**Implement a CI/CD pipeline using Jenkins multi-branch pipelines, SonarQube for code quality, Docker for containerization, GitHub webhooks and Kubernetes for deployment.**

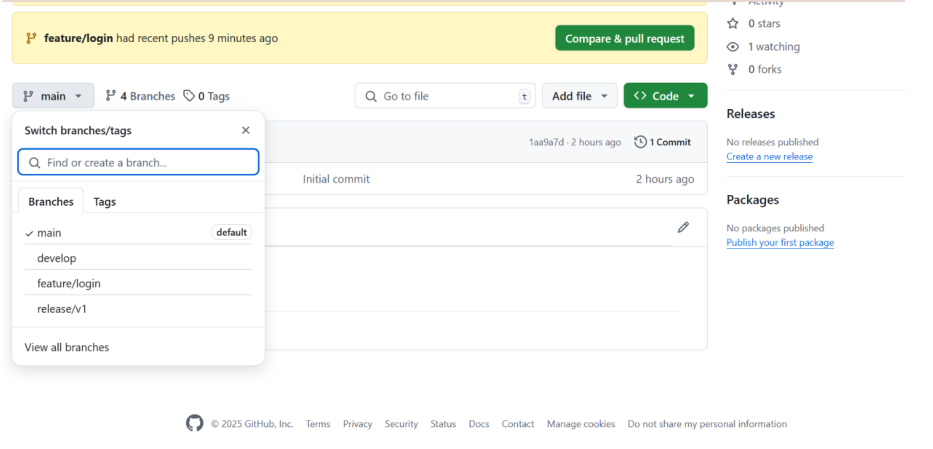
**Step 1: Created a repository name “CI-CD-MultiBranch in my Github**

**Github url :** [**https://github.com/Srikar2610/CI-CD-MultiBranch.git**](https://github.com/Srikar2610/CI-CD-MultiBranch.git)

****

**I.Branching Strategy and Execution rules**

**Step 2: Make into different branches for MultiBranch pipeline by compare and Pull request**

****

**Here you can see the 3 branches they are as follows –**

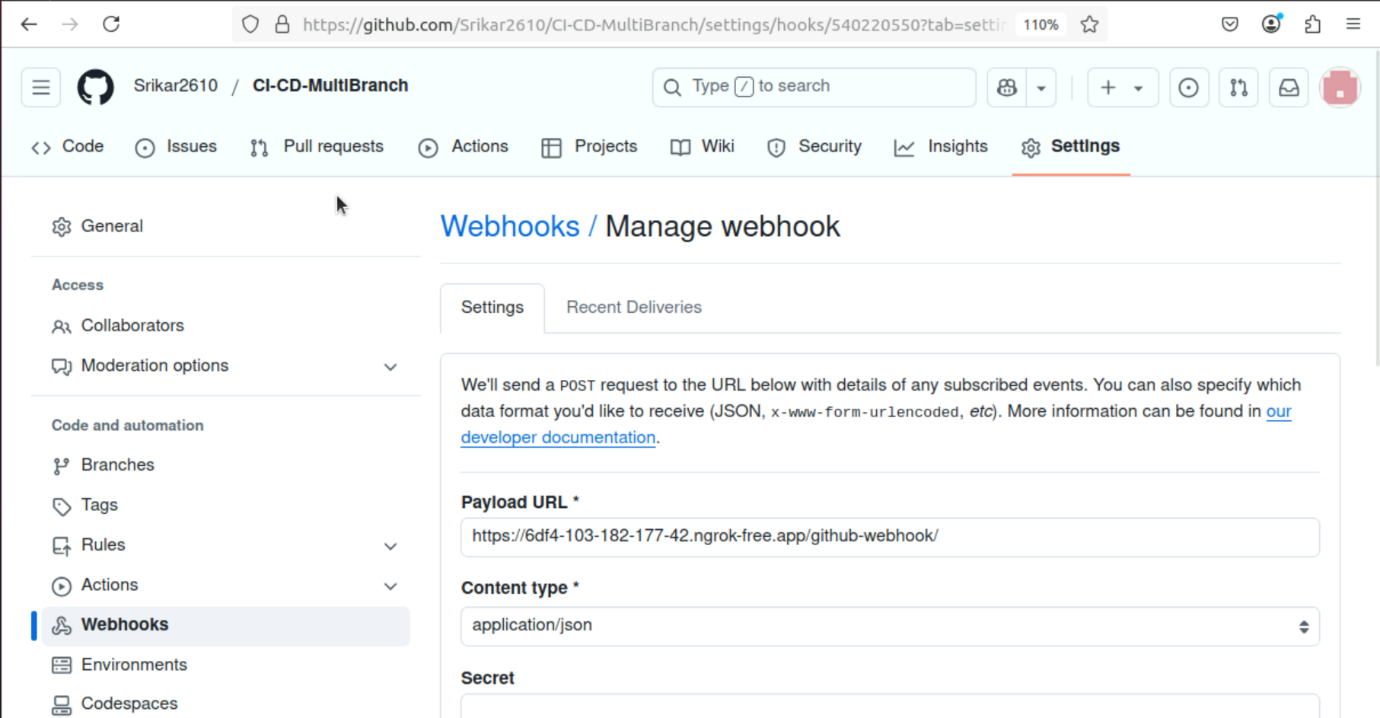
**develop, feature/login and release/v1**

**A screenshot of a computer

AI-generated content may be incorrect.**

**II. Webhook Integration**

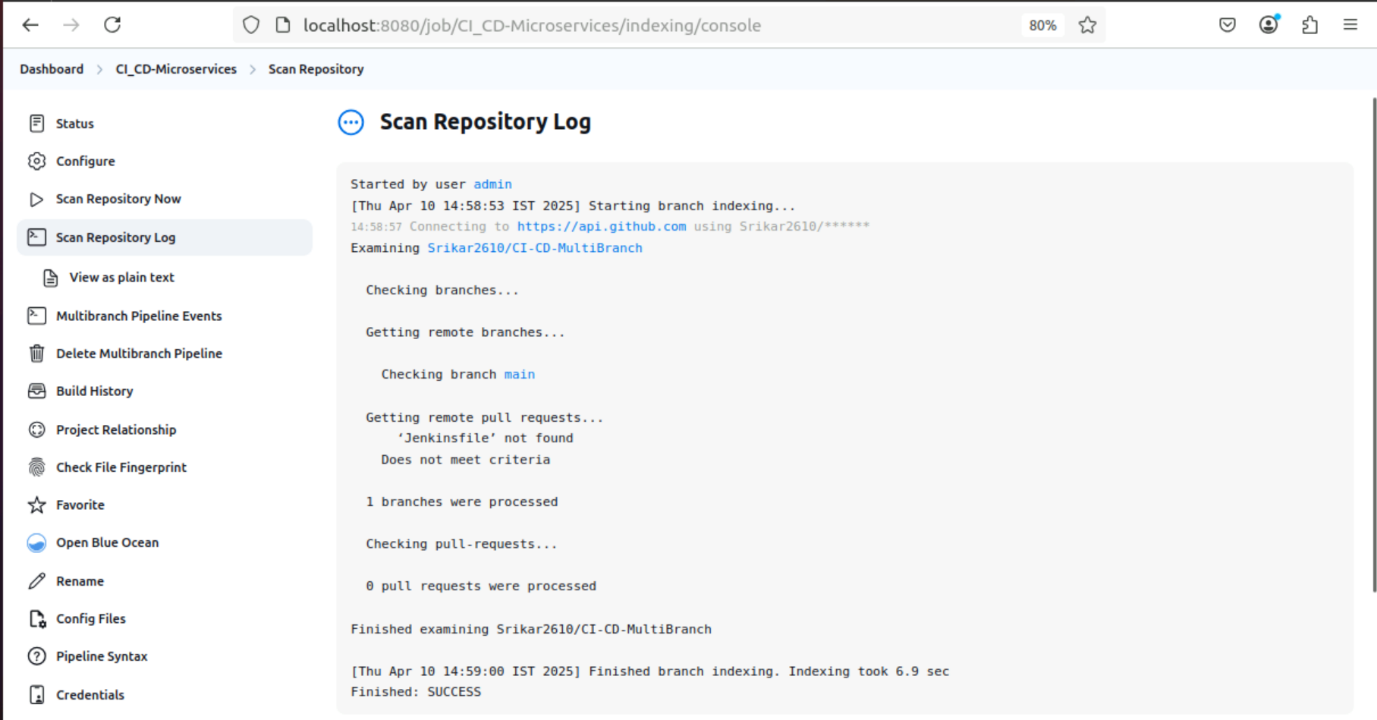
**Step 3: Now we are supposed to create a webhook for this project**

