

Day 5:

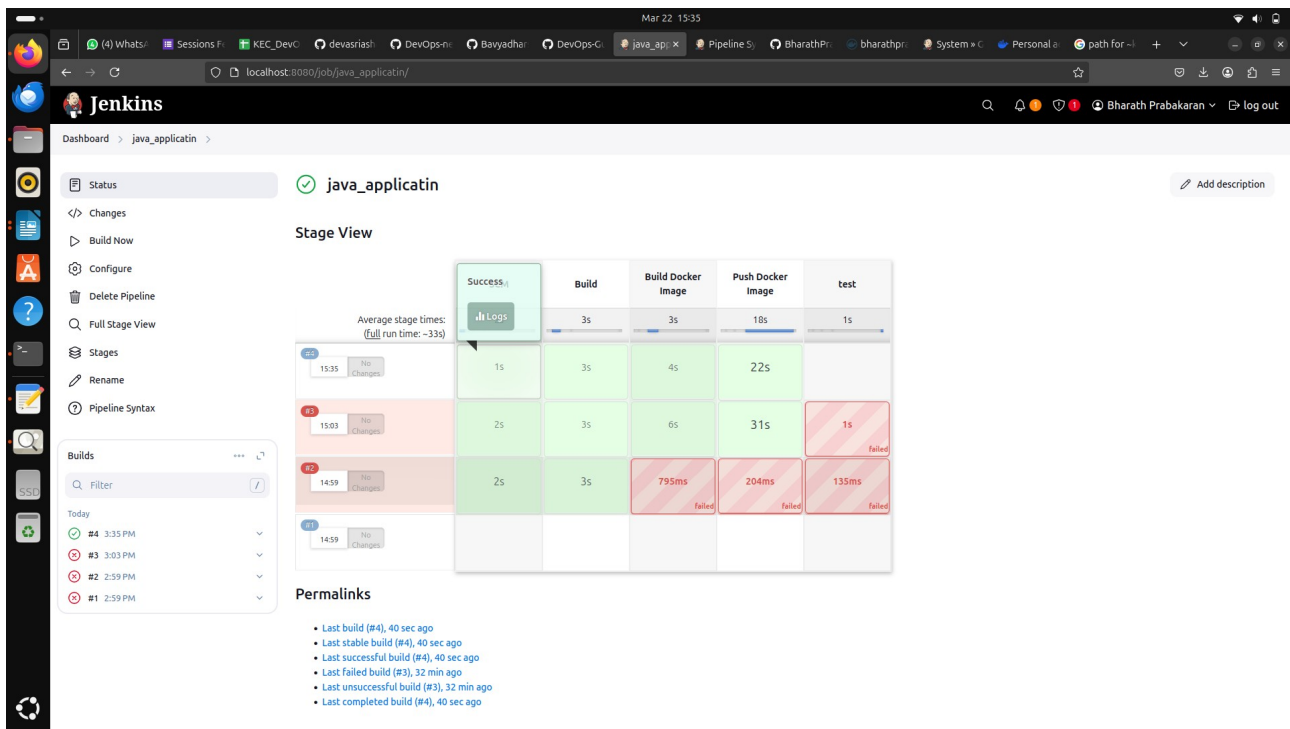
```
Mar 22 14:38
bharath-prabakaran@Dot: ~/kube

bharath-prabakaran@Dot:~/kube$ sudo vi config
[sudo] password for bharath-prabakaran:
bharath-prabakaran@Dot:~/kube$ kubectl get node
error: error loading config file "/home/bharath-prabakaran/.kube/config": yaml: line 42: could not find expected ':'
bharath-prabakaran@Dot:~/kube$ sudo vi config
bharath-prabakaran@Dot:~/kube$ kubectl get node
NAME                STATUS    ROLES    AGE   VERSION
minikube             Ready    control-plane   3d2h   v1.32.0
bharath-prabakaran@Dot:~/kube$
```

```
Mar 22 14:38
bharath-prabakaran@Dot: ~/kube

bharath-prabakaran@Dot:~/kube$ come /home/bharath-prabakaran/.minikube
Command 'come' not found, did you mean:
  command 'code' from snap code (ddc367ed)
  command 'comp' from deb mailutils-nh (1:3.16-1build1)
  command 'comp' from deb mmh (0.4-6)
  command 'comp' from deb mmh (1.8-1)
  command 'comp' from deb coreutils (0.4-2ubuntu2)
  command 'comet' from deb crimson (0.5.2-1build2)
  command 'cme' from deb cme (1.039-1)
See 'snap info <snapname>' for additional versions.

bharath-prabakaran@Dot:~/kube$ cat /home/bharath-prabakaran/.minikube/ca.crt | base64 -w 0; echo
LS0tLS1CRUdJTTlBRVJUSUJzQ0FURSB0tLS0tCk1JSURCaKN0QWU2ZGF3SUJzB20lCQVRBTKJna3Foa2lHOXcwQkRrc0ZBREFTVjN0d0VRWURWUWFERXdwGCFXNNAKYTNWApVtKJNQjRyRFRJMU1ETXhpREYTRnME9Gbi1hEVE0xTURNeE56QTJ0
RGcwT0Zvd0ZURVRNMQVHQTFVRQ0BeE1YLSd0dWdFZbVZUVRDQ0FTSXdEUVlKS29aSWZhY3R0bGVzY295bWVudC9FBRGdnRVBIBRENDQVZvQ2dnRUJzBTkFhckNadllkT2d5OTd0MTNzOFBNSE50N2xkUUFJXW80d2VnMnZFINDRMavhMe1QzWkV3NUT3ZUFi
Zy8rU30t1Zg21UKTVDM3R3cFVtUWpMDNENRFRMnYRkx3UGVEKzN2NC9BM19sMWI0d3JhN1FkdY9WR09kc1F0WkVkbGZkbzR6U0p3Z21lBgtzV0J0ZlptStlUnd0SVdndVGG0Y2Nlay9CQjVvE9XSMiYRE1rYjltGdGLQVFP2bnduOGV4NjQy
eG9FcncJ1NEddvd2cXQFDR2HOTzFha0w2Qm1mU3lHQmLLRHBt0WpNM29FL3doUmXPeNRUOUF40GJ1bVZjTW10UkYmYkOFdXZXBGSVPSMV3enRaODVwUUIv3RQBU9hOUhTNR0azFY3JlLXkLrZHR0W1ZnV08ehIV2JwdnY8aUJGLWpaQ8IT
