System Provisioning And Configuration Management

Assignment 1

Submitted By:

Name: Khushi Jain

Sap id: 500105912

Roll No: R2142220336

Batch: DevOps B1

Assignment 1

Write Terraform script to do perform following tasks on AWS cloud Platform

Step 1: Create two T2 Micro EC2 Instances.

Step2: Create a VPN on AWS

Step 3: Create a S3 Bucket

Step 4: Write the code for step 1,2 and 3 in a IaC terraform file and run terraform commands to execute these steps.

#Main.tf : It has description of the provider and access credentials

#Instance.tf : This file defines the configuration for creating **two EC2 instances**. It includes the AMI ID, instance type (t2.micro), and tags

assignment1 > instance.tf

```
resource "aws_instance" "Khushi-instance-1" {
        ami = "ami-03f4878755434977f"
        instance_type = "t2.micro"
    tags = {
      Name = "EC2-Instance-1"
    }
}

resource "aws_instance" "Khushi-instance-2" {
        ami = "ami-03f4878755434977f"
        instance_type = "t2.micro"
    tags = {
      Name = "EC2-Instance-2"
    }
}
```

#S3.tf : Contains the configuration to create an S3 bucket.

assignment1 > s3.tf

```
resource "aws_s3_bucket" "my_bucket" {
  bucket = "demo-s3-bucket-khushi"
  tags = {
    Name = "Terraform-S3-Bucket"
    }
}
```

#Vpn.tf:

Contains the config to handle the setup of the VPN infrastructure, including a vpc, subnet, Customer Gateway ,a Virtual Private Gateway, and the VPN Connection.

assignment1 > vpn.tf

```
resource "aws_vpc" "my-vpc" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_subnet" "my-subnet" {
  vpc_id     = aws_vpc.my-vpc.id
  cidr_block = "10.0.1.0/24"

  tags = {
    Name = "my-subnet"
  }
}

resource "aws_internet_gateway" "my-gw" {
  vpc_id = aws_vpc.my-vpc.id

  tags = {
    Name = "my-IG"
  }
}

resource "aws_customer_gateway" "customer_gw" {
  bgp_asn    = 65000
  ip_address = "203.0.113.12"
  type       = "ipsec.1"
}

resource "aws_vpn_gateway" "vpn_gw" {
  vpc_id = aws_vpc.my-vpc.id
}

resource "aws_vpn_connection" "vpn_connection" {
  customer_gateway_id = aws_customer_gateway.customer_gw.id
  vpn_gateway_id      = aws_vpn_gateway.vpn_gw.id
  type                = "ipsec.1"
  static_routes_only  = true
}
```

Running Commands to execute the steps:

1. Terraform init: initialises the project

```
C:\Users\KHUSHI JAIN\OneDrive\Desktop\spcmLab\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.

C:\Users\KHUSHI JAIN\OneDrive\Desktop\spcmLab\assignment1>
```

