

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
2022211592	李鑫元	13275315616

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	10

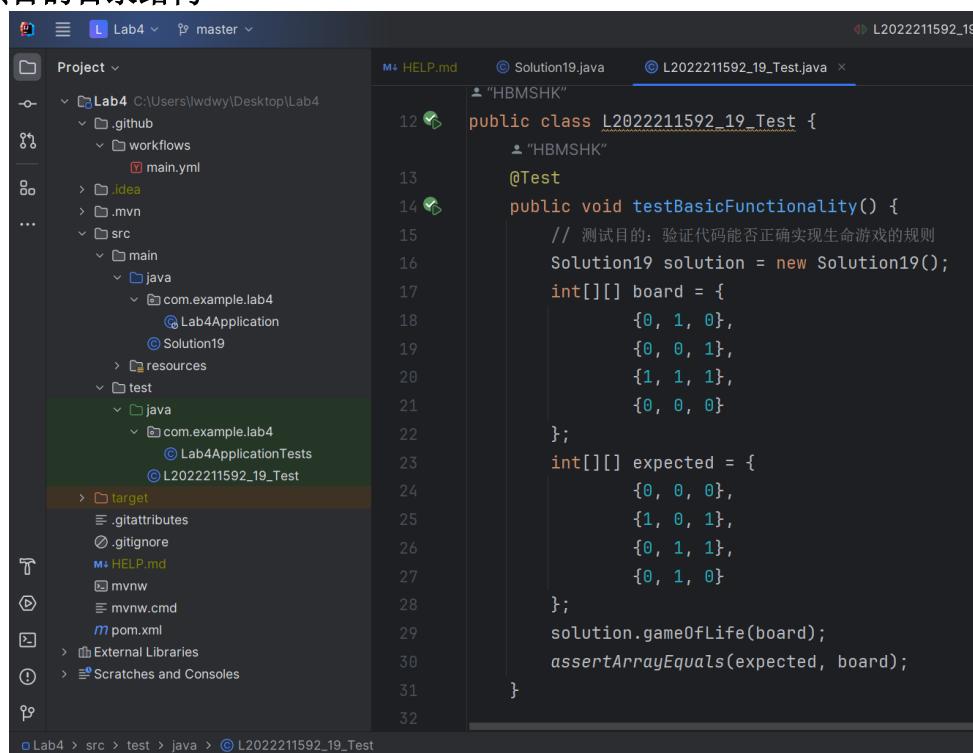
1 实验要求

熟悉基本 DevOps 操作：

- 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

2、1 项目的目录结构



The screenshot shows a Java IDE interface with the following details:

- Project View:** Shows the project structure under "Lab4". It includes a ".github" folder containing "main.yml", an ".idea" folder, an "mvn" folder, and a "src" folder. The "src" folder contains "main" and "test" packages. The "main" package has "com.example.lab4" and "Solution19" classes. The "test" package has "com.example.lab4" and "L2022211592_19_Test" classes. A "target" folder is also present.
- Code Editor:** Displays the "L2022211592_19_Test.java" file. The code is a unit test for the "Solution19" class, specifically for the "gameOfLife" method. It uses a 3x3 board state to verify the expected output.
- Status Bar:** Shows the path "Lab4 > src > test > java > L2022211592_19_Test" and the file name "L2022211592_19_Test.java".

2、2 项目测试用例运行结果

The screenshot shows the IntelliJ IDEA interface. The left sidebar displays the project structure under 'Lab4' with packages like 'com.example.lab4' and 'com.example.lab4.test'. The right pane shows the code for 'Solution19.java'. The 'Run' tab at the bottom shows the output of running the test class 'L2022211592_19_Test', which includes five test cases: 'testEmptyBoard()', 'testBasicFunctionality()', 'testSingleCell()', 'testBoundaryConditions()', and 'testIrregularBoard()'. All tests passed in 36ms. The status bar at the bottom right shows '8:1 CRLF UTF-8 4 spa'.

