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/Dhanushree1401/javaproject

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 {
      DOCKER_IMAGE = "dhanushree14/social-app"
      DOCKER_TAG = "latest"
      DOCKER_CREDENTIALS_ID = "docker-hub-creds"
      GITHUB_CREDENTIALS_ID = "github"
      KUBECONFIG = "/var/lib/jenkins/.kube/config"
```

```
    }

    stages {
        stage('Checkout Code') {
            steps {
                git credentialsId: GITHUB_CREDENTIALS_ID, url:
'https://github.com/Dhanushree1401/javaproject.git', branch: 'main'
            }
        }

        stage('Build Application') {
            steps {
                script {
                    sh '${MAVEN_HOME}/bin/mvn clean package -DskipTests'
                }
            }
        }

        stage('Run Maven Tests') {
            steps {
                script {
                    try {
                        sh 'mvn test'
                    } catch (Exception e) {
                        echo "Tests failed, but proceeding..."
                    }
                }
            }
        }

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

        stage('Push Docker Image') {
            steps {
                withDockerRegistry([credentialsId: DOCKER_CREDENTIALS_ID, url: '']) {
                    sh "docker push ${DOCKER_IMAGE}:${DOCKER_TAG}"
                }
            }
        }

        // stage('Deploy to Kubernetes') {
        //     steps {
```

```
        //        script {
        //            sh '''
        //                chmod +x scripts/deploy.sh
        //                ./scripts/deploy.sh
        //            '''
        //        }
        //    }
        // }
    }

