SYSTEM PROVISIONING AND CONFIGURATION MANAGEMENT

(ASSIGNMENT 1)

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

What is Terraform?

Terraform is an open source tool for infrastructure provisioning created by HashiCorp. It provides Infrastructure as code allowing you to automate and manage your infrastructure, platform and your services that run on the platform. Terraform can manage existing and popular service providers(aws, azure, GCP etc).

You do not have to prepare infrastructure like private network space, ec2 server instances, installing docker and other tools and security. Terraform does all that for you by preparing the whole infrastructure using terraform scripts. Thus, it is a software tool that provides Infrastructure as code.

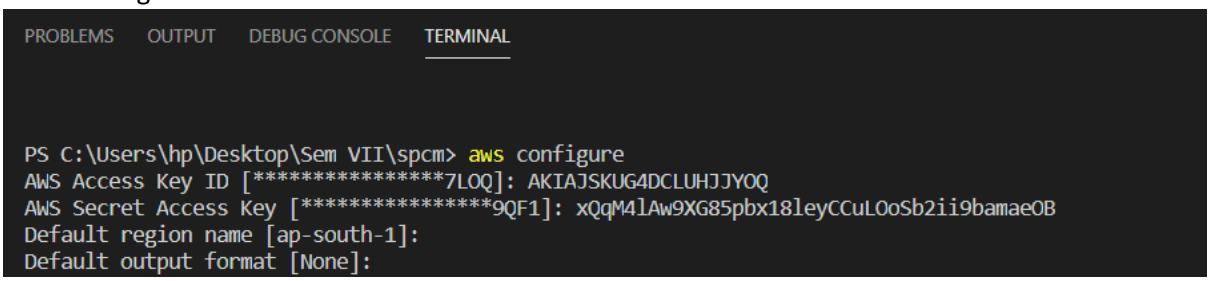
Terraform is declarative which means you define what you want.

Steps to provision

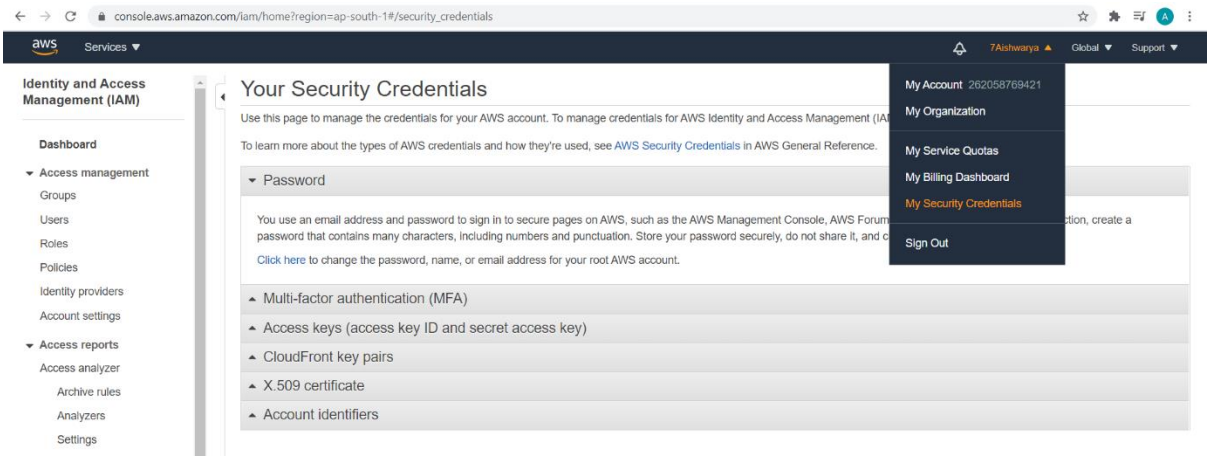
→ Download the terraform binary file https://www.terraform.io/downloads.html
→ Extract the zip file and add the terraform binary on the PATH.
→ Create a directory and go to your directory using the following commands
mkdir spcm
cd spcm

