

04_Deploy_Best Practice_Public Cloud-Copy

04_Deploy_Best Practice_Public Cloud-Copy

Issue 01
Date 2025-08-26



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.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: <https://www.huaweicloud.com/intl/en-us/>

Contents

1 CodeArts Deploy Best Practices.....	1
2 Deploying an Application on an Intranet Host Using a Proxy Host.....	2
3 Using Nginx for Gray Release.....	8
4 Using Kubernetes Nginx-Ingress for Gray Release.....	38
5 Security Best Practices.....	44
6 CodeArts Deploy Permission Management Best Practices.....	46
7 HE2E DevOps Practice: Deploying an Application.....	55
7.1 Overview.....	55
7.2 Deploying an Application on CCE.....	55
7.3 Deploying an Application on ECS.....	59
7.4 Releasing Resources.....	64

1 CodeArts Deploy Best Practices

Table 1-1 Best practices

Practice		Description
Internal network deployment	Deploying an Application on an Intranet Host Using a Proxy Host	This practice provides a complete operation guide for deploying an application on a host or server on the internal network through a proxy host.
Gray deployment	Using Nginx for Gray Release	This practice implements blue-green and gray release of applications based on the Nginx load balancing mechanism.
	Using Kubernetes Nginx-Ingress for Gray Release	This practice implements gray release based on native Kubernetes features.
HE2E DevOps practice	HE2E DevOps Practice: Deploying an Application	This practice uses the DevOps process example project to describe how to deploy applications on CCE and ECS.
Security Best Practices	Security Best Practices	This practice provides actionable guidance for enhancing the overall security of CodeArts Deploy.
Permission Management Best Practices	CodeArts Deploy Permission Management Best Practices	This practice provides the configuration process of CodeArts Deploy permission management, helping you understand the permission architecture of CodeArts Deploy.

2 Deploying an Application on an Intranet Host Using a Proxy Host

Application Scenario

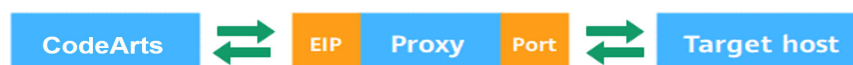
Deploy applications on the intranet through proxy hosts to effectively control intranet and extranet communication, enhance data security and network performance, and meet compliance requirements. This policy is widely used in various key scenarios, such as resource access control, secure communication between data centers, content cache acceleration, environment isolation, security audit, and sensitive data processing.

Solution Architecture

The Internet forward proxy function of Squid is used to specify the IP address and port of the target host on the proxy, enabling the target host to access the public network.

For more information about Squid, go to [Squid official website](#). The following procedure uses a Linux host as an example.

Figure 2-1 Principles of specifying a target host on a proxy



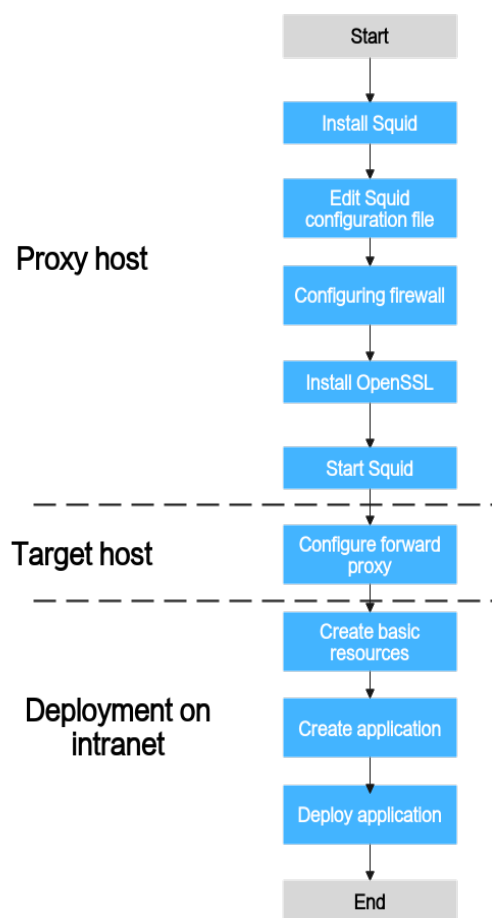
Prerequisites

- A host (**Proxy-B**) bound to a public IP address is available. If no proxy host is available, see [Preparations](#).
- A host (**Host-A**) not bound to a public IP address is available.
- **Proxy-B** and **Host-A** can access each other through the intranet.

Process

This section describes how to deploy an application on an intranet host or server using a proxy host.

Figure 2-2 Process flowchart

**Step 1** Install Squid.

Access the command line tool of **Proxy-B** and run the following command:

```
yum install squid -y
```

If **Complete** is displayed, run the following command:

```
yum install iptables-services
```

Enter **Y**. If **Complete** is displayed, the installation is complete.

Step 2 Edit the Squid configuration file.

1. Access the command line tool of **Proxy-B** and run the following command:
`vim /etc/squid/squid.conf`

```
# Example rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
acl localnet src 10.0.0.0/24          # RFC1918 possible internal network
acl localnet src 172.16.0.0/16       # RFC1918 possible internal network
acl localnet src 192.168.0.0/16      # RFC1918 possible internal network
acl localnet src 10.0.0.0/8          # RFC 4193 local private network range
acl localnet src ::0/0               # RFC 4291 link-local (directly plugged) machines
acl SSL_ports port 443
acl Safe_ports port 80               # http
```

2. Add the following command to the position marked in the red box in the preceding figure:

```
acl local src Internal IP address of the host/24
```

3. Press **Esc** and enter **:wq** to save the file and exit.

Step 3 Configure the firewall of **Proxy-B**.

Access the command line tool of **Proxy-B** and run the following commands in sequence:

```
systemctl stop firewalld.service
systemctl disable firewalld.service
yum install iptables-services iptables-devel -y
systemctl enable iptables.service
systemctl start iptables.service
iptables -I INPUT 1 -s Internal IP address of the host/24 -p tcp --dport 3128 -j ACCEPT
iptables -I INPUT 2 -p tcp --dport 3128 -j DROP
```

The IP address in the last but one line must be set to the internal IP address segment or IP address of **Host-A**. **3128** is the proxy port of Squid.

Step 4 Install OpenSSL.

Access the command line tool of **Proxy-B** and run the following command:

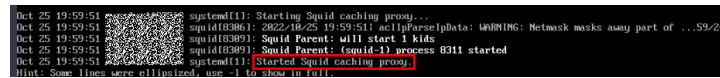
```
yum install openssl
```

If **Complete** is displayed, the installation is complete.

Step 5 Start Squid.

Access the command line tool of **Proxy-B** and run the following command:

```
systemctl start squid //Start Squid.
systemctl status squid //Check the status of Squid.
```



If Squid fails to be started, see [What Should I Do If Squid Fails to Be Started?](#)

Step 6 Configure the forward proxy.

Access the command line tool of **Host-A** and run the following command:

```
echo "export http_proxy=http://Internal IP address of the proxy host:3128" >>/etc/profile
echo "export https_proxy=http://Internal IP address of the proxy host:3128" >>/etc/profile
echo "export http_proxy=http://Internal IP address of the proxy host:3128" >>~/.bashrc
echo "export https_proxy=http://Internal IP address of the proxy host:3128" >>~/.bashrc
echo "export http_proxy=http://Internal IP address of the proxy host:3128" >>~/.bash_profile
echo "export https_proxy=http://Internal IP address of the proxy host:3128" >>~/.bash_profile
source /etc/profile
source ~/.bashrc
source ~/.bash_profile
```

Step 7 Create basic resources.

1. Click **Homepage** to view all created projects, and then go to the target project.
2. Choose **Settings > General > Basic Resources** to access the **Host Clusters** page.
3. Click **Create Host Cluster**, enter the following information, and click **Save**.

Parameter	Mandatory	Description
Cluster Name	Yes	Enter a custom name.
OS	Yes	Select Linux based on the OS of the host to be added.
Host Connection Mode	Yes	Select Proxy .
Execution Agent Pool	Yes	An agent pool is a collection of physical environments where commands are executed during software package deployment. In this scenario, select official .
Description	No	Enter a description.

- Click **Add Host**, select **Adding IP** for **Add hosts by**, enter the following information, and click **OK**. The proxy host is created.

Table 2-1 Parameters of a Linux proxy host


Parameter	Mandatory	Description
Host Name	Yes	Enter a custom name, for example, Proxy-B .
IP	Yes	Enter the public IP address bound to Proxy-B .
OS	Yes	Keep the default value because it is the OS of your host cluster.
Authorization	Yes	In this scenario, the Password is used for authentication. Enter the username and password of Proxy-B .
SSH Port	Yes	Port 22 is recommended.

- Click **Add Host**, select **Adding IP** for **Add hosts by**, enter the following information, and click **OK**. The target host is created.

Table 2-2 Parameters of a Linux target host

Parameter	Mandatory	Description
Host Name	Yes	Enter a custom name, for example, Host-A .

Parameter	Mandatory	Description
Proxy Host	Yes	Select Proxy-B as the network proxy for the target host that cannot connect to the public network.
IP	Yes	Enter the private IP address of Host-A .
OS	Yes	Keep the default value because it is the OS of your host cluster.
Authorization	Yes	In this scenario, the Password is used for authentication. Enter the username and password of Host-A .
SSH Port	Yes	Port 22 is recommended.


- Click  in the **Operation** column of a host to start the connectivity verification for the host. For details about connectivity verification, see [Host Management](#).

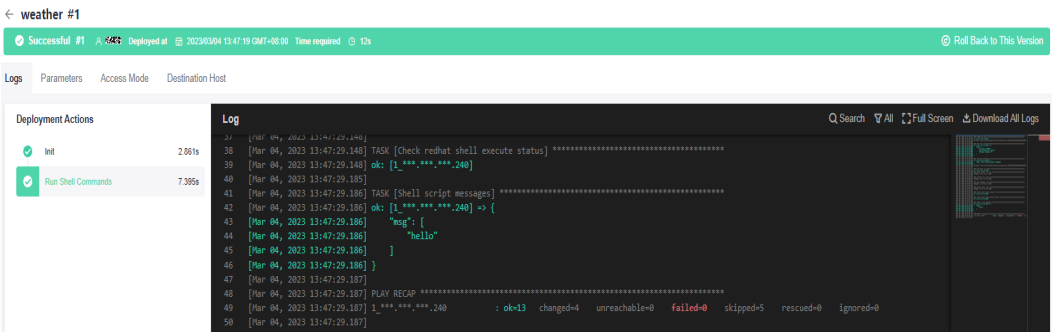
Step 8

- Log in to the CodeArts homepage and click the target project name to access the project.
- Choose **CICD > Deploy**.
- Click **Create Application**. On the **Basic Information** page, specify the **Name**, **Description**, and **Execution Resource Pool** as required.
- After editing the basic application information, click **Next**. On the **Select Template** page, select **Blank Template** and click **OK**.
- On the **Deployment Actions** tab page, find the action list on the right, click **Add** to add an action to the orchestration area on the left.
- On the **Environment Management** page, click **Create Environment**, enter the basic information, and click **Save**.
- Click **Import Host**. The system automatically filters all clusters that meet the requirements of the current environment. In the displayed dialog box, select the target host cluster and import **Proxy-B** and **Host-A** to the environment.

Step 9

Deploy the application. For details, see [Deploying an Application and Viewing the Result](#).

- Select the target application in the application list and click .
- After the deployment is complete, click the application name and click the target deployment record. The application status bar changes to green and the message **Successful** is displayed, indicating that the application is successfully deployed.



For more deployment FAQs, see [Application Deployment](#).

----End

3 Using Nginx for Gray Release

Application Scenario

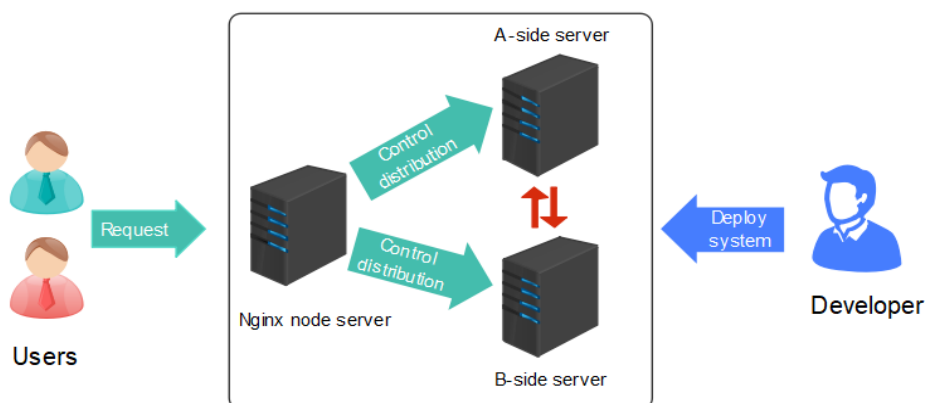
When you upgrade to a new system, services may be stopped or gray verification may fail. In this practice, you can use the Nginx load balancing mechanism for smooth system upgrade without affecting service running.

