

Migration of Database from EC2 to RDS (IAAS to PAAS)

Aim of Project

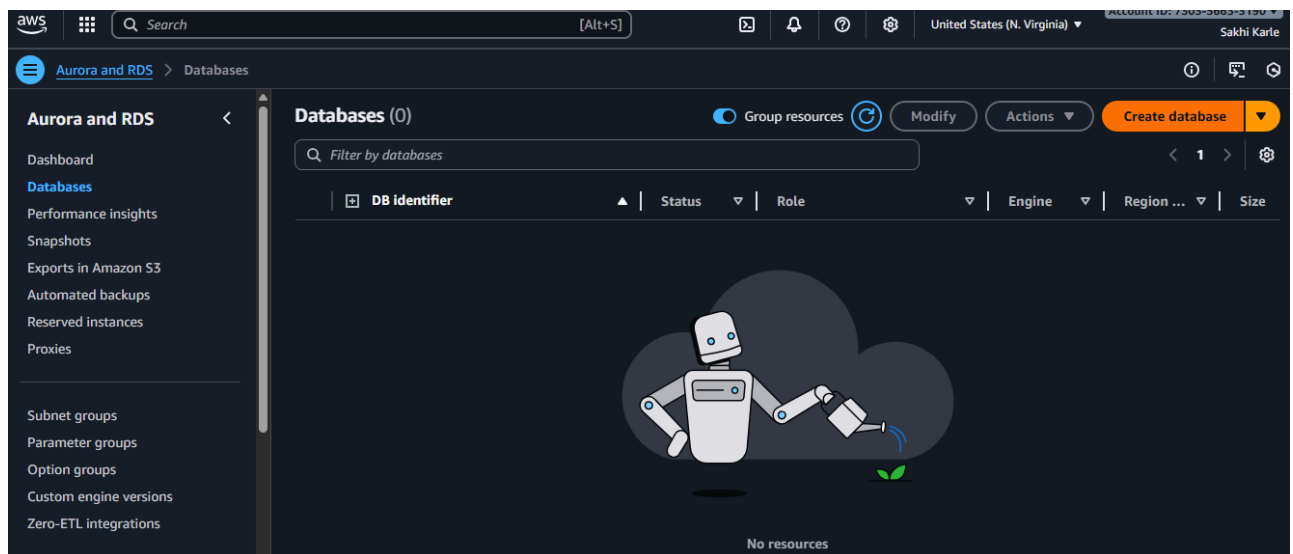
- To ensure automatic backups and high availability of data
- To understand the difference between Iaas and Paas
- To learn how to migrate data from EC2 MySQL to RDS MySQL
- To see how RDS makes database management easier.

