



System Provisioning and Configuration Module Lab

Assignment 1

Under the Guidance of: **Dr. Hitesh Kumar Sharma**

Submitted by: Vibhav Khaneja

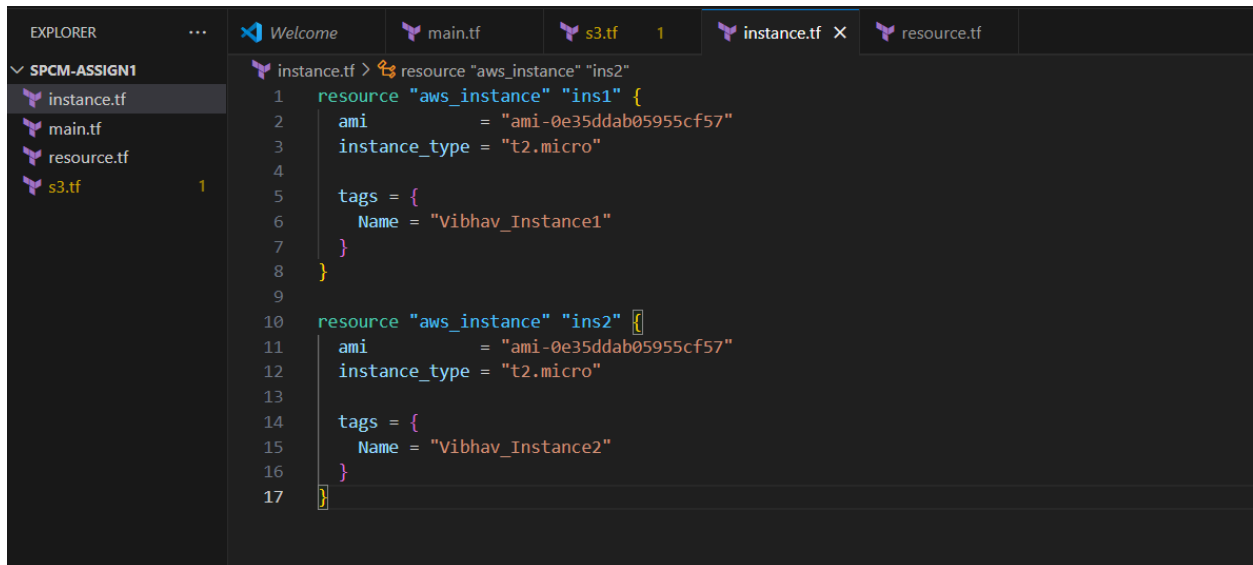
SAP: 500105662

Batch: DevOps-B1(N-H)

