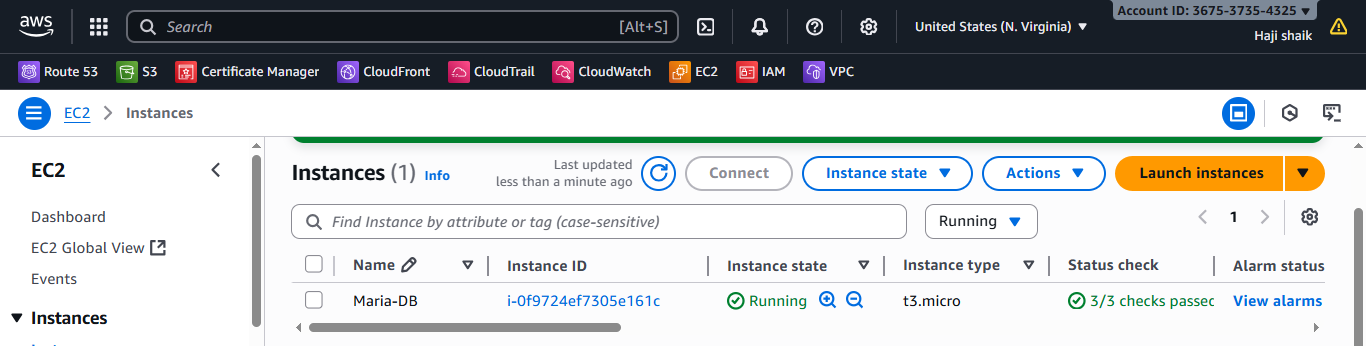
# RDS TASK

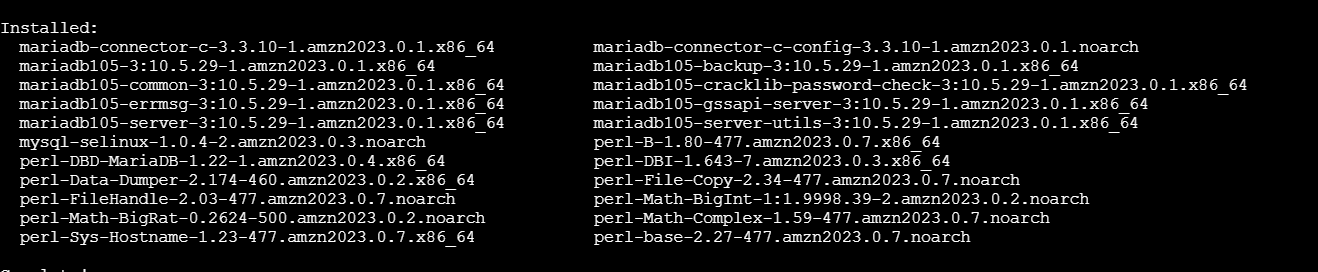
1. Create MariaDB DB on EC2.

* Go ec2 console and create one ec2 instance name with Maria-db



* Connect to the server using a command of ssh.
* Then use a command of sudo dnf install mariadb105-server -y
* Then sudo systemctl start nd enable mariadb.





* Then “sudo mysql\_secure\_installation”
* Follow prompts:

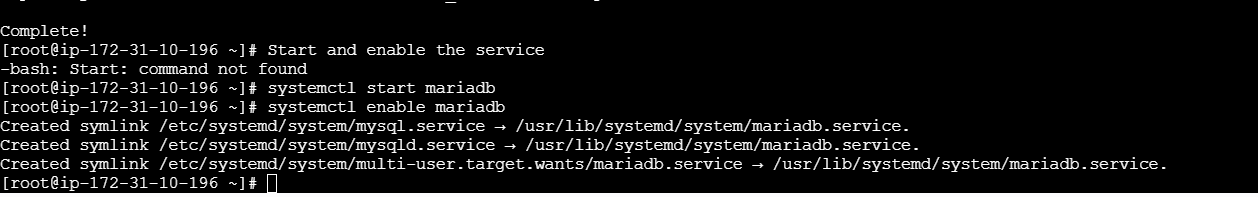
Set a **root password**

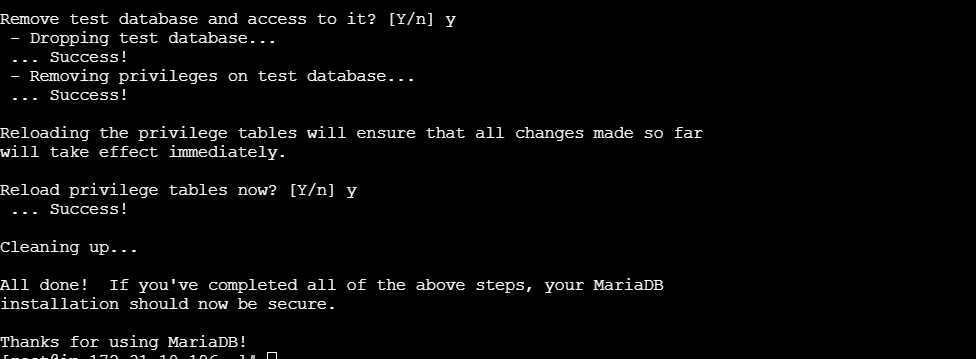
Remove anonymous users → **Yes**

Disallow root login remotely → **Yes**

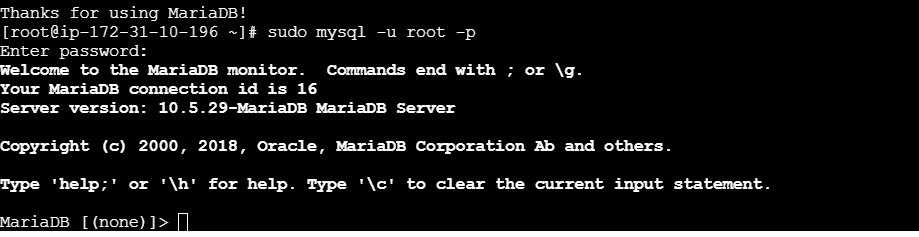
Remove test database → **Yes**

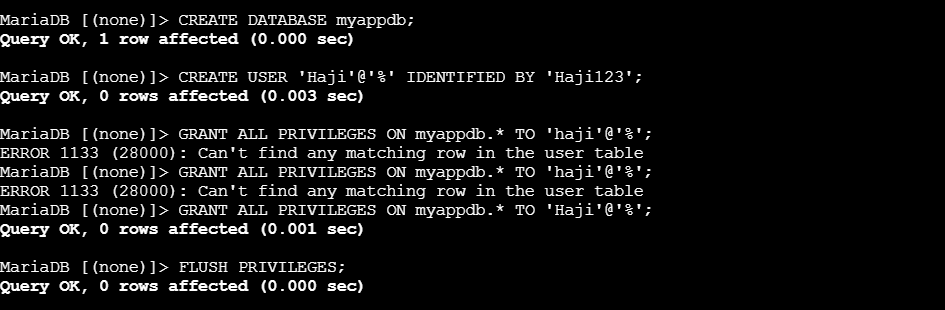
Reload privilege tables → **Yes**





* Then it will enter in maria db.
* Then create data base.
* “CREATE DATABASE myappdb”
* Then use a command of create a user.
* CREATE DATABASE myappdb;
* CREATE USER 'appuser'@'%' IDENTIFIED BY 'StrongPassword123';
* GRANT ALL PRIVILEGES ON myappdb.\* TO 'appuser'@'%';
* FLUSH PRIVILEGES;
* EXIT;





* Then follow the steps:

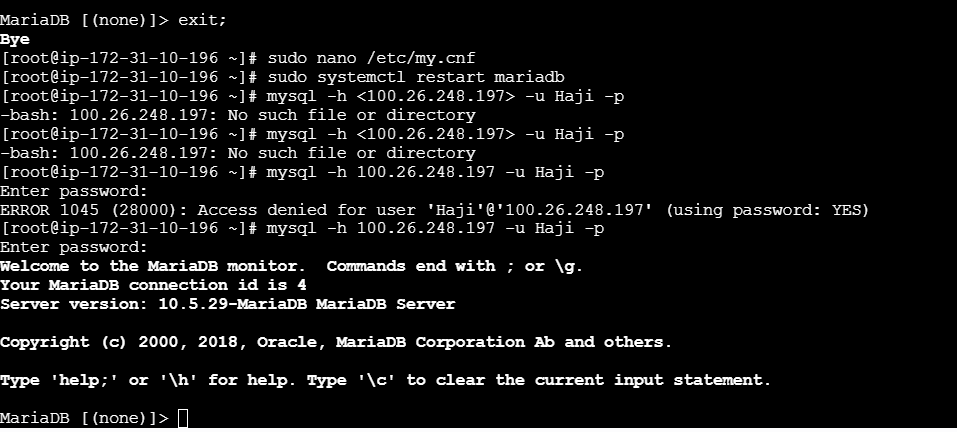
sudo nano /etc/my.cnf

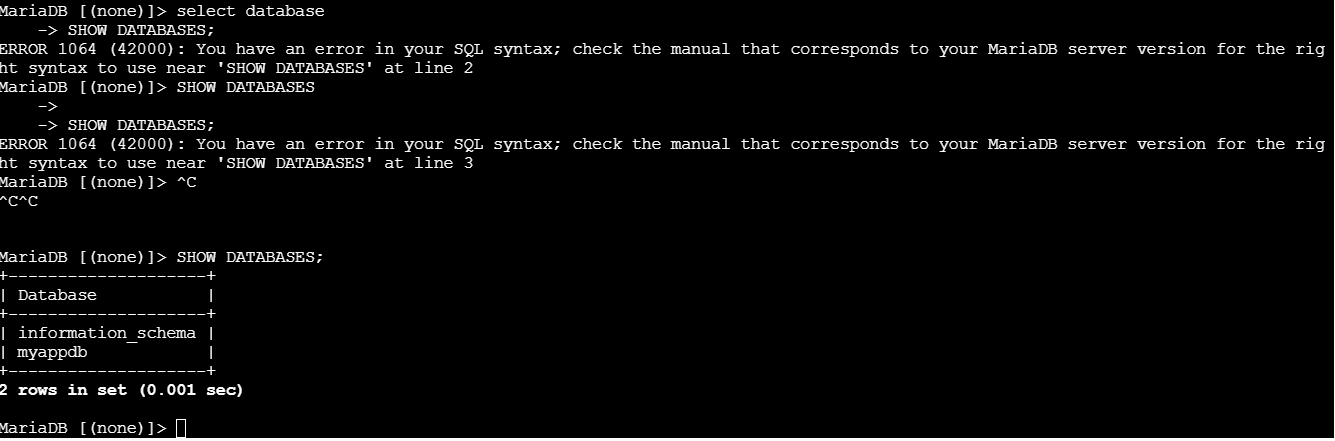
sudo systemctl restart mariadb

sudo systemctl restart mariadb

Ccheck for connection:

mysql -h <EC2-Public-IP> -u appuser -p





1. Insert some dummy data.

* If you are logged in **as that user**, say:
* mysql -u auser -p
* Then run:
* SHOW DATABASES;
* You’ll only see databases where that user has **permissions**.
* If it shows:
* Empty set (0.00 sec)
* that means you haven’t granted privileges yet.

## 🔐 ****How to Grant Access to a Database****

* Log in as **root** (or an admin) and run:
* GRANT ALL PRIVILEGES ON myappdb.\* TO 'auser'@'%';
* FLUSH PRIVILEGES;
* Now, if you log back in as auser:
* mysql -u auser -p
* and run:
* SHOW DATABASES;

Then create dummy data in instance:

mysql -u Haji -p

Then enter your password.

## 🧱 Step 2: Select Your Database

First, check available databases:

SHOW DATABASES;

Then choose your working one (for example myappdb):

USE myappdb;

✅ You should see:

Database changed

## 🧩 Step 3: Create a Table

Let’s create a simple table — for example, a **users** table:

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(50),

email VARCHAR(100),

age INT

);

✅ You should see:

Query OK, 0 rows affected

## 🧍 Step 4: Insert Dummy Data

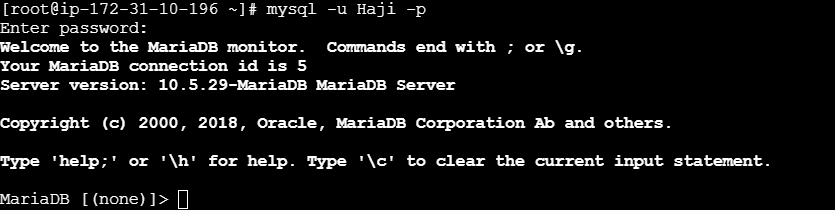
Now add a few rows of fake data:

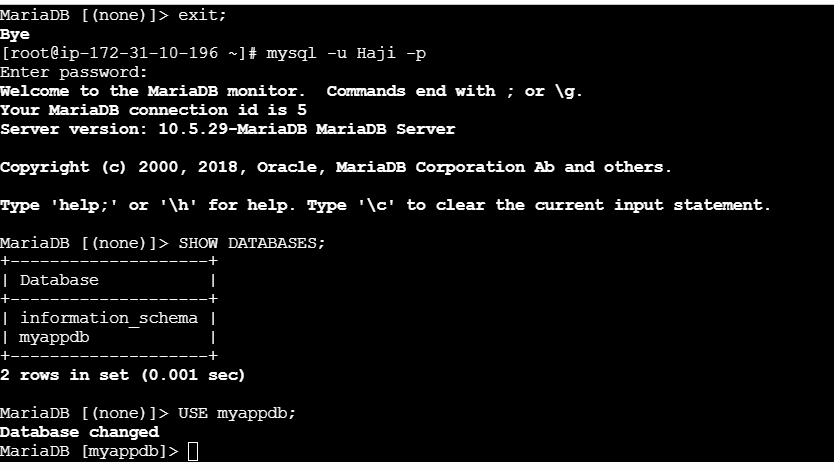
INSERT INTO users (name, email, age)VALUES

('Alice', 'alice@example.com', 25),

('Bob', 'bob@example.com', 30),

('Charlie', 'charlie@example.com', 28);











1. Take the backup of dummy data on EC2.

Exit MariaDB

The mysqldump utility lets you export your database (schema + data) into a .sql file.

### Basic syntax:

mysqldump -u <username> -p <database\_name> > <backup\_filename>.sql

### Example for your setup:

mysqldump -u Haji -p myappdb > myappdb\_backup.sql

It will ask for your **MariaDB password**, then create a file named myappdb\_backup.sql in your current directory.

✅ You can check it:

ls -l myappdb\_backup.sql

## 🧩 ****Step 3: Verify Backup Contents****

To view the backup file:

cat myappdb\_backup.sql | head -20

You’ll see SQL statements like:

CREATE TABLE `users` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(50) DEFAULT NULL,

`email` varchar(100) DEFAULT NULL,

`age` int(11) DEFAULT NULL,

PRIMARY KEY (`id`)

);

INSERT INTO `users` VALUES (1,'Alice','alice@example.com',25),(2,'Bob','bob@example.com',30),(3,'Charlie','charlie@example.com',28);

## 🧱 ****Step 4: Restore Backup (if needed)****

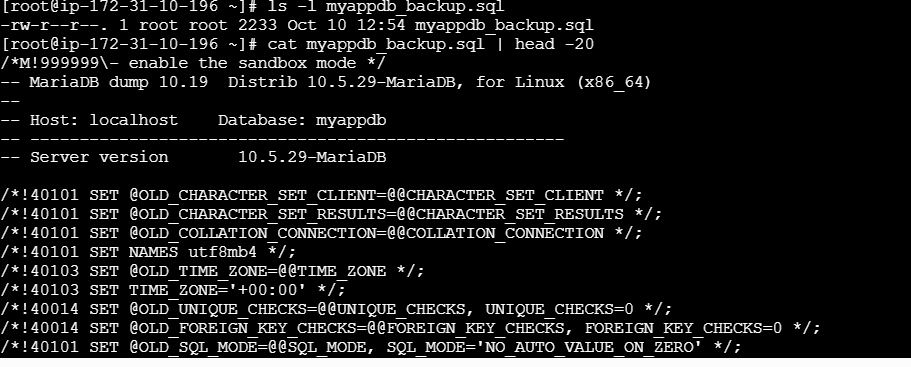
If you ever need to restore it:

mysql -u Haji -p myappdb < myappdb\_backup.sql

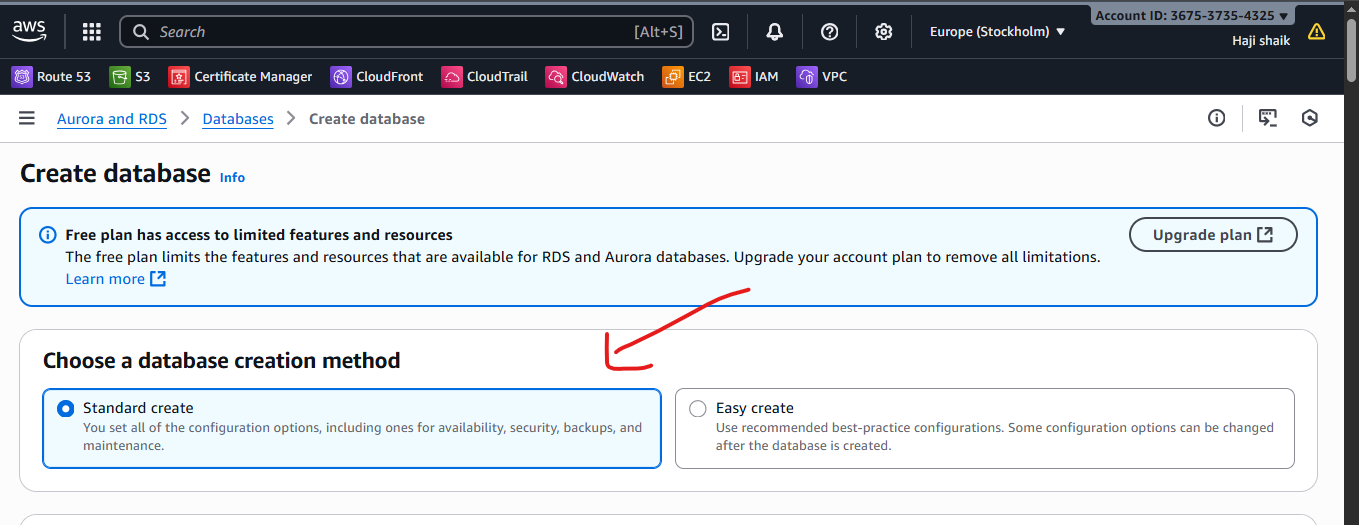
## 🧰 Optional: Backup All Databases

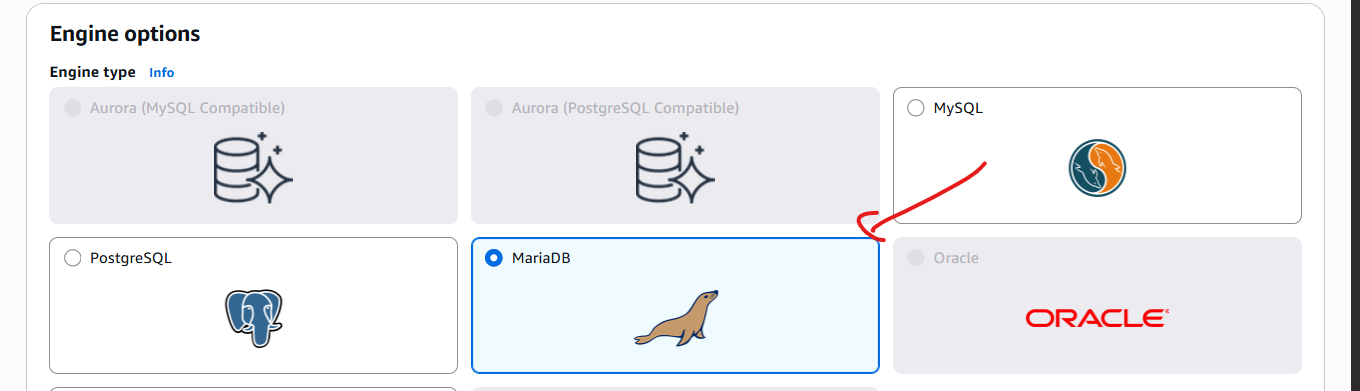
If you want to back up everything:

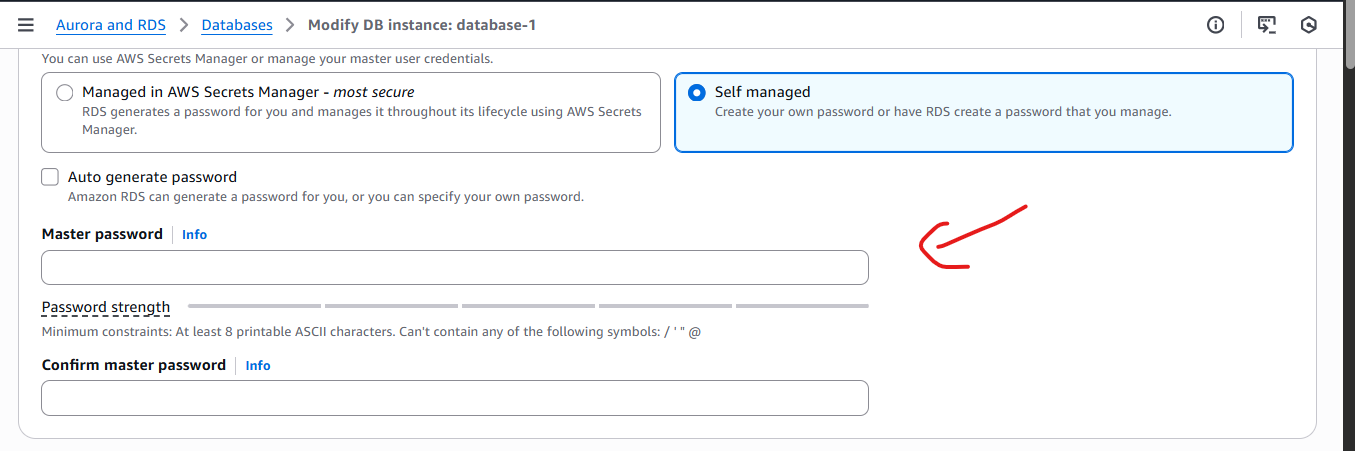
mysqldump -u root -p --all-databases > all\_db\_backup.sql



1. Launch MariaDB RDS instance.







### ****Step 2: Create Database****

Click **Databases** → **Create database**.

Select **Standard Create**.

Under **Engine options**, choose:

**Engine type**: MariaDB

**Version**: Select the latest stable version (e.g., 11.x).

### ****Step 3: Configure Database****

**DB instance identifier**: e.g., my-mariadb-rds

**Master username**: e.g., admin

**Master password**: Choose a strong password and confirm.

**DB instance class**: e.g., db.t3.micro for free tier or testing.

**Storage**: Choose allocated storage (e.g., 20 GB) and enable auto-scaling if desired.

### ****Step 4: Connectivity****

**Virtual Private Cloud (VPC)**: Select your existing VPC or default.

**Subnet group**: Default or custom

**Public access**: Select **Yes** if you want to connect from outside the VPC (for testing; otherwise, keep No).

**VPC security group**: Choose an existing one or create a new group that allows **port 3306** for MariaDB.

**Availability zone**: keep N.verginia region.

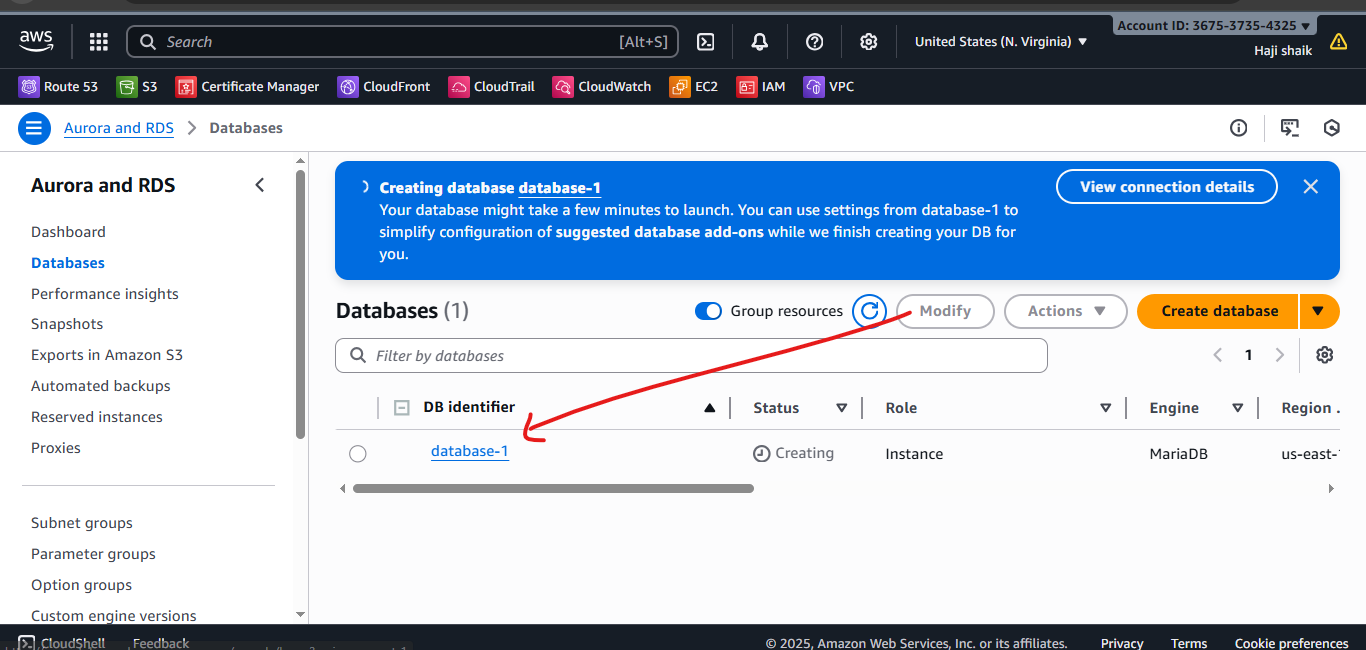
### ****Step 5: Additional Configuration****

