

# Elastic Load Balance

## Best Practice

**Issue** 01  
**Date** 2022-03-30



**Copyright © Huawei Technologies Co., Ltd. 2023. 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 Technologies Co., Ltd.

## **Trademarks and Permissions**



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

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

## **Notice**

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

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

---

# Contents

---

<b>1 Locating an Unhealthy Backend Server Using Access Logs.....</b>	<b>1</b>
<b>2 Viewing Traffic Usage.....</b>	<b>5</b>
<b>3 Routing Traffic to Backend Servers in Different VPCs.....</b>	<b>8</b>
3.1 Overview.....	8
3.2 Routing Traffic to Backend Servers in Different VPCs from the Load Balancer.....	10
3.3 Routing Traffic to Backend Servers in the Same VPC as the Load Balancer.....	19
<b>4 Using Advanced Forwarding for Application Iteration.....</b>	<b>27</b>

# 1 Locating an Unhealthy Backend Server Using Access Logs

---



## Scenarios

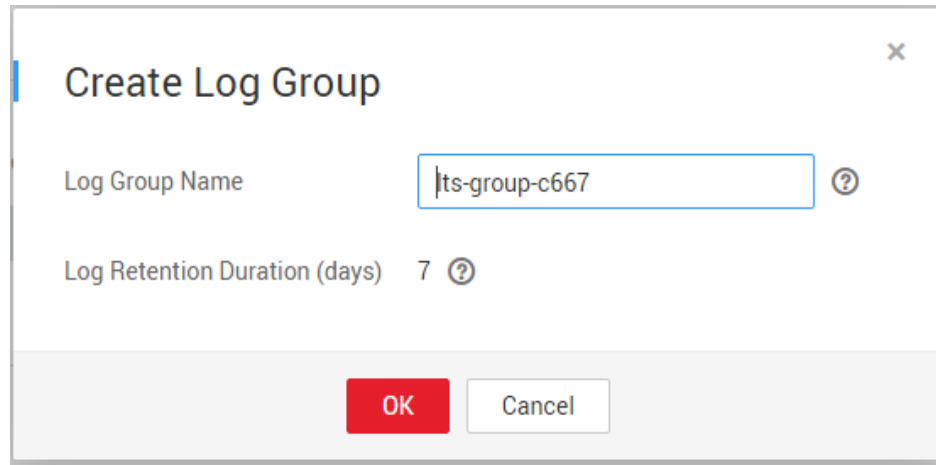
With LTS, you can view logs of requests to shared load balancers at Layer 7 and analyze response status codes to quickly locate unhealthy backend servers.

## Preparations

- You have created a shared load balancer that can work at Layer 7.
- You have enabled LTS.

## Creating a Log 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. Hover on  in the upper left corner to display **Service List** and choose **Management & Governance > Log Tank Service**.
4. In the navigation pane on the left, choose **Log Management**.
5. On the displayed page, click **Create Log Group**. In the displayed dialog box, enter a name for the log group.

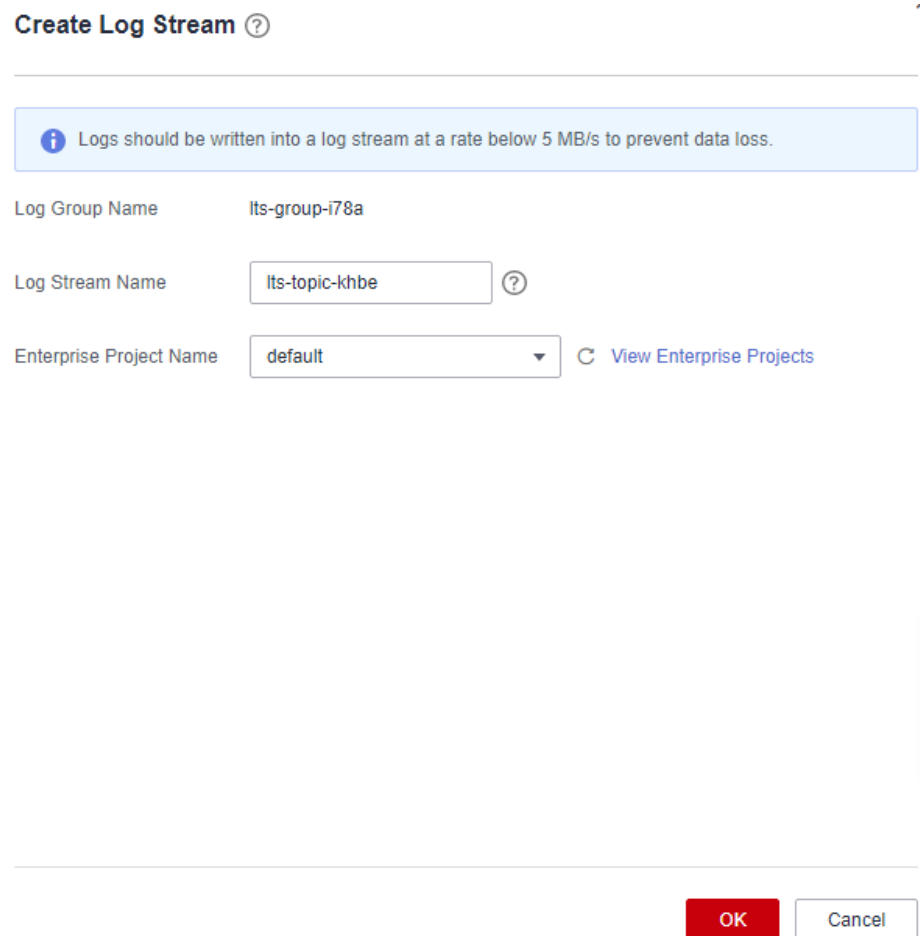


6. Click **OK**.

## Creating a Log Stream

1. Locate the created log group and click its name.
2. Click **Create Log Stream**. In the displayed dialog box, enter a name for the log stream.

**Figure 1-1** Creating a log stream



3. Click **OK**.

## Configuring Access Logging

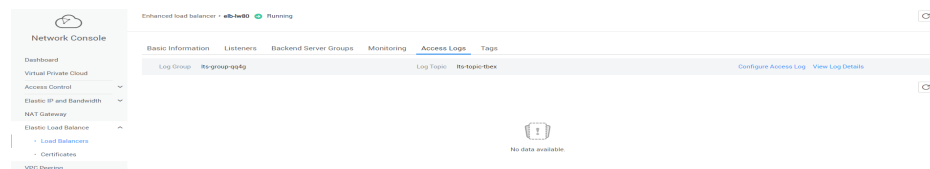
1. Click **Service List**. Under **Networking**, click **Elastic Load Balance**.
2. Locate the load balancer and click its name.
3. Under **Access Logs**, click **Configure Access Log**.
4. Enable access logging and select the created log group and log stream.
5. Click **OK**.

### NOTICE

Ensure that the log group is in the same region as the load balancer.

## Viewing Access Logs

- On the ELB console, click the name of the load balancer and click **Access Logs** to view logs.



- On the LTS console, go to the page that displays all log streams, locate the log stream and click **View** or **Search** in the **Operation** column.



## Locating the Server

The following is a log that records an exception:

```
1554944564.344 - [2019-04-11T09:02:44+08:00] elb 10.133.251.171:51527 500 "GET http://10.154.73.58/lrange/guestbook HTTP/1.1" 411 3726 3545 19.028 "500" "0.009" "19.028" "19.028" "172.17.0.82:3000" "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36" "http://10.154.73.58:5971/" "-" loadbalancer_ed0f790b-e194-4657-9f97-53426227099e listener_b21dd0a9-690a-4945-950e-b134095c6bd9 6b6aaf84d72b40fcb2d2b9b28f6a0b83
```

### Log analysis

At 09:02:44 GMT+08:00 of April 11, 2019, the load balancer received a GET/HTTP/1.1 request from the client (whose IP address and port number are 10.133.251.171 and 51527 respectively) and then routed the request to a backend server (which uses 172.17.0.82 and port 3000 to receive requests). The load balancer then received 500 Internal Server Error from the backend server and returned the status code to the client.

### Analysis result

The backend server was abnormal and failed to respond to the request.

 **NOTE**

172.17.0.82:3000 is the private IP address of the backend server.

# 2 Viewing Traffic Usage

---

## Scenarios


For livestreaming platforms, traffic often increases suddenly, which make the service unstable. To address this issue, most of them use ELB to distribute traffic. By working with Cloud Eye, ELB allows you to monitor the traffic usage in real time. You can view the traffic consumed by the EIPs bound to public network load balancers to better balance your application workloads.

## Prerequisites

Load balancers are running properly.

