Migration of Databse form EC2 to RDS(Iaas to Paas)

📓 Overview

This project demonstrates how to migrate an existing database hosted on an Amazon EC2 instance (IaaS) to Amazon RDS (PaaS). The goal is to move from a self-managed infrastructure to a managed database service to improve scalability, reliability, and operational efficiency.

🎯 Objectives

• Understand the difference between IaaS (EC2) and PaaS (RDS).

• Perform a database migration from EC2 to RDS securely.

• Validate the migration and ensure minimal downtime.

• Optimize performance and manage backups in RDS.

🧰 Prerequisites

• AWS Account

• Existing EC2 instance with database (MySQL or PostgreSQL)

• AWS RDS instance created

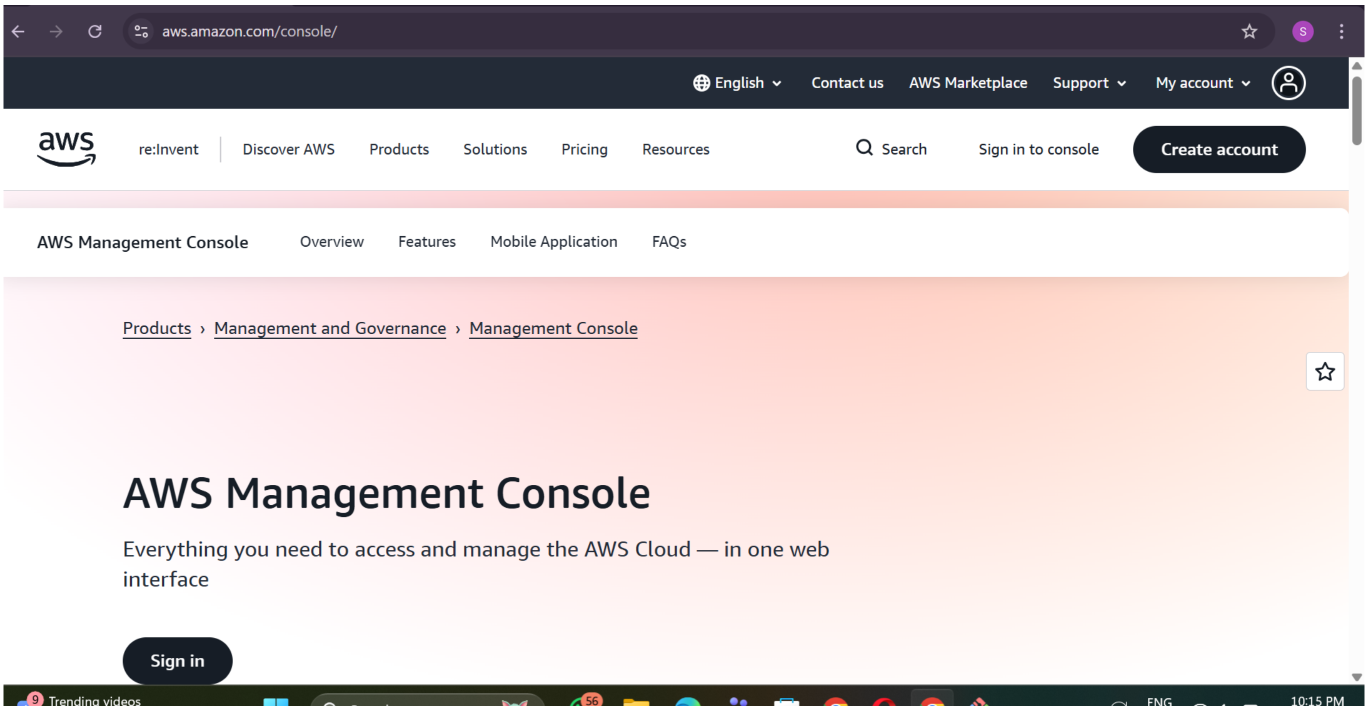
• Proper IAM Role and Security Group configurations

• AWS CLI or AWS Management Console access

⚙️ Steps

Step-1: Login to AWS Management Console

1. Go to https://aws.amazon.com/console/.

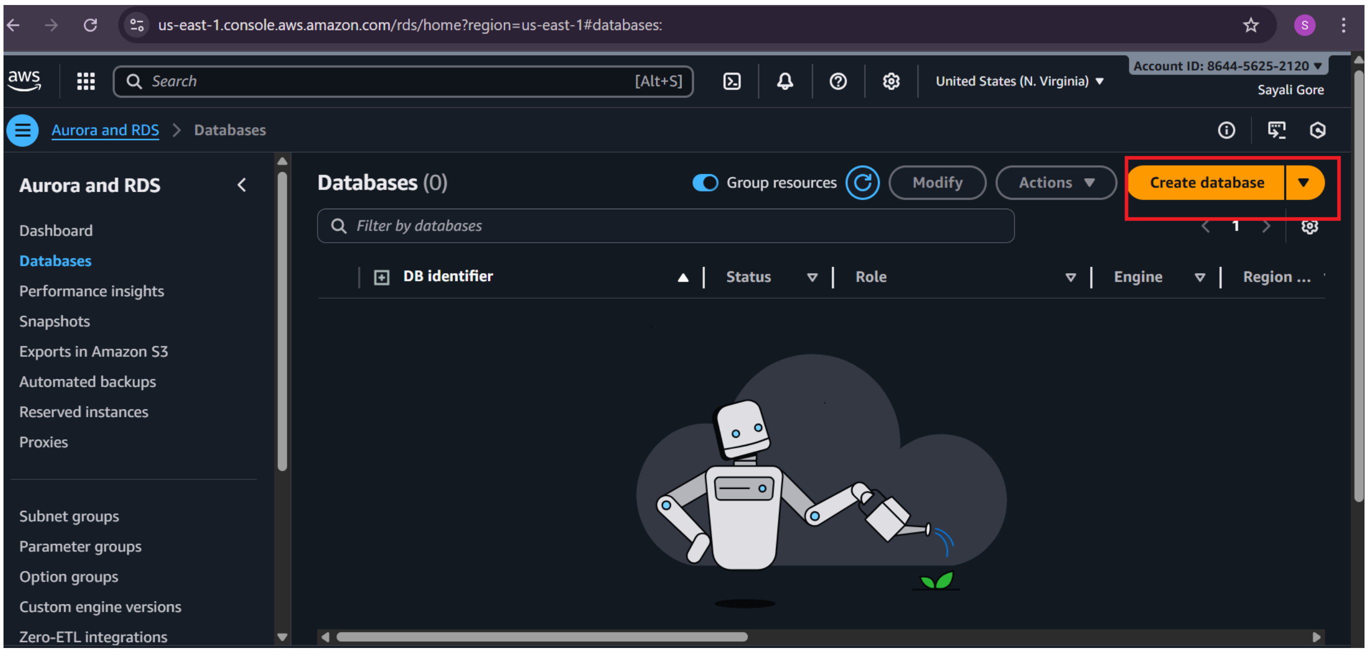


AWS Management Console

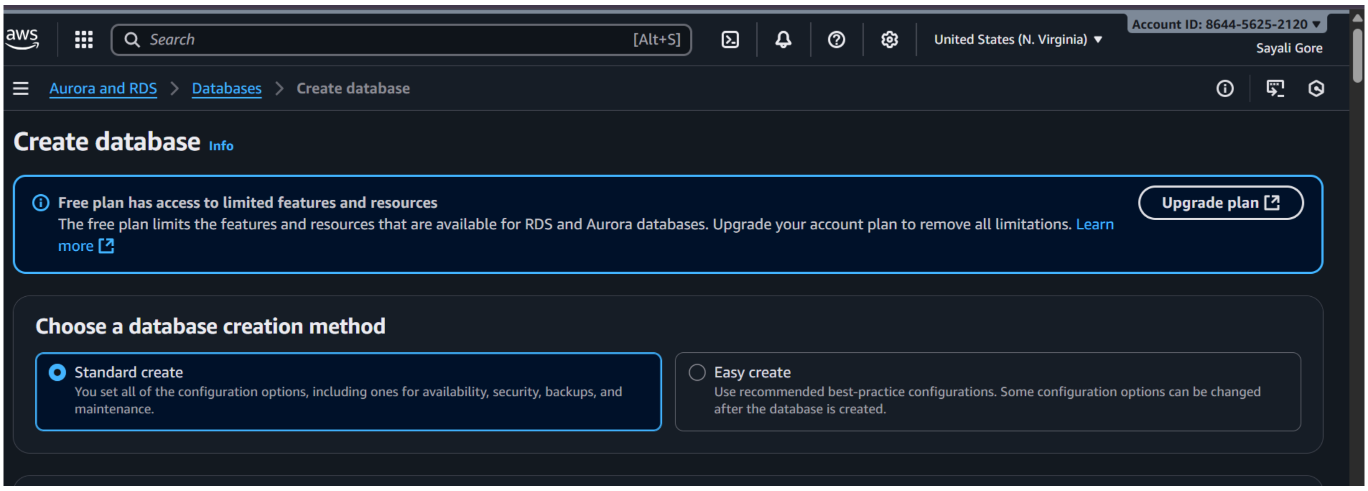
Everything you need to access and manage the AWS Cloud — in one web interface

Step-2:rCeate RDS

1.click create database.



