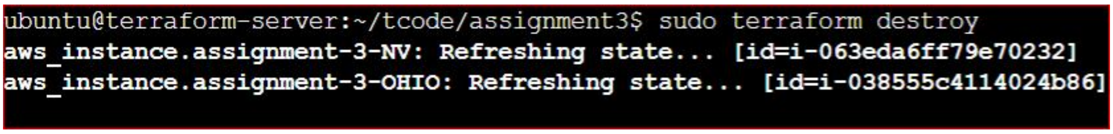
Module 8: Terraform Assignment - 4

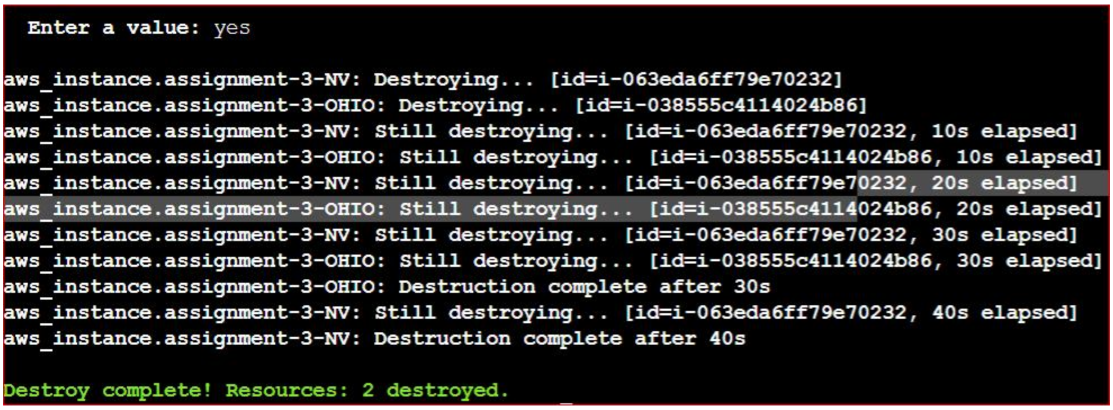
1. Destroy the previous deployments
2. Create a VPC with the required components using Terraform
3. Deploy an EC2 instance inside the VPC

Solution:-

$ sudo terraform destroy

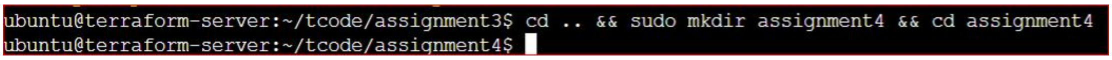


```
ubuntu@terraform-server:~/tcode/assignment3$ sudo terraform destroy
aws_instance.assignment-3-NV: Refreshing state... [id=i-063eda6ff79e70232]
aws_instance.assignment-3-OHIO: Refreshing state... [id=i-038555c4114024b86]
```



```
  Enter a value: yes

aws_instance.assignment-3-NV: Destroying... [id=i-063eda6ff79e70232]
aws_instance.assignment-3-OHIO: Destroying... [id=i-038555c4114024b86]
aws_instance.assignment-3-NV: Still destroying... [id=i-063eda6ff79e70232, 10s elapsed]
aws_instance.assignment-3-OHIO: Still destroying... [id=i-038555c4114024b86, 10s elapsed]
aws_instance.assignment-3-NV: Still destroying... [id=i-063eda6ff79e70232, 20s elapsed]
aws_instance.assignment-3-OHIO: Still destroying... [id=i-038555c4114024b86, 20s elapsed]
aws_instance.assignment-3-NV: Still destroying... [id=i-063eda6ff79e70232, 30s elapsed]
aws_instance.assignment-3-OHIO: Still destroying... [id=i-038555c4114024b86, 30s elapsed]
aws_instance.assignment-3-OHIO: Destruction complete after 30s
aws_instance.assignment-3-NV: Still destroying... [id=i-063eda6ff79e70232, 40s elapsed]
aws_instance.assignment-3-NV: Destruction complete after 40s

Destroy complete! Resources: 2 destroyed.
```

