aftab0045 / **AWS-Projects**

Code Issues Pull requests Actions Projects Wiki Security

AWS-Projects / **Database-Migration-to-AWS-RDS** /

aftab0045 add cfce9de · 4 minutes ago

AWS-Projects / **Database-Migration-to-AWS-RDS** / Top

Name	Name	Last commit date
..		
img	add	4 minutes ago
README.md	add	4 minutes ago

README.md

Database Migration to AWS RDS (MySQL)

Project Overview

This project demonstrates how to migrate a **traditional MySQL database** to **AWS RDS MySQL** using the **mysqldump backup and restore method**.
The migration is performed manually to understand the core concepts of database migration, networking, and security in AWS.

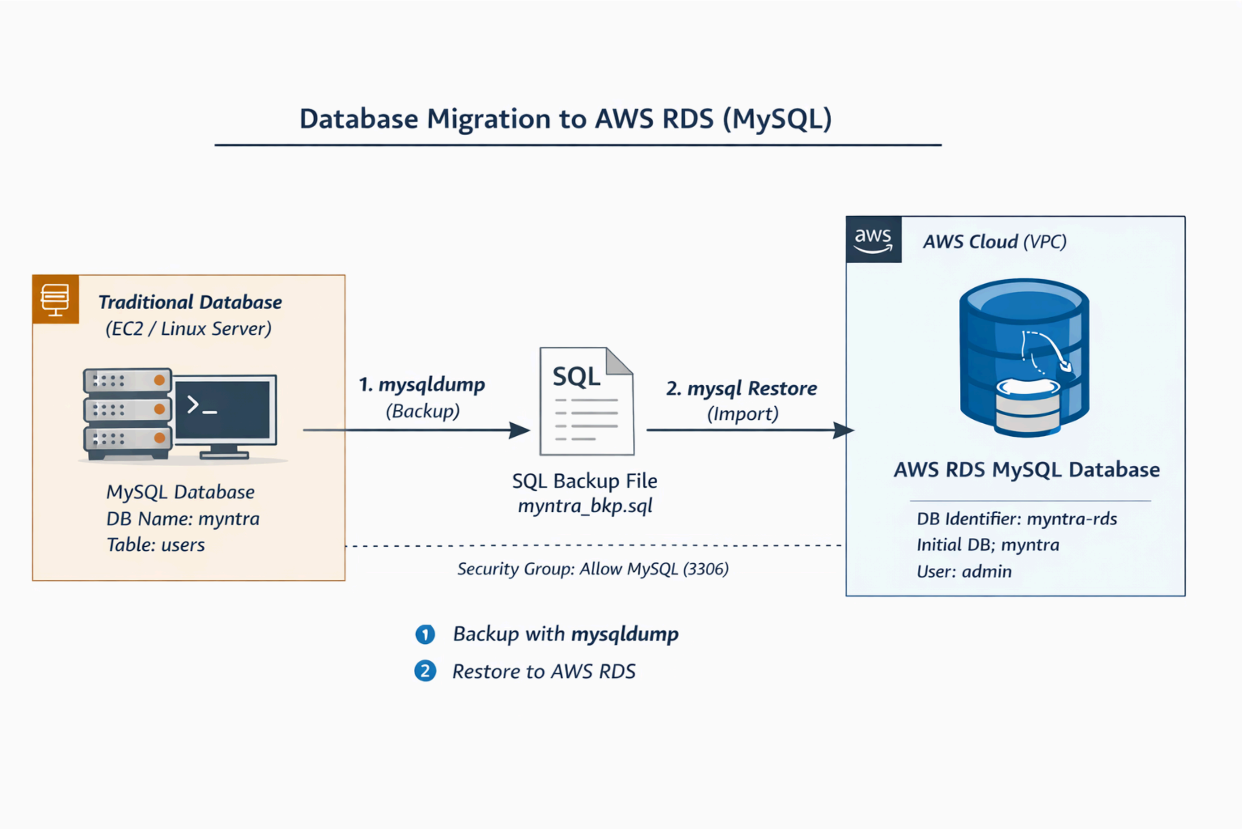
Project Objective

- Create a traditional MySQL database
- Take a logical backup using `mysqldump`
- Create an AWS RDS MySQL database
- Restore the backup into RDS
- Verify successful data migration