2. choose a database creation method.



Account ID: 8644-5625-2120
Sayali Gore
United States (N. Virginia)

Aurora and RDS > Databases > Create database

Create database Info

Free plan has access to limited features and resources
The free plan limits the features and resources that are available for RDS and Aurora databases. Upgrade your account plan to remove all limitations. Learn more

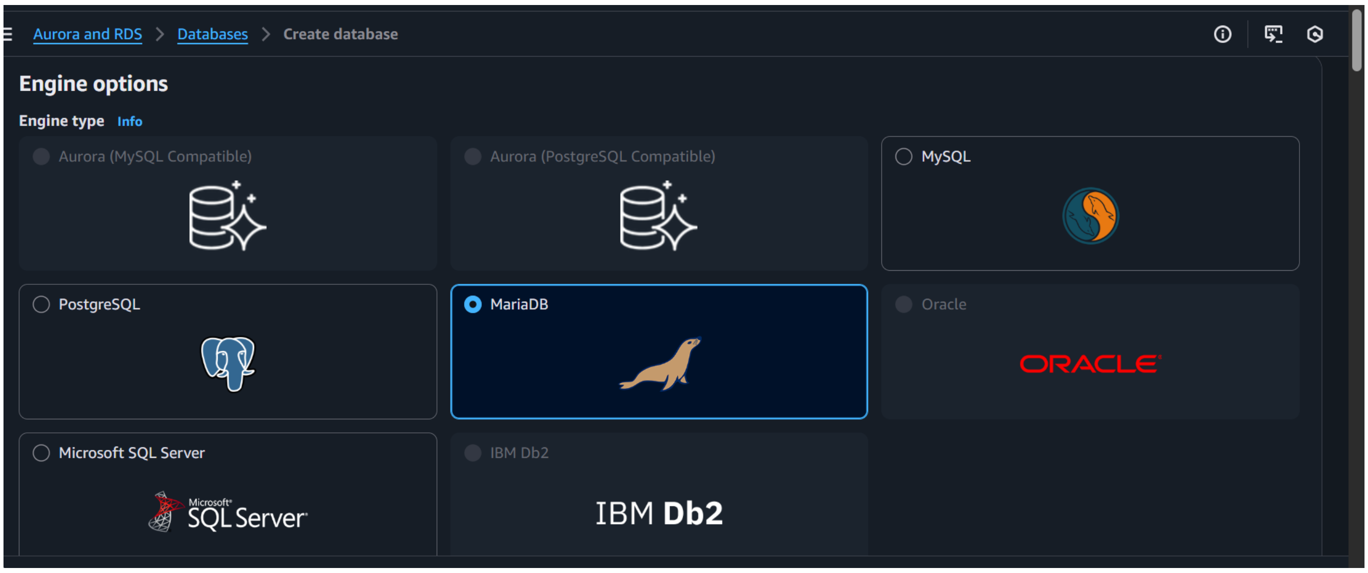
Upgrade plan

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.

3. Select Database Engine.



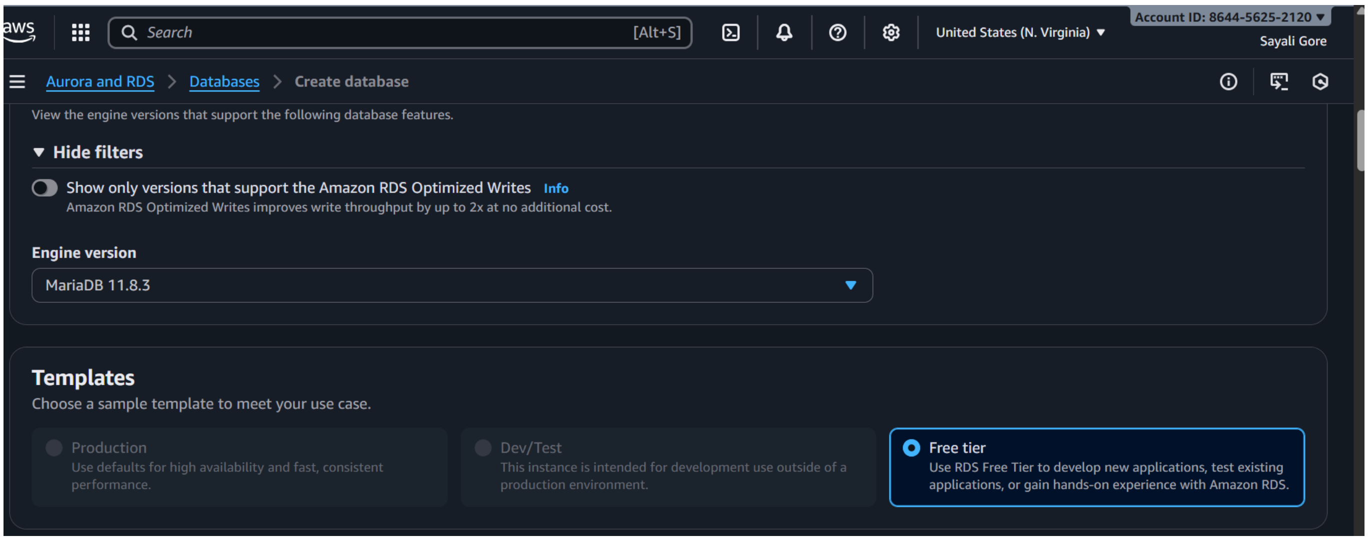
Aurora and RDS > Databases > Create database

Engine options

Engine type Info

Aurora (MySQL Compatible) | Aurora (PostgreSQL Compatible) | MySQL | PostgreSQL | [✓ MariaDB] | Oracle | Microsoft SQL Server | IBM Db2

4. Choose engine version and Template.



Account ID: 8644-5625-2120
Sayali Gore
United States (N. Virginia)

Aurora and RDS > Databases > Create database

View the engine versions that support the following database features.

Hide filters

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: MariaDB 11.8.3

Templates
Choose a sample template to meet your use case.

Production — Use defaults for high availability and fast, consistent performance.

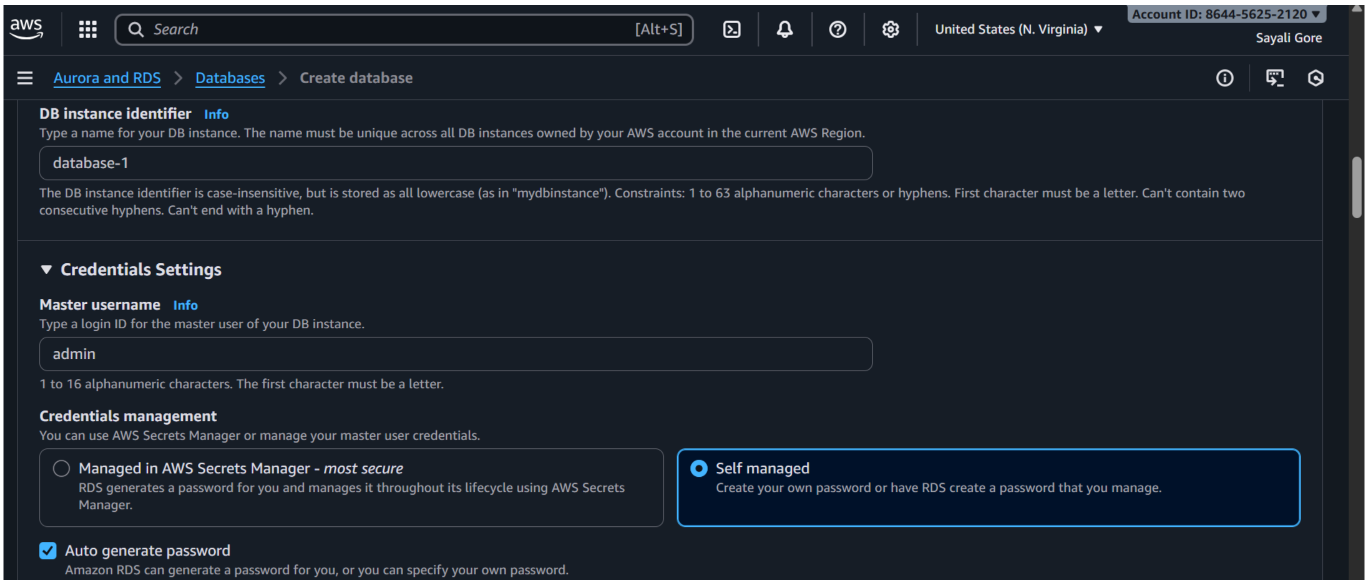
Dev/Test — This instance is intended for development use outside of a production environment.

