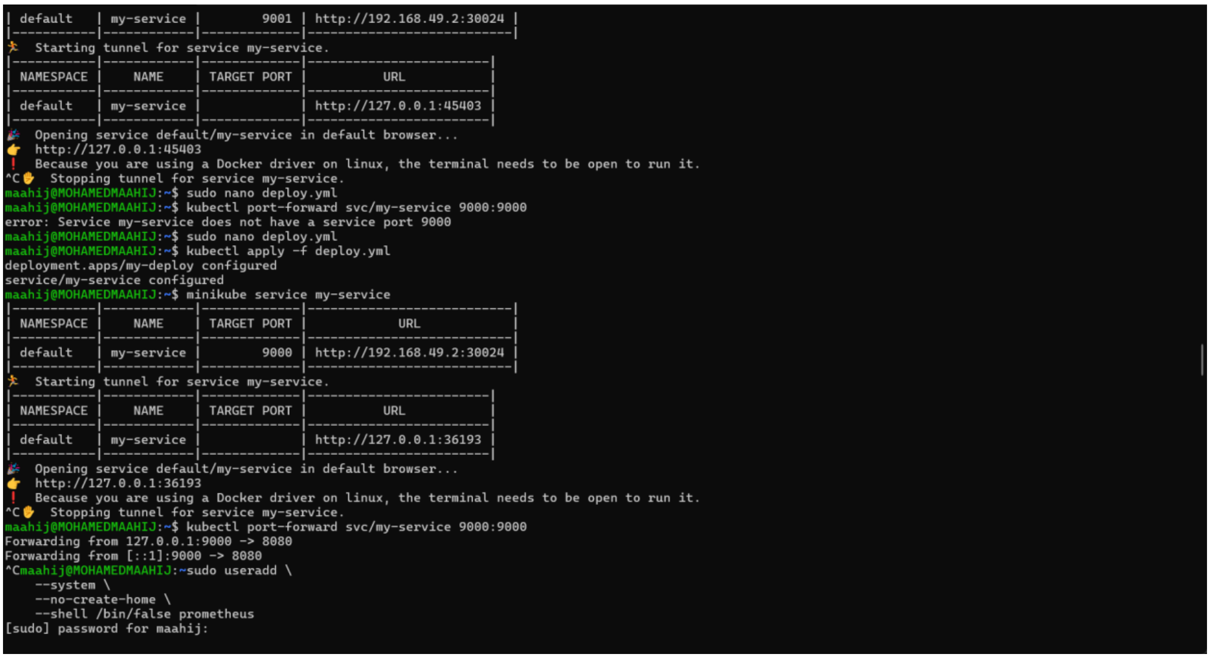
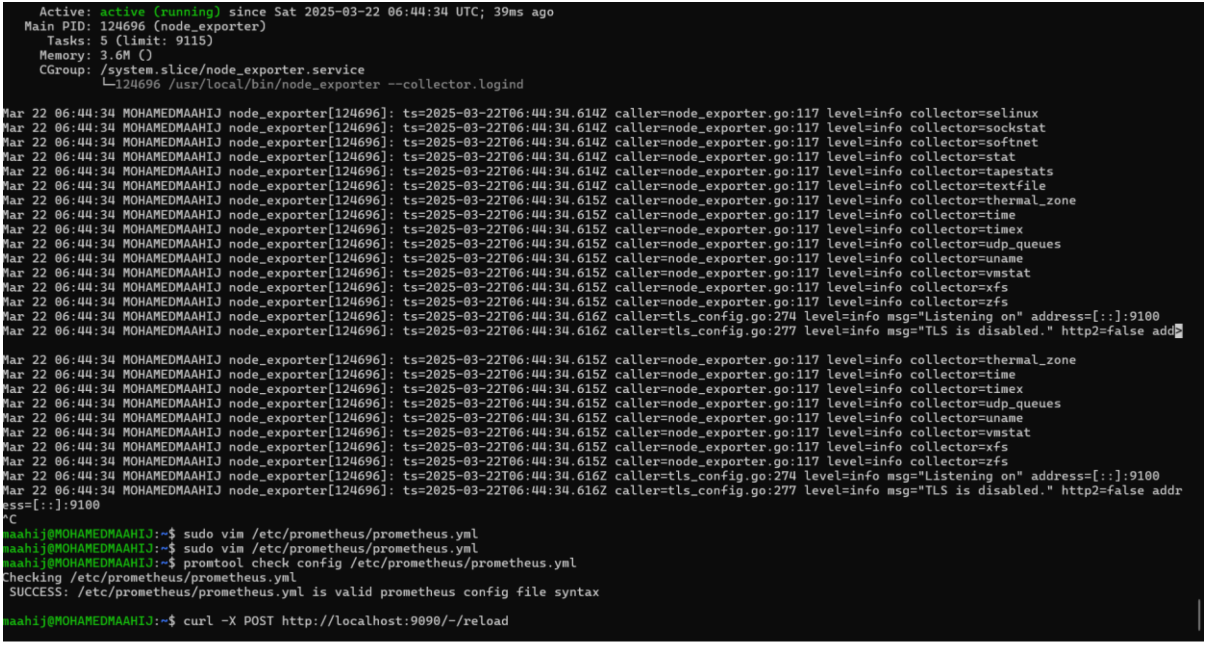
DAY-6

