**Building using Maven**

Definition:Maven is like a project manager for Java applications.Just like a manager

Organizes tasks,resources,and deadlines,Maven organizes dependencies,builds,tests,and deployments, ensuring everything runs smoothly and efficiently.

Step1:Install Java and Maven on jenkins

Step2:Fork the eKart Repository on GitHub

# Step 3: Configure Jenkins CreateaNewJobinJenkins

1. Open **Jenkins** in your browser.
2. Clickon **NewItem**→Select **FreestyleProject**→Name it Maven→Click **OK**. **Configure the Job**

# SetupBuild Tools:

* + - Under **GlobalToolConfiguration**,add **Java** and **Maven** if not configured.

# SetGitHub Repository:

* + - Goto**SourceCodeManagement** →Select**Git**.
    - Paste the forked repositoryURL.
    - Set the branch to main.

# AddBuildCommand:

* + - Goto**Build**→AddBuildStep→Select**Invoketop-levelMaven targets**.
    - Enter:cleanpackage-DskipTests
    - Then**BuildNow**.

Step4:Navigate to Jenkins Workspace cd /var/lib/jenkins/workspace

ls#Listavailableprojects cd Maven

cd target

ls#Verifygeneratedartifacts(e.g.,.jarfile)

Step5:Check Docker Image and Kubernetes Deployment docker build -t test -f docker/Dockerfile

Docker push shubhashree1311/shubha@2004

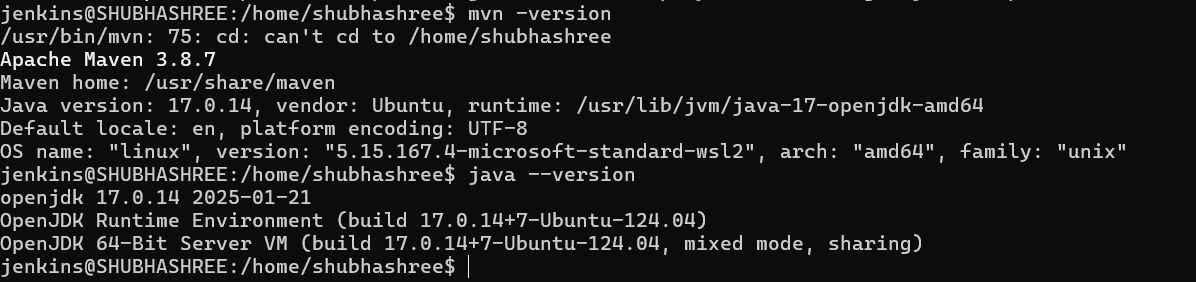
Kubectl create deployment maven --image=test–port80

