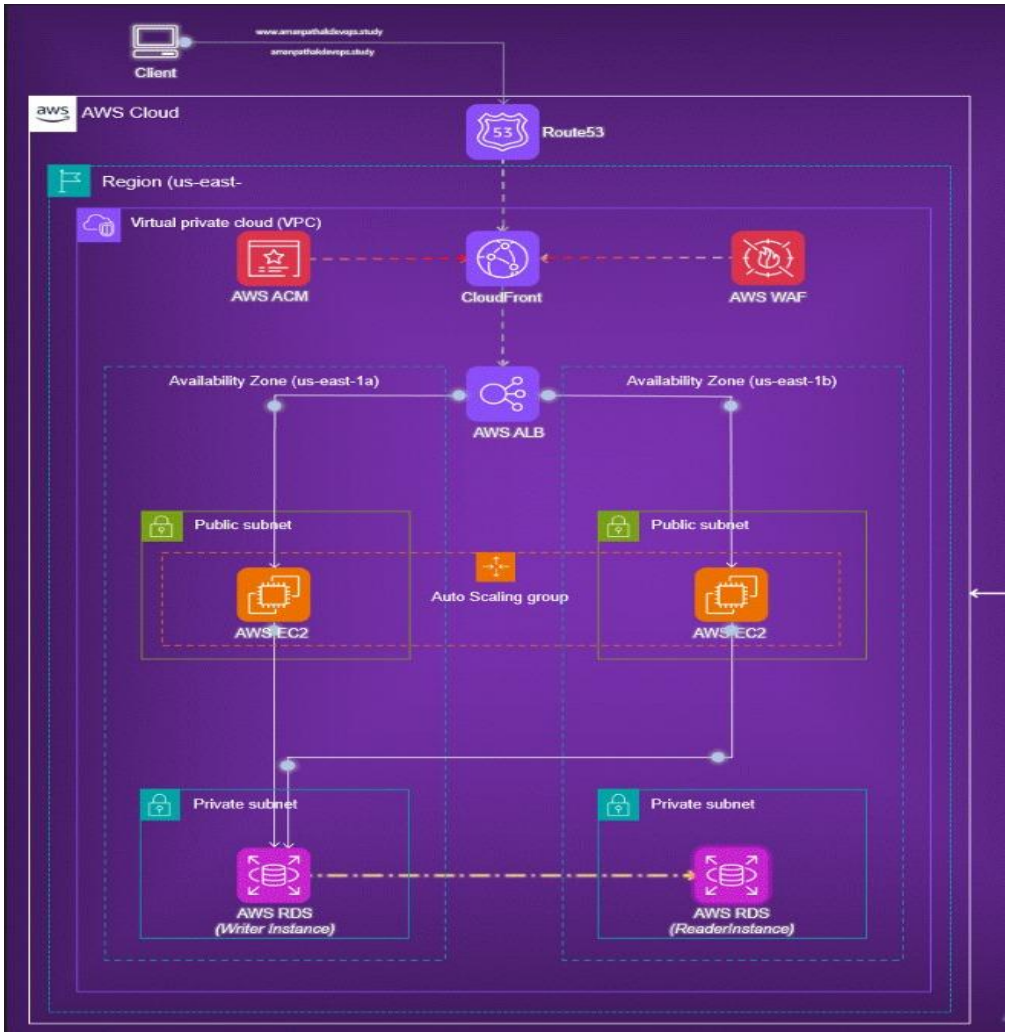
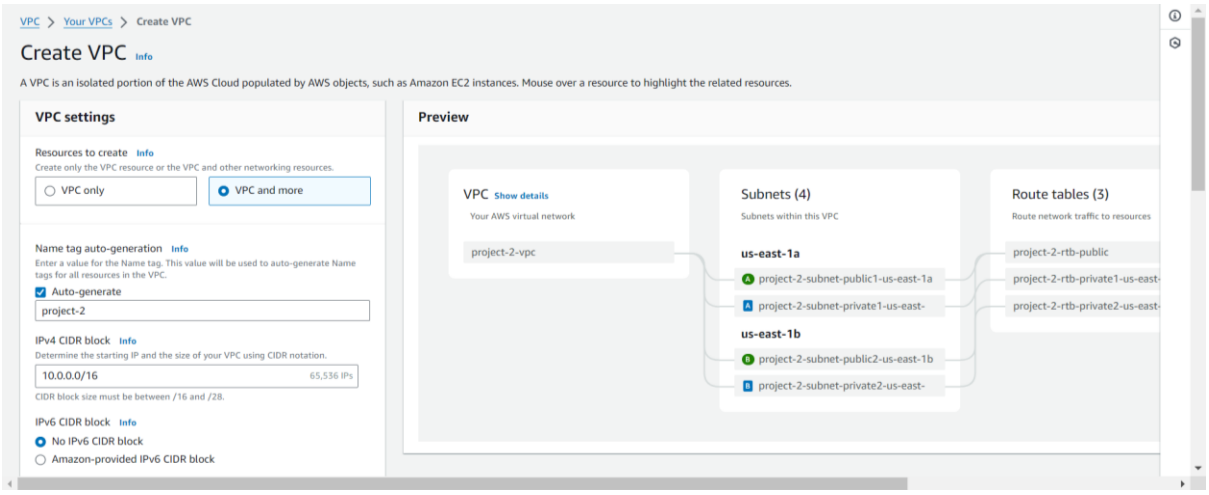
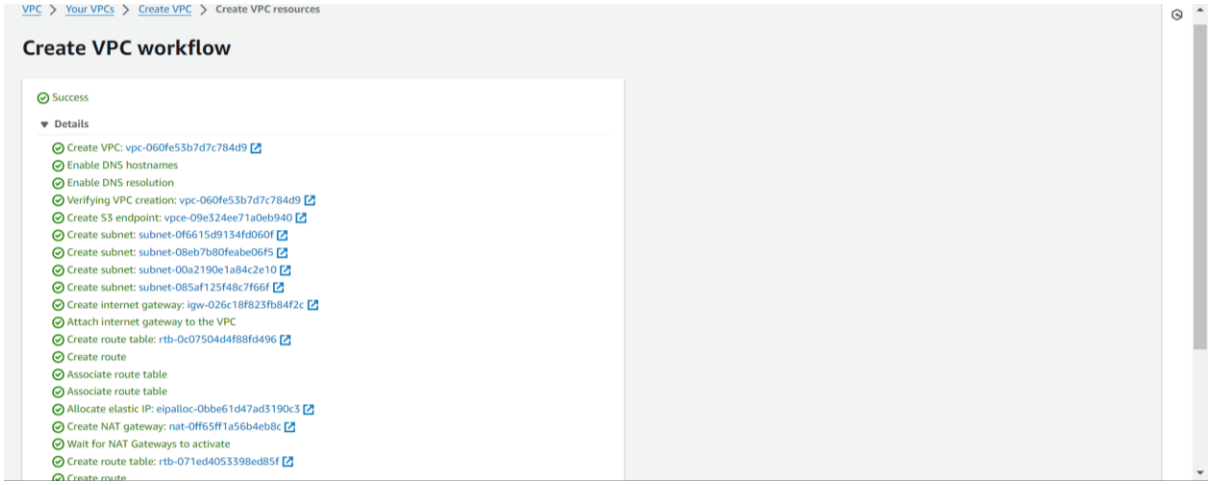
TWO TIER ARCHITECTURE USING ROUTE53

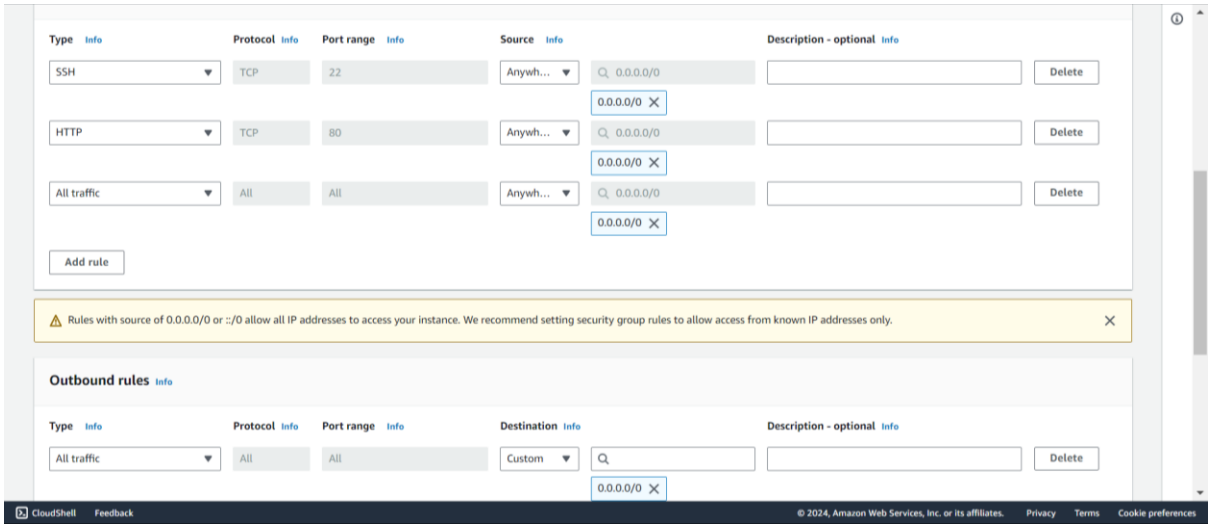


1. Create 2 Public subnets,2 Private subnets, Route tables, VPC, Internet Gateway, Nat gateway and subnet associations using VPC and more