****

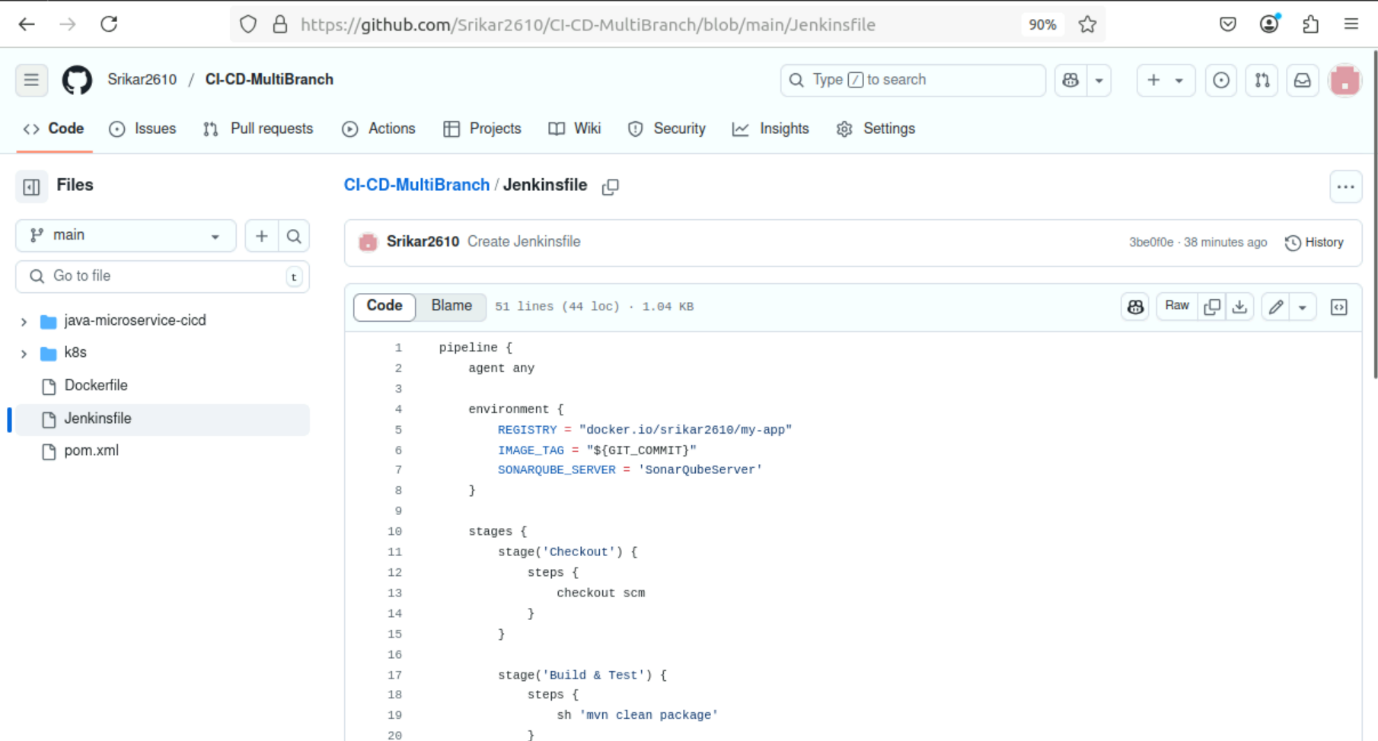
**We can get the Payload URL from the ngrok file to authenticate**

