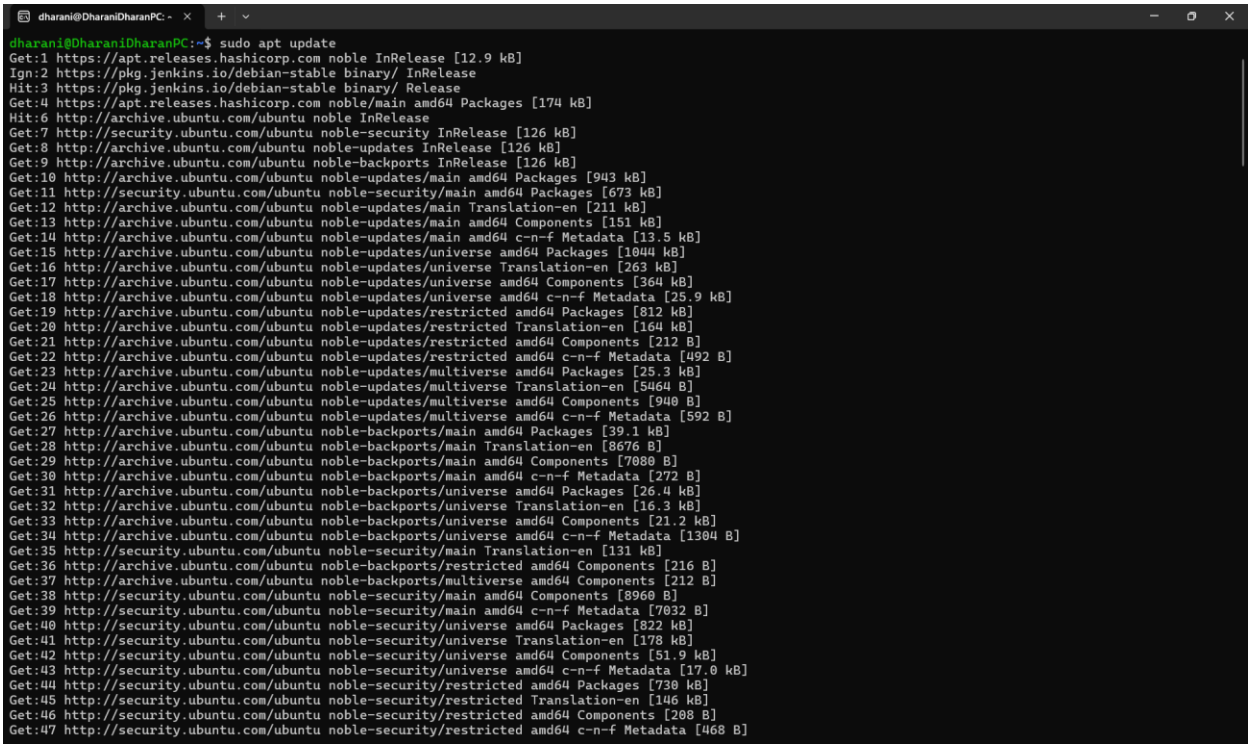
