
**SCHOOL OF COMPUTER SCIENCE**

**SPCM ASSIGNMENT-1**

**Submitted by**



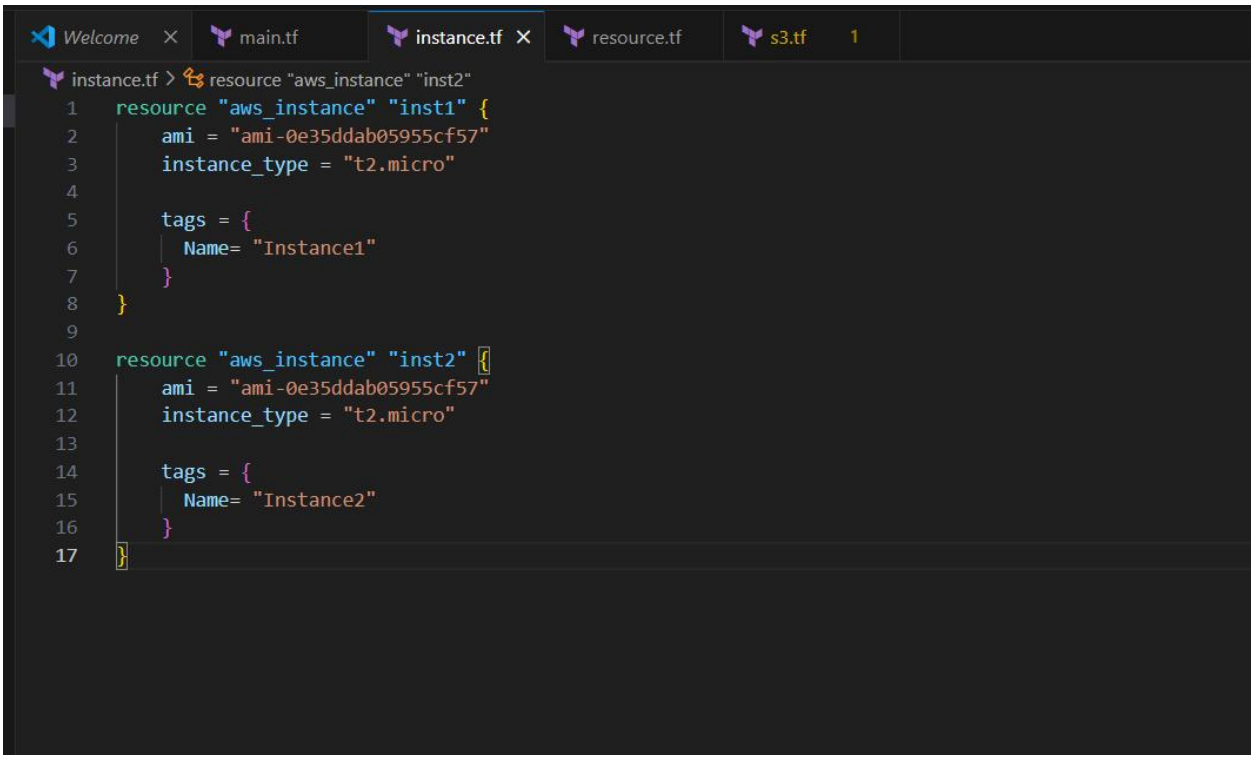
| | |
|---|---|
| **Name** | **Kashish** |
| **Branch** | **BTech CSE(DevOps)B-1(NH)** |
| **Semester** | **6** |
| **SAPID** | **500107137** |
| **Roll no** | **R2142220335** |

**ASSIGNMENT 1**

**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.




Step2: Create a VPN on AWS

This resource.tf file contains the complete code to make a VPN. It consists of resources like vpc, customer gateway and vpn connection.

```
Welcome | main.tf | instance.tf | resource.tf | s3.tf 1
resource.tf > resource "aws_vpn_connection" "sample" > static_routes_only
1  resource "aws_vpc" "main" {
2      cidr_block = "10.0.0.0/16"
3
4      tags = {
5        Name="Kashish_VPC"
6      }
7
8  }
9  resource "aws_vpn_gateway" "sample" {
10     vpc_id = aws_vpc.main.id
11
12     tags = {
13       Name = "Kashish-vpn-gateway"
14     }
15
16 }
17 resource "aws_customer_gateway" "sample" {
18
19     bgp_asn = 6500
20     ip_address = "192.0.2.1"
21     type = "ipsec.1"
22
23     tags = {
24       Name="Mycustomergateway"
25     }
26 }
27
28 resource "aws_vpn_connection" "sample" {
29   customer_gateway_id = aws_customer_gateway.sample.id
30   vpn_gateway_id = aws_vpn_gateway.sample.id
31   type = "ipsec.1"
32   static_routes_only = true
33 }
```

