

# **ASSIGNMENT - 1**

## **SPCM**

**Pratyush Pandey**

**R171217047**

**500061663**

**B.TECH-CSE-DEVOPS**

## Terraform scripts to perform following tasks on AWS cloud Platform

1. Creating two T2 micro ec2 instances
2. Creating a VPN on AWS
3. Creating a S3 bucket

### Steps:

1. Configure your aws credentials using aws configure

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\hp\Desktop\Sem VII\spcm> aws configure
AWS Access Key ID [*****7LOQ]: AKIAJSKUG4DCLUHJJYQO
AWS Secret Access Key [*****9QF1]: xQqM4lAw9XG85pbx18leyCCuL0oSb2ii9bamaeOB
Default region name [ap-south-1]:
Default output format [None]:
```

## Get your security credentials (secret and access key) from aws console

**Your Security Credentials**

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the [IAM Console](#).

To learn more about the types of AWS credentials and how they're used, see [AWS Security Credentials](#) in [AWS General Reference](#).

▲ Password

▲ Multi-factor authentication (MFA)

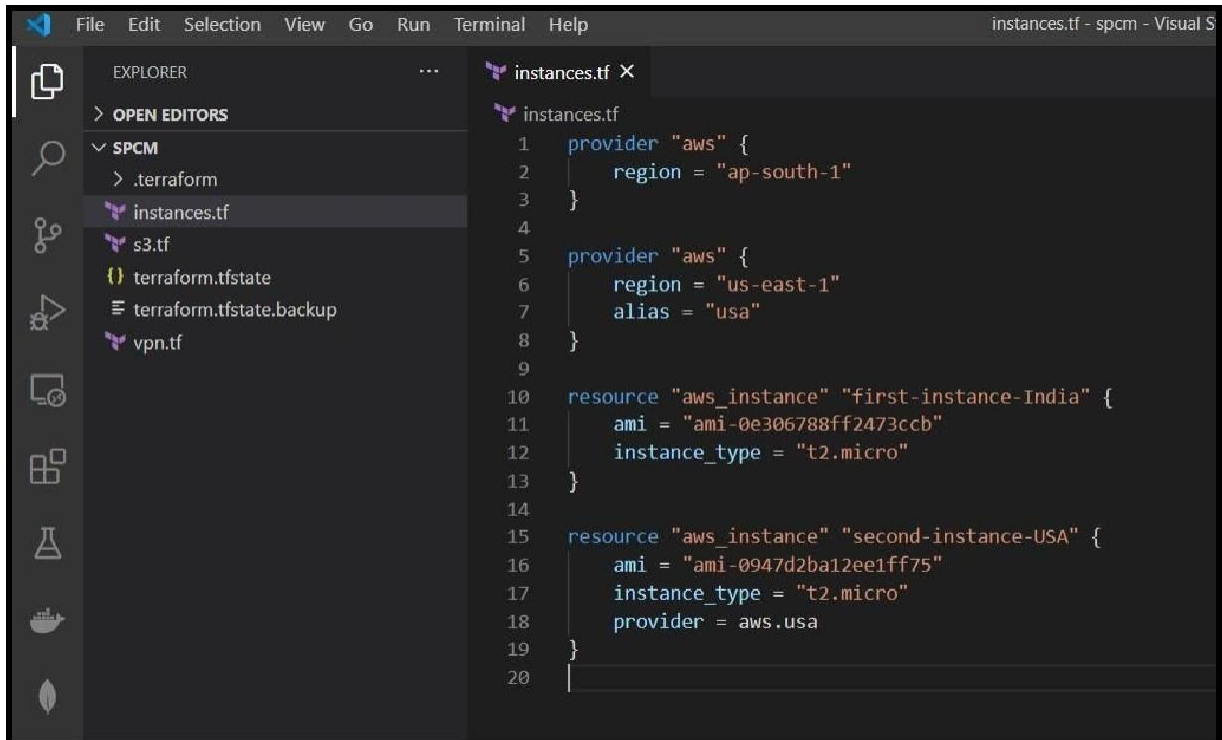
