RAGULANDIRAN M – 22CSR157 || KONGU ENGINNERING COLLEGE

DAY 5 Tasks – minikube deployment, Terraform

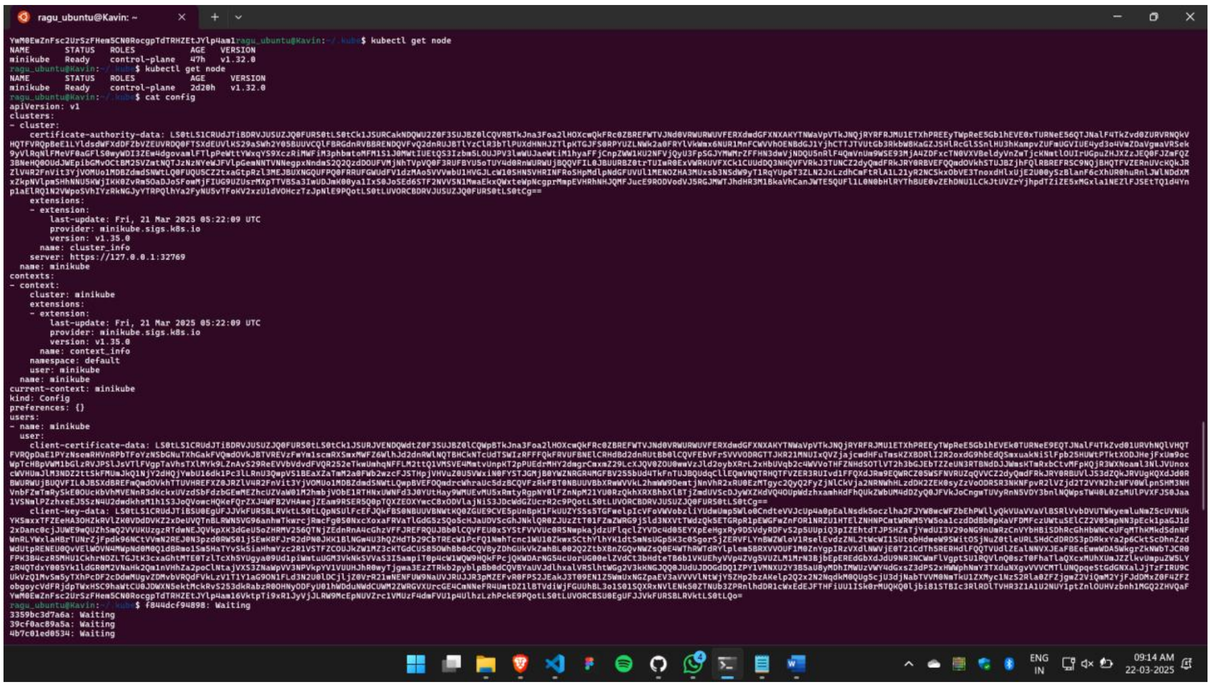
Minikube Deployment:

Step 1: Config file updation

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

Data updation in config file:

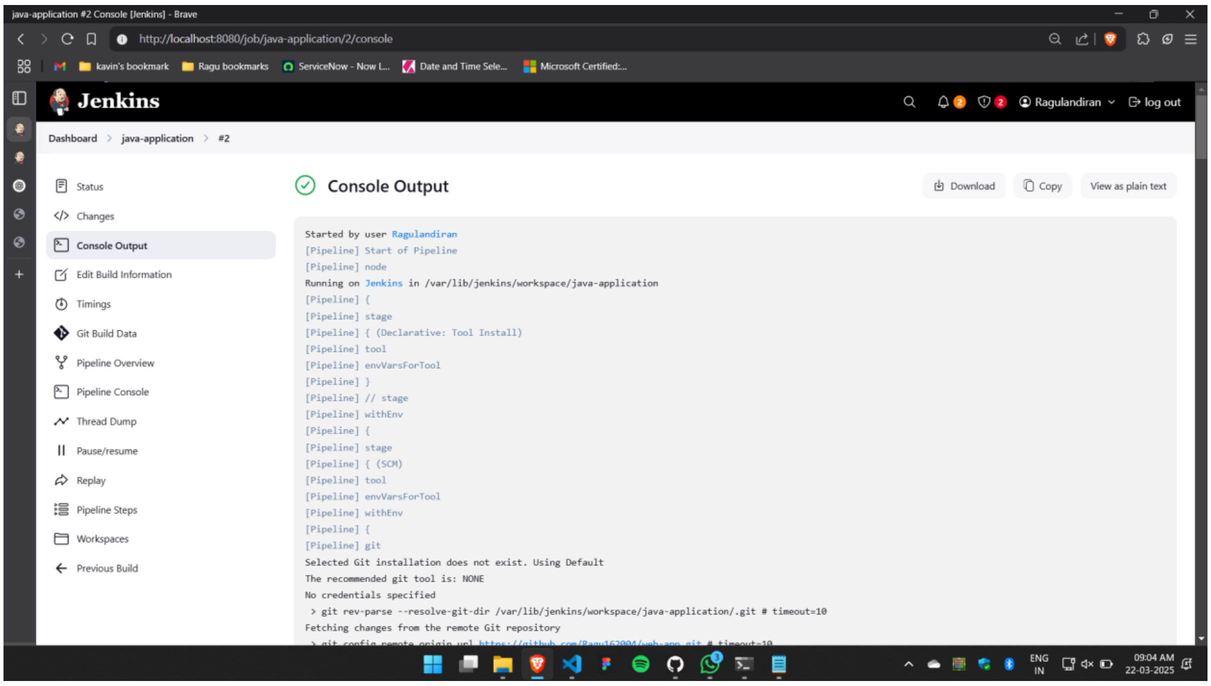
- sudo cat /home/ragu_ubuntu/.minikube/ca.crt
- sudo cat /home/ragu_ubuntu/.minikube/ca.crt | base64 -w 0; echo
- sudo cat /home/ragu_ubuntu/.minikube/profiles/minikube/client.crt | base64 -w 0; echo
- sudo cat /home/ragu_ubuntu/.minikube/profiles/minikube/client.key | base64 -w 0; echo



