**Terraform**

**Terrafrom Commands:**

terraform init

terraform plan

terraform apply

terraform destroy

terraform fmt

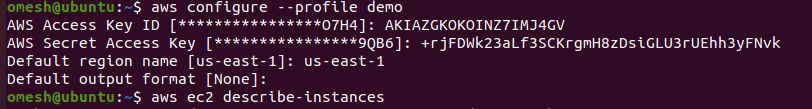
Install Visual Code Editor and Terraform plug-in

install Python, PIP and AWSCLI

Configure a demo profile:

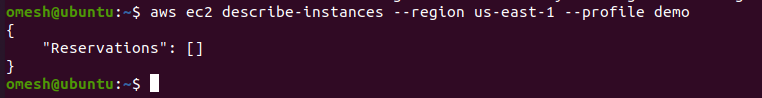
Create a profile named “demo” in IAM and provide it programming access with appropriate privilege to ensure aws account is safe.

omesh@ubuntu:~$ aws configure --profile demo



Check Profile:

omesh@ubuntu:~$ aws ec2 describe-instances --region us-east-1 --profile demo

(No Resources at the moment)

Create an ec2 instance in the editor and create a file as testec2.tf in the Linux Machine.

omesh@ubuntu:~$ nano testec2.tf and save below configuration.

provider "aws" {

profile = "demo"

region = "us-east-1"

}

resource "aws\_instance" "demo\_instance" {

ami = "ami-013f17f36f8b1fefb"

instance\_type = "t2.micro"

tags = {

Name = "DemoInstance"

}

}

Initialize Terraform:

omesh@ubuntu:~$ terraform init

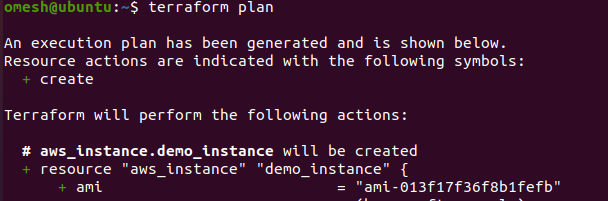
(It will initialize the terraform environment successfully and we can see the required plug-in in the directory .terraform for the respective prodivers e.g aws here)



Terraform Plan:

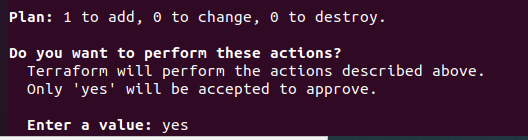
It will create and share a plan where + means resource will be created and –means resource will be deleted.

omesh@ubuntu:~$ terraform plan

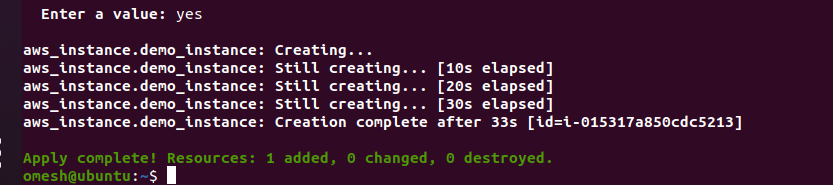


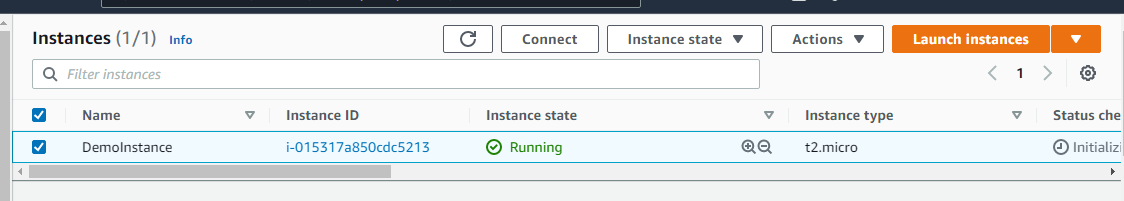
omesh@ubuntu:~$ terraform apply

It will ask to confirm with “yes” to configure these resources or not?



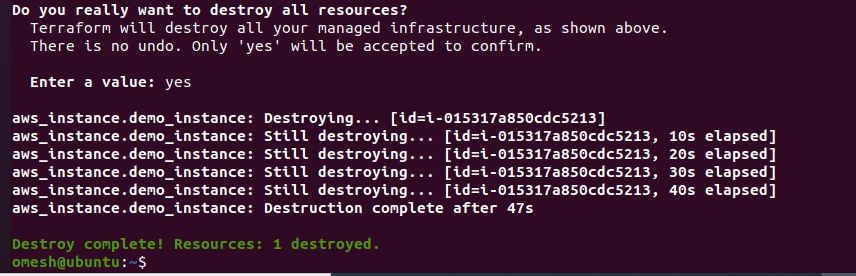
It will create an instance and can be verified on the AWS Console:

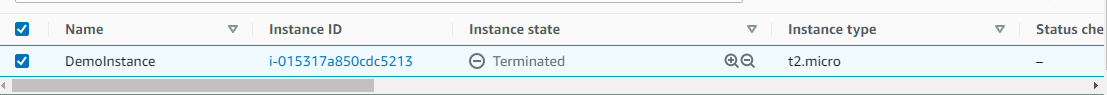




Run Terrafrom Destroy:

omesh@ubuntu:~$ terraform destroy





Create Variables.tf:

Plan.out will help create a plan and deploy with being asked for confirmation.

omesh@DESKTOP-0MQTKM3:~$ terraform plan -out plan.out