

Elastic Load Balance

Getting Started

Issue	01
Date	2025-08-22



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 Using ELB to Distribute Traffic to a Web Application Across ECSs.....	1
2 Using ELB to Distribute Traffic to Multiple Web Applications Across ECSs.....	13

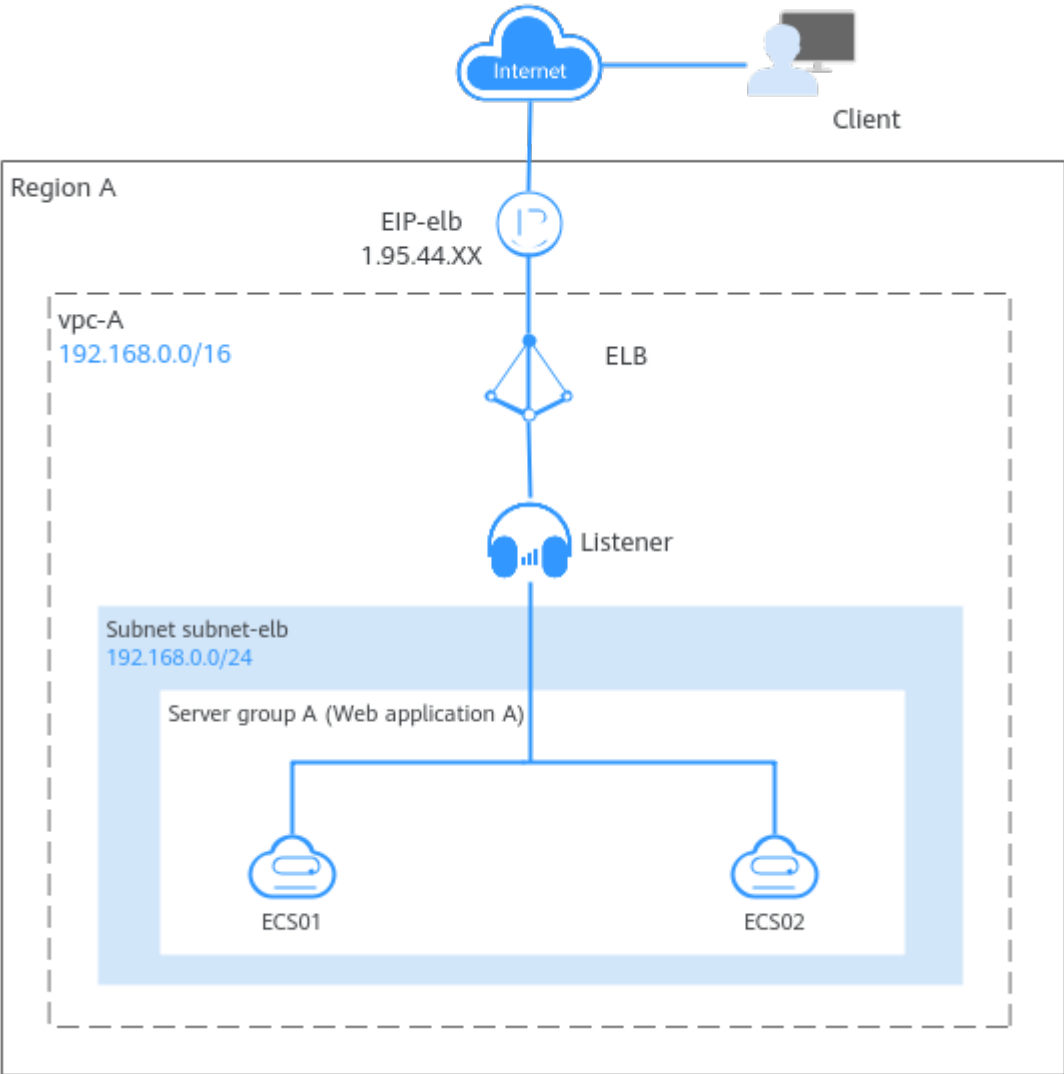
1 Using ELB to Distribute Traffic to a Web Application Across ECSs

Scenarios

ELB distributes incoming traffic across multiple servers based on the routing rules you configure. It improves their availability by eliminating single points of failure (SPOFs).

If you have a web application that needs to handle heavy traffic, you can deploy your application on two ECSs (**ECS01** and **ECS02** in this example) and create a load balancer to distribute traffic across the two ECSs.

Figure 1-1 Using ELB to distribute traffic to an application



Procedure

You can follow the process in [Figure 1-2](#) to use a load balancer to distribute traffic to a web application running on two ECSs.

Figure 1-2 Distributing traffic to a web application running on two ECSs



Procedure	What to Do
Preparations	Before using cloud services, sign up for a HUAWEI ID and enable Huawei Cloud services.

Procedure	What to Do
Step 1: Create a VPC and Two ECSs	<ul style="list-style-type: none">• Create a VPC with an IPv4 CIDR block and create a subnet in the VPC.<ul style="list-style-type: none">– VPC IPv4 CIDR block: 192.168.0.0/16– Subnet IPv4 CIDR block: 192.168.0.0/24• Buy two ECSs in the VPC subnet you have created.
Step 2: Deploy the Application	Deploy Nginx on the two ECSs.
Step 3: Create a Load Balancer	Create a load balancer with elastic specifications to receive requests from clients and distribute the requests to backend servers.
Step 4: Configure Security Group Rules	Configure security group rules to allow traffic from the backend subnet where the load balancer works to the backend servers.
Step 5: Add a Listener	Add a listener to the load balancer to check requests from clients and route requests to backend servers in the backend server group.
Step 6: Verify Load Balancing	Access the domain name of the load balancer to check whether it can route requests across the two backend servers.

Preparations

Before creating resources such as VPCs and ECSs, you need to [sign up for a HUAWEI ID and enable Huawei Cloud services](#).

If you already have a HUAWEI ID, skip this part.

Step 1: Create a VPC and Two ECSs

You need to plan the region for your load balancer, and create a VPC and two ECSs. Ensure that the ECSs and load balancer are in the same AZ and VPC.

