Prerequisites

1. **Install Terraform**: Terraform Download and Install Guide.

   https://phoenixnap.com/kb/how-to-install-terraform

   Download link: https://www.terraform.io/

   For Windows:
   https://developer.hashicorp.com/terraform/install?product_intent=terraform#windows

2. **AWS CLI and IAM Role**: Ensure the AWS CLI is installed and configured with credentials (`aws configure`).

   ```
   $aws configure

       AWS Access Key ID [****************NONP]: XXXXX

       AWS Secret Access Key [****************aXx+]: YYYYYY

       Default region name [ca-central-1]: ZZZZZ

       Default output format [json]: csv/json
   ```

3. **Terraform AWS Provider**: Ensure the `AWS` provider block is correctly set up in your configuration.

   ```
   terraform {

           required_providers {

               aws = {

               source  = "hashicorp/aws"

               version = "5.81.0"

               }

           }

   }
   ```

```
# Configure the AWS Provider

provider "aws" {

}
```

Step-by-Step Guide to Create a VPC Using Terraform

---

1. Create a Terraform Configuration File

1. Create a working directory for your Terraform project:

   ```
   mkdir terraform-vpc
   cd terraform-vpc
   ```

2. Create a file named `main.tf` in this directory:

   ```
   touch main.tf
   ```

---

2. Define Provider Configuration

Add the AWS provider configuration in `main.tf`: (region=ca-central-1)

```
provider "aws" {
  region = "ca-central-1" # Specify your AWS region
}
```

---

3. Add VPC Resource

Define the VPC resource:

```
resource "aws_vpc" "devops_aws" {
  cidr_block           = "10.15.0.0/23"
  enable_dns_support   = true
  enable_dns_hostnames = true
```

```
  tags = {
    Name = "devops-aws"
  }
}
```

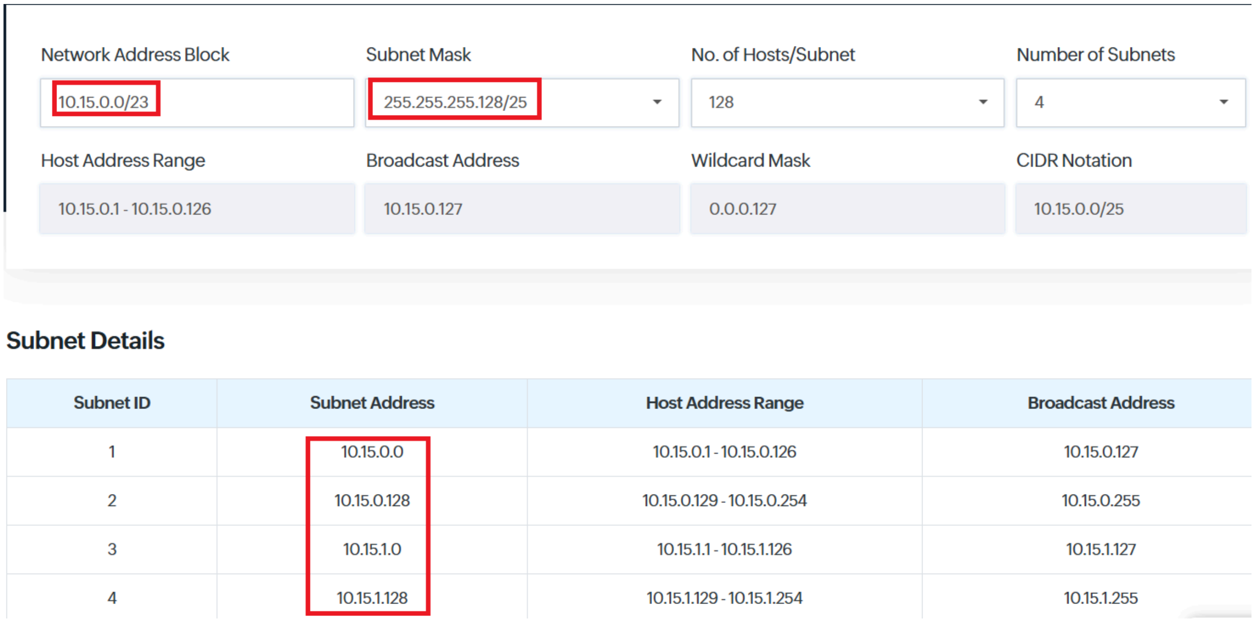
Explicitly enables:

- `enable_dns_support = true`: Enables DNS resolution within the VPC. DNS queries from instances will be resolved by Amazon Route 53.
- `enable_dns_hostnames = true`: Enables automatic assignment of DNS hostnames to instances. Each instance will be assigned a hostname that resolves to its private IP address.

---

4. Add Subnets : https://www.site24x7.com/tools/ipv4-subnetcalculator.html

Create public and private subnets: 4 subnets(2 public + 2 private) https://jodies.de/ipcalc

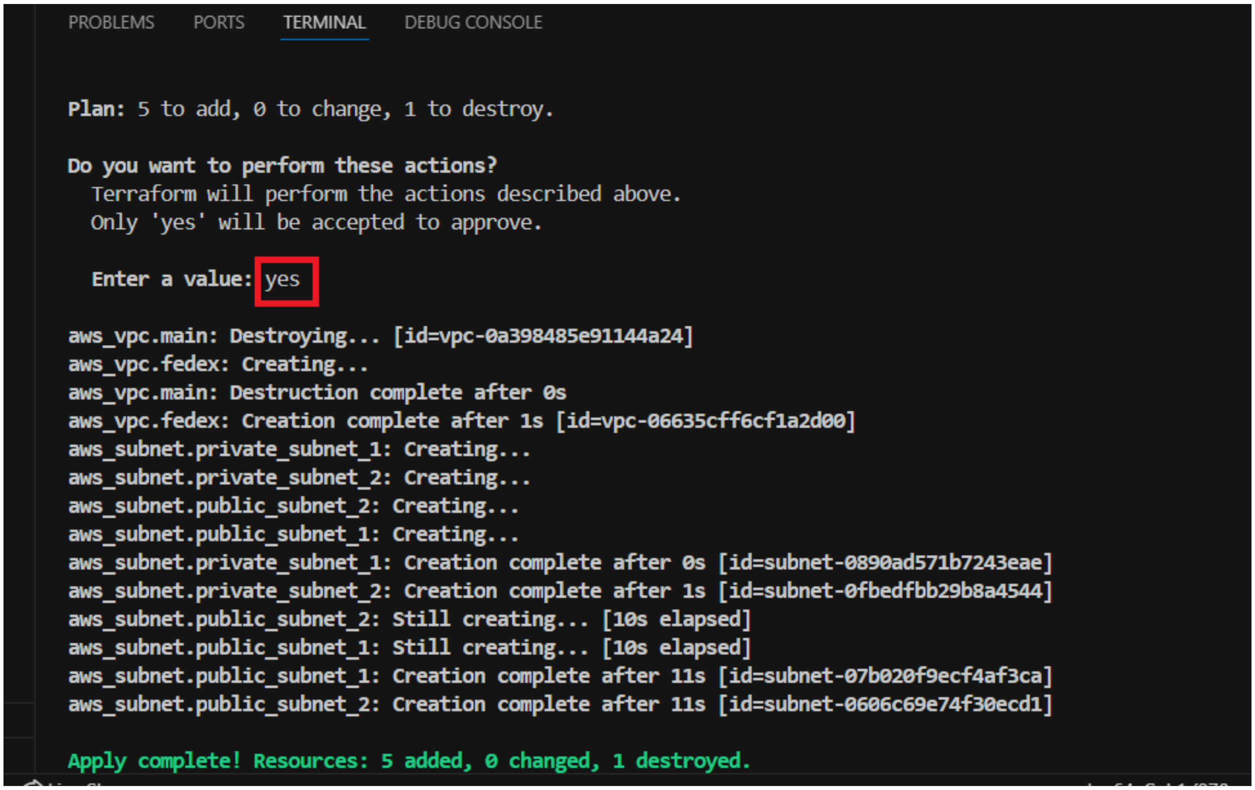


Network Address Block	Subnet Mask	No. of Hosts/Subnet	Number of Subnets
10.15.0.0/23	255.255.255.128/25	128	4
Host Address Range	Broadcast Address	Wildcard Mask	CIDR Notation
10.15.0.1 - 10.15.0.126	10.15.0.127	0.0.0.127	10.15.0.0/25

Subnet Details

Subnet ID	Subnet Address	Host Address Range	Broadcast Address
1	10.15.0.0	10.15.0.1 - 10.15.0.126	10.15.0.127
2	10.15.0.128	10.15.0.129 - 10.15.0.254	10.15.0.255
3	10.15.1.0	10.15.1.1 - 10.15.1.126	10.15.1.127
4	10.15.1.128	10.15.1.129 - 10.15.1.254	10.15.1.255

```
resource "aws_subnet" "public_subnet_1" {
  vpc_id                   = aws_vpc.devops_aws.id
  cidr_block               = "10.15.0.0/25"
  map_public_ip_on_launch = true
  availability_zone         = "ca-central-1a"
  tags = {
     Name = "PublicSubnet1-devops-aws"
  }
}

resource "aws_subnet" "private_subnet_1" {
  vpc_id         = aws_vpc.devops_aws.id
  cidr_block     = "10.15.0.128/25"
  availability_zone = "ca-central-1a"
  tags = {
     Name = "PrivateSubnet1-devops-aws"
  }
}
resource "aws_subnet" "public_subnet_2" {
  vpc_id                    = aws_vpc.devops_aws.id
  cidr_block                = "10.15.1.0/25"
  map_public_ip_on_launch = true
  availability_zone         = "ca-central-1b"
  tags = {
     Name = "PublicSubnet2-devops-aws"
  }
}

resource "aws_subnet" "private_subnet_2" {
  vpc_id         = aws_vpc.devops_aws.id
  cidr_block     = "10.15.1.128/25"
  availability_zone = "ca-central-1b"
  tags = {
     Name = "PrivateSubnet2-devops-aws"
  }
}
```



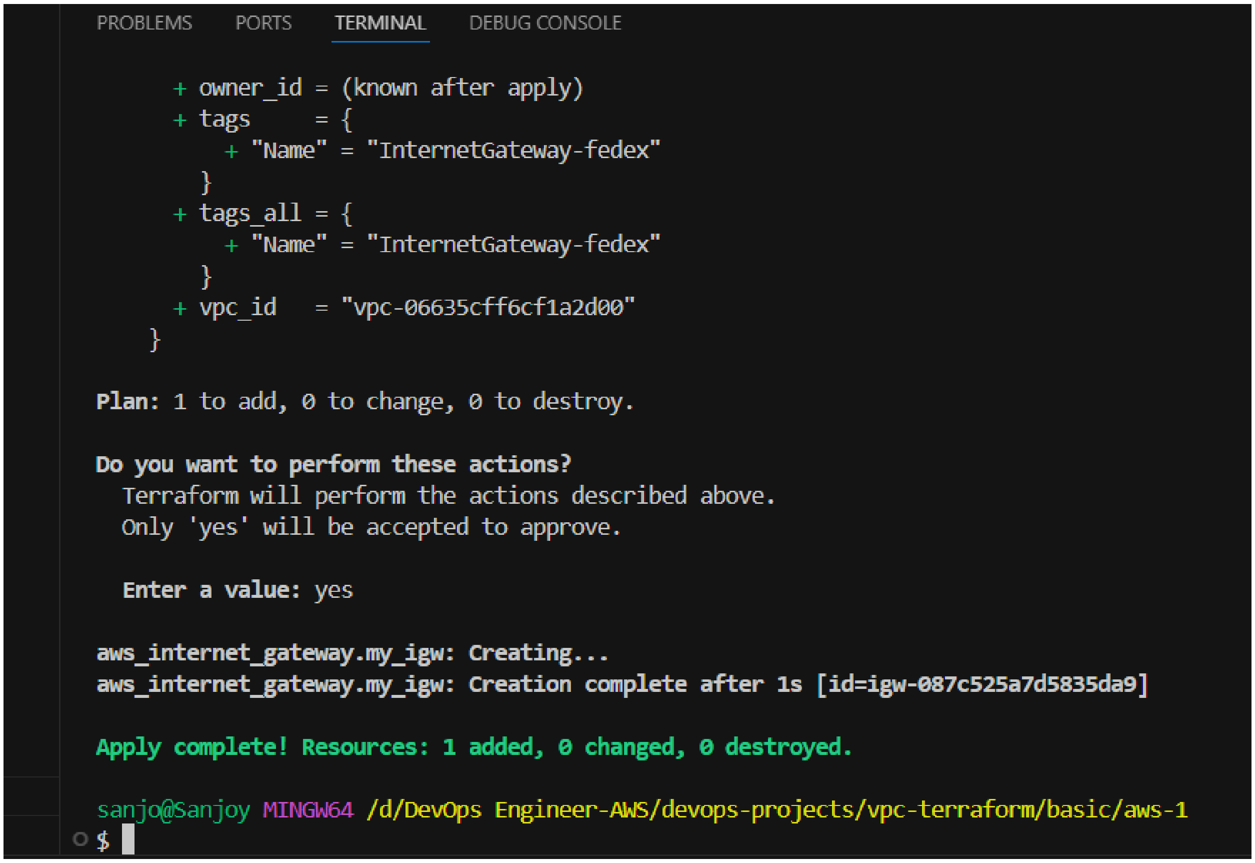
```
Plan: 5 to add, 0 to change, 1 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_vpc.main: Destroying... [id=vpc-0a398485e91144a24]
aws_vpc.fedex: Creating...
aws_vpc.main: Destruction complete after 0s
aws_vpc.fedex: Creation complete after 1s [id=vpc-06635cff6cf1a2d00]
aws_subnet.private_subnet_1: Creating...
aws_subnet.private_subnet_2: Creating...
aws_subnet.public_subnet_2: Creating...
aws_subnet.public_subnet_1: Creating...
aws_subnet.private_subnet_1: Creation complete after 0s [id=subnet-0890ad571b7243eae]
aws_subnet.private_subnet_2: Creation complete after 1s [id=subnet-0fbedfbb29b8a4544]
aws_subnet.public_subnet_2: Still creating... [10s elapsed]
aws_subnet.public_subnet_1: Still creating... [10s elapsed]
aws_subnet.public_subnet_1: Creation complete after 11s [id=subnet-07b020f9ecf4af3ca]
aws_subnet.public_subnet_2: Creation complete after 11s [id=subnet-0606c69e74f30ecd1]

Apply complete! Resources: 5 added, 0 changed, 1 destroyed.
```

5. Add an Internet Gateway

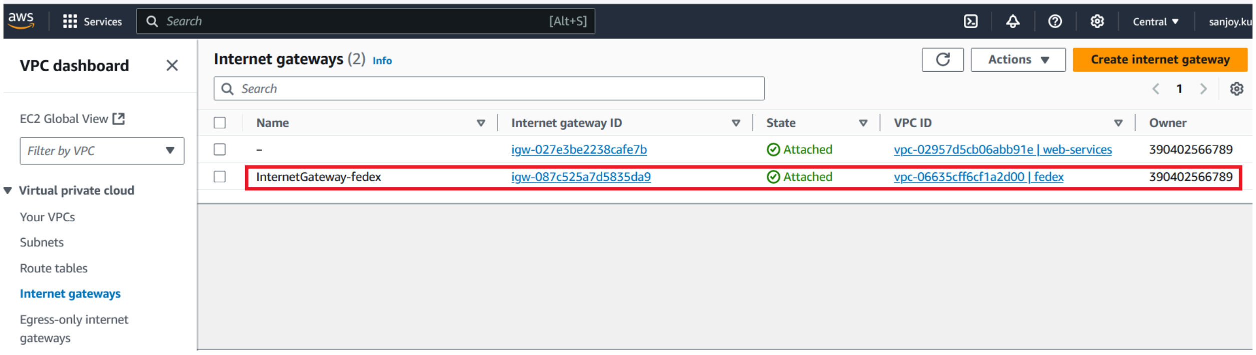
Define an Internet Gateway:

```
resource "aws_internet_gateway" "my_igw" {
  vpc_id = aws_vpc.devops_aws.id
  tags = {
    Name = "InternetGateway-devops-aws"
  }
}
```

$ terraform plan
$ terraform apply



```
      + owner_id = (known after apply)
      + tags     = {
          + "Name" = "InternetGateway-fedex"
        }
      + tags_all = {
          + "Name" = "InternetGateway-fedex"
        }
      + vpc_id   = "vpc-06635cff6cf1a2d00"
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_internet_gateway.my_igw: Creating...
aws_internet_gateway.my_igw: Creation complete after 1s [id=igw-087c525a7d5835da9]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

sanjo@Sanjoy MINGW64 /d/DevOps Engineer-AWS/devops-projects/vpc-terraform/basic/aws-1
$ 
```



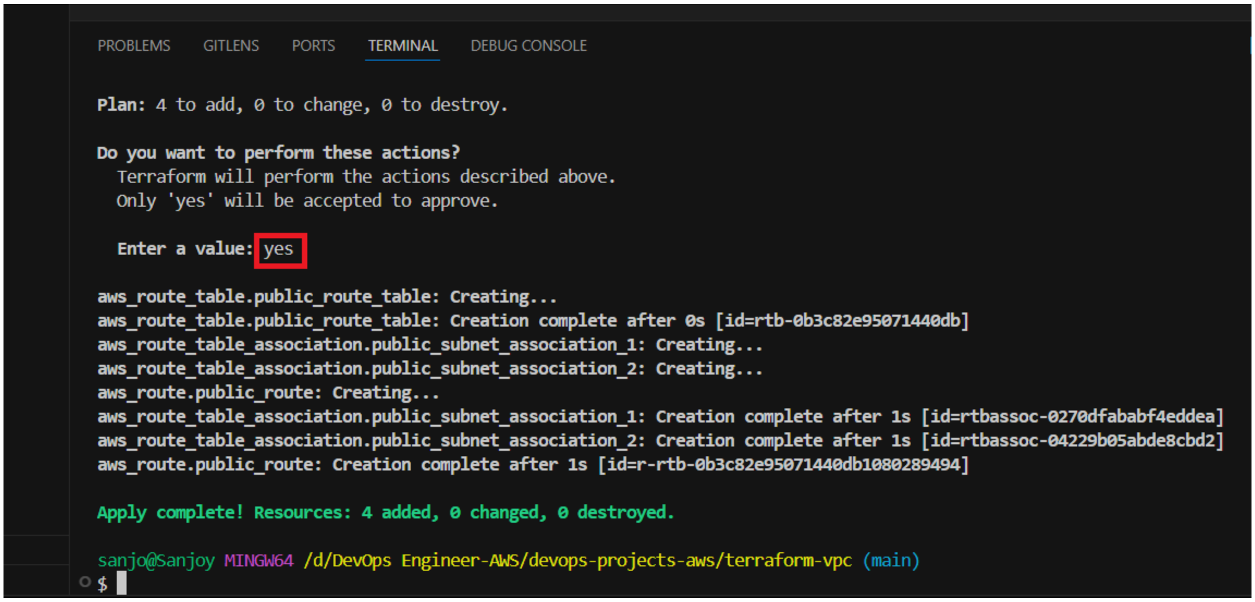
Internet gateways (2) Info

□	Name	Internet gateway ID	State	VPC ID	Owner
□	–	igw-027e3be2238cafe7b	✓ Attached	vpc-02957d5cb06abb91e | web-services	390402566789
□	InternetGateway-fedex	igw-087c525a7d5835da9	✓ Attached	vpc-06635cff6cf1a2d00 | fedex	390402566789

6. Create a Route Table for Public Subnets

```
resource "aws_route_table" "public_route_table" {

       vpc_id = aws_vpc.devops_aws.id

       tags = {

         Name = "PublicRouteTable-devops-aws"
```

```
        }
      }

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.my_igw.id
      }
resource "aws_route_table_association" "public_subnet_association_1" {
        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_association_2" {
        subnet_id      = aws_subnet.public_subnet_2.id
        route_table_id = aws_route_table.public_route_table.id
      }
```



```
Plan: 4 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_route_table.public_route_table: Creating...
aws_route_table.public_route_table: Creation complete after 0s [id=rtb-0b3c82e95071440db]
aws_route_table_association.public_subnet_association_1: Creating...
aws_route_table_association.public_subnet_association_2: Creating...
aws_route.public_route: Creating...
aws_route_table_association.public_subnet_association_1: Creation complete after 1s [id=rtbassoc-0270dfababf4eddea]
aws_route_table_association.public_subnet_association_2: Creation complete after 1s [id=rtbassoc-04229b05abde8cbd2]
aws_route.public_route: Creation complete after 1s [id=r-rtb-0b3c82e95071440db1080289494]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.

sanjo@Sanjoy MINGW64 /d/DevOps Engineer-AWS/devops-projects-aws/terraform-vpc (main)
$
```

7. Create a NAT Gateway for Private Subnets (Optional)

If you want to allow private subnets to access the internet (for updates or external resources), you'll need a NAT Gateway.

1. **Elastic IP for NAT Gateway:**

```
resource "aws_eip" "nat_eip" {
  domain = "vpc"
}
```

2. **NAT Gateway Resource:**

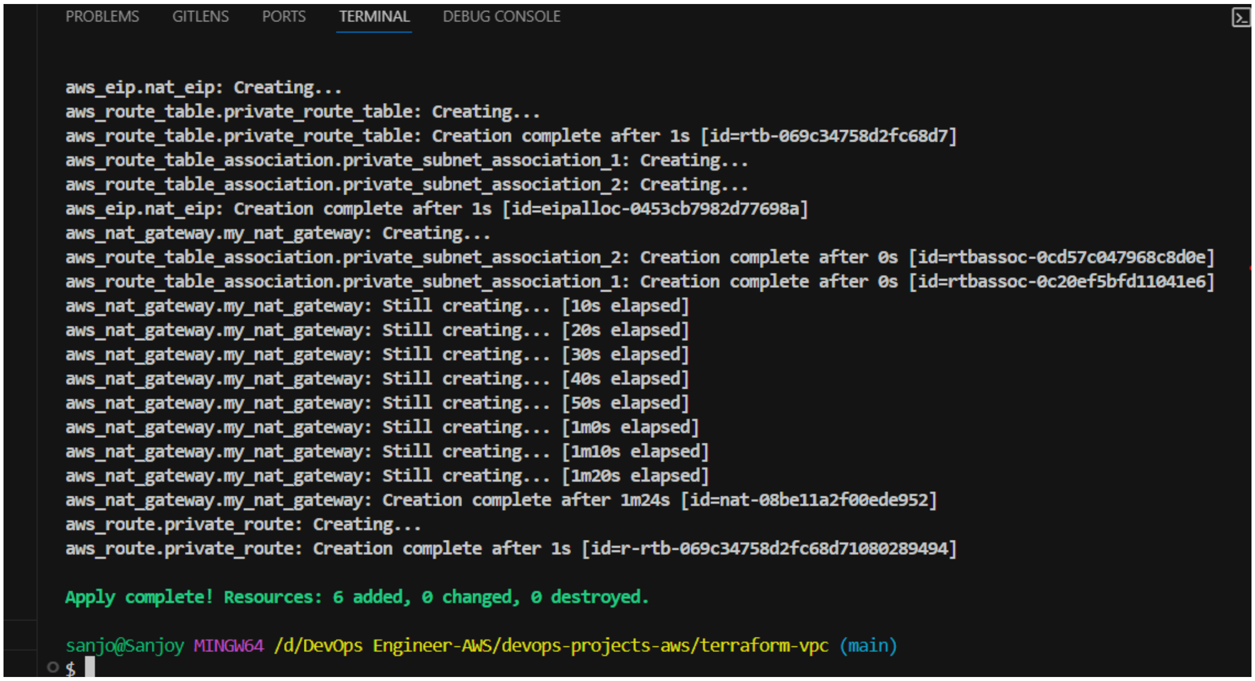
```
resource "aws_nat_gateway" "my_nat_gateway" {
  allocation_id = aws_eip.nat_eip.id
  subnet_id     = aws_subnet.public_subnet_1.id
  tags = {
    Name = "NATGateway-devops-aws"
  }
}
```

3. **Create a Route Table for Private Subnets**

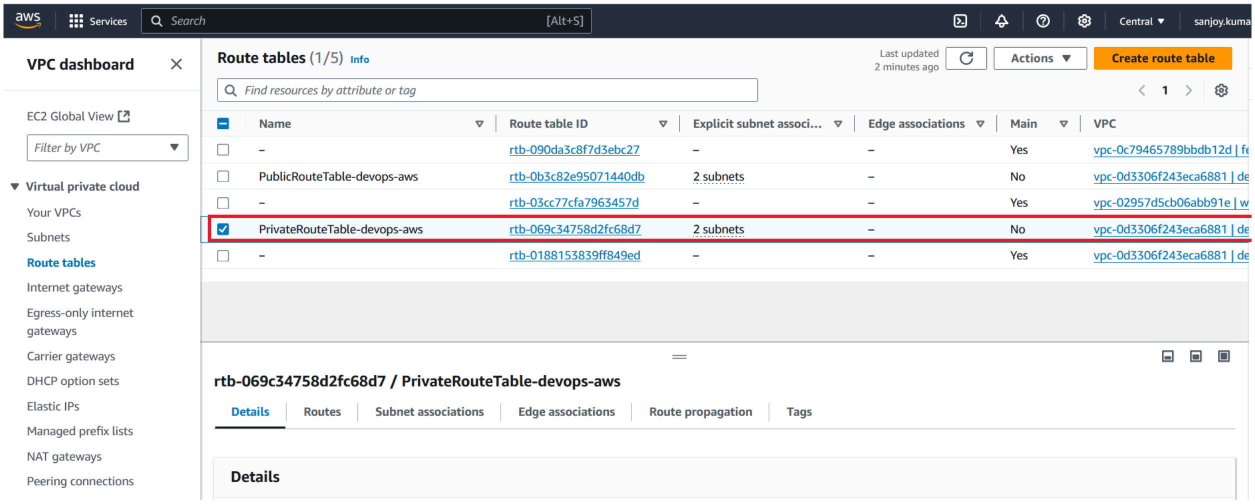
```
resource "aws_route_table" "private_route_table" {
  vpc_id = aws_vpc.devops_aws.id
  tags = {
    Name = "PrivateRouteTable-devops-aws"
  }
}


resource "aws_route" "private_route" {
  route_table_id         = aws_route_table.private_route_table.id
  destination_cidr_block = "0.0.0.0/0"
  nat_gateway_id         = aws_nat_gateway.my_nat_gateway.id
}
```

```
resource "aws_route_table_association" "private_subnet_association_1"
{
        subnet_id      = aws_subnet.private_subnet_1.id
        route_table_id = aws_route_table.private_route_table.id
}


resource "aws_route_table_association" "private_subnet_association_2"
{
  subnet_id      = aws_subnet.private_subnet_2.id
  route_table_id = aws_route_table.private_route_table.id

}
```

$terraform plan
$terraform apply



```
aws_eip.nat_eip: Creating...
aws_route_table.private_route_table: Creating...
aws_route_table.private_route_table: Creation complete after 1s [id=rtb-069c34758d2fc68d7]
aws_route_table_association.private_subnet_association_1: Creating...
aws_route_table_association.private_subnet_association_2: Creating...
aws_eip.nat_eip: Creation complete after 1s [id=eipalloc-0453cb7982d77698a]
aws_nat_gateway.my_nat_gateway: Creating...
aws_route_table_association.private_subnet_association_2: Creation complete after 0s [id=rtbassoc-0cd57c047968c8d0e]
aws_route_table_association.private_subnet_association_1: Creation complete after 0s [id=rtbassoc-0c20ef5bfd11041e6]
aws_nat_gateway.my_nat_gateway: Still creating... [10s elapsed]
aws_nat_gateway.my_nat_gateway: Still creating... [20s elapsed]
aws_nat_gateway.my_nat_gateway: Still creating... [30s elapsed]
aws_nat_gateway.my_nat_gateway: Still creating... [40s elapsed]
aws_nat_gateway.my_nat_gateway: Still creating... [50s elapsed]
aws_nat_gateway.my_nat_gateway: Still creating... [1m0s elapsed]
aws_nat_gateway.my_nat_gateway: Still creating... [1m10s elapsed]
aws_nat_gateway.my_nat_gateway: Still creating... [1m20s elapsed]
aws_nat_gateway.my_nat_gateway: Creation complete after 1m24s [id=nat-08be11a2f00ede952]
aws_route.private_route: Creating...
aws_route.private_route: Creation complete after 1s [id=r-rtb-069c34758d2fc68d71080289494]

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.

sanjo@Sanjoy MINGW64 /d/DevOps Engineer-AWS/devops-projects-aws/terraform-vpc (main)
$ 
```

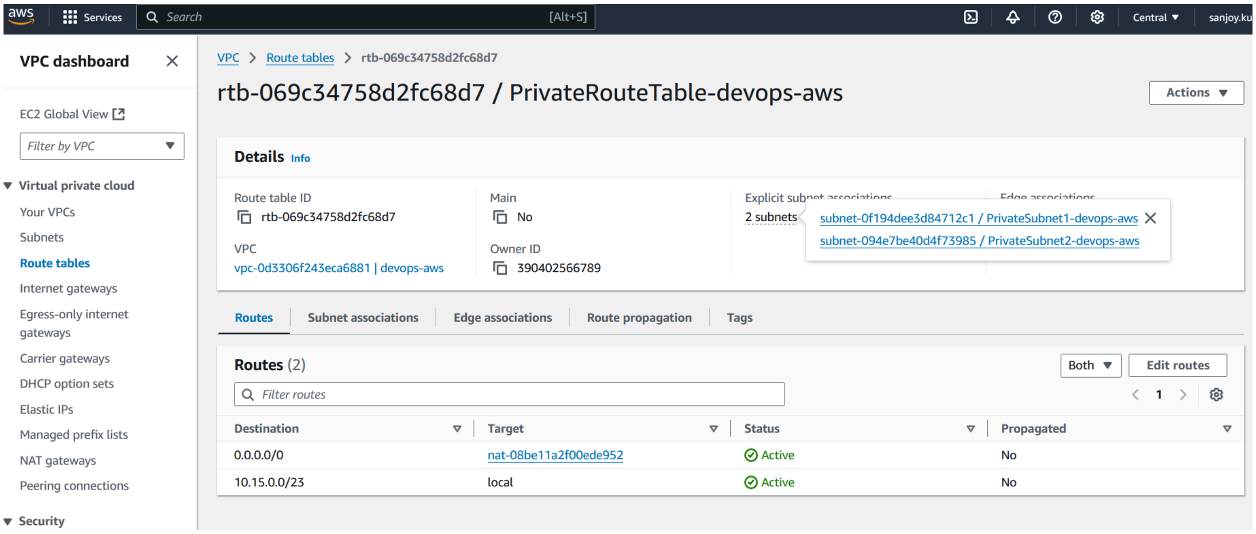


Route tables (1/5) Info

☐	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
☐	–	rtb-090da3c8f7d3ebc27	–	–	Yes	vpc-0c79465789bbdb12d | fe
☐	PublicRouteTable-devops-aws	rtb-0b3c82e95071440db	2 subnets	–	No	vpc-0d3306f243eca6881 | de
☐	–	rtb-03cc77cfa7963457d	–	–	Yes	vpc-02957d5cb06abb91e | w
☑	PrivateRouteTable-devops-aws	rtb-069c34758d2fc68d7	2 subnets	–	No	vpc-0d3306f243eca6881 | de
☐	–	rtb-0188153839ff849ed	–	–	Yes	vpc-0d3306f243eca6881 | de

rtb-069c34758d2fc68d7 / PrivateRouteTable-devops-aws

Details



VPC > Route tables > rtb-069c34758d2fc68d7

rtb-069c34758d2fc68d7 / PrivateRouteTable-devops-aws

Details

Route table ID: rtb-069c34758d2fc68d7

Main: No

Explicit subnet associations: 2 subnets

subnet-0f194dee3d84712c1 / PrivateSubnet1-devops-aws

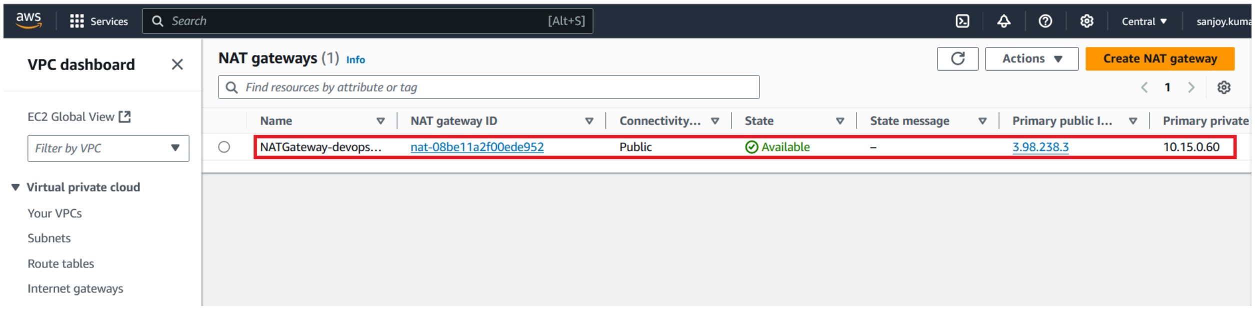
subnet-094e7be40d4f73985 / PrivateSubnet2-devops-aws

VPC: vpc-0d3306f243eca6881 | devops-aws

Owner ID: 390402566789

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	nat-08be11a2f00ede952	Active	No
10.15.0.0/23	local	Active	No



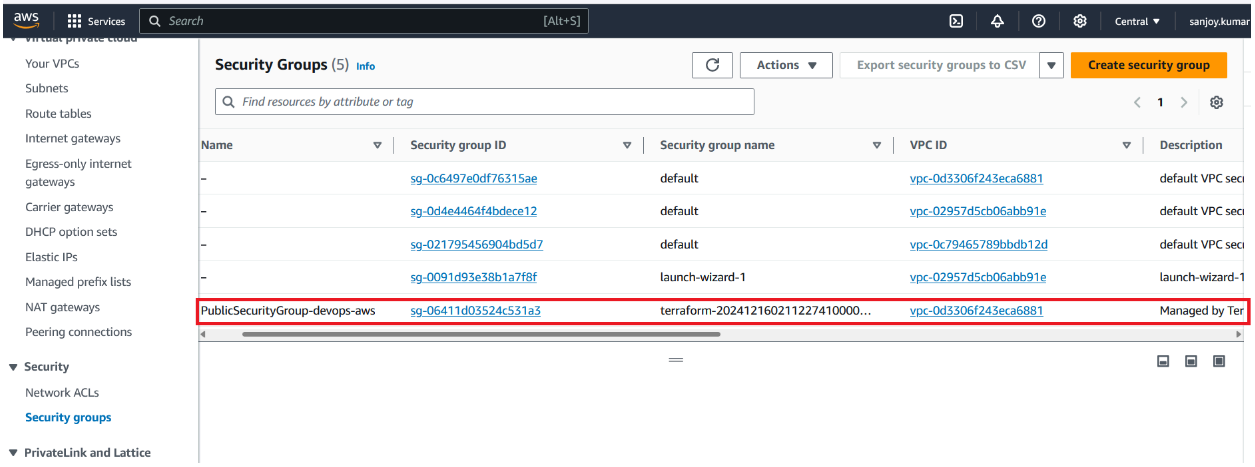
NAT gateways (1)

Name	NAT gateway ID	Connectivity...	State	State message	Primary public I...	Primary private
NATGateway-devops...	nat-08be11a2f00ede952	Public	Available	–	3.98.238.3	10.15.0.60

8. Add Security Groups (Optional)

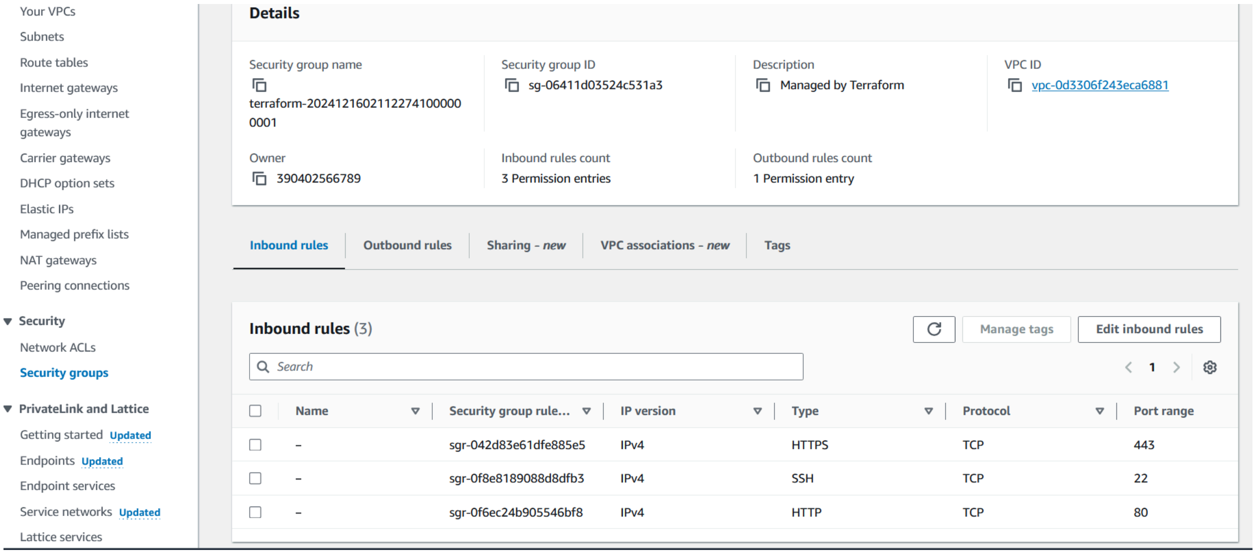
Define a security group allowing SSH, HTTP, and HTTPS access:

```
resource "aws_security_group" "public_sg" {
  vpc_id = aws_vpc.devops_aws.id
  tags = {
    Name = "PublicSecurityGroup-devops-aws"
  }

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

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

  ingress {
    from_port   = 443
    to_port     = 443
    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"]
  }
}
```



Security Groups (5)

Name	Security group ID	Security group name	VPC ID	Description
–	sg-0c6497e0df76315ae	default	vpc-0d3306f243eca6881	default VPC secu
–	sg-0d4e4464f4bdece12	default	vpc-02957d5cb06abb91e	default VPC secu
–	sg-021795456904bd5d7	default	vpc-0c79465789bbdb12d	default VPC secu
–	sg-0091d93e38b1a7f8f	launch-wizard-1	vpc-02957d5cb06abb91e	launch-wizard-1
PublicSecurityGroup-devops-aws	sg-06411d03524c531a3	terraform-202412160211227410000...	vpc-0d3306f243eca6881	Managed by Ter



Details

Security group name: terraform-20241216021122741000000001

Security group ID: sg-06411d03524c531a3

Description: Managed by Terraform

VPC ID: vpc-0d3306f243eca6881

Owner: 390402566789

Inbound rules count: 3 Permission entries

Outbound rules count: 1 Permission entry

Inbound rules (3)

□	Name	Security group rule...	IP version	Type	Protocol	Port range
□	-	sgr-042d83e61dfe885e5	IPv4	HTTPS	TCP	443
□	-	sgr-0f8e8189088d8dfb3	IPv4	SSH	TCP	22
□	-	sgr-0f6ec24b905546bf8	IPv4	HTTP	TCP	80

---

9. Initialize Terraform

1. Initialize the Terraform configuration:

```
terraform init
```

---

10. Plan the Infrastructure

1. Preview the planned changes:

```
terraform plan
```

---

11. Apply the Configuration

1. Deploy the VPC:
   ```
   terraform apply
   ```
2. Type `yes` when prompted to confirm the changes.

---

12. Verify

Once Terraform applies the configuration, you should have:

- A VPC with CIDR block `10.15.0.0/23`.

- Two public subnets (`10.15.0.0/25`, `10.15.1.0/25`).
- Two private subnets (`10.15.0.128/25`, `10.15.1.128/25`).
- An Internet Gateway attached to the VPC.
- NAT Gateway set up for private subnets.
- Route tables and associations configured.

Final Steps

1. Verify the VPC in the AWS Management Console under **VPC**.
2. If needed, you can destroy the created infrastructure:

```
terraform destroy
```

Your VPC is now created using Terraform!