**开源操作系统**

**实验报告**

专业班级 软件工程2306班

学 号 8209230614

姓 名 李兴

日 期 2025/5/28

**实验成绩**

**批阅教师**

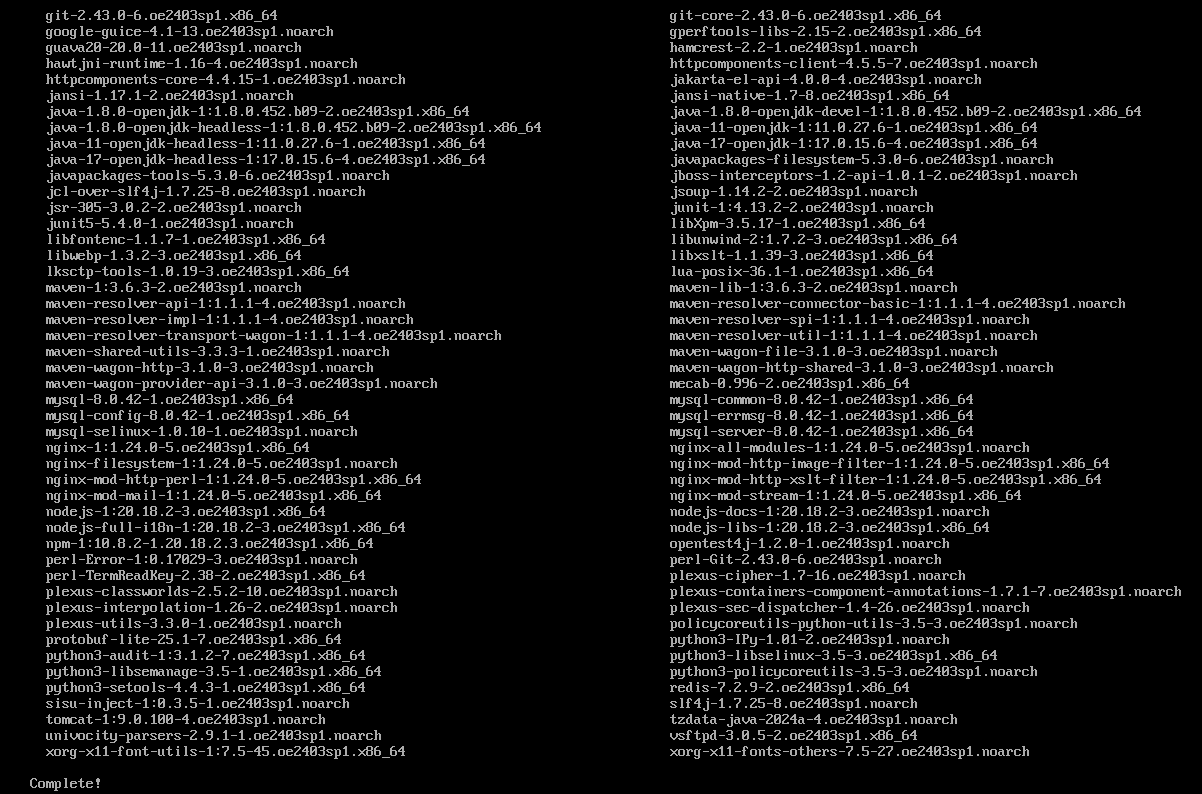
**批阅日期**

***中南大学计算机学院*实验二 搭建和配置Web服务器相关环境**

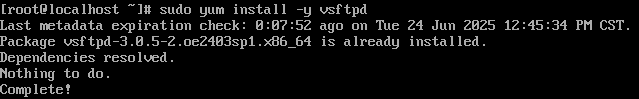
**一、实验内容**

1. **基于Linux服务器的前后端分离项目的部署**
2. 安装和配置前后端技术栈相关软件，包括Java、Node.js、Git、Maven、Tomcat、Nginx、MySQL、Redis等软件和环境；

sudo yum install -y java-17-openjdk nodejs git maven tomcat nginx mysql-server redis vsftpd



