



# 软件开发生产线 CodeArts 最佳实践

天翼云科技有限公司

---

# 目 录

---

<b>1 使用 CodeArts 管理电子商城项目开发流程</b> .....	<b>3</b>
1.1 方案概述 .....	3
1.2 资源规划 .....	6
1.3 操作流程 .....	7
1.4 实施步骤 .....	9
1.4.1 实践准备工作 .....	9
1.4.2 管理项目规划 .....	10
1.4.3 开发代码 .....	14
1.4.4 检查代码 .....	16
1.4.5 构建应用并推送镜像至 SWR.....	18
1.4.6 部署应用至 CCE.....	22
1.4.7 部署应用至 ECS.....	24
1.4.8 管理项目测试 .....	28
1.4.9 配置流水线 .....	32
1.4.10 释放资源 .....	34
<b>2 修订记录</b> .....	错误!未定义书签。

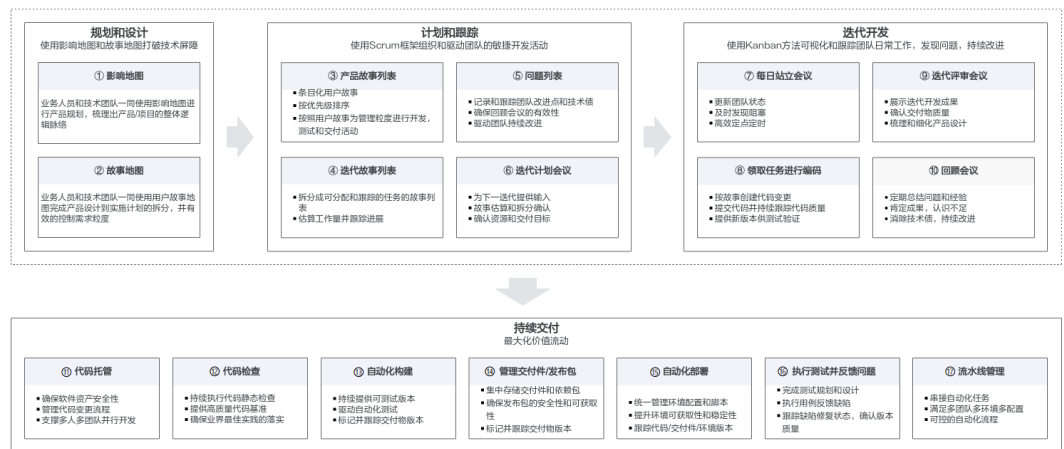
# 1 使用 CodeArts 管理电子商城项目开发流程

## 1.1 方案概述

### 背景信息

CodeArts 结合业界先进的实践提出了一套可操作可落地的敏捷开发方法论：HE2E DevOps 实施框架。

图1-1 HE2E DevOps 实施框架



- 规划和设计

步骤①和②是业务（或者是客户）与技术之间进行产品规划，梳理产品整体脉络，以及进行产品规划实施设计，并控制需求粒度与拆分的过程。

- 软件开发的本质是为了解决问题，提供用户价值的，而不仅是为了提供功能。影响地图就是用来鉴别用户需求是什么，深层的根因是什么。
- 用户故事就是目标和需求的载体，以用户的场景来讲故事，便于在客户、业务与开发之间进行信息的传递。在这个过程中，独立的需求条目的堆积，很容易导致只能看到各个需求条目，不能从整个解决方案思考需求。用户故事以用户使用的场景为主线，将大的阶段点，及其细分的活动，以树状的结构进行梳理和展现，既可以看到独立的需求条目，又能够看到整体需求场景。

- 计划和跟踪、迭代开发  
步骤③~⑩是 Scrum 框架过程，是主要的管理实践。
  - Scrum 定义了一个相对完整的敏捷过程管理的框架。在 CodeArts 中，将 Scrum 的框架与团队日常的开发活动，很好的融合起来。主要的过程产物包括产品故事列表、迭代故事列表、潜在可交付的产品增量、以及过程中产生的问题列表；核心的团队活动包括 Sprint 计划会议、团队每日站会、Sprint 演示会议、Sprint 回顾会议等会议、以及团队的日常更新。
  - 同时，将 Kanban 方法与 Scrum 框架进行了结合，团队借鉴 Kanban 方法中的精益思想，可视化价值流，发现并解决阻塞与瓶颈，加速价值流交付，并加快反馈回路，持续进行改进。
- 持续交付  
从步骤⑪开始，进入到工程实践，也就是通常说的 CI/CD 过程。
  - 持续交付以代码配置管理为基础，除了传统意义的代码资产安全与管控、多人并行开发、版本与基线管理外，也体现了团队的协作与沟通。
  - 代码检查（即静态扫描）、自动化的构建、各阶段的自动化测试、以及相应的自动化部署过程，都被有机的串联在流水线上。
  - 除了代码检查、构建、测试、部署等动态的阶段与活动，还有制品管理，以及各级的环境管理，包括开发环境、测试环境、准生产环境，以及生产环境。
  - 持续交付流水线就是将整个持续交付中，都有哪些阶段，分别运行在什么环境，每个阶段执行什么活动，准入与准出的质量门禁，以及每个阶段的输入与输出的制品进行管理。

