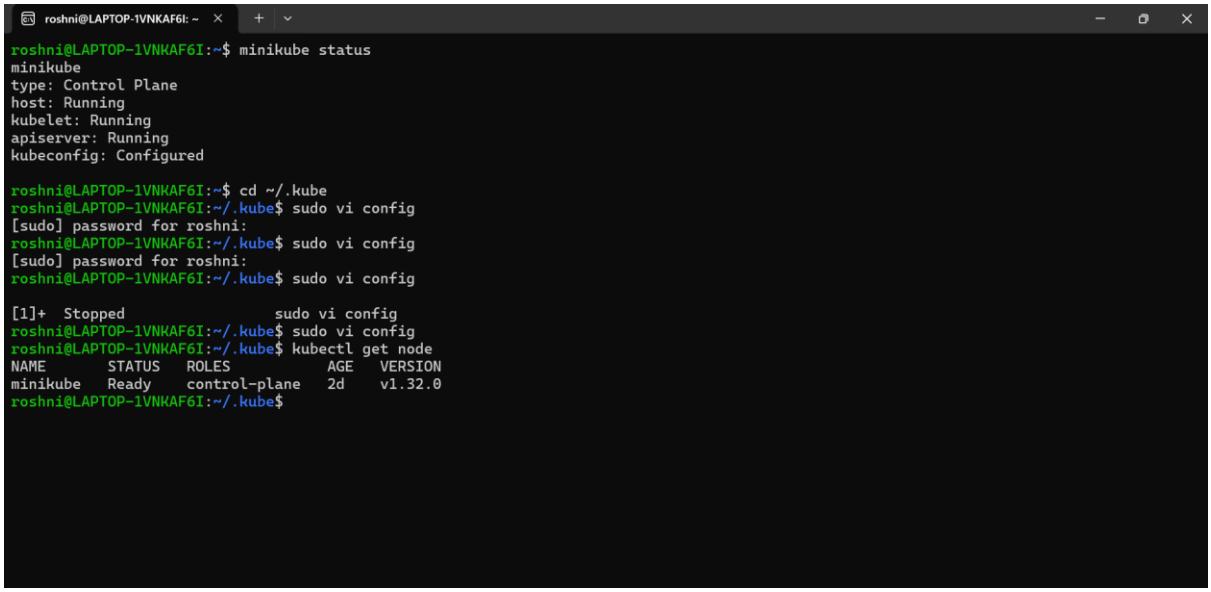
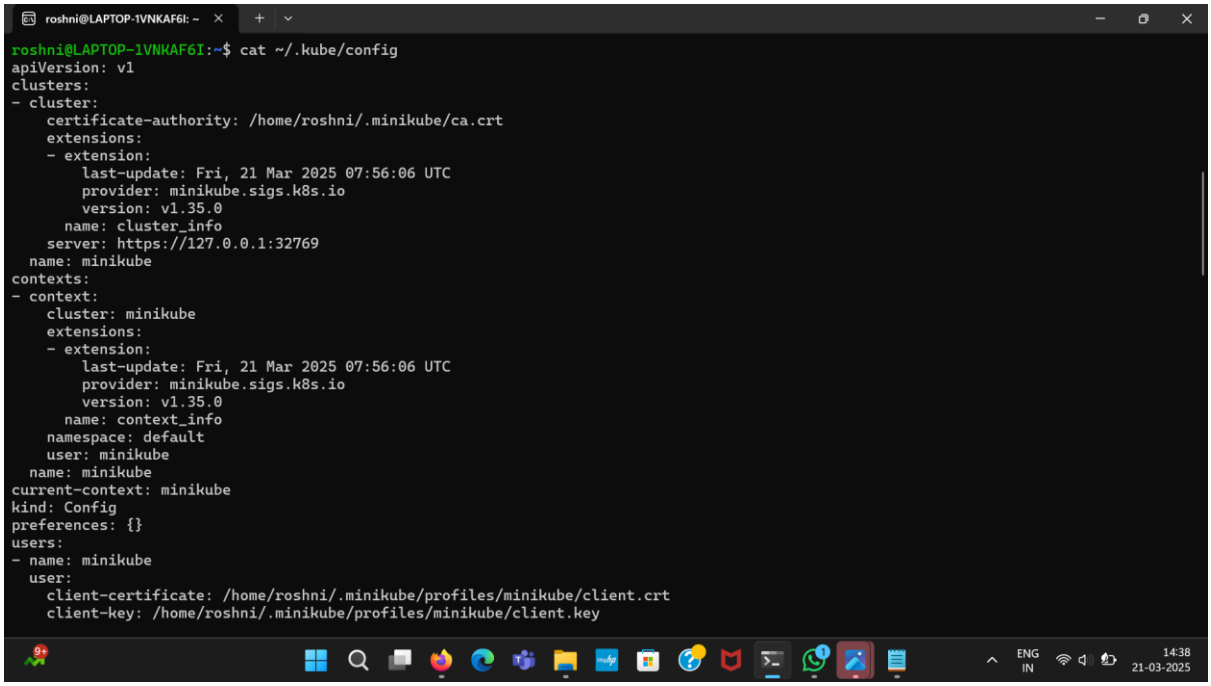
