MapReduce Assignment

Group Case Study

Prepared By

ATHENI MADHU SUDHAN
ABDULLAH QANEEH
NEETU SINGH

Task 1

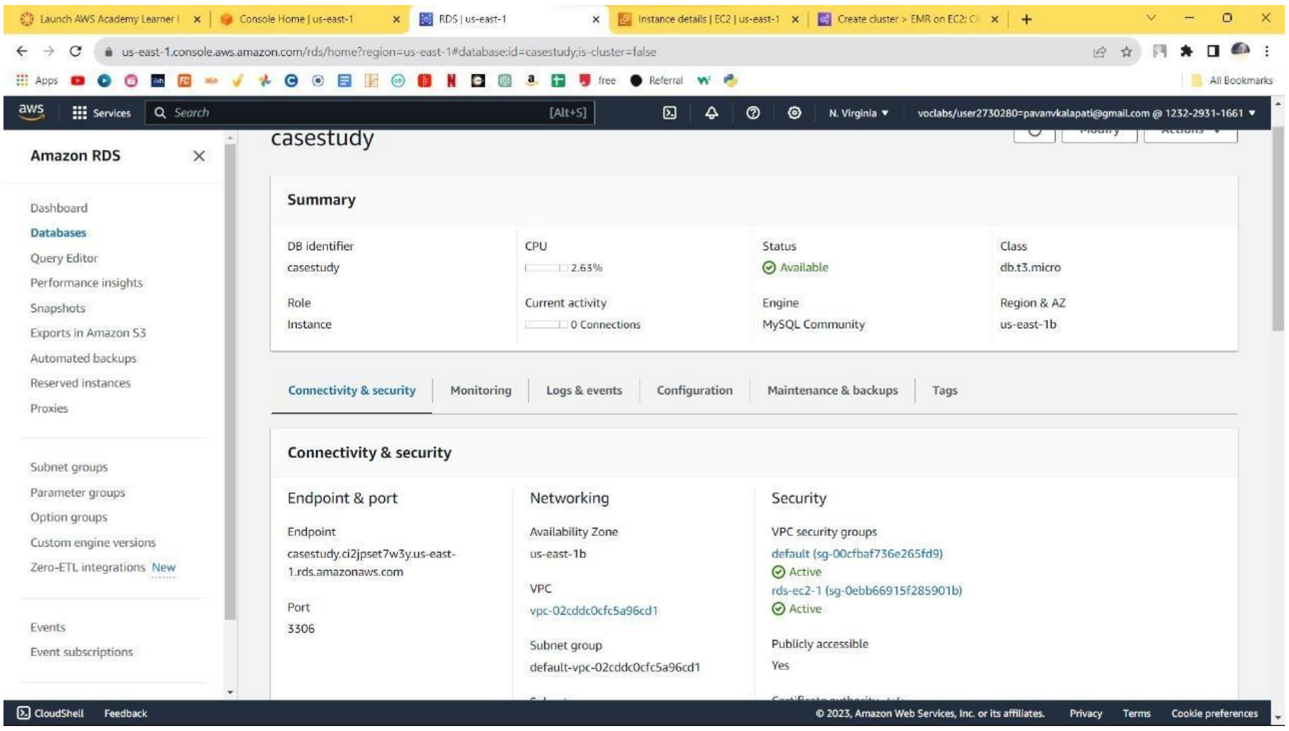
Create an RDS instance in your AWS account and upload the data to the RDS instance

1) We've generated an RDS instance using the Learner Lab and initiated an EMR cluster.

2) We obtained the necessary files for this task, namely yellow_tripdata_2017-01.csv and yellow_tripdata_201702.csv.

3) We've established a connection from EMR to the RDS instance, created a table, and imported records from the CSV files.

4) The data has been successfully loaded into the RDS instance we created.



us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#database:id=casestudy;is-cluster=false

casestudy

Summary

DB identifier	CPU	Status	Class
casestudy	2.63%	Available	db.t3.micro
Role	**Current activity**	**Engine**	**Region & AZ**
Instance	0 Connections	MySQL Community	us-east-1b

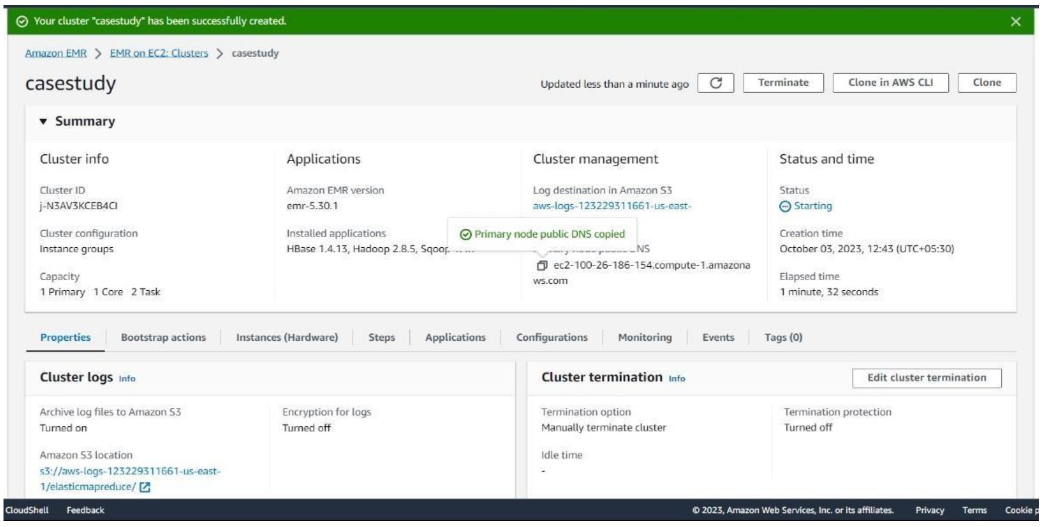
Connectivity & security

Endpoint & port
Endpoint: casestudy.ci2jpset7w3y.us-east-1.rds.amazonaws.com
Port: 3306

Networking
Availability Zone: us-east-1b
VPC: vpc-02cddc0cfc5a96cd1
Subnet group: default-vpc-02cddc0cfc5a96cd1

Security
VPC security groups: default (sg-00cfbaf736e265fd9) Active; rds-ec2-1 (sg-0ebb66915f285901b) Active
Publicly accessible: Yes

EMR Cluster:



Your cluster "casestudy" has been successfully created.

