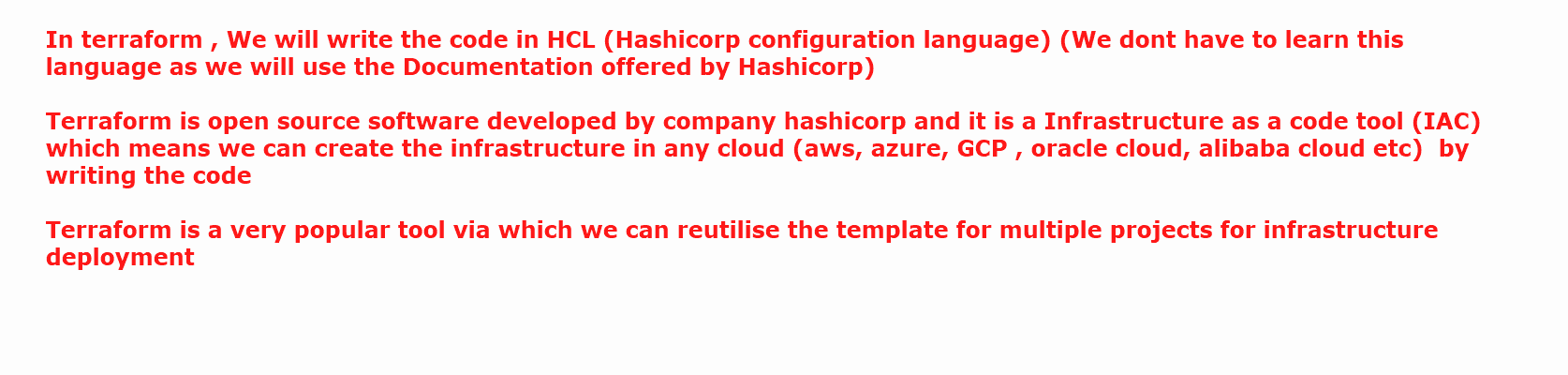
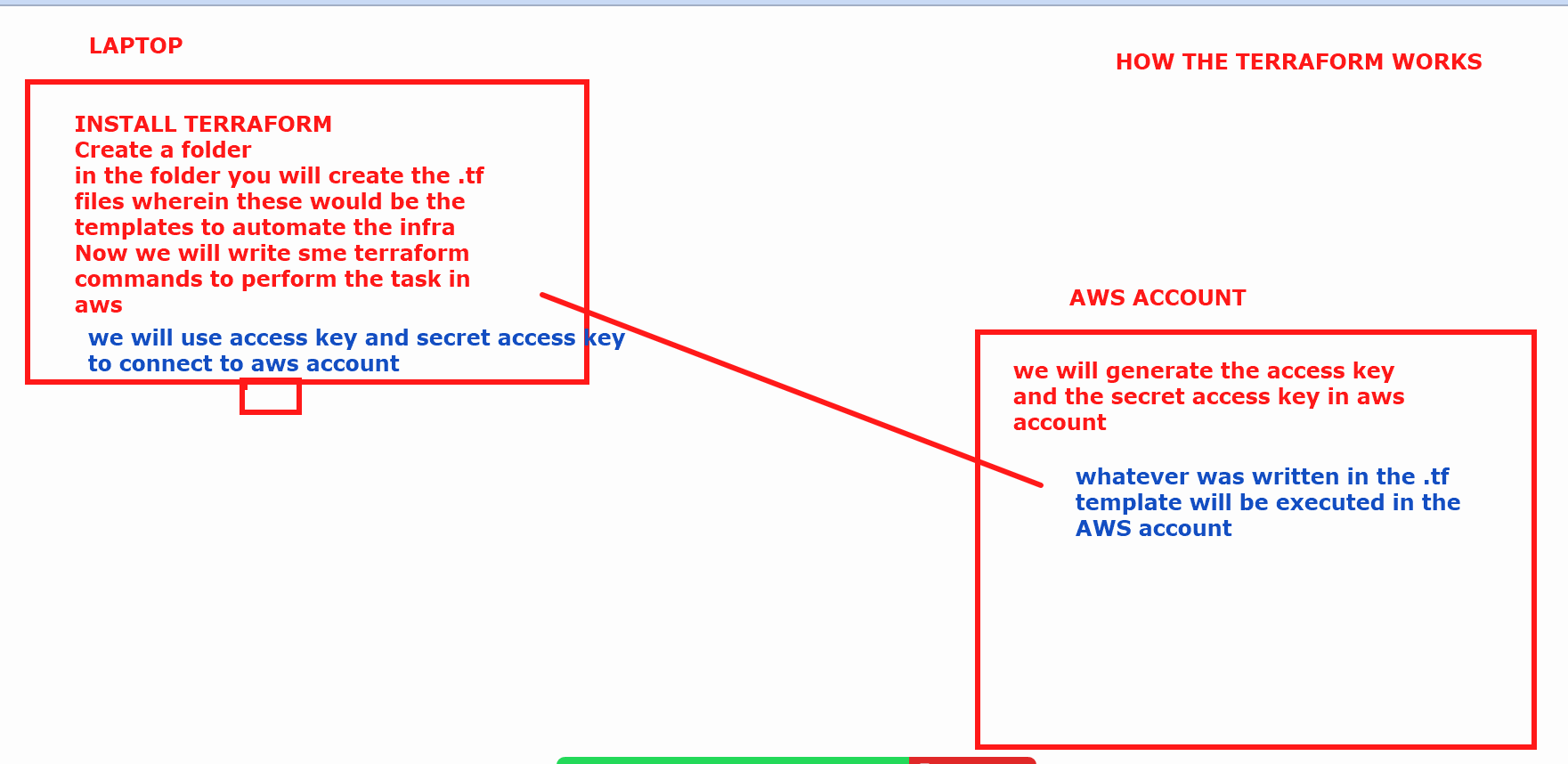
Terraform : Infrastructure as a code service (IAC) tool





What are the commands which we use in the terraform 👍

1. Terraform init : Init stands for initialise. When you run terraform init terraform is getting everything ready to start building your infrastructure.
2. Terraform plan : Creates an execution plan showing what terraform will do when you run the apply command . It helps you understand the changes that will made to your infrastructure.
3. Terraform apply : applies the changes required to reach the desired state of the configuration files. It will create the required infra mentioned in the .tf files
4. Terraform validate : validates the configuration files
5. Terraform show: displays information about the current state or a specific plan file
6. Terraform destroy : Destroys the infrastructure managed by terraform .

###

We will create a ec2 machine and we will connect machine to aws account to perform the tasks in aws .

(now this machine could be any machine …either your laptop , or any other machine present at any location)

