

Problem Statement: Many organizations start with self-managed MYSQL databases (community edition) on Servers due to flexibility and control. So, the task is to migrate the **MYSQL database on EC2 to Amazon RDS**.

Why Migrate:

1. AWS handles patches and backups
2. AWS offers Auto Scaling and High Availability
3. Enhanced performance and security

Pre-requisites:

1. Amazon Free Tier Account.
2. Knowledge on how to create IAM users and add permissions.
3. Mysqldump – a simple method for full database migration
4. AWS Database Migration Service – To enable real-time replication and minimize downtime

Steps:

1. Launch an EC2 instance
2. Choose Ubuntu
3. Select Free Tier
4. Security Groups - Allow the following SSH, MySQL Ports (3306) – Inbound Rule
5. Click on launch instance.
6. The steps are similar to the previous posts
7. Once the EC2 instance is launched, install MySQL-Server

```
root@ip-██████████:/home/ubuntu# sudo apt install mysql-server
```

8. Check the sql version and the status with the following commands

```
root@ip-██████████:/home/ubuntu# mysql --version
mysql Ver 8.0.41-0ubuntu0.24.04.1 for Linux on x86_64 ((Ubuntu))
root@ip-██████████:/home/ubuntu# sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-03-31 11:00:16 UTC; 2min 38s ago
     Process: 2205 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 2214 (mysqld)
      Status: "Server is operational"
        Tasks: 37 (limit: 1129)
      Memory: 349.6M (peak: 378.7M)
         CPU: 1.422s
       CGroup: /system.slice/mysql.service
               └─2214 /usr/sbin/mysqld

Mar 31 11:00:15 ip-██████████ systemd[1]: Starting mysql.service - MySQL Community Server...
Mar 31 11:00:16 ip-██████████ systemd[1]: Started mysql.service - MySQL Community Server.
root@ip-██████████:/home/ubuntu#
```

9. You can secure the SQL Installation with the following commands. Since it is a demo, will go with defaults settings

```
root@ip-: /home/ubuntu# sudo mysql_secure_installation
```

10. Connect to the MySQL database

```
root@ip-: /home/ubuntu# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.41-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.03 sec)

mysql>
```

11. Create a database and user as shown

```
mysql> CREATE DATABASE testdb;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| testdb |
+-----+
5 rows in set (0.00 sec)

mysql>
```

12. Type **\! Clear** to clear the console screen

- 13.

```
mysql> CREATE USER 'migrate_user'@'%' IDENTIFIED BY 'password123';
Query OK, 0 rows affected (0.05 sec)
```

14. The above command will create a user `migrate_user`
15. The `@%` means the user can connect from any ip address
16. If used `@localhost` then the user can connect only from the same server
17. If you want to restrict access to a specific IP you can mention `@'192.x.x.x'`
- 18.

```
mysql> CREATE USER 'migrate_user'@'%' IDENTIFIED BY 'password123';
Query OK, 0 rows affected (0.05 sec)

mysql> GRANT ALL PRIVILEGES ON testdb.* TO 'migrate_user'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

19. Brief explanation of the above code
 - a. Gives the user full access to perform all actions (SELECT, INSERT, UPDATE, DELETE, etc.).
 - b. **ON testdb.*** → Grants access to **all tables** inside the **testdb**
 - c. **TO 'migrate_user'@'%'** → Specifies that the privileges are assigned to the `migrate_user`.
 - d. FLUSH means reload the user privilege table immediately

20. Create an RDS MYSQL Instance

Create database

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

Create database

Note: your DB instances will launch in the **US East (N. Virginia)** region

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)



☒ MySQL



☐ MariaDB



☐ Microsoft SQL Server



☐ Aurora (PostgreSQL Compatible)



☐ PostgreSQL



☐ Oracle



☐ IBM Db2



Edition

☒ MySQL Community

Engine version [info](#)

View the engine versions that support the following database features.

▼ Hide filters

☒ Show only versions that support the Multi-AZ DB cluster [info](#)

Create a Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 30 seconds.

☒ Show only versions that support the Amazon RDS Optimized Writes [info](#)

Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine version

☒ MySQL 8.0.40

☐ Enable RDS Extended Support [info](#)

Amazon RDS Extended Support is a [paid offering](#) by selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the RDS for MySQL [documentation](#).

Templates

Choose a 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:

- 99.99% uptime
- Redundancy across Availability Zones
- Increased read capacity
- Reduced write latency

Write/read endpoint

AZ 1

Primary instance + SSD

Reader endpoints

AZ 2

Readable standby + SSD

AZ 3

Readable standby + 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:

- 99.99% uptime
- Redundancy across Availability Zones

Write/read endpoint

AZ 1

Primary instance

Standby (no endpoint)

AZ 2

Standby

Single-AZ DB instance deployment (1 instance)

Creates a single DB instance without standby instances. This setup provides:

- 99.9% uptime
- No data redundancy

Write/read endpoint

AZ 1

Primary instance

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 "myrdsinstance"). 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.

Use 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](#)

Connectivity info

Compute resource
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

☐ Don't connect to an EC2 compute resource.
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

☒ **Connect to an EC2 compute resource**
Set up a connection to an EC2 compute resource for this database.

EC2 instance info
Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

EC2 instance info
Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

Virtual private cloud (VPC) info
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Default VPC info
Default VPC: **default-vpc-73a7c714**
Subnets: **0** Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

DB subnet group info
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

☒ **Choose existing**
Choose existing DB subnet group

Existing DB subnet groups
default-vpc-041c0f73a7c714
Subnets: **0** Availability Zones

☐ **Automatic setup**
RDS creates a new subnet group for you or reuses an existing subnet group.

Public access info
☐ **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

Additional VPC security group
Choose one or more options

default X

Amazon RDS will add a new VPC security group **rds-ec2-1** to allow connectivity with your compute resource.

21. Click on create database.
22. On the EC2 Ubuntu server, migrate the database using mysqldump.
23. Provide the necessary process permissions for the command to execute

