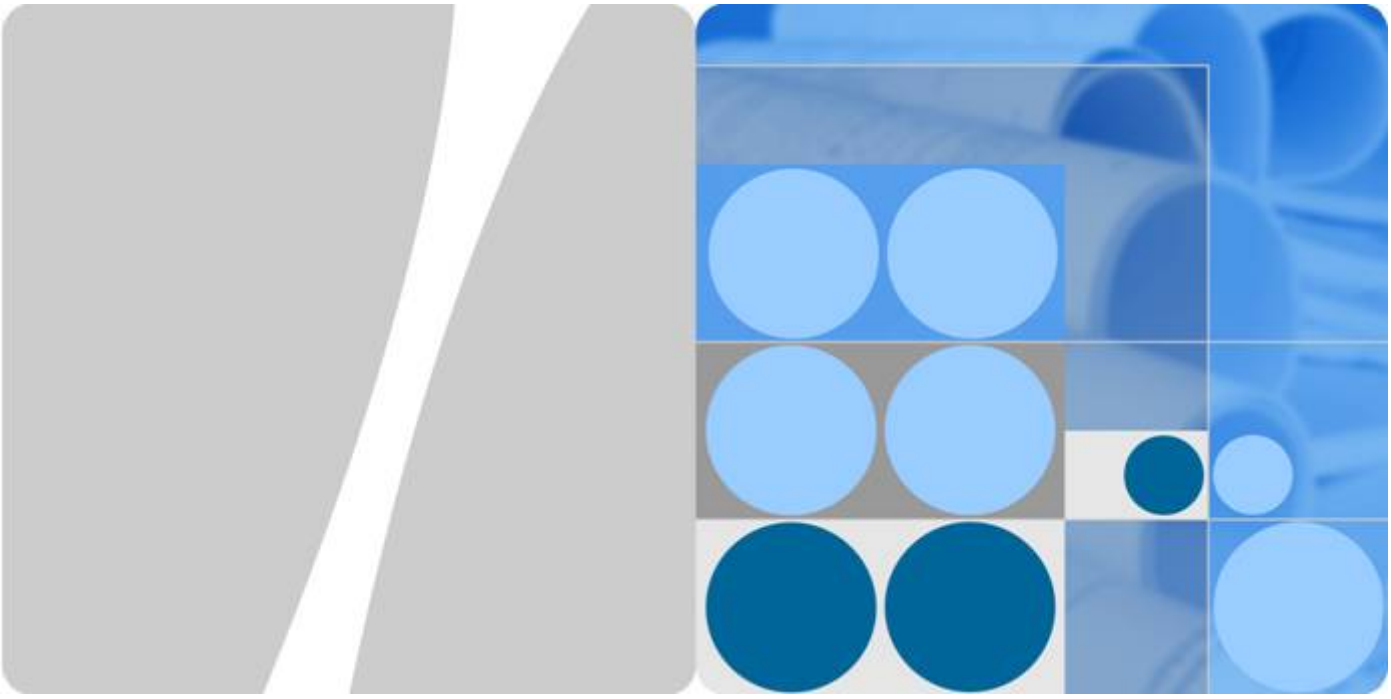


Elastic Load Balance

Getting Started

Date **2020-05-30**

Contents

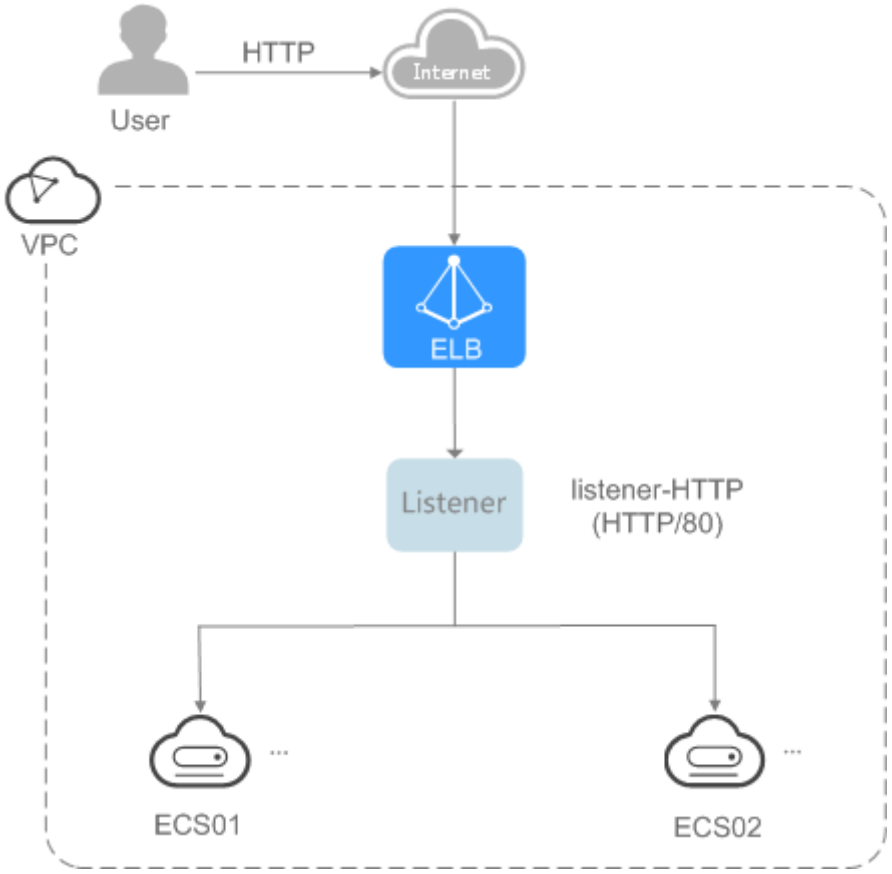
1 Overview....1
2 Using Shared Load Balancers — Entry Level....3
3 Using Shared Load Balancers — Advanced Level....13
4 Change History....26

1 Overview

This chapter gives examples that show how you can quickly create a shared load balancer to distribute incoming traffic across two backend servers.

- **Entry level**: A large number of requests need to be routed to backend servers. In addition, health checks are required to monitor the running statuses of backend servers and route incoming traffic only to healthy servers to eliminate SPOFs and improve service availability.

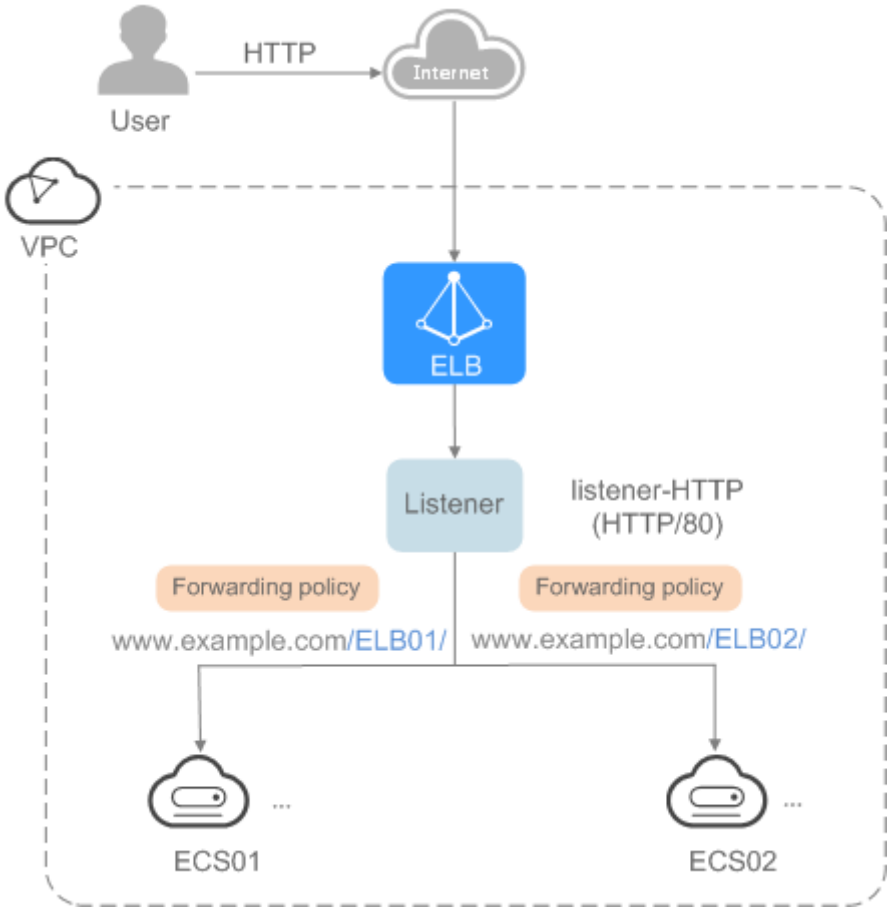
Figure 1-1 Entry level



As the incoming traffic increases, you can add more servers to better balance the load of your application.