[✓ Free tier] Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.

5.DB instance identifier: give your database a name

6.Master username: e.g., admin

7.Master password: Auto Genarate Password.


aws
Search [Alt+S]
United States (N. Virginia)
Account ID: 8644-5625-2120
Sayali Gore
Aurora and RDS > Databases > Create database

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.
database-1
The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). 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.

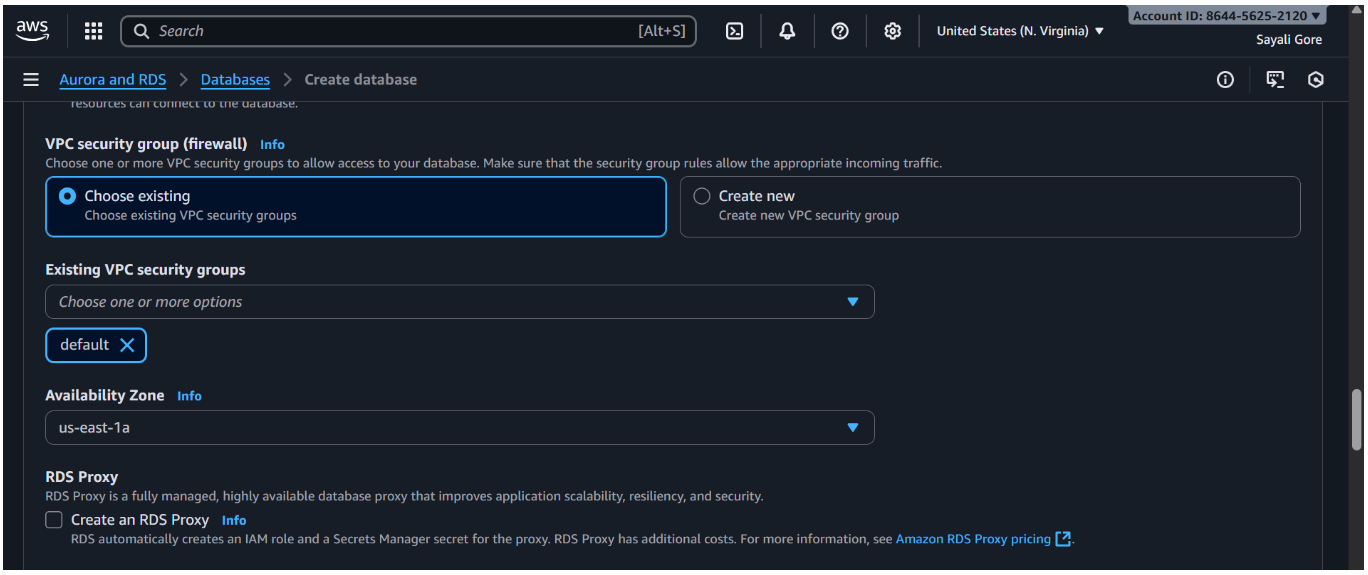
Credentials Settings

Master username Info
Type a login ID for the master user of your DB instance.
admin
1 to 16 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.

8.Choose your VPC


aws
Search [Alt+S]
United States (N. Virginia)
Account ID: 8644-5625-2120
Sayali Gore
Aurora and RDS > Databases > Create database

resources can connect to the database.

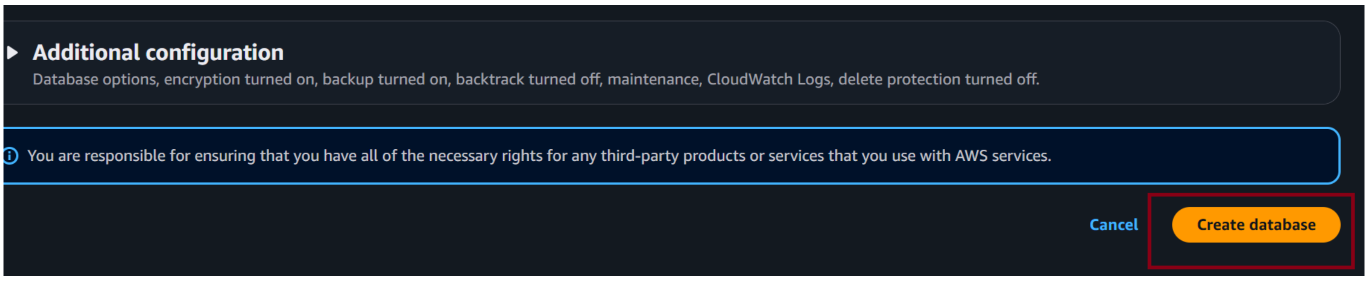
VPC security group (firewall) Info
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.
- [✓ Choose existing] — Choose existing VPC security groups
- Create new — Create new VPC security group

Existing VPC security groups
Choose one or more options
default

Availability Zone Info
us-east-1a

RDS Proxy
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.
[ ] Create an RDS Proxy Info
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see Amazon RDS Proxy pricing.

10. Click Create database

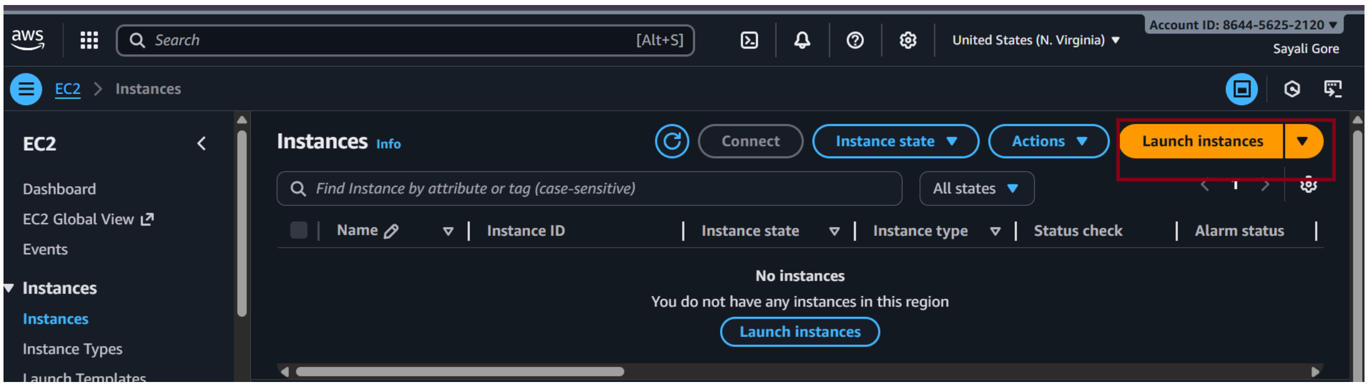


Additional configuration
Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

Step-3:lunch Ec2 Instance

1.Click the “Launch instance” button.

