TASK 5

Step 1: Clone the Git Repository

```
 # Clone the repository
git clone https://github.com/original-repo/spring-framework.git

 # Change to the project directory
 cd spring-framework

 # Add your GitHub repository as the remote
 git remote set-url origin https://github.com/DivyadharshiniA/
  spring-framework.git

# Push the project to your own repo
git push origin main
```

Step 2: Install Maven

```
 # Update package lists
 sudo apt update

 # Install Maven
 sudo apt install maven -y
mvn -version
# Remove existing Maven
sudo apt remove maven -y
```

```
# Download the latest Maven version
cd /opt
sudo wget https://downloads.apache.org/maven/maven-3/3.9.6/binaries/apache-maven-3.9.6-bin.tar.gz

# Extract and move Maven
sudo tar -xvzf apache-maven-3.9.6-bin.tar.gz
sudo mv apache-maven-3.9.6 /opt/maven

# Set up environment variables
echo 'export M2_HOME=/opt/maven' | sudo tee -a /etc/profile.d/maven.sh
echo 'export PATH=$M2_HOME/bin:$PATH' | sudo tee -a /etc/profile.d/maven.sh
source /etc/profile.d/maven.sh
```

Verify installation:

```
mvn -version
which mvn
```

If necessary, create a symbolic link:

```
sudo ln -s /opt/maven/bin/mvn /usr/bin/mvn
```

Step 3: Build the Project with Maven

Navigate to the Jenkins workspace and build the project:

```
cd /var/lib/jenkins/workspace/Spring-framework
```

```
# Clean and package the project (skipping tests)
mvn clean package -DskipTests
```

Step 4: **Set Up Jenkins Pipeline**

1. Open **Jenkins Dashboard** → Click **New Item** → Select **Pipeline**.

2. Go to **Pipeline** section and add the following script:

```
pipeline {
  agent any

  environment {
    IMAGE_NAME = "docker-user-name/my-app"
    REGISTRY = "docker.io"
    DOCKER_USER = "docker-user-name"
    DOCKER_PASS = "your-docker-password"
  }

  stages {
    stage('Checkout Code') {
      steps {
        git url: 'https://github.com/git-user-name/git-repo-name.git', branch: 'main'
      }
    }

    stage('Build Docker Image') {
      steps {
        script {
          sh "docker build -t $IMAGE_NAME:latest ."
        }
      }
    }

    stage('Login to Docker Registry') {
```

```
steps {
        script {
          sh 'echo $DOCKER_PASS | docker login -u $DOCKER_USER --password-stdin'
        }
      }
    }

    stage('Push Image to Docker Registry') {
      steps {
        script {
          sh "docker push $IMAGE_NAME:latest"
        }
      }
    }
  }

  post {
    success {
      echo 'Pipeline executed successfully!'
    }
    failure         {
      echo 'Pipeline failed! Check the logs for errors.'
    }
  }
}
```

Step 5: Fix Permissions for Jenkins

Ensure Jenkins has the correct permissions:

```
sudo chown -R jenkins:jenkins /var/lib/jenkins/workspace/Spring-framework
sudo chmod -R 775 /var/lib/jenkins/workspace/Spring-framework
```

Restart Jenkins to apply changes:

sudo systemctl restart jenkins

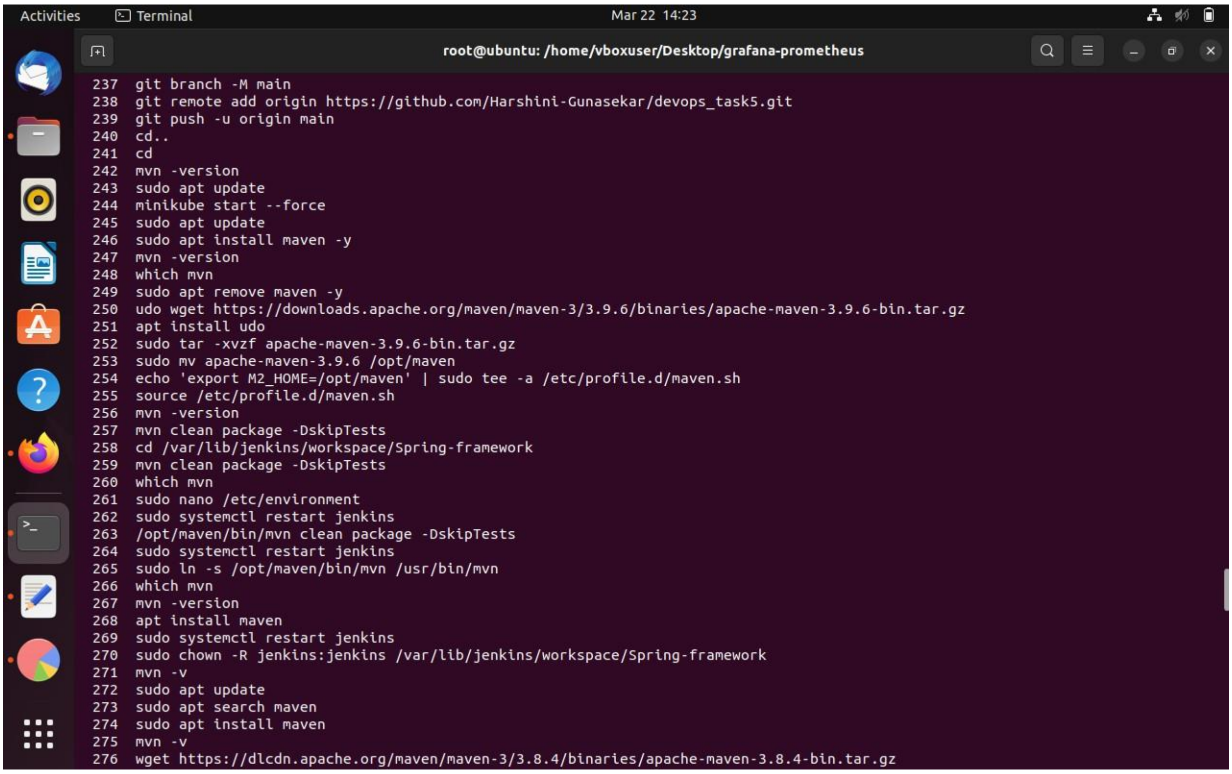
Step 6 : Run and Debug the Pipeline

After setting up everything, go to Jenkins and **trigger the build**. If there are any errors:

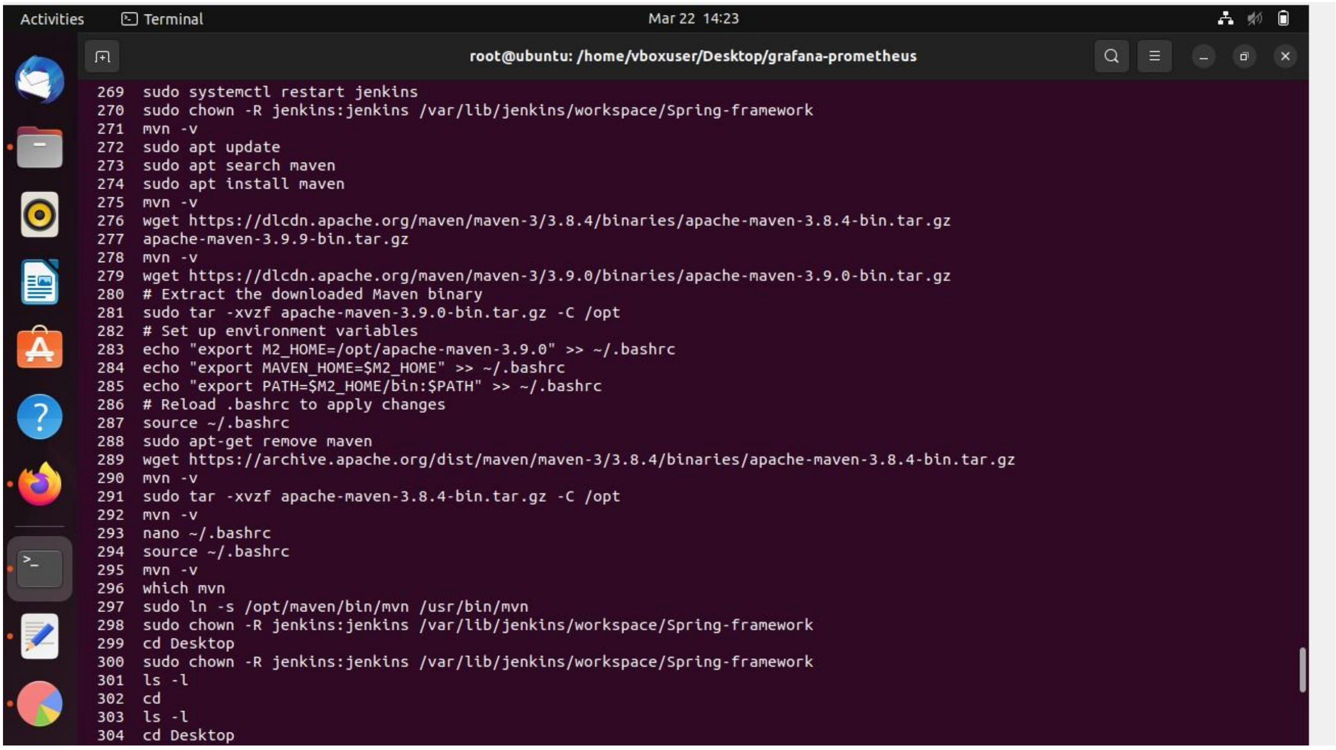
docker images # Check if the image exists

