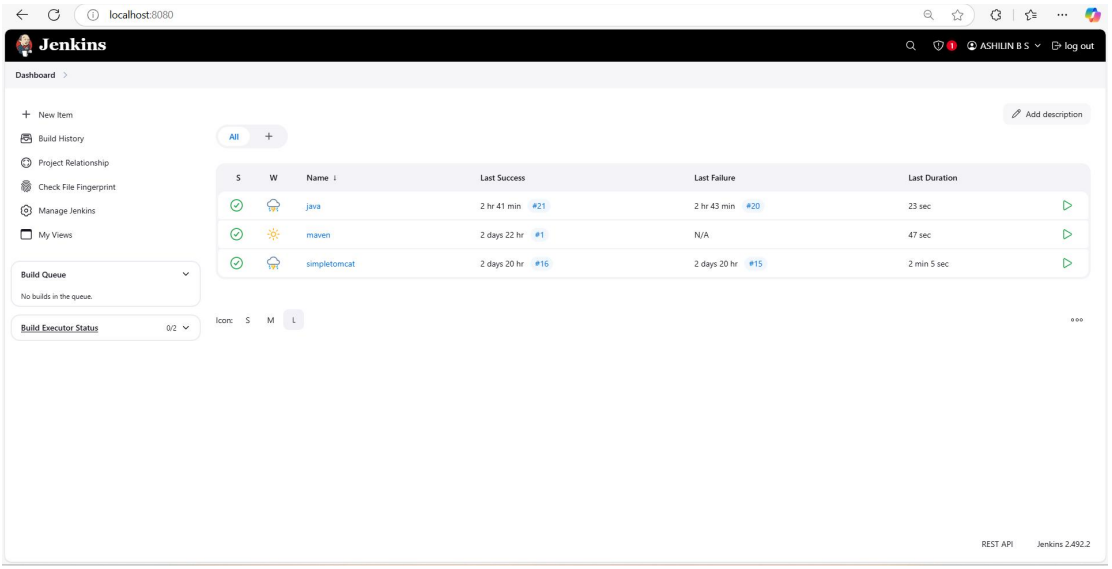
DAY 6:

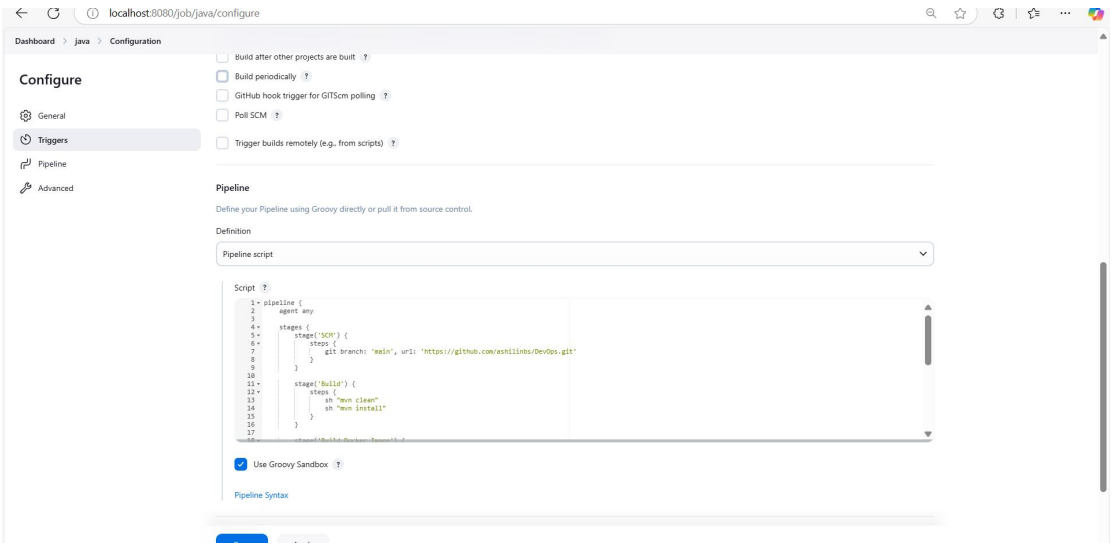
JENKINS:

Jenkins is an open-source automation server used for **Continuous Integration (CI) and Continuous Deployment (CD)**. It helps automate software development processes, including **building, testing, and deploying applications**.