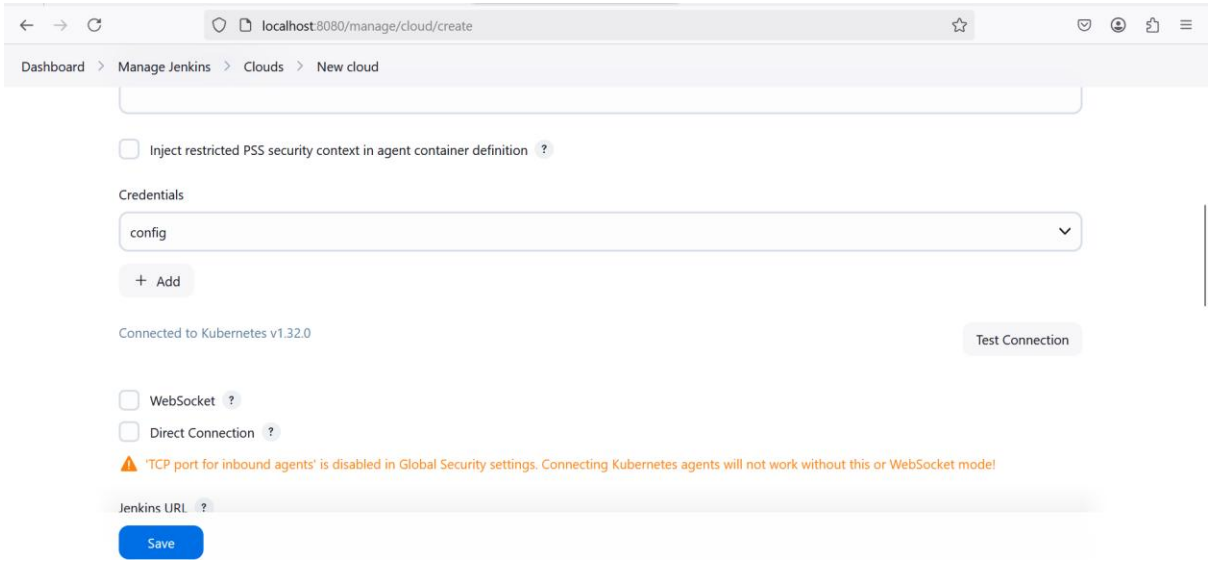
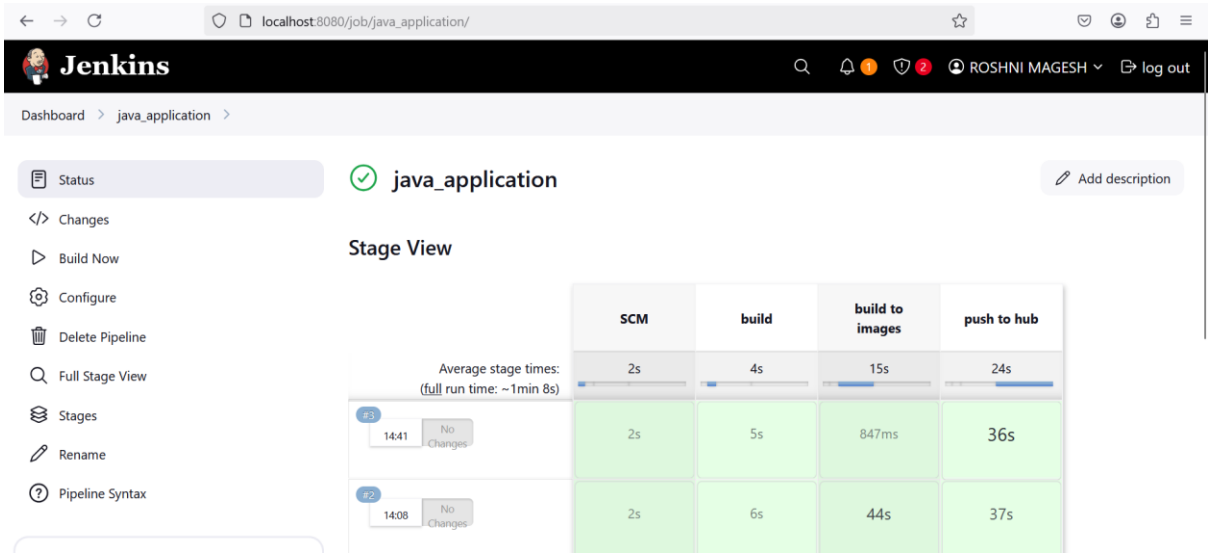
JAVA APPLICATION MINIKUBE DEPLOYMENY

