

CodeArts

Best Practices

Issue 01
Date 2026-04-09



Copyright © Huawei Technologies Co., Ltd. 2026. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:

<https://www.huawei.com/en/psirt/vul-response-process>

For vulnerability information, enterprise customers can visit the following web page:

<https://securitybulletin.huawei.com/enterprise/en/security-advisory>

Contents

1 Best Practices Summary.....	1
2 Developing an E-mall Project with CodeArts.....	4
2.1 Overview.....	4
2.2 Resource Planning.....	8
2.3 Process.....	9
2.4 Implementation Procedure.....	11
2.4.1 Preparations.....	11
2.4.2 Managing Project Plans.....	12
2.4.3 Developing Code.....	16
2.4.4 Checking Code.....	20
2.4.5 Building Images and Pushing Them to SWR.....	22
2.4.6 Deploying Applications to CCE.....	29
2.4.7 Deploying Applications to ECS.....	33
2.4.8 Managing Project Tests.....	37
2.4.9 Configuring Pipelines.....	42
2.4.10 Releasing Resources.....	45
3 Configuring CodeArts Security.....	46
4 Configuring CodeArts Permissions.....	52
4.1 Overview.....	52
4.2 Implementation Procedure.....	59
4.2.1 Preparations.....	59
4.2.2 Configuring Permissions for the Project Manager Role.....	60
4.2.3 Configuring Permissions for the R&D Team.....	61
4.2.4 Configuring Permissions for the Middle-end Team.....	64
4.2.5 Configuring Permissions for the Architecture Team.....	65
4.2.6 Configuring Permissions for the Test Team.....	66
4.2.7 Configuring Permissions for the PO Team.....	68
4.2.8 Configuring Permissions for the PM Team.....	70
4.2.9 Configuring Permissions for the Big Data Team.....	71
4.2.10 Configuring Permissions for the O&M Team.....	73
4.2.11 Configuring Permissions for the DBA Team.....	76

4.2.12 Configuring Permissions for the Security Team..... 78

1 Best Practices Summary

Table 1-1 Best practices

Service	Best Practice
General	<ul style="list-style-type: none">• Developing an E-mall Project with CodeArts• Security Configuration• Configuring CodeArts Permissions
CodeArts Req	<ul style="list-style-type: none">• Scrum Project Best Practices• Managing Smart Watch RRs with an IPD-System Device Project• Managing Smart Watch Bugs with an IPD-System Device Project• Managing Smart Watch Baseline Reviews with an IPD-System Device Project• Managing Smart Watch Feature Tree with an IPD-System Device Project• Using an IPD-System Device Project to Review Changes to a Smart Watch R&D Project• Best Practices for CodeArts Req Permission Management
CodeArts Repo	<ul style="list-style-type: none">• Managing Repository Members and Permissions in an Enterprise• Migrating GitLab Intranet Repositories to CodeArts Repo in Batches• Importing Local Repositories to CodeArts Repo in Batches• Overview of CodeArts Repo Security Configuration• Configuring HTTPS Password• Best Practices for Configuring Protected Branch Rules

Service	Best Practice
CodeArts Pipeline	<ul style="list-style-type: none">● Fixing a Bug for Quick Release Through a Change-triggered Pipeline● Configuring Pass Conditions for Automated Code Checks● Orchestrating Build and Deployment Tasks with Pipeline Parameters● Creating a Repository Tag Using Pipeline Contexts● Configuring a Pipeline Gate for a Code Repository Merge Request● Pipeline-level Permission Management● Controlling the Release Time with a Custom Pipeline Extension
CodeArts Check	<ul style="list-style-type: none">● Checking Code from Git with Preset Rules● Checking Code from CodeArts Repo with Custom Rules● Checking Code with Custom Executors● Executing Code Check Tasks
CodeArts Build	<ul style="list-style-type: none">● Creating a Docker Image with a Maven Artifact and Pushing the Image to SWR● Building with Maven and Uploading the Software Package to the Self-Hosted Repo● Building with Maven to Generate a Private Dependency for Another Build● Building with npm and Uploading the Software Package to the Release Repo● Building with Maven on Custom Executors● Building with Maven, Uploading the Software Package, and Pushing the Image to SWR● Running a Multi-Task Maven Build Project● Building with Maven Based on Non-default Dependencies and Uploading the Software Package to a Self-Hosted Repo● Running a Build Task on a Custom Environment

Service	Best Practice
CodeArts Artifact	<ul style="list-style-type: none"> ● Releasing Maven Components and Archiving to a Self-Hosted Repo ● Releasing/Obtaining an npm Component via a Build Task ● Releasing/Obtaining a Go Component via a Build Task ● Releasing/Obtaining a PyPI Component via a Build Task ● Uploading/Obtaining an RPM Component Using Linux Commands ● Uploading/Obtaining a Debian Component Using Linux Commands ● Migrating Repository Data from Nexus to CodeArts Artifact ● Migrating Repository Data from Local Repository to CodeArts Artifact
CodeArts Deploy	<ul style="list-style-type: none"> ● Deploying an Application on an Intranet Host Using a Proxy Host ● Using Nginx for Gray Release ● Using Kubernetes Nginx-Ingress for Gray Release ● CodeArts Deploy Security Best Practices ● Best Practices for CodeArts Deploy Permission Management
CodeArts TestPlan	<ul style="list-style-type: none"> ● Three-Layer Test Case Management ● E-Commerce Platform Test Driven by API Automation Test Cases and Keywords ● Designing a Test Based on Requirements
CodeArts PerfTest	<ul style="list-style-type: none"> ● Performance Tests of All-in-One Systems for Government Services ● Native Performance Pressure Test of JMeter Test Projects ● Best Practices for Configuring Global Variables ● Using Private Resource Groups to Perform Pressure Tests Through the Private Network in the Same VPC
CodeArts Governance	<ul style="list-style-type: none"> ● Performing Binary Software Composition Analysis (SCA)

2 Developing an E-mall Project with CodeArts

2.1 Overview

This practice demonstrates an e-mall project to show how a product R&D team uses the HE2E DevOps framework with CodeArts to develop software from end to end.

This document defines a virtual team and the roles involved in software R&D. It explains how different roles utilize CodeArts' tools—work items, code repositories, tests, artifact repositories, and pipelines—alongside other cloud services, for analysis, design, development, and testing. It serves as a reference for E2E continuous software delivery.

HE2E DevOps Framework

This framework is an operable and implementable agile development methodology proposed by CodeArts using both our experience and advanced industry practices.

Figure 2-1 HE2E DevOps implementation framework

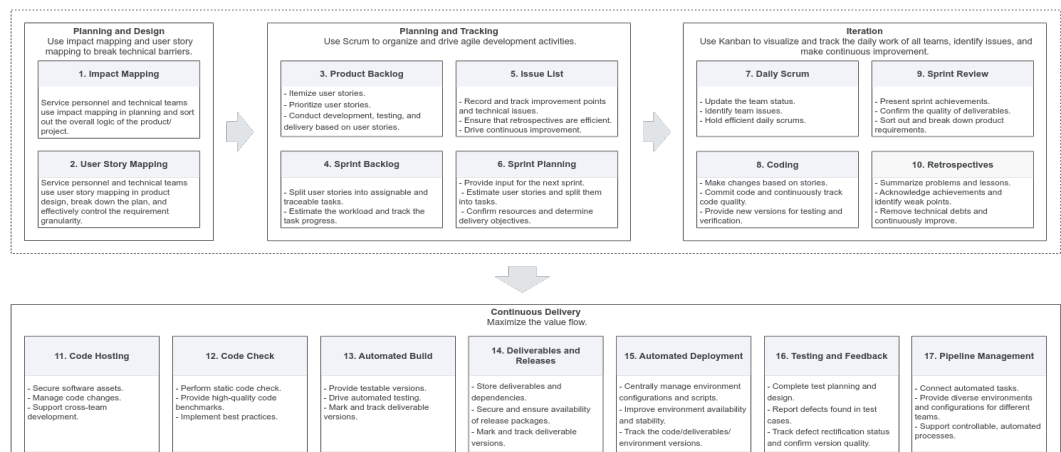


Table 2-1 HE2E DevOps implementation framework

Stage	Description
Planning and Design	<p>The first two steps in the diagram depict the product planning process where service personnel (or customers) collaborate with technical personnel to establish the product's overall logic, plan and design the product, and break down requirements at a specific granularity.</p> <ul style="list-style-type: none"> • Software development solves issues and delivers value, not just simply provides functions. Impact mapping identifies user requirements and root causes. • User stories convey objectives and requirements in specific scenarios, facilitating communication between customers, service personnel, and developers. If you only view separate requirement items, you cannot consider them from the entire solution's perspective. User stories focus on scenarios, sorting out and displaying stages and activities in a tree structure. In this way, you will view both requirement items and overall requirement scenarios.
Planning and Tracking	<p>Steps 3 to 10 belong to the Scrum framework and are also the main management practices.</p>
Iteration	<ul style="list-style-type: none"> • Scrum defines a relatively complete framework for agile process management. In CodeArts, the Scrum framework is integrated with your team's daily development activities. Main process deliverables include the product backlog, sprint backlog, potential deliverable product increments, and issue list. Core team activities include sprint planning meetings, daily Scrums, sprint reviews, sprint retrospective meetings, and daily updates. • In addition, your team can learn from the lean thinking in Kanban, visualize value streams, and identify and resolve blockage and bottlenecks. These actions help to accelerate value stream delivery and feedback loop for continuous improvement.

Stage	Description
Continuous Delivery	<p>Starting from step 11 is the engineering practice, that is, the CI/CD process.</p> <ul style="list-style-type: none"> Continuous delivery is based on code repositories. It not only covers traditional security control of code assets, concurrent development, and version and baseline management, but also reflects team collaboration and communication. The pipeline connects code check (or static scanning), automated build, automated testing in all stages, and automated deployment. In addition to dynamic stages and activities like code check, build, testing, and deployment, continuous delivery also involves artifact and environment (development, test, quasi-production, and production) management. The continuous delivery pipeline manages stages, environments, stage activities, entry and pass quality gates, and input and output artifacts in each stage.

Background

Company A is an automobile parts dealer. Company A develops "Phoenix Mall", an online store for automobile parts, to boost sales.

The mall manages members, parts, and orders, meeting company A's management requirements and allowing customers to query and purchase products.

The Scrum mode is used for iterative development. Each iteration period is two weeks.

Solution Architecture

This automobile parts e-mall consists of five microservice components.

Figure 2-2 Technical architecture of Phoenix Mall

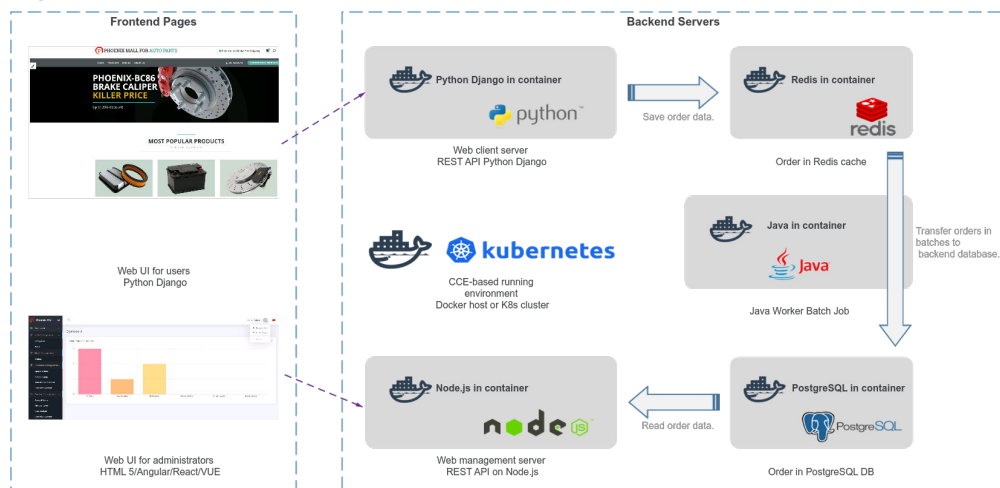


Table 2-2 Architecture

Microservice Component	Description
Web client server (corresponding to the Vote function in the sample code)	<ul style="list-style-type: none">• Service logic: Users can use a browser to access the web UI of this service. When a user clicks Like on a specific offering, the service saves the record of the selected offering in the Redis cache.• Technology stack: Python and Flask frameworks• Application server: Gunicorn
Web management server (corresponding to the Result function in the sample code)	<ul style="list-style-type: none">• Service logic: Users can use a browser to access the web UI of this service. The statistics about Like clicked by users on the UI are dynamically displayed. The data is obtained from the PostgreSQL database.• Technology stack: Node.js and Express frameworks• Application server: server.js
Background order batch processing program (corresponding to the Worker function in the sample code)	<ul style="list-style-type: none">• Service logic: Serves as a background process to monitor product records in Redis, collect new records, and save them in the PostgreSQL database for extraction and display by the management UI.• Technology stack: .NET Core or Java (This service provides two technology stacks to implement the same function. You can modify the configuration and select one of the technology stacks as the runtime process.)
Order cache	<ul style="list-style-type: none">• Service logic: Persists data for the client UI.• Technology stack: Redis
Order database	<ul style="list-style-type: none">• Service logic: Serves as the data source of the management UI.• Technology stack: PostgreSQL

The following members are involved in the project R&D.

Table 2-3 Project role list

Project Member	Project Role	Responsibility
Sarah	Project admin	Plans the project and sets up a team.
Maggie	Project manager	Manages project delivery plans.

Project Member	Project Role	Responsibility
Chris	Developer	Develops, compiles, deploys, and verifies project code.
Billy	Tester	Writes and executes test cases.

2.2 Resource Planning

The required resources for this practice are listed in the table below. Prepare resources in the same region. This practice provides Cloud Container Engine (CCE) and Elastic Cloud Server (ECS) based deployment methods. Choose to prepare CCE or ECS resources to suite your needs.

⚠ CAUTION

ECS and CCE will be billed once purchased. For details, see [ECS Price Calculator](#) and [CCE Price Calculator](#).

Table 2-4 Resource planning

Service Name	Usage	Resource Description	Operation Guide
CodeArts	For E2E project R&D.	<ul style="list-style-type: none">Package edition: BasicUsers: 5Required duration: 1 month	Purchasing CodeArts.
CCE	For deploying applications.	Cluster configuration: Select the pay-per-use billing mode. <ul style="list-style-type: none">Cluster type: CCE standard clusterCluster name: phoenix-cceCluster version: the latest versionController node architecture: x86Network model: tunnel networkContainer CIDR block: auto select	Buying a CCE Standard/Turbo Cluster

Service Name	Usage	Resource Description	Operation Guide
		<p>Node configuration: Select the pay-per-use billing mode.</p> <ul style="list-style-type: none">• Node type: Elastic Cloud Server (VM)• Specifications: 2 vCPUs 8 GiB or above• Container engine: Docker• OS: Public image - EulerOS• Node IP: auto assign• EIP: auto create	Creating a Node
ECS	For deploying applications.	<p>Select the pay-per-use billing mode.</p> <ul style="list-style-type: none">• CPU architecture: x86• Specifications: 2 vCPUs 8 GiB or above• OS: Public image - Ubuntu 16.04• Elastic IP (EIP): billed by bandwidth, 5 Mbit/s• ECS Name: phoenix-ecs <p>After the purchase, add an inbound rule for the ECS's security group. Set the protocol to TCP, port to 5000-5001, and source address to 0.0.0.0/0.</p>	<ul style="list-style-type: none">• Buying an ECS• Configuring Security Group Rules

2.3 Process

This document describes the operation process of the HE2E DevOps practice.

Figure 2-3 Process

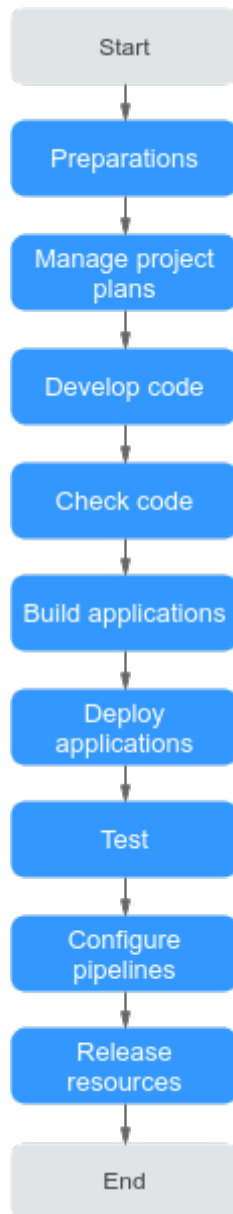


Table 2-5 Process description

Step	Description
Preparations	Project administrator Sarah completes preparations before the practice, including creating a project and adding members.
Manage project plans	Project administrator Sarah plans project requirements, and project manager Maggie plans sprint requirements.
Write code	Developer Chris writes code using a branch, including creating the branch, writing and committing code, and submitting a branch merge request. Project manager Maggie reviews the merge request and merges the code.

Step	Description
Check code	Developer Chris performs static scanning on the code and optimizes the code based on fix suggestions to improve the code quality.
Build application	Developer Chris builds environment images and compiles and packages code into software packages.
Deploy application	Developer Chris installs the built environment images and software packages and runs them in the environment. This document provides methods for CCE and ECS-based deployments.
Manage project tests	Tester Billy creates a sprint test plan, designs test cases, and executes test cases as planned.
Configure pipeline	Developer Chris orchestrates code check, build, and deployment tasks into a pipeline. When code is updated, the pipeline is automatically triggered for continuous delivery.
Release resources	After completing this practice, project administrator Sarah deletes unused pay-per-use resources to prevent extra charges.


2.4 Implementation Procedure

2.4.1 Preparations

Before starting the practice, complete the following preparations. In this practice, project administrator Sarah performs the preparations.

Creating a Project

Step 1 Go to the CodeArts homepage.

1. Log in to the [CodeArts console](#), click , and select a region where you have enabled CodeArts.
2. Click **Go to Workspace**.
If your account uses the old billing mode (see [Old Billing Modes](#)), click **Access Service**.

Step 2 Click **Create > Create Project**, and select **DevOps Full-Process**.

NOTE

If there are no projects displayed, select **DevOps Full-Process**.

Step 3 Enter the project name **Phoenix Mall** and click **OK**.

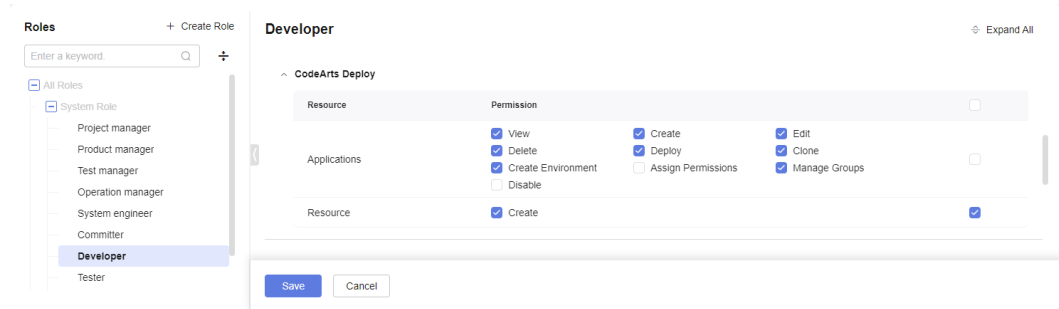
A success message is displayed, indicating that the project is created.

----End

Adding Project Members

- Step 1** Create three IAM users: **Maggie, Chris, and Billy**. For details, see [Creating an IAM User](#).
- Step 2** Return to CodeArts homepage, go to the **Phoenix Mall** project, and choose **Settings > Members** from the left navigation pane.
- Step 3** On the **Member View** tab, click **Add Members**, and then click **From My Account**.
- Step 4** In the displayed dialog box, select **Maggie, Chris, and Billy**, and click **Next**.
- Step 5** Click the **Role** drop-down list in each row, select **Project manager** for Maggie, **Developer** for Chris, and **Tester** for Billy, and then click **Save**.
The added members are displayed.
- Step 6** In the navigation pane, choose **Settings > Permissions**.
- Step 7** Select **Developer** from the role list, and click **Edit**. Enable **Create Environment** under **CodeArts Deploy > Applications**, and click **Save**.

Figure 2-4 Updating permissions



The modified permissions are displayed.

----End

2.4.2 Managing Project Plans

CodeArts Req facilitates easy and efficient team collaboration through various features, such as multi-project management, agile iteration, and task management.

This practice uses the Scrum mode for iterative development. Each sprint lasts for two weeks. The **Phoenix Mall** has been developed in the first three sprints, and sprint 4 is about to start.

Sprint 4 will add a new requirement: store query.

This section describes how project administrator Sarah and project manager Maggie manage requirements and sprints and track progress.

Adding Requirements to the Plan

In this practice, project requirements are managed using a mind map, which presents work items in the hierarchical structure of "Epic > Feature > Story >

Task". For more information, see [Scrum Project Requirement Management Process](#).