    post {
        success {
            echo "Deployment Successful!"
        }
        failure {
            echo "Deployment Failed!"
        }
    }
}
```

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

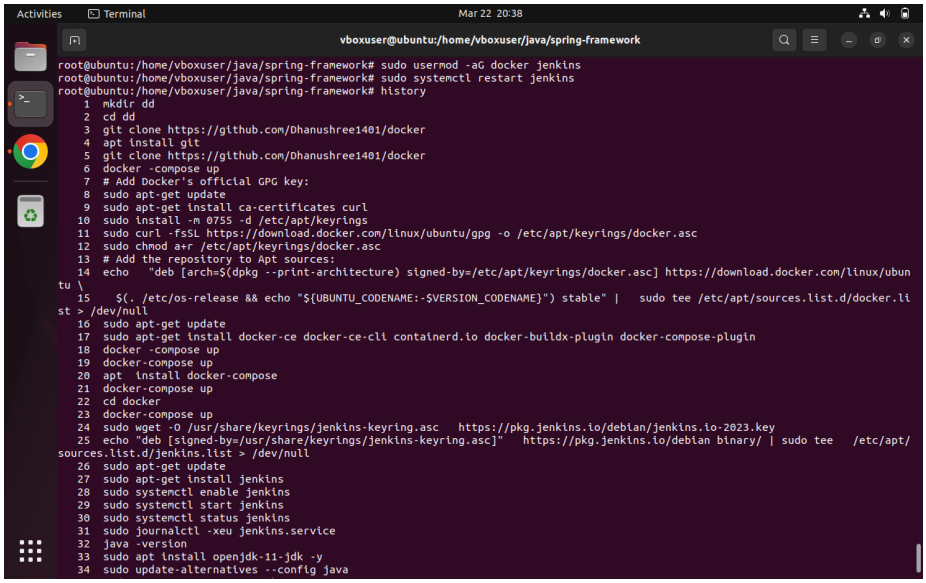
After setting up everything, go to Jenkins and trigger the build. If there

