**中国矿业大学计算机学院**

**2020 级本科生实验报告**

课程名称 软件工程实践

报告时间 2022-11-1

学生姓名 李建业

学 号 12203743

专 业 计算机科学与技术

任课教师 刘迎春

**实验三 集成编程环境与编码规范**

**一、实验过程和截图**

（绘制每一个数据流图，说明其名称、编号、基本逻辑）代码托管

### 实验介绍

在本实验中，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

### 使用CloudIDE修改和提交代码

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”。

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＂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。

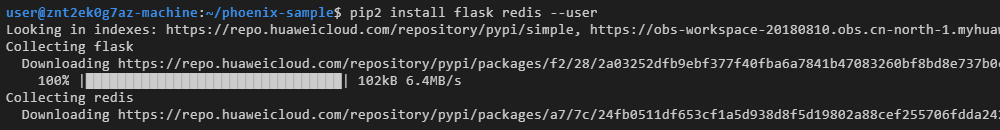
点击左下角选择“Python 3.7.x”版本解释器

修改Dockerfile

打开vote下Dockerfile文件，修改“FROM python:2.7-alpine”为“python:3.7-alpine”。

终端窗口安装两个运行必须模块：

pip install flask redis --user





替还原有Python2.7.x版本代码

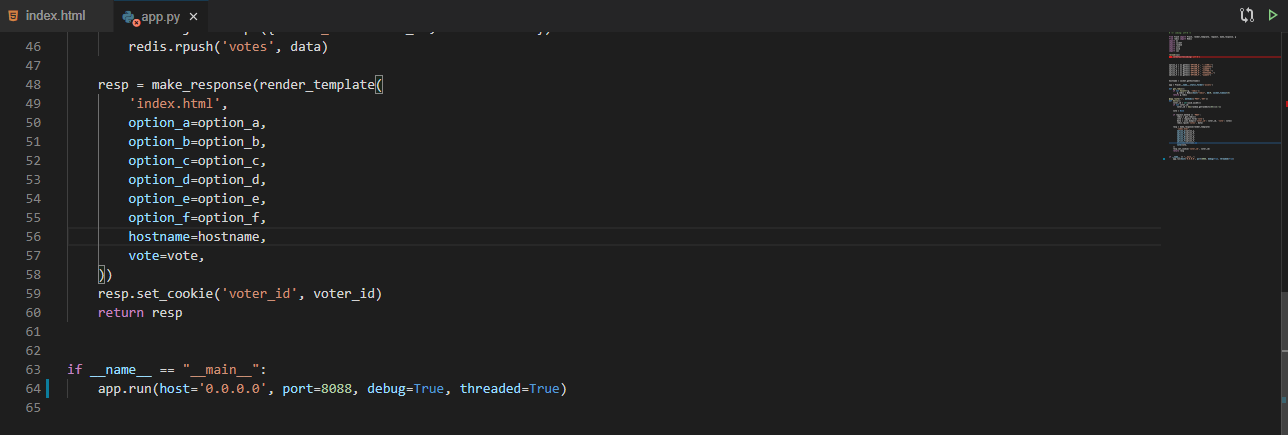
导入import importlib

修改reload(sys)为importlib.reload(sys)

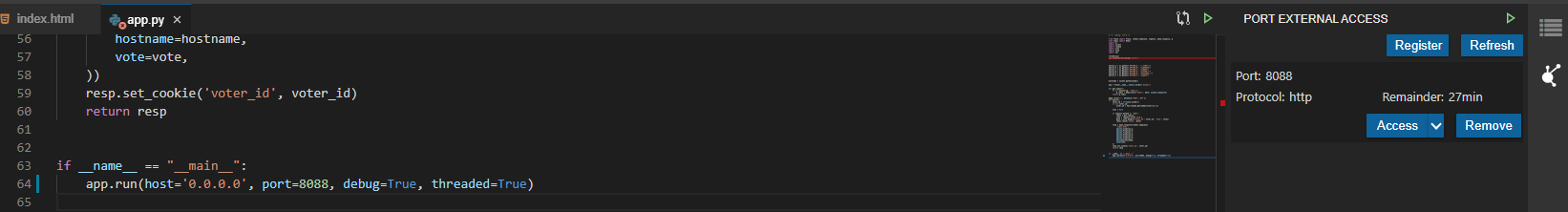
删除sys.setdefaultencoding('utf-8')

运行代码

安装完成后将app.py中第64行代码中的端口号改为8088（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Access，点击Access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。



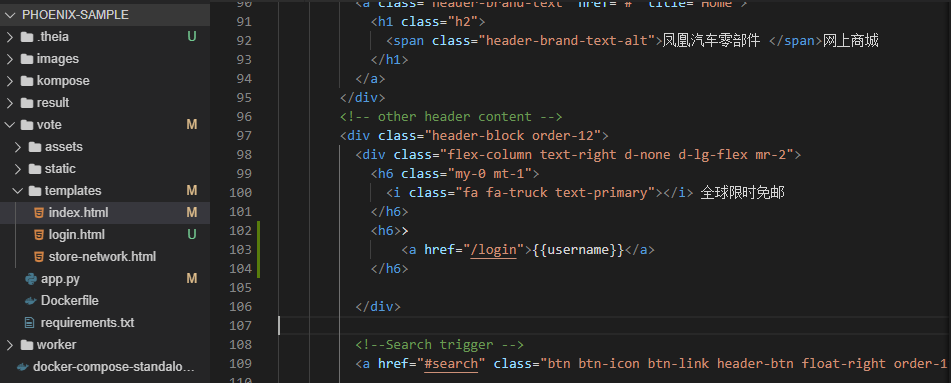


如果程序代码报错，可使用CloudIDE中的Debug进行调试。

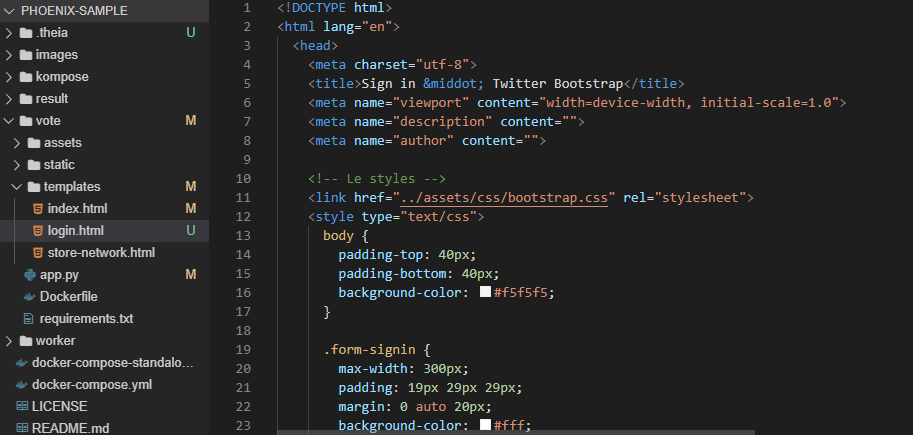
程序运行预览成功以后，需要把app.py源程序代码第64行的端口改回80端口（docker容器内外的端口映射）。

