

## Pipeline script

Pipelinescript-

Jenkins JAVA

Application

```
pipeline {
```

```
    agent any
```

```
    stages { stage('scm'){
```

```
        steps{
```

```
            gitbranch:" https://github.com/NavinkumarD/Guvi-Tasks-DevOps.git"
```

```
        }
```

```
    }
```

```
    stage('builb
```

```
        -clean'){
```

```
        steps {
```

```
            sh"mvn clean"
```

```
        }
```

```
    }
```

```
    stage('build-
```

```
        validate'){
```

```
        steps {
```

```
            sh"mvn validate"
```

```
}
```

```
}
```

```
stage('buil
```

```
d-com'){
```

```
steps {
```

```
sh"mvn compile"
```

```
}
```

```
}
```

```
stage('buil
```

```
d-test'){
```

```
steps {
```

```
sh"mvn test"
```

```
}
```

```
}
```

```
stage('build-
```

```
install'){
```

```
steps {
```

```
sh"mvn package"
```

```
}
```

```
}
```

```
stage('buildtoima
```

```
ges'){
```

```
steps {
```

```
script{
```

```

        sh'docker build-t.'
    }
}
}
stage('pushtohub
    '{ steps {
        script{
            withDockerRegistry(credentialsId:'Docker_cred',url: https://index.docker.io/v1/){ sh 'docker push '
            }
        }
    }
}

stage('Deploy
    App'){ steps
    {
        withKubeConfig(caCertificate:'',clusterName:'minikube',contextName:'minikube',
credentialsId:'mukubeconfig_011',namespace:'',restrictKubeConfigAccess:false,serverUrl:
        ' ') {
            sh'kubectl apply -fdeployment.yml --validate=false'
        }
    }
}

stage('Test'){ steps {

```

```
withKubeConfig(caCertificate: "", clusterName: 'minikube', contextName: 'minikube',
credentialsId: 'mukubeconfig_011', namespace: "", restrictKubeConfigAccess: false, serverUrl:
    ') {
```

```
    sh'minikubeservice my-service --url | xargs curl'
```

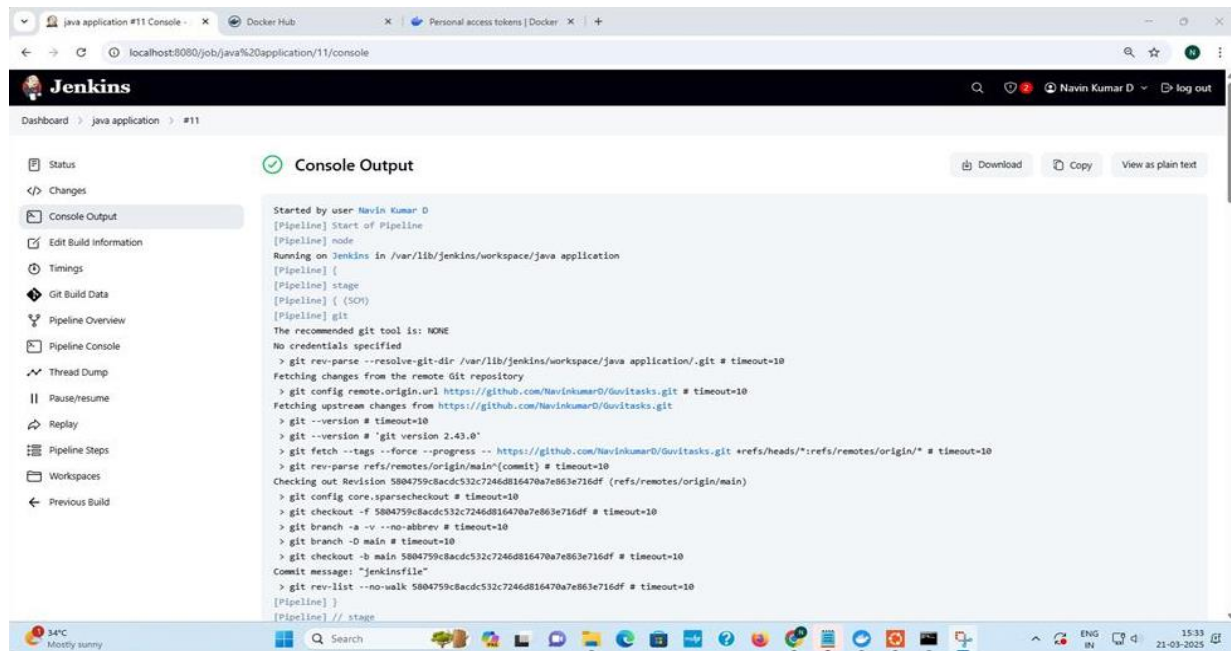
```
    }
```

```
  }
```

```
  }
```

```
  }
```

```
}
```



## Deployment.yml

```

apiVersion:apps/v1 kind:Deployment
metadata:
  name: my-deploy
  namespace:my-bank
  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-web
        image: 22csl258/mysimplewebapplication:tagname
        ports:
          - containerPort: 9001
  apiVersion:
  v1 kind:
  Service
  metadata:
    name: my-
    service
    namespace: my-
    y-bank
    labels:
      app: my-
  service spec:
    type: NodePort
    ports:
      - port: 9001
        targetPort: 8080

```

#### Service.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: 22csl258/simplewebapp:latest
          ports:

```

