软件开发生产线

最佳实践

文档版本 11

发布日期 2023-09-06

华为云计算技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

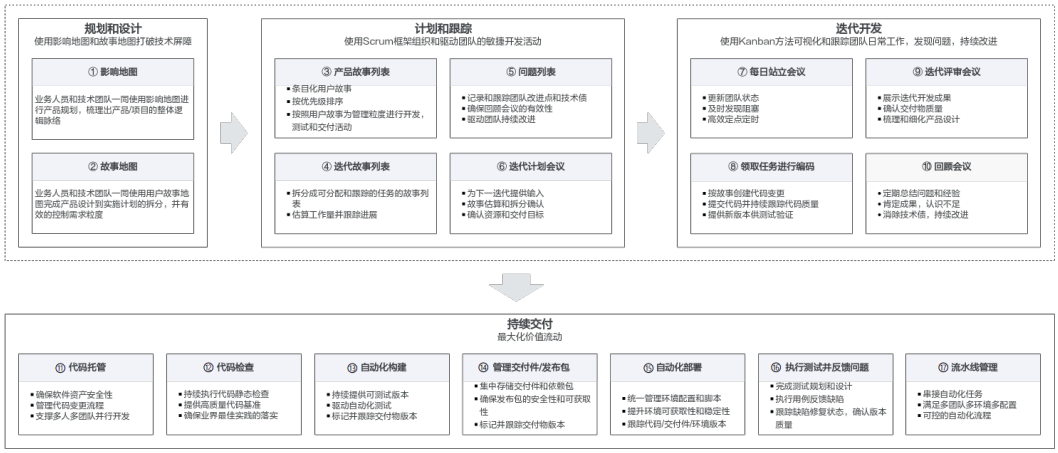
1 使用 CodeArts 管理电子商城项目开发流程

1.1 方案概述

背景信息

CodeArts结合多年研发经验与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。

应用场景

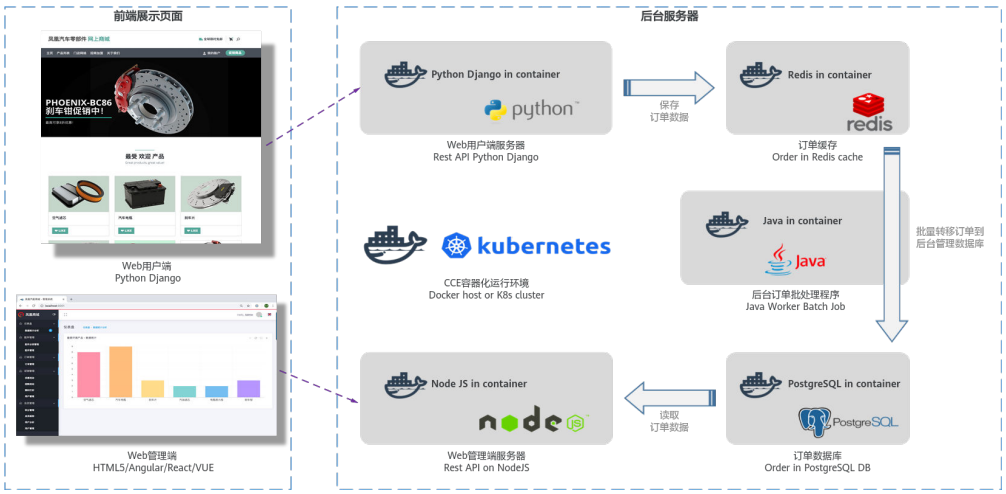
通过一套汽车零部件配件电子商城示例代码“凤凰商城”，以及“DevOps全流程示例项目”，介绍如何使用CodeArts实现HE2E DevOps框架。该方案适用于Scrum研发项目。

方案架构

- “凤凰商城”示例程序架构

  “凤凰商城”示例程序的架构图如图1-2所示。

图 1-2 凤凰商城技术架构图



示例程序由表1-1中的5个可以独立开发、测试和部署的微服务组件构成。

表 1-1 凤凰商城微服务组件表

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）	• 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。 • 技术栈：Python、Flask框架。 • 应用服务器：Gunicorn。
Web管理端服务器（对应样例代码中的“Result”功能）	• 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。 • 技术栈：Node.js、express框架。 • 应用服务器：server.js。
后台订单批处理程序（对应样例代码中的“Worker”功能）	• 业务逻辑：此服务为后台进程，会监控Redis缓存中物品记录，并将新纪录取出并保存在PostgreSQL数据库中，以便管理端UI可以抽取数据进行统计显示。 • 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）。
订单缓存	• 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。 • 技术栈：Redis
订单数据库	• 业务逻辑：此服务作为管理端UI服务的数据源。 • 技术栈：PostgreSQL

- “DevOps全流程样例项目”构成

  “DevOps全流程样例项目”是一个Scrum类型的模板项目，项目中预置了部分服务的使用模板。项目实践过程中涉及到的产品及服务如下表。

表 1-2 实践涉及产品/服务列表

服务		说明
软件开发生产线	需求管理	预置3个已规划并已完成的迭代、项目的模块设置、以及若干统计报表。
	代码托管	预置代码仓库“phoenix-sample”，存放项目示例代码。
	代码检查	预置4个任务，任务详情介绍请参见**步骤四：检查代码**。
	编译构建	预置5个任务，任务详情介绍请参见**步骤五：构建应用**。

服务		说明
	制品仓库	用于存储通过构建任务生成的软件包。
	部署	预置3个应用，应用详情介绍请参见**步骤六：部署应用（CCE篇）**。
	测试计划	功能测试用例库，预置十余个测试用例。
	流水线	预置5条流水线，流水线详情介绍请参见**步骤八：配置流水线，实现持续交付**。
其它组件和服务	统一身份认证服务	用于管理账号。
	容器镜像服务	用于存放构建任务生成的Docker镜像。
	云容器引擎	用于软件包部署，与ECS部署属于两种不同的部署方式。
	弹性云服务器	用于软件包部署，与CCE部署属于两种不同的部署方式。

方案优势

- 针对需求变动频繁、开发测试环境复杂、多版本分支维护困难、无法有效监控进度和质量等研发痛点，提供一站式云端管理平台，管理软件开发全过程。
- 提供可视化、可定制的持续交付流水线服务，实现持续交付，让软件上线提速一倍。

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

资源名称	数量
软件开发生产线 CodeArts	开通基础版即可。
云容器引擎 CCE	1
弹性云服务器 ECS	1

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 HE2E DevOps 实践操作流程图

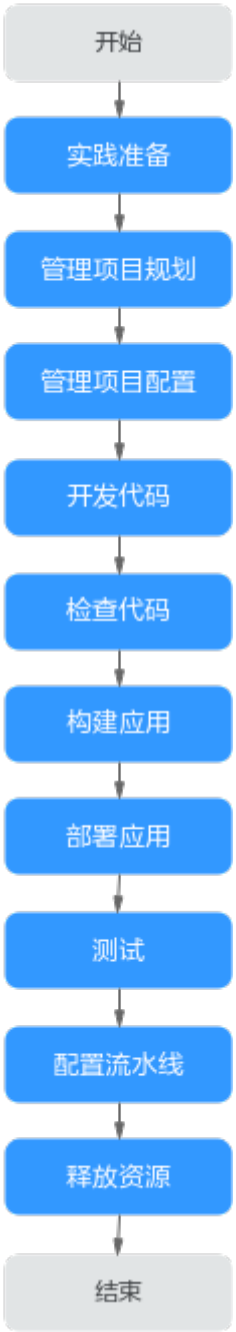


表 1-4 HE2E DevOps 实践操作流程说明

步骤	说明
实践准备	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	完成项目的整体规划，包括项目需求规划、迭代需求规划等。
管理项目配置	根据项目需求，对工作项变更的通知方式、工作项状态的流转方式等进行自定义设置。

步骤	说明
开发代码	通过分支来进行代码的编写，包括创建分支、代码提交、合并分支等操作。
检查代码	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	构建环境镜像、将代码编译打包成软件包。
部署应用	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	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	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	实践完成，释放CodeArts、CCE等资源。