1. Create a VPC.

VPC Configurations

- Configuring a VPC

Parameter	Example Value	Description
Name	vpc-A	The VPC name. Set it to vpc-A .
IPv4 CIDR Block	192.168.0.0/16	The IPv4 CIDR block of vpc-A .

Parameter	Example Value	Description
Enterprise Project	default	The enterprise project by which VPCs are centrally managed. Select an existing enterprise project for vpc-A .
Advanced Settings (Optional)	-	In this example, retain the default values.

– Configuring a VPC subnet

Parameter	Example Value	Description
AZ	AZ1	A geographic location with independent power supply and network facilities in a region. Each region contains multiple AZs. AZs are physically isolated but connected through an internal network. Subnets of a VPC can be located in different AZs without affecting communications. You can select any AZ in a region. The ECSs and VPC can be in different AZs. For example, you can select AZ1 for the ECS and AZ3 for the VPC subnet.
Subnet Name	subnet-elb	The subnet name. Set it to subnet-elb .
IPv4 CIDR Block	192.168.0.0/24	The IPv4 CIDR block of subnet-elb , which is a unique CIDR block with a range of IP addresses in vpc-A .
IPv6 CIDR Block (Optional)	Do not enable	Whether to assign IPv6 addresses.
Associated Route Table	Default	The default route table that subnet-elb is associated with. The default route table has a preset system route that allows subnets in a VPC to communicate with each other.
Advanced Settings (Optional)	-	In this example, retain the default values.

2. Create two ECSs.
ECS Configurations

- Configuring the network parameters for the two ECSs

Parameter	Example Value	Description
ECS Name	<ul style="list-style-type: none">• ECS01• ECS02	Names of each ECS. Set them to ECS01 and ECS02 .
AZ	<ul style="list-style-type: none">• AZ1• AZ2	The AZ of each ECS. Select different AZs for the two ECSs.
Network	<ul style="list-style-type: none">• vpc-A• subnet-elb	The VPC and subnet where the ECSs work. Select vpc-A and subnet-elb for the two ECSs.
EIP	<ul style="list-style-type: none">• EIP01• EIP02	The EIP bound to each ECS for Internet access.

Step 2: Deploy the Application

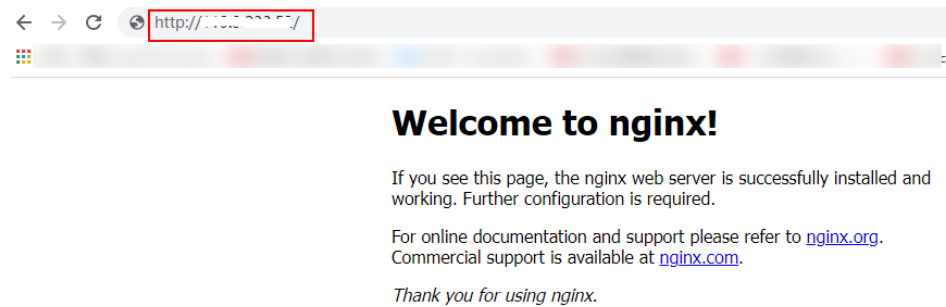
Deploy Nginx on the two ECSs and edit two HTML pages so that a page with message "Welcome to ELB test page one!" is returned when **ECS01** is accessed, and the other page with message "Welcome to ELB test page two!" is returned when **ECS02** is accessed.

1. [Log in to the ECSs.](#)
2. Install and start Nginx.

In this example, the two ECSs use CentOS 7.6 as the operating system.

Deploying Nginx

- a. Run the **wget** command to download the Nginx installation package for your operating system in use.
`wget http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7ngx.noarch.rpm`
- b. Run the following command to create the Nginx yum repository.
`rpm -ivh nginx-release-centos-7-0.el7ngx.noarch.rpm`
- c. Run the following command to install Nginx:
`yum -y install nginx`
- d. Run the following commands to start Nginx and configure automatic Nginx enabling upon ECS startup:
`systemctl start nginx`
`systemctl enable nginx`
- e. Enter **http://EIP bound to any ECS** in the address box of your browser. If the following page is displayed, Nginx has been installed.

Figure 1-3 Nginx installed successfully

3. Modify the HTML page of **ECS01**.

Modify the **index.html** file in the default root directory (**/usr/share/nginx/html**) of Nginx to identify access to **ECS01**.

Modifying the HTML Page of **ECS01**

a. Run the following command to open the **index.html** file:

```
vim /usr/share/nginx/html/index.html
```

b. Press **i** to enter editing mode.

c. Modify the **index.html** file.

The following is the content to be modified:

```
...
<body>
  <h1>Welcome to <strong>ELB</strong> test page one!</h1>

  <div class="content">
    <p>This page is used to test the <strong>ELB</strong>!</p>

    <div class="alert">
      <h2>ELB01</h2>
      <div class="content">
        <p><strong>ELB test (page one)!</strong></p>
        <p><strong>ELB test (page one)!</strong></p>
        <p><strong>ELB test (page one)!</strong></p>
      </div>
    </div>
  </div>
</body>
```

d. Press **Esc** to exit the editing mode. Then, enter **:wq** to save the settings and exit the file.

4. Modify the HTML page of **ECS02** by referring to step 3 to identify the access to **ECS02**.

Modifying the HTML Page of **ECS02**

a. Run the following command to open the **index.html** file:

```
vim /usr/share/nginx/html/index.html
```

b. Press **i** to enter editing mode.

c. Modify the **index.html** file.

