DAY 5 Tasks – minikube deployment, Terraform

Minikube Deployment:

Step 1: Config file updation

Sudo visudo: update this, jenkins ALL=(ALL) NOPASSWD: ALL

Data updation in config file:

- sudo cat /home/ragu_ubuntu/.minikube/ca.crt
- sudo cat /home/ragu_ubuntu/.minikube/ca.crt | base64 -w 0; echo
- sudo cat /home/ragu_ubuntu/.minikube/profiles/minikube/client.crt | base64 -w 0; echo
- sudo cat /home/ragu_ubuntu/.minikube/profiles/minikube/client.key | base64 -w 0; echo

step 2: Pipeline script for minikube deployment

```
pipeline {

  agent any

  tools {maven "maven"}

  stages {

    stage('SCM') {

      steps {

        git branch: 'main', url: 'https://github.com/Ragu162004/web-app.git'
```
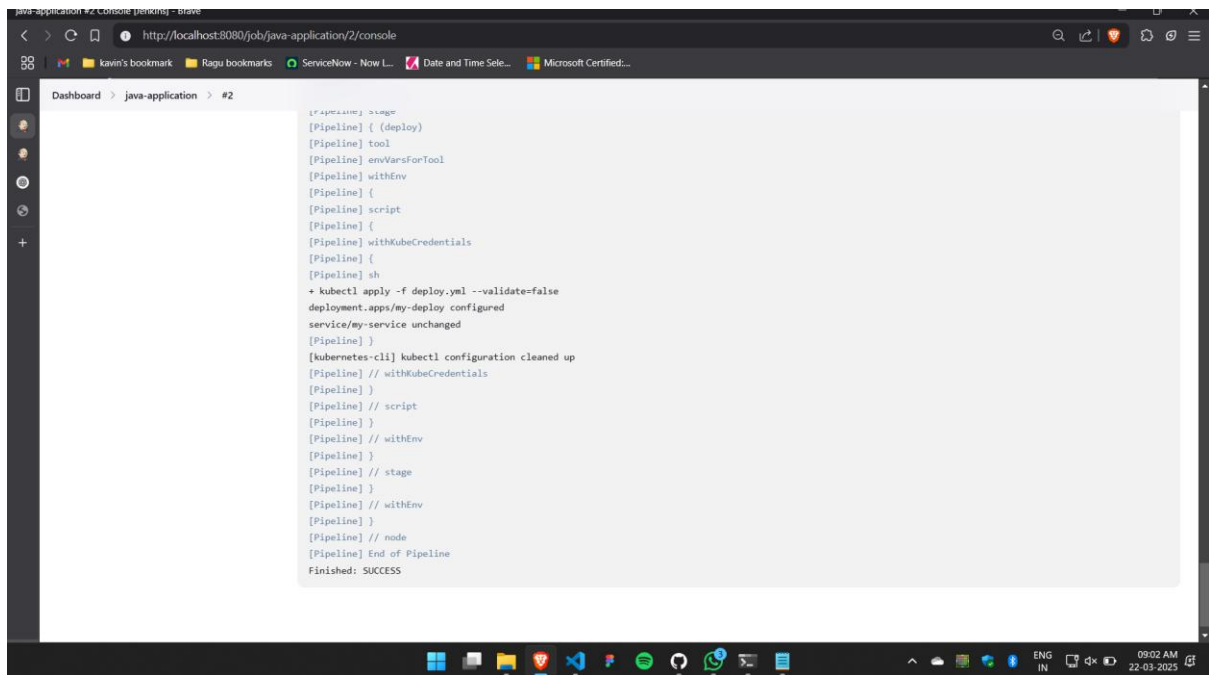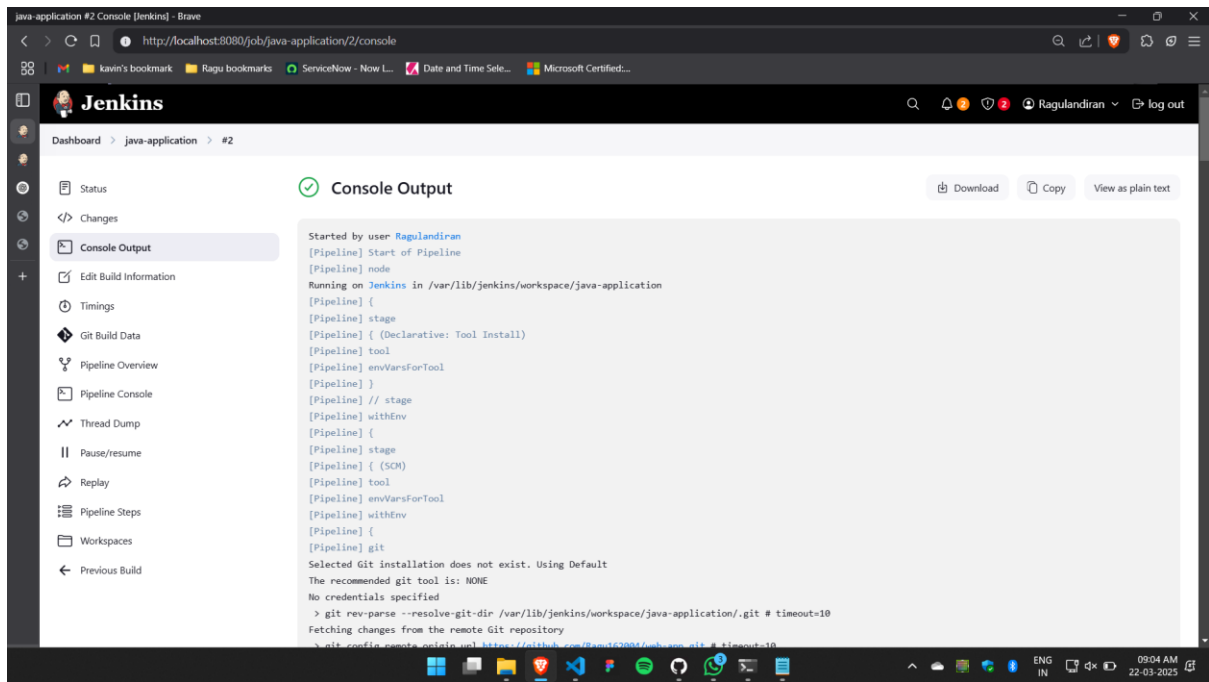
```
        }
    }
    stage('Build-clean') {
        steps {
            sh 'mvn clean'
        }
    }
    stage('Build-validate') {
        steps {
            sh 'mvn validate'
        }
    }
    stage('Build-complie') {
        steps {
