**Task on RDS**

**1) Create mariadb db on ec2.**

**MariaDB Installation**

Update the system and install the MariaDB server package suitable for Amazon Linux 2023 (AL2023). The default version available is 10.5.

bash

sudo dnf update -y

sudo dnf search mariadb

sudo dnf install -y mariadb105-server

Enable and start the MariaDB service, then verify its status.

bash

sudo systemctl enable --now mariadb

systemctl status mariadb

Check if MariaDB is listening on port 3306.

bash

lsof -i :3306

**Database Setup**

Set environment variables for the database configuration.

bash

DBName=ec2db

DBPassword=admin123456

DBRootPassword=admin123456

DBUser=ec2dbuser

Create a script to set up the database, user, and privileges.

bash

echo "CREATE DATABASE $DBName;" > tmpdb.setup

echo "CREATE USER '$DBUser'@'localhost' IDENTIFIED BY '$DBPassword';" >> tmpdb.setup

echo "GRANT ALL PRIVILEGES ON \*.\* TO '$DBUser'@'localhost';" >> tmpdb.setup

echo "FLUSH PRIVILEGES;" >> tmpdb.setup

Set the root password and execute the setup script.

bash

mysqladmin -u root password "$DBRootPassword"

mysql -u root --password="$DBRootPassword" < tmpdb.setup

rm tmpdb.setup

Verify the database creation.

bash

mysql -u root -p -e "SHOW DATABASES;"

**A black background with a black square

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**2) Insert some dummy data**

To insert some dummy data into the MariaDB database on your EC2 instance, follow these steps:

**Step 1: Connect to MariaDB on EC2**

Use the following command to log in:

text

mysql -u root -p

Enter the root password when prompted.

**Step 2: Switch to the created database**

sql

**USE** ec2db;

**Step 3: Create a sample table**

Run the following SQL command to create a sample table:

sql

**CREATE** **TABLE** users (

id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

name **VARCHAR**(50),

email **VARCHAR**(50),

created\_at **TIMESTAMP** **DEFAULT** **CURRENT\_TIMESTAMP**

);

**Step 4: Insert dummy data**

Insert sample records into the users table:

sql

**INSERT** **INTO** users (name, email) **VALUES**

('John Doe', 'john@example.com'),

('Jane Smith', 'jane@example.com'),

('Alice Johnson', 'alice@example.com');

**Step 5: Verify data insertion**

Check the inserted data:

sql

**SELECT** \* **FROM** users;

This process will add dummy data into your MariaDB database on EC2. If you'd like, I can generate specific SQL commands for other tables or data types.

**A computer screen with white text

AI-generated content may be incorrect.**

**3) Take the backup of dummy data on ec2**

To take a backup of the dummy data in your MariaDB database on EC2, follow these steps:

**Step 1: Use mysqldump to backup database**

Run this command on the EC2 instance:

text

mysqldump -u root -p ec2db > tmpec2db.sql

Enter the root password when prompted. This will create a dump file named tmpec2db.sql containing the database backup.

**Step 2: Verify the backup file creation**

Check if the dump file exists and its size:

text

ls -lh tmpec2db.sql

**Step 3: Optional - View the start of the dump file**

text

head tmpec2db.sql

This will create a backup of your MariaDB database ec2db in the file tmpec2db.sql on the EC2 instance, which can be used later for restoring or migrating the database. Let me know if you want instructions on restoring or migrating this backup.To create a MariaDB database on an EC2 instance as described in your document, follow these steps:

1. Update system packages:

text

sudo dnf update -y

1. Search available MariaDB versions:

text

sudo dnf search mariadb

1. Install MariaDB server (version 10.5 as default on AL2023):

text

sudo dnf install -y mariadb105-server

1. Enable and start MariaDB service:

text

sudo systemctl enable --now mariadb

1. Check MariaDB service status:

text

systemctl status mariadb

1. Check if MariaDB listens on port 3306:

text

lsof -i tcp:3306

1. Set environment variables:

text

DBName=ec2db

DBUser=ec2dbuser

DBPassword=admin123456

DBRootPassword=admin123456

1. Create database and user with grants via SQL commands saved to a file:

text

echo CREATE DATABASE $DBName > tmpdb.setup

echo CREATE USER $DBUser IDENTIFIED BY '$DBPassword' >> tmpdb.setup

echo GRANT ALL PRIVILEGES ON \*.\* TO $DBUser >> tmpdb.setup

echo FLUSH PRIVILEGES >> tmpdb.setup

1. Set root password for MariaDB:

text

mysqladmin -u root password $DBRootPassword

1. Apply SQL setup file:

text

mysql -u root --password=$DBRootPassword < tmpdb.setup

rm tmpdb.setup

1. Verify database creation:

text

mysql -u root -p

SHOW DATABASES;

This will successfully create a MariaDB database on your EC2 instance.

