

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
2022210159	施昊辰	1403525867@qq.com

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	4
4	小结	9

1 实验要求

掌握 DevOps 基本流程：通过实验，学生需要熟悉并掌握开源软件开发中 DevOps 的基本流程与工具，包括自动化测试、持续集成(CI)和持续交付(CD)。

Github Actions 实践：学生需要在 Github 上使用 Github Actions 进行 DevOps 实践，配置自动化 workflow，完成项目的自动化测试、代码提交等操作。

Jenkins 实践：学生还需要通过 Jenkins 实现 DevOps 操作，包括自动化构建、测试、PR 提交等功能。

实验环境配置：实验要求在本地环境中配置 Java JDK、Git、Maven 等开发工具，并将代码上传到 Github 仓库中，结合自动化工具完成流程的验证。

2 实验内容 1 Github Actions DevOps 实践

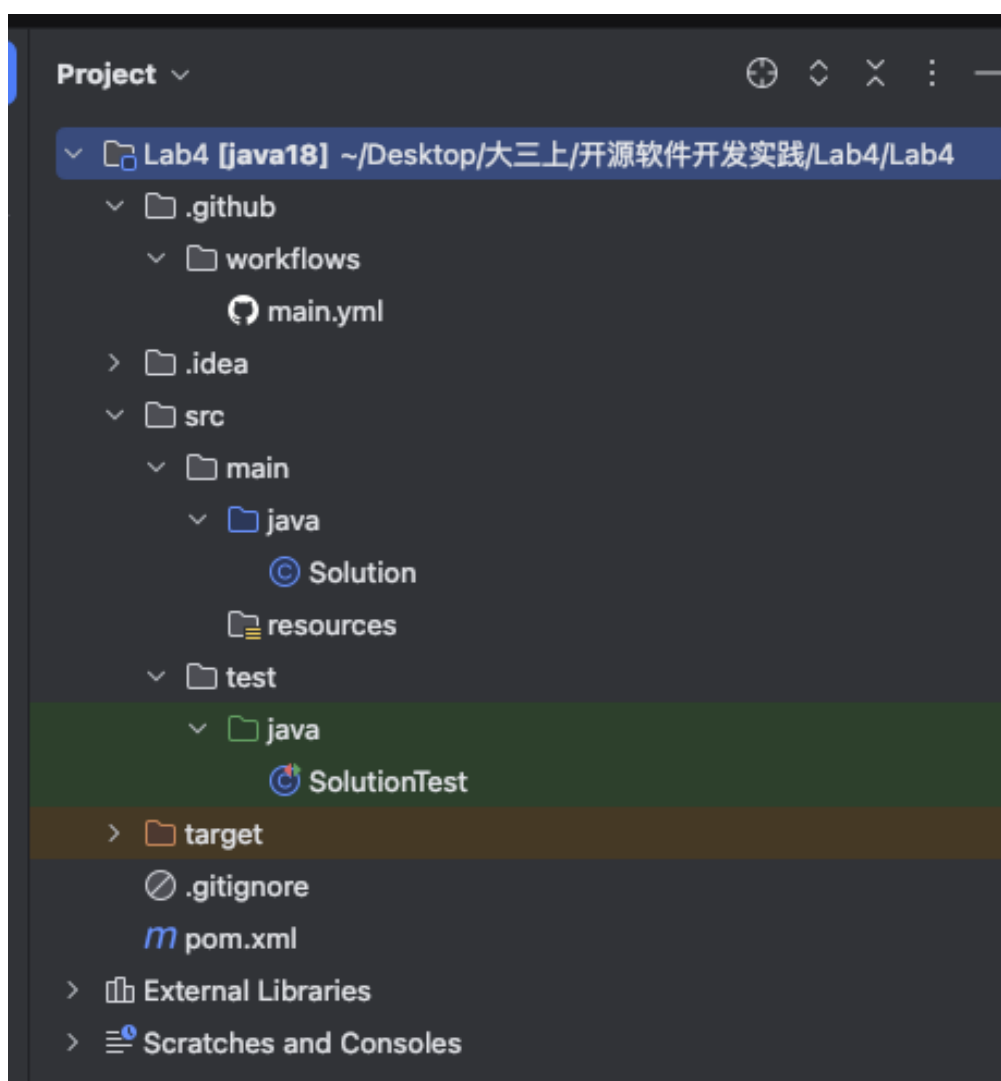


图 1 项目目录

yaml 文件内容示例：

```
1  name: tests
2  on: push
3  jobs:
4    run_tests:
5      runs-on: ubuntu-latest
6      steps:
7        - name: Checkout the repository
8          uses: actions/checkout@v3
9
10       - name: Set up JDK 20
11         uses: actions/setup-java@v3 # 升级到 v3
12         with:
13           distribution: 'temurin' # 指定发行版
14           java-version: 20       # 指定 JDK 版本
15
16       - name: Cache Maven packages
17         uses: actions/cache@v3
18         with:
19           path: ~/.m2
20           key: ${ runner.os }}-m2-${ hashFiles('**/pom.xml') }}
21           restore-keys: ${ runner.os }}-m2
22
23       - name: Run tests with Maven
24         run: mvn -B -X test --file pom.xml
25
```

