PROJECT

3 - TIER ARCHITECTURE

Name: M. Sreenath

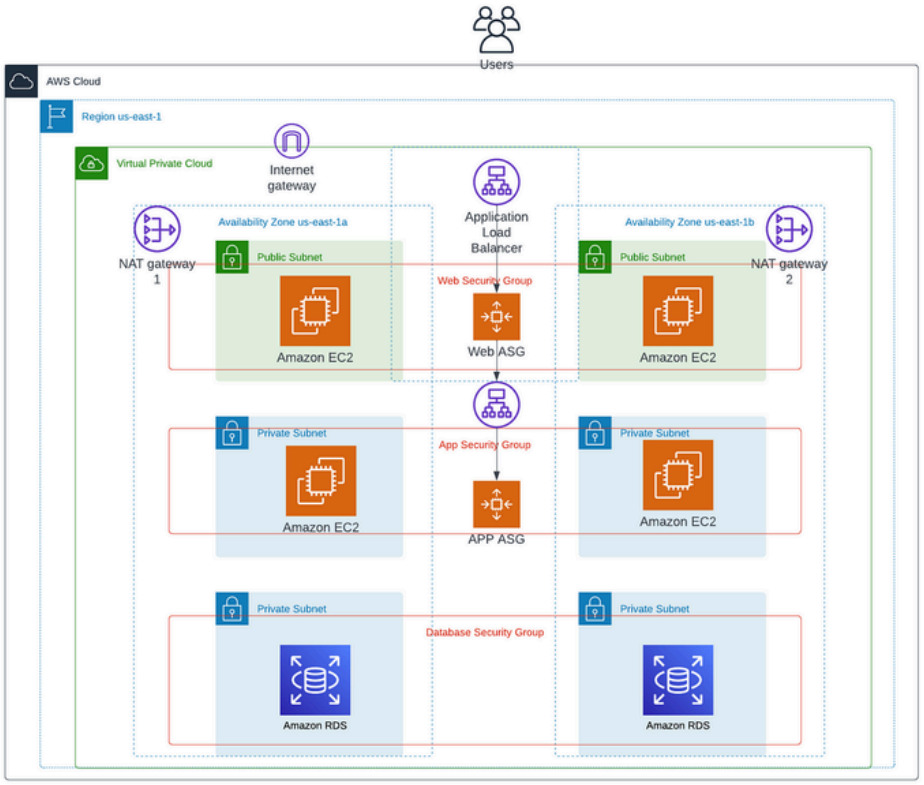


Figure: 3 - Tier Architecture

Before We Dive In...

Every grand structure rests on a foundation you rarely see — but without it, nothing stands. This project isn't just servers and subnets; it's a carefully crafted bridge between user curiosity and data precision

1. Architecture Overview

This section explains the purpose and high-level design of the 3-tier architecture.

The 3-tier architecture is designed to separate concerns and improve scalability, security, and manageability.

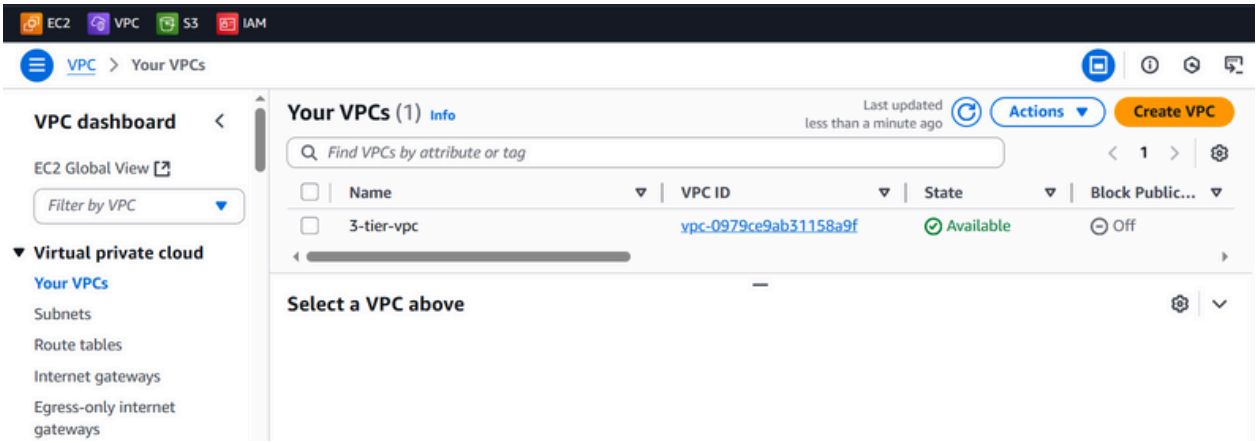
- **Web Tier (Presentation Layer):** Handles user interactions. Hosts the front-end or web servers that users access via a browser.
- **Application Tier (Logic Layer):** Contains business logic and processes the data between web and database layers.
- **Database Tier (Data Layer):** Stores and manages structured data securely. Only the app tier can communicate with this layer.

Steps to Create the 3-Tier Architecture:

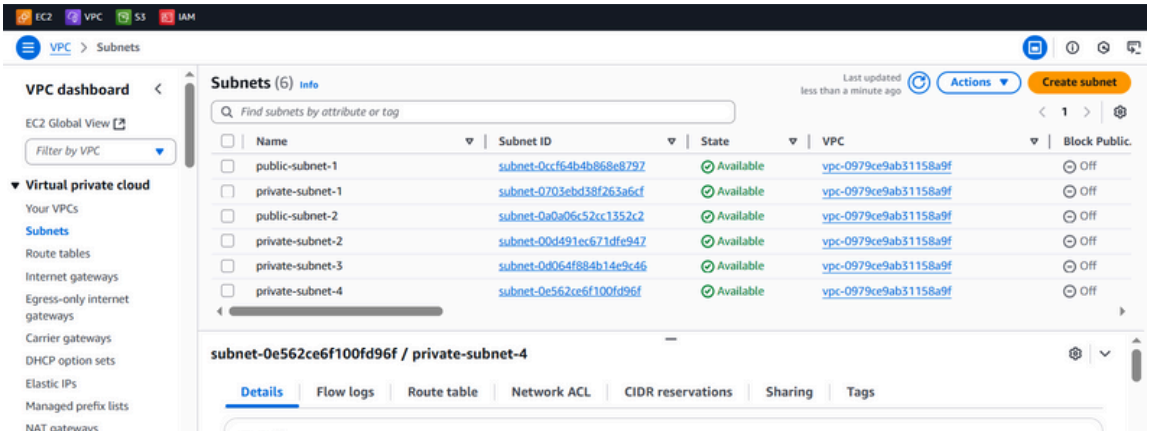
1. Create VPC, Subnets – 6, Internet gate way – 1, Route tables – 2, Nat gate way – 1.
2. Launch an EC2 instance.
3. Create Load Balancer
4. Create an AMI (image).
5. Create Autoscaling group, Create launch template.
6. Create Subnet group.
7. Create Database (RDS).
8. Establish connection.

Step: 1 Create VPC and its components.

