

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:

```
mkdir terraform-vpc
cd terraform-vpc
```

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

main.tf

```
provider "aws" {
  region = "us-east-1"
}

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 = "us-east-1a"  
  map_public_ip_on_launch = true  
  
  tags = {  
    Name = "MySubnet-${count.index + 1}"  
  }  
}
```

In this configuration, we define an AWS provider, a VPC with a specified CIDR block, and two subnets within the VPC.

2. Initialize and Apply:

- Run the following Terraform commands to initialize and apply the configuration:

```
terraform init  
terraform apply
```

- Terraform will prompt you to confirm the creation of the VPC and subnets. Type yes and press Enter.

3. Verify Resources in AWS Console:

- Log in to the AWS Management Console and navigate to the VPC service.
- Verify that the VPC and subnets with the specified names and settings have been created.

4. Update VPC Configuration:

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

```
terraform apply
```

5. Clean Up:

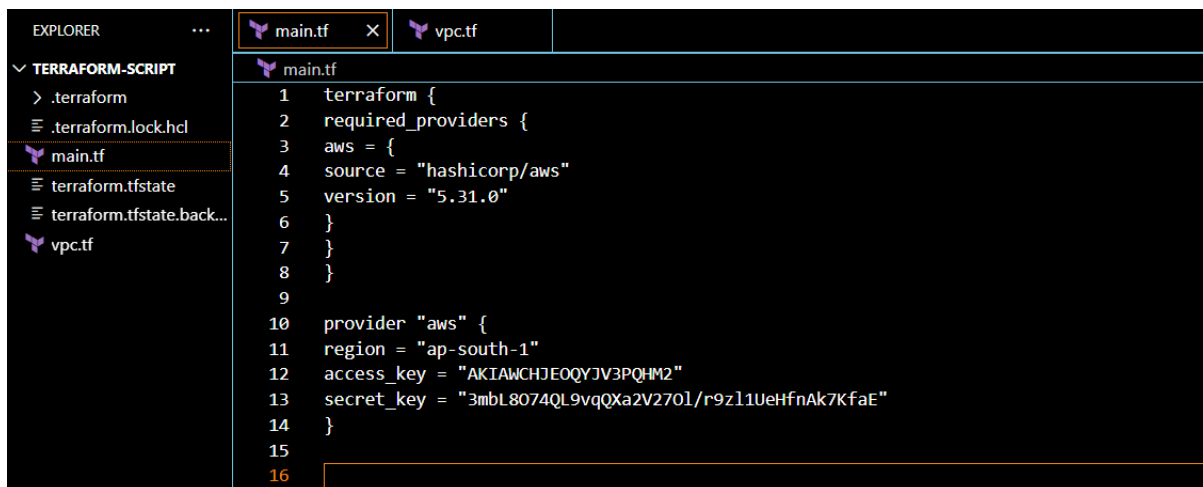
After testing, you can clean up the VPC and subnets:

```
terraform destroy
```

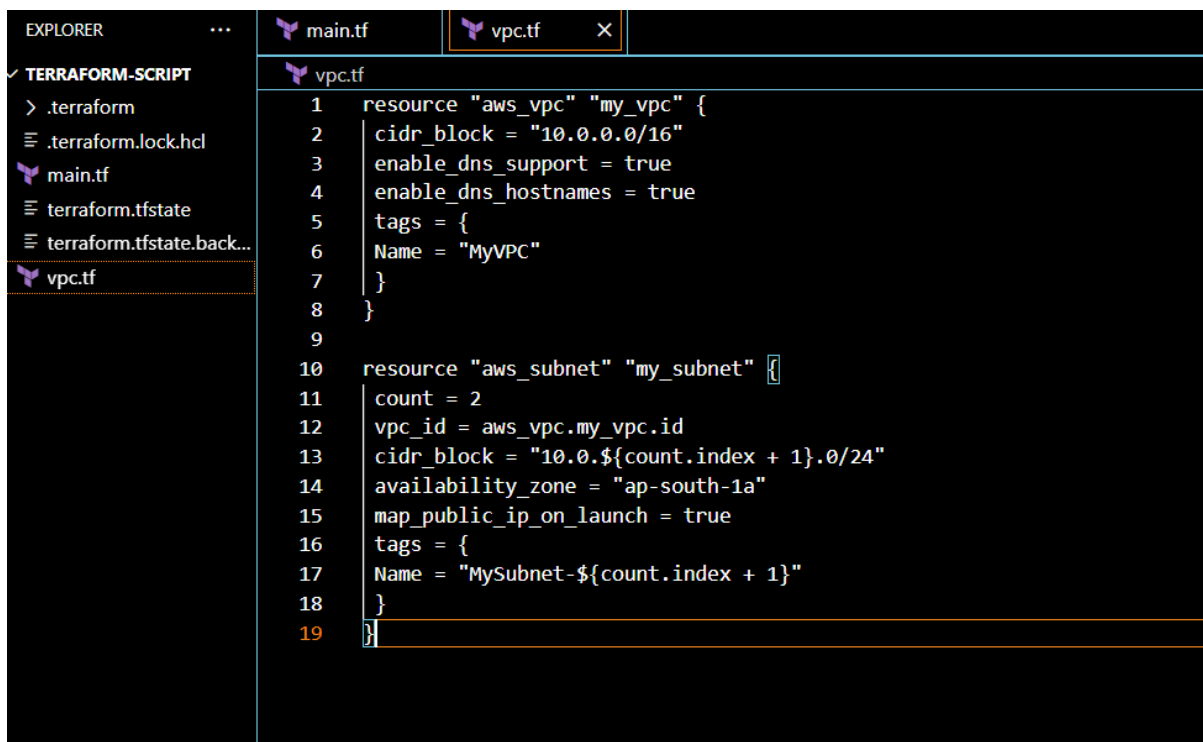
Confirm the destruction by typing yes.

