Lab Exercise 8– Creating a VPC in Terraform Objective:

Objective:

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

Prerequisites:

• Terraform installed on your machine.
• AWS CLI configured with the necessary credentials.

Steps:

1. Create a Terraform Directory:

```
→ Terraform-SPCM-LAB  cd EXP-8
→ EXP-8
```

• Create Terraform Configuration Files:
• Create a file named main.tf:

main.tf

main.tf

EXP-8 > main.tf > resource "aws_vpc" "my_vpc"

```hcl
provider "aws" {
  region     = "ap-south-1"
  access_key = ""
  secret_key = ""
}
resource "aws_vpc" "my_vpc" {
  cidr_block           = "10.0.0.0/16"
  enable_dns_support   = true
  enable_dns_hostnames = true
  tags = {
    Name = "my_vpc"
  }
}
resource "aws_subnet" "my_subnet" {
  count                   = 2
  vpc_id                  = aws_vpc.my_vpc.id
  cidr_block              = "10.0.${count.index + 1}.0/24"
  availability_zone       = "ap-south-1"
  map_public_ip_on_launch = true
  tags = {
    Name = "MySubnet-${count.index + 1}"
  }
}

```

2. Initialize, Validate and Apply:

terraform init:

```
● → EXP-8  terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.36.0...
- Installed hashicorp/aws v5.36.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

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.
○ → EXP-8
```

terraform validate:

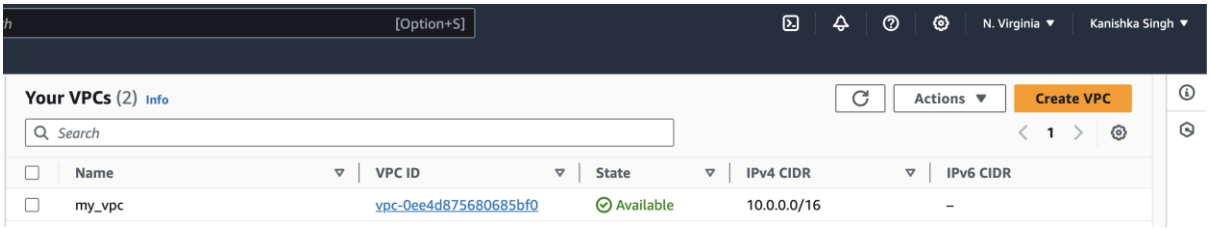
```
● → EXP-8  terraform validate
Success! The configuration is valid.
```

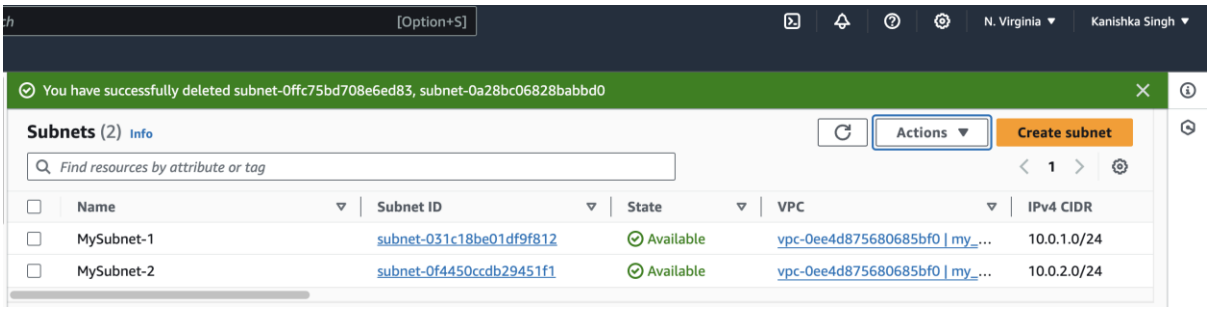
terraform apply:

```
● → EXP-8  terraform apply
aws_vpc.my_vpc: Refreshing state... [id=vpc-0ebfd7d33d4d26fa7]

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last "terraform apply" which
may have affected this plan:

  # aws_vpc.my_vpc has been deleted
  - resource "aws_vpc" "my_vpc" {
      - id                                   = "vpc-0ebfd7d33d4d26fa7" -> null
      - tags                                 = {
          "Name" = "my_vpc"
        }
        # (15 unchanged attributes hidden)
    }


Unless you have made equivalent changes to your configuration, or ignored the relevant attributes using
ignore_changes, the following plan may include actions to undo or respond to these changes.

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

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

Terraform will perform the following actions:
```

```
      + owner_id                             = (known after apply)
      + tags                                 = {
          + "Name" = "my_vpc"
        }
      + tags_all                             = {
          + "Name" = "my_vpc"
        }
    }

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.my_vpc: Creating...
aws_vpc.my_vpc: Still creating... [10s elapsed]
aws_vpc.my_vpc: Creation complete after 17s [id=vpc-0e27d65d1e795867b]
aws_subnet.my_subnet[0]: Creating...
aws_subnet.my_subnet[1]: Creating...
aws_subnet.my_subnet[1]: Still creating... [10s elapsed]
aws_subnet.my_subnet[0]: Still creating... [10s elapsed]
aws_subnet.my_subnet[1]: Creation complete after 13s [id=subnet-0262d72ce6ba8c547]
aws_subnet.my_subnet[0]: Creation complete after 13s [id=subnet-0b543e0f729b5ece6]

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

3. **Verify the vpc in AWS Console.**





4. **Clean up the resources.**

   terraform destroy:

```
→  EXP-8  terraform destroy
aws_vpc.my_vpc: Refreshing state... [id=vpc-0e27d65d1e795867b]
aws_subnet.my_subnet[1]: Refreshing state... [id=subnet-0262d72ce6ba8c547]
aws_subnet.my_subnet[0]: Refreshing state... [id=subnet-0b543e0f729b5ece6]

Terraform used the selected providers to generate the following execution plan. Resource actions are
indicated with the following symbols:
  - destroy

Terraform will perform the following actions:

  # aws_subnet.my_subnet[0] will be destroyed
  - resource "aws_subnet" "my_subnet" {
      - arn                                            = "arn:aws:ec2:us-east-1:220886439816:subnet/s
-0b543e0f729b5ece6" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "us-east-1a" -> null
      - availability_zone_id                           = "use1-az6" -> null
      - cidr_block                                     = "10.0.1.0/24" -> null
      - enable_dns64                                   = false -> null
      - enable_lni_at_device_index                     = 0 -> null
      - enable_resource_name_dns_a_record_on_launch    = false -> null
      - enable_resource_name_dns_aaaa_record_on_launch = false -> null
      - id                                             = "subnet-0b543e0f729b5ece6" -> null
```

```
Plan: 0 to add, 0 to change, 3 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_subnet.my_subnet[0]: Destroying... [id=subnet-0b543e0f729b5ece6]
aws_subnet.my_subnet[1]: Destroying... [id=subnet-0262d72ce6ba8c547]
aws_subnet.my_subnet[1]: Destruction complete after 4s
aws_subnet.my_subnet[0]: Destruction complete after 5s
aws_vpc.my_vpc: Destroying... [id=vpc-0e27d65d1e795867b]
aws_vpc.my_vpc: Destruction complete after 3s

Destroy complete! Resources: 3 destroyed.
→  EXP-8
```