2、3 maven 项目推送到 GitHub 仓库

```
李鑫元@I-MINGW64 ~/Desktop/Lab4 (master)
$ git add .

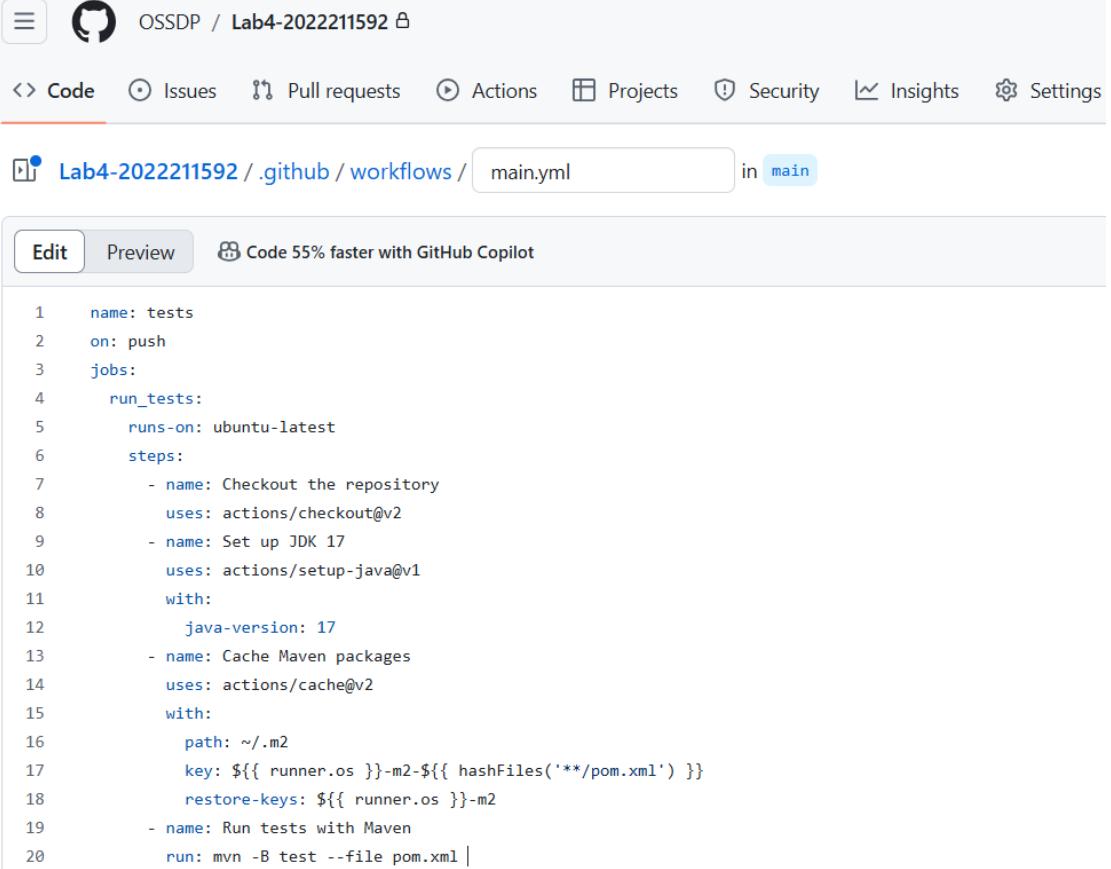
李鑫元@I-MINGW64 ~/Desktop/Lab4 (master)
$ git commit -m "first commit"
[master (root-commit) 3dfd133] first commit
11 files changed, 762 insertions(+)
create mode 100644 .gitattributes
create mode 100644 .gitignore
create mode 100644 .mvn/wrapper/maven-wrapper.properties
create mode 100644 mvnw
create mode 100644 mvnw.cmd
create mode 100644 pom.xml
create mode 100644 src/main/java/Solution19.java
create mode 100644 src/main/java/com/example/lab4/Lab4Application.java
create mode 100644 src/main/resources/application.properties
create mode 100644 src/test/java/L2022211592_19_Test.java
create mode 100644 src/test/java/com/example/lab4/Lab4ApplicationTests.java

李鑫元@I-MINGW64 ~/Desktop/Lab4 (master)
$ git status
On branch master
nothing to commit, working tree clean

李鑫元@I-MINGW64 ~/Desktop/Lab4 (master)
$ git remote add origin4 https://github.com/OSSDP/Lab4-2022211592

李鑫元@I-MINGW64 ~/Desktop/Lab4 (master)
$ git push -u origin4 master
Enumerating objects: 27, done.
Counting objects: 100% (27/27), done.
Delta compression using up to 20 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (27/27), 11.45 KiB | 5.72 MiB/s, done.
Total 27 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:     https://github.com/OSSDP/Lab4-2022211592/pull/new/master
remote:
To https://github.com/OSSDP/Lab4-2022211592
 * [new branch]      master -> master
branch 'master' set up to track 'origin4/master'.
```