图 2 yaml 文件内容

执行自动化测试成功界面：

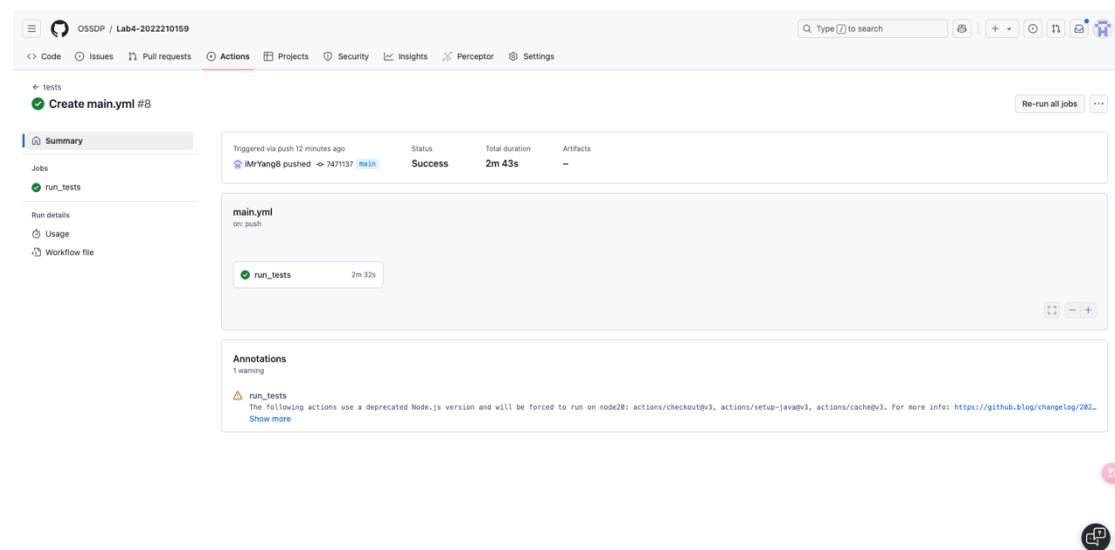


图 3 执行自动化测试成功界面

执行自动化测试失败界面：

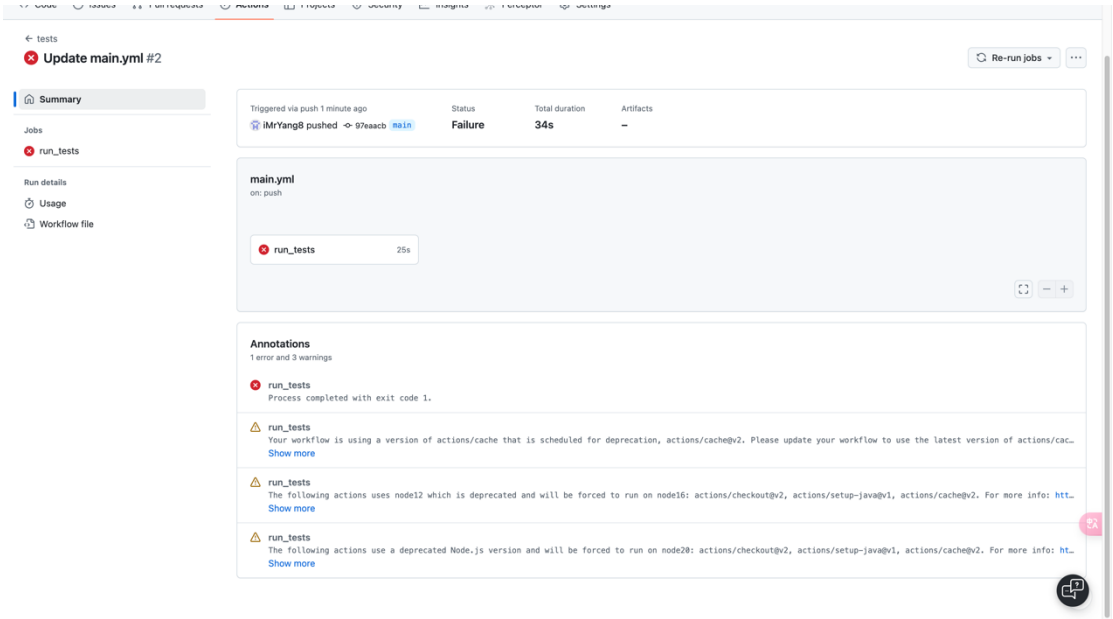


图 4 执行自动化测试失败界面

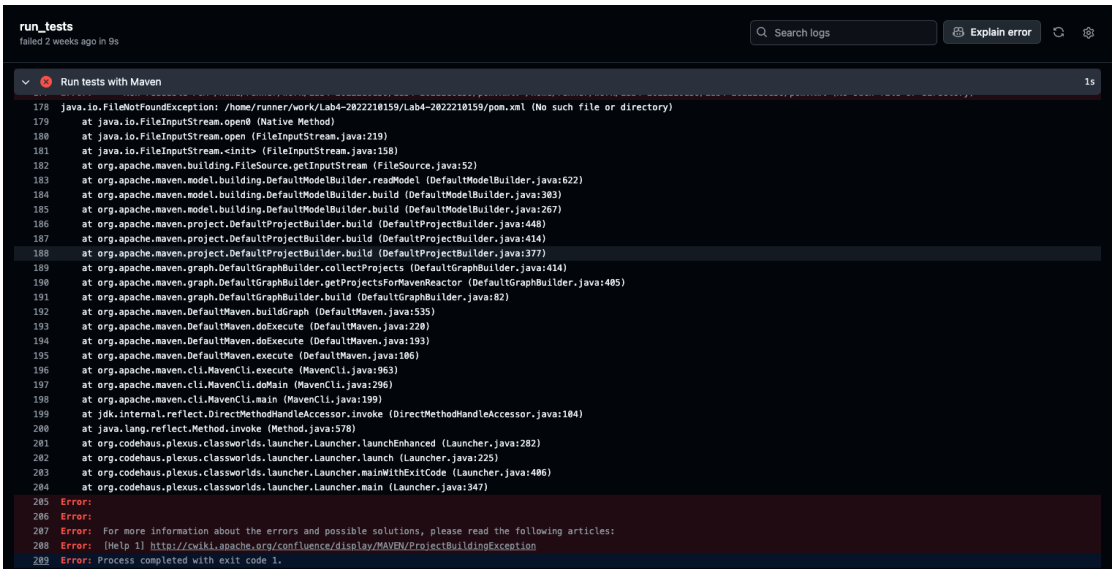


图 5 失败截图

选作任务：

PR 的自动 merge 功能

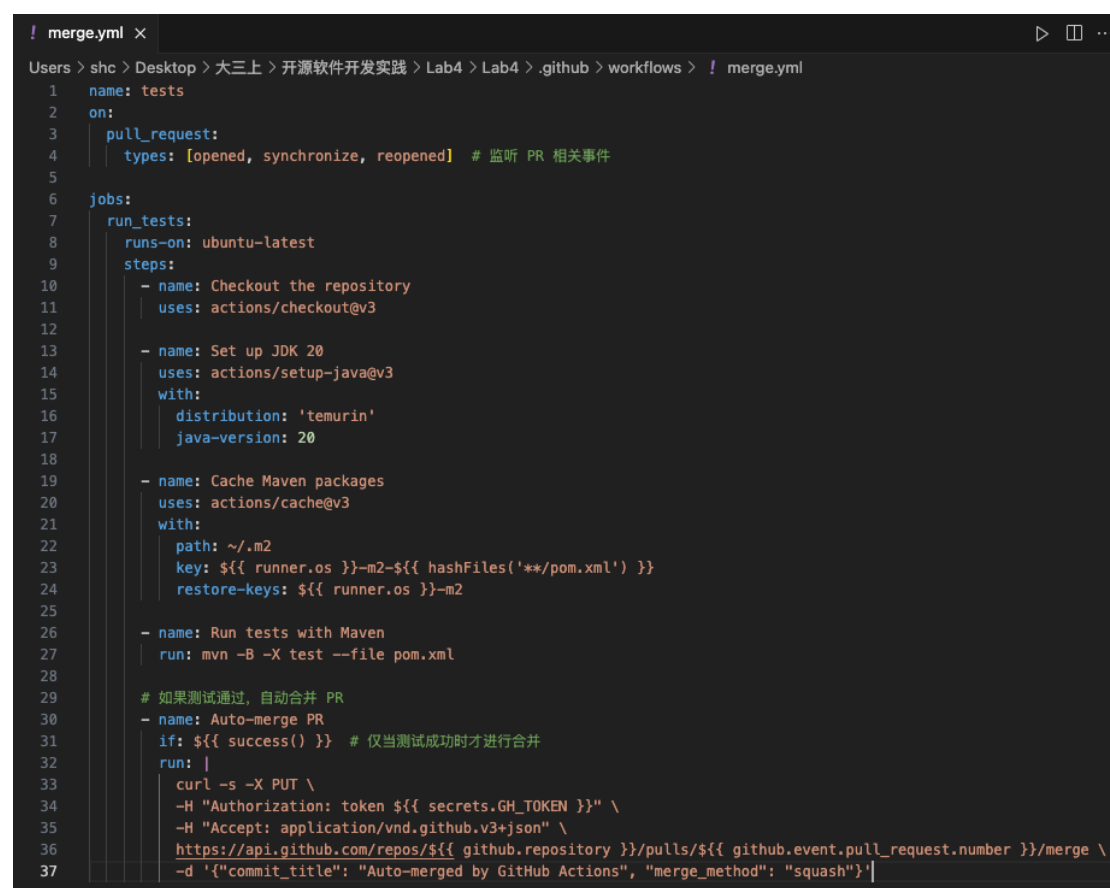


图 6 PR 自动 merge 的 yml

触发事件: 我们监听 `pull_request` 事件(包括 `opened`, `synchronize`, 和 `reopened`), 这意味着每当有 PR 被创建、更新或重新打开时, 工作流都会自动运行。

Auto-merge PR 步骤:

条件: 该步骤使用 `if: ${ success() }`, 确保只有在所有前面的测试步骤成功通过, 才会执行自动合并。

GitHub API 调用: 我们使用 `curl` 通过 GitHub API 自动合并 PR。