1. 搭建ftp文件服务器环境，使用vsftp或者sftp服务，将图片、文档等静态资源上传到服务器上；

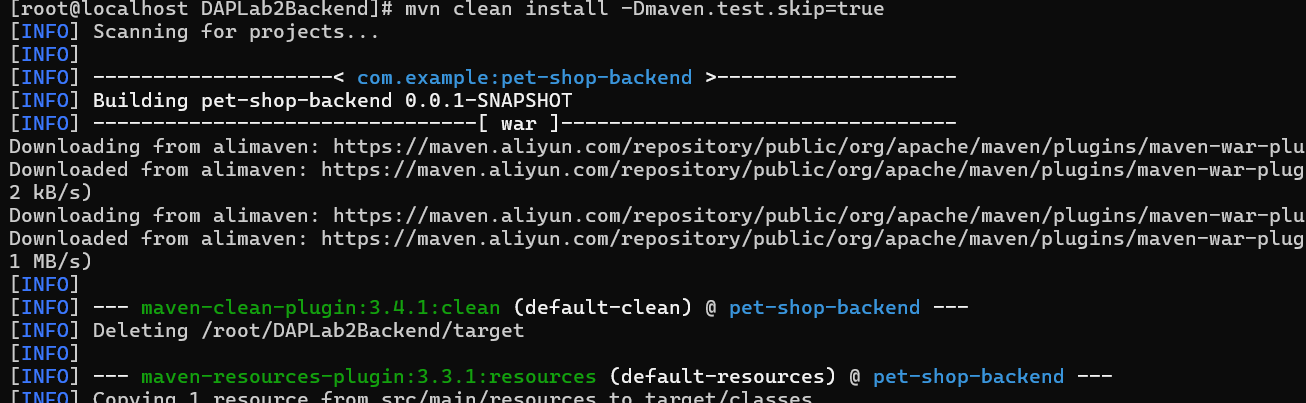


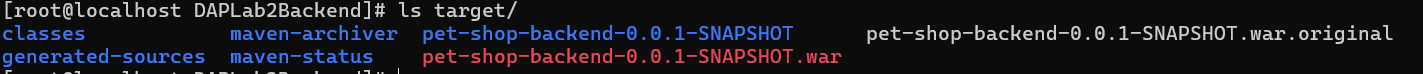


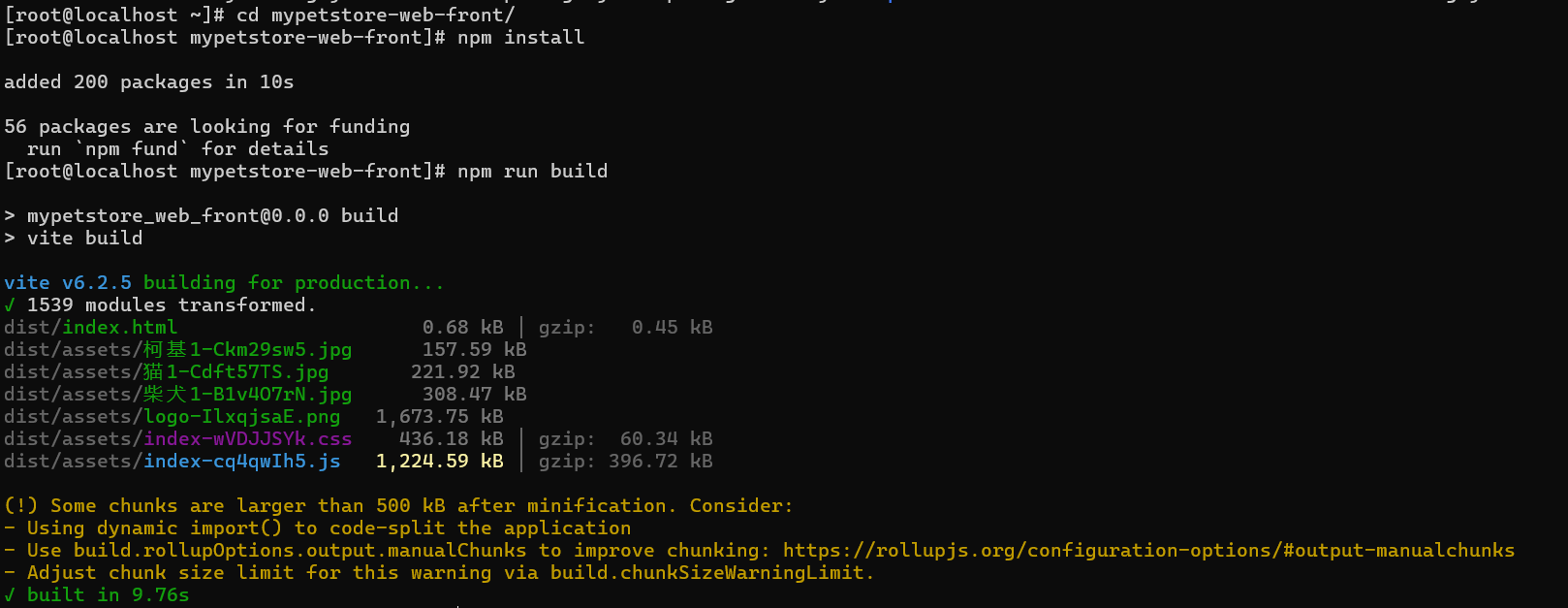
1. 使用Git clone或pull代码到服务器上，运行数据库脚本构建MySQL服务器；

