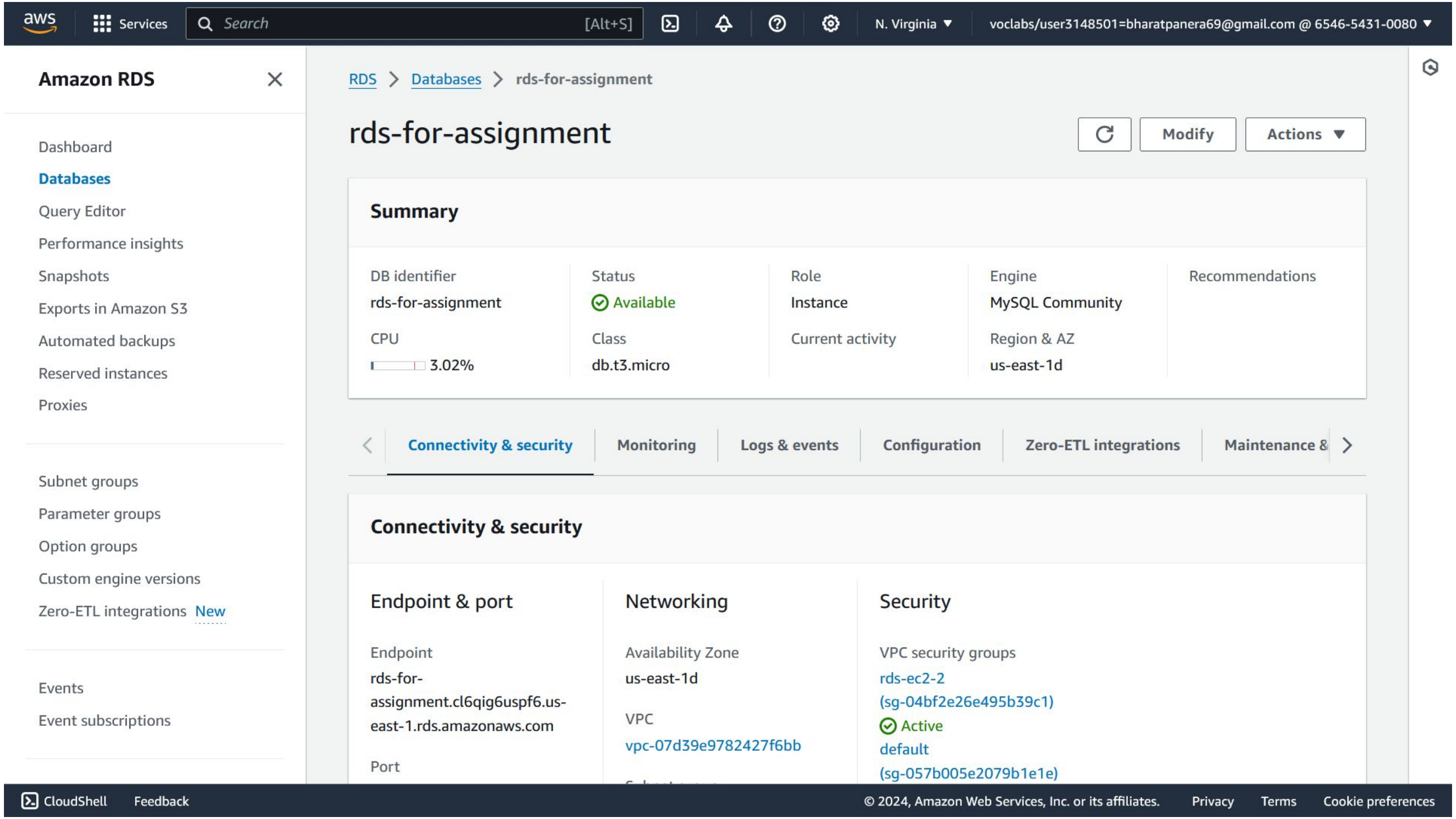
Data Ingestion Tasks

Task 1: Create RDS and import the data into it

Create RDS instance in AWS



rds-for-assignment

Summary

DB identifier: rds-for-assignment
Status: Available
Role: Instance
Engine: MySQL Community
Recommendations
CPU: 3.02%
Class: db.t3.micro
Current activity
Region & AZ: us-east-1d

