凤凰商城项目实践

实验手册



华为技术有限公司

目录

[前 言 3](#_Toc58851921)

[1 持续规划与设计 4](#_Toc58851922)

[1.1 敏捷项目规划 4](#_Toc58851923)

[1.1.1 实验介绍 4](#_Toc58851924)

[1.1.2 实验目的 4](#_Toc58851925)

[1.1.3 创建华为云DevCloud项目 4](#_Toc58851926)

[1.1.4 使用Scrum项目模板进行项目规划，并管理Epic和Feature 5](#_Toc58851927)

[1.1.5 使用Scrum项目模板管理Backlog并进行迭代开发 6](#_Toc58851928)

[1.1.6 使用效率工具监测和跟踪项目状态 9](#_Toc58851929)

[1.2 项目管理配置 11](#_Toc58851930)

[1.2.1 实验介绍 11](#_Toc58851931)

[1.2.2 实验目的 11](#_Toc58851932)

[1.2.3 维护项目基本信息 11](#_Toc58851933)

[1.2.4 定制项目工作流程 12](#_Toc58851934)

[2 持续开发与集成 15](#_Toc58851935)

[2.1 代码托管 15](#_Toc58851936)

[2.1.1 实验介绍 15](#_Toc58851937)

[2.1.2 实验目的 15](#_Toc58851938)

[2.1.3 使用CloudIDE修改和提交代码 15](#_Toc58851939)

[2.1.4 使用Git分支+合并请求方式提交代码并进行代码检视 18](#_Toc58851940)

[2.2 代码检查确保编码规范 26](#_Toc58851941)

[2.2.1 实验介绍 26](#_Toc58851942)

[2.2.2 实验目的 26](#_Toc58851943)

[2.2.3 为示例项目配置代码检查任务并执行 26](#_Toc58851944)

[2.2.4 分析代码检查结果并修复问题 28](#_Toc58851945)

[2.2.5 配置和管理代码检查规则集 29](#_Toc58851946)

[2.3 使用自动化编译提高代码质量验证速度 30](#_Toc58851947)

[2.3.1 实验介绍 30](#_Toc58851948)

[2.3.2 实验目的 30](#_Toc58851949)

[2.3.3 编译构建并完成应用的镜像打包推送 30](#_Toc58851950)

[2.3.4 配置持续集成并通过提交代码触发自动化编译 39](#_Toc58851951)

[2.3.5 使用开源镜像站服务改进自动化编译 41](#_Toc58851952)

[3 持续测试与反馈 44](#_Toc58851953)

[3.1 实验介绍 44](#_Toc58851954)

[3.2 实验目的 44](#_Toc58851955)

[3.3 使用追溯视图进行需求驱动的测试 44](#_Toc58851956)

[3.4 使用测试套件进行迭代测试规划 46](#_Toc58851957)

[3.5 用例自定义 47](#_Toc58851958)

[3.6 使用测试用例和测试套件视图进行用例驱动测试 48](#_Toc58851959)

[3.7 使用测试总览跟踪测试进度和质量 52](#_Toc58851960)

[4 持续部署与发布 54](#_Toc58851961)

[4.1 实验介绍 54](#_Toc58851962)

[4.2 实验目的 54](#_Toc58851963)

[4.3 购买并配置弹性云服务器 54](#_Toc58851964)

[4.4 使用DevCloud部署服务配置主机环境 58](#_Toc58851965)

[4.5 使用自动化部署实现一键部署 61](#_Toc58851966)

[4.6 持续交付流水线 64](#_Toc58851967)

[4.6.1 配置流水线串接代码检查，编译构建和自动化部署步骤 64](#_Toc58851968)

[4.6.2 更新流水线配置，实现代码变更触发的持续交付流水线 69](#_Toc58851969)

[4.6.3 流水线配置高级功能 70](#_Toc58851970)

[4.7 资源释放 72](#_Toc58851971)

前 言

简介

现代的软件工程中，敏捷让开发团队拥抱变化、快速迭代，DevOps则覆盖全流程，加入了运维环节，用于促进开发、运维和质量保障部门之间的沟通、协作与整合，实现工程效率最大化。本实验手册基于一个模拟案例项目【凤凰商城】，所有故事及操作均以此模拟案例项目为背景。通过还原软件开发不同阶段的场景，让读者置身软件项目开发中，并利用华为云DevCloud的便捷工具使读者能快速掌握用敏捷的方法和DevOps思想完成软件项目的迭代规划、以及软件的开发、测试、部署。

内容描述

本实验手册共包含4个实验。

**实验一 持续规划与设计实验**，通过使用DevCloud进行项目规划，掌握产品待办列表管理，管理迭代计划和开发进度，使用看板模式进行项目展示及管理。

**实验二 持续开发与集成实验**，通过使用DevCloud提供的Git代码托管服务进行源代码管理，并对用户故事【门店网络】、【账户登录】进行开发，以及使用功能分支的方式进行代码的开发合并。掌握代码托管服务代码管理，和配置分支策略。在代码开发完成后，使用DevCloud代码检查服务对源代码进行检查，从而保证代码开发的质量。然后通过创建＂编译构建任务＂，完成应用的Docker镜像打包以及推送。启动持续集成，实现代码变更后自动触发应用的Docker镜像打包以及推送。使用华为云提供的＂开源镜像站＂服务提高依赖包获取速度，提高自动化编译效率。

**实验三 持续测试与反馈实验**，通过使用DevCloud来管理项目的测试生命周期，掌握测试用例的编写和配置，管理套件测试规划，执行手工测试用例，并使用测试总览跟踪测试进度和质量。

实验四 持续部署与发布实验，通过使用DevCloud部署服务，将软件项目的发布包一键式部署到服务器主机上，从而让应用落地运行。同时利用DevCloud流水线服务，完成应用程序持续交付流水线的搭建。

读者知识背景

为了更好地掌握本实验手册内容，阅读本手册的读者应首先具备以下基本条件：

熟悉程序设计

了解软件工程

实验环境说明

本实验环境为真实的华为公有云平台<http://www.huaweicloud.com/>，本实验手册中所有工具的操作及使用均在该平台上进行。在学习过程中可通过华为云帮助中心进行咨询，网址为http://support.huaweicloud.com/。

# 持续规划与设计

## 敏捷项目规划

### 实验介绍

本实验基于一个模拟案例项目【凤凰商城】，所有故事及操作均以此模拟案例项目为背景。【凤凰商城】示例项目是一个汽车零部件配件电子商城，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1.0＂版本的开发，当前正在进行＂迭代4＂的规划。

当前状态：

刚刚接到业务部门的最后通牒，要求月底必须上线 【门店网络查询功能】和【账号系统】，可以在凤凰商城中查询各个门店的相关信息。

### 实验目的

学习如何使用DevCloud进行项目规划。

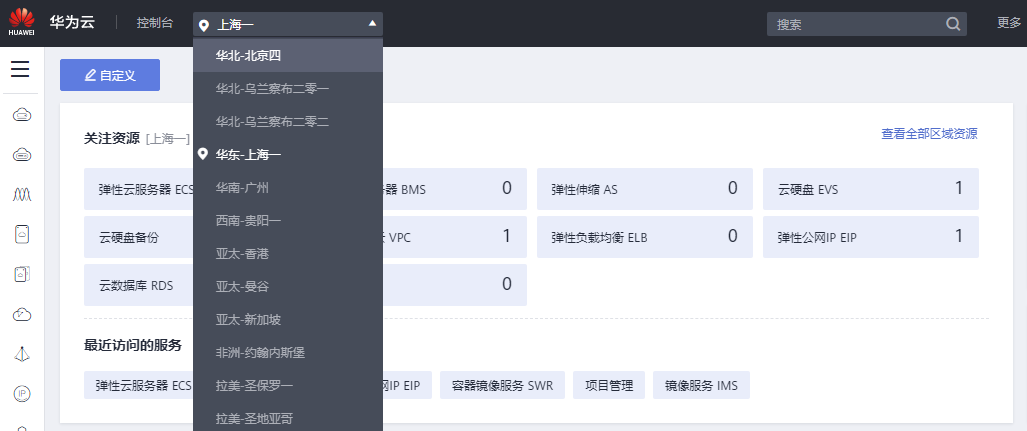
掌握如何进行产品待办列表管理。

掌握如何管理迭代计划，掌握开发进度。

掌握如何使用看板模式进行项目展示及管理。

### 创建华为云DevCloud项目

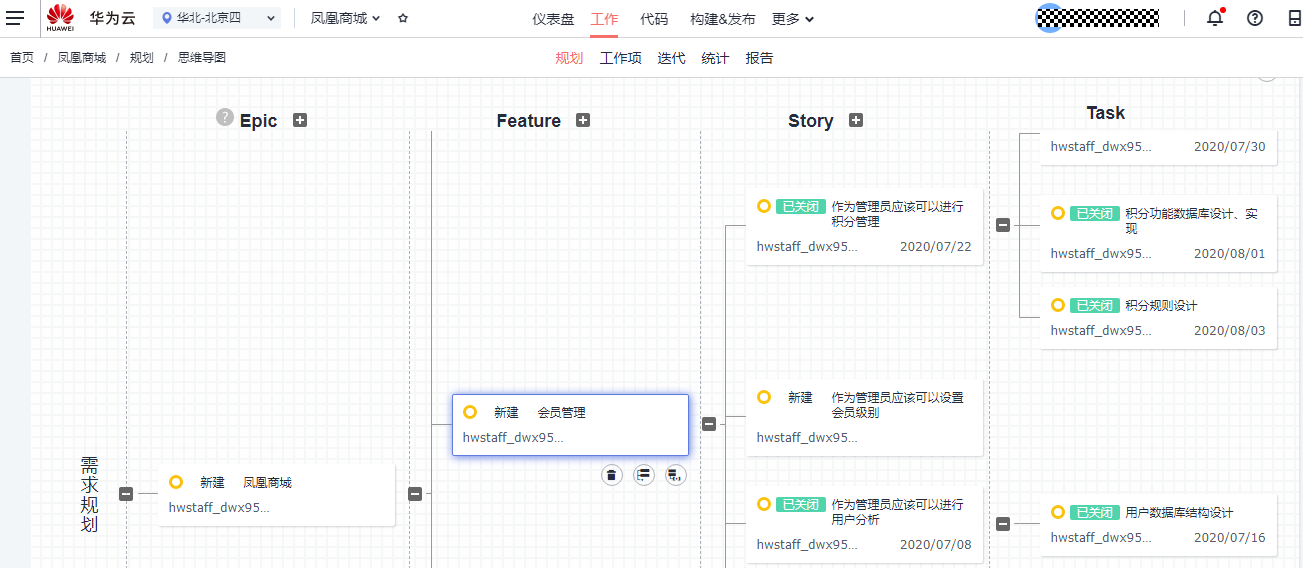
账号登录华为云http://www.huaweicloud.com/，进入控制台，在控制台中选择区域为＂华北-北京四＂。



搜索DevCloud并登陆，单击页面上方右侧＂新建项目＂。在弹窗中选择＂DevOps全流程样例项目＂，输入项目名称＂凤凰商城＂（可自定义），项目代号、描述可选项，单击＂确定＂，完成项目创建。

### 使用Scrum项目模板进行项目规划，并管理Epic和Feature

项目规划。打开凤凰商城项目，单击＂工作 > 规划＂，点击＂思维导图＂，项目规划视图以树形结构列出了需求从＂Epic>Feature>Story>Task＂的逐级关系。



下表是对不同类型工作项的说明：

|  |  |
| --- | --- |
| 工作项类型 | 说明 |
| Epic | 通常是公司重要战略举措。Epic的粒度比较大，需要分解为Feature，并通过Feature继续分解细化为User Story来完成最终的开发和交付。Epic通常持续数月（months），需要多个迭代才能完成最终的交付。 |
| Feature | 通常是对用户有价值的功能，相比Epic，Feature更具体形象，客户可以直接感知，通常在产品发布时作为ReleaseNotes的一部分发布给客户。通常持续数个星期（weeks），需要多个迭代完成交付。 |
| Story | User Story的简称。是从用户角度对产品需求的详细描述，更小粒度的功能。Story承接Feature，并放入有优先级的backlog中，持续规划、滚动调整优先级，始终让高优先级的Story更早的交付给客户。Story通常需要满足INVEST原则。Story通常持续数天（days），并应在一个迭代内完成交付。 |
| Task | 在迭代计划会议中，将纳入迭代的Story指派给具体成员，并分解成一个或多个Task，填写**“**预计工时**”**。 |

在Epic＂凤凰商城＂下方单击图标＂插入子节点＂新建Feature，输入标题＂门店网络＂，回车保存；相同方式新建Feature“账号系统”。

为Feature＂门店网络＂插入子节点，添加Story＂作为用户应该可以查看、查询所有门店网络＂；为Feature“账号系统”插入子节点Story“作为用户可以登录账号”、“作为用户可以注册账号”。

导出项目规划。您可以将项目规划导出到Excel，以条目化的方式查看以及管理。单击右上角图标C:\Users\swx941157\AppData\Roaming\eSpace_Desktop\UserData\swx941157\imagefiles\9F84C19B-D129-436D-9D34-6CC45A7C3747.png，在下拉列表中选择导出方式。

### 使用Scrum项目模板管理Backlog并进行迭代开发

通过本节，将学习如何使用项目流程管理来快速地制定团队计划，管理和追踪相关工作进度。

Backlog管理。在市场部门的施压下，Story＂作为用户应该可以查看、查询所有门店网络＂被指定为最高优先级处理，因为许多客户需要通过它来查询最近的门店网络地址，从而获取服务。

