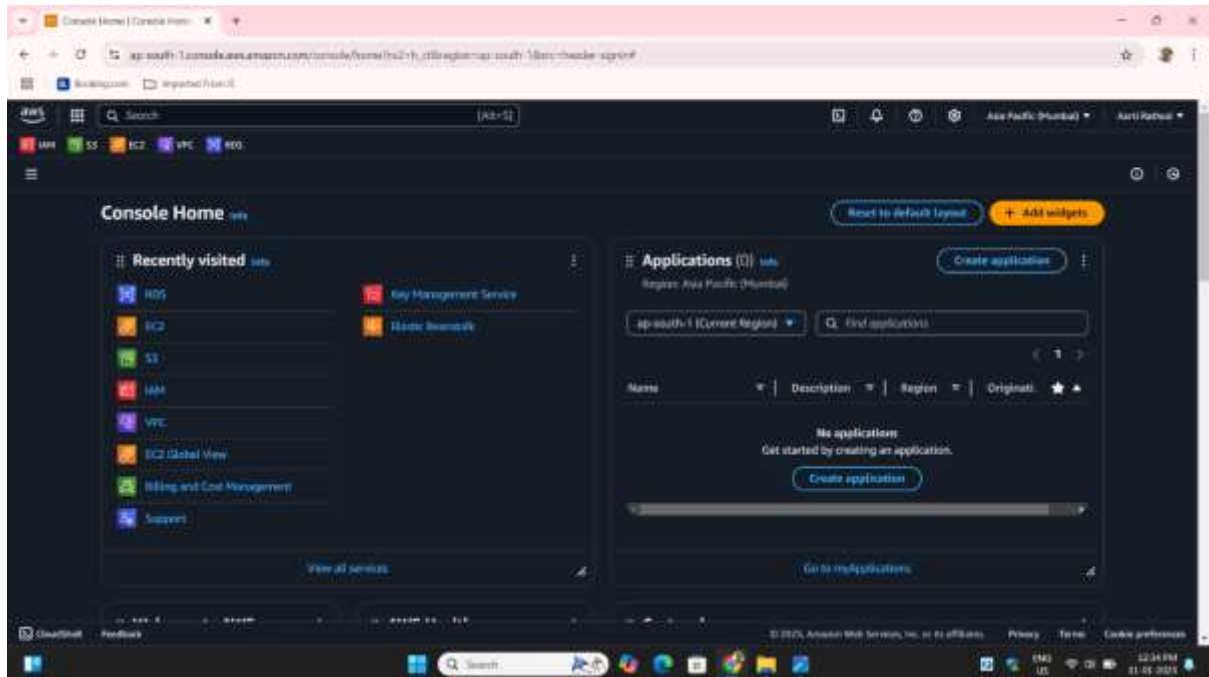
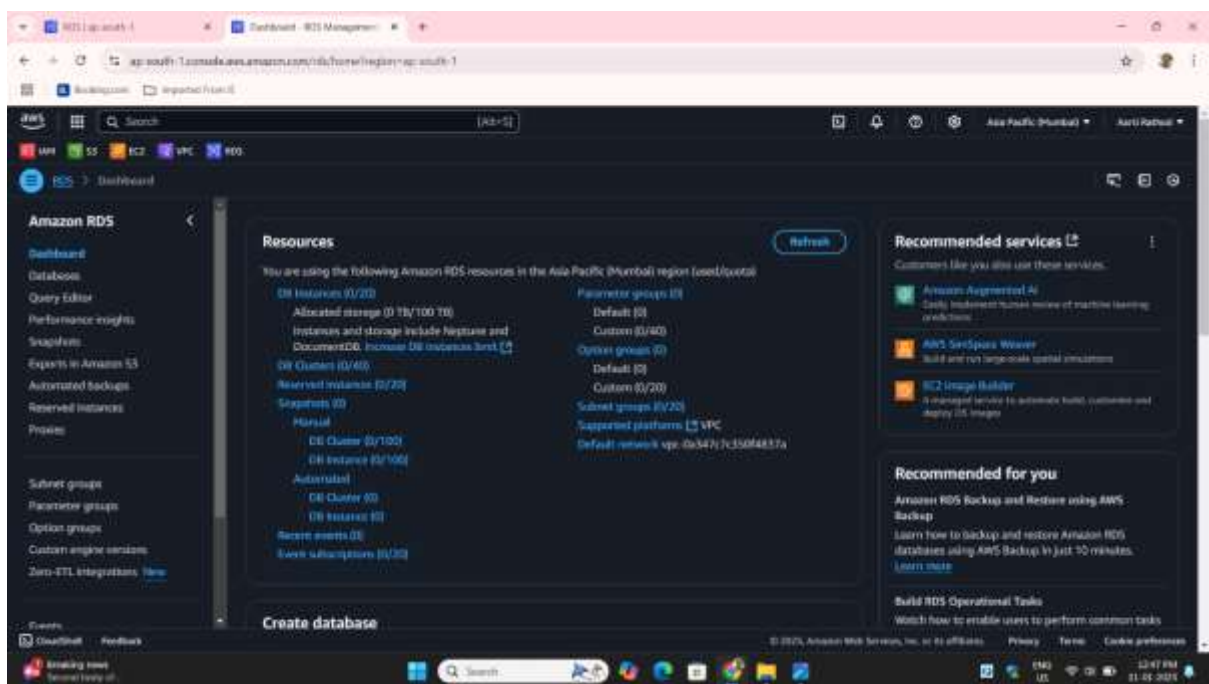


Deploying and Managing MariaDB on AWS using EC2 and RDS

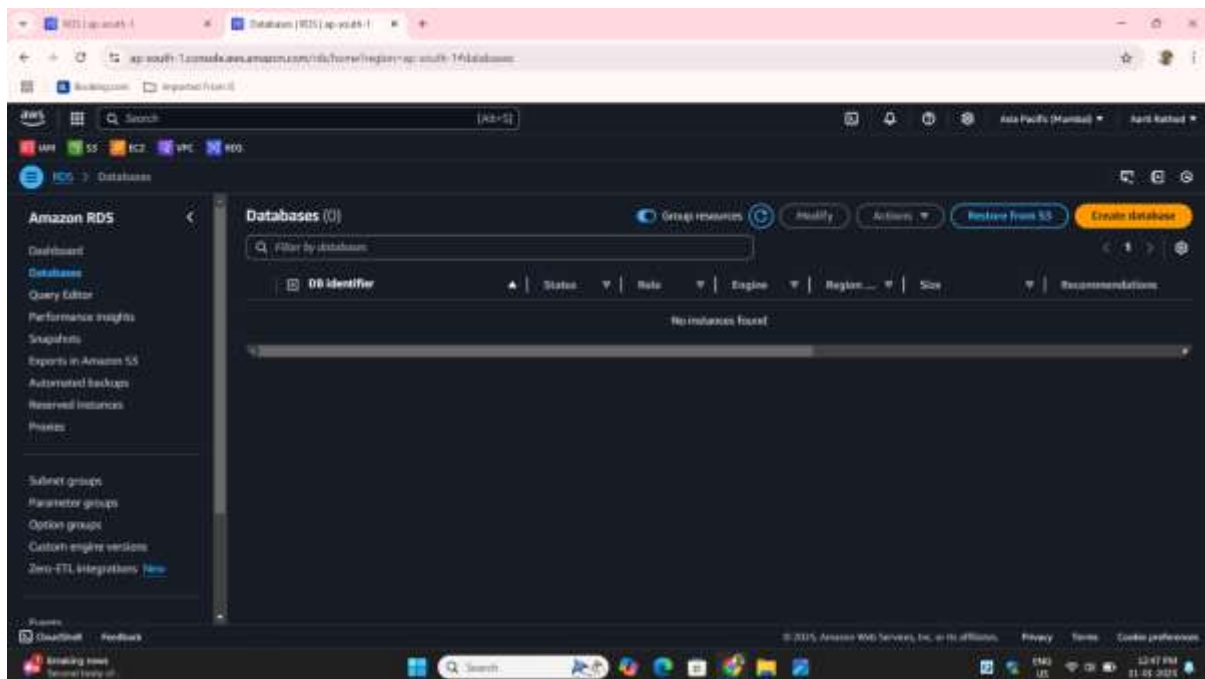
Step 1 : Log in to the AWS Management Console



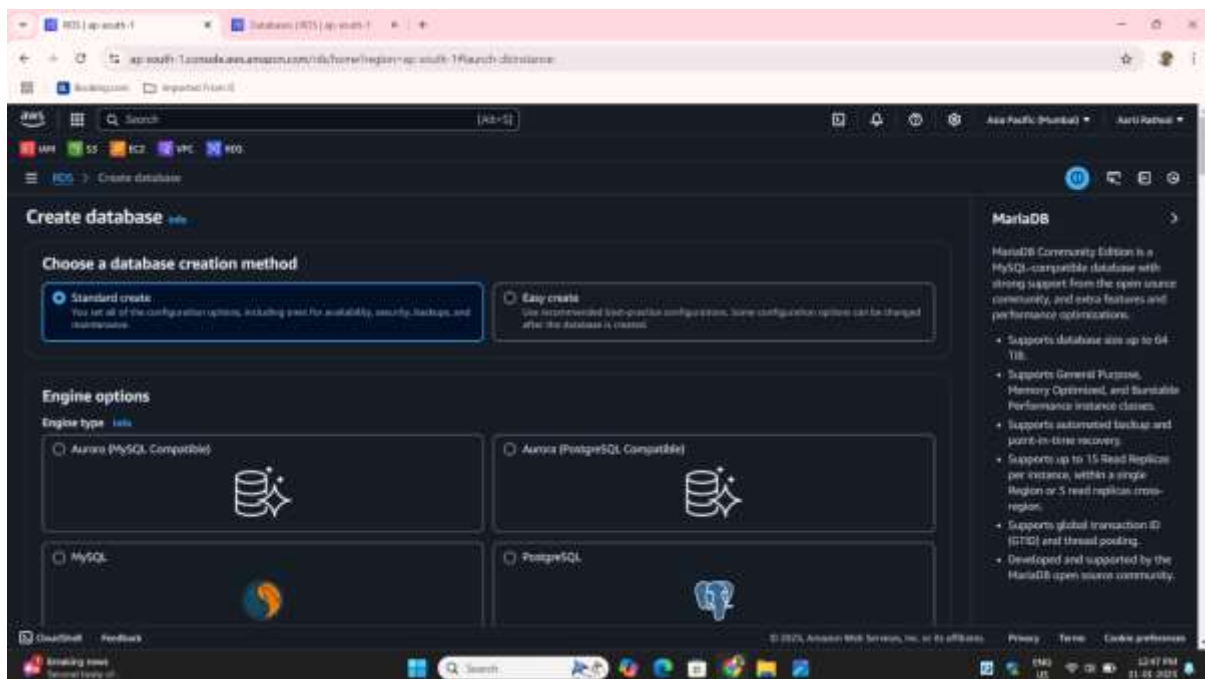
Step 2 : Navigate to the RDS service.



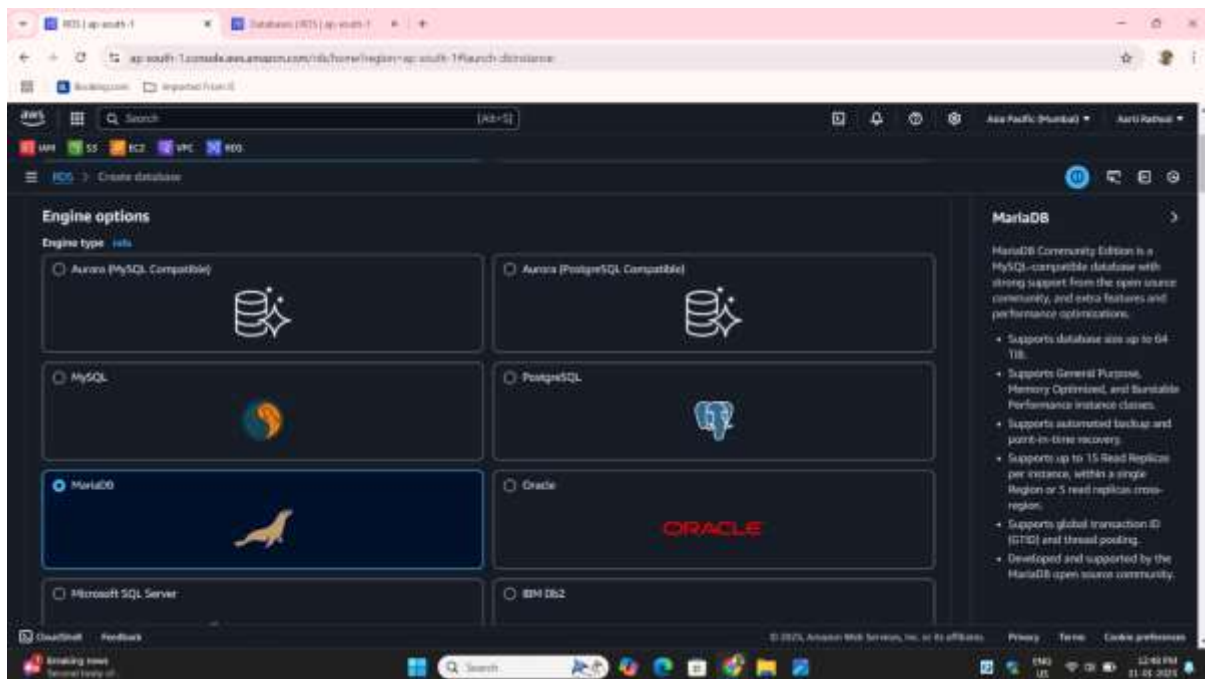
Step 3 : Click Create database.



