CodeArts

User Guide

Issue 01

Date 2026-01-12

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

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 Service Overview

1.1 What Is CodeArts?

CodeArts is a one-stop platform that provides out-of-the-box cloud services for requirement delivery, code commit, check, build, verification, deployment, and release throughout the entire software lifecycle.

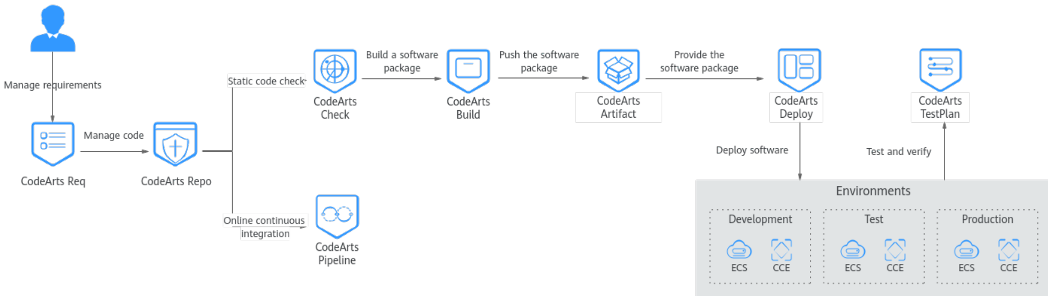
Architecture

CodeArts consists of the following services:

- CodeArts Req provides agile development teams with easy and efficient coordination services, including multi-project management, agile iteration management, Kanban collaboration, requirement management, defect tracing, document management, multi-layered billboard, and statistics reports.
- CodeArts Repo: Provides Git-based distributed code management and collaborative development capabilities, including member management, permission control, code hosting, code check, code review, code tracing, and continuous integration. It helps enterprises of different scales improve R&D quality and efficiency.
- CodeArts Pipeline: A pipeline service that enables you to visualize and customize your delivery, improving delivery efficiency and quality.
- CodeArts Check manages code quality in the cloud. You can easily perform static checks and security checks on code in multiple programming languages and obtain comprehensive quality reports. CodeArts Check also allows you to view grouped defects with fix suggestions provided, effectively controlling quality.
- CodeArts Build provides an easy-to-use code compile and build platform that supports multiple programming languages in the cloud. It helps enterprises improve build efficiency.
- CodeArts Deploy: Supports deployment in hosts and containers by using multiple languages and technology stacks, including Tomcat and Spring Boot. The plug-in encapsulation and orchestration of deployment functions help you quickly and efficiently release software.

- CodeArts TestPlan: Covers the entire process of test plan, test design, test cases, test execution, and test evaluation, and aims to help enterprises with collaborative, efficient, and trustworthy test activities before product release.
- CodeArts Artifact: Manages source code build products, including Maven and npm. CodeArts Artifact seamlessly interconnects with local build tools and on-cloud CI/CD so that you can manage software package lifecycle to improve release quality and efficiency. It also provides features such as artifact package version control, granular permission management, and more.

Figure 1-1 Architecture



1.2 Advantages

One-Stop Software Development Pipeline

- Full-lifecycle software development services, including CodeArts Req, CodeArts Repo, CodeArts Pipeline, CodeArts Check, CodeArts Build, CodeArts Deploy, CodeArts TestPlan, and CodeArts Artifact

Built-in R&D Security

- Security guidelines and protection capabilities from application design, development, test, and running to secure the application R&D supply chain

High-Quality, Efficient, and Agile Delivery

- Supports customization and automation of code check, build, test, and deployment tasks, and provides continuous delivery pipelines with visualized orchestration, one-click application deployment, and zero wait for release.

1.3 Application Scenarios

Internet Enterprises

- **Challenges**

  Facing rapid market changes and narrow product profit windows, enterprises cannot deliver high-quality products to customers in a timely manner because R&D tools cannot meet actual project requirements. In addition, the R&D capabilities of enterprises cannot be measured, and whether new projects can be delivered on time cannot be determined based on data.
- **Recommended Services**

CodeArts Req, CodeArts Repo, CodeArts Check, CodeArts Build, CodeArts Deploy, CodeArts TestPlan

- **Benefits**

  New functions and features are released at any time every day, shortening the period for feedback closed-loop management.

Software and Solution Carriers

- **Challenges**

  For software and solution enterprises, communication is difficult among developers who work in different places and use different tools and environments. Customer requirements change fast, which requires fast responses and often causes rework. Enterprises urgently need tools for automated, continuous integration.

- **Recommended Services**

  CodeArts Req, CodeArts Repo, CodeArts Check, CodeArts Build, CodeArts Deploy, CodeArts TestPlan, CodeArts Artifact

- **Benefits**

  Efficient collaboration among developers and controllable project development period allows you to quickly respond to customer requirements.

1.4 Billing

Billing Mode

CodeArts uses pay-per-use billing. You are charged by duration from the time when the pay-per-use service is enabled to the time when the service is disabled.

Billing Items

The billing items of CodeArts include the number of members and resource usage.

- Members
  - To view the total number of members in all projects of a tenant after deduplication, choose **All Account Settings** > **General** > **Projects and Members** and click the **Project Members** tab.
  - The charging unit is person-hour.
- Resource usage
  - **Table 1-1** lists the billing items of each individual CodeArts service.
  - APITest needs to be enabled separately and is billed based on usage.

Table 1-1 Resource usage billing items

Service	Billing Item
CodeArts Req	Storage space
CodeArts Repo	Storage space

Service	Billing Item
CodeArts Build	Build duration
CodeArts TestPlan (test management)	Storage space
CodeArts Artifact	Storage space

Fee Calculation Method

After the pay-per-use mode is enabled, the total fee of CodeArts is calculated as follows: Fees generated by project members + Fees generated by the resource usage in each service.

Free Quota

After you enable the pay-per-use package of CodeArts for the first time, you will be granted a free quota.

The resources included in this free quota are listed in the following table. You will be billed for any excess usage.

Table 1-2 Free quota

Service	Billing Item	Free Quota
-	Members	5
CodeArts TestPlan (API test)	API test duration	30 min

2 Getting Started

2.1 Setting Up an ECS-based Code Development Pipeline

This section describes how to use the built-in code repository of CodeArts to develop, build, and deploy an application.

The deployment is based on Elastic Cloud Server (ECS). This approach is suitable for traditional software packages.

To use container-based deployment, see **Setting Up a CCE-based Code Development Pipeline**.

Procedure

Step	Description
Preparations	Create an ECS.
Step 1: Enable the Pay-per-Use Package	CodeArts is billed based on usage. Enable the pay-per-use package before using CodeArts.
Step 2: Create a Project	Projects are foundational for using CodeArts services. Create a project first before proceeding with subsequent operations.
Step 3: Create a Code Repository	Use CodeArts Repo to create a repository from the built-in template **Java Web Demo**. CodeArts Repo helps you manage your project code by version.
Step 4: Check Code	Use CodeArts Check to examine your code statically to control quality.
Step 5: Build and Archive the Software Package	Use CodeArts Build to compile your source code into object files, package them together with configuration and resource files, and then archive them to the release repository.

Step	Description
Step 6: Deploy the Build Package	Use CodeArts Deploy to deploy software packages in the release repository to a VM and then run the application.
Step 7: Configure a Pipeline	Use CodeArts Pipeline to link code check, build, and deployment tasks for continuous delivery. It runs automatically when you update your code.
Releasing Resources	After completing this practice, delete unused pay-per-use resources to prevent extra charges.

