



**SPCM - LAB**  
**6th Sem**

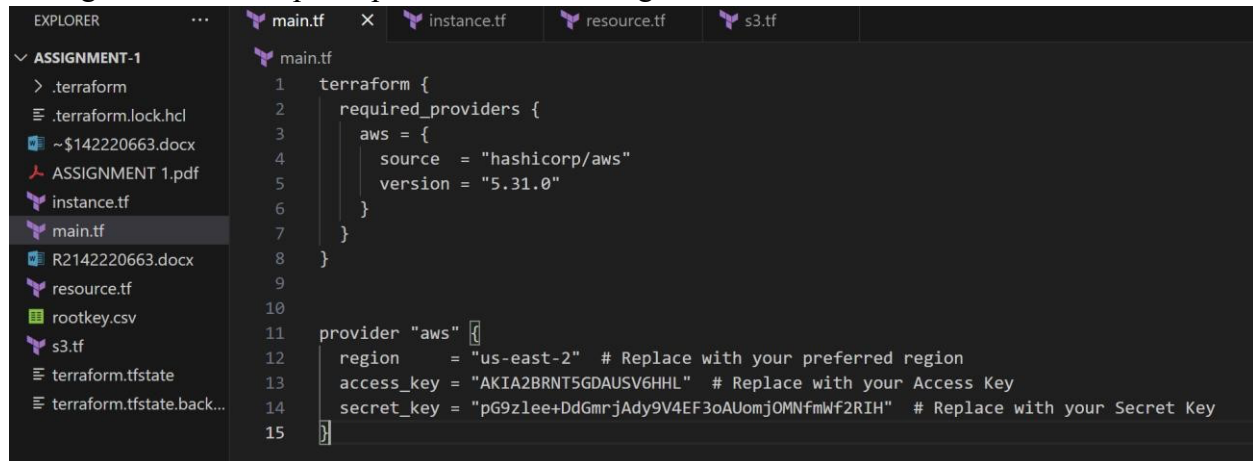
**Submitted To:**

**Dr. Hitesh Kumar Sharma**  
**School of Computer Science**

**Submitted By:**

**Raghav Kamboj**  
**R2142220137**  
**500102126**  
**Batch 1-NH**

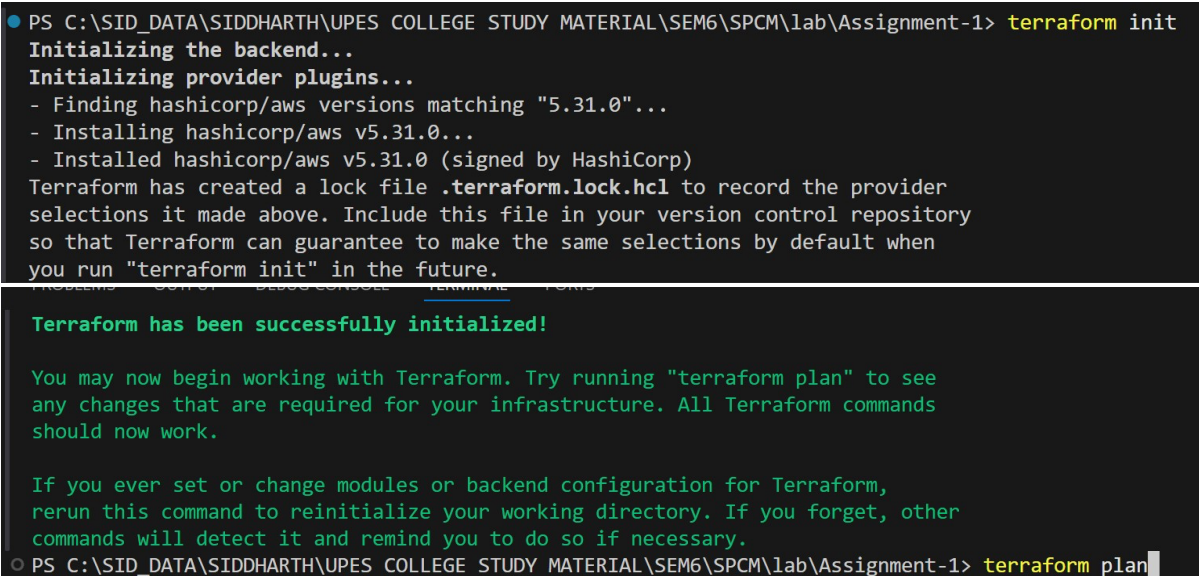
Writing Terraform Scripts to perform the following task 2 ec2 Instances, VPN and S3 **main.tf**



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 'ASSIGNMENT-1' with files like '.terraform', '.terraform.lock.hcl', and 'main.tf'. The code editor shows the content of 'main.tf' with the following code:

```
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     = "us-east-2" # Replace with your preferred region
13   access_key = "AKIA2BRNT5GDAUSV6HHL" # Replace with your Access Key
14   secret_key = "pG9zlee+DdGmrjAdy9V4EF3oAUomjOMNfmWf2RIH" # Replace with your Secret Key
15 }
```

### Running terraform init



The screenshot shows a terminal window with the following output:

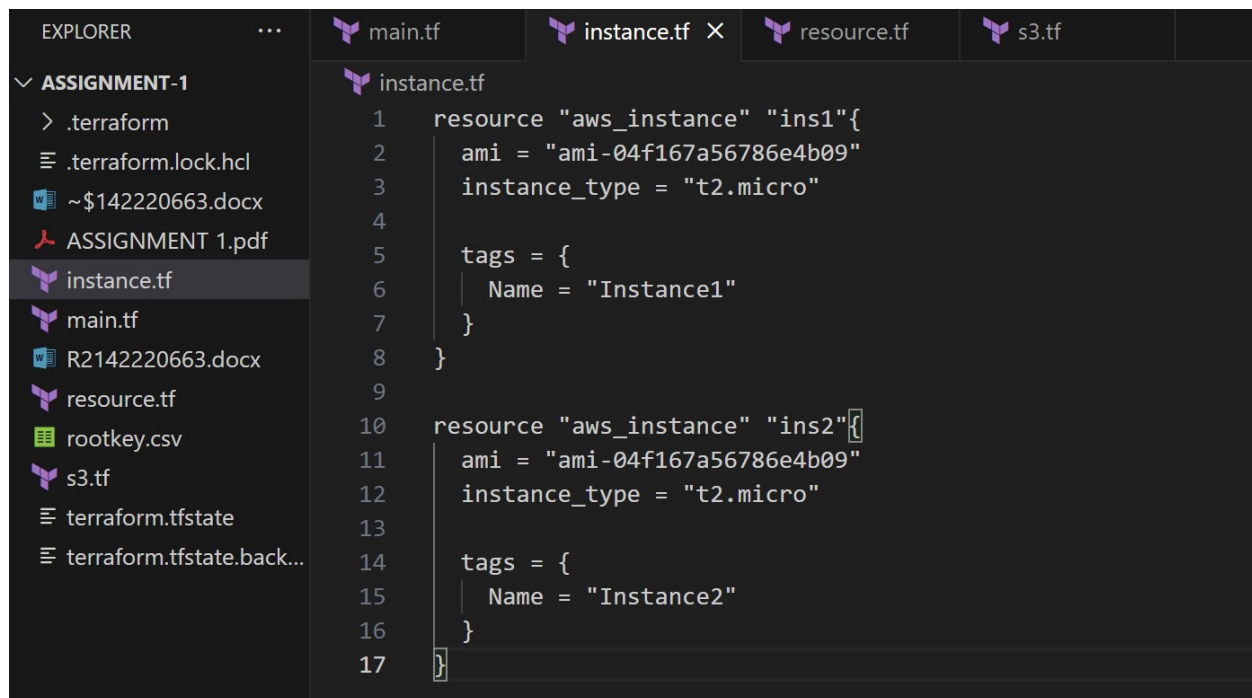
```
PS C:\SID_DATA\SIDDHARTH\UPES COLLEGE STUDY MATERIAL\SEM6\SPCM\lab\Assignment-1> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.31.0"...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (signed by HashiCorp)
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:\SID_DATA\SIDDHARTH\UPES COLLEGE STUDY MATERIAL\SEM6\SPCM\lab\Assignment-1> terraform plan
```

Terraform init to initialize the terraform folder which will have the aws provider plugin installed **instance.tf**



```
1 resource "aws_instance" "ins1"{
2     ami = "ami-04f167a56786e4b09"
3     instance_type = "t2.micro"
4
5     tags = {
6         Name = "Instance1"
7     }
8 }
9
10 resource "aws_instance" "ins2"{
11     ami = "ami-04f167a56786e4b09"
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 "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"

  tags = {
    Name = "SiddoooVPC"
  }
}

resource "aws_vpn_gateway" "example" {
  vpc_id = aws_vpc.main.id

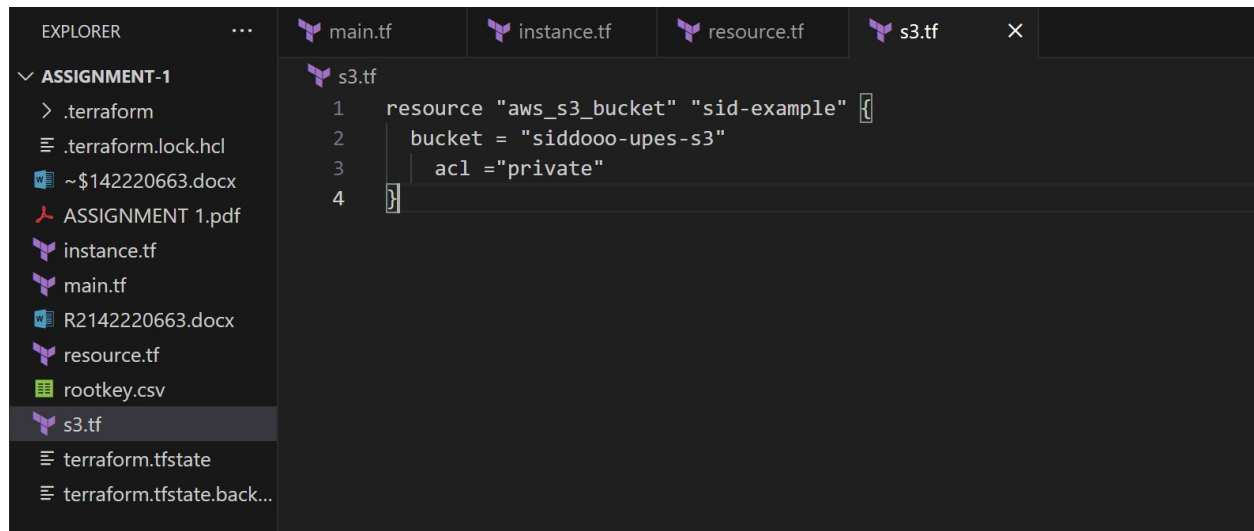
  tags = {
    Name = "MyVPNGateway"
  }
}

resource "aws_customer_gateway" "example" {
  bgp_asn = 65000
  ip_address = "203.0.113.1" # Replace with actual IP
  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"
}
```