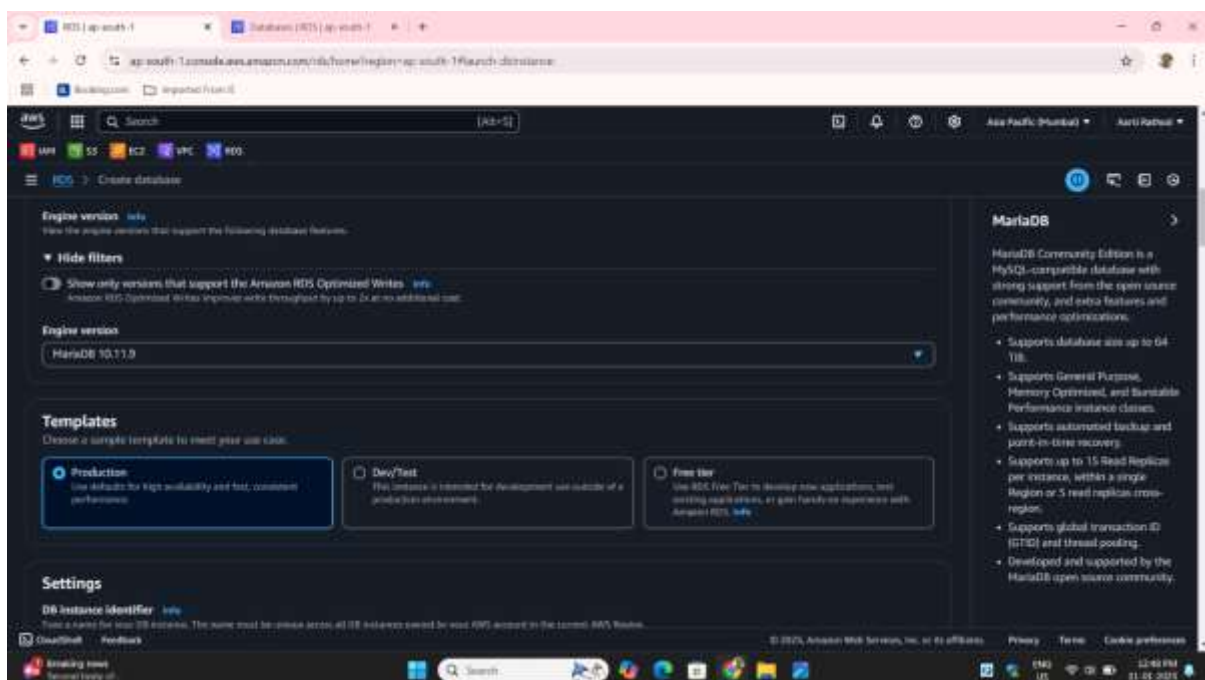
Select Standard create for more control over settings.



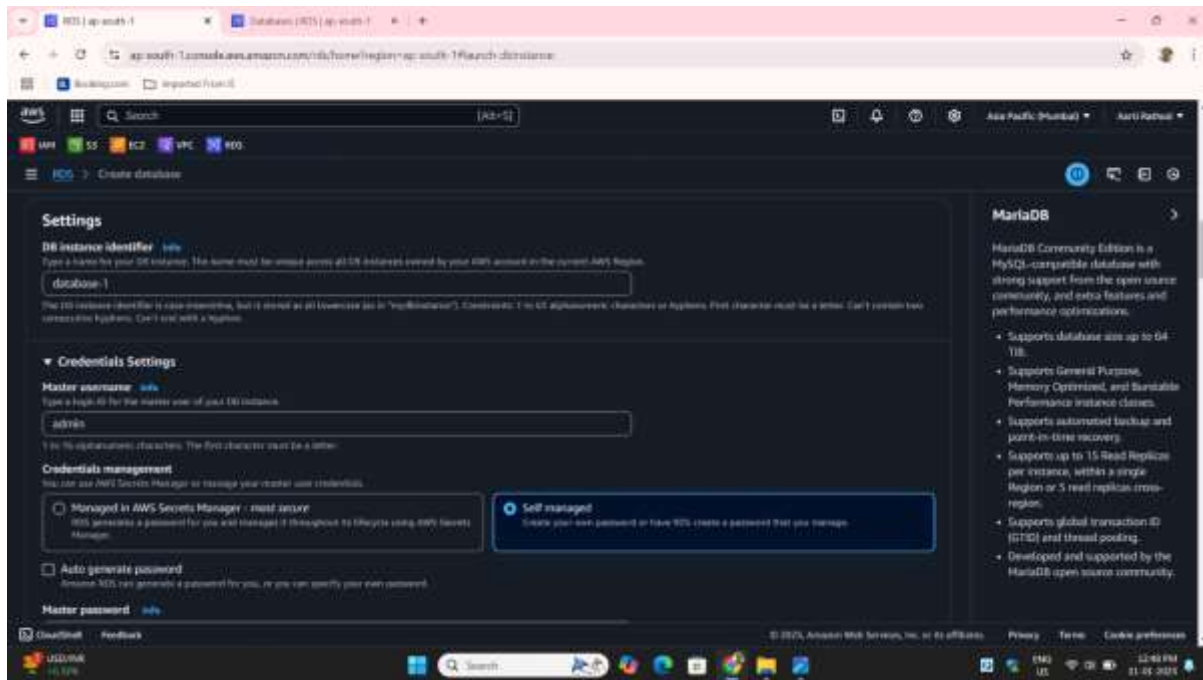
choose your database engine MariaDB



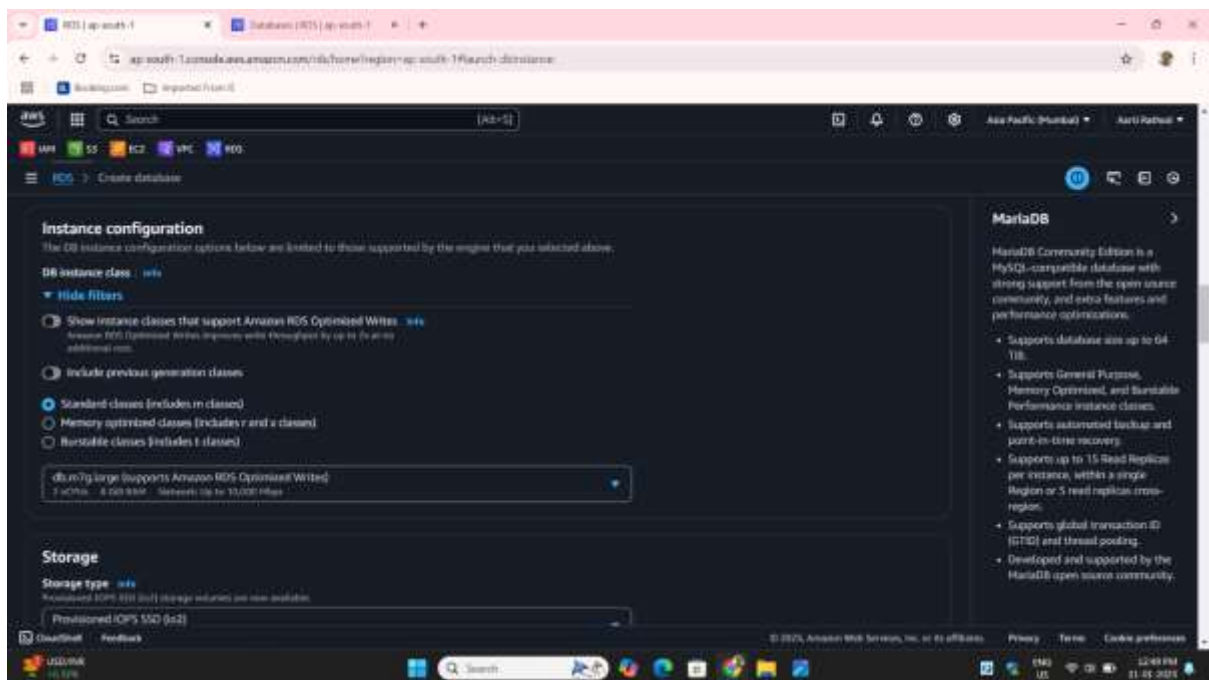
Choose Engine Version: Select the required version. Choose Templates: Choose Production



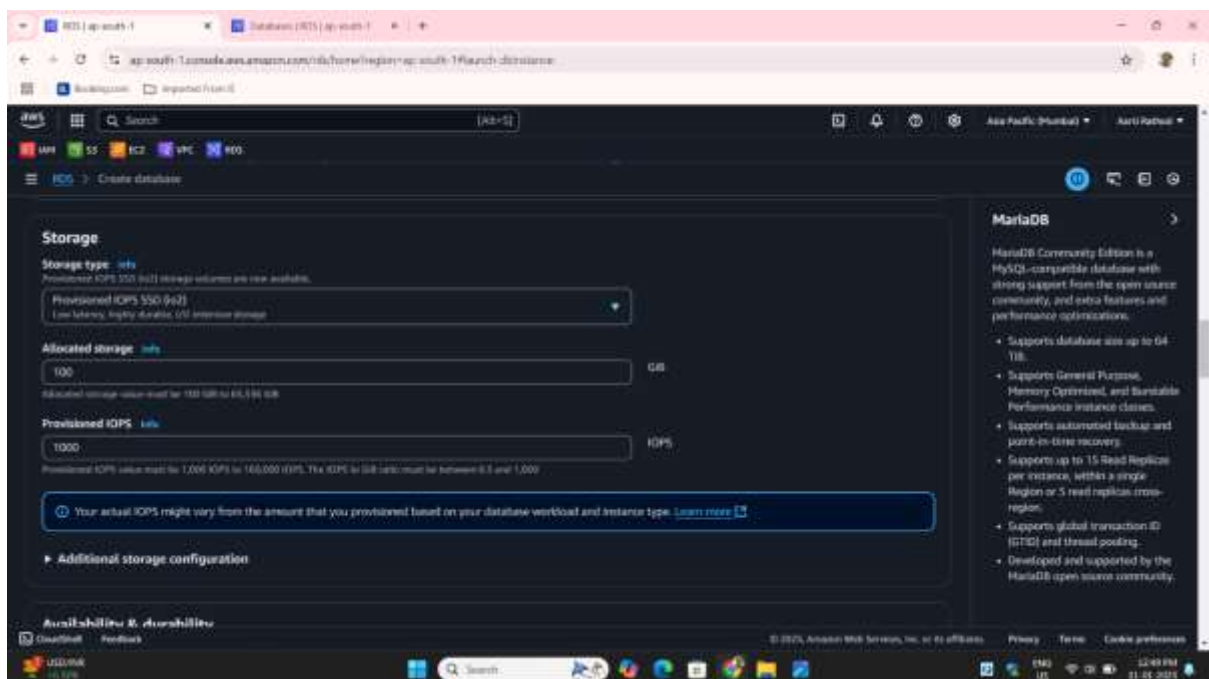
1. **DB Instance Identifier:** Enter a unique name for your database (e.g., mydb-instance).
2. **Master Username:** Set the admin username for managing the database.
3. **Credentials Management:** choose self manage
 - Select Auto-generate a password if you want AWS to create a strong password.
 - Alternatively, enter your own password manually.



Choose standard classes for cpu and memory for everyday use

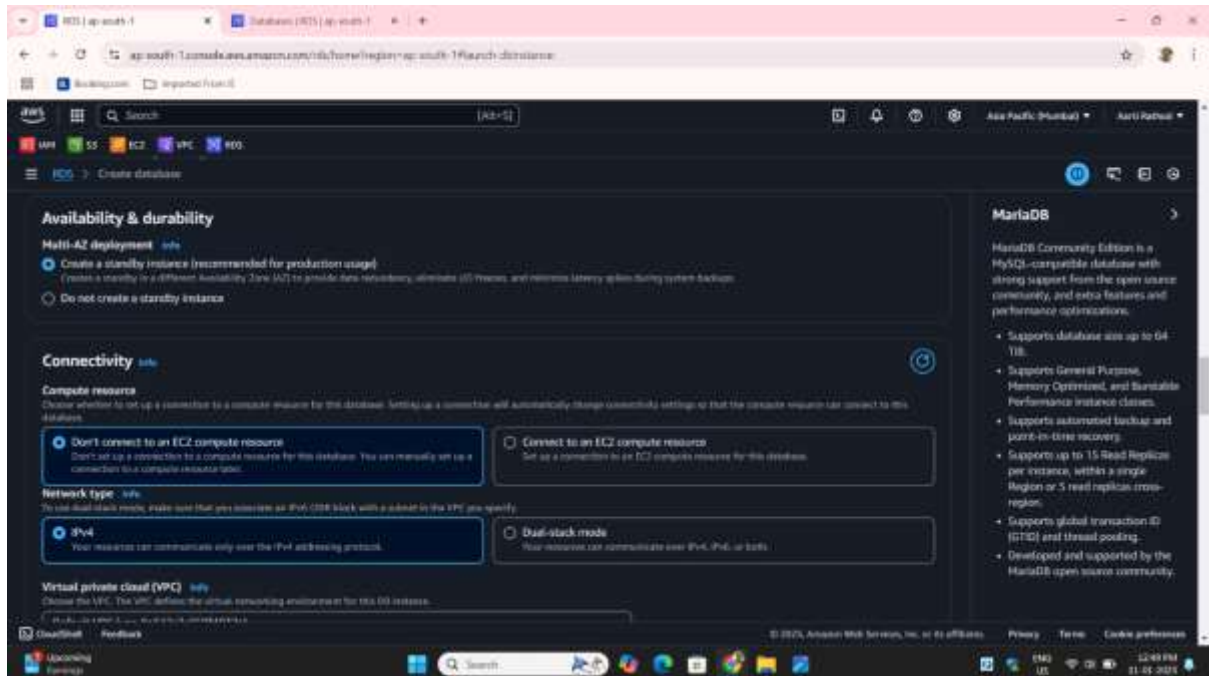


Set the storage type (e.g., General Purpose SSD). Allocate sufficient storage space (minimum 20GB for production). Enable storage autoscaling if required.

