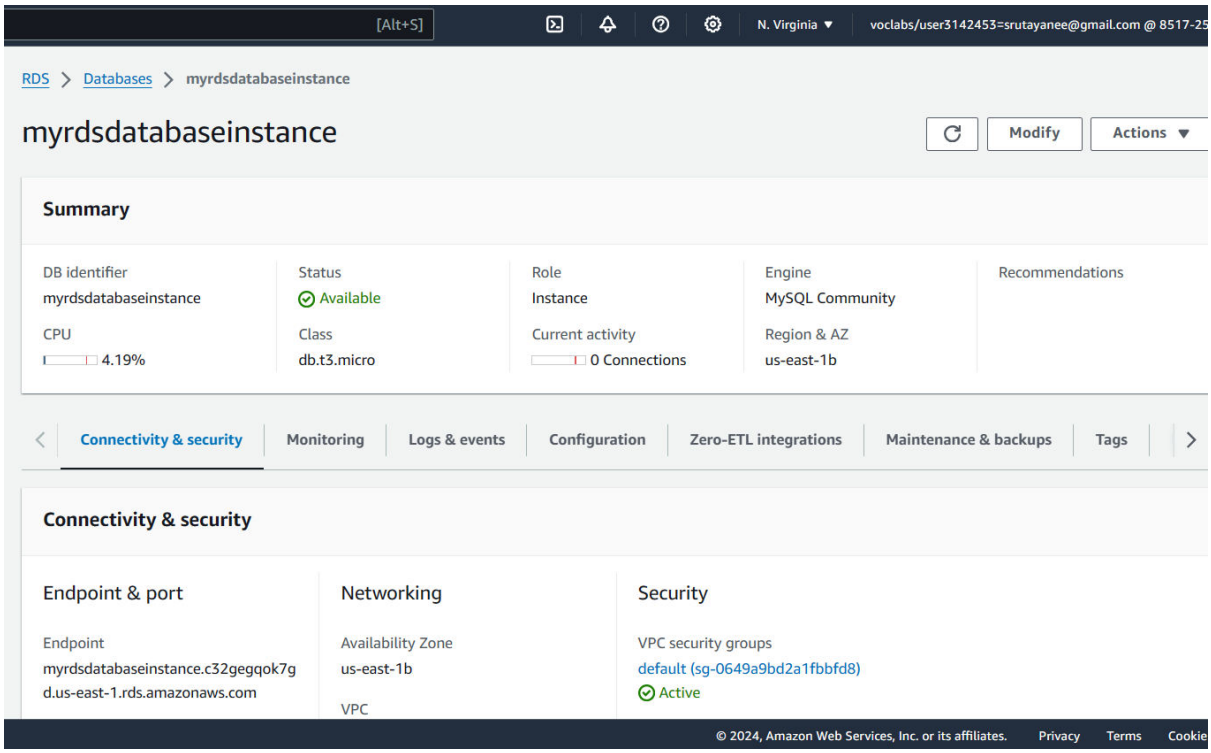
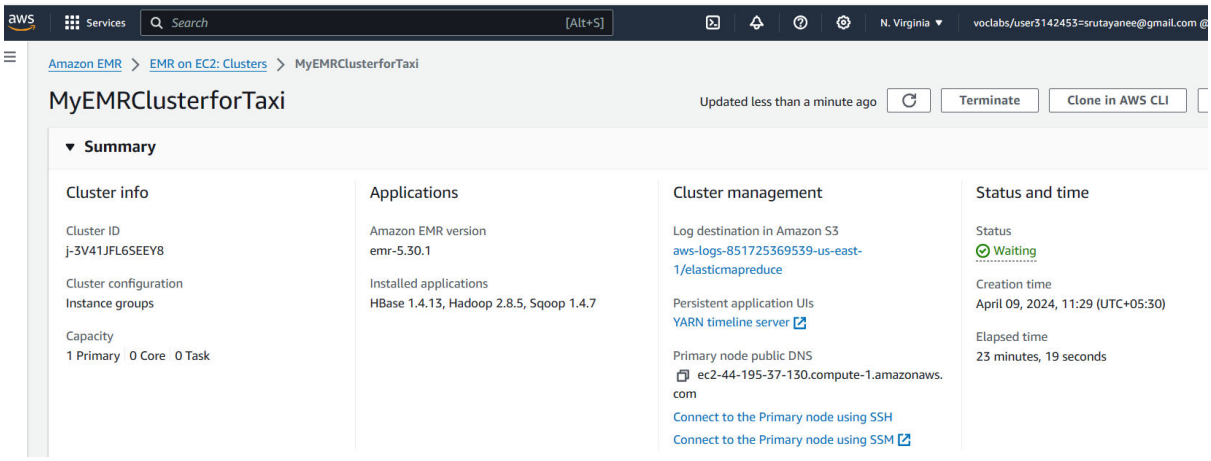
Task 1: Create an RDS instance in your AWS account and upload the data to the RDS instance. Since the dataset is huge, you need to upload the data from only two files (*i.e.* yellow_tripdata_2017-01.csv & yellow_tripdata_2017-02.csv) from the dataset.