Preparations

- Purchase an ECS that meets the following requirements by referring to the "Purchasing an ECS" chapter in the *Elastic Cloud Server User Guide*.
  - **Billing Mode**: Select **Pay-per-use**.
  - **CPU Architecture**: Select **x86**.
  - **Specifications**: 2 vCPUs | 4 GiB or above, system disk ≥ 80 GiB
  - **OS**: **Public image** > **CentOS 7.6**
  - **EIP**: **Auto assign**

  After completing the purchase, add two inbound rules by referring to the "Configuring Security Group Rules" chapter in the *Elastic Cloud Server User Guide*.
  - Protocol: **TCP**; port: **22**; source: **0.0.0.0/0**
  - Protocol: **TCP**; port: **8080**; source: **0.0.0.0/0**

Step 1: Enable the Pay-per-Use Package

Step 1 Log in to the console, view the **Overview** page, and click **Pay-per-Use Package**.

Step 2 View the service specifications and click **Next**.

Step 3 View and confirm the package details. Click **Submit**. In the dialog box that is displayed, click **Yes**.

Step 4 Check whether the system displays a message indicating that CodeArts is subscribed to successfully.

Return to the **Overview** page. On the **My Services** tab, the status shows enabled **Subscription**.

----End

Step 2: Create a Project

Step 1 Click **go to Workspace** on the CodeArts console.

Step 2 On the CodeArts homepage, find the **Scrum** template and click **Select**.

Step 3 Enter a project name (for example, **Demo**), and click **OK**. Keep the name under 128 characters.

The project is created, and the **Work Items** page is displayed.

----End

Step 3: Create a Code Repository

Step 1 In the project, choose **Code** > **Repo** from the navigation pane.

Step 2 Click **Create Repository**.

Step 3 Select **Template** and click **Next**.

Step 4 Select the **Java Web Demo** template and click **Next**.

Step 5 Enter a repository name (for example, **Web-Demo**), and click **OK**. Start the name with a letter, digit, or underscore (_), and use letters, digits, hyphens (-), underscores (_), and periods (.). Do not end the name with **.git**, **.atom**, or a period.

The repository is created, and its code file list is displayed.

----End

Step 4: Check Code

Step 1 In the navigation pane, choose **Code** > **Check**.

The automatically created task **Web-Demo-check** is displayed.

Step 2 Click and choose **Settings**.

Step 3 Click **Rule Sets** in the navigation pane and retain the **JAVASCRIPT** and **JAVA** rules. Disable other rules.

Step 4 Click **Start Check** to start the task.

If Success is displayed, the task is successful.

If the task fails, rectify the fault by referring to CodeArts Check FAQs.

Step 5 Click the **Issues** tab to view the issue list.

Since there are no critical or major issues, you do not need to modify the code.

----End

Step 5: Build and Archive the Software Package

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

The automatically created build task **Web-Demo-build** is displayed.

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

If is displayed, the task is successful.

If the task fails, rectify the fault based on the failed action and the error message in logs. For details, see CodeArts Build FAQs.

Step 3 Click the task name to view its details. On the **Build History** tab, find the latest build ID and record it.

Step 4 In the navigation pane, choose **Artifact** > **Release Repos**.

In the repository view, find the repository with the same name as your project, and go to the **Web-Demo-build** > *Build ID recorded in* ***Step 3*** folder to find the generated software package **demoapp.jar**.

----End

Step 6: Deploy the Build Package

Step 1 Configure the target host.

1. In the navigation pane, choose **Settings** > **General** > **Basic Resources**.
2. Click **Create Host Cluster**, configure the following information, and click **Save**.

Table 2-1 Creating a host cluster

Parameter	Example	Description
Cluster Name	host-group	The name of the host cluster to create. Enter 3 to 128 characters, including letters, digits, hyphens (-), underscores (_), and periods (.).
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**.
Execution Resource Pool	Official	A resource pool (or agent pool) is a collection of physical environments where commands are executed to deploy software packages. Choose the official agent pool or a self-hosted agent pool that contains your own servers.

3. After the system displays a message indicating that the host cluster is created, click **Add Host** on the **Target Hosts** tab, select **Importing ECS**, locate the ECS purchased in **Preparations**, and click **Import**.
4. Configure the following information and click **OK**.

Table 2-2 Adding a host

Parameter	Example	Description
Username	root	The username for logging in to the ECS. By default, it is **root** for a Linux ECS.
Password	Enter the password set when you purchase the ECS in **Preparations**.	The password for logging in to the ECS.
SSH Port	22	The default port is **22**. You can also use another one.

5. Check the host record. If the **Verification Result** column shows successful, the host is added.

   If the host fails to be added, rectify the fault based on the failure details. For details, see CodeArts Deploy FAQs.

Step 2 Choose **CICD** > **Deploy** from the navigation pane.

The automatically created application **Web-Demo-deploy** is displayed.

Step 3 Click ··· and choose **Edit**.

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

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

Table 2-3 Creating an environment

Parameter	Example	Description
Environment	demo-env	The name of the environment to create. Enter 3 to 128 characters, including letters, digits, hyphens (-), underscores (_), and periods (.).
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. When the system displays a message indicating that the creation is successful, click **Import Host** on the **Resources** tab. In the displayed dialog box, select the host cluster and host configured in **Step 1** and click **Import**.
3. When the system shows a success message for the import, close the window.

Step 5 Click the **Deployment Actions** tab and configure actions.

- **Install JDK**: Check that the JDK version is **openjdk-1.8.0**.
- **Select a Deployment Source**: Set the parameters based on the following table. Retain the default values for other parameters.

Table 2-4 Deployment source configuration

Parameter	Example	Description
Source	Build task	The source of the software package to deploy. Select **Artifact** or **Build task**.
Build Task	Web-Demo-build	Available only when **Source** is set to **Build task**.
Download Path	/usr/local/${package_name}/	The path on your target host for saving the software package.

- **Start/Stop SpringBoot**: When you run deployments for the first time, this action will fail because Spring Boot has not yet run on the target host. Disable this action by clicking ··· on the action card and choosing **Disable**.
- **Start SpringBoot**: Retain the default settings.
- **Health Test Through URLs**: This action is optional. Disable it for this example.

Step 6 Click the **Parameters** tab and set parameters by referring to the following table.

Table 2-5 Configuration parameters

Name	Default Value
host_group	The environment **demo-env** added in **Step 4**
package_url	Not required for this example. Click in the same row to delete it.
service_port	**8080**
package_name	**demoapp**

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

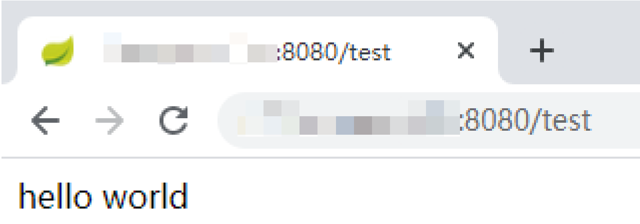
Wait until Successful is displayed on the page. If the deployment fails, rectify the fault based on the failure step and the error message in logs. For details, see CodeArts Deploy FAQs.

Step 8 View the deployment result.

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

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

Figure 2-1 Deployment result



----End

Step 7: Configure a Pipeline

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

On the **Pipelines** tab, the automatically created pipeline **Web-Demo-pipeline** is displayed.

Step 2 Click ··· and choose **Edit**.

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

1. API testing is not involved in this example. So remove the API test task from the pipeline.

   Click 🗑 next to the **Apitest** job. In the displayed dialog box, click **OK**.

2. Click the **Deploy** job, select the build task **Build**, and keep other parameters the same as those set in **Step 6: Deploy the Build Package**.

Step 4 Click the **Trigger Settings** tab, select **Code commit**, and select **master** from the branch filter drop-down list.