## 背景信息

A 公司是一家汽车零部件经销商。为了开拓销售市场，A 公司开发了一个汽车零部件配件电子商城“凤凰商城”。

商城包含会员管理、配件管理、订单管理等功能，满足了 A 公司的管理需求、客户的查询与购买需求。

采用 Scrum 模式进行迭代开发，每个迭代周期为“两周”。

## 方案架构

汽车零部件配件电子商城由 5 个微服务组件构成。

图1-2 凤凰商城技术架构图

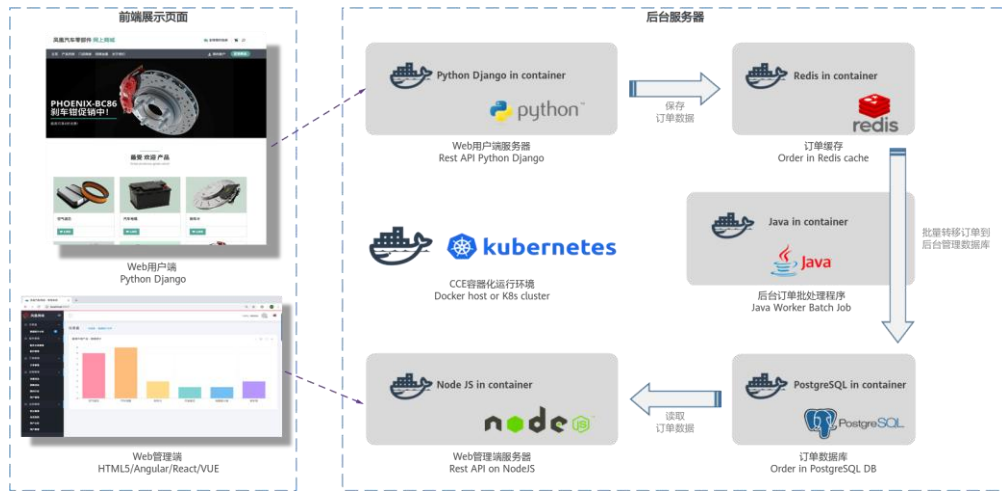


表1-1 产品构成

微服务组件	说明
Web 用户端服务器（对应样例代码中的“Vote”功能）	<ul style="list-style-type: none"> <li>业务逻辑：用户可以通过浏览器访问此服务的 WebUI。当用户在特定商品上单击“Like”时，服务将用户所选择物品的记录保存在 Redis 缓存中。</li> <li>技术栈：Python、Flask 框架。</li> <li>应用服务器：Gunicorn。</li> </ul>
Web 管理端服务器（对应样例代码中的“Result”功能）	<ul style="list-style-type: none"> <li>业务逻辑：用户可以通过浏览器访问此服务的 WebUI，会动态显示用户端 UI 上用户单击“Like”的统计数据，此数据来自 PostgreSQL 数据库。</li> <li>技术栈：Node.js、express 框架。</li> <li>应用服务器：server.js。</li> </ul>
后台订单批处理程序（对应样例代码中的“Worker”功能）	<ul style="list-style-type: none"> <li>业务逻辑：此服务为后台进程，会监控 Redis 缓存中物品记录，并将新记录取出并保存在 PostgreSQL 数据库中，以便管理端 UI 可以抽取数据进行统计显示。</li> <li>技术栈：.net core 或者 Java（此服务提供两种技术栈实现了同样的功能，可根据需要修改配置选择其中一个作为运行时进程）。</li> </ul>
订单缓存	<ul style="list-style-type: none"> <li>业务逻辑：此服务作为用户端 UI 服务的数据持久化服务存在。</li> <li>技术栈：Redis</li> </ul>
订单数据库	<ul style="list-style-type: none"> <li>业务逻辑：此服务作为管理端 UI 服务的数据源。</li> <li>技术栈：PostgreSQL</li> </ul>

项目研发过程中涉及到以下成员。

表1-2 项目角色列表

项目成员	项目角色	工作职责
Sarah	项目管理员	负责项目整体规划、项目团队的组建。
Maggie	项目经理	负责管理项目交付计划。
Chris	开发人员	负责项目代码的开发、编译、部署及验证。
Billy	测试人员	负责编写测试用例并执行。

## 1.2 资源规划

完成本实践所需的资源如下，实践预计用时 2~3 小时。

表1-3 资源规划

服务名称	用途	资源说明	操作指导
软件开发生产线 CodeArts	承载项目端到端研发全流程	<ul style="list-style-type: none"> <li>套餐版本：基础版。</li> <li>购买人数：5 人。</li> <li>购买时长：1 个月。</li> </ul>	参考《用户指南》中“购买 CodeArts”章节。
云容器引擎 CCE	用于部署应用	<b>集群配置：</b> <ul style="list-style-type: none"> <li>计费模式：建议选择按需计费。</li> <li>集群类型：CCE Standard 集群。</li> <li>集群名称：phoenix-cce。</li> <li>集群版本：建议选择最新版本。</li> <li>控制节点架构：X86。</li> <li>容器网络模型：容器隧道网络。</li> <li>容器网段：自动设置网段。</li> </ul>	参考《云容器引擎用户指南》中的“购买集群”章节。
		<b>节点配置：</b> <ul style="list-style-type: none"> <li>计费模式：建议选择按需计费。</li> <li>节点类型：弹性云服务器-虚拟机。</li> <li>节点规格：2vCPUs   8GiB 或以上规格。</li> </ul>	

服务名称	用途	资源说明	操作指导
		<ul style="list-style-type: none"><li>• 容器引擎: Docker。</li><li>• 操作系统: 公共镜像-EulerOS。</li><li>• 节点 IP: 自动分配。</li><li>• 弹性公网 IP: 自动创建。</li></ul>	
弹性云主机 ECS	用于部署应用	<ul style="list-style-type: none"><li>• 建议选择按需计费。</li><li>• CPU 架构: x86 计算。</li><li>• 节点规格: 2vCPUs   8GiB 或以上规格。</li><li>• 操作系统: 公共镜像-Ubuntu 16.04。</li><li>• 绑定弹性 IP 规格: 按带宽计费, 5Mbit/s。</li><li>• 云主机名称: phoenix-ecs。</li><li>• 完成购买后, 为 ECS 的安全组中添加一条协议为“TCP”、端口为“5000-5001”、源地址为“0.0.0.0/0”的入方向规则。</li></ul>	参考《弹性云主机快速入门》中的“购买弹性云主机”章节。

## 1.3 操作流程

本文档将按照以下步骤介绍 HE2E DevOps 实践的操作流程。

图1-3 实践操作流程

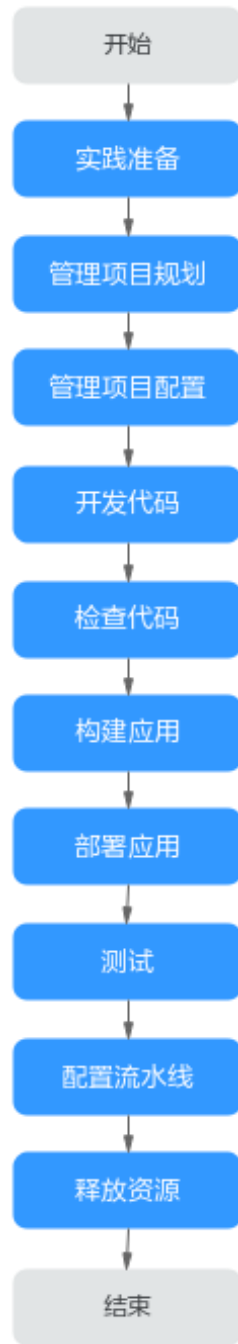


表1-4 实践操作流程说明

步骤	说明
实践准备	项目管理员 Sarah 完成实践开始前的准备工作，包括创建项目、添加项目成员等操作。
管理项目规划	项目管理员 Sarah 完成项目需求规划、项目经理 Maggie 完成迭代

步骤	说明
	需求规划。
开发代码	开发人员 Chris 通过分支来进行代码的编写，包括创建分支、编写并提交代码、提交分支合并请求；项目经理 Maggie 审核合并请求后完成合入。
检查代码	开发人员 Chris 对代码进行静态扫描，根据修复建议优化代码，提高代码质量。
构建应用	开发人员 Chris 构建环境镜像、将代码编译打包成软件包。
部署应用	开发人员 Chris 将构建好的环境镜像及软件包安装并运行在环境中。本文档提供两种环境的部署方法：CCE 与 ECS。
管理项目测试	测试人员 Billy 为迭代创建测试计划、设计测试用例，并按照计划执行测试用例。
配置流水线	开发人员 Chris 将代码检查、构建、部署等任务串联成流水线。当代码有更新时，可自动触发流水线，实现持续交付。
释放资源	按需计费的资源会产生持续计费，为了避免不必要的费用产生，完成体验后，建议由项目管理员 Sarah 释放不再使用的资源。

## 1.4 实施步骤

### 1.4.1 实践准备工作

在进行具体的任务操作前，您需要完成以下准备工作。

#### 创建项目

- 步骤 1 登录 CodeArts 首页。
- 步骤 2 单击“新建项目”，选择“DevOps 全流程样例项目”。
- 步骤 3 输入项目名称“凤凰商城”，单击“确定”。

页面提示创建成功，完成项目的创建。

---结束

#### 添加项目成员

- 步骤 1 参考《统一身份认证用户指南》中的“创建用户”章节，创建 3 个 IAM 用户，用户名分别为“Maggie”、“Chris”、“Billy”。
- 步骤 2 返回 CodeArts 首页，进入“凤凰商城”项目，进入“设置 > 成员管理”页面。

- 步骤 3 在“成员视图”页签中，单击“添加成员 > 从本账号导入 IAM 用户”。
- 步骤 4 在弹框中勾选成员“Maggie”、“Chris”、“Billy”，单击“下一步”。
- 步骤 5 单击每一行的“项目角色”下拉列表，为成员 Maggie 选择角色“项目经理”、Chris 选择角色“开发人员”、Billy 选择角色“测试人员”，单击“保存”。

添加成功，页面中显示已添加的成员。

- 步骤 6 单击导航“设置 > 权限管理”，进入权限管理页面。
- 步骤 7 在角色列表中选择“开发人员”，单击“编辑”，在“部署 > 应用”中勾选“新建环境”，单击“保存”。

图1-4 更新权限



---结束

## 1.4.2 管理项目规划

需求管理服务提供简单高效的团队协作服务，包含多项目管理、敏捷迭代、任务管理等功能。

本实践采用 Scrum 模式进行迭代开发，每个迭代周期为两周，前 3 个迭代已经完成凤凰商城版本的开发，当前即将启动迭代 4 的规划。

迭代 4 要完成一个新增的需求：门店网络查询功能。

本章节介绍项目管理员 Sarah 与项目经理 Maggie 如何进行项目规划的管理，包括管理需求规划、迭代规划、跟踪项目进度。

### 在需求规划中添加需求

本实践使用思维导图的形式管理项目需求规划，将工作项的层级结构“Epic>Feature>Story>Task”展示出来。更多介绍请参考《需求管理用户指南》中的“Scrum 项目需求管理流程介绍”。

本操作由项目管理员 Sarah 完成。

- 步骤 1 为新需求创建工作项。

由于门店网络查询功能是新增的需求，因此项目管理员 Sarah 要将它加入需求规划视图中。

1. 进入“凤凰商城”项目，单击导航“工作 > 需求管理”。
2. 选择“规划”页签，单击“凤凰商城思维导图”。
3. 新建 Feature “门店网络”。
  - a. 鼠标右键单击 Epic “凤凰商城”，选择“新增子工作项”。

图1-5 新建 Feature



- b. 待创建成功，单击 Epic 后的数字，展开子工作项，找到 Feature “凤凰商城”，单击名称展开工作项详情。
  - c. 修改名称为“门店网络”。
4. 按照同样的方式，为 Feature “门店网络”添加 Story “作为用户应该可以查询所有门店网络”。

#### 步骤 2 编辑 Story。

1. 单击 Story “作为用户应该可以查询所有门店网络”，参照下表编辑 Story 信息。

表1-5 Story 配置

配置项	示例	说明
描述信息	作为用户，我想要查询所有门店，以便于挑选合适的门店获取服务。	Story 的描述信息，不超过 50000 个字符。
优先级	高	Story 的优先级，可以选择“高”、“中”、“低”。
重要程度	关键	Story 的重要程度，可以选择“关键”、“重要”、“一般”、“提示”。

2. 为了便于开发人员理解，在本地准备一个文件“门店网络列表”，表格内容参照下表。

表1-6 门店网络列表

分店名称	分店地址
A 分店	E 机场 1 号航站楼出发层靠右直行 123 米右侧。
B 分店	F 区 G 路 456 号。
C 分店	H 区 J 街 789 号。
D 分店	K 区 L 大道 K 大楼西侧。

3. 返回 Story 编辑页面，单击“点击添加附件或拖拽文件到此处上传”，选择“本地上传”，将列表文件上传至工作项中作为附件。
4. 单击“保存”，完成 Story 详情的编辑。  
保存成功后，页面中显示编辑后的 Story 信息。

---结束

## 管理迭代规划

在迭代开始前，项目经理 Maggie 组织召开计划会议，根据规划将本次迭代中待实现的 Story 添加在迭代中，并将 Story 分解为 Task，分配给开发人员进行开发。

### 步骤 1 创建迭代。


1. 进入“凤凰商城”项目，单击导航“工作 > 需求管理”，进入需求管理服务。
2. 选择“迭代”页签，单击页面左上角“迭代”字样后的，在弹框中配置以下迭代信息，单击“确定”。

表1-7 迭代信息配置

配置项	示例	说明
迭代名称	迭代 4	迭代的名称，支持汉字、大小写英文字母、数字、小数点、下划线和连字符，不超过 60 个字符。
计划时间	设置时长为 2 周。	迭代的计划开始、结束时间。

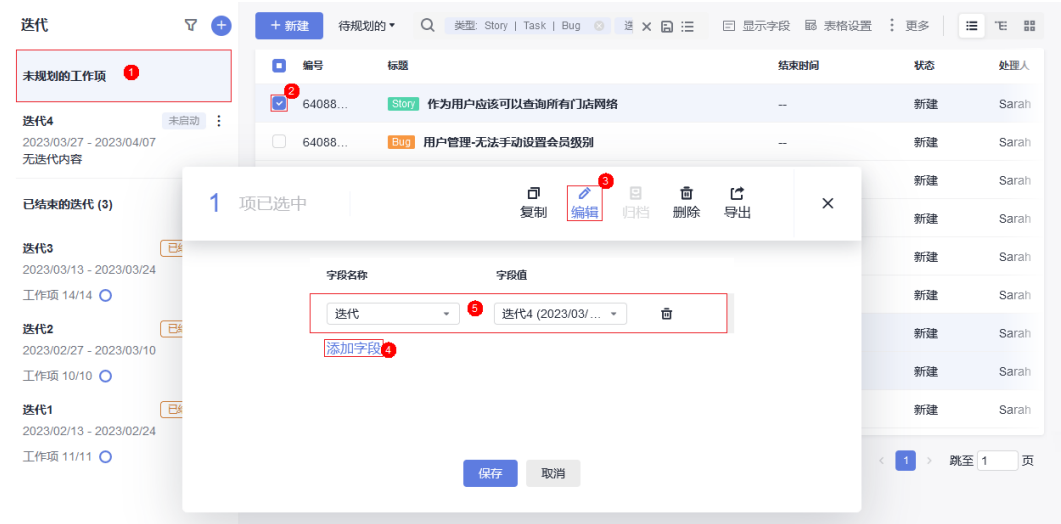
新建成功后，页面中显示新增的迭代。

### 步骤 2 向迭代中添加 Story。

1. 单击页面左侧导航“未规划的工作项”。
2. 在列表中勾选 Story “作为用户应该可以查询所有门店网络”。
3. 在页面底部单击“编辑”。
4. 单击“添加字段”。

5. 在字段名称下拉列表中选择“迭代”，并在字段值下拉列表中选择“迭代4”，单击“保存”。

图1-6 规划迭代



保存成功后，单击“迭代4”，页面中显示已添加的 Story。

### 步骤3 分配 Story。

1. 单击页面左侧导航“迭代4”。
2. 在列表中找到 Story “作为用户应该可以查询所有门店网络”，单击 Story 的名称，在 Story 的详情页中将字段“处理人”的值设置为“Chris”。
3. 分解 Story。
  - a. 在列表中找到 Story “作为用户应该可以查询所有门店网络”，单击 Story 标题打开详情页。
  - b. 选择“子工作项”页签，单击“快速新建子工作项”，输入标题“前端展示 - 添加门店网络菜单”，并选择处理人“Chris”，单击“确定”。  
新建成功后，页面中显示新建的子工作项。
  - c. 按照同样的方式，添加 Task “后台管理 - 添加门店网络管理维护模块”。

----结束

## 跟踪项目状态

在项目进展过程中，项目组可以通过每日站会、迭代评审会议、回顾会议等活动跟踪团队日常工作，发现问题，持续改进。

CodeArts 需求管理服务提供了多样的统计报表，也支持自定义统计报表，项目组可以使用报表可视化地展示当前的项目进展。

关于项目进展跟踪的更多介绍，请参考《需求管理用户指南》中的“跟踪 Scrum 项目进展”章节。

### 1.4.3 开发代码

代码托管服务提供基于 Git 的在线代码管理服务，包括代码克隆/提交、分支管理等功能。

本实践中采用分支来进行代码的开发。分支是用来将特性开发并行独立出来的工具，使用分支意味着把工作从开发主线上分离出来，以免影响开发主线。

在创建代码仓库时，会有一个默认分支“master”，即主线。为了保证凤凰商城的稳定运行，需要有一个稳定的持续可用 master。因此，开发人员不直接在 master 分支上进行代码开发，而是统一采用功能分支+合并请求的方式，并且每一个功能分支的代码，必须经过团队的其他成员评审后，才可以进行合并。

本章节介绍如何通过分支完成“门店网络查询功能”的开发代码。

#### 创建需求分支

开发人员 Chris 在仓库中创建一个需求分支，用于开发新需求的代码。

**步骤 1** 进入项目“凤凰商城”，单击导航“代码 > 代码托管”，进入代码托管服务。

**步骤 2** 单击代码仓库“phoenix-sample”，进入仓库。

**步骤 3** 选择“代码 > 分支”页签，单击“新建分支”，在弹框中配置以下信息，单击“确定”。

表1-8 新建分支

配置项	示例	说明
基于	master	选择已有的分支。
分支名称	Feature-Store	分支的名称，最长 200 个字节。 <ul style="list-style-type: none"><li>不支持以“-”、“.”、“refs/heads/”、“refs/remotes/”开头。</li><li>不支持空格、“[”、“\”、“&lt;”、“~”、“^”、“:”、“?”、“*”、“!”、“()”、“!”、“”、“ ”等特殊字符。</li><li>不支持以“.”、“/”、“.lock”结尾。</li></ul>
关联 Req 工作项	前端展示 - 添加门店网络菜单	分支关联的工作项，选择已有的工作项。


页面提示操作成功，页面自动跳转至“文件”页面，页面中显示分支“Feature-Store”。

---结束

## 修改并提交代码

开发人员 Chris 在需求分支中完成代码的开发。

**步骤 1** 在代码仓库“phoenix-sample”中的分支“Feature-Store”中，找到“vote/templates/store-network.html”并打开。

**步骤 2** 单击，添加以下代码，并在页面底部文本框中输入提交信息“添加门店列表”，单击“确定”。

```
<ul>
  <li>A 分店: E 机场 1 号航站楼出发层靠右直行 123 米右侧</li>
  <li>B 分店: F 区 G 路 456 号</li>
  <li>C 分店: H 区 J 街 789 号</li>
  <li>D 分店: K 区 L 大道 K 大楼西侧</li>
</ul>
```

保存成功后，页面中显示修改后的代码。

**步骤 3** 以同样方法打开并编辑“/vote/templates/index.html”。

在 179 行添加以下代码，输入提交信息“fix #xxxxxx 前端展示 - 添加门店网络菜单”，单击“确定”。其中，“xxxxxx”是 Task “前端展示 - 添加门店网络菜单”的编号。

```
<li class="nav-item"> <a href="store-network" class="nav-link">门店网络</a> </li>
```

保存成功后，页面中显示修改后的代码。

**步骤 4** 查看提交记录。

1. 单击导航“工作 > 需求管理”，选择“迭代”页签。
2. 在“工作项”页签中找到 Task “前端展示 - 添加门店网络菜单”，Task 的状态已自动更新为“已解决”。
3. 单击 Task 标题打开详情页。
4. 选择“关联”页签，在“代码提交记录”下可看到一条记录，记录的描述与**步骤 3**中输入的提交信息相同。

---结束

## 合并代码

开发人员 Chris 在完成代码开发与自测后，提交分支合并请求，项目经理 Maggie 审核合并请求后完成合入。

**步骤 1** Chris 提交合并请求。

1. 在代码仓库“phoenix-sample”中选择“合并请求”页签，单击“新建合并请求”。
2. 源分支选择“Feature-Store”，目标分支选择“master”，单击“下一步”。
3. 参照下表编辑合并请求详情。

表1-9 合并请求配置

配置项	示例	说明
源分支	Feature-Store	合并请求的源分支。
目标分支	master	合并请求的目标分支。
标题	添加门店网络列表	合并请求的标题，最多 255 个字符。
合并人	Maggie	完成合入代码操作的成员。
审核人	Maggie	审核合并请求的成员。

- 单击“确认”完成合并请求的创建。  
新建成功，页面中显示合并请求的详情。

**步骤 2** Maggie 审核合并请求。

- 进入代码仓库“phoenix-sample”，选择“合并请求”页签，可找到由开发人员 Chris 创建的合并请求。
- 单击该请求，查看合并请求详情。
- 可在页面中留下评审意见。单击入库审核门禁中“通过”完成审核。
- 单击“合入”，将分支合入“master”。  
提示成功，完成合入。

---结束

## 1.4.4 检查代码

代码检查服务提供基于云端实现代码质量管理服务，支持代码静态检查（包括代码质量、代码风格等）和安全检查，并提供缺陷的改进建议和趋势分析。

随着凤凰商城越来越庞大，线上出现的缺陷也越来越多，修复成本太大；且开发人员写代码也比较随性，没有统一标准。因此项目经理建议制定一些基本的标准，并对代码进行持续的静态代码扫描，一旦发现问题立即在迭代内修复。

本章节介绍开发人员 Chris 如何完成针对不同技术栈的代码静态扫描、问题收集与修复。

### 预置任务简介

样例项目中预置了以下 4 个代码检查任务。

表1-10 预置任务

预置任务	任务说明
phoenix-codecheck-worker	检查 Worker 功能对应代码的任务。

预置任务	任务说明
phoenix-codecheck-result	检查 Result 功能对应代码的任务。
phoenix-codecheck-vote	检查 Vote 功能对应代码的任务。
phoenix-sample-javas	检查整个代码仓库对应的 JavaScript 代码的任务。


本章节以任务“phoenix-codecheck-worker”为例进行讲解。

## 配置并执行任务

开发人员可以对样例项目中预置的任务做一些简单的配置，使检查更全面。


本实践中以增加 Python 语言检查规则集为例介绍操作方法。


**步骤 1** 进入项目“凤凰商城”，单击导航“代码 > 代码检查”，页面中显示样例项目内置的 4 个任务。

**步骤 2** 在列表中找到任务“phoenix-codecheck-worker”，在“操作”列中单击，选择“设置”，进入设置页面。

**步骤 3** 单击导航“规则集”，规则集中默认包含的语言是“JAVA”。

**步骤 4** 增加 Python 语言检查规则集。

1. 单击“已包含语言”之后的图标，重新获取代码仓库语言，刷新后的列表新增了多种语言。

2. 将 PYTHON 语言对应的开关状态设置为.

