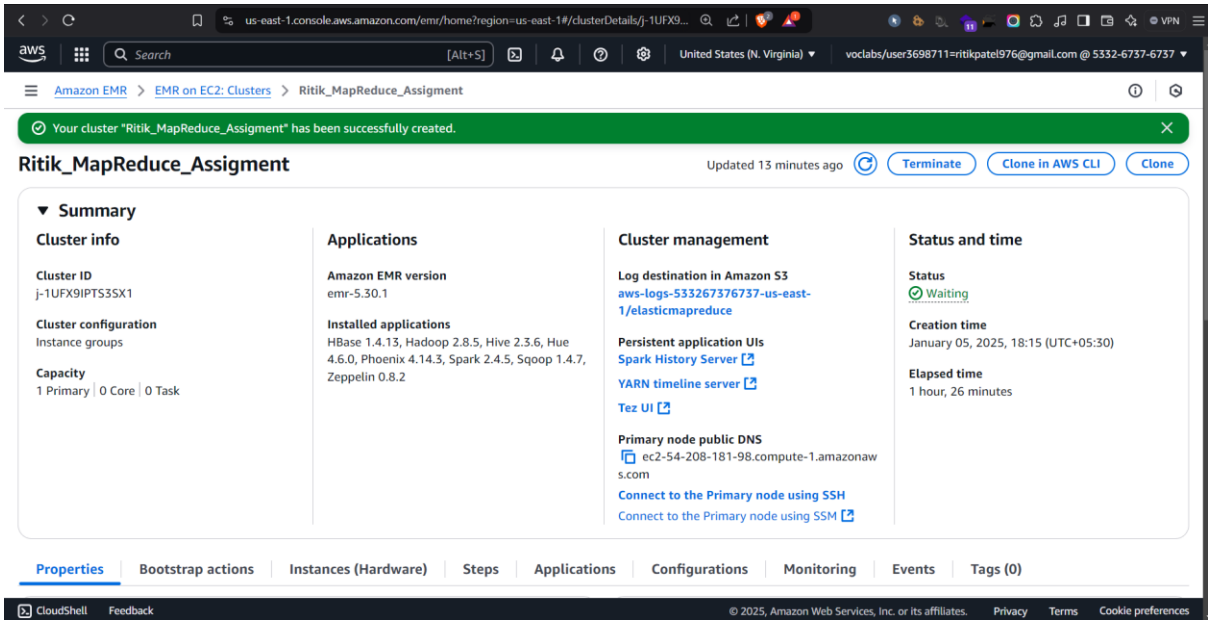
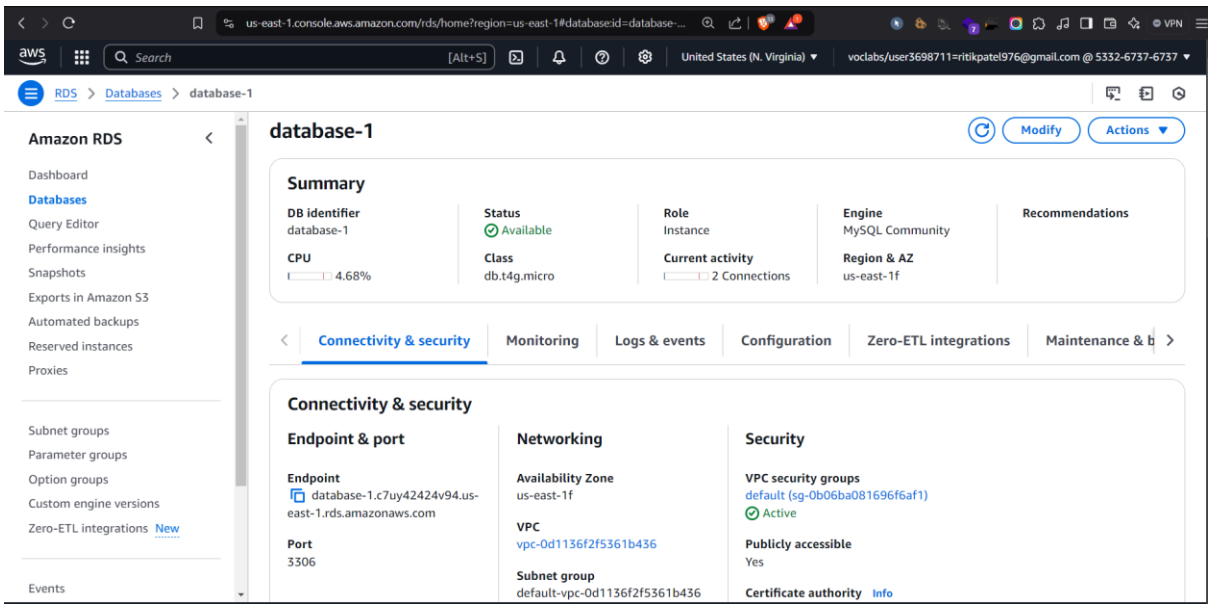
Following are the commands and steps to upload data to RDS Instance:

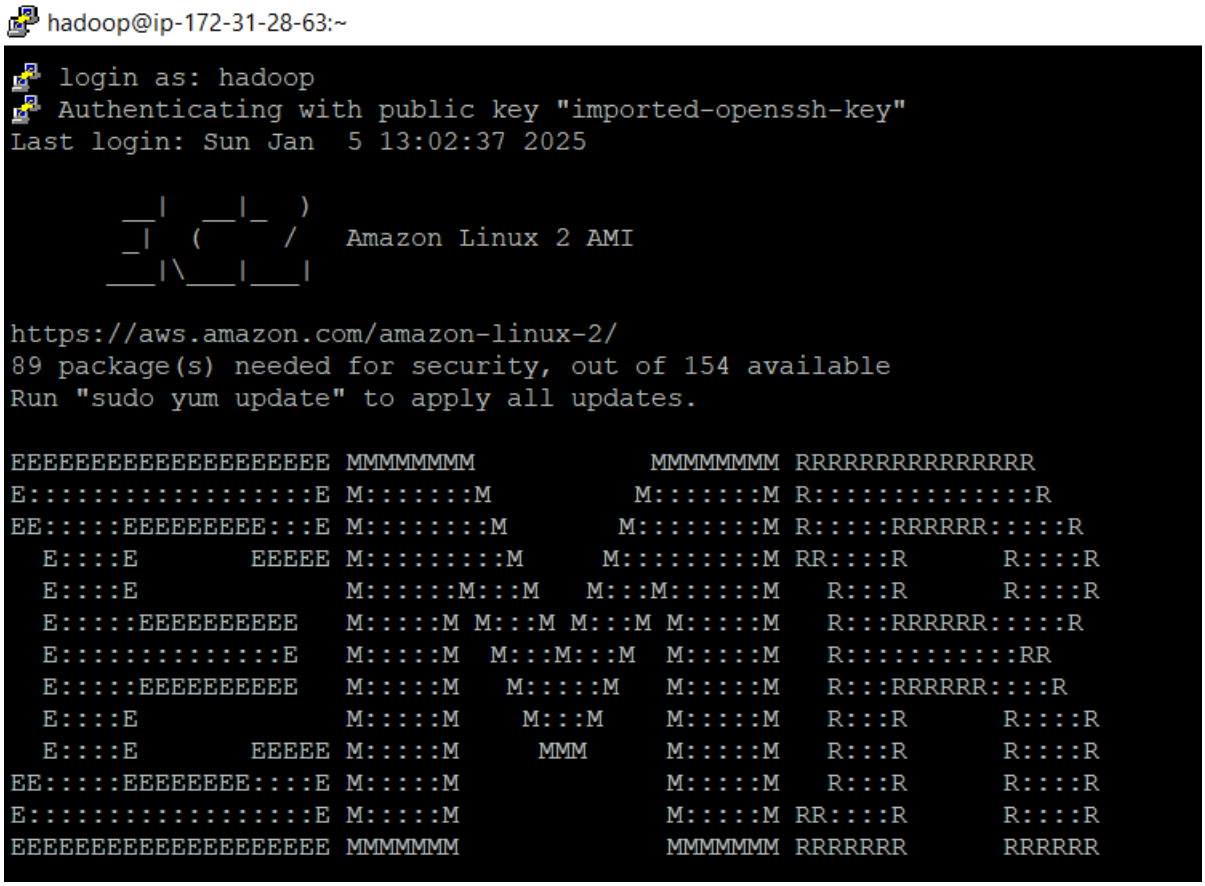
Step 1: First we need to create the emr cluster with required disk space. Given below is my EMR Cluster.



Step 2: We also need to create our database.



Step 3: Now we need to login to our EMR SSH. I am using widows device so I have logged in using Putty software.



Step 4: Now we will first download the dataset in our system using “wget” command.

