Rahul Siddharth D H – 22CSR158

DEVOPS DAY 5 – Deploy Java_Application to minikube automated with Jenkins via pipeline services

Deploy Java App to Minikube Automated with Jenkins

1. Overview

Automating the deployment of a Java application to Minikube using Jenkins involves building the application, creating a Docker image, pushing it to a container registry, and deploying it to Minikube using Kubernetes manifests.

---

2. Key Concepts

A. Jenkins Pipeline

Jenkins automates the CI/CD process using a declarative pipeline. The pipeline consists of multiple stages such as:

- **SCM Checkout:** Fetches code from a repository (GitHub/GitLab).
- **Build & Test:** Uses Maven (mvn package) to compile and test the Java application.
- **Docker Build & Push:** Builds a Docker image of the application and pushes it to Docker Hub.
- **Deploy to Minikube:** Uses kubectl to apply Kubernetes deployment and service files.

B. Minikube

Minikube is a lightweight Kubernetes cluster for local development and testing. It allows developers to run Kubernetes locally and deploy applications without needing a cloud-based cluster.

Commands to Start Minikube:

sh

CopyEdit

```
minikube start
```

```
kubectl cluster-info
kubectl get nodes
```

C. Docker

Docker is used to package the Java application into a container image, making it portable and easy to deploy across environments.

Dockerfile Example:

dockerfile

CopyEdit

```
FROM openjdk:11
COPY target/webapp.jar /app/webapp.jar
WORKDIR /app
CMD ["java", "-jar", "webapp.jar"]
```

D. Kubernetes Deployment

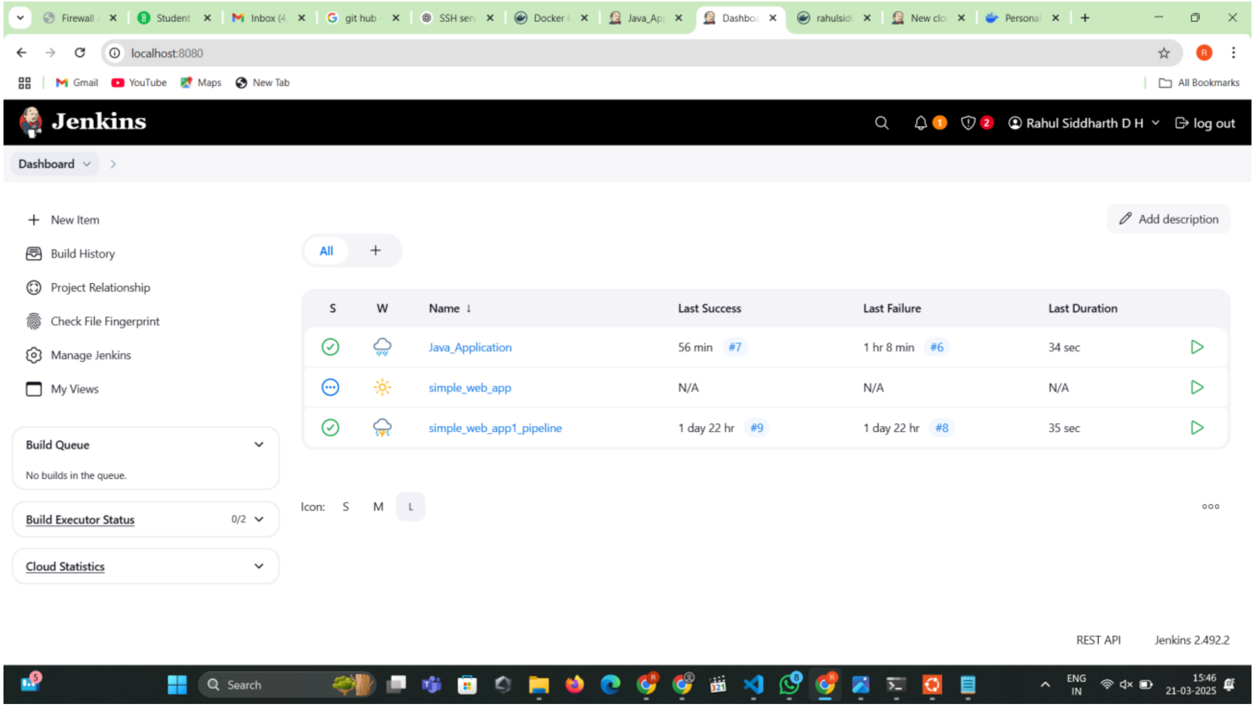
Kubernetes YAML files define how the application should be deployed inside the Minikube cluster.