## we will launch a ec2 ubuntu machine

Inside the machine

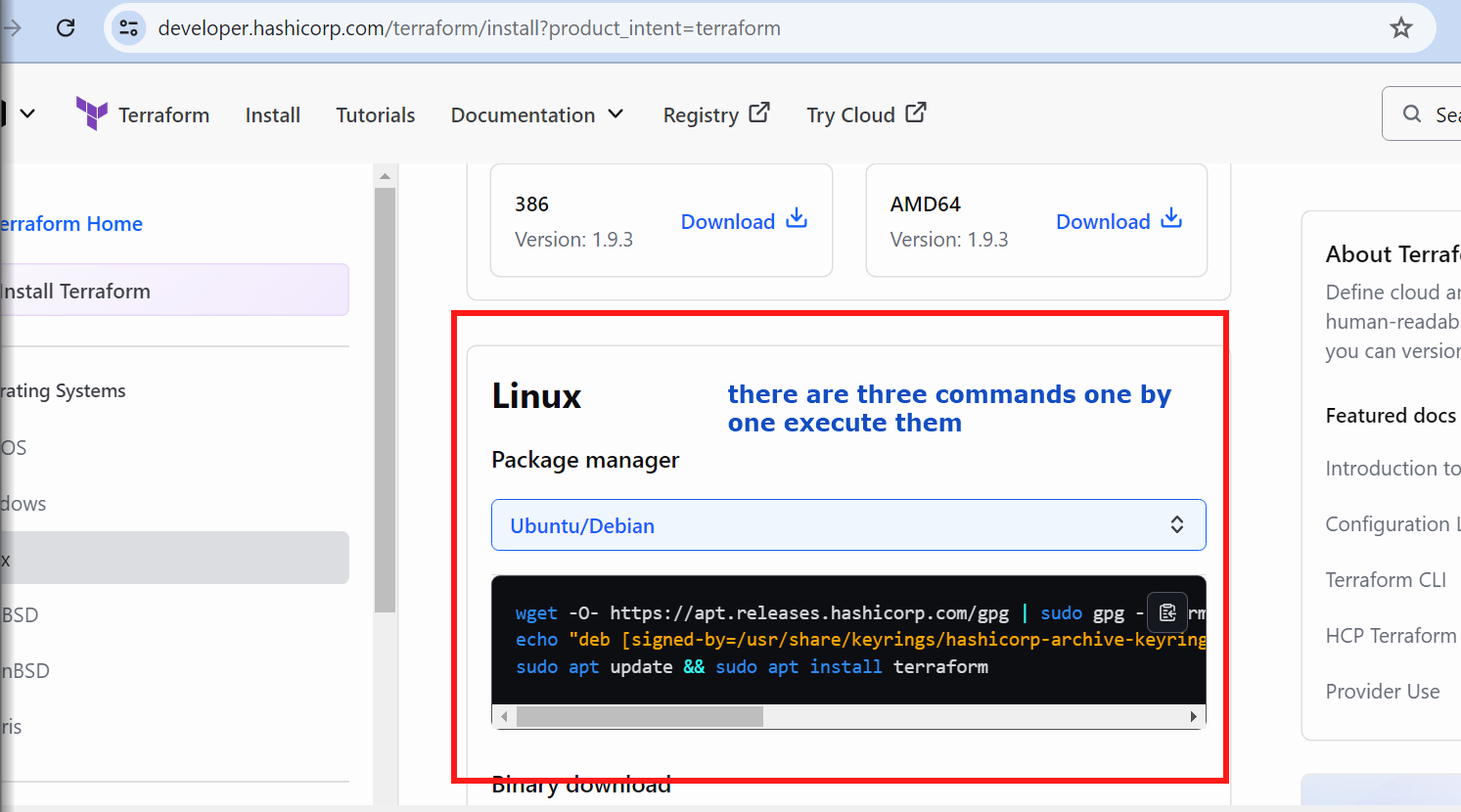
# sudo su

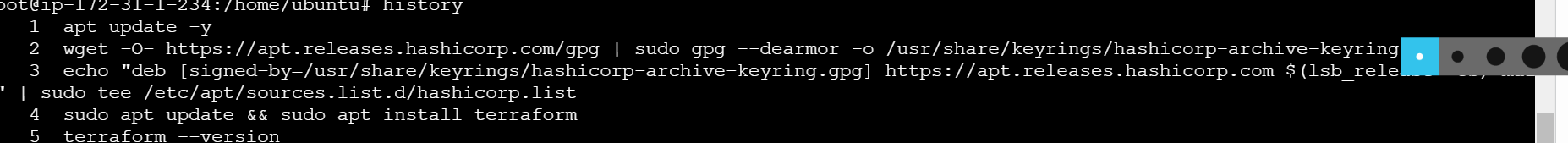
# apt update

Now we need to install terraform

Go to Terraform.io

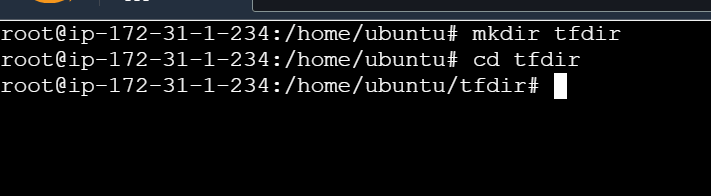
Click on install





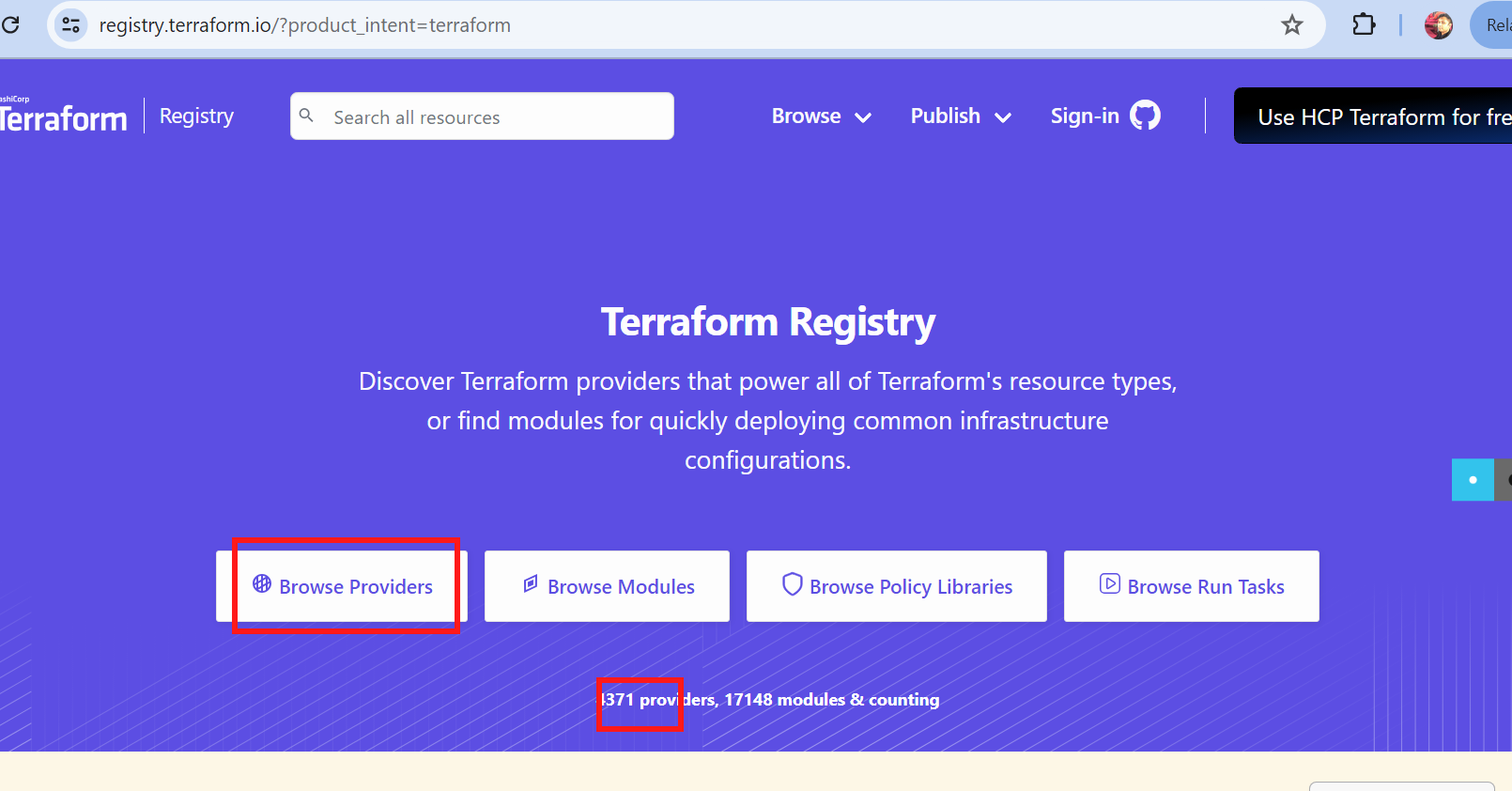
### After terraform is installed

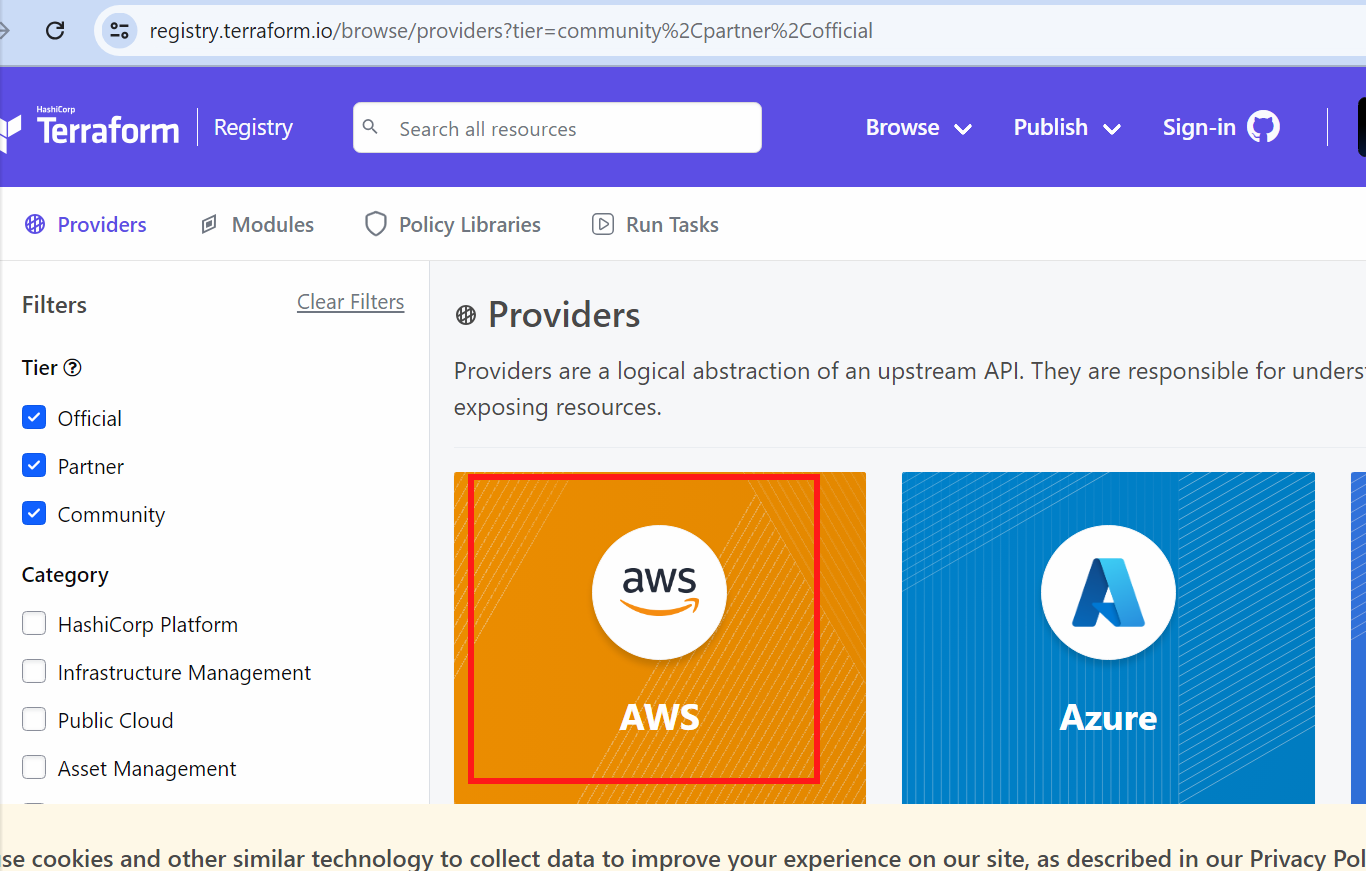
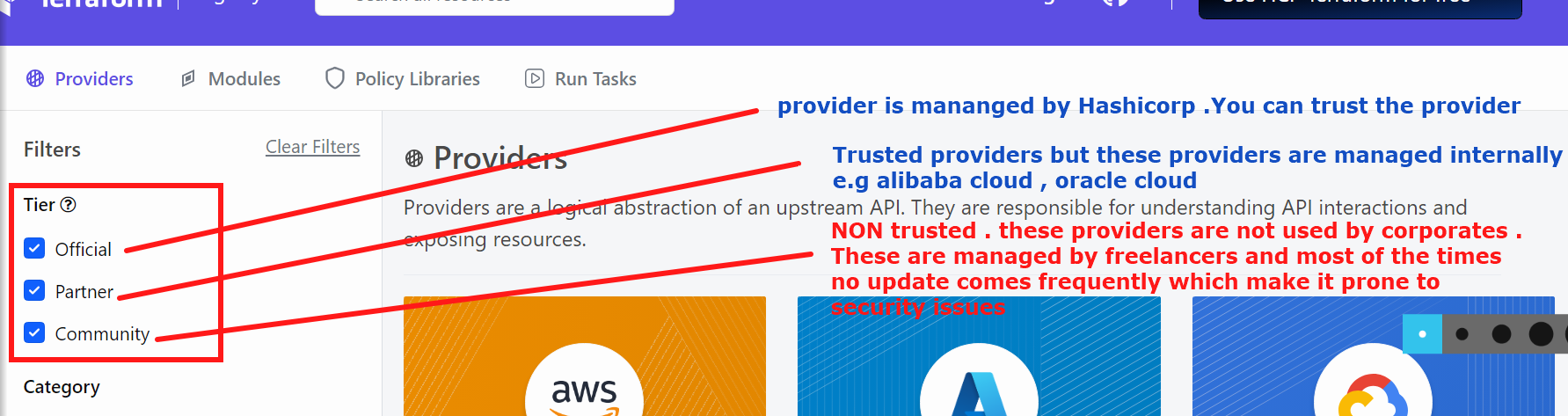
We will now create a new folder - directory