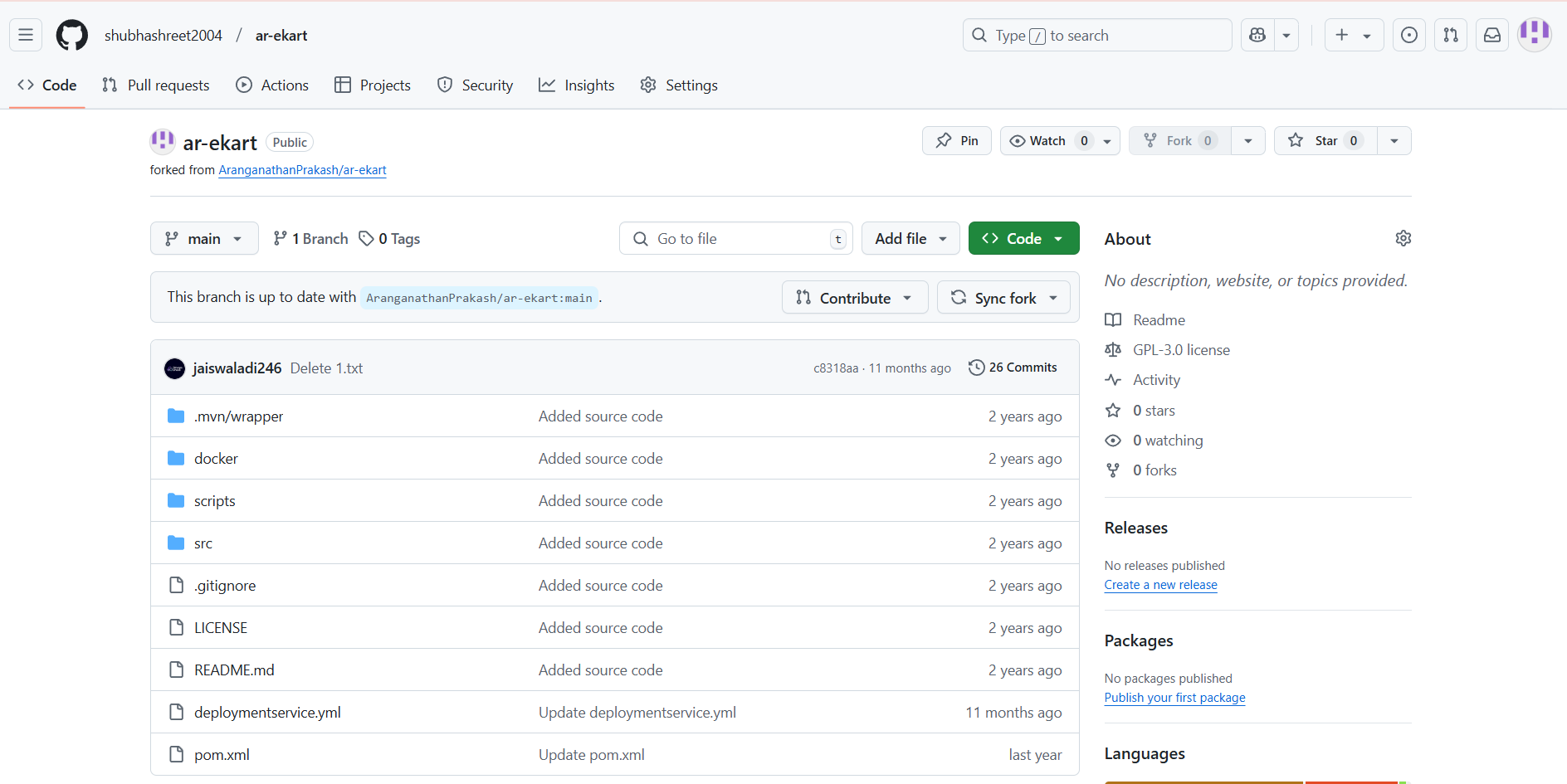
Kubectl expose deployment maven --type=NodePort --port=80 --target-port=8070 docker images | grep shubhashree1311/shubha@2004 Verify Docker image is built