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。

购买 CodeArts

完成本实践全部操作，需购买CodeArts基础版套餐。

步骤1 进入**购买CodeArts套餐页面**。

步骤2 选择“基础版”，购买人数保持默认值，购买时长选择“1个月”，勾选同意声明，单击“下一步”。

步骤3 确认订单内容，单击“去支付”。

步骤4 根据页面提示完成支付。

步骤5 开通成功，返回“软件开发生产线”页面，列表中显示已开通套餐记录。

----结束

创建项目

在开展项目实践前，由产品负责人Sarah创建项目。

步骤1 在CodeArts控制台单击“立即使用”。

步骤2 单击“新建项目”，选择“DevOps全流程示例项目”。

说明

如果登录后页面中展示的是项目类型，选择“DevOps全流程示例项目”即可。

步骤3 输入项目名称“凤凰商城”，单击“确定”，完成项目创建。

----结束

添加项目成员

由产品负责人Sarah为团队成员创建账号，并添加项目中。

本样例项目涉及四个项目角色，为了方便介绍，本文档中每个角色对应一个人，如表1-5所示。

表 1-5 项目角色列表

项目成员	项目角色	工作职责
Sarah	产品负责人（项目创建者）	负责产品整体规划与产品团队的组建。
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。

步骤1 进入“凤凰商城”项目，进入“设置 > 通用设置 > 服务权限管理 > 成员”页面。

步骤2 单击项目成员列表上方“添加成员 > 从本企业导入用户”。

步骤3 在弹框中单击“创建用户”，跳转至“用户”页面。

图 1-4 添加成员

步骤4 单击“创建用户”，依次创建三个用户“Maggie”、“Chris”、“Billy”。

步骤5 返回CodeArts，刷新浏览器，重新单击项目成员列表上方“添加成员 > 从本企业导入用户”，勾选成员“Maggie”、“Chris”、“Billy”，单击“下一步”。

步骤6 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”。

----结束

1.4.2 步骤一：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。

本样例项目采用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，当前正在进行迭代4的规划。

按照项目规划，迭代4要完成的功能为：限时打折管理、团购活动管理。

由于业务与市场的变化，临时新增一个紧急需求：门店网络查询功能，因此迭代4的规划中增加此功能的开发。

通过本章节，您将了解产品负责人Sarah与项目经理Maggie如何进行项目规划的管理，包括管理需求规划与迭代规划、跟踪项目进度。

管理需求规划

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，各层级工作项类型代表的含义如表1-6所示。

表 1-6 工作项类型说明

工作项类型	说明
Epic	通常是公司重要战略举措，比如本样例项目中的“凤凰商城”，对于公司是一个与企业生存攸关的关键战略措施。
Feature	通常是对客户有价值的功能，可以通过使用特性满足客户的需求。比如凤凰商城中的“门店网络查询功能”，特性通常会通过多个迭代持续交付。
Story	通常是对一个功能进行用户场景细分，并且能在一个迭代内完成。
Task	通常是用户故事的细分，准备环境、准备测试用例等都可以是完成Story的细分任务。

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，因此产品负责人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”，在页面中选择“规划”页签。
2. 单击“凤凰商城思维导图”。

说明

如果“规划”页签中显示为空白，请创建思维导图。

1. 单击 

，在下拉列表中选择“思维导图规划”。
在弹框中输入名称“需求规划”，单击“确定”，页面跳转至思维导图详情。
2. 单击“添加Epic”，在弹框中勾选“凤凰商城”，单击“确定”。

3. 新建Feature“门店网络”。

   a. 在Epic“凤凰商城”下方单击图标。

   b. 输入标题“门店网络”，回车保存。

图 1-5 新建 Feature

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 1-7 Story 配置

配置项	配置建议
描述信息	输入“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”。
优先级	选择“高”。
重要程度	选择“关键”。

2. 为了便于开发人员理解，在本地准备一个“门店网络列表”文件（本文档中为excel表，表格内容参照下表）。

表 1-8 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。

----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。

通过本节，您将了解如何完成迭代4的规划。

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”，在页面中选择“迭代”页签。
2. 单击页面左上角“迭代”字样后的  ，参照**表1-9**在弹框中配置迭代信息，单击“确定”。

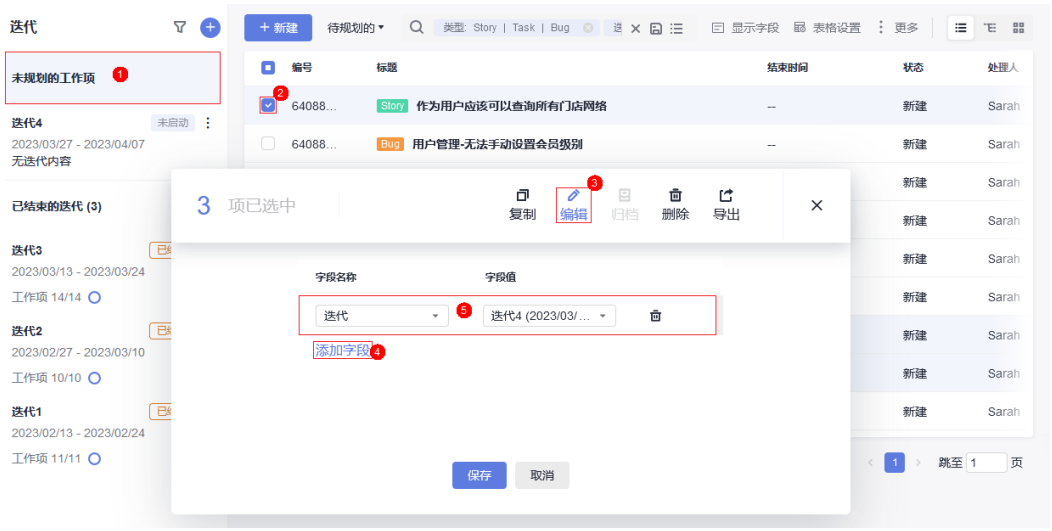
表 1-9 迭代信息配置

配置项	配置建议
迭代名称	输入“迭代4”。
计划时间	设置时长为2周。

步骤2 规划迭代。

1. 单击页面左侧导航“未归划的工作项”。
2. 根据规划，在列表中勾选以下三个Story。
   - 作为用户应该可以查询所有门店网络。
   - 作为管理员应该可以添加团购活动。
   - 作为管理员应该可以添加限时打折。
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



步骤3 分配Story。

1. 单击页面左侧导航“迭代4”。
2. 选择全部Story，参照**规划迭代**的方式，将字段“处理人”的值设置为“Chris”。

步骤4 分解Story。

1. 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名称。
2. 在页面右侧滑出窗口中选择“子工作项”页签。
3. 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”完成。
4. 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

- 每日站立会议跟踪任务进度。

  迭代开始后，项目组通过每日站立会议沟通每个工作项的当前进展，并对工作项状态进行更新。

  使用卡片模式能够简单直观的查看迭代中各工作项的当前状态。

  进入“迭代”页面，单击图标，切换到卡片模式。页面中展示了处于每种状态下的工作项卡片，通过拖拽工作项卡片即可更新其状态。

- 迭代评审会议验收迭代成果。

  在到达迭代的预计结束时间前，项目组召开迭代评审会议，展示当前迭代的工作成果。

  “迭代”页面提供了迭代统计图表，团队可以方便的统计当前迭代的进度情况，包括需求完成情况、迭代燃尽图、工作量等。

  进入“迭代”页面，单击“统计”，即可展开迭代进度视图。

1.4.3 步骤二：管理项目配置

管理项目通知

项目经理Maggie希望当任务（工作项）分配给团队成员时，该成员能够收到通知，以便及时处理。

步骤1 进入“凤凰商城”项目，单击导航“设置 > 工作配置 > 通知设置”。

步骤2 页面中显示样例项目中的默认配置。

由于默认配置可以满足需求，因此本文档中暂不做配置的修改。如果您有需要，直接勾选所需选项即可，服务将自动保存您的选择。

步骤3 验证配置结果。

当项目经理完成分解Story操作后，开发人员Chris将收到以下两类通知。

- 站内信：Chris登录首页后，页面右上角图标处将显示数字，单击该图标即可看到通知。
- 邮件：如果为项目成员创建用户时配置了邮箱，且项目成员在个人设置中开启了邮件通知，则将会收到服务发出的邮件。

  说明

  每个成员均可以设置是否接收邮件通知。开启邮件通知的方法为：

  1. 单击页面右上角的用户名，在下拉列表中选择“个人设置”。
  2. 在“消息设置”页面中找到“邮件通知”，勾选“开启”。可以单击“更改设置”，修改邮箱地址。

----结束

定制项目工作流程

在迭代Review会议中，团队将向产品负责人做产品演示，并出示测试报告，由产品负责人确认Story是否完成。而当前的Story状态中没有能够显示测试已完成的状态，因此测试人员建议增加一个状态“验收中”。