Solution Architecture

When upgrading the system using blue-green deployment, perform the following operations:

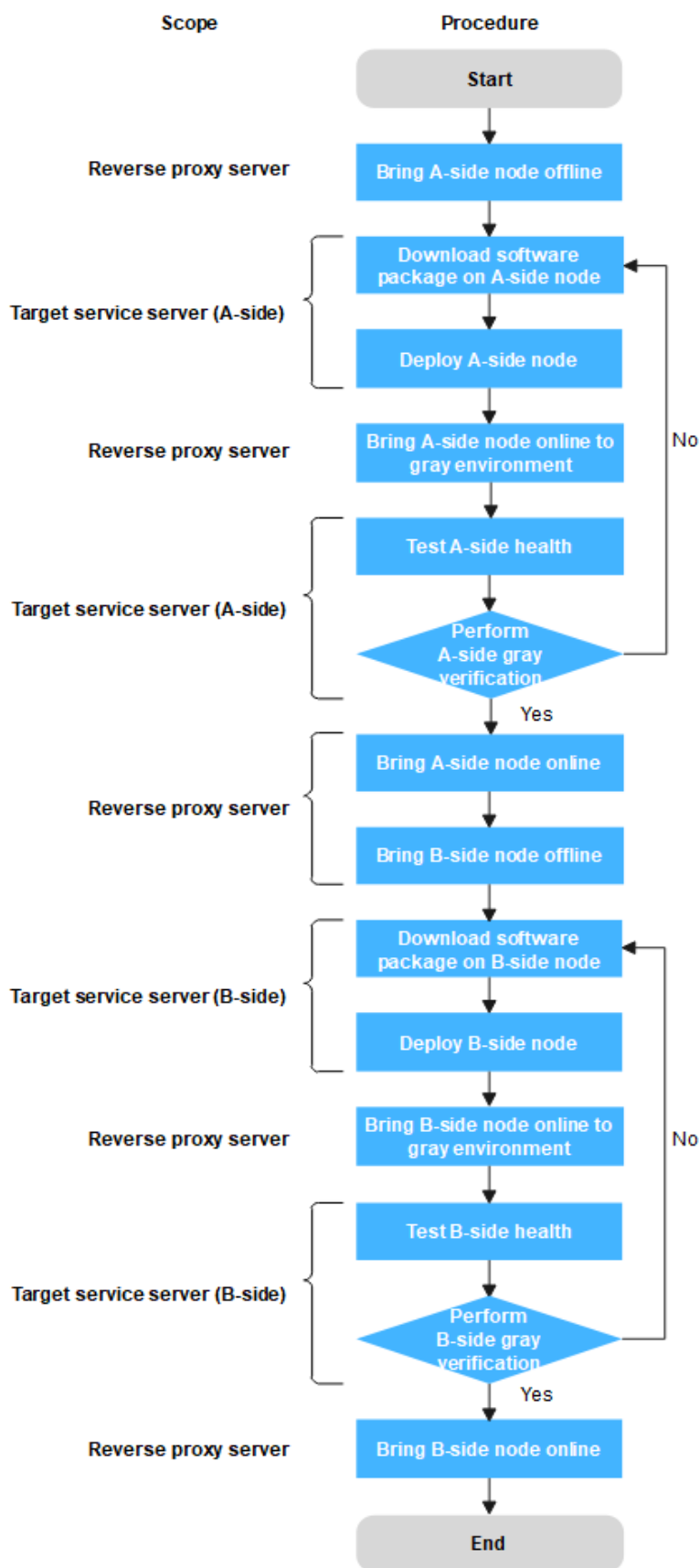
1. Bring server A (blue environment, carrying the current workloads) offline and distribute all access traffic to server B. Upgrade server A.
2. After server A is upgraded, set it as the gray test environment and let testers perform gray verification on it.
3. After the gray verification is complete and the functions become normal, release server A (green environment) and move all traffic from B to it. Now, the blue-green deployment is complete.
4. If an emergency occurs on server A during service running, perform a blue-green switchover to quickly restore services.

Figure 3-1 Gray release scheme



If you use canary release (grayscale release), upgrade server B by referring to operations described in blue-green deployment. Complete the gray test on it and release it to complete the gray release.

Figure 3-2 Procedure



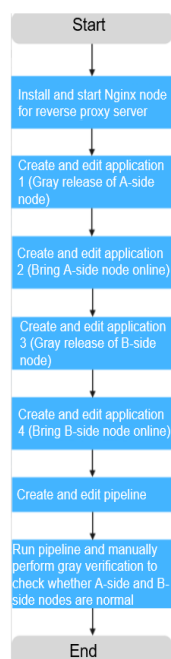
Prerequisites

- A project is available. If there is no project, [Creating a Scrum Project and a Work Item](#) first.
- You have the permissions to create applications. For details, see [Configuring Permissions for Different Roles](#).
- Target service servers **A_test** and **B_test** are available and application services are running on the servers. This practice uses Linux as an example.
- A reverse proxy server **Gray_release** is available. This practice uses Linux as an example.
- A gray verification host is available. This host represents a gray tester.
- Ensure that servers can communicate with each other. To ensure it, you can add all servers to the same virtual private cloud (VPC).

Process

Based on the Nginx load balancing mechanism, this practice implements blue-green release and gray release in host deployment scenarios. For more information about Nginx, visit [Nginx official website](#).

Figure 3-3 Process flowchart



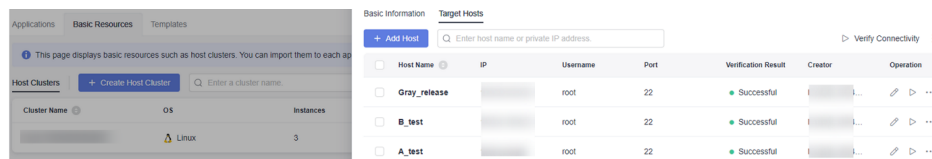
Step 1: (Optional) Install and Start the Nginx Node on the Reverse Proxy Server

If the Nginx node has been installed and started on your reverse proxy server, skip this step.

Step 1 Create basic resources.

1. Go to the CodeArts homepage and click the target project name to access the project.

2. Choose **Settings > General > Basic Resources** to access the **Host Clusters** page.
Alternatively, choose **CICD > Deploy > Basic Resources** to access the **Host Clusters** page.
3. Click **Create Host Cluster**, enter basic information such as the **Cluster Name**, **OS**, **Proxy**, **Execution Resource Pool**, and **Description**, and click **Save**.
4. Create three hosts (**A_test**, **B_test**, and **Gray_release**) and verify their connectivity. Specifically, click **Add Host**, select **Adding IP**, enter the host name, **IP**, **Username**, **Password** or **Key**, and **SSH Port**, and click **OK**. For details about connectivity verification, see [Host Management](#).



Step 2 Create an application.

1. Choose **CICD > Deploy**.
2. Click **Create Application**. On the **Basic Information** page, specify the **Name**, **Description**, and **Execution Resource Pool** as required.
3. After editing the basic application information, click **Next**. The deployment template selection page is displayed.
4. Select **Blank Template** and click **OK**. The **Deployment Actions** tab page is displayed.

Step 3 Edit the application.


1. Switch to the **Environment Management** tab page. Create an environment and edit it.
 - Click **Create Environment**, enter the environment name, for example, **Reverse_proxy_server_group**, select the OS corresponding to each server, and enter the description information.
 - Click **Save**. The environment is created.
 - Click **Import Host**. The system automatically filters all clusters that meet the requirements of the current environment. In the displayed dialog box, select the target host cluster and click  in the **Operation** column of the target host to import the host to the environment.
2. Switch to the **Deployment Actions** tab page. Add the following actions and edit them.
 - Add the **Install Nginx** action and modify the parameters in the following table (Linux is used as an example).

Table 3-1 Parameter description

Parameter	Mandatory	Description
Action Name	Yes	Enter a custom action name. You can retain the default value. Enter 1 to 128 characters. Do not start or end with a space. Use letters, digits, spaces, and these special characters: -_.,:./()
Environment	Yes	Select Reverse_proxy_server_group .
Nginx Version	Yes	Select a target version. Example: nginx-1.14.2 .
Installation Path	Yes	Enter the installation path of the Nginx service in the target environment. Example: /usr/local/nginx .

- Add the **Start/Stop Nginx** action and modify the parameters in the following table (Linux is used as an example).

Table 3-2 Parameter description

Parameter	Mandatory	Description
Action Name	Yes	Enter a custom action name. You can retain the default value. Enter 1 to 128 characters. Do not start or end with a space. Use letters, digits, spaces, and these special characters: -_.,:./()
Environment	Yes	Select Reverse_proxy_server_group .
Operation	Yes	Select Start Nginx .
Nginx Installation Path	Yes	Enter the installation path of the Nginx service in the target environment. Example: /usr/local/nginx .

3. Click **Save & Deploy** to deploy the application.

Step 4 Deploy the application.

If the application status bar turns green and displays **Successful**, the application is deployed successfully.

If the application status bar turns red and displays **Failed**, the application fails to be deployed. In this case, click **View Solution**.

For more deployment FAQs, see [Application Deployment](#).

----End

Step 2: Create and Edit Application 1 (Gray Release of A-Side Node)

Step 1 Create an application.

1. Choose **CICD > Deploy**.
2. Click **Create Application**. On the **Basic Information** page, specify the **Name**, **Description**, and **Execution Resource Pool** as required.
3. After editing the basic application information, click **Next**. The deployment template selection page is displayed.
4. Select the **Deploy a General Application** template and click **OK**.

Step 2 Edit the application.


1. Switch to the **Environment Management** tab page. Create an environment and edit it.
 - Click **Create Environment**, enter the environment name, for example, **Reverse_proxy_server_group**, select the OS corresponding to each server, and enter the description information.
 - Click **Save**. The environment is created.
 - Click **Import Host**. The system automatically filters all clusters that meet the requirements of the current environment. In the displayed dialog box, select the target host cluster and click  in the **Operation** column of the target host to import the host to the environment.
 - Repeat the preceding steps to create **Target service server group_A-side node** and add the **A_test** server.
2. Switch to the **Parameters** tab page. Add the following parameters.

Table 3-3 Parameter description

Parameter	Mandatory	Description
Name	Yes	Parameter name. You can customize it. The name of a custom parameter cannot be the same as that of a predefined parameter. Enter 1 to 128 characters, including letters, digits, and underscores (_). Examples: app_name and service_port .

Parameter	Mandatory	Description
Type	Yes	Parameter types, including String , Enumeration , and Environment . Select String in this example.
Default Value	No	Enter or select a parameter value. Examples: test and 3000 .
Private Parameter	No	If a parameter is private, the system encrypts the input for storage and only decrypts the parameter when you use it. If you enable Private Parameter , Runtime Settings cannot be enabled. It is disabled by default.
Runtime Settings	No	If it is enabled, the parameter value can be changed during application deployment and the value will be reported to the application. It is disabled by default.
Description	No	Parameter description.

3. Switch to the **Deployment Actions** tab page. Add the following actions and edit them.
 - Add the **Start/Stop Nginx** action and modify the parameters in the following table (Linux is used as an example).

Table 3-4 Parameter description

Parameter	Mandatory	Description
Action Name	Yes	Enter a customized action name displayed in the deployment actions. Enter 1 to 128 characters. Do not start or end with a space. Use letters, digits, spaces, and these special characters: <code>-_;/./()</code> Enter a name such as Bring_A-side_node_offline .
Environment	Yes	Select a target environment. Example: Reverse_proxy_server_group .
Operation	Yes	Specify an operation type. Example: Reload configuration file .

Parameter	Mandatory	Description
Nginx Installation Path	Yes	Enter the installation path of the Nginx service in the target environment. Example: /usr/local/nginx .
Modify configuration file before execution	No	Select this option for this example.
Nginx Configuration File Path	Yes	Enter the path of the Nginx configuration file on the target host. Example: /usr/local/nginx/conf/nginx.conf .
Configuration File Backup Path	No	Enter the target path for backing up the original Nginx configuration file on the target host. Example: /usr/local/nginx/conf/nginx_bak.conf .
Configuration File Content	Yes	Enter content of the new configuration file. See Example code to bring A-side node offline in the appendix.

- Edit the **Download Software Package** action and modify the parameters in the following table (Linux is used as an example).

Table 3-5 Parameter description

Parameter	Mandatory	Description
Action Name	Yes	Enter a customized action name displayed in the deployment actions. Enter 1 to 128 characters. Do not start or end with a space. Use letters, digits, spaces, and these special characters: <code>-_.,:./()</code> Example: Download_software_package_on_A-side_node .
Source	Yes	Select a source. Example: Artifact .

Parameter	Mandatory	Description
Environment	Yes	Select a target environment. Example: Target service server group_A-side node .
Software Package	Yes	Select a software package to be deployed in CodeArts Artifact.
Download Path	Yes	Enter the path for downloading the software package to the target host. Example: /usr/local/ .

- Edit the **Run Shell Commands** action and modify the parameters in the following table (Linux is used as an example).

Table 3-6 Parameter description

Parameter	Mandatory	Description
Action Name	Yes	Enter a customized action name displayed in the deployment actions. Enter 1 to 128 characters. Do not start or end with a space. Use letters, digits, spaces, and these special characters: -_.,:./() Example: Deploy A-side node .
Environment	Yes	Select a target environment. Example: Target service server group_A-side node .
Shell Commands	Yes	Enter the commands to be executed. Example: See Deployment node in the appendix.

- Add the **Start/Stop Nginx** action and modify the parameters in the following table (Linux is used as an example).