Kubectl get pods#Checkrunning pods

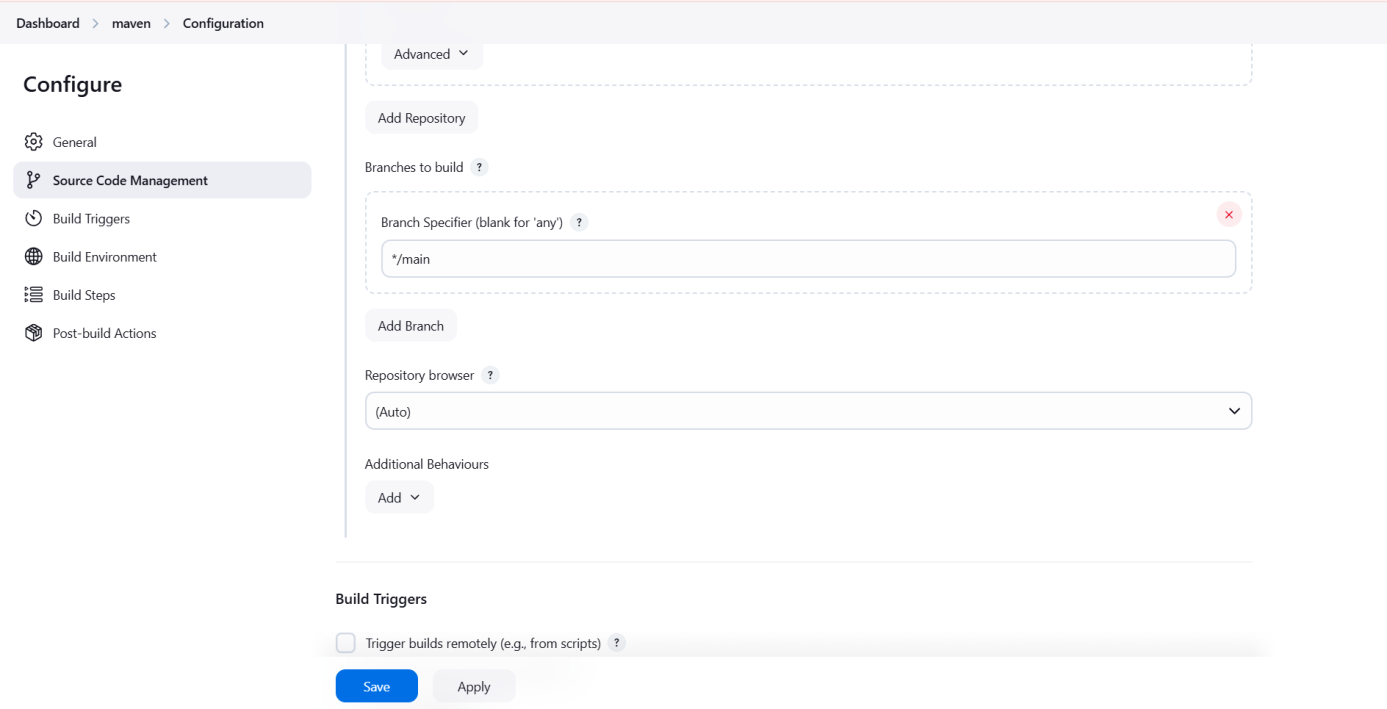
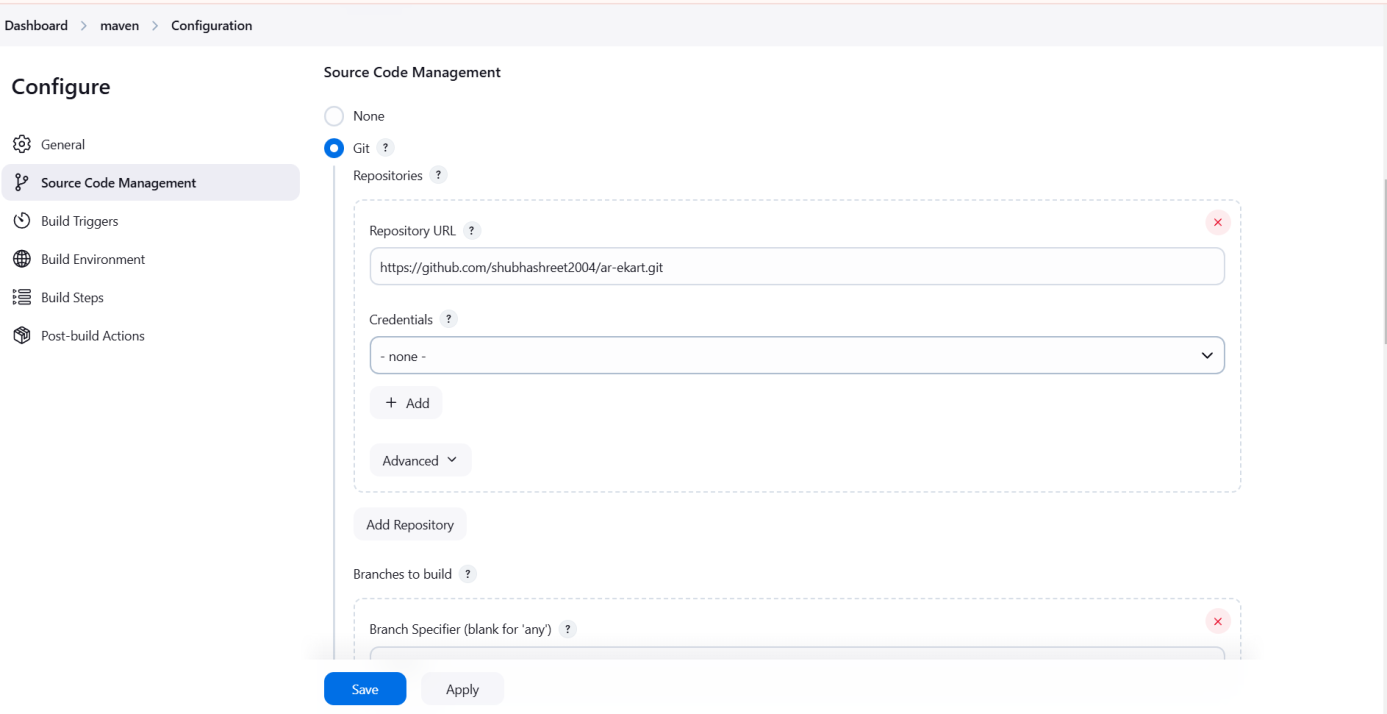
Minikube service maven