Step 5 Click **Save**.

The updated triggering settings is displayed.

Step 6 Go to **Deploy**, edit the deployment actions, and enable **Stop SpringBoot**.

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

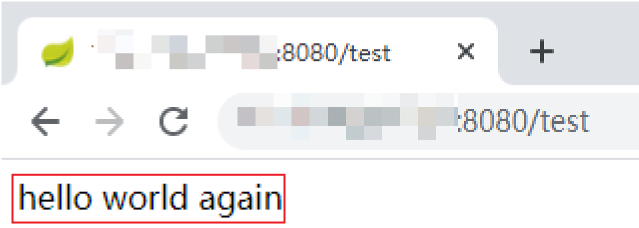
Click ✎, change **hello world** to **hello world again**, enter a commit message, and click **OK**.

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 pipeline fails, click the cause to view logs. Then rectify the fault by referring to CodeArts Pipeline FAQs.

Figure 2-2 Pipeline execution result



----End

Releasing Resources

⚠ WARNING

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

In this example, the pay-per-use resources involved are from ECS.

If you do not need the ECS after the trial, you are advised to release the ECS by referring to "How Do I Delete or Restart an ECS?" in *Elastic Cloud Service FAQs*.

2.2 Setting Up a CCE-based Code Development Pipeline

This section describes how to use the built-in code repository of CodeArts to develop, build, and deploy an application.

This chapter uses Cloud Container Engine (CCE) for container-based deployment. To use traditional software package deployment, see **Setting Up an ECS-based Code Development Pipeline**.

Procedure

Step	Description
Preparations	Buy a Cloud Container Engine (CCE) cluster and create a SoftWare Repository for Container (SWR) organization.
Step 1: Enable the Pay-per-Use Package	CodeArts is billed based on usage. Enable the pay-per-use package before using CodeArts.
Step 2: Create a Project	Projects are foundational for using CodeArts services. Create a project first before proceeding with subsequent operations.
Step 3: Create a Code Repository	Use CodeArts Repo to create a repository from the built-in template **Java Web Demo**. CodeArts Repo helps you manage your project code by version.

Step	Description
Step 4: Prepare a Dockerfile	Create a Dockerfile. It 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 5: Build an Image and Push It to SWR	Run a build task to compile the software source code into an image and archive the image to SWR.
Step 6: Create a Workload	Create a Deployment in CCE to load and run the demo image.
Step 7: Deploy the Image	Create an application in CodeArts Deploy to automate image deployment.
Step 8: Configure a Pipeline	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.
Releasing Resources	After completing this practice, delete unused pay-per-use resources to prevent extra charges.

Preparations

- Buy a CCE cluster that meets the requirements listed in the following table.

Table 2-6 Cluster configuration requirements

Category	Configuration	Reference
Cluster	**Pay-per-use** recommended • **Type**: **CCE Standard Cluster** • **Cluster Version**: Select the latest version. • **Controller Node Architecture**: **X86** • **Network Model**: **VPC network** • **Container CIDR Block**: **Auto select**	"Buying a CCE Standard/Turbo Cluster" chapter in the *Cloud Container Engine User Guide*
Node	**Pay-per-use** recommended • **Node Type**: **Elastic Cloud Server (VM)** • **Specifications**: **2 vCPUs | 8 GiB** or above • **Container Engine**: **Docker** • **OS**: **Public image** > **CentOS 7.6**	"Creating a Node" chapter in the *Cloud Container Engine User Guide*

- You have created an organization named **web-demo** in SWR. If the system displays a message indicating that the organization already exists, customize another name. For details, see "Creating an Organization" in the *Software Repository for Container User Guide*.

Step 1: Enable the Pay-per-Use Package

Step 1 Log in to the console, view the **Overview** page, and click **Pay-per-Use Package**.

Step 2 View the service specifications and click **Next**.

Step 3 View and confirm the package details. Click **Submit**. In the dialog box that is displayed, click **Yes**.

Step 4 Check whether the system displays a message indicating that CodeArts is subscribed to successfully.

Return to the **Overview** page. On the **My Services** tab, the status shows enabled **Subscription**.

----End

Step 2: Create a Project

Step 1 Click **go to Workspace** on the CodeArts console.

Step 2 On the CodeArts homepage, find the **Scrum** template and click **Select**.

Step 3 Enter a project name (for example, **Demo**), and click **OK**. Keep the name under 128 characters.

The project is created, and the **Work Items** page is displayed.

----End

Step 3: Create a Code Repository

Step 1 In the project, choose **Code** > **Repo** from the navigation pane.

Step 2 Click **Create Repository**.

Step 3 Select **Template** and click **Next**.

Step 4 Select the **Java Web Demo** template and click **Next**.

Step 5 Enter a repository name (for example, **Web-Demo**), and click **OK**. Start the name with a letter, digit, or underscore (_), and use letters, digits, hyphens (-), underscores (_), and periods (.). Do not end the name with **.git**, **.atom**, or a period.

The repository is created, and its code file list is displayed.

----End

Step 4: Prepare a Dockerfile

Step 1 In CodeArts Repo, click **Create** above the file list. Select **Create File** from the drop-down list.

Step 2 Enter the file name **Dockerfile** and then enter the following code:

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

Step 3 Enter a commit message and click **OK**.

----End

Step 5: Build an Image and Push It to SWR

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

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

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

Table 2-7 Basic information

Parameter	Example	Description
Name	Web-Demo-docker	Build task name. Use a maximum of 115 characters, including letters, digits, underscores (_), and hyphens (-).
Code Source	Repo	Select **Repo**, **GitHub**, or other sources.
Repository	Web-Demo	The code repository to compile.
Default Branch	master	The repository branch to compile.

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

Step 3 Configure build actions.

1. Click **Add Build Actions**, find **Build with Maven** in the list, and click **Add**.
2. Click **Add Action**. In the action list, find **Build Image and Push to SWR**, and 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 this table.)

Table 2-8 Configuring image information

Parameter	Example	Description
Organization	Name of the organization created in **Preparations**	The organization to which the image will belong after being pushed to SWR.

Parameter	Example	Description
Image Tag	v1.0.0	Image version. Use a maximum of 128 characters, including letters, digits, periods (.), underscores (_), and hyphens (-). Do not start with a period or hyphen.

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

If is displayed, the task is successful. If the task fails, rectify the fault based on the failed action and the error message in logs. For details, see CodeArts Build FAQs.

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

Step 6: Create a Workload

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

Step 2 Choose **Workloads** in the navigation pane, and click **Create Workload**.

Step 3 Complete the configurations by referring to the following table and click **Create Workload**.

For details about the parameters, see "Creating a Deployment" chapter in the *Cloud Container Engine User Guide*.

Table 2-9 Creating workload

Category	Parameter	Example
Basic Info	Workload Type	Deployment
	Workload Name	web-demo
	Pods	1
Container Settings	Image Name	Click **Select Image**. In the dialog box that is displayed, select **demo** and click **OK**.
	Pull Policy	Always
	Image Tag	v1.0.0

Category	Parameter	Example
Service Settings	Service Name	web-demo
	Service Type	LoadBalancer
	Service Affinity	Cluster level
	Load Balancer	1. Choose **Shared** > **Auto create**. 2. Configure the following parameters: – **Instance Name**: **web-demo-test** – **EIP**: **Auto create**
	Port	• **Protocol**: **TCP** • **Container Port**: **8080** • **Service Port**: **8080**
Advanced Settings	Upgrade Mode	Replace upgrade

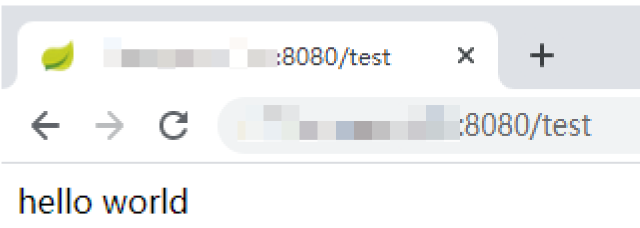
Step 4 When the creation is complete, click **View Workload Details** to go back to the details page. A record is displayed on the **Pods** tab.

