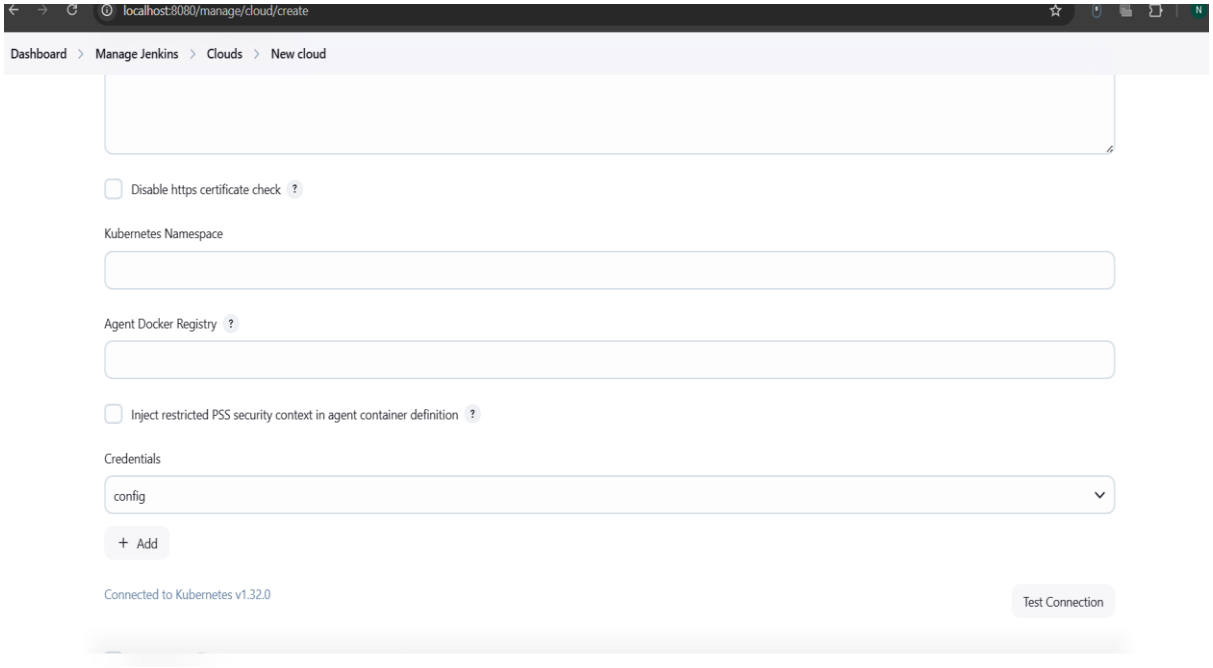
```
nandhu2645@LAPTOP-1TVBN
nandhu2645@LAPTOP-1TVBND2B:~$ sudo apt update
sudo apt install openssh-server
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:2 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:4 http://archive.ubuntu.com/ubuntu noble InRelease
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://archive.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://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [922 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [8960 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [6936 B]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [820 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [177 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [13.5 kB]
Get:16 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1041 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [51.9 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [17.0 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [208 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
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 [208 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 5s (957 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
90 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
nandhu2645@LAPTOP-1TVBN
nandhu2645@LAPTOP-1TVBND2B:~$ sudo systemctl restart ssh
nandhu2645@LAPTOP-1TVBND2B:~$ 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 05:47:47 UTC; 7s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 1666 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 1667 (sshd)
      Tasks: 1 (limit: 4576)
     Memory: 1.2M ()
     CGroup: /system.slice/ssh.service
             └─1667 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Mar 21 05:47:47 LAPTOP-1TVBND2B systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Mar 21 05:47:47 LAPTOP-1TVBND2B sshd[1667]: Server listening on :: port 22.