**Database name**: e.g., myappdb (optional, can create later).

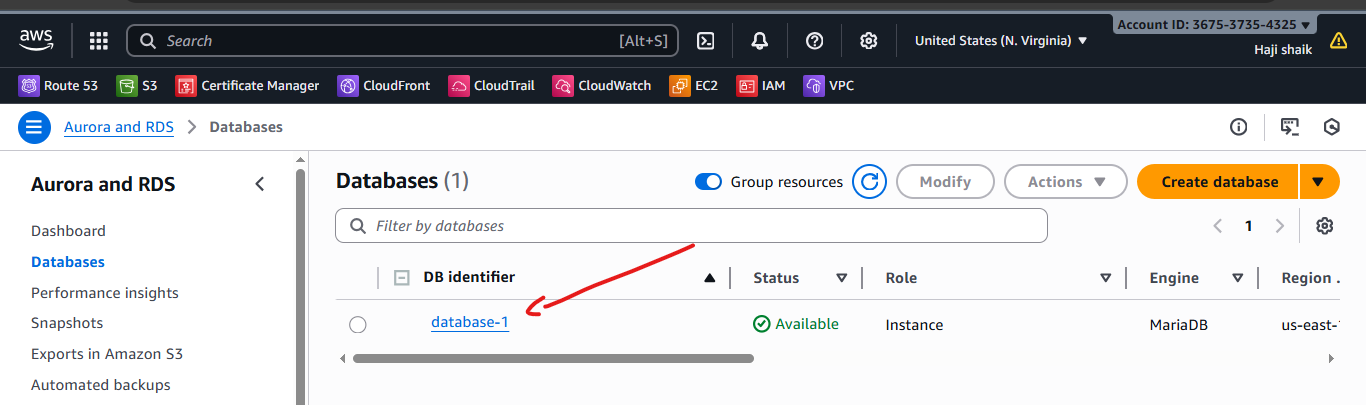
**Backup retention**: e.g., 7 days (optional, recommended for production).

**Encryption**: Enable if required.

**Monitoring & Maintenance**: Optional.



It will take Atlest 15 mins to create this data base



1. Migrate database from EC2 to RDS.

* Create one instance in ec2 the name of maria-db.
* Then go to **Aurora and RDS.**
* Then create one RDS data base in rds database.
* Gave name database-2.
* Then select maria-DB image then selcet family and choose defult vpc if uh want.
* Gave a master name As A **ADMIN.**
* **Then** use ur strong password.
* And create a data base.
* Then use follow the commands;

mysqldump -u root -p myappdb > myappdb\_backup.sql

mysql -h <RDS-endpoint> -u <rds\_username> -p

CREATE DATABASE myappdb;

EXIT;

mysql -h <RDS-endpoint> -u <rds\_username> -p myappdb < myappdb\_backup.sql

mysql -h <RDS-endpoint> -u <rds\_username> -p

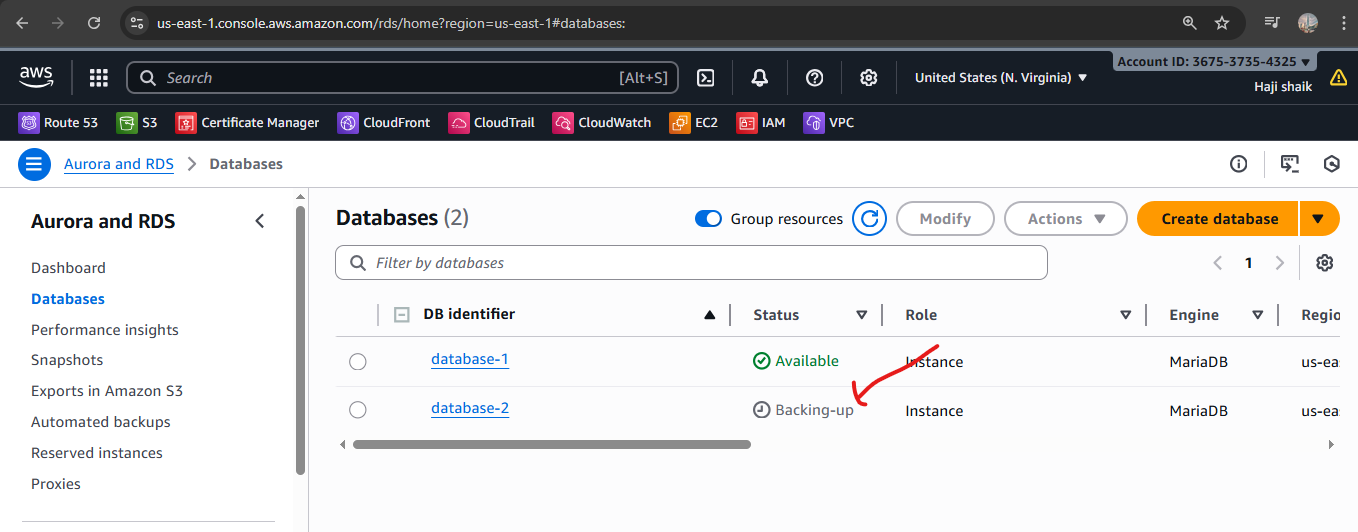
SHOW DATABASES;

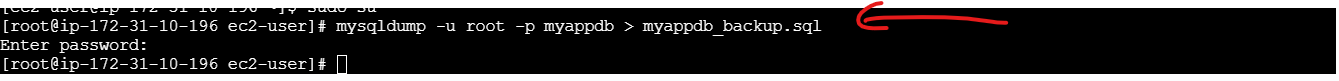
USE myappdb;

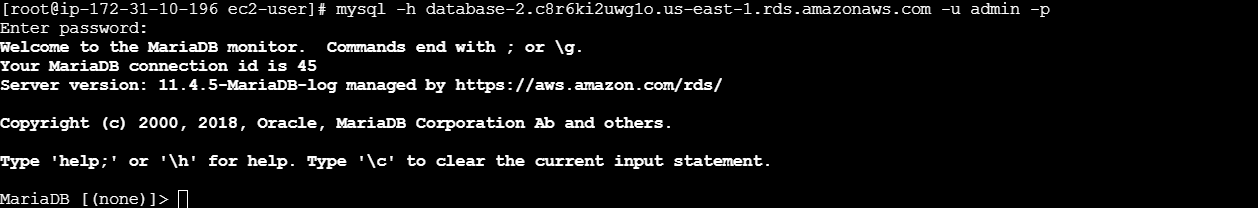
SHOW TABLES;