2、4 在 Github 上编写 DevOps workflow 文件



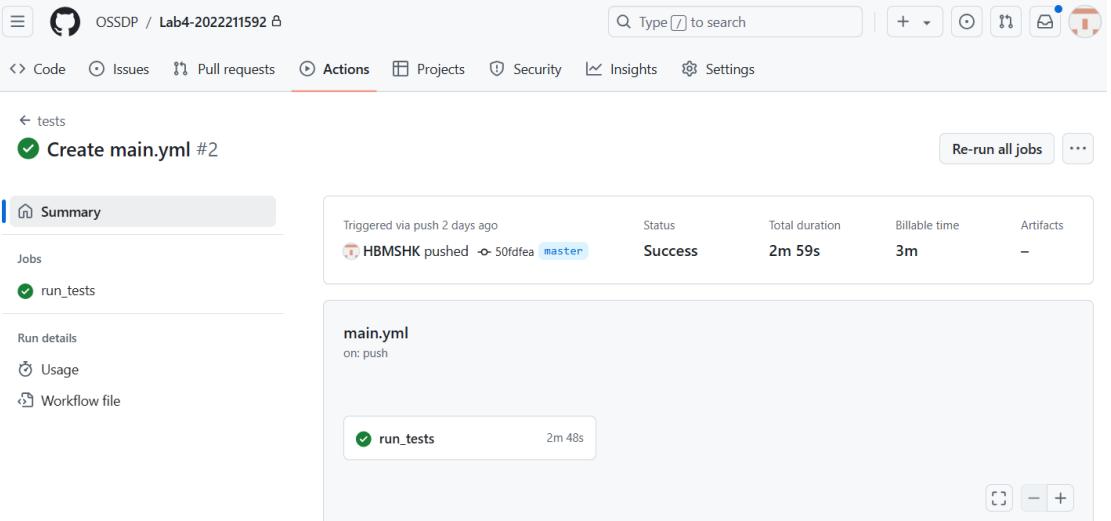
The screenshot shows the GitHub repository interface for the repository `OSSDP / Lab4-2022211592`. The user is viewing the `main.yml` file under the `.github/workflows` directory. The file content is as follows:

```

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@v2
9        - name: Set up JDK 17
10         uses: actions/setup-java@v1
11         with:
12           java-version: 17
13        - name: Cache Maven packages
14          uses: actions/cache@v2
15          with:
16            path: ~/.m2
17            key: ${{ runner.os }}-m2-${{ hashFiles('**/pom.xml') }}
18            restore-keys: ${{ runner.os }}-m2
19        - name: Run tests with Maven
20          run: mvn -B test --file pom.xml

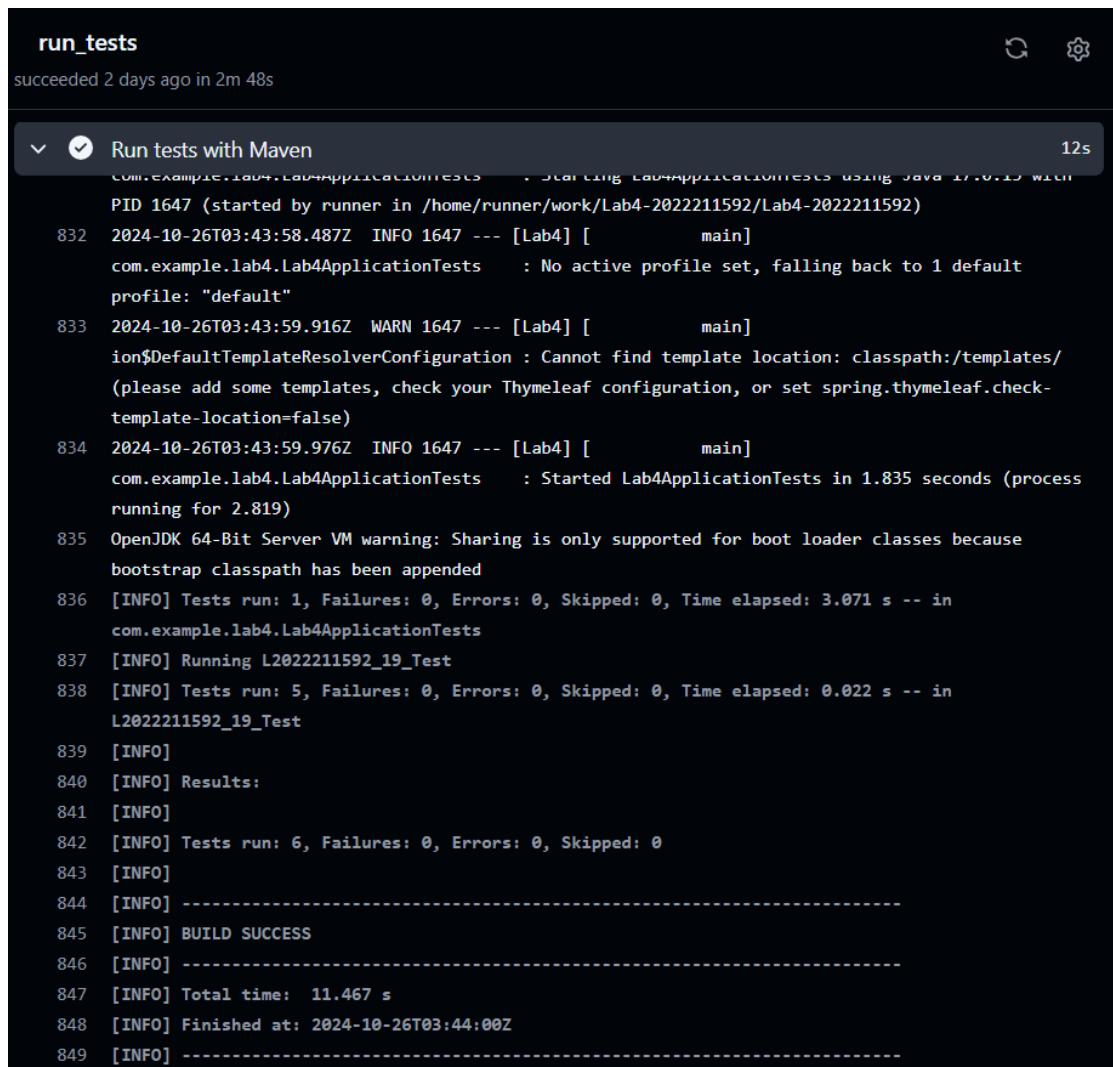
```