Amazon EMR > EMR on EC2: Clusters > casestudy

casestudy

Updated less than a minute ago

Summary

Cluster info
Cluster ID: j-N3AV3KCEB4CI
Cluster configuration: Instance groups
Capacity: 1 Primary 1 Core 2 Task

Applications
Amazon EMR version: emr-5.30.1
Installed applications: HBase 1.4.13, Hadoop 2.8.5, Sqoop

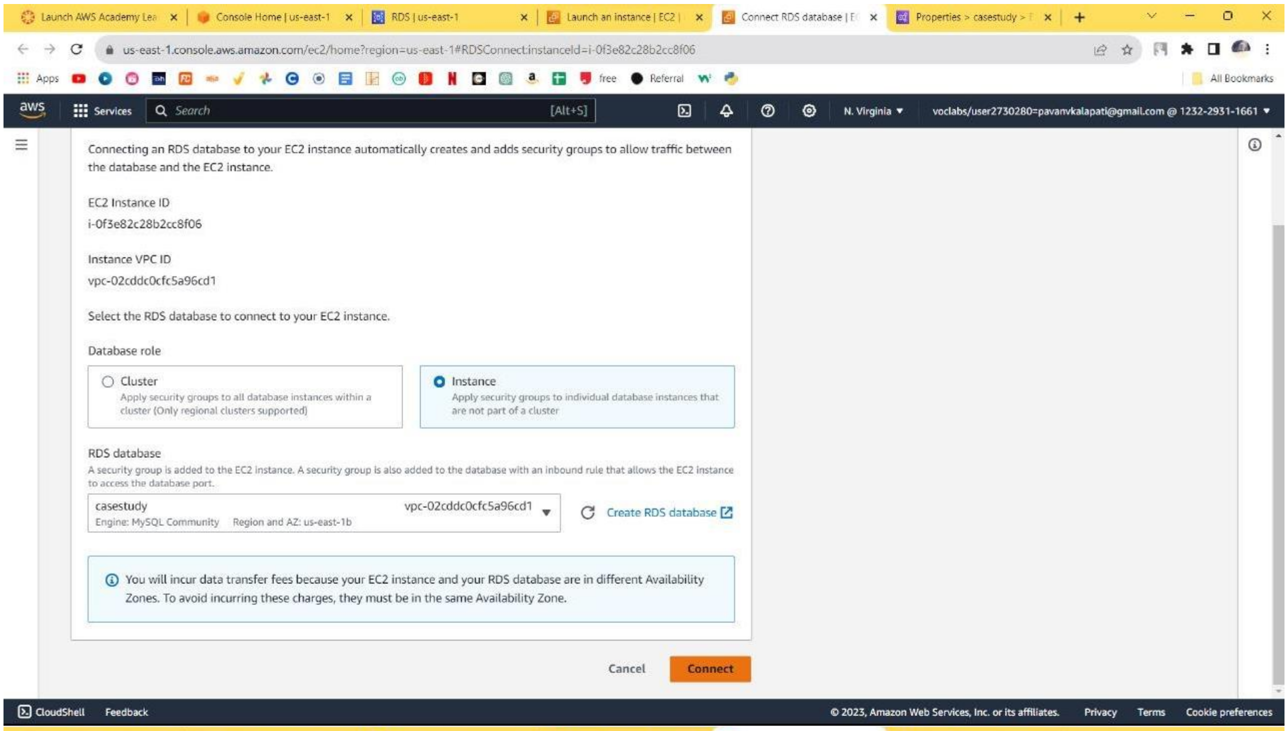
Cluster management
Log destination in Amazon S3: aws-logs-123229311661-us-east-
Primary node public DNS copied
ec2-100-26-186-154.compute-1.amazonaws.com

Status and time
Status: Starting
Creation time: October 03, 2023, 12:43 (UTC+05:30)
Elapsed time: 1 minute, 32 seconds

Cluster logs
Archive log files to Amazon S3: Turned on
Encryption for logs: Turned off
Amazon S3 location: s3://aws-logs-123229311661-us-east-1/elasticmapreduce/

Cluster termination
Termination option: Manually terminate cluster
Termination protection: Turned off
Idle time: -

Connecting RDS with EMR EC2 instance:



Connecting an RDS database to your EC2 instance automatically creates and adds security groups to allow traffic between the database and the EC2 instance.

EC2 Instance ID
i-0f3e82c28b2cc8f06

Instance VPC ID
vpc-02cddc0cfc5a96cd1

Select the RDS database to connect to your EC2 instance.

Database role

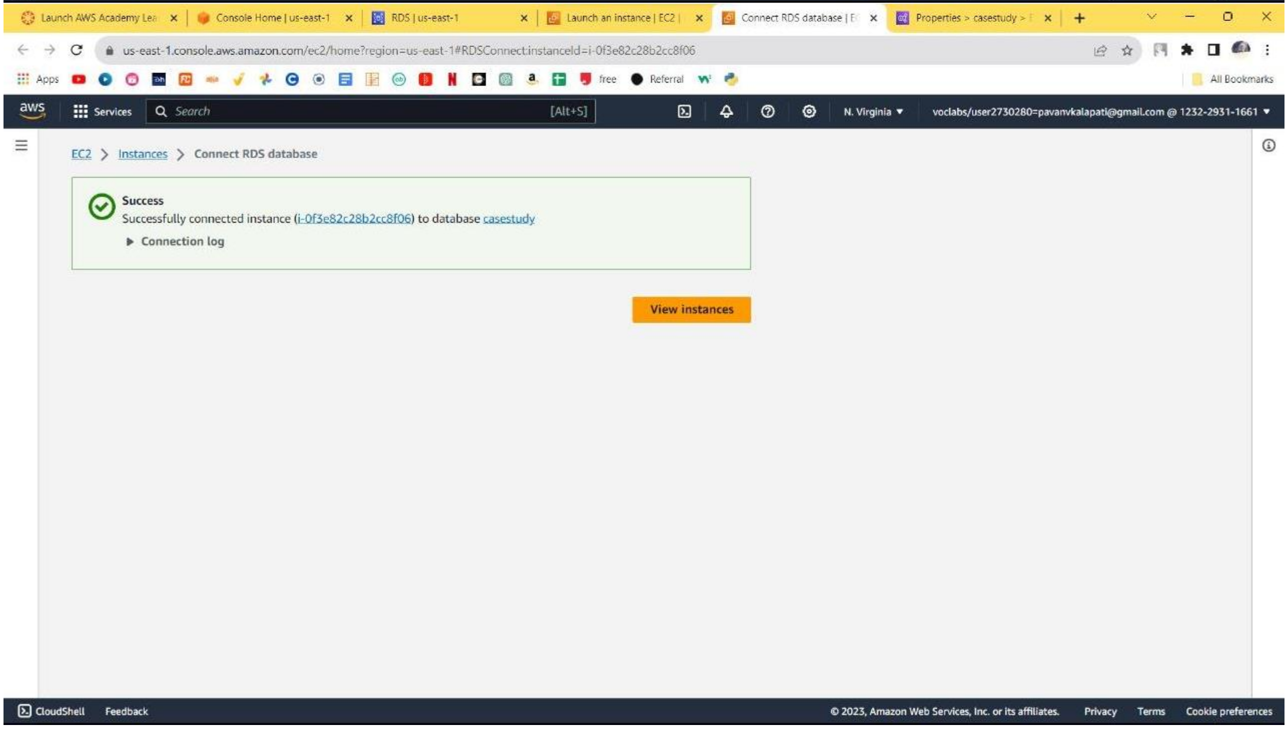
○ Cluster
Apply security groups to all database instances within a cluster (Only regional clusters supported)