Introduction

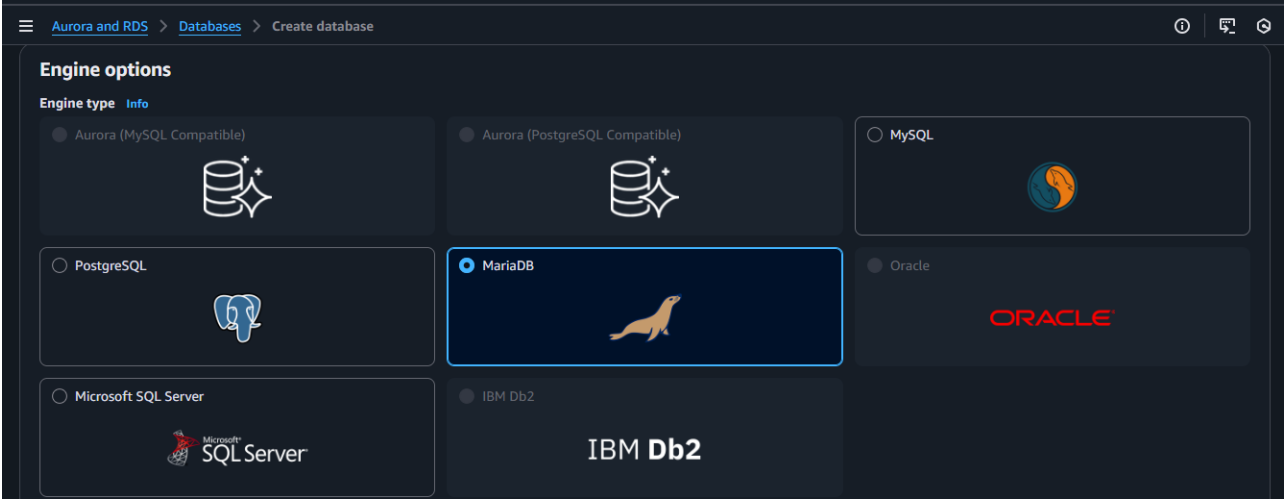
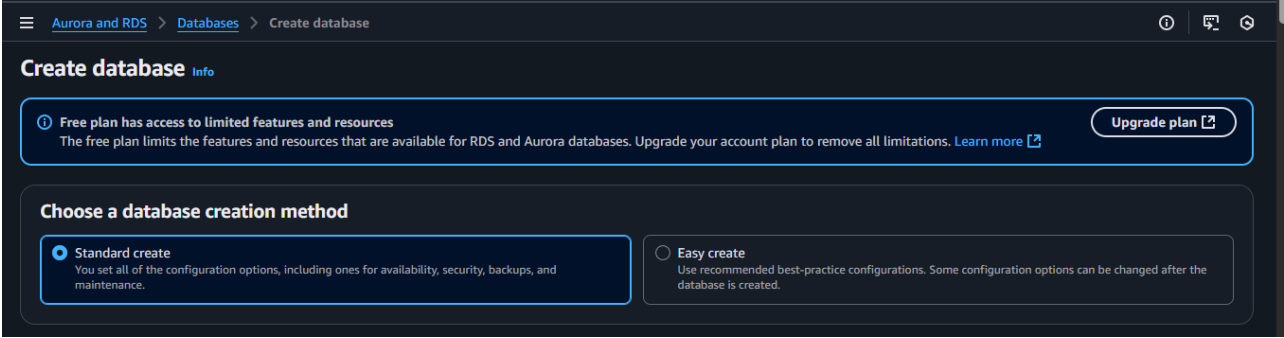
In this mini project, a database migration was performed from an EC2 instance to an Amazon RDS instance using the MariaDB engine. The goal was to move the Myntra database from a self-managed EC2 server to a fully managed RDS service for better scalability, security, and easy maintenance.

Step 1: Create Amazon RDS (Paas)