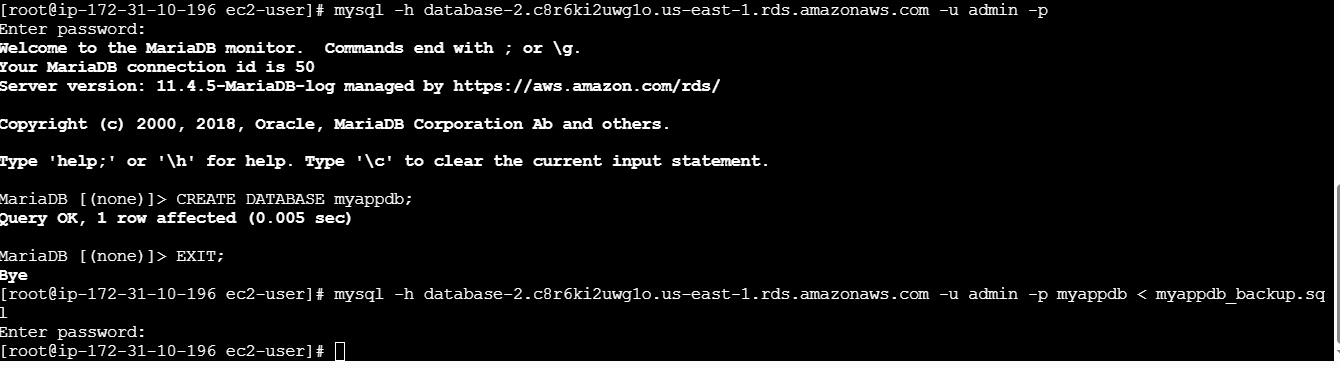
SELECT \* FROM users;

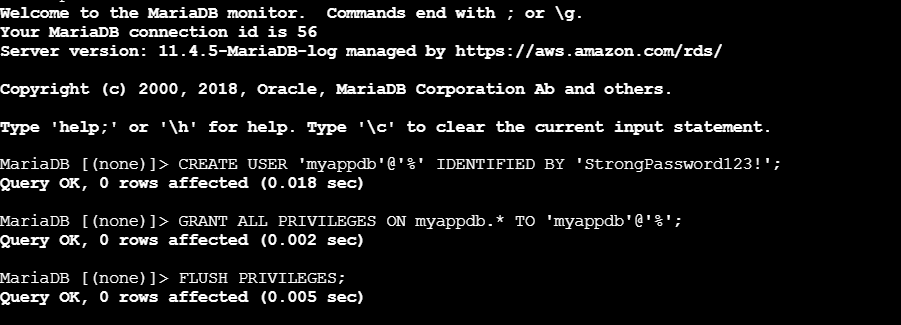
Here the steps for migarate data base from ec2 to RDS

