LAB-8

Creating a VPC in Terraform

Step 1: Create a Terraform-VPC Directory

Step 2: Create a main.tf

main.tf

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "5.35.0"
    }
  }
}

provider "aws" {
  region ="ap-south-1"
  access_key = "AKIA2FD5AJBR2NUSDMTH"
  secret_key = "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"
}

resource "aws_vpc" "my_vpc" {
  cidr_block           = "10.0.0.0/16"
  enable_dns_support   = true
  enable_dns_hostnames = true

  tags = {
    Name = "MyVPC"
  }
}

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-1a"
  map_public_ip_on_launch = true

  tags = {
    Name = "MySubnet-${count.index + 1}"
  }
}
```

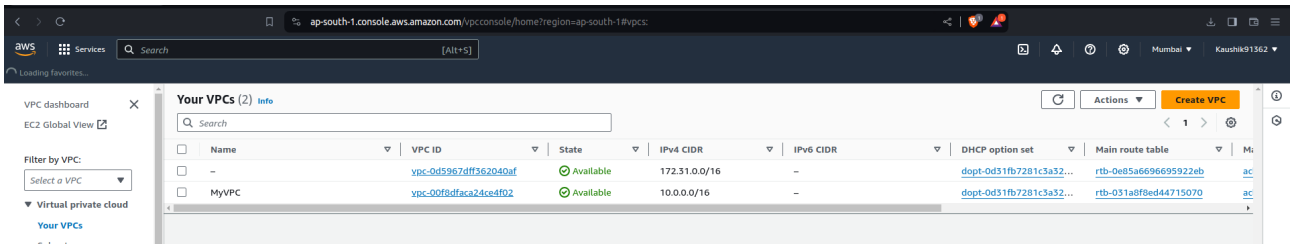
Step 3: Initialize and Plan

```
~/Documents/SPCM/Terraform/Terraform-VPC  v1.7.1default as 💻
→ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.35.0"...
- Installing hashicorp/aws v5.35.0...
- Installed hashicorp/aws v5.35.0 (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.

~/Documents/SPCM/Terraform/Terraform-VPC  v1.7.1default as 💻 took 11s
→ 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                              = "ap-south-1"
      + 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                        = true
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags                                           = {
          + "Name" = "MySubnet-1"
        }
```

Step 4: Apply terraform

```
~/Documents/SPCM/Terraform/Terraform-VPC  v1.7.1default as 💻 took 2s
→ terraform apply

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                              = "ap-south-1"
      + 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                        = true
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags                                           = {
          + "Name" = "MySubnet-1"
        }
      + tags_all                                       = {
          + "Name" = "MySubnet-1"
        }
      + vpc_id                                         = (known after apply)
    }

  # aws_subnet.my_subnet[1] will be created
  + resource "aws_subnet" "my_subnet" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "ap-south-1"
```

Step 5: Verify recources in AWS Console



Step 7: Clean Up

```
~/Documents/SPCM/Terraform/Terraform-VPC  v1.7.1default as 💻 took 16s
➜ terraform destroy
aws_vpc.my_vpc: Refreshing state... [id=vpc-00f8dfaca24ce4f02]
aws_subnet.my_subnet[0]: Refreshing state... [id=subnet-000c41c7bec298186]
aws_subnet.my_subnet[1]: Refreshing state... [id=subnet-0887264a09b58a064]

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-1:698194348131:subnet/subnet-000c41c7bec298186" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "ap-south-1a" -> null
      - availability_zone_id                           = "aps1-az1" -> 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-000c41c7bec298186" -> null
      - ipv6_native                                    = false -> null
      - map_customer_owned_ip_on_launch                = false -> null
      - map_public_ip_on_launch                        = true -> null
      - owner_id                                       = "698194348131" -> null
      - private_dns_hostname_type_on_launch            = "ip-name" -> null
      - tags                                           = {
          - "Name" = "MySubnet-1"
        } -> null
      - tags_all                                       = {
          - "Name" = "MySubnet-1"
        } -> null
      - vpc_id                                         = "vpc-00f8dfaca24ce4f02" -> null
    }

  # aws_subnet.my_subnet[1] will be destroyed
  - resource "aws_subnet" "my_subnet" {
      - arn                                            = "arn:aws:ec2:ap-south-1:698194348131:subnet/subnet-0887264a09b58a064" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "ap-south-1a" -> null
      - availability_zone_id                           = "aps1-az1" -> null
```

If you want to modify the VPC configuration, update the main.tf file with desired changes and rerun the terraform apply command to apply changes.