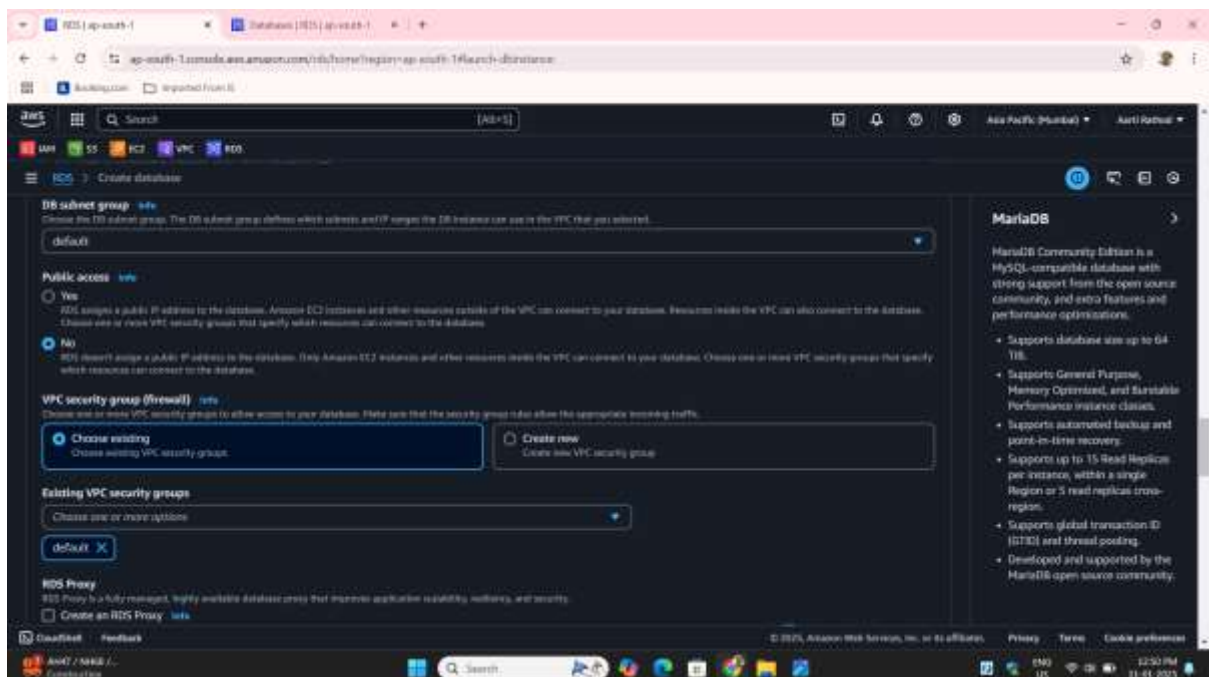


Enable Multi-AZ Deployment for high availability. Create a standby instance for a backup to keep your database available if it fails.

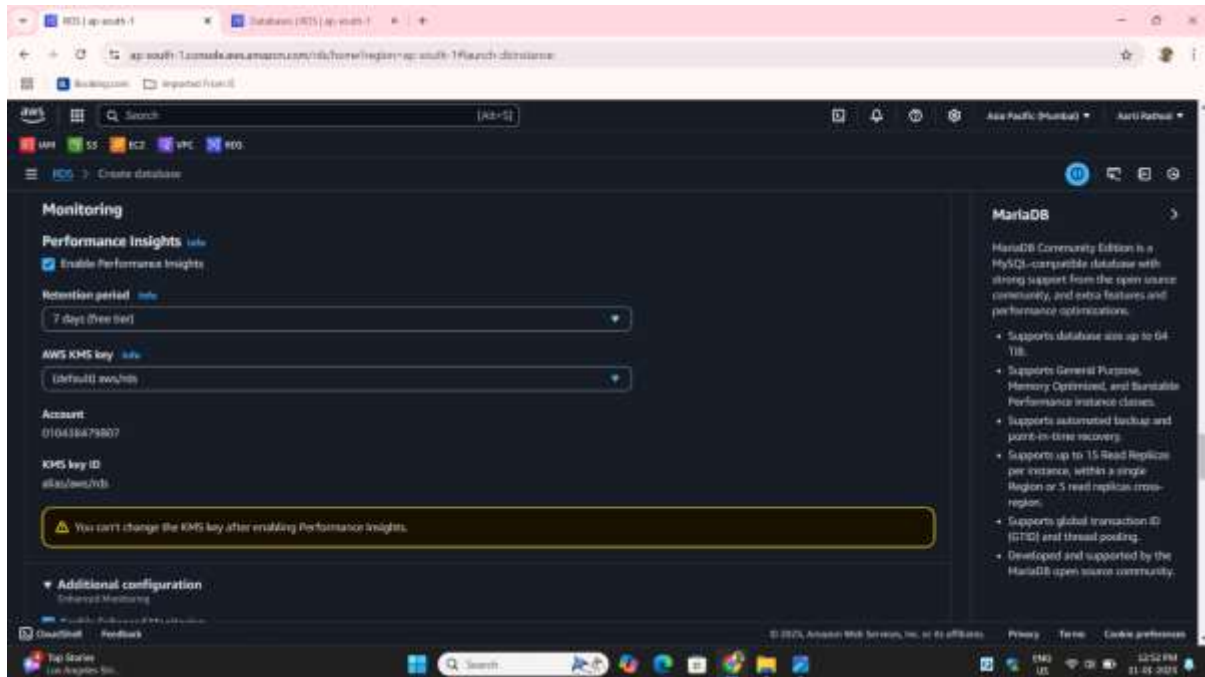
Choose Don't connect EC2 instance compute resource this we will do manually. Choose network type IPV4



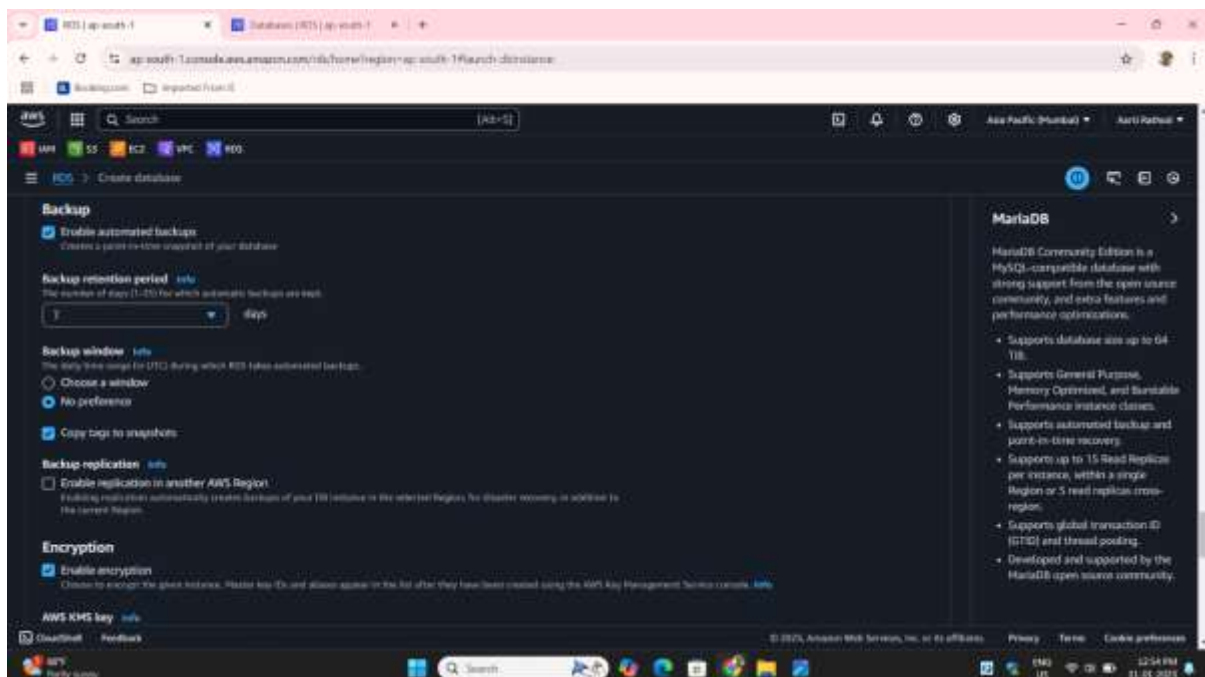
VPC: Select the VPC for the database. Public Access: Enable if the database needs external access.



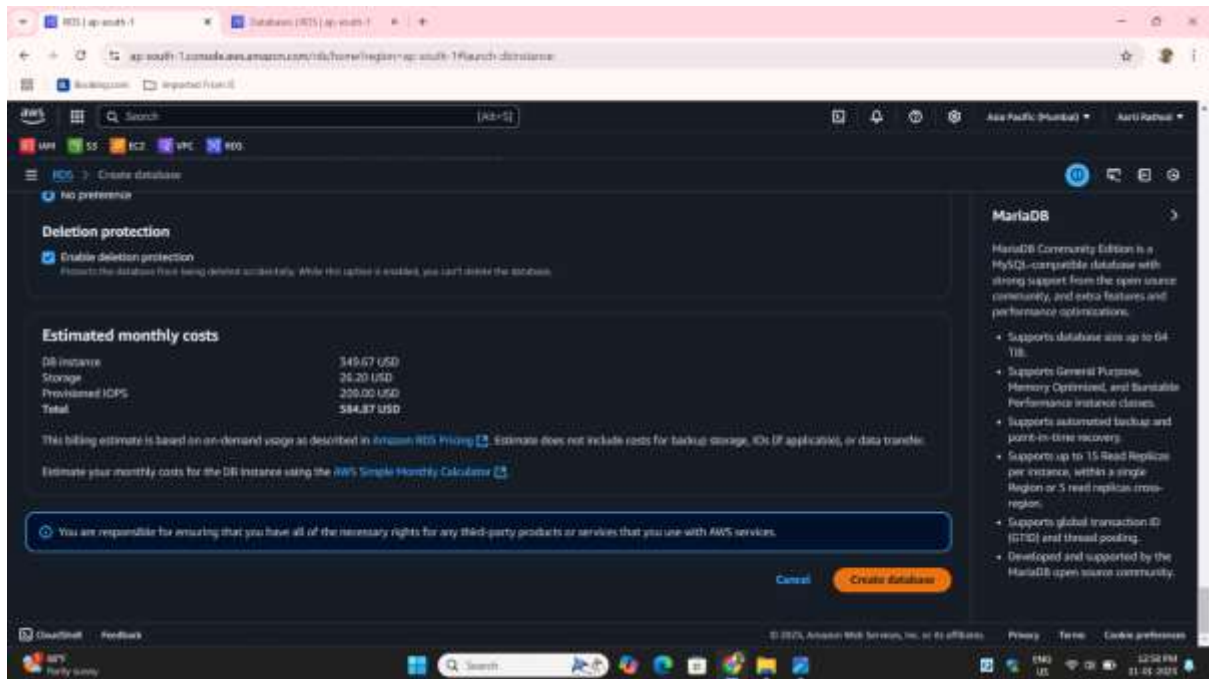
Enable performance insights for track and analyze database performance



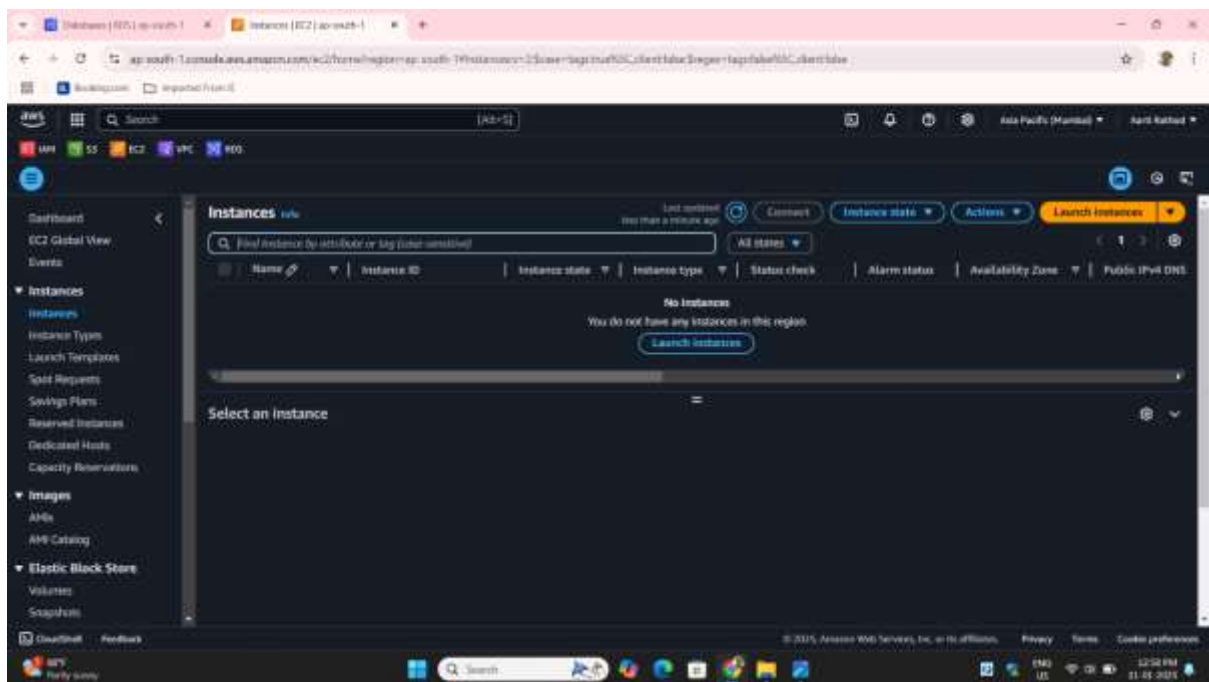
It refers to automatically saving copies of your database to protect against data loss.