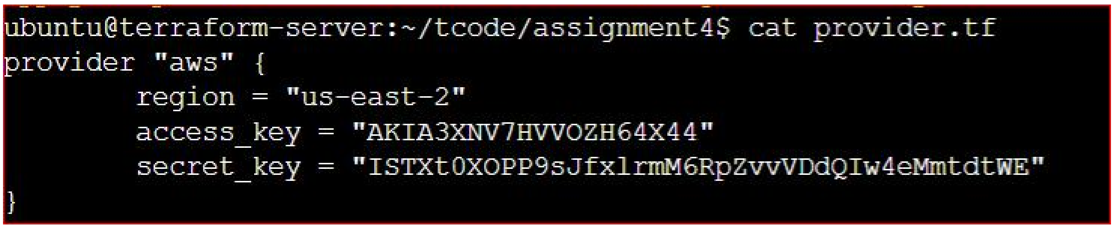
$ cd .. && sudo mkdir assignment4 && cd assignment4



```
ubuntu@terraform-server:~/tcode/assignment3$ cd .. && sudo mkdir assignment4 && cd assignment4
ubuntu@terraform-server:~/tcode/assignment4$
```

```
$ sudo vi provider.tf
```

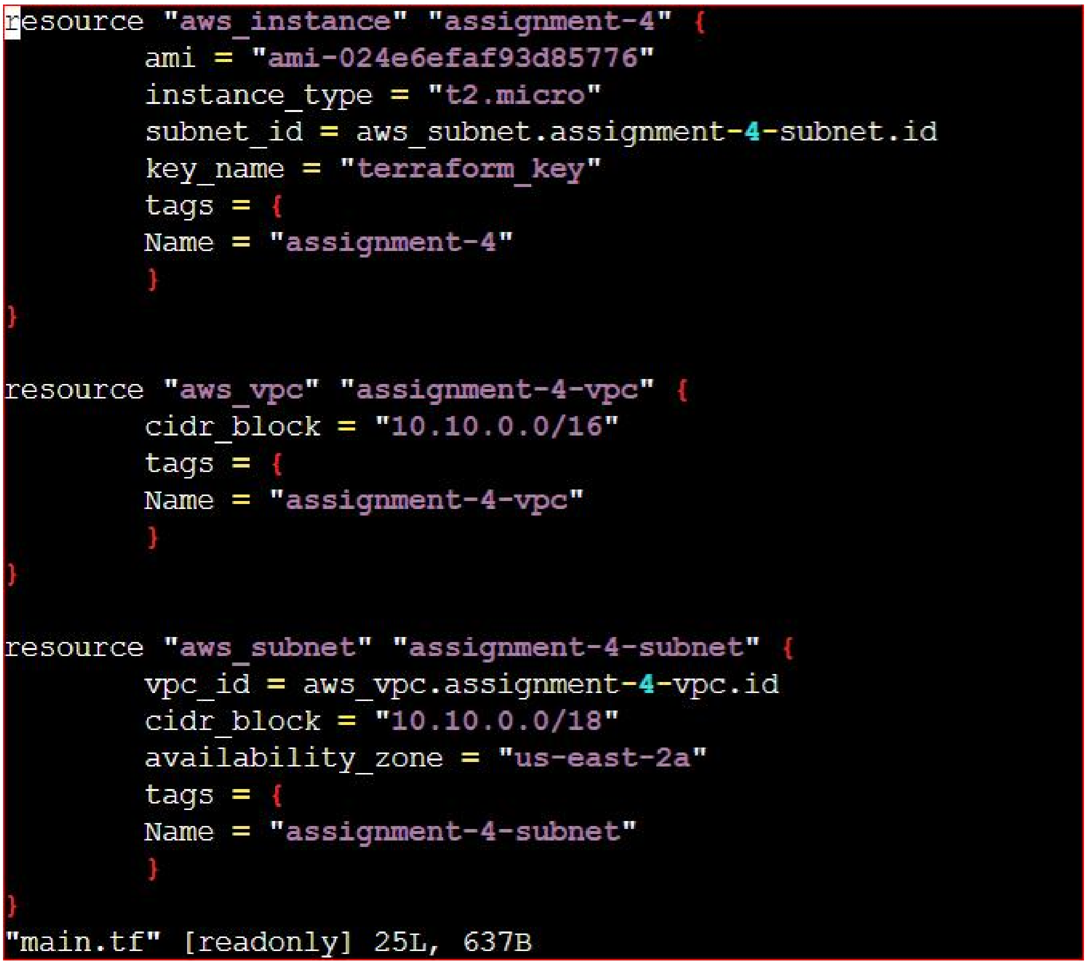
```
provider "aws" {
        region = "us-east-2"
        access_key = "AKIA3XNV7HVVOZH64X44"
        secret_key = "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"
}
```



```
ubuntu@terraform-server:~/tcode/assignment4$ cat provider.tf
provider "aws" {
        region = "us-east-2"
        access_key = "AKIA3XNV7HVVOZH64X44"
        secret_key = "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"
}
```