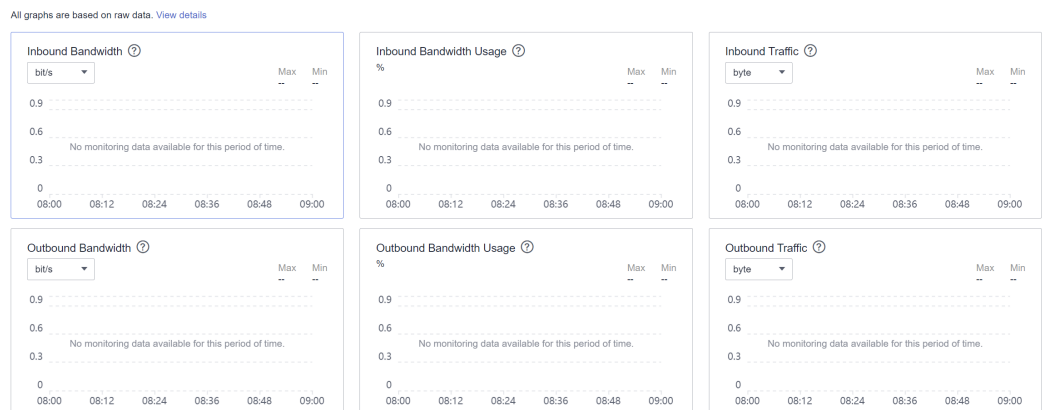
The associated backend servers are running normally and are not deleted or in the stopped or faulty state.

## Viewing Traffic Usage of the Bound EIP

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 **Service List**. Under **Networking**, click **Virtual Private Cloud**.
4. In the navigation pane on the left, choose **Elastic IP and Bandwidth > EIPs**.
5. Locate the EIP bound to the load balancer and click its name. On the **Bandwidth** page, you can view the data for the last 1, 3, 12 hours, last day, or last 7 days.



**Figure 2-1** EIP traffic usage




**Table 2-1** EIP and bandwidth metrics

Metric	Meaning	Value Range	Monitored Object	Monitoring Period (Raw Data)
Outbound Bandwidth (originally named "Upstream Bandwidth")	Network rate of outbound traffic	$\geq 0$ bits/s	Bandwidth or EIP	1 minute
Inbound Bandwidth (originally named "Downstream Bandwidth")	Network rate of inbound traffic	$\geq 0$ bits/s	Bandwidth or EIP	1 minute
Outbound Bandwidth Usage	Usage of outbound bandwidth in percentage.	0–100%	Bandwidth or EIP	1 minute
Inbound Bandwidth Usage	Usage of inbound bandwidth in the unit of percent.	0–100%	Bandwidth or EIP	1 minute

Metric	Meaning	Value Range	Monitored Object	Monitoring Period (Raw Data)
Outbound Traffic (originally named "Upstream Traffic")	Network traffic going out of the cloud platform	≥ 0 bytes	Bandwidth or EIP	1 minute
Inbound Traffic (originally named "Downstream Traffic")	Network traffic going into the cloud platform	≥ 0 bytes	Bandwidth or EIP	1 minute

## Viewing Load Balancer Traffic Metrics

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 **Service List**. Under **Networking**, click **Elastic Load Balance**.
4. Locate the load balancer and click its name.
5. Click the **Monitoring** tab, select load balancer for **Dimension**, and view the graphs of inbound and outbound rates.

You can view data from the last 1, 3, 12 hours, last day, or the last 7 days. For details, see [ELB Metrics](#).

# 3 Routing Traffic to Backend Servers in Different VPCs

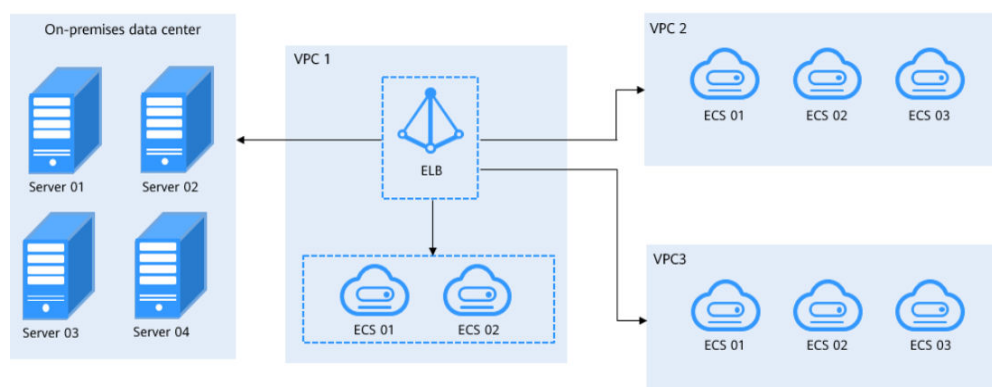
## 3.1 Overview

### Scenarios

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

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

**Figure 3-1** Routing traffic across cloud and on-premises servers



### Solution

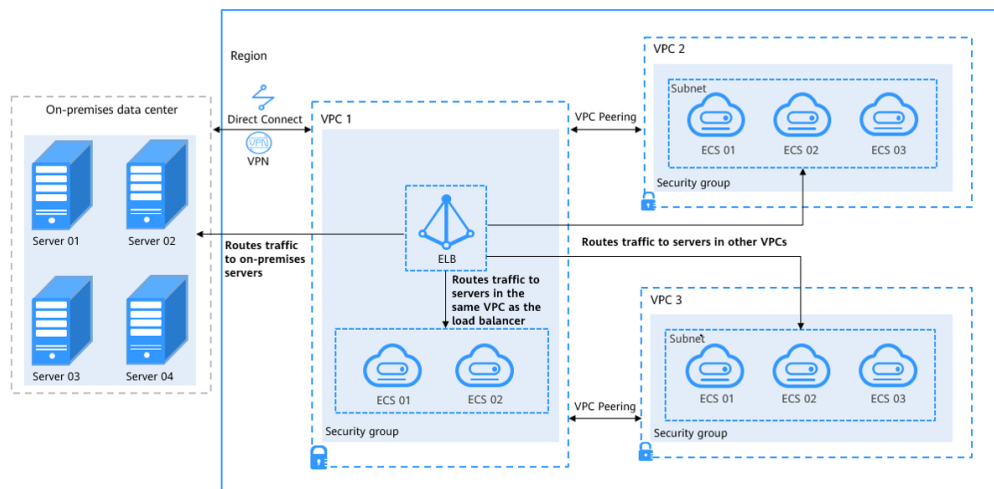
Dedicated load balancers can satisfy your needs. You can enable **IP as a Backend** when creating a dedicated load balancer and associate on-premises servers with this dedicated load balancer using their IP addresses.

As shown in [Figure 3-2](#), 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 backend servers in the same VPC where the load balancer is running.

**Figure 3-2** 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 backend servers in a VPC that is not the VPC where the load balancer is running by establishing a VPC peering connection between the two VPCs.
- You can add backend servers in your on-premises data center with the load balancer by connecting your on-premises data center to the cloud through Direct Connect or VPN.

## Restrictions and Limitations

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

- If you do not enable the function when you create a load balancer, you can still enable it on the **Basic Information** page of the load balancer.
- IP as backend servers must use IPv4 addresses.

- IP as backend servers cannot use public IP addresses or IP addresses from the VPC where the load balancer works. Otherwise, requests cannot be routed to backend servers.
- The subnet where the load balancer works must have at least 16 IP addresses. Otherwise, IP as backend servers cannot be added. You can add more subnets for more IP addresses on the **Basic Information** page of the load balancer.
- Security group rules of IP as backend servers must allow traffic from the subnet of the load balancer. Otherwise, health checks will fail.
- **IP as a Backend** cannot be disabled after it is enabled.

## 3.2 Routing Traffic to Backend Servers in Different VPCs from the Load Balancer

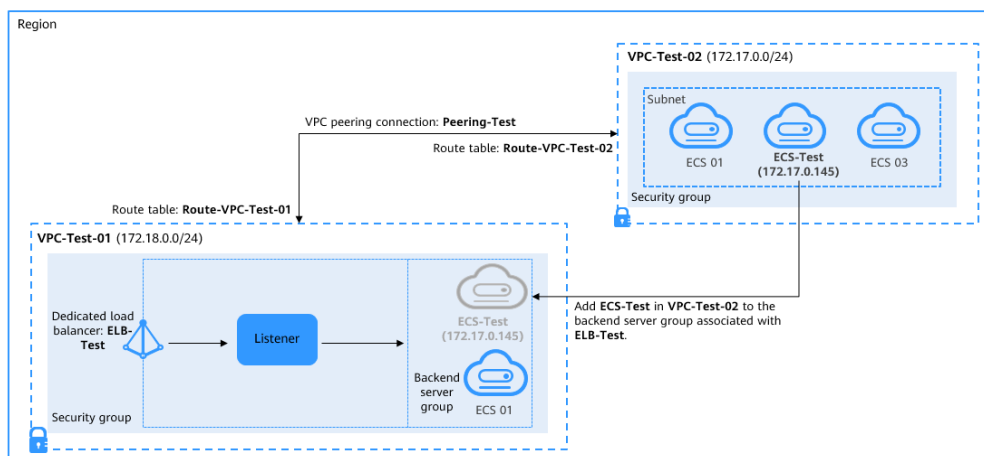
### Scenarios

You can use ELB to route traffic to backend servers in two VPCs connected over a VPC peering connection.

### Solution

- A dedicated load balancer named **ELB-Test** is running in **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 the dedicated load balancer **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**.

