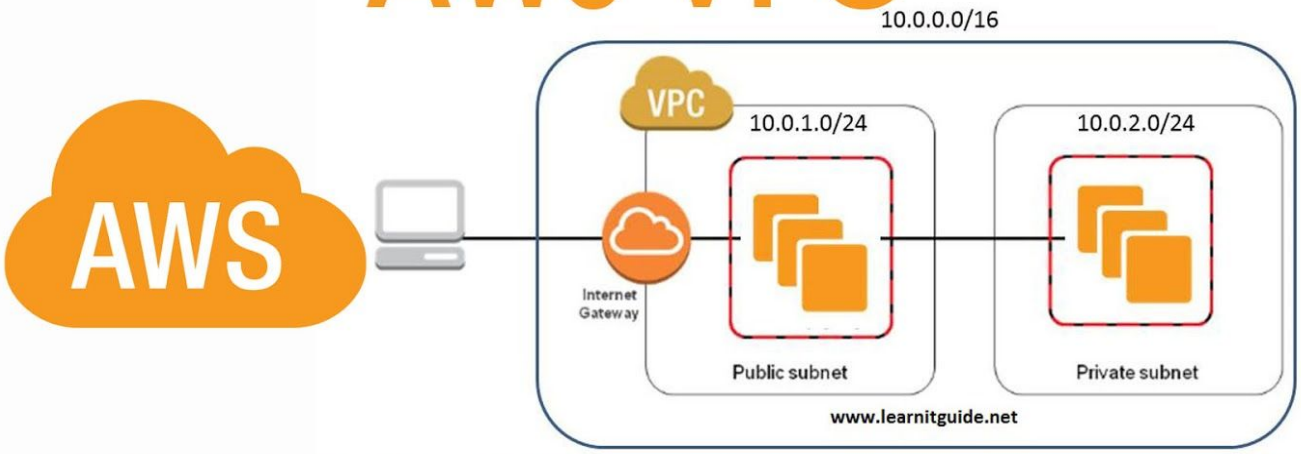
AWS VPC



HOW TO CREATE NEW VPC SUBNETS, INTERNET GATEWAY

Terraform

Amazon Machine Image (AMI)

