哈尔滨工业大学 计算学部

2024 年秋季学期《开源软件开发实践》

Lab4：开源软件开发中的 DevOps

学号	姓名	联系方式
2022211830	徐耀	15318051170

目 录

1 实验要求

本次实验训练开源软件开发中的基本 DevOps 操作，具体来说：

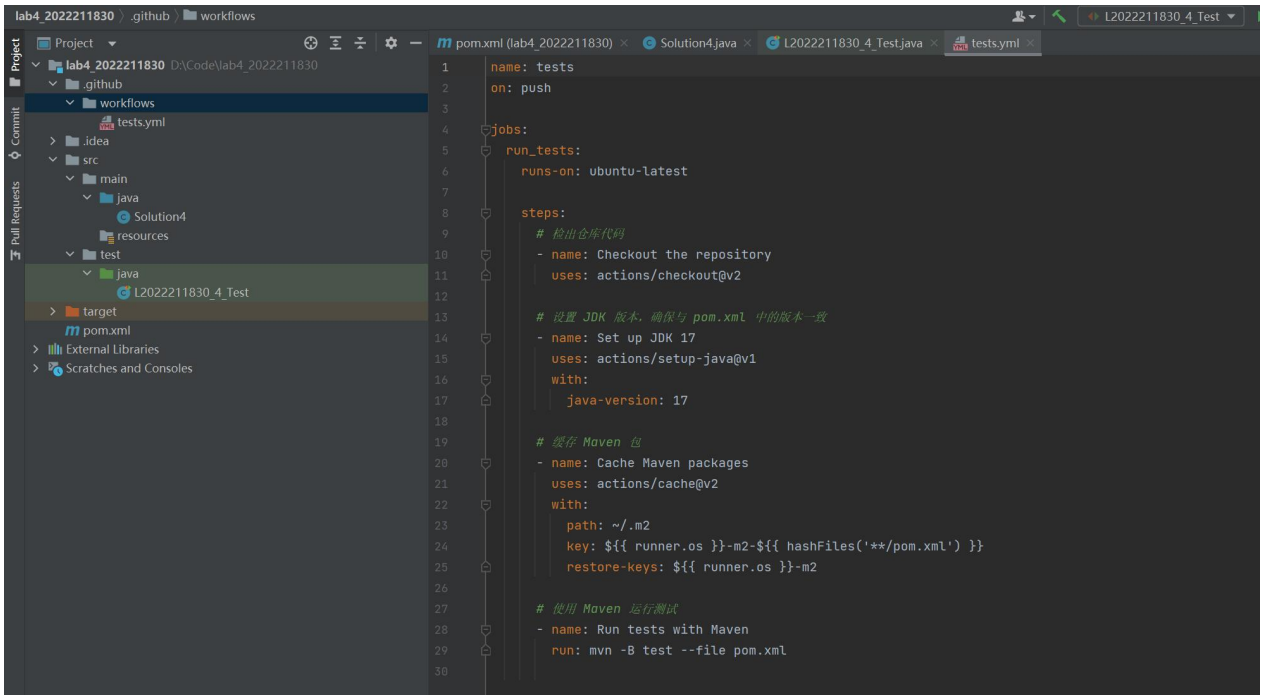
- 掌握开源软件开发中的基本 DevOps 流程和工具的使用
- 熟悉利用 Github Actions 进行 DevOps
- 熟悉利用 Jenkins 进行 DevOps