Figure 3-3 Topology



### Advantages

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

## Resource and Cost Planning

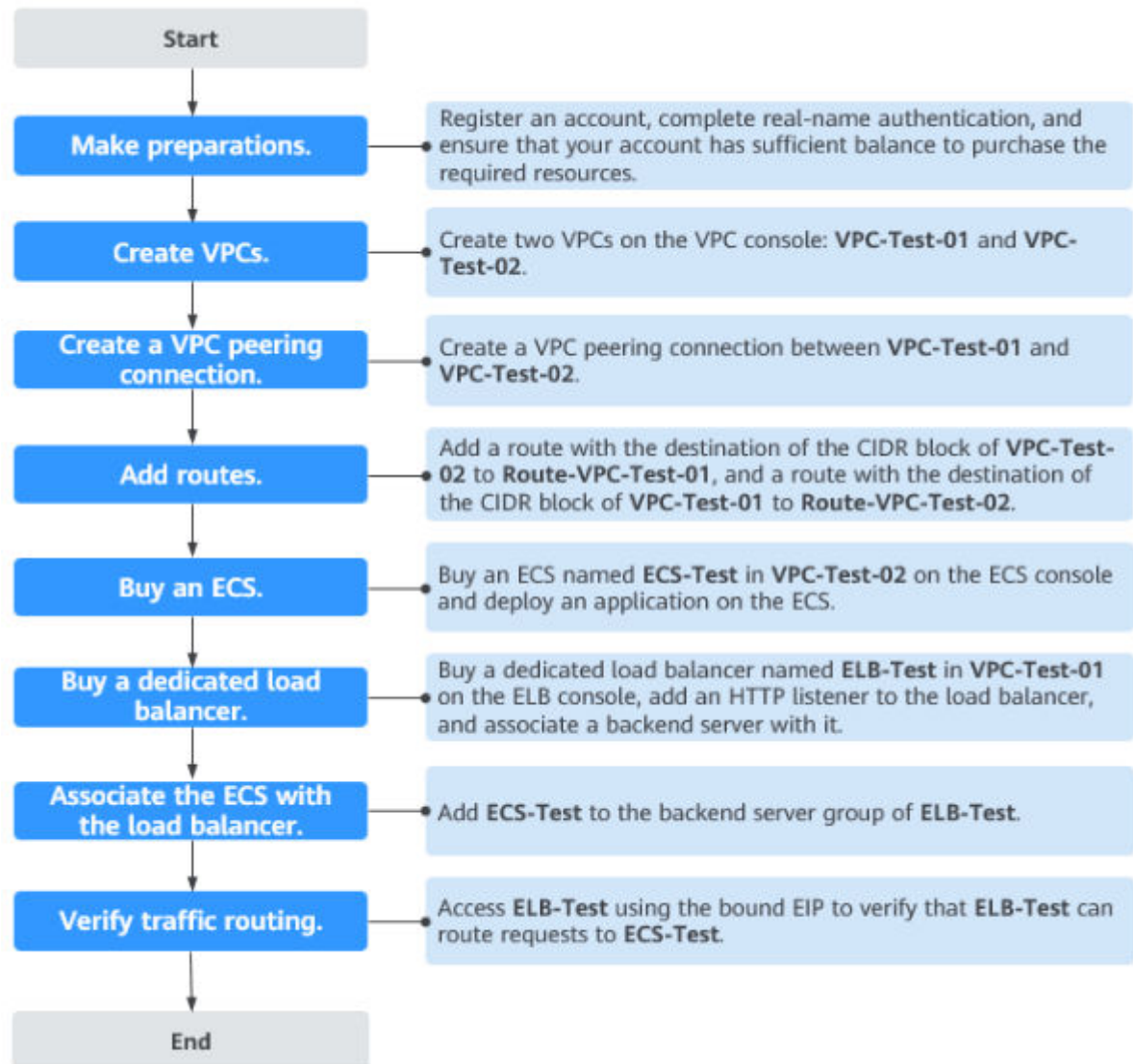
The estimated cost (CNY 782.78) provided in this document is for reference only. The actual cost shown on the Huawei Cloud console is used.

**Table 3-1** Resource planning

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

## Operation Process

**Figure 3-4** Process of associating servers in a VPC that is different from the dedicated load balancer



## Creating VPCs

**Step 1** Log in to the management console.

**Step 2** Under **Networking**, select **Virtual Private Cloud**. On the **Virtual Private Cloud** page displayed, click **Create VPC**.

**Step 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-Test-01**
- **IPv4 CIDR Block:** **172.18.0.0/24**
- Configure other parameters as required.

**Figure 3-5** Creating VPC-Test-01

**Basic Information**

Region: [Region dropdown]

Name: VPC-Test-01

IPv4 CIDR Block: 172 · 18 · 0 · 0 / 24

Recommended: 10.0.0.0/8-24 (Select) 172.16.0.0/12-24 (Select) 192.168.0.0/16-24 (Select)

Enterprise Project: longterm-EPSTest- [Create Enterprise Project ?]

Advanced Settings ▾ Tag | Description

**Default Subnet**

**Step 4** Repeat **Step 2** and **Step 3** to create the other VPC.

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

**Figure 3-6** Creating VPC-Test-02

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

----End



## Creating a VPC Peering Connection

**Step 1** In the navigation pane on the left, click **VPC Peering**.

**Step 2** In the upper right corner, click **Create VPC Peering Connection**.

**Step 3** Configure the parameters as follows and click **OK**. 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.

**Figure 3-7** Creating **Peering-Test**

### Create VPC Peering Connection

Local VPC Settings

\* Name

\* Local VPC

Local VPC CIDR Block 172.18.0.0/24

Peer VPC Settings

\* Account  My account  Another account

\* Peer Project

\* Peer VPC

Peer VPC CIDR Block 172.17.0.0/24

Description

0/255

----End

## Adding Routes for the VPC Peering Connection

**Step 1** In the navigation pane on the left, click **Route Tables**.

**Step 2** In the upper right corner, click **Create Route Table**.

**Step 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-Test-01
- **Destination:** 172.17.0.0/24
- **Next Hop Type:** VPC peering connection
- **Next Hop:** Peering-Test

**Figure 3-8** Creating Route-VPC-Test-01

Create Route Table

\* Name

\* VPC

IPv4 CIDR Block: 172.18.0.0/24

You can create 0 more route tables for the selected VPC.

Description  0/255

Route Settings

Destination ?	Next Hop Type ?	Next Hop ?	Description
Local	Local	Local	Default route that enables instance communication within a VPC

**Step 4** Repeat **3** and **4** 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

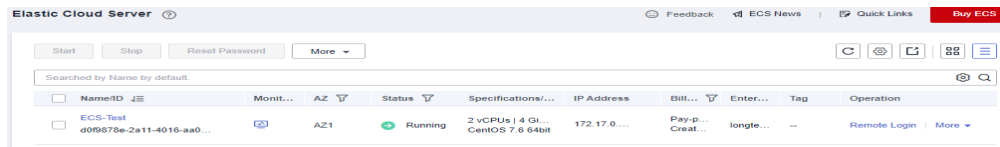
-----End

## Creating an ECS

**Step 1** Under **Computing**, click **Elastic Cloud Server**.

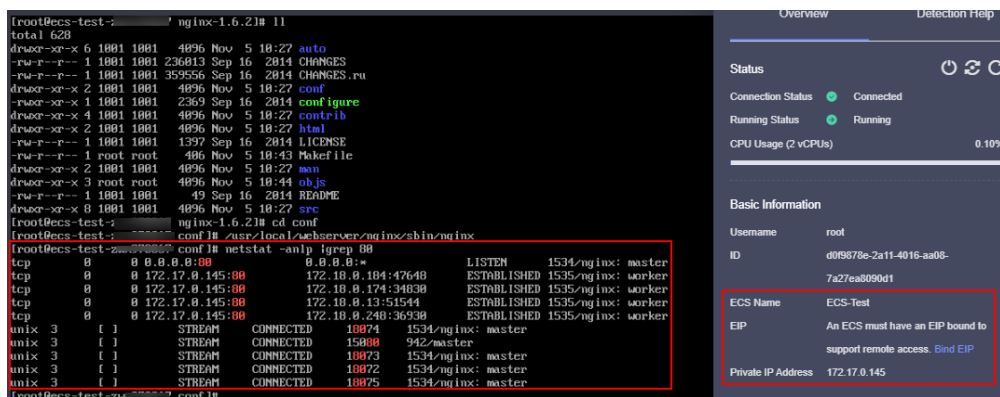
- Step 2** In the upper right corner, click **Buy ECS**.
- Step 3** Select **VPC-Test-02** as the **VPC** and set **ECS Name** to **ECS-Test**. Configure other parameters as required. For details, see [Elastic Cloud Server User Guide](#).

**Figure 3-9** Buying ECS-Test



- Step 4** Deploy Nginx on the ECS.

