

## Minikube Deployment:

### Step 1: Config file updation

```
prasanndh.raaju@DESKTOP-S03TDPN:~$ kubectl get node
NAME        STATUS    ROLES    AGE   VERSION
minikube    Ready    control-plane  47h   v1.32.0
prasanndh.raaju@DESKTOP-S03TDPN:~$ cd ~/.kube
prasanndh.raaju@DESKTOP-S03TDPN:~/.kube$ ls
cache  config
prasanndh.raaju@DESKTOP-S03TDPN:~/.kube$ cat config
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /home/prasanndh.raaju/.minikube/ca.crt
    extensions:
    - extension:
        last-update: Fri, 21 Mar 2025 04:54:53 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 04:54:53 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/prasanndh.raaju/.minikube/profiles/minikube/client.crt
    client-key: /home/prasanndh.raaju/.minikube/profiles/minikube/client.key
prasanndh.raaju@DESKTOP-S03TDPN:~/.kube$ sudo vi config
prasanndh.raaju@DESKTOP-S03TDPN:~/.kube$ kubectl get node
NAME        STATUS    ROLES    AGE   VERSION
minikube    Ready    control-plane  47h   v1.32.0
prasanndh.raaju@DESKTOP-S03TDPN:~/.kube$
```

```
prasanndh.raaju@DESKTOP-S03TDPN:~$ kubectl get node
NAME        STATUS    ROLES    AGE   VERSION
minikube    Ready    control-plane  47h   v1.32.0
prasanndh.raaju@DESKTOP-S03TDPN:~$ cd ~/.kube
prasanndh.raaju@DESKTOP-S03TDPN:~/.kube$ ls
cache  config
prasanndh.raaju@DESKTOP-S03TDPN:~/.kube$ cat config
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: /home/prasanndh.raaju/.minikube/ca.crt
    extensions:
    - extension:
        last-update: Fri, 21 Mar 2025 04:54:53 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 04:54:53 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/prasanndh.raaju/.minikube/profiles/minikube/client.crt
    client-key: /home/prasanndh.raaju/.minikube/profiles/minikube/client.key
prasanndh.raaju@DESKTOP-S03TDPN:~/.kube$
```

```
prasanndh.raaju@DESKTOP-503TDPN:~$ minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

prasanndh.raaju@DESKTOP-503TDPN:~$ kubectl get node
NAME          STATUS    ROLES    AGE   VERSION
minikube      Ready     control-plane  47h   v1.32.0

prasanndh.raaju@DESKTOP-503TDPN:~$ cd ~/.kube
prasanndh.raaju@DESKTOP-503TDPN:~/.kube$ ls
cache  config

prasanndh.raaju@DESKTOP-503TDPN:~/.kube$ cat config
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /home/prasanndh.raaju/.minikube/ca.crt
    extensions:
    - extension:
        last-update: Fri, 21 Mar 2025 04:54:53 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 04:54:53 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/prasanndh.raaju/.minikube/profiles/minikube/client.crt
    client-key: /home/prasanndh.raaju/.minikube/profiles/minikube/client.key
prasanndh.raaju@DESKTOP-503TDPN:~/.kube$

Failed to restart sshd.service: Unit sshd.service not found.
prasanndh.raaju@DESKTOP-503TDPN:~$ sudo systemctl restart sshd.service
Failed to restart sshd.service: Unit sshd.service not found.
prasanndh.raaju@DESKTOP-503TDPN:~$ dpkg -l | grep openssh
ii  openssh-client  1:9.6p1-3ubuntu13.5      amd64        secure shell (SSH) client, for secure access to remote machines
prasanndh.raaju@DESKTOP-503TDPN:~$ sudo apt update
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:2 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:4 http://archive.ubuntu.com/ubuntu noble InRelease
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [671 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [8960 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [6936 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [820 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [922 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [177 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [51.9 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [17.0 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [288 B]
Get:17 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [288 B]
Get:18 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:19 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [13.5 kB]
Get:20 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1041 kB]
Get:21 http://archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [262 kB]
Get:22 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [364 kB]
Get:23 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [25.9 kB]
Get:24 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:25 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:26 http://archive.ubuntu.com/ubuntu noble-backports/main Translation-en [2588 B]
Get:27 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [288 B]
Get:28 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [14.6 kB]
Get:29 http://archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [14.6 kB]
Get:30 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [20.0 kB]
Get:31 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [212 B]
Get:32 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Fetched 4965 kB in 6s (982 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
56 packages can be upgraded. Run 'apt list --upgradable' to see them.
prasanndh.raaju@DESKTOP-503TDPN:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

