UPES

SPCM - LAB
6th Sem

Submitted To:

Dr. Hitesh Kumar Sharma

School of Computer Science

Submitted By:

Ujjwal chauhan

R2142220994

500105692

Batch 2-NH

Writing Terraform Scripts to perform the following task 2 ec2 Instances, VPN and S3
main.tf

```
assignment1 > main.tf > terraform > required_providers > aws
  1    terraform {
  2      required_providers {
  3        aws = {
  4          source  = "hashicorp/aws"
  5          version = "5.31.0"
  6        }
  7      }
  8    }
  9
 10
 11    provider "aws" {
 12      region     = "ap-south-1"   # Replace with your preferred region
 13      access_key = "AKIA4ZZIDPTHCMH23BLX"   # Replace with your Access Key
 14      secret_key = "coOvWKRV+kzN+VTm+O6lxwsbba3P+09FHkcoRKww"   # Replace with your Secret Key
 15    }
 16
```

Running terraform init

```
(base) →  assignment1 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.
(base) →  assignment1
```

Terraform init to initialize the terraform folder which will have the aws provider plugin installed

instance.tf

```
assignment1 >  instance.tf >  resource "aws_instance" "ins2"
  1    resource "aws_instance" "ins1" {
  2      ami              = "ami-0e35ddab05955cf57"
  3      instance_type = "t2.micro"
  4
  5      tags = {        string
  6        Name = "Instance1"
  7      }
  8    }
  9
 10    resource "aws_instance" "ins2" {
 11      ami              = "ami-0e35ddab05955cf57"
 12      instance_type = "t2.micro"
 13
 14      tags = {
 15        Name = "Instance2"
 16      }
 17    }
```

This file holds the iac code to make 2 instances - t2-micro ec2 machines

resource.tf

```
assignment1 >  resource.tf
   1    resource "aws_vpc" "main"{
   2    cidr_block = "10.0.0.0/16"
   3
   4    tags = {
   5      Name ="BhaveshVPC"
   6      }
   7    }
   8
   9
  10    resource "aws_vpn_gateway" "example" {
  11      vpc_id = aws_vpc.main.id
  12
  13      tags = {
  14        Name = "MyVPNGateway"
  15      }
  16    }
  17
```

```
  19    resource "aws_customer_gateway" "example" {
  20      bgp_asn    = 65000
  21      ip_address = "203.0.113.1"  # Replace with actual IP
  22      type       = "ipsec.1"
  23
  24      tags = {
  25        Name = "MyCustomerGateway"
  26      }
  27    }
  28
  29
  30
  31    resource "aws_vpn_connection" "example" {
  32      customer_gateway_id = aws_customer_gateway.example.id
  33      vpn_gateway_id       = aws_vpn_gateway.example.id
  34      type                 = "ipsec.1"
  35      static_routes_only   = true
  36    }
```

This resource.tf hold the iac code to create vpc , the customer gateway and the vpn connection.

s3.tf

```
1
2    resource "aws_s3_bucket" "example" {
3      bucket = "bhavesh-kapur-upes-s3"
4        acl    = "private"
5    }
```

Outputs:

terraform plan

```
# aws_vpn_gateway.example will be created
+ resource "aws_vpn_gateway" "example" {
    + amazon_side_asn = (known after apply)
    + arn             = (known after apply)
    + id              = (known after apply)
    + tags            = {
        + "Name" = "MyVPNGateway"
      }
    + tags_all        = {
        + "Name" = "MyVPNGateway"
      }
    + vpc_id          = (known after apply)
  }

Plan: 7 to add, 0 to change, 0 to destroy.
```

Terraform apply

```
aws_vpn_connection.example: Still creating... [5m0s elapsed]
aws_vpn_connection.example: Creation complete after 5m7s [id=vpn-029da44f6ca1c9be9]

│ Warning: Argument is deprecated
│
│   with aws_s3_bucket.example,
│   on s3.tf line 4, in resource "aws_s3_bucket" "example":
│    4:      acl    = "private"
│
│ Use the aws_s3_bucket_acl resource instead


Apply complete! Resources: 7 added, 0 changed, 0 destroyed.
(base) → assignment1 □
```
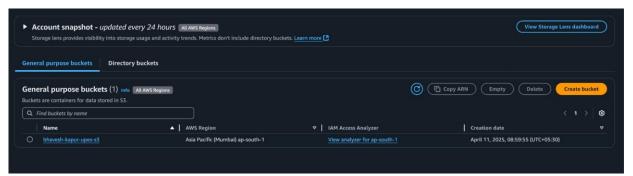
Customer Gateway

Vpc

S3

Instances