

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.

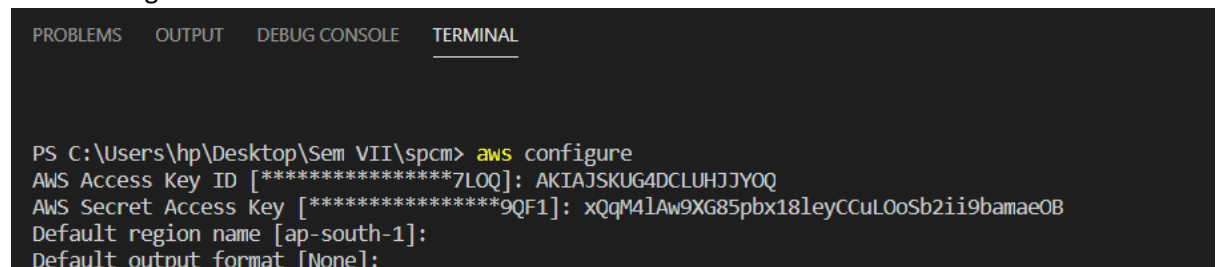
Terraform is declarative which means you define what you want.

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



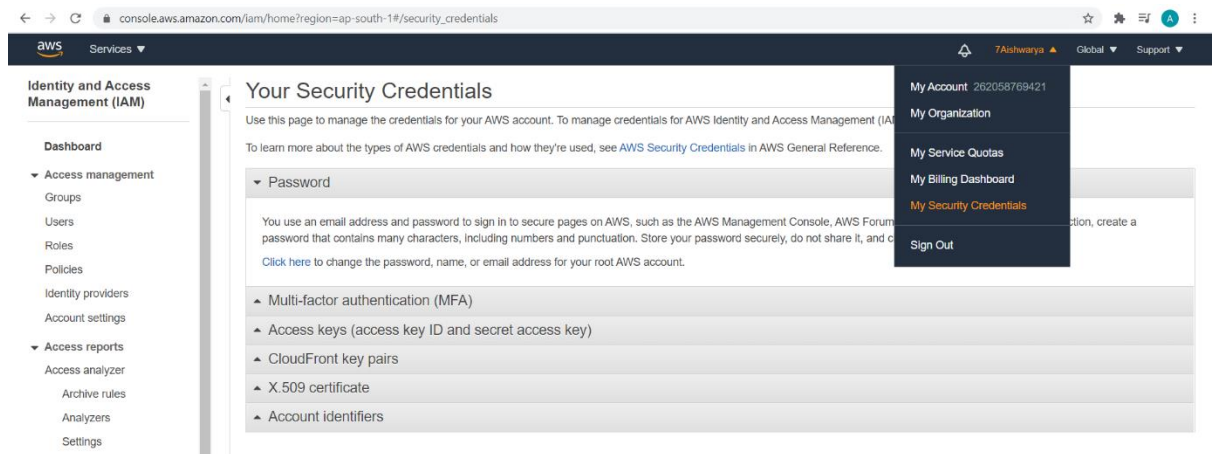
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

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

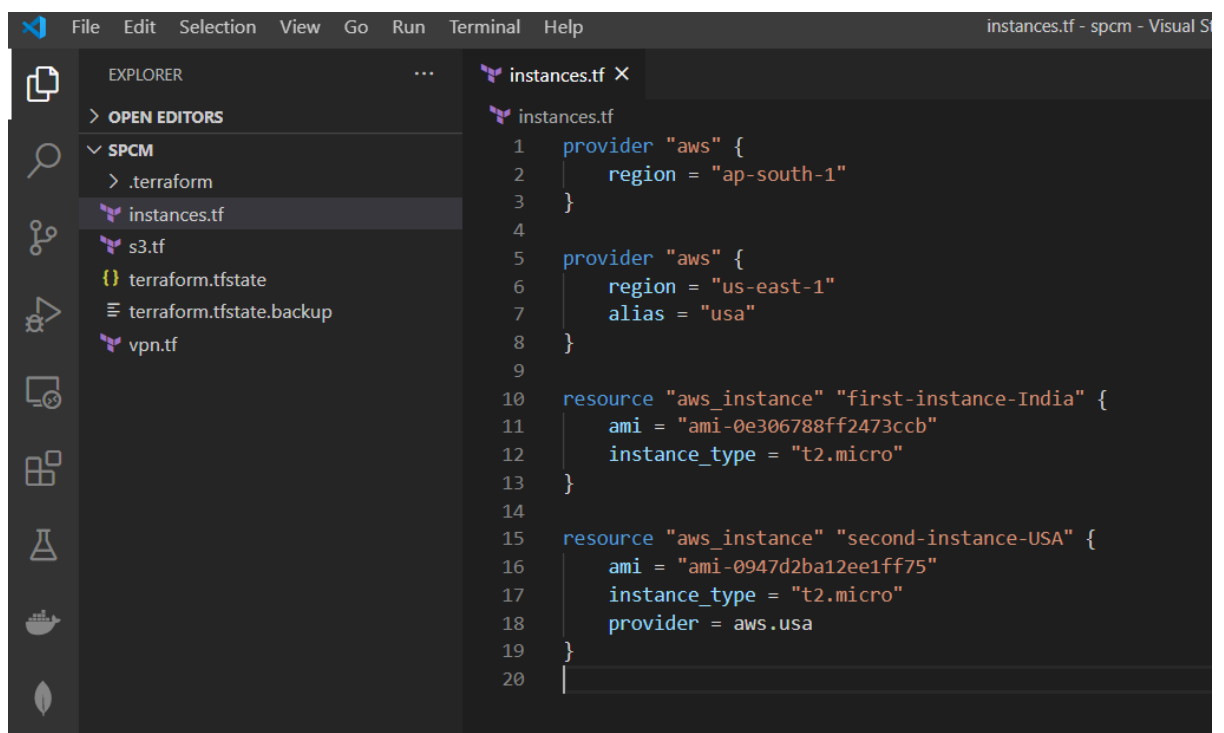
You can get your secret key and access key from your aws console.