are any errors:

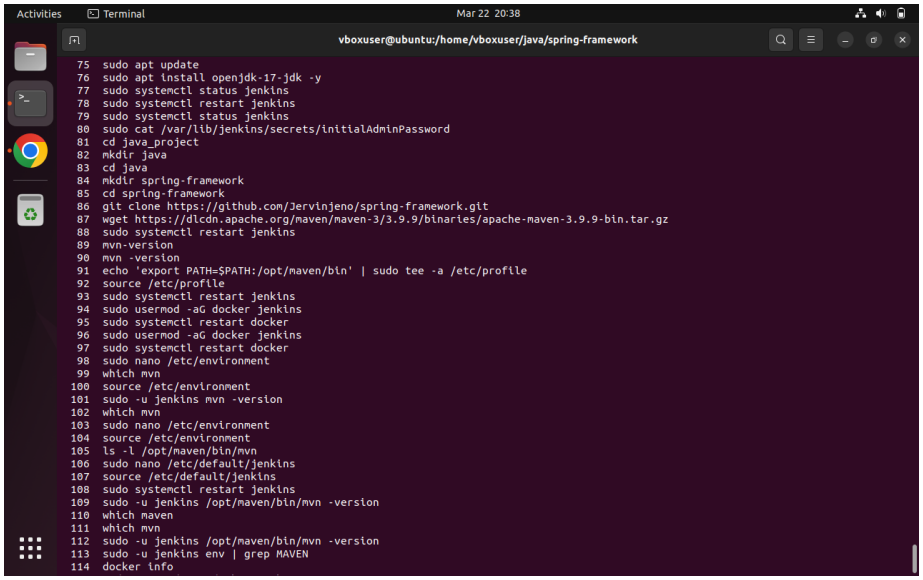
docker images # Check if the image exists

docker ps -a # Check running containers

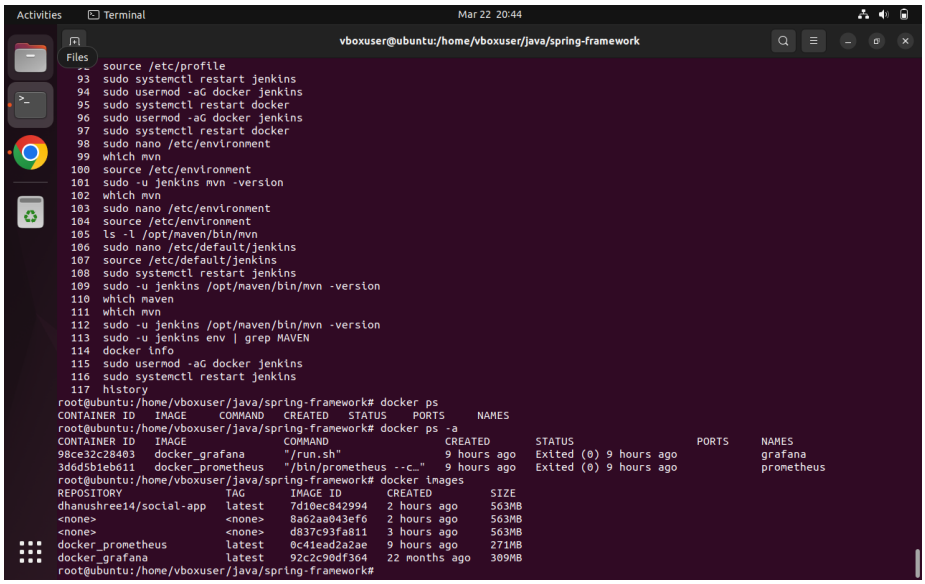
docker logs <container_id> # View container logs



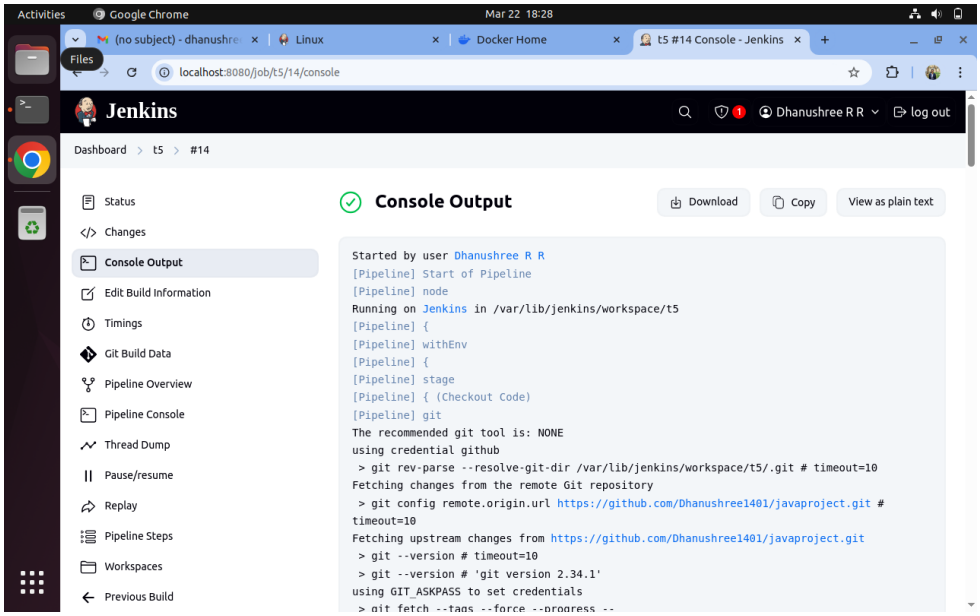

