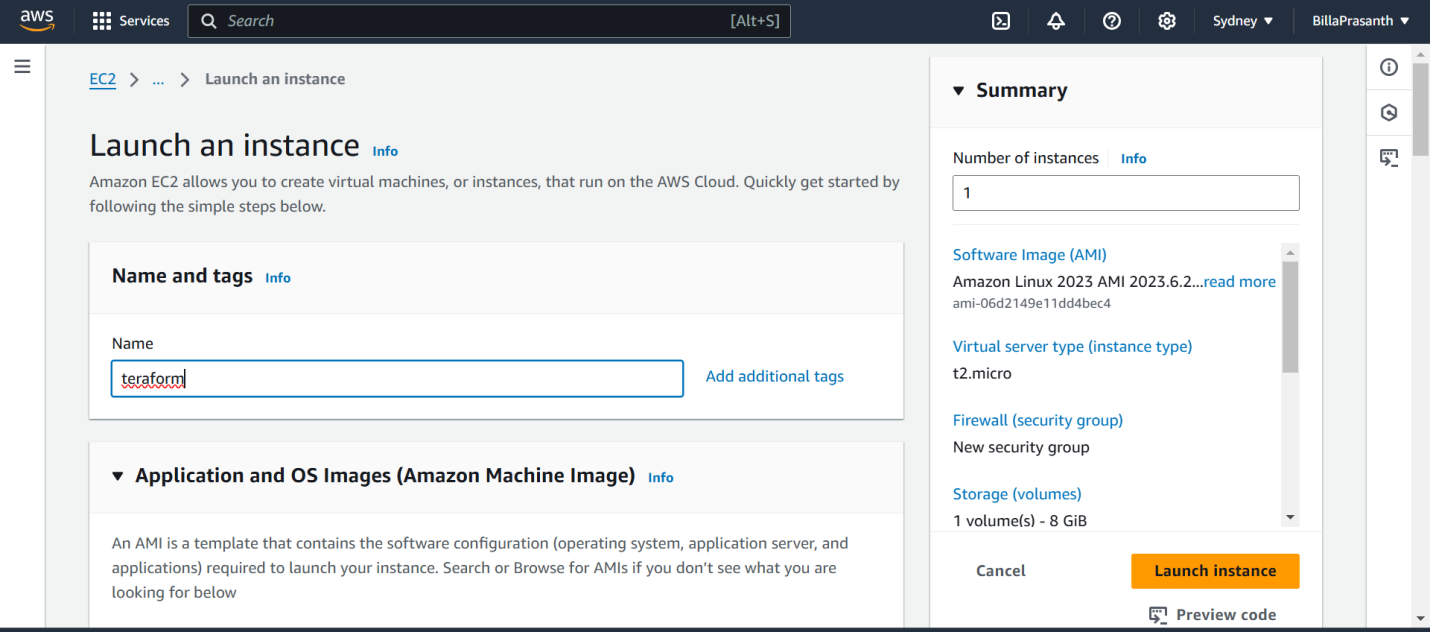
­­­­­­­­­­­­­CONNECTING TWO VPC’S USING PEERING BY TERRAFORM

**TERRAFORM**

Terraform is a popular infrastructure as code tool that allows you to automate the provisioning and managing of infrastructure resources in cloud.

First we need create one instance. Using that instance communicate with cloud by installing AWSCLI.

So first create ec2 instance.