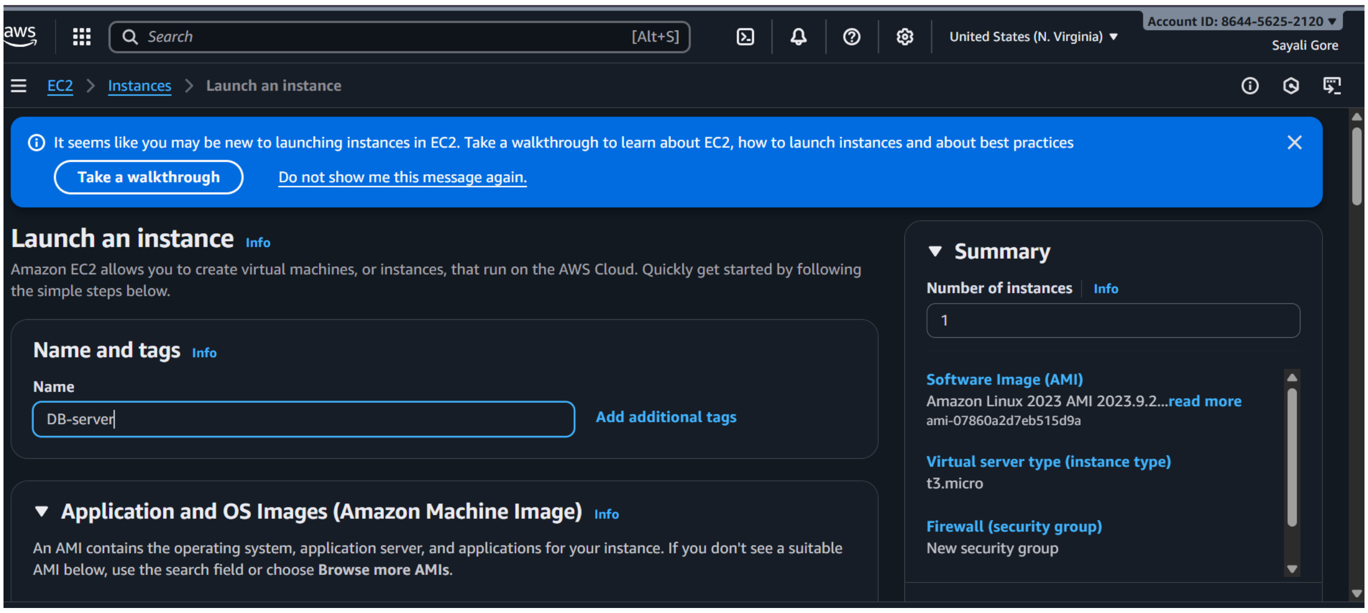


Instances Info

Find Instance by attribute or tag (case-sensitive)

No instances
You do not have any instances in this region

2. Name and Tags


EC2 > Instances > Launch an instance
Take a walkthrough
Do not show me this message again.
Add additional tags

It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: DB-server

Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose **Browse more AMIs**.

Summary

Number of instances: 1

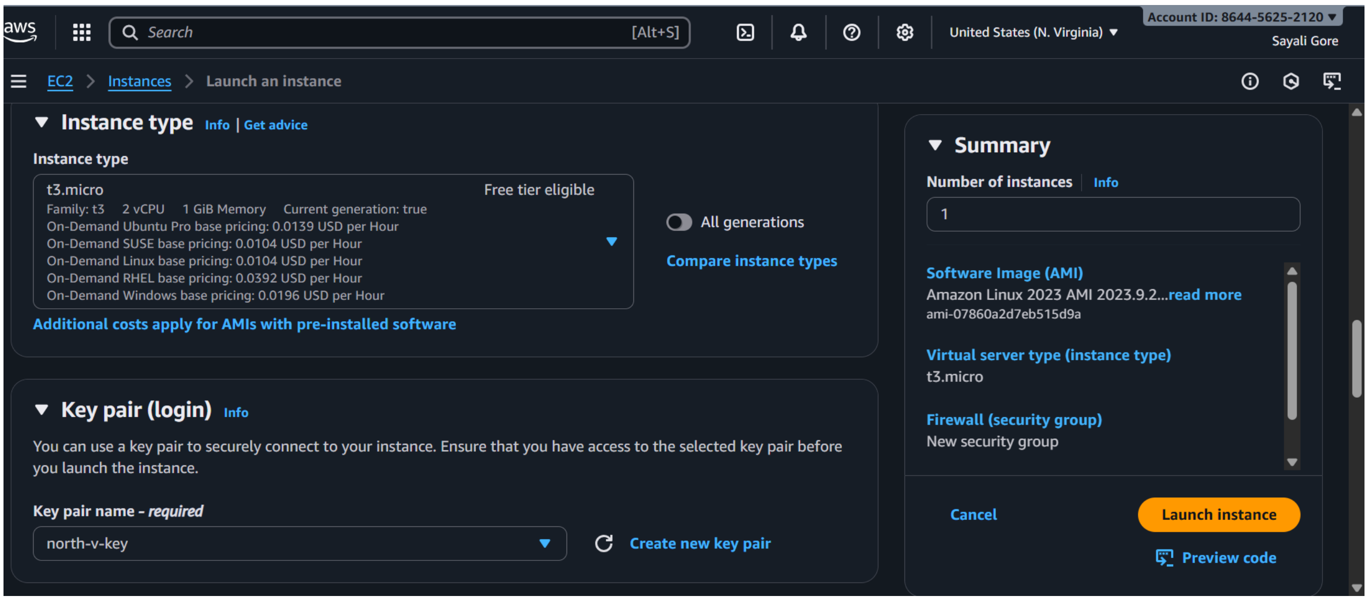
Software Image (AMI): Amazon Linux 2023 AMI 2023.9.2...read more
ami-07860a2d7eb515d9a

Virtual server type (instance type): t3.micro

Firewall (security group): New security group

3. Istance types.

4. Configure Key Pair



Instance type Info | Get advice

Instance type

```
t3.micro                                          Free tier eligible
Family: t3    2 vCPU    1 GiB Memory    Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour
On-Demand SUSE base pricing: 0.0104 USD per Hour
On-Demand Linux base pricing: 0.0104 USD per Hour
On-Demand RHEL base pricing: 0.0392 USD per Hour
On-Demand Windows base pricing: 0.0196 USD per Hour
```

Additional costs apply for AMIs with pre-installed software

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*: north-v-key

Summary

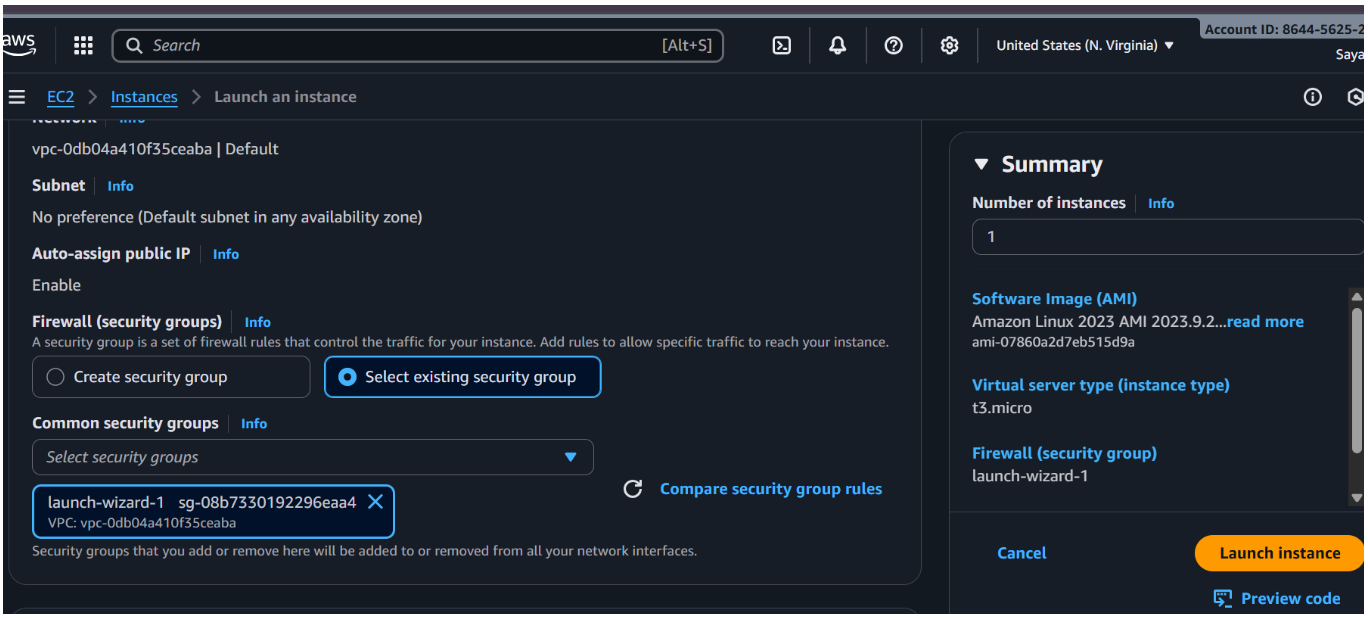
Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.9.2...read more
ami-07860a2d7eb515d9a

Virtual server type (instance type): t3.micro

Firewall (security group): New security group

5. Add a Security Group rule



vpc-0db04a410f35ceaba | Default

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group [✓ Select existing security group]