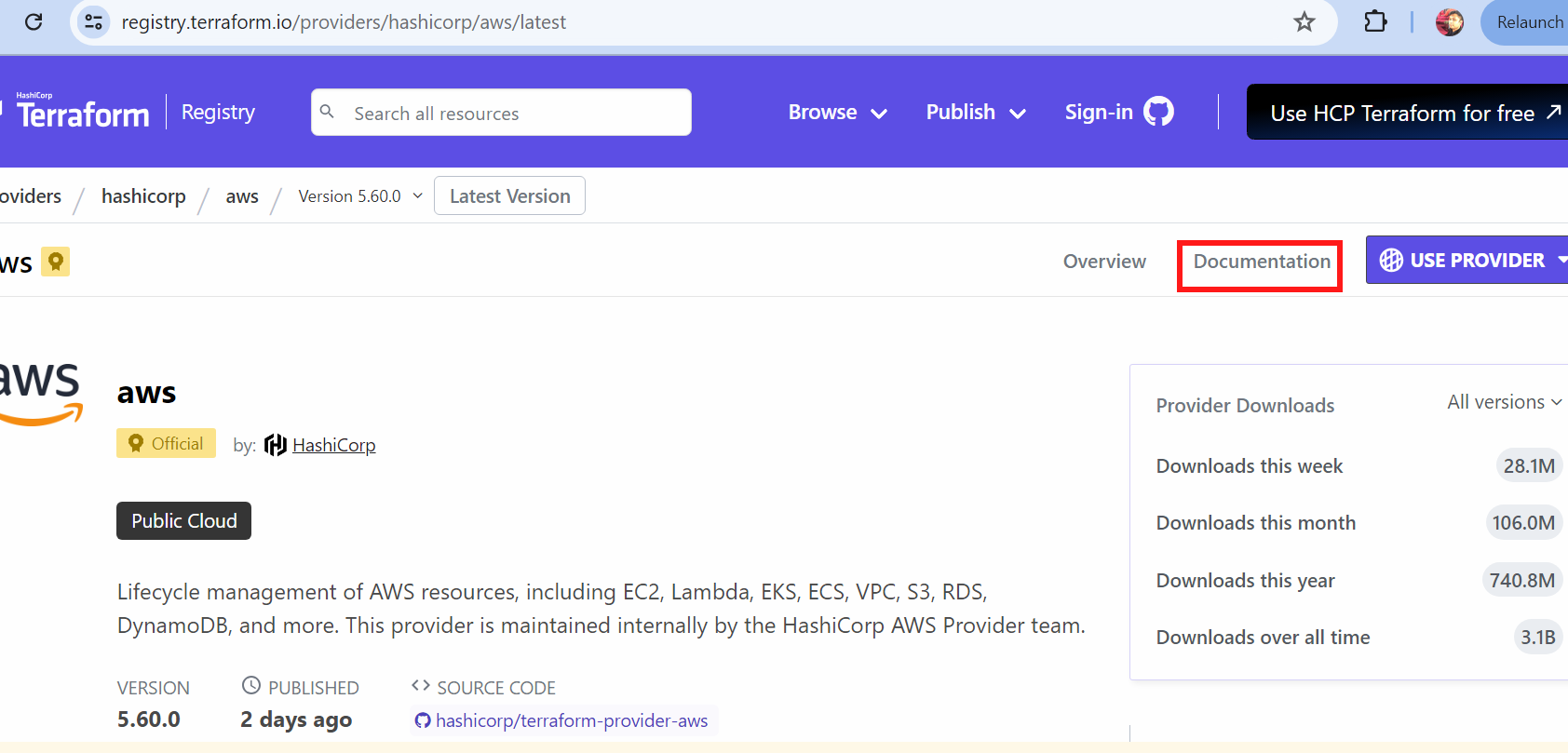


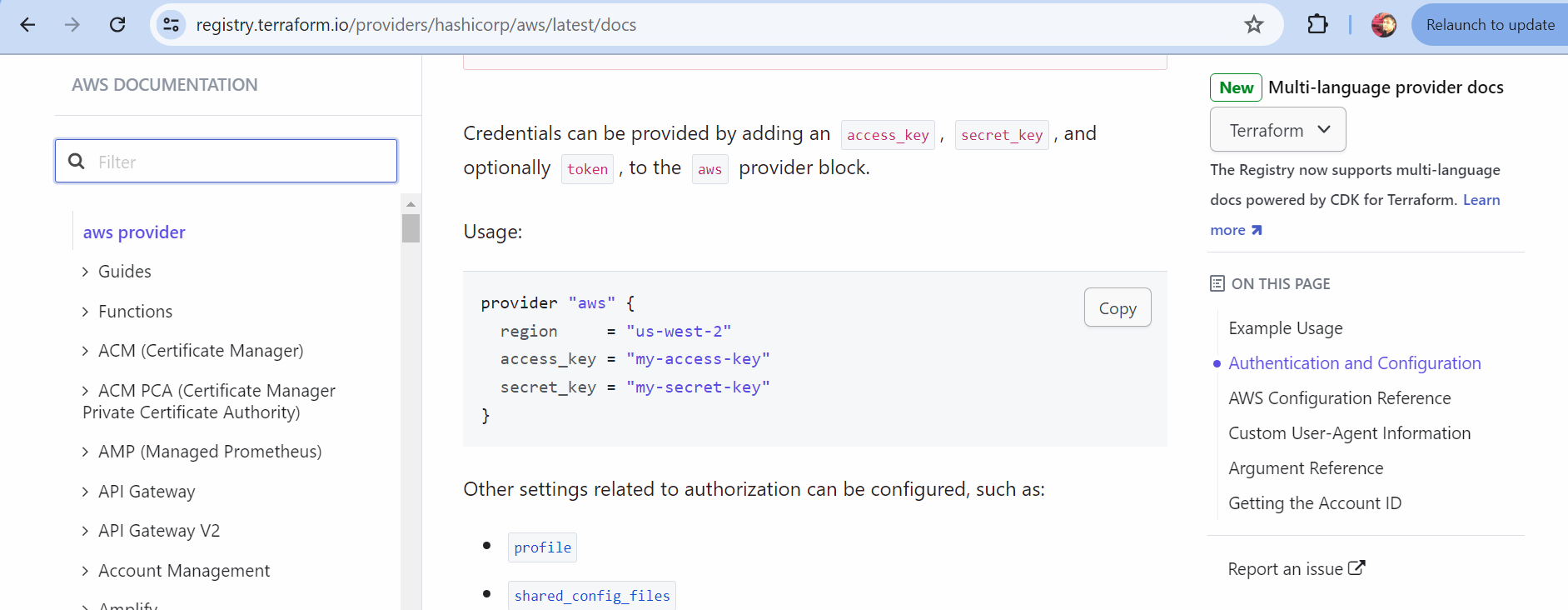
Inside this terraform directory i will write the scripts

