in to the server where the CA certificate is generated.
2. Create a directory at the same level as the directory of the CA certificate and switch to the directory.

```
mkdir client
```

```
cd client
```

3. Create the certificate configuration file **client_cert.conf**. The file content is as follows:

```
[ req ]
distinguished_name = req_distinguished_name
prompt = no

[ req_distinguished_name ]
O = ELB
CN = www.test.com
```

NOTE

Set the **CN** field to the domain name or IP address of the Linux server.

4. Create the client certificate private key **client.key**.

```
openssl genrsa -out client.key 2048
```

Figure 4-5 Creating a client certificate private key

```
[root@elbv30003 client]# openssl genrsa -out client.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
[root@elbv30003 client]#
```

5. Create the CSR file **client.csr** for the client certificate.

```
openssl req -out client.csr -key client.key -new -config ./client_cert.conf
```

Figure 4-6 Creating a client certificate CSR file

```
[root@elbv30003 client]# openssl req -out client.csr -key client.key -new -config ./client_cert.conf
```

6. Use the CA certificate to issue the client certificate **client.crt**.

```
openssl x509 -req -in client.csr -out client.crt -sha1 -CAcreateserial -days 5000 -CA ../ca/ca.crt -CAkey ../ca/ca.key
```

Figure 4-7 Issuing a client certificate

```
[root@elbv30003 client]# openssl x509 -req -in client.csr -out client.crt -sha1 -CAcreateserial -days 5000 -CA ../ca/ca.crt -CAkey ../ca/ca.key
Signature ok
subject=O = ELB, CN = www.test.com
Getting CA Private Key
[root@elbv30003 client]#
```

7. Convert the client certificate to a **.p12** file that can be identified by the browser.

```
openssl pkcs12 -export -clcerts -in client.crt -inkey client.key -out client.p12
```

 **NOTE**

A password is required during command execution. Save this password, which will be required when you import the certificate using the browser.

Step 4: Upload the Server Certificate to ELB

1. Log in to the load balancer management console.
2. In the navigation pane on the left, choose **Certificates**.
3. In the navigation pane on the left, choose **Certificates**. On the displayed page, click **Add Certificate**. In the **Add Certificate** dialog box, select **Server certificate**, copy the content of server certificate **server.crt** to the **Certificate Content** area and the content of private key file **server.key** to the **Private Key** area, and click **OK**.

 **NOTE**

Delete the last newline character before you copy the content.

 **NOTE**

The certificate and private key must be PEM-encoded.

Step 5: Upload the CA Certificate to ELB

Step 1 Log in to the load balancer management console.

Step 2 In the navigation pane on the left, choose **Certificates**.

Step 3 Click **Add Certificate**. In the **Add Certificate** dialog box, select **CA certificate**, copy the content of CA certificate **ca.crt** created in [Step 1: Add a CA Certificate Using OpenSSL](#) to the **Certificate Content** area, and click **OK**.

 **NOTE**

Delete the last newline character before you copy the content.

Figure 4-8 Adding a CA certificate

Add Certificate X

Certificate Type: Server certificate CA certificate ?

* Certificate Name:

* Enterprise Project: 🔍 ? [Create Enterprise Project](#)

* Certificate Content:

The certificate content starts with -----BEGIN CERTIFICATE----- and ends with -----END CERTIFICATE-----. Each line contains 64 characters, the last line contains a maximum of 64 characters, and blank lines are not allowed. ?

[View Example](#) 📄

Description: 0/255 ↵

NOTE

The certificate must be PEM-encoded.

-----End

Step 6: Configure HTTPS Mutual Authentication

1. Log in to the load balancer management console.
2. Locate the target load balancer and click its name. Under **Listeners**, click **Add Listener**. Select **HTTPS** for **Frontend Protocol** and **Mutual authentication** for **SSL Authentication**, and select the CA certificate and server certificate you have added.