Roll No.: R2142220297

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

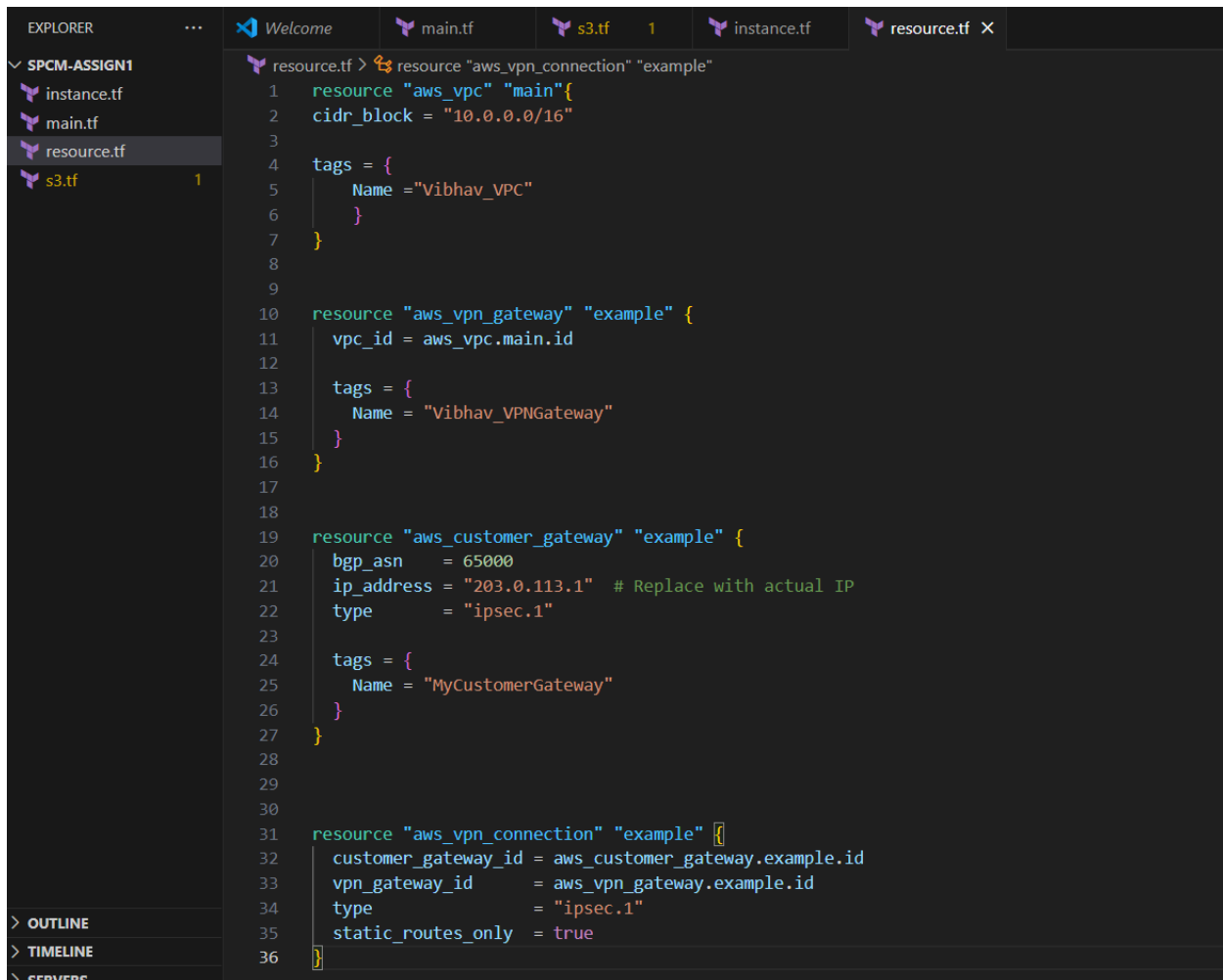
Step 1: Create two T2 Micro EC2 Instances. This instance.tf file contains the Iac code to create two instances of type t2.micro and ami of ubuntu.



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'SPCM-ASSIGN1' with files 'instance.tf', 'main.tf', 'resource.tf', and 's3.tf'. The 'instance.tf' file is selected. The code editor shows the following Terraform configuration:

```
instance.tf > resource "aws_instance" "ins2"
1 resource "aws_instance" "ins1" {
2     ami           = "ami-0e35ddab05955cf57"
3     instance_type = "t2.micro"
4
5     tags = {
6         Name = "Vibhav_Instance1"
7     }
8 }
9
10 resource "aws_instance" "ins2" {
11     ami           = "ami-0e35ddab05955cf57"
12     instance_type = "t2.micro"
13
14     tags = {
15         Name = "Vibhav_Instance2"
16     }
17 }
```

