Module 8 – Terraform

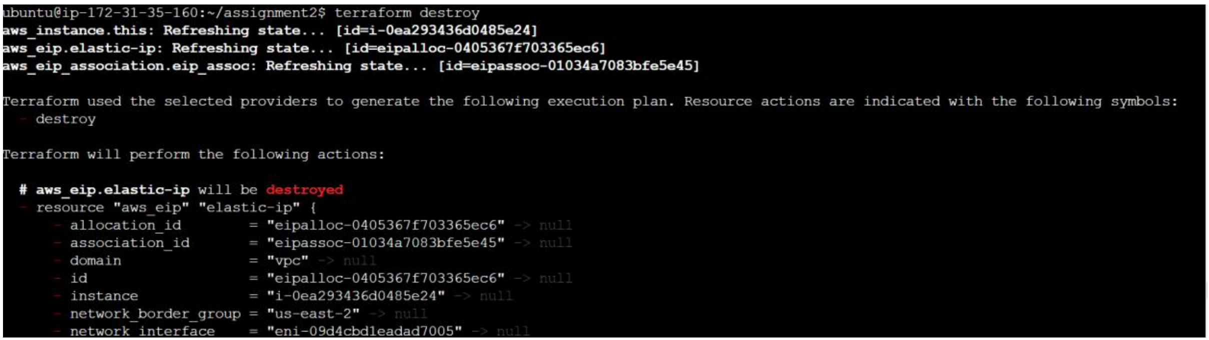
Assignment-3

1. Destroy the previous deployment

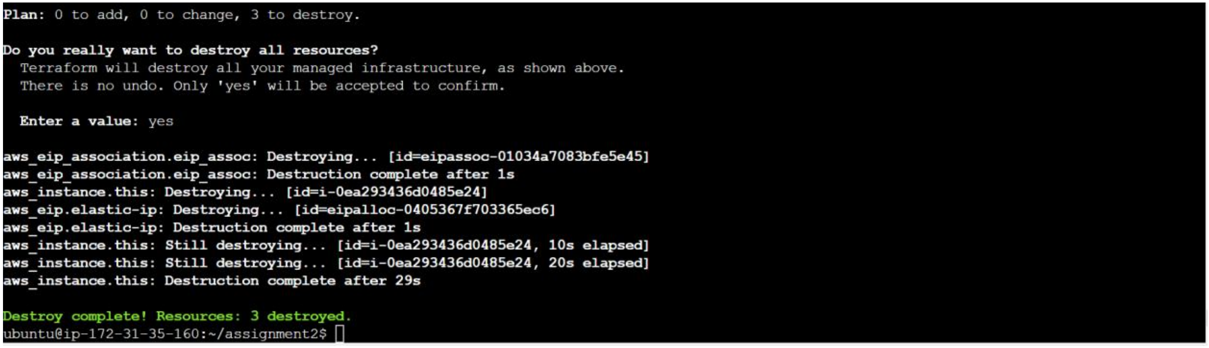
2. Create 2 EC2 instances in Ohio and N.Virginia respectively

