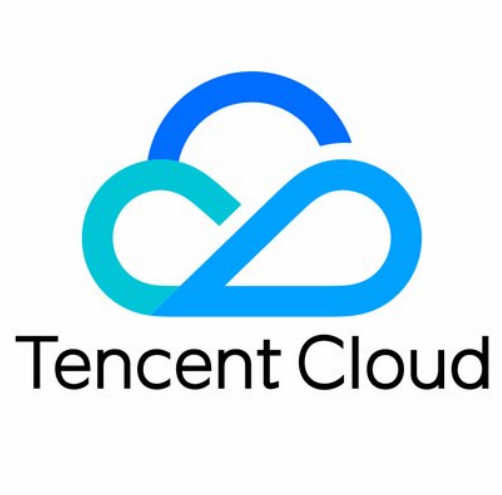
Cloud Load Balancer

Getting Started

Product Documentation



Contents

Getting Started
- Getting Started with Domain Name-Based CLB
- Getting Started with CLB
- Getting Started with IPv6 CLB
- Getting Started with Classic CLB
- Deploying Nginx on CentOS
- Deploying Java Web on CentOS

Getting Started
Getting Started with Domain Name-Based CLB

Last updated：2024-01-04 09:44:16

Tencent Cloud Cloud Load Balancer (CLB) supports various protocols such as TCP, UDP, TCP SSL, QUIC, HTTP, and HTTPS, providing businesses with both domain name- and URL-based forwarding services. This document describes how to quickly create a domain name-based CLB instance to forward client requests to two cloud virtual machine (CVM) instances.

Prerequisites

1. You have created two CVM instances. This document takes `rs-1` and `rs-2` as two sample instances. For information about how to create CVM instances, see Creating Instances via CVM Purchase Page.
2. You have deployed real servers on the two CVM instances. This document takes HTTP forwarding as an example. Nginx servers have been deployed on CVM instances `rs-1` and `rs-2`, and the two instances return two HTML static pages saying "Hello nginx! This is rs-1!" and "Hello nginx! This is rs-2!". For more information, see Deploying Nginx on CentOS.

Note:
This document describes the steps for bill-by-IP accounts. For bill-by-CVM accounts, first purchase public network bandwidth for CVM instances. If you are unsure of your account type, see Checking Account Type.
In this example, different services deployed on real servers will return different values. In practice, the services deployed on real servers are identical, to provide a consistent experience for all users.

Directions Overview

1. Register a domain name
2. Purchase a CLB instance
3. Configure a CLB listener
4. Configure a security group
5. Add a CNAME record
6. Verify the CLB service

Directions

Step 1: Register a domain name

Domain name registration is the prerequisite for building a service on the Internet.
If you have already registered a domain name with another registrar, you can transfer it to Tencent Cloud domain service. For more information, see Domain Transfer In.
If you do not have a domain name, you must register a domain name first.

Step 2: Purchase a CLB instance

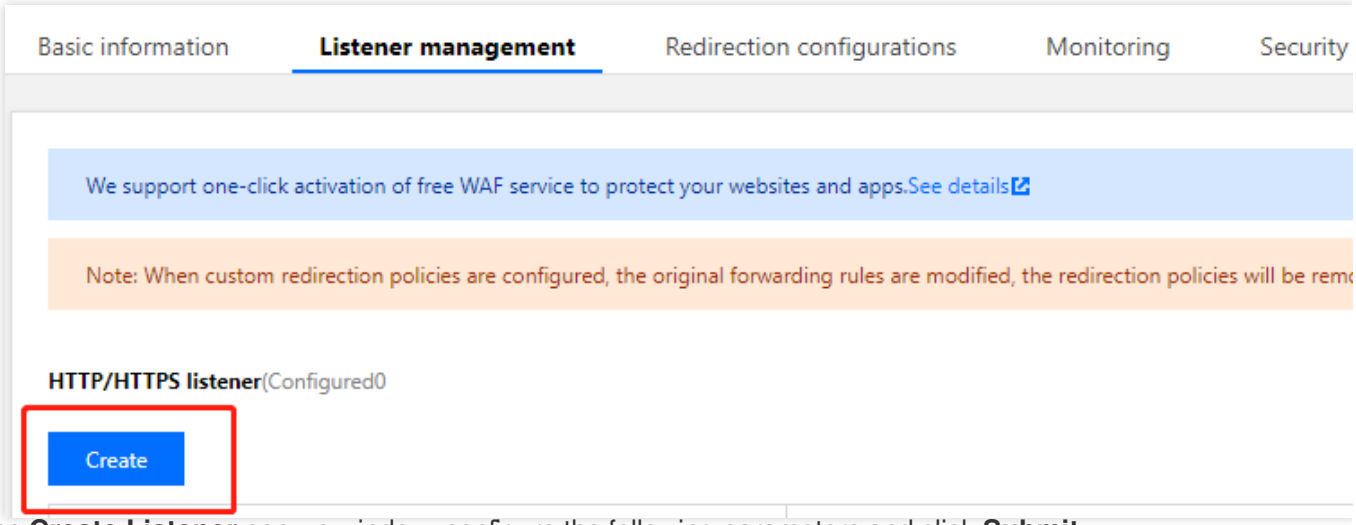
After you purchase a CLB instance, the sytem automatically allocates a domain name to the instance. You can use this domain name to access the CLB service.
1. Log in to the Tencent Cloud console and go to the CLB purchase page.
2. On the CLB purchase page, select the region in which your CVM instances are located, and select **CLB** as the instance type and **Public Network** as the network type. For more information, see Product Attribute Selection.
3. Click **Buy now** and complete the payment.
4. Return to the **Instance management** page and select the region to view the newly created instance.

Step 3: Configure a CLB listener

CLB listeners implement forwarding based on the designated protocol and port. This document takes configuring a CLB listener to forward HTTP requests from clients as an example. For more information about CLB listeners, see CLB Listener Overview.

Configure the HTTP listening protocol and port

When a client initiates a request, the CLB instance will receive the request according to the listening frontend protocol and port, and forward the request to the real server.
1. Log in to the CLB Console.
2. On the **Instance management** page, click **Configure listener** in the **Operation** column of the target CLB instance.
3. On the **Listener Management** tab, click **Create** in the **HTTP/HTTPS Listener** section.



4. In the **Create Listener** pop-up window, configure the following parameters and click **Submit**.

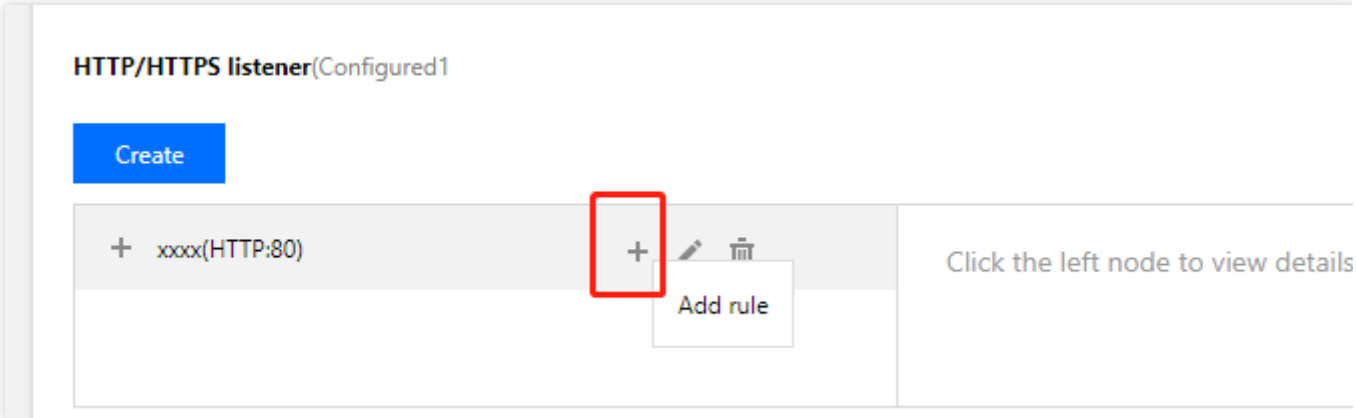
Listener name: The name can contain up to 60 characters, including letters, digits, hyphens (-), underscores (_), and dots(.).

Listener protocol port: For example, enter `HTTP:80` .

Configure a forwarding rule for the listener.

If a client initiates a request, the CLB instance will forward the request according to the configured forwarding rule of the listener.

1. On the **Listener Management** tab, locate the listener you created, and click + on the right of the listener to add a rule.



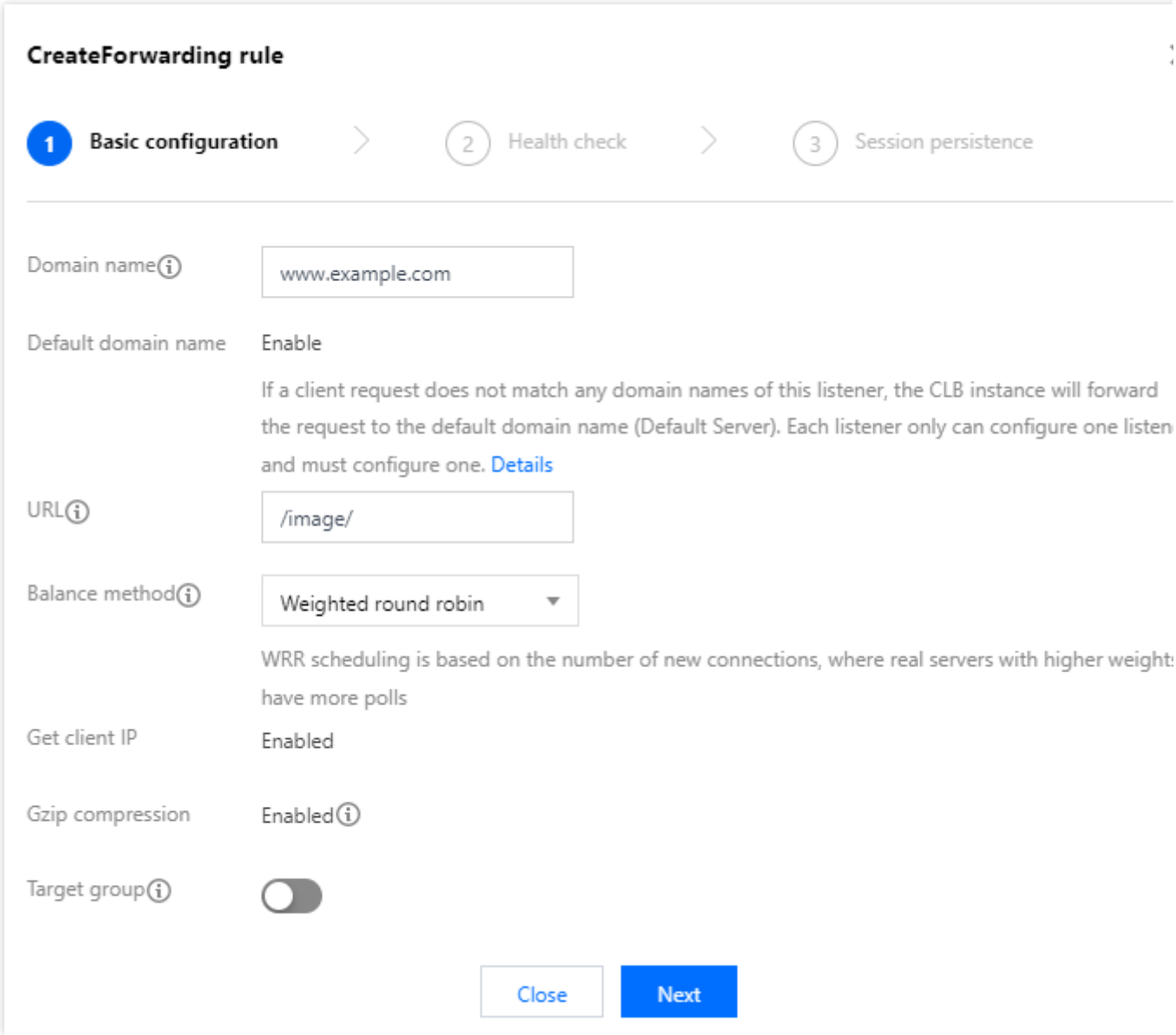
2. In the **Create Forwarding Rules** window, configure the domain name, URL, balancing method, and then click **Next**.

Domain Name: The domain name of your real server, for example, `www.example.com` .

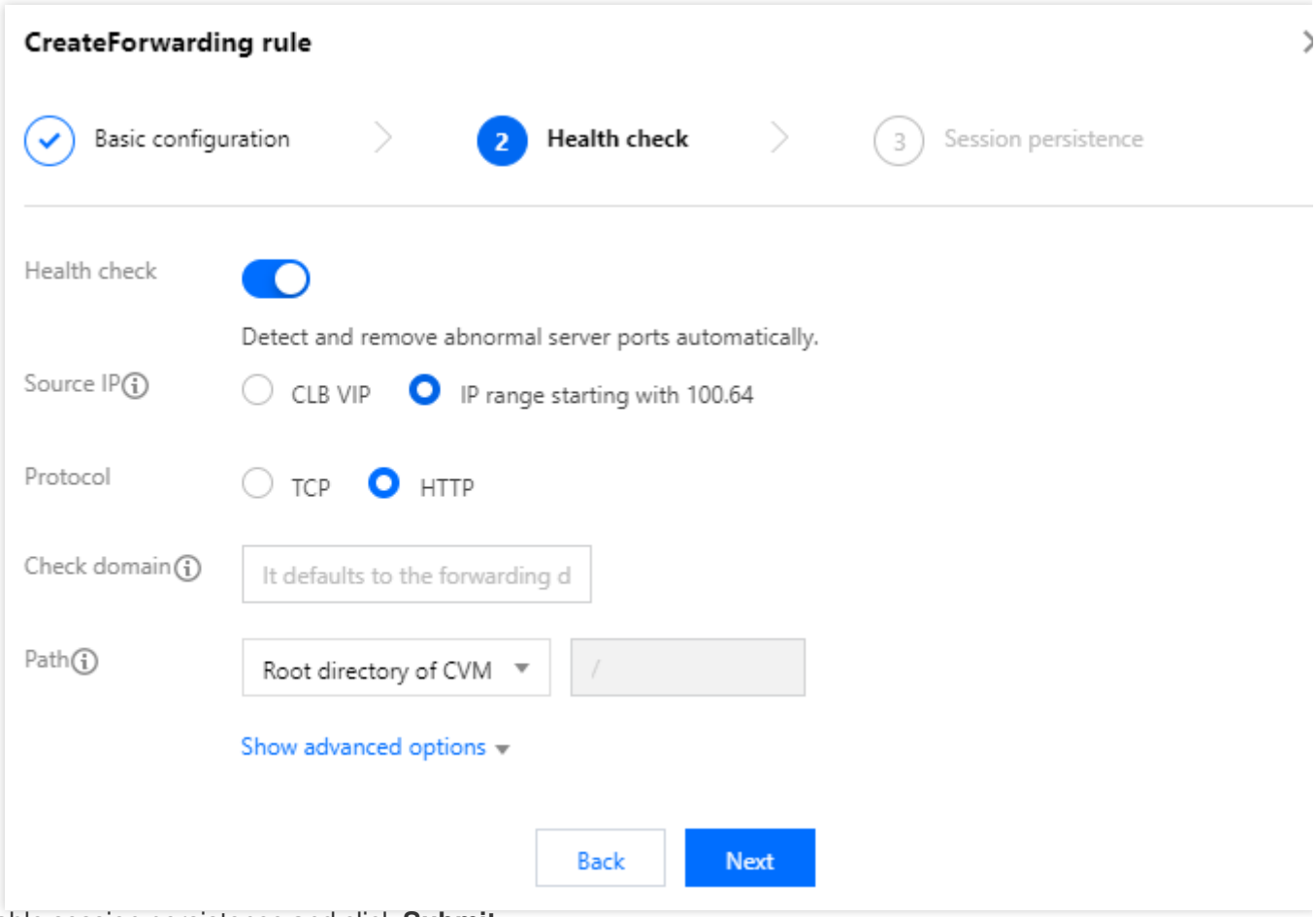
Default Domain Name: If a client request does not match any listener domain names, the CLB instance will forward the request to the default domain name (default server). Each listener can be configured with only one default domain name. If a listener has no default domain name, the CLB instance will forward the request to the first domain name. In this example, the default domain name is not configured.

URL: The access path to your real server, for example, `/image/` .

Balance Method: Select **Weighted Round Robin**. For more information about balancing methods, see Load Balancing Methods.



3. On the **Health Check** tab, enable **Health Check**, retain the default values for the **Check Domain** and **Path** fields, and click **Next**.



4. Disable session persistence and click **Submit**.

Note:

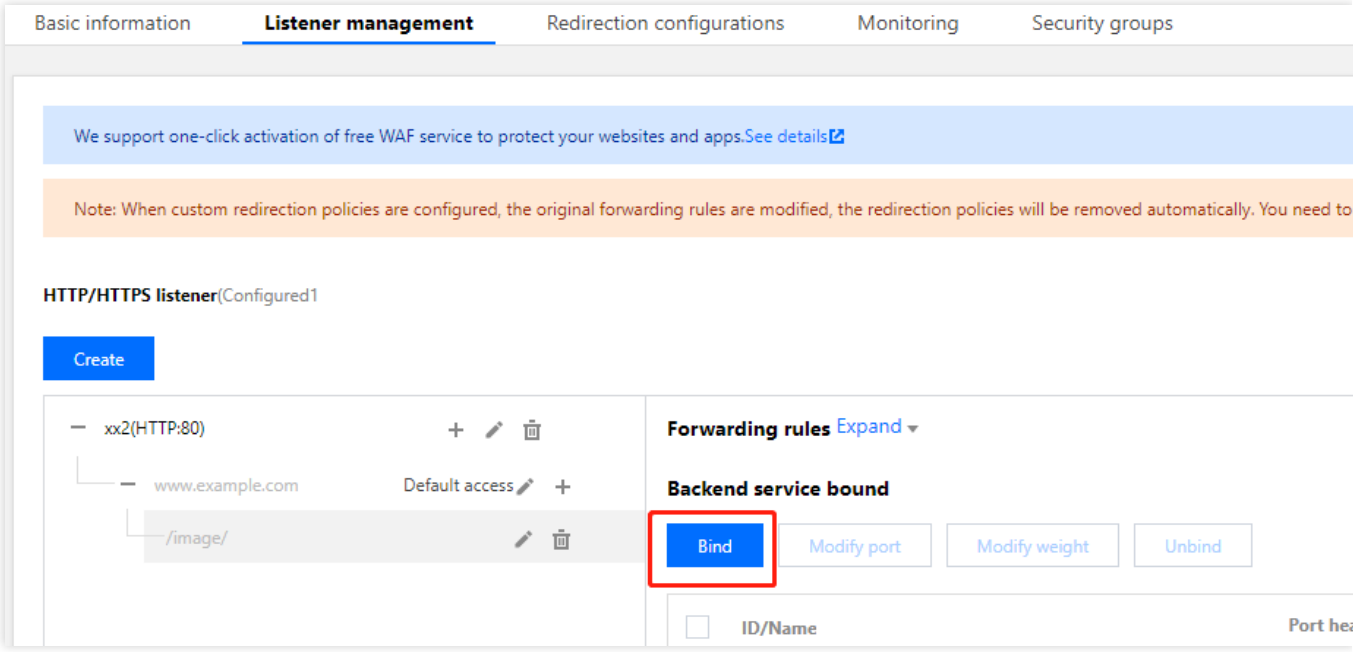
Forwarding rules: Each listener can be configured with multiple domain names, and each domain name can be configured with multiple URLs. You can select a listener or domain name, and then click the **+** icon to create new rules.

Session persistence: If session persistence is disabled and the round-robin balancing method is selected, requests from the same client will be assigned to different real servers in sequence; if session persistence is enabled, or session persistence is disabled but the `ip_hash` balancing method is selected, requests from the same client will always be assigned to the same real server.