2、5 初次自动化测试



The screenshot shows the GitHub Actions summary for the workflow run `Create main.yml #2`. The run was triggered via push 2 days ago by the user `HBMSHK` on the `master` branch. The status is **Success**, total duration is **2m 59s**, and billable time is **3m**. The workflow file is `main.yml` and it has one job named `run_tests` which completed successfully in **2m 48s**.

Triggered via push 2 days ago	Status	Total duration	Billable time	Artifacts
<code>HBMSHK pushed → 50fdea master</code>	Success	2m 59s	3m	-



```

run_tests
succeeded 2 days ago in 2m 48s

Run tests with Maven
com.example.lab4.Lab4ApplicationTests . Starting Lab4ApplicationTests using Java 17.0.19 with
PID 1647 (started by runner in /home/runner/work/Lab4-2022211592/Lab4-2022211592)
832 2024-10-26T03:43:58.487Z INFO 1647 --- [Lab4] [           main]
com.example.lab4.Lab4ApplicationTests : No active profile set, falling back to 1 default
profile: "default"
833 2024-10-26T03:43:59.916Z  WARN 1647 --- [Lab4] [           main]
ion$DefaultTemplateResolverConfiguration : Cannot find template location: classpath:/templates/
(please add some templates, check your Thymeleaf configuration, or set spring.thymeleaf.check-
template-location=false)
834 2024-10-26T03:43:59.976Z  INFO 1647 --- [Lab4] [           main]
com.example.lab4.Lab4ApplicationTests : Started Lab4ApplicationTests in 1.835 seconds (process
running for 2.819)
835 OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because
bootstrap classpath has been appended
836 [INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 3.071 s -- in
com.example.lab4.Lab4ApplicationTests
837 [INFO] Running L2022211592_19_Test
838 [INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.022 s -- in
L2022211592_19_Test
839 [INFO]
840 [INFO] Results:
841 [INFO]
842 [INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0
843 [INFO]
844 [INFO] -----
845 [INFO] BUILD SUCCESS
846 [INFO] -----
847 [INFO] Total time: 11.467 s
848 [INFO] Finished at: 2024-10-26T03:44:00Z
849 [INFO] -----

