CodeArts Pipeline

Getting Started

Issue 01

Date 2024-11-08

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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/

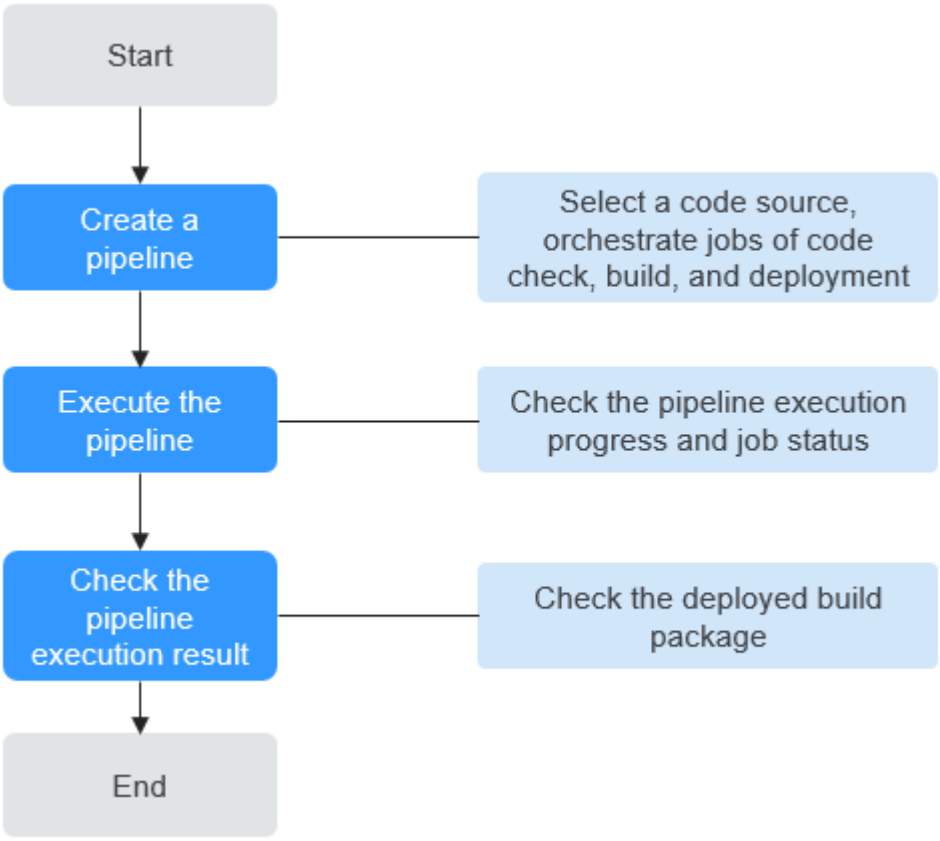
Contents

1 Generating a Software Package and Deploying It on a Host Through CodeArts Pipeline

CodeArts Pipeline is a visualized and automated job scheduling platform. It needs to be used together with automated jobs of services such as CodeArts Build, CodeArts Check, CodeArts TestPlan, and CodeArts Deploy. You can orchestrate these automated jobs as needed. A single configuration triggers executions repeatedly to avoid inefficient manual operations.

In this section, we will create a pipeline to sequentially run code checks, build the software package, and deploy it on a host.

Figure 1-1 Basic operation process



Prerequisites

- **You have enabled and authorized CodeArts Pipeline**.
- You need to prepare a host with an EIP. You can use an existing host or **purchase a Huawei Cloud ECS**.

Preparations

Step 1 Create a project

1. **Log in to the Huawei Cloud console**.
2. Click ☰ in the upper left corner of the page and choose **Developer Services** > **CodeArts Pipeline** from the service list.
3. Click **Access Service** to access the CodeArts Pipeline homepage.
4. On the top navigation bar, click **Homepage**. On the displayed page, click **Create Project**, select **Scrum**, and enter a name **Project01**.
5. Click **OK**.

For details, see **Creating a Scrum Project and a Work Item**.

Step 2 Create a code repository and a code check task.

1. Click the created project to access it, choose **Code** > **Repo** from the left navigation pane.

2. Click **New Repository**, select **Template**, and select the **Java Maven Demo** template.
3. Click **Next** and enter the repository name **Repo01**.

   A code check task with the same name as the code repository is automatically created. Change the task name to **CheckTask01** by referring to **Creating a Task**.
4. Click **OK**.

For details, see **Creating a Repository Using a Template**.

Step 3 Create a build task

1. In the left navigation pane, choose **CICD** > **Build**.
2. Click **Create Task** and configure task information.
   a. Enter a name **BuildTask01**, select the code source **Repo**, select the created repository, select the default branch **master**, and click **Next**.
   b. Select the system template **Maven**, click **OK**. On the displayed **Build Actions** tab page, retain the default configurations.
3. Click **Save**.

For details, see **Creating a Build Task**.

