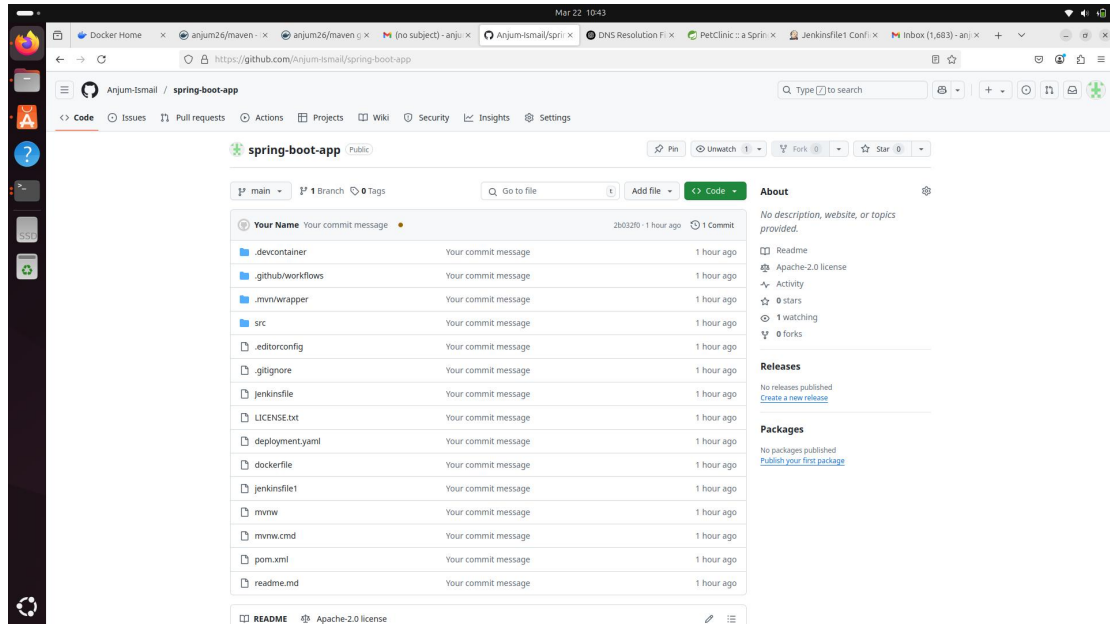


## TASK -5

**PUSH THE GIVEN REPO GIVEN INTO YOUR GITHUB:-**



**THEN USE THE BELOW PIPELINESCRIPT ACCORDING TO YOUR OWN CREDENTIALS:-**

```
pipeline {
  agent any

  environment {
    DOCKER_IMAGE = "anjum26/maven"
    DOCKER_TAG = "latest"
    DOCKER_CREDENTIALS_ID = "docker"
    GITHUB_CREDENTIALS_ID = "github_secured"
    KUBECONFIG = "/var/lib/jenkins/.kube/config"
    MAVEN_HOME = "/usr/share/maven"
  }

  stages {
    stage('Checkout Code') {
      steps {
        script {
          echo "Checking out code from GitHub..."
          git credentialsId: GITHUB_CREDENTIALS_ID, url: 'https://github.com/Anjum-Ismail/spring-boot-app.git', branch:
'main'
        }
      }
    }

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

    stage('Run Maven Tests') {
      steps {
        script {
          echo "Running tests..."
          catchError(buildResult: 'SUCCESS', stageResult: 'FAILURE') {
            sh "${MAVEN_HOME}/bin/mvn test"
          }
        }
      }
    }
  }
}
```

```

    }
  }
}

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

stage('Push Docker Image') {
  steps {
    script {
      echo " Pushing Docker image to registry..."
    }
    withDockerRegistry([credentialsId: DOCKER_CREDENTIALS_ID, url: ""]) {
      sh "docker push ${DOCKER_IMAGE}:${DOCKER_TAG}"
    }
  }
}

// Uncomment this stage when Kubernetes deployment is ready
// stage('Deploy to Kubernetes') {
//   steps {
//     script {
//       echo " Deploying to Kubernetes..."
//       sh ""
//       chmod +x scripts/deploy.sh
//       ./scripts/deploy.sh
//     }
//   }
// }

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

```

## BUILD THE PIPELINE:-

### In Tools:-

Use Maven 3.9.6

and Maven Home as /opt/home

The screenshot shows the Jenkins web interface for a pipeline named 'Jenkinsfile1'. The left sidebar contains navigation links: Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Stages, Rename, and Pipeline Syntax. The main area displays the 'Stage View' with a table of stages and their durations. Below the table, there are 'Permalinks' for various build states.

Stage	Checkout Code	Build Application	Run Maven Tests	Build Docker Image	Push Docker Image	Declarative: Post Actions
Average stage times: (Full run time: ~10min 57s)	3s	1min 54s	12s	4min 55s	3min 49s	52ms
1012 No Changes	3s	1min 54s	12s	4min 55s	3min 49s	52ms
1008 No Changes						

**Permalinks**

- Last build (#4), 25 min ago
- Last stable build (#4), 25 min ago
- Last successful build (#4), 25 min ago
- Last failed build (#3), 28 min ago
- Last unsuccessful build (#3), 28 min ago
- Last completed build (#4), 25 min ago