▼ Access keys (access key ID and secret access key)

Use access keys to make programmatic calls to AWS from the AWS CLI, Tools for PowerShell, the AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

Created	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions
Nov 18th 2020	AKIA-YHLVVQZ1MLYU7FA	N/A	N/A	N/A	Active	<a href="#">Make inactive</a>   <a href="#">Delete</a>
Nov 18th 2020	AKIAJULJE53DJHO7ZHTQ	2020-11-18 15:37 UTC+0530	us-east-1	ec2	Active	<a href="#">Make inactive</a>   <a href="#">Delete</a>

[Create New Access Key](#)

## 2. Creating terraform scripts to fulfill above mentioned tasks

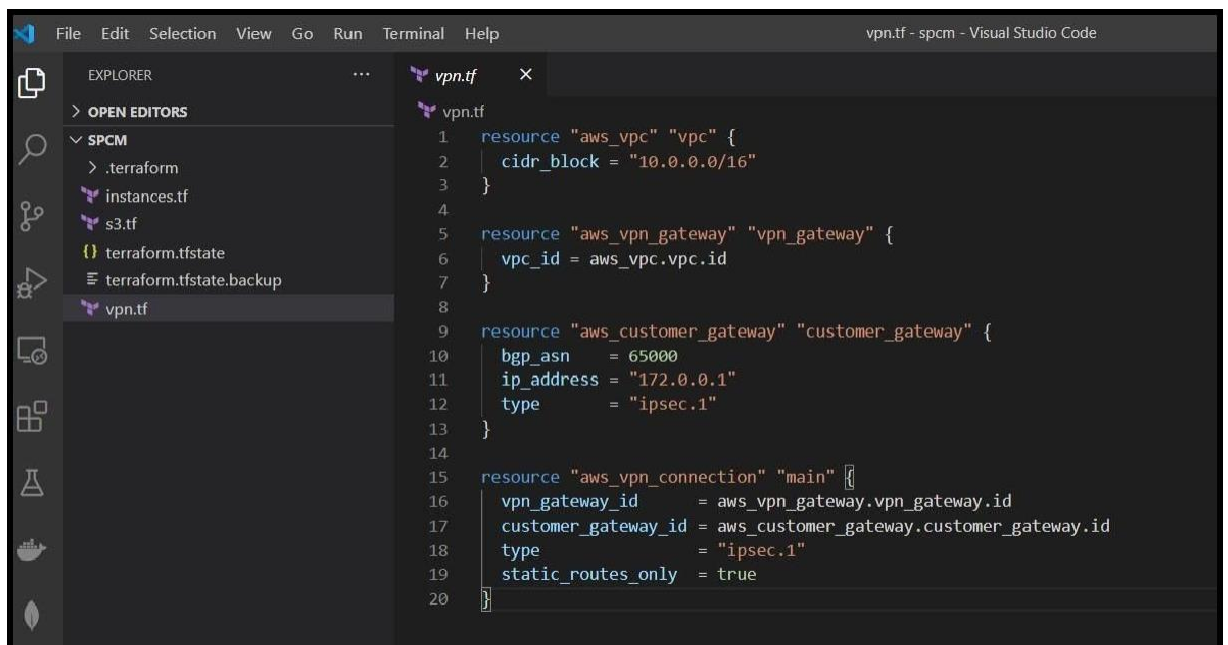


The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left and the Editor window on the right. The Explorer sidebar shows the project structure for 'SPCM' with files: instances.tf, s3.tf, terraform.tfstate, terraform.tfstate.backup, and vpn.tf. The Editor window displays the content of instances.tf, which defines two AWS providers and two AWS instances.