The following is the content to be modified:

```
...
<body>
  <h1>Welcome to <strong>ELB</strong> test page two!</h1>

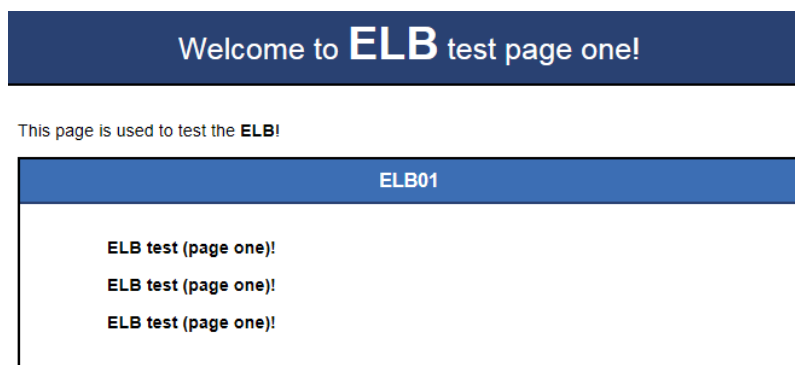
  <div class="content">
    <p>This page is used to test the <strong>ELB</strong>!</p>

    <div class="alert">
```

```
<h2>ELB02</h2>
<div class="content">
  <p><strong>ELB test (page two)!</strong></p>
  <p><strong>ELB test (page two)!</strong></p>
  <p><strong>ELB test (page two)!</strong></p>
</div>
</div>
</div>
</body>
```

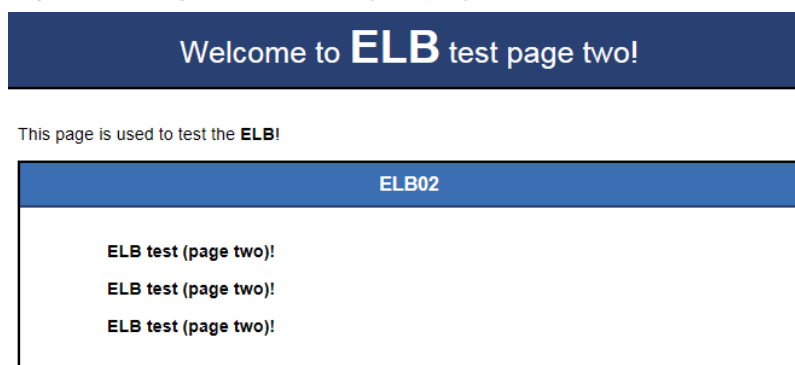
- d. Press **Esc** to exit the editing mode. Then, enter **:wq** to save the settings and exit the file.
5. Use your browser to access **http://EIP bound to ECS01** and **http://EIP bound to ECS02** in sequence to verify that Nginx has been deployed.
If the modified HTML pages are displayed, Nginx has been deployed.
 - HTML page of **ECS01**

Figure 1-4 Nginx successfully deployed on **ECS01**



- HTML page of **ECS02**

Figure 1-5 Nginx successfully deployed on **ECS02**



Step 3: Create a Load Balancer

In this example, the load balancer needs an EIP to forward Internet requests to the application deployed on the ECSs. You can determine whether to bind an EIP to the load balancer based on service requirements.

1. Go to the [Buy Elastic Load Balancer](#) page.
2. On the displayed page, set the parameters as required.

Basic Configurations

- Configuring the basic information

Parameter	Example Value	Description
Type	Dedicated load balancer	Specifies the type of the load balancer. Select Dedicated load balancer . A dedicated load balancer uses dedicated resources. Its performance is not affected by other load balancers.
Region	-	Specifies the desired region. Resources in different regions cannot communicate with each other over internal networks. Select the same region as the ECSs.
AZ	<ul style="list-style-type: none">• AZ1• AZ2	Specifies the AZs of the load balancer. Select multiple AZs if you need DR capability. The load balancer performance multiplies as the number of AZs increases.
Name	elb-test	Specifies the name of the load balancer.
Enterprise Project	default	Specifies an enterprise project by which cloud resources and members are centrally managed.


3. Set **Specification Type** to **Elastic**.
4. Configure the network parameters and EIP information.

Network Configurations

- Configuring network parameters

Parameter	Example Value	Description
Network Type	Private IPv4 network	Specifies the network where the load balancer works. In this example, select Private IPv4 network . The load balancer uses the private IPv4 address to process private network requests. To let the load balancer route requests over the Internet, bind an EIP to it.
VPC	-	Specifies the VPC where the load balancer works. In this example, select vpc-A .
Frontend Subnet	-	Specifies the frontend subnet from which an IPv4 address will be assigned to the load balancer to receive client requests. If IPv6 is enabled, an IPv6 address will also be assigned to the load balancer.

Parameter	Example Value	Description
IPv4 Address	Automatically assign IP address	Specifies how you want the IPv4 address to be assigned.
Backend Subnet	Subnet of the load balancer	Specifies the backend subnet from which IP addresses will be assigned to the load balancer to forward requests to backend servers.
IP as a Backend	-	Specifies whether to add IP addresses as backend servers that are not in the VPC of the load balancer. In this example, leave this feature disabled.
Internet Access		
EIP	Auto assign	Specifies the EIP that will be bound to the load balancer for receiving and forwarding requests over the Internet.
EIP Type	Dynamic BGP	Specifies the link type (BGP) when a new EIP is used.
Billed By	Traffic	Specifies how the bandwidth will be billed. In this example, select Traffic . You need to specify a maximum bandwidth and pay for the outbound traffic you use.
Bandwidth (Mbit/s)	100	Specify the maximum bandwidth.

- Specifies the maximum bandwidth. Click  to expand the advanced settings and add a description and tags to the load balancer.
- Click **Buy Now**.
- Confirm the configuration and submit your request.
- View the newly created load balancer in the load balancer list.

Step 4: Configure Security Group Rules

To ensure normal communications between the load balancer and backend servers, you need to check the security group and network ACL rules.

- The security groups configured for backend servers must have inbound rules to allow health check and service traffic from backend subnet of the load balancer to backend servers. By default, the backend subnet of a load balancer is the same as the subnet where the load balancer works.
- Network ACL rules are optional for subnets. If network ACL rules are configured for the subnet where backend servers are running, the inbound rules must allow traffic from the backend subnet of the load balancer to the subnet of backend servers.

For details about how to configure security group and network ACL rules, see [Security Group and Network ACL Rules](#).

You can configure security group rules based on [Table 1-1](#).

Table 1-1 Security group rules

Direction	Action	Type	Protocol & Port	Source/Destination	Description
Inbound	Allow	IPv4	TCP: 80	Source: 192.168.0.0/24	Allows outbound traffic to ECSs in the security group.
Outbound	Allow	IPv4	All	Destination: 0.0.0.0/0	Allows the ECSs in the security group to access the external networks.

Step 5: Add a Listener

Add a listener to the created load balancer. When you add the listener, create a backend server group, configure a health check, and add the two ECSs to this backend server group. If a backend server is detected unhealthy, the load balancer will stop routing traffic to it until the backend server recovers.