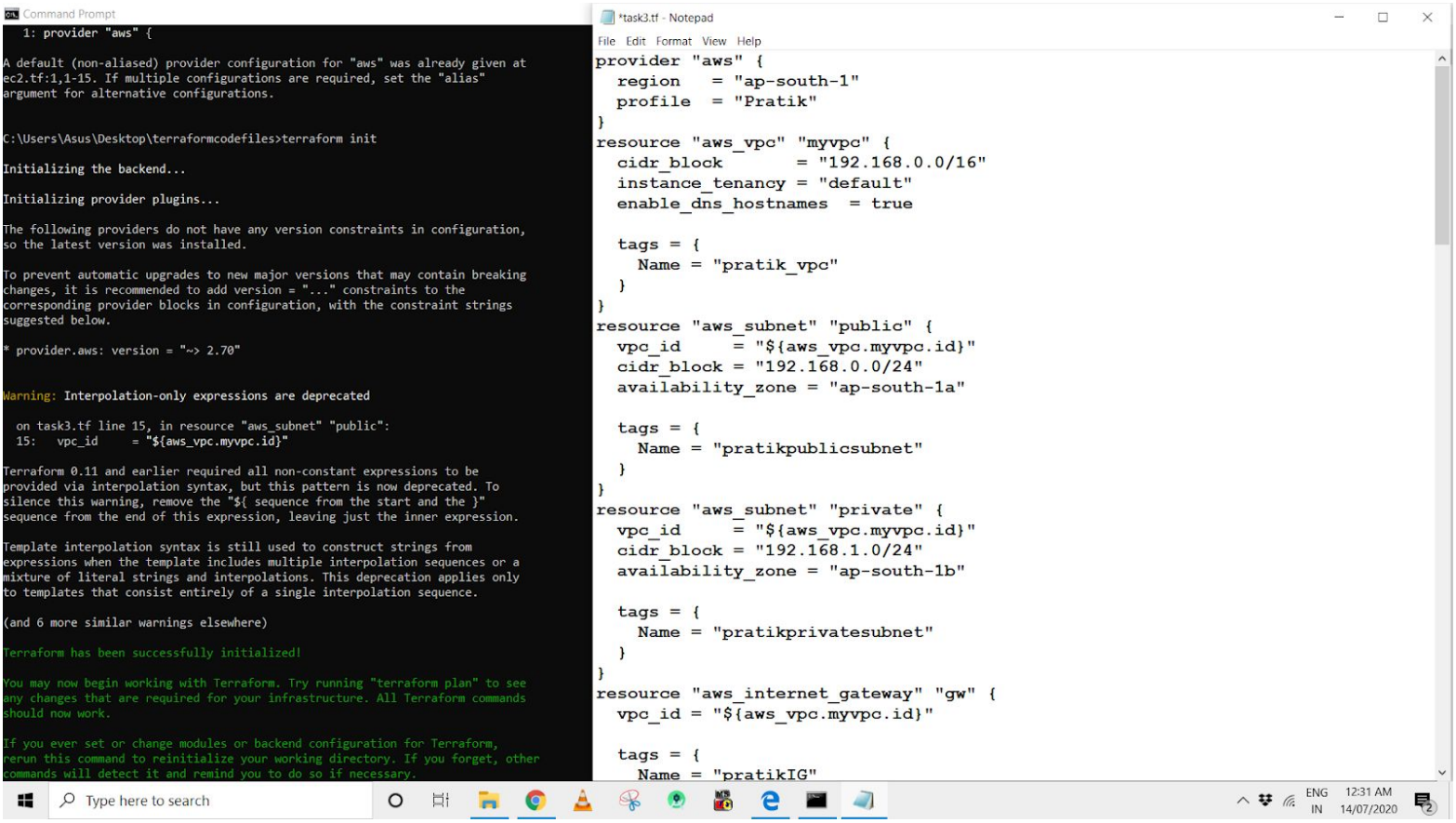
aws

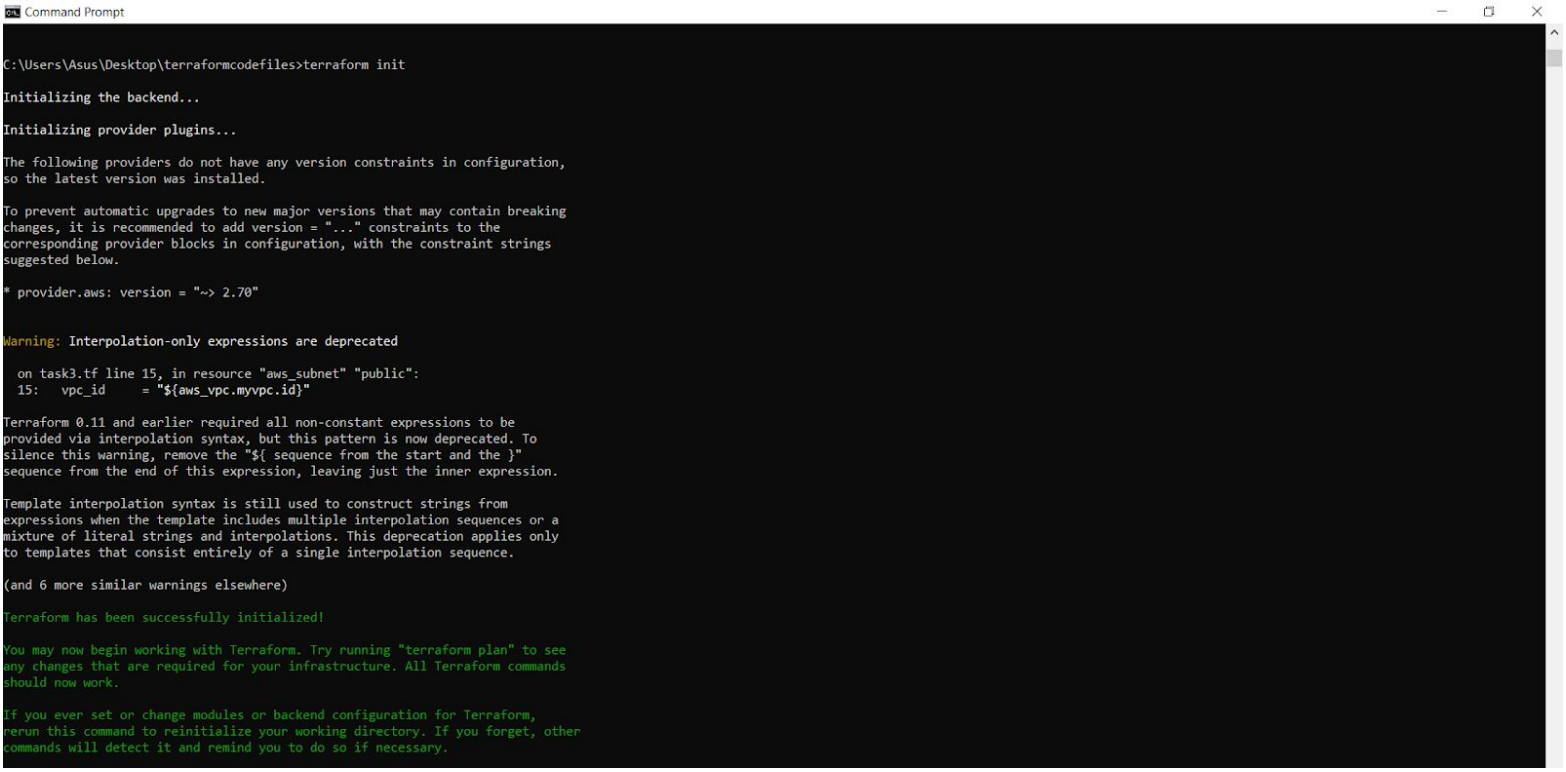
Amazon EC2

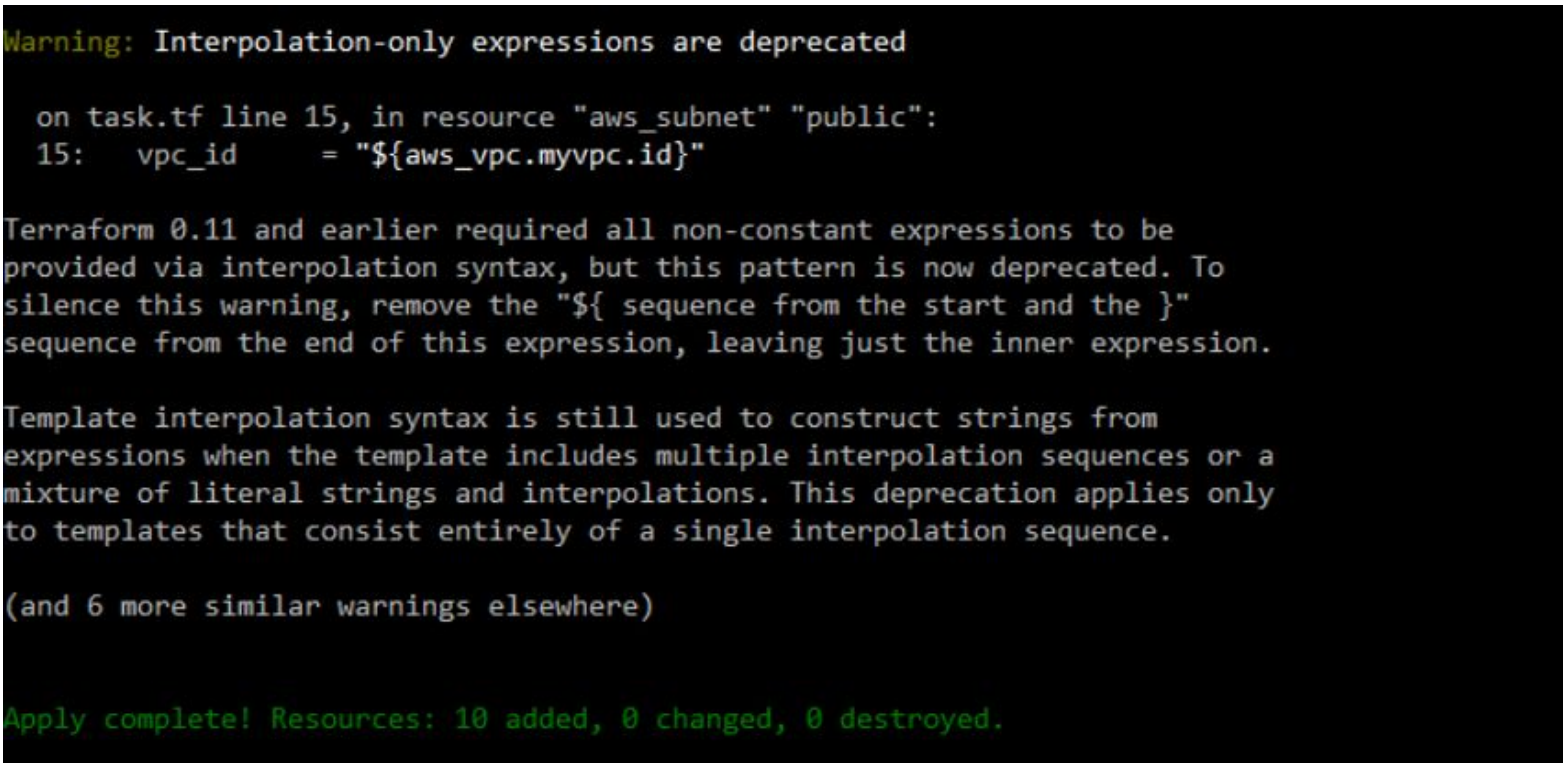
Task 3 :- We have to create a web portal for our company with all the security

Command prompt :-

```
Command Prompt
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Asus>cd desktop

C:\Users\Asus\Desktop>terraformcodefiles
'terraformcodefiles' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Asus\Desktop>cd terraformcodefiles

C:\Users\Asus\Desktop\terraformcodefiles>notepad task3.tf
```



