Amazon VPC or Amazon Virtual Private Cloud is a service that allows its users to launch their virtual machines in a protected as well as isolated virtual environment defined by them. You have complete control over your VPC, from creation to customization and even deletion.

**VPC**

**CODE**:

provider "aws" {

  region = "eu-west-1"

}

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

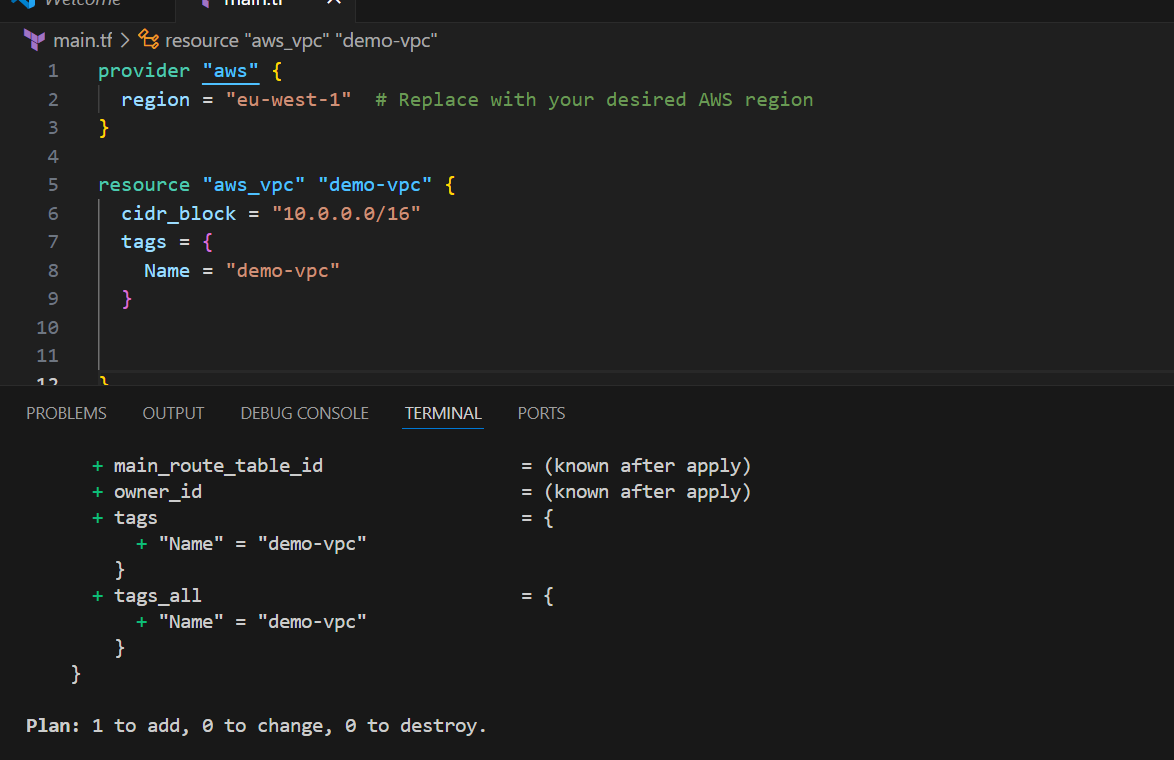
  cidr\_block = "10.0.0.0/16"

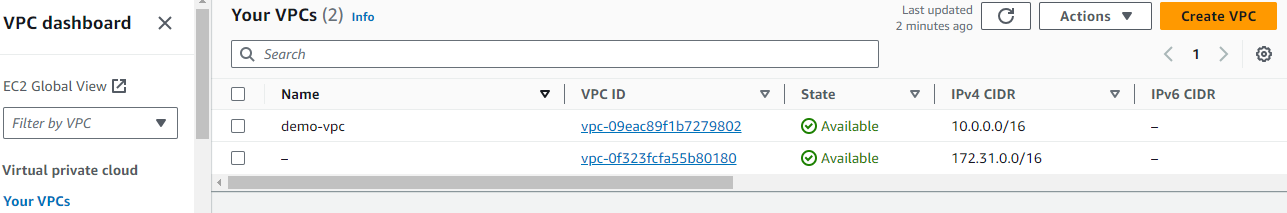
  tags = {

    Name = "demo-vpc"

