CodeArts

Getting Started

Issue 01

Date 2023-11-30

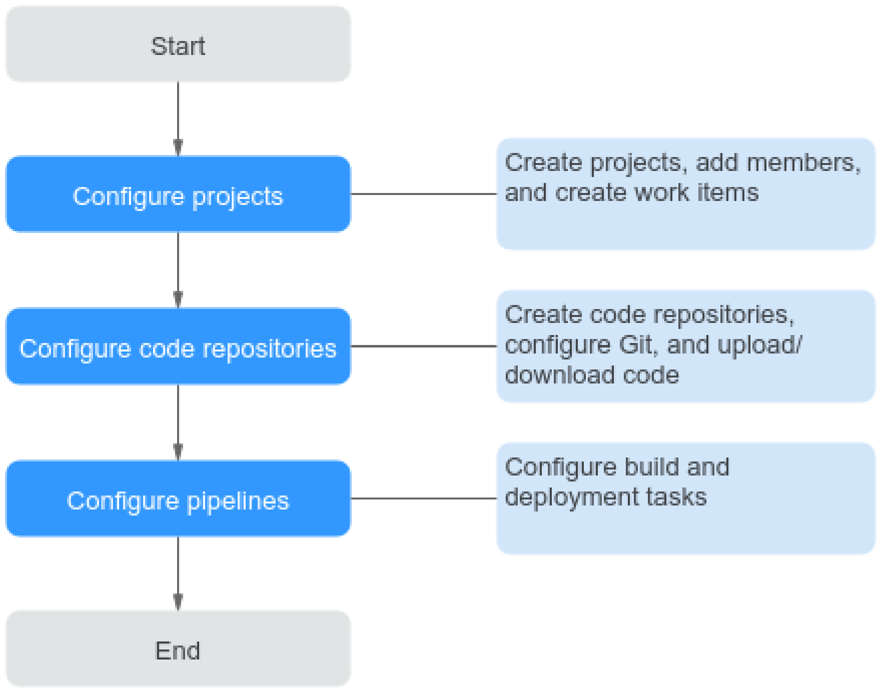
HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Contents

1 Using CodeArts

This section describes the basic operation process of CodeArts.

Figure 1-1 Basic operation process


Start
Configure projects
Create projects, add members, and create work items
Configure code repositories
Create code repositories, configure Git, and upload/download code
Configure pipelines
Configure build and deployment tasks
End

Prerequisites

1. You have purchased CodeArts. For details, see **Purchasing CodeArts**.
2. To deploy applications on a host, you need to prepare a host with an EIP. You can use an existing host or **purchase a Huawei Cloud ECS**.

Configuring a Project

CodeArts Req is the basis for using services on CodeArts. You need to create a project, add project members, and add work items based on your project plan.

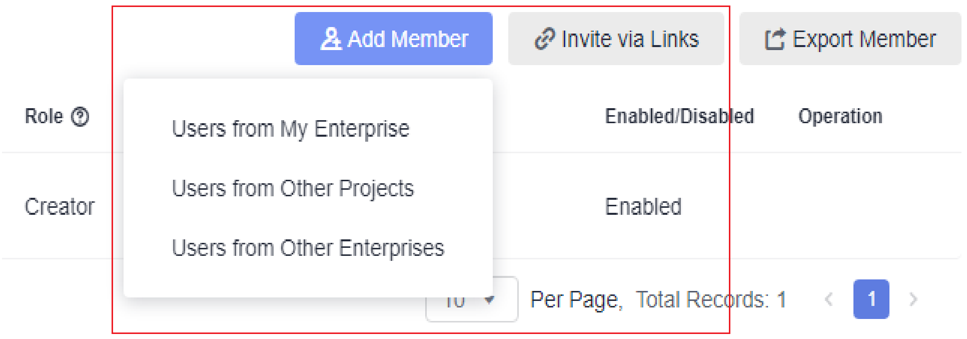
Step 1 Create a project.

1. Log in to the CodeArts console.
2. Click and select a region.
3. Click **Access Service**.
4. Click **Create Project**.
5. Click **Scrum**, enter a project name, and click **OK**.

Step 2 Add a project member.

1. Click a created project and choose **Settings** > **General** > **Members**.
   In the upper right corner of the page, click **Add Member** or **Invite via Links**.
   You can select one of the following options as required:
   – **Users from My Enterprise**
   – **Users from Other Projects**
   – **Users from Other Enterprises**

Figure 1-2 Adding project members



Step 3 Create a work item.

1. Click a project, choose **Work**, and click the **Work Items** tab.
2. Click **Create Work Item**, select a work item type, and enter information such as the title, priority, and handler.

----End

For more operations on CodeArts Req, see CodeArts Req User Guide.

Configuring a Code Repository

The code repository is based on Git to manage versions of project code. Therefore, you need to install the Git client locally in advance.

Step 1 Install and configure the Git client.

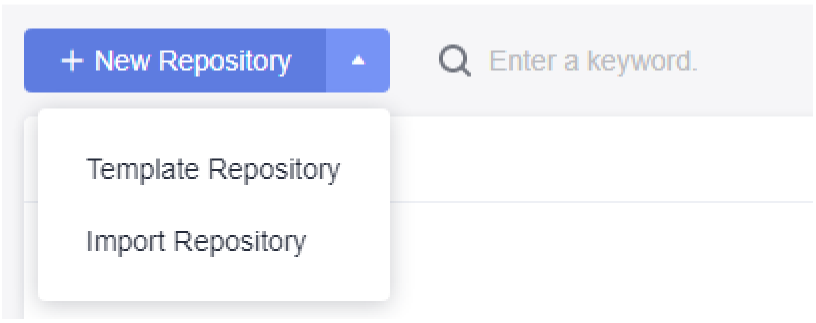
1. Download the installation package from the Git website and install the Git client with the default configurations on the local host.
2. Run **Git Bash** and enter the following commands to configure the username and email address:
```
git config  --global user.name "<name>"
git config  --global user.email "<email_address>"
```
3. Generate a pair of SSH keys: The generated key is stored in **~/.ssh/id_rsa.pub**.
```
ssh-keygen -t rsa -C "email_address"
```

4. Display the key content.
```
cat ~/.ssh/id_rsa.pub
```

Step 2 Create a code repository.