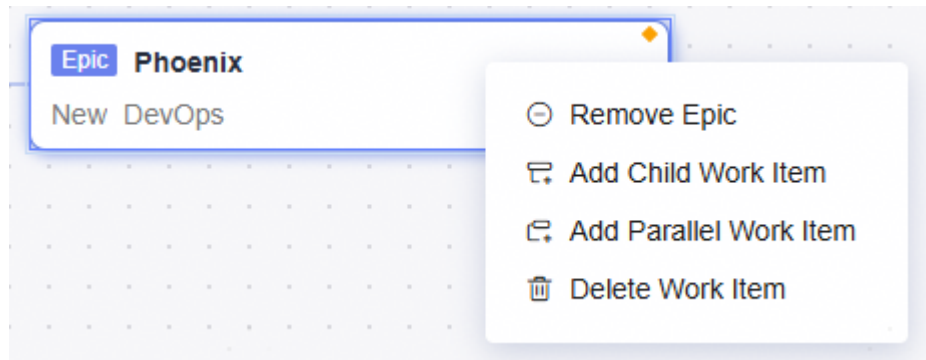
This operation is performed by project administrator Sarah.

Step 1 Create a work item for the new requirement.

Since the store query function is a new requirement, project administrator Sarah needs to add it to the requirement planning view.

1. Go to the **Phoenix Mall** project, and choose **Work Item > Req** to enter the CodeArts Req service.
2. Click the **Plans** tab and click **Phoenix-MindMap**.
3. Create a feature named **Store Network**.
 - a. Right-click **Phoenix** and choose **Add Child Work Item**.

Figure 2-5 Creating a feature



- b. After the feature is created, click the number next to **Epic** to expand the child work items, find feature **Phoenix**, and click the name to expand the work item details.
 - c. Change the name to **Store Network**.
4. Use the same method to add story **User can query network of all stores** to feature **Store Network**.

Step 2 Edit the story.

1. Click story **User can query network of all stores** and edit it by referring to the following table.

Table 2-6 Story configurations

Parameter	Example	Description
Description	As a user, I want to query all stores so that I can select a proper store for services.	Description of the story. Enter a maximum of 50,000 characters.
Priority	High	Priority of the story. Select High , Middle , or Low .

Parameter	Example	Description
Severity	Critical	Severity of the story. Select Critical , Major , Minor , or Trivial .

- To facilitate understanding, prepare a local file named **Store List** that includes the content in the table below.

Table 2-7 Store list

Branch Name	Branch Address
Branch A	123 meters to the departure floor, Terminal 1, Airport E
Branch B	No. 456, Street G, Area F
Branch C	No. 789, Street J, Area H
Branch D	West side of Building K, Avenue L, Area K

- Return to the story editing page, click **Select or Drag & Drop File**, choose **Upload** from the drop-down list, and upload the store list file to the work item as an attachment.
- Click **Save**. The story details are edited.
The edited story information is displayed.

Figure 2-6 Story details

Story Created by Sarah at Jul 11, 2025 14:49:52 GMT+08:00 | DevOps Full-Process

#69748089 User can query network of all stores

Description Child Work Items (0) Associations (0) Modeling Person-Hour Details Operation History

As a <user role>

I want <results>

So that <purposes>

Tag

Attachment

Store List.xlsx

Comment

Status:

* Assigned To:

Module:

Sprint:

Planned Start Date:

Planned Delivery D...:

Order:

* Priority:

* Severity:

Notify:

ParentId:

Domain:

* Expected:

[More](#)

----End

Adding the Story to a Sprint

Before iteration starts, the project manager Maggie organizes a plan meeting to add the new story to a sprint, break down the story into tasks, and assign the tasks to developers.

Step 1 Create a sprint.


1. Go to the **Phoenix Mall** project, and choose **Work Item > Req** to enter the CodeArts Req service.
2. Click the **Sprints** tab, and click  next to **Sprints** in the upper left corner. In the displayed dialog box, configure the sprint, and click **OK**.

Table 2-8 Sprint information configurations

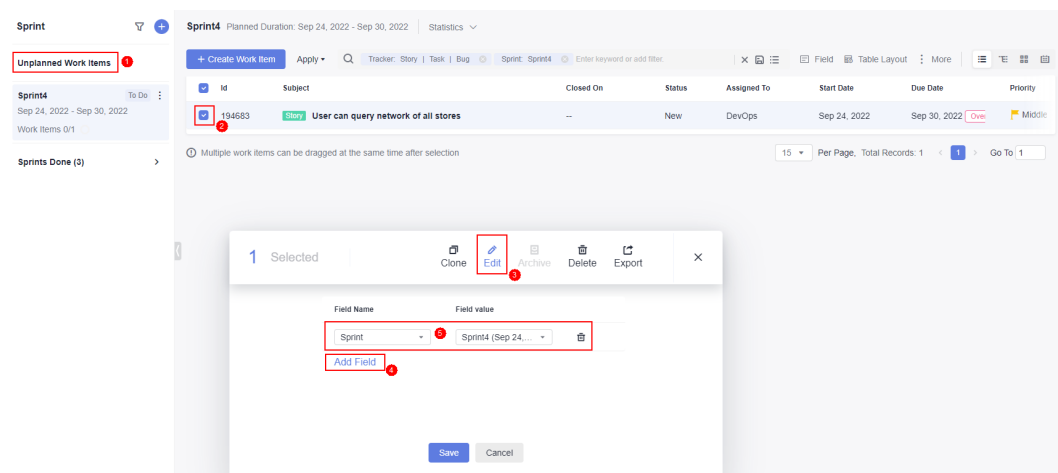
Parameter	Example	Description
Sprint Name	Sprint4	Sprint name. Enter a maximum of 60 characters. Letters, digits, periods (.), underscores (_), and hyphens (-) are supported.
Planned Duration	Set the duration to 2 weeks.	The planned start and end time of the sprint.

The new sprint is displayed.

Step 2 Add a story to the sprint.

1. From the left navigation pane, choose **Unplanned Work Items**.
2. Select the story **User can query network of all stores**.
3. Click **Edit** at the bottom of the page.
4. Click **Add Field**.
5. Choose **Sprint** from the **Field Name** drop-down list box, select **Sprint4** from the **Field Value** drop-down list box, and click **Save**.

Figure 2-7 Planning the sprint



Click **Sprint4**. The added story is displayed.

Step 3 Assign the story.

1. Choose **Sprint4** from the left navigation pane.
2. Find the story **User can query network of all stores** in the list, and click the story name. On the story details page, set the **Handler** field to **Chris**.

Step 4 Break down the story.

1. Find the story **User can query network of all stores**. Click the story name to open the details page.
2. Click the **Child Work Items** tab, and click **Quick Create**. Enter the title **Frontend display - Add store network menu**, assign it to **Chris**, and click **OK**.
The new child work item is displayed.
3. Use the same method to add task **Background management - Add store network management and maintenance module**.

----End

Tracking Project Status

As the project progresses, the project team tracks daily work through activities, such as daily Scrums, sprint reviews, and retrospective meetings, to identify problems and continuously improve.

CodeArts Req offers multiple statistical reports and allows you to create custom ones to visualize the project progress.

For details about Scrum project progress, see [Tracking the Progress of a Scrum Project](#).

2.4.3 Developing Code

CodeArts Repo offers Git-based online code management, including code cloning/commit and branch management.

This practice uses branches for code development. Branches let you develop features separately. They keep your work isolated from the main branch, ensuring it stays unaffected.

When a code repository is created, a default branch named **master**, that is, the main line, is generated. To ensure stable running of the **Phoenix Mall**, a stable and continuously available master branch is required. Instead of developing code on the master branch, developers are advised to use "function branches + merge requests". Each function branch must be reviewed by other members in the team before merge.

This section describes how to develop code for the store query function using branches.

Creating a Requirement Branch

Developer Chris creates a requirement branch in the repository to develop code for the new requirement.

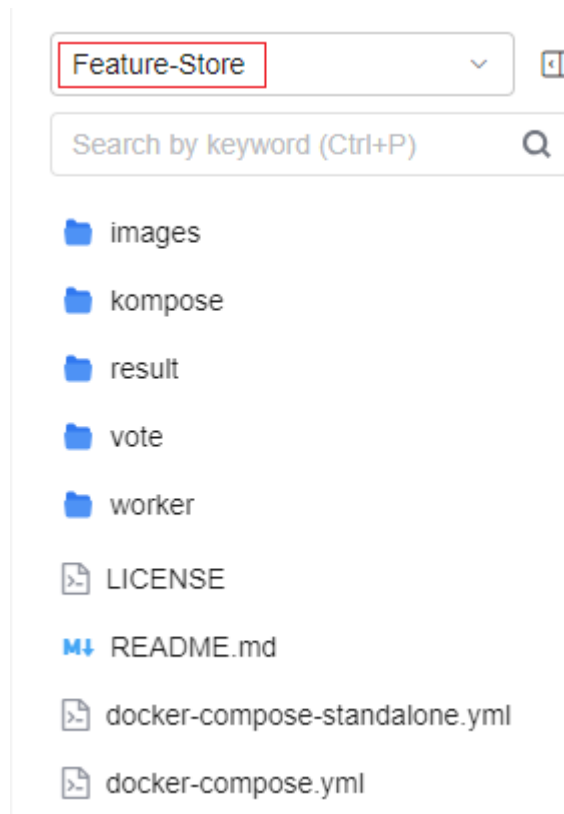
- Step 1** Go to the **Phoenix Mall** project, and choose **Code > Repo** from the navigation pane.
- Step 2** Click the **phoenix-sample** repository.
- Step 3** Choose **Code > Branches** and click **Create Branch**. In the displayed dialog box, set the following parameters and click **OK**.

Table 2-9 Creating a branch

Parameter	Example	Description
Based On	master	Select an existing branch.
Branch Name	Feature-Store	Name of the branch you are creating. The name cannot exceed 200 bytes. <ul style="list-style-type: none">Do not start with a hyphen (-), period (.), refs/heads/, or refs/remotes/.Do not use spaces or these special characters: [<code><~^:?!()''</code>]Do not end with a period (.), slash (/), or .lock.
CodeArts Req Work Items to Associate	Frontend display - Add store network menu	Associate an existing work item with the branch.

The **Files** page is displayed, showing the **Feature-Store** branch.

Figure 2-8 Feature-Store branch




----End

Modifying and Committing Code

Developer Chris develops code in the requirement branch.

Step 1 In the **Feature-Store** branch of the **phoenix-sample** repository, find **vote/templates/store-network.html** and open it.

Step 2 Click , add the following code to the file, enter the commit message "Store list added" in the text box at the bottom of the page, and click **OK**.

```
<ul>
  <li>Branch A: 123 meters to the departure floor, Terminal 1, Airport E</li>
  <li>Branch B: No. 456, Street G, Area F</li>
  <li>Branch C: No. 789, Street J, Area H</li>
  <li>Branch D: West side of Building K, Avenue L, Area K</li>
</ul>
```

The modified code is displayed.

Step 3 Open and edit the **/vote/templates/index.html** file in the same way.

Add the following code to line 179, enter the commit message "fix #xxxxxx Frontend display - Add store network menu", and click **OK**. **xxxxxx** indicates the ID of the task **Frontend display - Add store network menu**.

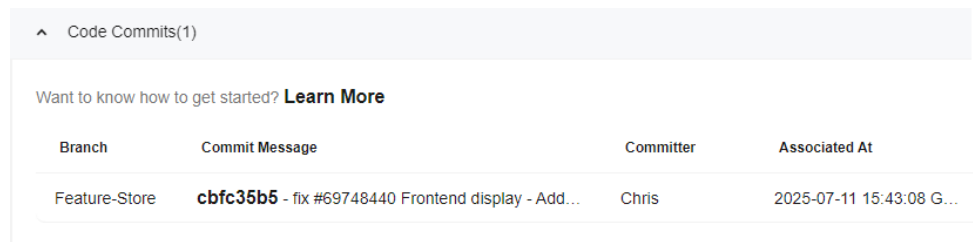
```
<li class="nav-item"> <a href="store-network" class="nav-link">STORE NETWORK</a> </li>
```

The modified code is displayed.

Step 4 Viewing commit records

1. Choose **Work Item > Req** to enter the CodeArts Req service.
2. On the **Work Items** tab, find the task **Frontend display - Add store network menu**. The task status has automatically updated to **Resolved**.
3. Click the task title to open the details page.
4. Click the **Associations** tab. Under **Code Commits**, a record is displayed, showing the commit message configured in **Step 3**.

Figure 2-9 Code commit records



Branch	Commit Message	Committer	Associated At
Feature-Store	cbfc35b5 - fix #69748440 Frontend display - Add...	Chris	2025-07-11 15:43:08 G...

----End

Merging Code

After completing code development and self-testing, developer Chris submits a branch merge request. Project manager Maggie reviews the request and merges the branch.

Step 1 Chris submits a merge request.

1. In the **phoenix-sample** code repository, click the **Merge Requests** tab, and click **Create MR**.
2. Set **Source Branch** to **Feature-Store** and **Target Branch** to **master**.
3. Edit the merge request details by referring to the table below.

Table 2-10 Configuring a merge request

Parameter	Example	Description
Source Branch	Feature-Store	The source branch of the merge request.
Target Branch	master	The target branch of the merge request.
Title	Add store list	The title of the merge request. Enter a maximum of 255 characters.
Mergers	Maggie	Members who will merge the code.
Approvers	Maggie	Members who will review the merge request.

4. Click **OK** to complete the merge request creation. The merge request details are displayed.

Step 2 Maggie reviews the merge request.

1. Go to the **phoenix-sample** code repository, click the **Merge Requests** tab, and find the merge request created by developer Chris.
2. Click the request to view details.
3. Leave comments on the page. Click **Approve** in the **Approval Gate** area.

Figure 2-10 Approval gate



4. Click **Merge** to merge the branch into **master**.
The merge request status is "Merged".

----End

2.4.4 Checking Code

CodeArts Check provides cloud-based code quality management, static code check (quality and style), security check, issue fixing suggestions, and trend analysis.

As Phoenix Mall becomes increasingly larger, many issues have occurred, requiring high fixing costs. However, no unified coding standards are available. The project manager suggests some basic standards, continuous static code scanning, and issue fixing within sprints.

This section describes how developer Chris performs static code scans and identifies and fixes issues for different technology stacks.

Preset Tasks

The sample project presets four code check tasks.

Table 2-11 Preset tasks



Preset Task	Description
phoenix-codecheck-worker	Checks the Worker function code.
phoenix-codecheck-result	Checks the Result function code.
phoenix-codecheck-vote	Checks the Vote function code.
phoenix-sample-javas	Checks the JavaScript code of the entire code repository.

This section uses the **phoenix-codecheck-worker** task as an example.

Configuring and Executing a Task

Developers can slightly adjust preset tasks in the sample project to make the check more comprehensive.

This practice uses the Python check rule set as an example.

- Step 1** Go to the **Phoenix Mall** project, and choose **Code > Check**. The preset four tasks are displayed.
- Step 2** Find the **phoenix-codecheck-worker** task in the list, click **⋮** in the **Operation** column, and choose **Settings**.
- Step 3** In the navigation pane, choose **Rule Sets**. The default language of each rule set is Java.
- Step 4** Add the Python rule set.
1. Click  next to **Languages Included** to refresh the language list.
 2. Enable Python by setting the switch to the  status.
- The rule set is configured.
- Step 5** Click **Execute Check** to start the task.

If **Success** is displayed, the task is successfully executed.

If the task fails, rectify the fault by referring to [CodeArts Check FAQs](#).

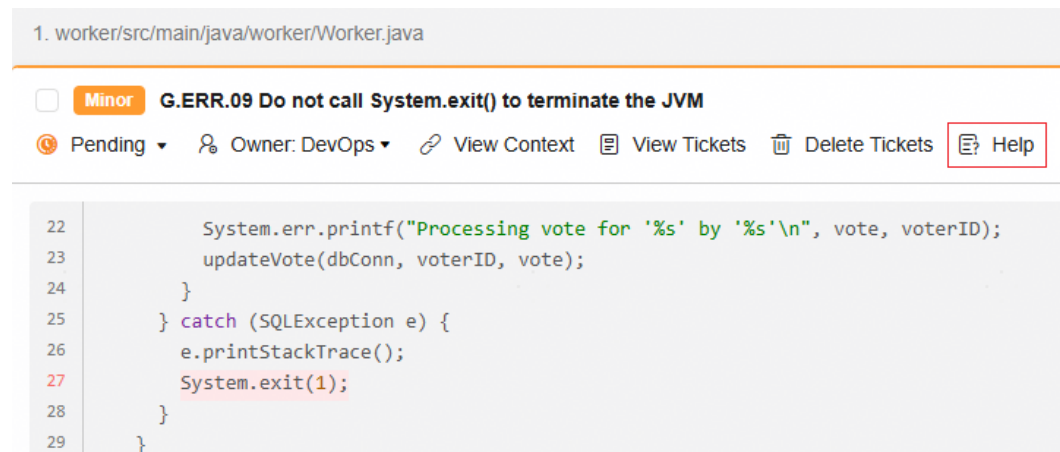
----End

Viewing the Code Check Result

CodeArts Check collects check results and provides fix suggestions for detected issues. Optimize the project code based on the suggestions.

- Step 1** On the task details page, click the **Overview** tab to view the result statistics.
- Step 2** Click the **Issues** tab to view the issue list.

Click **Help** in the question box to view fix suggestions. You can find the corresponding file and code location in the code repository, and optimize the code based on the fix suggestions.

Figure 2-11 Viewing help information

----End

2.4.5 Building Images and Pushing Them to SWR

CodeArts Build provides a hybrid language build platform with simple configurations, supports one-click task creation, configuration, and execution, and automates activities such as code obtaining, building, and packaging.

Deployment failures often occur due to inconsistent environments. For example, the upgraded JDK in the R&D debugging environment is not marked in the environment list. As a result, the production environment is not upgraded accordingly, causing failures. To avoid problems caused by environment inconsistency, we package microservice applications and environments into images in this example to ensure that the environments (development and debugging, test, QA, and production) are consistent.

This section describes how developer Chris builds and archives images to SWR.

Preset Tasks

The sample project presets five build tasks.

Table 2-12 Preset tasks

Preset Task	Description
phoenix-sample-ci	Basic build task
phoenix-sample-ci-test	For building images in the test environment
phoenix-sample-ci-worker	For creating a Worker function image
phoenix-sample-ci-result	For creating a Result function image
phoenix-sample-ci-vote	For creating a Vote function image

This section uses the **phoenix-sample-ci** task as an example. The preset actions and their functions are described in the table below. After configuring the task parameters, you can execute the task.

Table 2-13 Build actions

Build Action	Description
Build Vote Image and Push It to SWR	Create a Vote image using the vote/Dockerfile file, and push the image to SWR for storage.
Build Result Image and Push It to SWR	Create a Result image using the result/Dockerfile file, and push the image to SWR for storage.
Install Worker Dependency Using Maven	Use Maven to install the dependencies required by the Worker function.
Build Worker Image and Push It to SWR	Create a Worker image using the worker/Dockerfile file, and push the image to SWR for storage.
Create Postgres and Redis Dockerfiles	Use shell commands to generate a Dockerfile for creating Postgres (database) and Redis (cache) images.
Build Postgres Image and Push It to SWR	Create a Postgres image using the Dockerfile generated in action Create Postgres and Redis Dockerfiles , and push the image to SWR for storage.
Build Redis Image and Push It to SWR	Create a Redis image using the Dockerfile generated in action Create Postgres and Redis Dockerfiles , and push the image to SWR for storage.
Replace Image Version of Docker-Compose Deployment File	To pull the correct image for deployment on ECS, run the shell commands to perform the following operations: <ul style="list-style-type: none">• Run sed commands to replace the values of parameters in docker-compose-standalone.yml with those of dockerServer, dockerOrg, and BUILDNUMBER in the build task in sequence.• Run the tar command to compress the docker-compose-standalone.yml file into the docker-stack.tar.gz file. Compress the files required for deployment so that the files can be uploaded and archived in subsequent steps.

Build Action	Description
Replace Image Version of Kubernetes Deployment File	To pull the correct image for deployment on CCE, run the shell commands to perform the following operations: <ul style="list-style-type: none">• Run the sed commands to replace the values of docker-server and docker-org in all the files whose names end with deployment in the kompose directory with those of dockerServer and dockerOrg in the build task.• Run the sed commands to replace the value of image-version in result-deployment.yaml, vote-deployment.yaml, and worker-deployment.yaml with the value of BUILDNUMBER.
Upload Kubernetes Deployment File to Release Repo	Upload all .yaml files modified in action Replace Image Version of Kubernetes Deployment File to the release repo for archiving.
Upload Docker-Compose Deployment File to Release Repo	Upload the docker-stack.tar.gz package generated in action Replace Image Version of Docker-Compose Deployment File to the release repo for archiving.