Table 3-7 Parameter description

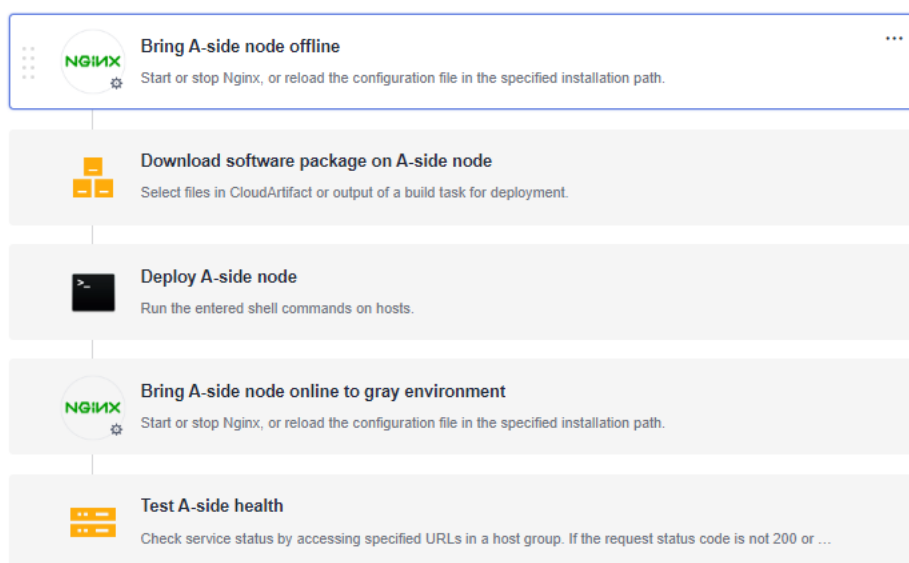
Parameter	Mandatory	Description
Action Name	Yes	Enter a customized action name displayed in the deployment actions. Enter 1 to 128 characters. Do not start or end with a space. Use letters, digits, spaces, and these special characters: -_;;./() Example: Bring A-side node online to gray environment.
Environment	Yes	Select a target environment. Example: Reverse_proxy_server_group.
Operation	Yes	Specify an operation type. Example: Reload configuration file.
Nginx Installation Path	Yes	Enter the installation path of the Nginx service in the target environment. Example: /usr/local/nginx.
Modify configuration file before execution	No	Select this option for this example.
Nginx Configuration File Path	Yes	Enter the path of the Nginx configuration file on the target host. Example: /usr/local/nginx/conf/nginx.conf.
Configuration File Backup Path	No	Enter the target path for backing up the original Nginx configuration file on the target host. Example: /usr/local/nginx/conf/nginx_bak.conf.
Configuration File Content	Yes	Enter content of the new configuration file. See Example code to bring A-side node online to the gray environment in the appendix.

- Edit the **Health Test via URLs** action and modify the parameters in the following figure (Linux is used as an example).

Table 3-8 Parameter description

Parameter	Mandatory	Description
Action Name	Yes	Enter a customized action name displayed in the deployment actions. Enter 1 to 128 characters. Do not start or end with a space. Use letters, digits, spaces, and these special characters: -_;/() Enter a name such as Test_A-side_health .
Environment	Yes	Select a target environment. Example: Target service server group_A-side node .
Retries	Yes	If a service does not start up when the health test reaches the maximum retries, the service fails this test. Example: 1
Interval (s)	Yes	Interval between two retries, in seconds. Example: 60
Test Path	Yes	Used for the health test via URLs. You can add multiple URLs. Example: http://127.0.0.1:3000 (IP address and port number of the application service)

Step 3 Click **Save**. The application is created.




----End

Step 3: Create and Edit Application 2 (Bring_A-side_node_online)

Step 1 Create an application.

1. Click **Create Application**. On the **Basic Information** page, specify the **Name**, **Description**, and **Execution Resource Pool** as required.
2. After editing the basic application information, click **Next**. The deployment template selection page is displayed.
3. Select **Blank Template** and click **OK**.

Step 2 Edit the application.

1. Switch to the **Environment Management** tab page. Create an environment and edit it.
 - Click **Create Environment**, enter the environment name, for example, **Reverse_proxy_server_group**, select the OS corresponding to each server, and enter the description information.
 - Click **Save**. The environment is created.
 - Click **Import Host**. The system automatically filters all clusters that meet the requirements of the current environment. In the displayed dialog box, select the target host cluster and click  in the **Operation** column of the target host to import the host to the environment.

2. Switch to the **Deployment Actions** tab page. Add the following actions and edit them.

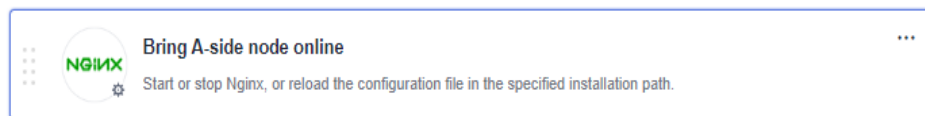
Add the **Start/Stop Nginx** action and modify the parameters in the following table (Linux is used as an example).

Table 3-9 Parameter description

Parameter	Mandatory	Description
Action Name	Yes	Enter a customized action name displayed in the deployment actions. Enter 1 to 128 characters. Do not start or end with a space. Use letters, digits, spaces, and these special characters: -_.,:./() Example: Bring_A-side_node_online .
Environment	Yes	Select a target environment. Example: Reverse_proxy_server_group .
Operation	Yes	Specify an operation type. Example: Reload configuration file .
Nginx Installation Path	Yes	Enter the installation path of the Nginx service in the target environment. Example: /usr/local/nginx .

Parameter	Mandatory	Description
Modify configuration file before execution	No	Select this option for this example.
Nginx Configuration File Path	Yes	Enter the path of the Nginx configuration file on the target host. Example: <code>/usr/local/nginx/conf/nginx.conf</code> .
Configuration File Backup Path	No	Enter the target path for backing up the original Nginx configuration file on the target host. Example: <code>/usr/local/nginx/conf/nginx_bak.conf</code> .
Configuration File Content	Yes	Enter content of the new configuration file. See Example code to bring a node online in the appendix.


Step 3 Click **Save**. The application is created.

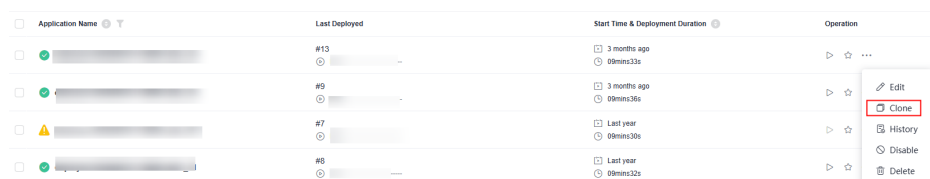


-----End

Step 4: Create and Edit Application 3 (Gray_release_of_B-side_node)

Step 1 Clone an application.

Click  and choose **Clone**.



Step 2 Edit the application.

- Switch to the **Environment Management** tab page. Create an environment and edit it.
 - Click **Create Environment**, enter the environment name, for example, **Reverse_proxy_server_group**, select the OS corresponding to each server, and enter the description information.
 - Click **Save**. The environment is created.
 - Click **Import Host**. The system automatically filters all clusters that meet the requirements of the current environment. In the displayed dialog box,


- select the target host cluster and click  in the **Operation** column of the target host to import the host to the environment.
- Repeat the preceding steps to create **Target service server group_B-side node** and add the **B_test** server.
2. Switch to the **Deployment Actions** tab page. Add the following actions and edit them.
- Edit the **Bring A-side node offline** action and modify the parameters as follows (Linux is used as an example).

Table 3-10 Parameter description

Parameter	Mandatory	Description
Action Name	Yes	Enter a customized action name displayed in the deployment actions. Enter 1 to 128 characters. Do not start or end with a space. Use letters, digits, spaces, and these special characters: -_.,:./() Example: Bring_B-side_node_offline .
Environment	Yes	Select a target environment. Example: Reverse_proxy_server_group .
Operation	Yes	Specify an operation type. Example: Reload configuration file .
Nginx Installation Path	Yes	Enter the installation path of the Nginx service in the target environment. Example: /usr/local/nginx .
Modify configuration file before execution	No	Select this option for this example.
Nginx Configuration File Path	Yes	Enter the path of the Nginx configuration file on the target host. Example: /usr/local/nginx/conf/nginx.conf .
Configuration File Backup Path	No	Enter the target path for backing up the original Nginx configuration file on the target host. Example: /usr/local/nginx/conf/nginx_bak.conf .

Parameter	Mandatory	Description
Configuration File Content	Yes	Enter content of the new configuration file. See Example code to bring B-side node offline in the appendix.

- Edit the **Download software package on A-side node** action and change the parameter values to those listed in the following table (Linux is used as an example).

Table 3-11 Parameter description

Parameter	Mandatory	Description
Action Name	Yes	Enter a customized action name displayed in the deployment actions. Enter 1 to 128 characters. Do not start or end with a space. Use letters, digits, spaces, and these special characters: -_.,:./() Example: Download software package on B-side node.
Source	Yes	Select a source. Example: Artifact.
Environment	Yes	Select a target environment. Example: B_group.
Software Package	Yes	Select a software package to be deployed in CodeArts Artifact.
Download Path	Yes	Enter the path for downloading the software package to the target host. Example: /usr/local/.

- Edit the **Deploy A-side node** action and modify the parameters as follows (Linux is used as an example).

Table 3-12 Parameter description

Parameter	Mandatory	Description
Action Name	Yes	Enter a customized action name displayed in the deployment actions. Enter 1 to 128 characters. Do not start or end with a space. Use letters, digits, spaces, and these special characters: -_;;./() Example: Deploy_B-side_node .
Environment	Yes	Select a target environment. Example: B_group .
Shell Commands	Yes	Enter the commands to be executed. Example: See Deployment node in the appendix.

- Edit the **Bring A-side node online to gray environment** action and modify the parameters as follows (Linux is used as an example).

Table 3-13 Parameter description

Parameter	Mandatory	Description
Action Name	Yes	Enter a customized action name displayed in the deployment actions. Enter 1 to 128 characters. Do not start or end with a space. Use letters, digits, spaces, and these special characters: -_;;./() Example: Bring_B-side_node_online_to_gray_environment .
Environment	Yes	Select a target environment. Example: Reverse_proxy_server_group .
Operation	Yes	Specify an operation type. Example: Reload configuration file .
Nginx Installation Path	Yes	Enter the installation path of the Nginx service in the target environment. Example: /usr/local/nginx .

Parameter	Mandatory	Description
Modify configuration file before execution	No	Select this option for this example.
Nginx Configuration File Path	Yes	Enter the path of the Nginx configuration file on the target host. Example: /usr/local/nginx/conf/nginx.conf .
Configuration File Backup Path	No	Enter the target path for backing up the original Nginx configuration file on the target host. Example: /usr/local/nginx/conf/nginx_bak.conf .
Configuration File Content	Yes	Enter content of the new configuration file. See Example code to bring B-side node online to the gray environment in the appendix.

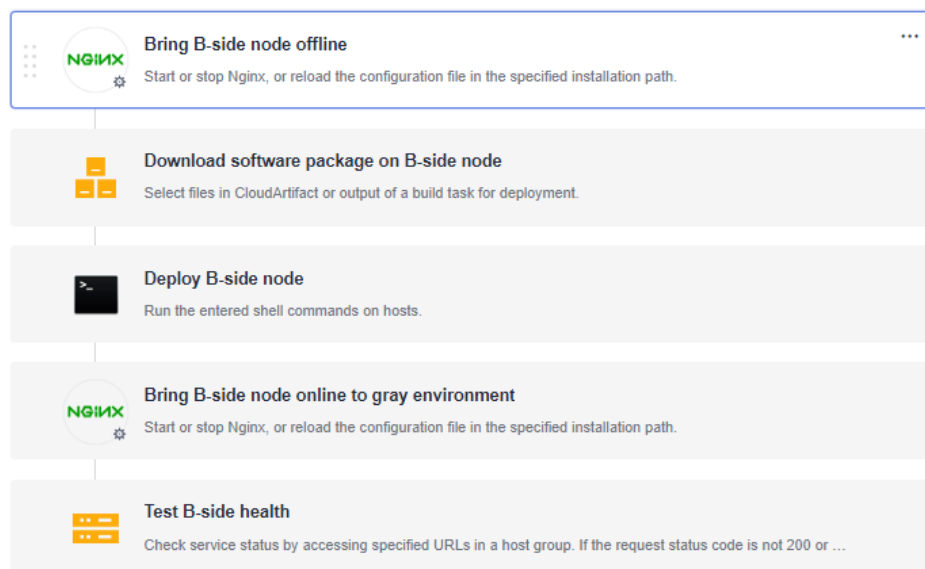
- Edit the **Test A-side health** action and modify the parameters as follows (Linux is used as an example).

Table 3-14 Parameter description

Parameter	Mandatory	Description
Action Name	Yes	Enter a customized action name displayed in the deployment actions. Enter 1 to 128 characters. Do not start or end with a space. Use letters, digits, spaces, and these special characters: -_!;:./() Enter a name such as Test_B-side_health .
Environment	Yes	Select a target environment. Example: B_group .
Retries	Yes	If a service does not start up when the health test reaches the maximum retries, the service fails this test. Example: 1

Parameter	Mandatory	Description
Interval (s)	Yes	Interval between two retries, in seconds. Example: 60
Test Path	Yes	Used for the health test via URLs. You can add multiple URLs. Example: http://127.0.0.1:3000 (IP address and port number of the application service)

Step 3 Click **Save**. The application is created.



----End


Step 5: Create and Editing Application 4 (Bring_B-side_node_online)

Step 1 Clone an application.

Click and choose **Clone**.

Step 2 Edit the application.

1. Switch to the **Environment Management** tab page. Create an environment and edit it.
 - Click **Create Environment**, enter the environment name, for example, **Reverse_proxy_server_group**, select the OS corresponding to each server, and enter the description information.
 - Click **Save**. The environment is created.
 - Click **Import Host**. The system automatically filters all clusters that meet the requirements of the current environment. In the displayed dialog box,

select the target host cluster and click  in the **Operation** column of the target host to import the host to the environment.