When the status of the record is **Running**, click the **Access Mode** tab, find web-demo-test, move the cursor to the load balancer name under the access type, and copy the public network address in the displayed dialog box.

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

If the following information is displayed, the workload is running properly.

Figure 2-3 Deployment result



----End

Step 7: Deploy the Image

Step 1 Return to the CodeArts page, and choose **CICD** > **Deploy** from the navigation pane.

1. Click **Create Application**, enter an application name (for example, **web-demo-k8s**), and click **Next**. The name can contain 3–128 characters, including letters, digits, hyphens (-), and underscores (_).
2. Select **Blank Template** and click **OK**.

Step 2 Search for and add action **Kubernetes Quick Deployment (CCE cluster)**. Configure this action by referring to the following table.

Table 2-10 Configuring deployment actions

Parameter	Example	Description
Region	The region where the cluster purchased in **Preparations** is located	The region of the target cluster.
Cluster Name	The name of the cluster purchased in **Preparations**.	The name of the target cluster.
Namespace	default	The namespace of the target cluster.
Workload	web-demo	The workload to deploy.
Container	The container name displayed in the **Container Settings** area involved in **Step 6: Create a Workload**	The name of the container to deploy the workload in.

Step 3 Click **Save & Deploy**.

If  is displayed, the test 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.

----End

Step 8: Configure a Pipeline

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

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

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

Table 2-11 Pipeline basic information

Parameter	Example	Description
Name	web-demo-pipeline-k8s	Pipeline name. Use a maximum of 128 characters, including letters, digits, hyphens (-), and underscores (_).
Pipeline Source	Repo	The code source of the pipeline. Select **Repo**, **Git**, or other sources.
Repository	Web-Demo	The code repository to be associated with the pipeline.
Default Branch	master	The repository branch to be associated with the pipeline.

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

Step 3 Configure the workflow.

1. Click ✎ next to **Stage_1**. In the **Edit Stage** dialog box, enter a stage name and click **Confirm**. The name can contain 1–128 characters, including letters, digits, spaces, and special characters (-_,;:./()). Do not start or end with a space.
2. Click **Create Task**. If a drop-down list is available, select **From Scratch**.
3. Locate **Build** in the right window and click **Add**.
4. Configure the job information by referring to the following table and click **OK**.

Table 2-12 Editing a build job

Parameter	Example	Description
Name	Retain the default value.	Job name. Use 1–128 characters, including letters, digits, spaces, and special characters (-_,;:./()). Do not start or end the name with a space.
Select Task	Web-Demo-docker	Select a build task whose code source is pipeline or the same repository as the pipeline you are configuring.
Repository	Web-Demo	Select the code repository associated with the build task.

5. Click **Stage** and change the stage name to **Deploy**. The new stage is displayed.
6. Click **New Job** and add the **Deploy** extension.
7. Select the **web-demo-k8s** task (or application), associate the build task set in **Step 3.4**, and click **OK**.

Step 4 Click the **Trigger Settings** tab, select **Code commit**, and select **master** from the branch filter drop-down list.

Step 5 Click **Save**.

The updated triggering settings is displayed.

Step 6 Go to the code repository and search for and open the **TestController.java** file.

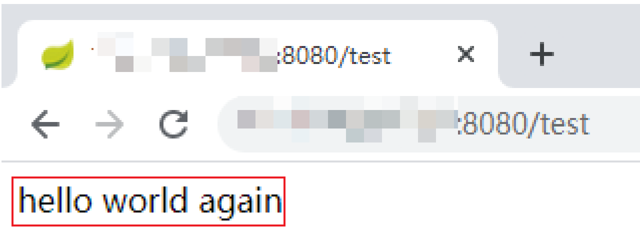
Click , change **hello world** to **hello world again**, enter a commit message, and click **OK**.

Step 7 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 pipeline fails, click the cause to view logs. Then rectify the fault by referring to CodeArts Pipeline FAQs.

Figure 2-4 Pipeline execution result



----End

Releasing Resources

WARNING

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

In this example, the pay-per-use resources involved are from CCE.

If you do not need the CCE after the trial, delete the cluster by referring to chapter "Deleting a Cluster" in the *Cloud Container Engine User Guide* to release its resources.

3 User Guide

3.1 Purchasing CodeArts

Pay-per-Use Package

Step 1 Log in to the console, view the **Overview** page, and click **Pay-per-Use Package**.

Step 2 View the service specifications and click **Next**.

Step 3 View and confirm the package details. Click **Submit**. In the dialog box that is displayed, click **Yes**.

Step 4 Check whether the system displays a message indicating that CodeArts is subscribed to successfully.

Return to the **Overview** page. On the **My Services** tab, the status shows enabled **Subscription**.

----End

NOTE

APITest is not included in the pay-per-use package. If you need to use APITest, **subscribe to it**.

Unsubscribing from CodeArts

Step 1 Log in to the console. On the **Overview** page, click the **My Services** tab.

Step 2 View the subscribed CodeArts package displayed on the page. Click .

Step 3 In the dialog box that is displayed, enter **UNSUBSCRIBE** and click **Yes**.

Step 4 Check whether the system displays a message **Pay-per-use package unsubscribed successfully**, the page displays a disabled **Subscription**.

----End

Subscribing to API Test

Before subscribing to APITest, **enable the pay-per-use package**.

Step 1 Log in to the console, choose **TestPlan**, and click **Pay-per-Use Package**.

Step 2 View and confirm the package details. Click **Subscribe Now**. In the dialog box that is displayed, click **Yes**.

Step 3 Check whether the system displays a message indicating that the service is subscribed to successfully.

Return to the **Overview** page. On the **My Services** tab, the status of CodeArts TestPlan (API Test) is subscribed to.

----End

Unsubscribing to API Test

Step 1 Log in to the console. On the **Overview** page, click the **My Services** tab.

Step 2 View subscribed API Test displayed on the page. Click .

Step 3 In the dialog box that is displayed, determine whether to release data.

- To release the data, enter **RELEASE DATA** and click **Yes**.
- If you do not want to release data, click **No**.

Step 4 When the system displays a message indicating that CodeArts TestPlan (APITest) has been unsubscribed successfully, the service package is displayed as **disabled**.

----End

3.2 CodeArts Console

3.2.1 Console Permission Management

3.2.1.1 CodeArts Console Permission Description

If you need to assign different permissions to employees in your enterprise to access your purchased CodeArts resources, Identity and Access Management (IAM) is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you secure access to your resources.

With IAM, you can use your account to create IAM users, and assign permissions to the users to control their access to specific resources. For example, some software developers in your enterprise need to use CodeArts resources but should not be allowed to perform any other high-risk operations, such as buying CodeArts resources. In this scenario, you can create IAM users for the software developers and grant them only the permissions required for viewing CodeArts resources.

If you do not require individual IAM users, skip this chapter.

CodeArts Console Permissions

By default, new IAM users do not have any permissions. You need to add them to one or more groups, and then add permissions policies or roles to these groups. The users inherit permissions from their groups and can then perform specified operations on cloud services.

