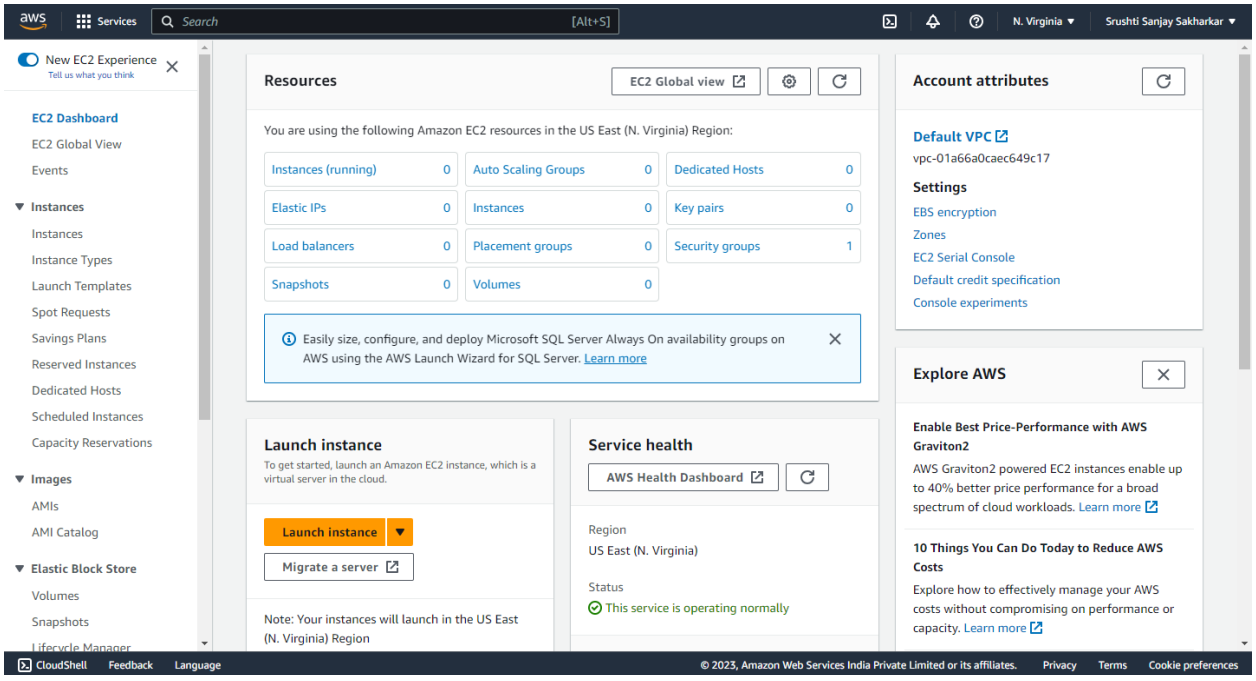
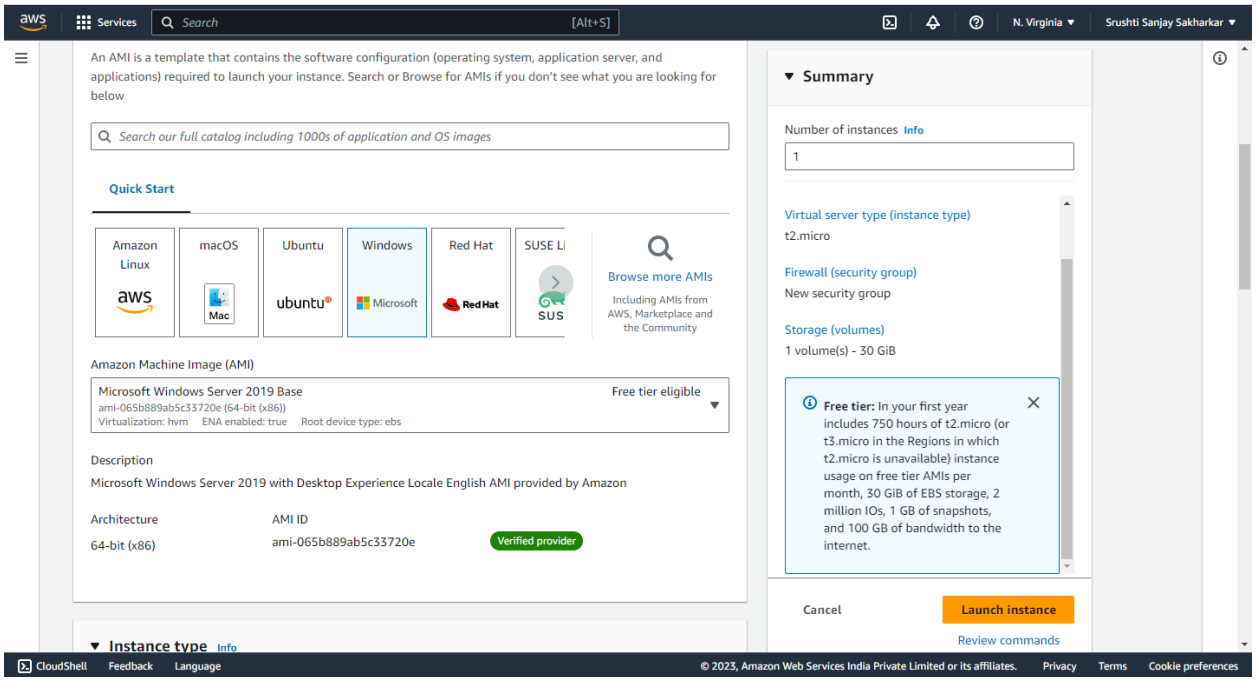
7. Laboratory Exercise