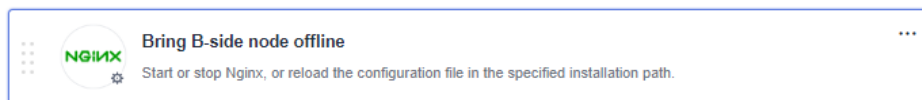
2. Switch to the **Deployment Actions** tab page. Add the following actions and edit them.

Edit the **Bring A-side node online** action and modify the parameters as follows (Linux is used as an example):

Table 3-15 Parameter description

Parameter	Mandatory	Description
Action Name	Yes	Enter a customized action name displayed in the deployment actions. Enter 1 to 128 characters. Do not start or end with a space. Use letters, digits, spaces, and these special characters: -_.,:./() Example: Bring B-side_node_online .
Environment	Yes	Select a target environment. Example: Reverse_proxy_server_group .
Operation	Yes	Specify an operation type. Example: Reload configuration file .
Nginx Installation Path	Yes	Enter the installation path of the Nginx service in the target environment. Example: /usr/local/nginx .
Modify configuration file before execution	No	Select this option for this example.
Nginx Configuration File Path	Yes	Enter the path of the Nginx configuration file on the target host. Example: /usr/local/nginx/conf/nginx.conf .
Configuration File Backup Path	No	Enter the target path for backing up the original Nginx configuration file on the target host. Example: /usr/local/nginx/conf/nginx_bak.conf .
Configuration File Content	Yes	Enter content of the new configuration file. See Example code to bring a node online in the appendix.

Step 3 Click **Save**. The application is created.

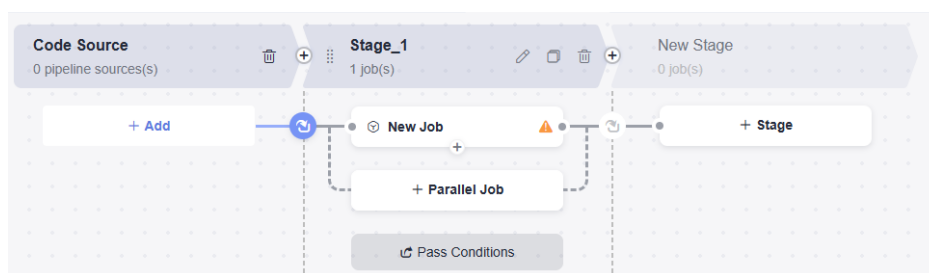


----End

Step 6: Create and Edit a Pipeline

Step 1 Create a pipeline.

- Choose **CICD > Pipeline**.
- Click **Create Pipeline**, select a **Project**, enter a **Name**, set **Pipeline Source** to **None**, and click **Next**.
- Select **Blank Template** and click **OK**.



Step 2 Edit job 1 (**Gray release of A-side node**) in the pipeline stage.


- Click . In the displayed dialog box, set the parameters as follows and click **Confirm**.

Table 3-16 Parameter description

Parameter	Ma nd ato ry	Description
Stage Name	Yes	Example: Gray_release_of A-side_node .
Always Run	Yes	Select No .


- Click . In the displayed dialog box, set **Entry Type** to **Automatic** and click **OK**.
- Click **New Job**, click the **Deploy** tab, select **Deploy**, and click **Add**. In the displayed dialog box, set the parameters as follows and click **OK**.

Table 3-17 Parameter description

Parameter	Mandatory	Description
Name	Yes	Example: Gray_release_of A-side_node .
Select Task	Yes	Select Gray_release_of A-side_node .
Build Task	No	Leave it blank for this example.

Step 3 Create and edit job 2 (**Bring_A-side_node_online**) in the pipeline stage.



- Click  and . In the displayed dialog box, set the parameters as follows and click **Confirm**.

Table 3-18 Parameter description

Parameter	Mandatory	Description
Name	Yes	Example: Bring_A-side_node_online .
Always Run	Yes	Select No .


- Click . In the displayed dialog box, set **Entry Type** to **Automatic** and click **OK**.
- Click **New Job**. In the window that is displayed, click the **Normal** tab, select **Manual Review** and click **Add**, set the parameters as follows, and click **OK**.

Table 3-19 Parameter description

Parameter	Mandatory	Description
Name	Yes	Example: Gray_release_of A-side_node .
Reviewer	Yes	Select the service verification personnel.
Review Mode	Yes	Select Review by all .
Timeout Processing	Yes	Select Review failed and pipeline terminated .

Parameter	Ma nd ato ry	Description
Review Duration	Yes	Example: 4 hours.
Description	No	Enter the review description.

- Click , click the **Deploy** tab, select **Deploy**, and click **Add**. In the displayed dialog box, set the parameters as follows and click **OK**.

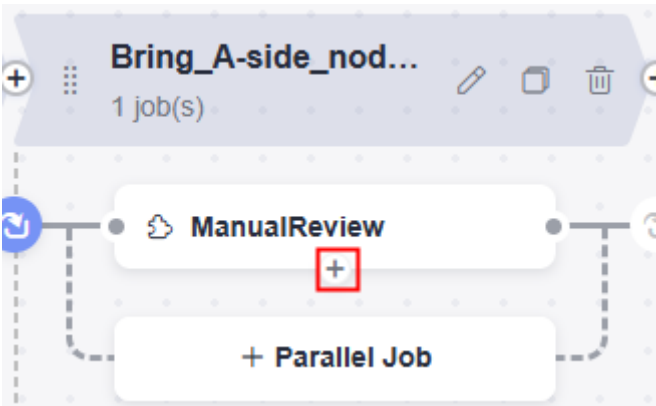


Table 3-20 Parameter description

Parameter	Ma nd ato ry	Description
Name	Yes	Example: Bring_A-side_node_online .
Select Task	Yes	Select Bring_A-side_node_online .
Build Task	No	Leave it blank for this example.

Step 4 Edit job 3 (**Gray_release_of_B-side_node**) in the pipeline stage.



- Click  and . In the displayed dialog box, set the parameters as follows and click **Confirm**.

Table 3-21 Parameter description

Parameter	Ma nd ato ry	Description
Name	Yes	Example: Gray_release_of_B-side_node .

Parameter	Mandatory	Description
Always Run	Yes	Select No .


- Click . In the displayed dialog box, set **Entry Type** to **Automatic** and click **OK**.
- Click **New Job**, click the **Deploy** tab, select **Deploy**, and click **Add**. In the displayed dialog box, set the parameters as follows and click **OK**.

Table 3-22 Parameter description

Parameter	Mandatory	Description
Name	Yes	Example: Gray_release_of_B-side_node .
Select Task	Yes	Select Gray_release_of_B-side_node .
Build Task	No	Leave it blank for this example.

Step 5 Create and edit job 4 (**Bring_B-side_node_online**) in the pipeline stage.



- Click  and . In the displayed dialog box, set the parameters as follows and click **Confirm**.

Table 3-23 Parameter description

Parameter	Mandatory	Description
Name	Yes	Example: Bring_B-side_node_online .
Always Run	Yes	Select No .


- Click . In the displayed dialog box, set **Entry Type** to **Automatic** and click **OK**.
- Click **New Job**. In the window that is displayed, click the **Normal** tab, select **Manual Review** and click **Add**, set the parameters as follows, and click **OK**.

Table 3-24 Parameter description

Parameter	Ma nd ato ry	Description
Name	Yes	Example: Gray_verification_of_B-side_node.
Reviewer	Yes	Select the service verification personnel.
Review Mode	Yes	Select Review by all.
Timeout Processing	Yes	Select Review failed and pipeline terminated.
Review Duration	Yes	Example: 4 hours.
Description	No	Enter the review description.

- Click  , click the **Deploy** tab, select **Deploy**, and click **Add**. In the displayed dialog box, set the parameters as follows and click **OK**.

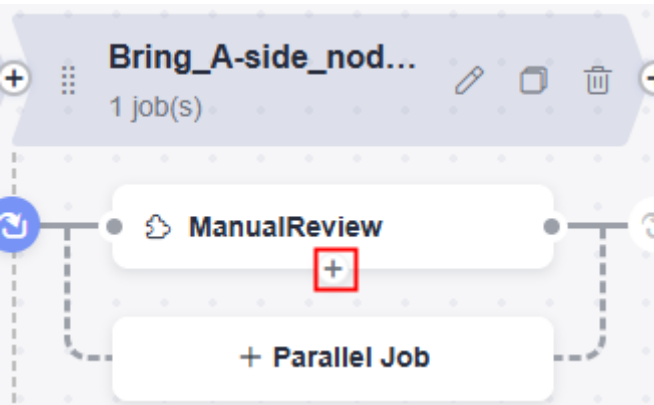
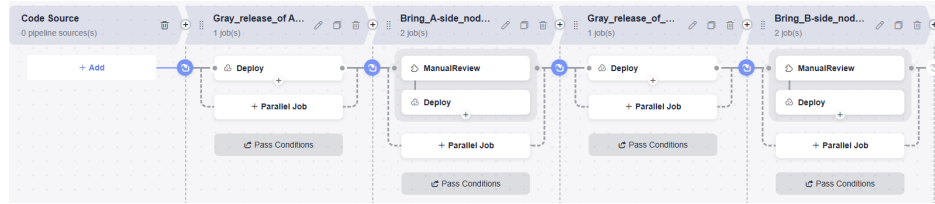


Table 3-25 Parameter description

Parameter	Ma nd ato ry	Description
Name	Yes	Example: Bring_B-side_node_online.
Select Task	Yes	Select Bring_B-side_node_online.
Build Task	No	Leave it blank for this example.

Step 6 After the preceding operations are complete, click **Save and Run** to run pipeline jobs.



----End

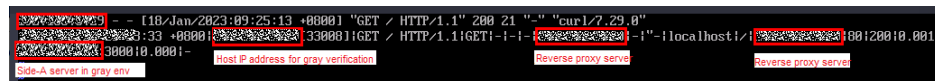
Step 7: Run Pipeline and Manually Perform Gray Verification to Check Whether A-Side and B-Side Nodes Are Normal

When CodeArts Pipeline is executed to bring node A or B online, pipeline execution is suspended. Gray users need to manually verify whether the servers on node A or B in the gray environment are working. Continue to run the pipeline if the servers are working.

Gray users can run the **curl** command to check whether the gray environment is normal.

```
curl http://IP address of the reverse proxy server:Nginx port
```

To check whether the gray users have accessed the target gray environment server, log in to the reverse proxy server and go to the path **logs/access.log** to check logs. If the following command output is displayed, the status is normal.



Appendixes

- Example code to bring A-side node offline

```
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    log_format main '$time_local$remote_addr[$remote_port]$request$request_method|'
$content_length|'
$content_type|$http_referer|$host|$http_x_forwarded_for|'
'$http_true_client_ip|$server_name|$request_uri|$server_addr|$server_port|'
'$status|$request_time|$upstream_addr|$upstream_response_time|$cookie_domain_tag';
access_log logs/access.log main; #Access log: storage path and log level
error_log logs/error.log; #Error log: storage path
sendfile on;
keepalive_timeout 65;
upstream portal {
    #Enter the IP address and application service port number of host A.
    #server X.X.X.X:X; #Bring node A offline.
    #Enter the IP address and application service port number of host B.
    server X.X.X.X:X;
}
upstream portal_test {
    #Enter the IP address and application service port number of host A.
    server X.X.X.X:X;
    #Enter the IP address and application service port number of host B.
    server X.X.X.X:X;
}
```

```
server {
    listen    XXX;#Enter the Nginx port number.
    server_name localhost;

    location / {
        set $backend portal;
        set $test portal_test;
        #Enter the IP address of the gray verification host.
        #if ( $remote_addr ~* "X.X.X.X") {
        #    set $backend $test;
        #}
        proxy_pass https://$backend;
    }
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root html;
    }
}
```

- **Deployment node**

```
#Obtain the application process ID.
pid=`ps -ef | grep app_name | grep -v grep | awk '{print $2}'`
if [ -z "$pid" ];
then
    echo "[app_name pid is not exist.]"
else
    echo "app_name pid: $pid "
    #End the process.
    kill -15 $pid
fi
#Restart the application. You can run the deployment script or command to start the application.
#Method 1: Run the deployment script to start the application.
# sh startup.sh
#Method 2: Run the command to start the application. nohup is recommended for backend startup.
# nohup java -jar /usr/local/app/SpringbootDemo.jar &
```

- **Example code to bring A-side node online to the gray environment**

```
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    log_format main '$time_local$remote_addr[$remote_port]$request$request_method|
$content_length|
$content_type|$http_referer|$host|$http_x_forwarded_for|
$http_true_client_ip|$server_name|$request_uri|$server_addr|$server_port|
$status|$request_time|$upstream_addr|$upstream_response_time|$cookie_domain_tag';
    access_log logs/access.log main; #Access log: storage path and log level
    error_log logs/error.log; #Error log: storage path
    sendfile on;
    keepalive_timeout 65;
    upstream portal {
        #Enter the IP address and application service port number of host A.
        #server X.X.X.X:X; #Bring node A offline.
        #Enter the IP address and application service port number of host B.
        server X.X.X.X:X;
    }
    upstream portal_test {
        #Enter the IP address and application service port number of host A.
        server X.X.X.X:X; #Gray release of node A
        #Enter the IP address and application service port number of host B.
        #server X.X.X.X:X;
    }

    server {
        listen XXX;#Enter the Nginx port number.
        server_name localhost;
```

```
location / {
    set $backend portal;
    set $test portal_test;
    #Enter the IP address of the gray verification host.
    if ( $remote_addr ~* "X.X.X.X" ) {
        set $backend $test;
    }
    proxy_pass https://$backend;
}
error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root html;
}
}
```