**Figure 3-10** Deploying Nginx on ECS-Test

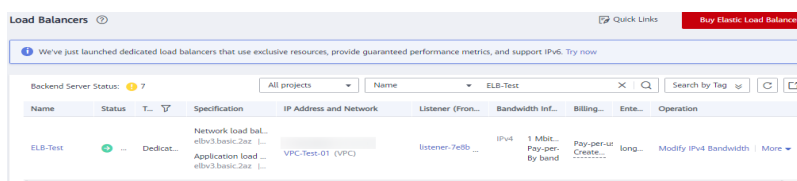


----End

## Buying a Dedicated Load Balancer and Adding an HTTP Listener and a Backend Server Group to the Load Balancer

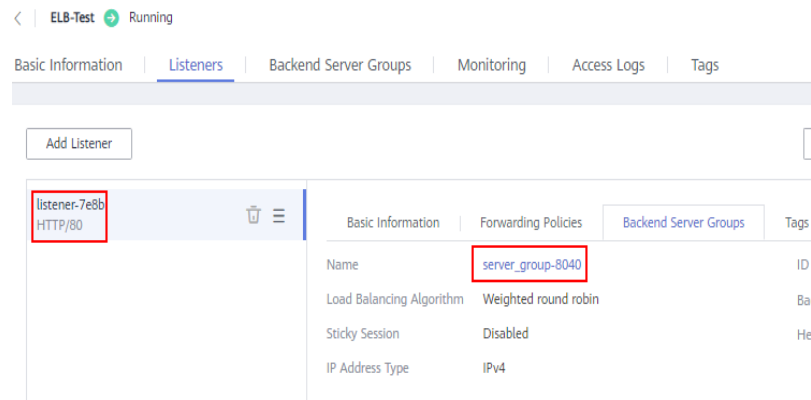
- Step 1** Under **Networking**, click **Elastic Load Balance**.
- Step 2** In the upper right corner, click **Buy Elastic Load Balancer**.
- Step 3** Configure the parameters as follows. For details, see [Elastic Load Balance User Guide](#).
  - **Type:** **Dedicated**
  - **IP as a Backend:** **Enable**
  - **VPC:** **VPC-Test-01**
  - **Name:** **ELB-Test**
  - Configure other parameters as required.

**Figure 3-11** Buying ELB-Test



- Step 4** Add an HTTP listener and a backend server group to the dedicated load balancer. For details, see [Elastic Load Balance User Guide](#).

**Figure 3-12** HTTP listener and backend server group



----End

## Adding the ECS to the Backend Server Group

- Step 1** Locate the created dedicated load balancer and click its name **ELB-Test**.
- Step 2** On the **Listeners** tab page, locate the HTTP listener added to the dedicated load balancer and click its name.
- Step 3** In the **Backend Server Groups** tab on the right, click **IP as Backend Servers**.

**Figure 3-13** IP as backend servers



- Step 4** Click **Add IP as Backend Server**, configure the parameters, and click **OK**. For details, see *Elastic Load Balance User Guide*.
- **Backend Server IP Address:** 172.17.0.145 (private IP address of **ECS-Test**)
  - **Backend Port:** the port enabled for Nginx on **ECS-Test**
  - **Weight:** Set this parameter as required.

**Figure 3-14** Adding ECS-Test using its IP address

✕

### Add IP as Backend Server

**i** • Use the TOA module to obtain IP addresses of clients. [Learn more](#)

• Ensure that the security group that contains the backend servers has rules allowing access from the backend subnet of the load balancer. If access is not allowed, health checks will fail.

Batch Add Ports

**i** You can add 495 more IP as Backend Servers. [Increase quota](#)

Backend Server IP Address	Backend Port <span>?</span>	Weight <span>?</span>	Operation
<input type="text" value="0 . 0 . 0 . 0"/>	<input type="text"/>	<input type="text" value="1"/>	Remove

----End

## Verifying Traffic Routing

- Step 1** Locate the dedicated load balancer **ELB-Test** and click **More** in the **Operation** column.
- Step 2** Select **Bind IPv4 EIP** to bind an EIP (119.3.233.52) to **ELB-Test**.

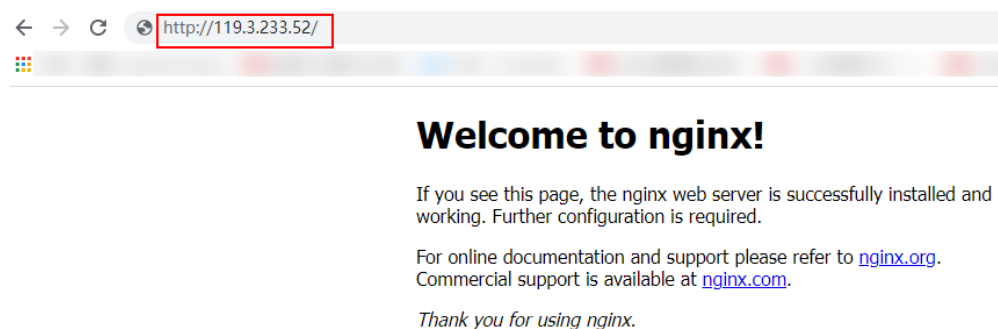
**Figure 3-15** EIP bound to the load balancer

The screenshot shows the Elastic Load Balance console. At the top, there are navigation links for 'Elastic Load Balance', 'Process Flow', 'Feedback', 'Quick Links', and 'Buy Elastic Load Balancer'. A notification banner states: 'We've just launched dedicated load balancers that use exclusive resources, provide guaranteed performance metrics, and support IPv6. Try now'. Below the banner, there are buttons for 'Renew', 'Change Billing Mode', 'Unsubscribe', and 'Backend Server Status'. A search bar is present with the text 'Specify filter criteria'. The main content is a table with the following columns: NameID, Monit..., Status, Type, Specifications, IP Address and Network, Listener (Frontend Protoc..., Bandwidth Informa..., Billing Mode, Enterprise Project, and Operation. The table contains one entry for 'ELB\_Test' with a status of 'Running', type of 'Dedicated', and specifications of 'Application load bala...'. The IP Address and Network column shows '(Private IPv4 ad... VPC\_Test\_01 (VPC)'. The Listener (Frontend Protoc... column shows 'Listener HTTP(HTTP80)'. The Billing Mode column shows 'Pay-per-use Created on May 28, ...'. The Operation column has 'Add Listener' and 'More' options.

NameID	Monit...	Status	Type	Specifications	IP Address and Network	Listener (Frontend Protoc...	Bandwidth Informa...	Billing Mode	Enterprise Project	Operation
ELB_Test		Running	Dedicated	Application load bala...	(Private IPv4 ad... VPC_Test_01 (VPC)	Listener HTTP(HTTP80)	--	Pay-per-use Created on May 28, ...	default	Add Listener More

- Step 3** Enter **http://119.3.233.52/** in the address box of your browser to access the dedicated load balancer. If the following page is displayed, the load balancer routes the request to **ECS-Test**, which processes the request and returns the requested page.

**Figure 3-16** Verifying that the request is routed to ECS-Test



----End

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

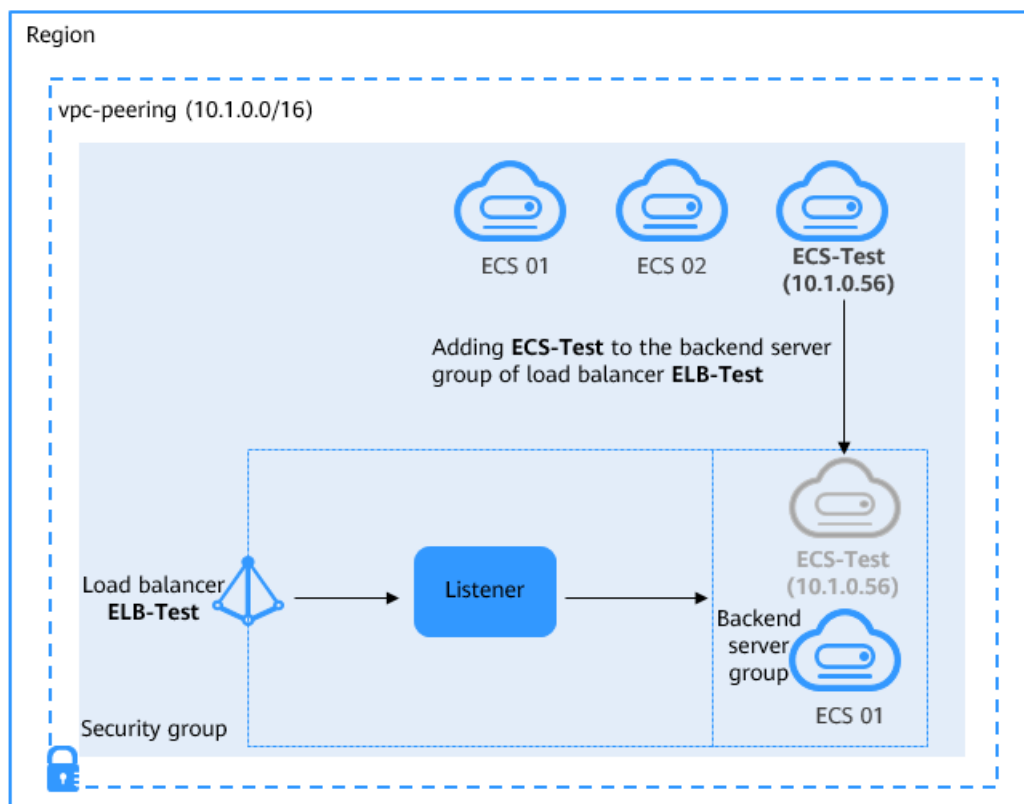
### Scenarios

You can route traffic to backend servers in the VPC where the load balancer is running.

