### **Multi-Branch Project**

#### **Project Overview**

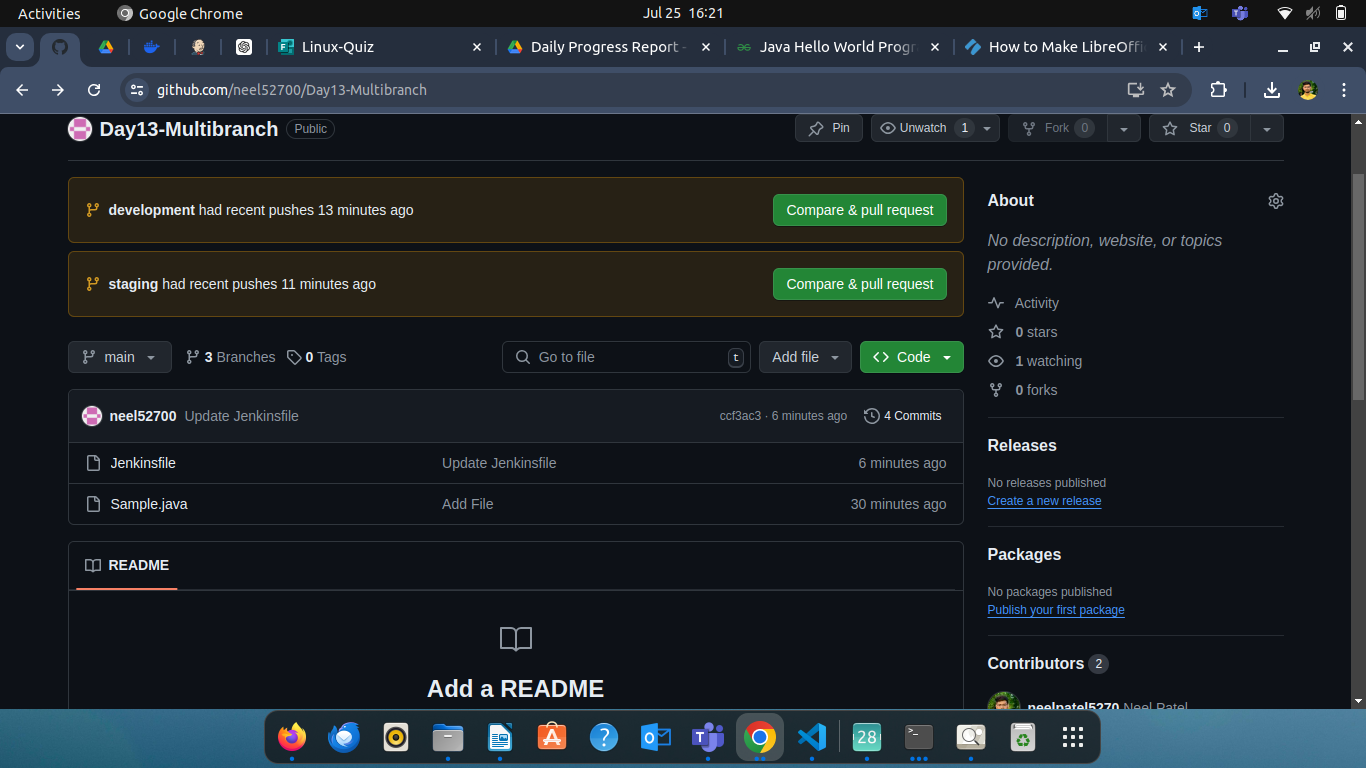
* Create a simple Java Maven project.
* Version-control the project using Git with multiple branches.
* Set up Jenkins multi-branch pipeline for automated build and deployment.
* Utilize Jenkins environment variables in the Jenkinsfile.