Tools & Technologies Used

- AWS EC2 (Traditional Database)
- AWS RDS (MySQL)
- MySQL
- mysqldump
- AWS Console
- Linux (Amazon Linux)

Architecture Diagram



Flow: Traditional MySQL → Backup (.sql) → AWS RDS MySQL

Step-by-Step Implementation

Step 1: Create Traditional MySQL Database

Login to MySQL on EC2 / Linux server:

```
sudo mysql -u root -p
```

Create database:

```
CREATE DATABASE myntra;
```

Use database:

```
USE myntra;
```

Create table:

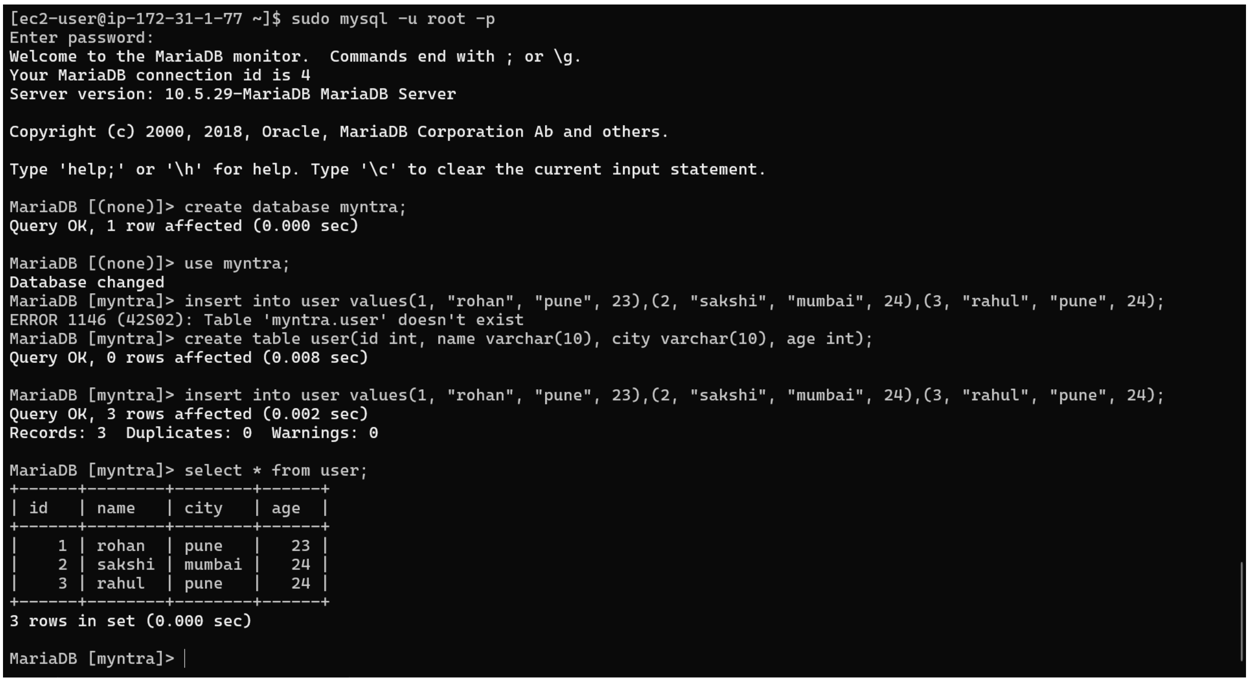
```
CREATE TABLE users (
  id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(50),
  city VARCHAR(10),
  age INT
);
```

Insert sample data:

```
INSERT INTO users VALUES
(1, "rohan", "pune", 23),
(2, "sakshi", "mumbai", 24),
(3, "rahul", "pune", 24)
;
```

Verify data:

```
SELECT * FROM users;
```



```
[ec2-user@ip-172-31-1-77 ~]$ 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)]> create database myntra;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> use myntra;
Database changed
MariaDB [myntra]> insert into user values(1, "rohan", "pune", 23),(2, "sakshi", "mumbai", 24),(3, "rahul", "pune", 24);
ERROR 1146 (42S02): Table 'myntra.user' doesn't exist
MariaDB [myntra]> create table user(id int, name varchar(10), city varchar(10), age int);
Query OK, 0 rows affected (0.008 sec)

MariaDB [myntra]> insert into user values(1, "rohan", "pune", 23),(2, "sakshi", "mumbai", 24),(3, "rahul", "pune", 24);
Query OK, 3 rows affected (0.002 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [myntra]> select * from user;
+------+--------+--------+------+
| id   | name   | city   | age  |
+------+--------+--------+------+
|    1 | rohan  | pune   |   23 |
|    2 | sakshi | mumbai |   24 |
|    3 | rahul  | pune   |   24 |
+------+--------+--------+------+
3 rows in set (0.000 sec)

MariaDB [myntra]>
```