Authorization: 使用存储在 GitHub Secrets 中的 `GH_TOKEN` 进行身份验证。

合并策略: 使用 `merge_method: "squash"`, 将所有提交压缩为一个提交(推荐使用 `squash` 合并, 以保持主分支的整洁)。

PR 编号: `${ github.event.pull_request.number }` 动态获取当前 PR 的编号。

3 实验内容 2 Jenkins DevOps 实践

Jenkins 安装与配置

在 mac 上安装 jenkins

- Install the latest LTS version: `brew install jenkins-lts`

- Start the Jenkins service: brew services start jenkins-lts

```
(base) shc@Jimmy-MacBook-Air JavaVirtualMachines % brew services start jenkins-lts
==> Tapping homebrew/services
Cloning into '/opt/homebrew/Library/Taps/homebrew/homebrew-services'...
remote: Enumerating objects: 3445, done.
remote: Counting objects: 100% (591/591), done.
remote: Compressing objects: 100% (224/224), done.
remote: Total 3445 (delta 434), reused 416 (delta 363), pack-reused 2854 (from 1)
Receiving objects: 100% (3445/3445), 1004.95 KiB | 1.77 MiB/s, done.
Resolving deltas: 100% (1674/1674), done.
Tapped 2 commands (51 files, 1.2MB).
TMPDIR is world-writable: /private/tmp
==> Successfully started `jenkins-lts` (label: homebrew.mxcl.jenkins-lts)
```