Figure 4-9 Configuring mutual authentication

< | Add Listener

1 Configure Listener — 2 Configure Routing Policy — 3 Add Backend Server — 4 Confirm

* Name

Frontend Protocol The protocol used by the load balancer to receive requests from the clients. Select TCP, UDP for listeners at Layer 4, and select HTTP, HTTPS for listeners at Layer 7.
TCP UDP HTTP **HTTPS**

* Frontend Port Value range: 1 to 65535

SSL Authentication SSL enables the clients and backend servers to mutually authenticate each other.
One-way authentication **Mutual authentication**
The clients and the load balancer will authenticate each other. Only authenticated clients will be allowed to access the load balancer.

* CA Certificate [View Certificate](#)

* Server Certificate [View Certificate](#)

Enable SNI

Access Control [?](#)

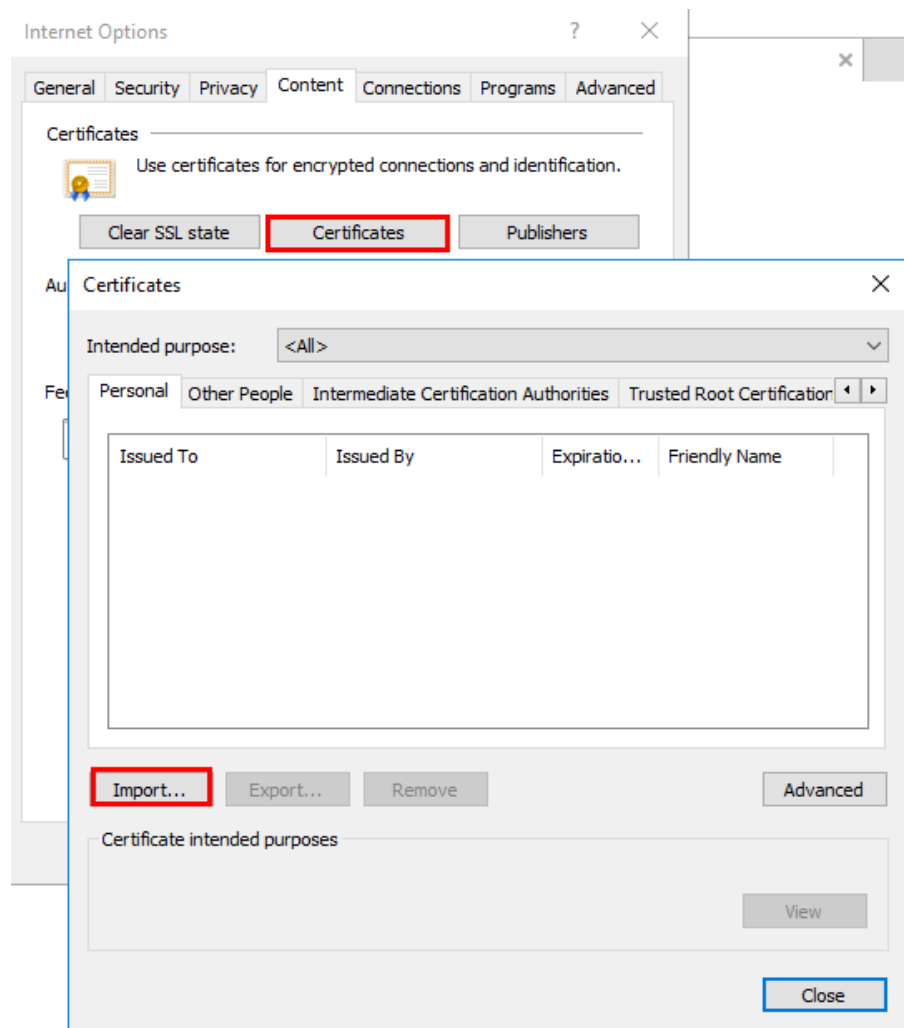
Transfer Client IP Address [?](#)

Step 7: Import the Client Certificate and Verify Mutual Authentication

Method 1: Using a browser

1. Import the client certificate using a browser (Internet Explorer 11 is used as an example).
 - a. Export **client.p12** from the Linux server.
 - b. Open the browser, choose **Settings > Internet Options** and click **Content**.
 - c. Click **Certificates** and then **Import** to import the **client.p12** certificate.