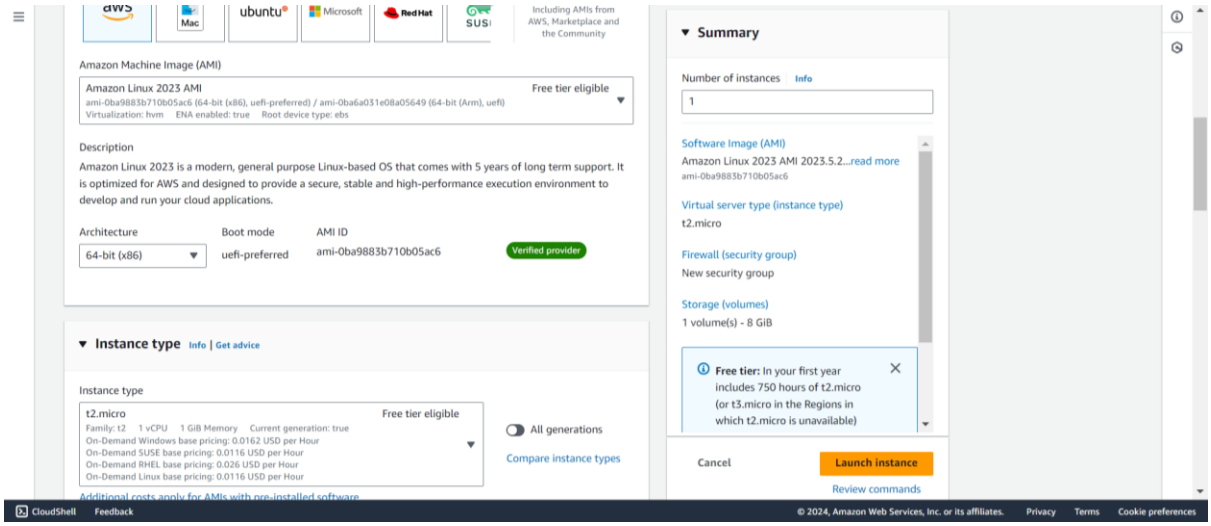


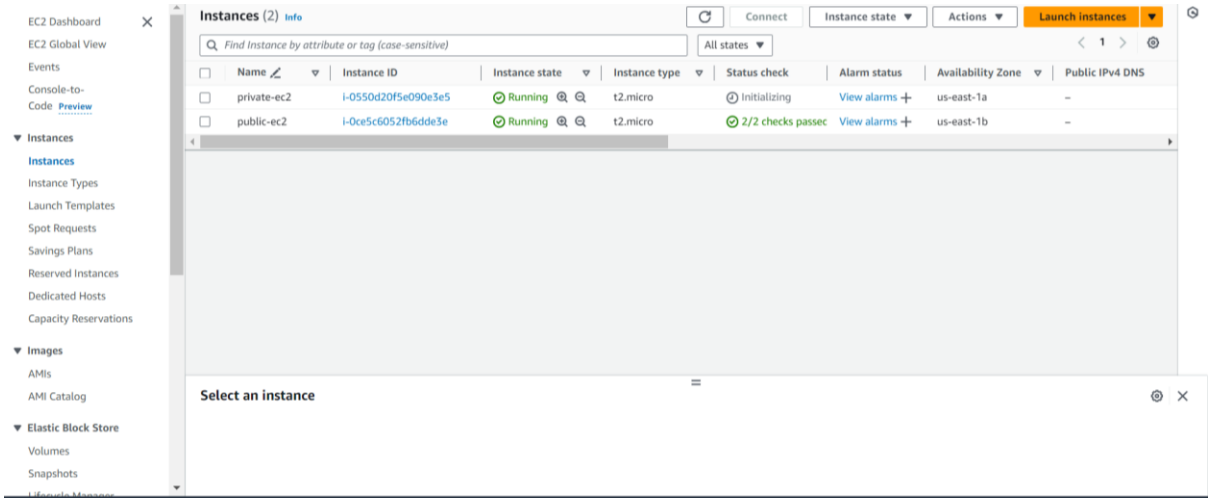


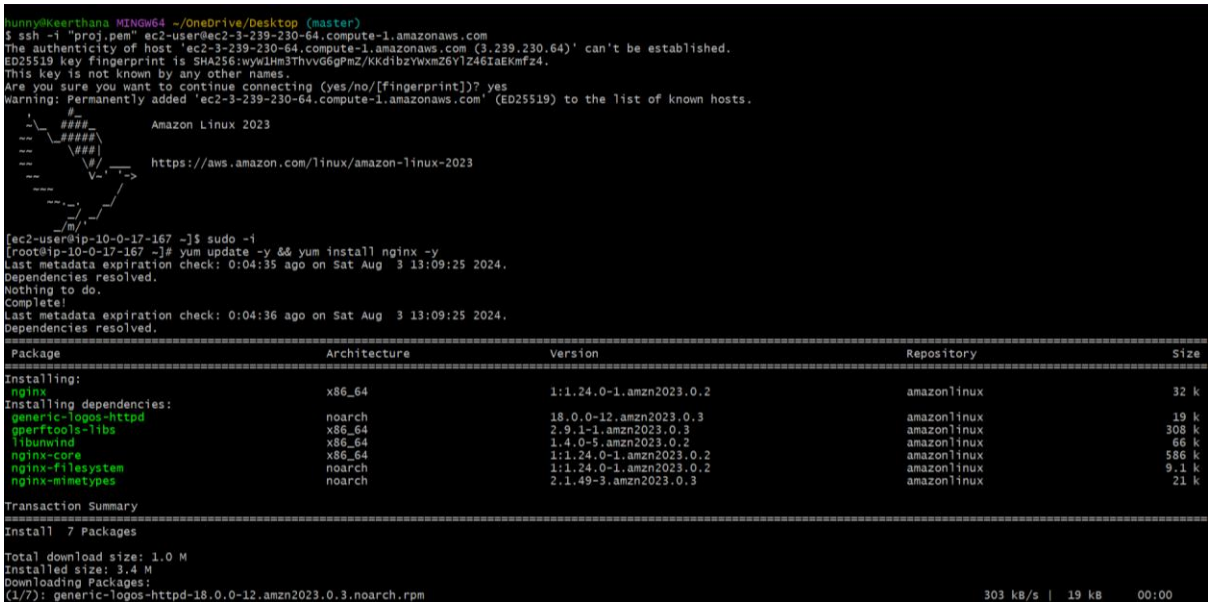
2. Create security group



3. Launch two instances, one public instance and one private instance and connect them.

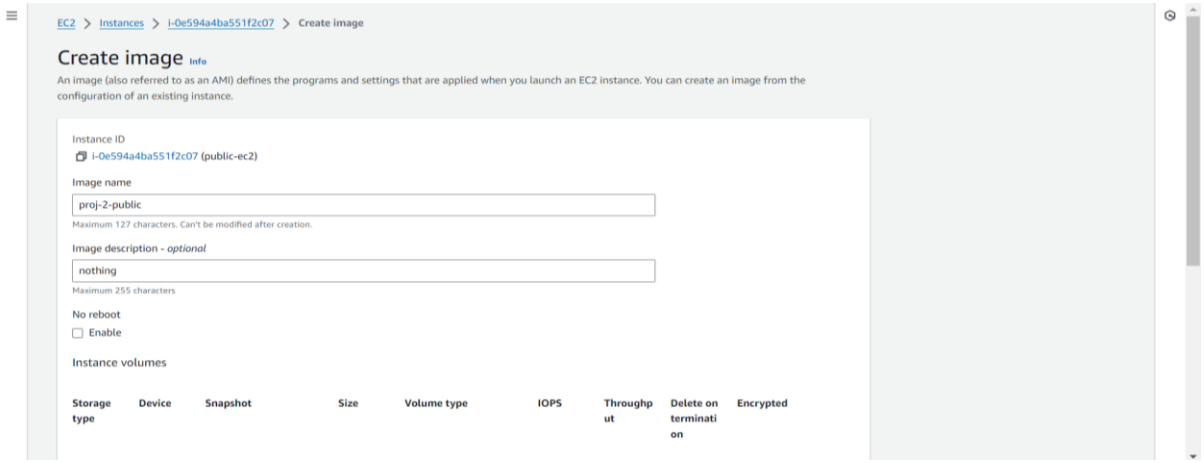


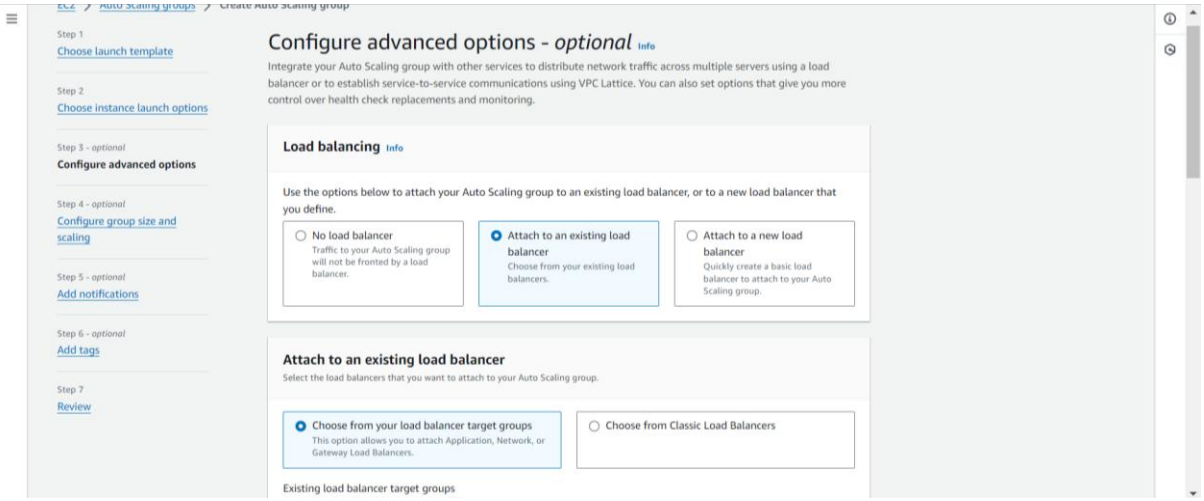


