**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** |
| 2022211972 | 吴俣辰 | 19955918221 |

**目 录**

[1 实验要求 1](#_Toc1529164088)

[2 实验内容1 Github Actions DevOps实践 1](#_Toc1724023367)

[3 实验内容2 Jenkins DevOps实践 1](#_Toc1811363845)

[4 小结 1](#_Toc863963043)

[文档全部完成之后，请更新上述区域]

# 实验要求

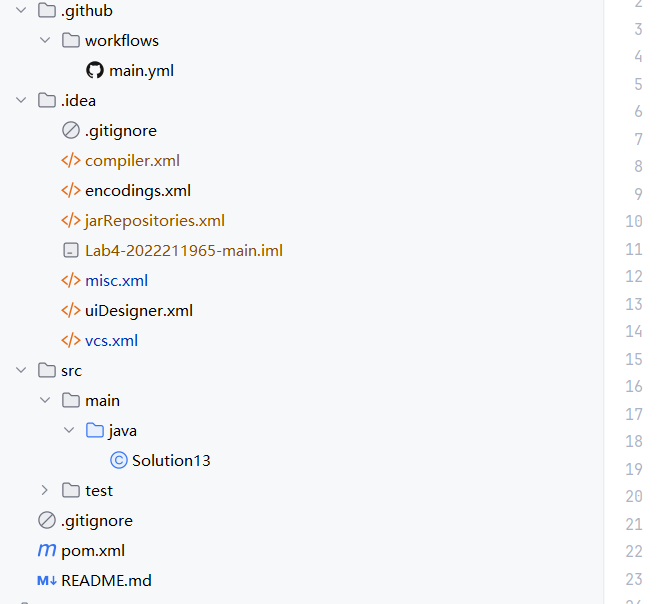
本次实验训练开源软件开发中的基本 DevOps 操作，具体来说：

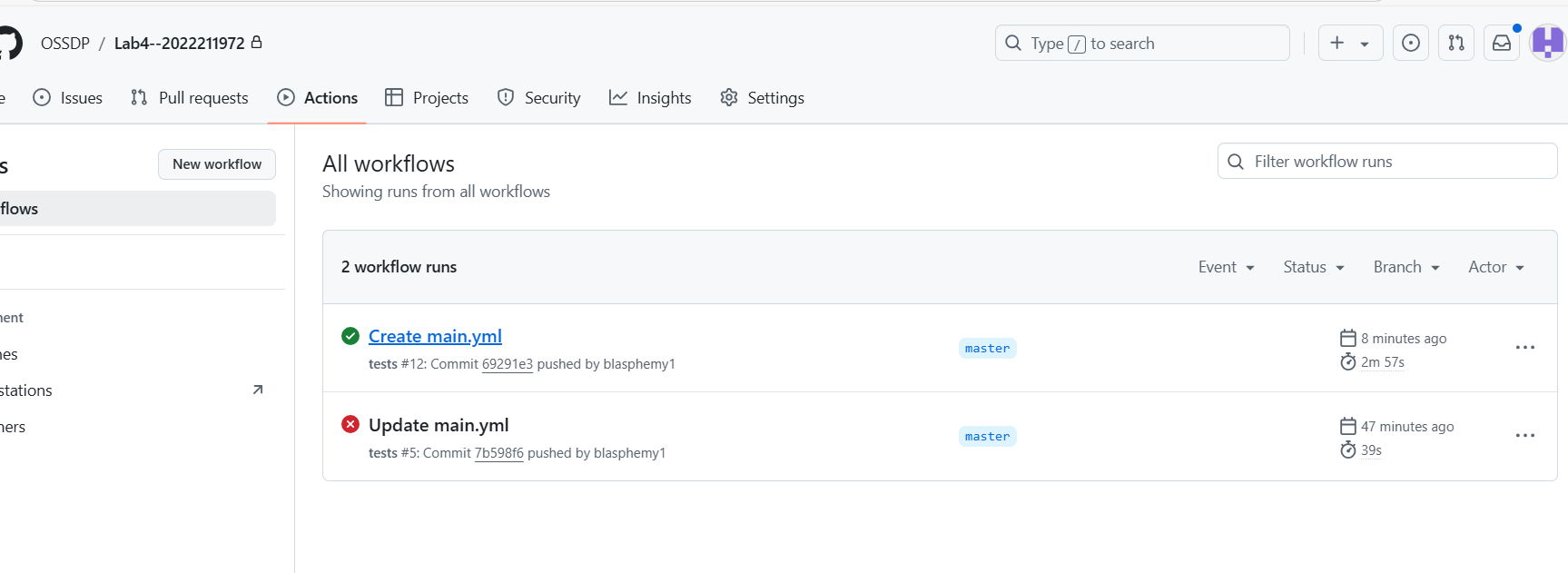
⚫ 掌握开源软件开发中的基本 DevOps 流程和工具的使用