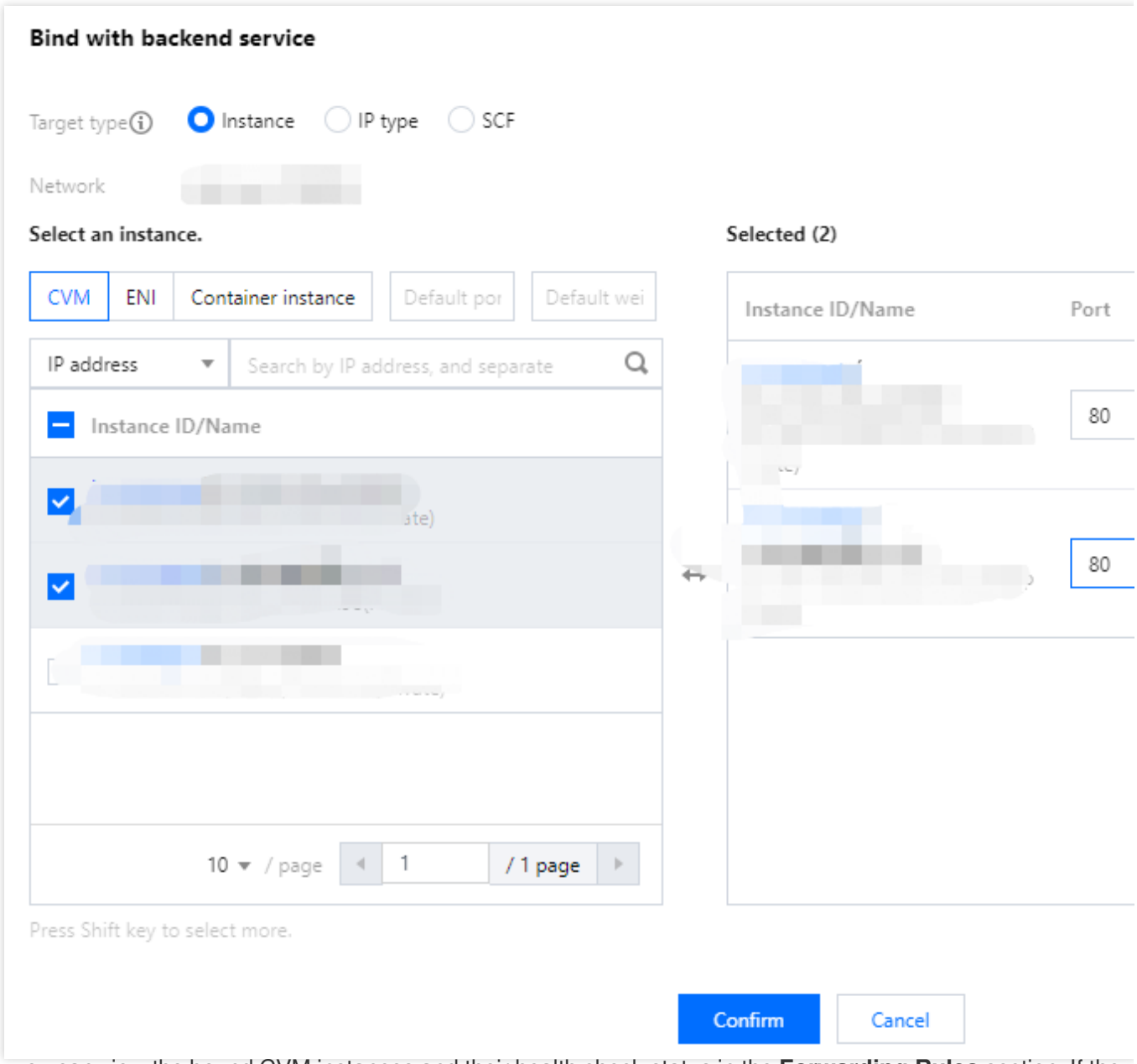
Bind real servers to the listener

If a client initiates a request, the CLB instance will forward the request to the CVM instance that is bound to its listener for processing.

1. On the **Listener Management** tab, click **+** on the left of the listener you created to display the listener information. Select the URL, and click **Bind** in the **Forwarding Rules** section on the right.



2. In the pop-up window, select **CVM** as the instance type, select the two CVM instances `rs-1` and `rs-2` that are located in the same region as the CLB instance, set their ports to **80** and weights to **10** (the default value), and then click **Confirm**.



3. Now you can view the bound CVM instances and their health check status in the **Forwarding Rules** section. If the port health status is **Healthy**, the CVM instances can normally process requests forwarded by the CLB instance.

Note:

One forwarding rule (containing the listening protocol, port, domain name, and URL) can be bound with multiple ports of the same CVM instance. If the same service is deployed on ports `80` and `81` of `rs-1`, both ports can be bound with the sample forwarding rule and both will receive requests forwarded by the CLB instance.

Step 4: Configure a security group

After creating a CLB instance, you can configure a security group to isolate public network traffic. For more information, see Configuring CLB Security Group.

After configuring a security group, you can enable or disable the **Allow by Default** feature:

Method 1: Enable "Allow by Default" for the security group

For more information, see Configure Allow by Default.

Method 2: Allow specific client IPs on the CVM security group

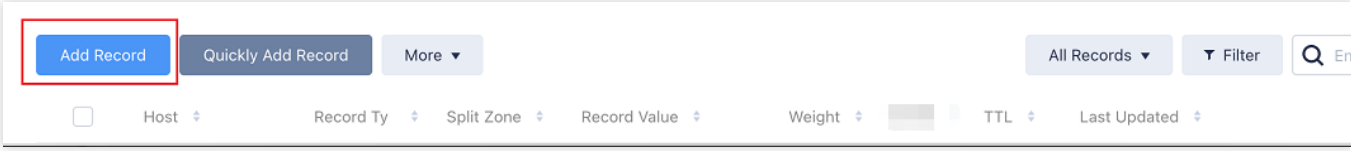
For more information, see Configure Allow by Default.

Step 5: Add a CNAME record

After registering a domain name, you can add a CNAME record for the domain name so that the domain name can be used to access your website.

1. Log in to the DNSPod console. On the domain name list page, click **DNS** in the **Operation** column of the target domain name.

2. On the **Record Management** tab, click **Add Record**.



3. In the **Add Record** section, set the following parameters:

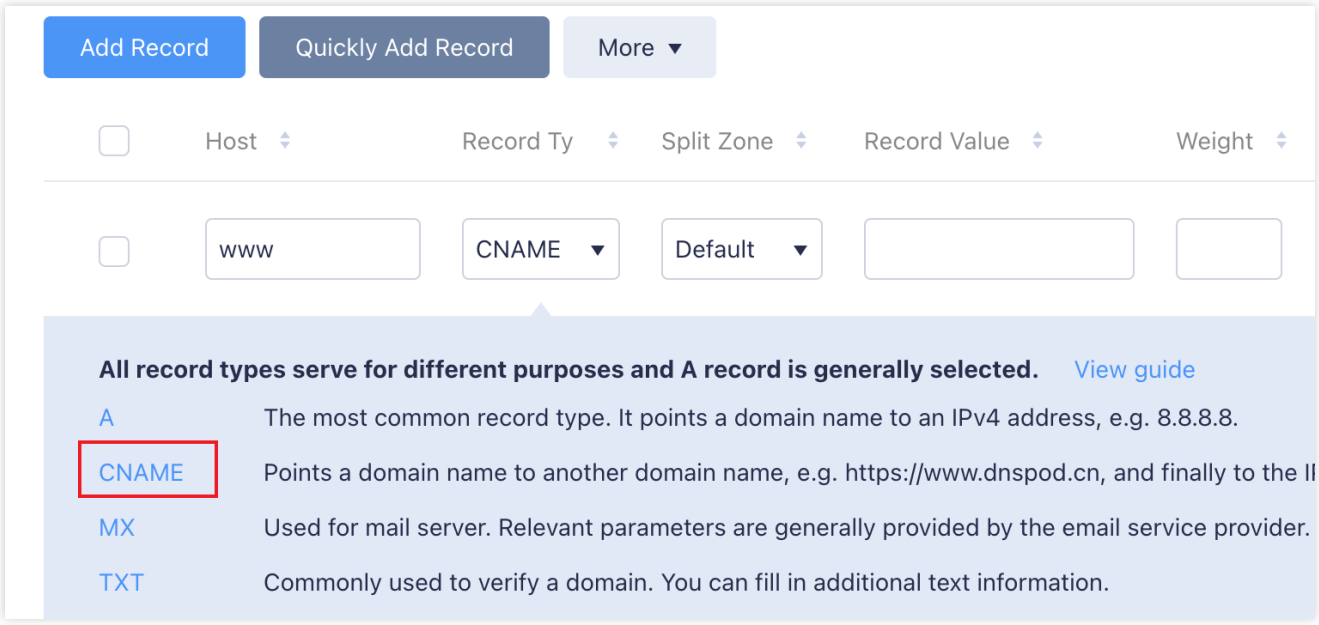
3.1 **Host**: specifies the prefix of the domain name. Valid values:

`www` : The domain name is resolved to `www.example.com` .

`@` : The domain name is resolved to `example.com` .

`*` : matches all domain names in the format of `*.example.com` .

3.2 **Record Type**: We recommend that you select `CNAME` .



3.3 **Split Zone**: specifies the lines through which users access the domain name.

If the host service provider provides only one IP address or domain name, you can select **Default**.

Common line types are as follows:

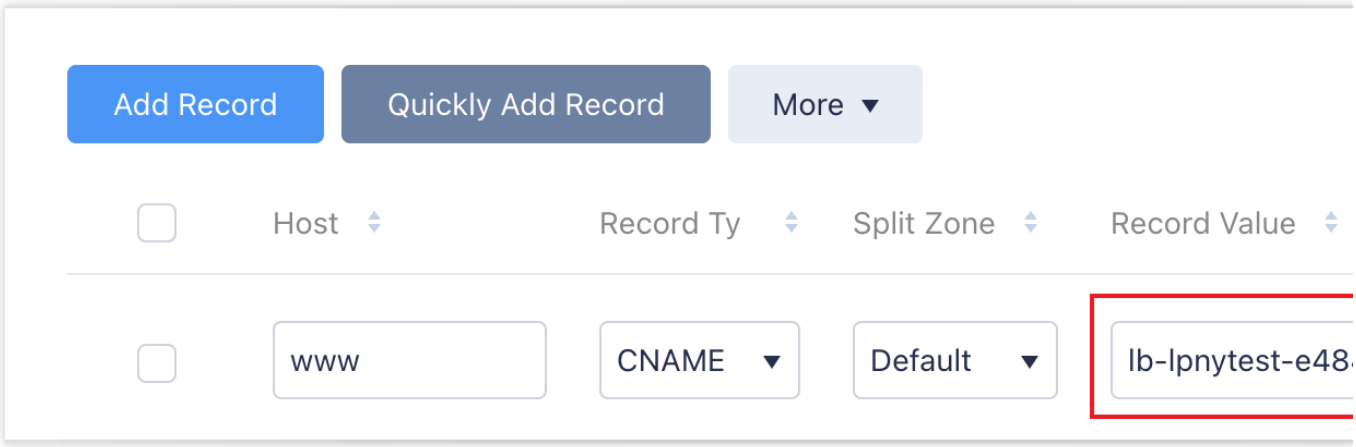
`Default` : The default line must be added. Otherwise, your website can be accessed only from specified lines. If your website can be accessed from two lines, we recommend you select **China Telecom** as the default line.

`China Unicom` : specifies the server IP address for **China Unicom users**. Other users still access the domain name by using the **default** line.

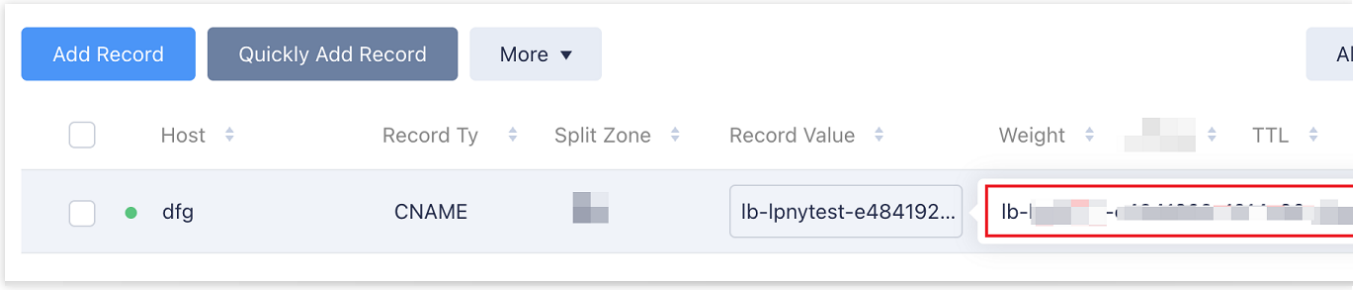
`Search Engines` : specifies a server IP address for web crawlers to fetch information from the website.

3.4 **Record Value**: You can enter the domain name allocated by CLB.

3.5 Retain the default values for other parameters and click **Save**.



4. After adding the record, you can view the record in the record list on the **Record Management** tab.



Step 6: Verify the CLB service

After adding the record, wait for about 10 minutes and enter the bound CNAME domain name in the address bar of your browser. In this example, enter `www.example.com` . If the page appears normally, the CLB service is valid.

Configuring Redirection (optional)

CLB supports automatic redirection and manual redirection. For more information, see Layer-7 Redirection Configuration.

Automatic redirection (forced HTTPS): When a PC or mobile browser accesses a web service with an HTTP request, an HTTPS response is returned to the browser after the request passes through the CLB proxy, forcing the browser to access the webpage by using HTTPS.
Manual redirection: If you want to temporarily deactivate your web page in case of product sellout, page maintenance, or update and upgrade, you need to redirect the original page to a new page. Otherwise, a 404 or 503 error message page is returned when a user visits the original page. This results in compromised user experience and a waste of the access traffic, and may even invalidate the accumulated scores of the page on search engines.

References

Deploying Java Web on CentOS
Install and Configure PHP

Getting Started with CLB

Last updated：2024-01-04 09:44:16

Tencent Cloud CLB comes with various protocols such as TCP, UDP, TCP SSL, HTTP, and HTTPS, providing businesses with domain names and URL-based forwarding services. This document guides you to quickly create a CLB instance and forward client requests to two CVM instances.

Prerequisites

1. You have created two CVM instances. This document takes `rs-1` and `rs-2` as two sample instances. For information about how to create CVM instances, see Creating Instances via CVM Purchase Page.
2. You have deployed real servers on the two CVM instances. This document takes HTTP forwarding as an example. Nginx servers have been deployed on CVM instances `rs-1` and `rs-2`, and the two instances return two HTML static pages saying "Hello nginx! This is rs-1!" and "Hello nginx! This is rs-2!". For more information, see Deploying Nginx on CentOS.

Note:
This document describes the steps for bill-by-IP accounts. For bill-by-CVM accounts, first purchase public network bandwidth for CVM instances. This is because the current bandwidth attributes are on CVM instances instead of CLB instances. If you are unsure of your account type, see Checking Account Type.
In this example, different services deployed on real servers will return different values. In practice, the services deployed on real servers are identical to provide a consistent experience for all users.

Step 1: Purchasing a CLB Instance

After a successful purchase, the system will automatically assign a VIP to the CLB instance. The VIP will be used as the IP address to provide services to clients.
1. Log in to the Tencent Cloud console and go to the CLB purchase page.
2. On the CLB purchase page, select the region in which your CVM instances are located, and select **CLB** as the instance type and **Public network** as the network type. For more information, see Product Attribute Selection.

Note:
Currently, the static single-line IP is supported only in Guangzhou, Shanghai, Nanjing, Jinan, Hangzhou, Fuzhou, Beijing, Shijiazhuang, Wuhan, Changsha, Chengdu, and Chongqing. For the support information in other regions, see the console. This feature is currently in beta. To try it out, contact the sales rep for application. Once your application is approved, you can select an ISP (China Mobile, China Unicom, or China Telecom) on the purchase page.
3. Click **Buy now** and complete the payment.

4. Return to the **Instance management** page, select the region to see the new instance.

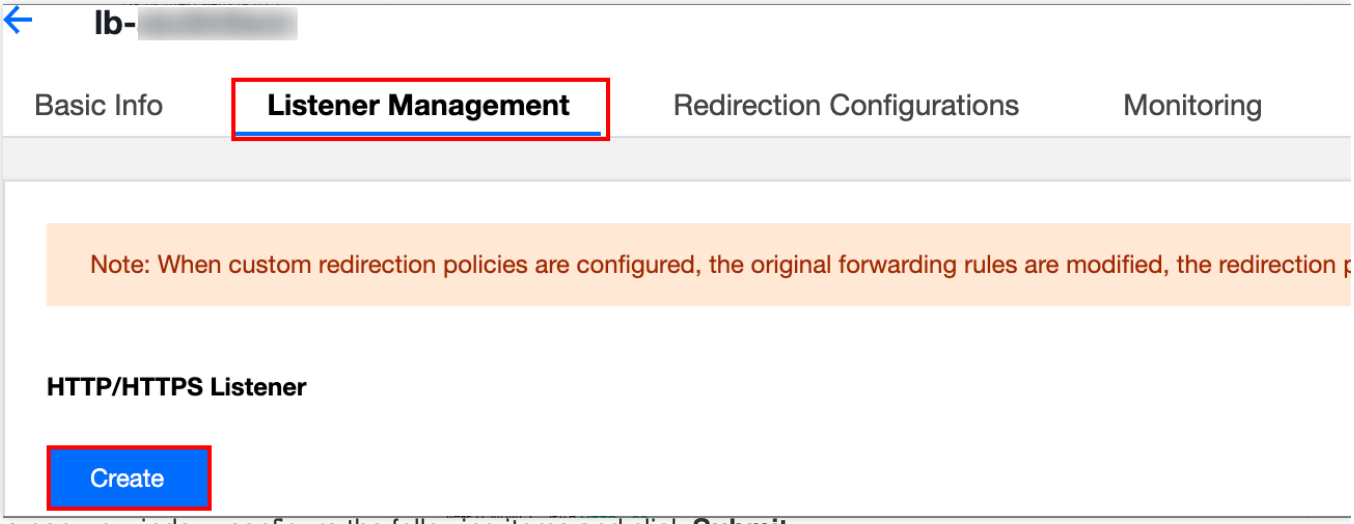
Step 2: Configuring a CLB Listener

A CLB listener is used for forwarding through a specified protocol and port. This document demonstrates how to configure a CLB instance to forward client HTTP requests.

Configuring the HTTP listening protocol and port

When a client initiates a request, the CLB instance will receive the request according to the listening frontend protocol and port, and forward the request to the real server.

1. Log in to the CLB console.
2. On the **Instance management** page, click **Configure listener** in the **Operation** column of the target CLB instance.
3. On the **Listener management** tab, click **Create** in the **HTTP/HTTPS listener** section.



4. In the pop-up window, configure the following items and click **Submit**.

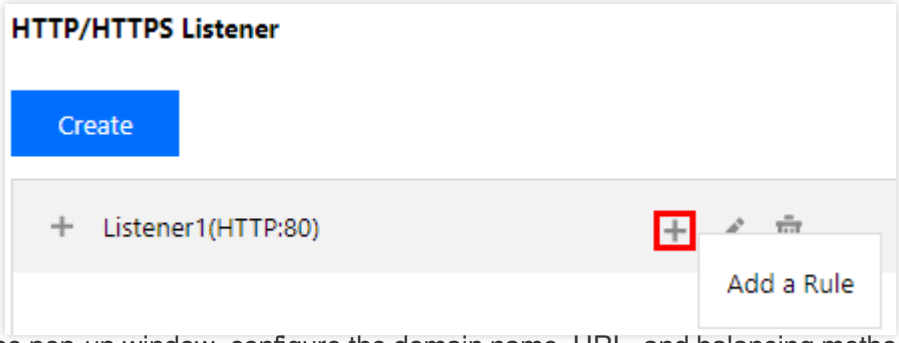
Listener name: The name can contain up to 60 characters, including letters, digits, hyphens (-), underscores (_), and dots(.).

Listener protocol and port: For example, `HTTP:80` .

Configuring the listener's forwarding rule

If a client initiates a request, the CLB instance will forward the request according to the configured forwarding rule of the listener.