6. Conclusion:

This lab exercise demonstrates how to create a basic Virtual Private Cloud (VPC) with subnets in AWS using Terraform. The example includes a simple VPC configuration with two subnets. Experiment with different CIDR blocks, settings, and additional AWS resources to customize your VPC.



```
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.31.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   region = "ap-south-1"
12   access_key = "AKIAWCHJEQYJV3PQH2"
13   secret_key = "3mbL8074QL9vqQXa2V270l/r9z11UeHfnAk7KfaE"
14 }
15
16
```



```
1 resource "aws_vpc" "my_vpc" {
2   cidr_block = "10.0.0.0/16"
3   enable_dns_support = true
4   enable_dns_hostnames = true
5   tags = {
6     Name = "MyVPC"
7   }
8 }
9
10 resource "aws_subnet" "my_subnet" [
11   count = 2
12   vpc_id = aws_vpc.my_vpc.id
13   cidr_block = "10.0.${count.index + 1}.0/24"
14   availability_zone = "ap-south-1a"
15   map_public_ip_on_launch = true
16   tags = {
17     Name = "MySubnet-${count.index + 1}"
18   }
19 }
```

```
C:\Users\anu39>cd C:\Users\anu39\Terraform-Script
C:\Users\anu39\Terraform-Script>terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.31.0"...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (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.

C:\Users\anu39\Terraform-Script>terraform validate
Success! The configuration is valid.
```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

- + create

```
# 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"
+   availability_zone_id              = (known after apply)
+   cidr_block                        = "10.0.2.0/24"
```

```
+ cidr_block = "10.0.2.0/24"
+ 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-2"
+ }
+ tags_all = {
+   + "Name" = "MySubnet-2"
+ }
+ vpc_id = (known after apply)
}
```

[illegible]

```

+ default_route_table_id      = (known after apply)
+ default_security_group_id   = (known after apply)
+ dhcp_options_id             = (known after apply)
+ enable_dns_hostnames         = true
+ enable_dns_support           = true
+ enable_network_address_usage_metrics = (known after apply)
+ id                           = (known after apply)
+ instance_tenancy             = "default"
+ ipv6_association_id          = (known after apply)
+ ipv6_cidr_block              = (known after apply)
+ ipv6_cidr_block_network_border_group = (known after apply)
+ main_route_table_id          = (known after apply)
+ owner_id                     = (known after apply)
+ tags                         = {
  + "Name" = "MyVPC"
}
+ tags_all                     = {
  + "Name" = "MyVPC"
}
}

```

Plan: 3 to add, 0 to change, 0 to destroy.

aws_vpc.my_vpc: Creating...

aws_vpc.my_vpc: Still creating... [10s elapsed]

aws_vpc.my_vpc: Creation complete after 12s [id=vpc-00721de38432b6ca5]

aws_subnet.my_subnet[0]: Creating...

aws_subnet.my_subnet[1]: Creating...

aws_subnet.my_subnet[1]: Still creating... [10s elapsed]

aws_subnet.my_subnet[0]: Still creating... [10s elapsed]

aws_subnet.my_subnet[0]: Creation complete after 11s [id=subnet-0101ced7e2a85494f]

aws_subnet.my_subnet[1]: Creation complete after 11s [id=subnet-0704b115c4afedd3a]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Search [Alt+S]

Subnets (5) Info

Find resources by attribute or tag

Actions Create subnet

<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR
<input type="checkbox"/>	-	subnet-0ccdc6856663c4701	Available	vpc-0e4fe532fd5496e97	172.31.16.0/20
<input type="checkbox"/>	-	subnet-0b40c3b2a9aa1dd07	Available	vpc-0e4fe532fd5496e97	172.31.0.0/20
<input type="checkbox"/>	-	subnet-0bf96d4af079e8d53	Available	vpc-0e4fe532fd5496e97	172.31.32.0/20
<input type="checkbox"/>	MySubnet-1	subnet-0101ced7e2a85494f	Available	vpc-00721de38432b6ca5 MyV...	10.0.1.0/24
<input type="checkbox"/>	MySubnet-2	subnet-0704b115c4afedd3a	Available	vpc-00721de38432b6ca5 MyV...	10.0.2.0/24

