

SPCM - LAB

6th Sem

Submitted To:

Dr. Hitesh Kumar Sharma

School of Computer Science

Submitted By:

Sujal Bhandari

R2142220181

500106865

Batch 1-NH

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

```
main.tf  X
main.tf > provider "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"
13     access_key = "AKIA4ZZIDPT" string "KB"
14     secret_key = "QLYtXlk4Jk+Ndqf/Jj5E1SzuAOee0NI1qAGtkNxs"
15 }
```

Running terraform init

```
PS C:\Users\sujal\OneDrive\Desktop\SPCM_Assignment> 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.

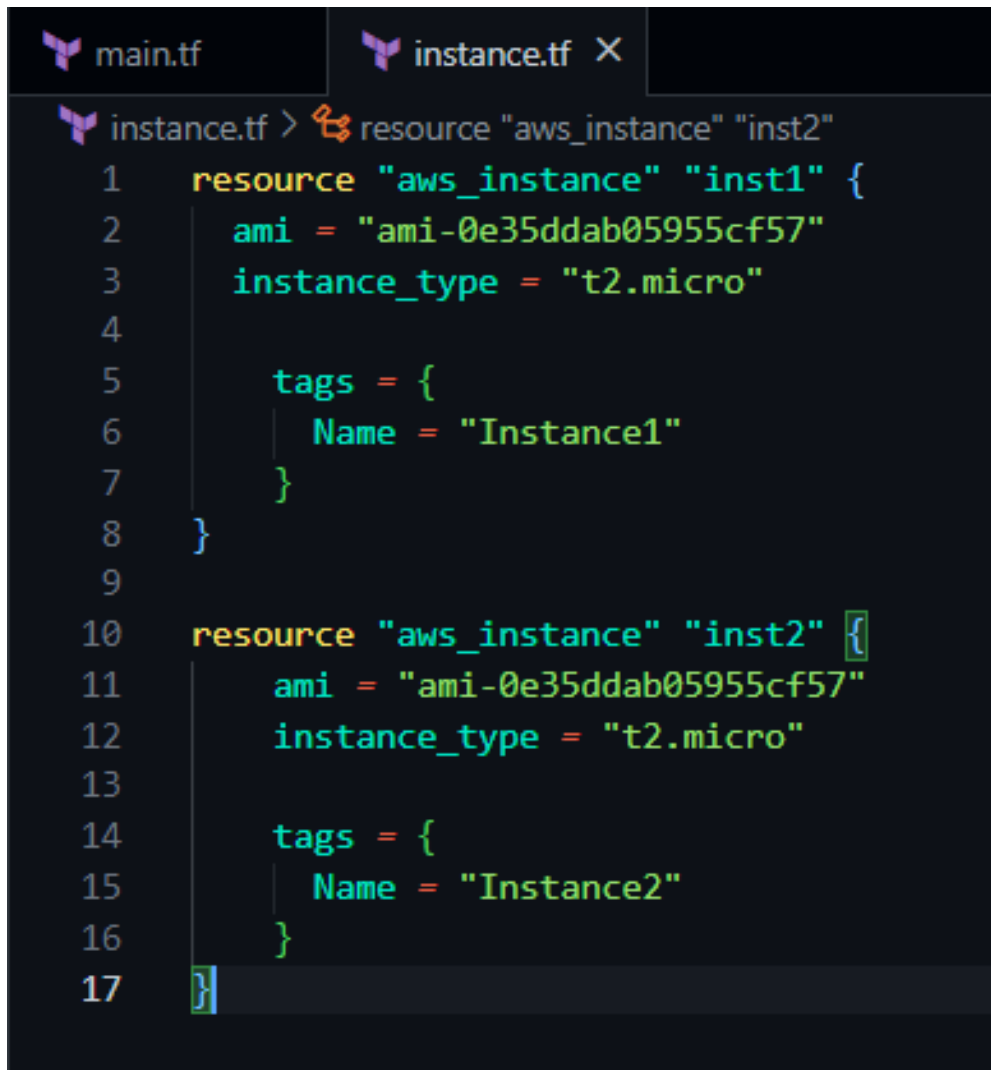
Terraform has been successfully initialized!

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.
PS C:\Users\sujal\OneDrive\Desktop\SPCM_Assignment>
```

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

instance.tf



```
main.tf instance.tf X
instance.tf > resource "aws_instance" "inst2"
1  resource "aws_instance" "inst1" {
2      ami = "ami-0e35ddab05955cf57"
3      instance_type = "t2.micro"
4
5      tags = {
6          Name = "Instance1"
7      }
8  }
9
10 resource "aws_instance" "inst2" {
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