```
prasanndh.raaju@DESKTOP-S03TDPN:~$
Setting up libwrap0:amd64 (7.6.q-33) ...
Setting up ncurses-term (6.4+20240113-1ubuntu2) ...
Setting up openssh-sftp-server (1:9.6p1-3ubuntu13.8) ...
Setting up openssh-server (1:9.6p1-3ubuntu13.8) ...

Creating config file /etc/ssh/sshd_config with new version
Creating SSH2 RSA key; this may take some time ...
3072 SHA256:qA2Ftd1NfwULwBrd3iF3DPiRp06bWYA2X776wEVU7Y root@DESKTOP-S03TDPN (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:BHD005cFblwdeTTvPXfR+YVDCVcqhnlYL85Bnws6Jji root@DESKTOP-S03TDPN (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:vvNZzTT9yJPUCQktX0ZTq5FjVVKcdpH4xDbAei92MFU root@DESKTOP-S03TDPN (ED25519)
Created symlink /etc/systemd/system/sockets.target.wants/ssh.socket → /usr/lib/systemd/system/ssh.socket.
Created symlink /etc/systemd/system/ssh.service.requires/ssh.socket → /usr/lib/systemd/system/ssh.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.4) ...
prasanndh.raaju@DESKTOP-S03TDPN:~$ sudo systemctl restart ssh
Unknown command verb 'restart ssh', did you mean 'restart'?
prasanndh.raaju@DESKTOP-S03TDPN:~$ sudo systemctl status ssh
Unknown command verb 'status ssh', did you mean 'status'?
prasanndh.raaju@DESKTOP-S03TDPN:~$ ls /etc/systemd/system/sshd.service or ls /usr/lib/systemd/system/sshd.service
ls: cannot access '/etc/systemd/system/sshd.service': No such file or directory
ls: cannot access 'or': No such file or directory
ls: cannot access 'ls': No such file or directory
ls: cannot access '/usr/lib/systemd/system/sshd.service': No such file or directory
prasanndh.raaju@DESKTOP-S03TDPN:~$ sudo systemctl daemon-reload
prasanndh.raaju@DESKTOP-S03TDPN:~$ sudo systemctl restart ssh
prasanndh.raaju@DESKTOP-S03TDPN:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
   Active: active (running) since Fri 2025-03-21 05:44:42 UTC; 13s ago
 TriggeredBy: ● ssh.socket
           Docs: man:sshd(8)
                man:sshd_config(5)
   Process: 16139 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 16140 (sshd)
       Tasks: 1 (Limit: 9337)
      Memory: 1.2M ()
         CGroup: /system.slice/ssh.service
                └─16140 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Mar 21 05:44:42 DESKTOP-S03TDPN systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Mar 21 05:44:42 DESKTOP-S03TDPN sshd[16140]: Server listening on :: port 22.
Mar 21 05:44:42 DESKTOP-S03TDPN systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
prasanndh.raaju@DESKTOP-S03TDPN:~$ sudo systemctl daemon-reload
prasanndh.raaju@DESKTOP-S03TDPN:~$ sudo systemctl restart ssh.service
prasanndh.raaju@DESKTOP-S03TDPN:~$
```