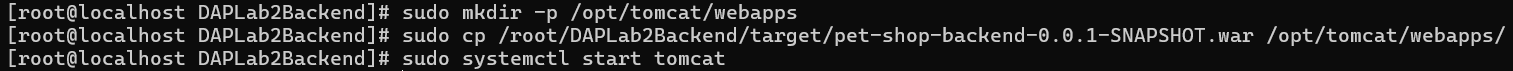


1. 使用Maven和NPM分别构建后端和前端项目，其中后端项目构建为war包；

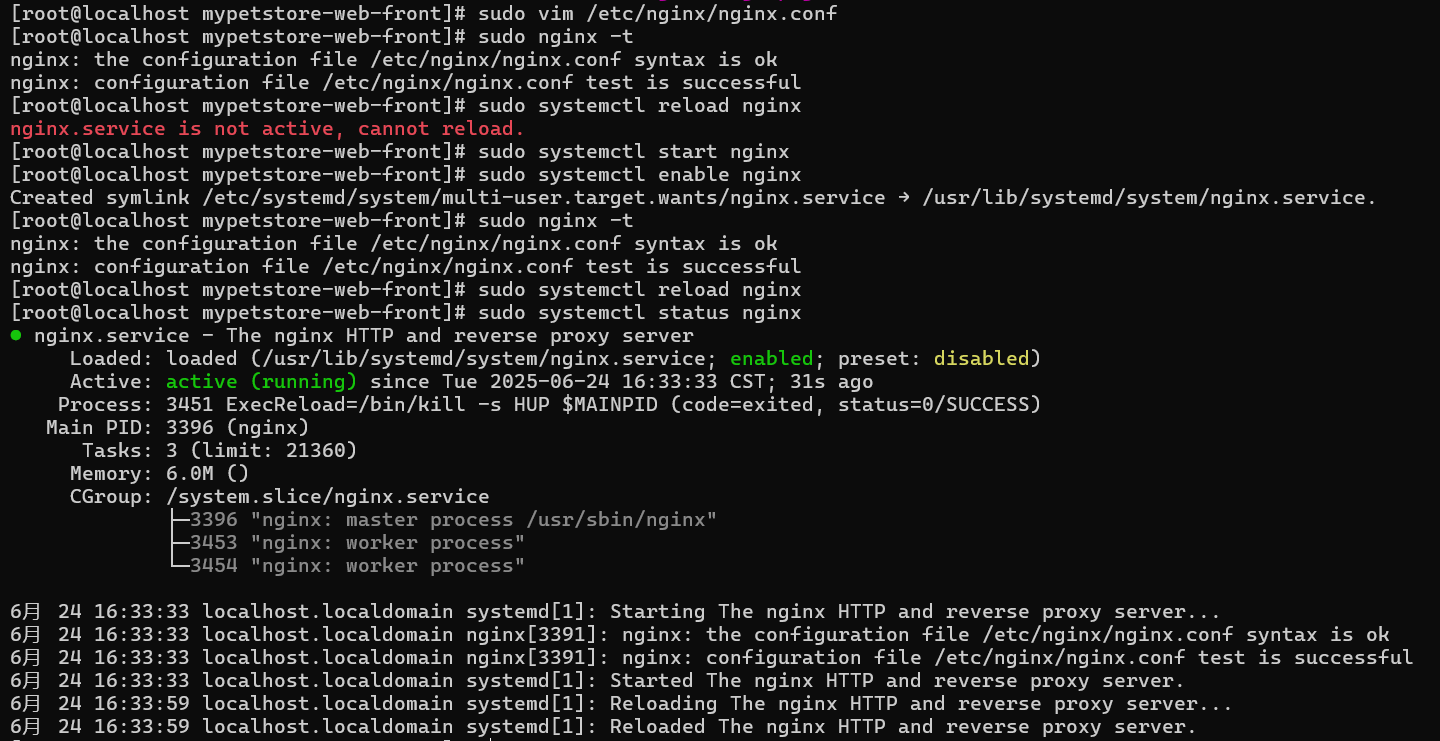