Connectivity & security

Endpoint & port

Endpoint
rds-for-assignment.cl6qig6uspf6.us-east-1.rds.amazonaws.com

Port

Networking

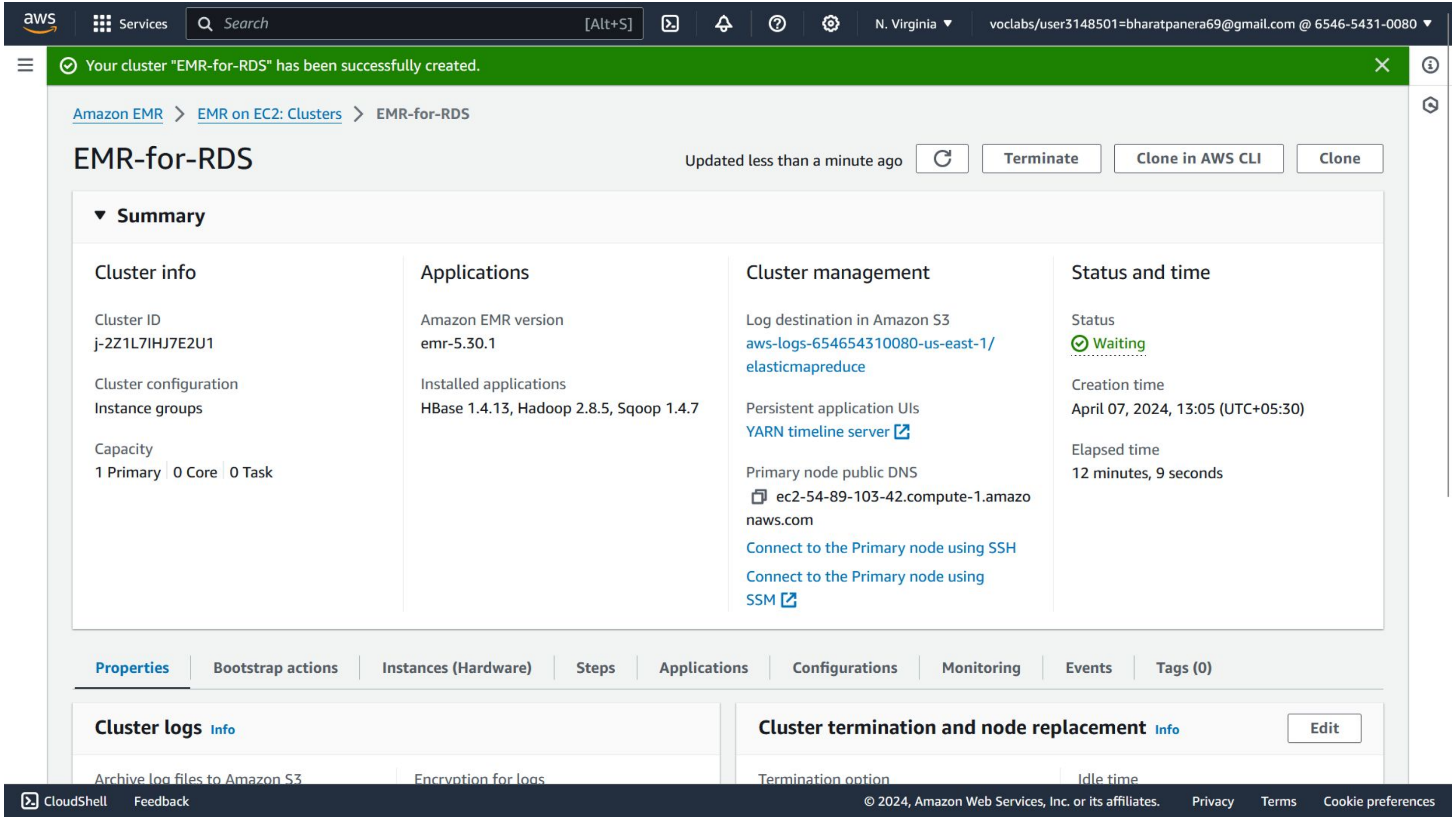
Availability Zone
us-east-1d

VPC
vpc-07d39e9782427f6bb

Security

VPC security groups
rds-ec2-2
(sg-04bf2e26e495b39c1)
Active
default
(sg-057b005e2079b1e1e)

Create EMR in AWS



Your cluster "EMR-for-RDS" has been successfully created.

Amazon EMR > EMR on EC2: Clusters > EMR-for-RDS

EMR-for-RDS

Updated less than a minute ago

Summary

Cluster info

Cluster ID
j-2Z1L7IHJ7E2U1

Cluster configuration
Instance groups

Capacity
1 Primary | 0 Core | 0 Task

Applications

Amazon EMR version
emr-5.30.1

Installed applications
HBase 1.4.13, Hadoop 2.8.5, Sqoop 1.4.7

Cluster management

Log destination in Amazon S3
aws-logs-654654310080-us-east-1/elasticmapreduce

Persistent application UIs
YARN timeline server

Primary node public DNS
ec2-54-89-103-42.compute-1.amazonaws.com

Connect to the Primary node using SSH

Connect to the Primary node using SSM

Status and time

Status
Waiting

Creation time
April 07, 2024, 13:05 (UTC+05:30)