### Solution

- A dedicated load balancer **ELB-Test** is running in a VPC named **vpc-peering** (10.1.0.0/16).
- The backend server **ECS-Test** is also running in **vpc-peering** (10.1.0.0/16).
- **ECS-Test** needs to be added to the backend server group associated with **ELB-Test**.

**Figure 3-17** Adding a backend server in the same VPC as the load balancer



## Advantages

You can add servers in the same VPC as the load balancer to the backend server group of the load balancer and then route incoming traffic to the servers.

## Resource and Cost Planning

The estimated cost (CNY 782.78) provided in this document is for reference only. The actual cost shown on the Huawei Cloud console is used.

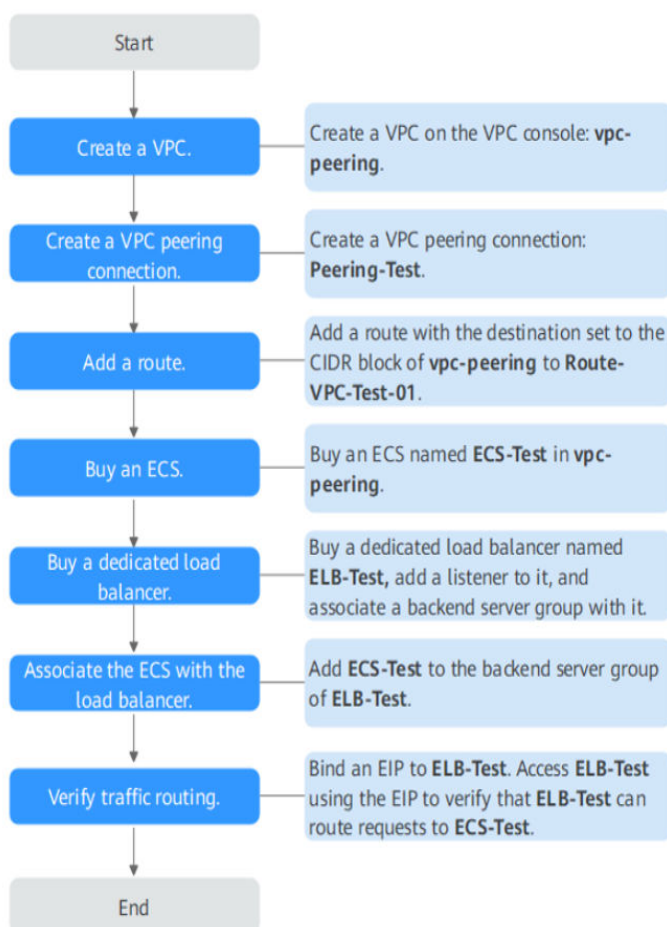
**Table 3-2** Resource planning

Resource Type	Resource Name	Description	Quantity
VPC	vpc-peering	The VPC where <b>ELB-Test</b> and <b>ECS-Test</b> are running: 10.1.0.0/16	1
VPC peering connection	Peering-Test	The connection that connects the VPC where <b>ELB-Test</b> is running and other VPCs <b>Local VPC: 10.1.0.0/16</b> <b>Peer VPC: any VPC</b>	1

Resource Type	Resource Name	Description	Quantity
Route table	Route-VPC-Test-01	The route table of <b>VPC-Test-01</b> <b>Destination: 10.1.0.0/16</b>	1
ELB	ELB-Test	The dedicated load balancer named <b>ELB-Test</b> <b>Private IP address: 10.1.0.9</b>	1
EIP	EIP-Test	The EIP (120.46.131.153) bound to <b>ELB-Test</b> 120.46.131.153	1
ECS	ECS-Test	The ECS works in <b>vpc-peering</b> <b>Private IP address: 10.1.0.56</b>	1

## Operation Process

**Figure 3-18** Process for adding backend servers in the same VPC as the load balancer





## Creating a VPC

- Step 1** Log in to the management console.
- Step 2** Under **Networking**, select **Virtual Private Cloud**. On the **Virtual Private Cloud** page displayed, click **Create VPC**.
- Step 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.

**Figure 3-19** Creating vpc-peering

**Basic Information**

Region

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

Name

IPv4 CIDR Block  ·  ·  ·  /

Enterprise Project  [Create Enterprise Project](#) [?](#)

Advanced Settings  Tag | Description

---

**Default Subnet**

AZ  [?](#)

Name

IPv4 CIDR Block [?](#)  ·  ·  ·  /

Available IP Addresses: 251  
The CIDR block cannot be modified after the subnet has been created.

IPv6 CIDR Block  Enable [?](#)

----End

## Creating a VPC Peering Connection

- Step 1** In the navigation pane on the left, click **VPC Peering**.
- Step 2** In the upper right corner, click **Create VPC Peering Connection**.

**Step 3** Configure the parameters as follows and click **OK**. 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.

**Figure 3-20** Creating Peering-Test

**Create VPC Peering Connection**

**i** A VPC peering connection allows two VPCs to communicate with each other if they are in the same region.  
If you need two VPCs in different regions to communicate with each other, use [Cloud Connect](#).

Local VPC Settings

* Name	Peering-Test
* Local VPC	vpc-peering
Local VPC CIDR Block	10.1.0.0/16

Peer VPC Settings

* Account	My account	Another account	?
* Peer Project			?
* Peer VPC			
Peer VPC CIDR Block			

**OK** Cancel

----End

## Adding Routes for the VPC Peering Connection

**Step 1** In the navigation pane on the left, click **Route Tables**.

**Step 2** In the upper right corner, click **Create Route Table**.

**Step 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**

**Figure 3-21** Creating Route-VPC-Test-01

**Create Route Table**

Name: Route-VPC-Test-01

VPC: vpc-peering

IPv4 CIDR block: 10.1.0.0/16

You can create 1 more route tables for the selected VPC.

Description: 0/255

Route Settings

Destination	Next Hop Type	Next Hop	Description
Local	Local	Local	Default route that enables instance communication within a VPC
10.1.0.0/16	VPC peerin...	Peering-Test(dc0e99f2-4419-4ed9-9...	

+ Add Route

OK Cancel

----End

## Creating an ECS

**Step 1** Under **Computing**, click **Elastic Cloud Server**.

**Step 2** In the upper right corner, click **Buy ECS**.

**Step 3** Configure the parameters as required. For details, see [Elastic Cloud Server User Guide](#).

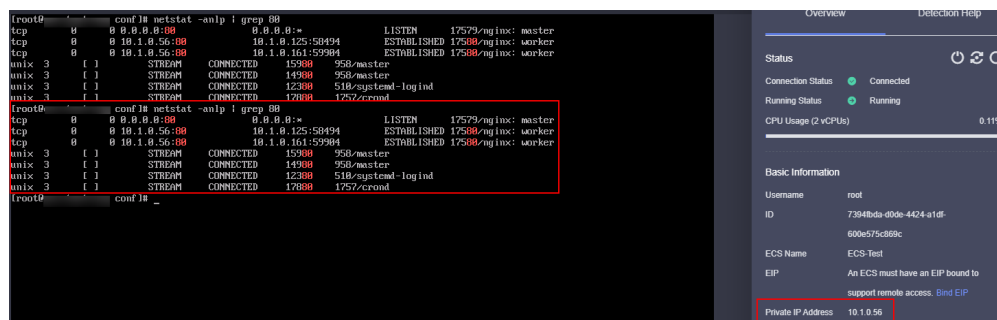
Select **vpc-peering** for VPC and set **Name** to **ECS-Test**.

**Figure 3-22** Buying ECS-Test

Name/ID	Monitoring	AZ	Status	Specifications/Image	IP Address	Billing Mode	Enterprise Project	Tag	Operation
ECS-Test i-7394fbc5-0506-4424-a10f-600e575c899c		AZ2	Running	2 vCPUs   4 GB   CentOS 7.6 64bit	10.1.0.58 (Private L...	Pay-per-use Created on May ...	longterm-EPSTe...	--	Remote Login   More

**Step 4** Deploy Nginx on the ECS.

Figure 3-23 Deploying Nginx on ECS-Test



----End

## Buying a Dedicated Load Balancer and Adding an HTTP Listener and a Backend Server Group to the Load Balancer

- Step 1** Under **Networking**, click **Elastic Load Balance**.
- Step 2** In the upper right corner, click **Buy Elastic Load Balancer**.
- Step 3** Configure the parameters as follows. For details, see [Elastic Load Balance User Guide](#).
  - **Type:** Dedicated
  - **IP as a Backend:** Enable
  - **VPC:** vpc-peering
  - **Name:** ELB-Test
  - Configure other parameters as required.

Figure 3-24 Creating a dedicated load balancer named ELB-Test



- Step 4** Add an HTTP listener and a backend server group to the created dedicated load balancer. For details, see [Elastic Load Balance User Guide](#).