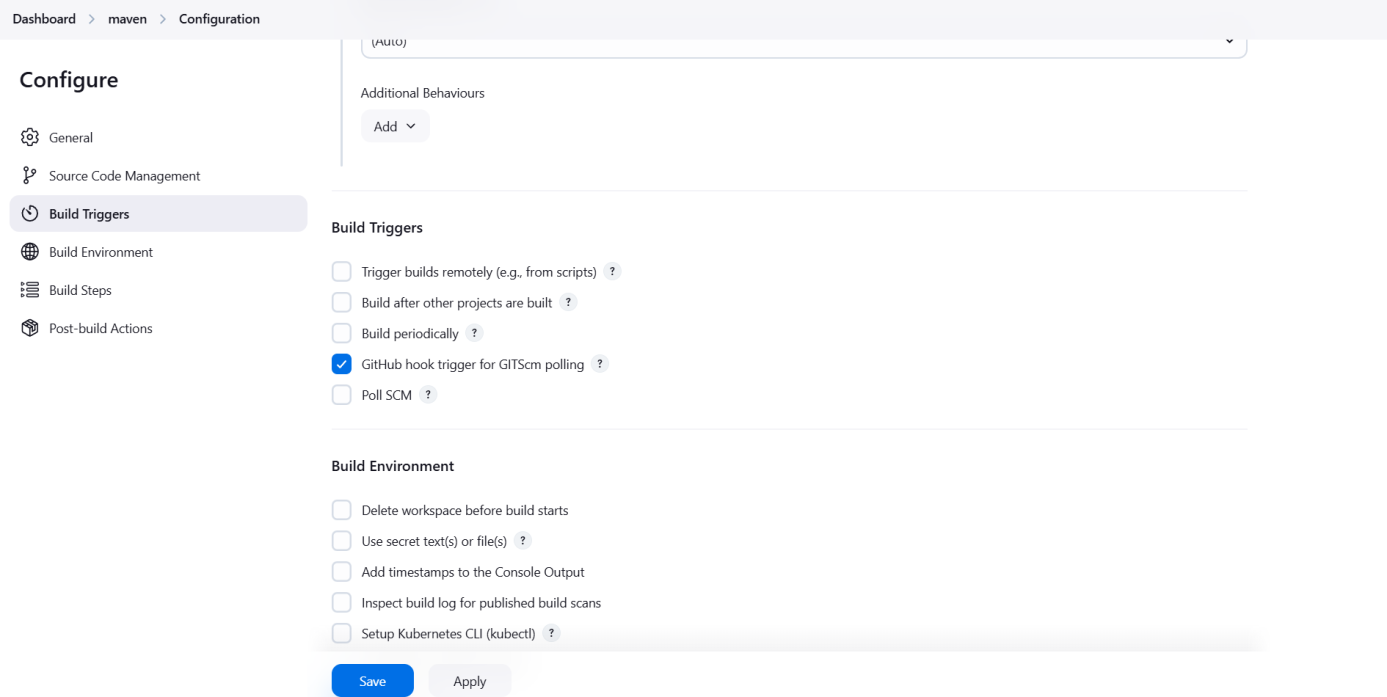
#GettheserviceURL Output and screenshots

