Assignment I

Name: Nishkarsh Raj

Subject: System Provisioning and Configuration Management

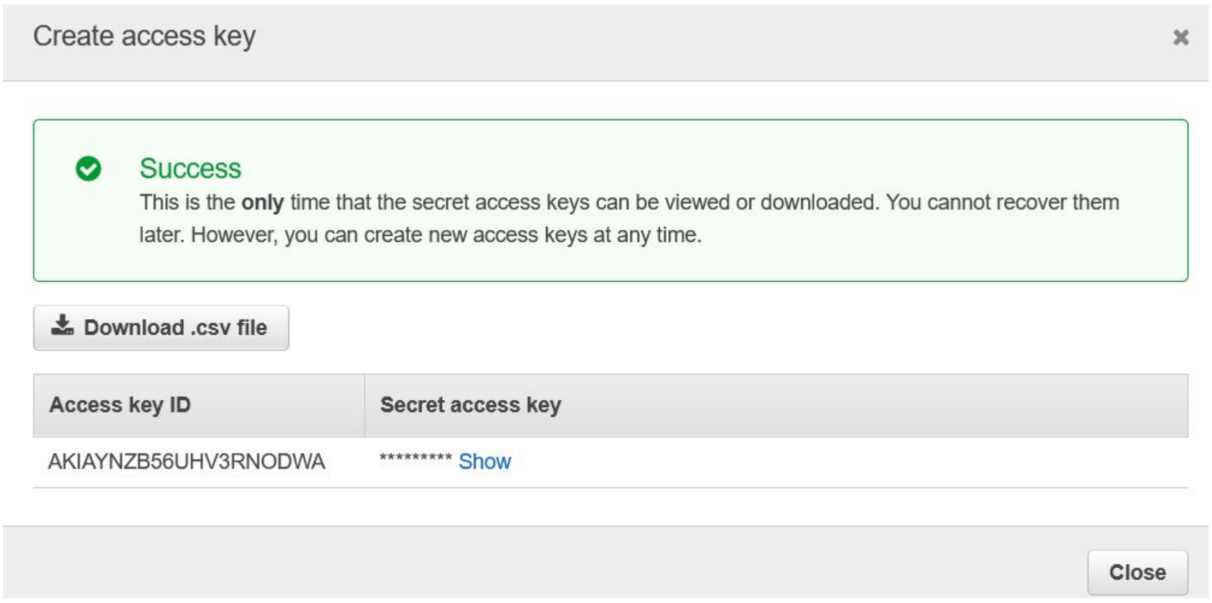
Faculty: Dr. Hitesh Kumar Sharma

Batch: CSE-DevOps - Xebia

SAP ID: 500060720

Roll Number: 41

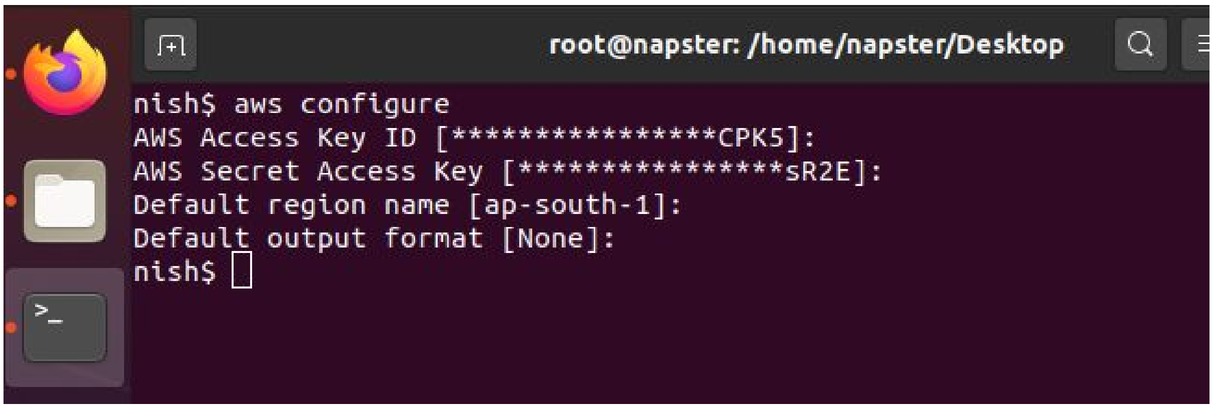
- Generate AWS CLI Credentials via IAM


Create access key

Success
This is the **only** time that the secret access keys can be viewed or downloaded. You cannot recover them later. However, you can create new access keys at any time.

Access key ID	Secret access key
AKIAYNZB56UHV3RNODWA	********* Show

- Setup AWS CLI locally



```
nish$ aws configure
AWS Access Key ID [****************CPK5]:
AWS Secret Access Key [****************sR2E]:
Default region name [ap-south-1]:
Default output format [None]:
nish$
```