Final project java application minikube deployment

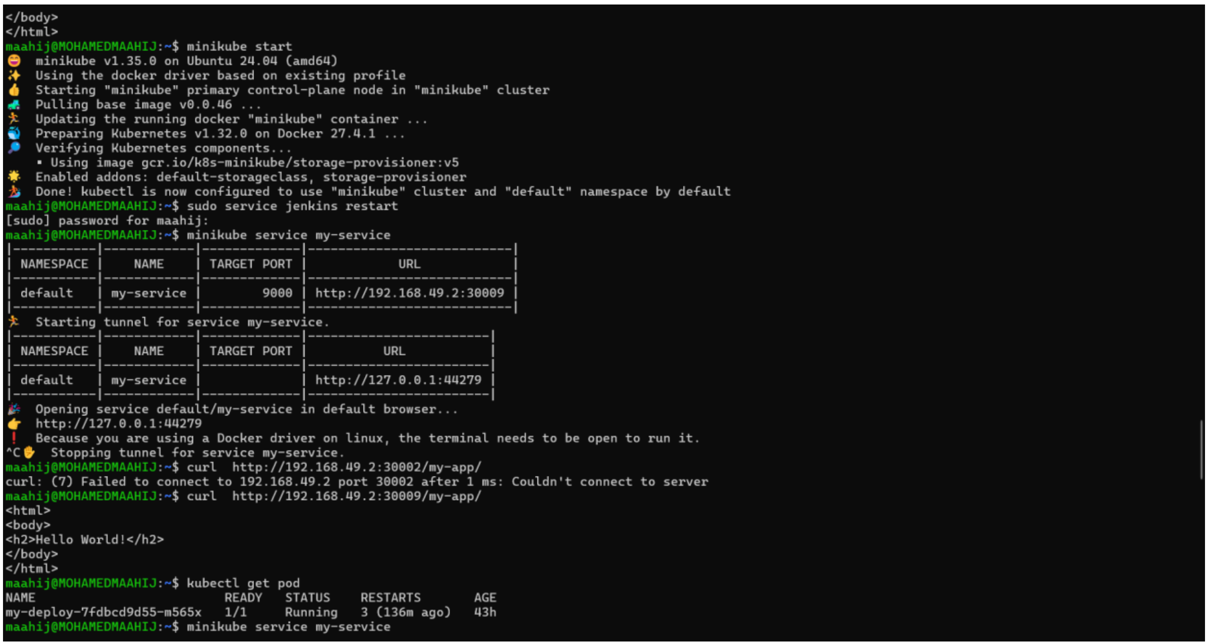


```
| default    | my-service |         9001 | http://192.168.49.2:30024 |
|------------|------------|--------------|---------------------------|
🏃  Starting tunnel for service my-service.
|------------|------------|--------------|------------------------|
| NAMESPACE  |    NAME    | TARGET PORT  |          URL           |
|------------|------------|--------------|------------------------|
| default    | my-service |              | http://127.0.0.1:45403 |
|------------|------------|--------------|------------------------|
🎉  Opening service default/my-service in default browser...
👉  http://127.0.0.1:45403
❗  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.
maahij@MOHAMEDMAAHIJ:~$ sudo nano deploy.yml
maahij@MOHAMEDMAAHIJ:~$ kubectl port-forward svc/my-service 9000:9000
error: Service my-service does not have a service port 9000
maahij@MOHAMEDMAAHIJ:~$ sudo nano deploy.yml
maahij@MOHAMEDMAAHIJ:~$ kubectl apply -f deploy.yml
deployment.apps/my-deploy configured
service/my-service configured
maahij@MOHAMEDMAAHIJ:~$ minikube service my-service
|------------|------------|--------------|---------------------------|
| NAMESPACE  |    NAME    | TARGET PORT  |            URL            |
|------------|------------|--------------|---------------------------|
| default    | my-service |         9000 | http://192.168.49.2:30024 |
|------------|------------|--------------|---------------------------|
🏃  Starting tunnel for service my-service.
|------------|------------|--------------|------------------------|
| NAMESPACE  |    NAME    | TARGET PORT  |          URL           |
|------------|------------|--------------|------------------------|
| default    | my-service |              | http://127.0.0.1:36193 |
|------------|------------|--------------|------------------------|
🎉  Opening service default/my-service in default browser...
👉  http://127.0.0.1:36193
❗  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.
maahij@MOHAMEDMAAHIJ:~$ kubectl port-forward svc/my-service 9000:9000
Forwarding from 127.0.0.1:9000 -> 8080
Forwarding from [::1]:9000 -> 8080
^Cmaahij@MOHAMEDMAAHIJ:~sudo useradd \
    --system \
    --no-create-home \
    --shell /bin/false prometheus
[sudo] password for maahij:
```