```
root@ubuntu:/home/vboxuser/java/spring-framework# sudo usermod -aG docker jenkins
root@ubuntu:/home/vboxuser/java/spring-framework# sudo systemctl restart jenkins
root@ubuntu:/home/vboxuser/java/spring-framework# history
    1  mkdir dd
    2  cd dd
    3  git clone https://github.com/Dhanushree1401/docker
    4  apt install git
    5  git clone https://github.com/Dhanushree1401/docker
    6  docker -compose up
    7  # Add Docker's official GPG key:
    8  sudo apt-get update
    9  sudo apt-get install ca-certificates curl
   10  sudo install -m 0755 -d /etc/apt/keyrings
   11  sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
   12  sudo chmod a+r /etc/apt/keyrings/docker.asc
   13  # Add the repository to Apt sources:
   14  echo   "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubun
tu \
   15    $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" |   sudo tee /etc/apt/sources.list.d/docker.li
st > /dev/null
   16  sudo apt-get update
   17  sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
   18  docker -compose up
   19  docker-compose up
   20  apt  install docker-compose
   21  docker-compose up
   22  cd docker
   23  docker-compose up
   24  sudo wget -O /usr/share/keyrings/jenkins-keyring.asc   https://pkg.jenkins.io/debian/jenkins.io-2023.key
   25  echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]"   https://pkg.jenkins.io/debian binary/ | sudo tee   /etc/apt/
sources.list.d/jenkins.list > /dev/null
   26  sudo apt-get update
   27  sudo apt-get install jenkins
   28  sudo systemctl enable jenkins
   29  sudo systemctl start jenkins
   30  sudo systemctl status jenkins
   31  sudo journalctl -xeu jenkins.service
   32  java -version
   33  sudo apt install openjdk-11-jdk -y
   34  sudo update-alternatives --config java
```

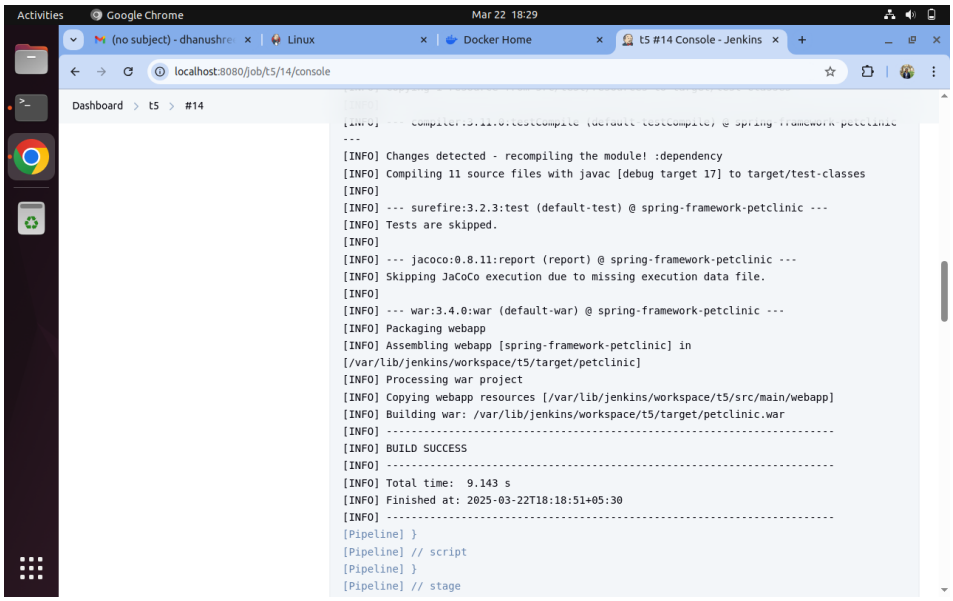


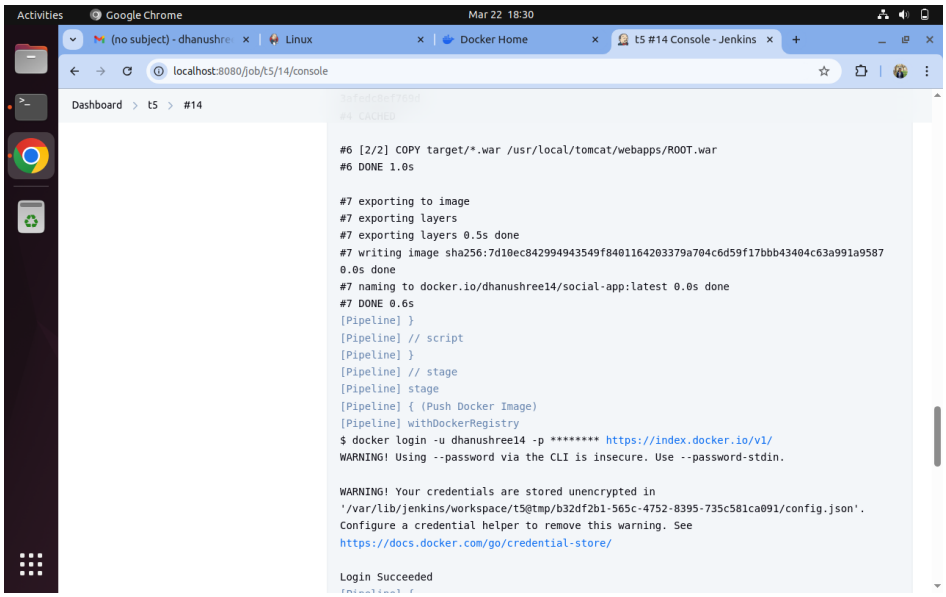