单击＂工作>工作项>Backlog＂，进入Backlog页面。

单击Story名称＂作为用户应该可以查看、查询所有门店网络＂，编辑Story。

可输入用户故事描述信息、开始日期、结束日期、预计工时、优先级、重要程度字段信 息。工时日期包含周末，应选“包含周末”选项。



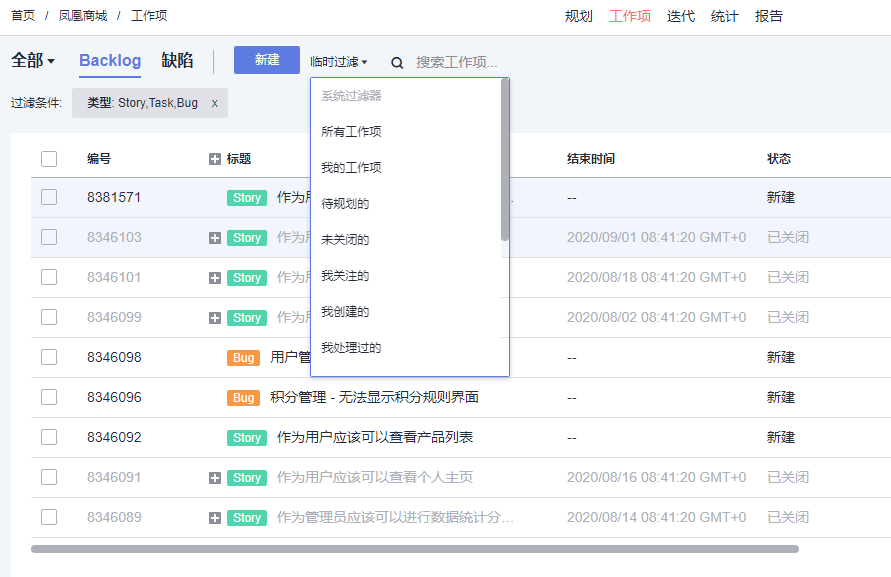
可通过本地文件拖拽到＂附件＂框中，作为工作项的附件。

完成编辑后，单击＂保存＂。

在迭代列表中，我们同时可以设置自己关注的用户故事，方便查询。拖动列表下方滚动条到最右方，单击图标即可关注，当图标变成蓝色，表示关注成功。

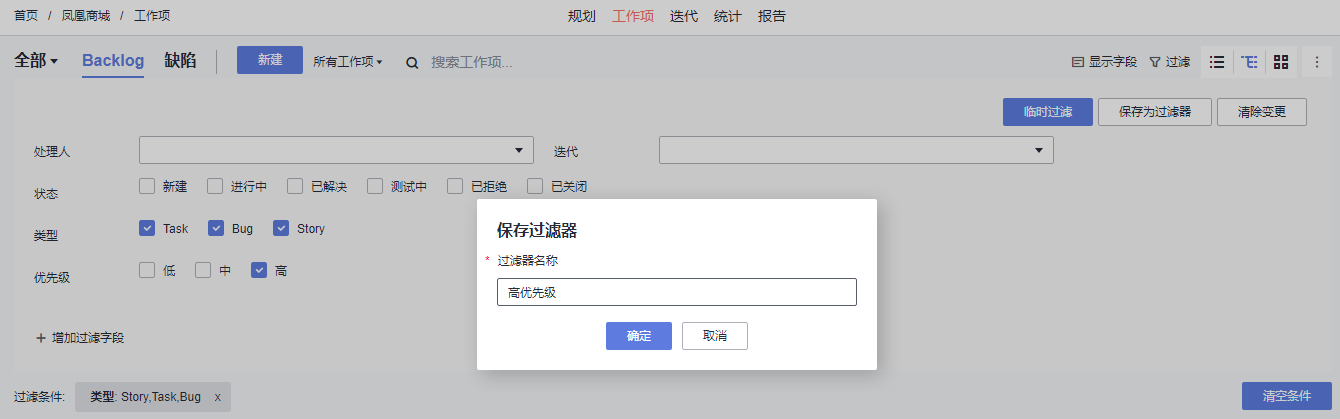
Backlog高级管理。用户可以通过快速过滤器方便地查询特定的工作项，也可以使用＂高级过滤＂实现特定字段指定条件的过滤。

我关注的：在Backlog页面，单击列表上方＂临时过滤＂下拉列表，选择＂我关注的＂，即可过滤已关注的工作项列表。



自定义过滤器。单击＂过滤＂，选择＂增加过滤字段＂，在弹框中勾选＂优先级＂，单击＂确定＂。在＂优先级＂字段勾选＂高＂，单击＂保存为过滤器＂，输入过滤器名称＂高优先级＂，单击＂确定＂保存。此时可＂所有工作项＂下拉列表中选择＂高优先级＂过滤工作项。





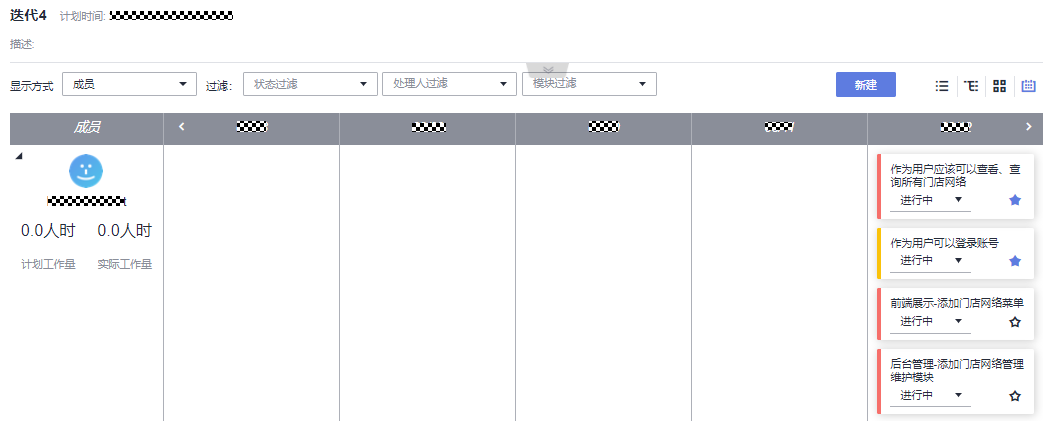
迭代的创建与管理。单击＂工作 > 迭代＂，进入迭代管理视图。单击＂创建迭代＂，在弹框中输入迭代名称＂迭代4＂、设置迭代计划时间，单击＂新建＂。本练习中，设置迭代设置迭代的开始日期为本周一，结束日期为下周五。按照同样方式创建迭代5，并设置迭代周期为下一个周期。

迭代规划。接下来我们需要对当前迭代（近两周）的工作进行规划，其中最重要的工作＂作为用户应该可以查看、查询所有门店网络＂需要在本迭代完成并上线。在迭代页面，单击＂未规划工作项＂，找到Story＂作为用户应该可以查看、查询所有门店网络＂，鼠标拖拽工作项至＂迭代4＂。单击＂迭代4＂的＂作为用户应该可以查看、查询所有门店网络＂，可以设置Story的预计开始日期与预计结束日期。 按照同样的方式，添加以下两个Story添加至迭代4：作为用户可以登录账号；作为用户可以注册账号。

Story分解。接下来我们需要将Story＂作为用户应该可以查看、查询所有门店网络＂拆分到开发任务级别，并指派给对应的负责人。单击工作项列表最右侧＂操作＂列中的图标 ，添加子工作项。输入Task标题＂前端展示-添加门店网络菜单＂，并选择处理人，单击＂确定＂完成。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＂。

看板视图使用。点击右上角图标，切换视图为＂卡片模式＂。在此模式下，可通过拖拽修改工作项状态。

在每日站立会议中，可以通过电子白板报告和更新任务进度。在迭代页面单击右上角图标，可切换视图为＂成员模式＂。通过成员模式视图可以看到当前团队每个成员的工作饱和度以及每个需求预估的完成时间点。



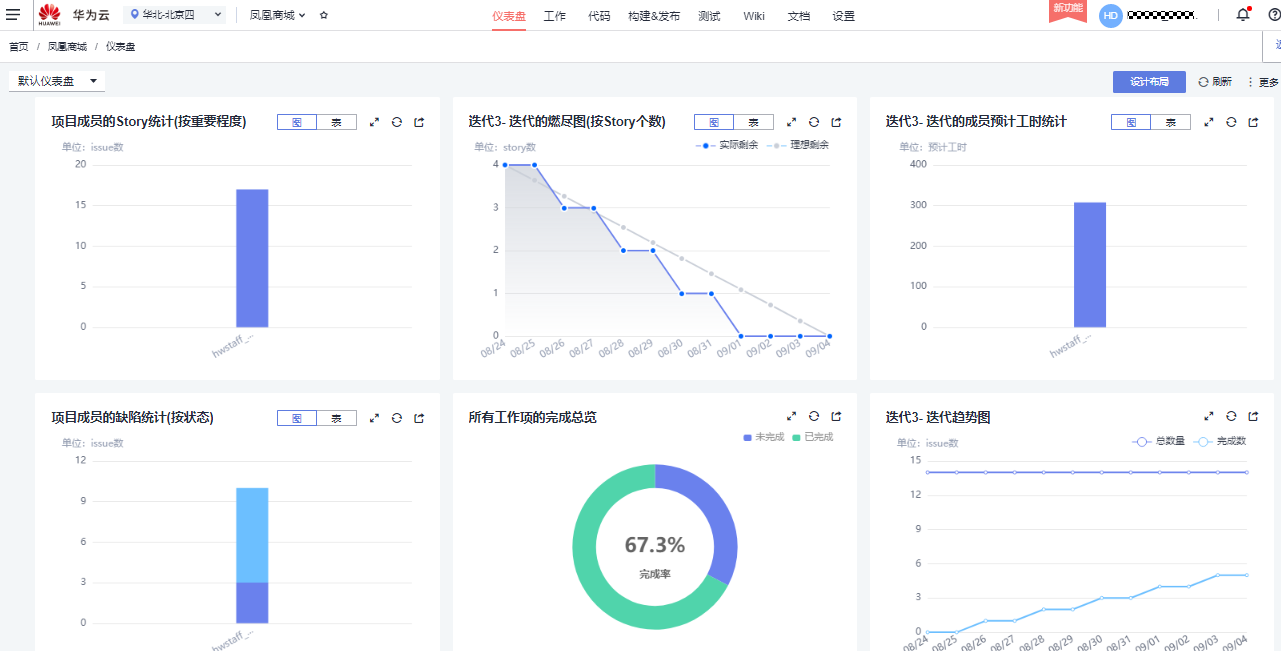
### 使用效率工具监测和跟踪项目状态

通过迭代图表，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。

在迭代页面，单击工作项列表上方的图标，即可展开迭代进度视图。如果是新创建的项目，在项目创建第二天可看到报表效果。



项目仪表盘查看以及管理。项目仪表盘提供了强大的项目进度跟进能力、包括需求进度统计、燃尽图、工作完成度、工时统计等等。 单击页面上方导航＂仪表盘＂，进入＂仪表盘＂页面。



单击页面右上角＂设计布局＂，进入编辑模式。点击任意组件右上角的，可将此组件移除。通过鼠标拖拽，可改变组件在仪表盘中的位置。单击页面右上角＂添加卡片＂，在页面右侧滑框中可拖拽新组件至仪表盘中。完成编辑后，单击页面下方＂保存＂。

## 项目管理配置

### 实验介绍

通过本章的学习，大家可以了解并掌握针对具体项目可以进行哪些维度的配置和自定义修改，帮助大家将项目配置的更接近自己真实的场景及应用方式。

### 实验目的

通过此次实验学员可以熟练掌握如何维护项目基本信息，成员和通知功能，如何定制项目工作流程。

### 维护项目基本信息

通过本章节您将学习如何对 ＂凤凰商城＂ 进行基本信息等维护，以及团队成员管理，并学习如何通过定制站内消息通知以及邮件通知来跟进项目进度。

项目基本信息设置。进入项目，单击页面上方导航＂设置 > 通用设置＂。



单击页面左侧菜单＂基本信息＂，进入＂基本信息＂页面。管理员可根据情况进行修改项目名称、项目描述，移交创建人，完成编辑，单击＂保存＂。



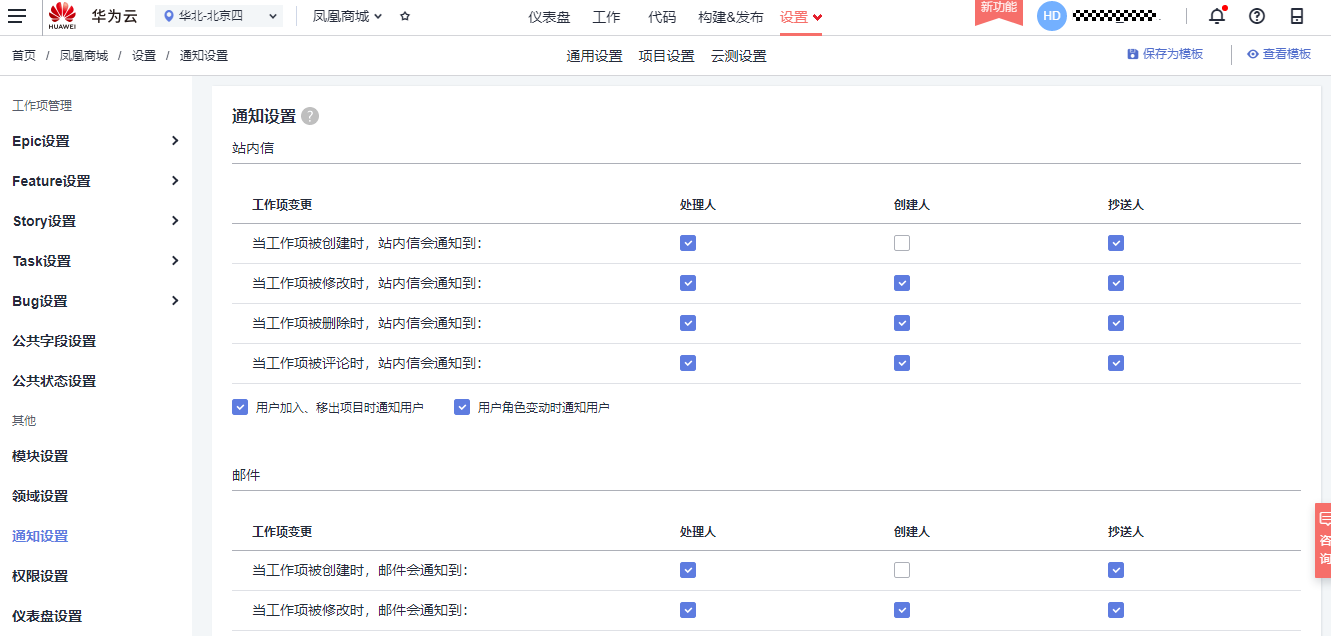
项目成员管理。单击页面左侧菜单“成员管理”，进入“成员管理”页面。点击添加成员，在下拉菜单中选择“邀请其他企业用户”，在弹出窗口的“企业用户”文本框输入小组成员账号进行添加。重复该步骤直到小组全部成员添加完毕。





作为参与者被添加到项目中后，成员账号登录DevCloud可看到参与的项目，并进行操作。

站内通知以及邮件通知设置。进入项目，单击页面上方导航＂设置 > 项目设置＂。单击页面左侧菜单＂通知设置＂，进入＂通知设置＂页面。根据需要勾选通知场景、方式、对象即可，系统将自动保存。