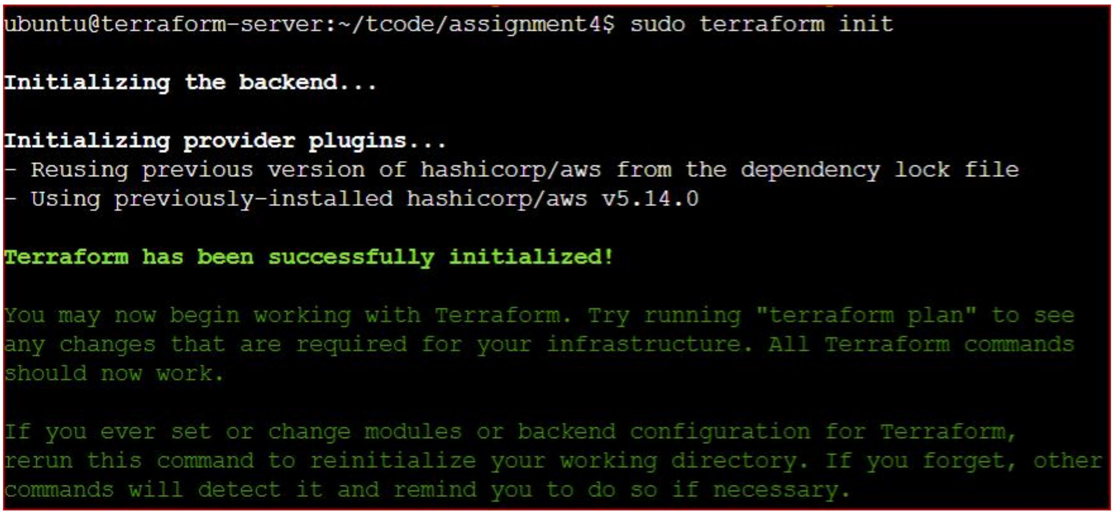
```
$ sudo vi main.tf
```

```
resource "aws_instance" "assignment-4" {
        ami = "ami-024e6efaf93d85776"
        instance_type = "t2.micro"
        subnet_id = aws_subnet.assignment-4-subnet.id
        key_name = "terraform_key"
        tags = {
        Name = "assignment-4"
        }
}
resource "aws_vpc" "assignment-4-vpc" {
        cidr_block = "10.10.0.0/16"
        tags = {
        Name = "assignment-4-vpc"
        }
}
resource "aws_subnet" "assignment-4-subnet"
{
        vpc_id = aws_vpc.assignment-4-vpc.id
        cidr_block = "10.10.0.0/18"
        availability_zone = "us-east-2a"
        tags = {
        Name = "assignment-4-subnet"
        }
}
```