aV0ZaWdrZ2R0VnMBdFBNQ0F3RUBFYU50TUy4d0RmUWUJBOQVFLB2BUURB20trTUIW0EXVNRKUVFKK1CUddQ23NHQVFRKJ3TUCZ2dyQndFRKJRYBRBVEFRQndQVkhSTUJBZJhFQlRREFRSC9NQJhQTFVZERnUvckKQJRbEXRQzB852p2
TVLHNU10MhNUKZU2MDR3SGZ0USCZ2xvG6R213HE3BUXNQCUFQ08FRUFZV19FUDNLNggH2DnWkqT0lFVJXcE820FVOCVENENEN9wX2cAXRb0NP02V5GgrUzUyTnSZ0Z60d1hmkK8C50811CjBkM1yNUd0Sj0hR1p0k1T
OXLUK244Mh1v10y1Bx0XKxHVC1BSRE1LWTRWHTHksSURvSv1vOULnR3EKcB3DVEKwXkdTneKz8ZUFEVsw1ma0FKTUXLFNFUTkNAyFFMUJ3VWk3d2Qv01RkKcSvR1Y4M1p1SmwbnpX8p0HfC55FV1dXLB2FTUzFkcTpwMhVzXAwQvPp
dFRaV2JTDGdWYnFUnkZYmWvS1dnB2ZTbzdzFQWpneSMNj3CC1900FVGbXhSv1FYTFovVMGtGZx03TnVla1hKYndYMXdBaHLSaJUrWEJnK9xeFp0aJ3Pd5tOULB0eVFnXRaTkckM2p5dVrJN1tLkH2JUUC9P0tLS0tLUVORCBDRVJUSUJzQ0FUR
RS0tLS0tCg==
bharath-prabakaran@Dot:~/kube$ cat /home/bharath-prabakaran/.minikube/profiles/minikube/client.crt | base64 -w 0; echo
LS0tLS1CRUdJTTlBRVJUSUJzQ0FURSB0tLS0tCk1JSURJVEN0QWdtZGF3SUJzB20lCQWpBTKJna3Foa2lHOXcwQkRrc0ZBREFTVjN0d0VRWURWUWFERXdwGCFXNNAKYTNWApVtKJNQjRyRFRJMU1ETXhpREYTRnME9Gbi1hEVE0xTURNeE9EQTJ0
RGcwT0Zvd0ZURVhVHQVHQTFVRQ0BeE1PYzNsenRHHVnRPbTFoYzN5bQNuTXhGakFVQndQVkJBTVEVZzFwYm1ScnRXSmxMMFZ6WlHjD2dnRwLWl0BCKNTcUdTSMWZFFFKFRVUFBNELCRHdbd2dnRU1Bb0lCQVFB3d3SenVZMDArVkdKd29LcVFs
QlI2QjVUODRrcy8KwJjG5FUwV03jQzIzN11kUy9KvGZau1FvZ25aRkXhwnR10FVucVdxQ1RzanJZTFVSTRZTEPYUvBHRFRId0ZUagpnKzhpMX01R2FlWkNnSTVKUEZLN004NCtyVXk2bTBDZE5R3jQzVGKMM1uenFTbFRnMkt4T2JIOXR1cctG
MjkdCis3RUXLZjF0aHxellVNFZIU11lVgK5Nvp6T2JGdkpKkUaNUVZKUVZVTHJ3UntHUmRlCvBkZVA0QkpwckFFVECUHpyd1cwcLRH0XEcR2U3BljThpJNUZTmo4cjdfY3VYafD0aGMSY3pW3fS2ZVCRDhFSVRlMcnlBgo2RGn6Tmo0d1FS
Qj1PmM3MR80V0BuzK5LZzBLVGI2U0IR0s1MEZwmJ0NDQ2NURMT2hVdLJBu5KeJuzckFnTnU3BQddqCL1EQmVNRHQTFVZER3RU1vd1FFQxdJ3Rm9EQMRZ05W5FNVRUZqQVVC22dyQndFRk3RY0RBUVL353dZQk3RVUgKQxdJd0RBURUWJBU