1. On the load balancer list page, locate load balancer **elb-test** and click its name.
2. On the **Listeners** tab, click **Add Listener** and configure parameters based on [Table 1-2](#).

Table 1-2 Parameters for configuring a listener

Parameter	Example Value	Description
Name	listener	Specifies the listener name.
Frontend Protocol	HTTP	Specifies the protocol that will be used by the listener to receive requests from clients.
Frontend Port	80	Specifies the port that will be used by the listener to receive requests from clients.
Redirect	-	Specifies whether to redirect requests from the HTTP listener to an HTTPS listener. Disable it in this example.
Access Control	All IP addresses	Specifies how access to the listener is controlled.
Transfer Client IP Address	-	Specifies whether to transmit IP addresses of the clients to backend servers. This feature is enabled for dedicated load balancers by default and cannot be disabled.

Parameter	Example Value	Description
Advanced Forwarding	-	Specifies whether to enable advanced forwarding. Once it is enabled, more forwarding rules and actions are supported. Enable it in this example.

3. Retain the default values for **Advanced Settings (Optional)**. Click **Next: Configure Request Routing Policy**. On the displayed page, select **Create new** for **Backend Server Group**.
Set **Load Balancing Algorithm** to **Weighted round robin** and retain the default values for other parameters.
4. Click **Next: Add Backend Server**.
 - Backend servers: Click **Add Cloud Server** and select **ECS01** and **ECS02** from the server list.
 - Backend ports: Set them to **80**. **ECS01** and **ECS02** will use this port to communicate with the load balancer.
5. Configure health check parameters. In this example, retain the default settings.
6. Click **Next: Confirm**, confirm the settings, and click **Submit**.

Step 6: Verify Load Balancing

After the load balancer is configured, you can access it using its domain name (www.example.com in this example) to check whether the load balancer can route requests across the two backend servers.

1. Modify the **C:\Windows\System32\drivers\etc\hosts** file on your PC to map the domain name to the EIP bound to the load balancer.
View the EIP on the **Summary** page of the load balancer.

Figure 1-6 hosts file on your PC

```
# localhost name resolution is handled within DNS itself.
# 127.0.0.1          localhost
# ::1                localhost

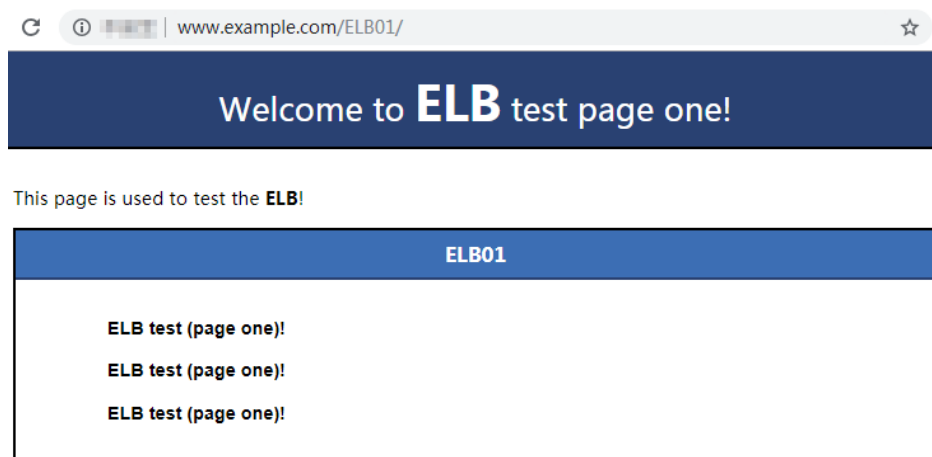
11[redacted]14 www.example.com
```

2. Choose **Start** and enter **cmd** to open the CLI.
3. Run the following command to check whether the domain name is mapped to the load balancer EIP:

```
ping www.example.com
```

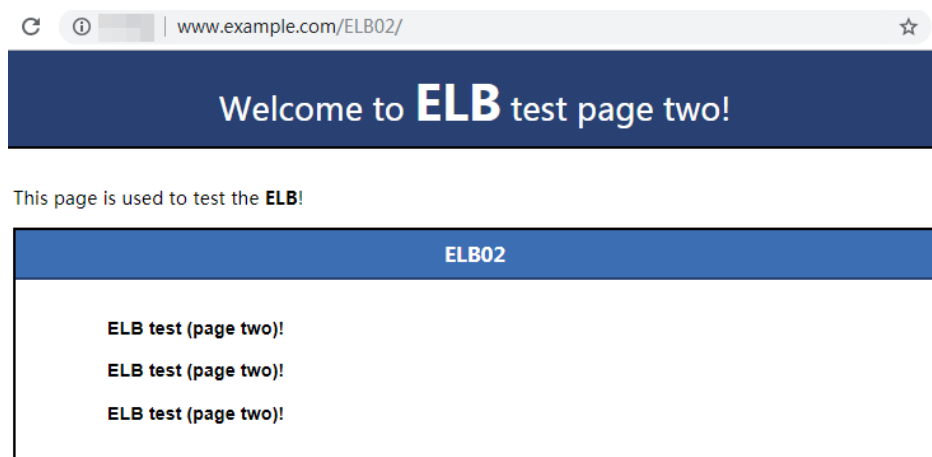
If data packets are returned, the domain name has been mapped to the load balancer EIP.
4. Use your browser to access **http://www.example.com**.
If the following page is displayed, the load balancer has routed the request to **ECS01**.

Figure 1-7 Accessing ECS01



5. Use your browser to access **http://www.example.com** again. If the following page is displayed, the load balancer has routed the request to **ECS02**.