Configure aws credentials

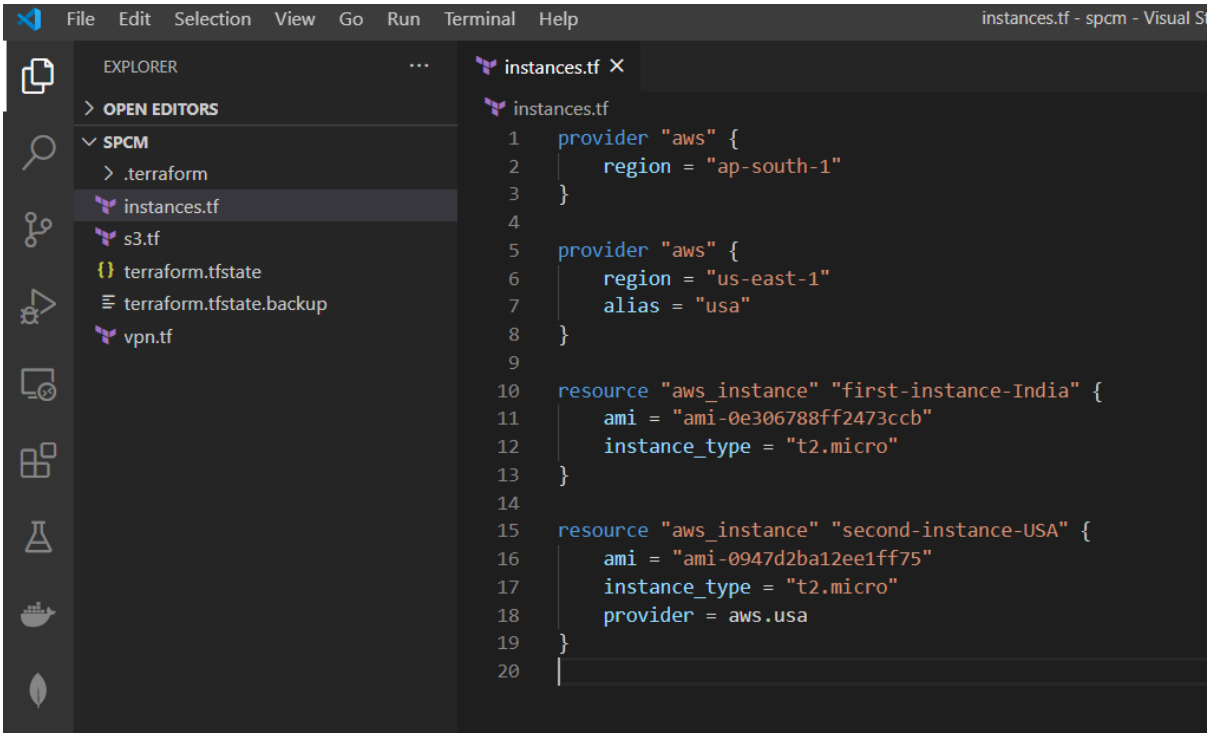
aws configure



You can get your secret key and access key from your aws console.
Go to profile > My security credentials > Access keys
Use an existing access key or create a new key