            sh 'mvn compile'
        }
    }
    stage('Build-package') {
        steps {
            sh 'mvn package'
        }
    }
    stage('build to images') {
        steps {
            script {
                sh 'docker build -t ragu162004/webapp1 .'
            }
        }
    }
    stage('push to hub') {
        steps {
```
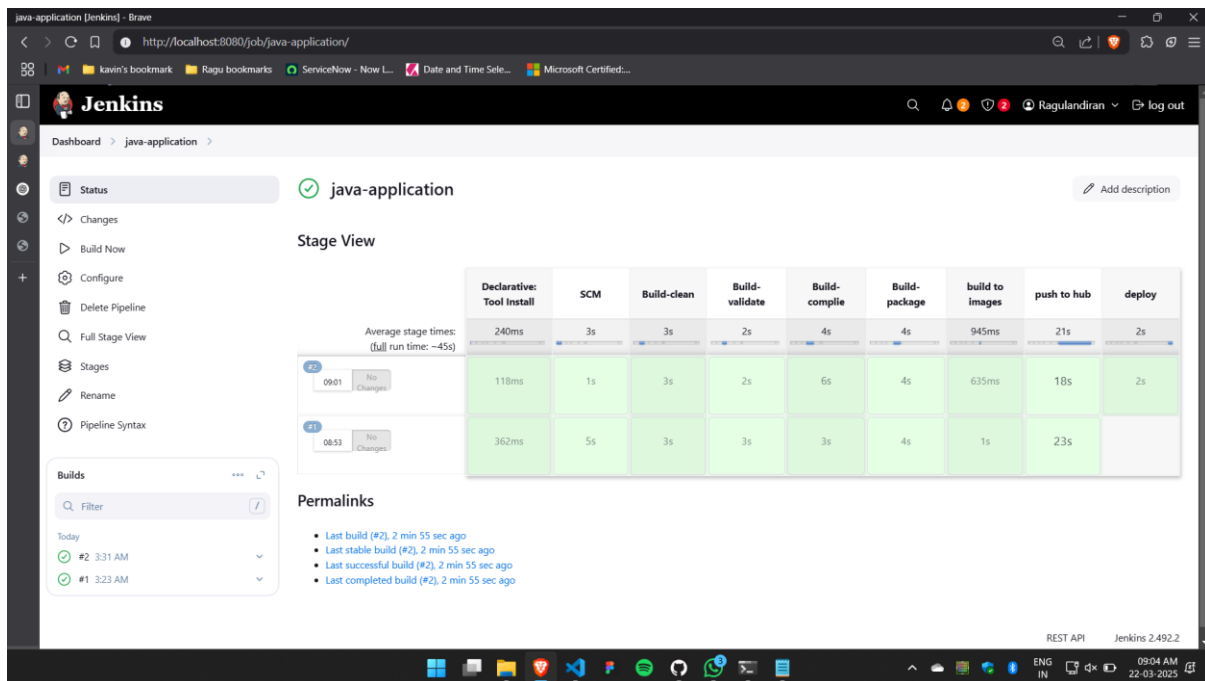
```
        script {

            withDockerRegistry(credentialsId: 'docker_cred', toolName: 'docker', url:
'https://index.docker.io/v1/') {

                sh 'docker push ragu162004/webapp1'

            }

          }

        }

      }

    stage('deploy') {

      steps {

        script {

            withKubeCredentials(kubectlCredentials: [[caCertificate: '', clusterName: 'minikube',
contextName: 'minikube', credentialsId: 'kube_cred', namespace: '', serverUrl:
'https://192.168.39.226:8443']]) {

                sh 'kubectl apply -f deploy.yml --validate=false'

            }

          }

        }

      }

    }

  }

}
```