Elapsed time
12 minutes, 9 seconds

Cluster logs Info

Archive log files to Amazon S3

Encryption for logs

Cluster termination and node replacement Info

Termination option

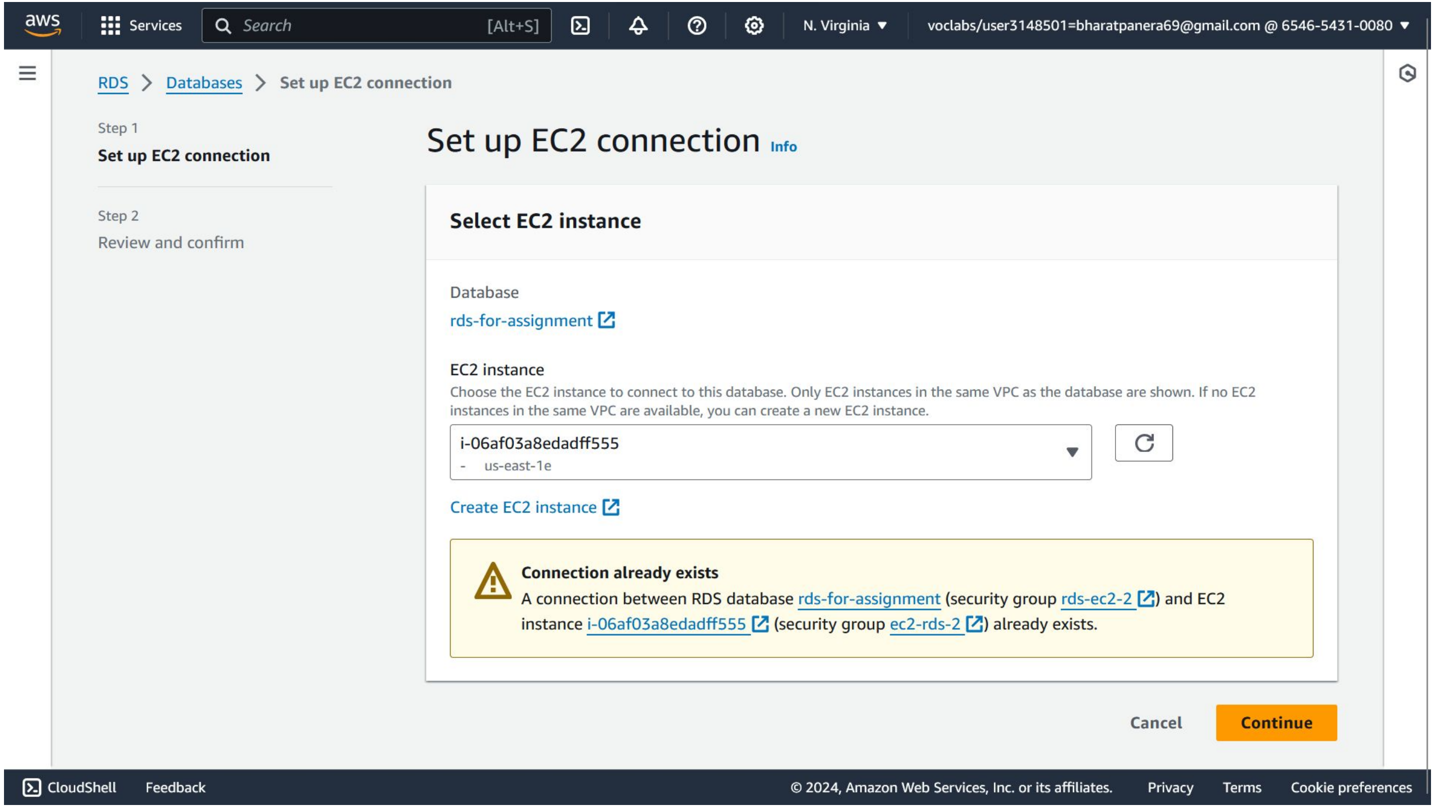
Idle time

Connect RDS to EMR

- To connect RDS to EMR select the RDS and click the “Actions” button
- Then click “set up EC2 connection”
- Choose an appropriate EC2 instance and click “continue”
- Then review and confirm
- It will connect the RDS to EMR cluster
- Then login to EMR from local machine and try to access mysql

Command:
```
mysql -h rds-for-assignment.cl6qig6uspf6.us-east-1.rds.amazonaws.com -P
3306 -u admin -p
```

Connect RDS to EMR



aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3148501=bharatpanera69@gmail.com @ 6546-5431-0080
RDS > Databases > Set up EC2 connection
Step 1
Set up EC2 connection
Step 2
Review and confirm