3. Rename Ohio’s instance to ‘hello-ohio’ and Virginia’s instance to ‘hello-virginia’

- Destoyed the previous deployment with terraform destroy command

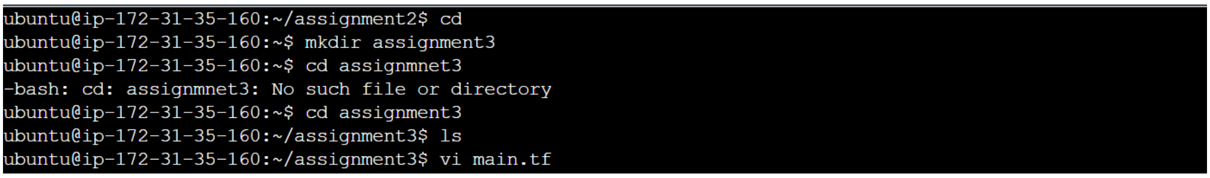


```
ubuntu@ip-172-31-35-160:~/assignment2$ terraform destroy
aws_instance.this: Refreshing state... [id=i-0ea293436d0485e24]
aws_eip.elastic-ip: Refreshing state... [id=eipalloc-0405367f703365ec6]
aws_eip_association.eip_assoc: Refreshing state... [id=eipassoc-01034a7083bfe5e45]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  - destroy

Terraform will perform the following actions:

  # aws_eip.elastic-ip will be destroyed
  - resource "aws_eip" "elastic-ip" {
      - allocation_id        = "eipalloc-0405367f703365ec6" -> null
      - association_id       = "eipassoc-01034a7083bfe5e45" -> null
      - domain               = "vpc" -> null
      - id                   = "eipalloc-0405367f703365ec6" -> null
      - instance             = "i-0ea293436d0485e24" -> null
      - network_border_group = "us-east-2" -> null
      - network_interface    = "eni-09d4cbd1eadad7005" -> null
```

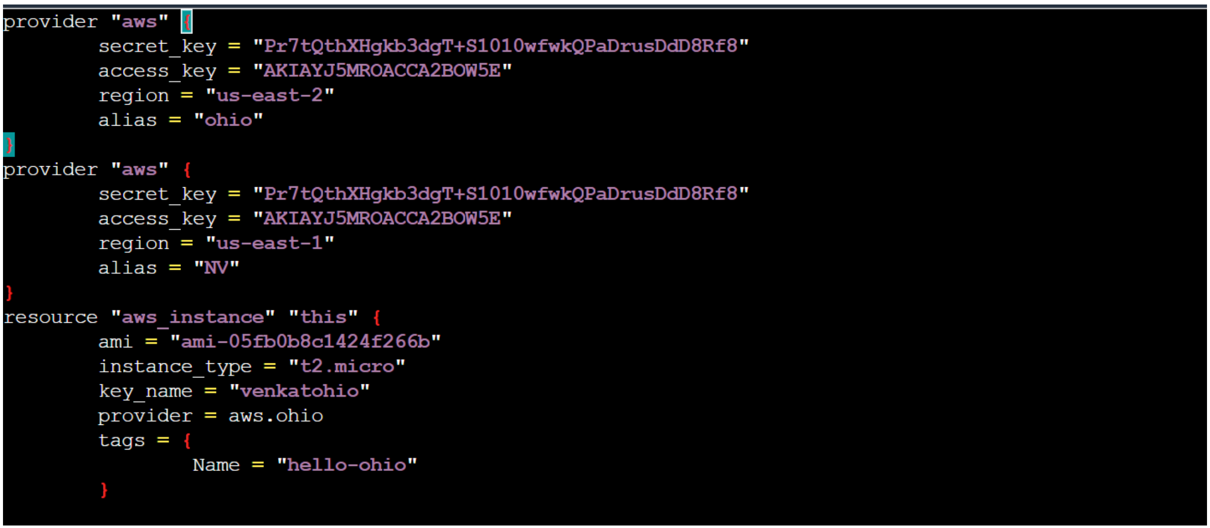


```
Plan: 0 to add, 0 to change, 3 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_eip_association.eip_assoc: Destroying... [id=eipassoc-01034a7083bfe5e45]
aws_eip_association.eip_assoc: Destruction complete after 1s
aws_instance.this: Destroying... [id=i-0ea293436d0485e24]
aws_eip.elastic-ip: Destroying... [id=eipalloc-0405367f703365ec6]
aws_eip.elastic-ip: Destruction complete after 1s
aws_instance.this: Still destroying... [id=i-0ea293436d0485e24, 10s elapsed]
aws_instance.this: Still destroying... [id=i-0ea293436d0485e24, 20s elapsed]
aws_instance.this: Destruction complete after 29s

Destroy complete! Resources: 3 destroyed.
ubuntu@ip-172-31-35-160:~/assignment2$ 
```