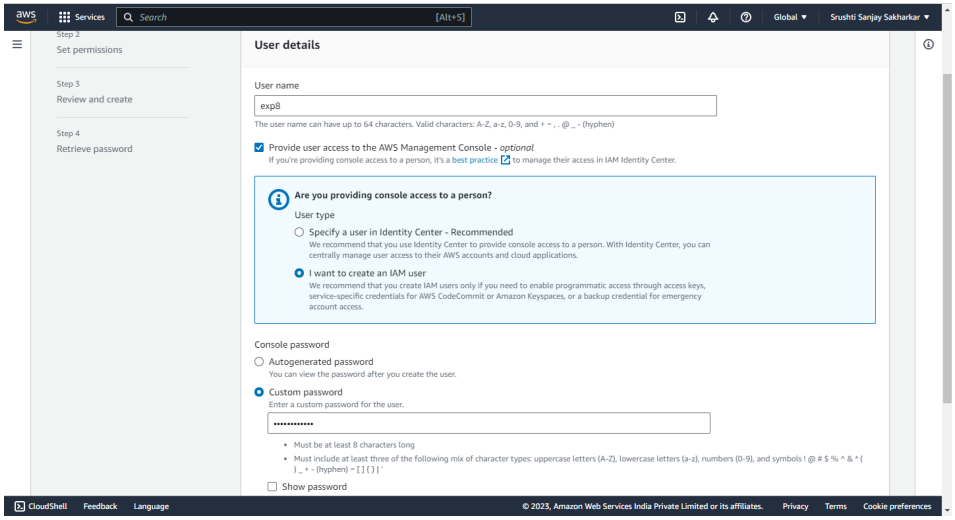
Step 1: First we will check that no instance is running on EC2.