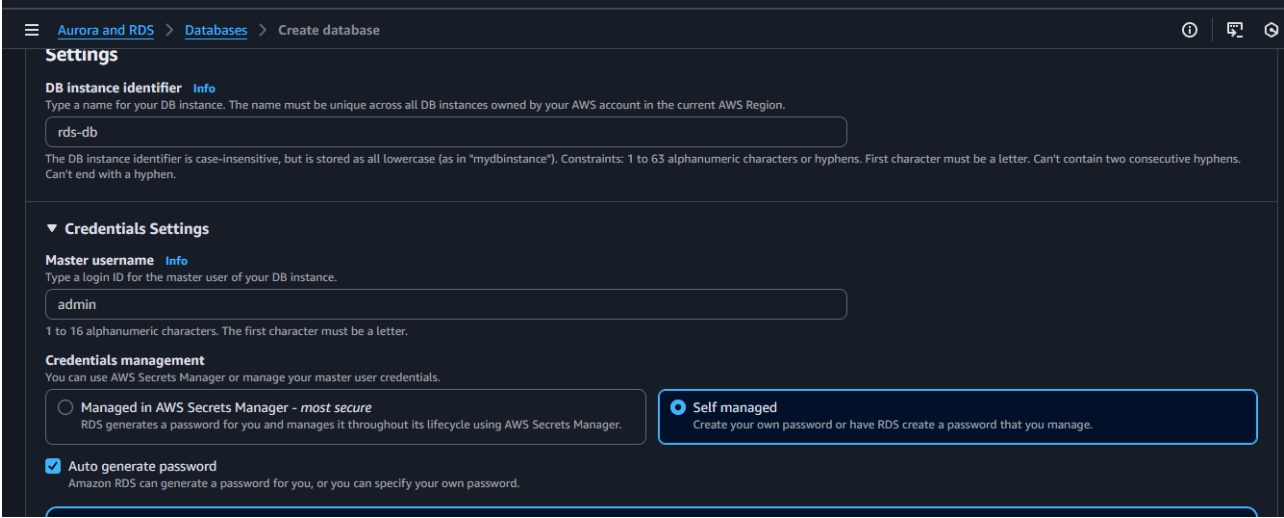
1. Click on create database

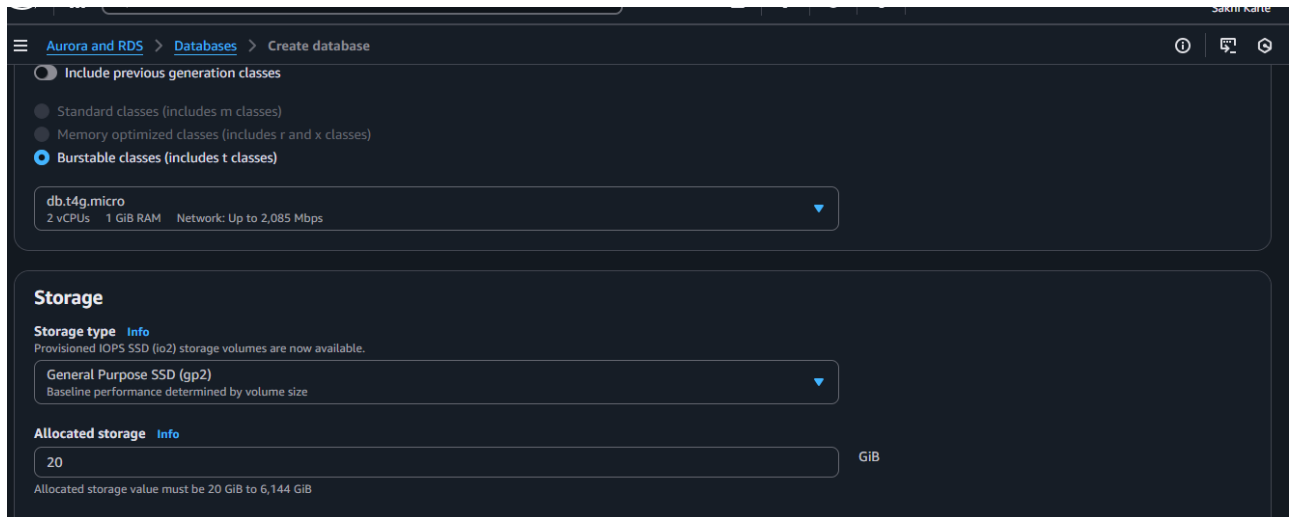


2. Select database creation method and engine type



3. Give name for your db instance and select security group





Aurora and RDS > Databases > Create database

☐ Include previous generation classes

- ☐ Standard classes (includes m classes)
- ☐ Memory optimized classes (includes r and x classes)
- ☒ Burstable classes (includes t classes)