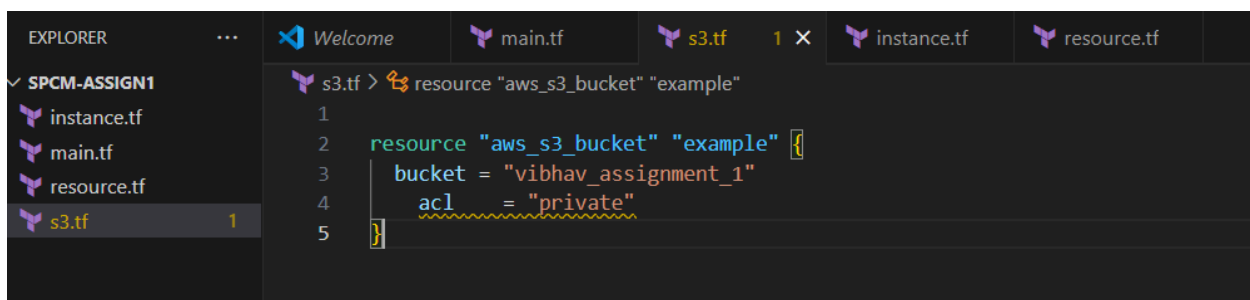
Step2: Create a VPN on AWS This resource.tf file contains the complete code to make a VPN. It consists of resources likevpn, customer gateway and vpn connection.



The screenshot shows the VS Code interface with the Explorer on the left and the Editor on the right. The Explorer shows a project named 'SPCM-ASSIGN1' with files 'instance.tf', 'main.tf', 'resource.tf', and 's3.tf'. The Editor has tabs for 'Welcome', 'main.tf', 's3.tf', 'instance.tf', and 'resource.tf'. The 'resource.tf' tab is active, displaying Terraform code for creating an AWS VPC, VPN Gateway, Customer Gateway, and VPN Connection. The code is as follows:

```
resource.tf > resource "aws_vpn_connection" "example"
1  resource "aws_vpc" "main"{
2    cidr_block = "10.0.0.0/16"
3
4    tags = {
5      Name = "Vibhav_VPC"
6    }
7  }
8
9
10 resource "aws_vpn_gateway" "example" {
11   vpc_id = aws_vpc.main.id
12
13   tags = {
14     Name = "Vibhav_VPNGateway"
15   }
16 }
17
18
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 }
```

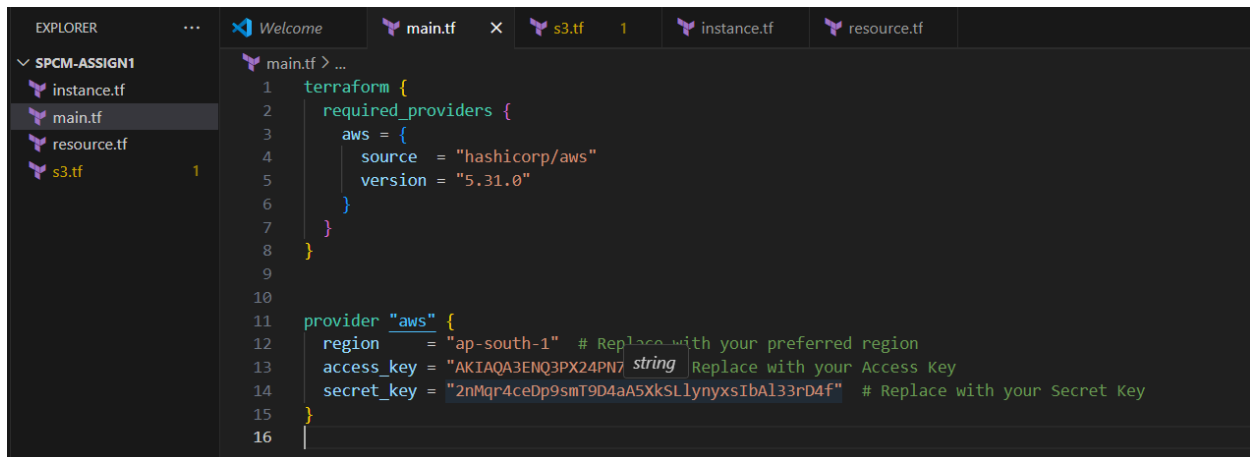
Step 3: Create a S3 Bucket Code to create a s3 bucket.