Set up EC2 connection Info

Select EC2 instance

Database
rds-for-assignment

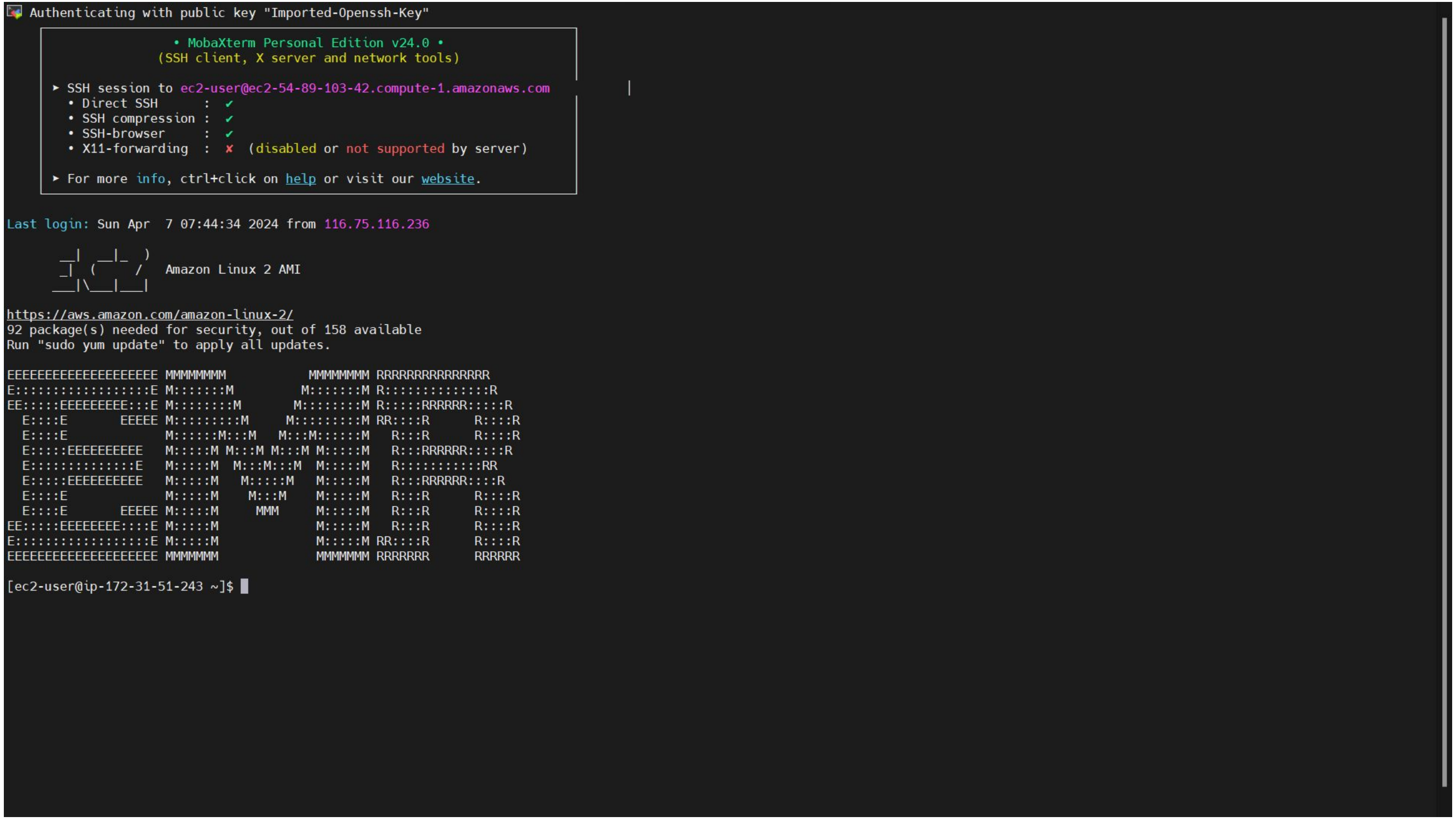
EC2 instance
Choose the EC2 instance to connect to this database. Only EC2 instances in the same VPC as the database are shown. If no EC2 instances in the same VPC are available, you can create a new EC2 instance.

