CI/CD Mini Project

PART A – Jenkins Freestyle Job

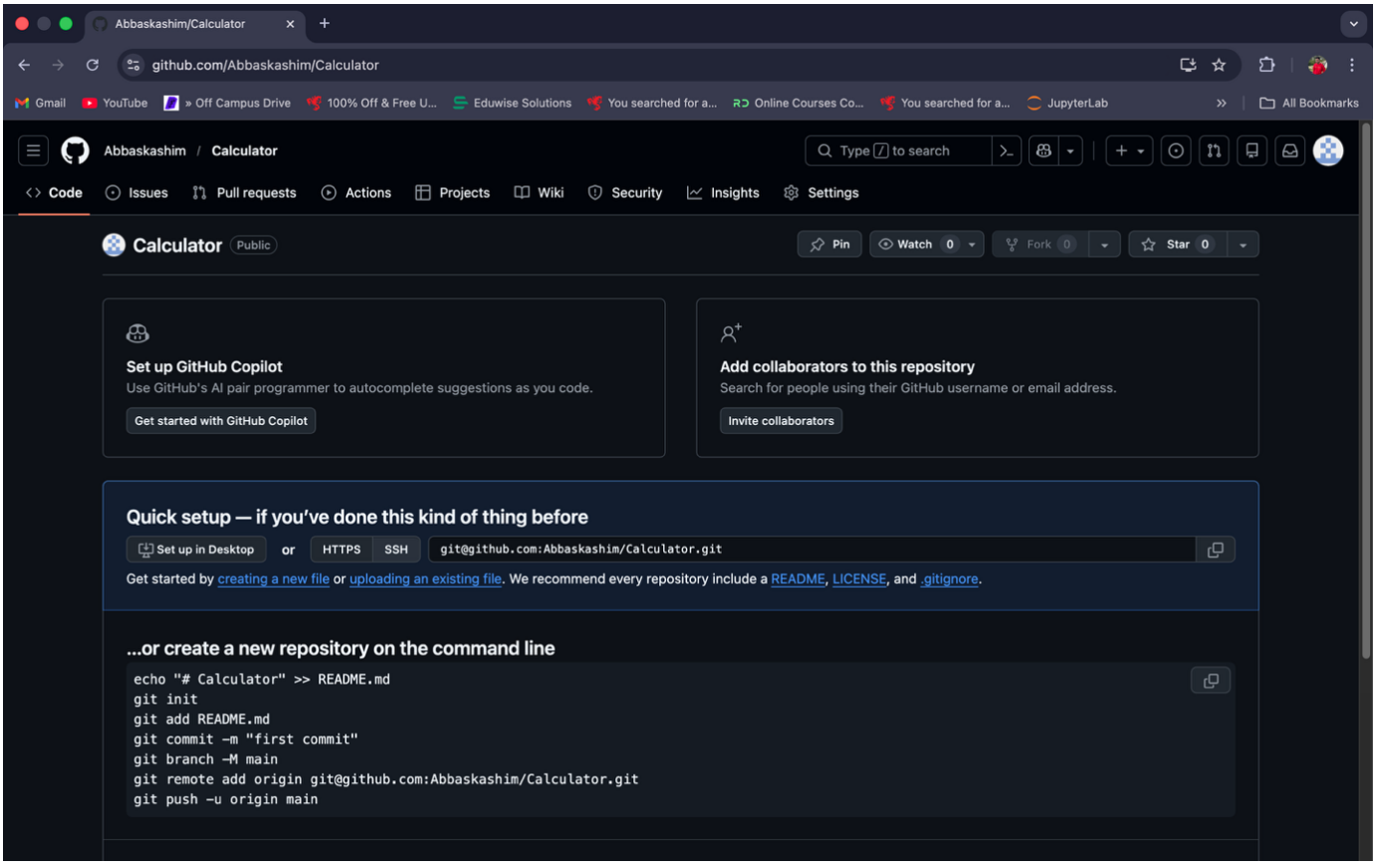
Requirements:

✓ GIT REPOSITORIES : https://github.com/Abbaskashim/Calculator.git

---

STEP A1: CREATE GITHUB REPOSITORY

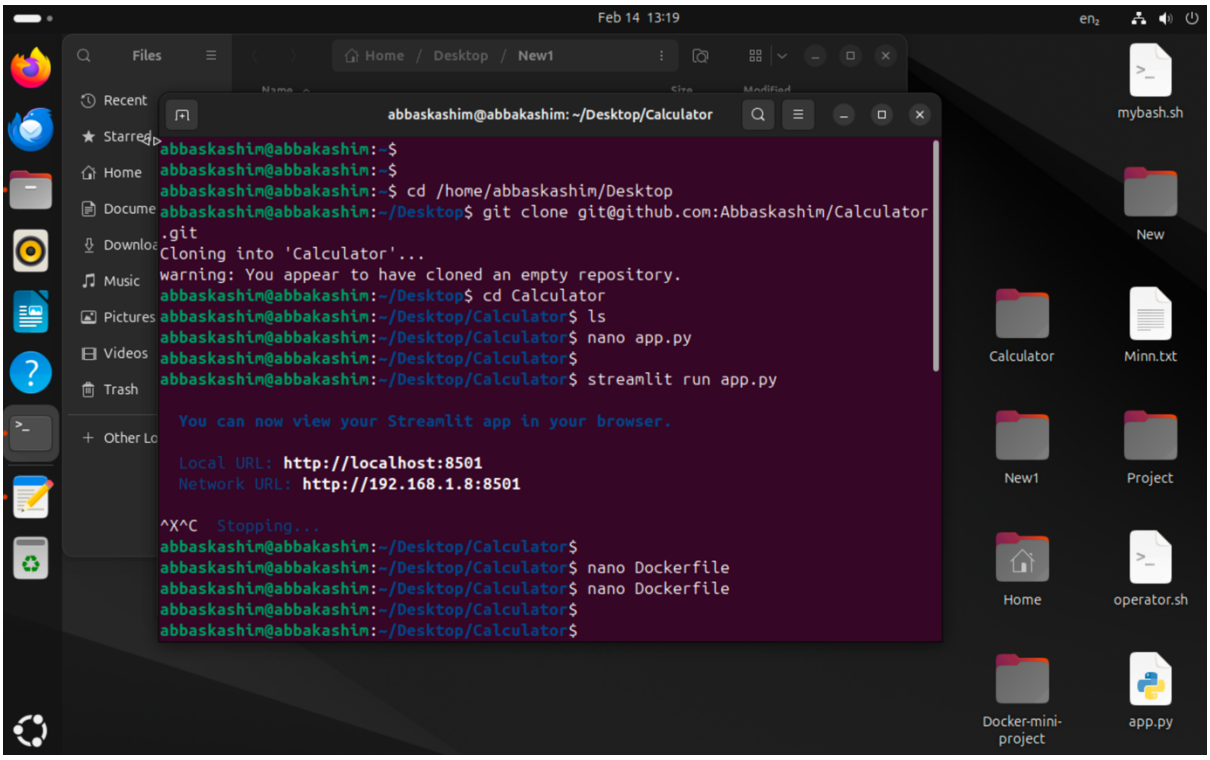
1. Open **github.com**
2. Click **New Repository**
3. Repository name: `Calculator`
4. Select **Public**
5. Click **Create Repository**

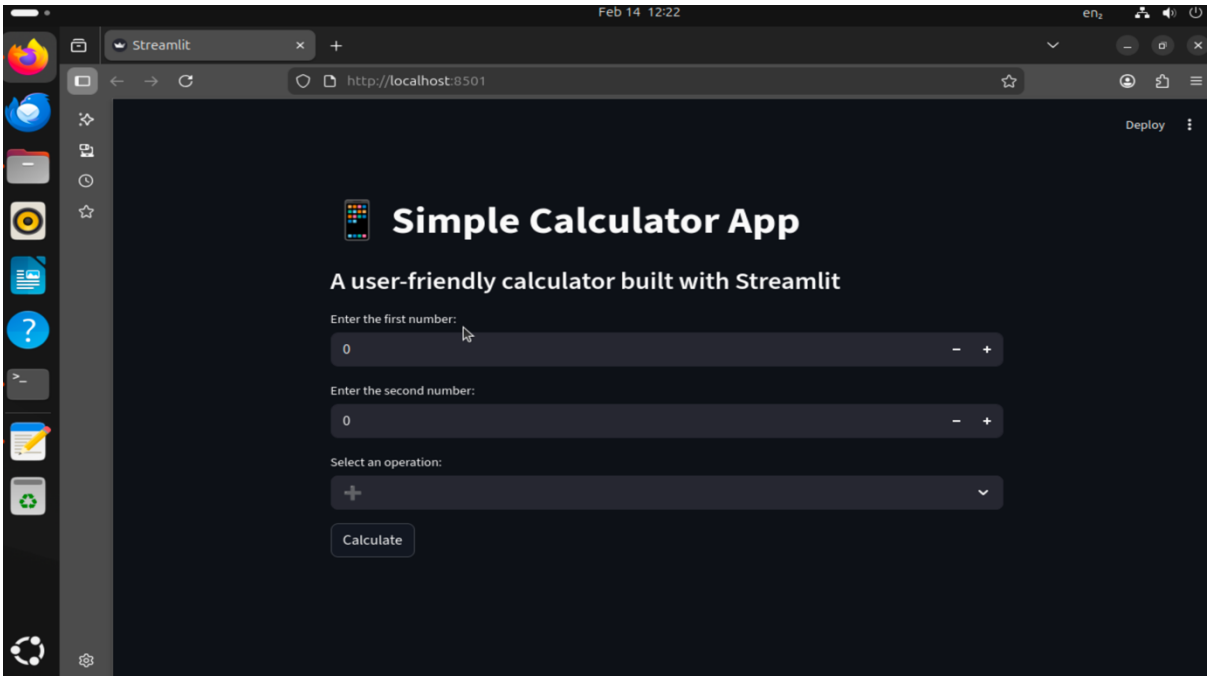


STEP A2: CREATE PROJECT FILES (LOCAL SYSTEM)

