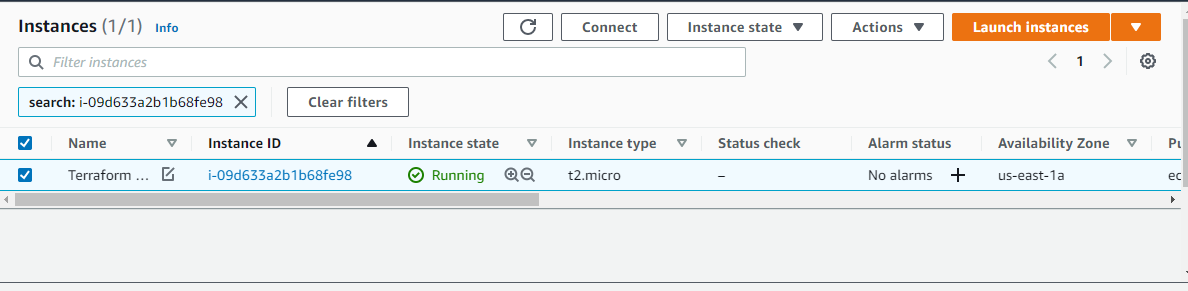
Create an Instance UBUNTU



**Terraform Installation.**

**Open putty. Login as Ubuntu**

curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -

sudo apt-add-repository "deb [arch=$(dpkg --print-architecture)] https://apt.releases.hashicorp.com $(lsb\_release -cs) main"