Step 4 Create an application

1. In the left navigation pane, choose **Settings** > **General** > **Basic Resources**, create a host cluster, and add the purchased host to the cluster.
2. In the left navigation pane, choose **CICD** > **Deploy**.
3. Click **Create Application**. On the displayed page, enter an application name **DeployTask01**, click **Next**, select **Blank Template**, and click **OK**.
4. On the **Deployment Actions** tab page, add the **Select Deployment Source** action and configure the following information.
   - Source: Set the source to **Build task**
   - Environment: Click **Create**. On the displayed **Environment Management** page, click **Create Environment** to import the host to the environment.
   - Build task: Select the **created build task**.
   - Download path: Enter the deployment directory of the host. In this example, the directory is **/usr/local/**.
   - Retain default settings for other parameters.
5. Click **Save**.

For details, see **Creating an Application with a Blank Template**.

----End

Create a pipeline

Step 1 Access the **created project**, choose **CICD** > **Pipeline** from the left navigation pane.

Step 2 Click **Create Pipeline** and configure pipeline information.

1. Configure the following information and click **Next**.

Table 1-1 Pipeline basic information

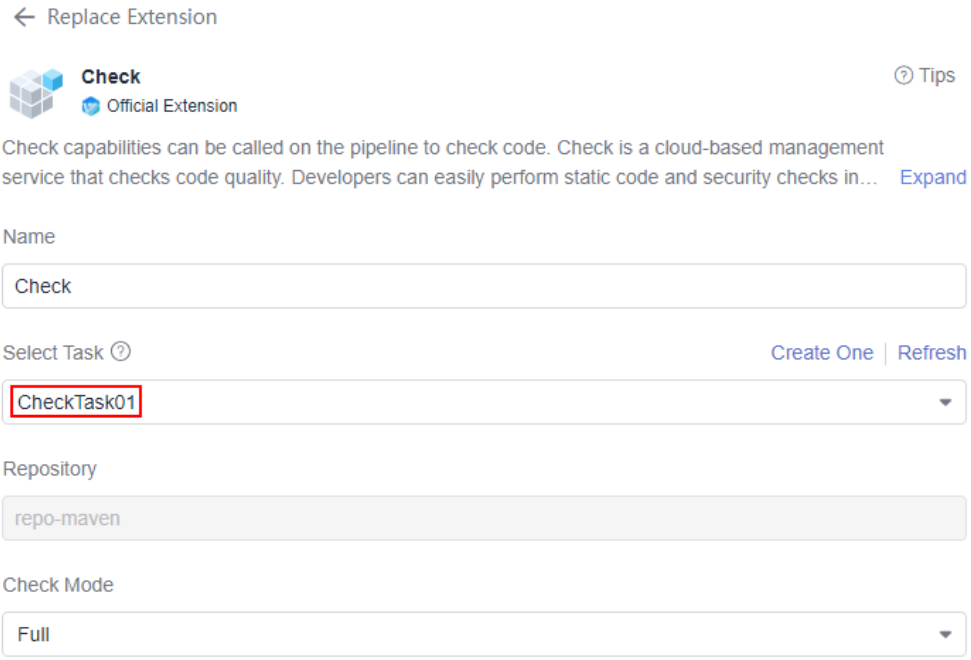
Parameter	Configuration
Name	Enter **Pipeline01**.
Code Source	Select **Repo**.
Repository	Select the **created repository**.
Default Branch	Select **master**.

2. Select **Blank Template** and click **OK**.

Step 3 On the **Task Orchestration** page, two stages (**Code Source** and **Stage_1**) are generated by default. Click **Stage** to add a new stage (**Stage_2**).

1. Add a code check task
   a. Click **New Job** under **Stage_1**.
   b. Click the **Check** type and search for the **Check** extension.
   c. Move the cursor to the extension, click **Add**, select the **created code check task**, and set **Check Mode** to **Full**.

Figure 1-2 Adding a code check task



2. Add a build task
   a. Click **Parallel Job** under **Stage_1**, or click + under the code check job.

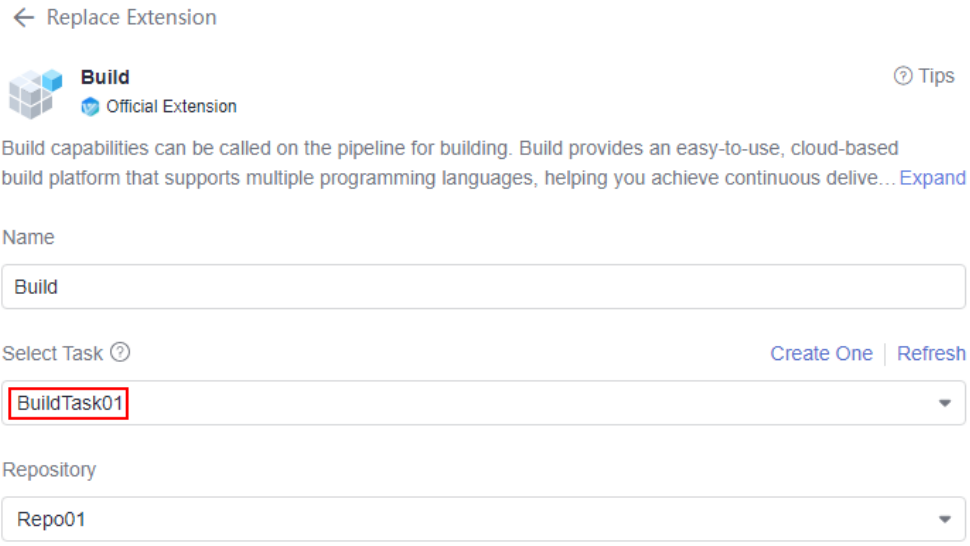
      NOTE

      The code check job and build job can be in parallel or serial.

   b. Click the **Build** type and search for the **Build** extension.