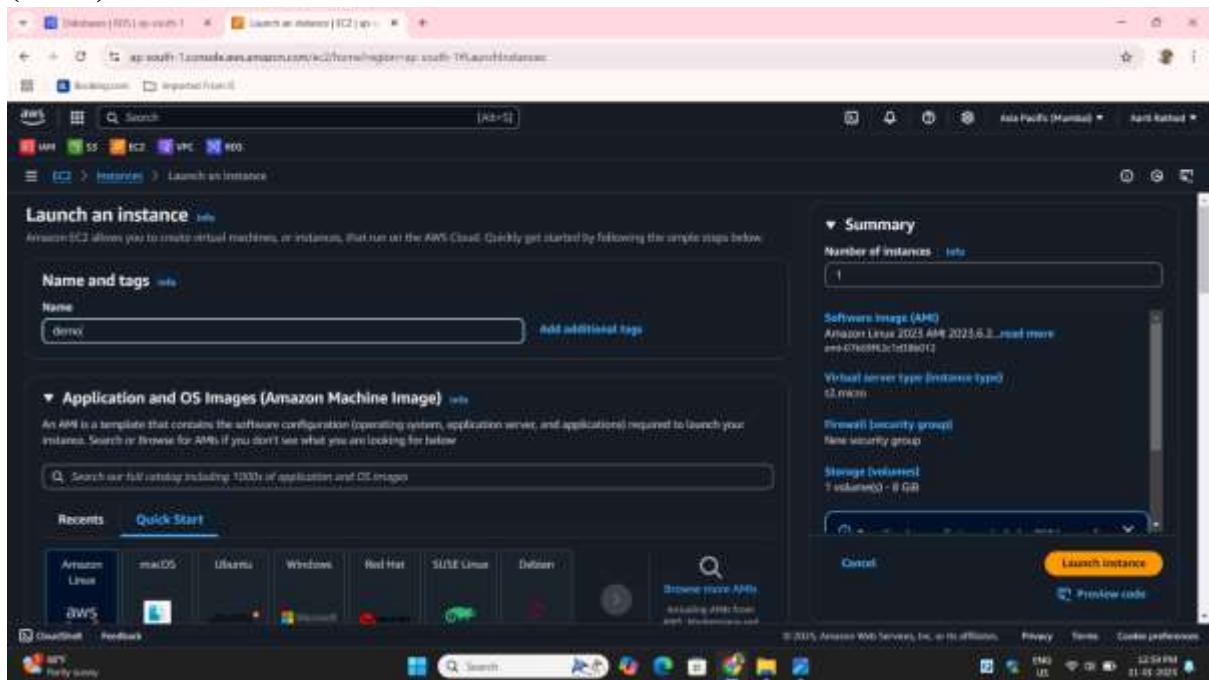
Delete Protection in RDS prevents accidental deletion of your database instance. Click on Create database



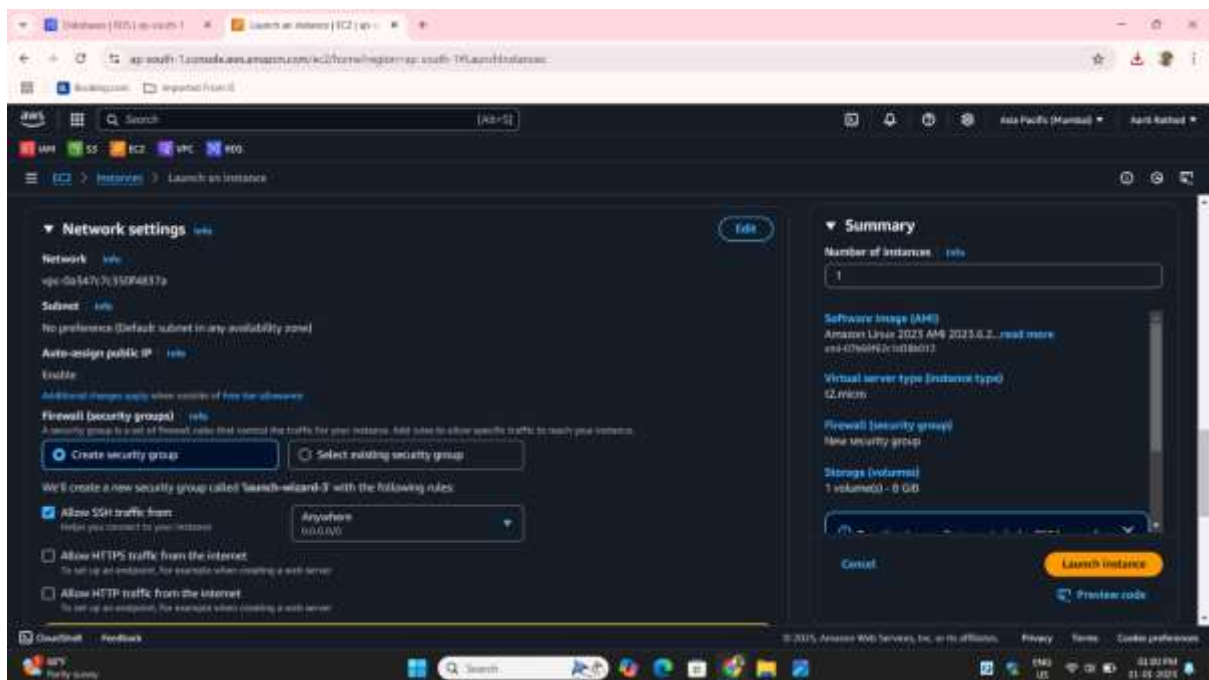
Log in to AWS Management Console: Go to EC2 Dashboard. Launch Instance: Click on Launch Instance to start the creation process.



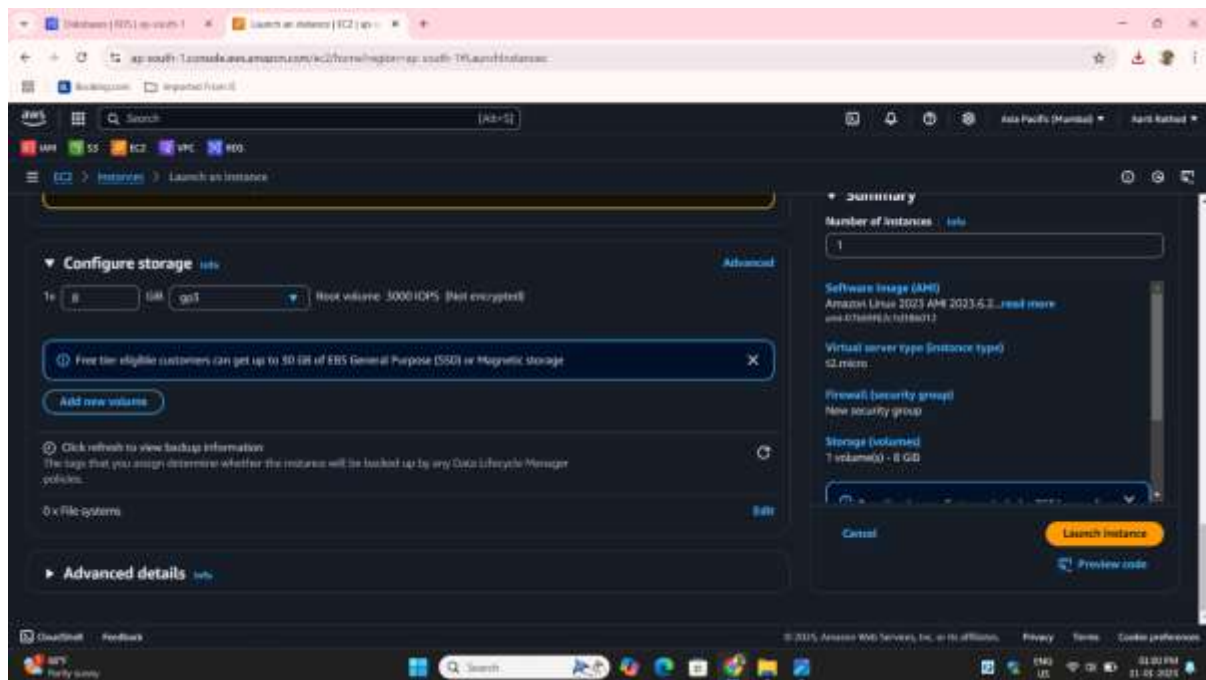
Give a name Demo to your Instance Choose an Amazon Machine Image (AMI): Amazon Machine