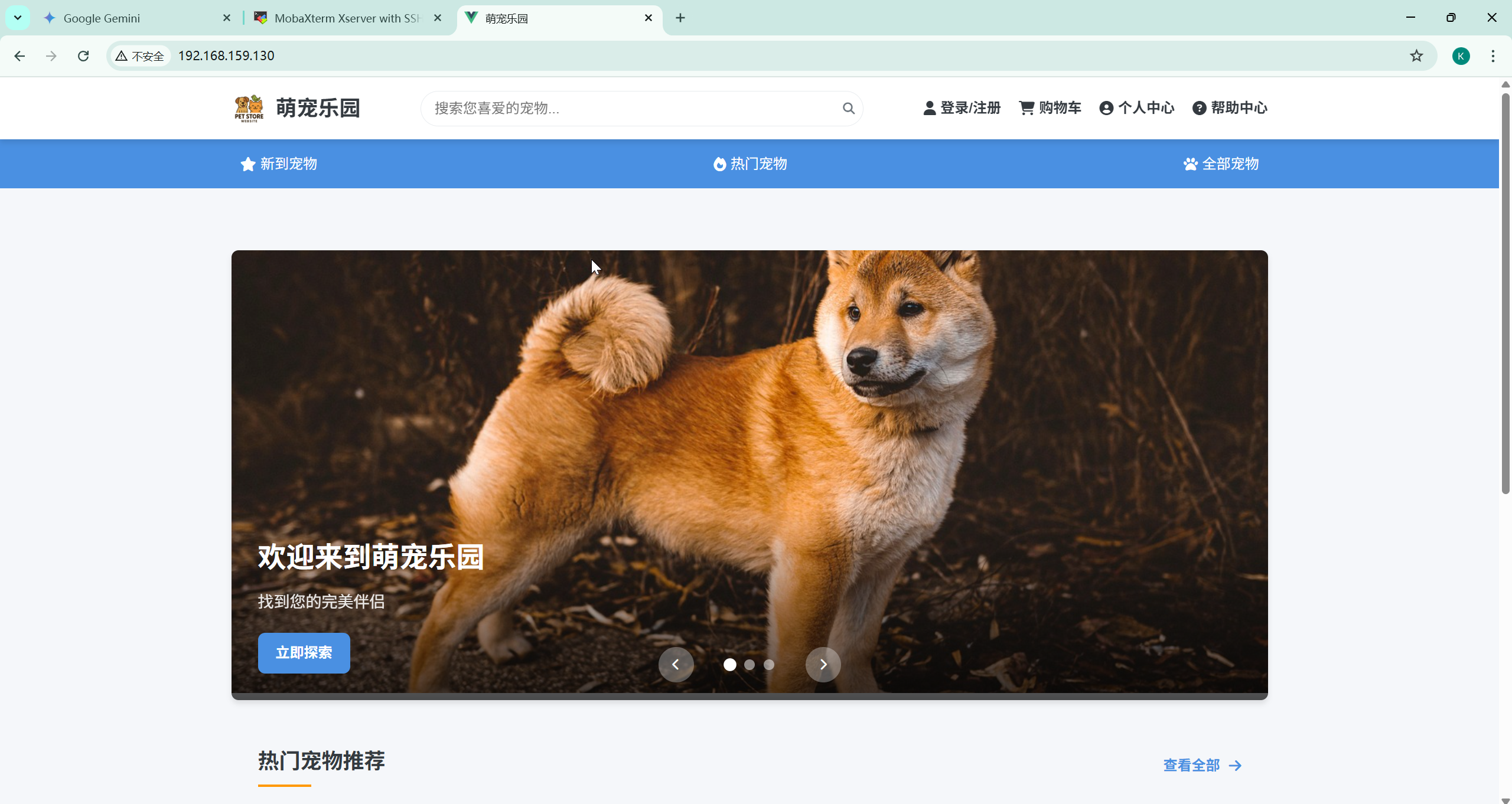




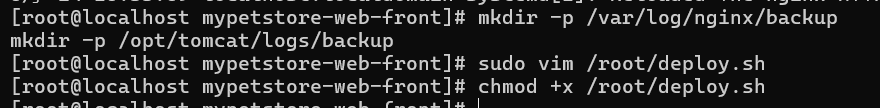