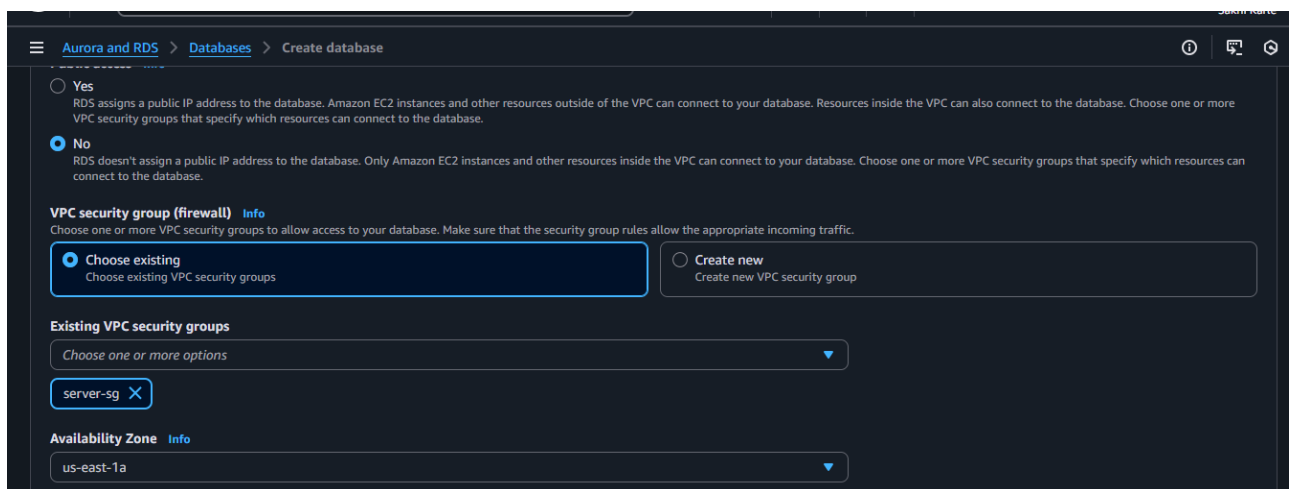
db.t4g.micro
2 vCPUs 1 GiB RAM Network: Up to 2,085 Mbps

Storage

Storage type [Info](#)
Provisioned IOPS SSD (io2) storage volumes are now available.

General Purpose SSD (gp2)
Baseline performance determined by volume size

Allocated storage [Info](#)
20 GiB
Allocated storage value must be 20 GiB to 6,144 GiB



Aurora and RDS > Databases > Create database

☐ Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

☒ No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

VPC security group (firewall) [Info](#)
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

☒ Choose existing
Choose existing VPC security groups

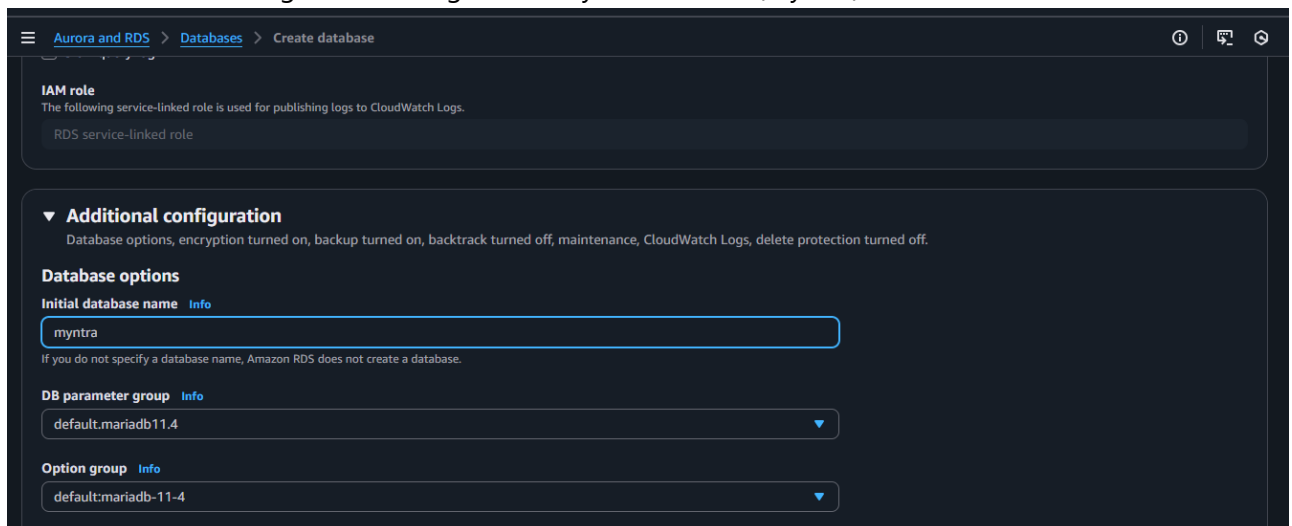
☐ Create new
Create new VPC security group