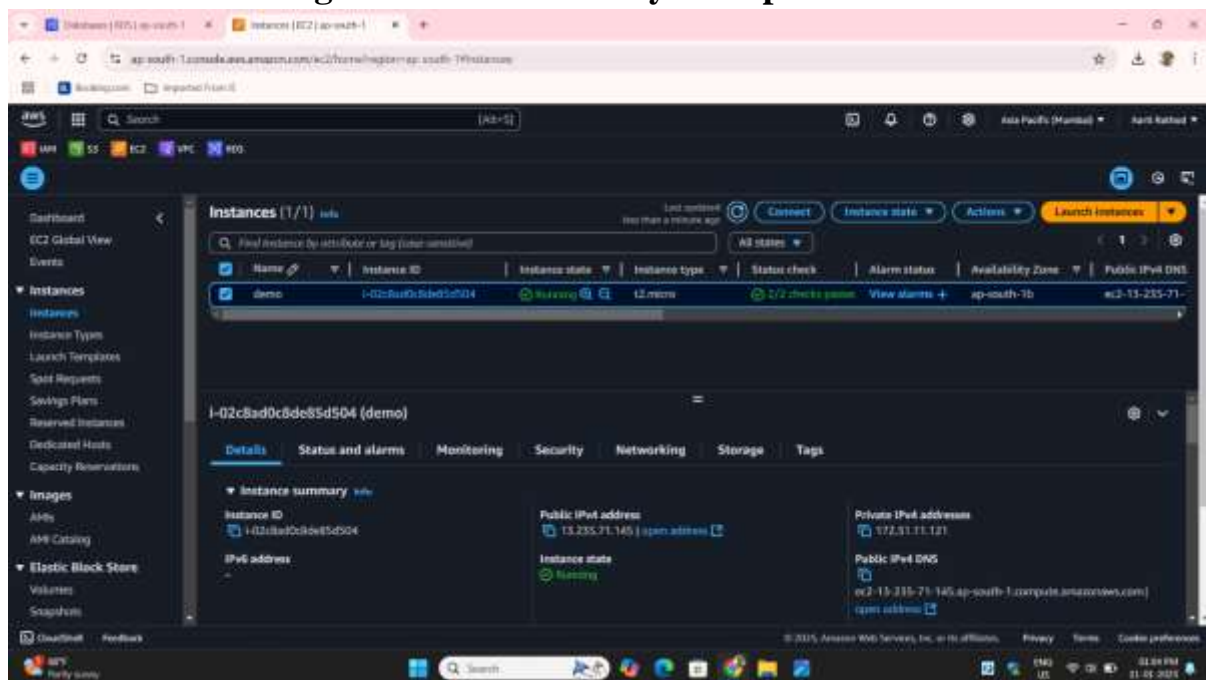
Create key pair



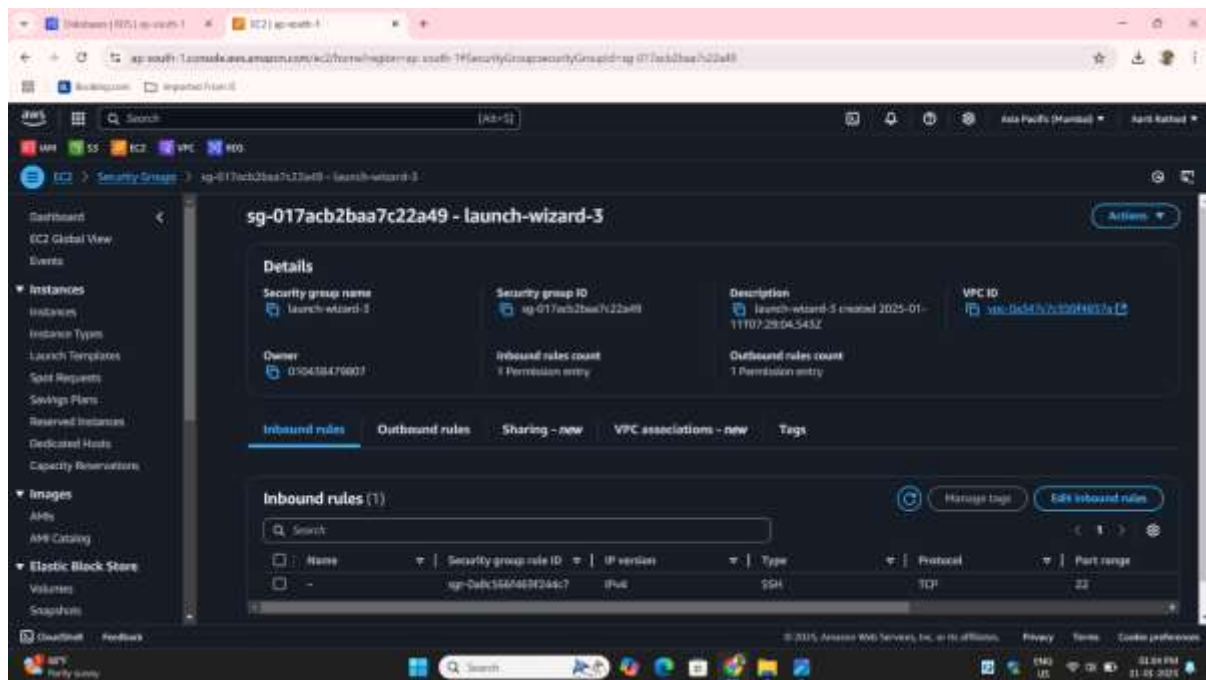
Click on Launch Instance



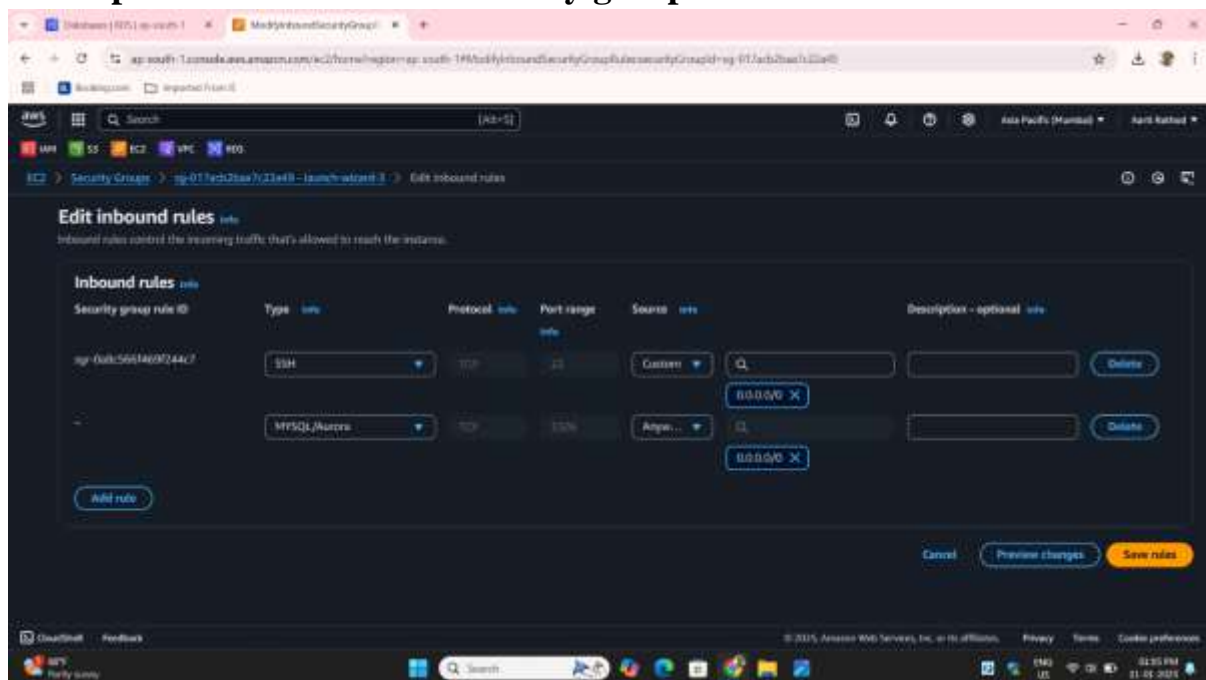
Instance is created go to instance security Group



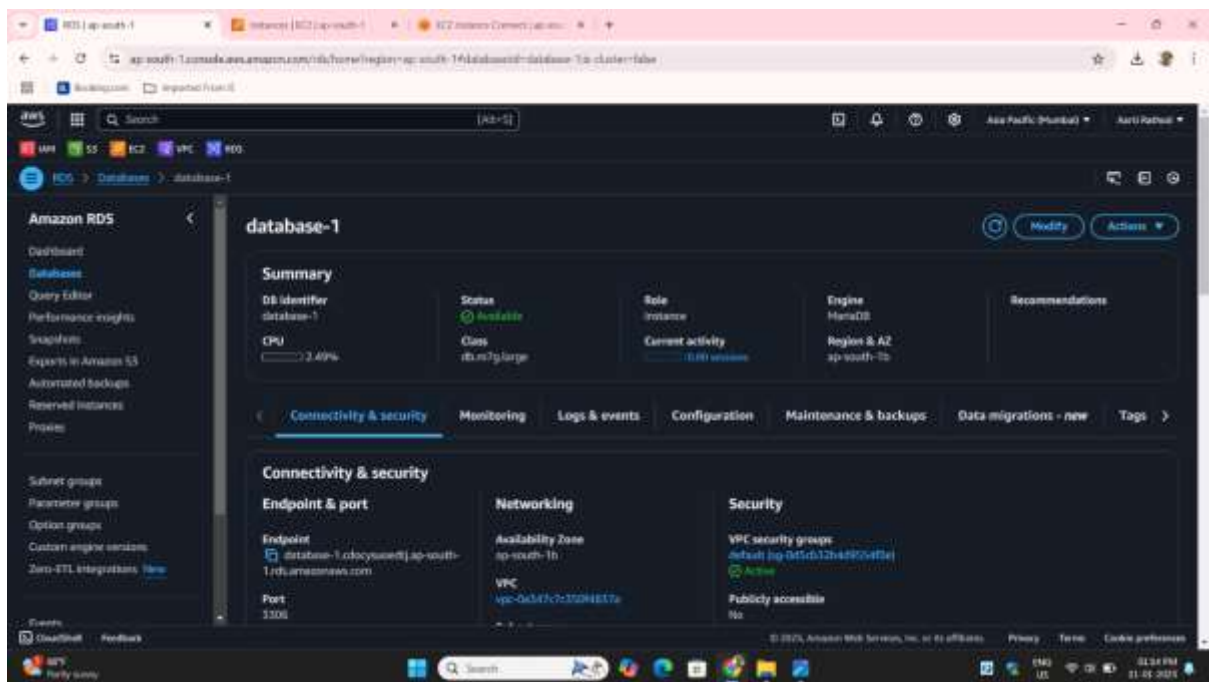
Click on Edit Inbound Rules



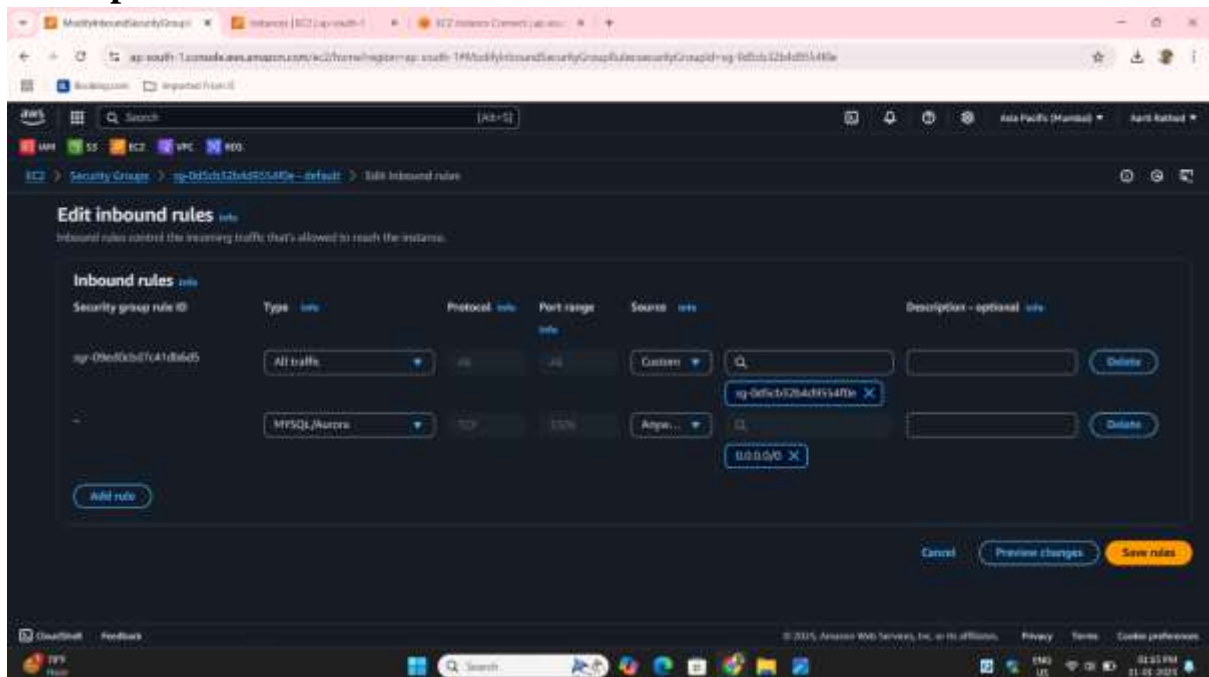
Allow port 3306 in Instance security group