### 添加登录功能。

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

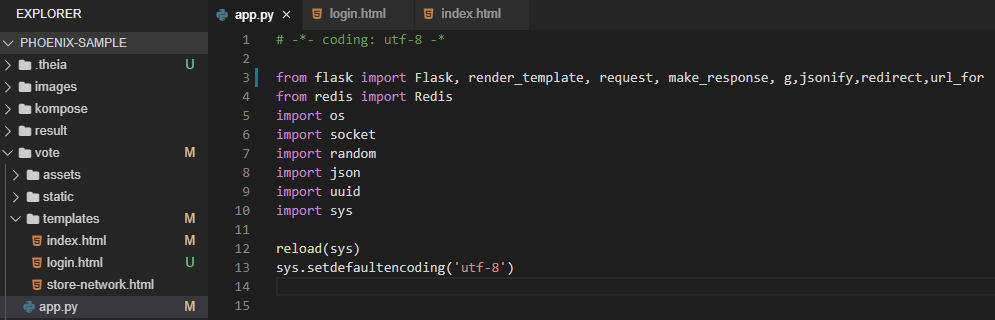


在/vote/template目录下新建登录界面模板login.html



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

from flask import Flask, render\_template, request, make\_response, g,jsonify,redirect,url\_for



在/vote/app.py文件的47行添加如下代码。

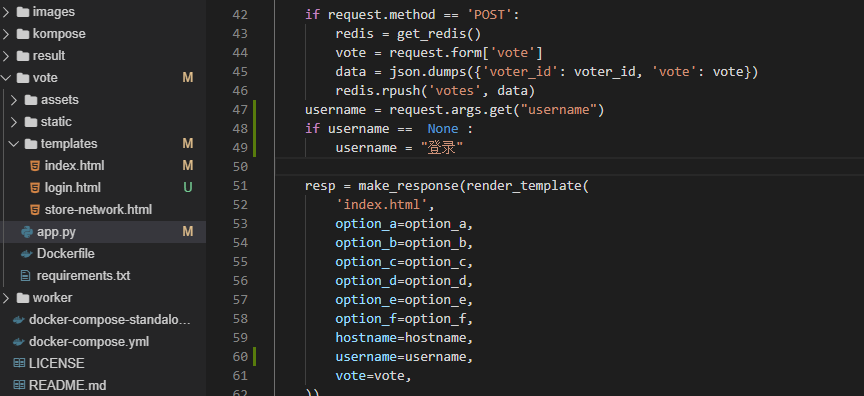
username = request.args.get("username")

if username == None :

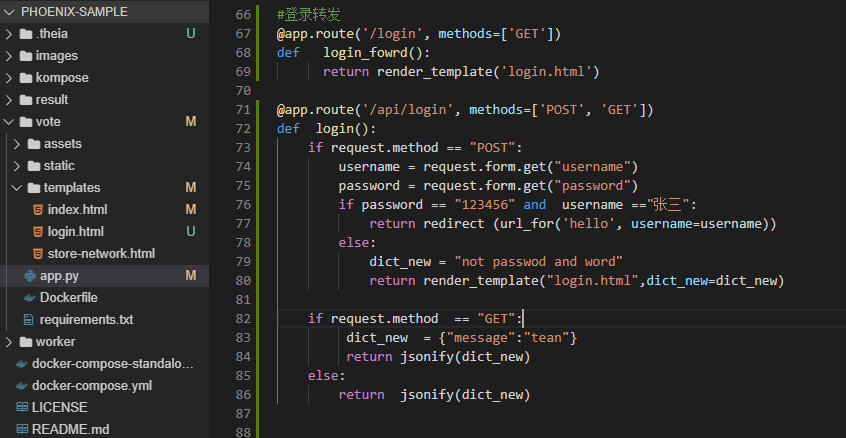
username = "登录"

在60行添加

username=username,



在/vote/app.py文件的 main入口前添加如下代码。



调试登录功能并推送至代码仓库。



登录界面





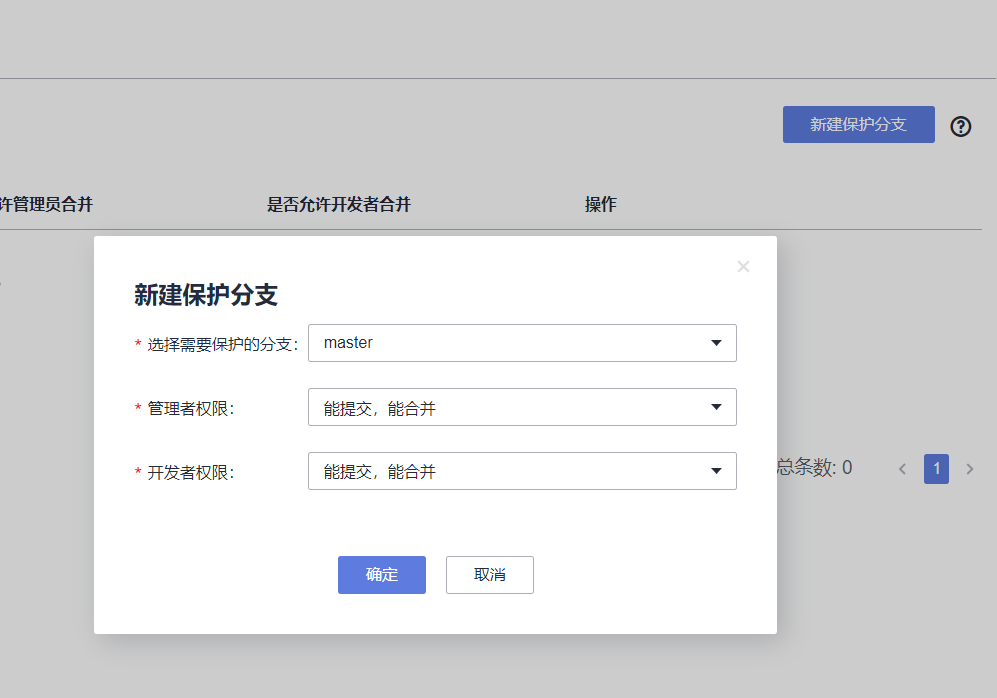
参考2.1.3 使用CloudIDE修改和提交代码-步骤4操作，将修改的代码推送至代码仓库。