  }

}





**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.

**CODE**:

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

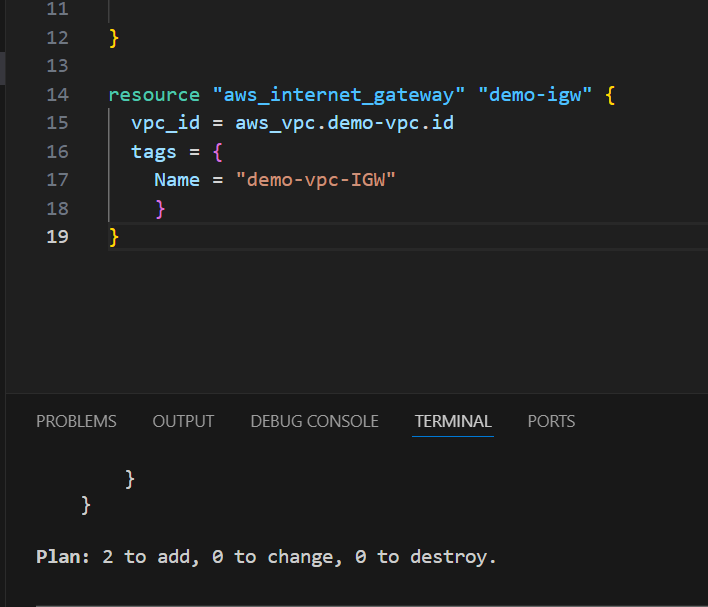
  vpc\_id = aws\_vpc.demo-vpc.id

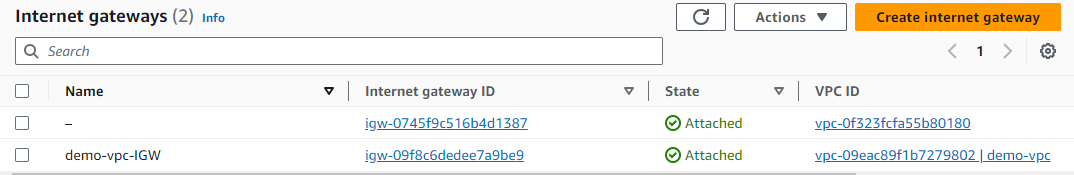
  tags = {

    Name = "demo-vpc-IGW"

    }

}





**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.
* Traffic to and from instances can be controlled with network firewall rules.

**CODE:**

  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"

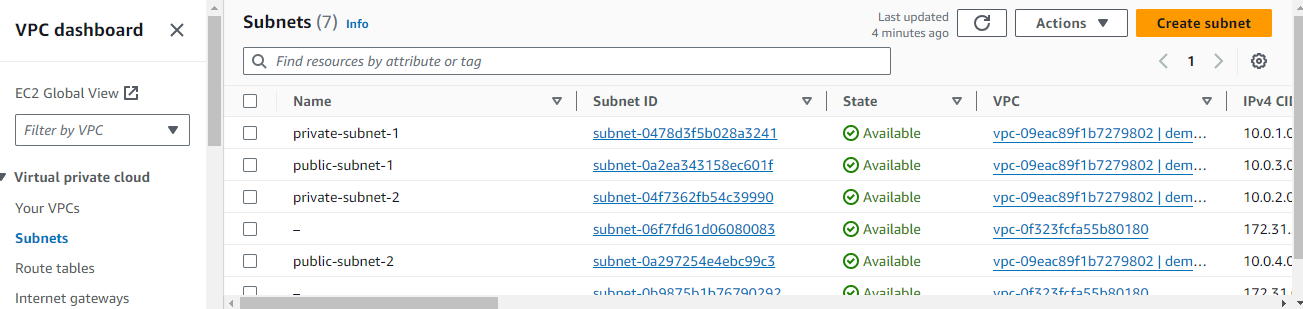
  availability\_zone = "eu-west-1b"

  tags = {

    Name = "public-subnet-2"

