

Elastic Load Balance

Best Practices

Issue 01
Date 2025-09-29



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2025. 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 Load Balancer Selection and Service Planning..... 1

1.1 Adding the IP Address of an On-Premises Server as a Backend Server of a Dedicated Load Balancer
..... 1

1.2 Using IP as a Backend to Route Traffic Across Backend Servers..... 5

1.2.1 Overview..... 5

1.2.2 Routing Traffic to Backend Servers in a Different VPC from the Load Balancer..... 7

1.2.3 Routing Traffic to Backend Servers in the Same VPC as the Load Balancer..... 12

2 Forwarding Policies..... 18

2.1 Using Advanced Forwarding for Application Iteration..... 18

1 Load Balancer Selection and Service Planning

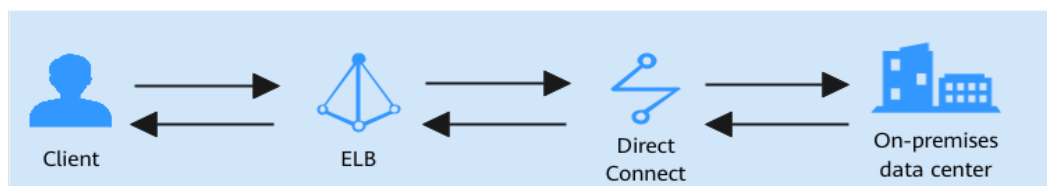
1.1 Adding the IP Address of an On-Premises Server as a Backend Server of a Dedicated Load Balancer

Scenarios

You can use Direct Connect or VPN to connect on-premises servers to Huawei Cloud, and then add the IP addresses of on-premises servers as backend servers of a dedicated load balancer. In this way, the load balancer can distribute traffic to these servers.

Solution Architecture

Figure 1-1 Distributing traffic to an on-premises server



In this practice, a Direct Connect connection is required to connect on-premises servers to a VPC and a dedicated load balancer is required to route requests to on-premises servers.

- Dedicated load balancer **ELB-Test** is running in a VPC named **VPC-Test-01** (172.16.0.0/12).
- An on-premises server (**IDC-IP-Test**) is deployed in the on-premises data center.
- The IP address of **IDC-IP-Test** is added to the backend server group of **ELB-Test**.

Resource Planning

In this practice, you need to create a VPC, dedicated load balancer, EIP, Direct Connect connection, ECS, and on-premises server. For details about the resource planning, see [Table 1-1](#).

This practice uses public network access as an example. You can also use a load balancer to route client requests over a private network.

Table 1-1 Resource planning

Resource Type	Quantity	Description
VPC and subnet	One VPC and two subnets	• VPC-Test-01: The CIDR block is 172.16.0.0/12. • subnet-frontend: The CIDR block is 172.16.0.0/16. • subnet-backend: The CIDR block is 172.18.0.0/16.
Load balancer	1	The load balancer that is used to distribute client requests. • VPC: VPC-Test-01 • Frontend subnet: subnet-frontend • Backend subnet: subnet-backend
Direct Connect connection	1	The connection that is used to connect the on-premises data center to a VPC.
ECS	1	ECS-client , which is used to access the on-premises server.
EIP	2	• One EIP will be bound to the load balancer to distribute client requests over the public network. • The other EIP will be bound to ECS-client to access the load balancer over the public network.
On-premises server	1	IDC-IP-Test , which is deployed in the on-premises data center.

Preparations

- Purchase an ECS (**ECS-client**) as the client and bind an EIP to it. For details about how to purchase an ECS, see .
- Create a VPC (**VPC-Test-01**) with two subnets (**subnet-frontend** and **subnet-backend**). For details, see .
- Create a dedicated load balancer, enable **IP as a Backend**, and bind an EIP to the load balancer. For details, see and .

Step 1: Connect an On-Premises Data Center to a VPC Using Direct Connect

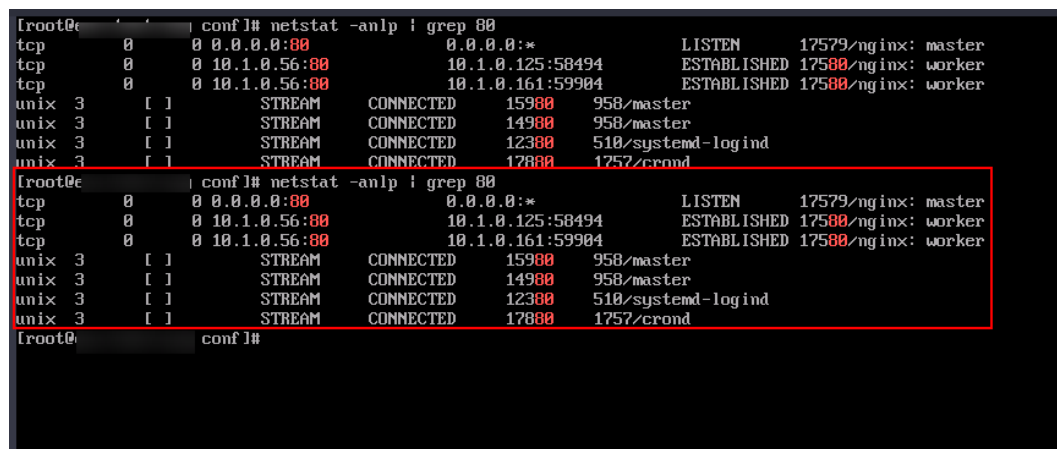
You can create a Direct Connect connection on the console to connect your on-premises data center or on-premises private network to a VPC.

For details, see .

Step 2: Deploy a Service on the On-Premises Server

Deploy Nginx on the on-premises server **IDC-IP-Test** to verify network connectivity.