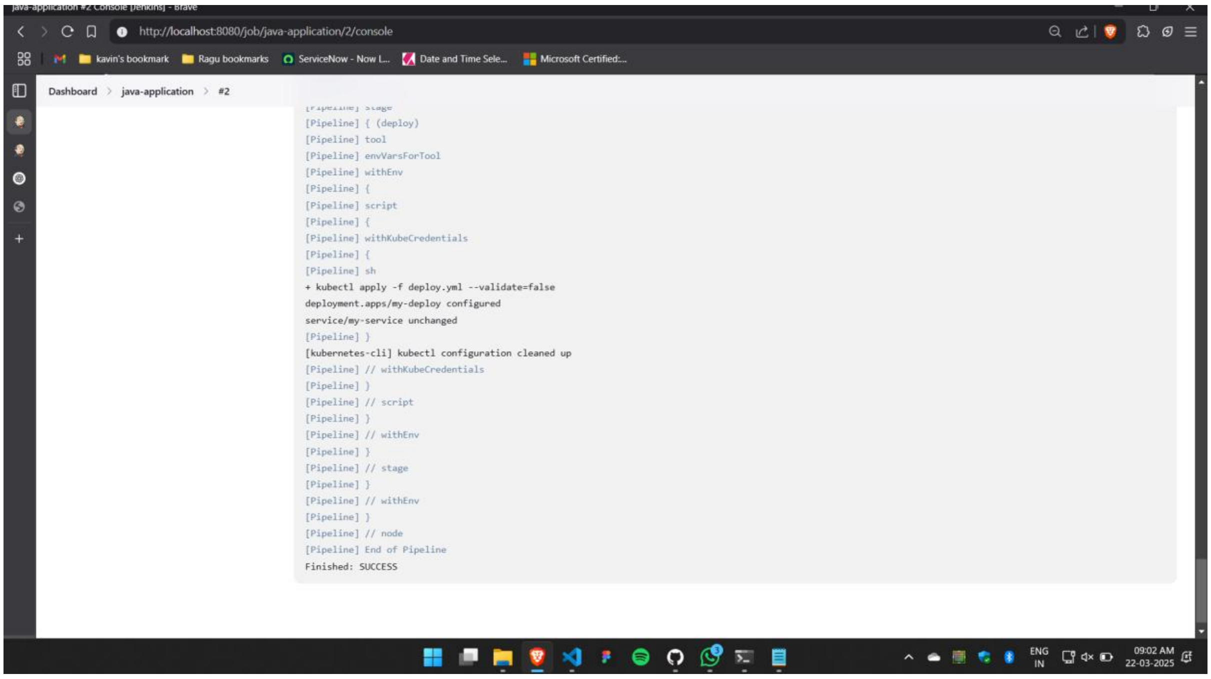
```
ragu_ubuntu@Kavin: ~
YwM0EwZnFsc2UrSzFHem5CN0RocgpTdTRHZEtJYlp4am1ragu_ubuntu@Kavin:~/.kube$ kubectl get node
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   47h   v1.32.0
ragu_ubuntu@Kavin:~/.kube$ kubectl get node
NAME       STATUS   ROLES           AGE     VERSION
minikube   Ready    control-plane   2d20h   v1.32.0
ragu_ubuntu@Kavin:~/.kube$ cat config
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCakNDQWU2Z0F3SUJBZ0lCQVRBTkJ[illegible]
HQTFVRQpBeE1LYldsdWFXdDFZbVZEUVRDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBTlYzClR3bTlPUX[illegible]
9yVlRqNlFMeVF0aGFlS0wyWDI3ZEw4dgovamlFTlpPeWttYWxqYS9XczRiMWFiM3phbmtoMFM1S1J0MWtIUEtQS3Izbm5LOUJPV3lwW[illegible]
3BNeHQ0OUdJWEpibGMvOCtBM25VZmtNQTJzNzNYeWJFVlpGemNNTVNNegpxNndmS2Q2QzdDOUFVMjNhTVpVQ0F3RUFBYU5oTUY4d0Rn[illegible]
ZlV4R2FnVit3YjVOMUo1MDBZdmdSNWtLQ0FUQU5CZ2txaGtpRzl3MEJBUXNGQUFPQ0FRRUFGWUdFV1dzMAo5VVVwbU1HVGJLcW10SHN[illegible]
xZkpNVlpmSHhNNU5KWjIKK0ZvRm5OaDJoSFowMjFIUG9UZUsrMXpTTVBSa3IwUDJmK00ya1IxS0JoSEd6STF2NVVSN1MwaEkxQWxteW[illegible]
p1aElRQ1N2VWpoSVhIYzRkNGJyYTRPQlhYa2FyNU5vTFoKV2xzU1dVOHczTzJpNlE9PQotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==
    extensions:
    - extension:
        last-update: Fri, 21 Mar 2025 05:22:09 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 05:22:09 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-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURJVENDQWdtZ0F3SUJBZ0lCQWpBTkJna3[illegible]
FVRQpDaE1PYzNsemRHVnRPbTFoYzNSbGNuTXhGakFVQmdOVkJBTVREVzFwYm1scmRXSmxMWFZ6WlhJd2dnRWlNQTBHCkNTcUdTSIzR[illegible]
WpTcHBpVWM1bGlzRVJPSlJsVTlFVgpTaVhsTXlMYk9LZnAvS29ReEVVbVdvdFVQR252eTkwUmhqNFFLM2ttQ1VMSVE4MmtvUnpKT2pP[illegible]