图 7 运行 Jenkins 成功截图

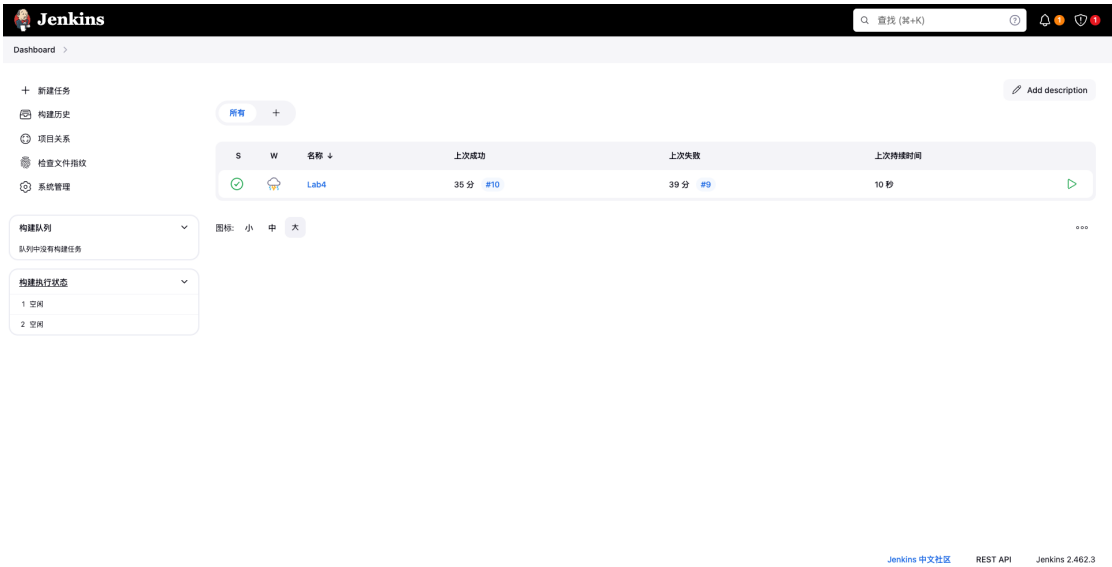


图 8 Jenkins 界面

Github CLI

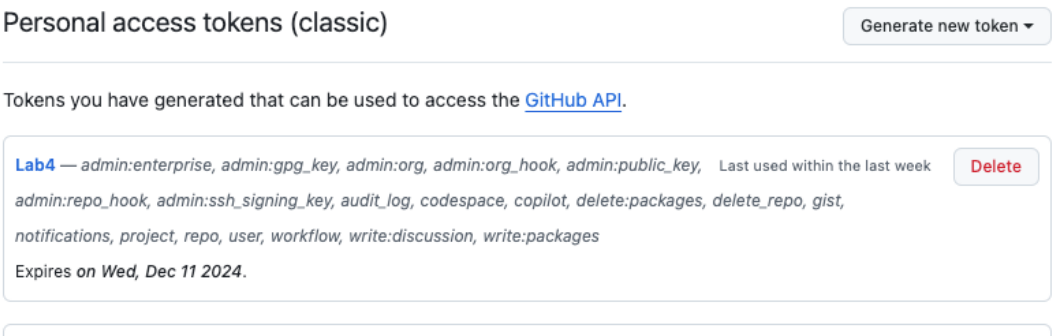


图 9 token 创建

构建 DevOps workflow

Repositories ?

Repository URL ?

git@github.com:OSSDP/Lab4-2022210159.git

Credentials ?

iMrYang8 (Github ssh 私钥)

+ 添加

高级 ▾

图 10 配置 URL

Jenkins 凭据提供者: Jenkins

添加凭据

Domain

全局凭据 (unrestricted)

类型

SSH Username with private key

范围 ?

全局 (Jenkins, nodes, items, all child items, etc)

ID ?

Github_Private_Key

描述 ?

Github ssh 私钥

Username

iMrYang8

☐ Treat username as secret ?

Private Key

构建触发器

☐ 其他工程构建后触发 ?

☐ 定时构建 ?

☐ GitHub hook trigger for GITScm polling ?

☒ 轮询 SCM ?

日程表 ?

H/3 * * * *

上次运行的时间 2024年11月11日星期一 中国标准时间 12:27:23; 下次运行的时间 2024年11月11日星期一 中国标准时间 12:30:23.

☐ 忽略钩子 post-commit ?