- **Advanced level**: Two or more applications use one domain name to provide services, and requests to a specific application are routed based on the URL. Forwarding policies are required to forward requests from different URLs to the corresponding backend server groups.

Figure 1-2 Advanced level



As the incoming traffic increases, you can add more backend servers to the two backend server groups. In addition, you can configure health checks to monitor the running statuses of backend servers and route incoming traffic only to healthy servers to eliminate SPOFs and improve service availability.

2 Using Shared Load Balancers — Entry Level

Scenarios

You have a web application deployed on two ECSs, and the application often needs to handle a heavy load.

You can create a shared load balancer to distribute traffic evenly across the two ECSs, which not only makes your application more available, but also eliminates SPOFs.

Prerequisites

- You have assigned an EIP for each ECS.
- You have added security group rules to allow traffic from the ports used by the two ECSs. (You can enable all ports first and then disable the ports that are not used after service deployment.) For details, see **Configuring Security Group Rules**.
- The security group containing the two ECSs allows traffic from 100.125.0.0/16. (ELB uses these IP addresses to perform health checks and route requests to backend servers.)

Create ECSs

To create two ECSs, perform the following operations:

1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.

   For example, select **CN North-Beijing1** as the region.
3. Click **Service List**. Under **Computing**, click **Elastic Cloud Server**.
4. Click **Buy ECS**. Set the parameters and click **Next**.

   The following table lists the specifications of the two ECSs.

Table 2-1 ECS specifications

Item	Example Value
Name	ECS01 and ECS02
OS	CentOS 7.2 64bit
vCPUs	2
Memory	4 GB
System disk	40 GB
Data disk	100 GB
Bandwidth	5 Mbit/s
Region	CN North-Beijing1

5. Submit your request.

Deploy the Application

Deploy Nginx on the two ECSs and edit two HTML pages for the web application 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 Nginx.
   a. Run the following command to install Nginx:
      yum -y install nginx
   b. Run the following command to start Nginx:
      systemctl start nginx.service
   c. Enter **http://***ECS public IP address* in the address box of your browser.
      If the following page is displayed, Nginx has been installed.

Figure 2-1 Nginx installed successfully



3. Modify the HTML page of ECS01.

   The default root directory of Nginx is **/usr/share/nginx/html**. Modify the **index.html** file to identify access to 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 to be as follows:

```
...
  <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 editing mode. Then, enter **:wq** to save the settings and exit the file.

4. Modify the HTML page of ECS02.

   The default root directory of Nginx is **/usr/share/nginx/html**. Modify the **index.html** file to identify access to 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 to be as follows:

```
...
  <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>
```

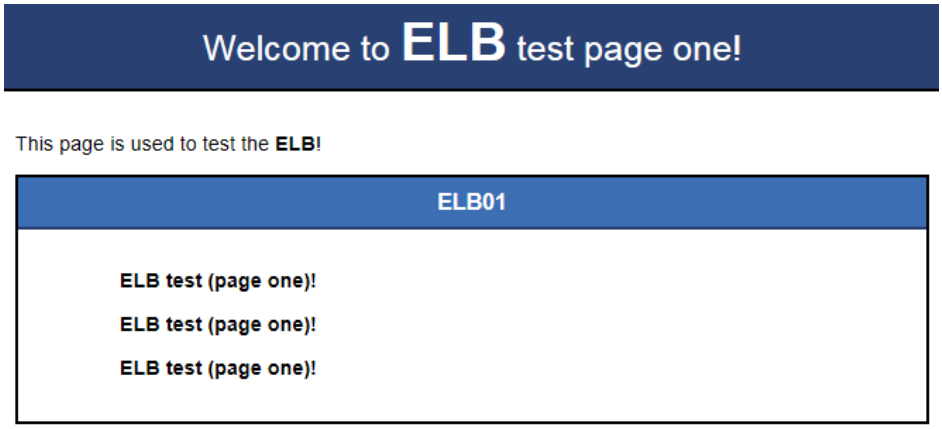
   d. Press **Esc** to exit editing mode. Then, enter **:wq** to save the settings and exit the file.

5. Use your browser to access **http://***ECS01 EIP* and **http://***ECS02 EIP* to verify the Nginx service.

   If the modified HTML pages are displayed, Nginx has been deployed.

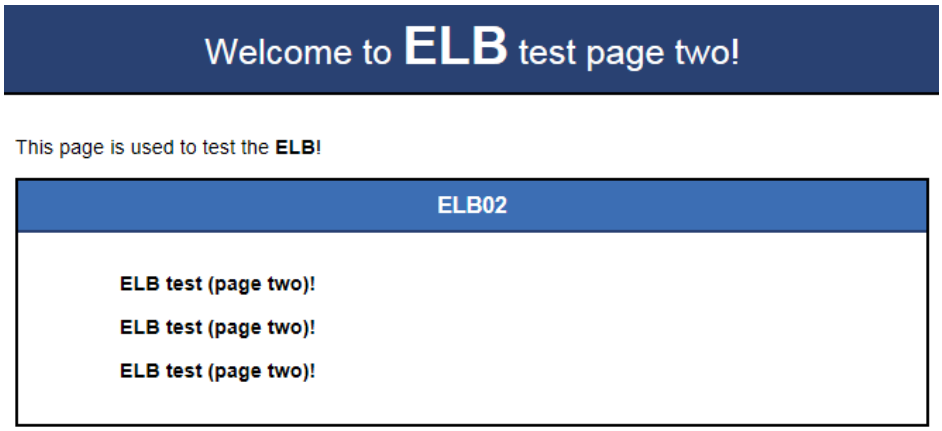
   – HTML page of ECS01

Figure 2-2 Nginx successfully deployed on ECS01



- HTML page of ECS02

Figure 2-3 Nginx successfully deployed on ECS02

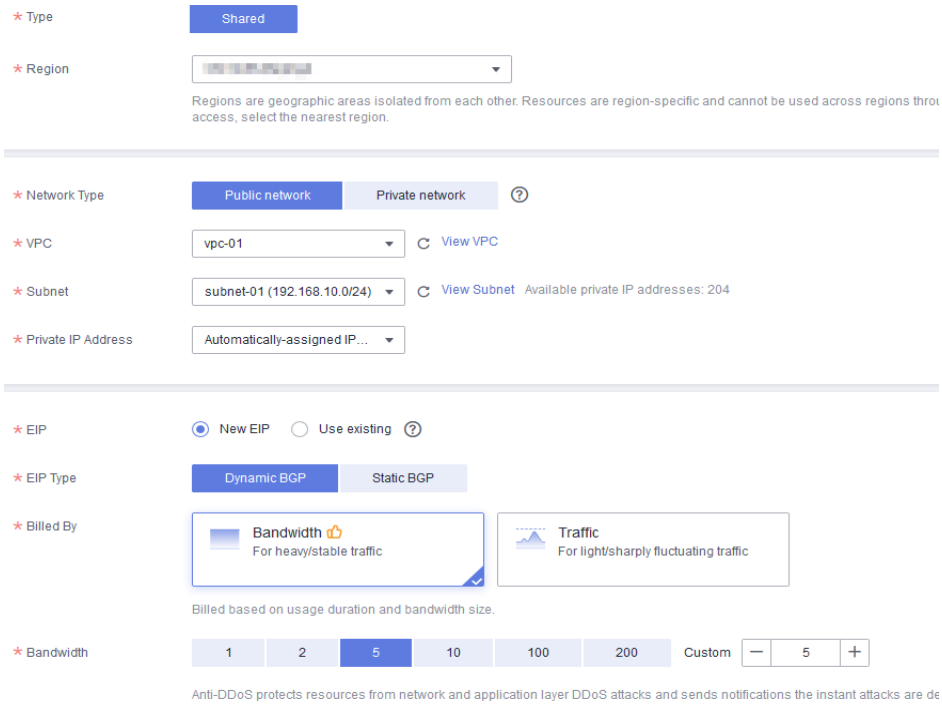


Create a Load Balancer

1. In the upper left corner of the page, click and select the desired region and project.

   Select **CN-North-Beijing1** because the load balancer must be in the same region as the two ECSs.
2. Click **Service List**. Under **Network**, click **Elastic Load Balance**.
3. Click **Buy Elastic Load Balancer** and then set the parameters.

Figure 2-4 Buy Elastic Load Balancer

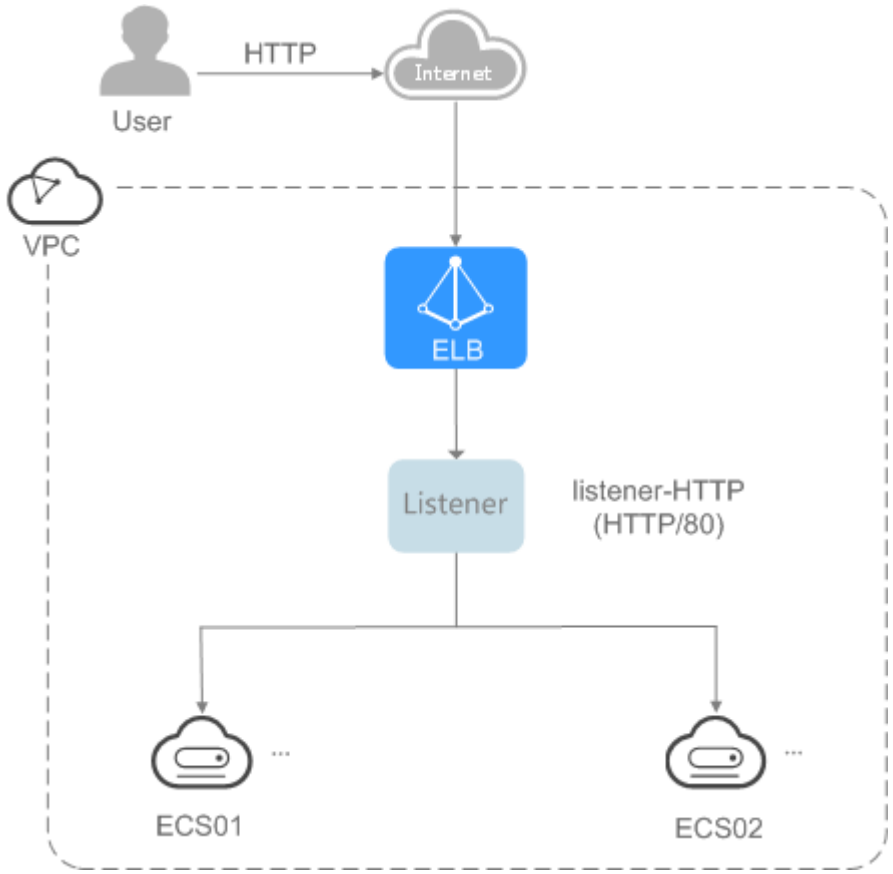


4. Click **Next**.
5. Confirm the configuration and submit your request.
6. View the newly created load balancer (**elb-01**) in the load balancer list.

Add a Listener

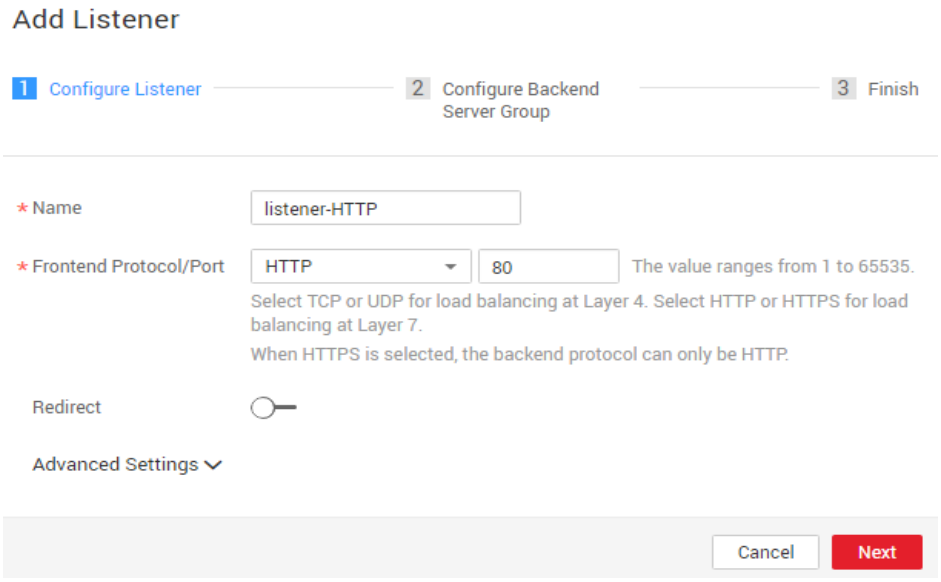
Add a listener to the created load balancer. When you add the listener, create a backend server group, configure a health check, and add the two ECSs to the created backend server group.

Figure 2-5 Traffic forwarding



1. Click **Service List**. Under **Network**, click **Elastic Load Balance**.
2. Locate the created load balancer (**elb-01**) and click its name.
3. Under **Listeners**, click **Add Listener**.
4. Configure the listener and click **Next**.
   - **Name**: Enter a name, for example, **listener-HTTP**.
   - **Frontend Protocol/Port**: Select a protocol and enter a port for the load balancer to receive requests. For example, set it to **HTTP** and **80**.

Figure 2-6 Configuring a listener



5. Create a backend server group and configure a health check.
   - Backend server group
     - **Name**: Enter a name, for example, **server_group-ELB**.
     - **Load Balancing Algorithm**: Select an algorithm that the load balancer will use to route requests, for example, **Weighted round robin**.
   - Health check
     - **Protocol**: Select a protocol for the load balancer to perform health checks on backend servers. If the load balancer uses TCP, HTTP, or HTTPS to receive requests, the health check protocol can be TCP or HTTP. Set it to **HTTP**. Note that this setting cannot be changed after the listener is added.
     - **Domain Name**: Enter a domain name that will be used for health checks, for example, **www.example.com**.
     - **Port**: Enter a port for the load balancer to perform health checks on backend servers, for example, **80**.

Figure 2-7 Creating a backend server group

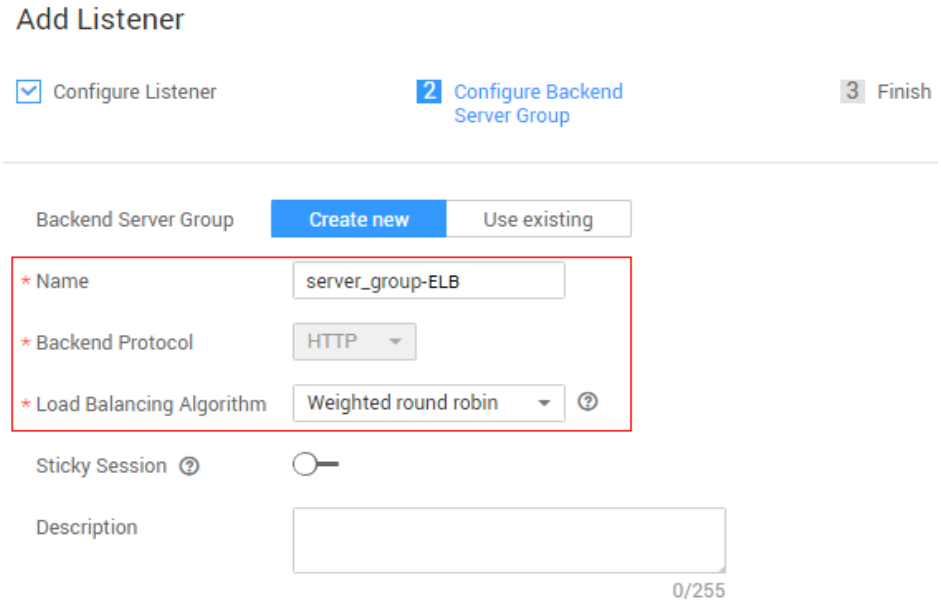
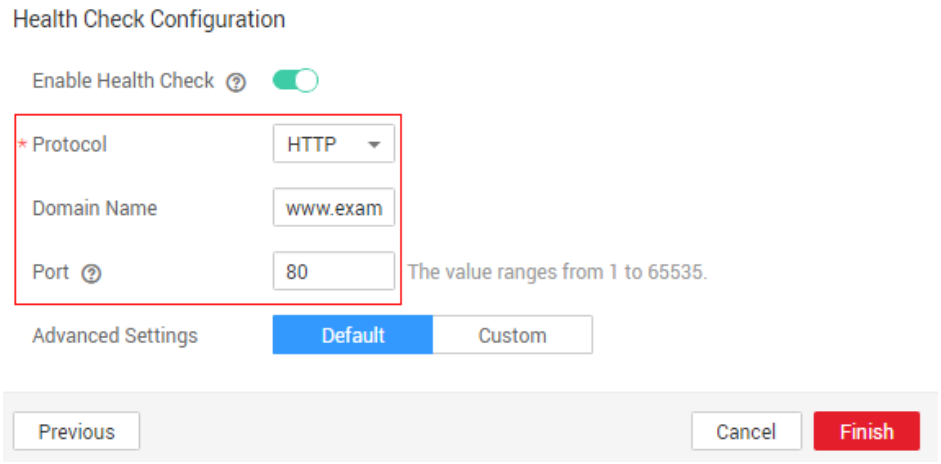
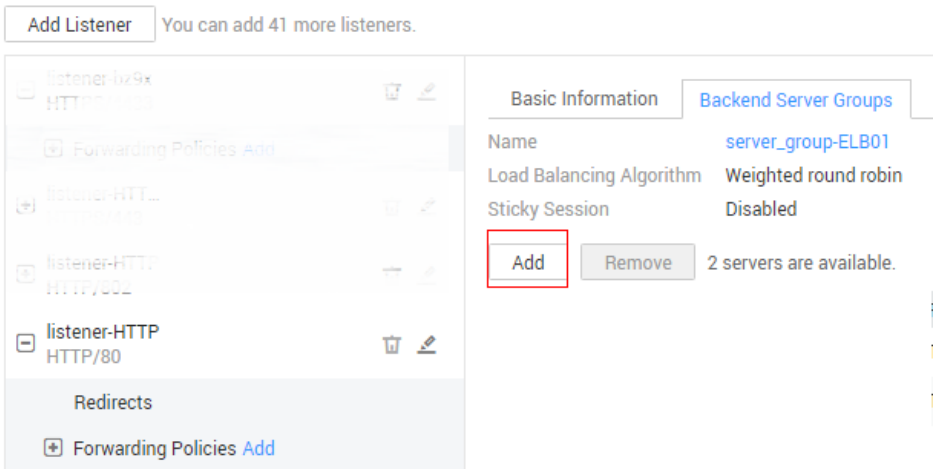


Figure 2-8 Configuring a health check



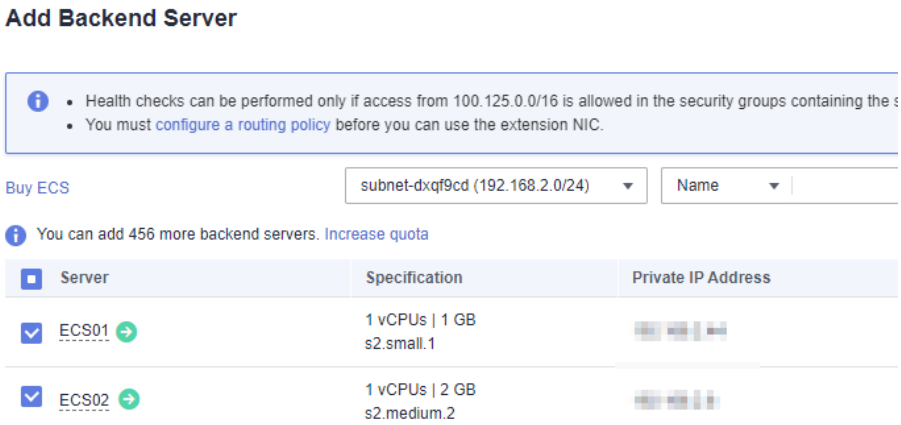
6. Click the name of the newly added listener. On the **Backend Server Groups** tab page on the right, click **Add**.

Figure 2-9 Adding backend servers



7. Select the servers you want to add, set the backend port, and click **Finish**.
   - Backend servers: **ECS01** and **ECS02**
   - Backend port: Set it to **80**. Backend servers will use this port to communicate with the load balancer.

Figure 2-10 Adding backend servers



Verify Load Balancing

After the load balancer is configured, you can access the corresponding domain name to check whether the two ECSs are accessible.

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

   View the load balancer EIP on the basic information page of the load balancer.

Figure 2-11 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
```