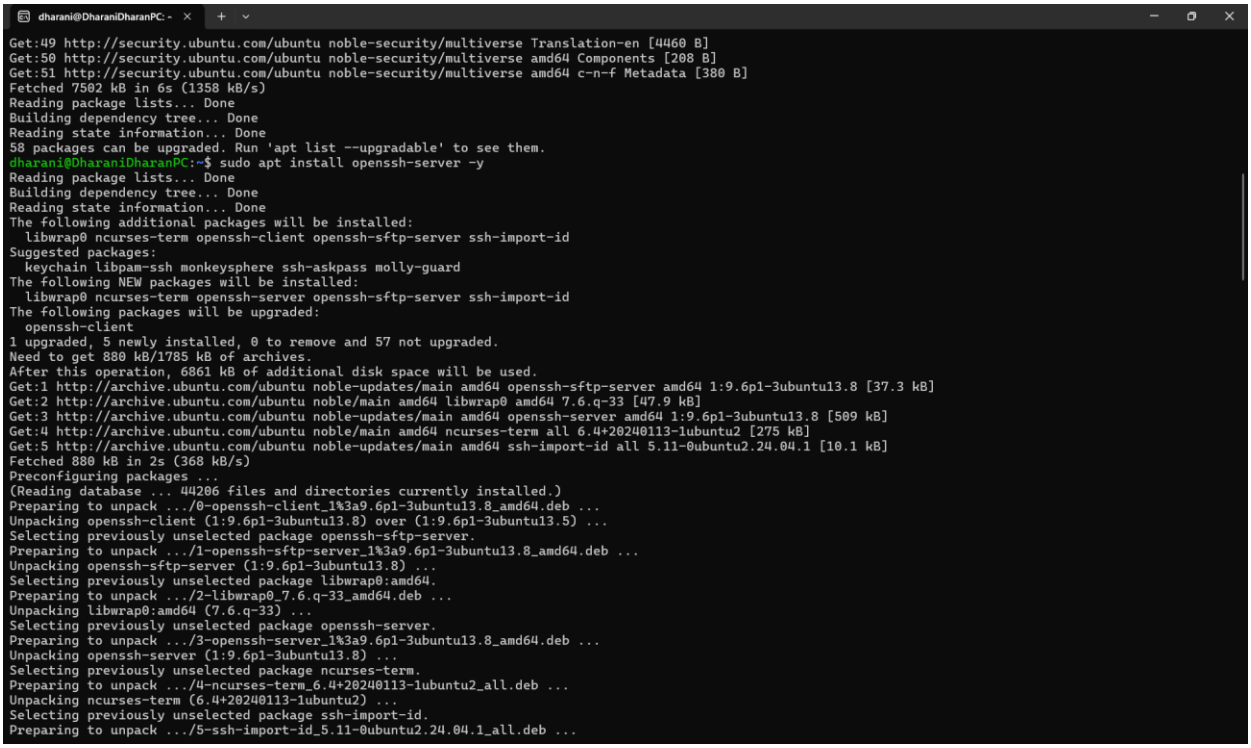
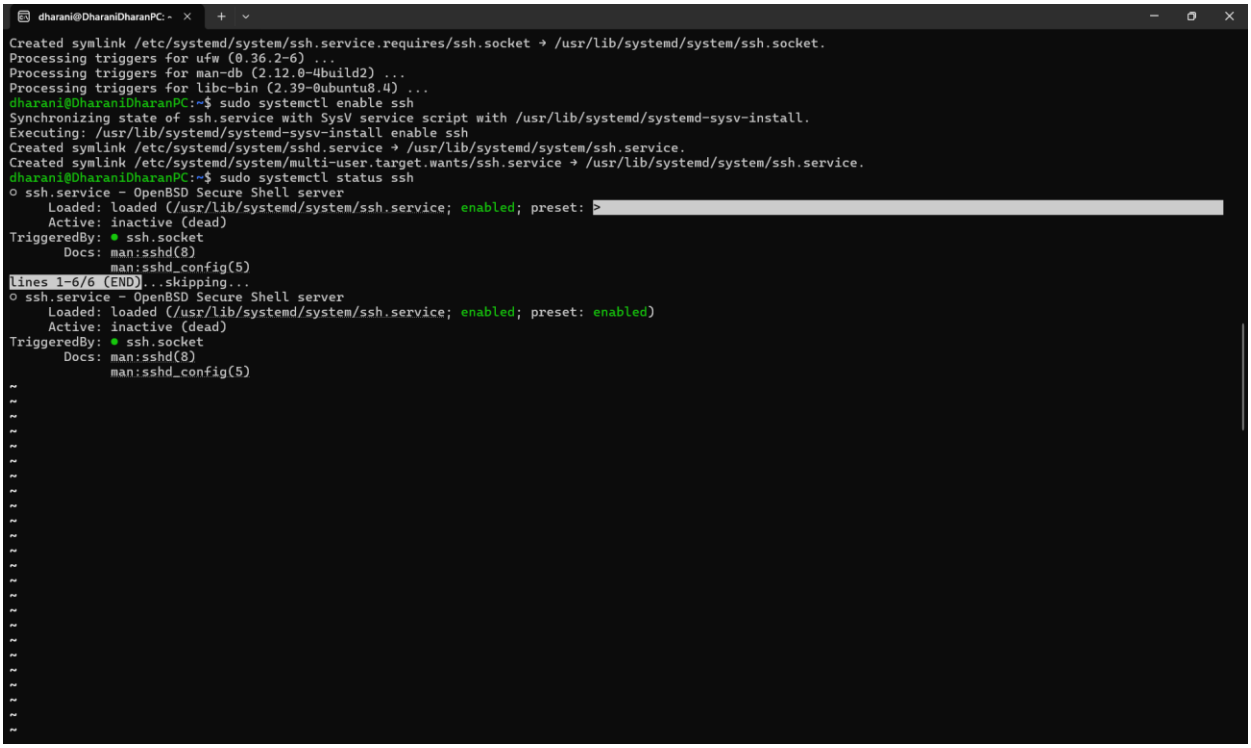