● Instance
Apply security groups to individual database instances that are not part of a cluster

RDS database
A security group is added to the EC2 instance. A security group is also added to the database with an inbound rule that allows the EC2 instance to access the database port.

casestudy — vpc-02cddc0cfc5a96cd1
Engine: MySQL Community — Region and AZ: us-east-1b

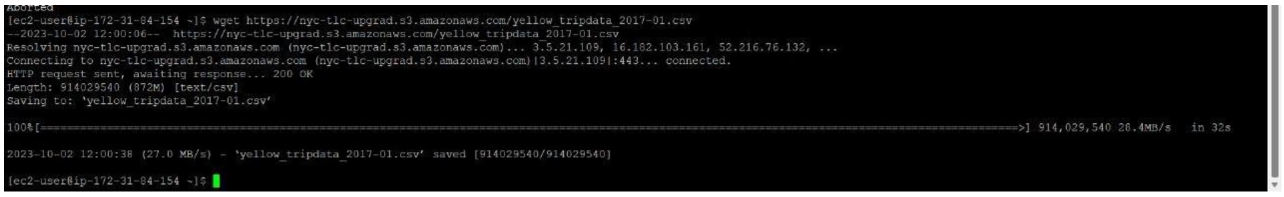
You will incur data transfer fees because your EC2 instance and your RDS database are in different Availability Zones. To avoid incurring these charges, they must be in the same Availability Zone.



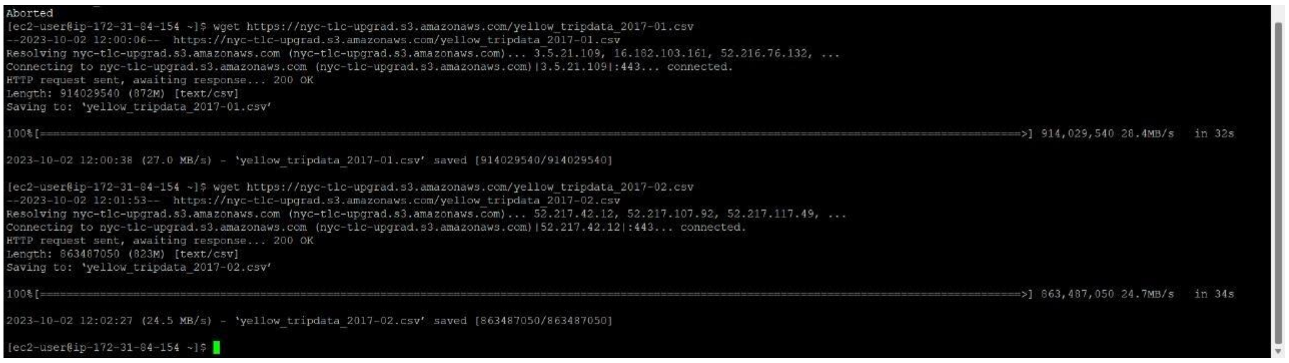
EC2 > Instances > Connect RDS database

Success
Successfully connected instance (i-0f3e82c28b2cc8f06) to database casestudy

Connecting EMR instance with PuTTy and then downloading **yellow_tripdata_2017-01.csv** & **yellow_tripdata_2017-02.csv** :



```
Aborted
[ec2-user@ip-172-31-84-154 ~]$ wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv
--2023-10-02 12:00:06--  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)... 3.5.21.109, 16.182.103.161, 52.216.76.132, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|3.5.21.109|: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 28.4MB/s   in 32s

2023-10-02 12:00:38 (27.0 MB/s) - 'yellow_tripdata_2017-01.csv' saved [914029540/914029540]

[ec2-user@ip-172-31-84-154 ~]$
```

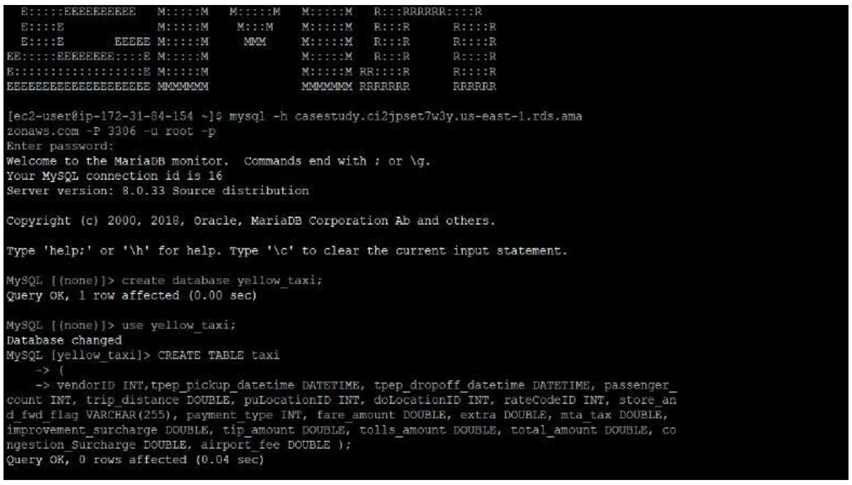


```
Aborted
[ec2-user@ip-172-31-84-154 ~]$ wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv
--2023-10-02 12:00:06--  https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv
Resolving nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)... 3.5.21.109, 16.182.103.161, 52.216.76.132, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|3.5.21.109|: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 28.4MB/s   in 32s

2023-10-02 12:00:38 (27.0 MB/s) - 'yellow_tripdata_2017-01.csv' saved [914029540/914029540]

[ec2-user@ip-172-31-84-154 ~]$ wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-02.csv
--2023-10-02 12:01:53--  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)... 52.217.42.12, 52.217.107.92, 52.217.117.49, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|52.217.42.12|: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 24.7MB/s   in 34s

2023-10-02 12:02:27 (24.5 MB/s) - 'yellow_tripdata_2017-02.csv' saved [863487050/863487050]

[ec2-user@ip-172-31-84-154 ~]$
```

Connecting RDS with EMR Instance:

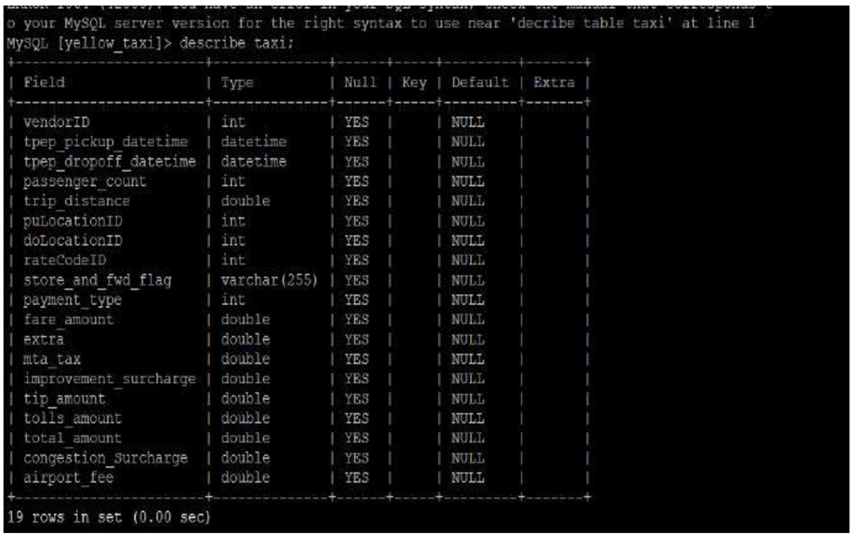
Hostname: casestudy.ci2jpset7w3y.us-east-1.íds.amazonaws.com Command:

mysql -h case-study-dbb.ck4jzoqb1yn7.us-east-1.rds.amazonaws.com -P 3306 -u root -p



```
  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-172-31-84-154 ~]$ mysql -h casestudy.ci2jpset7w3y.us-east-1.rds.ama
zonaws.com -P 3306 -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.33 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)]> create database yellow_taxi;
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> use yellow_taxi;
Database changed
MySQL [yellow_taxi]> CREATE TABLE taxi
    -> (
    -> vendorID INT,tpep_pickup_datetime DATETIME, tpep_dropoff_datetime DATETIME, passenger_
count INT, trip_distance DOUBLE, puLocationID INT, doLocationID INT, rateCodeID INT, store_an
d_fwd_flag VARCHAR(255), payment_type INT, fare_amount DOUBLE, extra DOUBLE, mta_tax DOUBLE,
improvement_surcharge DOUBLE, tip_amount DOUBLE, tolls_amount DOUBLE, total_amount DOUBLE, co
ngestion_Surcharge DOUBLE, airport_fee DOUBLE );
Query OK, 0 rows affected (0.04 sec)
```

```
create database
yellow_taxi;
 use yellow_taxi;
Creating table—
CREATE TABLE taxi
(
vendorID INT,
tpep_pickup_datetime DATETIME,
tpep_dropoff_datetime DATETIME,
passenger_count INT, trip_distance
DOUBLE, puLocationID INT,
doLocationID INT, rateCodeID INT,
store_and_fwd_flag VARCHAR(255),
payment_type INT, fare_amount
DOUBLE, extra DOUBLE, mta_tax
DOUBLE, improvement_surcharge
DOUBLE, tip_amount DOUBLE,

tolls_amount DOUBLE, total_amount
DOUBLE, congestion_Surcharge
DOUBLE, airport_fee DOUBLE
);
```



```
o your MySQL server version for the right syntax to use near 'decribe table taxi' at line 1
MySQL [yellow_taxi]> describe taxi;
+-----------------------+--------------+------+-----+---------+-------+
| Field                 | Type         | Null | Key | Default | Extra |
+-----------------------+--------------+------+-----+---------+-------+
| vendorID              | int          | YES  |     | NULL    |       |
| tpep_pickup_datetime  | datetime     | YES  |     | NULL    |       |
| tpep_dropoff_datetime | datetime     | YES  |     | NULL    |       |
| passenger_count       | int          | YES  |     | NULL    |       |
| trip_distance         | double       | YES  |     | NULL    |       |
| puLocationID          | int          | YES  |     | NULL    |       |
| doLocationID          | int          | YES  |     | NULL    |       |
| rateCodeID            | int          | YES  |     | NULL    |       |
| store_and_fwd_flag    | varchar(255) | YES  |     | NULL    |       |
| payment_type          | int          | YES  |     | NULL    |       |
| fare_amount           | double       | YES  |     | NULL    |       |
| extra                 | double       | YES  |     | NULL    |       |
| mta_tax               | double       | YES  |     | NULL    |       |
| improvement_surcharge | double       | YES  |     | NULL    |       |
| tip_amount            | double       | YES  |     | NULL    |       |
| tolls_amount          | 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)
```

Load Data into Above table

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

INTO TABLE taxi

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n'

IGNORE 1 LINES;

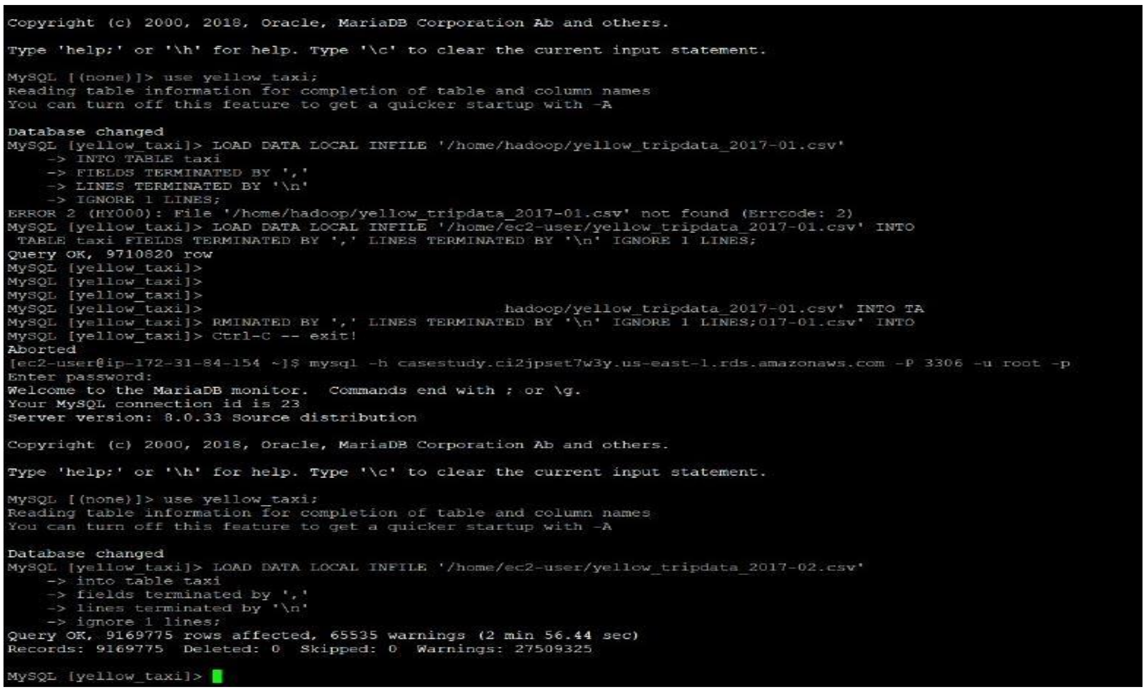
LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-02.csv'

INTO TABLE taxi

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n'

IGNORE 1 LINES;