sudo apt install terraform

**Create MKDIR terraform**

**Cd terraform -> Vim ec2.tf**

**resource "aws\_instance" "myec2" {**

**ami = "ami-09e67e426f25ce0d7"**

**instance\_type = "t2.micro"**

**}**

**Save this and Create a file for Login Credentials**

**All within Terraform folder Vim providers.tf**

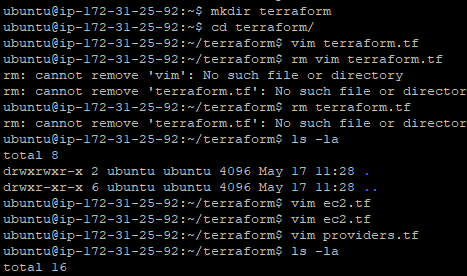
**provider "aws" {**

**region = "us-east-1"**

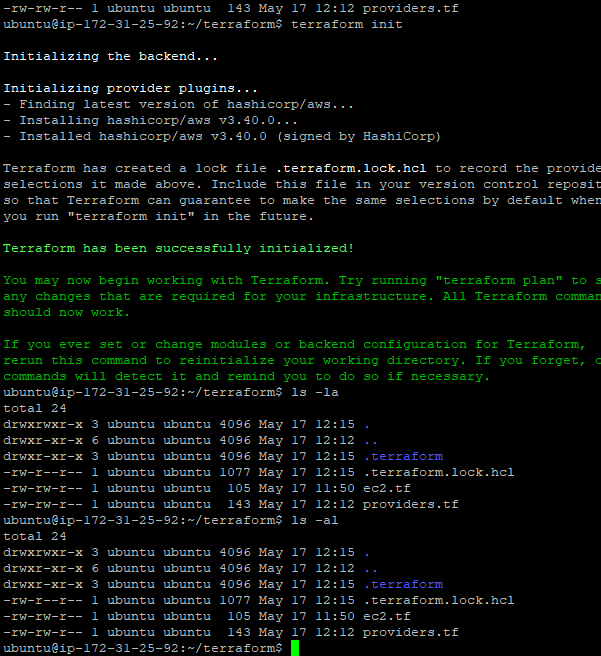
**access\_key = " access\_key "**

**secret\_key = " secret\_key "**

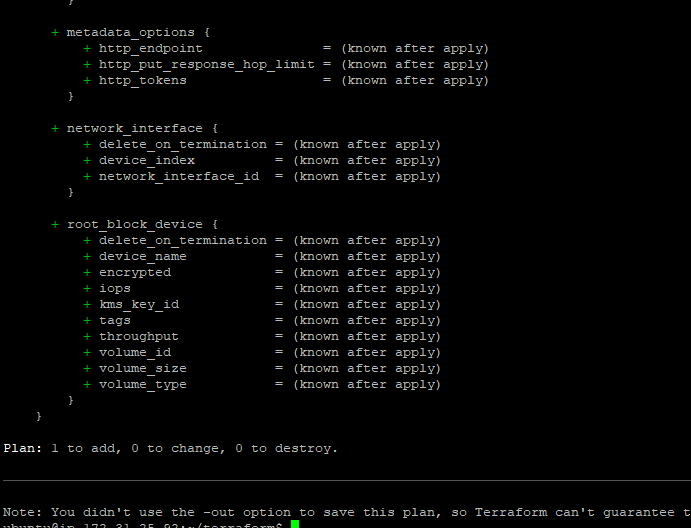
**}**



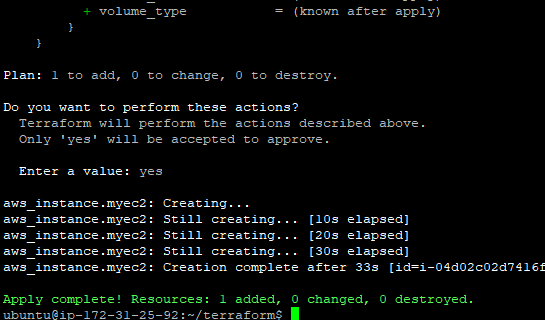
Give **terraform init** command (Initiates new .terraform file )



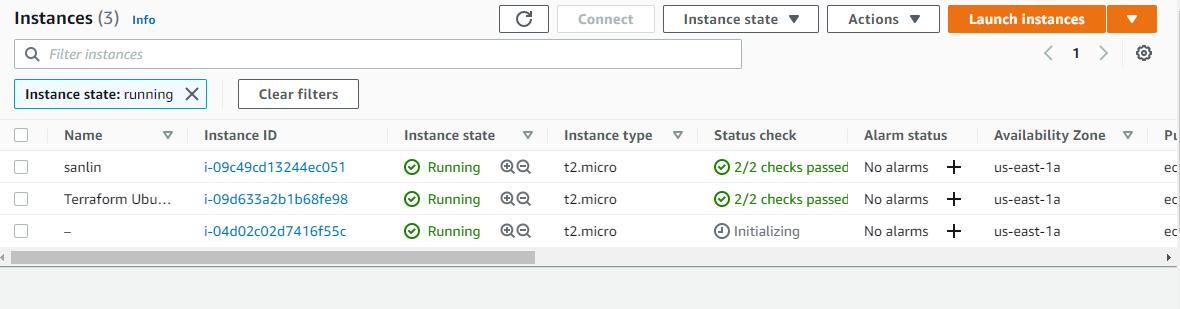
Give **terraform plan** command (show what all the thing will happen if we give plan command)



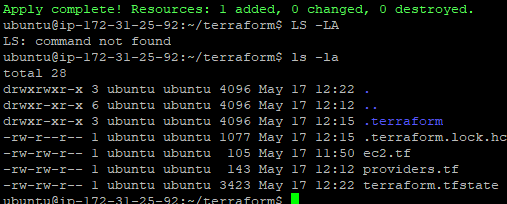
Give **terraform apply** command (similar to plan it will ask only yes or no)



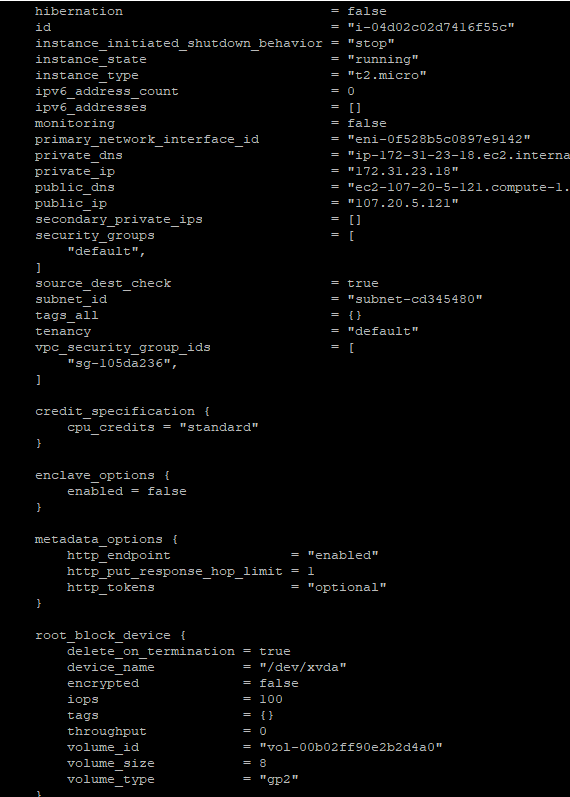
**NEW INSTANCE IS CREATED IN AWS CONSOLE**



