Elastic Load Balance

Best Practices

Issue	01
Date	2024-07-11

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

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

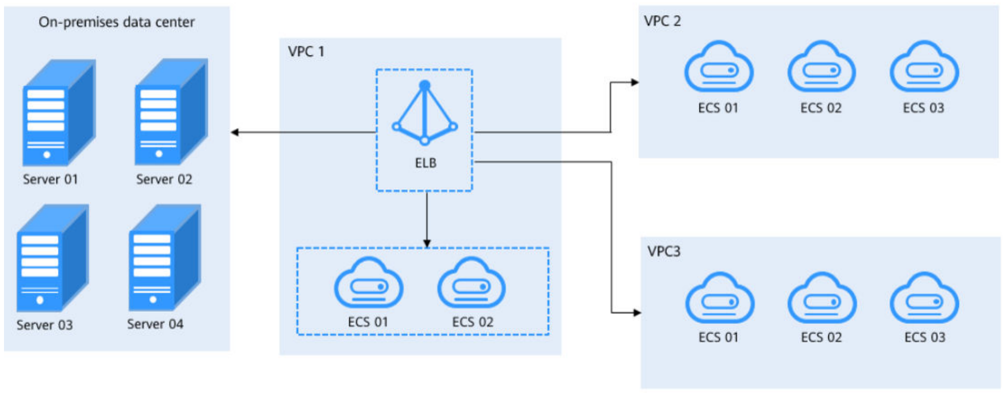
1.1 Overview

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

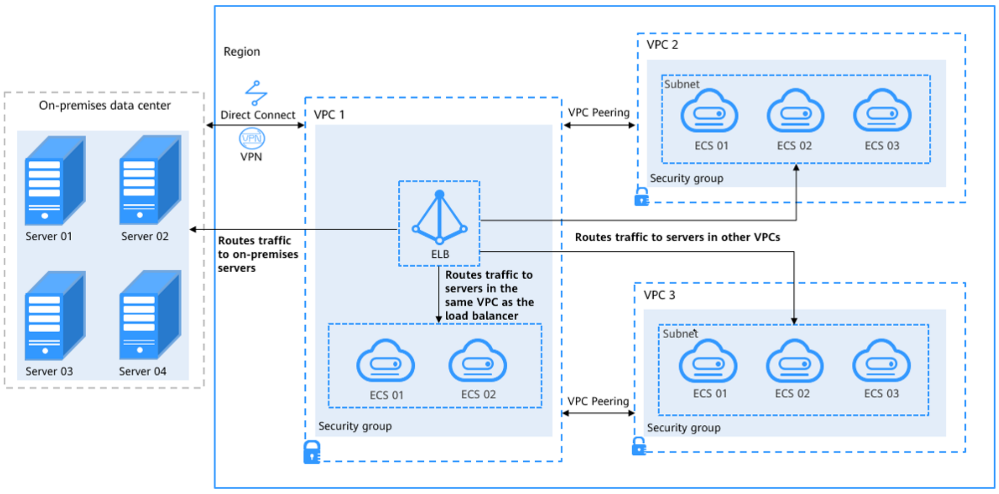
You can enable **IP as a Backend** when creating a dedicated load balancer and associate cloud and on-premises servers with this dedicated load balancer using their IP addresses.

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

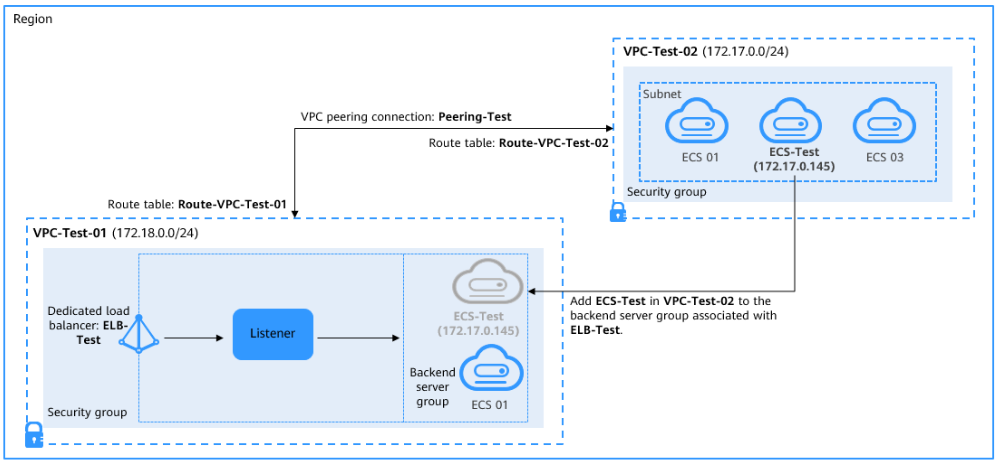
Scenarios

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


Region
VPC-Test-02 (172.17.0.0/24)
Subnet
ECS 01
ECS-Test (172.17.0.145)
ECS 03
Security group
VPC peering connection: Peering-Test
Route table: Route-VPC-Test-02
Route table: Route-VPC-Test-01
VPC-Test-01 (172.18.0.0/24)
Dedicated load balancer: ELB-Test
Listener
ECS-Test (172.17.0.145)
Backend server group
ECS 01
Security group
Add ECS-Test in VPC-Test-02 to the backend server group associated with ELB-Test.

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

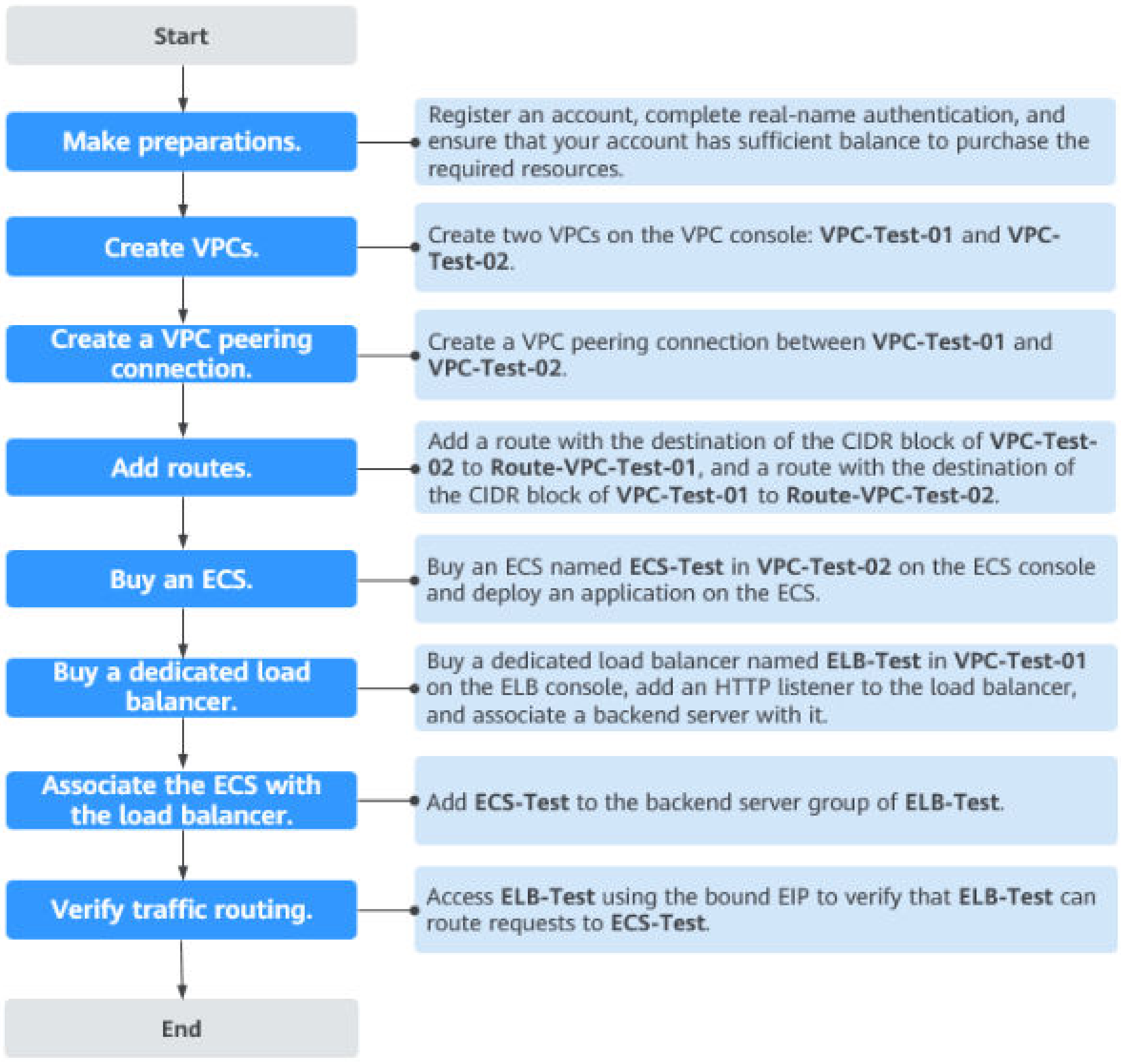
The actual cost shown on the Huawei Cloud console is used.

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

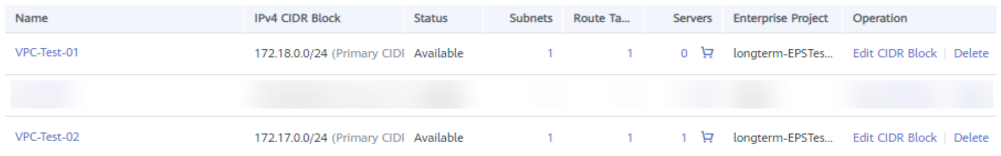

Start
Make preparations.
Register an account, complete real-name authentication, and ensure that your account has sufficient balance to purchase the required resources.
Create VPCs.
Create two VPCs on the VPC console: VPC-Test-01 and VPC-Test-02.
Create a VPC peering connection.
Create a VPC peering connection between VPC-Test-01 and VPC-Test-02.
Add routes.
Add a route with the destination of the CIDR block of VPC-Test-02 to Route-VPC-Test-01, and a route with the destination of the CIDR block of VPC-Test-01 to Route-VPC-Test-02.
Buy an ECS.
Buy an ECS named ECS-Test in VPC-Test-02 on the ECS console and deploy an application on the ECS.
Buy a dedicated load balancer.
Buy a dedicated load balancer named ELB-Test in VPC-Test-01 on the ELB console, add an HTTP listener to the load balancer, and associate a backend server with it.
Associate the ECS with the load balancer.
Add ECS-Test to the backend server group of ELB-Test.
Verify traffic routing.
Access ELB-Test using the bound EIP to verify that ELB-Test can route requests to ECS-Test.
End

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 CIDI	Available	1	1	0 🛒	longterm-EPSTes...	Edit CIDR Block | Delete
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
VPC-Test-02	172.17.0.0/24 (Primary CIDI	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**

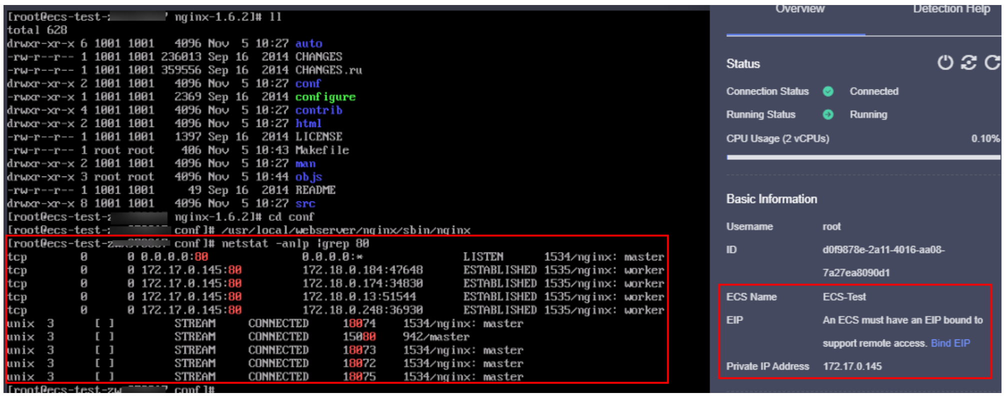
Step 4: Create an ECS

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 **Elastic Cloud Server User Guide**.