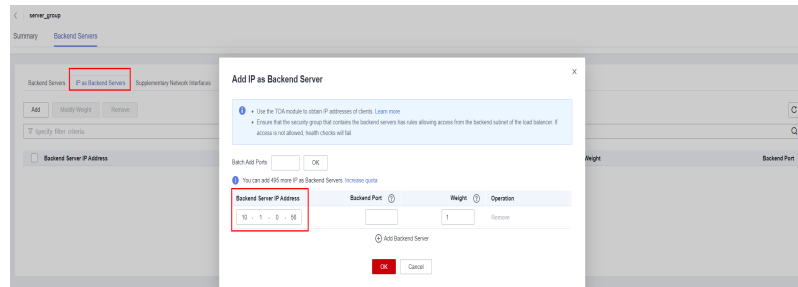
----End

## Adding the ECS to the Backend Server Group

- Step 1** Locate the dedicated load balancer and click its name **ELB-Test**.
- Step 2** On the **Listeners** tab page, locate the HTTP listener added to the dedicated load balancer and click its name.
- Step 3** In the **Backend Server Groups** tab on the right, click **IP as Backend Servers**.
- Step 4** Click **Add IP as Backend Server**, configure the parameters, and click **OK**. For details, see *Elastic Load Balance User Guide*.

- **Backend Server IP Address: 10.1.0.56** (private IP address of **ECS-Test**)
- **Backend Port:** the port enabled for Nginx on **ECS-Test**
- **Weight:** Configure this parameter as required.

Figure 3-25 Adding IP as backend servers

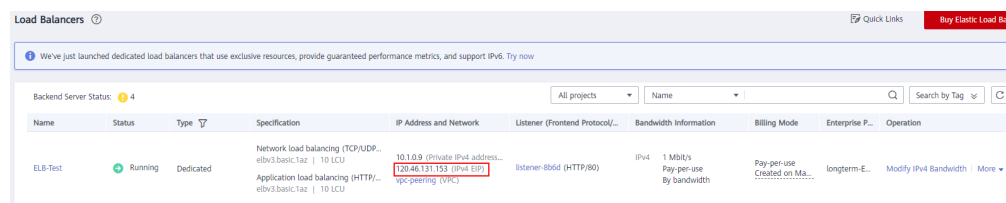


----End

## Verifying Traffic Routing

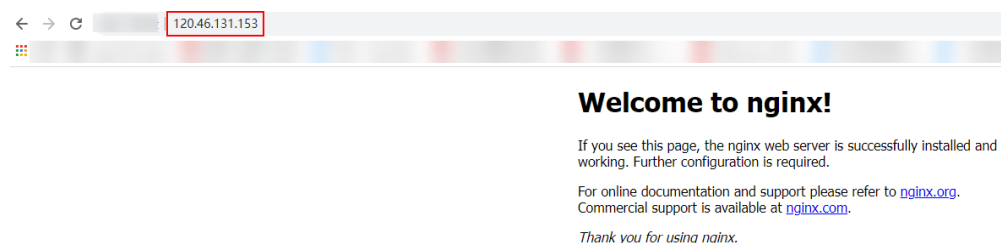
- Step 1** Locate the dedicated load balancer **ELB-Test** and click **More** in the **Operation** column.
- Step 2** Select **Bind IPv4 EIP** to bind an EIP (120.46.131.153) to **ELB-Test**.

Figure 3-26 EIP bound to the load balancer



- Step 3** Enter **http://120.46.131.153/** in the address box of your browser to access the dedicated load balancer. If the following page is displayed, the load balancer routes the request to **ECS-Test**. After receiving the request from the load balancer, **ECS-Test** processes the request and returns the requested page.

Figure 3-27 Verifying traffic routing



----End

# 4 Using Advanced Forwarding for Application Iteration

---

## Scenarios

As the business grows, you may need to upgrade your application. Both the old and new versions are used. Now, the new version is optimized based on users' feedback, and you want all the users to use the new version. In this process, you can use advanced forwarding to route requests to different versions.

## Prerequisites

- A HUAWEI CLOUD account is available and real-name authentication has been completed.
- The account is not in arrears and the account balance is sufficient to pay for the resources involved in this best practice.
- 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 4-1 Flowchart

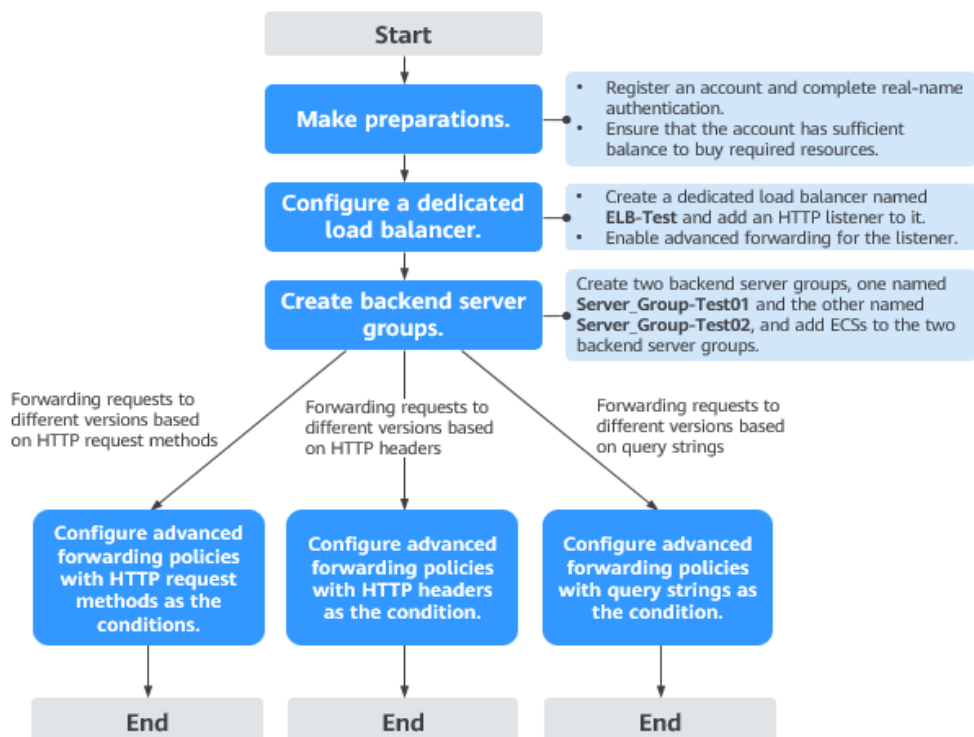


Table 4-1 Resource planning

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 <b>Server_Group-Test01</b> .
ECS02	ECS	Used to deploy the application of the old version and added to <b>Server_Group-Test01</b> .
ECS03	ECS	Used to deploy the application of the old version and added to <b>Server_Group-Test01</b> .

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

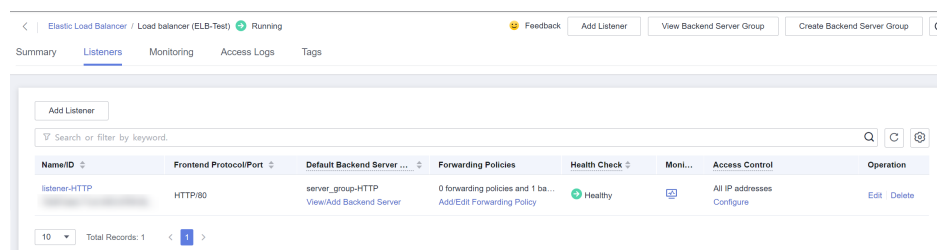
**NOTE**

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 [Routing Traffic to Backend Servers in Different VPCs](#).

## Configuring a Dedicated Load Balancer

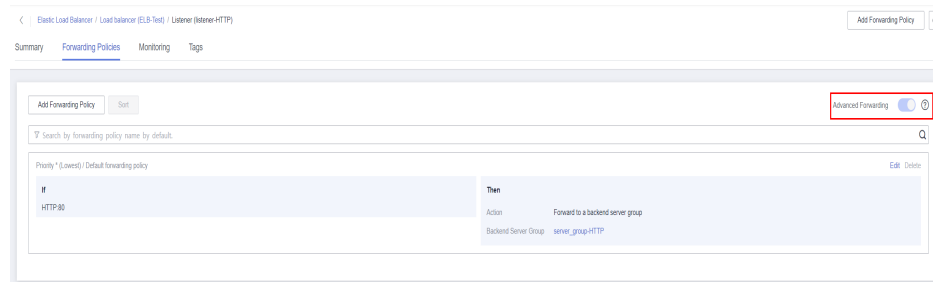
- Step 1** Log in to the management console.
- Step 2** Under **Networking**, click **Elastic Load Balance**.
- Step 3** In the upper right corner, click **Buy Elastic Load Balancer**.
- Step 4** Create a dedicated load balancer **ELB-Test**. Configure the parameters as follows. For details, see [Elastic Load Balance User Guide](#).
  - **Type: Dedicated**
  - **Name: ELB-Test**
  - Configure other parameters as required.
- Step 5** Add an HTTP listener to **ELB-Test**. For details, see [Elastic Load Balance User Guide](#).