Open Terminal : `Clone the Repository`
Create app.py & Dockerfile

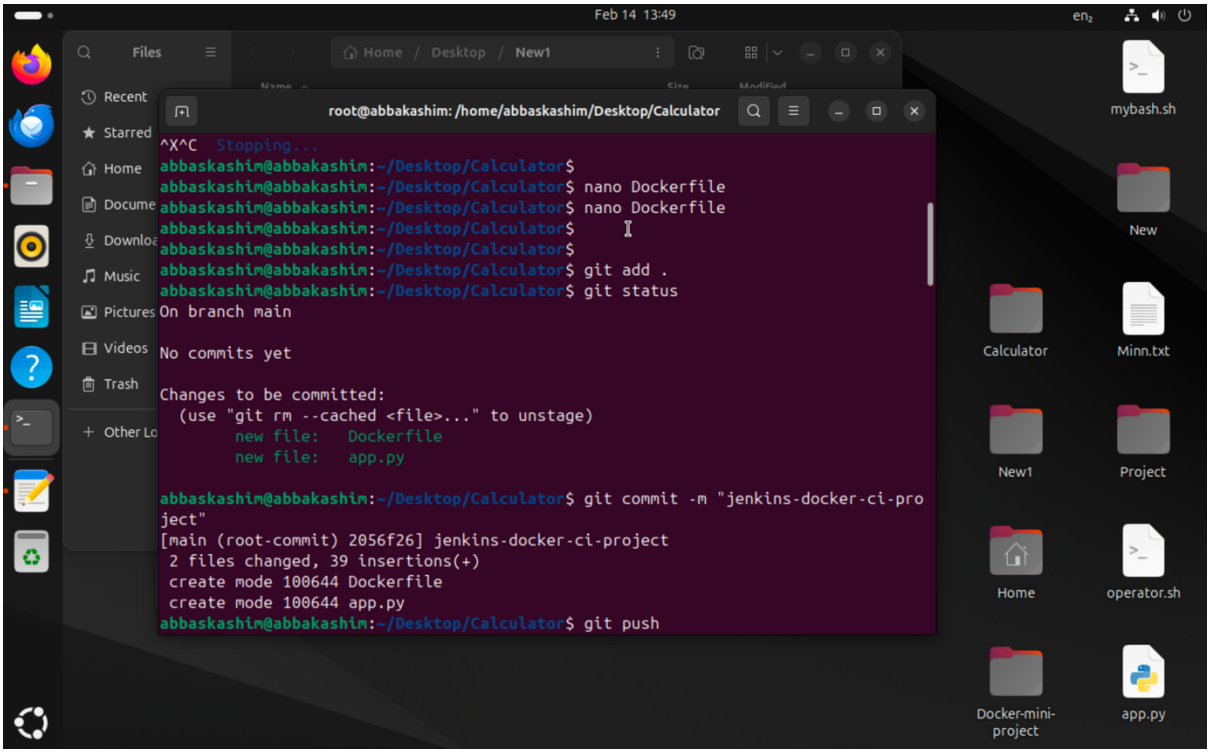
```
nano app.py
nano Dockerfile
```

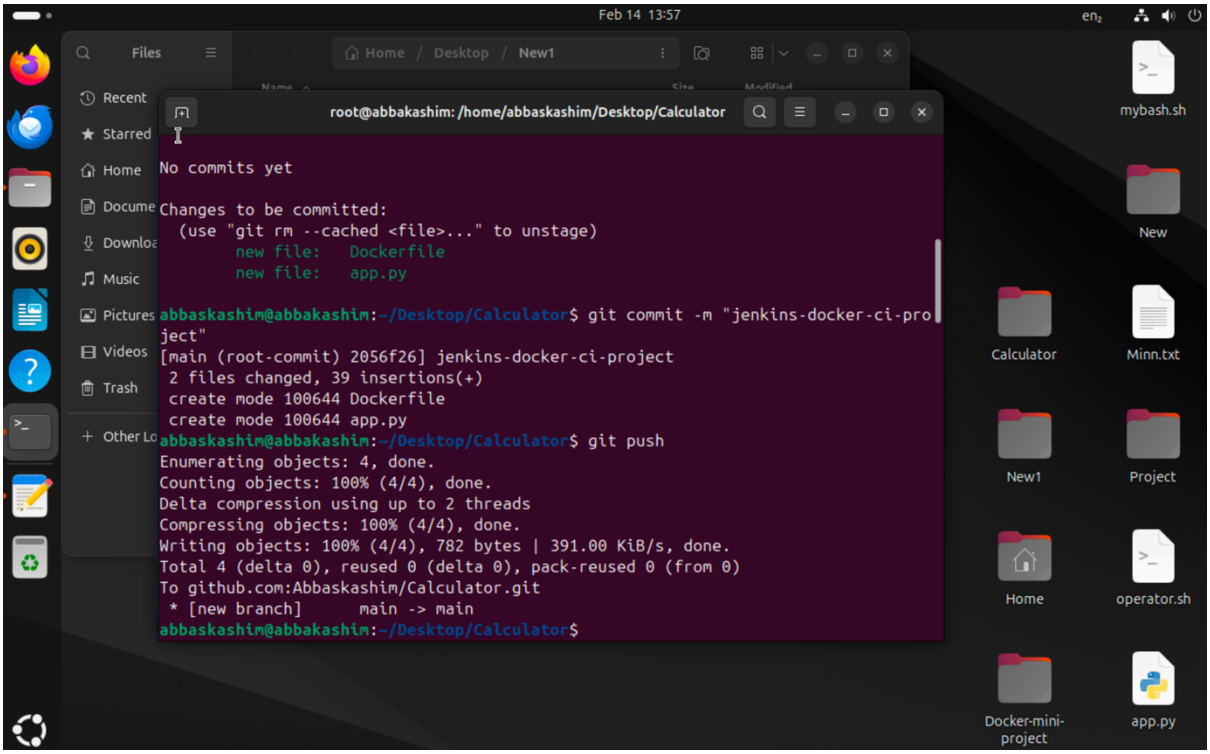




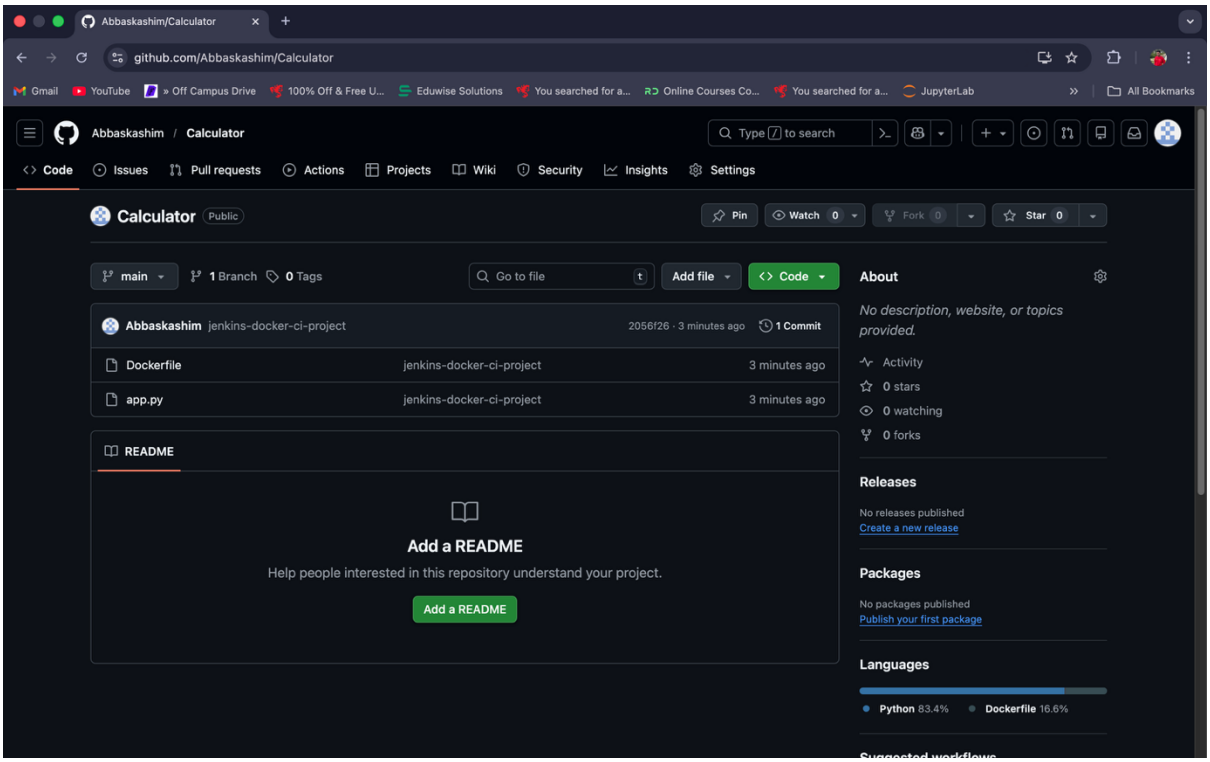
STEP A3: PUSH CODE TO GITHUB

```
git add .
git status
git commit -m "jenkins-docker-ci-project"
git remote add origin git@github.com:Abbaskashim/Calculator.git
git push
```