4. Deploy Nginx on the **ECS-Test**.

Figure 1-6 Deploying Nginx on **ECS-Test**



```
[root@ecs-test-[redacted] nginx-1.6.2]# ll
total 628
drwxr-xr-x 6 1001 1001   4096 Nov  5 10:27 auto
-rw-r--r-- 1 1001 1001 236013 Sep 16  2014 CHANGES
-rw-r--r-- 1 1001 1001 359556 Sep 16  2014 CHANGES.ru
drwxr-xr-x 2 1001 1001   4096 Nov  5 10:27 conf
-rwxr-xr-x 1 1001 1001   2369 Sep 16  2014 configure
drwxr-xr-x 4 1001 1001   4096 Nov  5 10:27 contrib
drwxr-xr-x 2 1001 1001   4096 Nov  5 10:27 html
-rw-r--r-- 1 1001 1001   1397 Sep 16  2014 LICENSE
-rw-r--r-- 1 root root    406 Nov  5 10:43 Makefile
drwxr-xr-x 2 1001 1001   4096 Nov  5 10:27 man
drwxr-xr-x 3 root root   4096 Nov  5 10:44 objs
-rw-r--r-- 1 1001 1001     49 Sep 16  2014 README
drwxr-xr-x 8 1001 1001   4096 Nov  5 10:27 src
[root@ecs-test-[redacted] nginx-1.6.2]# cd conf
[root@ecs-test-[redacted] conf]# /usr/local/webserver/nginx/sbin/nginx
[root@ecs-test-z[redacted] conf]# netstat -anlp |grep 80
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      1534/nginx: master
tcp        0      0 172.17.0.145:80         172.18.0.184:47648      ESTABLISHED 1535/nginx: worker
tcp        0      0 172.17.0.145:80         172.18.0.174:34830      ESTABLISHED 1535/nginx: worker
tcp        0      0 172.17.0.145:80         172.18.0.13:51544       ESTABLISHED 1535/nginx: worker
tcp        0      0 172.17.0.145:80         172.18.0.248:36930      ESTABLISHED 1535/nginx: worker
unix  3      [ ]         STREAM     CONNECTED     18074    1534/nginx: master
unix  3      [ ]         STREAM     CONNECTED     15080    942/master
unix  3      [ ]         STREAM     CONNECTED     18073    1534/nginx: master
unix  3      [ ]         STREAM     CONNECTED     18072    1534/nginx: master
unix  3      [ ]         STREAM     CONNECTED     18075    1534/nginx: master
[root@ecs-test-zw[redacted] conf]#
```

Status

Connection Status	Connected
Running Status	Running
CPU Usage (2 vCPUs)	0.10%

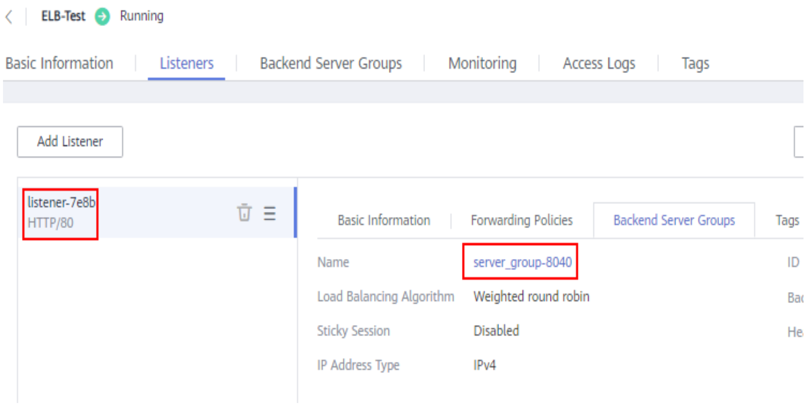
Basic Information

Username	root
ID	d0f9878e-2a11-4016-aa08-7a27ea8090d1
ECS Name	ECS-Test
EIP	An ECS must have an EIP bound to support remote access. Bind EIP
Private IP Address	172.17.0.145

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



Name	server_group-8040
Load Balancing Algorithm	Weighted round robin
Sticky Session	Disabled
IP Address Type	IPv4

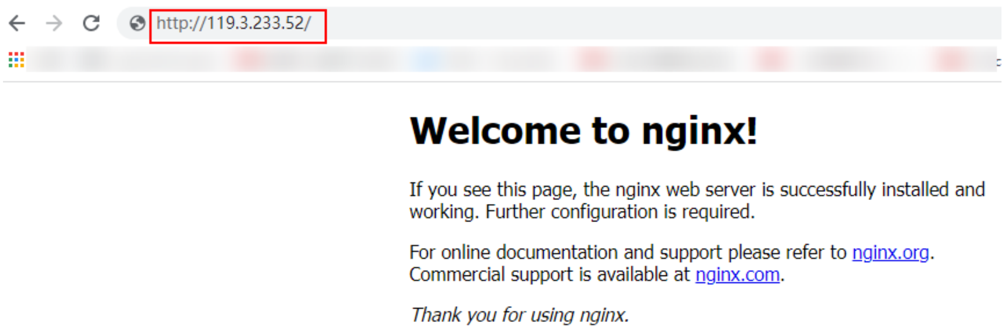
Step 6: Add ECS-Test to the Backend Server Group

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**



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

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

Scenarios

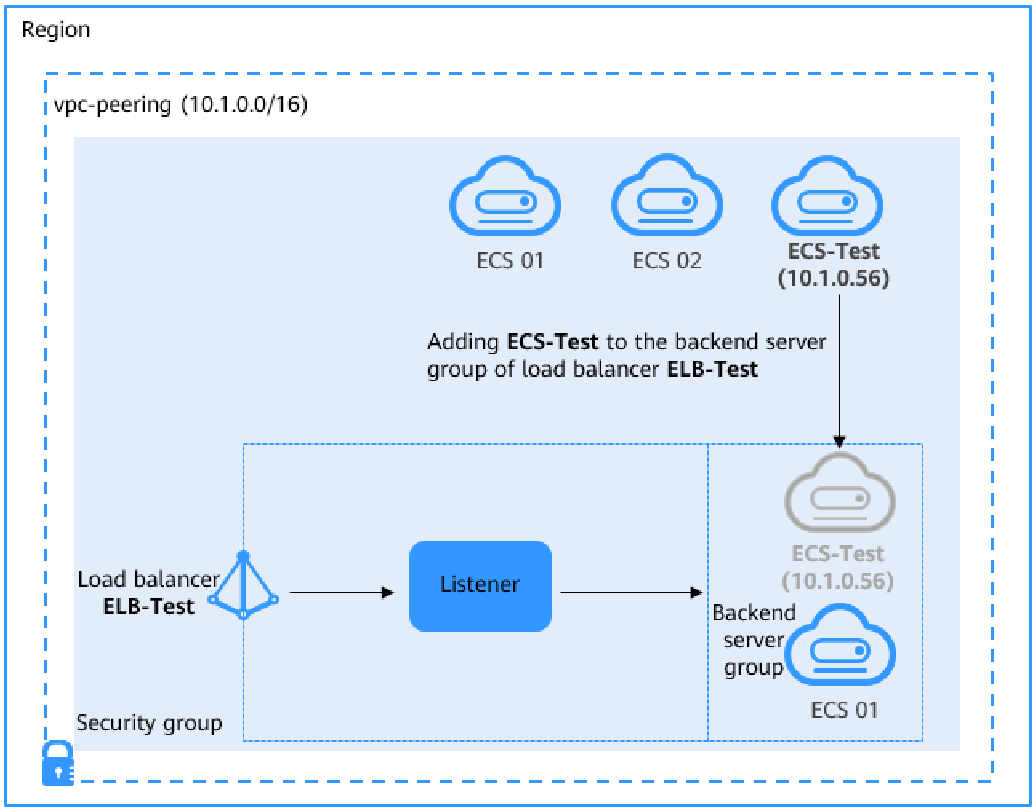
You can use ELB to route traffic to backend servers in the same VPC as the load balancer.

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.

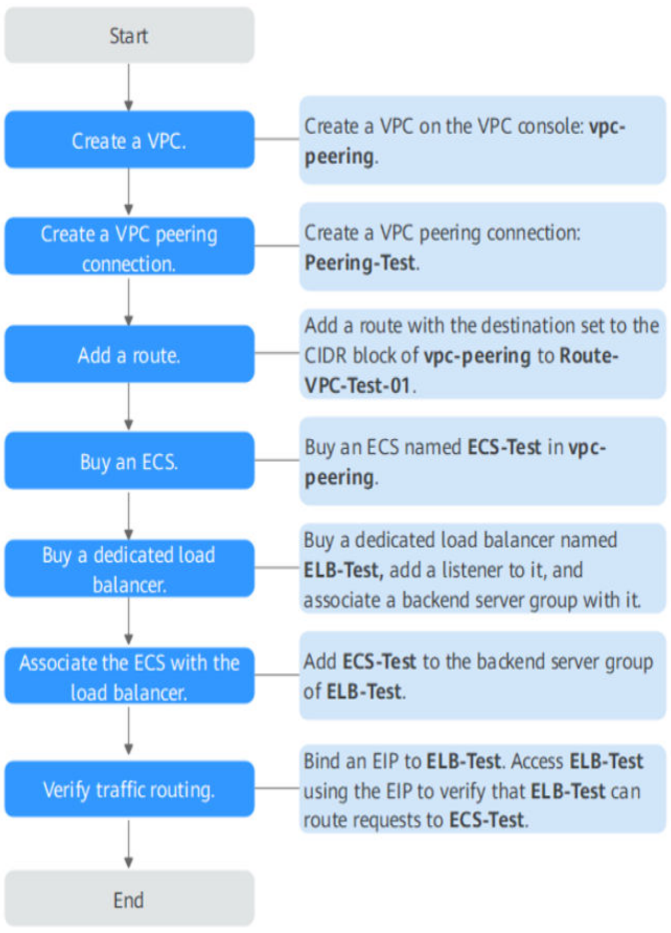
Table 1-2 Resource planning

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


Start
Create a VPC.
Create a VPC on the VPC console: **vpc-peering**.
Create a VPC peering connection.
Create a VPC peering connection: **Peering-Test**.
Add a route.
Add a route with the destination set to the CIDR block of **vpc-peering** to **Route-VPC-Test-01**.
Buy an ECS.
Buy an ECS named **ECS-Test** in **vpc-peering**.
Buy a dedicated load balancer.
Buy a dedicated load balancer named **ELB-Test,** add a listener to it, and associate a backend server group with it.
Associate the ECS with the load balancer.
Add **ECS-Test** to the backend server group of **ELB-Test**.
Verify traffic routing.
Bind an EIP to **ELB-Test**. Access **ELB-Test** using the EIP to verify that **ELB-Test** can route requests to **ECS-Test**.
End