1. On the **Listener management** tab, click **+** on the right of the new listener.



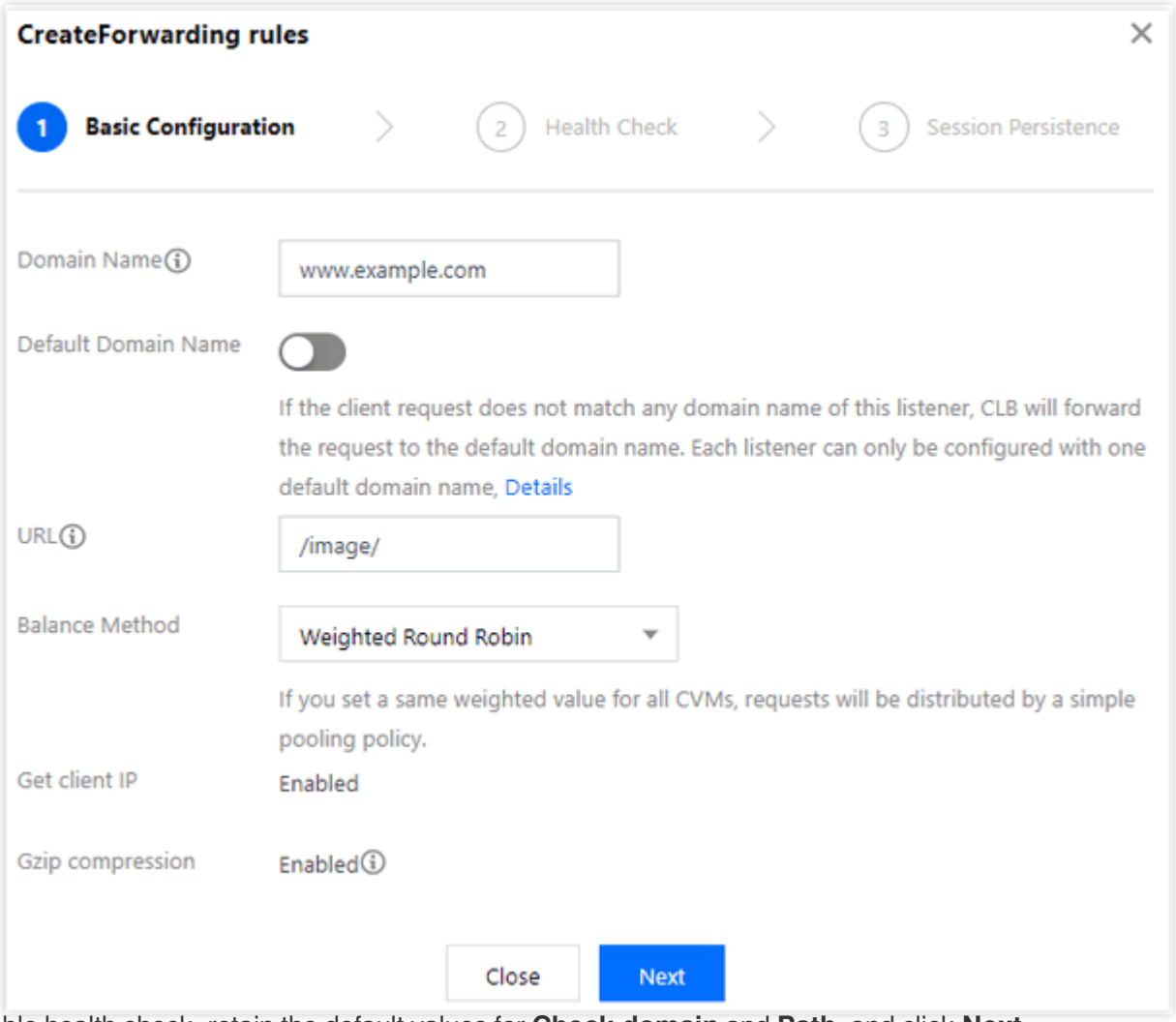
2. In the pop-up window, configure the domain name, URL, and balancing method on the **Basic configuration** tab and then click **Next**.

Domain name: The domain name of your real server, for example, `www.example.com`.

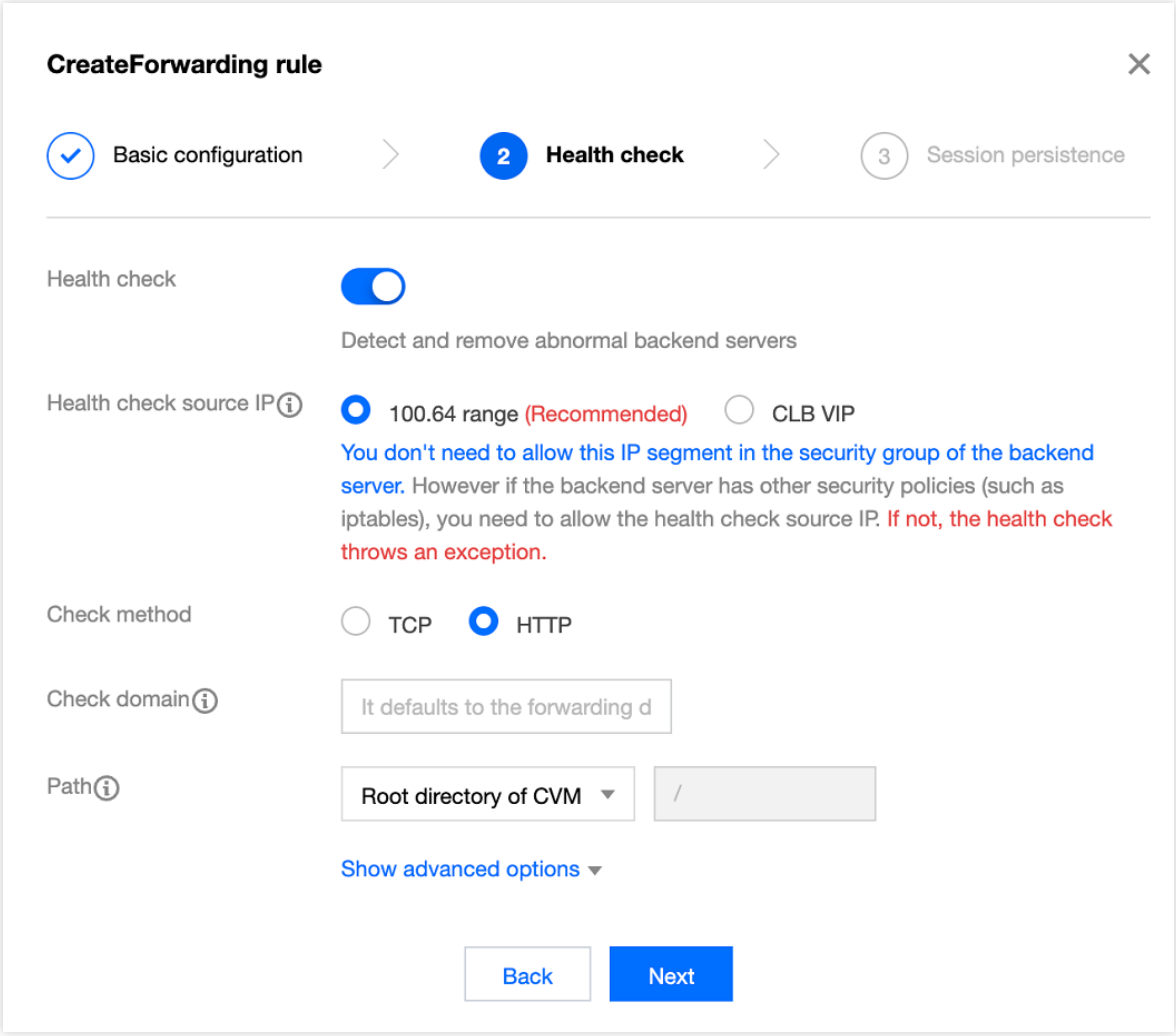
Default Domain: If a client request does not match any listener domain names, the CLB instance will forward the request to the default domain name (default server). Each listener can be configured with only one default domain name. If a listener has no default domain name, the CLB instance will forward the request to the first domain name. In this example, the default domain name is not configured.

URL: The access path to your real server, for example, `/image/`.

Balancing method: Select **Weighted round robin**. For more information, see Load Balance Methods.



3. Enable health check, retain the default values for **Check domain** and **Path**, and click **Next**.



4. Disable session persistence and click **Submit**.

For more information about CLB listeners, see CLB Listener Overview.

Note:

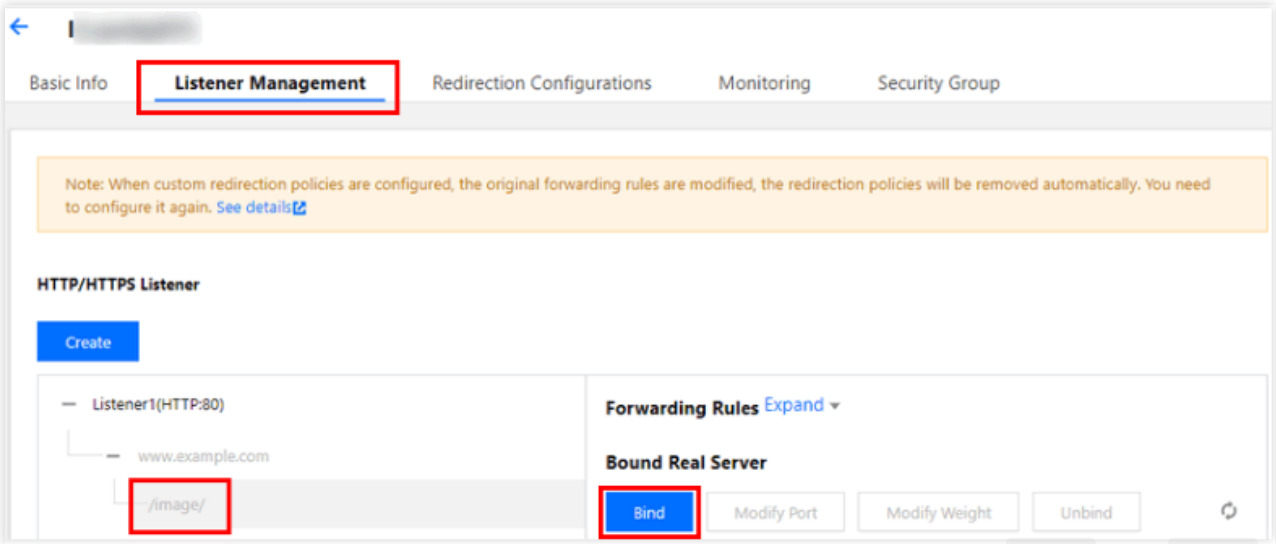
Forwarding rules: Each listener can be configured with multiple domain names, and each domain name can be configured with multiple URLs. You can select a listener or domain name, and then click the **+** icon to create new rules.

Session persistence: If session persistence is disabled and the round-robin balancing method is selected, requests from the same client will be assigned to different real servers in sequence; if session persistence is enabled, or session persistence is disabled but the `ip_hash` balancing method is selected, requests from the same client will always be assigned to the same real server.

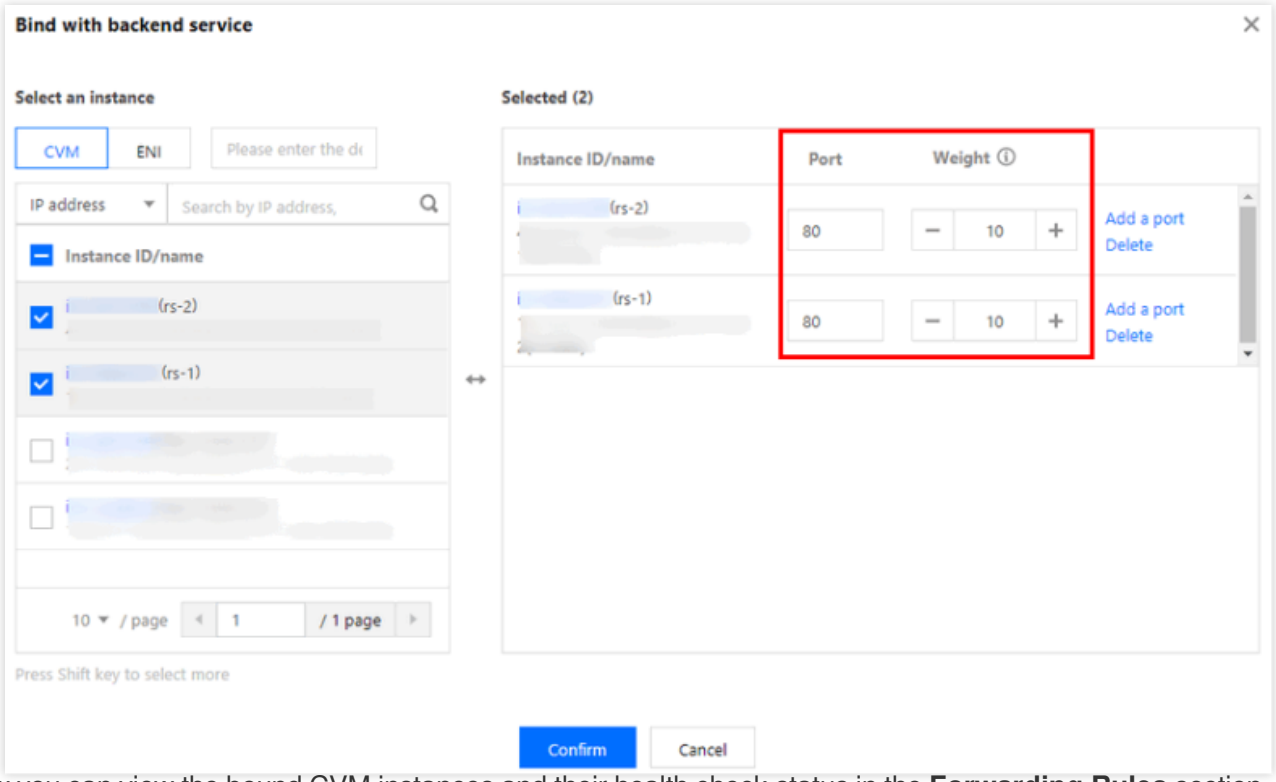
Binding real servers to the listener

If a client initiates a request, the CLB instance will forward the request to the CVM instance that is bound to its listener for processing.

1. On the **Listener management** tab, click **+** to expand the new listener. Click the URL, and click **Bind** in the **Forwarding rules** section on the right.



2. In the pop-up window, select **CVM** as the instance type, select the two CVM instances `rs-1` and `rs-2` (which are in the same region as the CLB instance), set their ports to `80` and weights to **10** (the default value), and click **Confirm**.



3. Now you can view the bound CVM instances and their health check status in the **Forwarding Rules** section. If the port health status is **Healthy**, the CVM instances can normally process requests forwarded by the CLB instance.

Note:

One forwarding rule (listening protocol, port, domain name, and URL) can be bound with multiple ports of the same CVM instance. If a user deploys the same service on the ports `80` and `81` of `rs-1`, both ports can be bound with the sample forwarding rule and both will receive requests forwarded by the CLB instance.

Step 3: Configuring a Security Group

After creating a CLB instance, you can configure a security group to isolate public network traffic. For more information, see Configuring CLB Security Group.
After configuring a security group, you can enable or disable the **Allow Traffic by Default** feature:

Method 1: Enabling "Allow Traffic by Default" for a security group

For more information, see Configure Allow by Default.

Method 2: Allowing specific client IP addresses in the CVM security group

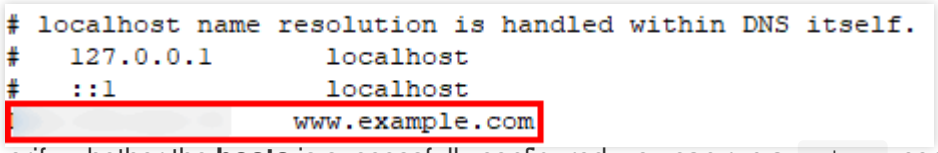
For more information, see Configure Allow by Default.

Step 4: Verifying the CLB Service

After configuring a CLB instance, you can verify whether it is effective by accessing different real servers via different **domain names and URLs** under the same CLB instance, or verifying the **Content-based Routing** feature.

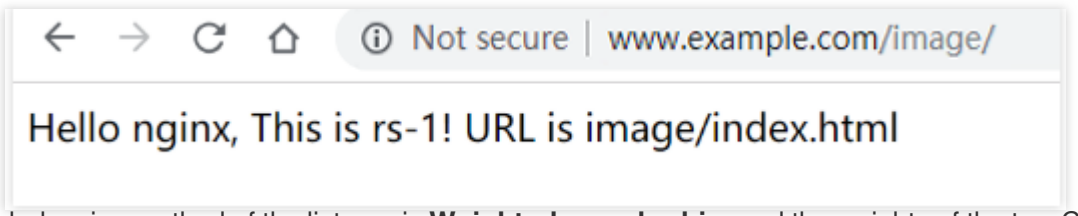
Method 1: Configuring `hosts` and mapping the domain name to the CLB instance

1. In a Windows device, modify the **hosts** file in the directory `C:\\Windows\\System32\\drivers\\etc` , and map the domain name to the CLB instance's VIP.

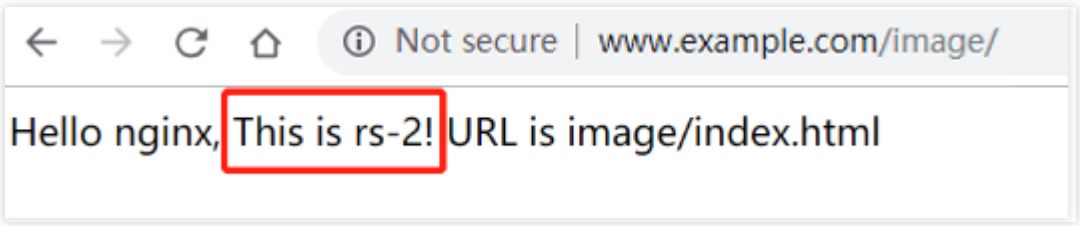


2. To verify whether the **hosts** is successfully configured, you can run a `ping` command in the **cmd.exe** to test whether the domain name is successfully bound with the VIP. If there are data packs returned, they are successfully bound.

3. Test the CLB service by accessing `http://www.example.com/image/` via a browser. If your page returns the image below, then the request has been forwarded to the CVM instance `rs-1` by the CLB instance, and the CVM has normally processed the request and returned the service page.



4. The balancing method of the listener is **Weighted round robin**, and the weights of the two CVM instances are **10**. You can refresh the browser to initiate the request again. If a result is returned as shown in the image below, the request is forwarded to the CVM instance `rs-2` by the CLB instance.



Note:

The `/` in the `image/` cannot be omitted. `/` indicates that **image** is a default directory instead of a file name.

Method 2: Mapping the domain name to the CLB instance through DNSPod

1. Go to the Tencent Cloud domain registration page to query and register a domain name. `example.com` is used in this example.

2. Log in to the DNSPod console. In the domain name list, click **DNS** in the **Operation** column of the target domain name.

3. Open the **Record Management** tab, click **Add Records** to add an A record for the domain name with the following parameters:

Host: The prefix of the domain name. `*.example.com` is used in this example, indicating resolving all prefixes.

Record Type: Select `A` .
Split Zone: Select **Default**.
Value: Click **Associate Tencent Cloud Resources** and then tick the CLB instance created above.
TTL: Leave it as the default value **600s**.
4. Click **Save**.
5. About 10 minutes later, open the bound CNAME domain name (`www.example.com`) in a browser. If the corresponding page can be normally displayed, it indicates that the CLB instance is in effect.

Configuring Redirection (optional)

CLB supports automatic redirection and manual redirection. For more information, see Layer-7 Redirection Configuration.
Automatic redirection (forced HTTPS): When a PC or mobile browser accesses a web service with an HTTP request, an HTTPS response is returned to the browser after the request passes through the CLB proxy, forcing the browser to access the webpage by using HTTPS.
Manual redirection: If you want to temporarily deactivate your web business in cases such as product sellout, page maintenance, or update and upgrade, you need to redirect the original page to a new page. Otherwise, the old address in a visitor's favorites and search engine database will return a 404 or 503 error message page, degrading the user experience, resulting in traffic waste, and even invalidating the accumulated scores on search engines.

Related Operations

Deploying Java Web on CentOS
Install and Configure PHP

Getting Started with IPv6 CLB

Last updated：2024-01-04 09:44:16

Tencent Cloud CLB supports three IP versions: IPv4, IPv6, and IPv6 NAT64. IPv6 CLB supports the TCP, UDP, TCP SSL, HTTP, and HTTPS protocols and provides flexible forwarding capabilities based on domain names and URL paths. This document guides you through how to get started with IPv6 CLB.

Note:

The IPv6 CLB is in beta. To use it, submit a ticket.

Prerequisites

1. CLB only forwards traffic but cannot process requests; therefore, you need to create a CVM instance that processes user requests and configure its IPv6 settings first.
2. This document takes HTTP forwarding as an example. The corresponding web server (such as Apache, Nginx, or IIS) must be deployed on the CVM instance, and the port used by the server needs to listen on IPv6.