```

2、6 修改代码后再次进行自动化测试

```

李鑫元@I-MINGW64 MINGW64 ~/Desktop/Lab4 (master)
$ git add .

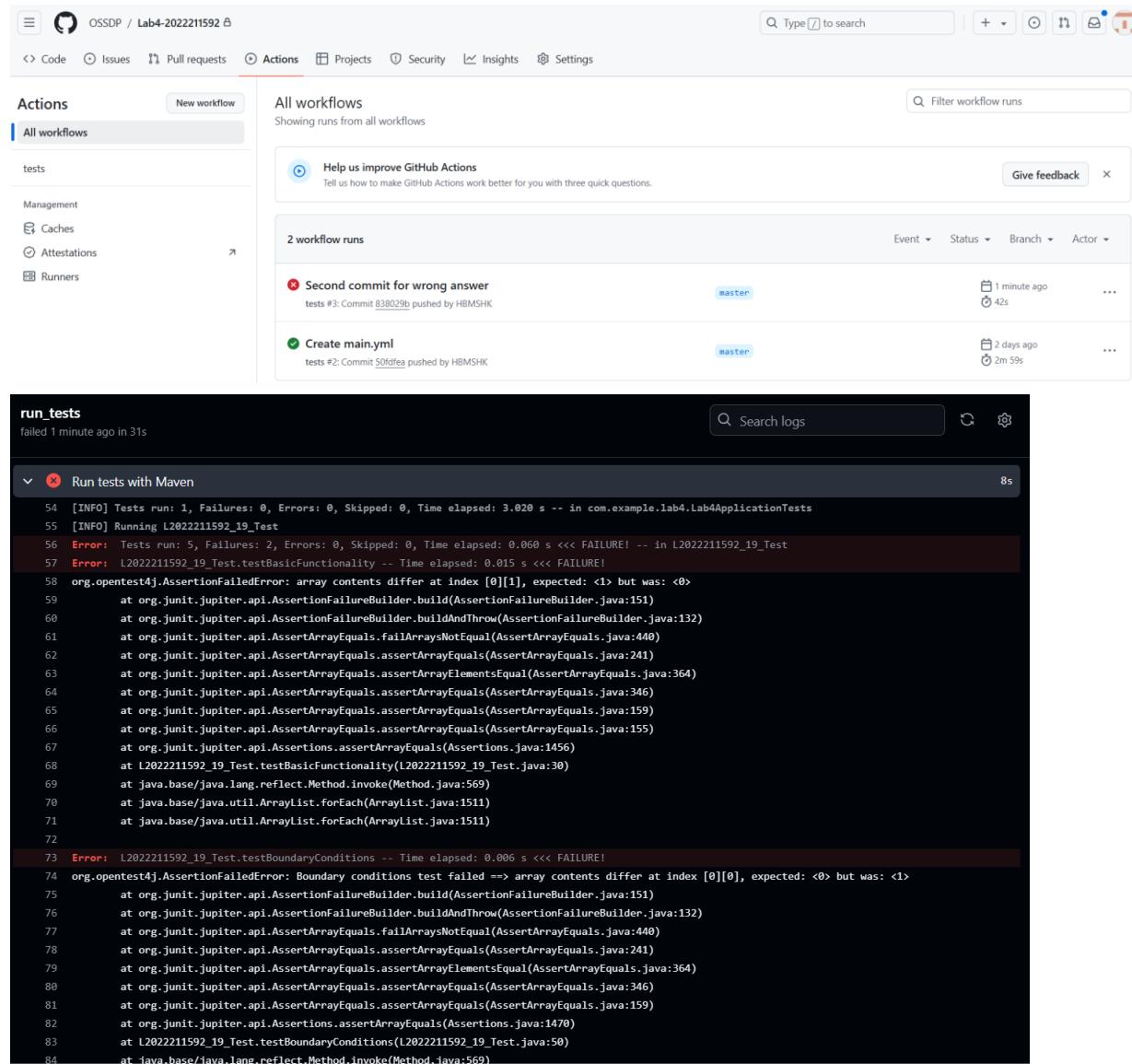
李鑫元@I-MINGW64 MINGW64 ~/Desktop/Lab4 (master)
$ git commit -m "Second commit for wrong answer"
[master 838029b] Second commit for wrong answer
 1 file changed, 5 insertions(+), 5 deletions(-)

李鑫元@I-MINGW64 MINGW64 ~/Desktop/Lab4 (master)
$ git status
On branch master
Your branch is ahead of 'origin4/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

李鑫元@I-MINGW64 MINGW64 ~/Desktop/Lab4 (master)
$ git push origin4 master
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 20 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 554 bytes | 554.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/OSSDP/Lab4-2022211592
  50fdfea..838029b  master -> master

```



3 实验内容 2 Jenkins DevOps 实践

1、Jenkins 安装与配置

(1) 更改国内镜像

在 jenkins 的工作目录 .jenkins 中，找到 hudson.model.UpdateCenter.xml 文件打开
将 <https://updates.jenkins.io/update-center.json> 替换成国内镜像网址(需要管理员权限修改)

国内镜像网址：<https://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/update-center.json>

再进入到 updates 目录下，编辑 default.json 文件，将该文件中国外的地址全部替
换成国内的（需要管理员权限修改）

<https://www.google.com> 替换成 <https://www.baidu.com>

<https://updates.jenkins.io/download> 替换成 <https://mirrors.tuna.tsinghua.edu.cn/jenkins>