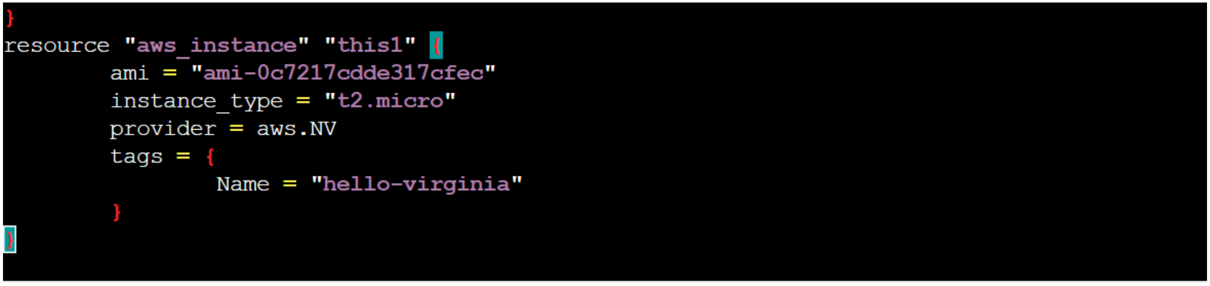
- Then created a new direcoty named assignment3 and written a main.tf with a hcl script to create instances in ohio and N.Virginia regions and renamed them to “hello-ohio” and “hello-Virginia respectively



```
ubuntu@ip-172-31-35-160:~/assignment2$ cd
ubuntu@ip-172-31-35-160:~$ mkdir assignment3
ubuntu@ip-172-31-35-160:~$ cd assignmnet3
-bash: cd: assignmnet3: No such file or directory
ubuntu@ip-172-31-35-160:~$ cd assignment3
ubuntu@ip-172-31-35-160:~/assignment3$ ls
ubuntu@ip-172-31-35-160:~/assignment3$ vi main.tf
```

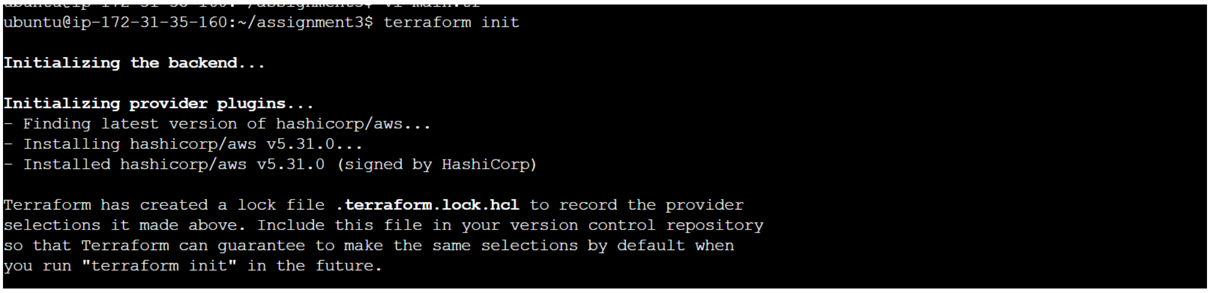


```
provider "aws" {
        secret_key = "Pr7tQthXHgkb3dgT+S1010wfwkQPaDrusDdD8Rf8"
        access_key = "AKIAYJ5MROACCA2BOW5E"
        region = "us-east-2"
        alias = "ohio"
}
provider "aws" {
        secret_key = "Pr7tQthXHgkb3dgT+S1010wfwkQPaDrusDdD8Rf8"
        access_key = "AKIAYJ5MROACCA2BOW5E"
        region = "us-east-1"
        alias = "NV"
}
resource "aws_instance" "this" {
        ami = "ami-05fb0b8c1424f266b"
        instance_type = "t2.micro"
        key_name = "venkatohio"
        provider = aws.ohio
        tags = {
                Name = "hello-ohio"
        }
```