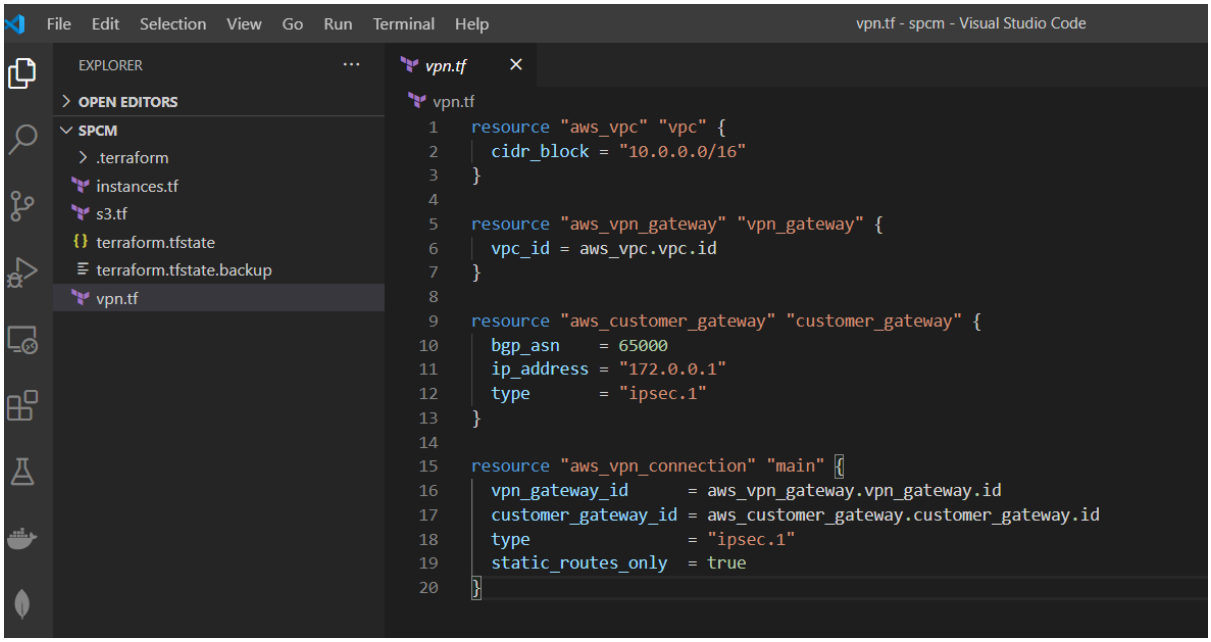


Create terraform scripts

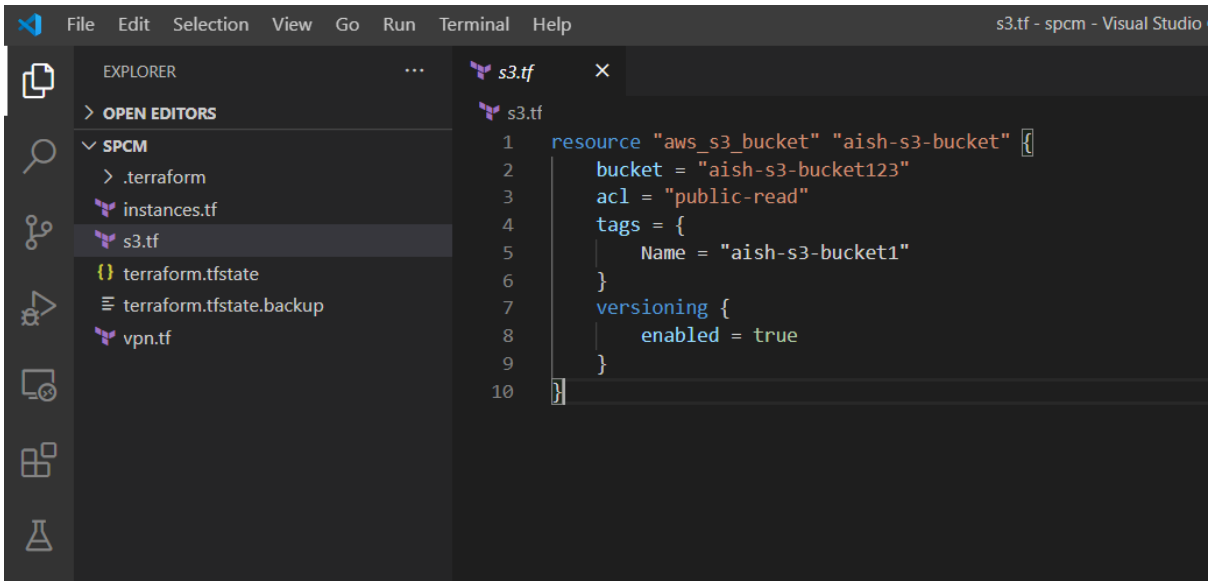


We create 2 aws providers, one in ap-south-1(Mumbai region) which is the default region and the other in us-east-1(Virginia region - USA)

