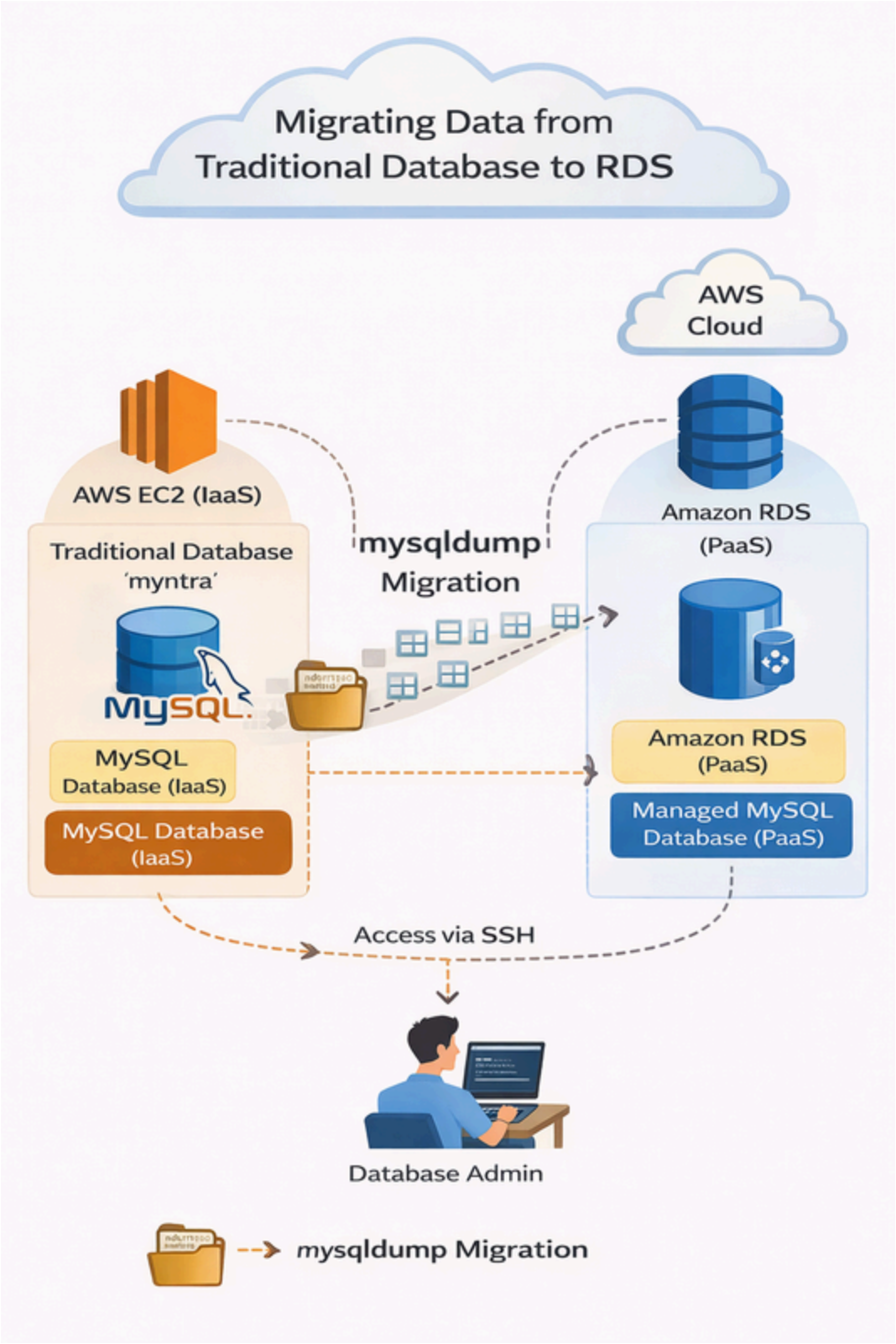
Migration of Traditional Database to AWS RDS (IaaS to PaaS)

Introduction

This project demonstrates the migration of data from a **traditional database hosted on an AWS EC2 instance (IaaS)** to **Amazon RDS (PaaS)**. The objective of the project is to understand how organizations can move from self-managed databases to managed database services provided by AWS for better scalability, availability, and reduced operational overhead.

In this project, a traditional database named `myntra` is hosted on an EC2 instance and migrated to **Amazon RDS**, showcasing the transition from **Infrastructure as a Service (IaaS)** to **Platform as a Service (PaaS)**.