Notes

Currently, IPv6 CLB is supported only in the following regions: Guangzhou, Shanghai, Nanjing, Beijing, Chengdu, Chongqing, Hong Kong (China), Singapore, Virginia, and São Paulo.

IPv6 CLB does not support classic CLB.

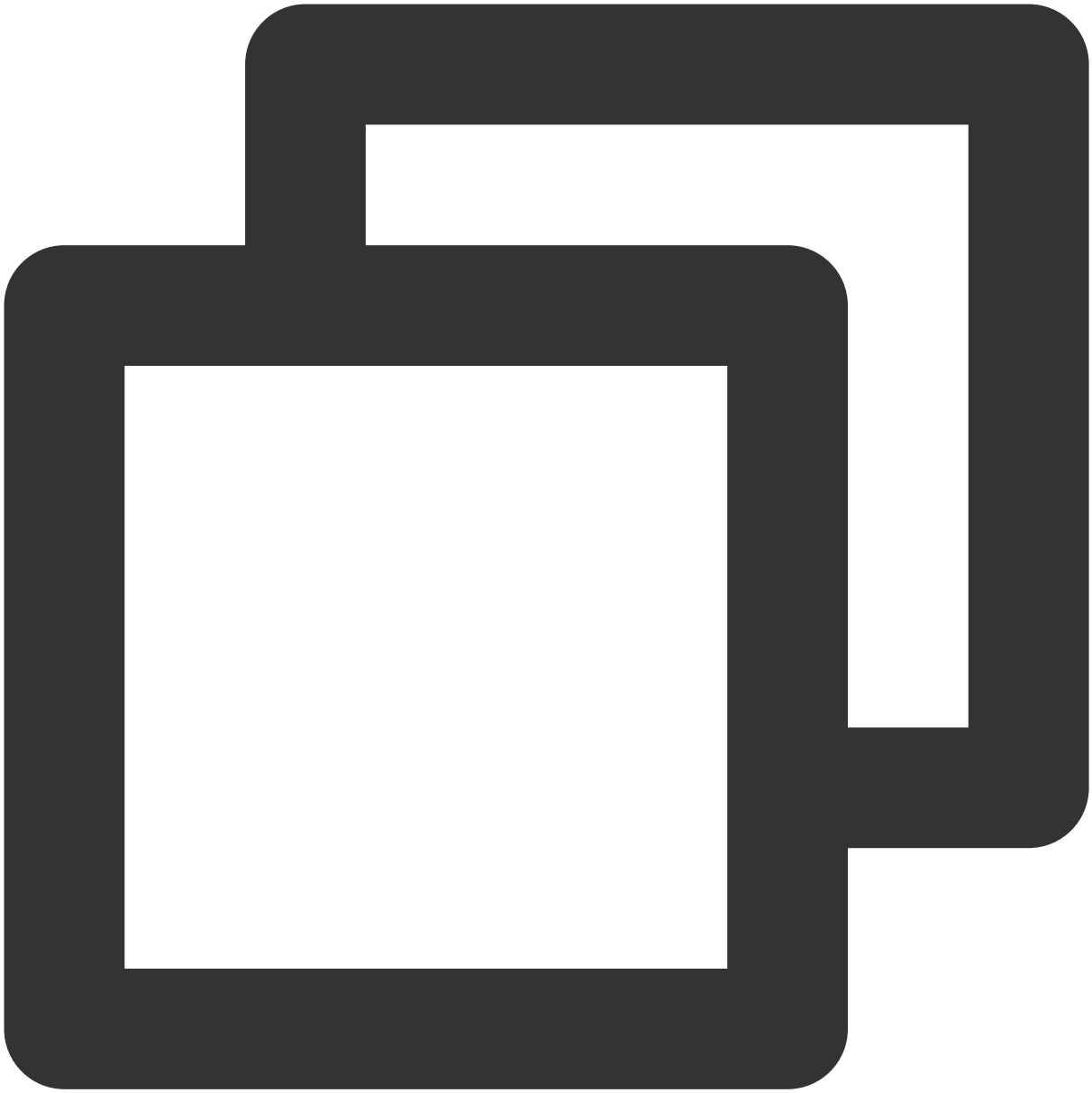
IPv6 CLB supports obtaining the client's IPv6 source address, which can be directly obtained by layer-4 IPv6 CLB or through the `X-Forwarded-For` header of HTTP layer-7 IPv6 CLB.

Currently, IPv6 CLB balances the load completely over a public network. Clients in the same VPC cannot access IPv6 CLB over a private network.

IPv6 implementations are still at the preliminary stage across the internet. In case of access failure, you can submit a ticket. SLA is not guaranteed during the beta test period.

Step 1. Create a CVM instance and configure IPv6

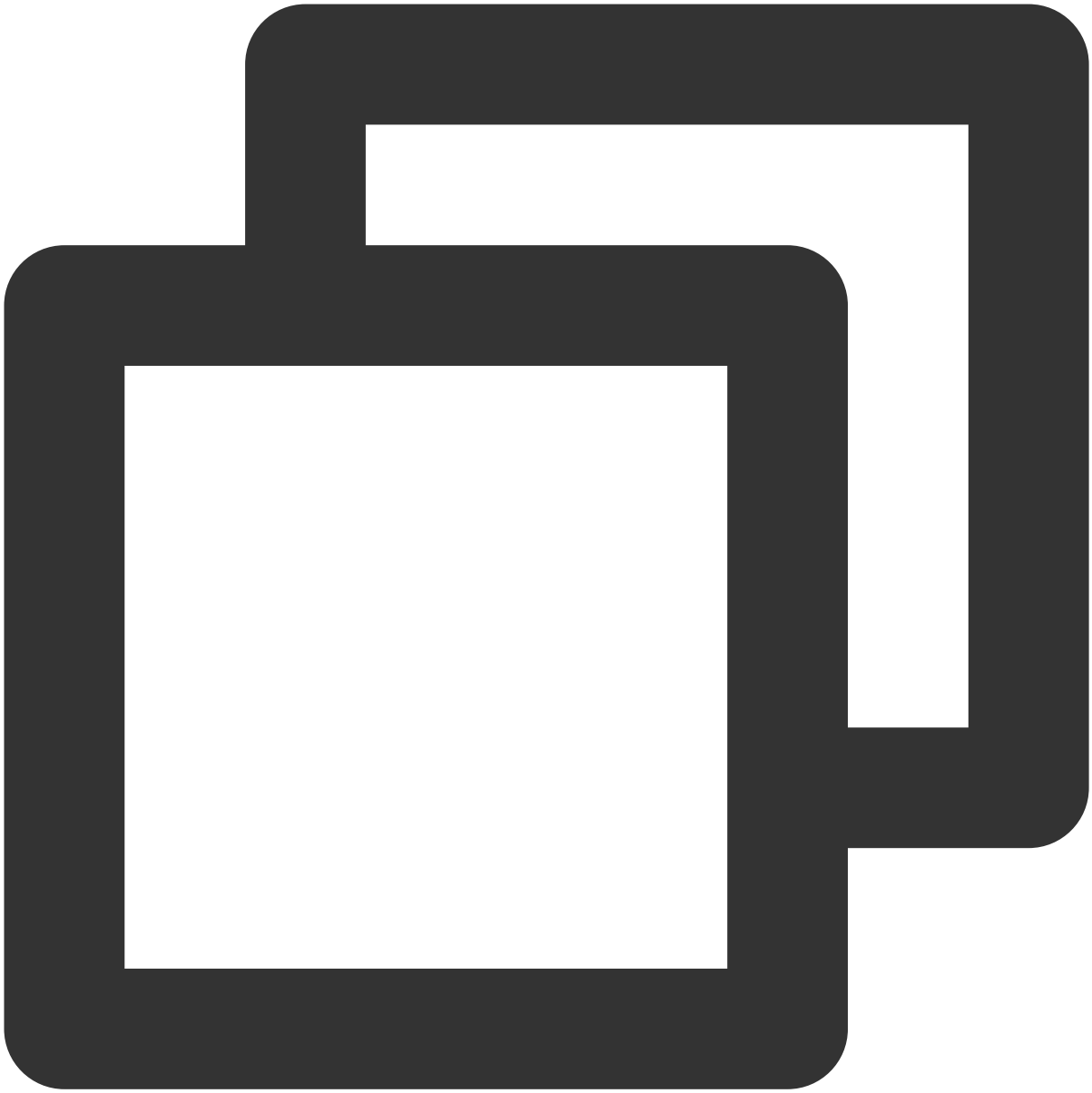
1. Log in to a CVM instance in the CVM console to complete the basic configurations of IPv6.
2. On the CVM instance, run the following commands in sequence to deploy and restart the Nginx service.



```
yum install nginx
service nginx restart
```

3. Check whether the Nginx service deployed on the CVM instance is listening on IPv6.

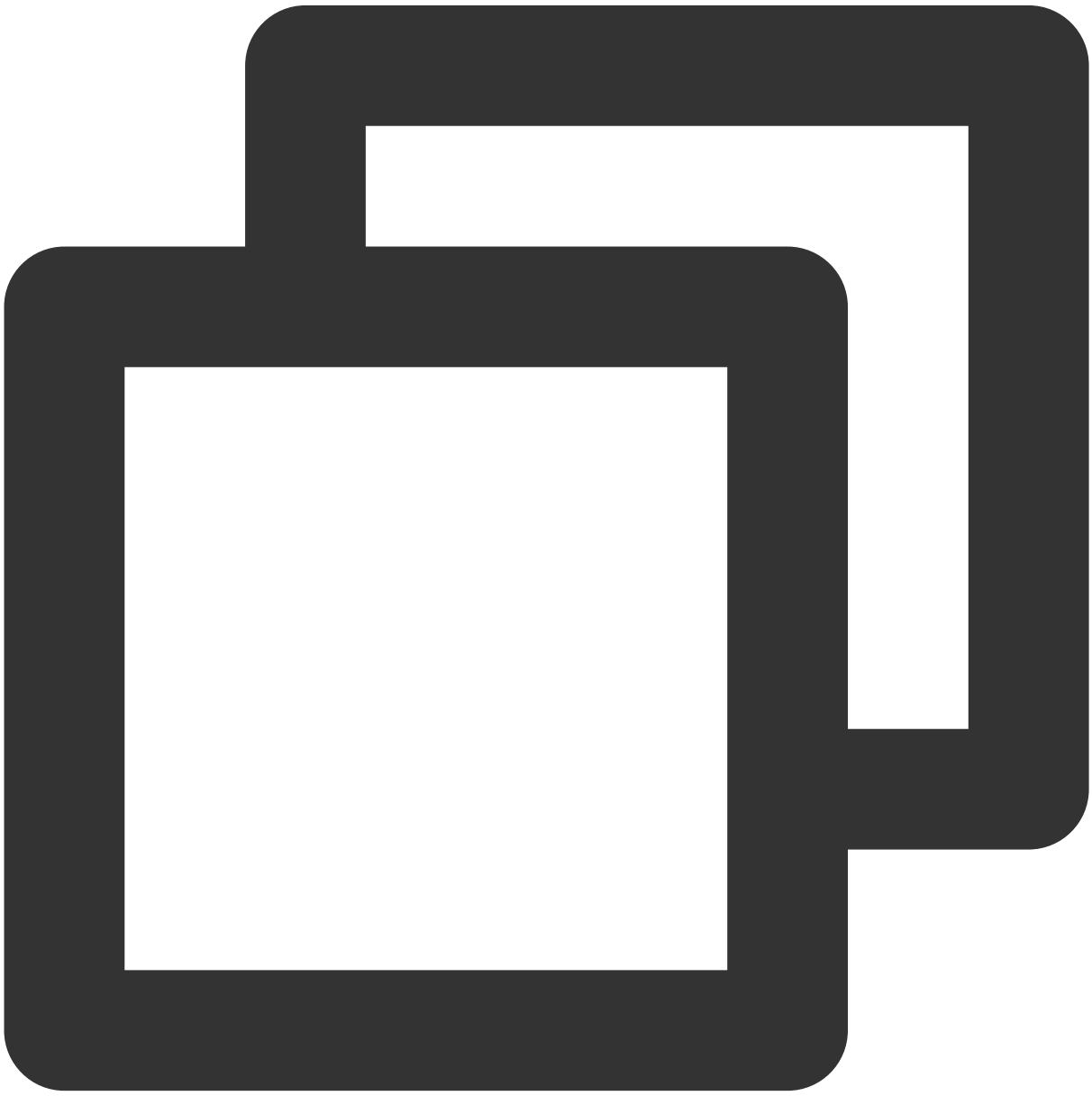
3.1 Run the following command for check.



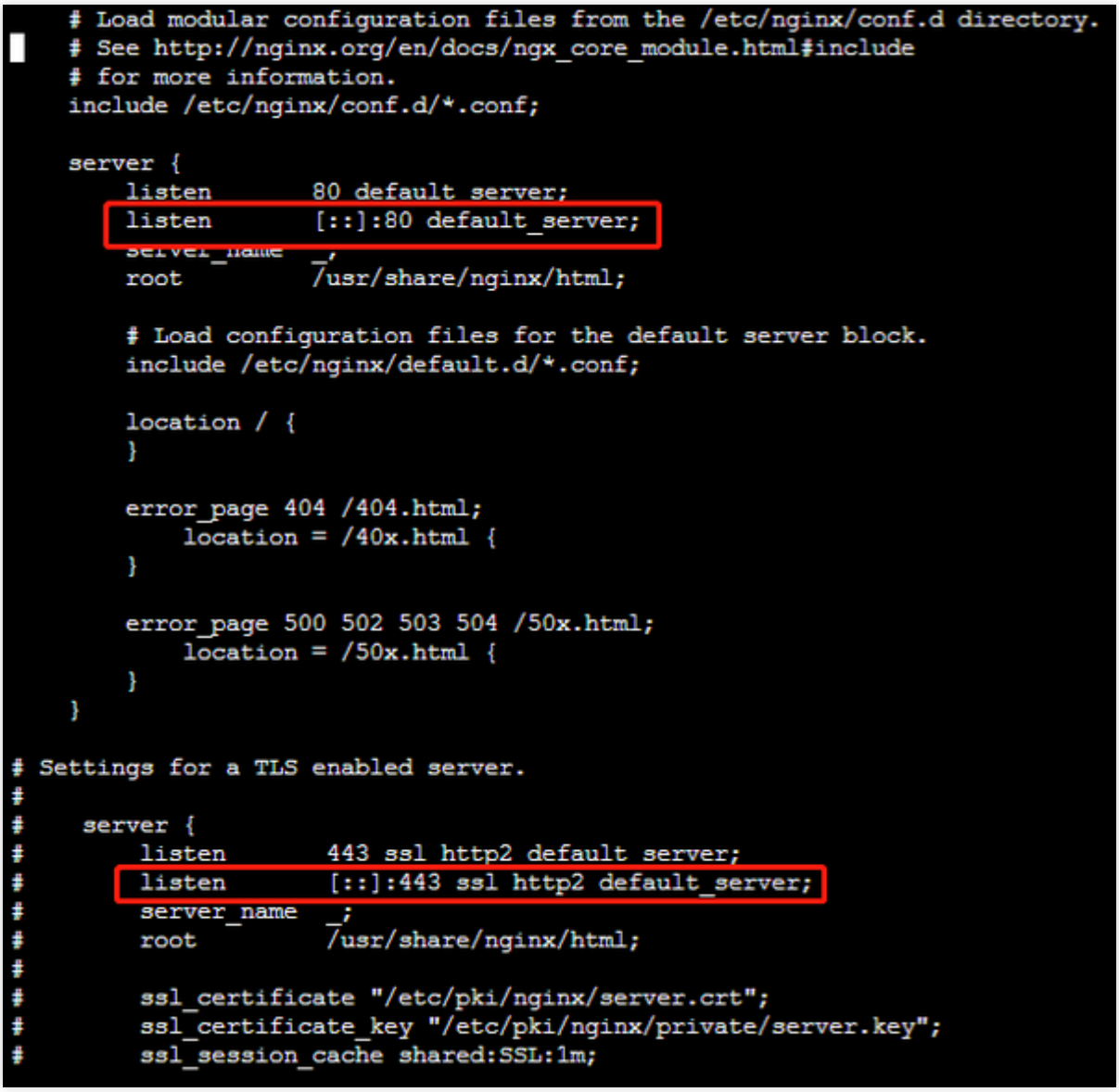
```
netstat -tupln
```

```
[root@VM_0_14_centos ~]# netstat -tupln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      4314/nginx: master
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      3175/sshd
tcp6       0      0 :::80                   :::*                    LISTEN      4314/nginx: master
udp        0      0 0.0.0.0:68              0.0.0.0:*                           2890/dhclient
udp        0      0 10.24.0.14:123          0.0.0.0:*                           3369/ntpd
udp        0      0 127.0.0.1:123           0.0.0.0:*                           3369/ntpd
udp        0      0 0.0.0.0:56713           0.0.0.0:*                           4333/ntpdate
udp6       0      0 fe80::5054:ff:fe3d::546 :::*                                4119/dhclient
udp6       0      0 2402:4e00:1400:1217:123 :::*                                3369/ntpd
udp6       0      0 fe80::5054:ff:fe3d::123 :::*                                3369/ntpd
udp6       0      0 ::1:123                 :::*                                3369/ntpd
```

3.2 Run the following command to open the Nginx configuration file for check.



```
vim  /etc/nginx/nginx.conf
```



Step 2. Create an IPv6 CLB instance

1. Log in to the Tencent Cloud console and go to the CLB purchase page.

2. Set the following parameters:

Billing Mode: supports pay-as-you-go billing only.

Region: select the target region.

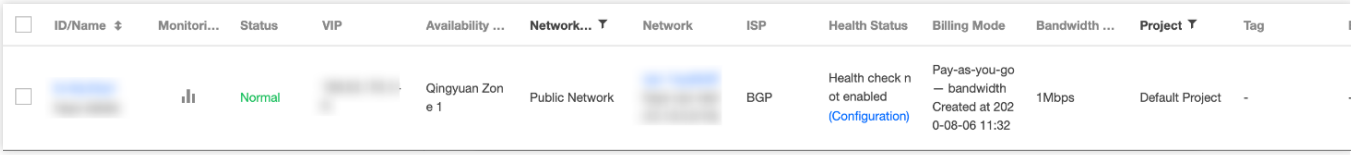
IP Version: IPv6.

ISP: BGP (multi-line).

Network: select a VPC and subnet that have already obtained IPv6 CIDR.

3. Select various configuration items on the purchase page and click **Buy now**.

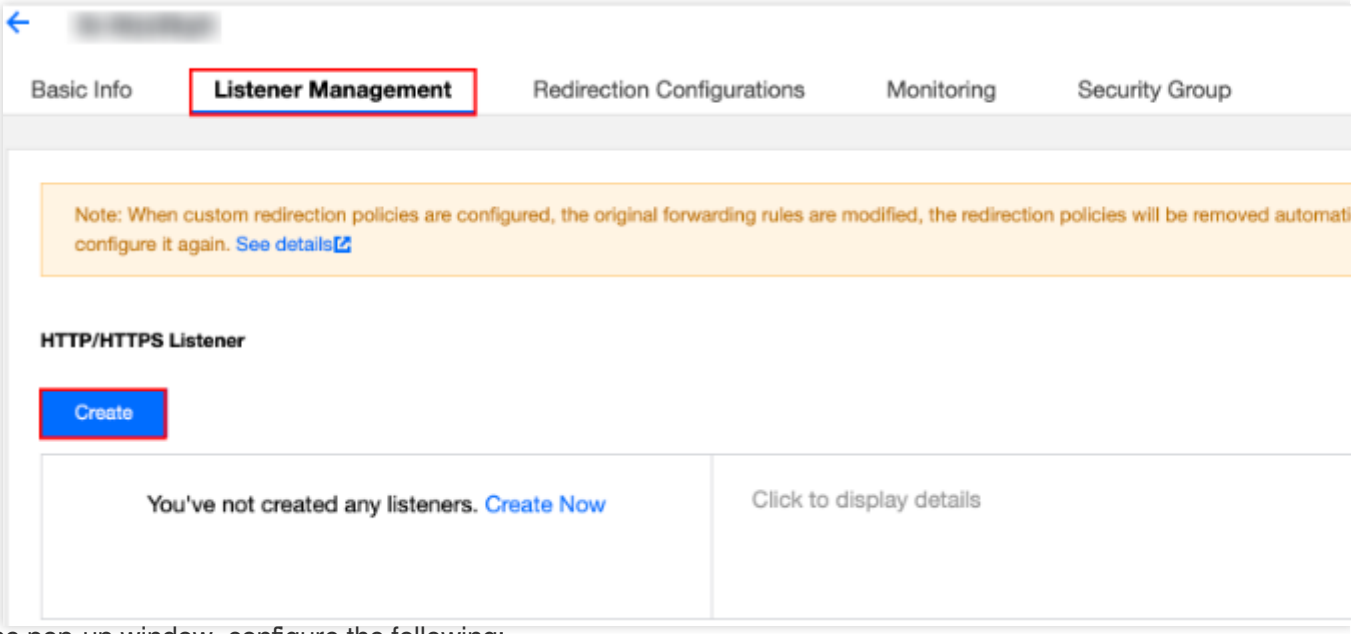
4. On the CLB instance list page, select the corresponding region to view the instance that you just created.