Common security groups Info

Select security groups

launch-wizard-1 sg-08b7330192296eaa4
VPC: vpc-0db04a410f35ceaba

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Summary

Number of instances Info: 1

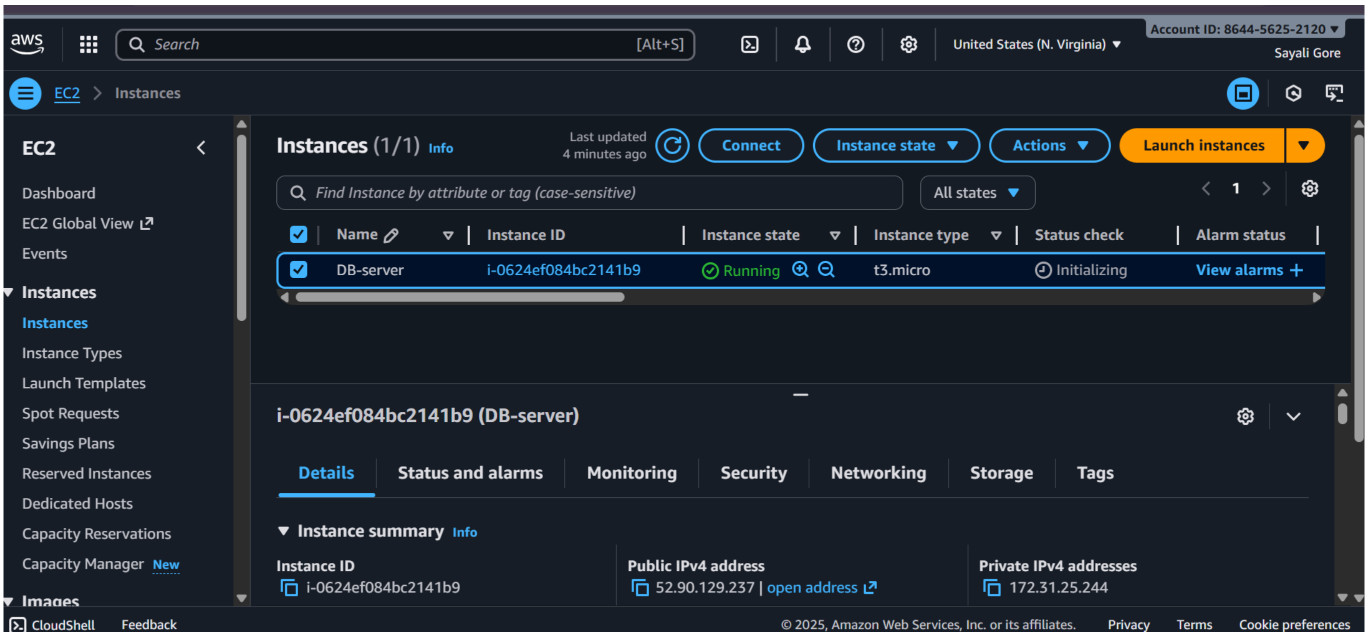
Software Image (AMI): Amazon Linux 2023 AMI 2023.9.2...read more
ami-07860a2d7eb515d9a

Virtual server type (instance type): t3.micro

Firewall (security group): launch-wizard-1

Cancel | Launch instance | Preview code

6. Review and Launch



Instances (1/1) Info

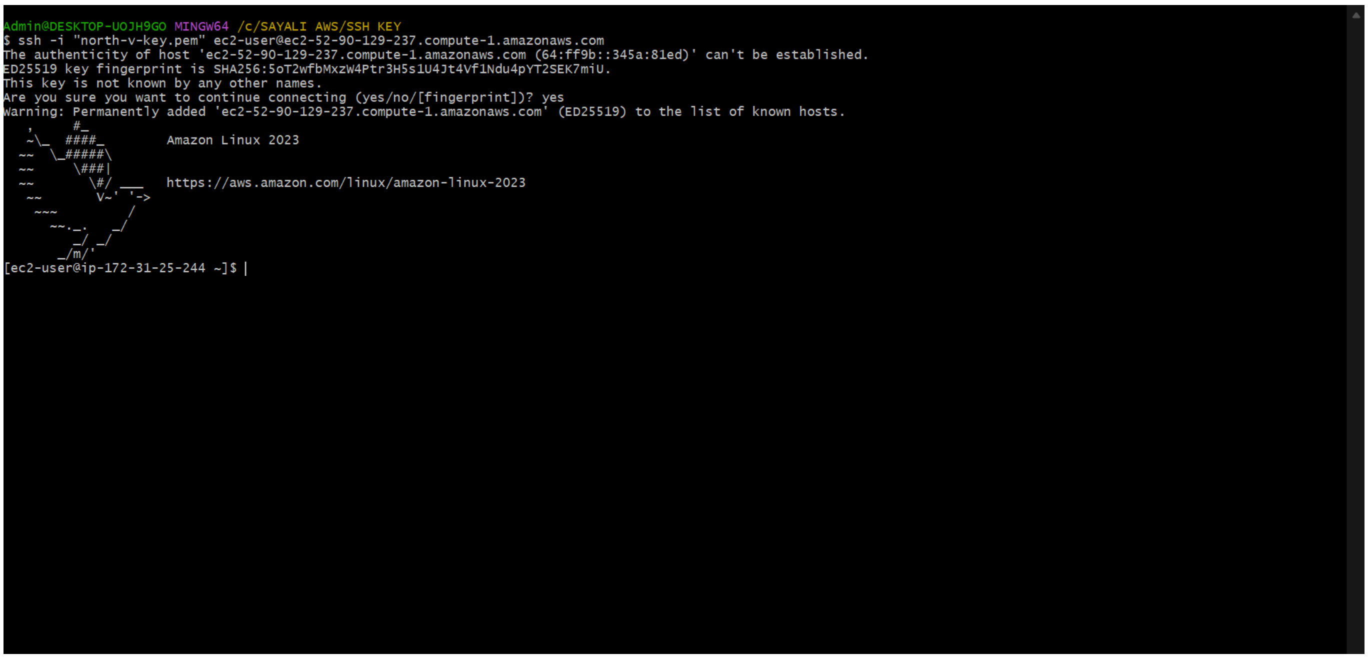
Name	Instance ID	Instance state	Instance type	Status check	Alarm status
DB-server	i-0624ef084bc2141b9	Running	t3.micro	Initializing	View alarms

i-0624ef084bc2141b9 (DB-server)

Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0624ef084bc2141b9	52.90.129.237 | open address	172.31.25.244

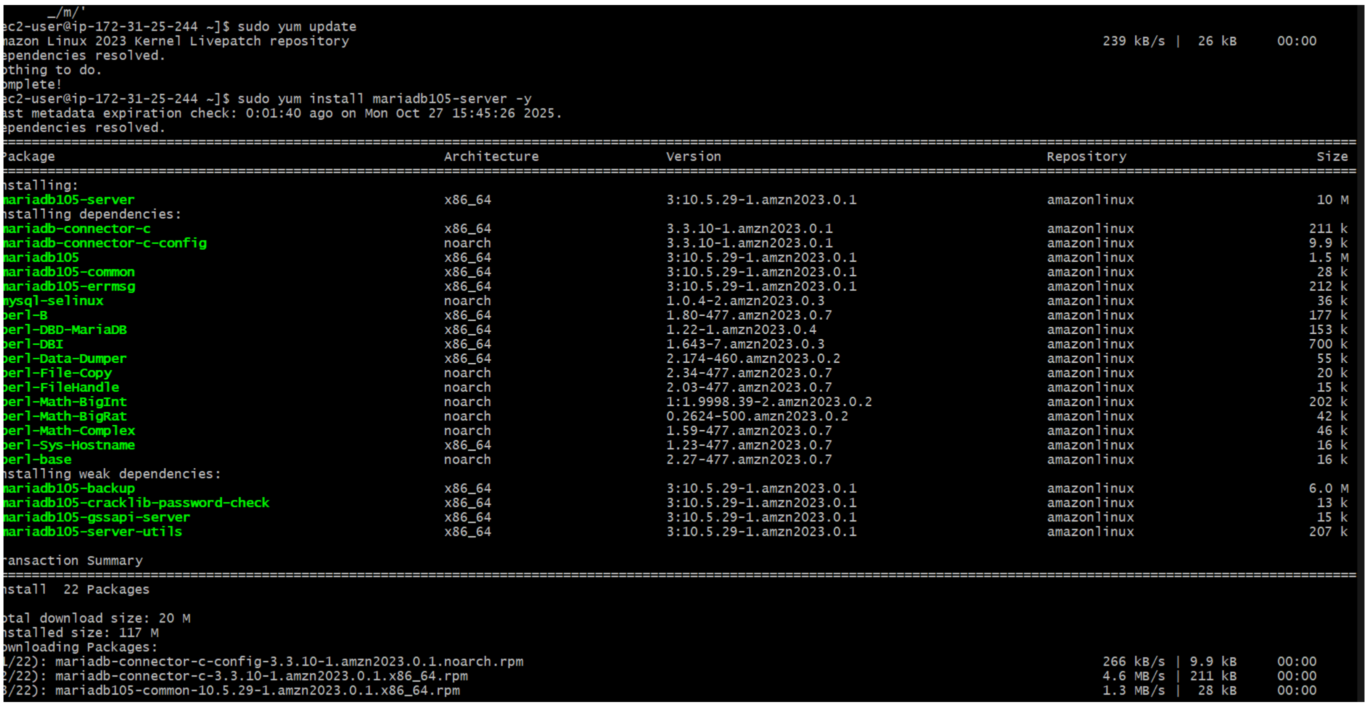
Step-4:connecting to EC2 Instance Terminal.



```
Admin@DESKTOP-UOJH9GO MINGW64 /c/SAYALI AWS/SSH KEY
$ ssh -i "north-v-key.pem" ec2-user@ec2-52-90-129-237.compute-1.amazonaws.com
The authenticity of host 'ec2-52-90-129-237.compute-1.amazonaws.com (64:ff9b::345a:81ed)' can't be established.
ED25519 key fingerprint is SHA256:5oT2wfbMxzW4Ptr3H5s1U4Jt4Vf1Ndu4pYT2SEK7miU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-90-129-237.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-25-244 ~]$ |
```

