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'

Solution:-

$ sudo terraform destroy

```
ubuntu@terraform-server:~/tcode/assignment2$ sudo terraform destroy
aws_instance.assignment-2: Refreshing state... [id=i-0a40a8e13ab43fb31]
aws_eip.eip: Refreshing state... [id=eipalloc-06180f6ec27311cc4]
aws_eip_association.eip_assoc: Refreshing state... [id=eipassoc-0cf037c5186e6ff92]

Terraform used the selected providers to generate the following execution plan. Resource acti
  - destroy

Terraform will perform the following actions:

  # aws_eip.eip will be destroyed
  - resource "aws_eip" "eip" {
      - allocation_id        = "eipalloc-06180f6ec27311cc4" -> null
      - association_id       = "eipassoc-0cf037c5186e6ff92" -> null
```

```
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-0cf037c5186e6ff92]
aws_eip_association.eip_assoc: Destruction complete after 1s
aws_instance.assignment-2: Destroying... [id=i-0a40a8e13ab43fb31]
aws_eip.eip: Destroying... [id=eipalloc-06180f6ec27311cc4]
aws_eip.eip: Destruction complete after 2s
aws_instance.assignment-2: Still destroying... [id=i-0a40a8e13ab43fb31, 10s elapsed]
aws_instance.assignment-2: Still destroying... [id=i-0a40a8e13ab43fb31, 20s elapsed]
aws_instance.assignment-2: Destruction complete after 30s

Destroy complete! Resources: 3 destroyed.
ubuntu@terraform-server:~/tcode/assignment2$
```

$ cd .. && sudo mkdir assignment3 && cd assignment3

```
ubuntu@terraform-server:~/tcode/assignment2$ cd .. && sudo mkdir assignment3 && cd assignment3
ubuntu@terraform-server:~/tcode/assignment3$
```

$ sudo vi provider.tf

```
provider "aws" {
      region = "us-east-1"
      access_key = "AKIA3XNV7HVVOZH64X44"
      secret_key =
"ISTXT0XOPP9sJfxlrmM6RpZvvVDdQIw4eMmtd
tWE"
}
provider "aws" {
      region = "us-east-2"
      access_key = "AKIA3XNV7HVVOZH64X44"
      secret_key =
"ISTXT0XOPP9sJfxlrmM6RpZvvVDdQIw4eMmtd
tWE"
}
```

$ sudo vi main.tf

```
resource "aws_instance" "assignment-3-NV" {
        provider = aws.NV
        ami = "ami-053b0d53c279acc90"
        instance_type = "t2.micro"
                key_name = "Common-25-08-2023-
09-47"
        tags = {
        Name = "hello-virginia"
        }
}

resource "aws_instance" "assignment-3-OHIO"
{
        provider = aws.ohio
        ami = "ami-024e6efaf93d85776"
        instance_type = "t2.micro"
        key_name = "terraform_key"
        tags = {
        Name = "hello-ohio"
        }
}
```

```
ubuntu@terraform-server:~/tcode/assignment3$ cat main.tf
resource "aws_instance" "assignment-3-NV" {
        provider = aws.NV
        ami = "ami-053b0d53c279acc90"
        instance_type = "t2.micro"
        key_name = "Common-25-08-2023-09-47"
        tags = {
        Name = "hello-virginia"
        }
}

resource "aws_instance" "assignment-3-OHIO" {
        provider = aws.ohio
        ami = "ami-024e6efaf93d85776"
        instance_type = "t2.micro"
        key_name = "terraform_key"
        tags = {
        Name = "hello-ohio"
        }
}
```

$ sudo terraform init

```
ubuntu@terraform-server:~/tcode/assignment3$ sudo terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.14.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

$ sudo terraform plan

```
ubuntu@terraform-server:~/tcode/assignment3$ sudo terraform plan

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

Terraform will perform the following actions:

  # aws_instance.assignment-3-NV will be created
  + resource "aws_instance" "assignment-3-NV" {
      + ami                                  = "ami-053b0d53c279acc90"
      + 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)
```

```
  # aws_instance.assignment-3-OHIO will be created
  + resource "aws_instance" "assignment-3-OHIO" {
      + ami                                  = "ami-024e6efaf93d85776"
      + 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)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (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 guarante
 to take exactly these actions if you run "terraform apply" now.
```

$ sudo terraform apply

```
ubuntu@terraform-server:~/tcode/assignment3$ sudo terraform apply

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

Terraform will perform the following actions:

  # aws_instance.assignment-3-NV will be created
  + resource "aws_instance" "assignment-3-NV" {
      + ami                                  = "ami-053b0d53c279acc90"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
```

```
  Enter a value: yes

aws_instance.assignment-3-NV: Creating...
aws_instance.assignment-3-OHIO: Creating...
aws_instance.assignment-3-NV: Still creating... [10s elapsed]
aws_instance.assignment-3-OHIO: Still creating... [10s elapsed]
aws_instance.assignment-3-NV: Still creating... [20s elapsed]
aws_instance.assignment-3-OHIO: Still creating... [20s elapsed]
aws_instance.assignment-3-NV: Still creating... [30s elapsed]
aws_instance.assignment-3-NV: Creation complete after 32s [id=i-063eda6ff79e70232]
aws_instance.assignment-3-OHIO: Still creating... [30s elapsed]
aws_instance.assignment-3-OHIO: Creation complete after 32s [id=i-038555c4114024b86

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
ubuntu@terraform-server:~/tcode/assignment3$
```

* Instances created present in AWS console with their tags:-

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
hello-ohio	i-038555c4114024b86	Running	t2.micro	Initializing	No alarms	us-east-2a	ec2-3-139-67-227.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
hello-virginia	i-063eda6ff79e70232	Running	t2.micro	Initializing	No alarms	us-east-1c	ec2-3-85-31-174