Terraform Assignment - 3

You have been asked to:
- Destroy the previous deployment
- Create 2 EC2 instances in Ohio and N.Virginia respectively
- Rename Ohio's instance to 'hello-ohio' and Virginia's instance to 'hello-virginia'
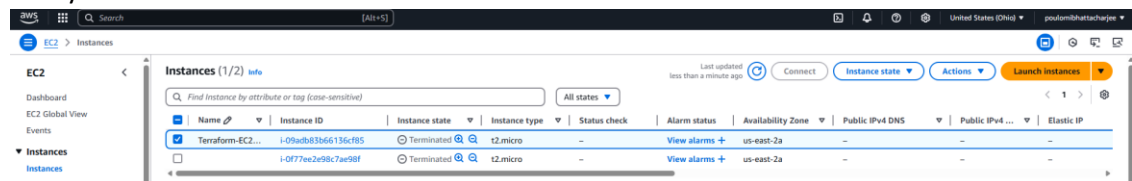
Solution:

1) Run Terraform Destroy: terraform destroy

```
aws_eip.example: Destroying... [id=eipalloc-05ca384efa1179919]
aws_eip.example: Destruction complete after 3s
aws_instance.example: Destroying... [id=i-09adb83b66136cf85]
aws_instance.example: Still destroying... [id=i-09adb83b66136cf85, 10s elapsed]
aws_instance.example: Still destroying... [id=i-09adb83b66136cf85, 20s elapsed]
aws_instance.example: Still destroying... [id=i-09adb83b66136cf85, 30s elapsed]
aws_instance.example: Still destroying... [id=i-09adb83b66136cf85, 40s elapsed]
aws_instance.example: Still destroying... [id=i-09adb83b66136cf85, 50s elapsed]
aws_instance.example: Still destroying... [id=i-09adb83b66136cf85, 1m0s elapsed]
aws_instance.example: Destruction complete after 1m2s

Destroy complete! Resources: 2 destroyed.
ubuntu@ip-172-31-7-35:~/terraform-project$
```

i-0c2ef8ee473810c4e (my-terraform)

PublicIPs: 3.110.54.55   PrivateIPs: 172.31.7.35

2) Verify :

3) Modify the main.tf file to Include the following Terraform configuration for creating EC2 instances in Ohio and N. Virginia:

```
# Define the AWS provider for both regions
provider "aws" {
  alias  = "ohio"
  region = "us-east-2"
}

provider "aws" {
  alias  = "virginia"
  region = "us-east-1"
}

# Create EC2 instance in Ohio
resource "aws_instance" "ohio_instance" {
  provider      = aws.ohio
  ami           = "ami-0cb91c7de36eed2cb"
  instance_type = "t2.micro"

  tags = {
    Name = "hello-ohio"
  }
}

# Create EC2 instance in N. Virginia
resource "aws_instance" "virginia_instance" {
  provider      = aws.virginia
  ami           = "ami-04b4f1a9cf54c11d0"
  instance_type = "t2.micro"

  tags = {
    Name = "hello-virginia"
  }
}
}
```

4) Initialize Terraform:

```
ubuntu@ip-172-31-7-35:~/terraform-project$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.91.0

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.
ubuntu@ip-172-31-7-35:~/terraform-project$
```

i-0c2ef8ee473810c4e (my-terraform)

PublicIPs: 3.110.54.55   PrivateIPs: 172.31.7.35

5) Plan:

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



Note: You didn't use the -out option to save this
ubuntu@ip-172-31-7-35:~/terraform-project$
```

i-0c2ef8ee473810c4e (my-terraform)

PublicIPs: 3.110.54.55   PrivateIPs: 172.31.7.35

6) Apply:

```
aws_instance.ohio_instance: Creating...
aws_instance.virginia_instance: Creating...
aws_instance.ohio_instance: Still creating... [10s elapsed]
aws_instance.virginia_instance: Still creating... [10s elapsed]
aws_instance.ohio_instance: Creation complete after 15s [id=i-09fcf47e3eb6fadc8]
aws_instance.virginia_instance: Creation complete after 15s [id=i-05fc8a9a8f3d1717e]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-7-35:~/terraform-project$
```

i-0c2ef8ee473810c4e (my-terraform)

PublicIPs: 3.110.54.55   PrivateIPs: 172.31.7.35

7) Verify: check the Ohio (us-east-2) and N. Virginia (us-east-1) regions.