```
prasanndh.raaju@DESKTOP-S03TDPN:~$
E0321 04:54:21.887491 1622 memcache.go:265] "Unhandled Error" err="couldn't get current server API group list: Get \"https://127.0.0.1:32769/api?timeout=32s\": dial tcp 127.0.0.1:32769: connect: connection refused"
The connection to the server 127.0.0.1:32769 was refused - did you specify the right host or port?
prasanndh.raaju@DESKTOP-S03TDPN:~$ minikube start
🐳 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: default-storageclass, storage-provisioner
🎉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
prasanndh.raaju@DESKTOP-S03TDPN:~$ kubectl get pod
No resources found in default namespace.
prasanndh.raaju@DESKTOP-S03TDPN:~$ sudo nano deploy.yml
[sudo] password for prasanndh.raaju:
prasanndh.raaju@DESKTOP-S03TDPN:~$ kubectl apply -f deploy.yml
deployment.apps/my-deploy created
service/my-service created
prasanndh.raaju@DESKTOP-S03TDPN:~$ kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
my-deploy-5fd77cc989-z2qkv         1/1     Running   0           8s
prasanndh.raaju@DESKTOP-S03TDPN:~$ 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://127.0.0.1:39637 |


🔧 Opening service default/my-service in default browser...
🔥 http://127.0.0.1:39637
! 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.
prasanndh.raaju@DESKTOP-S03TDPN:~$ curl http://192.168.49.2:30002/maven-web-app/
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
prasanndh.raaju@DESKTOP-S03TDPN:~$
```

```
prasanndh.raaju@DESKTOP-: ~$ kubectl get pod
prasanndh.raaju@DESKTOP-S03TDPN:~$ kubectl get pod
E0321 04:54:21.878791 1622 memcache.go:265] "Unhandled Error" err="couldn't get current server API group list: Get \"https://127.0.0.1:32769/api?timeout=32s\": dial tcp 127.0.0.1:32769: connect: connection refused"
E0321 04:54:21.881465 1622 memcache.go:265] "Unhandled Error" err="couldn't get current server API group list: Get \"https://127.0.0.1:32769/api?timeout=32s\": dial tcp 127.0.0.1:32769: connect: connection refused"
E0321 04:54:21.883728 1622 memcache.go:265] "Unhandled Error" err="couldn't get current server API group list: Get \"https://127.0.0.1:32769/api?timeout=32s\": dial tcp 127.0.0.1:32769: connect: connection refused"
E0321 04:54:21.885890 1622 memcache.go:265] "Unhandled Error" err="couldn't get current server API group list: Get \"https://127.0.0.1:32769/api?timeout=32s\": dial tcp 127.0.0.1:32769: connect: connection refused"
E0321 04:54:21.887491 1622 memcache.go:265] "Unhandled Error" err="couldn't get current server API group list: Get \"https://127.0.0.1:32769/api?timeout=32s\": dial tcp 127.0.0.1:32769: connect: connection refused"
The connection to the server 127.0.0.1:32769 was refused - did you specify the right host or port?
prasanndh.raaju@DESKTOP-S03TDPN:~$ minikube start
🐳 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: default-storageclass, storage-provisioner
👉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
prasanndh.raaju@DESKTOP-S03TDPN:~$ kubectl get pod
No resources found in default namespace.
prasanndh.raaju@DESKTOP-S03TDPN:~$ sudo nano deploy.yml
[sudo] password for prasanndh.raaju:
prasanndh.raaju@DESKTOP-S03TDPN:~$ kubectl apply -f deploy.yml
deployment.apps/my-deploy created
service/my-service created
prasanndh.raaju@DESKTOP-S03TDPN:~$ kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
my-deploy-5fd77cc989-z2qkv          1/1     Running   0           8s
prasanndh.raaju@DESKTOP-S03TDPN:~$ 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://127.0.0.1:39637 |


```

