Lab Exercise 8– Creating a VPC in Terraform
Objective:

Objective:

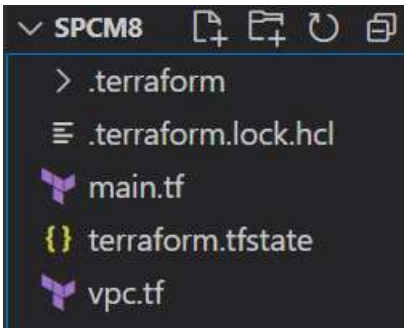
Learn how to use Terraform to create a basic Virtual Private Cloud (VPC) in AWS.

Prerequisites:

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

Steps:

1. Create a Terraform Directory:



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

main.tf

main.tf

main.tf > provider "aws"

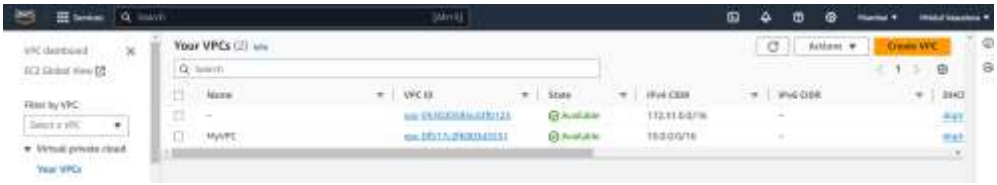
```hcl
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.31.0"
    }
  }
}

provider "aws" {
    region = "ap-south-1"
    access_key = "AKIAZI2LIAJGSHGMMMHP"
    secret_key = "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"
}
```

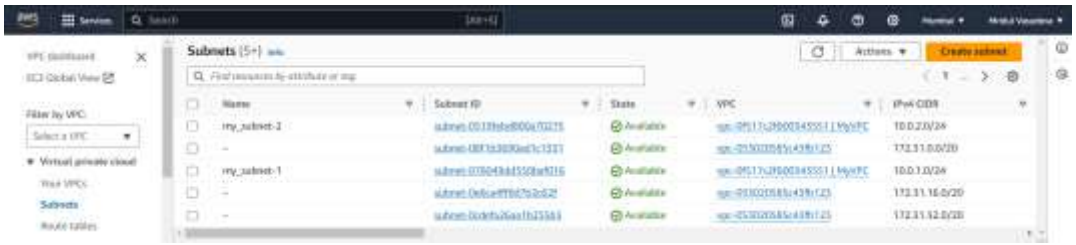
2. Initialize and Apply:

```
PS E:\Desktop\DevOps\SPCMB> terraform plan

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:

  # aws_subnet.my_subnet[0] will be created
  + resource "aws_subnet" "my_subnet" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = (known after apply)
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.1.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags                                           = {
          + "Name" = "my_subnet-1"
        }
      + tags_all                                       = {
          + "Name" = "my_subnet-1"
        }
      + vpc_id                                         = (known after apply)
    }
```

```
PS E:\Desktop\DevOps\SPCMB> terraform apply -auto-approve

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:

  # aws_subnet.my_subnet[0] will be created
  + resource "aws_subnet" "my_subnet" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = (known after apply)
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.1.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags                                           = {
          + "Name" = "my_subnet-1"
        }
      + tags_all                                       = {
          + "Name" = "my_subnet-1"
        }
      + vpc_id                                         = (known after apply)
    }
```

3. Verify Resources in AWS Console:





5. Clean Up:

```
PS E:\Desktop\DevOps\SPCM8> terraform destroy -auto-approve
aws_vpc.my_vpc: Refreshing state... [id=vpc-0f517c2f600343551]
aws_subnet.my_subnet[0]: Refreshing state... [id=subnet-076049dd350baf016]
aws_subnet.my_subnet[1]: Refreshing state... [id=subnet-0519febd800a70275]

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:ap-south-1:637423583821:subnet/subnet-076049dd350baf016" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "ap-south-1c" -> null
      - availability_zone_id                           = "aps1-az2" -> 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-076049dd350baf016" -> null
      - ipv6_native                                    = false -> null
      - map_customer_owned_ip_on_launch                = false -> null
      - map_public_ip_on_launch                        = false -> null
      - owner_id                                       = "637423583821" -> null
      - private_dns_hostname_type_on_launch            = "ip-name" -> null
      - tags                                           = {
          - "Name" = "my_subnet-1"
        } -> null
      - tags_all                                       = {
          - "Name" = "my_subnet-1"
```

```
Plan: 0 to add, 0 to change, 3 to destroy.
aws_subnet.my_subnet[1]: Destroying... [id=subnet-0519febd800a70275]
aws_subnet.my_subnet[0]: Destroying... [id=subnet-076049dd350baf016]
aws_subnet.my_subnet[1]: Destruction complete after 2s
aws_subnet.my_subnet[0]: Destruction complete after 2s
aws_vpc.my_vpc: Destroying... [id=vpc-0f517c2f600343551]
aws_vpc.my_vpc: Still destroying... [id=vpc-0f517c2f600343551, 10s elapsed]
aws_vpc.my_vpc: Destruction complete after 16s

Destroy complete! Resources: 3 destroyed.
```


Your VPCs (1)
Actions
Create VPC
Name
VPC ID
State
IPv4 CIDR
IPv6 CIDR
172.31.0.0/16


Subnets (3)
Actions
Create subnet
Name
Subnet ID
State
VPC
IPv4 CIDR
172.31.0.0/20
172.31.16.0/20
172.31.32.0/20