```
resource.tf > ...
1  resource "aws_vpc" "main" {
2      cidr_block = "10.0.0.0/16"
3
4      tags = {
5          Name = "Suja1VPC"
6      }
7  }
8
9  resource "aws_vpn_gateway" "example" {
10     vpc_id = aws_vpc.main.id
11
12     tags = {
13         Name = "MyVPNGateway"
14     }
15 }
16
```

```
resource "aws_customer_gateway" "example" {
    bgp_asn = 65000
    ip_address = "203.0.113.1"
    type = "ipsec.1"

    tags = {
        Name = "MyCustomerGateway"
    }
}

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

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

s3.tf

```
s3.tf > ...
1  resource "aws_s3_bucket" "example" {
2      bucket = "sujal-s3"
3      acl = "private"
4  }
5
6
```

The s3.tf hold the code to create a s3 bucket which has a unique name

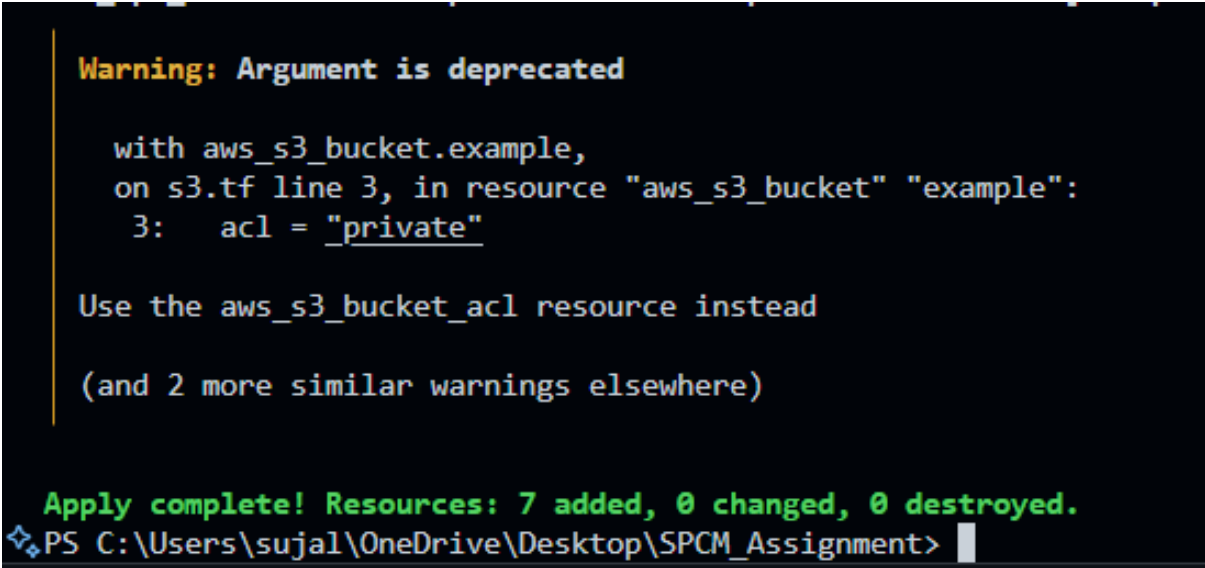
Outputs:

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



Customer Gateway

Customer gateways (1) info							Actions Create customer gateway	
<input type="text" value="Find resource by attribute or tag"/>							< 1 > ⚙	
<input type="checkbox"/>	Name	Customer gateway ID	State	BGP ASN	IP address	Type		
<input type="radio"/>	MyCustomerGateway	cgw-0f6619c592d68af7e	Available	65000	203.0.113.1	ipsec.1		

Vpc

Your VPCs (2) info							Last updated less than a minute ago Actions Create VPC	
<input type="text" value="Find VPCs by attribute or tag"/>							< 1 > ⚙	
<input type="checkbox"/>	Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR		
<input type="checkbox"/>	SujalVPC	vpc-09aa0f67370c7913c	Available	Off	10.0.0.0/16	-		
<input type="checkbox"/>	-	vpc-0aa7c2c88a69a7678	Available	Off	172.31.0.0/16	-		

S3

General purpose buckets				Directory buckets			
General purpose buckets (1) info				All AWS Regions Copy ARN Empty Delete Create bucket			
Buckets are containers for data stored in S3.							
<input type="text" value="Find buckets by name"/>				< 1 > ⚙			
<input type="radio"/>	Name	AWS Region	IAM Access Analyzer	Creation date			
<input type="radio"/>	sujal-s3	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	April 23, 2025, 17:19:34 (UTC+05:30)			

Instances

Instances (2) info							Last updated less than a minute ago Connect Instance state Actions Launch instances	
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>							All states < 1 > ⚙	
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>	Instance2	i-0d2ed74adf8d5679b	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a	ec2-13-201
<input type="checkbox"/>	Instance1	i-0c8d0b20b4d21cad3	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a	ec2-13-126