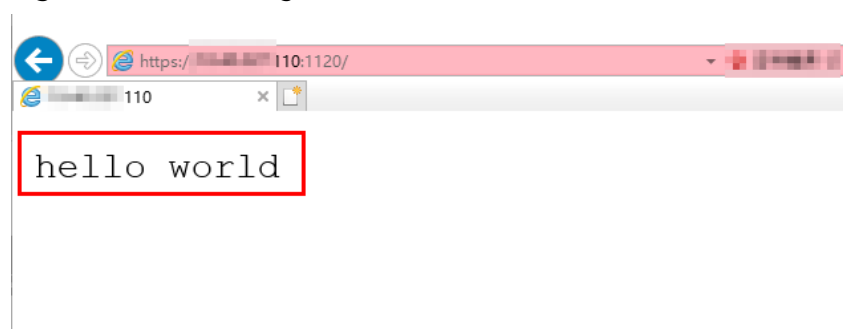
Figure 4-10 Importing the **client.p12** certificate



2. Verify the import.

Enter the access address in the address box of your browser. A window is displayed asking you to select the certificate. Select the client certificate and click **OK**. If the website can be accessed, the certificate is successfully imported.

Figure 4-11 Accessing the website



Method 2: Using cURL

1. Import the client certificate.
Copy client certificate **client.crt** and private key **client.key** to a new directory, for example, **/home/client_cert**.

2. Verify the import.

On the Shell screen, run the following command:

```
curl -k --cert /home/client_cert/client.crt --key /home/client_cert/client.key https://  
XXX.XXX.XXX.XXX:XXX/ -I
```

Ensure that the certificate address, private key address, IP address and listening port of the load balancer are correct. Replace **https://XXX.XXX.XXX.XXX:XXX** with the actual IP address and port number. If the expected response code is returned, the certificate is successfully imported.

Figure 4-12 Example of a correct response code

```
[192.168.10.216 test]#curl -k --cert client.crt --key client.key https://192.168.10.16:4500 -I  
HTTP/1.1 200 OK  
Date: Fri, 25 Sep 2020 10:11:17 GMT  
Content-Type: application/octet-stream  
Connection: keep-alive  
Set-Cookie: name=d92f80b6-55e9-4b61-9c37-932ccd7b02f2; path=/; Expires=Sat, 26-Sep-20 10:11:19 GMT  
Server: elb
```


5 Using ELB to Redirect HTTP Requests to an HTTPS Listener for Higher Service Security

Scenarios

HTTPS is an extension of HTTP. HTTPS encrypts data between a web server and a browser. You can use ELB to redirect HTTP requests to an HTTPS listener to improve your service security.

CAUTION

- If the listener protocol is HTTP, only the GET or HEAD method can be used for redirection. If you create a redirect for an HTTP listener, the client browser will change POST or other methods to GET. If you want to use other methods rather than GET and HEAD, add an HTTPS listener.
 - HTTP requests are forwarded to the HTTPS listener as HTTPS requests, which are then routed to backend servers over HTTP.
 - If HTTP requests are redirected to an HTTPS listener, no certificate can be deployed on the backend servers associated with the HTTPS listener. If certificates are deployed, HTTPS requests will not take effect.
-

Prerequisites

- You have created a dedicated load balancer. For details, see [Creating a Dedicated Load Balancer](#).
- You have created two ECSs (ECS_client and ECS_server) that are running in the same VPC as the dedicated load balancer. ECS_client sends HTTP requests, while ECS_server processes requests. For details, see [Purchasing an ECS](#).
- You have gotten a server certificate ready for adding an HTTPS listener. For details, see [Adding a Server Certificate](#).

Procedure

Figure 5-1 Procedure for redirecting HTTPS requests to an HTTPS listener



Step 1: Create an HTTPS Listener

1. Go to the [load balancer list page](#).
2. On the displayed page, locate the target load balancer and click its name.
3. On the **Listeners** tab, click **Add Listener**. Configure the parameters based on [Table 5-1](#).