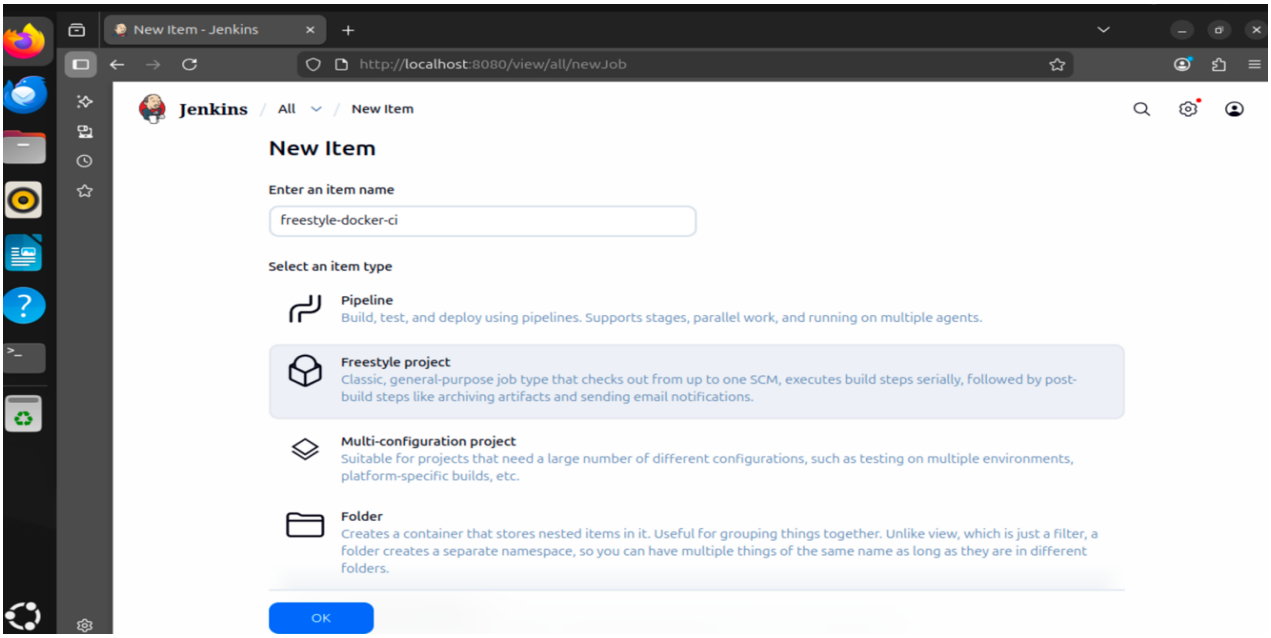
Output:



STEP A4: CREATE FREESTYLE JOB IN JENKINS

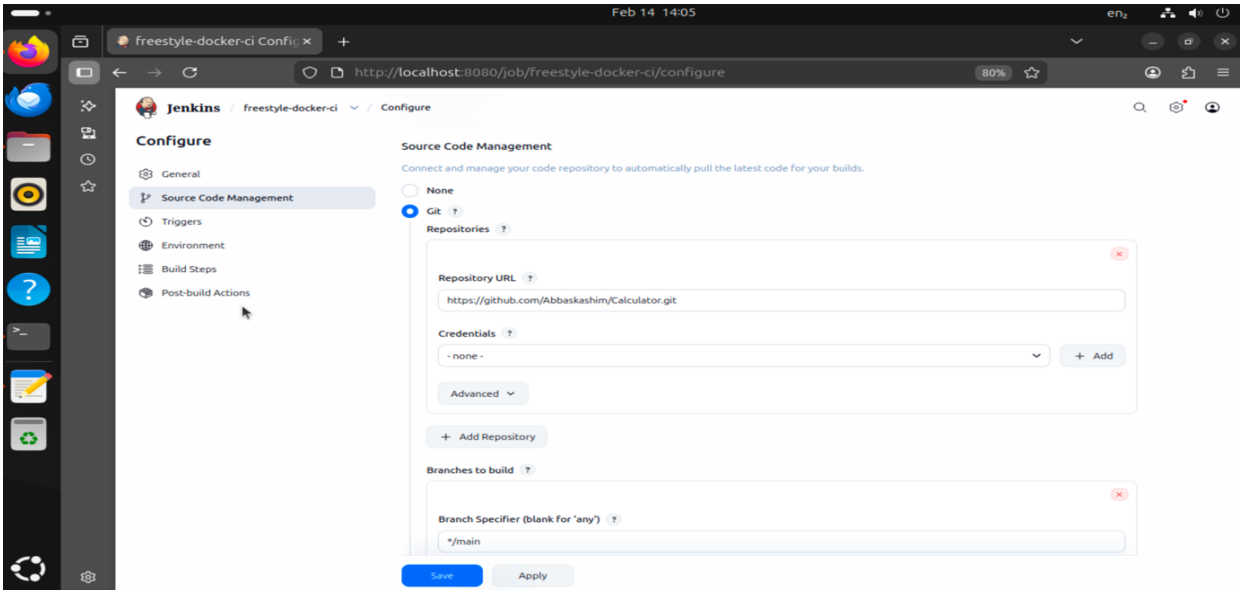
What to do

1. Open Jenkins dashboard
2. Click **New Item**
3. Enter name: `freestyle-docker-ci`
4. Select **Freestyle project**
5. Click **OK**