Figure 1-2 Nginx deployed on the on-premises server



```
[root@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
```

Step 3: Create a Backend Server Group and Add the IP Address of the On-Premises Server as a Backend Server

1. Go to the [backend server group list page](#).
2. Click **Create Backend Server Group** in the upper right corner.
3. Configure the parameters based on [Table 1-2](#). Retain the default values for other parameters.

Table 1-2 Parameters required for configuring a backend server group

Parameter	Example Value	Description
Name	server_group	Specifies the name of the backend server group.
Type	Dedicated	Specifies the type of the load balancer that can use the backend server group.
Backend Protocol	HTTP	Specifies the protocol that backend servers in the backend server group use to receive requests from the listeners. Select HTTP .

Parameter	Example Value	Description
Load Balancing Algorithm	Weighted round robin	Specifies the load balancing algorithm used by the load balancer to distribute traffic. Weighted round robin: Requests are routed to different servers based on their weights. Backend servers with higher weights receive proportionately more requests, whereas equal-weighted servers receive the same number of requests. For more information, see .

- Click **Next** to add backend servers and configure health check.
- Switch to the **IP as Backend Servers** tab and click **Add IP as Backend Server**.
- On the displayed page, add the IP address of the on-premises server as a backend server and set the parameters as follows:
 - IP Address:** Set it to **10.1.0.56** in this example, which is the private IP address of the on-premises server **IDC-IP-Test**.
 - Backend Port:** Set this parameter based on service requirements. In this practice, port 80 is used.
 - Weight:** Retain the default value 1.
- Click **OK**.
- Enable health check and retain the default values for other health check parameters.
- Click **Next**.
- Confirm the configurations and click **Create Now**.

Step 4: Add a Listener and Select the Created Backend Server Group

- Go to the [load balancer list page](#).
- Locate the target load balancer and click **Add Listener** in the **Operation** column.
- On the **Add Listener** page, set **Frontend Protocol** to **HTTP** and **Frontend Port** to **80**. Retain the default values for other parameters.
- Click **Next: Configure Request Routing Policy** and select **Use existing** for **Backend Server Group**.
Select the backend server group created in [Step 3](#) and click **Next: Confirm**.
- Confirm the configurations and click **Submit**.

Step 5: Verify Request Routing to the On-Premises Server

- Use the client to access the on-premises server to verify network connectivity.
 - Remotely log in to **ECS_client**.
Multiple methods are available for logging in to an ECS. For details, see [Logging In to an ECS](#).

- b. Run the **curl -v http://<IP-address>:<port>** command to test network connectivity.

In this practice, run the following command:

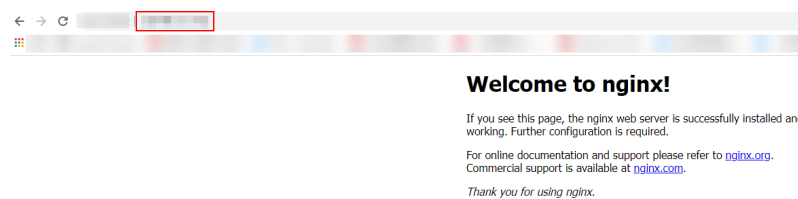
```
curl -v http://<EIP-of-the-load-balancer>:<Listening-port>
```

If the default Nginx welcome page is displayed, ELB successfully forwards the client request to the on-premises server.

2. Use a browser to verify the network connectivity.

Use a browser to access **http://<EIP-of-the-load-balancer>**. If the following page is displayed, the request is forwarded by the load balancer to the on-premises server.

Figure 1-3 The Nginx default welcome page when accessed using a browser



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

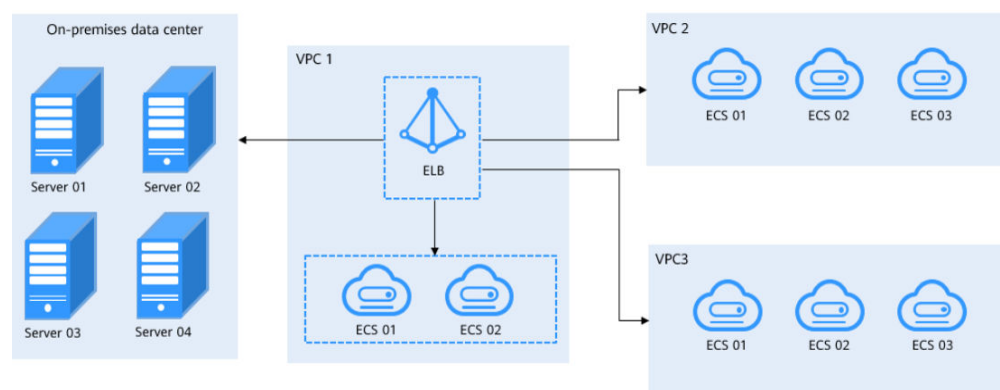
1.2.1 Overview

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 1-4 Routing traffic across cloud and on-premises servers



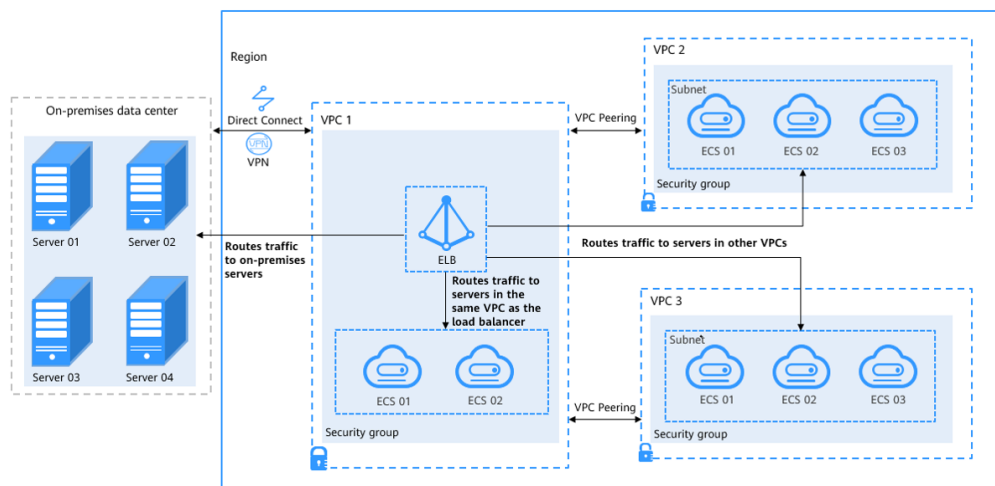
Solution

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 1-5](#), 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 1-5 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.