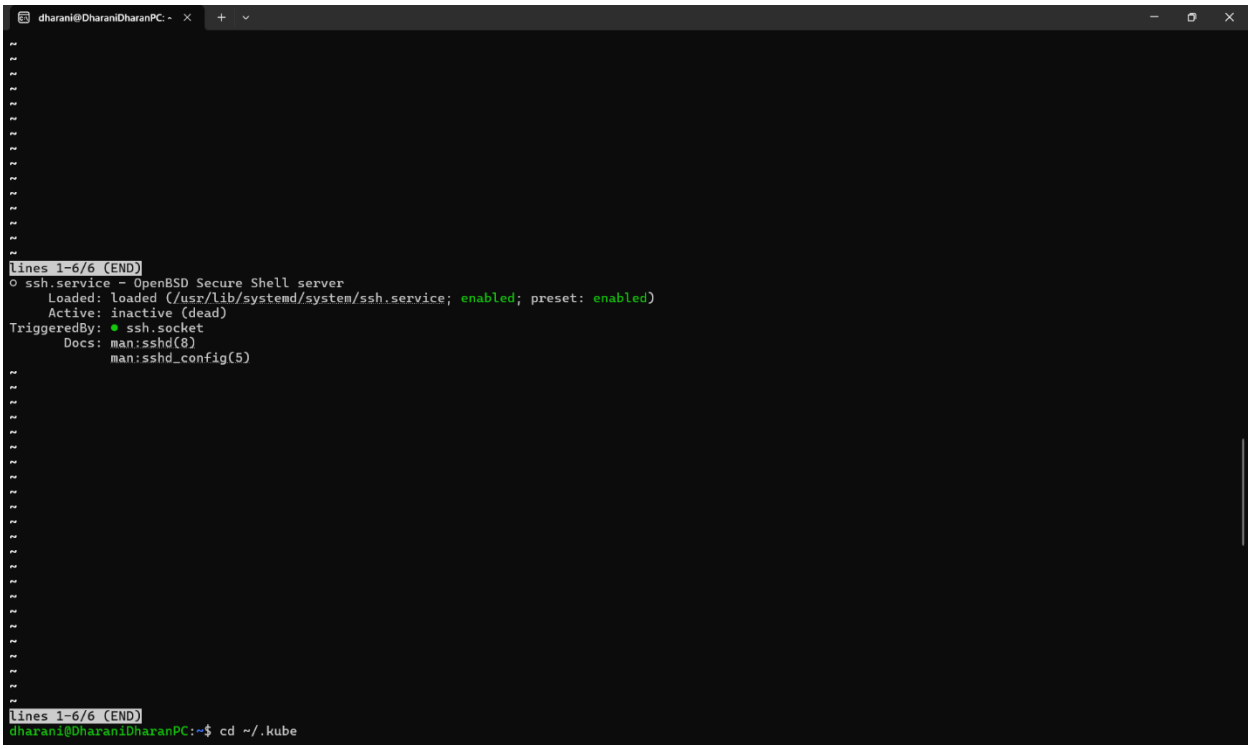
FINAL PROJECT