```
mysql> GRANT PROCESS, SELECT, RELOAD, LOCK TABLES, REPLICATION CLIENT ON *.* TO 'migrate_user'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql>
```


24. In the EC2 SSH Terminal type the following command

```
root@ip-10.10.10.10: /home/ubuntu# mysqldump -u migrate_user -p --databases testdb > testdb.sql
Enter password:
root@ip-10.10.10.10: /home/ubuntu# ls
testdb.sql
root@ip-10.10.10.10: /home/ubuntu#
```

25. Now we will copy the dump file to the RDS Instance
26. Note down the end point of the RDS instance

Connectivity & security

Endpoint & port

Endpoint
 myrds-**cg9iykak48zv.us-east-1.rds.amazonaws.com**

Port
3306

27. In the SSH terminal type the following command – copy the dump file to the RDS instance

```
root@ip-10.10.10.10: /home/ubuntu# mysql -h myrds-db.cg9iykak48zv.us-east-1.rds.amazonaws.com -u admin -p < testdb.sql
Enter password:
```

28. Verify migration in the RDS instance by connecting to the RDS endpoint.

```
root@ip-10-10-10-10: /home/ubuntu# mysql -h myrds-db.cg9iykak48zv.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 30
Server version: 8.0.40 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| testdb |
+-----+
5 rows in set (0.00 sec)
```

29. To verify the version type the following command. Check the RDS version and it matches.

```
mysql> SELECT VERSION();
+-----+
| VERSION() |
+-----+
| 8.0.40 |
+-----+
```

DB identifier	Status	Role	Engine	Engine version
myrds-db	Available	Instance	MySQL Community	8.0.40

30. Select the hostname and it will show RDS-specific naming such as IP-xx.xx.xx and it likely an RDS instance.

```
mysql> SELECT @@hostname;
+-----+
| @@hostname |
+-----+
| ip-10-10-10-10 |
+-----+
1 row in set (0.00 sec)
```

The migration is complete.

I will create a table called users in the rds instance. Subsequently there is no table called users in the EC2 instance SQL Server.

```
mysql> CREATE TABLE users ( name varchar(255), age varchar(255));
Query OK, 0 rows affected (0.06 sec)

mysql> show tables;
+-----+
| Tables_in_testdb |
+-----+
| users             |
+-----+
1 row in set (0.00 sec)
```

EC2 SQL Instance – Connection – There no table called users in testdb.

```
root@ip-172-31-84-20:/home/ubuntu# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.41-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> show databases;
ERROR 1064 (42000): You have an error in your SQL syntax; c
line 1
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql       |
| performance_schema |
| sys        |
| testdb     |
+-----+
5 rows in set (0.00 sec)

mysql> use testdb;
Database changed
mysql> show tables;
Empty set (0.01 sec)

mysql> |
```

Note: Optional

31. If we have a large database, AWS Database migration is a better option.
32. Click on Get Started
33. Select Migrate
34. Select Instance Based Migrations
35. Create replication instance
36. Select Free Tier Instance (dme.t3.micro)
37. Create Source end point will be EC2 MYSQL
38. EC2 Private IP and the migrate_user credentials
39. Create Target Endpoint (RDS MYSQL)
40. Select MYSQL as Target
41. Enter RDS Endpoint and credentials
42. Create a migration task
43. Select Full Load + ongoing replication

44. Choose testdb as the database

45. Start the migration