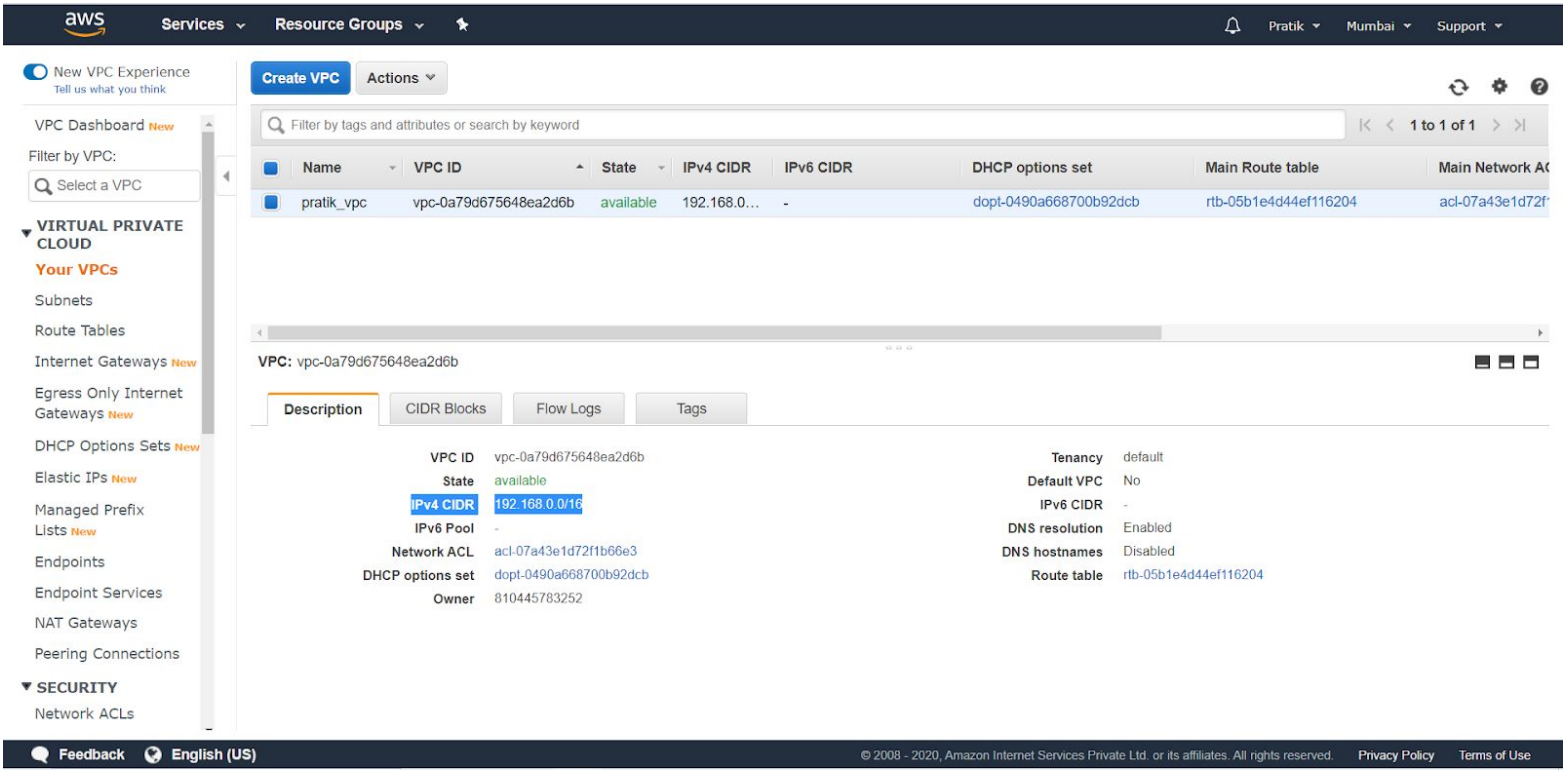




1.login in aws and create a vpc

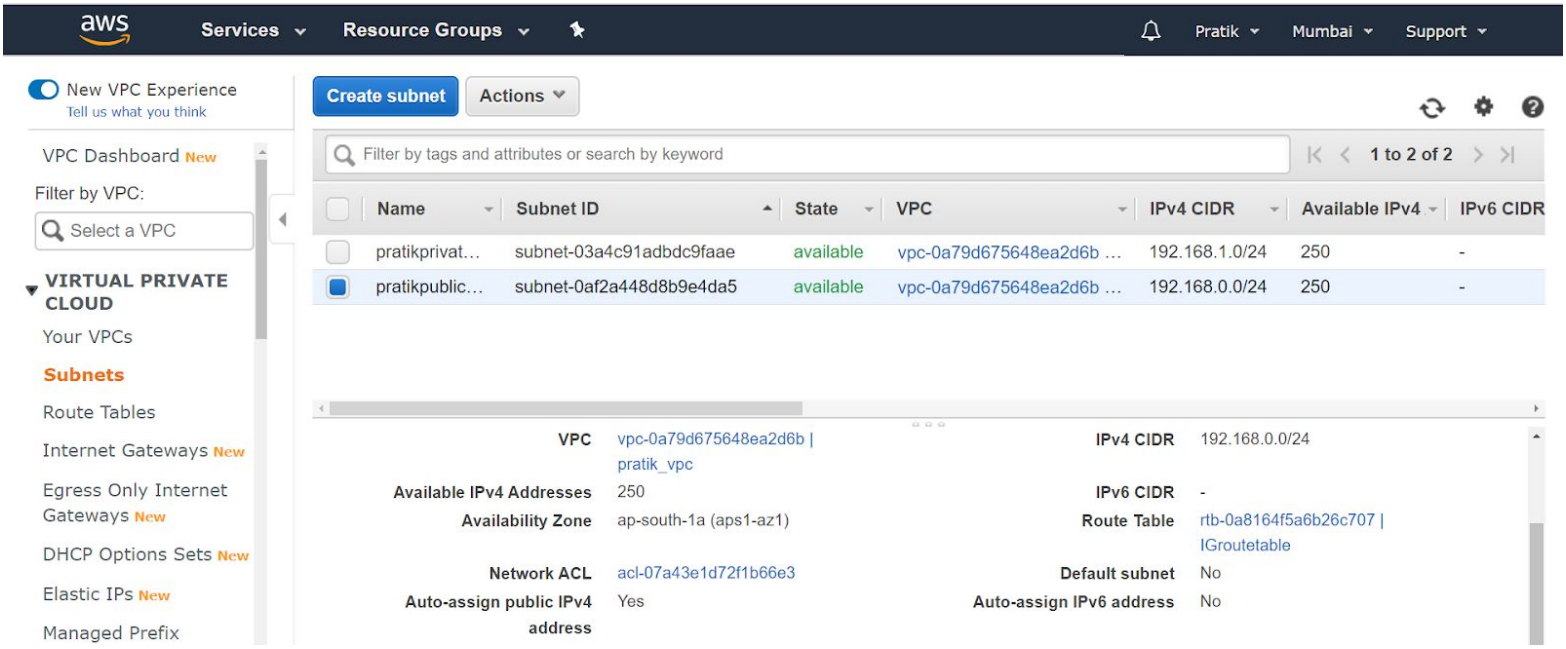
```
provider "aws" {
  region   = "ap-south-1"
  profile  = "Pratik"
}
resource "aws_vpc" "myvpc" {
  cidr_block       = "192.168.0.0/16"
  instance_tenancy = "default"
  enable_dns_hostnames  = true

  tags = {
```