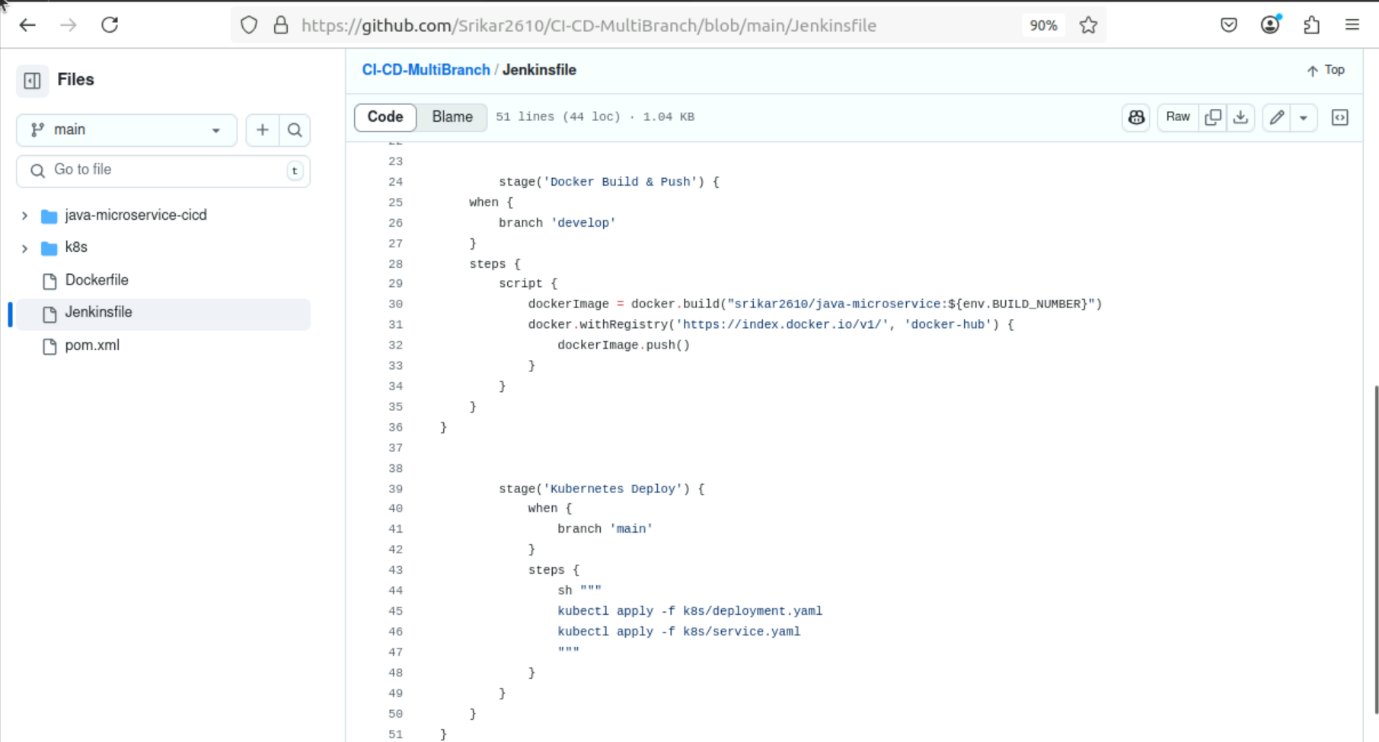
****

**Step 4: Check for Scan repository log to connect the webhook with Jenkins and it is connected successfully**

****

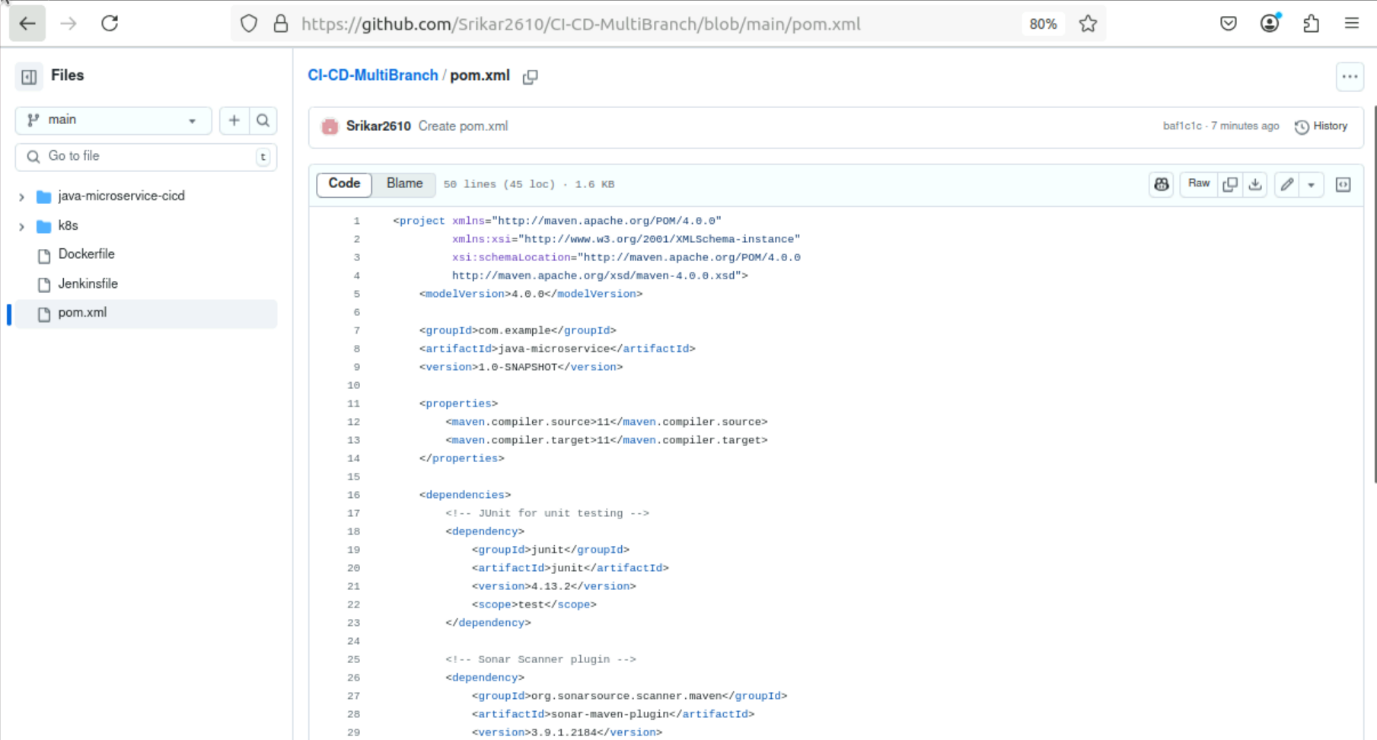
**Jenkinsfile**

****

****

**III. Building Java Code with Maven**

**pom.xml file for maven Build and integration**

****

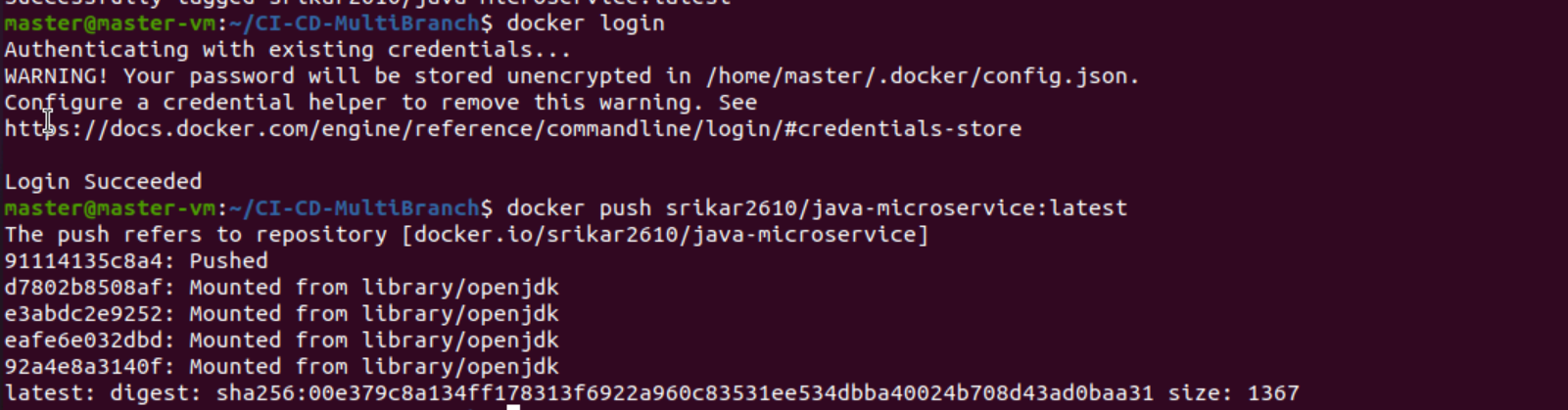
**V. Docker Containerization and Image Push**

**Dockerfile**

****

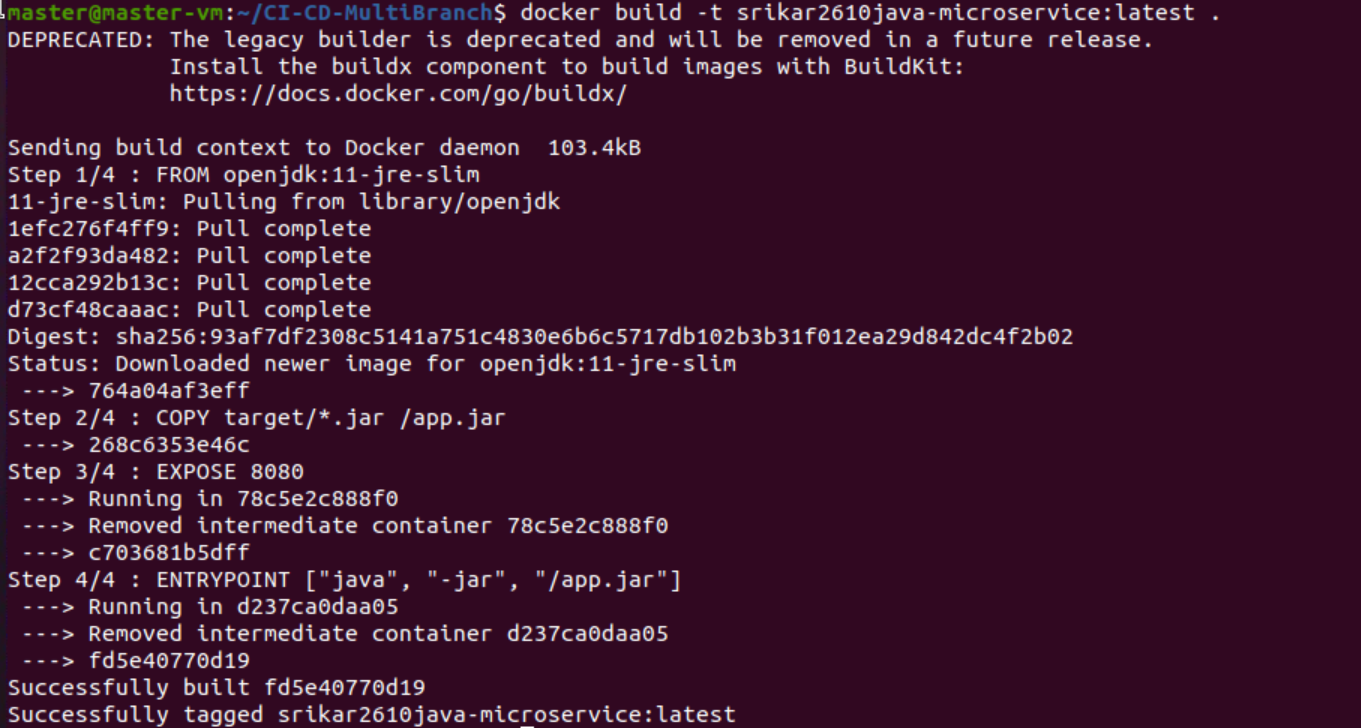
**After creating Dockerfile we need to build and push the image into our local dockerhub**