Deployment YAML Example:

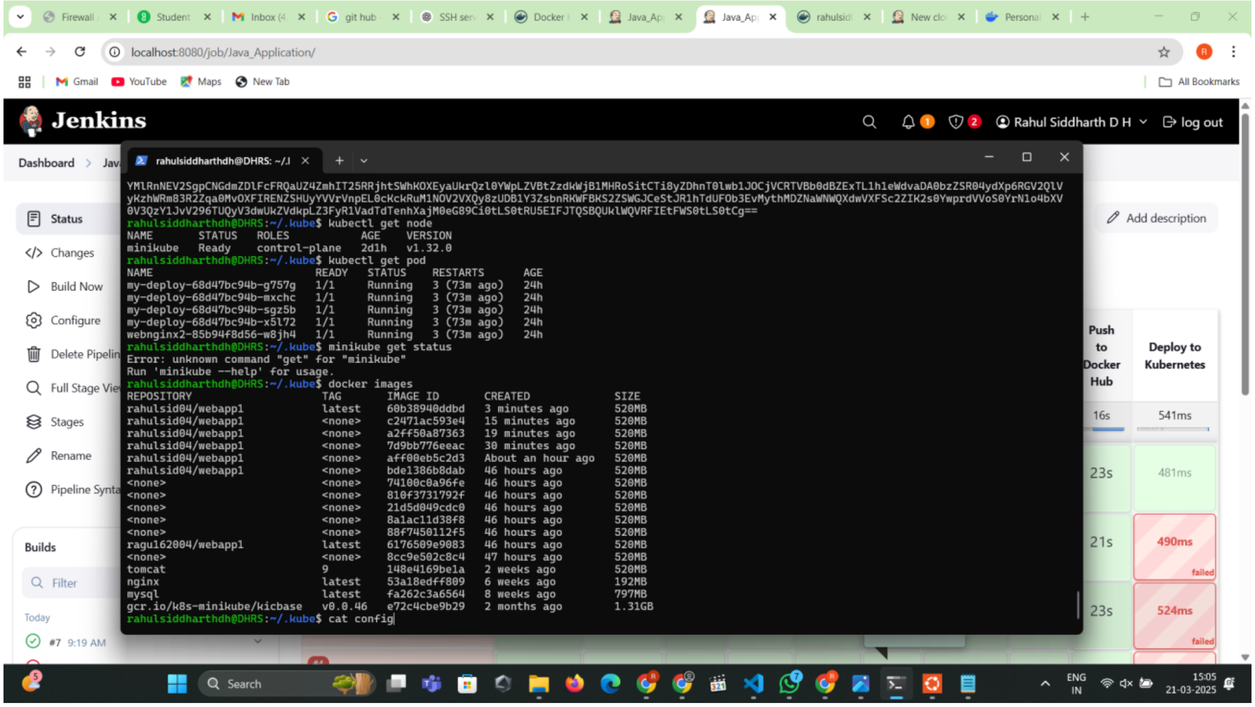
```
pipeline {
    agent any
    tools { maven "maven" }
    stages {
        stage('Clone Repository') {
            steps {
                git branch: 'main', url: 'https://github.com/RahulSiddharth04/webapp.git'
            }
        }
```

```
stage('Clean Project') {
    steps {
        sh 'mvn clean'
    }
}
stage('Validate') {
    steps {
        sh 'mvn validate'
    }
}
stage('Compile') {
    steps {
        sh 'mvn compile'
    }
}
stage('Run Tests') {
    steps {
        sh 'mvn test'
    }
}
stage('Package Application') {
    steps {
        sh 'mvn package'
    }
}
stage('Build Docker Image') {
```