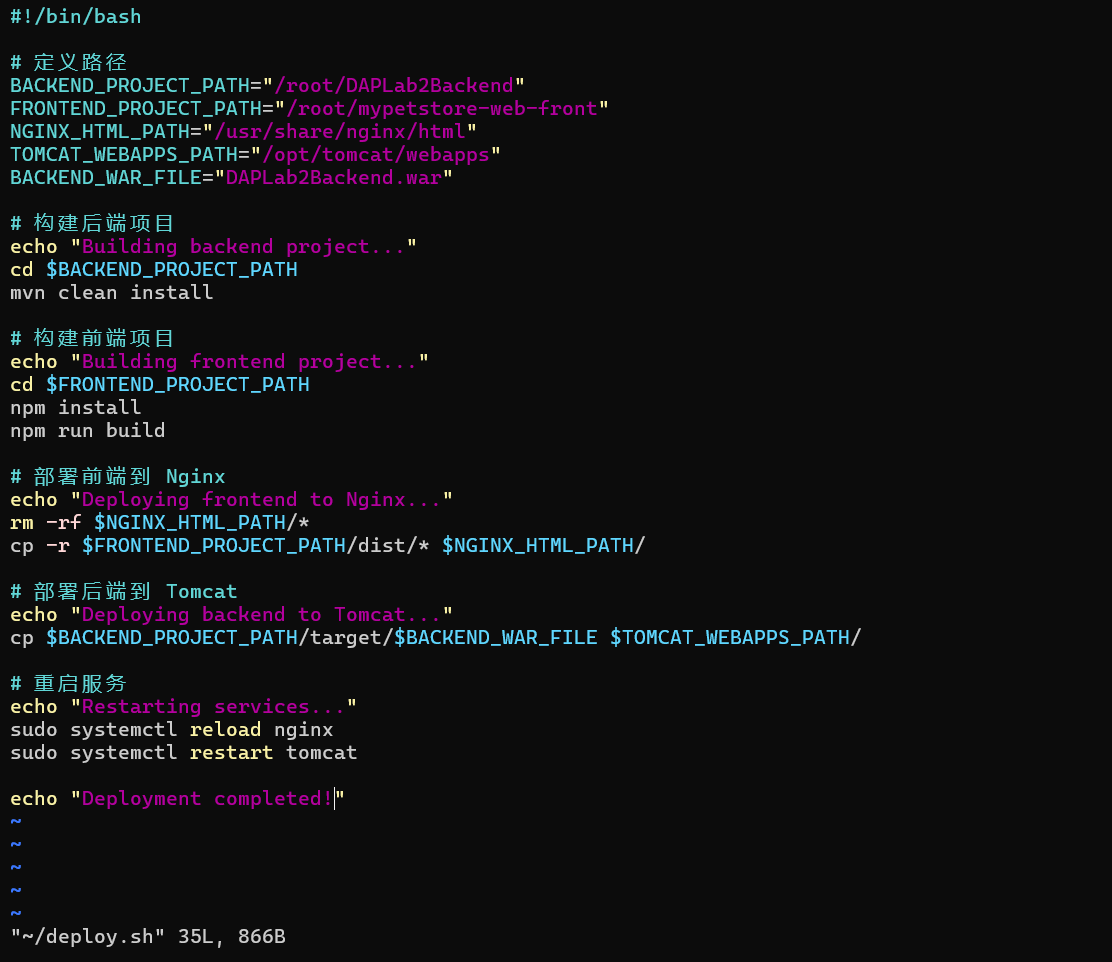
1. 将前端部署到Nginx中运行，后端项目部署到tomcat中运行，然后将后端请求通过Nginx反向代理到tomcat，完成前后端项目的部署；