1. Go to the created project and choose **Code** > **Repo**.
2. Click **New Repository**, **Template Repository**, or **Import Repository**, enter basic information such as the repository name as prompted, and click **OK**.

Figure 1-3 Creating a code repository



3. After a code repository is created, click the repository name on the repository list page to view the files in the repository.

Step 3 Clone or push code.

1. Click the username in the upper right corner of the page and choose **This Account Settings** from the drop-down list.
2. In the navigation bar, choose **Repo** > **SSH Keys**.
3. Click **Add SSH Key**, enter the name, enter the key generated in **Installing and Configuring the Git Client**, agree to statements, and click **OK**.
4. Return to the CodeArts Repo page and click the name of the repository to be cloned.
5. Click **Clone/Download** to copy the repository SSH download link.
6. Run **Git Bash** and enter the following command to clone the cloud repository to your local address:
```
git clone repository_SSH_URL
```
7. After editing the code locally, enter the following commands in sequence in Git Bash to save the code and push it to the code repository:
```
git add .
git commit -m "commit_message"
git push origin master
```
8. Return to CodeArts Repo to view the updated file.

----End

For more operations on CodeArts Repo, see **CodeArts Repo User Guide**.

Configuring a Pipeline

CodeArts Pipeline integrates with CodeArts Check, CodeArts Build, and CodeArts Deploy tasks. You can flexibly configure tasks in the pipeline as required. The pipeline is optional.

- CodeArts Check performs static and security checks. The code check is optional.

- CodeArts Build compiles the source code of software into a target file and packs the configuration file and resource file. Build is optional. For some projects, such as PHP and Node.js frontend code, you do not need to configure build tasks.
- CodeArts Deploy deploys a software package or code to a VM or container. Deployment is optional. For some projects, such as mobile app development, this deployment mode is not required.

Step 1 Create a code check task.

1. Go to the created project and choose **Code** > **Check**.
2. Click **Create Task**, select the code repository to check, and click **New** next to the repository name.
3. Enter the created task, go to the **Overview** tab page and click **Start Check**.
4. After the task is successfully executed, you can view the check result, issues, and suggestions.

For more operations on CodeArts Check, see **CodeArts Check User Guide**.

Step 2 Create a build task.

1. Go to the created project and choose **CICD** > **Build** in the navigation bar.
2. Click **Create Task** and configure task information as required.
   a. Basic information: Configure the following information and click **Next**.

Table 1-1 Basic information

Item	Suggestion
Task Name	Enter a custom name.
Code Source	Select **Repo**.
Source Code Repository	Select the code repository created in **Configuring a Code Repository**.
Branch	Select a repository branch as required.

   b. **Build Template**: You can select a built-in or **Blank Template** and click **Next**.
3. Configure the build actions, parameters, execution plan. Click **Create and Run**.
4. After the task is complete, you can view the build result and logs.

For more operations on CodeArts Build, see **CodeArts Build User Guide**.

Step 3 Create and deploy an application.

1. Go to the created project. In the navigation bar, choose **Settings** > **General** > **Basic Resources**, create a host cluster, and add the hosts prepared in **Prerequisites** to the cluster.

2. Go to the created project and choose **CICD** > **Deploy** from the navigation bar.
3. Click **Create Application** and configure task information as required.
   a. Basic information: Enter a custom application name and click **Next**.
   b. Select a deployment template. You can select a built-in or **Blank Template**. Then click **OK**.
4. Configure the deployment actions, parameters, and environments as required, and click **Save & Deploy**.
5. After the application is successfully deployed, you can view the deployment result, logs, and error information on the page.

   For more operations on CodeArts Deploy, see **CodeArts Deploy User Guide**.

Step 4 Configure a pipeline.

1. Go to the created project and choose **CICD** > **Pipeline** from the navigation bar.
2. Click the **Pipelines** tab, click **Create Pipeline**, and configure pipeline information.
   a. Basic information: Configure the following information and click **Next**.

Table 1-2 Pipeline basic information

Item	Suggestion
Project	Enter a custom name.
Pipeline Source	Select **Repo**.
Repository	Select the code repository created in **Configuring a Code Repository**.
Default Branch	Select a repository branch as required.

   b. Template: Select a built-in or **Blank Template**. Then click **Confirm**.
3. Configure the task orchestration, parameters, and execution plan as required, and click **Save and Run**.
4. After the task is successfully executed, click it to view details.

For more operations on CodeArts Pipeline, see **CodeArts Pipeline User Guide**

----End

2 Using CodeArts to Quickly Set Up a Project (ECS)

This section describes how to use the built-in code repository of CodeArts to develop, build, and deploy projects for continuous delivery.

The following describes deployment using ECS. For details about deployment using Cloud Container Engine (CCE), see **Using CodeArts to Quickly Set Up a Project (CCE)**.

Preparations

1. You have purchased CodeArts. For details, see **Purchasing CodeArts**.
2. You have **purchased an ECS**. The following table lists the mandatory configurations. You can select the configurations that are not listed in the table based on the site requirements. After the purchase is complete, add inbound rules for ports 22 and 8080 by referring to **Configuring Security Group Rules**.