Figure 1-8 Accessing ECS02



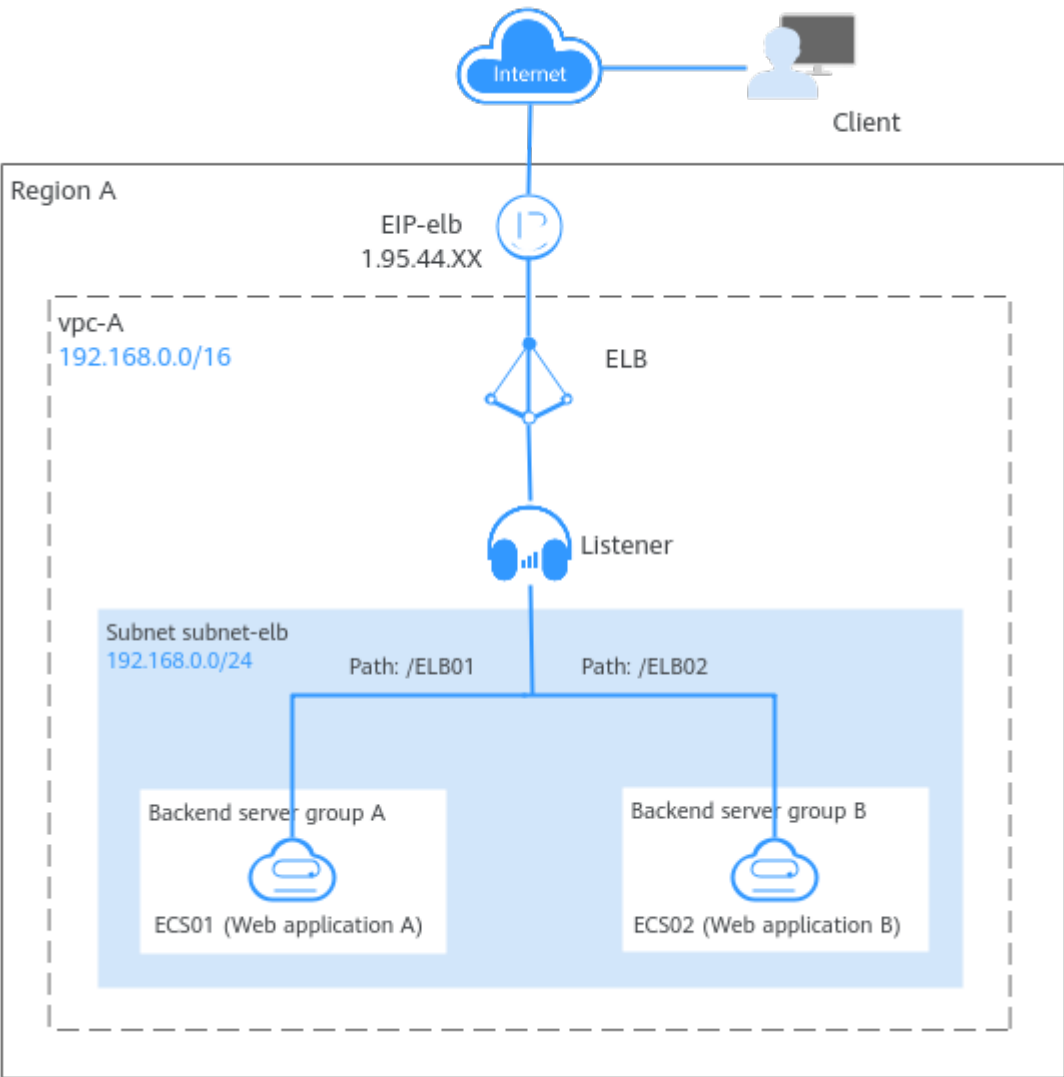
2 Using ELB to Distribute Traffic to Multiple Web Applications Across ECSs

Scenarios

To forward requests based on domain names and paths, you need to create a load balancer, add an HTTP or HTTPS listener, and add forwarding policies to specify the domain names and paths.

If you have two web applications that are deployed on two ECSs (**ECS01** and **ECS02** in this example) but use the same domain name for access, you can set different paths to process requests.

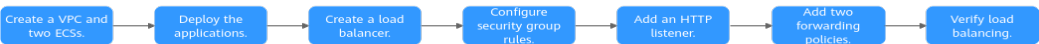
Figure 2-1 Using ELB to distribute traffic to two applications



Procedure

You can follow the process in [Figure 2-2](#) to use a load balancer to route requests to two web applications.

Figure 2-2 Routing requests to two web applications deployed in separated ECSs



Procedure	What to Do
Preparations	Before using cloud services, sign up for a HUAWEI ID, enable Huawei Cloud services, and complete real-name authentication.

Procedure	What to Do
Step 1: Create a VPC and Two ECSs	<ul style="list-style-type: none">• Create a VPC with an IPv4 CIDR block and create a subnet in the VPC.<ul style="list-style-type: none">– VPC IPv4 CIDR block: 192.168.0.0/16– Subnet IPv4 CIDR block: 192.168.0.0/24• Buy two ECSs in the VPC subnet you have created.
Step 2: Deploy the Applications	Deploy Nginx on the two ECSs.
Step 3: Create a Load Balancer	Create a load balancer with elastic specifications to receive requests from clients and distribute the requests to backend servers.
Step 4: Configure Security Group Rules	Configure security group rules to allow traffic from the backend subnet where the load balancer works to the backend servers.
Step 5: Add a Listener	Add an HTTP listener to the load balancer to check requests from clients and route requests to backend servers in the backend server group.
Step 6: Add Two Forwarding Policies	Configure two forwarding policies for the HTTP listener to enable the listener to forward requests to different backend server groups based on the configured domain name and path.
Step 7: Verify Load Balancing	Access the domain name of the load balancer to check whether it can route requests across the two backend servers.

Preparations

Before creating resources such as VPCs and ECSs, you need to [sign up for a HUAWEI ID and enable Huawei Cloud services](#).

If you already have a HUAWEI ID, skip this part.

Step 1: Create a VPC and Two ECSs

You need to plan the region for your load balancer, and create a VPC and two ECSs. Ensure that the ECSs and load balancer are in the same AZ and VPC.

1. Create a VPC.
 - VPC Configurations
 - Configuring a VPC

Parameter	Example Value	Description
Name	vpc-A	The VPC name. Set it to vpc-A .
IPv4 CIDR Block	192.168.0.0/16	The IPv4 CIDR block of vpc-A .
Enterprise Project	default	The enterprise project by which VPCs are centrally managed. Select an existing enterprise project for vpc-A .
Advanced Settings (Optional)	-	In this example, retain the default values.