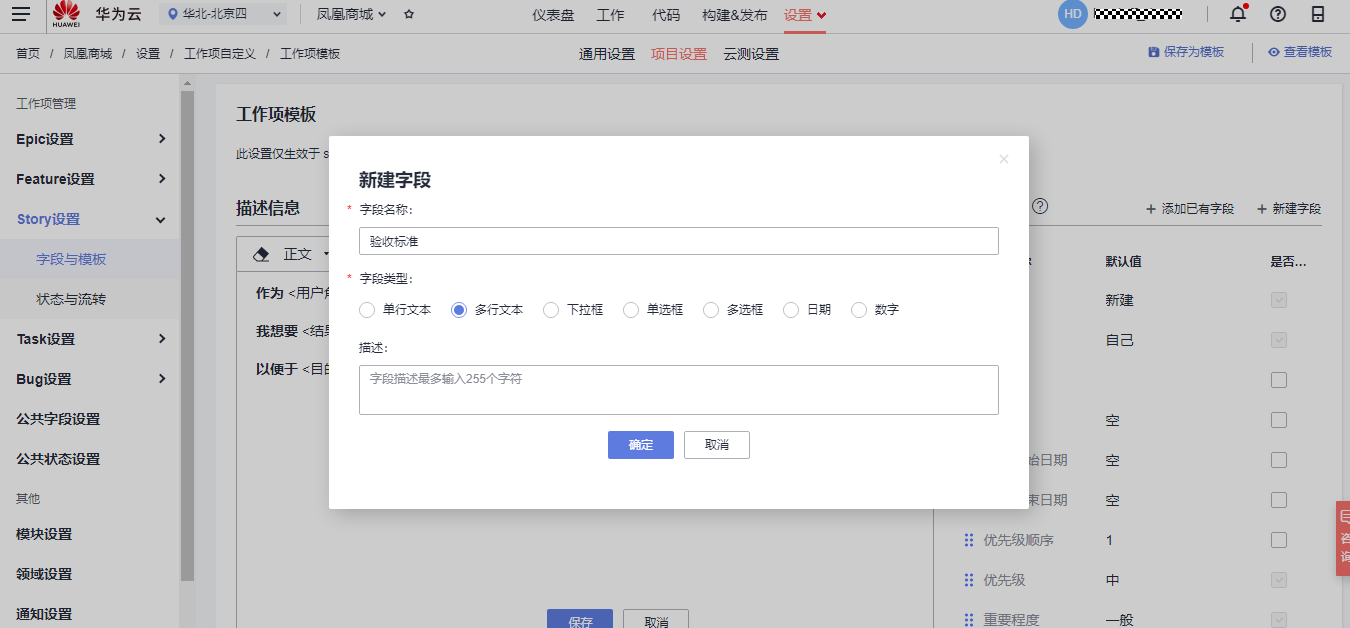


模块添加与管理。进入项目，单击页面上方导航＂设置 > 项目设置＂。单击页面左侧菜单＂模块设置＂，进入＂模块设置＂页面。单击＂添加＂，输入名称、描述，选择负责人，单击＂确定＂保存。

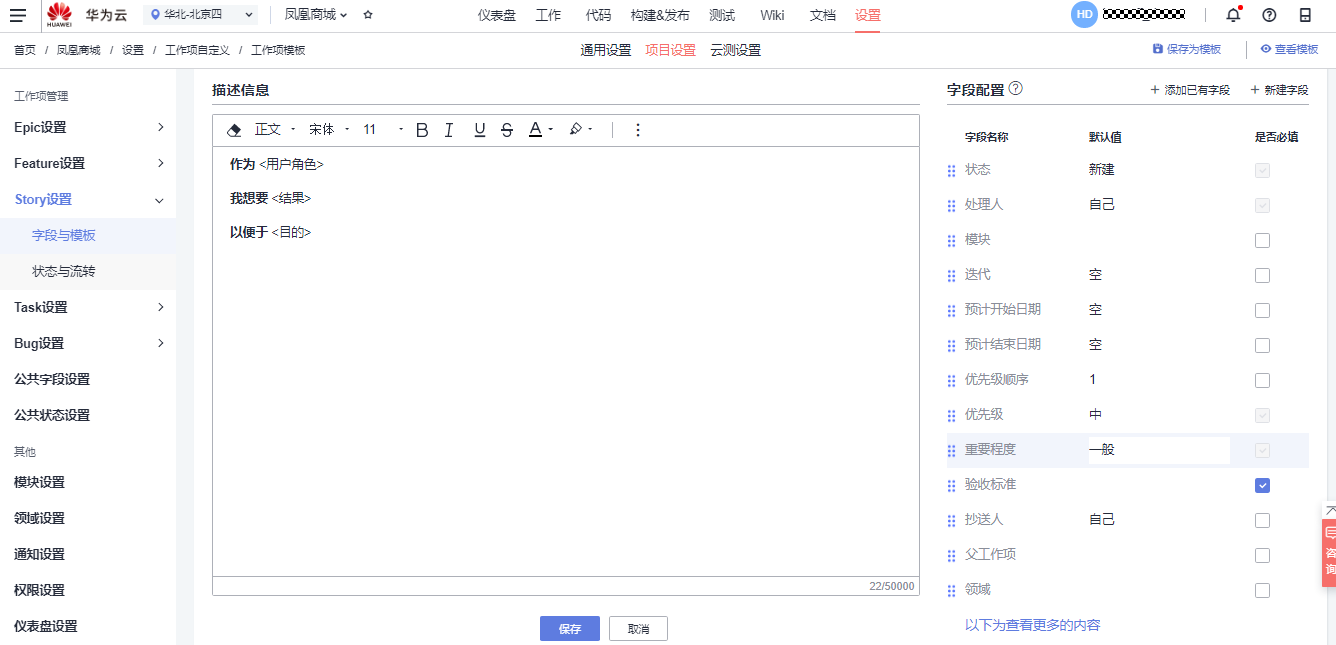
### 定制项目工作流程

通过本节，您将学习如何使用＂自定义工作项模版＂功能来定制个性化的工作项表单，以及工作流程定制。

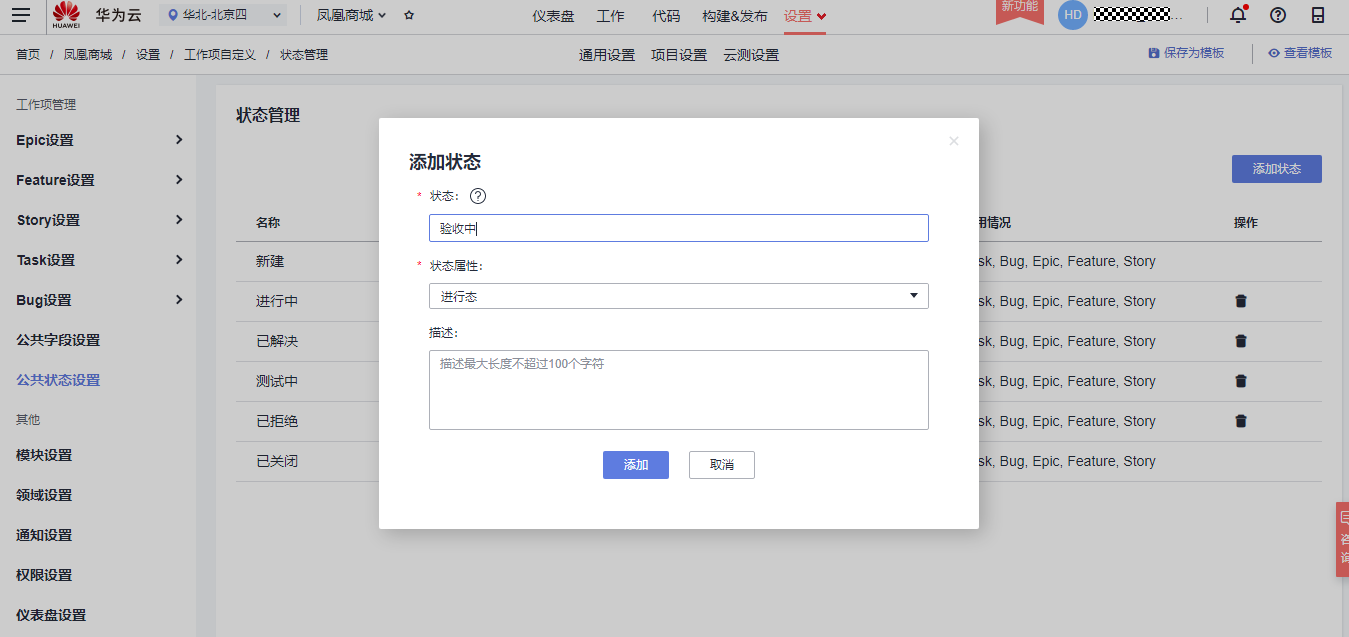
增加工作项字段。进入项目，单击页面上方导航＂设置 > 项目设置＂。单击页面左侧菜单＂Story设置 > 字段与模板＂，进入＂工作项模板＂页面。单击页面右上角＂编辑模板＂，选择＂新建字段＂。在弹框中输入字段名称＂验收标准＂，勾选字段类型＂多行文本＂，单击＂确定＂保存。



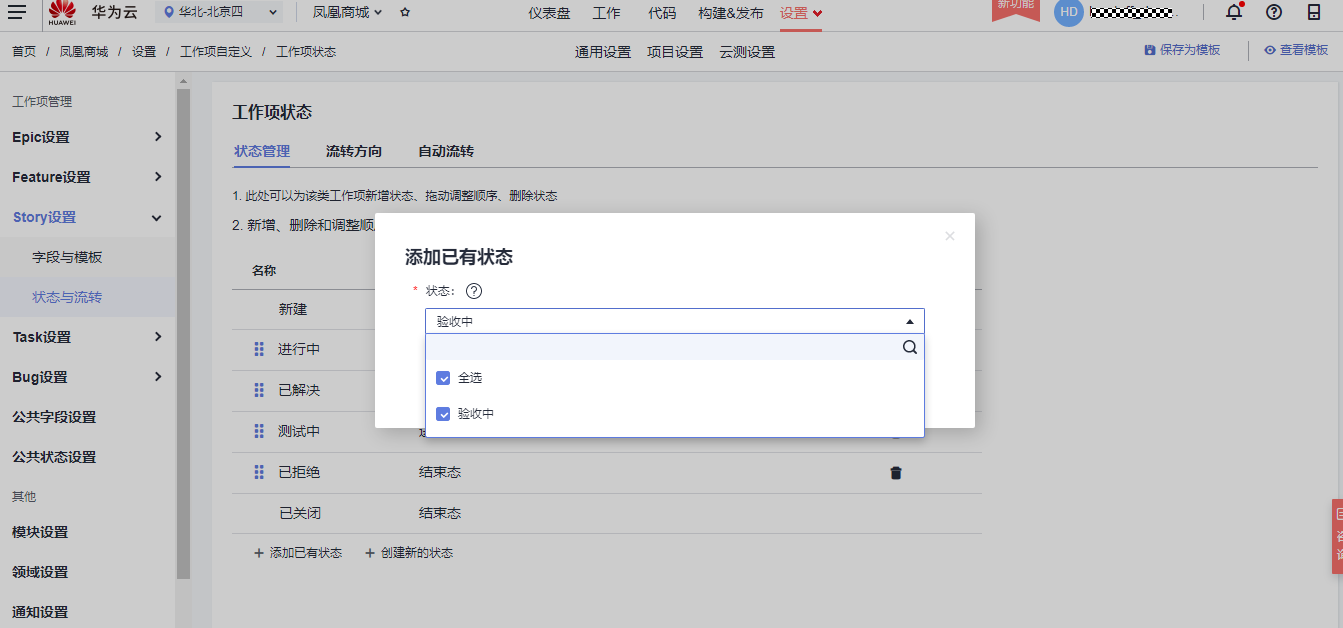
将＂验收标准＂拖至＂以下为查看更多的内容＂上方，并勾选为必填字段，单击＂保存＂。



增加工作项状态。进入项目，单击页面上方导航＂设置 > 项目设置＂。单击页面左侧菜单＂公共状态设置＂，进入＂状态管理＂页面。单击右上角＂添加状态＂，在弹框中输入状态＂验收中＂，选择状态属性为＂进行态＂，单击＂添加＂保存。



单击页面左侧菜单＂Story设置 > 状态与流转＂，进入＂工作项状态＂页面。单击＂添加已有状态＂，在弹框中勾选＂验收中＂，单击＂确定＂保存。



通过拖拽将该状态的顺序至于＂测试中＂之后。



# 持续开发与集成

## 代码托管

### 实验介绍

在本实验中，您将了解如何使用DevCloud提供的代码托管服务进行源代码管理，并对Story“作为用户应该可以查看、查询所有门店网络”进行开发，以及如何使用功能分支的方式进行代码的开发合并。

### 实验目的

通过此次实验学员可以熟练掌握如何使用代码托管服务进行代码管理，如何配置分支策略。

### 使用CloudIDE修改和提交代码

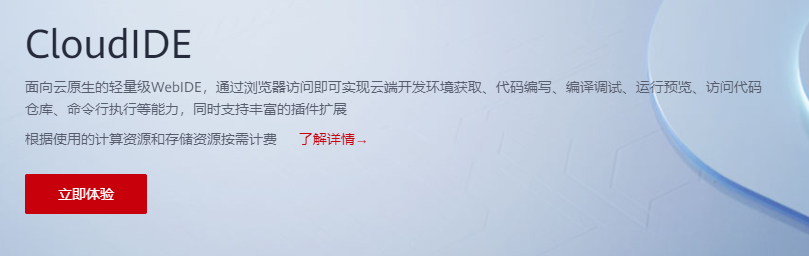
通过本节，您将学习如何通过CloudIDE完成需求＂添加门店网络菜单＂的代码开发。

登录DevCloud，单击＂工作>工作项＂，进入＂Backlog＂页面。找到Story＂作为用户应该可以查看、查询所有门店网络＂，记录该Story的编号。



在华为云主页<http://www.huaweicloud.com/> “产品”栏选择“开发者”项下的“CloudIDE”，“立即体验”进入CloudIDE服务。