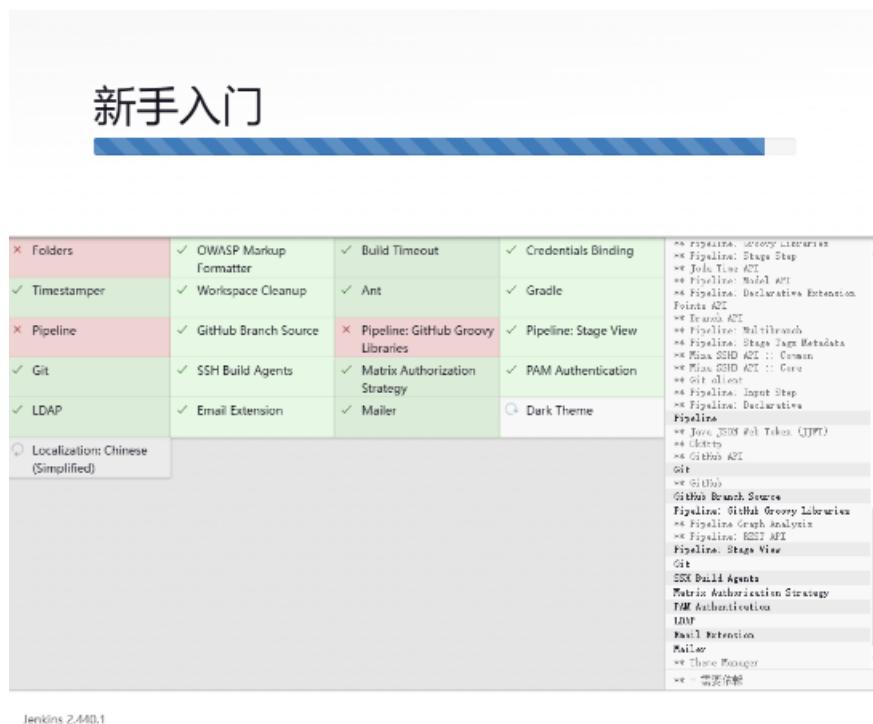
(2) 更改工作目录

```
<service>
  <id>jenkins</id>
  <name>Jenkins</name>
  <description>This service runs Jenkins automation server.</description>
  <env name="JENKINS_HOME" value="%ProgramData%\Jenkins\jenkins"/>
<!--
  if you'd like to run Jenkins with a specific version, put it here.
  The following value assumes that you have java in your PATH.
-->
```

将该路径修改为目标路径，例如: D:\jenkins\windows\jenkins-2.440-work

(3) 安装插件

新手入门

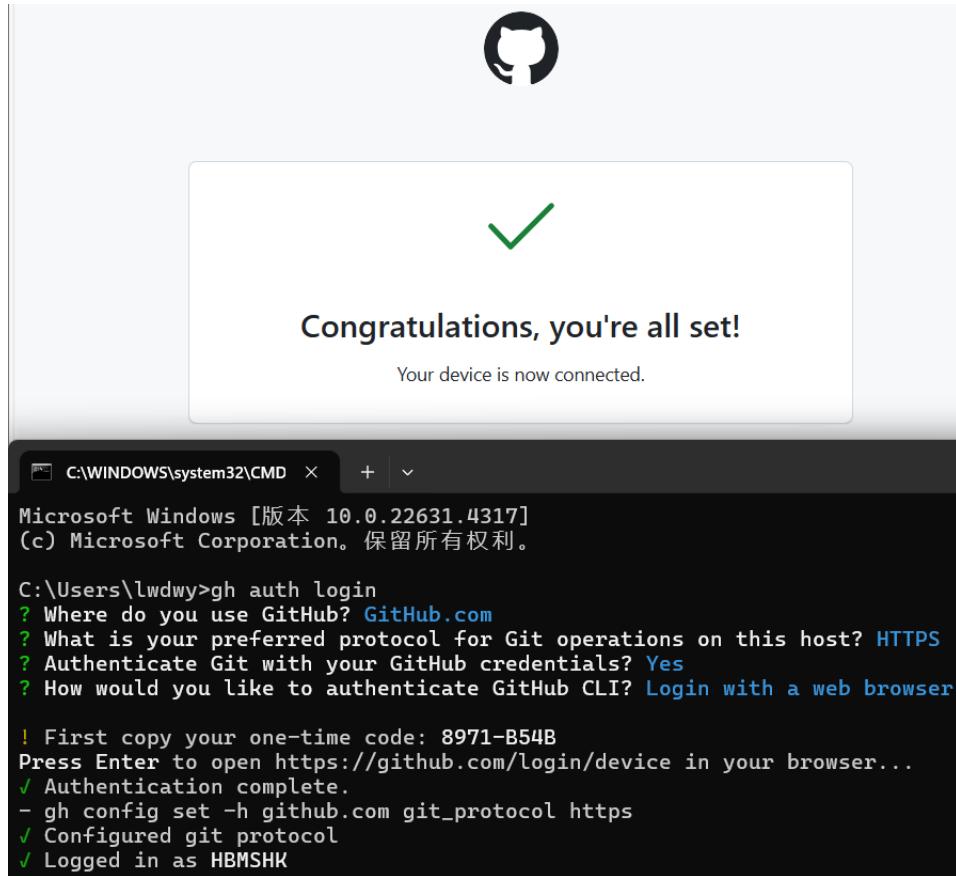


Jenkins 2.440.1

(4) 创建非管理员账号

Role permission	Overall	Agent	Job	Run	View
Role	Read	Build	Create	Configure	Delete
Role: admin	<input checked="" type="checkbox"/>				
admin	<input checked="" type="checkbox"/>				
ff	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

2、安装 Github CLI



New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

What's this token for?

Expiration *

The token will expire on Sun, Dec 8 2024

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input checked="" type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input checked="" type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects

3、构建 DevOps workflow

(1) General 部分

(2) 源码管理

Git ?

Repositories ?

Repository URL ?

git@github.com:OSSDP/Lab4-2022211592.git

Credentials ?

HBMSHK (Github SSH 私钥)

+ 添加

高级 ▾

Add Repository

Branches to build ?

指定分支 (为空时代表any) ?

*/master

(3) 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 ?

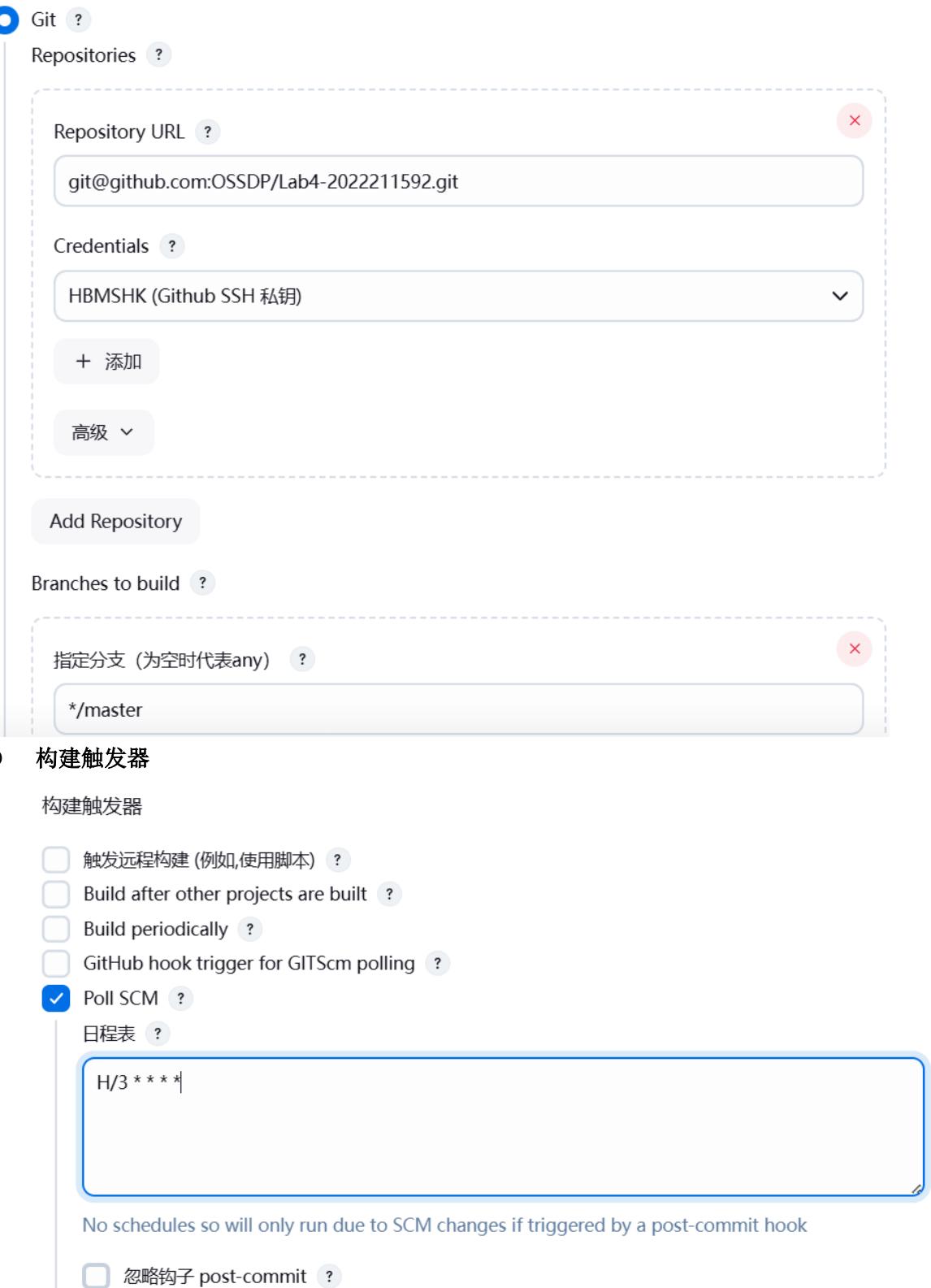
Poll SCM ?

日程表 ?

H/3 * * * *

No schedules so will only run due to SCM changes if triggered by a post-commit hook

忽略钩子 post-commit ?