项目经理Maggie通过以下操作为Story添加状态。

步骤1 进入“凤凰商城”项目，单击导航“设置 > 工作配置”。

步骤2 在页面左侧导航中选择“公共状态设置”，页面将显示样例项目默认的工作项状态列表。

步骤3 单击“添加状态”，参照**表1-10**在弹框中编辑状态信息，单击“添加”保存。

表 1-10 状态信息配置

配置项	配置建议
状态	输入“验收中”。
状态属性	选择“进行态”。

步骤4 在导航中选择“Story设置 > 状态与流转”，页面将显示样例项目默认的Story状态列表。

步骤5 单击“添加已有状态”，在弹框中勾选“验收中”，单击“确定”保存。

图 1-7 添加 Story 状态

步骤6 通过拖拽将状态“验收中”的顺序至于“测试中”之后。

步骤7 验证配置结果。

1. 单击“工作 > 需求管理”，在页面中选择“工作项”页签。
2. 在列表中单击任意Story名称，查看Story详情。
3. 单击“状态”项的值，在下拉列表中可以看到选项“验收中”。

----结束

1.4.4 步骤三：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。

由于门店网络查询功能为高优先级Story，本章节将以此功能为例进行介绍如何进行源代码管理与开发。

本样例项目中采用分支来进行代码的开发。首先由开发人员Chris在代码仓库中创建分支，并进行代码开发；然后开发人员Chris在代码仓库中提交分支合并请求，项目经理Maggie评审通过后合并分支至主干。

使用分支管理代码

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开来，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，项目经理建议：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。

步骤1 将master分支设置为受保护分支（本文档中由项目经理Maggie操作）。

1. 进入“凤凰商城”项目，单击导航“代码 > 代码托管”，找到代码仓库“phoenix-sample”。
2. 单击仓库名称进入代码仓库，选择“设置”页签。在导航中单击“策略设置 > 保护分支”。
3. 单击“新建保护分支”，参照下表在弹框中完成配置，单击“确定”保存。

表 1-11 新建保护分支配置

配置项	配置建议
选择需要保护的分支	选择“master”。
能推送	根据需要配置，本文档中保持默认配置。
能合并	根据需要配置，本文档中保持默认配置。
成员	根据需要勾选“能推送”、“能合并”，并在下拉列表中选择成员。 本文档中保持默认配置。

说明

如果页面中已存在保护分支“master”，可单击，根据需要修改保护分支配置。

步骤2 创建功能分支（本文档中由开发者Chris操作）。

1. 进入“凤凰商城”项目，在代码托管页面中找到仓库“phoenix-sample”。
2. 单击仓库名称进入代码仓库，在“代码”页签中单击“分支”。
3. 单击“新建分支”，参照表1-12输入分支信息，单击“确定”保存。