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

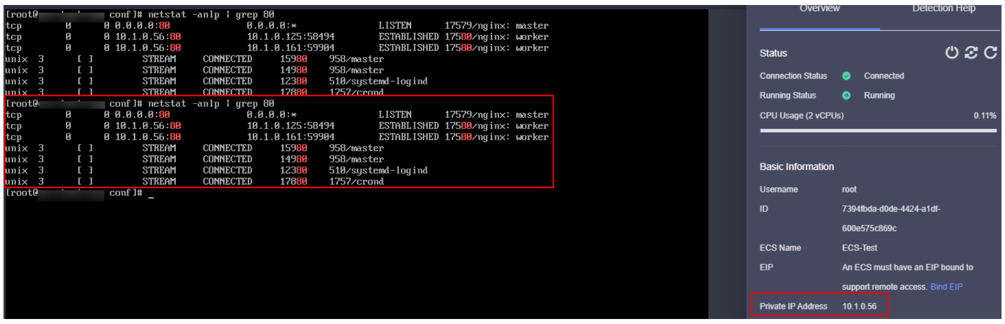
1. In the navigation pane on the left, click **Route Tables**.
2. In the upper right corner, click **Create Route Table**.
3. Configure the parameters as follows and click **OK**. For details on how to create a route table, see the ***Virtual Private Cloud User Guide***.
   - **Name**: **Route-VPC-Test-01**
   - **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 **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**



```
[root@[redacted] conf]# netstat -anlp | grep 80
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      17579/nginx: master
tcp        0      0 10.1.0.56:80            10.1.0.125:58494        ESTABLISHED 17580/nginx: worker
tcp        0      0 10.1.0.56:80            10.1.0.161:59904        ESTABLISHED 17580/nginx: worker
unix  3      [ ]         STREAM     CONNECTED     15980    958/master
unix  3      [ ]         STREAM     CONNECTED     14980    958/master
unix  3      [ ]         STREAM     CONNECTED     12380    510/systemd-logind
unix  3      [ ]         STREAM     CONNECTED     17880    1757/crond
[root@[redacted] conf]# netstat -anlp | grep 80
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      17579/nginx: master
tcp        0      0 10.1.0.56:80            10.1.0.125:58494        ESTABLISHED 17580/nginx: worker
tcp        0      0 10.1.0.56:80            10.1.0.161:59904        ESTABLISHED 17580/nginx: worker
unix  3      [ ]         STREAM     CONNECTED     15980    958/master
unix  3      [ ]         STREAM     CONNECTED     14980    958/master
unix  3      [ ]         STREAM     CONNECTED     12380    510/systemd-logind
unix  3      [ ]         STREAM     CONNECTED     17880    1757/crond
[root@[redacted] conf]# _
```

Status

Connection Status	Connected
Running Status	Running
CPU Usage (2 vCPUs)	0.11%

Basic Information

Username	root
ID	7394fbda-d0de-4424-a1df-600e575c869c
ECS Name	ECS-Test
EIP	An ECS must have an EIP bound to support remote access. Bind EIP
Private IP Address	10.1.0.56

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

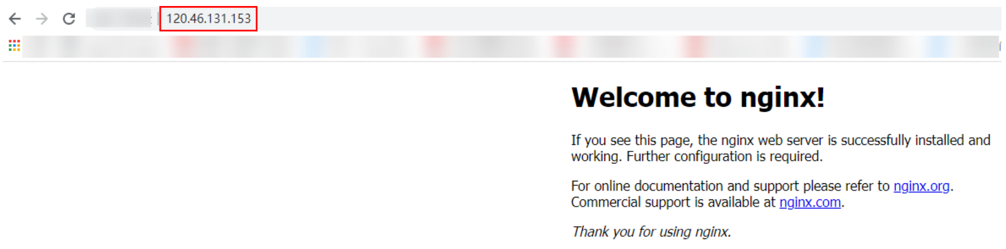
Locate **ELB-Test** and click its name.

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**



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

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

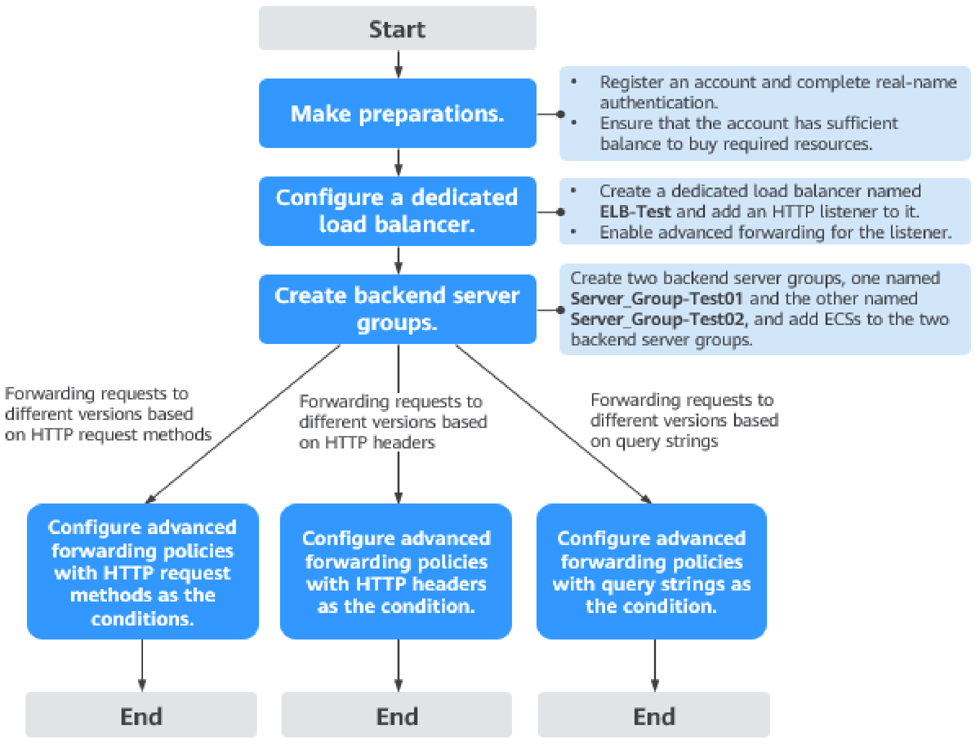

Start
Make preparations.
• Register an account and complete real-name authentication.
• Ensure that the account has sufficient balance to buy required resources.
Configure a dedicated load balancer.
• Create a dedicated load balancer named ELB-Test and add an HTTP listener to it.
• Enable advanced forwarding for the listener.
Create backend server groups.
Create two backend server groups, one named Server_Group-Test01 and the other named Server_Group-Test02, and add ECSs to the two backend server groups.
Forwarding requests to different versions based on HTTP request methods
Forwarding requests to different versions based on HTTP headers
Forwarding requests to different versions based on query strings
Configure advanced forwarding policies with HTTP request methods as the conditions.
Configure advanced forwarding policies with HTTP headers as the condition.
Configure advanced forwarding policies with query strings as the condition.
End
End
End

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**.

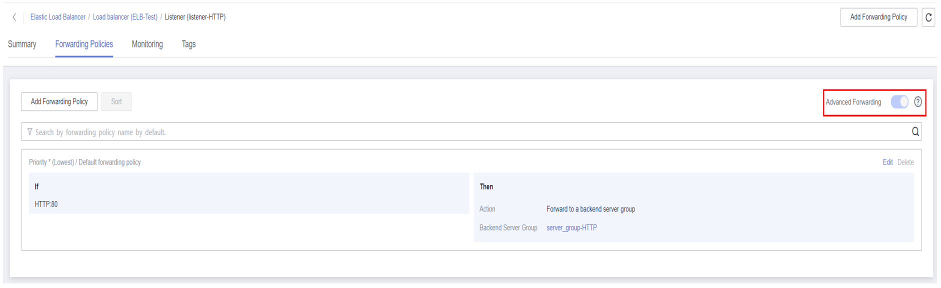
NOTE

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


Elastic Load Balancer / Load balancer (ELB-Test) / Listener (listener-HTTP)
Add Forwarding Policy
Summary
Forwarding Policies
Monitoring
Tags
Add Forwarding Policy
Sort
Advanced Forwarding
Search by forwarding policy name by default.
Edit
Delete

Priority * (Lowest) / Default forwarding policy

If
HTTP:80

Then
Action Forward to a backend server group
Backend Server Group server_group-HTTP

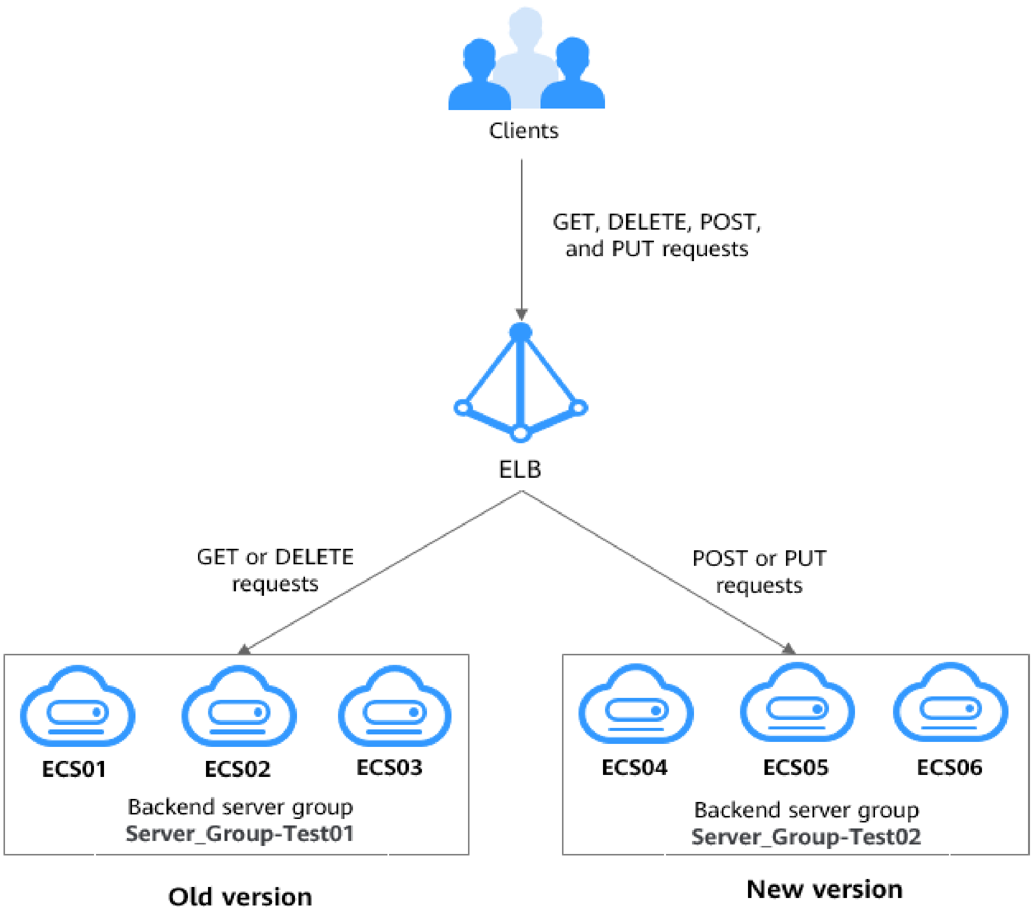
Step 2: Create Two Backend Server Groups and Adde Backend Servers to Them

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.