### 使用Git分支+合并请求方式提交代码并进行代码检视

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。

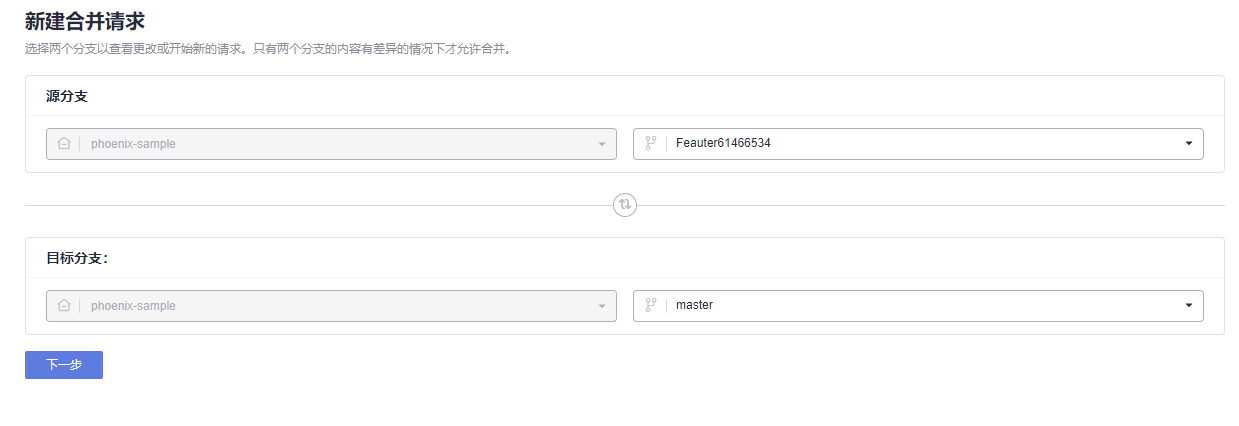


用CloudIDE打开代码仓库，选择新创建的分支进入（如果是已打开的CloudIDE界面，可单击页面左下角分支名称，在页面上方的下拉列表中选择刚刚创建分支。可以看到左下角从＂master＂变成了新建分支名）。

### 合并请求（登录模块）

新增登录功能为例，使用合并请求（如遇其它合并参照此处即可）。

创建合并请求，进入代”码托管，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＂页签中发表评审意见，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＂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＂。

## 代码检查确保编码规范

### 实验介绍

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

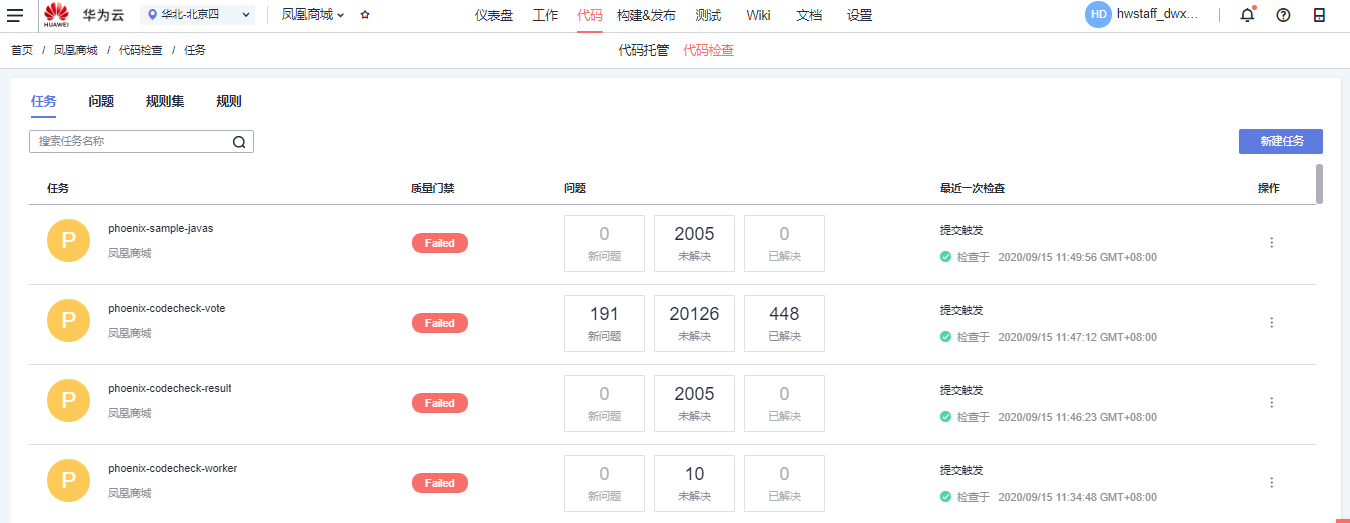
理解并掌握代码检查的任务配置功能

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

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。

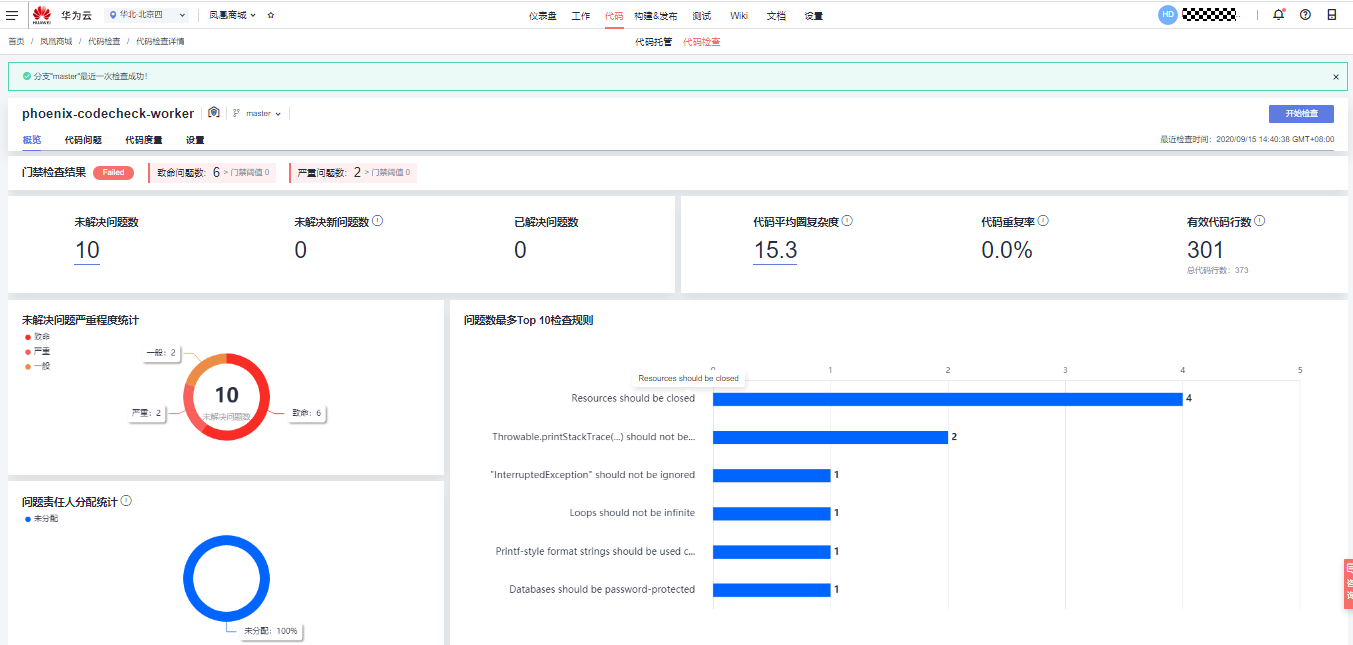


选择＂设置＂页签，在页面左侧导航中单击＂规则集＂。规则集中默认包含的语言是＂Java＂。单击图标重新获取代码仓库语言，在刷新的列表中将Python语言对应的开关打开。