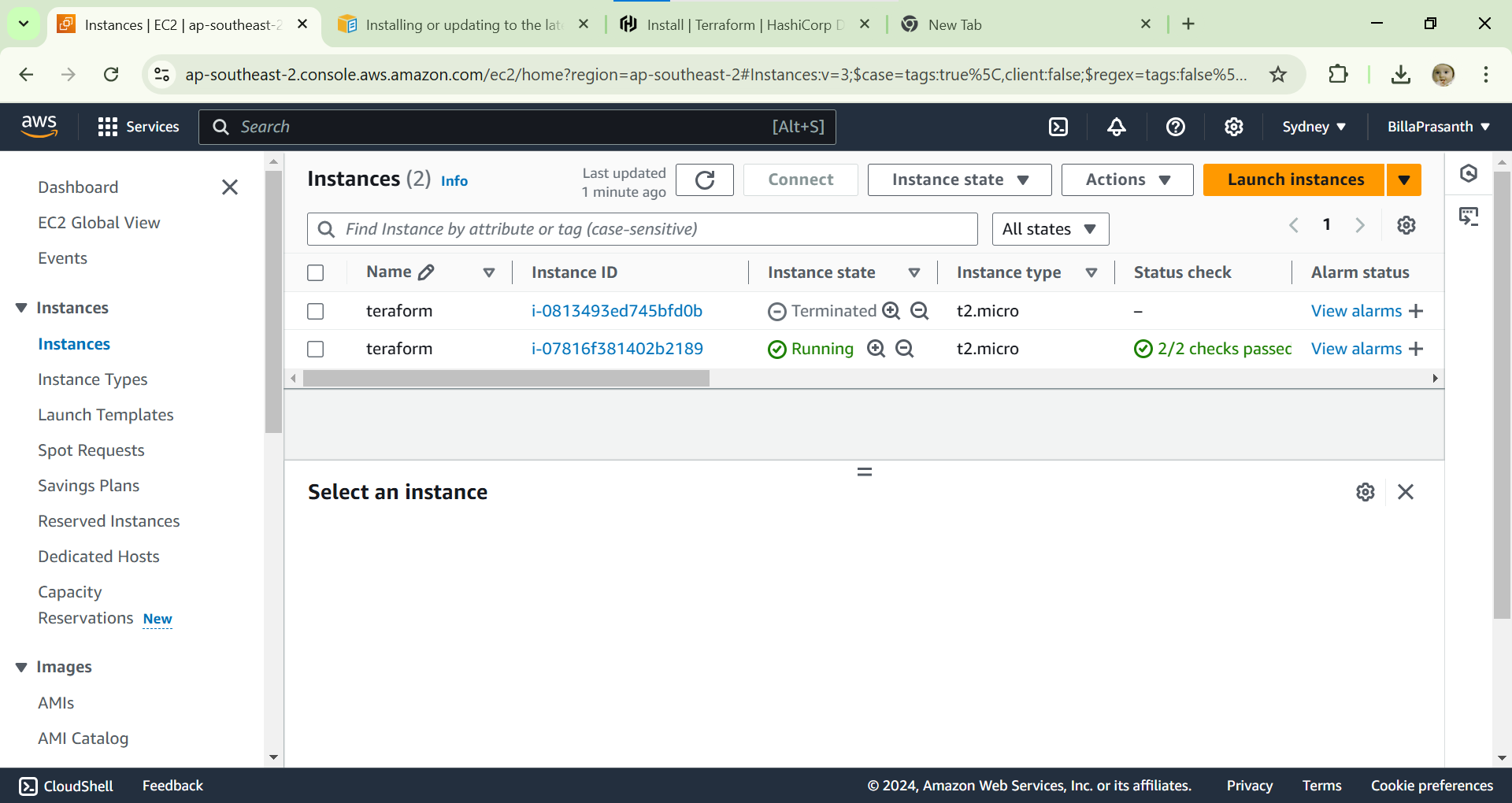
Give the instance name.

Select the AMI id.

Select the instance type.

Select the key pair.

Click on launch instance.



Instance is created.

Connect to ssh.

After connecting to ssh install unzip and awscli.

By using this url.