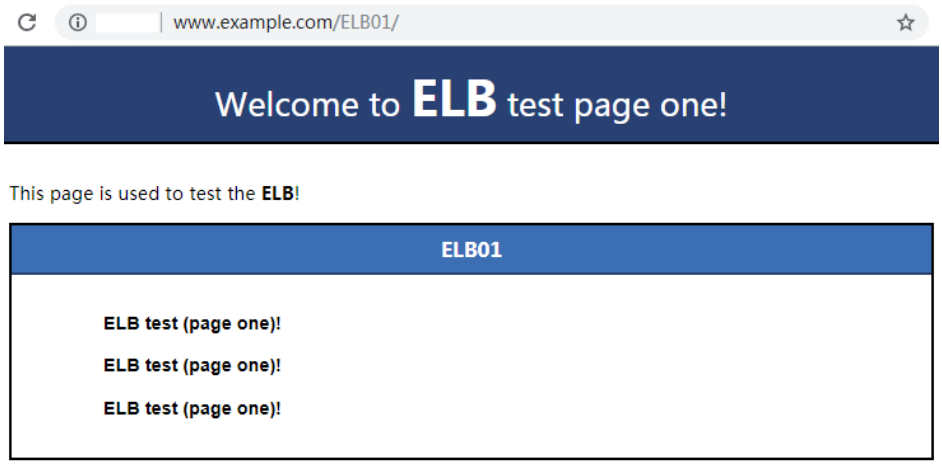
2. On the CLI of your PC, run the following command to check whether the domain name is mapped to the load balancer EIP:

   ping www.example.com

   If data packets are returned, the domain name has been mapped to the load balancer EIP.

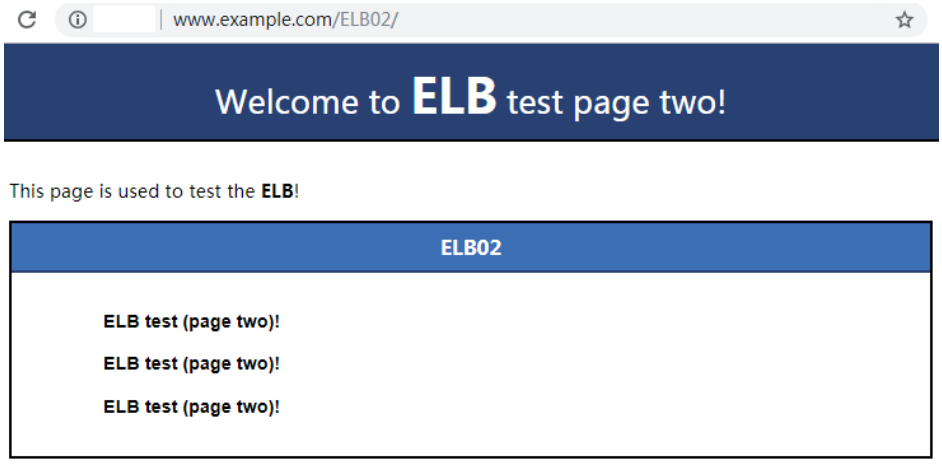
3. Use your browser to access **http://www.example.com**. If the following page is displayed, the load balancer has routed the request to ECS01.

Figure 2-12 Accessing ECS01



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

Figure 2-13 Accessing ECS02



3 Using Shared Load Balancers — Advanced Level

Scenarios

You have two web applications deployed on separated ECSs but use the same domain name for access. Two different URLs are required to process requests.

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

An HTTP listener is used as an example to describe how to route requests from two URLs (**/ELB01** and **/ELB02**) of the same domain name (www.example.com).

Prerequisites

- You have assigned an EIP for each ECS.
- You have added security group rules to allow traffic from the ports used by the two ECSs. (You can enable all ports first and then disable the ports that are not used after service deployment.) For details, see **Configuring Security Group Rules**.
- The security group containing the two ECSs allows traffic from 100.125.0.0/16. (ELB uses these IP addresses to perform health checks and route requests to backend servers.)

Create ECSs

To create two ECSs, perform the following operations:

1. Log in to the management console.
2. In the upper left corner of the page, click  and select the desired region and project.