Configuring SWR

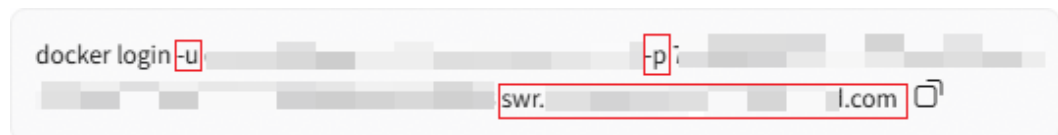
Step 1 Log in to the [SWR console](#).

Step 2 Click **Generate Login Command**. The login command is displayed in a dialog box.

In this command:

- The character string following **-u** is the username.
- The character string following **-p** is the password.
- The remaining string is the SWR server address.

Figure 2-12 Login command



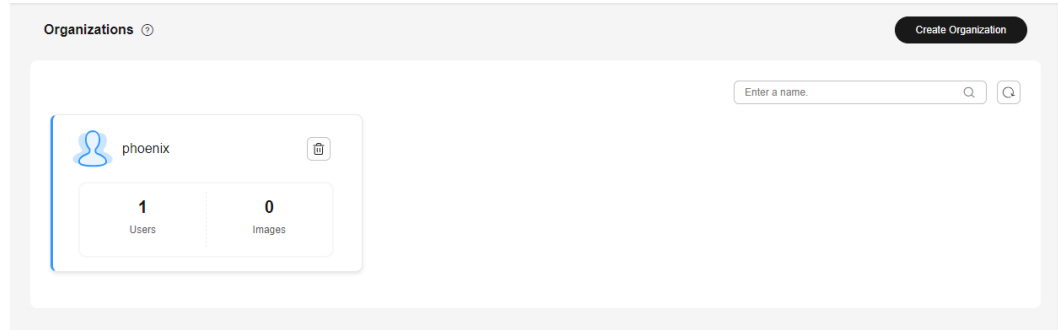
NOTE

The generated login command is valid for 24 hours. For details about how to obtain a long-term valid login command, see [Obtaining a Long-Term Valid Login or Image Push/Pull Command](#).

Step 3 Click **Create Organization**. In the displayed dialog box, enter **phoenix** as the organization name and click **OK**. (If a message is displayed indicating that the organization already exists, enter another name.)

Choose **Organizations** from the navigation pane. The created organization is displayed.

Figure 2-13 Organizations



----End



Configuring and Executing a Task

- Step 1** Go to the **Phoenix Mall** project, and choose **CICD > Build** from the navigation pane.
- Step 2** In the **Operation** column of the **phoenix-sample-ci** task, click ******* and choose **Edit**.
- Step 3** Click the **Parameters** tab, and set default values for the parameters listed in the following table.

Table 2-14 Setting parameters

Name	Default Value
codeBranch	master
dockerOrg	The name of the organization created in Configuring SWR
version	1.0.0
dockerServer	The SWR server address obtained in Configuring SWR

- Step 4** Click **Save and Execute**. In the displayed dialog box, click **OK** to start the build task.

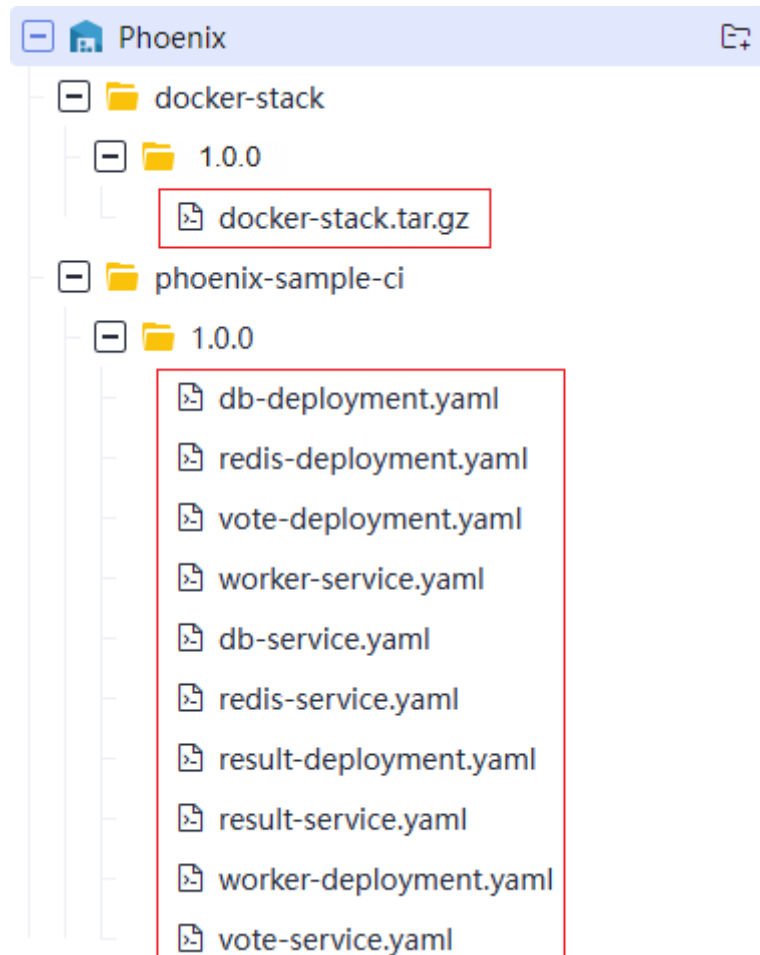
If  is displayed, the task is successfully executed. Record the character string starting with # (for example,  **#20250521.1**).

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

- Step 5** Check the build artifact.
 1. In the navigation pane, choose **Artifact > Release Repos**.
 2. In the repository named after the project, find the **docker-stack** and **phoenix-sample-ci** folders.

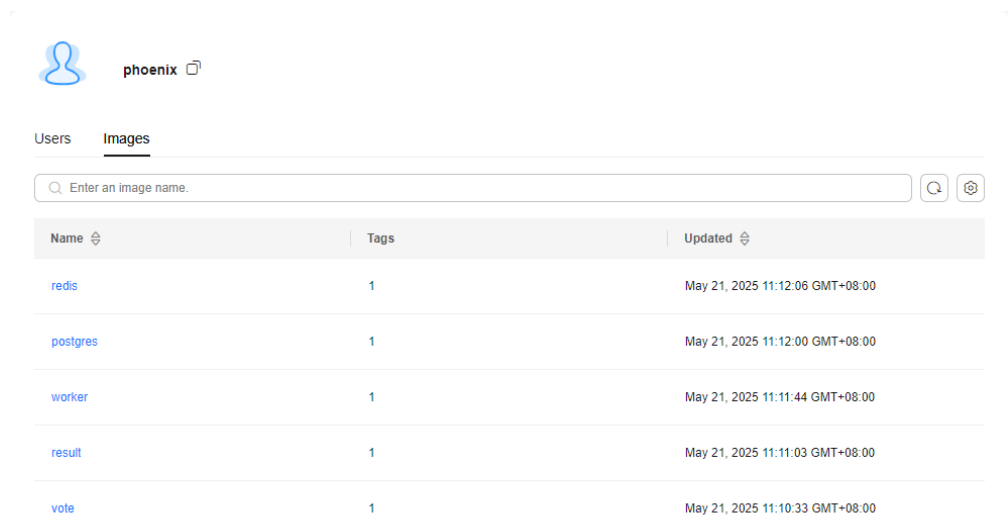
- In the **docker-stack** folder, find the folder named after the character string recorded in **Step 4**. Then find the build artifact **docker-stack.tar.gz** in this folder.
- In the **phoenix-sample-ci/1.0.0** folder, find the 10 archived **.yaml** files.

Figure 2-14 Checking the release repo



3. Go to the SWR console, choose **Organizations** from the navigation pane, and click the name of the organization created in **Configuring SWR**. Click the **Images** tab. Then find the five images (**redis**, **postgres**, **worker**, **result**, and **vote**) in the list.

Figure 2-15 Viewing SWR



4. Click the names of the five images in sequence to go to their details page. View the image tag on the **Tags** tab page.
 - The tag of **redis** is **alpine**.
 - The tag of **postgres** is **9.4**.
 - The tags of **worker**, **result**, and **vote** are the same as those recorded in **Step 4**.

----End

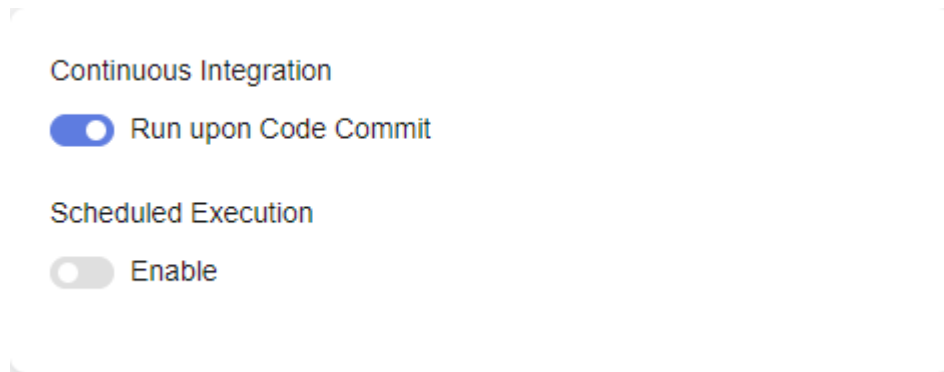
Configuring Auto Compilation upon Code Commits

After the following configuration is complete, code changes will automatically trigger the application build task, achieving continuous integration.

- Step 1** Go to the **Phoenix Mall** project, and choose **CICD > Build** from the navigation pane.
- Step 2** In the **Operation** column of the **phoenix-sample-ci** task, click ******* and choose **Edit**.
- Step 3** Click the **Schedule** tab.
- Step 4** Enable **Run upon Code Commit**. When is displayed, click **Save**.

Since the default value of **codeBranch** on the **Parameters** tab is **master**, the build is automatically run when the code in **master** is changed.

Figure 2-16 Configuring the execution plan



Step 5 Modify the project code and commit it to **master**. Then check whether the build task is automatically executed.

----End

Configuring Scheduled Execution

You can perform the following operations to schedule the build task for a specific time:

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

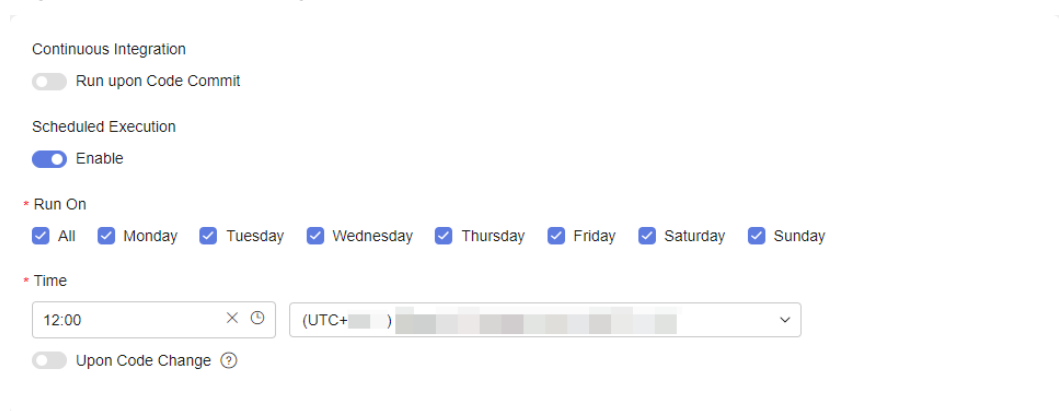
Step 2 On the details page of task **phoenix-sample-ci**, click **Edit**.

Step 3 Click the **Schedule** tab.

Step 4 Enable **Scheduled Execution** ( means enabled), select days and time, disable **Run upon code change**, and click **Save**.

For this example, select **All**, and set the execution time to **12:00**. (This example uses the default time zone. You can change it to the one you use.)

Figure 2-17 Scheduling executions



Step 5 Check whether the build task is automatically executed at the configured time.

----End

2.4.6 Deploying Applications to CCE

CodeArts Deploy provides visualized and automated deployment. Various deployment actions are provided to help you formulate a standard process, reducing deployment costs and improving release efficiency.

To deliver software more quickly and stably, the development team needs automated deployment capabilities to reduce maintenance workload.

This section describes how developer Chris deploys a build artifact on CCE. For details about ECS-based deployment, see [Deploying Applications to ECS](#).

For details about how to prepare CCE, see [Resource Planning](#).

Preset Applications

The sample project presets three deployment applications.

Table 2-15 Preset applications

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

This section uses the **phoenix-cd-cce** application as an example.

Checking Your CCE Cluster

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

- Step 1** Log in to the [CCE console](#).
- Step 2** Find the **phoenix-cce** cluster prepared in [Resource Planning](#) and click the cluster name to access the cluster.
- Step 3** Choose **Workloads** from the navigation pane, click the **Deployments** tab, and verify that no record exists in the list.

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

----End

Configuring and Executing an Application

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

- Step 2** In the **Operation** column of the **phoenix-cd-cce** application, click ******* and choose **Edit**.
- Step 3** On the **Deployment Actions** tab, complete the following configurations in each action.

Table 2-16 Configuring deployment actions

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

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

Table 2-17 Parameters

Name	Default Value
ci_task_name	phoenix-sample-ci
version	1.0.0

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

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







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

----End

Verifying the Deployment Result

- Step 1** Go to the CCE console, and click the **phoenix-cce** cluster.
- Step 2** In the navigation pane, choose **Workloads**. Click the **Deployments** tab, and verify that the namespace displayed in the upper left corner of the page is **default**.
- Five records are displayed on the page. All of them are in the **Running** state.

Figure 2-18 Viewing workloads

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

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

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

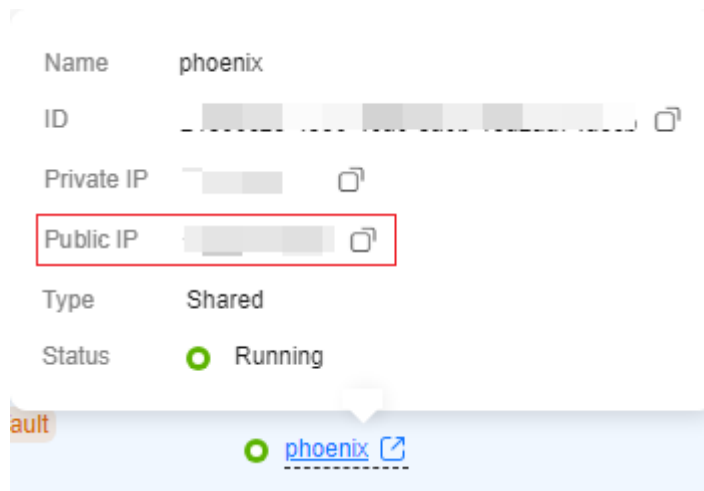
For details about the parameters, see [Creating a LoadBalancer Service](#).

Table 2-18 Updating a Service

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

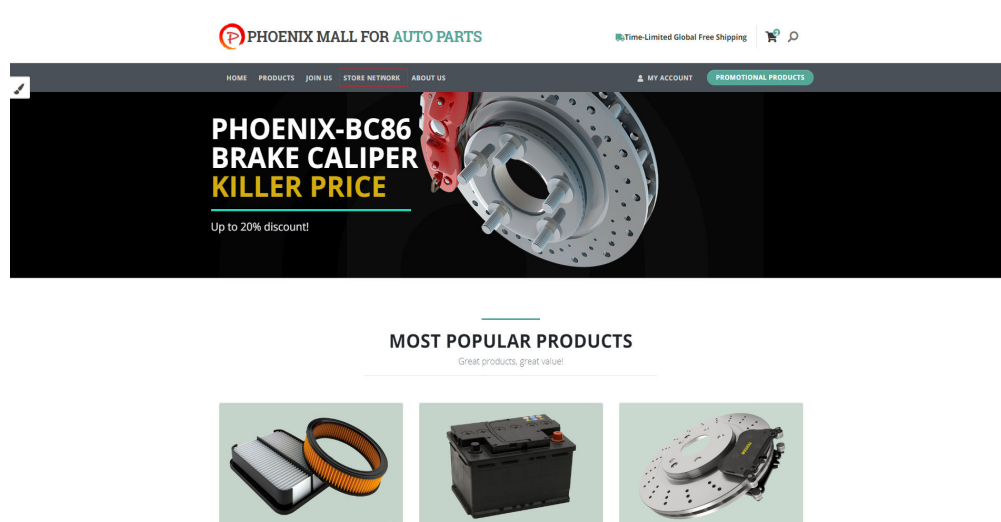
Step 5 Hover over the load balancer name  [phoenix](#) , and copy the public IP address.

Figure 2-19 Copying the access address



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

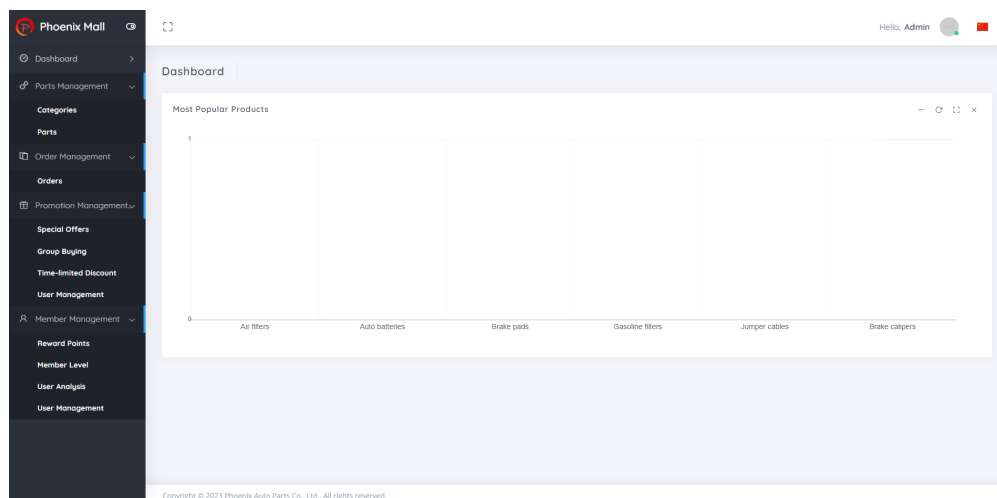
Figure 2-20 Phoenix Mall homepage



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

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

Figure 2-21 Phoenix Mall dashboard



----End

2.4.7 Deploying Applications to ECS

This section uses **phoenix-sample-standalone** as an example to describe how to deploy the build artifact to a host. For details about CCE-based deployment, see [Deploying Applications to CCE](#).

For details about how to prepare ECS, see [Resource Planning](#).

Adding a Target Host to the Project

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

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

Table 2-19 Creating a host cluster

Parameter	Example	Description
Cluster Name	phoenix-hostgroup	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 .

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


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

Step 4 Select **Importing ECS**, and click **Import** in the **Operation** column of the **phoenix-ecs** host in [Resource Planning](#).

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

Table 2-20 Adding a host

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

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

If the host connectivity verification fails, troubleshoot the problem by referring to [How to Troubleshoot Host and Proxy Connectivity Verification Failures?](#)

----End

Installing Dependency Tools on ECS

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

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

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

- Step 3** Click the **Environment Management** tab and configure the host environment.
1. Click **Create Environment**, configure the following information, and click **Save**.

Table 2-21 Creating an environment

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

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

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

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

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

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

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

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

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

Figure 2-22 Viewing deployment logs

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

----End

Configuring and Executing an Application

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

Table 2-22 Creating an environment

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

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

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

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

Table 2-23 Configuring the deployment source

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

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

Step 5 Click the **Parameters** tab, and set default values for the parameters listed in the following table.

Table 2-24 Parameters

Name	Default Value
docker_server	Enter the SWR server address obtained in Configuring the SWR Service .
docker_username	Enter the username obtained in Configuring the SWR Service .
docker_password	Enter the password obtained in Configuring the SWR Service .

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

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

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

Step 7 Verify the deployment result.

1. Open a browser, enter **http://IP:5000** in the address box, and press **Enter**. *IP* indicates the EIP of the ECS. The Phoenix Mall homepage is displayed.
2. Enter **http://IP:5001** and press **Enter**. *IP* indicates the EIP of the ECS. The Phoenix Mall dashboard is displayed.

----End

2.4.8 Managing Project Tests

CodeArts TestPlan provides a one-stop cloud test platform that integrates DevOps agile test concepts, helping to efficiently manage test activities and ensure high-quality product delivery.

This section describes how tester Billy manages the test cycle of a project, including creating plans, and designing and executing cases.

Creating a Test Plan

After the requirement (that is, story) to be implemented in sprint 4 is determined (that is, the operations in [Managing Project Plans](#) are complete), the tester can write test cases while the developer codes.

Step 1 Go to the **Phoenix Mall** project, and choose **Testing > Testing Plan**.

Step 2 Click **Create Plan** and configure test plan information.