**By using Git we can login into docker using “docker login”**

****

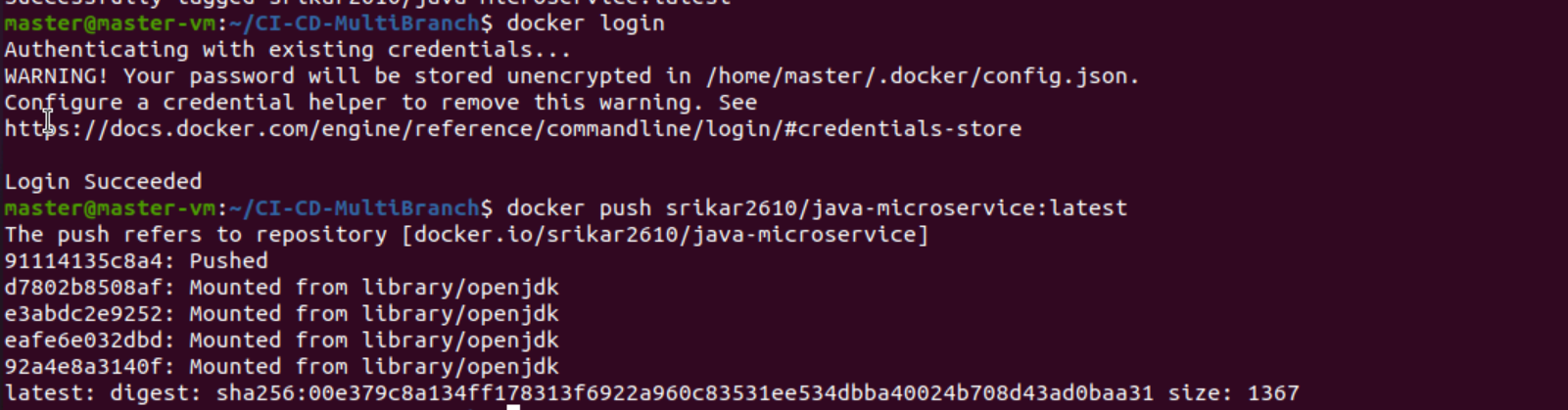
**Step 5: Building the docker image**

**docker build -t srikar2610/java-microservice: latest .**

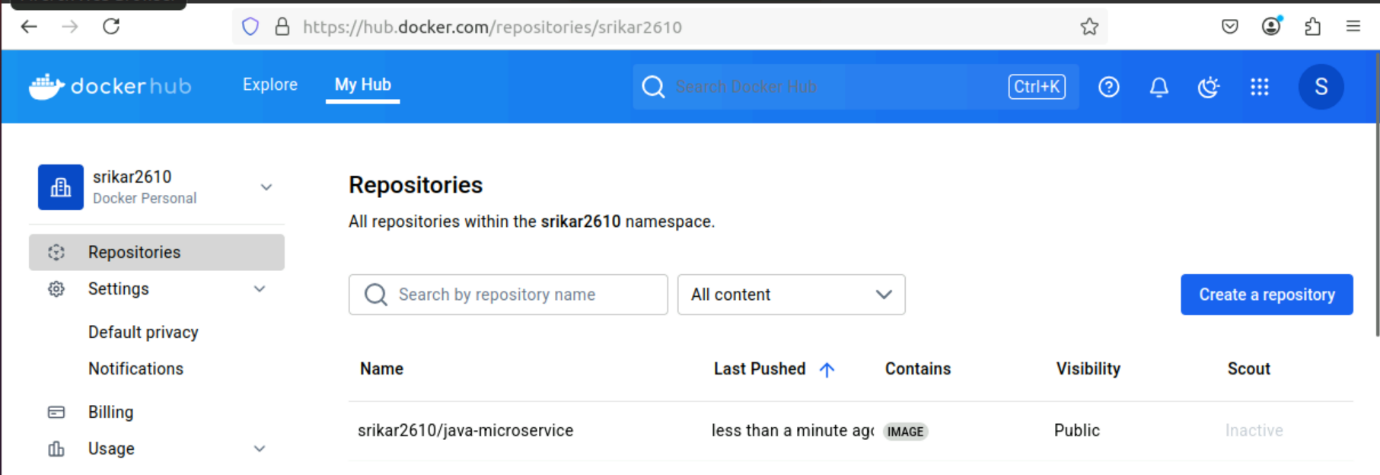
****

**Step 6: Pushing the image into dockerhub**

**docker push srikar2610/java-microservice: latest**

****

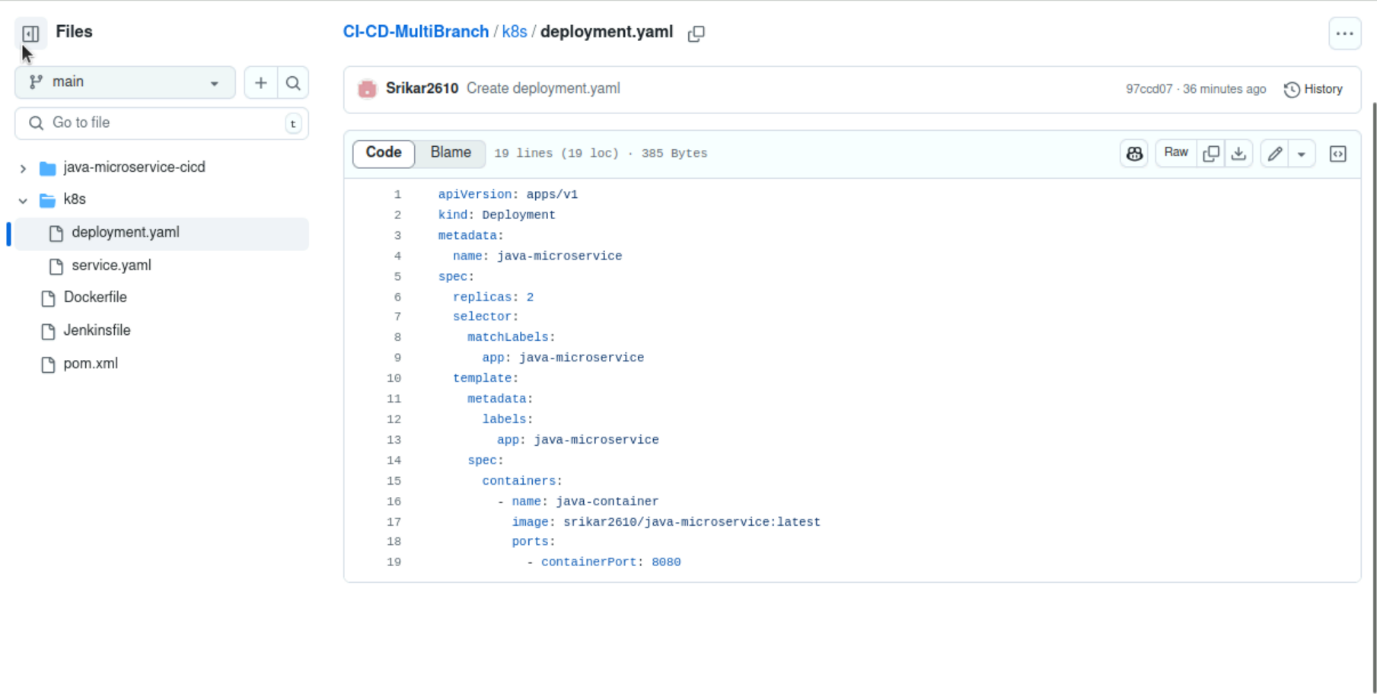
**Here we can see in below screenshot “srikar2610/java-microservice” image has been pushed into my docker hub account**