1. Create VPC:

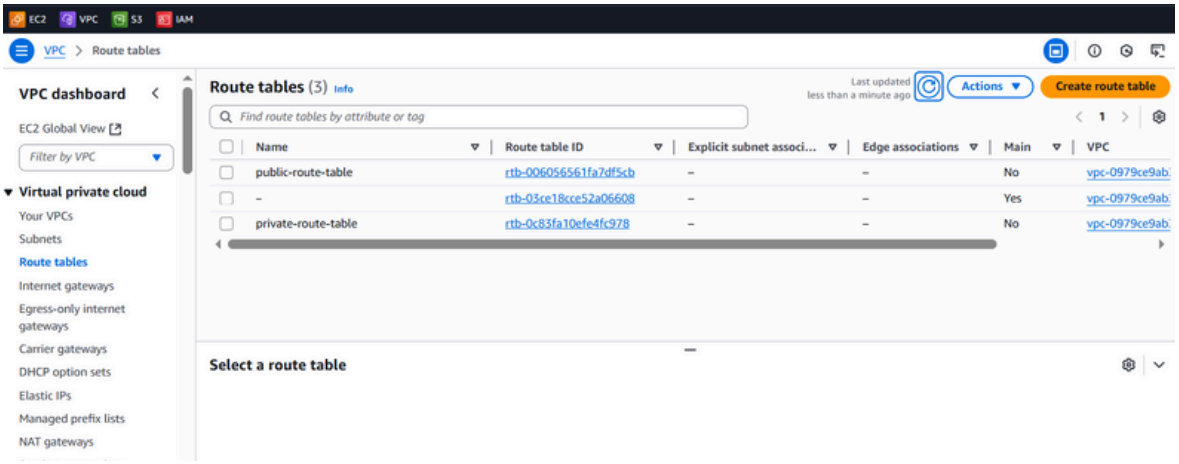


2. Subnet Setup: 2 Public, 4 Private

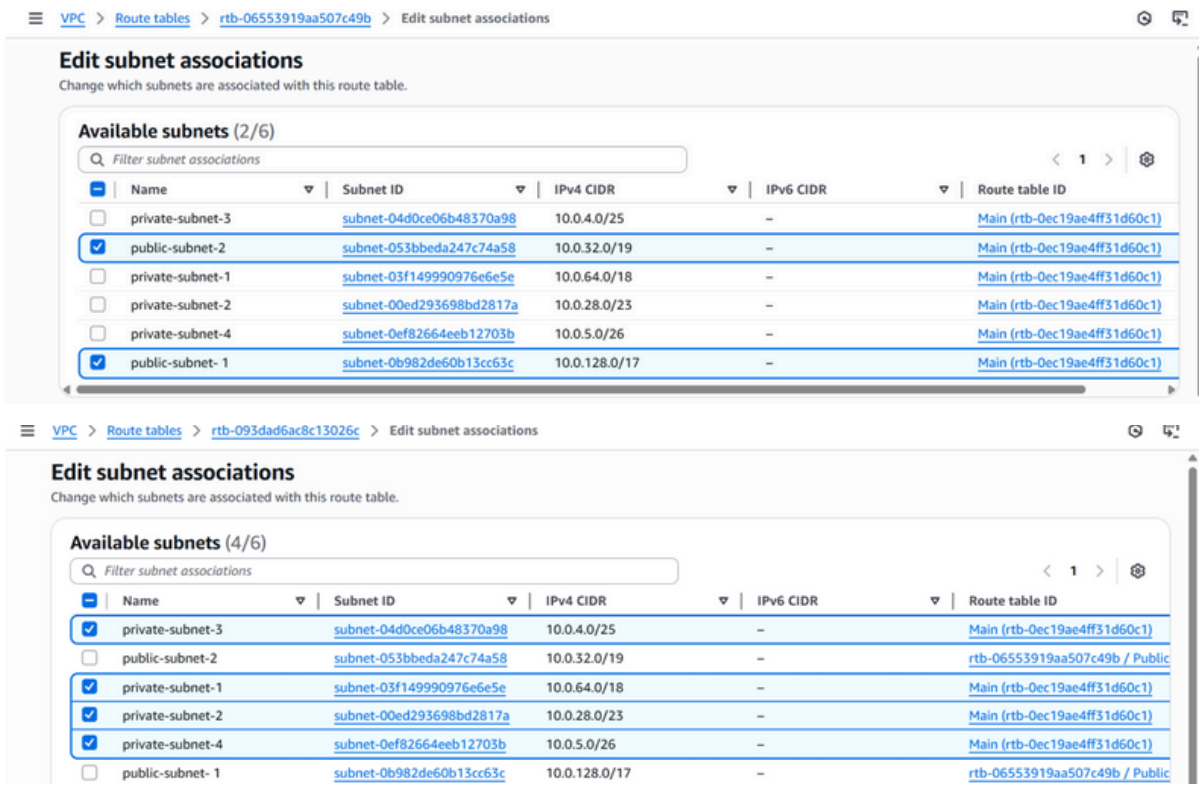


3. Internet Gateway Setup and attach to new VPC.

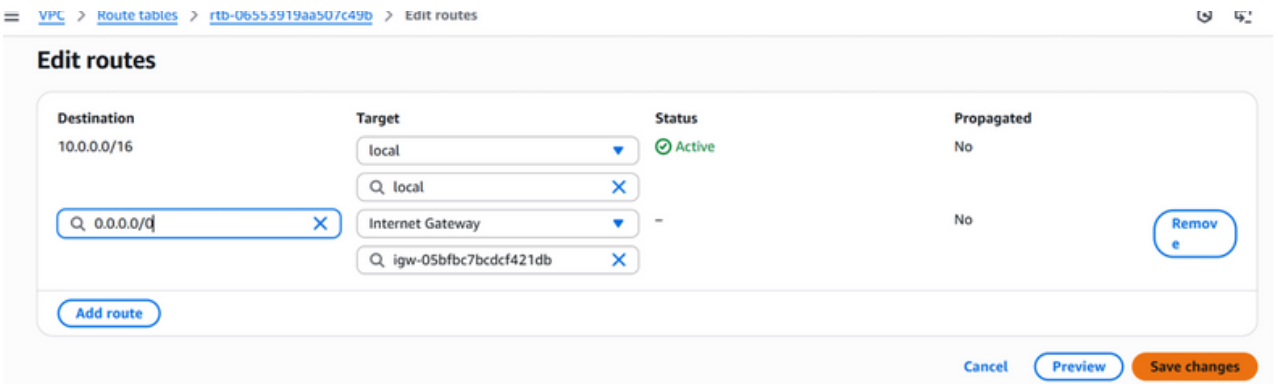
4. Create Route tables



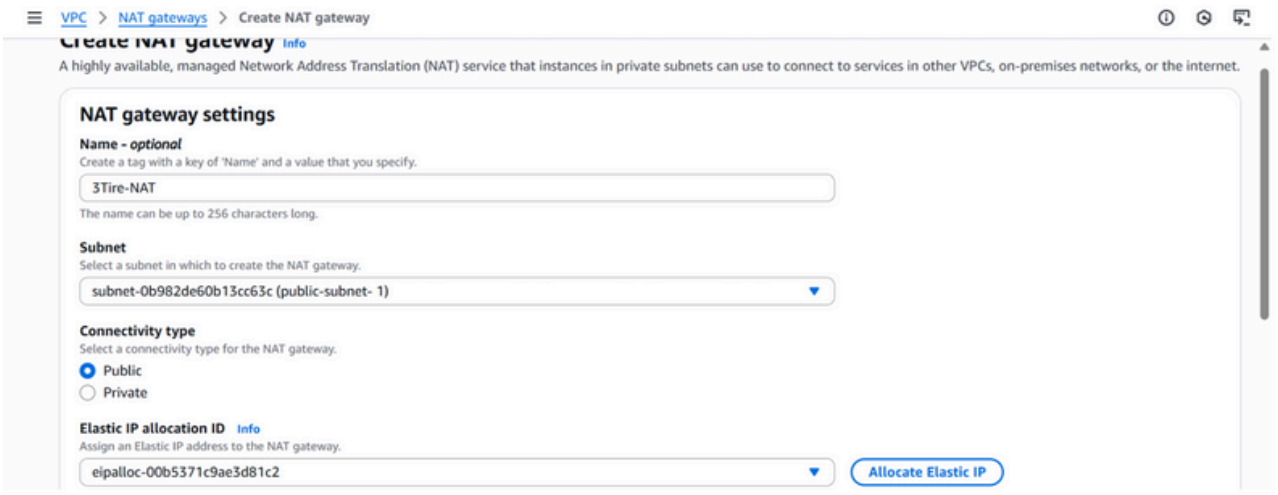
5. Associate Subnets with Route Tables



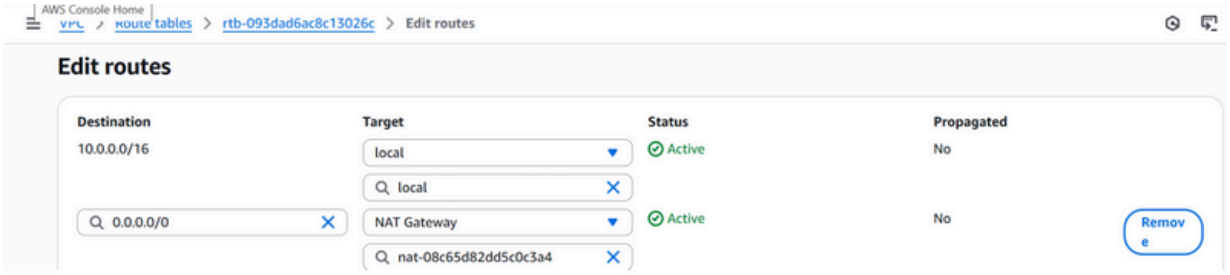
6. Attach Public Route Table to Internet Gateway (via Edit Routes)



7. Create NAT gateway

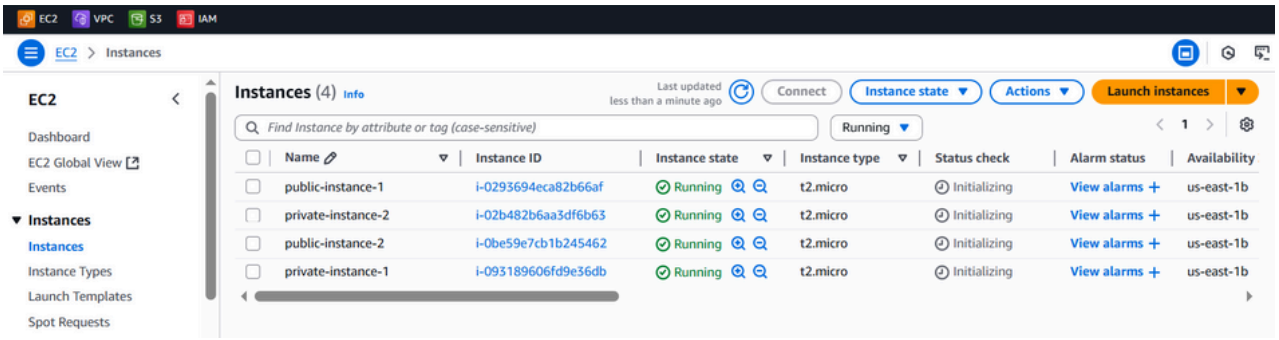


8. Attach Private Route Table to NAT Gateway



Step: 2 Launch an EC2 instance.