The commands to download the files are:

1. wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv

2. wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-02.csv

```
[hadoop@ip-172-31-28-63 ~]$ wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv
--2025-01-05 13:06:51--  https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv
Resolving nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)... 54.231.234.65, 3.5.29.249, 16.15.192.125, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|54.231.234.65|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 914029540 (872M) [text/csv]
Saving to: 'yellow_tripdata_2017-01.csv'

100%[==========>] 914,029,540 53.2MB/s   in 15s

2025-01-05 13:07:07 (56.6 MB/s) - 'yellow_tripdata_2017-01.csv' saved [914029540/914029540]

[hadoop@ip-172-31-28-63 ~]$ ls
yellow_tripdata_2017-01.csv
[hadoop@ip-172-31-28-63 ~]$ wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-02.csv
--2025-01-05 13:07:23--  https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-02.csv
Resolving nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)... 3.5.28.148, 52.217.197.225, 52.217.124.49, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|3.5.28.148|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 863487050 (823M) [text/csv]
Saving to: 'yellow_tripdata_2017-02.csv'

100%[==========>] 863,487,050 51.8MB/s   in 14s

2025-01-05 13:07:37 (57.3 MB/s) - 'yellow_tripdata_2017-02.csv' saved [863487050/863487050]
```

Step 5: using ls we can check whether the files are downloaded or not. And using wc -l we can get the line count for each file so we can check that do we have same number of records in the database.

1: ls

2: wc -l *

```
[hadoop@ip-172-31-28-63 ~]$ ls
yellow_tripdata_2017-01.csv  yellow_tripdata_2017-02.csv
[hadoop@ip-172-31-28-63 ~]$ ls -lh
total 1.7G
-rw-rw-r-- 1 hadoop hadoop 872M Nov 25  2022 yellow_tripdata_2017-01.csv
-rw-rw-r-- 1 hadoop hadoop 824M Nov 25  2022 yellow_tripdata_2017-02.csv
[hadoop@ip-172-31-28-63 ~]$ wc -l *
   9710821 yellow_tripdata_2017-01.csv
   9169776 yellow_tripdata_2017-02.csv
  18880597 total
```

Step 6: We will use head command to check weather our files contain header line or not.

1: head -10 yellow_tripdata_2017-01.csv

```
[hadoop@ip-172-31-28-63 ~]$ head -10 yellow_tripdata_2017-01.csv
VendorID,tpep_pickup_datetime,tpep_dropoff_datetime,passenger_count,trip_distance,RatecodeID,store_and_fwd_flag,PULocationID,DOLocationID,payment_type,fare_amount,extra,mta_tax,tip_amoun
olls_amount,improvement_surcharge,total_amount,congestion_surcharge,airport_fee
1,2017-01-01 00:32:05,2017-01-01 00:37:48,1,1.2,1,N,140,236,2,6.5,0.5,0.5,0.0,0.0,0.3,7.8,,
1,2017-01-01 00:43:25,2017-01-01 00:47:42,2,0.7,1,N,237,140,2,5.0,0.5,0.5,0.0,0.0,0.3,6.3,,
1,2017-01-01 00:49:10,2017-01-01 00:53:53,2,0.8,1,N,140,237,2,5.5,0.5,0.5,0.0,0.0,0.3,6.8,,
1,2017-01-01 00:36:42,2017-01-01 00:41:09,1,1.1,1,N,41,42,2,6.0,0.5,0.5,0.0,0.0,0.3,7.3,,
1,2017-01-01 00:07:41,2017-01-01 00:18:16,1,3.0,1,N,48,263,2,11.0,0.5,0.5,0.0,0.0,0.3,12.3,,
1,2017-01-01 00:20:52,2017-01-01 00:24:59,2,0.7,1,N,236,262,2,5.0,0.5,0.5,0.0,0.0,0.3,6.3,,
1,2017-01-01 00:33:49,2017-01-01 00:42:38,2,1.6,1,N,236,238,1,8.0,0.5,0.5,1.85,0.0,0.3,11.15,,
1,2017-01-01 00:48:22,2017-01-01 00:52:15,2,0.6,1,N,238,239,1,5.0,0.5,0.5,1.25,0.0,0.3,7.55,,
1,2017-01-01 00:57:12,2017-01-01 01:06:28,2,1.0,1,N,239,48,1,7.5,0.5,0.5,1.75,0.0,0.3,10.55,,
[hadoop@ip-172-31-28-63 ~]$ head -10 yellow_tripdata_2017-02.csv
VendorID,tpep_pickup_datetime,tpep_dropoff_datetime,passenger_count,trip_distance,RatecodeID,store_and_fwd_flag,PULocationID,DOLocationID,payment_type,fare_amount,extra,mta_tax,tip_amoun
olls_amount,improvement_surcharge,total_amount,congestion_surcharge,airport_fee
1,2017-02-01 00:19:20,2017-02-01 00:25:56,1,2.9,1,N,75,162,2,9.5,0.5,0.5,0.0,0.0,0.3,10.8,,
1,2017-02-01 00:19:55,2017-02-01 00:33:06,1,4.9,1,N,246,166,1,15.0,0.5,0.5,3.25,0.0,0.3,19.55,,
1,2017-02-01 00:01:15,2017-02-01 00:09:03,2,1.5,1,N,237,170,1,7.5,0.5,0.5,1.5,0.0,0.3,10.3,,
2,2017-02-01 00:06:36,2017-02-01 00:14:50,5,1.51,1,N,137,236,2,7.5,0.5,0.5,0.0,0.0,0.3,8.8,,
1,2017-02-01 00:07:53,2017-02-01 00:14:36,1,1.4,1,N,112,112,1,7.0,0.5,0.5,2.45,0.0,0.3,10.75,,
1,2017-02-01 00:30:59,2017-02-01 00:47:30,1,3.8,1,N,255,36,2,15.0,0.5,0.5,0.0,0.0,0.3,16.3,,
1,2017-02-01 00:00:40,2017-02-01 00:18:23,1,4.7,1,N,186,166,1,16.5,0.5,0.5,1.78,0.0,0.3,19.58,,
1,2017-02-01 00:24:48,2017-02-01 00:30:57,1,1.1,1,N,151,239,1,6.5,0.5,0.5,1.55,0.0,0.3,9.35,,
2,2017-02-01 00:05:16,2017-02-01 00:41:23,1,12.89,1,N,132,181,1,39.0,0.5,0.5,8.06,0.0,0.3,48.36,,
[hadoop@ip-172-31-28-63 ~]$
```