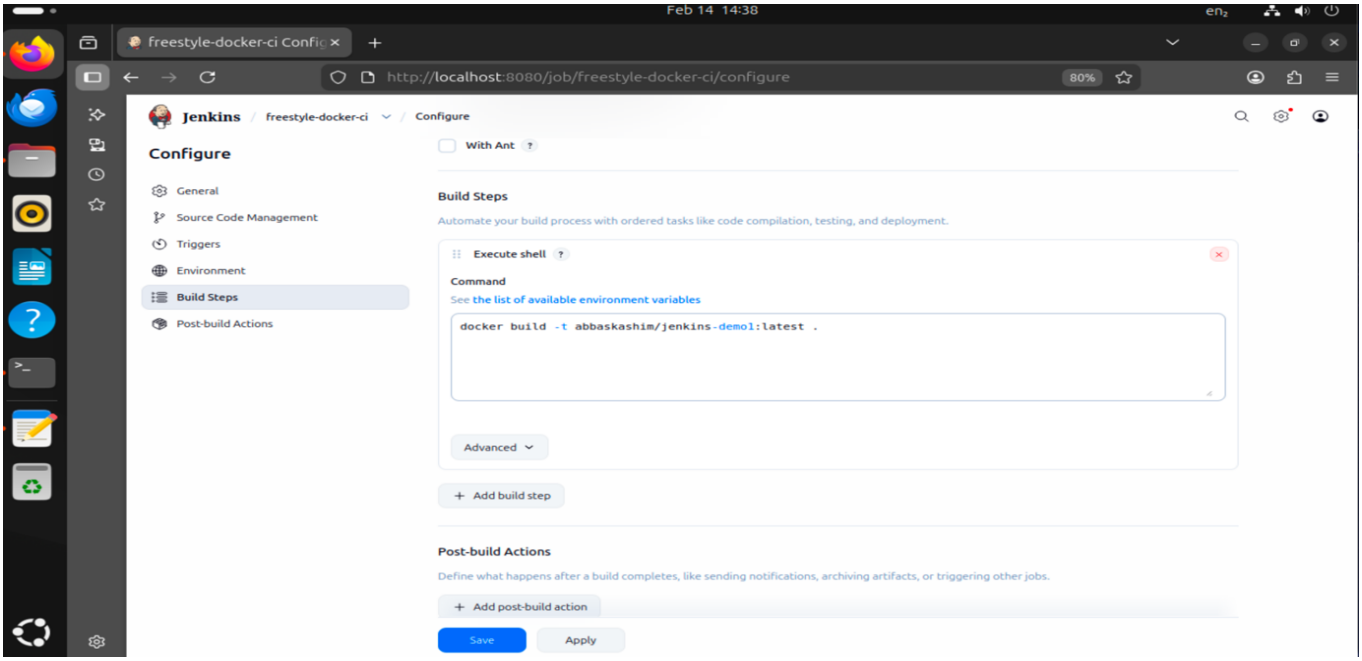
STEP A5: CONNECT JENKINS TO GITHUB

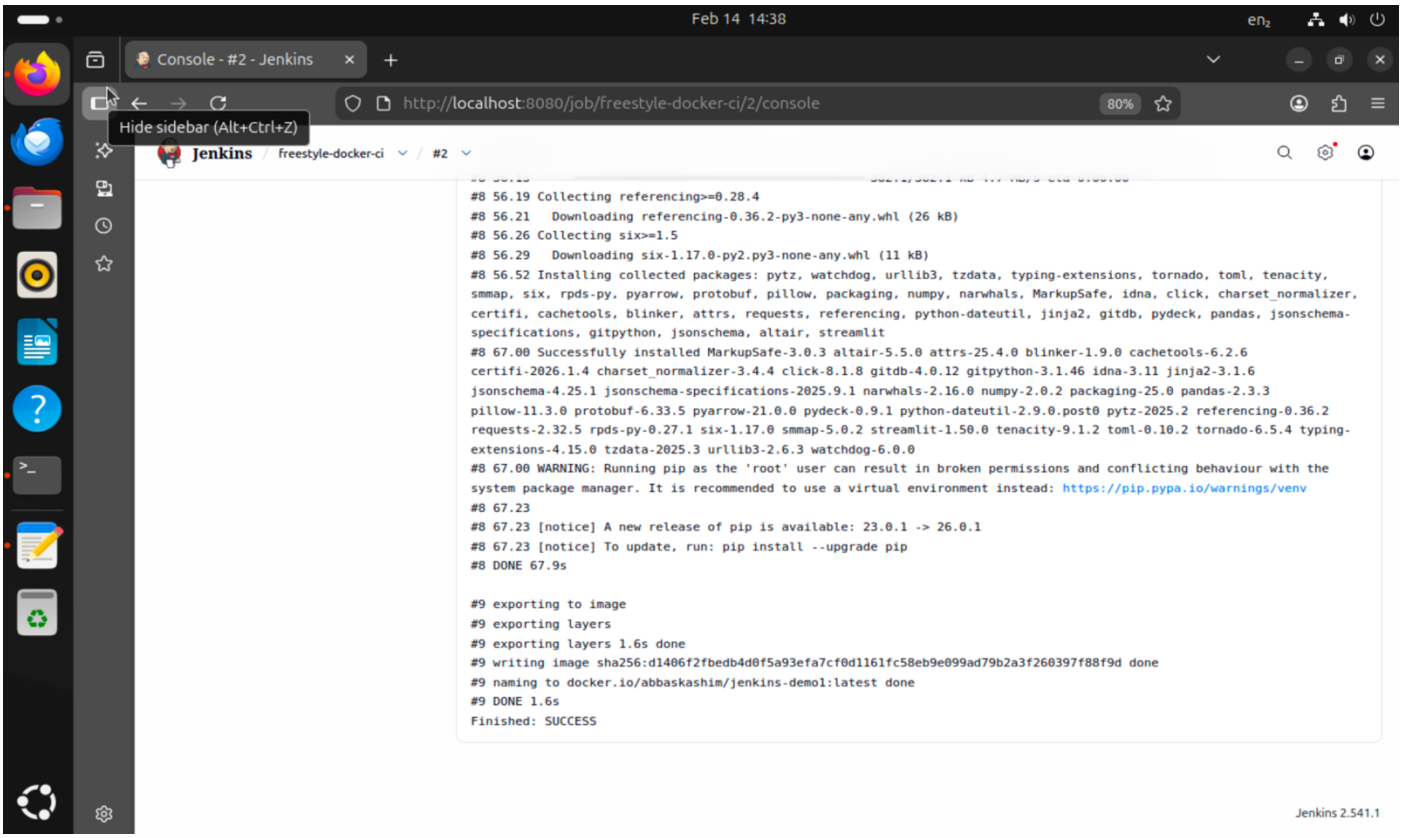
1. Click **Configure**
2. Scroll to **Source Code Management**
3. Select **Git**
4. Repository URL: https://github.com/Abbaskashim/Calculator.git
5. Branch: `main`
6. Click **Save**



STEP A6: BUILD DOCKER IMAGE

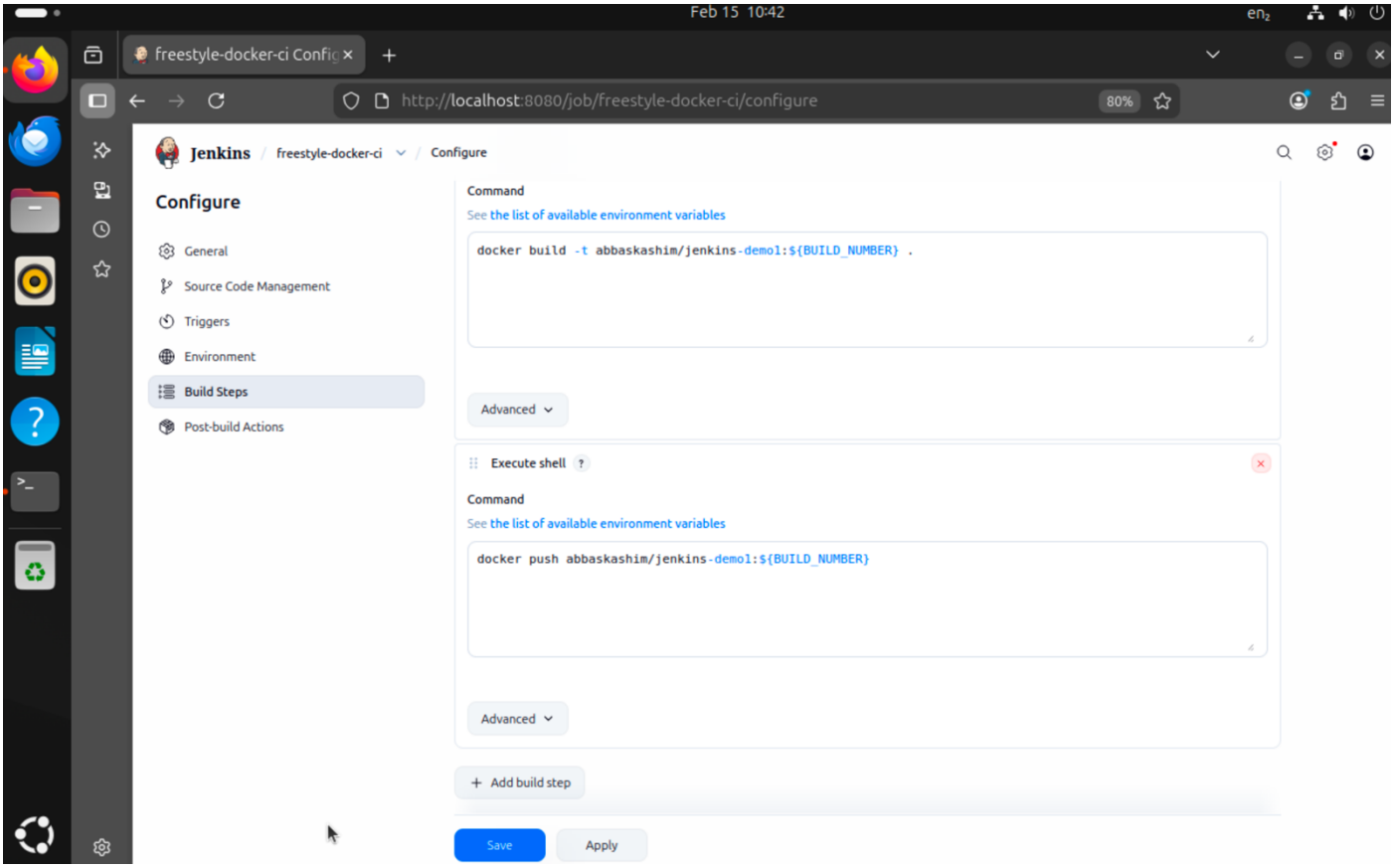
1. Click Configure
2. Scroll to Build
3. Click Add Build Step
4. Select Execute shell
5. Paste: `docker build -t abbaskashim/jenkins-demo1:latest .`