⚫ 熟悉利用 Github Actions 进行 DevOps

⚫ 熟悉利用 Jenkins 进行 DevOps

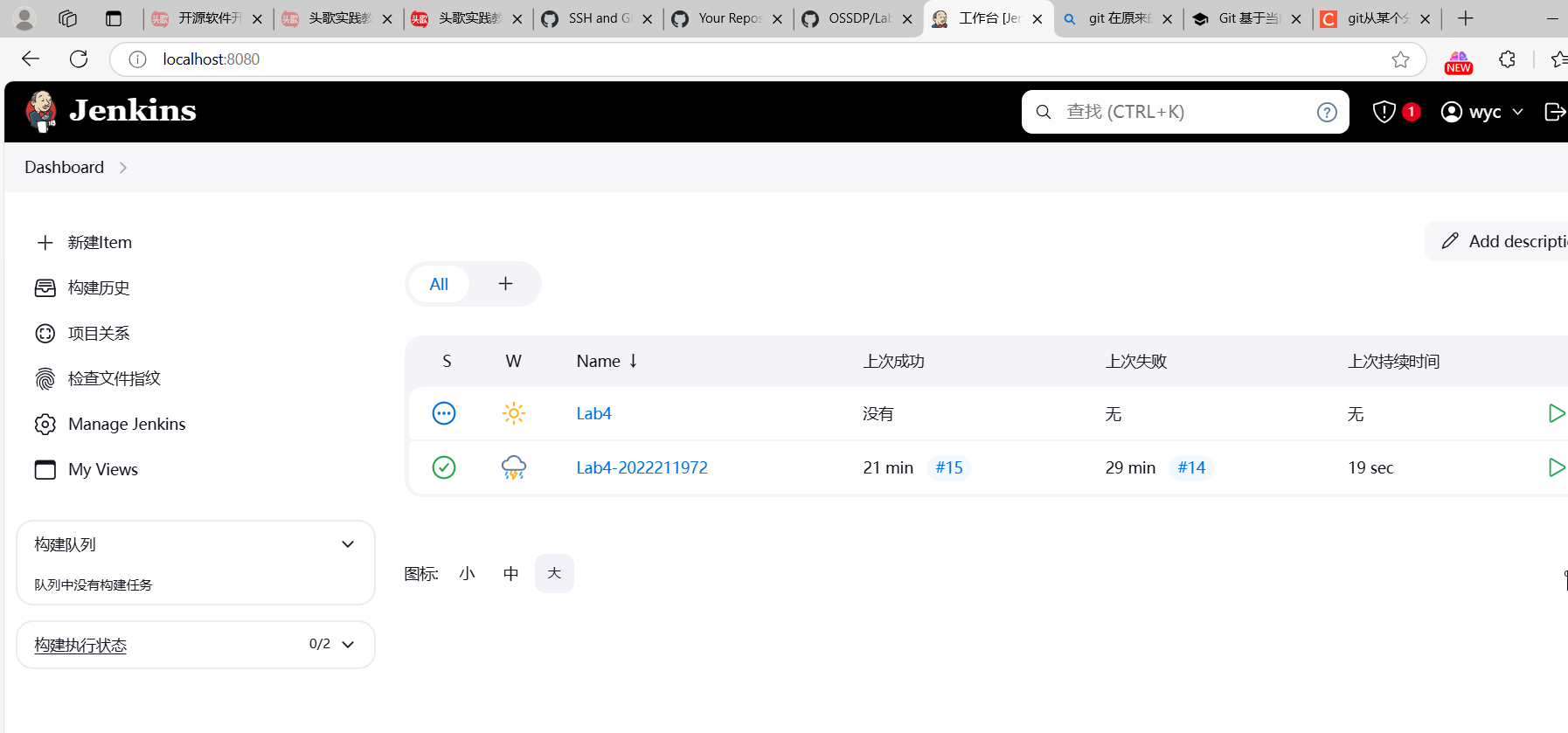
# 实验内容1 Github Actions DevOps实践