i-06af03a8edadff555
- us-east-1e

Create EC2 instance

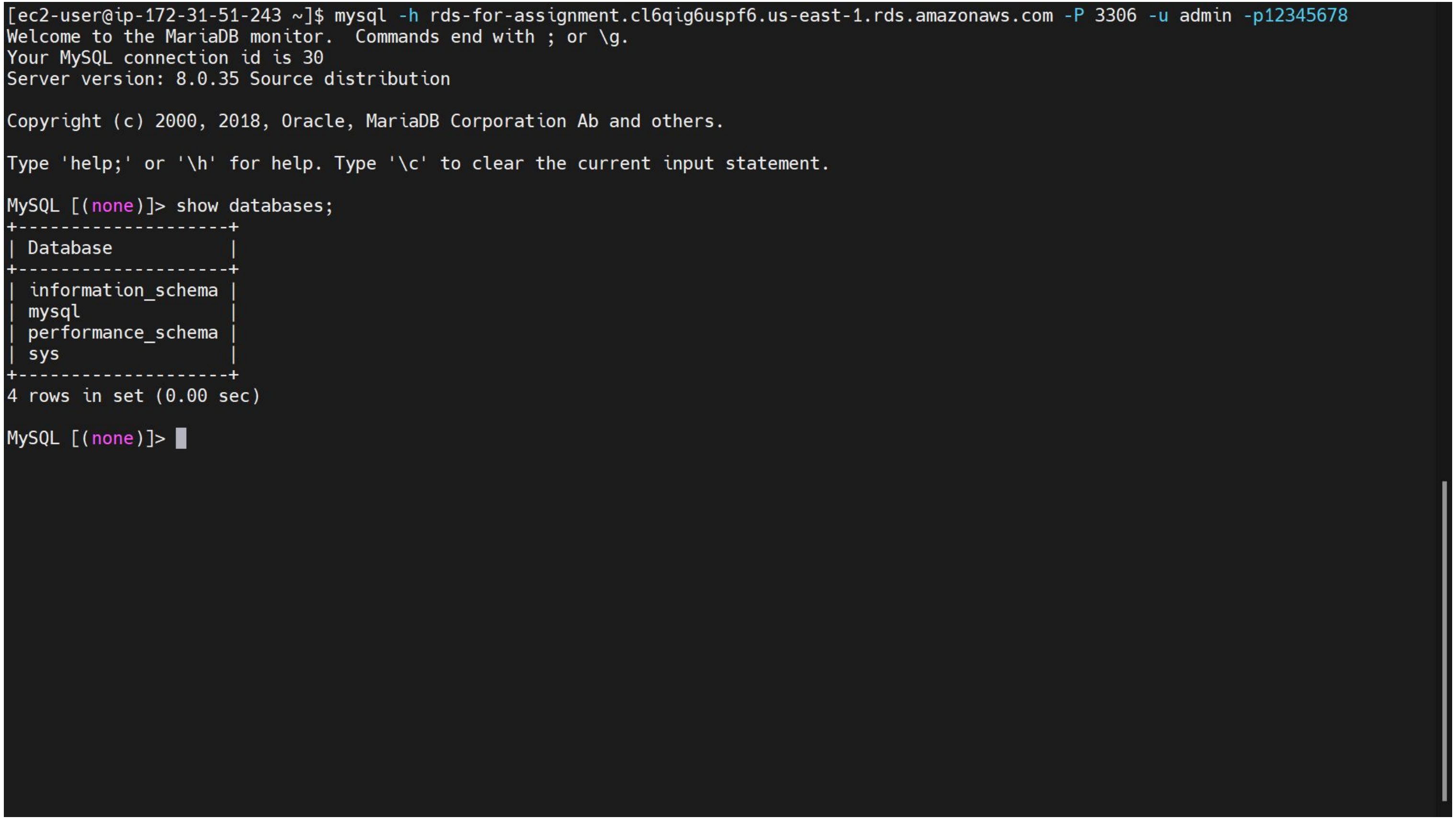
Connection already exists
A connection between RDS database rds-for-assignment (security group rds-ec2-2) and EC2 instance i-06af03a8edadff555 (security group ec2-rds-2) already exists.

Login to EMR



```
Authenticating with public key "Imported-Openssh-Key"
    ┌──────────────────────────────────────────────────────────────────────┐
    │                 • MobaXterm Personal Edition v24.0 •                 │
    │               (SSH client, X server and network tools)               │
    │                                                                      │
    │ ▸ SSH session to ec2-user@ec2-54-89-103-42.compute-1.amazonaws.com
    │   • Direct SSH      :  ✔                                             │
    │   • SSH compression :  ✔                                             │
    │   • SSH-browser     :  ✔                                             │
    │   • X11-forwarding  :  ✘  (disabled or not supported by server)      │
    │                                                                      │
    │ ▸ For more info, ctrl+click on help or visit our website.            │
    └──────────────────────────────────────────────────────────────────────┘

Last login: Sun Apr  7 07:44:34 2024 from 116.75.116.236

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
92 package(s) needed for security, out of 158 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[ec2-user@ip-172-31-51-243 ~]$ █
```