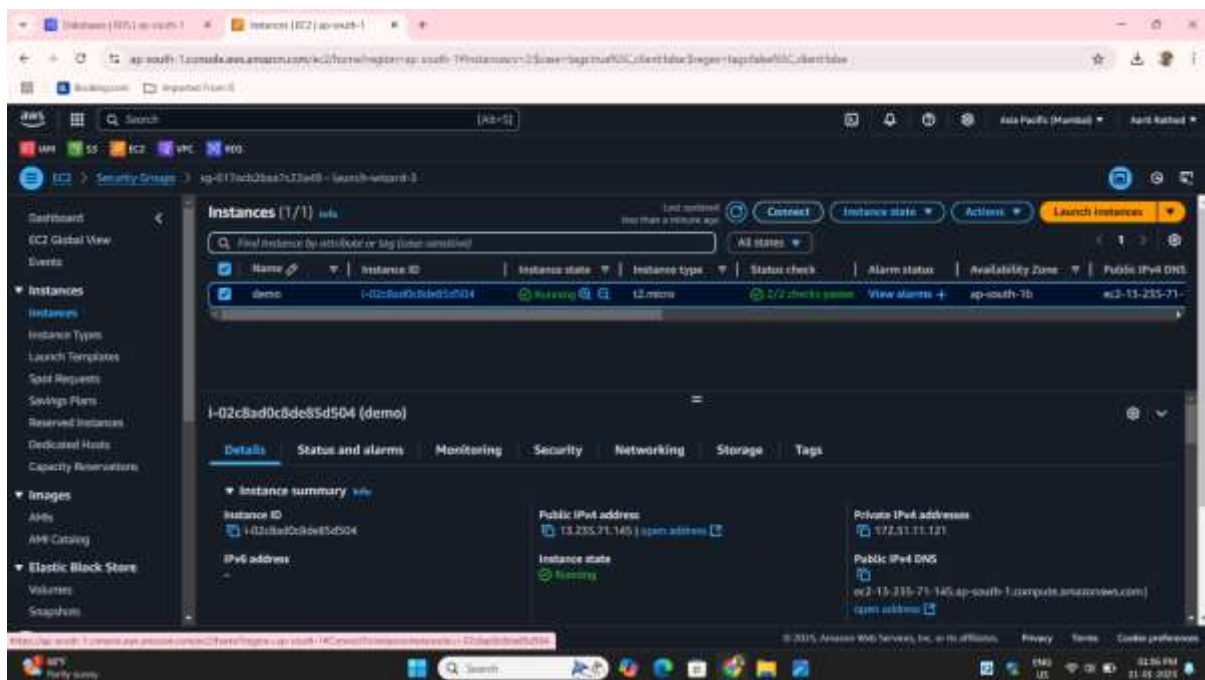
Goto Rds security group



Allow port 3306

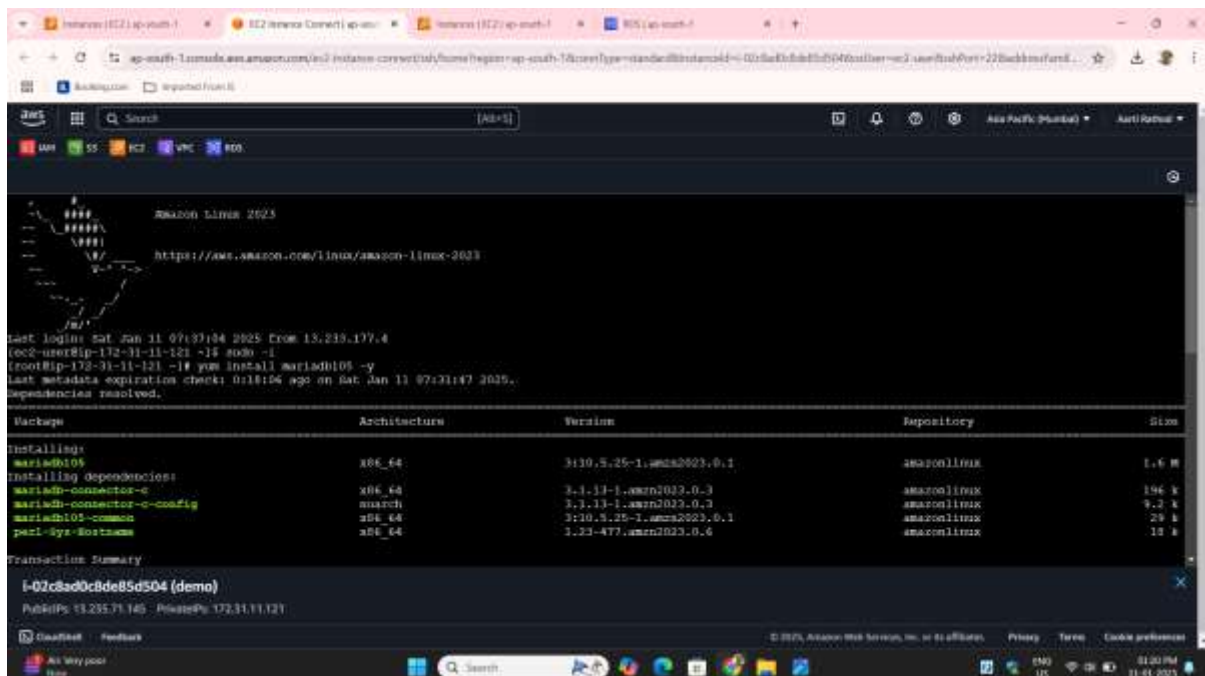


Come to ec2 instance select instance click on connect

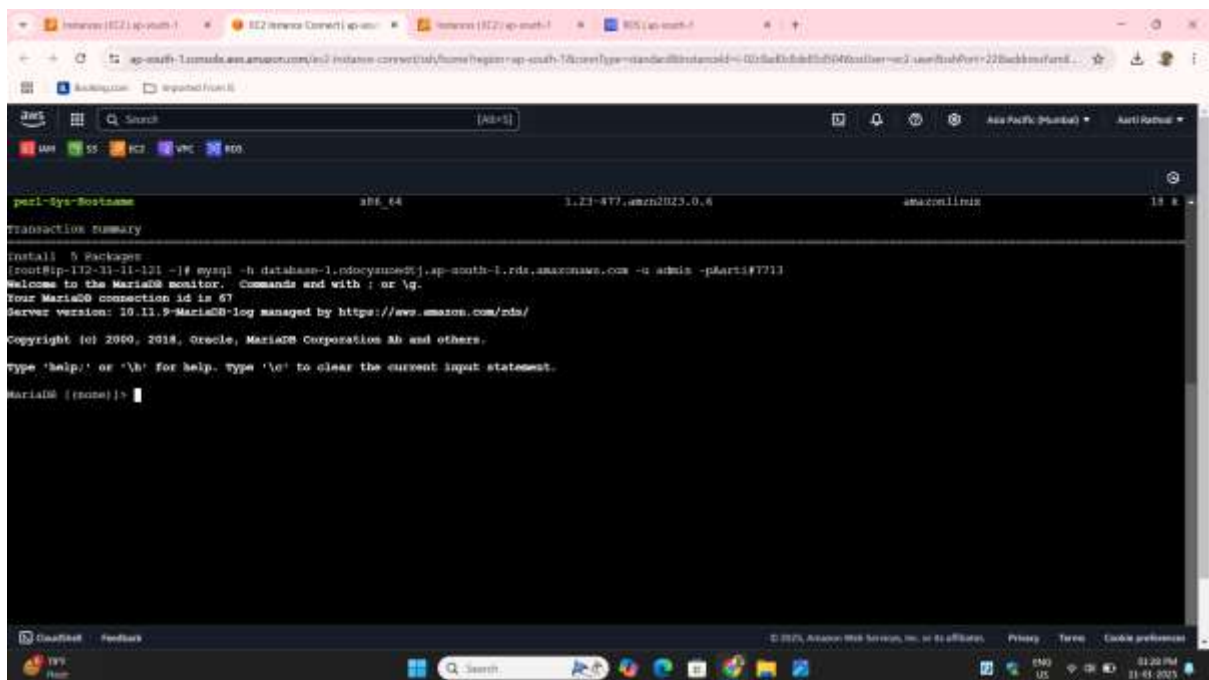


Run Commands on CLI Now you're login to CLI switch to root user

Install Mariadb: install mariadb using package manager



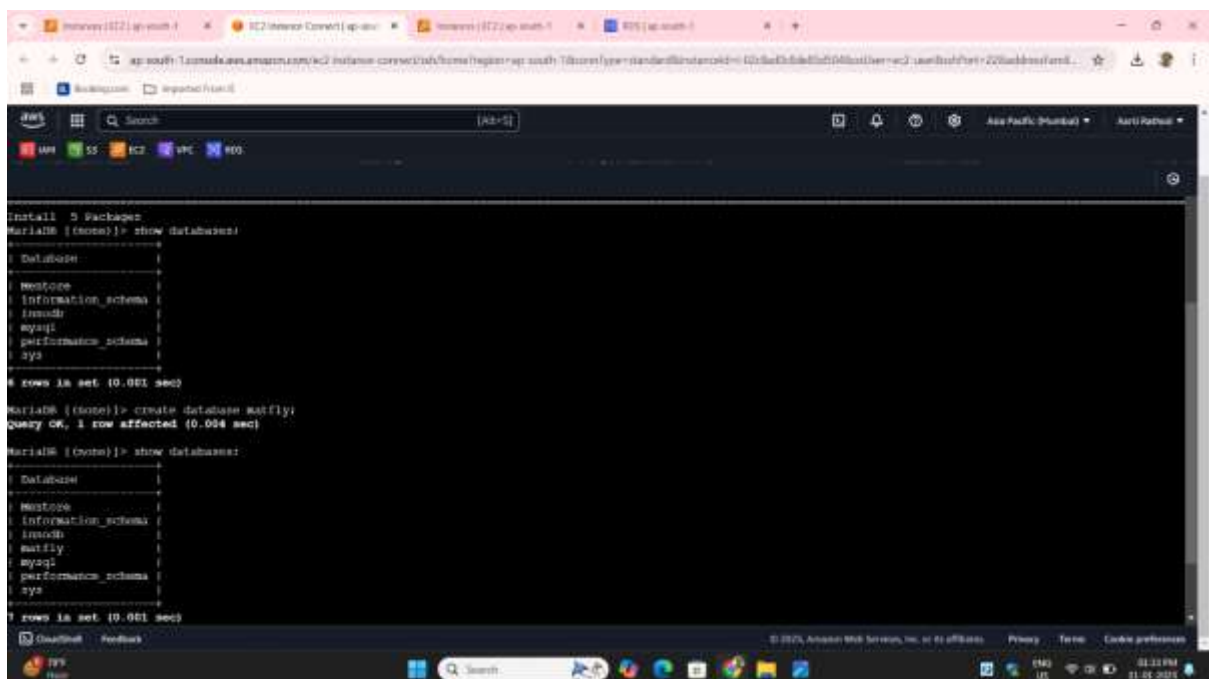
Connect to mariaDB With Rds Endpoint



The screenshot shows a terminal window with a dark background. At the top, there's a header bar with the AWS logo and some navigation icons. Below that, the terminal displays the output of a command to connect to a MariaDB instance. The output includes a transaction summary, the command used to connect, and a welcome message from the MariaDB monitor. The user is now at the MariaDB prompt.

```
perl-Sys-Hostname          sff_64          1.23-477.amzn2025.0.6          amazonlinux          1.5
Transaction summary
[install] 5 Packages
[root@ip-172-31-11-121 ~]# mysql -h database-1.rds.amazonaws.com -u admin -p123456
Welcome to the MariaDB monitor.  Commands and with ; or \g.
Your MariaDB connection id is 67
Server version: 10.11.9-MariaDB-log managed by https://aws.amazon.com/rds/
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]>
```

**Run Some Command For Database Create And Access Show databases-
check databases Create one database give name matfly**



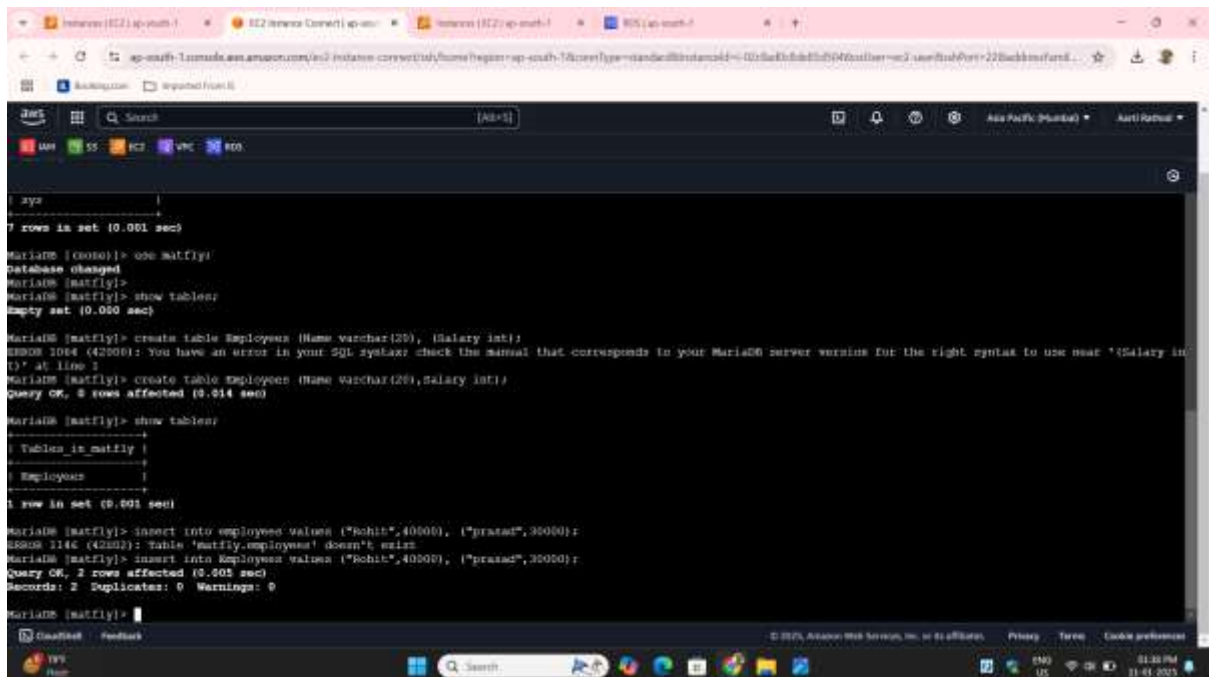
The screenshot shows a terminal window with a dark background. The terminal displays the output of several commands. First, the user runs 'show databases;' which lists the existing databases. Then, the user runs 'create database matfly;' which creates a new database. Finally, the user runs 'show databases;' again, which now includes 'matfly' in the list of databases.

```
[install] 5 Packages
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| test     |
| information_schema |
| mysql   |
| performance_schema |
| sys     |
+-----+
4 rows in set (0.001 sec)

MariaDB [(none)]> create database matfly;
Query OK, 1 row affected (0.004 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| test     |
| information_schema |
| mysql   |
| matfly   |
| performance_schema |
| sys     |
+-----+
5 rows in set (0.001 sec)
```

Use matfly command to write inside database Create table employee Insert into employees



```
mysql> use matfly;
Database changed
mysql> show tables;
Empty set (0.000 sec)

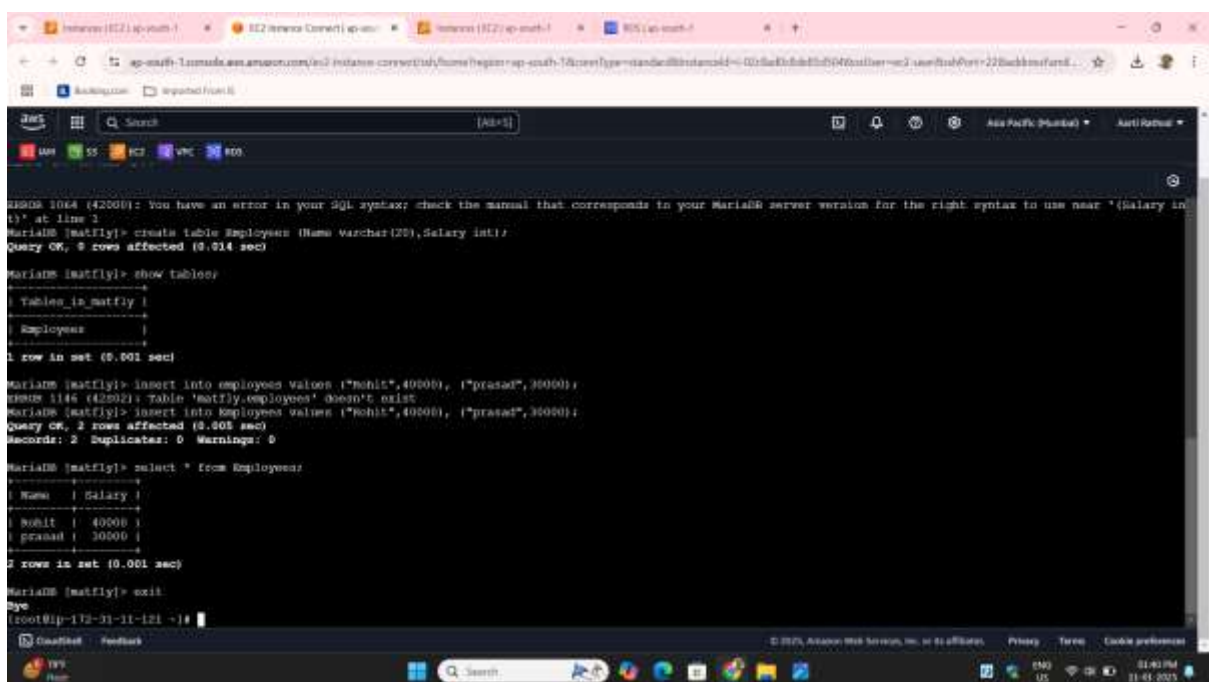
mysql> create table Employees (Name varchar(20), Salary int);
ERROR 1146 (42000): Table 'matfly.employees' doesn't exist
mysql> create table Employees (Name varchar(20),Salary int);
Query OK, 0 rows affected (0.014 sec)

mysql> show tables;
+-----+
| Tables_in_matfly |
+-----+
| Employees         |
+-----+
1 row in set (0.001 sec)

mysql> insert into employees values ("Rohit",40000), ("prasad",30000);
ERROR 1146 (42000): Table 'matfly.employees' doesn't exist
mysql> insert into Employees values ("Rohit",40000), ("prasad",30000);
Query OK, 2 rows affected (0.005 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql>
```

