

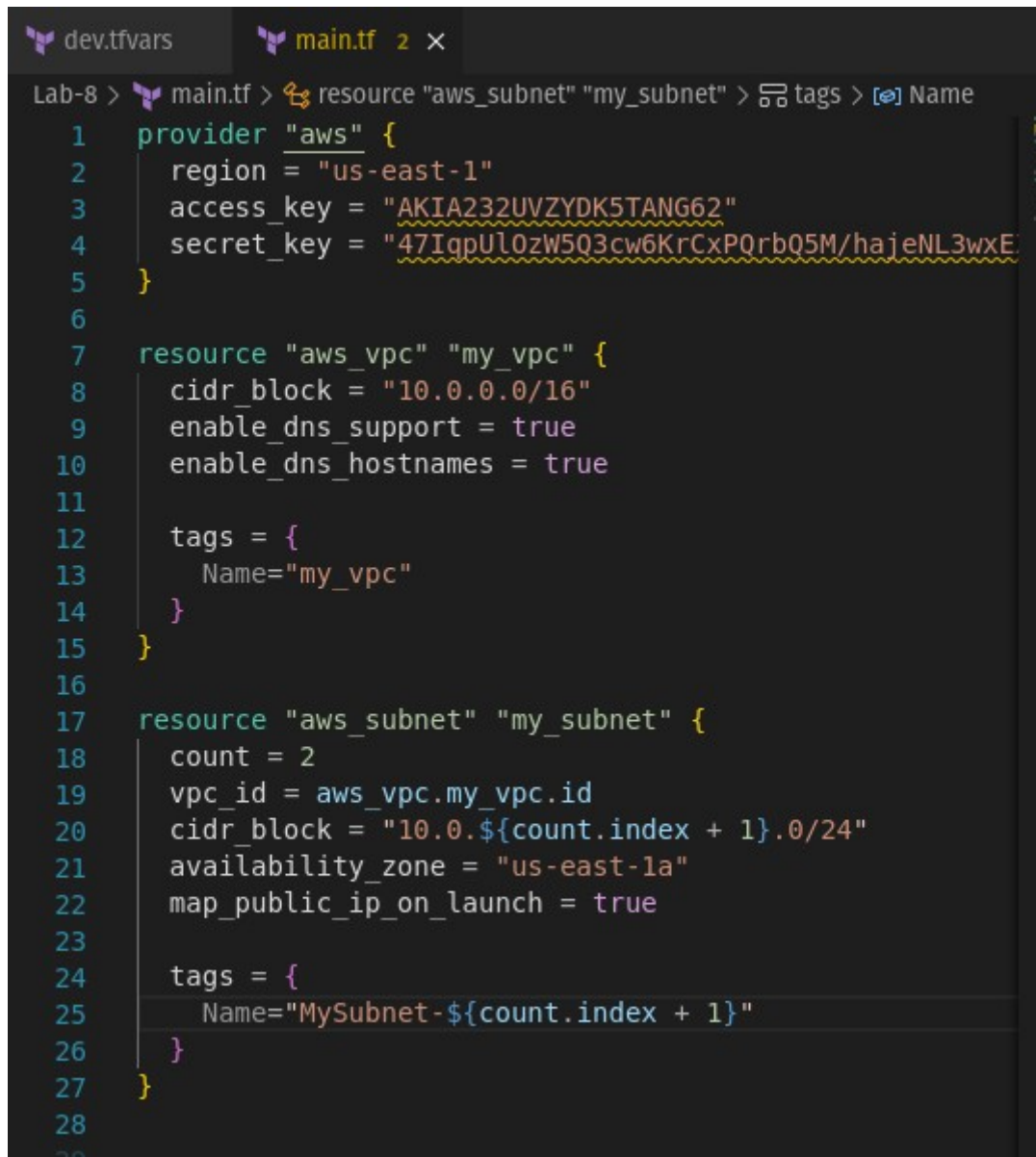
# SPCM Lab-8

## 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) :



The screenshot shows a code editor with two tabs: 'dev.tfvars' and 'main.tf'. The 'main.tf' tab is active, displaying Terraform configuration code. The code defines an AWS provider, an AWS VPC resource named 'my\_vpc', and an AWS Subnet resource named 'my\_subnet' with a count of 2. The VPC is configured with region 'us-east-1', CIDR block '10.0.0.0/16', and DNS support enabled. The Subnets are configured with the VPC ID, a variable CIDR block, availability zone 'us-east-1a', and public IP on launch enabled. Both resources are tagged with 'Name'.

```
Lab-8 > main.tf > resource "aws_subnet" "my_subnet" > tags > Name
1  provider "aws" {
2      region = "us-east-1"
3      access_key = "AKIA232UVZYDK5TANG62"
4      secret_key = "47IqpUl0zW503cw6KrCxP0rb05M/hajeNL3wxE"
5  }
6
7  resource "aws_vpc" "my_vpc" {
8      cidr_block = "10.0.0.0/16"
9      enable_dns_support = true
10     enable_dns_hostnames = true
11
12     tags = {
13         Name="my_vpc"
14     }
15 }
16
17 resource "aws_subnet" "my_subnet" {
18     count = 2
19     vpc_id = aws_vpc.my_vpc.id
20     cidr_block = "10.0.${count.index + 1}.0/24"
21     availability_zone = "us-east-1a"
22     map_public_ip_on_launch = true
23
24     tags = {
25         Name="MySubnet-${count.index + 1}"
26     }
27 }
28
29
```

### 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 validate
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 :

<input type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	
<input type="checkbox"/>	-	<a href="#">vpc-02c4e50a1abfffd3b</a>	✓ Available	172.31.0.0/16	-
<input type="checkbox"/>	my_vpc	<a href="#">vpc-0c5f757b483b4d7ed</a>	✓ Available	10.0.0.0/16	-

<input type="checkbox"/>	Name	Subnet ID	State
<input type="checkbox"/>	MySubnet-2	<a href="#">subnet-0716cca6ad5e73a74</a>	✓ Available
<input type="checkbox"/>	MySubnet-1	<a href="#">subnet-06be928c05048e8cf</a>	✓ 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" {
```