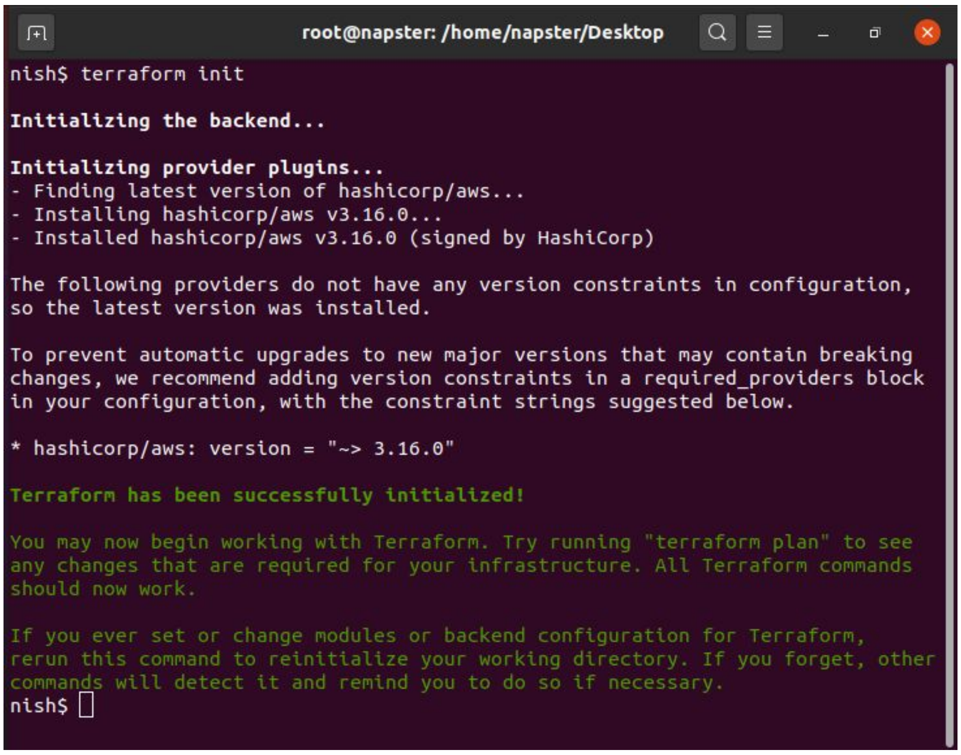
- Terraform script to launch two EC2 Instance, VPN and a S3 bucket:

```
provider "aws" {
  region = "ap-south-1"
}

resource "aws_instance" "nish" {
  ami           = "ami-02b5fbc2cb28b77b8"
  count         = 2
  instance_type = "t2.micro"
  tags = {
          Name = "noicecurse"
  }
}
```

```
resource "aws_s3_bucket" "nish" {
  bucket = "noicecurse123" # must be unique in universal namespace
  acl    = "private"
}
resource "aws_vpc" "vpc" {
  cidr_block = "10.0.0.0/16"
}
resource "aws_vpn_gateway" "vpn_gateway" {
  vpc_id = "vpc-056b8f722fbfcb60f" # must be your default vpc id
}
resource "aws_customer_gateway" "customer_gateway" {
  bgp_asn    = 65000
  ip_address = "172.0.0.1"
  type       = "ipsec.1"
}
resource "aws_vpn_connection" "main" {
  vpn_gateway_id      = aws_vpn_gateway.vpn_gateway.id
  customer_gateway_id = aws_customer_gateway.customer_gateway.id
  type                = "ipsec.1"
  static_routes_only  = true
}
```

- Initialize Terraform plugins

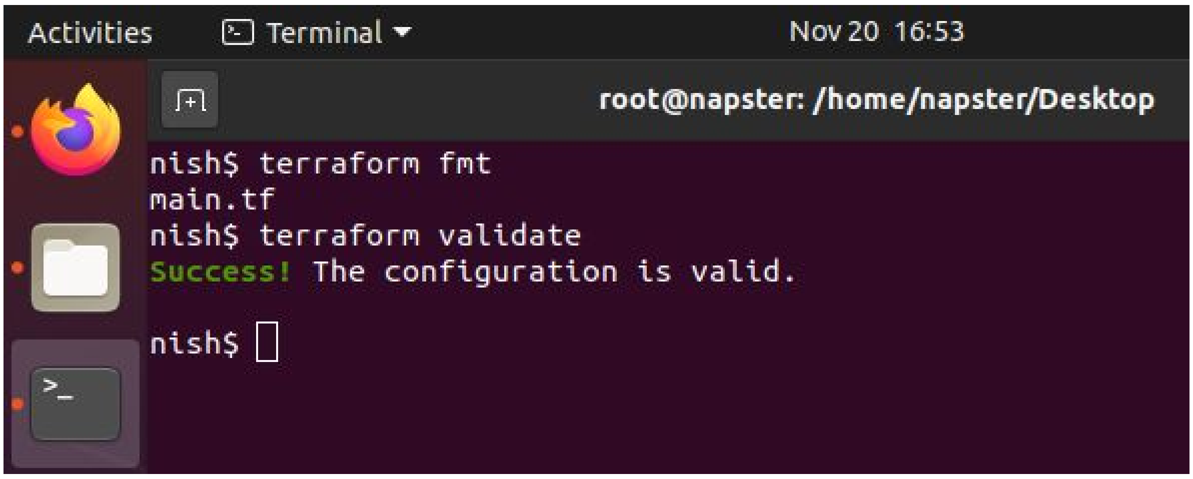
$ terraform init



```
nish$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v3.16.0...
- Installed hashicorp/aws v3.16.0 (signed by HashiCorp)

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 = "~> 3.16.0"

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.
nish$ 
```