Check the data with command select * from Employees



```
mysql> create table Employees (Name varchar(20),Salary int);
Query OK, 0 rows affected (0.014 sec)

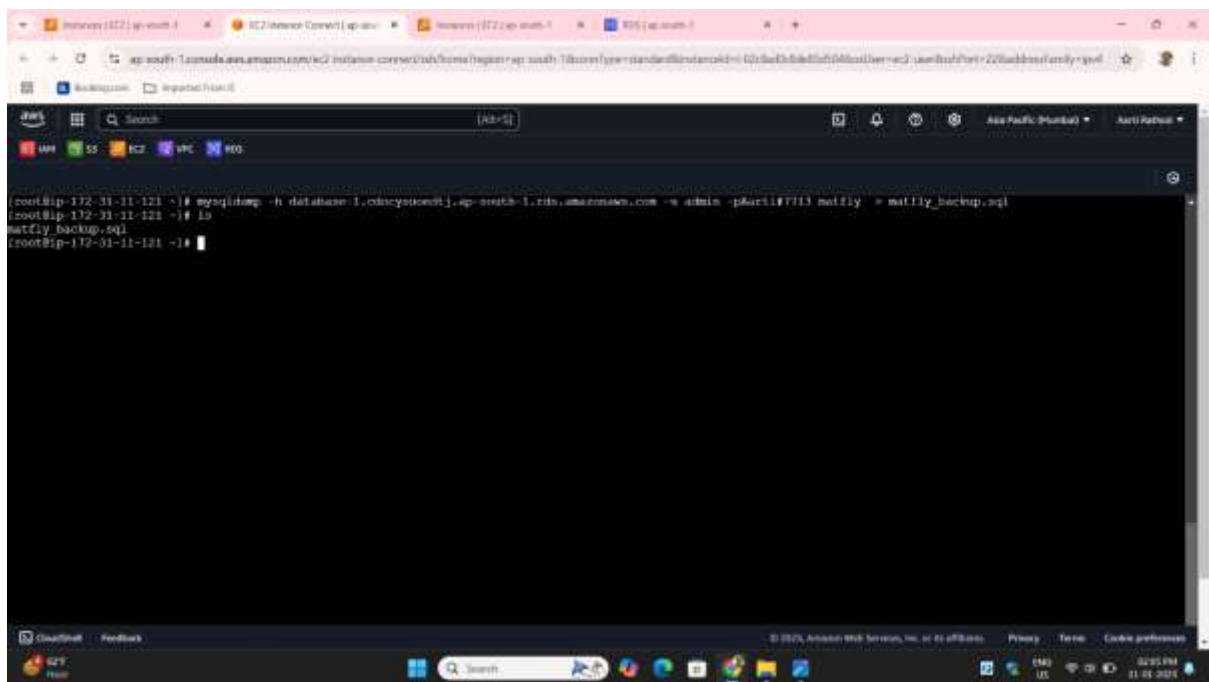
mysql> show tables;
+-----+
| Tables_in_matfly |
+-----+
| Employees         |
+-----+
1 row in set (0.001 sec)

mysql> insert into employees values ("Rohit",40000), ("prasad",30000);
ERROR 1146 (42000): Table 'matfly.employees' doesn't exist
mysql> insert into Employees values ("Rohit",40000), ("prasad",30000);
Query OK, 2 rows affected (0.005 sec)
Records: 2 Duplicates: 0 Warnings: 0

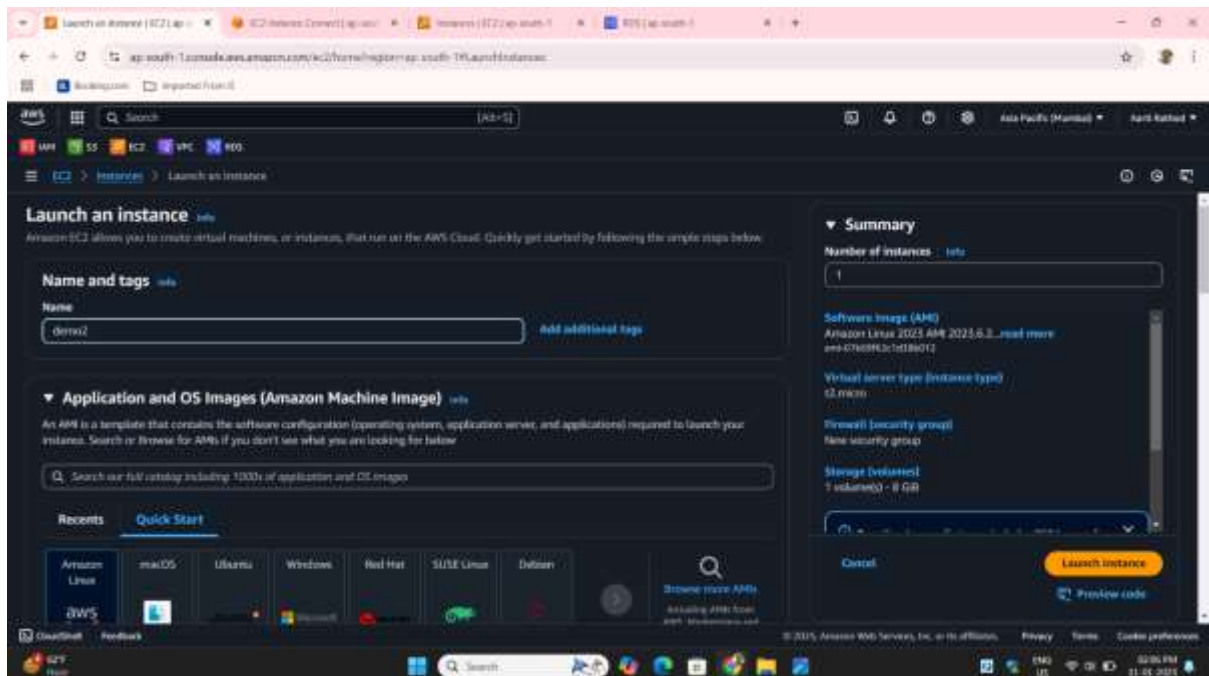
mysql> select * from Employees;
+-----+-----+
| Name | Salary |
+-----+-----+
| Rohit | 40000  |
| prasad | 30000  |
+-----+-----+
2 rows in set (0.001 sec)

mysql> exit
bye
root@ip-172-31-11-121 ~#
```

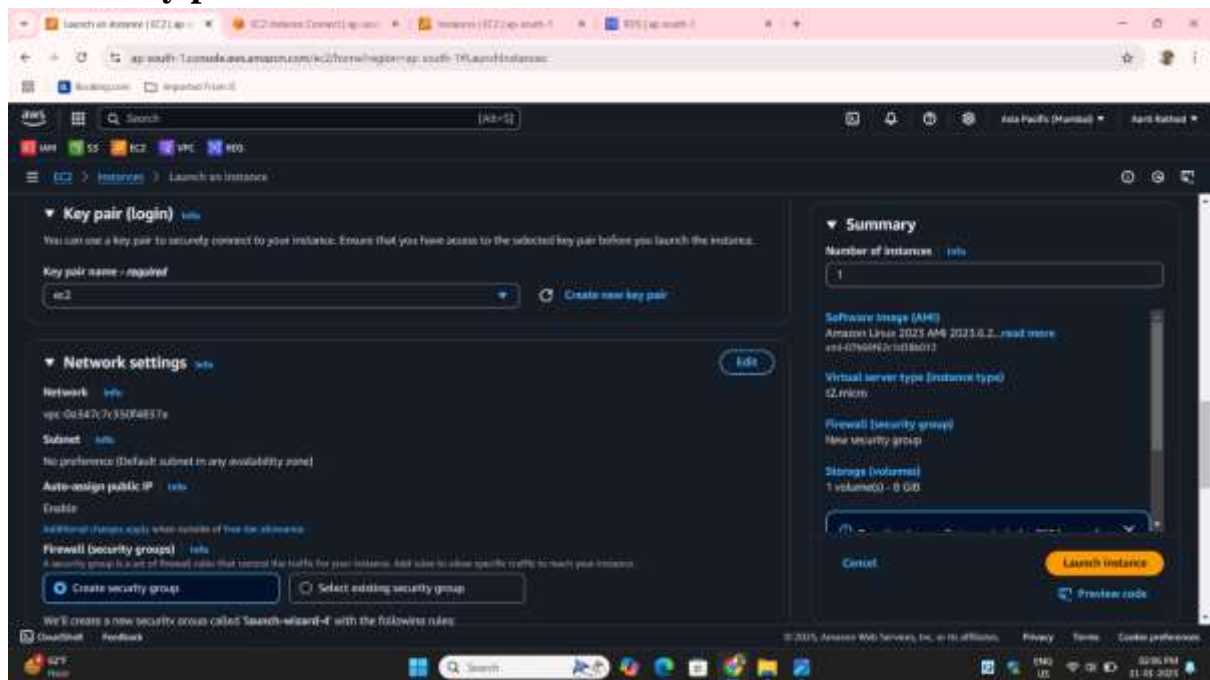
Take backup of the database with mysqldump command



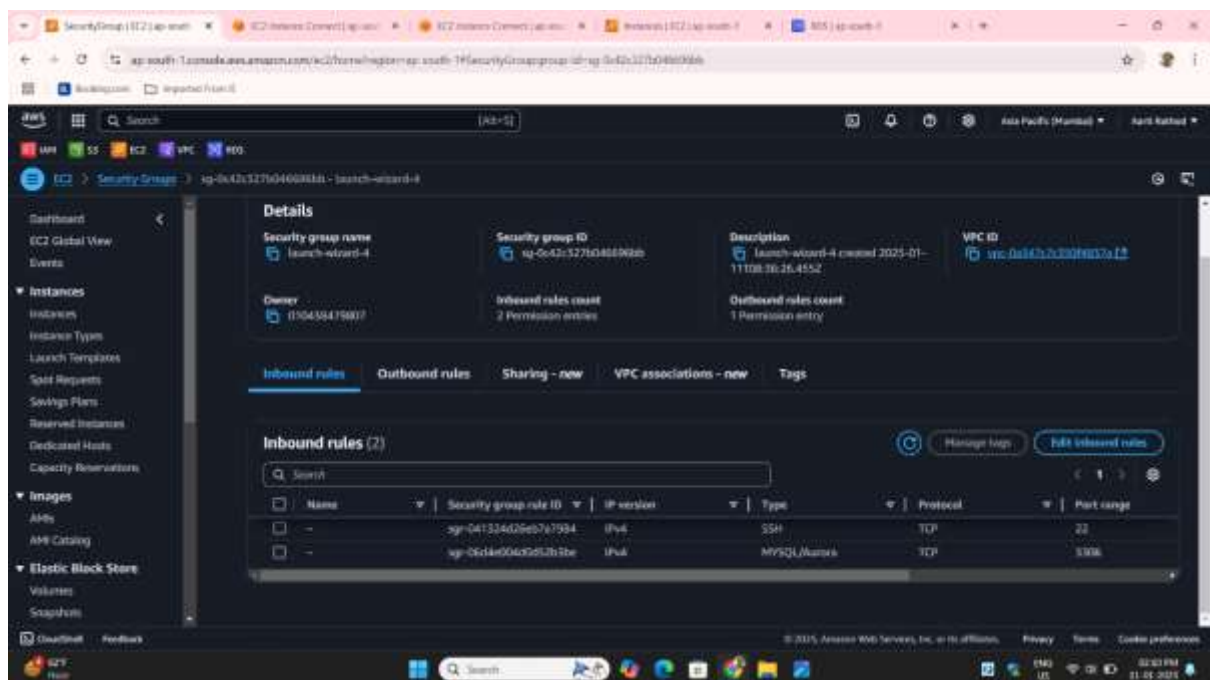
Launch another instance demo2



Attach key pair And launch instance



Go to security group edit inbound rules allow port 3306



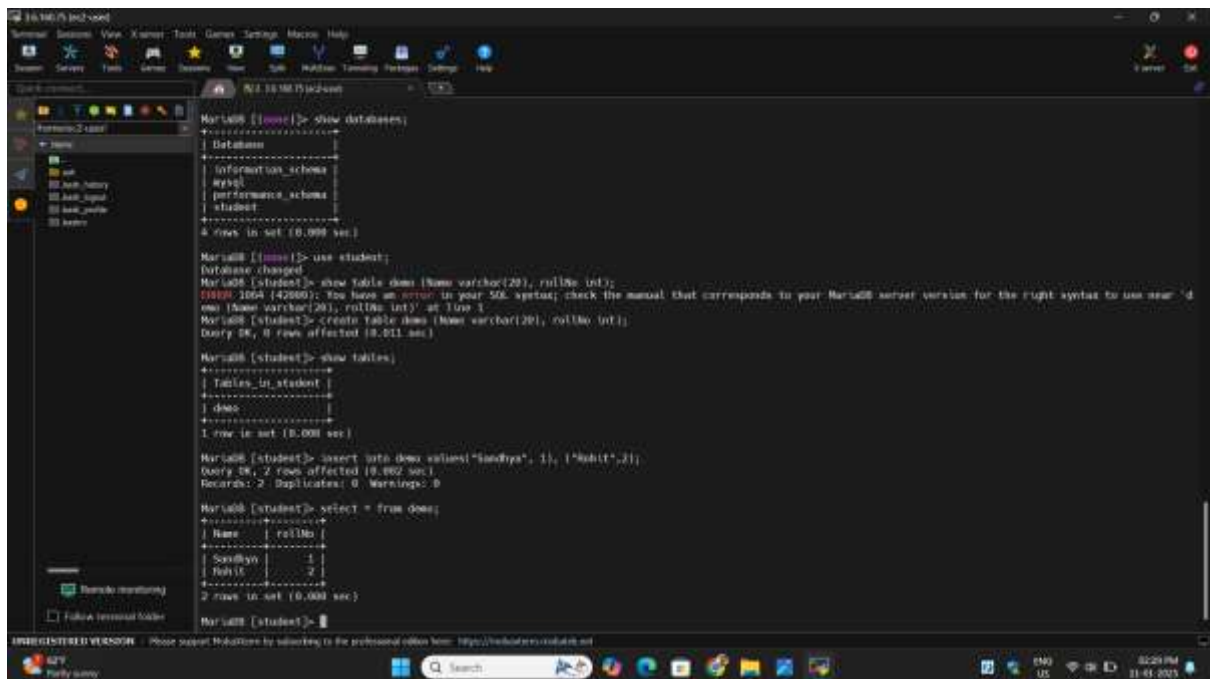
Connect that demo2 instace on MobaXterm switch to root using sudo -I command. Install mariadb. Start mariadb.