Creating a Project

A project is the basis for using services on CodeArts. Subsequent operations can be performed only after a project is created.

Step 1 **Log in to the CodeArts console**.

Step 2 Click and select a region.

Step 3 Click **Access Service**.

Step 4 Click **Create Project**.

Step 5 Select **Scrum**, enter the project name **Demo**, and click **OK**.

----End

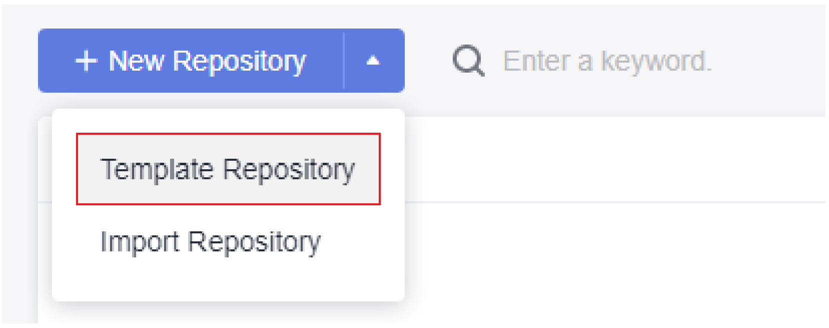
Creating a Code Repository

You can use a code repository to manage project code versions. This section describes how to use the built-in template **Java Web Demo** to create a code repository.

Step 1 On the navigation bar, choose **Code** > **Repo**.

Step 2 Click ▾ next to **New Repository** and select **Template Repository**.

Figure 2-1 Creating a code repository



Step 3 On the page displayed, select **Java Web Demo** and click **Next**.

Step 4 Enter the code repository name **Web-Demo** and click **OK**.

----End

Checking Code

You can use CodeArts Check to perform static code check and control code quality.

Step 1 On the navigation bar, choose **Code** > **Check**. The automatically created task **Web-Demo-codecheck** is displayed on the page.

NOTE

This section uses built-in tasks associated with code repository templates.

In an actual development project, click **Create Task**. For details, see **Creating a Code Check Task**.

Step 2 Locate the row that contains the task and click **Execute immediately**.

NOTE

If the code check task has been run, click the task name to access overview page and click **Start Check** to run the task again.

Step 3 When ✓ is displayed, the task is successfully executed. Click the task name, go to the **Overview** tab page, and view the check result.

If the task fails, check and fix errors based on the message displayed on the page.

----End

Building and Archiving Software Packages

You can use CodeArts Build to compile the source code of the software into a target file, packs the configuration file and resource file, and archives them to a release repo.

Step 1 In the navigation bar, choose **CICD** > **Build**. The automatically created build task **Web-Demo-cloudbuild** is displayed.

NOTE

This section uses built-in tasks associated with code repository templates.

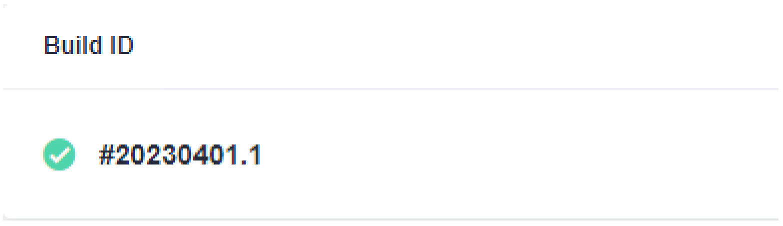
In an actual development project, you need to click **Create Task** and create a task based on the service scenario. For details, see **Creating a Build Task**.

Step 2 Click in the row where the task is located to start the task. If a dialog box is displayed, confirm the parameter settings and click **Confirm**.

Step 3 When is displayed, the task is successfully executed. Click the task name. On the **Build History** page that is displayed, find the **Build ID** of the latest build in the list and record the ID.

If the build fails, rectify the fault based on the failed action information and error information in logs.

Figure 2-2 Build ID



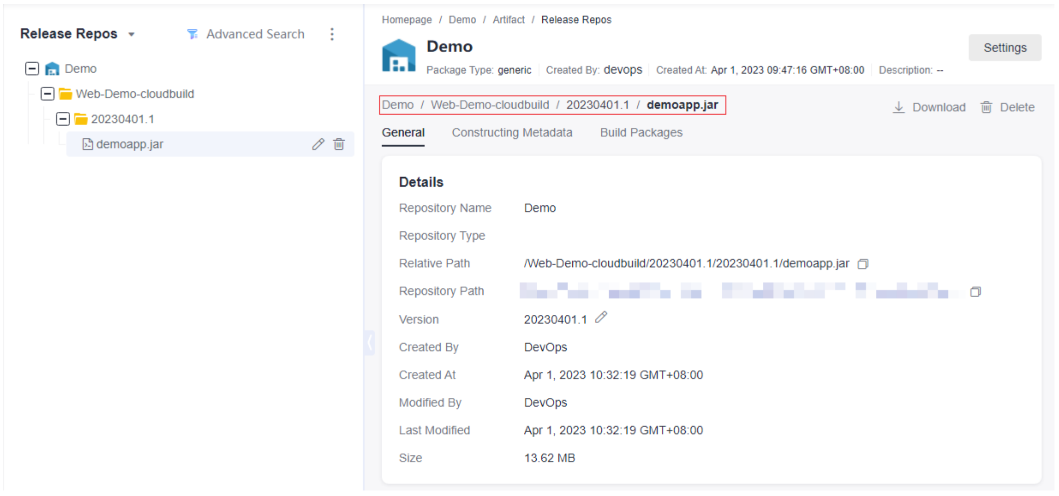
Build ID

#20230401.1

Step 4 Choose **Artifact** in the navigation bar and click the **Release Repos** tab.