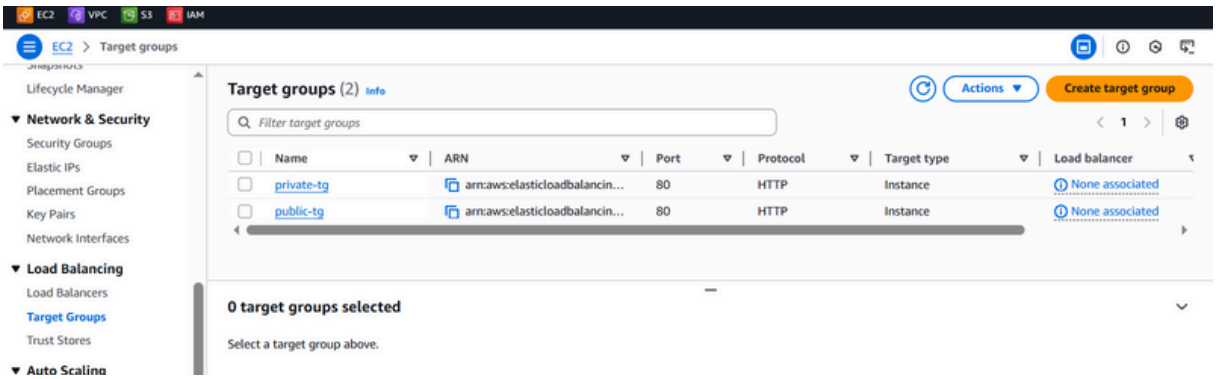
1. Launch EC2 Instances: 2 Public & 2 Private



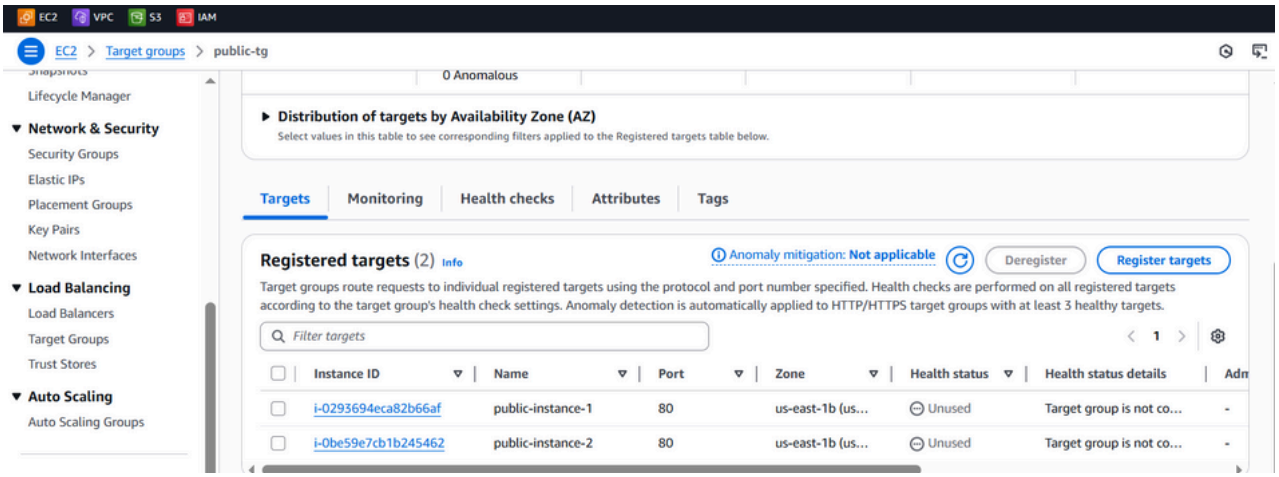
Step 3: Create Load Balancer

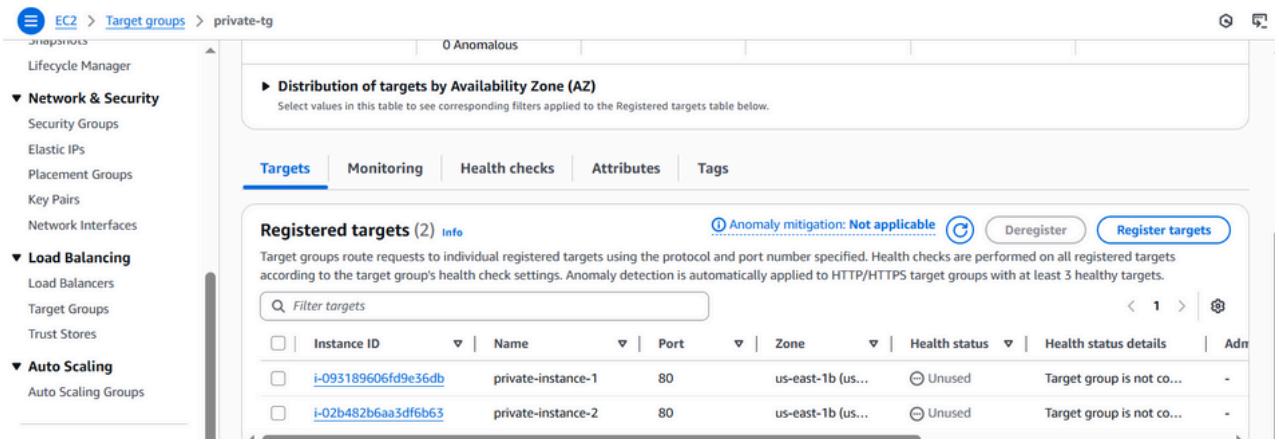
1. Create Two Target Groups

- Public Target Group
- Private Target Group



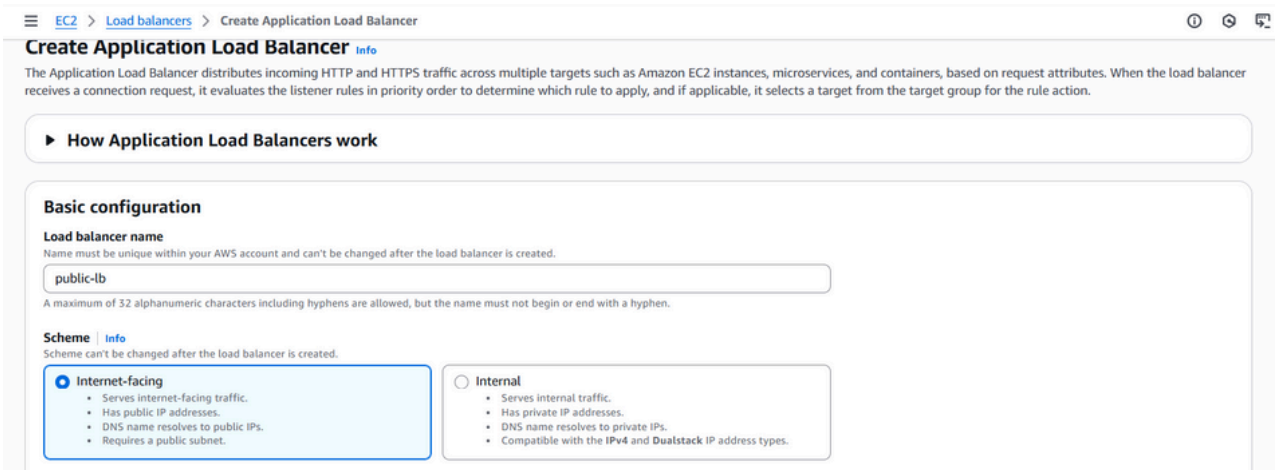
2. Associate EC2 Instances with Their Respective Target Groups