cWVHUmJlM3NDZ2ttSkFMUmJkQ1NjY2dHQjYwbU16dk1Pc3lLRnU3QwpVS1BEaXZaTmM2a0FWb2wzcFJSTHpjVHVuZ0U5VWxiN0FYSTJ[illegible]
BWURWUjBUQVFIL0JBSXdBREFmQmdOVkhTTUVHREFXZ0JRZlV4R2FnVit3YjVOMUo1MDBZdmdSNWtLQwpBVEFOQmdrcWhraUc5dzBCQV[illegible]
VnbFZwTmRySkE0OUcKbVhMVENnR3dKckxUVzdSbFdzbGEwMEZhcUZVaW01M2hmbjVObE1RTHNxUWNFd3J0YUtHay9WMUEvMU5xRmtyR[illegible]
1VSNmlPZzhxeEJ5SzN4U2dwdkhsM1h1S3JoQVowcHQKeFQrZXJ4WFB2VHAwejZEam9RSER5Q0pTQXZEOXYwcC8xODVlajNiS3JDcWdG[illegible]
    client-key-data: LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlFcEFJQkFBS0NBUUVBNWtKQ0ZGUE9CVE5pUn[illegible]
YKSmxxTFZEeHA3OHZkRVlZK0VDdDVKZ2xDeUVQTnBLRWN5VG96anhwTkwrcjRmcFg0S0NxcXoxaFRVaTlGdG5zSQo5cHJaUDVScGhJN[illegible]
2xDanc0cjJUWE9wQUZhSmQ2VVUKUzgzRTdwNEJQVkpXK3dGeU5oZHRMV256QTNjZEdnRnA4cGhzVFFJREFRQUJBb0lCQVFEU0xSYStF[illegible]
WnRLYWxlaHBrTUNrZjFpdk96NCtVVmN2REJ0N3pzd0RWS01jSEwKRFJrR2dPN0JKK1BlNGw4U3hQZHdTb29CbTREcW1PcFQ1NmhTcnc[illegible]
WdUtpRENEU0QvVElWOVN4MWpNd0M0Q1dBRmo1Sm5HaTYvSk5iaHhmYzc2R1VSTFZCOUJkZW1MZ3cKTGdCUS85OWhBb0dCQVByZDhGUk[illegible]
FPK3B4czR5MHU1CkhrNDZLTGJtK3cxaGhtMTE0TzlTcXh5YUgya09Ud1piWmtuUGM3VkNkSVVaS3I5ampiT0p4cW1WQW9HQkFPcjQKW[illegible]
zR4QTdxY005Yk1ldGR0M2VNaHk2Qm1nVHhZa2poClNtajVXS3ZNaWpVV3NPVkpYV1VUUHJhR0wyTjgwa3EzZTRkb2pyblpBb0dCQVBY[illegible]
UkVzQ1MvSm5yTXhPcDF2cDdwMUgvZDMvbVRQdFVkLzV1T1Y1aG9ON1FLd3N2U0lDCjljZ0VrR21wNENFUW9NaUVJRUJJR3pMZEFvR0F[illegible]
obgoycVdFRjdpTWxHSC9haWtCU0JDWXN5ektMckRvS253dkRabzR0OHNyODFyU01hWDduNWdCUWM2ZWRGVXUrcGE4CmNNeFR4UmtDZ1[illegible]
YwM0EwZnFsc2UrSzFHem5CN0RocgpTdTRHZEtJYlp4am16VktpTi9xR1JyVjJLRW9McEpNUVZrc1VMUzF4dmFVU1p4UlhzLzhPckE9P[illegible]
ragu_ubuntu@Kavin:~/.kube$ f844dcf94898: Waiting
3359bc3d7a6a: Waiting
39cf0ac89a5a: Waiting
4b7c01ed0534: Waiting
```