- **Example code to bring B-side node offline**

```
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    log_format main '$time_local[$remote_addr[$remote_port]]$request[$request_method]
$content_length'
    '$content_type|$http_referer|$host|$http_x_forwarded_for|'
    '$http_true_client_ip|$server_name|$request_uri|$server_addr|$server_port|'
    '$status|$request_time|$upstream_addr|$upstream_response_time|$cookie_domain_tag';
    access_log logs/access.log main; #Access log: storage path and log level
    error_log logs/error.log; #Error log: storage path
    sendfile on;
    keepalive_timeout 65;
    upstream portal {
        #Enter the IP address and application service port number of host A.
        server X.X.X.X:X;
        #Enter the IP address and application service port number of host B.
        #server X.X.X.X:X; #Bring node B offline.
    }
    upstream portal_test {
        #Enter the IP address and application service port number of host A.
        server X.X.X.X:X;
        #Enter the IP address and application service port number of host B.
        server X.X.X.X:X;
    }

    server {
        listen XXX;#Enter the Nginx port number.
        server_name localhost;

        location / {
            set $backend portal;
            set $test portal_test;
            #Enter the IP address of the gray verification host.
            #if ( $remote_addr ~* "X.X.X.X" ) {
            # set $backend $test;
            #}
            proxy_pass https://$backend;
        }
        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
            root html;
        }
    }
}
```

- **Example code to bring B-side node online to the gray environment**

```
worker_processes 1;
events {
```

```
worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    log_format main '$time_local$remote_addr[$remote_port]$request$request_method|
$content_length|
    '$content_type|$http_referer|$host|$http_x_forwarded_for|
    '$http_true_client_ip|$server_name|$request_uri|$server_addr|$server_port|
    '$status|$request_time|$upstream_addr|$upstream_response_time|$cookie_domain_tag';
    access_log logs/access.log main; #Access log: storage path and log level
    error_log logs/error.log; #Error log: storage path
    sendfile on;
    keepalive_timeout 65;
    upstream portal {
        #Enter the IP address and application service port number of host A.
        server X.X.X.X:X;
        #Enter the IP address and application service port number of host B.
        #server X.X.X.X:X; #Bring node B offline.
    }
    upstream portal_test {
        #Enter the IP address and application service port number of host A.
        #server X.X.X.X:X;
        #Enter the IP address and application service port number of host B.
        server X.X.X.X:X; #Gray release of node B
    }

    server {
        listen XXX;#Enter the Nginx port number.
        server_name localhost;

        location / {
            set $backend portal;
            set $test portal_test;
            #Enter the IP address of the gray verification host.
            if ( $remote_addr ~* "X.X.X.X") {
                set $backend $test;
            }
            proxy_pass https://$backend;
        }
        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
            root html;
        }
    }
}
```

- **Example code to bring a node online**

```
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    log_format main '$time_local$remote_addr[$remote_port]$request$request_method|
$content_length|
    '$content_type|$http_referer|$host|$http_x_forwarded_for|
    '$http_true_client_ip|$server_name|$request_uri|$server_addr|$server_port|
    '$status|$request_time|$upstream_addr|$upstream_response_time|$cookie_domain_tag';
    access_log logs/access.log main; #Access log: storage path and log level
    error_log logs/error.log; #Error log: storage path
    sendfile on;
    keepalive_timeout 65;
    upstream portal {
        #Enter the IP address and application service port number of host A.
        server X.X.X.X:X;
        #Enter the IP address and application service port number of host B.
        server X.X.X.X:X;
    }
}
```

```
upstream portal_test {
    #Enter the IP address and application service port number of host A.
    server X.X.X.X:X;
    #Enter the IP address and application service port number of host B.
    server X.X.X.X:X;
}

server {
    listen      XXX;#Enter the Nginx port number.
    server_name localhost;

    location / {
        set $backend portal;
        set $test portal_test;
        #Enter the IP address of the gray verification host.
        #if ( $remote_addr ~* "X.X.X.X") {
        #    set $backend $test;
        #}
        proxy_pass https://$backend;
    }
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root html;
    }
}
```


4 Using Kubernetes Nginx-Ingress for Gray Release

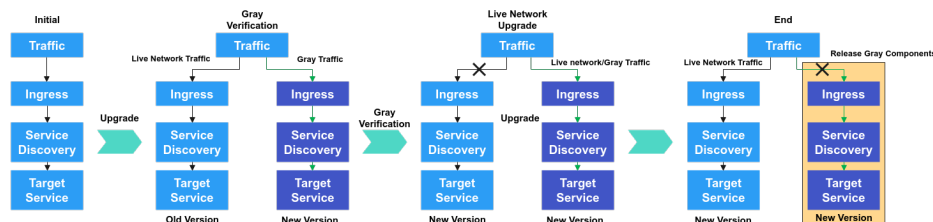
Application Scenario

This practice implements gray release based on native Kubernetes features. When you upgrade to a new system, services may be stopped or gray verification may fail. The native Kubernetes service features help you upgrade system smoothly without affecting services.

Solution Architecture

During system upgrade, a group of gray loads are created when developers deploy applications for the first time. In this case, the system version in the gray loads is the new version. The Service forwards some traffic to the gray loads, and the testers verify the version in the gray loads. After the version verification is complete, the developers start to deploy the application for the second time to upgrade the services on the live network. In this case, the Service forwards all traffic to the gray loads and upgrades the services to the latest version on the live network. After the upgrade is complete, the Service forwards all traffic back to the live network load and releases the gray loads. Now, the new system is released.

Figure 4-1 Gray release scheme



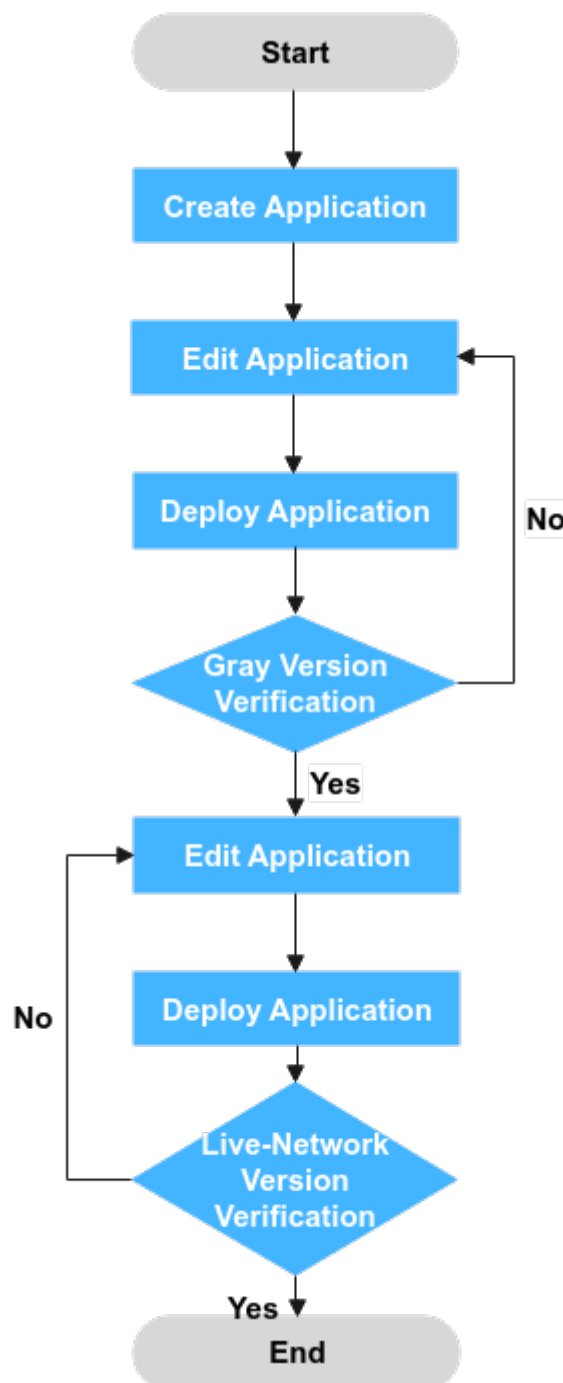
Prerequisites

- A project is available. If there is no project, [create one](#) first.
- You have the permissions to create applications. For details, see [Configuring Permissions for Different Roles](#).
- The service contains the following resources and they are defined as version 1:

- A CCE cluster, for example, **cce-demo**, is available.
- A deployment, for example, **deployment-doc**, has been created in the CCE cluster.
- A Service, for example, **service-doc**, has been created in the CCE cluster.
- A route, for example, **ingress-doc**, has been created in the CCE cluster.
- The nginx-ingress plug-in has been installed in the CCE cluster.

Process

Figure 4-2 Process flowchart



Step 1 Create an application.

1. Go to the CodeArts homepage and click the target project name to access the project.
2. Choose **CICD > Deploy** and click **Create Application**. The **Set Basic Information** page is displayed.
3. You can modify the following basic information as required:

Parameter	Mandatory	Description
Name	Yes	Name of an application. Example: Kubernetes_Nginx-Ingress_Gray_Deployment
Project	Yes	Retain the default value. Project to which an application belongs.
Description	No	Description of an application. Example: None
Execution Agent Pool	No	An agent pool is a collection of physical environments where commands are executed during software package deployment. You can use the official agent pool hosted by Huawei Cloud or host your own servers as a self-hosted agent pool on Huawei Cloud. For details about how to host your own servers, see Creating a Self-hosted Agent Pool . Example: Official
Deploy from pipeline	No	After this function is enabled, the app can be executed only by the pipeline driver and cannot be executed independently.

4. After editing the basic application information, click **Next**. On the **Select Template** page, select **Blank Template** and click **OK**.

Step 2 Edit the application.

On the **Deployment Actions** tab page, add **Kubernetes Nginx-Ingress Gray Deployment (CCE cluster)** and modify the parameters described in the following table.

Table 4-1 Parameter description

Parameter	Description	Example
Action Name	Name of an action displayed in Deployment Actions area.	Retain the default value.

Parameter	Description	Example
Tenant	<ul style="list-style-type: none">• Current tenant: The software package is deployed in your CCE cluster for release. Select Current tenant. You must have the CCE cluster operation permission. If you do not have it, select Authorized User for deployment.• Other tenant: The software package is deployed in the CCE cluster of another tenant for release in IAM authorization mode. If you select Other tenant, you must select an authorized tenant to deploy the CCE cluster.	Select Current tenant .
Authorized User	If you do not have the permission to execute an API, this parameter enables you to obtain the temporary AK/SK of the parent user to execute the CCE API through IAM.	Deselect it.
Region	Select the region for deployment.	Retain the default value.
Cluster Name	Select the Kubernetes cluster applied on CCE.	cce-ldf
Namespace	Select the namespace of the Kubernetes cluster on CCE.	Retain the default value.
Workload	Select the target deployment.	deployment-doc
Service	Name of the service bound to the target workload.	service-doc
Ingress	Select the name of the route bound to the target service.	ingress-doc
Container	Select the name of the CCE container to be deployed.	container-1
Image	Select the image to be deployed.	Retain the default value.
Image Tag	Select the tag of the image to be deployed.	v2

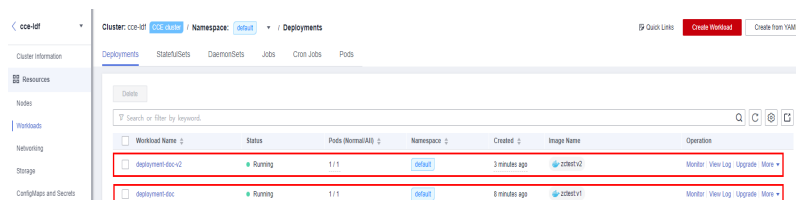
Parameter	Description	Example
Enabling grayscale configuration	Gray Strategy: <ul style="list-style-type: none">• Header Header-Key: You can enter the key of a custom header. Header-Value: You can enter a custom header value. The value can be a character string or a regular expression. The regular expression format is <code>^....\$</code>. Gray Traffic Weight (%): Traffic can be customized.• Cookie Cookie: Custom cookie content can be entered. Gray Traffic Weight (%): Traffic can be customized. Length limit for Header and Cookie : 500 characters each.	Selected Gray Strategy: Header Header-Key: foo Header-Value: bar Gray Traffic Weight(%): 30

Step 3 Deploy an application (create a gray version).

Click **Save & Deploy** to deploy the application. CodeArts Deploy has created the following gray version resources in the CCE cluster and diverts 30% of the live network traffic to the gray load.

- **Workload:** **deployment-doc-v2**. The image version is V2.

Figure 4-3 Adding a workload whose image version is V2



- **Service:** **service-doc-v2**
- **Route:** **ingress-doc-v2**

In this case, you can add a data record (the value of **Key** is **foo** and the value of **Value** is **bar**) to the header to verify the latest version in the gray load.

Step 4 Edit the application (deploy the latest version).

Go to the application created in **Step 1** and modify the following parameters.

Table 4-2 Parameter description

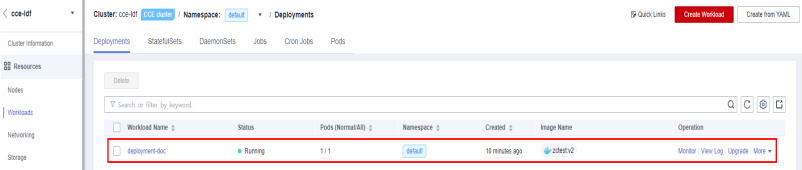
Parameter	Mandatory	Example
Enabling grayscale configuration	No	Deselect it.

Step 5 Deploy the application (deploy the latest version).

