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

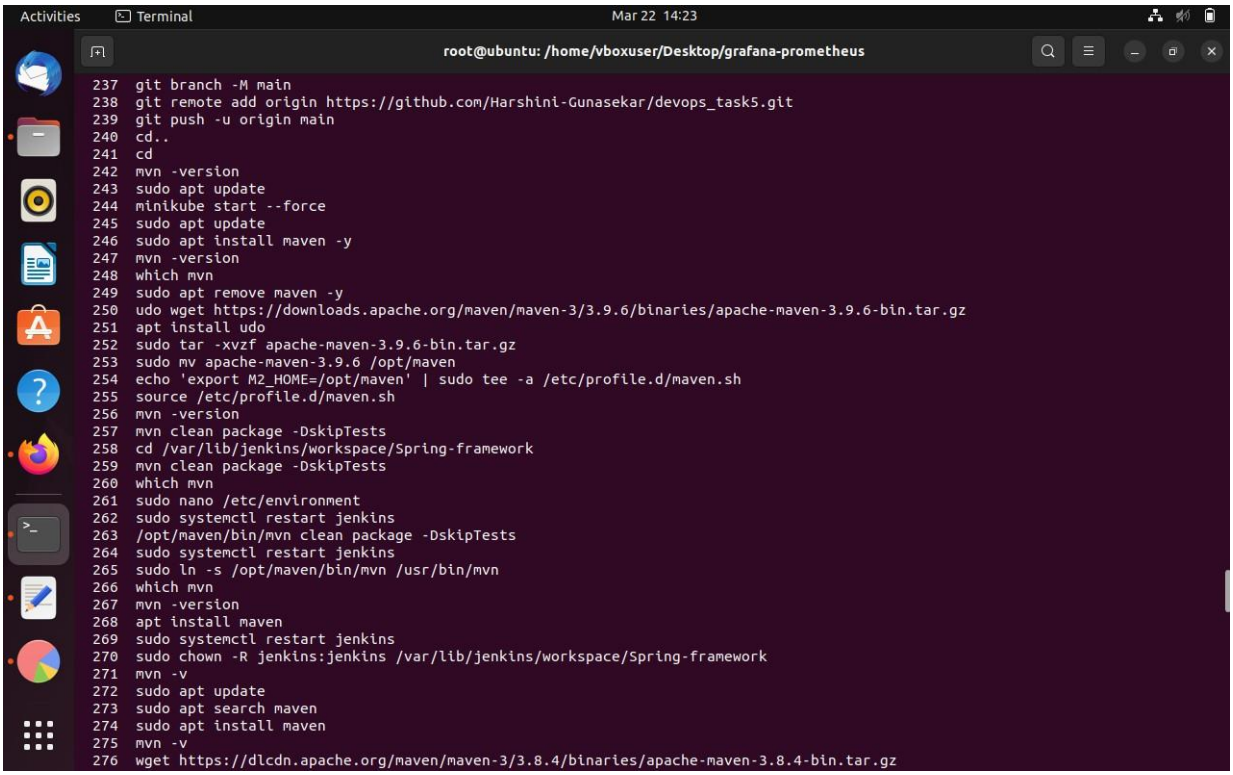
Step 6 : Run and Debug the Pipeline

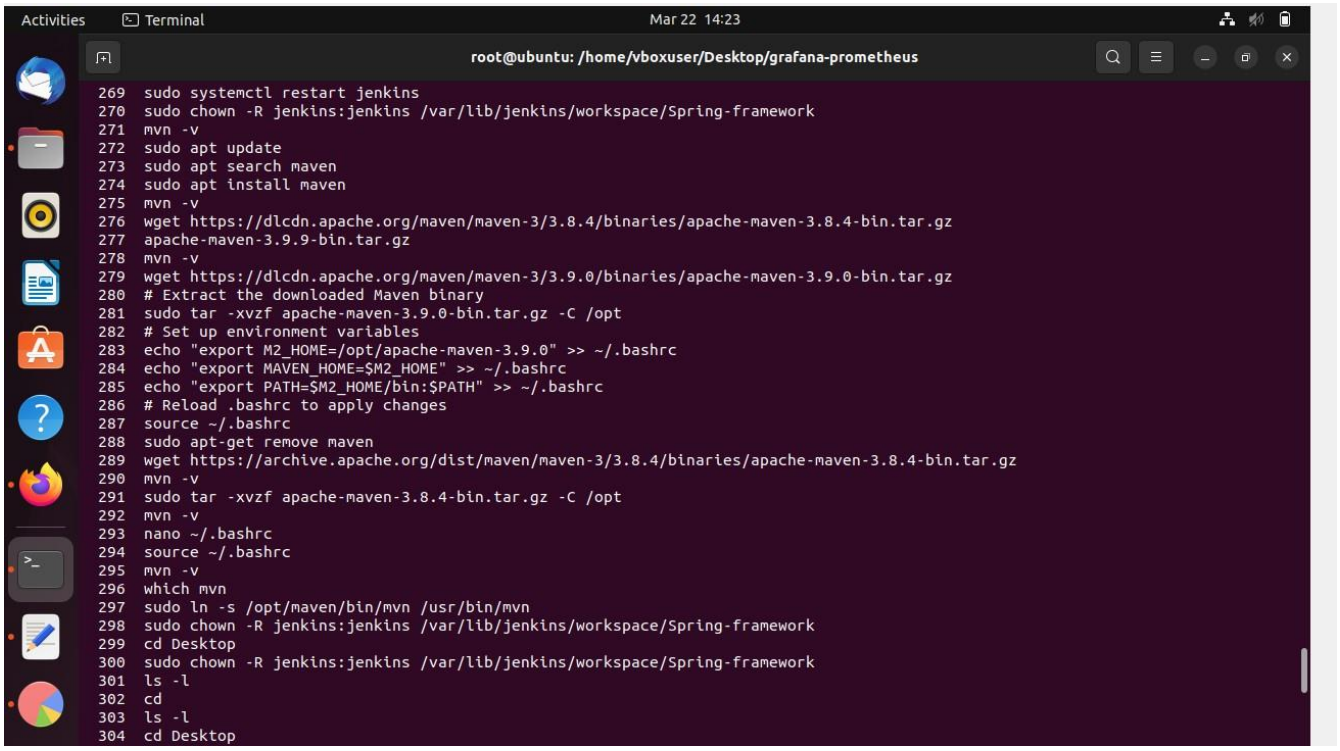
After setting up everything, go to Jenkins and **trigger the build**. If there are any errors:

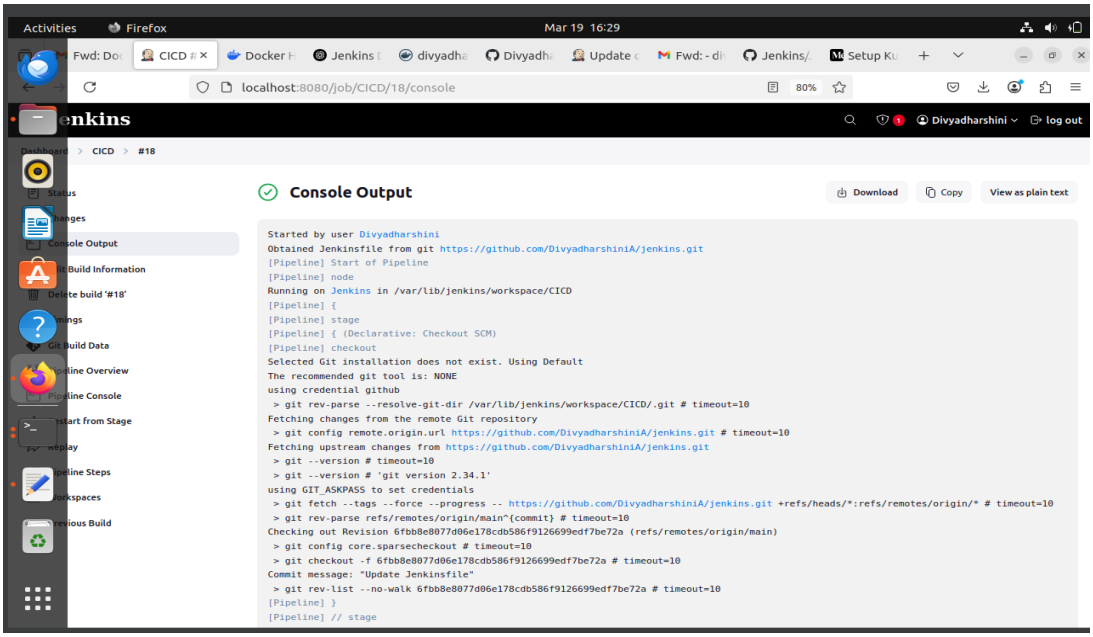
docker images # Check if the image exists

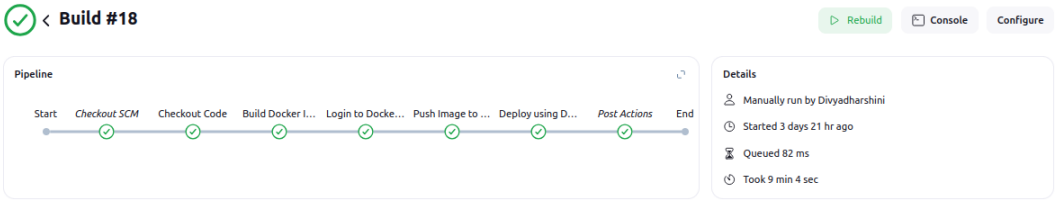
docker ps -a # Check running containers

docker logs <container_id> # View container logs




Activities
Terminal
Mar 22 14:23
root@ubuntu: /home/vboxuser/Desktop/grafana-prometheus
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


Activities
Firefox
Mar 19 16:29
localhost:8080/job/CICD/18/console
Jenkins
Divyadharshini
log out
Dashboard
CICD
#18
Console Output
Download
Copy
View as plain text
Started by user Divyadharshini
Obtained Jenkinsfile from git https://github.com/DivyadharshiniA/jenkins.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/CICD
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/CICD/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/DivyadharshiniA/jenkins.git # timeout=10
Fetching upstream changes from https://github.com/DivyadharshiniA/jenkins.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/DivyadharshiniA/jenkins.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 6fbb8e8077d06e178cdb586f9126699edf7be72a (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 6fbb8e8077d06e178cdb586f9126699edf7be72a # timeout=10
Commit message: "Update Jenkinsfile"
> git rev-list --no-walk 6fbb8e8077d06e178cdb586f9126699edf7be72a # timeout=10
[Pipeline] }
[Pipeline] // stage


Build #18
Rebuild
Console
Configure
Pipeline
Start
Checkout SCM
Checkout Code
Build Docker I...
Login to Docke...
Push Image to ...
Deploy using D...
Post Actions
End
Details
Manually run by Divyadharshini
Started 3 days 21 hr ago
Queued 82 ms
Took 9 min 4 sec