Mar 21 05:47:47 LAPTOP-1TVBND2B systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
nandhu2645@LAPTOP-1TVBND2B:~$ 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
nandhu2645@LAPTOP-1TVBND2B:~$ ls /etc/systemd/system/sshd.servic
ls: cannot access '/etc/systemd/system/sshd.servic': No such file or directory
nandhu2645@LAPTOP-1TVBND2B:~$ ls /usr/lib/systemd/system/sshd.service
ls: cannot access '/usr/lib/systemd/system/sshd.service': No such file or directory
nandhu2645@LAPTOP-1TVBND2B:~$ sudo systemctl daemon-reload
nandhu2645@LAPTOP-1TVBND2B:~$ 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 05:47:47 UTC; 1min 39s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 1667 (sshd)
      Tasks: 1 (limit: 4576)
     Memory: 1.2M ()
```

```
nandhu2645@LAPTOP-1TVBND2B:~$ 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" ...
🤦  StartHost failed, but will try again: provision: get ssh host-port: get port 22 for "minikube": docker container inspect -f "'{{(index (inde
x .NetworkSettings.Ports "22/tcp") 0).HostPort}}'" minikube: exit status 1
stdout:


stderr:
template parsing error: template: :1:4: executing "" at <index (index .NetworkSettings.Ports "22/tcp") 0>: error calling index: reflect: slice i
ndex out of range

🏃  Updating the running docker "minikube" container ...
🐳  Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
E0321 06:17:00.796627    6054 logFile.go:53] failed to close the audit log: invalid argument
nandhu2645@LAPTOP-1TVBND2B:~$ cd ~/.kube
nandhu2645@LAPTOP-1TVBND2B:~/.kube$ sudo vi config
[sudo] password for nandhu2645:
nandhu2645@LAPTOP-1TVBND2B:~/.kube$ kubectl get node
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   14h   v1.32.0
nandhu2645@LAPTOP-1TVBND2B:~/.kube$
```

```
nandhu2645@LAPTOP-1TVBND2B: ~/.kube
nandhu2645@LAPTOP-1TVBND2B:~$ sudo visudo