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

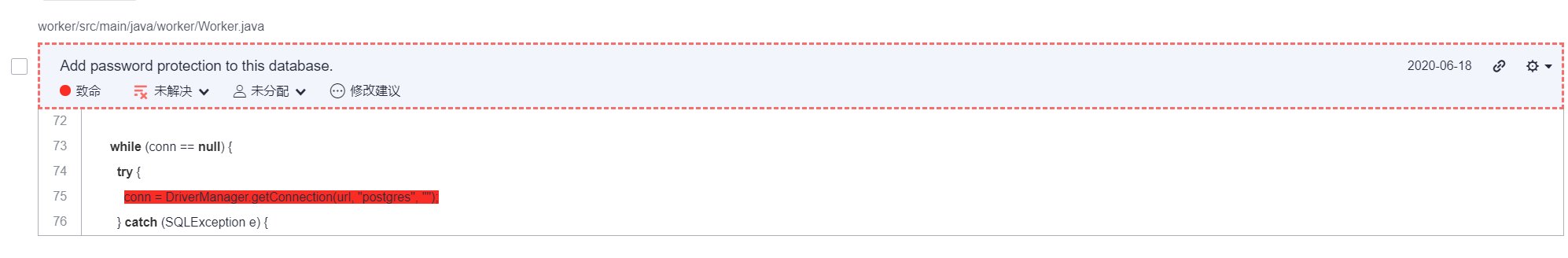
分析代码检查报表。进入代码检查任务＂phoenix-codecheck-worker＂的＂代码检查详情＂页面。成功执行代码检查任务，查看检查结果。



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。

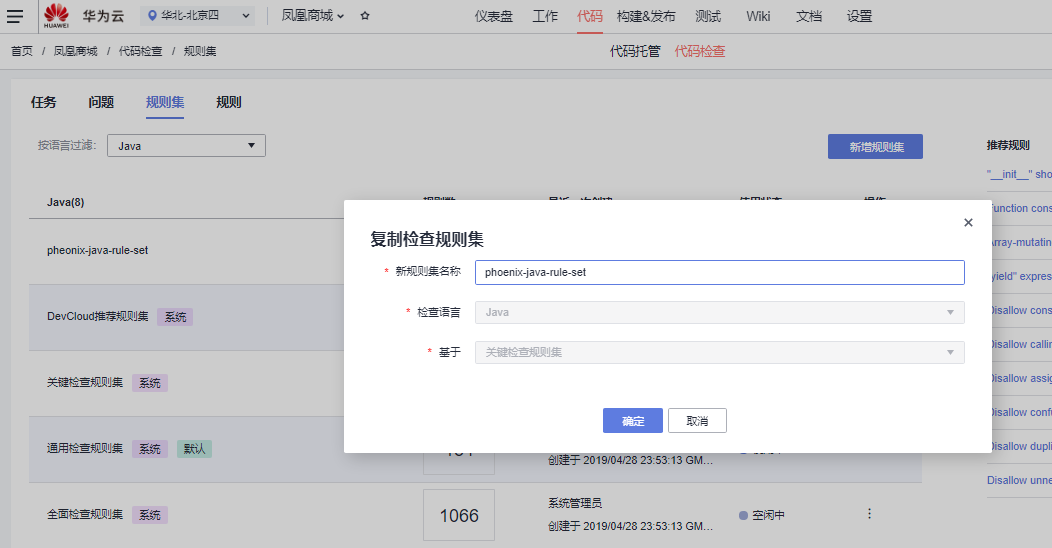


单击问题卡片，页面右侧将滑出窗口。单击＂修改建议＂，可查看系统对此问题的修改建议。单击窗口上方的图片，可进入相应的代码文件修改代码。单击＂未解决＂，可将此问题的状态设置为＂已解决＂或＂已忽略＂。

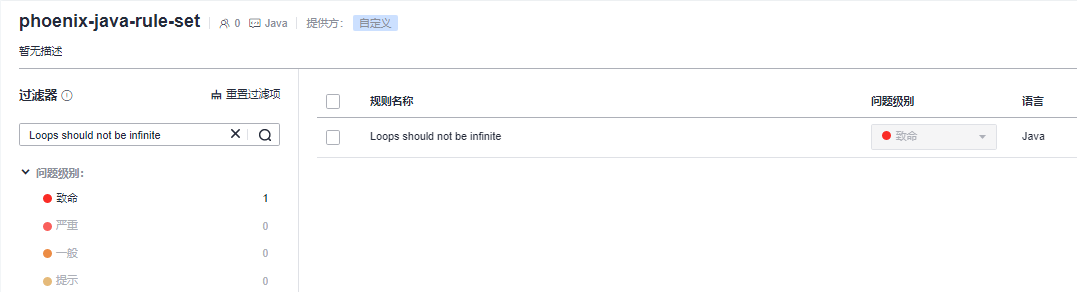


### 配置和管理代码检查规则集

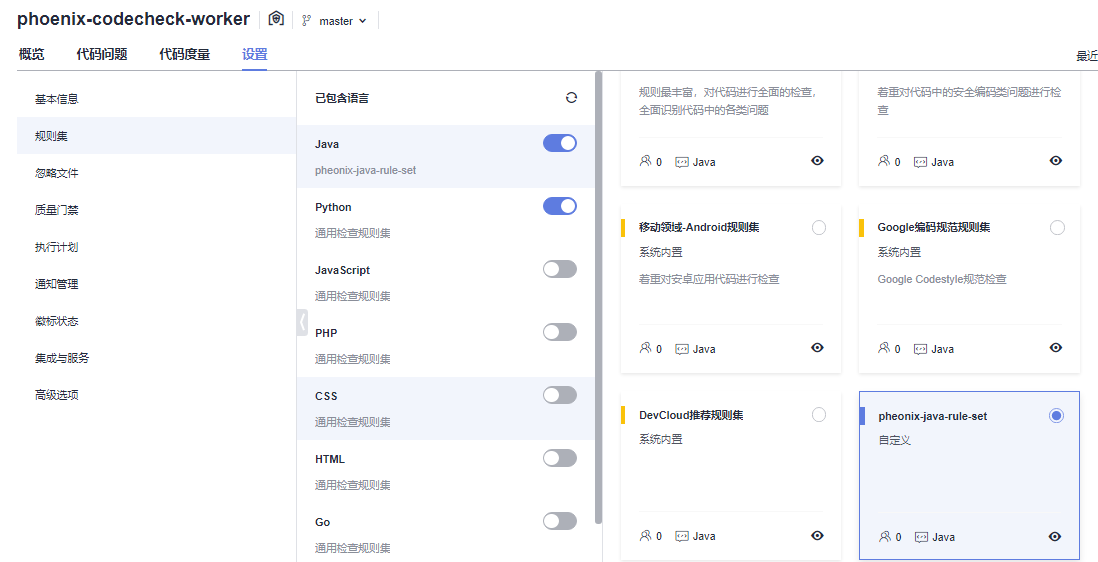
创建自定义检查规范。进入代码检查服务，选择＂规则集＂页签，在下拉列表中选择＂Java＂。在过滤出的列表中找到规则集＂关键检查规则集＂，单击图标 ，在下拉列表中选择＂复制＂。在弹框中输入新规则集名称为＂phoenix-java-rule-set＂，单击＂确定＂保存。