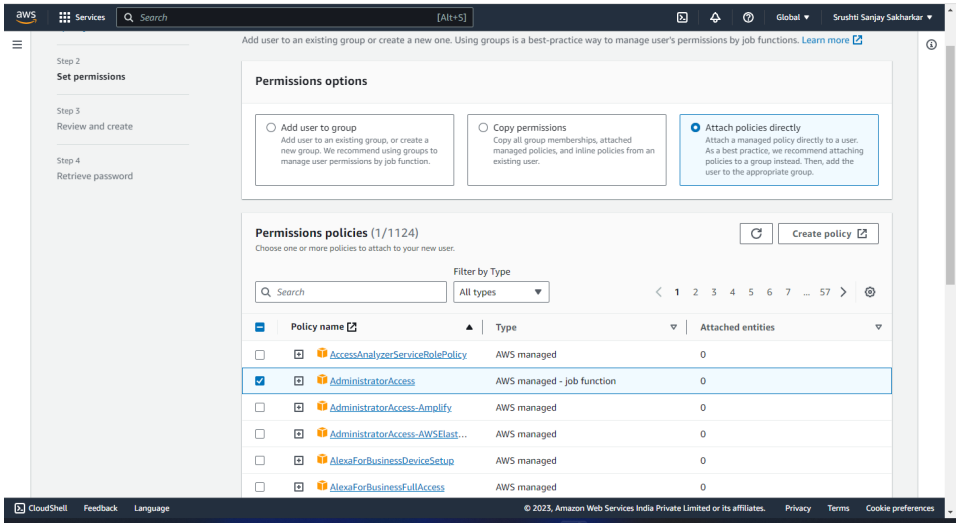


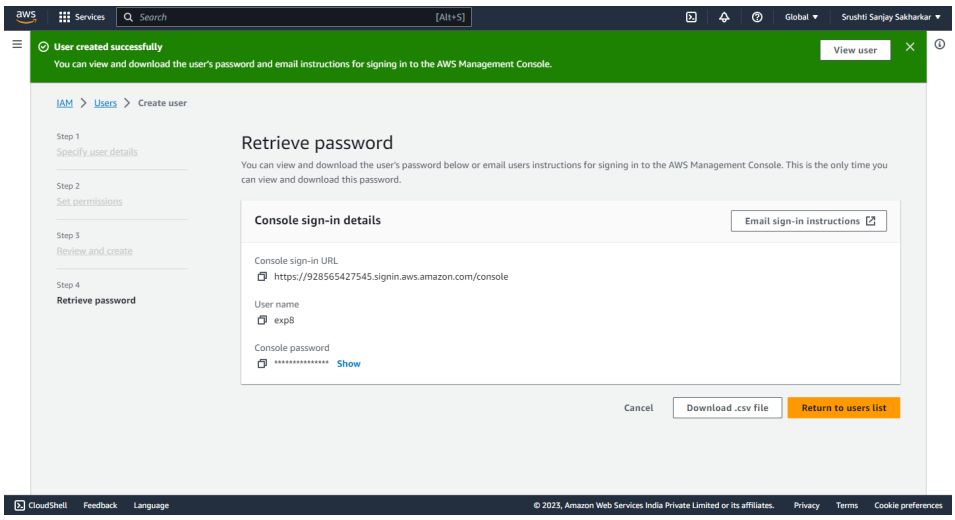
Step 2: Create an IAM user with Programmatic Password, Administrator access and download access key and secret key from download.csv



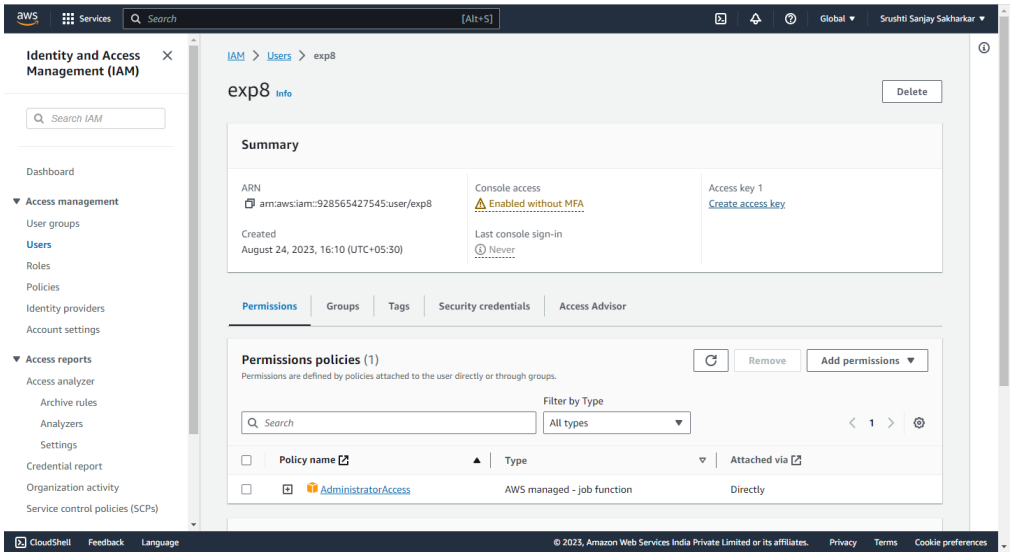
Srushti Sanjay Sakharkar TEITB ADL exp. 8 Roll No. 38