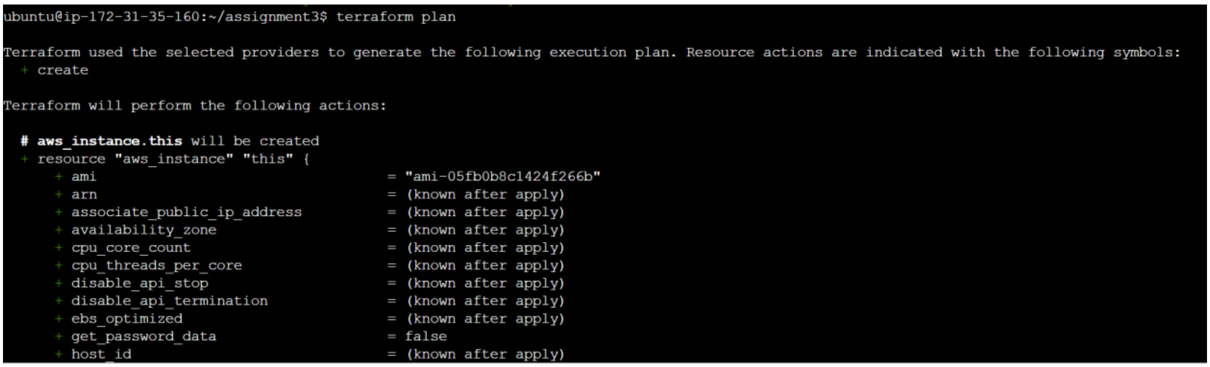
```
}
resource "aws_instance" "this1" {
        ami = "ami-0c7217cdde317cfec"
        instance_type = "t2.micro"
        provider = aws.NV
        tags = {
                Name = "hello-virginia"
        }
}
```

Initialised terraform

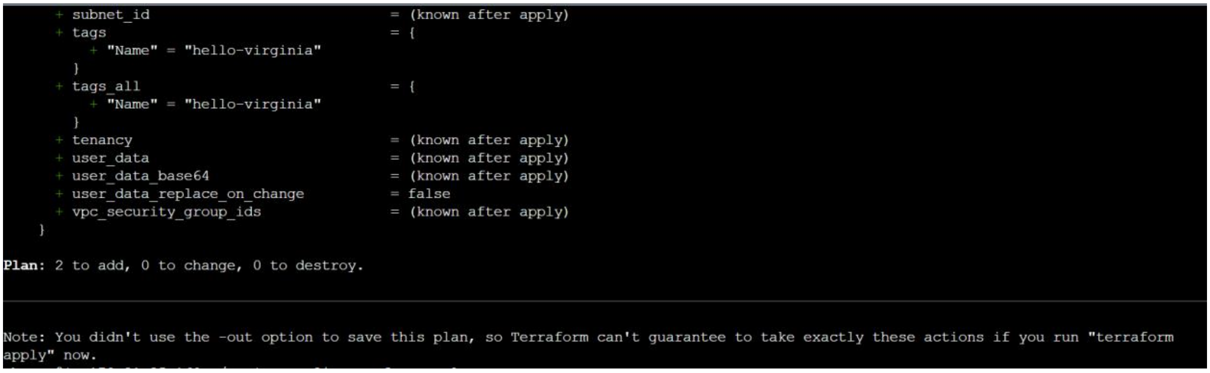


```
ubuntu@ip-172-31-35-160:~/assignment3$ vi main.tf
ubuntu@ip-172-31-35-160:~/assignment3$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.
```