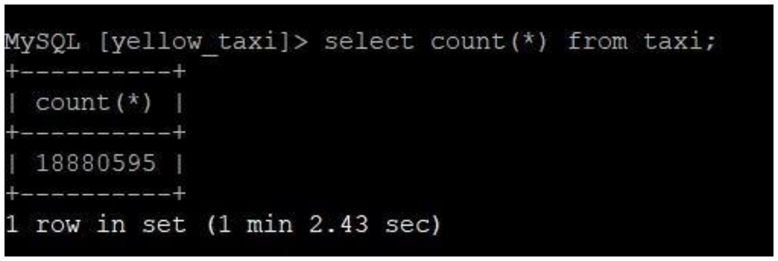
```
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MySQL [(none)]> use yellow_taxi;
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
MySQL [yellow_taxi]> LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-01.csv'
    -> INTO TABLE taxi
    -> FIELDS TERMINATED BY ','
    -> LINES TERMINATED BY '\n'
    -> IGNORE 1 LINES;
ERROR 2 (HY000): File '/home/hadoop/yellow_tripdata_2017-01.csv' not found (Errcode: 2)
MySQL [yellow_taxi]> LOAD DATA LOCAL INFILE '/home/ec2-user/yellow_tripdata_2017-01.csv' INTO
 TABLE taxi FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES;
Query OK, 9710820 row
MySQL [yellow_taxi]>
MySQL [yellow_taxi]>
MySQL [yellow_taxi]>
MySQL [yellow_taxi]>                                        hadoop/yellow_tripdata_2017-01.csv' INTO TA
MySQL [yellow_taxi]> RMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES;017-01.csv' INTO
MySQL [yellow_taxi]> Ctrl-C -- exit!
Aborted
[ec2-user@ip-172-31-84-154 ~]$ mysql -h casestudy.ci2jpset7w3y.us-east-1.rds.amazonaws.com -P 3306 -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 23
Server version: 8.0.33 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)]> use yellow_taxi;
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
MySQL [yellow_taxi]> LOAD DATA LOCAL INFILE '/home/ec2-user/yellow_tripdata_2017-02.csv'
    -> into table taxi
    -> fields terminated by ','
    -> lines terminated by '\n'
    -> ignore 1 lines;
Query OK, 9169775 rows affected, 65535 warnings (2 min 56.44 sec)
Records: 9169775  Deleted: 0  Skipped: 0  Warnings: 27509325

MySQL [yellow_taxi]>
```

TASK 2

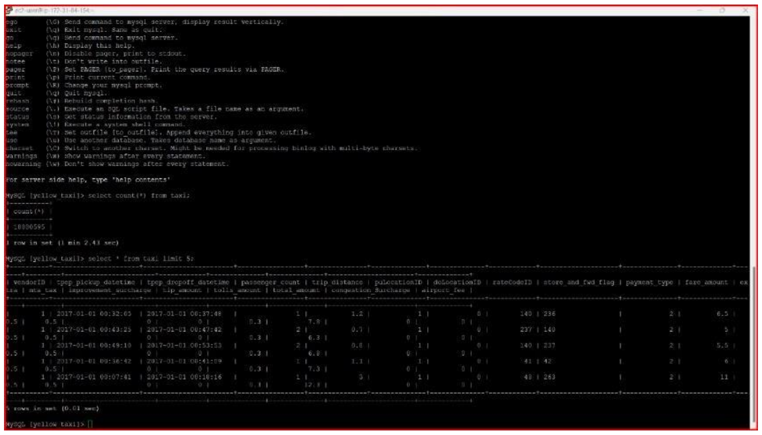
Task 2. Use Sqoop command to ingest the data from RDS into the HBase Table.

The following steps were followed for data ingestion from RDS into HBase table

1) Load the data to RDS instance
Following screen short for reference from previous step (i.e. Task 1)



```
MySQL [yellow_taxi]> select count(*) from taxi;
+----------+
| count(*) |
+----------+
| 18880595 |
+----------+
1 row in set (1 min 2.43 sec)
```



```
ego       (\G) Send command to mysql server, display result vertically.
exit      (\q) Exit mysql. Same as quit.
go        (\g) Send command to mysql server.
help      (\h) Display this help.
nopager   (\n) Disable pager, print to stdout.
notee     (\t) Don't write into outfile.
pager     (\P) Set PAGER [to_pager]. Print the query results via PAGER.
print     (\p) Print current command.
prompt    (\R) Change your mysql prompt.
quit      (\q) Quit mysql.
rehash    (\#) Rebuild completion hash.
source    (\.) Execute an SQL script file. Takes a file name as an argument.
status    (\s) Get status information from the server.
system    (\!) Execute a system shell command.
tee       (\T) Set outfile [to_outfile]. Append everything into given outfile.
use       (\u) Use another database. Takes database name as argument.
charset   (\C) Switch to another charset. Might be needed for processing binlog with multi-byte charsets.
warnings  (\W) Show warnings after every statement.
nowarning (\w) Don't show warnings after every statement.

For server side help, type 'help contents'

MySQL [yellow_taxi]> select count(*) from taxi;
+----------+
| count(*) |
+----------+
| 18880595 |
+----------+
1 row in set (1 min 2.43 sec)

MySQL [yellow_taxi]> select * from taxi limit 5;
| vendorID | tpep_pickup_datetime | tpep_dropoff_datetime | passenger_count | trip_distance | puLocationID | doLocationID | rateCodeID | store_and_fwd_flag | payment_type | fare_amount | ex
tra | mta_tax | improvement_surcharge | tip_amount | tolls_amount | total_amount | congestion_Surcharge | airport_fee |
|        1 | 2017-01-01 00:32:05  | 2017-01-01 00:37:48   |               1 |           1.2 |            1 |            0 |        140 | 236                |            2 |         6.5 |
0.5 |     0.5 |                     0 |          0 |          0.3 |          7.8 |                    0 |           0 |
|        1 | 2017-01-01 00:43:25  | 2017-01-01 00:47:42   |               2 |           0.7 |            1 |            0 |        237 | 140                |            2 |           5 |
0.5 |     0.5 |                     0 |          0 |          0.3 |          6.3 |                    0 |           0 |
|        1 | 2017-01-01 00:49:10  | 2017-01-01 00:53:53   |               2 |           0.8 |            1 |            0 |        140 | 237                |            2 |         5.5 |
0.5 |     0.5 |                     0 |          0 |          0.3 |          6.8 |                    0 |           0 |
|        1 | 2017-01-01 00:36:42  | 2017-01-01 00:41:09   |               1 |           1.1 |            1 |            0 |         41 | 42                 |            2 |           6 |
0.5 |     0.5 |                     0 |          0 |          0.3 |          7.3 |                    0 |           0 |
|        1 | 2017-01-01 00:07:41  | 2017-01-01 00:10:16   |               1 |             3 |            1 |            0 |         48 | 263                |            2 |          11 |
0.5 |     0.5 |                     0 |          0 |          0.3 |         12.3 |                    0 |           0 |
5 rows in set (0.01 sec)

MySQL [yellow_taxi]>
```