Access the MySQL from EMR

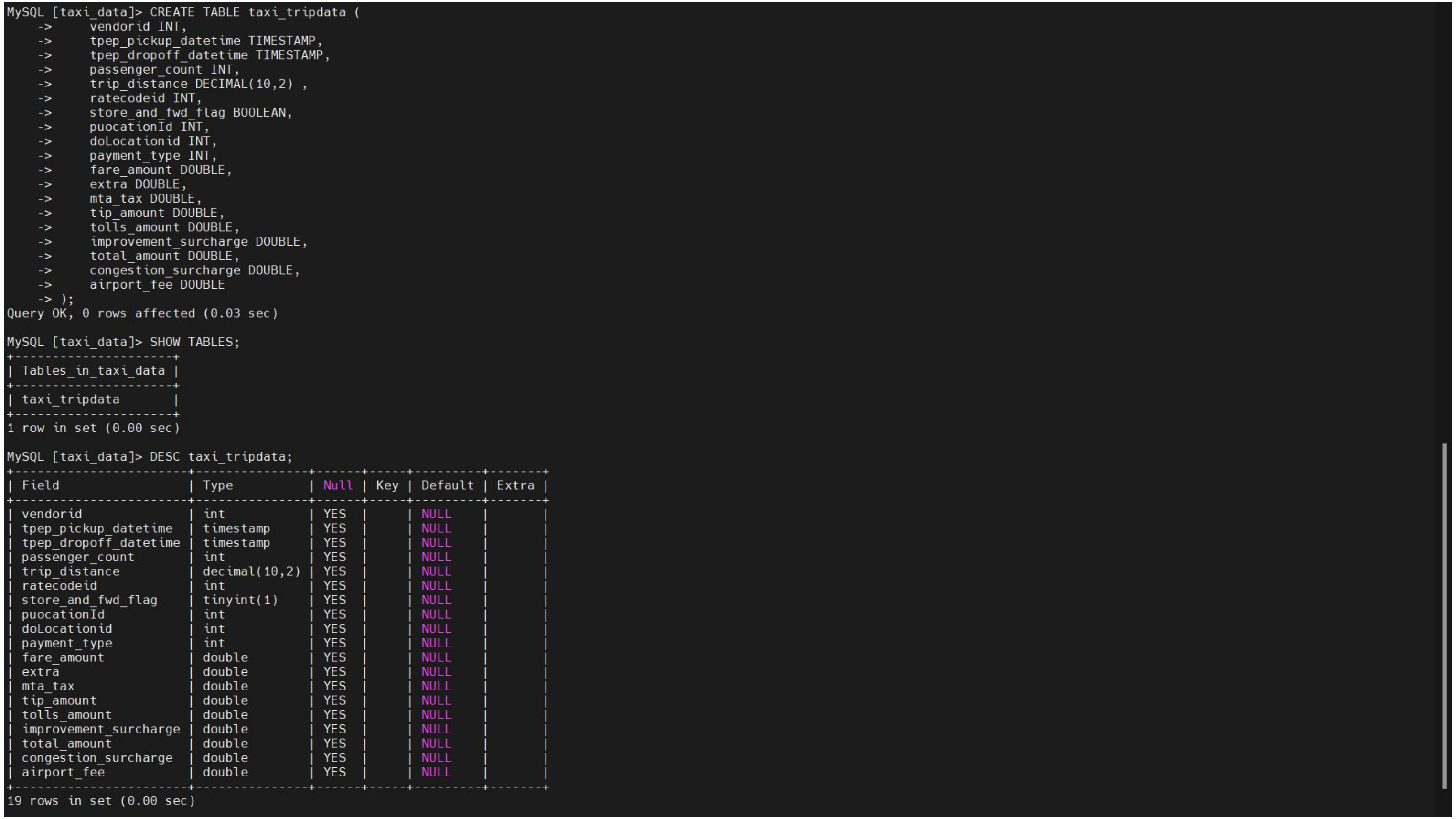


```
[ec2-user@ip-172-31-51-243 ~]$ mysql -h rds-for-assignment.cl6qig6uspf6.us-east-1.rds.amazonaws.com -P 3306 -u admin -p12345678
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 30
Server version: 8.0.35 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)

MySQL [(none)]> 
```

Create database “taxi_data” & table “taxi_tripdata”

- We will create a database “taxi_data”
  - **Syntax:** CREATE DATABASE taxi_data;
- Now we will create a table “taxi_tripdata”
  - **Syntax:** CREATE TABLE taxi_tripdata (vendorid INT, tpep_pickup_datetime TIMESTAMP, tpep_dropoff_datetime TIMESTAMP, passenger_count INT, trip_distance DECIMAL(10,2), ratecodeid INT, store_and_fwd_flag BOOLEAN, puocationId INT, doLocationid INT, payment_type INT, fare_amount DOUBLE, extra DOUBLE, mta_tax DOUBLE, tip_amount DOUBLE, tolls_amount DOUBLE, improvement_surcharge DOUBLE, total_amount DOUBLE, congestion_surcharge DOUBLE, airport_fee DOUBLE);