Step 2: Take Backup Using mysqldump

Exit MySQL:

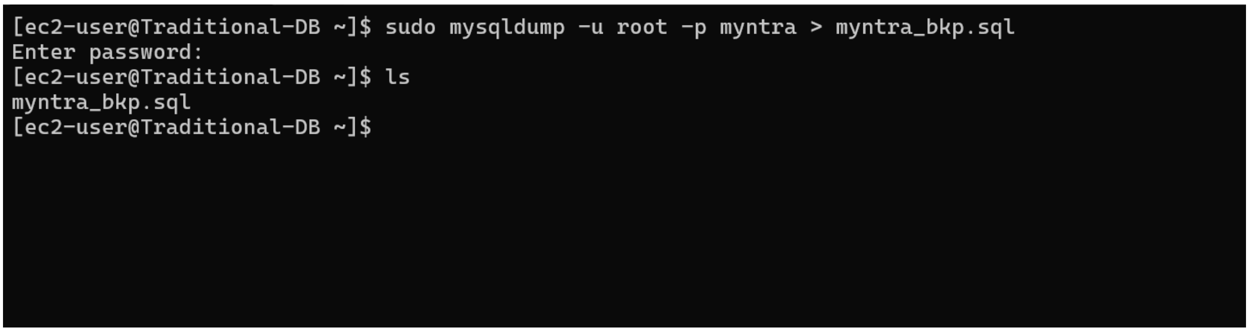
```
EXIT;
```

Take backup:

```
sudo mysqldump -u root -p myntra > myntra_bkp.sql
```

Verify backup file:

```
ls
```



```
[ec2-user@Traditional-DB ~]$ sudo mysqldump -u root -p myntra > myntra_bkp.sql
Enter password:
[ec2-user@Traditional-DB ~]$ ls
myntra_bkp.sql
[ec2-user@Traditional-DB ~]$
```

Step 3: Create AWS RDS MySQL Database

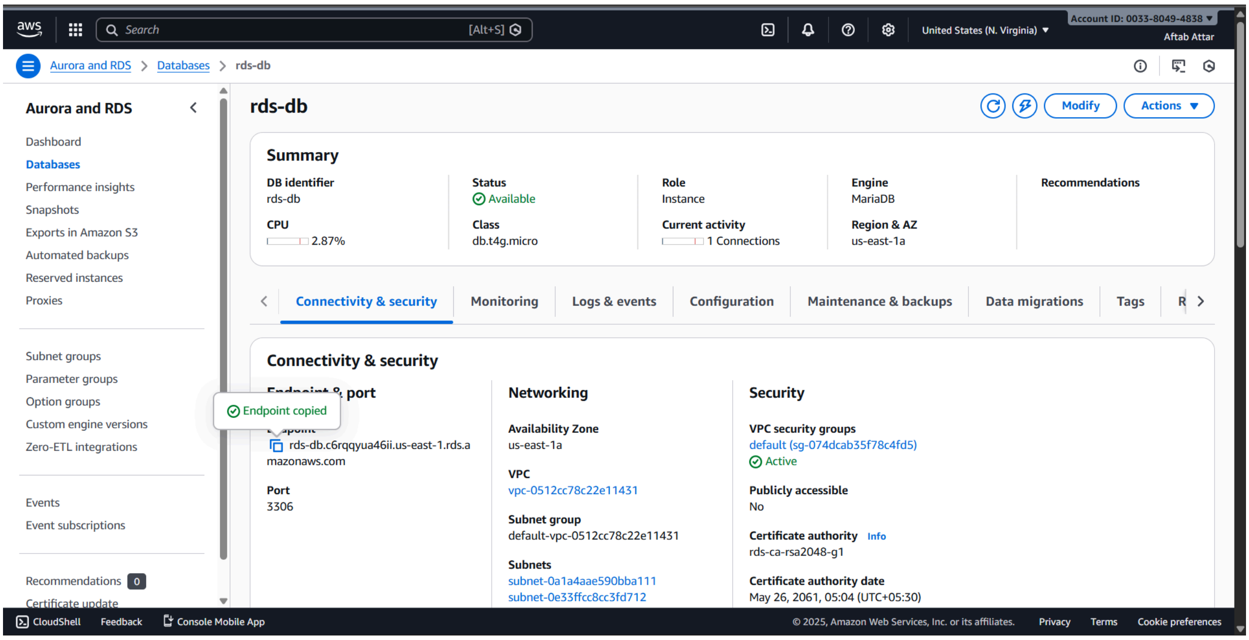
Go to **AWS Console → RDS → Create Database**