Go to terraform.io -> click on registry



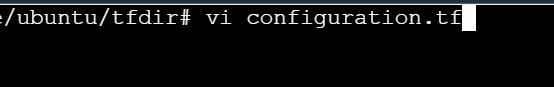


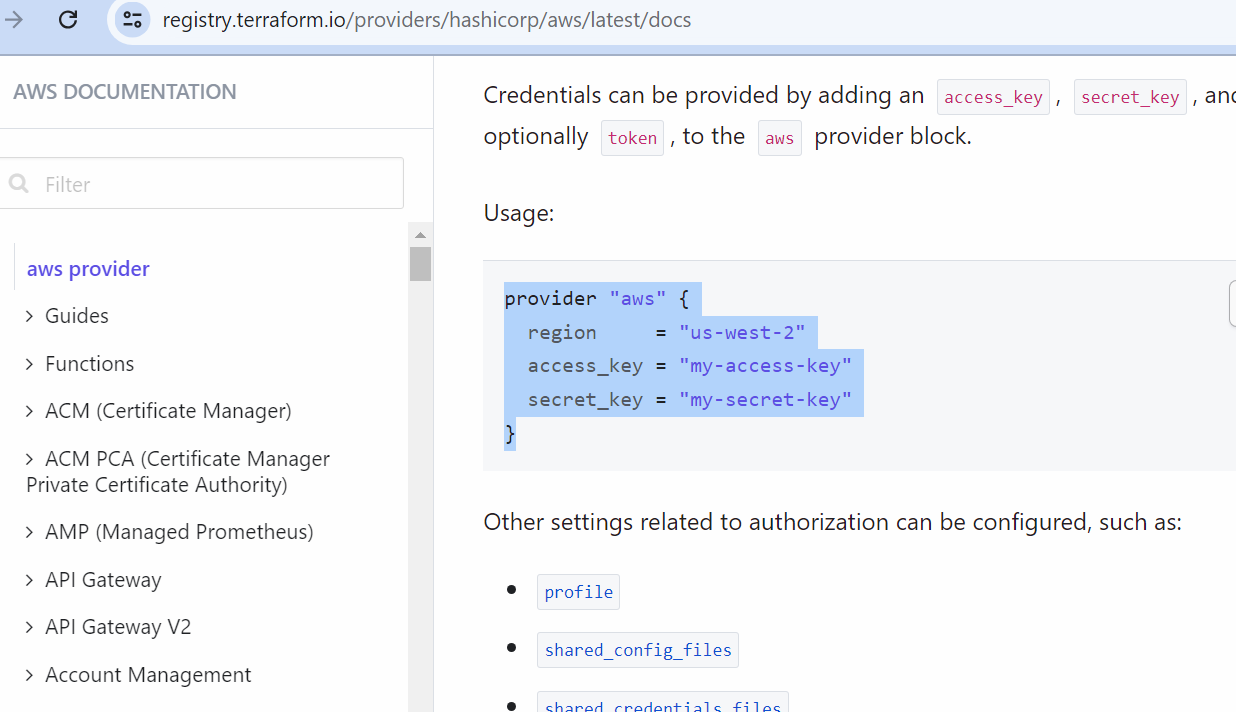


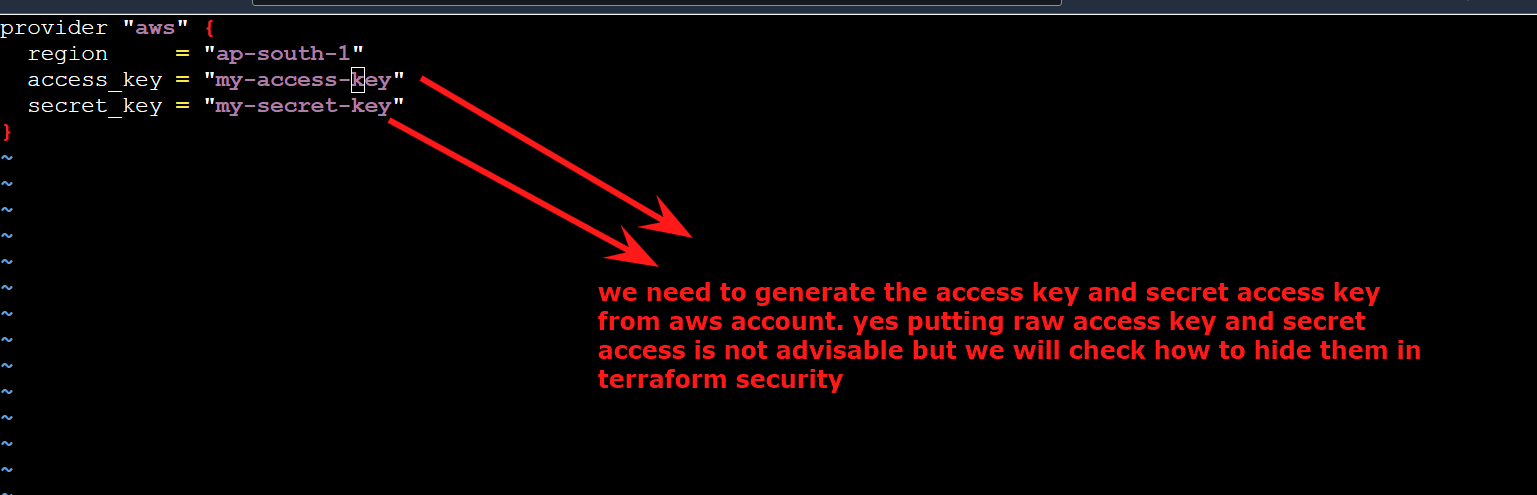


We will use this documentation to get the tasks done …

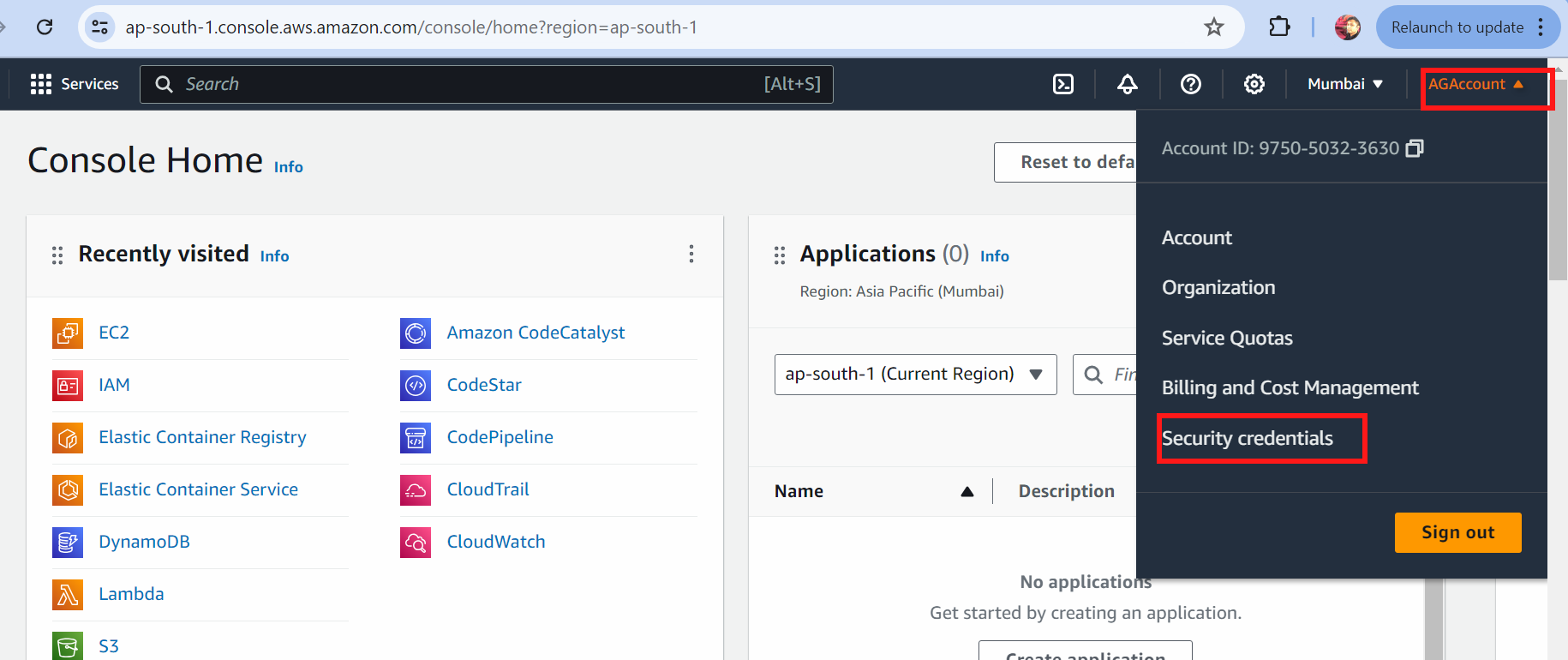
Now we will connect with the aws account by creating a configuration.tf file (any name)