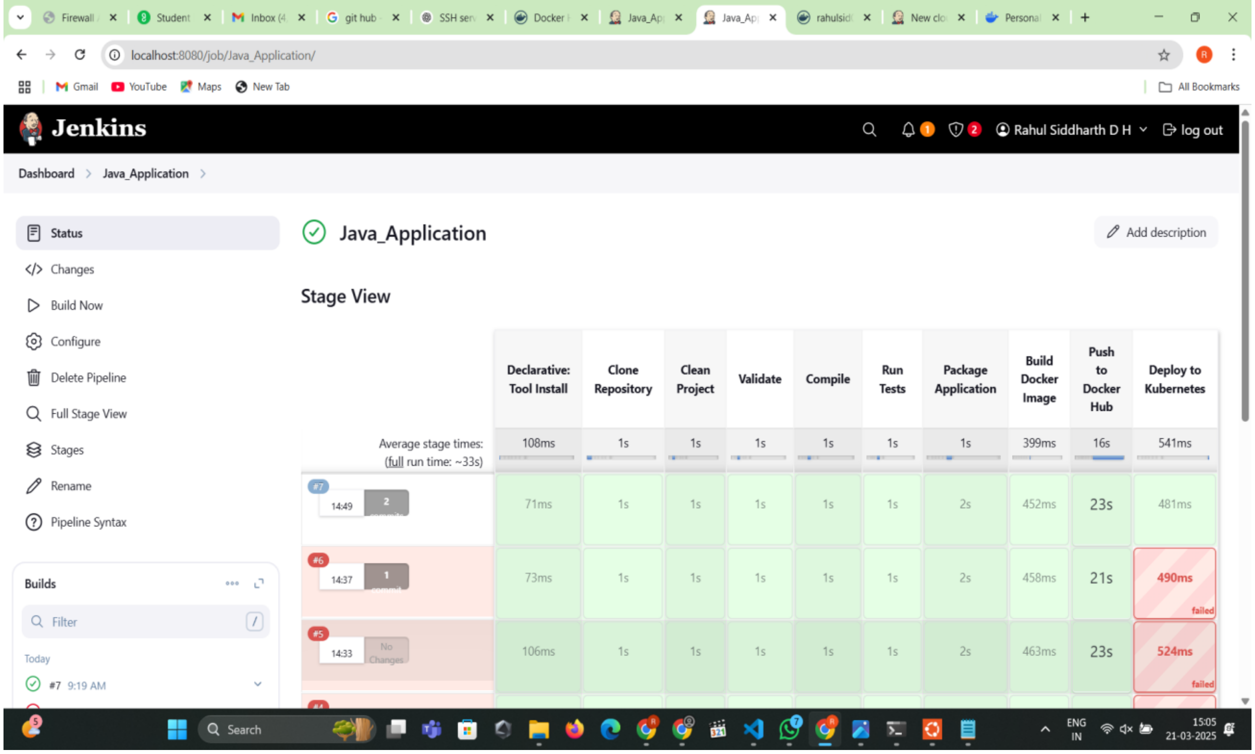
```
            steps {
                script {
                    sh 'docker build -t rahulsid04/webapp1 .'
                }
            }
        }
        stage('Push to Docker Hub') {
            steps {
                script {
                    withDockerRegistry(credentialsId: 'docker_cred', url: 'https://index.docker.io/v1/') {
                        sh 'docker push rahulsid04/webapp1'
                    }
                }
            }
        }
        stage('Deploy to Kubernetes') {
            steps {
                withKubeConfig(
                    caCertificate: '',
                    clusterName: 'minikube',
                    contextName: 'minikube',
                    credentialsId: 'kub_id',
                    namespace: '',
                    restrictKubeConfigAccess: false,
                    serverUrl: 'https://192.168.39.226:8443'
                ) {
                    sh 'kubectl apply -f my-deploy.yml --validate=false'
                }
            }
        }
    }
}
```



S	W	Name ↓	Last Success	Last Failure	Last Duration
✓		Java_Application	56 min #7	1 hr 8 min #6	34 sec
		simple_web_app	N/A	N/A	N/A
✓		simple_web_app1_pipeline	1 day 22 hr #9	1 day 22 hr #8	35 sec