2. Terraform validate : To check the configuration for syntax errors and validates the format

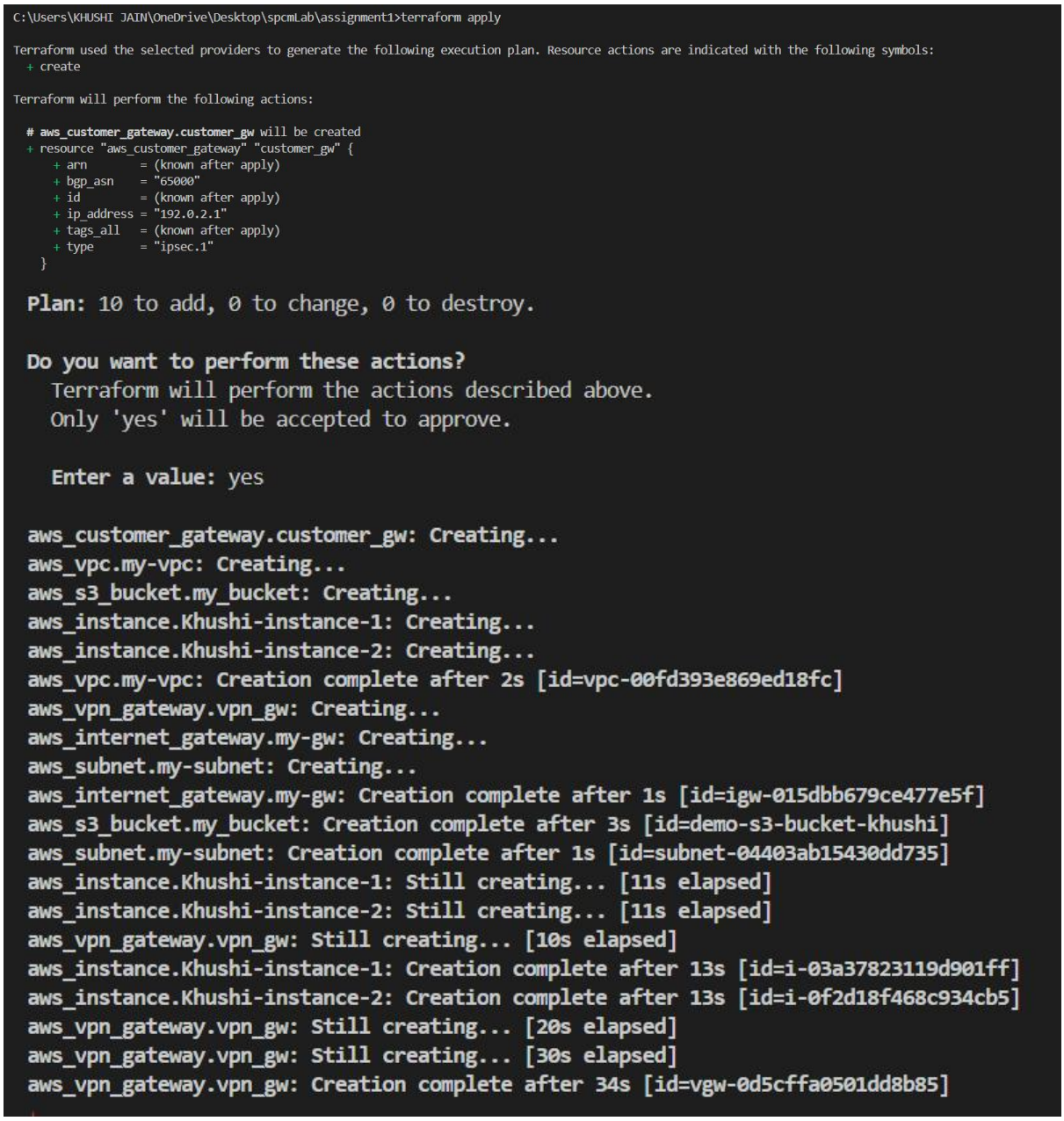
```
C:\Users\KHUSHI JAIN\OneDrive\Desktop\spcmLab\assignment1>terraform validate
Success! The configuration is valid.
```