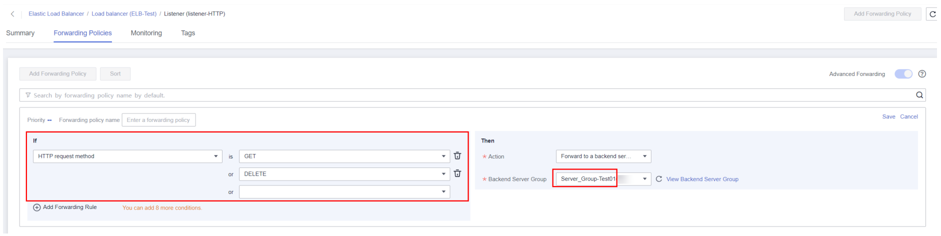
Figure 2-3 Forwarding requests based on HTTP request methods


Clients
GET, DELETE, POST, and PUT requests
ELB
GET or DELETE requests
POST or PUT requests
ECS01
ECS02
ECS03
Backend server group Server_Group-Test01
ECS04
ECS05
ECS06
Backend server group Server_Group-Test02
Old version
New version

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

   
   Elastic Load Balancer / Load balancer (ELB-Test) / Listener (listener-HTTP)

   Priority -- Forwarding policy name: Enter a forwarding policy

   If: HTTP request method is GET, or DELETE, or

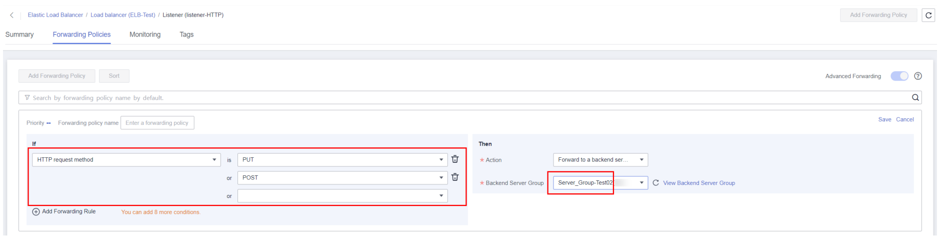
   Then: Action: Forward to a backend ser... ; Backend Server Group: Server_Group-Test01

   You can add 8 more conditions.

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


Elastic Load Balancer / Load balancer (ELB-Test) / Listener (listener-HTTP)
Add Forwarding Policy
Summary
Forwarding Policies
Monitoring
Tags
Add Forwarding Policy
Sort
Advanced Forwarding
Search by forwarding policy name by default.
Save
Cancel
Add Forwarding Rule
View Backend Server Group

Priority -- Forwarding policy name: Enter a forwarding policy

If
HTTP request method is PUT
or POST
or

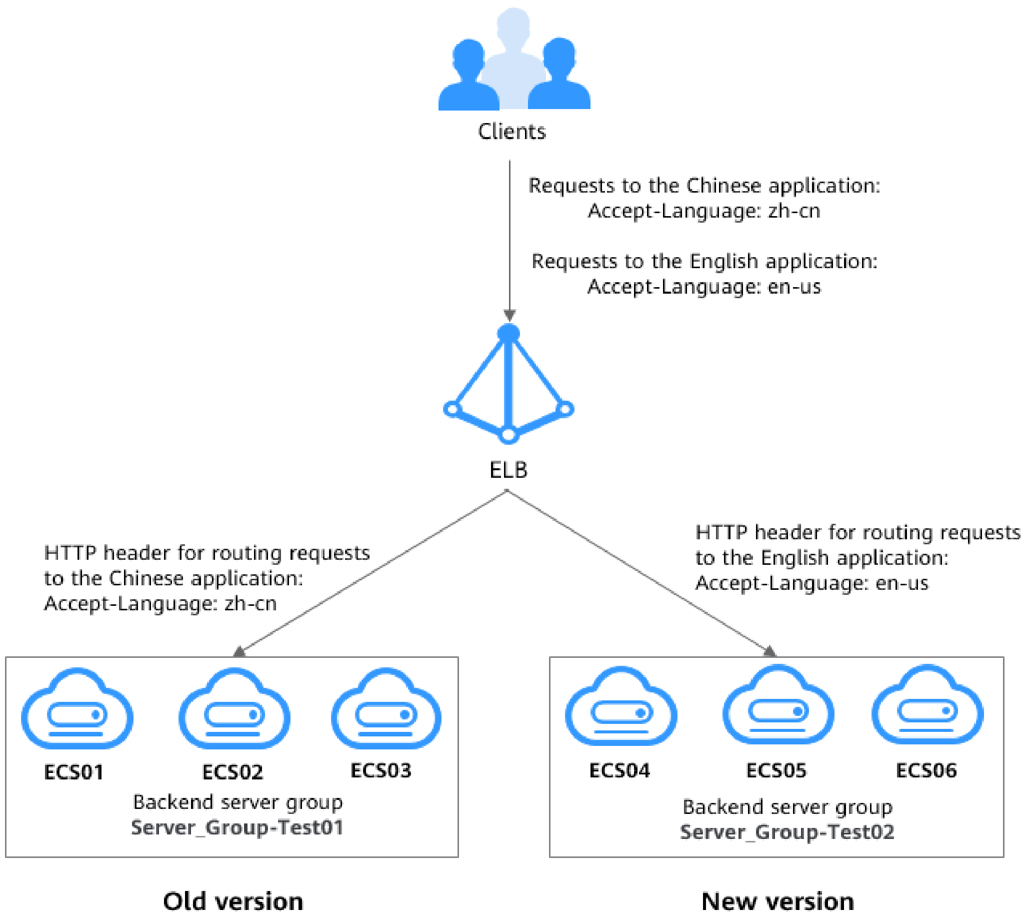
You can add 8 more conditions.

Then
* Action: Forward to a backend ser...
* Backend Server Group: Server_Group-Test02

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.

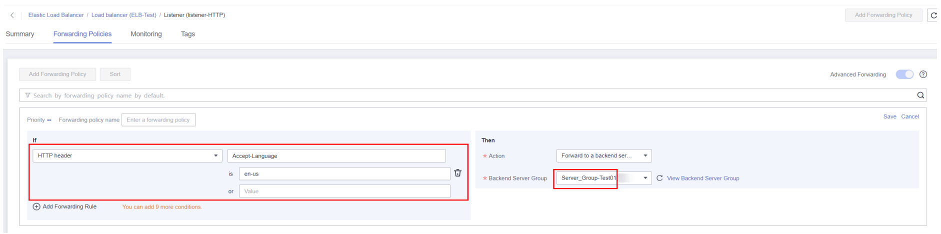
Figure 2-6 Smooth application transition between the old and new versions based on the HTTP request header



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

   Priority -- Forwarding policy name: Enter a forwarding policy

   If: HTTP header — Accept-Language — is en-us — or Value

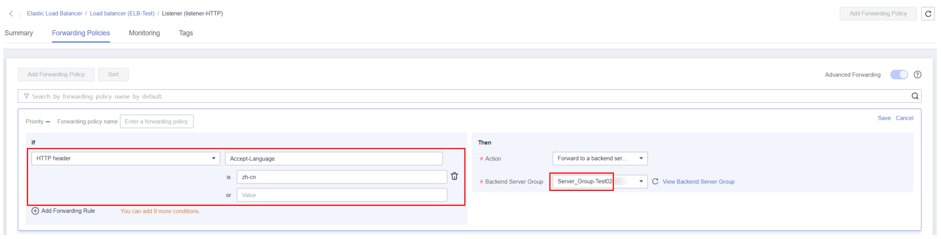
   You can add 9 more conditions.

   Then: * Action: Forward to a backend ser... — * Backend Server Group: Server_Group-Test01 — View Backend Server Group

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


Elastic Load Balancer / Load balancer (ELB-Test) / Listener (listener-HTTP)
Add Forwarding Policy
Summary
Forwarding Policies
Monitoring
Tags
Add Forwarding Policy
Sort
Advanced Forwarding
Search by forwarding policy name by default.
Save
Cancel
Add Forwarding Rule
View Backend Server Group

Priority -- Forwarding policy name: Enter a forwarding policy

If

HTTP header — Accept-Language

is zh-cn

or Value

You can add 9 more conditions.

Then

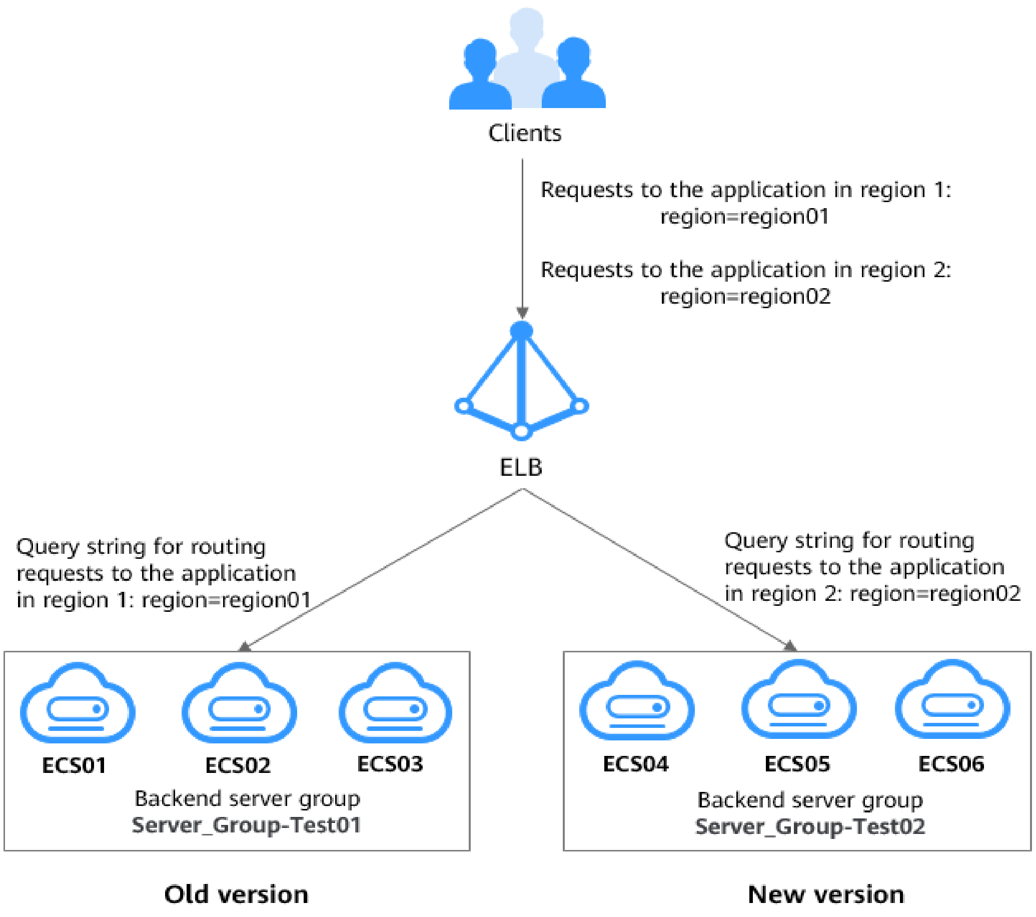
* Action: Forward to a backend ser...