****

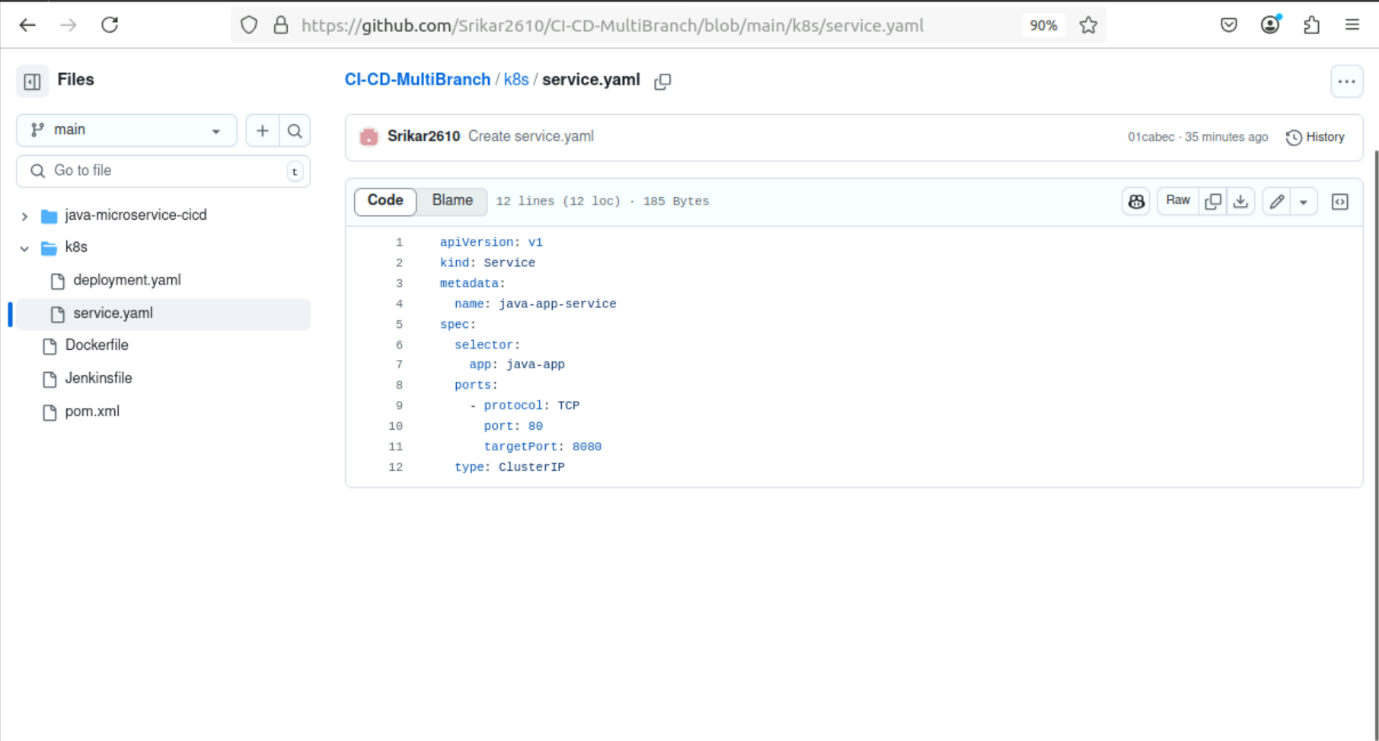
**VI. Kubernetes Deployment**

**Step 7: Created deployment and service files for kubernetes**

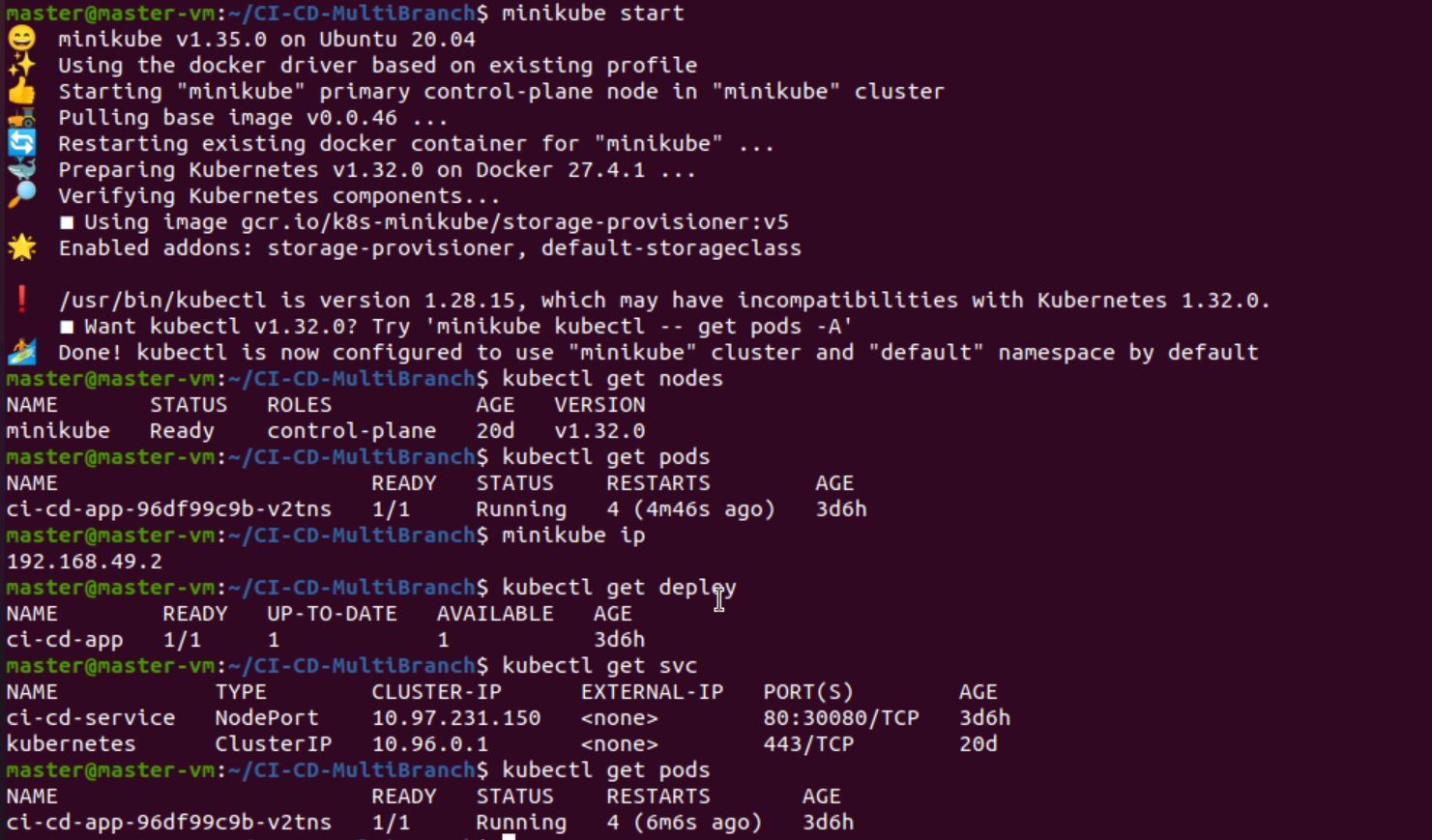
**k8s/deployment.yaml**

****

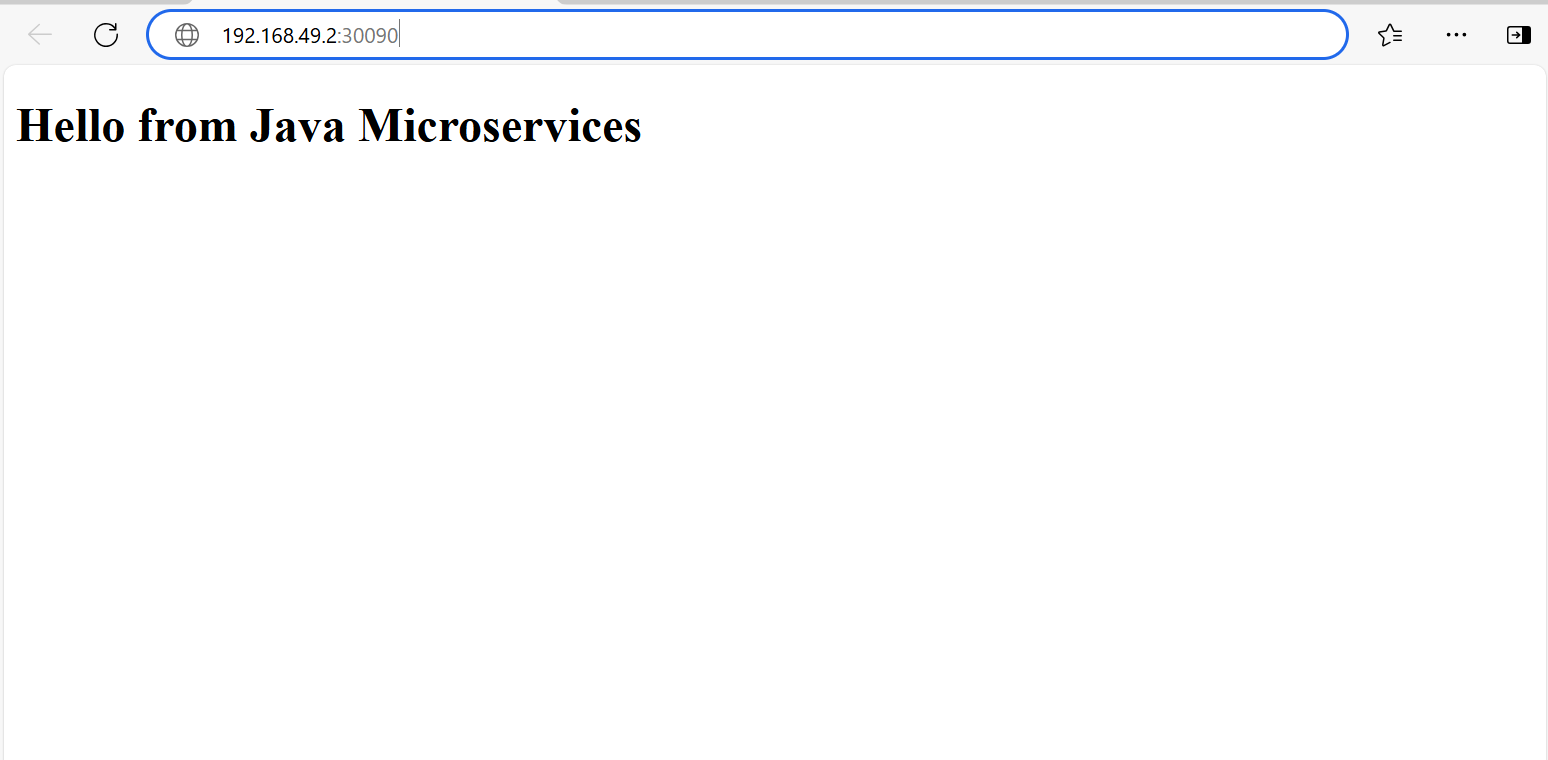
**k8s/service.yaml**

****

**Step 8: Start Minikube and