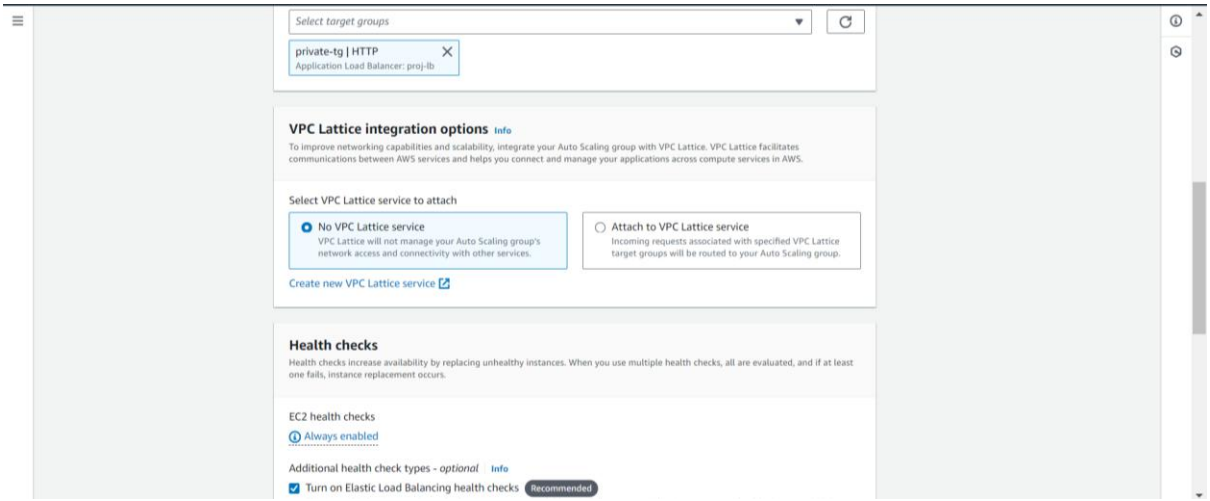


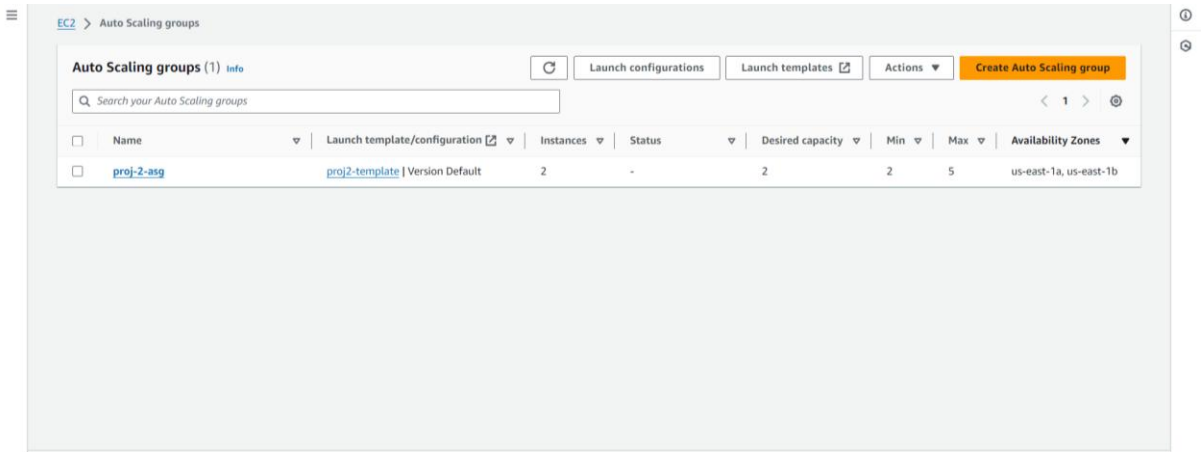
this is public instance

4. Create autoscaling group, for that first create ami image and launch template

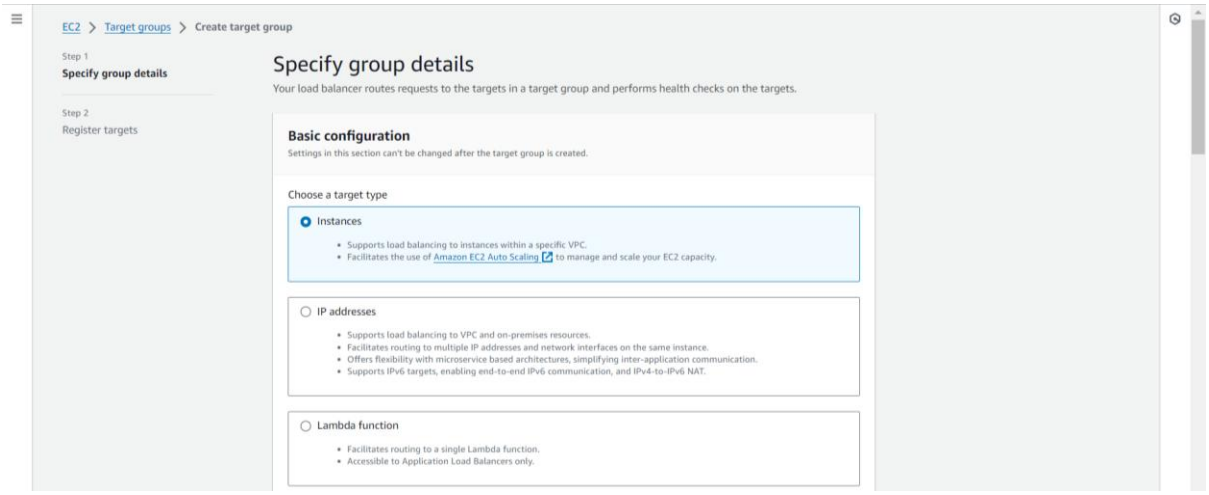


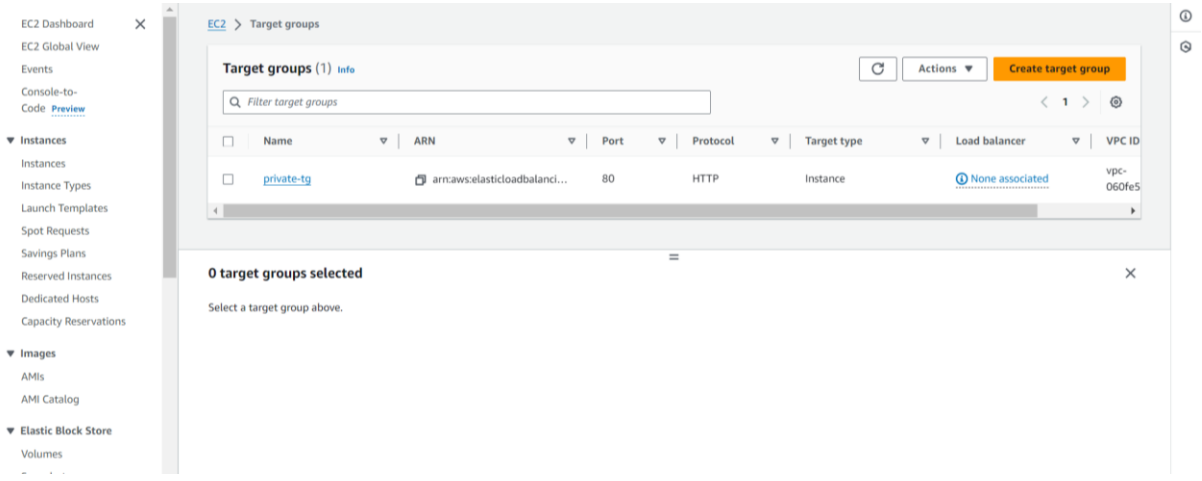