Creating VPN



Creating S3 bucket



Run the following commands

terraform init

Initializes working directory containing terraform configuration files. It is safe to run this command multiple times

terraform validate

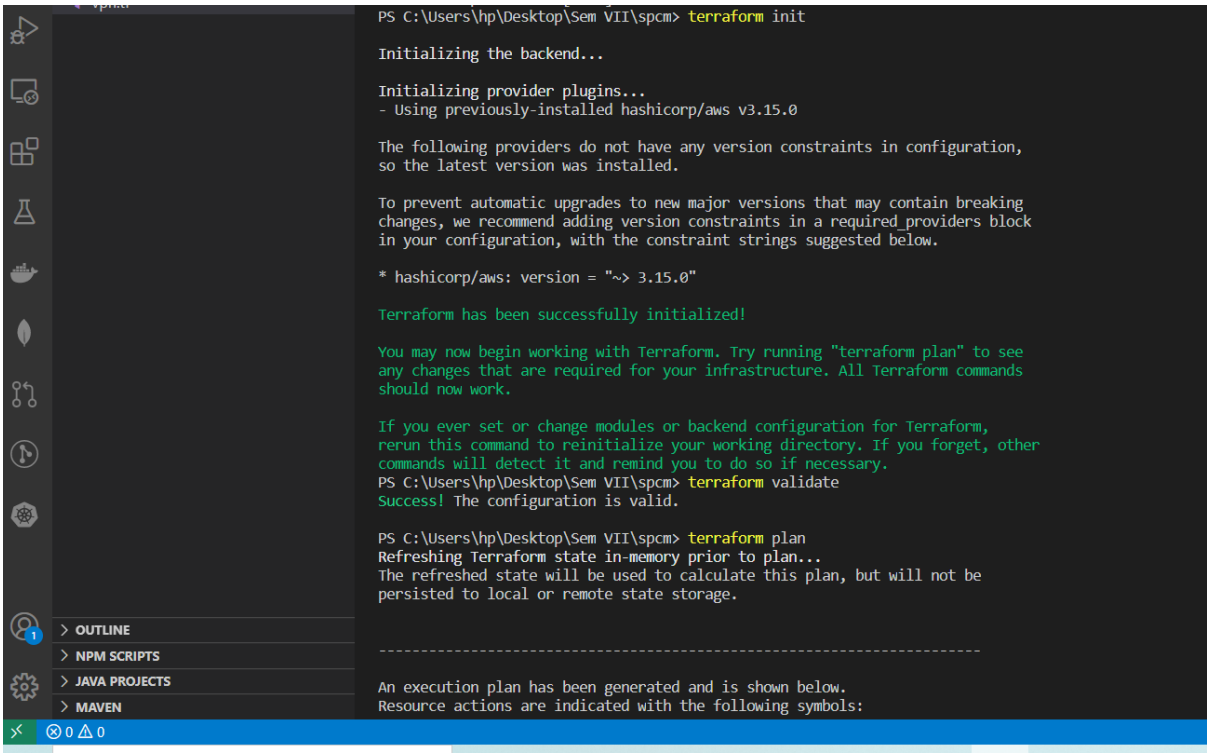
To check if terraform scripts are no syntax errors and is internally consistent