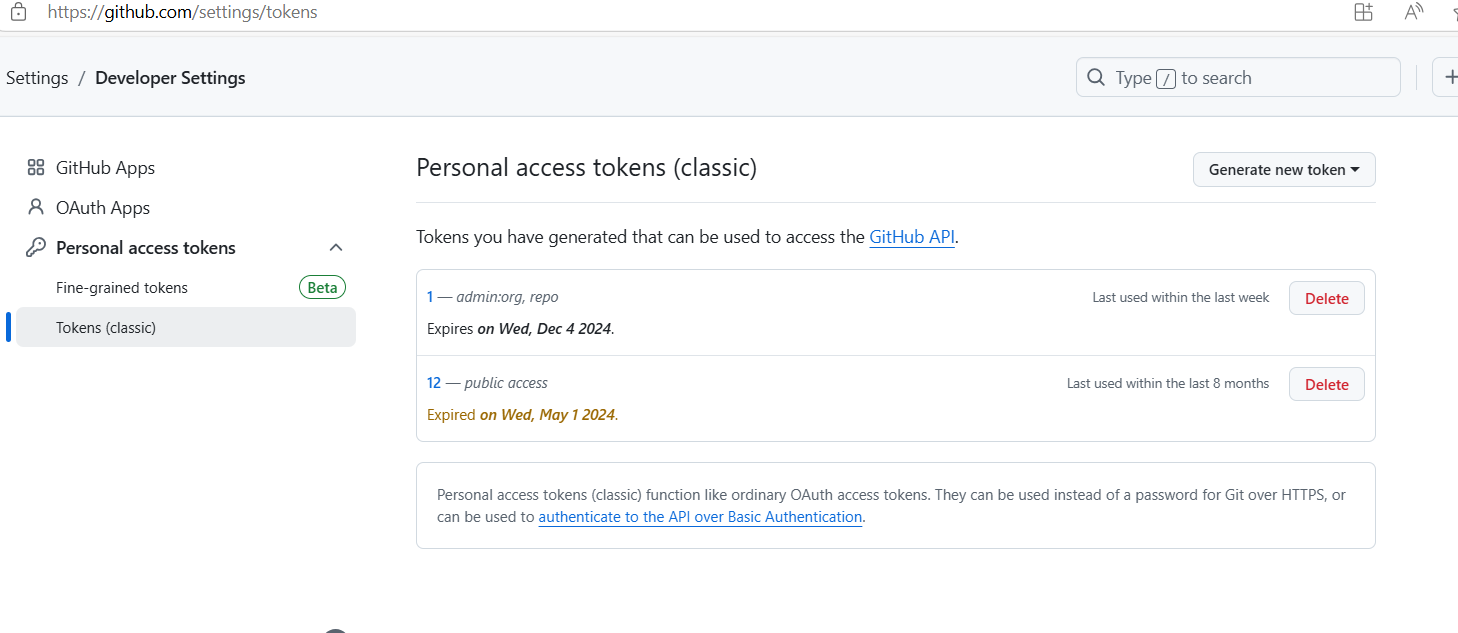


# 实验内容2 Jenkins DevOps实践

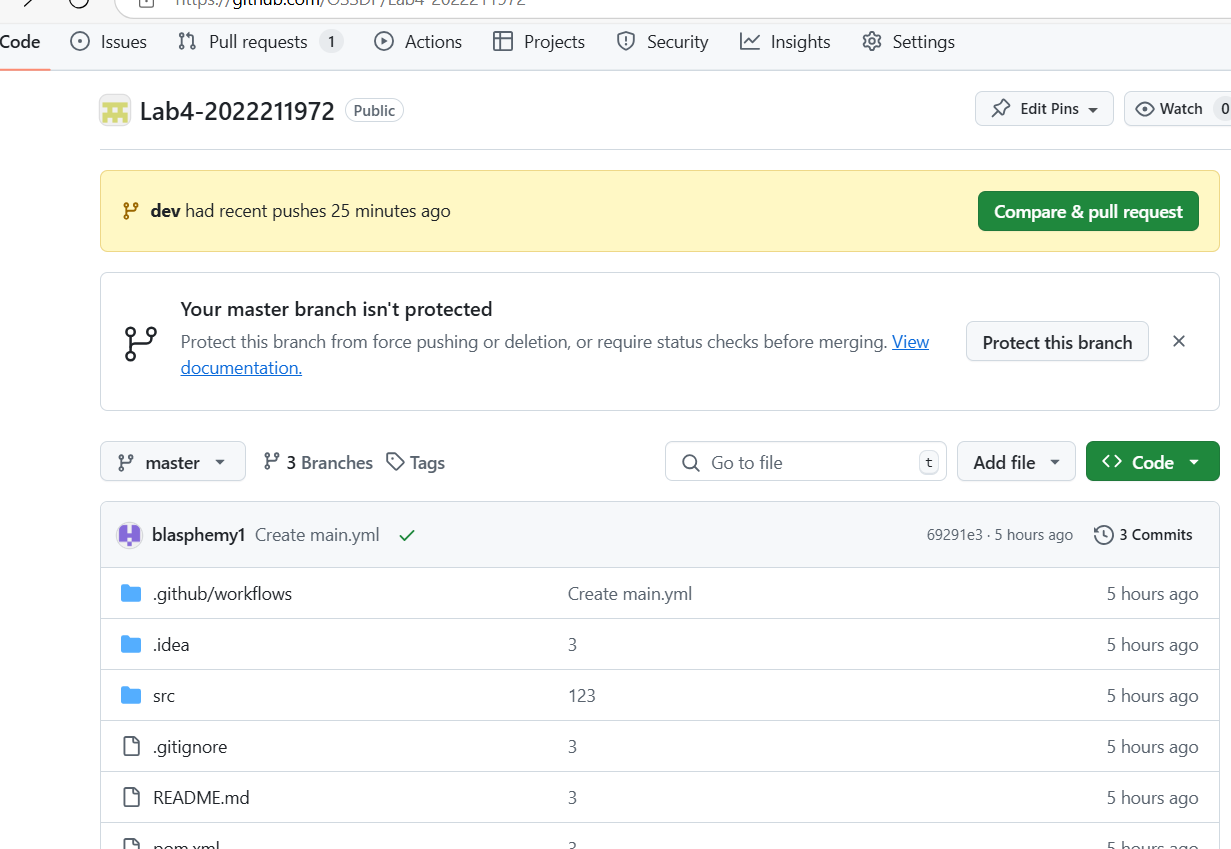
Jenkins安装与配置



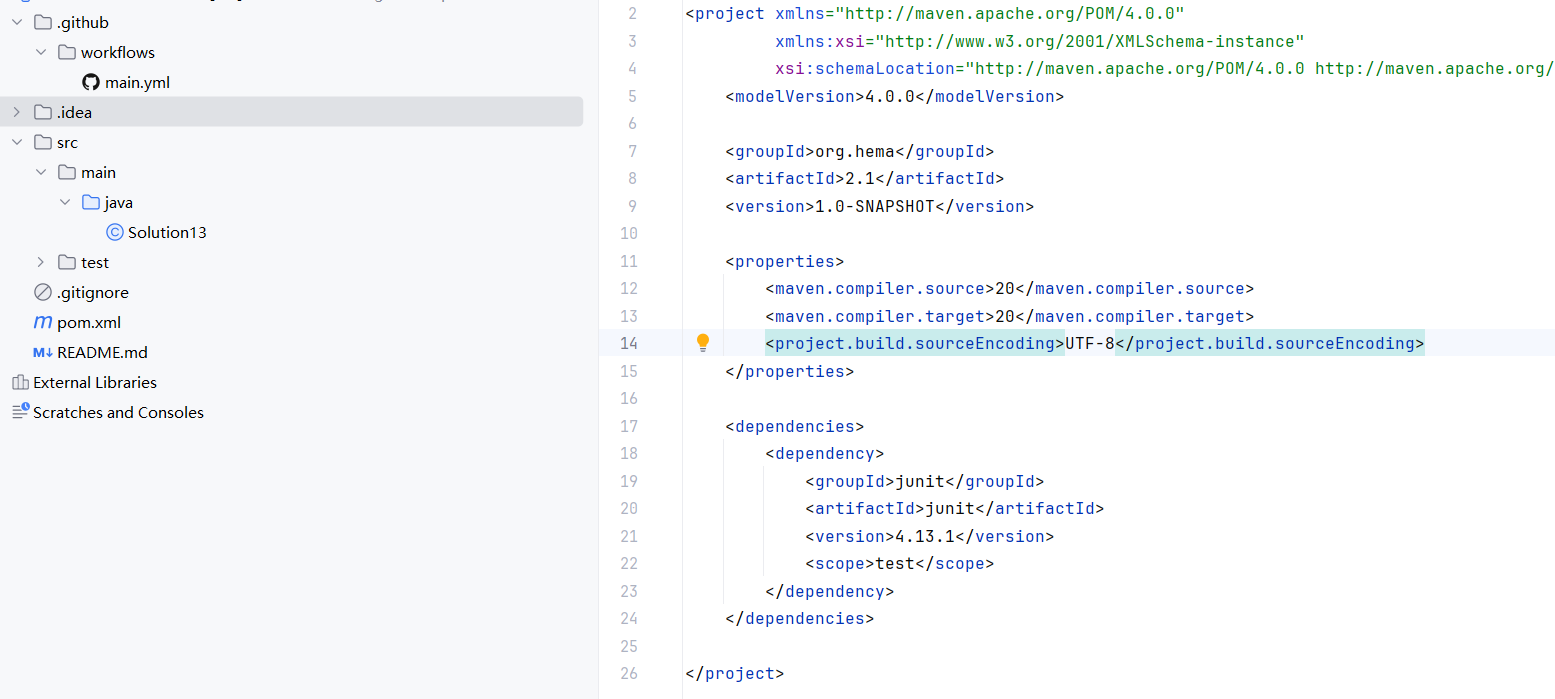