   For example, select **CN North-Beijing1** as the region.
3. Click **Service List**. Under **Computing**, click **Elastic Cloud Server**.
4. Click **Buy ECS**. Set the parameters and click **Next**.
5. The following table lists the specifications of the two ECSs.

Table 3-1 ECS specifications

Item	Example Value
Name	ECS01 and ECS02
OS	CentOS 7.2 64bit
vCPUs	2
Memory	4 GB
System disk	40 GB
Data disk	100 GB
Bandwidth	5 Mbit/s
Region	CN North-Beijing1

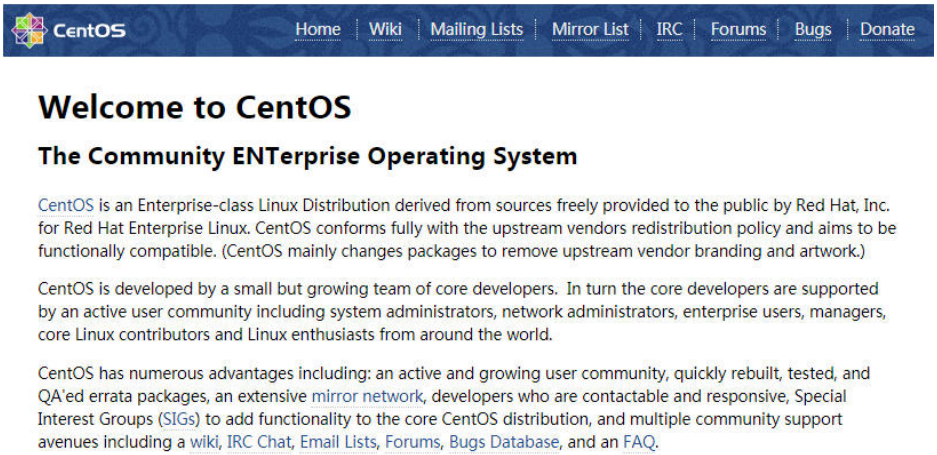
6. Click **Submit**.

Deploy the Application

Deploy Nginx on the two ECSs and edit two HTML pages for the web applications 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 Nginx.
   a. Run the following command to install Nginx:
      yum -y install nginx
   b. Run the following command to start Nginx:
      systemctl start nginx.service
   c. Enter **http://***ECS public IP address* in the address box of your browser.
      If the following page is displayed, Nginx has been installed.

Figure 3-1 Nginx installed successfully



3. Modify the HTML page of ECS01.

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

   a. Run the following commands in sequence to 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 to be as follows:

```
...
  <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 editing mode. Then, enter **:wq** to save the settings and exit the file.

4. Modify the HTML page of ECS02.

   The default root directory of Nginx is **/usr/share/nginx/html**. Move the **index.html** file from **/usr/share/nginx/html** to the **ELB02** directory and modify the file to identify access to ECS02.

   a. Run the following commands in sequence to 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 to be as follows:

```
...
  <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>
```

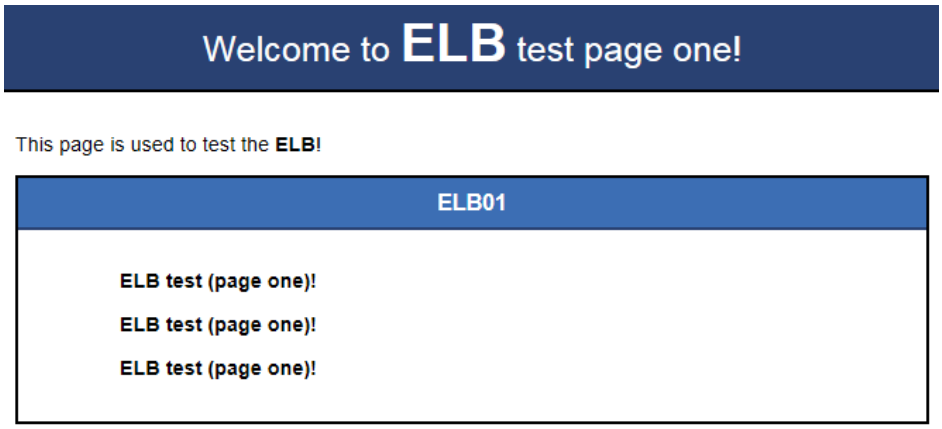
e. Press **Esc** to exit editing mode. Then, enter **:wq** to save the settings and exit the file.

5. Use your browser to access **http://***ECS01 EIP*/**ELB01/** and **http://***ECS02 EIP*/**ELB02/** to verify the Nginx service.

   If the modified HTML pages are displayed, Nginx has been deployed.

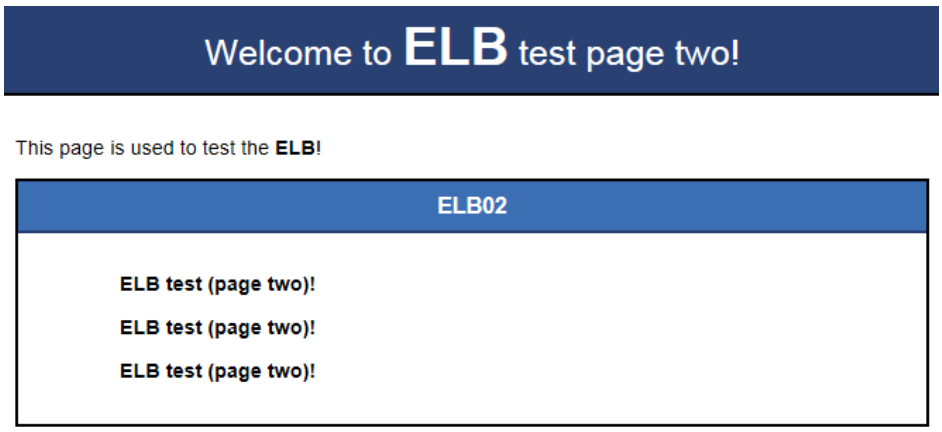
   – HTML page of ECS01

   Figure 3-2 Nginx successfully deployed on ECS01

   – HTML page of ECS02

   Figure 3-3 Nginx successfully deployed on ECS02

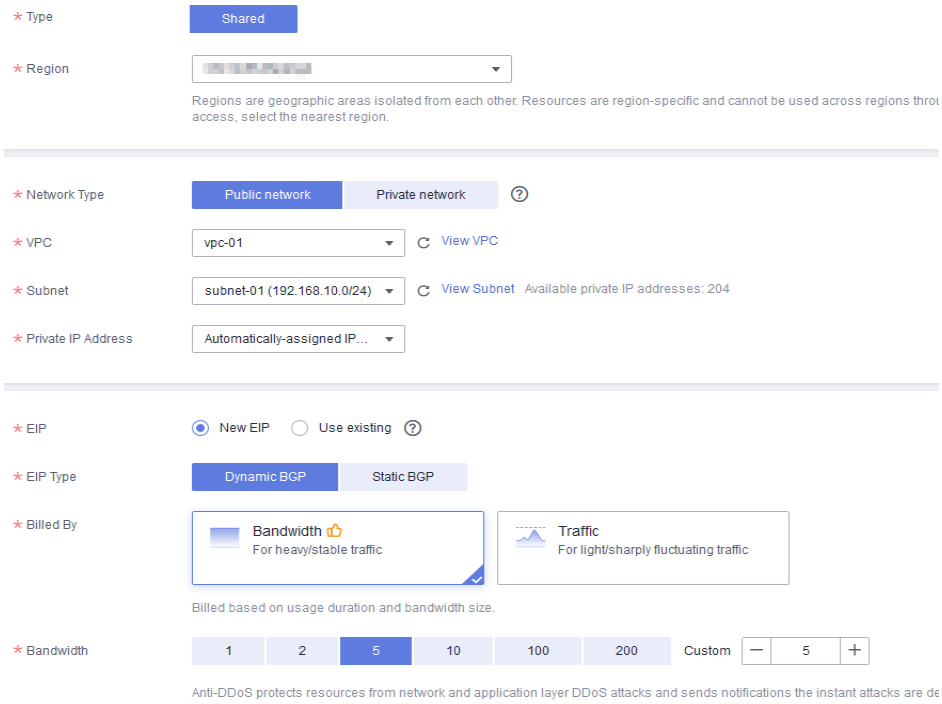
Create a Load Balancer

1. In the upper left corner of the page, click and select the desired region and project.

   Select **CN-North-Beijing1** as the region because the load balancer must be in the same region as the two ECSs.

2. Click **Service List**. Under **Network**, click **Elastic Load Balance**.
3. Click **Buy Elastic Load Balancer** and then set the parameters.

Figure 3-4 Buy Elastic Load Balancer



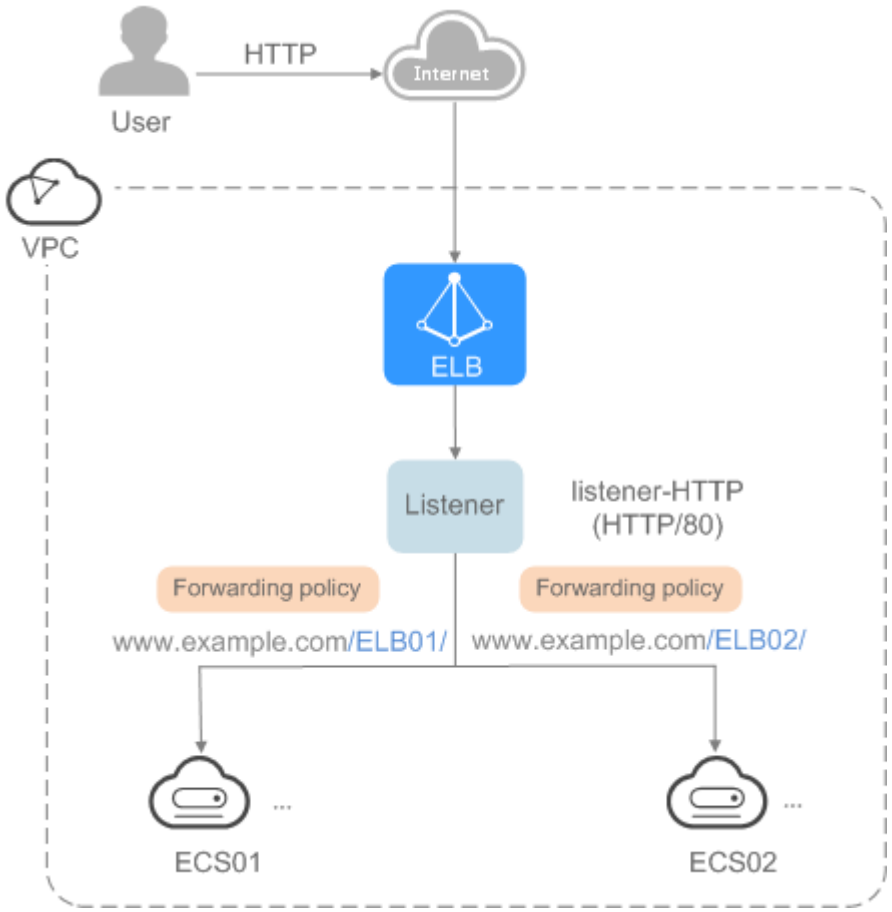
4. Click **Next**.
5. Confirm the configuration and click **Submit**.
6. View the newly created load balancer (**elb-01**) in the load balancer list.

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 the created backend server group.

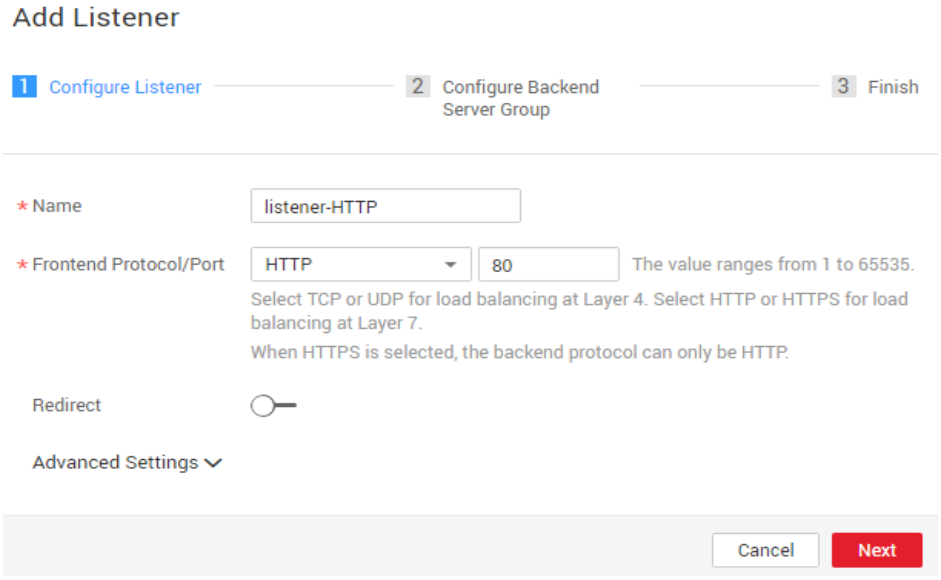
Configure two forwarding policies to forward HTTP requests to the two ECSs, one forwarding policy for each ECS. Foe example, requests from **www.example.com/ELB01/** are routed to ECS01, and those from **www.example.com/ELB02/** to ECS02.

Figure 3-5 Traffic forwarding



1. Click **Service List**. Under **Network**, click **Elastic Load Balance**.
2. Locate the created load balancer (**elb-01**) and click its name.
3. Under **Listeners**, click **Add Listener**.
4. Configure the listener and click **Next**.
   - **Name**: Enter a name, for example, **listener-HTTP**.
   - **Frontend Protocol/Port**: Select a protocol and enter a port for the load balancer to receive requests. For example, set it to **HTTP** and **80**.

Figure 3-6 Configuring a listener



5. Create a backend server group, configure a health check, and click **Finish**.
   - Backend server group
     - **Name**: Enter a name, for example, **server_group-ELB**.
     - **Load Balancing Algorithm**: Select an algorithm that the load balancer will use to route requests, for example, **Weighted round robin**.
   - Health check
     - **Protocol**: Select a protocol for the load balancer to perform health checks on backend servers. If the load balancer uses TCP, HTTP, or HTTPS to receive requests, the health check protocol can be TCP or HTTP. Set it to **HTTP**. Note that this setting cannot be changed after the listener is added.
     - **Domain Name**: Enter a domain name that will be used for health checks, for example: www.example.com.
     - **Port**: Enter a port for the load balancer to perform health checks on backend servers, for example, **80**.

Figure 3-7 Creating a backend server group

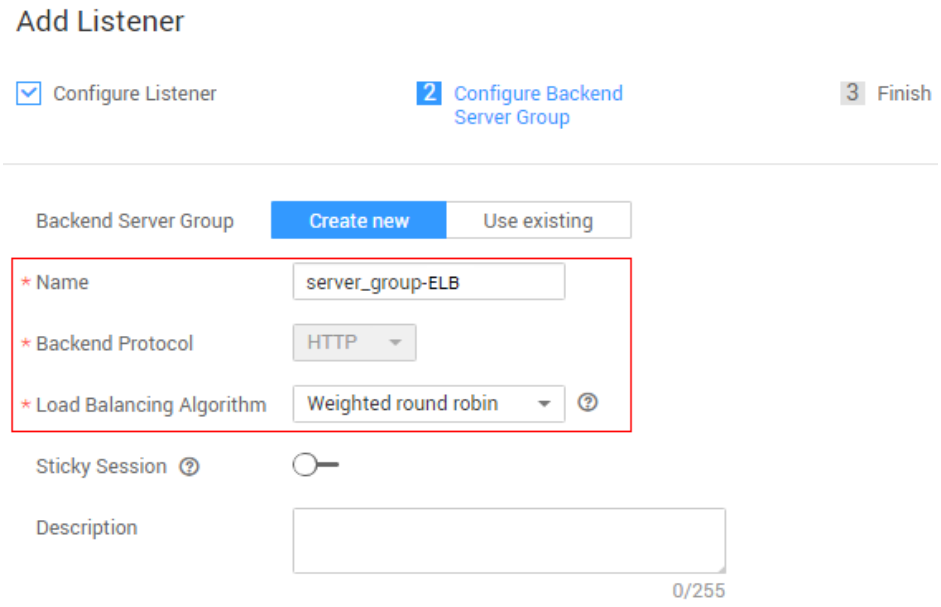
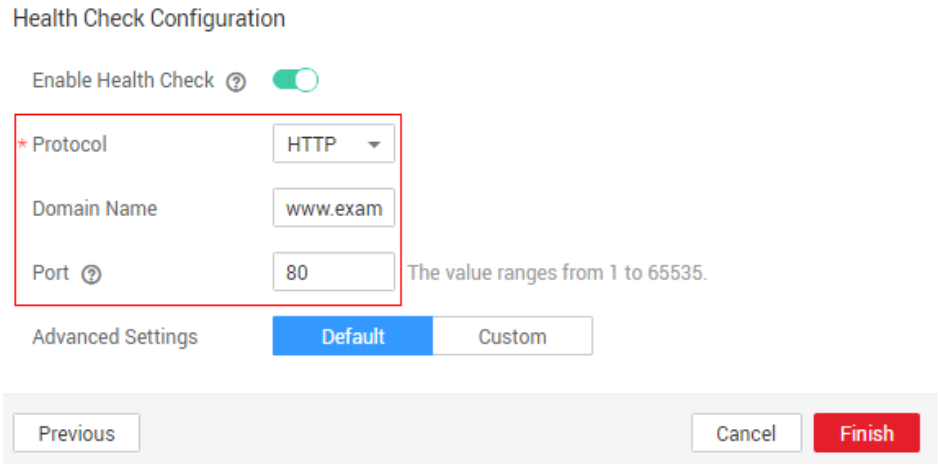


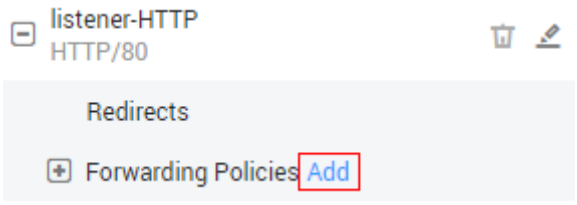
Figure 3-8 Configuring a health check



Add Forwarding Policies

1. Click the name of the newly added listener and then click **Add** next to **Forwarding Policies**.

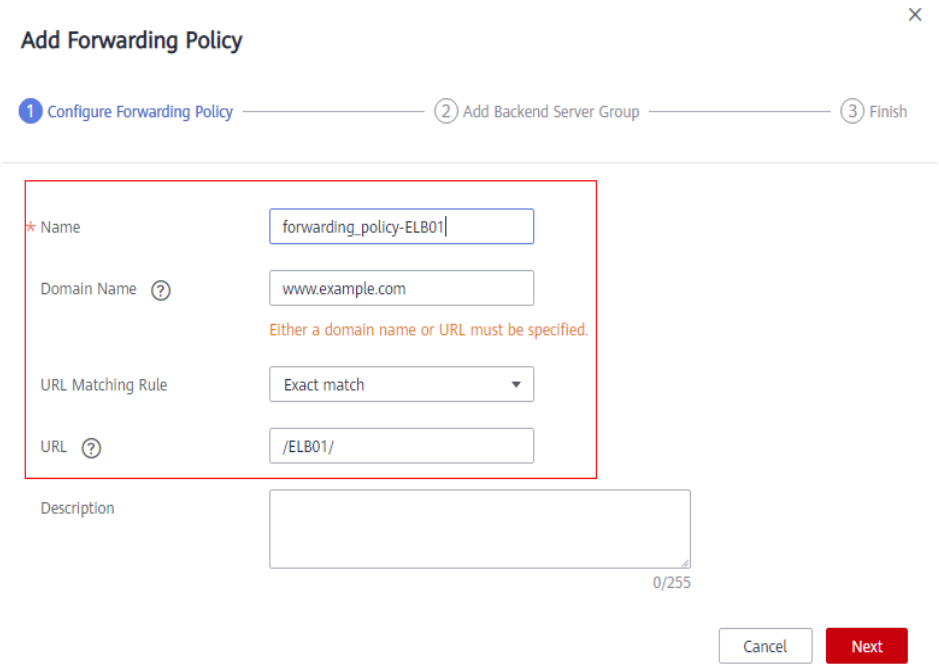
Figure 3-9 Adding a forwarding policy



2. Configure the forwarding policy and click **Next**.

- **Name**: specifies the forwarding policy name, for example, **forwarding_policy-ELB01**.
- **Domain Name**: specifies the domain name for triggering the forwarding policy, for example, www.example.com. Only exact match is supported.
- **URL**: specifies the URL for triggering the forwarding policy, for example, **/ELB01/**.
- **URL Matching Rule**: specifies the rule for matching specified URL string with the requested RUL. Three options are available, **Exact match**, **Prefix match**, and **Regular expression match**. **Exact match** enjoys the highest priority, and **Regular expression match** the lowest priority. Select **Exact match** here.

Figure 3-10 Configuring a forwarding policy



3. Add the backend server group and configure a health check.
   - Backend server group
     - **Name**: Enter a name for the backend server group, for example, **server_group-ELB01**.
     - **Load Balancing Algorithm**: Select an algorithm that the load balancer will use to route requests, for example, **Weighted round robin**.
   - Health check
     - **Protocol**: Select a protocol for the load balancer to perform health checks on backend servers. If the load balancer uses TCP, HTTP, or HTTPS to receive requests, the health check protocol can be TCP or HTTP. Set it to **HTTP**. Note that this setting cannot be changed after the listener is added.
     - **Domain Name**: Enter a domain name that will be used for health checks, for example, **www.example.com**.

- **Port**: Enter a port for the load balancer to perform health checks on backend servers, for example, **80**.

Figure 3-11 Configuring the backend server group

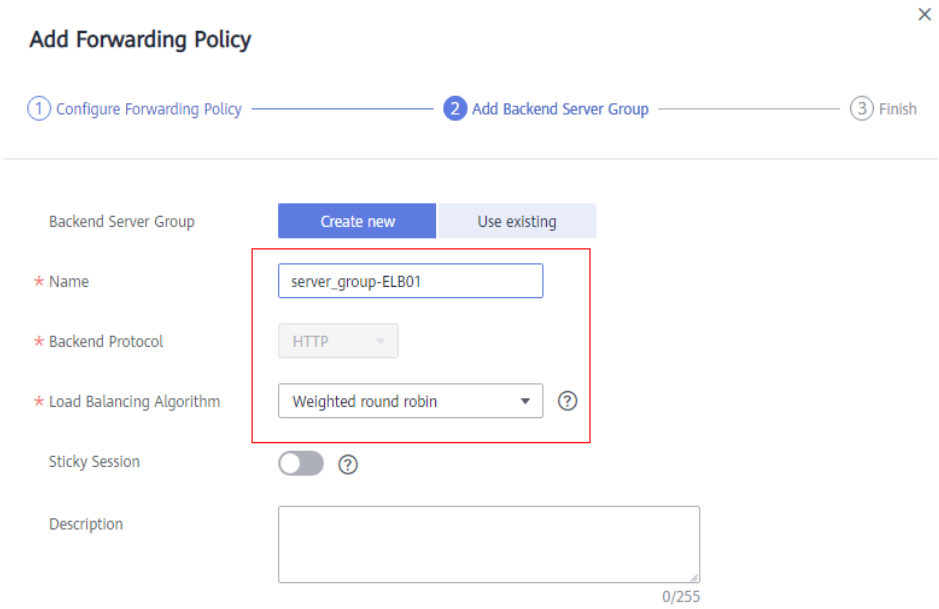
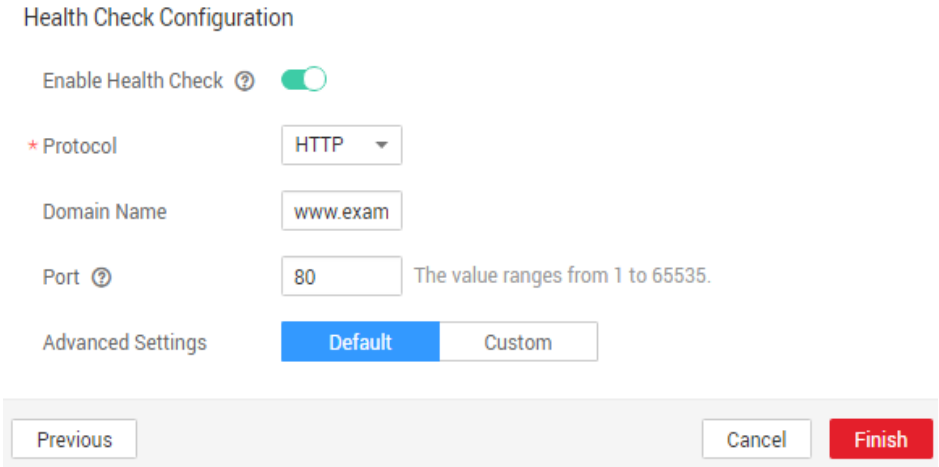
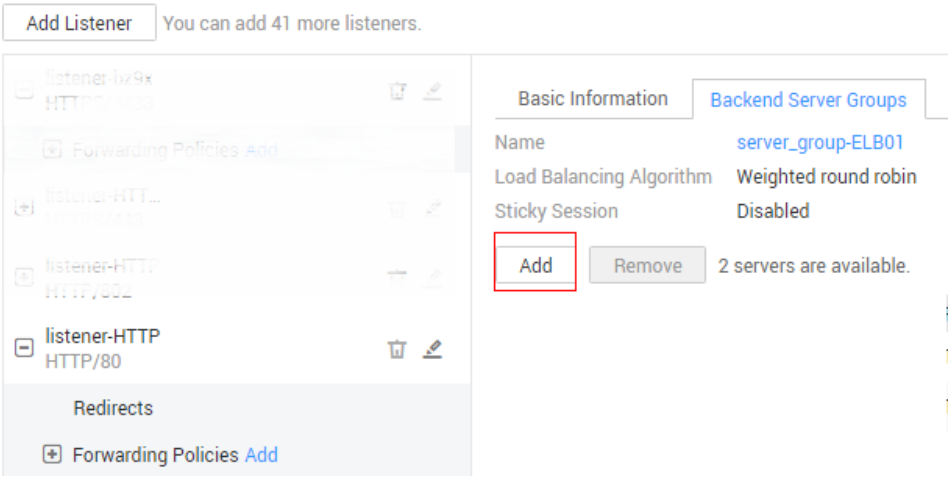


Figure 3-12 Configuring a health check



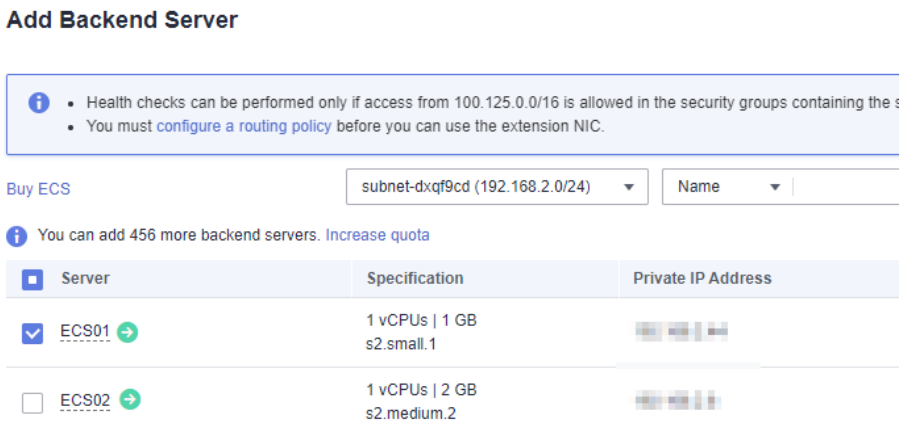
4. Select the newly added forwarding policy. On the **Backend Server Groups** tab page on the right, click **Add**.

Figure 3-13 Adding backend servers



5. Select the server you want to add, set the backend port, and click **Finish**.
   - Backend server: ECS01
   - Backend port: specifies the port used by the backend servers, for example, 80.

Figure 3-14 Adding backend servers



6. Repeat 1 to 5 to add another forwarding policy, create a backend server group, and add ECS02 to the backend server group. Set the parameters as shown in the following figures.

Figure 3-15 Basic information

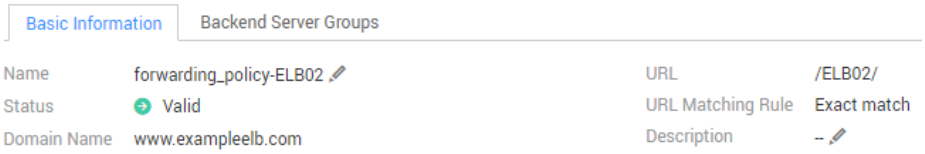
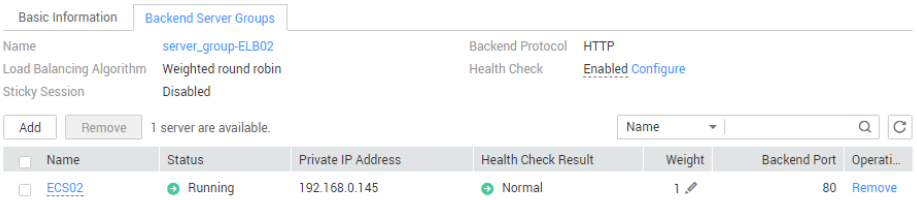


Figure 3-16 Backend server group



Verify Load Balancing

After the load balancer is configured, you can access the domain name or the specified URL to check whether the two ECSs are accessible.

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

   You can view the load balancer EIP on the basic information page of the load balancer.

   Figure 3-17 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
   ```