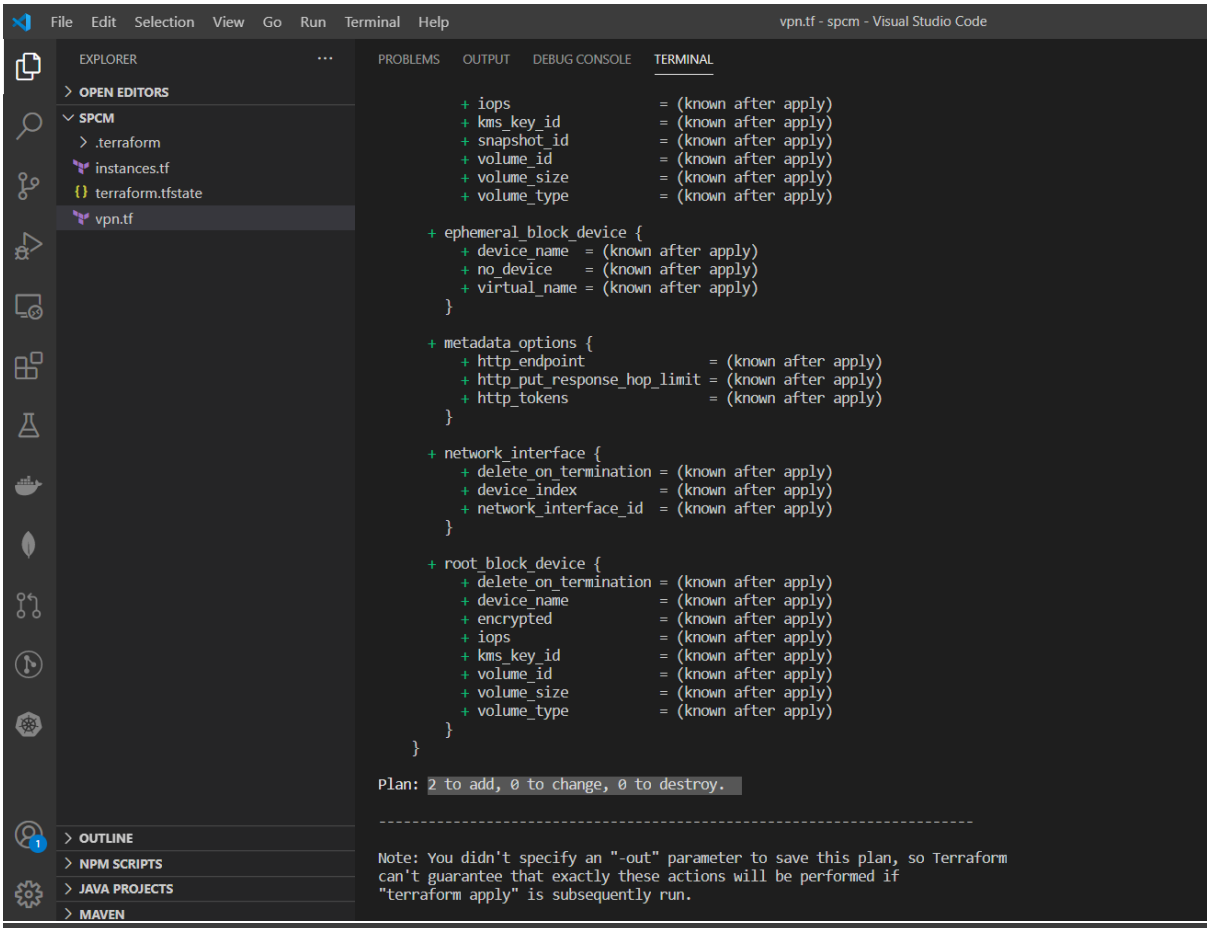
terraform plan

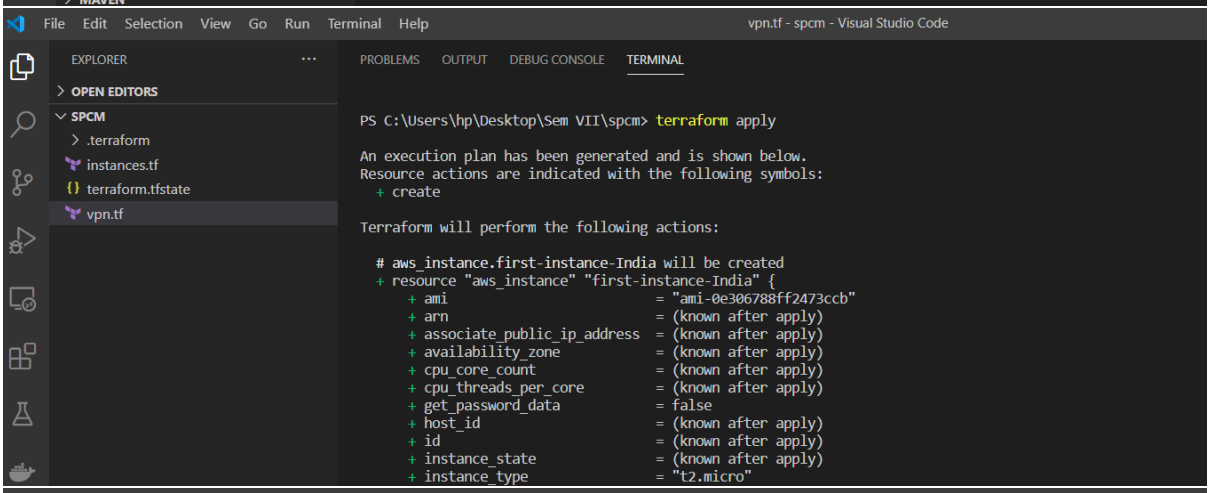
To create execution plan that helps you check whether execution plan matches your Expectations