Existing VPC security groups
Choose one or more options

server-sg

Availability Zone [Info](#)
us-east-1a

3. Go to Additional configuration and give name your database(Myntna) and click on create database



Aurora and RDS > Databases > Create database

IAM role
The following service-linked role is used for publishing logs to CloudWatch Logs.
RDS service-linked role

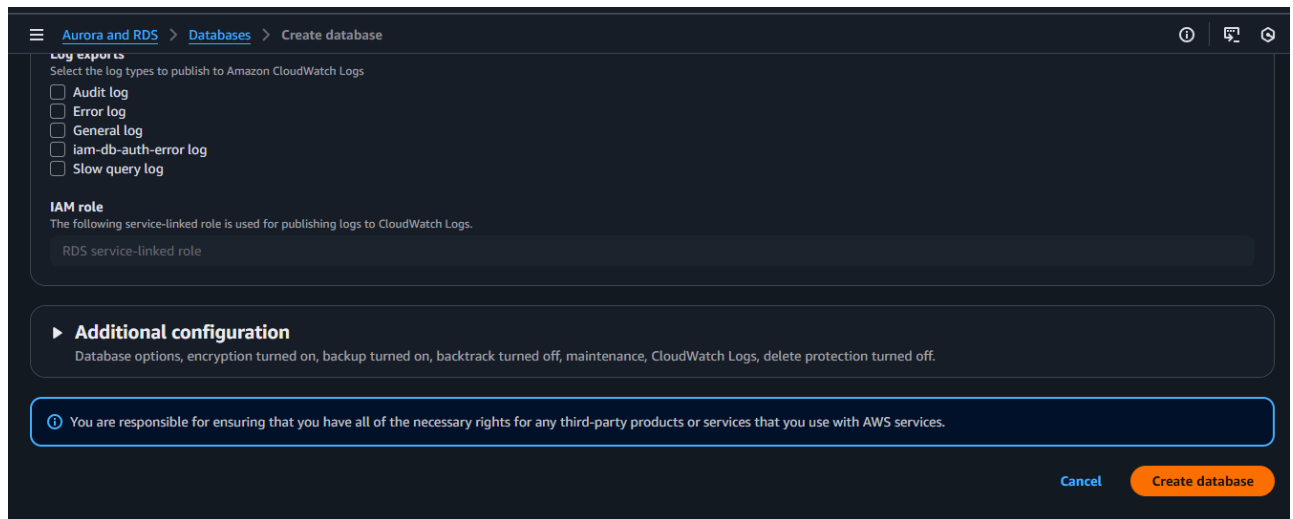
Additional configuration
Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Database options

Initial database name [Info](#)
myntna
If you do not specify a database name, Amazon RDS does not create a database.

