

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
2022211876	万谔岐	13664371086

## 目 录

1	实验要求.....	1
2	实验内容 1 Github Actions DevOps 实践 .....	1
3	实验内容 2 Jenkins DevOps 实践.....	5
4	小结.....	7

[文档全部完成之后，请更新上述区域]

## 1 实验要求

目标:

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。

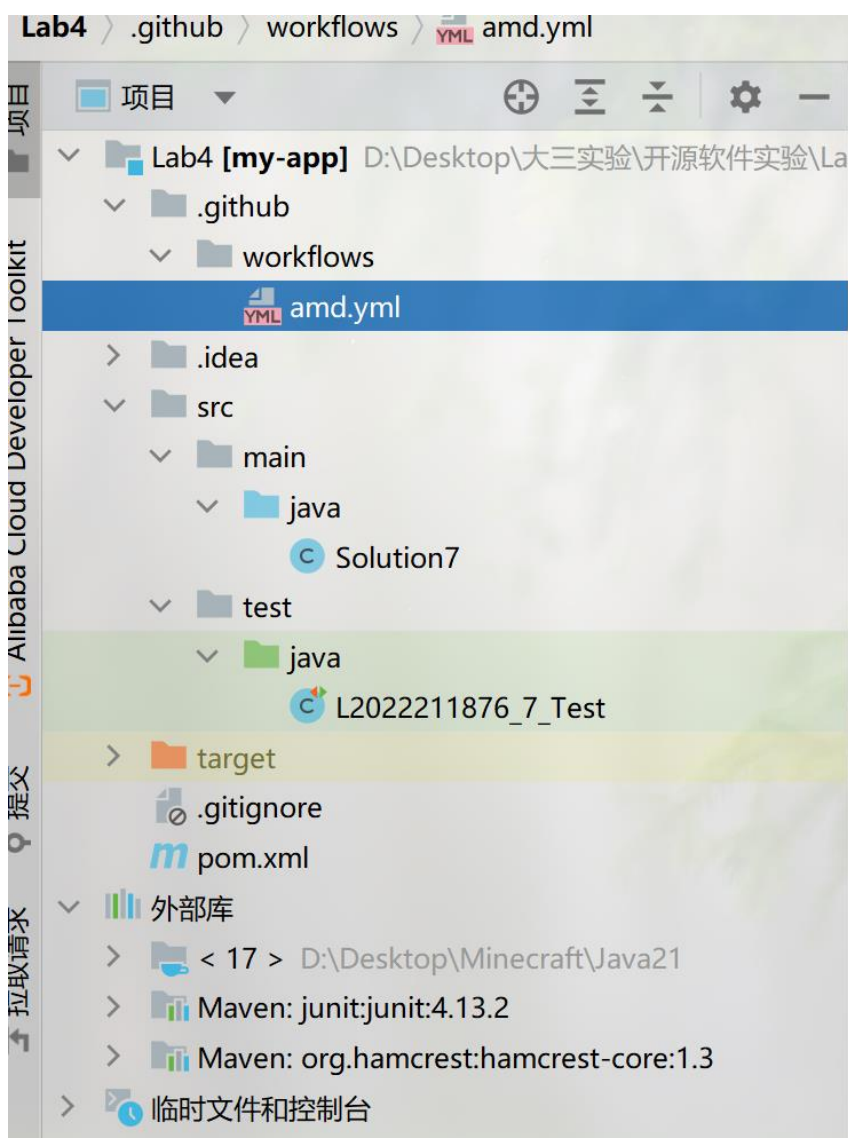
过程:

1、进行 Github Actions DevOps 实践

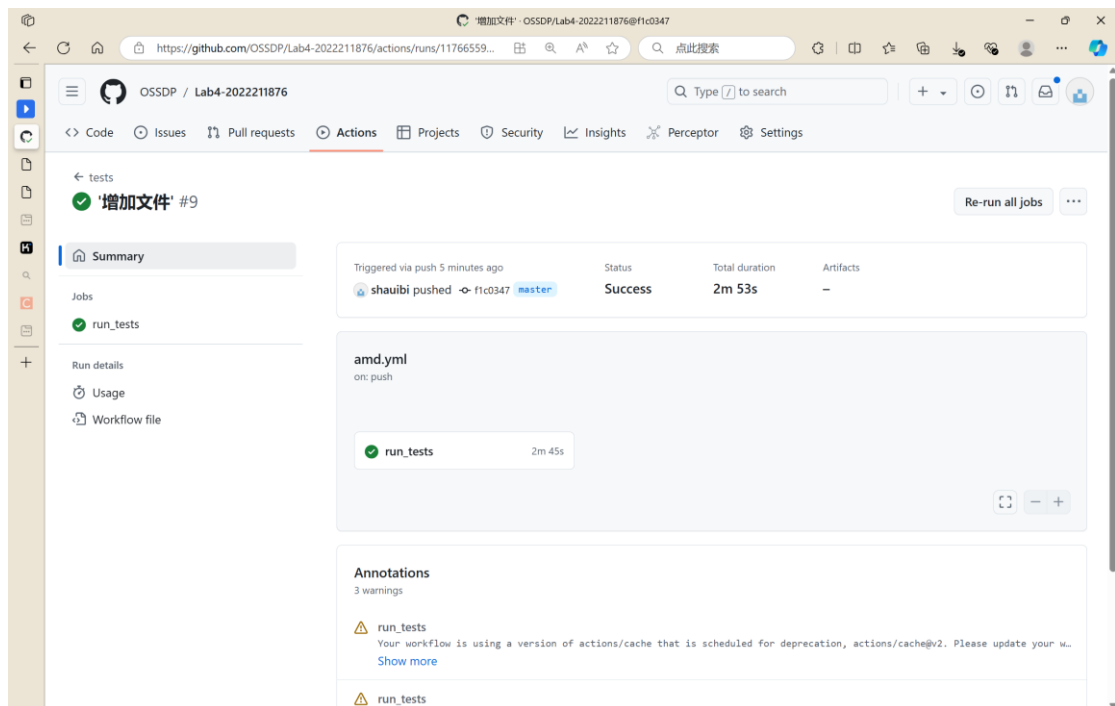
2、进行 Jenkins DevOps 实践

## 2 实验内容 1 Github Actions DevOps 实践

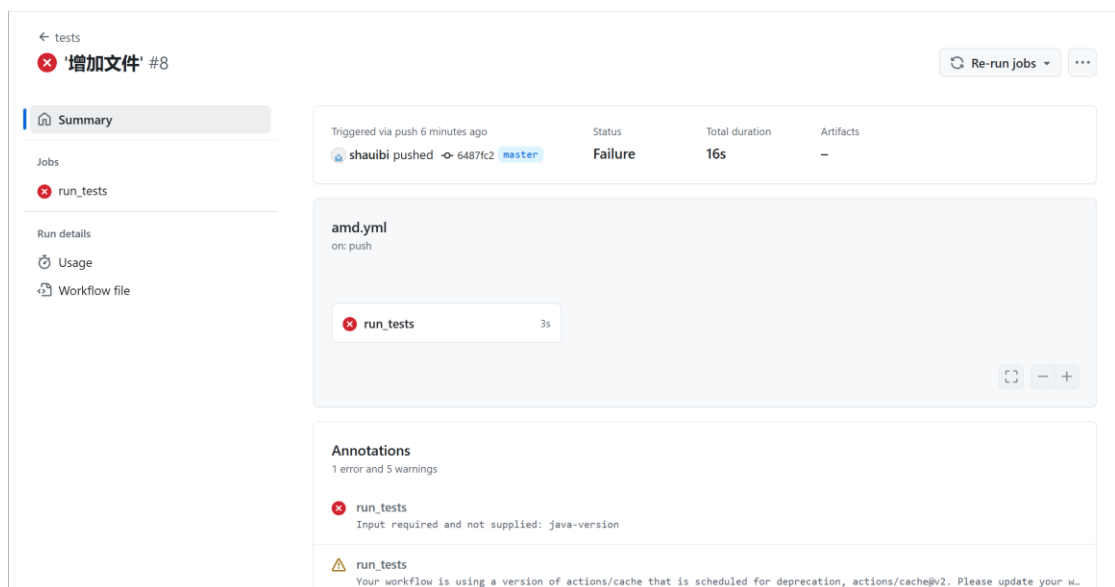
项目的目录结构截图



## 执行自动化成功界面：



## 执行自动化测试失败界面：



## 执行具体信息：

The top screenshot shows a successful GitHub Actions workflow run for '增加文件' #9. The job 'run\_tests' succeeded 4 minutes ago. The workflow steps are: Set up job (1s), Checkout the repository (1s), Set up JDK 21 (18s), Cache Maven packages (1s), Run tests with Maven (6s), Post Cache Maven packages (2m 12s), Post Set up JDK 21 (0s), Post Checkout the repository (0s), and Complete job (0s).