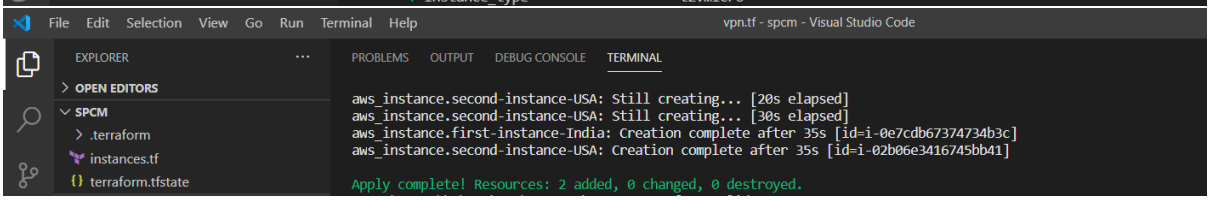
terraform apply

To apply the changes to reach the desired state of the configuration



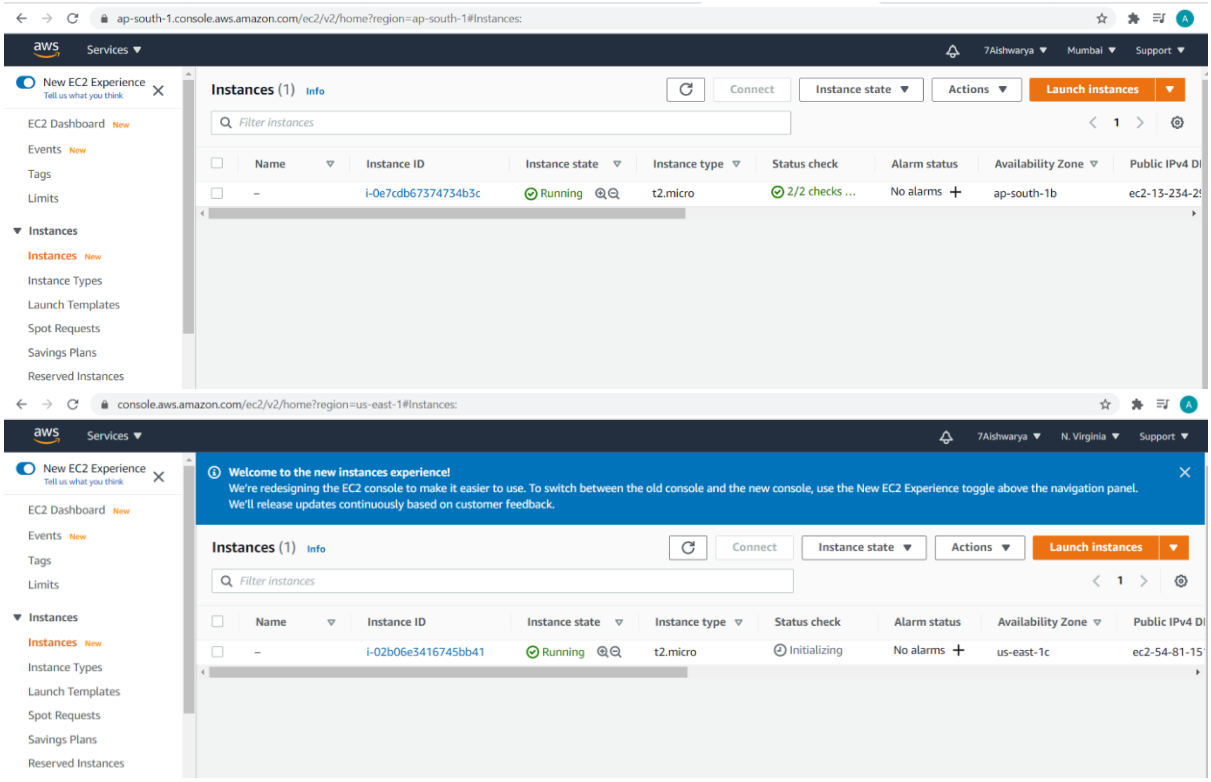

File Edit Selection View Go Run Terminal Help
vpn.tf - spcm - Visual Studio Code
EXPLORER
OPEN EDITORS
SPCM
.terraform
instances.tf
terraform.tfstate
vpn.tf
OUTLINE
NPM SCRIPTS
JAVA PROJECTS
MAVEN
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
+ iops = (known after apply)
+ kms_key_id = (known after apply)
+ snapshot_id = (known after apply)
+ volume_id = (known after apply)
+ volume_size = (known after apply)
+ volume_type = (known after apply)
+ ephemeral_block_device {
+ device_name = (known after apply)
+ no_device = (known after apply)
+ virtual_name = (known after apply)
}
+ metadata_options {
+ http_endpoint = (known after apply)
+ http_put_response_hop_limit = (known after apply)
+ http_tokens = (known after apply)
}
+ network_interface {
+ delete_on_termination = (known after apply)
+ device_index = (known after apply)
+ network_interface_id = (known after apply)
}
+ root_block_device {
+ delete_on_termination = (known after apply)
+ device_name = (known after apply)
+ encrypted = (known after apply)
+ iops = (known after apply)
+ kms_key_id = (known after apply)
+ volume_id = (known after apply)
+ volume_size = (known after apply)
+ volume_type = (known after apply)
}
}
Plan: 2 to add, 0 to change, 0 to destroy.
Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.


File Edit Selection View Go Run Terminal Help
vpn.tf - spcm - Visual Studio Code
EXPLORER
OPEN EDITORS
SPCM
.terraform
instances.tf
terraform.tfstate
vpn.tf
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
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-0e306788ff2473ccb"
+ 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)
+ get_password_data = false
+ host_id = (known after apply)
+ id = (known after apply)
+ instance_state = (known after apply)
+ instance_type = "t2.micro"