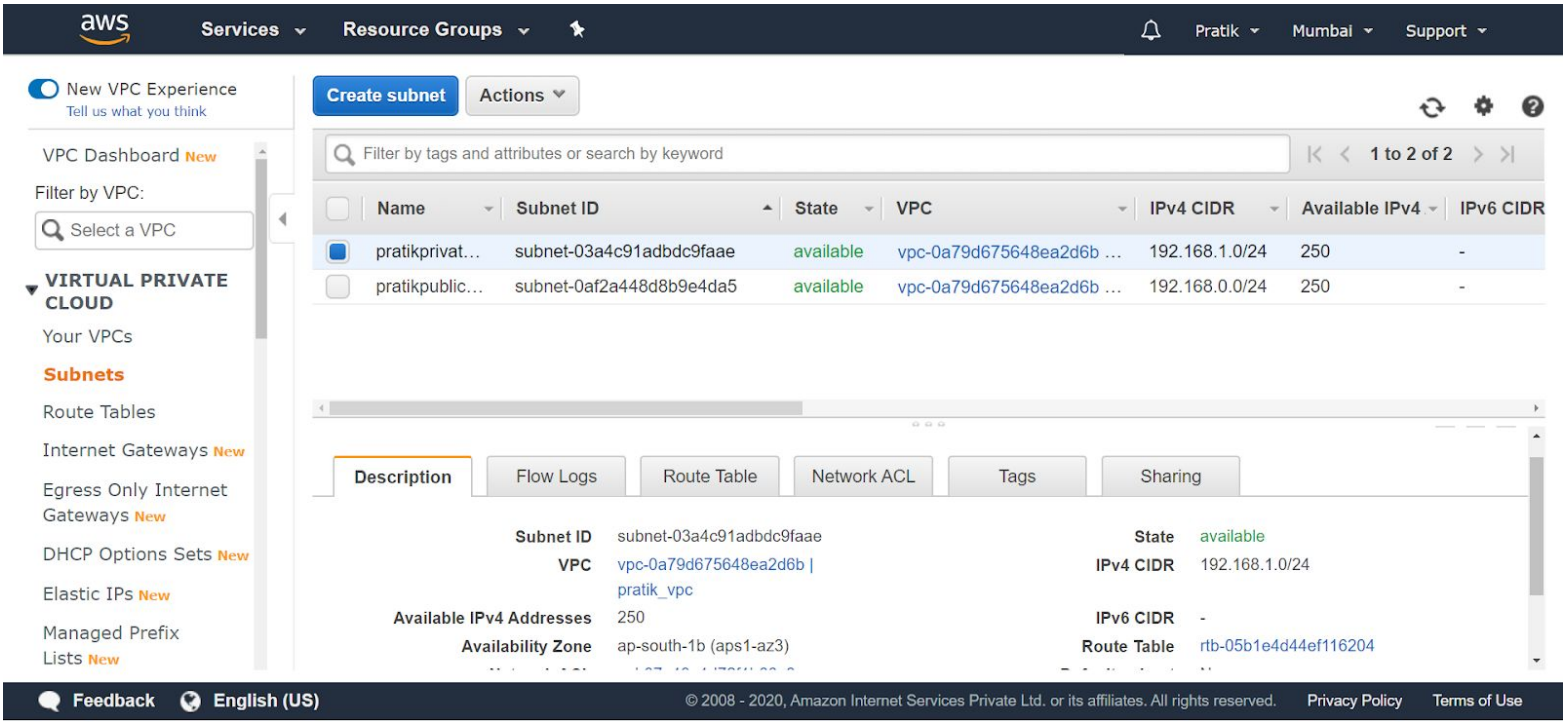
```
  Name = "pratik_vpc"
 }
}
```



2. creating two subnet 1 and has auto-launch ip

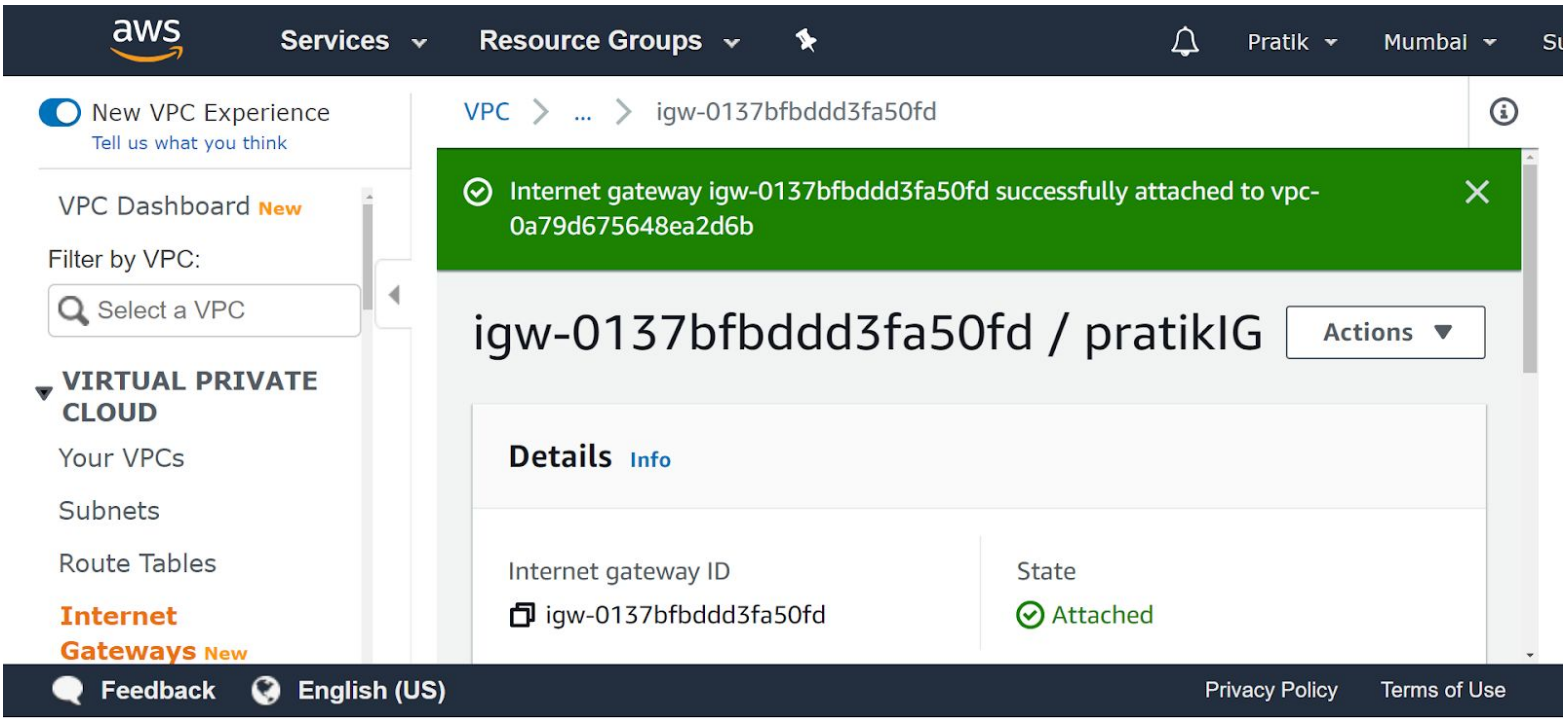
```
resource "aws_subnet" "public" {
 vpc_id     = "${aws_vpc.myvpc.id}"
 cidr_block = "192.168.0.0/24"
 availability_zone = "ap-south-1a"

 tags = {
  Name = "pratikpublicsubnet"
 }
}
resource "aws_subnet" "private" {
 vpc_id     = "${aws_vpc.myvpc.id}"
 cidr_block = "192.168.1.0/24"
 availability_zone = "ap-south-1b"

 tags = {
  Name = "pratikprivatesubnet"
 }
}
```