In the repository named after the project, go to the folder named after the build task and the folder named after the build number in sequence to find the generated software package **demoapp.jar**.

Figure 2-3 Viewing the software package



Package Type: generic Created By: devops Created At: Apr 1, 2023 09:47:16 GMT+08:00 Description: --

Details

Repository Name	Demo
Repository Type	
Relative Path	/Web-Demo-cloudbuild/20230401.1/20230401.1/demoapp.jar
Repository Path	[redacted]
Version	20230401.1
Created By	DevOps
Created At	Apr 1, 2023 10:32:19 GMT+08:00
Modified By	DevOps
Last Modified	Apr 1, 2023 10:32:19 GMT+08:00
Size	13.62 MB

----End

Deploying the Build Package

You can use CodeArts Deploy to deploy software packages in the release repo to a VM and run it.

Step 1 Configuring the Target Host

1. In the navigation bar, choose **Settings** > **General** > **Basic Resources**.
2. Click **Create Host Cluster**, enter the name **hosts**, set the OS to **Linux**, and click **Save**.
3. Click **Add Target Host**. In the dialog box that is displayed, configure the following information, agree to statements, and click **OK**.

Table 2-1 Add a target host

Item	Suggestion
Host Name	Enter a custom host name. For easy identification, set this parameter to the name of the ECS purchased on **Preparations**.
IP	Enter the IP address of the ECS purchased in **Preparations**.
Username	Enter **root**.
Password	Enter the password set when you purchase the ECS in **Preparations**.
SSH Port	Enter **22**.

4. A host record is displayed on the page. If **Succeed** is displayed in the **Verification Result** column, the host is added successfully.

   If the host fails to be added, check the host configuration based on the failure details.

Step 2 Choose **CICD** > **Deploy** from the navigation bar. The automatically created application **Web-Demo-deploy** is displayed on the page.

NOTE

This section uses built-in applications associated with code repository templates.

In an actual development project, click **Create Application** and create an application based on the service scenario. For details, see **Creating an Application**.

Step 3 Click ··· and choose **Edit** from the drop-down list.

Step 4 Click the **Environment Management** tab and configure the host environment.

1. Click **Create Environment**, enter the environment name **host-group**, select the resource type **Hosts** and operating system **Linux**, and click **Save**.
2. A new environment record is added to the list. Click the environment name. In the window that is displayed, click the **Resource** tab.
3. Click **Import Host**. In the dialog box that is displayed, select the host cluster created in **Step 1** from the drop-down list, select the host from the list, and click **Import**.
4. A message is displayed, indicating that the import is successful. Close the window.

Step 5 Click the **Parameters** tab and set task parameters by referring to the following table.

Parameter Name	Value
host_group	Select the environment name **host-group** added in **Step 4**.
package_url	This parameter is not required. Click 🗑 in the corresponding row to delete it.
service_port	Enter **8080**.
package_name	Enter **demoapp**.

Step 6 Click the **Deployment Actions** tab and configure information.

- **Stop Spring Boot**: If you perform this step for the first time, this step will fail because no service is running on the target host. Therefore, do not select **Enable this action**.
- Install the JDK: Change the JDK version to **openjdk-1.8.0**.
- Select the deployment source. Set the parameters based on the following table.

Table 2-2 Deployment source configuration

Item	Value
Source	Click **Build task**.
Build Task	Select **Web-Demo-cloudbuild**.
Download Path	Enter **/usr/local/${package_name}/**.

- **Health Test Through URLs**: This step is optional. Determine whether to enable it as required. (In this document, do not select **Enable this action**.)

Step 7 Click **Save & Deploy**. If a dialog box is displayed, confirm the parameter settings and click **OK**.

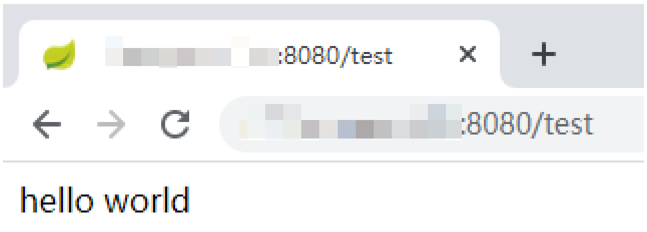
Wait until ✓ Successful is displayed on the page. If the deployment fails, rectify the fault based on the failed action information and error information in logs.

Step 8 View the deployment result.

Open a new browser page and enter the access address **http://IP:8080/test**. **IP** is the IP address of the ECS purchased in **Preparations**.

If the following information is displayed, the deployment is successful.

Figure 2-4 Deployment result



hello world

----End

Configuring a Pipeline

You can use a CodeArts pipeline to connect code check, build, and deployment tasks. When code changes, the pipeline is automatically triggered for continuous delivery.

Step 1 Choose **CICD** > **Pipeline** from the navigation bar. On the **Pipelines** tab, the automatically created pipeline **Web-Demo-pipeline** is displayed.

NOTE

This document uses the built-in pipeline associated with the code repository template.

In an actual development project, click **Create Pipeline** and create a pipeline based on the service scenario. For details, see **Creating a Pipeline**.