Github CLI 注意事项



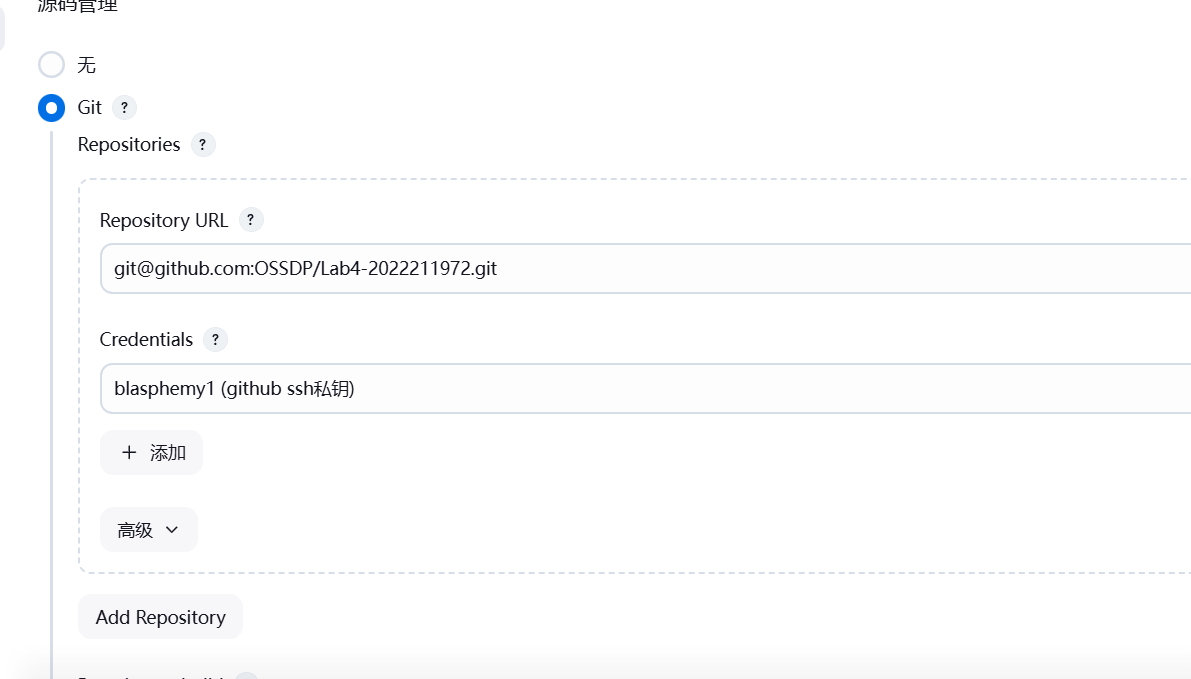
（四）测试用github仓库

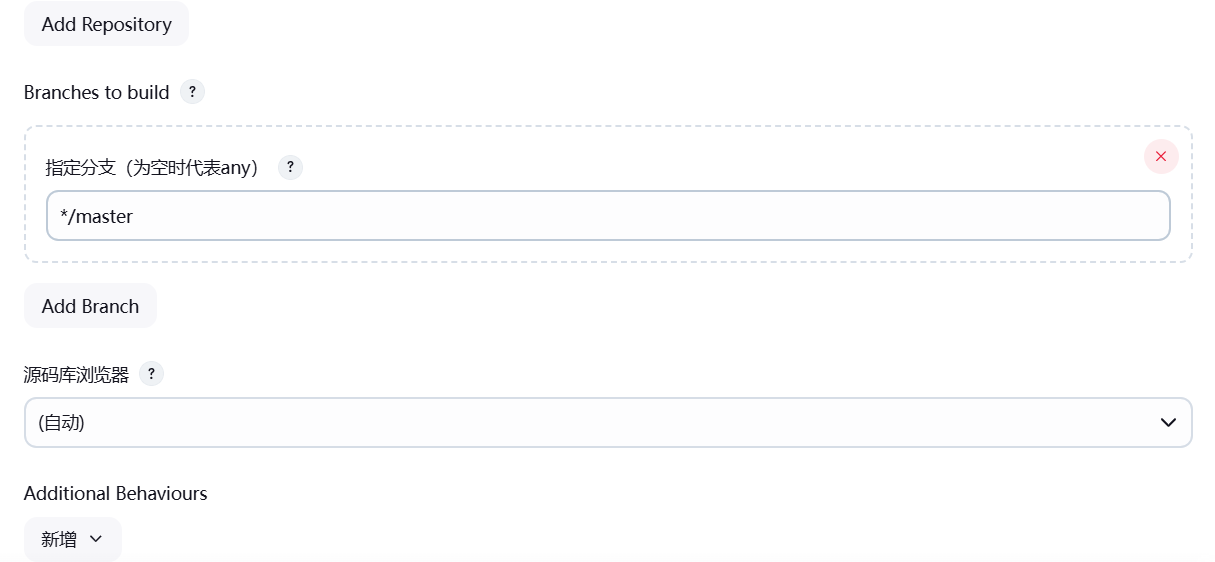


（五）本地创建maven项目并提交