Figure 4-2 HTTP listener



- Step 6** Enable advanced forwarding. For details, see [Elastic Load Balance User Guide](#)



**Figure 4-3** Enabling advanced forwarding

----End

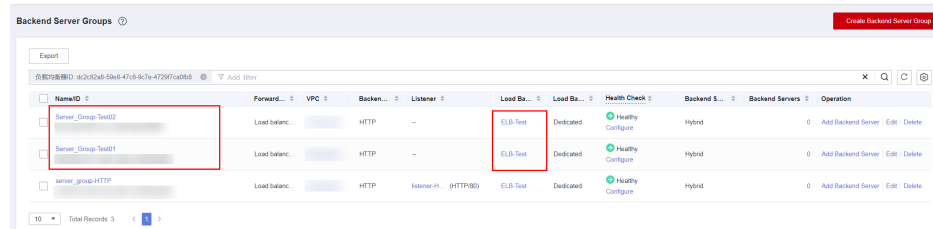
## Creating Backend Server Groups and Adding Backend Servers

**Step 1** Locate **ELB-Test** and click its name.

**Step 2** On the **Listeners** tab, click **Create Backend Server Group** in the upper right corner.

- Name: **Server\_Group-Test01**
- **Backend Protocol: HTTP**
- Configure other parameters as required.

**Step 3** Repeat **Step 2** to create backend server group **Server\_Group-Test02**.

**Figure 4-4** Backend server groups

NameID	Forward...	VPC	Backen...	Listener	Load Ba...	Load Ba...	Health Check	Backend S...	Backend Servers	Operation
Server_Group-Test02	Load balanc.		HTTP	--	ELB-Test	Dedicated	Healthy Configure	Hybrid	0	Add Backend Server Edit Delete
Server_Group-Test01	Load balanc.		HTTP	--	ELB-Test	Dedicated	Healthy Configure	Hybrid	0	Add Backend Server Edit Delete
server_group_HTTP	Load balanc.		HTTP	Internet H... (HTTP:80)	ELB-Test	Dedicated	Healthy Configure	Hybrid	0	Add Backend Server Edit Delete

**Step 4** Add **ECS01**, **ECS02**, and **ECS03** to backend server group **Server\_Group-Test01**.

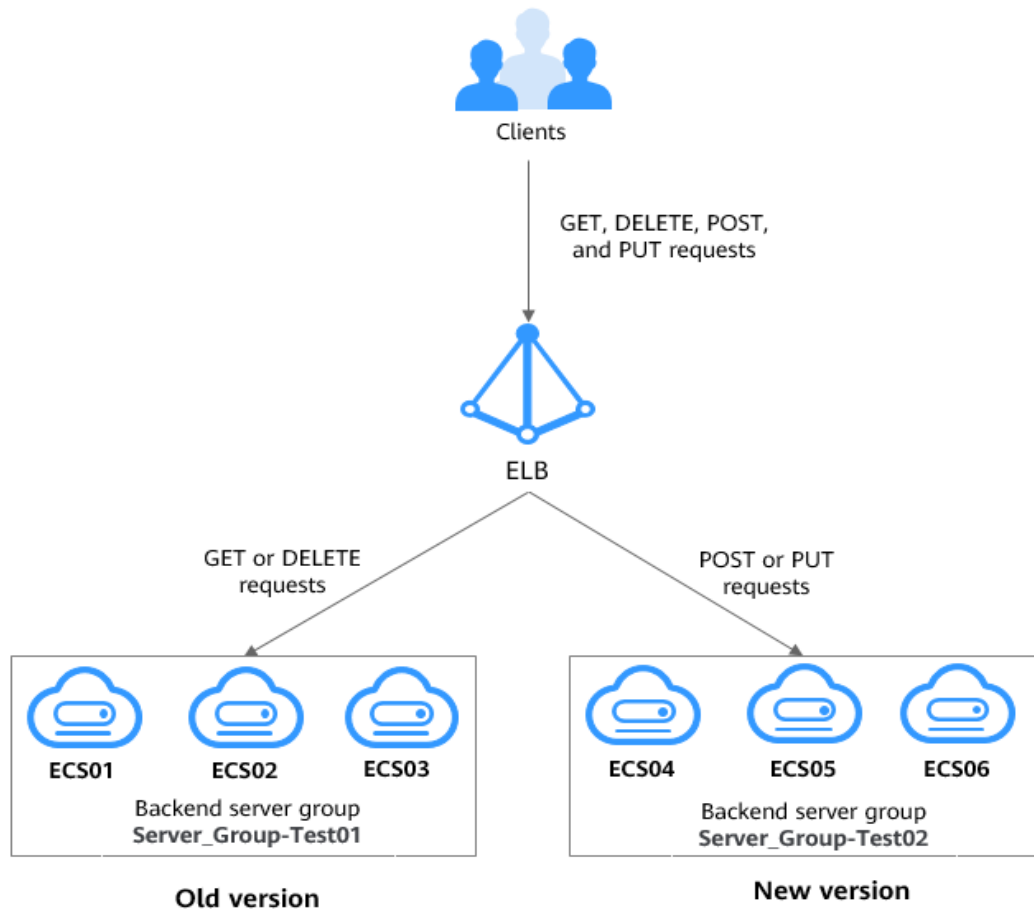
**Step 5** Add **ECS04**, **ECS05**, and **ECS06** to backend server group **Server\_Group-Test02**

----End

## 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 application.

**Figure 4-5** Forwarding requests based on HTTP request methods



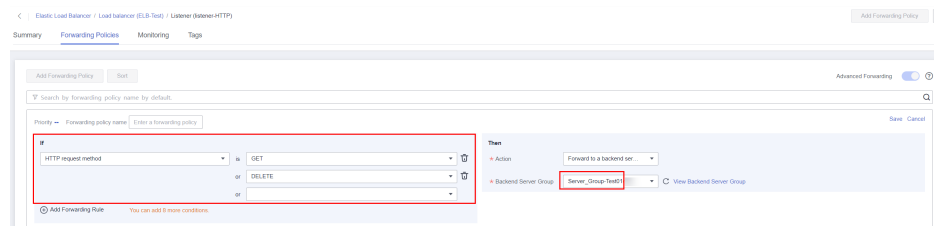
**Step 1** Locate the dedicated load balancer and click its name **ELB-Test**.

**Step 2** On the **Listeners** tab page, locate the HTTP listener added to the dedicated load balancer and click its name.

**Step 3** On the **Forwarding Policies** tab page on the right, click **Add Forwarding Policy** to forward GET and DELETE requests to the old version.

Select **GET** and **DELETE** from the **HTTP request method** drop-down list, select **Forward to backend server group** for **Action**, and select **Server\_Group-Test01** from the **Backend Server Group** drop-down list.

**Figure 4-6** Forwarding GET and DELETE requests to the application of the old version

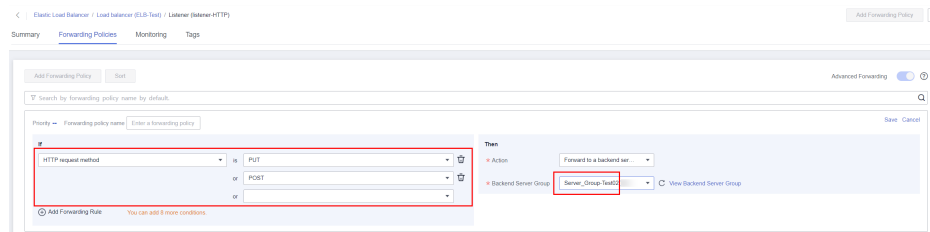


**Step 4** Click **Save**.

**Step 5** Repeat **Step 3** and **Step 4** 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 backend server group** for **Action**, and select **Server\_Group-Test02** from the **Backend Server Group** drop-down list.