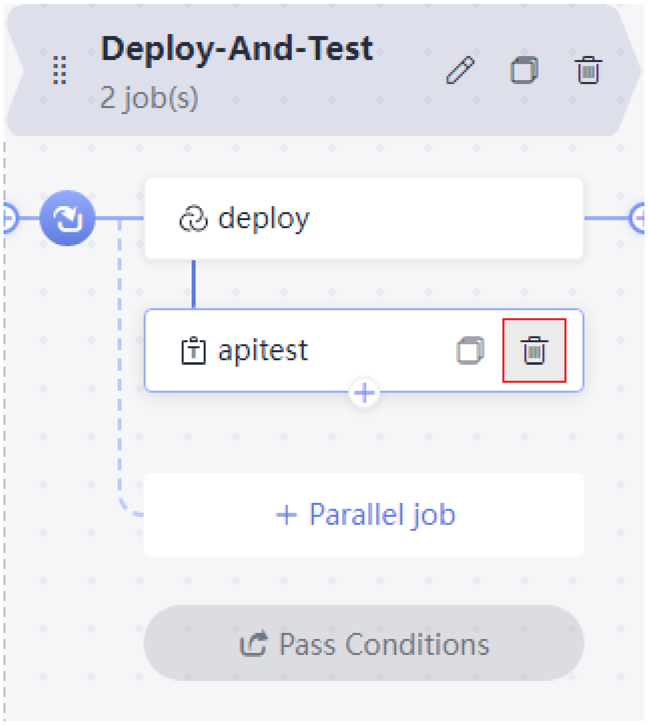
Step 2 Click ··· and choose **Edit** from the drop-down list.

Step 3 On the **Task Orchestration** tab page, configure the pipeline.

1. APITest is not involved in this document. Therefore, remove the API test task from the pipeline.

   Click 🗑 corresponding to **apitest**. In the dialog box that is displayed, click **OK**.

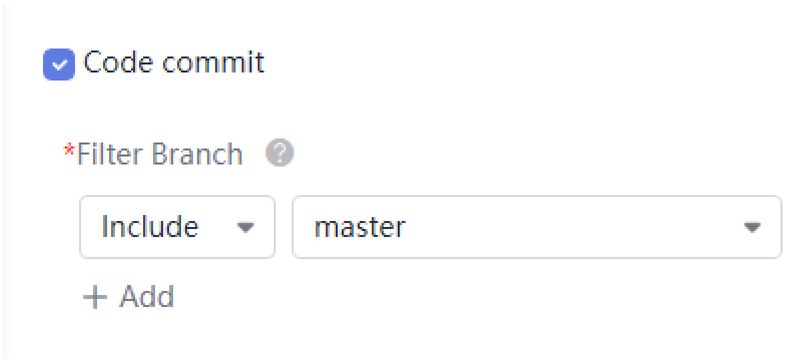
Figure 2-5 Deleting a job



2. Click the **codecheck** task.
3. Click the **deploy** job, associated the build task **cloudbuild**, and set other parameters based on the parameter settings in **Deploying the Build Package**.

Step 4 Click the **Execution Plan** tab, select **Code Commit**, select **master** from the branch filter drop-down list.

Figure 2-6 Configuring the execution plan


Code commit ✓

*Filter Branch

Include master

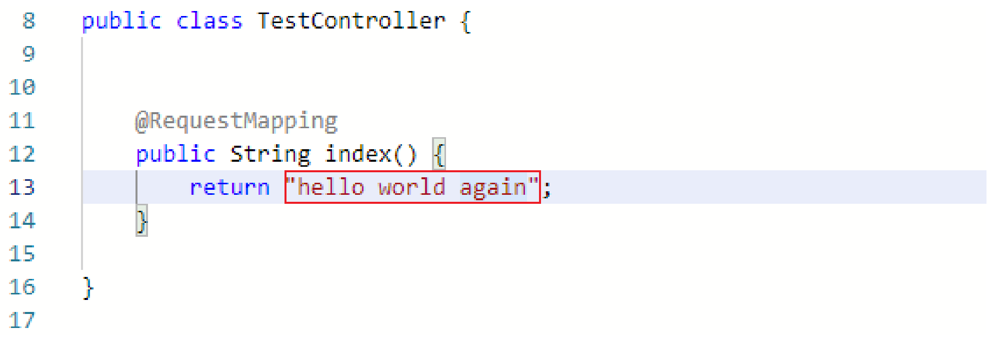
Step 5 Click **Save** to exit the editing mode.

Step 6 Go to **Deploy**, edit the deployment actions, and select **Enable this action** in **Stop Spring Boot**.

Step 7 Go to the code repository and search for and open the **TestController.java** file.

Click ✎, change **hello world** to **hello world again**, submit the information, and click **OK**.

Figure 2-7 Modifying code



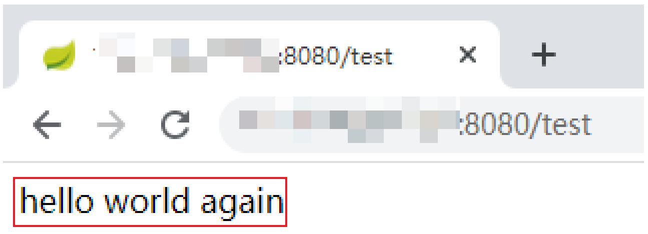
```
public class TestController {


    @RequestMapping
    public String index() {
        return "hello world again";
    }

}

```

Step 8 Return to the Pipeline page. You can see that the pipeline is running.

When ✓ is displayed, access **http://IP:8080/test** again. The following figure shows the access result.

If the task fails to be executed, check the failure cause. You can open the step details page to view the task logs and rectify the fault based on the logs.

Figure 2-8 Pipeline execution result



hello world again

----End

Releasing Resources

To avoid unnecessary fees, you can release resources after completing this example.

NOTICE

Released resources cannot be restored. Exercise caution when performing this operation.

Step 1 Delete a project.

1. Choose **Settings** > **General > Basic Information**.
2. Click **Delete Project**, in the displayed dialog box, enter the project name and click **Delete**.

Step 2 Delete the ECS.

1. Log in to the ECS console, locate the ECS to be deleted in the list, click **More**, and choose **Delete** from the drop-down list.