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

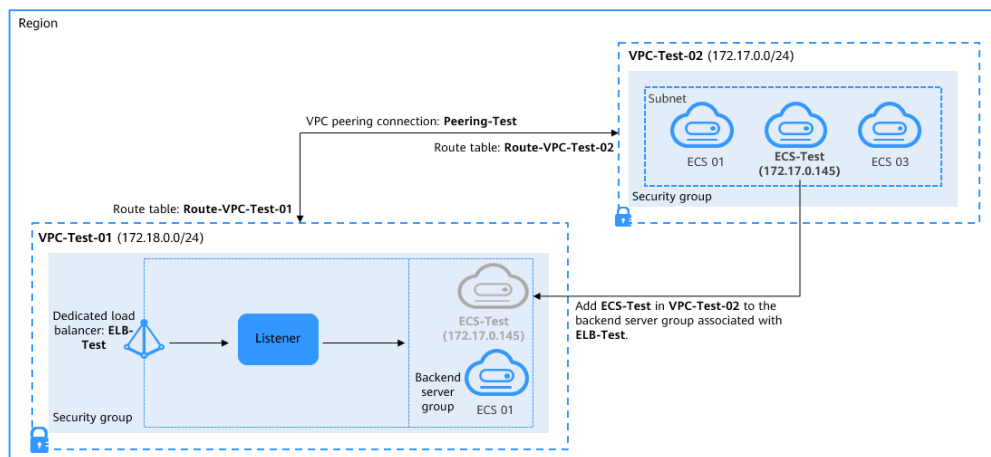
Scenarios

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 1-6 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

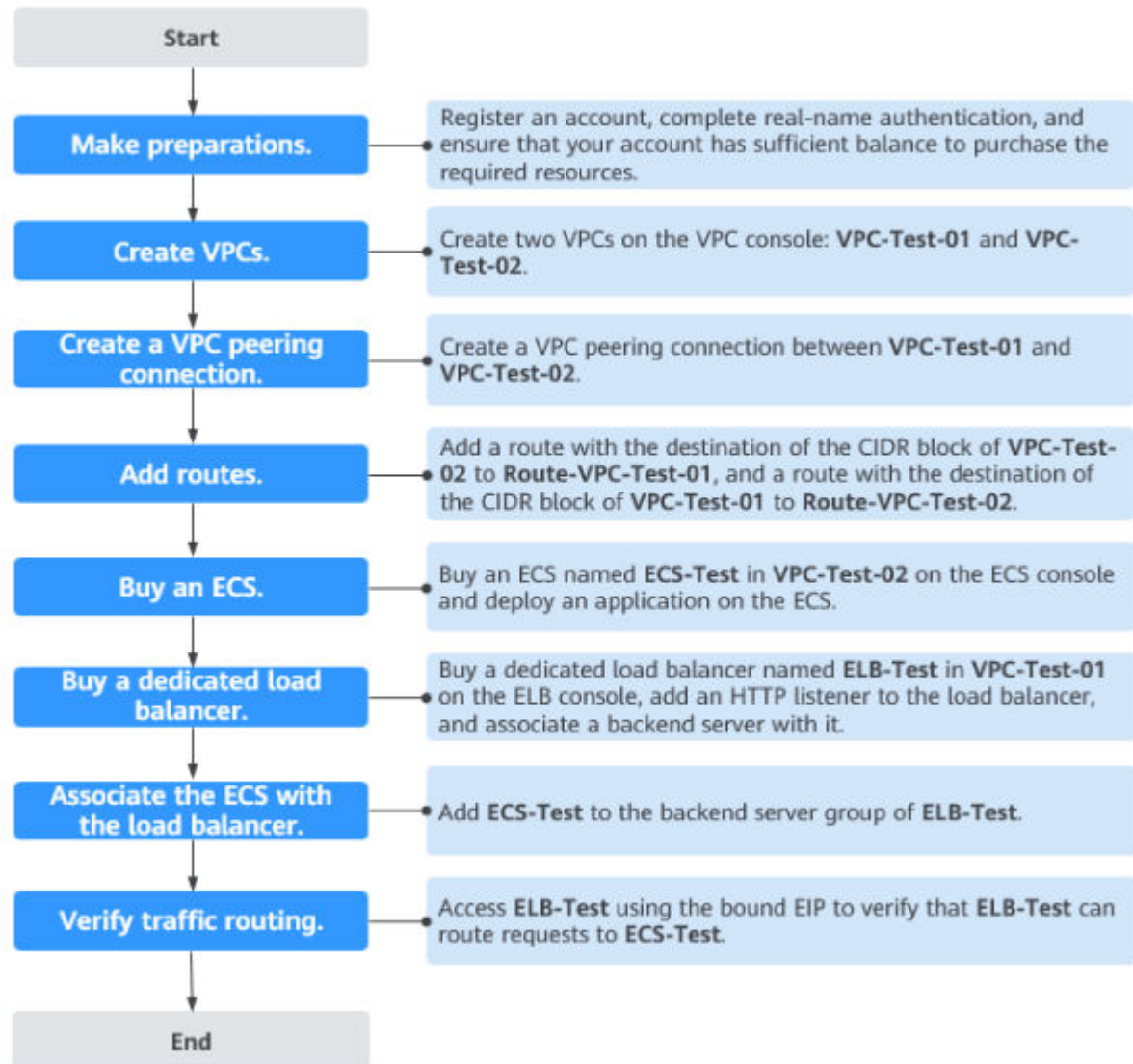
The actual cost shown on the Huawei Cloud console is used.