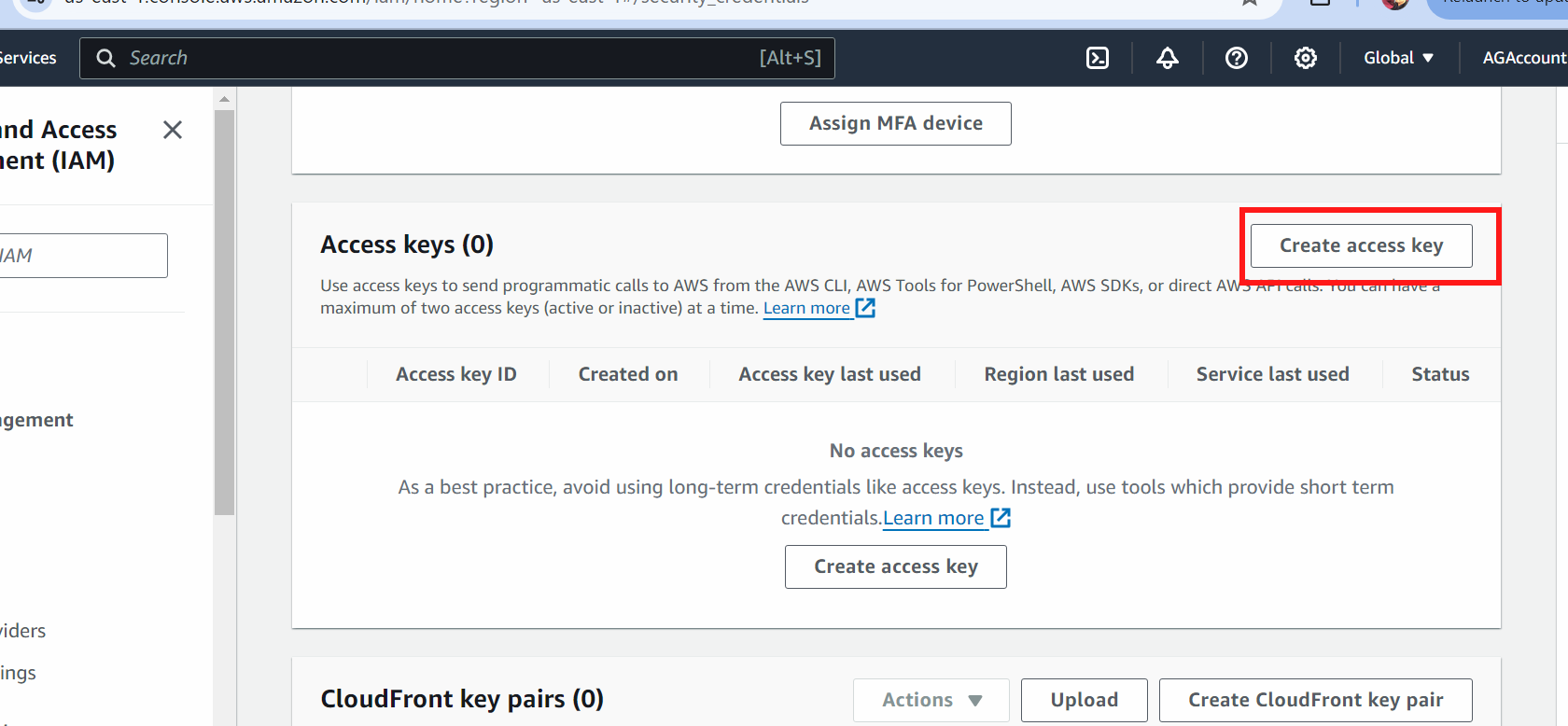


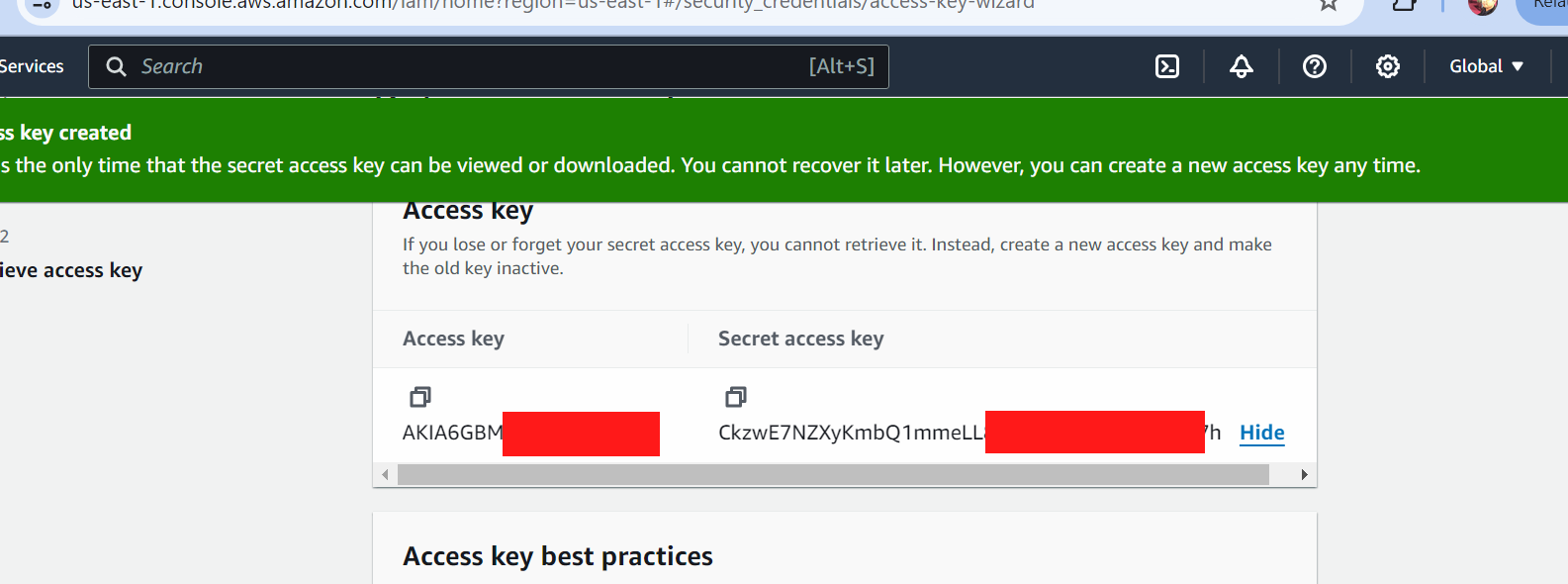




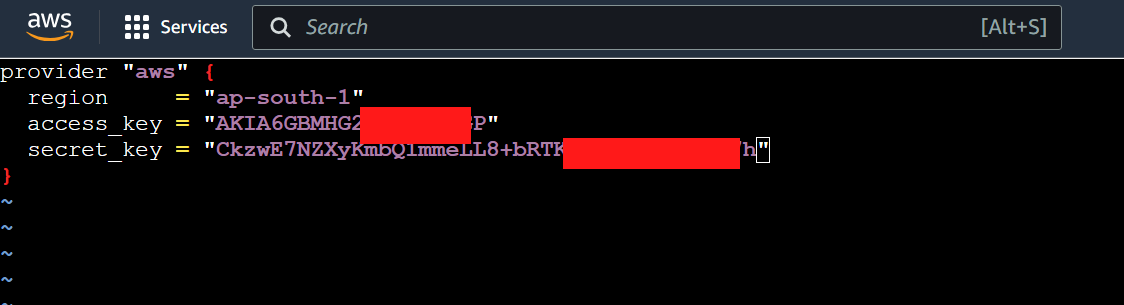
If you are using my account then use access key and secret access which I have shared otherwise if you are using your own aws account follow below method to generate access key and secret key:



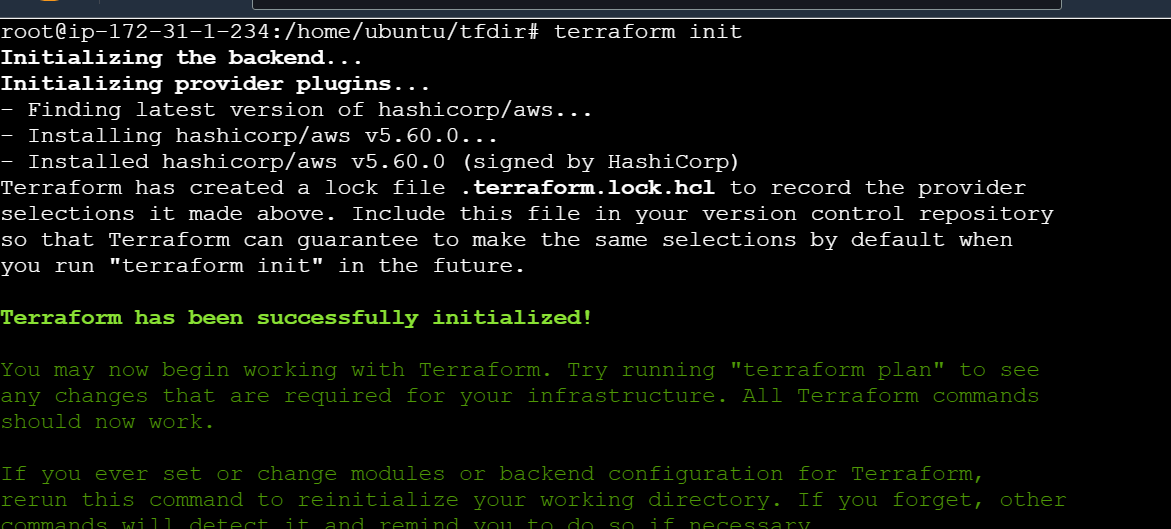




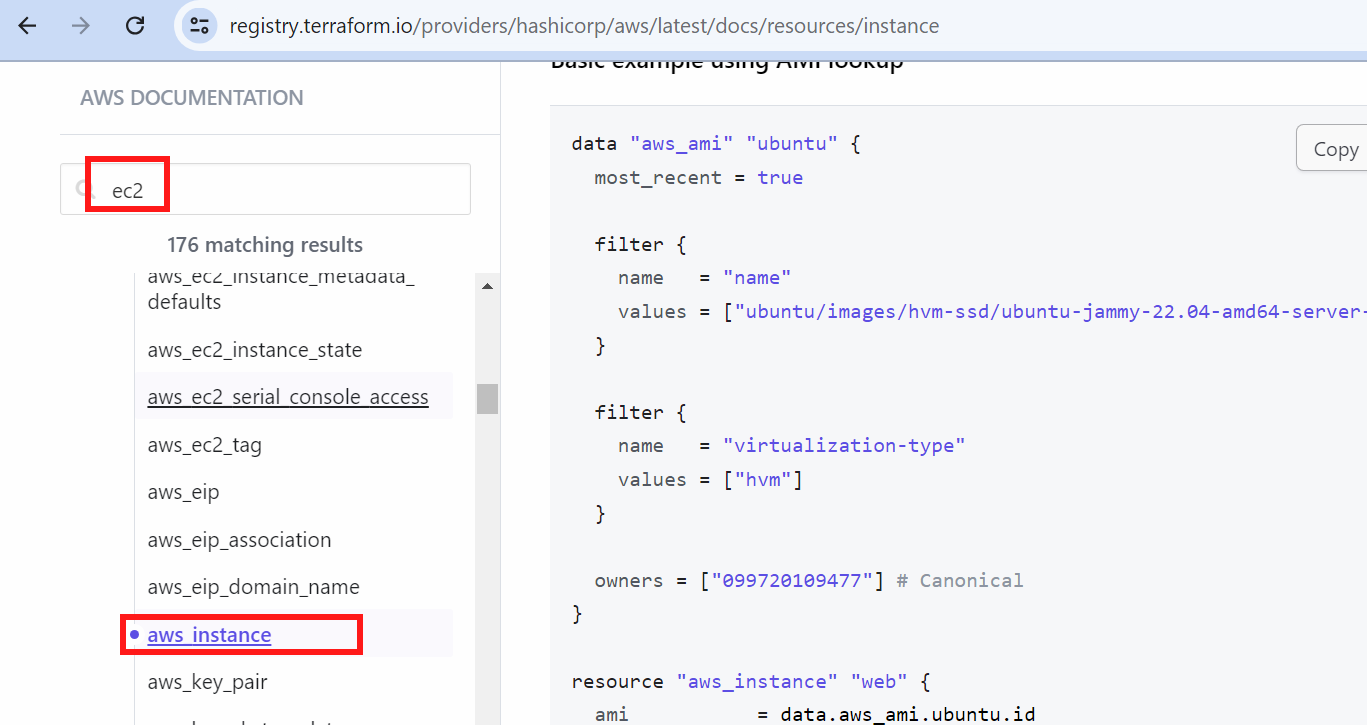
We will use these access key and secret access key in terraform tf configuration file