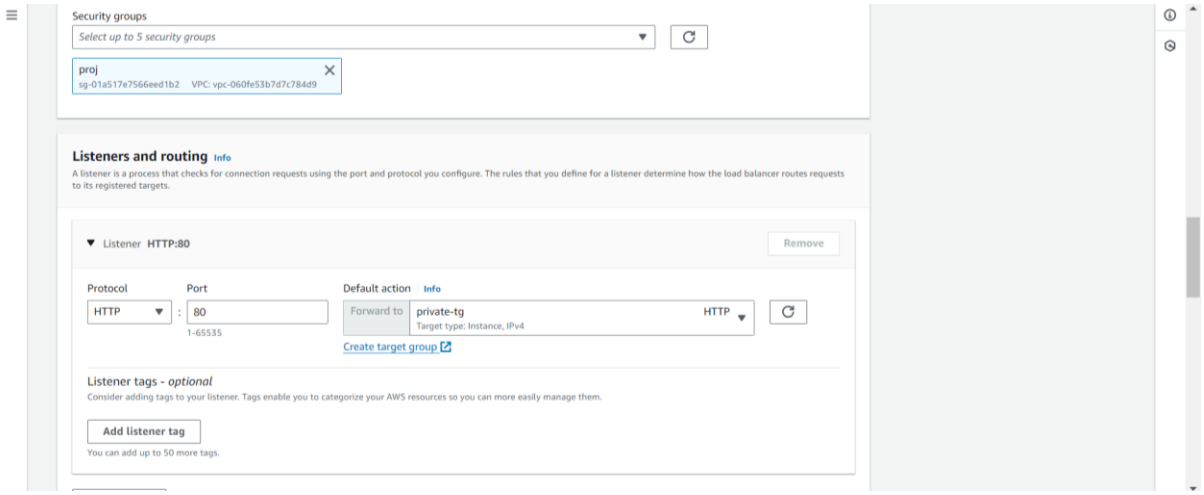


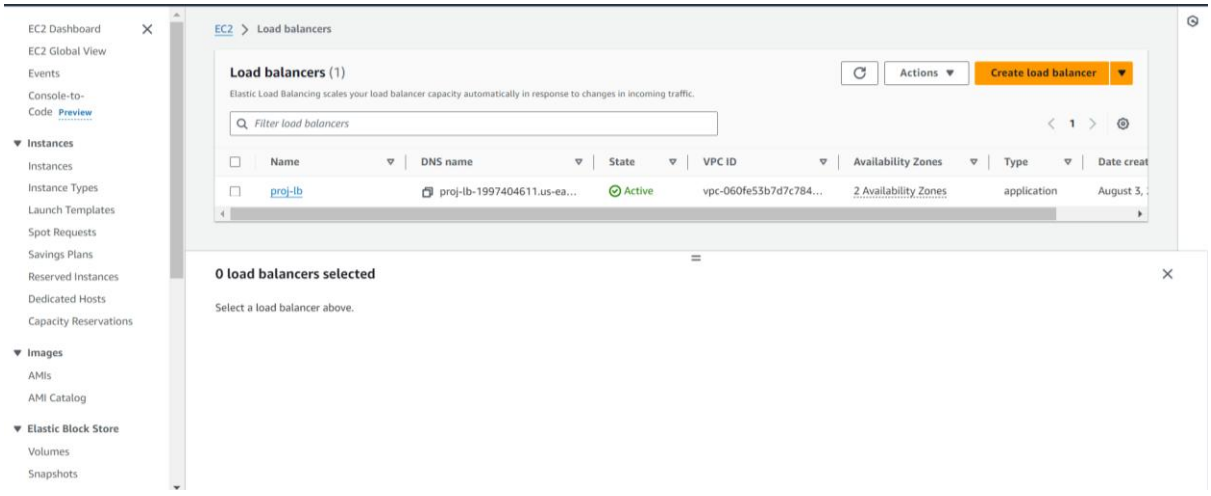
5. Create target group and load balancer





Create load balancer



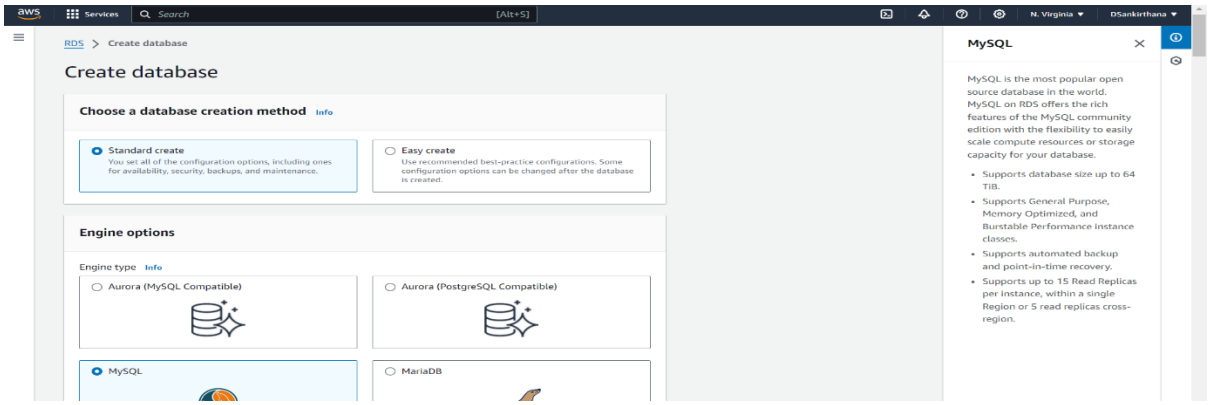


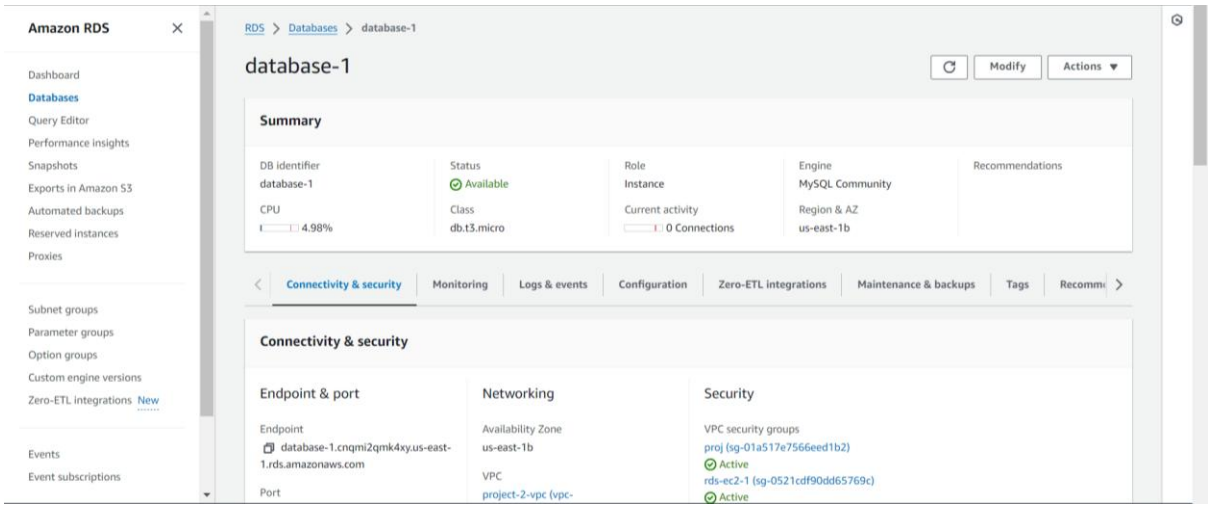
Copy DNS from above image and keep refreshing to get both the instances output