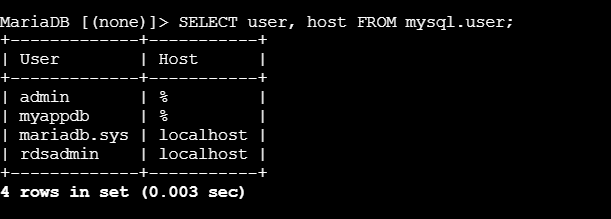


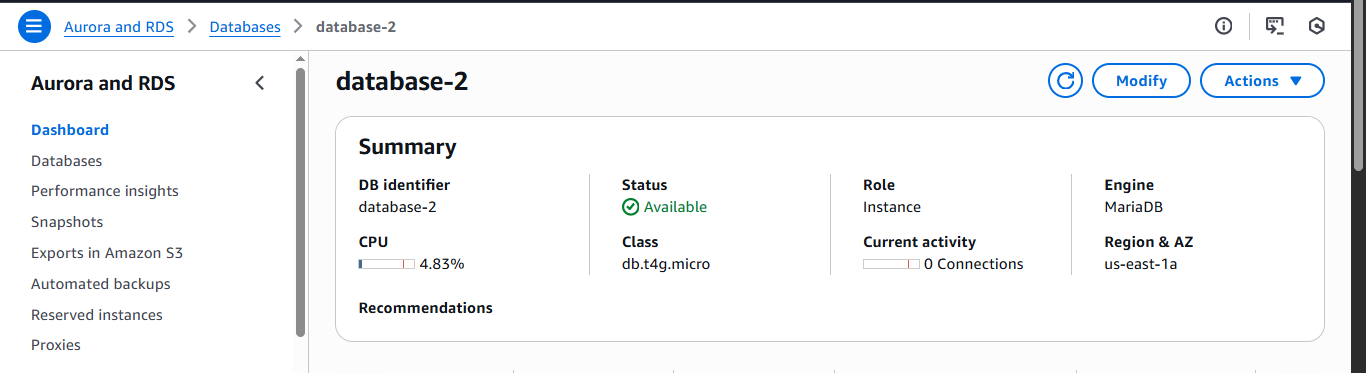


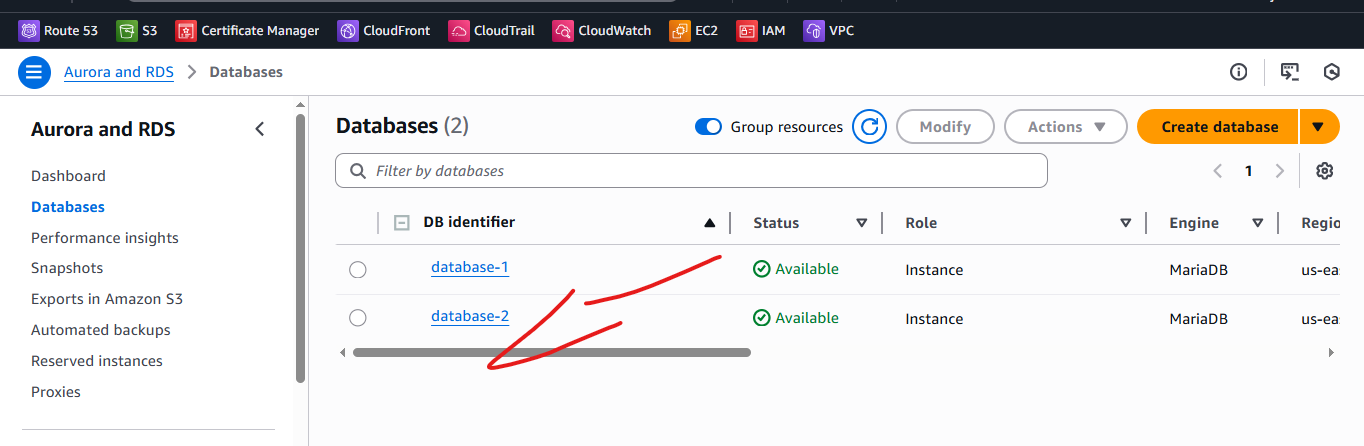






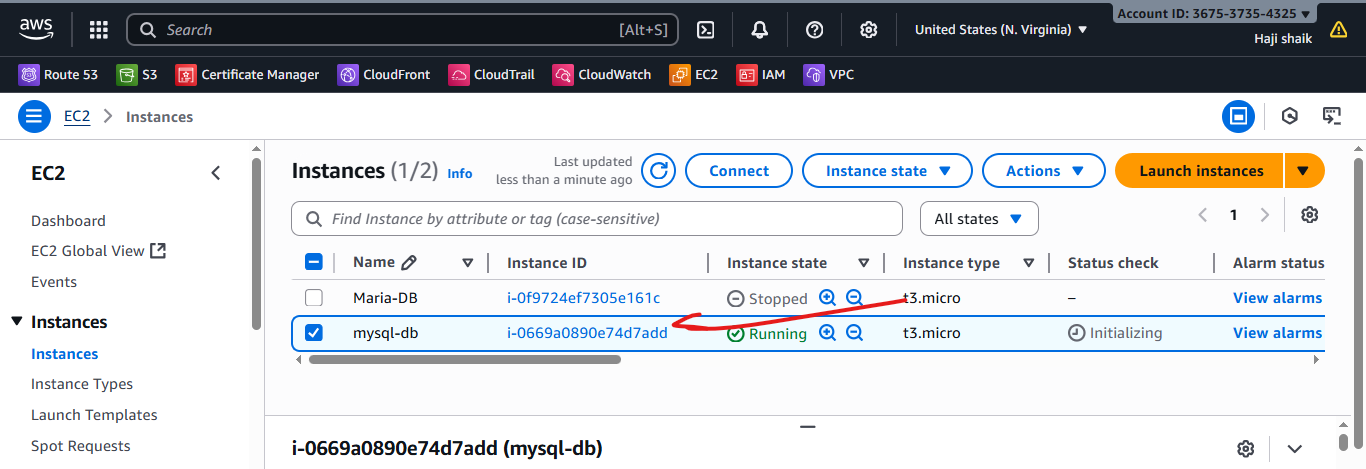


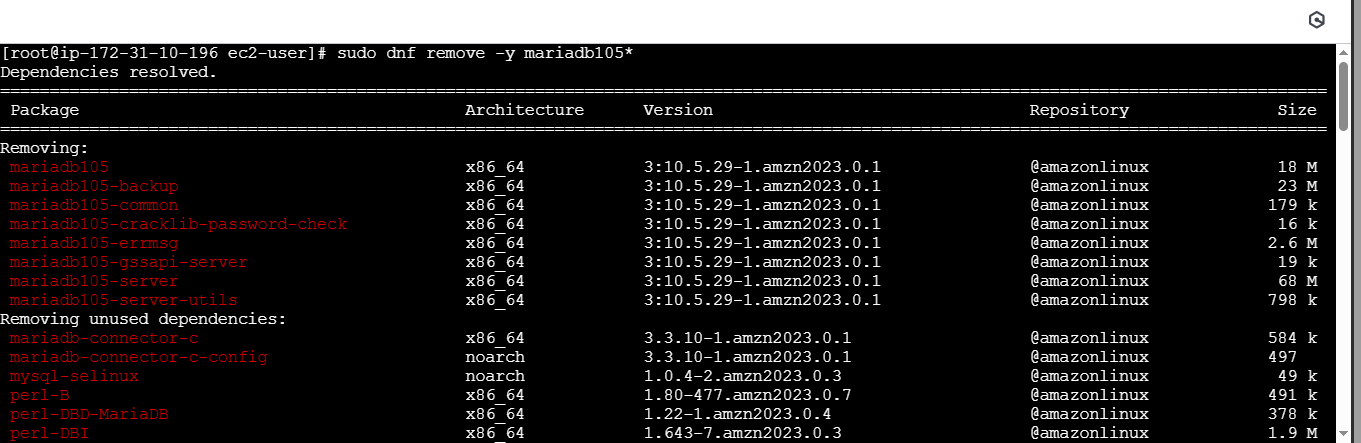


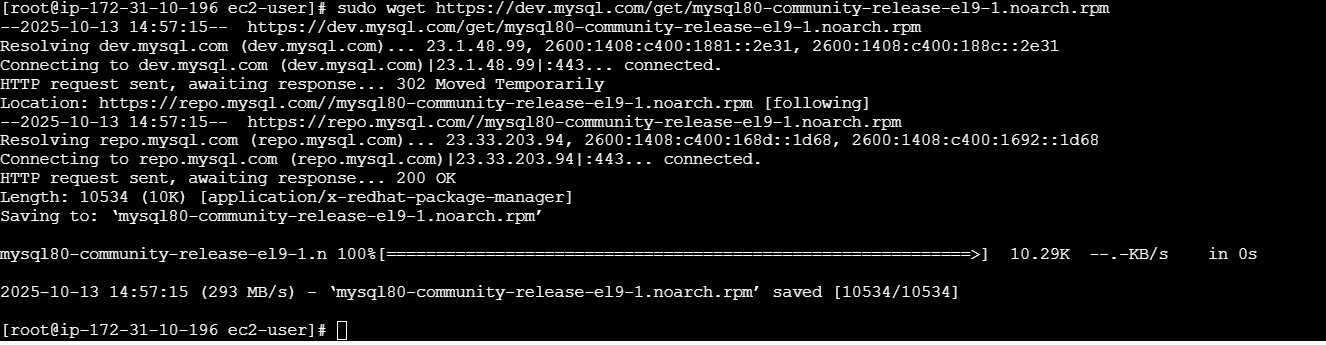


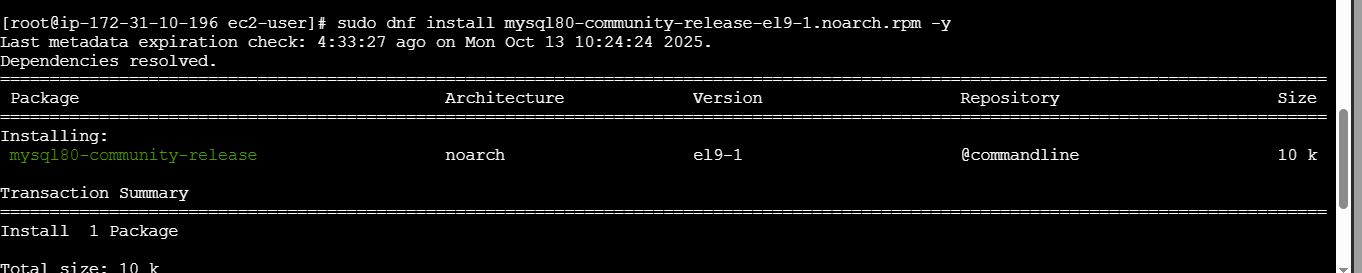
1. Install MySQL DB on EC2.

Step 1: Remove Mariadb using following commands  
sudo dnf remove -y mariadb105\*  
sudo dnf clean packages

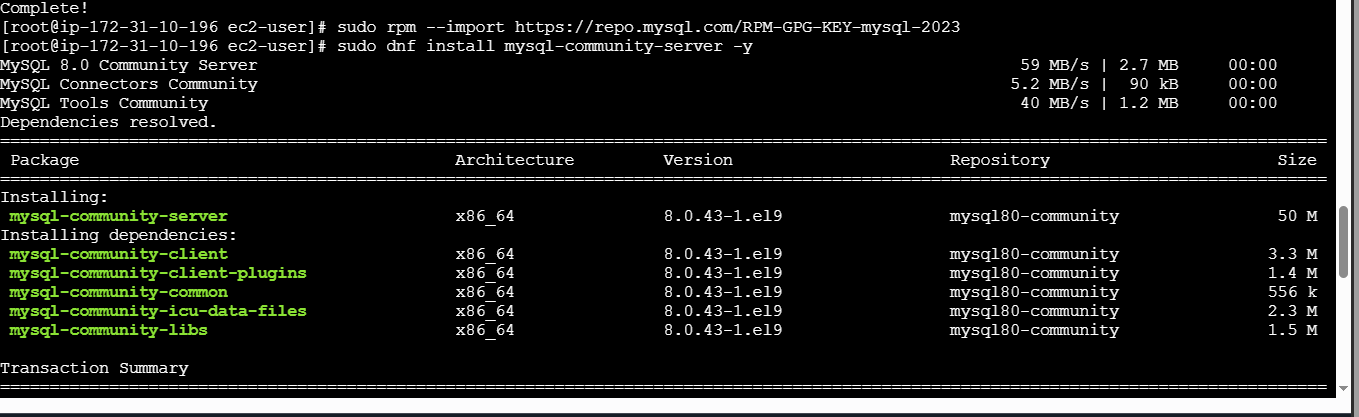


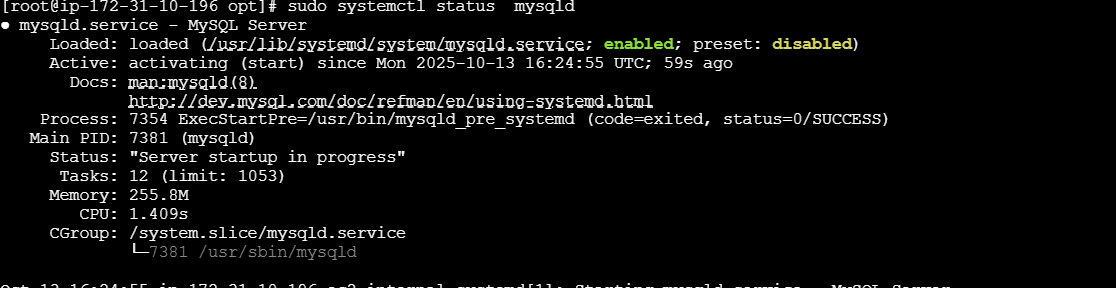






Step 2: Goto /opt directory and run the following commands to download and   
install the MYSQL.  
sudo wget https://dev.mysql.com/get/mysql80-community-release-el9-  
1.noarch.rpm  
sudo dnf install mysql80-community-release-el9-1.noarch.rpm -y  
sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2023  
sudo dnf install mysql-community-server -y  
sudo systemctl start mysqld  
sudo systemctl enable mysqld  
sudo mysql\_secure\_installation  
sudo systemctl status mysqld





1. Launch MySQL RDS image.

Step 1: Login to AWS, select Aurora and RDS>>Database>>create   
database>>standard create>>Mysql

### ****Step 2: Choose Database Creation Method****

Select **Standard create**

**Step 3: Choose Engine**

Under **Engine options**, choose:

**Engine type:** MySQL

**Version:** Latest stable (e.g., MySQL 8.0.x)

**Step 4: Templates**

Choose a deployment option:

**Free tier** (for practice or testing)

**Production** (for real workloads)

**Step 5: Settings**

**DB instance identifier:** myappdb-rds

**Master username:** admin

**Master password:** choose a strong one (e.g. StrongPassword123!)

Confirm password.