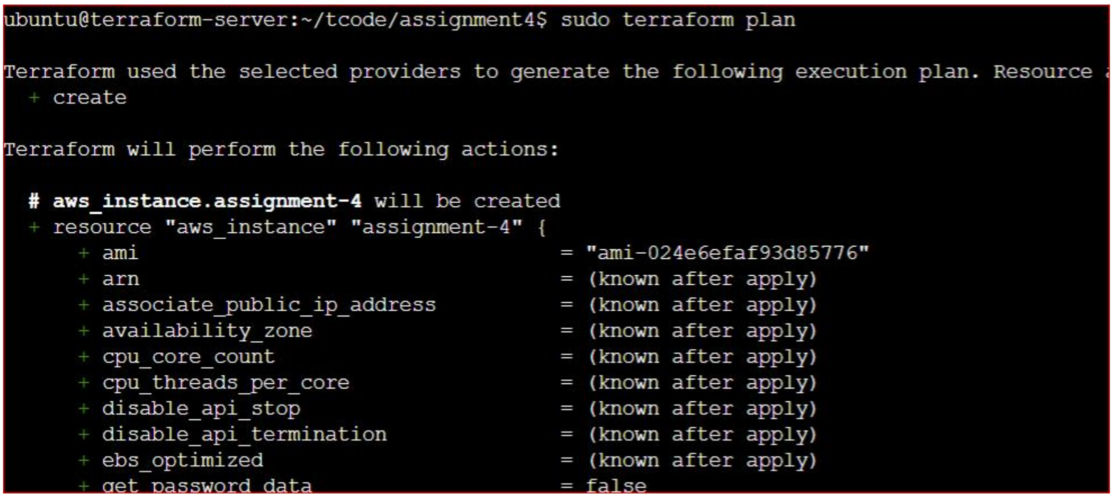
```
resource "aws_instance" "assignment-4" {
        ami = "ami-024e6efaf93d85776"
        instance_type = "t2.micro"
        subnet_id = aws_subnet.assignment-4-subnet.id
        key_name = "terraform_key"
        tags = {
        Name = "assignment-4"
        }
}

resource "aws_vpc" "assignment-4-vpc" {
        cidr_block = "10.10.0.0/16"
        tags = {
        Name = "assignment-4-vpc"
        }
}

resource "aws_subnet" "assignment-4-subnet" {
        vpc_id = aws_vpc.assignment-4-vpc.id
        cidr_block = "10.10.0.0/18"
        availability_zone = "us-east-2a"
        tags = {
        Name = "assignment-4-subnet"
        }
}
"main.tf" [readonly] 25L, 637B
```

