

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
2022111654	李宸	Lichen2022@stu.hit.edu.cn

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	5

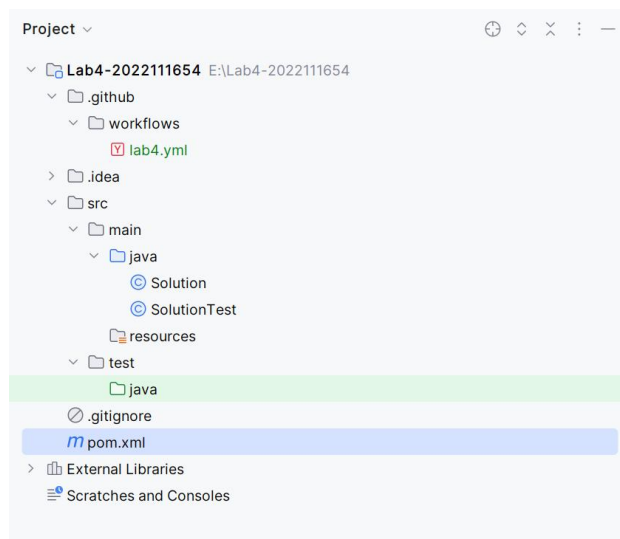
1 实验要求

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

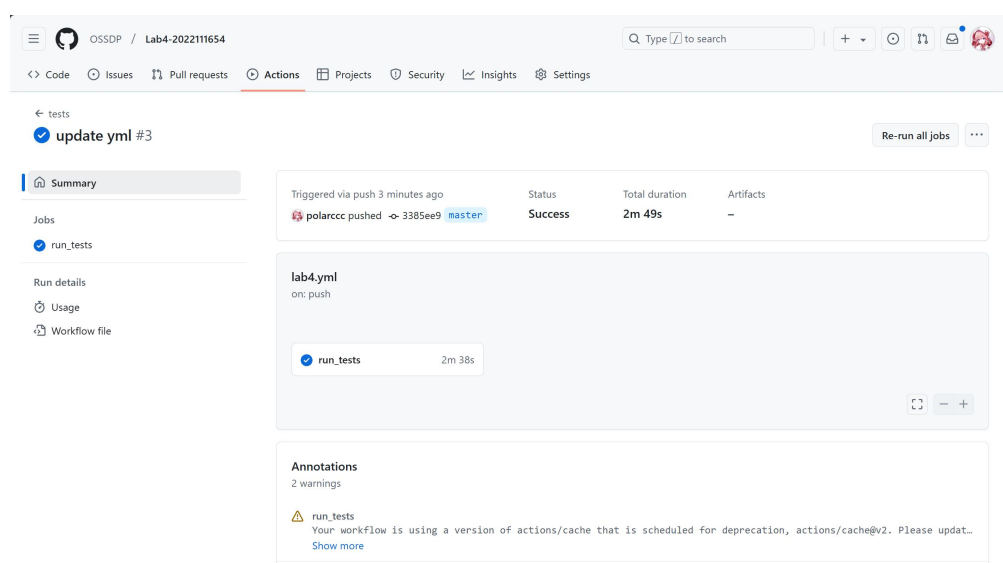
2 实验内容 1 Github Actions DevOps 实践

- 给出项目的目录结构截图，文件管理器或编程 IDE 的界面均可。