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

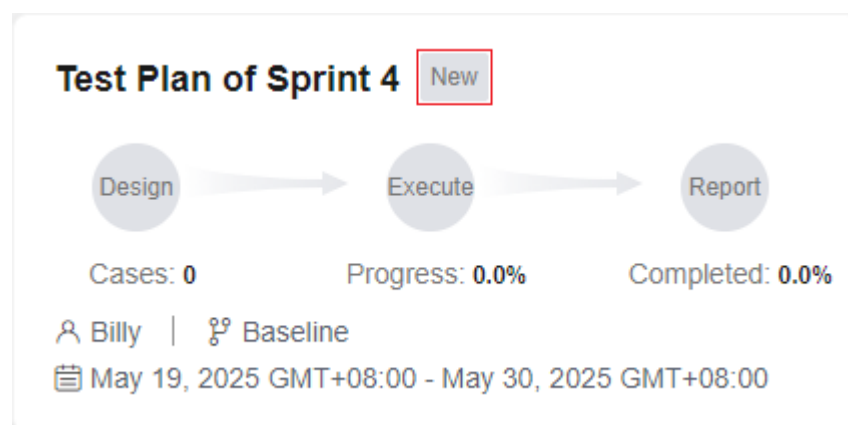
Table 2-25 Basic information about a test plan

Parameter	Example
Name	Test Plan of Sprint4
Processor	Billy
Plan Period	Use the same period as Sprint4 created in Req.
Sprint	Sprint4

2. **Advanced Settings:** Select **Manual Test**. Confirm that the requirement in the list is the same as that of **Sprint4** in CodeArts Req, and then click **Save and Use**.

Step 3 Check the new test plan **Test Plan of Sprint4** in the list. The plan is in the **New** state.

Figure 2-23 Viewing test plan status



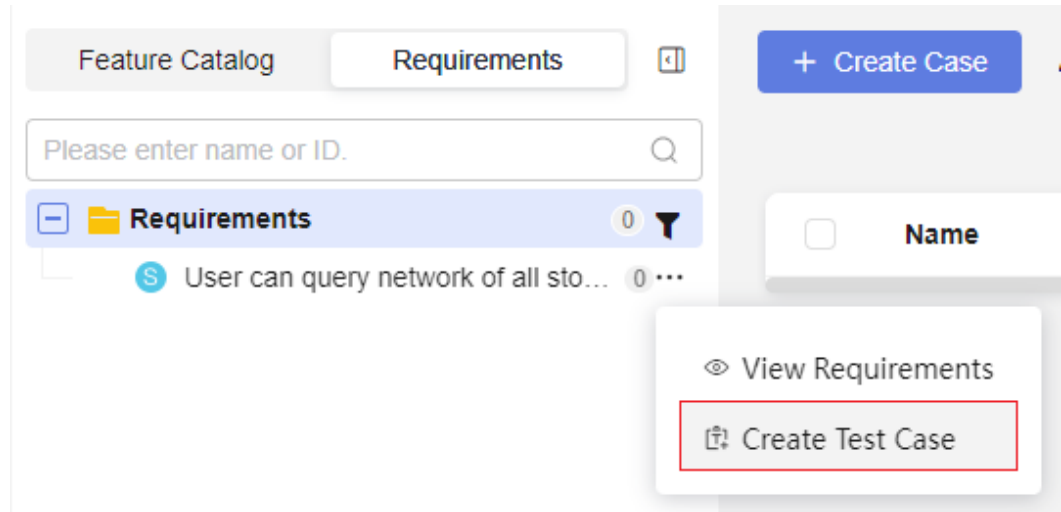
----End

Designing a Test Case

Step 1 In **Test Plan of Sprint4**, click **Design**. The **Testing Case** page is displayed.

- Step 2** Click **Requirements**, and select the story **User can query network of all stores**.
Click **⋮** and choose **Create Test Case**.

Figure 2-24 Creating a test case



- Step 3** Enter **Store Query**, edit the test steps and expected results, and click **Save**.

Table 2-26 Test steps

Test Step	Expected Result
Opening the homepage of Phoenix Mall	The page is displayed.
Clicking the Store Network menu	The Store Network page is displayed. It allows you to filter locations and shows recommended stores at the bottom.
Selecting City A	The store information list of city A is displayed.

- Step 4** Choose **Testing > Testing Plan** from the navigation pane.
Find **Test Plan of Sprint4**. The plan status has changed to **Designing**.
----End

Executing a Test Case

After developing story code and deploying applications to the test environment (see [Deploying Applications to CCE](#) or [Deploying Applications to ECS](#)), developer Chris sets the story status to **Testing** and the story handler to tester Billy.

Then, tester Billy can execute the test case of the story.

Step 1 In **Test Plan of Sprint4**, click **Execute**. The **Testing Execution** page is displayed.

Step 2 Click **Create Suite**, configure the following information, and click **Save**.

- Enter the name **Execute Test Case of Sprint4**.
- Click **Add Case**. In the displayed dialog box, select the created case and click **OK**. The added case is displayed in the list.

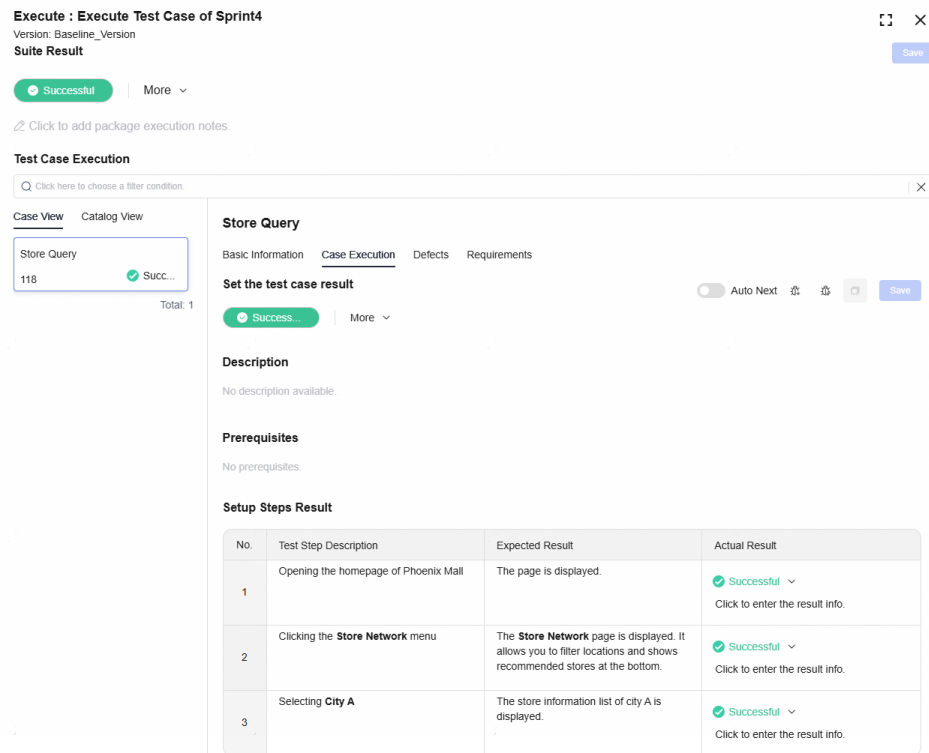
The new suite is displayed in the list.

Step 3 Click **▶** in the **Operation** column. The **Execute** window is displayed on the right.

Step 4 Execute the steps one by one in the test environment.

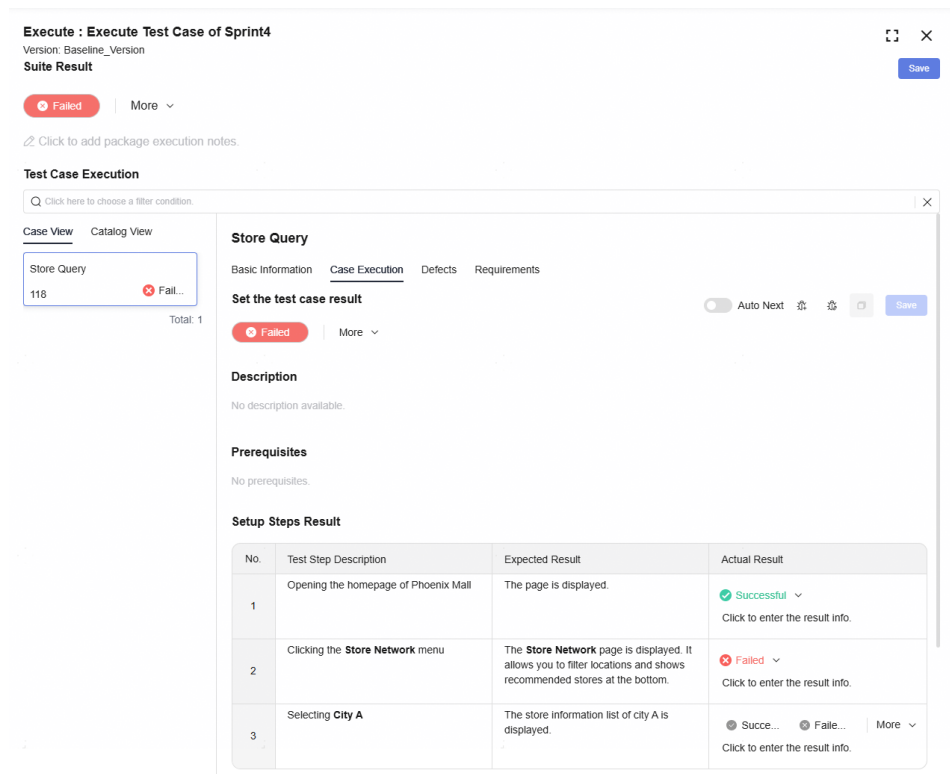
- If the execution is successful, update **Setting Result** to **Successful** for all steps. When a success message shows up, the case execution is complete.


Figure 2-25 Execution successful



- If the execution fails, update **Setting Result** to **Failed** for the failed step. Assume that page redirection fails in step 2. A 404 error occurs.
 - Update **Setting Result** to **Successful** for step 1.
 - Update **Setting Result** to **Failed** for step 2, and enter "Redirection failure. 404 is displayed on the page."
 - Update **Set the test case result** to **Failed**.
 - Update **Suite Result** to **Failed**.

Figure 2-26 Execution failed



- e. After the system displays a message indicating that the update is successful, click .
- f. Enter the title **Store network page 404** in the displayed dialog box, assign the bug to **Chris**, and click **OK**.
- g. After a message indicating that the creation is successful is displayed, return to the test suite execution window and click **Save**.
- h. After developer Chris fixes the bug, verify the case again.

If the verification is successful, set the status of the bug created in [Step 4.f](#) to **Closed**, and update the test case execution result to **Successful**.

Step 5 Choose **Testing > Testing Plan** from the navigation pane.

Find **Test Plan of Sprint4**. The plan status has changed to **Completed**.

----End

Tracking Test Progress

CodeArts TestPlan uses a quality dashboard to visualize various metrics, including requirement coverage, defects (or bugs), case pass rate, and case completion rate.

Your project team can check test plan progress and evaluate product quality on the dashboard. They can also create custom reports as needed.

For more information about tracking test plan progress, see [Viewing and Evaluating Test Quality](#).

2.4.9 Configuring Pipelines

CodeArts Pipeline provides a visualized and customizable software pipeline for automatic delivery. It supports multiple task types, such as code check, build, and deployment tasks.

As the project progresses, the build, release, and deployment become more and more standardized. However, each stage is relatively independent and incomplete and cannot deliver business value directly. Only by effectively connecting each stage to form a complete continuous delivery pipeline can we truly improve the efficiency and quality of software release and continuously create business value.

This section describes how developer **Chris** connects code check, build, and deployment tasks for continuous delivery.

Introduction to Preset Pipelines

There are five pipeline tasks preset in the sample project. You can view and use them as needed.

Table 2-27 Preset pipeline tasks

Preset Pipeline Task	Description
phoenix-workflow	Implements basic functions.
phoenix-workflow-test	Runs in the test environment.
phoenix-workflow-work	Implements the Worker function.
phoenix-workflow-result	Implements the Result function.
phoenix-workflow-vote	Implements the Vote function.

Configuring and Executing a Pipeline

Assume that you use the **phoenix-cd-ccc** application for deployment. Clear the workloads in the cluster before executing a pipeline. For details, see [Checking Your CCE Cluster](#).

- Step 1** Go to the **Phoenix Mall** project, and choose **CICD > Pipeline** from the navigation pane.
- Step 2** In the **Operation** column of the **phoenix-workflow** pipeline, click ******* and choose **Edit**.
- Step 3** Click the **Parameters** tab, and verify that the default values of **dockerOrg** and **dockerServer** are the same as those of the **phoenix-sample-ci** build task.

Step 4 Add a code check stage.



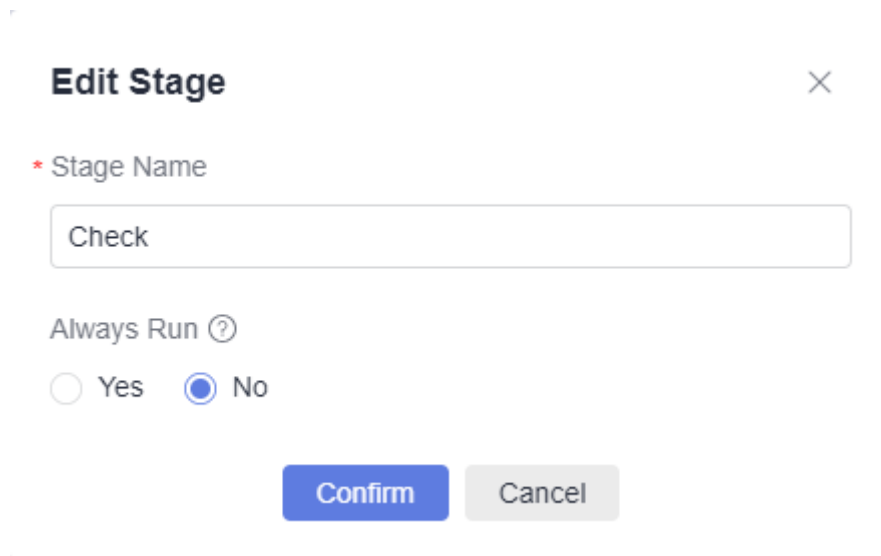
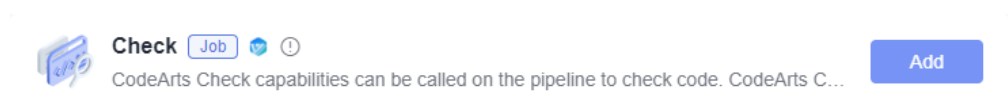
1. Click the **Task Orchestration** tab and click  between **Pipeline Source** and **Build**. A new stage **Stage_1** is displayed.
2. Click  next to **Stage_1**. In the **Edit Stage** window, enter the stage name **Check** and click **Confirm**.

Figure 2-27 Editing the stage name



3. Click **Job**, and select **From Scratch**. The job creation window is displayed on the right.
4. Find the **Check** job in the list, and click **Add**.

Figure 2-28 Adding the Check job



5. Select the **phoenix-codecheck-worker** task and click **OK**.
The **Check** job is displayed.

Step 5 Configure a deployment task.

Click the deployment task name, select the associated build task **phoenix-sample-ci**, and check the values of configuration items.

- The configurations of task **phoenix-sample-standalone** must be the same as those on the **Parameters** page of the task with the same name in CodeArts Deploy.
- The configurations of task **phoenix-cd-cce** must be the same as those on the **Parameters** page of the task with the same name in CodeArts Deploy.

 **NOTE**

Two deployment tasks are added in this example. If you selected only one deployment mode in preceding steps, keep the corresponding task and delete the other one.

Step 6 Click **Save** and then **Execute**. In the displayed dialog box, click **Execute** to start the pipeline.

If  is displayed, the pipeline is successfully executed.

If the pipeline fails, click the cause to view logs. Then rectify the fault by referring to [CodeArts Pipeline FAQs](#).

----End

Configuring Pass Conditions

To control the code quality, the code must be scanned and the number of errors must be within a reasonable range before being released. By adding pass conditions, you can effectively automate the control process.

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

Step 2 In the **Operation** column of the **phoenix-workflow** pipeline, click ******* and choose **Edit**.

Step 3 Click the **Task Orchestration** tab. In the **Check** stage, click **Pass Conditions**. The **Pass Conditions** window is displayed on the right.

Step 4 Click **Add** next to **Pass-Conditions-of-Standard-Policies**.

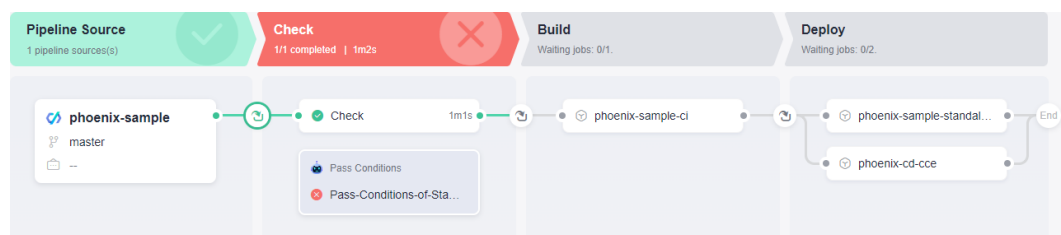
Step 5 Select **SystemPolicy** and click **OK**.

The new pass condition is displayed.

Step 6 Click **Save and Execute**.

If the number of check issues does not meet the pass conditions, the pipeline will fail.

Figure 2-29 Executing a pipeline



----End

Configuring Code Changes to Automatically Trigger a Pipeline

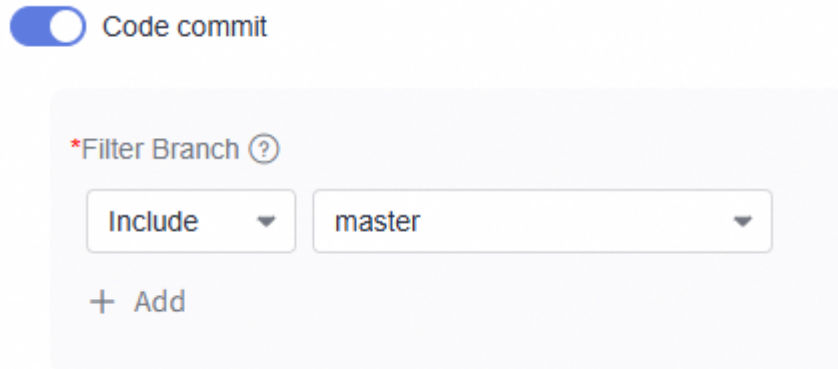
Through the following configuration, code changes can automatically trigger pipeline execution, implementing continuous project delivery.

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

Step 2 In the **Operation** column of the **phoenix-workflow** pipeline, click ******* and choose **Edit**.

Step 3 Click the **Execution Plan** tab, select **Code commit** under **Trigger Settings**, select the **master** branch, and click **Save**.

Figure 2-30 Configuring the execution plan



The modified execution plan is displayed.

Step 4 Modify the code and push it to the **master** branch. Then check whether the pipeline is automatically executed.

----End

2.4.10 Releasing Resources

 **WARNING**

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

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

Table 2-28 Releasing Resources

Resource	Operation Guide
ECS	For details, see Deleting an ECS .
CCE	For details, see Deleting a Pay-per-Use Cluster .

3 Configuring CodeArts Security

CodeArts offers various security features, including operational, code, and continuous delivery security. You can configure them as needed while using CodeArts.

Operational Security

CodeArts provides refined permission control, audit, and tracing to help you keep your data assets secure.

Table 3-1 Operational security

Security Configuration	Description	Suggestion	Reference
Refined permission control	CodeArts has a three-layer permission model for managing tenant-, project-, and instance-level permissions.	Assign permissions to members by adhering to the principle of least privilege.	CodeArts Authentication

Security Configuration	Description	Suggestion	Reference
Audit logs	CodeArts services connect to Cloud Trace Service (CTS) to collect, store, and query operation records.	Enable CTS for security analysis, compliance audit, resource tracing, and troubleshooting.	<ul style="list-style-type: none">• For details about the auditable operations of each service, see:<ul style="list-style-type: none">– CodeArts Req– CodeArts Repo– CodeArts Check– CodeArts Build– CodeArts Deploy– CodeArts Artifact– CodeArts Pipeline– CodeArts TestPlan• For details about how to enable CTS, see CTS Overview.

Security Configuration	Description	Suggestion	Reference
IP address whitelist	CodeArts Repo and CodeArts Artifact allow only whitelisted IP addresses to access data assets such as code and artifacts.	Whitelist trusted IP addresses to prevent unauthorized users and attackers from accessing the system, reducing brute force cracking and DDoS attacks.	<ul style="list-style-type: none"> • Configuring a Code Repo IP Address Whitelist • Configuring an Artifact Repo IP Address Whitelist
Watermark	CodeArts Repo allows you to add a visitor watermark to the source code page, enhancing code security and source tracing.	Enable watermarking to protect the intellectual property rights of your code repos.	Adding Watermarks to a Repository

Code Security

CodeArts Repo provides access tokens, deploy keys, and protected branches to safeguard your code assets.