JENKINS PIPELINE:

A **Jenkins Pipeline** is a set of automated steps that define how software is built, tested, and deployed. It is written as code in a **Jenkinsfile** and helps implement **Continuous Integration (CI) and Continuous Deployment (CD).**



KUBERNETES DEPLOYMENT:

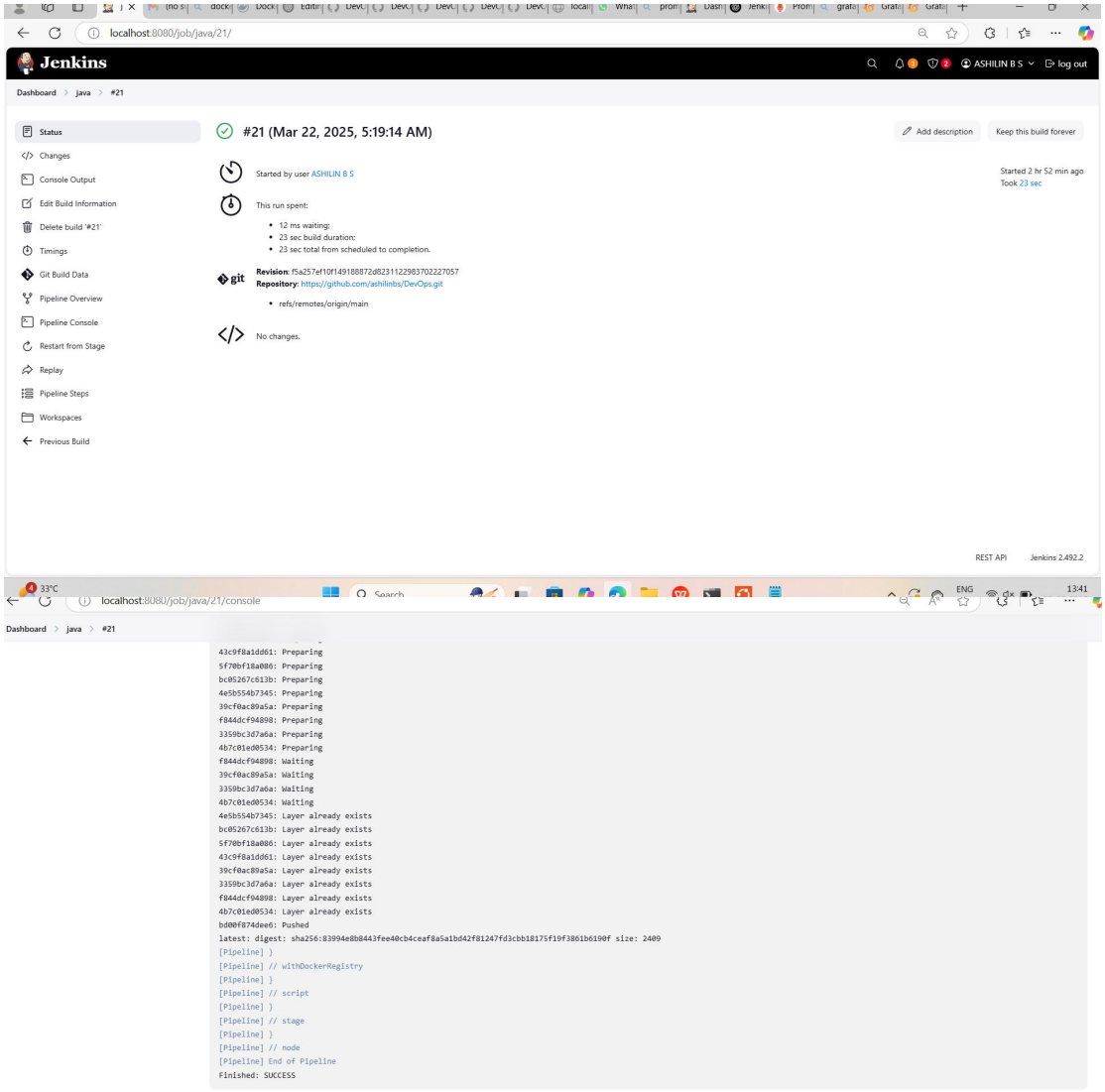
apiVersion: apps/v1
kind: Deployment

```
metadata:
  name: my-deploy
  labels:
    apptype: web-backend  # Ensure this matches Service selector
spec:
  replicas: 4
  selector:
    matchLabels:
      apptype: web-backend  # Ensure this matches Service selector
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend  # Ensure this matches Service selector
    spec:
      containers:
      - name: my-app
        image: ashilin20/app:latest
        ports:
        - containerPort: 5050
---
apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    apptype: web-backend  # Make labels consistent
spec:
  type: NodePort
  ports:
    - targetPort: 8080
      port: 5050
      nodePort: 30001
```