Project Overview

- Traditional database hosted on **EC2 instance**
- Database name: **myntra**

- Target database: **Amazon RDS**
- Migration performed from **IaaS → PaaS**
- Data successfully transferred from EC2 database to RDS

Technologies and AWS Services Used

- **Amazon EC2** – Hosting traditional database (IaaS)
- **Amazon RDS** – Managed relational database service (PaaS)
- **MySQL / MariaDB** – Database engine
- **Amazon Linux** – Operating system
- **AWS Security Groups** – Network security
- **mysqldump** – Database migration tool
- **SSH** – Secure server access

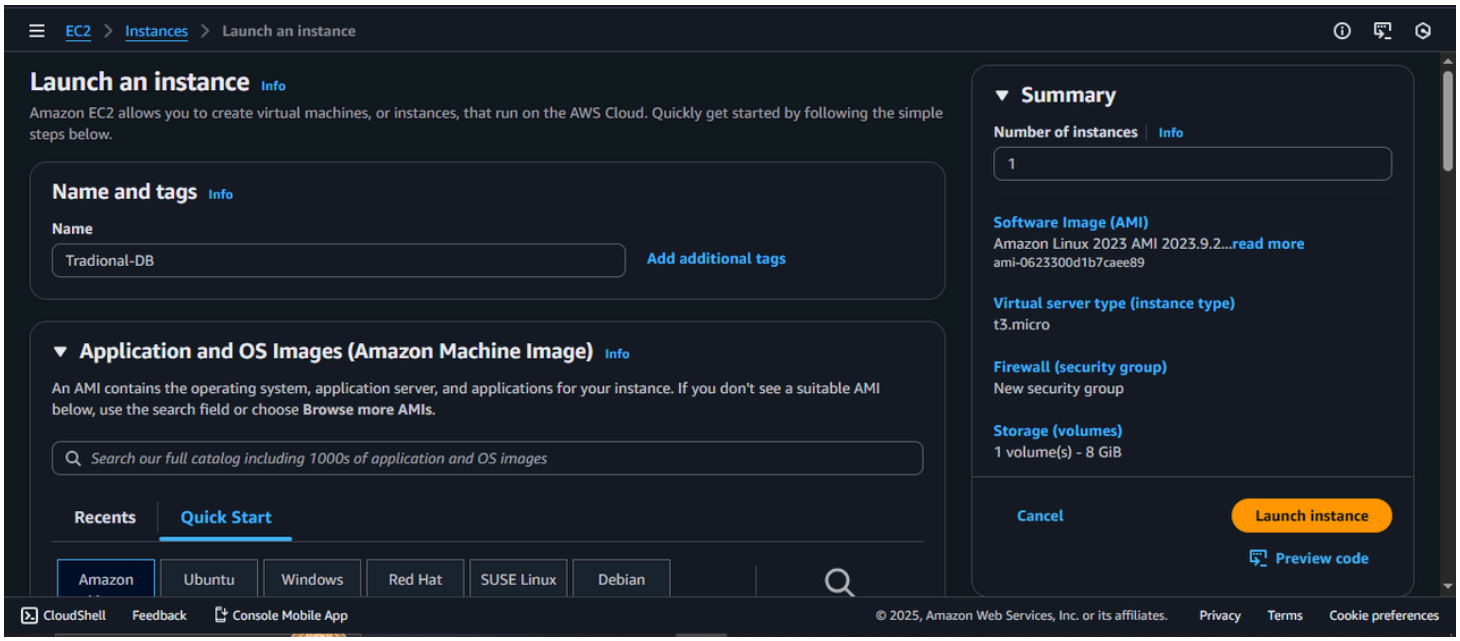
Project Architecture

1. Traditional database **myntra** runs on an EC2 instance.
2. Amazon RDS instance is created with the same database engine.
3. Database data is exported from EC2 using dump utilities.
4. Exported data is imported into Amazon RDS.
5. Application can now connect to RDS instead of EC2 database.