Figure 5-2 Adding an HTTPS listener

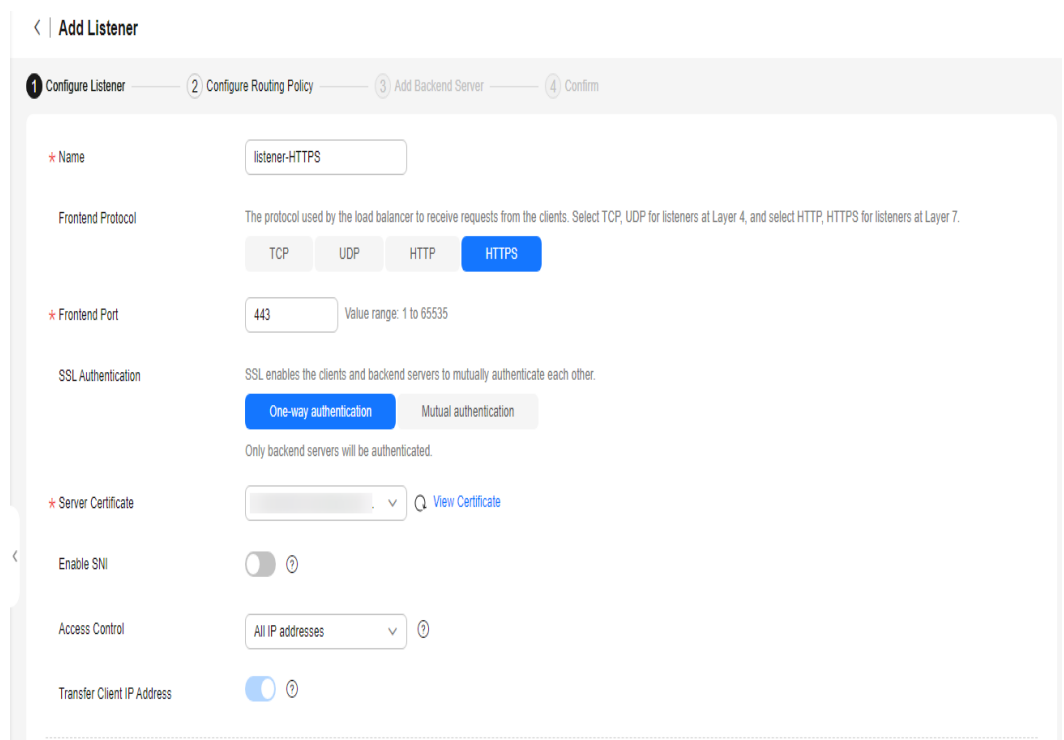


Table 5-1 Parameters for configuring an HTTPS listener

Parameter	Example Value	Description
Name	listener-HTTPS	Specifies the listener name.
Frontend Protocol	HTTPS	Specifies the protocol that will be used by the load balancer to receive requests from clients.

Parameter	Example Value	Description
Frontend Port	443	Specifies the port that will be used by the load balancer to receive requests from clients.
SSL Authentication	One-way authentication	Specifies how you want the clients and backend servers to be authenticated. In this practice, One-way authentication is selected.
Server Certificate	The existing server certificate	Specifies the certificate that will be used by the backend server for SSL handshake negotiation to authenticate clients and ensure encrypted transmission.
Enable SNI	Not enabled	Specifies whether to enable SNI when HTTPS is used as the frontend protocol. SNI can be used when a server uses multiple domain names and certificates.
Access Control	All IP addresses	Specifies how access to the listener is controlled. Access from specific IP addresses can be controlled using a whitelist or blacklist.
Transfer Client IP Address	Enabled by default	Specifies whether to transmit IP addresses of the clients to backend servers.
Advanced Forwarding	Enabled	Specifies whether to enable the advanced forwarding policy. You can add advanced forwarding policies to HTTP or HTTPS listeners to forward requests to different backend server groups.

4. Retain the default values for parameters under **Advanced Settings** and click **Next: Configure Request Routing Policy**.
5. Select **Create new** for **Backend Server Group**, retain the default values for other parameters, and click **Next: Add Backend Server**.
6. Add **ECS_server** to the backend server group you have created, enable **Health Check**, and retain the default values for the health check.
7. Click **Next: Confirm** and then click **Submit**.