- Format the terraform script and validate it

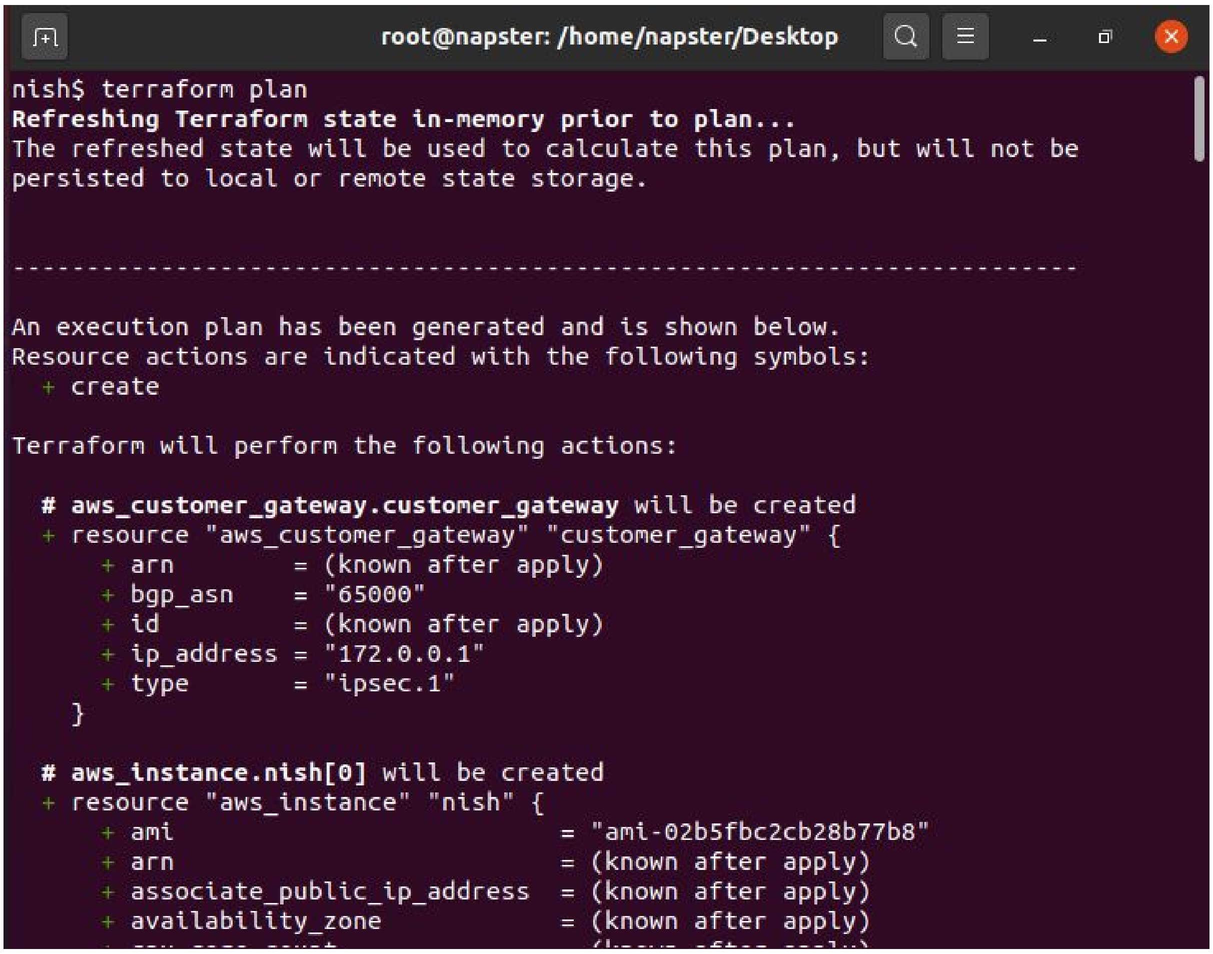
$ terraform fmt
$ terraform validate



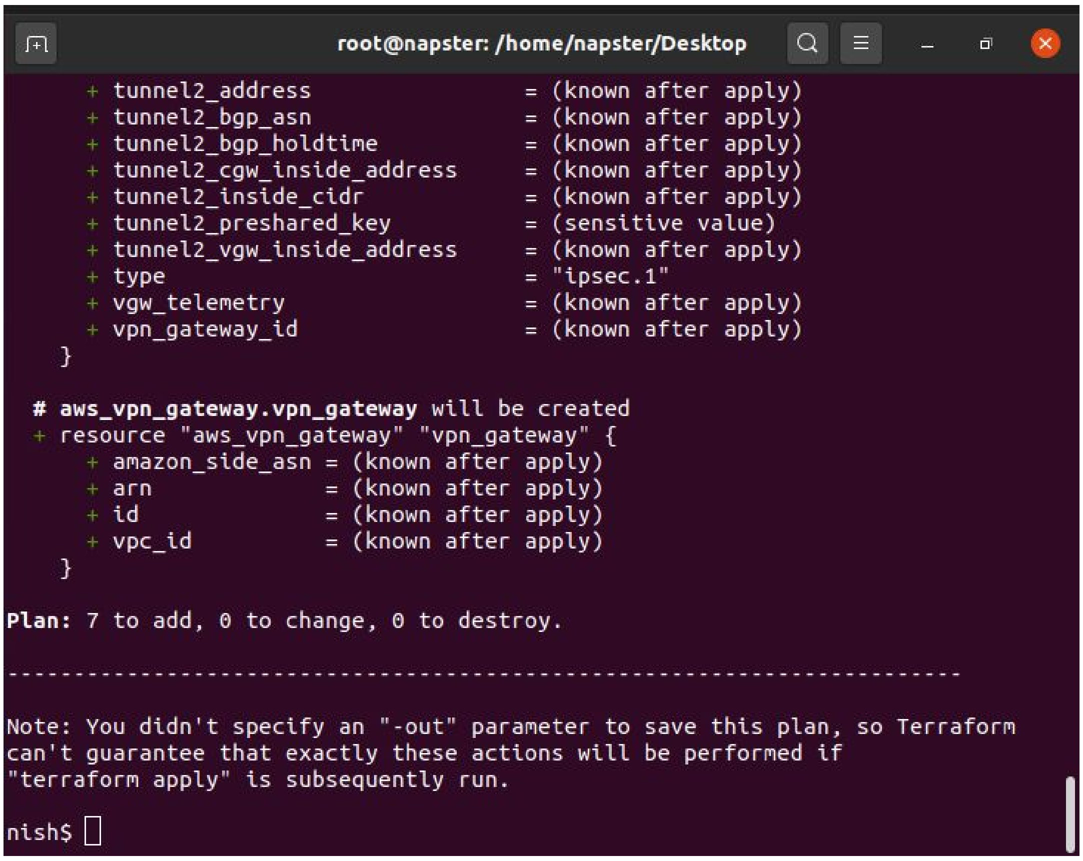
```
nish$ terraform fmt
main.tf
nish$ terraform validate
Success! The configuration is valid.

nish$
```