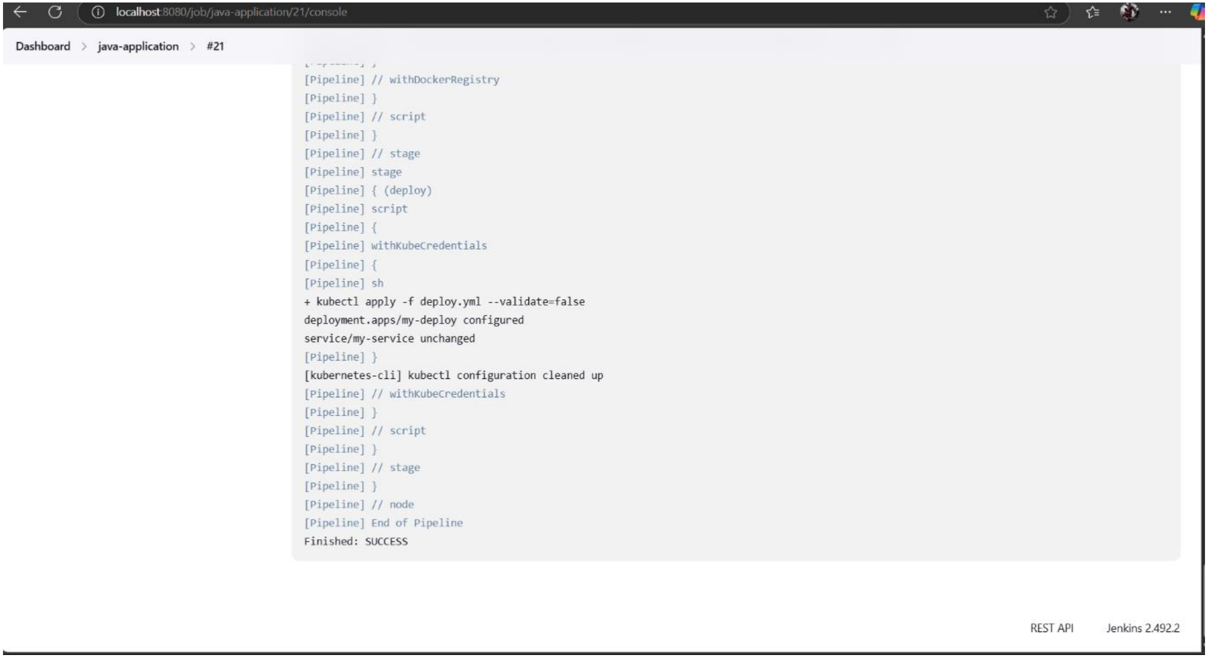


```
     Active: active (running) since Sat 2025-03-22 06:44:34 UTC; 39ms ago
   Main PID: 124696 (node_exporter)
      Tasks: 5 (limit: 9115)
     Memory: 3.6M ()
     CGroup: /system.slice/node_exporter.service
             └─124696 /usr/local/bin/node_exporter --collector.logind

Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.614Z caller=node_exporter.go:117 level=info collector=selinux
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.614Z caller=node_exporter.go:117 level=info collector=sockstat
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.614Z caller=node_exporter.go:117 level=info collector=softnet
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.614Z caller=node_exporter.go:117 level=info collector=stat
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.614Z caller=node_exporter.go:117 level=info collector=tapestats
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.614Z caller=node_exporter.go:117 level=info collector=textfile
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.615Z caller=node_exporter.go:117 level=info collector=thermal_zone
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.615Z caller=node_exporter.go:117 level=info collector=time
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.615Z caller=node_exporter.go:117 level=info collector=timex
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.615Z caller=node_exporter.go:117 level=info collector=udp_queues
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.615Z caller=node_exporter.go:117 level=info collector=uname
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.615Z caller=node_exporter.go:117 level=info collector=vmstat
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.615Z caller=node_exporter.go:117 level=info collector=xfs
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.615Z caller=node_exporter.go:117 level=info collector=zfs
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.616Z caller=tls_config.go:274 level=info msg="Listening on" address=[::]:9100
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.616Z caller=tls_config.go:277 level=info msg="TLS is disabled." http2=false add>

Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.615Z caller=node_exporter.go:117 level=info collector=thermal_zone
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.615Z caller=node_exporter.go:117 level=info collector=time
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.615Z caller=node_exporter.go:117 level=info collector=timex
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.615Z caller=node_exporter.go:117 level=info collector=udp_queues
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.615Z caller=node_exporter.go:117 level=info collector=uname
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.615Z caller=node_exporter.go:117 level=info collector=vmstat
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.615Z caller=node_exporter.go:117 level=info collector=xfs
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.615Z caller=node_exporter.go:117 level=info collector=zfs
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.616Z caller=tls_config.go:274 level=info msg="Listening on" address=[::]:9100
Mar 22 06:44:34 MOHAMEDMAAHIJ node_exporter[124696]: ts=2025-03-22T06:44:34.616Z caller=tls_config.go:277 level=info msg="TLS is disabled." http2=false addr
ess=[::]:9100
^C
maahij@MOHAMEDMAAHIJ:~$ sudo vim /etc/prometheus/prometheus.yml
maahij@MOHAMEDMAAHIJ:~$ sudo vim /etc/prometheus/prometheus.yml
maahij@MOHAMEDMAAHIJ:~$ promtool check config /etc/prometheus/prometheus.yml
Checking /etc/prometheus/prometheus.yml
 SUCCESS: /etc/prometheus/prometheus.yml is valid prometheus config file syntax

maahij@MOHAMEDMAAHIJ:~$ curl -X POST http://localhost:9090/-/reload
```