Our both the files contain the header line so we will ignore the first line while importing the data to the database.

step 7: we need to modify some permission of the file using chmod command so that it can be accessed by database to load the data.

1: chmod 775 yellow_tripdata_2017-01.csv

2: chmod 775 yellow_tripdata_2017-02.csv

```
[hadoop@ip-172-31-28-63 ~]$ chmod 775 yellow_tripdata_2017-01.csv
[hadoop@ip-172-31-28-63 ~]$ ls -l
total 1735860
-rwxrwxr-x 1 hadoop hadoop 914029540 Nov 25  2022 yellow_tripdata_2017-01.csv
-rw-rw-r-- 1 hadoop hadoop 863487050 Nov 25  2022 yellow_tripdata_2017-02.csv
[hadoop@ip-172-31-28-63 ~]$ chmod 775 yellow_tripdata_2017-02.csv
[hadoop@ip-172-31-28-63 ~]$ ls -l
total 1735860
-rwxrwxr-x 1 hadoop hadoop 914029540 Nov 25  2022 yellow_tripdata_2017-01.csv
-rwxrwxr-x 1 hadoop hadoop 863487050 Nov 25  2022 yellow_tripdata_2017-02.csv
```

Step 8: Now we will login to our mysql database.

1: mysql -h database-1.c7uy42424v94.us-east-1.rds.amazonaws.com -P 3306 -u admin -p

```
[hadoop@ip-172-31-28-63 ~]$ mysql -h database-1.c7uy42424v94.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 37
Server version: 8.0.39 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                |
+--------------------+
```

Step 9: Here I have created database named assignment where I have created the tables yellow_taxi_data where I have imported the data form csv files.

1: create database assigment;

2: use assigment;

```
MySQL [(none)]> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.01 sec)

MySQL [(none)]> create database assigment;
Query OK, 1 row affected (0.02 sec)

MySQL [(none)]> use assigment;
Database changed
MySQL [assigment]>
```