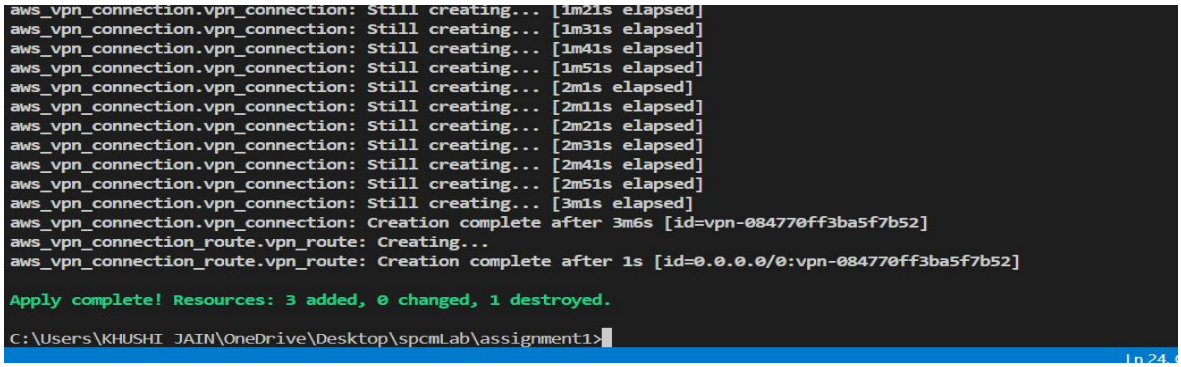
3. Terraform plan : to preview changes

```
C:\Users\KHUSHI JAIN\OneDrive\Desktop\spcmLab\assignment1>terraform plan

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_customer_gateway.customer_gw will be created
  + resource "aws_customer_gateway" "customer_gw" {
      + arn        = (known after apply)
      + bgp_asn    = "65000"
      + id         = (known after apply)
      + ip_address = "192.0.2.1"
      + tags_all   = (known after apply)
      + type       = "ipsec.1"
    }

  # aws_instance.Khushi-instance-1 will be created
  + resource "aws_instance" "Khushi-instance-1" {
      + ami                                  = "ami-03f4878755434977f"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
```

```
      + tunnel2_inside_ipv6_cidr       = (known after apply)
      + tunnel2_preshared_key          = (sensitive value)
      + tunnel2_vgw_inside_address     = (known after apply)
      + tunnel_inside_ip_version       = (known after apply)
      + type                           = "ipsec.1"
      + vgw_telemetry                  = (known after apply)
      + vpn_gateway_id                 = (known after apply)

      + tunnel1_log_options (known after apply)

      + tunnel2_log_options (known after apply)
    }

  # aws_vpn_connection_route.vpn_route will be created
  + resource "aws_vpn_connection_route" "vpn_route" {
      + destination_cidr_block = "0.0.0.0/0"
      + id                     = (known after apply)
      + vpn_connection_id      = (known after apply)
    }

  # aws_vpn_gateway.vpn_gw will be created
  + resource "aws_vpn_gateway" "vpn_gw" {
      + amazon_side_asn = (known after apply)
      + arn             = (known after apply)
      + id              = (known after apply)
      + tags_all        = (known after apply)
      + vpc_id          = (known after apply)
    }

Plan: 10 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
```

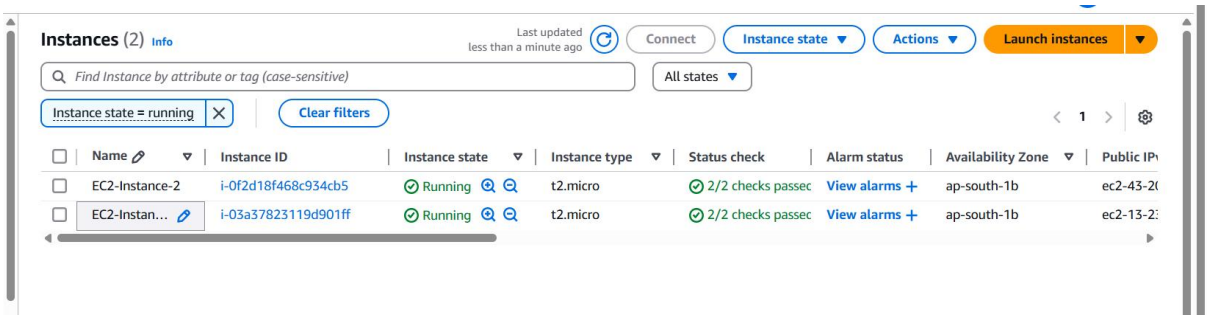
4. Terraform apply : Applies the configuration changes required, needs manual approval here



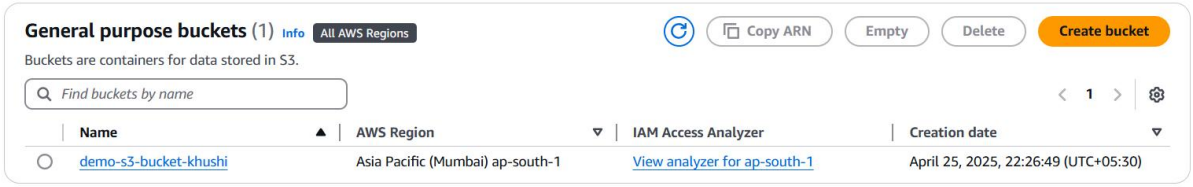


Verifying the Resources created on AWS:

1. Two Instances of type micro



2. S3 Bucket



3. VPC