# Lab Exercise 8– Creating a VPC in Terraform


## Anmol Ghai

## 500090959, R2142210118

## B. Tech. CSE-DevOps – B1, 6th SEM


1. Create Terraform Configuration Files

TERRAFORM-VPC > main.tf > terraform

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

provider "aws" {
    region = "ap-south-1"
    access_key = "AKIATQMU37PQBDMOBLE2"
    secret_key = "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"
}
```

TERRAFORM-VPC > var.tf > variable "instance_type"

```
variable "ami" {
 type = string
 default = "ami-03f4878755434977f"
}

variable "instance_type" {
 type = string
 default = "t2.micro"
}
```

2. Define a VPC with a specified CIDR block, and two subnets within the VPC

TERRAFORM-VPC > resources.tf > resource "aws_subnet" "my_subnet"

```hcl
resource "aws_instance" "UPES" {
 ami = var.ami
 instance_type = var.instance_type

 tags = {
    Name = "EC2-Instnace"
    }
}

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}"
  }
}
```

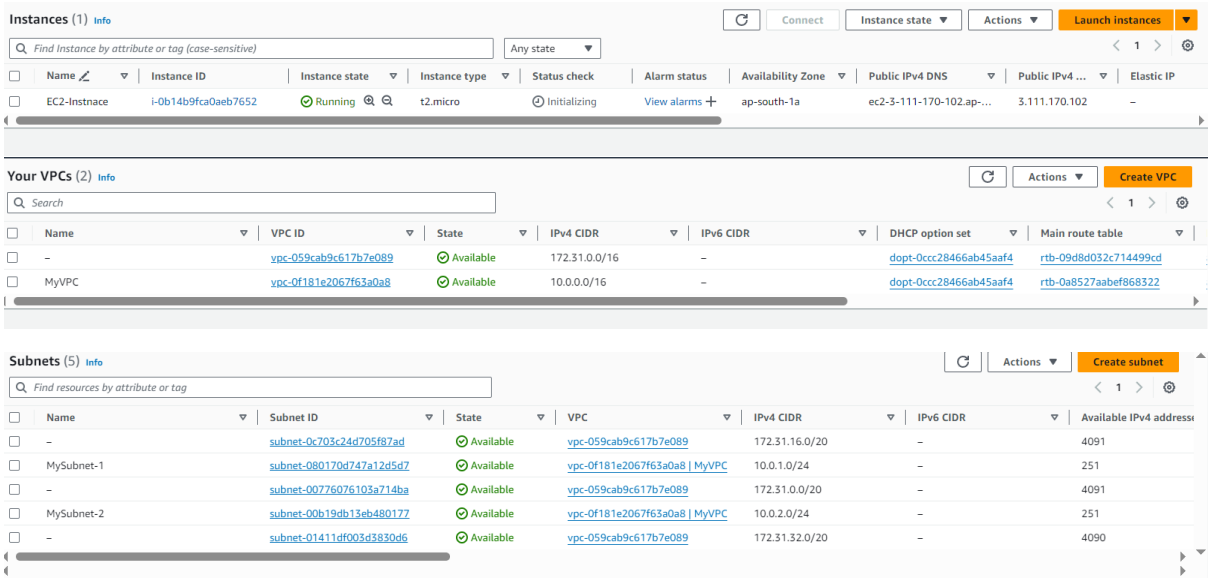
3. Initialize & Apply

```
      + public_dns                          = (known after apply)
      + public_ip                           = (known after apply)
      + secondary_private_ips               = (known after apply)
      + security_groups                     = (known after apply)
      + source_dest_check                   = true
      + spot_instance_request_id            = (known after apply)
      + subnet_id                           = (known after apply)
      + tags                                = {
          + "Name" = "My-EC2-Instance"
        }
      + tags_all                            = {
          + "Name" = "My-EC2-Instance"
        }
      + tenancy                             = (known after apply)
      + user_data                           = (known after apply)
      + user_data_base64                    = (known after apply)
      + user_data_replace_on_change         = false
      + vpc_security_group_ids              = (known after apply)
    }

Plan: 4 to add, 0 to change, 0 to destroy.
aws_iam_user.iam_users[1]: Creating...
aws_iam_user.iam_users[0]: Creating...
aws_iam_user.iam_users[2]: Creating...
aws_instance.UPES: Creating...
aws_iam_user.iam_users[1]: Creation complete after 1s [id=user2]
aws_iam_user.iam_users[2]: Creation complete after 1s [id=user3]
aws_iam_user.iam_users[0]: Creation complete after 1s [id=user1]
aws_instance.UPES: Still creating... [10s elapsed]
aws_instance.UPES: Still creating... [20s elapsed]
aws_instance.UPES: Still creating... [30s elapsed]
aws_instance.UPES: Creation complete after 32s [id=i-0378e8cdaa8cc8659]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
PS D:\DevOps\LAB\SPCM\TERRAFORM-SCRIPTS\TERRAFORM-IAM-USER>
```

## 4. Check the AWS Console




## 5. Clean Up

```
PS D:\DevOps\LAB\SPCM\TERRAFORM-SCRIPTS\TERRAFORM-VPC> terraform destroy -auto-approve
aws_vpc.my_vpc: Refreshing state... [id=vpc-0f181e2067f63a0a8]
aws_instance.UPES: Refreshing state... [id=i-0b14b9fca0aeb7652]
aws_subnet.my_subnet[1]: Refreshing state... [id=subnet-00b19db13eb480177]
aws_subnet.my_subnet[0]: Refreshing state... [id=subnet-080170d747a12d5d7]

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_instance.UPES will be destroyed
  - resource "aws_instance" "UPES" {
      - ami                                  = "ami-03f4878755434977f" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:241367251936:instance/i-0b14b9fca0aeb7652" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "ap-south-1a" -> null
      - cpu_core_count                       = 1 -> null
      - cpu_threads_per_core                 = 1 -> null
      - disable_api_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
```

```
Plan: 0 to add, 0 to change, 4 to destroy.
aws_subnet.my_subnet[1]: Destroying... [id=subnet-00b19db13eb480177]
aws_subnet.my_subnet[0]: Destroying... [id=subnet-080170d747a12d5d7]
aws_instance.UPES: Destroying... [id=i-0b14b9fca0aeb7652]
aws_subnet.my_subnet[1]: Destruction complete after 1s
aws_subnet.my_subnet[0]: Destruction complete after 1s
aws_vpc.my_vpc: Destroying... [id=vpc-0f181e2067f63a0a8]
aws_vpc.my_vpc: Destruction complete after 1s
aws_instance.UPES: Still destroying... [id=i-0b14b9fca0aeb7652, 10s elapsed]
aws_instance.UPES: Still destroying... [id=i-0b14b9fca0aeb7652, 21s elapsed]
aws_instance.UPES: Still destroying... [id=i-0b14b9fca0aeb7652, 31s elapsed]
aws_instance.UPES: Destruction complete after 31s

Destroy complete! Resources: 4 destroyed.
PS D:\DevOps\LAB\SPCM\TERRAFORM-SCRIPTS\TERRAFORM-VPC>
```