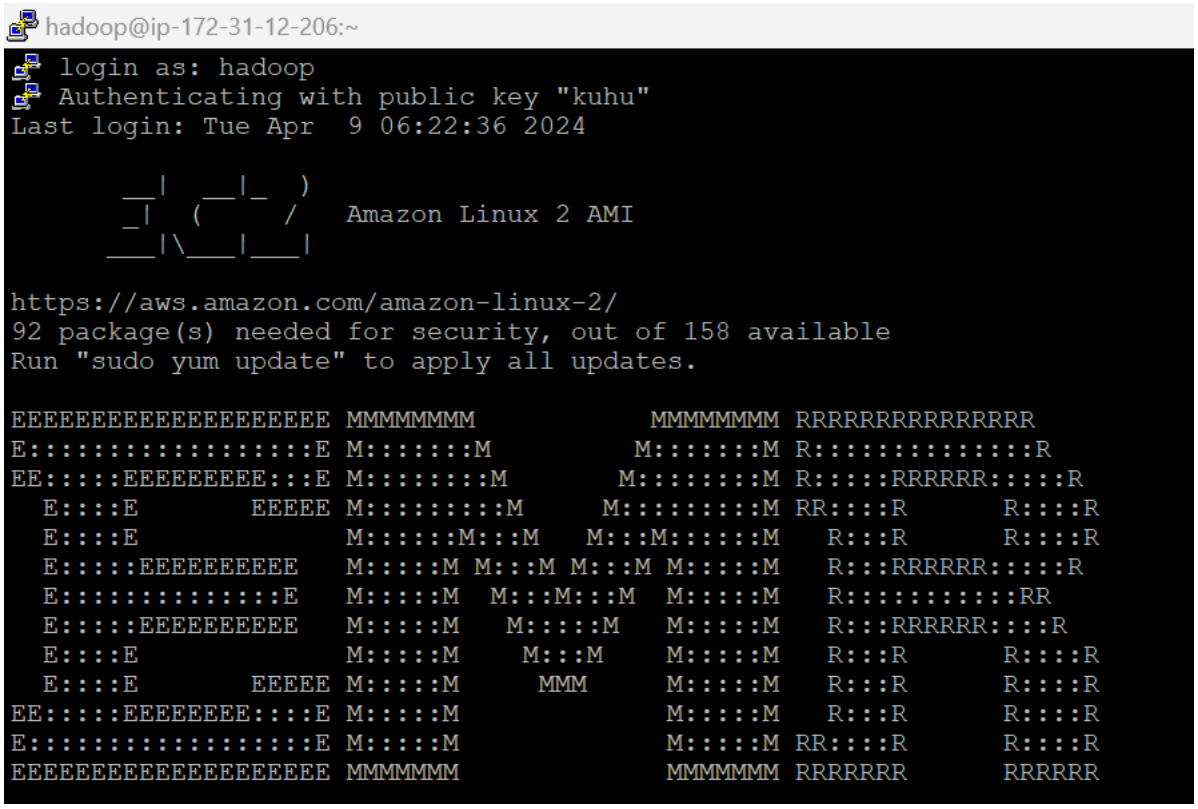
Solution: We followed the below steps to complete the above task:

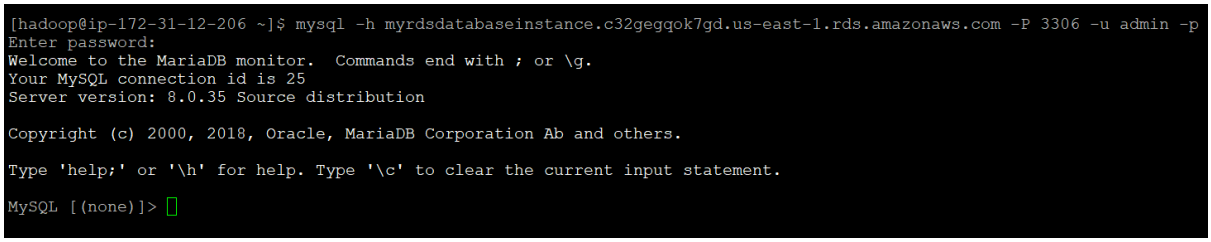
1. First, we created the RDS instance and then EMR cluster (containing Hadoop, Sqoop and HBase) and then established a connection between the two so that we can access the RDS from the EMR cluster.



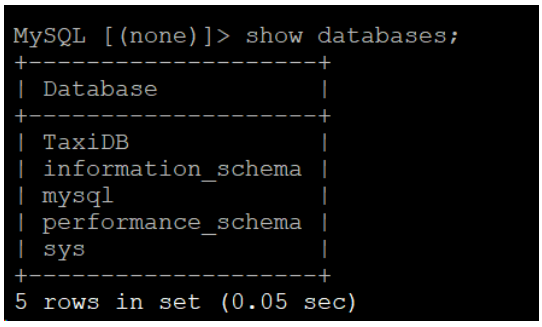


2. Next, we logged into the EMR cluster and ran the below command to access our RDS database: **mysql -h myrdsdatabaseinstance.c32gegqok7gd.us-east-1.rds.amazonaws.com -P 3306 -u admin -p**





3. After entering the password, we ran the **show databases;** command to see what all databases were present.



4. Next, we ran the command **use TaxiDB;** (this is the database which we created during creation of RDS instance). This is the database inside which we will create the table for inserting the yellow taxi data into.

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

5. Next, we ran the command **show tables;** which will show all the tables that are available and then we ran the below command to create the table where we will be pushing the yellow taxi data:

```
CREATE TABLE TripData (
        VendorID INT,
        tpep_pickup_datetime TIMESTAMP NOT NULL DEFAULT '0000-00-00 00:00:00',
        tpep_dropoff_datetime TIMESTAMP NOT NULL DEFAULT '0000-00-00 00:00:00',
        passenger_count INT,
        trip_distance DOUBLE,
        RatecodeID INT,
        store_and_fwd_flag VARCHAR(2),
        PULocationID 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
);
```

```
MySQL [TaxiDB]> CREATE TABLE TripData (
    -> VendorID INT,
    -> tpep_pickup_datetime TIMESTAMP NOT NULL DEFAULT '0000-00-00 00:00:00',
    -> tpep_dropoff_datetime TIMESTAMP NOT NULL DEFAULT '0000-00-00 00:00:00',
    -> passenger_count INT,
    -> trip_distance DOUBLE,
    -> RatecodeID INT,
    -> store_and_fwd_flag VARCHAR(2),
    -> PULocationID 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)
```

6. Next, we ran the command **show tables;** to see if our table got created.

```
MySQL [TaxiDB]> show tables;
+------------------+
| Tables_in_TaxiDB |
+------------------+
| TripData         |
+------------------+
1 row in set (0.00 sec)
```

7. Next, we ran the command **desc TripData;** to see the table schema.

```
MySQL [TaxiDB]> desc TripData;
+-----------------------+------------+------+-----+---------------------+-------+
| Field                 | Type       | Null | Key | Default             | Extra |
+-----------------------+------------+------+-----+---------------------+-------+
| VendorID              | int        | YES  |     | NULL                |       |
| tpep_pickup_datetime  | timestamp  | NO   |     | 0000-00-00 00:00:00 |       |
| tpep_dropoff_datetime | timestamp  | NO   |     | 0000-00-00 00:00:00 |       |
| passenger_count       | int        | YES  |     | NULL                |       |
| trip_distance         | double     | YES  |     | NULL                |       |
| RatecodeID            | int        | YES  |     | NULL                |       |
| store_and_fwd_flag    | varchar(2) | YES  |     | NULL                |       |
| PULocationID          | 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)
```

