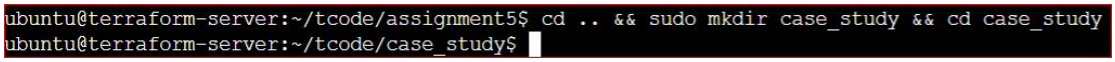
Module 8: Case Study - Creating an Architecture using Terraform on AWS

You work as a DevOps Engineer in a leading software company. You have been
asked to build an infrastructure safely and efficiently.
The company's requirements:
1. Use AWS Cloud Provider and the software to be installed is Apache2
2. Use Ubuntu AMI
The company wants the architecture to have the following services:
1. Create a template with a VPC, 2 subnets and 1 instance in each subnet
2. Attach security groups, internet gateway and network interface to the
instance

Solution:-

```
$ cd .. && mkdir case_study && cd case_study
```




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

```
provider "aws" {
      region = "us-east-2"
      access_key = "AKIA3XNV7HVVOZH64X44"
      secret_key =
"ISTXt0XOPP9sJfxlrmM6RpZvvVDdQIw4eMmtd
tWE"
```

}

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

$ sudo vi main.tf

```
# Terraform-casestudy-instance1

resource "aws_instance" "case_study_server1"
{
 ami = "ami-024e6efaf93d85776"
 instance_type = "t2.micro"
 user_data = <<-EOL
 #!/bin/bash -xe
 sudo apt-get update
 sudo apt-get install apache2
 EOL
 subnet_id = "${aws_subnet.first.id}"
 key_name = "terraform_key"
 tags = {
    Name = "Terraform-casestudy-instance1"
       }
}
resource "aws_vpc" "main1" {
 cidr_block = "172.31.0.0/16"
 enable_dns_support = "1"
 enable_dns_hostnames = "1"
 tags = {
    Name = "myfirstvpc"
```

```
    }
}
```

```
# Terraform-casestudy-instance1

resource "aws_instance" "case_study_server1"
  ami = "ami-024e6efaf93d85776"
  instance_type = "t2.micro"
  user_data = <<-EOL
  #!/bin/bash -xe
  sudo apt-get update
  sudo apt-get install apache2
  EOL
  subnet_id = "${aws_subnet.first.id}"
  key_name = "terraform_key"
  tags = {
      Name = "Terraform-casestudy-instance1"
          }
}
resource "aws_vpc" "main1" {
  cidr_block = "172.31.0.0/16"
  enable_dns_support = "1"
  enable_dns_hostnames = "1"
 tags = {
      Name = "myfirstvpc"
          }

}
```

```
resource "aws_subnet" "first" {
 availability_zone = "us-east-2a"
 cidr_block = "172.31.0.0/16"
 map_public_ip_on_launch = "1"
 vpc_id = "${aws_vpc.main1.id}"
 tags = {
    Name = "myfirstsubnet"
       }
}
resource "aws_default_security_group"
"default_myfirst" {
```

```
ingress {
 from_port   = 0
 to_port     = 0
 protocol    = "-1"
 cidr_blocks = ["0.0.0.0/0"]
}
egress {
 from_port   = 0
 to_port     = 0
 protocol    = "-1"
 cidr_blocks = ["0.0.0.0/0"]
}
vpc_id = "${aws_vpc.main1.id}"
tags = {
   Name = "myfirstsecuritygroup"
```

