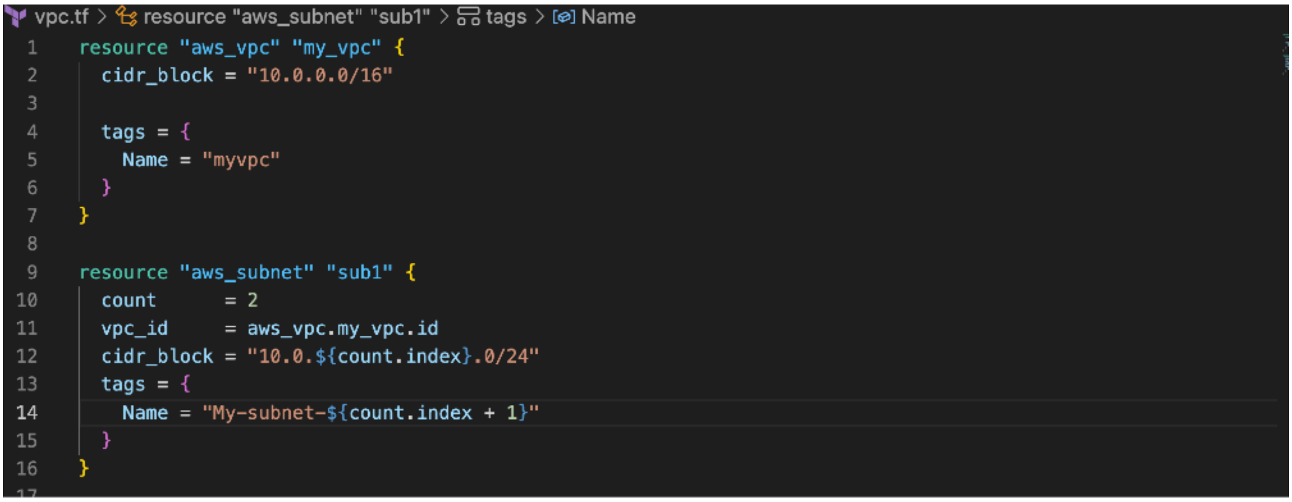
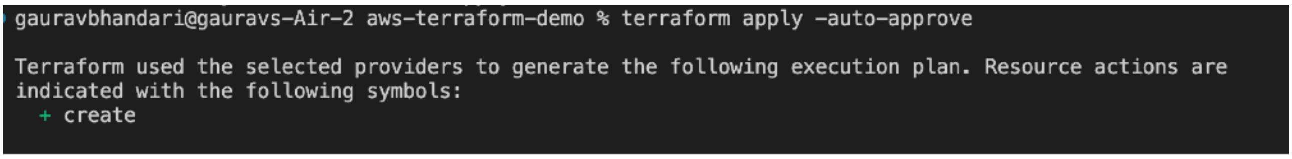
SPCM LAB-8

Objective:Learn how to use Terraform to create a basic Virtual Private Cloud (VPC) in AWS.

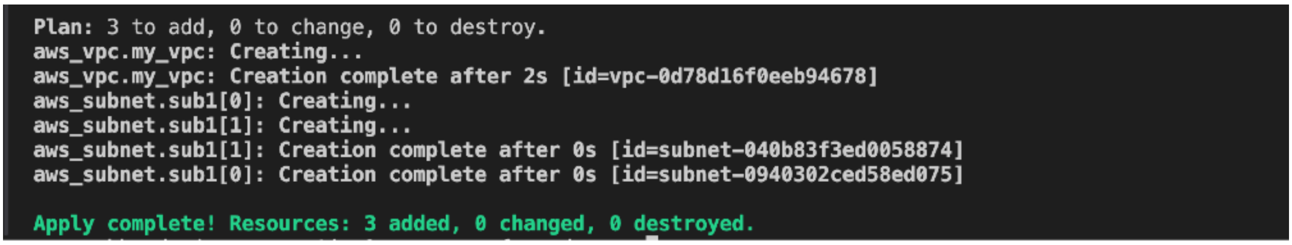
- Create a VPC tf file and run terraform apply -auto-approve



```
resource "aws_vpc" "my_vpc" {
  cidr_block = "10.0.0.0/16"

  tags = {
    Name = "myvpc"
  }
}

resource "aws_subnet" "sub1" {
  count      = 2
  vpc_id     = aws_vpc.my_vpc.id
  cidr_block = "10.0.${count.index}.0/24"
  tags = {
    Name = "My-subnet-${count.index + 1}"
  }
}
```

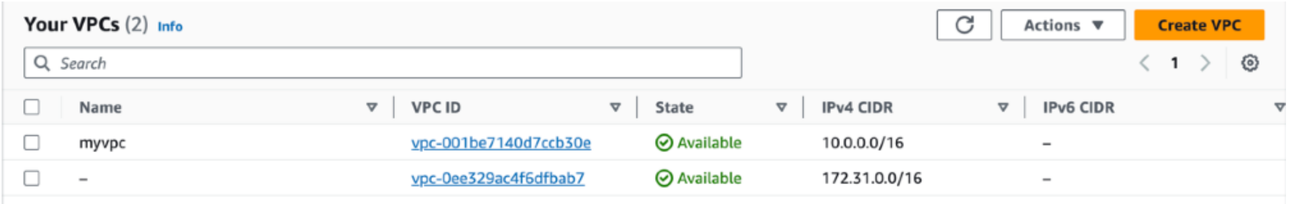


```
gauravbhandari@gauravs-Air-2 aws-terraform-demo % terraform apply -auto-approve

Terraform used the selected providers to generate the following execution plan. Resource actions are
indicated with the following symbols:
  + create
```