Step-by-Step Project Implementation

Step 1: Launch EC2 Instance for Traditional Database

- Launch an EC2 instance using Amazon Linux.
- Install database server (MySQL/MariaDB).
- Create database named **myntra**.




```
sudo yum update -y
sudo yum install mariadb-server -y
sudo systemctl start mariadb
sudo systemctl enable mariadb
```

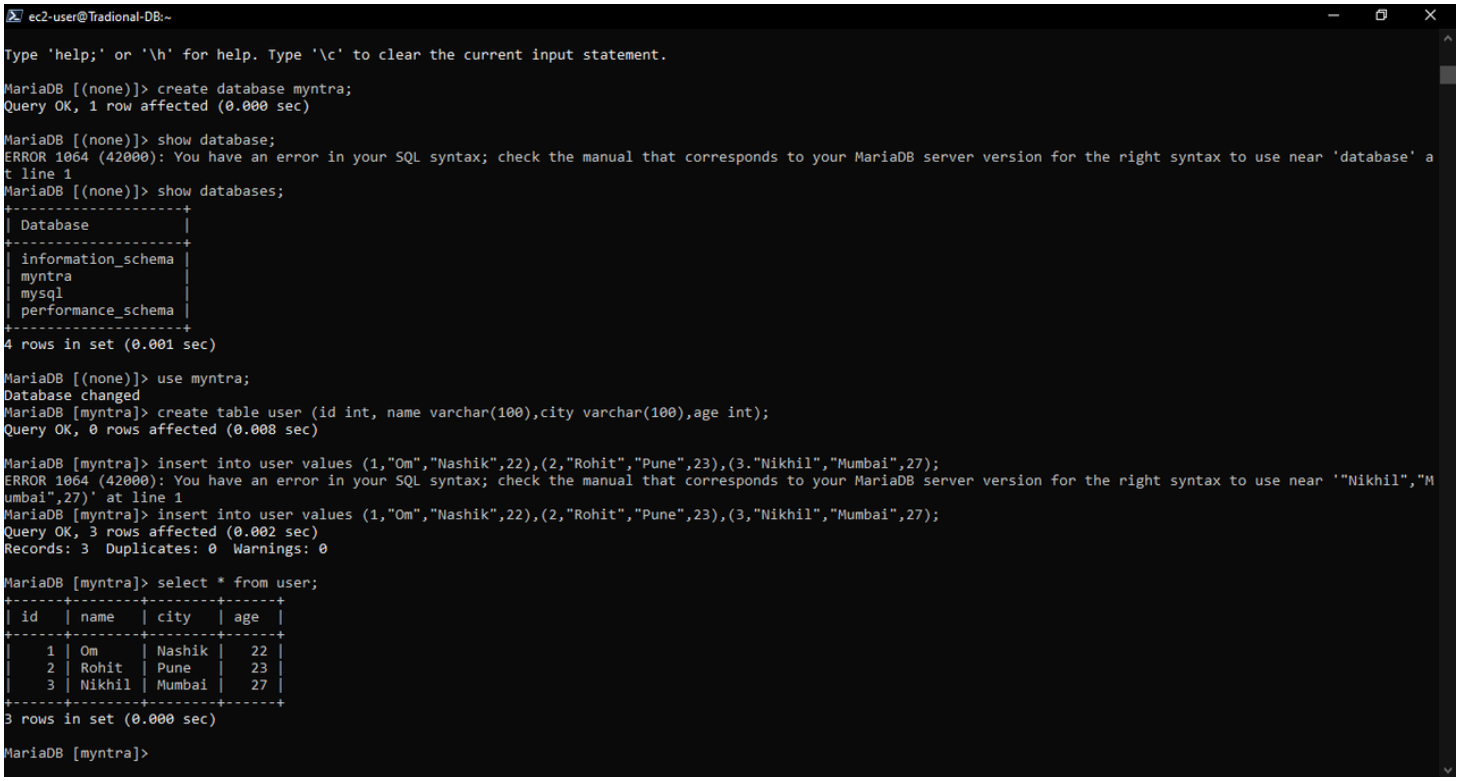
Create database:

```
CREATE DATABASE myntra;
```