$ sudo terraform init

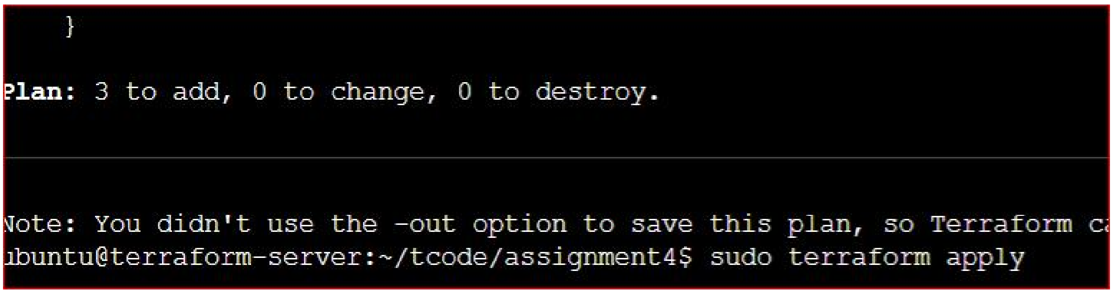


```
ubuntu@terraform-server:~/tcode/assignment4$ sudo terraform init

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

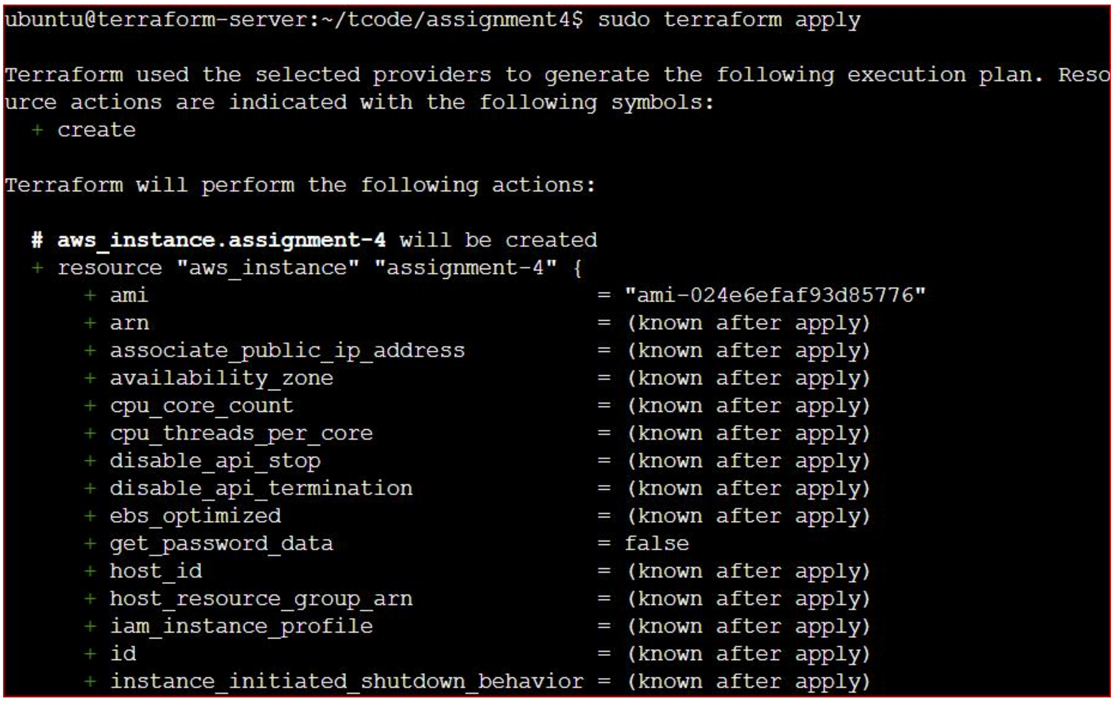


```
ubuntu@terraform-server:~/tcode/assignment4$ sudo terraform plan

Terraform used the selected providers to generate the following execution plan. Resource
  + create

Terraform will perform the following actions:

  # aws_instance.assignment-4 will be created
  + resource "aws_instance" "assignment-4" {
      + 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
```

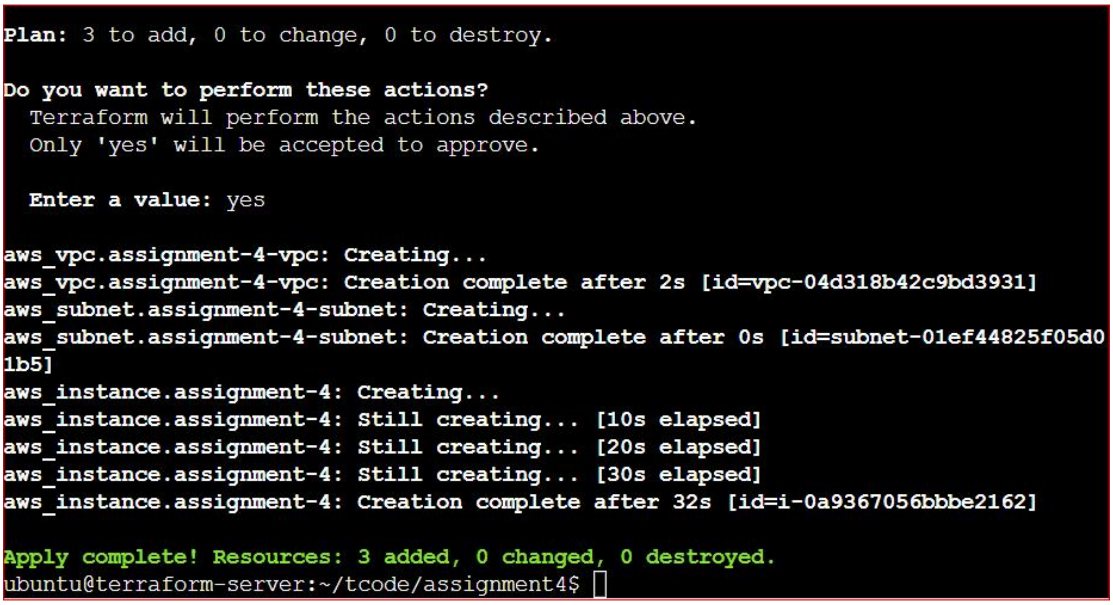