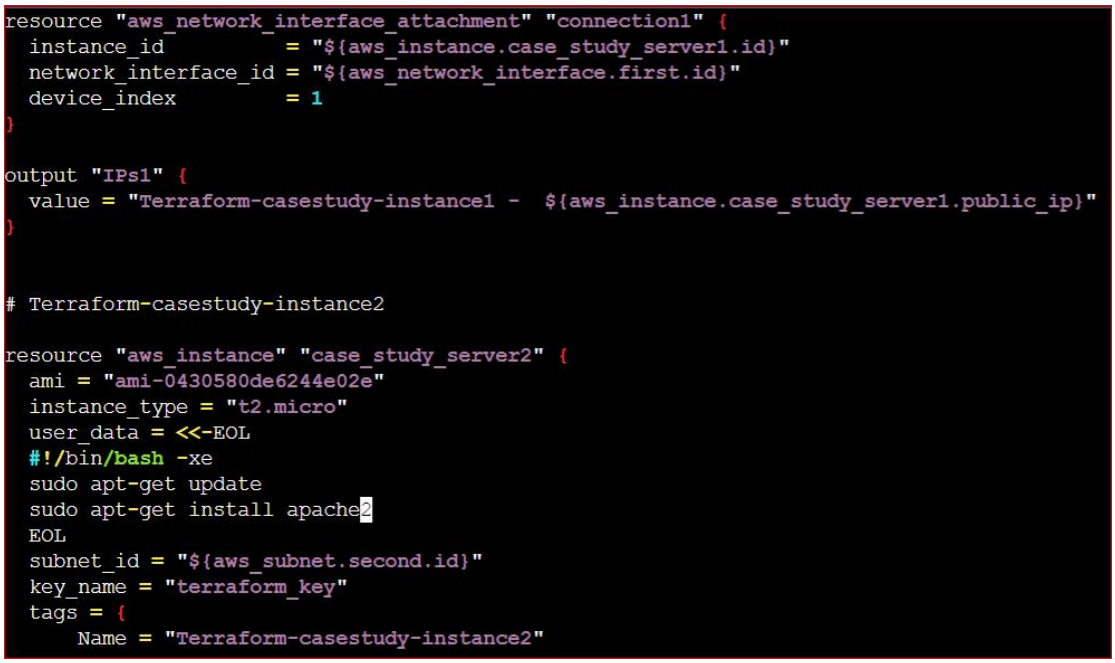
```
resource "aws_subnet" "first" {
  availability_zone = "us-east-2a"
  cidr_block = "172.31.0.0/16"
  map_public_ip_on_launch = "1"
  vpc_id = "${aws_vpc.main1.id}"
  tags = {
      Name = "myfirstsubnet"
          }
}
resource "aws_default_security_group" "default_myfirst" {
  ingress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
  vpc_id = "${aws_vpc.main1.id}"
  tags = {
      Name = "myfirstsecuritygroup"
```

```
resource "aws_internet_gateway" "internet1" {
 vpc_id = "${aws_vpc.main1.id}"
 tags = {
    Name = "myinternetgateway1"
       }
}
resource "aws_route" "internet1" {
 route_table_id            =
"${aws_vpc.main1.default_route_table_id}"
 destination_cidr_block    = "0.0.0.0/0"
 gateway_id =
"${aws_internet_gateway.internet1.id}"
}
resource "aws_route_table_association" "a1" {
 subnet_id      = "${aws_subnet.first.id}"
 route_table_id =
"${aws_vpc.main1.default_route_table_id}"
}
resource "aws_network_interface" "first" {
 subnet_id = "${aws_subnet.first.id}"
 tags = {
    Name = "mynetworkinterface1"
       }
```

```
resource "aws_internet_gateway" "internet1" {
  vpc_id = "${aws_vpc.main1.id}"
  tags = {
      Name = "myinternetgateway1"
          }

}


resource "aws_route" "internet1" {
  route_table_id            = "${aws_vpc.main1.default_route_table_id}"
  destination_cidr_block    = "0.0.0.0/0"
  gateway_id = "${aws_internet_gateway.internet1.id}"

}

resource "aws_route_table_association" "a1" {
  subnet_id      = "${aws_subnet.first.id}"
  route_table_id = "${aws_vpc.main1.default_route_table_id}"
}
resource "aws_network_interface" "first" {
  subnet_id = "${aws_subnet.first.id}"
  tags = {
      Name = "mynetworkinterface1"
          }
```

```
resource "aws_network_interface_attachment"
"connection1" {
 instance_id        =
"${aws_instance.case_study_server1.id}"
 network_interface_id =
"${aws_network_interface.first.id}"
 device_index       = 1
}
output "IPs1" {
 value = "Terraform-casestudy-instance1 -
${aws_instance.case_study_server1.public_ip}"
}

# Terraform-casestudy-instance2
resource "aws_instance" "case_study_server2" {
 ami = "ami-0430580de6244e02e"
```

```
 instance_type = "t2.micro"
 user_data = <<-EOL
 #!/bin/bash -xe
 sudo apt-get update
 sudo apt-get install apache2
 EOL
 subnet_id = "${aws_subnet.second.id}"
 key_name = "terraform_key"
 tags = {
    Name = "Terraform-casestudy-instance2"
```



```
resource "aws_network_interface_attachment" "connection1" {
  instance_id          = "${aws_instance.case_study_server1.id}"
  network_interface_id = "${aws_network_interface.first.id}"
  device_index         = 1
}

output "IPs1" {
  value = "Terraform-casestudy-instance1 -  ${aws_instance.case_study_server1.public_ip}"
}


# Terraform-casestudy-instance2

resource "aws_instance" "case_study_server2" {
  ami = "ami-0430580de6244e02e"
  instance_type = "t2.micro"
  user_data = <<-EOL
  #!/bin/bash -xe
  sudo apt-get update
  sudo apt-get install apache2
  EOL
  subnet_id = "${aws_subnet.second.id}"
  key_name = "terraform_key"
  tags = {
      Name = "Terraform-casestudy-instance2"
```