[sudo] password for nandhu2645:
nandhu2645@LAPTOP-1TVBND2B:~$ sudo systemctl restart ssh
nandhu2645@LAPTOP-1TVBND2B:~$ cd ~/.kube
nandhu2645@LAPTOP-1TVBND2B:~/.kube$ ls
cache  config
nandhu2645@LAPTOP-1TVBND2B:~/.kube$ sudo vi config
nandhu2645@LAPTOP-1TVBND2B:~/.kube$ cat /home/nandhu2645/.minikube/ca.crt | base64 -w 0; echo
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCakNDQWU2Z0F3SUJBZ0lCQVRBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURVUVFERXdwdGFXNXAKYTNWaVpVTkJNQjRYRF
RJMU1ETXhPREE0TWpFMU0xb1hEVE0xTURNeE56QTRNakUxTTFvd0ZURVRNQkVHQTFVRQpBeE1LYldsdWFXdDFZbVZEUVRDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFv
Q2dnRUJBT11lXCnNFMFRLWW5SY29idTJzQnVpSjl0dHBkd0R2WWNnWXEwN0t5VU1Ub1NJUndrZmlsRXJMbVhDK0ZNRVhYMXVpVU8KYkFKb3o1TUJ4dG01Mmtid2JYekJVVHHhXczFKSnBJYj
F6WGZpTXJ3VVNHZVBkQmxSSTdJS1IyY0hXZDRLQ3F2Qgo4WUxka2o3ZjR3RytQUXVkZENnNE1RaFJhNkF4L25CTWh0U2c0ZGVybmhZUFFVUi9udlBFeno1Q29NVk1jelIzCmw2QTI0dzJ6
UmxZa01aMDk2dE1sRDZwR1F0dWwwc3oyTU9ienl1YlFIU3BVdEprTjM5VzJCeFE2VXpIeUhNcGYKSkEzWTJ4MTNKUEt4ZnpWUTk5N2ZVMHhzMmxHbk5KUWNHZkJPNE00RWFLYmFHaERCck
5UaG90VVRvZ2xPWEk3cwpkY2l6ZGZrUmw1QUZWVmp5LzgwQ0F3RUFBYU5oTUY4d0RnWURWUjBQQVFIL0JBUURBZ0trTUIwR0ExVWRKUVFXCk1CUUdDQ3NHQVFVRkJ3TUNCZ2dyQmdFRkJR
Y0RBVEFQQmdOVkhSTUJBZjhFQlRBREFRSC9NQjBHQTFVZERnUVcKQkJRanJLbTJaU1FBWTZRUHFRWEVoVkc2ZG1xNkNqQU5CZ2txaGtpRzl3MEJBUXNGQUFPQ0FRRUFJNmd0YTlYZWpDT2
FocGpUd3FySk1uSE1Qc1h1UTVDY0Ivc2JYYkhhZmZIUm1PRy9tOFVTTGJCeTNyTHBHWWVVZitXWjd4ZDJFCjlIcXJUaSs5bFBPYlVuQ2ZQbFJZa1JlQW0yQzlUSUI5N2FSeE1KWE92K0w2
RGVMd0h0UCtDT2NvUG9yeVlpdWUKK2hXemV0eUlGdjIyUXBJT0h4ZGdBclVRZDlCRnlKUXlWSGZmeHlhWlRTRmkvWXpXTnRnSkNWaFVLMGlCUjNQdwpod05hQkRGK3hITDBtVUNzVzBUYT
kwUytId2hJbVJGd1d6QktuVUluQ2VHVkdRT1VMR285cmpZZHE1VnVWRTJvCmNVVXhnNWpjZTkwUEljYTI4ajYxd0ZzajJNOGQwWmhrVGdhS3ZyWHQ1ZmtoL0xNaHRESDVkTTdDMHk4dGJF
SVEKU1plamZCWkNoT3czVXc9PQotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==