Configuration:

- Engine: **MySQL**
- Template: **Free Tier**
- DB Identifier: `myntra-rds`
- Username: `admin`
- Password: `********`

Additional Configuration:

- Initial database name: `myntra`
- Initial database name set to `myntra`



rds-db

Summary

DB identifier
rds-db

CPU
2.87%

Status
✓ Available

Class
db.t4g.micro

Role
Instance

Current activity
1 Connections

Engine
MariaDB

Region & AZ
us-east-1a

Recommendations

Connectivity & security

Endpoint & port

Endpoint
rds-db.c6rqqyua46ii.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1a

VPC
vpc-0512cc78c22e11431

Subnet group
default-vpc-0512cc78c22e11431

Subnets
subnet-0a1a4aae590bba111
subnet-0e33ffcc8cc3fd712

Security

VPC security groups
default (sg-074dcab35f78c4fd5)
✓ Active

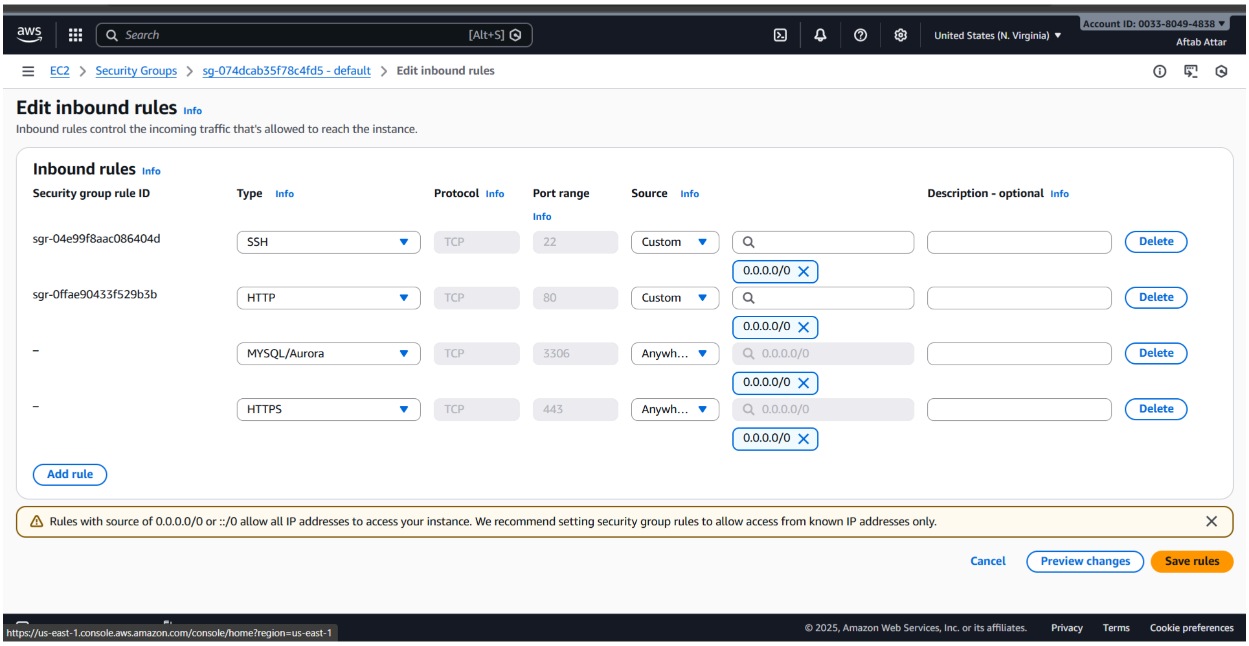
Publicly accessible
No

Certificate authority Info
rds-ca-rsa2048-g1

Certificate authority date
May 26, 2061, 05:04 (UTC+05:30)