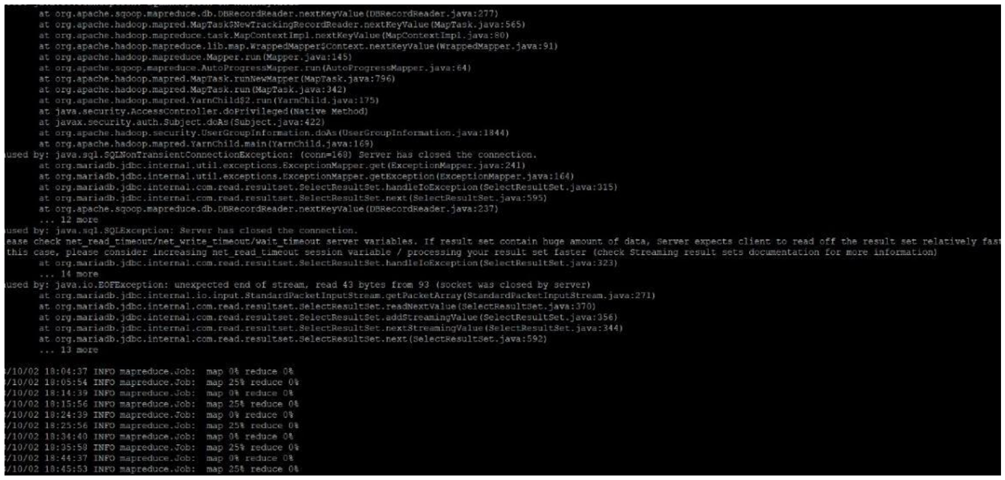
2) Exit form RDS and load the table data in to the hbase.
Created the hbase table : hbasetaxi, set the column-family as cf. Copied the data from the RDS taxi table

Sqoop import command:

```
sqoop import \
  --connect "jdbc:mysql://casestudy.ci2jpset7w3y.us-east-1.rds.amazonaws.com:3306/yellow_taxi" \
  --username root \
  --password 123456789 \
  --table taxi \
  --columns
"vendorID,tpep_pickup_datetime,tpep_dropoff_datetime,passenger_count,trip_distance,puLocationID,doLoc
```

```
ationID,rateCodeID,store_and_fwd_flag,payment_type,fare_amount,extra,mta_tax,improvement_surcharge,t
ip_amount,tolls_amount,total_amount,congestion_Surcharge,airport_fee" \
 --hbase-create-table \
 --hbase-table hbasetaxi \
 --column-family trip_details \
 --hbase-row-key "vendorID,tpep_pickup_datetime,tpep_dropoff_datetime" \
 --split-by tpep_dropoff_datetime \
 -m 8
```

Screenshots for reference:



```
        at org.apache.sqoop.mapreduce.db.DBRecordReader.nextKeyValue(DBRecordReader.java:277)
        at org.apache.hadoop.mapred.MapTask$NewTrackingRecordReader.nextKeyValue(MapTask.java:565)
        at org.apache.hadoop.mapreduce.task.MapContextImpl.nextKeyValue(MapContextImpl.java:80)
        at org.apache.hadoop.mapreduce.lib.map.WrappedMapper$Context.nextKeyValue(WrappedMapper.java:91)
        at org.apache.hadoop.mapreduce.Mapper.run(Mapper.java:145)
        at org.apache.sqoop.mapreduce.AutoProgressMapper.run(AutoProgressMapper.java:64)
        at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:796)
        at org.apache.hadoop.mapred.MapTask.run(MapTask.java:342)
        at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:175)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1844)
        at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:169)
used by: java.sql.SQLNonTransientConnectionException: (conn=160) Server has closed the connection.
        at org.mariadb.jdbc.internal.util.exceptions.ExceptionMapper.get(ExceptionMapper.java:241)
        at org.mariadb.jdbc.internal.util.exceptions.ExceptionMapper.getException(ExceptionMapper.java:164)
        at org.mariadb.jdbc.internal.com.read.resultset.SelectResultSet.handleIoException(SelectResultSet.java:315)
        at org.mariadb.jdbc.internal.com.read.resultset.SelectResultSet.next(SelectResultSet.java:595)
        at org.apache.sqoop.mapreduce.db.DBRecordReader.nextKeyValue(DBRecordReader.java:237)
        ... 12 more
used by: java.sql.SQLException: Server has closed the connection.
ease check net_read_timeout/net_write_timeout/wait_timeout server variables. If result set contain huge amount of data, Server expects client to read off the result set relatively fast
 this case, please consider increasing net_read_timeout session variable / processing your result set faster (check Streaming result sets documentation for more information)
        at org.mariadb.jdbc.internal.com.read.resultset.SelectResultSet.handleIoException(SelectResultSet.java:323)
        ... 14 more
used by: java.io.EOFException: unexpected end of stream, read 43 bytes from 93 (socket was closed by server)
        at org.mariadb.jdbc.internal.io.input.StandardPacketInputStream.getPacketArray(StandardPacketInputStream.java:271)
        at org.mariadb.jdbc.internal.com.read.resultset.SelectResultSet.readNextValue(SelectResultSet.java:370)
        at org.mariadb.jdbc.internal.com.read.resultset.SelectResultSet.addStreamingValue(SelectResultSet.java:356)
        at org.mariadb.jdbc.internal.com.read.resultset.SelectResultSet.nextStreamingValue(SelectResultSet.java:344)
        at org.mariadb.jdbc.internal.com.read.resultset.SelectResultSet.next(SelectResultSet.java:592)
        ... 13 more

/10/02 18:04:37 INFO mapreduce.Job:  map 0% reduce 0%
/10/02 18:05:54 INFO mapreduce.Job:  map 25% reduce 0%
/10/02 18:14:39 INFO mapreduce.Job:  map 0% reduce 0%
/10/02 18:15:56 INFO mapreduce.Job:  map 25% reduce 0%
/10/02 18:24:39 INFO mapreduce.Job:  map 0% reduce 0%
/10/02 18:25:56 INFO mapreduce.Job:  map 25% reduce 0%
/10/02 18:34:40 INFO mapreduce.Job:  map 0% reduce 0%
/10/02 18:35:58 INFO mapreduce.Job:  map 25% reduce 0%
/10/02 18:44:37 INFO mapreduce.Job:  map 0% reduce 0%
/10/02 18:45:53 INFO mapreduce.Job:  map 25% reduce 0%
```