2. Select all options in the dialog box that is displayed and click **Yes**.

----End

3 Using CodeArts to Quickly Set Up a Project (CCE)

This section describes how to use the built-in code repository of CodeArts to develop, build, and deploy projects for continuous delivery.

The following describes deployment using CCE. For details about deployment using ECS, see **Using CodeArts to Quickly Set Up a Project (ECS)**.

Preparations

1. You have purchased CodeArts. For details, see **Purchasing CodeArts**.
2. You have **purchased a CCE cluster**. The cluster is configured according to **Table 3-1** and **Table 3-2**. Default values can be retained for configurations not listed in the tables.

Table 3-1 Cluster configurations

Category	Item	Suggestion
Basic Settings	Billing Mode	Select **Pay-per-use**.
	Cluster Version	You are advised to select the latest version.
Network Settings	Network Model	Select **VPC network**.
	VPC	Select a VPC. If no proper VPC is available in the list, click **Create VPC** to create one.
	Master Node Subnet	Select a subnet. If no proper subnet is available in the list, click **Create Subnet** to create one.
	Container CIDR Block	Select **Auto select**.

Table 3-2 Node configurations

Category	Item	Suggestion
Compute Settings	Billing Mode	Select **Pay-per-use**.
	Node Type	Select **Elastic Cloud Server (VM)**.
	Specifications	Select 2 vCPUs and 8 GB memory or higher.
	Container Engine	Select **Docker**.
	OS	Select **Public image** > **CentOS 7.6**.
	Login Mode	Select **Password**.
	Password	Enter a password.
Network Settings	Node IP	Select **Random**.
	EIP	Select **Do not use**.

3. You have **created an organization** in SoftWare Repository for Container (SWR). In this example, organization name **web-demo** is used as an example.

Creating a Project

A project is the basis for using services on CodeArts. Subsequent operations can be performed only after a project is created.

Step 1 **Log in to the CodeArts console**.

Step 2 Click and select a region.

Step 3 Click **Access Service**.

Step 4 Click **Create Project**.

Step 5 Select **Scrum**, enter the project name **Demo**, and click **OK**.

----End

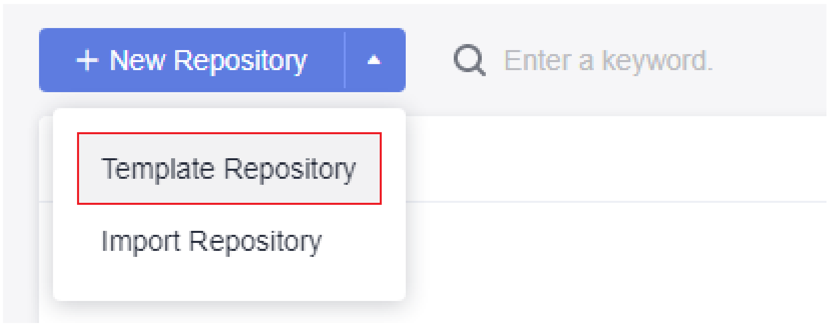
Creating a Code Repository

You can use a code repository to manage project code versions. This section describes how to use the built-in template **Java Web Demo** to create a code repository.

Step 1 On the navigation bar, choose **Code** > **Repo**.

Step 2 Click next to **New Repository** and select **Template Repository**.

Figure 3-1 Creating a code repository



Step 3 On the page displayed, select **Java Web Demo** and click **Next**.

Step 4 Enter the code repository name **Web-Demo** and click **OK**.

----End

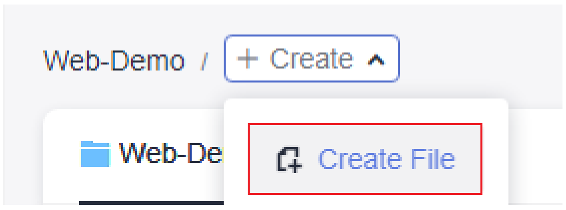
Preparing a Dockerfile

A Dockerfile is a text file that contains the instructions and descriptions required for building an image. For details about Dockerfile, see the **Docker official website**.

Step 1 Click a repository name to go to the repository.

Step 2 Click **Create** above the file list. Select **Create File** from the drop-down list.

Figure 3-2 Creating a file



Step 3 Enter the file name **Dockerfile** and then enter the following content:

```
FROM openjdk:8-alpine
ADD target /demo
COPY ./target/demoapp.jar /demo
CMD ["java","-jar","/demo/demoapp.jar"]
```

Step 4 Enter a commit message and click **OK**.

----End

Building and Pushing an Image

Use a build task to compile the software source code into an image and push and archive the image to SWR.

Step 1 In the navigation bar, choose **CICD** > **Build**.

Step 2 Click **Create Task** and configure task information.

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

Table 3-3 Basic information

Item	Suggestion
Task Name	Enter a custom name (**Web-Demo-docker** as an example).
Code Source	Select **Repo**.
Source Code Repository	Select **Web-Demo**.
Branch	Select **master**.

2. **Build Template**: Select the **Blank Template** and click **Next**.

Step 3 Configure build actions.

1. Click **Add Build Actions**, find **Build with Maven** in the list, and click **Add**.
2. Click **Add step**. In the step list, find **Build Image and Push to SWR**. Click **Add**.
3. Configure **Build Image and Push to SWR** by referring to the following table. (Retain the default values for the fields not listed in the table.)

Table 3-4 Configuring image information

Item	Suggestion
Organization	Enter the organization name **web-demo** created in **Preparations**.
Image Tag	v1.0.0

Step 4 After the configuration is complete, click **Create and Run**.

When ✓ is displayed, the task is successfully executed. If the build fails, rectify the fault based on the failed action information and error information in logs.