```
1 provider "aws" {
2   region = "ap-south-1"
3 }
4
5 provider "aws" {
6   region = "us-east-1"
7   alias = "usa"
8 }
9
10 resource "aws_instance" "first-instance-India" {
11   ami = "ami-0e306788ff2473ccb"
12   instance_type = "t2.micro"
13 }
14
15 resource "aws_instance" "second-instance-USA" {
16   ami = "ami-0947d2ba12ee1ff75"
17   instance_type = "t2.micro"
18   provider = aws.usa
19 }
20
```

We create 2 aws providers, one in ap-south-1(Mumbai region) which is the default region and the other in us-east-1(Virginia region - USA)

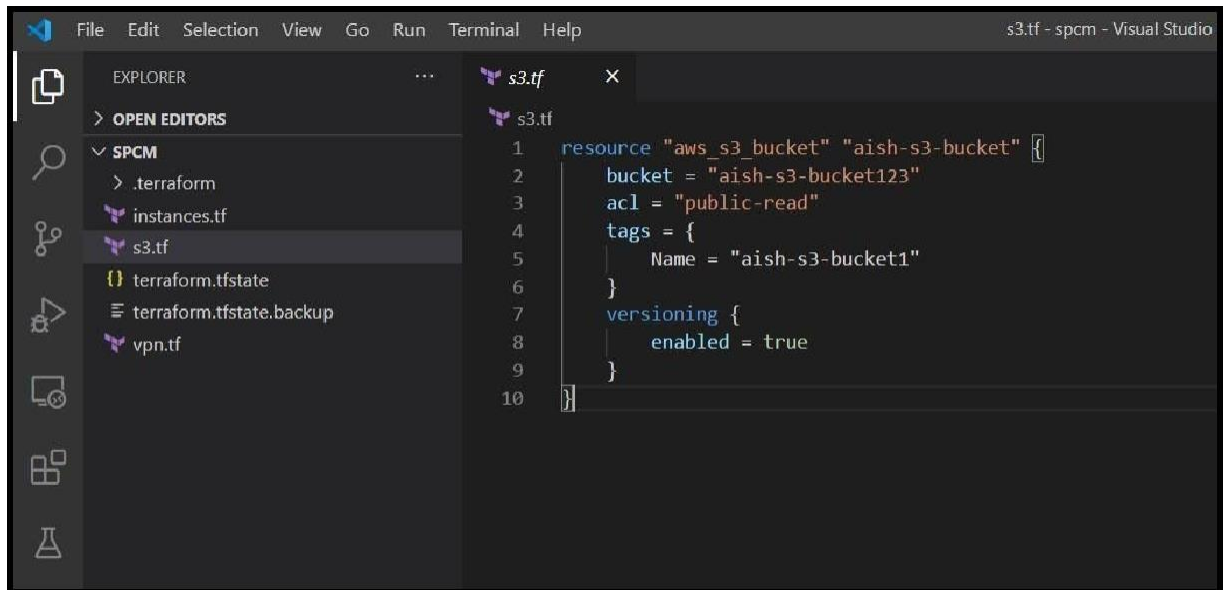
- Creating VPN



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left and the Editor window on the right. The Explorer sidebar shows the project structure for 'SPCM' with files: instances.tf, s3.tf, terraform.tfstate, terraform.tfstate.backup, and vpn.tf. The Editor window displays the content of vpn.tf, which defines an AWS VPC, an AWS VPN Gateway, an AWS Customer Gateway, and an AWS VPN Connection.