docker ps -a # Check running containers

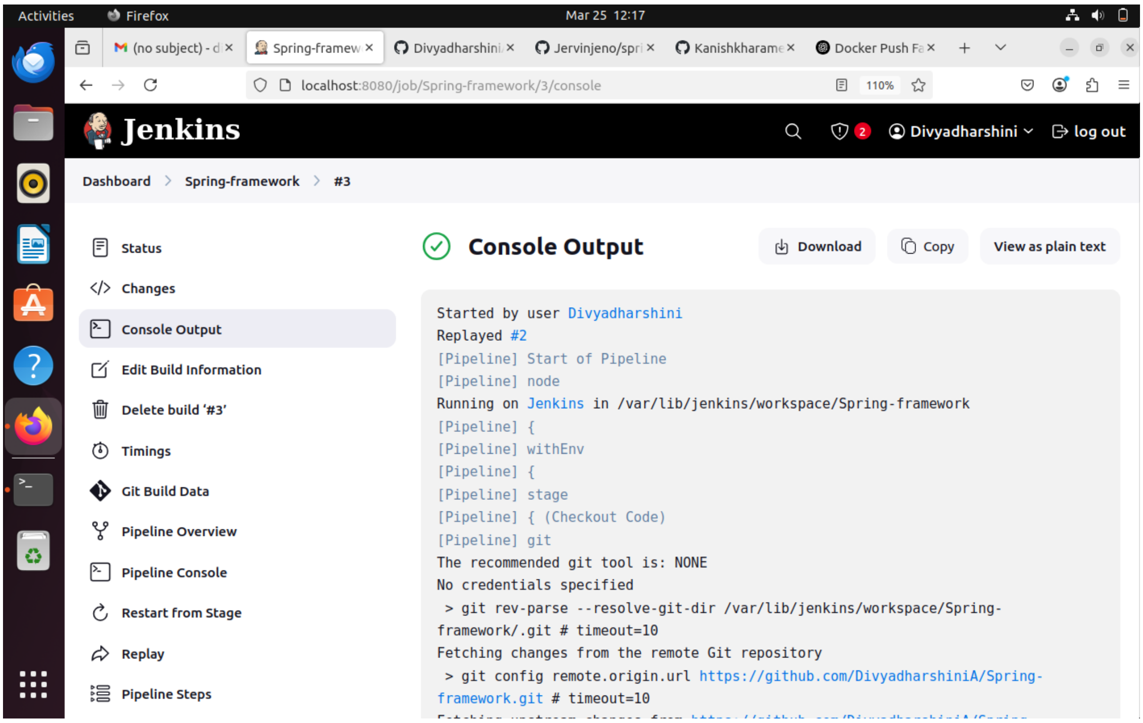
docker logs <container_id> # View container logs



```
237  git branch -M main
238  git remote add origin https://github.com/Harshini-Gunasekar/devops_task5.git
239  git push -u origin main
240  cd..
241  cd
242  mvn -version
243  sudo apt update
244  minikube start --force
245  sudo apt update
246  sudo apt install maven -y
247  mvn -version
248  which mvn
249  sudo apt remove maven -y
250  udo wget https://downloads.apache.org/maven/maven-3/3.9.6/binaries/apache-maven-3.9.6-bin.tar.gz
251  apt install udo
252  sudo tar -xvzf apache-maven-3.9.6-bin.tar.gz
253  sudo mv apache-maven-3.9.6 /opt/maven
254  echo 'export M2_HOME=/opt/maven' | sudo tee -a /etc/profile.d/maven.sh
255  source /etc/profile.d/maven.sh
256  mvn -version
257  mvn clean package -DskipTests
258  cd /var/lib/jenkins/workspace/Spring-framework
259  mvn clean package -DskipTests
260  which mvn
261  sudo nano /etc/environment
262  sudo systemctl restart jenkins
263  /opt/maven/bin/mvn clean package -DskipTests
264  sudo systemctl restart jenkins
265  sudo ln -s /opt/maven/bin/mvn /usr/bin/mvn
266  which mvn
267  mvn -version
268  apt install maven
269  sudo systemctl restart jenkins
270  sudo chown -R jenkins:jenkins /var/lib/jenkins/workspace/Spring-framework
271  mvn -v
272  sudo apt update
273  sudo apt search maven
274  sudo apt install maven
275  mvn -v
276  wget https://dlcdn.apache.org/maven/maven-3/3.8.4/binaries/apache-maven-3.8.4-bin.tar.gz
```