FINAL PROJECT DAY_6

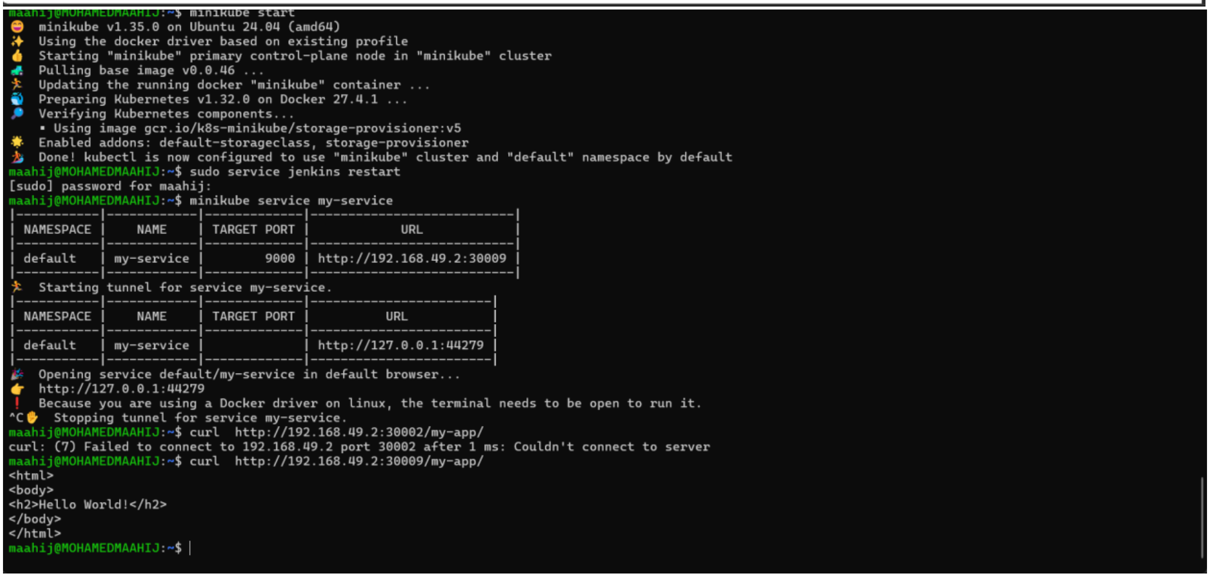


```
</body>
</html>
maahij@MOHAMEDMAAHIJ:~$ minikube start
😄  minikube v1.35.0 on Ubuntu 24.04 (amd64)
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.46 ...
🏃  Updating the running docker "minikube" 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
maahij@MOHAMEDMAAHIJ:~$ sudo service jenkins restart
[sudo] password for maahij:
maahij@MOHAMEDMAAHIJ:~$ minikube service my-service
|-----------|------------|-------------|---------------------------|
| NAMESPACE |    NAME    | TARGET PORT |            URL            |
|-----------|------------|-------------|---------------------------|
| default   | my-service |        9000 | http://192.168.49.2:30009 |
|-----------|------------|-------------|---------------------------|
🏃  Starting tunnel for service my-service.
|-----------|------------|-------------|------------------------|
| NAMESPACE |    NAME    | TARGET PORT |          URL           |
|-----------|------------|-------------|------------------------|
| default   | my-service |             | http://127.0.0.1:44279 |
|-----------|------------|-------------|------------------------|
🎉  Opening service default/my-service in default browser...
👉  http://127.0.0.1:44279
❗  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.
maahij@MOHAMEDMAAHIJ:~$ curl  http://192.168.49.2:30002/my-app/
curl: (7) Failed to connect to 192.168.49.2 port 30002 after 1 ms: Couldn't connect to server
maahij@MOHAMEDMAAHIJ:~$ curl  http://192.168.49.2:30009/my-app/
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
maahij@MOHAMEDMAAHIJ:~$ kubectl get pod
NAME                           READY   STATUS    RESTARTS       AGE
my-deploy-7fdbcd9d55-m565x     1/1     Running   3 (136m ago)   43h
maahij@MOHAMEDMAAHIJ:~$ minikube service my-service
```



