

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
2022211915	周李铧	2469662369@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.....	2
4	小结.....	6

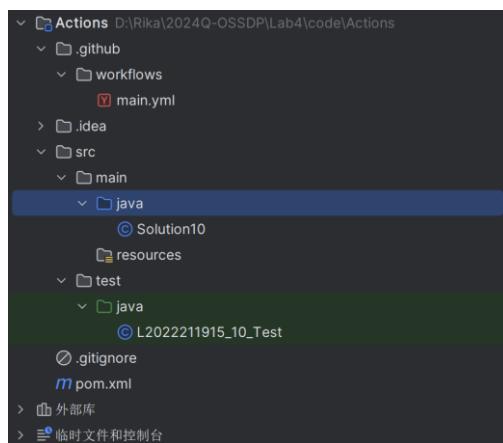
1 实验要求

本次实验训练开源软件开发中的基本 DevOps 操作，具体来说：

- ① 掌握开源软件开发中的基本 DevOps 流程和工具的使用
- ② 熟悉利用 Github Actions 进行 DevOps
- ③ 熟悉利用 Jenkins 进行 DevOps

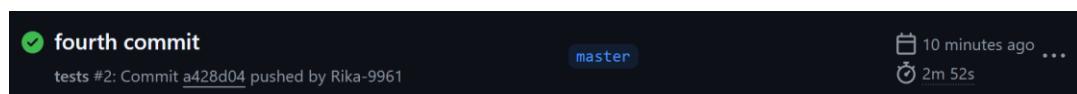
2 实验内容 1 Github Actions DevOps 实践

(一) 项目目录结构



(二) Github 中 Actions 的界面

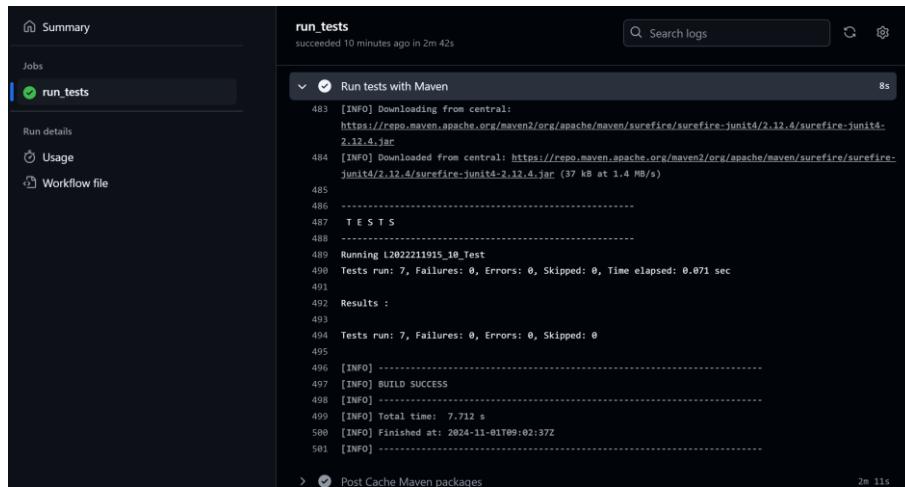
(1) 执行自动化测试成功界面



(2) 执行自动化测试失败界面



(3) 执行具体信息



3 实验内容 2 Jenkins DevOps 实践

(一) 实验开发环境

Java JDK 版本:

```
C:\Users\Rika>java -version
openjdk version "17.0.9" 2023-10-17 LTS
OpenJDK Runtime Environment Microsoft-8552009 (build 17.0.9+8-LTS)
OpenJDK 64-Bit Server VM Microsoft-8552009 (build 17.0.9+8-LTS, mixed mode, sharing)
```

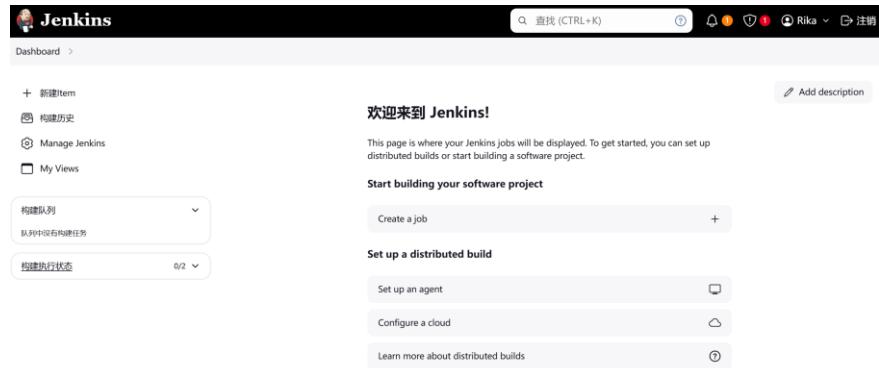
Git 版本:

```
C:\Users\Rika>git -v
git version 2.42.0.windows.2
```