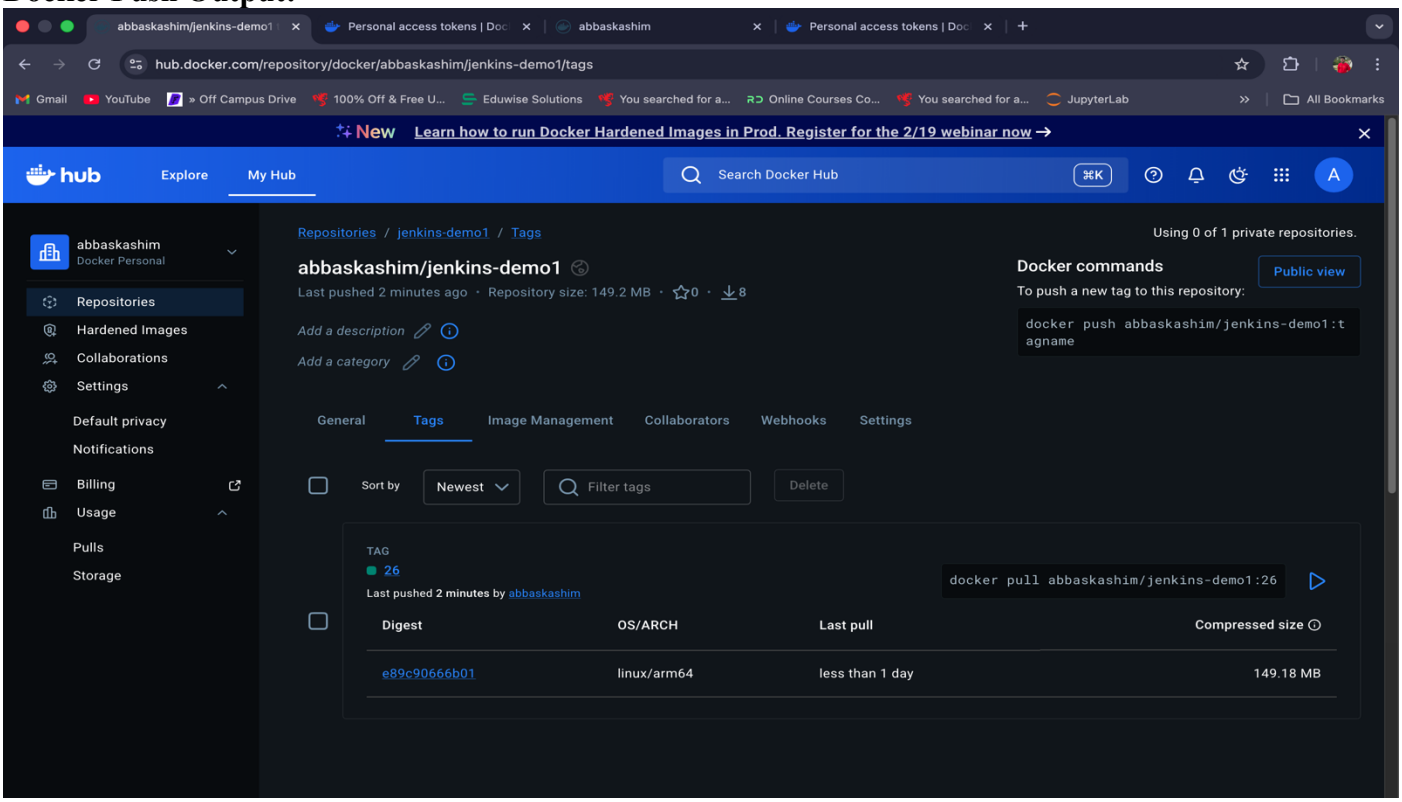


STEP A7: PUSH IMAGE TO DOCKER HUB

1. Click **Add Build Step → Execute shell**
2. Paste: `docker build -t abbaskashim/jenkins-demo1:latest .`

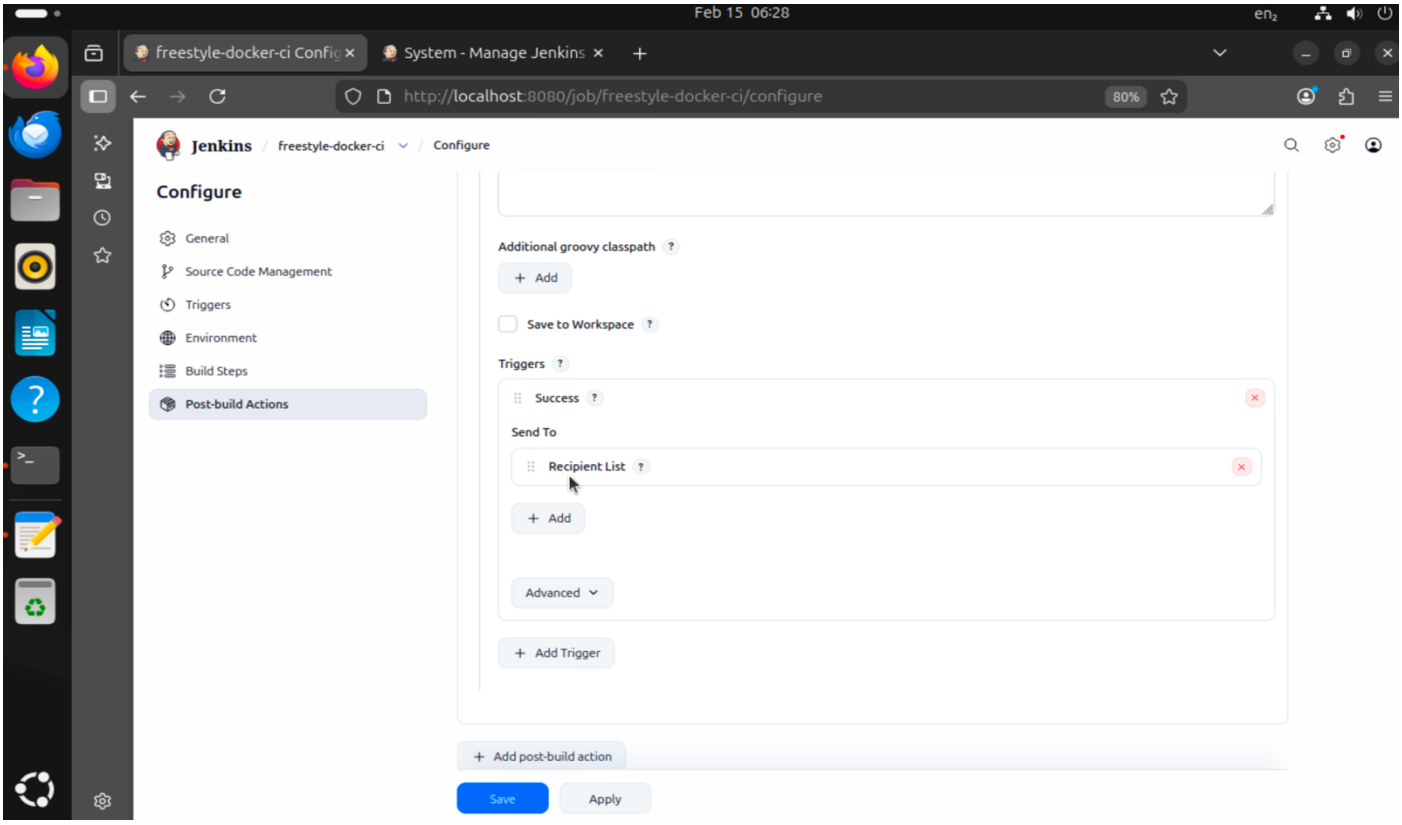


Docker Push Output:



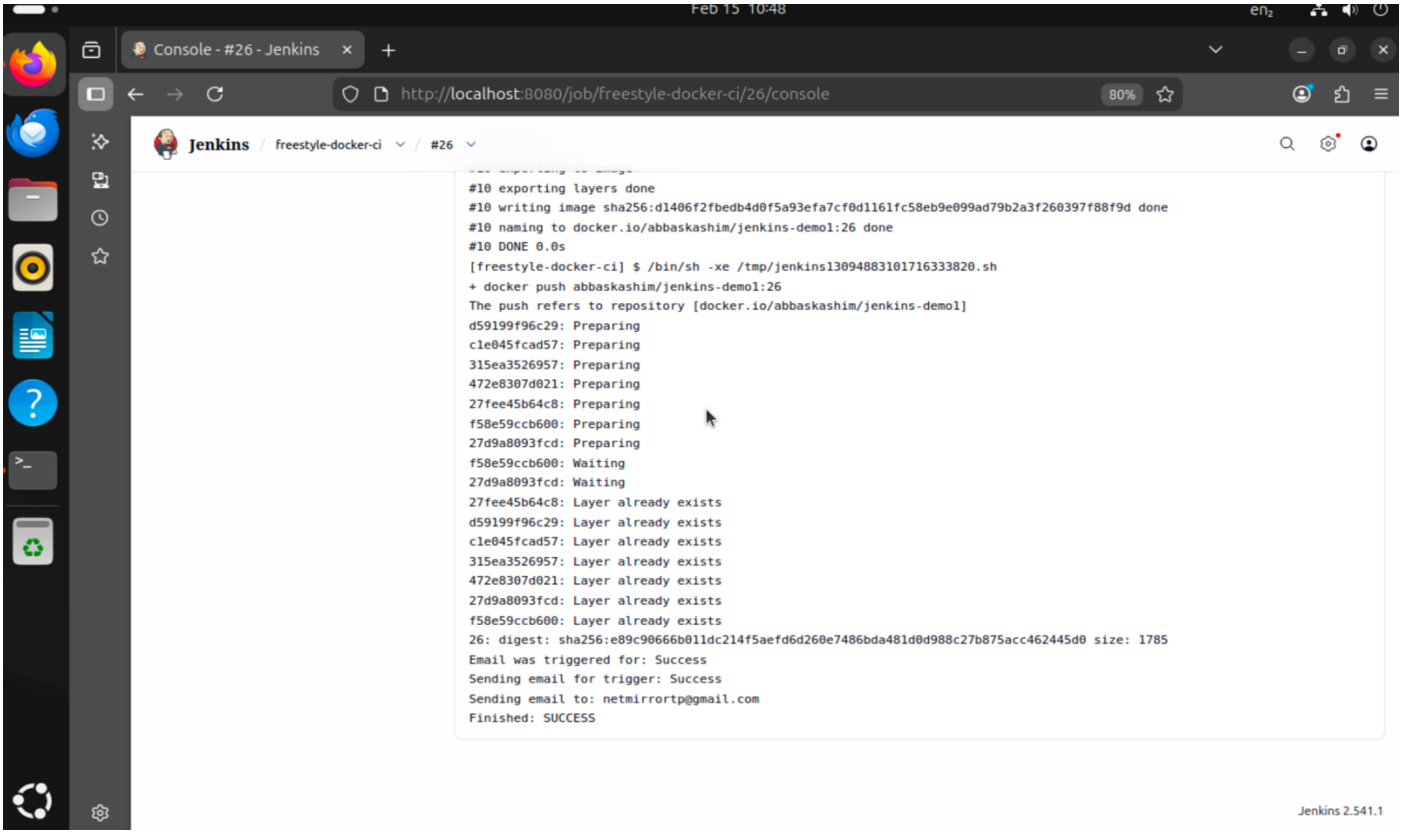
STEP A8: EMAIL NOTIFICATION

1. Scroll to **Post-build Actions**
2. Select **Email Notification**
3. Enable **Send on success**
4. Add your email

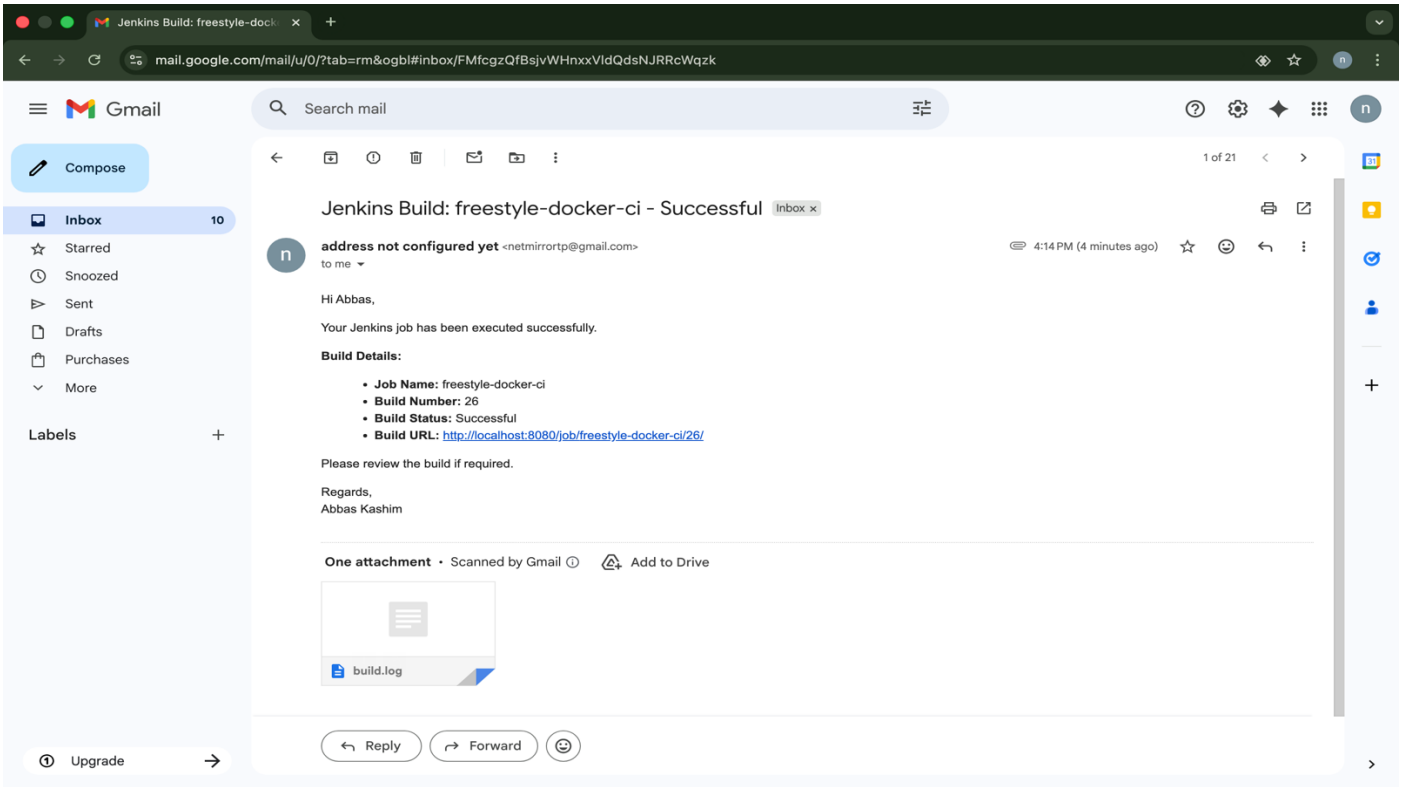


STEP A9: RUN FREESTYLE JOB

1. Click **Build Now**
2. Open **Console Output**



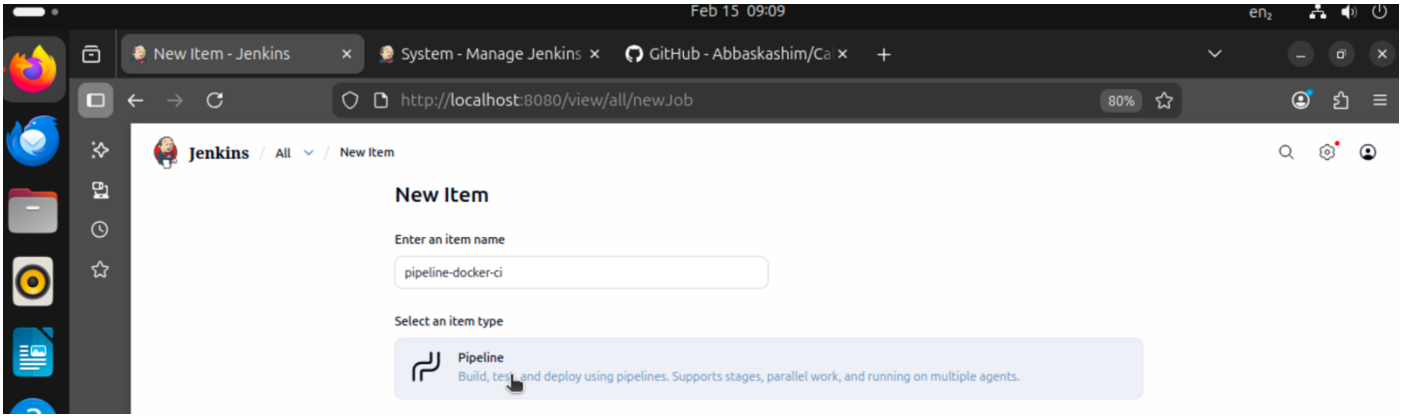
Email Notification:



PART B – Jenkins Pipeline Job

STEP B1: CREATE PIPELINE JOB

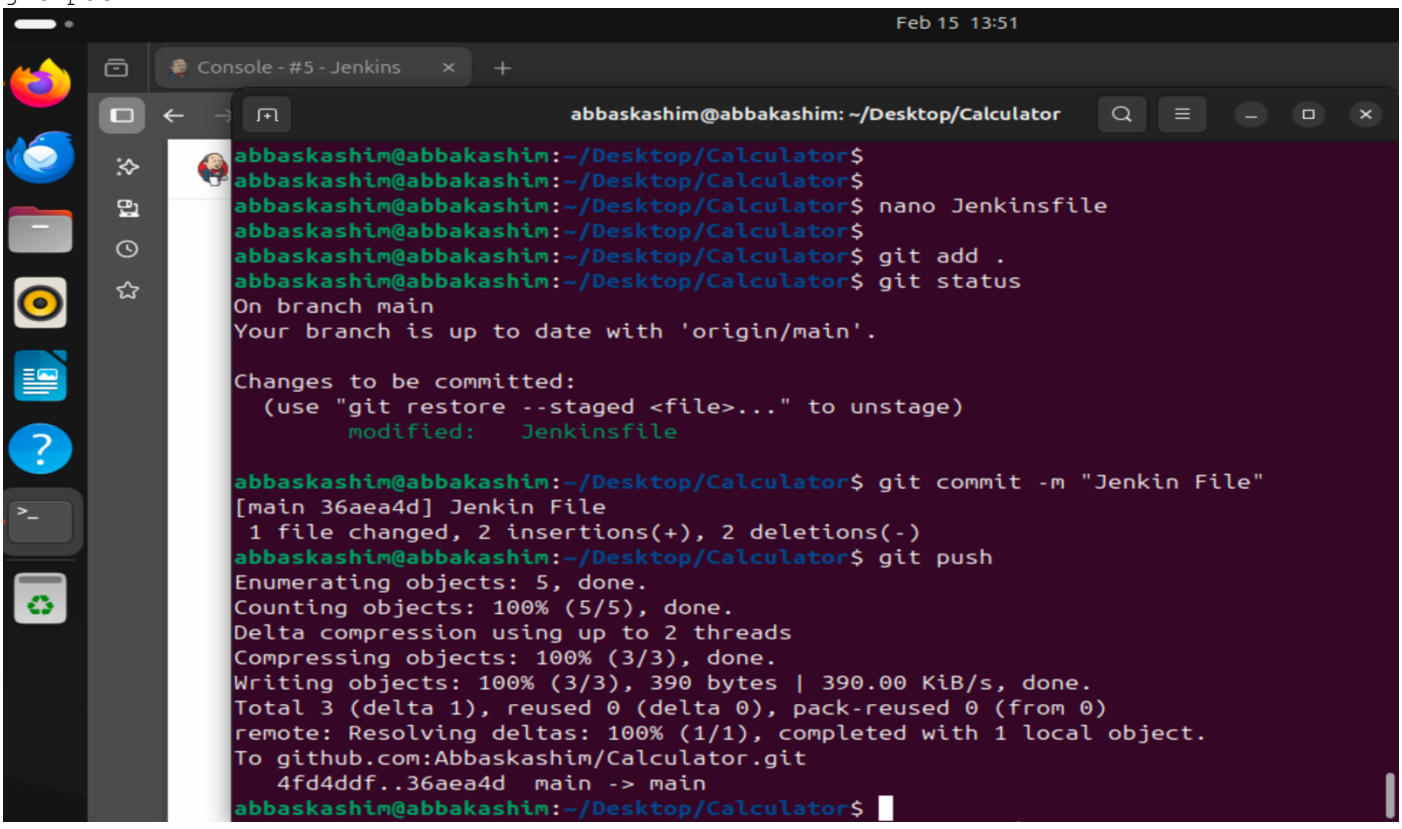
1. Jenkins → **New Item**
2. Job name: `pipeline-docker-ci`
3. Select **Pipeline**
4. Click **OK**