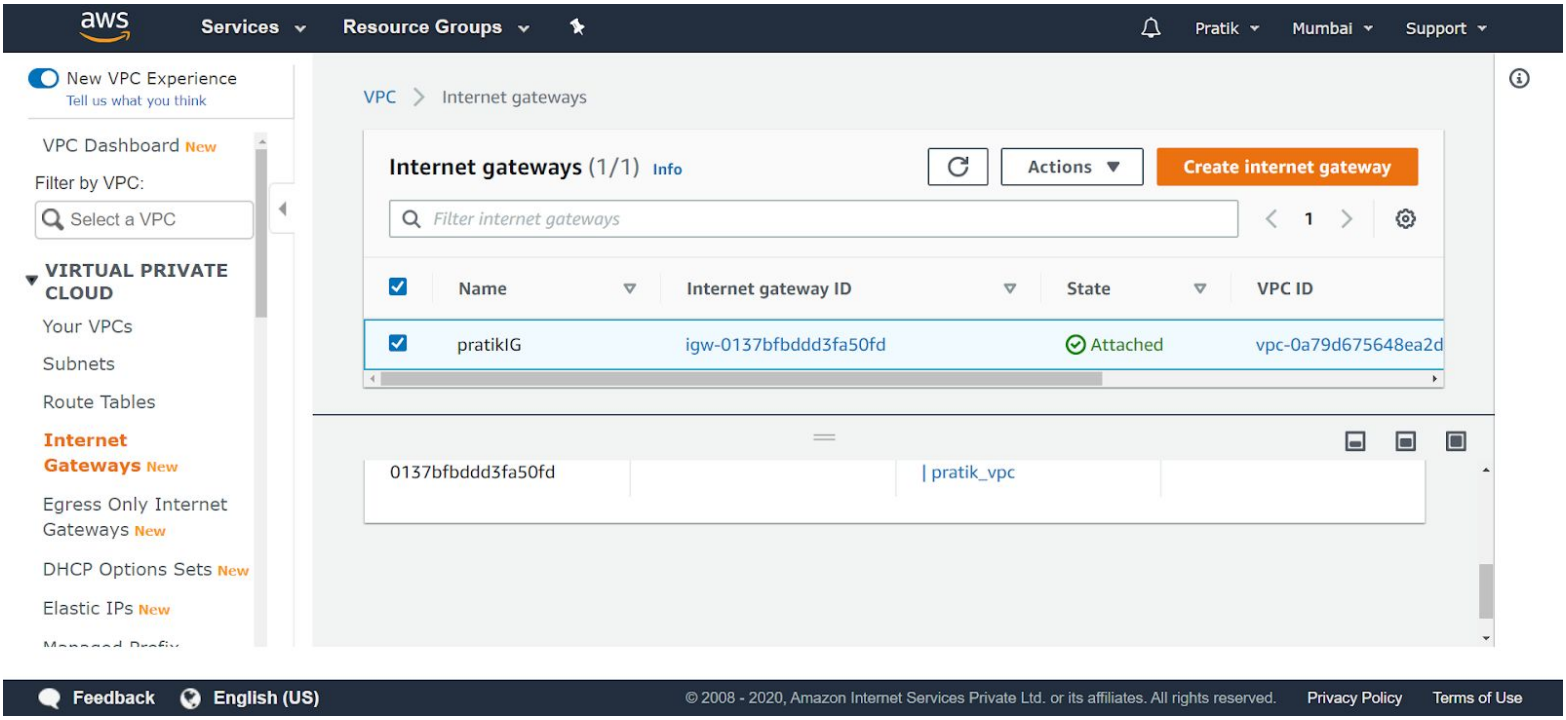




3. creating an internet gateway for a subnet id in south-1a

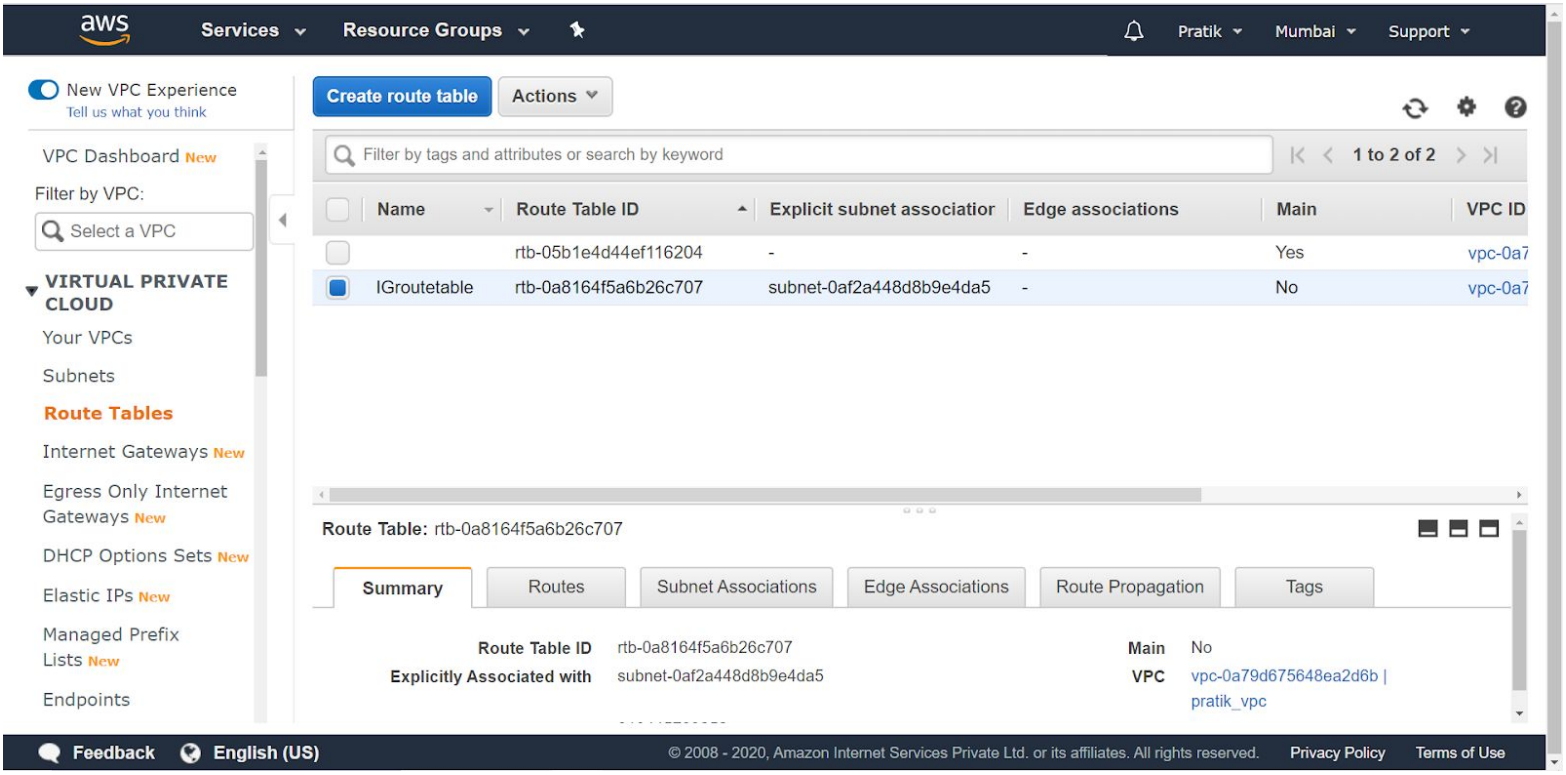
```
resource "aws_internet_gateway" "gw" {
  vpc_id = "${aws_vpc.myvpc.id}"

  tags = {
    Name = "pratikIG"
  }
}
```