页面提示设置成功，完成规则集的配置。

**步骤 5** 单击“开始检查”，启动任务。

当页面显示 **检查成功**，表示任务执行成功。

如果任务执行失败，请根据报错提示，参考《代码检查常见问题》排查处理。

----结束

## 查看检查结果

代码检查服务提供检查结果统计，并对检查出的问题提供修改建议，可以根据修改建议优化项目代码。

**步骤 1** 在代码检查任务中，选择“概览”页签，即可查看任务执行结果统计。

**步骤 2** 单击“代码问题”页签，即可看到问题列表。

单击问题框中的“问题帮助”，可以查看系统对此问题的修改建议。可以根据需要在代码仓库中找到对应文件及代码位置，参考修改建议优化代码。

图1-7 查看问题帮助



---结束

## 1.4.5 构建应用并推送镜像至 SWR

编译构建服务提供配置简单的混合语言构建平台，支持任务一键创建、配置和执行，实现获取代码、构建、打包等活动自动化。

在项目部署过程中，经常遇到由于环境不一致而导致的部署失败，例如研发调试环境的 JDK 升级后，未在环境清单中标记清楚，导致生产环境未做相应升级而引发失败。为了避免因为环境不一致导致的各种问题，本样例项目中将各微服务应用与环境统一打包到镜像，保持环境（开发调测环境、测试环境、QA 环境、生产环境）一致。

本章节介绍开发人员 Chris 如何构建并归档镜像至 SWR。

### 预置任务简介

样例项目中预置了以下 5 个构建任务。

表1-11 预置任务

预置任务	任务说明
phoenix-sample-ci	基本的构建任务。
phoenix-sample-ci-test	构建测试环境可用镜像的任务。
phoenix-sample-ci-worker	构建 Worker 功能镜像的任务。
phoenix-sample-ci-result	构建 Result 功能镜像的任务。

预置任务	任务说明
phoenix-sample-ci-vote	构建 Vote 功能镜像的任务。

本章节以任务“phoenix-sample-ci”为例进行讲解，任务内置的步骤及作用如下表所示，用户完成任务的参数配置后即可执行该任务。

表1-12 构建步骤

构建步骤	说明
制作 Vote 镜像并推送到 SWR 仓库	依据代码仓库中的“vote/Dockerfile”文件制作 Vote 功能镜像，并将镜像推送到容器镜像服务。
制作 Result 镜像并推送到 SWR 仓库	依据代码仓库中的“result/Dockerfile”文件制作并推送 Result 功能镜像，并将镜像推送到容器镜像服务。
使用 Maven 安装 Worker 依赖包	使用 Maven 安装 Worker 功能所需的依赖。
制作 Worker 镜像并推送到 SWR 仓库	依据代码仓库中的“worker/Dockerfile”文件制作并推送 Worker 功能镜像，并将镜像推送到容器镜像服务。
生成 Postgres and Redis Dockerfile	通过 shell 命令生成 Dockerfile 文件，用以制作 Postgres（数据库）和 Redis（缓存）镜像。
制作 Postgres 镜像并推送到 SWR 仓库	依据“生成 Postgres and Redis Dockerfile”步骤中所生成的 Dockerfile 文件制作 Postgres 镜像，并将镜像推送到容器镜像服务。
制作 Redis 镜像并推送到 SWR 仓库	依据“生成 Postgres and Redis Dockerfile”步骤中所生成的 Dockerfile 文件制作 Redis 镜像，并将镜像推送到容器镜像服务。
替换 Docker-Compose 部署文件镜像版本	<p>为了将镜像部署到 ECS 时，能够拉取到正确的镜像，使用 shell 命令进行完成以下操作。</p> <ol style="list-style-type: none"> <li>1. 使用 sed 命令，依次将文件“docker-compose-standalone.yml”中的参数替换为构建任务的参数“dockerServer”、“dockerOrg”、“BUILDNUMBER”进行替换。</li> <li>2. 使用 tar 命令，将文件“docker-compose-standalone.yml”压缩为“docker-stack.tar.gz”，将部署所需文件进行打包，以便于后续步骤将该文件上传归档。</li> </ol>
替换 Kubernetes 部署文件镜像版本	<p>为了将镜像部署到 CCE 时，能够拉取到正确的镜像，使用 shell 命令进行完成以下操作。</p> <ol style="list-style-type: none"> <li>1. 使用 sed 命令，将代码仓库中目录“kompose”下所有以“deployment”结尾的文件中的参数“docker-server”、</li> </ol>

构建步骤	说明
	“docker-org”，替换为构建任务的参数“dockerServer”、“dockerOrg”。 2. 使用 sed 命令，将代码仓库中“result-deployment.yaml”、“vote-deployment.yaml”、“worker-deployment.yaml”三个文件中的参数“image-version”用构建任务参数“BUILDNUMBER”进行替换。
上传 Kubernetes 部署文件到软件发布库	将“替换 Kubernetes 部署文件镜像版本”步骤中修改后的所有“.yaml”文件上传到软件发布库中归档。
上传 docker-compose 部署文件到软件发布库	将“替换 Docker-Compose 部署文件镜像版本”步骤中压缩好的“docker-stack.tar.gz”上传到软件发布库中归档。

## 配置 SWR 服务

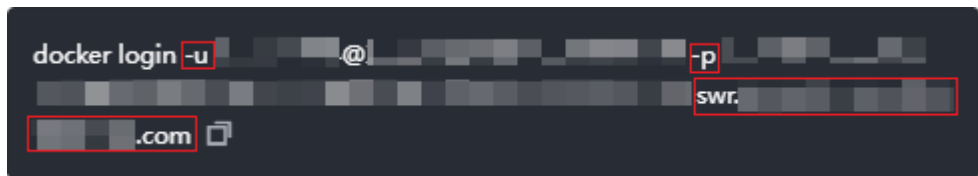
步骤 1 登录 SWR 控制台。

步骤 2 单击“登录指令”，页面弹框显示登录指令。

其中，

- “-u”之后的字符串为用户名。
- “-p”之后的字符串为密码。
- 最后的字符串为 SWR 服务器地址，此地址即为后续配置并执行任务中的参数“dockerServer”。

图1-8 登录指令



步骤 3 单击“创建组织”，在弹框中输入组织名称“phoenix”（如果页面提示“组织已存在”，请自定义其他名称），单击“确定”。

新建成功后，在左侧导航栏中单击“组织管理”，在组织管理页面中显示已创建的组织。

----结束

## 配置并执行任务

步骤 1 进入“凤凰商城”项目，单击导航“持续交付 > 编译构建”。页面中显示样例项目内置的任务。



步骤 2 在列表中找到任务“phoenix-sample-ci”。单击图标，选择“编辑”。

步骤 3 选择“参数设置”页签，为以下参数配置默认值。

表1-13 参数设置

名称	默认值
codeBranch	master。
dockerOrg	输入在配置 SWR 服务中创建的组织名称。
version	1.0.0
dockerServer	输入在配置 SWR 服务中获取的 SWR 服务器地址。

步骤 4 单击“保存并执行”，在弹框中单击“确定”，启动构建任务。

当页面中显示时，表示任务执行成功。请记录以“#”开头的字符串（例如 #20230401.1 ）。

如果构建失败，请根据失败步骤信息与日志中的报错，参考《编译构建常见问题》处理。

步骤 5 检查发布件。

- 单击导航“制品仓库 > 软件发布库”，进入软件发布库。
- 在与项目同名的仓库中，可以找到“docker-stack”、“phoenix-sample-ci”两个文件夹。
  - 在文件夹“docker-stack/1.0.0”中，可以找到发布件“docker-stack.tar.gz”。
  - 在文件夹“phoenix-sample-ci/1.0.0”中，可以找到归档的 10 个“.yaml”格式文件。
- 进入 SWR 控制台，在导航中选择“组织管理”，单击在配置 SWR 服务中创建的组织名称。  
选择“镜像”页签，可以在列表中找到 5 个镜像“redis”、“postgres”、“worker”、“result”、“vote”。
- 依次在列表中单击 5 个镜像的名称进入详情页。在“镜像版本”页签中查看镜像版本。
  - redis 的镜像版本为“alpine”。
  - postgres 的镜像版本为“9.4”。
  - worker、result、vote 的镜像版本均与在步骤 4 中记录的字符串相同。

---结束

## 1.4.6 部署应用至 CCE

部署服务提供可视化、自动化部署服务。提供丰富的部署步骤，有助于用户制定标准的部署流程，降低部署成本，提升发布效率。

为了可以更快的、更稳定的持续交付软件，开发团队需要一部分自助化部署服务的能力，以减轻部分后续维护工作。

本章节介绍开发人员 Chris 如何将发布件部署至云容器引擎。如果您需要了解如何部署至 ECS，请参照[部署应用至 CCE](#)操作。

关于 CCE 的准备，请参考[资源规划](#)。

### 预置应用简介

样例项目中预置了以下 3 个部署应用。

表1-14 预置应用

预置应用	应用说明
phoenix-cd-cce	部署至 CCE 流程对应的应用。
phoenix-sample-standalone	部署至 ECS 流程对应的应用。
phoenix-sample-predeploy	向 ECS 中安装依赖工具操作对应的应用。

本章节以应用“phoenix-cd-cce”为例进行讲解。

### 检查 CCE 集群

在部署应用前，需保证 CCE 集群中没有工作负载运行。

步骤 1 登录 CCE 控制台。

步骤 2 单击集群“phoenix-cce”，进入集群。

步骤 3 在导航中单击“工作负载”，选择“无状态负载”页签，确认列表中无记录。

如果列表中有记录，则勾选全部记录后，单击“批量删除”。然后在弹框中勾选全部资源释放选项，单击“是”，将列表记录清空。

---结束

### 配置并执行应用

步骤 1 进入“凤凰商城”项目，单击导航“持续交付 > 部署”，页面中显示样例项目内置的应用。

步骤 2 找到应用“phoenix-cd-cce”。单击图标\*\*，选择“编辑”。

步骤 3 选择“部署步骤”页签，在每个步骤中完成以下配置。

表1-15 配置部署步骤

配置项	示例	说明
集群名称	phoenix-ccc	目标集群的名称。
命名空间名称	default	目标集群中的命名空间。

步骤 4 选择“参数设置”页签，检查参数的默认值是否如下表所示。

表1-16 参数设置

名称	默认值
ci_task_name	phoenix-sample-ci
version	1.0.0

步骤 5 单击“保存并部署”，在弹框中单击“确定”，启动部署应用。

当出现页面提示“部署成功”时，表示部署成功。

如果部署失败，请根据失败步骤信息与日志中的报错信息，参考《部署常见问题》排查处理。

----结束

## 验证部署结果

步骤 1 进入 CCE 控制台。

步骤 2 单击集群“phoenix-ccc”，进入集群。

步骤 3 在导航中单击“工作负载”，选择“无状态负载”页签。

页面中显示 5 条记录，状态均为“运行中”。

步骤 4 单击“vote”进入详情页，在“访问方式”页签中单击“更多 > 更新”。

步骤 5 完成以下配置，单击“确定”。

配置项详细说明请参考《云容器引擎用户指南》中的“负载均衡(LoadBalancer)”章节。

表1-17 更新服务

配置项	示例
服务亲和	集群级别

配置项	示例
负载均衡器	1. 选择“共享型 > 自动创建”。 2. 实例名称：输入“phoenix”。 3. 弹性公网 IP：选择“自动创建”。
端口配置	<ul style="list-style-type: none"><li>容器端口：80。</li><li>服务端口：5000。</li></ul>



步骤 6 更新成功，返回列表中，刷新页面。当列表中显示  phoenix  时，鼠标悬停在该负载均衡器名称处，在弹窗中复制公网地址。

图1-9 复制访问地址



步骤 7 打开新的浏览器页面，在地址栏中输入“http://IP:端口号”（其中，IP 为步骤 6 记录的公网地址），页面中显示商城主页。

步骤 8 返回“无状态负载”页面，参照步骤 4 更新“result”（其中，负载均衡器选择已创建的“phoenix”，服务端口输入“5001”）。

创建成功后，在新的浏览器页面中输入“http://IP:5001”，页面中显示商城仪表盘。

---结束

## 1.4.7 部署应用至 ECS

本章节以应用“phoenix-sample-standalone”为例，介绍如何将发布件部署至主机。如果您需要了解如何部署至 CCE，请参照[部署应用至 CCE](#)操作。

关于 ECS 的准备，请参考[资源规划](#)。

## 添加目标主机至项目

部署应用到 ECS 之前，需要先将目标主机添加到项目基础资源中。

- 步骤 1** 进入项目“凤凰商城”，单击导航栏“设置 > 通用设置 > 基础资源管理”，进入基础资源管理页面。
- 步骤 2** 单击“新建主机集群”，配置以下信息，单击“保存”。

表1-18 新建主机集群

配置项	示例	说明
集群名称	phoenix-hostgroup	自定义主机集群名称，仅支持 3-128 位数字、中文、英文字母、“-”、“_”、“.”。
操作系统	Linux	选择加入集群的主机的操作系统类型，可以选择“Linux”或“Windows”。
主机联通方式	直连模式	选择部署服务与目标主机的联通方式，可以选择“直连模式”或“代理模式”。
执行主机	默认资源池	资源池是部署软件包时执行部署命令的物理环境的集合。 可以选择服务提供的“默认资源池”，也可以将自有的服务器作为资源池。

- 步骤 3** 待保存成功，在“目标主机”页签中单击“添加或导入主机”。
- 步骤 4** 选择添加方式“导入已购 ECS”，找到[资源规划](#)时准备的主机“phoenix-ecs”，在“操作”列中单击“导入”。
- 步骤 5** 配置以下信息，单击“确定”。