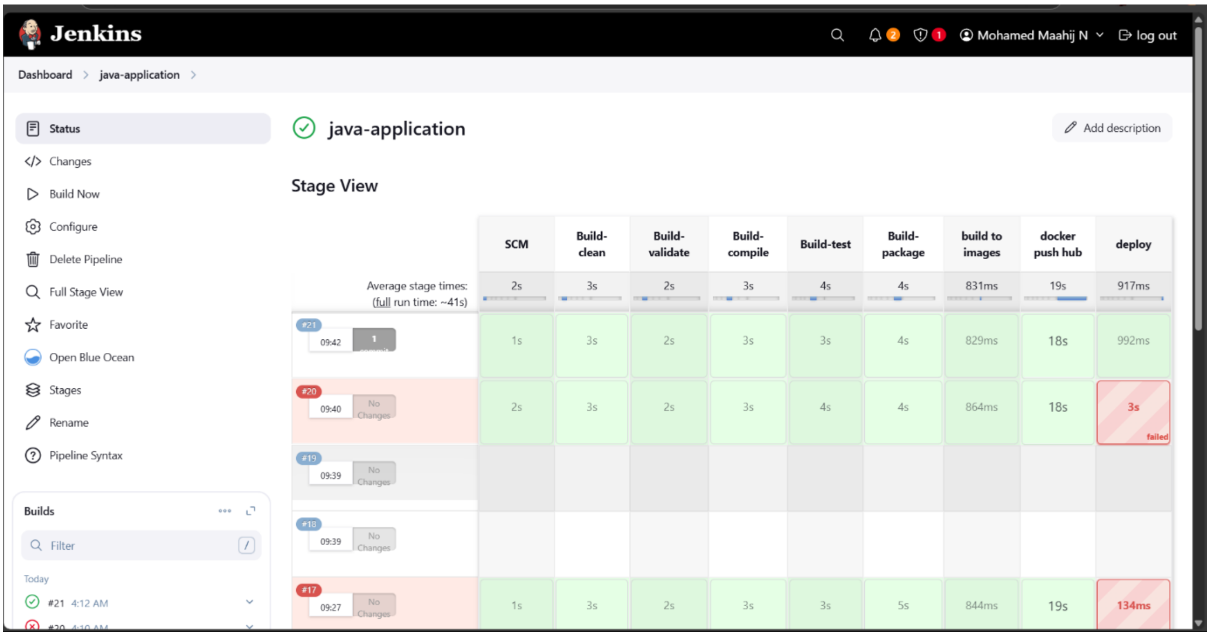
localhost:8080/job/java-application/21/console
Dashboard > java-application > #21

```
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (deploy)
[Pipeline] script
[Pipeline] {
[Pipeline] withKubeCredentials
[Pipeline] {
[Pipeline] sh
+ kubectl apply -f deploy.yml --validate=false
deployment.apps/my-deploy configured
service/my-service unchanged
[Pipeline] }
[kubernetes-cli] kubectl configuration cleaned up
[Pipeline] // withKubeCredentials
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```



