SPCM Lab-8

Objective : Creating a VPC in Terraform

1. Create Terraform directory.

```
vidhant@psyches-safehouse:~$ mkdir terraform-vpc
vidhant@psyches-safehouse:~$ cd terraform-vpc/
vidhant@psyches-safehouse:~/terraform-vpc$
```

2. Create terraform configuration file (main.tf) :

dev.tfvars | main.tf 2

Lab-8 > main.tf > resource "aws_subnet" "my_subnet" > tags > Name

```
provider "aws" {
  region = "us-east-1"
  access_key = "AKIA232UVZYDK5TANG62"
  secret_key = "47IqpUlOzW5Q3cw6KrCxPQrbQ5M/hajeNL3wxE
}

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 = "us-east-1a"
  map_public_ip_on_launch = true

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

3. Initialize, validate and Apply :

terraform init :

```
vidhant@psyches-safehouse:~/Documents/Terraform/Lab-7$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.35.0...
```

terraform validate :

```
vidhant@psyches-safehouse:~/Documents/Terraform/Lab-7$ terraform v
alidate
Success! The configuration is valid.

vidhant@psyches-safehouse:~/Documents/Terraform/Lab-7$
```

terraform apply :

```
vidhant@psyches-safehouse:~/Documents/Terraform/Lab-8$ 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                              = "us-east-1a"
      + availability_zone_id                           = (known after apply)
```

4. Verify Users in AWS console :

Name	VPC ID	State	IPv4 CIDR
–	vpc-02c4e50a1abfffd3b	Available	172.31.0.0/16
my_vpc	vpc-0c5f757b483b4d7ed	Available	10.0.0.0/16

Name	Subnet ID	State
MySubnet-2	subnet-0716cca6ad5e73a74	Available
MySubnet-1	subnet-06be928c05048e8cf	Available

5. Clean up Resources (terraform destroy) :

```
vidhant@psyches-safehouse:~/Documents/Terraform/Lab-8$ terraform destroy
aws_vpc.my_vpc: Refreshing state... [id=vpc-0c5f757b483b4d7ed]
aws_subnet.my_subnet[0]: Refreshing state... [id=subnet-06be928c05048e8cf]
aws_subnet.my_subnet[1]: Refreshing state... [id=subnet-0716cca6ad5e73a74]

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" {
```