Lab-8

Creating a VPC in Terraform

Step 1: Create a Terraform Directory

```
C:\Users\hp>mkdir terrafrom-vpc

C:\Users\hp>cd  terrafrom-vpc
```

main.tf

```
provider "aws" {
 region = "ap-south-1"
 access_key = "AKIAV2D7UZ5ZAAX5TNVG"
 secret_key = "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"
}
variable "iam_users" {
 type = list(string)
 default = ["user1", "user2", "user3"]
}
resource "aws_iam_user" "iam_users" {
 count = length(var.iam_users)
 name = var.iam_users[count.index]
 tags = {
 Name = "${var.iam_users[count.index]}-user"
 }
}
```

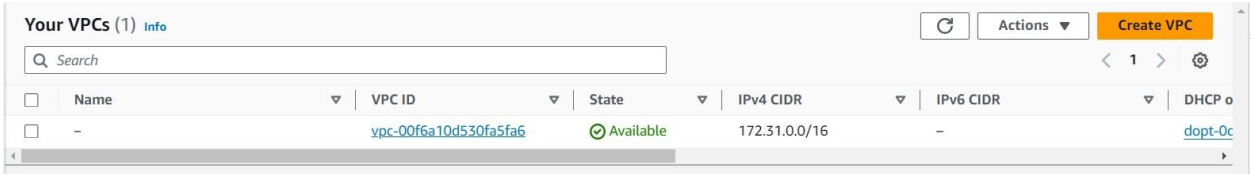
Step 2: Run the following commands

```
C:\Users\hp\terrafrom-vpc>terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.36.0...
- Installed hashicorp/aws v5.36.0 (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\hp\terrafrom-vpc>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)
      + 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"
```

```
      + owner_id                              = (known after apply)
      + tags                                  = {
          + "Name" = "MyVPC"
        }
      + tags_all                              = {
          + "Name" = "MyVPC"
        }
    }

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

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_vpc.my_vpc: Creating...
aws_vpc.my_vpc: Still creating... [10s elapsed]
aws_vpc.my_vpc: Creation complete after 18s [id=vpc-0ccb810e9b0816eb0]
aws_subnet.my_subnet[0]: Creating...
aws_subnet.my_subnet[1]: Creating...
aws_subnet.my_subnet[0]: Still creating... [10s elapsed]
aws_subnet.my_subnet[1]: Still creating... [10s elapsed]
aws_subnet.my_subnet[0]: Creation complete after 15s [id=subnet-0d23ee873bde8cbec]
aws_subnet.my_subnet[1]: Creation complete after 15s [id=subnet-0159de7521427b1d6]

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

Step 3: Verify the Resources



Step 4: Clean up

```
C:\Users\hp\terrafrom-vpc>terraform destroy
aws_vpc.my_vpc: Refreshing state... [id=vpc-0ccb810e9b0816eb0]
aws_subnet.my_subnet[1]: Refreshing state... [id=subnet-0159de7521427b1d6]
aws_subnet.my_subnet[0]: Refreshing state... [id=subnet-0d23ee873bde8cbec]

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:us-east-1:399699660658:subnet/subnet-0d23ee873bde8
cbec" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "us-east-1a" -> null
      - availability_zone_id                           = "use1-az2" -> 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-0d23ee873bde8cbec" -> null
      - ipv6_native                                    = false -> null
      - map_customer_owned_ip_on_launch                = false -> null
      - map_public_ip_on_launch                        = true -> null
      - owner_id                                       = "399699660658" -> null
      - private_dns_hostname_type_on_launch            = "ip-name" -> null
      - tags                                           = {
```

```
      - instance_tenancy                    = "default" -> null
      - ipv6_netmask_length                 = 0 -> null
      - main_route_table_id                 = "rtb-0a25ebb03b7bf05c3" -> null
      - owner_id                            = "399699660658" -> null
      - tags                                = {
          - "Name" = "MyVPC"
        } -> null
      - tags_all                            = {
          - "Name" = "MyVPC"
        } -> null
    }

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

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_subnet.my_subnet[1]: Destroying... [id=subnet-0159de7521427b1d6]
aws_subnet.my_subnet[0]: Destroying... [id=subnet-0d23ee873bde8cbec]
aws_subnet.my_subnet[0]: Destruction complete after 2s
aws_subnet.my_subnet[1]: Destruction complete after 3s
aws_vpc.my_vpc: Destroying... [id=vpc-0ccb810e9b0816eb0]
aws_vpc.my_vpc: Destruction complete after 3s

Destroy complete! Resources: 3 destroyed.
```