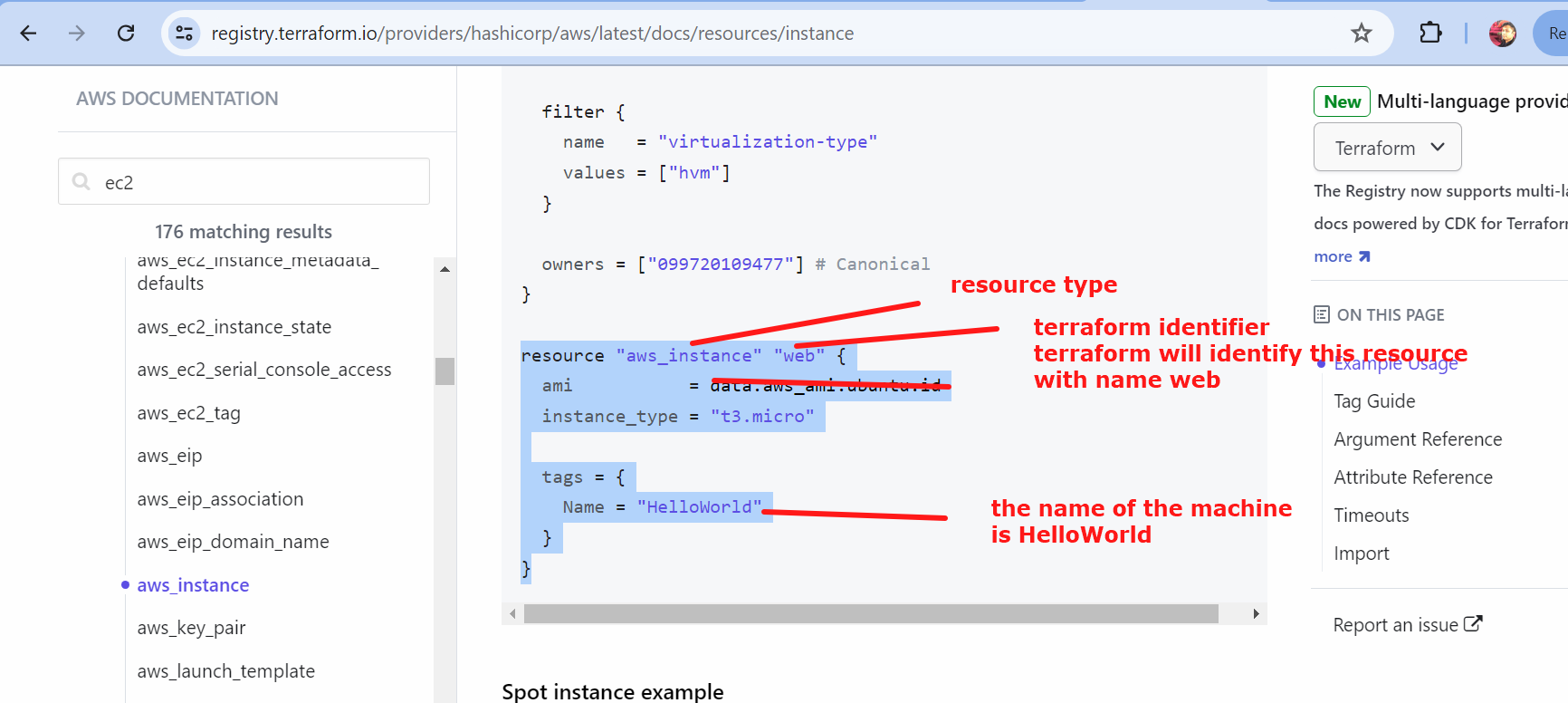






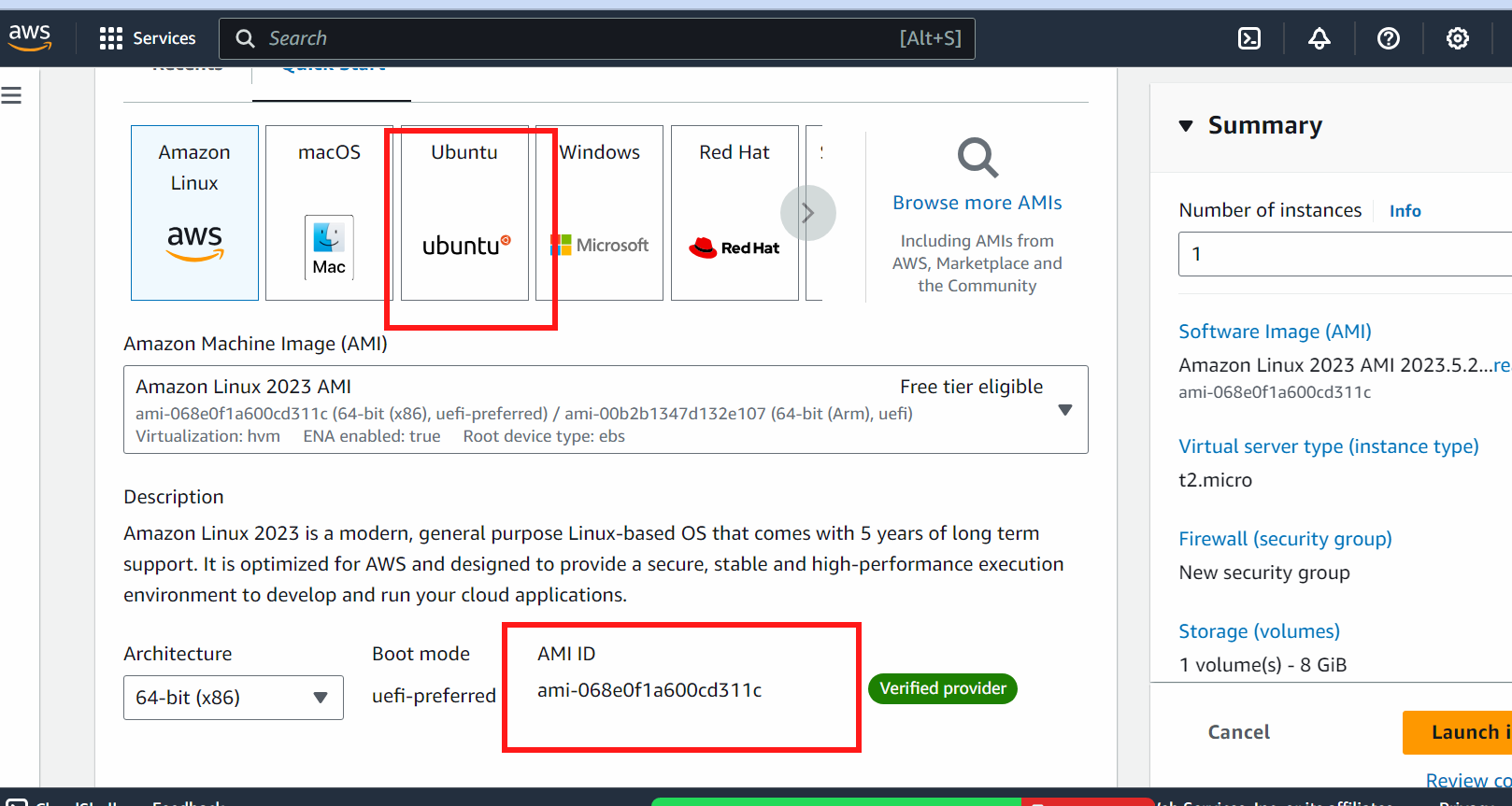
Now we will write the code to create ec2 machine via terraform







To find the ami…go to launch instance



(in mumbai region -> ubuntu -> ami id will be same for all accounts)

resource "aws\_instance" "web" {

ami = "ami-068e0f1a600cd311c" #amazon linux ami

instance\_type = "t3.medium"

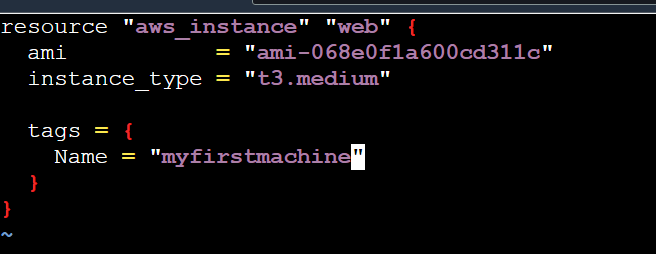
security\_groups = [“alltrafficsg”]

tags = {

Name = "myfirstmachine"

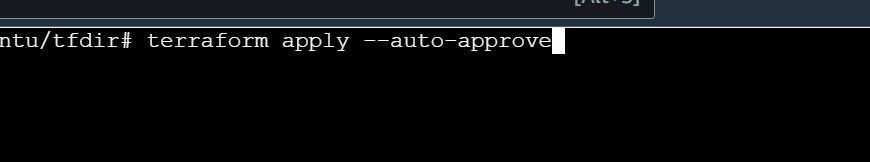
}

}



# terraform plan

# terraform apply --auto-approve



You can see the resource is created:



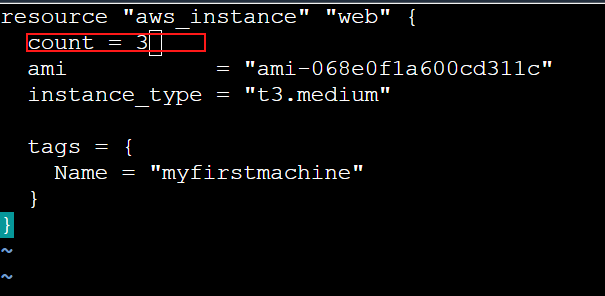
### i want to delete the resource created



