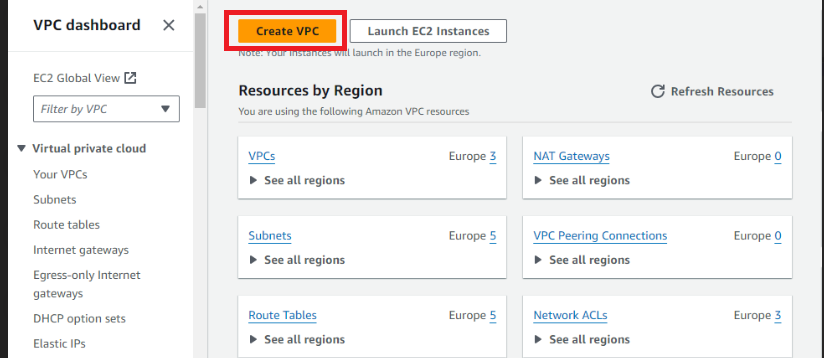
VPC

**What is VPC?**

* VPC stands for Virtual Private Cloud.
* Amazon Virtual Private Cloud (Amazon VPC) provides a logically isolated area of the AWS cloud where you can launch AWS resources in a virtual network that you define.

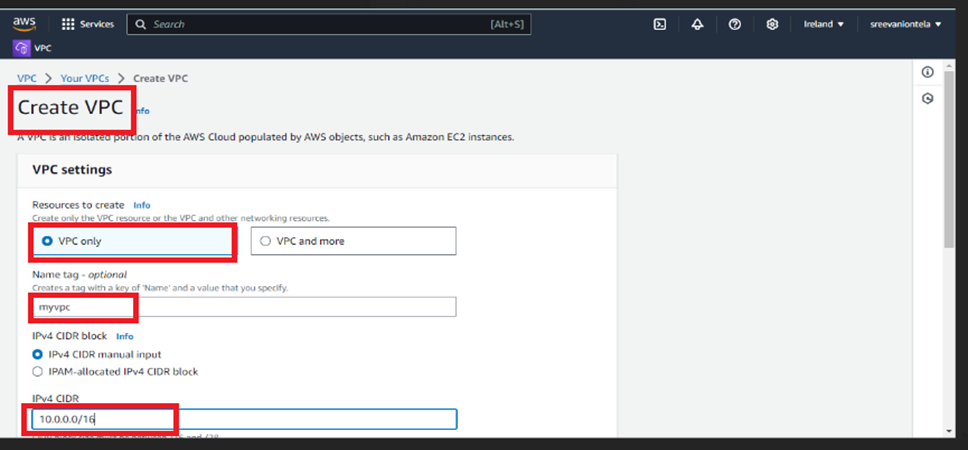
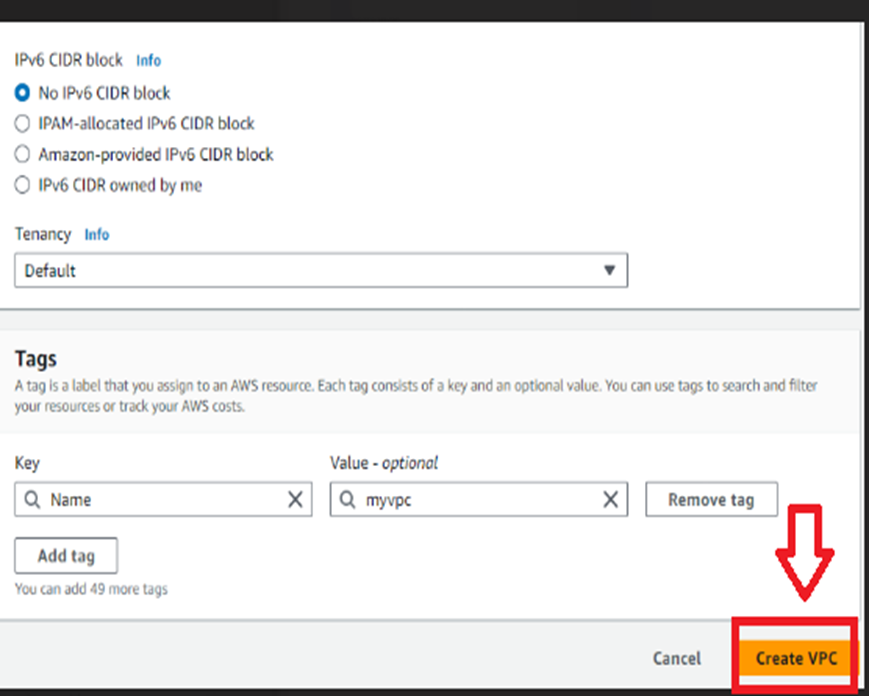
**Step1: we have to create VPC:**

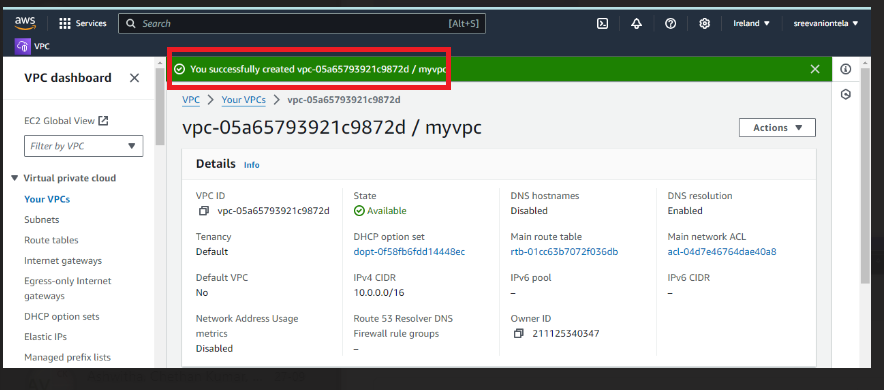
* Sign in to the AWS management console
* Click on the VPC service
* Click on yours VPC’S

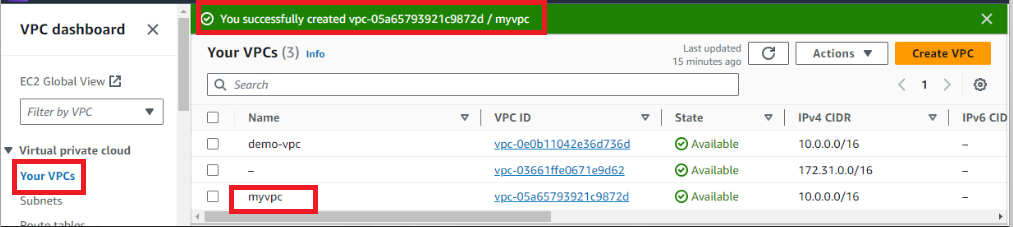


* Create VPC-Click on VPC only-VPC name(optional)-IPV4 CIDR block-

Create VPC.

****

****

****

The above images shows that my vpc is successfully created.

**How to create VPC by using TERRAFORM?**

**TERRAFORM:**

Terraform is an infrastructure as code tool that lets you build, change, and version cloud and on-prem resources safely and efficiently.

**Code:**

**provider "aws" {**

**region = "eu-west-1" # Replace with your desired AWS region**

**}**

**resource "aws\_vpc" "demo-vpc" {**

**cidr\_block = "10.0.0.0/16"**

**tags = {**

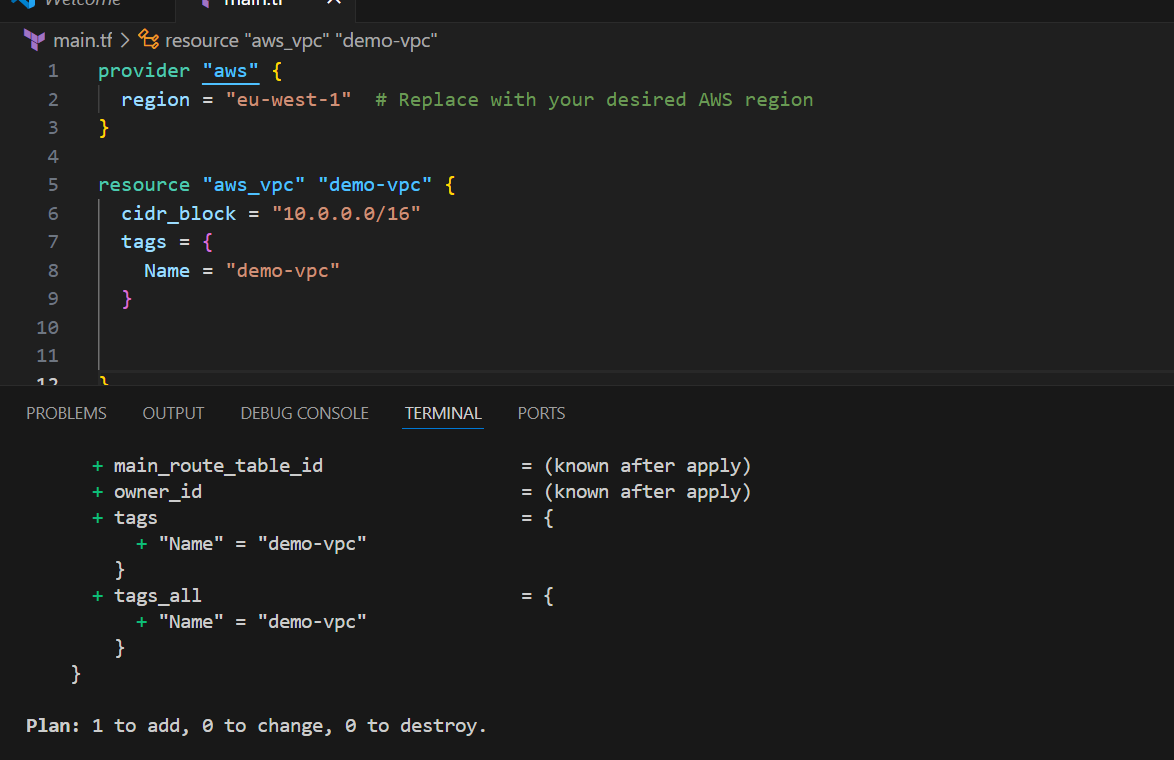
**Name = "demo-vpc"**

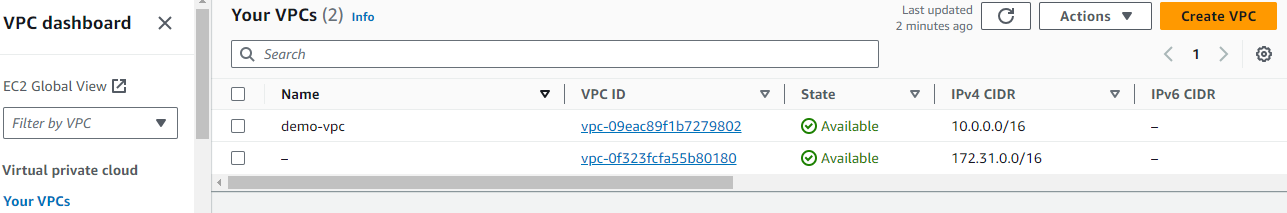
**}**

}

After giving the code in terraform we are using these commands.

* Aws configure
* Access key (security credentials)
* Secret access key
* Terraform init
* Terraform validate
* Terraform plan
* Terraform apply

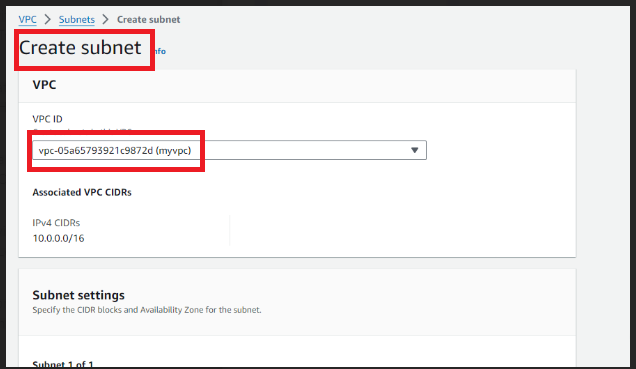


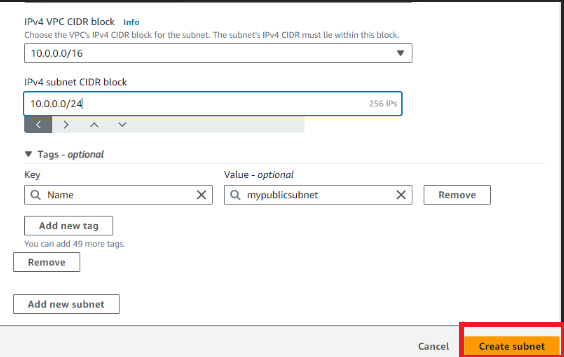


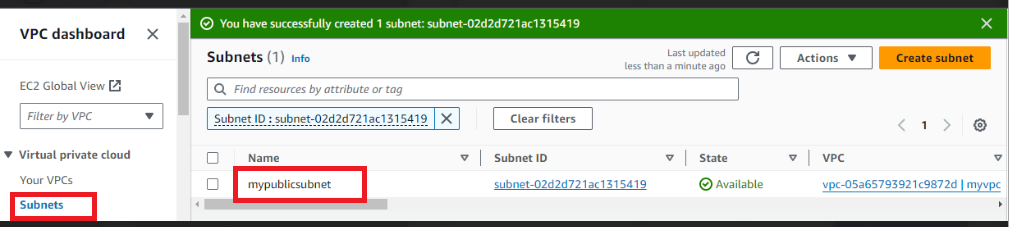
**What is subnets?**

* A subnet is a range of IP addresses in your VPC.
* You launch AWS resources, such as Amazon EC2 instances, into your subnets.
* Subnets are regional resources.
* Each subnet defines a range of IPv4 addresses

**Step2: we have to create subnets:** Create subnet-VPC ID-Subnet name -IPV4 subnet CIDR block-create subnet

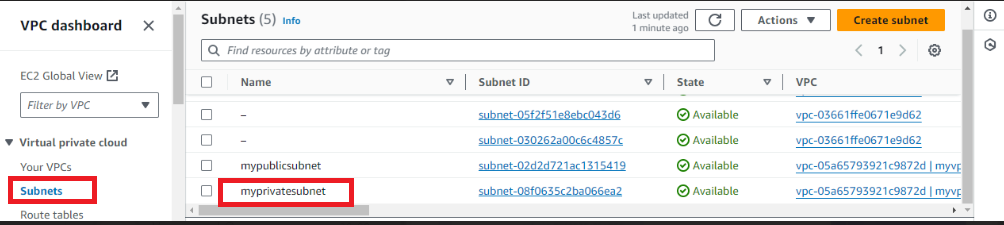
****

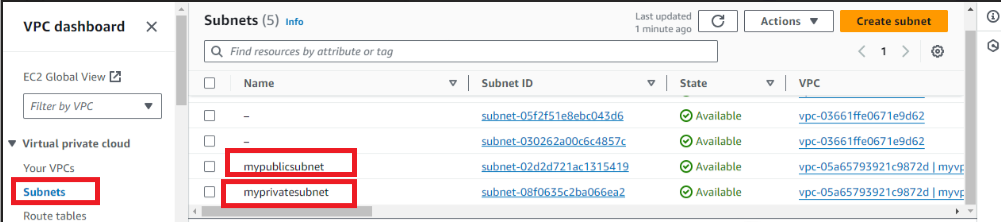
****

****

The above image shows successfully created public subnet.

As we created the public subnet and follow the same process and change IPV4 CIDR block as 10.0.1.0/24 then created the private subnet.





The above image shows that the two subnets created successfully.

**CODE:**

**resource "aws\_subnet" "public-subnet-1" {**

**vpc\_id     = aws\_vpc.demo-vpc.id**

**cidr\_block = "10.0.3.0/24"**

**availability\_zone = "eu-west-1a"**

**tags = {**

**Name = "public-subnet-1"**

**}**

**}**

**resource "aws\_subnet" "public-subnet-2" {**

**vpc\_id     = aws\_vpc.demo-vpc.id**

**cidr\_block = "10.0.4.0/24"**

**availability\_zone = "eu-west-1b"**

**tags = {**

**Name = "public-subnet-2"**

**}**

**}**

**resource "aws\_subnet" "private-subnet-1" {**

**vpc\_id     = aws\_vpc.demo-vpc.id**

**cidr\_block = "10.0.1.0/24"**

**availability\_zone = "eu-west-1a"**

**tags = {**

**Name = "private-subnet-1"**

**}**

**}**

**resource "aws\_subnet" "private-subnet-2" {**

**vpc\_id     = aws\_vpc.demo-vpc.id**

**cidr\_block = "10.0.2.0/24"**

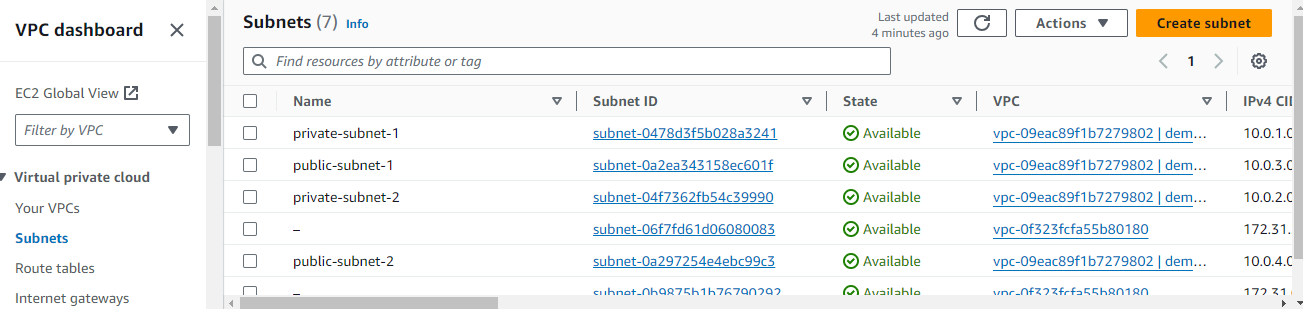
**availability \_zone = "eu-west-1b"**

**tags = {**

**Name = "private-subnet-2"**

**}**

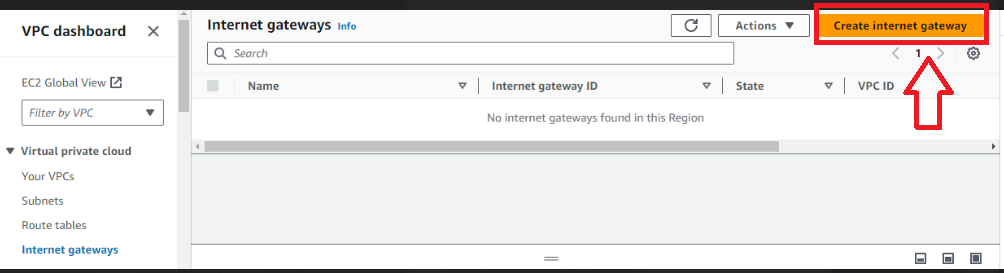
**}**

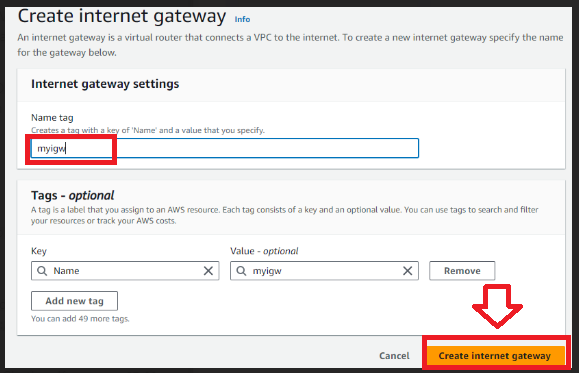
****

**INTERNET GATEWAY**

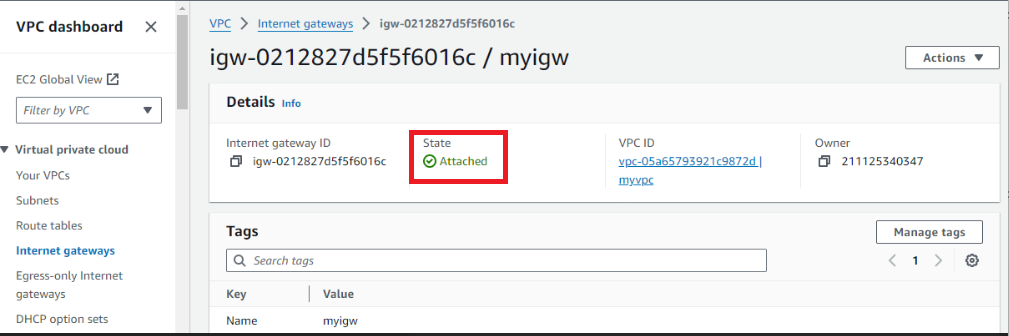
An internet gateway is a logical connection between the vpc and internet. It allows communication between resources within the vpc and the internet. Each vpc has only one IGW and supports both IPv4 and IPv6 traffic.

Here we are create the internet gateway.





The below image shows the internet gateways is attached to the vpc.



**CODE:**

**resource "aws\_internet\_gateway" "demo-igw" {**

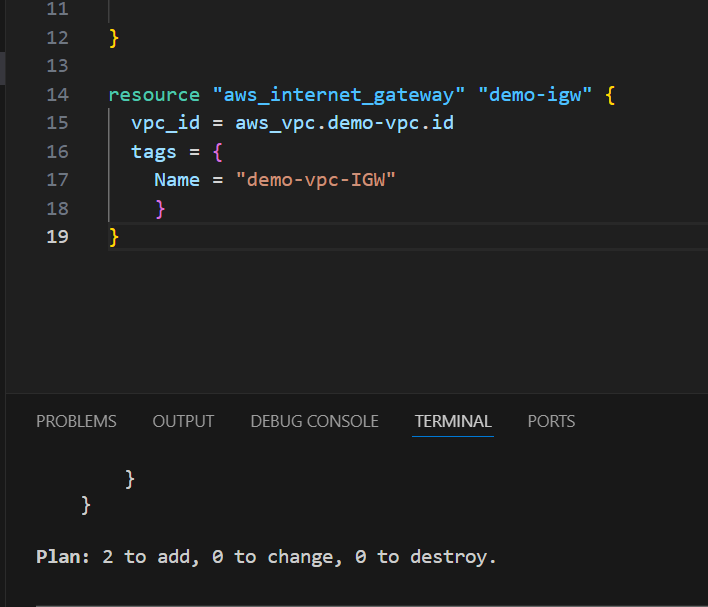
**vpc\_id = aws\_vpc.demo-vpc.id**

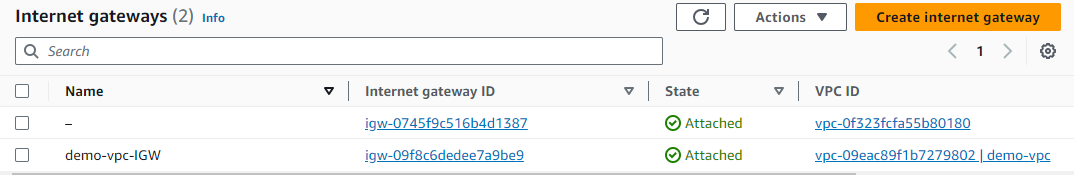
**tags = {**

**Name = "demo-vpc-IGW"**

**}**

**}**

****

****

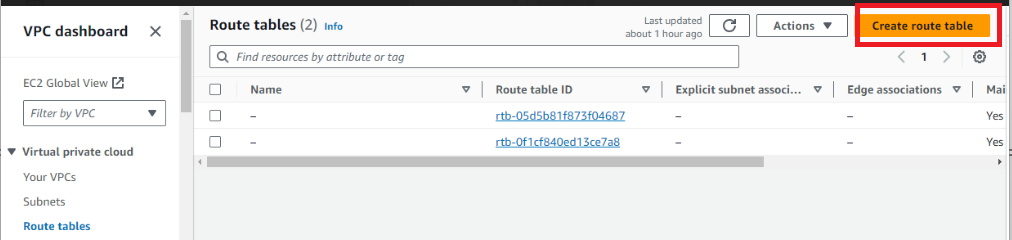
**ROUTE TABLES**

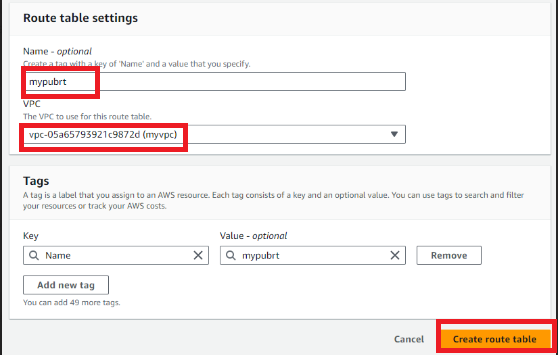
**What is ROUTE TABLES?**

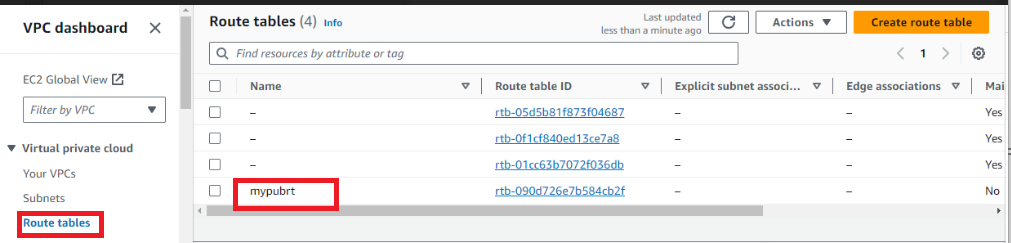
In AWS, a route table is a set of rules that determines where network traffic is directed. Each subnet in your aws virtual private cloud is associated with a route table traffic flow between subnets. The route tables includes details like the route table ID and the ID of its associated vpc**.**

**Here we are creating route table, the below images shows**

Create route table-route table name -select vpc-create route table.

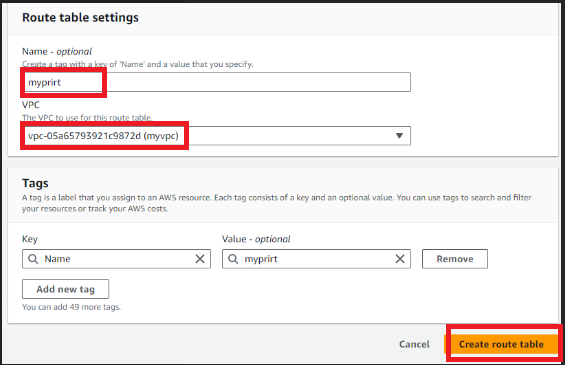
****

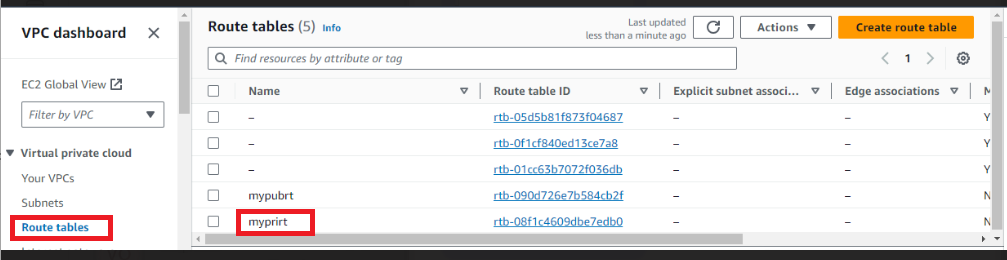
****

****

The above image shows we are successfully created public route table.

The below image shows that we are creating private route table.



****

The above image shows that the public and private route tables.

**CODE:**

**resource "aws\_route\_table" "public-route-table" {**

**vpc\_id = aws\_vpc.demo-vpc.id**

**tags = {**

**Name = "public-route-table"**

**}**

**}**

**resource "aws\_route" "public-route" {**

**route\_table\_id         = aws\_route\_table.public- route-table.id**

**destination\_cidr\_block = "0.0.0.0/0"**

**gateway\_id             = aws\_internet\_gateway.demo-igw.id**

**}**

**resource "aws\_route\_table\_association" "public-subnet-1-association" {**

**subnet\_id      = aws\_subnet.public-subnet-1.id**

**route\_table\_id = aws\_route\_table.public-route-table.id**

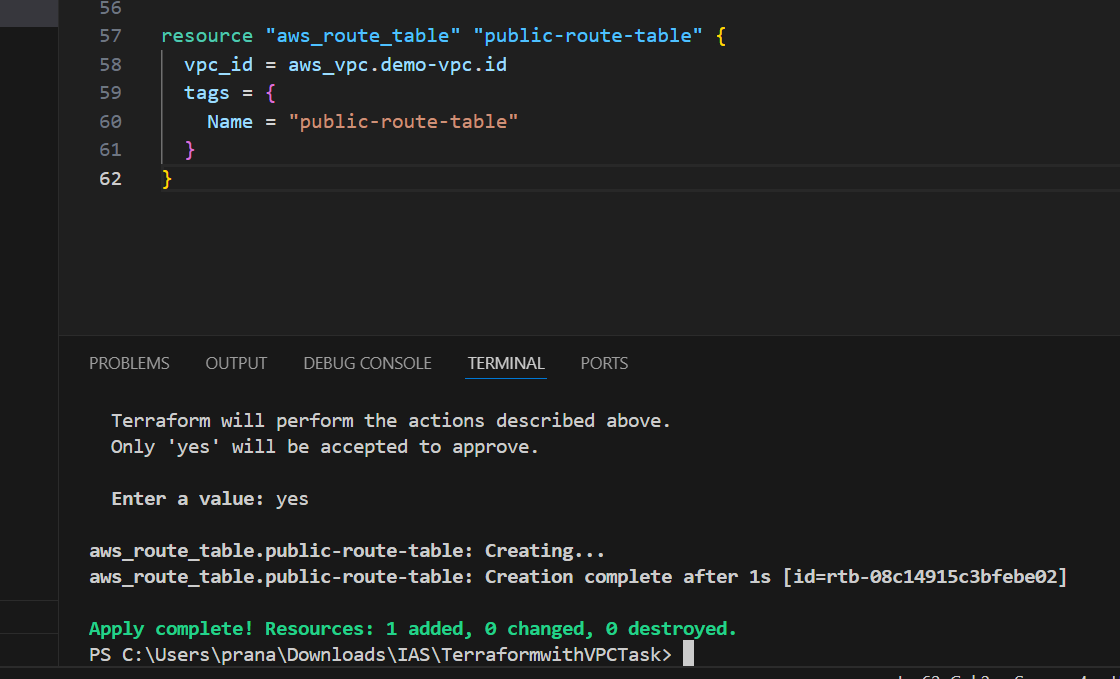
**}**

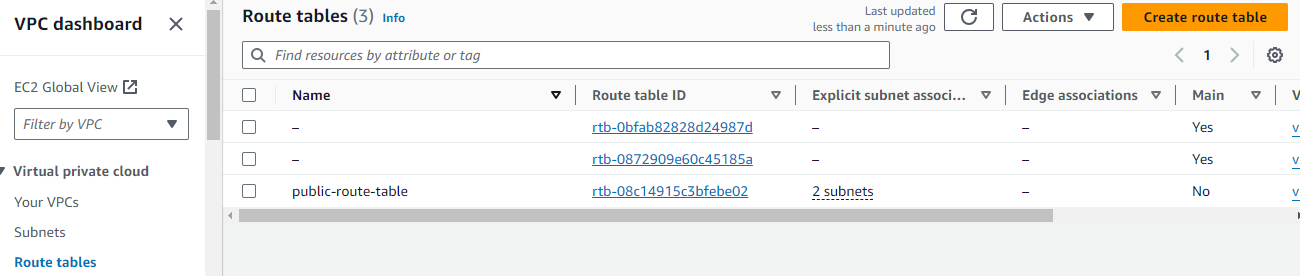
**resource "aws\_route\_table\_association" "public-subnet-2-association" {**

**subnet\_id      = aws\_subnet.public-subnet-2.id**

**route\_table\_id = aws\_route\_table.public-route-table.id**

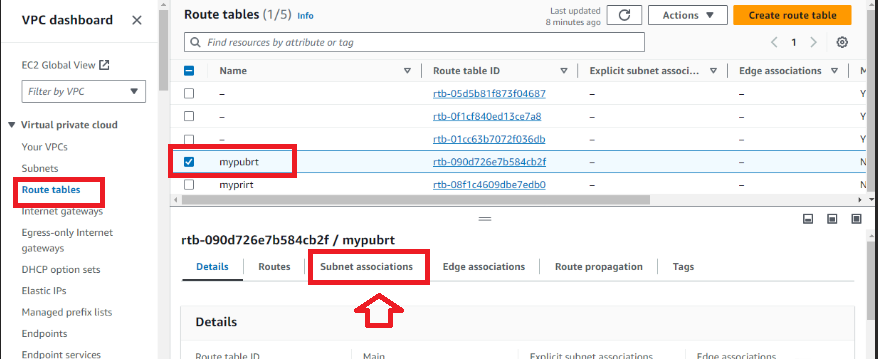
**}**

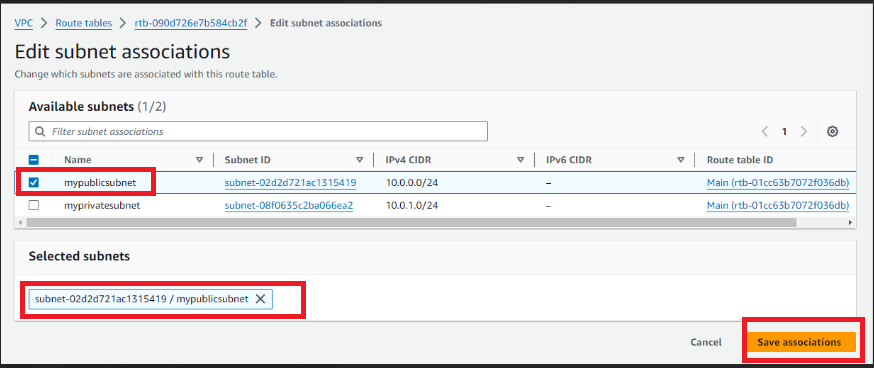
****

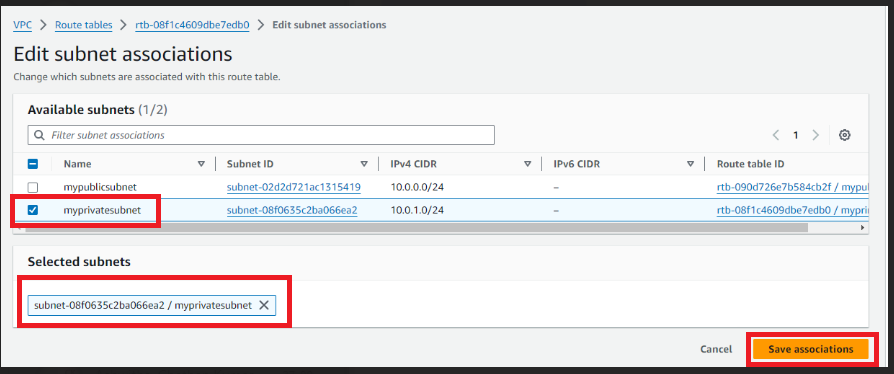
****

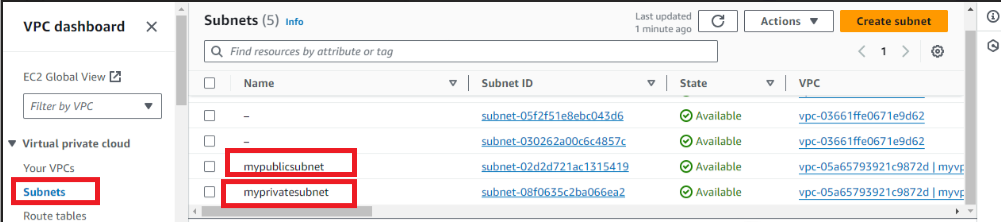
Here we are creating subnet associations to our public and private Subnets.

Go to route tables-public route table-subnet associations-my public subnet-save associations.

****

****

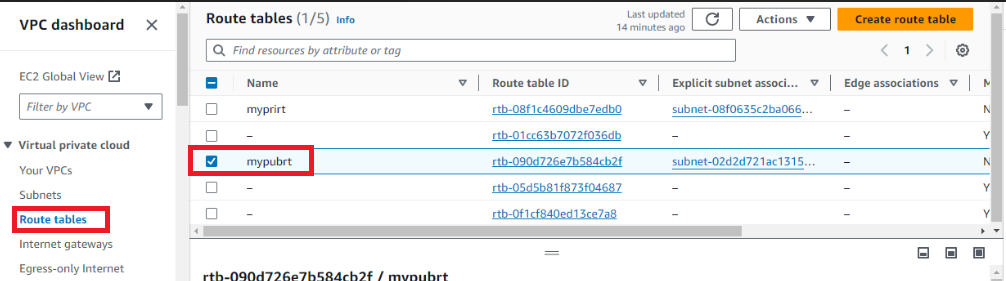
****

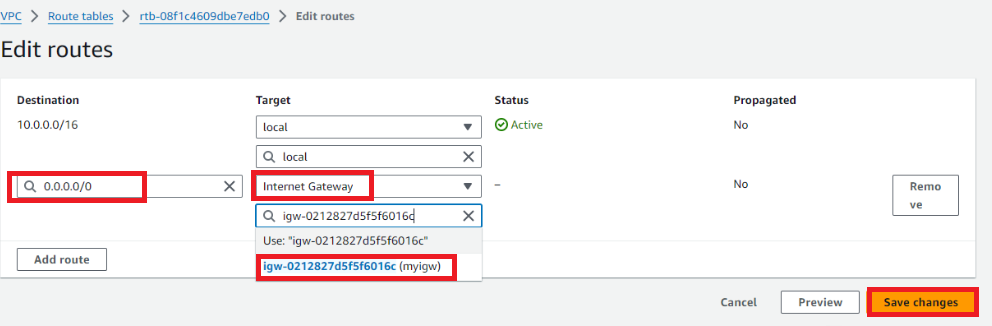
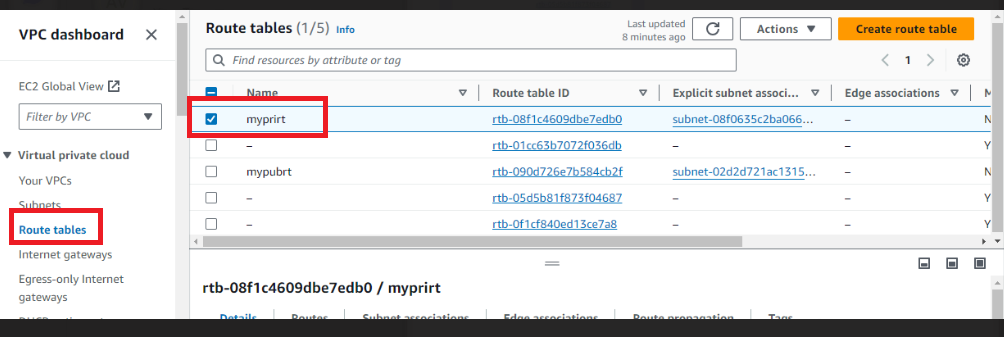
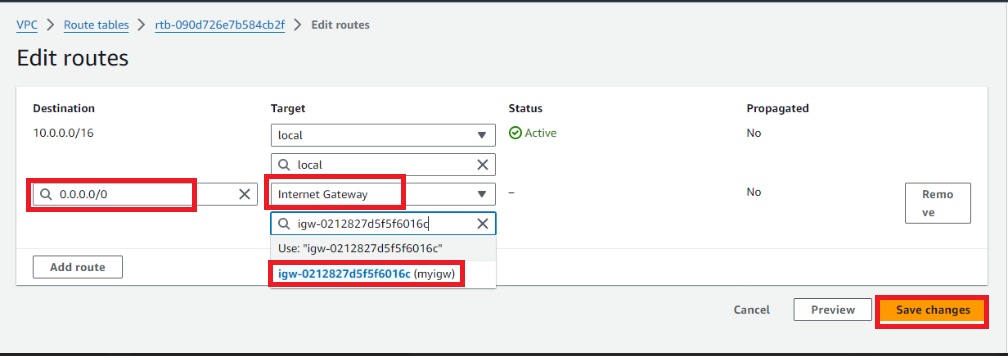
****

The above image shows the successfully associated subnets.

We are edit routes to our public and private route tables.

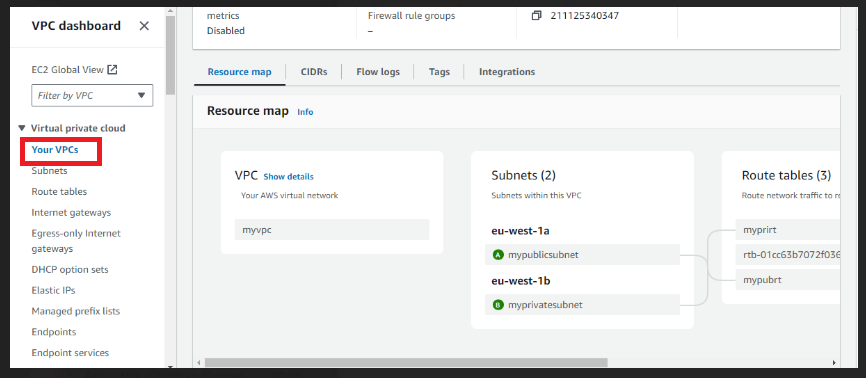
Click on public route table-edit routes-internet gateway-save changes.





Go to your vpc’s-click on vpc-click on resource map

Here we can see the final output in vpc .



**CODE:**

**provider "aws" {**

**region = "eu-west-1" # Replace with your desired AWS region**

**}**

**resource "aws\_vpc" "demo-vpc" {**

**cidr\_block = "10.0.0.0/16"**

**tags = {**

**Name = "demo-vpc"**

**}**

**}**

**resource "aws\_internet\_gateway" "demo-igw" {**

**vpc\_id = aws\_vpc.demo-vpc.id**

**tags = {**

**Name = "demo-vpc-IGW"**

**}**

**}**

**resource "aws\_subnet" "private-subnet-1" {**

**vpc\_id = aws\_vpc.demo-vpc.id**

**cidr\_block = "10.0.1.0/24"**

**availability\_zone = "eu-west-1a"**

**tags = {**

**Name = "private-subnet-1"**

**}**

**}**

**resource "aws\_subnet" "private-subnet-2" {**

**vpc\_id = aws\_vpc.demo-vpc.id**

**cidr\_block = "10.0.2.0/24"**

**availability\_zone = "eu-west-1b"**

**tags = {**

**Name = "private-subnet-2"**

**}**

**}**

**resource "aws\_subnet" "public-subnet-1" {**

**vpc\_id = aws\_vpc.demo-vpc.id**

**cidr\_block = "10.0.3.0/24"**

**availability\_zone = "eu-west-1a"**

**tags = {**

**Name = "public-subnet-1"**

**}**

**}**

**resource "aws\_subnet" "public-subnet-2" {**

**vpc\_id = aws\_vpc.demo-vpc.id**

**cidr\_block = "10.0.4.0/24"**

**availability\_zone = "eu-west-1b"**

**tags = {**

**Name = "public-subnet-2"**

**}**

**}**

**resource "aws\_route\_table" "public-route-table" {**

**vpc\_id = aws\_vpc.demo-vpc.id**

**tags = {**

**Name = "public-route-table"**

**}**

**}**

**resource "aws\_route" "public-route" {**

**route\_table\_id = aws\_route\_table.public-route-table.id**

**destination\_cidr\_block = "0.0.0.0/0"**

**gateway\_id = aws\_internet\_gateway.demo-igw.id**

**}**

**resource "aws\_route\_table\_association" "public-subnet-1-association" {**

**subnet\_id = aws\_subnet.public-subnet-1.id**

**route\_table\_id = aws\_route\_table.public-route-table.id**

**}**

**resource "aws\_route\_table\_association" "public-subnet-2-association" {**

**subnet\_id = aws\_subnet.public-subnet-2.id**

**route\_table\_id = aws\_route\_table.public-route-table.id**

**}**

**resource "aws\_eip" "nat-eip" {**

**vpc = true**

**tags = {**

**Name = "nat-eip"**

**}**

**}**

**resource "aws\_nat\_gateway" "nat-gateway" {**

**allocation\_id = aws\_eip.nat-eip.id**

**subnet\_id = aws\_subnet.public-subnet-1.id**

**tags = {**

**Name = "nat-gateway"**

**}**

**}**

**resource "aws\_security\_group" "secgroup" {**

**name = "secgroup"**

**description = "awssecuritygroup"**

**vpc\_id = aws\_vpc.demo-vpc.id**

**ingress {**

**from\_port = 0**

**to\_port = 65535**

**protocol = "tcp"**

**cidr\_blocks = ["0.0.0.0/0"]**

**}**

**egress {**

**from\_port =0**

**to\_port = 65535**

**protocol ="tcp"**

**cidr\_blocks = ["0.0.0.0/0"]**

**}**

**tags = {**

**name ="secgroup"**

**}**

**}**

**resource "aws\_network\_acl" "pub\_az1\_nacl" {**

**vpc\_id = aws\_vpc.demo-vpc.id**

**subnet\_ids = [ aws\_subnet.public-subnet-1.id]**

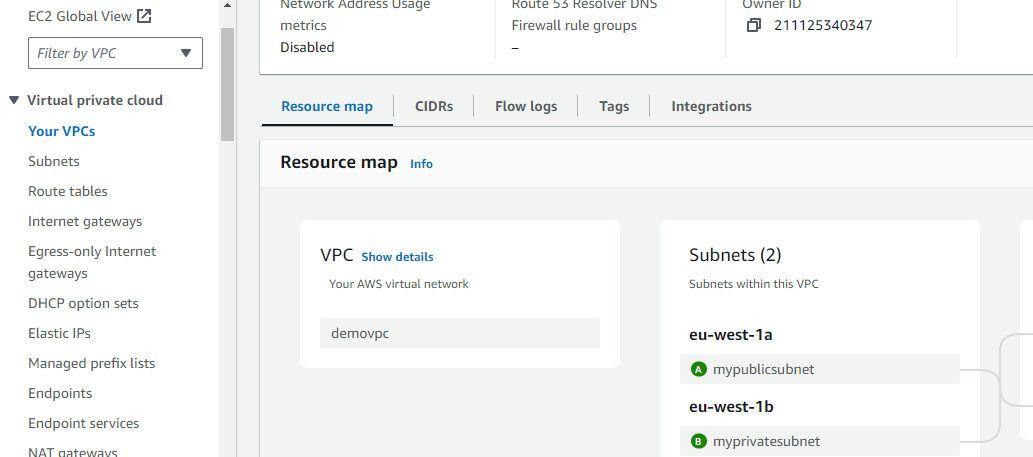
**tags = {**

**Name = "Public-AZ1-NACL"**

**}**

**}**

By applying the code in our terraform we are using commands-terraform init-terraform validate-terraform plan-terraform apply command we can see the vpc is reflected in our console and then go to your vpc’s and click on the particular vpc and click on the resource map**.**

Here the above image also showing the create vpc by using terraform is successfully executed and finally result is as shown in the image.