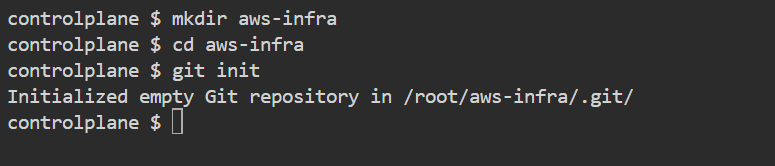
1. **Set up the Git repository:**

***mkdir aws-infra***

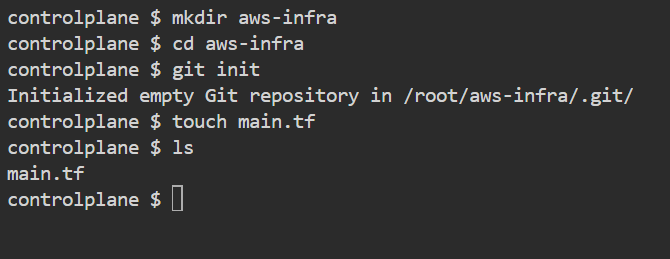
***cd aws-infra***

***git init***



1. **Create Terraform configuration files:**

***touch main.tf***

******

***nano main.tf***

*provider "aws" {*

*region = "us-east-1" # Change to your preferred region*

*}*

*data "aws\_availability\_zones" "available" {}*

*resource "aws\_vpc" "ionginx\_vpc" {*

*cidr\_block = "10.0.0.0/16"*

*tags = {*

*Name = "ionginx-vpc"*

*}*

*}*

*resource "aws\_subnet" "public" {*

*count = 3*

*vpc\_id = aws\_vpc.ionginx\_vpc.id*

*cidr\_block = "10.0.${count.index}.0/24"*

*map\_public\_ip\_on\_launch = true*

*availability\_zone = element(data.aws\_availability\_zones.available.names, count.index)*

*}*

*resource "aws\_subnet" "private" {*

*count = 3*

*vpc\_id = aws\_vpc.ionginx\_vpc.id*

*cidr\_block = "10.0.${count.index + 3}.0/24"*

*map\_public\_ip\_on\_launch = false*

*availability\_zone = element(data.aws\_availability\_zones.available.names, count.index)*