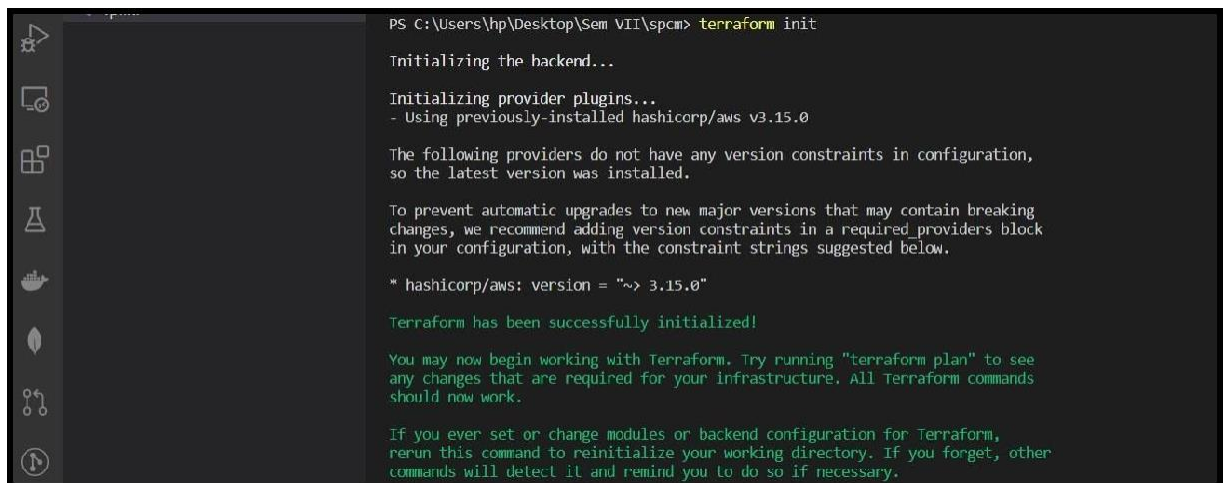
```
1 resource "aws_vpc" "vpc" {
2   cidr_block = "10.0.0.0/16"
3 }
4
5 resource "aws_vpn_gateway" "vpn_gateway" {
6   vpc_id = aws_vpc.vpc.id
7 }
8
9 resource "aws_customer_gateway" "customer_gateway" {
10   bgp_asn = 65000
11   ip_address = "172.0.0.1"
12   type = "ipsec.1"
13 }
14
15 resource "aws_vpn_connection" "main" {
16   vpn_gateway_id = aws_vpn_gateway.vpn_gateway.id
17   customer_gateway_id = aws_customer_gateway.customer_gateway.id
18   type = "ipsec.1"
19   static_routes_only = true
20 }
```

- Creating S3 bucket

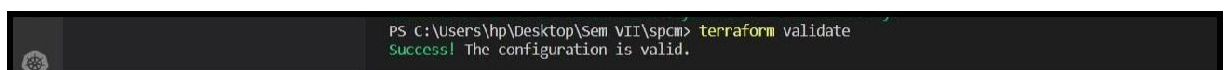


3. We will initialise , validate , plan , apply terraform which is installed on your system by running the following commands:

### Terraform init



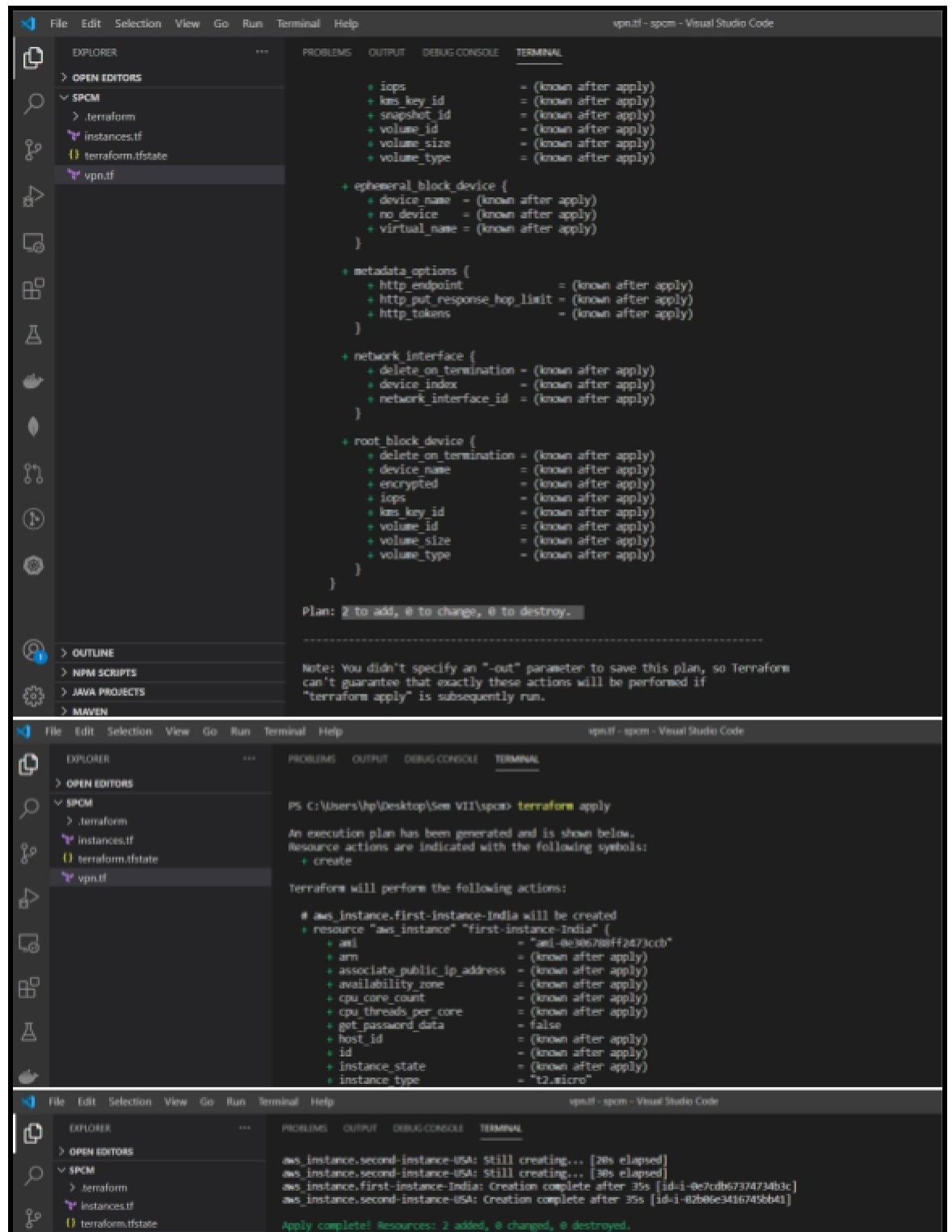
### Terraform validate



## Terraform plan

