

DAY 5

```
roshni@LAPTOP-1VNKAF61: ~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

roshni@LAPTOP-1VNKAF61:~$ cd ~/.kube
roshni@LAPTOP-1VNKAF61:~/.kube$ sudo vi config
[sudo] password for roshni:
roshni@LAPTOP-1VNKAF61:~/.kube$ sudo vi config
[sudo] password for roshni:
roshni@LAPTOP-1VNKAF61:~/.kube$ sudo vi config

[1]+  Stopped                  sudo vi config
roshni@LAPTOP-1VNKAF61:~/.kube$ sudo vi config
roshni@LAPTOP-1VNKAF61:~/.kube$ kubectl get node
NAME          STATUS    ROLES    AGE   VERSION
minikube      Ready    control-plane   2d    v1.32.0
roshni@LAPTOP-1VNKAF61:~/.kube$
```

```
roshni@LAPTOP-1VNKAF61:~$ cat ~/.kube/config
apiVersion: v1
clusters:
- cluster:
  certificate-authority: /home/roshni/.minikube/ca.crt
  extensions:
  - extension:
    last-update: Fri, 21 Mar 2025 07:56:06 UTC
    provider: minikube.sigs.k8s.io
    version: v1.35.0
    name: cluster_info
  server: https://127.0.0.1:32769
  name: minikube
contexts:
- context:
  cluster: minikube
  extensions:
  - extension:
    last-update: Fri, 21 Mar 2025 07:56:06 UTC
    provider: minikube.sigs.k8s.io
    version: v1.35.0
    name: context_info
  namespace: default
  user: minikube
  name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
  client-certificate: /home/roshni/.minikube/profiles/minikube/client.crt
  client-key: /home/roshni/.minikube/profiles/minikube/client.key
```

Dashboard > java_application > Configuration

Configure

General

Triggers

Pipeline

Advanced

```
10 stage('build') {
11   steps {
12     bat "mvn clean"
13     bat "mvn install"
14   }
15 }
16 stage('build to images') {
17   steps {
18     script{
19       bat 'docker build -t roshni2108/devops .'
20     }
21   }
22 }
23 stage('push to hub') {
24   steps {
25     script{
26       withDockerRegistry(credentialsId: 'Docker_cred', url: 'https://index.docker.io/v1/') {
27         bat 'docker push roshni2108/devops'
28       }
29     }
30   }
31 }
32 stage('Deploy App') {
33   steps {
34     script{
35       withKubeConfig(caCertificate: '', clusterName: 'minikube', contextName: 'minikube', credentialsId: 'KubeConfig_cred') {
36         sh 'kubectl apply -f deployment.yml --validate=false'
37       }
38     }
39   }
40 }
41 }
```

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 model!

Jenkins URL ?

Save

Dashboard > java_application >

Jenkins

ROSHNI MAGESH

log out

Status

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Stages

Rename

Pipeline Syntax

java_application

Add description

Stage View

Average stage times:
(full run time: ~1min 8s)

	SCM	build	build to images	push to hub
14:41 No Changes	2s	5s	847ms	36s
14:08 No Changes	2s	6s	44s	37s

```
roshni@LAPTOP-1VNKAF61: ~$ sudo vi config
[sudo] password for roshni:
roshni@LAPTOP-1VNKAF61: ~$ cd ..
roshni@LAPTOP-1VNKAF61: ~$ 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-1VNKAF61: ~$ 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" /><p><b>Type</b> Status Report</p><p><b>Description</b> The origin server did not find a current repre
sentation for the target resource or is not willing to disclose that one exists.</p><hr class="line" /><h3>Apache Tomcat/9.0.102</h3>
</body></html>roshni@LAPTOP-1VNKAF61: ~$
roshni@LAPTOP-1VNKAF61: ~$ 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" /><p><b>Type</b> Status Report</p><p><b>Description</b> The origin server did not find a current repre
sentation for the target resource or is not willing to disclose that one exists.</p><hr class="line" /><h3>Apache Tomcat/9.0.102</h3>
</body></html>roshni@LAPTOP-1VNKAF61: ~$
```

sudo visudo

(below root => jenkins ALL=(ALL) NOPASSWD: ALL)

sudo systemctl restart ssh.service

sudo systemctl restart sshd.service

sudo apt update

sudo apt install openssh-server

sudo systemctl restart ssh.service

sudo systemctl status ssh

sudo systemctl daemon-reload

cd ~/.kube

sudo vi config

CODE

pipeline {

agent any

```
stages {
  stage('scm') {
    steps {
      git branch: "
    }
  }

  stage('builb-clean') {
    steps {
      sh "mvn clean"
    }
  }

  stage('build-validate') {
    steps {
      sh "mvn validate"
    }
  }

  stage('build-com') {
    steps {
      sh "mvn compile"
    }
  }

  stage('build-test') {
    steps {
      sh "mvn test"
    }
  }
}
```

```

stage('build-install') {
    steps {
        sh "mvn package"
    }
}

stage('build to images') {
    steps {
        script{
            sh 'docker build -t roshni2108/devops .'
        }
    }
}

stage('push to hub') {
    steps {
        script{
            withDockerRegistry(credentialsId: 'Docker_cred', url: 'https://index.docker.io/v1/') {
                sh 'docker push roshni2108/devops'
            }
        }
    }
}

stage('Deploy App') {
    steps {
        withKubeConfig(caCertificate: '', clusterName: 'minikube', contextName: 'minikube',
credentialsId: 'mukubeconfig_011', namespace: '', restrictKubeConfigAccess: false, serverUrl:
'https://192.168.49.2:8443') {
            sh 'kubectl apply -f deployment.yml --validate=false'
        }
    }
}

```

```
    }  
    stage('Test') {  
        steps {  
            withKubeConfig(caCertificate: '', clusterName: 'minikube', contextName: 'minikube',  
credentialsId: 'mukubeconfig_011', namespace: '', restrictKubeConfigAccess: false, serverUrl:  
'https://192.168.49.2:8443') {  
  
                sh 'minikube service my-service --url | xargs curl'  
            }  
        }  
    }  
}
```