Click **Save & Deploy** to deploy the application. CodeArts Deploy has deleted the following gray environment resources from the CCE cluster and replaced the V1 image with the V2 image:

- **Workload:** deployment-doc-v2
- **Service:** service-doc-v2
- **Route:** ingress-doc-v2

Figure 4-4 The image version is upgraded to V2.



You can check whether the system is the latest version on the live network. For more deployment FAQs, see [Application Deployment](#).

----End

5 Security Best Practices

Huawei Cloud and you share the responsibility for security. Huawei Cloud is responsible for the security of cloud services for a secure cloud. As a tenant, you should utilize the security capabilities provided by cloud services to protect data and securely use the cloud. For details, see [Shared Responsibilities](#).

This section provides actionable guidance for enhancing the overall security of CodeArts Deploy. You can use multiple security capabilities provided by CodeArts Deploy by referring to this document, improving the overall security defense and preventing data leakage and tampering during transmission.

You can evaluate the usage of CodeArts Deploy and make security configurations from the following dimensions as required.

- [Optimizing Network Configurations to Reduce Network Attack Risks](#)
- [Properly Managing Host Accounts and Passwords/Keys to Reduce Data Leakage Risks](#)
- [Strengthening Permissions Management to Reduce Related Risks](#)
- [Enabling the CTS Log Audit Function for Backtracking](#)
- [Hardening Sensitive Deployment Parameters](#)

Optimizing Network Configurations to Reduce Network Attack Risks

1. Deploy target hosts on the internal network.
Target hosts carry your services. You should deploy target hosts on the internal network and use routers or firewalls to protect target hosts. Service nodes cannot be accessed from the Internet by EIP binding. In this way, unauthorized access and DDoS attacks can be prevented.
You can deploy services on the internal network by referring to the [agent mode](#).
2. Configure security group for target hosts in the direct connection mode.
For target hosts that are directly connected using EIPs, SSH is used to connect to the target hosts during task execution. The default port is **22** (Windows hosts use WinRM to connect to target hosts. The default port is **5986**). Related ports are vulnerable to attacks. To prevent ports from being exposed to the public network, you need to [configure a security group](#) for your target hosts. The target hosts can be accessed only by using IP addresses related to CodeArts Deploy.

For details about the external IP address of CodeArts Deploy, see [Notes and Constraints](#).

Properly Managing Host Accounts and Passwords/Keys to Reduce Data Leakage Risks

1. Periodically change the host passwords/keys.
The host passwords or keys managed by CodeArts Deploy have high permissions. You are advised to periodically change the passwords or keys.
2. Set Password Complexity.
Target hosts are easy to be attacked. Keep your database accounts and passwords secure.
Set password complexity for your hosts to avoid weak passwords.

Strengthening Permissions Management to Reduce Related Risks

Do not manage hosts as the **root** user.

CodeArts Deploy uses the account passwords or keys set during host management for deployment task execution. The **root** user has high system permissions. Therefore, do not manage hosts as the **root** user.

Enabling the CTS Log Audit Function for Backtracking

After CodeArts Deploy interconnects with CTS, key operations performed on CodeArts Deploy can be recorded in CTS for future audit.

For details about how to enable and configure CTS, see [Overview](#).

Hardening Sensitive Deployment Parameters

Set sensitive parameters to private parameters.

If a parameter is private, the system encrypts the input for storage and only decrypts the parameter when you use it. **Runtime Settings** cannot be set.

Basic Information Deployment Actions Parameters Deployment Records Environment Management Permissions Notifications						
Custom	Predefined	Q Enter a name or default value.				
Name	Type	Default Value	Private Parameter	Runtime Settings	Description	
password	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
sk	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

6 CodeArts Deploy Permission Management Best Practices

Overview

The CodeArts Deploy permission management mainly involves the management of these three parts: application, host cluster, and environment. Application permissions are divided into project-level application permissions and application-level permissions.


Project-level application permissions serve as templates for all applications within a project. These permissions are applied by default when creating a new application. If you need to modify the permissions for a specific application, you can cancel the project-level configurations in the application's permission management and make the necessary adjustments.

Preparations

- You have [enabled and authorized CodeArts Deploy](#).
- You have [created a project](#) (select the Scrum template and name it **Project_Test**).

Accessing CodeArts Deploy in a Project

Step 1 [Log in to the Huawei Cloud console](#).

Step 2 Click  in the upper left corner of the page and choose **Developer Services > CodeArts**.

Step 3 Click **Go to Workspace**. The CodeArts homepage is displayed.

Step 4 Go to the **Project_Test** project page created in the preparation.

Step 5 In the navigation pane, choose **CICD > Deploy**.

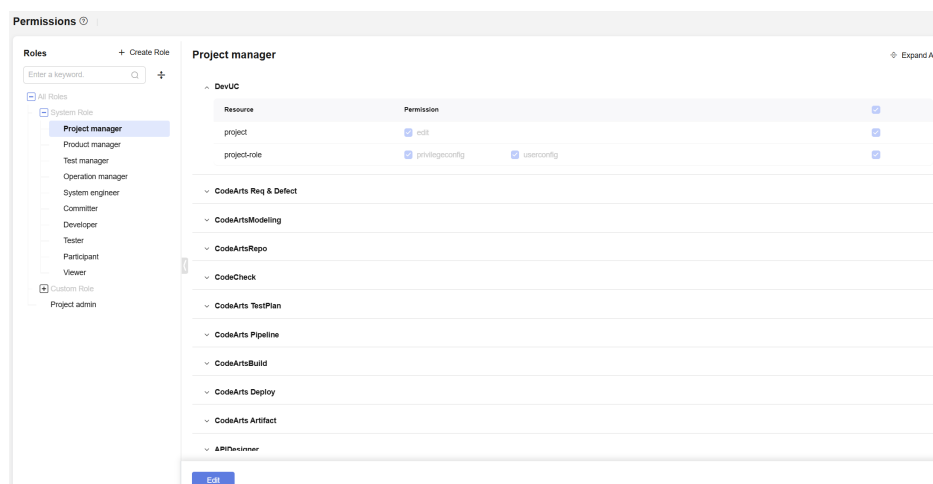
----End

Editing Project-Level Application Permissions

Step 1 [Accessing CodeArts Deploy in a Project.](#)

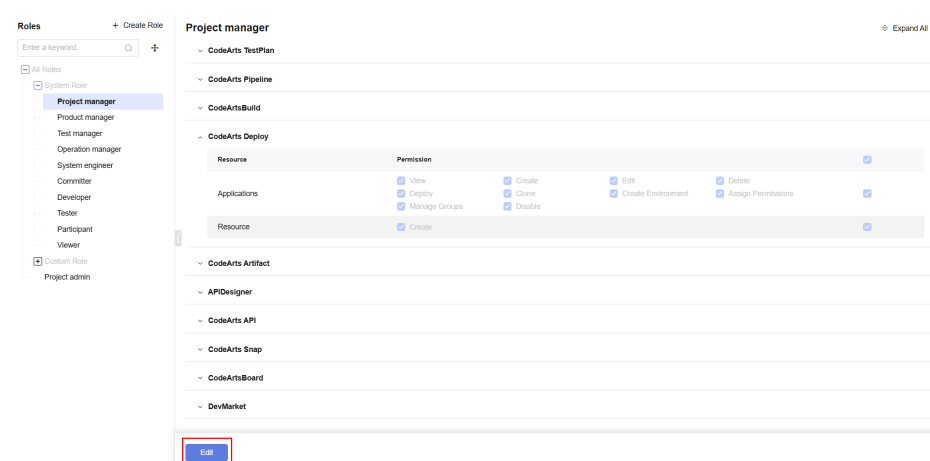
Step 2 In the navigation pane, choose **Settings > Permissions**. The project-level permission management page is displayed, as shown in [Project-level permission management](#).

Figure 6-1 Project-level permission management



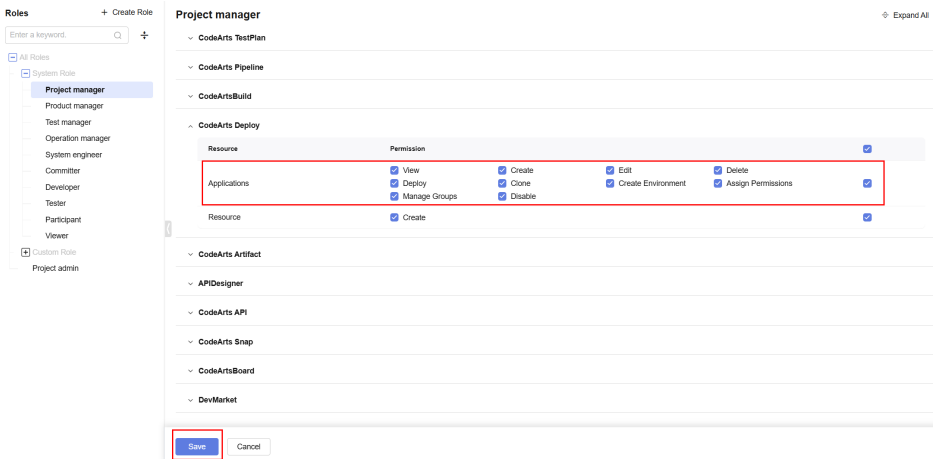
Step 3 On the role list page on the left, select the role whose permissions you want to edit and expand the **Deploy** service tab page on the right. Click **Edit** in the lower left corner to edit the project-level deployment application permissions, as shown in [Figure 6-2](#).

Figure 6-2 Editing project-level deployment application permissions



Step 4 After the editing is complete, click **Save** to save the project-level deployment application permissions of the corresponding role, as shown in [Saving project-level deployment application permissions](#).

Figure 6-3 Saving project-level deployment application permissions



-----End

Editing Application Permissions

- Step 1 **Deploy a Service in a Project.**
- Step 2 Click **Create Application**. On the displayed **Basic Information** page, as shown in **Basic Information**, configure the basic information about the deployment application by referring to **Application basic information**.

Figure 6-4 Basic information

Create Application

Basic Information

Name

Deploy-Test

Project

Project-Test

Agency URN

Create Agency

Example: iam::{{Account ID}}:agency:{{Agency name}}

Description

Enter a description.

0/1,024

Execution Resource Pool

☒ Default ☐ Self-hosted

Deploy from pipeline

☐

Only pipelines will be able to execute this application.

Table 6-1 Basic application parameters

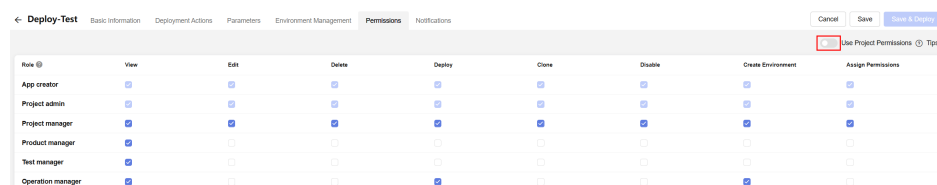
Parameter	Description
Name	Application name. Enter Deploy-Test . Enter 3 to 128 characters, including letters, digits, hyphens (-), and underscores (_).
Project	Project where the deployment application is located. Select the created project Project_Test .
Description	Enter an application description with a maximum of 1,024 characters.
Execution Agent Pool	Select Default .
Deploy from pipeline	Deselect this parameter.

Step 3 After configuring the basic information, click **Next**. On the displayed **Select Template** page, select the **Deploy a General Application** template and click **OK**.

Step 4 Switch to the **Permissions** page. By default, **Use Project Permissions** is enabled, and the **Project-level permission management** change is synchronized.

Step 5 Click **Use Project Permissions** to disable the synchronization of project-level permission changes and modify role permissions at the application level.

Figure 6-5 Disabling the functions of using project permission configuration and customizing permissions



Step 6 Customize and adjust the role permissions of the application as required. After the adjustment is complete, click **Save** to save the application information.

----End

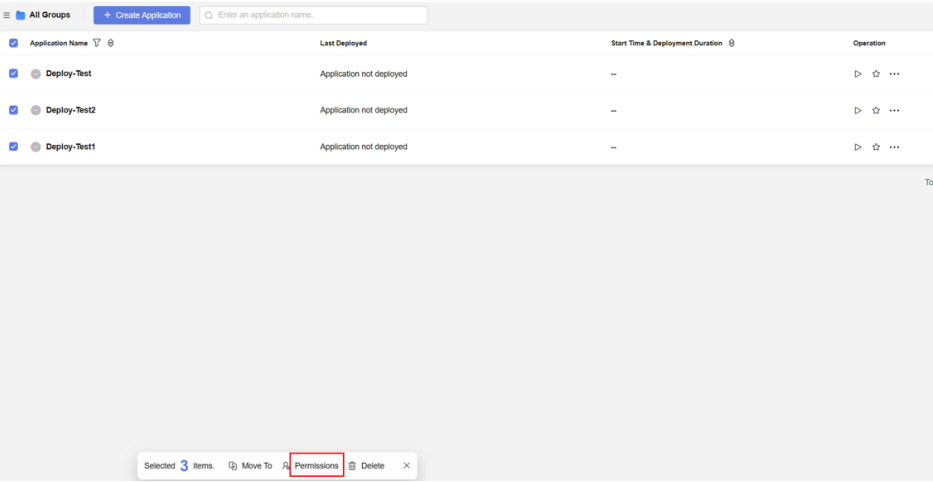
Editing Application Permissions in Batches

Step 1 **Deploy a Service in a Project.**

Step 2 You can select multiple applications at a time.