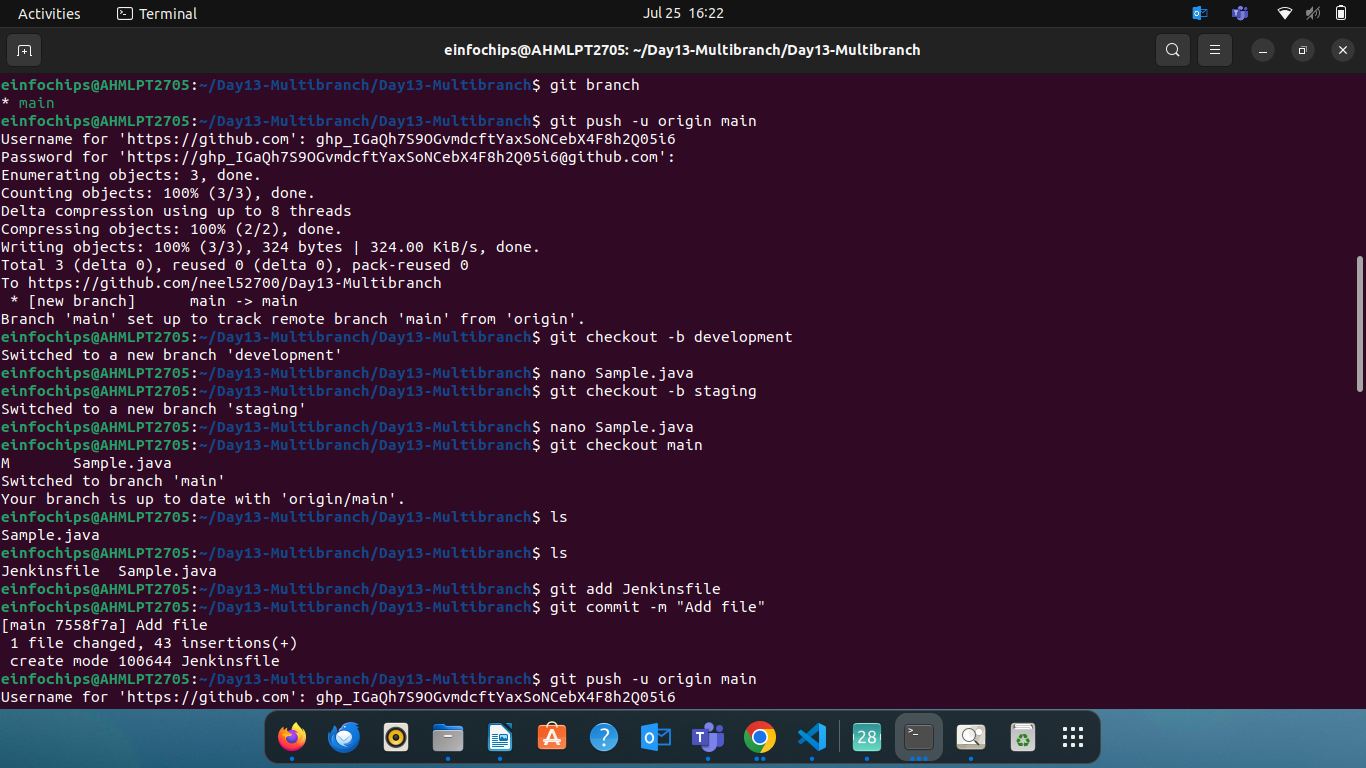
#### **Project Objectives**

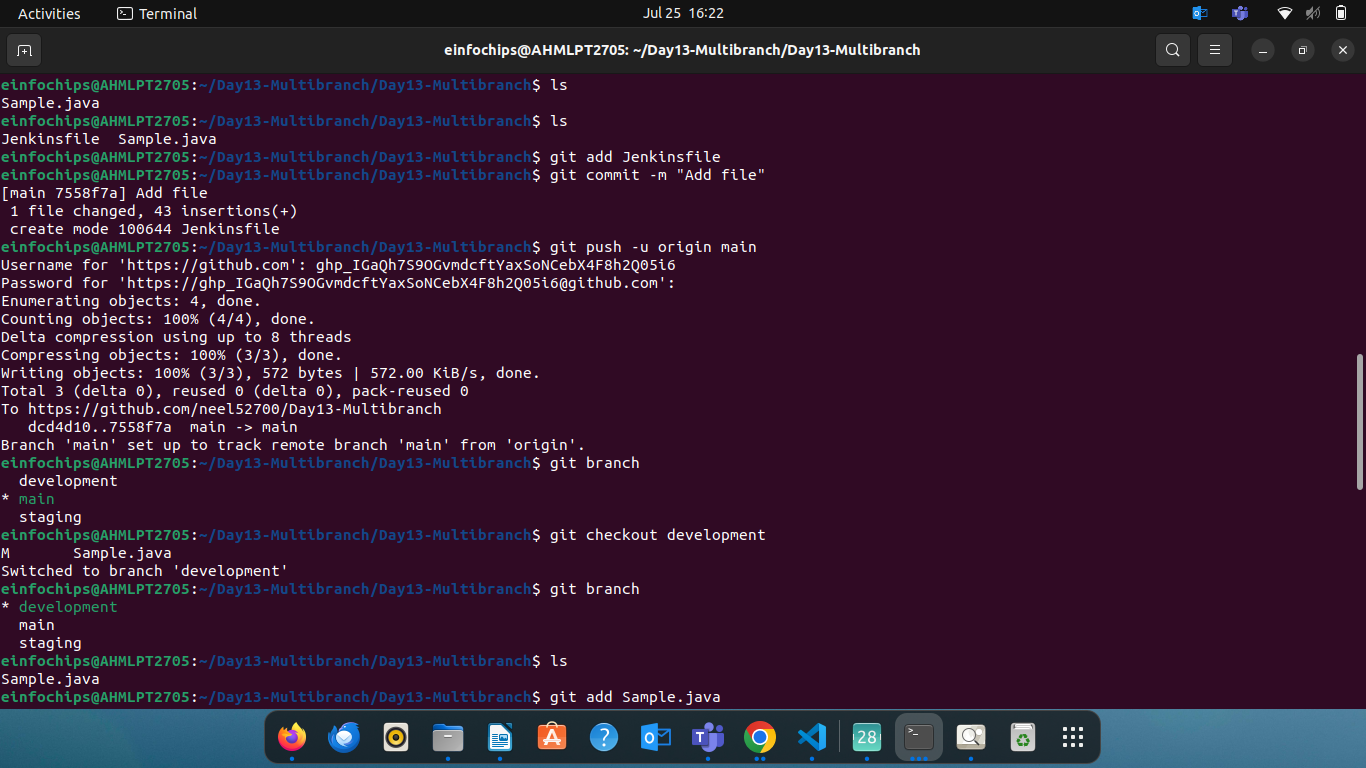
* Version-control using Git.
* Jenkins multi-branch pipeline setup.
* Environment variable management using Jenkinsfile.