Performed terraform plan

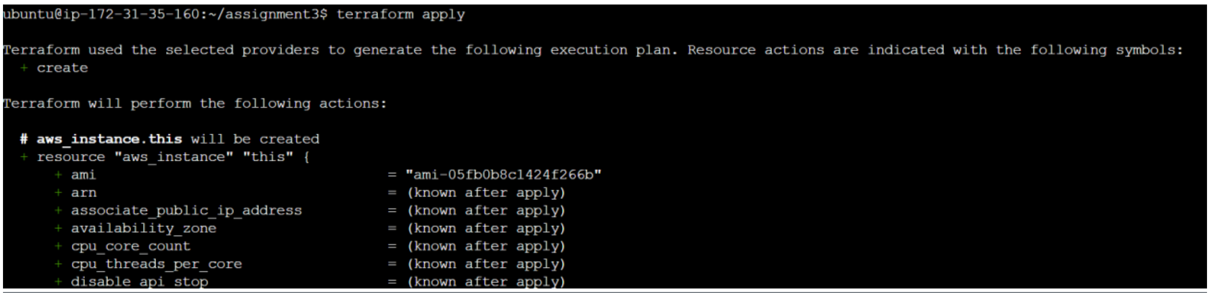


```
ubuntu@ip-172-31-35-160:~/assignment3$ terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.this will be created
  + resource "aws_instance" "this" {
      + ami                                  = "ami-05fb0b8c1424f266b"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
```

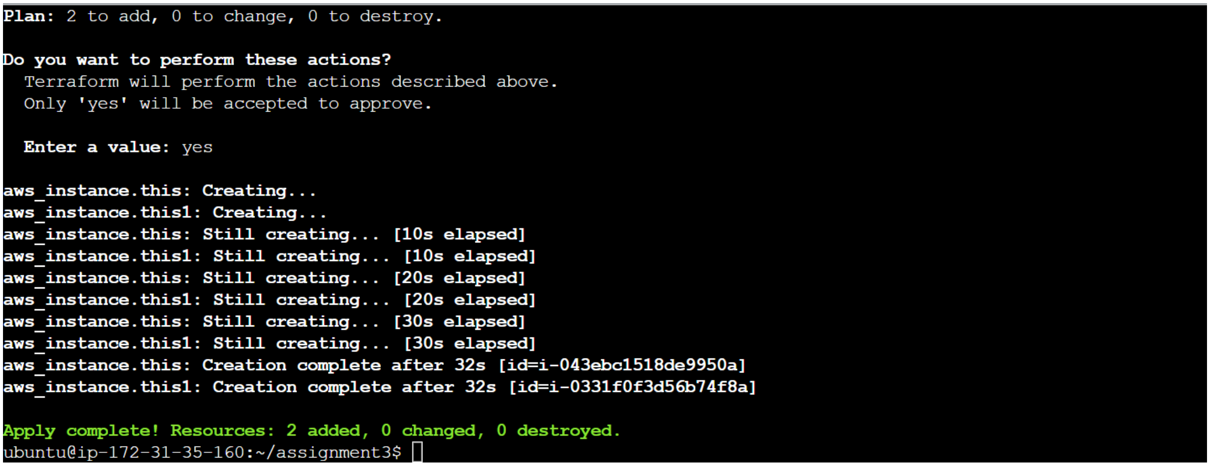


```
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "hello-virginia"
        }
      + tags_all                             = {
          + "Name" = "hello-virginia"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }

Plan: 2 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform
apply" now.
```

Applied changes with terraform aplly command



```
ubuntu@ip-172-31-35-160:~/assignment3$ terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.this will be created
  + resource "aws_instance" "this" {
      + ami                                  = "ami-05fb0b8c1424f266b"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
```



```
Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.this: Creating...
aws_instance.this1: Creating...
aws_instance.this: Still creating... [10s elapsed]
aws_instance.this1: Still creating... [10s elapsed]
aws_instance.this: Still creating... [20s elapsed]
aws_instance.this1: Still creating... [20s elapsed]
aws_instance.this: Still creating... [30s elapsed]
aws_instance.this1: Still creating... [30s elapsed]
aws_instance.this: Creation complete after 32s [id=i-043ebc1518de9950a]
aws_instance.this1: Creation complete after 32s [id=i-0331f0f3d56b74f8a]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-35-160:~/assignment3$
```