* Backend Server Group: Server_Group-Test02

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


Clients
Requests to the application in region 1: region=region01
Requests to the application in region 2: region=region02
ELB
Query string for routing requests to the application in region 1: region=region01
Query string for routing requests to the application in region 2: region=region02
ECS01
ECS02
ECS03
Backend server group Server_Group-Test01
ECS04
ECS05
ECS06
Backend server group Server_Group-Test02
Old version
New version

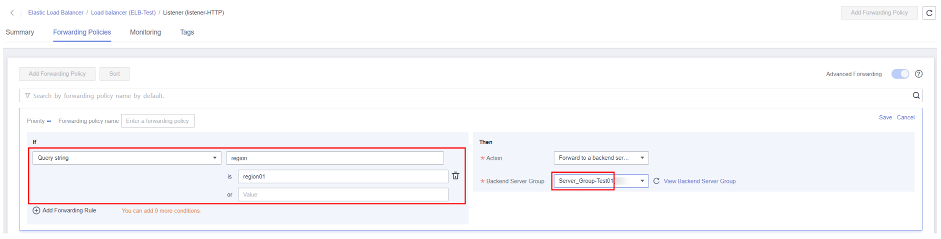
NOTE

- 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



Elastic Load Balancer / Load balancer (ELB-Test) / Listener (listener-HTTP)

Priority -- Forwarding policy name: Enter a forwarding policy

If: Query string, region, is region01, or Value

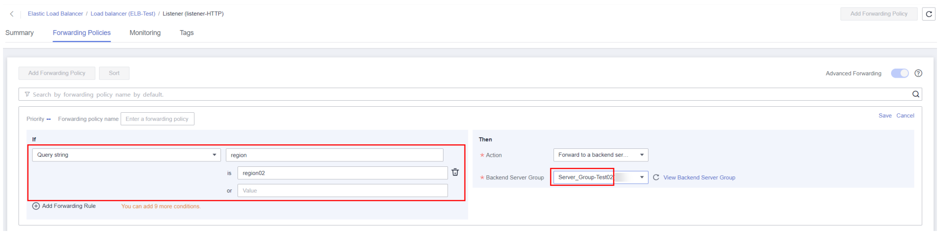
You can add 9 more conditions.

Then: Action: Forward to a backend ser...; Backend Server Group: Server_Group-Test01

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



Elastic Load Balancer / Load balancer (ELB-Test) / Listener (listener-HTTP)

Priority -- Forwarding policy name: Enter a forwarding policy

If: Query string, region, is region02, or Value

You can add 9 more conditions.

Then: Action: Forward to a backend ser...; Backend Server Group: Server_Group-Test02

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.

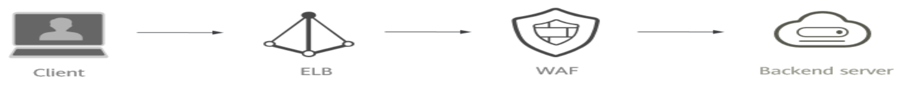
Constraints

- 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

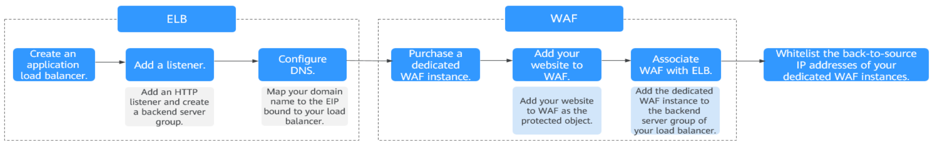
After WAF is integrated with ELB, the traffic flow is as illustrated in **Figure 3-1**.

Figure 3-1 Traffic path



Procedure

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

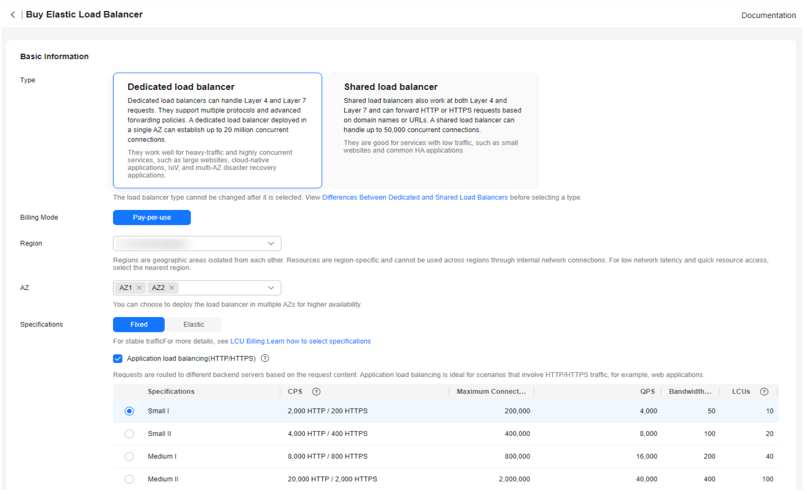

ELB
Create an application load balancer.
Add a listener.
Add an HTTP listener and create a backend server group.
Configure DNS.
Map your domain name to the EIP bound to your load balancer.
WAF
Purchase a dedicated WAF instance.
Add your website to WAF.
Add your website to WAF as the protected object.
Associate WAF with ELB.
Add the dedicated WAF instance to the backend server group of your load balancer.
Whitelist the back-to-source IP addresses of your dedicated WAF instances.

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)

   Basic Information

   Type

   Dedicated load balancer
   Dedicated load balancers can handle Layer 4 and Layer 7 requests. They support multiple protocols and advanced forwarding policies. A dedicated load balancer deployed in a single AZ can establish up to 20 million concurrent connections.
   They work well for heavy-traffic and highly concurrent services, such as large websites, cloud-native applications, IoV, and multi-AZ disaster recovery applications.

   Shared load balancer
   Shared load balancers also work at both Layer 4 and Layer 7 and can forward HTTP or HTTPS requests based on domain names or URLs. A shared load balancer can handle up to 50,000 concurrent connections.
   They are good for services with low traffic, such as small websites and common HA applications.

   The load balancer type cannot be changed after it is selected. View Differences Between Dedicated and Shared Load Balancers before selecting a type.

   Billing Mode: Pay-per-use

   Region

   Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region.

   AZ: AZ1 × AZ2 ×

   You can choose to deploy the load balancer in multiple AZs for higher availability.

   Specifications: Fixed | Elastic

   For stable trafficFor more details, see LCU Billing.Learn how to select specifications

   ☑ Application load balancing(HTTP/HTTPS)

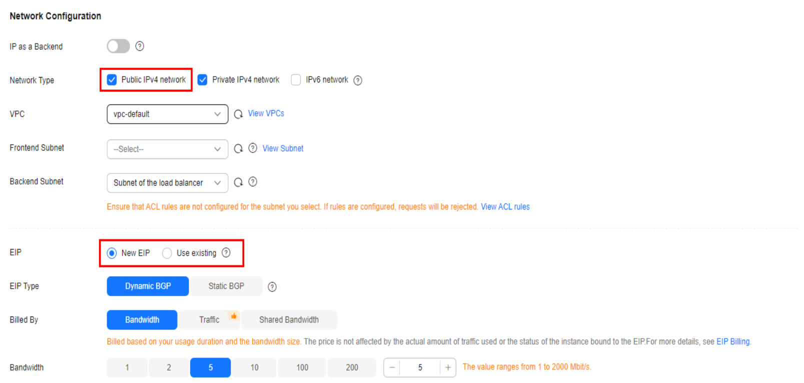
   Requests are routed to different backend servers based on the request content. Application load balancing is ideal for scenarios that involve HTTP/HTTPS traffic, for example, web applications.

	Specifications	CPS	Maximum Connect...	QPS	Bandwidth...	LCUs
●	Small I	2,000 HTTP / 200 HTTPS	200,000	4,000	50	10
○	Small II	4,000 HTTP / 400 HTTPS	400,000	8,000	100	20
○	Medium I	8,000 HTTP / 800 HTTPS	800,000	16,000	200	40
○	Medium II	20,000 HTTP / 2,000 HTTPS	2,000,000	40,000	400	100

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



Network Configuration

IP as a Backend	Off
Network Type	[✓ Public IPv4 network] [✓ Private IPv4 network] IPv6 network
VPC	vpc-default · View VPCs
Frontend Subnet	--Select-- · View Subnet
Backend Subnet	Subnet of the load balancer
	Ensure that ACL rules are not configured for the subnet you select. If rules are configured, requests will be rejected. View ACL rules
EIP	[✓ New EIP] Use existing
EIP Type	[✓ Dynamic BGP] Static BGP
Billed By	[✓ Bandwidth] Traffic Shared Bandwidth
	Billed based on your usage duration and the bandwidth size. The price is not affected by the actual amount of traffic used or the status of the instance bound to the EIP.For more details, see EIP Billing.
Bandwidth	1 2 [✓ 5] 10 100 200 · 5 · The value ranges from 1 to 2000 Mbit/s.

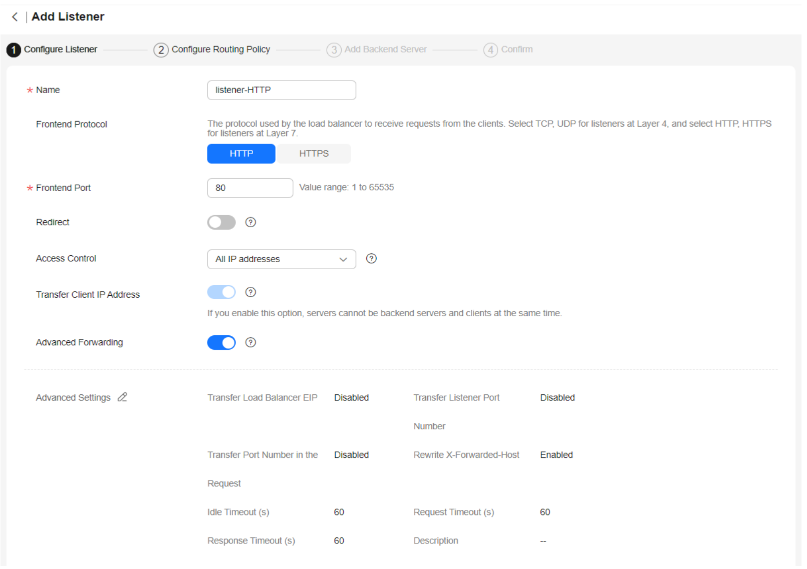
6. Confirm the information, click **Next**, and submit your request.