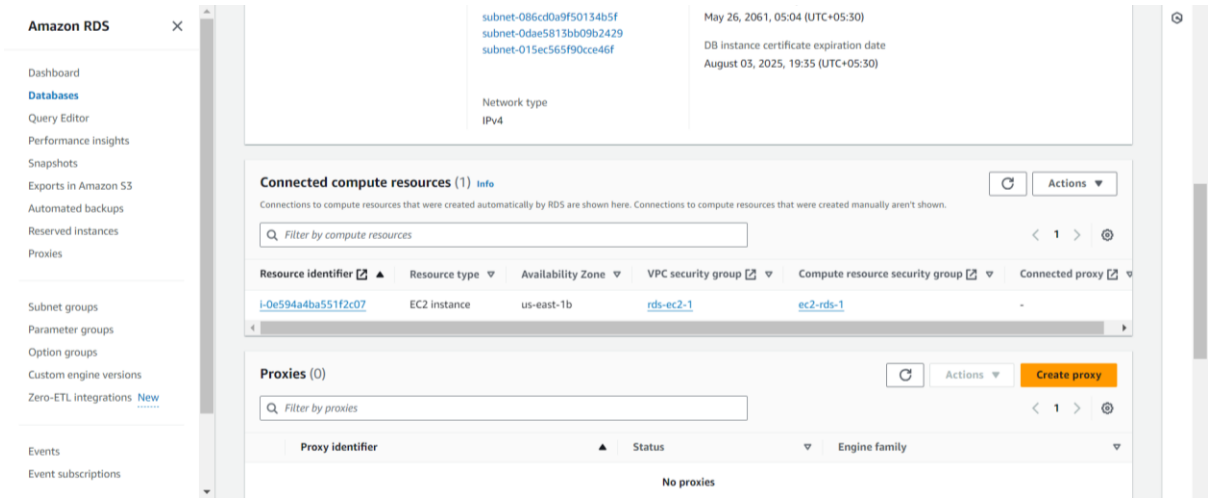
this is public instance

this is private instance

6. Create RDS


RDS > Create database
Create database
Choose a database creation method Info
Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.
Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.
Engine options
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
MariaDB
MySQL
MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.
Supports database size up to 64 TiB.
Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
Supports automated backup and point-in-time recovery.
Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.


Amazon RDS
Dashboard
Databases
Query Editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations New
Events
Event subscriptions
RDS > Databases > database-1
database-1
Modify
Actions
Summary
DB identifier
database-1
Status
Available
Role
Instance
Engine
MySQL Community
Recommendations
CPU
4.98%
Class
db.t3.micro
Current activity
0 Connections
Region & AZ
us-east-1b
Connectivity & security
Monitoring
Logs & events
Configuration
Zero-ETL integrations
Maintenance & backups
Tags
Connectivity & security
Endpoint & port
Endpoint
database-1.cnqmi2qmk4xy.us-east-1.rds.amazonaws.com
Port
Networking
Availability Zone
us-east-1b
VPC
project-2-vpc (vpc-
Security
VPC security groups
proj (sg-01a517e7566eed1b2)
Active
rds-ec2-1 (sg-0521cdf90dd65769c)
Active

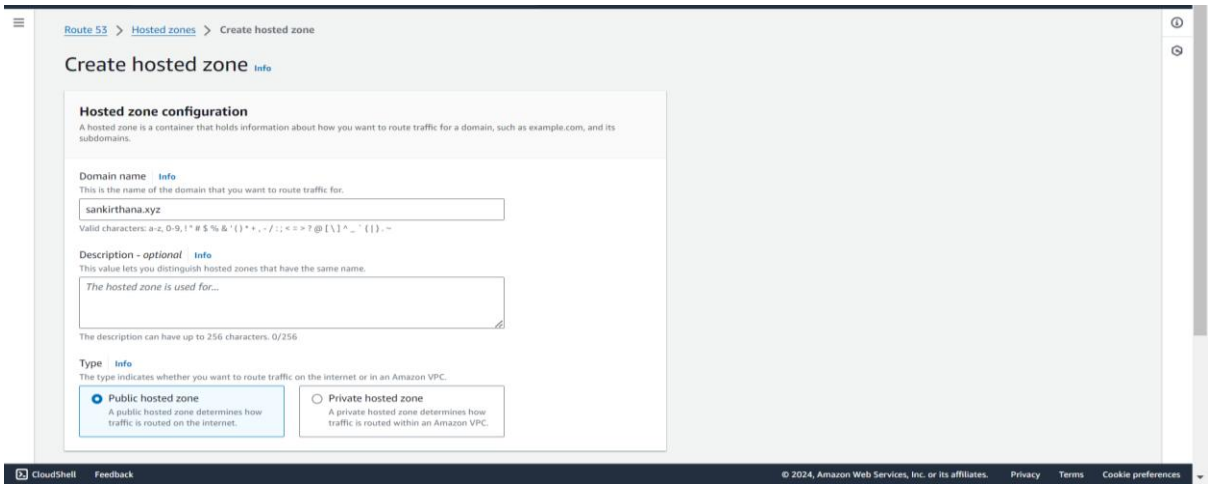

Amazon RDS
Dashboard
Databases
Query Editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations New
Events
Event subscriptions
subnet-086cd0a9f50134b5f
subnet-0dae5813bb09b2429
subnet-015ec565f90cce46f
Network type
IPv4
May 26, 2061, 05:04 (UTC+05:30)
DB instance certificate expiration date
August 03, 2025, 19:35 (UTC+05:30)
Connected compute resources (1) Info
Actions
Connections to compute resources that were created automatically by RDS are shown here. Connections to compute resources that were created manually aren't shown.
Filter by compute resources
Resource identifier
Resource type
Availability Zone
VPC security group
Compute resource security group
Connected proxy
i-0e594a4ba551f2c07
EC2 instance
us-east-1b
rds-ec2-1
ec2-rds-1
Proxies (0)
Actions
Create proxy
Filter by proxies
Proxy identifier
Status
Engine family
No proxies

7. The output will be shown as follows

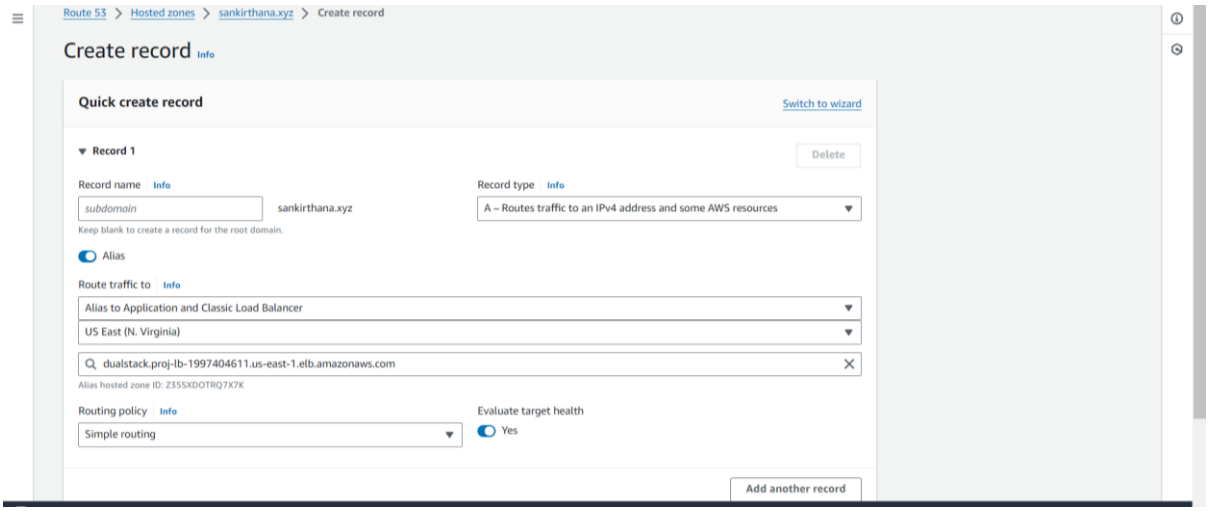
```
   ~~        \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
  ~~       V~' '->
   ~~~         /
     ~~._.   _/
        _/ _/
      _/m/'
Last login: Sat Aug  3 13:11:58 2024 from 175.101.32.98
[ec2-user@ip-10-0-17-167 ~]$ sudo -i
[root@ip-10-0-17-167 ~]# yum update -y && yum install nginx -y
Last metadata expiration check: 0:58:45 ago on Sat Aug  3 13:09:25 2024.
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 0:58:46 ago on Sat Aug  3 13:09:25 2024.
Package nginx-1:1.24.0-1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-10-0-17-167 ~]# echo "Saving value to MySQL RDS"
Saving value to MySQL RDS
[root@ip-10-0-17-167 ~]#  mysql -h $DB_HOST -u $DB_USER -p $DB_PASSWORD -D $DB_NAME -e "INSERT INTO versionReact (version) VALUES ('$REACT_APP_VERSION');"
-bash: mysql: command not found
[root@ip-10-0-17-167 ~]#  wget https://repo.mysql.com//mysql80-community-release-el9-3.noarch.rpm
--2024-08-03 14:10:05--  https://repo.mysql.com//mysql80-community-release-el9-3.noarch.rpm
Resolving repo.mysql.com (repo.mysql.com)... 23.33.204.253, 2600:1408:ec00:88f::1d68, 2600:1408:ec00:888::1d68
Connecting to repo.mysql.com (repo.mysql.com)|23.33.204.253|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10715 (10K) [application/x-redhat-package-manager]
Saving to: 'mysql80-community-release-el9-3.noarch.rpm'