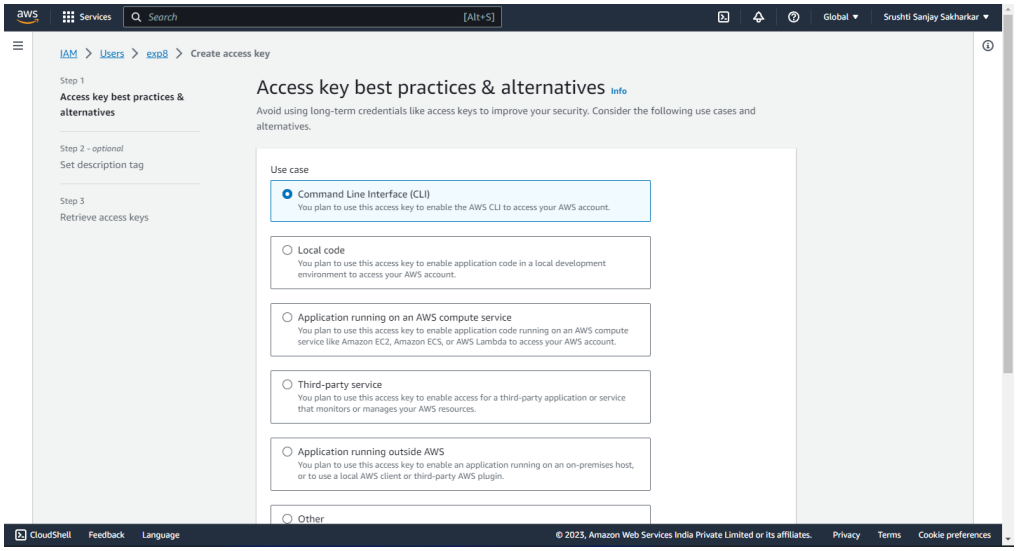


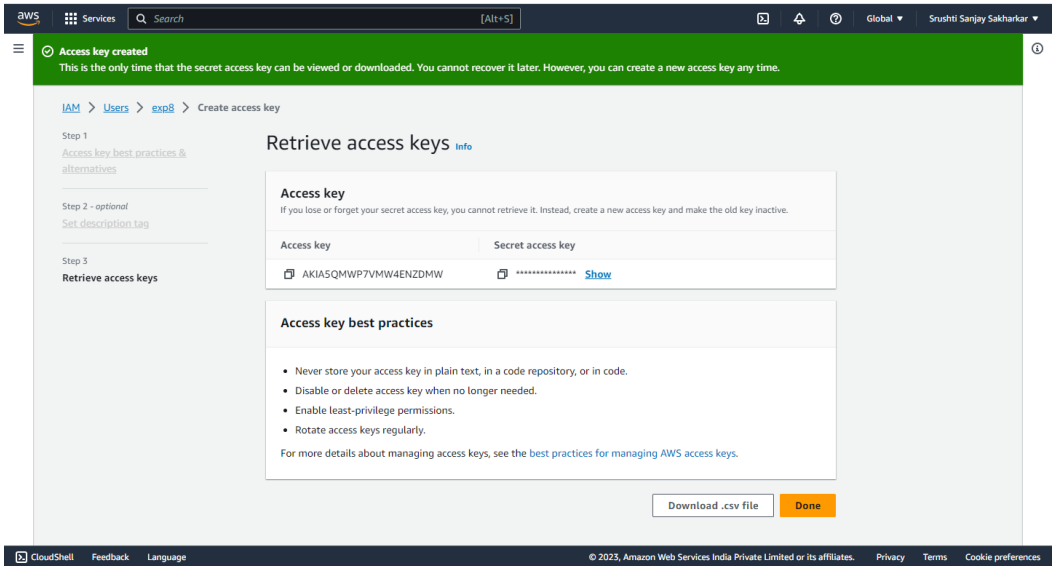




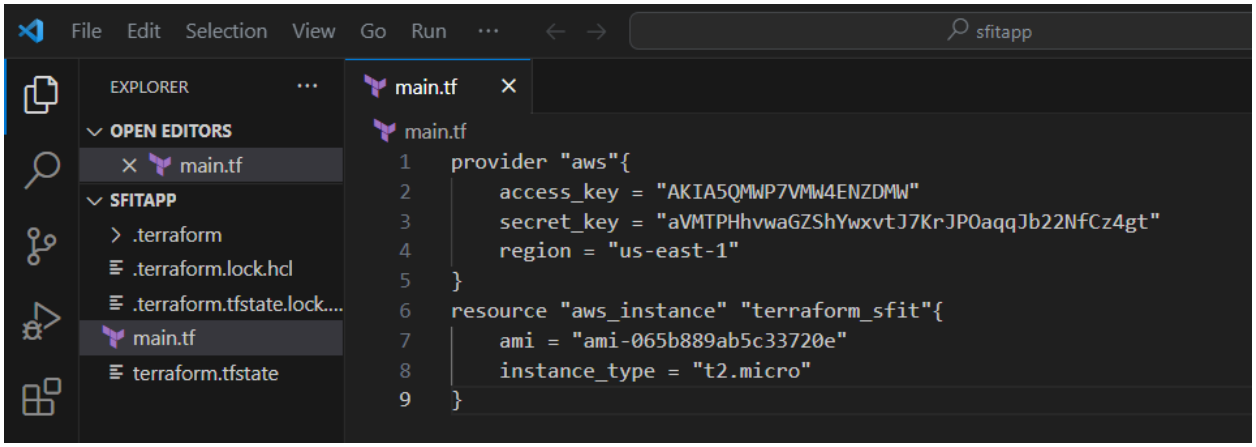
Srushti Sanjay Sakharkar TEITB ADL exp. 8 Roll No. 38