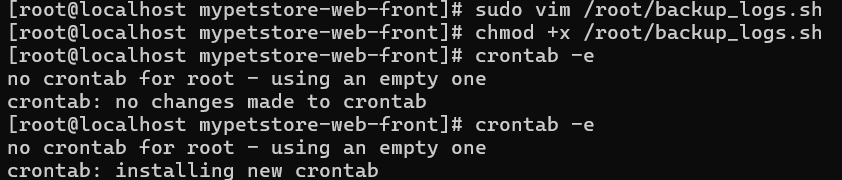


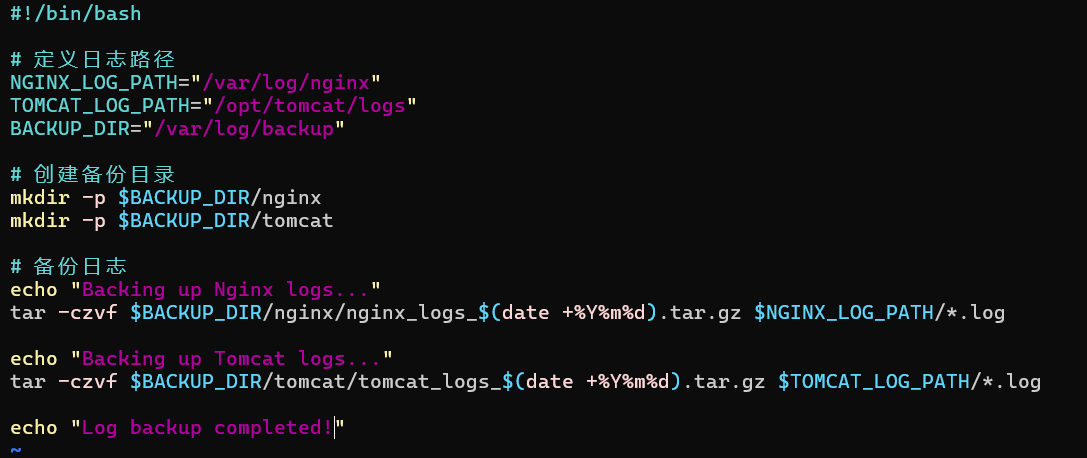
1. 配置相关服务器的日志文件目录；
2. 将上述部署归纳合并到一个脚本文件中，实现自动化部署；





1. 编写计划任务，对日志文件进行备份和分析。







**二、心得体会**

本次实验围绕基于Linux服务器的前后端分离项目部署展开，涵盖了从环境搭建、服务配置到自动化部署的完整流程。通过这次综合性的实验，我不仅掌握了现代Web应用部署的核心技术，更深入理解了企业级项目部署的复杂性和系统性。

在实验初期，我系统地学习了Java、Node.js、Git、Maven、Tomcat、Nginx、MySQL、Redis等核心技术栈的安装和配置。这个过程让我深刻认识到每个组件在整个系统架构中都承担着不可替代的职责：Java和Maven构成了后端项目的开发和构建基础，Node.js和NPM为前端项目提供了完整的开发生态，Nginx既是Web服务器又是反向代理的核心，MySQL负责数据的持久化存储，Redis则提供了高性能的缓存支持。在实际配置过程中，我遇到了端口冲突、权限设置、版本兼容等各种问题，通过逐一解决这些问题，我对Linux系统管理有了更加深入和实用的理解。