STEP B2: CREATE & PUSH JENKINSFILE TO GITHUB

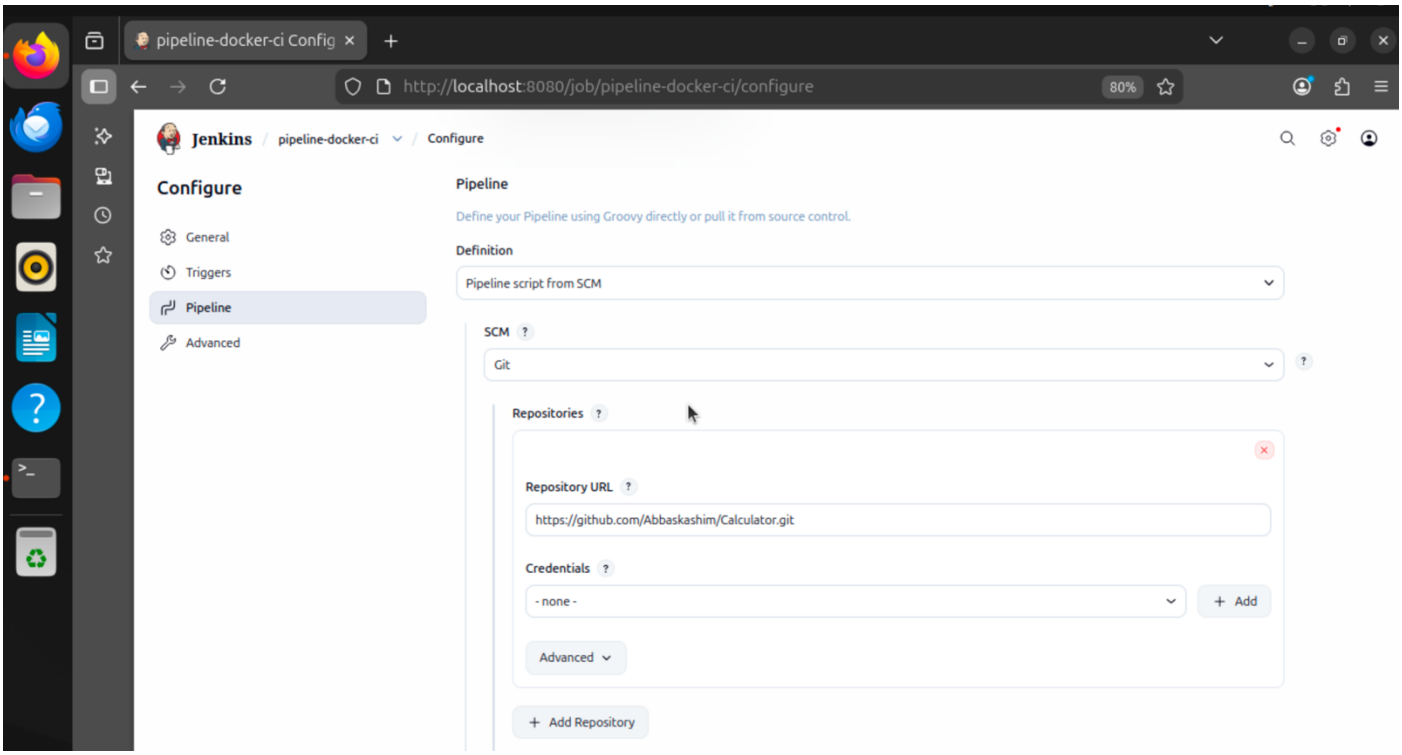
In your project folder: `nano Jenkinsfile`

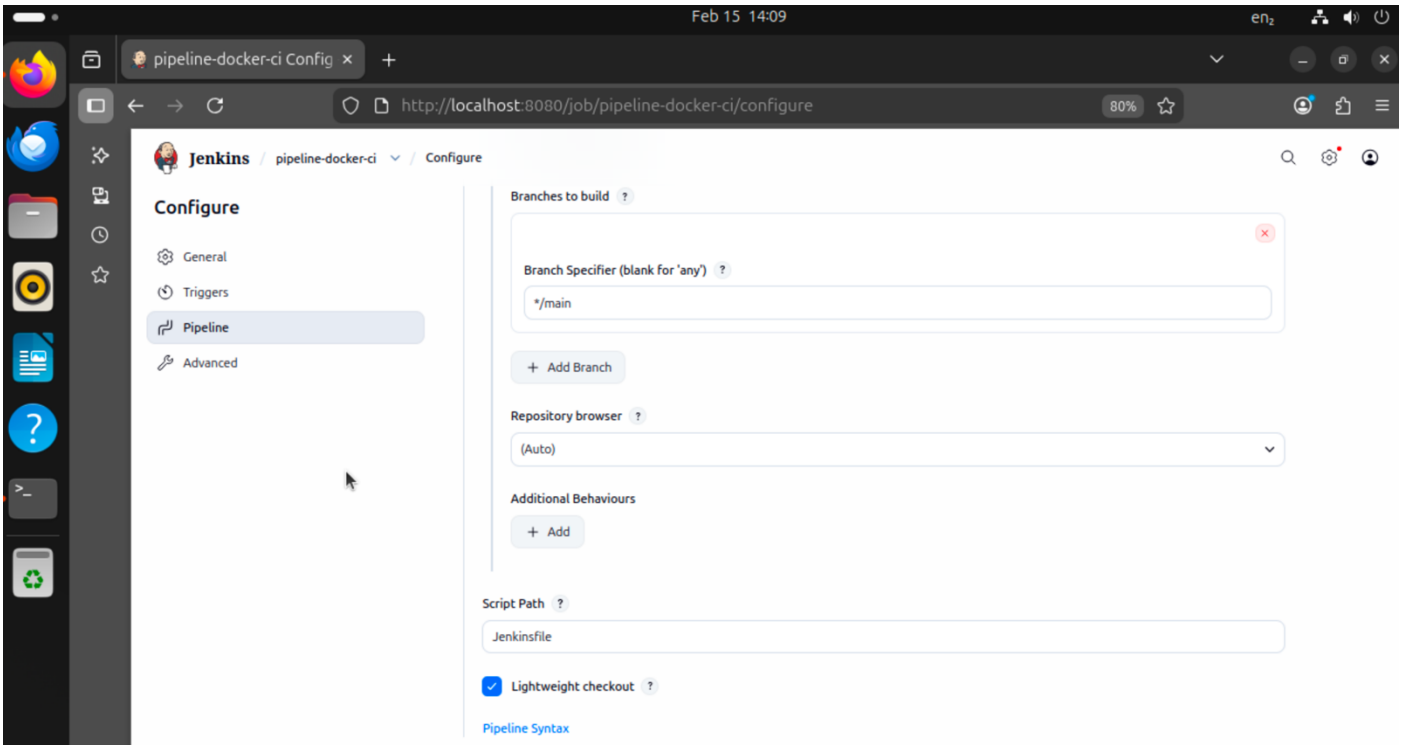
```
git add .
git commit -m "Jenkin file"
git push
```



STEP B4: CONFIGURE PIPELINE JOB

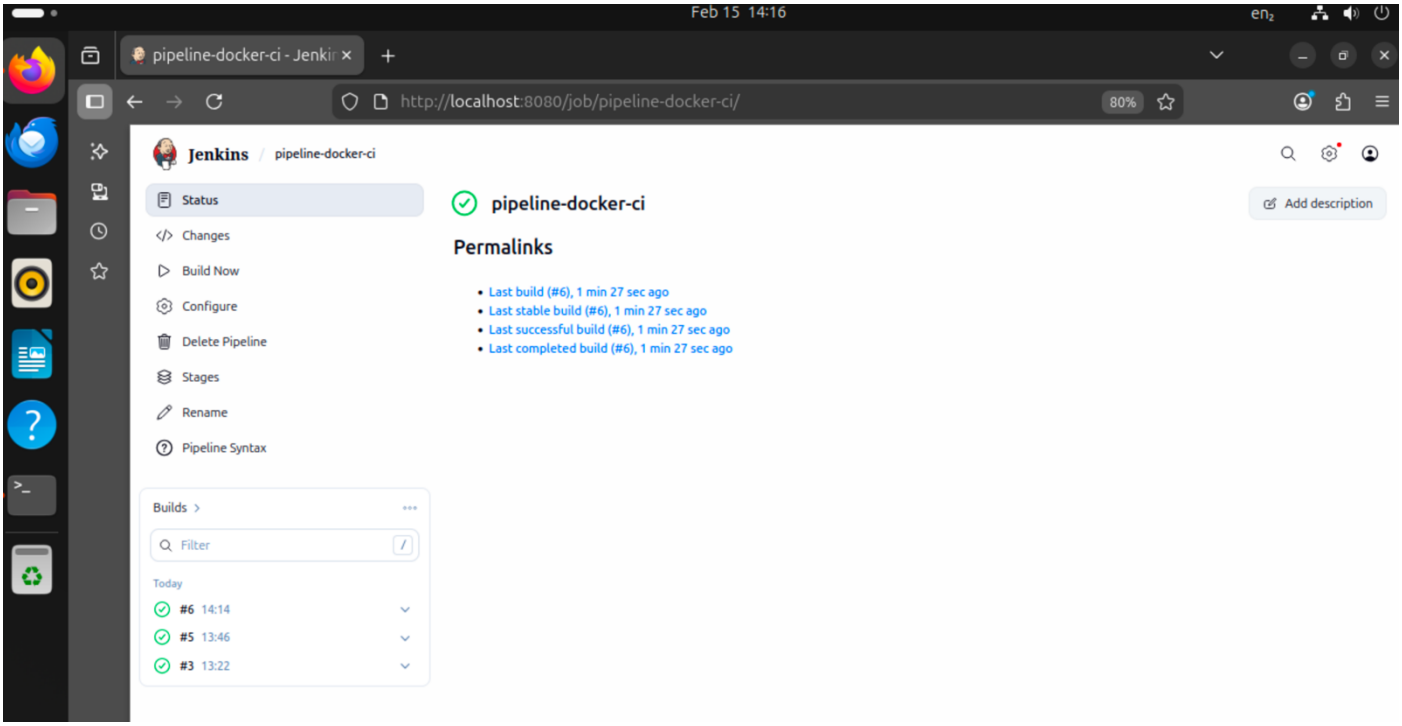
1. Job → **Configure**
2. Definition: `Pipeline script from SCM`
3. SCM: Git
4. Repo URL
5. Branch: `main`
6. Script Path: `Jenkinsfile`
7. Save