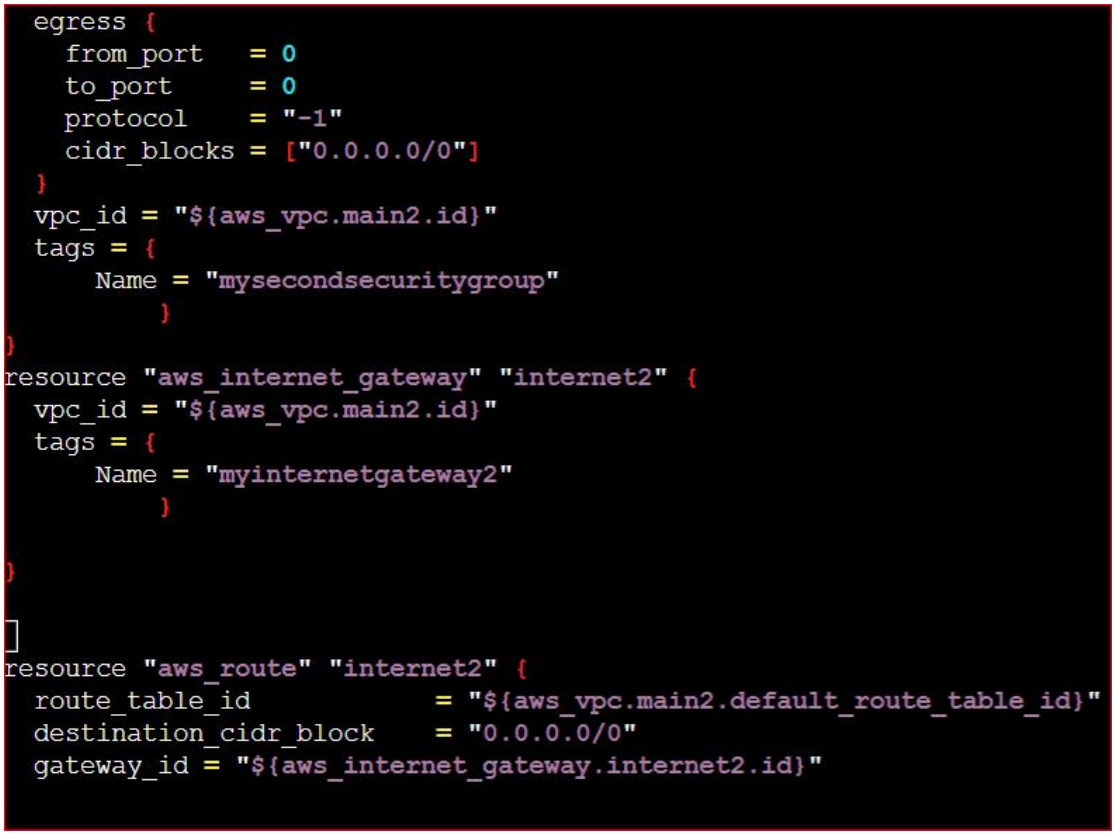
```
 }
resource "aws_vpc" "main2" {
 cidr_block = "172.31.0.0/16"
 enable_dns_support = "1"
 enable_dns_hostnames = "1"
tags = {
    Name = "mysecondvpc"
        }
```

```
}
resource "aws_subnet" "second" {
  availability_zone = "us-east-2a"
  cidr_block = "172.31.0.0/16"
  map_public_ip_on_launch = "1"
  vpc_id = "${aws_vpc.main2.id}"
  tags = {
     Name = "mysecondsubnet"
        }
}
resource "aws_default_security_group"
"default_mysecond" {
  ingress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
```

```
}
resource "aws_vpc" "main2" {
  cidr_block = "172.31.0.0/16"
  enable_dns_support = "1"
  enable_dns_hostnames = "1"
 tags = {
      Name = "mysecondvpc"
          }

}

resource "aws_subnet" "second" {
  availability_zone = "us-east-2a"
  cidr_block = "172.31.0.0/16"
  map_public_ip_on_launch = "1"
  vpc_id = "${aws_vpc.main2.id}"
  tags = {
      Name = "mysecondsubnet"
          }
}
resource "aws_default_security_group" "default_mysecond" {
  ingress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
```

```
egress {
  from_port   = 0
  to_port     = 0
  protocol    = "-1"
  cidr_blocks = ["0.0.0.0/0"]
 }
 vpc_id = "${aws_vpc.main2.id}"
 tags = {
    Name = "mysecondsecuritygroup"
       }
}
resource "aws_internet_gateway" "internet2" {
 vpc_id = "${aws_vpc.main2.id}"
 tags = {
```

