

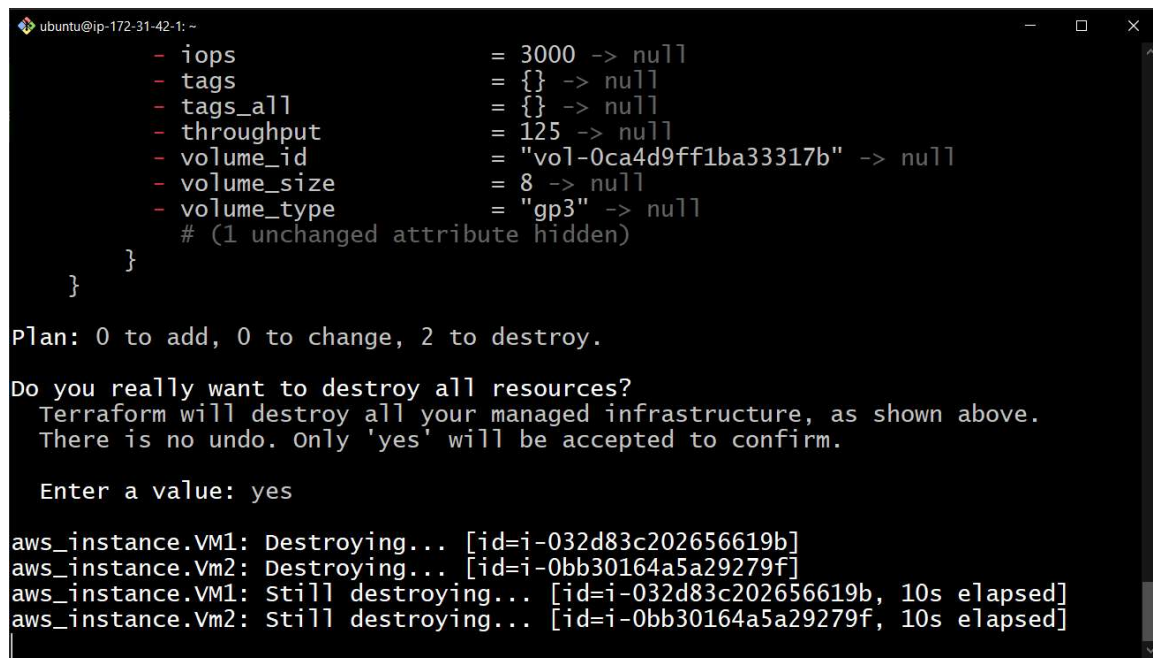
Assignment3

Terraform3

Tasks To Be Performed:

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'

Step1: Destruction of previous deployments

A terminal window with a black background and white text. The title bar shows 'ubuntu@ip-172-31-42-1: ~'. The output of a Terraform destroy command is shown. It lists attributes for two resources: iops, tags, tags_all, throughput, volume_id, volume_size, and volume_type, all of which are being destroyed (indicated by '-> null'). A comment indicates one unchanged attribute is hidden. The plan shows 0 to add, 0 to change, and 2 to destroy. A confirmation prompt asks 'Do you really want to destroy all resources?' and the user enters 'yes'. The final output shows two AWS instances being destroyed, with their IDs and the time elapsed (10s) for each.

```
ubuntu@ip-172-31-42-1: ~  
- iops                = 3000 -> null  
- tags                = {} -> null  
- tags_all            = {} -> null  
- throughput          = 125 -> null  
- volume_id           = "vol-0ca4d9ff1ba33317b" -> null  
- volume_size         = 8 -> null  
- volume_type         = "gp3" -> null  
  # (1 unchanged attribute hidden)  
}  
}  
  
Plan: 0 to add, 0 to change, 2 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_instance.VM1: Destroying... [id=i-032d83c202656619b]  
aws_instance.VM2: Destroying... [id=i-0bb30164a5a29279f]  
aws_instance.VM1: Still destroying... [id=i-032d83c202656619b, 10s elapsed]  
aws_instance.VM2: Still destroying... [id=i-0bb30164a5a29279f, 10s elapsed]
```