搭建FTP文件服务器环境的过程让我学会了使用vsftpd和SFTP服务来管理静态资源。在配置FTP服务器时，我不仅掌握了基本的文件传输协议，还深入理解了用户权限管理、目录访问控制等安全机制。处理图片、文档等静态资源的上传和管理过程，让我认识到合理的文件组织结构对项目维护的重要性，这为后续的资源管理打下了良好基础。

使用Git进行代码管理是现代开发不可缺少的技能。通过Git clone和pull操作，我不仅熟练掌握了版本控制的基本操作，还深入理解了分支管理、代码合并等高级功能在团队协作中的重要作用。运行数据库脚本构建MySQL服务器的过程中，我学会了数据库的初始化、用户权限设置、数据导入等关键操作，这些技能为应用的数据层部署提供了坚实保障。

项目构建阶段是整个部署流程的核心环节。使用Maven构建后端项目为war包的过程，让我深入理解了Java企业级应用的构建流程，包括依赖管理、生命周期管理、插件配置等重要概念。同时，使用NPM构建前端项目让我掌握了现代前端工程化的核心思想，从包管理到构建优化，每个环节都体现了现代开发的高效性和规范性。

部署阶段是理论转化为实践的关键步骤。将前端项目部署到Nginx，后端项目部署到Tomcat，并通过Nginx反向代理实现前后端通信，这个过程让我深刻理解了前后端分离架构的核心价值。Nginx作为反向代理服务器，不仅处理静态资源访问，还实现了负载均衡、请求转发等高级功能，体现了现代Web架构的灵活性和可扩展性。

配置服务器日志文件目录的工作让我认识到日志管理在运维工作中的重要地位。通过合理的日志配置，我学会了如何有效收集、存储和分析系统运行数据。日志不仅是问题排查的重要工具，更是系统性能优化和预警监控的数据基础，这为后续的系统维护提供了有力支撑。

将整个部署过程归纳到自动化脚本中是本次实验的重要收获。通过编写Shell脚本，我将复杂的部署流程标准化和自动化，不仅大幅提高了部署效率，还有效降低了人为操作错误的风险。在脚本编写过程中，我特别注重错误处理机制、状态检查逻辑和回滚功能的实现，这些细节体现了生产环境部署的严谨性和可靠性要求。

编写计划任务对日志文件进行定期备份和分析，让我深入理解了运维自动化的核心理念。通过crontab设置定时任务，实现日志的自动备份、清理和初步分析，不仅保证了系统的持续稳定运行，还为系统监控和性能优化提供了宝贵的数据支持。

通过本次实验，我深刻体会到现代Web应用部署的系统性和复杂性。前后端分离架构不仅提高了开发效率和代码可维护性，更重要的是为系统的横向扩展和技术栈选择提供了更大的灵活性。在实验过程中，我逐渐认识到系统性思维在技术工作中的重要性，每个组件的配置都需要考虑与其他组件的协调配合，这要求我们具备全局视野和整体把控能力。

此外，我也深刻认识到文档规范和自动化脚本在项目管理中的价值。详细的配置文档能够帮助团队成员快速理解系统架构和部署流程，而自动化脚本则能够确保部署过程的一致性和可重复性。在编写脚本和文档时，我始终坚持代码可读性和注释完整性的原则，这不仅便于自己的后续维护，更重要的是为团队协作和知识传承奠定了基础。

虽然本次实验采用的是相对传统的部署方式，但其中蕴含的核心理念和实践经验对于理解现代云原生、容器化等前沿技术仍然具有重要的指导意义。随着技术的不断发展，我期待能够将本次实验中积累的基础知识与Docker、Kubernetes、CI/CD等现代技术相结合，不断提升自己的技术水平和实践能力。

总的来说，本次实验让我对企业级Web应用的部署和运维有了全面而深入的认识，从技术栈选择到架构设计，从环境配置到自动化部署，每个环节都让我收获颇丰。这些宝贵的知识和实践经验不仅为我未来的学习和工作奠定了坚实基础，更重要的是培养了我解决复杂技术问题的系统性思维和实践能力。我相信这次实验的收获将在我的职业发展道路上发挥重要作用。