Table 1-3 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 1-7 Process of associating servers in a VPC that is different from the dedicated load balancer



Step 1: Create VPCs

1. Log in to the [VPC console](#).
2. Configure the parameters as described in [Table 1-3](#) 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.
3. Create the other VPC.
 - **Name:** VPC-Test-02
 - **IPv4 CIDR Block:** 172.17.0.0/24

- Configure other parameters as required.

Figure 1-8 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

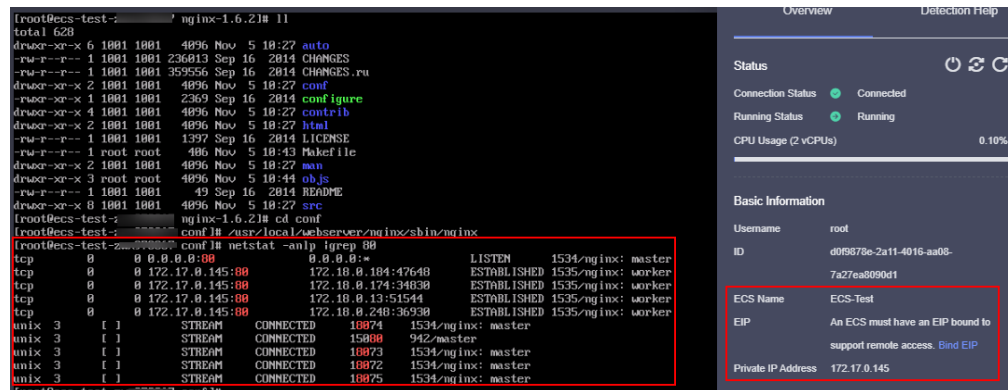
1. In the navigation pane on the left, click **Route Tables**.
2. In the upper right corner, click **Create Route Table**.
3. Configure the parameters as described in [Table 1-3](#) and click **OK**. For details on how to create a route table, see the [Virtual Private Cloud User Guide](#).
 - **Name:** 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 [Purchasing an ECS](#).

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

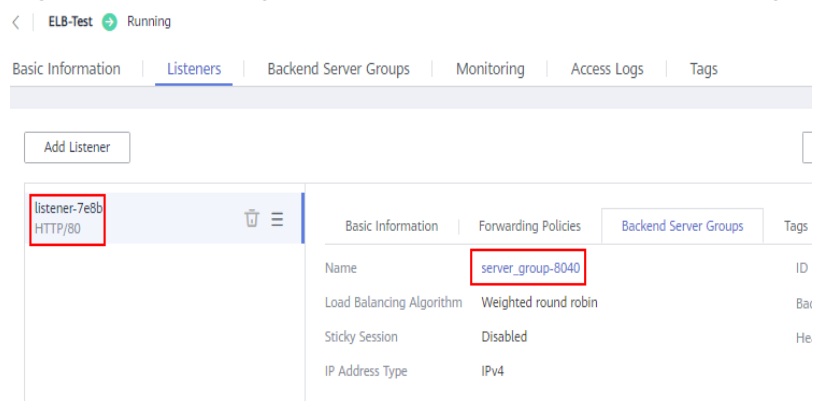
Figure 1-9 Deploying Nginx on ECS-Test



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

- On the management console, choose **Network > Elastic Load Balance**.
- In the upper right corner, click **Buy Elastic Load Balancer**.
- 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.
- Add an HTTP listener to **ELB-Test** and associate a backend server group with it.

Figure 1-10 Viewing the HTTP listener and backend server group



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

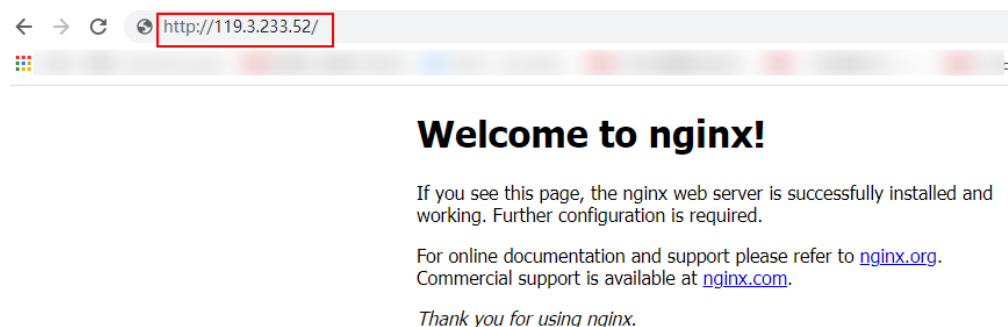
- Locate **ELB-Test** and click its name.
- On the **Listeners** tab, locate the HTTP listener added to **ELB-Test** and click its name.

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 1-11 Verifying that the request is routed to **ECS-Test**