mysql80-community-release-el9-3.noarch.rpm 100%[=========================================================================================>]  10.46K  --.-KB/s    in 0s

2024-08-03 14:10:05 (239 MB/s) - 'mysql80-community-release-el9-3.noarch.rpm' saved [10715/10715]

[root@ip-10-0-17-167 ~]# sudo ls
mysql80-community-release-el9-3.noarch.rpm
[root@ip-10-0-17-167 ~]#  sudo yum install mysql80-community-release-el9-1.noarch.rpm
Last metadata expiration check: 1:01:20 ago on Sat Aug  3 13:09:25 2024.
Can not load RPM file: mysql80-community-release-el9-1.noarch.rpm.
Could not open: mysql80-community-release-el9-1.noarch.rpm
[root@ip-10-0-17-167 ~]# sudo dnf update -y
Last metadata expiration check: 1:10:26 ago on Sat Aug  3 13:09:25 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-10-0-17-167 ~]# sudo dnf install mariadb105
Last metadata expiration check: 1:10:42 ago on Sat Aug  3 13:09:25 2024.
```

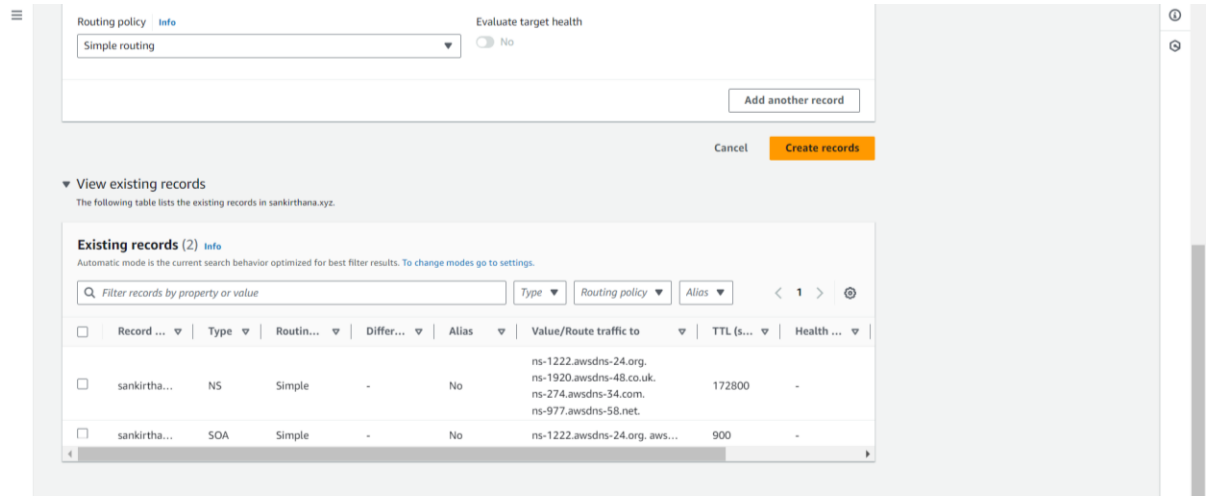
```
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.001 sec)

MySQL [(none)]> create database keerthana;
Query OK, 1 row affected (0.006 sec)

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| keerthana          |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.001 sec)

MySQL [(none)]> use keerthana;
Database changed
MySQL [keerthana]> 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 (0.027 sec)

MySQL [keerthana]> INSERT INTO Persons (ID, LastName, FirstName, Age)
    -> VALUES (101, 'Keerthana', 'Dasari', 22);
Query OK, 1 row affected (0.010 sec)