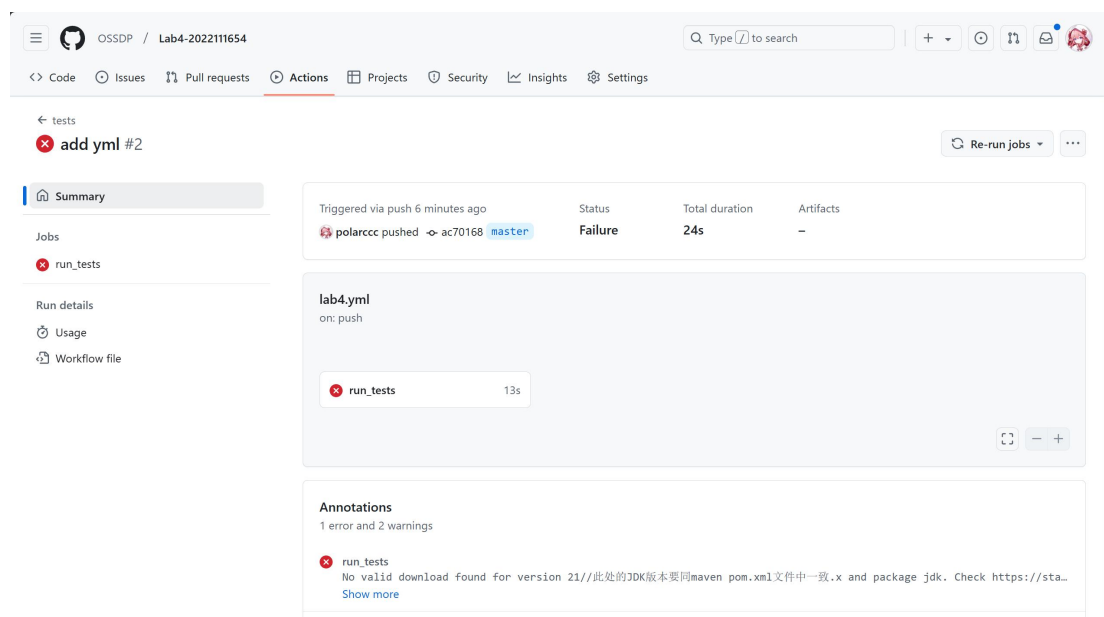


- **Github 中的 Actions 的若干界面**

1. 执行自动化测试成功界面

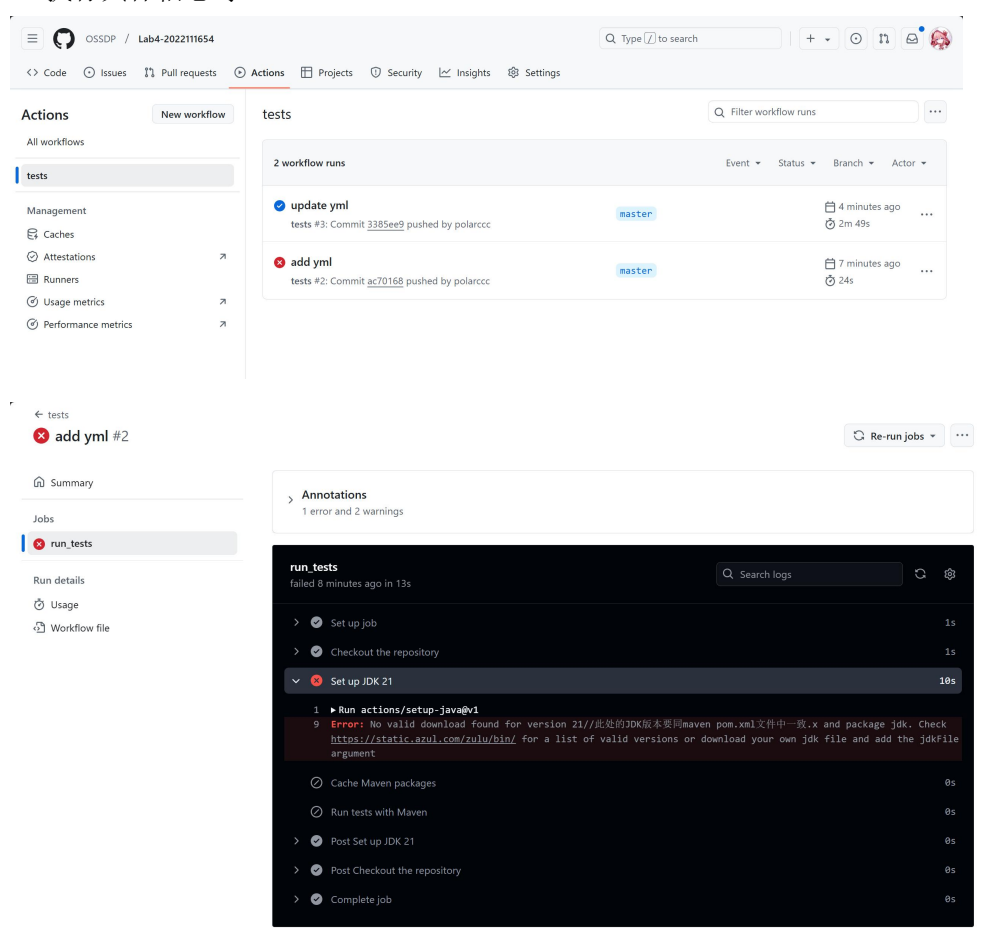


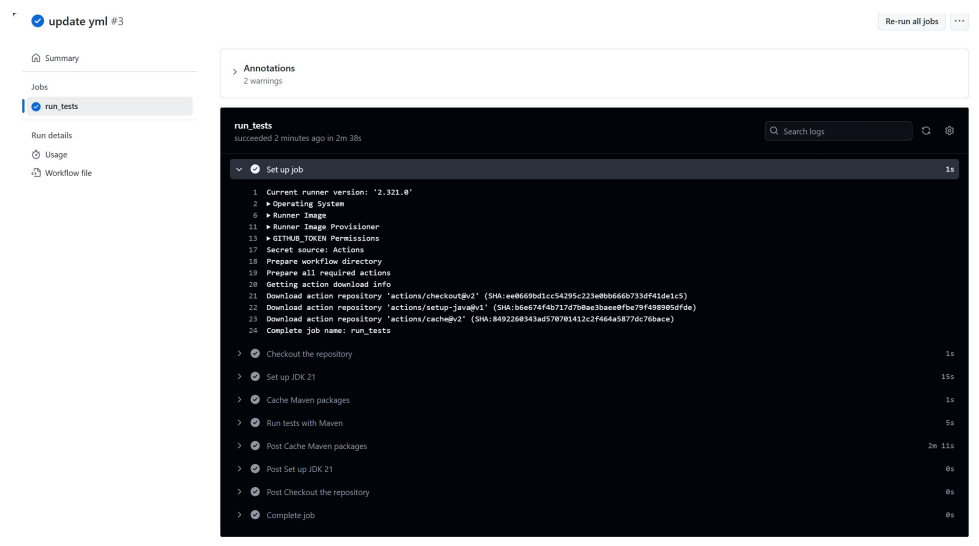
2. 执行自动化测试失败界面



失败的原因是注释不符合规范，调整正确即可。

3. 执行具体信息等





- 给出针对步骤八的 YML 代码（如果完成了）。

name: Auto Merge Pull Requests

on:

pull_request:

types: [opened, synchronize, reopened] # 当 PR 被创建、更新或重新打开时触发

branches:

- main # 针对 main 分支的 PR 进行操作

jobs:

#（与原代码相同）

merge:

needs: test # 确保在测试通过后才执行合并

runs-on: ubuntu-latest

if: \${{ github.event.pull_request.mergeable == true && github.event.pull_request.state == 'open' }} # 仅当 PR 可合并并且是开放状态时触发

steps:

- name: Checkout code

uses: actions/checkout@v3

- name: Auto-merge PR

uses: "peter-evans/merge-pull-request@v3"

with:

pull-request: \${{ github.event.pull_request.number }}

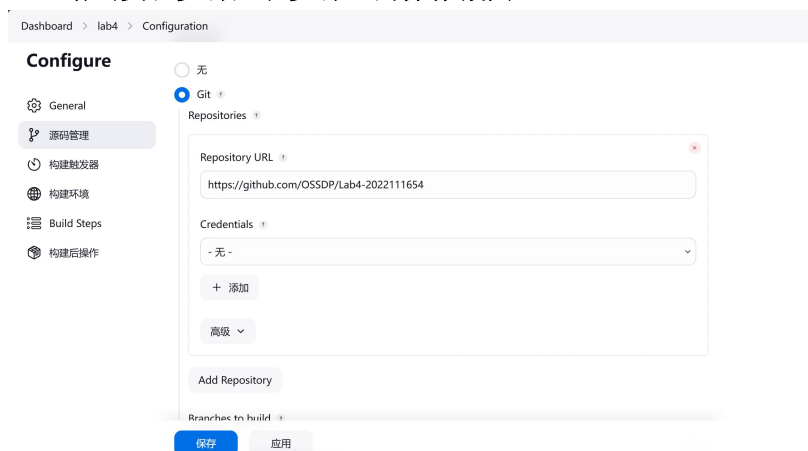
merge-method: "merge" # 可以选择 "merge", "squash", 或 "rebase"

env:

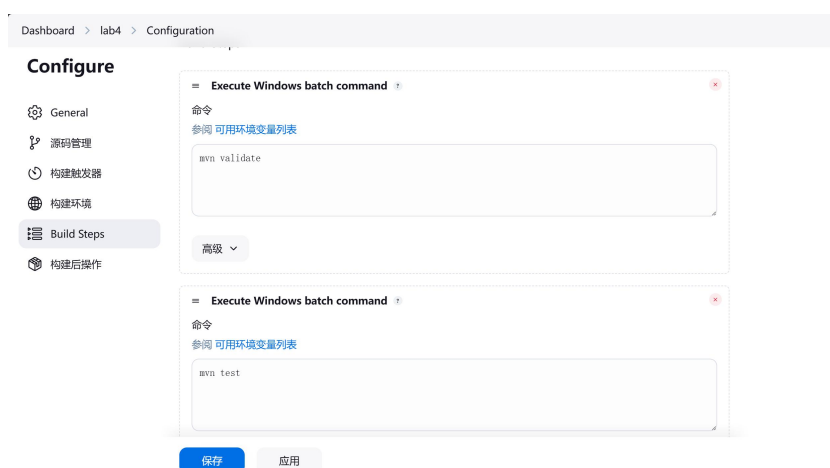
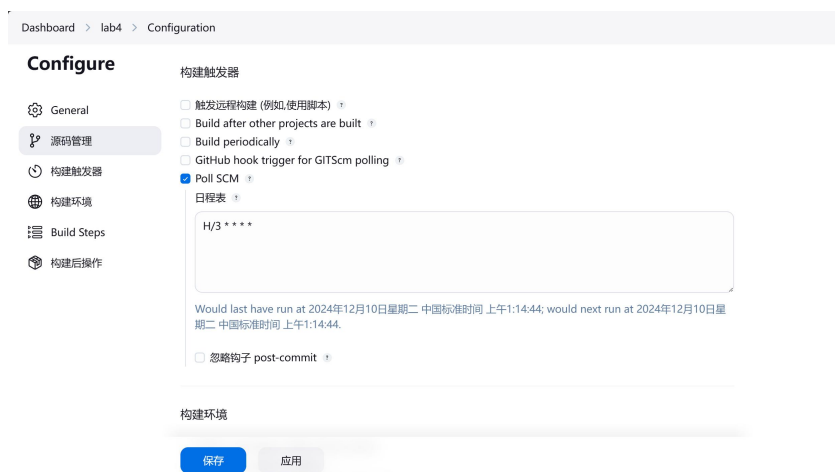
GITHUB_TOKEN: \${{ secrets.GITHUB_TOKEN }}