Step-5: Install MariaDB Srever.

```
sudo yum update
sudo yum install mariadb105-Server
```

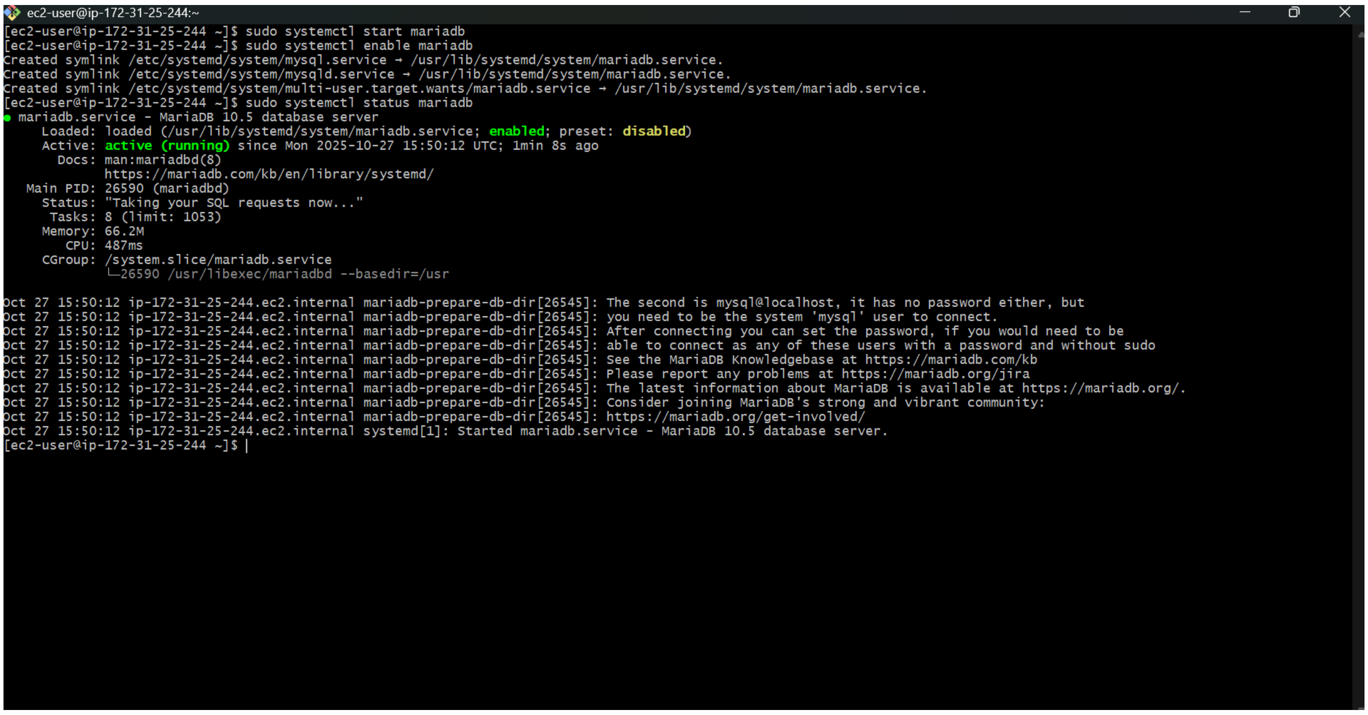


```
       _/m/'
ec2-user@ip-172-31-25-244 ~]$ sudo yum update
mazon Linux 2023 Kernel Livepatch repository                                                          239 kB/s |  26 kB     00:00
ependencies resolved.
othing to do.
omplete!
ec2-user@ip-172-31-25-244 ~]$ sudo yum install mariadb105-server -y
ast metadata expiration check: 0:01:40 ago on Mon Oct 27 15:45:26 2025.
ependencies resolved.
==================================================================================================================================
Package                                 Architecture        Version                               Repository                 Size
==================================================================================================================================
nstalling:
mariadb105-server                       x86_64              3:10.5.29-1.amzn2023.0.1              amazonlinux                10 M
nstalling dependencies:
mariadb-connector-c                     x86_64              3.3.10-1.amzn2023.0.1                 amazonlinux               211 k
mariadb-connector-c-config              noarch              3.3.10-1.amzn2023.0.1                 amazonlinux               9.9 k
mariadb105                              x86_64              3:10.5.29-1.amzn2023.0.1              amazonlinux               1.5 M
mariadb105-common                       x86_64              3:10.5.29-1.amzn2023.0.1              amazonlinux                28 k
mariadb105-errmsg                       x86_64              3:10.5.29-1.amzn2023.0.1              amazonlinux               212 k
mysql-selinux                           noarch              1.0.4-2.amzn2023.0.3                  amazonlinux                36 k
perl-B                                  x86_64              1.80-477.amzn2023.0.7                 amazonlinux               177 k
perl-DBD-MariaDB                        x86_64              1.22-1.amzn2023.0.4                   amazonlinux               153 k
perl-DBI                                x86_64              1.643-7.amzn2023.0.3                  amazonlinux               700 k
perl-Data-Dumper                        x86_64              2.174-460.amzn2023.0.2                amazonlinux                55 k
perl-File-Copy                          noarch              2.34-477.amzn2023.0.7                 amazonlinux                20 k
perl-FileHandle                         noarch              2.03-477.amzn2023.0.7                 amazonlinux                15 k
perl-Math-BigInt                        noarch              1:1.9998.39-2.amzn2023.0.2            amazonlinux               202 k
perl-Math-BigRat                        noarch              0.2624-500.amzn2023.0.2               amazonlinux                42 k
perl-Math-Complex                       noarch              1.59-477.amzn2023.0.7                 amazonlinux                46 k
perl-Sys-Hostname                       x86_64              1.23-477.amzn2023.0.7                 amazonlinux                16 k
perl-base                               noarch              2.27-477.amzn2023.0.7                 amazonlinux                16 k
nstalling weak dependencies:
mariadb105-backup                       x86_64              3:10.5.29-1.amzn2023.0.1              amazonlinux               6.0 M
mariadb105-cracklib-password-check      x86_64              3:10.5.29-1.amzn2023.0.1              amazonlinux                13 k
mariadb105-gssapi-server                x86_64              3:10.5.29-1.amzn2023.0.1              amazonlinux                15 k
mariadb105-server-utils                 x86_64              3:10.5.29-1.amzn2023.0.1              amazonlinux               207 k

ransaction Summary
==================================================================================================================================
nstall  22 Packages

otal download size: 20 M
nstalled size: 117 M
ownloading Packages:
1/22): mariadb-connector-c-config-3.3.10-1.amzn2023.0.1.noarch.rpm                                    266 kB/s | 9.9 kB     00:00
2/22): mariadb-connector-c-3.3.10-1.amzn2023.0.1.x86_64.rpm                                           4.6 MB/s | 211 kB     00:00
3/22): mariadb105-common-10.5.29-1.amzn2023.0.1.x86_64.rpm                                            1.3 MB/s |  28 kB     00:00
```