表 1-12 新建分支

配置项	配置建议
基于	选择“master”。
分支名称	输入“Feature-Store”。
关联工作项	选择“作为用户可以查询所有门店网络”。

----结束

修改、提交代码

在迭代规划时将门店查询功能分解为前端展示与后台管理两个task，本节以Task“前端展示 - 添加门店网络菜单”介绍如何使用修改与提交代码。

步骤1 单击导航“工作 > 需求管理”，选择“迭代”页签。

在迭代4中找到Task“前端展示 - 添加门店网络菜单”，将Task的状态修改为“进行中”。

步骤2 单击导航“代码 > 代码托管”，找到仓库“phoenix-sample”。

步骤3 单击仓库名称进入代码仓库，选择“代码”页签。

步骤4 单击文件列表上方的“master”，在下拉列表中选择分支“Feature-Store”。

步骤5 在文件列表中找到“vote/templates/store-network.html”并打开。

步骤6 单击，根据Story添加门店地址，并在页面底部文本框中输入备注信息“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>
```

步骤7 以同样方法打开并编辑“/vote/templates/index.html”。

在179行添加菜单“门店网络”，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，在工作项列表中获取，实践中修改为实际Task的编号。

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

步骤8 单击导航“工作 > 需求管理 > 迭代”，在迭代4中找到Task“前端展示 - 添加门店网络菜单”。

- 单击Task名称，在详情页中可看到状态自动变为“已解决”。
- 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**上一步**中输入的提交信息相同。

图 1-8 代码提交记录

----结束

检视代码、合并分支

步骤1 开发人员发起合并请求。

开发人员Chris完成代码开发，确认无误后，即可发起合并请求，将功能分支合并到master中。

1. 进入代码仓库，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. 参照表1-13编辑合并请求详情。

表 1-13 合并请求配置

配置项	配置建议
标题	输入“添加门店网络列表”。
合并人	单击，在弹框中勾选“Maggie”，单击“确定”。
审核人	单击，在弹框中勾选“Maggie”，单击“确定”。

4. 单击“新建合并请求”完成合并请求的创建。

步骤2 项目经理评审并完成代码合入。

本文档中，合并请求的评审人与合并人均是项目经理Maggie。因此Maggie可评审合并请求内容，并在评审通过后完成分支合入。

1. 进入代码仓库后，选择“合并请求”页签，可找到由开发人员Chris创建的合并请求。
2. 单击该请求，查看合并请求详情。
3. 可在页面中留下评审意见。单击审核门禁中“通过”完成审核。
4. 单击“合入”，将分支合入“master”。

说明

如果发起分支合并请求时勾选了“合并后删除源分支”，分支“Feature-Store”将在分支合并完成后被删除。

----结束

1.4.5 步骤四：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。

通过本章节，您将了解开发人员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 1-14 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。
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，增加Python语言检查规则集，使检查更全面。

步骤1 编辑任务。

1. 进入“凤凰商城”项目，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。
2. 在列表中找到任务“phoenix-codecheck-worker”。
3. 单击任务名称进入详情页，选择“设置”页签。
4. 单击导航“规则集”，规则集中默认包含的语言是“JAVA”。
5. 增加Python语言检查规则集。
   a. 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。

      说明

      如果页面中已显示“PYTHON”，则忽略此步骤。

   b. 将PYTHON语言对应的开关打开。
   c. 在弹框中单击“确定”。

步骤2 执行任务。

1. 单击“开始检查”，启动任务。
2. 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，请根据页面弹出报错提示排查修改。

----结束

查看检查结果

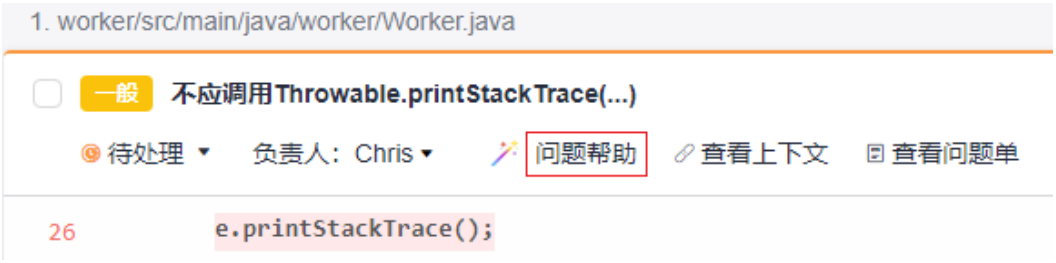
代码检查服务提供检查结果统计，并对检查出的问题提供修改建议，可以根据修改建议优化项目代码。

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 1-9 查看问题帮助



----结束

1.4.6 步骤五：构建应用

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。

通过本章节，您将了解开发人员Chris如何构建并归档镜像和软件包。

预置任务简介

样例项目中预置了以下5个构建任务。

表 1-15 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，此任务包含的步骤如下。

表 1-16 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	为了将镜像部署到ECS时，能够可以拉取到正确的镜像，使用shell命令进行完成以下操作。 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。 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。

构建步骤	说明
替换Kubernetes部署文件镜像版本	为了将镜像部署到CCE时，能够可以拉取到正确的镜像，使用shell命令进行完成以下操作。 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”、“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

本文档使用SWR来保存环境镜像，需要首先配置容器镜像服务（SWR）。

步骤1 在CodeArts中单击导航“控制台”。在控制台中单击左上角搜索并进入SWR服务。

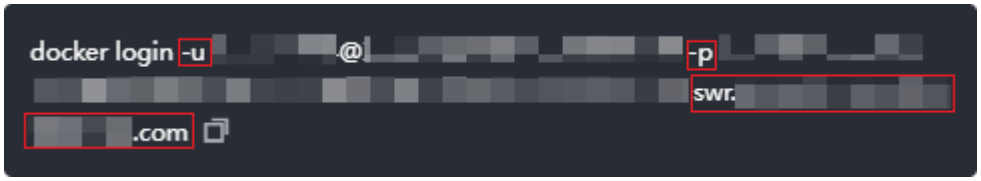
请检查页面左上方的“区域”列表，请确保与编译构建任务所在区相同的区域。如果区域不同，请选择相同区域。

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 1-10 登录指令



说明

此处生成的登录指令为临时登录指令，有效期为24小时。如果需要长期有效的登录指令，请参见**获取长期有效登录指令**。

步骤3 单击“创建组织”，在弹框中输入组织名称“phoenix”（此名称全局唯一，如果页面提示“组织已存在”，请自定义其它名称），单击“确定”保存。

这里的组织名称，即为后续配置并执行任务中的参数“dockerOrg”。

----结束

配置并执行任务

步骤1 配置任务。

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。
2. 在列表中找到任务“phoenix-sample-ci”。单击图标“...”，在下拉列表中单击“编辑”。
3. 选择“参数设置”页签，参照表1-17编辑参数值。

表 1-17 参数设置

参数名称	默认值
codeBranch	master。
dockerOrg	输入在SWR服务中创建的组织（本文中为“phoenix”）。
version	1.0.0
dockerServer	输入在SWR服务中获取的SWR服务器地址。

说明

请务必确保参数“dockerOrg”、“dockerServer”的输入值是正确的，否则将导致任务失败。

步骤2 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 

）。

如果构建失败，请根据失败步骤信息与日志中的报错信息排查。

步骤3 检查发布件。

1. 单击导航“制品仓库 > 软件发布库”，进入软件发布库。
2. 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“docker-stack”文件夹中，可找到与步骤2中记录的字符串同名的文件夹，在此文件夹中可以找到发布件“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。
3. 进入容器镜像服务，在导航中选择“组织管理”，单击与构建任务参数“dockerOrg”的值中同名的组织。

选择“镜像”页签，可以在列表中找到5个镜像（redis、postgres、worker、result、vote）。

4. 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2中记录的字符串相同。

----结束

设置提交代码触发自动编译

通过以下配置，可实现代码变更后自动触发构建任务的执行，从而实现项目的持续集成。

步骤1 在任务“phoenix-sample-ci”的详情页，单击“编辑”。

步骤2 选择“执行计划”页签。

步骤3 打开“提交代码触发执行”开关，保存任务。

由于在参数设置页面为参数codeBranch配置的默认值为“master”，因此本次设置的结果是当master有代码变更时自动触发构建。

步骤4 验证配置结果：修改项目代码并提交至master，即可查看构建任务是否自动执行。

----结束

设置定时执行任务

为了防止问题代码进入生产环境，确保应用总是处于可部署的状态，团队建议对应用进行持续不断的验证。

通过以下设置，可实现构建任务的定时执行。

步骤1 在任务“phoenix-sample-ci”的详情页，单击“编辑”。

步骤2 选择“执行计划”页签。

步骤3 打开“启用定时执行”开关，根据需要选择执行日与执行时间，保存任务。

本文档中勾选“全选”，执行时间为“12:00”（本文中使用默认时区，可以根据实际需要修改时区）。

步骤4 验证配置结果：根据配置时间查看构建任务是否自动执行，本节不再赘述。

----结束

1.4.7 步骤六：部署应用（CCE 篇）

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、更稳定的持续地交付软件，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署至ECS，请参照**步骤六：部署应用（ECS篇）**操作。

预置应用简介

样例项目中预置了以下3个部署应用。

表 1-18 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。

购买并配置云容器引擎

本节中使用的是云容器引擎CCE。

通过控制台可**购买CCE集群**。

其中集群及节点的必要配置建议参照**表1-19**与**表1-20**，表中未涉及的可根据实际情况选择。

表 1-19 CCE 集群购买配置

配置分类	配置项	配置建议
基础配置	计费模式	选择“按需计费”。
	集群版本	根据需要选择，建议选择最新版本。
网络配置	容器网络模型	选择“容器隧道网络”。
	虚拟私有云	选择已有的虚拟私有云，如果列表中没有合适的选项，单击“新建虚拟私有云”完成创建。
	子网	选择已有的子网，如果列表中合适的选项，单击“新建子网”完成创建。
	容器网段	勾选“自动设置网段”。

表 1-20 节点配置

配置分类	配置项	配置建议
计算配置	计费模式	选择“按需计费”。
	节点类型	选择“弹性云服务器-虚拟机”。
	节点规格	选择2核8G及以上规格即可。
	操作系统	选择公共镜像中的Euler镜像。
	节点名称	输入自定义名称 。
	登录方式	选择“密码”。
	密码	输入自定义密码 。
网络配置	节点IP	选择“自动分配”。
	弹性公网IP	选择“自动创建”。

（可选）调整 yaml 文件配置

如果CCE集群版本高于v1.15（不包括v1.15），需调整代码仓库中的yaml文件，使其适配CCE集群版本。

步骤1 进入“凤凰商城”项目，单击导航“代码 > 代码托管”，选择代码仓库“phoenix-sample”。

步骤2 更新文件“kompose/db-deployment.yaml”。

- 将第1行中“extensions/v1beta1”修改为“apps/v1”。
- 找到“spec”代码段，添加以下代码行。

```
selector:
  matchLabels:
    io.kompose.service: db
```

- 找到“imagePullSecrets”代码段，将“regcred”修改为“default-secret”。

图 1-11 更新文件

```
1   apiVersion: apps/v1
2   kind: Deployment
3   metadata:
4     annotations:
5       kompose.cmd: kompose convert --file docker-stack-k8s.yml
6       kompose.version: 1.11.0 (39ad614)
7     creationTimestamp: null
8     labels:
9       io.kompose.service: db
10    name: db
11  spec:
12    replicas: 1
13    selector:
14      matchLabels:
15        io.kompose.service: db
16    strategy:
17      type: Recreate
18    template:
19      metadata:
20        creationTimestamp: null
21        labels:
22          io.kompose.service: db
23      spec:
24        containers:
25        - image: docker-server/docker-org/postgres:9.4
26          name: db
27          command: [ "/bin/bash", "-c", "--" ]
28          args: [ "while true; do sleep 30; done;" ]
29        restartPolicy: Always
30        imagePullSecrets:
31        - name: default-secret
32        restartPolicy: Always
33  status: {}
```

步骤3 更新文件“kompose/redis-deployment.yaml”。

- 将第1行中“extensions/v1beta1”修改为“apps/v1”。
- 找到文件中第一次出现的“spec”，在代码段中添加以下代码行。

```
selector:
  matchLabels:
    io.kompose.service: redis
```

- 找到“imagePullSecrets”代码段，将“regcred”修改为“default-secret”。

步骤4 更新文件“kompose/result-deployment.yaml”。

- 将第1行中“extensions/v1beta1”修改为“apps/v1”。
- 找到文件中第一次出现的“spec”代码段，添加以下代码行。

```
selector:
  matchLabels:
    io.kompose.service: result
```

- 找到“imagePullSecrets”代码段，将“regcred”修改为“default-secret”。

步骤5 更新文件“kompose/vote-deployment.yaml”。

- 将第1行中“extensions/v1beta1”修改为“apps/v1”。
- 找到文件中第一次出现的“spec”代码段，添加以下代码行。

```
selector:
  matchLabels:
    io.kompose.service: vote
```

- 找到“imagePullSecrets”代码段，将“regcred”修改为“default-secret”。

步骤6 更新文件“kompose/worker-deployment.yaml”。

- 将第1行中“extensions/v1beta1”修改为“apps/v1”。
- 找到文件中第一次的“spec”代码段，添加以下代码行。