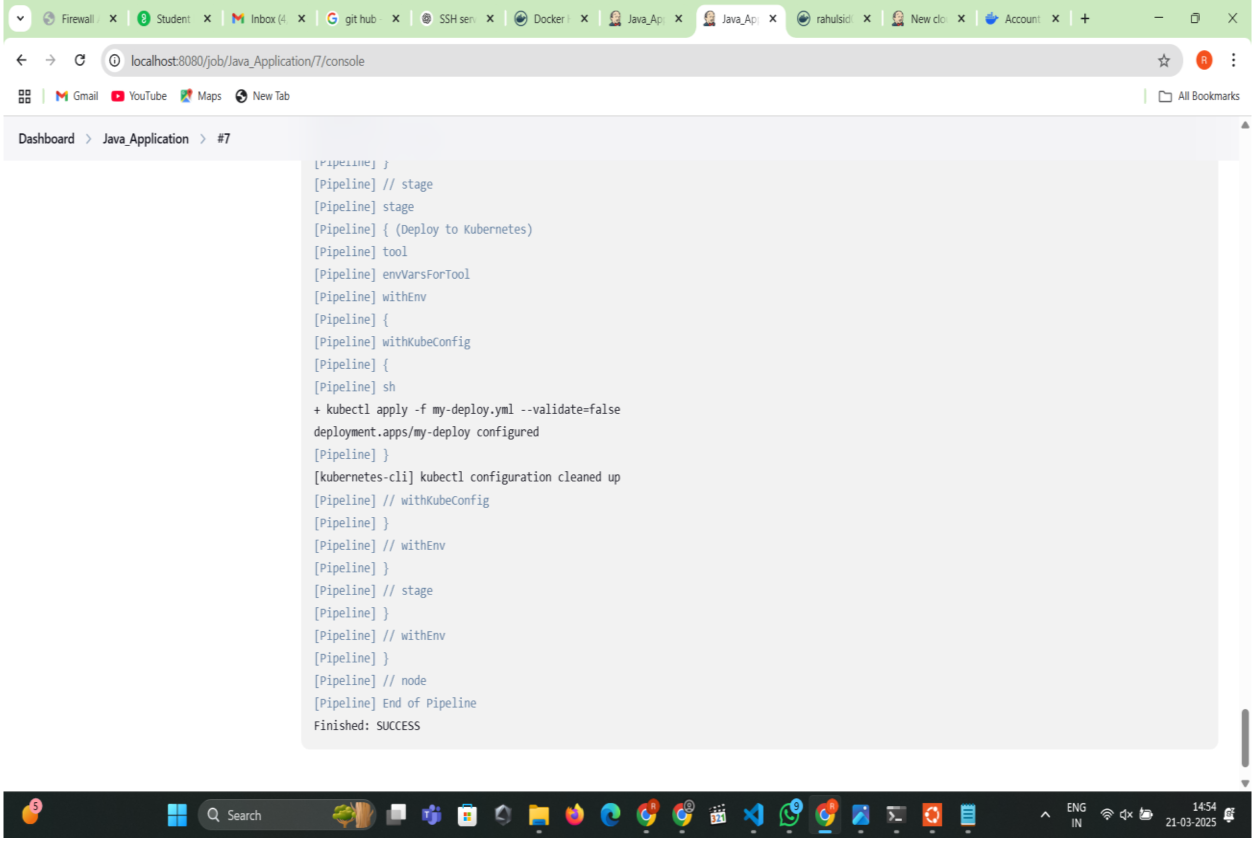
```
YMlRnNEV2SgpCNGdmZDlFcFRQaUZ4ZmhIT25RRjhtSWhKOXEyaUkrQzl0YWpLZVBtZzdkWjB1MHRoSitCTi8yZDhnT0lwb1JOCjVCRTVBb0dBZExTL1hleWdvaDA0bzZSR04ydXp6RGV2QlV
yKzhWRm83R2Zqa0MvOXFIRENZSHUyYVVrVnpEL0cKckRuM1NOV2VXQy8zUDB1Y3ZsbnRKWFBKS2ZSWGJCeStJR1hTdUFOb3EvMythMDZNaWNWQXdwVXFSc2ZIK2s0YwprdVVoS0YrN1o4bXV
0V3QzY1JvV296TUQyV3dwUkZVdkpLZ3FyR1VadTdTenhXajM0eG89Ci0tLS0tRU5EIFJTQSBQUklWQVRFIEtFWS0tLS0tCg==
rahulsiddharthdh@DHRS:~/.kube$ kubectl get node
NAME       STATUS   ROLES           AGE    VERSION
minikube   Ready    control-plane   2d1h   v1.32.0
rahulsiddharthdh@DHRS:~/.kube$ kubectl get pod
NAME                          READY   STATUS    RESTARTS      AGE
my-deploy-68d47bc94b-g757g    1/1     Running   3 (73m ago)   24h
my-deploy-68d47bc94b-mxchc    1/1     Running   3 (73m ago)   24h
my-deploy-68d47bc94b-sgz5b    1/1     Running   3 (73m ago)   24h
my-deploy-68d47bc94b-x5l72    1/1     Running   3 (73m ago)   24h
webnginx2-85b94f8d56-w8jh4    1/1     Running   3 (73m ago)   24h
rahulsiddharthdh@DHRS:~/.kube$ minikube get status
Error: unknown command "get" for "minikube"
Run 'minikube --help' for usage.
rahulsiddharthdh@DHRS:~/.kube$ docker images
REPOSITORY                     TAG       IMAGE ID       CREATED              SIZE
rahulsid04/webapp1             latest    60b38940ddbd   3 minutes ago        520MB