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



Add Listener

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

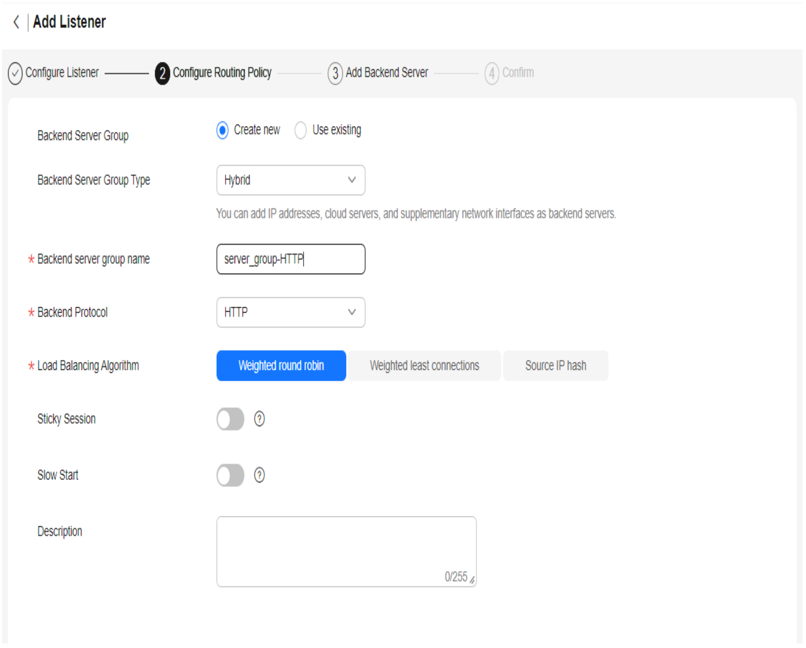
* Name	listener-HTTP
Frontend Protocol	The protocol used by the load balancer to receive requests from the clients. Select TCP, UDP for listeners at Layer 4, and select HTTP, HTTPS for listeners at Layer 7.
	[✓ HTTP] HTTPS
* Frontend Port	80 · Value range: 1 to 65535
Redirect	Off
Access Control	All IP addresses
Transfer Client IP Address	On
	If you enable this option, servers cannot be backend servers and clients at the same time.
Advanced Forwarding	On

Advanced Settings

Transfer Load Balancer EIP	Disabled	Transfer Listener Port Number	Disabled
Transfer Port Number in the Request	Disabled	Rewrite X-Forwarded-Host	Enabled
Idle Timeout (s)	60	Request Timeout (s)	60
Response Timeout (s)	60	Description	--

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



Backend Server Group: [✓ Create new] Use existing

Backend Server Group Type: Hybrid

You can add IP addresses, cloud servers, and supplementary network interfaces as backend servers.

* Backend server group name: server_group-HTTP

* Backend Protocol: HTTP

* Load Balancing Algorithm: [✓ Weighted round robin] Weighted least connections Source IP hash

Sticky Session

Slow Start

Description: 0/255

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.

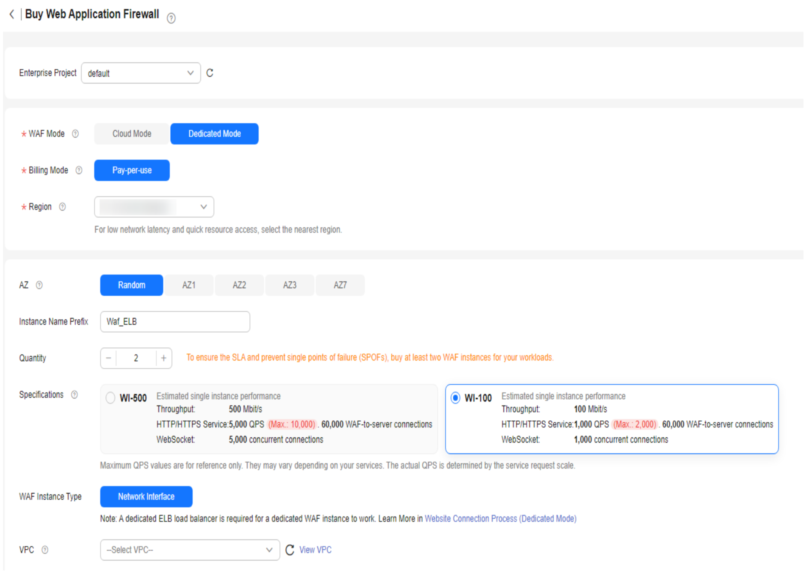
For details about how to configure DNS, see **Routing Internet Traffic to a Website**.

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**.

   For more details, see **Buying a Dedicated WAF Instance**.

Figure 3-7 Buying a dedicated WAF instance



Buy Web Application Firewall

Enterprise Project: default

* WAF Mode: Cloud Mode [✓ Dedicated Mode]

* Billing Mode: [✓ Pay-per-use]

* Region: [redacted]

For low network latency and quick resource access, select the nearest region.

AZ: [✓ Random] AZ1 AZ2 AZ3 AZ7

Instance Name Prefix: Waf_ELB

Quantity: 2 To ensure the SLA and prevent single points of failure (SPOFs), buy at least two WAF instances for your workloads.

Specifications:

WI-500 Estimated single instance performance
Throughput: 500 Mbit/s
HTTP/HTTPS Service:5,000 QPS (Max.: 10,000). 60,000 WAF-to-server connections
WebSocket: 5,000 concurrent connections

[✓ WI-100] Estimated single instance performance
Throughput: 100 Mbit/s
HTTP/HTTPS Service:1,000 QPS (Max.: 2,000). 60,000 WAF-to-server connections
WebSocket: 1,000 concurrent connections

Maximum QPS values are for reference only. They may vary depending on your services. The actual QPS is determined by the service request scale.

WAF Instance Type: [✓ Network Interface]

Note: A dedicated ELB load balancer is required for a dedicated WAF instance to work. Learn More in Website Connection Process (Dedicated Mode)

VPC: --Select VPC-- View VPC

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

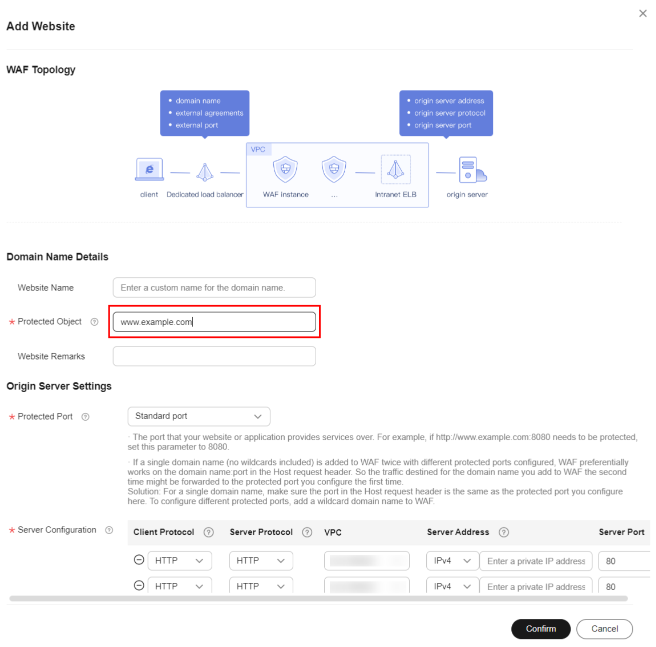
Adding Your Website to WAF

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



Domain Name Details

Website Name: Enter a custom name for the domain name.

* Protected Object: www.example.com

Website Remarks:

Origin Server Settings

* Protected Port: Standard port

· The port that your website or application provides services over. For example, if http://www.example.com:8080 needs to be protected, set this parameter to 8080.

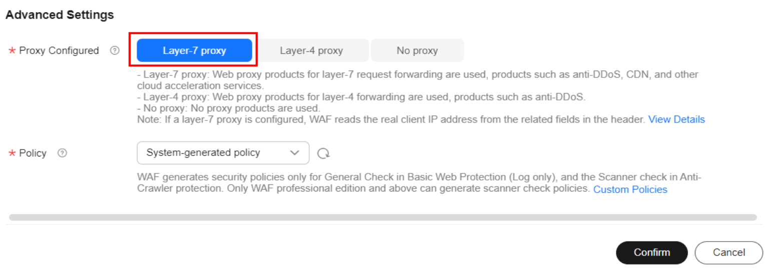
· If a single domain name (no wildcards included) is added to WAF twice with different protected ports configured, WAF preferentially works on the domain name:port in the Host request header. So the traffic destined for the domain name you add to WAF the second time might be forwarded to the protected port you configure the first time.
Solution: For a single domain name, make sure the port in the Host request header is the same as the protected port you configure here. To configure different protected ports, add a wildcard domain name to WAF.

* Server Configuration

Client Protocol	Server Protocol	VPC	Server Address		Server Port
HTTP	HTTP	[redacted]	IPv4	Enter a private IP address	80
HTTP	HTTP	[redacted]	IPv4	Enter a private IP address	80

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

Figure 3-9 Confirming advanced settings



* Proxy Configured: Layer-7 proxy

- Layer-7 proxy: Web proxy products for layer-7 request forwarding are used, products such as anti-DDoS, CDN, and other cloud acceleration services.
- Layer-4 proxy: Web proxy products for layer-4 forwarding are used, products such as anti-DDoS.
- No proxy: No proxy products are used.
Note: If a layer-7 proxy is configured, WAF reads the real client IP address from the related fields in the header. View Details

* Policy: System-generated policy

WAF generates security policies only for General Check in Basic Web Protection (Log only), and the Scanner check in Anti-Crawler protection. Only WAF professional edition and above can generate scanner check policies. Custom Policies

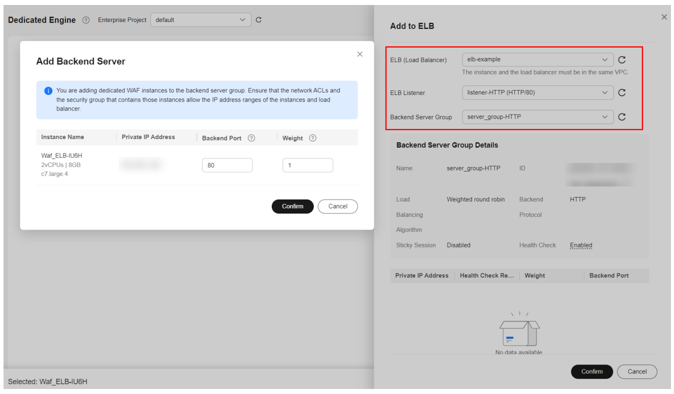
Associating WAF with Your Load Balancer

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



Add Backend Server

You are adding dedicated WAF instances to the backend server group. Ensure that the network ACLs and the security group that contains those instances allow the IP address ranges of the instances and load balancer.

Instance Name	Private IP Address	Backend Port	Weight
Waf_ELB-IU6H 2vCPUs | 8GB c7.large.4	[redacted]	80	1

Add to ELB

ELB (Load Balancer): elb-example
The instance and the load balancer must be in the same VPC.
ELB Listener: listener-HTTP (HTTP/80)
Backend Server Group: server_group-HTTP

Backend Server Group Details

Name	server_group-HTTP	ID	[redacted]
Load Balancing Algorithm	Weighted round robin	Backend Protocol	HTTP
Sticky Session	Disabled	Health Check	Enabled

Private IP Address	Health Check Re...	Weight	Backend Port

No data available

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

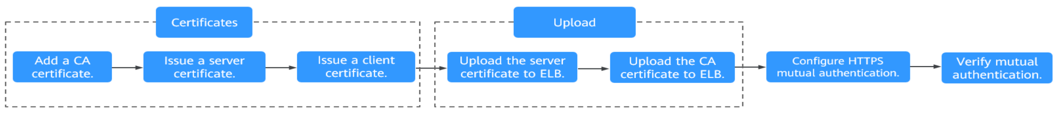
Scenarios

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


Certificates
Add a CA certificate.
Issue a server certificate.
Issue a client certificate.
Upload
Upload the server certificate to ELB.
Upload the CA certificate to ELB.
Configure HTTPS mutual authentication.
Verify mutual authentication.