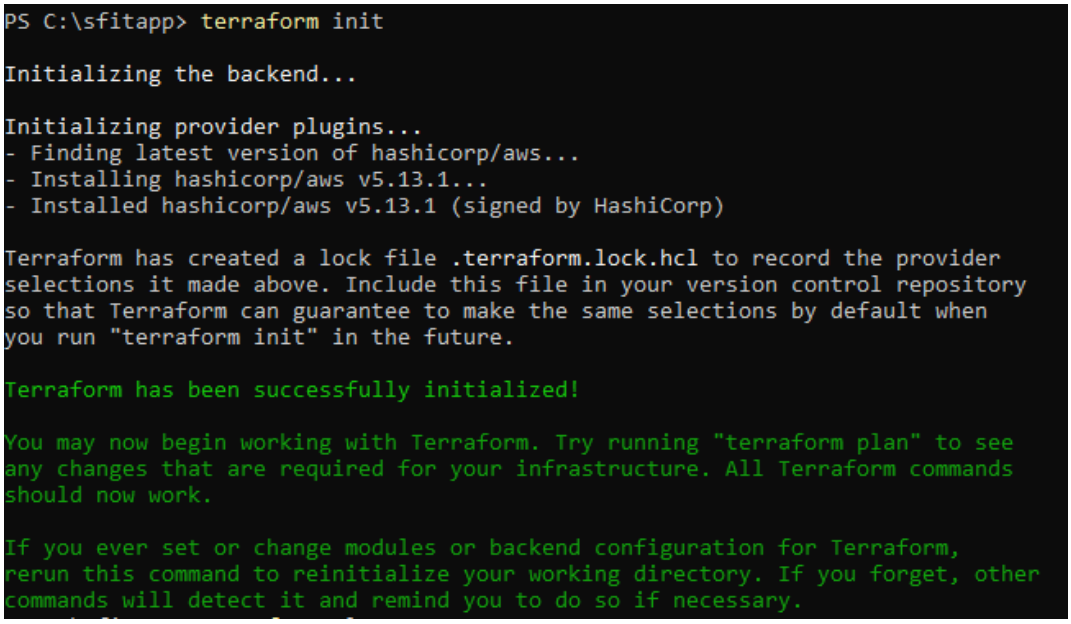




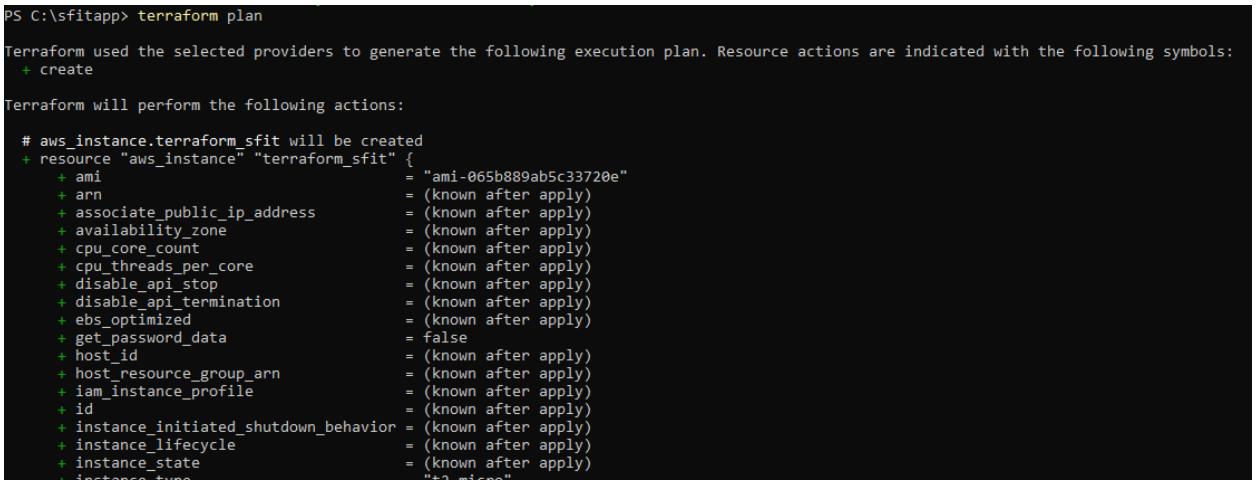
Step 3: Now write a Terraform program in vs code, create new file with .tf extension



Step 4: Now initialize the terraform ...type c:\SfitApp> terraform init

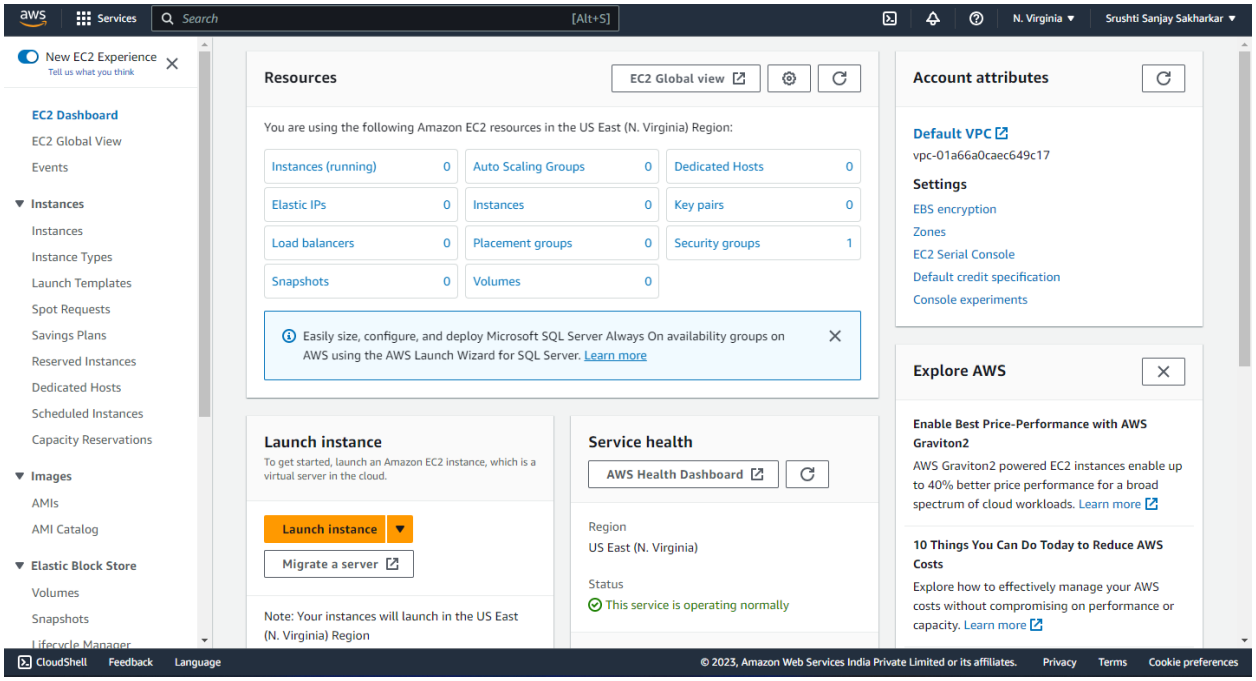


Step 5: c:\sfitApp>terraform plan



```
      + subnet_id                                  = (known after apply)
      + tags_all                                   = (known after apply)
      + tenancy                                    = (known after apply)
      + user_data                                  = (known after apply)
      + user_data_base64                           = (known after apply)
      + user_data_replace_on_change                = false
      + vpc_security_group_ids                     = (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.
```