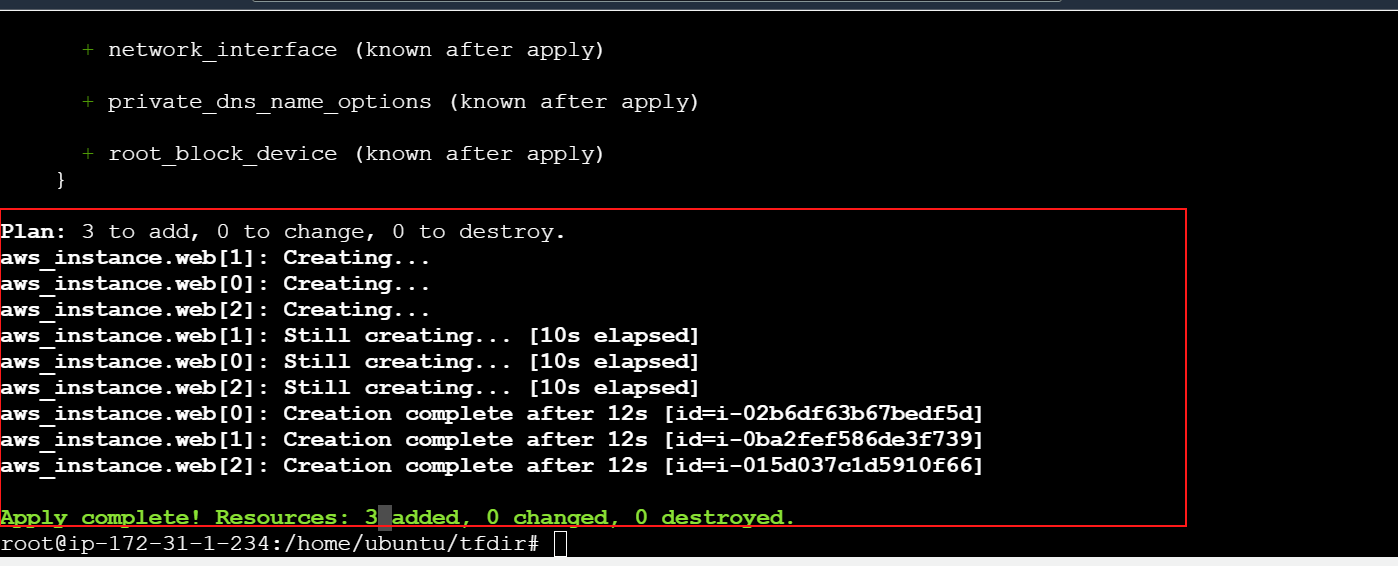
In enter value put yes

######################

## if i want to create 3 machines together



terraform apply --auto-approve

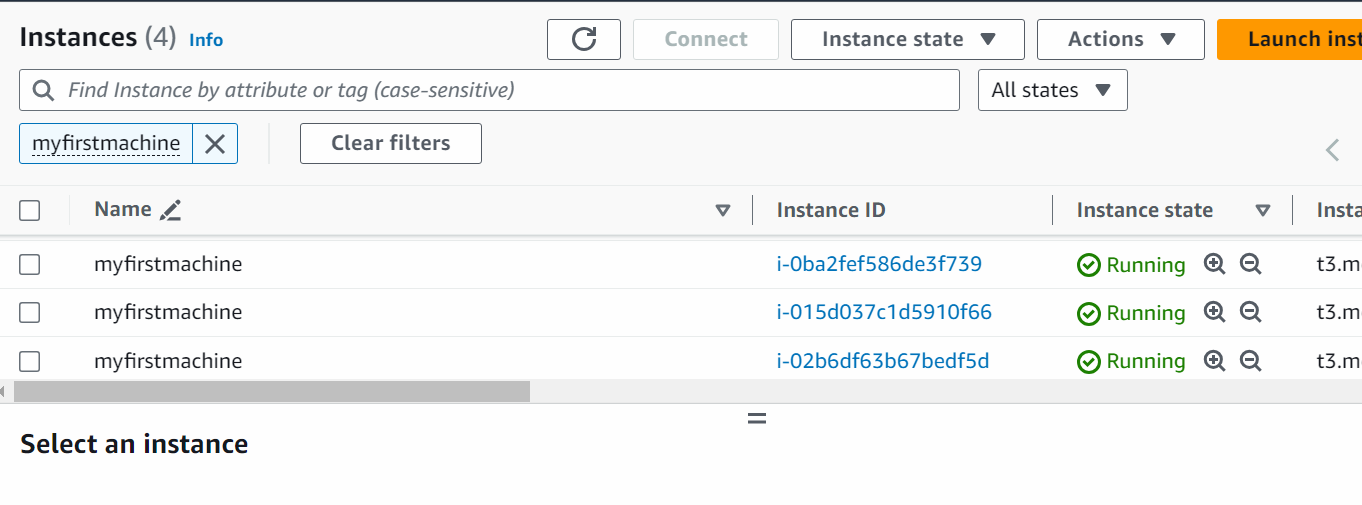


How the terraform will identify the machines:

Aws\_instance.web[0] - first machine

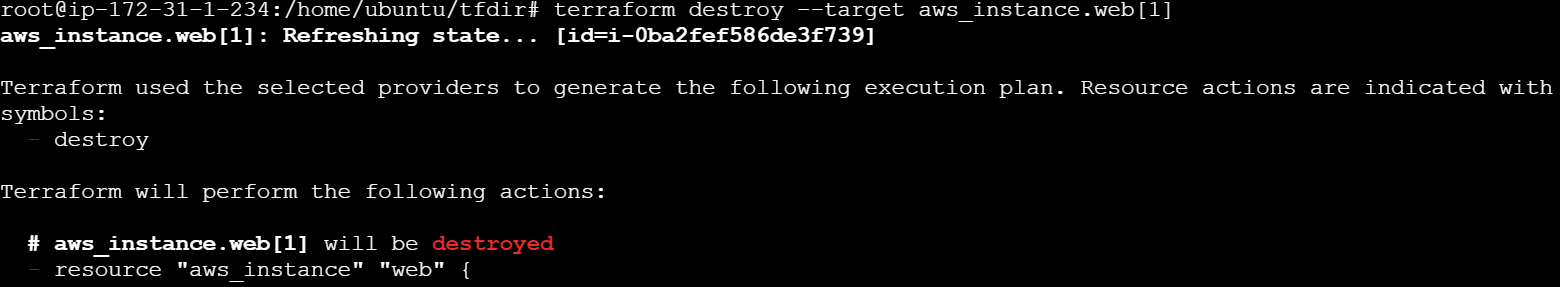
Aws\_instance.web[1] - second

Aws\_instance.web[3] third machine



Now if we want to only delete 2nd machine

terraform destroy --target aws\_instance.web[1]

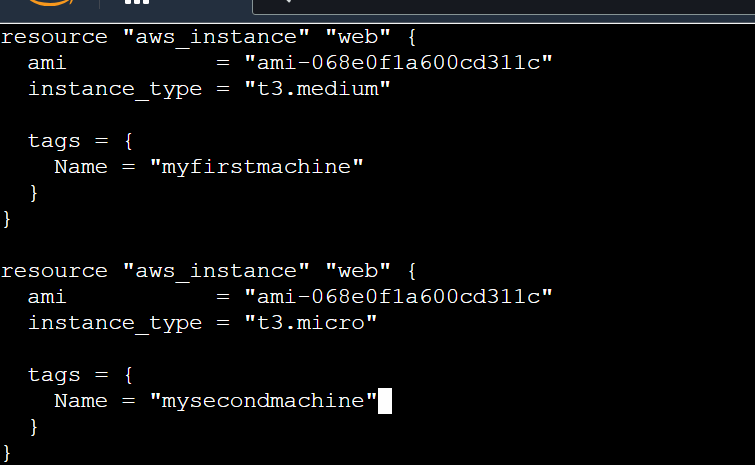


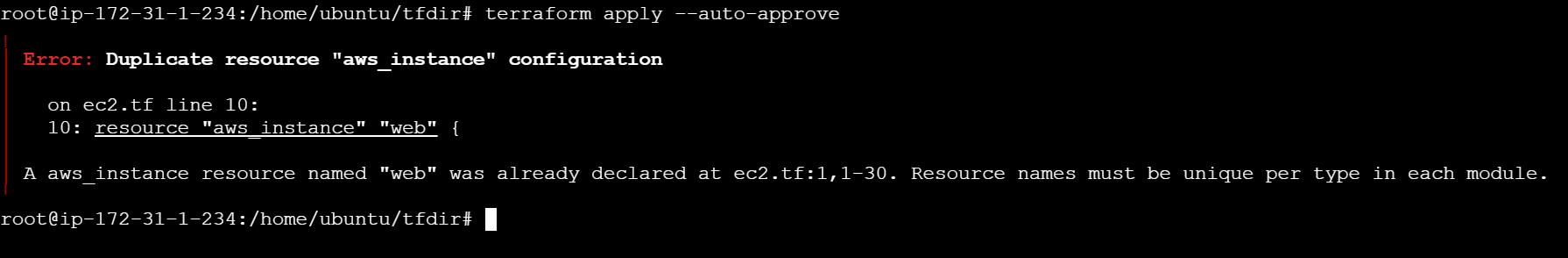


#terraform destroy

###

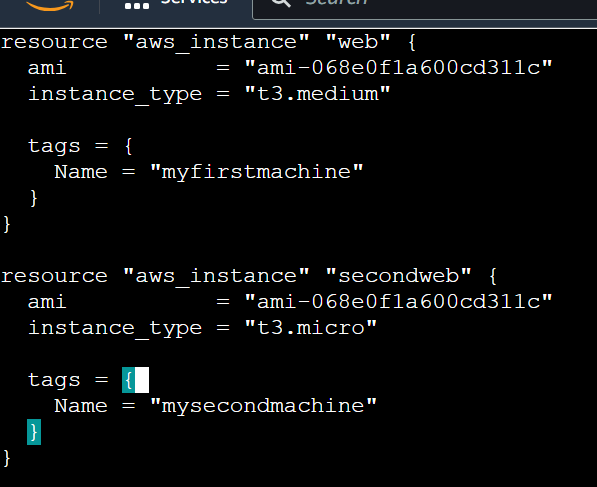
Lets check if we have have two terraform identifiers with the same name ?





ERROR WILL COME UP AS THE IMAGE ABOVE

NOW LETS CHANGE THE TERRAFORM IDENTIFIER



# terraform apply

You will see resource will create because terraform identifier name is different