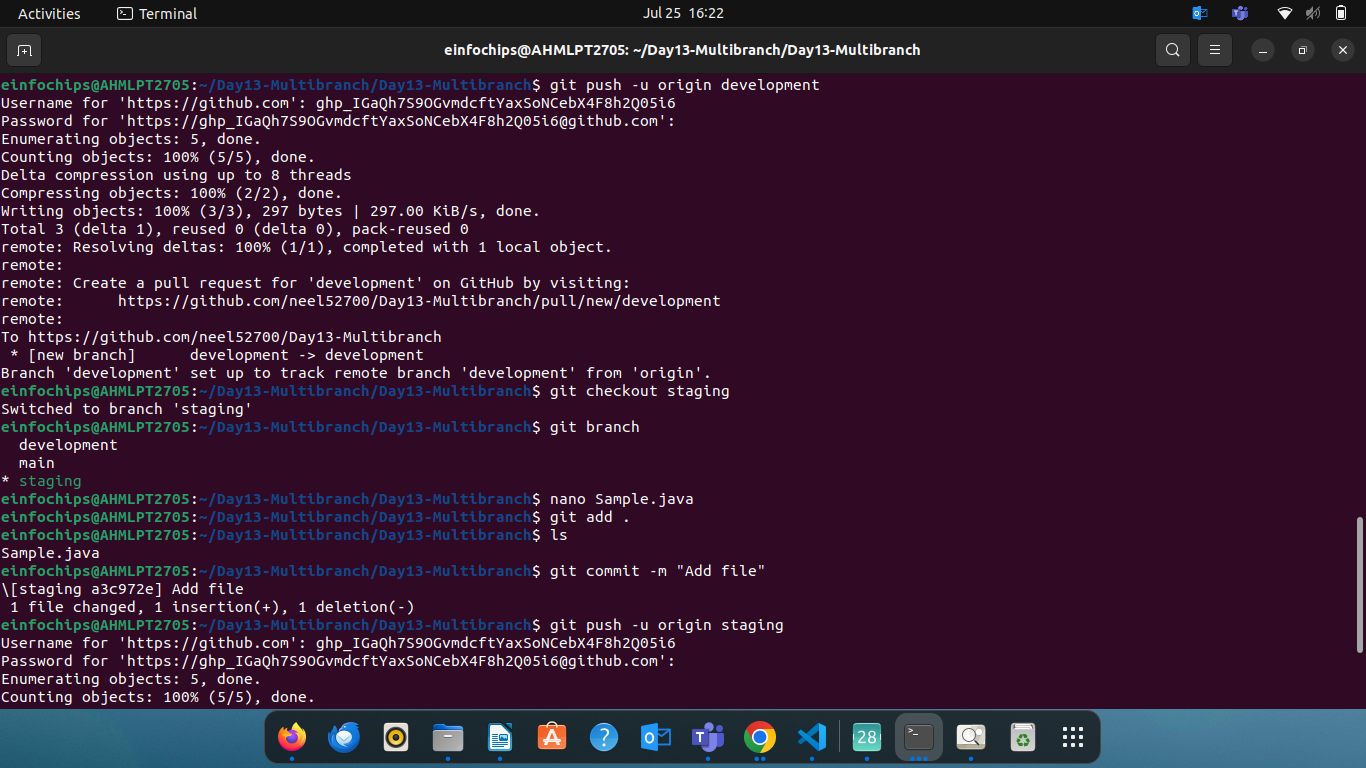
#### **Project Deliverables**

1. **Git Repository:**
   * Local Git repository initialized.
   * Branches: development, staging, and production.
   * Repository pushed to remote Git server (e.g., GitHub, GitLab, Bitbucket).