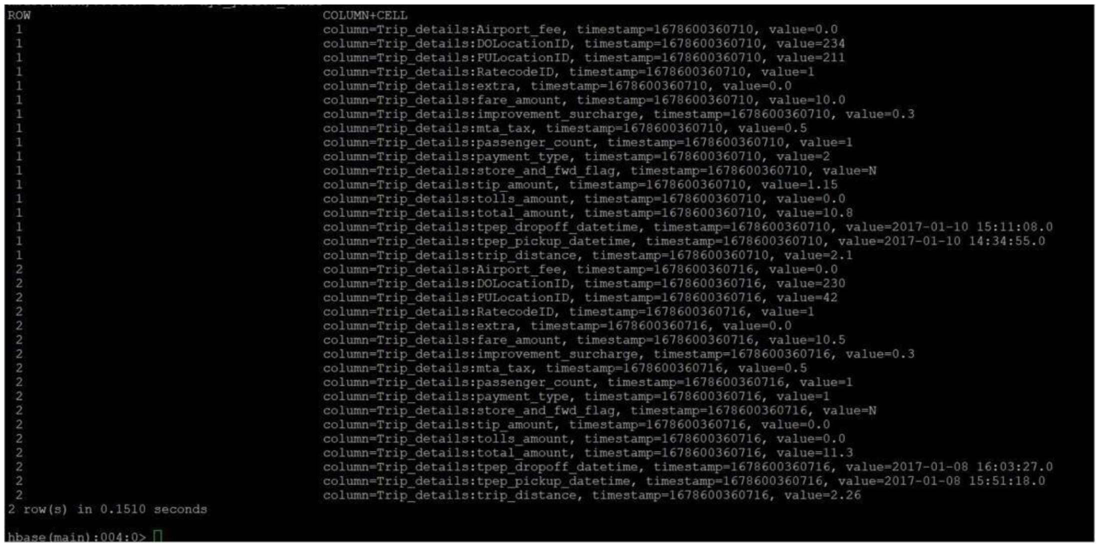
TASK 2

After the mapreduce

Lets check for the hbasetaxi by running

Scan ‘hbasetaxi’

In hbase shell



```
ROW                                          COLUMN+CELL
 1                                           column=Trip_details:Airport_fee, timestamp=1678600360710, value=0.0
 1                                           column=Trip_details:DOLocationID, timestamp=1678600360710, value=234
 1                                           column=Trip_details:PULocationID, timestamp=1678600360710, value=211
 1                                           column=Trip_details:RatecodeID, timestamp=1678600360710, value=1
 1                                           column=Trip_details:extra, timestamp=1678600360710, value=0.0
 1                                           column=Trip_details:fare_amount, timestamp=1678600360710, value=10.0
 1                                           column=Trip_details:improvement_surcharge, timestamp=1678600360710, value=0.3
 1                                           column=Trip_details:mta_tax, timestamp=1678600360710, value=0.5
 1                                           column=Trip_details:passenger_count, timestamp=1678600360710, value=1
 1                                           column=Trip_details:payment_type, timestamp=1678600360710, value=2
 1                                           column=Trip_details:store_and_fwd_flag, timestamp=1678600360710, value=N
 1                                           column=Trip_details:tip_amount, timestamp=1678600360710, value=1.15
 1                                           column=Trip_details:tolls_amount, timestamp=1678600360710, value=0.0
 1                                           column=Trip_details:total_amount, timestamp=1678600360710, value=10.8
 1                                           column=Trip_details:tpep_dropoff_datetime, timestamp=1678600360710, value=2017-01-10 15:11:08.0
 1                                           column=Trip_details:tpep_pickup_datetime, timestamp=1678600360710, value=2017-01-10 14:34:55.0
 1                                           column=Trip_details:trip_distance, timestamp=1678600360710, value=2.1
 2                                           column=Trip_details:Airport_fee, timestamp=1678600360716, value=0.0
 2                                           column=Trip_details:DOLocationID, timestamp=1678600360716, value=230
 2                                           column=Trip_details:PULocationID, timestamp=1678600360716, value=42
 2                                           column=Trip_details:RatecodeID, timestamp=1678600360716, value=1
 2                                           column=Trip_details:extra, timestamp=1678600360716, value=0.0
 2                                           column=Trip_details:fare_amount, timestamp=1678600360716, value=10.5
 2                                           column=Trip_details:improvement_surcharge, timestamp=1678600360716, value=0.3
 2                                           column=Trip_details:mta_tax, timestamp=1678600360716, value=0.5
 2                                           column=Trip_details:passenger_count, timestamp=1678600360716, value=1
 2                                           column=Trip_details:payment_type, timestamp=1678600360716, value=1
 2                                           column=Trip_details:store_and_fwd_flag, timestamp=1678600360716, value=N
 2                                           column=Trip_details:tip_amount, timestamp=1678600360716, value=0.0
 2                                           column=Trip_details:tolls_amount, timestamp=1678600360716, value=0.0
 2                                           column=Trip_details:total_amount, timestamp=1678600360716, value=11.3
 2                                           column=Trip_details:tpep_dropoff_datetime, timestamp=1678600360716, value=2017-01-08 16:03:27.0
 2                                           column=Trip_details:tpep_pickup_datetime, timestamp=1678600360716, value=2017-01-08 15:51:18.0
 2                                           column=Trip_details:trip_distance, timestamp=1678600360716, value=2.26
2 row(s) in 0.1510 seconds

hbase(main):004:0>
```

TASK 3

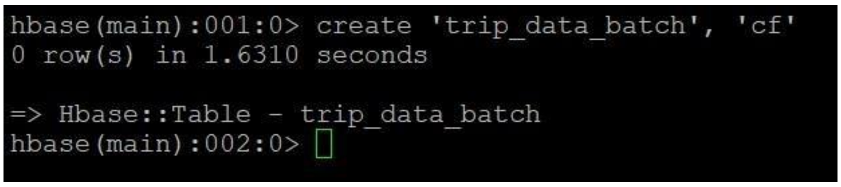
Bulk import data from two files in the dataset on your EMR cluster to your HBase Table using the relevant codes.

1) In this task, our objective is to import data from two CSV files, namely "yellow_tripdata_2017-03.csv" and "yellow_tripdata_2017-04.csv," into an HBase table.

2) We will initiate a new HBase table specifically for this task.

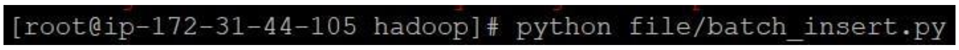
Note: Since the datasets do not contain a primary key, we will need to modify these datasets by adding a primary column ID. Given the large size of the dataset, we will split the files and work with them in smaller portions for further processing.

Create new HBase table –



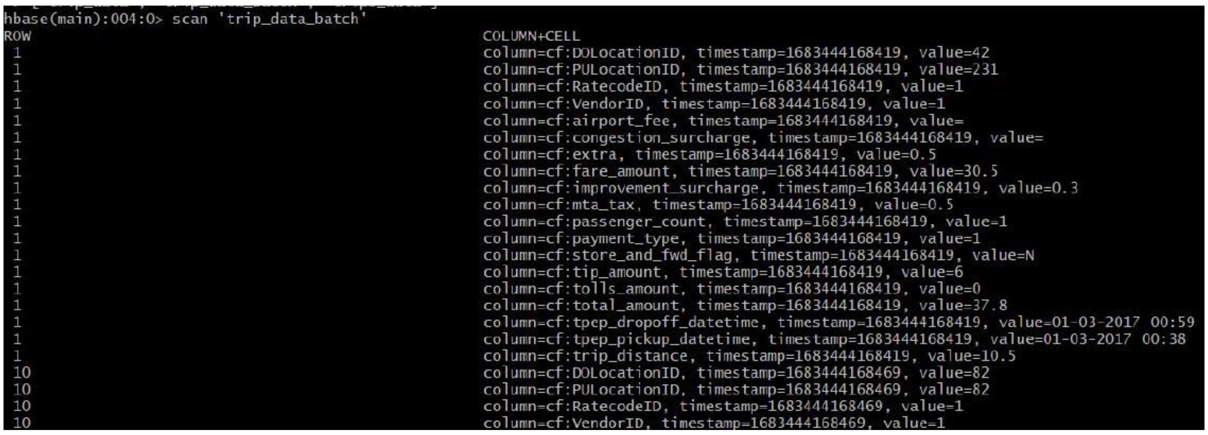
```
hbase(main):001:0> create 'trip_data_batch', 'cf'
0 row(s) in 1.6310 seconds

=> Hbase::Table - trip_data_batch
hbase(main):002:0> 
```