PIPELINE SCRIPT:

```
pipeline {
    agent any

    stages {
        stage('SCM') {
            steps {
                git branch: 'main', url: 'https://github.com/ashilinbs/DevOps.git'
            }
        }

        stage('Build') {
            steps {
```

```
            sh "mvn clean"
            sh "mvn install"
        }
    }

    stage('Build Docker Image') {
        steps {
            script {
                sh 'docker build -t ashilin20/app .'
            }
        }
    }

    stage('Push to Docker Hub') {
        steps {
            script {
                withDockerRegistry(credentialsId: 'cur', url:
'https://index.docker.io/v1/') {
                    sh 'docker push ashilin20/app'
                }
            }
        }
    }


  }
}
```



MINIKUBE COMMANDS:

Start & Stop Minikube

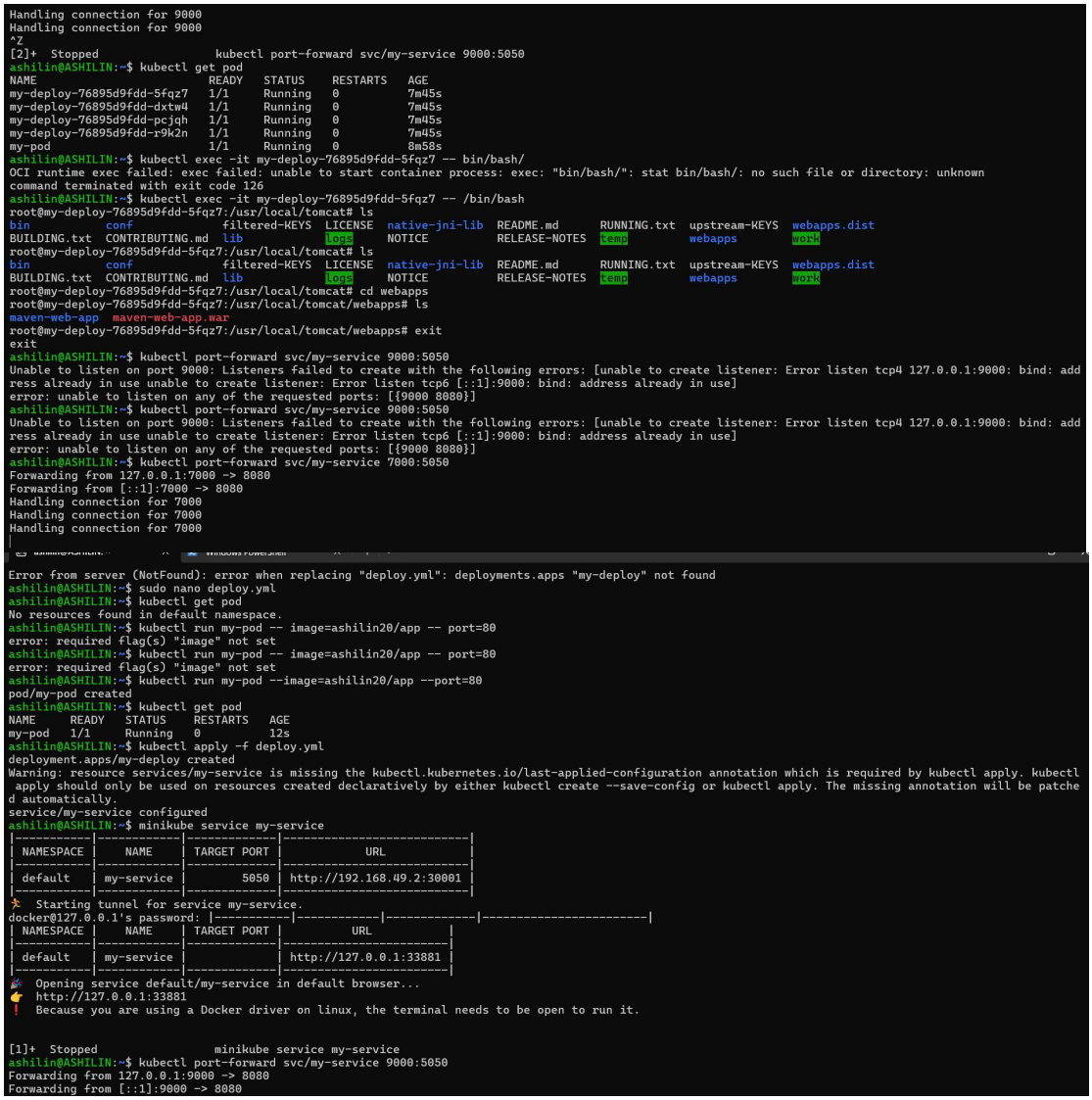
sh
CopyEdit
minikube start # Start Minikube cluster
minikube start --driver=docker # Start with Docker driver
minikube status # Check cluster status
minikube stop # Stop Minikube
minikube delete # Delete the cluster