```
selector:
  matchLabels:
    io.kompose.service: worker
```

- 找到“imagePullSecrets”代码段，将“regcred”修改为“default-secret”。

步骤7 单击导航“持续交付 > 编译构建”，执行任务“phoenix-sample-ci”。

----结束

配置并执行应用

将在**步骤五：构建应用**中生成的“.yaml”文件逐一部署在CCE集群中。

步骤1 配置应用。

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。
2. 找到应用“phoenix-cd-cce”。单击图标“...”，在下拉列表中单击“编辑”，进入编辑页面。
3. “部署步骤”页签，在每个步骤中完成以下配置。

表 1-21 配置部署步骤

配置项	配置建议
集群名称	选择在购买云容器引擎时设置的集群名称。
命名空间名称	本文中选择“default”。

4. 选择“参数设置”页签，配置以下参数。

表 1-22 参数设置

参数名称	参数值
ci_task_name	输入“phoenix-sample-ci”。
version	输入任务“phoenix-sample-ci”的参数“version”值。

5. 单击“保存”，完成应用的编辑。

步骤2 单击导航“控制台”，通过服务列表进入云容器引擎服务。

找到目标集群，单击集群名称进入总览页。

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，单击“批量删除”，并勾选全部资源释放选项，单击“是”，将列表记录清空。

步骤3 返回应用列表页面，单击应用“phoenix-cd-cce”所在行的▷，在弹框中单击“确定”，启动部署。

当页面中显示✔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排查。

步骤4 验证部署结果。

1. 进入云容器引擎服务。
2. 找到目标集群，单击集群名称进入总览页，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”。
3. 单击“vote”进入详情页，在“访问方式”页签中单击“更多 > 更新”。
   参照表1-23配置参数，单击“确定”。

表 1-23 更新服务

参数名称	参数值
服务亲和	选择“集群级别”。
负载均衡器	– 选择“共享型 > 自动创建”。 – 实例名称：输入“phoenix”。 – 弹性公网IP：选择“自动创建”。 **说明** 如果账号下已有负载均衡器，可选择“共享型 > 使用已有”，并选择已存在的负载均衡器名称。
端口配置	– 容器端口：80 – 服务端口：5000

4. 更新成功，返回列表中，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 1-12 复制访问地址

5. 打开新的浏览器页面，在地址栏中输入“http://ip:5000”（其中，ip为上一步记录的公网地址），页面显示成功。
6. 返回“无状态负载”页面，参照**步骤4.3**更新“result”（其中，负载均衡器选择**步骤4.3**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”，页面显示成功。

----结束

1.4.8 步骤六：部署应用（ECS 篇）

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署至CCE，请参照**步骤六：部署应用（CCE篇）**操作。

购买并配置 ECS

本节使用的是ECS，您也可以使用自己的Linux主机（Ubuntu 16.04操作系统）。

步骤1 **购买弹性云服务器**。

购买时的必要配置参照下表，表中未列出的配置可根据实际情况选择。

表 1-24 弹性云服务器购买配置

配置分类	配置项	配置建议
基础配置	计费模式	选择“按需计费”。
	CPU架构	选择“x86计算”。
	规格	选择2核8G或以上规格。
	镜像	选择“公共镜像 > Ubuntu > Ubuntu 16.04 Server 64bit”。
网络配置	弹性公网IP	选择“现在购买”。
	公网带宽	选择“按带宽计费”。

配置分类	配置项	配置建议
高级配置	登录凭证	选择“密码”。
	密码	输入自定义密码。

步骤2 配置安全组规则。

样例项目的验证需要用到端口5000与5001，因此添加一条允许访问5000以及5001端口的入方向规则。

操作步骤如下：

1. 登录ECS页面，在列表中找到步骤**步骤1**中购买的ECS，单击服务器名称。
2. 选择“安全组”页签，参考**配置安全组规则**添加一条协议为TCP、端口为5000-5001的入方向规则。

----结束

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“凤凰商城”项目，单击导航栏“设置 > 通用设置 > 基础资源管理”。

步骤2 单击“新建主机集群”，输入集群名称“hosts”、选择操作系统（Linux），单击“保存”。

步骤3 单击“新增目标主机”，在弹框中配置以下信息，勾选同意声明后，单击“添加”。

表 1-25 添加主机

配置项	配置建议
主机名	输入自定义主机名称。为了方便辨认，可输入在购买ECS时设置的名称。
IP	输入在购买ECS时生成的弹性IP。
用户名	输入“root”。
密码	输入在购买ECS时设置的密码。
ssh端口	输入“22”。

步骤4 页面显示一条主机记录，当“连通性验证”列的值显示为“成功”，表示主机添加完成。

如果主机添加失败，请根据失败详情排查主机配置。

----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”，在列表中找到应用“phoenix-sample-predeploy”。

步骤2 单击···，在下列表中选择“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”，输入环境名称“phoenix-hostgroup”、选择资源类型“主机”、操作系统“Linux”，单击“保存”。
2. 单击“导入主机”，在弹框中的下拉列表中选择已创建的主机集群，并在列表中勾选主机，单击“导入”。

说明

如果无新建环境权限，请联系管理员通过应用的“权限管理”页面添加权限。

步骤4 在“部署步骤”页签，编辑应用的步骤。

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

步骤5 单击“保存并执行”，启动部署任务。

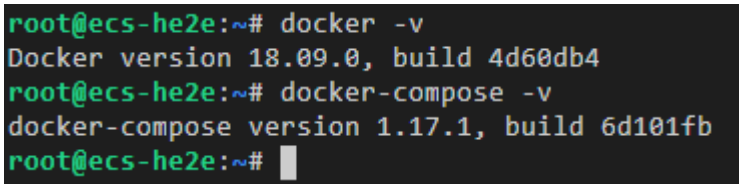
当出现页面提示“部署成功”时，表示任务执行成功。

步骤6 登录弹性云服务器，执行以下命令，检测依赖工具是否安装成功。

- 查看Docker镜像版本。
  ```
  docker -v
  ```
- 查看Docker-Compose版本。
  ```
  docker-compose -v
  ```

当出现类似图1-13所示提示时，表示安装成功。

图 1-13 查看 Docker 及 Docker-Compose 版本



----结束

配置并执行应用

部署时需要将ECS配置在应用的环境列表中，并将构建任务“phoenix-sample-ci”设置为部署来源。

步骤1 进入“凤凰商城”项目，单击导航“持续交付 > 部署”，在列表中找到应用“phoenix-sample-standalone”。

步骤2 单击···，在下列表中选择“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”，输入环境名称“phoenix-hostgroup”、选择资源类型“主机”、操作系统“Linux”，单击“保存”。

2. 单击“导入主机”，在弹框中的下拉列表中选择已创建的主机集群，并在列表中勾选主机，单击“导入”。
3. 页面提示导入成功，关闭此窗口。

步骤4 在“部署步骤”页签，编辑应用的步骤。

1. 在步骤“选择部署来源”中，参照**表1-26**设置部署来源。

表 1-26 部署来源配置

配置项	配置建议
选择源类型	选择“构建任务”。
请选择构建任务	选择“phoenix-sample-ci”。
环境	选择“phoenix-hostgroup”。 如果页面显示弹框“是否将后续步骤的环境也修改为phoenix-hostgroup？”，单击“确定”。

2. 最后两个步骤“解压文件”与“执行shell命令”保持默认配置即可。

步骤5 选择“参数设置”页签，根据SWR登录指令填写参数。

登录指令通过控制台获取，操作方式请参考**配置SWR服务**。

步骤6 单击“保存并部署”，启动部署。

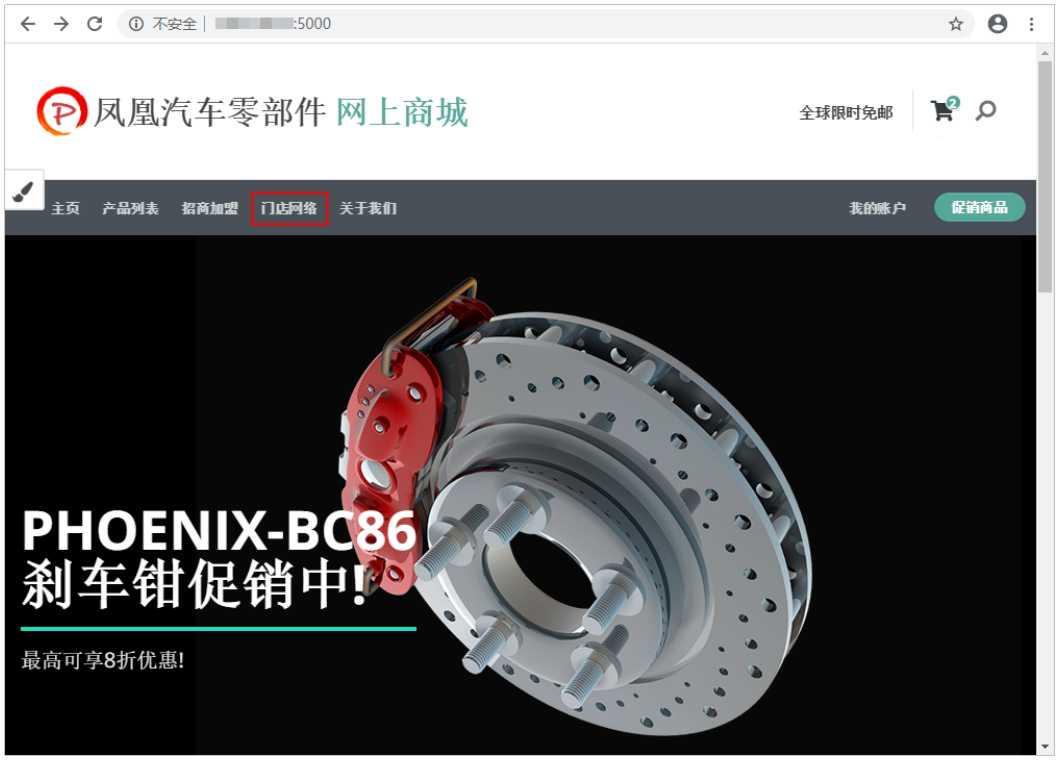
当页面显示“部署成功”时，表示部署成功。如果部署失败，请根据失败步骤信息与日志中的报错信息排查。

步骤7 验证部署结果。

打开浏览器，输入“http://ip:5000”，其中ip为ECS的IP地址。

页面显示成功，在导航栏中可看到菜单项“门店网络”。

图 1-14 用户端 UI



输入“http://ip:5001”，其中ip为ECS的IP地址，页面显示成功。

图 1-15 管理端 UI

----结束

1.4.9 步骤七：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。

通过本章节，您将了解测试人员Billy如何管理项目的测试周期，包括创建与执行测试用例、跟踪测试进度等。

创建迭代测试计划

在确定迭代4中计划实现的需求（Story）后（即完成**步骤一：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 创建测试计划。

1. 进入“凤凰商城”项目，单击导航“测试 > 测试计划”。
2. 单击“新建计划”，配置测试计划信息。
   a. 基本信息：配置以下信息，单击“下一步”。

表 1-27 测试计划基本信息

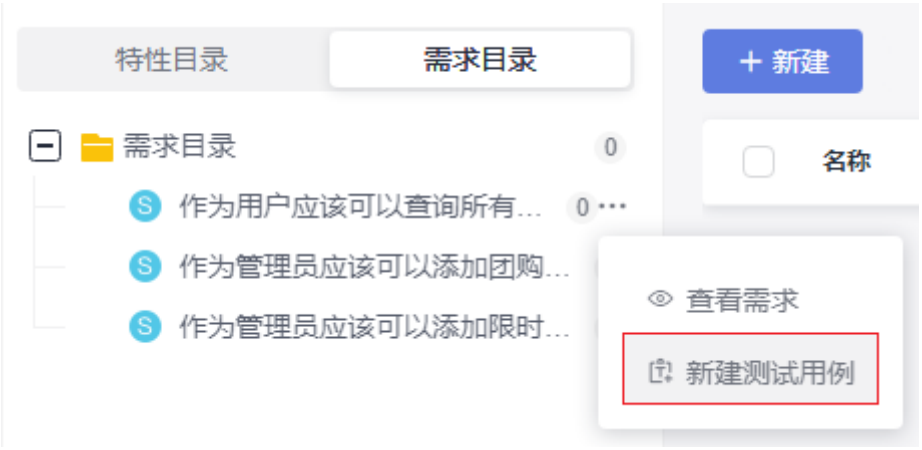
子配置项	配置建议
名称	输入“迭代4”。
处理者	选择“Billy”。
计划周期	建议与在需求管理中创建的“迭代4”的周期一致。