MySQL [keerthana]> select * from Persons;
+-----+----------+-----------+------+
| ID  | LastName | FirstName | Age  |
+-----+----------+-----------+------+
| 101 | Keerthana | Dasari    |   22 |
+-----+----------+-----------+------+
1 row in set (0.001 sec)
```

8. Create Route53 → Hosted Zones → Create hosted zone

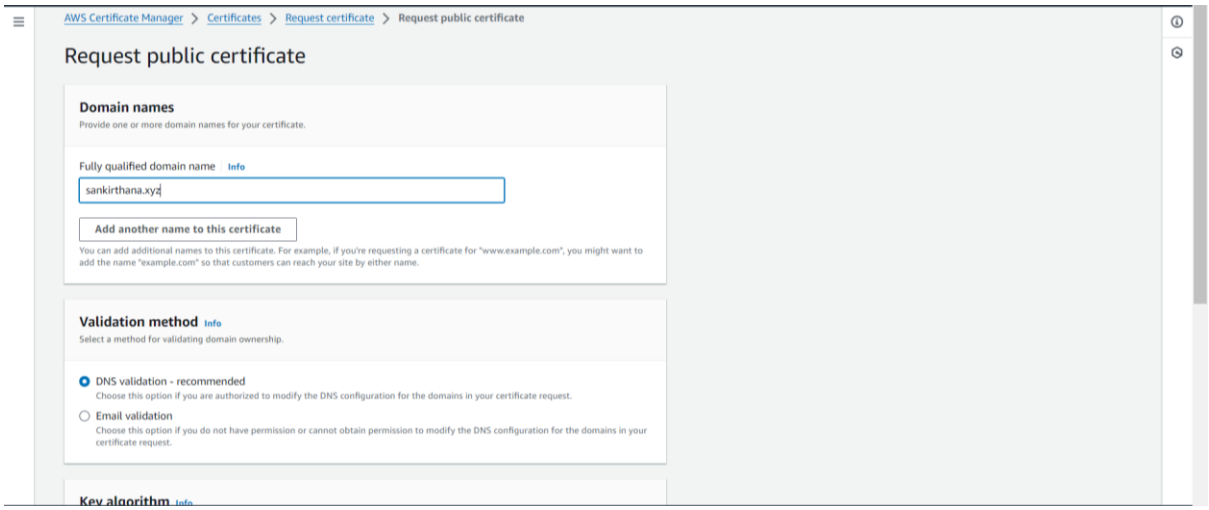


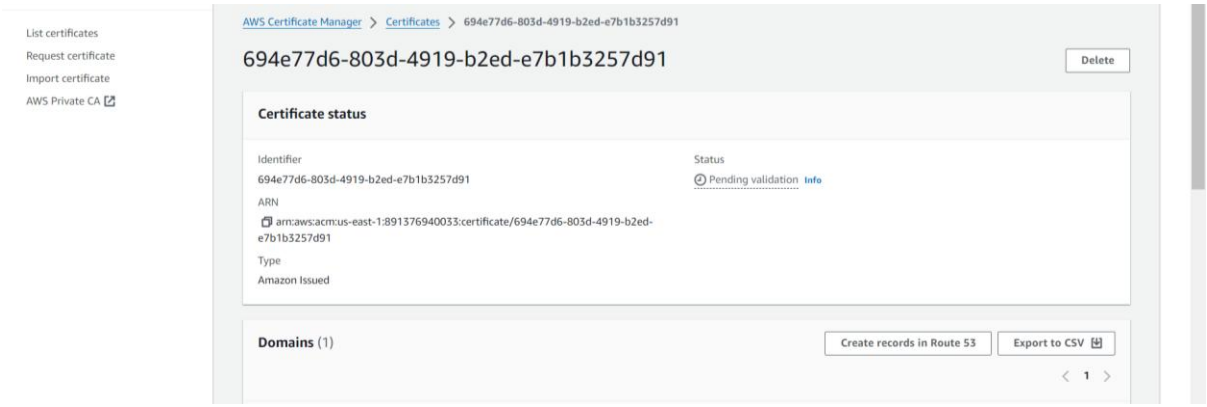
9. Create record

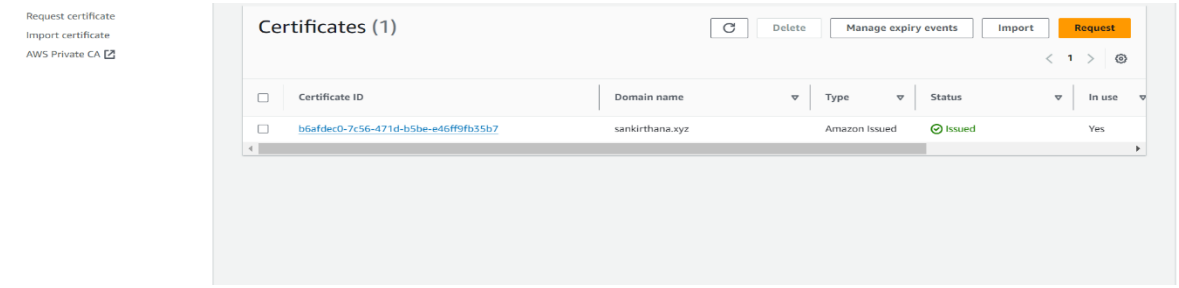




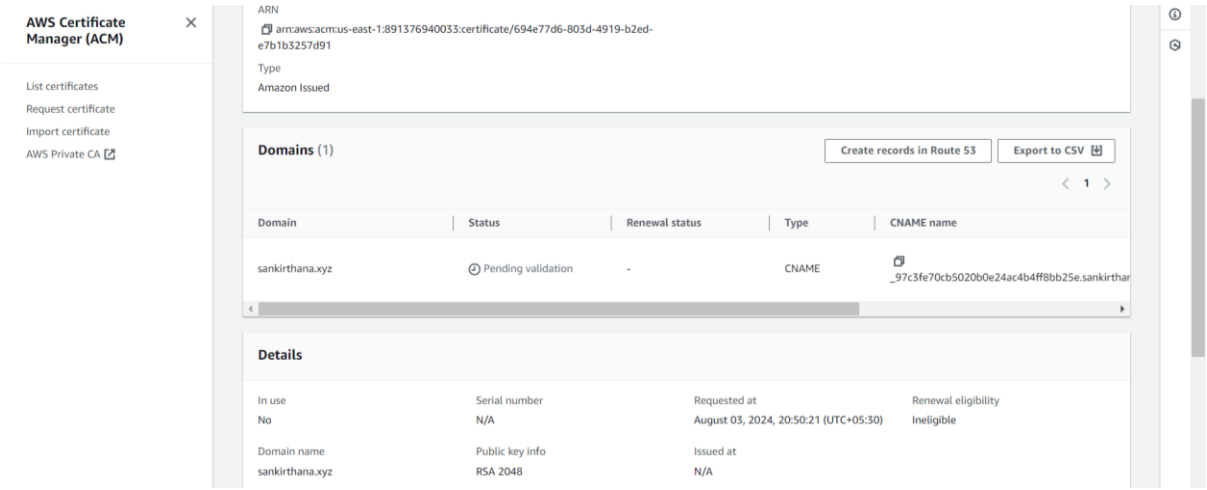
10. Create ACM certificate