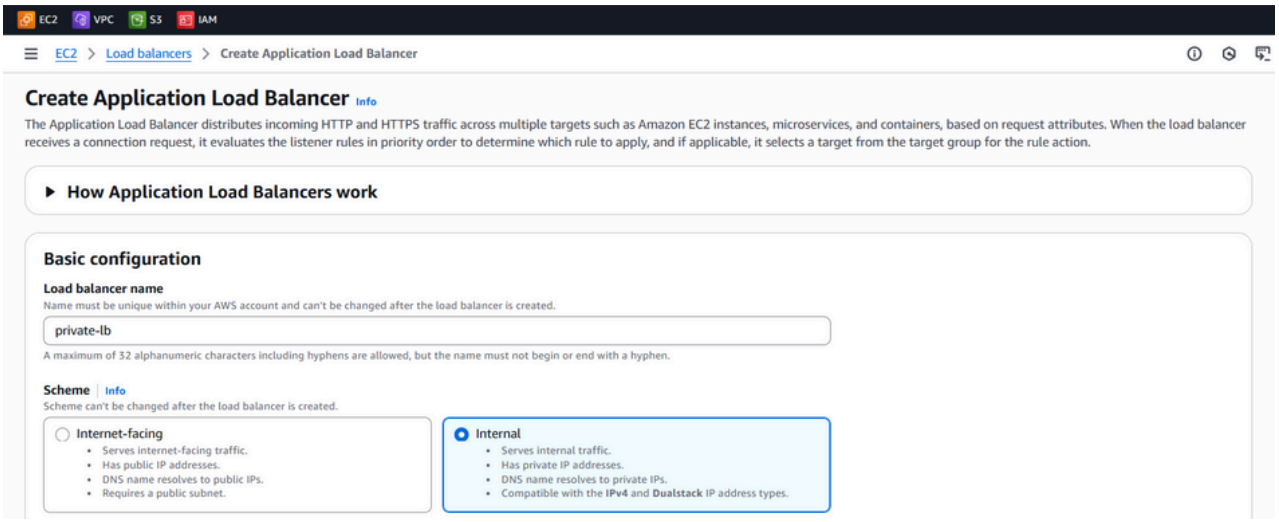




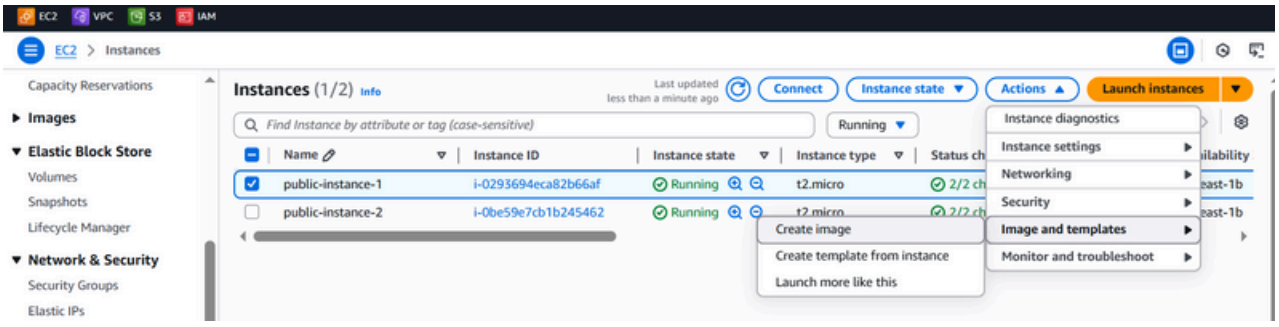
3. Create Application Load Balancers: Public & Private

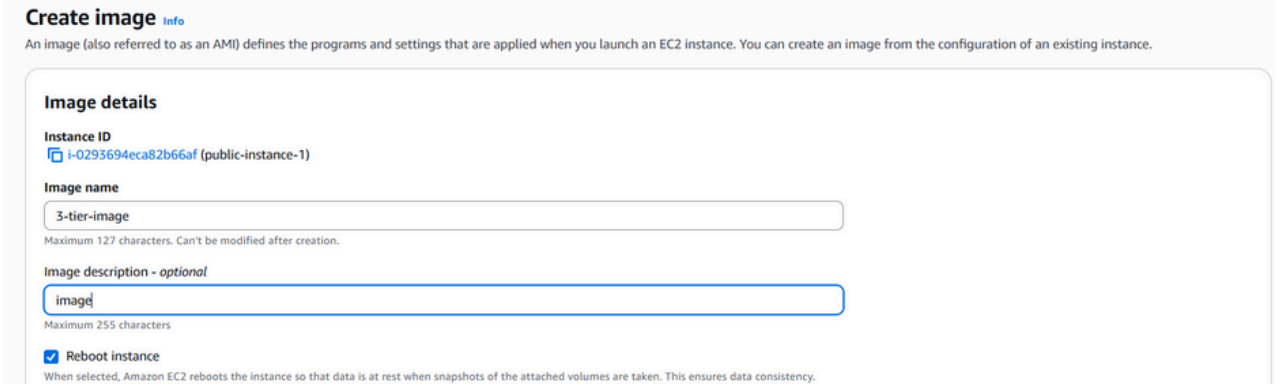
- Public Load Balancer – Internet-Facing
- Private Load Balancer – Internal-Facing





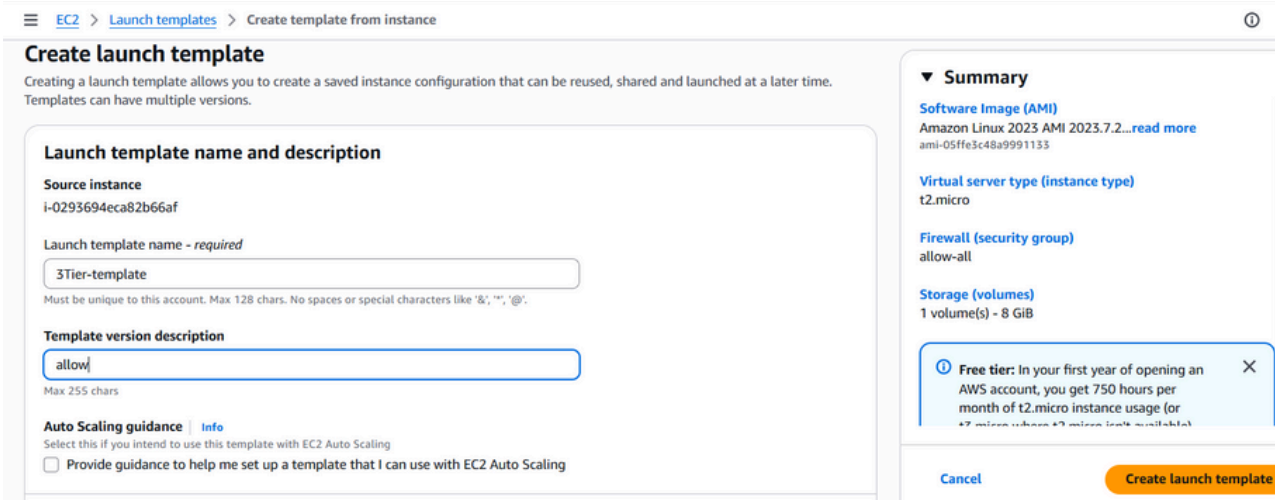
Step 4: Create an Amazon Machine Image (AMI)