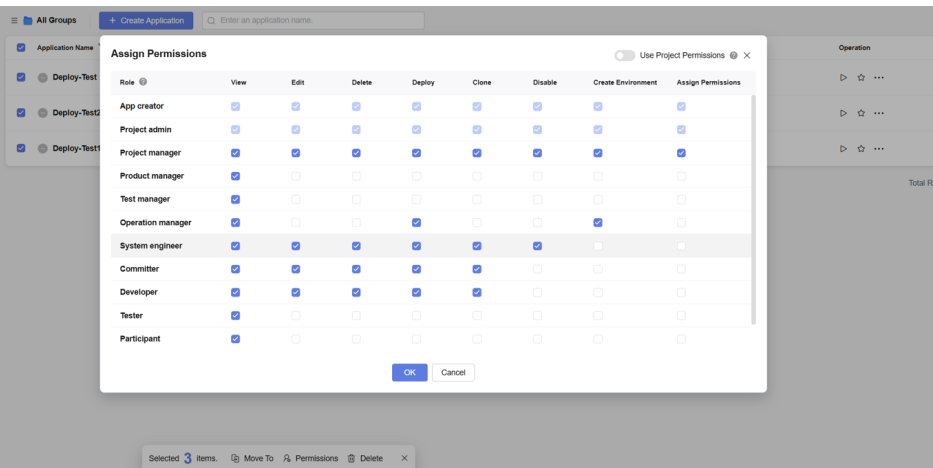
Step 3 Click **Permissions** in the batch operation area in the lower part of the page. The **Permissions** dialog box is displayed.

Figure 6-6 Application permission management



Step 4 Enable or disable **Use Project Permissions** as required. If it is disabled, you can set permissions for selected applications in batches.

Figure 6-7 Setting role permissions for multiple applications in batches



Step 5 Click **OK** to save the permission settings.

----End

Editing Host Cluster Permissions

- Step 1 [Deploy a Service in a Project.](#)
- Step 2 Click **Basic Resources** to switch to the **Basic Resources** page.
- Step 3 Click **Create Host Cluster**. For details, see [Creating a Host Cluster](#). Configure the basic information about the host cluster by referring to [Basic information about the host cluster](#).

Figure 6-8 Creating a host cluster

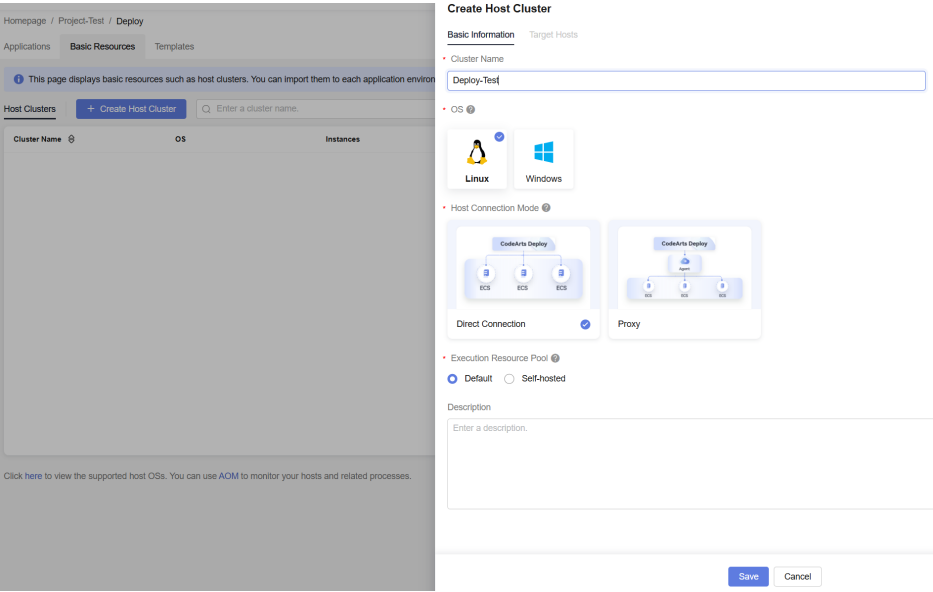
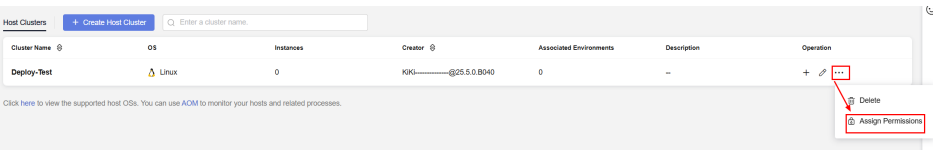


Table 6-2 Basic information about the host cluster

Parameter	Description
Cluster Name	Cluster name. Enter Deploy-Test . Enter 3 to 128 characters, including letters, digits, hyphens (-), and underscores (_).
OS	Select Linux .
Host Connection Mode	Select Direct Connection .
Execution Agent Pool	Select Default .
Description	Enter the description of the host cluster. Enter 0 to 500 characters.

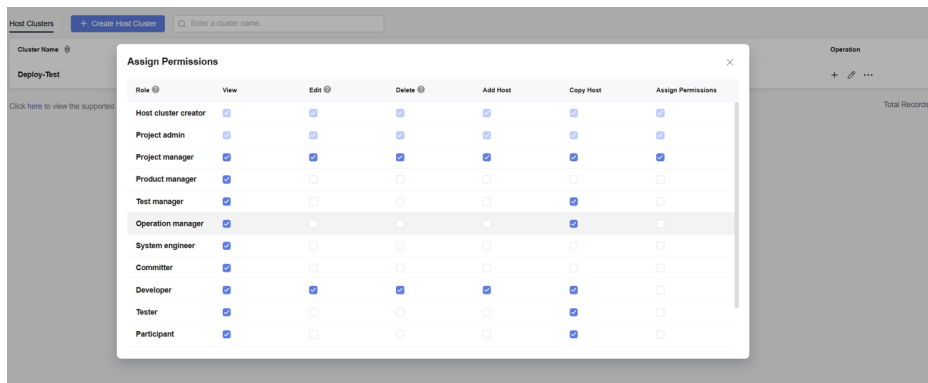
- Step 4 Click **Save** to complete the creation of basic information about the host cluster. Close the dialog box to return to the host cluster list page.
- Step 5 Click **...** in the **Operation** column of the corresponding cluster and choose **Assign Permissions**.

Figure 6-9 Cluster permissions management



Step 6 In the **Assign Permissions** dialog box, set host cluster permissions by selecting or deselecting role permissions.

Figure 6-10 Setting host cluster permissions



Step 7 Close the dialog box.

----End

Editing Environment Permissions

Step 1 [Deploy a Service in a Project](#).

Step 2 Click the name of the created **Deploy-Test** application. The application details page is displayed.

Step 3 Click **Environment Management** to switch to the environment management page.

Step 4 Click **Create Environment**. The **Create Environment** dialog box is displayed, as shown in [Creating an Environment](#).

Figure 6-11 Creating an environment**Create Environment**

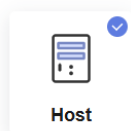
Basic Information

Resources

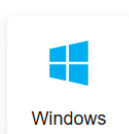
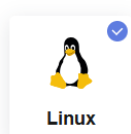
* Environment

Deploy-Test

* Resource Type



* OS



Description

Enter a description.

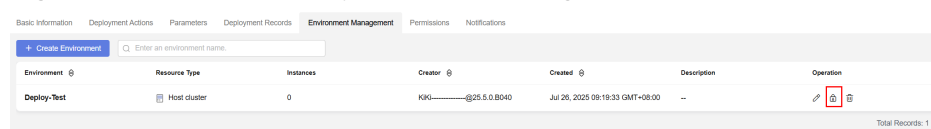
Step 5 Configure basic environment information by referring to the following table.

Table 6-3 Basic host cluster information

Parameter	Parameter Description
Environment	Environment name. Enter Deploy-Test .
Resource Type	Select Host .
OS	Select Linux .
Description	Enter the description of the environment. Max. 500 characters.

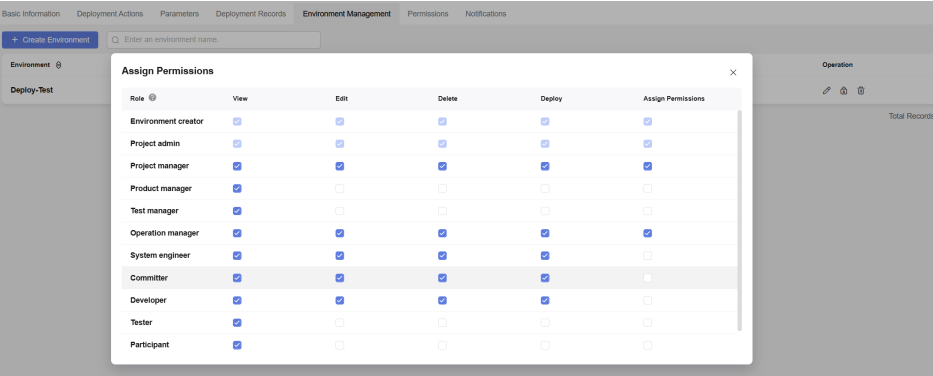
Step 6 Click **Save** to complete the environment creation and close the dialog box to return to the environment list page.

Step 7 Click the  icon in the **Operation** column. The **Permissions** dialog box is displayed, as shown in [Environment permission management](#).

Figure 6-12 Environment permission management

Step 8 Select or deselect role permissions based on service requirements.

Figure 6-13 Setting environment permissions



Step 9 Close the dialog box.

-----End

7 HE2E DevOps Practice: Deploying an Application

7.1 Overview

This section uses the DevOps process example project to describe how to deploy applications on CCE and ECS.

There are three deployment applications preset in the sample project. The first application is used for CCE deployment, and the second and third applications are used for ECS deployment.

Table 7-1 Preset applications

Preset Application	Description
phoenix-cd-cce	For CCE-based deployment.
phoenix-sample-predeploy	For installing dependency tools before ECS-based deployment.
phoenix-sample-standalone	For ECS-based deployment.

7.2 Deploying an Application on CCE

Checking Your CCE Cluster

Before deploying applications, ensure that no workload is running in your CCE cluster.

Step 1 Log in to the [CCE console](#).

Step 2 Click the **phoenix-cce** cluster.

Step 3 Choose **Workloads** from the navigation pane, click the **Deployments** tab, and verify that no record exists in the list.

If there are records in the list, select all records, click **Delete**, select all resource release options, and click **Yes** to clear the records in the list.

----End

Configuring and Executing an Application

Step 1 Go to the **Phoenix Mall** project, and choose **CICD > Deploy** from the navigation pane.

Step 2 In the **Operation** column of the **phoenix-cd-cce** application, click ******* and choose **Edit**.

Step 3 On the **Deployment Actions** tab, complete the following configurations in each action.

Table 7-2 Configuring deployment actions

Parameter	Example	Description
Cluster Name	phoenix-cce	The name of the target cluster.
Namespace	default	The namespace of the target cluster.

Step 4 Click the **Parameters** tab, and check whether the default values of the parameters are the same as those listed in the following table.

Table 7-3 Parameters

Name	Default Value
ci_task_name	phoenix-sample-ci
version	1.0.0

Step 5 Click **Save & Deploy**. In the displayed dialog box, click **OK** to start the deployment.

If a success message shows up, the deployment is successful.

If the deployment fails, rectify the fault based on the failed action and the error information in logs. For details, see [CodeArts Deploy FAQs](#).

----End







Verifying the Deployment Result

Step 1 Go to the CCE console, and click the **phoenix-cce** cluster.

Step 2 In the navigation pane, choose **Workloads**. Click the **Deployments** tab, and verify that the namespace displayed in the upper left corner of the page is **default**.

Five records are displayed on the page. All of them are in the **Running** state.

Figure 7-1 Viewing workloads

<input type="checkbox"/> Workload Name 	Status
<input type="checkbox"/> worker	 Running
<input type="checkbox"/> vote	 Running
<input type="checkbox"/> result	 Running
<input type="checkbox"/> redis	 Running
<input type="checkbox"/> db	 Running

Step 3 Click **vote** to go to the details page. On the **Access Mode** tab, choose **More > Update**.

Step 4 Configure the parameters in the table below and click **OK**.

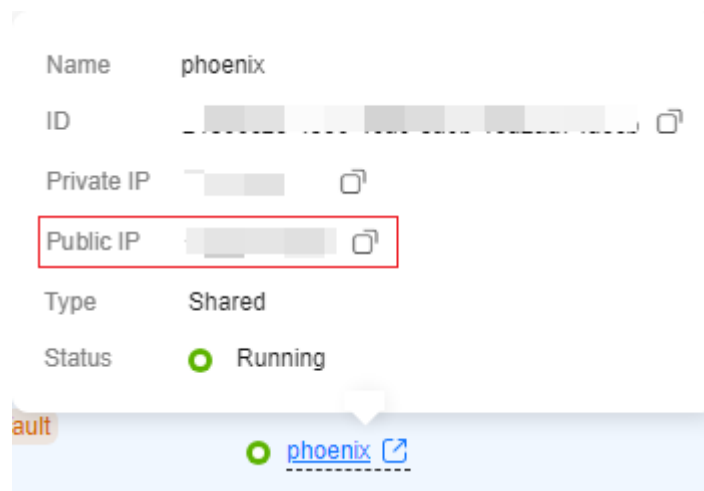
For details about the parameters, see [Creating a LoadBalancer Service](#).