Table 3-2 Code security

Security Configuration	Description	Suggestion	Reference
Access tokens	CodeArts Repo allows each user to generate access tokens. Tokens are displayed only when generated. You can set the validity period (max. 1 year) of a token. By default, a token is valid for 1 month.	When granting repo access to a third party, create an access token with a specific validity period. Access tokens prevent account and password disclosure.	Configuring an Access Token

Security Configuration	Description	Suggestion	Reference
Deploy keys	CodeArts Repo allows you to add deploy keys for each code repo. Users only have read permissions when accessing a repo using a deploy key.	In code repo reading scenarios, such as builds, use a deploy key to clone a repo to improve code repo security.	Configuring a Deploy Key for a Repository
Protected branches	You can set branch protection rules in a code repo to prevent branches from being modified or mis-deleted.	Set a protection rule for the master branch so that code can only be merged into it via merge requests. Only authorized roles can push code to protected branches.	Configuring Protected Branch Rules
Visibility	CodeArts Repo allows you to set the following visibility options for code repos: <ul style="list-style-type: none"> • Private (A repo can only be read, written, and accessed by its members.) • Public <ul style="list-style-type: none"> - Read-only for project members - Read-only for tenant members - Read-only for all visitors 	Set the visibility when creating a repo or adjust the visibility for an existing repo to scale to your needs. The administrator can determine whether to allow members to create "Public" code repos.	<ul style="list-style-type: none"> • For details about how to set the visibility of a repo, see Creating a Custom Repository. • To set whether to allow members to create "Public" repos, see Adjusting Repository Visibility.
Commit rules	CodeArts Repo control code commits using specific rules. You can use the preconfigured commit rules or create new ones.	Set commit rules for each repo to prevent your code from being modified without permission.	Configuring Commit Rules

Security Configuration	Description	Suggestion	Reference
Deleting a repository	This operation cannot be undone. Please exercise caution. Once you delete a repository, all its content will be permanently deleted.	Before doing this, ensure that all important data has been backed up. If the repository reaches its size limit, consider purging redundant large files or pruning commit history rather than deleting the repository entirely.	Risky Operations on Code Repositories

Continuous Delivery Security

You can keep continuous delivery secure by setting private parameters and configuring host security groups.

Table 3-3 Continuous delivery security

Security Configuration	Description	Suggestion	Reference
Private parameters	CodeArts Build, CodeArts Deploy, CodeArts Pipeline, and CodeArts TestPlan provide private parameter settings. Private parameters are encrypted before storage. They are decrypted before use and are invisible in run logs.	Set parameters that contain sensitive information to private parameters to prevent information leakage.	<ul style="list-style-type: none"> • Configuring Build Parameters • Configuring Parameters of an Application • Configuring Pipeline Parameters • Configuring Sensitive Parameters for Auto API Test Cases

Security Configuration	Description	Suggestion	Reference
Host security groups	<p>When deploying an application with the default CodeArts agent pool, you can use a specified IP address to connect CodeArts Deploy to your hosts to run deployment scripts.</p> <p>To connect CodeArts Deploy to your hosts, configure a security group.</p>	<p>Configure a security group that allows target and proxy hosts to communicate with CodeArts Deploy's default agent pool only through its public IP address.</p>	<p>Configuring a Security Group</p>

4 Configuring CodeArts Permissions

4.1 Overview

CodeArts offers role-based permissions management to facilitate project development from start to finish.

CodeArts presets multiple system roles and allows you to customize roles and permissions. You can save the roles and permissions in a project as a permission template and apply the template to other projects. You can also adjust the role permissions for your projects as needed.

This practice describes how to configure permissions for IPD project members. In addition to reusing the system roles of CodeArts, the practice customizes roles with required permissions. You can refer to this practice to configure permissions for members of different roles in your project, save the permissions as a template, and reuse the template in other projects.

Solution Architecture

The following figure shows the project architecture. There are nine teams in the project team.

Figure 4-1 DevOps role architecture



The responsibilities and required permissions of each role in the preceding architecture are described in the table below.

Table 4-1 Role responsibilities and permissions

Team	Role	Responsibility	Required Permission	Authorizati on
/	Project manager	<ul style="list-style-type: none"> Develops overall project plans. Specifies the responsibilities and tasks of each role and assigns permissions to the roles. 	<ul style="list-style-type: none"> Edit project Set members of all roles and assign them permissions 	Configuring Permissions for the Project Manager Role
R&D team	R&D leader	<ul style="list-style-type: none"> Plans and assigns R&D tasks. Tracks R&D progress, coordinates resources, and solves technical problems during development. Participates in major technical decision-making. 	<ul style="list-style-type: none"> Manage code branches, including but not limited to adding and deleting branches Review code, including but not limited to scoring code 	Configuring Permissions for the R&D Team
	Developer	<ul style="list-style-type: none"> Develops code based on the main branch, including coding, unit testing, bug fixing, and continuous optimization. Participates in technical solution review, document writing, and code review. Continuously optimizes code quality and system performance. 	<ul style="list-style-type: none"> View code repositories Create code branches Create merge requests 	
Middle-end team	R&D leader (middle-end)	<ul style="list-style-type: none"> Designs the architecture of the middle-end system and manages the development. Plans and promotes the reuse of middle-end capabilities, and builds general services that can be shared across business lines. 	<ul style="list-style-type: none"> Manage code branches, including but not limited to adding and deleting branches Review merge requests Assign requirements 	Configuring Permissions for the Middle-end Team
	Developer (middle-end)	<ul style="list-style-type: none"> Develops and maintains middle-end services. Keeps the middle-end system stable, scalable, and reusable. 	<ul style="list-style-type: none"> View code repositories Create code branches Create merge requests 	

Team	Role	Responsibility	Required Permission	Authorizati on
Archite ctur e team	Architecture leader	<ul style="list-style-type: none"> Leads the design and evolution of the technical architecture, and formulates technical specifications and development standards. Streamlines and selects architectures across teams. Optimizes the system for high availability, scalability, performance, and security. 	<ul style="list-style-type: none"> Manage code repositories, including but not limited to adding and deleting branches Review merge requests Assign requirements 	Configuring Permissions for the Architecture Team
	System engineer	<ul style="list-style-type: none"> Participates in architecture design, technical research, and implementation. Writes architecture documents, evaluates system design, and provides technical support. 	<ul style="list-style-type: none"> View code repositories Create code branches Create merge requests 	
Test team	Test manager	<ul style="list-style-type: none"> Establishes the test process and coordinates test resources. Develops test strategies, including functions, performance, and security, and promotes automated testing. 	<ul style="list-style-type: none"> Create test plans Deploy, update, delete, and view test cases Create, modify, and delete mind maps Generate test reports <p>NOTE Test managers do not require edit access to code repositories but must have view access for white-box testing.</p>	Configuring Permissions for the Test Team
	Tester	<ul style="list-style-type: none"> Writes test cases, executes test plans, records bugs, and pushes for bug fixing. Develops test scripts and automated testing tools to enhance product quality. 	<ul style="list-style-type: none"> Maintain test cases Deploy, update, and view test cases Create, modify, and delete mind maps <p>NOTE Testers do not require edit access to code repositories but must have view access for white-box testing.</p>	

Team	Role	Responsibility	Required Permission	Authorizati on
PO team	Product leader	<ul style="list-style-type: none"> • Maintains product features and analyzes requirements. • Assigns tasks to product team members. 	<ul style="list-style-type: none"> • Maintain raw requirements (RRs), initial requirements (IRs), and user stories (USs) • Manage version requirements and feature iterations 	Configuring Permissions for the PO Team
	Product manager	<ul style="list-style-type: none"> • Maintains product features and analyzes RRs and IRs. • Synchronizes requirements between product and R&D teams, and organizes IR and US review meetings. 	Maintain RRs, IRs, and USs	
PM team	Project management leader	<ul style="list-style-type: none"> • Develops overall project plans, tracks progress, and organizes regular meetings. • Coordinates R&D, test, and business resources for the project. 	<ul style="list-style-type: none"> • Maintain test plans 	Configuring Permissions for the PM Team
	Project management team member	<ul style="list-style-type: none"> • Tracks project execution status and writes project documents. • Identifies project risks and develops solutions to keep work on track. 	<ul style="list-style-type: none"> • Maintain test plans 	
Big data team	Big data leader	<ul style="list-style-type: none"> • Builds a big data platform, governs data, and designs and plans the data architecture. • Aligns business data requirements with the platform capabilities, and continuously optimizes the platform. 	<ul style="list-style-type: none"> • Manage code repositories, including but not limited to adding and deleting branches • Review merge requests • Assign requirements 	Configuring Permissions for the Big Data Team

Team	Role	Responsibility	Required Permission	Authorizati on
	Big data product manager	<ul style="list-style-type: none"> Plans and designs data products, and outputs feasible data requirement solutions. Assists business departments in understanding data capabilities, and promotes data-driven decision-making. 	Maintain RRs, IRs, and USs	
	Big data developer	<ul style="list-style-type: none"> Develops data collection, cleansing, modeling, and service APIs. Maintains the data warehouse architecture, and optimizes data processing and query. 	<ul style="list-style-type: none"> View code repositories Create branches Create merge requests 	
O&M team	O&M leader	<ul style="list-style-type: none"> Develops the infrastructure O&M system with O&M processes and emergency plans. Promotes automated O&M, and improves the monitoring system and fault response mechanism. 	Maintain and build pipelines	Configuring Permissions for the O&M Team
	O&M engineer	<ul style="list-style-type: none"> Executes deployment, change, and maintenance tasks, and participates in development of the automated O&M platform. Assists in handling various emergencies during service operation. 	Execute pipelines	
	On-call engineer	<ul style="list-style-type: none"> Handles alarms, inspects systems, and responds to emergencies. Keeps systems run stably, and reports and handles any issues that arise. 	No CodeArts permission required	

Team	Role	Responsibility	Required Permission	Authorizati on
DBA team	DBA leader	<ul style="list-style-type: none"> • Designs the database architecture, plans the capacity, and formulates and implements data security policies. • Develops database standards and review specifications. 	Pipeline-related permissions (for SQL automation)	Configuring Permissions for the DBA Team
	DBA team member	<ul style="list-style-type: none"> • Maintains databases, backs up and restores data, optimizes SQL statements, and troubleshoots faults. • Collaborates with developers on data modeling and performance optimization. 	Pipeline-related permissions	
Security group	Security architect	<ul style="list-style-type: none"> • Designs and implements information security architecture to enhance system, network, and application security. • Identifies and evaluates security risks, and develops mitigation strategies. • Monitors and improves the security architecture to cope with changing security threats and technical environments. 	<ul style="list-style-type: none"> • View code repositories • Create branches • Create merge requests 	Configuring Permissions for the Security Team

Advantages

- **Extended CodeArts role matrix**
This practice extends the system role matrix of CodeArts by adding more roles. For example, in addition to the developer role of CodeArts, you can also add the R&D leader and developer roles to facilitate your project development. You can assign permissions to different roles so that they can complete their tasks.
- **Permission templates**
You can save the project roles and permissions in this practice as a template, and reuse the template in other projects with small changes.

Constraints

- To complete this practice, you must have the **Tenant Administrator** role. After this practice, your project members who have the **DevUC > project-role > privilegeconfig** permission can modify the permission matrix on the **Permissions** page.
For details about how to grant the **Tenant Administrator** role to a user, see [Creating a User Group and Assigning Permissions](#).
- This practice uses an IPD-system device project (currently available in **AP-Singapore**) as an example. Purchase a CodeArts Pro package in advance by referring to [Enabling CodeArts Pro](#).

Concepts

- **Middle-end team:** Builds and maintains a middle-end system to share an enterprise's reusable service capabilities, data, and technical resources with the frontend.
- **Middle-end system:** A key part of an enterprise's IT architecture. It reuses and shares service capabilities, data, and technical resources to improve operational efficiency and service quality. The middle-end system can be divided into multiple layers, such as the business, data, and technical layers. These layers have different functions and roles.

4.2 Implementation Procedure

4.2.1 Preparations


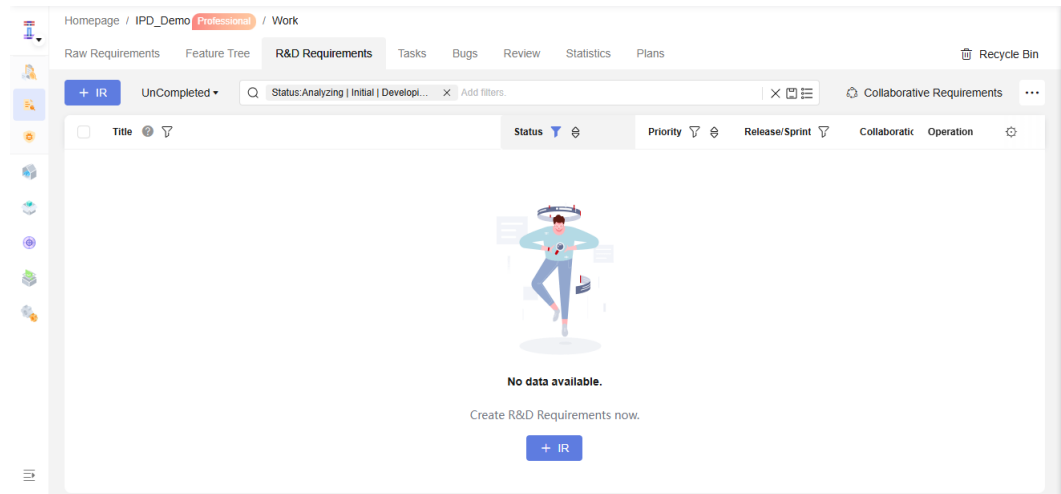
- Step 1** Log in to the [CodeArts console](#), click , and select a region where you have enabled CodeArts Pro.
Click **Go to Workspace** in the upper right corner. The CodeArts homepage is displayed.
- Step 2** Click **Create > Create Project**.
- Step 3** Select **IPD-System Device**, enter the project name **IPD_Demo**, and click **OK**.
The Req service page is automatically displayed.

Figure 4-2 Project created



----End

4.2.2 Configuring Permissions for the Project Manager Role

Step 1 On the **Permissions** page, click the **Project manager** role.

Step 2 Click **Edit**, and configure permissions for the role.

- General: Retain the role permissions, as shown in the following figure.

Figure 4-3 General

Project manager

General			
Resource	Permission		
project	<input type="checkbox"/> edit		<input type="checkbox"/>
project-role	<input checked="" type="checkbox"/> privilegeconfig	<input checked="" type="checkbox"/> userconfig	<input checked="" type="checkbox"/>
Organization Structure	<input type="checkbox"/> Associate Department	<input type="checkbox"/> Disassociate Departm...	<input type="checkbox"/>

- CodeArts Req & Defect: Select all permissions.
- CodeArts Repo: Deselect all permissions.
- CodeArts Check: Deselect all permissions.
- CodeArts TestPlan: Deselect all permissions.
- CodeArts Pipeline: Select permissions by referring to the following figure.

Figure 4-4 CodeArts Pipeline

CodeArts Pipeline						
Resource	Permission					
Change	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> update	<input type="checkbox"/> execute		<input type="checkbox"/>
Development Environment	<input type="checkbox"/> read <input type="checkbox"/> rollback	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	<input type="checkbox"/>
Microservice	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> update		<input type="checkbox"/>
Pipeline	<input checked="" type="checkbox"/> read <input checked="" type="checkbox"/> group	<input checked="" type="checkbox"/> create <input checked="" type="checkbox"/> tag	<input checked="" type="checkbox"/> delete <input checked="" type="checkbox"/> ban	<input checked="" type="checkbox"/> update	<input type="checkbox"/> execute	<input type="checkbox"/>
Policy	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> update		<input checked="" type="checkbox"/>
Pite_production Environment	<input type="checkbox"/> read <input type="checkbox"/> rollback	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	<input type="checkbox"/>
Production Environment	<input type="checkbox"/> read <input type="checkbox"/> rollback	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	<input type="checkbox"/>
Test Environment	<input type="checkbox"/> read <input type="checkbox"/> rollback	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	<input type="checkbox"/>
VariableGroup	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> bind		<input checked="" type="checkbox"/>
ReusableJob	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> update			<input checked="" type="checkbox"/>

- CodeArts Build: Select all permissions.
- CodeArts Deploy: Select all permissions except **Applications > Deploy**, as shown in the following figure.

Figure 4-5 CodeArts Deploy

Resource	Permission
Applications	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> Create <input checked="" type="checkbox"/> Edit <input checked="" type="checkbox"/> Delete <input type="checkbox"/> Deploy <input checked="" type="checkbox"/> Clone <input checked="" type="checkbox"/> Create Environment <input checked="" type="checkbox"/> Assign Permissions <input checked="" type="checkbox"/> Manage Groups <input checked="" type="checkbox"/> Disable
Resource	<input checked="" type="checkbox"/> Create

- CodeArts Artifact: Select all permissions.
- DevMarket (extension marketplace): Select all permissions.

Step 3 Click **Save**.

The configured permissions are displayed.

----End

4.2.3 Configuring Permissions for the R&D Team

Creating an R&D Leader Role and Configuring Permissions

Step 1 On the **Permissions** page, click **Create Role**.

Step 2 In the displayed dialog box, enter the role name **R&D leader** and click **OK**.

The new role is displayed in the custom role list.

Step 3 Click **R&D leader**.

Step 4 Click **Edit**, and configure permissions for the role.

- General: Deselect all permissions.
- CodeArts Req & Defect: Select permissions by referring to the following figure.

Figure 4-6 CodeArts Req & Defect

Resource	Permission
Files	<input type="checkbox"/> upload <input type="checkbox"/> edit <input type="checkbox"/> delete <input type="checkbox"/> download <input type="checkbox"/> preview
Bug	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> View list <input checked="" type="checkbox"/> Create/Clone <input checked="" type="checkbox"/> Edit <input checked="" type="checkbox"/> Upload attachments <input checked="" type="checkbox"/> Add workloads <input type="checkbox"/> Delete/Restore/Delete ... <input checked="" type="checkbox"/> Associate/Disassociat... <input checked="" type="checkbox"/> Associate/Disassociat... <input type="checkbox"/> Migrate <input checked="" type="checkbox"/> Assign <input checked="" type="checkbox"/> Suspend/Cancel susp... <input checked="" type="checkbox"/> Set statuses <input checked="" type="checkbox"/> Associate/Disassociat... <input checked="" type="checkbox"/> Export
Config	<input type="checkbox"/> Basic <input checked="" type="checkbox"/> Manage tags <input checked="" type="checkbox"/> Module <input type="checkbox"/> R&D Downstream Proj... <input type="checkbox"/> Set work types <input type="checkbox"/> Notifications <input type="checkbox"/> Configure work item te... <input type="checkbox"/> Configure workflows <input type="checkbox"/> RR Downstream Proje... <input type="checkbox"/> Automation <input type="checkbox"/> Review
SF	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> View list <input type="checkbox"/> Create/Clone <input type="checkbox"/> Edit <input type="checkbox"/> Upload attachments <input type="checkbox"/> Add workloads <input type="checkbox"/> Delete/Restore/Delete ... <input type="checkbox"/> Set statuses <input type="checkbox"/> Break down/Associat... <input type="checkbox"/> Break down/Associat... <input type="checkbox"/> Create/Associate/Disa... <input type="checkbox"/> Baseline/Cancel baseli... <input type="checkbox"/> Import <input checked="" type="checkbox"/> Export <input type="checkbox"/> Associate/Disassociat... <input type="checkbox"/> Associate/Disassociat... <input checked="" type="checkbox"/> View older versions <input type="checkbox"/> Suspend/Cancel susp...
SF Collection	<input type="checkbox"/> Inherit <input type="checkbox"/> Create <input type="checkbox"/> Update <input type="checkbox"/> Delete <input type="checkbox"/> Baseline Snapshot <input checked="" type="checkbox"/> Read Snapshot <input checked="" type="checkbox"/> Compare Snapshot
IR	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> View list <input checked="" type="checkbox"/> Create/Clone <input checked="" type="checkbox"/> Edit <input checked="" type="checkbox"/> Upload attachments <input checked="" type="checkbox"/> Add workloads <input type="checkbox"/> Delete/Restore/Delete ... <input checked="" type="checkbox"/> Set statuses <input checked="" type="checkbox"/> Break down requireme... <input checked="" type="checkbox"/> Associate/Disassociat... <input checked="" type="checkbox"/> Baseline/Cancel baseli... <input checked="" type="checkbox"/> Assign/Cancel assign... <input checked="" type="checkbox"/> Accept/Reject/Return... <input checked="" type="checkbox"/> Associate/Disassociat... <input checked="" type="checkbox"/> Associate/Disassociat... <input type="checkbox"/> Migrate <input checked="" type="checkbox"/> Import <input checked="" type="checkbox"/> Export <input checked="" type="checkbox"/> Suspend/Cancel susp...
Plan Management	<input type="checkbox"/> Create <input type="checkbox"/> Edit <input type="checkbox"/> Delete <input type="checkbox"/> Baseline/Cancel baseli... <input type="checkbox"/> Set statuses <input type="checkbox"/> Import <input type="checkbox"/> Export <input type="checkbox"/> Baseline Snapshot
Recycle	<input type="checkbox"/> Clear recycle bin
Review	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> View list <input checked="" type="checkbox"/> Edit/Cancel <input type="checkbox"/> Delete <input checked="" type="checkbox"/> Export
RR	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> View list <input type="checkbox"/> Create/Submit/Copy <input type="checkbox"/> Edit <input type="checkbox"/> Upload attachments <input type="checkbox"/> Add workloads <input type="checkbox"/> Delete/Restore/Delete ... <input type="checkbox"/> Break down/Associat... <input type="checkbox"/> Create/Associate/Disa... <input type="checkbox"/> Associate/Disassociat... <input type="checkbox"/> Associate/Disassociat... <input type="checkbox"/> Migrate <input type="checkbox"/> Assign <input type="checkbox"/> Suspend/Cancel susp... <input type="checkbox"/> Set statuses <input type="checkbox"/> Import <input type="checkbox"/> Export
Task	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> View list <input checked="" type="checkbox"/> Create/Clone <input checked="" type="checkbox"/> Edit <input type="checkbox"/> Upload attachments <input type="checkbox"/> Add workloads <input checked="" type="checkbox"/> Delete/Restore/Delete ... <input checked="" type="checkbox"/> Set statuses <input checked="" type="checkbox"/> Break down tasks <input checked="" type="checkbox"/> Associate/Disassociat... <input checked="" type="checkbox"/> Associate/Disassociat... <input checked="" type="checkbox"/> Associate/Disassociat... <input checked="" type="checkbox"/> Associate/Disassociat... <input checked="" type="checkbox"/> Import <input checked="" type="checkbox"/> Associate/Disassociat... <input checked="" type="checkbox"/> Suspend/Cancel susp...