- Configuring a VPC subnet

Parameter	Example Value	Description
AZ	AZ1	A geographic location with independent power supply and network facilities in a region. Each region contains multiple AZs. AZs are physically isolated but connected through an internal network. Subnets of a VPC can be located in different AZs without affecting communications. You can select any AZ in a region. The ECSs and VPC can be in different AZs. For example, you can select AZ1 for the ECS and AZ3 for the VPC subnet.
Subnet Name	subnet-elb	The subnet name. Set it to subnet-elb .
IPv4 CIDR Block	192.168.0.0/24	The IPv4 CIDR block of subnet-elb , which is a unique CIDR block with a range of IP addresses in vpc-A .
IPv6 CIDR Block (Optional)	Do not enable	Whether to assign IPv6 addresses.
Associated Route Table	Default	The default route table that subnet-elb is associated with. The default route table has a preset system route that allows subnets in a VPC to communicate with each other.

Parameter	Example Value	Description
Advanced Settings (Optional)	-	In this example, retain the default values.

2. Create two ECSs.

ECS Configurations

- Configuring the network parameters for the two ECSs

Parameter	Example Value	Description
ECS Name	<ul style="list-style-type: none">• ECS01• ECS02	Names of each ECS. Set them to ECS01 and ECS02 .
AZ	<ul style="list-style-type: none">• AZ1• AZ2	The AZ of each ECS. Select different AZs for the two ECSs.
Network	<ul style="list-style-type: none">• vpc-A• subnet-elb	The VPC and subnet where the ECSs work. Select vpc-A and subnet-elb for the two ECSs.
EIP	<ul style="list-style-type: none">• EIP01• EIP02	The EIP bound to each ECS for Internet access.

Step 2: Deploy the Applications

Deploy Nginx on the two ECSs and edit two HTML pages so that a page with message "Welcome to ELB test page one!" is returned when **ECS01** is accessed, and the other page with message "Welcome to ELB test page two!" is returned when **ECS02** is accessed.

1. [Log in to the ECSs.](#)
2. Install and start Nginx.

In this example, the two ECSs use CentOS 7.6 as the operating system.

Deploying Nginx

- a. Run the **wget** command to download the Nginx installation package for your operating system in use.

```
wget http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7.ngx.noarch.rpm
```
- b. Run the following command to create the Nginx yum repository. CentOS 7.6 is used as an example here.

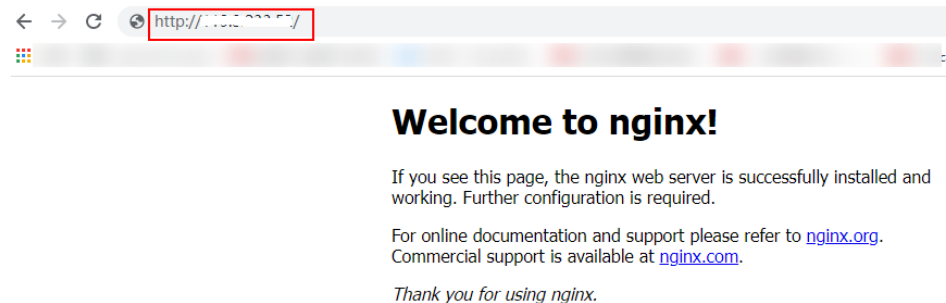
```
rpm -ivh nginx-release-centos-7-0.el7.ngx.noarch.rpm
```
- c. Run the following command to install Nginx:

```
yum -y install nginx
```
- d. Run the following commands to start Nginx and configure automatic Nginx enabling upon ECS startup:

```
systemctl start nginx
systemctl enable nginx
```

- e. Enter **http://EIP bound to any ECS** in the address box of your browser. If the following page is displayed, Nginx has been installed.

Figure 2-3 Nginx installed successfully



3. Modify the HTML page of **ECS01**.

Move the **index.html** file from the default root directory of Nginx **/usr/share/nginx/html** to the **ELB01** directory and modify the file to identify access to ECS01.

- a. Create the **ELB01** directory and copy the **index.html** file to this directory:

```
mkdir /usr/share/nginx/html/ELB01
cp /usr/share/nginx/html/index.html /usr/share/nginx/html/ELB01/
```

- b. Run the following command to open the **index.html** file:

```
vim /usr/share/nginx/html/ELB01/index.html
```

- c. Press **i** to enter editing mode.

- d. Modify the **index.html** file.

The following is the content to be modified:

```
...
<body>
  <h1>Welcome to <strong>ELB</strong> test page one!</h1>

  <div class="content">
    <p>This page is used to test the <strong>ELB</strong>!</p>

    <div class="alert">
      <h2>ELB01</h2>
      <div class="content">
        <p><strong>ELB test (page one)!</strong></p>
        <p><strong>ELB test (page one)!</strong></p>
        <p><strong>ELB test (page one)!</strong></p>
      </div>
    </div>
  </div>
</body>
```

- e. Press **Esc** to exit the editing mode. Then, enter **:wq** to save the settings and exit the file.

4. Modify the HTML page of ECS02 by referring to step 3 to identify the access to ECS02.

Modifying the HTML Page of ECS02

- a. Create the **ELB02** directory and copy the **index.html** file to this directory:

```
mkdir /usr/share/nginx/html/ELB02
cp /usr/share/nginx/html/index.html /usr/share/nginx/html/ELB02/
```

- b. Run the following command to open the **index.html** file:

```
vim /usr/share/nginx/html/ELB02/index.html
```

- c. Press **i** to enter editing mode.
- d. Modify the **index.html** file.