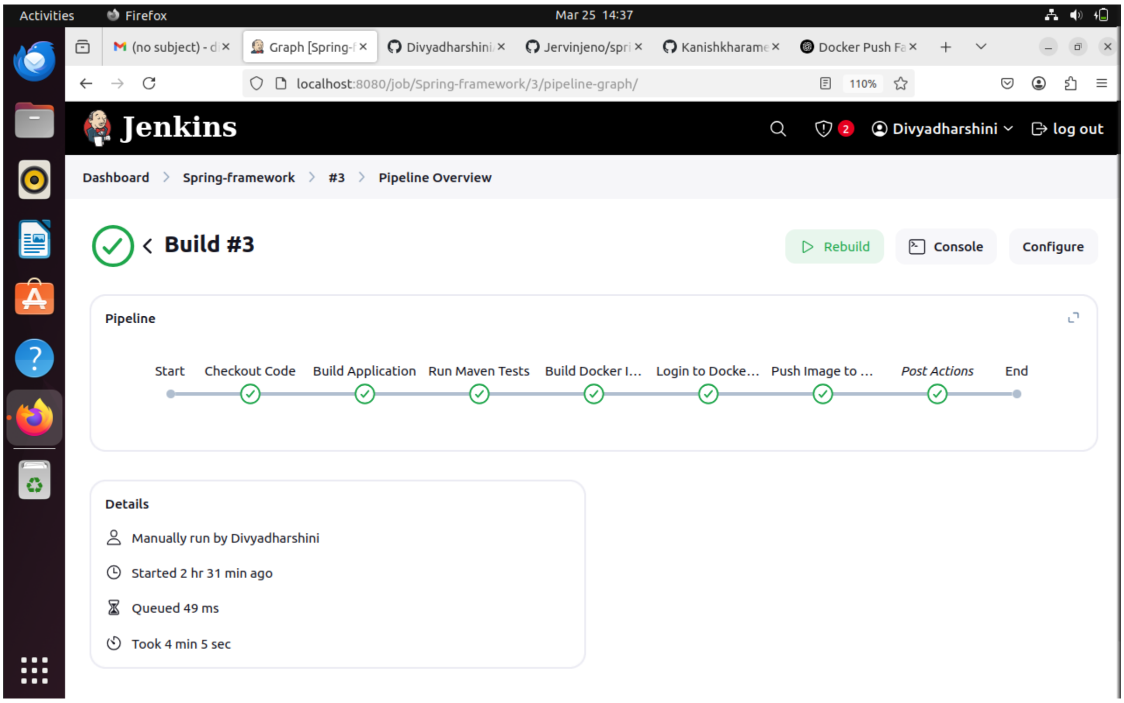


```
269  sudo systemctl restart jenkins
270  sudo chown -R jenkins:jenkins /var/lib/jenkins/workspace/Spring-framework
271  mvn -v
272  sudo apt update
273  sudo apt search maven
274  sudo apt install maven
275  mvn -v
276  wget https://dlcdn.apache.org/maven/maven-3/3.8.4/binaries/apache-maven-3.8.4-bin.tar.gz
277  apache-maven-3.9.9-bin.tar.gz
278  mvn -v
279  wget https://dlcdn.apache.org/maven/maven-3/3.9.0/binaries/apache-maven-3.9.0-bin.tar.gz
280  # Extract the downloaded Maven binary
281  sudo tar -xvzf apache-maven-3.9.0-bin.tar.gz -C /opt
282  # Set up environment variables
283  echo "export M2_HOME=/opt/apache-maven-3.9.0" >> ~/.bashrc
284  echo "export MAVEN_HOME=$M2_HOME" >> ~/.bashrc
285  echo "export PATH=$M2_HOME/bin:$PATH" >> ~/.bashrc
286  # Reload .bashrc to apply changes
287  source ~/.bashrc
288  sudo apt-get remove maven
289  wget https://archive.apache.org/dist/maven/maven-3/3.8.4/binaries/apache-maven-3.8.4-bin.tar.gz
290  mvn -v
291  sudo tar -xvzf apache-maven-3.8.4-bin.tar.gz -C /opt
292  mvn -v
293  nano ~/.bashrc
294  source ~/.bashrc
295  mvn -v
296  which mvn
297  sudo ln -s /opt/maven/bin/mvn /usr/bin/mvn
298  sudo chown -R jenkins:jenkins /var/lib/jenkins/workspace/Spring-framework
299  cd Desktop
300  sudo chown -R jenkins:jenkins /var/lib/jenkins/workspace/Spring-framework
301  ls -l
302  cd
303  ls -l
304  cd Desktop
```



Console Output

```
Started by user Divyadharshini
Replayed #2
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Spring-framework
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout Code)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Spring-
framework/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/DivyadharshiniA/Spring-
framework.git # timeout=10
```



Build #3

Details

Manually run by Divyadharshini

Started 2 hr 31 min ago

Queued 49 ms

Took 4 min 5 sec