Step 6: Check the instance on Ec2 before terraform apply

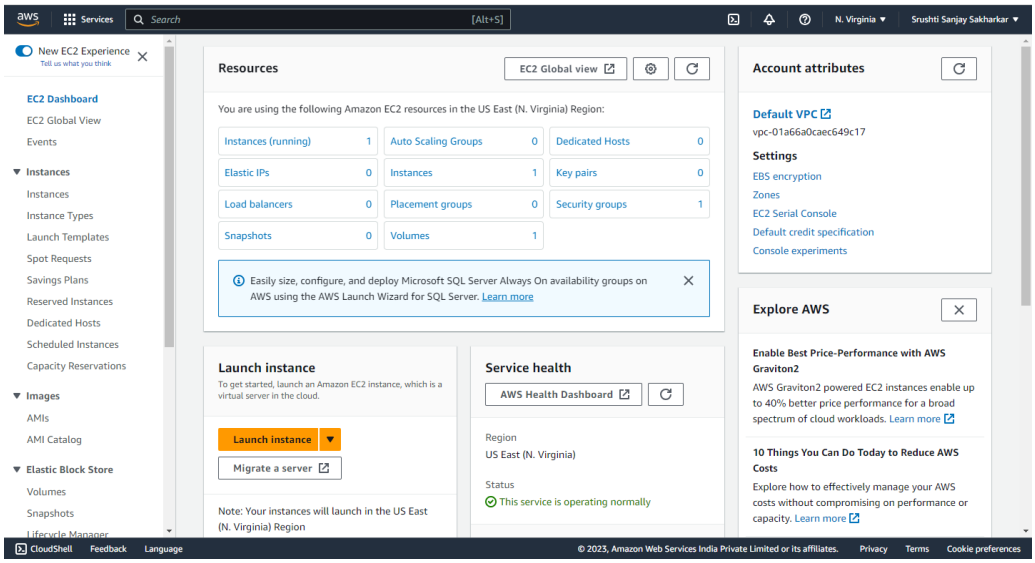


Step 7: Terraform apply

```
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
PS C:\sfitapp> terraform apply

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_instance.terraform_sfit will be created
  + resource "aws_instance" "terraform_sfit" {
      + ami                                  = "ami-065b889ab5c33720e"
      + 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
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
```

```
Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.terraform_sfit: Creating...
aws_instance.terraform_sfit: Still creating... [10s elapsed]
aws_instance.terraform_sfit: Still creating... [20s elapsed]
aws_instance.terraform_sfit: Still creating... [30s elapsed]
aws_instance.terraform_sfit: Creation complete after 36s [id=i-04f61b42162d119c1]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

Step 8: Check terraform created instances on EC2...we have created 3 instances.



Step 9: Now destroy the instance from command prompt....c:\SfitApp> terraform destroy

```
PS C:\sfitapp> terraform destroy
aws_instance.terraform_sfit: Refreshing state... [id=i-04f61b42162d119c1]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  - destroy

Terraform will perform the following actions:

  # aws_instance.terraform_sfit will be destroyed
  - resource "aws_instance" "terraform_sfit" {
      - ami                                  = "ami-065b889ab5c33720e" -> null
      - arn                                  = "arn:aws:ec2:us-east-1:928565427545:instance/i-04f61b42162d119c1" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "us-east-1d" -> null
      - cpu_core_count                       = 1 -> null
      - cpu_threads_per_core                 = 1 -> null
```

```
Plan: 0 to add, 0 to change, 1 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_instance.terraform_sfit: Destroying... [id=i-04f61b42162d119c1]
aws_instance.terraform_sfit: Still destroying... [id=i-04f61b42162d119c1, 10s elapsed]
aws_instance.terraform_sfit: Still destroying... [id=i-04f61b42162d119c1, 20s elapsed]
aws_instance.terraform_sfit: Still destroying... [id=i-04f61b42162d119c1, 30s elapsed]
aws_instance.terraform_sfit: Destruction complete after 33s

Destroy complete! Resources: 1 destroyed.
```