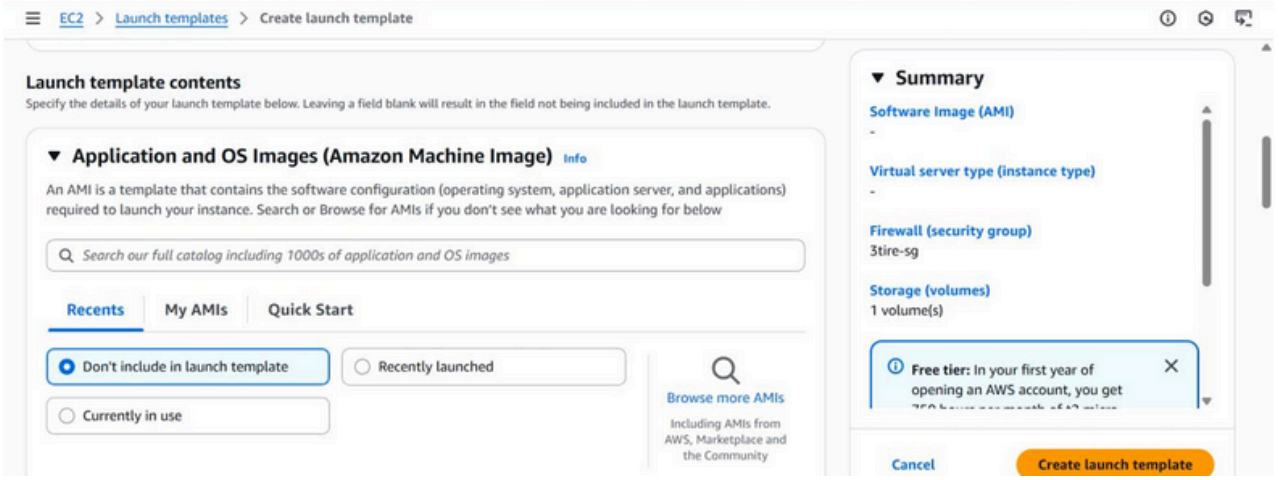




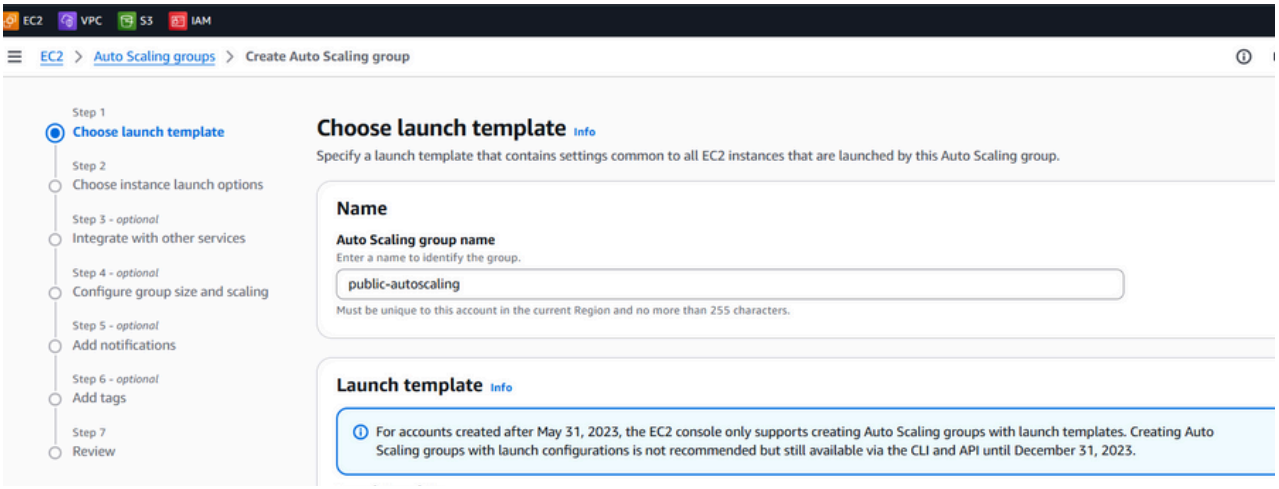
Step 5: Create an Auto Scaling Group

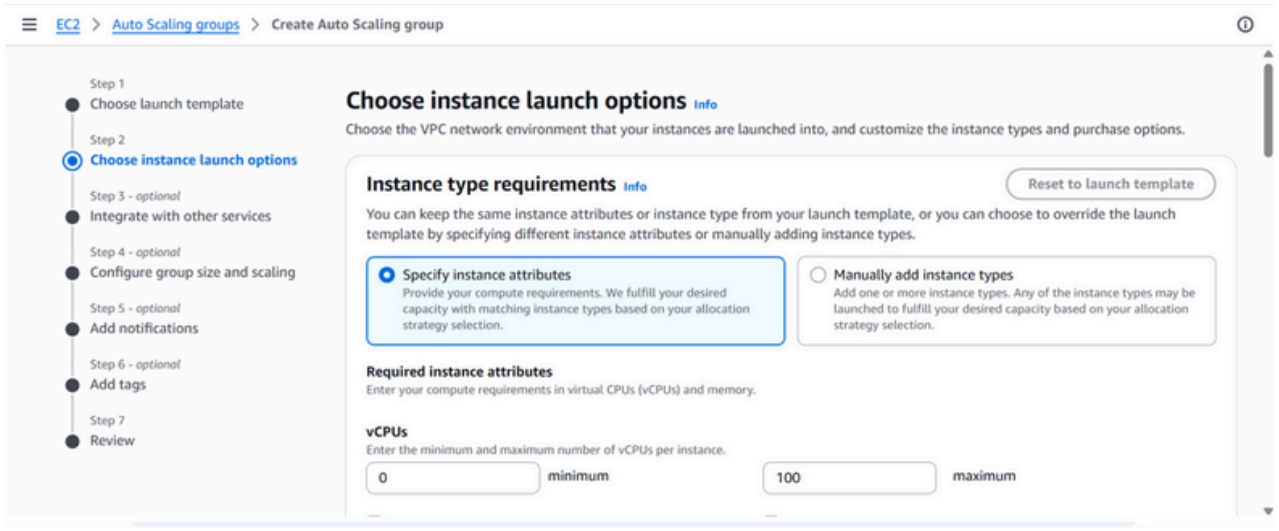
1. Create launch template - Public

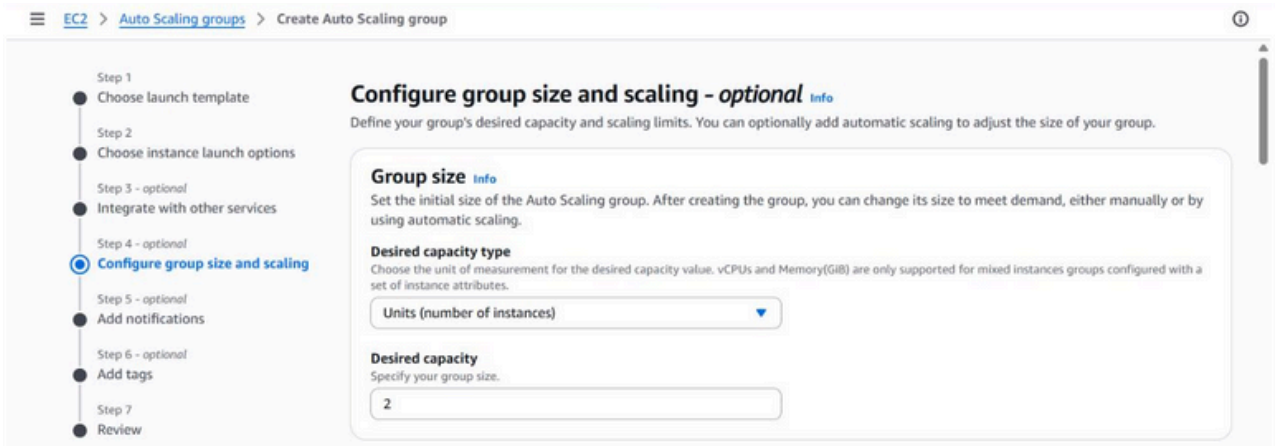


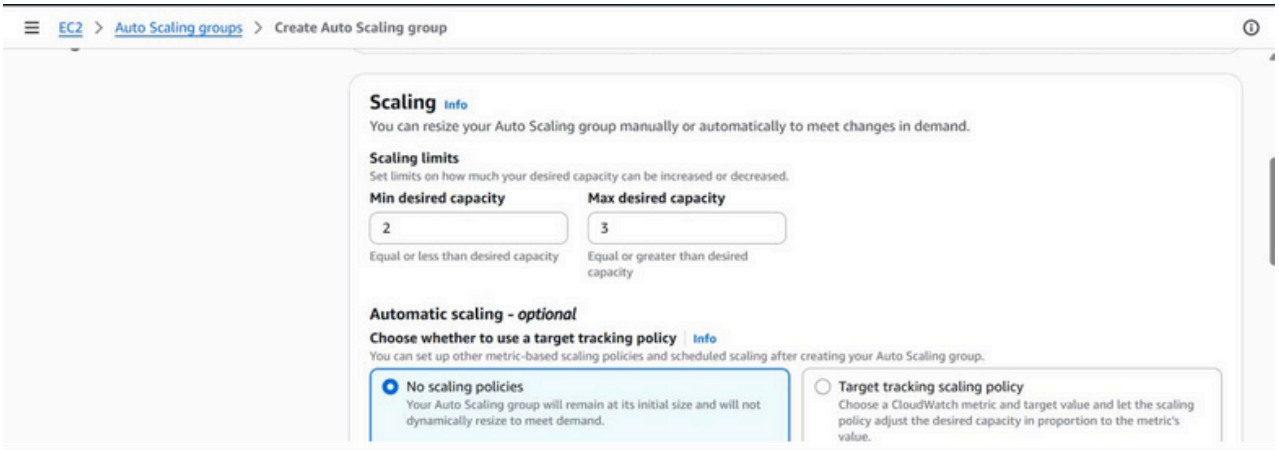


2. Create Auto Scaling Group - Public

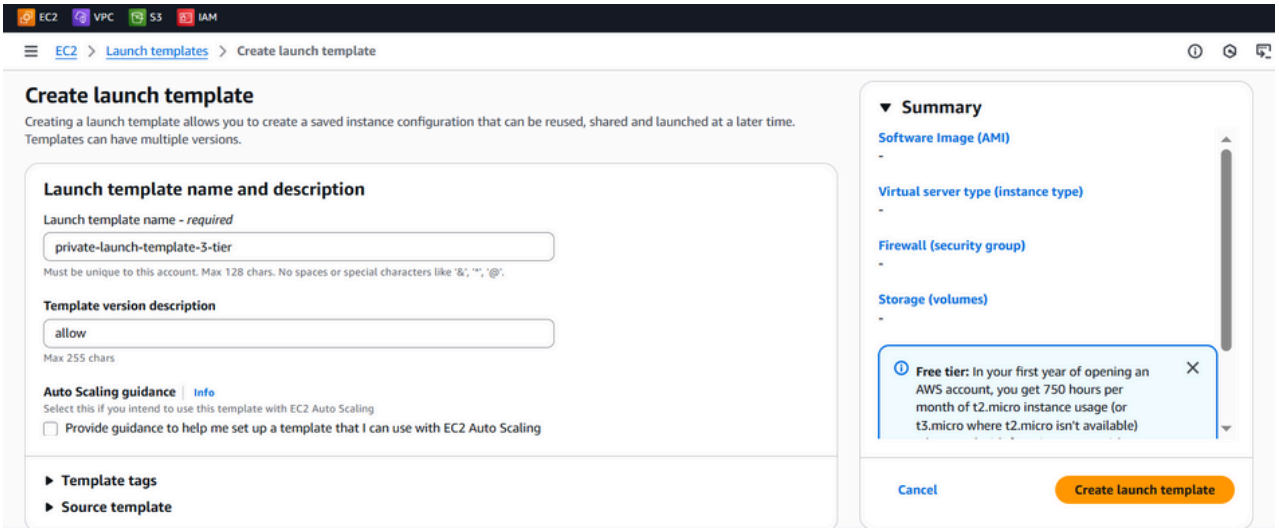


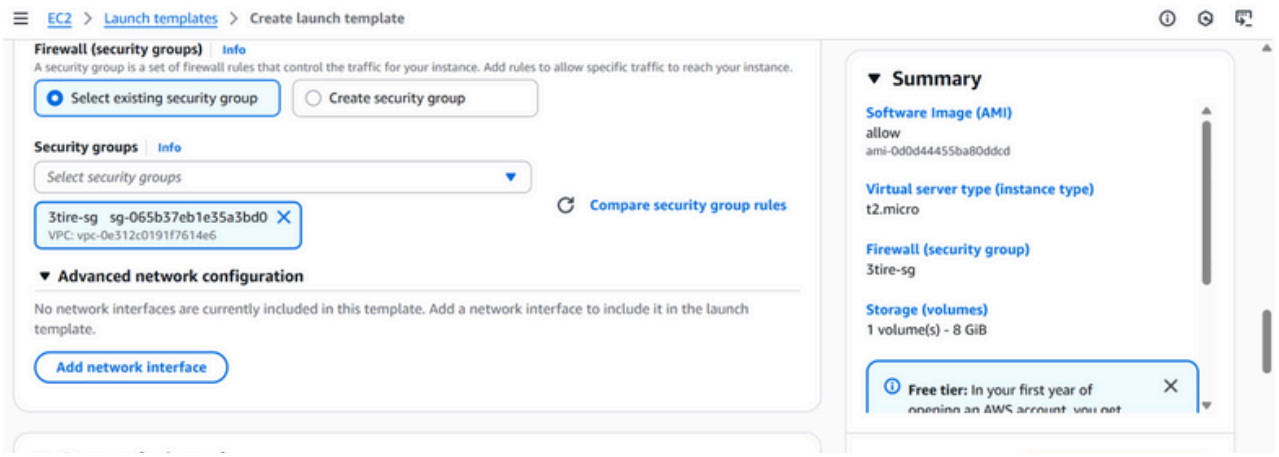




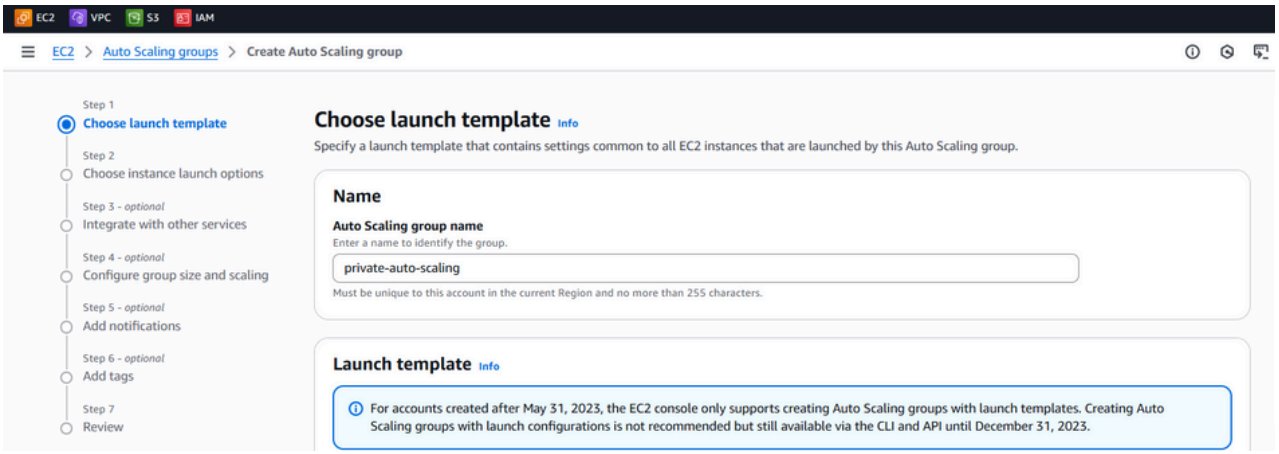


3. Create launch template - Private

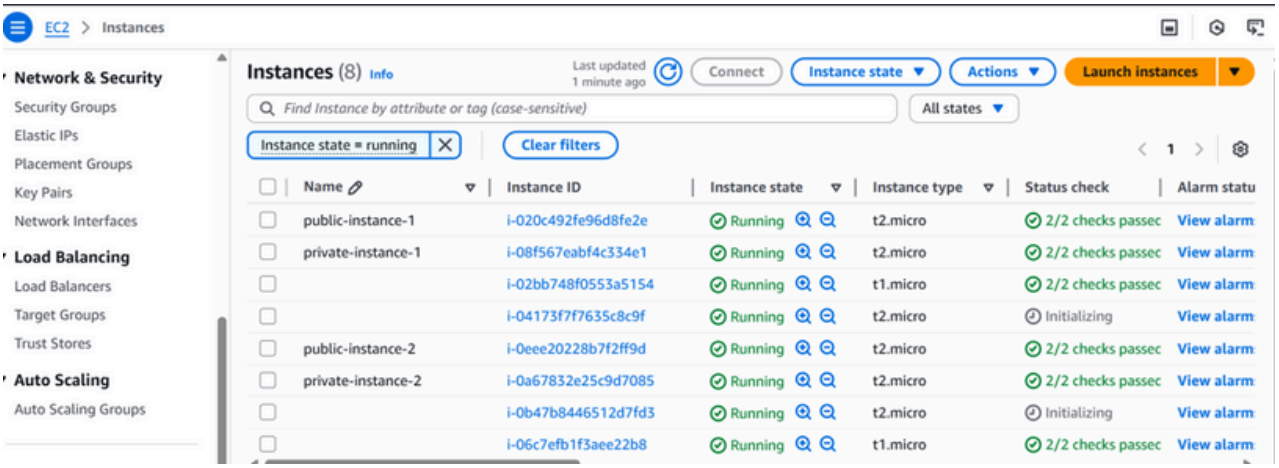




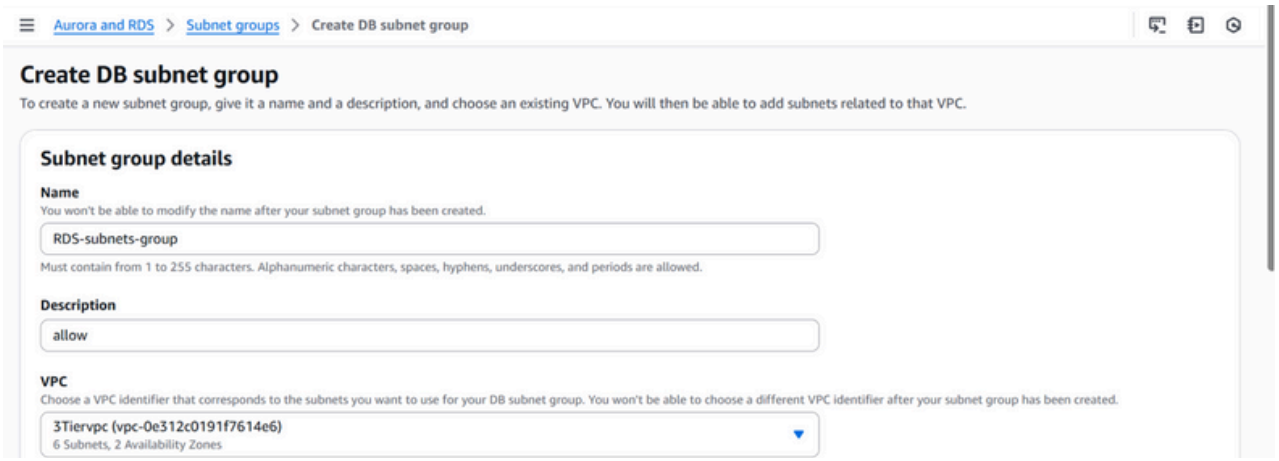
4. Create Autoscaling group - Private

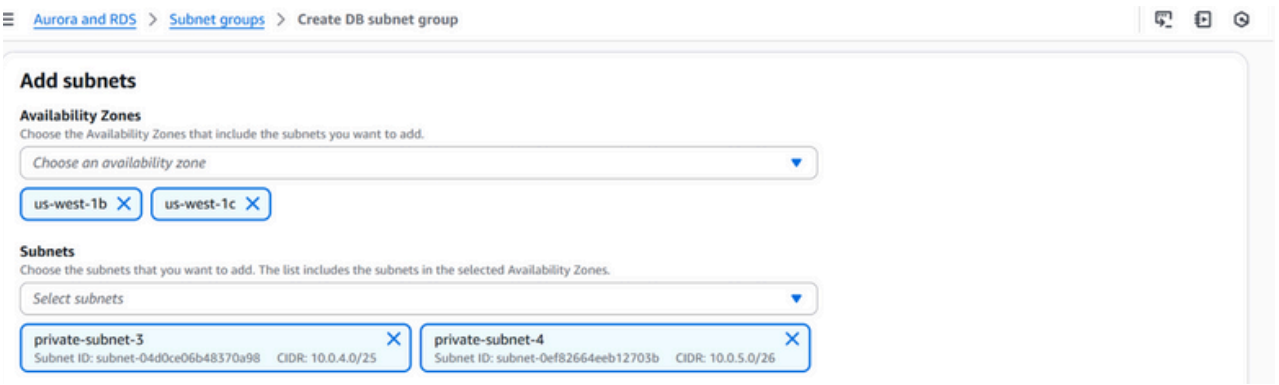


5. Auto Scaling Launches 4 Additional Instances (2 Public, 2 Private)

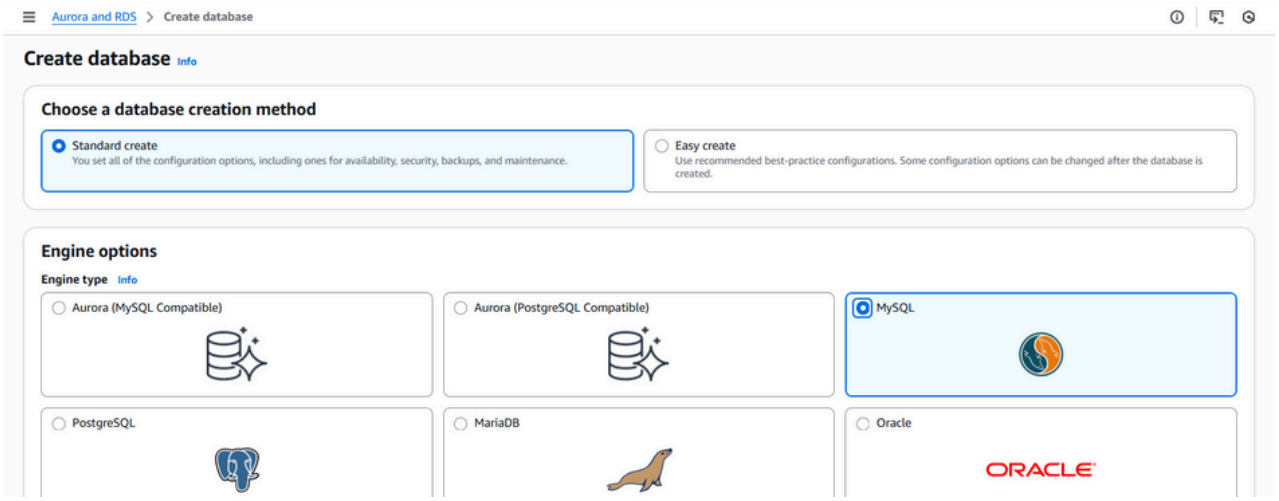


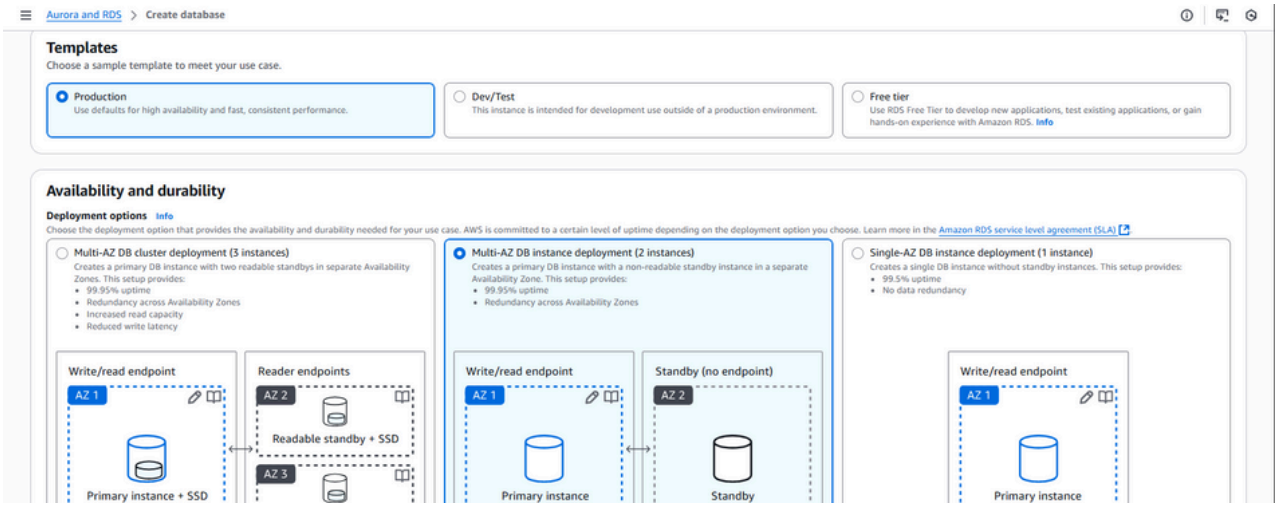
Step 6: Create Subnet Group

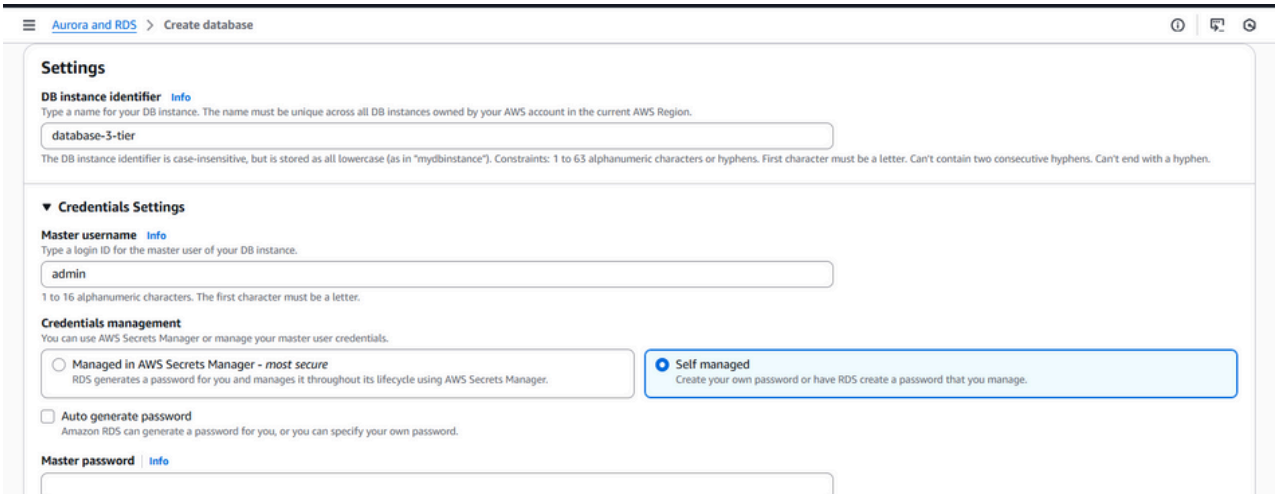


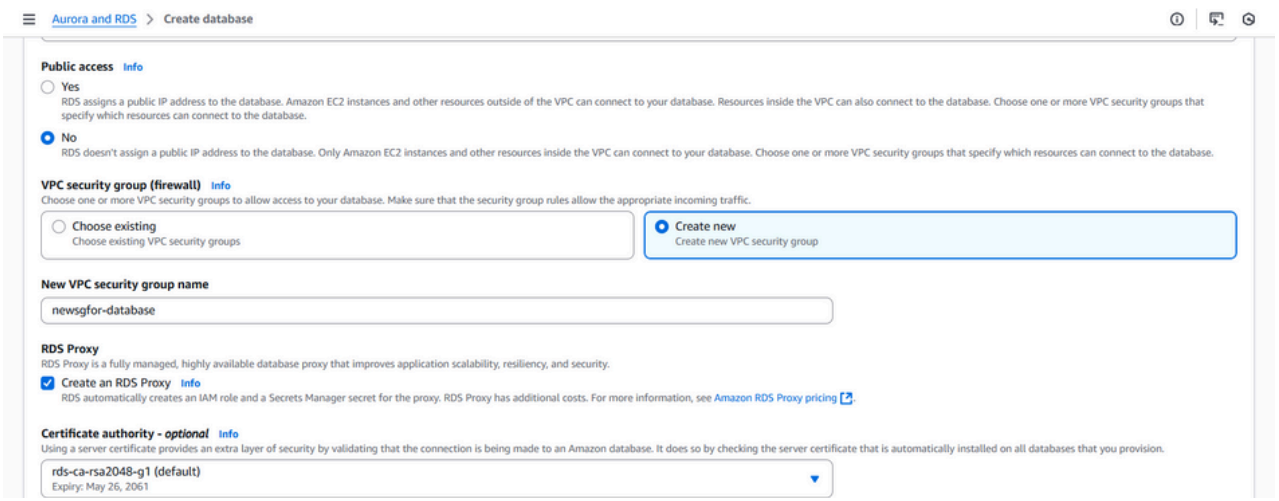


Step 7: Create Database (RDS Instance)

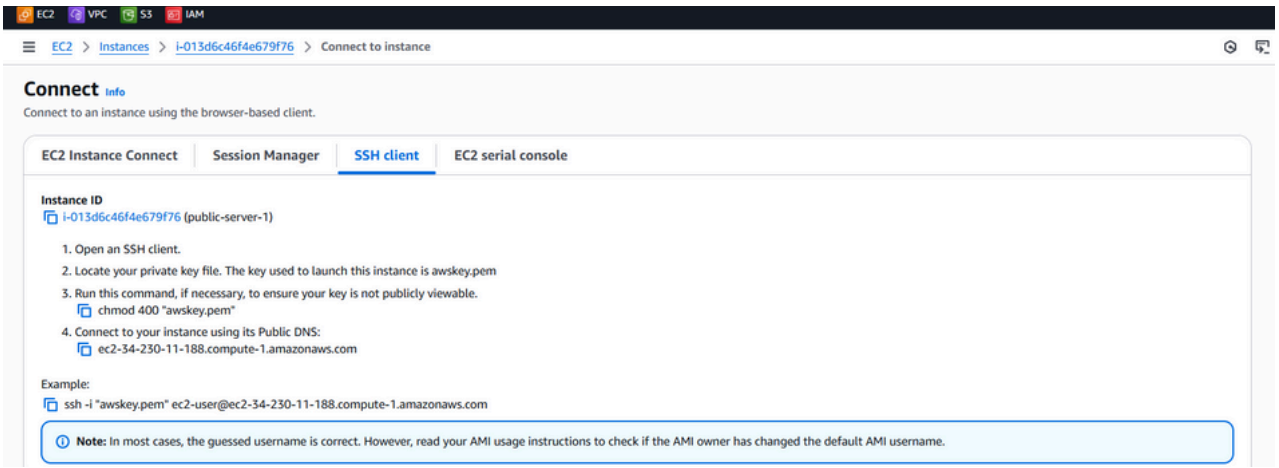

Aurora and RDS > Create database
Create database Info
Choose a database creation method
Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.
Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.
Engine options
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