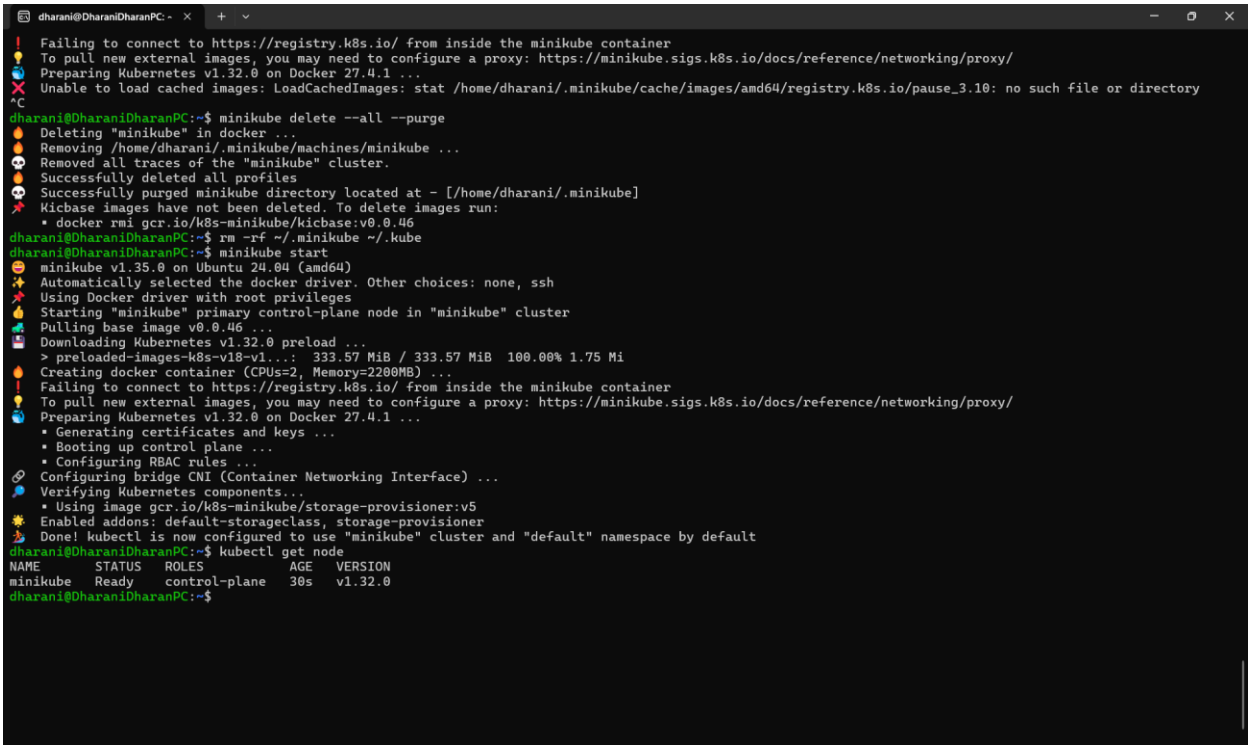
JAVA-APPLICATION DEPLOYMENT IN MINIKUBE



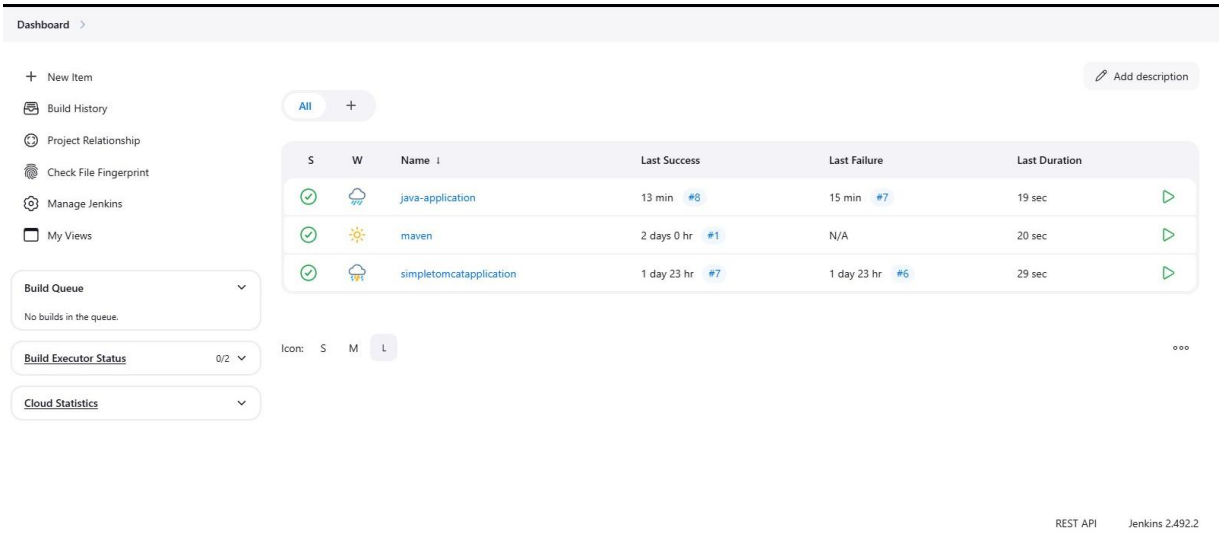



dharani@DharaniDharanPC: ~
Created symlink /etc/systemd/system/ssh.service.requires/ssh.socket → /usr/lib/systemd/system/ssh.socket.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.4) ...
dharani@DharaniDharanPC:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/sshd.service → /usr/lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /usr/lib/systemd/system/ssh.service.
dharani@DharaniDharanPC:~$ sudo systemctl status ssh
○ ssh.service - OpenBSD Secure Shell server
Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset:
Active: inactive (dead)
TriggeredBy: ● ssh.socket
Docs: man:sshd(8)
man:sshd_config(5)
lines 1-6/6 (END)...skipping...
○ ssh.service - OpenBSD Secure Shell server
Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
Active: inactive (dead)
TriggeredBy: ● ssh.socket
Docs: man:sshd(8)
man:sshd_config(5)