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

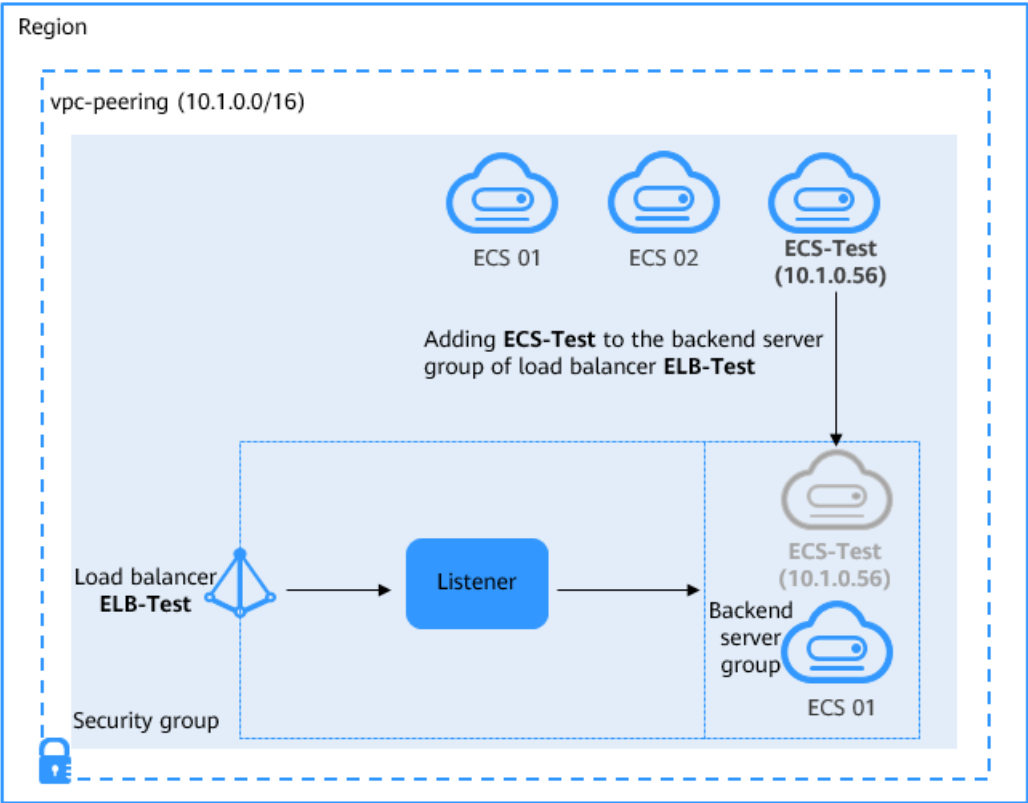
Scenarios

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** is added to the backend server group associated with **ELB-Test** in **vpc-peering**.

Figure 1-12 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.

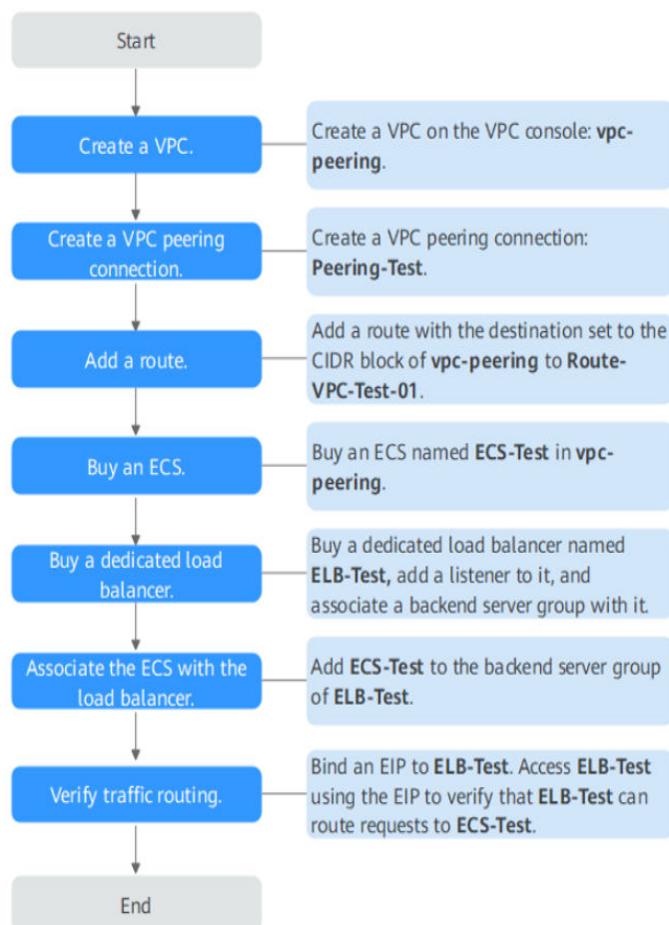
Table 1-4 Resource planning

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
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

Resource Type	Resource Name	Description	Quantity
Route table	Route-VPC-Test-01	The route table of vpc-peering . 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 1-13 Process for adding backend servers in the same VPC as the load balancer



Step 1: Create a VPC

1. Log in to the [VPC console](#).
2. 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.

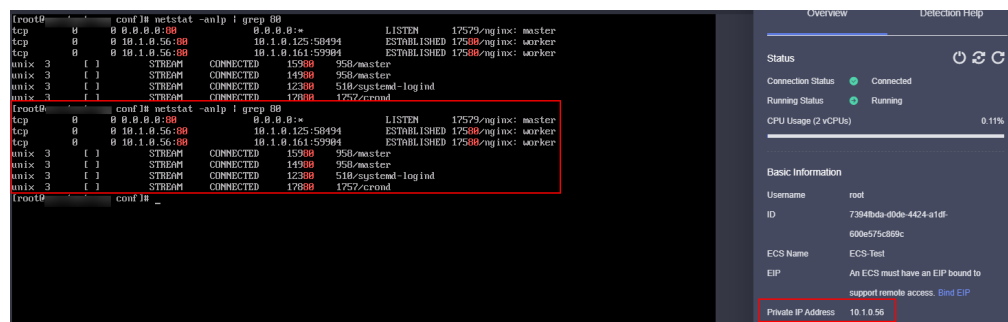
Step 3: Add Routes for Peering-Test

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 1-14 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 **Network > 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

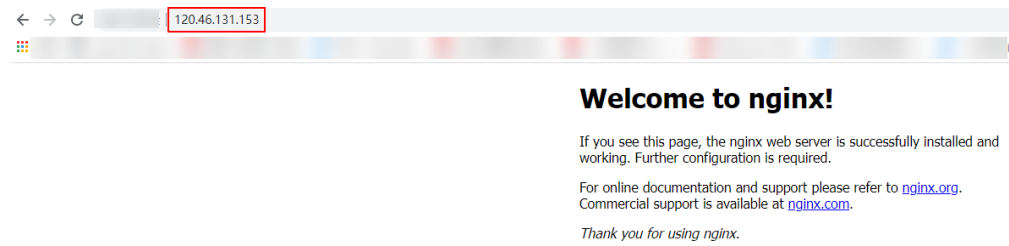
Locate **ELB-Test** and click its name.