Step2: Creation of 2ec2 instance logic in terraform configuration file.

```

    access_key = var.aws_access_key
    secret_key = var.aws_secret_key
    region = var.region2
  )

resource "aws_instance" "VM1" {
  provider = aws.ohio
  ami      = "ami-003932de22c285676"
  instance_type = "t2.micro"
  key_name  = "keyohio.pem"
  tags = {
    Name = "hello-ohio"
  }
}

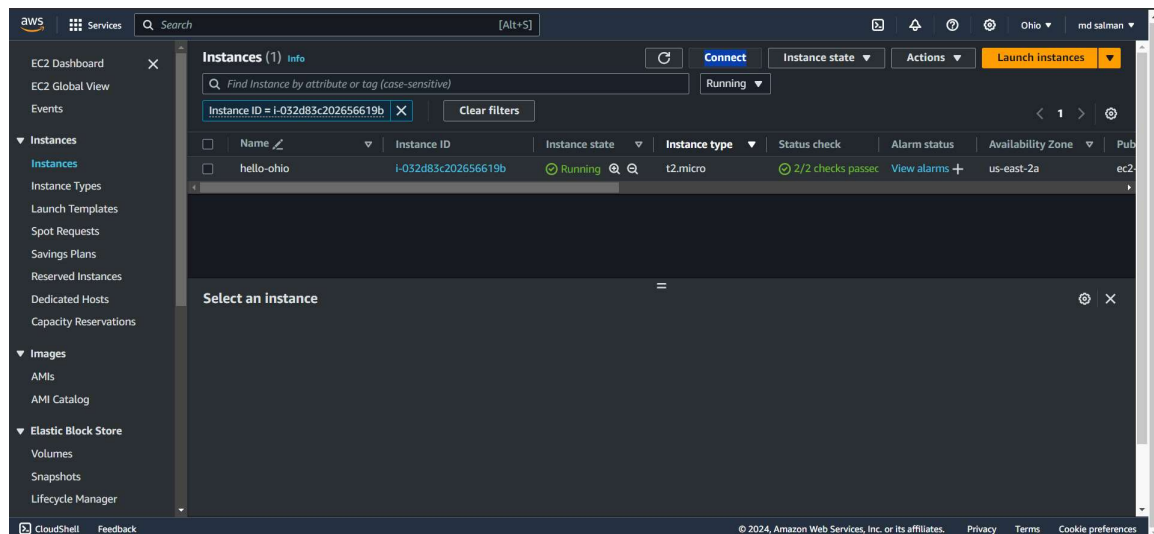
resource "aws_instance" "Vm2" {
  provider = aws.NV
  ami      = "ami-04a81a99f5ec58529"
  instance_type = "t2.micro"
  key_name  = "New-key"
  tags = {
    Name = "hello-virginia"
  }
}

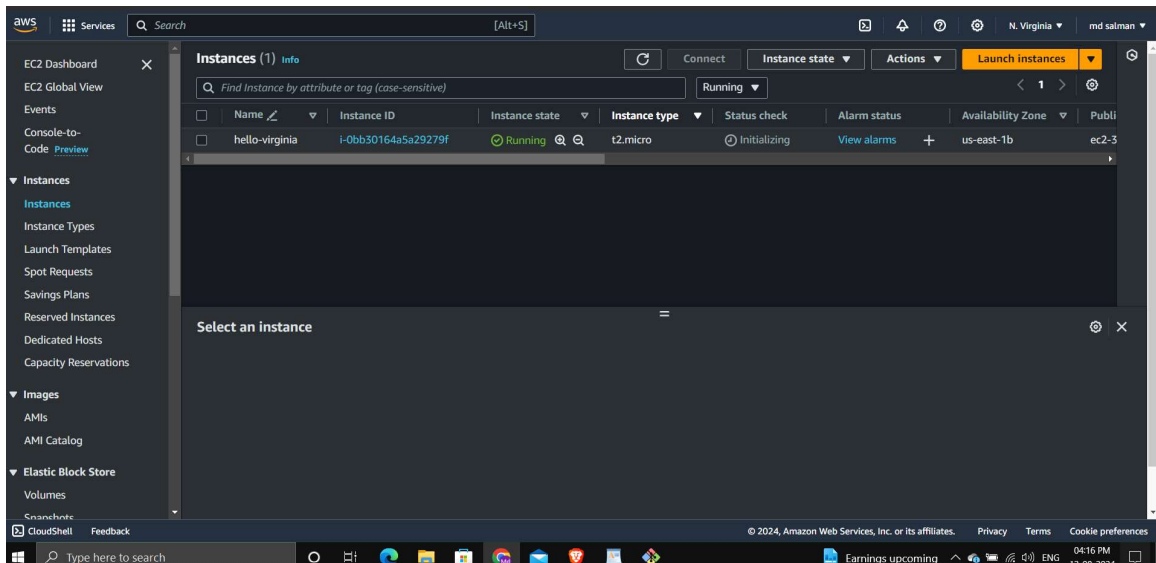
```

i-06b72e810e4ee9a1a (Terraform)

PublicIPs: 18.221.26.80 PrivateIPs: 172.31.42.1

Step:3: As you can see our hello-ohio instance which is in ohio region is running fine.





As you can see the 2nd instance is also running file which is hello virginia.