- CodeArts Repo: Select all permissions except **repository > delete**, as shown in the following figure.

Figure 4-7 CodeArts Repo

Resource	Permission	
branch	<input checked="" type="checkbox"/> create <input checked="" type="checkbox"/> delete	<input type="checkbox"/>
code	<input checked="" type="checkbox"/> push <input checked="" type="checkbox"/> download	<input checked="" type="checkbox"/>
group	<input checked="" type="checkbox"/> create <input checked="" type="checkbox"/> delete <input checked="" type="checkbox"/> setting	<input checked="" type="checkbox"/>
member	<input checked="" type="checkbox"/> create <input checked="" type="checkbox"/> update <input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/>
mr	<input checked="" type="checkbox"/> create <input checked="" type="checkbox"/> merge <input checked="" type="checkbox"/> update <input checked="" type="checkbox"/> close <input checked="" type="checkbox"/> comment <input checked="" type="checkbox"/> reopen <input checked="" type="checkbox"/> review <input checked="" type="checkbox"/> approve	<input checked="" type="checkbox"/>
repository	<input checked="" type="checkbox"/> create <input checked="" type="checkbox"/> fork <input type="checkbox"/> delete <input checked="" type="checkbox"/> setting	<input type="checkbox"/>
tag	<input checked="" type="checkbox"/> create <input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/>

- CodeArts Check: Select permissions by referring to the following figure.

Figure 4-8 CodeArts Check

Resource	Permission	
CodeCheck_Report	<input checked="" type="checkbox"/> read <input checked="" type="checkbox"/> edit	<input checked="" type="checkbox"/>
CodeCheck_Rule_Set	<input type="checkbox"/> setDefault	<input type="checkbox"/>
CodeCheck_Task	<input type="checkbox"/> create <input checked="" type="checkbox"/> execute <input checked="" type="checkbox"/> read <input type="checkbox"/> edit <input type="checkbox"/> delete	<input type="checkbox"/>

- CodeArts TestPlan: Select permissions by referring to the following figure.

Figure 4-9 CodeArts TestPlan

Resource	Permission	
Test Case	<input type="checkbox"/> create <input checked="" type="checkbox"/> read <input type="checkbox"/> update <input type="checkbox"/> delete <input type="checkbox"/> exportToFile <input type="checkbox"/> importFromFile <input type="checkbox"/> executeFromVersion <input type="checkbox"/> stop <input type="checkbox"/> mergeToMaster	<input type="checkbox"/>
Test Suite	<input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete <input checked="" type="checkbox"/> read <input type="checkbox"/> execute <input type="checkbox"/> stop	<input type="checkbox"/>
Test Version	<input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete	<input checked="" type="checkbox"/> read
Test Plan	<input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete	<input checked="" type="checkbox"/> read
Test Quality Dashboard	<input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete	<input checked="" type="checkbox"/> read
Test Quality Evaluation	<input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete	<input checked="" type="checkbox"/> read <input checked="" type="checkbox"/> download
Test Case Recycle Bin	<input checked="" type="checkbox"/> delete <input checked="" type="checkbox"/> recover <input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/>
Testing Settings	<input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete	<input type="checkbox"/> read
Key Word	<input type="checkbox"/> create <input checked="" type="checkbox"/> read <input type="checkbox"/> delete <input type="checkbox"/> update	<input type="checkbox"/>
Global Variable	<input type="checkbox"/> create <input checked="" type="checkbox"/> read <input type="checkbox"/> delete <input type="checkbox"/> update	<input type="checkbox"/>
Echo Test Task	<input checked="" type="checkbox"/> read <input type="checkbox"/> stop <input type="checkbox"/> create <input type="checkbox"/> configureAdvancedSet	<input type="checkbox"/> update <input type="checkbox"/> execute
Echo Test Alert Configuration	<input checked="" type="checkbox"/> read <input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> update	<input type="checkbox"/>
Echo Test Alert Notification	<input checked="" type="checkbox"/> read <input type="checkbox"/> ignore	<input type="checkbox"/>
Echo Test Dashboard	<input checked="" type="checkbox"/> read <input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> update	<input type="checkbox"/>
Mindmap	<input checked="" type="checkbox"/> read <input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> update <input type="checkbox"/> deleteAll <input type="checkbox"/> updateAll	<input type="checkbox"/>
Mindmap Backup	<input checked="" type="checkbox"/> read <input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> recover <input type="checkbox"/> deleteAll <input type="checkbox"/> recoverAll	<input type="checkbox"/>
Mindmap Template	<input checked="" type="checkbox"/> read <input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> update <input type="checkbox"/> deleteAll <input type="checkbox"/> updateAll	<input type="checkbox"/>
Recycle	<input checked="" type="checkbox"/> read <input type="checkbox"/> delete <input type="checkbox"/> recover	<input type="checkbox"/>
Mindmap Asset	<input checked="" type="checkbox"/> read <input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> update <input type="checkbox"/> import <input type="checkbox"/> export	<input type="checkbox"/>
Service Group	<input type="checkbox"/> create <input checked="" type="checkbox"/> read <input type="checkbox"/> update <input type="checkbox"/> delete	<input type="checkbox"/>
Service	<input type="checkbox"/> create <input checked="" type="checkbox"/> read <input type="checkbox"/> update <input type="checkbox"/> delete	<input type="checkbox"/>
Interface	<input type="checkbox"/> create <input checked="" type="checkbox"/> read <input type="checkbox"/> update <input type="checkbox"/> delete <input type="checkbox"/> copy <input type="checkbox"/> import	<input type="checkbox"/>
Rule	<input type="checkbox"/> create <input checked="" type="checkbox"/> read <input type="checkbox"/> update <input type="checkbox"/> delete	<input type="checkbox"/>
Instance	<input type="checkbox"/> deploy <input type="checkbox"/> start <input type="checkbox"/> stop <input type="checkbox"/> update <input type="checkbox"/> delete <input checked="" type="checkbox"/> read	<input type="checkbox"/>

- CodeArts Pipeline: Select the **Pipeline > read**, **Pipeline > execute**, and **Policy > read** permissions, as shown in the following figure.

Figure 4-10 CodeArts Pipeline

Resource	Permission						
Change	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> execute			
Development Environment	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	<input type="checkbox"/> rollback	
Microservice	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> update			
Pipeline	<input checked="" type="checkbox"/> read <input type="checkbox"/> tag	<input type="checkbox"/> create <input type="checkbox"/> ban	<input type="checkbox"/> delete	<input type="checkbox"/> update	<input checked="" type="checkbox"/> execute	<input type="checkbox"/> group	
Policy	<input checked="" type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> update			
Pre_production Environment	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	<input type="checkbox"/> rollback	
Production Environment	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	<input type="checkbox"/> rollback	
Test Environment	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	<input type="checkbox"/> rollback	
VariableGroup	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> update	<input type="checkbox"/> bind			
ReusableJob	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> update				

- CodeArts Build: Select the **view** and **execute** permissions, as shown in the following figure.

Figure 4-11 CodeArts Build

Resource	Permission					
job	<input type="checkbox"/> create	<input type="checkbox"/> modify	<input type="checkbox"/> delete	<input checked="" type="checkbox"/> view	<input checked="" type="checkbox"/> execute	<input type="checkbox"/> copy
	<input type="checkbox"/> forbidden	<input type="checkbox"/> manager	<input type="checkbox"/> group			

- CodeArts Deploy: Select the **Applications > View** and **Deploy** permissions, as shown in the following figure.

Figure 4-12 CodeArts Deploy

Resource	Permission						
Applications	<input checked="" type="checkbox"/> View	<input type="checkbox"/> Create	<input type="checkbox"/> Edit	<input type="checkbox"/> Delete	<input checked="" type="checkbox"/> Deploy	<input type="checkbox"/> Clone	
	<input type="checkbox"/> Create Environment	<input type="checkbox"/> Assign Permissions	<input type="checkbox"/> Manage Groups	<input type="checkbox"/> Disable			
Resource	<input type="checkbox"/> Create						

- CodeArts Artifact: Select permissions by referring to the following figure.

Figure 4-13 CodeArts Artifact

Resource	Permission									
artifact_gsm	<input checked="" type="checkbox"/> changePkgStatus	<input checked="" type="checkbox"/> upload	<input type="checkbox"/> delOrRestoreTestPkg	<input type="checkbox"/> delOrRestoreProdPkg	<input checked="" type="checkbox"/> editTestPkg	<input checked="" type="checkbox"/> createFolder				
	<input checked="" type="checkbox"/> download	<input type="checkbox"/> restoreAll	<input type="checkbox"/> ckAllInRecycleBin							
artifact_pt	<input checked="" type="checkbox"/> createRepository	<input checked="" type="checkbox"/> editRepository	<input type="checkbox"/> deleteRepository	<input type="checkbox"/> restore	<input type="checkbox"/> physicdelete	<input type="checkbox"/> restoreall				
	<input type="checkbox"/> clearall									
component	<input checked="" type="checkbox"/> upload	<input checked="" type="checkbox"/> downloadOrView	<input checked="" type="checkbox"/> deleteOrRedeploy							<input checked="" type="checkbox"/>

- DevMarket (extension marketplace): Select the **read** permission, as shown in the following figure.

Figure 4-14 DevMarket

Resource	Permission				
EndPoaint	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input checked="" type="checkbox"/> read	<input type="checkbox"/>

Step 5 Click **Save**.

The configured permissions are displayed.

----End

Configuring Permissions for the Developer Role

- Step 1** On the **Permissions** page, click the **Developer** role.
- Step 2** Click **Edit**, and configure permissions for the role.
 - CodeArts Repo: Select permissions by referring to the following figure.

Figure 4-15 CodeArts Repo

Resource	Permission	
branch	<input checked="" type="checkbox"/> create <input type="checkbox"/> delete	<input type="checkbox"/>
code	<input checked="" type="checkbox"/> push <input checked="" type="checkbox"/> download	<input checked="" type="checkbox"/>
group	<input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> setting	<input type="checkbox"/>
member	<input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete	<input type="checkbox"/>
mr	<input checked="" type="checkbox"/> create <input checked="" type="checkbox"/> update <input checked="" type="checkbox"/> comment <input checked="" type="checkbox"/> review <input type="checkbox"/> approve <input type="checkbox"/> merge <input checked="" type="checkbox"/> close <input checked="" type="checkbox"/> reopen	<input type="checkbox"/>
repository	<input type="checkbox"/> create <input type="checkbox"/> fork <input type="checkbox"/> delete <input type="checkbox"/> setting	<input type="checkbox"/>
tag	<input type="checkbox"/> create <input type="checkbox"/> delete	<input type="checkbox"/>

- Keep other permissions the same as those of the **R&D leader** role.

Step 3 Click **Save**.

The configured permissions are displayed.

----End

4.2.4 Configuring Permissions for the Middle-end Team

Creating an R&D Leader (Middle-end) Role and Configuring Permissions

- Step 1** On the **Permissions** page, click **Create Role**.
- Step 2** In the displayed dialog box, enable **Copy Role**, set the following parameters, and click **OK**.

Table 4-2 Creating a role

Parameter	Example
Role Name	R&D leader (middle-end)
Project	IPD_Demo
Role	Custom Role > R&D leader

The new role is displayed in the custom role list.

Keep the permissions the same as those of the **R&D leader** role, and do not make any changes.

----End

Creating a Developer (Middle-end) Role and Configuring Permissions

- Step 1** On the **Permissions** page, click **Create Role**.

- Step 2** In the displayed dialog box, enable **Copy Role**, set the following parameters, and click **OK**.

Table 4-3 Creating a role

Parameter	Example
Role Name	Developer (middle-end)
Project	IPD_Demo
Role	System Role > Developer

The new role is displayed in the custom role list.

Keep the permissions the same as those of the **Developer** role, and do not make any changes.

----End

4.2.5 Configuring Permissions for the Architecture Team

Creating an Architecture Leader Role and Configuring Permissions

- Step 1** On the **Permissions** page, click **Create Role**.
- Step 2** In the displayed dialog box, enable **Copy Role**, set the following parameters, and click **OK**.

Table 4-4 Creating a role

Parameter	Example
Role Name	Architecture leader
Project	IPD_Demo
Role	Custom Role > R&D leader

The new role is displayed in the custom role list.

Keep the permissions the same as those of the **R&D leader** role, and do not make any changes.

----End

Configuring Permissions for the System Engineer Role

- Step 1** On the **Permissions** page, click the **System engineer** role, and click **Edit**.
- Step 2** Configure permissions for the role by referring to [Configuring Permissions for the Developer Role](#).

Step 3 Click **Save**.

The configured permissions are displayed.

----End

4.2.6 Configuring Permissions for the Test Team

Configuring Permissions for the Test Manager Role

Step 1 On the **Permissions** page, click the **Test manager** role.

Step 2 Click **Edit**, and configure permissions for the test manager.

- General: Deselect all permissions.
- CodeArts Req & Defect: Select permissions by referring to the following figure.

Figure 4-16 CodeArts Req & Defect

Resource	Permission	
Files	upload	<input type="checkbox"/>
	edit	<input type="checkbox"/>
	delete	<input type="checkbox"/>
	download	<input type="checkbox"/>
	preview	<input type="checkbox"/>
Bug	View	<input checked="" type="checkbox"/>
	Add workloads	<input checked="" type="checkbox"/>
	Migrate	<input checked="" type="checkbox"/>
	View list	<input checked="" type="checkbox"/>
	Delete/Restore/Delete	<input checked="" type="checkbox"/>
	Assign	<input checked="" type="checkbox"/>
	Create/Clone	<input checked="" type="checkbox"/>
	Associate/Dissociat	<input checked="" type="checkbox"/>
	Suspend/Cancel susp...	<input checked="" type="checkbox"/>
	Edit	<input checked="" type="checkbox"/>
	Associate/Dissociat...	<input checked="" type="checkbox"/>
	Set statuses	<input checked="" type="checkbox"/>
	Import	<input checked="" type="checkbox"/>
	Export	<input checked="" type="checkbox"/>
Config	Basic	<input type="checkbox"/>
	Notifications	<input type="checkbox"/>
	Manage tags	<input checked="" type="checkbox"/>
	Configure work item ta...	<input type="checkbox"/>
	Module	<input checked="" type="checkbox"/>
	Configure workflows	<input type="checkbox"/>
	R&D Downstream Proje...	<input type="checkbox"/>
	RR Downstream Proje...	<input type="checkbox"/>
	Set work types	<input type="checkbox"/>
	Automation	<input type="checkbox"/>
SF	View	<input checked="" type="checkbox"/>
	Add workloads	<input checked="" type="checkbox"/>
	Create/Associate/Disa...	<input type="checkbox"/>
	Associate/Dissociat...	<input type="checkbox"/>
	View older versions	<input checked="" type="checkbox"/>
	View list	<input checked="" type="checkbox"/>
	Delete/Restore/Delete	<input type="checkbox"/>
	Baseline/Cancel basel...	<input type="checkbox"/>
	Import	<input type="checkbox"/>
	Suspend/Cancel susp...	<input type="checkbox"/>
SF Collection	Inherit	<input type="checkbox"/>
	Read Snapshot	<input checked="" type="checkbox"/>
	Create	<input type="checkbox"/>
	Compare Snapshot	<input checked="" type="checkbox"/>
	Update	<input type="checkbox"/>
	Delete	<input type="checkbox"/>
	Baseline Snapshot	<input type="checkbox"/>
IR	View	<input checked="" type="checkbox"/>
	Add workloads	<input checked="" type="checkbox"/>
	Baseline/Cancel basel...	<input type="checkbox"/>
	Migrate	<input type="checkbox"/>
	View list	<input checked="" type="checkbox"/>
	Delete/Restore/Delete	<input checked="" type="checkbox"/>
	Assign/Cancel assign...	<input type="checkbox"/>
	Import	<input checked="" type="checkbox"/>
	Export	<input checked="" type="checkbox"/>
	Create/Clone	<input type="checkbox"/>
	Set statuses	<input checked="" type="checkbox"/>
	Accept/Reject/Return...	<input type="checkbox"/>
	Suspend/Cancel susp...	<input type="checkbox"/>
	Edit	<input type="checkbox"/>
	Baseline/Cancel basel...	<input type="checkbox"/>
	Set statuses	<input type="checkbox"/>
Plan Management	Create	<input type="checkbox"/>
	Import	<input type="checkbox"/>
	Edit	<input type="checkbox"/>
	Export	<input type="checkbox"/>
	Delete	<input type="checkbox"/>
	Baseline Snapshot	<input type="checkbox"/>
	Baseline/Cancel basel...	<input type="checkbox"/>
Recycle	Clear recycle bin	<input type="checkbox"/>
Review	View	<input type="checkbox"/>
	View list	<input type="checkbox"/>
	Edit/Cancel	<input type="checkbox"/>
	Delete	<input type="checkbox"/>
	Export	<input type="checkbox"/>
RR	View	<input checked="" type="checkbox"/>
	Add workloads	<input checked="" type="checkbox"/>
	Associate/Dissociat...	<input type="checkbox"/>
	Import	<input type="checkbox"/>
	View list	<input checked="" type="checkbox"/>
	Delete/Restore/Delete	<input type="checkbox"/>
	Migrate	<input type="checkbox"/>
	Export	<input type="checkbox"/>
	Create/Submit/Copy	<input type="checkbox"/>
	Break down/Associate...	<input type="checkbox"/>
	Assign	<input type="checkbox"/>
	Create/Associate/Disa...	<input type="checkbox"/>
	Suspend/Cancel susp...	<input type="checkbox"/>
	Edit	<input type="checkbox"/>
	Associate/Dissociat...	<input type="checkbox"/>
	Set statuses	<input type="checkbox"/>
Task	View	<input checked="" type="checkbox"/>
	Add workloads	<input checked="" type="checkbox"/>
	Associate/Dissociat...	<input checked="" type="checkbox"/>
	View list	<input checked="" type="checkbox"/>
	Delete/Restore/Delete	<input checked="" type="checkbox"/>
	Associate/Dissociat...	<input checked="" type="checkbox"/>
	Create/Clone	<input checked="" type="checkbox"/>
	Set statuses	<input checked="" type="checkbox"/>
	Associate/Dissociat...	<input checked="" type="checkbox"/>
	Edit	<input checked="" type="checkbox"/>
	Break down tasks	<input checked="" type="checkbox"/>
	Associate/Dissociat...	<input checked="" type="checkbox"/>
	Export	<input checked="" type="checkbox"/>
	Suspend/Cancel susp...	<input checked="" type="checkbox"/>

- CodeArts Repo: Deselect all permissions.
- CodeArts Check: Select permissions by referring to the following figure.

Figure 4-17 CodeArts Check

Resource	Permission	
CodeCheck_Report	read	<input checked="" type="checkbox"/>
	edit	<input type="checkbox"/>
CodeCheck_Rule_Set	setDefault	<input type="checkbox"/>
CodeCheck_Task	create	<input type="checkbox"/>
	execute	<input checked="" type="checkbox"/>
	read	<input checked="" type="checkbox"/>
	edit	<input type="checkbox"/>
	delete	<input type="checkbox"/>

- CodeArts TestPlan: Select all permissions.
- CodeArts Pipeline: Select permissions by referring to the following figure.