[illegible]

Start secure installation Set new password

[illegible]

Access local mariaDB create database student Create table demo insert



```

MariaDB [(root)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| student |
+-----+
4 rows in set (0.000 sec)

MariaDB [(root)]> use student;
Database changed
MariaDB [student]> show table demo (Name varchar(20), rollNo int);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'd
emo (Name varchar(20), rollNo int)' at line 1
MariaDB [student]> create table demo (Name varchar(20), rollNo int);
Query OK, 0 rows affected (0.011 sec)

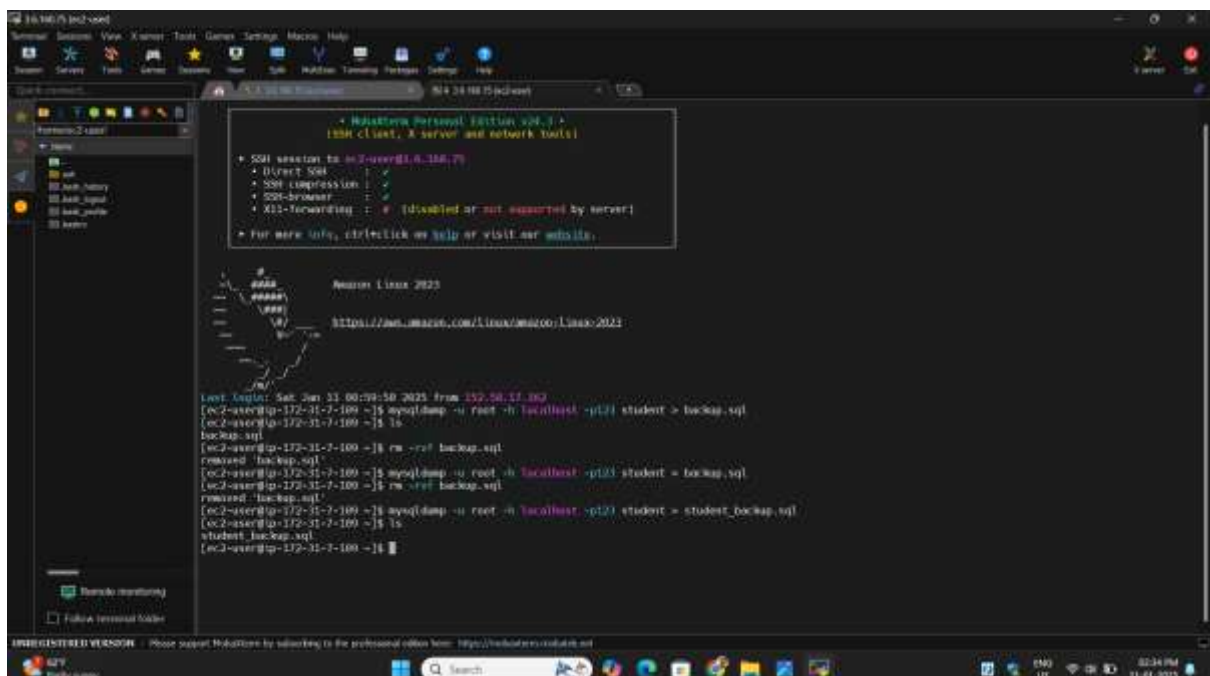
MariaDB [student]> show tables;
+-----+
| Tables_in_student |
+-----+
| demo |
+-----+
1 row in set (0.000 sec)

MariaDB [student]> insert into demo values('Sandeep', 1), ('Rohit', 2);
Query OK, 2 rows affected (0.002 sec)
Records: 2 Duplicates: 0 Warnings: 0

MariaDB [student]> select * from demo;
+-----+-----+
| Name | rollNo |
+-----+-----+
| Sandeep | 1 |
| Rohit | 2 |
+-----+-----+
2 rows in set (0.000 sec)

```

some data and exit the marisDB Create backup with mysqldump command
Dump the local database to a .sql file



```

+-----+
| Tables_in_student |
+-----+
| demo |
+-----+
1 row in set (0.000 sec)

MariaDB [student]> insert into demo values('Sandeep', 1), ('Rohit', 2);
Query OK, 2 rows affected (0.002 sec)
Records: 2 Duplicates: 0 Warnings: 0

MariaDB [student]> select * from demo;
+-----+-----+
| Name | rollNo |
+-----+-----+
| Sandeep | 1 |
| Rohit | 2 |
+-----+-----+
2 rows in set (0.000 sec)

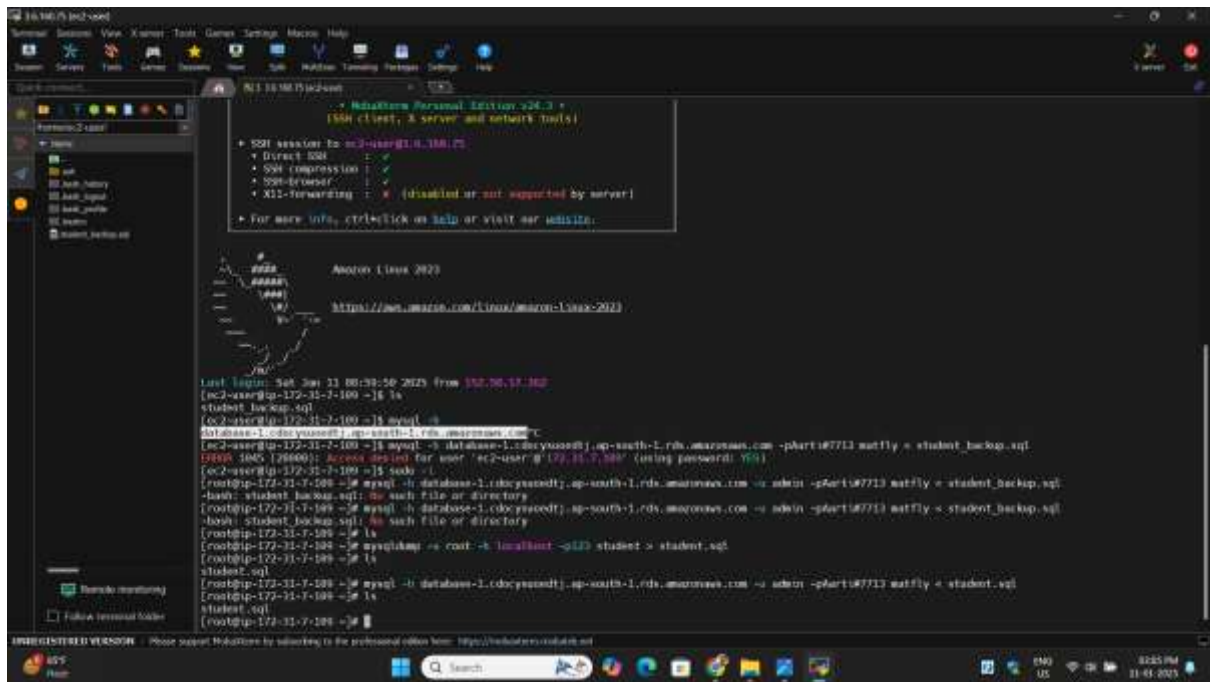
MariaDB [student]>

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023/

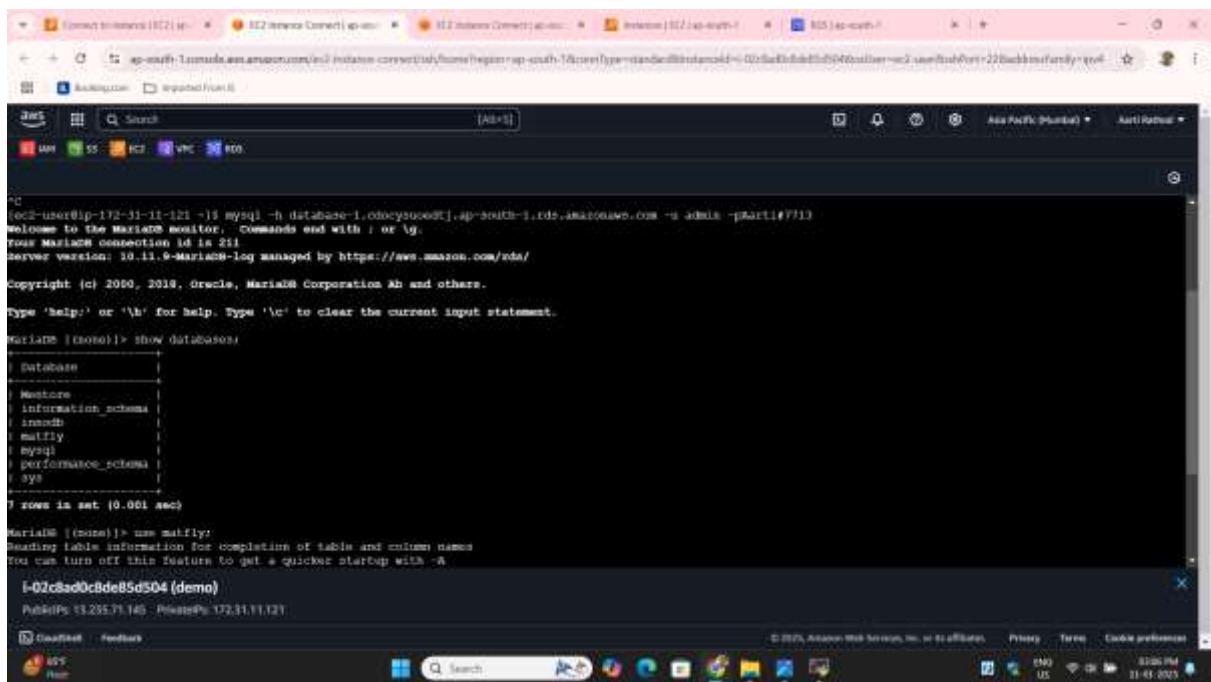
Last login: Sat Jun 15 00:09:50 2023 from 152.50.17.102
[ec2-user@ip-172-31-7-109 ~]$ mysqldump -u root -h localhost -p123 student > backup.sql
[ec2-user@ip-172-31-7-109 ~]$ ls
backup.sql
[ec2-user@ip-172-31-7-109 ~]$ rm -rf backup.sql
removed backup.sql
[ec2-user@ip-172-31-7-109 ~]$ mysqldump -u root -h localhost -p123 student > backup.sql
[ec2-user@ip-172-31-7-109 ~]$ rm -rf backup.sql
removed backup.sql
[ec2-user@ip-172-31-7-109 ~]$ mysqldump -u root -h localhost -p123 student > student_backup.sql
[ec2-user@ip-172-31-7-109 ~]$ ls
student_backup.sql
[ec2-user@ip-172-31-7-109 ~]$

```

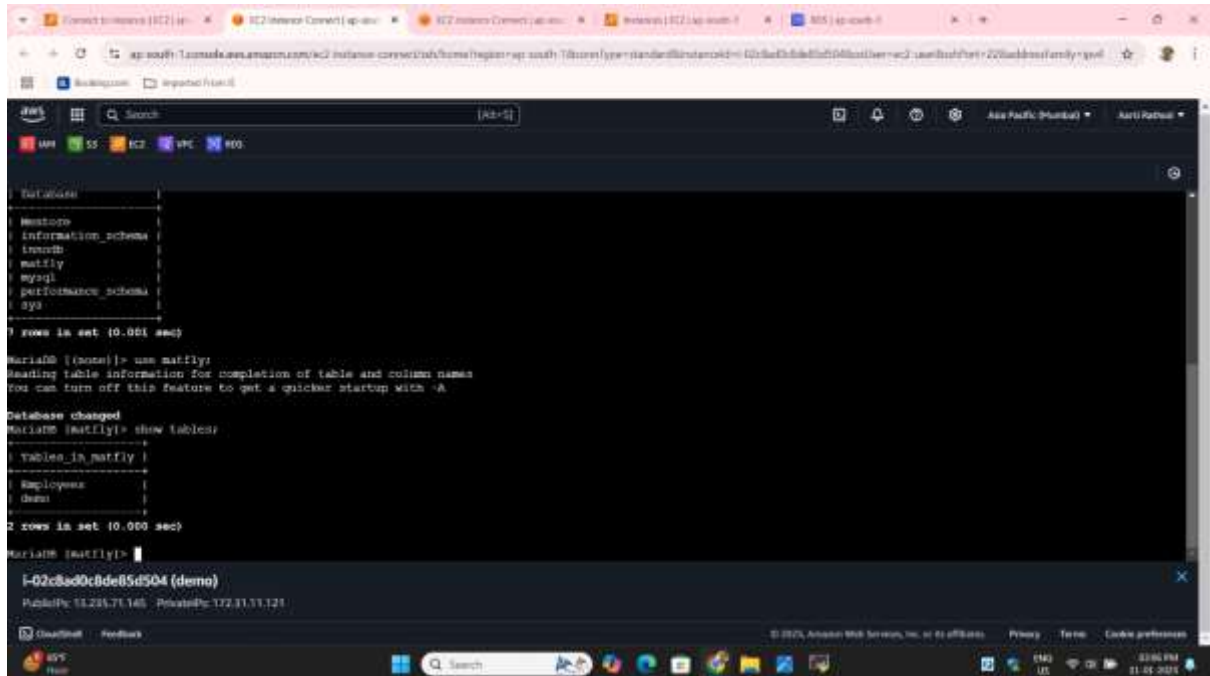
Import the backup to RDS



Connect the rds again with endpoint Run the command show tables



Here you can see the table which we have created in locally on ec2
this is how we can take back up



The screenshot shows an AWS CloudShell terminal window with the following content:

```
Database
|-----|
| hostodb |
| information_schema |
| innodb |
| mysql |
| performance_schema |
| sys |
|-----|

1 rows in set (0.001 sec)

MariaDB [(none)]> use matfly;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [matfly]> show tables;
+-----+
| Tables_in_matfly |
+-----+
| Employees |
| Users |
+-----+

2 rows in set (0.000 sec)

MariaDB [matfly]>
```

Below the terminal window, a metadata box shows the instance ID: `i-02c8ad0c8de85d504 (demo)`, with Public IP: `54.235.71.145` and Private IP: `172.31.31.121`.