The screenshot shows the VS Code interface with the Explorer on the left and the Editor on the right. The Explorer shows the same project 'SPCM-ASSIGN1' with files 'instance.tf', 'main.tf', 'resource.tf', and 's3.tf'. The Editor has tabs for 'Welcome', 'main.tf', 's3.tf', 'instance.tf', and 'resource.tf'. The 's3.tf' tab is active, displaying Terraform code for creating an S3 bucket. The code is as follows:

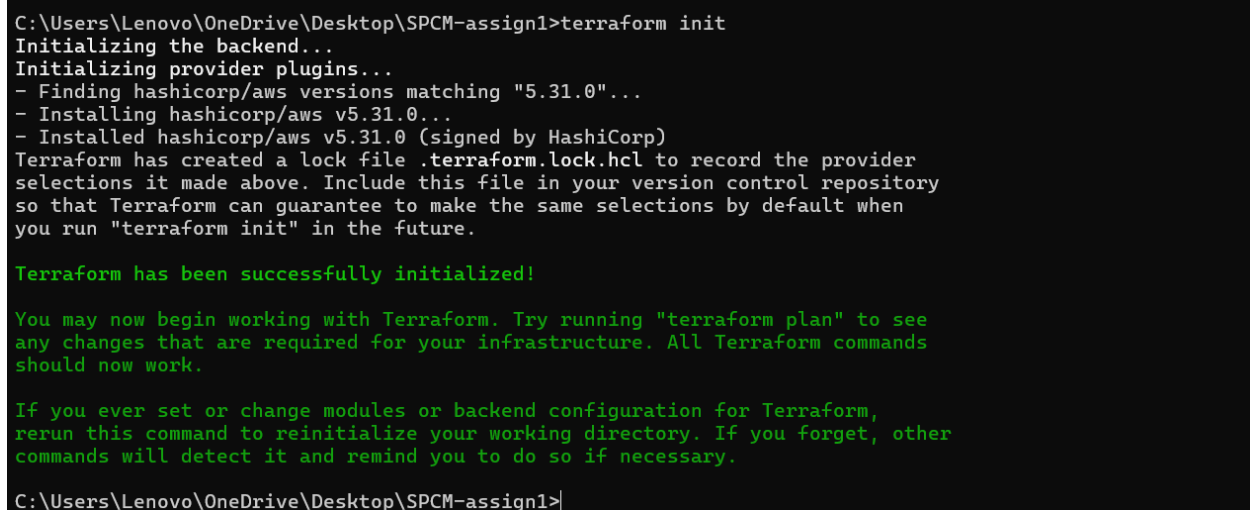
```
s3.tf > resource "aws_s3_bucket" "example"
1
2  resource "aws_s3_bucket" "example" {
3    bucket = "vibhav_assignment_1"
4    acl    = "private"
5  }
```

Main.tf file to perform the above-mentioned tasks

A screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project named 'SPCM-ASSIGN1' with files 'instance.tf', 'main.tf', 'resource.tf', and 's3.tf'. The 'main.tf' file is selected and open in the main editor. The code in 'main.tf' is a Terraform configuration for the AWS provider. It includes a 'terraform' block with 'required_providers' and an 'aws' provider block with configuration for region, access key, and secret key. The code is as follows:

```
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 = "AKIAQA3ENQ3PX24PN7" # Replace with your Access Key
14   secret_key = "2nMqr4ceDp9smT9D4aA5XkSLlynyxsIbAl33rD4f" # Replace with your Secret Key
15 }
16
```