QVFL0JBSXDBREFQnd0VkhT1UVRHFXZ03RbEXRQzB852p2TVLHNU10MhNUKZU2MDR3SAPmakFQndrcwHraUc5dz8QVfZkFBT0NBuUvBSU0vbmJHeGtKcS9zdW0xYz24aTBNVdoTQnYRDZnVRSUUXcK1vN080VnNMhXcDdwVthXk83
SFZ2bZVnK2LaNVQM3F3c0duUGVjkdIdQy6SDF2Ump4ZEN3QUHMDhGHEKc01IR1CdeEpQ203ZWRBTdPRHGHUJZBRNtwbm9jBWNtMHV4dQNVRGZyQudHSE14Njr6a0pGZFZJTHJUMmxS2gpmR1dsbWQ4VutqT21R0VUscVEvRXH1LZY2SDhL
c296/vK3W96G/7B0wzP83LMAZ2g5VZ4fmg6GStG77Y1JwQ08aT1cawX
U1J2N2Ize11BU1g2aKc9P0uL50LUVORCBDRVJUSUJzQ0FURSB0tLS0tCg==
bharath-prabakaran@Dot:~/kube$ cat /home/bharath-prabakaran/.minikube/profiles/minikube/client.key
| base64 -w 0; echo
-----BEGIN RSA PRIVATE KEY-----
MIIIEogIBAAQCAQEAqMEC7mNNP1RncKcQkQ0UeU/OJLP2dhr1NFgawtt+2Huv3U
32UJlP2RMambXvFj6lqgk7I62C1PyOGcy0D0xg028BU44Pv1tc+Rm2LHy05Tx
uzPOPq1MuptAnTUH+N023dpp86kpl4G5StH/bXqfhdU3fuxCyn9TYb9c2K0FR0m
G04veCmxbYXUQUFXuFAC68EZKXqj3Xj+ASaawBExjB68FtK0xvavrHu5z3C
8yORVjY/K+XHL4VoYXPT6M5qpYHgQ/BCE29HHP+g3MzuL8EQFTjX0+wwPT53zS
oNCK290DhXludBAM4e0OuQyzoLV0VJSc+dWIDAQAABoh/Ex3NKTQWl/kbvrS0
k+wyZaB2QC9lVsxFOLU2KnpW53801Thr62KH5DBZLXDrAXubzeJXLnu+VuudI1p
pdNpTBX3L3qKtVh0ysndtMwz/NNKv+qX4+0prPFLR/UBENcX3CqzVeB1x6Hl
/c296/vK3W96G/7B0wzP83LMAZ2g5VZ4fmg6GStG77Y1JwQ08aT1cawX
N0B3b5VpCg/0L4d0T0tThUc/c296/c20ZbHr+00936c0aYvCg1/c296-12
```

```

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-1b"

tags = {
  Name = "sn1"
}
}
resource "aws_subnet" "prisub" {
  vpc_id    = aws_vpc.myvpc.id
  cidr_block = "10.0.3.0/24"
  availability_zone = "us-east-1b"

  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-1b"

  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     = "David"
  associate_public_ip_address = "true"
}

resource "aws_instance" "pri_ins" {
  ami          = "ami-0fc5d935ebf8bc3bc"
  instance_type = "t2.micro"
  subnet_id    = aws_subnet.prisub.id
  vpc_security_group_ids = [aws_security_group.allow_tfsg.id]
  key_name     = "David"
}

#terraform init
#terraform validate
#terraform plan
#terraform apply
#terraform destroy