```
PS C:\Users\hp\Desktop\Sem VII\spcm> terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.
```

## Terraform apply



The first screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The file explorer shows a project named 'SPCM' with files '.terraform', 'instances.tf', 'terraform.tfstate', and 'vpn.tf'. The 'vpn.tf' file is selected. The main editor area displays the Terraform configuration for the 'vpn' resource, which includes an 'ephemeral\_block\_device' and a 'root\_block\_device'. The configuration is as follows:

```
resource "aws_instance" "vpn" {
  ami           = "ami-0e306788f2473ccb"
  instance_type = "t2.micro"

  ephemeral_block_device {
    device_name = "ephemeral0"
    no_device   = false
    virtual_name = "ephemeral0"
  }

  metadata_options {
    http_endpoint        = "http"
    http_put_response_hop_limit = 1
    http_tokens          = "required"
  }

  network_interface {
    delete_on_termination = true
    device_index           = 0
    network_interface_id   = "eni-0123456789abcdef"
  }

  root_block_device {
    delete_on_termination = true
    device_name           = "xvda"
    encrypted              = false
    iops                   = 100
    kms_key_id             = "arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
    volume_id              = "vol-0123456789abcdef"
    volume_size            = 10
    volume_type             = "gp2"
  }
}
```

The second screenshot shows the terminal output of the 'terraform apply' command. The output indicates that an execution plan has been generated and shows the actions that Terraform will perform. The actions are as follows:

```
PS C:\Users\hp\Desktop\Sem VII\spcm> terraform apply
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.first-instance-India will be created
+ resource "aws_instance" "first-instance-India" {
  ami           = "ami-0e306788f2473ccb"
  instance_type = "t2.micro"
  ...
}
```

The third screenshot shows the terminal output of the 'terraform apply' command, indicating that the resources have been successfully created. The output is as follows:

```
aws_instance.second-instance-USA: Still creating... [28s elapsed]
aws_instance.second-instance-USA: Still creating... [38s elapsed]
aws_instance.first-instance-India: Creation complete after 35s [id=i-0e7c0db67374734b3c]
aws_instance.second-instance-USA: Creation complete after 35s [id=i-82b86e3416745bb41]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

- The instances, VPN and S3 bucket have been created on your AWS cloud. One t2-micro ec2-instance is created in Mumbai region and the other in N. Virginia region.

The image displays three screenshots of the AWS Management Console, illustrating the setup of various AWS services.

**Top Screenshot: EC2 Instances**

The console shows the 'Instances (2)' page. The left sidebar lists navigation options: EC2 Dashboard, Events, Tags, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, and Savings Plans. The main content area shows a table of running instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
jenkins_insta...	i-0f264579bd714b80a	Running	t2.micro	Initializing	No alarms	us-east-1c	ec2-54-24...
jenkins_insta...	i-04bc0d88f95671fc	Running	t2.micro	Initializing	No alarms	us-east-1c	ec2-54-22...

**Middle Screenshot: Customer Gateway**

The console shows the 'Create Customer Gateway' page. The left sidebar lists navigation options: AWS NETWORK FIREWALL, Firewalls, Firewall policies, Network Firewall rule groups, VIRTUAL PRIVATE NETWORK (VPN), Customer Gateways (selected), Virtual Private Gateways, Site-to-Site VPN, Connections, and Client VPN Endpoints. The main content area shows a table of Customer Gateways:

Name	ID	State	Type	IP Address	BGP ASN	Certificate ARN
cgw-9d841179-f6e89586	cgw-9d841179-f6e89586	available	ipsec.1	172.31.1	65500	

**Bottom Screenshot: Amazon S3 Buckets**

The console shows the 'Amazon S3' page. The left sidebar lists navigation options: Buckets (selected), Access points, Batch Operations, Access analyzer for S3, Account settings for Block Public Access, and Feature spotlight. The main content area shows a table of S3 Buckets:

Name	Region	Access	Creation date
s3bucket1995	US East (N. Virginia) us-east-1	Objects can be public	November 18, 2020, 15:19 (UTC+05:30)