**Step 1**



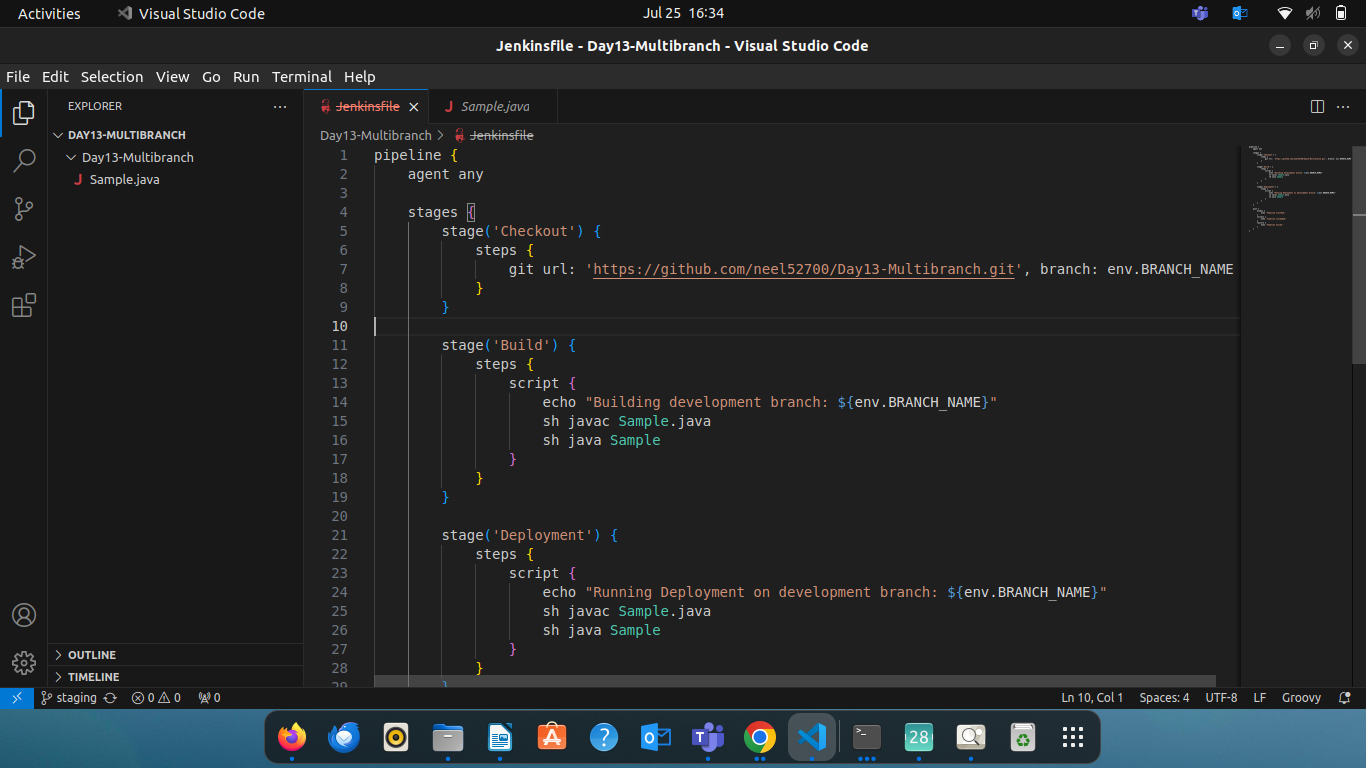


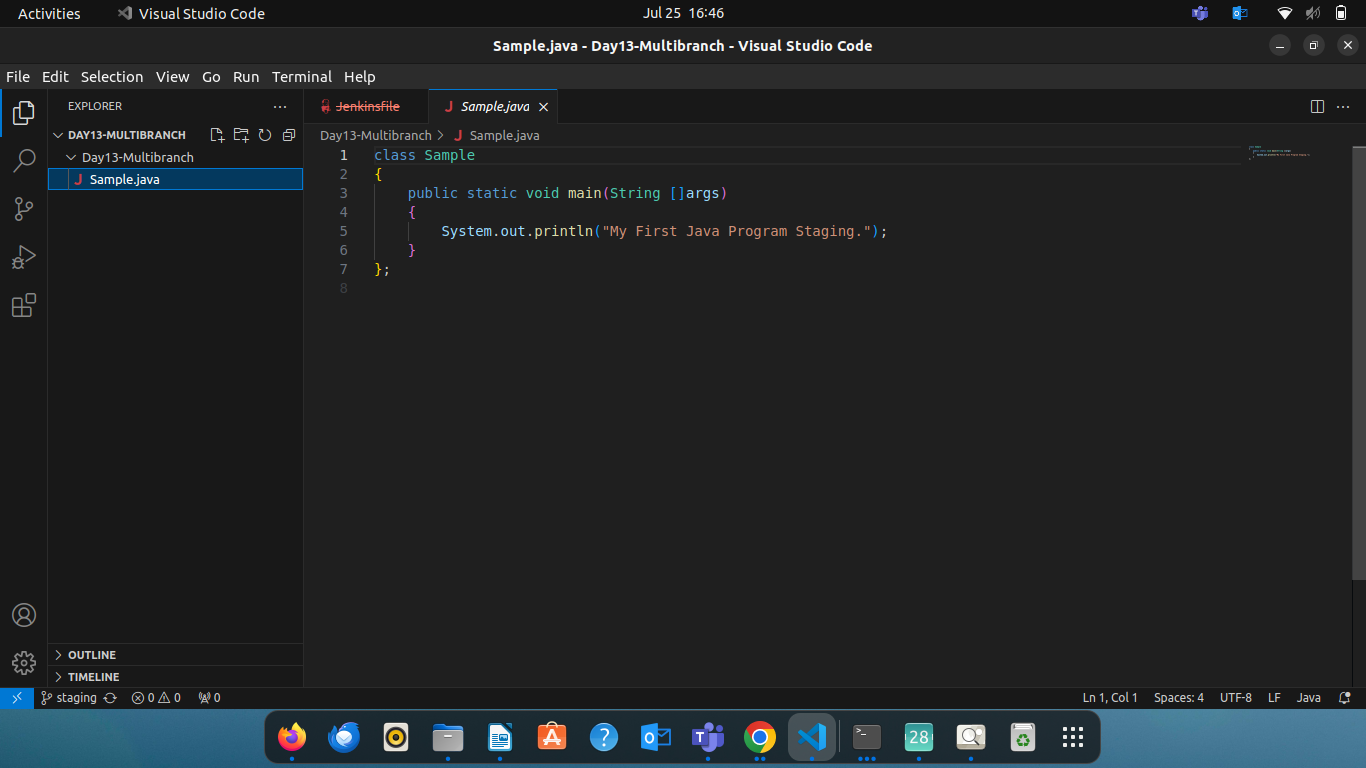


1. Initialize Local Git Repository
2. Create Branches