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'
                    }
                }
            }
        }
    }
}
```

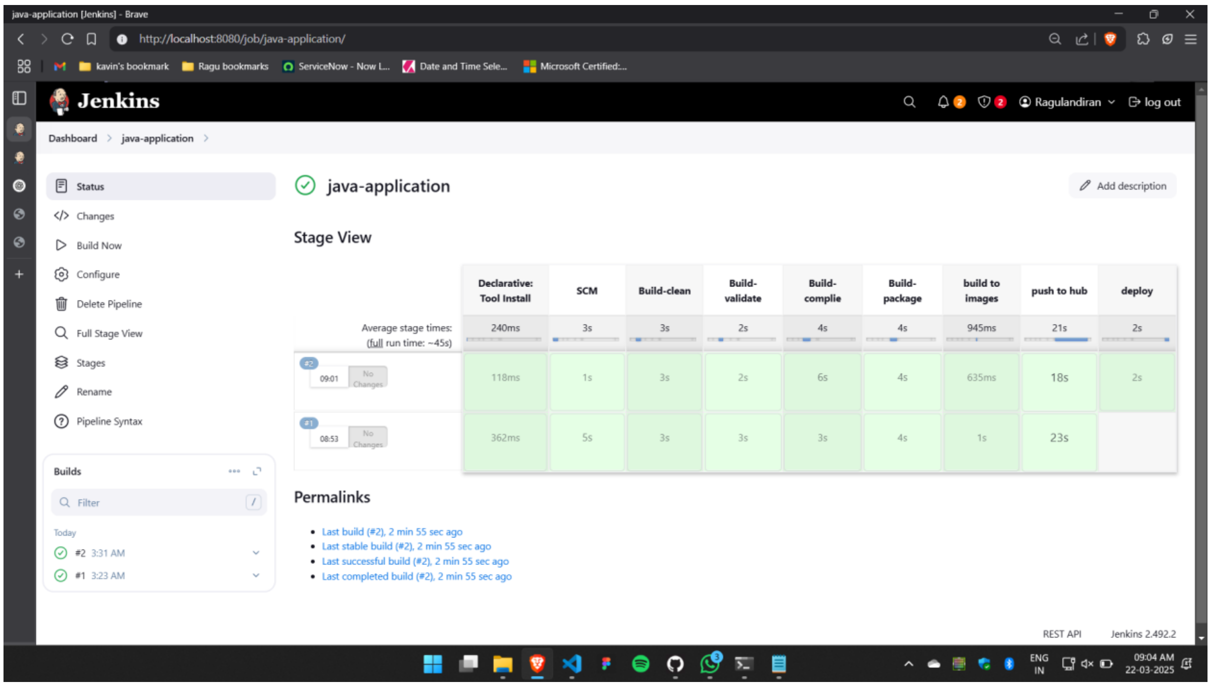

Console Output

```
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
```



```
[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
```

Stage View of Pipeline:



java-application

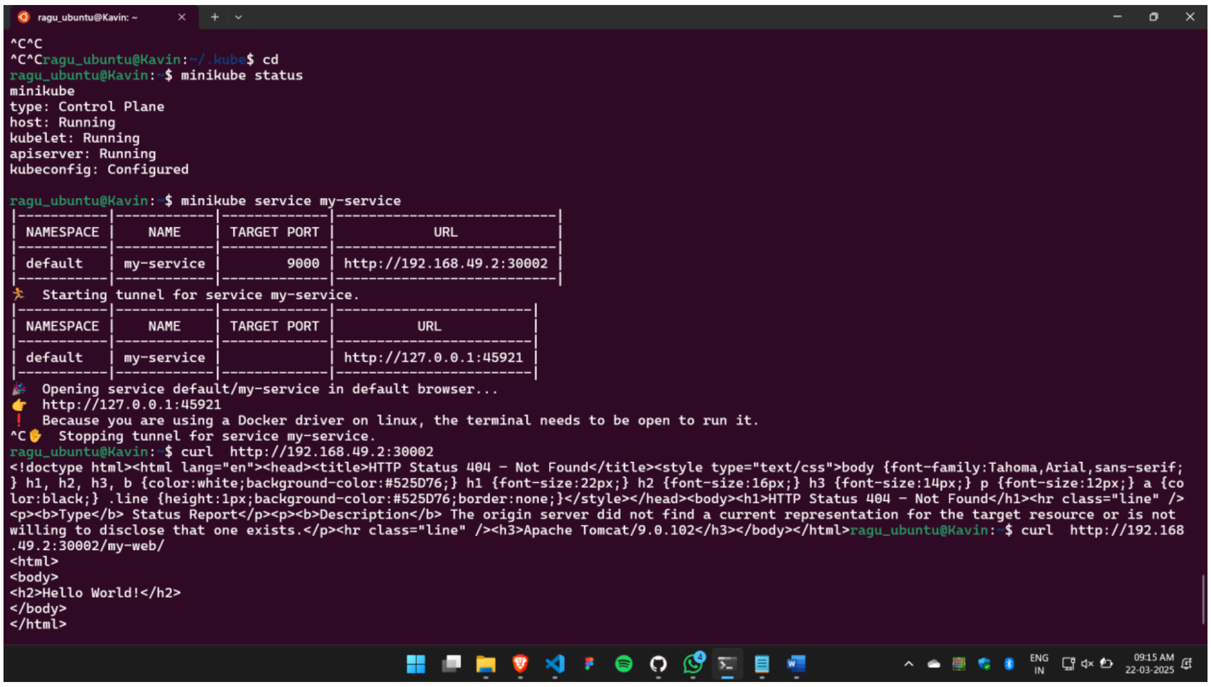
Stage View

	Declarative: Tool Install	SCM	Build-clean	Build-validate	Build-complie	Build-package	build to images	push to hub	deploy
Average stage times: (full run time: ~45s)	240ms	3s	3s	2s	4s	4s	945ms	21s	2s
#2 09:01 No Changes	118ms	1s	3s	2s	6s	4s	635ms	18s	2s
#1 08:53 No Changes	362ms	5s	3s	3s	3s	4s	1s	23s	

Permalinks

- Last build (#2), 2 min 55 sec ago
- Last stable build (#2), 2 min 55 sec ago
- Last successful build (#2), 2 min 55 sec ago
- Last completed build (#2), 2 min 55 sec ago

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



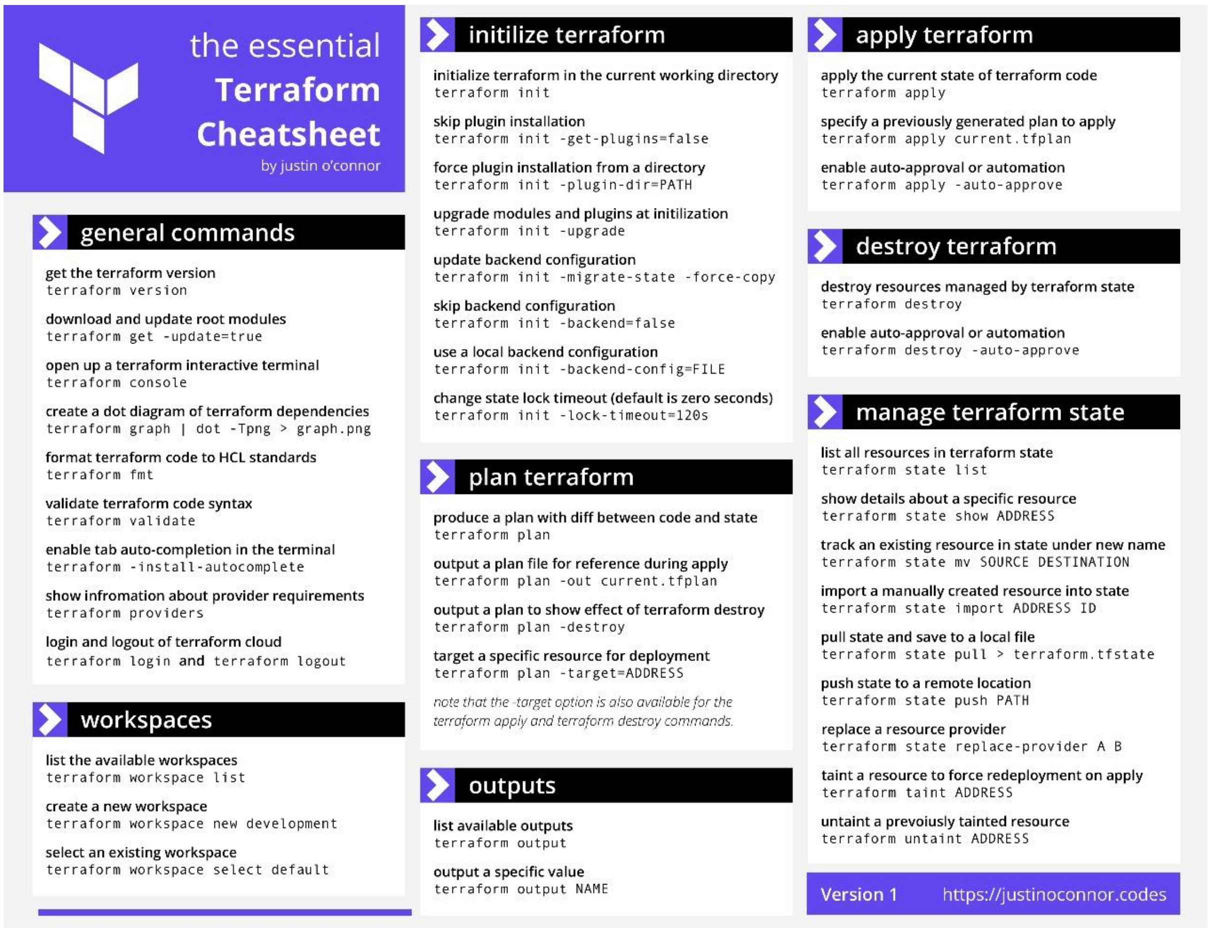
```
^C^C
^C^Cragu_ubuntu@Kavin:~/.kube$ cd
ragu_ubuntu@Kavin:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

ragu_ubuntu@Kavin:~$ 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://127.0.0.1:45921 |
|-----------|------------|-------------|------------------------|
🎉  Opening service default/my-service in default browser...
👉  http://127.0.0.1:45921
❗  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.
ragu_ubuntu@Kavin:~$ 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 {co
lor: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 representation 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>ragu_ubuntu@Kavin:~$ curl  http://192.168
.49.2:30002/my-web/
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
```

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 infromation 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
```

initilize 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 initilization
```
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 prevoiusly tainted resource
```
terraform untaint ADDRESS
```

Version 1 https://justinoconnor.codes