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 1-15 Verifying that the request is routed to **ECS-Test**



2 Forwarding Policies

2.1 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

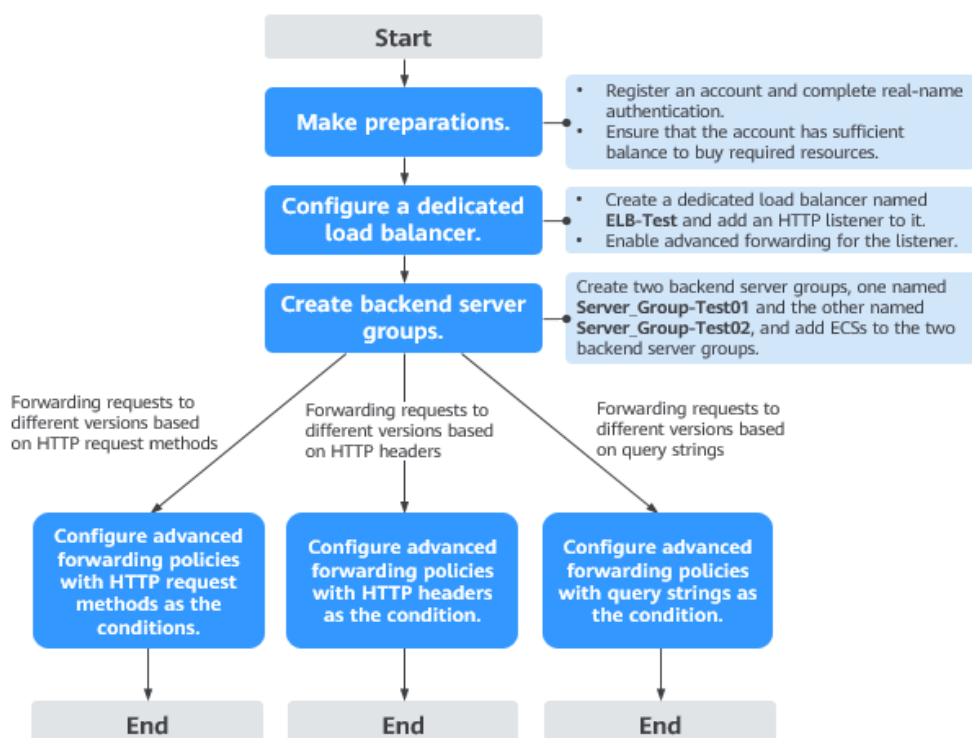


Table 2-1 Resource planning

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 .

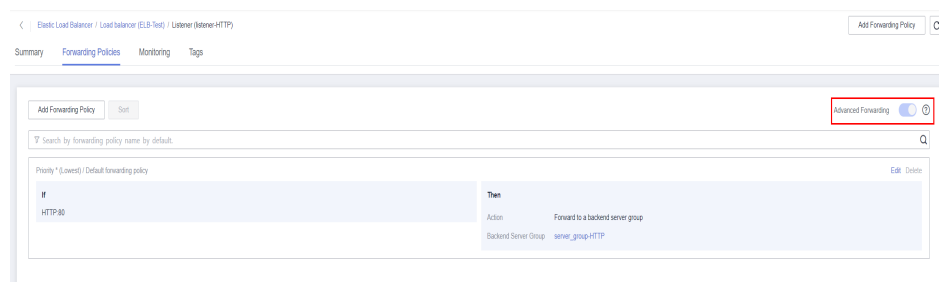
 NOTE

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 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



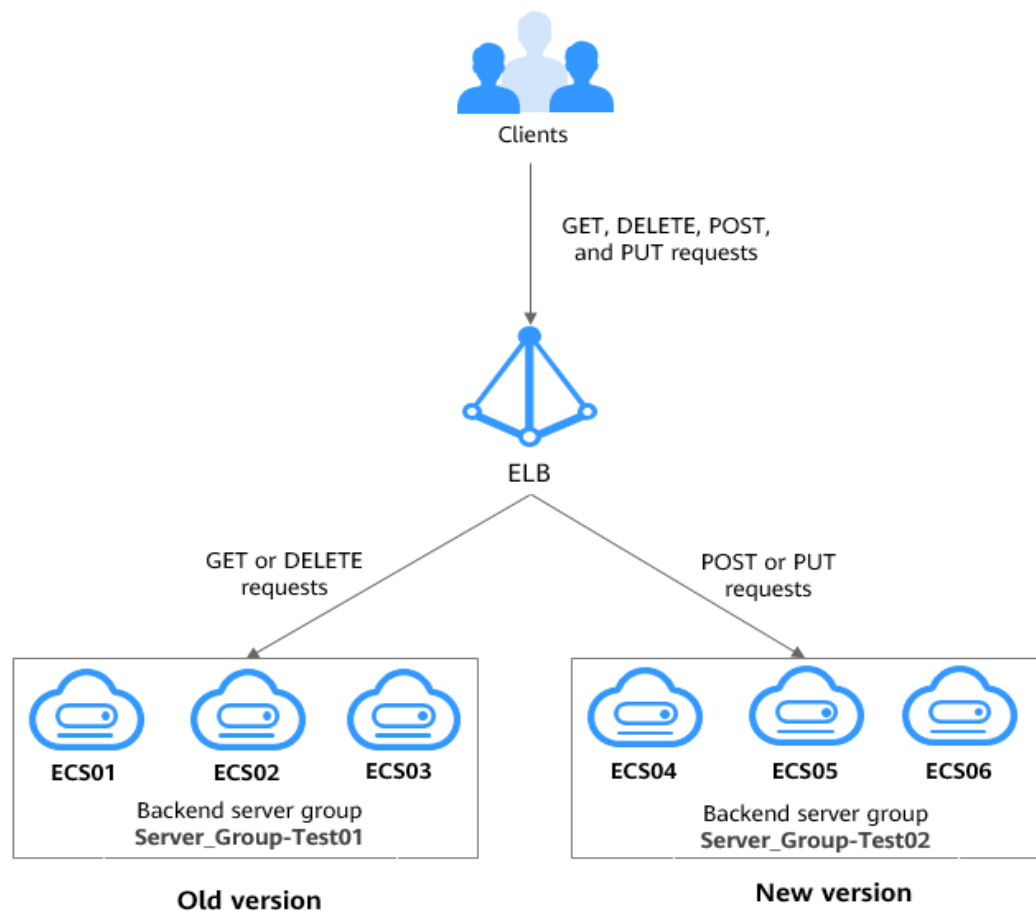
Step 2: Create Two Backend Server Groups and Add Backend Servers to Them