8. Next, we ran the below commands to download the yellow taxi csv files from the internet:

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

```
[hadoop@ip-172-31-12-206 ~]$ wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv
--2024-04-09 06:32:52--  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)... 3.5.25.171, 3.5.25.91, 16.182.32.185, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|3.5.25.171|: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 42.3MB/s   in 22s

2024-04-09 06:33:15 (39.0 MB/s) - 'yellow_tripdata_2017-01.csv' saved [914029540/914029540]

[hadoop@ip-172-31-12-206 ~]$ wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-02.csv
--2024-04-09 06:33:19--  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)... 3.5.29.172, 54.231.196.25, 3.5.25.143, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|3.5.29.172|: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 34.0MB/s   in 23s

2024-04-09 06:33:41 (36.4 MB/s) - 'yellow_tripdata_2017-02.csv' saved [863487050/863487050]
```

9. We ran **ls** command and **pwd** command to see the contents and the location of the csv files respectively.

```
[hadoop@ip-172-31-12-206 ~]$ ls
yellow_tripdata_2017-01.csv  yellow_tripdata_2017-02.csv
[hadoop@ip-172-31-12-206 ~]$ pwd
/home/hadoop
```

10. Then we ran the below commands to load the data from the local filesystem of the EMR cluster to the MySQL table of the RDS instance:

LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-01.csv'
INTO TABLE TripData
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 LINES;

LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-02.csv'
INTO TABLE TripData
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 LINES;

```
MySQL [TaxiDB]> LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-01.csv'
    -> INTO TABLE TripData
    -> FIELDS TERMINATED BY ','
    -> LINES TERMINATED BY '\n'
    -> IGNORE 1 LINES;
Query OK, 9710820 rows affected, 65535 warnings (2 min 52.00 sec)
Records: 9710820  Deleted: 0  Skipped: 0  Warnings: 19421640

MySQL [TaxiDB]> LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-02.csv'
    -> INTO TABLE TripData
    -> FIELDS TERMINATED BY ','
    -> LINES TERMINATED BY '\n'
    -> IGNORE 1 LINES;
Query OK, 9169775 rows affected, 65535 warnings (3 min 0.86 sec)
Records: 9169775  Deleted: 0  Skipped: 0  Warnings: 18339550
```

11. Next, we will use the below command to add a column which will play the role of a primary key for this table:

alter table TripData add column TripID INT AUTO_INCREMENT UNIQUE FIRST;

```
MySQL [TaxiDB]> alter table TripData add column TripID INT AUTO_INCREMENT UNIQUE FIRST;
Query OK, 0 rows affected (10 min 56.46 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

12. Next, we again ran the command **desc TripData;** to see the table schema (TripID column got created and added to the table successfully).

```
MySQL [TaxiDB]> desc TripData;
+-----------------------+------------+------+-----+---------------------+----------------+
| Field                 | Type       | Null | Key | Default             | Extra          |
+-----------------------+------------+------+-----+---------------------+----------------+
| TripID                | int        | NO   | PRI | NULL                | auto_increment |
| VendorID              | int        | YES  |     | NULL                |                |
| tpep_pickup_datetime  | timestamp  | NO   |     | 0000-00-00 00:00:00 |                |
| tpep_dropoff_datetime | timestamp  | NO   |     | 0000-00-00 00:00:00 |                |
| passenger_count       | int        | YES  |     | NULL                |                |
| trip_distance         | double     | YES  |     | NULL                |                |
| RatecodeID            | int        | YES  |     | NULL                |                |
| store_and_fwd_flag    | varchar(2) | YES  |     | NULL                |                |
| PULocationID          | 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                |                |
+-----------------------+------------+------+-----+---------------------+----------------+
20 rows in set (0.04 sec)
```

13. Next, we ran the below command to see how many rows of data got inserted (18880595 rows of data got inserted):

    select count(*) from TripData;

```
MySQL [TaxiDB]> select count(*) from TripData;
+----------+
| count(*) |
+----------+
| 18880595 |
+----------+
1 row in set (55.63 sec)
```