```
   75  sudo apt update
   76  sudo apt install openjdk-17-jdk -y
   77  sudo systemctl status jenkins
   78  sudo systemctl restart jenkins
   79  sudo systemctl status jenkins
   80  sudo cat /var/lib/jenkins/secrets/initialAdminPassword
   81  cd java_project
   82  mkdir java
   83  cd java
   84  mkdir spring-framework
   85  cd spring-framework
   86  git clone https://github.com/Jervinjeno/spring-framework.git
   87  wget https://dlcdn.apache.org/maven/maven-3/3.9.9/binaries/apache-maven-3.9.9-bin.tar.gz
   88  sudo systemctl restart jenkins
   89  mvn-version
   90  mvn -version
   91  echo 'export PATH=$PATH:/opt/maven/bin' | sudo tee -a /etc/profile
   92  source /etc/profile
   93  sudo systemctl restart jenkins
   94  sudo usermod -aG docker jenkins
   95  sudo systemctl restart docker
   96  sudo usermod -aG docker jenkins
   97  sudo systemctl restart docker
   98  sudo nano /etc/environment
   99  which mvn
  100  source /etc/environment
  101  sudo -u jenkins mvn -version
  102  which mvn
  103  sudo nano /etc/environment
  104  source /etc/environment
  105  ls -l /opt/maven/bin/mvn
  106  sudo nano /etc/default/jenkins
  107  source /etc/default/jenkins
  108  sudo systemctl restart jenkins
  109  sudo -u jenkins /opt/maven/bin/mvn -version
  110  which maven
  111  which mvn
  112  sudo -u jenkins /opt/maven/bin/mvn -version
  113  sudo -u jenkins env | grep MAVEN
  114  docker info
```

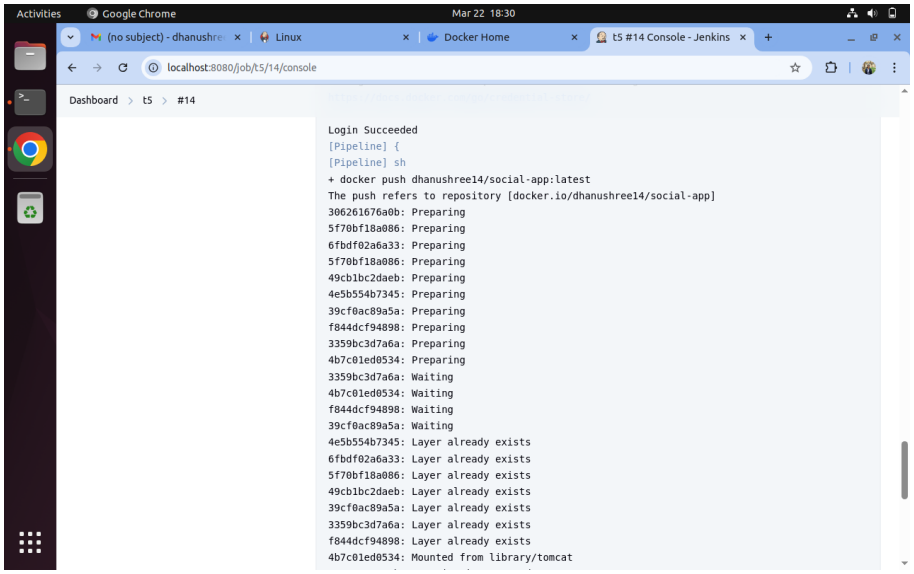


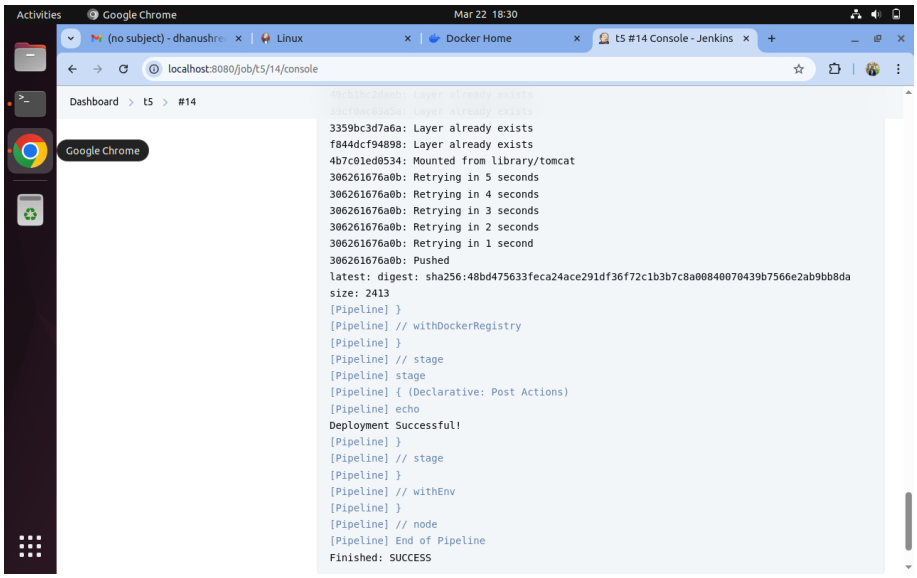