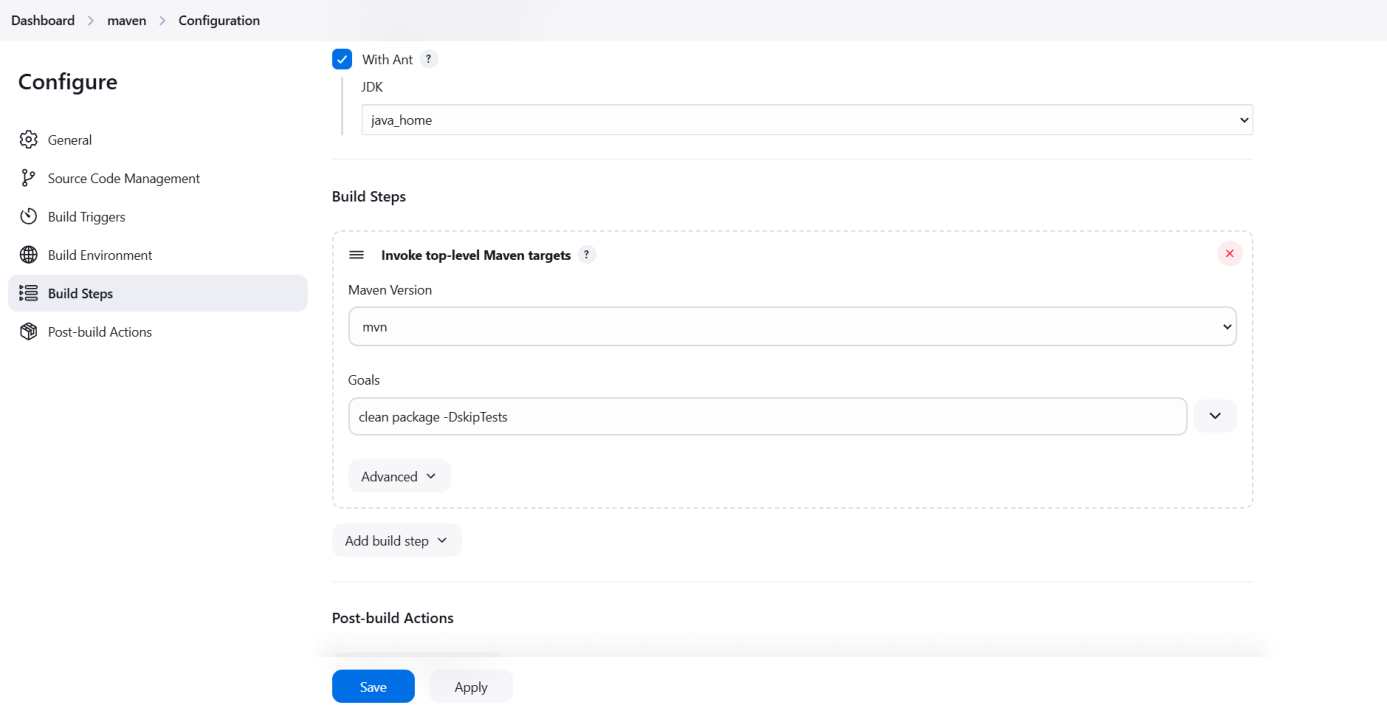


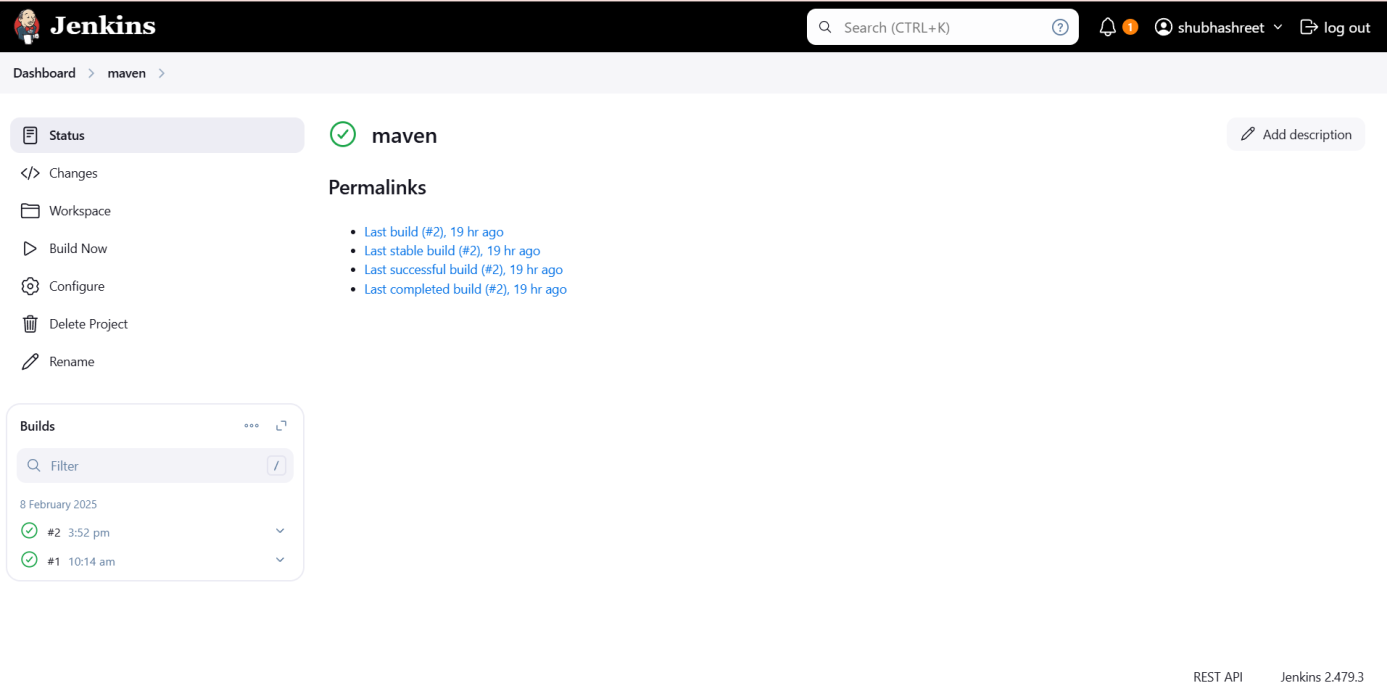