Step 3: Create a S3 Bucket

Code to create a s3 bucket.

```
Welcome | main.tf | instance.tf | resource.tf | s3.tf 1
s3.tf > resource "aws_s3_bucket" "sample"
1  resource "aws_s3_bucket" "sample"{
2      bucket = "kashish-assign-1"
3      acl = "private"
4  }
```

Main.tf file to perform the above-mentioned tasks.

main.tf > terraform > required_providers

```
terraform {
  required_providers {
    aws = {
        source = "hashicorp/aws"
        version = "5.30.1"
    }
  }
}
provider "aws" {
    region = "ap-south-1"
    access_key = "AKIAWAA66PDJ6OKADSAD"
    secret_key = "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"
}
```

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

```
C:\Users\Lenovo\OneDrive\Desktop>cd assign1-spcm

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

Terraform plan to see the resources that will be created.

```
C:\Users\Lenovo\OneDrive\Desktop\assign1-spcm>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.sample will be created
  + resource "aws_customer_gateway" "sample" {
      + arn        = (known after apply)
      + bgp_asn    = "6500"
      + id         = (known after apply)
      + ip_address = "192.0.2.1"
      + tags       = {
          + "Name" = "Mycustomergateway"
        }
      + tags_all   = {
          + "Name" = "Mycustomergateway"
        }
      + type       = "ipsec.1"
    }

  # aws_instance.inst1 will be created
  + resource "aws_instance" "inst1" {
      + 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)
```

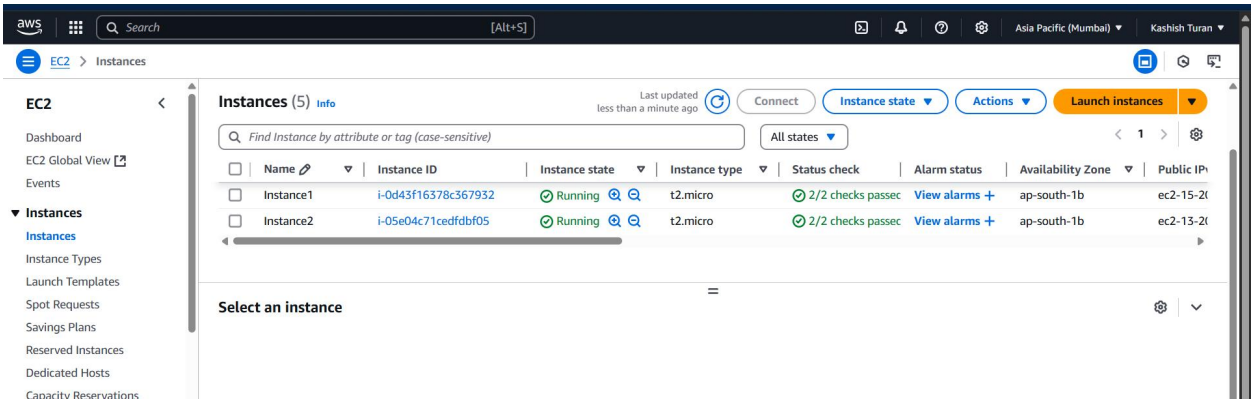
```
      + tunnel1_bgp_asn               = (known after apply)
      + tunnel1_bgp_holdtime          = (known after apply)
      + tunnel1_cgw_inside_address    = (known after apply)
      + tunnel1_inside_cidr           = (known after apply)
      + tunnel1_inside_ipv6_cidr      = (known after apply)
      + tunnel1_preshared_key         = (sensitive value)
      + tunnel1_vgw_inside_address    = (known after apply)
      + tunnel2_address               = (known after apply)
      + tunnel2_bgp_asn               = (known after apply)
      + tunnel2_bgp_holdtime          = (known after apply)
      + 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.sample will be created
  + resource "aws_vpn_gateway" "sample" {
      + amazon_side_asn = (known after apply)
      + arn             = (known after apply)
      + id              = (known after apply)
      + tags            = {
          + "Name" = "Kashish-vpn-gateway"
        }
      + tags_all        = {
          + "Name" = "Kashish-vpn-gateway"
        }
      + vpc_id          = (known after apply)
    }

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

Terraform apply to create the mentioned resources.