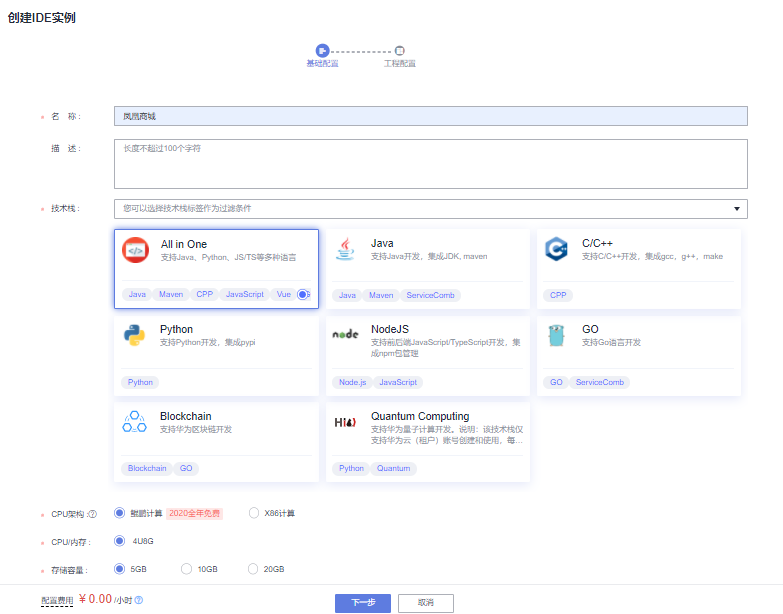




选择“我的IDE”栏，“新建实例”。



自定义IDE实例名称，其他项可保持默认设置，点击“下一步”。



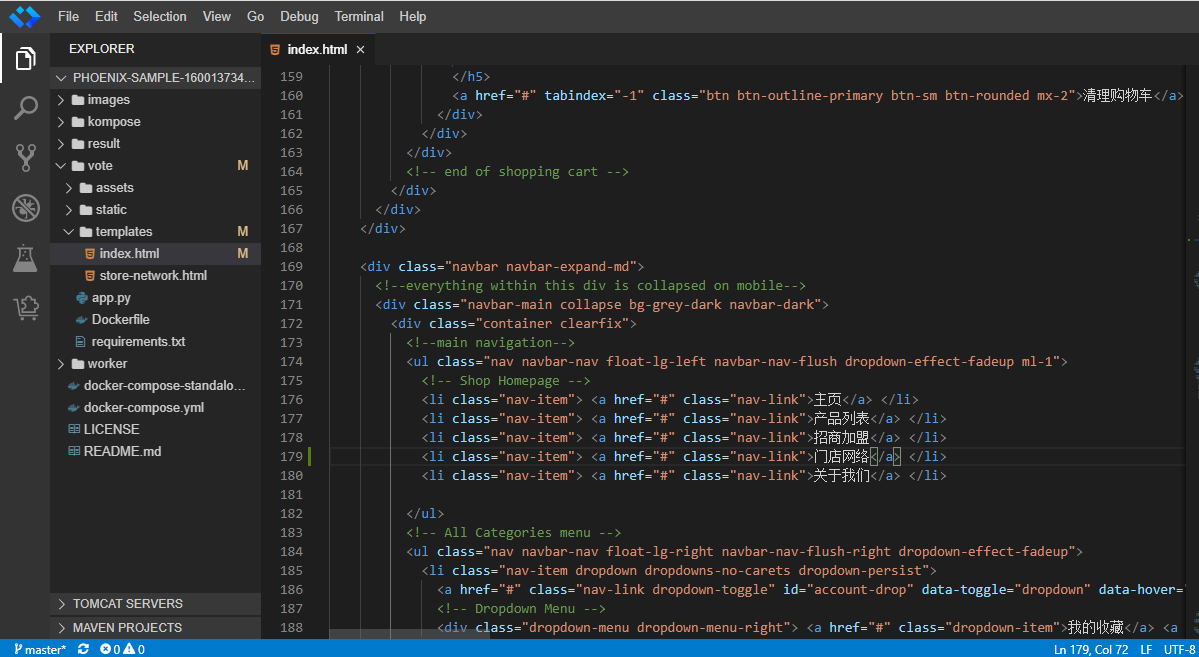
来源选择“私有仓库”，项目名称选择您在DevCloud创建的样例项目名称，仓库地址选择“phoenix-sample.git”，确定。



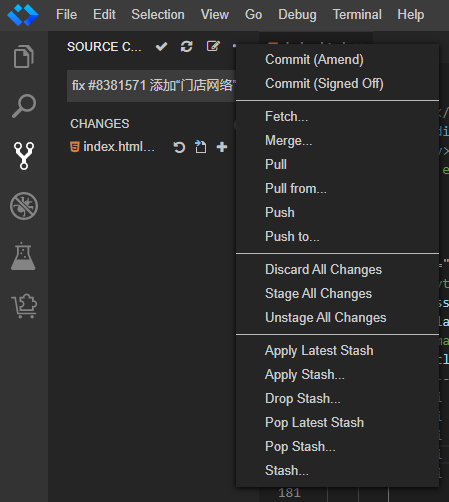
注：也可通过单击DevCloud项目页面上方导航＂代码>代码托管＂，进入代码托管服务。进入样例项目预置的代码仓库＂phoenix-sample＂，点击右上角“CloudIDE”，跳转至CloudIDE。（从此处入口进入开发调试有一定限制，推荐按步骤2、3新建CloudIDE实例进行软件开发）



修改代码。等待CloudIDE加载完毕后，在左侧导航中找到前端界面代码文件＂/vote/templates/index.html＂并打开，在179行添加菜单＂门店网络＂。



提交修改至代码仓库。单击左侧边栏图标，打开Git功能。单击修改文件后方的图标 (git add)，将修改内容添加进提交内容当中。在输入框中输入提交信息：＂fix #工作项编码 本次提交的注释信息＂ (fix #工作项编码 格式要注意，若不一致则不能将提交信息关联到下一个步骤的提交历史查看 )。单击打勾图标 (git commit)提交本次修改。单击图标，在下拉列表中单击＂推送Push＂，推送代码到代码仓库。



**查看代码提交记录**。返回＂代码托管＂页面，在phoenix-sample代码仓库中选择＂历史＂页签，即可查看是否提交成功。



跳转至＂工作>工作项>Backlog＂页面，单击Story＂作为用户应该可以查看、查询所有门店网络＂。在＂关联＂页签中，单击＂代码提交记录＂，也可看到相应提交记录。



预览调试。回到CloudIDE页面，打开index.html，点击右上角按钮，可预览前端界面。

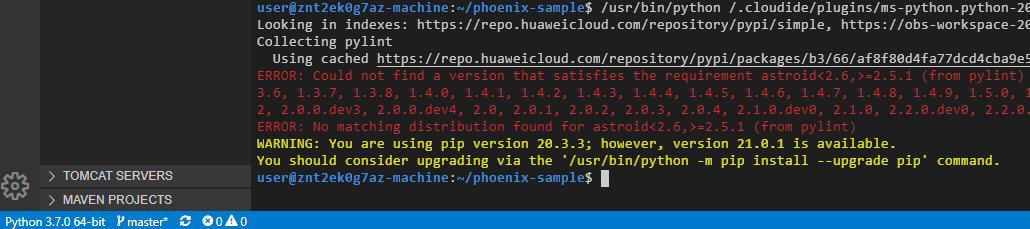


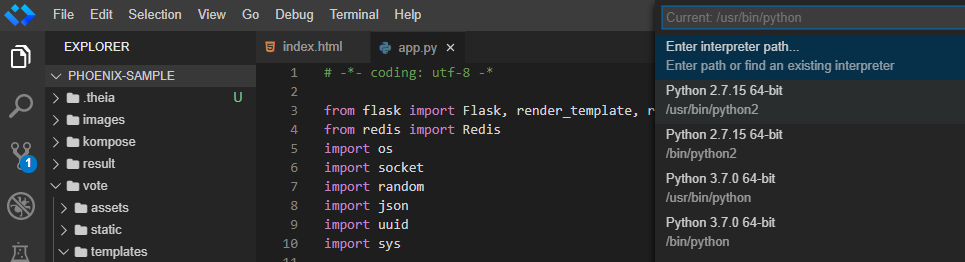
打开vote目录下的app.py源文件，如右下角出现“Linter pylint is not installed”，点击“Install”安装。





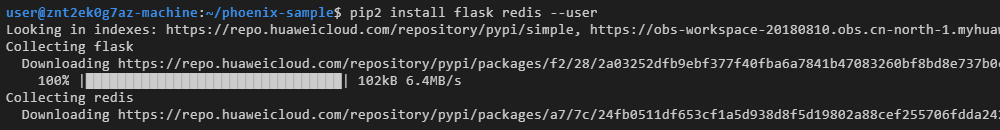
如出现下图报错信息，点击左下角“Python 3.7.0”在窗口上部选择“Python2.7.15 64-bit”版本，重新安装Linter pylint。





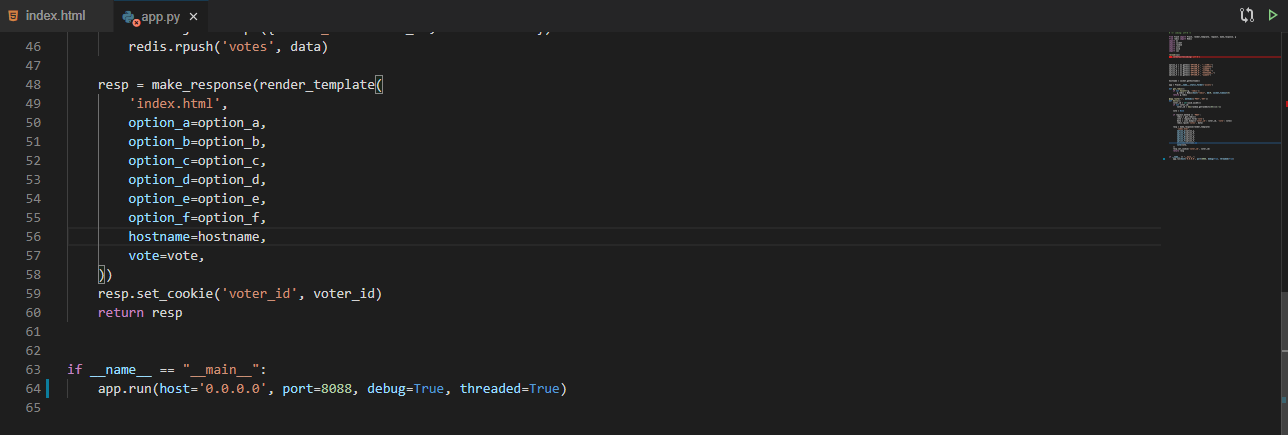
在终端窗口安装两个运行必须模块：

pip2 install flask redis --user

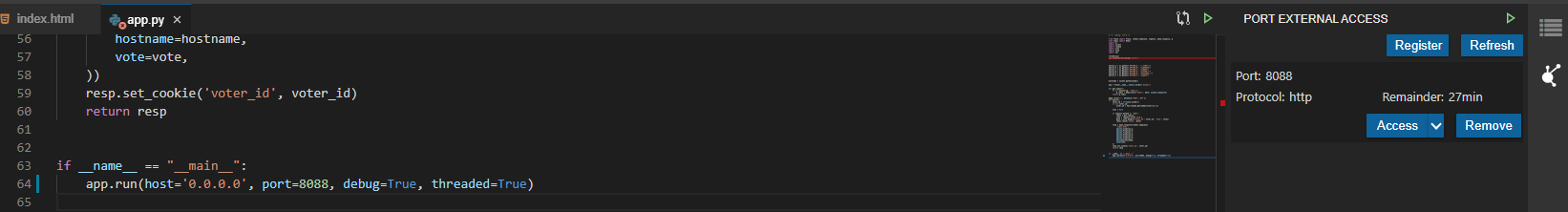




安装完成后将app.py中第64行代码中的端口号改为8088（如果端口已占有，8000-9000皆可），点击右上角运行图标，此时前端程序已执行。



再将对应端口号开放，允许访问：点击右上图标，输入端口号8088，点击OK，等待其变为”访问”，点击”访问”会转到浏览器打开前端页面，可以看到主页上已经有了“门店网络”菜单栏。





如果程序代码报错，可使用CloudIDE中的Debug进行调试。

程序运行预览成功以后，需要把app.py源程序代码原第64(新行号？)行的端口改回80端口（docker容器内外的端口映射）。

### 使用Git分支+合并请求方式提交代码并进行代码检视

通过本节，您将学习如何通过功能分支以及拉取请求的方式完成用户故事的开发，以及在线代码检视。

背景：收到新的邮件通知

发件人：开发经理

收件人：凤凰商城开发团队

优先级：高

主题：凤凰商城开发模式 - 建议使用功能分支+合并请求的方式

Hi 兄弟们：

为了保证凤凰商城的稳定运行，我们需要有一个稳定的持续可用的分支＂Master＂，所以我不建议任何人直接在Master分支上进行代码开发。 请大家在后面的功能开发中统一使用功能分支+合并请求的方式进行功能的开发与合并。