Maven 版本:

```
C:\Users\Rika>mvn -v
Apache Maven 3.9.6 (bc0240f3c744dd6b6ec2920b3cd08dcc295161ae)
Maven home: D:\develop\apache-maven-3.9.6
Java version: 17.0.9, vendor: Microsoft, runtime: C:\Program Files\Microsoft\jdk-17.0.9.8-hotspot
Default locale: zh_CN, platform encoding: GBK
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"
```

(二) Jenkins 安装与配置



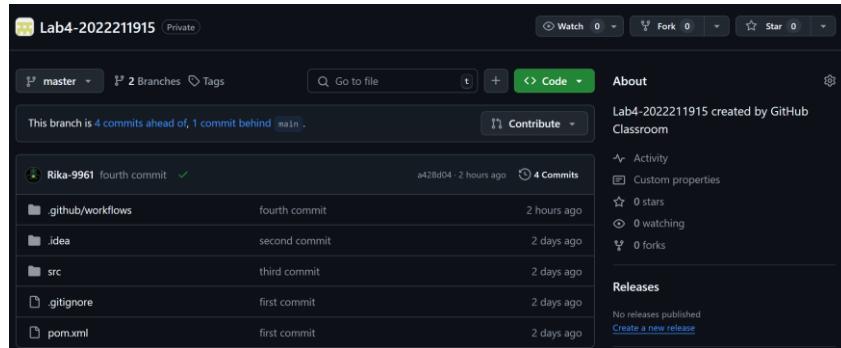
(三) Github CLI 安装

```
C:\Users\Rika>gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'workflow'.
? Paste your authentication token: *****
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as Rika-9961

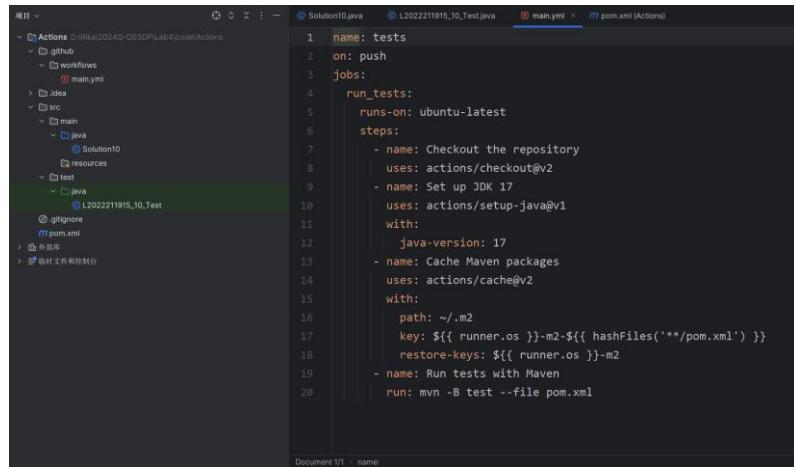
C:\Users\Rika>gh repo list
Showing 3 of 3 repositories in @Rika-9961

NAME DESCRIPTION INFO UPDATED
Rika-9961/OSSDP-Lab2-fork public, fork about 14 days ago
Rika-9961/Lab1 public about 7 months ago
Rika-9961/Test01 public about 7 months ago
```

(四) 测试用 github 仓库



(五) 本地创建 maven 项目并提交



(六) 构建 DevOps Workflow

1. General 部分

该部分是对任务的描述以及一些涉及并发构建、构建结果处理、参数配置的内容。

2. 源码管理

在该部分我们需要对前文设置的 Github 仓库进行访问配置。

使用 SSH 访问 Github 仓库需要额外的认证凭证，点击“添加”按钮进行配置。

在 Username 处填入 Github 账户名，Private Key 处粘贴私钥内容。

Jenkins Credentials Provider: Jenkins

类型: SSH Username with private key

范围: Global (Jenkins, nodes, items, all child items, etc)

ID: Github_Private_Key

描述: Github SSH Private Key

Username: Rika-9961

Treat username as secret

Private Key: Enter directly

Key:
-----BEGIN OPENSSH PRIVATE KEY-----
MIIBdQIBAAEABAAAAQDwBewYHAAAACGAAQD9tC1fCtf
-----END OPENSSH PRIVATE KEY-----

指定 Local 项目代码所在的分支。

Branches to build ?