表1-19 添加主机

配置项	示例	说明
认证方式	选择“密码”。	连接 ECS 的认证方式，支持密码和密钥两种方式。
用户名	输入“root”。	ECS 的登录用户名，Linux 主机通常为“root”。
密码	输入在购买 ECS 时设置的密码。	ECS 的登录密码。
ssh 端口	输入“22”。	通常为“22”，也可以自定义端口。

**步骤 6** 保存成功，目标主机列表中新增一条主机记录，当“连通性验证”列的值显示为“成功”时，表示主机添加完成。

如果主机添加失败，请根据失败详情，参考《部署常见问题》中“主机/代理机连通性验证问题排查方法有哪些？”章节排查处理。

---结束

## 在 ECS 中安装依赖工具

样例程序的运行需要 Docker 及 Docker-Compose 环境，需要将依赖环境安装到目标 ECS 中。

**步骤 1** 进入“凤凰商城”项目，单击导航“持续交付 > 部署”，进入部署服务。

**步骤 2** 在应用“phoenix-sample-predeploy”的“操作”列中单击\*\*>

**步骤 3** 选择“环境管理”页签，配置主机环境。

1. 单击“新建环境”，配置以下信息，单击“保存”。

表1-20 新建环境

配置项	示例	说明
环境名称	phoenix-env	自定义环境名称，仅支持 3-128 位数字、中文、英文字母、“-”、“_”、“.”。
资源类型	主机	环境中的资源类型，默认为“主机”。
操作系统	Linux	选择即将添加到环境中的主机的操作系统类型，可以选择“Linux”或“Windows”。

2. 待保存成功，在“资源列表”页签中单击“导入主机”，在弹框中选择已配置好的主机集群与主机，单击“导入”。

3. 页面提示导入成功，资源列表中显示新增的主机。

关闭此窗口，环境列表中显示新建的环境。

**步骤 4** 在“部署步骤”页签，编辑应用的步骤。

1. 选择步骤“安装 Docker”，在环境下拉列表中选择“phoenix-env”。如果页面显示弹框“是否将后续步骤的环境也修改为 phoenix-env? ”，单击“确定”。

2. 选择“执行 shell 命令”，在“shell 命令”框中添加以下两行命令。

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

**步骤 5** 单击“保存并部署”，启动部署任务。

当出现页面提示“部署成功”时，表示任务执行成功。如果部署失败，请根据失败步骤信息与日志中的报错信息，参考《部署常见问题》排查处理。

- 步骤 6 查看日志，找到如下图所示的日志内容，表示说明安装 Docker 及 Docker-Compose 成功。

图1-10 查看部署日志

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

---结束

## 配置并执行应用

- 步骤 1 进入“凤凰商城”项目，单击导航“持续交付 > 部署”，进入部署服务。
- 步骤 2 在应用“phoenix-sample-standalone”的“操作”列中单击\*\*\*，选择“编辑”，进入编辑页面。
- 步骤 3 选择“环境管理”页签，配置主机环境。
1. 单击“新建环境”，配置以下信息，单击“保存”。

表1-21 新建环境

配置项	示例	说明
环境名称	phoenix-env	自定义环境名称，仅支持 3-128 位数字、中文、英文字母、“-”、“_”、“.”。
资源类型	主机	环境中的资源类型，默认为“主机”。
操作系统	Linux	选择即将添加到环境中的主机的操作系统类型，可以选择“Linux”或“Windows”。

2. 待保存成功，在“资源列表”页签中单击“导入主机”，在弹框中选择已配置好的主机集群与主机，单击“导入”。
  3. 页面提示导入成功，资源列表中显示新增的主机。  
关闭此窗口，环境列表中显示新建的环境。
- 步骤 4 在“部署步骤”页签，编辑应用的步骤。
1. 选择步骤“选择部署来源”，完成以下配置。

表1-22 部署来源配置

配置项	示例	说明
选择源类型	构建任务	软件包的来源，可以选择“制品仓库”或“构建任务”。
环境	phoenix-env 如果页面显示弹框“是否将后续步骤的环境也修改为 phoenix-env? ”，单击“确定”。	目标部署环境，可以选择“环境管理”页签添加的环境。
请选择构建任务	phoenix-sample-ci	当“选择源类型”为“构建任务”时，显示此配置项。

步骤 5 选择“参数设置”页签，为以下参数配置默认值。

表1-23 参数设置

名称	默认值
docker_server	输入通过配置 SWR 服务获取的 SWR 服务器地址。
docker_username	输入通过配置 SWR 服务获取的用户名。
docker_password	输入通过配置 SWR 服务获取的密码。

步骤 6 单击“保存并部署”，启动部署任务。

当出现页面提示“部署成功”时，表示任务执行成功。如果部署失败，请根据失败步骤信息与日志中的报错信息，参考《部署常见问题》排查处理。

步骤 7 验证部署结果。

1. 打开浏览器，输入“http://IP:5000”，其中 IP 为 ECS 的弹性公网 IP 地址，页面中显示商城主页。
2. 输入“http://IP:5001”，页面中显示商城仪表盘。

---结束

## 1.4.8 管理项目测试

测试计划服务提供一站式云端测试平台，融入 DevOps 敏捷测试理念，有助于高效管理测试活动，保障产品高质量交付。

本章节介绍测试人员 Billy 如何管理项目的测试周期，包括创建测试计划、设计与执行测试用例等。

## 创建测试计划

在确定迭代 4 中计划实现的需求（Story）后（即完成[管理项目规划](#)），测试人员即可在开发人员进行代码开发的同时编写测试用例。

**步骤 1** 进入“凤凰商城”项目，单击导航“测试 > 测试计划”。

**步骤 2** 单击“新建计划”，配置测试计划信息。

1. 基本信息：配置以下信息，单击“下一步”。

表1-24 测试计划基本信息

配置项	示例
名称	迭代 4 测试计划
处理者	Billy
计划周期	建议与在需求管理中创建的“迭代 4”的周期一致。
迭代	迭代 4

2. 高级配置：勾选“手工测试”。确认列表中的需求与需求管理中“迭代 4”的需求一致，单击“保存并使用”。

**步骤 3** 返回测试计划页面，在页面中可找到新创建的测试计划“迭代 4 测试计划”，状态为“新建”。

---结束

## 设计测试用例

**步骤 1** 在测试计划“迭代 4 测试计划”中，单击“设计”，进入测试用例页面。

**步骤 2** 展开页面左侧“需求目录”，找到 Story “作为用户应该可以查询所有门店网络”。

单击图标, 选择“新建测试用例”。

图1-11 新建测试用例



步骤 3 输入名称“门店网络查询”，编辑测试步骤与预期结果，单击“保存”。

表1-25 测试步骤

测试步骤	预期结果
打开凤凰商城首页。	页面正常显示。
单击菜单“门店网络”。	进入“门店网络”界面，页面中存在省份筛选，页面最下面显示推荐门店信息。
城市选择“A市”。	列出A市的门店信息列表。

步骤 4 单击导航“测试 > 测试计划”，进入测试计划页面。

在页面中找到测试计划“迭代 4 测试计划”，此时测试计划的状态更新为“设计中”。

---结束

## 执行测试计划

当开发人员完成 Story 的代码开发、并将应用部署到测试环境后（即完成[部署应用至 CCE](#) 或[部署应用至 ECS](#)），可将 Story 的状态设置为“测试中”，并将 Story 的处理人设置为测试人员。