deploying**

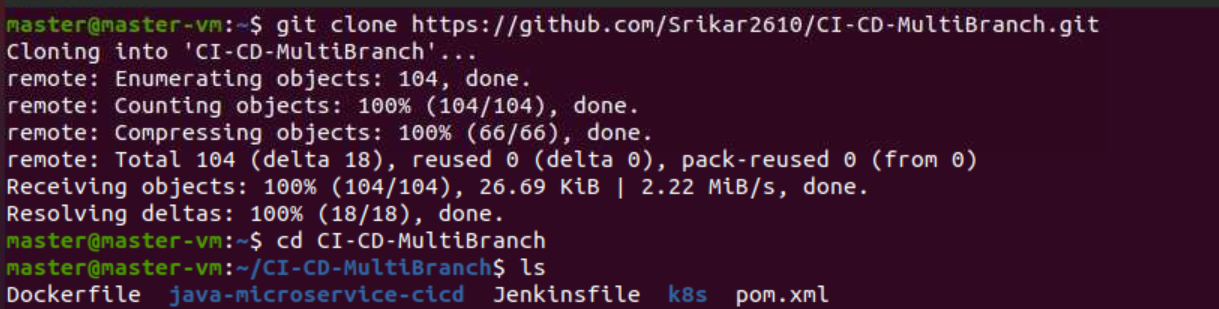
****

**OUTPUT**

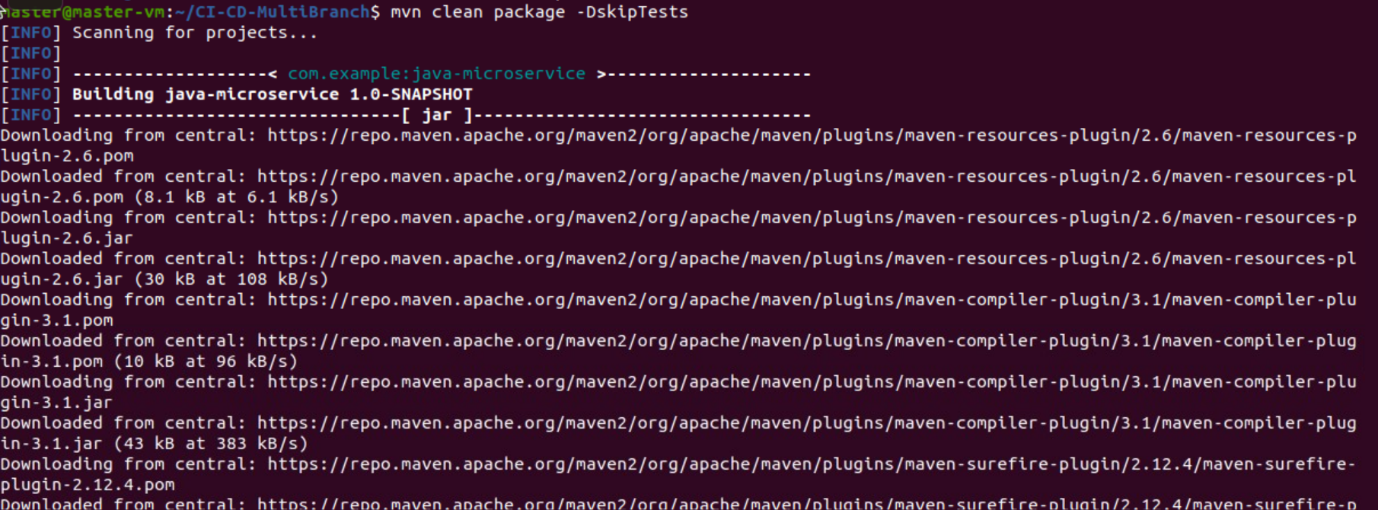
****

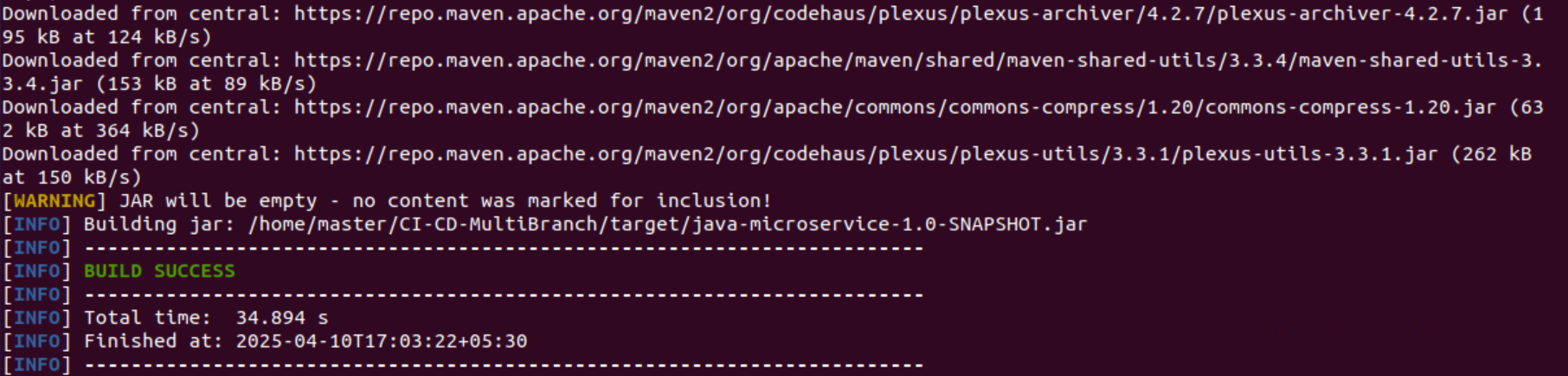
**VII. Jenkins Integration**

**Step 9: Cloning the repository**

****

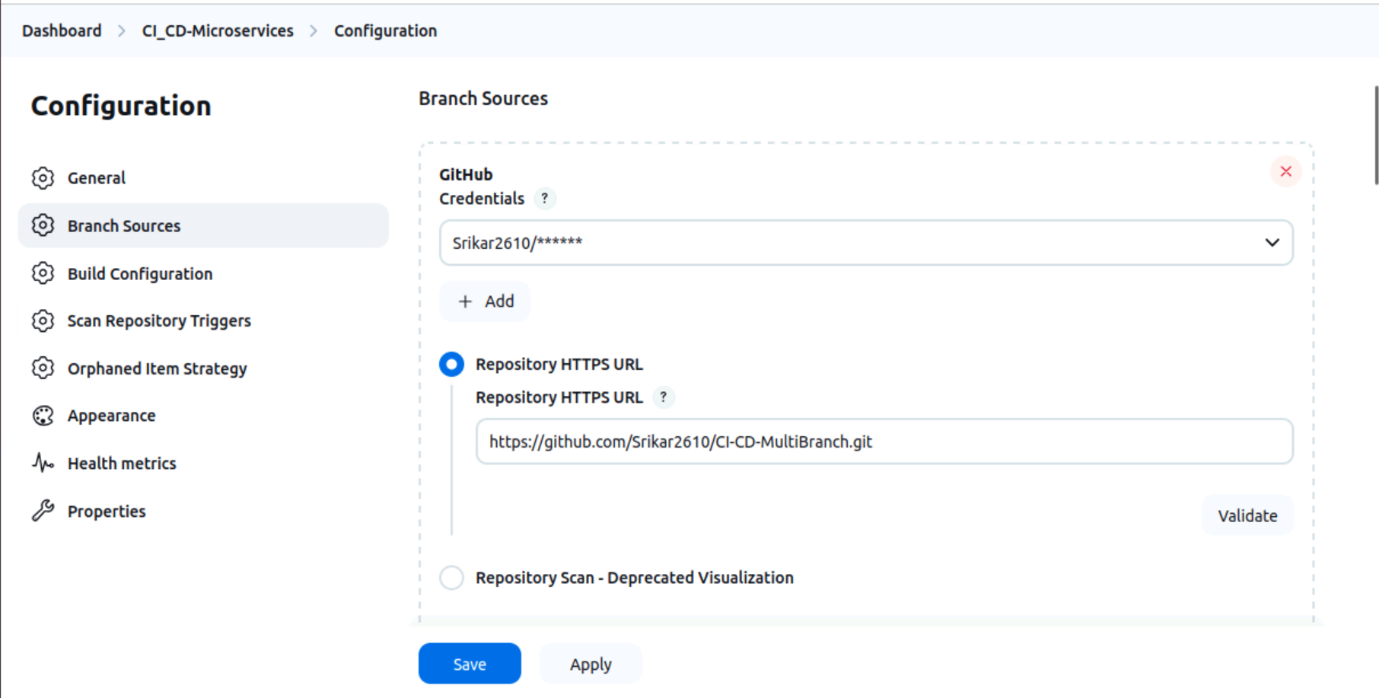
**Step 10: Building the mvn clean packages**