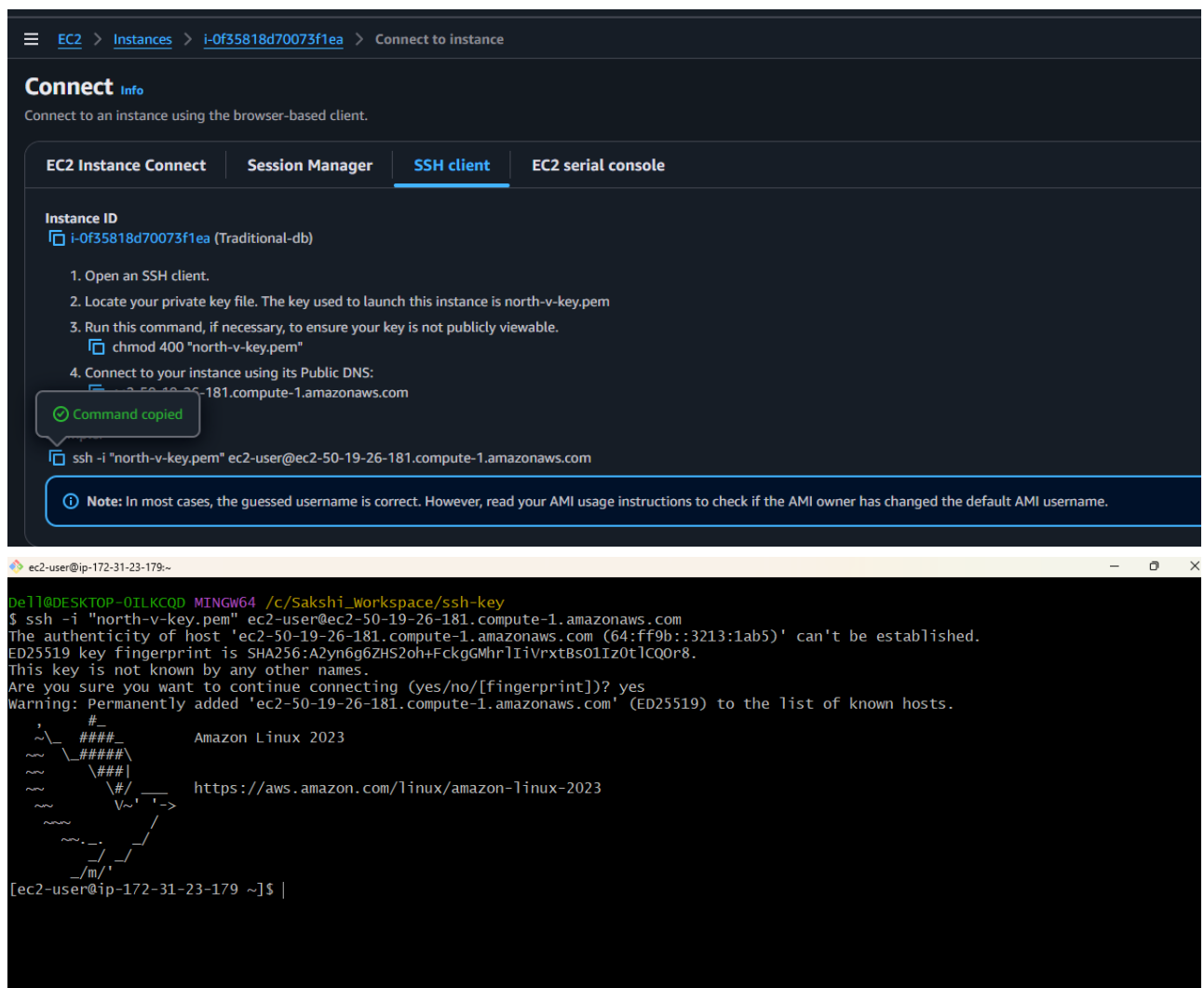
DB parameter group [Info](#)
default:mariadb11.4

Option group [Info](#)
default:mariadb-11-4



Step 2: Create Database on EC2(Iaas)

1. Launch and temporary EC2 instance
2. Copy ssh and take access of your EC2 instance



3. Install and start mariadb105-server

```
ec2-user@ip-172-31-23-179:~$ sudo yum update
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-23-179 ~]$ sudo yum install mariadb105-server -y
Last metadata expiration check: 0:00:43 ago on Fri Oct 10 06:09:59 2025.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
mariadb105-server	x86_64	3:10.5.29-1.amzn2023.0.1	amazonlinux	10 M
Installing dependencies:				
mariadb-connector-c	x86_64	3.3.10-1.amzn2023.0.1	amazonlinux	211 k
mariadb-connector-c-config	noarch	3.3.10-1.amzn2023.0.1	amazonlinux	9.9 k
mariadb105	x86_64	3:10.5.29-1.amzn2023.0.1	amazonlinux	1.5 M
mariadb105-common	x86_64	3:10.5.29-1.amzn2023.0.1	amazonlinux	28 k
mariadb105-errmsg	x86_64	3:10.5.29-1.amzn2023.0.1	amazonlinux	212 k
mysql-selinux	noarch	1.0.4-2.amzn2023.0.3	amazonlinux	36 k
perl-B	x86_64	1.80-477.amzn2023.0.7	amazonlinux	177 k
perl-DBD-MariaDB	x86_64	1.22-1.amzn2023.0.4	amazonlinux	153 k
perl-DBI	x86_64	1.643-7.amzn2023.0.3	amazonlinux	700 k
perl-Data-Dumper	x86_64	2.174-460.amzn2023.0.2	amazonlinux	55 k
perl-File-Copy	noarch	2.24-477.amzn2023.0.7	amazonlinux	20 k