而且我建议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。

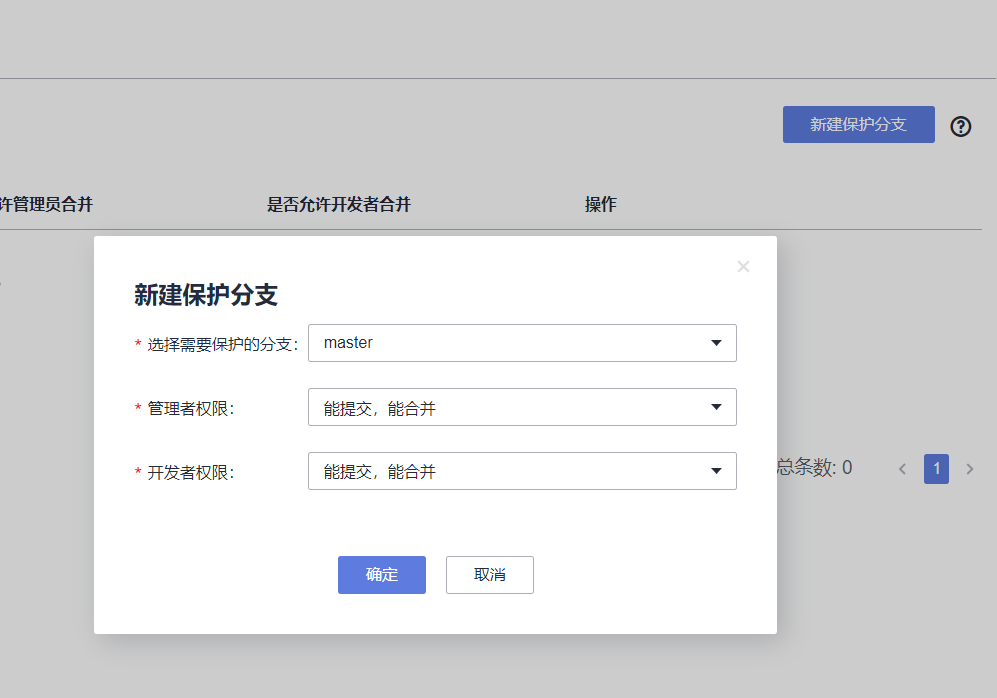
谢谢大家的配合。

将master分支设置为受保护分支。进入项目，单击页面上方导航＂代码 > 代码托管＂，进入代码托管服务。单击仓库名称，进入代码仓库。

选择＂设置＂页签，在左侧导航中单击＂仓库管理 > 保护分支管理＂。



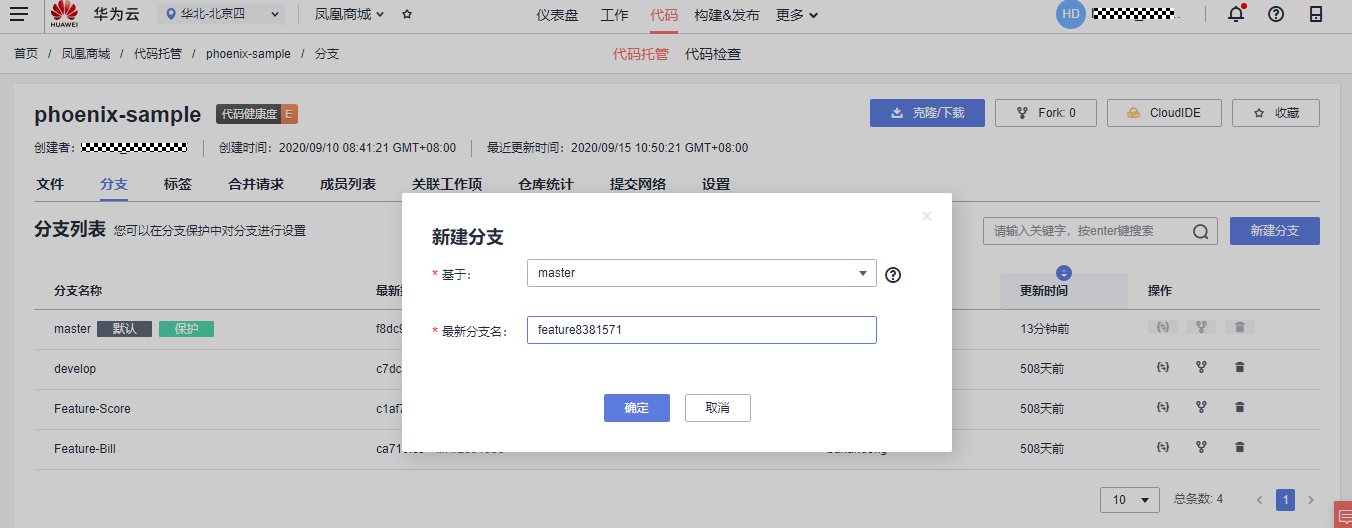
单击＂新建分支保护＂，根据需要在弹框中选择配置，单击＂确定＂保存。



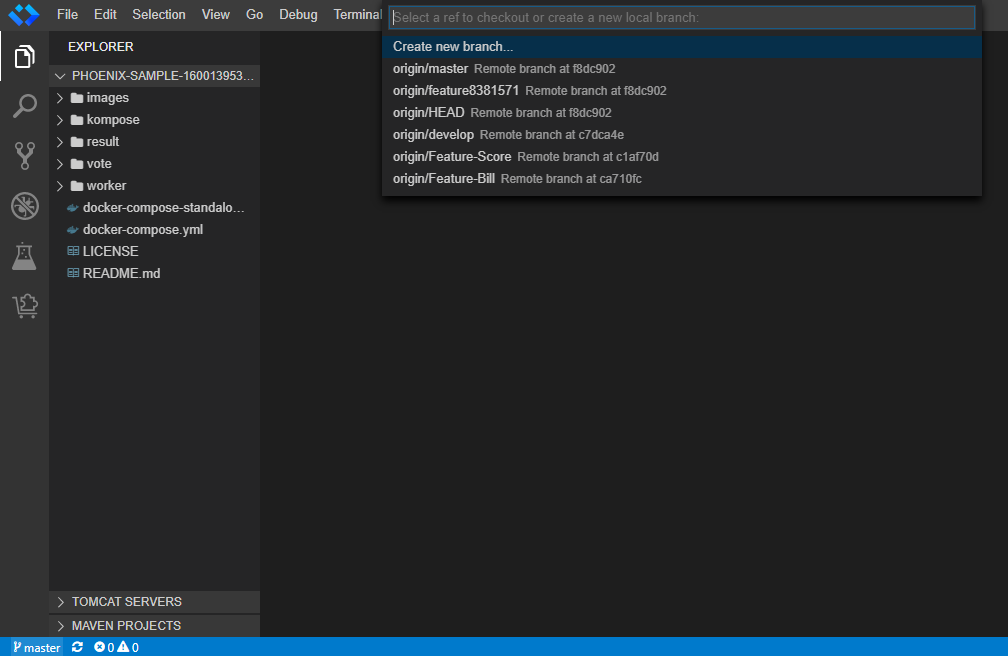
管理分支合并策略。进入代码仓库，选择＂设置＂页签，在左侧导航中单击＂基础设置 > 合并请求设置＂。选择允许分支合并的最低评分，单击＂确认＂保存。可以设置为＂0＂。



创建分支并进行新功能开发。进入代码仓库，选择＂分支＂页签，单击右上角＂新建分支＂，在弹框中输入分支名称＂feature工作项编号＂ (注意：不要有+号等特殊符号) 命名分支，单击＂确定＂保存。



用CloudIDE打开代码仓库，选择新创建的分支进入（如果是已打开的CloudIDE界面，可单击页面左下角分支名称，在页面上方的下拉列表中选择刚刚创建分支。可以看到左下角从＂master＂变成了新建分支名）。



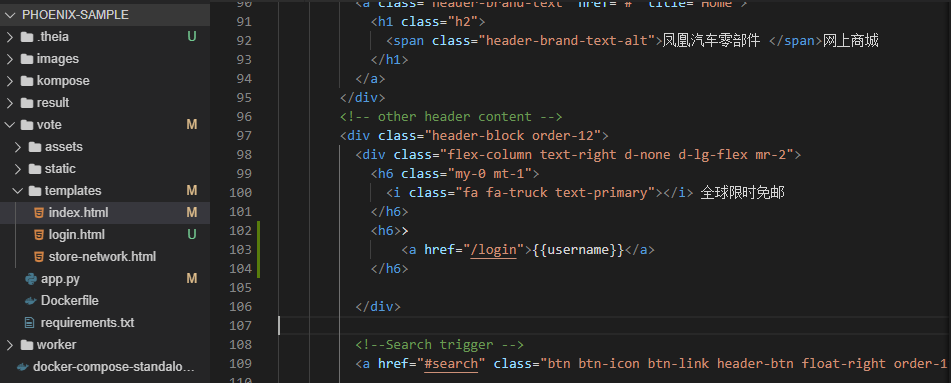
添加登录功能代码。

在/vote/templates/index.html文件的第102行添加如下代码

<h6>>

<a href="/login">{{username}}</a>

</h6>



在/vote/template目录下新建登录界面模板login.html

<![endif]-->

<!-- Fav and touch icons -->

<link rel="apple-touch-icon-precomposed" sizes="144x144" href="../assets/ico/apple-touch-icon-144-precomposed.png">

<link rel="apple-touch-icon-precomposed" sizes="114x114" href="../assets/ico/apple-touch-icon-114-precomposed.png">

<link rel="apple-touch-icon-precomposed" sizes="72x72" href="../assets/ico/apple-touch-icon-72-precomposed.png">

<link rel="apple-touch-icon-precomposed" href="../assets/ico/apple-touch-icon-57-precomposed.png">

<link rel="shortcut icon" href="../assets/ico/favicon.png">

</head>

<body>

<div class="container">

<form class="form-signin" action="/api/login" method="POST">

<h2 class="form-signin-heading">请登录 </h2>

<input type="text" name="username"class="input-block-level" placeholder="用户名">

<input type="password" name="password" class="input-block-level" placeholder="Password">

<h3>{{dict\_new}}</h3>

<label class="checkbox">

<input type="checkbox" value="remember-me"> Remember me

</label>

<button class="btn btn-large btn-primary" type="submit">登录</button>

</form>

</div> <!-- /container -->

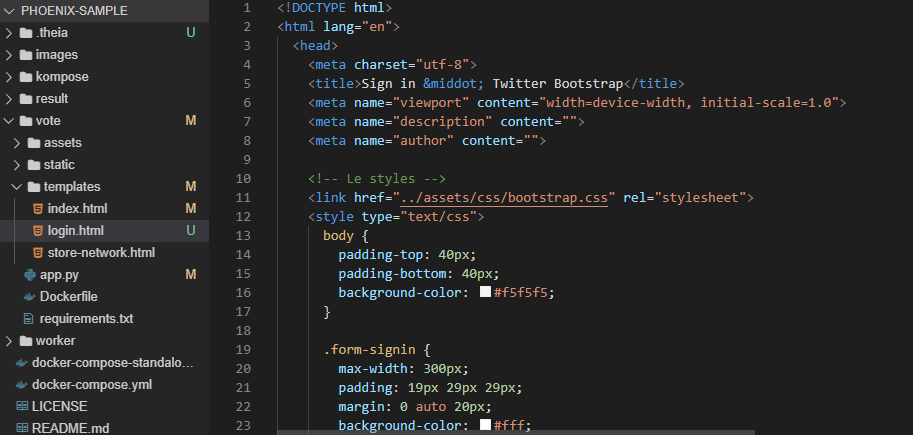
<!-- Le javascript

================================================== -->

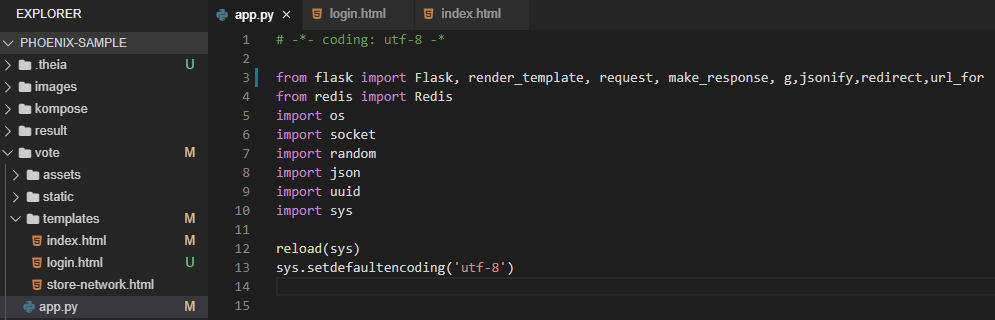
<!-- Placed at the end of the document so the pages load faster -->

</body>

</html>



在/vote/app.py文件的第3行增加” , jsonify, redirect, url\_for”三个包。



在/vote/app.py文件的47行添加如下代码。

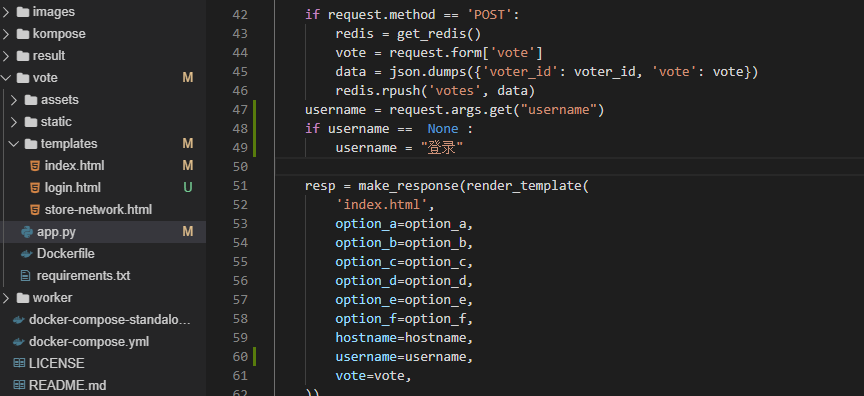
username = request.args.get("username")

if username == None :

username = "登录"

在60行添加

username=username,



在/vote/app.py文件的66行添加如下代码

#登录转发

@app.route('/login', methods=['GET'])

def login\_fowrd():

return render\_template('login.html')

@app.route('/api/login', methods=['POST', 'GET'])

def login():

if request.method == "POST":

username = request.form.get("username")

password = request.form.get("password")

if password == "123456" and username =="张三":

return redirect (url\_for('hello', username=username))

else:

dict\_new = "not passwod and word"

return render\_template("login.html",dict\_new=dict\_new)

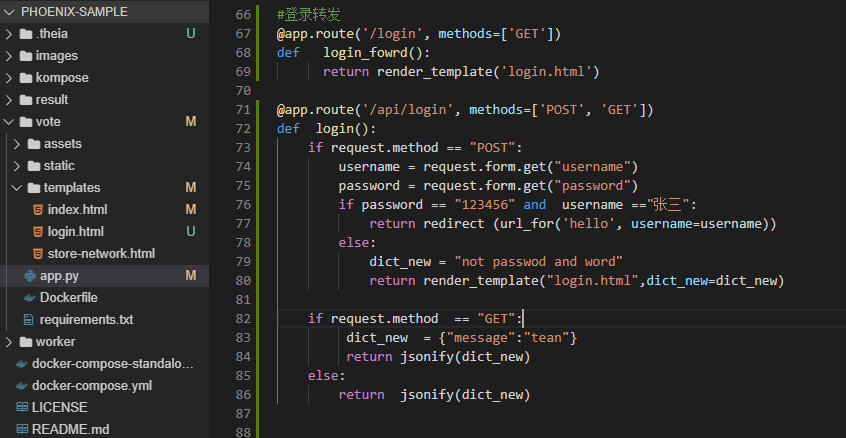
if request.method == "GET":

dict\_new = {"message":"tean"}

return jsonify(dict\_new)

else:

return jsonify(dict\_new)



参考2.1.3 使用CloudIDE修改和提交代码-步骤7操作，对修改后的代码进行调试。调试运行后可看到商城页面新增了“登录”链接



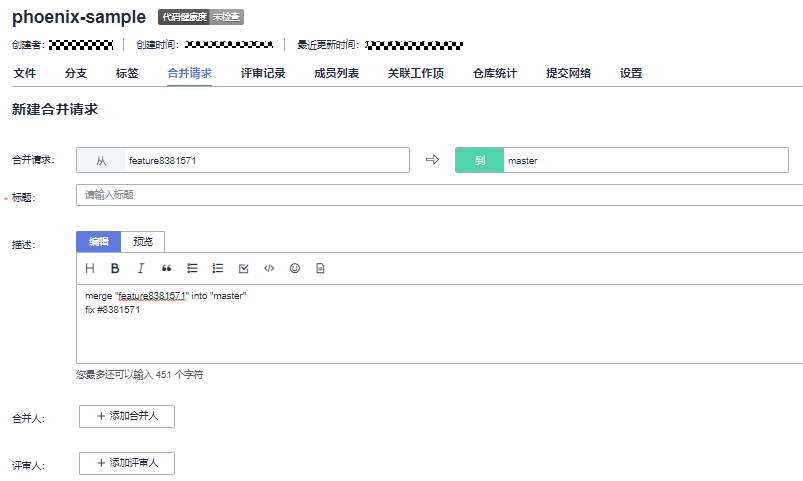
点击“登录”，在登录框中输入用户名“张三”，密码“123456”，点击“登录”跳转到如下页面，则登陆成功。（这里账号密码已写死，后续可以进一步关联数据库完善账号功能模块）