rahulsid04/webapp1             <none>    c2471ac593e4   15 minutes ago       520MB
rahulsid04/webapp1             <none>    a2ff50a87363   19 minutes ago       520MB
rahulsid04/webapp1             <none>    7d9bb776eeac   30 minutes ago       520MB
rahulsid04/webapp1             <none>    aff00eb5c2d3   About an hour ago    520MB
rahulsid04/webapp1             <none>    bde1386b8dab   46 hours ago         520MB
<none>                         <none>    74100c0a96fe   46 hours ago         520MB
<none>                         <none>    810f3731792f   46 hours ago         520MB
<none>                         <none>    21d5d049cdc0   46 hours ago         520MB
<none>                         <none>    8a1ac11d38f8   46 hours ago         520MB
<none>                         <none>    88f7450112f5   46 hours ago         520MB
ragul62004/webapp1             latest    6176509e9083   46 hours ago         520MB
<none>                         <none>    8cc9e502c8c4   47 hours ago         520MB
tomcat                         9         148e4169be1a   2 weeks ago          520MB
nginx                          latest    53a18edff809   6 weeks ago          192MB
mysql                          latest    fa262c3a6564   8 weeks ago          797MB
gcr.io/k8s-minikube/kicbase    v0.0.46   e72c4cbe9b29   2 months ago         1.31GB
rahulsiddharthdh@DHRS:~/.kube$ cat config
```