(4) 构建步骤

Build Steps

The screenshot shows two build step configurations in Jenkins:

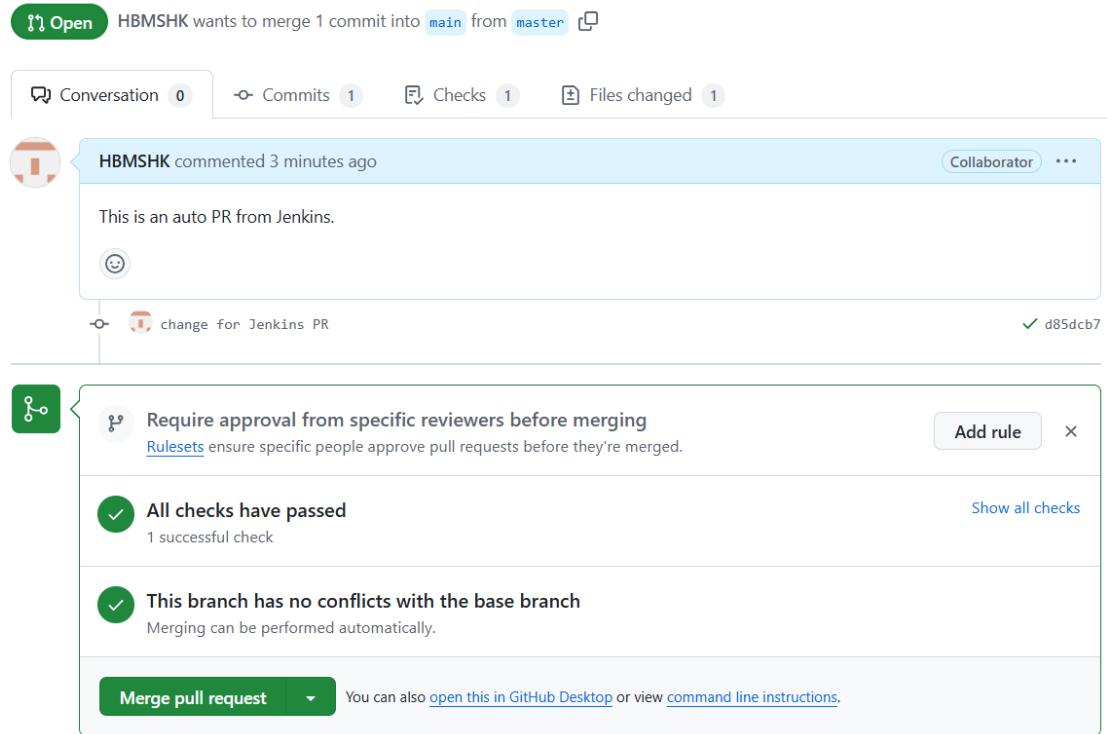
- Execute Windows batch command**:
 - 命令: mvn validate
 - 参阅 [可用环境变量列表](#)
 - 高级 ▾
- Execute Windows batch command**:
 - 命令: mvn test
 - 参阅 [可用环境变量列表](#)
 - 高级 ▾

(5) 验证实验结果

A Jenkins job run history entry for #11 (2024年11月10日下午 12:14:46):

- 启动用户** 李鑫元
- Started** 1 min 24 sec ago
- Took** 32 sec
- This run spent:**
 - 8 ms waiting;
 - 32 sec build duration;
 - 32 sec total from scheduled to completion.
- Revision:** d85dcb7cdb180cc911b7de88351d7f0648761f57
- Repository:** git@github.com:OSSDP/Lab4-2022211592.git
 - refs/remotes/origin/master
- Changes**
 - 1. change for Jenkins PR ([details](#) / [githubweb](#))

Auto PR from master-Jenkins to main #1



4 小结

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 和 Jenkins 进行 DevOps。

(1) Github Actions 实践总结:

项目目录结构: 了解了项目的目录结构，为后续操作提供了基础。

测试用例运行结果: 查看了项目测试用例的运行结果，确保了代码的正确性。

Maven 项目推送: 成功将 Maven 项目推送到 GitHub 仓库，为 DevOps 实践提供了平台。

Workflow 文件编写: 在 GitHub 上编写了 DevOps workflow 文件，实现了自动化构建和测试。

自动化测试: 完成了初次自动化测试，并修改了代码后再次进行了自动化测试，验证了 DevOps 流程的有效性。

(2) Jenkins 实践总结:

安装与配置: 详细描述了 Jenkins 的安装与配置过程，包括更改国内镜像、更改工作目录、安装插件等。

非管理员账号创建: 创建了非管理员账号，提高了系统的安全性。

Github CLI 安装: 安装了 Github CLI，为 Jenkins 与 GitHub 的集成提供了便利。

DevOps Workflow 构建: 在 Jenkins 中构建了 DevOps workflow，包括源码管理、构建触发器、构建步骤等。

实验结果验证: 验证了实验结果，确保了 Jenkins DevOps 实践的顺利进行。