参考2.1.3 使用CloudIDE修改和提交代码-步骤4操作，将修改的代码推送至代码仓库。

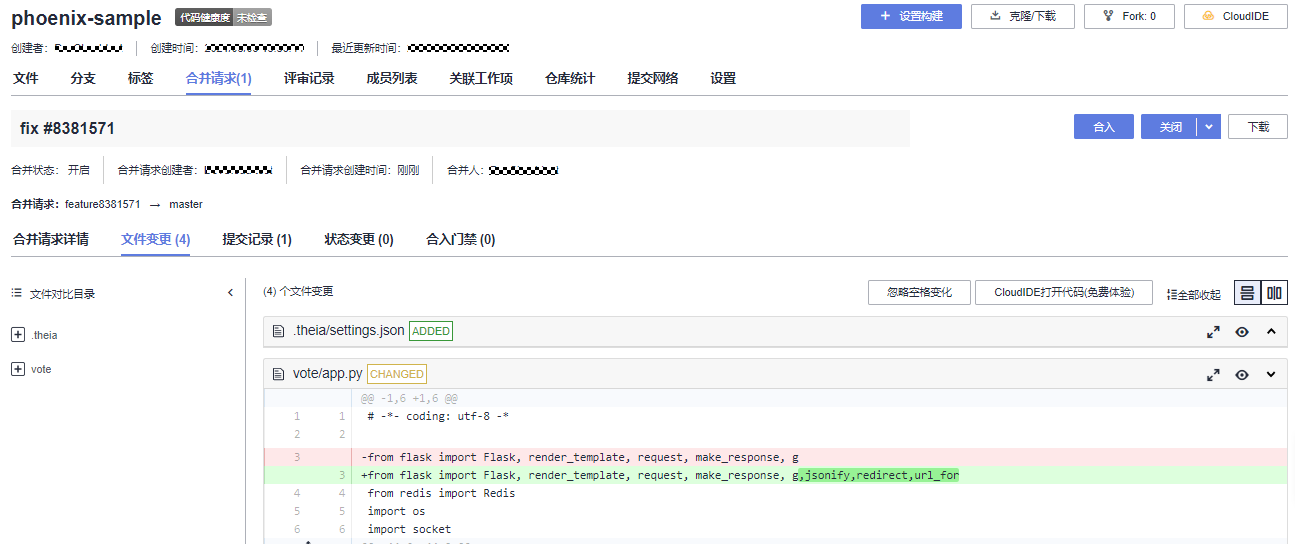
创建合并请求，进入代码仓库，选择＂合并请求＂页签，单击＂新建合并请求＂。源分支选择刚刚创建的分支，与目标分支选择＂master＂，单击＂下一步＂。输入标题、描述（选填），选择合并人、评审人，单击＂确定＂完成。＂合并人＂即接受合并请求的人，＂评审人＂是由合并发起人邀请参与的评审者。对于不合格的合并请求，管理员可以关闭。在评分不够时，无法完成该合并请求。



代码检视以及评分，评审人进入代码仓库后，在＂合并请求＂页签中找到需要评审的合并请求，单击该请求，查看合并请求详情。评审者可以在＂合并请求详情＂页签中发表评论、发表讨论，对合并请求进行评分。



也可以在文件变更页签，针对某一行代码发表讨论。评论与讨论的区别在于：评论无需回复并解决。而讨论是一个需要解决的问题。在评论记录里，可以看到差别。



开发人员在＂合并请求详情＂页面可查评论/讨论记录。



评审成员提交评分。若只有一个账号，请将允许分支合并的最低评分设置为2分。

合并人进入代码仓库后，在＂合并请求＂页签中找到需要评审的合并请求，单击该请求，查看合并请求详情。选择＂合入＂，系统将提示＂合并成功＂。



代码托管服务提供了以下功能，能够满足更多的使用场景。

|  |  |
| --- | --- |
| 功能 | 说明 |
| [仓库统计](https://support.huaweicloud.com/usermanual-codehub/devcloud_hlp_00074.html) | 可以查看代码贡献量、提交记录、频率、涉及语言等信息。 |
| [IP白名单](https://support.huaweicloud.com/usermanual-codehub/devcloud_hlp_0734.html) | 只有在IP白名单范围内的仓库访问才是允许的，除此之外其他IP发起的访问一律被拒绝。 |
| [子模块设置](https://support.huaweicloud.com/usermanual-codehub/codehub_ug_1001.html) | Git为管理仓库共用而衍生出的一个工具。通过子模块您可以将公共仓库作为子目录包含到您的仓库中，并能够双向更新该公共仓库的代码。 |
| [WebHook设置](https://support.huaweicloud.com/usermanual-codehub/codehub_ug_0222.html) | WebHook允许开发人员通过订阅 CodeHub 的仓库的分支推送(push)，标签推送(tag push)等事件来触发自己系统的构建、更新镜像、部署等操作。 |
| 通知设置 | 当仓库发生重大变更（例如删除、冻结）时，系统会发送邮件通知。通过＂设置 > 基础设置 > 通知设置＂页面，可设置接收者范围。 |

## 代码检查确保编码规范

### 实验介绍

随着凤凰商城越来越庞大，线上出现的缺陷以及安全问题也越来越多，修复的成本太大了，其实很多问题是可以在上线之前就应该发现以及修复的。

而且开发人员写代码也比较随性，没有统一标准，我们需要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。这样才能保证我们上线的应用相对可靠。

在本实验中，您将了解如何使用DevCloud提供的＂代码检查＂服务完成针对不同技术栈的代码静态扫描，问题采集与修复。

本次实验主要完成以下内容：

学习如何创建代码检查任务，并执行代码检查任务，完成应用的静态代码检查。

学习如何根据代码检查结果分析代码中存在的问题，并解决代码问题。

学习如何新增以及管理规则集，定制自定义的代码检查规范。

### 实验目的

理解并掌握代码检查的任务配置功能

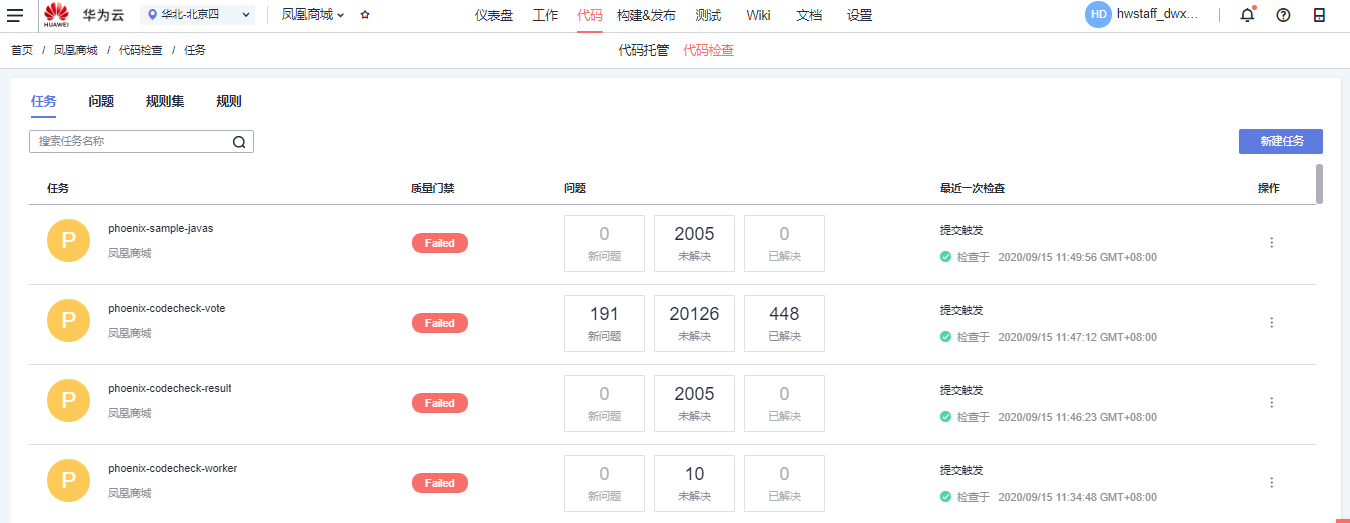
理解代码检查任务的分析结果并掌握如何修复

理解并掌握如何进行代码检查规则集的配置及管理

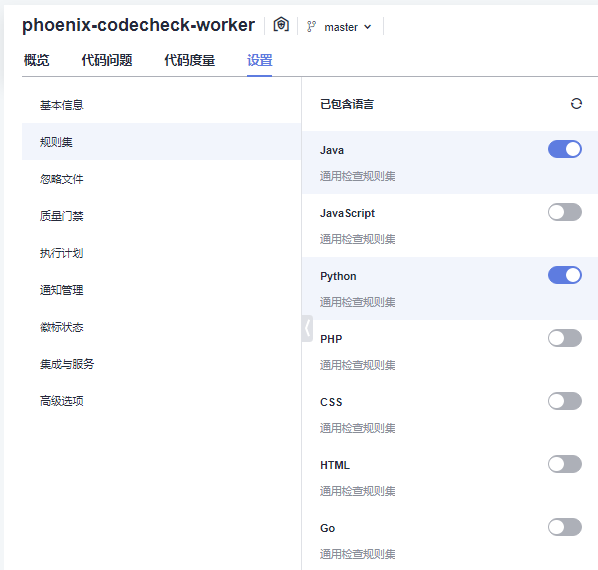
### 为示例项目配置代码检查任务并执行

通过本节，您将学习如何编辑并执行代码检查任务，完成应用的静态代码检查。

编辑代码检查任务包含语言。进入项目，单击页面上方导航＂代码 > 代码检查＂，进入代码检查服务。单击代码检查任务＂phoenix-codecheck-worker＂，进入＂代码检查详情＂页面。



选择＂设置＂页签，在页面左侧导航中单击＂规则集＂。规则集中默认包含的语言是＂Java＂。单击图标重新获取代码仓库语言，在刷新的列表中将Python语言对应的开关打开。



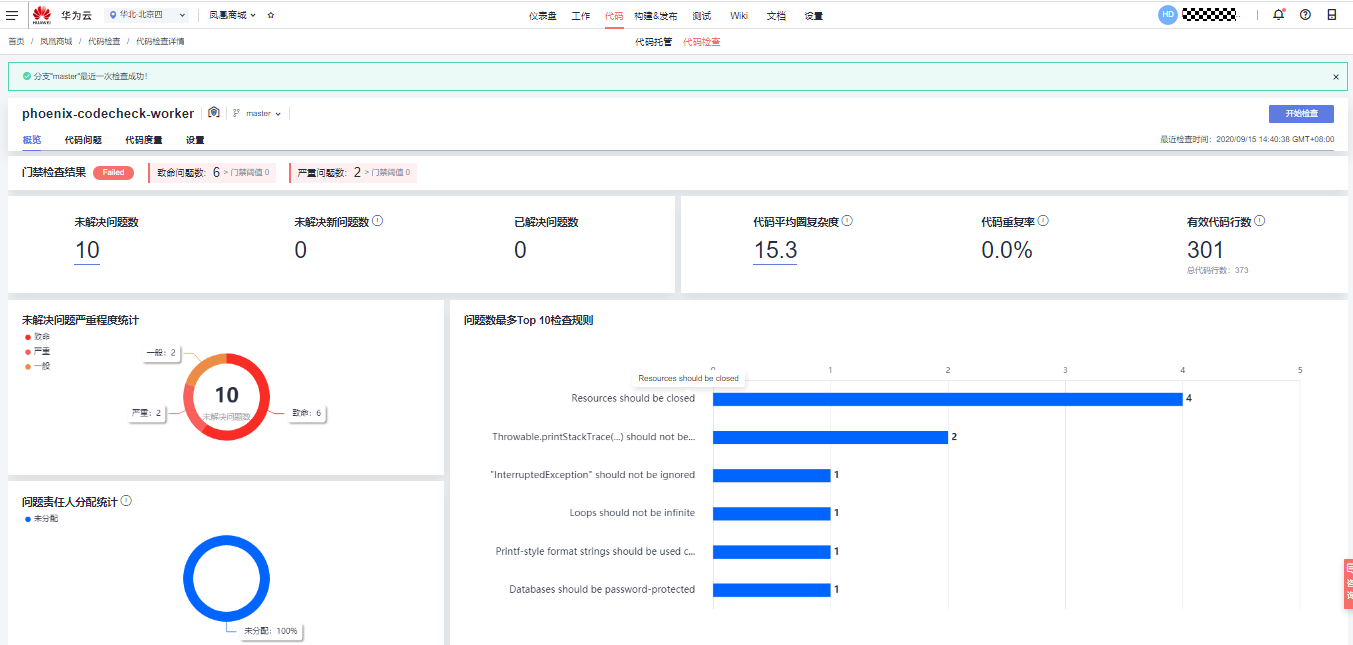
启动代码检查任务。进入代码检查任务＂phoenix-codecheck-worker＂的＂代码检查详情＂页面。单击＂开始检查＂，启动代码检查任务。当页面提示＂分支＂master＂最近一次检查成功！＂，表示任务执行成功。



### 分析代码检查结果并修复问题

通过本节，您将学习如何根据代码检查结果分析代码中存在的问题，并解决代码问题。

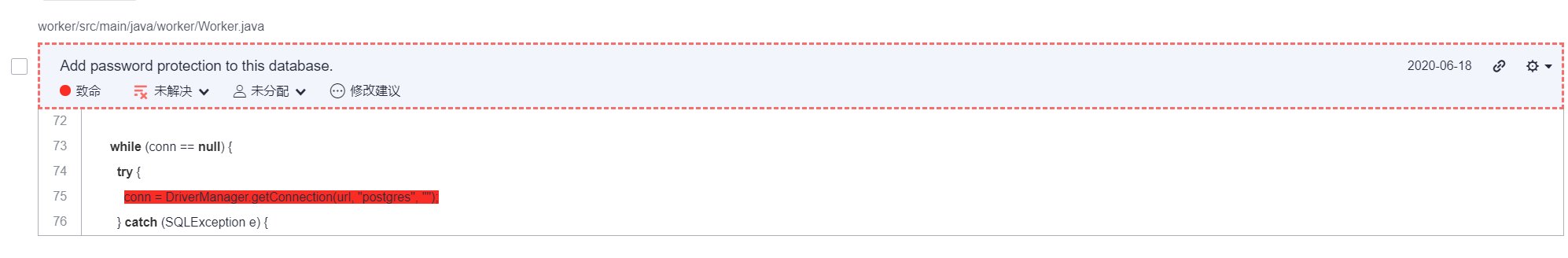
分析代码检查报表。进入代码检查任务＂phoenix-codecheck-worker＂的＂代码检查详情＂页面。成功执行代码检查任务，查看检查结果。



在代码检查任务＂设置 > 质量门禁＂页面中，默认配置＂致命问题数<=0＂和＂严重问题数<=0＂，从上图可以看出，检查结果不符合门禁要求，因此门禁检查结果显示为＂Failed＂。