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。



PHP项目检测

  
实验选用经费管理系统的登录部分文件进行php语言验证，使用php规则集



进行php检测





最后已通过华为云的代码检查。

Java项目检测

设置java规则集



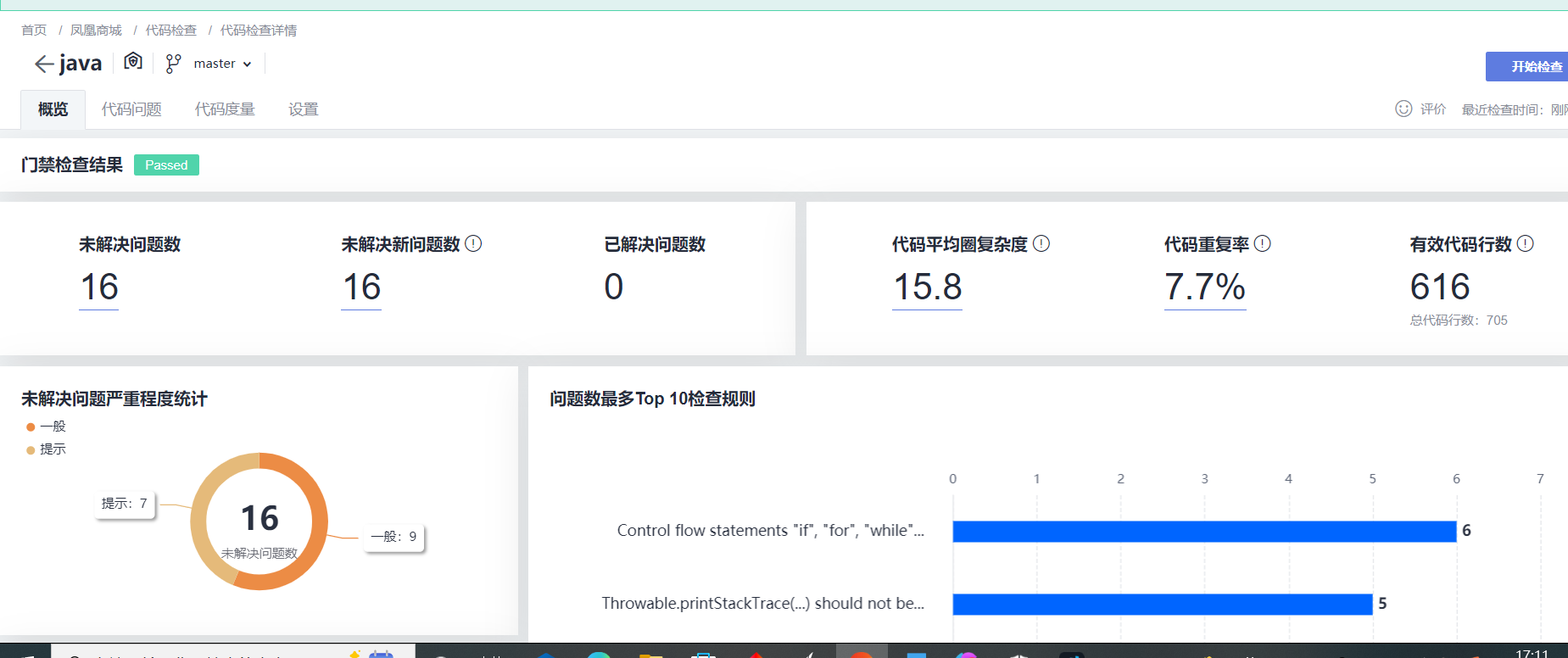
对master分支下的java项目进行检测



检测出问题

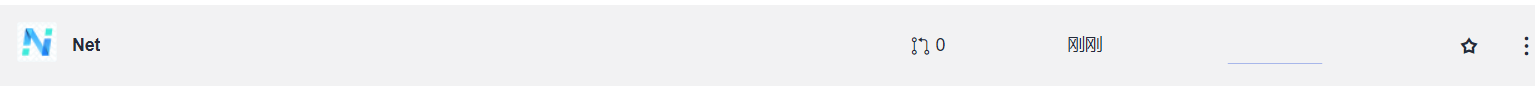


修改后



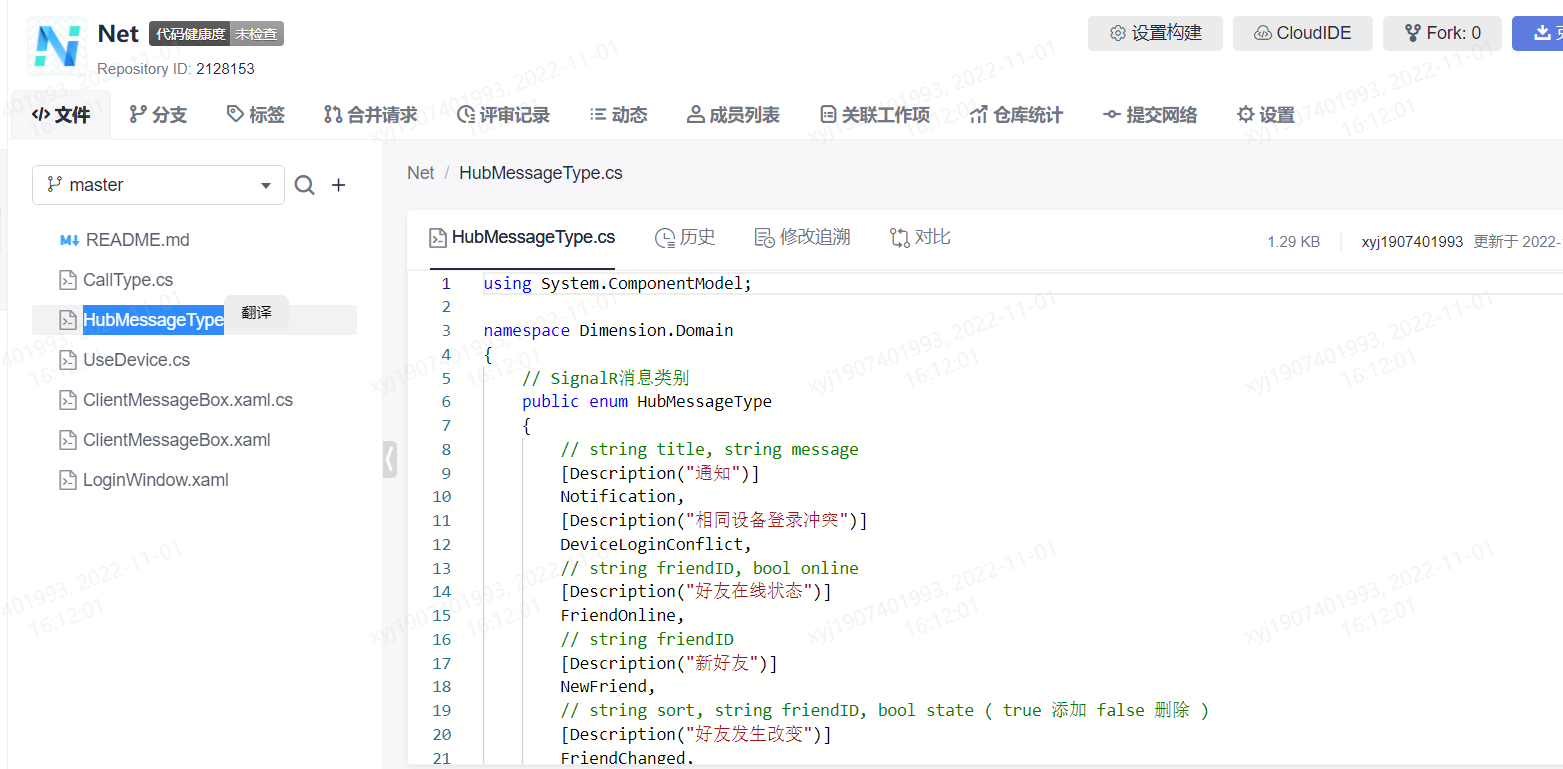
C++，C#和net框架的检测

新建仓库



上传构建Net聊天项目





进行项目代码检查



代码修改

再监测



配置和管理代码检查规则集



使用自动化编译

## 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，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

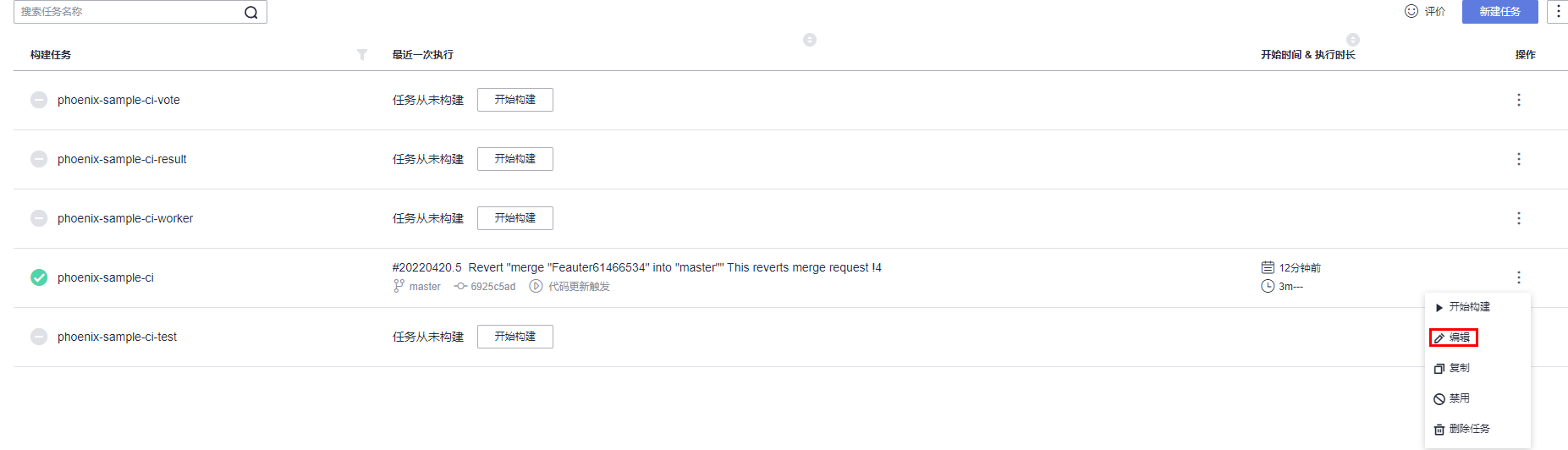