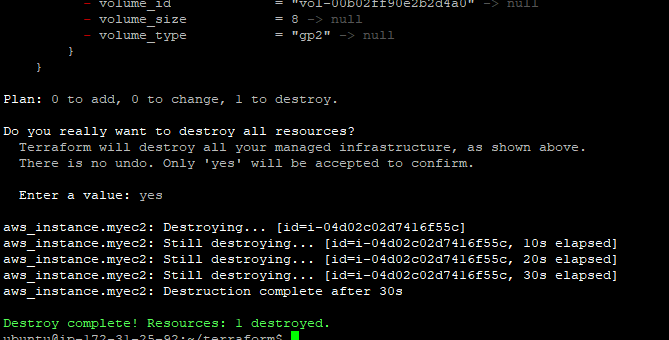
ls –la you can also see a state file created new



Give **terraform show** command (Gives complete details of the created Instance)



Give **terraform destroy** to delete the instance



**Install AWSCLI**

sudo apt install awscli

aws configure

for Ubuntu

sudo apt-get update

apt-get install awscli

aws configure

**Terraform VPC Creation**

**Vim resources.tf**

resource "aws\_vpc" "main" {

cidr\_block = "190.160.0.0/16"

instance\_tenancy = "default"

tags = {

Name = "main"

location = "chennai"

}

}

resource "aws\_subnet" "subnet1" {

vpc\_id = aws\_vpc.main.id

cidr\_block = "190.160.0.0/24"

tags = {

Name = "main"

}

}

**Vim provider.tf**

provider "aws"{

region = "us-east-1"

}

terraform init

terraform plan

terraform apply

Terraform S3 Bucket creation

**resource "aws\_s3\_bucket" "onebucket" {**

**bucket = "testing-s3-with-terraform"**

**acl = "private"**

**versioning {**

**enabled = true**

**}**

**tags = {**

**Name = "Bucket1"**

**Environment = "Test"**

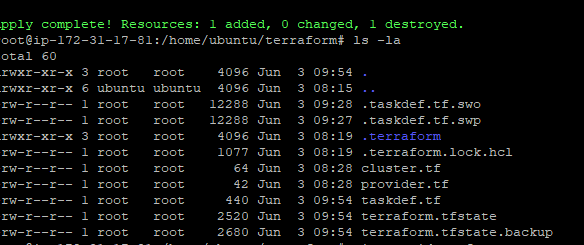
**}**

**}**

Terraform ECS :

Create:

1.Provider, cluster, task definition service files



Provider

provider "aws" {

region = "us-east-1"

}

Cluster.tf to create an (empty cluster)

resource "aws\_ecs\_cluster" "main" {

name = "myapp-cluster"

}

Taskdef.tf

resource "aws\_ecs\_task\_definition" "service" {

family = "service"

container\_definitions = jsonencode([

{

name = "firstdef"

image = "ngnix"

cpu = 1

memory = 350

essential = true

portMappings = [

{

containerPort = 80

hostPort = 80

}

]

}

])

volume {

name = "service-storage"

host\_path = "/ecs/service-storage"

}

}

Terraform VPC

module "vpc" {

source = "terraform-aws-modules/vpc/aws"

name = "santhoshtest VPC"

cidr = "10.10.10.0/24"

azs = ["us-east-1a"]

private\_subnets = ["10.10.10.0/27" , "10.10.10.32/27"]

public\_subnets = ["10.10.10.96/27" , "10.10.10.128/27"]

enable\_nat\_gateway = true

single\_nat\_gateway = true

tags = {

Environment = "dev"

Owner = "me"

}

}

**Terraform Service**

resource "aws\_ecs\_service" "testserve" {

name = "testserve"

cluster = aws\_ecs\_cluster.main.id

task\_definition = aws\_ecs\_task\_definition.testdef.id

desired\_count = 2

}

**Vim Volume.tf**

resource "aws\_ebs\_volume" "test\_volume" {

availability\_zone = "us-east-1a"

size= 10

tags = {

Name = "santhosh\_\_test\_volume"

}

}