The following is the content to be modified:

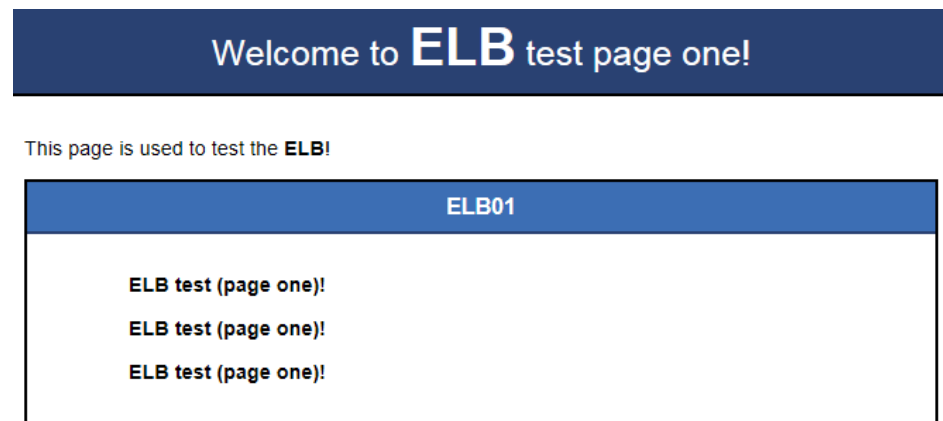
```
...
<body>
  <h1>Welcome to <strong>ELB</strong> test page two!</h1>

  <div class="content">
    <p>This page is used to test the <strong>ELB</strong>!</p>

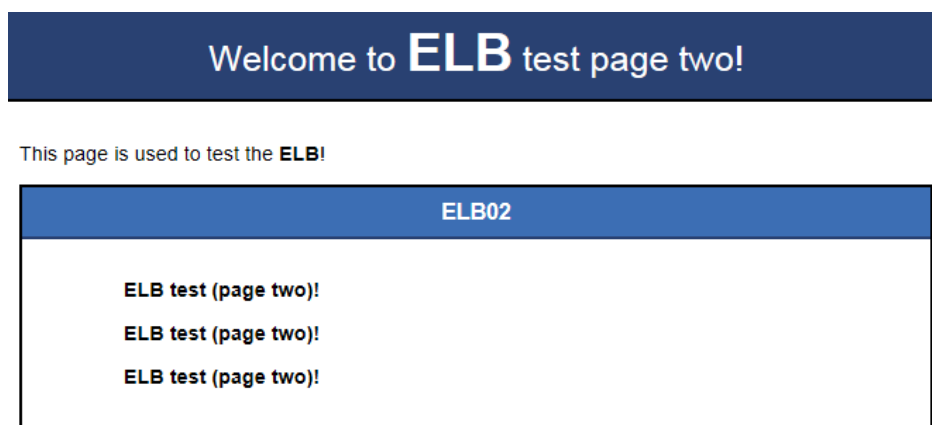
    <div class="alert">
      <h2>ELB02</h2>
      <div class="content">
        <p><strong>ELB test (page two)!</strong></p>
        <p><strong>ELB test (page two)!</strong></p>
        <p><strong>ELB test (page two)!</strong></p>
      </div>
    </div>
  </div>
</body>
```

- e. Press **Esc** to exit the editing mode. Then, enter **:wq** to save the settings and exit the file.
5. Use your browser to access **http://EIP bound to ECS01/ELB01/** and **http://EIP bound to ECS02/ELB02/** in sequence to verify that Nginx has been deployed. If the modified HTML pages are displayed, Nginx has been deployed.
- HTML page of **ECS01**

Figure 2-4 Nginx successfully deployed on ECS01



- HTML page of **ECS02**

Figure 2-5 Nginx successfully deployed on ECS02

Step 3: Create a Load Balancer

In this example, the load balancer needs an EIP to forward Internet requests to the application deployed on the ECSs. You can determine whether to bind an EIP to the load balancer based on service requirements.

1. Go to the [Buy Elastic Load Balancer](#) page.
2. On the displayed page, set the parameters as required.

Basic Configurations

- Configuring the basic information

Parameter	Example Value	Description
Type	Dedicated load balancer	Specifies the type of the load balancer. Select Dedicated load balancer . A dedicated load balancer uses dedicated resources. Its performance is not affected by other load balancers.
Region	-	Specifies the desired region. Resources in different regions cannot communicate with each other over internal networks. Select the same region as the ECSs.
AZ	<ul style="list-style-type: none">• AZ1• AZ2	Specifies the AZs of the load balancer. Select multiple AZs if you need DR capability. The load balancer performance multiplies as the number of AZs increases.
Name	elb-test	Specifies the name of the load balancer.
Enterprise Project	default	Specifies an enterprise project by which cloud resources and members are centrally managed.


3. Set **Specification Type** to **Elastic**.

4. Configure the network parameters and EIP information.

Network Configurations

– Configuring network parameters

Parameter	Example Value	Description
Network Type	Private IPv4 network	Specifies the network where the load balancer works. In this example, select Private IPv4 network . The load balancer uses the private IPv4 address to process private network requests. To let the load balancer route requests over the Internet, bind an EIP to it.
VPC	-	Specifies the VPC where the load balancer works. In this example, select vpc-A .
Frontend Subnet	-	Specifies the frontend subnet from which an IPv4 address will be assigned to the load balancer to receive client requests. If IPv6 is enabled, an IPv6 address will also be assigned to the load balancer.
IPv4 Address	Automatically assign IP address	Specifies how you want the IPv4 address to be assigned.
Backend Subnet	Subnet of the load balancer	Specifies the backend subnet from which IP addresses will be assigned to the load balancer to forward requests to backend servers.
IP as a Backend	-	Specifies whether to add IP addresses as backend servers that are not in the VPC of the load balancer. In this example, leave this feature disabled.
Internet Access		
EIP	Auto assign	Specifies the EIP that will be bound to the load balancer for receiving and forwarding requests over the Internet.
EIP Type	Dynamic BGP	Specifies the link type (BGP) when a new EIP is used.
Billed By	Traffic	Specifies how the bandwidth will be billed. In this example, select Traffic . You need to specify a maximum bandwidth and pay for the outbound traffic you use.
Bandwidth (Mbit/s)	100	Specify the maximum bandwidth.

5. Specifies the maximum bandwidth. Click  to expand the advanced settings and add a description and tags to the load balancer.
6. Click **Buy Now**.
7. Confirm the configuration and submit your request.
8. View the newly created load balancer in the load balancer list.

Step 4: Configure Security Group Rules

To ensure normal communications between the load balancer and backend servers, you need to check the security group and network ACL rules.

- The security groups configured for backend servers must have inbound rules to allow health check and service traffic from backend subnet of the load balancer to backend servers. By default, the backend subnet of a load balancer is the same as the subnet where the load balancer works.
- Network ACL rules are optional for subnets. If network ACL rules are configured for the subnet where backend servers are running, the inbound rules must allow traffic from the backend subnet of the load balancer to the subnet of backend servers.