掌握华为云＂开源镜像站＂的使用方法

### 编译构建并完成应用的镜像打包推送

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＂。

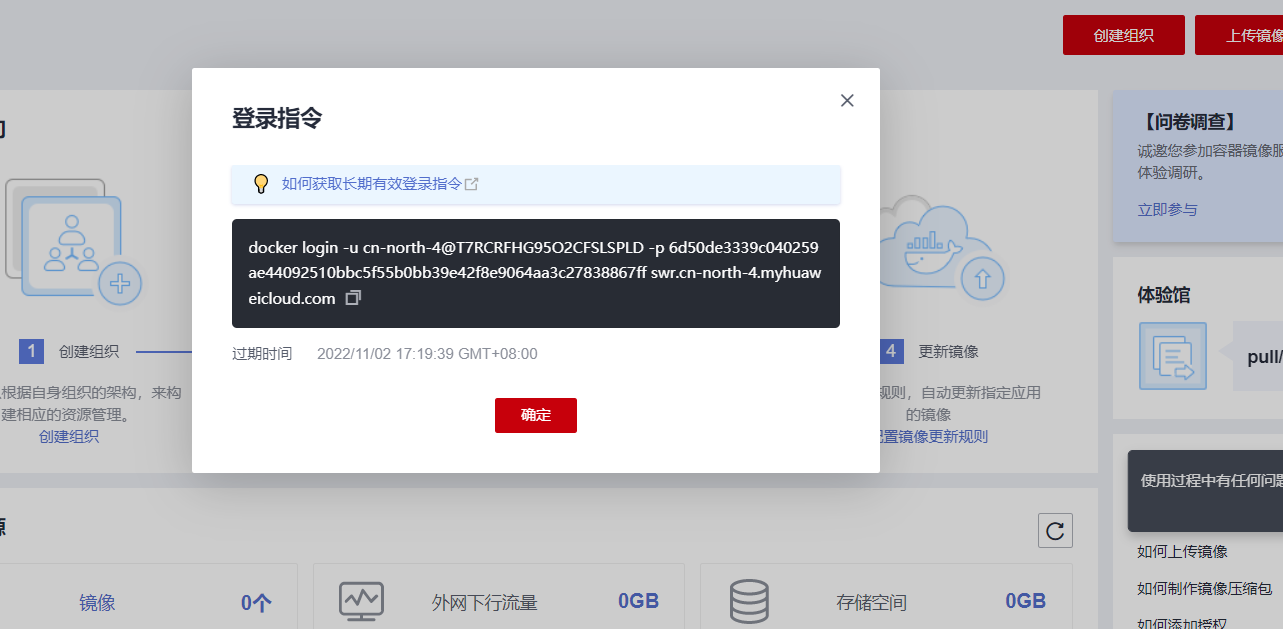


单击＂phoenix-sample-ci＂右侧，进入＂编辑＂页面。

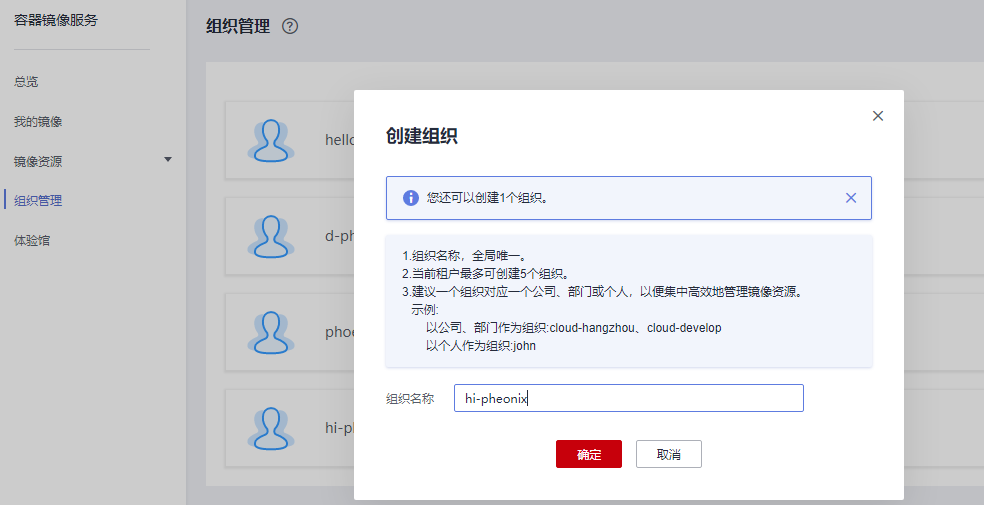


配置SWR服务。项目应用镜像存放需要使用到华为云容器镜像服务 (SWR) ，因此需要首先配置SWR服务。点击＂构建步骤＂栏，找到＂华为云容器镜像服务＂并点击，





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5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4中创建的组织。

dockerServer：输入在步骤4中记录的内容。

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

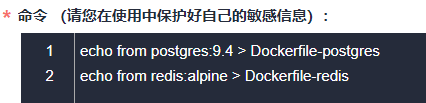