This resource.tf hold the iac code to create vpc , the customer gateway and the vpn connection.  
s3.tf



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

## Outputs: terraform plan

```
PS C:\SID_DATA\SIDDHARTH\UPES COLLEGE STUDY MATERIAL\SEM6\SPCM\lab\Assignment-1> 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"

+ 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" = "MyVPNGateway"
+   }
+   tags_all         = {
+     + "Name" = "MyVPNGateway"
+   }
+   vpc_id           = (known after apply)
}

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

## Terraform apply

```
PS C:\SID_DATA\SIDDHARTH\UPES COLLEGE STUDY MATERIAL\SEM6\SPCM\lab\Assignment-1> terraform apply -auto-approve

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.ins1 will be created
+ resource "aws_instance" "ins1" {
  + ami                    = "ami-04f167a56786e4b09"
  + 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
}

3:      acl = "private"

Use the aws_s3_bucket_acl resource instead

(and 2 more similar warnings elsewhere)

(and 2 more similar warnings elsewhere)

(and 2 more similar warnings elsewhere)

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.
PS C:\SID_DATA\SIDDHARTH\UPES COLLEGE STUDY MATERIAL\SEM6\SPCM\lab\Assignment-1> 
```

## Customer Gateway



aws

Search

[Alt+S]

United States (Ohio)

Siddharth Agarwal

VPC dashboard

EC2 Global View

Filter by VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Customer gateways (3)

Find resource by attribute or tag

Actions

Create customer gateway

	Name	Customer gateway ID	State	BGP ASN	IP address	Type
<input type="radio"/>	MyCustomerGateway	cgw-053fbc190f289fd02	Available	65000	203.0.113.1	ipsec
<input type="radio"/>	MyCustomerGateway	cgw-0c3a83d22dec3ba7c	Deleted	65000	203.0.113.1	ipsec
<input type="radio"/>	MyCustomerGateway	cgw-02fd7bbe854059df9	Deleted	65000	203.0.113.1	ipsec

VPC

aws

Search

[Alt+S]

United States (Ohio)

Siddharth Agarwal

VPC dashboard

EC2 Global View

Filter by VPC

Virtual private cloud

Your VPCs

Subnets

Your VPCs (2)

Search

Last updated less than a minute ago

Actions

Create VPC

	Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	-	vpc-04085bbd7204f5959	Available	Off	172.31.0.0/16	-
<input type="checkbox"/>	SiddoooVPC	vpc-0e26294ee298f65a7	Available	Off	10.0.0.0/16	-

S3

aws

Search

[Alt+S]

United States (Ohio)

Siddharth Agarwal

Amazon S3

General purpose buckets

Directory buckets

Access Grants

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Account snapshot - updated every 24 hours

View Storage Lens dashboard

General purpose buckets

Directory buckets

General purpose buckets (1)

Copy ARN

Empty

Delete

Create bucket

Buckets are containers for data stored in S3.

Find buckets by name

	Name	AWS Region	IAM Access Analyzer	Creation date
<input type="radio"/>	siddooo-upes-s3	US East (Ohio) us-east-2	View analyzer for us-east-2	April 11, 2025, 12:15:48 (UTC+05:30)

Instances

aws

Search

[Alt+S]

United States (Ohio)

Siddharth Agarwal

EC2 > Instances

Dashboard

EC2 Global View

Events

Instances

Instance Types

Launch Templates

Spot Requests

Instances (4)

Find Instance by attribute or tag (case-sensitive)

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	Instance2	i-0f218988196a09e08	Terminated	t2.micro	-	View alarms +	us-east-2a
<input type="checkbox"/>	Instance1	i-0b34743202d330a45	Terminated	t2.micro	-	View alarms +	us-east-2a
<input type="checkbox"/>	Instance2	i-037f3dd6a1dfdada5a	Running	t2.micro	2/2 checks passec	View alarms +	us-east-2a
<input type="checkbox"/>	Instance1	i-03050527f4d98b664	Running	t2.micro	2/2 checks passec	View alarms +	us-east-2a