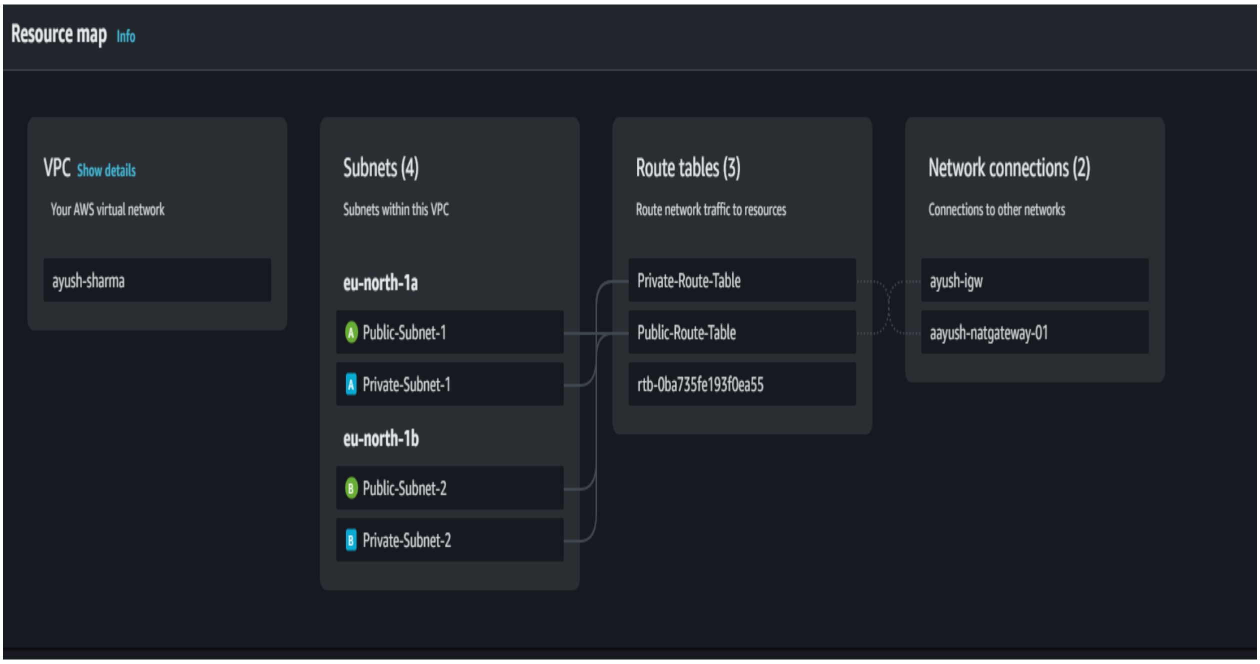
Setting up VPC infrastructure

Step 1: Create the VPC

1. **Go to the VPC Dashboard** in the AWS Management Console.
2. Click **Create VPC**.
3. **VPC Name**: my-vpc.
4. **IPv4 CIDR block**: 10.0.0.0/22 (This will define the address range for your entire VPC).
5. Leave the rest as default and click **Create VPC**.



Step 2: Create Subnets

Create 4 subnets, 2 public and 2 private, with different CIDR blocks and availability zones:

1. Go to **Subnets** under the **VPC Dashboard** and click **Create subnet**.
2. **Subnet 1 (Public Subnet 1)**:
   - **Name tag**: public-subnet-1
   - **VPC**: my-vpc
   - **Availability Zone**: us-east-1a
   - **IPv4 CIDR block**: 10.0.1.0/24
3. **Subnet 2 (Public Subnet 2)**:

   - **Name tag**: public-subnet-2
   - **VPC**: my-vpc
   - **Availability Zone**: us-east-1b
   - **IPv4 CIDR block**: 10.0.2.0/24

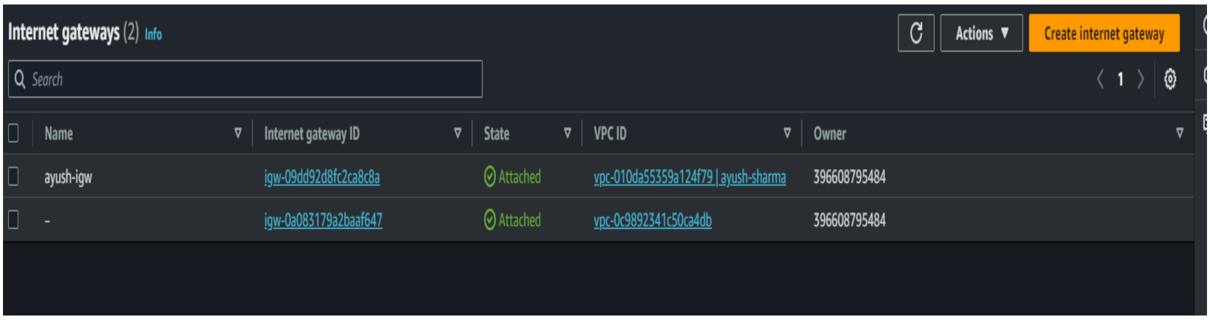
4. **Subnet 3 (Private Subnet 1)**:
   - **Name tag**: private-subnet-1
   - **VPC**: my-vpc
   - **Availability Zone**: us-east-1a
   - **IPv4 CIDR block**: 10.0.3.0/24

5. **Subnet 4 (Private Subnet 2)**:
   - **Name tag**: private-subnet-2
   - **VPC**: my-vpc
   - **Availability Zone**: us-east-1b
   - **IPv4 CIDR block**: 10.0.4.0/24

Click **Create Subnet** after filling in each subnet's details.

Step 3: Create and Attach the Internet Gateway

1. Go to **Internet Gateways** in the **VPC Dashboard**.
2. Click **Create Internet Gateway**.
   - **Name tag**: my-igw.
3. Click **Create Internet Gateway**.
4. **Attach the Internet Gateway** to the VPC:
   - Select my-igw, click **Actions** and select **Attach to VPC**.
   - Choose my-vpc and click **Attach Internet Gateway**.



Internet gateways (2)

□	Name	Internet gateway ID	State	VPC ID	Owner
□	ayush-igw	igw-09dd92d8fc2ca8c8a	✓ Attached	vpc-010da55359a124f79 | ayush-sharma	396608795484
□	-	igw-0a083179a2baaf647	✓ Attached	vpc-0c9892341c50ca4db	396608795484

Step 4: Create the NAT Gateway for Private Subnets

1. Go to **NAT Gateways** in the **VPC Dashboard**.
2. Click **Create NAT Gateway**.
3. **Subnet**: Select public-subnet-1.
4. **Elastic IP**: Click **Allocate Elastic IP** and choose it.
5. Click **Create NAT Gateway**.
   - This allows instances in private subnets to communicate with the internet for outbound traffic.

Step 5: Create Route Tables

Public Route Table (For Public Subnets)

1. Go to **Route Tables** in the **VPC Dashboard**.
2. Click **Create Route Table**.
   - **Name tag**: public-route-table.
   - **VPC**: Select my-vpc.
3. Click **Create**.
4. **Add a route to the Internet Gateway**:
   - Select public-route-table.
   - Click **Routes**, then **Edit Routes**.
   - Add a route with:
     - **Destination**: 0.0.0.0/0
     - **Target**: Select your **Internet Gateway (igw-xxxxxx)**.
   - Click **Save Routes**.
5. **Associate public subnets** with the public route table:
   - Go to the **Subnet Associations** tab.
   - Click **Edit Subnet Associations**.
   - Select both public-subnet-1 and public-subnet-2.
   - Click **Save Associations**.

Private Route Table (For Private Subnets)

1. Click **Create Route Table**.
   - **Name tag**: private-route-table.

   - **VPC**: Select my-vpc.
2. Click **Create**.
3. **Add a route to the NAT Gateway**:
   - Select private-route-table.
   - Click **Routes**, then **Edit Routes**.
   - Add a route with:
     - **Destination**: 0.0.0.0/0
     - **Target**: Select your **NAT Gateway (nat-xxxxxx)**.
   - Click **Save Routes**.
4. **Associate private subnets** with the private route table:
   - Go to the **Subnet Associations** tab.
   - Click **Edit Subnet Associations**.
   - Select both private-subnet-1 and private-subnet-2.
   - Click **Save Associations**.

Step 6: Launch EC2 Instances

Public EC2 Instance in Public Subnet

1. Go to the **EC2 Dashboard** and click **Launch Instance**.
2. **Name**: Public-Instance.
3. **AMI**: Select an Amazon Linux 2 AMI or your preferred AMI.
4. **Instance Type**: t2.micro (or any type you prefer).
5. **Key Pair**: Choose or create a new key pair.
6. **Network Settings**:
   - **VPC**: Select my-vpc.
   - **Subnet**: Select public-subnet-1.
   - **Auto-assign Public IP**: Ensure this is **enabled**.
7. Click **Launch Instance**.

Private EC2 Instances in Private Subnets

1. Go to the **EC2 Dashboard** and click **Launch Instance**.
2. **Name**: Private-Instance-1.
3. **AMI**: Select an Amazon Linux 2 AMI or your preferred AMI.

4. **Instance Type**: t2.micro (or any type you prefer).
5. **Key Pair**: Choose or create a new key pair.
6. **Network Settings**:
   - **VPC**: Select my-vpc.
   - **Subnet**: Select private-subnet-1.
   - **Auto-assign Public IP**: Ensure this is **disabled**.
7. Click **Launch Instance**.

Repeat these steps to create **Private-Instance-2** in private-subnet-2.

Summary of Network Setup:

- **VPC CIDR Block**: 10.0.0.0/16
- **Subnets**:
  - **Public Subnet 1**: 10.0.1.0/24 in us-east-1a
  - **Public Subnet 2**: 10.0.2.0/24 in us-east-1b
  - **Private Subnet 1**: 10.0.3.0/24 in us-east-1a
  - **Private Subnet 2**: 10.0.4.0/24 in us-east-1b
- **Route Tables**:
  - **Public Route Table** with route 0.0.0.0/0 to the **Internet Gateway (IGW)**.
  - **Private Route Table** with route 0.0.0.0/0 to the **NAT Gateway**.
- **Gateways**:
  - **Internet Gateway (IGW)** for public subnets.
  - **NAT Gateway** for private subnets.

Conclusion

This setup establishes a secure 3-tier architecture on AWS, with public and private subnets configured appropriately, and includes the use of a NAT Gateway for outbound internet access from private subnets. Ensure that your security groups and network ACLs are configured correctly to allow necessary traffic between layers while maintaining security.