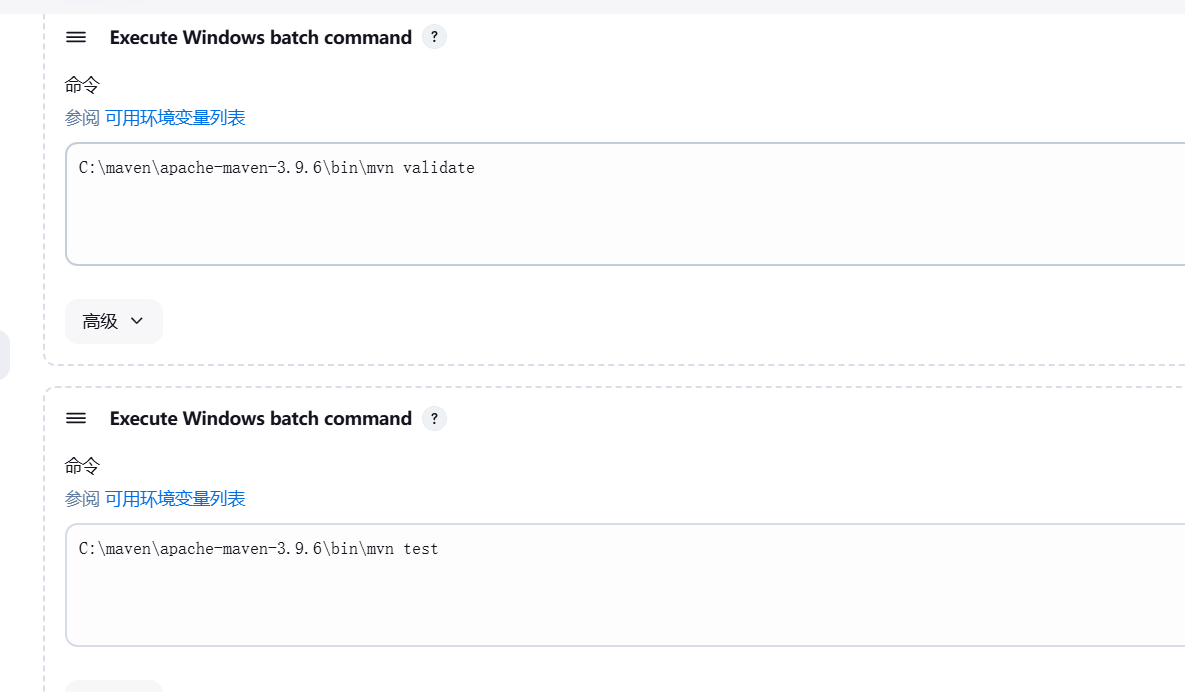


（六）构建DevOps workflow

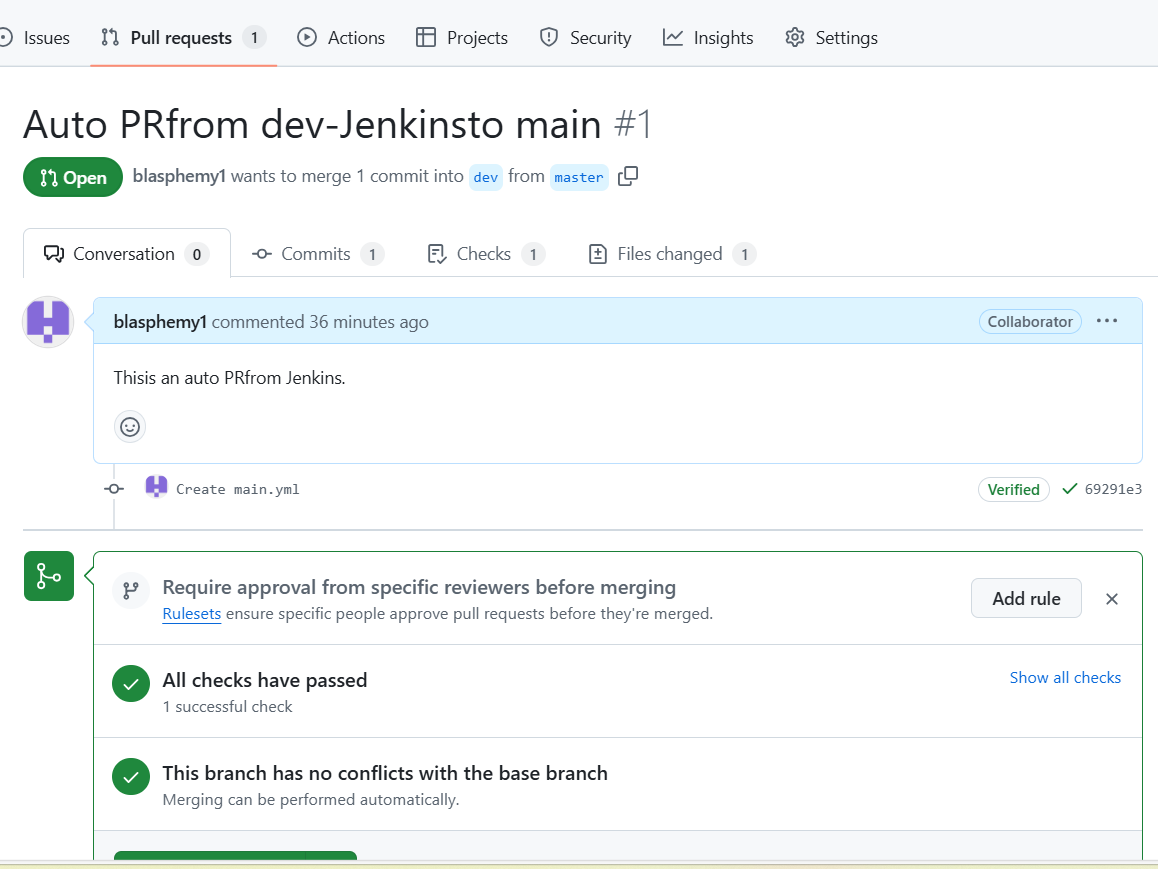
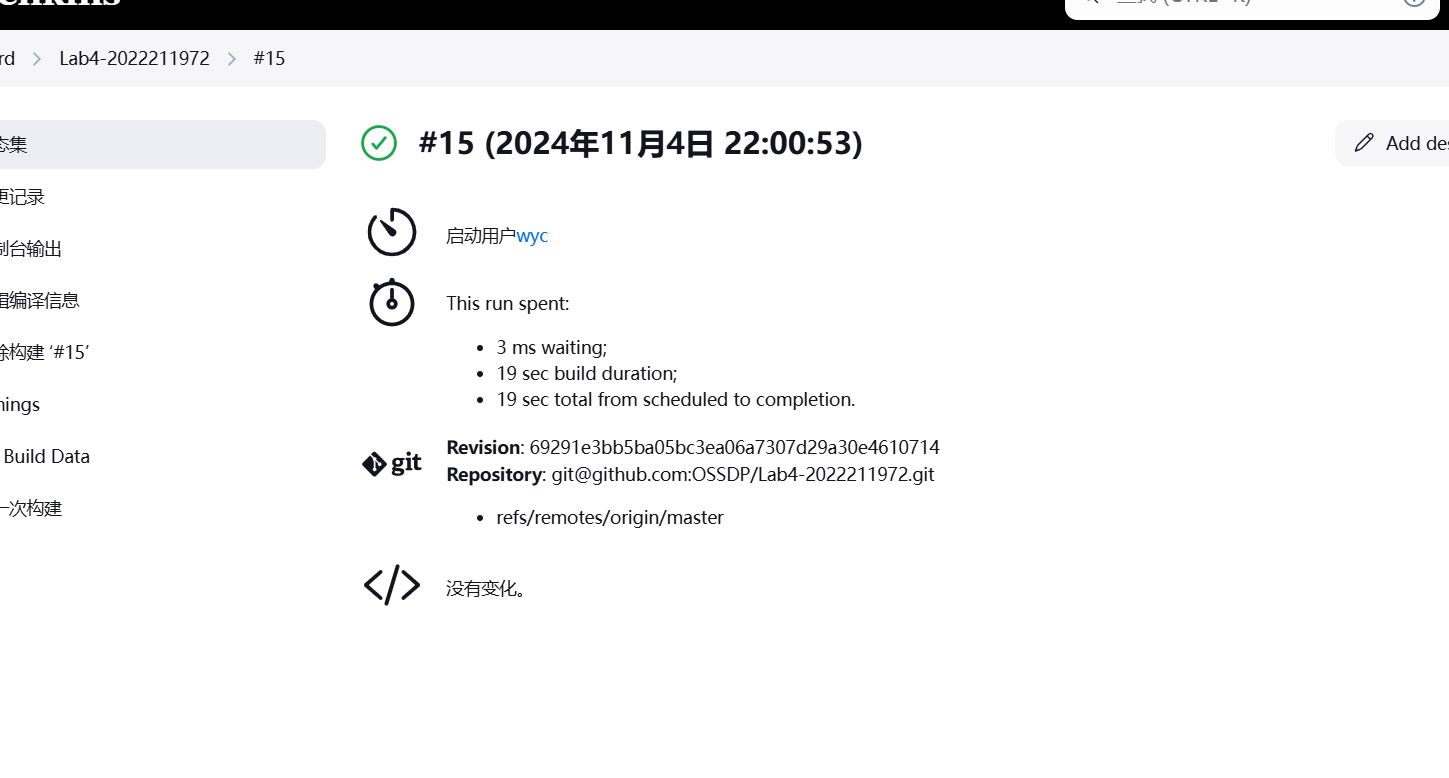