3. Create a Simple Java Project on main branch
4. Push to Remote Git Server

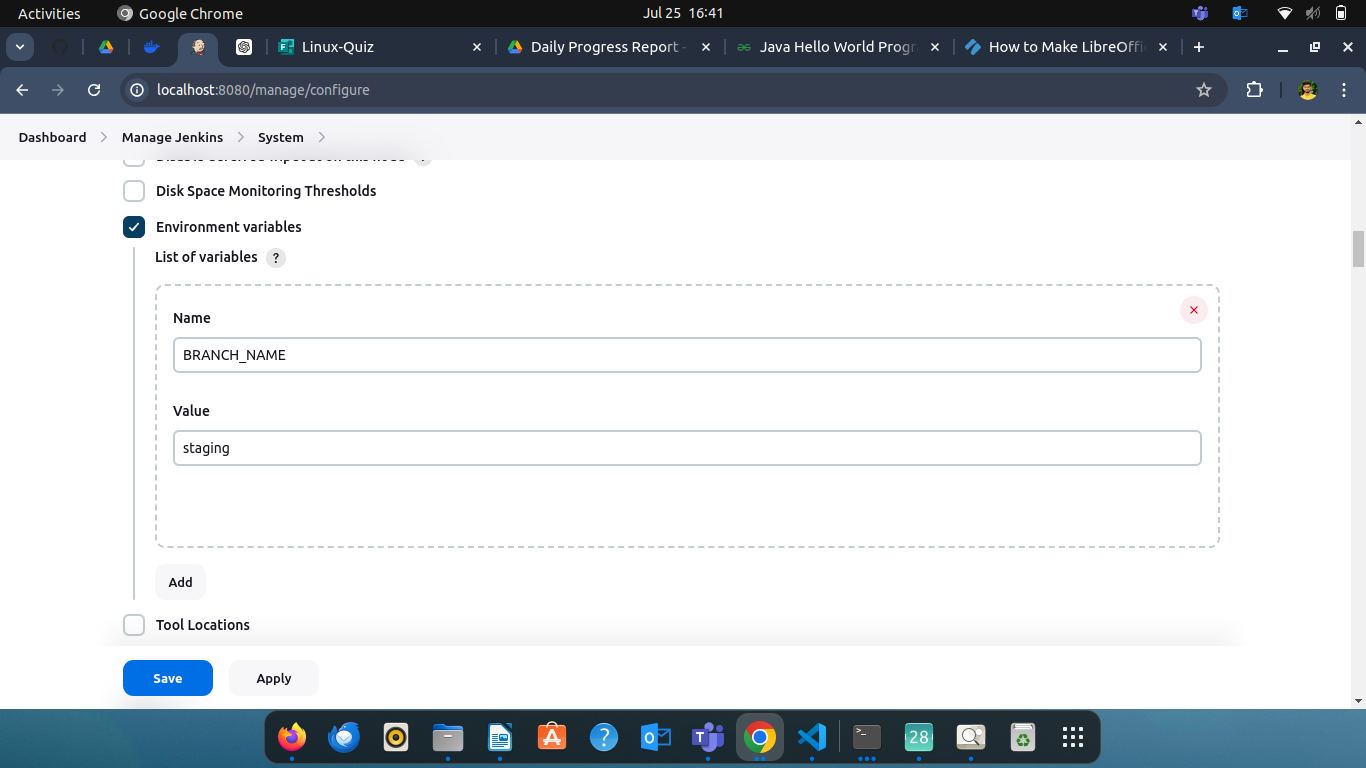
* Add the remote repository
* Push branches to the remote repository