2 实验内容 1 Github Actions DevOps 实践

1.在本地创建一个 maven 项目，将 Lab2 的程序和测试文件导入后提交至仓库。

```
Administrator@CHINAMI-O7ABOOG MINGW64 /d/Code/lab4_2022211830 (master)
$ git push -u origin master
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 20 threads
Compressing objects: 100% (17/17), done.
Writing objects: 100% (21/21), 5.72 KiB | 2.86 MiB/s, done.
Total 21 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/OSSDP/Lab4-2022211830/pull/new/master
remote:
To github.com:OSSDP/Lab4-2022211830.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

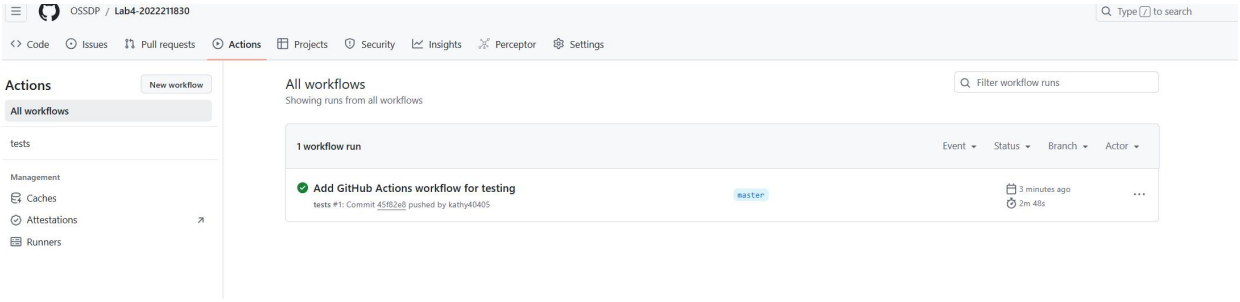
2.编写 workflow 文件，编写后项目目录结构如下：



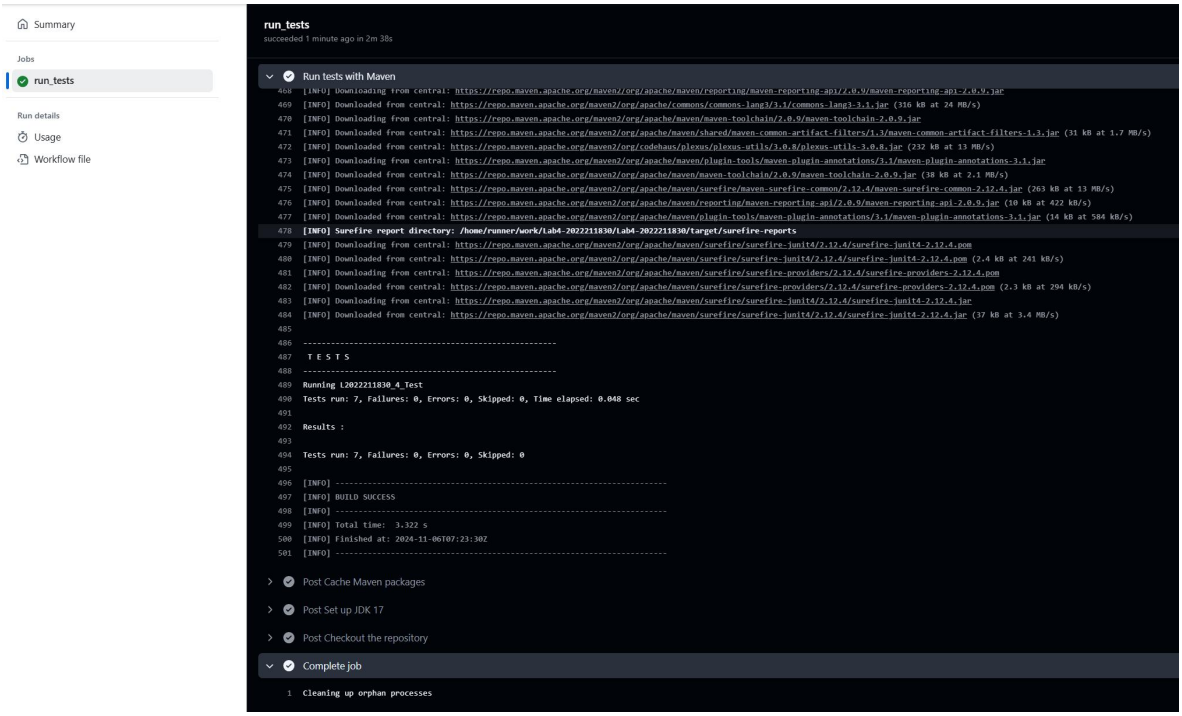
3.将工作流文件提交至 GitHub 仓库

```
Administrator@CHINAMI-O7ABOOG MINGW64 /d/Code/lab4_2022211830 (master)
$ git add .github/workflows/tests.yml

Administrator@CHINAMI-O7ABOOG MINGW64 /d/Code/lab4_2022211830 (master)
$ git commit -m "Add GitHub Actions workflow for testing"
[master 45f82e8] Add GitHub Actions workflow for testing
 2 files changed, 35 insertions(+)
 create mode 100644 .github/workflows/tests.yml
 create mode 100644 .idea/vcs.xml