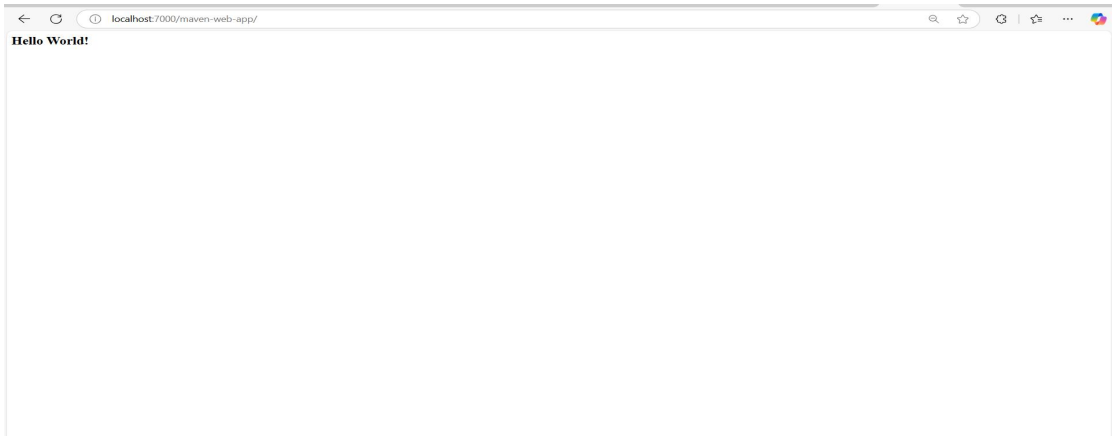
Configure Minikube

sh

CopyEdit
minikube config set memory 4096 # Set memory to 4GB
minikube config set cpus 2

OUTPUT:






```
Handling connection for 9000
^Z
[2]+  Stopped                 kubectl port-forward svc/my-service 9000:5050
ashilin@ASHILIN:~$ kubectl get pod
NAME                          READY   STATUS    RESTARTS   AGE
my-deploy-76895d9fdd-5fqz7    1/1     Running   0          7m45s
my-deploy-76895d9fdd-dxtw4    1/1     Running   0          7m45s
my-deploy-76895d9fdd-pcjqh    1/1     Running   0          7m45s
my-deploy-76895d9fdd-r9k2n    1/1     Running   0          7m45s
my-pod                        1/1     Running   0          8m58s
ashilin@ASHILIN:~$ kubectl exec -it my-deploy-76895d9fdd-5fqz7 -- bin/bash/
OCI runtime exec failed: exec failed: unable to start container process: exec: "bin/bash/": stat bin/bash/: no such file or directory: unknown
command terminated with exit code 126
ashilin@ASHILIN:~$ kubectl exec -it my-deploy-76895d9fdd-5fqz7 -- /bin/bash
root@my-deploy-76895d9fdd-5fqz7:/usr/local/tomcat# ls
bin           conf            filtered-KEYS  LICENSE  native-jni-lib  README.md      RUNNING.txt  upstream-KEYS  webapps.dist
BUILDING.txt  CONTRIBUTING.md  lib            logs     NOTICE          RELEASE-NOTES  temp         webapps        work
root@my-deploy-76895d9fdd-5fqz7:/usr/local/tomcat# ls
bin           conf            filtered-KEYS  LICENSE  native-jni-lib  README.md      RUNNING.txt  upstream-KEYS  webapps.dist
BUILDING.txt  CONTRIBUTING.md  lib            logs     NOTICE          RELEASE-NOTES  temp         webapps        work
root@my-deploy-76895d9fdd-5fqz7:/usr/local/tomcat# cd webapps
root@my-deploy-76895d9fdd-5fqz7:/usr/local/tomcat/webapps# ls
maven-web-app  maven-web-app.war
root@my-deploy-76895d9fdd-5fqz7:/usr/local/tomcat/webapps# exit
exit
ashilin@ASHILIN:~$ kubectl port-forward svc/my-service 9000:5050
Unable to listen on port 9000: Listeners failed to create with the following errors: [unable to create listener: Error listen tcp4 127.0.0.1:9000: bind: add
ress already in use unable to create listener: Error listen tcp6 [::1]:9000: bind: address already in use]
error: unable to listen on any of the requested ports: [{9000 8080}]
ashilin@ASHILIN:~$ kubectl port-forward svc/my-service 9000:5050
Unable to listen on port 9000: Listeners failed to create with the following errors: [unable to create listener: Error listen tcp4 127.0.0.1:9000: bind: add
ress already in use unable to create listener: Error listen tcp6 [::1]:9000: bind: address already in use]
```