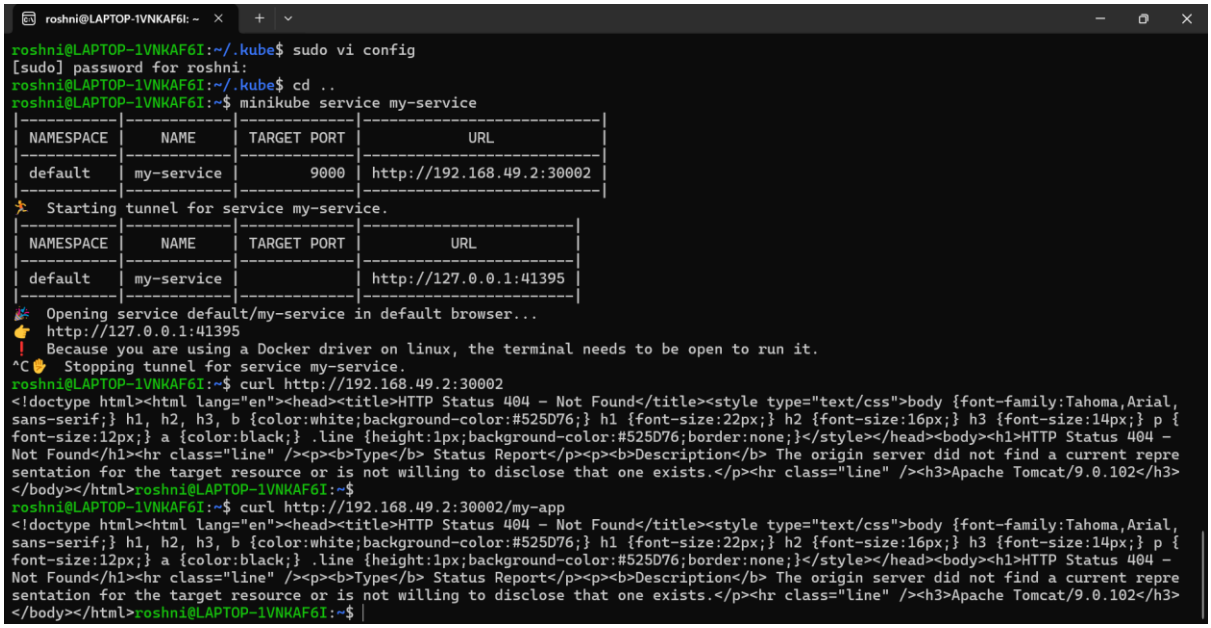


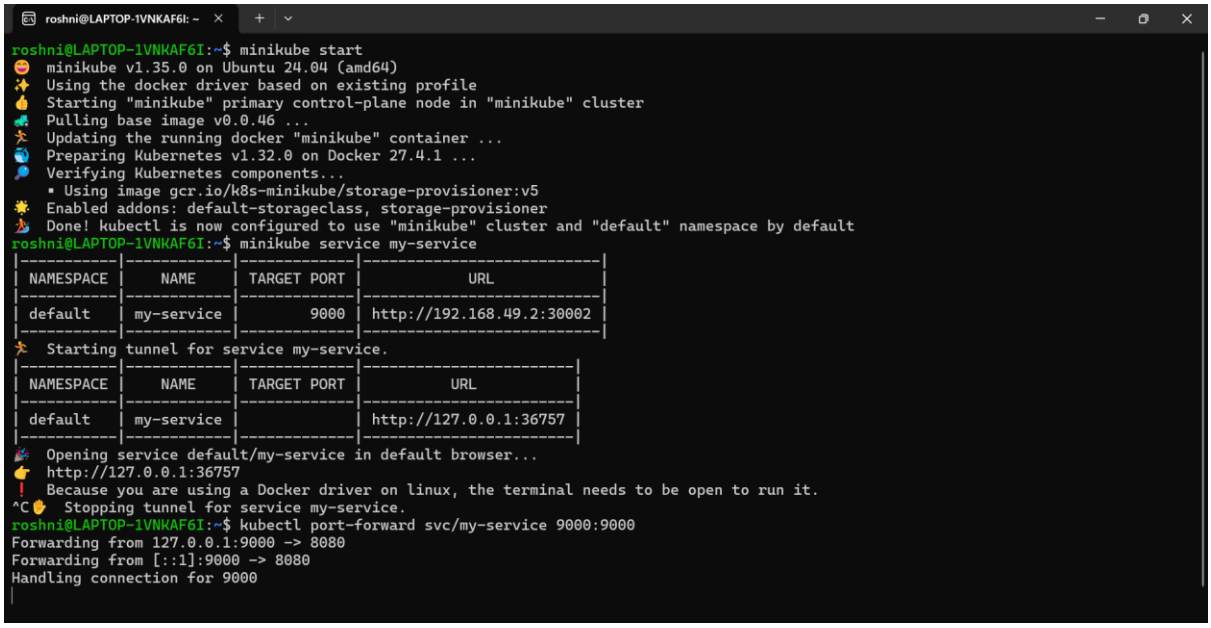


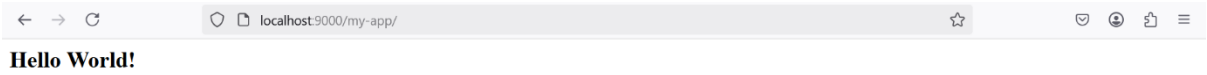

localhost:8080/job/java_application/configure
Dashboard > java_application > Configuration
Configure
General
Triggers
Pipeline
Advanced
stage('build') {
steps {
bat "mvn clean"
bat "mvn install"
}
}
stage('build to images') {
steps {
script{
bat 'docker build -t roshni2108/devops .'
}
}
}
stage('push to hub') {
steps {
script{
withDockerRegistry(credentialsId: 'Docker_cred', url: 'https://index.docker.io/v1/') {
bat 'docker push roshni2108/devops'
}
}
}
}
stage('Deploy App') {
steps {
script{
withKubeConfig(caCertificate: '', clusterName: 'minikube', contextName: 'minikube', credent
sh 'kubectl apply -f deployment.yml --validate=false'
}
}
}
}
}


localhost:8080/manage/cloud/create
Dashboard > Manage Jenkins > Clouds > New cloud
Inject restricted PSS security context in agent container definition
Credentials
config
Add
Connected to Kubernetes v1.32.0
Test Connection
WebSocket
Direct Connection
'TCP port for inbound agents' is disabled in Global Security settings. Connecting Kubernetes agents will not work without this or WebSocket mode!
Jenkins URL
Save