Step 2: Insert Sample Data

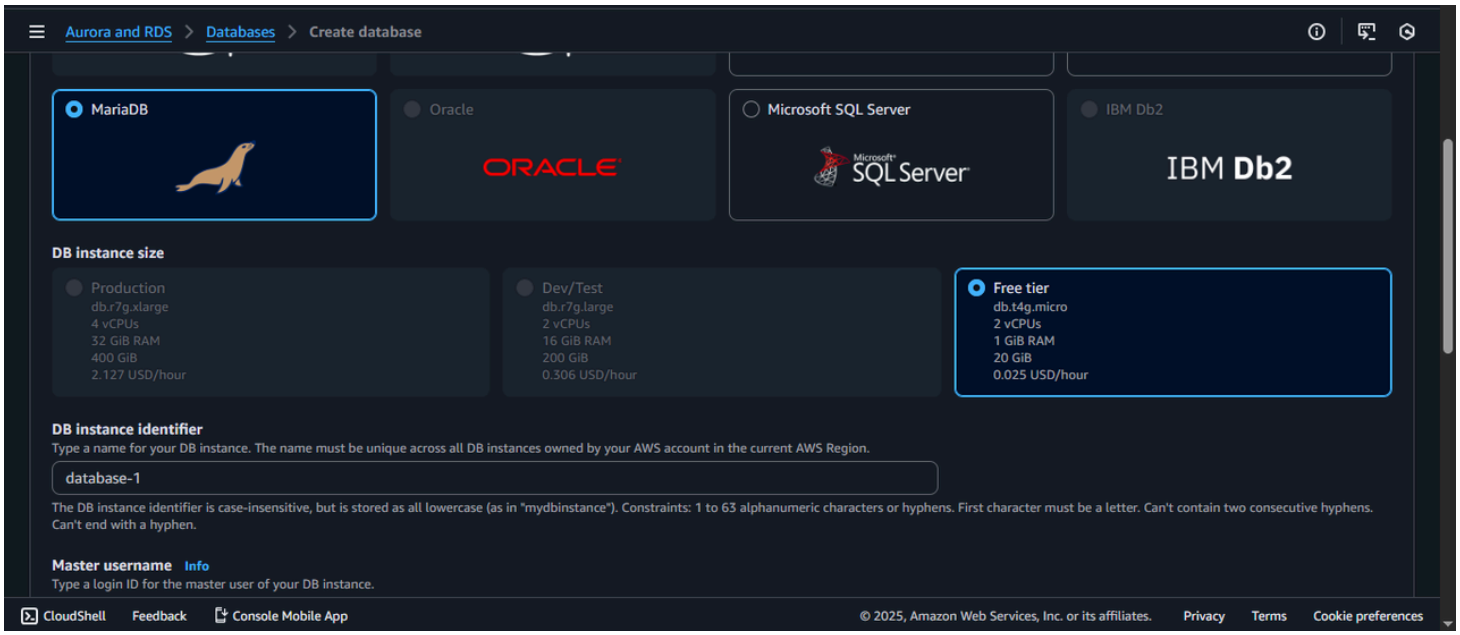
```
USE myntra;
create table user (id int, name varchar(100),city varchar(100),age int);

insert into user values (1,"Om","Nashik",22);
insert into user values(2,"Rohit","Pune",23);
insert into user values(3,"Nikhil","Mumbai",27);
```



Step 3: Create Amazon RDS Instance

- Go to **RDS Dashboard**.
- Create a new RDS instance (MySQL/MariaDB).
- Configure:
  - DB instance identifier
  - Username and password
  - VPC and security group
- Enable inbound access from EC2 security group.



Step 4: Export Database from EC2 (IaaS)

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

Step 5: Import Data into Amazon RDS (PaaS)

```
mysql -h <RDS-ENDPOINT> -u <username> -p myntra < myntra.sql>
sudo mysql -h rds-db.c16w68u6ormo.us-west-1.rds.amazonaws.com -u admin -p
```

Step 6: Verify Migration

- Login to RDS database.
- Verify tables and data.

```
USE myntra;
SELECT * FROM user;
```

Key Differences: IaaS vs PaaS

Feature	Traditional DB (EC2)	Amazon RDS
Server Management	Manual	AWS Managed
Backups	Manual	Automatic
Scaling	Manual	Automated
Patching	Manual	AWS Managed
Availability	User Managed	Built-in

Benefits of Migrating to Amazon RDS

- Reduced administrative overhead
- Automatic backups and patching
- High availability and fault tolerance
- Easy scalability
- Improved security

Conclusion

This project successfully demonstrates the migration of a traditional database named **myntra** from an **EC2 instance (IaaS)** to **Amazon RDS (PaaS)**. The migration highlights how organizations can modernize their infrastructure by adopting managed database services to improve reliability, scalability, and operational efficiency.

This approach is widely adopted in real-world cloud environments and forms a crucial step in cloud migration strategies.

Migrated from IaaS to PaaS using AWS EC2 and Amazon RDS ☁🚀