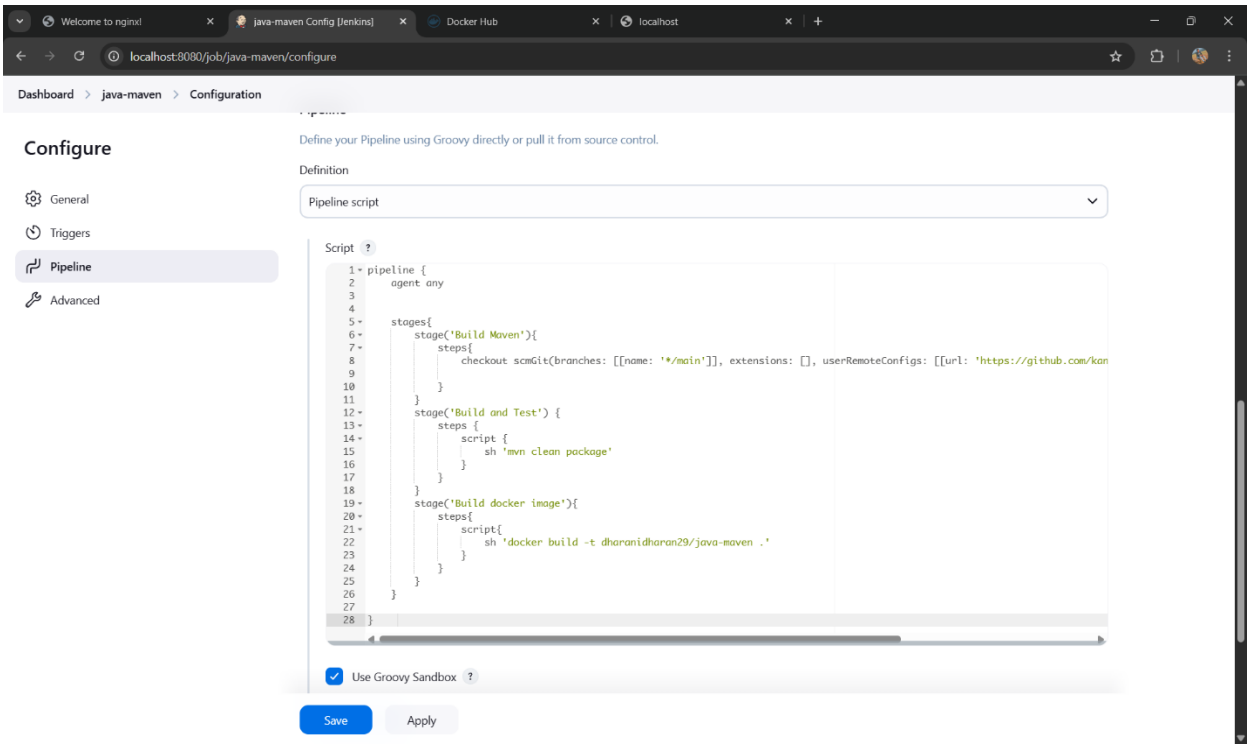

dharani@DharaniDharanPC: ~
lines 1-6/6 (END)
○ ssh.service - OpenBSD Secure Shell server
Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
Active: inactive (dead)
TriggeredBy: ● ssh.socket
Docs: man:sshd(8)
man:sshd_config(5)
lines 1-6/6 (END)
dharani@DharaniDharanPC:~$ cd ~/.kube



CREATION OF A PIPELINE PROJECT NAMED “java-application”



PIPELINE SCRIPT



PIPELINE SCRIPT CODE

```
pipeline {
   agent any

   stages {
      stage('scm') {
         steps {
      git branch: 'main', url: 'https://github.com/_dharanidharan-palanisamy /simple-web-app.git'
         }

      }

      stage('build') {
         steps {
            sh "mvn clean"
            sh "mvn install"
}

}
```

```
stage('build to images') { steps
        {
          script{