CodeArts is a project-level service deployed and accessed in specific physical regions. To assign permissions to a user group, specify the scope as region-specific projects and select projects for the permissions to take effect. If **All projects** is selected, the permissions will take effect for the user group in all region-specific projects. When accessing CodeArts, the users need to switch to a region where they have been authorized.

CodeArts uses policies for fine-grained authorization.

- Policies: a type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible authorization. Policies allow you to meet requirements for more secure access control. For example, you can grant CodeArts users only the permissions for performing specific operations on the CodeArts console.

Table 3-1 lists all system permissions of CodeArts.

Table 3-1 System permissions of CodeArts

Policy	Description	Type
DevCloud Console FullAccess	Full permissions for the CodeArts console. Users with these permissions can buy CodeArts resources.	System-defined policy
DevCloud Console ReadOnlyAccess	Full permissions for the CodeArts console. Users with these permissions can only view the usage of CodeArts resources.	System-defined policy

Table 3-2 lists the common operations supported by each system-defined policy of CodeArts. Select a proper policy as required.

Table 3-2 Common operations and system permissions

Console Operation	DevCloud Console FullAccess	DevCloud Console ReadOnlyAccess
Check CodeArts Req resource usage	√	√
View CodeArts Req subscription records	√	√

Console Operation	DevCloud Console FullAccess	DevCloud Console ReadOnlyAccess
View CodeArts Req resources	√	√
Check CodeArts Repo resource usage	√	√
View CodeArts Repo subscription records	√	√
View CodeArts Repo resources	√	√
Check CodeArts Check resource usage	√	√
View CodeArts Check subscription records	√	√
View CodeArts Check resources	√	√
Check CodeArts Build resource usage	√	√
View CodeArts Build subscription records	√	√
View CodeArts Build resources	√	√
Check CodeArts TestPlan – Test Management resource usage	√	√
View CodeArts TestPlan – Test Management subscription records	√	√
View CodeArts TestPlan – Test Management resources	√	√
Subscribe to CodeArts TestPlan – APITest with pay-per-use billing	√	×
Unsubscribe from CodeArts TestPlan – APITest with pay-per-use billing	√	×
View CodeArts TestPlan – APITest subscription records	√	√

Console Operation	DevCloud Console FullAccess	DevCloud Console ReadOnlyAccess
View CodeArts TestPlan – APITest resources	√	√
Check CodeArts TestPlan – APITest resource usage	√	√
Check CodeArts Artifact resource usage	√	√
View CodeArts Artifact subscription records	√	√
View CodeArts Artifact resources	√	√
Authorize an enterprise account	√	×
Cancel the authorization granted to an enterprise account	√	×
Accept or reject authorization to an enterprise account	√	×
View the authorization list	√	√
Subscribe to a pay-per-use package	√	×
Unsubscribe from a pay-per-use package	√	×
View pay-per-use package subscription records	√	√

The service names in permission policies may differ from those on the console. The mapping relationship between these names is shown in **Table 3-3**.

Table 3-3 Service name mapping

Service Name in Permission Policies	Service Name on the Console
DevCloud	CodeArts
ProjectMan	CodeArts Req
CodeHub	CodeArts Repo

Service Name in Permission Policies	Service Name on the Console
CodeCheck	CodeArts Check
CloudBuild	CodeArts Build
CloudDeploy	CodeArts Deploy
CloudArtifact	CodeArts Artifact
CloudTest	CodeArts TestPlan
CloudPipeline	CodeArts Pipeline

3.2.1.2 Creating a User and Granting CodeArts Permissions

You can use Identity and Access Management (IAM) to implement fine-grained permissions control for your CodeArts console. With IAM, you can:

- Use your account to create IAM users for employees based on the organizational structure of your enterprise. Each IAM user has their own security credentials for accessing CodeArts resources.
- Assign only the permissions required for users to perform a specific task.

If your account does not require individual IAM users, skip this section.

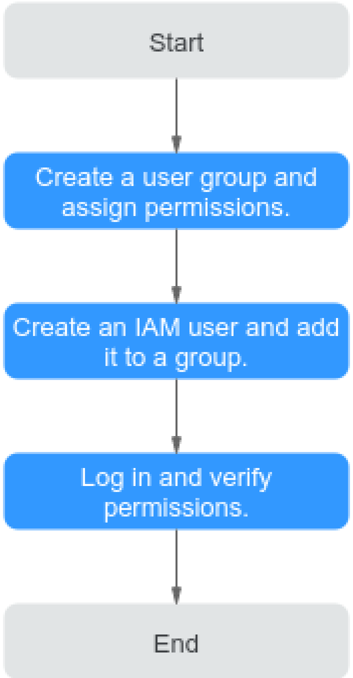
This section describes the procedure for granting permissions (see **Figure 3-1**).

Prerequisites

Learn about the permissions supported by CodeArts and choose permissions according to your requirements.

Process

Figure 3-1 Process for granting CodeArts console permissions



1. Create a user group and assign permissions.

   Create a user group on the IAM console, and attach the **DevCloud Console ReadOnlyAccess** policy to the group.

2. Create an IAM user.

   Create a user on the IAM console and add the user to the group created in **1**.

3. Logging In as a User and Verifying Permissions

   Log in to the CloudPond console with the credentials of the user created in the previous step and verify that the user can only view the resources.

   Choose **CodeArts** in **Service List**, and then choose **Enterprise Account Authorization** in the navigation pane. If a message is displayed indicating that you have insufficient permissions to access the page, the **DevCloud Console ReadOnlyAccess** policy has already taken effect.

3.2.1.3 CodeArts Custom Policies

You can customize policies to supplement system-defined policies of CodeArts.

You can create custom policies using one of the following methods:

- Visual editor: Select cloud services, actions, resources, and request conditions. You do not need to have knowledge of the policy syntax.
- JSON: Create a policy in the JSON format from scratch or based on an existing policy.

This section provides examples of custom policies on the CodeArts console.

CodeArts Custom Policy Examples

- Example 1: Authorize users to purchase CodeArts on the console

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "
                devcloud:monthlyPackage:subscribe
             "
         ]
      }
   ]
}
```

- Example 2: Define actions for multiple services in a policy

  A custom policy can contain the actions of multiple services that are of the global or project-level type. The following is an example policy containing actions of multiple services:

```
{
        "Version": "1.1",
        "Statement": [
                {
                        "Action": [
                                "devcloud:*:listResourceDetail",
                                "ecs:cloudServers:delete"
                        ],
                        "Effect": "Allow"
                }
        ]
}
```

3.2.2 Enterprise Account Authorization

Authorize other enterprise accounts to use the resources in your enterprise account. You can grant and cancel the authorization as required.

An enterprise account is essentially a tenant account, and users in the account are referred to as IAM users.

Authorizing an Enterprise Account

Step 1 Log in to the console.

Step 2 Choose **Enterprise Account Authorization** in the navigation pane on the left, and click **Authorize Enterprise Account**.

Step 3 Enter the ID of the target enterprise account, and click **Authorize**.

NOTE

- Obtain the enterprise account ID from the **My Credential** page of the enterprise to be invited.
- The authorized account must be from the same site as your account.

Step 4 View the new record after the enterprise account is successfully authorized.

----End

Accepting Authorization of Another Enterprise Account

If The current enterprise account can automatically accept authorization. To enable manual authorization acceptance, click (disabled) is displayed, authorizations of other enterprise accounts are automatically accepted.

When this option is enabled, you need to manually accept an authorization as follows:

Step 1 Log in to the console.

Step 2 Choose **Enterprise Account Authorization** from the navigation pane on the left, and click the **Received Authorizations** tab. The received authorization is displayed in the list, pending to accept.

Step 3 Accept or reject the authorization as you need.

- Accept: In the dialog box that is displayed, click **OK**. The status changes to **Enabled**.

  For the enterprise that grants the authorization, the status of the corresponding record on the **Granted Authorizations** tab page is also updated to **Enabled**.
- Reject: The status changes to **Rejected**.

  For the enterprise that grants the authorization, the status of the corresponding record on the **Granted Authorizations** tab page is also updated to **Rejected**.

----End

NOTE

An invited account does not need to subscribe to CodeArts. After accepting an invitation, the invited account is added to the inviting account's projects and can perform operations as a project member.

3.3 CodeArts Homepage

3.3.1 Homepage Overview

CodeArts provides an entry for each subservice. You can manage your projects, and configure personal and tenant settings.

Navigation Bar

CodeArts provides multiple choices of themes and layouts. This section uses "Infinite and Classics" as an example to introduce the navigation bar.

Table 3-4 Navigation bar on the homepage

Menu	Description
	The entry for each service so that you can go to their homepage.
Console	The console is displayed. You can select other products as required.
Homepage	Go to the CodeArts homepage.
Workspace	View the work items under your account. For details, see **Workspace**.
Services	Provides an entry for each service of CodeArts so that you can go to their homepage.
	View notification messages about operations on each service.
	View the user guide of each service.
	Summary of your to-do items.
	Alias, preference, **personal**, and **tenant** settings.

Projects

All projects in which you are involved are displayed. The project list displays followed, all, and archived projects on different tabs.

- Followed projects: projects marked with
- All projects: all projects in which you have joined
- Archived projects: projects that have been archived on the **Settings** > **General** > **Basic Information** page

You can perform the following operations.

Table 3-5 Managing projects

Operation	Description
Create a project	To create a project, you must have the project creator permission. For details about how to configure the permission, see **Project Creators**. Click **Create Project** and select a project type.
Add a project to a group	1. Click **Create Group**, enter a group name, and click **OK**. 2. Click a group name and click **Add an existing project**. Select the target projects and click **OK**.

Operation	Description
Search for a project	Enter a keyword in the search box above the project list.
Switch the display mode	Projects can be displayed in card view or list view. Click in the upper right corner of the project list.

3.3.2 Workspace

The workspace displays your work, including work items, wikis, and test cases.

Accessing the Workspace

Step 1 Log in to the CodeArts homepage.

Step 2 Choose **Workspace** on the navigation bar. The **My Work Items** tab is displayed by default.

Step 3 Switch to the desired tab.

----End

My Work Items

The **My Work Items** tab displays the work items and projects in which you have participated.

NOTE

Only the work items in your Scrum projects are displayed.

On this tab page, you can perform the operations described in the following table.

Table 3-6 Work item management

Operation	Description
Filter work items	Click the drop-down list to select predefined filters.
Search for work items	Enter a keyword or add a filter in the search box, and click .
Select fields	Click in the upper right corner of the list, perform the following operations as required, and click **OK**. • Select the fields to be displayed on the left pane. • Drag the fields on the right pane to adjust the display sequence.

Operation	Description
View/Edit work item details	Click a work item title in the list, view the work item details, and edit them if you need. For details, see "Viewing Work Item Details" in *CodeArts Req User Guide*.

My Wiki

The **My Wiki** tab page displays all entries created in all projects that you have joined.

On this tab page, you can perform the operations described in the following table.

Operation	Description
Search for entries	Enter a keyword in the search box, and click .
View entry details	Click an entry title in the list to go to the details page. On this page, you can edit the entry.

My Tests

The **My Tests** tab page displays all test cases in all projects that the current user participates in. **Owner** is the current user.

Click a case ID in the list to go to the case details page. You can view the case details or edit the case as required. For more operations related to test cases, see "Test Case" in the *CodeArts TestPlan User Guide*.

3.3.3 Setting an Alias

You can only set an alias for yourself. Aliases are visible to all project members.

When you set the work item handler, the alias of a user is displayed by default. If no alias has been set before, the username is displayed.

Step 1 Log in to the CodeArts homepage and click the username icon on the tool bar.

Step 2 Click next to the username.

Step 3 In the dialog box displayed, enter an alias and click **OK**.

Step 4 Refresh the page. The new alias is displayed in the upper right corner. Refresh the page if the alias is not displayed.

----End

NOTE

If you have selected **Disable Custom Alias** after choosing **All Account Settings** > **General** > **Global Settings**, no IAM users of the tenant can set an alias. is unavailable.

3.3.4 This Account Settings

3.3.4.1 Notifications

Configure whether to receive notifications related to CodeArts services on the **Notifications** page.

Enabling/Disabling Notifications

Step 1 Log in to the CodeArts homepage and click the username icon on the tool bar.

Step 2 Choose **This Account Settings**.

Step 3 Enable or disable notifications on the **Notifications** page.

To change the email address for receiving notifications, click **Edit Settings** and change the email address as prompted.

----End

Setting Do-Not-Disturb Time Period

By default, you receive notifications from CodeArts 24 hours a day. You can specify a time period during which you do not want to receive any notifications.

Step 1 Turn on the **Do-Not-Disturb** () switch.

Step 2 Click **Edit Settings**, enter the start and end time, and click **OK**.

----End

3.3.5 All Account Settings

3.3.5.1 General

3.3.5.1.1 Project Creators

Prerequisites

Your account meets any of the following requirements:

- Tenant account
- With the **Tenant Administrator** permission

Procedure

Step 1 Log in to the CodeArts homepage and click the username icon on the tool bar.

Step 2 Choose **All Account Settings**.

Step 3 Choose **General** > **Project Creators**.

Step 4 Select **Only some members can create projects**. The member list is displayed.

Step 5 Enable or disable the authorization by clicking . Unauthorized members cannot create projects.

----End

3.3.5.1.2 Managing Projects and Members

Prerequisites

You can view projects that you have joined or have not joined, and the project members.

- The joined projects page is available for all members.
- To view unjoined projects and their members, ensure that your account meets any of the following requirements:
  - Tenant account
  - With the **Tenant Administrator** permission

Procedure

Step 1 Log in to the CodeArts homepage and click the username icon on the tool bar.

Step 2 Choose **All Account Settings**.

Step 3 Choose **General** > **Projects and Members**.

Step 4 View projects and members.

- By default, the **Unjoined Projects** tab is displayed.

  To join a project, select it and click **Join Project**.
- Click the **Joined Projects** tab. The joined projects are displayed.

  To delete a project, click .
- Click the **Project Members** tab. All projects in your account and the project members are displayed.

  To remove a member from a project, click .

  To remove multiple members, select them and click **Batch Remove**.

----End

3.3.5.1.3 Importing IAM Users

Prerequisites

Your account meets any of the following requirements:

- Tenant account
- With the **Tenant Administrator** permission

Procedure

Step 1 Log in to the CodeArts homepage and click the username icon on the tool bar.

Step 2 Choose **All Account Settings**.

Step 3 Choose **General** > **IAM Users**.

Step 4 Click **Batch IAM User Import**.

Step 5 Click **Download Template**.

Step 6 Open the template. Modify and save information.

Step 7 Return to the **IAM Users** page, click , select the edited file, and agree to the statement.

Step 8 When a message indicating a successful import is displayed, go to the IAM console to view the imported users.

----End

3.3.5.1.4 Managing Global Settings

Setting an Alias

By default, all users in your account can set an alias in CodeArts. If you do not need to use this function, disable it.

Step 1 Log in to the CodeArts homepage and click the username icon on the tool bar.

Step 2 Choose **All Account Settings**.

Step 3 Choose **General** > **Global Settings**.

Step 4 Click next to **Disable Custom Alias**.

Step 5 Refresh the current page. When you **set an alias**, the system displays a message indicating that the modification is not allowed.

----End

3.3.5.2 Agent Pools

3.3.5.2.1 Creating an Agent Pool

Background

You can add your own execution resources to the agent pool. When executing a task, you can select an agent in the pool to execute the task, improving efficiency without depending on public execution resources.

Procedure

Step 1 Log in to the CodeArts homepage and click the username icon on the tool bar.

Step 2 Choose **All Account Settings**.

Step 3 Click **Agent Pool > Agent Pool**. On the displayed **Pools** page, click **Create Pool**, configure the pool, and click **Save**.

Table 3-7 Configuring the agent pool

Configuration Item	Description
Pool Name	Name of the agent pool, which can be customized as required. Enter a maximum of 50 characters, including digits, letters, periods (.), underscores (_), and hyphens (-).
Pool Type	Supported pool types: ● **LINUX**: Tasks are executed on a Linux VM. ● **LINUX_DOCKER**: A Linux Docker container is started and tasks are executed in the container. ● **WINDOWS**: Tasks are executed on a Windows VM. ● **MAC**: The agent registration command is run on a MAC host.
Description	Enter a description. Enter a maximum of 1,024 characters.
This pool can be used by all users of the current account.	If this parameter is selected, all users of the current account can use the pool.

Step 4 View the new record after the creation is successful.

----End

3.3.5.2.2 Creating an Agent

Overview

You can install an agent on a host and connect the host to CodeArts for registration. You can customize execution resource connection.

Install only one agent on an agent host. If multiple agents are installed, some of them may go offline during task execution.

Prerequisites

- Your account must be:
  - A tenant account.
  - Or an IAM user who has the agent pool owner, manager, or user permission.

- The host where you want to install an agent must meet the following requirements:
  - Specifications: 4 vCPUs | 8 GB or above, disk space > 80 GB
  - Java 8 and Git are installed. If the agent pool type is LINUX_DOCKER, Docker must also be installed on the host.

    NOTE

    - If you use a Linux host, you can choose auto agent installation, or manually install the agent using the **root** account.
    - If you use a Windows or macOS host, manually install the agent on it.
  - **VPCEP has been configured**.

Configuring VPCEP

Step 1 Log in to the console, and click **VPC Endpoint**.

Step 2 Click **Create VPC Endpoint**, configure the following information, and click **Create Now**.

Table 3-8 Creating a VPC endpoint

Parameter	Description
Service Category	Select **Find a service by name**.
VPC Endpoint Service Name	Submit a service ticket to obtain the information and enter it. Click **Verify**. If the message "Service name found." is displayed, the verification is successful.
VPC	Select a VPC. If there is no VPC, click **Create VPC** in the drop-down list to create one.
Subnet	Select a subnet. If there is no subnet, click **Create Subnet** in the drop-down list to create one.
IPv4 address	Select **Automatically assign IP address**.

Step 3 In the VPC endpoint list, locate the created VPC endpoint and record its IP address.

Step 4 Click ☰ in the upper left, and click **Domain Name Service** in the service list.

Step 5 Click **Create Private Zone**. In the displayed dialog box, set the following information and click **OK**.

Table 3-9 Creating a private zone

Parameter	Description
Domain Name	Submit a service ticket to obtain the information and enter the information.
VPC	Select the same VPC as the one in **Step 2**.

Parameter	Description
Enterprise Project	Select an existing enterprise project. If there is no enterprise project, click **Create Enterprise Project** to create one.

Step 6 Locate the created private zone, and click **Manage Record Set** in the **Operation** column.

Step 7 Click **Add Record Set**, enter the IP address recorded in **Step 3** in the **Value** text box, and click **OK**.

----End

Creating an Agent

Step 1 After completing the operations described in **Creating an Agent Pool**, find the target agent pool in the agent pool list and click its name to display the **Agents** tab page.

Step 2 Click **Create Agent**. A dialog box is displayed.

Step 3 Configure parameters as required.

1. Configure the network environment for the host.

Table 3-10 Configuration 1

Configuration Item	Description
Install a JDK automatically	The host must have a Java 8 environment. After this function is enabled, the JDK is automatically installed in the **/usr/local** directory of the host. If this function is disabled, you need to manually install a JDK. **How Do I Install Java 8?** **NOTE** This configuration item is displayed only when the agent pool type is **LINUX** or **LINUX_DOCKER**.
Install Git automatically	The host must have a Git environment. After this function is enabled, Git is automatically installed on the host. If this function is disabled, you need to manually install Git. **How Do I Install Git?** **NOTE** This configuration item is displayed only when the agent pool type is **LINUX** or **LINUX_DOCKER**.

Configuration Item	Description
Install Docker	The host must have a Docker environment. After this function is enabled, the Docker is automatically installed in the **/usr/bin** directory of the host. After the installation is complete, the service is automatically registered and the Docker service is started. If this function is disabled, you need to manually install Docker. **How Do I Install Docker?** **NOTE** This configuration item is displayed only when the agent pool type is **LINUX** or **LINUX_DOCKER**.

2. Configure authentication information.

Table 3-11 Configuration 2

Configuration Item	Description
AK	Access key ID. Obtain it from the access key file by choosing **My Credential** > **Access Keys**. For details, see "Obtaining an AK/SK" in the *API Signature Guide*.
SK	Secret access key. Obtain it from the access key file by choosing **My Credential** > **Access Keys**. For details, see "Obtaining an AK/SK" in the *API Signature Guide*.
Agent Name	Agents with the same name cannot be installed on the same host. Set this parameter based on the site requirements. To facilitate management, it is recommended that the name be clear and associated with the agent IP address, such as **10.10.10.10-agent-01**.
Agent Workspace	The working directory where the agent runs on the host and the workspace where tasks are executed. Different agents of the same host cannot use the same agent workspace. For easy management, specify a data disk. Example: – Linux: **/opt/cloud/agent01** – Windows: **C:/opt/cloud/agent01** – macOS: **/opt/cloud/agent01**

3. Generate a command.

Set the parameters, agree to the statements, and click **Generate Command**. The command for installing the agent is automatically generated in the command box.

Click **Copy Command**.

Step 4 Log in to the host and run the installation command copied in the previous step.

- Linux host: Log in using the **root** account and then run the installation command.
- Windows host: Log in as the administrator, open Git Bash, and then run the installation command.
- macOS host: Log in using the **root** account and then run the installation command.

If the following message is displayed, the installation is complete.

Step 5 Return to the **Agents** tab page and close the dialog box.

After 10 to 30 seconds, check the status of the installed agent instance on the **Agents** tab page. If the status is **Idle**, the installation is successful.

----End

Follow-up Procedure

- View the agent list.

  In the agent pool list, find the target agent pool and click its name to display the **Agents** tab page.

  The agent list displays the agent status, memory usage, and available hard disk space.

  An agent can be in any of the following states:

  Table 3-12 Agent states

Agent State	Description
Running	The agent is performing tasks.
Idle	The agent is connected, and no tasks are running.
Disabled	The agent is connected, but is disabled to perform tasks.
Offline	The agent is disconnected. If you want it to go online, log in to the agent to re-register. Otherwise, you can delete the agent.