**Figure 4-7** Forwarding PUT and POST requests to the application of the new version

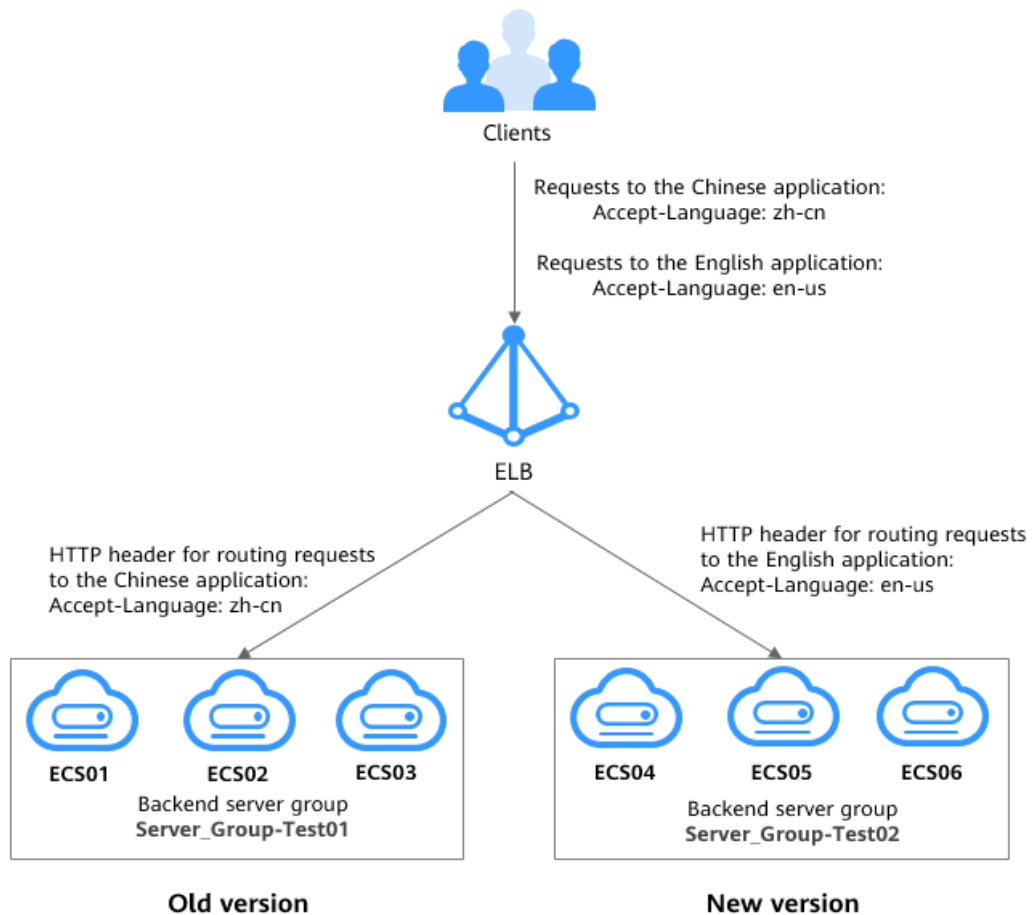


----End

## 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 language, direct all the requests to the application.

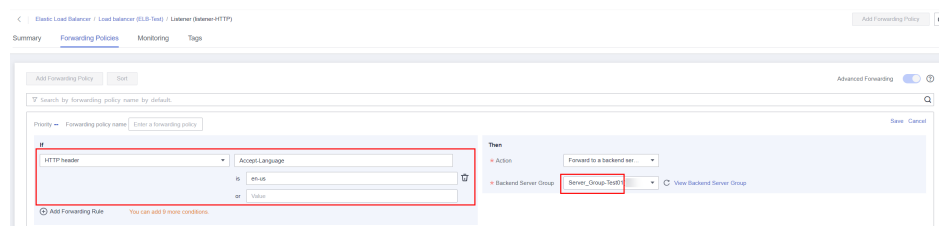
**Figure 4-8** Smooth application transition between the old and new versions based on the HTTP request header



- Step 1** Locate the dedicated load balancer and click its name **ELB-Test**.
- Step 2** On the **Listeners** tab page, locate the HTTP listener added to the dedicated load balancer and click its name.
- Step 3** On the **Forwarding Policies** tab page on the right, and click **Add Forwarding Policy** to forward requests to the old 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 backend server group**, and select **Server\_Group-Test01** as the backend server group.

**Figure 4-9** Forwarding requests to the application of the old version

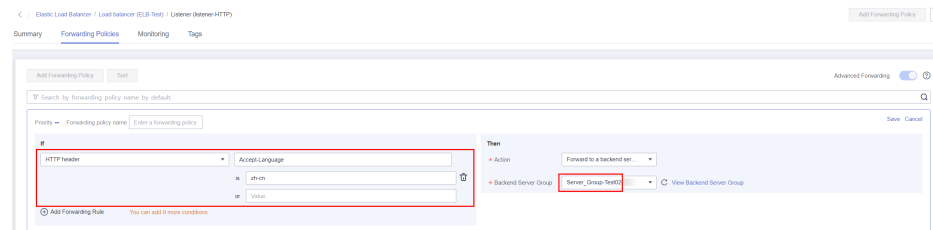


- Step 4** Click **Save**.

**Step 5** Repeat **Step 3** and **Step 4** 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 **en-us**, set the action to **Forward to backend server group**, and select **Server\_Group-Test02** as the backend server group.

**Figure 4-10** Forwarding requests to the application of the new version

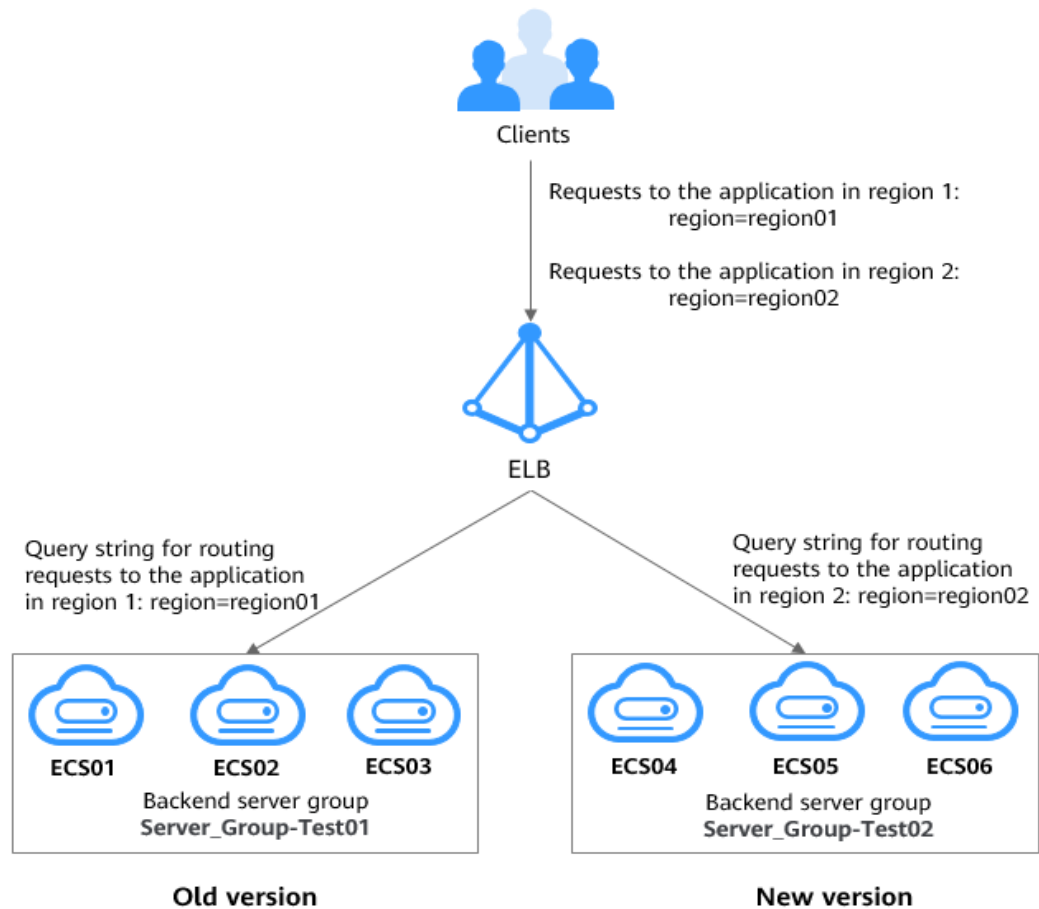


----End

## 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 4-11** Forwarding requests based on query strings



**NOTE**

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

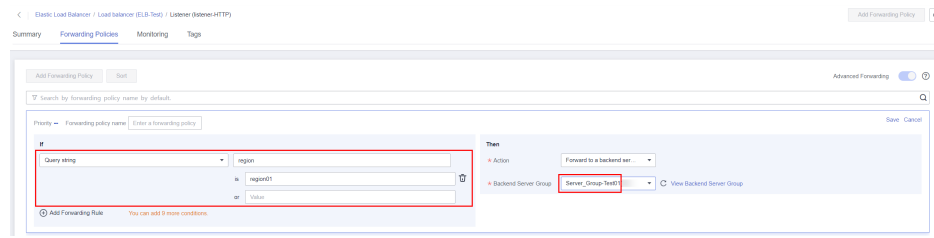
**Step 1** Locate the dedicated load balancer and click its name **ELB-Test**.

**Step 2** On the **Listeners** tab page, locate the HTTP listener added to the dedicated load balancer and click its name.

**Step 3** On the **Forwarding Policies** tab page 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 backend server group**, and select **Server\_Group-Test01** as the backend server group.

**Figure 4-12** Forwarding requests to the old version

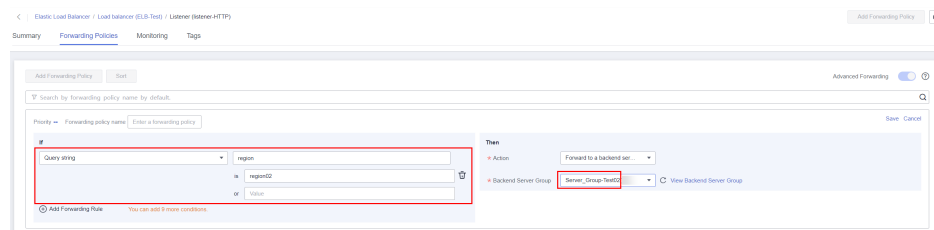


**Step 4** Click **Save**.

**Step 5** Repeat **Step 3** and **Step 4** 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 backend server group**, and select **Server\_Group-Test02** as the backend server group.

**Figure 4-13** Forwarding requests to the new version



----End