

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

## Aim of Project

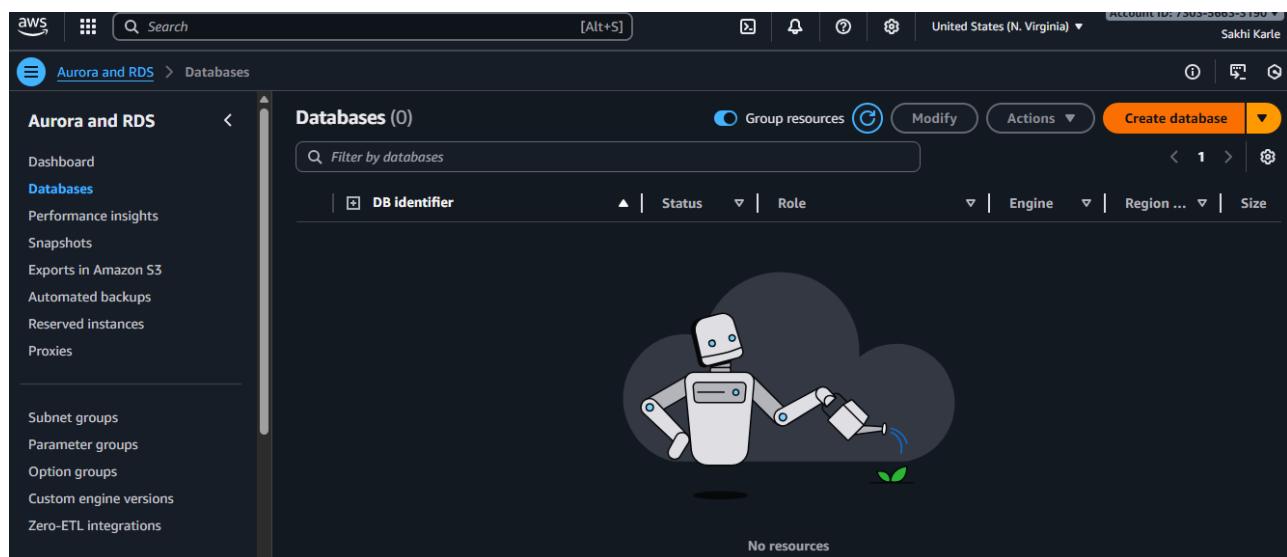
- To ensure automatic backups and high availability of data
- To understand the difference between IaaS and PaaS
- To learn how to migrate data from EC2 MySQL to RDS MySQL
- To see how RDS makes database management easier.

## 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

Aurora and RDS > Databases > Create database

### Create database Info

Free plan has access to limited features and resources  
The free plan limits the features and resources that are available for RDS and Aurora databases. Upgrade your account plan to remove all limitations. [Learn more](#)

[Upgrade plan](#)

#### 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.

Aurora and RDS > Databases > Create database

### Engine options

Engine type Info

- Aurora (MySQL Compatible) 
- Aurora (PostgreSQL Compatible) 
- MySQL 
- PostgreSQL 
- MariaDB 
- Oracle 
- Microsoft SQL Server 
- IBM Db2 

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

Aurora and RDS > Databases > 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.

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.

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.

The screenshot shows the 'Create database' step in the AWS RDS wizard. It's a two-step process: 'Storage' and 'Additional configuration'. In the first step, the user has selected a 'Burstable classes (includes t classes)' instance type, specifically 'db.t4g.micro' with 2 vCPUs, 1 GiB RAM, and up to 2,085 Mbps network. Under 'Storage', the 'General Purpose SSD (gp2)' option is chosen, and the 'Allocated storage' is set to 20 GiB. In the second step, under 'VPC', the 'No' option is selected, which prevents public IP assignment. The 'Choose existing' button for VPC security groups is highlighted, and it lists 'server-sg'. Other fields include 'Existing VPC security groups' (with a dropdown placeholder 'Choose one or more options') and 'Availability Zone' (set to 'us-east-1a').

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

This screenshot shows the 'Additional configuration' step of the RDS wizard. It includes sections for 'IAM role' (using the 'RDS service-linked role'), 'Database options' (with 'Initial database name' set to 'myntra'), 'DB parameter group' (set to 'default.mariadb11.4'), and 'Option group' (set to 'default:mariadb-11-4'). A note at the bottom of the 'Database options' section states: 'If you do not specify a database name, Amazon RDS does not create a database.'

**Aurora and RDS > Databases > Create database**

**Log exports**  
Select the log types to publish to Amazon CloudWatch Logs.

- Audit log
- Error log
- General log
- iam-db-auth-error log
- Slow query log

**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.

**Note:** You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

**Create database**

## Step 2: Create Database on EC2(IaaS)

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

**EC2 > Instances > i-0f35818d70073f1ea > Connect to instance**

**Connect** Info  
Connect to an instance using the browser-based client.

**EC2 Instance Connect** **Session Manager** **SSH client** **EC2 serial console**

**Instance ID**  
 i-0f35818d70073f1ea (Traditional-db)

1. Open an SSH client.  
2. Locate your private key file. The key used to launch this instance is north-v-key.pem  
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
 chmod 400 "north-v-key.pem"  
4. Connect to your instance using its Public DNS:  
 ec2-50-19-26-181.compute-1.amazonaws.com

**Command copied**

ssh -i "north-v-key.pem" ec2-user@ec2-50-19-26-181.compute-1.amazonaws.com

**Note:** In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

```

ec2-user@ip-172-31-23-179:~ 
DeTl@DESKTOP-0ILKQD MINGW64 /c/Sakshi_Workspace/ssh-key
$ ssh -i "north-v-key.pem" ec2-user@ec2-50-19-26-181.compute-1.amazonaws.com
The authenticity of host 'ec2-50-19-26-181.compute-1.amazonaws.com (64:ff9b::3213:1ab5)' can't be established.
ED25519 key fingerprint is SHA256:A2yn6g6ZHS2oh+FckgGMhrIiVrxtBs01Iz0tCQOr8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-50-19-26-181.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

,
  #_
  ~\_ #####_      Amazon Linux 2023
  ~\_\#\#\#\_
  ~\_\#\#\#
  ~\_\#\#\|_
  ~\_\#/`__>
  ~\_\~_.-`_/
  ~\_\~/`_/
[ec2-user@ip-172-31-23-179 ~]$ 

```

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   2.24-177.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 (mariadb)
  Status: "Taking your SQL requests now..."
   Tasks: 13 (limit: 1053)
  Memory: 66.4M
     CPU: 537ms
    CGroup: /system.slice/mariadb.service
           └─26456 /usr/libexec/mariadb --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 [myntra]> 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 [myntra]>
```

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

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