Go to profile > My security credentials > Access keys

Use an existing access key or create a new key

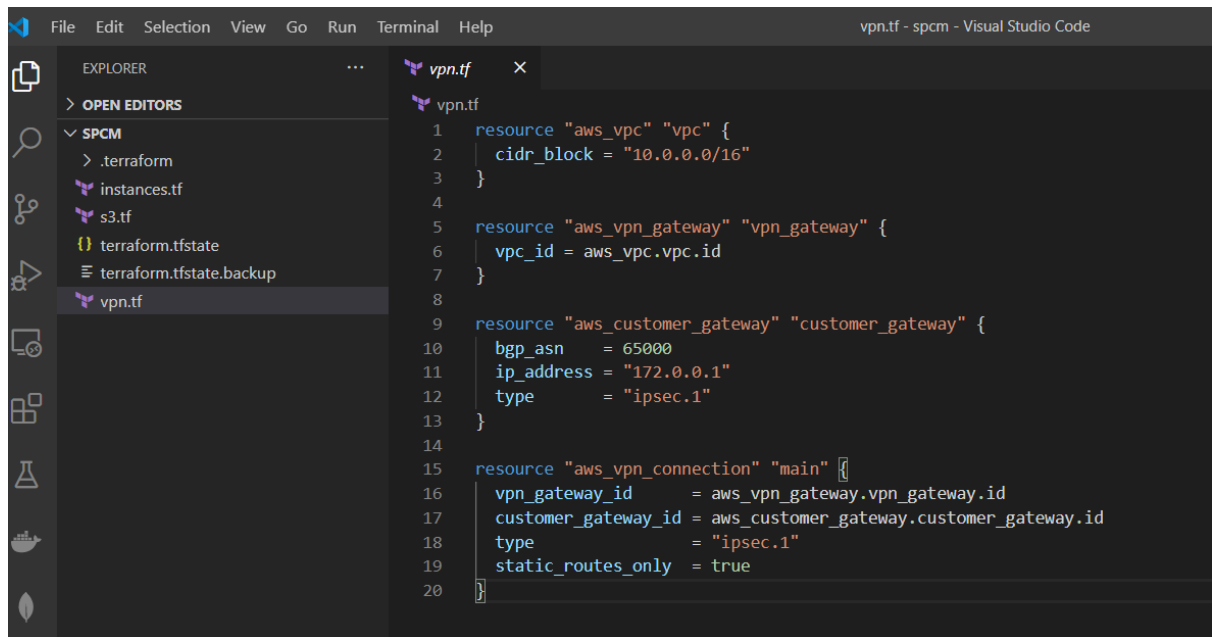


Create terraform scripts



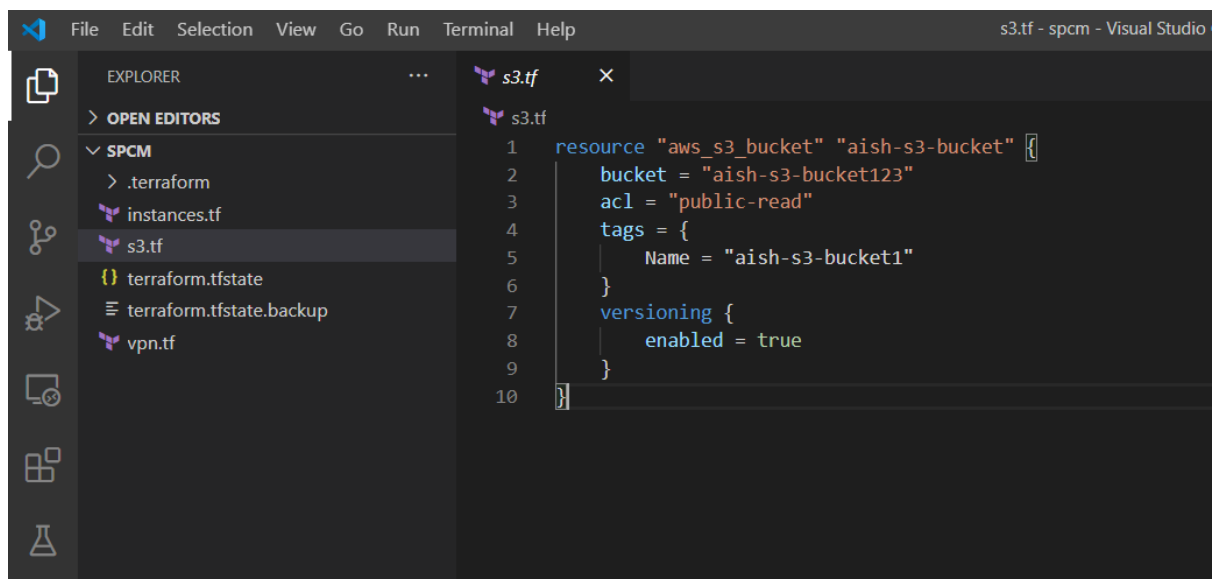
We create 2 aws providers, one in ap-south-1(Mumbai region) which is the default region and the other in us-east-1(Virginia region - USA)

Creating VPN



```
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



```
1 resource "aws_s3_bucket" "aish-s3-bucket" {
2   |   bucket = "aish-s3-bucket123"