Java_Application

Stage View

	Declarative: Tool Install	Clone Repository	Clean Project	Validate	Compile	Run Tests	Package Application	Build Docker Image	Push to Docker Hub	Deploy to Kubernetes
Average stage times: (full run time: ~33s)	108ms	1s	1s	1s	1s	1s	1s	399ms	16s	541ms
#7 14:49 2	71ms	1s	1s	1s	1s	1s	2s	452ms	23s	481ms
#6 14:37 1	73ms	1s	1s	1s	1s	1s	2s	458ms	21s	490ms failed
#5 14:33 No Changes	106ms	1s	1s	1s	1s	1s	2s	463ms	23s	524ms failed


localhost:8080/job/Java_Application/7/console
Dashboard > Java_Application > #7

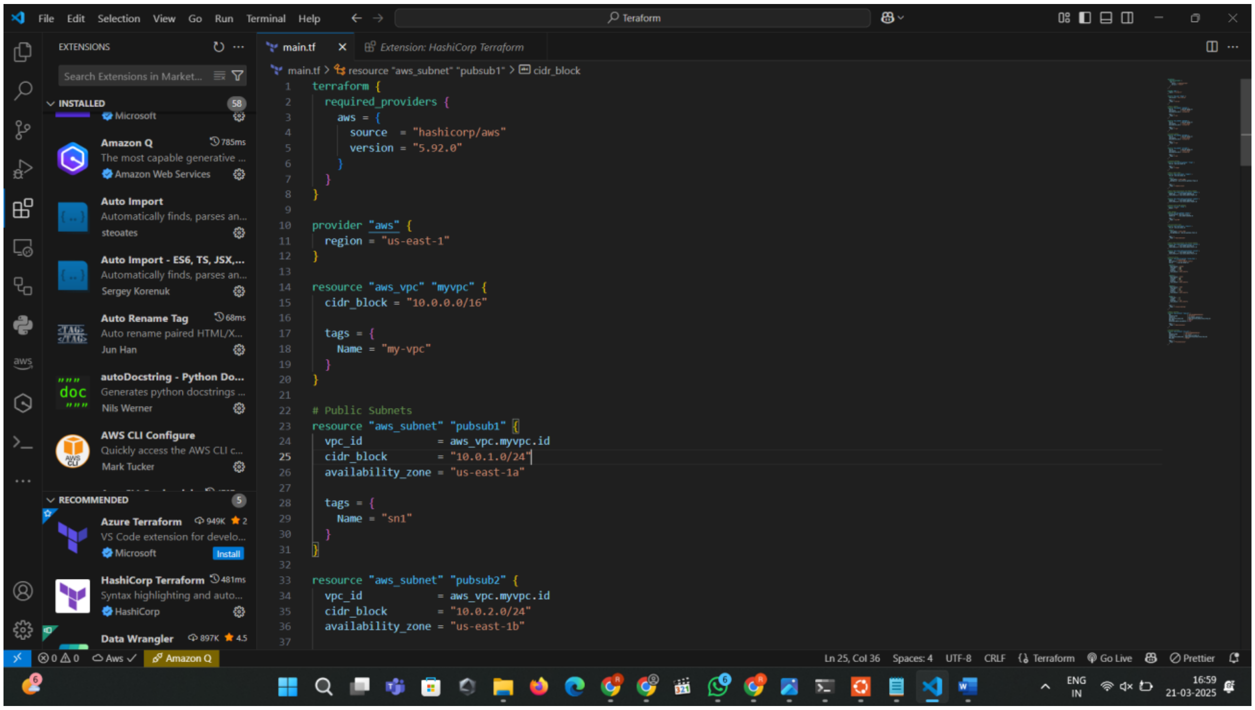
```
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy to Kubernetes)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withKubeConfig
[Pipeline] {
[Pipeline] sh
+ kubectl apply -f my-deploy.yml --validate=false
deployment.apps/my-deploy configured
[Pipeline] }
[kubernetes-cli] kubectl configuration cleaned up
[Pipeline] // withKubeConfig
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Teraform file

```
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 = "my-vpc"
  }
}

# Public Subnets
resource "aws_subnet" "pubsub1" {
  vpc_id            = aws_vpc.myvpc.id
  cidr_block        = "10.0.1.0/24"
  availability_zone = "us-east-1a"

  tags = {
    Name = "sn1"
  }
}

resource "aws_subnet" "pubsub2" {
  vpc_id            = aws_vpc.myvpc.id
  cidr_block        = "10.0.2.0/24"
  availability_zone = "us-east-1b"

  tags = {
    Name = "sn2"
  }
}

# Private Subnets
resource "aws_subnet" "prisub1" {
  vpc_id            = aws_vpc.myvpc.id
  cidr_block        = "10.0.3.0/24"
  availability_zone = "us-east-1a"

  tags = {
    Name = "sn3"
  }
```

```
}


resource "aws_subnet" "prisub2" {
  vpc_id            = aws_vpc.myvpc.id
  cidr_block        = "10.0.4.0/24"
  availability_zone = "us-east-1b"

  tags = {
    Name = "sn4"
  }
}

# Internet Gateway
resource "aws_internet_gateway" "tfigw" {
  vpc_id = aws_vpc.myvpc.id

  tags = {
    Name = "tfigw"
  }
}

# Public Route Table
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"
  }
}

# Route Table Associations for Public Subnets
resource "aws_route_table_association" "pubsn1" {
  subnet_id      = aws_subnet.pubsub1.id
  route_table_id = aws_route_table.tfpubrt.id
}

resource "aws_route_table_association" "pubsn2" {
  subnet_id      = aws_subnet.pubsub2.id
  route_table_id = aws_route_table.tfpubrt.id
}