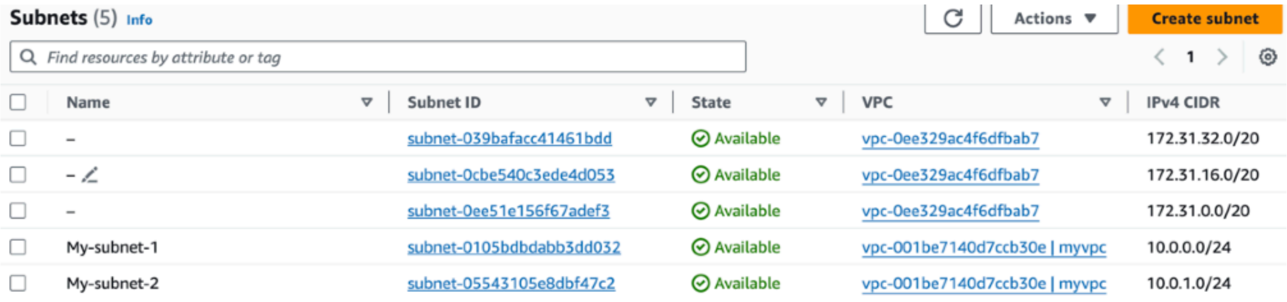


```
Plan: 3 to add, 0 to change, 0 to destroy.
aws_vpc.my_vpc: Creating...
aws_vpc.my_vpc: Creation complete after 2s [id=vpc-0d78d16f0eeb94678]
aws_subnet.sub1[0]: Creating...
aws_subnet.sub1[1]: Creating...
aws_subnet.sub1[1]: Creation complete after 0s [id=subnet-040b83f3ed0058874]
aws_subnet.sub1[0]: Creation complete after 0s [id=subnet-0940302ced58ed075]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
```

- Check the aws console and verify the creation of required resources.

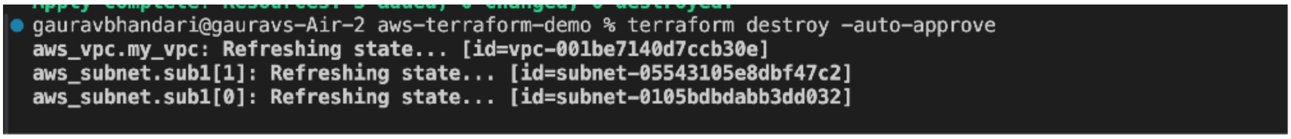


□	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
□	myvpc	vpc-001be7140d7ccb30e	✓ Available	10.0.0.0/16	–
□	–	vpc-0ee329ac4f6dfbab7	✓ Available	172.31.0.0/16	–

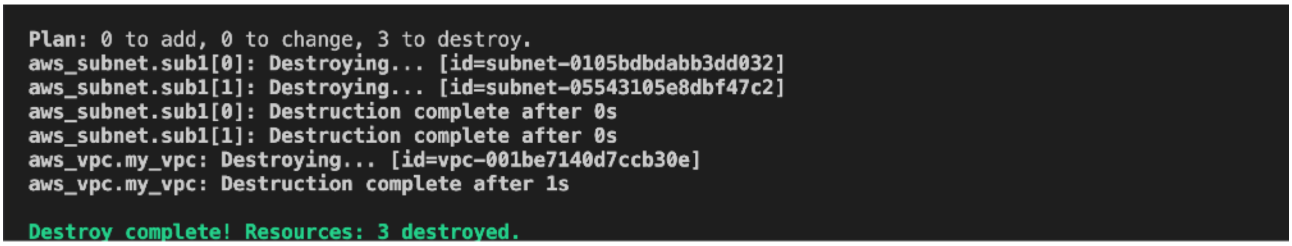


□	Name	Subnet ID	State	VPC	IPv4 CIDR
□	–	subnet-039bafacc41461bdd	✓ Available	vpc-0ee329ac4f6dfbab7	172.31.32.0/20
□	–	subnet-0cbe540c3ede4d053	✓ Available	vpc-0ee329ac4f6dfbab7	172.31.16.0/20
□	–	subnet-0ee51e156f67adef3	✓ Available	vpc-0ee329ac4f6dfbab7	172.31.0.0/20
□	My-subnet-1	subnet-0105bdbdabb3dd032	✓ Available	vpc-001be7140d7ccb30e | myvpc	10.0.0.0/24
□	My-subnet-2	subnet-05543105e8dbf47c2	✓ Available	vpc-001be7140d7ccb30e | myvpc	10.0.1.0/24

- After experimentation run terraform destroy -auto-approve



```
Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
● gauravbhandari@gauravs-Air-2 aws-terraform-demo % terraform destroy -auto-approve
  aws_vpc.my_vpc: Refreshing state... [id=vpc-001be7140d7ccb30e]
  aws_subnet.sub1[1]: Refreshing state... [id=subnet-05543105e8dbf47c2]
  aws_subnet.sub1[0]: Refreshing state... [id=subnet-0105bdbdabb3dd032]
```



```
Plan: 0 to add, 0 to change, 3 to destroy.
aws_subnet.sub1[0]: Destroying... [id=subnet-0105bdbdabb3dd032]
aws_subnet.sub1[1]: Destroying... [id=subnet-05543105e8dbf47c2]
aws_subnet.sub1[0]: Destruction complete after 0s
aws_subnet.sub1[1]: Destruction complete after 0s
aws_vpc.my_vpc: Destroying... [id=vpc-001be7140d7ccb30e]
aws_vpc.my_vpc: Destruction complete after 1s

Destroy complete! Resources: 3 destroyed.
```