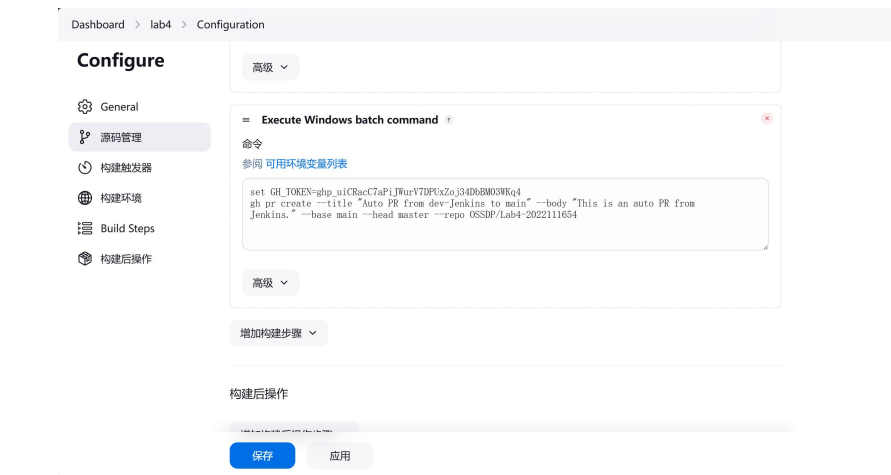
3 实验内容 2 Jenkins DevOps 实践

● 给出实验步骤六和步骤七的操作截图



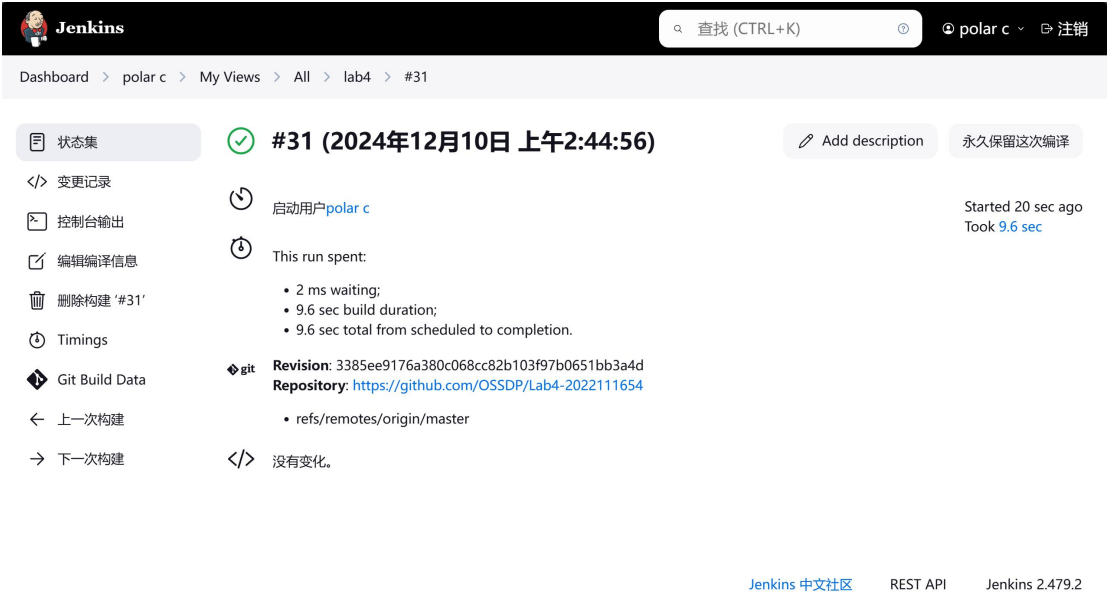
这里采用 https 连接是因为 ssh 连接有问题





ghp_uiCRacC7aPiJWurV7DPuxZoj34DbBM03WKq4

- 给出实验中各步骤结果的截图



尝试了 30 次后终于成功

4 小结

通过使用 Jenkins，我深刻体会到它在自动化构建和部署中的重要性。Jenkins 的持续集成功能可以帮助团队自动化构建、测试和发布流程，减少手动操作，提高开发效率。尽管在配置过程中遇到了一些挑战，但它的灵活性和可扩展性使得项目管理更加高效，尤其是在团队协作和快速迭代的环境中。