- See the potential changes using plan command

$ terraform plan



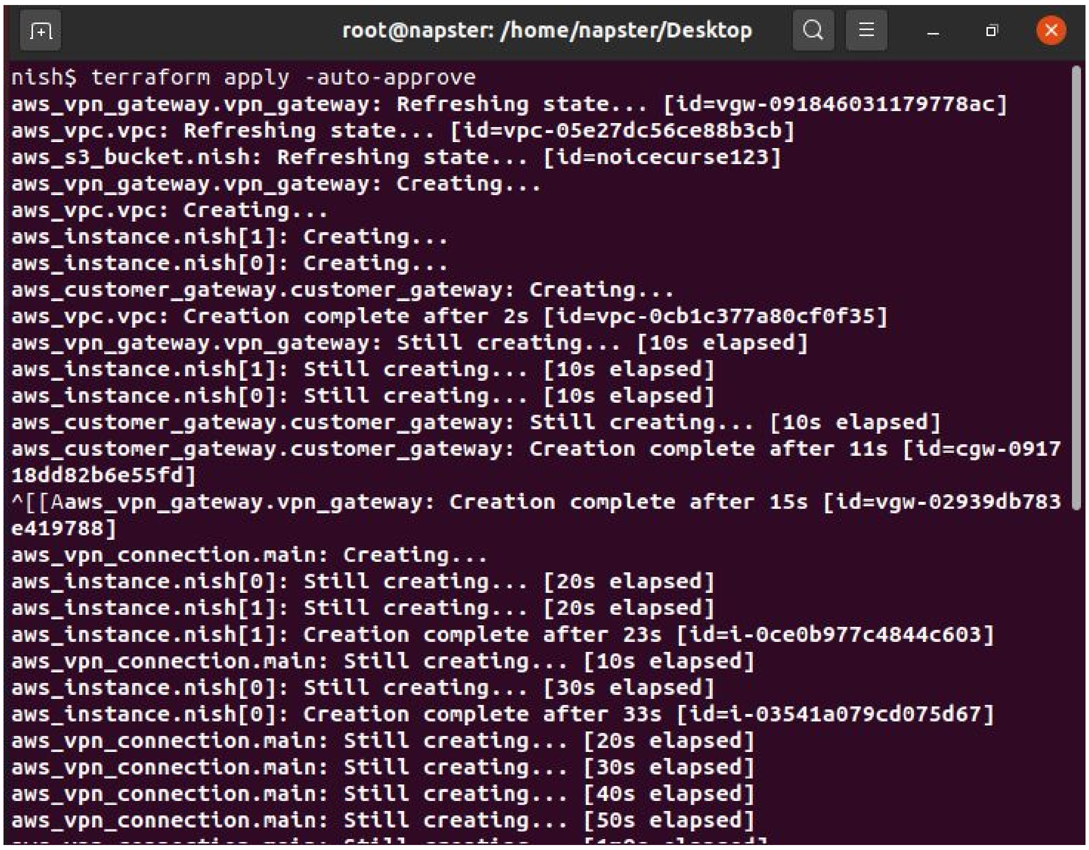
```
nish$ terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.


------------------------------------------------------------------------

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

Terraform will perform the following actions:

  # aws_customer_gateway.customer_gateway will be created
  + resource "aws_customer_gateway" "customer_gateway" {
      + arn        = (known after apply)
      + bgp_asn    = "65000"
      + id         = (known after apply)
      + ip_address = "172.0.0.1"
      + type       = "ipsec.1"
    }

  # aws_instance.nish[0] will be created
  + resource "aws_instance" "nish" {
      + ami                          = "ami-02b5fbc2cb28b77b8"
      + arn                          = (known after apply)
      + associate_public_ip_address  = (known after apply)
      + availability_zone            = (known after apply)
```