Step 2: Configure HTTP to HTTPS Redirection

You can enable redirection when adding an HTTP listener and select an HTTPS listener to which requests are redirected. Alternatively, you can add a forwarding policy for an HTTP listener to redirect requests to an HTTPS listener.

Adding an HTTP Listener and Enabling Redirection



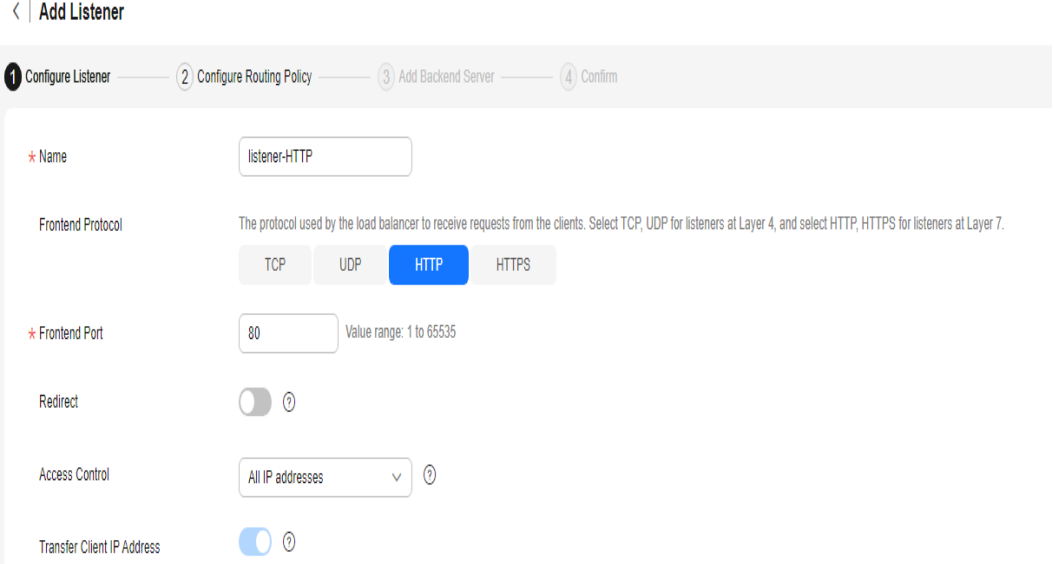
1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click  in the upper left corner to display **Service List** and choose **Networking > Elastic Load Balance**.
4. On the **Load Balancers** page, locate the target load balancer and click its name.
5. On the **Listeners** tab, click **Add Listener**. Configure the parameters based on [Table 5-2](#).

Figure 5-3 Adding an HTTP Listener



< | Add Listener

1 Configure Listener — 2 Configure Routing Policy — 3 Add Backend Server — 4 Confirm

* Name

Frontend Protocol The protocol used by the load balancer to receive requests from the clients. Select TCP, UDP for listeners at Layer 4, and select HTTP, HTTPS for listeners at Layer 7.
 TCP UDP HTTP HTTPS

* Frontend Port Value range: 1 to 65535

Redirect

Access Control

Transfer Client IP Address

Table 5-2 Parameters for configuring an HTTP listener

Parameter	Example Value	Description
Name	listener-HTTP	Specifies the listener name.
Frontend Protocol	HTTP	Specifies the protocol that will be used by the load balancer to receive requests from clients.
Frontend Port	80	Specifies the port that will be used by the load balancer to receive requests from clients.
Redirect	Enabled	Specifies whether to enable redirection. You can use this function to redirect the requests from an HTTP listener to an HTTPS listener to ensure security.

Parameter	Example Value	Description
Redirected To	listener-HTTPS	Specifies the HTTPS listener to which requests are redirected. Select the HTTPS listener created in section Step 1: Create an HTTPS Listener , listener-HTTPS .
Access Control	All IP addresses	Specifies how access to the listener is controlled. Access from specific IP addresses can be controlled using a whitelist or blacklist.
Transfer Client IP Address	Enabled by default	Specifies whether to transmit IP addresses of the clients to backend servers.
Advanced Forwarding	Enabled	Specifies whether to enable the advanced forwarding policy. You can add advanced forwarding policies to HTTP or HTTPS listeners to forward requests to different backend server groups.