****

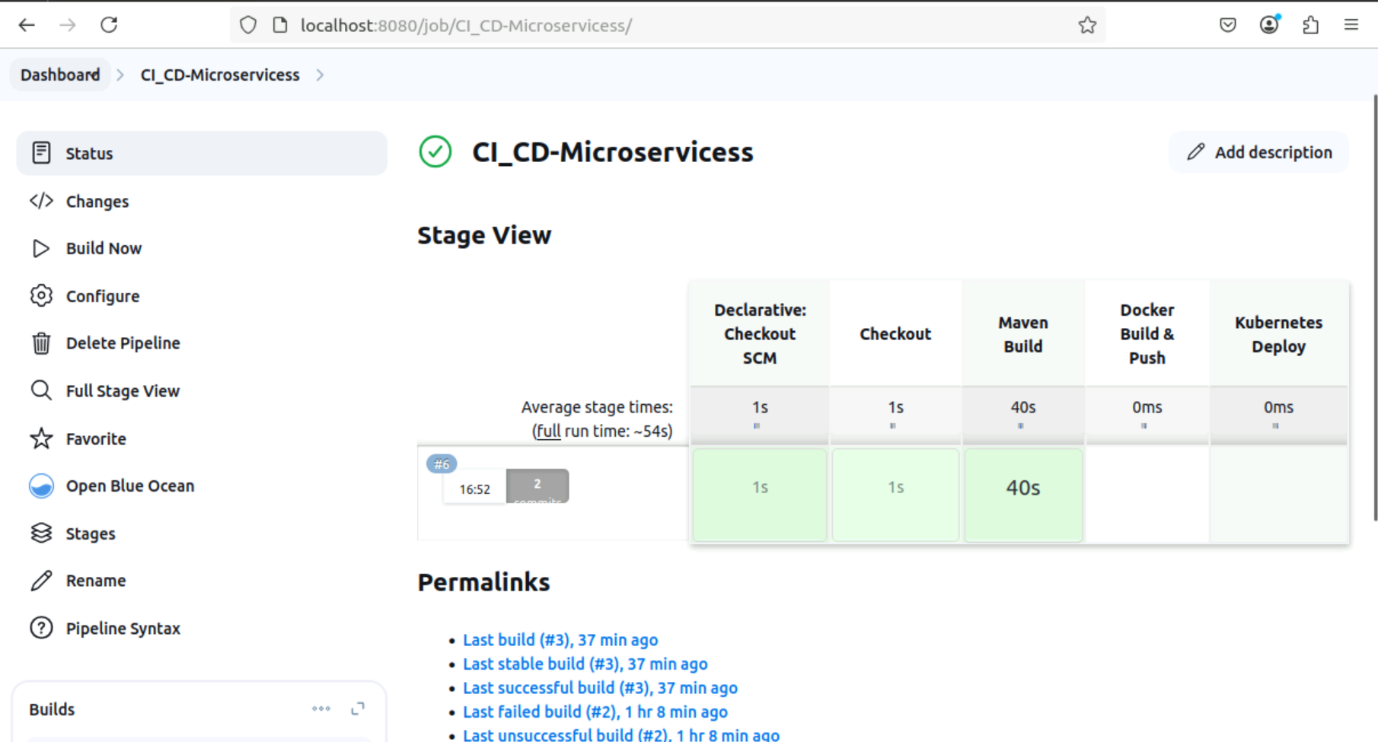
****

**The MVN package is build successfully**

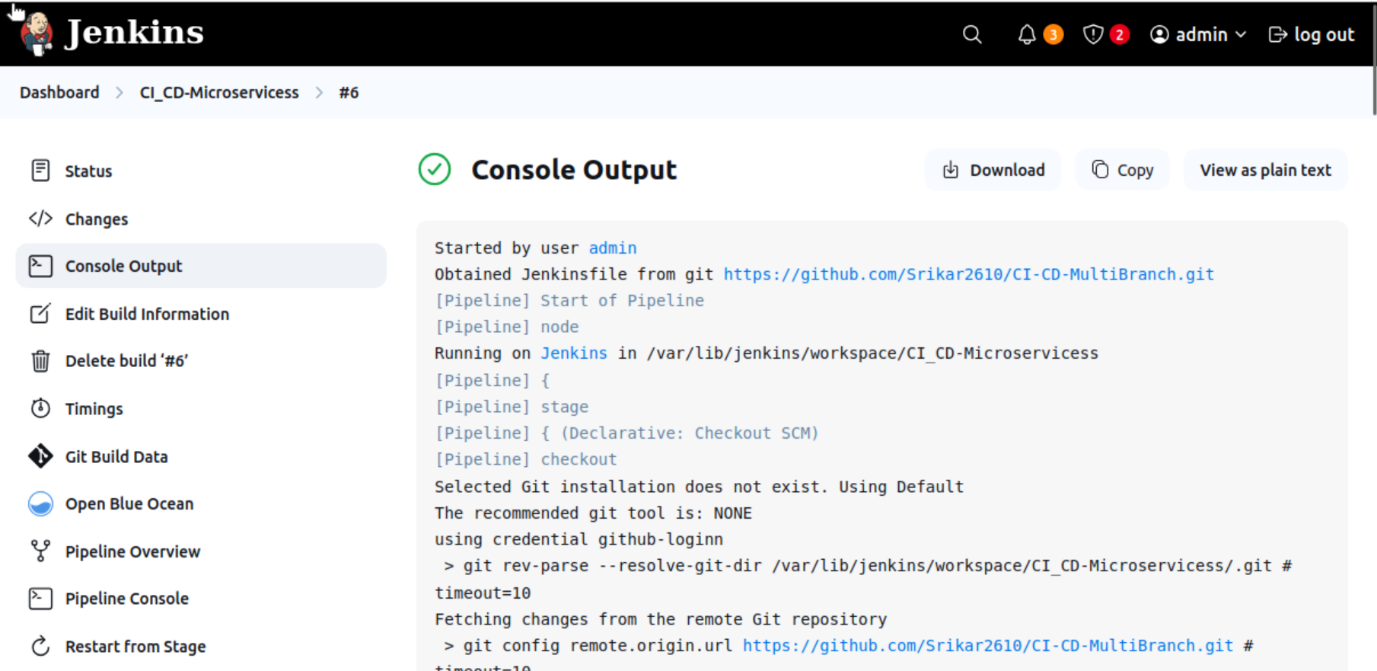
**Step 11: Creating the Multibranch pipeline**