```

1. sudo visudo
2. Add jenkins ALL=(ALL) NOPASSWD: ALL

ssh installtion:

1. sudo systemctl restart ssh.service

2. sudo systemctl restart sshd.service
3. sudo apt update
4. sudo apt install openssh-server
5. sudo systemctl restart ssh
6. sudo systemctl status ssh
7. ls /etc/systemd/system/sshd.service or ls /usr/lib/systemd/system/sshd.service
8. sudo systemctl daemon-reload
9. sudo systemctl status ssh

Deployment:

1. cd ~/.kube
2. ls
3. cat config
4. sudo vi config
5. i
6. -data
7. cat url | base64 -w 0; echo
8. minikube start
9. kubectl get node

Pipeline Script:

```
pipeline {
  agent any

  stages {
    stage('SCM Checkout') {
      steps {
        git branch: 'main', url:
'https://github.com/Bavyadharshini-Rajaganapathy/simple-web-app.git'
      }
    }

    stage('Build') {
      steps {
        sh 'mvn clean'
        sh 'mvn install'
      }
    }

    stage('Build Docker Image') {
      steps {
        script {
          sh 'docker build -t bavyadharshini/simplewebapp .'
        }
      }
    }
  }
}
```

```

    }

    stage('Push to Docker Hub') {
        steps {
            script {
                withDockerRegistry(credentialsId: 'Docker_cred', url:
'https://index.docker.io/v1/') {
                    sh 'docker push bavyadharshini/simplewebapp'
                }
            }
        }
    }

    stage('test') {
        steps {
            withKubeConfig(caCertificate: "", clusterName: 'minikube', contextName:
'minikube', credentialsId: 'minikube_cred', namespace: "", restrictKubeConfigAccess:
false, serverUrl: https://192.168.39.226:8443)
                sh 'kubectl apply -f deployment.yml --validate=false'
            }
        }
    }
}

```