3: CREATE TABLE yellow_taxi_data (

VendorID INT,

tpep_pickup_datetime DATETIME,

tpep_dropoff_datetime DATETIME,

passenger_count INT,

trip_distance FLOAT,

RatecodeID INT,

store_and_fwd_flag CHAR(1),

PULocationID INT,

DOLocationID INT,

payment_type INT,

fare_amount FLOAT,

extra FLOAT,

mta_tax FLOAT,

tip_amount FLOAT,

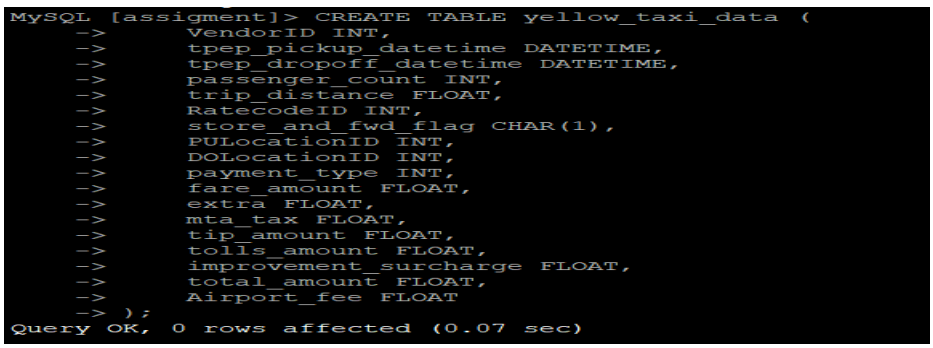
tolls_amount FLOAT,

improvement_surcharge FLOAT,

total_amount FLOAT,

Airport_fee FLOAT

);



step 10: Now we can load the data from both the files one after another using commands given below.

1: LOAD DATA INFILE '/home/hadoop/yellow_tripdata_2017-01.csv'

INTO TABLE yellow_taxi_data

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n'

IGNORE 1 ROWS

(VendorID, tpep_pickup_datetime, tpep_dropoff_datetime, passenger_count, trip_distance, RatecodeID,

store_and_fwd_flag, PULocationID, DOLocationID, payment_type, fare_amount, extra, mta_tax,

tip_amount, tolls_amount, improvement_surcharge, total_amount, Airport_fee);

2: LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-02.csv'

INTO TABLE yellow_taxi_data

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n'

IGNORE 1 ROWS

(VendorID, tpep_pickup_datetime, tpep_dropoff_datetime, passenger_count, trip_distance, RatecodeID,

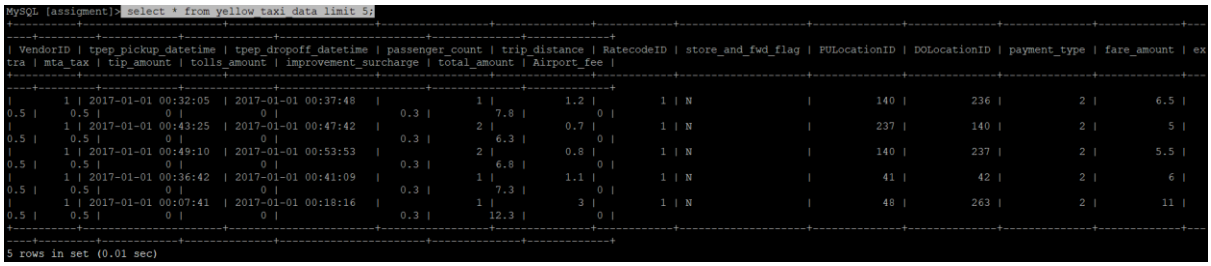
store_and_fwd_flag, PULocationID, DOLocationID, payment_type, fare_amount, extra, mta_tax,

tip_amount, tolls_amount, improvement_surcharge, total_amount, Airport_fee);

The above command specifies the file location from where the data is to be imported. Then, the table to which data should be imported, fields should be terminated by ‘,’ and lines by ‘\n’ and first line is to be ignored. Then at last all the columns are specified.

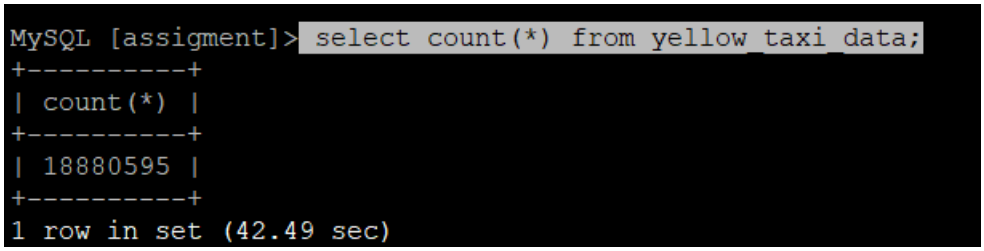
step 11: Now to check weather all the data is properly imported or not we have used following commands.

1: select * from yellow_taxi_data limit 5;



This shows that data is properly loaded into the table.

2: select count(*) from yellow_taxi_data;



when we did “wc -l * “ total lines from both the files where 18880897. And total records in the table is 18880595 because we have skipped head line form both the files.