# Elastic IP for NAT Gateway
resource "aws_eip" "tfeip" {
  domain = "vpc"
}

# NAT Gateway
```

```
resource "aws_nat_gateway" "tfnat" {
  allocation_id = aws_eip.tfeip.id
  subnet_id     = aws_subnet.pubsub2.id

  tags = {
    Name = "gw NAT"
  }
}

# Private Route Table
resource "aws_route_table" "tfprirt" {
  vpc_id = aws_vpc.myvpc.id

  route {
    cidr_block = "0.0.0.0/0"
    nat_gateway_id = aws_nat_gateway.tfnat.id
  }

  tags = {
    Name = "tfprivateroute"
  }
}

# Route Table Associations for Private Subnets
resource "aws_route_table_association" "prisn3" {
  subnet_id      = aws_subnet.prisub1.id
  route_table_id = aws_route_table.tfprirt.id
}

resource "aws_route_table_association" "prisn4" {
  subnet_id      = aws_subnet.prisub2.id
  route_table_id = aws_route_table.tfprirt.id
}

# Security Group
resource "aws_security_group" "allow_tfsg" {
  name        = "allow_tfsg"
  description = "Allow 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"
  }
}

# Public Instance
resource "aws_instance" "pub_ins" {
  ami                    = "ami-0fc5d935ebf8bc3bc"
  instance_type             = "t2.micro"
  subnet_id                  = aws_subnet.pubsub2.id
  vpc_security_group_ids     = [aws_security_group.allow_tfsg.id]
  key_name                   = "David"
  associate_public_ip_address = true

  tags = {
    Name = "PublicInstance"
  }
}

# Private Instance
resource "aws_instance" "pri_ins" {
  ami                  = "ami-0fc5d935ebf8bc3bc"
  instance_type         = "t2.micro"
  subnet_id            = aws_subnet.prisub1.id
  vpc_security_group_ids = [aws_security_group.allow_tfsg.id]
  key_name             = "David"

  tags = {
    Name = "PrivateInstance"
  }
}
```


File Edit Selection View Go Run Terminal Help
Teraform
EXTENSIONS
Search Extensions in Market...
INSTALLED 58
Amazon Q
The most capable generative ...
Amazon Web Services
Auto Import
Automatically finds, parses an...
steoates
Auto Import - ES6, TS, JSX,...
Sergey Korenuk
Auto Rename Tag
Auto rename paired HTML/X...
Jun Han
autoDocstring - Python Do...
Generates python docstrings ...
Nils Werner
AWS CLI Configure
Quickly access the AWS CLI c...
Mark Tucker
RECOMMENDED 5
Azure Terraform
VS Code extension for develo...
Microsoft
Install
HashiCorp Terraform
Syntax highlighting and auto...
HashiCorp
Data Wrangler
main.tf
Extension: HashiCorp Terraform
main.tf > resource "aws_subnet" "pubsub1" > cidr_block
Ln 25, Col 36 Spaces: 4 UTF-8 CRLF Terraform Go Live Prettier
Aws Amazon Q
ENG IN 16:59 21-03-2025

```
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 = "my-vpc"
  }
}

# Public Subnets
resource "aws_subnet" "pubsub1" {
  vpc_id            = aws_vpc.myvpc.id
  cidr_block        = "10.0.1.0/24"
  availability_zone = "us-east-1a"

  tags = {
    Name = "sn1"
  }
}

resource "aws_subnet" "pubsub2" {
  vpc_id            = aws_vpc.myvpc.id
  cidr_block        = "10.0.2.0/24"
  availability_zone = "us-east-1b"

```