Figure 4-18 CodeArts Pipeline

Resource	Permission					
Change	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> execute		
Development Environment	<input type="checkbox"/> read <input type="checkbox"/> rollback	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	
Microservice	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> update		
Pipeline	<input checked="" type="checkbox"/> read <input type="checkbox"/> group	<input type="checkbox"/> create <input type="checkbox"/> tag	<input type="checkbox"/> delete <input type="checkbox"/> ban	<input type="checkbox"/> update	<input checked="" type="checkbox"/> execute	
Policy	<input checked="" type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> update		
Pre_production Environment	<input type="checkbox"/> read <input type="checkbox"/> rollback	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	
Production Environment	<input type="checkbox"/> read <input type="checkbox"/> rollback	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	
Test Environment	<input type="checkbox"/> read <input type="checkbox"/> rollback	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	
VariableGroup	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> update	<input type="checkbox"/> bind		
ReusableJob	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> update			

- CodeArts Build: Select the **view** and **execute** permissions, as shown in the following figure.

Figure 4-19 CodeArts Build

Resource	Permission					
job	<input type="checkbox"/> create <input type="checkbox"/> copy	<input type="checkbox"/> modify <input type="checkbox"/> forbidden	<input type="checkbox"/> delete <input type="checkbox"/> manager	<input checked="" type="checkbox"/> view <input type="checkbox"/> group	<input checked="" type="checkbox"/> execute	

- CodeArts Deploy: Select the **Applications > View** and **Deploy** permissions, as shown in the following figure.

Figure 4-20 CodeArts Deploy

Resource	Permission					
Applications	<input checked="" type="checkbox"/> View <input type="checkbox"/> Clone	<input type="checkbox"/> Create <input type="checkbox"/> Create Environment	<input type="checkbox"/> Edit <input type="checkbox"/> Assign Permissions	<input type="checkbox"/> Delete <input type="checkbox"/> Manage Groups	<input checked="" type="checkbox"/> Deploy <input type="checkbox"/> Disable	
Resource	<input type="checkbox"/> Create					

- CodeArts Artifact: Select permissions by referring to the following figure.

Figure 4-21 CodeArts Artifact

Resource	Permission					
artifact_gene	<input checked="" type="checkbox"/> changePkgStatus <input checked="" type="checkbox"/> createFolder	<input checked="" type="checkbox"/> upload <input checked="" type="checkbox"/> download	<input type="checkbox"/> delOrRestoreTestPkg <input type="checkbox"/> restoreAll	<input type="checkbox"/> delOrRestoreProdPkg <input type="checkbox"/> clrAllInRecycleBin	<input checked="" type="checkbox"/> editTestPkg	
artifact_pri	<input checked="" type="checkbox"/> createRepository <input type="checkbox"/> restoreall	<input checked="" type="checkbox"/> editRepository <input type="checkbox"/> clearall	<input type="checkbox"/> deleteRepository	<input type="checkbox"/> restore	<input type="checkbox"/> physicdelete	
component	<input checked="" type="checkbox"/> upload	<input checked="" type="checkbox"/> downloadOrView	<input checked="" type="checkbox"/> deleteOrRedeploy			<input checked="" type="checkbox"/>

- DevMarket (extension marketplace): Select the **read** permission.

Figure 4-22 DevMarket

Resource	Permission				
EndPoint	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input checked="" type="checkbox"/> read	

Step 3 Click **Save**.

The configured permissions are displayed.

----End

Configuring Permissions for the Tester Role

Step 1 On the **Permissions** page, click the **Tester** role.

Step 2 Click **Edit**, and configure permissions for the role.

- CodeArts TestPlan: Select all permissions except **Instance > delete**, as shown in the following figure.

Figure 4-23 CodeArts TestPlan

Resource	Permission							
Test Case	<input checked="" type="checkbox"/> create <input checked="" type="checkbox"/> exportToFile	<input checked="" type="checkbox"/> update <input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> delete <input checked="" type="checkbox"/> importFromFile	<input checked="" type="checkbox"/> execute <input checked="" type="checkbox"/> importFromVersion	<input checked="" type="checkbox"/> stop <input checked="" type="checkbox"/> mergeToMaster	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Test Suite	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> execute	<input checked="" type="checkbox"/> stop	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Test Version	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> read				<input checked="" type="checkbox"/>
Test Plan	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> read				<input checked="" type="checkbox"/>
Test Quality Dashboard	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> read				<input checked="" type="checkbox"/>
Test Quality Evaluation	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> download			<input checked="" type="checkbox"/>
Test Case Recycle Bin	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> recover	<input checked="" type="checkbox"/> read					<input checked="" type="checkbox"/>
Testing Settings	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> read				<input checked="" type="checkbox"/>
Key Word	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> update				<input checked="" type="checkbox"/>
Global Variable	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> update				<input checked="" type="checkbox"/>
EchoTest Task	<input checked="" type="checkbox"/> read <input checked="" type="checkbox"/> stop	<input checked="" type="checkbox"/> create <input checked="" type="checkbox"/> configureAdvancedSet...	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> execute			<input checked="" type="checkbox"/>
EchoTest Alert Configuration	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> update				<input checked="" type="checkbox"/>
EchoTest Alert Notification	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> ignore						<input checked="" type="checkbox"/>
EchoTest Dashboard	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> update				<input checked="" type="checkbox"/>
Mindmap	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> deleteAll	<input checked="" type="checkbox"/> updateAll		<input checked="" type="checkbox"/>
Mindmap Backup	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> recover	<input checked="" type="checkbox"/> deleteAll	<input checked="" type="checkbox"/> recoverAll		<input checked="" type="checkbox"/>
Mindmap Template	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> deleteAll	<input checked="" type="checkbox"/> updateAll		<input checked="" type="checkbox"/>
Recycle	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> recover					<input checked="" type="checkbox"/>
Mindmap Asset	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> import	<input checked="" type="checkbox"/> export		<input checked="" type="checkbox"/>
Service Group	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> delete				<input checked="" type="checkbox"/>
Service	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> delete				<input checked="" type="checkbox"/>
Interface	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> copy	<input checked="" type="checkbox"/> import		<input checked="" type="checkbox"/>
Rule	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> delete				<input checked="" type="checkbox"/>
Instance	<input checked="" type="checkbox"/> deploy	<input checked="" type="checkbox"/> start	<input checked="" type="checkbox"/> stop	<input checked="" type="checkbox"/> update	<input type="checkbox"/> delete	<input checked="" type="checkbox"/> read		<input type="checkbox"/>

- Keep other permissions the same as those of the **Test manager** role.

Step 3 Click **Save**.

The configured permissions are displayed.

----End

4.2.7 Configuring Permissions for the PO Team

The PO team is composed of a product leader and a product manager.

In this practice, retain the default permissions for the product manager role. Configure permissions for the product leader role.

Step 1 On the **Permissions** page, click **Create Role**.

Step 2 In the displayed dialog box, enter the role name **Product leader** and click **OK**.

The new role is displayed in the custom role list.

Step 3 Click **Product leader**.

Step 4 Click **Edit**, and configure permissions for the role.

- General: Deselect all permissions.
- CodeArts Req & Defect: Select permissions by referring to the following figure.

Figure 4-24 CodeArts Req & Defect

Resource	Permission	
Files	<input type="checkbox"/> upload <input type="checkbox"/> edit <input type="checkbox"/> delete <input type="checkbox"/> download <input type="checkbox"/> preview	
Bug	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> Add workloads <input type="checkbox"/> Migrate <input checked="" type="checkbox"/> View list <input checked="" type="checkbox"/> Delete/Restore/Delete <input checked="" type="checkbox"/> Assign <input checked="" type="checkbox"/> Create/Clone <input checked="" type="checkbox"/> Associate/Dissociat... <input checked="" type="checkbox"/> Suspend/Cancel susp... <input checked="" type="checkbox"/> Edit <input checked="" type="checkbox"/> Associate/Dissociat... <input checked="" type="checkbox"/> Set statuses <input checked="" type="checkbox"/> Import <input checked="" type="checkbox"/> Export	
Config	<input type="checkbox"/> Basic <input type="checkbox"/> Notifications <input checked="" type="checkbox"/> Manage tags <input type="checkbox"/> Configure work item te... <input checked="" type="checkbox"/> Module <input type="checkbox"/> Configure workflows <input type="checkbox"/> R&D Downstream Proje... <input type="checkbox"/> RR Downstream Proje... <input type="checkbox"/> Set work types <input type="checkbox"/> Review <input type="checkbox"/> Automation	
SF	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> Add workloads <input checked="" type="checkbox"/> Create/Associate/Disa... <input checked="" type="checkbox"/> Associate/Dissociat... <input checked="" type="checkbox"/> View older versions <input checked="" type="checkbox"/> View list <input checked="" type="checkbox"/> Delete/Restore/Delete... <input checked="" type="checkbox"/> Baseline/Cancel baseli... <input checked="" type="checkbox"/> Assign/Cancel assign... <input checked="" type="checkbox"/> Migrate <input checked="" type="checkbox"/> Create/Clone <input checked="" type="checkbox"/> Set statuses <input checked="" type="checkbox"/> Break down/Associat... <input checked="" type="checkbox"/> Import <input checked="" type="checkbox"/> Suspend/Cancel susp... <input checked="" type="checkbox"/> Edit <input checked="" type="checkbox"/> Break down/Associat... <input checked="" type="checkbox"/> Export <input checked="" type="checkbox"/> Upload attachments <input checked="" type="checkbox"/> Break down/Associat... <input checked="" type="checkbox"/> Associate/Dissociat...	<input checked="" type="checkbox"/>
SF Collection	<input checked="" type="checkbox"/> Inherit <input checked="" type="checkbox"/> Read Snapshot <input checked="" type="checkbox"/> Create <input checked="" type="checkbox"/> Compare Snapshot <input checked="" type="checkbox"/> Update <input checked="" type="checkbox"/> Delete <input checked="" type="checkbox"/> Baseline Snapshot	<input checked="" type="checkbox"/>
IR	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> Add workloads <input checked="" type="checkbox"/> Baseline/Cancel baseli... <input checked="" type="checkbox"/> Migrate <input checked="" type="checkbox"/> View list <input checked="" type="checkbox"/> Delete/Restore/Delete... <input checked="" type="checkbox"/> Set statuses <input checked="" type="checkbox"/> Accept/Reject/Retur... <input checked="" type="checkbox"/> Export <input checked="" type="checkbox"/> Create/Clone <input checked="" type="checkbox"/> Break down/Associat... <input checked="" type="checkbox"/> Break down requireme... <input checked="" type="checkbox"/> Associate/Dissociat... <input checked="" type="checkbox"/> Associate/Dissociat... <input checked="" type="checkbox"/> Suspend/Cancel susp...	<input checked="" type="checkbox"/>
Plan Management	<input type="checkbox"/> Create <input type="checkbox"/> Import <input type="checkbox"/> Edit <input type="checkbox"/> Export <input type="checkbox"/> Baseline/Cancel baseli... <input type="checkbox"/> Set statuses	
Recycle	<input type="checkbox"/> Clear recycle bin	
Review	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> View list <input checked="" type="checkbox"/> Edit/Cancel <input type="checkbox"/> Delete <input checked="" type="checkbox"/> Export	
RR	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> Add workloads <input checked="" type="checkbox"/> Associate/Dissociat... <input checked="" type="checkbox"/> Import <input checked="" type="checkbox"/> View list <input checked="" type="checkbox"/> Delete/Restore/Delete... <input checked="" type="checkbox"/> Migrate <input checked="" type="checkbox"/> Create/Submit/Copy <input checked="" type="checkbox"/> Break down/Associat... <input checked="" type="checkbox"/> Assign <input checked="" type="checkbox"/> Associate/Dissociat... <input checked="" type="checkbox"/> Suspend/Cancel susp... <input checked="" type="checkbox"/> Edit <input checked="" type="checkbox"/> Create/Associate/Disa... <input checked="" type="checkbox"/> Set statuses <input checked="" type="checkbox"/> Upload attachments <input checked="" type="checkbox"/> Associate/Dissociat... <input checked="" type="checkbox"/> Set statuses	<input checked="" type="checkbox"/>
Task	<input type="checkbox"/> View <input type="checkbox"/> Add workloads <input type="checkbox"/> Associate/Dissociat... <input type="checkbox"/> View list <input type="checkbox"/> Delete/Restore/Delete... <input type="checkbox"/> Set statuses <input type="checkbox"/> Associate/Dissociat... <input type="checkbox"/> Create/Clone <input type="checkbox"/> Edit <input type="checkbox"/> Break down tasks <input type="checkbox"/> Import <input type="checkbox"/> Upload attachments <input type="checkbox"/> Associate/Dissociat... <input type="checkbox"/> Export <input type="checkbox"/> Suspend/Cancel susp...	

- CodeArts Repo: Deselect all permissions.
- CodeArts Check: Select the **CodeCheck_Report > read**, **CodeCheck_Task > execute**, and **CodeCheck_Task > read** permissions, as shown in the following figure.

Figure 4-25 CodeArts Check

Resource	Permission	
CodeCheck_Report	<input checked="" type="checkbox"/> read <input type="checkbox"/> edit	
CodeCheck_Rule_Set	<input type="checkbox"/> setDefault	
CodeCheck_Task	<input type="checkbox"/> create <input checked="" type="checkbox"/> execute <input checked="" type="checkbox"/> read <input type="checkbox"/> edit <input type="checkbox"/> delete	

- CodeArts TestPlan: Select permissions by referring to the following figure.

Figure 4-26 CodeArts TestPlan

Resource	Permission	
Test Case	<input type="checkbox"/> create <input type="checkbox"/> exportToFile <input checked="" type="checkbox"/> read <input type="checkbox"/> update <input type="checkbox"/> delete <input type="checkbox"/> execute <input type="checkbox"/> stop	
Test Suite	<input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete <input checked="" type="checkbox"/> read <input type="checkbox"/> execute <input type="checkbox"/> stop	
Test Version	<input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete <input checked="" type="checkbox"/> read	
Test Plan	<input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete <input checked="" type="checkbox"/> read	
Test Quality Dashboard	<input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete <input checked="" type="checkbox"/> read	
Test Quality Evaluation	<input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete <input checked="" type="checkbox"/> read <input checked="" type="checkbox"/> download	
Test Case Recycle Bin	<input type="checkbox"/> delete <input type="checkbox"/> recover <input type="checkbox"/> read	
Testing Settings	<input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete <input type="checkbox"/> read	
Key Word	<input type="checkbox"/> create <input checked="" type="checkbox"/> read <input type="checkbox"/> delete <input type="checkbox"/> update	
Global Variable	<input type="checkbox"/> create <input checked="" type="checkbox"/> read <input type="checkbox"/> delete <input type="checkbox"/> update	
Echo Test Task	<input type="checkbox"/> read <input type="checkbox"/> stop <input type="checkbox"/> create <input type="checkbox"/> configureAdvancesSet... <input type="checkbox"/> delete <input type="checkbox"/> update <input type="checkbox"/> execute	
Echo Test Alert Configuration	<input type="checkbox"/> read <input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> update	
Echo Test Alert Notification	<input type="checkbox"/> read <input type="checkbox"/> ignore	
Echo Test Dashboard	<input checked="" type="checkbox"/> read <input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> update	
Mindmap	<input checked="" type="checkbox"/> read <input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> update <input type="checkbox"/> deleteAll <input type="checkbox"/> updateAll	
Mindmap Backup	<input checked="" type="checkbox"/> read <input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> recover <input type="checkbox"/> deleteAll <input type="checkbox"/> recoverAll	
Mindmap Template	<input checked="" type="checkbox"/> read <input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> update <input type="checkbox"/> deleteAll <input type="checkbox"/> updateAll	
Recycle	<input checked="" type="checkbox"/> read <input type="checkbox"/> delete <input type="checkbox"/> recover	
Mindmap Asset	<input checked="" type="checkbox"/> read <input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> update <input type="checkbox"/> import <input type="checkbox"/> export	
Service Group	<input type="checkbox"/> create <input checked="" type="checkbox"/> read <input type="checkbox"/> update <input type="checkbox"/> delete	
Service	<input type="checkbox"/> create <input checked="" type="checkbox"/> read <input type="checkbox"/> update <input type="checkbox"/> delete	
Interface	<input type="checkbox"/> create <input checked="" type="checkbox"/> read <input type="checkbox"/> update <input type="checkbox"/> delete <input type="checkbox"/> copy <input type="checkbox"/> import	
Rule	<input type="checkbox"/> create <input checked="" type="checkbox"/> read <input type="checkbox"/> update <input type="checkbox"/> delete	
Instance	<input type="checkbox"/> deploy <input type="checkbox"/> start <input type="checkbox"/> stop <input type="checkbox"/> update <input type="checkbox"/> delete <input checked="" type="checkbox"/> read	

- CodeArts Pipeline: Deselect all permissions.
- CodeArts Build: Deselect all permissions.
- CodeArts Deploy: Deselect all permissions.

- CodeArts Artifact: Deselect all permissions.
- DevMarket (extension marketplace): Deselect all permissions.

Step 5 Click **Save**.

The configured permissions are displayed.

----End

4.2.8 Configuring Permissions for the PM Team

Creating a Project Management Leader Role and Configuring Permissions

Step 1 On the **Permissions** page, click **Create Role**.

Step 2 In the displayed dialog box, enable **Copy Role**, set the following parameters, and click **OK**.

Table 4-5 Creating a role

Parameter	Example
Role Name	Project management leader
Project	IPD_Demo
Role	Custom Role > Product leader

The new role is displayed in the custom role list.

Step 3 Click **Project management leader**.

Step 4 Click **Edit**, and configure permissions for the role.

- CodeArts Req & Defect: Select permissions by referring to the following figure.

Figure 4-27 CodeArts Req & Defect

Resource	Permission
Files	<input type="checkbox"/> upload <input type="checkbox"/> edit <input type="checkbox"/> delete <input type="checkbox"/> download <input type="checkbox"/> preview
Bug	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> View list <input checked="" type="checkbox"/> Create/Clone <input checked="" type="checkbox"/> Edit <input checked="" type="checkbox"/> Upload attachments <input type="checkbox"/> Add workloads <input checked="" type="checkbox"/> Delete/Restore/Delete... <input checked="" type="checkbox"/> Associate/Disassociat... <input checked="" type="checkbox"/> Associate/Disassociat... <input checked="" type="checkbox"/> Import <input type="checkbox"/> Migrate <input checked="" type="checkbox"/> Assign <input checked="" type="checkbox"/> Suspend/Cancel susp... <input checked="" type="checkbox"/> Set statuses <input checked="" type="checkbox"/> Export
Config	<input type="checkbox"/> Basic <input checked="" type="checkbox"/> Manage tags <input checked="" type="checkbox"/> Module <input type="checkbox"/> R&D Downstream Proj... <input type="checkbox"/> Set work types <input type="checkbox"/> Notifications <input type="checkbox"/> Configure work item te... <input type="checkbox"/> Configure workflows <input type="checkbox"/> Review <input type="checkbox"/> RR Downstream Proje... <input type="checkbox"/> Automation
SF	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> View list <input type="checkbox"/> Create/Clone <input type="checkbox"/> Edit <input type="checkbox"/> Upload attachments <input checked="" type="checkbox"/> Add workloads <input checked="" type="checkbox"/> Delete/Restore/Delete... <input type="checkbox"/> Set statuses <input type="checkbox"/> Break down/Associat... <input type="checkbox"/> Break down/Associat... <input type="checkbox"/> Create/Associate/Disa... <input type="checkbox"/> Baseline/Cancel baseli... <input type="checkbox"/> Import <input type="checkbox"/> Export <input type="checkbox"/> Associate/Disassociat... <input type="checkbox"/> Associate/Disassociat... <input checked="" type="checkbox"/> Associate/Disassociat... <input checked="" type="checkbox"/> View older versions <input checked="" type="checkbox"/> Suspend/Cancel susp...
SF Collection	<input type="checkbox"/> Inherit <input type="checkbox"/> Create <input type="checkbox"/> Update <input type="checkbox"/> Delete <input type="checkbox"/> Baseline Snapshot
	<input checked="" type="checkbox"/> Read Snapshot <input checked="" type="checkbox"/> Compare Snapshot
IR	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> View list <input type="checkbox"/> Create/Clone <input type="checkbox"/> Edit <input type="checkbox"/> Upload attachments <input checked="" type="checkbox"/> Add workloads <input checked="" type="checkbox"/> Delete/Restore/Delete... <input type="checkbox"/> Set statuses <input type="checkbox"/> Break down requireme... <input type="checkbox"/> Associate/Disassociat... <input type="checkbox"/> Associate/Disassociat... <input checked="" type="checkbox"/> Baseline/Cancel baseli... <input type="checkbox"/> Assign/Cancel assign... <input checked="" type="checkbox"/> Export <input type="checkbox"/> Suspend/Cancel susp... <input type="checkbox"/> Migrate <input type="checkbox"/> Import <input type="checkbox"/> Suspend/Cancel susp...
Plan Management	<input checked="" type="checkbox"/> Create <input checked="" type="checkbox"/> Edit <input checked="" type="checkbox"/> Delete <input checked="" type="checkbox"/> Baseline/Cancel baseli... <input checked="" type="checkbox"/> Set statuses <input checked="" type="checkbox"/> Import <input checked="" type="checkbox"/> Export <input checked="" type="checkbox"/> Baseline Snapshot
Recycle	<input type="checkbox"/> Clear recycle bin
Review	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> View list <input checked="" type="checkbox"/> Edit/Cancel <input type="checkbox"/> Delete <input checked="" type="checkbox"/> Export
RR	<input checked="" type="checkbox"/> View <input checked="" type="checkbox"/> View list <input type="checkbox"/> Create/Submit/Copy <input type="checkbox"/> Edit <input type="checkbox"/> Upload attachments <input checked="" type="checkbox"/> Add workloads <input checked="" type="checkbox"/> Delete/Restore/Delete... <input type="checkbox"/> Break down/Associat... <input type="checkbox"/> Associate/Disassociat... <input type="checkbox"/> Associate/Disassociat... <input checked="" type="checkbox"/> Baseline/Cancel baseli... <input type="checkbox"/> Migrate <input type="checkbox"/> Assign <input type="checkbox"/> Suspend/Cancel susp... <input type="checkbox"/> Set statuses <input type="checkbox"/> Import
Task	<input type="checkbox"/> View <input type="checkbox"/> View list <input type="checkbox"/> Create/Clone <input type="checkbox"/> Edit <input type="checkbox"/> Upload attachments <input checked="" type="checkbox"/> Add workloads <input checked="" type="checkbox"/> Delete/Restore/Delete... <input type="checkbox"/> Set statuses <input type="checkbox"/> Break down tasks <input type="checkbox"/> Associate/Disassociat... <input type="checkbox"/> Associate/Disassociat... <input type="checkbox"/> Associate/Disassociat... <input type="checkbox"/> Import <input type="checkbox"/> Export <input type="checkbox"/> Suspend/Cancel susp...