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

A screenshot of a terminal window showing the output of the 'terraform init' command. The output indicates that the AWS provider has been successfully initialized. The text in the terminal is as follows:

```
C:\Users\Lenovo\OneDrive\Desktop\SPCM-assign1>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.

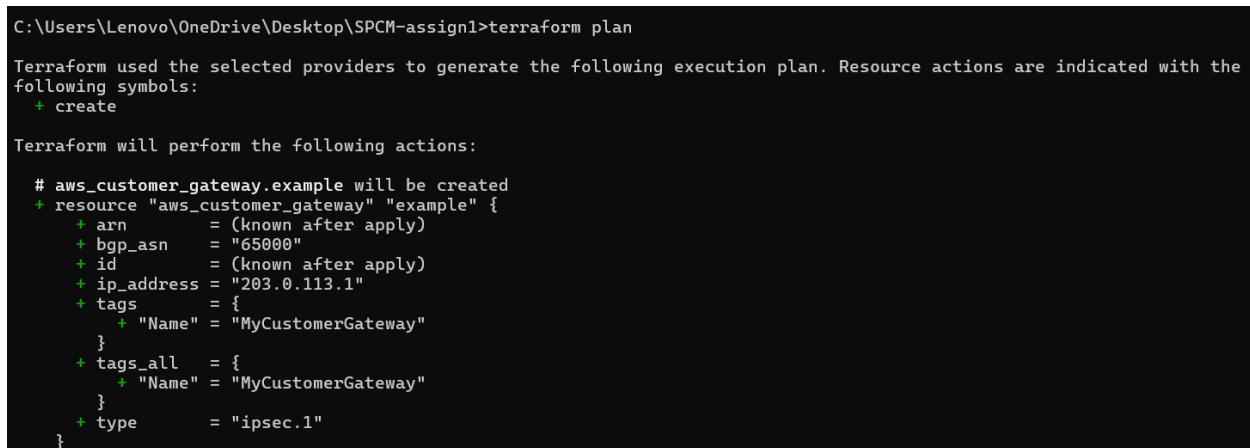
Terraform has been successfully initialized!

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\Lenovo\OneDrive\Desktop\SPCM-assign1>
```

Terraform plan to see the resources that will be created

A screenshot of a terminal window showing the output of the 'terraform plan' command. The output shows the execution plan for creating an AWS customer gateway. The text in the terminal is as follows:

```
C:\Users\Lenovo\OneDrive\Desktop\SPCM-assign1>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.example will be created
+ resource "aws_customer_gateway" "example" {
+   arn = (known after apply)
+   bgp_asn = "65000"
+   id = (known after apply)
+   ip_address = "203.0.113.1"
+   tags = {
+     "Name" = "MyCustomerGateway"
+   }
+   tags_all = {
+     "Name" = "MyCustomerGateway"
+   }
+   type = "ipsec.1"
}
```

```
# aws_instance.ins2 will be created
+ resource "aws_instance" "ins2" {
  + ami                      = "ami-0e35ddab05955cf57"
  + 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"
  + ipv6_address_count       = (known after apply)
  + ipv6_addresses           = (known after apply)
  + key_name                 = (known after apply)
  + monitoring               = (known after apply)
  + outpost_arn              = (known after apply)
  + password_data            = (known after apply)
  + placement_group          = (known after apply)
  + placement_partition_number = (known after apply)
  + primary_network_interface_id = (known after apply)
```

```
Command Prompt
+ tunnel2_cgw_inside_address = (known after apply)
+ tunnel2_inside_cidr        = (known after apply)
+ 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_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" = "Vibhav_VPNGateway"
+   }
+   tags_all         = {
+     + "Name" = "Vibhav_VPNGateway"
+   }
+   vpc_id           = (known after apply)
+ }

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

Terraform apply to create the mentioned resources