```
    }

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


Note: You didn't use the -out option to save this plan, so Terraform c
ubuntu@terraform-server:~/tcode/assignment4$ sudo terraform apply
```

$ sudo terraform apply

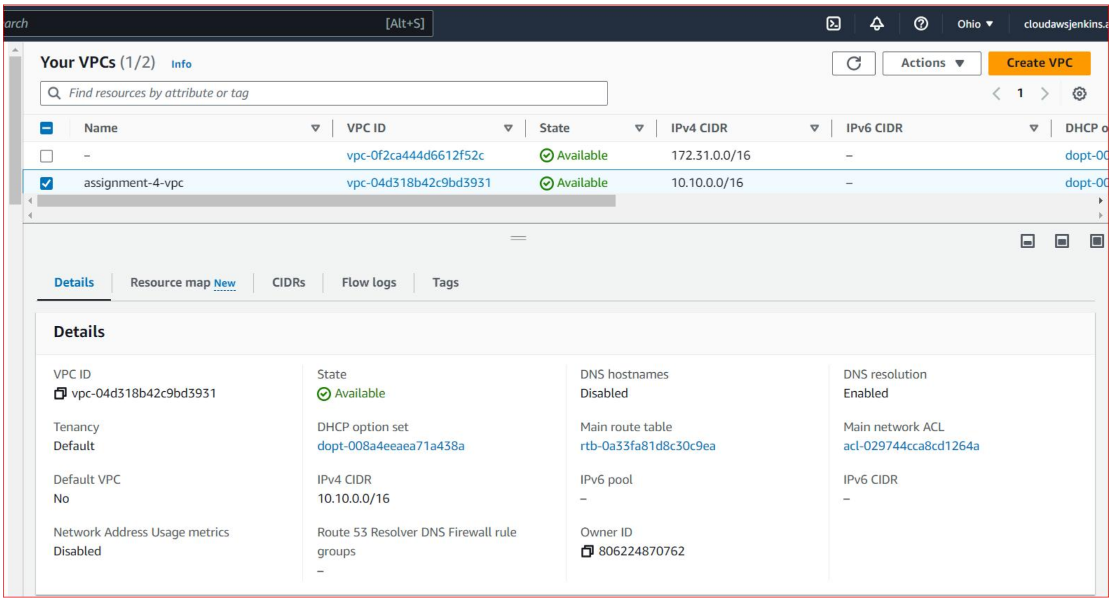


```
ubuntu@terraform-server:~/tcode/assignment4$ sudo terraform apply

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-4 will be created
  + resource "aws_instance" "assignment-4" {
      + 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)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
```