http://localhost:8080/job/java-application/2/console

kavin's bookmark   Ragu bookmarks   ServiceNow - Now L...   Date and Time Sele...   Microsoft Certified:...

Jenkins

Dashboard > java-application > #2

- Status
- Changes
- Console Output
- Edit Build Information
- Timings
- Git Build Data
- Pipeline Overview
- Pipeline Console
- Thread Dump
- Pause/resume
- Replay
- Pipeline Steps
- Workspaces
- Previous Build

✅ Console Output

[Download] [Copy] [View as plain text]

```
Started by user Ragulandiran
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/java-application
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (SCM)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] git
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/java-application/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote origin url https://github.com/Ragu162004/web-app.git # timeout=10
```

ENG
IN   09:04 AM
22-03-2025

---

http://localhost:8080/job/java-application/2/console

kavin's bookmark   Ragu bookmarks   ServiceNow - Now L...   Date and Time Sele...   Microsoft Certified:...

Dashboard > java-application > #2

```
[Pipeline] stage
[Pipeline] { (deploy)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[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] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

ENG
IN   09:02 AM
22-03-2025

Stage View of Pipeline:

- minikube service my-service
- curl http://192.168.49.2:30002/my-web/

Terraform:

```
#terraform init
#terraform validate
#terraform plan
#terraform apply
#terraform destroy
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.92.0"
    }
  }
}


provider "aws" {
    region = "us-east-1"
}


resource "aws_vpc" "myvpc" {
 cidr_block      = "10.0.0.0/16"

  tags = {
   Name = "demovpc"
  }
}


resource "aws_subnet" "pubsub" {
  vpc_id    = aws_vpc.myvpc.id
  cidr_block = "10.0.1.0/24"
  availability_zone = "us-east-1a"
```