Administrator@CHINAMI-O7ABOOG MINGW64 /d/Code/lab4_2022211830 (master)
$ git push origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 20 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 1.06 KiB | 1.06 MiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:OSSDP/Lab4-2022211830.git
   f6f995b..45f82e8  master -> master
```

提交后发现 GitHub 会进行自动化测试：



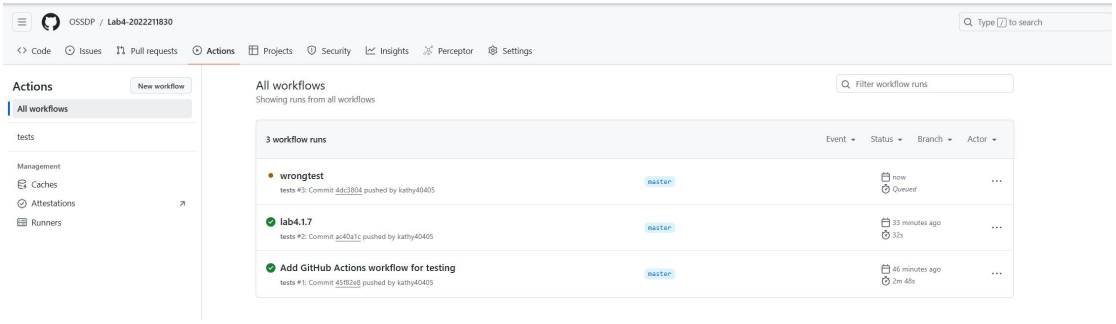
查看具体的执行流程：

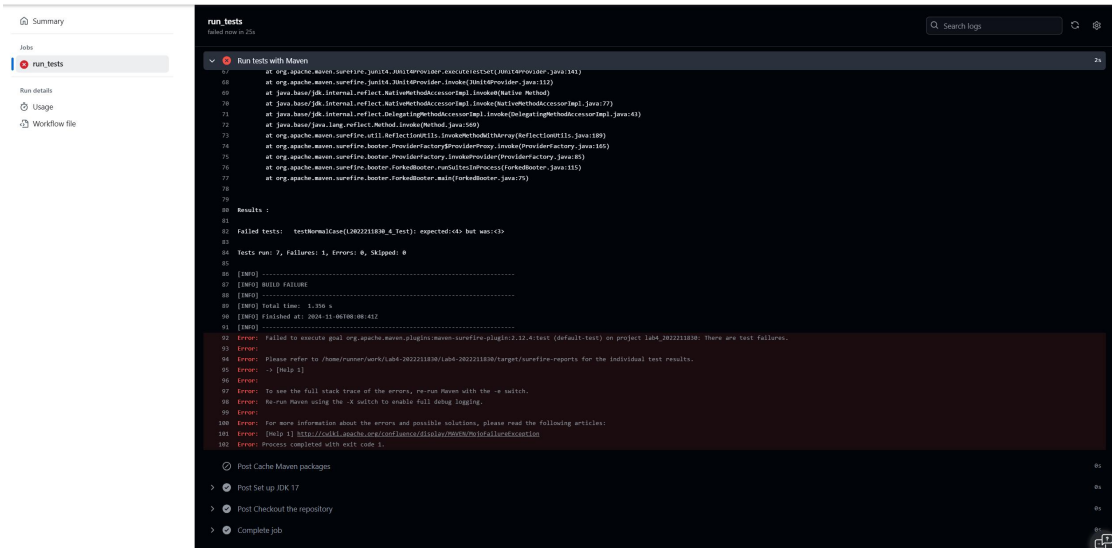


4.修改测试用例为错误的结果后再次提交

```
$ git add .
warning: in the working copy of '.idea/compiler.xml', LF will be replaced by CRL
F the next time Git touches it

Administrator@CHINAMI-O7ABOOG MINGW64 /d/Code/lab4_2022211830 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   src/test/java/L2022211830_4_Test.java
        new file:   target/test-classes/wrongtest.class


Administrator@CHINAMI-O7ABOOG MINGW64 /d/Code/lab4_2022211830 (master)
$ git commit -m "wrongtest"
[master 4dc3804] wrongtest
 2 files changed, 1 insertion(+), 1 deletion(-)
 create mode 100644 target/test-classes/wrongtest.class

Administrator@CHINAMI-O7ABOOG MINGW64 /d/Code/lab4_2022211830 (master)
$ git pusha origin master
git: 'pusha' is not a git command. See 'git --help'.