6. Retain the default values for parameters under **Advanced Settings** and click **Next: Confirm**.
7. Click **Submit**.

Adding an HTTP Listener and Configuring a Forwarding Policy to Redirect Requests



1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.
3. Click  in the upper left corner to display **Service List** and choose **Networking > Elastic Load Balance**.
4. On the **Load Balancers** page, locate the target load balancer and click its name.
5. On the **Listeners** tab, click **Add Listener**. Configure the parameters based on [Table 5-3](#).

Figure 5-4 Adding an HTTP Listener

< | Add Listener

1 Configure Listener — 2 Configure Routing Policy — 3 Add Backend Server — 4 Confirm

* Name

Frontend Protocol The protocol used by the load balancer to receive requests from the clients. Select TCP, UDP for listeners at Layer 4, and select HTTP, HTTPS for listeners at Layer 7.

TCP UDP HTTP HTTPS

* Frontend Port Value range: 1 to 65535

Redirect

Access Control ?

Transfer Client IP Address ?

Table 5-3 Parameters for configuring an HTTP listener

Parameter	Example Value	Description
Name	listener-HTTP	Specifies the listener name.
Frontend Protocol	HTTP	Specifies the protocol that will be used by the load balancer to receive requests from clients.
Frontend Port	80	Specifies the port that will be used by the load balancer to receive requests from clients.
Redirect	Not enabled	Specifies whether to enable redirection. You can use this function to redirect the requests from an HTTP listener to an HTTPS listener to ensure security.
Access Control	All IP addresses	Specifies how access to the listener is controlled. Access from specific IP addresses can be controlled using a whitelist or blacklist.
Transfer Client IP Address	Enabled by default	Specifies whether to transmit IP addresses of the clients to backend servers.
Advanced Forwarding	Enabled	Specifies whether to enable the advanced forwarding policy. You can add advanced forwarding policies to HTTP or HTTPS listeners to forward requests to different backend server groups.

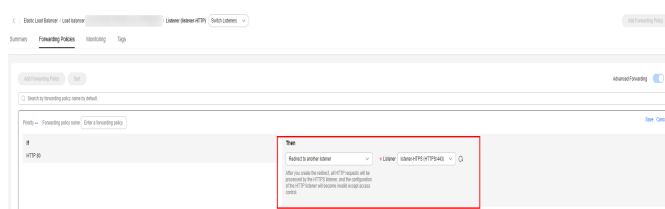
6. Retain the default values for parameters under **Advanced Settings** and click **Next: Configure Request Routing Policy**.
7. Select **Create new** for **Backend Server Group**, retain the default values for other parameters, and click **Next: Add Backend Server**.
8. Add **ECS_server** to the backend server group you have created, enable **Health Check**, and retain the default values for the health check.
9. Click **Next: Confirm** and then click **Submit**.
10. On the **Configuration Result** page, click **Add now** under the **Next: Add a Forwarding Policy (Optional)** area.
11. Click **Add Forwarding Policy** to configure redirection.

Table 5-4 Configuring parameters for redirection

Parameter	Setting
Action	Select Redirect to another listener .
Listener	Select the HTTPS listener to which requests are redirected.

12. After the forwarding policy is added, click **Save**.

Figure 5-5 Redirection to an HTTPS listener



NOTE

- After the redirection is added, the configurations for the HTTP listener will not be applied, but access control configured for that listener will still be applied.
- After the redirection is added for an HTTP listener, the backend server will return 301 Moved Permanently to the clients.

Step 3: Verify the Redirection to HTTPS

Remotely log in to **ECS_client** and run **curl -H "Accept-Language: zh-CN,zh" "http://ELB-private-IP-address.80"** to check whether HTTP requests are redirected.

If 301 Moved Permanently is returned, as shown in the below figure, HTTP requests are directed to an HTTP listener.

Figure 5-6 Verifying redirection to an HTTPS listener

```
]#curl -H "Accept-Language: zh-CN,zh" 'http:// :80'  
<html>  
<head><title>301 Moved Permanently</title></head>  
<body>  
<center><h1>301 Moved Permanently</h1></center>  
</body>  
</html>
```