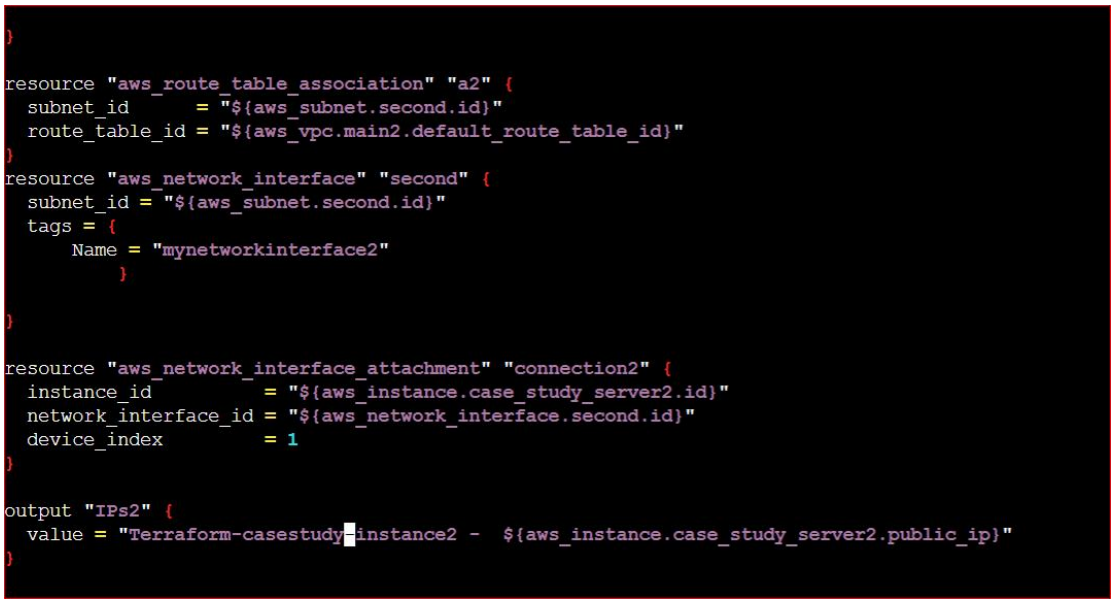
```
    Name = "myinternetgateway2"
        }
}
resource "aws_route" "internet2" {
 route_table_id          =
"${aws_vpc.main2.default_route_table_id}"
 destination_cidr_block    = "0.0.0.0/0"
 gateway_id =
"${aws_internet_gateway.internet2.id}"
```



```
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
  vpc_id = "${aws_vpc.main2.id}"
  tags = {
      Name = "mysecondsecuritygroup"
          }
}
resource "aws_internet_gateway" "internet2" {
  vpc_id = "${aws_vpc.main2.id}"
  tags = {
      Name = "myinternetgateway2"
          }

}

]
resource "aws_route" "internet2" {
  route_table_id            = "${aws_vpc.main2.default_route_table_id}"
  destination_cidr_block    = "0.0.0.0/0"
  gateway_id = "${aws_internet_gateway.internet2.id}"
```

```
}
resource "aws_route_table_association" "a2" {
 subnet_id      = "${aws_subnet.second.id}"
 route_table_id =
"${aws_vpc.main2.default_route_table_id}"
}
```

```
resource "aws_network_interface" "second" {
 subnet_id = "${aws_subnet.second.id}"
 tags = {
    Name = "mynetworkinterface2"
       }
}
resource "aws_network_interface_attachment"
"connection2" {
 instance_id         =
"${aws_instance.case_study_server2.id}"
 network_interface_id =
"${aws_network_interface.second.id}"
 device_index        = 1
}
output "IPs2" {
 value = "Terraform-casestudy-instance2 -
${aws_instance.case_study_server2.public_ip}"
}
```