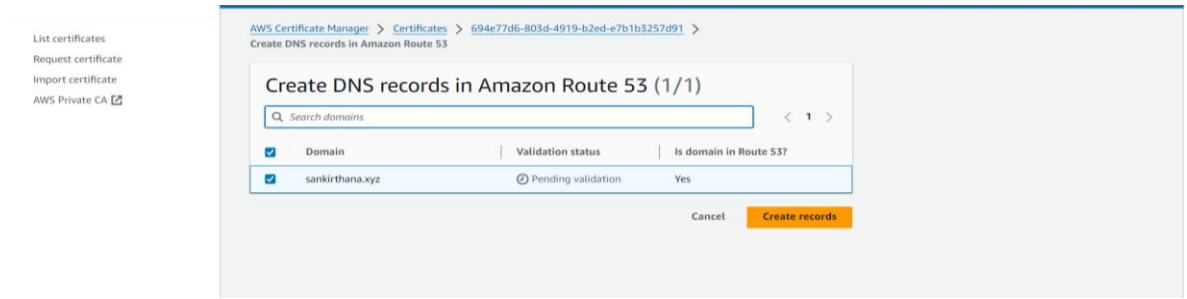




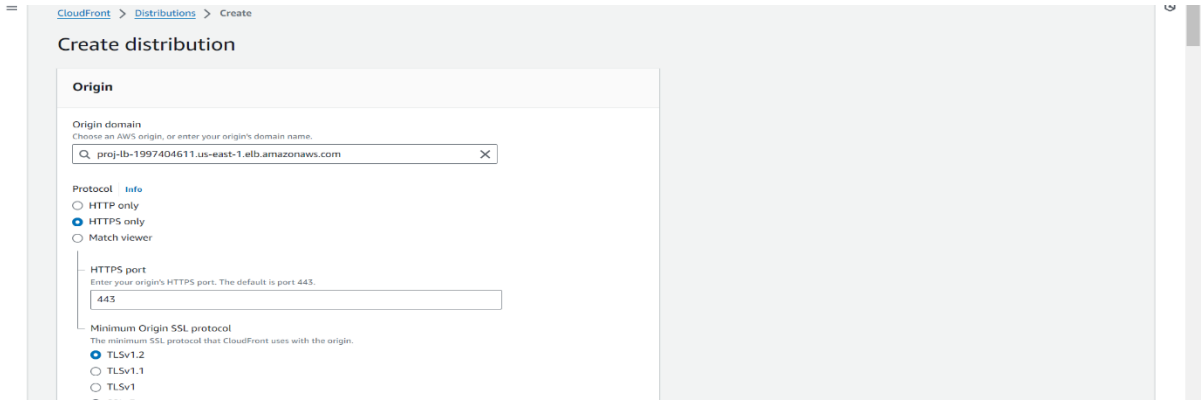


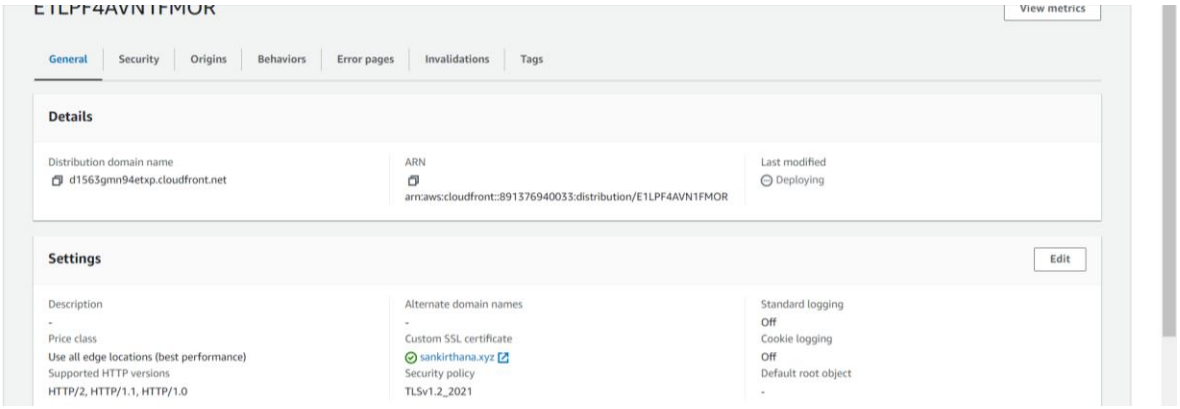
11. Create records in Route 53

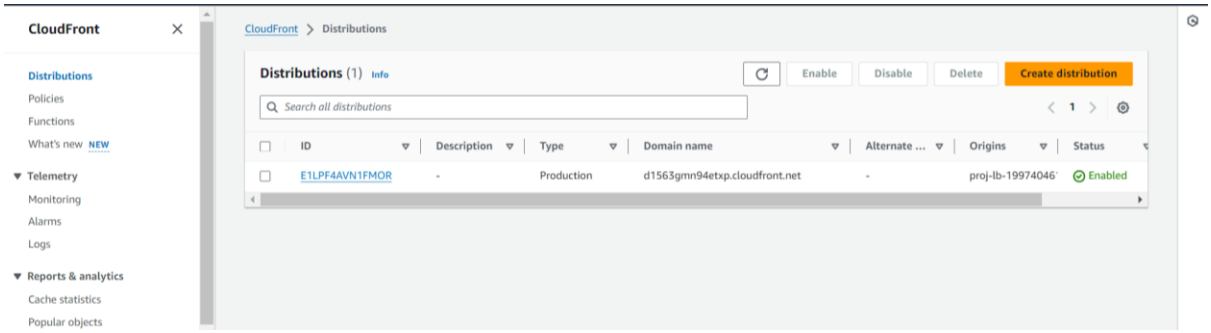




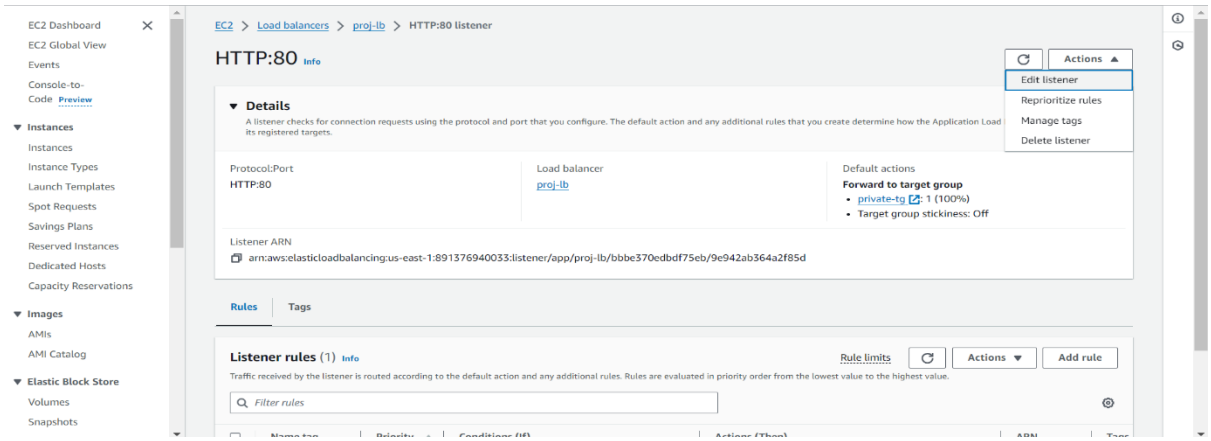
12. Create Cloud Front → Distributions

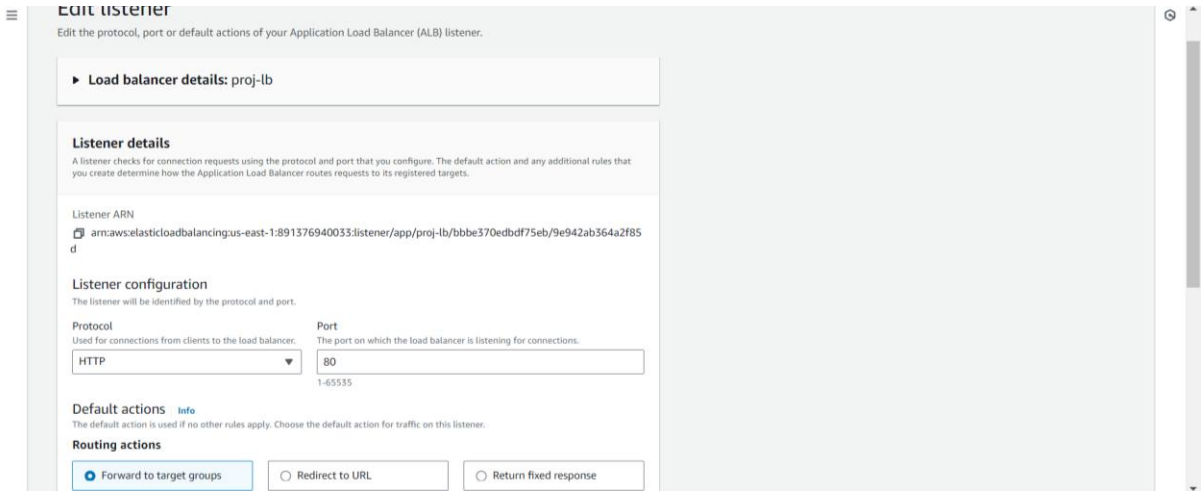


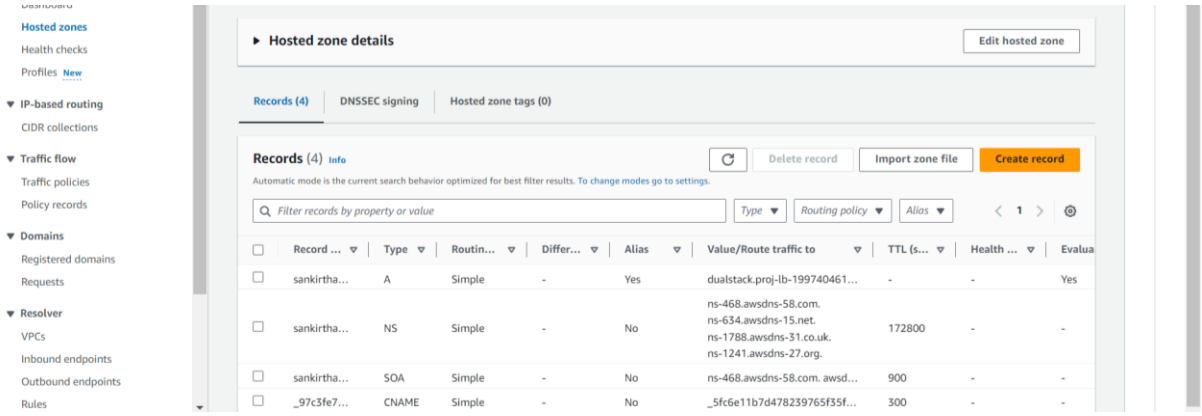




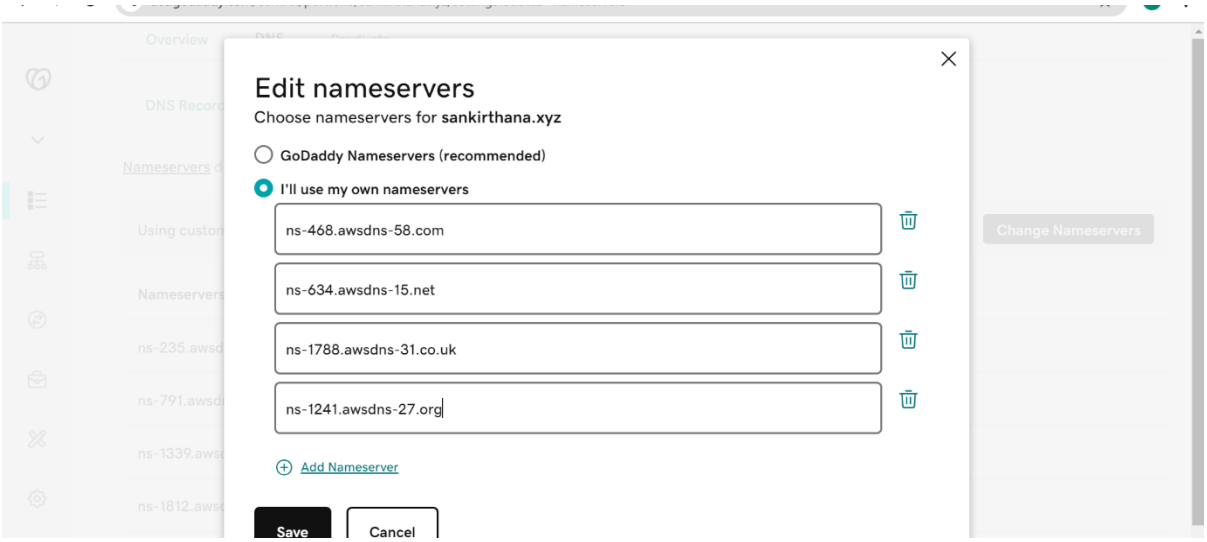
12. Create Cloud Front → Distributions







15. Go to godaddy website and change name servers.



16. Copy domain name and paste it in google to get the output