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

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

11. Create a database and user as shown

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)

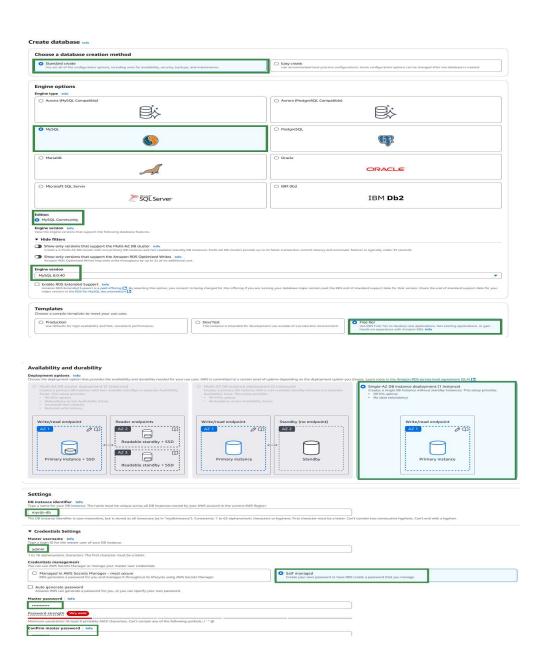
mysql> mysql> rows affected (0.00 sec)
```

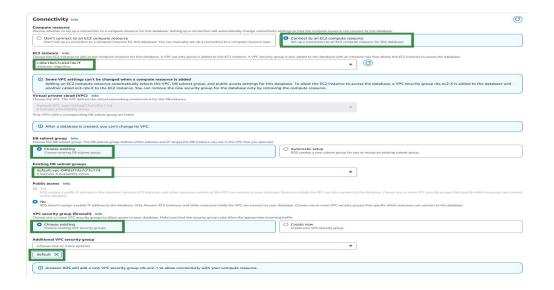
19. Brief explanation of the above code

- 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







- 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



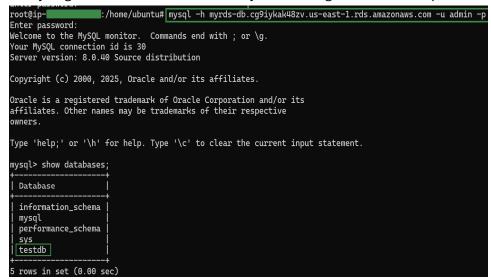
- 25. Now we will copy the dump file to the RDS Instance
- 26. Note down the end point of the RDS instance



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

```
root@ip-____:/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.



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



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

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.

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

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