*}*

*resource "aws\_internet\_gateway" "igw" {*

*vpc\_id = aws\_vpc.ionginx\_vpc.id*

*}*

*resource "aws\_nat\_gateway" "nat" {*

*allocation\_id = aws\_eip.nat.id*

*subnet\_id = aws\_subnet.public[0].id*

*}*

*resource "aws\_eip" "nat" {*

*domain = "vpc"*

*}*

*resource "aws\_route\_table" "public" {*

*vpc\_id = aws\_vpc.ionginx\_vpc.id*

*route {*

*cidr\_block = "0.0.0.0/0"*

*gateway\_id = aws\_internet\_gateway.igw.id*

*}*

*}*

*resource "aws\_route\_table\_association" "public" {*

*count = 3*

*subnet\_id = aws\_subnet.public[count.index].id*

*route\_table\_id = aws\_route\_table.public.id*

*}*

*resource "aws\_route\_table" "private" {*

*vpc\_id = aws\_vpc.ionginx\_vpc.id*

*route {*

*cidr\_block = "0.0.0.0/0"*

*nat\_gateway\_id = aws\_nat\_gateway.nat.id*

*}*

*}*

*resource "aws\_route\_table\_association" "private" {*

*count = 3*

*subnet\_id = aws\_subnet.private[count.index].id*

*route\_table\_id = aws\_route\_table.private.id*

*}*

*resource "aws\_launch\_configuration" "nginx" {*

*name = "nginx-lc"*

*image\_id = data.aws\_ami.ubuntu.id*

*instance\_type = "t2.micro"*

*associate\_public\_ip\_address = false*

*user\_data = <<-EOF*

*#!/bin/bash*

*apt-get update*

*apt-get install -y nginx*

*systemctl start nginx*

*systemctl enable nginx*

*EOF*

*lifecycle {*

*create\_before\_destroy = true*

*}*

*}*

*data "aws\_ami" "ubuntu" {*

*most\_recent = true*

*owners = ["099720109477"] # Canonical*