The most similar command is
        push

Administrator@CHINAMI-O7ABOOG MINGW64 /d/Code/lab4_2022211830 (master)
$ git push origin master
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 20 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 1.10 KiB | 565.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:OSSDP/Lab4-2022211830.git
   ac40a1c..4dc3804  master -> master

Administrator@CHINAMI-O7ABOOG MINGW64 /d/Code/lab4_2022211830 (master)
```

发现错误提交未通过自动化检测：

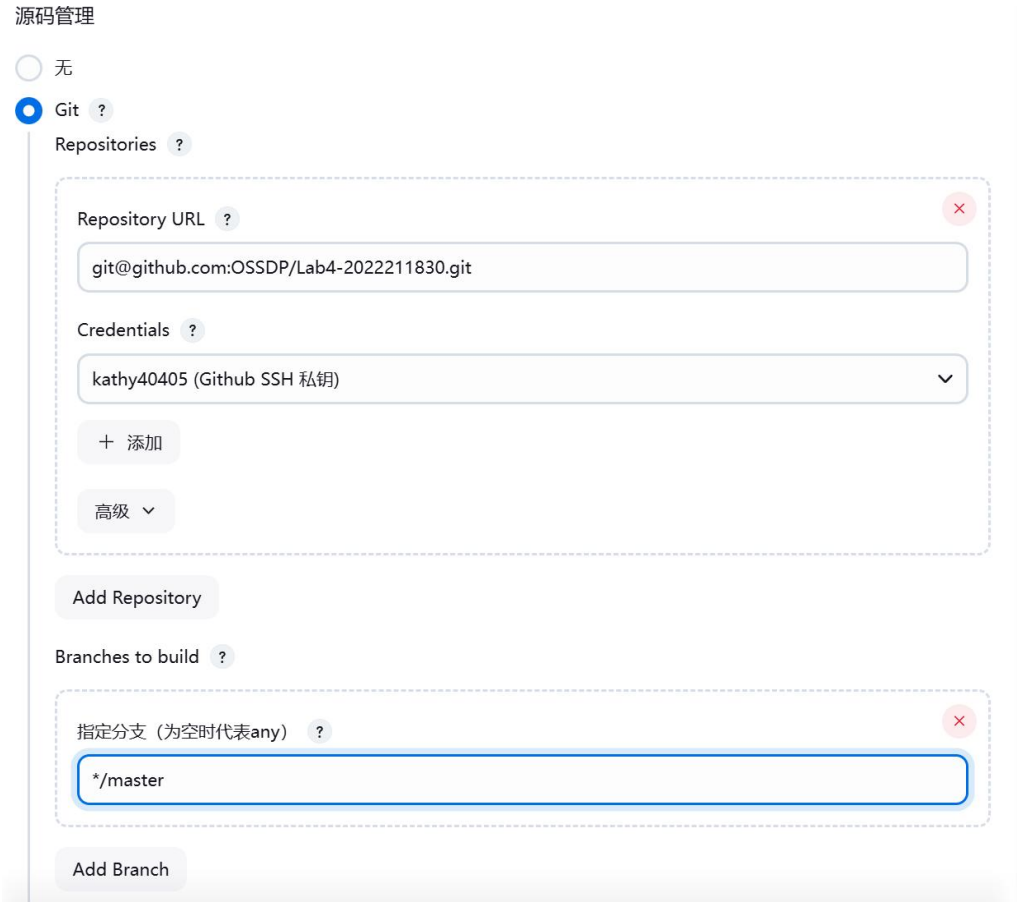




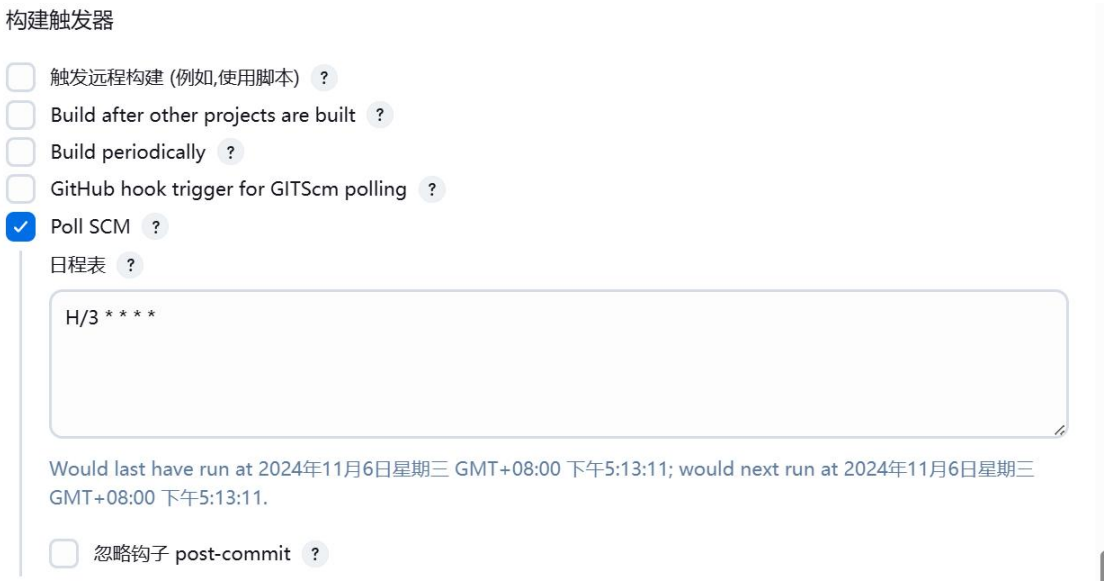
3 实验内容 2 Jenkins DevOps 实践

1. **按照实验指导书内容安装 Jenkins 与 GitHub CLI**
2. **构建 workflow**

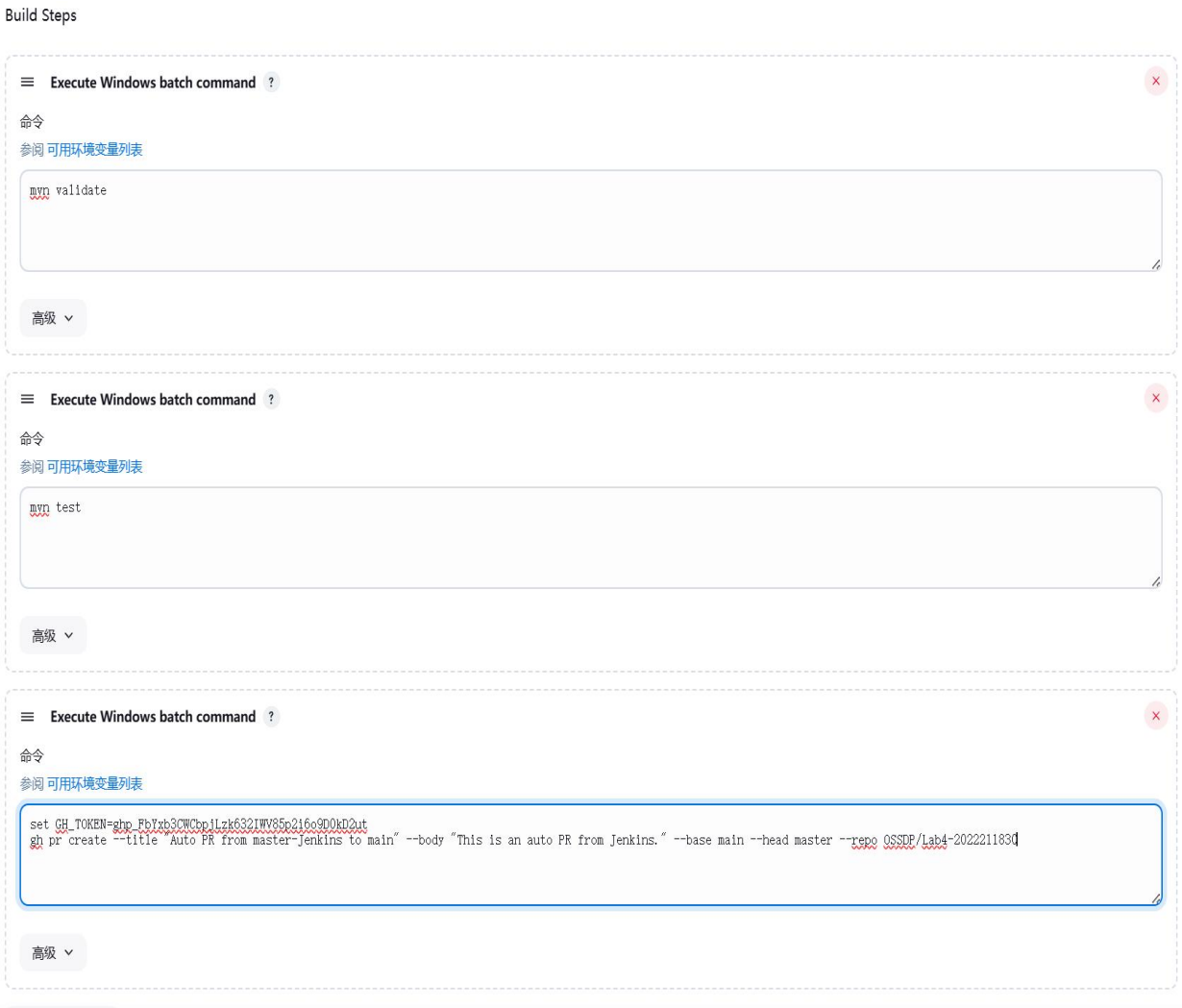