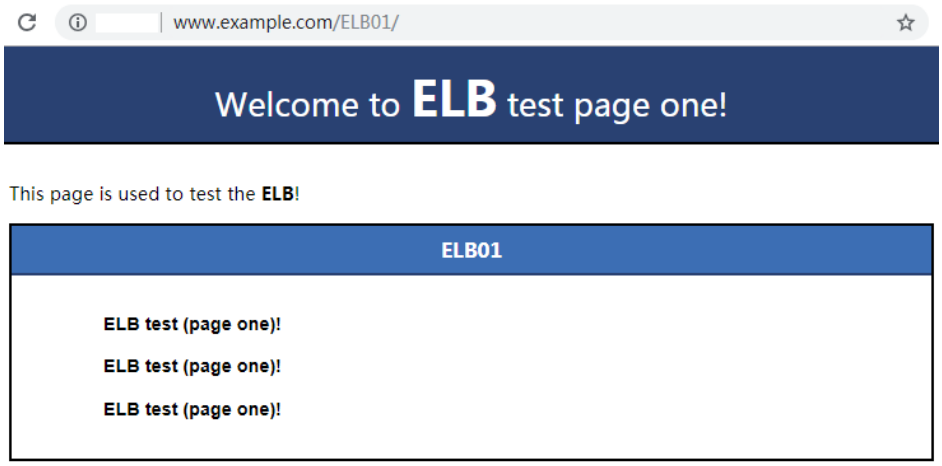
2. On the CLI of your PC, run the following command to check whether the domain name is mapped to the load balancer EIP:

   ping www.example.com

   If data packets are returned, the domain name has been mapped to the load balancer EIP.

3. 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 3-18 Accessing ECS01

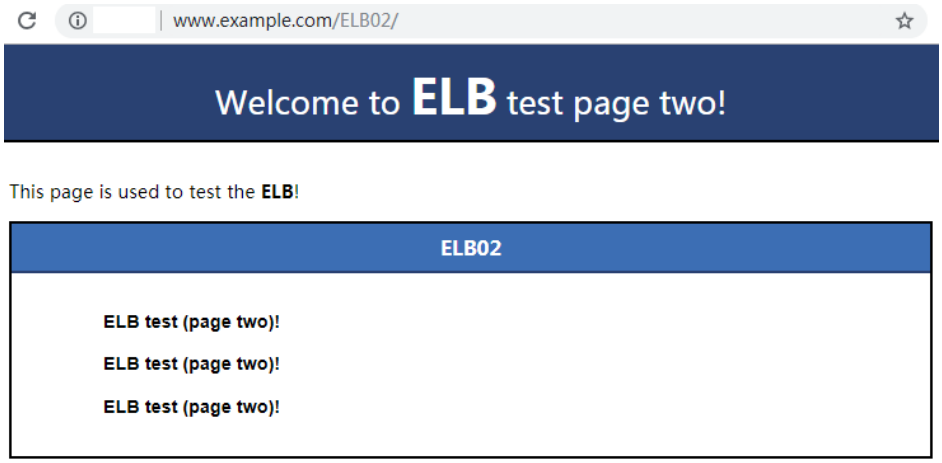
   

NOTE

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

4. 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 3-19 Accessing ECS02



4 Change History

Released On	Description
2020-05-30	This issue is the third official release, which incorporates the following changes: Renamed enhanced load balancer as shared load balancer.
2018-12-30	This issue is the second official release, which incorporates the following changes: Modified some parameters based on the latest console.
2018-06-30	This issue is the first official release.