**Step 6: Instance Configuration**

Choose instance type, e.g.:

db.t3.micro (Free tier)

Leave **Storage** defaults or set as

### ****Step 7: Connectivity****

**Virtual Private Cloud (VPC):** choose your application VPC

**Subnet group:** default or custom (if you have public/private subnets)

**Public access:**

No for private-only RDS

**VPC security groups:**

Choose an SG that allows inbound port 3306 from your EC2 instance’s SG

**Step 8: Additional Configuration**

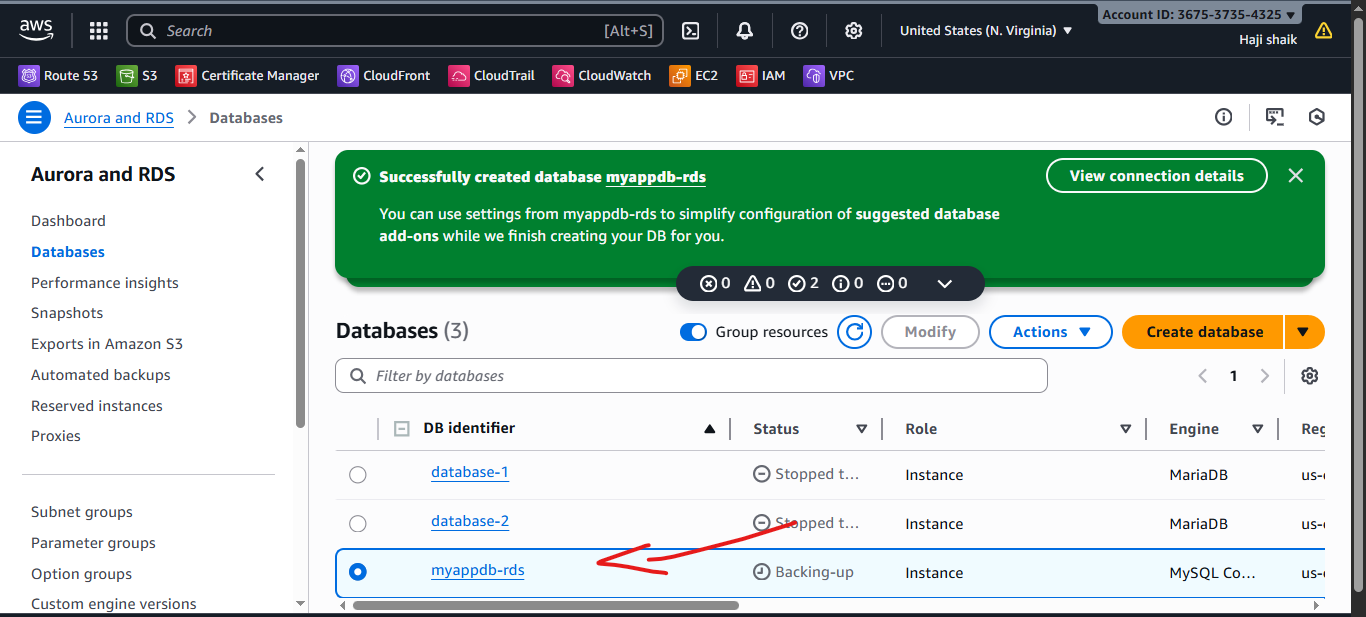
**Initial database name:** myappdb

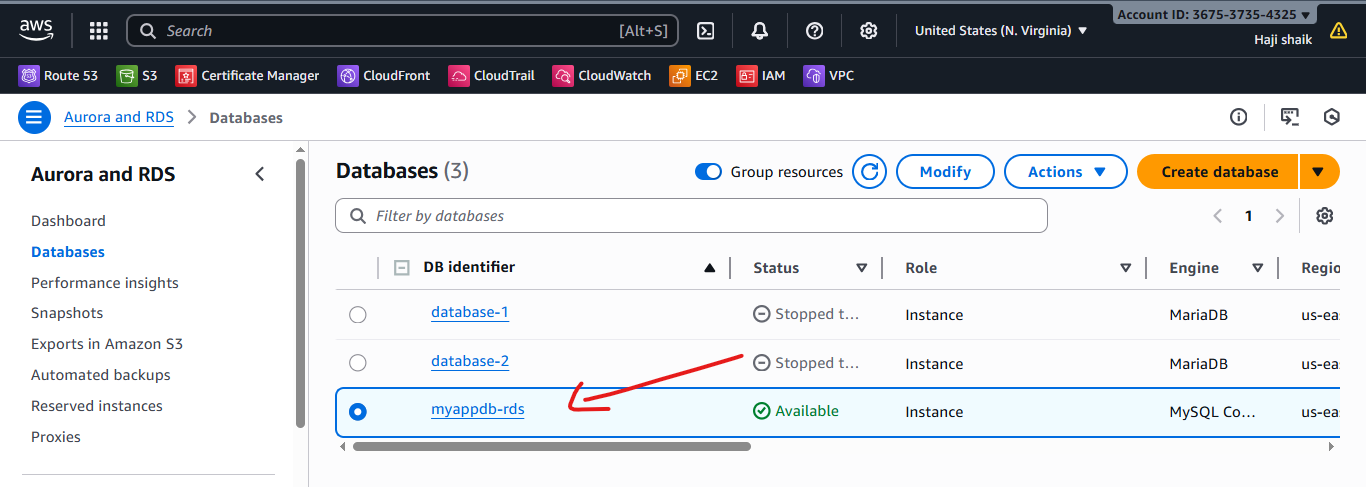
**Backup:** Enable automated backups (7 days default)