  }

}



**ROUTE TABLES**

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

**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

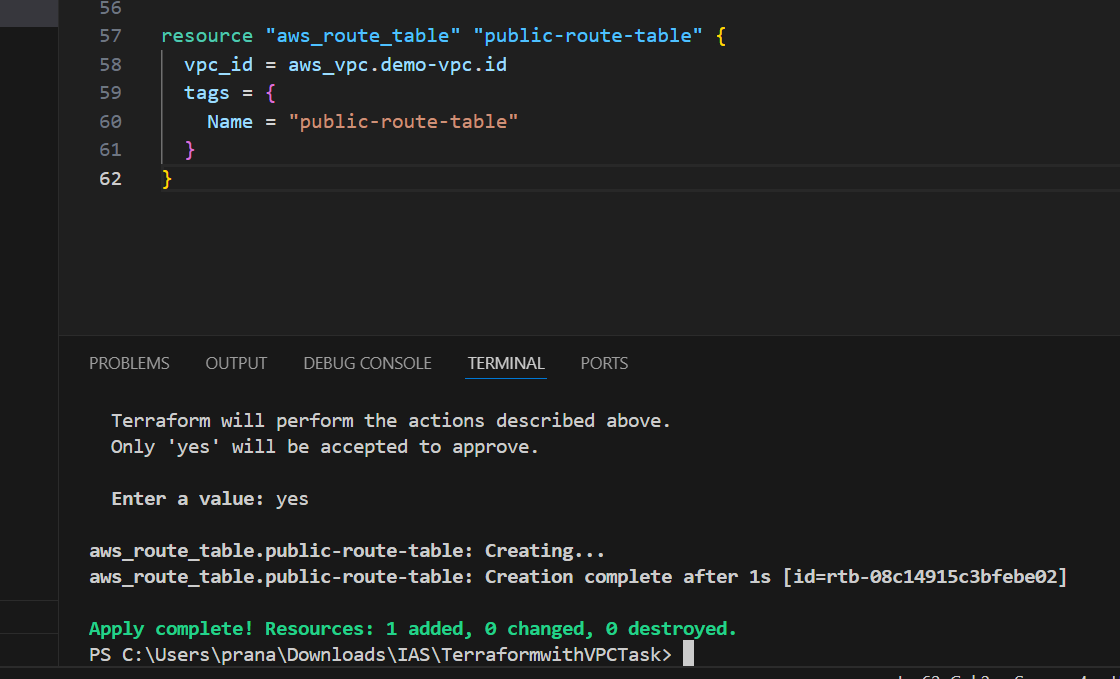
}

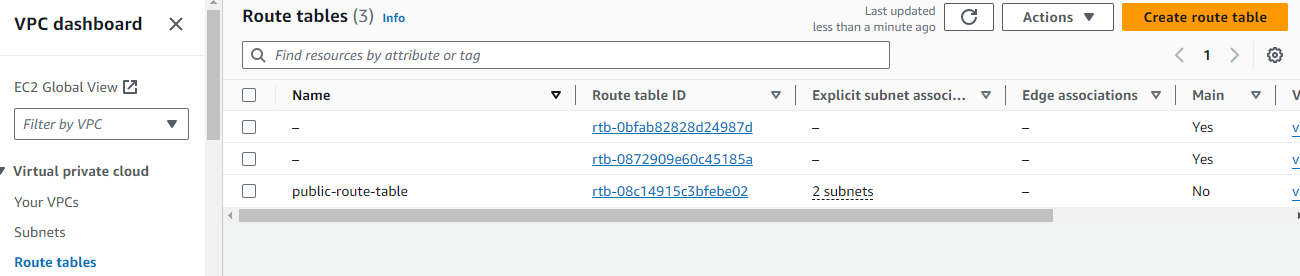
resource "aws\_route\_table\_association" "public-subnet-2-association" {

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

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

}





**NAT GATEWAY**

AWS NAT Gateway – stands for Network Address Translation. It is a managed AWS service that is scaled based on your usage. NAT Gateway will help you to access the internet which instances are configured in the private subnet but without proper routing, no one can access that instance from outside.

**CODE:**

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

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

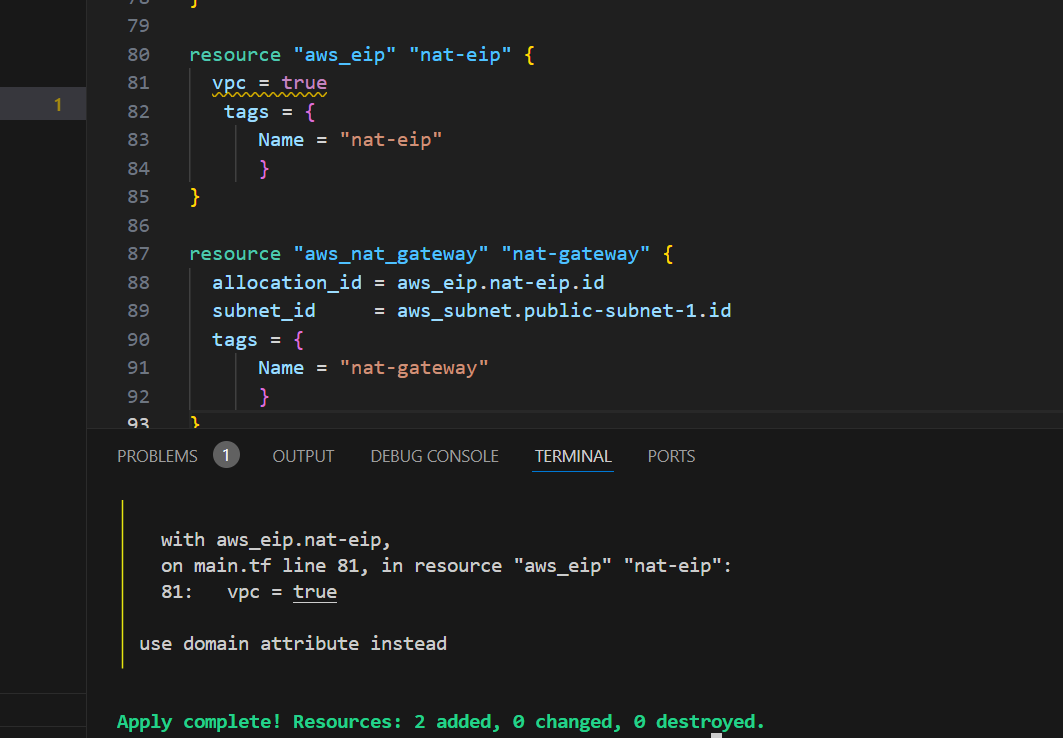
  subnet\_id     = aws\_subnet.public-subnet-1.id

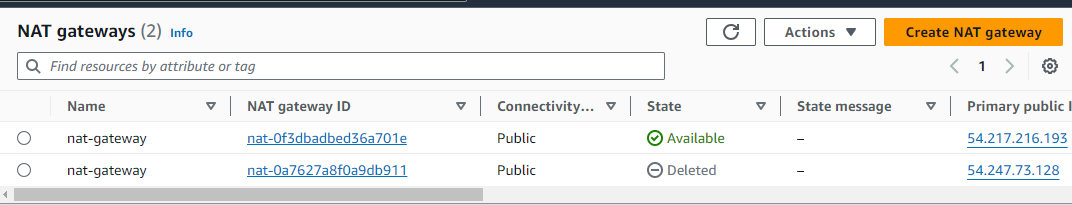
  tags = {

  Name = "nat-gateway"

      }

}





**SECURITY GROUPS**

Security group, which functions as a virtual firewall to regulate the inbound and outgoing traffic for AWS EC2 instances or other AWS resources in a VPC. We shall go over a security group’s definition and formation in this article.

**CODE:**

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"

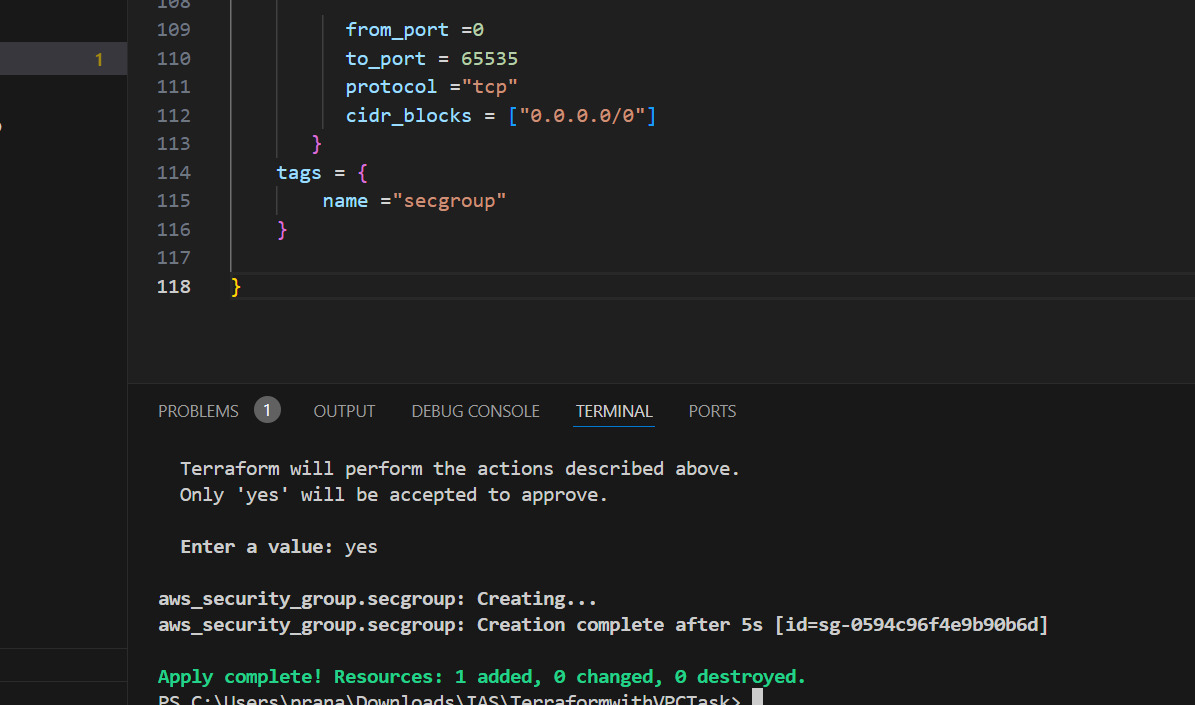
          cidr\_blocks = ["0.0.0.0/0"]

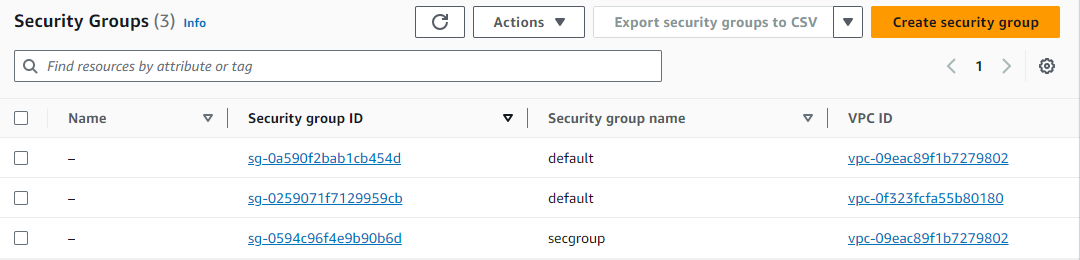
       }

    tags = {

        name ="secgroup"

    }





**PROCESS:**

Created vpc with region eu-west-1 with IP adddress 10.0.0.0/16 and named demo vpc.

Created a internet gateway for demo vpc.

Created 4 subnets; 2 private, 2 public –

* Private subnet-1, cidr 10.0.1.0/24, availability zone eu-west-1a
* Private subnet-2, cidr 10.0.2.0/24, availability zone eu-west-1b
* Public subnet-1, cidr 10.0.3.0/24, availability zone eu-west-1a
* Public subnet-1, cidr 10.0.4.0/24, availability zone eu-west-1b

Created route table for vpc named as public route table, connect to internet gateway through routes with cidr 0.0.0.0/0 and attaching public subnets to the route table.

Created NAT Gateway for the public subnet and named as nat gateway.

Created security group for vpc with inbound and outbound rules as port from port 0 to port 65535 and cidr blocks 0.0.0.0/0.

At the end we have allowed internet access to public subnets using IGW to private subnets using NGW.

**CONCLUSION:**

VPC provides a secure and flexible way to deploy applications in the cloud.

**THANKYOU**