Step 4: Configure RDS Connectivity & Security

- Public access: **No**
- VPC: **Same as EC2**
- Security Group:
  - Allow **MySQL (3306)** from **EC2 Security Group**
- RDS inbound rule allowing port **3306**
- RDS endpoint visible



aws
Search [Alt+S]
United States (N. Virginia) ▾
Account ID: 0033-8049-4838 ▾
Aftab Attar
EC2 > Security Groups > sg-074dcab35f78c4fd5 - default > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

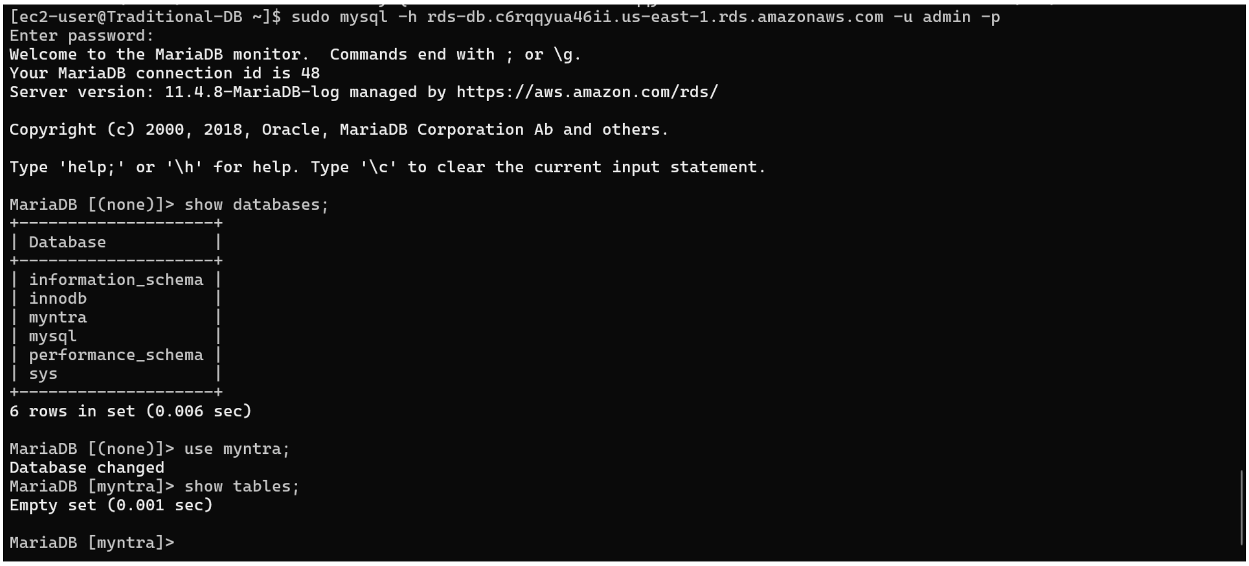
Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-04e99f8aac086404d	SSH	TCP	22	Custom	0.0.0.0/0		Delete
sgr-0ffae90433f529b3b	HTTP	TCP	80	Custom	0.0.0.0/0		Delete
–	MYSQL/Aurora	TCP	3306	Anywh...	0.0.0.0/0 0.0.0.0/0		Delete
–	HTTPS	TCP	443	Anywh...	0.0.0.0/0 0.0.0.0/0		Delete

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel
Preview changes
Save rules
https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Step 5: Connect to RDS MySQL

```
sudo mysql -h <RDS-ENDPOINT> -u admin -p
```



```
[ec2-user@Traditional-DB ~]$ sudo mysql -h rds-db.c6rqqyua46ii.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 48
Server version: 11.4.8-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.

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| innodb             |
| myntra             |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.006 sec)

MariaDB [(none)]> use myntra;
Database changed
MariaDB [myntra]> show tables;
Empty set (0.001 sec)

MariaDB [myntra]>
```

Step 6: Restore Backup into RDS:

Exit MySQL:

```
EXIT;
```

Restore database:

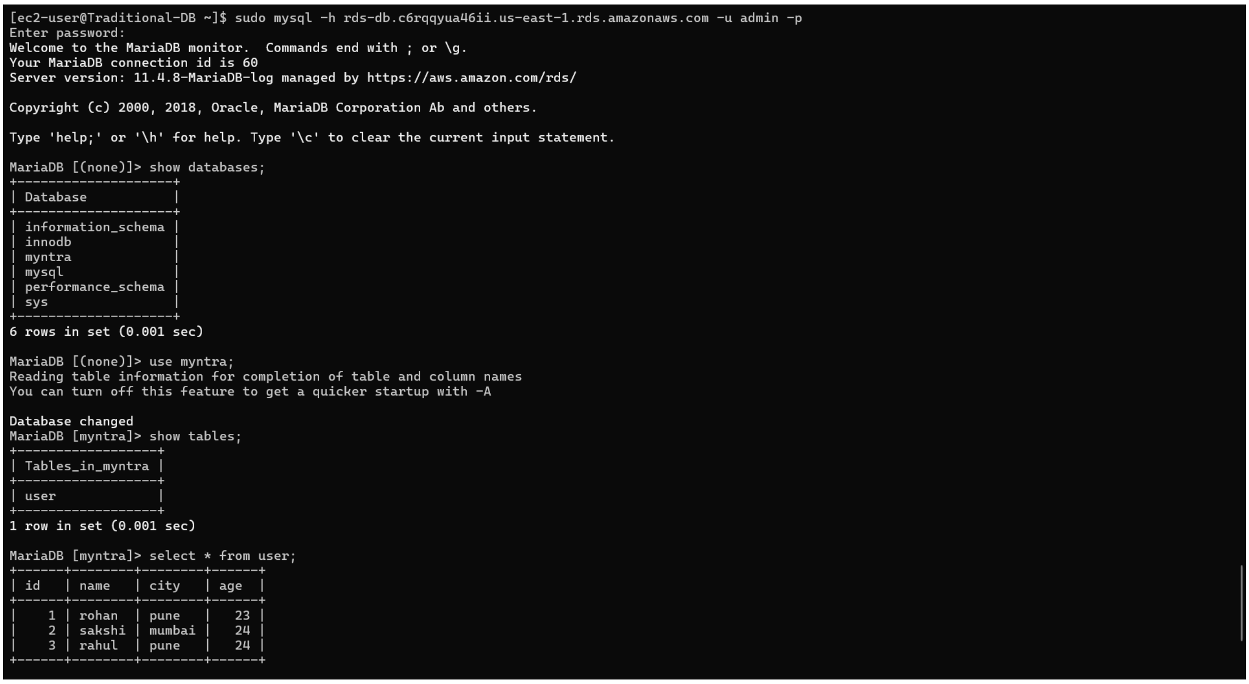
```
sudo mysql -h <RDS-ENDPOINT> -u admin -p myntra < myntra_bkp.sql
```

Step 7: Verify Data Migration Login again:

```
sudo mysql -h <RDS-ENDPOINT> -u admin -p
```

Verify data:

```
USE myntra;
SELECT * FROM users;
```



```
[ec2-user@Traditional-DB ~]$ sudo mysql -h rds-db.c6rqqyua46ii.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 60
Server version: 11.4.8-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.

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| innodb             |
| myntra             |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.001 sec)

MariaDB [(none)]> use myntra;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [myntra]> show tables;
+------------------+
| Tables_in_myntra |
+------------------+
| user             |
+------------------+
1 row in set (0.001 sec)

MariaDB [myntra]> select * from user;
+------+--------+--------+------+
| id   | name   | city   | age  |
+------+--------+--------+------+
|    1 | rohan  | pune   |   23 |
|    2 | sakshi | mumbai |   24 |
|    3 | rahul  | pune   |   24 |
+------+--------+--------+------+
```

Tip

> If the MySQL connection hangs, ensure that the RDS security group allows inbound MySQL (3306) traffic from the EC2 security group.

Conclusion

This project demonstrates a manual MySQL database migration to AWS RDS, a core skill for cloud developers. The mysqldump method is simple, reliable, and perfect for small databases or learning purposes. Using AWS RDS provides scalability, reliability, and automated backups, eliminating the overhead of managing your own database server.