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

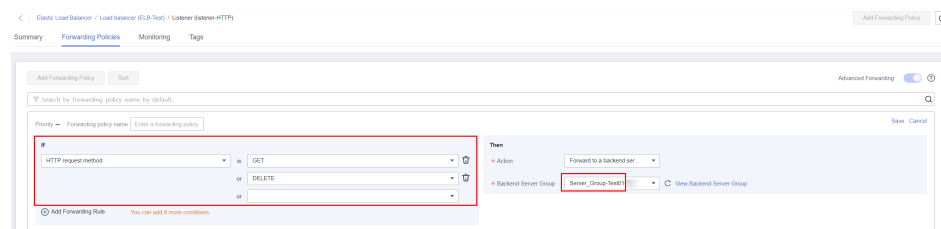
Configure two advanced forwarding policies with the HTTP request method as the condition to route GET and DELETE requests to the application of the old version and POST and PUT requests to the application of the new version. When the application of the new version runs stably, direct all the requests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

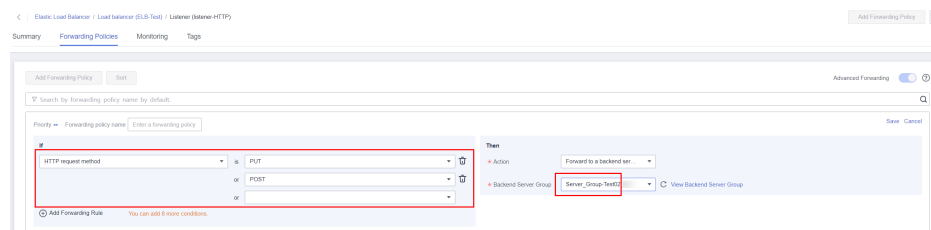
Figure 2-4 Forwarding GET and DELETE requests to the application of the old version



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.

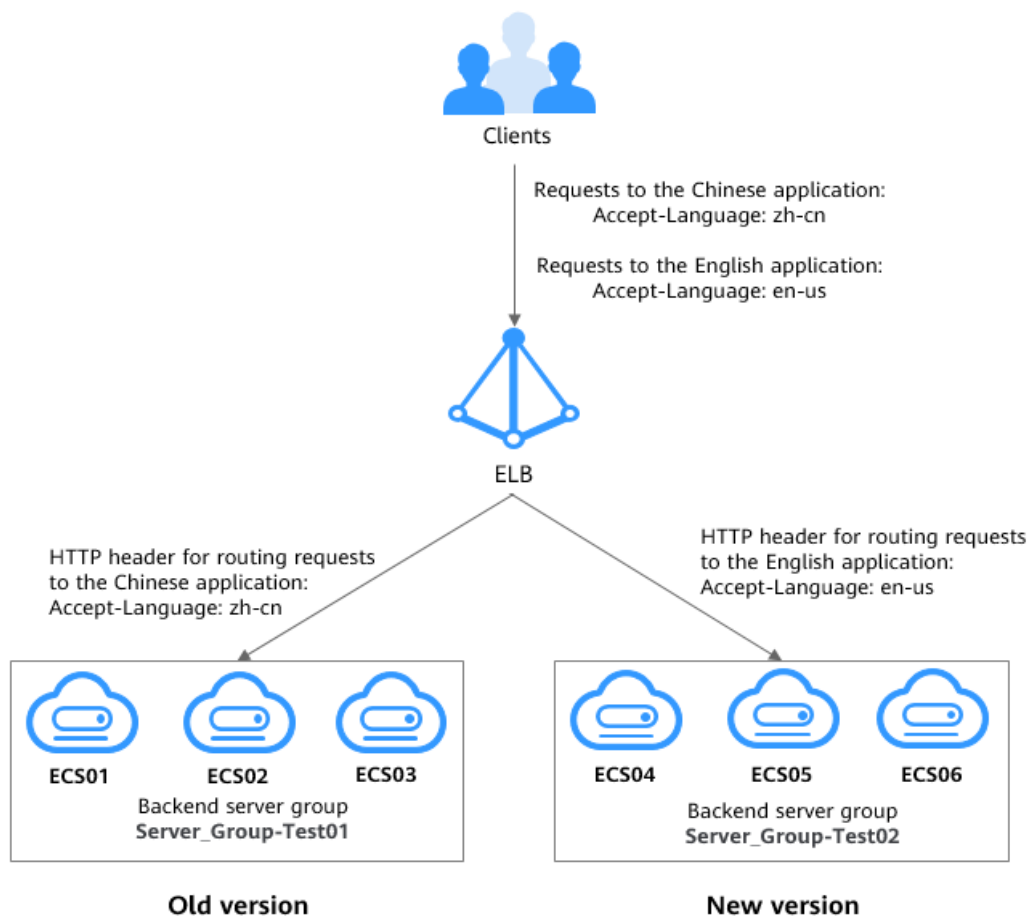
Figure 2-5 Forwarding PUT and POST requests to the application of the new version