   在源码管理部分对 GitHub 仓库进行访问配置：



配置触发器：

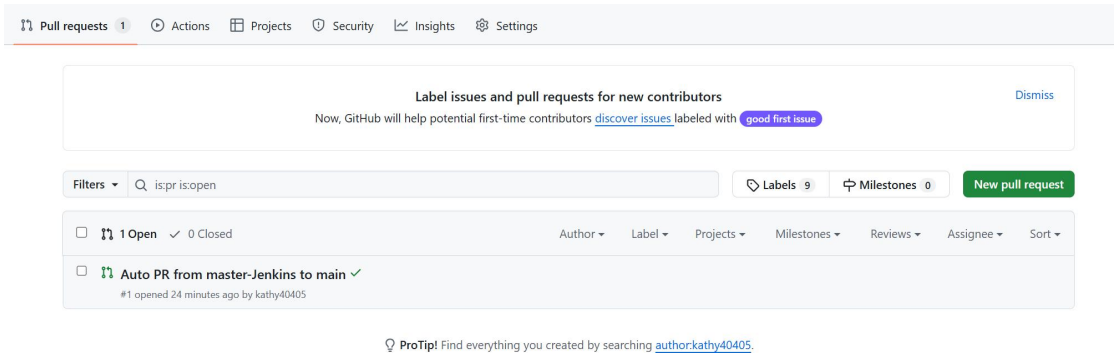

构建触发器
触发远程构建 (例如,使用脚本)
Build after other projects are built
Build periodically
GitHub hook trigger for GITScm polling
Poll SCM
日程表
H/3 * * * *
Would last have run at 2024年11月6日星期三 GMT+08:00 下午5:13:11; would next run at 2024年11月6日星期三 GMT+08:00 下午5:13:11.
忽略钩子 post-commit

构建步骤：


Build Steps
Execute Windows batch command
命令
参阅 可用环境变量列表
mvn validate
高级
Execute Windows batch command
命令
参阅 可用环境变量列表
mvn test
高级
Execute Windows batch command
命令
参阅 可用环境变量列表
set GH_TOKEN=ghp_FbYxb3CWCbp1Lzk632IWY85p2i6o9D0kD2ut
gh pr create --title "Auto PR from master-Jenkins to main" --body "This is an auto PR from Jenkins." --base main --head master --repo OSSDP/Lab4-2022211830