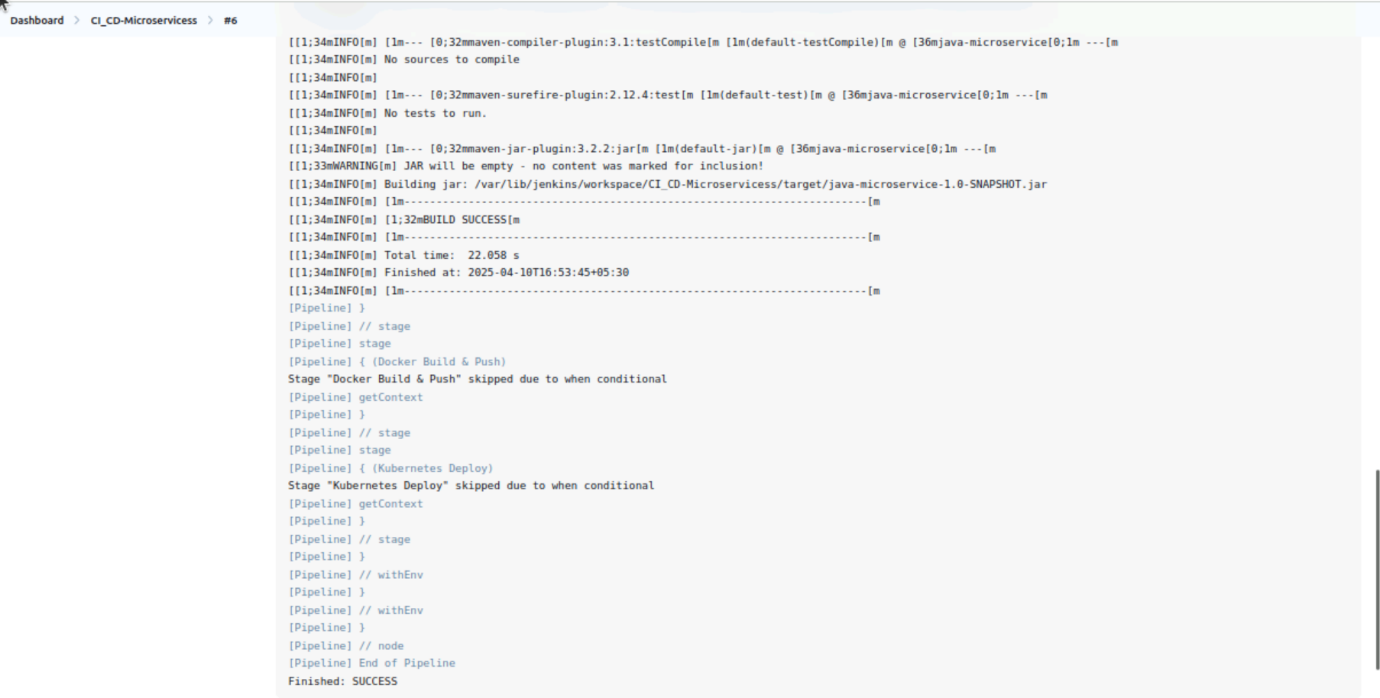
****

**The stages of the pipeline are built successfully**

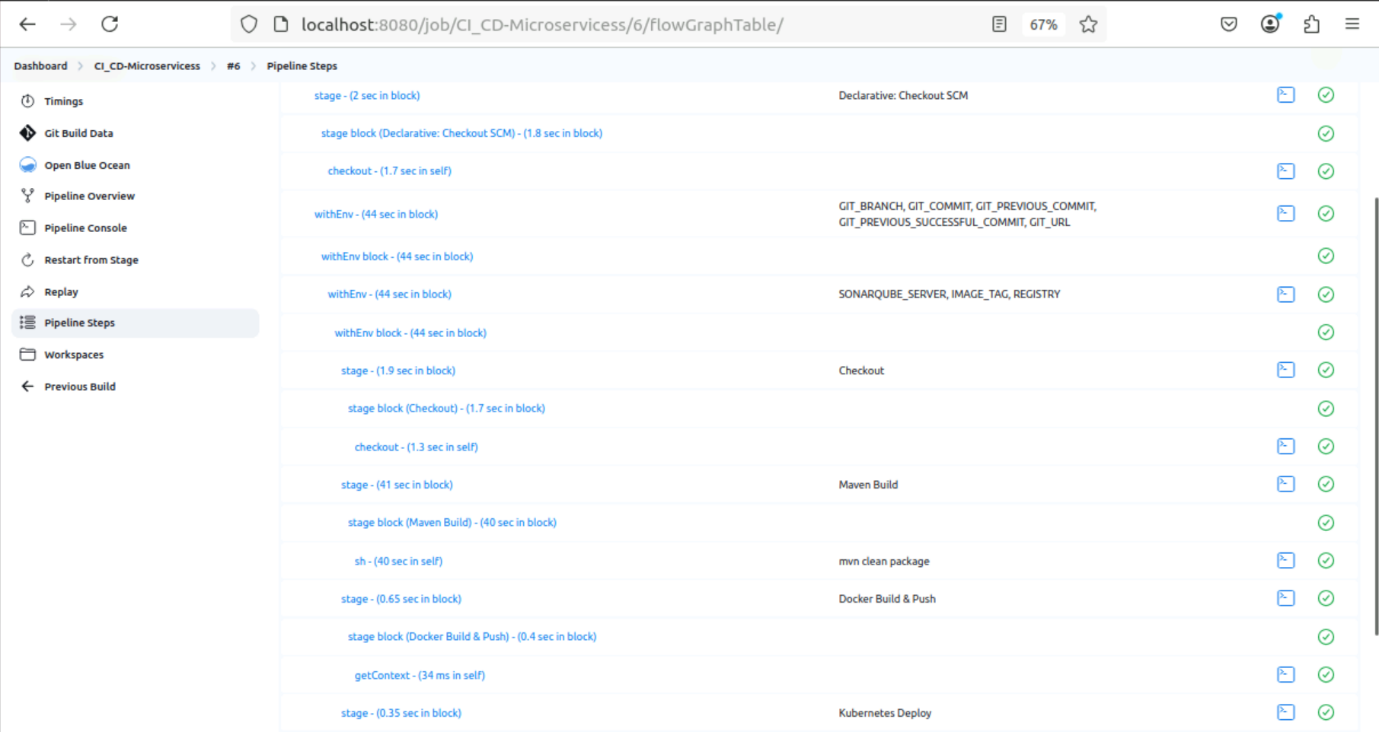
****

**OUTPUT**

****

****

**PIPELINE STAGES**

****