1.Start, Enable Status MariaDB Service.



```
ec2-user@ip-172-31-25-244:~
[ec2-user@ip-172-31-25-244 ~]$ sudo systemctl start mariadb
[ec2-user@ip-172-31-25-244 ~]$ sudo systemctl enable mariadb
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
[ec2-user@ip-172-31-25-244 ~]$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled)
     Active: active (running) since Mon 2025-10-27 15:50:12 UTC; 1min 8s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 26590 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 8 (limit: 1053)
     Memory: 66.2M
        CPU: 487ms
     CGroup: /system.slice/mariadb.service
             └─26590 /usr/libexec/mariadbd --basedir=/usr

Oct 27 15:50:12 ip-172-31-25-244.ec2.internal mariadb-prepare-db-dir[26545]: The second is mysql@localhost, it has no password either, but
Oct 27 15:50:12 ip-172-31-25-244.ec2.internal mariadb-prepare-db-dir[26545]: you need to be the system 'mysql' user to connect.
Oct 27 15:50:12 ip-172-31-25-244.ec2.internal mariadb-prepare-db-dir[26545]: After connecting you can set the password, if you would need to be
Oct 27 15:50:12 ip-172-31-25-244.ec2.internal mariadb-prepare-db-dir[26545]: able to connect as any of these users with a password and without sudo
Oct 27 15:50:12 ip-172-31-25-244.ec2.internal mariadb-prepare-db-dir[26545]: See the MariaDB Knowledgebase at https://mariadb.com/kb
Oct 27 15:50:12 ip-172-31-25-244.ec2.internal mariadb-prepare-db-dir[26545]: Please report any problems at https://mariadb.org/jira
Oct 27 15:50:12 ip-172-31-25-244.ec2.internal mariadb-prepare-db-dir[26545]: The latest information about MariaDB is available at https://mariadb.org/.
Oct 27 15:50:12 ip-172-31-25-244.ec2.internal mariadb-prepare-db-dir[26545]: Consider joining MariaDB's strong and vibrant community:
Oct 27 15:50:12 ip-172-31-25-244.ec2.internal mariadb-prepare-db-dir[26545]: https://mariadb.org/get-involved/
Oct 27 15:50:12 ip-172-31-25-244.ec2.internal systemd[1]: Started mariadb.service - MariaDB 10.5 database server.
[ec2-user@ip-172-31-25-244 ~]$ |
```

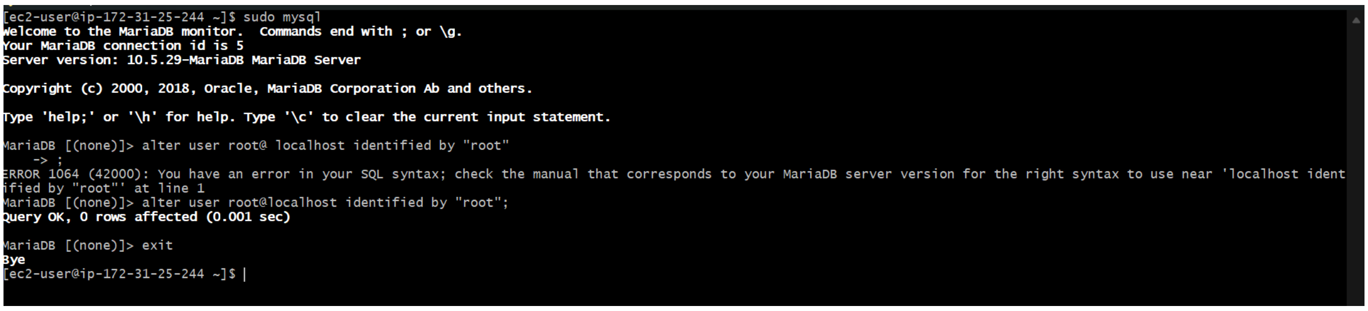
Step-6:Login to MySQL:

```
sudo mysql
```

1.Set root password:

```
 ALTER USER 'root'@'localhost' IDENTIFIED BY 'root';

  Exit MySQL:
```



```
[ec2-user@ip-172-31-25-244 ~]$ sudo mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5
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)]> alter user root@ localhost identified by "root"
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'localhost ident
ified by "root"' at line 1
MariaDB [(none)]> alter user root@localhost identified by "root";
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> exit
Bye
[ec2-user@ip-172-31-25-244 ~]$ |
```

Step-7: MySQL Login again with password

1. Login with Password

```
 Sudo mysql -u root -p
```

2. Create Databse

```
Create database myntra;

Use Myntra;
```

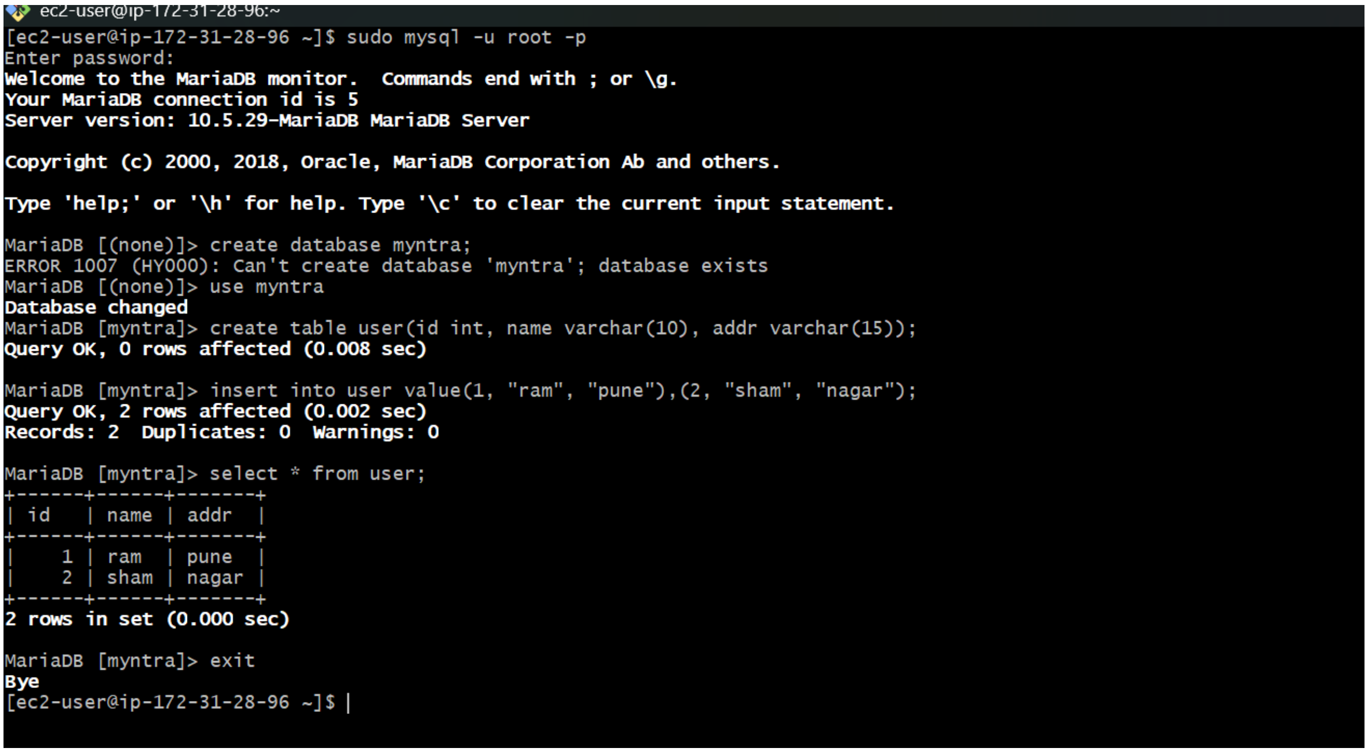
3.Create table and insert data

```
create teble user(id int, name varchar(10),Addr varchar(15));

insert into user values(1,"Ram", "pune"),(2, "Sham", "Nagar");

Select * from user;

exit;
```



```
ec2-user@ip-172-31-28-96:~
[ec2-user@ip-172-31-28-96 ~]$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5
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;
ERROR 1007 (HY000): Can't create database 'myntra'; database exists
MariaDB [(none)]> use myntra
Database changed
MariaDB [myntra]> create table user(id int, name varchar(10), addr varchar(15));
Query OK, 0 rows affected (0.008 sec)

MariaDB [myntra]> insert into user value(1, "ram", "pune"),(2, "sham", "nagar");
Query OK, 2 rows affected (0.002 sec)
Records: 2  Duplicates: 0  Warnings: 0

MariaDB [myntra]> select * from user;
+------+------+-------+
| id   | name | addr  |
+------+------+-------+
|    1 | ram  | pune  |
|    2 | sham | nagar |
+------+------+-------+
2 rows in set (0.000 sec)

MariaDB [myntra]> exit
Bye
[ec2-user@ip-172-31-28-96 ~]$ |
```