The bottom screenshot shows a failed GitHub Actions workflow run for '增加文件' #8. The job 'run\_tests' failed 7 minutes ago. The failure is due to an error in the 'Set up JDK 21' step: 'Warning: Unexpected input(s) 'jdk-version', valid inputs are ['java-version', 'java-package', 'architecture', 'jdkfile', 'server-id', 'server-username', 'server-password', 'settings-path', 'gpg-private-key', 'gpg-passphrase']' and 'Error: Input required and not supplied: java-version'.

代码:

name: tests with Auto Merge

on:

push:

branches:

- master

pull\_request:

types: [opened, synchronize, ready\_for\_review, review\_requested]

jobs:

run\_tests:

```
runs-on: ubuntu-latest
steps:
  - name: Checkout the repository
    uses: actions/checkout@v2
    with:
      ref: ${ { github.event.pull_request.head.sha } }
      fetch-depth: 0
  - name: Set up JDK 21
    uses: actions/setup-java@v1
    with:
      java-version: 21
  - name: Cache Maven packages
    uses: actions/cache@v2
    with:
      path: ~/.m2
      key: ${ { runner.os } }-m2-${ { hashFiles('**/pom.xml') } }
      restore-keys: ${ { runner.os } }-m2
  - name: Run tests with Maven
    run: mvn -B test --file ${GITHUB_WORKSPACE}/pom.xml

auto_merge:
  runs-on: ubuntu-latest
  needs: run_tests
  if: github.event_name == 'pull_request' && github.event.pull_request.draft == false &&
github.event.pull_request.mergeable == true
  steps:
    - name: Checkout the repository
      uses: actions/checkout@v2
      with:
        ref: ${ { github.event.pull_request.head.sha } }
    - name: Merge pull request
      uses: peters-seo/merge-pull-request@v1
      with:
        token: ${ { secrets.GITHUB_TOKEN } }
        merge-method: rebase
        author-email: ${ { github.event.pull_request.user.email } }
        author-name: ${ { github.event.pull_request.user.login } }
        commit-message: 'Merge pull request'
```

### 3 实验内容 2 Jenkins DevOps 实践

六部分的操作截图：

源码管理：

源码管理

☐ 无

☒ Git ?

Repositories ?

Repository URL ?

git@github.com:OSSDP/Lab4-2022211876.git

Credentials ?

shauibi (EQcEYM511kFAGLxNs2OKPi2QDeIr7UjOzS3vOQ+y69c)

+ 添加

高级 ▾

Add Repository

构建触发器：

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11日星期一 中国标准时间 11:28:40; would next run at 2024年11月11日星期一 中国标准时间 11:28:40.

☐ 忽略钩子 post-commit ?

构建步骤：

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=$(cat /dev/urandom | tr -dc 'a-z0-9' | fold -n 40 | shuf | xargs echo | sha256sum | cut -d ' ' -f 1)
gh pr create --title "Auto PR from dev-Jenkins to master" --body "This is an auto PR from Jenkins." --base master --head dev --repo OSSDP/Lab4-2022211876
```

步骤七的实验效果:

✅ #8 (2024年11月11日 11:26:54)



启动用户 [wanmingqi](#)



This run spent:

- 4 ms waiting;
- 13 sec build duration;
- 13 sec total from scheduled to completion.



Revision: 7faab70c64e356a5b176ec12e18707229f133a6b  
Repository: [git@github.com:OSSDP/Lab4-2022211876.git](#)

- [refs/remotes/origin/dev](#)



没有变化。



Auto PR from dev-Jenkins to master ✖

#2 opened 5 minutes ago by [shauibi](#)

实验各个步骤的操作截图:



## 4 小结

在完成 GitHub Actions 和 Jenkins 的 DevOps 实践实验后，我深刻体会到了持续集成和持续部署（CI/CD）在现代软件开发中的重要性。

通过 GitHub Actions 实验，我掌握了如何利用 YAML 文件配置自动化 workflow，实现了代码的自动测试和部署，这极大地提高了开发效率和代码质量。

同时，Jenkins 实验让我深入了解了如何搭建和配置 Jenkins 服务器，通过编写 Jenkinsfile 来定义 CI/CD 流水线，我学会了如何自动化处理复杂的构建和部署任务。

尽管在实验过程中遇到了配置错误和环境兼容性等挑战，但通过不断尝试和解决问题，我不仅提升了自己的技术能力，也增强了对 DevOps 工具链的理解和应用。这些实验不仅让我对自动化流程有了更全面的认识，也激发了我将这些知识应用到未来项目中的热情，以期达到更高效和可靠的软件交付。