```
   92  source /etc/profile
   93  sudo systemctl restart jenkins
   94  sudo usermod -aG docker jenkins
   95  sudo systemctl restart docker
   96  sudo usermod -aG docker jenkins
   97  sudo systemctl restart docker
   98  sudo nano /etc/environment
   99  which mvn
  100  source /etc/environment
  101  sudo -u jenkins mvn -version
  102  which mvn
  103  sudo nano /etc/environment
  104  source /etc/environment
  105  ls -l /opt/maven/bin/mvn
  106  sudo nano /etc/default/jenkins
  107  source /etc/default/jenkins
  108  sudo systemctl restart jenkins
  109  sudo -u jenkins /opt/maven/bin/mvn -version
  110  which maven
  111  which mvn
  112  sudo -u jenkins /opt/maven/bin/mvn -version
  113  sudo -u jenkins env | grep MAVEN
  114  docker info
  115  sudo usermod -aG docker jenkins
  116  sudo systemctl restart jenkins
  117  history
root@ubuntu:/home/vboxuser/java/spring-framework# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@ubuntu:/home/vboxuser/java/spring-framework# docker ps -a
CONTAINER ID   IMAGE              COMMAND                  CREATED       STATUS                   PORTS     NAMES
98ce32c28403   docker_grafana     "/run.sh"                9 hours ago   Exited (0) 9 hours ago             grafana
3d6d5b1eb611   docker_prometheus  "/bin/prometheus --c…"   9 hours ago   Exited (0) 9 hours ago             prometheus
root@ubuntu:/home/vboxuser/java/spring-framework# docker images
REPOSITORY                  TAG       IMAGE ID       CREATED         SIZE
dhanushree14/social-app     latest    7d10ec842994   2 hours ago     563MB
<none>                      <none>    8a62aa043ef6   2 hours ago     563MB
<none>                      <none>    d837c93fa811   3 hours ago     563MB
docker_prometheus           latest    0c41ead2a2ae   9 hours ago     271MB
docker_grafana              latest    92c2c90df364   22 months ago   309MB
root@ubuntu:/home/vboxuser/java/spring-framework#
```

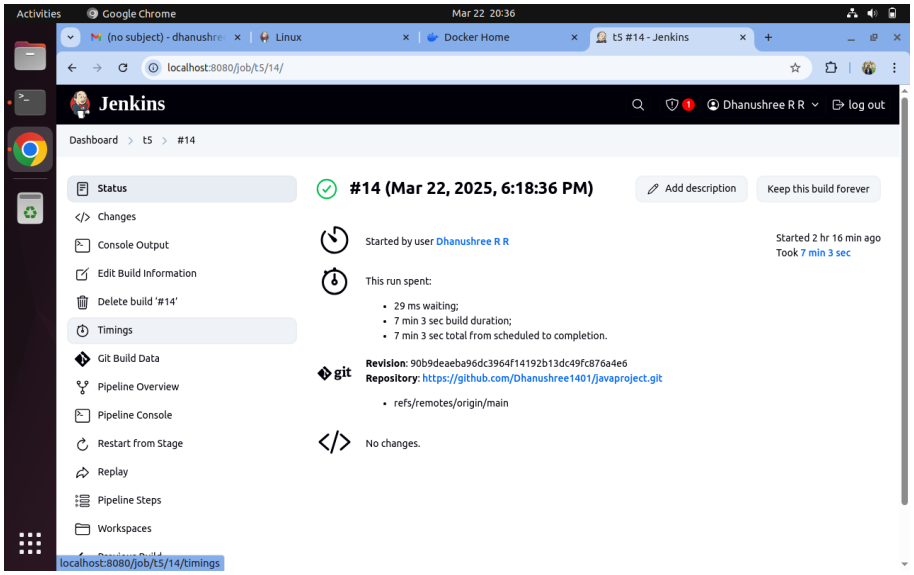

Activities
Google Chrome
Mar 22 18:28
(no subject) - dhanushre
Linux
Docker Home
t5 #14 Console - Jenkins
localhost:8080/job/t5/14/console
Jenkins
Dhanushree R R
log out
Dashboard
t5
#14
Status
Changes
Console Output
Edit Build Information
Timings
Git Build Data
Pipeline Overview
Pipeline Console
Thread Dump
Pause/resume
Replay
Pipeline Steps
Workspaces
Previous Build
Console Output
Download
Copy
View as plain text
Started by user Dhanushree R R
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/t5
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout Code)
[Pipeline] git
The recommended git tool is: NONE
using credential github
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/t5/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Dhanushree1401/javaproject.git # timeout=10
Fetching upstream changes from https://github.com/Dhanushree1401/javaproject.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials


Activities
Google Chrome
Mar 22 18:29
(no subject) - dhanushre
Linux
Docker Home
t5 #14 Console - Jenkins
localhost:8080/job/t5/14/console
Dashboard
t5
#14
[INFO] Changes detected - recompiling the module! :dependency
[INFO] Compiling 11 source files with javac [debug target 17] to target/test-classes
[INFO]
[INFO] --- surefire:3.2.3:test (default-test) @ spring-framework-petclinic ---
[INFO] Tests are skipped.
[INFO]
[INFO] --- jacoco:0.8.11:report (report) @ spring-framework-petclinic ---
[INFO] Skipping JaCoCo execution due to missing execution data file.
[INFO]
[INFO] --- war:3.4.0:war (default-war) @ spring-framework-petclinic ---
[INFO] Packaging webapp
[INFO] Assembling webapp [spring-framework-petclinic] in
[/var/lib/jenkins/workspace/t5/target/petclinic]