$ sudo terraform init

```
ubuntu@terraform-server:~/tcode/case_study$ 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/case_study$ 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_default_security_group.default_myfirst will be created
  + resource "aws_default_security_group" "default_myfirst" {
      + arn                    = (known after apply)
      + description            = (known after apply)
      + egress                 = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + description      = ""
              + from_port        = 0
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "-1"
```

```
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags                                 = {
          + "Name" = "mysecondvpc"
        }
      + tags_all                             = {
          + "Name" = "mysecondvpc"
        }
    }

Plan: 18 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + IPs1 = (known after apply)
  + IPs2 = (known after apply)

─────────────────────────────────────────────────────────────────────────────
─────────────────────────────────────────────────────────────────────────────
──

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee
 to take exactly these actions if you run "terraform apply" now.
```

$ sudo terraform apply

```
ubuntu@terraform-server:~/tcode/case_study$ 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_default_security_group.default_myfirst will be created
  + resource "aws_default_security_group" "default_myfirst" {
      + arn                    = (known after apply)
      + description            = (known after apply)
      + egress                 = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + description      = ""
              + from_port        = 0
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "-1"
              + security_groups  = []
              + self             = false
              + to_port          = 0
```

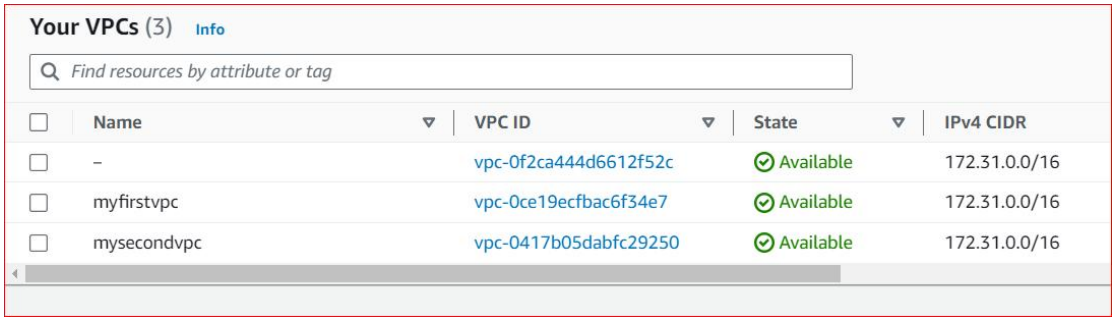
```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_vpc.main2: Creating...
aws_vpc.main1: Creating...
aws_vpc.main1: Still creating... [10s elapsed]
aws_vpc.main2: Still creating... [10s elapsed]
aws_vpc.main2: Creation complete after 11s [id=vpc-0417b05dabfc29250]
aws_vpc.main1: Creation complete after 11s [id=vpc-0ce19ecfbac6f34e7]
aws_subnet.second: Creating...
aws_internet_gateway.internet2: Creating...
aws_default_security_group.default_mysecond: Creating...
aws_default_security_group.default_myfirst: Creating...
aws_internet_gateway.internet1: Creating...
aws_subnet.first: Creating...
aws_internet_gateway.internet1: Creation complete after 1s [id=igw-0a667288974e80e4c
]
aws_internet_gateway.internet2: Creation complete after 1s [id=igw-07b0711367b1f6a1c
]
aws_route.internet1: Creating...
aws_route.internet2: Creating...
```

```
aws_internet_gateway.internet1: Creation complete after 1s [id=igw-0a667288974e80e4c
]
aws_internet_gateway.internet2: Creation complete after 1s [id=igw-07b0711367b1f6a1c
]
aws_route.internet1: Creating...
aws_route.internet2: Creating...
aws_route.internet2: Creation complete after 1s [id=r-rtb-09e6b231f310c19a6108028949
4]
aws_route.internet1: Creation complete after 1s [id=r-rtb-0c3310d13a22be36f108028949
4]
aws_default_security_group.default_myfirst: Creation complete after 3s [id=sg-000074
2fad550a9f9]
aws_default_security_group.default_mysecond: Creation complete after 3s [id=sg-00702
dba5b67c3f7f]
aws_subnet.second: Still creating... [10s elapsed]
aws_subnet.first: Still creating... [10s elapsed]
aws_subnet.second: Creation complete after 11s [id=subnet-0a1a7fed0cf01b624]
aws_subnet.first: Creation complete after 11s [id=subnet-08a6baae43a3b8ec1]
aws_instance.case_study_server2: Creating...
aws_route_table_association.a2: Creating...
aws_route_table_association.a1: Creating...
aws_network_interface.first: Creating...
aws_network_interface.second: Creating...
aws_instance.case_study_server1: Creating...
aws_route_table_association.a1: Creation complete after 0s [id=rtbassoc-0156a229b319
3ef3f]
```