3   |   acl    = "public-read"
4   |   tags = {
5   |     Name = "aish-s3-bucket1"
6   |   }
7   |   versioning {
8   |     enabled = true
9   |   }
10 }
```

Run the following commands

terraform init

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

terraform validate

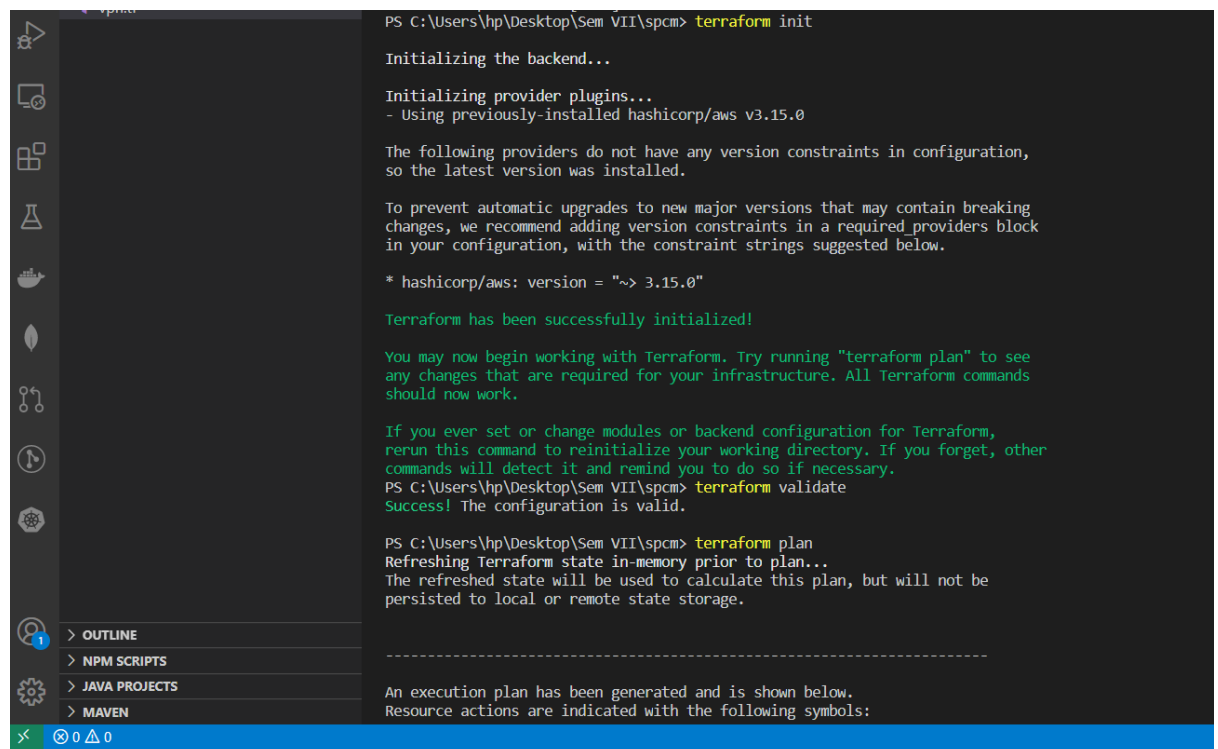
To check if terraform scripts are no syntax errors and is internally consistent

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



```
PS C:\Users\hp\Desktop\Sem VII\spcm> terraform init

Initializing the backend...