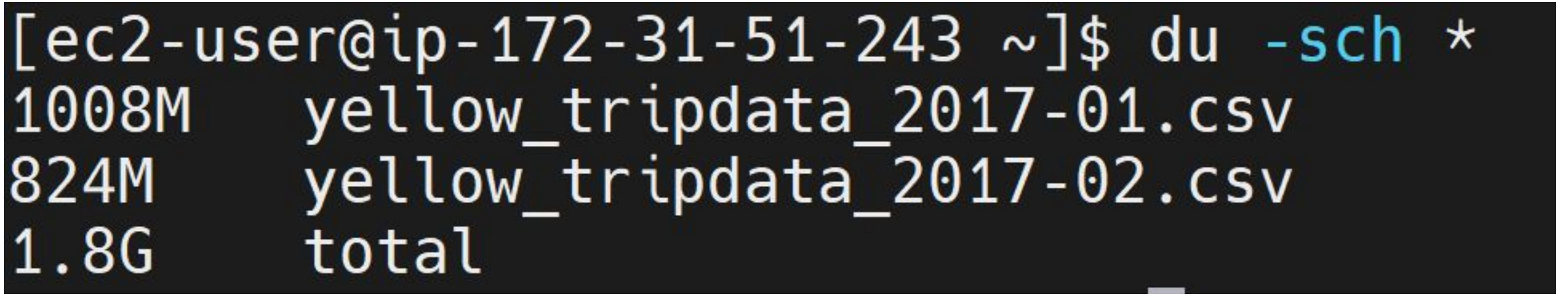
Create database “taxi_data” & table “taxi_tripdata”



```
MySQL [taxi_data]> CREATE TABLE taxi_tripdata (
    ->     vendorid INT,
    ->     tpep_pickup_datetime TIMESTAMP,
    ->     tpep_dropoff_datetime TIMESTAMP,
    ->     passenger_count INT,
    ->     trip_distance DECIMAL(10,2) ,
    ->     ratecodeid INT,
    ->     store_and_fwd_flag BOOLEAN,
    ->     puocationId INT,
    ->     doLocationid INT,
    ->     payment_type INT,
    ->     fare_amount DOUBLE,
    ->     extra DOUBLE,
    ->     mta_tax DOUBLE,
    ->     tip_amount DOUBLE,
    ->     tolls_amount DOUBLE,
    ->     improvement_surcharge DOUBLE,
    ->     total_amount DOUBLE,
    ->     congestion_surcharge DOUBLE,
    ->     airport_fee DOUBLE
    -> );
Query OK, 0 rows affected (0.03 sec)

MySQL [taxi_data]> SHOW TABLES;
+---------------------+
| Tables_in_taxi_data |
+---------------------+
| taxi_tripdata       |
+---------------------+
1 row in set (0.00 sec)

MySQL [taxi_data]> DESC taxi_tripdata;
+-----------------------+---------------+------+-----+---------+-------+
| Field                 | Type          | Null | Key | Default | Extra |
+-----------------------+---------------+------+-----+---------+-------+
| vendorid              | int           | YES  |     | NULL    |       |
| tpep_pickup_datetime  | timestamp     | YES  |     | NULL    |       |
| tpep_dropoff_datetime | timestamp     | YES  |     | NULL    |       |
| passenger_count       | int           | YES  |     | NULL    |       |
| trip_distance         | decimal(10,2) | YES  |     | NULL    |       |
| ratecodeid            | int           | YES  |     | NULL    |       |
| store_and_fwd_flag    | tinyint(1)    | YES  |     | NULL    |       |
| puocationId           | int           | YES  |     | NULL    |       |
| doLocationid          | int           | YES  |     | NULL    |       |
| payment_type          | int           | YES  |     | NULL    |       |
| fare_amount           | double        | YES  |     | NULL    |       |
| extra                 | double        | YES  |     | NULL    |       |
| mta_tax               | double        | YES  |     | NULL    |       |
| tip_amount            | double        | YES  |     | NULL    |       |
| tolls_amount          | double        | YES  |     | NULL    |       |
| improvement_surcharge | double        | YES  |     | NULL    |       |
| total_amount          | double        | YES  |     | NULL    |       |
| congestion_surcharge  | double        | YES  |     | NULL    |       |
| airport_fee           | double        | YES  |     | NULL    |       |
+-----------------------+---------------+------+-----+---------+-------+
19 rows in set (0.00 sec)
```

Download the required files into EMR

- Download the yellow_tripdata_2017-01.csv & yellow_tripdata_2017-02.csv files into EMR.
  - **Commands:**
    - wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv
    - wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-02.csv