            sh 'docker build -t dharanidharan-palanisamy /simplewebapp .'
          }

   }

}

stage('push to hub') {
        steps {
          script{
            withDockerRegistry(credentialsId: 'Docker_cred', url: 'https://index.docker.io/v1/') { sh
             'docker push dharanidharan-palanisamy /simplewebapp'
          }

        }

        }

}

stage('Deploy Web App') {
        steps {
           withKubeConfig(caCertificate: '', clusterName: 'minikube', contextName: 'minikube',
credentialsId: 'config_id', namespace: '', restrictKubeConfigAccess: false, serverUrl:
'https://192.168.39.226:8443')  {
         sh 'kubectl apply -f deployment.yml --validate=false'
        }

        }

}

}

}
```

DEPLOYMENT FILE

deployment.yml

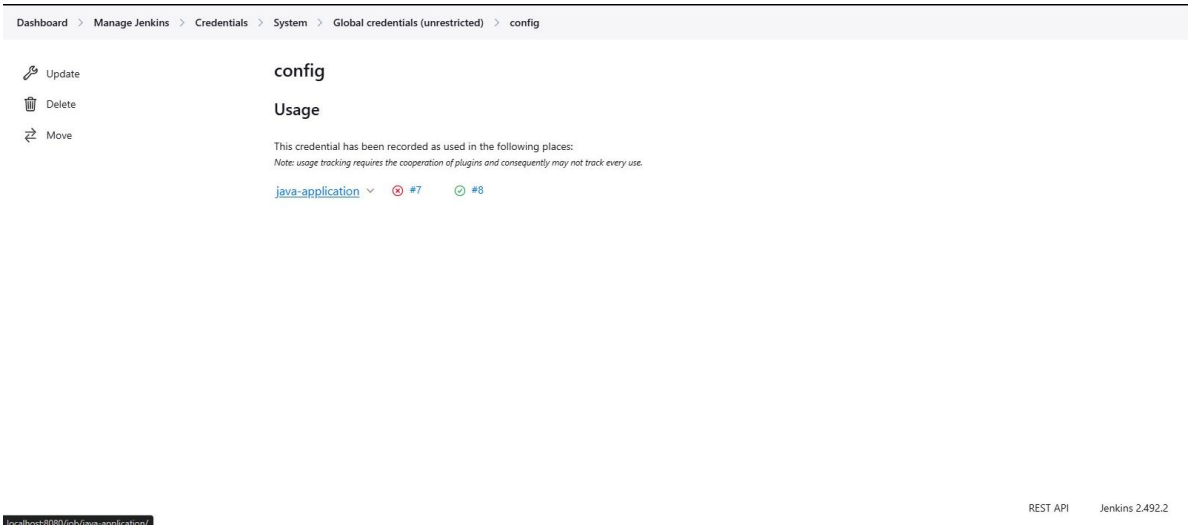
apiVersion: apps/v1 kind:

Deployment metadata:

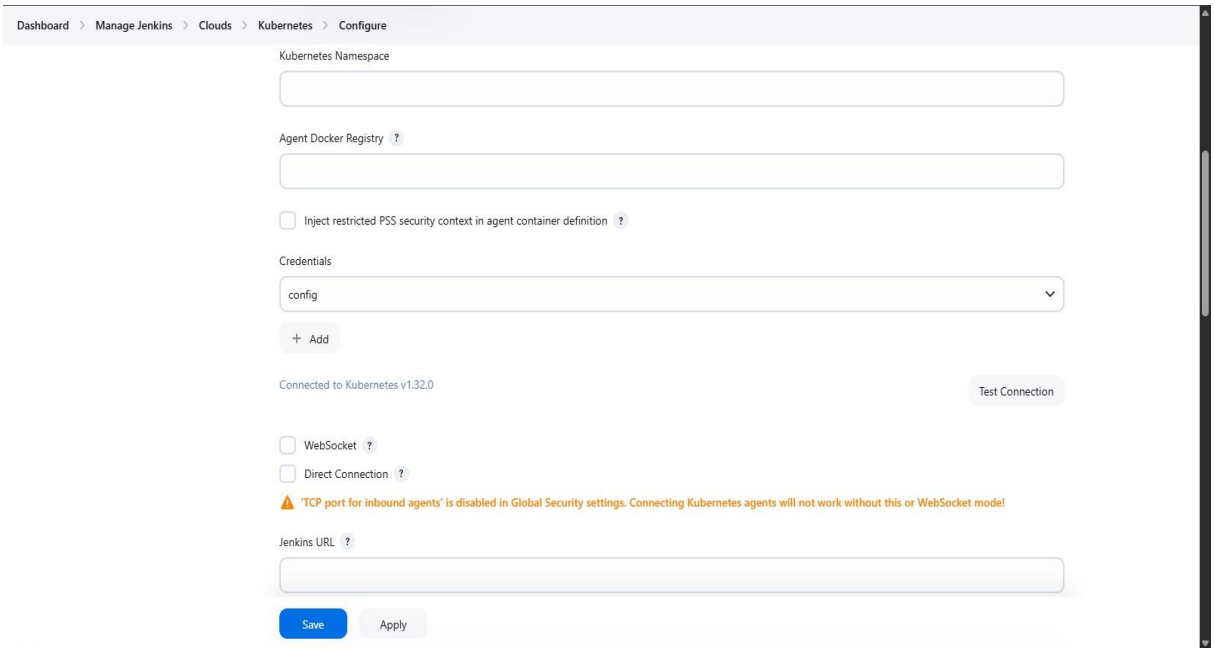
```
  name: webnginx2
  labels:
    name: webnginx2
spec:
  replicas: 1
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
      - name: webnginx2
        image: dharanidharan-palanisamy
        /simplewebapp:latest ports:
        - containerPort: 7070
apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-service
    type: backend-app
```

```
spec:
  type: NodePort
  ports:
   - targetPort: 8080
     port: 7070
     nodePort: 30002
  selector:
    apptype: web-backend
```