Table 7-4 Updating a Service

Parameter	Example
Service Affinity	Cluster-level
Load Balancer	1. Choose Shared > Auto create . 2. Instance Name : Enter phoenix . 3. EIP : Select Auto create .
Port	<ul style="list-style-type: none">• Container Port: 80• Service Port: 5000

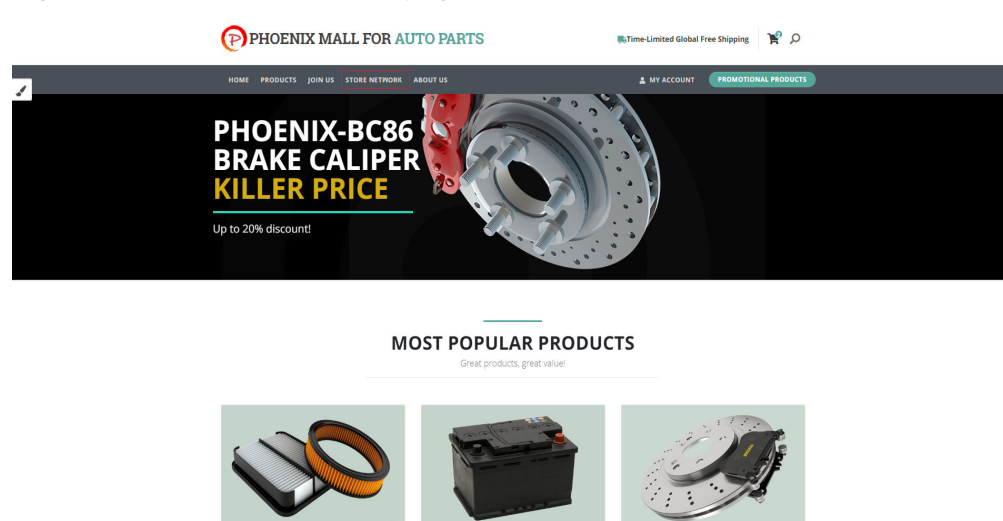
Step 5 Hover over the load balancer name  [phoenix](#) , and copy the public IP address.

Figure 7-2 Copying the access address



Step 6 Open a new browser and enter **http://IP:5000** in the address box. *IP* is the public IP address recorded in [Step 5](#). The **Phoenix Mall** homepage is displayed.

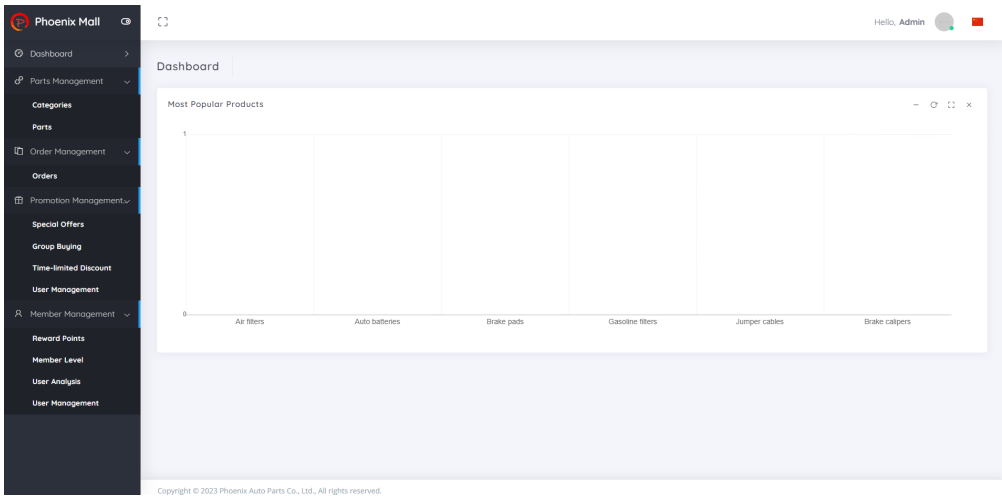
Figure 7-3 Phoenix Mall homepage



Step 7 Return to the **Deployments** page and update **result** (select the **phoenix** load balancer and enter service port **5001**) by referring to [Step 3](#).

Enter **http://IP:5001** in a new browser. The dashboard of **Phoenix Mall** is displayed.

Figure 7-4 Phoenix Mall dashboard



----End

7.3 Deploying an Application on ECS

Adding a Target Host to the Project

Before deploying applications to ECSs, add target hosts as basic resources for the project.

- Step 1
- Go to the **Phoenix Mall** project, and choose **Settings > General > Basic Resources**.
- Step 2
- Click **Create Host Cluster**, configure the following information, and click **Save**.

Table 7-5 Creating a host cluster

Parameter	Example	Description
Cluster Name	phoenix-hostgroup	The name of the host cluster to create. Enter 3 to 128 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are supported.
OS	Linux	The OS of the hosts to add to this cluster. Select Linux or Windows .
Host Connection Mode	Direct Connection	The way your target hosts will connect to CodeArts Deploy. Select Direct Connection or Proxy .

Parameter	Example	Description
Execution Resource Pool	Official	A resource pool (or agent pool) is a collection of physical environments where software packages are deployed using commands. Choose the official agent pool or a self-hosted agent pool that contains your own servers.


Step 3 Click **Add Host** on the **Target Hosts** tab.

Step 4 Select **Importing ECS**, and click **Import** in the **Operation** column of the **phoenix-ecs** host.

Step 5 Configure the following information and click **OK**.

Table 7-6 Adding a host

Parameter	Example	Description
Authorization	Select Password .	The authentication mode for connecting to the ECS. Select Password or Key .
Username	Enter root .	The username for logging in to the ECS. By default, it is root for a Linux ECS.
Password	Enter the password set when buying the ECS.	The password for logging in to the ECS.
SSH Port	Enter 22 .	The default port is 22 . You can also use another one.

Step 6 Check the new record in the target host list. If the **Verification Result** column displays  **Successful**, the host is added successfully.

If the host fails to be added, rectify the fault based on the failure details. For details, see [Host Management FAQs](#).

----End

Installing Dependency Tools on ECS

The sample program depends on Docker and Docker-Compose, which must be installed on the target ECS.

Step 1 Go to the **Phoenix Mall** project, and choose **CICD > Deploy** from the navigation pane.

Step 2 In the **Operation** column of the **phoenix-sample-predeploy** application, click **...** and choose **Edit**.

Step 3 Click the **Environment Management** tab and configure the host environment.

1. Click **Create Environment**, configure the following information, and click **Save**.

Table 7-7 Creating an environment

Parameter	Example	Description
Environment	phoenix-env	The name of the environment to create. Enter 3 to 128 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are supported.
Resource Type	Host	The resource type in the environment. The default value is Host .
OS	Linux	The OS of the hosts to add to this environment. Select Linux or Windows .

2. Click **Import Host** on the **Resources** tab. In the displayed dialog box, select the configured host cluster and host and click **Import**.
3. Check the new host in the resource list.
Close the window. The new environment is displayed in the list.

Step 4 On the **Deployment Actions** tab page, edit the actions of the application.

1. In action **Install Docker**, select **phoenix-env** from the **Environment** drop-down list. If a dialog box is displayed, asking you to confirm whether you want to change the environment to **phoenix-env** for the subsequent actions, click **OK**.
2. Select **Run Shell Commands**, and add the following two command lines to the **Shell Commands** box:

```
docker -v  
docker-compose -v
```

Step 5 Click **Save & Deploy** to start the deployment task.

If a message is displayed indicating successful deployment, the task is successfully executed.

If the deployment fails, rectify the fault based on the failure step and the error message in logs. For details, see [CodeArts Deploy FAQs](#).

Step 6 View the logs. If the log content is similar to the following figure, Docker and Docker-Compose are successfully installed.

Figure 7-5 Viewing deployment logs

```
"Docker version 19.03.9, build 9d988398e7",  
"docker-compose version 1.17.1, build 6d101fb"  
  
----End
```


Configuring and Executing an Application

- Step 1** Go to the **Phoenix Mall** project, and choose **CICD > Deploy** from the navigation pane.
- Step 2** In the **Operation** column of the **phoenix-sample-standalone** application, click **...** and choose **Edit**.
- Step 3** Click the **Environment Management** tab and configure the host environment.
1. Click **Create Environment**, configure the following information, and click **Save**.

Table 7-8 Creating an environment

Parameter	Example	Description
Environment	phoenix-env	The name of the environment to create. Enter 3 to 128 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are supported.
Resource Type	Host	The resource type in the environment. The default value is Host .
OS	Linux	The OS of the hosts to add to this environment. Select Linux or Windows .

2. Click **Import Host** on the **Resources** tab. In the displayed dialog box, select the configured host cluster and host and click **Import**.
3. Check the new host in the resource list.
Close the window. The new environment is displayed in the list.

- Step 4** On the **Deployment Actions** tab page, edit the actions of the application.

1. Add the action **Select a Deployment Source**, and configure the following parameters.

Table 7-9 Configuring the deployment source

Parameter	Example	Description
Source	Build task	The source of the software package to deploy. Select Artifact or Build task .

Parameter	Example	Description
Environment	phoenix-env If a dialog box is displayed, asking you to confirm whether you want to change the environment to phoenix-env for the subsequent actions, click OK .	The target deployment environment. Select the one added on the Environment Management tab.
Build Task	phoenix-sample-ci	Available only when Source is set to Build task .

Step 5 Click the **Parameters** tab page and set parameters.

Table 7-10 Parameters

Name	Default Value
docker_server	Enter the SWR server address obtained from the SWR login command.
docker_username	Enter the username obtained from the SWR login command.
docker_password	Enter the password obtained from the SWR login command.

Obtain the SWR login command from the console. For details, see [Obtaining a Long-Term Login or Image Push/Pull Command](#).

Step 6 Click **Save & Deploy**. In the displayed dialog box, click **OK** to start the deployment.

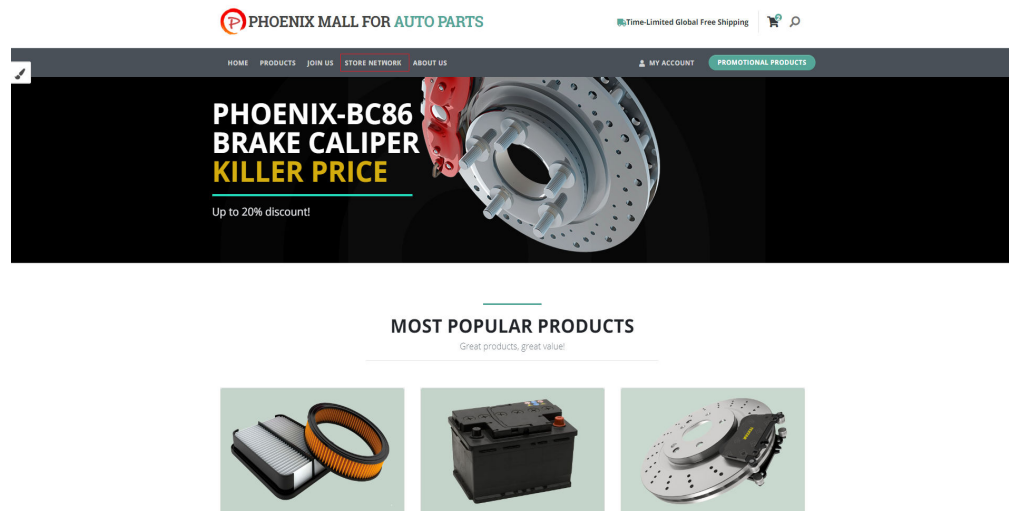
If a success message shows up, the deployment is successful.

If the deployment fails, rectify the fault based on the failure step and the error message in logs. For details, see [CodeArts Deploy FAQs](#).

Step 7 Verify the deployment result.

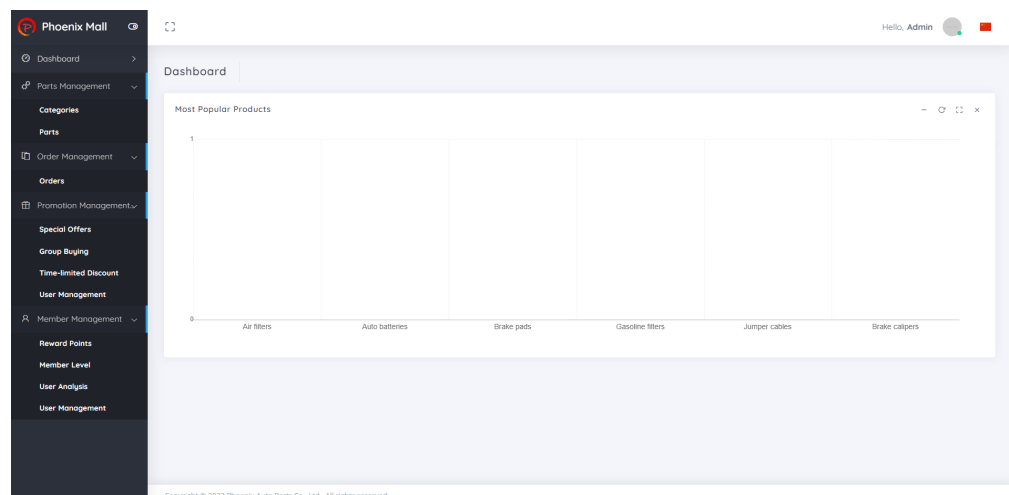
1. Open a browser, enter **http://IP:5000** in the address box, and press **Enter**. *IP* indicates the EIP of the ECS. The Phoenix Mall homepage is displayed.

Figure 7-6 Phoenix Mall homepage



2. Enter **http://IP:5001** and press **Enter**. *IP* indicates the EIP of the ECS. The Phoenix Mall dashboard is displayed.

Figure 7-7 Phoenix Mall dashboard



----End

7.4 Releasing Resources



WARNING

Released resources cannot be recovered. Exercise caution when performing these operations.

The pay-per-use resources involved in this document are from CCE and ECS. Release these resources if you no longer need them after the practice.

Table 7-11 Releasing Resources

Resource	Operation Guide
ECS	Deleting an ECS
CCE	Deleting a Pay-per-Use Cluster