Step 5 Log in to the SWR console. In the navigation pane, choose **My Images**.

There is a record whose **Name** is **demo** and **Organization** is **web-demo**.

Click the image name to view details. The image version is **v1.0.0**.

----End

Creating a Workload

Create a Deployment on CCE to deploy and run the demo image.

Step 1 Log in to the CCE console and click the cluster purchased in **Preparations** to go to the details page.

Step 2 In the navigation bar, choose **Workloads**. Click **Create Workload**, complete the configuration by referring to the following table, and click **Create Workload**.

Table 3-5 Creating workload

Category	Item	Suggestion
Basic Info	Workload Type	Select **Deployment**.
	Workload Name	User-defined. In this example, enter **web-demo**.
	Pods	Enter **1**.
Container Settings	Image Name	Click **Select Image**. In the dialog box that is displayed, select **demo** and click **OK**.
	Pull Policy	Select **Always**.
	Image Tag	Select **v1.0.0**.
Advanced Settings	Upgrade Mode	Set **Upgrade Mode** to **Replace upgrade**.

Step 3 A message is displayed, indicating that the creation is successful. Click the **View Workload Details** to go back to the details page. There is one record in the **Pods**.

If the pod status is **Running**, click the **Access Mode** tab, click **Create**, configure the service by referring to the following table, and click **OK**.

If the instance status is abnormal, rectify the fault by referring to **Workload Abnormalities**.

Table 3-6 Configuring access mode

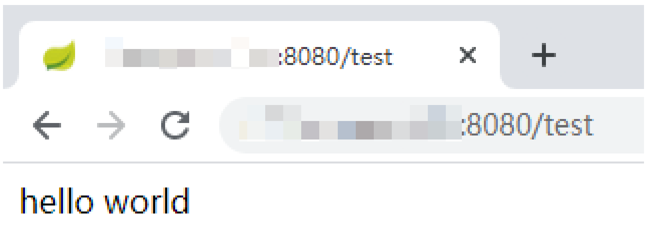
Item	Suggestion
Service Name	User-defined. In this example, enter **web-demo**.
Service Type	Select **LoadBalancer**.
Service Affinity	Select **Cluster-level**.
Load Balancer	Choose **Shared** > **Auto Create**. Then enter a load balancer name (use **web-demo-test** in this example) and check the box next to "I have read Notes on Using Load Balancers". **NOTE** If your account already has a load balancer, choose **Shared** > **Use existing** and select an existing load balancer.
Port	• Set **Protocol** to **TCP**. • Set **Container Port** to **8080**. • Set **Service Port** to **8080**.

Step 4 Move the cursor to the load balancer name in a displayed list under **Service Type** when ● web-demo is displayed next to a service name. Copy the **Public IP** displayed in the dialog box.

Step 5 Open a new browser page and enter **http://IP:8080/test** in the address box. Replace **IP** with the public network address copied in **Step 4**.

If the following information is displayed, the image has been deployed and started running.

Figure 3-3 Deployment result



[redacted]:8080/test

[redacted]:8080/test

```
hello world
```

----End

Deploying an image

You can create applications on Deploy to automatically deploy images.

Step 1 Return to the CodeArts page and choose **CICD** > **Deploy** in the navigation bar.

1. Click **Create Application**, enter the application name (**web-demo-k8s** as an example), and click **Next**.
2. Select the **Blank Template** and click **OK**.

Step 2 Search for and add Step **Kubernetes Quick Deployment (CCE Cluster)** to the step list. Configure the steps by referring to the following table.

Table 3-7 Configuring deployment actions

Item	Suggestion
Region	Select the region where the cluster located.
Cluster Name	Select the cluster purchased in **Preparations**.
Namespace	In this document, select **default**.
Workload	Select **web-demo**.
Container	Select a container name configured when **Create Workload** is selected.

Step 3 Click **Save & Deploy**.

If ✓ Successful is displayed, the test is successful. If the deployment fails, rectify the fault based on the failed action information and error information in logs.

----End

Configuring a Pipeline to Automatically Update Image Deployment

Configure a pipeline to integrate the code repository, build, and deployment. When a code commit action occurs in the code repository, the pipeline is automatically executed for continuous delivery.

Step 1 Choose **CICD** > **Pipeline** from the navigation bar.

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

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

Table 3-8 Pipeline basic information

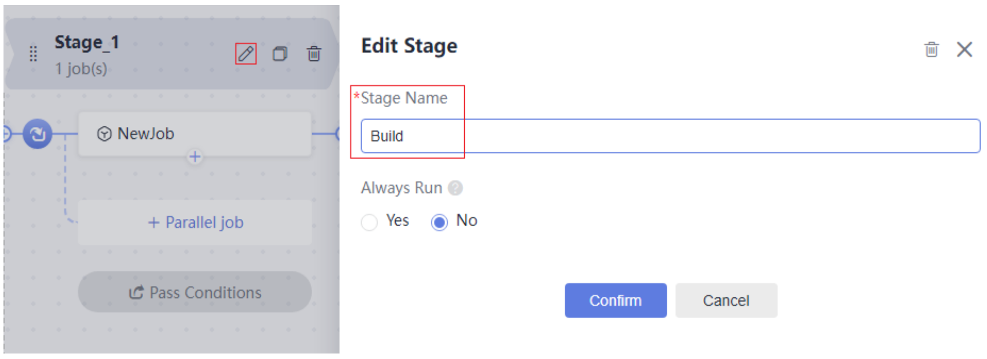
Item	Suggestion
Project	Enter **pipeline-web-demo**.
Pipeline Source	Select **Repo**.
Repository	Select **Web-Demo**.
Default Branch	Select **master**.

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

Step 3 Configure a workflow.

1. Click ✎ next to **Stage_1**. In the **Edit Stage** dialog box, enter the name **Build** and click **Confirm**.

Figure 3-4 Editing the stage name

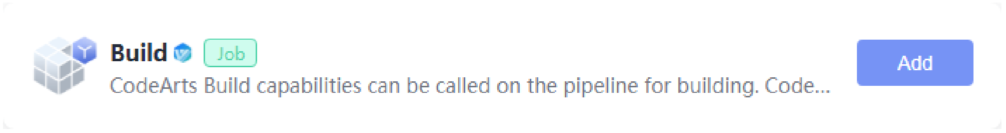


*Stage Name: Build

Always Run: Yes [✓ No]

2. Click **NewJob**.
   Click **Add** next to **Build** in the **NewJob** window.

Figure 3-5 Adding a job



CodeArts Build capabilities can be called on the pipeline for building. Code...

3. Configure task information by referring to the following table and click **OK**.

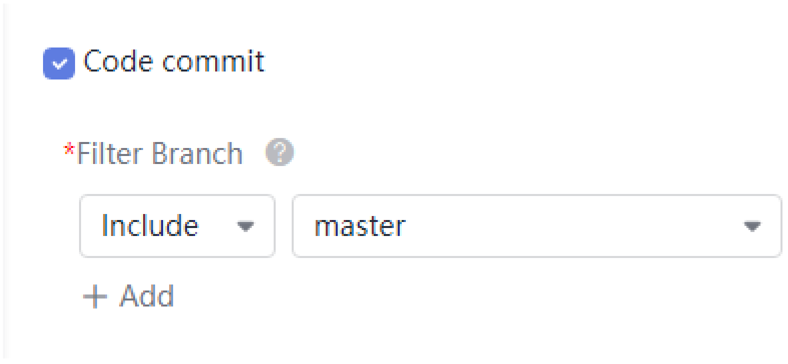
Table 3-9 Editing a build task

Item	Suggestion
Name	Enter a custom name (use the default value in this document).
Select Task	Select **Web-Demo-docker**.
Repository	Select **Web-Demo**.

4. Click **Stage** and change the stage name to **Deploy**.
5. Click **Job** and add the **Deploy** extension.
6. Select **web-demo-k8s** and select the job name configured in **Step 3.3**.

Step 4 Click the **Execution Plan** tab, select **Code commit**, select **master** from the branch filter drop-down list, and click **Save**.

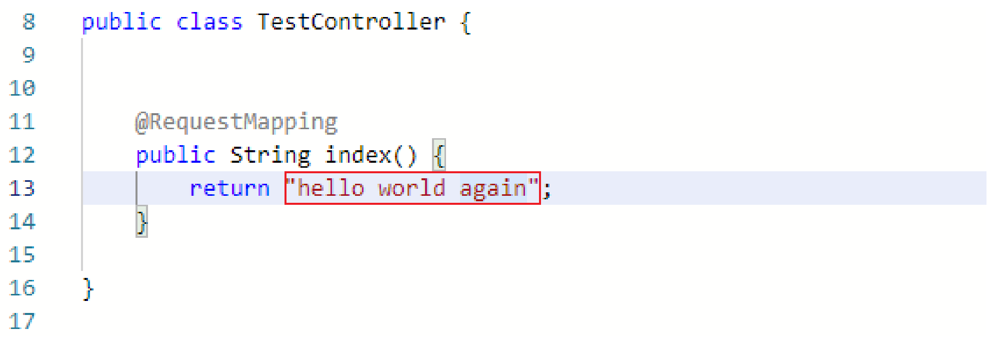
Figure 3-6 Configuring the execution plan



Step 5 Go to the code repository and search for and open the **TestController.java** file.

Click ✎, change **hello world** to **hello world again**, submit the information, and click **OK**.

Figure 3-7 Modifying code



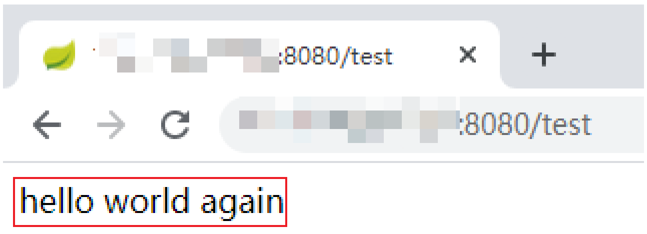
```
public class TestController {


    @RequestMapping
    public String index() {
        return "hello world again";
    }

}

```

Step 6 Return to the Pipeline page. You can see that the pipeline is running.

When ✓ is displayed, access **http://IP:8080/test** again. The following figure shows the access result.

If the task fails to be executed, check the failure cause. You can open the step details page to view the task logs and rectify the fault based on the logs.

Figure 3-8 Pipeline execution result



hello world again

----End

Releasing Resources

To avoid unnecessary fees, you can release resources after completing this example.

NOTICE

Released resources cannot be restored. Exercise caution when performing this operation.

Step 1 Delete a project.

1. Choose **Settings** > **General > Basic Information**.
2. Click **Delete Project**, in the displayed dialog box, enter the project name and click **Delete**.

Step 2 Delete the organization and the image.

1. Log in to the SWR console.
2. On the **My Images** page, select the image created in this example, and click **Delete**. In the displayed dialog box, click **Yes**.
3. On the **Organizations** page, click the name of the organization to be deleted. Click **Delete**. In the dialog box that is displayed, click **Yes**.

Step 3 Delete the cluster.

1. Log in to the CCE console. In the list, locate the cluster to be deleted and click 🗑.
2. Select all options in the dialog box that is displayed and click **Yes**.

----End