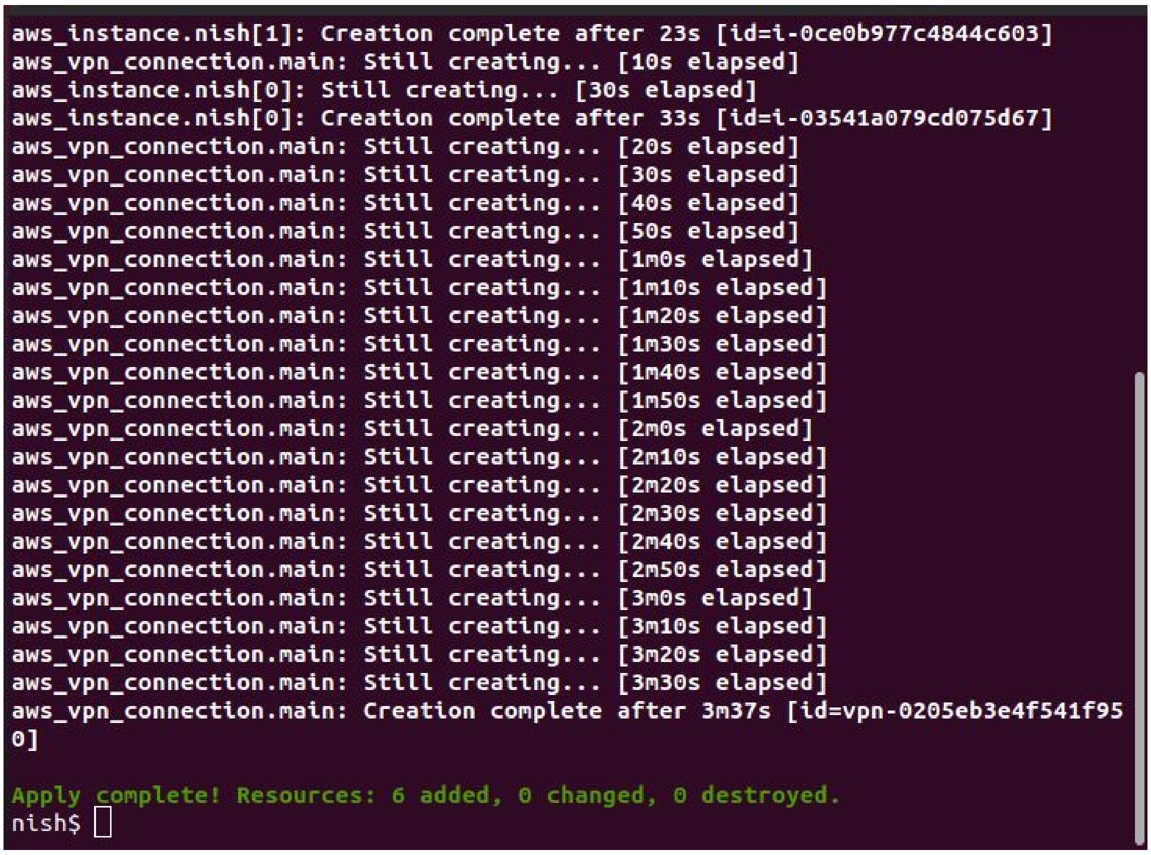
```
      + tunnel2_address                 = (known after apply)
      + tunnel2_bgp_asn                 = (known after apply)
      + tunnel2_bgp_holdtime            = (known after apply)
      + tunnel2_cgw_inside_address      = (known after apply)
      + tunnel2_inside_cidr             = (known after apply)
      + tunnel2_preshared_key           = (sensitive value)
      + tunnel2_vgw_inside_address      = (known after apply)
      + type                            = "ipsec.1"
      + vgw_telemetry                   = (known after apply)
      + vpn_gateway_id                  = (known after apply)
    }

  # aws_vpn_gateway.vpn_gateway will be created
  + resource "aws_vpn_gateway" "vpn_gateway" {
      + amazon_side_asn = (known after apply)
      + arn             = (known after apply)
      + id              = (known after apply)
      + vpc_id          = (known after apply)
    }

Plan: 7 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.

nish$ 
```

- Launch instances using terraform apply

$ terraform apply -auto-approve

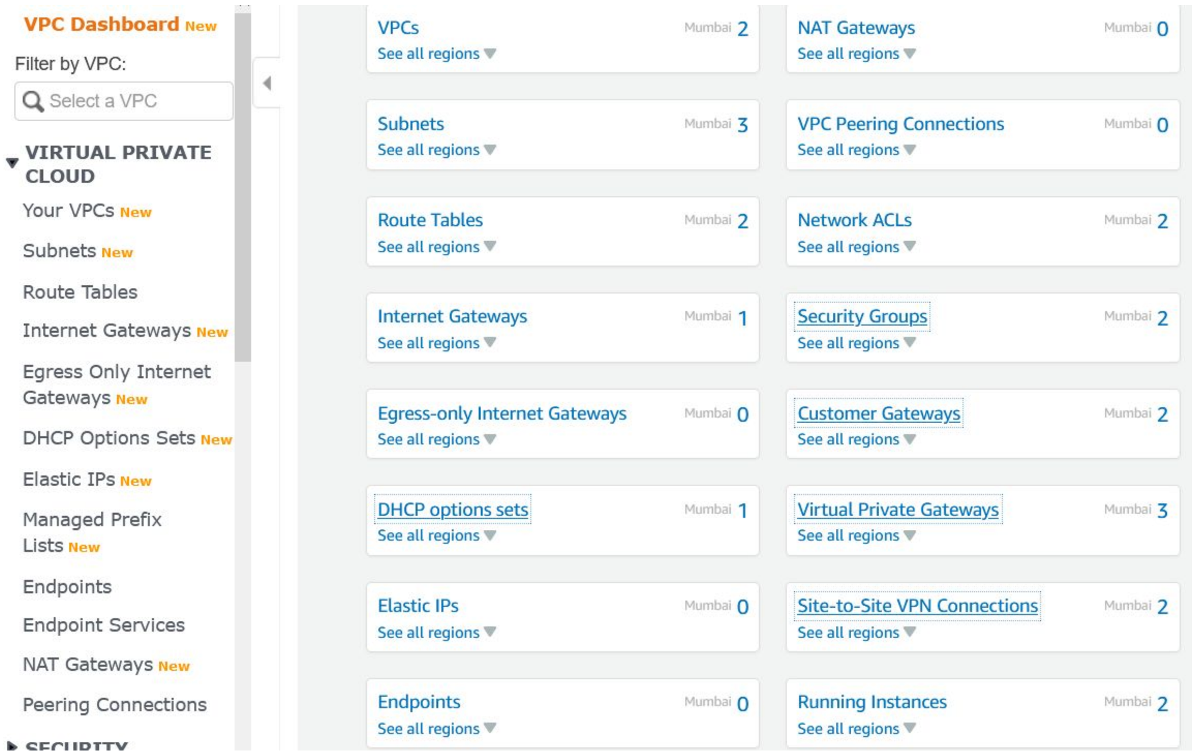


```
nish$ terraform apply -auto-approve
aws_vpn_gateway.vpn_gateway: Refreshing state... [id=vgw-091846031179778ac]
aws_vpc.vpc: Refreshing state... [id=vpc-05e27dc56ce88b3cb]
aws_s3_bucket.nish: Refreshing state... [id=noicecurse123]
aws_vpn_gateway.vpn_gateway: Creating...
aws_vpc.vpc: Creating...
aws_instance.nish[1]: Creating...
aws_instance.nish[0]: Creating...
aws_customer_gateway.customer_gateway: Creating...
aws_vpc.vpc: Creation complete after 2s [id=vpc-0cb1c377a80cf0f35]
aws_vpn_gateway.vpn_gateway: Still creating... [10s elapsed]
aws_instance.nish[1]: Still creating... [10s elapsed]
aws_instance.nish[0]: Still creating... [10s elapsed]
aws_customer_gateway.customer_gateway: Still creating... [10s elapsed]
aws_customer_gateway.customer_gateway: Creation complete after 11s [id=cgw-0917
18dd82b6e55fd]
^[[Aaws_vpn_gateway.vpn_gateway: Creation complete after 15s [id=vgw-02939db783
e419788]
aws_vpn_connection.main: Creating...
aws_instance.nish[0]: Still creating... [20s elapsed]
aws_instance.nish[1]: Still creating... [20s elapsed]
aws_instance.nish[1]: Creation complete after 23s [id=i-0ce0b977c4844c603]
aws_vpn_connection.main: Still creating... [10s elapsed]
aws_instance.nish[0]: Still creating... [30s elapsed]
aws_instance.nish[0]: Creation complete after 33s [id=i-03541a079cd075d67]
aws_vpn_connection.main: Still creating... [20s elapsed]
aws_vpn_connection.main: Still creating... [30s elapsed]
aws_vpn_connection.main: Still creating... [40s elapsed]
aws_vpn_connection.main: Still creating... [50s elapsed]
```



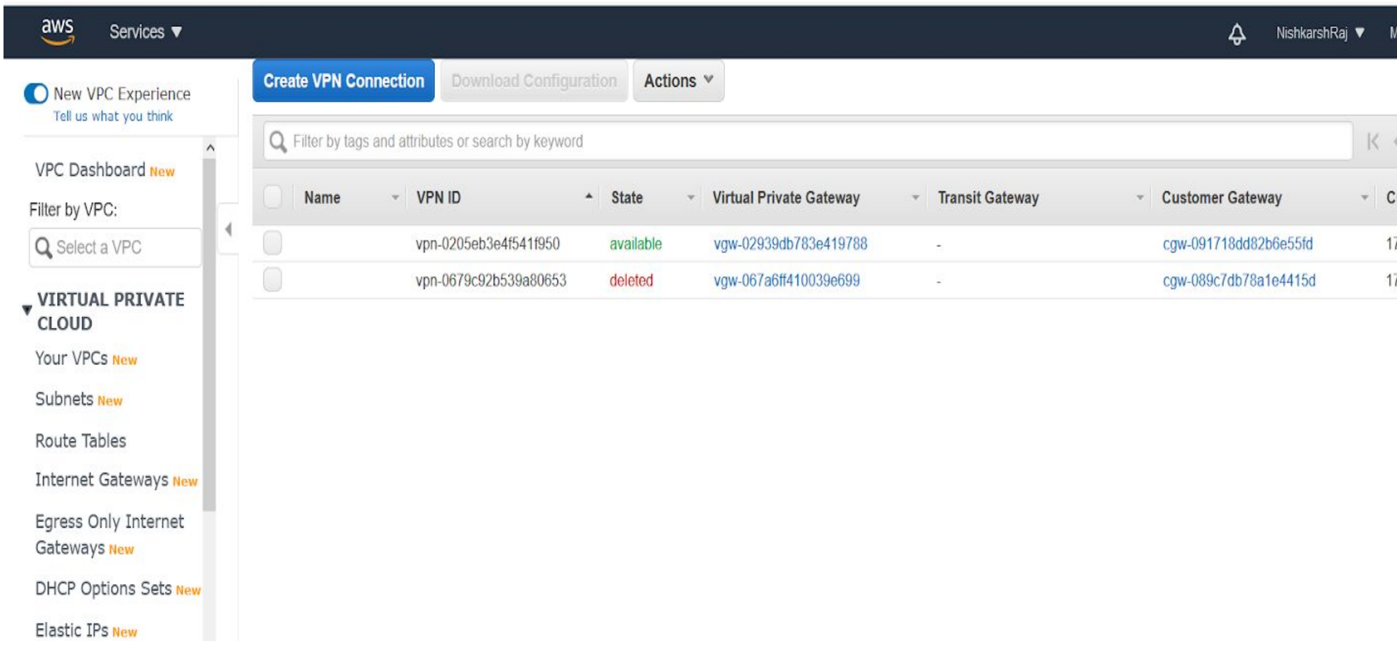
```
aws_instance.nish[1]: Creation complete after 23s [id=i-0ce0b977c4844c603]
aws_vpn_connection.main: Still creating... [10s elapsed]
aws_instance.nish[0]: Still creating... [30s elapsed]
aws_instance.nish[0]: Creation complete after 33s [id=i-03541a079cd075d67]
aws_vpn_connection.main: Still creating... [20s elapsed]
aws_vpn_connection.main: Still creating... [30s elapsed]
aws_vpn_connection.main: Still creating... [40s elapsed]
aws_vpn_connection.main: Still creating... [50s elapsed]
aws_vpn_connection.main: Still creating... [1m0s elapsed]
aws_vpn_connection.main: Still creating... [1m10s elapsed]
aws_vpn_connection.main: Still creating... [1m20s elapsed]
aws_vpn_connection.main: Still creating... [1m30s elapsed]
aws_vpn_connection.main: Still creating... [1m40s elapsed]
aws_vpn_connection.main: Still creating... [1m50s elapsed]
aws_vpn_connection.main: Still creating... [2m0s elapsed]
aws_vpn_connection.main: Still creating... [2m10s elapsed]
aws_vpn_connection.main: Still creating... [2m20s elapsed]
aws_vpn_connection.main: Still creating... [2m30s elapsed]
aws_vpn_connection.main: Still creating... [2m40s elapsed]
aws_vpn_connection.main: Still creating... [2m50s elapsed]
aws_vpn_connection.main: Still creating... [3m0s elapsed]
aws_vpn_connection.main: Still creating... [3m10s elapsed]
aws_vpn_connection.main: Still creating... [3m20s elapsed]
aws_vpn_connection.main: Still creating... [3m30s elapsed]
aws_vpn_connection.main: Creation complete after 3m37s [id=vpn-0205eb3e4f541f95
0]

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.
nish$ 
```

- Verify creation of VPC



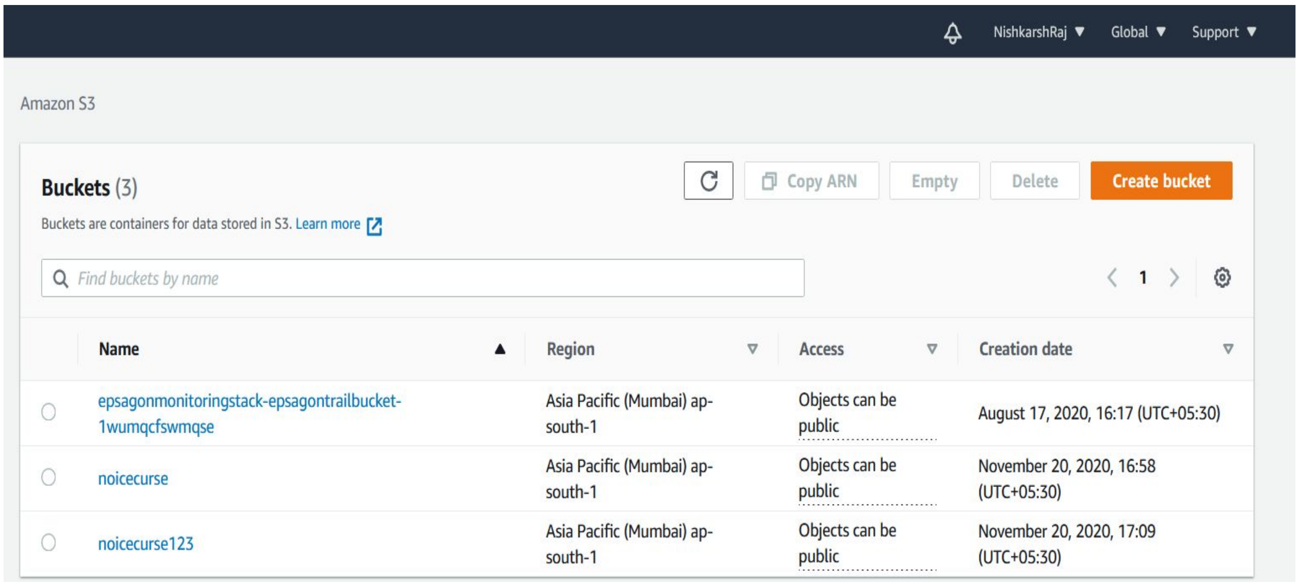
VPCs — Mumbai 2 — See all regions	NAT Gateways — Mumbai 0 — See all regions
Subnets — Mumbai 3 — See all regions	VPC Peering Connections — Mumbai 0 — See all regions
Route Tables — Mumbai 2 — See all regions	Network ACLs — Mumbai 2 — See all regions
Internet Gateways — Mumbai 1 — See all regions	Security Groups — Mumbai 2 — See all regions
Egress-only Internet Gateways — Mumbai 0 — See all regions	Customer Gateways — Mumbai 2 — See all regions
DHCP options sets — Mumbai 1 — See all regions	Virtual Private Gateways — Mumbai 3 — See all regions
Elastic IPs — Mumbai 0 — See all regions	Site-to-Site VPN Connections — Mumbai 2 — See all regions
Endpoints — Mumbai 0 — See all regions	Running Instances — Mumbai 2 — See all regions

- Verify creation of VPN



Name	VPN ID	State	Virtual Private Gateway	Transit Gateway	Customer Gateway
	vpn-0205eb3e4f541f950	available	vgw-02939db783e419788	-	cgw-091718dd82b6e55fd
	vpn-0679c92b539a80653	deleted	vgw-067a6ff410039e699	-	cgw-089c7db78a1e4415d