图 11 构建触发器

Build Steps

≡ 调用顶层 Maven 目标 ?

Maven 版本

Lab4

目标

validate

高级 ▾

≡ 调用顶层 Maven 目标 ?

Maven 版本

Lab4

目标

test

高级 ▾

≡ 执行 shell ?

命令

查看 可用的环境变量列表

```
export GH_TOKEN=ghp_XXX
gh pr create --title "Auto PR from dev~Jenkins to main" --body "This is an auto PR from Jenkins." --base main --head dev --repo OSSDP/Lab4-2022
```

图 12 Bulid Steps

✓ #10 (2024年11月11日 11:50:06)



由匿名用户触发



This run spent:

- 13 毫秒 waiting;
- 10 秒 build duration;
- 10 秒 total from scheduled to completion.



Revision: daa4ef346b633cbecf2ad086c5b05370a2cbfdef
Repository: git@github.com:OSSDP/Lab4-2022210159.git

- refs/remotes/origin/dev



没有变化。

图 13 实验成功截图

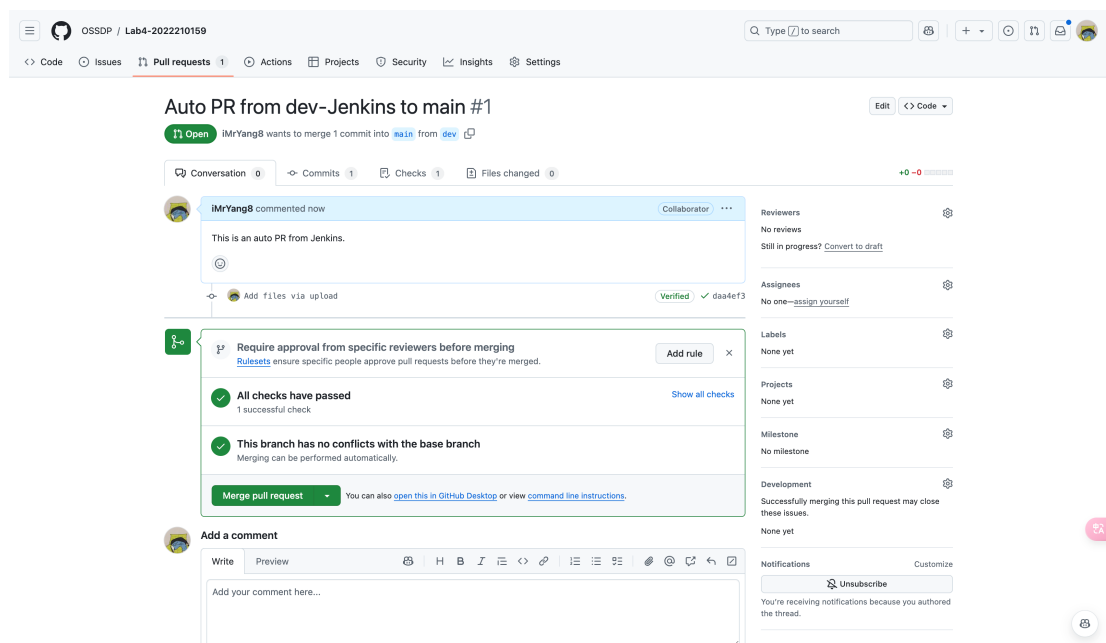


图 14 PR 推送成功

4 小结

DevOps 流程的理解：通过该实验，我对 DevOps 的概念有了更深入的理解，特别是如何通过自动化工具实现代码的持续集成与交付。Github Actions 和 Jenkins 各有特点，前者适合于 GitHub 的生态系统，后者则更为灵活，适应各种场景。

自动化工具的使用：通过配置 Github Actions 和 Jenkins，我掌握了如何编写工作流文件、触发自动化测试、提交代码并生成 PR，这些操作极大地提高了开发效率。

问题解决能力的提升：实验过程中遇到了 Github 权限问题、Jenkins 配置问题等，通过查阅文档和网上资料，我学会了如何解决这些实际问题，例如通过 SSH 认证解决 Jenkins 的仓库访问问题。

团队协作的启发：在实验过程中，我意识到 DevOps 在团队协作中的重要性。通过自动化工具，每个团队成员的代码都可以被及时测试和集成，减少了手动操作带来的风险。