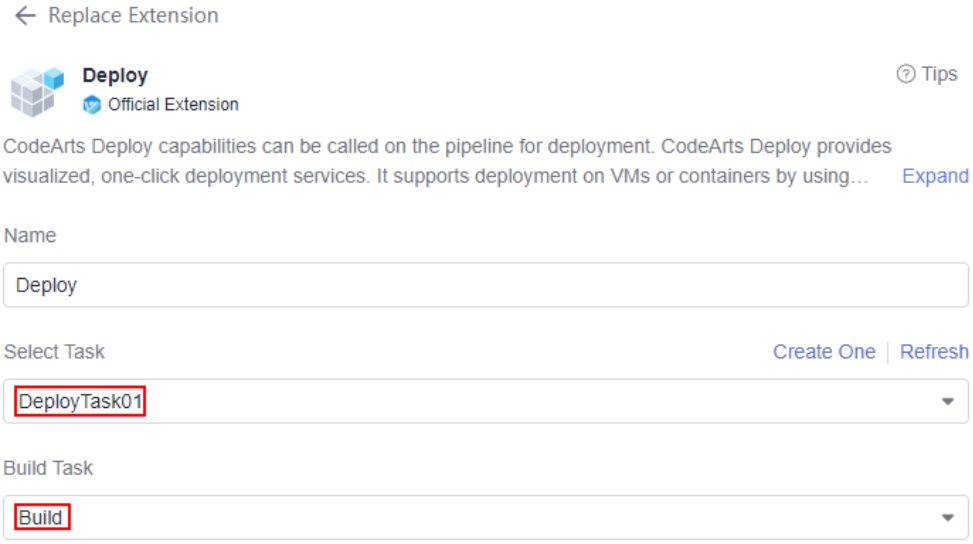
c. Move the cursor to the extension, click **Add**, select the **created build task**, and select the repository associated with the build task.

Figure 1-3 Adding a build task



3. Add an application
   a. Click **Job** under **Stage_2**.
   b. In the displayed dialog box, search for the **Deploy** extension.
   c. Move the cursor to the extension, click **Add**, select **the created application**, and associate it with the added build task.

Figure 1-4 Adding an application



Step 4 After the configuration, click **Save**.

----End

Executing a Pipeline

Step 1 Return to the pipeline list page and click ▷ in the **Operation** column.

Step 2 In the displayed dialog box, retain the default settings, and click **Execute**.

Step 3 Click the pipeline name to go to the **Execution History** page.

Step 4 Click the execution message to check pipeline execution situation in real time.

Click a job card to view its log.

----End

Checking the Pipeline Execution Result

Step 1 Click the avatar in the upper right corner.

Step 2 Click **CodeArts Console**.

Step 3 Click ☰ in the upper left corner and search for **Elastic Cloud Server**. Then, access the **Elastic Cloud Server** console.

Step 4 Locate the ECS used for deployment, click **Remote Login** in the **Operation** column.

Step 5 In the **Other Login Modes** area, select **Log in using Remote Login on the management console** and click **Log In**.

Step 6 Enter the username and password for purchasing the ECS. Press **Enter**.

Step 7 Enter the following command and press **Enter** to go to the directory **/usr/local** configured during **application creation**.

```
cd /usr/local
```

Step 8 Enter the following command and press **Enter** to check the deployed package.

```
ls -al
```

Figure 1-5 Checking the pipeline execution result

```
[root@ecs-71a6 ~]# cd /usr/local
[root@ecs-71a6 local]# ls -al
total 68
drwxr-xr-x. 15 root root 4096 Jun 13 10:16 .
drwxr-xr-x. 13 root root 4096 Feb 27 15:19 ..
drwxr-xr-x.  2 root root 4096 Feb 27 15:40 bin
drwxr-xr-x.  2 root root 4096 Apr 11  2018 etc
drwxr-xr-x.  2 root root 4096 Apr 11  2018 games
drwxr-xr-x  11 root root 4096 Jun 12 17:01 hostguard
drwxr-xr-x.  2 root root 4096 Apr 11  2018 include
-rwxr-x---   1 root root 2234 Jun 13 10:16 javaMavenDemo-1.0.jar
drwxr-xr-x.  3 root root 4096 Feb 27 15:40 lib
drwxr-xr-x.  3 root root 4096 Feb 27 15:40 lib64
drwxr-xr-x.  2 root root 4096 Apr 11  2018 libexec
drwxr-xr-x.  2 root root 4096 Apr 11  2018 sbin
```

----End

Related Information

CodeArts Pipeline provides built-in templates for you to quickly create pipelines. For more information, see **Managing Pipeline Templates**.