**A computer screen with text on it

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**4) launch MariaDB RDS instance.**

To launch a MariaDB RDS instance in AWS, follow these general steps:

**1. Sign in to AWS Management Console**

Navigate to the RDS service in AWS.

**2. Choose "Create database"**

Select the **"Create database"** button and choose the **MariaDB** engine.

**3. Select the database creation method**

* **Standard Create**: For full control over configuration.
* **Easy Create**: For quick setup with defaults.

**4. Configure the database settings**

* **Engine options**: Choose MariaDB, e.g., version 10.5 or later.
* **Templates**: Choose "Production" or "Dev/Test" based on your needs.
* **DB instance identifier**: Enter a unique name.
* **Master username**: Specify a username (default is "admin").
* **Master password**: Set a secure password or use the one from your EC2 setup ("admin123456" if following previous steps).

**5. Configure instance specifications**

Choose instance type, storage size, and network settings (VPC, Subnets, security groups).

**6. Set additional configurations**

* Enable Multi-AZ deployment if needed.
* Configure backup, maintenance, and monitoring options.

**7. Launch the instance**

Review all configurations and click **"Create database"**.

**8. Wait for the instance to be available**

It may take several minutes to provision.

**9. Connect to your RDS instance**

Use the endpoint provided in the RDS console, along with the master username and password, to connect.

A screenshot of a computer

AI-generated content may be incorrect.

**5) Migrate database from ec2 to RDS.**

To migrate a database from your EC2 instance to an RDS MariaDB instance, follow these steps:

**Step 1: Export the database from EC2**

On your EC2 instance, create a dump of the existing database:

bash

mysqldump -u root -p ec2db > ec2db\_backup.sql

Enter the root password when prompted.

**Step 2: Upload the dump file to a reachable location**

Transfer ec2db\_backup.sql to a location accessible from your local machine or directly to AWS S3 for easier handling.

**Step 3: Import the dump into RDS**

* **Using MySQL client:**

bash

mysql -h <RDS-endpoint> -u <master-username> -p <database\_name> < ec2db\_backup.sql

Replace <RDS-endpoint>, <master-username>, and <database\_name> with your RDS details.

Get the dump of your existing DB on EC2  
Taking backup → mysqldump -u root -p ec2db >   
ec2db.sql  
2. Login to maridb  
mysql -h database-1.c030o8c66sum.us-  
east-1.rds.amazonaws.com -P 3306 -u admin -p  
3. Create database rdsdb; and exit  
4. Migrate the DB dump that you have taken in step 1 to RDS.  
mysql -h database-1.c030o8c66sum.us-  
east-1.rds.amazonaws.com -P 3306 -u admin -p rdsdb < ec2db.sql  
5. Connect to your RDS DB instance  
 mysql -h database-1.c030o8c66sum.us-  
east-1.rds.amazonaws.com -P 3306 -u admin -p  
6. Show databases;

7.  
8. Use rdsdb;  
9. Show tables;  
10. Describe student;  
11. Select \* from student;

**A black background with a black square

AI-generated content may be incorrect.**

**A computer screen shot of a black background

AI-generated content may be incorrect.**

**6) Install MySQL DB on ec2**

To install MySQL DB on your EC2 instance, follow these steps:

**Step 1: Update the system packages**

text

sudo dnf update -y

**Step 2: Install MySQL server**

You can install the MySQL server package appropriate for your OS (Amazon Linux 2023). For example:

text

sudo dnf install -y mysql-server

**Step 3: Enable and start the MySQL service**

text

sudo systemctl enable --now mysqld

**Step 4: Check the status of MySQL service**

text

systemctl status mysqld

**Step 5: Secure MySQL installation (optional but recommended)**

Run the security script to set root password and remove default insecure settings:

text

sudo mysql\_secure\_installation

This will prompt you to:

* Set root password
* Remove anonymous users
* Disallow remote root login
* Remove test database

**Step 6: Login as root**

text

mysql -u root -p

Enter the root password you set during the secure installation.

Once installed, you can create databases, users, and begin using MySQL as needed.

Let me know if you need detailed commands for database and user setup on MySQL.To install MySQL DB on your EC2 instance, follow these steps:

1. Update the system packages:

text

sudo dnf update -y

1. Install the MySQL server:

text

sudo dnf install -y mysql-server

1. Enable and start the MySQL service:

text

sudo systemctl enable --now mysqld

1. Check the MySQL service status:

text

systemctl status mysqld

1. Optionally, run the security script to set root password and secure installation:

text

sudo mysql\_secure\_installation

1. Login to MySQL as root:

text

mysql -u root -p

Use the root password you set during the secure installation.

This installs and starts MySQL on your EC2 instance, ready for database setup.

A computer screen shot of a computer screen

AI-generated content may be incorrect.A computer screen with white text

AI-generated content may be incorrect.A screenshot of a computer screen

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer screen

AI-generated content may be incorrect.A screenshot of a computer screen

AI-generated content may be incorrect.A computer screen with white text and icons

AI-generated content may be incorrect.

**7) Launch MySQL RDS image**

To launch a MySQL RDS instance, follow these steps:

1. **Sign in to AWS Console** and open the RDS service.
2. Click **Create database**.
3. Choose **MySQL** as the database engine.
4. Select the creation method:
   * **Standard Create** for detailed options.
   * **Easy Create** for default settings.
5. Configure the database:
   * Select MySQL version as per your needs (e.g., 8.0).
   * Set a unique DB instance identifier.
   * Set master username (default is admin) and password (e.g., admin123456).
6. Choose instance class, storage type, and size.
7. Set networking options (VPC, subnet groups, security groups).
8. Configure additional settings:
   * Enable Multi-AZ deployment if required.
   * Set backup retention and monitoring preferences.
9. Review all settings and click **Create database**.
10. Wait for the status to show **Available**.
11. Use the provided endpoint to connect to the MySQL RDS instance:

text

mysql -h <RDS-endpoint> -u <master-username> -p

Enter your master password when prompted.

This process launches a MySQL database instance on Amazon RDS ready for your applications.

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**8) Configure multi AZ**

To configure Multi-AZ deployment for your RDS database instance, follow these steps:

1. Go to the AWS Management Console and open the RDS service.
2. When creating a new database or modifying an existing one, locate the option for **Availability & durability**.
3. Enable **Multi-AZ deployment** by selecting the checkbox or toggle switch.
4. Multi-AZ will provision a synchronous standby replica in a different Availability Zone (AZ) to ensure high availability and failover support.
5. Configure any additional settings like backup retention, maintenance windows, and monitoring preferences.
6. Complete the creation or modification to apply Multi-AZ configuration.

This setup ensures automatic failover in case the primary database instance becomes unavailable, improving the resilience of your database environment.

**9) Take Backup of DB and restore the DB**

To take a backup of your database and restore it, follow these steps:

**Backup Database (on EC2 or source server)**

1. Use mysqldump to export the database to a SQL dump file:

bash

mysqldump -u root -p ec2db > ec2db\_backup.sql

Enter the root password when prompted. This creates a backup file ec2db\_backup.sql.

1. Verify the backup file exists:

bash

ls -lh ec2db\_backup.sql

**Restore Database (to RDS or target server)**

1. Connect to the target database (RDS or another MariaDB/MySQL instance):

bash

mysql -h <RDS-endpoint> -u <master-username> -p

Enter your master password.

1. Create a new database for restoring:

sql

**CREATE** **DATABASE** rdsdb;

**EXIT**;

1. Import the SQL dump file into the new database:

bash

mysql -h <RDS-endpoint> -u <master-username> -p rdsdb < ec2db\_backup.sql

Enter the password when prompted.

1. Verify the import by connecting and checking tables and data:

bash

mysql -h <RDS-endpoint> -u <master-username> -p

SHOW DATABASES;

USE rdsdb;

SHOW TABLES;

SELECT \* FROM <table\_name> LIMIT 10;

This process backs up the database from the EC2 MariaDB server and restores it into the RDS instance or any MySQL-compatible database system.

Task-on-RD

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**10) Create Read Replica**

To create a Read Replica for your MySQL or MariaDB RDS instance, follow these steps:

1. Open the AWS Management Console and navigate to the RDS service.
2. Select your existing primary RDS instance for which you want to create a read replica.
3. In the instance actions menu, choose **Create read replica**.
4. Configure the read replica instance:
   * Specify a unique identifier for the read replica.
   * Choose the instance class and storage options.
   * Select the desired availability zone.
   * Configure network and security group settings.
   * Enable any additional options like encryption, backups if needed.
5. Review the settings and click **Create read replica**.
6. AWS will provision the read replica, which asynchronously replicates data from the primary database to increase read scalability and availability.
7. When ready, the read replica can be used for read-heavy workloads and reporting without impacting the primary instance.

This process helps distribute database reads and improve overall application performance and availability.

A screenshot of a computer

AI-generated content may be incorrect.

**A screenshot of a computer

AI-generated content may be incorrect.**