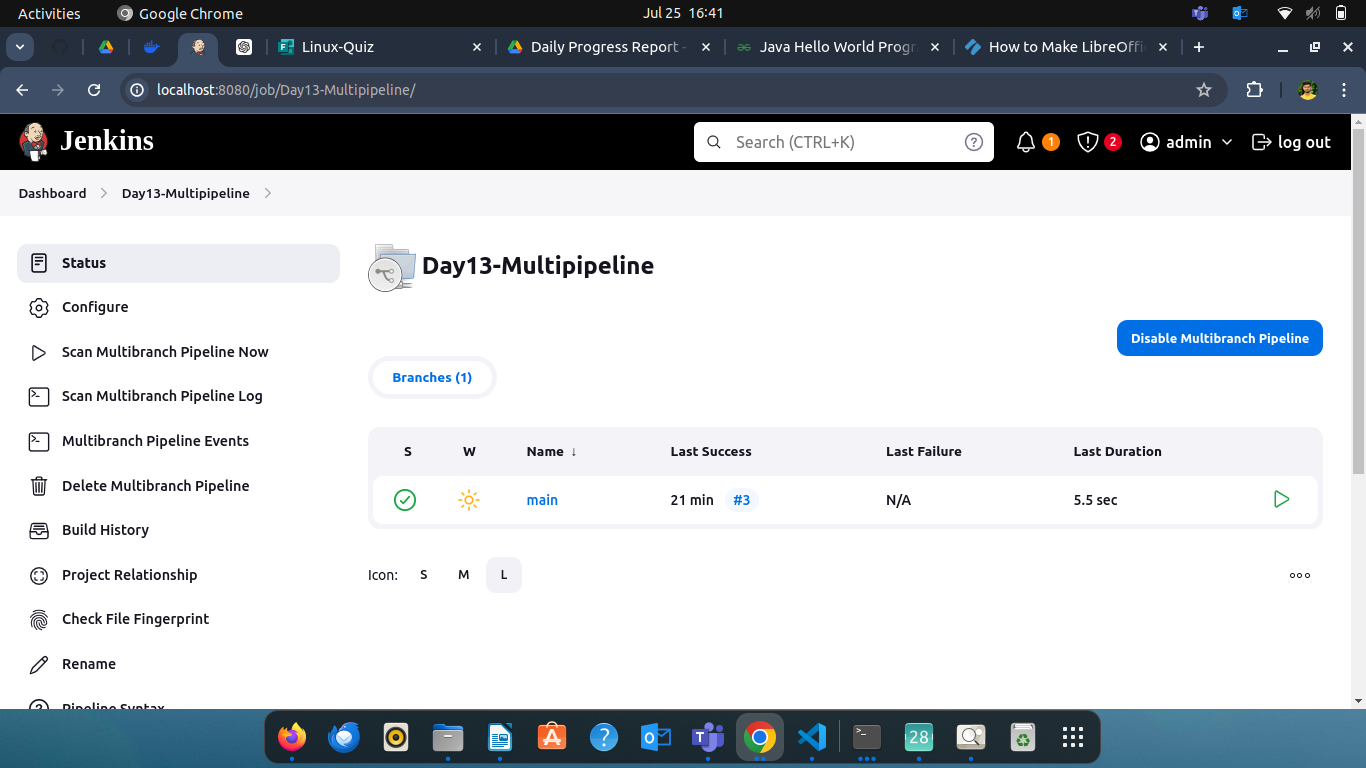
1. Create a jenkins file on main branch
2. **Maven Project:**
   * Simple Java Maven project created (HelloWorld application).
   * pom.xml with dependencies and build configurations.