此时测试人员即可开始执行 Story 对应的测试用例。


步骤 1 在测试计划“迭代 4 测试计划”中，单击“执行”，进入测试执行页面。

步骤 2 单击“新建套件”，配置以下信息，单击“保存”。

- 输入名称“执行迭代 4 测试用例”。
- 单击“添加用例”，在弹框中勾选已创建的用例，单击“确定”，列表中显示已添加的用例。

保存成功后，返回列表，页面中显示新增的套件。

步骤 3 进入“测试 > 测试用例”页面，在页面上方选择“迭代 4 测试计划”。

步骤 4 单击“操作”列中 ，页面右侧滑出“执行”窗口。

步骤 5 在测试环境中，按照测试步骤进行逐步操作。

- 如果执行成功：将所有步骤的“设置结果”更新为“成功”，页面提示更新成功，完成用例的执行。

图1-12 执行成功

设置结果 

\* 设置用例结果

成功 

序号	步骤描述	预期结果	设置结果
1	打开凤凰商城首页。	页面正常显示。	成功 点击输入结果信息
2	单击菜单“门店网络”。	进入“门店网络”界面，页面中存在省份筛选，页面最下面显示推荐门店信息。	成功 点击输入结果信息
3	在页面中查找A市门店信息。	可以找到A市的门店信息。	成功 点击输入结果信息

- 如果执行失败：将操作失败的步骤的“设置结果”更新为“失败”。  
假设执行第二步时页面跳转失败，页面显示 404。
  - a. 将第一步的“设置结果”更新为“成功”。
  - b. 将第二步的“设置结果”更新为“失败”，并输入“跳转失败，页面显示 404”。
  - c. 将“设置用例结果”更新为“失败”。

图1-13 执行失败

设置结果 

\* 设置用例结果

失败 

序号	步骤描述	预期结果	设置结果
1	打开凤凰商城首页。	页面正常显示。	成功 点击输入结果信息
2	单击菜单“门店网络”。	进入“门店网络”界面，页面中存在省份筛选，页面最下面显示推荐门店信息。	失败 跳转失败，页面显示404
3	在页面中查找A市门店信息。	可以找到A市的门店信息。	请选择 点击输入结果信息

- d. 待页面提示更新成功后，单击“新建缺陷”，打开新建 Bug 页面。
- e. 输入标题“门店网络页面显示 404”，设置当前责任人为“Chris”，单击“确定”。  
新建成功，页面中显示新建的 Bug。
- f. 待开发人员 Chris 修复缺陷后，重新验证该用例。

验证成功后，将步骤 5.e 中创建的 Bug 的状态设置为“已关闭”，并将测试用例的执行结果更新为“成功”。

步骤 6 单击导航“测试 > 测试计划”，进入测试计划页面。

在页面中找到测试计划“迭代 4 测试计划”，此时测试计划的状态更新为“完成”。

---结束

## 跟踪测试计划进展

CodeArts 测试计划服务提供了质量看板，可视化展示需求覆盖率、缺陷分布统计、用例通过率、用例完成率等的测试统计数据。

项目组可以通过看板查看测试计划的当前进展、评估产品质量，也可以根据需要创建自定义报表。

关于测试计划进展跟踪的更多介绍，请参考《测试计划用户指南》中的“查看与评估 CodeArts TestPlan 测试质量”章节。

## 1.4.9 配置流水线

流水线服务提供可视化、可定制的自动交付软件生产线，支持代码检查、构建、部署等多种任务类型。

随着项目的进行，构建、发布、部署等环节越来越标准化。但是每个环节都相对独立，仅是半成品，无法直接交付业务价值。只有将这些环节有效串联，形成完整的持续交付流水线，才能提升软件发布效率与质量，持续创造业务价值。

本章节介绍开发人员 Chris 如何将代码检查、构建、部署任务串联起来，实现持续交付。

## 预置流水线简介

示例项目中预置以下 5 个流水线任务，可根据需要查看并使用。


表1-26 预置流水线任务

预置流水线任务	任务说明
phoenix-workflow	基本的流水线任务。
phoenix-workflow-test	测试环境对应的流水线任务。
phoenix-workflow-work	Worker 功能对应的流水线任务。
phoenix-workflow-result	Result 功能对应的流水线任务。
phoenix-workflow-vote	Vote 功能对应的流水线任务。

## 配置并执行流水线

如果使用“phoenix-cd-ccc”部署应用，在执行流水线前，请参考[检查 CCE 集群](#)，将集群内的工作负载清空。

**步骤 1** 进入“凤凰商城”项目，单击导航“持续交付 > 流水线”，进入流水线服务。

**步骤 2** 在流水线“phoenix-workflow”的“操作”列单击 ，选择“编辑”，进入编辑页面。

**步骤 3** 选择“参数设置”页签，确认参数“dockerOrg”、“dockerServer”的默认值与构建任务“phoenix-sample-ci”保持一致。

**步骤 4** 添加代码检查阶段。



1. 单击“流水线源”与“构建”之间的 ，添加阶段。
2. 单击“阶段\_1”的 ，在“编辑阶段”窗口中输入阶段名称“代码检查”，单击“保存”。

图1-14 编辑阶段名称



3. 单击“添加任务”。  
在“添加任务”窗口中，配置以下信息，单击“保存”。

表1-27 代码检查任务配置

配置项	配置建议
类型	选择“代码检查”。
名称	输入自定义名称。
请选择需要调用的任务	选择“phoenix-codecheck-worker”。

配置项	配置建议
检查模式	提供三种检查模式，根据需要选择。 <ul style="list-style-type: none"><li>• Full：全量检查，扫描代码仓里的所有文件。</li><li>• Incremental（last commit）：增量检查，基于最近一次 commit 文件进行扫描。</li><li>• Incremental（last success）：增量检查，基于最近一次门禁通过后的变更文件进行扫描。</li></ul>

**步骤 5** 配置部署任务。

单击部署任务名称，在窗口中选择关联构建任务“phoenix-sample-ci”，并检查配置项的值。


- 任务“phoenix-sample-standalone”的配置需与部署服务中同名任务的“参数设置”页面内容保持一致。
- 任务“phoenix-cd-ccc”的配置需与部署服务中同名任务“参数设置”页面内容保持一致。

**说明**

部署任务中添加了两个部署任务，如果您在之前的步骤中只选择了一种部署方式，请保留对应的部署任务，将另一个删除。

**步骤 6** 配置完成，单击“保存”。

**步骤 7** 单击“保存并执行”，在弹框中单击“执行”，启动流水线。

当页面中显示时，表示任务执行成功。

如果任务执行失败，可单击失败原因提示，打开日志，参考《流水线常见问题》排查处理。

---结束

## 1.4.10 释放资源

**警告**

资源释放后无法恢复，请谨慎操作。

本文中涉及的按需计费资源有 CCE、ECS，如果完成实践后不再使用这些资源，建议释放。

表1-28 释放资源

资源名称	操作指导
------	------

资源名称	操作指导
ECS	参考《弹性云主机常见问题》中“创建与删除”章节。
CCE	参考《云容器引擎用户指南》中的“删除集群（按需计费）”章节。