For details about how to configure security group and network ACL rules, see [Security Group and Network ACL Rules](#).

You can configure security group rules based on [Table 2-1](#).

Table 2-1 Security group rules

Direction	Action	Type	Protocol & Port	Source/Destination	Description
Inbound	Allow	IPv4	TCP: 80	Source: 192.168.0.0/24	Allows outbound traffic to ECSs in the security group.
Outbound	Allow	IPv4	All	Destination: 0.0.0.0/0	Allows the ECSs in the security group to access the external networks.

Step 5: Add a Listener

Add a listener to the created load balancer. When you add the listener, create a backend server group, configure a health check, and add the two ECSs to this backend server group. If a backend server is detected unhealthy, the load balancer will stop routing traffic to it until the backend server recovers.

1. On the load balancer list page, locate load balancer **elb-test** and click its name.
2. On the **Listeners** tab, click **Add Listener** and configure parameters based on [Table 2-2](#).

Table 2-2 Parameters for configuring a listener

Parameter	Example Value	Description
Name	listener	Specifies the listener name.
Frontend Protocol	HTTP	Specifies the protocol that will be used by the listener to receive requests from clients.
Frontend Port	80	Specifies the port that will be used by the listener to receive requests from clients.
Redirect	-	Specifies whether to redirect requests from the HTTP listener to an HTTPS listener. Disable it in this example.
Access Control	All IP addresses	Specifies how access to the listener is controlled.
Transfer Client IP Address	-	Specifies whether to transmit IP addresses of the clients to backend servers. This feature is enabled for dedicated load balancers by default and cannot be disabled.
Advanced Forwarding	-	Specifies whether to enable advanced forwarding. Once it is enabled, more forwarding rules and actions are supported. Enable it in this example.

3. Retain the default values for **Advanced Settings (Optional)**. Click **Next: Configure Request Routing Policy**. On the displayed page, select **Create new** for **Backend Server Group**.
Set **Load Balancing Algorithm** to **Weighted round robin** and retain the default values for other parameters.
4. Click **Next: Add Backend Server**.
 - Backend servers: Click **Add Cloud Server** and select **ECS01** and **ECS02** from the server list.
 - Backend ports: Set them to **80**. **ECS01** and **ECS02** will use this port to communicate with the load balancer.
5. Configure health check parameters. In this example, retain the default settings.
6. Click **Next: Confirm**, confirm the settings, and click **Submit**.

Step 6: Add Two Forwarding Policies

The following describes how to configure forwarding policies to forward HTTP requests to the two ECSs, for example, requests from **www.example.com/ELB01/** to **ECS01** and **www.example.com/ELB02/** to **ECS02**.

1. On the listener list page, locate the listener you have added in the previous step and click **Add/Edit Forwarding Policy** in the **Forwarding Policies** column. In the displayed page, click **Add Forwarding Policy**.

Figure 2-6 Configuring a forwarding policy

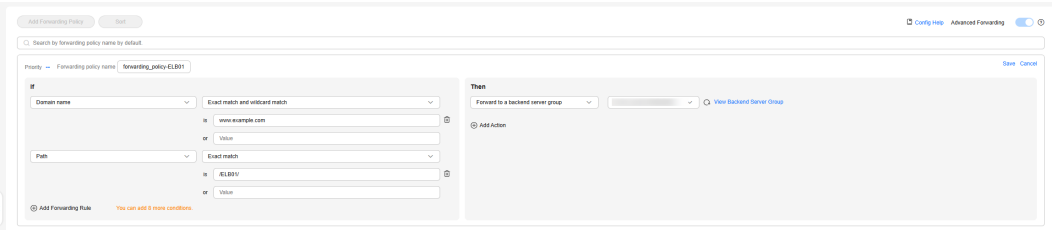


Table 2-3 An example forwarding policy

Forwarding Policy Item	Setting
Forwarding policy name	Enter a forwarding policy name, for example, forwarding_policy-ELB01 .
Forwarding rule	Domain name: Enter a domain name that will be used to forward requests, for example, www.example.com . Path: Specify a path (Exact match) to forward the requests, for example, /ELB01/ .
Action	Select Forward to a backend server group .

2. Create a backend server group for forwarding policy **forwarding_policy-ELB01**.
- a. Select **Create Backend Server Group** from the drop-down list to the right of **Forward to a backend server group**.

b. In the **Configure Routing Policy** step, set the backend server group name to **server_group-ELB01**.
Use default settings for other parameters.

c. Click **Next**. In the **Add Backend Server** step, click **Add Cloud Server**.
3. On the displayed page, select cloud server **ECS01**, set the backend port to **80**, and click **Finish**.
4. Repeat **1** to **3** to add another forwarding policy, create a backend server group, and add **ECS02** to its backend server group.

Step 7: Verify Load Balancing

After the load balancer is configured, you can access it using the domain name or specified path to check whether the load balancer can route requests across the two backend servers.

1. Modify the **C:\Windows\System32\drivers\etc\hosts** file on your PC to map the domain name to the EIP bound to the load balancer.
View the EIP on the **Summary** tab of the load balancer.

Figure 2-7 hosts file on your PC

```
# localhost name resolution is handled within DNS itself.
# 127.0.0.1      localhost
# ::1           localhost

11[REDACTED]14 www.example.com
```

2. Choose **Start** and enter **cmd** to open the CLI.
3. Run the following command to check whether the domain name is mapped to the load balancer EIP:
ping www.example.com
If data packets are returned, the domain name has been mapped to the load balancer EIP.
4. Use your browser to access **http://www.example.com/ELB01/**.
If the following page is displayed, the load balancer has routed the request to **ECS01**.

Figure 2-8 Accessing ECS01**NOTE**

ELB01/ indicates that the default directory named **ECS01** is accessed, while **ELB01** indicates the file name. This means the slash (/) following **ELB01** must be retained.

5. Use your browser to access **http://www.example.com/ELB02/**.
If the following page is displayed, the load balancer has routed the request to **ECS02**.

Figure 2-9 Accessing ECS02