```
  tags = {
    Name = "sn1"
  }
}
resource "aws_subnet" "pub_sub" {
  vpc_id    = aws_vpc.myvpc.id
  cidr_block = "10.0.2.0/24"
  availability_zone = "us-east-1a"
  tags = {
    Name = "sn1"
  }
}
resource "aws_subnet" "prisub" {
  vpc_id    = aws_vpc.myvpc.id
  cidr_block = "10.0.3.0/24"
  availability_zone = "us-east-1a"
  tags = {
    Name = "sn1"
  }
}
resource "aws_subnet" "pri_sub" {
  vpc_id    = aws_vpc.myvpc.id
  cidr_block = "10.0.4.0/24"
  availability_zone = "us-east-1a"

  tags = {
    Name = "sn1"
  }
}
resource "aws_internet_gateway" "tfigw" {
  vpc_id = aws_vpc.myvpc.id
  tags = {
    Name = "tfigw"
```

```
  }
}
resource "aws_route_table" "tfpubrt" {
 vpc_id = aws_vpc.myvpc.id
 route {
  cidr_block = "0.0.0.0/0"
  gateway_id = aws_internet_gateway.tfigw.id
 }
 tags = {
  Name = "tfpublicroute"
 }
}
resource "aws_route_table_association" "pubsn1" {
 subnet_id     = aws_subnet.pubsub.id
 route_table_id = aws_route_table.tfpubrt.id
}
resource "aws_route_table_association" "pubsn2" {
 subnet_id     = aws_subnet.pub_sub.id
 route_table_id = aws_route_table.tfpubrt.id
}

resource "aws_eip" "tfeip" {
 domain  = "vpc"
}

resource "aws_nat_gateway" "tfnat" {
 allocation_id = aws_eip.tfeip.id
 subnet_id     = aws_subnet.pub_sub.id
 tags = {
  Name = "gw NAT"
 }
}
```

```
resource "aws_route_table" "tfprirt" {
 vpc_id = aws_vpc.myvpc.id
 route {
  cidr_block = "0.0.0.0/0"
  gateway_id = aws_nat_gateway.tfnat.id
 }
 tags = {
  Name = "tfprivateroute"
 }
}
resource "aws_route_table_association" "prisn3" {
 subnet_id     = aws_subnet.prisub.id
 route_table_id = aws_route_table.tfprirt.id
}
resource "aws_route_table_association" "prisn4" {
 subnet_id     = aws_subnet.pri_sub.id
 route_table_id = aws_route_table.tfprirt.id
}

resource "aws_security_group" "allow_tfsg" {
 name       = "allow_tfsg"
 description = "Allow TLS inbound traffic"
 vpc_id      = aws_vpc.myvpc.id
 ingress {
  description    = "HTTPS "
  from_port      = 443
  to_port       = 443
  protocol       = "tcp"
  cidr_blocks     = ["0.0.0.0/0"]
 }
 ingress {
  description     = "HTTP "
  from_port      = 80
```

```
      to_port       = 80

      protocol      = "tcp"

      cidr_blocks    = ["0.0.0.0/0"]

    }

    ingress {

     description    = "SSH"

     from_port      = 22

     to_port       = 22

     protocol      = "tcp"

     cidr_blocks    = ["0.0.0.0/0"]

    }

    egress {

     from_port      = 0

     to_port       = 0

     protocol      = "-1"

     cidr_blocks    = ["0.0.0.0/0"]

    }

    tags = {

     Name = "TfsecurityGroup"

    }

}

resource "aws_instance" "pub_ins" {

   ami              = "ami-0fc5d935ebf8bc3bc"

   instance_type        = "t2.micro"

   subnet_id          = aws_subnet.pub_sub.id

   vpc_security_group_ids    = [aws_security_group.allow_tfsg.id]

   key_name          = ""

   associate_public_ip_address  = "true"

}
```