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

unzip awscliv2.zip

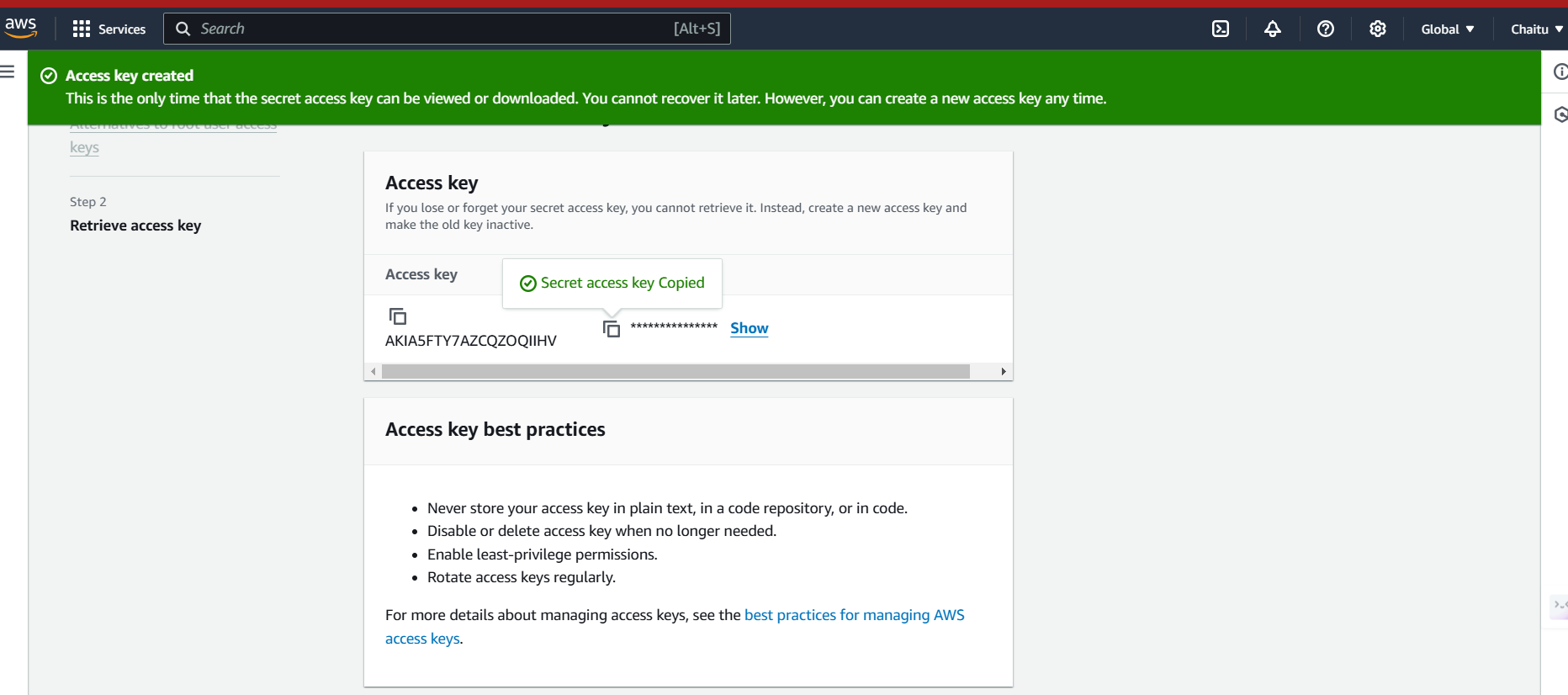
sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-cli –update

after installing aws cli

install terraform cli.

After installing terraform cli check it is installed are not.

By terraform – version.



Create access key for access to aws.

Enter the access key in server.

Create directory with name terraform.

Go to terraform folder.

Create terraformblock.tf file which contains terraform version.

Filename.tf which refers to it is a terraform file.

After creating terraformblock file paste the terraform version code.

terraform {

required\_providers {

aws = {

source = "hashicorp/aws"

version = "5.65.0"

}

}

}

Create provide.tf file which contains in which cloud you want to create infrastructure as code.

provider "aws" {

region = "ap-southeast-2"

profile = "default"

}

Create resource.tf it contains which resources is creating in cloud.

resource"aws\_vpc""vpc1" {

  cidr\_block="10.0.0.0/16"

  enable\_dns\_support=true

  enable\_dns\_hostnames=true

  tags ={

    Name = "vpc1"

  }

}

# Define VPC 2

resource"aws\_vpc""vpc2" {

  cidr\_block="10.1.0.0/16"

  enable\_dns\_support=true

  enable\_dns\_hostnames=true

  tags ={

    Name = "vpc2"

  }

}

# Define the VPC Peering Connection

resource"aws\_vpc\_peering\_connection""peer" {

  vpc\_id        = aws\_vpc.vpc1.id

  peer\_vpc\_id    = aws\_vpc.vpc2.id

  auto\_accept    =false

  tags ={

    Name = "vpc1-to-vpc2"

  }

}

# Define VPC Peering Connection Accepter

resource"aws\_vpc\_peering\_connection\_accepter""accept" {

  vpc\_peering\_connection\_id= aws\_vpc\_peering\_connection.peer.id

  auto\_accept               =true

  tags ={

    Name = "vpc2-to-vpc1"

  }

}

# Add route tables for VPC 1

resource"aws\_route\_table""vpc1\_route\_table" {

  vpc\_id= aws\_vpc.vpc1.id

  route {

    cidr\_block                ="10.1.0.0/16"

    vpc\_peering\_connection\_id= aws\_vpc\_peering\_connection.peer.id

  }

  tags ={

    Name = "vpc1-route-table"

  }

}

resource"aws\_route\_table\_association""vpc1\_association" {

  subnet\_id      = aws\_subnet.subnet1.id

  route\_table\_id= aws\_route\_table.vpc1\_route\_table.id

}

# Add route tables for VPC 2

resource"aws\_route\_table""vpc2\_route\_table" {

  vpc\_id= aws\_vpc.vpc2.id

  route {

    cidr\_block                ="10.0.0.0/16"

    vpc\_peering\_connection\_id= aws\_vpc\_peering\_connection.peer.id

  }

  tags ={

    Name = "vpc2-route-table"

  }

}

resource"aws\_route\_table\_association""vpc2\_association" {

  subnet\_id      = aws\_subnet.subnet2.id

  route\_table\_id= aws\_route\_table.vpc2\_route\_table.id

}

# Optionally, create subnets

resource"aws\_subnet""subnet1" {

  vpc\_id            = aws\_vpc.vpc1.id

  cidr\_block        ="10.0.1.0/24"

  availability\_zone="ap-southeast-2a"

  tags ={

    Name = "subnet1"

  }

}

resource"aws\_subnet""subnet2" {

  vpc\_id            = aws\_vpc.vpc2.id

  cidr\_block        ="10.1.1.0/24"

  availability\_zone="ap-southeast-2b"

  tags ={

    Name = "subnet2"

  }

}

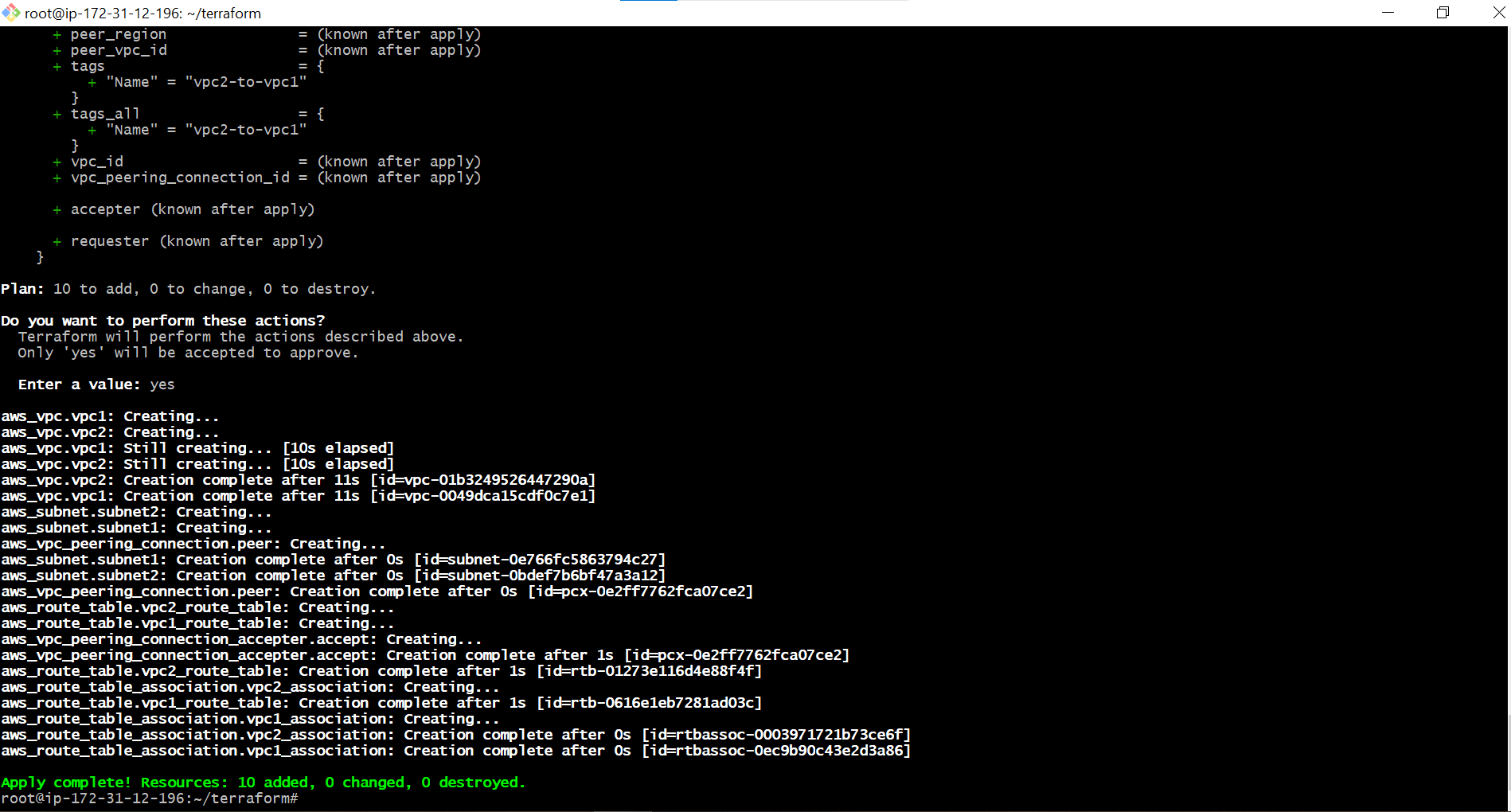
After creating run the terraform files by using.

Terraform init.

Terraform validate.

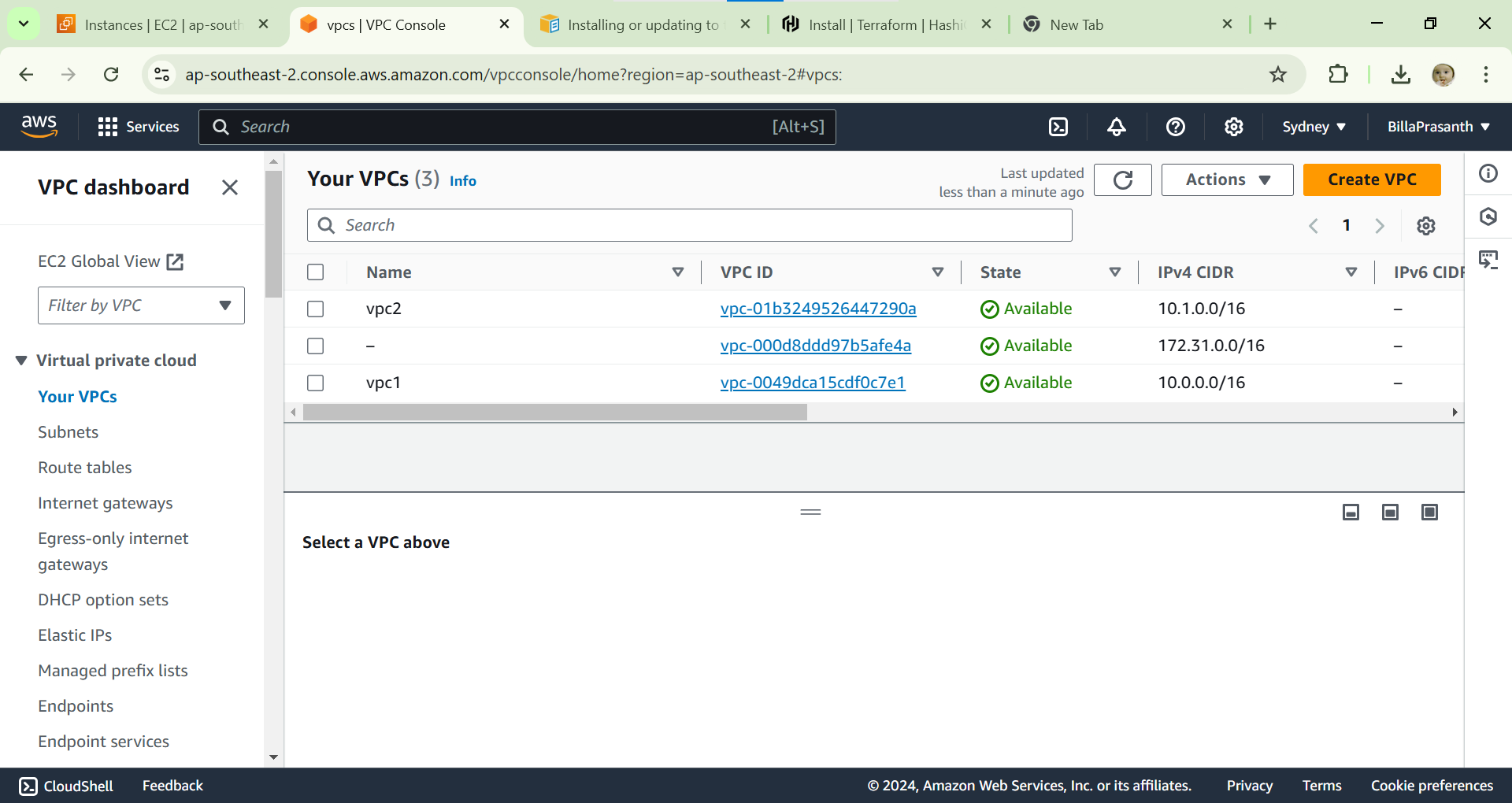
Terraform plan.

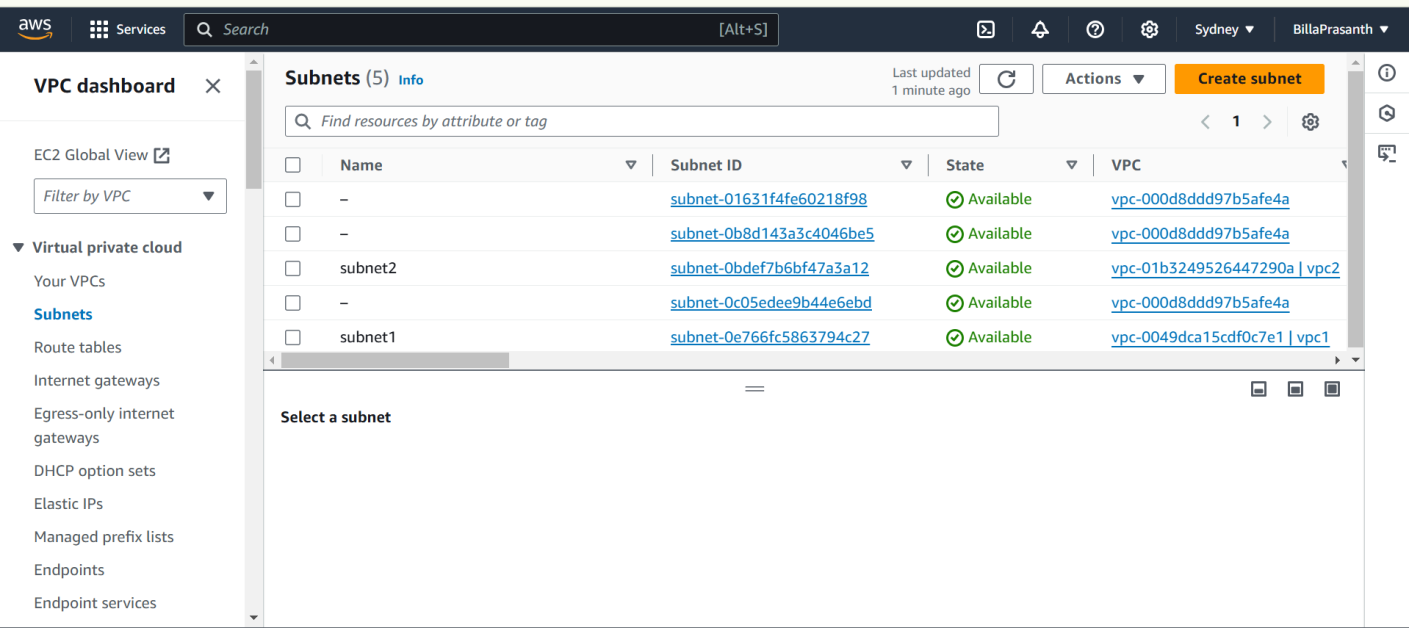
Terraform apply.



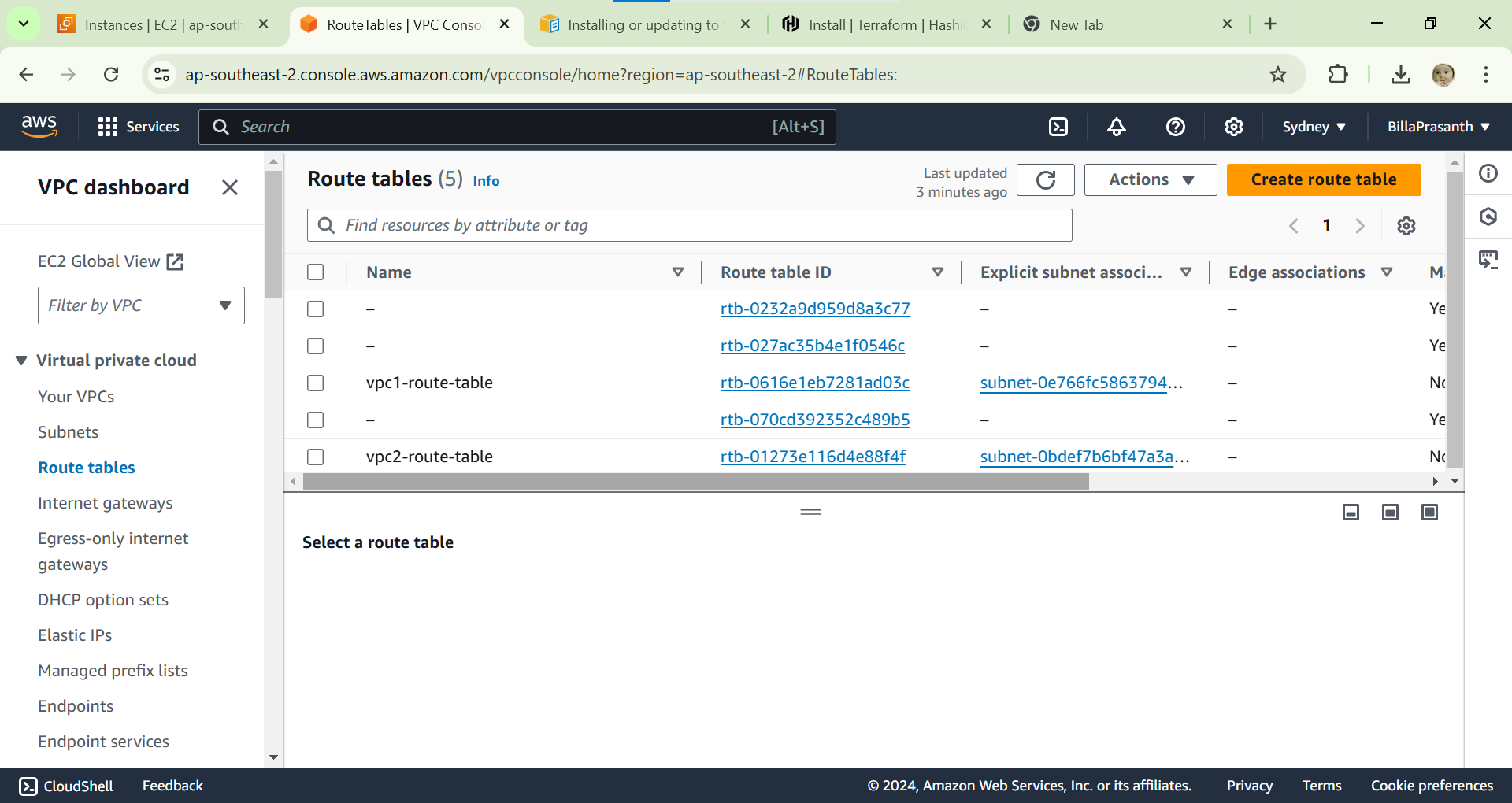
It is successfully created.

Go and check in the cloud if it created are not.

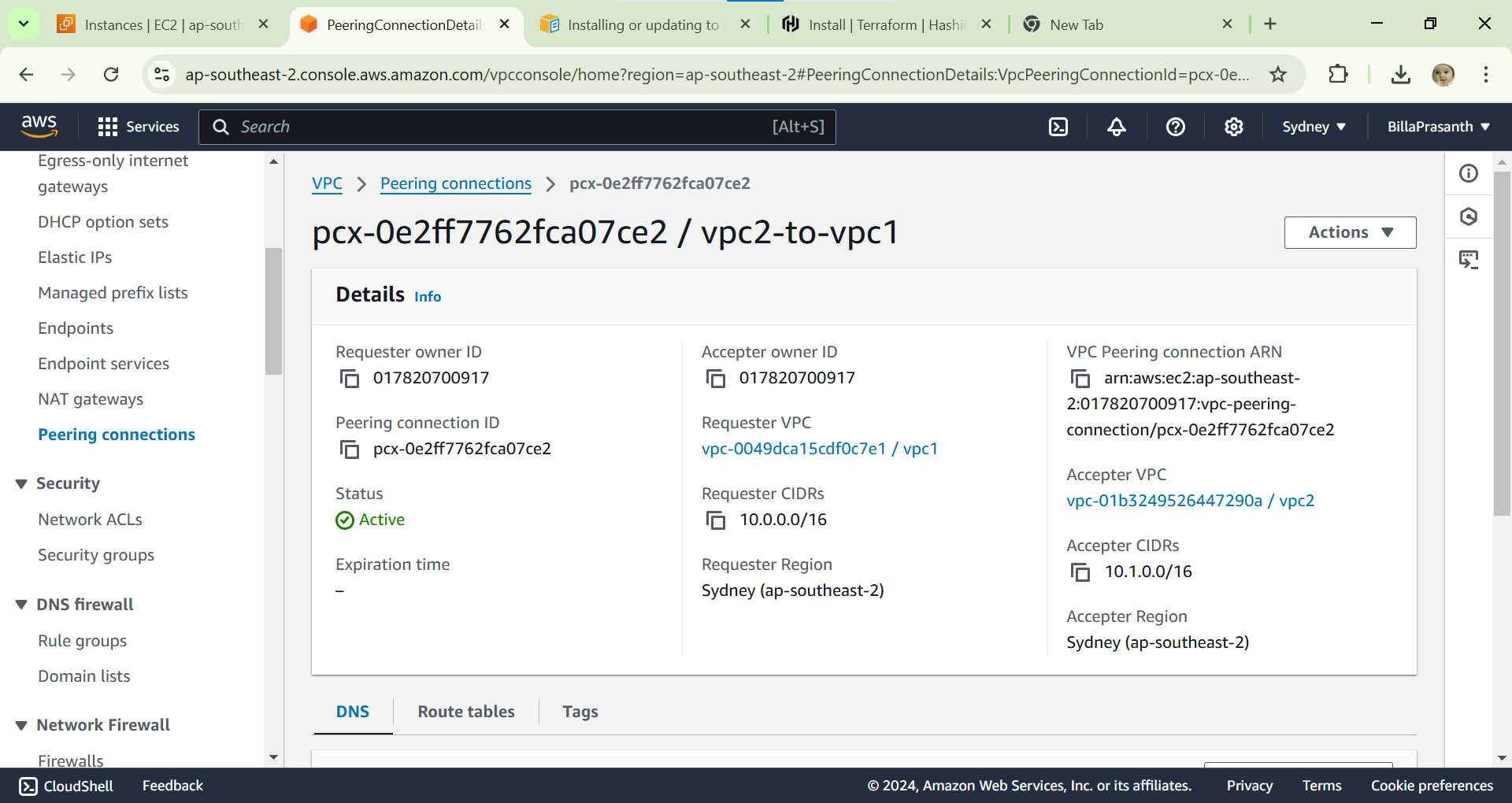
vpc’s is created



Subnets also created.



In route table peering is routed.



Peering also created.

For deleting the every thing by using

Terraform destroy.