Step 3. Create an IPv6 CLB listener

Configuring the HTTP listening protocol and port

1. Log in to the CLB console.
2. In the CLB instance list, find the created CLB instance and click its ID to go to its details page.
3. In the **Basic information** module, you can click the modification icon next to the instance name to rename it.
4. On the **Listener management** tab, click **Create** in the **HTTP/HTTPS listener** section to create a CLB listener.



5. In the pop-up window, configure the following:

Set the name to `IPv6test` .

Set the listening protocol port to `HTTP:80` .

6. Click **Submit**.

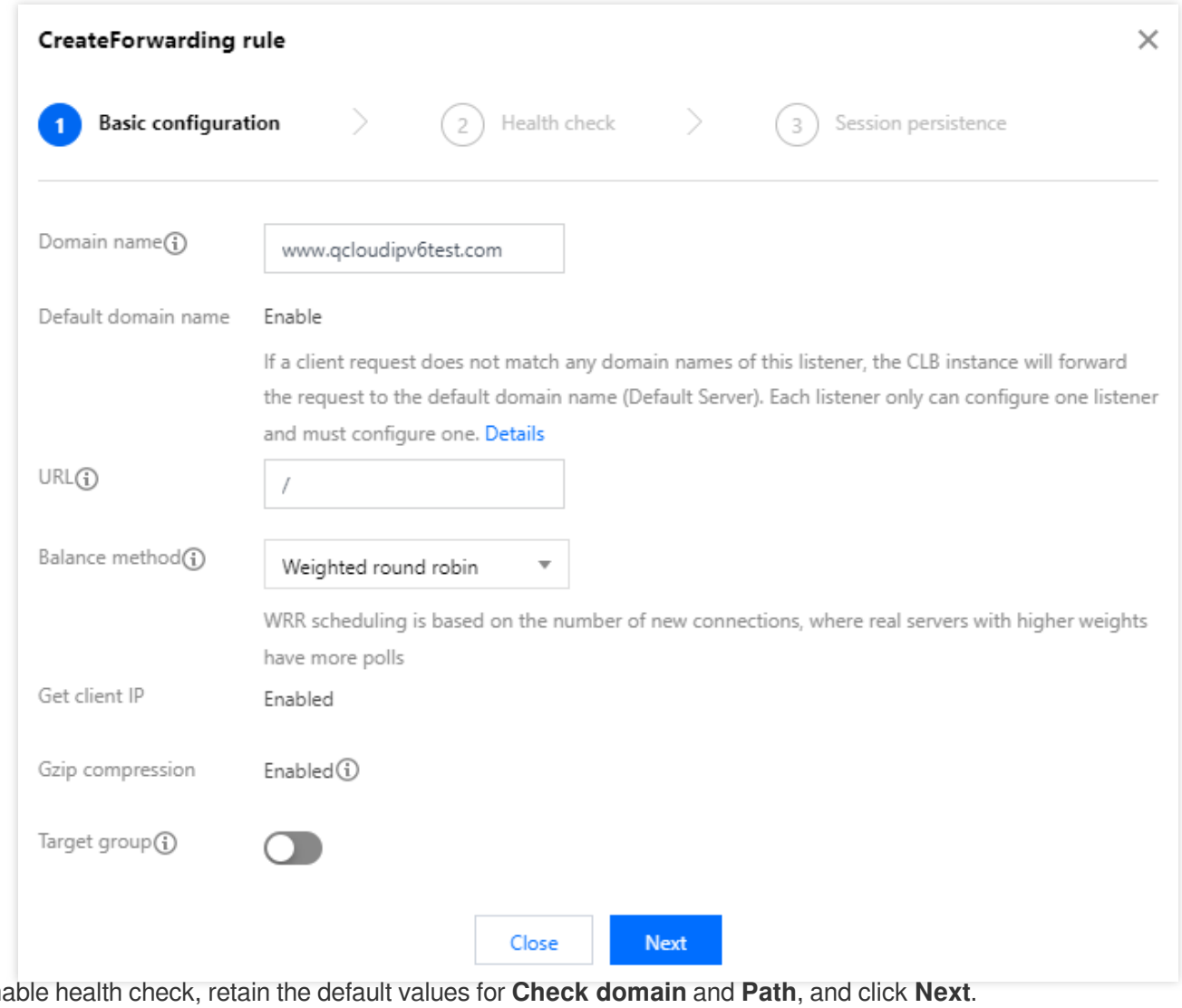
Configuring the listener's forwarding rule

1. In **Listener management**, select the new listener `IPv6test` and click **+** to add a rule.
2. In the pop-up window, configure the domain name, URL path, and balancing method, and click **Next**.

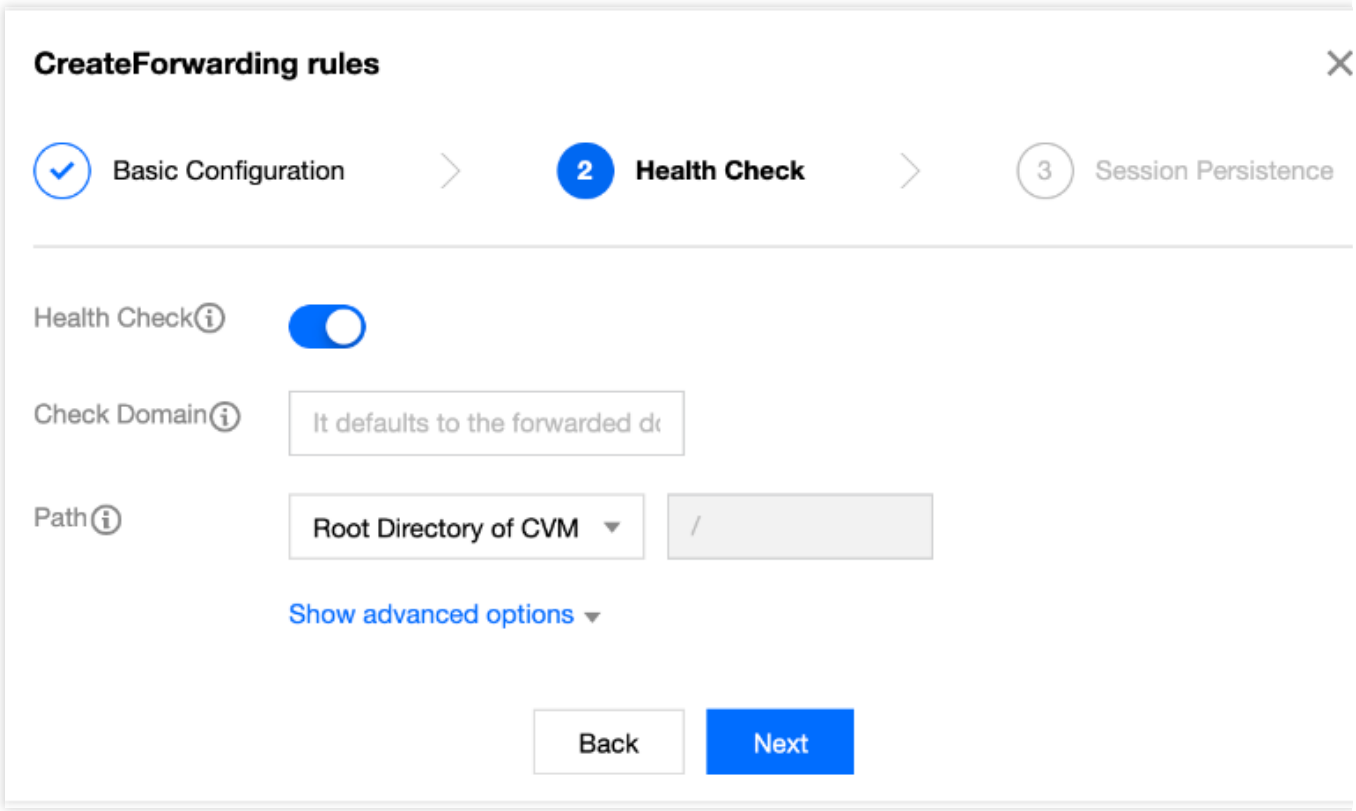
Domain name: Domain name used by your real server, which can contain a wildcard. In this example, `www.xxxxxxxxxtest.com` is used. For more information, see Layer-7 Domain Name Forwarding and URL Rules.

URL: Access path of your real server. `/` is used in this example.

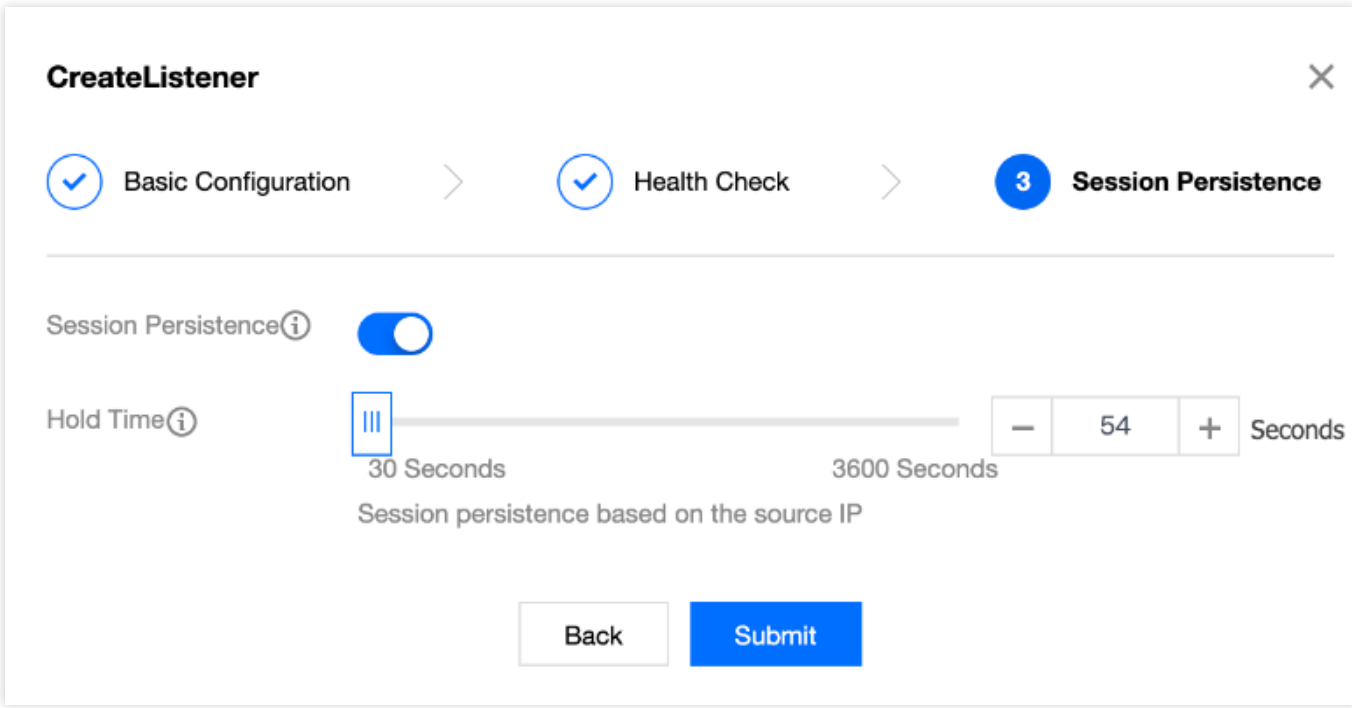
Balancing method: Select **Weighted round robin**.



3. Enable health check, retain the default values for **Check domain** and **Path**, and click **Next**.



4. Enable session persistence, configure the persistence period, and click **Submit**.



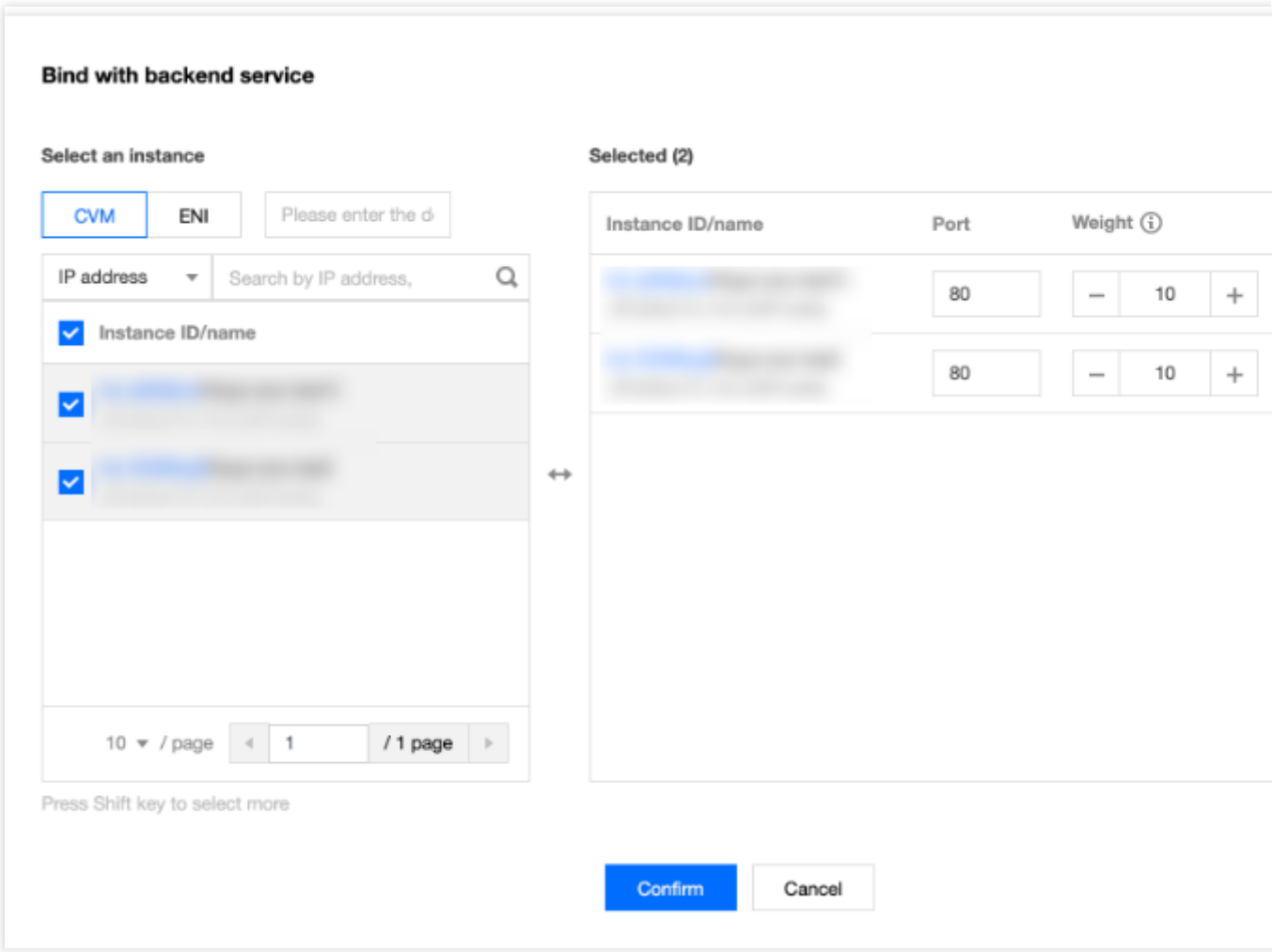
For more information about CLB listeners, see CLB Listener Overview.

Note:

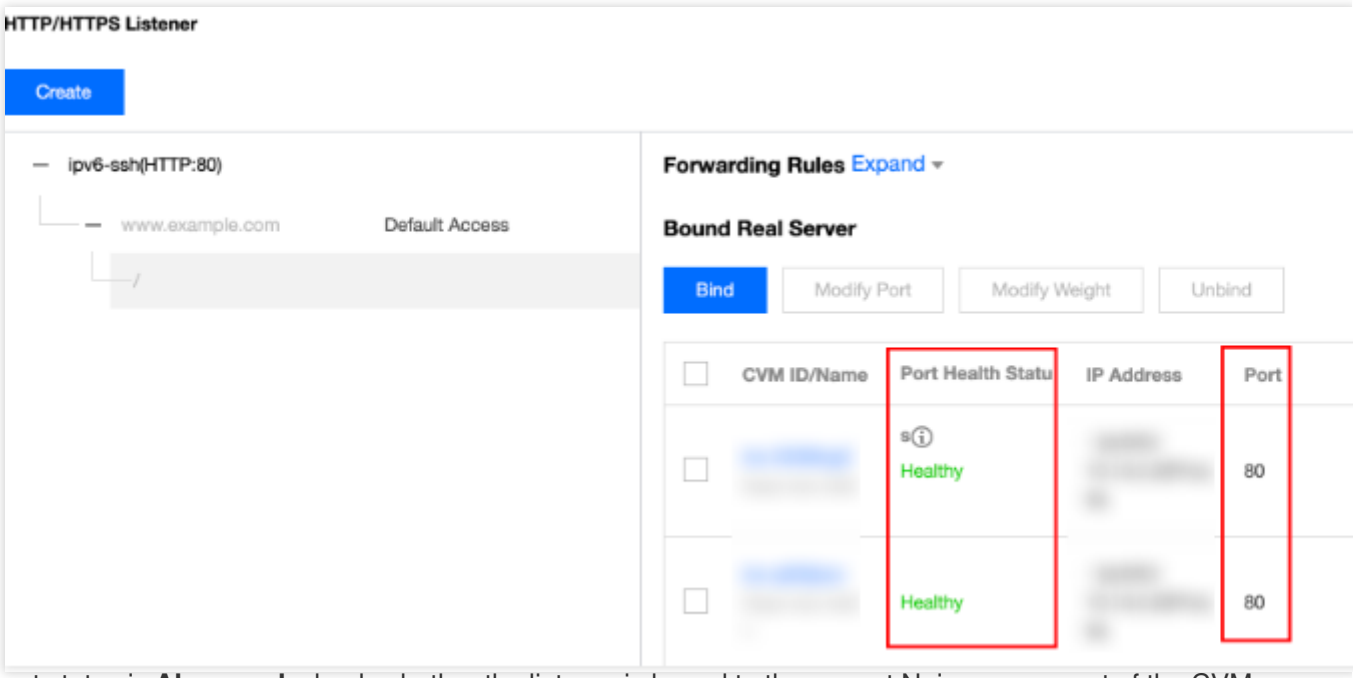
A listener (i.e., listening protocol:port) can be configured with multiple domain names, and a domain name can be configured with multiple URL paths. Select a listener or domain name and click **+** to create a new rule.
Session persistence: If session persistence is disabled and a round-robin method is used for scheduling, requests will be assigned to different real servers in sequence; if session persistence is enabled, or it is disabled but ip_hash scheduling is used, requests will always be assigned to the same real server.

Binding to a CVM instance

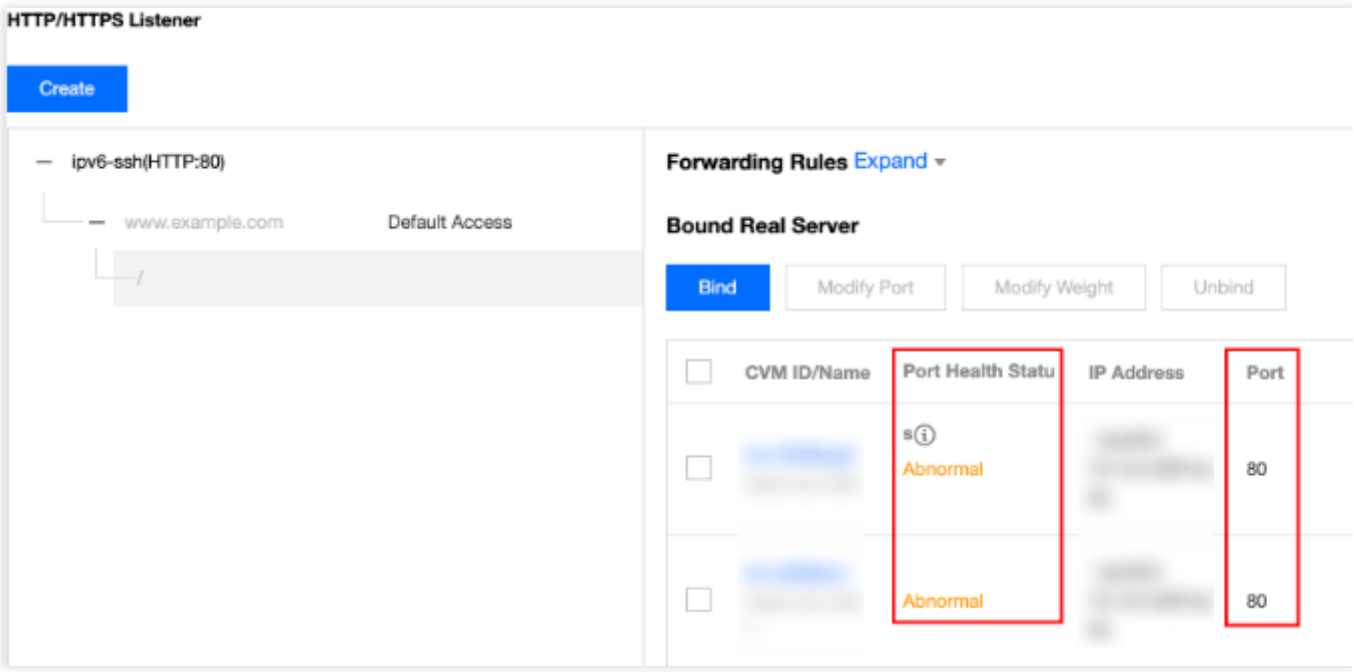
Note:
Before binding the listener to a CVM instance, make sure that the CVM instance has obtained an IPv6 address.
1. On the **Listener management** page, select and expand the listener that you just created and select the domain name and URL path. Then the IPv6 information of the CVM instance bound to the URL path will be displayed on the right. Click **Bind**.
2. In the pop-up window, select the CVM instance, set the default Nginx service port to 80, set the weight (10 by default), and click **OK**.



3. After the CVM instance is successfully bound, perform the following:
Check whether the port status is **Healthy**. If yes, proceed to Step 4. Test IPv6 CLB.



If the port status is **Abnormal**, check whether the listener is bound to the correct Nginx server port of the CVM instance, and log in to the CVM instance to check whether the port is normally listening on IPv6. You can perform the check as instructed in substep 3 in step 1.



Step 4. Test IPv6 CLB

After configuring an IPv6 CLB instance, you can verify whether the architecture takes effect by checking whether different domain names and URLs under a CLB instance can access different real servers, i.e., checking whether the content-based routing feature is available.

Use a client with IPv6 public network access capabilities to access the domain name or IPv6 address of the CLB instance. If it can properly access the web service of the CVM instance, the IPv6 CLB instance is working normally.

1. Go to the Tencent Cloud domain registration page to query and register a domain name. `xxxxxxxxxxtest.com` is used in this example.
2. Log in to the DNSPod console, click the **domain name** you just purchased, and click **Add Record** on the **Record Management** page to add an AAAA record to the domain name. Enter and save the following content:

Host Record: Domain name prefix. Set it to `www` in this example.

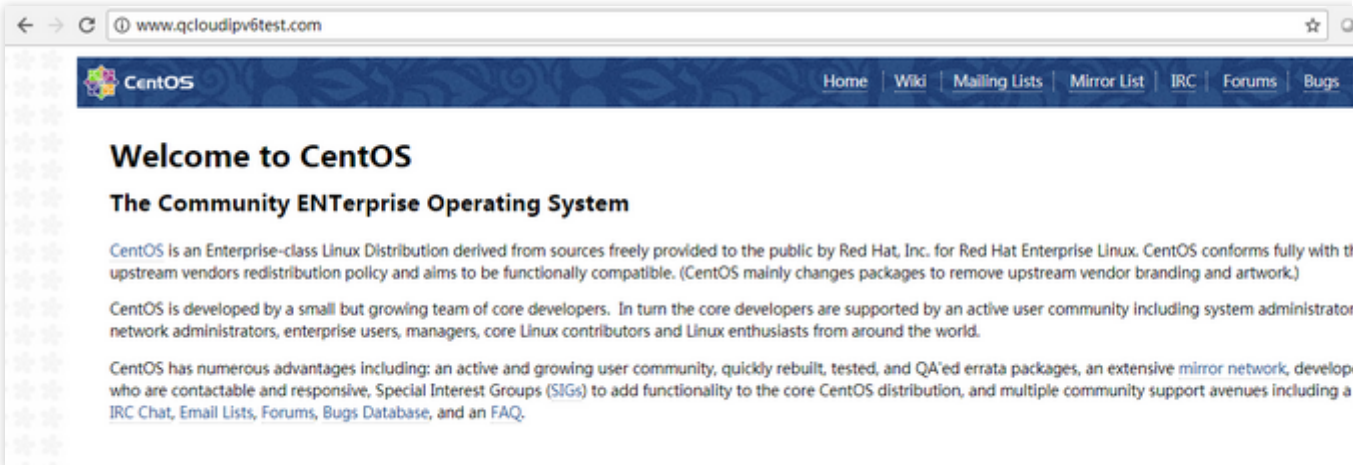
Record Type: Select **AAAA record**.

Split Zone: Select **Default**.

Record Value: Enter the IPv6 address of the CLB instance.

TTL: Leave it as the default value **600s**.

3. After adding the domain name resolution, ping the domain name to verify it.
4. You can use a browser to access the domain name to verify it.



Getting Started with Classic CLB

Last updated：2024-01-04 09:44:16

This document describes how to create a public network Classic CLB instance named `clb-test` and forward requests from clients to two real servers.

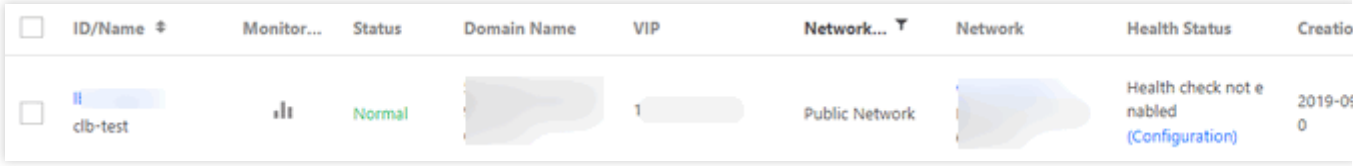
Prerequisites

1. CLB only forwards traffic but cannot process requests; therefore, you need to have a CVM instance that processes user requests.
In this example, two CVM instances are enough, but you can also configure more instances. CVM instances `rs-1` and `rs-2` have been created in the Guangzhou region in this example. For more information on how to create a CVM instance, please see Purchasing and Launching CVM Instances.
2. This document takes HTTP forwarding as an example. The corresponding web server (such as Apache, Nginx, or IIS) must be deployed on the CVM instance.
To verify the result, in this example, Apache is deployed on both `rs-1` and `rs-2` . Apache returns "Hello Tomcat! This is rs-1!" on `rs-1` and "Hello Tomcat! This is rs-2!" on `rs-2` . For more information on how to deploy components on a CVM instance, please see Deploying Java Web on Linux (CentOS) and Installing and Configuring PHP on Windows.
3. Access the public IP and path of your CVM instances. If the deployed page is displayed, the service has been successfully deployed.
Note:
For traditional accounts, public network bandwidth must be purchased for the CVM instances, as the bandwidth is billed by CVM rather than CLB. You can determine the account type as instructed in Billing Overview.
In this example, the values returned by the service deployed on two real servers are different. In actual scenarios, to ensure that all users have a uniform experience, generally the same service should be deployed on all real servers.

Purchasing Classic CLB Instance

1. Log in to Tencent Cloud's official website and go to the CLB purchase page.
2. In this example, select **Guangzhou** as the region, which is the same as that of the CVM instances. Select **Classic** as the instance type, **Public Network** as the network attribute, and **Default-VPC (Default)** as the network and enter "clb-test" as the instance name.
3. Click **Buy Now** and make the payment. For more information on CLB instances, please see Product Attribute Selection.

4. On the "CLB Instance List" page, select the corresponding region to view the instance just created.



Creating CLB Listener

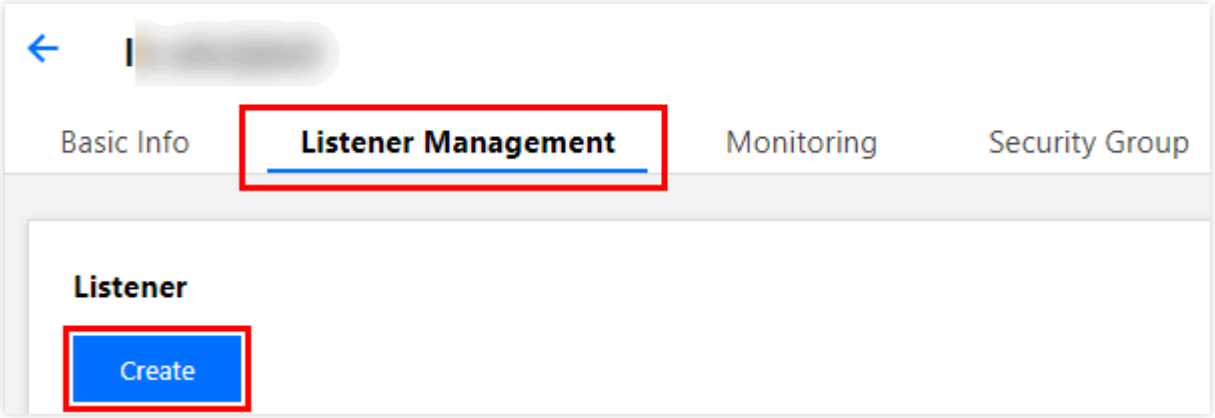
A CLB listener forwards requests by specifying protocols and ports. This document takes configuration of forwarding HTTP client requests by CLB as an example.

1. Log in to the CLB Console.

2. In the "CLB Instance List", find the created Classic CLB instance `clb-test` and click its ID to enter its details page.

3. In the "Basic Info" section, you can click "Edit" next to the instance name to rename it.

4. In **Listeners** in "Listener Management", click **Create** to create a CLB listener.



5. In the pop-up box, configure the following:

Set the name to "Listener1".

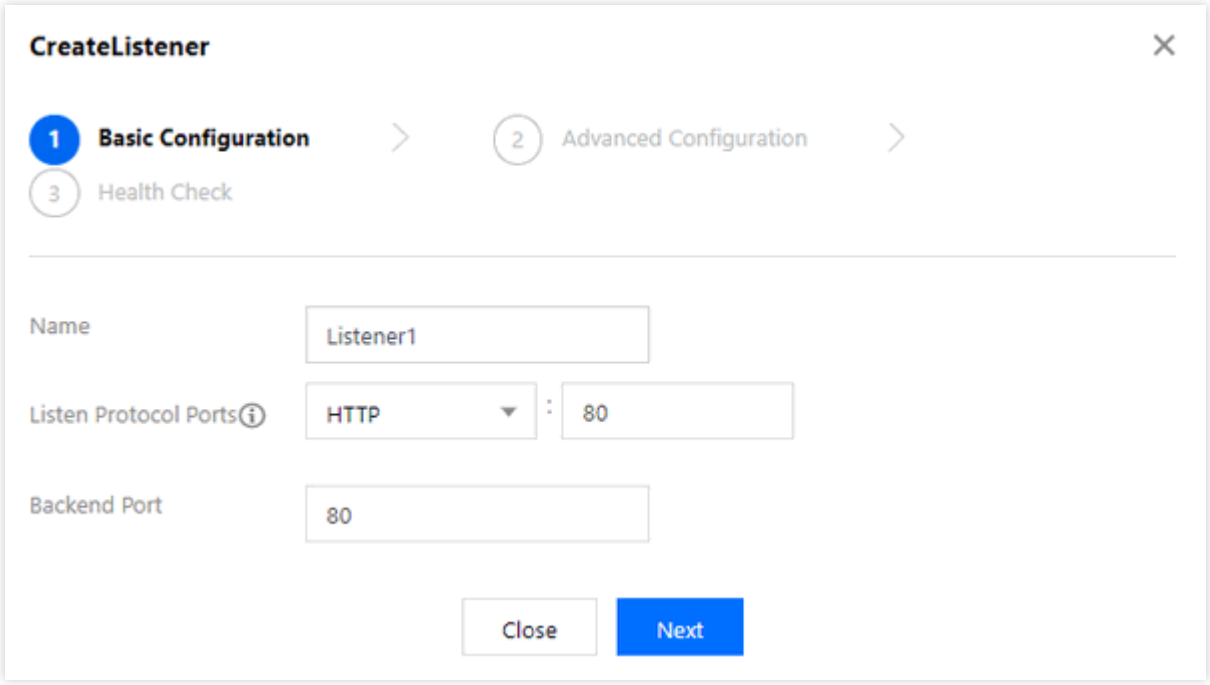
Set the listener protocol and port to `HTTP:80` .

Set the backend port to `80` .

Select "WRR" as the load balancing mode.

Do not check session persistence.

Enable health check.

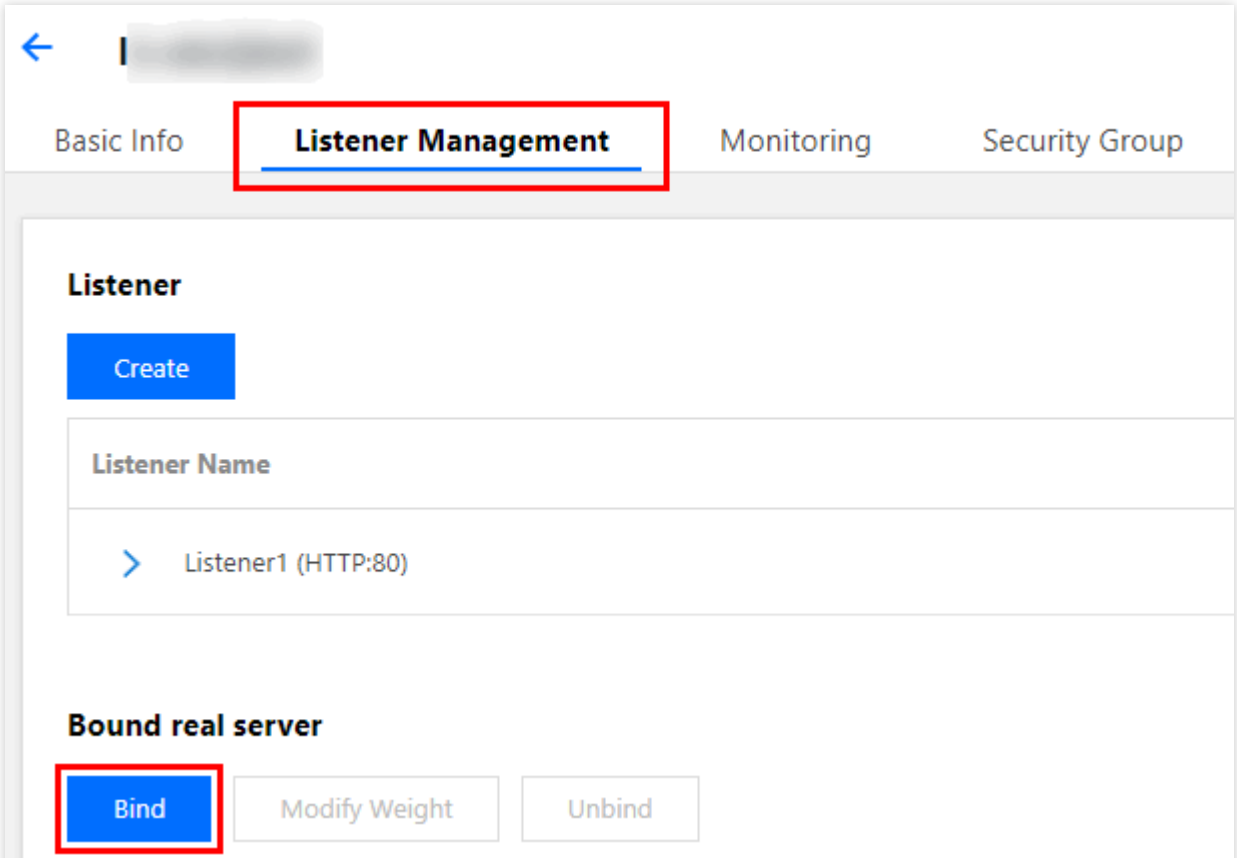


6. Click **Complete** to create the CLB listener.

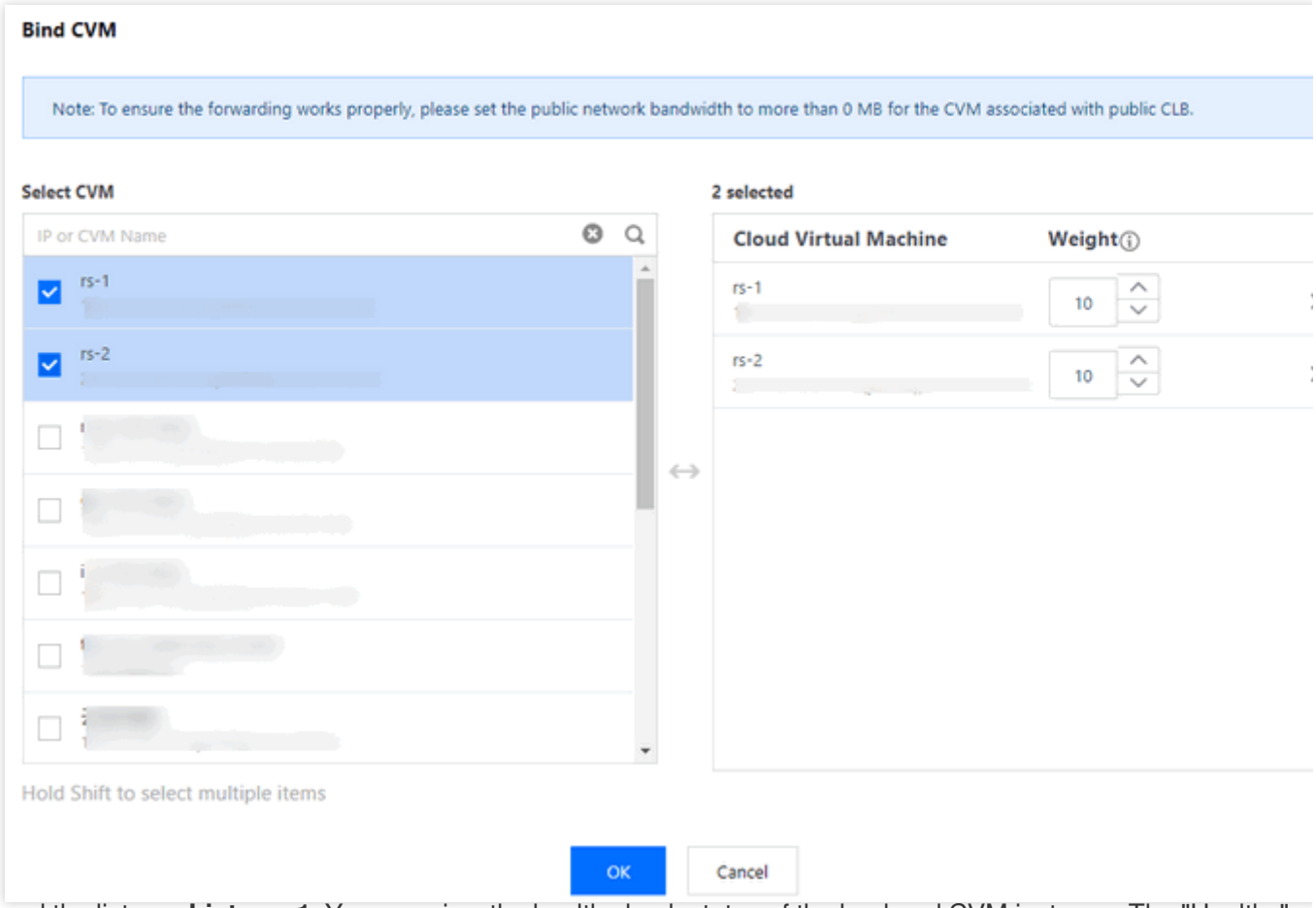
For more information on CLB listeners, please see CLB Listener Overview.

Binding Real Server

1. In the "CLB Instance List", find the created `clb-test` and click its ID to enter its details page.
2. In the "Bind Real Server" module in "Listener Management", click **Bind**.



3. In the pop-up box, select CVM instances `rs-1` and `rs-2` in the same region as the CLB instance and keep the default weight "10" for them.
4. Click **OK** to complete binding.



5. Expand the listener **Listener1**. You can view the health check status of the backend CVM instance. The "Healthy" status indicates that the CVM instance can properly process requests forwarded by CLB.

Configuring Security Group

After creating a CLB instance, you can configure a CLB security group to isolate public network traffic. For more information, please see Configuring CLB Security Group.

After configuring a security group, you can choose to enable or disable "Allow Traffic by Default in Security Group" with different configurations as follows:

Method 1. Enable "Allow Traffic by Default in Security Group"

Note:

This feature is currently in beta test. To try it out, please submit a ticket for application. This feature is not supported for classic private network CLB.

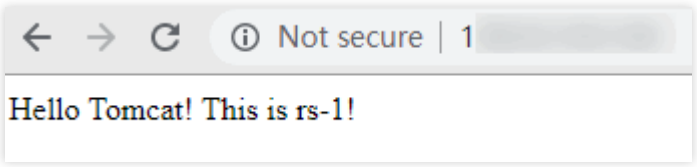
For detailed directions, please see Configuring CLB Security Group.

Method 2. Allow the client IP in the CVM security group

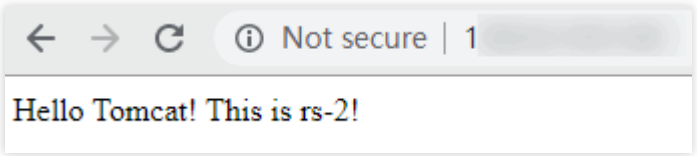
For detailed directions, please see Configuring CLB Security Group.

Verifying CLB Service

1. Enter the CLB service address and port `http://vip:80` in a browser to test the CLB service. If a message is displayed as shown below, the request has been forwarded to the CVM instance `rs-1` by CLB, and the CVM instance has properly processed the request and returned the result.



2. The round robin algorithm of the listener is "weighted round robin", and the weights of the two CVM instances are both "10". If you refresh the webpage in the browser to send a new request, you can see that the request is forwarded to the CVM instance `rs-2` by CLB.



Note:
If session persistence is disabled and a round-robin method is used for scheduling, requests will be assigned to different real servers in sequence.
If session persistence is enabled, or it is disabled but ip_hash scheduling is used, requests will always be assigned to the same real server.

Deploying Nginx on CentOS

Last updated：2024-01-04 09:44:16

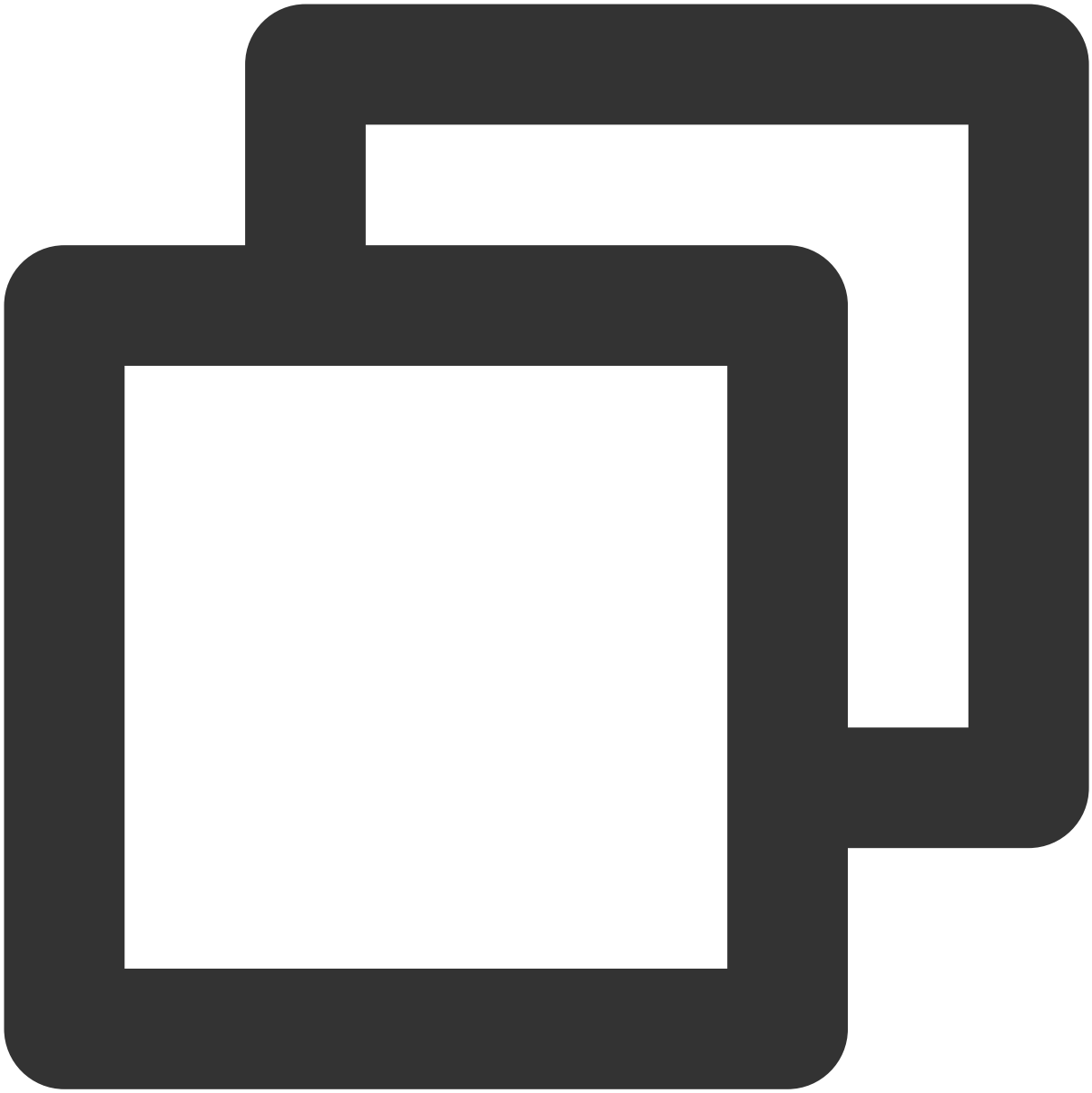
This document describes how to deploy Nginx projects on CentOS and is suitable for new individual users of Tencent Cloud.

Software Version

The versions of software tools used in this document are as follows, which may be different from your software versions during actual operations.
Operating system: CentOS 7.5
Nginx: Nginx 1.16.1

Installing Nginx

1. After completing the purchase, click **Log in** on the CVM details page to log in to the CVM instance and then enter your username and password to set up an Nginx environment. For more information on how to create a CVM instance, please see Creating CVM Instances.

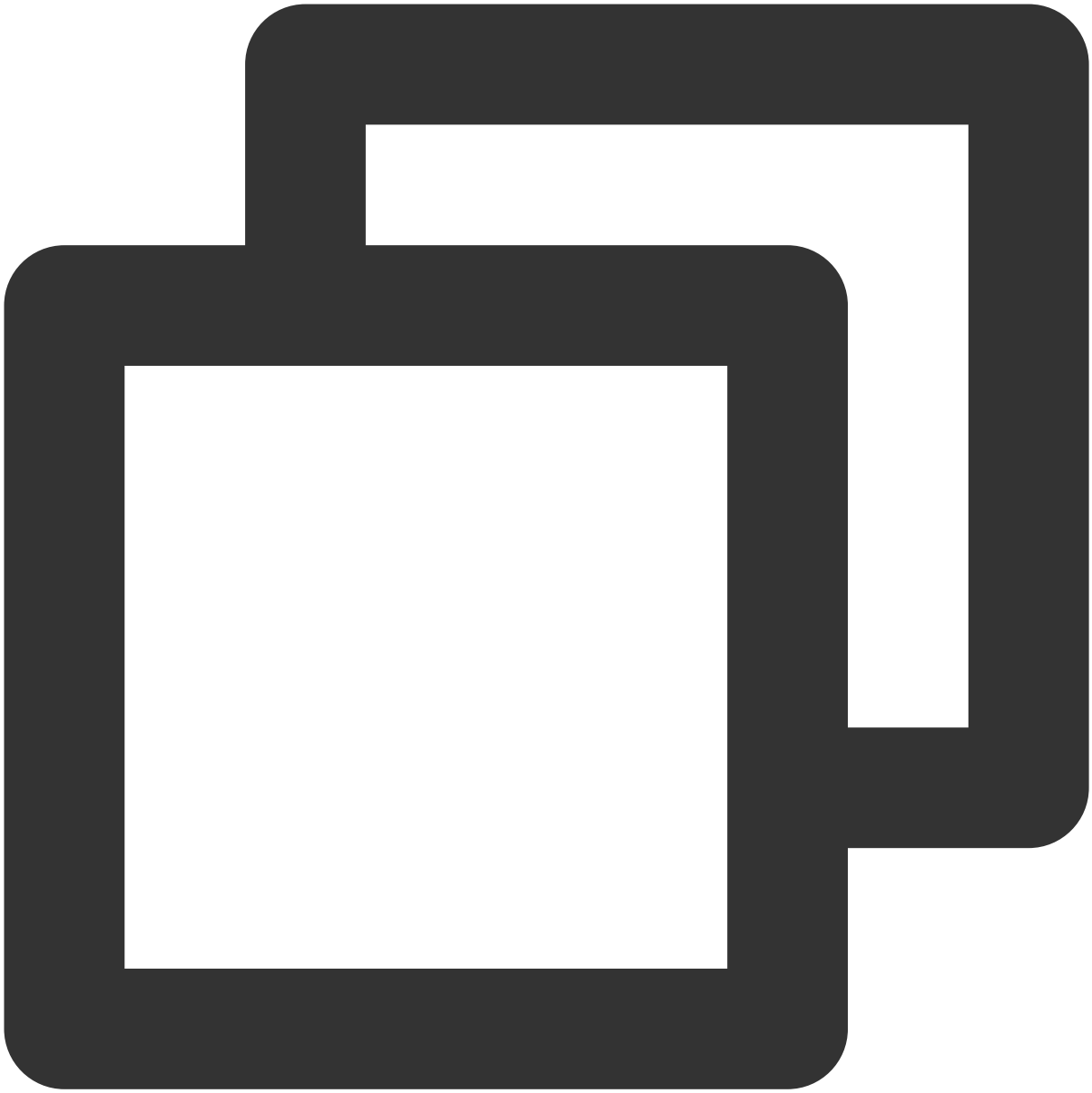


```
# Install Nginx
yum -y install nginx
# View Nginx version
nginx -v
# View Nginx installation directory
rpm -ql nginx
# Start Nginx
service nginx start
```

2. Access the public IP address of the CVM instance and if the following page appears, Nginx is successfully deployed:

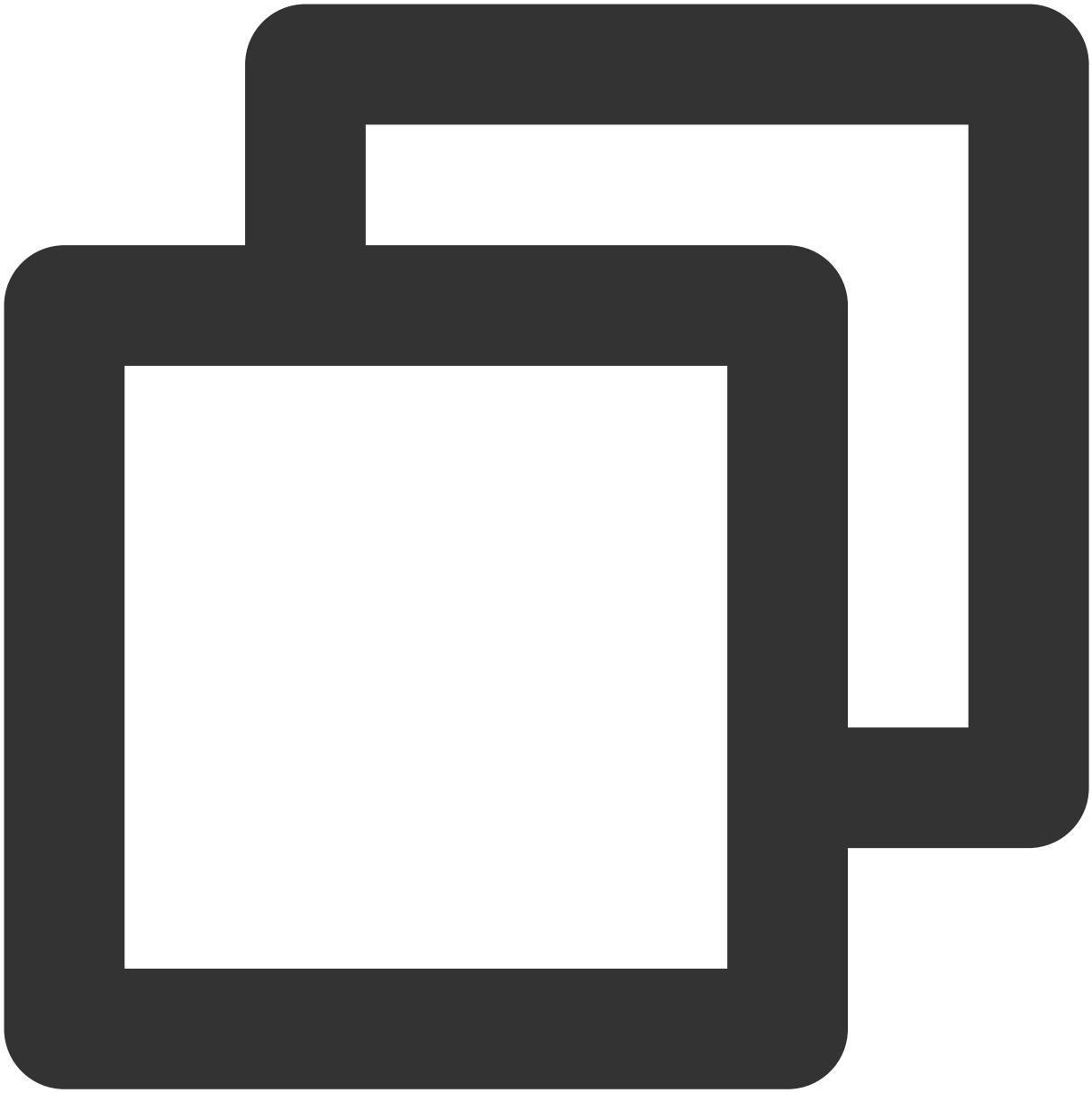
3. The default root directory of Nginx is `/usr/share/nginx/html`. Modify the `index.html` static page in the `html` directory to mark the specialness of this page. Relevant operations are as follows:

3.1 Run the following command to enter the `index.html` static page in `html`:

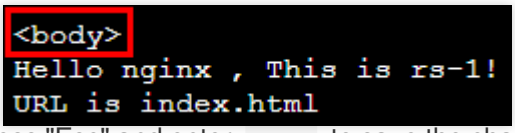


```
vim /usr/share/nginx/html/index.html
```

3.2 Press "i" to enter the editing mode and add the following in the `<body></body>` tag:



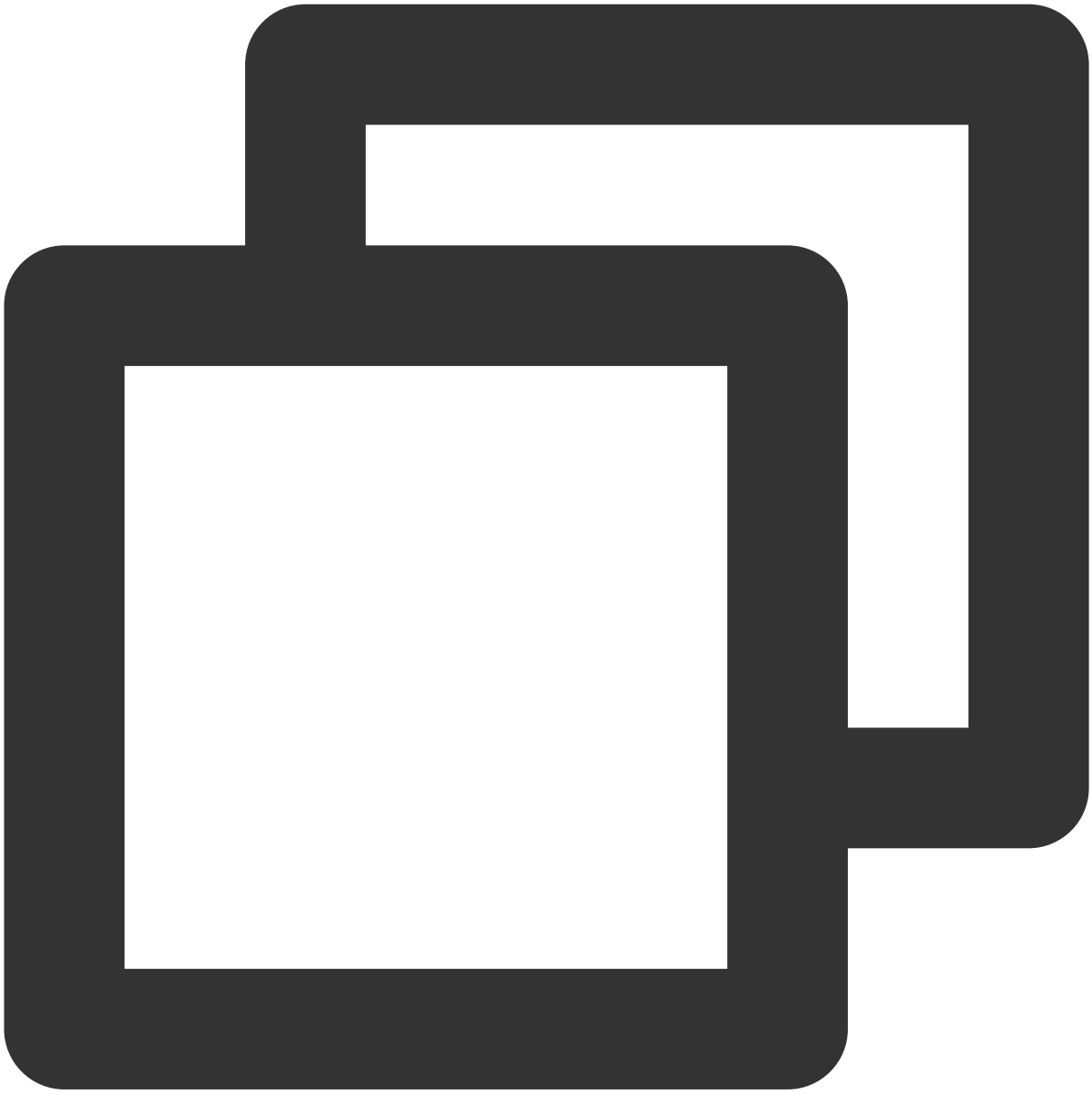
```
# You are recommended to enter directly under `<body>`
Hello nginx , This is rs-1!
URL is index.html
```



3.3 Press "Esc" and enter `:wq` to save the change.

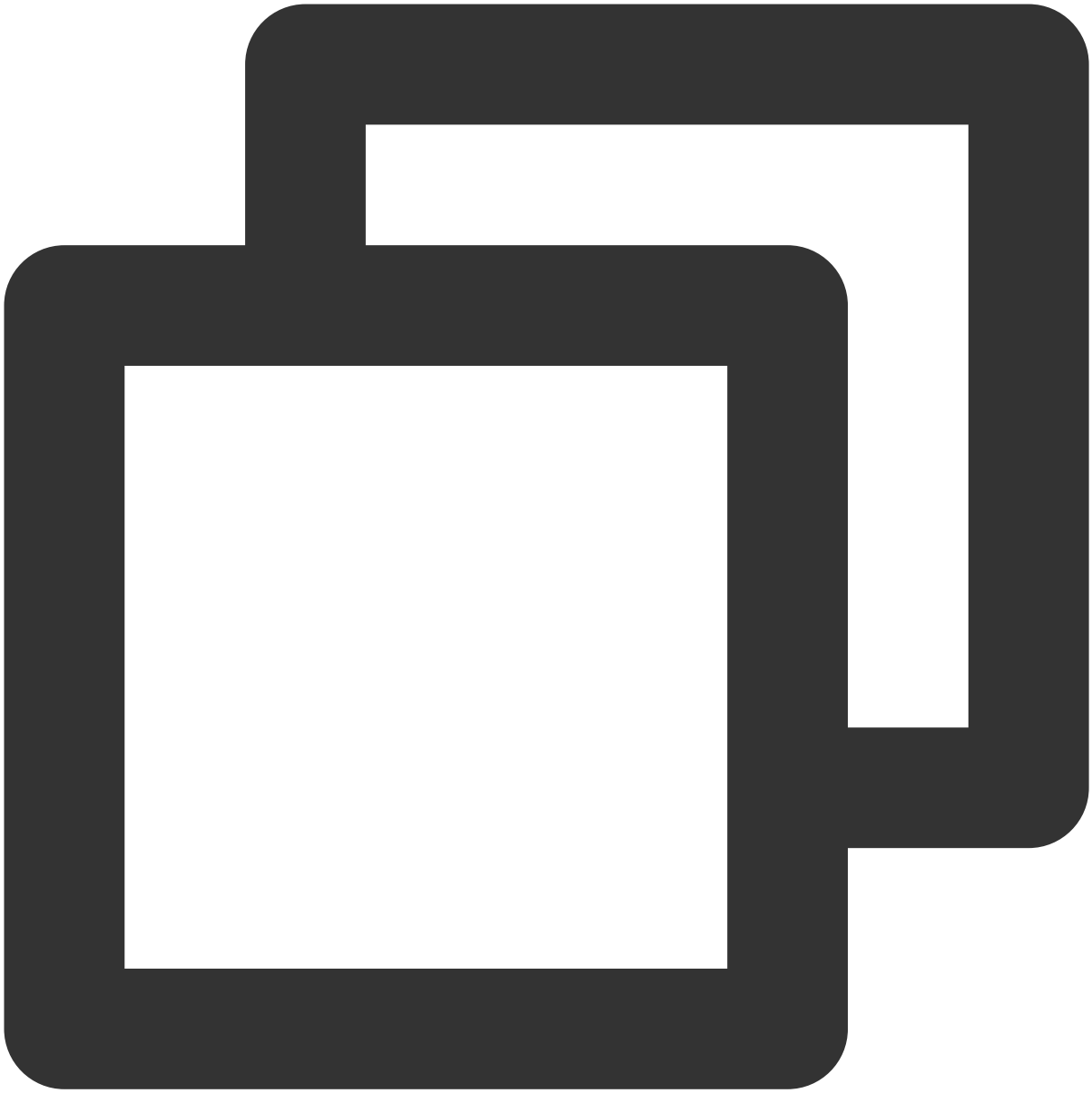
4. CLB (formerly "Application CLB") can forward requests according to the real server path and deploy a static page in the `/image` path. Relevant operations are as follows:

4.1 Run the following commands to create and enter an `image` directory:



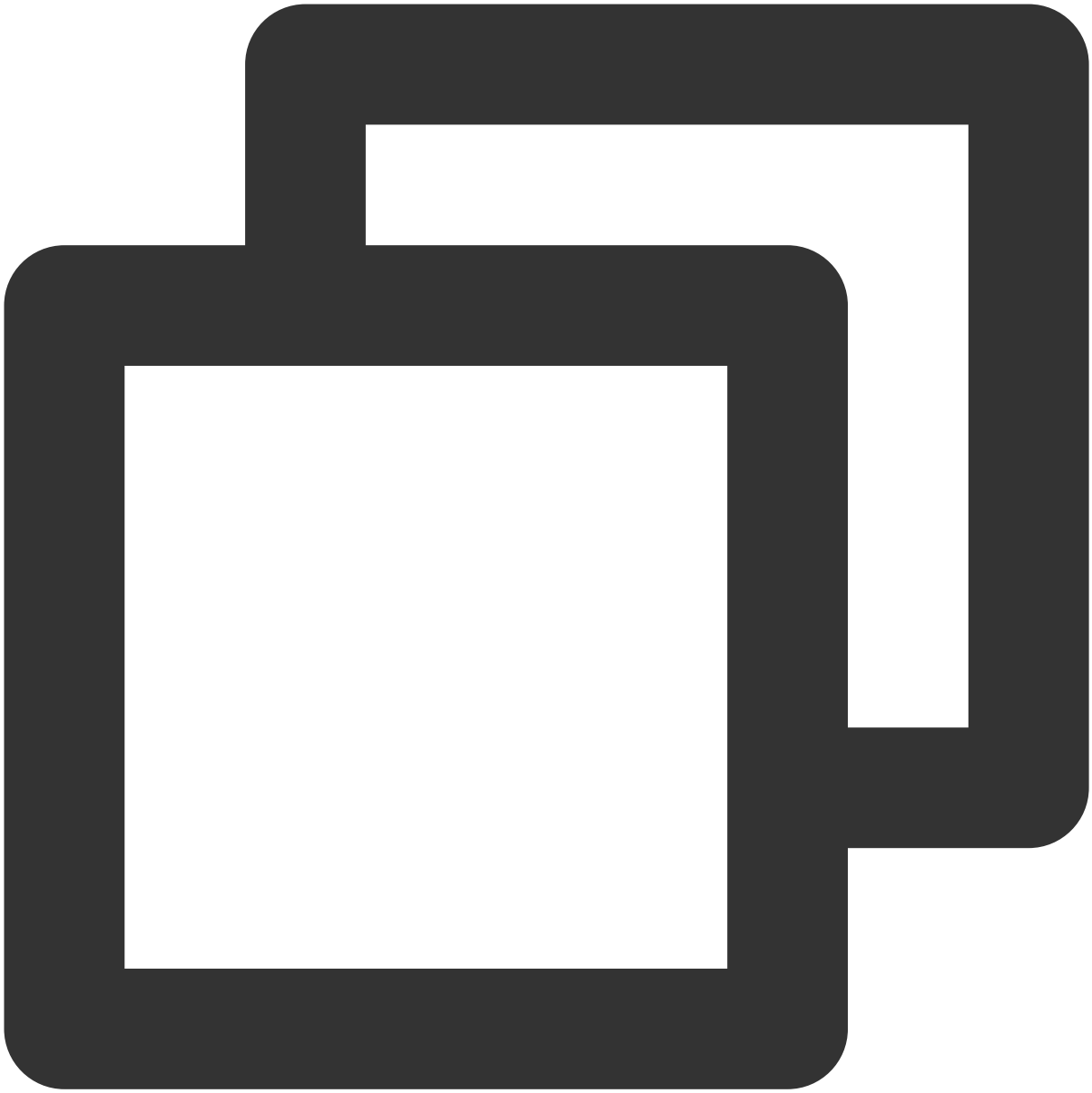
```
mkdir /usr/share/nginx/html/image
cd /usr/share/nginx/html/image
```

4.2 Run the following command to create an `index.html` static page in the `image` directory:



```
vim index.html
```

4.3 Press "i" to enter the editing mode and add the following in the page:



```
Hello nginx , This is rs-1!
URL is image/index.html
```

4.4 Press "Esc" and enter `:wq` to save the change.

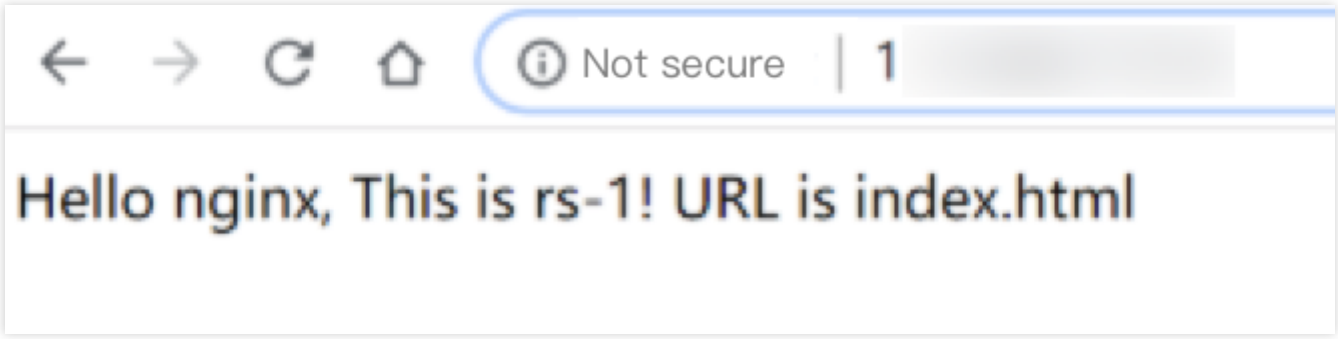
Note:

The default port of Nginx is `80` . To change the port, please modify the configuration file and restart Nginx.

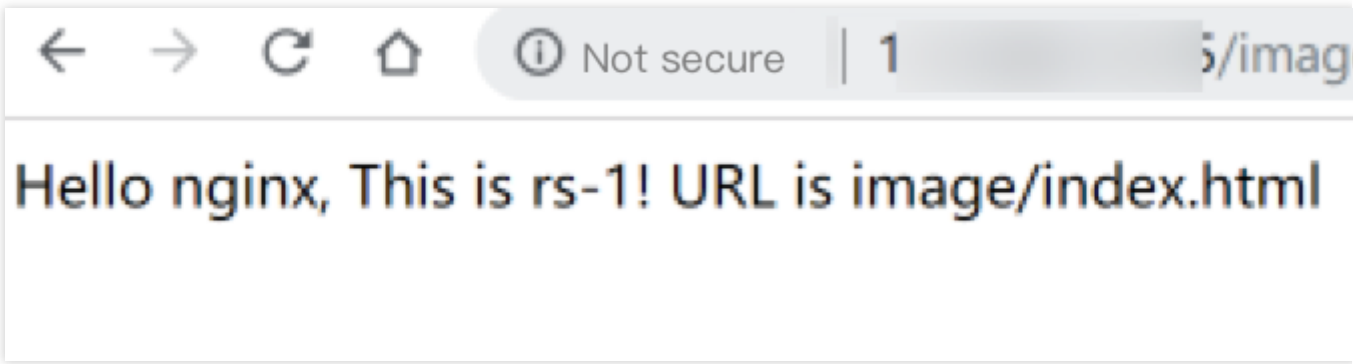
Verifying the Nginx Service

Access the public IP and path of your CVM instance. If the deployed static page is displayed, Nginx has been successfully deployed.

`index.html` page of `rs-1` :



`/image/index.html` page of `rs-1` :



Deploying Java Web on CentOS

Last updated：2021-07-06 19:57:04

This document describes how to deploy Java Web projects on CentOS and is suitable for new individual users of Tencent Cloud.

Software Version

The versions of software tools used in this document are as follows, which may be different from your software versions during actual operations.

- Operating system: CentOS 7.5
- Tomcat: apache-tomcat-8.5.39
- JDK: JDK 1.8.0_201

Installing JDK

After purchasing the CVM, you can click **Login** on the CVM details page to log in to your CVM instance where you can enter your username and password to set up the Java web environment. For more information on how to create a CVM instance, please see CVM - Creating Instance.

Downloading JDK

Enter the following command:

```
mkdir /usr/java # Create a `java` folder
cd /usr/java # Enter the `java` folder
```

```
# Upload JDK installation package (recommended)
You are recommended to use tools such as WinSCP to upload the JDK installation pa
ckage to the above `java` folder and then decompress it.
Or
# Use a command (you are recommended to upload the installation package): run `wg
et` to download the package, which cannot be decompressed because a downloaded pa
ckage declines the Oracle BSD License by default. Please go to https://www.oracl
e.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html to accept the
license agreement and obtain the download link with your cookies.
wget --no-check-certificate --no-cookies --header "Cookie: oraclelicense=accept-s
```

```
ecurebackup-cookie" https://download.oracle.com/otn-pub/java/jdk/8u201-b09/42970487e3af4f5aa5bca3f542482c60/jdk-8u201-linux-x64.tar.gz

# Decompress
chmod +x jdk-8u201-linux-x64.tar.gz
tar -xzvf jdk-8u201-linux-x64.tar.gz
```

Setting environmental variable

1. Open the `/etc/profile` file.

```
vi /etc/profile
```

2. Press I to enter the editing mode and add the following information to the file.

```
# set java environment
export JAVA_HOME=/usr/java/jdk1.8.0_201
export CLASSPATH=$JAVA_HOME/lib/tools.jar:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib
export PATH=$JAVA_HOME/bin:$PATH
```

3. Press Esc to exit the editing mode and enter `:wq` to save and close the file.
4. Load the environmental variable.

```
source /etc/profile
```

Viewing JDK installation result

Run the `java -version` command. If the JDK version information is displayed, JDK has been successfully installed.

```
[root@emma /]# java -version
java version "1.8.0_201"
Java(TM) SE Runtime Environment (build 1.8.0_201-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.201-b09, mixed mode)
```

Installing Tomcat

Downloading Tomcat

Enter the following commands:

```
# The mirror address may change and the Tomcat version may be continuously upgrad
ed. If the download link expired, please go to [Tomcat official website](https://
tomcat.apache.org/download-80.cgi) and select an appropriate installation package
address.
wget http://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-8/v8.5.39/bin/apach
e-tomcat-8.5.39.tar.gz
tar -xzvf apache-tomcat-8.5.39.tar.gz
mv apache-tomcat-8.5.39 /usr/local/tomcat/
```

The following files are in the `/usr/local/tomcat/` directory:

- bin: script file, which contains scripts for starting and stopping the Tomcat service.
- conf: global configuration files, of which the most important ones are `server.xml` and `web.xml` .
- webapps: the main web release directory in Tomcat, which is the default directory for storing web application files.
- logs: Tomcat log files.

Note：

If the download link expired, please replace it with the latest link at Tomcat's official website.

Adding user

```
# Add a general user `www` to run Tomcat
useradd www
# Create a website root directory
mkdir -p /data/wwwroot/default
# Upload the Java web project file (WAR package) to the website root directory an
d modify the file permission under the directory to `www`. This example shows how
to create a Tomcat test page in the website root directory:
echo Hello Tomcat! > /data/wwwroot/default/index.jsp
chown -R www.www /data/wwwroot
```

Setting JVM memory parameter

1. Create a `/usr/local/tomcat/bin/setenv.sh` script file.

```
vi /usr/local/tomcat/bin/setenv.sh
```

2. Press I to enter the editing mode and add the following.

```
JAVA_OPTS='-Djava.security.egd=file:/dev/./urandom -server -Xms256m -Xmx496m -D
file.encoding=UTF-8'
```

3. Press Esc to exit the editing mode and enter `:wq` to save and exit.

Configuring server.xml

1. Switch to the `/usr/local/tomcat/conf/` directory.

```
cd /usr/local/tomcat/conf/
```

2. Back up the `server.xml` file.

```
mv server.xml server_default.xml
```

3. Create a new `server.xml` file.

```
vi server.xml
```

4. Press I to enter the editing mode and add the following.

```
<?xml version="1.0" encoding="UTF-8"?>
<Server port="8006" shutdown="SHUTDOWN">
<Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener"/
>
<Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListene
r"/>
<Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListene
r"/>
<Listener className="org.apache.catalina.core.AprLifecycleListener"/>
<GlobalNamingResources>
<Resource name="UserDatabase" auth="Container"
type="org.apache.catalina.UserDatabase"
description="User database that can be updated and saved"
factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
pathname="conf/tomcat-users.xml"/>
</GlobalNamingResources>
<Service name="Catalina">
<Connector port="8080"
protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443"
maxThreads="1000"
minSpareThreads="20"
acceptCount="1000"
maxHttpHeaderSize="65536"
debug="0"
disableUploadTimeout="true"
useBodyEncodingForURI="true"
enableLookups="false"
```

```
URIEncoding="UTF-8"/>
<Engine name="Catalina" defaultHost="localhost">
<Realm className="org.apache.catalina.realm.LockOutRealm">
<Realm className="org.apache.catalina.realm.UserDatabaseRealm"
resourceName="UserDatabase"/>
</Realm>
<Host name="localhost" appBase="/data/wwwroot/default" unpackWARs="true" autoDe
ploy="true">
<Context path="" docBase="/data/wwwroot/default" debug="0" reloadable="false" c
rossContext="true"/>
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
prefix="localhost_access_log." suffix=".txt" pattern="%h %l %u %t &amp;quot;%r&amp;quo
t; %s %b" />
</Host>
</Engine>
</Service>
</Server>
```

5. Press Esc to exit the editing mode and enter `:wq` to save and exit.

Starting Tomcat

Method 1

Enter the `bin` directory of the Tomcat server and run the `./startup.sh` command to start the Tomcat server.

```
cd /usr/local/tomcat/bin
./startup.sh
```

The execution result is as follows:

```
[root@emma bin]# ./startup.sh
Using CATALINA_BASE:   /usr/local/tomcat
Using CATALINA_HOME:   /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JRE_HOME:        /usr/java/jdk1.8.0_201
Using CLASSPATH:       /usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar
Tomcat started.
```

Method 2

1. Set up quick start, so that the Tomcat server can be started anywhere through `service tomcat start` .

```
wget https://github.com/lj2007331/oneinstack/raw/master/init.d/Tomcat-init
mv Tomcat-init /etc/init.d/tomcat
chmod +x /etc/init.d/tomcat
```

2. Run the following command and set the `JAVA_HOME` startup script.

```
sed -i 's@^export JAVA_HOME=.*@export JAVA_HOME=/usr/java/jdk1.8.0_201@' /etc/init.d/tomcat
```

3. Set auto-run.

```
chkconfig --add tomcat
chkconfig tomcat on
```

4. Start Tomcat.

```
# Start Tomcat
service tomcat start
# View Tomcat server status
service tomcat status
# Stop Tomcat
service tomcat stop
```

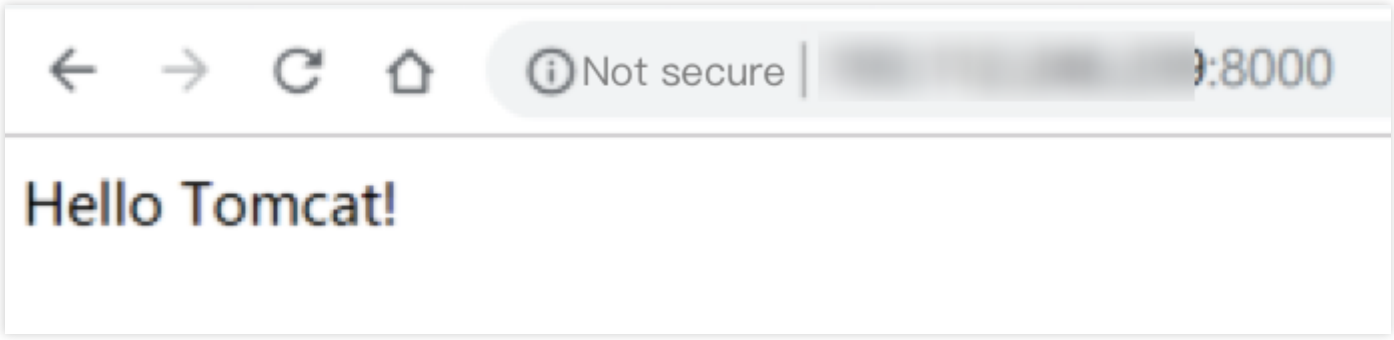
The execution result is as follows:

```
[root@emma bin]# service tomcat start
Tomcat is already running (pid: 27293)
[root@emma bin]# service tomcat status
Tomcat is running with pid: 27293
[root@emma bin]# service tomcat stop
Stoping Tomcat
Using CATALINA_BASE:   /usr/local/tomcat
Using CATALINA_HOME:   /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JRE_HOME:        /usr/java/jdk1.8.0_201
Using CLASSPATH:       /usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar
waiting for processes to exit
```

5. If the system prompts that you have no permissions, switch to the root user and modify the permissions.

```
cd /usr/local
chmod -R 777 tomcat
```

6. Enter `http://public IP:port` (where the port is the connector port set in `server.xml`) in the address bar of the browser. If the following page appears, the installation is successful.



Configuring security group

In case of access failure, check the security group. As shown in the above example, the connector port is 8080 in `server.xml` , so you need to open TCP:8080 to the internet in the security group bound to the corresponding CVM instance.