```

aws_vpn_connection.example: Still creating... [1m10s elapsed]
aws_vpn_connection.example: Still creating... [1m20s elapsed]
aws_vpn_connection.example: Still creating... [1m30s elapsed]
aws_vpn_connection.example: Still creating... [1m40s elapsed]
aws_vpn_connection.example: Still creating... [1m50s elapsed]
aws_vpn_connection.example: Still creating... [2m0s elapsed]
aws_vpn_connection.example: Still creating... [2m10s elapsed]
aws_vpn_connection.example: Still creating... [2m20s elapsed]
aws_vpn_connection.example: Still creating... [2m30s elapsed]
aws_vpn_connection.example: Still creating... [2m40s elapsed]
aws_vpn_connection.example: Still creating... [2m50s elapsed]
aws_vpn_connection.example: Still creating... [3m0s elapsed]
aws_vpn_connection.example: Still creating... [3m10s elapsed]
aws_vpn_connection.example: Still creating... [3m20s elapsed]
aws_vpn_connection.example: Still creating... [3m30s elapsed]
aws_vpn_connection.example: Still creating... [3m40s elapsed]
aws_vpn_connection.example: Creation complete after 3m46s [id=vpn-0f0c4e5e46947bb1a]

```

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

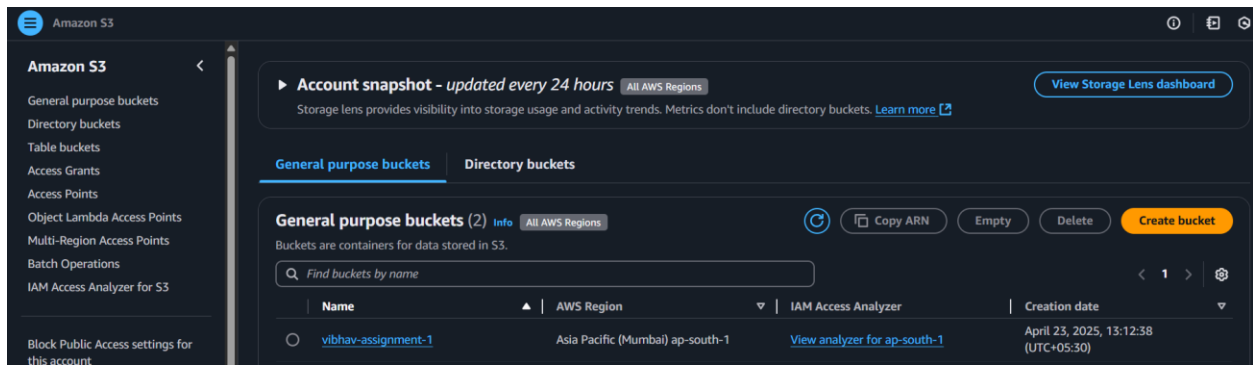
Apply complete! Resources: 7 added, 0 changed, 0 destroyed.

Resources created:

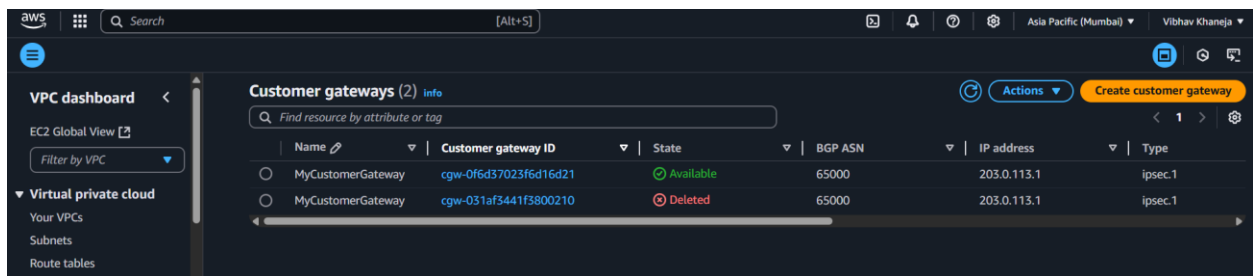
Instances

Instances (4) Info								
Last updated less than a minute ago Refresh Connect Instance state Actions Launch instances								
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/> All states < 1 > Settings								
<input type="checkbox"/>	Name ↗	Instance ID	Instance state ↕	Instance type ↕	Status check	Alarm status	Availability Zone ↕	Public IPv4
<input type="checkbox"/>	Vibhav_Instan...	i-0504c174d833ffa82	Terminated 🔍 🔍	t2.micro	-	View alarms +	ap-south-1a	-
<input type="checkbox"/>	Vibhav_Instan...	i-0bc3f0b3c0b95ca79	Running 🔍 🔍	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-3-127-
<input type="checkbox"/>	Vibhav_Instan...	i-029ade133f600c8a8	Running 🔍 🔍	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-3-109-1

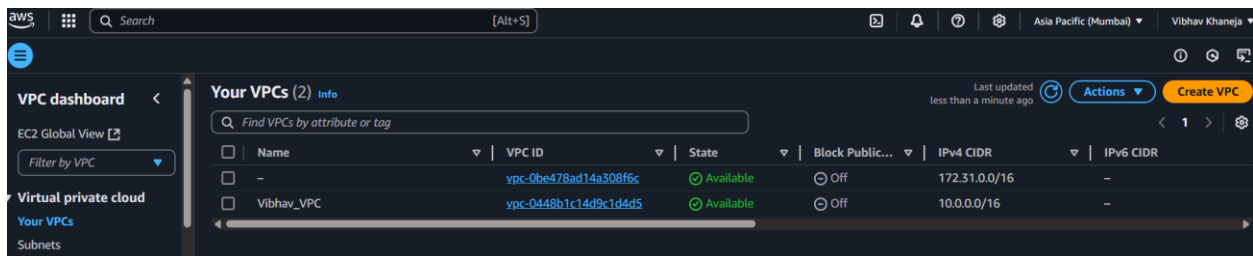
S3 bucket



Customer Gateway



Vpc



Then use Terraform Destroy to destroy all the resources