nandhu2645@LAPTOP-1TVBND2B:~/.kube$ cat /home/nandhu2645/.minikube/profiles/minikube/client.crt | base64 -w 0; echo
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURJVENDQWdtZ0F3SUJBZ0lCQWpBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURVUVFERXdwdGFXNXAKYTNWaVpVTkJNQjRYRF
RJMU1ETXhPVEUyTXpBeU1Wb1hEVEk0TURNeE9URTJNekF5TVZvd01URVhNQlVHQTFVRQpDaE1PYzNsemRHVnRPbTFoYzNSbGNuTXhGakFVQmdOVkJBTVREVzFwYm1scmRXSmxMWFZ6WlhJ
d2dnRWlNQTBHCkNTcUdTSWIzRFFFQkFRVUFBNElCRHdBd2dnRUtBb0lCQVFEZE8yM1NRc0I0NlhHanVSZjdaNWllVkNRUUR2TjMKSFRiMGpLN2hOYkx4a09WV1B0S3UzYzVoRThLaTVmcW
ExTjUwallsN2F6WnIvQW9OV21QTy93ckI2RUtRNnNOSQpjYlRpcEFxaGZYYkY1M2NGU0t4RnlZRGhSYnNHS2xoQWgvMVp2L05yQndzWmJaRGdJSzRlakZyWUJMZkVjZzF3CmR3dGdnV3Ft
a1JQMEVjSXo1Mmw2ZGxueldWemF2dlhZbC93YXhUNGtxUlhYVS9XdjRhdEtDNE5pR2gvQWlXMVUKblRBcUVxUS9aVjBoOEh5SzFrUmlaQUFkaG1nNzJaYjg0ekZMTG91ZWIyQzBNdnpkVW
EzdU1Hcjh5UFNma3hzWgp1VEgrTWhFVHhJL2lTVkx5N1NweFpuRHZESmc5S1hwbzVrY0RRRnBweUphc085U29MVmlMYVRnaEFnTUJBQUdqCllEQmVNQTRHQTFVZER3RUIvd1FFQXdJRm9E
QWRCZ05WSFNVRUZqQVVCZ2dyQmdFRkJRY0RBUV1JS3dZQkJRVUgKQXdJd0RBWURWUjBUQVFIL0JBSXdBREFFmQmdOVkhTTUVHREFXZ0JRanJLbTJaU1FBWTZRUHFRWEVoVkc2ZG1xNgpDak
FOQmdrcWhraUc5dzBCQVFzRkFBT0NBUUVBR0VQNGEvZlk3MXF5Nng3dFBPL0xsQlFYeEZMRjN6WVFXTHhKCnd4dG1OMmFwOEQrMnFISW9qZnYrcEFCNWdMMGMrNklJZDRuZ2pGK3l3aXNL
VmxWSjIxclNVY3Bnb0QyNlVVY3UKdmt3Y3hmSkpSb3YvNWZUM0tIcy93L0xJenY5c2VRSjhRMGZJVGxrY1VZc09OTW9hR3lraDRqU3ZMeFJ3ZkNMKwpidHg2cGc4QUIxTld0K2hWcGxDeH
NQVTFGbGhlaHNSLzdMbEZJS0JlMGxQLzY5K0RtVDdrNExNWFgvQlJmWmptCkh6cSs0RFgxbUNML3IrWGVIT0Y5ZFRTa3ZyaHkrSEwwemVzSGx6Tk9FdCsrcGE3TnppUCs0ckRITVNRMFJy
ZTEKcWdpQ0FFMzA3cW9sWmhaOUN0V2dnQlZSZ0NQYUpHMHl3WFZEb0dtQU9HQVI3UzVGRGc9PQotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==
nandhu2645@LAPTOP-1TVBND2B:~/.kube$ cat /home/nandhu2645/.minikube/profiles/minikube/client.key | base64 -w 0; echo
LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlFb2dJQkFBS0NBUUVBM1R0dDBrTEFlT2x4bzdrWCsyZVlubFFrRUE3emR4MDI5SXl1NFRXeThaRGxWajdTCnJ0M09ZU1BDb3
VYNm10VGVkSTJKZTJzMmEvd0tEVnBqenY4S3dlaENrT3JEU0hHMDRxUUtvWDEyeGVkM0JVaXMKUmNtQTRVVzdCaXBZUUlmOVdiL3phd2NMR1cyUTRDQ3VIb3hhMkFTM3hISU5jSGNMWUlG
cXBwRVQ5QkhDTStkcAplblpaODFsYzJyNzEySmY4R3NVK0pLa1YxMVAxcitHclNndURZaG9md01sdFZKMHdLaEtrUDJWZElmQjhpdFpFClltUUFIWVpvTzltVy9PTXhTeTZMbm05Z3RETD
gzVkd0N2pCcS9NajBuNU1iR2JreC9qSVJFOFNQNGtsUzh1MHEKY1dadzd3eVlQU1Y2YU9aSEEwQmFhY2lXckR2VXFDMVlpMms0SVFJREFRQUJBb0lCQUZ2MEg2bU5aVUxjaFFkdwpINisz
VjVBM21PV3owUzVwVGpaSFkwam9jUUJISVl1L2dRSjkzeElxakhQNDh6ajMwWjdRVlp5ZTYzZm13cWJhCk16SU9nQUlvNzhJY1VoOENaQlR2Nmh0RmN1Qy9YSkV0cTYwMkhBT0dZRDRueT
ViRzcxOG1rWjZpeHRKQzRESekkV0pjSjc1aHFIWmZoY0FTVzludWVuDM4bnphS2hNQWtWZzRiU2VVdEpuV2prNmFkKzV3bHVhc1dvWkMrbWtOMAo2S25jQ1g4UnRwVWRWSlhrZHJNVVd1
Tm1iaTJDLzhSWlRyL0k1WlJNQkh0Ni9iUjk2dWgrd2FHSEtZYXhFTW53CmFDS2JoZXlxRGttWnhpQmJVTVZkSlg1cnVIRU1iVG1UNFRiYXBvOU1idUF5TUxBdGVQM0JObm10aE15SktaY3
UKdGNwRVBnRUNnWUVBM2FSTWEwRmJUUGpTZm1JT0xPWU9RRXNrK0l6YytSRWU2U05XRzMrNEh6cnkwQUlQenB0ZwpZNHEzSUtGT3dTdlJHcEdlVEw2WDFUNVpwRnNVL1FjRDRISDh2VXJ1
SEFRRHRuWGZoU0V6TWppK0VQUHRCL3VzCmlPZkFEblF2b3hCTEtIZFluTkloUmV6eEVES3ZhUkdhVGVQaGxHbzJYSXl6dHRPY0RDa3pVOGtDZ11FQS80YmYKdWs4eWJzbTkzb1U2SWhnd2
k0WHZBbkpISk9aaVd6OHdab2F4Q2RRTHFvQnBjSWZWQUxxZTBwQUVQK1pTcnZlMApjVE9OUTN1eEovTzRIQnVuMWdOdW5BYVZzSVdIL3hvU0lXZUlPeXQreTIxU0xDSXJ4cURndXBCaUhK
ZWF5aDI5CnB3SXM2SFBWQnNHL3FnUThlVzQza0NQRnFycVVVRm0yL21pN2Zaa0NnWUFqQmZkZGEzSjE4MHV2MU9iUWlYbnYKQnZETXBkVzFOZE5nbThqYmg0OGN3RC9tUnEwZ3hNU3lkaF
VCL3Z0bTR2bWo3U2JLa2NPRVd0bm5jczBhUHFXSwpMZmoySUlVSHdFcWw1bkp2b3hObitEZHdqbHNDUys2em80dGh2VkpadmhWaVM1bWhEa05idlZsS2FmZjE3bVluCnNMN1FxUWYyRC9Y
R0NpSTZlYmcyb1FLQmdGSUJ6a0d6OHlLZE9vTjB6SUJ1dFAzQ1FUL092MGgzYSsrTzZ4Q3MKZjVzQTQ3R1h2cjh2eU11UUdUTElwNlIrWGVEcmZJeXNGOVVsbUdpVERtNXU3Y0dBbm5vU2