单击＂代码问题＂页签，页面中列出问题的严重级别、状态、所属文件等信息。

可根据需要对代码进行修改。例如：下图中的问题，是在代码里Connection没有关闭可能会引起的资源泄漏问题，属于非常严重的问题，需要尽快修复。



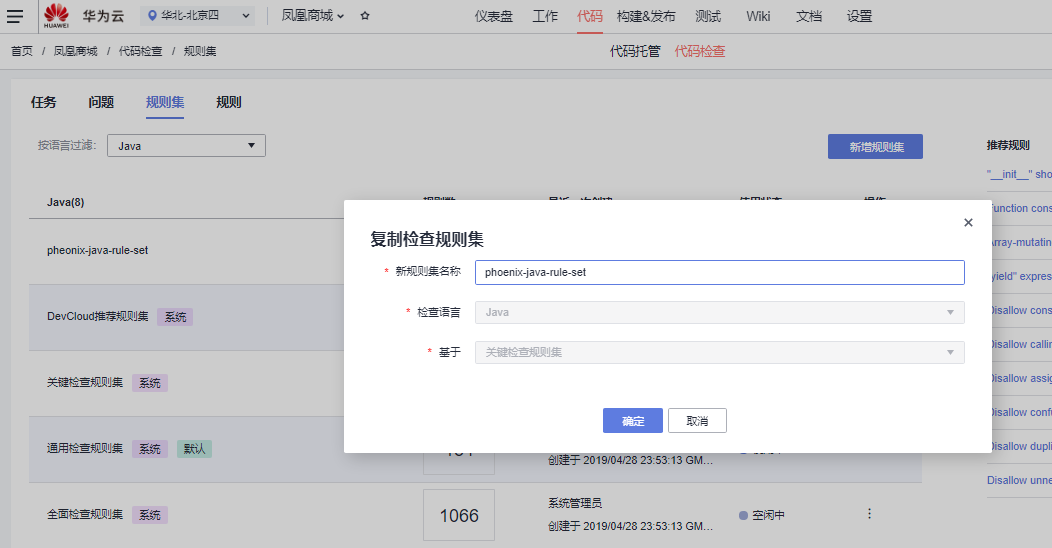
单击问题卡片，页面右侧将滑出窗口。单击＂修改建议＂，可查看系统对此问题的修改建议。单击窗口上方的图片，可进入相应的代码文件修改代码。单击＂未解决＂，可将此问题的状态设置为＂已解决＂或＂已忽略＂。



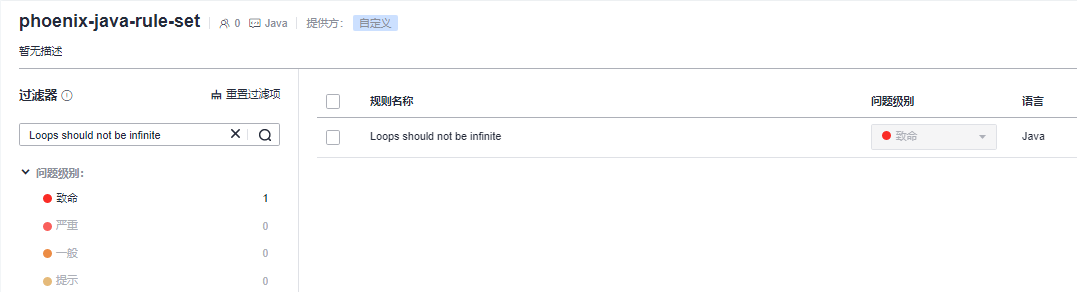
### 配置和管理代码检查规则集

通过本节，您将学习如何新增以及管理规则集，定制自定义的代码检查规范。

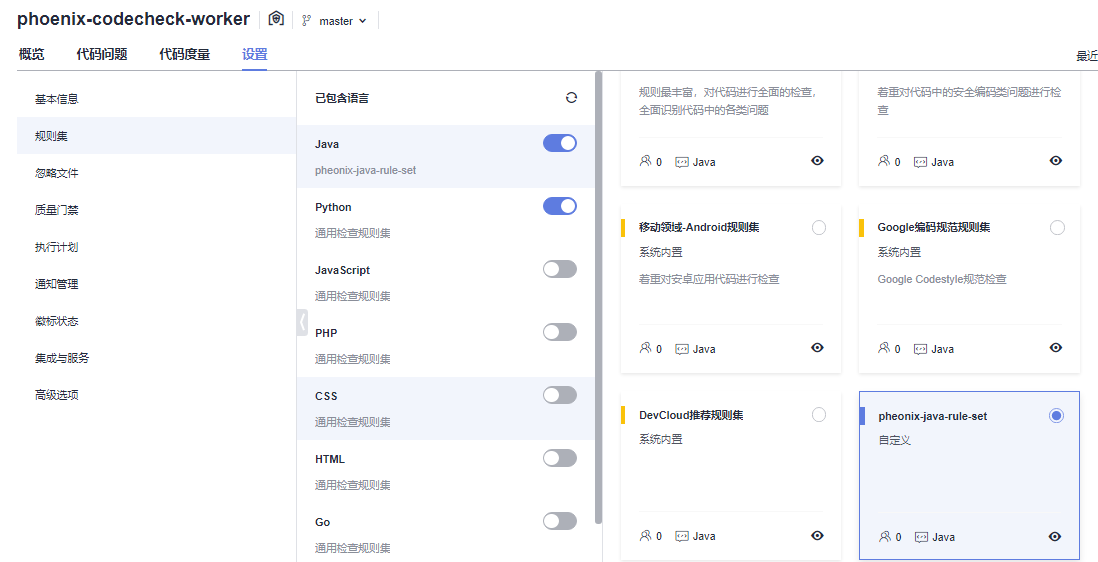
创建自定义检查规范。进入代码检查服务，选择＂规则集＂页签，在下拉列表中选择＂Java＂。在过滤出的列表中找到规则集＂关键检查规则集＂，单击图标 ，在下拉列表中选择＂复制＂。在弹框中输入新规则集名称为＂phoenix-java-rule-set＂，单击＂确定＂保存。



系统自动进入新建规则集的＂规则集配置＂页面。通过搜索框搜索规则＂Loops should not be infinite＂，取消勾选此规则，单击＂保存＂。由于这里凤凰团队采用了取巧的方案临时支持功能的展示，暂时可以关闭此问题。



启用自定义检查规范。进入代码检查任务＂phoenix-codecheck-worker＂的＂代码检查详情＂页面。选择＂设置＂页签，在页面左侧导航中单击＂规则集＂。在Java语言规则集中，勾选规则集＂phoenix-java-rule-set＂。



启动代码检查任务，任务成功执行后，再次查看检查结果。此时的检查结果忽略了规则＂Loops should not be infinite＂，检查出来的问题是团队关心的问题，应该尽快修复掉。



## 使用自动化编译提高代码质量验证速度

### 实验介绍

在本实验中，您将学习：

如何创建 ＂编译构建任务＂，完成应用的Docker镜像打包以及推送。

如何启动持续集成，实现代码变更后自动触发应用的Docker镜像打包以及推送。

如何使用华为云提供的＂开源镜像站＂服务提高依赖包获取速度，提高自动化编译效率。

编译构建功能介绍

编译构建（CloudBuild）基于云端大规模并发加速，为客户提供高速、低成本、配置简单的混合语言构建能力，帮助客户缩短构建时间，提升构建效率。

### 实验目的

理解并掌握DevCloud编译构建的功能与使用

理解SWR镜像仓库的功能与使用

掌握华为云＂开源镜像站＂的使用方法

### 编译构建并完成应用的镜像打包推送

通过本节，您将学习如何通过＂编译构建＂服务完成应用的Docker镜像打包以及推送。

样例项目中预置了以下5个编译构建任务，本章节以任务＂phoenix-sample-ci＂为例进行讲解。

|  |  |
| --- | --- |
| 预置编译构建任务 | 任务说明 |
| phoenix-sample-ci | 基本的编译构建任务。 |
| phoenix-sample-ci-test | 测试环境对应的编译构建任务。 |
| phoenix-sample-ci-worker | worker功能对应的编译构建任务。 |
| phoenix-sample-ci-result | result功能对应的编译构建任务。 |
| phoenix-sample-ci-vote | vote功能对应的编译构建任务。 |

查看编译构建任务。进入项目，单击页面上方导航＂构建&发布 > 编译构建＂，进入编译构建服务。找到编译构建任务＂phoenix-sample-ci＂。



单击＂phoenix-sample-ci＂对应操作列中的图标，进入＂编译构建编辑任务＂页面。



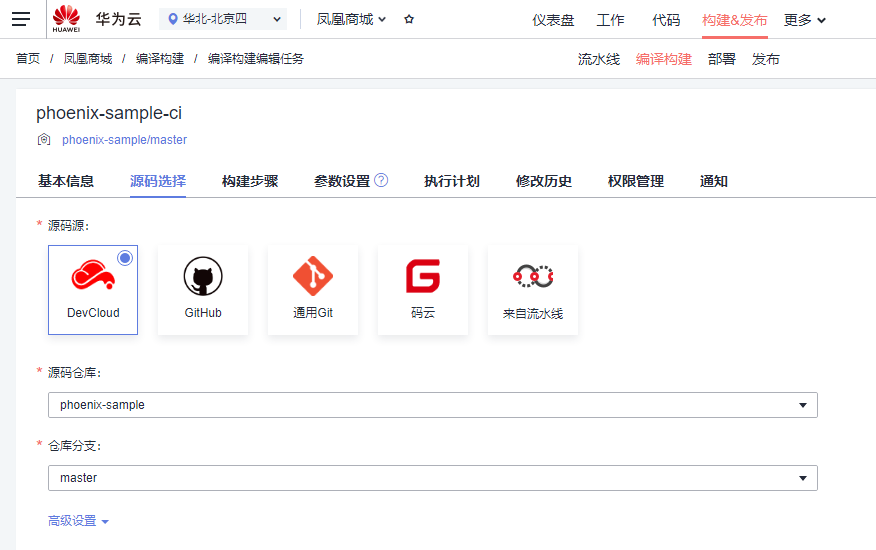
在可查看或编辑以下内容：

|  |  |
| --- | --- |
| 页签 | 说明 |
| 基本信息 | 编辑任务名称、描述。 |
| 源码选择 | 设置源码源、源码仓库、分支、子模组等信息。  分支内容可直接填写分支名称，如master；也可填写调用的分支参数名称，调用格式为${参数名称}。 |
| 构建步骤 | 配置构建步骤、步骤参数。 |
| 参数设置 | 配置执行任务时的自定义参数。 |
| 执行计划 | 配置构建任务的执行时间。 |
| 修改历史 | 查看任务的历史修改记录。 |
| 权限管理 | 配置不同项目角色的操作权限。 |
| 通知 | 配置各种事件类型的通知方式。 |

编译构建任务包含以下构建步骤（实际操作中保持默认设置即可）：

|  |  |
| --- | --- |
| 构建步骤 | 说明 |
| 制作Vote镜像并推送到SWR仓库 | 通过路径＂*工作目录*./vote＂及＂*Dockerfile路径*./Dockerfile＂找到Dockerfile文件，依据Dockerfile文件制作并推送Vote功能镜像。 |
| 制作Result镜像并推送到SWR仓库 | 通过路径＂工作目录./result＂及＂Dockerfile路径./Dockerfile＂找到Dockerfile文件，依据Dockerfile文件制作并推送Result功能镜像。 |
| 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 | 通过路径＂工作目录./worker＂及＂Dockerfile路径Dockerfile.j2＂找到Dockerfile文件，依据Dockerfile文件制作并推送Worker功能镜像。 |
| 生成Postgres and Redis Dockerfile | 通过Shell命令生成Postgres和Redis的Dockerfile文件。Redis和Postgres分别是缓存和数据库。 |
| 制作Postgres镜像并推送到SWR仓库 | 通过＂工作目录.＂及＂Dockerfile路径./Dockerfile-postgres＂找到Dockerfile文件，依据Dockerfile文件制作并推送Postgres镜像。 |
| 制作Redis镜像并推送到SWR仓库 | 通过＂工作目录.＂及＂Dockerfile路径./Dockerfile-redis＂找到Dockerfile文件，依据Dockerfile文件制作并推送Redis功能镜像。 |
| 替换Docker-Compose部署文件镜像版本 | 首先，将文件docker-compose-standalone.yml中的参数docker-server、docker-org、image-version依次用构建任务当中的参数dockerServer、dockerOrg、BuildNumber进行替换，以便后续进行ECS部署时可以拉取到正确的镜像。  然后，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。 |
| 替换Kubernates部署文件镜像版本 | 首先，将在目录kompose下，所有以deployment结尾的文件中，参数docker-server、docker-org、image-version依次用构建任务当中的参数dockerServer、dockerOrg、BuildNumber进行替换，以便后续进行CCE部署时可以拉取到正确的镜像。  然后，使用tar命令将整个kompose目录压缩为phoenix-sample-k8s.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。 |
| 上传Kubernates部署文件到软件发布库 | 将之前压缩好的phoenix-sample-k8s.tar.gz（构建包路径）上传到软件发布库中归档，包名命名为phoenix-sample-k8s，实现软件包的版本管理。 |
| 上传docker-compose部署文件到软件发布库 | 将之前压缩好的docker-stack.tar.gz（构建包路径）上传到软件发布库中归档，包名命名为docker-stack，实现软件包的版本管理。 |

点击＂源码选择＂栏，＂仓库分支＂选择＂master＂



配置SWR服务。项目应用镜像存放需要使用到华为云容器镜像服务 (SWR) ，因此需要首先配置SWR服务。点击＂构建步骤＂栏，找到＂华为云容器镜像服务＂并点击，



此时进入到＂容器镜像服务＂主页，点击立即使用；



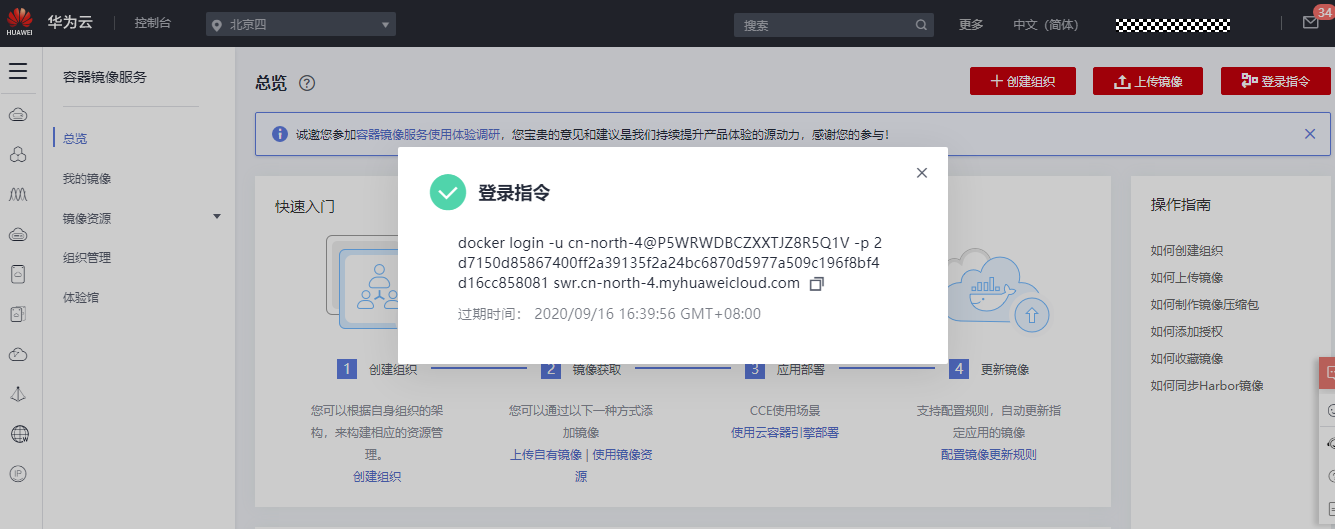
检查页面左上方的＂区域＂列表，选择与编译构建任务所在区相同的区域。

单击右上角＂登录指令＂，系统生成并弹框显示docker login指令。

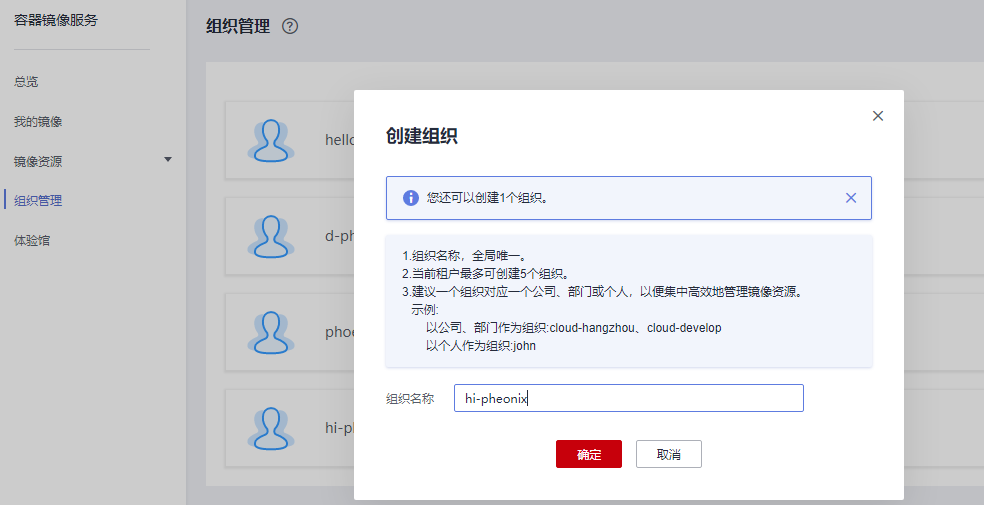
指令中，-u之后的字符串为用户名，-p之后的字符串为密码（此为临时用户名和密码，24小时刷新，隔天使用需重新查看用户名、密码；如需获取长期有效登录指令可参考文档https://support.huaweicloud.com/usermanual-swr/swr\_01\_1000.html）。