localhost:8080/job/java_application/
Jenkins
ROSHNI MAGESH
log out
Dashboard > java_application >
Status
Changes
Build Now
Configure
Delete Pipeline
Full Stage View
Stages
Rename
Pipeline Syntax
java_application
Add description
Stage View
SCM
build
build to images
push to hub
Average stage times:
(full run time: ~1min 8s)
2s
4s
15s
24s
#3
14:41
No Changes
2s
5s
847ms
36s
#2
14:08
No Changes
2s
6s
44s
37s


roshni@LAPTOP-1VNKAF6I:~/.kube$ sudo vi config
[sudo] password for roshni:
roshni@LAPTOP-1VNKAF6I:~/.kube$ cd ..
roshni@LAPTOP-1VNKAF6I:~$ minikube service my-service
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | 9000 | http://192.168.49.2:30002 |
Starting tunnel for service my-service.
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | | http://127.0.0.1:41395 |
Opening service default/my-service in default browser...
http://127.0.0.1:41395
Because you are using a Docker driver on linux, the terminal needs to be open to run it.
^C Stopping tunnel for service my-service.
roshni@LAPTOP-1VNKAF6I:~$ curl http://192.168.49.2:30002
<!doctype html><html lang="en"><head><title>HTTP Status 404 – Not Found</title><style type="text/css">body {font-family:Tahoma,Arial,
sans-serif;} h1, h2, h3, b {color:white;background-color:#525D76;} h1 {font-size:22px;} h2 {font-size:16px;} h3 {font-size:14px;} p {
font-size:12px;} a {color:black;} .line {height:1px;background-color:#525D76;border:none;}</style></head><body><h1>HTTP Status 404 –
Not Found</h1><hr class="line" /> Type Status Report Description The origin server did not find a current repre
sentation for the target resource or is not willing to disclose that one exists. <hr class="line" /><h3>Apache Tomcat/9.0.102</h3>
</body></html>roshni@LAPTOP-1VNKAF6I:~$
roshni@LAPTOP-1VNKAF6I:~$ curl http://192.168.49.2:30002/my-app
<!doctype html><html lang="en"><head><title>HTTP Status 404 – Not Found</title><style type="text/css">body {font-family:Tahoma,Arial,
sans-serif;} h1, h2, h3, b {color:white;background-color:#525D76;} h1 {font-size:22px;} h2 {font-size:16px;} h3 {font-size:14px;} p {
font-size:12px;} a {color:black;} .line {height:1px;background-color:#525D76;border:none;}</style></head><body><h1>HTTP Status 404 –
Not Found</h1><hr class="line" /> Type Status Report Description The origin server did not find a current repre
sentation for the target resource or is not willing to disclose that one exists. <hr class="line" /><h3>Apache Tomcat/9.0.102</h3>
</body></html>roshni@LAPTOP-1VNKAF6I:~$


roshni@LAPTOP-1VNKAF6I:~$ minikube start
minikube v1.35.0 on Ubuntu 24.04 (amd64)
Using the docker driver based on existing profile
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.46 ...
Updating the running docker "minikube" container ...
Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
Verifying Kubernetes components...
▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: default-storageclass, storage-provisioner
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
roshni@LAPTOP-1VNKAF6I:~$ minikube service my-service
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | 9000 | http://192.168.49.2:30002 |
Starting tunnel for service my-service.
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | | http://127.0.0.1:36757 |
Opening service default/my-service in default browser...
http://127.0.0.1:36757
Because you are using a Docker driver on linux, the terminal needs to be open to run it.
^C Stopping tunnel for service my-service.
roshni@LAPTOP-1VNKAF6I:~$ kubectl port-forward svc/my-service 9000:9000
Forwarding from 127.0.0.1:9000 -> 8080
Forwarding from [::1]:9000 -> 8080
Handling connection for 9000



DEPLOYMENT.YML

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
```

```
      containers:
      - name: my-app
        image: roshni2108/devops:latest
        ports:
        - containerPort: 9000

---

apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-service
spec:
  type: NodePort
  ports:
    - port: 9000
      targetPort: 8080
      nodePort: 30002
  selector:
    apptype: web-backend
```