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

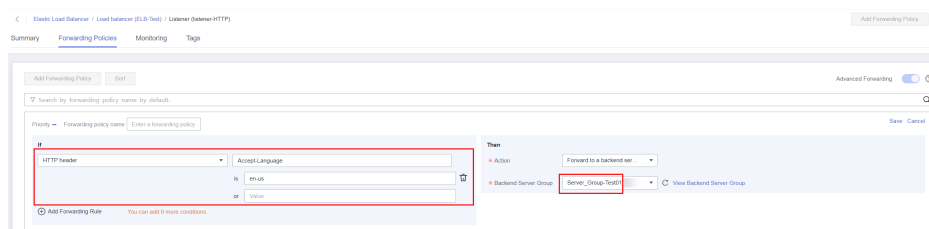
If the old version supports both Chinese and English, but the new version only supports English because the Chinese version is still under development, you can configure two advanced forwarding policies with the HTTP header as the condition to route requests to the Chinese application to the old version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

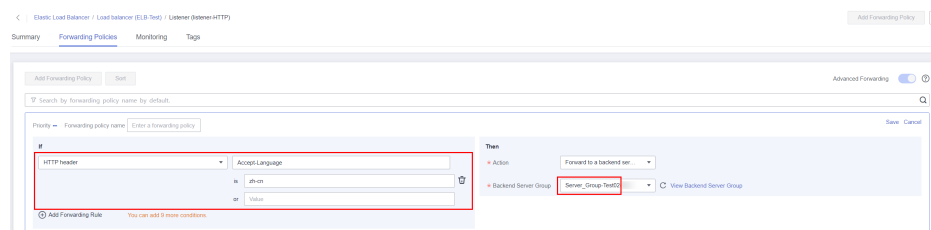
Figure 2-7 Forwarding requests to the application of the old version



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.

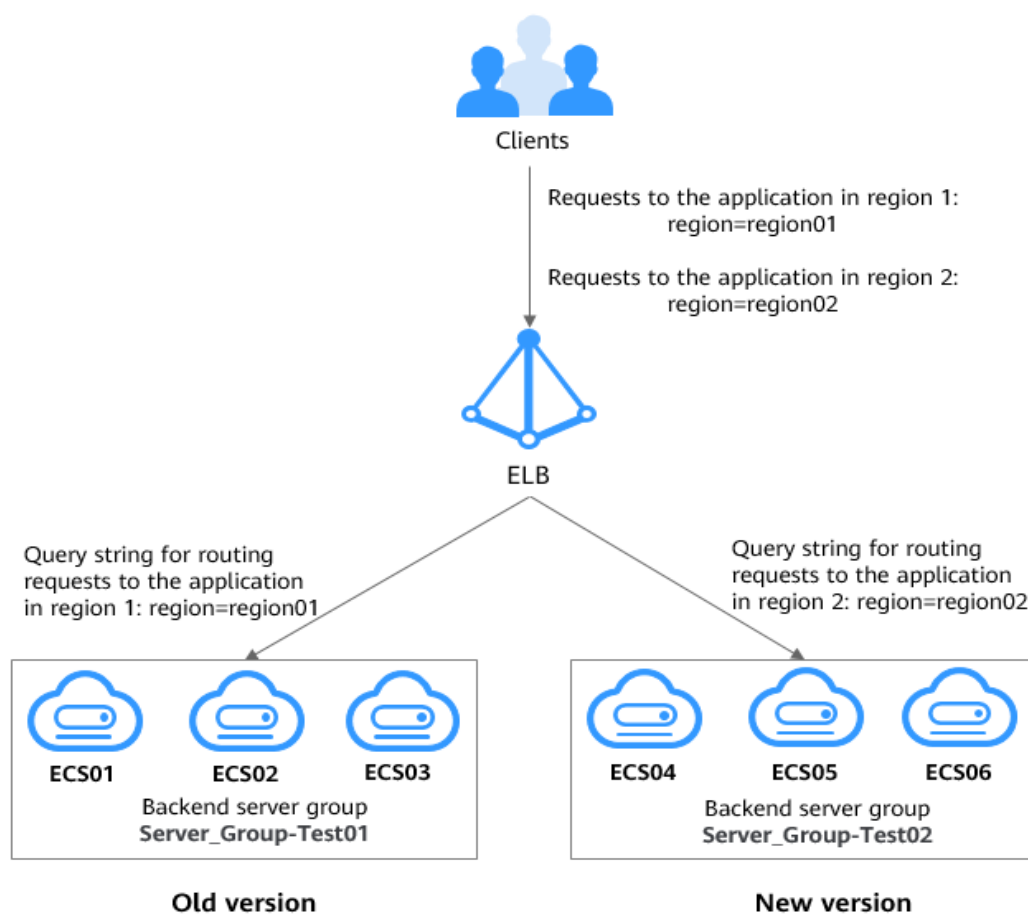
Figure 2-8 Forwarding requests to the application of the new version



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



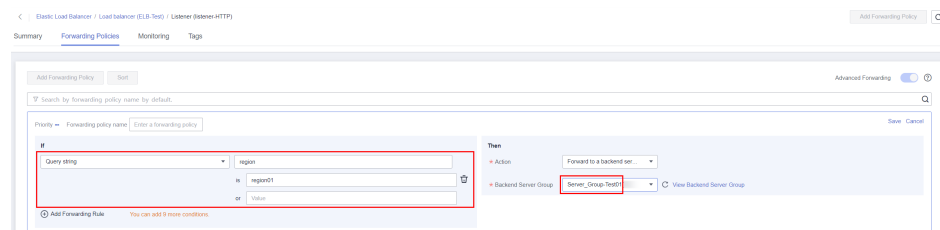
 NOTE

- 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