指定分支 (为空时代表any) ?

/master

Add Branch

3. 构建触发器

使用 Poll SCM, 定时检查仓库是否有更新, 进行自动化构建, 每 3 分钟检查一次仓库。

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 * * * *

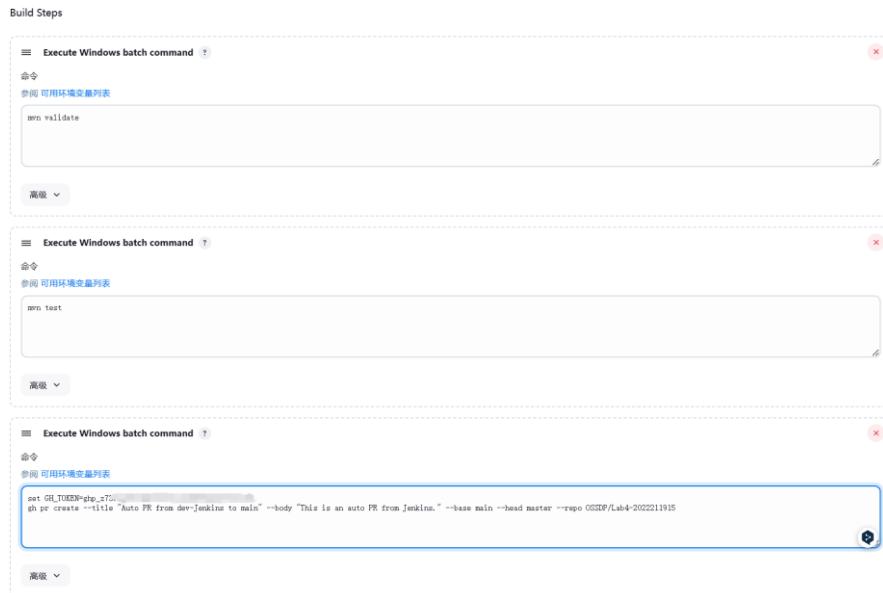
Would last have run at 2024年11月8日星期五 中国标准时间 上午11:18:23; would next run at 2024年11月8日星期五 中国标准时间 上午11:18:23.

4. 构建环境

该部分用于设置构建流程的环境变量。

5. 构建步骤

这部分将使用 cmd 脚本进行代码的测试与 PR 的提交。



(七) 验证实验效果

将项目代码提交到选定的仓库的指定分支，等待定时任务进行构建流程，查看 Jenkins Build 的结果。

#19 (2024年11月9日 下午1:47:37)

启动用户Rika

This run spent:

- 2 ms waiting;
- 9 sec build duration;
- 9 sec total from scheduled to completion.

Revision: ea21475445f4f95055c1cba6bc491658a6fe72b5
Repository: <https://github.com/OSSDP/Lab4-2022211915.git>

- refs/remotes/origin/master

Changes

</> 1. sixth commit ([details](#) / [githubweb](#))

在 Github 上查看仓库的 PR 推送结果。

Auto PR from dev-Jenkins to main #1

Rika-9961 wants to merge 2 commits into `main` from `master`

Conversation 0 Commits 2 Checks 1 Files changed 1

Rika-9961 commented 3 minutes ago
This is an auto PR from Jenkins.

Rika-9961 added 2 commits 17 minutes ago
sixth commit

ea21475

4 小结

通过本次实验，我深入了解了开源软件开发中的基本 DevOps 流程及其常用工具。`GitHub Actions` 提供了一种简洁、无缝的方式来实现 CI/CD，并且直接集成在 GitHub 上，适合开源项目的自动化管理；`Jenkins` 则是一个功能更强大、插件丰富的工具，适合复杂的企业级应用，它的灵活性和可扩展性使其成为大规模项目的首选工具。

这两个工具都极大地提高了软件开发中的自动化水平，减少了手动操作的出错概率，并提升了软件发布的效率，有利于提升自动化程度，并帮助开发团队加速软件的交付和迭代。