高级

3.提交项目代码，等待定时任务进行构建流程。查看结果成功

在 GitHub 上查看仓库 PR 推送成功：



4 小结

本次实验让我更深入地理解了 DevOps 的流程及工具应用，尤其是在开源项目开发中的实践。从 Github Actions 的自动化测试与工作流配置，到 Jenkins 的自动化构建流程，每一步都帮助我强化了对 DevOps 自动化的理解和实践能力。在使用 Github Actions 过程中，我学会了编写 YML 文件来自动运行测试，进一步掌握了基于事件触发的自动化测试的实现。这种自动化的流程提高了开发效率和代码质量控制的效果，有助于在开发中及时发现和修复错误。

在 Jenkins 部分的学习中，我体验了使用 Jenkins 进行代码构建与发布的过程。通过设置自动触发和定时构建，使我理解了如何将构建、测试、部署流程无缝衔接。遇到的一些细节问题，例如 Github 权限设置和 Jenkins 的 SSH 凭证配置，也让我更熟悉了配置的具体步骤与相关工具的兼容性问题。实验中的挑战是如何协调 Github Actions

与 Jenkins 的任务，确保自动化操作的流畅进行。

总的来说，这次实验不仅加深了我对 DevOps 工具的使用理解，也让我意识到自动化流程在软件开发中的重要性。通过反复的实验和调整，我逐渐掌握了在开源环境中进行持续集成和持续部署的技能，为今后的项目实践打下了坚实的基础。