最后的字符串为SWR服务器地址，此地址即为后续步骤8中的参数dockerServer。

记录图中框出的部分。



在左侧导航栏中选择＂组织管理＂，进入＂组织管理＂页面。单击右上角＂创建组织＂，在弹框中输入组织名称＂hi-pheonix＂ (或自定义，此名称全局唯一)，单击＂确定＂保存。这里的组织名称，即为后续步骤7中的参数dockerOrg。



返回DevCloud页面，选择＂参数设置＂页签，编辑以下两个参数。

dockerOrg：输入在步骤6中创建的组织。

dockerServer：输入在步骤6中记录的内容。

请务必确保这参数dockerOrg、dockerServer的输入值是正确的。



单击＂保存并执行＂，启动编译构建任务。系统自动跳转至＂编译详情＂页面，可以查看任务执行进展。当出现以下页面时，表示任务执行成功。请记录＂构建成功＂字样前面的数字串（即任务的构建编号），该数字串为构建任务中的系统预定义参数BuildNumber，在后续步骤11查看镜像详情中，镜像版本即此数字串。



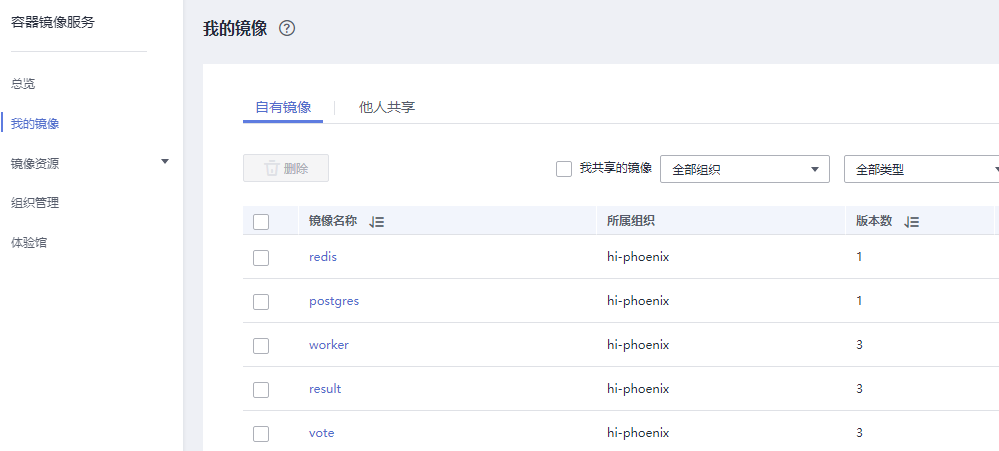
初次执行时构建时间较长，后续一般执行时长约为4-6分钟。

若构建失败，请仔细检查参数dockerOrg、dockerServer的设置是否正确。

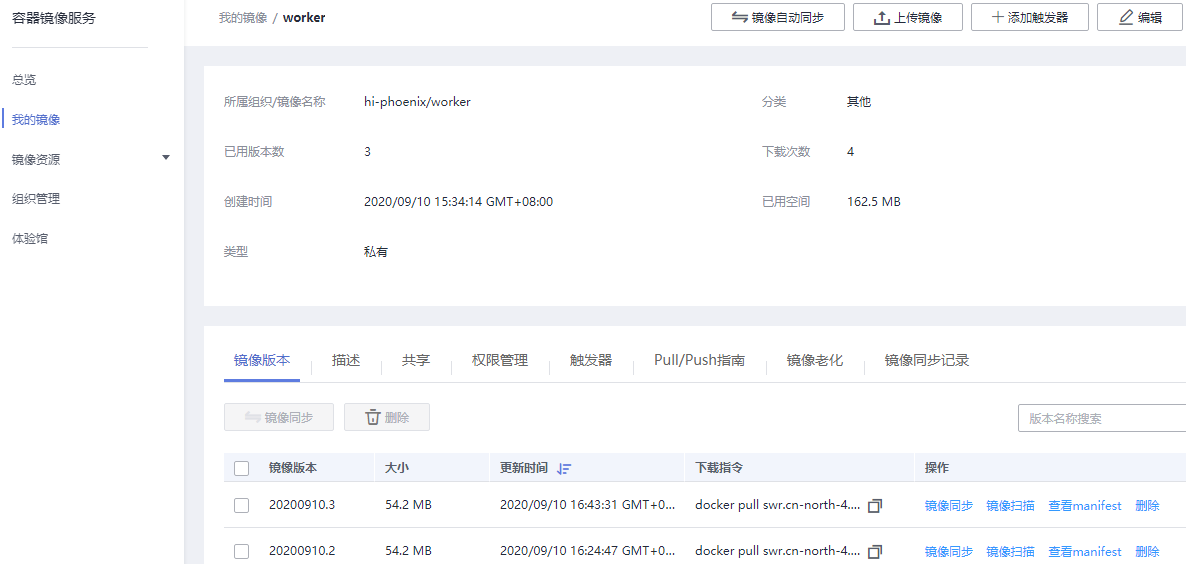
单击页面上方导航＂构建&发布 > 发布＂，进入发布服务。列表中可找到下图所示的两个文件夹。



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5个镜像，镜像的所属组织为实验开始创建的组织，版本数如下图所示。



在列表中单击＂worker＂查看镜像详情。在镜像列表页签中找到最新版本镜像，镜像版本与最新一次构建任务的执行编号一致，执行编号的查看方法请参考步骤8。



### 配置持续集成并通过提交代码触发自动化编译

通过本节，您将学习如何启动持续集成，实现代码变更后自动触发应用的Docker镜像打包以及推送。

持续集成配置。进入编译构建任务＂phoenix-sample-ci＂的＂编译构建编辑任务＂页面，选择＂执行计划＂页签，打开＂提交代码触发执行＂开关，单击＂保存＂。



设置完成后，当有代码变更时，将自动触发该任务。

持续集成执行计划配置。在＂执行计划＂页签，开启“定时执行”，根据需要设置定时执行计划，单击＂保存＂。



### 使用开源镜像站服务改进自动化编译

通过本节，您将学习如何使用开源镜像站服务提高依赖包获取速度，提高自动化编译效率。

背景

项目经理：Alex，凤凰商城的持续集成任务，每次执行时间都太长了，分析下什么原因？尽快反馈给我。

Alex：主要是我们凤凰商城每个服务中都有外部依赖包，包括（maven、npm、nuget、pip）等，这些服务编译过程中都要先下载依赖包，而且由于网络问题，甚至会下载失败导致持续集成失败！

项目经理：OK，那有没有什么好办法解决这个问题？

Alex：这些依赖包可以使用我们自己的镜像源，这样速度会就可以达到数量级的提升！

项目经理：太棒了，马上执行！

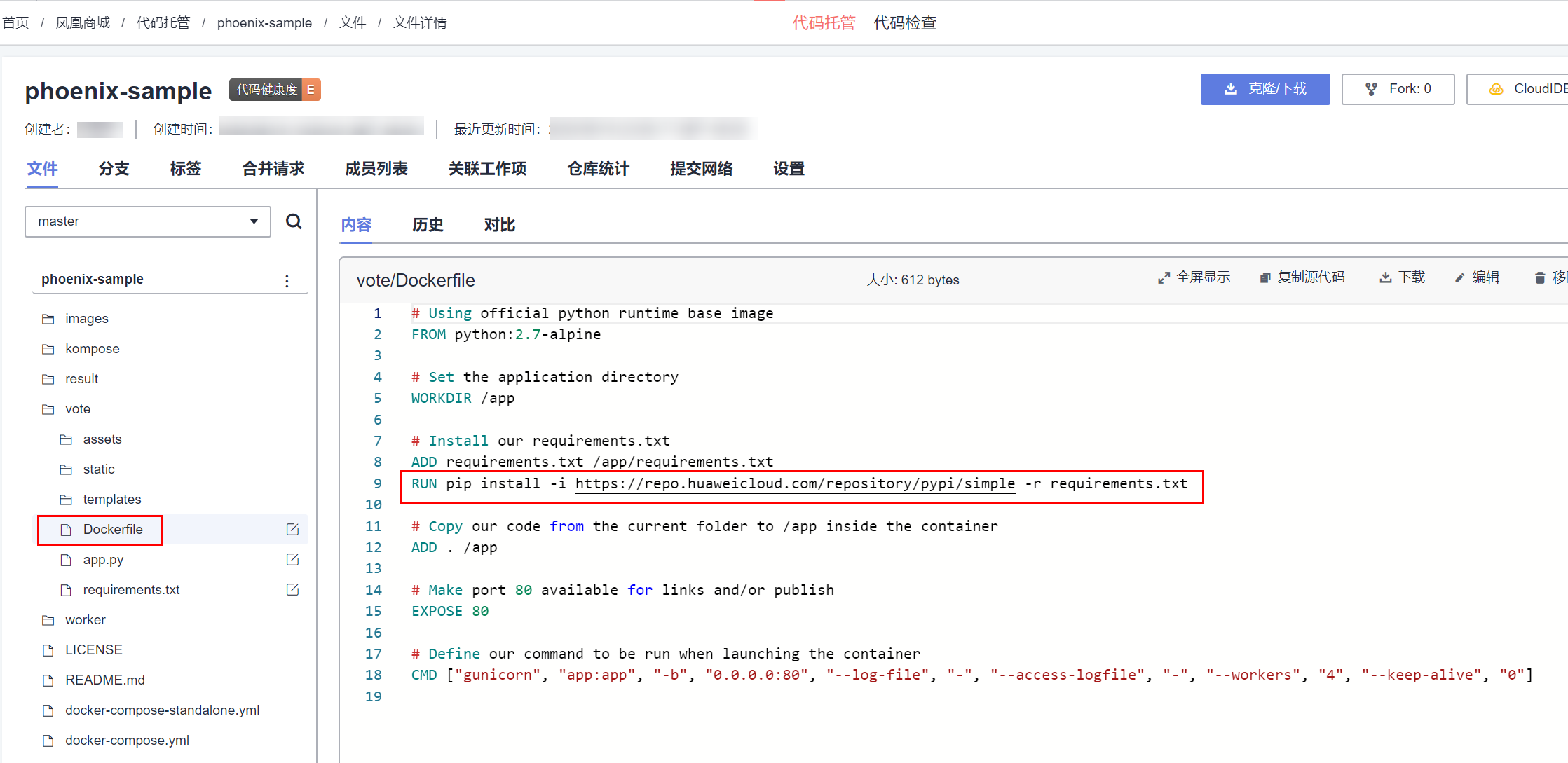
通过以下两种配置方式，可以在执行编译构建任务时自动获取开源镜像站中的镜像：

方式一：使用系统内置的Maven构建、Npm构建、PyInstaller构建步骤，系统会自动通过开源镜像站获取相应的依赖包。

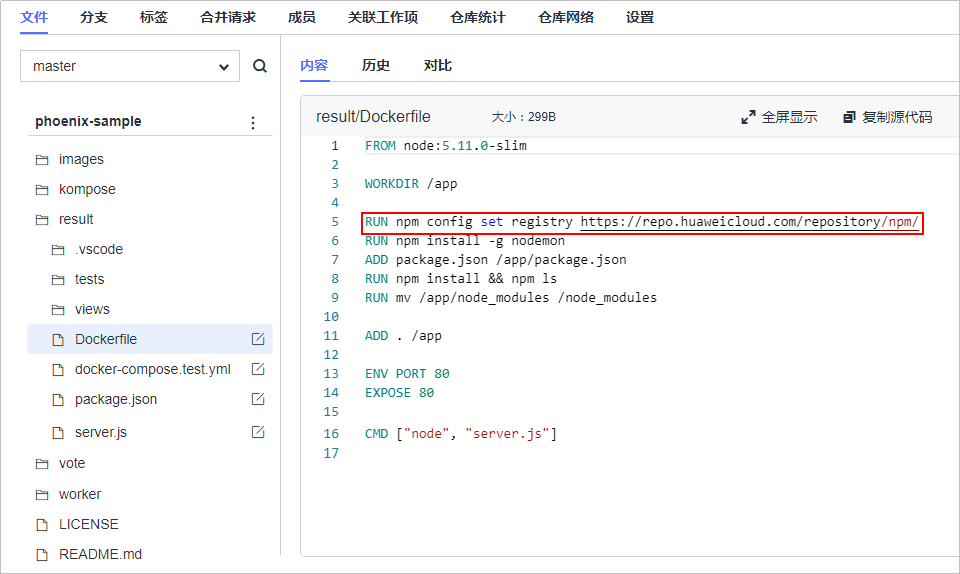


方式二：将镜像源设置为对应的开源镜像站仓库地址配置在代码仓库的配置文件中，查看方式＂代码>代码托管>phoenix-sample>文件＂。配置方式有以下三种：

Python配置方法如下图：



NPM配置方法如下图：



Nuget配置方法如下图所示：