```
Command Prompt

aws_s3_bucket.example: Destroying... [id=vibhav-assignment-1]
aws_instance.ins2: Destroying... [id=i-029ade133f600c8a8]
aws_instance.ins1: Destroying... [id=i-0bc3f0b3c0b95ca79]
aws_vpn_connection.example: Destroying... [id=vpn-0f0c4e5e46947bb1a]
aws_s3_bucket.example: Destruction complete after 1s
aws_instance.ins1: Still destroying... [id=i-0bc3f0b3c0b95ca79, 10s elapsed]
aws_instance.ins2: Still destroying... [id=i-029ade133f600c8a8, 10s elapsed]
aws_vpn_connection.example: Still destroying... [id=vpn-0f0c4e5e46947bb1a, 10s elapsed]
aws_vpn_connection.example: Destruction complete after 11s
aws_customer_gateway.example: Destroying... [id=cgw-0f6d37023f6d16d21]
aws_vpn_gateway.example: Destroying... [id=vgw-04ffa9d2277dd4a53]
aws_customer_gateway.example: Destruction complete after 0s
aws_instance.ins1: Still destroying... [id=i-0bc3f0b3c0b95ca79, 20s elapsed]
aws_instance.ins2: Still destroying... [id=i-029ade133f600c8a8, 20s elapsed]
aws_vpn_gateway.example: Still destroying... [id=vgw-04ffa9d2277dd4a53, 10s elapsed]
aws_instance.ins2: Still destroying... [id=i-029ade133f600c8a8, 30s elapsed]
aws_instance.ins1: Still destroying... [id=i-0bc3f0b3c0b95ca79, 30s elapsed]
aws_vpn_gateway.example: Still destroying... [id=vgw-04ffa9d2277dd4a53, 20s elapsed]
aws_vpn_gateway.example: Destruction complete after 24s
aws_vpc.main: Destroying... [id=vpc-0448b1c14d9c1d4d5]
aws_vpc.main: Destruction complete after 1s
aws_instance.ins2: Still destroying... [id=i-029ade133f600c8a8, 40s elapsed]
aws_instance.ins1: Still destroying... [id=i-0bc3f0b3c0b95ca79, 40s elapsed]
aws_instance.ins1: Destruction complete after 41s
aws_instance.ins2: Destruction complete after 41s

Destroy complete! Resources: 7 destroyed.

C:\Users\Lenovo\OneDrive\Desktop\SPCM-assign1>
```