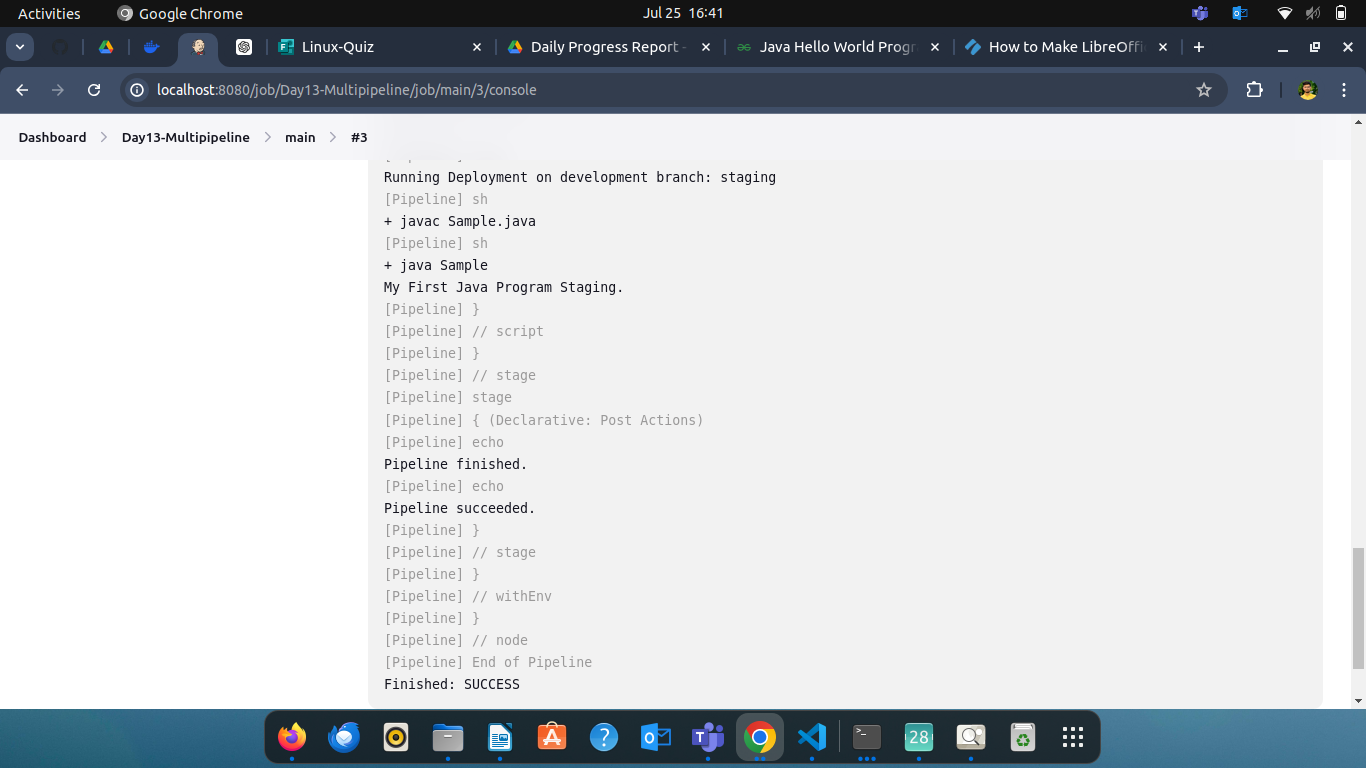




1. **Jenkins Setup:**
   * Multi-branch pipeline job configured in Jenkins.
   * Jenkinsfile defining build and deployment steps.
   * Environment variables managed using Jenkins environment variable settings.





****

* Trigger the Pipeline Manually:
* Go to Jenkins dashboard.
* Select the pipeline job.
* Click on "Build Now".
* Monitor the Build Process:
  + Open the build details in Jenkins.
  + Check the console output to monitor the progress and status of each stage.