Search [Alt+S]

Your VPCs (1/2) Info

Search

Actions Create VPC

<input checked="" type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP o
<input checked="" type="checkbox"/>	MyVPC	vpc-00721de38432b6ca5	Available	10.0.0.0/16	-	dopt-05

```
C:\Users\anu39\Terraform-Script>terraform destroy -auto-approve
aws_vpc.my_vpc: Refreshing state... [id=vpc-00721de38432b6ca5]
aws_subnet.my_subnet[1]: Refreshing state... [id=subnet-0704b115c4afedd3a]
aws_subnet.my_subnet[0]: Refreshing state... [id=subnet-0101ced7e2a85494f]

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-1:417100756016:subnet/subnet-0101ced7e2a85494f" -> 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-0101ced7e2a85494f" -> null
  - ipv6_native                        = false -> null
  - map_customer_owned_ip_on_launch    = false -> null
  - map_public_ip_on_launch            = true -> null
  - owner_id                           = "417100756016" -> null
  - private_dns_hostname_type_on_launch = "ip-name" -> null
  - tags                               = {
    - "Name" = "MySubnet-1"
  } -> null
  - tags_all                           = {
    - "Name" = "MySubnet-1"
  } -> null
  - vpc_id                             = "vpc-00721de38432b6ca5" -> null
}

# aws_subnet.my_subnet[1] will be destroyed
- resource "aws_subnet" "my_subnet" {
```

```
# aws_subnet.my_subnet[1] will be destroyed
- resource "aws_subnet" "my_subnet" {
  - arn                                = "arn:aws:ec2:ap-south-1:417100756016:subnet/subnet-0704b115c4afedd3a" -> null
  - assign_ipv6_address_on_creation    = false -> null
  - availability_zone                  = "ap-south-1a" -> null
  - availability_zone_id                = "aps1-az1" -> null
  - cidr_block                         = "10.0.2.0/24" -> 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-0704b115c4afedd3a" -> null
  - ipv6_native                        = false -> null
  - map_customer_owned_ip_on_launch    = false -> null
  - map_public_ip_on_launch            = true -> null
  - owner_id                           = "417100756016" -> null
  - private_dns_hostname_type_on_launch = "ip-name" -> null
  - tags                               = {
    - "Name" = "MySubnet-2"
  } -> null
  - tags_all                           = {
    - "Name" = "MySubnet-2"
  } -> null
  - vpc_id                             = "vpc-00721de38432b6ca5" -> null
}

# aws_vpc.my_vpc will be destroyed
- resource "aws_vpc" "my_vpc" {
  - arn                                = "arn:aws:ec2:ap-south-1:417100756016:vpc/vpc-00721de38432b6ca5" -> null
  - assign_generated_ipv6_cidr_block    = false -> null
  - cidr_block                         = "10.0.0.0/16" -> null
  - default_network_acl_id              = "acl-09bd37eaaef0272f6" -> null
  - default_route_table_id              = "rtb-0488d81d0aeb52748" -> null
  - default_security_group_id           = "sg-06aff075e816db355" -> null
  - dhcp_options_id                     = "dopt-09933feda09ff0cf5" -> null
  - enable_dns_hostnames                = true -> null
  - enable_dns_support                  = true -> null
  - enable_network_address_usage_metrics = false -> null
  - id                                 = "vpc-00721de38432b6ca5" -> null
  - instance_tenancy                    = "default" -> null
}
```

```
- dhcp_options_id = "dopc-09933feda097f0c75" -> null
- enable_dns_hostnames = true -> null
- enable_dns_support = true -> null
- enable_network_address_usage_metrics = false -> null
- id = "vpc-00721de38432b6ca5" -> null
- instance_tenancy = "default" -> null
- ipv6_netmask_length = 0 -> null
- main_route_table_id = "rtb-0488d81d0aeb52748" -> null
- owner_id = "417100756016" -> null
- tags = {
  - "Name" = "MyVPC"
} -> null
- tags_all = {
  - "Name" = "MyVPC"
} -> null
}
```

Plan: 0 to add, 0 to change, 3 to destroy.

aws_subnet.my_subnet[1]: Destroying... [id=subnet-0704b115c4afedd3a]

aws_subnet.my_subnet[0]: Destroying... [id=subnet-0101ced7e2a85494f]

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-00721de38432b6ca5]

aws_vpc.my_vpc: Destruction complete after 0s

Destroy complete! Resources: 3 destroyed.

C:\Users\anu39\Terraform-Script>