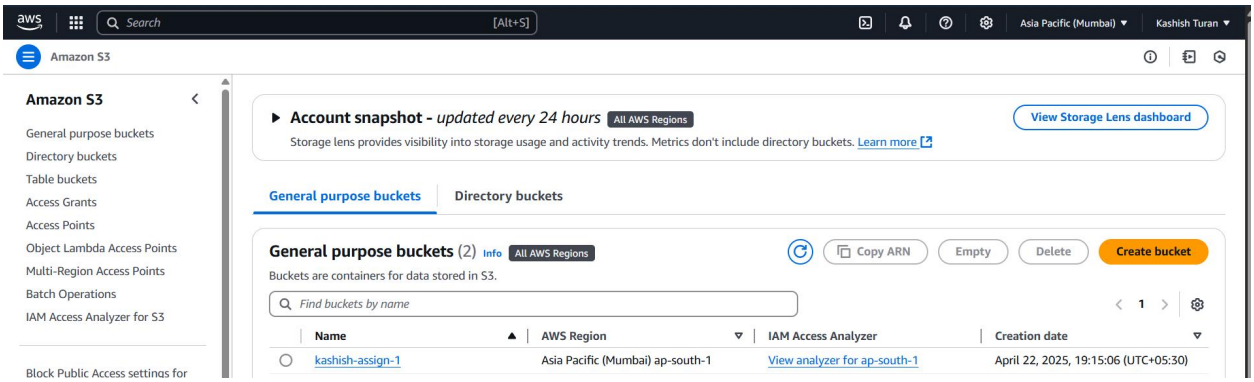
```
C:\Users\Lenovo\OneDrive\Desktop\assign1-spcm>terraform apply
aws_vpc.main: Refreshing state... [id=vpc-067dec781b49527b6]
aws_s3_bucket.sample: Refreshing state... [id=kashish-assign-1]
aws_instance.inst1: Refreshing state... [id=i-0d43f16378c367932]
aws_instance.inst2: Refreshing state... [id=i-05e04c71cedfdbf05]
aws_vpn_gateway.sample: Refreshing state... [id=vgw-0d7976a5454edee15]

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.sample will be created
  + resource "aws_customer_gateway" "sample" {
      + arn        = (known after apply)
      + bgp_asn    = "6500"
      + id         = (known after apply)
      + ip_address = "203.0.113.1"
      + tags       = {
          + "Name" = "Mycustomergateway"
        }
      + tags_all   = {
          + "Name" = "Mycustomergateway"
        }
      + type       = "ipsec.1"
    }
```

Resources created:

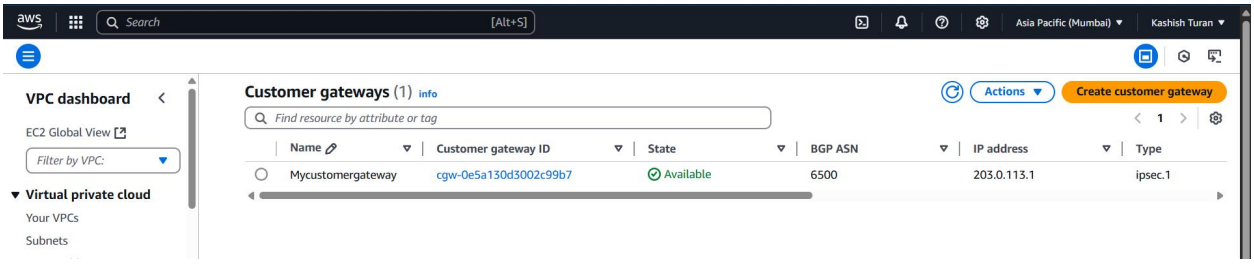
Instances



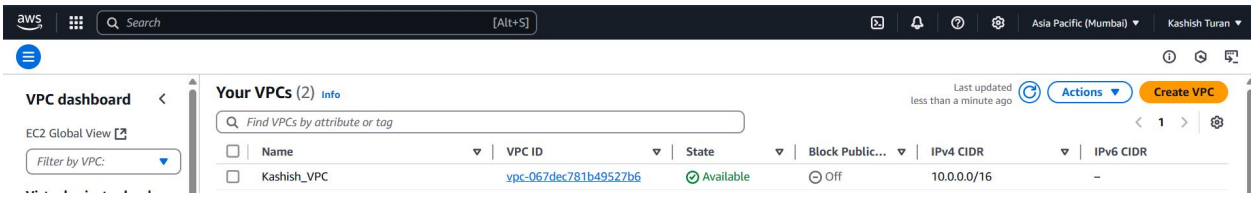

S3 bucket




Customer Gateway




Vpc


aws
Search
[Alt+S]
Asia Pacific (Mumbai)
Kashish Turan
VPC dashboard
EC2 Global View
Filter by VPC:
Your VPCs (2) Info
Last updated
less than a minute ago
Actions
Create VPC
Find VPCs by attribute or tag
1
Name
VPC ID
State
Block Public...
IPv4 CIDR
IPv6 CIDR
Kashish_VPC
vpc-067dec781b49527b6
Available
Off
10.0.0.0/16
–