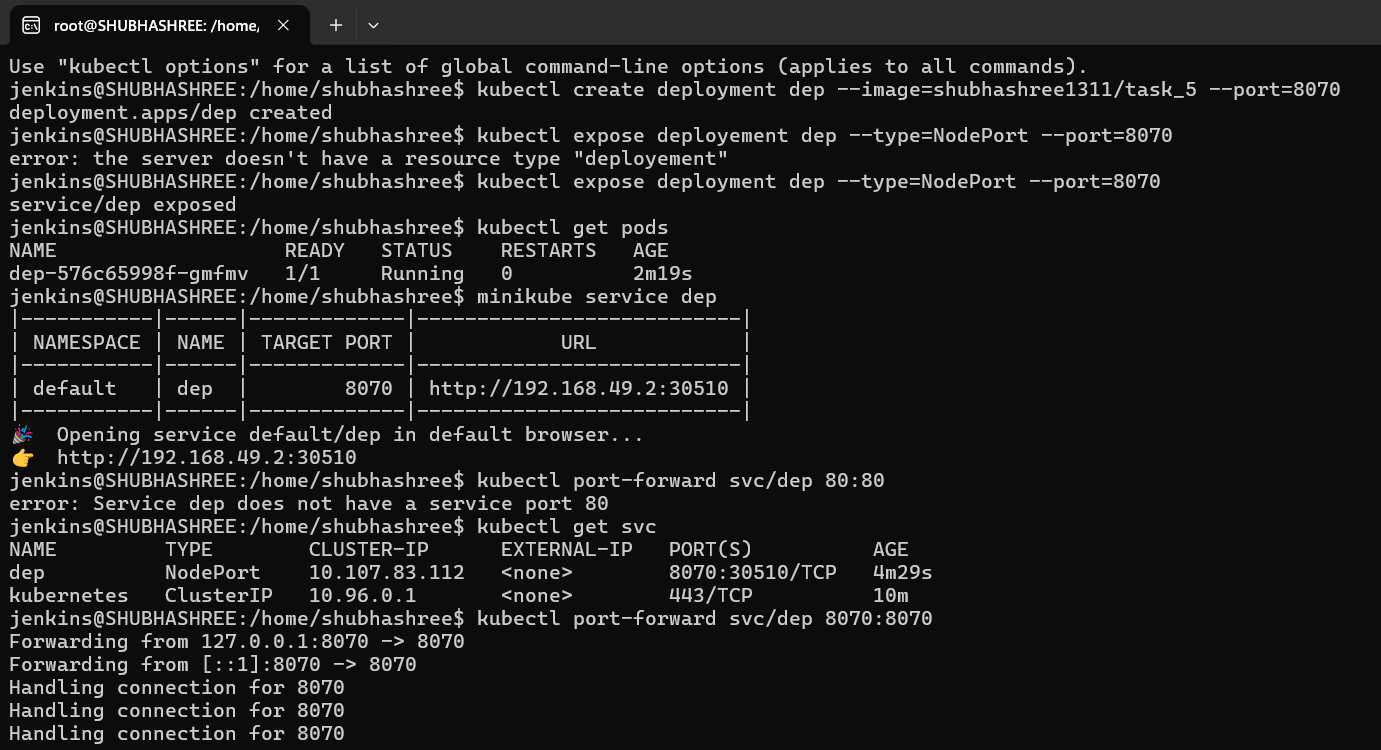
JenkinsConfigration











Output:

