Name: B V N Rahul
Reg no: RA2111028010192

Lab Activity-1

Create an EC2 Instance using Terraform:

Steps:

1.Install The terraform in your device and configure it.

2.Now Login to AWS console and install the AWS Command Line Interface (CLI) and configure it with your access key ID and secret access key. Create an IAM user with the necessary permissions.

3. Now Generate an SSH key pair for accessing Linux-based EC2 instances via SSH.

4.Create a Working directory where you will create terraform configuration files

5.Create a .tf file (e.g.: main.tf) in your working directory. Define your EC2 instance configuration using Terraform syntax. Specify details such as instance type, AMI, security groups, and key pair.

6.Initalize the directory by running the command "**terraform init**".

7.After initialization, run the command to create an execution plan "**terraform plan**".

8.Finally, apply the configuration to create the EC2 instance "**terraform apply**".

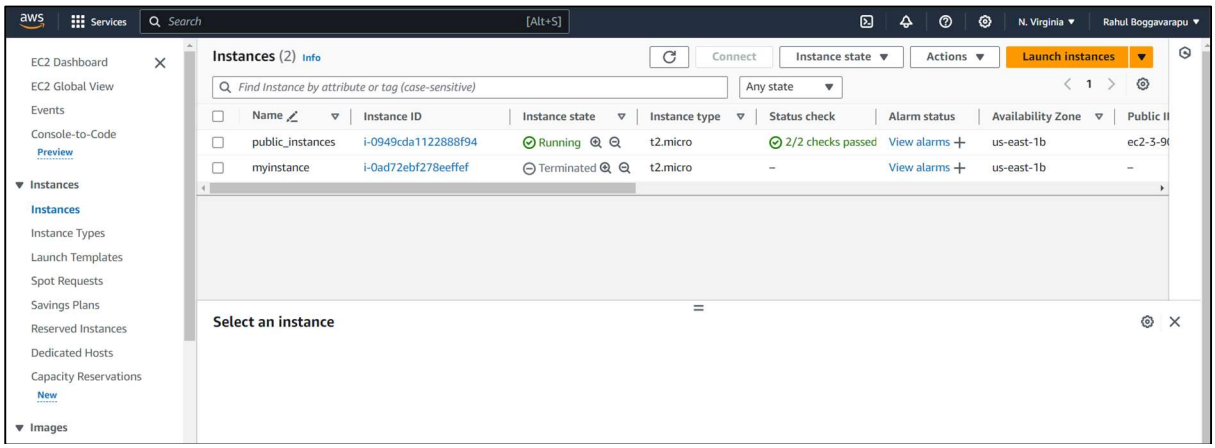
9.Now check the output in the AWS console after that you can change the inbound rules of the instance.

Code:

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "~> 4.0"
    }
  }
}
provider "aws" {
  region = "us-east-1"
  access_key = "AKIASQLWVLENQXCMZ6GL"
  secret_key = "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"
}
resource "tls_private_key" "rsa-4096" {
  algorithm = "RSA"
  rsa_bits = 4096
}
variable "key_name" {}
resource "aws_key_pair" "key_pair" {
  key_name   = " var.key_name "
  public_key = tls_private_key. rsa_4096.private_key_openssh
}
```

```
resource "local_file" "private_key"{
        content = tls_private_key.rsa_4096.private_key_pem
         filename = var.key_name
}
resource "aws_instance" "public_instances" {
  ami           = " ami-0440d3b780d96b29d "
  instance_type = "t2.micro"

  tags = {
    Name = "public_instances"
  }
}
```

Output:



Create an VPC using Terraform:

Steps:

1.Install The terraform in your device and configure it.

2.Create a file named main.tf and specify the AWS provider and the region where you want to create the VPC

```
provider "aws" {
region = "us-east-1"
}
```

3.In a file named main.tf define the VPC with a CIDR block (IP address range) and give it a name

```
resource "aws_vpc" "my_vpc" {
  cidr_block       = "10.0.0.0/16"
  instance_tenancy = "default"

  tags = {
    Name = "MyVPC"
  }
```

4.create a subnet within the VPC. Specify the VPC ID and the CIDR block for the subnet. You can also enable public IP assignment to instances launched in this subnet.

```
resource "aws_subnet" "my_subnet" {
  vpc_id            = aws_vpc.my_vpc.id
  cidr_block        = "10.0.1.0/24"
  availability_zone = "us-east-1"

  tags = {
    Name = "MySubnet"
  }
}
```

5. To allow communication between instances in your subnet and the internet, create an internet gateway

```
resource "aws_internet_gateway" "my_igw" {
  vpc_id = aws_vpc.my_vpc.id

  tags = {
    Name = "MyIGW"
  }
}
```

6. Associate the internet gateway with the VPC. In a file named main.tf

```
resource "aws_vpc_attachment" "igw_attach" {
  vpc_id             = aws_vpc.my_vpc.id
  internet_gateway_id = aws_internet_gateway.my_igw.id
}
```

7.Create a routing table for your subnet.

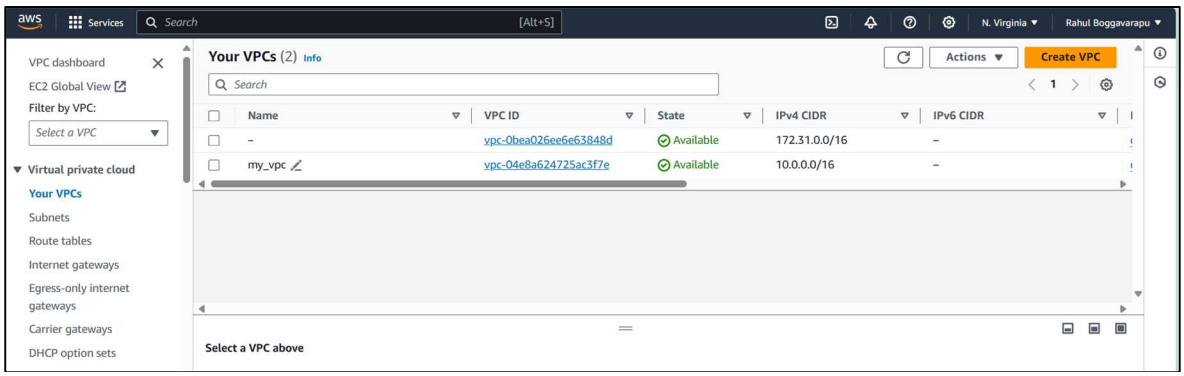
```
resource "aws_route_table" "my_route_table" {
  vpc_id = aws_vpc.my_vpc.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.my_igw.id
  }

  tags = {
    Name = "MyRouteTable"
  }
}
```

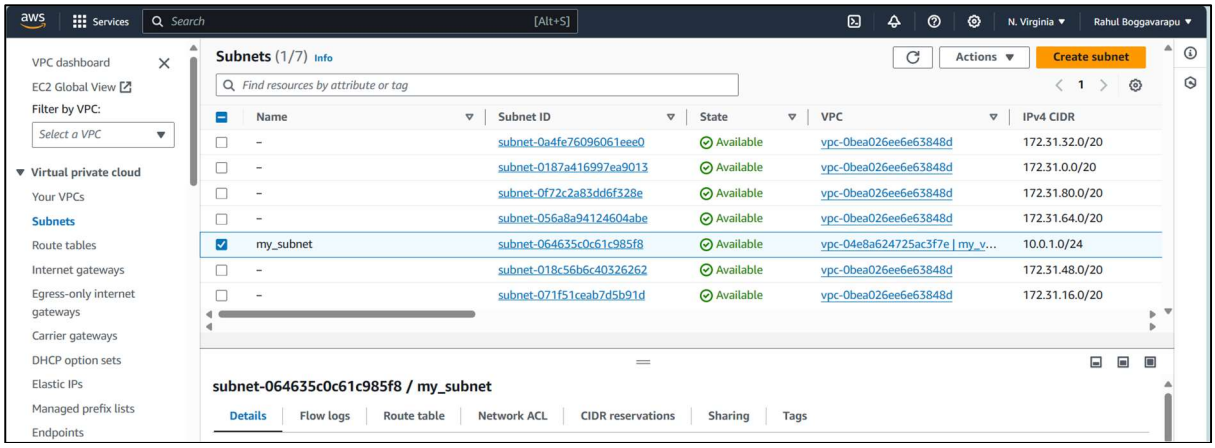
8. You can create security groups and network ACLs to control inbound and outbound traffic to your instances within the VPC

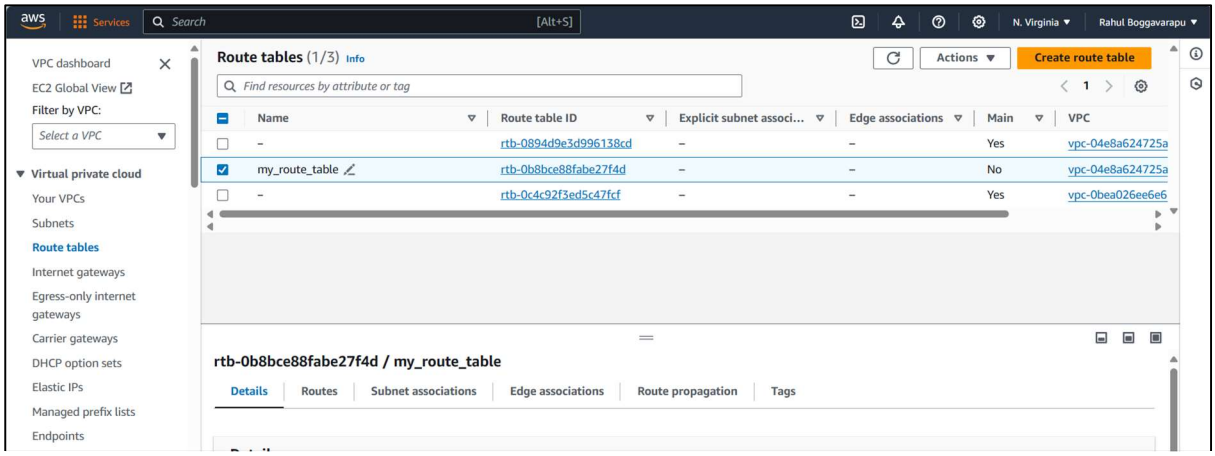
```
resource "aws_route_table_association" "my_route_table_association" {
  subnet_id      = aws_subnet.my_subnet.id
  route_table_id = aws_route_table.my_route_table.id
}

resource "aws_security_group" "My_vpc_sg" {
  name        = "My_vpc_sg"
  vpc_id      = aws_vpc.my_vpc.id

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
    ipv6_cidr_blocks = [“: :/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
    ipv6_cidr_blocks = [“: :/0"]
  }

  tags = {
    Name = "allow_tls"
  }
}
```

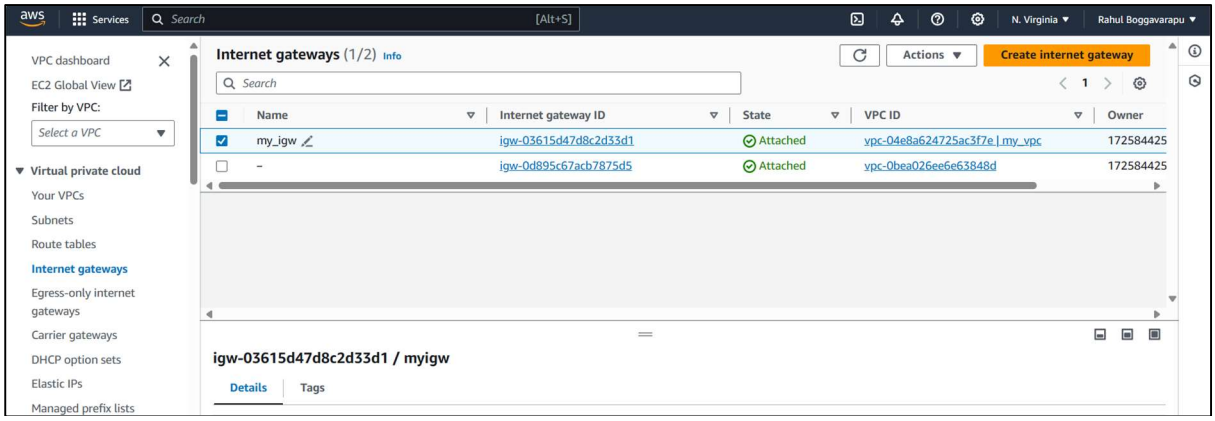
9. Apply Terraform Configuration:
   - Run “**terraform init”** to initialize your working directory.
   - Run “**terraform apply”** to create the VPC and associated resources.

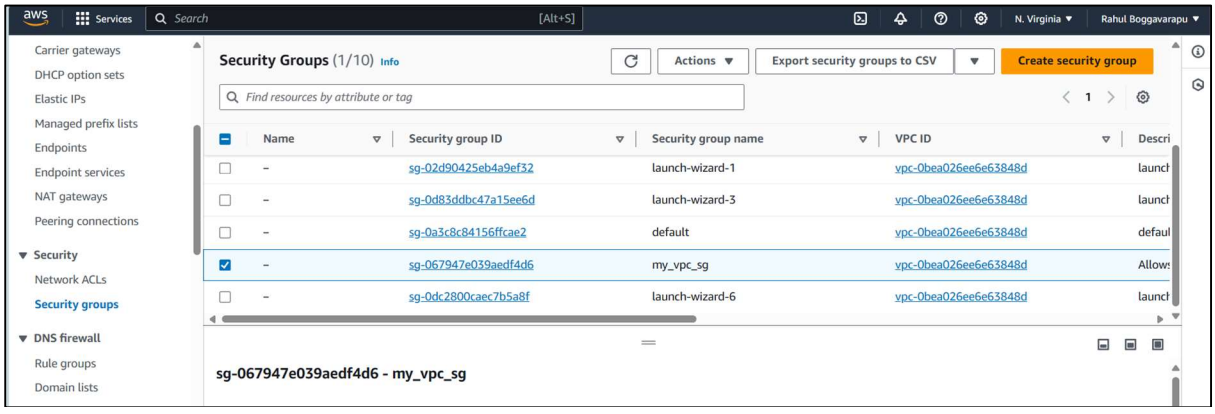
Output:




Subnets (1/7) Info
Actions
Create subnet
Find resources by attribute or tag
Name
Subnet ID
State
VPC
IPv4 CIDR
subnet-0a4fe76096061eee0 Available vpc-0bea026ee6e63848d 172.31.32.0/20
subnet-0187a416997ea9013 Available vpc-0bea026ee6e63848d 172.31.0.0/20
subnet-0f72c2a83dd6f328e Available vpc-0bea026ee6e63848d 172.31.80.0/20
subnet-056a8a94124604abe Available vpc-0bea026ee6e63848d 172.31.64.0/20
my_subnet subnet-064635c0c61c985f8 Available vpc-04e8a624725ac3f7e | my_v... 10.0.1.0/24
subnet-018c56b6c40326262 Available vpc-0bea026ee6e63848d 172.31.48.0/20
subnet-071f51ceab7d5b91d Available vpc-0bea026ee6e63848d 172.31.16.0/20
subnet-064635c0c61c985f8 / my_subnet
Details
Flow logs
Route table
Network ACL
CIDR reservations
Sharing
Tags


Route tables (1/3) Info
Actions
Create route table
Find resources by attribute or tag
Name
Route table ID
Explicit subnet associ...
Edge associations
Main
VPC
rtb-0894d9e3d996138cd Yes vpc-04e8a624725a
my_route_table rtb-0b8bce88fabe27f4d No vpc-04e8a624725a
rtb-0c4c92f3ed5c47fcf Yes vpc-0bea026ee6e6
rtb-0b8bce88fabe27f4d / my_route_table
Details
Routes
Subnet associations
Edge associations
Route propagation
Tags


Internet gateways (1/2) Info
Actions
Create internet gateway
Search
Name
Internet gateway ID
State
VPC ID
Owner
my_igw igw-03615d47d8c2d33d1 Attached vpc-04e8a624725ac3f7e | my_vpc 172584425
igw-0d895c67acb7875d5 Attached vpc-0bea026ee6e63848d 172584425
igw-03615d47d8c2d33d1 / myigw
Details
Tags


Security Groups (1/10) Info
Actions
Export security groups to CSV
Create security group
Find resources by attribute or tag
Name
Security group ID
Security group name
VPC ID
sg-02d90425eb4a9ef32 launch-wizard-1 vpc-0bea026ee6e63848d
sg-0d83ddbc47a15ee6d launch-wizard-3 vpc-0bea026ee6e63848d
sg-0a3c8c84156ffcae2 default vpc-0bea026ee6e63848d
sg-067947e039aedf4d6 my_vpc_sg vpc-0bea026ee6e63848d
sg-0dc2800caec7b5a8f launch-wizard-6 vpc-0bea026ee6e63848d
sg-067947e039aedf4d6 - my_vpc_sg