```
prasanndh.raaju@DESKTOP-: ~$ kubectl get pod
E0321 04:54:21.887491 1622 memcache.go:265] "Unhandled Error" err="couldn't get current server API group list: Get \"https://127.0.0.1:32769/api?timeout=32s\": dial tcp 127.0.0.1:32769: connect: connection refused"
The connection to the server 127.0.0.1:32769 was refused - did you specify the right host or port?
prasanndh.raaju@DESKTOP-S03TDPN:~$ minikube start
🐳 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: default-storageclass, storage-provisioner
👉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
prasanndh.raaju@DESKTOP-S03TDPN:~$ kubectl get pod
No resources found in default namespace.
prasanndh.raaju@DESKTOP-S03TDPN:~$ sudo nano deploy.yml
[sudo] password for prasanndh.raaju:
prasanndh.raaju@DESKTOP-S03TDPN:~$ kubectl apply -f deploy.yml
deployment.apps/my-deploy created
service/my-service created
prasanndh.raaju@DESKTOP-S03TDPN:~$ kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
my-deploy-5fd77cc989-z2qkv          1/1     Running   0           8s
prasanndh.raaju@DESKTOP-S03TDPN:~$ 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://127.0.0.1:39637 |


🌐 Opening service default/my-service in default browser...
👉 http://127.0.0.1:39637
! 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.
prasanndh.raaju@DESKTOP-S03TDPN:~$ curl http://192.168.49.2:30002/maven-web-app/
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
prasanndh.raaju@DESKTOP-S03TDPN:~$ |
```

```
prasanndh.raaju@DESKTOP-503TDPN:~$ kubectl get node
NAME          STATUS    ROLES    AGE   VERSION
minikube      Ready    control-plane  47h   v1.32.0
prasanndh.raaju@DESKTOP-503TDPN:~$ cd ~/.kube
prasanndh.raaju@DESKTOP-503TDPN:~/.kube$ ls
cache  config
prasanndh.raaju@DESKTOP-503TDPN:~/.kube$ cat config
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /home/prasanndh.raaju/.minikube/ca.crt
    extensions:
    - extension:
        last-update: Fri, 21 Mar 2025 04:54:53 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 04:54:53 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/prasanndh.raaju/.minikube/profiles/minikube/client.crt
    client-key: /home/prasanndh.raaju/.minikube/profiles/minikube/client.key
prasanndh.raaju@DESKTOP-503TDPN:~/.kube$ sudo vi config
prasanndh.raaju@DESKTOP-503TDPN:~/.kube$ kubectl get node
NAME          STATUS    ROLES    AGE   VERSION
minikube      Ready    control-plane  47h   v1.32.0
prasanndh.raaju@DESKTOP-503TDPN:~/.kube$
```

## Commands:

jenkins ALL=(ALL) NOPASSWD: ALL

sudo systemctl restart ssh.service

sudo systemctl restart sshd.service

sudo apt update

#Installing SSH key

sudo apt install openssh-server

sudo systemctl restart ssh

sudo systemctl status ssh

ls /etc/systemd/system/sshd.service or ls /usr/lib/systemd/system/sshd.service

sudo systemctl daemon-reload

sudo systemctl status ssh

sudo systemctl restart ssh.service

cat /home/david/.minikube/ca.crt | base64 -w 0; echo

sudo chmod 666 /var/run/docker.sock

<https://192.168.39.226:8443>

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

minikube service my-service --url | xargs curl

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 .'
            }
        }
    }

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

    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'
            }
        }
    }

```

```

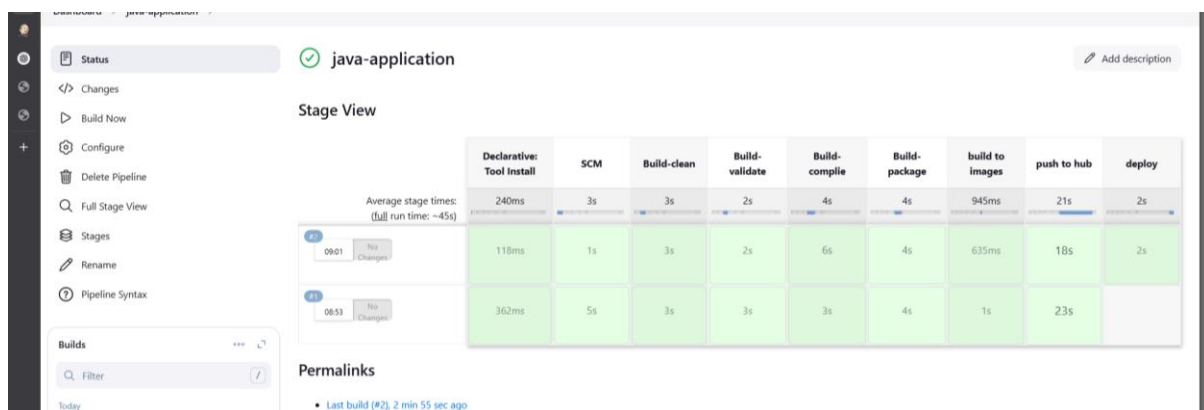
    }
    stage('Test') {
        steps {
            withKubeConfig(caCertificate: '', clusterName: 'minikube', contextName: 'minikube',
credentialsId: 'mukubeconfig_011', namespace: '', restrictKubeConfigAccess: false, serverUrl:
'https://192.168.49.2:8443') {

                sh 'minikube service my-service --url | xargs curl'
            }
        }
    }
}
}
}
}
}
terraform {
    required_providers {
        aws = {
            source = "hashicorp/aws"
            version = "5.92.0"
        }
    }
}

provider "aws" {
    # Configuration options
}

```

### 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"
}

```

## Terraform commands:



# the essential Terraform Cheatsheet

by justin o'connor

### general commands

- get the terraform version  
`terraform version`
- download and update root modules  
`terraform get -update=true`
- open up a terraform interactive terminal  
`terraform console`
- create a dot diagram of terraform dependencies  
`terraform graph | dot -Tpng > graph.png`
- format terraform code to HCL standards  
`terraform fmt`
- validate terraform code syntax  
`terraform validate`
- enable tab auto-completion in the terminal  
`terraform -install-autocomplete`
- show information about provider requirements  
`terraform providers`
- login and logout of terraform cloud  
`terraform login` and `terraform logout`

### workspaces

- list the available workspaces  
`terraform workspace list`
- create a new workspace  
`terraform workspace new development`
- select an existing workspace  
`terraform workspace select default`

### initialize terraform

- initialize terraform in the current working directory  
`terraform init`
- skip plugin installation  
`terraform init -get-plugins=false`
- force plugin installation from a directory  
`terraform init -plugin-dir=PATH`
- upgrade modules and plugins at initialization  
`terraform init -upgrade`
- update backend configuration  
`terraform init -migrate-state -force-copy`
- skip backend configuration  
`terraform init -backend=false`
- use a local backend configuration  
`terraform init -backend-config=FILE`
- change state lock timeout (default is zero seconds)  
`terraform init -lock-timeout=120s`

### plan terraform

- produce a plan with diff between code and state  
`terraform plan`
- output a plan file for reference during apply  
`terraform plan -out current.tfplan`
- output a plan to show effect of terraform destroy  
`terraform plan -destroy`
- target a specific resource for deployment  
`terraform plan -target=ADDRESS`

*note that the -target option is also available for the terraform apply and terraform destroy commands.*

### outputs

- list available outputs  
`terraform output`
- output a specific value  
`terraform output NAME`

### apply terraform

- apply the current state of terraform code  
`terraform apply`
- specify a previously generated plan to apply  
`terraform apply current.tfplan`
- enable auto-approval or automation  
`terraform apply -auto-approve`

### destroy terraform

- destroy resources managed by terraform state  
`terraform destroy`
- enable auto-approval or automation  
`terraform destroy -auto-approve`

### manage terraform state

- list all resources in terraform state  
`terraform state list`
- show details about a specific resource  
`terraform state show ADDRESS`
- track an existing resource in state under new name  
`terraform state mv SOURCE DESTINATION`
- import a manually created resource into state  
`terraform state import ADDRESS ID`
- pull state and save to a local file  
`terraform state pull > terraform.tfstate`
- push state to a remote location  
`terraform state push PATH`
- replace a resource provider  
`terraform state replace-provider A B`
- taint a resource to force redeployment on apply  
`terraform taint ADDRESS`
- untaint a previously tainted resource  
`terraform untaint ADDRESS`

Version 1    <https://justinoconnor.codes>