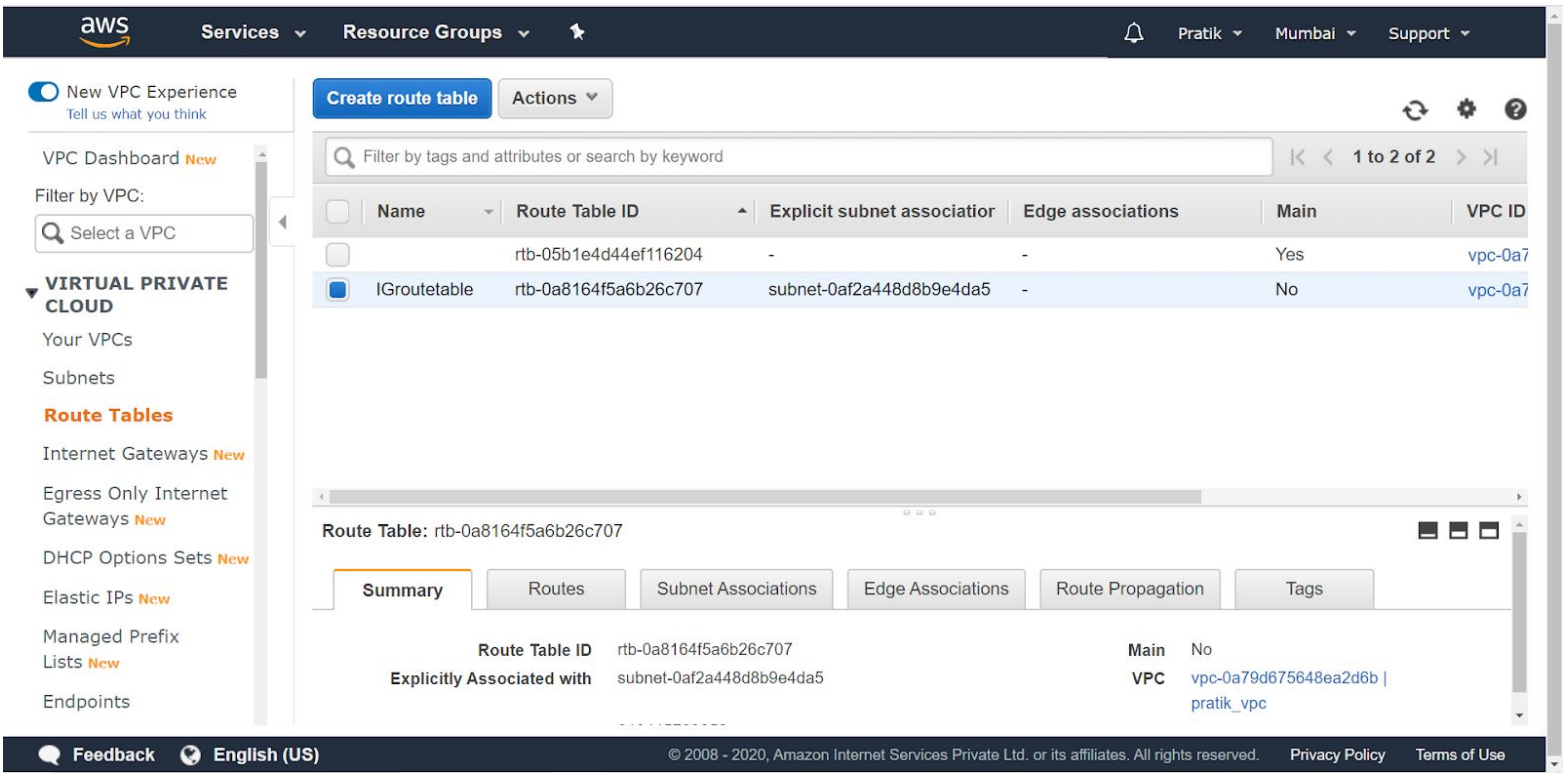
4. creating a route-table > associating route-table with the internet gateway

```
resource "aws_route_table" "forig" {
  vpc_id = "${aws_vpc.myvpc.id}"

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

  tags = {
    Name = "IGroutetable"
  }
}
```



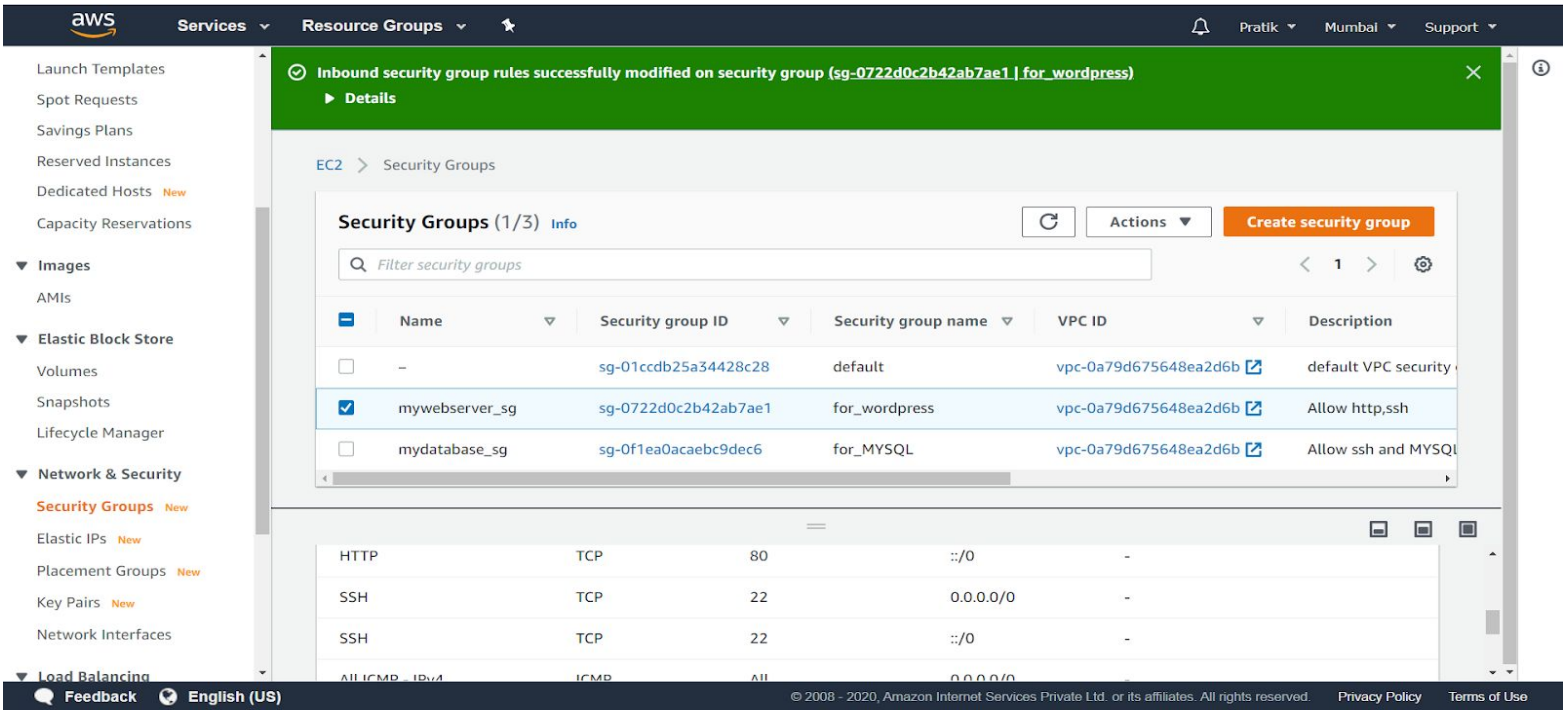
5. Associating route table with subnet

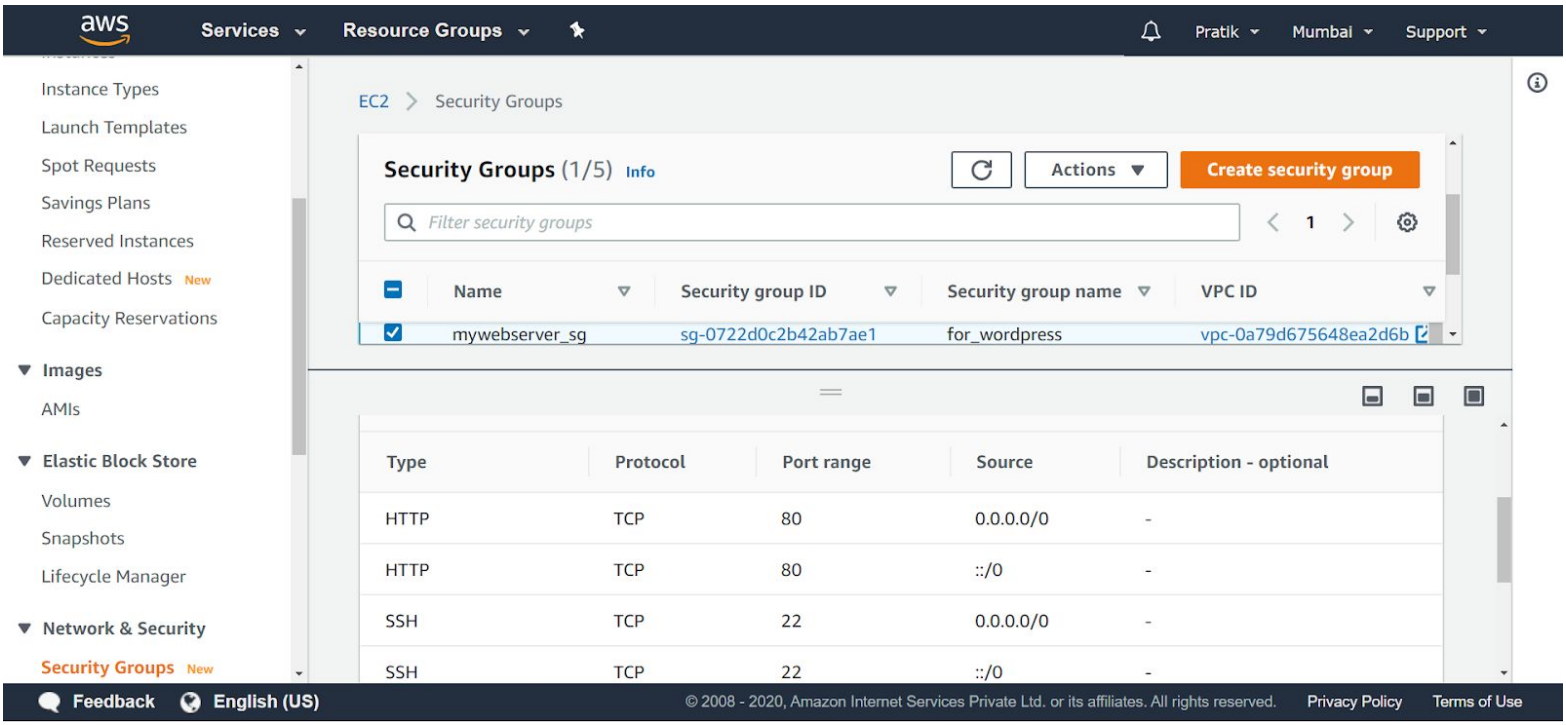
```
resource "aws_route_table_association" "asstopublic" {
  subnet_id      = aws_subnet.public.id
  route_table_id = aws_route_table.forig.id
}
```



6.Creating the security group with ingress(ssh,http and icmpv4 protocol) - mywebserver_sg

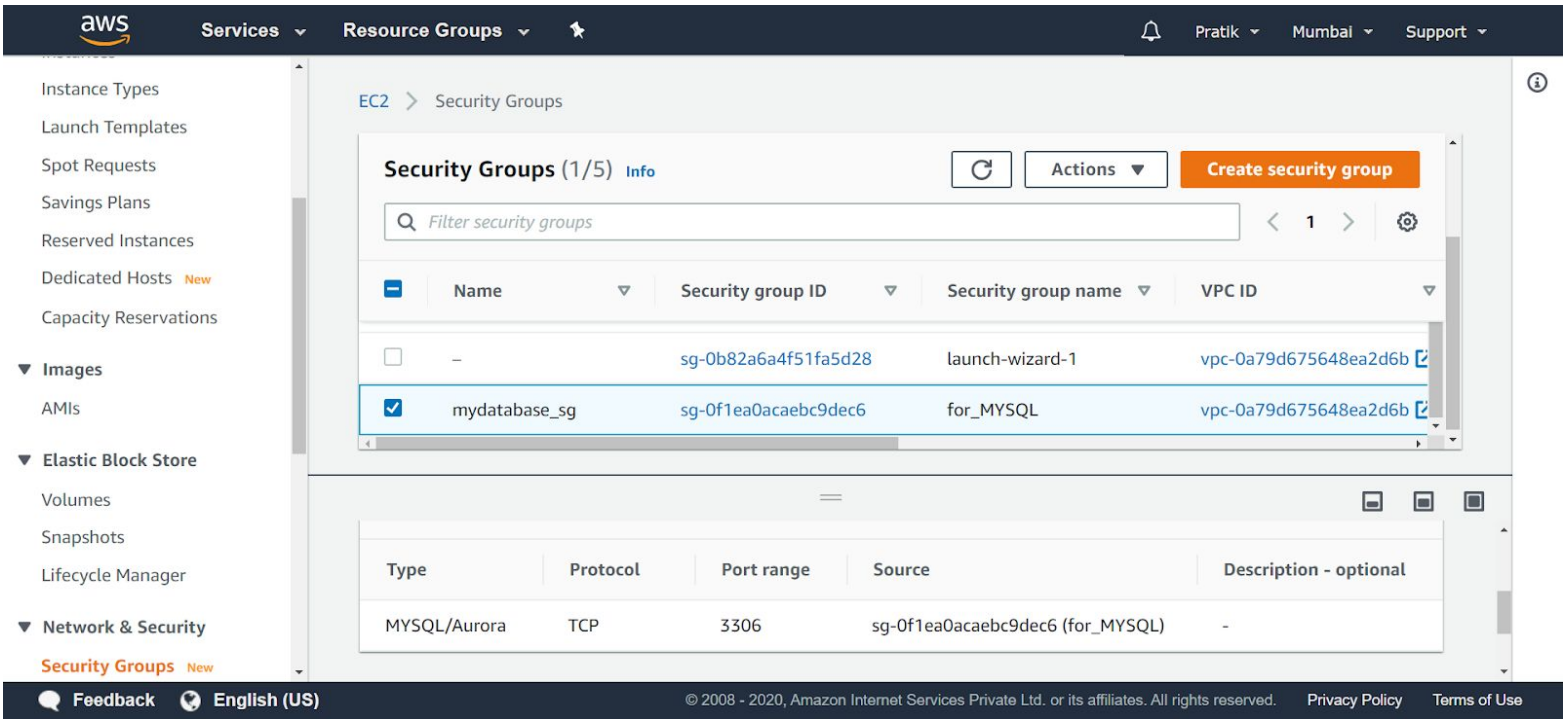
```
resource "aws_security_group" "webserver" {
 name        = "for_wordpress"
 description = "Allow http,ssh"
 vpc_id      = "${aws_vpc.myvpc.id}"

 ingress {
  description = "HTTP"
  from_port   = 80
  to_port     = 80
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
 }
ingress {
  description = "SSH"
  from_port   = 22
  to_port     = 22
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
 }


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

 tags = {
  Name = "mywebserver_sg"
 }
}
```





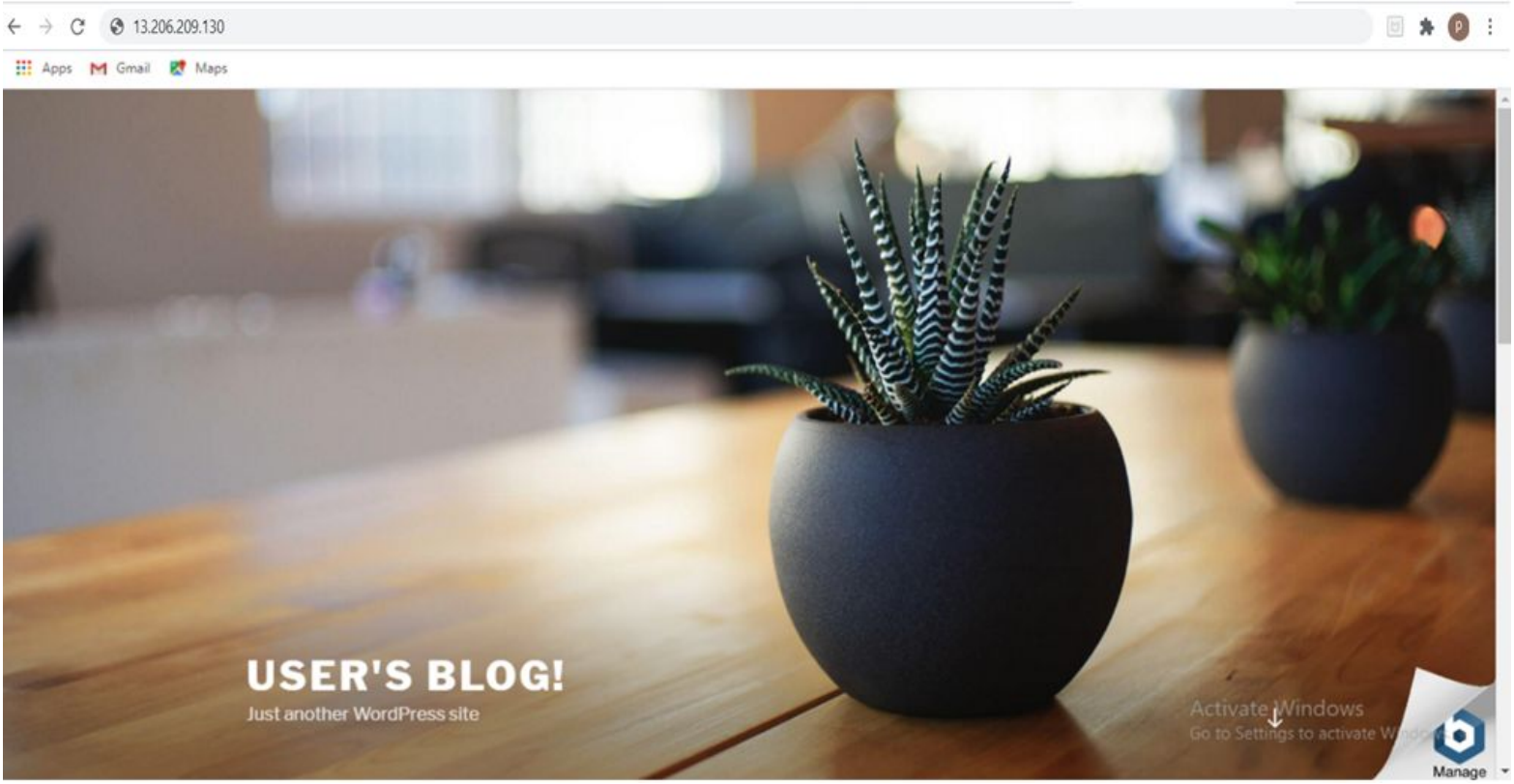
7. Creating a subnet group with MYSQL protocol and value of security_id(myweb) - mydatabase_sg

```
resource "aws_security_group" "database" {
  name        = "for_MYSQL"
  description = "Allow ssh and MYSQL"
  vpc_id      = "${aws_vpc.myvpc.id}"

  ingress {
    description = "MYSQL"
    security_groups = [aws_security_group.webserver.id]
    from_port   = 3306
    to_port     = 3306
    protocol    = "tcp"

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

  tags = {
    Name = "mydatabase_sg"
  }
}
```



8. Launching the instance

```
resource "aws_instance" "wordpress" {
  ami           = "ami-00b494a3f139ba61f"
  instance_type = "t2.micro"
  associate_public_ip_address = true
  subnet_id = aws_subnet.public.id
  vpc_security_group_ids = [aws_security_group.webserver.id]
  key_name = "mykey111"


  tags = {
    Name = "wordpress"
  }

}
resource "aws_instance" "mysql" {
  ami           = "ami-0019ac6129392a0f2"
  instance_type = "t2.micro"
  subnet_id = aws_subnet.private.id
  vpc_security_group_ids = [aws_security_group.database.id]
  key_name = "mykey111"


 tags = {
    Name = "mysql"
  }

}
```




```
Warning: Interpolation-only expressions are deprecated

  on task.tf line 15, in resource "aws_subnet" "public":
  15:   vpc_id     = "${aws_vpc.myvpc.id}"

Terraform 0.11 and earlier required all non-constant expressions to be
provided via interpolation syntax, but this pattern is now deprecated. To
silence this warning, remove the "${ sequence from the start and the }"
sequence from the end of this expression, leaving just the inner expression.

Template interpolation syntax is still used to construct strings from
expressions when the template includes multiple interpolation sequences or a
mixture of literal strings and interpolations. This deprecation applies only
to templates that consist entirely of a single interpolation sequence.

(and 6 more similar warnings elsewhere)


Destroy complete! Resources: 10 destroyed.
```