关联迭代	选择“迭代4”。

   b. 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”。
3. 返回测试计划页面，在列表中可找到新创建的测试计划“迭代4”，状态为“新建”。

步骤2 设计测试用例。

1. 在测试计划“迭代4”中，单击“设计”。
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 1-16 新建测试用例



3. 输入名称“门店网络查询”，参照**表1-28**编辑测试步骤与预期结果，单击“保存”。

表 1-28 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。

4. 按照同样的方式，为其它两个Story创建测试用例。
5. 单击导航“测试 > 测试计划”，返回测试计划列表。
   在列表中可看到测试计划“迭代4”的状态为“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**步骤六：部署应用（CCE篇）**或**步骤六：部署应用（ECS篇）**），可将Story的状态设置为“已解决”，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。

本节以门店网络查询功能为例，介绍如何执行测试用例、以及测试用例执行失败如何反馈Bug信息。

步骤1 在“凤凰商城”项目中，单击导航“工作 > 迭代”。

在迭代4中找到Story“作为用户应该可以查询所有门店网络”，将Story的状态修改为“测试中”。

步骤2 进入“测试 > 测试用例”页面，在页面上方选择“迭代4”。

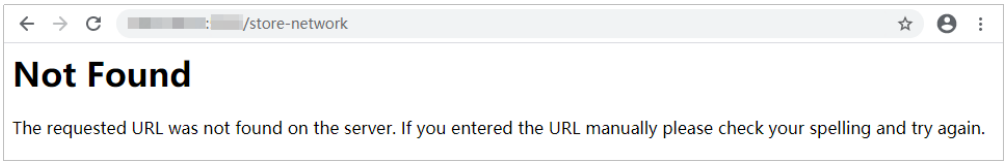
步骤3 在列表中单击用例“门店网络查询”，将状态修改为“测试中”，单击“保存”。

步骤4 选择“手工测试”页签，单击用例“门店网络查询”所在行的▷，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。

- 执行成功，跳转至**步骤6**继续操作。
- 执行失败，例如：执行第二步时页面跳转失败，页面显示404，跳转至**步骤7**继续操作。

图 1-17 页面显示失败



步骤6 返回测试用例执行窗口，记录执行结果。

1. 在表格中，设置所有步骤的实际结果为“成功”。
2. 在表格上方，设置测试用例的结果为“成功”。
3. 勾选“同时将用例状态设为已完成”。
4. 单击页面右上角“保存”。

图 1-18 测试用例执行成功

此时测试用例的状态将自动变更为“完成”。跳转至**步骤13**继续操作。

步骤7 返回测试用例执行窗口，记录执行结果。

1. 在表格中，设置步骤1的实际结果为“成功”。
2. 在表格中，设置步骤2的实际结果为“失败”，并输入实际显示内容“跳转失败，页面显示404”。
3. 在表格上方，设置测试用例的结果为“失败”。
4. 单击页面右上角“保存”。

图 1-19 测试用例执行失败

步骤8 单击页面右上角⋮，选择“新建缺陷”，页面将跳转至新建缺陷（新建工作项）页面。

步骤9 在页面左下方的文本框的最后，可以看到自动填充缺陷的重现步骤。

参照**表1-29**编辑缺陷详情，单击“保存”，页面将跳转到工作项列表页面。

表 1-29 缺陷详情配置

配置项	配置建议
标题	输入“门店网络页面显示404”。
处理人	选择“Chris”。
迭代	选择“迭代4”。

步骤10 在列表中找到Bug“门店网络页面显示404”，单击名称，选择“关联”页签，在“关联用例”下可看到测试用例“门店网络查询”。

步骤11 单击关联用例的编号，可跳转到用例详情页。

选择“缺陷列表”页签，可看到一条缺陷记录，即在**步骤9**中创建的缺陷。

步骤12 当开发人员修复缺陷后并验证成功后，参考**步骤6**设置用例结果，并将对应的缺陷状态设置为“已关闭”。

步骤13 执行其它测试用例。

步骤14 当所有用例的状态均为“完成”时，单击导航“测试 > 测试计划”，返回测试计划列表，在列表中可看到测试计划“迭代4”的状态为“完成”。

----结束

跟踪测试计划进展

- 查看质量报告。

  通过质量报告，团队可以直观的查看测试计划的当前进展，包括需求覆盖率、缺陷、用例通过率、用例完成率等。

  在“测试 > 测试计划”页面，在测试计划“迭代4”中，单击卡片中的“报告”，即可查看此迭代质量报告。

- 自定义报表。

  除了内置的质量报告，团队可以根据需要自定义统计报表。

  下面以统计测试用例执行结果为例，介绍如何自定义统计报表。

  a. 在“测试质量看板”页面，单击页面下方空白处“点击添加报表”，在弹框中选择“自定义报表”。

  b. 参照表1-30编辑报表信息，单击“保存”。

  表 1-30 报表配置

配置项	配置建议
报表标题	输入“测试用例执行结果统计”。
工件类型	选择“测试用例”。
分析维度	选择“结果”。

  c. 页面跳转回“测试质量看板”，在页面最下方显示新建的报表。

1.4.10 步骤八：配置流水线，实现持续交付

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，各个环节（构建、发布、部署）越来越标准化。但是每个环节都相对独立，是半成品，不能交付业务价值。将每一个环节有效的串联起来形成一套完整的持续交付流水线，才能够真正提高软件的发布效率与质量，持续不断的创造业务价值。

通过本章节，您将了解开发人员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 1-31 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。

预置流水线任务	任务说明
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

一条流水线通常由多个阶段构成，每个阶段中可以添加多个子任务。

步骤1 配置流水线。

1. 进入“凤凰商城”项目，单击导航“持续交付 > 流水线”。
2. 找到流水线“phoenix-workflow”。单击图标，在下拉列表中单击“编辑”，进入编辑页面。
3. 添加代码检查阶段。
   a. 单击“流水线源”与“构建”之间的，添加阶段。
   b. 单击“阶段_1”后的，在“编辑阶段”窗口中输入阶段名称“代码检查”，单击“确定”。

图 1-20 编辑阶段名称

   c. 单击“新建任务”。
   在“新建任务”窗口中，单击“Check代码检查”插件后的“添加”。
   d. 选择调用任务“phoenix-codecheck-worker”，单击“确定”。

说明

代码检查任务有三种检查模式，本文保持默认值“Full”，可根据需要修改。

- Full：全量检查，扫描代码仓里的所有文件。
- Incremental（last commit）： 增量检查，基于最近一次commit文件进行扫描。
- 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。

4. 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-cce”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。

5. 配置完成，单击“保存”。

步骤2 如果在步骤1配置了“phoenix-cd-cce”任务，进入云容器引擎服务。找到目标集群，单击集群名称进入总览页。

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，单击“批量删除”，并在弹框中勾选所有选项，单击“是”，将列表记录清空。

步骤3 返回流水线列表页面，单击“phoenix-workflow”所在行的▷，在滑出的窗口单击“运行”，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，请于执行失败的任务处检查失败原因，可打开步骤详情查看任务日志，根据日志进行排查。

----结束

配置准出条件

为了控制代码的质量，代码必须经过扫描，并且错误数量控制在合理范围内，才允许发布。通过添加质量门禁可以有效的自动化控制流程。

步骤1 在流水线任务“phoenix-workflow”详情页，单击“编辑”。

步骤2 在阶段“代码检查”中，单击“准出条件”。

步骤3 在“准出条件”窗口中，单击“标准策略准出条件”插件后的“添加”。

步骤4 选择“系统策略”，单击“确定”。

步骤5 单击“保存并运行”，启动流水线任务。

如果代码检查问题数未达到准出条件，流水线任务将执行失败。

----结束

配置代码变更自动触发流水线

通过以下配置，可实现代码变更自动触发流水线执行，从而实现项目的持续交付。

步骤1 在流水线任务“phoenix-workflow”详情页，单击“编辑”。

步骤2 选择“执行计划”页签，在“事件触发”目录下勾选“代码提交时触发”开关，在分支过滤下拉列表中勾选分支“master”，单击“保存”。

步骤3 验证配置结果：修改代码并推送至master，即可查看流水线是否自动执行。

----结束

1.4.11 释放资源

为了避免不必要的费用产生，完成本样例项目体验后，产品负责人Sarah可根据实际使用的需要，释放部分不再使用的资源。

可以释放的资源如下。

须知

资源释放后无法恢复，请谨慎操作。

步骤1 删除项目。

进入项目“设置 > 通用设置 > 基本信息”页面，单击“删除项目”，根据页面提示完成删除操作。

步骤2 删除组织与镜像。

1. 登录SWR控制台。
2. 在“我的镜像”页面中，勾选本文中创建的镜像，单击“删除”，根据页面提示完成删除操作。
3. 在“组织管理”页面中，单击待删除组织的名称，进入详情页。单击“删除”，根据页面提示完成删除操作。

步骤3 删除集群。

登录CCE控制台。在列表中找到待删除的集群，单击“删除集群”，根据页面提示完成删除操作。

步骤4 删除ECS。

登录ECS控制台，在列表中找到待删除的ECS，单击“更多”，在下拉列表中选择“删除”，根据页面提示完成删除操作。

----结束

1.5 附录

1.5.1 构建失败，报错“too many requests”

问题现象

构建失败，报错信息如下：