PostgreSQL
MariaDB
Oracle
ORACLE


Aurora and RDS > Create database
Templates
Choose a sample template to meet your use case.
Production
Use defaults for high availability and fast, consistent performance.
Dev/Test
This instance is intended for development use outside of a production environment.
Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info
Availability and durability
Deployment options Info
Choose the deployment option that provides the availability and durability needed for your use case. AWS is committed to a certain level of uptime depending on the deployment option you choose. Learn more in the Amazon RDS service level agreement (SLA)
Multi-AZ DB cluster deployment (3 instances)
Creates a primary DB instance with two readable standbys in separate Availability Zones. This setup provides:
99.95% uptime
Redundancy across Availability Zones
Increased read capacity
Reduced write latency
Write/read endpoint
Reader endpoints
AZ 1
AZ 2
AZ 3
Readable standby + SSD
Primary instance + SSD
Multi-AZ DB instance deployment (2 instances)
Creates a primary DB instance with a non-readable standby instance in a separate Availability Zone. This setup provides:
99.95% uptime
Redundancy across Availability Zones
Write/read endpoint
Standby (no endpoint)
AZ 1
AZ 2
Primary instance
Standby
Single-AZ DB instance deployment (1 instance)
Creates a single DB instance without standby instances. This setup provides:
99.5% uptime
No data redundancy
Write/read endpoint
AZ 1
Primary instance


Aurora and RDS > Create database
Settings
DB instance identifier Info
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.
database-3-tier
The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.
Credentials Settings
Master username Info
Type a login ID for the master user of your DB instance.
admin
1 to 16 alphanumeric characters. The first character must be a letter.
Credentials management
You can use AWS Secrets Manager or manage your master user credentials.
Managed in AWS Secrets Manager - most secure
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.
Self managed
Create your own password or have RDS create a password that you manage.
Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.
Master password Info

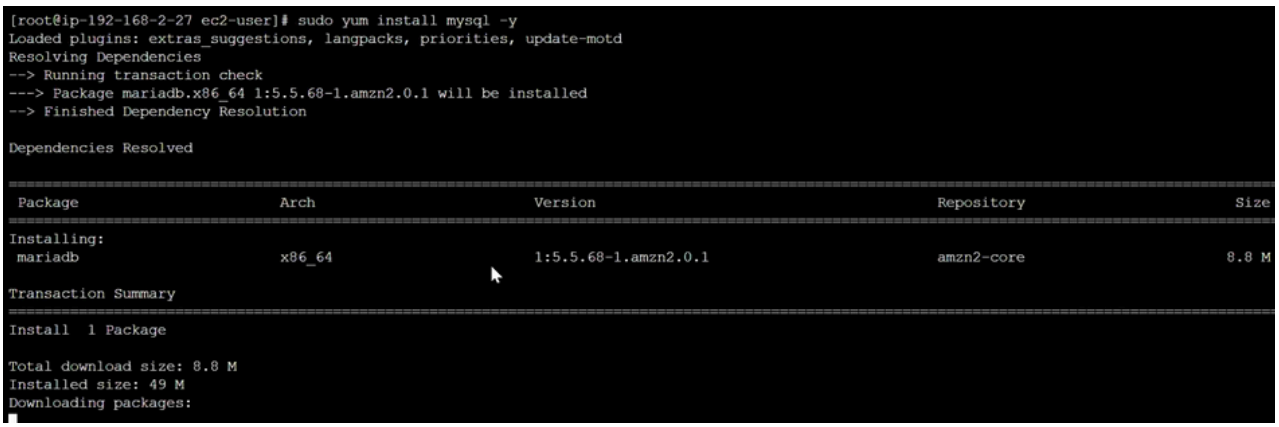


Step 8: Establish Connection



After connecting to the server, run:

- sudo -i
- apt update -y
- sudo apt install mysql-server -y



```
[root@ip-192-168-2-27 ec2-user]# mysql -h database-1.c380a08uukyc.ap-south-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 28
Server version: 8.0.35 Source distribution

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

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

```
[root@ip-192-168-2-27 ec2-user]# mysql -h database-1.c380a08uukyc.ap-south-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 28
Server version: 8.0.35 Source distribution

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

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> CREATE DATABASE webappdb;
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| webappdb           |
+--------------------+
5 rows in set (0.00 sec)

MySQL [(none)]>
```

```
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| webappdb           |
+--------------------+
5 rows in set (0.00 sec)

MySQL [(none)]> USE webappdb;
Database changed
MySQL [webappdb]> clear
MySQL [webappdb]> CREATE TABLE IF NOT EXISTS transactions(
    ->   id INT NOT NULL AUTO_INCREMENT,
    ->   amount DECIMAL(10,2),
    ->   description VARCHAR(100),
    ->   PRIMARY KEY(id)
    -> );
Query OK, 0 rows affected (0.04 sec)

MySQL [webappdb]> SHOW TABLES;
+--------------------+
| Tables_in_webappdb |
+--------------------+
| transactions       |
+--------------------+
1 row in set (0.02 sec)

MySQL [webappdb]>
```