51M2xIUTFoSjVweQo2TVRjbzdmVDRRQnJBeE9mUkNCOEFCRGZFdjEwNkEzMlk4VkNqNGE0bmJPSjRvS3lQY0RUeWplTjQ5V1p4V0lHCjZNMVpBb0dBU0NIOE53dDNLVWdETWMrQUkzcW9M
Nno2MVBZUlZScnBPK3ArVHRBSjZIalNaUDQrNDhwTUVjVnAKNVN1ZXVod2VzcU0xY2F2TkgxZmhpUzQySFgzUkVNZEpNeGVBL3pMYVVRRk9EbmFQcTdCWVlhK2lhMS9Ib2FmMwo2ZHUxck
NWZGxmRTRKZyt5SVRpQTlQU1JpZDVrSUlSWXFESjBFUWRNZzUxU2V3TVpQL1k9Ci0tLS0tRU5EIFJTQSBQUklWQVRFIEtFWS0tLS0tCg==
nandhu2645@LAPTOP-1TVBND2B:~/.kube$
```



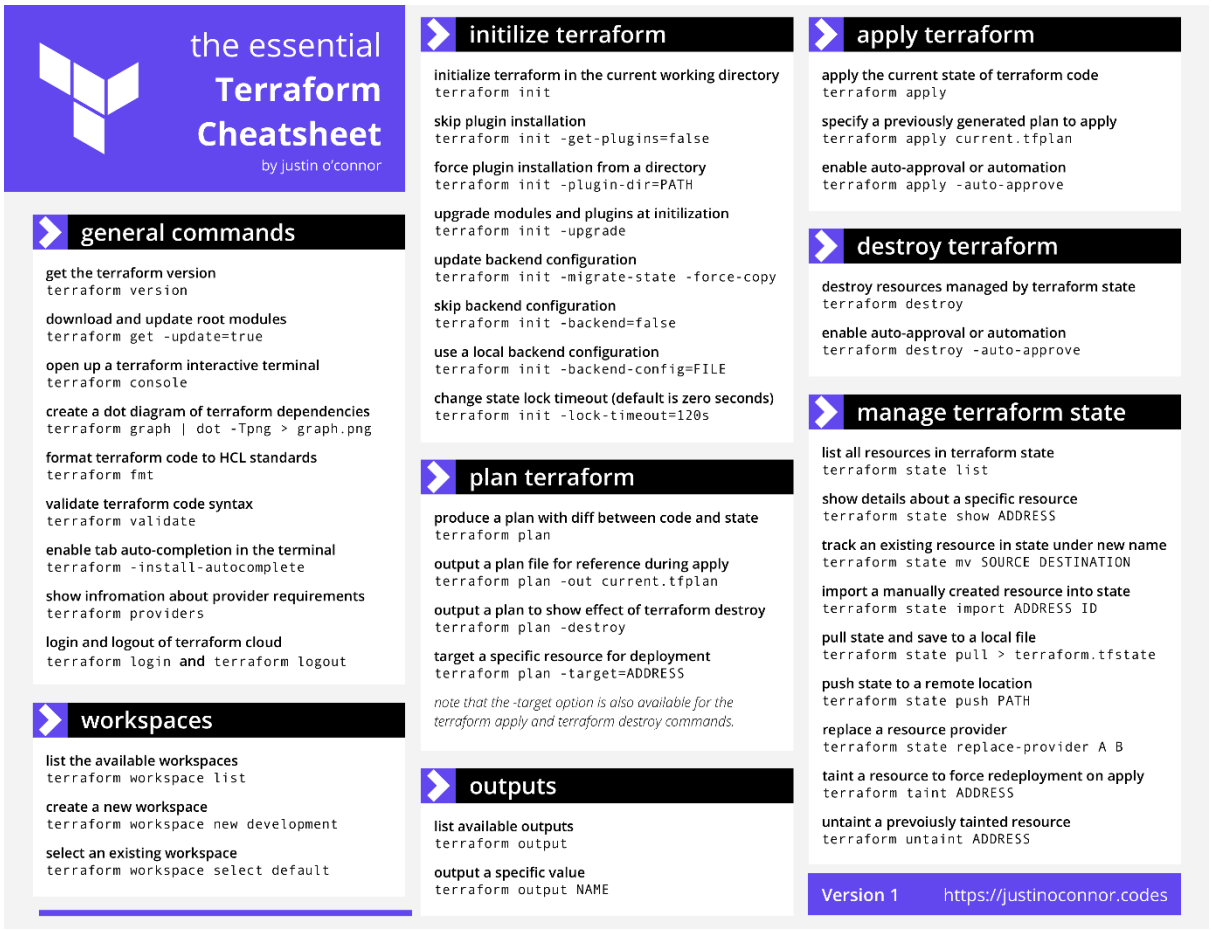
PIPELINE:

```
pipeline {
   agent any

   stages {
      stage('scm') {
         steps {
      git branch: 'master', url:'https://github.com/nandhu6804/devops_main.git'
         }
      }
      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  nandhita22cse126/mysimplewebapp .'
                }
    }
}
stage('push to hub') {
            steps {
                script{
```

```
            withDockerRegistry(credentialsId: 'Docker_cred', url: 'https://index.docker.io/v1/') {
             sh 'docker push nandhita22cse126/mysimplewebapp'
           }
         }
         }
}
      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




```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.92.0"
    }
  }
}
provider "aws" {
  region = "us-east-1"
}
```

```
# Create a VPC
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 = "10.0.1.0/24"
 availability_zone = "us-east-1a"

 tags = {
   Name = "sn2"
 }
}
resource "aws_subnet" "prisub" {
 vpc_id     = aws_vpc.myvpc.id
 cidr_block = "10.0.1.0/24"
 availability_zone = "us-east-1a"

 tags = {
```

```
  Name = "sn3"
 }
}
resource "aws_subnet" "pri_sub" {
 vpc_id     = aws_vpc.myvpc.id
 cidr_block = "10.0.1.0/24"
 availability_zone = "us-east-1a"

 tags = {
  Name = "sn4"
 }
}
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