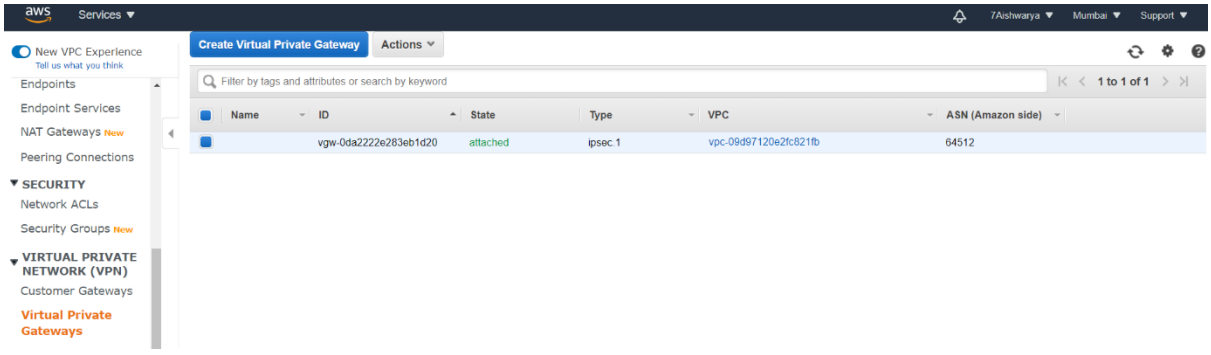

File Edit Selection View Go Run Terminal Help
vpn.tf - spcm - Visual Studio Code
EXPLORER
OPEN EDITORS
SPCM
.terraform
instances.tf
terraform.tfstate
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
aws_instance.second-instance-USA: Still creating... [20s elapsed]
aws_instance.second-instance-USA: Still creating... [30s elapsed]
aws_instance.first-instance-India: Creation complete after 35s [id=i-0e7cdb67374734b3c]
aws_instance.second-instance-USA: Creation complete after 35s [id=i-02b06e3416745bb41]
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Now you can check the instances, VPN and S3 bucket have been created on your AWS cloud.

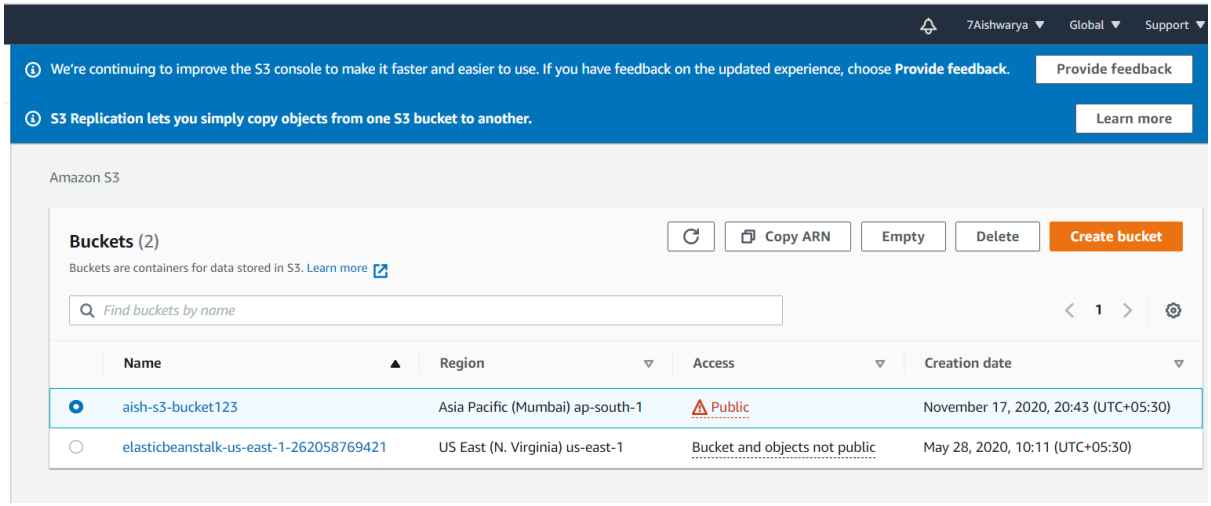
One t2-micro ec2-instance is created in Mumbai region and the other in N. Virginia region.



VPN



S3 bucket



You can destroy all the resources you created by using only a single command, i.e., terraform destroy and all your resources including all instances will be destroyed.