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/t5/src/main/webapp]
[INFO] Building war: /var/lib/jenkins/workspace/t5/target/petclinic.war
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 9.143 s
[INFO] Finished at: 2025-03-22T18:18:51+05:30
[INFO] --
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage


Activities
Google Chrome
Mar 22 18:30
(no subject) - dhanushre
Linux
Docker Home
t5 #14 Console - Jenkins
localhost:8080/job/t5/14/console
Dashboard
t5
#14
#6 [2/2] COPY target/*.war /usr/local/tomcat/webapps/ROOT.war
#6 DONE 1.0s
#7 exporting to image
#7 exporting layers
#7 exporting layers 0.5s done
#7 writing image sha256:7d10ec842994943549f8401164203379a704c6d59f17bbb43404c63a991a9587 0.0s done
#7 naming to docker.io/dhanushree14/social-app:latest 0.0s done
#7 DONE 0.6s
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Push Docker Image)
[Pipeline] withDockerRegistry
$ docker login -u dhanushree14 -p ******** https://index.docker.io/v1/
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your credentials are stored unencrypted in
'/var/lib/jenkins/workspace/t5@tmp/b32df2b1-565c-4752-8395-735c581ca091/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/
Login Succeeded


Activities
Google Chrome
Mar 22 18:30
(no subject) - dhanushre
Linux
Docker Home
t5 #14 Console - Jenkins
localhost:8080/job/t5/14/console
Dashboard
t5
#14
Login Succeeded
[Pipeline] {
[Pipeline] sh
+ docker push dhanushree14/social-app:latest
The push refers to repository [docker.io/dhanushree14/social-app]
306261676a0b: Preparing
5f70bf18a086: Preparing
6fbdf02a6a33: Preparing
5f70bf18a086: Preparing
49cb1bc2daeb: Preparing
4e5b554b7345: Preparing
39cf0ac89a5a: Preparing
f844dcf94898: Preparing
3359bc3d7a6a: Preparing
4b7c01ed0534: Preparing
3359bc3d7a6a: Waiting
4b7c01ed0534: Waiting
f844dcf94898: Waiting
39cf0ac89a5a: Waiting
4e5b554b7345: Layer already exists
6fbdf02a6a33: Layer already exists
5f70bf18a086: Layer already exists
49cb1bc2daeb: Layer already exists
39cf0ac89a5a: Layer already exists
3359bc3d7a6a: Layer already exists
f844dcf94898: Layer already exists
4b7c01ed0534: Mounted from library/tomcat


Activities
Google Chrome
Mar 22 18:30
(no subject) - dhanushre
Linux
Docker Home
t5 #14 Console - Jenkins
localhost:8080/job/t5/14/console
Dashboard
t5
#14
Google Chrome
3359bc3d7a6a: Layer already exists
f844dcf94898: Layer already exists
4b7c01ed0534: Mounted from library/tomcat
306261676a0b: Retrying in 5 seconds
306261676a0b: Retrying in 4 seconds
306261676a0b: Retrying in 3 seconds
306261676a0b: Retrying in 2 seconds
306261676a0b: Retrying in 1 second
306261676a0b: Pushed
latest: digest: sha256:48bd475633feca24ace291df36f72c1b3b7c8a00840070439b7566e2ab9bb8da
size: 2413
[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Deployment Successful!
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS


Activities
Google Chrome
Mar 22 20:36
(no subject) - dhanushre
Linux
Docker Home
t5 #14 - Jenkins
localhost:8080/job/t5/14/
Jenkins
Dhanushree R R
log out
Dashboard
t5
#14
Status
Changes
Console Output
Edit Build Information
Delete build '#14'
Timings
Git Build Data
Pipeline Overview
Pipeline Console
Restart from Stage
Replay
Pipeline Steps
Workspaces
#14 (Mar 22, 2025, 6:18:36 PM)
Add description
Keep this build forever
Started by user Dhanushree R R
Started 2 hr 16 min ago
Took 7 min 3 sec
This run spent:
29 ms waiting;
7 min 3 sec build duration;
7 min 3 sec total from scheduled to completion.
Revision: 90b9deaeba96dc3964f14192b13dc49fc876a4e6
Repository: https://github.com/Dhanushree1401/javaproject.git
refs/remotes/origin/main
No changes.
localhost:8080/job/t5/14/timings