Step 1: Add a CA Certificate Using OpenSSL

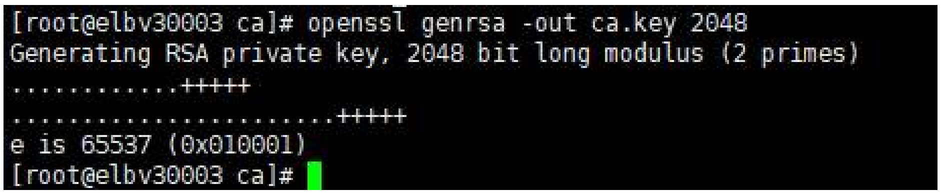
1. Log in to a Linux server with OpenSSL installed.
2. Create the **server** directory and switch to the directory:

   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]#
```

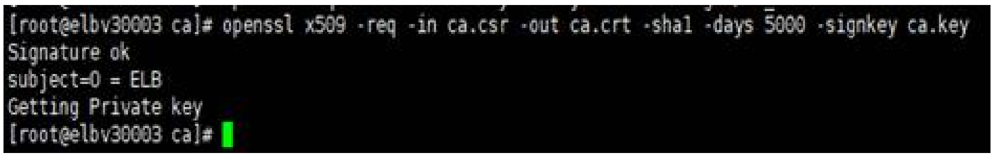
5. Create the certificate signing request (CSR) file **ca.csr** for the CA certificate.

   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

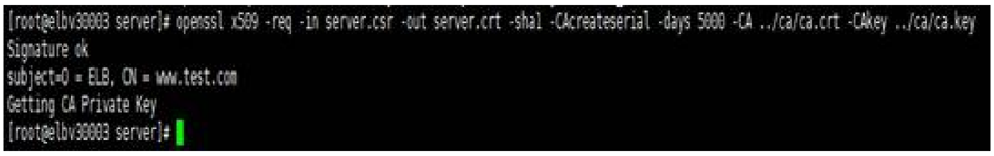
5. Create the CSR file **server.csr** for the server certificate.

   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:

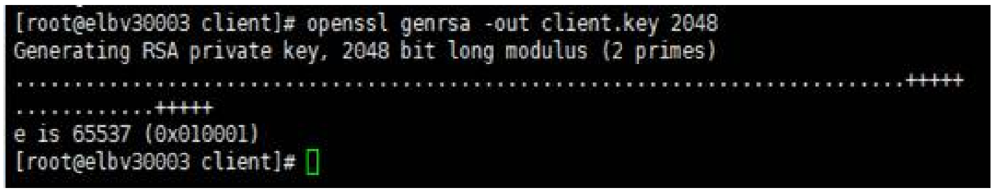
```
[ req ]
distinguished_name     = req_distinguished_name
prompt                 = no

[ 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

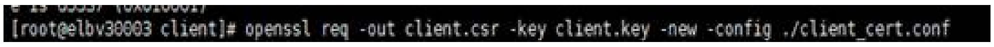


```
[root@elbv30003 client]# openssl genrsa -out client.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.........................................................................+++++
............+++++
e is 65537 (0x010001)
[root@elbv30003 client]#
```

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

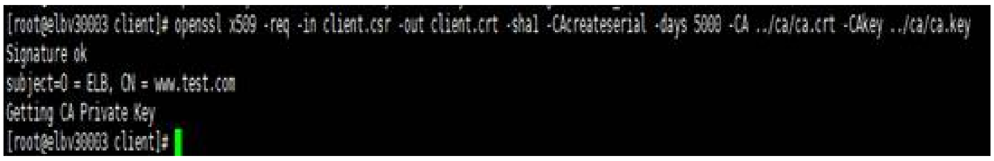


```
[root@elbv30003 client]# openssl req -out client.csr -key client.key -new -config ./client_cert.conf
```

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.

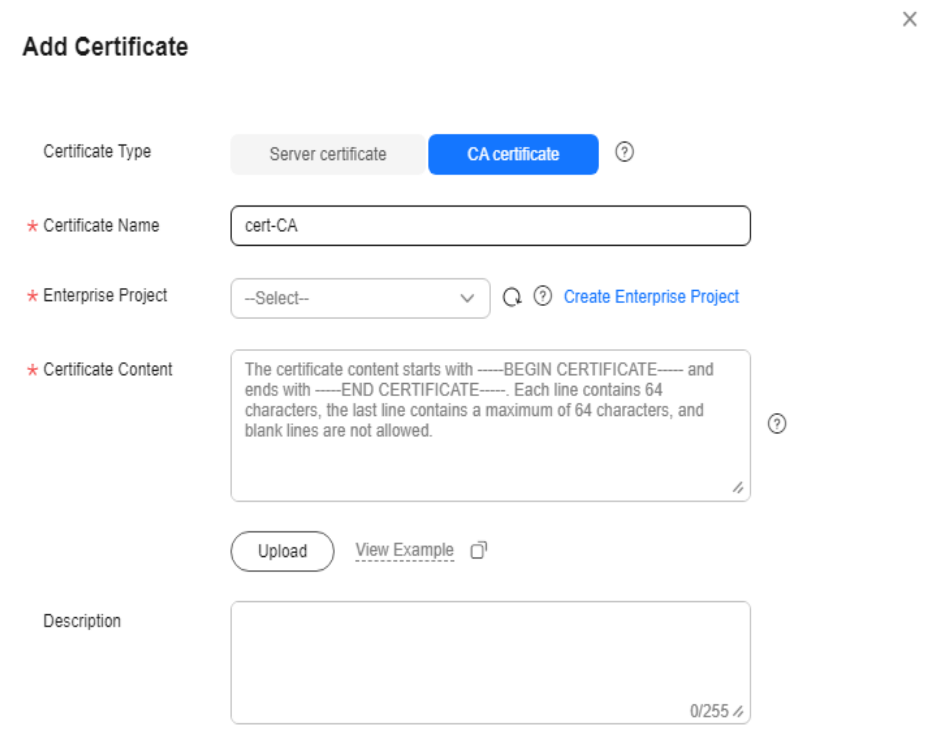
Step 2 In the navigation pane on the left, choose **Certificates**.

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

Certificate Type: Server certificate / CA certificate

* Certificate Name: cert-CA

* Enterprise Project: --Select-- 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.

Upload View Example

Description: 0/255

NOTE

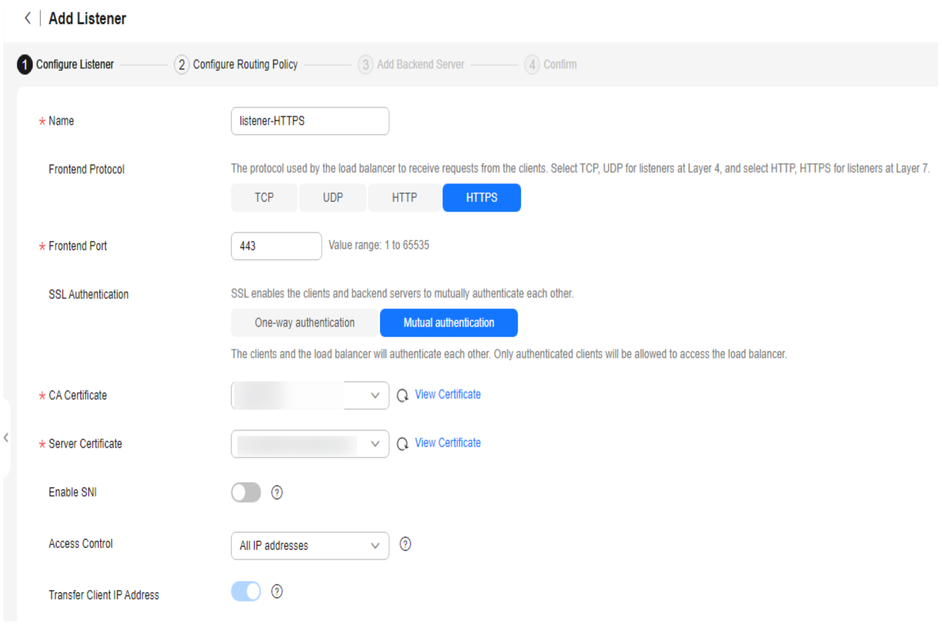
The certificate must be PEM-encoded.

----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

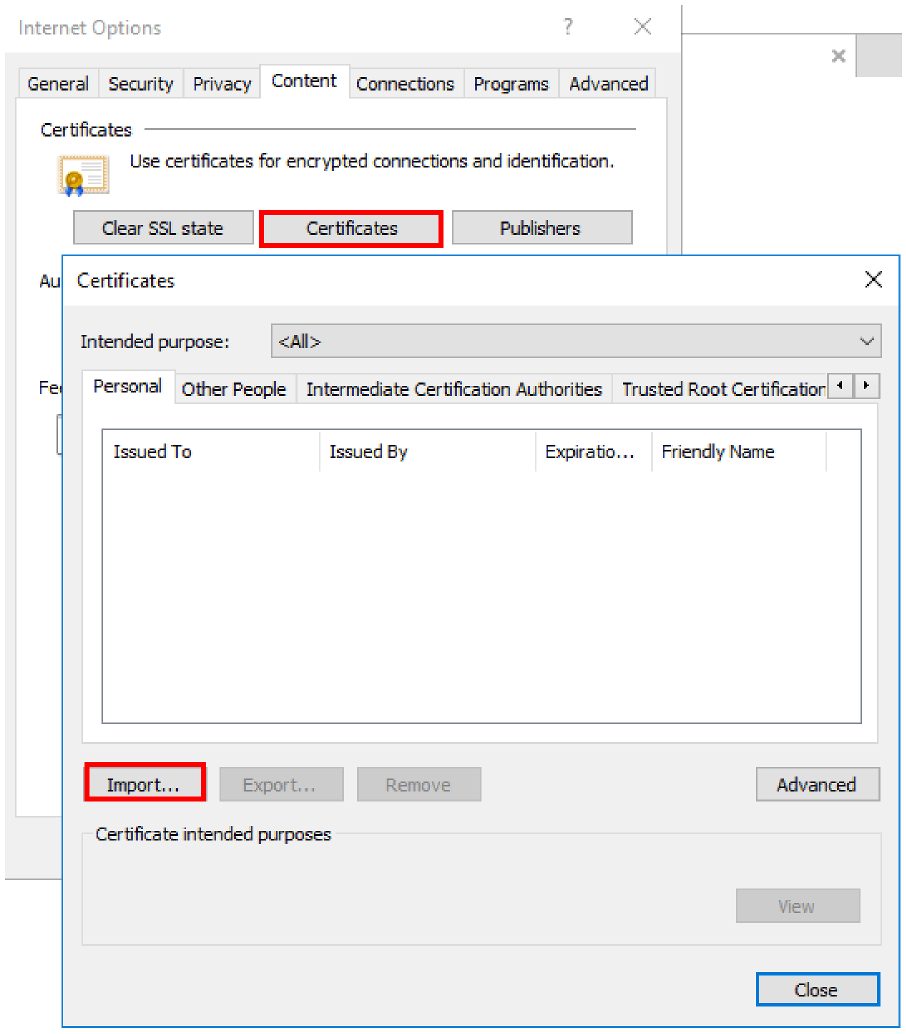
* Name	listener-HTTPS
Frontend Protocol	The protocol used by the load balancer to receive requests from the clients. Select TCP, UDP for listeners at Layer 4, and select HTTP, HTTPS for listeners at Layer 7. TCP UDP HTTP [✓ HTTPS]
* Frontend Port	443 Value range: 1 to 65535
SSL Authentication	SSL enables 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	All IP addresses
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.