```
maahij@MOHAMEDMAAHIJ:~$ 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
maahij@MOHAMEDMAAHIJ:~$ sudo service jenkins restart
[sudo] password for maahij:
maahij@MOHAMEDMAAHIJ:~$ minikube service my-service
|-----------|------------|-------------|---------------------------|
| NAMESPACE |    NAME    | TARGET PORT |            URL            |
|-----------|------------|-------------|---------------------------|
| default   | my-service |        9000 | http://192.168.49.2:30009 |
|-----------|------------|-------------|---------------------------|
🏃  Starting tunnel for service my-service.
|-----------|------------|-------------|------------------------|
| NAMESPACE |    NAME    | TARGET PORT |          URL           |
|-----------|------------|-------------|------------------------|
| default   | my-service |             | http://127.0.0.1:44279 |
|-----------|------------|-------------|------------------------|
🎉  Opening service default/my-service in default browser...
👉  http://127.0.0.1:44279
❗  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.
maahij@MOHAMEDMAAHIJ:~$ curl  http://192.168.49.2:30002/my-app/
curl: (7) Failed to connect to 192.168.49.2 port 30002 after 1 ms: Couldn't connect to server
maahij@MOHAMEDMAAHIJ:~$ curl  http://192.168.49.2:30009/my-app/
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
maahij@MOHAMEDMAAHIJ:~$ |
```

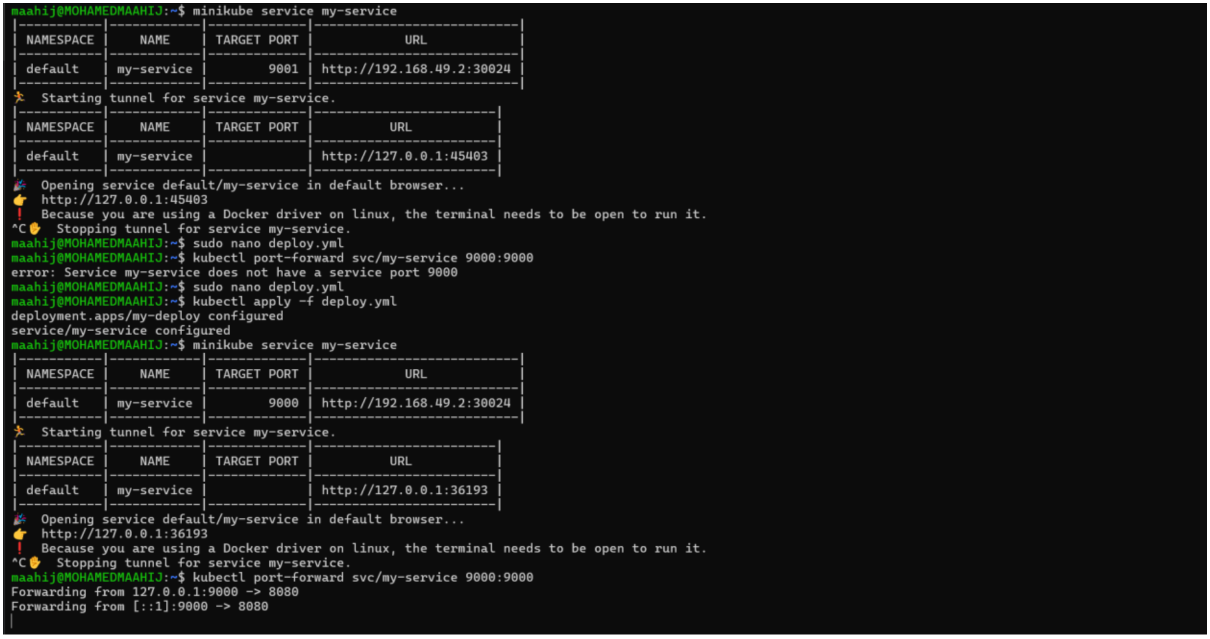
FINAL PROJECT DAY_6



java-application

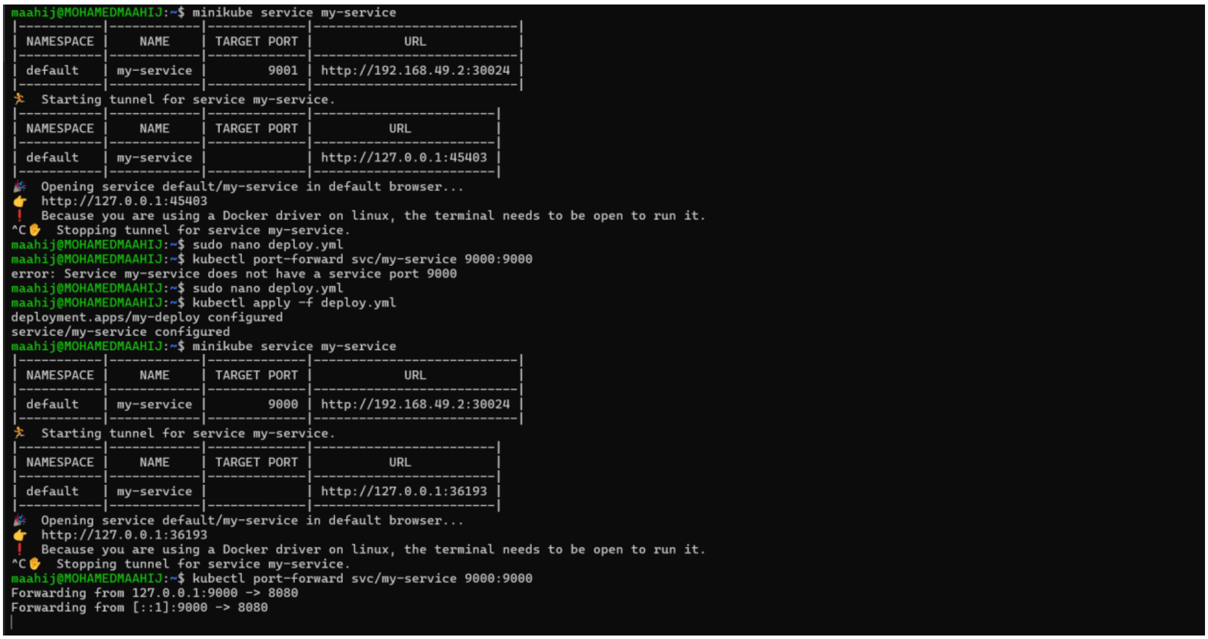
Stage View

	SCM	Build-clean	Build-validate	Build-compile	Build-test	Build-package	build to images	docker push hub	deploy
Average stage times: (full run time: ~41s)	2s	3s	2s	3s	4s	4s	831ms	19s	917ms
#21 09:42 1 commit	1s	3s	2s	3s	3s	4s	829ms	18s	992ms
#20 09:40 No Changes	2s	3s	2s	3s	4s	4s	864ms	18s	3s failed
#19 09:39 No Changes									
#18 09:39 No Changes									
#17 09:27 No Changes	1s	3s	2s	3s	3s	5s	844ms	19s	134ms

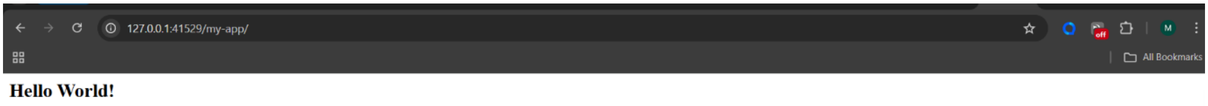


```
maahij@MOHAMEDMAAHIJ:~$ minikube service my-service
|-----------|------------|-------------|---------------------------|
| NAMESPACE |    NAME    | TARGET PORT |            URL            |
|-----------|------------|-------------|---------------------------|
| default   | my-service |        9001 | http://192.168.49.2:30024 |
|-----------|------------|-------------|---------------------------|
🏃  Starting tunnel for service my-service.
|-----------|------------|-------------|------------------------|
| NAMESPACE |    NAME    | TARGET PORT |          URL           |
|-----------|------------|-------------|------------------------|
| default   | my-service |             | http://127.0.0.1:45403 |
|-----------|------------|-------------|------------------------|
🎉  Opening service default/my-service in default browser...
👉  http://127.0.0.1:45403
❗  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.
maahij@MOHAMEDMAAHIJ:~$ sudo nano deploy.yml
maahij@MOHAMEDMAAHIJ:~$ kubectl port-forward svc/my-service 9000:9000
error: Service my-service does not have a service port 9000
maahij@MOHAMEDMAAHIJ:~$ sudo nano deploy.yml
maahij@MOHAMEDMAAHIJ:~$ kubectl apply -f deploy.yml
deployment.apps/my-deploy configured
service/my-service configured
maahij@MOHAMEDMAAHIJ:~$ minikube service my-service
|-----------|------------|-------------|---------------------------|
| NAMESPACE |    NAME    | TARGET PORT |            URL            |
|-----------|------------|-------------|---------------------------|
| default   | my-service |        9000 | http://192.168.49.2:30024 |
|-----------|------------|-------------|---------------------------|
🏃  Starting tunnel for service my-service.
|-----------|------------|-------------|------------------------|
| NAMESPACE |    NAME    | TARGET PORT |          URL           |
|-----------|------------|-------------|------------------------|
| default   | my-service |             | http://127.0.0.1:36193 |
|-----------|------------|-------------|------------------------|
🎉  Opening service default/my-service in default browser...
👉  http://127.0.0.1:36193
❗  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.
maahij@MOHAMEDMAAHIJ:~$ kubectl port-forward svc/my-service 9000:9000
Forwarding from 127.0.0.1:9000 -> 8080
Forwarding from [::1]:9000 -> 8080
```

FINAL PROJECT   DAY_6



```
maahij@MOHAMEDMAAHIJ:~$ minikube service my-service
|-----------|------------|-------------|---------------------------|
| NAMESPACE |    NAME    | TARGET PORT |            URL            |
|-----------|------------|-------------|---------------------------|
| default   | my-service |        9001 | http://192.168.49.2:30024 |
|-----------|------------|-------------|---------------------------|
🏃  Starting tunnel for service my-service.
|-----------|------------|-------------|------------------------|
| NAMESPACE |    NAME    | TARGET PORT |          URL           |
|-----------|------------|-------------|------------------------|
| default   | my-service |             | http://127.0.0.1:45403 |
|-----------|------------|-------------|------------------------|
🎉  Opening service default/my-service in default browser...
👉  http://127.0.0.1:45403
❗  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.
maahij@MOHAMEDMAAHIJ:~$ sudo nano deploy.yml
maahij@MOHAMEDMAAHIJ:~$ kubectl port-forward svc/my-service 9000:9000
error: Service my-service does not have a service port 9000
maahij@MOHAMEDMAAHIJ:~$ sudo nano deploy.yml
maahij@MOHAMEDMAAHIJ:~$ kubectl apply -f deploy.yml
deployment.apps/my-deploy configured
service/my-service configured
maahij@MOHAMEDMAAHIJ:~$ minikube service my-service
|-----------|------------|-------------|---------------------------|
| NAMESPACE |    NAME    | TARGET PORT |            URL            |
|-----------|------------|-------------|---------------------------|
| default   | my-service |        9000 | http://192.168.49.2:30024 |
|-----------|------------|-------------|---------------------------|
🏃  Starting tunnel for service my-service.
|-----------|------------|-------------|------------------------|
| NAMESPACE |    NAME    | TARGET PORT |          URL           |
|-----------|------------|-------------|------------------------|
| default   | my-service |             | http://127.0.0.1:36193 |
|-----------|------------|-------------|------------------------|
🎉  Opening service default/my-service in default browser...
👉  http://127.0.0.1:36193
❗  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.
maahij@MOHAMEDMAAHIJ:~$ kubectl port-forward svc/my-service 9000:9000
Forwarding from 127.0.0.1:9000 -> 8080
Forwarding from [::1]:9000 -> 8080
|
```



Hello World!

FINAL PROJECT   DAY_6

PIPELINE SCRIPT

```
pipeline {
    agent any


    stages {
        stage('SCM') {
            steps {
                git branch: 'main', url: 'https://github.com/Maahij2004/web-app.git'
            }
        }
        stage('Build-clean') {
            steps{
                sh 'mvn clean'
            }
        }
              stage('Build-validate') {
            steps{
                            sh 'mvn validate'
            }
        }
              stage('Build-compile') {
            steps{
                            sh 'mvn compile'
            }
        }
              stage('Build-test') {
            steps{
                            sh 'mvn test'
            }
        }
              stage('Build-package') {
```

```
      steps{
              sh 'mvn package'
      }
    }
    stage('build to images') {
      steps {
      script{
        sh "docker build -t maahij14/simplewebapp2 ."
      }
      }
    }
    stage('docker push hub') {
      steps {
      script{
       withDockerRegistry(credentialsId: 'docker_cred', url: 'https://index.docker.io/v1/') {
        sh 'docker push maahij14/simplewebapp2'
      }
      }
      }

    }
    stage('deploy') {
      steps {
      script{
       withKubeCredentials(kubectlCredentials: [[caCertificate: '', clusterName: 'minikube', contextName: 'minikube', credentialsId: 'cred_1', namespace: '', serverUrl: 'https://192.168.39.226:8443']]) {
  sh 'kubectl apply -f deploy.yml --validate=false'
}
      }
      }
      }
```

```
        }

    }
```

Deploy.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:
        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
```