Deleting	The agent is being deleted.

- Delete an agent.
  - Deleting a single agent: On the **Agents** tab, click  in the **Operation** column of the target agent to delete it.
  - Deleting multiple agents: On the **Agents** tab, select the target agents and click **Batch Delete**.

NOTE

Agents in the **Running** state cannot be deleted.

- Start or stop the agent.

  On the **Agents** tab page, click in the **Operation** column to disable the agent. After the agent is disabled, the status changes to **Disabled**. Click to enable the agent again and the status changes to **Idle**.

3.3.5.2.3 Managing Agent Pools

Viewing Pool Details

Step 1 After completing the operations described in **Creating an Agent Pool**, find the target agent pool in the agent pool list and click its name to display the **Agents** tab page.

Step 2 Click the **Pool Details** tab to view information including the owner, pool type, creation time, and description.

----End

Managing Pool Permissions

Agent pools involve the following roles.

Table 3-13 Agent pool roles

Role Name	Description
Manager	Has full permissions for the current agent pool, including viewing, using, configuring, and modifying the agent pool.
User	Can view the agent pool's details, operation records, and agent list, and use agents to execute tasks.

Step 1 After completing the operations described in **Creating an Agent Pool**, find the target agent pool in the agent pool list and click its name to display the **Agents** tab page.

Step 2 Click the **Permissions** tab to manage agent pool permissions.

Table 3-14 Managing permissions

Operation	Description
Set manager	• Click **Add** next to **Manager**, select a user from the drop-down list, and click . • To cancel the manager role of a user, click next to the username.

Operation	Description
Set user	• Click the switch to grant the user role to all users in your account. • Click **Add** next to **Projects**, select a project name from the drop-down list box, and click ✓ to set all members in the selected project as agent pool users. • To cancel the authorization for members in a project, click next to the project name.

----End

Viewing Historical Records of an Agent Pool

Step 1 After completing the operations described in **Creating an Agent Pool**, find the target agent pool in the agent pool list and click its name to display the **Agents** tab page.

Step 2 Click the **Operation Records** tab to view the operation records of the pool.

----End

Configuring Message Notifications

Step 1 After completing the operations described in **Creating an Agent Pool**, find the target agent pool in the agent pool list and click its name to display the **Agents** tab page.

Step 2 Click the **Notify** tab and configure event notifications for the agent pool as required.

You can determine whether to send service news or emails to permission managers when the following operations are triggered:

- Create an agent
- Delete an agent
- Disable an agent
- Enable an agent
- Bring an agent offline
- Bring an agent online

----End

4 FAQs

4.1 Concepts and Functions

4.1.1 What Languages, Environments, and Applications Does CodeArts Support?

CodeArts supports end-to-end R&D of Internet, web, mobile, embedded, and enterprise applications.

It also supports mainstream languages including Java, C, C++, C#, Python, and Go, multiple build environments and tools such as Maven, Ant, and CMake, as well as multiple deployment templates such as Tomcat and Spring Boot.

For more information, go to the *Service Overview* of each individual service.

4.1.2 Can I View My Tasks and Their Progress on CodeArts?

Yes.

- You can switch to the **Worktable** page from the homepage and view the work items assigned to you and the projects to which the work items belong.
- On the **Work** > **Work Items** page of a project, you can use filters to query the work items assigned to a specific person in a project. For details, see "Managing Work Items" in *CodeArts Req User Guide*.

4.1.3 How Do I View Operation Logs of Members on CodeArts?

Currently, CodeArts cannot record operation logs for each project member. However, you can view the operation history of each service. For example:

- Work items: For details, see "Managing Work Items" in *CodeArts Req User Guide*.
- Code: For details, see "Viewing Commit History" in *CodeArts Repo User Guide*.

- Deployment: "Viewing Applications" in *CodeArts Deploy User Guide*.

For details, see the user guide of each service.

4.1.4 Can CodeArts Check, Build, and Deploy Local Code?

Partially yes.

- CodeArts Check and CodeArts Build support code sources such as CodeArts Repo, GitHub, and Git. You can set the code source to a general-purpose standard Git repository on the local network accessible from the Internet.
- Software packages used by CodeArts Deploy need to be obtained from release repos of CodeArts Artifact. Therefore, you need to upload the software packages to release repos.

4.1.5 Can I Migrate Code and Work Items to CodeArts from a Local Path or Another Platform?

Yes.

- Work item migration: CodeArts Req provides the batch upload function. Download the template from CodeArts Req, fill in the work item information in the template, and upload the template. For details, see "Managing Work Items" in CodeArts Req User Guide.
- Code migration: CodeArts Repo allows code migration from a local or online Git repository. For details, see "Migrating to CodeArts Repo" in *CodeArts Repo User Guide*.

4.1.6 How Do I Collect or Archive Project Information Locally After a CodeArts Project Is Complete?

CodeArts provides cloud-based project archiving. Archived projects are read-only to all members and cannot be added, deleted, or modified.

CodeArts has a comprehensive data security and management mechanism to ensure that data on the cloud will not be lost and can be viewed at any time.

In addition, each service provides the following functions to meet your archiving requirements.

- "Managing Work Items" in *CodeArts Req User Guide*.
- "Cloning Code to the Local PC" in *CodeArts Repo User Guide*.
- "Performing Basic Operations" in *CodeArts Artifact User Guide*.
- "Migrating Manual Test Cases" in *CodeArts TestPlan User Guide*.

4.2 Members and Permissions

4.2.1 Can I View the Time When a Member Is Invited into or Removed from a Project?

No.

- When a member is invited into or removed from a project, the system sends a notification to the member via the email or pop-up message.
- CodeArts does not display records of a member who joins and quits a project.

4.2.2 Why Cannot an IAM User View Project Information After Login?

Symptom

An administrator has created an IAM user for a team member. After the member logs in to the system as an IAM user, no project is displayed on the page.

Root Cause

The administrator has not added the IAM user to any project. The IAM user can view the project information only after it becomes a project member.

Solution

Contact the administrator to add the IAM user as a project member. For details, see "Members" in *CodeArts Req User Guide*.

4.2.3 Will Operation Records of an IAM User Be Deleted When the IAM User Is Deleted?

No.

To ensure information consistency in a project, historical records of operations performed by an IAM user are still stored in the system after the IAM user is deleted. The historical records include projects, work items, code repositories, and service tasks.

For example, after an IAM user is deleted, their data is not released. Projects created by the IAM user are automatically handed over to the tenant administrator (master account), and documents in the projects are retained.

4.2.4 Why Cannot a Tenant Administrator View Projects Created by an IAM User?

Symptom

After login, the tenant account can view only their own projects on the homepage, but cannot view the projects created by IAM users.

Root Cause

Only project members can view project details, such as work items, test cases, and build and deployment tasks.

When an IAM user creates a project, the tenant administrator will not become the project member by default. The tenant administrator cannot view all projects of the tenant on the homepage.

Solution

Perform the following operations to enable the tenant administrator to manage all projects and members of the tenant:

Step 1 Log in to the CodeArts homepage and click the username icon on the tool bar.

Step 2 Choose **All Account Settings**.

Step 3 Choose **General** > **Projects and Members**.

Step 4 On the **Unjoined Projects** tab page, view the projects that are created by IAM users but do not include the tenant administrator as their member.

If you need to view more detailed project information, select projects as required and click **Join Project**.

----End

4.3 Security

4.3.1 Can I Set Project Members to View Only Tasks (Work Items) But Not Codes?

Yes.

CodeArts supports member management policies for code repositories. Project members who are not added to a code repository have no permission to view the code repository information. After a user logs in to CodeArts Repo, code repositories that the user does not have permission to access are not displayed.

For details, see "Configuring Member Management" in *CodeArts Req User Guide*.

4.3.2 Can I Restrict Employees to Access Code Repositories Only in the Office?

Yes.

CodeArts Repo allows you to set IP address whitelists. Only accesses from IP addresses in the whitelist are allowed. The administrator can configure an IP address whitelist for each code repository to restrict accesses. For details, see "IP Address Whitelist" in *CodeArts Repo User Guide*.