修改镜像为swr容器上的镜像地址

制作swr镜像redis

制作swr镜像postgrepsql

替换前

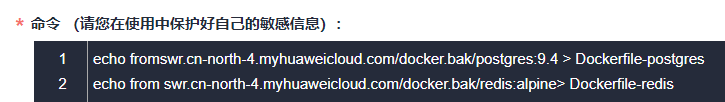


替换后

命令

echo from swr.cn-north-4.myhuaweicloud.com/docker.bak/postgres:9.4 > Dockerfile-postgres

echo from swr.cn-north-4.myhuaweicloud.com/docker.bak/redis:alpine > Dockerfile-redis



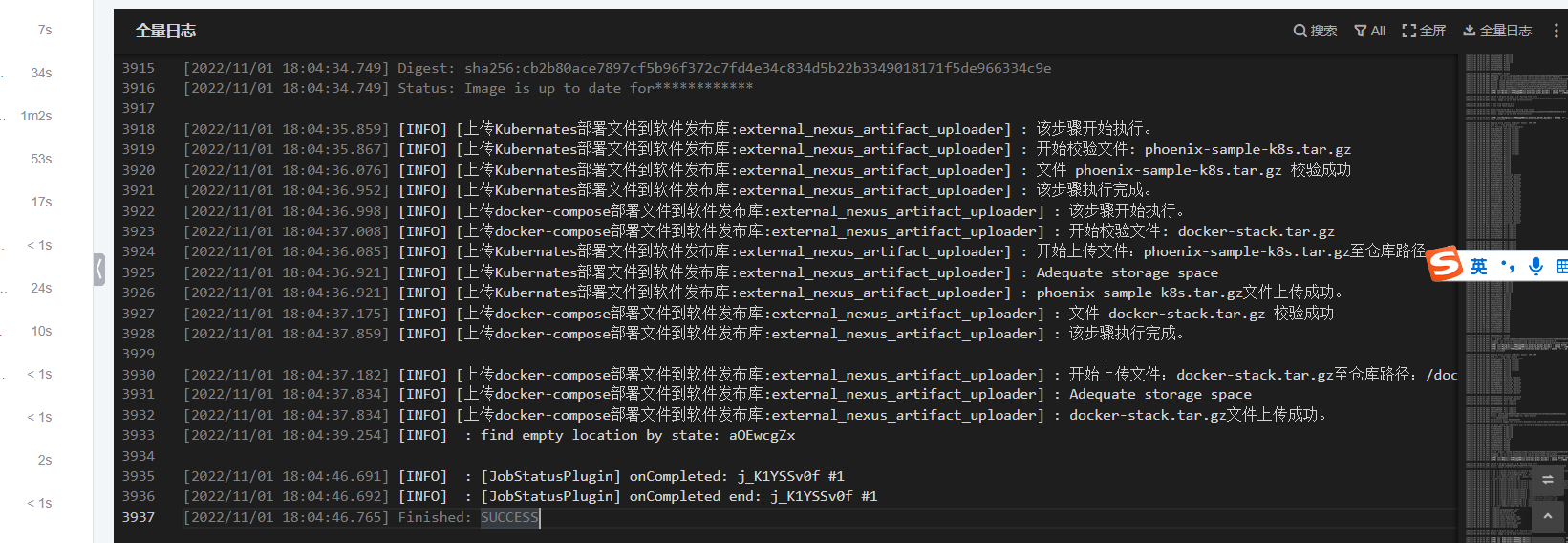
注意：

1. 替换不是必须的，除非报错。
2. 地址替换为

swr.cn-north-4.myhuaweicloud.com/docker.bak/postgres:9.4

swr.cn-north-4.myhuaweicloud.com/docker.bak/redis:alpine必须保证SWR有docker.bak组织且该组织含有postgres:9.4 redis:alpine镜像。