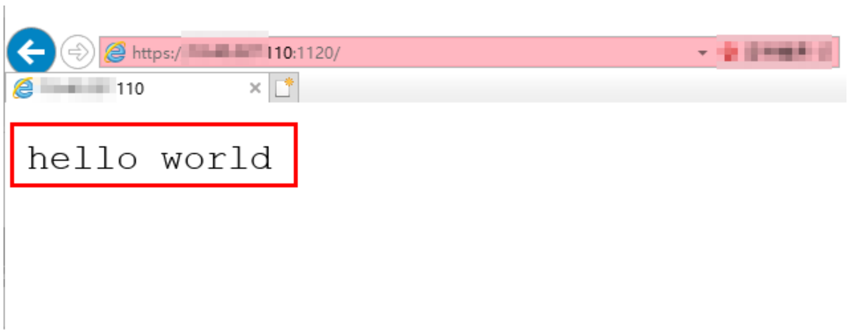
Figure 4-10 Importing the **client.p12** certificate



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

```
hello world
```

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**.

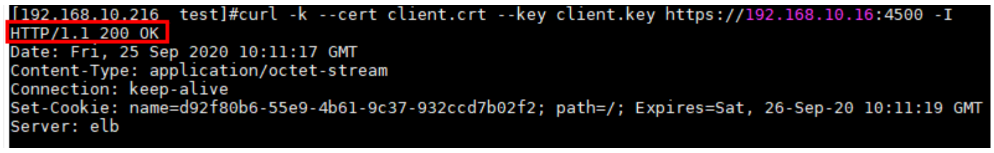
2. Verify the import.

   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

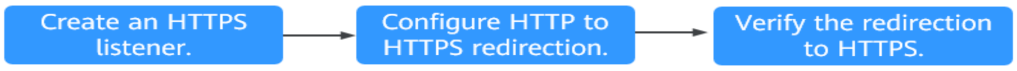
- If the listener protocol is HTTP, only the GET or HEAD method can be used for redirection. If you create a redirect for an HTTP listener, the client browser will change POST or other methods to GET. If you want to use other methods rather than GET and HEAD, add an HTTPS listener.
- HTTP requests are 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

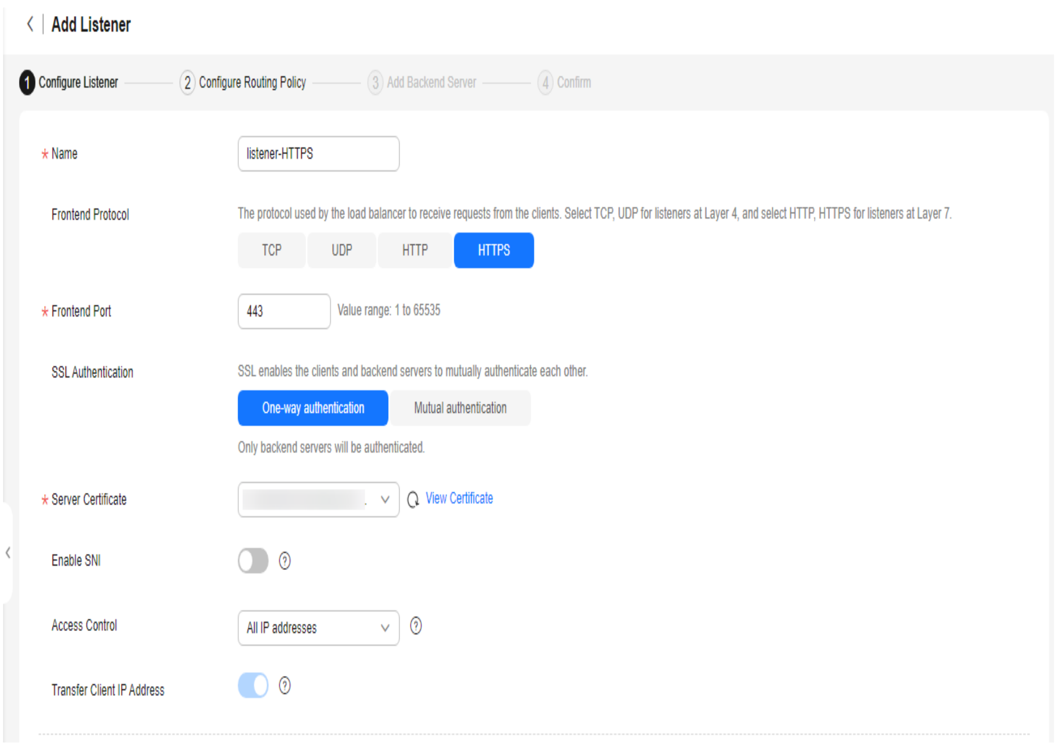
Figure 5-1 Procedure for redirecting HTTPS requests to an HTTPS listener


Create an HTTPS listener.
Configure HTTP to HTTPS redirection.
Verify the redirection to HTTPS.

Step 1: Create an HTTPS Listener

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 5-1**.

Figure 5-2 Adding an HTTPS listener



Add Listener

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

* Name	listener-HTTPS
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	443 Value range: 1 to 65535
SSL Authentication	SSL enables the clients and backend servers to mutually authenticate each other. [✓ One-way authentication] Mutual authentication. Only backend servers will be authenticated.
* Server Certificate	View Certificate
Enable SNI	
Access Control	All IP addresses
Transfer Client IP Address	

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.
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.

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**.

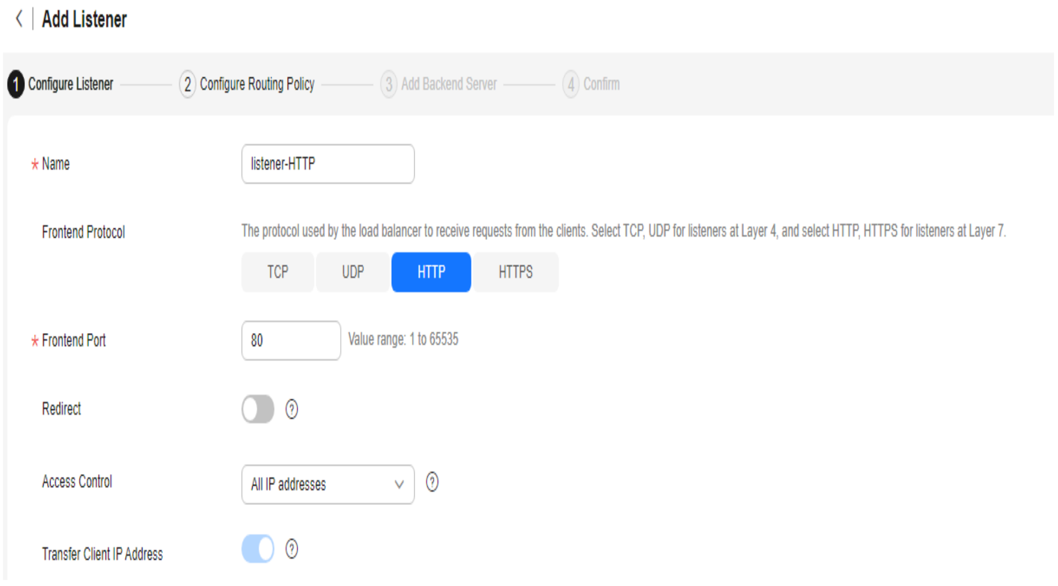
Step 2: Configure HTTP to HTTPS Redirection

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: listener-HTTP

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 (HTTP selected)

* Frontend Port: 80 Value range: 1 to 65535

Redirect: off

Access Control: All IP addresses

Transfer Client IP Address: on

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.

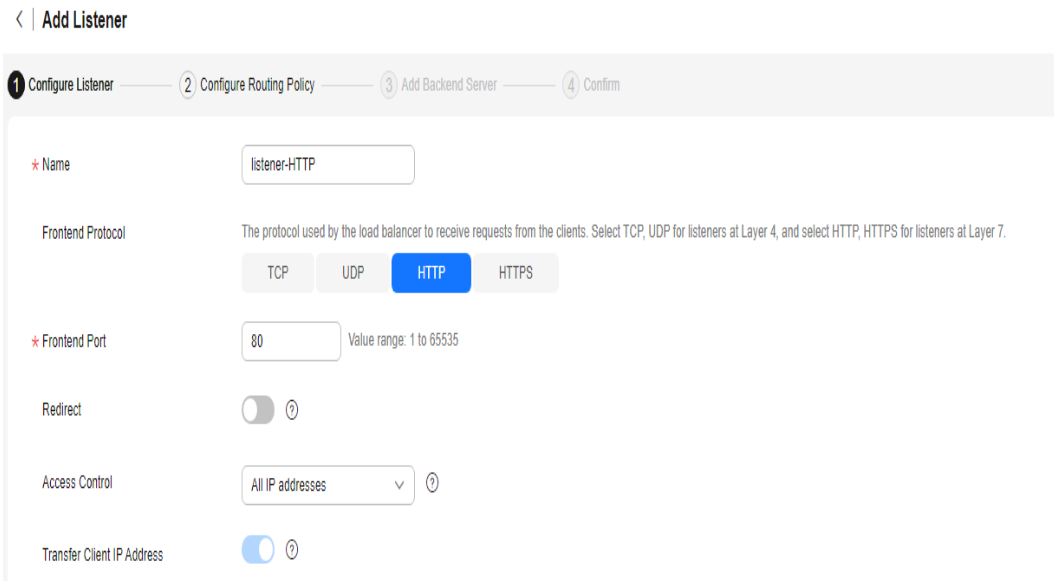
Parameter	Example Value	Description
Frontend Port	80	Specifies the port that will be used by the load balancer to receive requests from clients.
Redirect	Enabled	Specifies whether to enable redirection. You can use this function to redirect the requests from an HTTP listener to an HTTPS listener to ensure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

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 (HTTP selected)

* Frontend Port: 80 Value range: 1 to 65535

Redirect: off

Access Control: All IP addresses

Transfer Client IP Address: on

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.

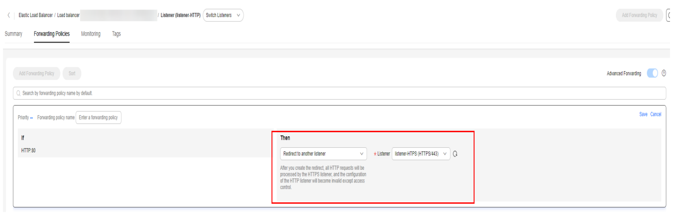
6. Retain the default values for parameters under **Advanced Settings** and click **Next: Configure Request Routing Policy**.
7. Select **Create new** for **Backend Server Group**, retain the default values for other parameters, and click **Next: Add Backend Server**.
8. Add **ECS_server** to the backend 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



HTTP:80

Redirect to another listener

* Listener: listener-HTPS (HTTPS/443)

After you create the redirect, all HTTP requests will be processed by the HTTPS listener, and the configuration of the HTTP listener will become invalid except access control.

NOTE

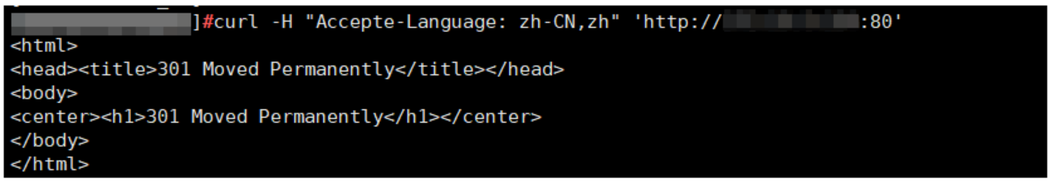
- After the redirection is added, the configurations for the HTTP listener will not be applied, but access control configured for that listener will still be applied.
- After the redirection is added for an HTTP listener, the backend server will return 301 Moved Permanently to the clients.

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



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