

DevOps Day-6

Kubectrl pipeline code:

```
pipeline {
  agent any

  stages {
    stage('SCM') {
      steps {
        git branch: 'main', url: 'https://github.com/ragules/simple-
web-app'
      }
    }
    stage('Build'){
      steps {
        bat 'mvn clean install'
      }
    }
    stage('build to images') {
      steps {
        script{
          bat "docker build -t ragul1177/webapplication ."
        }
      }
    }
  }
}
```

```

    }
  }
  stage('docker push hub') {
    steps {
      script{
        withDockerRegistry(credentialsId: 'docker_crud', url:
'https://index.docker.io/v1/') {
          bat 'docker push ragul1177/webapplication'
        }
      }
    }
  }
  // stage('test') {
  //   steps {
  //     withKubeConfig(caCertificate: "", clusterName: 'minikube',
contextName: 'minikube', credentialsId: 'minikube', namespace: "",
restrictKubeConfigAccess: false, serverUrl:
'https://192.168.39.226:8443')
  //     sh 'kubectl apply -f deploy.yml --validate=false'
  //   }
  }
}

```

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: praneshc/webapplication
          ports:
            - containerPort: 9000
```

apiVersion: v1

kind: Service

metadata:

name: my-service

labels:

app: my-service

spec:

type: NodePort

ports:

- port: 9000

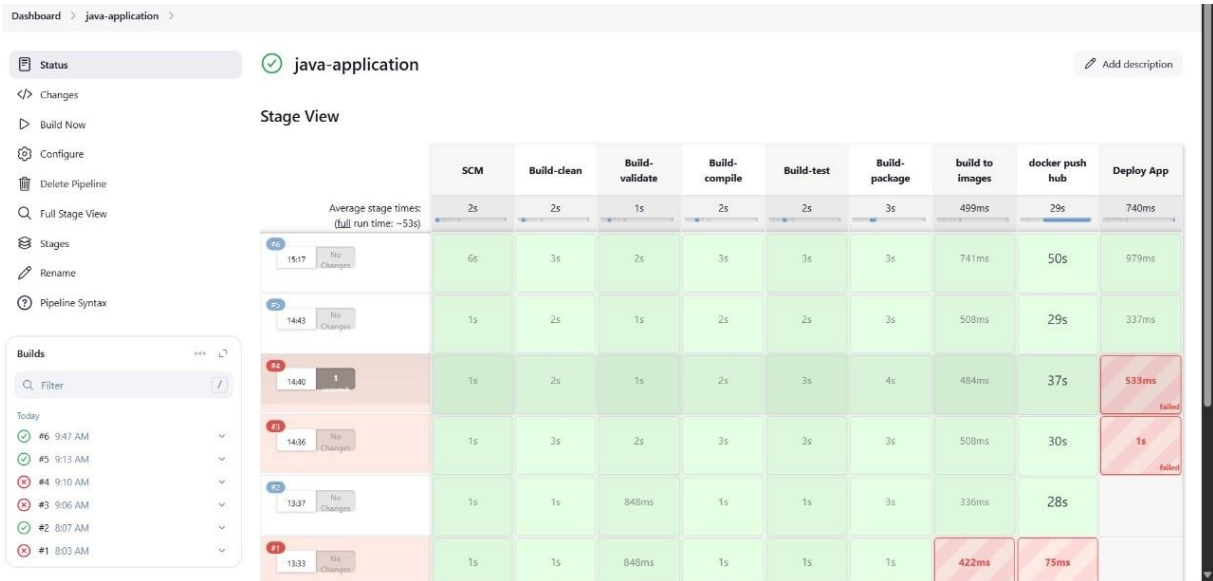
targetPort: 8080

nodePort: 30002

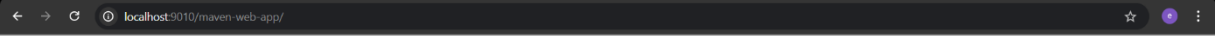
selector:

apptype: web-backend

Output:



```
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 ...
Restarting existing docker container for "minikube" ...
Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
Verifying Kubernetes components...
  * Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: storage-provisioner, default-storageclass
Done! kubectrl is now configured to use "minikube" cluster and "default" namespace by default
ragul@Admin:~$ 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://127.0.0.1:42561           |
+-----+-----+-----+-----+
Opening service default/my-service in default browser...
http://127.0.0.1:42561
! 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.
ragul@Admin:~$ 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
t:1px;background-color:#525D76;border:none;}</style></head><body><h1>HTTP Status 404 - Not Found</h1><hr class="line" /><p><b>Type</b> Status Report</p><p><b>
Description</b> The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.</p><hr clas
s="line" /><h3>Apache Tomcat/9.0.102</h3></body></html>ragul@Admin:~$ 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
t:1px;background-color:#525D76;border:none;}</style></head><body><h1>HTTP Status 404 - Not Found</h1><hr class="line" /><p><b>Type</b> Status Report</p><p><b>
Description</b> The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.</p><hr clas
s="line" /><h3>Apache Tomcat/9.0.102</h3></body></html>ragul@Admin:~$ curl http://192.168.49.2:30002
kubectrl get pod^C
ragul@Admin:~$ ^C
ragul@Admin:~$ curl http://192.168.49.2:30002/maven-web-app/
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
ragul@Admin:~$ kubectrl port-forward svc/my-service 9010:9000
Forwarding from 127.0.0.1:9010 -> 8080
Forwarding from [::1]:9010 -> 8080
Handling connection for 9010
Handling connection for 9010
```



Hello World!