REST API Jenkins 2.492.2

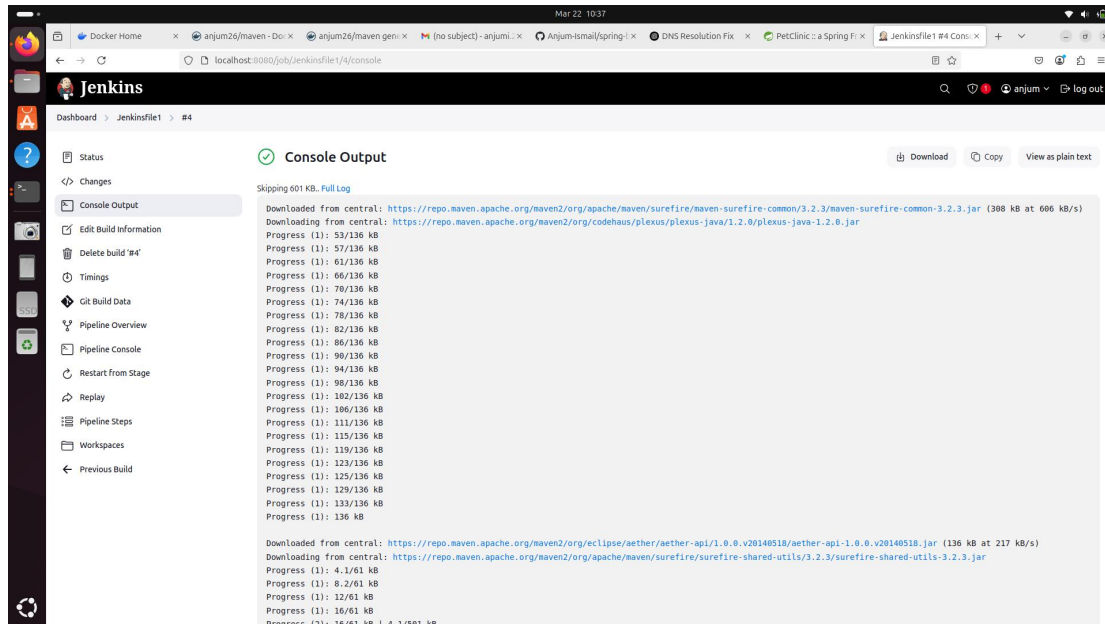
```
anjan@anjan-IdeaPad-5-15IAL7: ~/Downloads/spring-boot-app-main(1)/spring-boot-app-main
anjan@anjan-IdeaPad-5-15IAL7: ~/Downloads/spring-boot-app-main(1)/spring-boot-app-main$ sudo systemctl start jenkins
anjan@anjan-IdeaPad-5-15IAL7: ~/Downloads/spring-boot-app-main(1)/spring-boot-app-main$ sudo systemctl status jenkins
● Jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: 
   Active: active (running) since Sat 2025-03-22 08:56:16 IST; 1h 6min ago
     Main PID: 1322 (java)
       Tasks: 55 (limit: 18655)
      Memory: 772.8M (peak: 783.1M)
         CPU: 1min 17.843s
        CGroup: /system.slice/jenkins.service
                └─1322 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/

Mar 22 08:56:16 anjan-IdeaPad-5-15IAL7 jenkins[1322]: at hudson.util.Re
Mar 22 08:56:16 anjan-IdeaPad-5-15IAL7 jenkins[1322]: at hudson.Plugin
Mar 22 08:56:16 anjan-IdeaPad-5-15IAL7 jenkins[1322]: at jenkins.Daily
Mar 22 08:56:16 anjan-IdeaPad-5-15IAL7 jenkins[1322]: at hudson.model.A
Mar 22 08:56:16 anjan-IdeaPad-5-15IAL7 jenkins[1322]: at java.base/java
Mar 22 08:56:16 anjan-IdeaPad-5-15IAL7 jenkins[1322]: 2025-03-22 03:26:16.0444
Mar 22 08:56:16 anjan-IdeaPad-5-15IAL7 jenkins[1322]: 2025-03-22 03:26:16.0474
Mar 22 08:56:16 anjan-IdeaPad-5-15IAL7 jenkins[1322]: 2025-03-22 03:26:16.0474
Mar 22 08:56:16 anjan-IdeaPad-5-15IAL7 jenkins[1322]: 2025-03-22 03:26:16.0484
lines 1-20/20 (END)
anjan@anjan-IdeaPad-5-15IAL7: ~/Downloads/spring-boot-app-main(1)/spring-boot-app-main$ mvn -version
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 21.0.6, vendor: Ubuntu, runtime: /usr/lib/jvm/java-21-openjdk-amd64
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "6.11.0-19-generic", arch: "amd64", family: "unix"
anjan@anjan-IdeaPad-5-15IAL7: ~/Downloads/spring-boot-app-main(1)/spring-boot-app-main$ ^C
anjan@anjan-IdeaPad-5-15IAL7: ~/Downloads/spring-boot-app-main(1)/spring-boot-app-main$ docker ps
CONTAINER ID   IMAGE          COMMAND                  STATUS    PORTS    NAMES
anjan@anjan-IdeaPad-5-15IAL7: ~/Downloads/spring-boot-app-main(1)/spring-boot-app-main$ docker run -d -p 9098:8080
"docker run" requires at least 1 argument.
See "docker run --help".

Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Create and run a new container from an image
anjan@anjan-IdeaPad-5-15IAL7: ~/Downloads/spring-boot-app-main(1)/spring-boot-app-main$ docker run -d -p 9098:8080 --name ny-app anjan26/maven
anjan@anjan-IdeaPad-5-15IAL7: ~/Downloads/spring-boot-app-main(1)/spring-boot-app-main$
```

## BUILD SUCCESS OUTPUT:-



## FINAL OUTPUT IN localhost:9098