- Keep other permissions the same as those of the **Product leader** role.

Step 5 Click **Save**.

The configured permissions are displayed.

----End

Creating a Project Management Team Member Role and Configuring Permissions

Step 1 On the **Permissions** page, click **Create Role**.

Step 2 In the displayed dialog box, enable **Copy Role**, set the following parameters, and click **OK**.

Table 4-6 Creating a role

Parameter	Example
Role Name	Project management team member
Project	IPD_Demo
Role	Custom Role > Project management leader

The new role is displayed in the custom role list.

Keep the permissions the same as those of the **Project management leader** role, and do not make any changes.

----End

4.2.9 Configuring Permissions for the Big Data Team

Creating a Big Data Leader Role and Configuring Permissions

Step 1 On the **Permissions** page, click **Create Role**.

Step 2 In the displayed dialog box, enable **Copy Role**, set the following parameters, and click **OK**.

Table 4-7 Creating a role

Parameter	Example
Role Name	Big data leader
Project	IPD_Demo
Role	Custom Role > R&D leader

The new role is displayed in the custom role list.

Keep the permissions the same as those of the **R&D leader** role, and do not make any changes.

----End

Creating a Big Data Product Manager Role and Configuring Permissions

Step 1 On the **Permissions** page, click **Create Role**.

Step 2 In the displayed dialog box, enable **Copy Role**, set the following parameters, and click **OK**.

Table 4-8 Creating a role

Parameter	Example
Role Name	Big data product manager
Project	IPD_Demo
Role	System Role > Product manager

The new role is displayed in the custom role list.

Keep the permissions the same as those of the **Product manager** role, and do not make any changes.

----End

Creating a Big Data Developer Role and Configuring Permissions

Step 1 On the **Permissions** page, click **Create Role**.

Step 2 In the displayed dialog box, enable **Copy Role**, set the following parameters, and click **OK**.

Table 4-9 Creating a role

Parameter	Example
Role Name	Big data developer
Project	IPD_Demo
Role	System Role > Developer

The new role is displayed in the custom role list.

Keep the permissions the same as those of the **Developer** role, and do not make any changes.

----End

4.2.10 Configuring Permissions for the O&M Team

Creating an O&M Leader Role and Configuring Permissions

Step 1 On the **Permissions** page, click **Create Role**.

Step 2 In the displayed dialog box, enter the role name **O&M leader** and click **OK**.

The new role is displayed in the custom role list.

Step 3 Click **O&M leader**.

Step 4 Click **Edit**, and configure permissions for the role.

- General: Deselect all permissions.
- CodeArts Req & Defect: Select permissions by referring to the following figure.

Figure 4-28 CodeArts Req & Defect

.. CodeArts Req & Defect

Resource	Permission					
Files	<input type="checkbox"/> upload	<input type="checkbox"/> edit	<input type="checkbox"/> delete	<input type="checkbox"/> download	<input type="checkbox"/> preview	<input type="checkbox"/>
Bug	<input checked="" type="checkbox"/> View	<input checked="" type="checkbox"/> Add workloads	<input checked="" type="checkbox"/> Migrate	<input checked="" type="checkbox"/> View list	<input checked="" type="checkbox"/> Delete/Restore/Delete ...	<input checked="" type="checkbox"/> Assign
Config	<input type="checkbox"/> Basic	<input type="checkbox"/> Notifications	<input type="checkbox"/> Manage tags	<input type="checkbox"/> Configure work item te...	<input type="checkbox"/> Module	<input type="checkbox"/> Configure workflows
SF	<input type="checkbox"/> View	<input type="checkbox"/> Add workloads	<input type="checkbox"/> Create/Associate/Disa...	<input type="checkbox"/> Baseline/Cancel baseli...	<input type="checkbox"/> View older versions	<input type="checkbox"/> Suspend/Cancel susp...
SF Collection	<input type="checkbox"/> Inherit	<input type="checkbox"/> Read Snapshot	<input type="checkbox"/> Create	<input type="checkbox"/> Compare Snapshot	<input type="checkbox"/> Update	<input type="checkbox"/> Delete
IR	<input type="checkbox"/> View	<input type="checkbox"/> Add workloads	<input type="checkbox"/> Baseline/Cancel baseli...	<input type="checkbox"/> Migrate	<input type="checkbox"/> Create/Clone	<input type="checkbox"/> Set statuses
Plan Management	<input type="checkbox"/> Create	<input type="checkbox"/> Import	<input type="checkbox"/> Edit	<input type="checkbox"/> Export	<input type="checkbox"/> Delete	<input type="checkbox"/> Baseline Snapshot
Recycle	<input type="checkbox"/> Clear recycle bin					
Review	<input type="checkbox"/> View	<input type="checkbox"/> View list	<input type="checkbox"/> Edit/Cancel	<input type="checkbox"/> Delete	<input type="checkbox"/> Export	
RR	<input type="checkbox"/> View	<input type="checkbox"/> Add workloads	<input type="checkbox"/> Associate/Disassociat...	<input type="checkbox"/> Import	<input type="checkbox"/> Create/Submit/Copy	<input type="checkbox"/> Edit
Task	<input type="checkbox"/> View	<input type="checkbox"/> Add workloads	<input type="checkbox"/> Associate/Disassociat...	<input type="checkbox"/> View list	<input type="checkbox"/> Delete/Restore/Delete ...	<input type="checkbox"/> Set statuses

- CodeArts Repo: Deselect all permissions.
- CodeArts Check: Deselect all permissions.
- CodeArts TestPlan: Select permissions by referring to the following figure.

Figure 4-29 CodeArts TestPlan

Resource	Permission							
Test Case	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	<input type="checkbox"/> stop	<input type="checkbox"/> exportToFile	<input type="checkbox"/> read	<input type="checkbox"/>
Test Suite	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> read	<input type="checkbox"/> execute	<input type="checkbox"/> stop	<input type="checkbox"/>	<input type="checkbox"/>
Test Version	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input checked="" type="checkbox"/> read	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Test Plan	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input checked="" type="checkbox"/> read	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Test Quality Dashboard	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input checked="" type="checkbox"/> read	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Test Quality Evaluation	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> download	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Test Case Recycle Bin	<input type="checkbox"/> delete	<input type="checkbox"/> recover	<input type="checkbox"/> read	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Testing Settings	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> read	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Key Word	<input type="checkbox"/> create	<input type="checkbox"/> read	<input type="checkbox"/> delete	<input type="checkbox"/> update	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Global Variable	<input type="checkbox"/> create	<input type="checkbox"/> read	<input type="checkbox"/> delete	<input type="checkbox"/> update	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
EchoTest Task	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> update	<input type="checkbox"/> execute	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
EchoTest Alert Configuration	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> update	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
EchoTest Alert Notification	<input type="checkbox"/> read	<input type="checkbox"/> ignore	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
EchoTest Dashboard	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> update	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mindmap	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> update	<input type="checkbox"/> deleteAll	<input type="checkbox"/> updateAll	<input type="checkbox"/>	<input type="checkbox"/>
Mindmap Backup	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> recover	<input type="checkbox"/> deleteAll	<input type="checkbox"/> recoverAll	<input type="checkbox"/>	<input type="checkbox"/>
Mindmap Template	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> update	<input type="checkbox"/> deleteAll	<input type="checkbox"/> updateAll	<input type="checkbox"/>	<input type="checkbox"/>
Recycle	<input type="checkbox"/> read	<input type="checkbox"/> delete	<input type="checkbox"/> recover	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mindmap Asset	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> update	<input type="checkbox"/> import	<input type="checkbox"/> export	<input type="checkbox"/>	<input type="checkbox"/>
Service Group	<input type="checkbox"/> create	<input type="checkbox"/> read	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Service	<input type="checkbox"/> create	<input type="checkbox"/> read	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interface	<input type="checkbox"/> create	<input type="checkbox"/> read	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> copy	<input type="checkbox"/> import	<input type="checkbox"/>	<input type="checkbox"/>
Rule	<input type="checkbox"/> create	<input type="checkbox"/> read	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Instance	<input type="checkbox"/> deploy	<input type="checkbox"/> start	<input type="checkbox"/> stop	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> read	<input type="checkbox"/>	<input type="checkbox"/>

- CodeArts Pipeline: Select permissions by referring to the following figure.

Figure 4-30 CodeArts Pipeline

Resource	Permission							
Change	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> execute	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Development Environment	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Microservice	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> delete	<input type="checkbox"/> update	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pipeline	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> execute	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Policy	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> update	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Pre_production Environment	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Production Environment	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Test Environment	<input type="checkbox"/> read	<input type="checkbox"/> create	<input type="checkbox"/> update	<input type="checkbox"/> delete	<input type="checkbox"/> execute	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
VariableGroup	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> bind	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
ReusableJob	<input checked="" type="checkbox"/> create	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> update	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

- CodeArts Build: Select all permissions.
- CodeArts Deploy: Select all permissions.
- CodeArts Artifact: Select all permissions.
- DevMarket (extension marketplace): Select all permissions.

Step 5 Click **Save**.

The configured permissions are displayed.

----End

Creating an O&M Engineer Role and Configuring Permissions

Step 1 On the **Permissions** page, click **Create Role**.

Step 2 In the displayed dialog box, enable **Copy Role**, set the following parameters, and click **OK**.

Table 4-10 Creating a role

Parameter	Example
Role Name	O&M engineer
Project	IPD_Demo
Role	Custom Role > O&M leader

The new role is displayed in the custom role list.

Step 3 Click **O&M engineer**.

Step 4 Click **Edit**, and configure permissions for the role.

- CodeArts Pipeline: Select permissions by referring to the following figure.

Figure 4-31 CodeArts Pipeline

Resource	Permission
Change	<input type="checkbox"/> read <input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> execute
Development Environment	<input type="checkbox"/> read <input type="checkbox"/> rollback <input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete <input type="checkbox"/> execute
Microservice	<input type="checkbox"/> read <input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> update
Pipeline	<input checked="" type="checkbox"/> read <input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> update <input checked="" type="checkbox"/> execute
Policy	<input checked="" type="checkbox"/> read <input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> update
Pie_production Environment	<input type="checkbox"/> read <input type="checkbox"/> rollback <input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete <input type="checkbox"/> execute
Production Environment	<input type="checkbox"/> read <input type="checkbox"/> rollback <input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete <input type="checkbox"/> execute
Test Environment	<input type="checkbox"/> read <input type="checkbox"/> rollback <input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete <input type="checkbox"/> execute
VariableGroup	<input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> update <input type="checkbox"/> bind
ReusableJob	<input type="checkbox"/> create <input type="checkbox"/> delete <input type="checkbox"/> update

- CodeArts Build: Select the **view** and **execute** permissions, as shown in the following figure.

Figure 4-32 CodeArts Build

Resource	Permission
job	<input type="checkbox"/> create <input type="checkbox"/> modify <input type="checkbox"/> delete <input checked="" type="checkbox"/> view <input checked="" type="checkbox"/> execute
	<input type="checkbox"/> copy <input type="checkbox"/> forbidden <input type="checkbox"/> manager <input type="checkbox"/> group

- CodeArts Deploy: Select the **Applications > View** and **Deploy** permissions, as shown in the following figure.

Figure 4-33 CodeArts Deploy

Resource	Permission
Applications	<input checked="" type="checkbox"/> View <input type="checkbox"/> Create <input type="checkbox"/> Edit <input type="checkbox"/> Delete <input checked="" type="checkbox"/> Deploy
	<input checked="" type="checkbox"/> Clone <input type="checkbox"/> Create Environment <input type="checkbox"/> Assign Permissions <input type="checkbox"/> Manage Groups <input type="checkbox"/> Disable
Resource	<input type="checkbox"/> Create

- CodeArts Artifact: Select the **artifact_gene > download** and **component > downloadOrView** permissions, as shown in the following figure.

Figure 4-34 CodeArts Artifact

Resource	Permission	
artifact_gene	<input type="checkbox"/> changePkgStatus <input type="checkbox"/> createFolder <input type="checkbox"/> createRepository <input type="checkbox"/> restoreall	<input type="checkbox"/> upload <input checked="" type="checkbox"/> download <input type="checkbox"/> editRepository <input type="checkbox"/> clearall
artifact_pri	<input type="checkbox"/> delOrRestoreTestPkg <input type="checkbox"/> restoreAll <input type="checkbox"/> deleteRepository <input type="checkbox"/> restore	<input type="checkbox"/> delOrRestoreProdPkg <input type="checkbox"/> clearAllRecycleBin <input type="checkbox"/> physicdelete
component	<input type="checkbox"/> upload <input checked="" type="checkbox"/> downloadOrView <input type="checkbox"/> deleteOrRedeploy	

- DevMarket (extension marketplace): Select the **read** permission, as shown in the following figure.

Figure 4-35 DevMarket

Resource	Permission	
EndPoint	<input type="checkbox"/> create <input type="checkbox"/> update <input type="checkbox"/> delete <input checked="" type="checkbox"/> read	

- Keep other permissions the same as those of the **O&M leader** role.

Step 5 Click **Save**.

The configured permissions are displayed.

----End

Creating an On-Call Engineer Role and Configuring Permissions

Step 1 On the **Permissions** page, click **Create Role**.

Step 2 In the displayed dialog box, enter the role name **On-call engineer** and click **OK**.

The new role is displayed in the custom role list.

This role does not require any permissions.

----End

4.2.11 Configuring Permissions for the DBA Team

Creating a DBA Leader Role and Configuring Permissions

Step 1 On the **Permissions** page, click **Create Role**.

Step 2 In the displayed dialog box, enter the role name **DBA leader** and click **OK**.

The new role is displayed in the custom role list.

Step 3 Click **DBA leader**.

Step 4 Click **Edit**, and configure permissions for the role.

- General: Select the **project-role > privilegeconfig** permission, as shown in the following figure.

Figure 4-36 General

Resource	Permission	
project	<input type="checkbox"/> edit	
project-role	<input checked="" type="checkbox"/> privilegeconfig <input type="checkbox"/> userconfig	
Organization Structure	<input type="checkbox"/> Associate Department <input type="checkbox"/> Disassociate Departm...	

- CodeArts Pipeline: Select all permissions.
- Deselect all other permissions.

Step 5 Click **Save**.

The configured permissions are displayed.

----End

Creating a DBA Team Member Role and Configuring Permissions

Step 1 On the **Permissions** page, click **Create Role**.

Step 2 In the displayed dialog box, enable **Copy Role**, set the following parameters, and click **OK**.

Table 4-11 Creating a role

Parameter	Example
Role Name	DBA team member
Project	IPD_Demo
Role	Custom Role > DBA leader

The new role is displayed in the custom role list.

Step 3 Click **DBA team member**.

Step 4 Click **Edit**, and configure permissions for the role.

- CodeArts Pipeline: Select permissions by referring to the following figure.

Figure 4-37 CodeArts Pipeline

Resource	Permission	
Change	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> create
Change	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> execute
Change	<input checked="" type="checkbox"/> delete	<input checked="" type="checkbox"/> execute
Development Environment	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> create
Development Environment	<input checked="" type="checkbox"/> update	<input type="checkbox"/> delete
Development Environment	<input checked="" type="checkbox"/> execute	<input type="checkbox"/> execute
Development Environment	<input checked="" type="checkbox"/> rollback	
Microservice	<input checked="" type="checkbox"/> read	<input type="checkbox"/> create
Microservice	<input type="checkbox"/> delete	<input type="checkbox"/> update
Pipeline	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> create
Pipeline	<input type="checkbox"/> group	<input type="checkbox"/> delete
Pipeline	<input type="checkbox"/> tag	<input type="checkbox"/> update
Pipeline	<input type="checkbox"/> ban	<input checked="" type="checkbox"/> execute
Policy	<input checked="" type="checkbox"/> read	<input checked="" type="checkbox"/> create
Policy	<input type="checkbox"/> delete	<input checked="" type="checkbox"/> update
Pre_production Environment	<input checked="" type="checkbox"/> read	<input type="checkbox"/> create
Pre_production Environment	<input type="checkbox"/> rollback	<input type="checkbox"/> update
Pre_production Environment	<input type="checkbox"/> delete	<input type="checkbox"/> execute
Production Environment	<input checked="" type="checkbox"/> read	<input type="checkbox"/> create
Production Environment	<input type="checkbox"/> rollback	<input type="checkbox"/> update
Production Environment	<input type="checkbox"/> delete	<input type="checkbox"/> execute
Test Environment	<input checked="" type="checkbox"/> read	<input type="checkbox"/> create
Test Environment	<input type="checkbox"/> rollback	<input type="checkbox"/> update
Test Environment	<input type="checkbox"/> delete	<input type="checkbox"/> execute
VariableGroup	<input checked="" type="checkbox"/> create	<input type="checkbox"/> delete
VariableGroup	<input type="checkbox"/> update	<input checked="" type="checkbox"/> bind
ReusableJob	<input checked="" type="checkbox"/> create	<input type="checkbox"/> delete
ReusableJob	<input checked="" type="checkbox"/> update	

- Keep other permissions the same as those of the **DBA leader** role.

Step 5 Click **Save**.

The configured permissions are displayed.

----End

4.2.12 Configuring Permissions for the Security Team

Step 1 On the **Permissions** page, click **Create Role**.

Step 2 In the displayed dialog box, enter the role name **Security leader** and click **OK**.

The new role is displayed in the custom role list.

Step 3 Click **Security leader**.

Step 4 Click **Edit**, and configure permissions for the role.

- CodeArts Check: Select all permissions.
- CodeArts Artifact: Select the **artifact_gene** > **download** and **component** > **downloadOrView** permissions, as shown in the following figure.

Figure 4-38 CodeArts Artifact

Resource	Permission					
artifact_gene	<input type="checkbox"/> changePkgStatus	<input type="checkbox"/> upload	<input type="checkbox"/> delCrRestoreTestPkg	<input type="checkbox"/> delCrRestoreProdPkg	<input type="checkbox"/> editTestPkg	<input type="checkbox"/>
	<input type="checkbox"/> createFolder	<input checked="" type="checkbox"/> download	<input type="checkbox"/> restoreAll	<input type="checkbox"/> crAllInRecycleBin		<input type="checkbox"/>
artifact_pri	<input type="checkbox"/> createRepository	<input type="checkbox"/> editRepository	<input type="checkbox"/> deleteRepository	<input type="checkbox"/> restore	<input type="checkbox"/> physicdelete	<input type="checkbox"/>
	<input type="checkbox"/> restoreall	<input type="checkbox"/> clearall				
component	<input type="checkbox"/> upload	<input checked="" type="checkbox"/> downloadOrView	<input type="checkbox"/> deleteCrRedeploy			<input type="checkbox"/>

- Deselect all other permissions.

Step 5 Click **Save**.

The configured permissions are displayed.

----End