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10查看镜像详情中，镜像版本即此数字串。



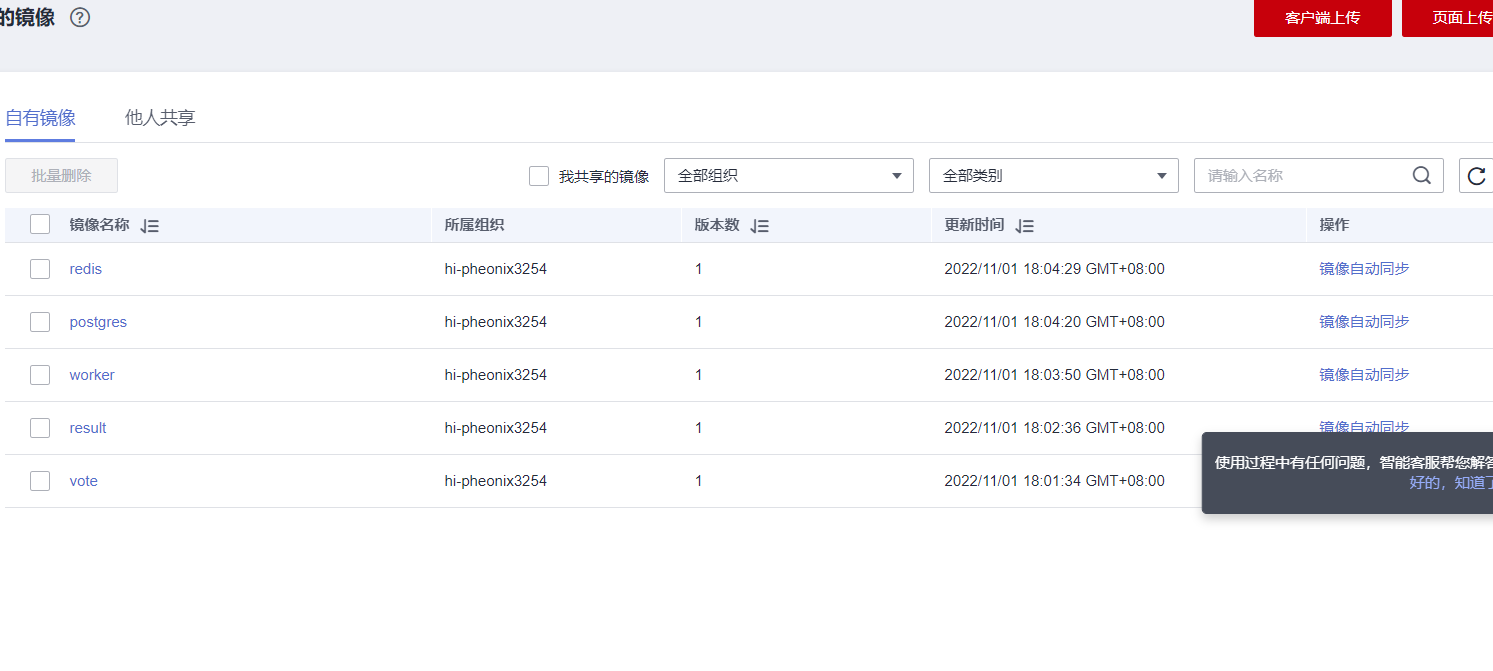
初次执行时构建时间较长，后续一般执行时长约为4-6分钟。

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7。



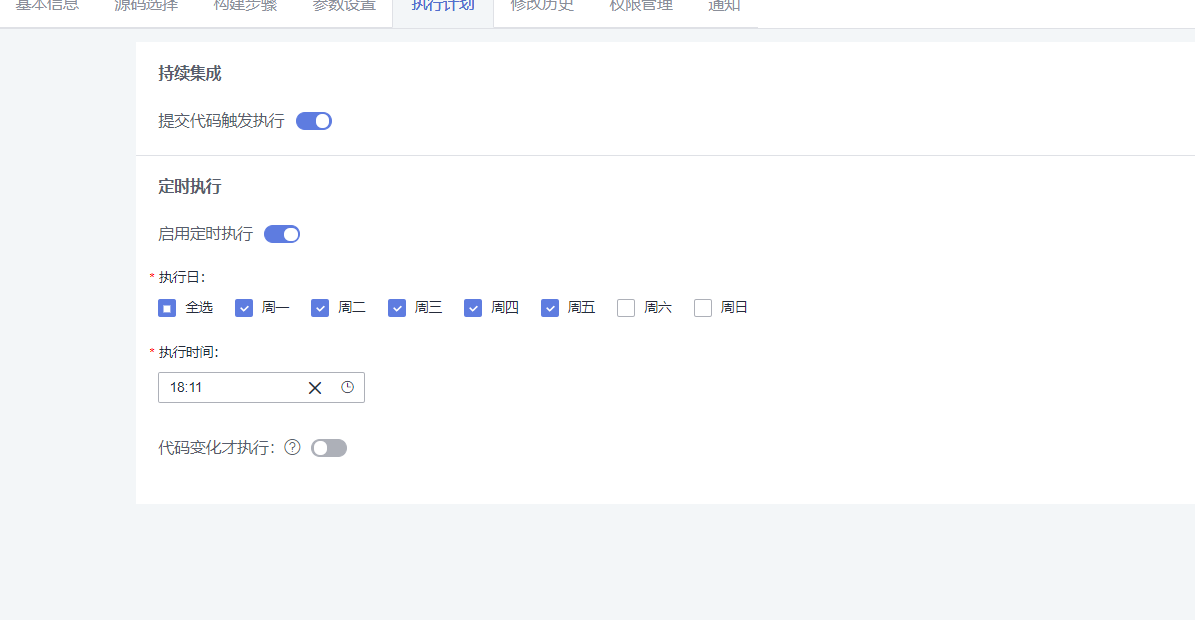
### 配置持续集成并通过提交代码触发自动化编译

持续集成配置。进入编译构建任务＂phoenix-sample-ci＂的＂编辑＂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

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：

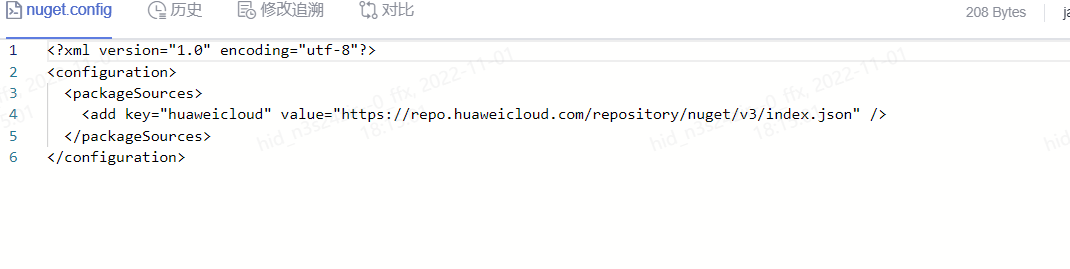
Python配置方法如下图：



NPM配置方法如下图：



Nuget配置方法如下图所示：



二、实验体会

通过本次学习了解到如何对 ＂凤凰商城＂ 进行基本信息等维护，理解并掌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。