Step-8 Take a database backup

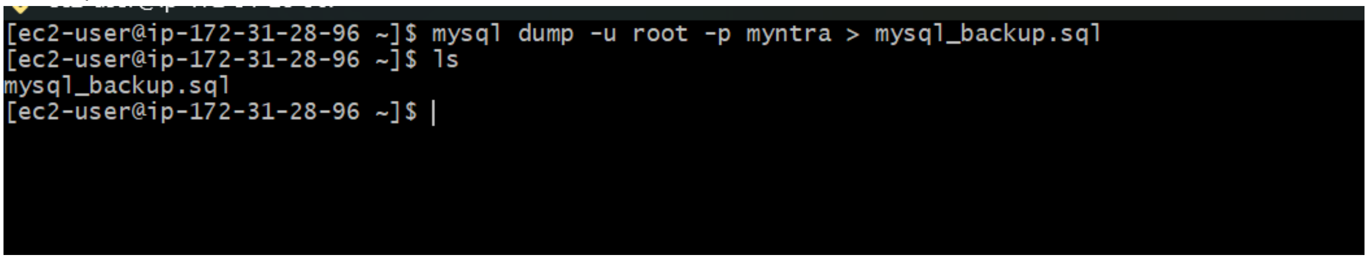
```
Mysqldump -u root -p myntra > mysql_backup.sql
```

(Enter password : root)

Step-9 Connect to RDS Instence

```
Sudo mysql -h <endpoint> -u admin -p
```

(Enter password :)



```
[ec2-user@ip-172-31-28-96 ~]$ mysql dump -u root -p myntra > mysql_backup.sql
[ec2-user@ip-172-31-28-96 ~]$ ls
mysql_backup.sql
[ec2-user@ip-172-31-28-96 ~]$ |
```

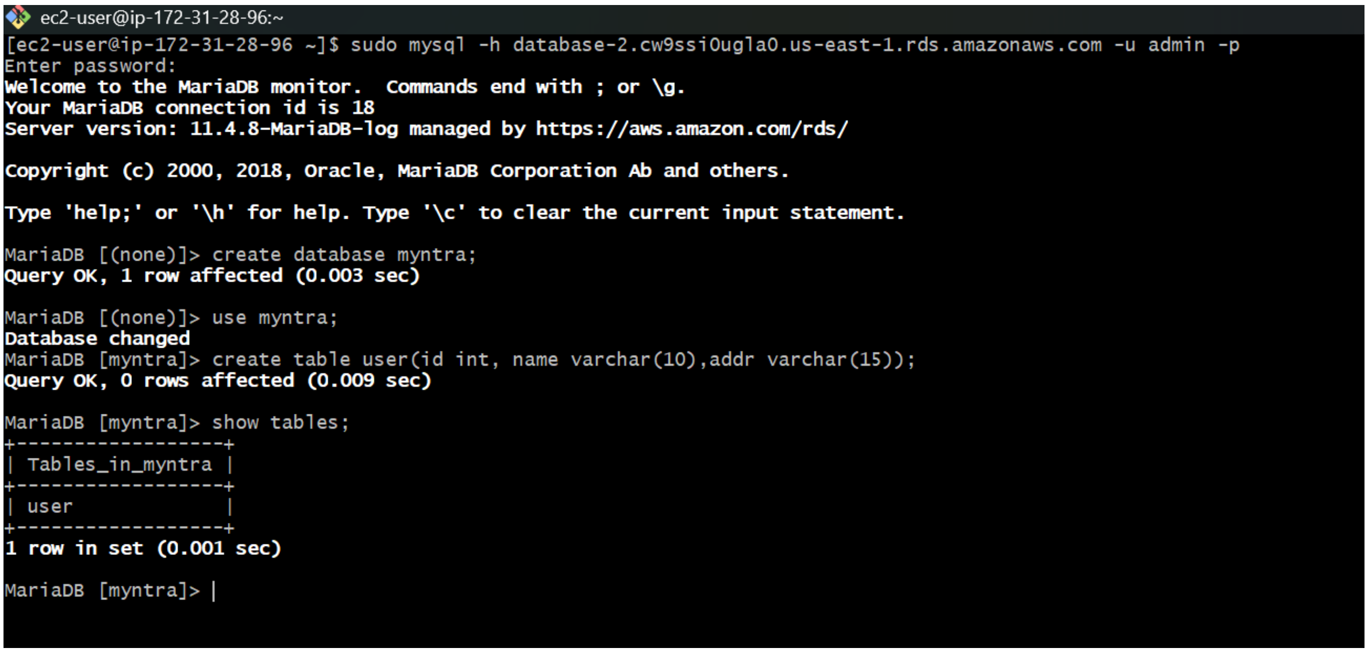
Step-10 Create database and table in RDS:

```
CREATE DATABASE myntra;

USE myntra;

CREATE TABLE user(id int, name varchar(10), Addr varchar(15));

Show table;
```



```
ec2-user@ip-172-31-28-96:~
[ec2-user@ip-172-31-28-96 ~]$ sudo mysql -h database-2.cw9ssi0ugla0.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 18
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)]> create database myntra;
Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]> use myntra;
Database changed
MariaDB [myntra]> create table user(id int, name varchar(10),addr varchar(15));
Query OK, 0 rows affected (0.009 sec)

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

MariaDB [myntra]> |
```

Step-11 Import the Backup into RDS:

```
mysql -h <endpoint> -u admin -p myntra < mysql_backup.sql
```

(Enter password:)

Step-11 Verify data in RDS:

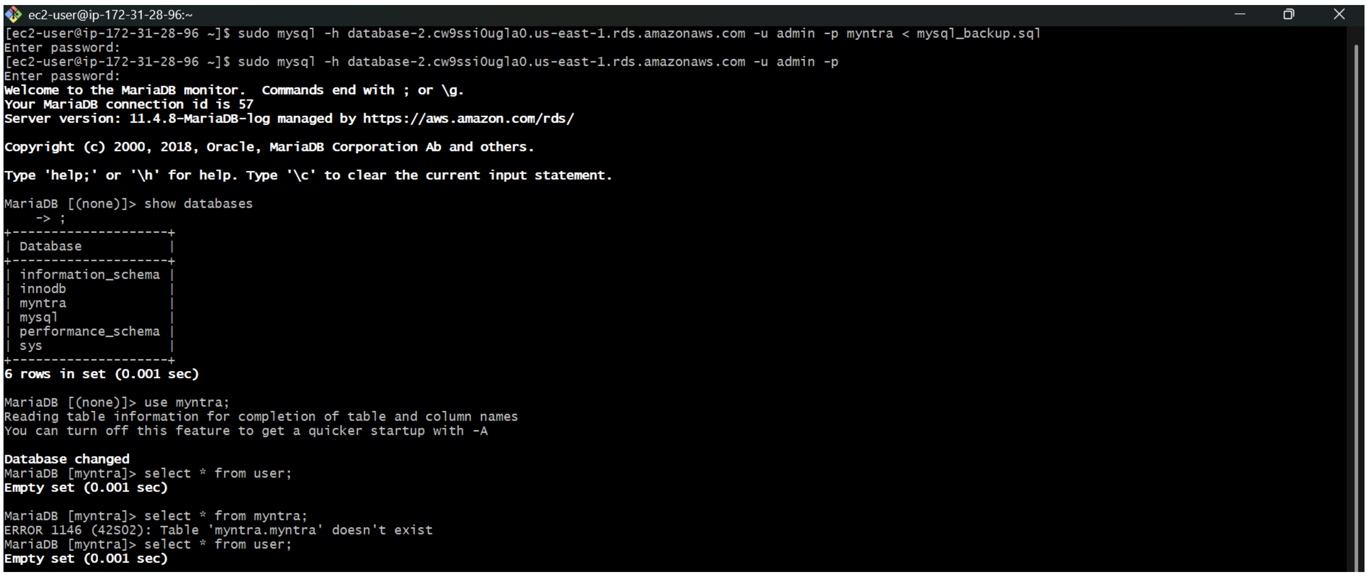
```
sudo mysql -h <endpoint> -u admin -p
```

```
    Show databases;

    use myntr;

    Select * from user;

     exit
```



```
[ec2-user@ip-172-31-28-96 ~]$ sudo mysql -h database-2.cw9ssi0ugla0.us-east-1.rds.amazonaws.com -u admin -p myntra < mysql_backup.sql
Enter password:
[ec2-user@ip-172-31-28-96 ~]$ sudo mysql -h database-2.cw9ssi0ugla0.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 57
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]> select * from user;
Empty set (0.001 sec)

MariaDB [myntra]> select * from myntra;
ERROR 1146 (42S02): Table 'myntra.myntra' doesn't exist
MariaDB [myntra]> select * from user;
Empty set (0.001 sec)
```

☑ Output

All data from EC2's local MariaDB database is now successfully migrated to Amazon RDS.

☑ Conclusion

Migrating your database from EC2 to RDS simplifies management and improves reliability by leveraging AWS managed services. It's a crucial step in moving from IaaS to PaaS architecture on AWS.