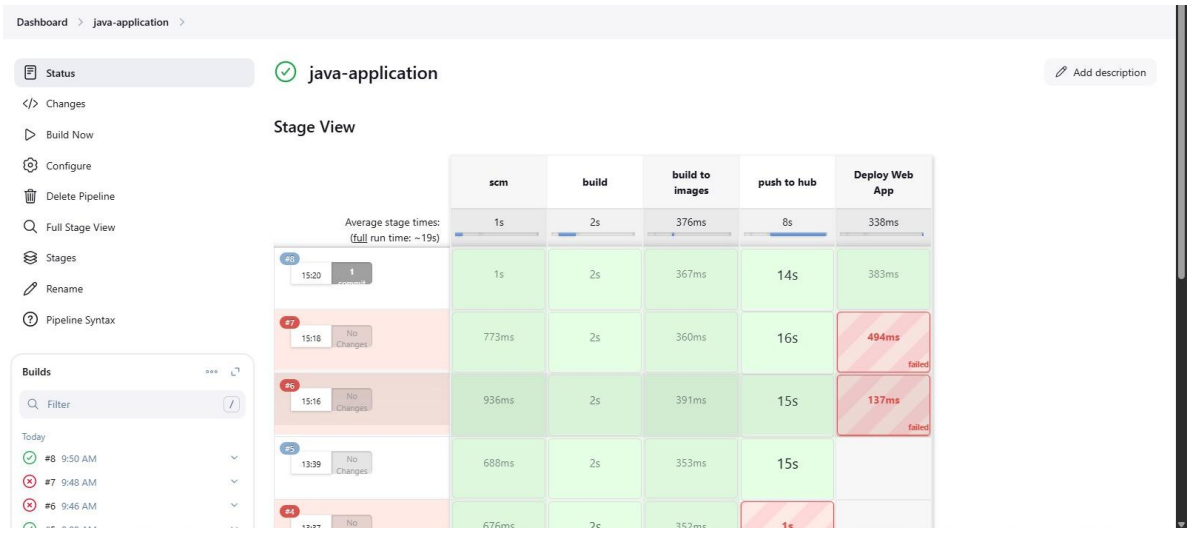
CREATION OF CREDENTIALS



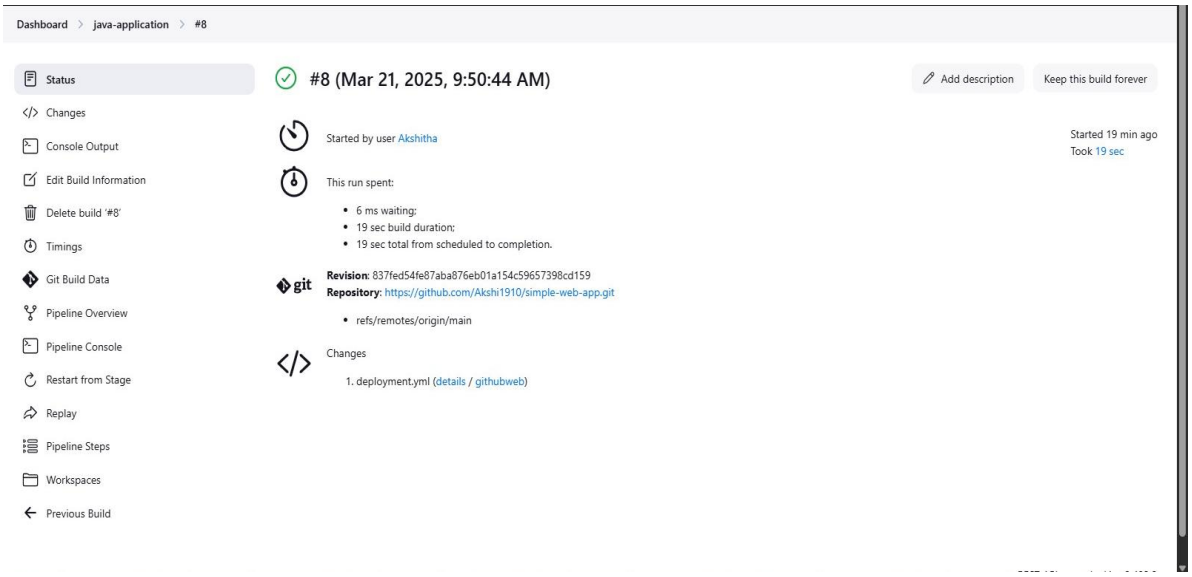
CONNECTING TO KUBERNETES



STAGE VIEW OF BUILD



BUILD SUCCESS



DEPLOYMENT FROM MINIKUBE(TERMINAL)

Commands:

minikube start

sudo nano deployment.yml kubectl

apply -f deployment.yml kubectl get

pod

minikube service my-service

kubectl port-forward svc/my-service 7070:7070

On browser run https://127.0.0.1:7070/maven-web-app

Hello World!