*filter {*

*name = "name"*

*values = ["ubuntu/images/hvm-ssd/ubuntu-\*-20.04-amd64-server-\*"]*

*}*

*}*

*resource "aws\_autoscaling\_group" "nginx" {*

*desired\_capacity = 2*

*max\_size = 4*

*min\_size = 2*

*vpc\_zone\_identifier = aws\_subnet.private.\*.id*

*launch\_configuration = aws\_launch\_configuration.nginx.id*

*tag {*

*key = "Name"*

*value = "nginx"*

*propagate\_at\_launch = true*

*}*

*}*

*resource "aws\_route53\_zone" "main" {*

*name = "example.com" # Change to your domain*

*}*

*resource "aws\_route53\_record" "www" {*

*zone\_id = aws\_route53\_zone.main.zone\_id*

*name = "www"*

*type = "A"*

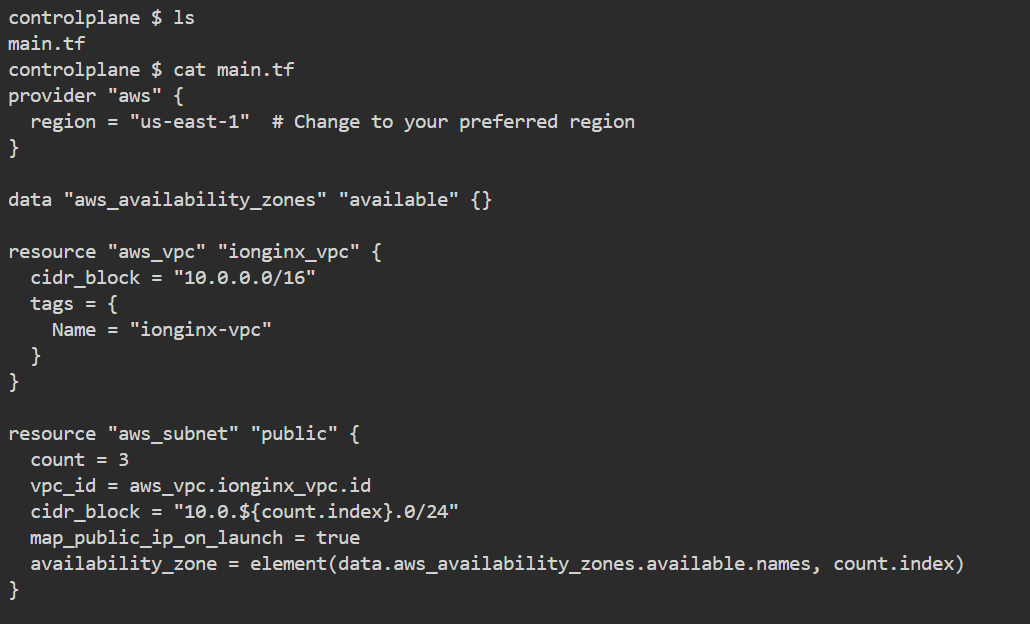
*ttl = 300*

*records = [aws\_eip.nat.public\_ip]*

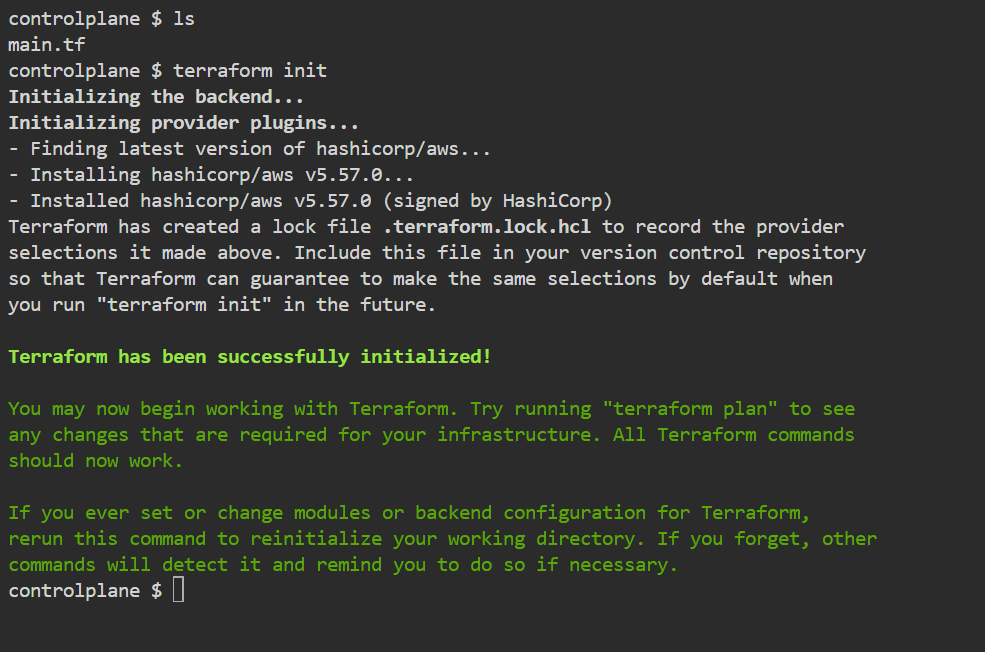
*}*

**

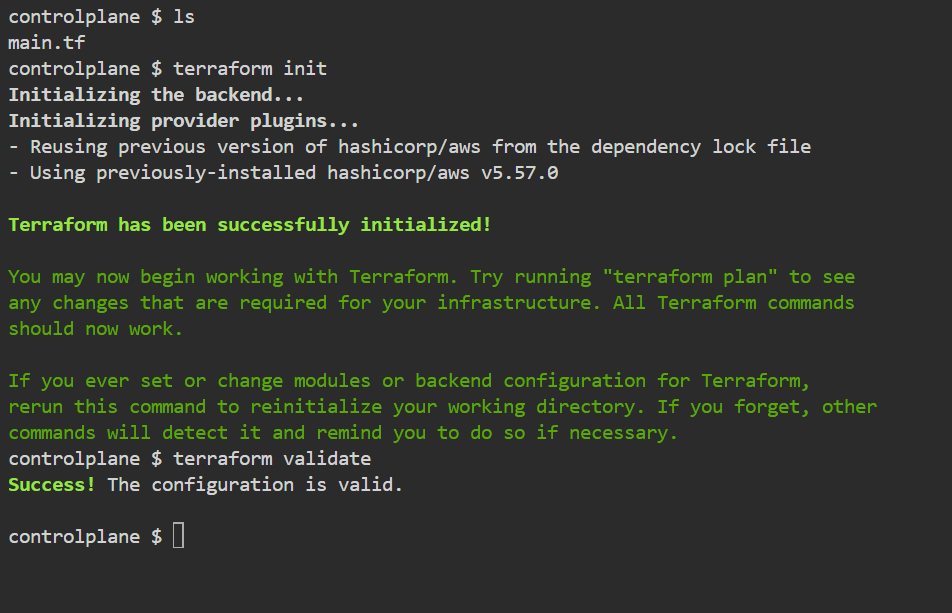
**Verify the main.tf file using the command :- *cat main.tf***



1. **Initialize Terraform:** terraform init



1. **Validate the configuration:** terraform validate

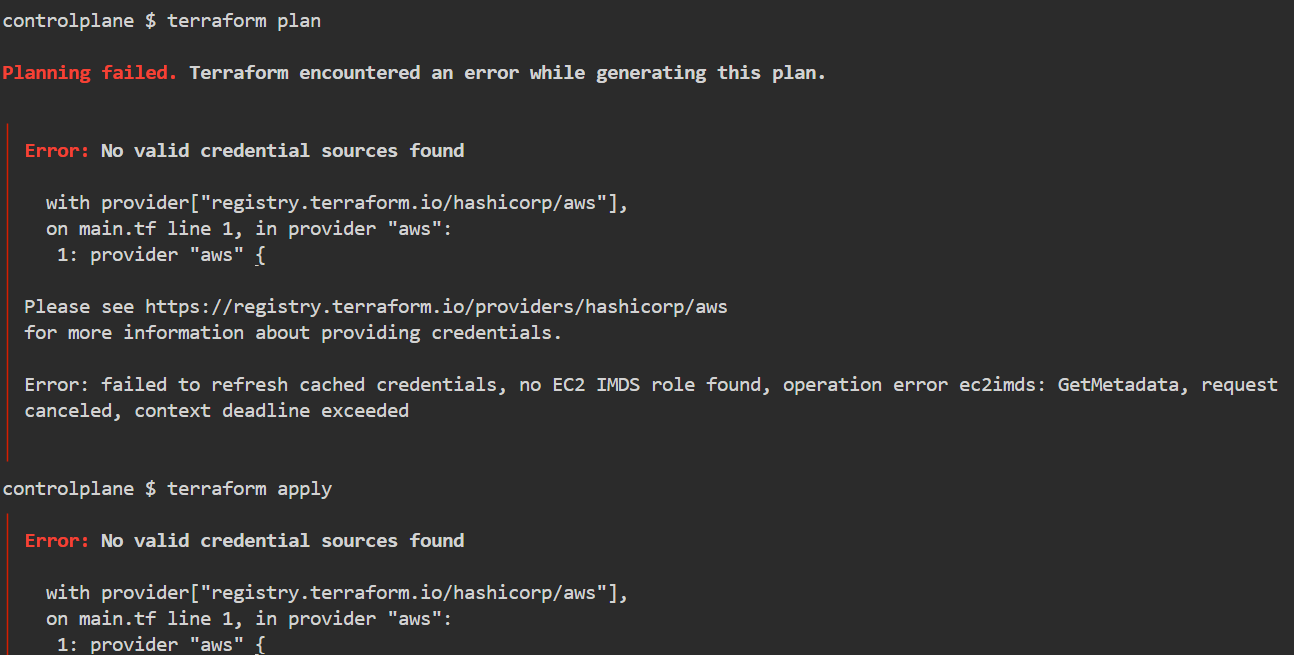


1. **Plan and apply the configuration:**

terraform plan

terraform apply

As I am not having to AWS platform it will throw an error similar in the below attachment :-



1. **Commit and push the code to GitHub:**

git add main.tf

git commit -m "Initial commit of AWS infrastructure as code"

git remote add origin https://github.com/itsme-aswindev/aws-infra.git

git push -u origin master