

```

nandhu2645@LAPTOP-1TVBN:~$ sudo apt update
nandhu2645@LAPTOP-1TVBN:~$ 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:~$ sudo systemctl restart ssh
nandhu2645@LAPTOP-1TVBN:~$ 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-1TVBN systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Mar 21 05:47:47 LAPTOP-1TVBN sshd[1667]: Server listening on :: port 22.
Mar 21 05:47:47 LAPTOP-1TVBN systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
nandhu2645@LAPTOP-1TVBN:~$ 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 '/usr/lib/systemd/system/sshd.service': No such file or directory
nandhu2645@LAPTOP-1TVBN:~$ ls /etc/systemd/system/sshd.servic
ls: cannot access '/etc/systemd/system/sshd.servic': No such file or directory
nandhu2645@LAPTOP-1TVBN:~$ ls /usr/lib/systemd/system/sshd.service
ls: cannot access '/usr/lib/systemd/system/sshd.service': No such file or directory
nandhu2645@LAPTOP-1TVBN:~$ sudo systemctl daemon-reload
nandhu2645@LAPTOP-1TVBN:~$ 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 ( )

```


← → ↻ localhost:8080/manage/cloud/create ☆

Dashboard > Manage Jenkins > Clouds > New cloud

☐ Disable https certificate check ?

Kubernetes Namespace

Agent Docker Registry ?

☐ Inject restricted PSS security context in agent container definition ?

Credentials

config ▼

+ Add

Connected to Kubernetes v1.32.0

Test Connection

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



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>

```
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" "prsn3" {  
  subnet_id    = aws_subnet.prisub.id  
  route_table_id = aws_route_table.tfprirt.id  
}
```

```
resource "aws_route_table_association" "prsn4" {  
  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