```
Plan: 3 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_vpc.assignment-4-vpc: Creating...
aws_vpc.assignment-4-vpc: Creation complete after 2s [id=vpc-04d318b42c9bd3931]
aws_subnet.assignment-4-subnet: Creating...
aws_subnet.assignment-4-subnet: Creation complete after 0s [id=subnet-01ef44825f05d0
1b5]
aws_instance.assignment-4: Creating...
aws_instance.assignment-4: Still creating... [10s elapsed]
aws_instance.assignment-4: Still creating... [20s elapsed]
aws_instance.assignment-4: Still creating... [30s elapsed]
aws_instance.assignment-4: Creation complete after 32s [id=i-0a9367056bbbe2162]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
ubuntu@terraform-server:~/tcode/assignment4$ 
```

* VPC created present in AWS console:-



Your VPCs (1/2)

	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP o
□	–	vpc-0f2ca444d6612f52c	Available	172.31.0.0/16	–	dopt-0(
☑	assignment-4-vpc	vpc-04d318b42c9bd3931	Available	10.10.0.0/16	–	dopt-0(

Details

VPC ID
vpc-04d318b42c9bd3931

State
Available

DNS hostnames
Disabled

DNS resolution
Enabled

Tenancy
Default

DHCP option set
dopt-008a4eeaea71a438a

Main route table
rtb-0a33fa81d8c30c9ea

Main network ACL
acl-029744cca8cd1264a

Default VPC
No

IPv4 CIDR
10.10.0.0/16

IPv6 pool
–

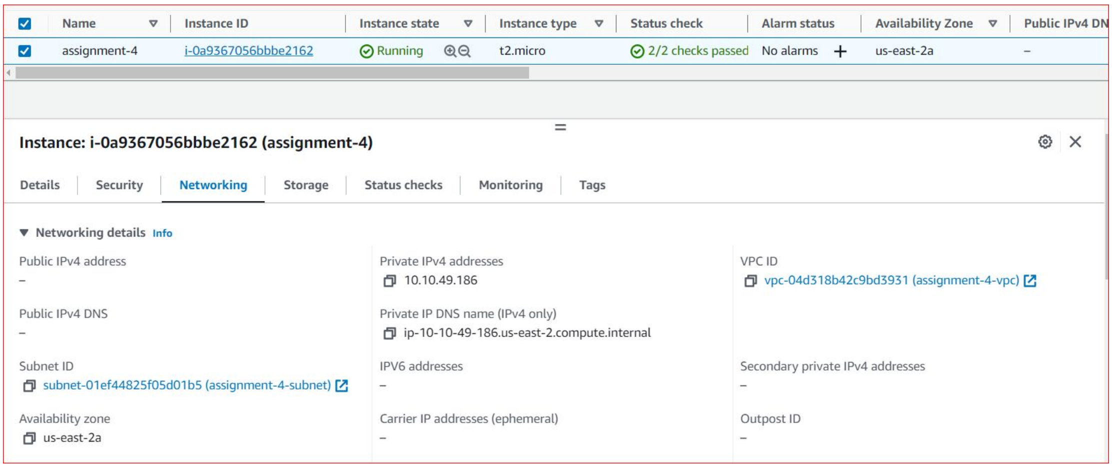
IPv6 CIDR
–

Network Address Usage metrics
Disabled

Route 53 Resolver DNS Firewall rule groups
–

Owner ID
806224870762

* Instance created present in AWS console contain above VPC:-



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
☑	assignment-4	i-0a9367056bbbe2162	Running	t2.micro	2/2 checks passed	No alarms +	us-east-2a	–

Instance: i-0a9367056bbbe2162 (assignment-4)

Networking details

Public IPv4 address
–

Private IPv4 addresses
10.10.49.186

VPC ID
vpc-04d318b42c9bd3931 (assignment-4-vpc)

Public IPv4 DNS
–

Private IP DNS name (IPv4 only)
ip-10-10-49-186.us-east-2.compute.internal

Subnet ID
subnet-01ef44825f05d01b5 (assignment-4-subnet)

IPV6 addresses
–

Secondary private IPv4 addresses
–

Availability zone
us-east-2a

Carrier IP addresses (ephemeral)
–

Outpost ID
–