Terraform Assignment - 2

You have been asked to:

- Destroy the previous deployment
- Create a new EC2 instance with an Elastic IP

Solution:

1) Run Terraform Destroy to destroy the previous deployment : terraform destroy

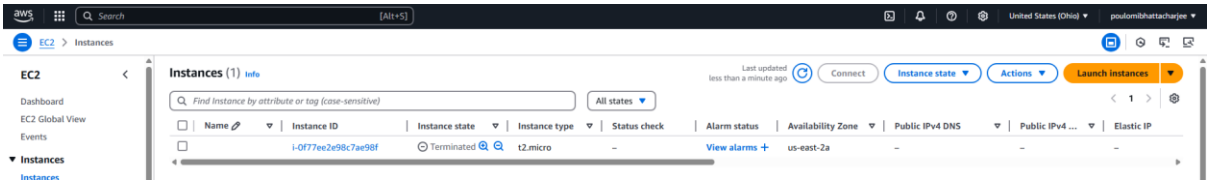
```
aws_instance.example: Destroying... [id=i-0f77ee2e98c7ae98f]
aws_instance.example: Still destroying... [id=i-0f77ee2e98c7ae98f, 10s elapsed]
aws_instance.example: Still destroying... [id=i-0f77ee2e98c7ae98f, 20s elapsed]
aws_instance.example: Still destroying... [id=i-0f77ee2e98c7ae98f, 30s elapsed]
aws_instance.example: Destruction complete after 32s

Destroy complete! Resources: 1 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 : we can see that the EC2 I ohio region is terminated



3) Modify main.tf file to launch an EC2 instance and assign an Elastic IP:

```
  GNU nano 7.2
provider "aws" {
  region = "us-east-2"
}

# Create EC2 instance
resource "aws_instance" "example" {
  ami           = "ami-0cb91c7de36eed2cb"
  instance_type = "t2.micro"

  tags = {
    Name = "Terraform-EC2-With-EIP"
  }
}

# Create Elastic IP
resource "aws_eip" "example" {
  instance = aws_instance.example.id
```

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 and apply :

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


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

i-0c2ef8ee473810c4e (my-terraform)

PublicIPs: 3.110.54.55 PrivateIPs: 172.31.7.35

```
aws_instance.example: Creating...
aws_instance.example: Still creating... [10s elapsed]
aws_instance.example: Creation complete after 15s [id=i-09adb83b66136cf85]
aws_eip.example: Creating...
aws_eip.example: Creation complete after 3s [id=eipalloc-05ca384efa1179919]

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

6) Verify : go to ohio region and we can see EC2 with elastic IP