- Verify creation of S3 Bucket

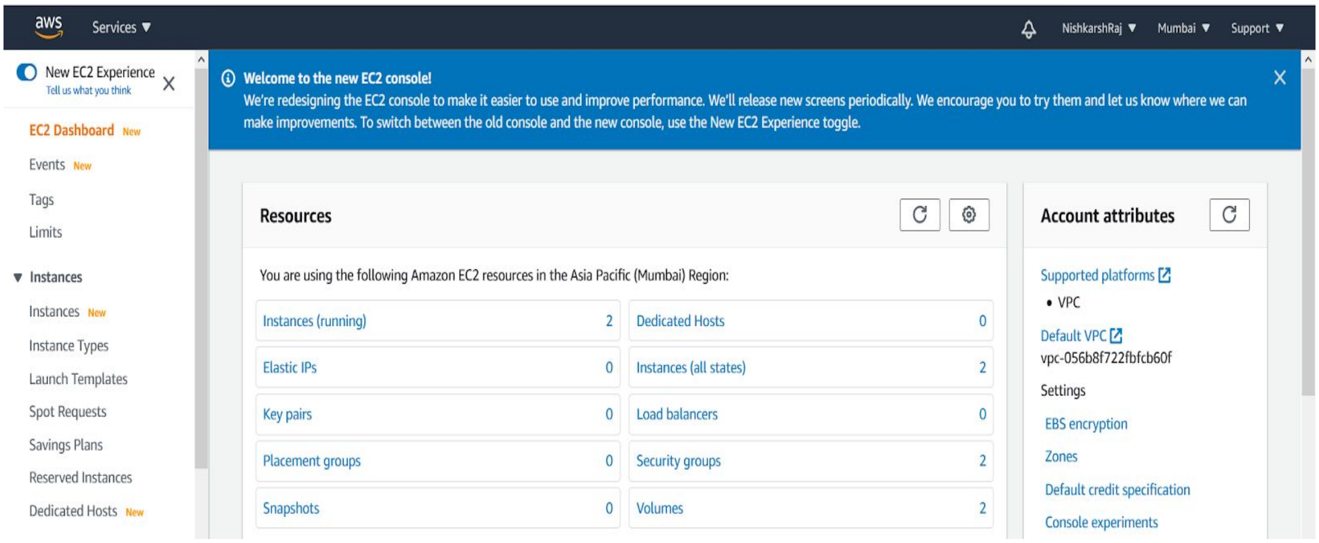


Buckets (3)

Buckets are containers for data stored in S3. Learn more

Name	Region	Access	Creation date
epsagonmonitoringstack-epsagontrailbucket-1wumqcfswmqse	Asia Pacific (Mumbai) ap-south-1	Objects can be public	August 17, 2020, 16:17 (UTC+05:30)
noicecurse	Asia Pacific (Mumbai) ap-south-1	Objects can be public	November 20, 2020, 16:58 (UTC+05:30)
noicecurse123	Asia Pacific (Mumbai) ap-south-1	Objects can be public	November 20, 2020, 17:09 (UTC+05:30)

- Verity creation of two EC2 instances



Welcome to the new EC2 console!
We're redesigning the EC2 console to make it easier to use and improve performance. We'll release new screens periodically. We encourage you to try them and let us know where we can make improvements. To switch between the old console and the new console, use the New EC2 Experience toggle.

Resources

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

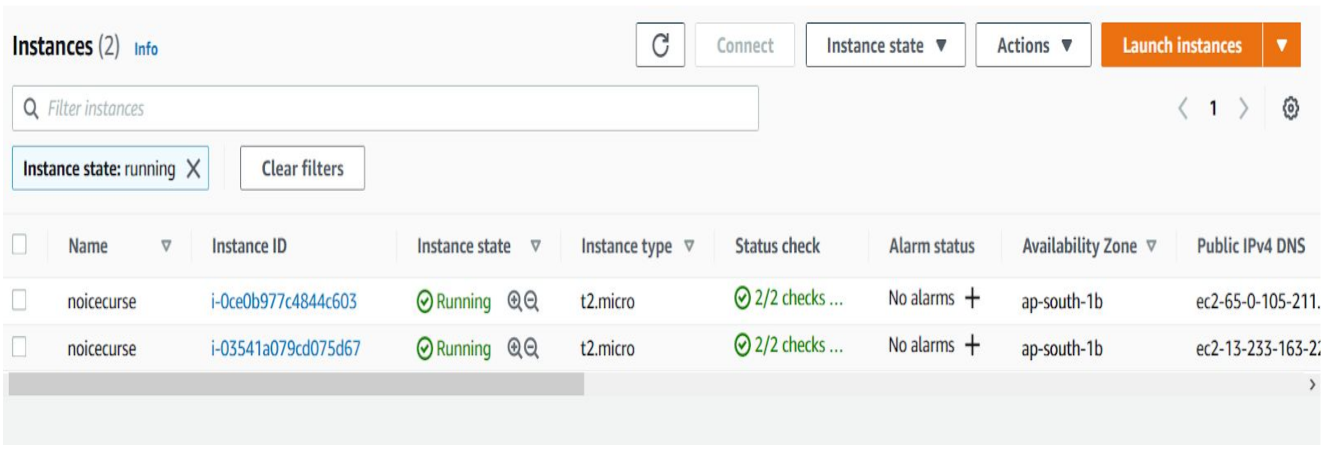
Instances (running)	2	Dedicated Hosts	0
Elastic IPs	0	Instances (all states)	2
Key pairs	0	Load balancers	0
Placement groups	0	Security groups	2
Snapshots	0	Volumes	2

Account attributes

Supported platforms
- VPC

Default VPC
vpc-056b8f722fbfcb60f

Settings
EBS encryption
Zones
Default credit specification
Console experiments



Instances (2) Info

Instance state: running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
	noicecurse	i-0ce0b977c4844c603	Running	t2.micro	2/2 checks ...	No alarms	ap-south-1b	ec2-65-0-105-211.
	noicecurse	i-03541a079cd075d67	Running	t2.micro	2/2 checks ...	No alarms	ap-south-1b	ec2-13-233-163-2: