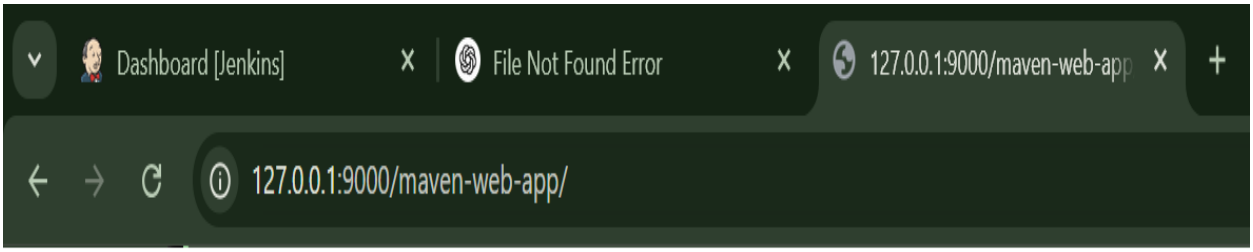
JAVA APPLICATION MINIKUBE DEPLOYMENT



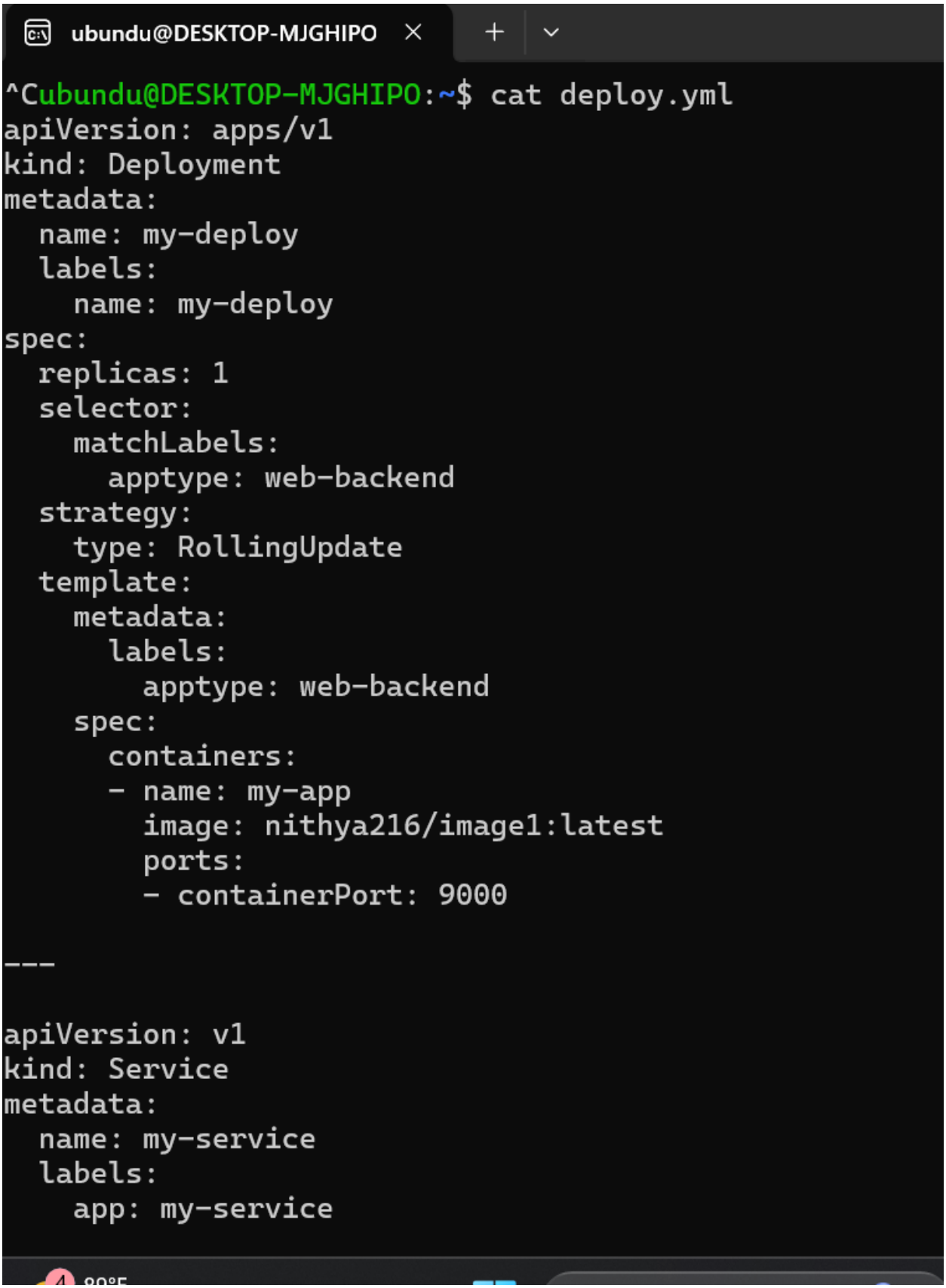
Hello World!

```
service/my-service created
ubundu@DESKTOP-MJGHIPO:~$ 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.
docker@127.0.0.1's password: |-----------|------------|-------------|------------------------|
| NAMESPACE |    NAME    | TARGET PORT |          URL           |
|-----------|------------|-------------|------------------------|
| default   | my-service |             | http://127.0.0.1:37003 |
|-----------|------------|-------------|------------------------|
🎉  Opening service default/my-service in default browser...
👉  http://127.0.0.1:37003
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.

✋  Stopping tunnel for service my-service.
ubundu@DESKTOP-MJGHIPO:~$ 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
Handling connection for 9000
Handling connection for 9000
Handling connection for 9000
```

DEPLOYMENT.YML

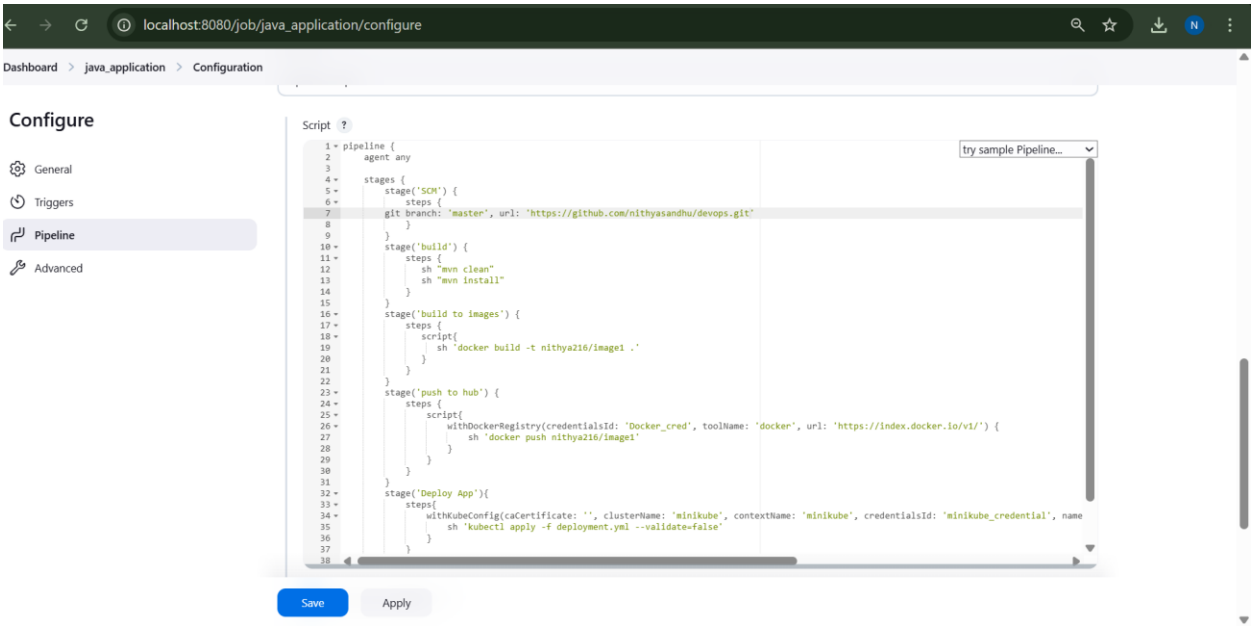
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
      - name: my-app
        image: nithya216/image1:latest
```

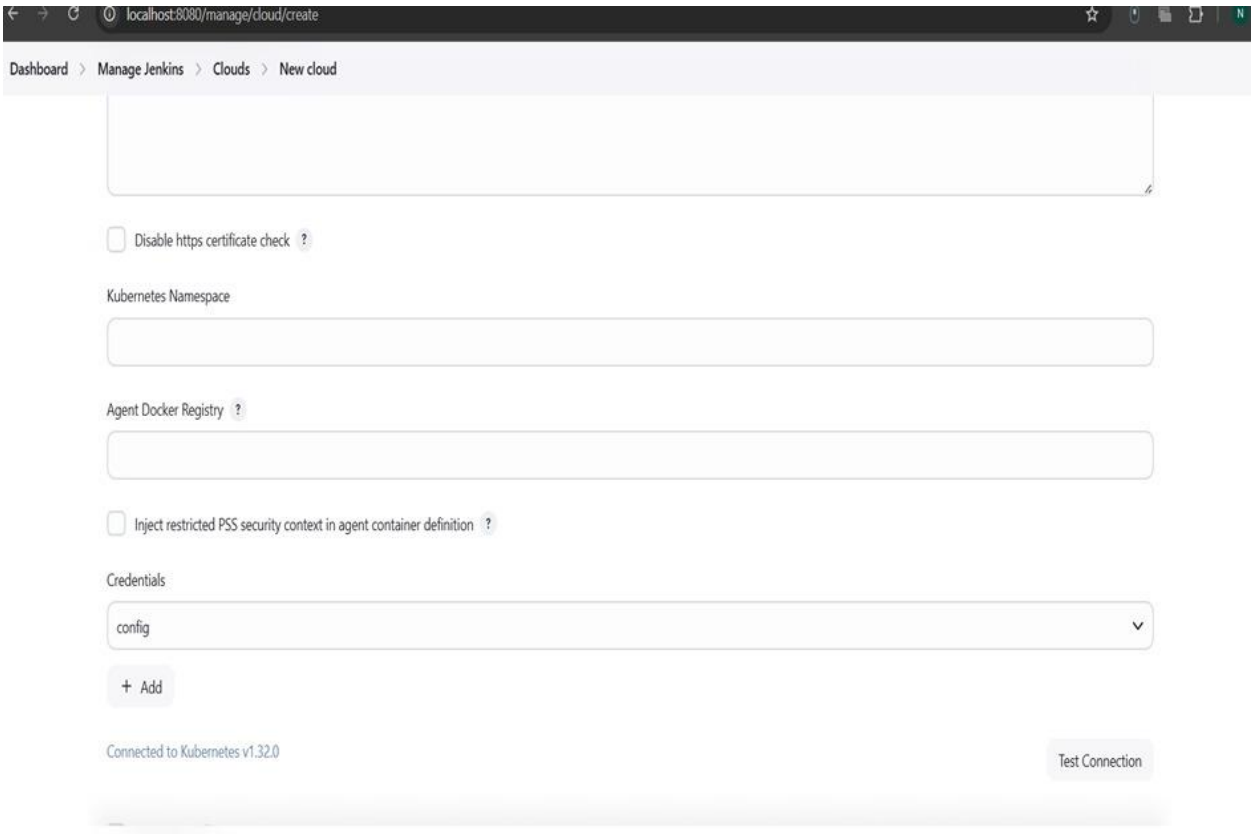
```
        ports:
        - containerPort: 9000


---


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
```


ubundu@DESKTOP-MJGHIPO
^Cubundu@DESKTOP-MJGHIPO:~$ cat 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: nithya216/image1:latest
ports:
- containerPort: 9000
---
apiVersion: v1
kind: Service
metadata:
name: my-service
labels:
app: my-service

```
pipeline {
    agent any

    stages {
        stage('SCM') {
            steps {
        git branch: 'master', url: 'https://github.com/nithyasandhu/devops.git'
            }
        }
        stage('build') {
            steps {
                sh "mvn clean"
                sh "mvn install"
            }
        }
        stage('build to images') {
            steps {
                script{
                    sh 'docker build -t nithya216/image1 .'
                }
            }
        }
        stage('push to hub') {
            steps {
                script{
```

```
                withDockerRegistry(credentialsId: 'Docker_cred', toolName: 'docker', url:
'https://index.docker.io/v1/') {
                    sh 'docker push nithya216/image1'
                }
            }
        }
    }
    stage('Deploy App'){
        steps{
            withKubeConfig(caCertificate: '', clusterName: 'minikube', contextName: 'minikube',
credentialsId: 'minikube_credential', namespace: '', restrictKubeConfigAccess: false, serverUrl:
'https://192.168.39.226:8443') {
                sh 'kubectl apply -f deployment.yml --validate=false'
            }
        }
    }
  }
}
```


localhost:8080/job/java_application/configure
Dashboard > java_application > Configuration
Configure
General
Triggers
Pipeline
Advanced
Script
try sample Pipeline...
pipeline {
agent any
stages {
stage('SCM') {
steps {
git branch: 'master', url: 'https://github.com/nithyasandhu/devops.git'
}
}
stage('build') {
steps {
sh "mvn clean"
sh "mvn install"
}
}
stage('build to images') {
steps {
script{
sh 'docker build -t nithya216/image1 .'
}
}
}
stage('push to hub') {
steps {
script{
withDockerRegistry(credentialsId: 'Docker_cred', toolName: 'docker', url: 'https://index.docker.io/v1/') {
sh 'docker push nithya216/image1'
}
}
}
}
stage('Deploy App'){
steps{
withKubeConfig(caCertificate: '', clusterName: 'minikube', contextName: 'minikube', credentialsId: 'minikube_credential', name
sh 'kubectl apply -f deployment.yml --validate=false'
}
}
Save
Apply


localhost:8080/manage/cloud/create
Dashboard > Manage Jenkins > Clouds > New cloud
Disable https certificate check
Kubernetes Namespace
Agent Docker Registry
Inject restricted PSS security context in agent container definition
Credentials
config
+ Add
Connected to Kubernetes v1.32.0
Test Connection