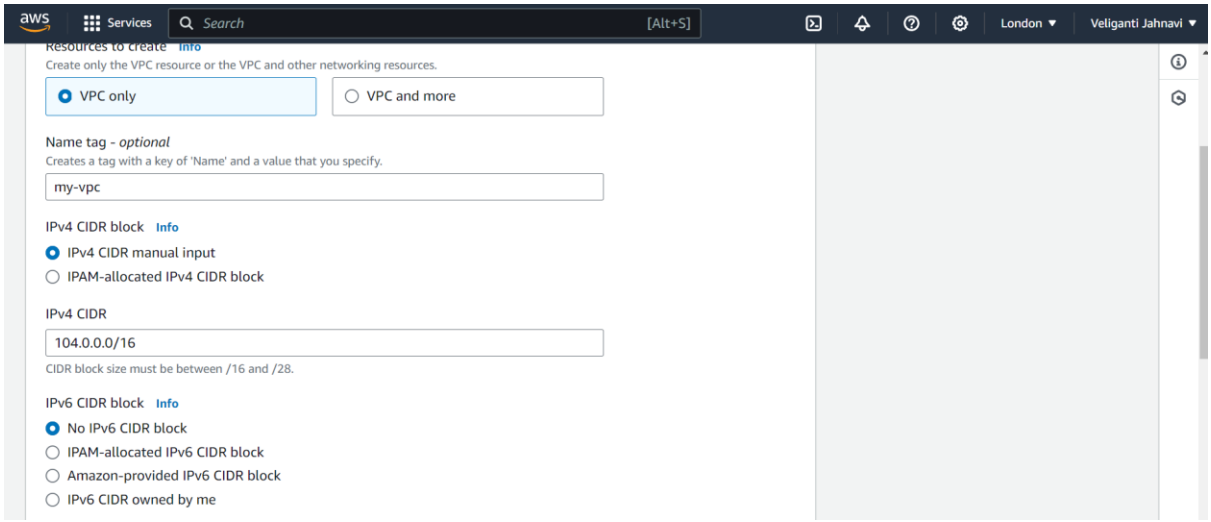
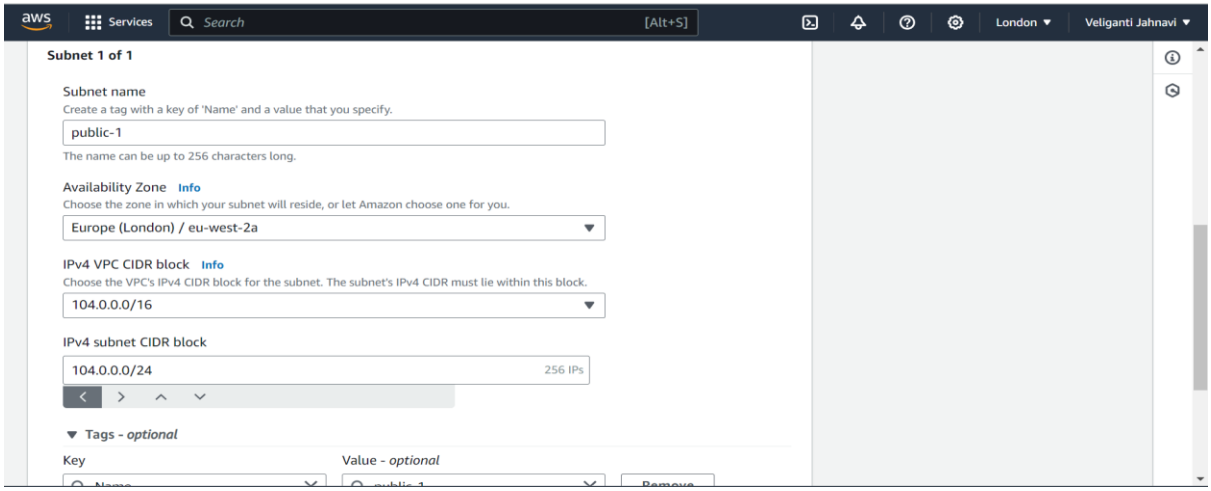
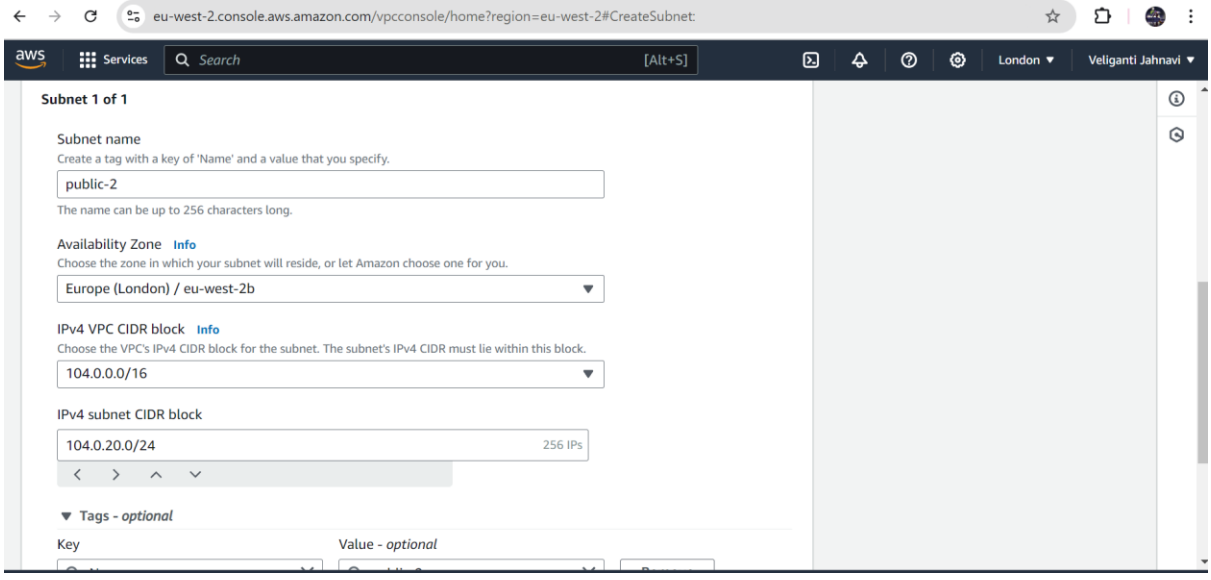
3-Tier Architecture.

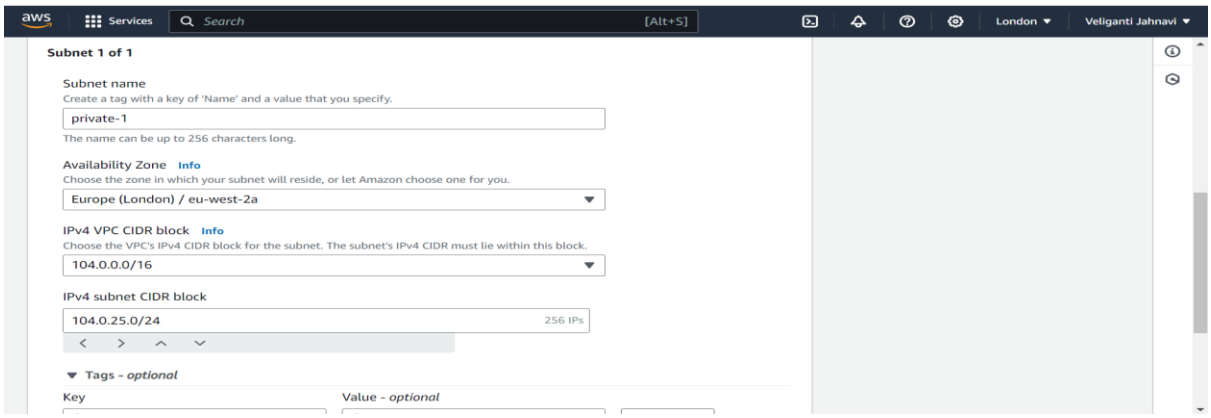
Step 1: Choose the region (London)→Create a VPC.

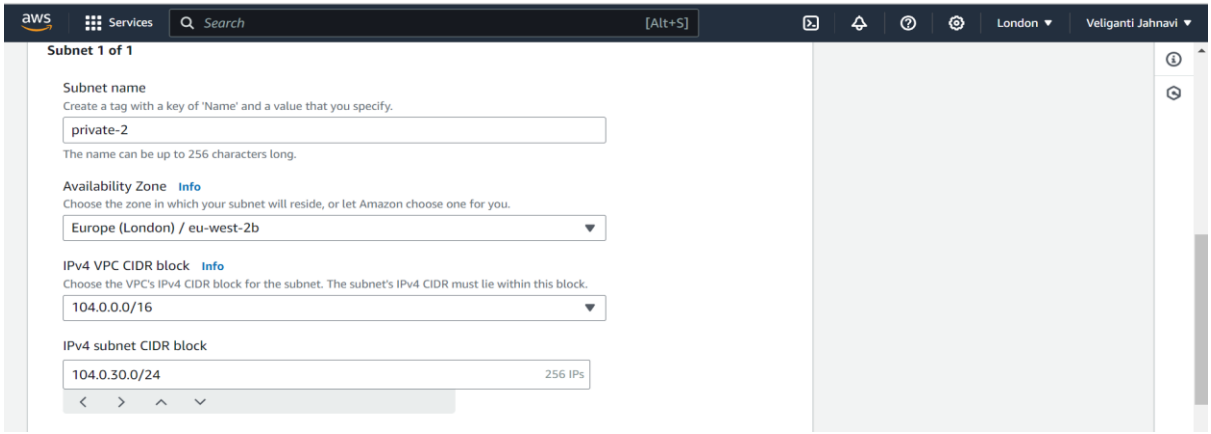


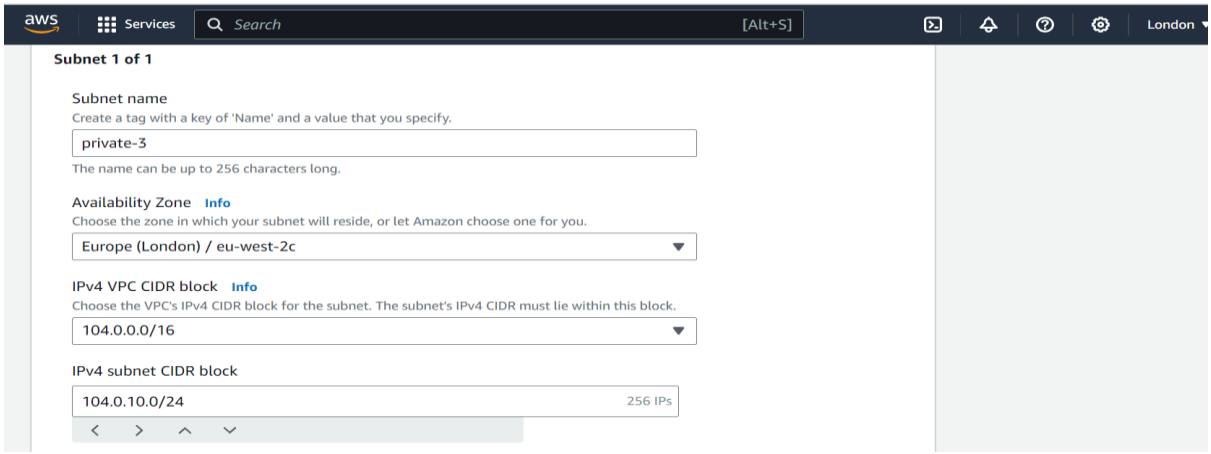
Step 2: Create 2 public(2a,2b) subnets and 4(2a,2b,2c,2d) private subnets.

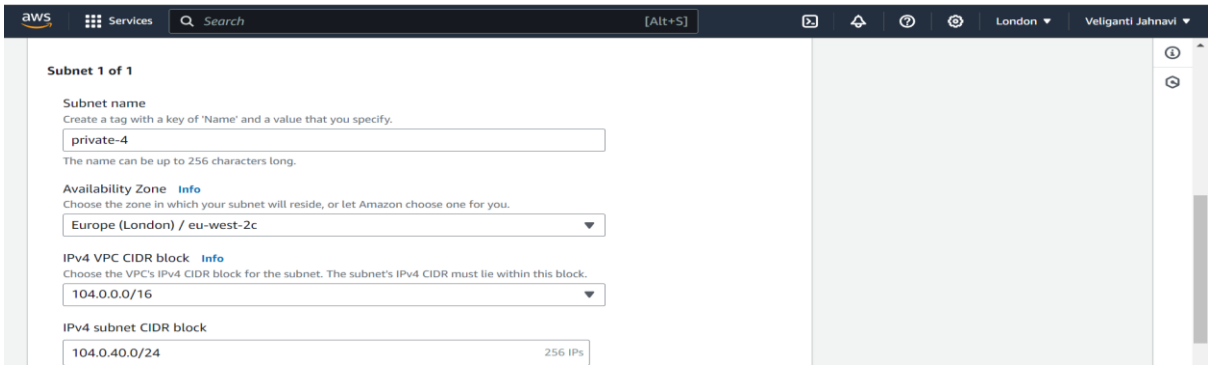


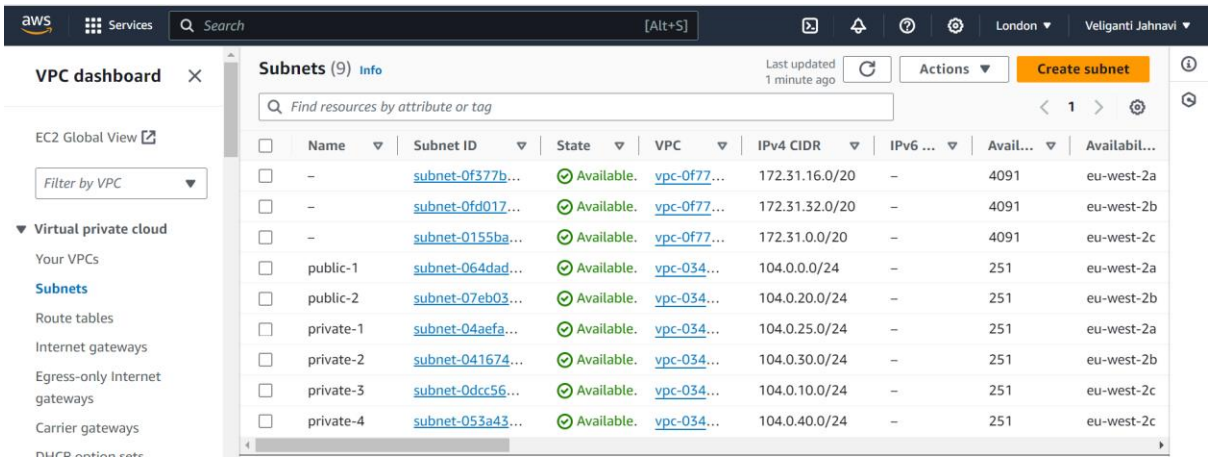



aws
Services
Search
[Alt+S]
London
Veliganti Jahnavi
Subnet 1 of 1
Subnet name
Create a tag with a key of 'Name' and a value that you specify.
private-1
The name can be up to 256 characters long.
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
Europe (London) / eu-west-2a
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
104.0.0.0/16
IPv4 subnet CIDR block
104.0.25.0/24
256 IPs
Tags - optional
Key
Value - optional


aws
Services
Search
[Alt+S]
London
Veliganti Jahnavi
Subnet 1 of 1
Subnet name
Create a tag with a key of 'Name' and a value that you specify.
private-2
The name can be up to 256 characters long.
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
Europe (London) / eu-west-2b
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
104.0.0.0/16
IPv4 subnet CIDR block
104.0.30.0/24
256 IPs

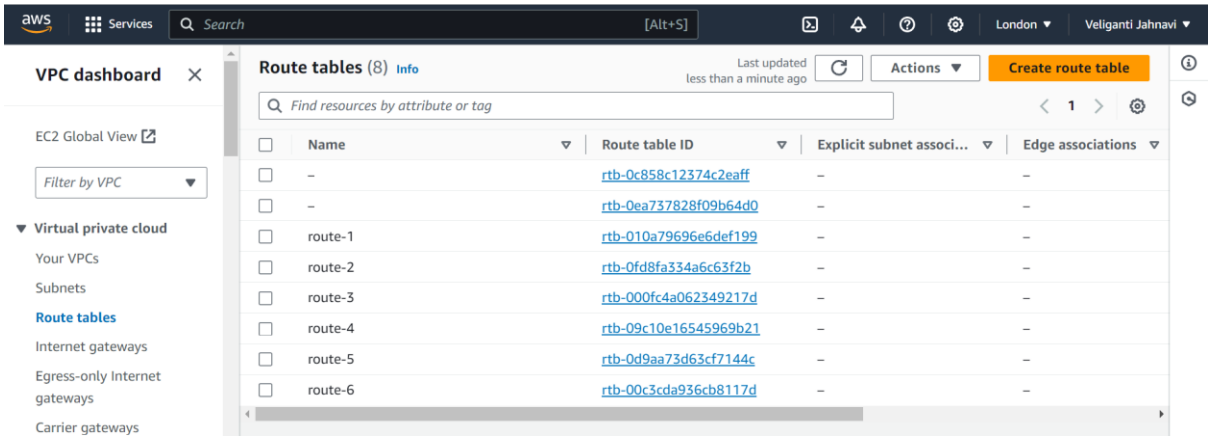

aws
Services
Search
[Alt+S]
London
Subnet 1 of 1
Subnet name
Create a tag with a key of 'Name' and a value that you specify.
private-3
The name can be up to 256 characters long.
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
Europe (London) / eu-west-2c
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
104.0.0.0/16
IPv4 subnet CIDR block
104.0.10.0/24
256 IPs


aws
Services
Search
[Alt+S]
London
Veliganti Jahnavi
Subnet 1 of 1
Subnet name
Create a tag with a key of 'Name' and a value that you specify.
private-4
The name can be up to 256 characters long.
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
Europe (London) / eu-west-2c
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
104.0.0.0/16
IPv4 subnet CIDR block
104.0.40.0/24
256 IPs

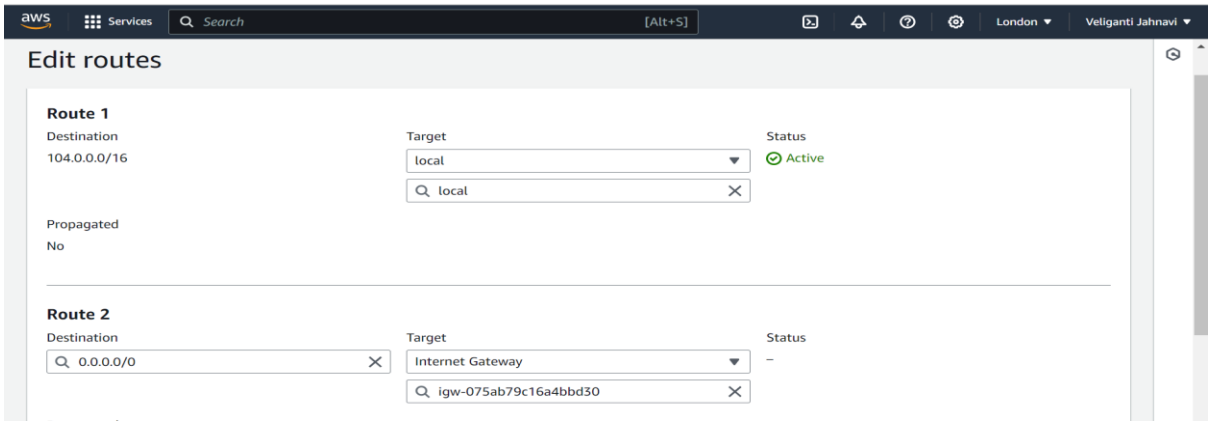


Step 3: Create an Internet Gateway and attach it to VPC.

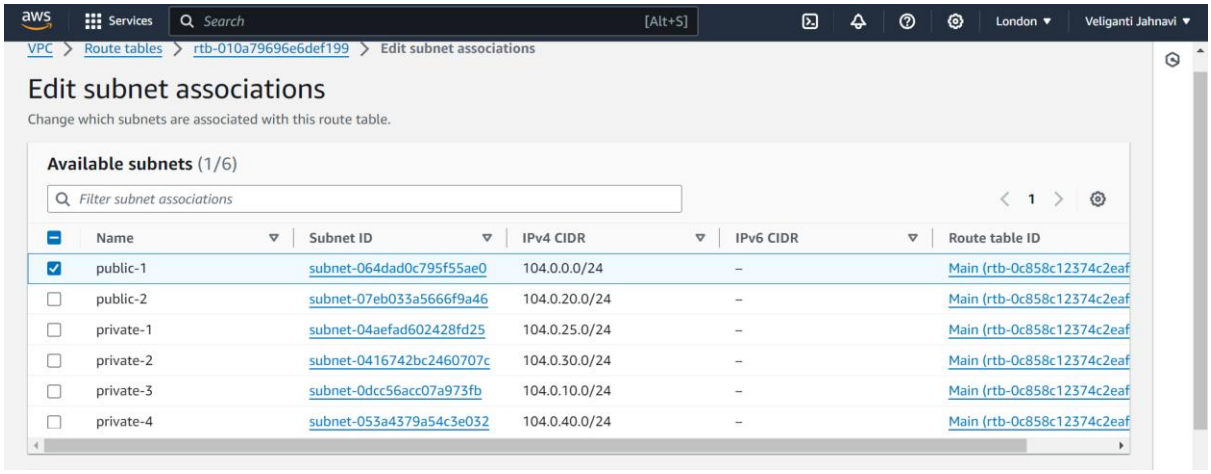
Step 4: Create Route tables.



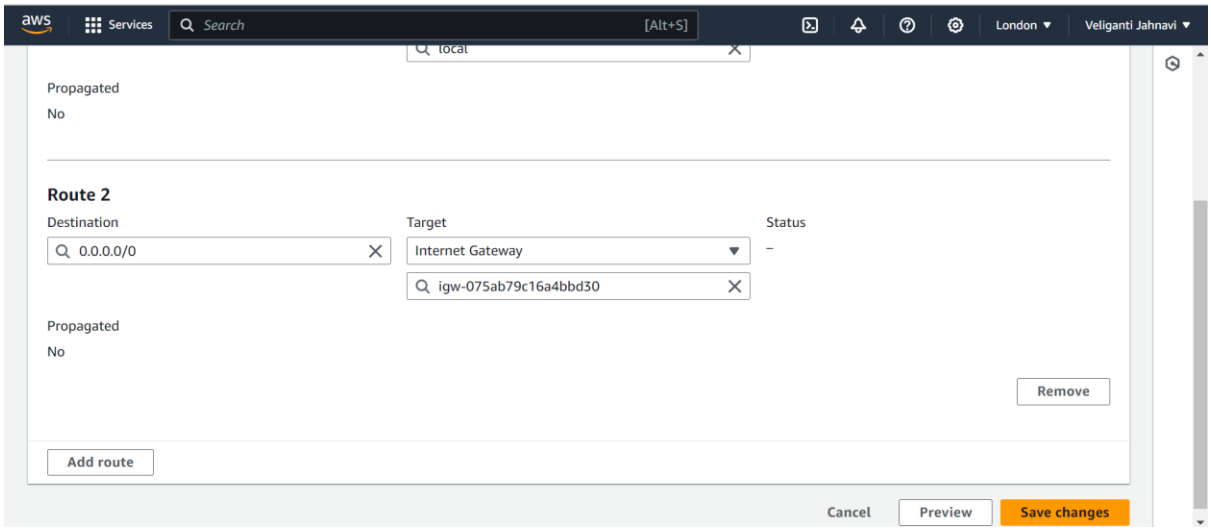
Step 6: Go to route 1 →Edit routes→Attach to IGW.



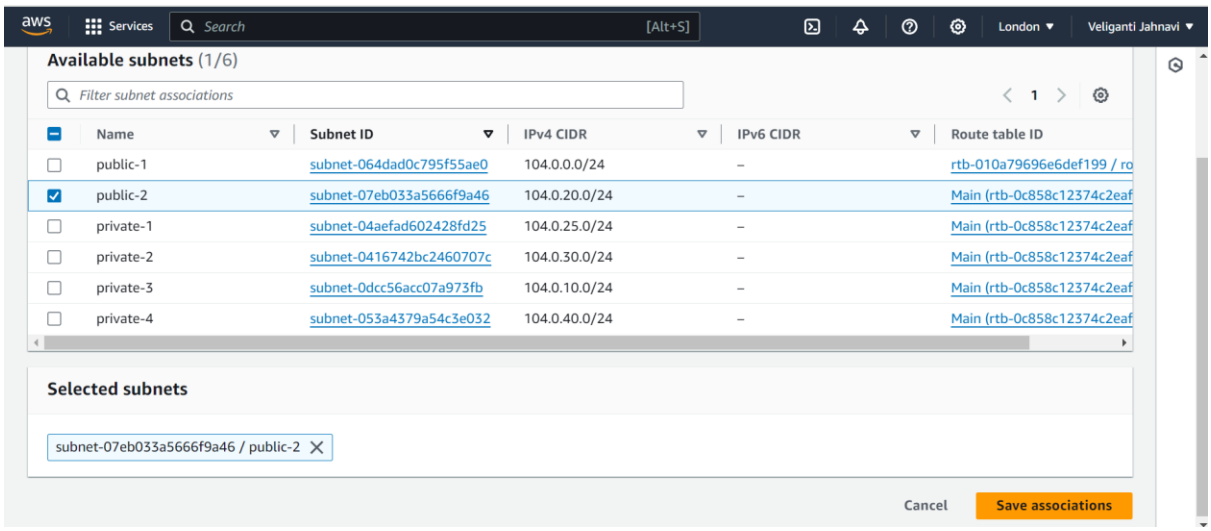
Here edit the subnet association, select public-1, and save.



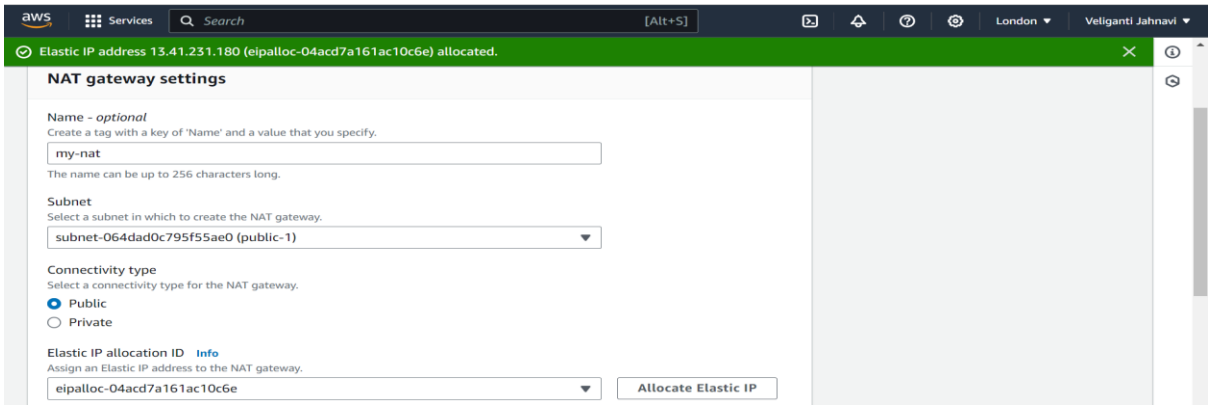
Step 6: Go to route 2 →Edit routes→Attach to IGW.

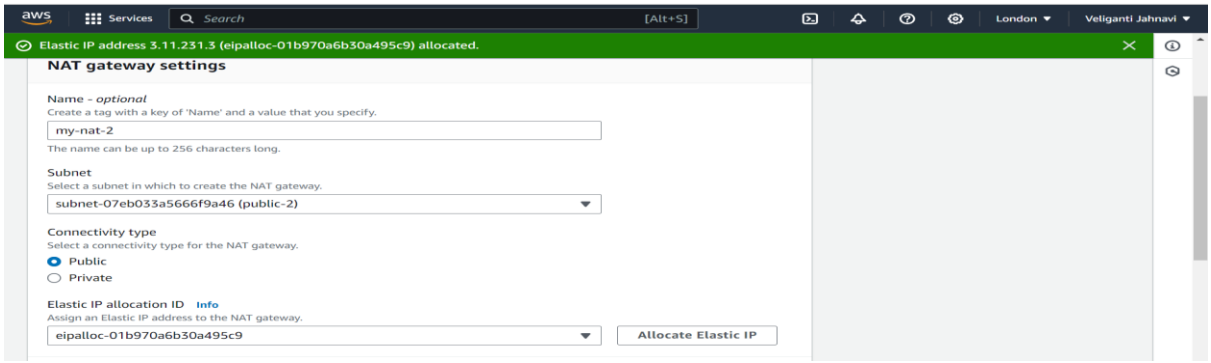


Here edit the subnet association, select public-2, and save.



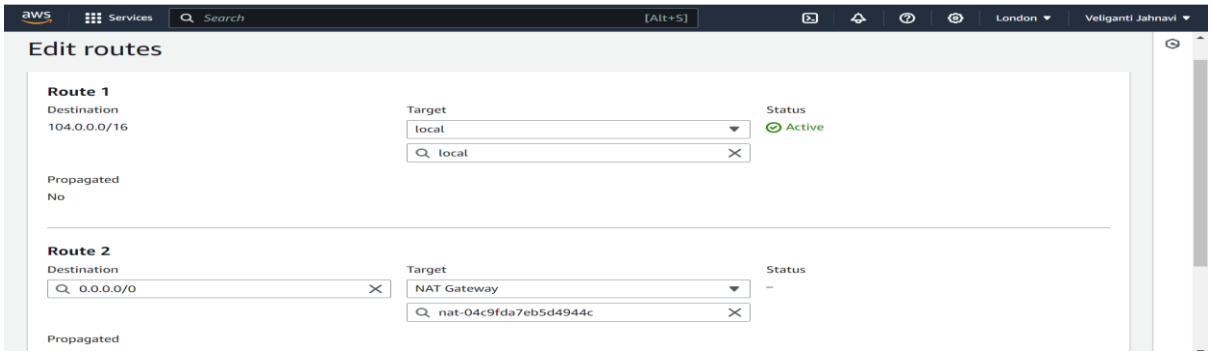
Step 7: Create 2 NAT Gateways (public1, public2).



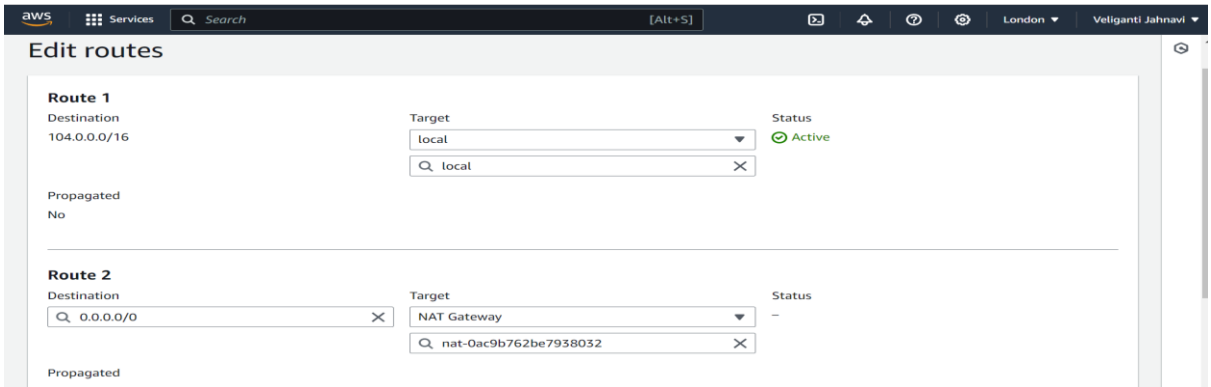


Go to the route table→edit route→ Add NAT Gateway→[Route (3,5)-NAT (1)]→Route (4,6)-NAT (2)→edit subnet association and tick and save[route(3,4,5,6)→subnet(private-1,2,3,4)].

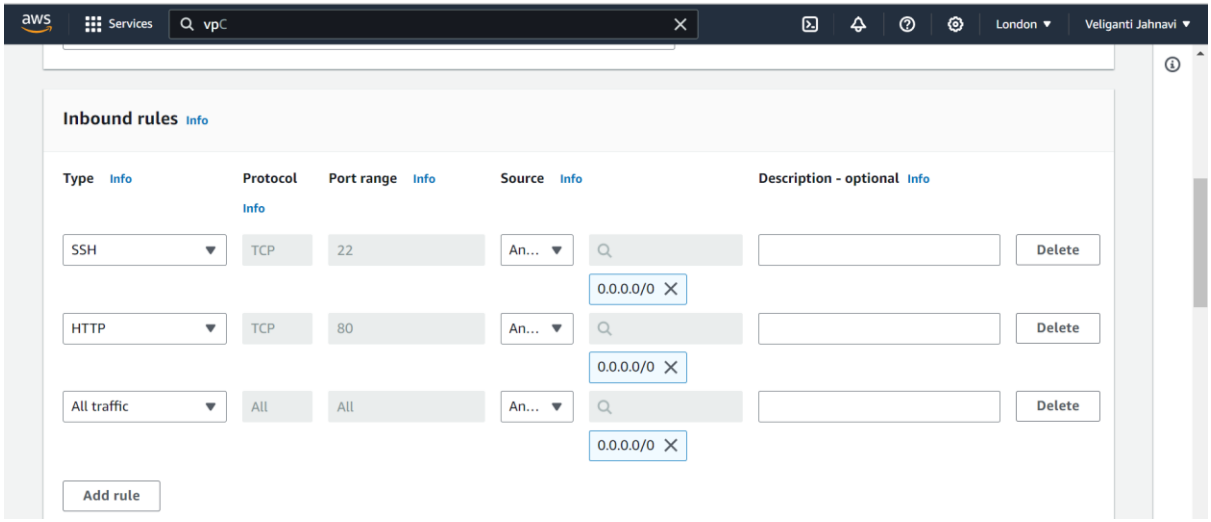
NAT-1(SUBNET PRIVATE-1,3)



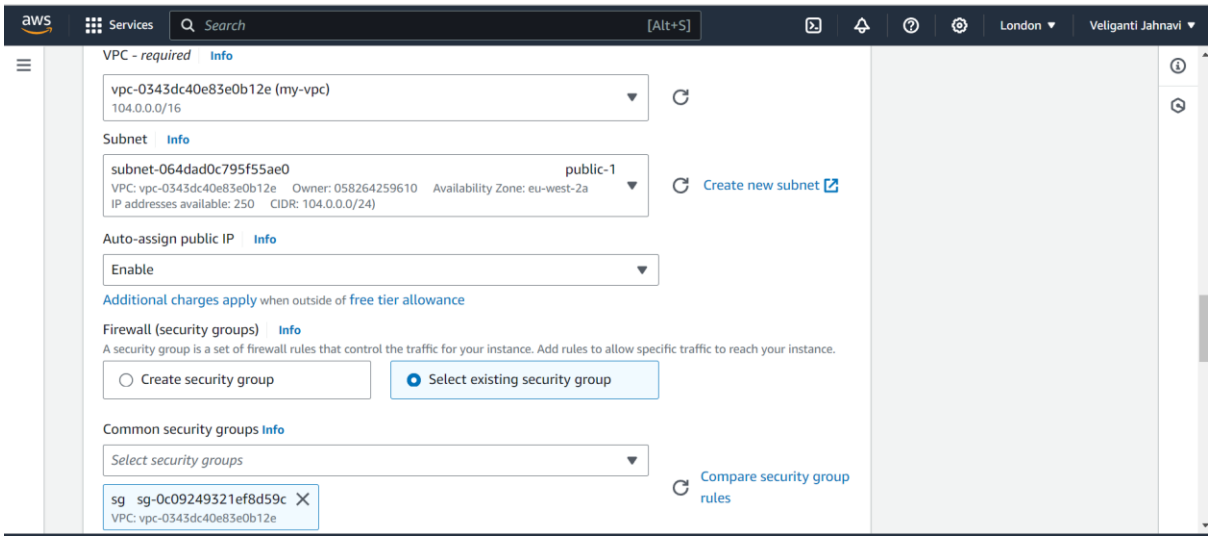
NAT-2(SUBNET PRIVATE-2,4)

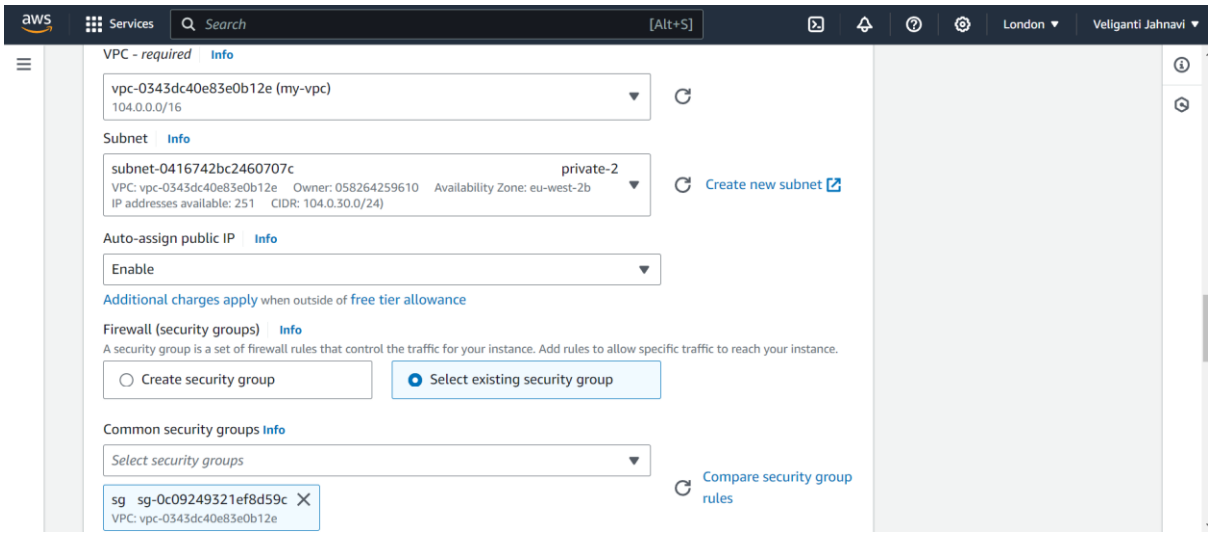


Step 8: Create a Security Group.



Step 9: Create 2 EC2 Instances (Public and Private).





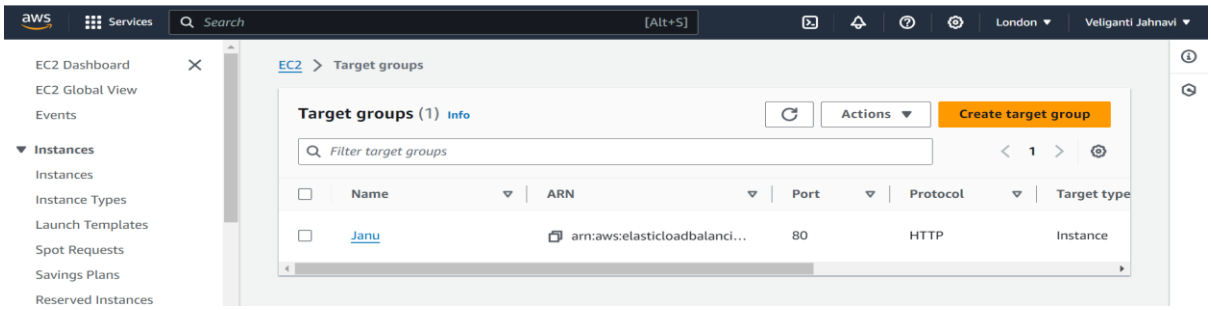
Connect to EC2 instance (PUBLIC[EC1-public]).

```
root@ip-104-0-0-191:/usr/share/nginx/html

banu@LAPTOP-Q7RDOEPG MINGW64 ~
$ ssh -i "Hello.pem" ec2-user@18.130.133.209
       #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sun Jul 28 07:25:32 2024 from 205.254.168.146
[ec2-user@ip-104-0-0-191 ~]$ sudo -i
[root@ip-104-0-0-191 ~]# yum update -y
Last metadata expiration check: 0:06:12 ago on Sun Jul 28 07:23:35 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-104-0-0-191 ~]# yum install nginx -y
Last metadata expiration check: 0:06:26 ago on Sun Jul 28 07:23:35 2024.
Dependencies resolved.
================================================================================
 Package                 Architecture  Version                    Repository  Size
================================================================================
Installing:
 nginx                   x86_64        1:1.24.0-1.amzn2023.0.2    amazonlinux 32 k
Installing dependencies:
 generic-logos-httpd     noarch        18.0.0-12.amzn2023.0.3     amazonlinux 19 k
 gperftools-libs         x86_64        2.9.1-1.amzn2023.0.3       amazonlinux 308 k
 libunwind               x86_64        1.4.0-5.amzn2023.0.2       amazonlinux 66 k
 nginx-core              x86_64        1:1.24.0-1.amzn2023.0.2    amazonlinux 586 k
 nginx-filesystem        noarch        1:1.24.0-1.amzn2023.0.2    amazonlinux 9.1 k
 nginx-mimetypes         noarch        2.1.49-3.amzn2023.0.3      amazonlinux 21 k
```

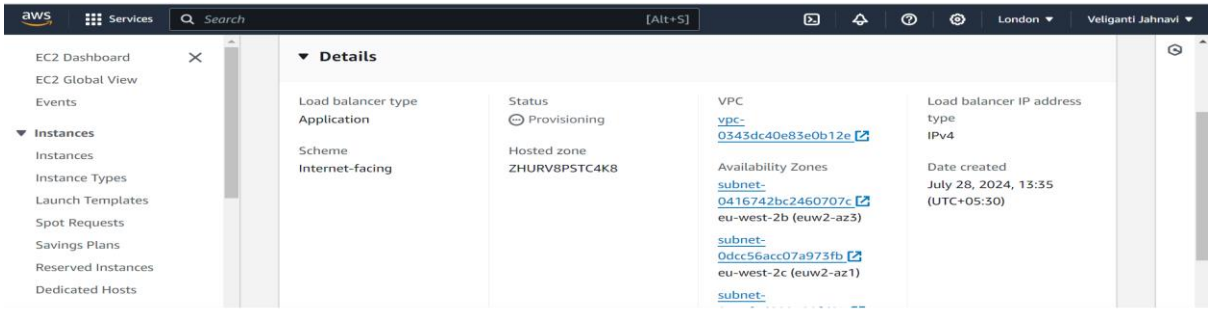
```
  nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!
[root@ip-104-0-0-191 ~]# cd /usr/share/nginx/html
[root@ip-104-0-0-191 html]# rm index.html
rm: remove regular file 'index.html'? yes
[root@ip-104-0-0-191 html]# vi index.html
[root@ip-104-0-0-191 html]# systemctl restart nginx
[root@ip-104-0-0-191 html]#
```

This is Public Instance

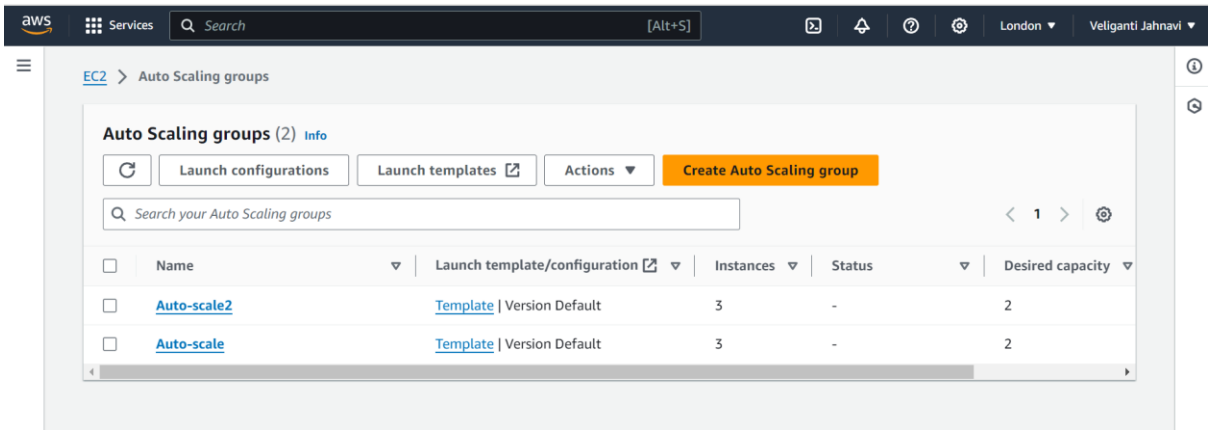
Step 10: Create a Target Group.



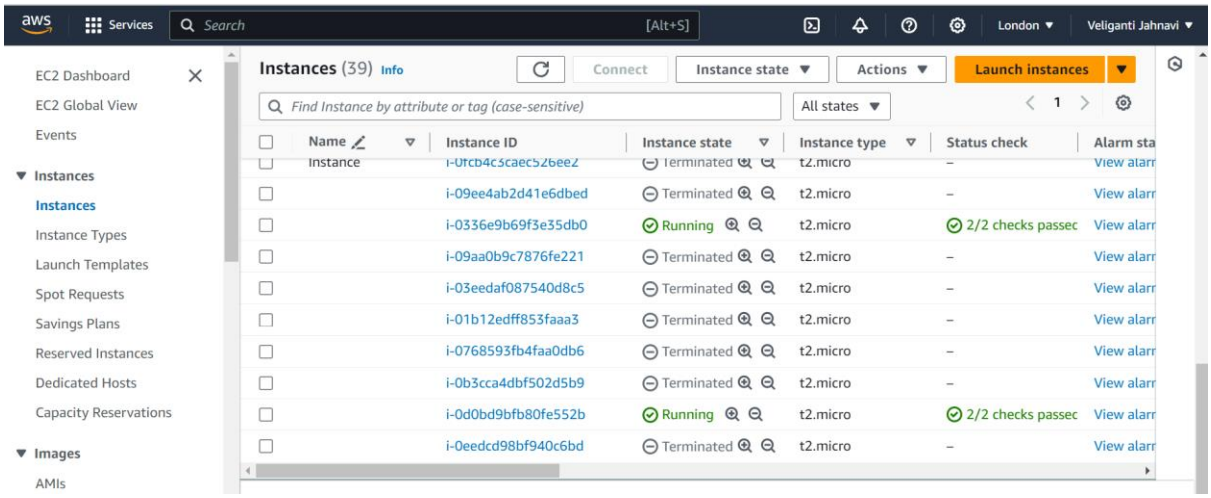
Step 11: Create a Load Balancer.



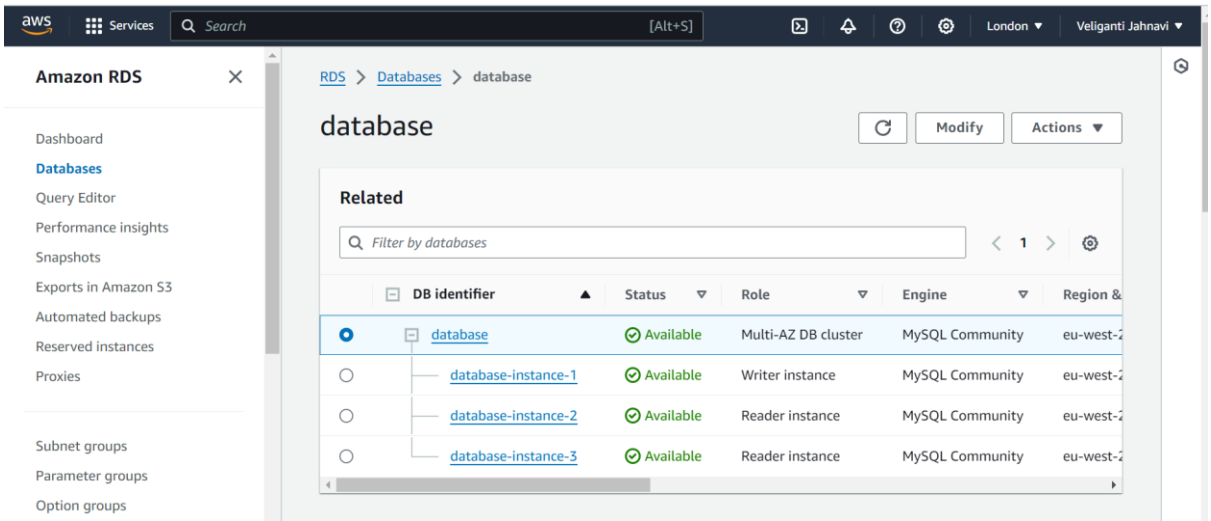
Step 12: Create an Auto Scaling Group (Public, private).



Auto Create Instance and Terminated.



Step 13: Go to RDS and Create databases.



Connect to Server→Install MariaDB→Connect to MYSQL Server.

```
root@ip-104-0-0-191:~
[root@ip-104-0-0-191 ~]# sudo yum install -y https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
Last metadata expiration check: 1:13:53 ago on Sun Jul 28 11:47:00 2024.
mysql57-community-release-el7-11.noarch.rpm                                                    104 kB/s |  2
Error:
 Problem: problem with installed package mysql80-community-release-el9-5.noarch
  - package mysql80-community-release-el9-5.noarch from @System conflicts with mysql57-community-release provided by mysql57-c
7-11.noarch from @commandline
  - package mysql80-community-release-el9-1.noarch from mysql80-community conflicts with mysql57-community-release provided by
release-el7-11.noarch from @commandline
  - package mysql80-community-release-el9-3.noarch from mysql80-community conflicts with mysql57-community-release provided by
release-el7-11.noarch from @commandline
  - package mysql80-community-release-el9-4.noarch from mysql80-community conflicts with mysql57-community-release provided by
release-el7-11.noarch from @commandline
  - package mysql80-community-release-el9-5.noarch from mysql80-community conflicts with mysql57-community-release provided by
release-el7-11.noarch from @commandline
  - conflicting requests
(try to add '--allowerasing' to command line to replace conflicting packages or '--skip-broken' to skip uninstallable packages
[root@ip-104-0-0-191 ~]# sudo dnf update -y
Last metadata expiration check: 1:15:43 ago on Sun Jul 28 11:47:00 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-104-0-0-191 ~]# sudo dnf install mariadb105-server
Last metadata expiration check: 1:15:59 ago on Sun Jul 28 11:47:00 2024.
Package mariadb105-server-3:10.5.23-1.amzn2023.0.1.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-104-0-0-191 ~]# mysql -h database.cluster-cxs28mgewqag.eu-west-2.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 96
Server version: 8.0.37 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
```

```
root@ip-104-0-0-191:~

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.002 sec)

MySQL [(none)]> create database Jahnavi;
Query OK, 1 row affected (0.041 sec)

MySQL [(none)]> use Jahnavi;
Database changed
MySQL [Jahnavi]> CREATE TABLE Persons(
    -> ID int NOT NULL,
    -> LastName varchar(255) NOT NULL,
    -> FirstName varchar(255),
    -> Age int,
    -> PRIMARY KEY(ID)
    -> );
Query OK, 0 rows affected (0.012 sec)

MySQL [Jahnavi]> show databases;
+--------------------+
| Database           |
+--------------------+
| Jahnavi            |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
```

```
root@ip-104-0-0-191:~
Query OK, 0 rows affected (0.012 sec)

MySQL [Jahnavi]> show databases;
+--------------------+
| Database           |
+--------------------+
| Jahnavi            |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.001 sec)

MySQL [Jahnavi]> show tables;
+------------------+
| Tables_in_Jahnavi |
+------------------+
| Persons          |
+------------------+
1 row in set (0.002 sec)

MySQL [Jahnavi]> INSERT INTO Persons(ID,LastName,FirstN
ame,Age)
    -> VALUES(101,'Jahnavi','Veliganti',21);
Query OK, 1 row affected (0.013 sec)

MySQL [Jahnavi]> select * from Persons;
+-----+----------+-----------+------+
| ID  | LastName | FirstName | Age  |
+-----+----------+-----------+------+
| 101 | Jahnavi  | Veliganti |   21 |
+-----+----------+-----------+------+
1 row in set (0.001 sec)

MySQL [Jahnavi]>
```