**Monitoring:** Optional (enable enhanced monitoring if needed)

**Maintenance:** Auto minor version upgrade – Yes

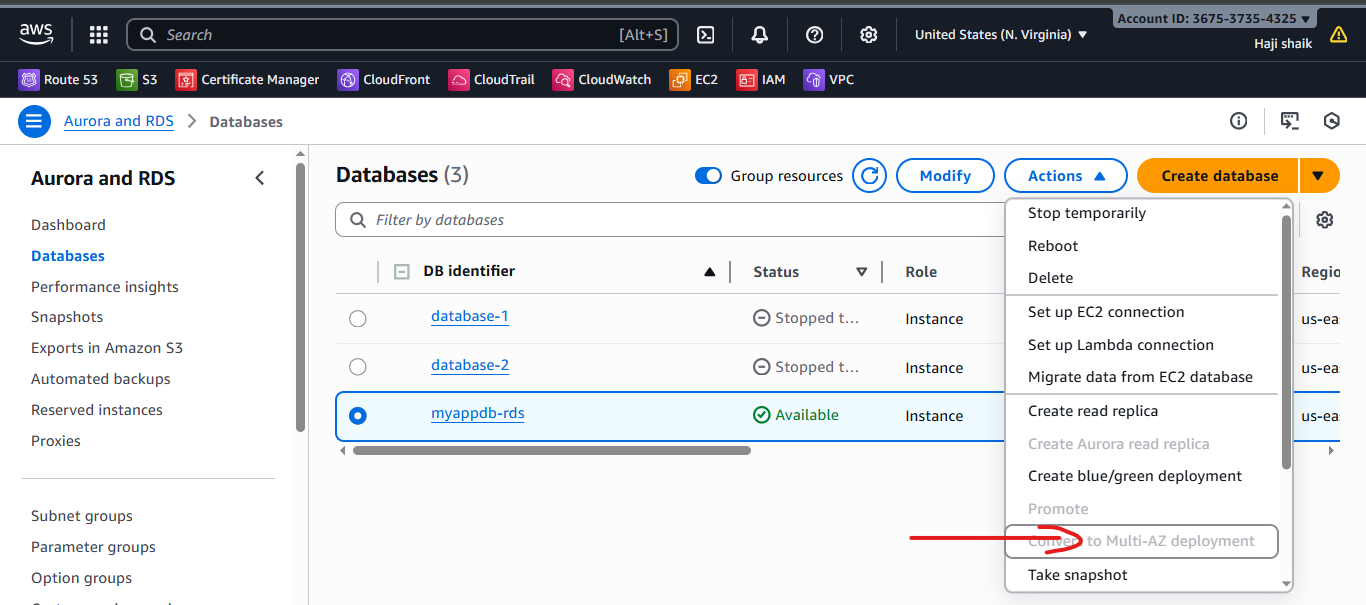
**Step 9: Create Database**





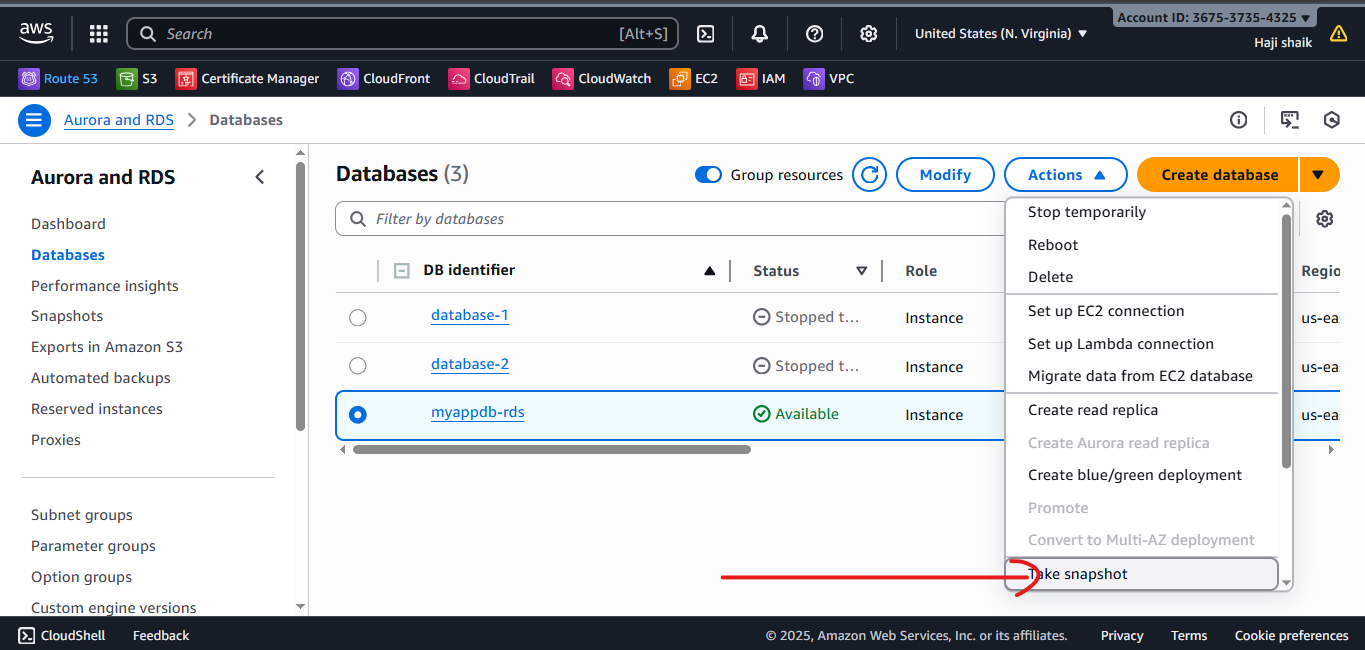
1. Configure Multi-AZ.

* GO to actions nd choose multi az option.
* But the thging is this option is use full for only free account user.

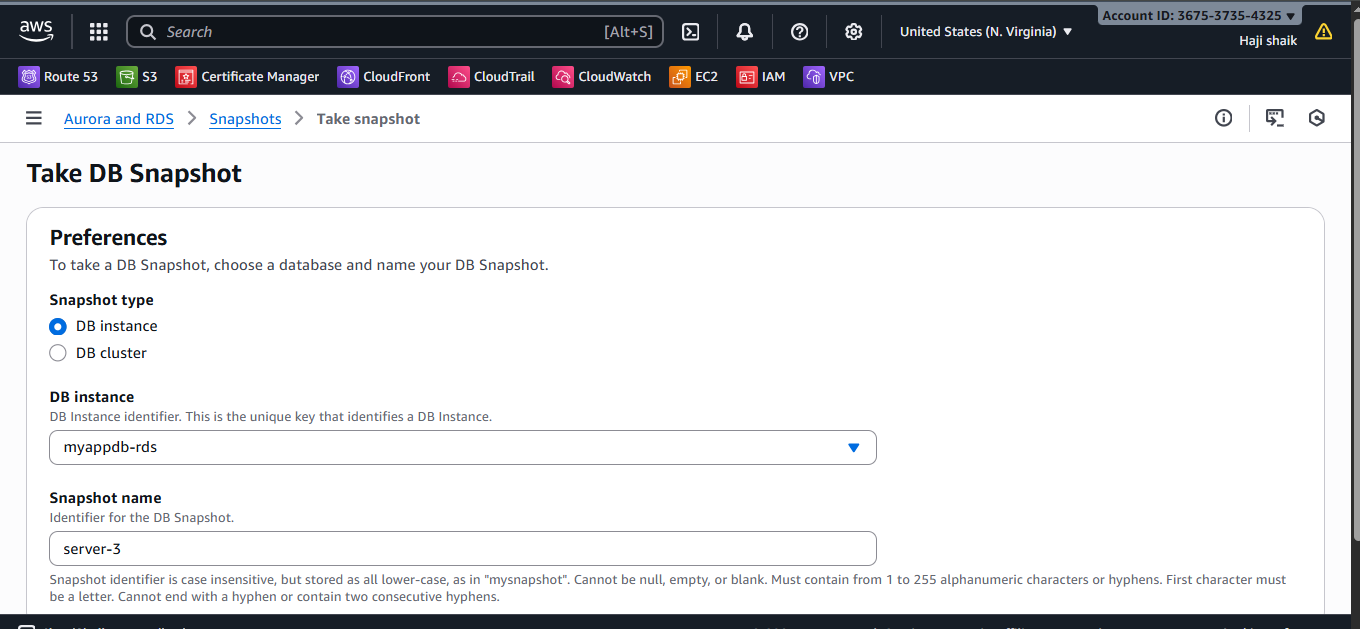


1. Take backup of DB and restore the DB.

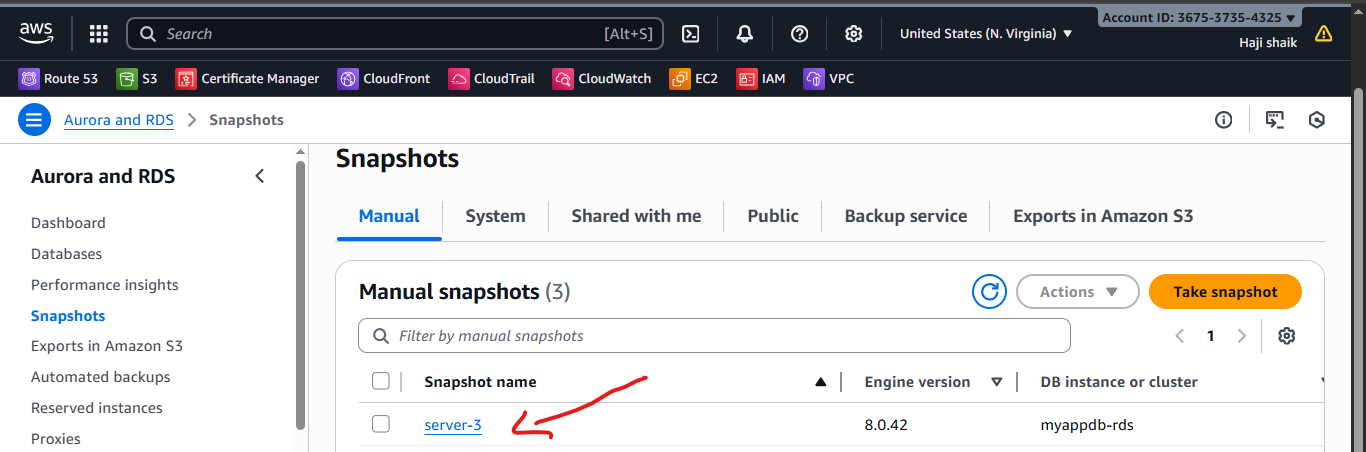
* Choose the data base which backup uh want.
* Then go the actions.
* Then use snapshot.



* Then create snapshot.



* Then gave a proper name for the snapshot.
* Then delete the data base.
* Use snapshot to recover that data base.



1. Create read replica.

* Use the data base and go the actions.
* Nd use the option of
* Create read replica.