Initializing provider plugins...
- Using previously-installed hashicorp/aws v3.15.0

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, we recommend adding version constraints in a required providers block
in your configuration, with the constraint strings suggested below.

* hashicorp/aws: version = "~> 3.15.0"

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\hp\Desktop\Sem VII\spcm> terraform validate
Success! The configuration is valid.

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.

-----

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
```

File Edit Selection View Go Run Terminal Helpvpn.tf - spcm - Visual Studio Code

EXPLORER...

> OPEN EDITORS

SPCM

> .terraform

instances.tf

terraform.tfstate

vpn.tf

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 Helpvpn.tf - spcm - Visual Studio Code

EXPLORER...

> OPEN EDITORS

SPCM

> .terraform

instances.tf

terraform.tfstate

vpn.tf

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\hnp\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 Helpvpn.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.

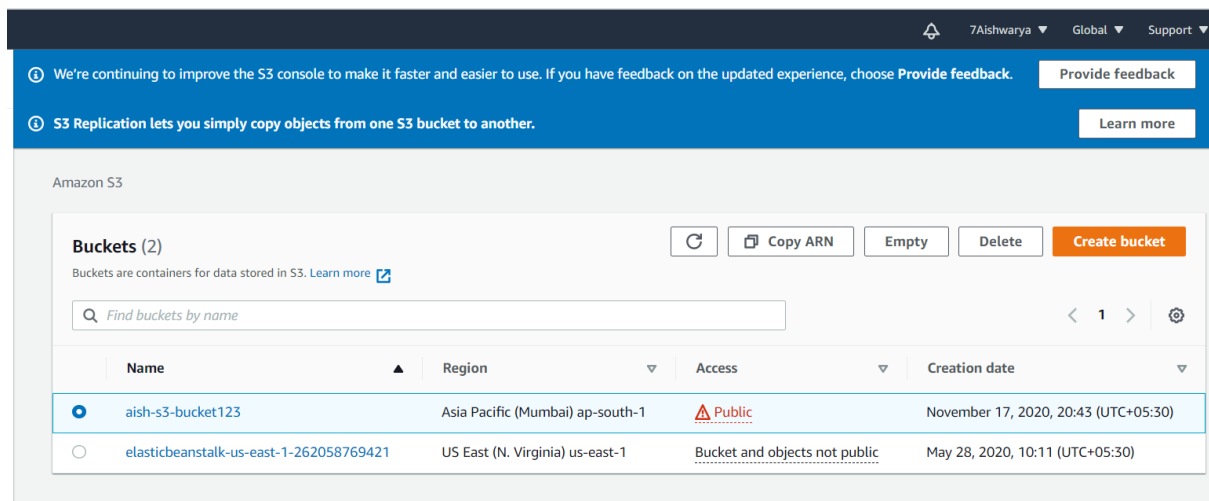
The first screenshot shows the AWS Management Console for the Mumbai region (ap-south-1). It displays a single EC2 instance with the ID i-0e7cdb67374734b3c, which is in a 'Running' state. The instance is a t2.micro type, located in the ap-south-1b availability zone. The console interface includes a sidebar with navigation options like 'EC2 Dashboard', 'Events', 'Tags', 'Limits', and 'Instances'. The main panel shows the 'Instances (1)' list with a search bar and a table of instance details.

The second screenshot shows the AWS Management Console for the N. Virginia region (us-east-1). It displays a single EC2 instance with the ID i-02b06e3416745bb41, which is in a 'Running' state. The instance is a t2.micro type, located in the us-east-1c availability zone. The console interface is similar to the first screenshot, but it includes a 'Welcome to the new instances experience!' banner at the top. The main panel shows the 'Instances (1)' list with a search bar and a table of instance details.

VPN

The screenshot shows the AWS Management Console for the Mumbai region (ap-south-1). It displays the 'Virtual Private Gateways' page. The console interface includes a sidebar with navigation options like 'Endpoints', 'Endpoint Services', 'NAT Gateways', 'Peering Connections', 'SECURITY', and 'VIRTUAL PRIVATE NETWORK (VPN)'. The main panel shows the 'Create Virtual Private Gateway' button and a table of existing gateways. The table has columns for Name, ID, State, Type, VPC, and ASN (Amazon side). There is one gateway listed with the ID vgw-0da2222e283eb1d20, which is in an 'attached' state. It is of type 'ipsec.1' and is associated with the VPC vpc-09d97120e2fc821fb. The ASN (Amazon side) is 64512.

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.