```
toomanyrequests: You have reached your pull rate limit. You may increase the limit by authenticating and upgrading: https://www.docker.com/increase-rate-limit
```


原因分析

构建任务中使用的基础镜像源为DockerHub。由于DockerHub的限制，短时间内拉取次数较多时将受限无法拉取，因此可能会造成构建失败。

处理方法

可首先制作基础依赖镜像，推送到容器镜像服务中，以供正式构建时获取。

步骤1 新建构建任务。

1. 进入“凤凰商城”项目，单击导航“持续交付 > 编译构建”。
2. 单击“新建任务”，任务名称设置为“phoenix-prebuild”，源码源及仓库信息与预置任务“phoenix-sample-ci”保持一致，单击“下一步”。
3. 选择“空白构建模板”，单击“下一步”。

步骤2 配置参数。

1. 选择“参数设置”页签，单击“新增参数”。
2. 输入名称“dockerOrg”，在“默认值”中输入在SWR服务中创建的组织名称。

步骤3 选择“构建步骤”页签，配置构建步骤。

1. 添加步骤“执行shell命令”，删除命令框中的命令行，输入以下命令。
```
echo from postgres:9.4 > Dockerfile-postgres
echo from redis:alpine > Dockerfile-redis
echo from node:8.16-slim > Dockerfile-node
echo from python:2.7-alpine > Dockerfile-python
echo from java:openjdk-8-jdk-alpine > Dockerfile-java
```
2. 添加步骤“制作镜像并推送到SWR仓库”，修改步骤显示名称为“制作Postgres镜像并推送到SWR仓库”，参照**表1-32**完成配置。

表 1-32 Postgres 镜像配置

配置项	配置建议
组织	${dockerOrg}
镜像名字	postgres
镜像标签	9.4
Dockerfile路径	./Dockerfile-postgres

3. 添加步骤“制作镜像并推送到SWR仓库”，修改步骤显示名称为“制作Redis镜像并推送到SWR仓库”，参照**表1-33**完成配置。

表 1-33 Redis 镜像配置

配置项	配置建议
组织	${dockerOrg}
镜像名字	redis
镜像标签	alpine

配置项	配置建议
Dockerfile路径	./Dockerfile-redis

4. 添加步骤“制作镜像并推送到SWR仓库”，修改步骤显示名称为“制作Node镜像并推送到SWR仓库”，参照**表1-34**完成配置。

表 1-34 Node 镜像配置

配置项	配置建议
组织	${dockerOrg}
镜像名字	node
镜像标签	8.16-slim
Dockerfile路径	./Dockerfile-node

5. 添加步骤“制作镜像并推送到SWR仓库”，修改步骤显示名称为“制作Python镜像并推送到SWR仓库”，参照**表1-35**完成配置。

表 1-35 Python 镜像配置

配置项	配置建议
组织	${dockerOrg}
镜像名字	python
镜像标签	2.7-alpine
Dockerfile路径	./Dockerfile-python

6. 添加步骤“制作镜像并推送到SWR仓库”，修改步骤显示名称为“制作Java镜像并推送到SWR仓库”，参照**表1-36**完成配置。

表 1-36 Java 镜像配置

配置项	配置建议
组织	${dockerOrg}
镜像名字	java
镜像标签	openjdk-8-jdk-alpine
Dockerfile路径	./Dockerfile-java

步骤4 生成基础镜像。

1. 单击“新建并执行”，启动编译构建任务。

   当页面中显示  时，表示任务执行成功。

如果构建时仍出现同样的报错信息，请参考**相关操作**设置镜像加速器。

2. 进入SWR控制台，在页面左侧导航选择“我的镜像”。
   列表中可看到新增的5个镜像（java、python、node、redis、postgres）。

步骤5 替换基础镜像源地址。

1. 单击导航“代码 > 代码托管”，选择代码仓库“phoenix-sample”。
2. 参照**表1-37**编辑代码文件中的基础镜像源地址。

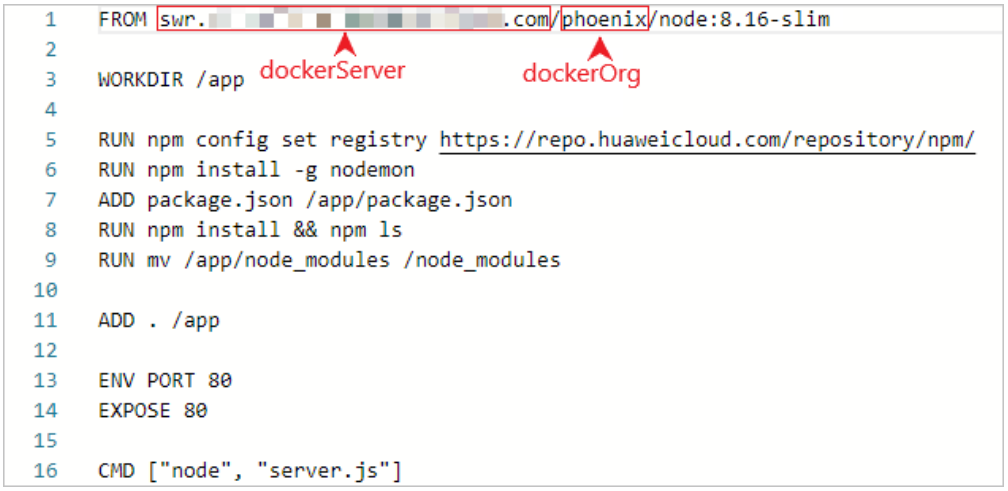
说明

表中的“${dockerServer}”、“${dockerOrg}”仅为参数示例。实际替换镜像源地址时，请参照**图1-21**所示，将“${dockerServer}”在**配置SWR服务**中记录的SWR服务器地址，“${dockerOrg}”替换为在**配置SWR服务**中创建的组织。

表 1-37 替换基础镜像源地址

路径与位置	修改前内容	修改后内容
文件“result/Dockerfile”第1行	node:5.11.0-slim	${dockerServer}/${dockerOrg}/node:8.16-slim
文件“/vote/Dockerfile”第2行	python:2.7-alpine	${dockerServer}/${dockerOrg}/python:2.7-alpine
文件“/worker/Dockerfile.j2”第1行	java:openjdk-8-jdk-alpine	${dockerServer}/${dockerOrg}/java:openjdk-8-jdk-alpine

图 1-21 替换基础镜像源地址



----结束

说明

完成以上操作步骤后，Postgres和Redis镜像已制作完成，因此在编辑预置任务“phoenix-sample-ci”时，请禁用以下步骤：

- 生成Postgres and Redis Dockerfile
- 制作Postgres镜像并推送到SWR仓库
- 制作Redis镜像并推送到SWR仓库

相关操作

容器镜像服务（SWR）提供了镜像加速器功能，

步骤1 登录SWR控制台。

步骤2 单击页面左侧导“镜像资源 > 镜像中心”，进入“镜像中心”页面。

步骤3 单击“镜像加速器”，在弹框中找到加速器地址，复制“https://”之后的内容。

图 1-22 镜像加速器

步骤4 进入代码托管服务，修改代码中引用的镜像地址。

- 将文件“result/Dockerfile”中第一行代码修改为以下内容，如所示。
  ```
  FROM 加速器地址/library/node:8.16-slim
  ```
- 将文件“/vote/Dockerfile”中第二行代码修改为以下内容。
  ```
  FROM 加速器地址/library/python:2.7-alpine
  ```
- 将文件“/worker/Dockerfile.j2”中第一行代码修改为以下内容。
  ```
  FROM 加速器地址/library/java:openjdk-8-jdk-alpine
  ```


步骤5 进入编译构建服务，编辑任务“phoenix-prebuild”。

将步骤“执行Shell命令”中的命令行修改为以下内容。