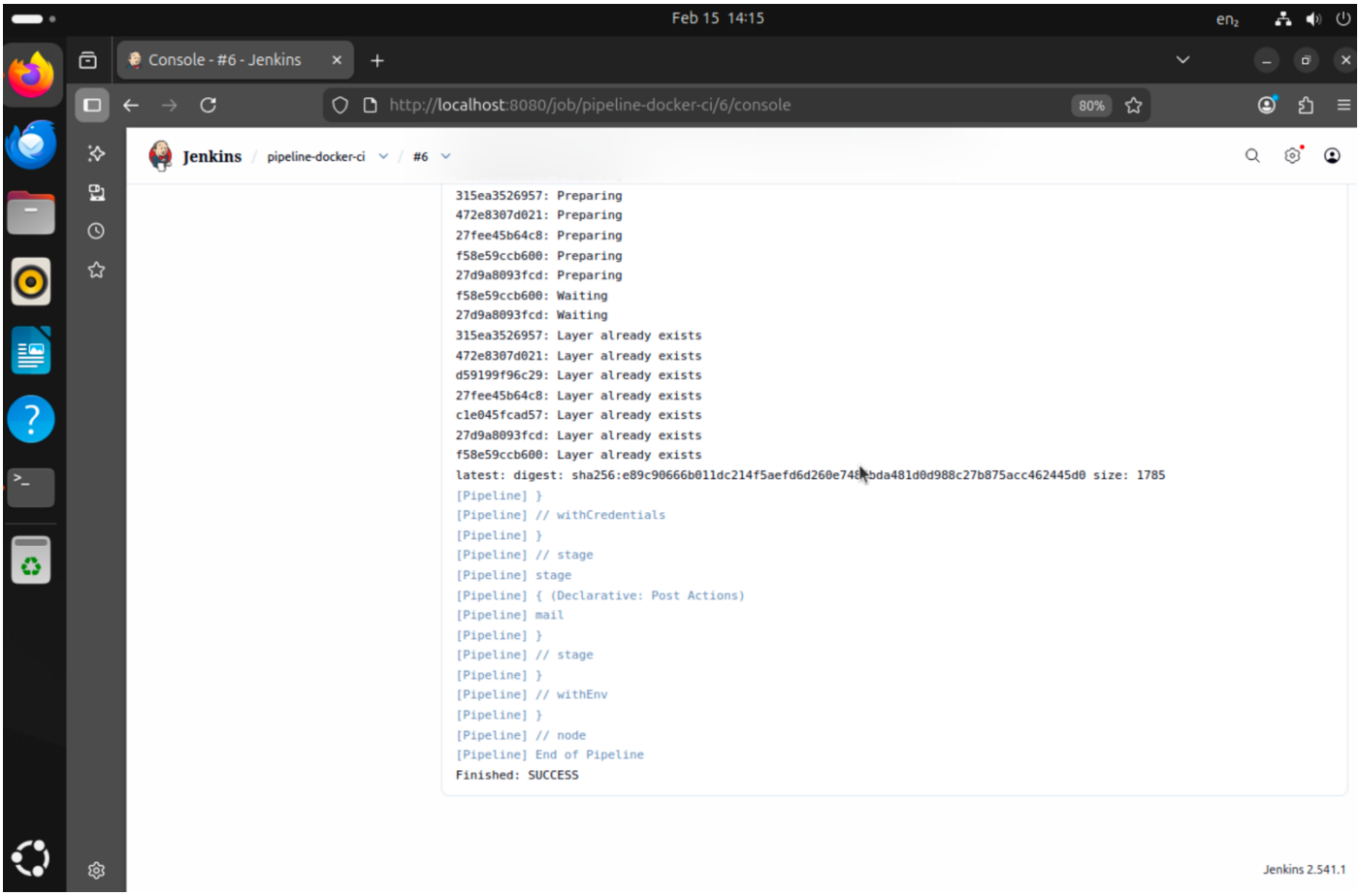




STEP B5: RUN PIPELINE

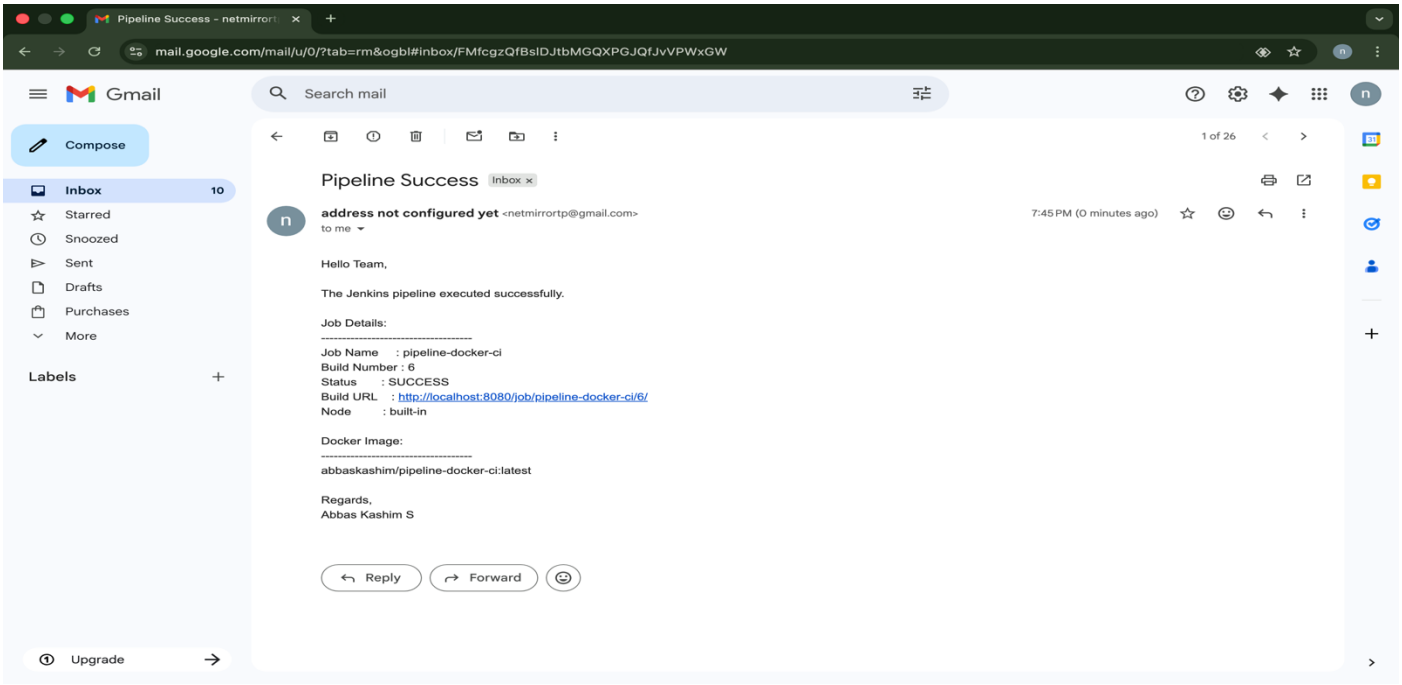
1. Click **Build Now**
2. Watch stages turn green





Step B6: Email Notification from Pipeline

1. Email sent after successful pipeline execution
2. Must include job and build detail



Part C: Comparison & Analysis

Step 11: Freestyle vs Pipeline Comparison

Criteria	Freestyle Job	Pipeline Job
Configuration Method	Configured using the Jenkins web UI (manual setup)	Configured using code written in a Jenkinsfile
Version Control	❌ Not version controlled (stored in Jenkins UI)	✅ Version controlled along with source code
Reusability	Low – difficult to reuse across jobs	High – pipeline code can be reused and shared
Complexity Handling	Suitable for simple, linear tasks	Designed for complex workflows (CI/CD pipelines)
Maintainability	Harder to maintain as changes are manual	Easy to maintain due to code-based configuration
Scalability	Limited scalability	Highly scalable for large projects
Error Handling	Basic error handling	Advanced error handling using stages and conditions
Collaboration	Limited team collaboration	Better collaboration through shared Jenkinsfile
Automation Level	Partial automation	Full CI/CD automation
Best Use Case	Small projects or beginner tasks	Enterprise-level CI/CD pipelines