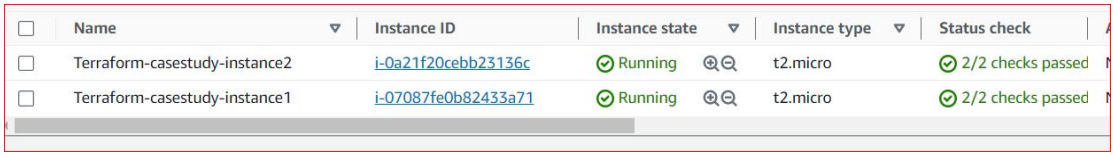
```
aws_network_interface.first: Creating...
aws_network_interface.second: Creating...
aws_instance.case_study_server1: Creating...
aws_route_table_association.a1: Creation complete after 0s [id=rtbassoc-0156a229b319
3ef3f]
aws_route_table_association.a2: Creation complete after 0s [id=rtbassoc-0f4a1e16ba25
e82b9]
aws_network_interface.second: Creation complete after 0s [id=eni-083caa5b2ae641c70]
aws_network_interface.first: Creation complete after 0s [id=eni-077b3cfcf056c854a]
aws_instance.case_study_server2: Still creating... [10s elapsed]
aws_instance.case_study_server1: Still creating... [10s elapsed]
aws_instance.case_study_server2: Still creating... [20s elapsed]
aws_instance.case_study_server1: Still creating... [20s elapsed]
aws_instance.case_study_server2: Still creating... [30s elapsed]
aws_instance.case_study_server1: Still creating... [30s elapsed]
aws_instance.case_study_server2: Creation complete after 31s [id=i-0a21f20cebb23136c
]
aws_instance.case_study_server1: Creation complete after 31s [id=i-07087fe0b82433a71
]
aws_network_interface_attachment.connection2: Creating...
aws_network_interface_attachment.connection1: Creating...
aws_network_interface_attachment.connection2: Still creating... [10s elapsed]
aws_network_interface_attachment.connection1: Still creating... [10s elapsed]
aws_network_interface_attachment.connection1: Creation complete after 15s [id=eni-at
tach-02281837cdcc67cfa]
aws_network_interface_attachment.connection2: Still creating... [20s elapsed]
```

```
aws_network_interface_attachment.connection1: Creation complete after 15s [id=eni-at
tach-02281837cdcc67cfa]
aws_network_interface_attachment.connection2: Still creating... [20s elapsed]
aws_network_interface_attachment.connection2: Creation complete after 25s [id=eni-at
tach-0ac3d9a4403eff7d8]

Apply complete! Resources: 18 added, 0 changed, 0 destroyed.

Outputs:

IPs1 = "Terraform-casestudy-instance1 -  3.15.219.193"
IPs2 = "Terraform-casestudy-instance2 -  3.145.173.202"
```

* VPCs created present in AWS console:-



* Instances created present in AWS console contain above VPCs:-



Name	Instance ID	Instance state	Instance type	Status check
Terraform-casestudy-instance2	i-0a21f20cebb23136c	Running	t2.micro	2/2 checks passed
Terraform-casestudy-instance1	i-07087fe0b82433a71	Running	t2.micro	2/2 checks passed