```
echo from 加速器地址/library/postgres:9.4 > Dockerfile-postgres
echo from 加速器地址/library/redis:alpine > Dockerfile-redis
echo from 加速器地址/library/node:8.16-slim > Dockerfile-node
echo from 加速器地址/library/python:2.7-alpine > Dockerfile-python
echo from 加速器地址/library/java:openjdk-8-jdk-alpine > Dockerfile-java
```

步骤6 保存并执行构建任务。

----结束

1.5.2 ECS 部署成功，但访问网页失败

问题现象

应用“phoenix-sample-standalone”部署成功，但访问网页（“http://*ip地址*:5000”与“http://*ip地址*:5001”）失败。

原因分析

- 主机未添加入方向规则“允许访问5000以及5001端口”。
- 本文建议使用操作系统为Ubuntu 16.04的主机，其它操作系统可能会无法访问部署后的网站。

处理方法

- 检查主机的安全组配置中，是否存在协议为TCP、端口为5000-5001的入方向规则。如果不存在，请添加此规则，配置方式请参考**配置安全组规则**。
- 可根据需要重新购买一台操作系统为Ubuntu 16.04的主机（ECS配置请参考**购买并配置ECS**，购买方式请参考**购买弹性云服务器**），或将当前主机操作系统切换为Ubuntu 16.04（切换操作系统方式请参考**切换操作系统**）。

1.5.3 ECS 部署失败，报错“docker login failed”或“Get https://XXX denied”

问题现象

应用“phoenix-sample-standalone”部署失败，报错信息为“docker login failed”或“Get https://XXX denied”。

图 1-23 报错信息

原因分析

由于应用的参数配置错误、连接超时等多种可能原因，导致Docker登录认证失败。

处理方法

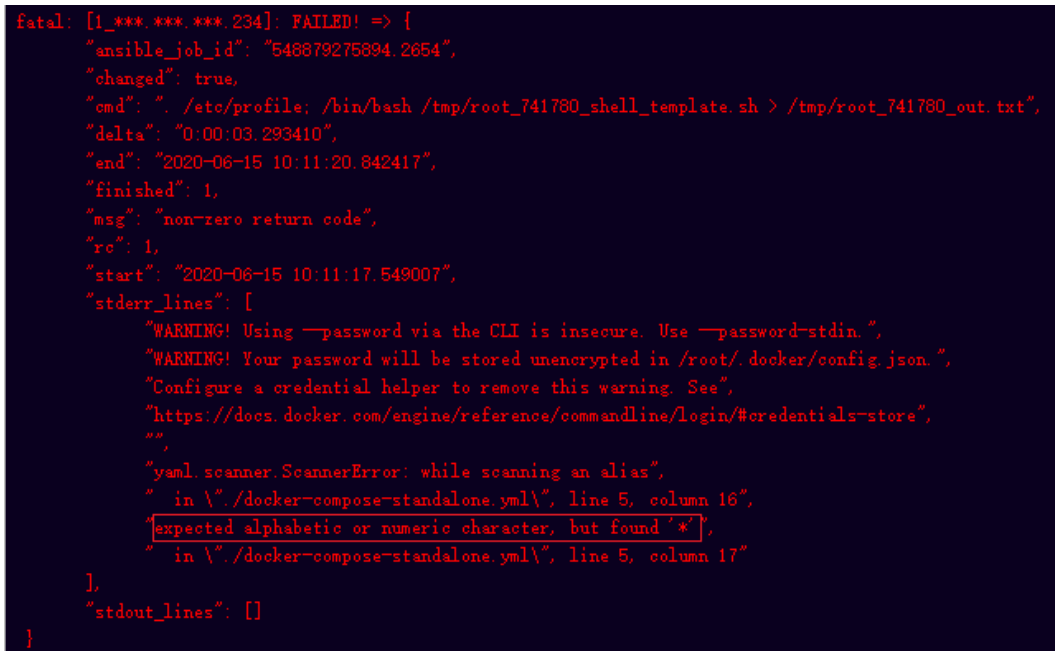
参照**配置SWR服务**重新获取SWR参数，配置到应用“phoenix-sample-standalone”的参数中，重新部署应用。

1.5.4 ECS 部署失败，报错“expected alphabetic or numeric character, but found '*'”

问题现象

应用“phoenix-sample-standalone”部署失败，报错信息为“expected alphabetic or numeric character, but found '*'”。

图 1-24 报错信息



原因分析

由于构建任务参数设置不正确，导致部署应用时获取不到正确的部署来源数据。

处理方法

参照**配置SWR服务**重新获取SWR参数，配置到构建任务中，并确保应用“phoenix-sample-standalone”的参数设置准确，重新执行构建任务与部署应用。

1.5.5 CCE 部署失败，报错“Invalid value: map[string]sting{\"io.kompose.serivce\":\"db\"”

问题现象

应用phoenix-cd-cce部署失败，报错信息如下：

图 1-25 报错信息示例

原因分析

在修改yaml文件时，文件中代码的缩进格式不正确，导致部署失败。

处理方法

检查代码仓库中以“-deployment.yaml”结尾的文件，其中“selector”代码段的缩进格式与**图1-11**保持一致。

完成检查后重新执行构建任务与部署应用。