```
ec2-user@ip-172-31-23-179:~$ sudo systemctl start mariadb
[ec2-user@ip-172-31-23-179 ~]$ sudo systemctl enable mariadb
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
[ec2-user@ip-172-31-23-179 ~]$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled)
   Active: active (running) since Fri 2025-10-10 06:12:04 UTC; 23s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 26456 (mariabdd)
   Status: "Taking your SQL requests now..."
    Tasks: 13 (limit: 1053)
   Memory: 66.4M
      CPU: 537ms
   CGroup: /system.slice/mariadb.service
           └─26456 /usr/libexec/mariabdd --basedir=/usr
```

4. Go to mysql

```
[ec2-user@ip-172-31-23-179 ~]$ sudo mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.5.29-MariaDB MariaDB Server

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)]> alter user root@localhost identified by "root";
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> exit
Bye
[ec2-user@ip-172-31-23-179 ~]$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.5.29-MariaDB MariaDB Server

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)]> |
```

5. Create database myntra and create table and insert values into table

```
MariaDB [(none)]> create database myntra;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> use myntra;
Database changed
MariaDB [myntra]> create table user (id int, name varchar(10), age int, addr varchar(10));
Query OK, 0 rows affected (0.008 sec)

MariaDB [myntra]> |
```

```
MariaDB [mynttra]> insert into user values(1, "riya", 20, "pune"), (2, "rohan", 22, "pune");
Query OK, 2 rows affected (0.001 sec)
Records: 2  Duplicates: 0  Warnings: 0

MariaDB [mynttra]>
```

Step 3: Extract data from EC2 server to convert into file

```
ec2-user@ip-172-31-23-179:~
[ec2-user@ip-172-31-23-179 ~]$ ls
[ec2-user@ip-172-31-23-179 ~]$ mysqldump -u root -p mynttra > mynttra_bkp.sql
Enter password:
[ec2-user@ip-172-31-23-179 ~]$ ls
mynttra_bkp.sql
[ec2-user@ip-172-31-23-179 ~]$
```

Step 4: Migrate data from EC2 to RDS

```
ec2-user@ip-172-31-23-179 ~]$ sudo mysql -h rds-database.c67myyqqc3fn.us-east-1.rds.amazonaws.com -u admin -p mynttra < mynttra_bkp.sql
Enter password:
[ec2-user@ip-172-31-23-179 ~]$ sudo mysql -h rds-database.c67myyqqc3fn.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 92
Server version: 11.4.5-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.
```

Step 5: Check the imported database from mynttra database

```
MariaDB [(none)]> show databases;
```

Database
information_schema
innodb
myntra
mysql
performance_schema
sys

```
6 rows in set (0.007 sec)
```

```
MariaDB [myntra]> show tables;
```

Tables_in_myntra
user

```
1 row in set (0.001 sec)
```

```
MariaDB [myntra]> select * from user;
```

id	name	age	addr
1	riya	20	pune
2	rohan	22	pune

```
2 rows in set (0.001 sec)
```

Conclusion:

The migration of the Myntra database from the EC2 instance to Amazon RDS was completed successfully. All database tables and records were verified and found to be correctly transferred. By using Amazon RDS, we no longer need to manage database maintenance manually — it saves time, provides automatic backups, and makes scaling much easier when needed.