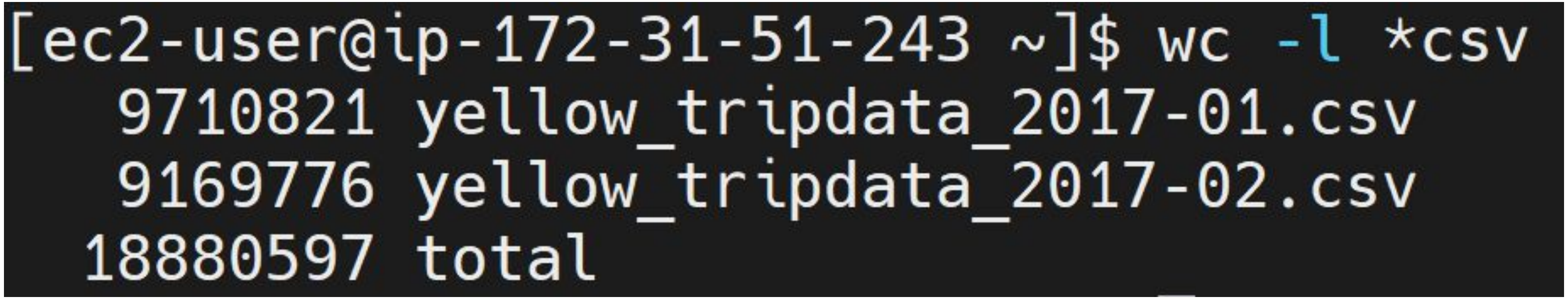


```
[ec2-user@ip-172-31-51-243 ~]$ du -sch *
1008M   yellow_tripdata_2017-01.csv
824M    yellow_tripdata_2017-02.csv
1.8G    total
```

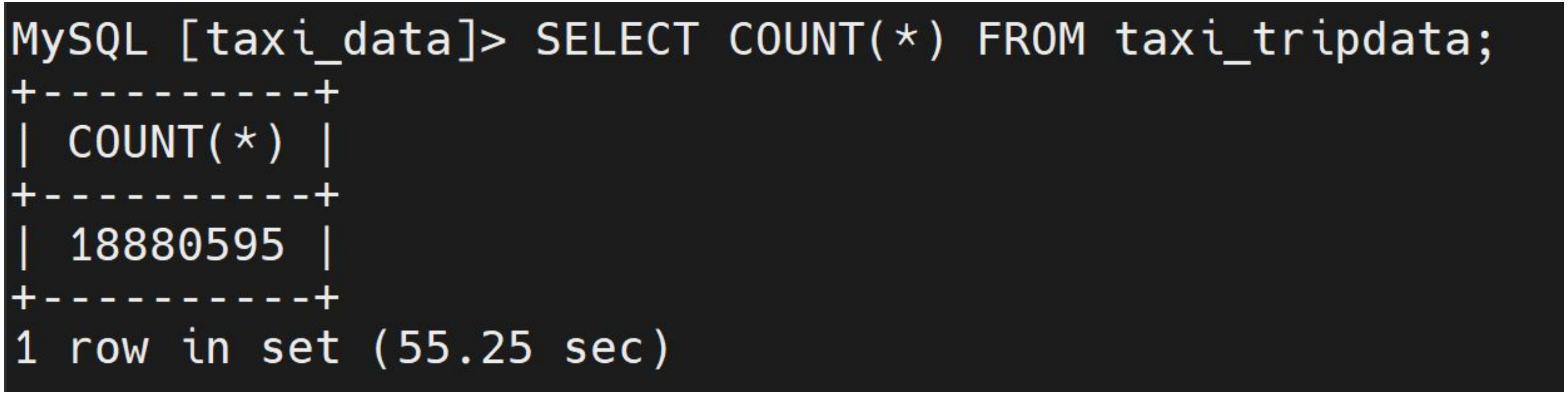
Load the taxi data into MySQL table

- Take absolute path of the CSV files using command "readlink -f *csv"
  - /home/ec2-user/yellow_tripdata_2017-01.csv
  - /home/ec2-user/yellow_tripdata_2017-02.csv
- Load the .csv files into MySQL table using below query.
  - LOAD DATA LOCAL INFILE '/home/ec2-user/yellow_tripdata_2017-01.csv' INTO TABLE taxi_data.taxi_tripdata FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES;
  - LOAD DATA LOCAL INFILE '/home/ec2-user/yellow_tripdata_2017-02.csv' INTO TABLE taxi_data.taxi_tripdata FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES;
- Then we can cross check with counting the number of imported records with number of lines present in csv files.
  - To check number of lines in csv files: “wc -l *csv”
  - To check imported records in MySQL table:
    - SELECT COUNT(*) FROM taxi_data.taxi_tripdata;
  - It should be totale of both the csv files.

Load the taxi data into MySQL table



```
[ec2-user@ip-172-31-51-243 ~]$ wc -l *csv
   9710821 yellow_tripdata_2017-01.csv
   9169776 yellow_tripdata_2017-02.csv
  18880597 total
```



```
MySQL [taxi_data]> SELECT COUNT(*) FROM taxi_tripdata;
+----------+
| COUNT(*) |
+----------+
| 18880595 |
+----------+
1 row in set (55.25 sec)
```