# 小结

在当今软件开发领域，持续集成和持续交付（CI/CD）已成为提高开发效率和软件质量的重要实践。Github Actions和Jenkins是实现这一目标的两种流行工具，各自拥有独特的优势和应用场景。

首先，Github Actions是GitHub平台原生提供的一项功能，允许开发者在代码库中直接定义自动化工作流。这种整合性使得开发者能够轻松地基于代码库的事件，如推送代码、创建拉取请求等，触发一系列操作。此外，Github Actions支持灵活的配置，用户可以通过简单的YAML文件设定工作流，极大地降低了入门门槛。

与此相比，Jenkins作为一款开源的自动化服务器，在功能和扩展性方面具备强大的优势。Jenkins拥有丰富的插件生态系统，能够集成多种第三方工具与服务，为复杂的构建和部署需求提供全面支持。然而，它的配置和维护相对较为复杂，尤其是在大规模项目中，用户需投入更多精力以确保系统的稳定性和安全性。

在DevOps的理念下，选择Github Actions或Jenkins往往取决于团队的具体需求和技术栈。如果团队已经在使用GitHub进行代码托管，Github Actions能迅速上手并且与现有流程无缝对接。而对于需要复杂构建过程或多种集成的场景，Jenkins则可能是更为适宜的选择，尽管其学习曲线较陡。

总结而言，Github Actions与Jenkins各具特色，开发团队应根据项目需求、团队规模及技术熟悉度选择合适的工具。只要有效地利用这些工具，便能在DevOps实践中实现流畅的开发与交付过程，最终提升软件的质量和交付速度。