Batch Insert command and Execution –

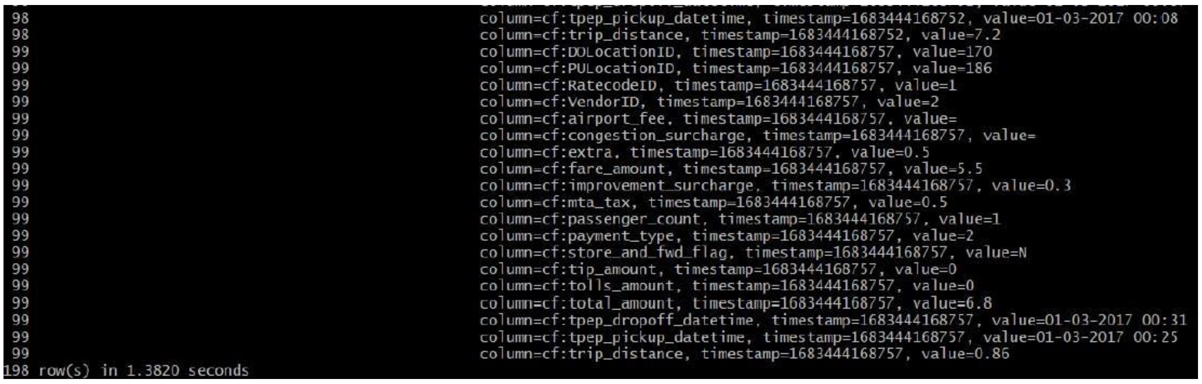


```
[root@ip-172-31-44-105 hadoop]# python file/batch_insert.py
```

HBase Table Records After Import –



```
hbase(main):004:0> scan 'trip_data_batch'
ROW                                                         COLUMN+CELL
 1                                                          column=cf:DOLocationID, timestamp=1683444168419, value=42
 1                                                          column=cf:PULocationID, timestamp=1683444168419, value=231
 1                                                          column=cf:RatecodeID, timestamp=1683444168419, value=1
 1                                                          column=cf:VendorID, timestamp=1683444168419, value=1
 1                                                          column=cf:airport_fee, timestamp=1683444168419, value=
 1                                                          column=cf:congestion_surcharge, timestamp=1683444168419, value=
 1                                                          column=cf:extra, timestamp=1683444168419, value=0.5
 1                                                          column=cf:fare_amount, timestamp=1683444168419, value=30.5
 1                                                          column=cf:improvement_surcharge, timestamp=1683444168419, value=0.3
 1                                                          column=cf:mta_tax, timestamp=1683444168419, value=0.5
 1                                                          column=cf:passenger_count, timestamp=1683444168419, value=1
 1                                                          column=cf:payment_type, timestamp=1683444168419, value=1
 1                                                          column=cf:store_and_fwd_flag, timestamp=1683444168419, value=N
 1                                                          column=cf:tip_amount, timestamp=1683444168419, value=6
 1                                                          column=cf:tolls_amount, timestamp=1683444168419, value=0
 1                                                          column=cf:total_amount, timestamp=1683444168419, value=37.8
 1                                                          column=cf:tpep_dropoff_datetime, timestamp=1683444168419, value=01-03-2017 00:59
 1                                                          column=cf:tpep_pickup_datetime, timestamp=1683444168419, value=01-03-2017 00:38
 1                                                          column=cf:trip_distance, timestamp=1683444168419, value=10.5
 10                                                         column=cf:DOLocationID, timestamp=1683444168469, value=82
 10                                                         column=cf:PULocationID, timestamp=1683444168469, value=82
 10                                                         column=cf:RatecodeID, timestamp=1683444168469, value=1
 10                                                         column=cf:VendorID, timestamp=1683444168469, value=1
```



```
 98                                                         column=cf:tpep_pickup_datetime, timestamp=1683444168752, value=01-03-2017 00:08
 98                                                         column=cf:trip_distance, timestamp=1683444168752, value=7.2
 99                                                         column=cf:DOLocationID, timestamp=1683444168757, value=170
 99                                                         column=cf:PULocationID, timestamp=1683444168757, value=186
 99                                                         column=cf:RatecodeID, timestamp=1683444168757, value=1
 99                                                         column=cf:VendorID, timestamp=1683444168757, value=2
 99                                                         column=cf:airport_fee, timestamp=1683444168757, value=
 99                                                         column=cf:congestion_surcharge, timestamp=1683444168757, value=
 99                                                         column=cf:extra, timestamp=1683444168757, value=0.5
 99                                                         column=cf:fare_amount, timestamp=1683444168757, value=5.5
 99                                                         column=cf:improvement_surcharge, timestamp=1683444168757, value=0.3
 99                                                         column=cf:mta_tax, timestamp=1683444168757, value=0.5
 99                                                         column=cf:passenger_count, timestamp=1683444168757, value=1
 99                                                         column=cf:payment_type, timestamp=1683444168757, value=2
 99                                                         column=cf:store_and_fwd_flag, timestamp=1683444168757, value=N
 99                                                         column=cf:tip_amount, timestamp=1683444168757, value=0
 99                                                         column=cf:tolls_amount, timestamp=1683444168757, value=0
 99                                                         column=cf:total_amount, timestamp=1683444168757, value=6.8
 99                                                         column=cf:tpep_dropoff_datetime, timestamp=1683444168757, value=01-03-2017 00:31
 99                                                         column=cf:tpep_pickup_datetime, timestamp=1683444168757, value=01-03-2017 00:25
 99                                                         column=cf:trip_distance, timestamp=1683444168757, value=0.86
198 row(s) in 1.3820 seconds
```

TASK 3

# connecting to HBase server and opening the table
HBase Table Records After Import –
Code –

1) Create HBase Table

```
create 'trip_data_batch', 'cf'
```

2) Ingest Batch data code –

batch_ingest.py

```
import csv
import happybase
import glob
# Define the HBase table and column family
table_name = 'trip_data_batch'
column_family = 'cf'

# connecting to HBase server and opening the table
connection = happybase.Connection(host='ec2-3-80-189-243.compute-1.amazonaws.com')
table = connection.table(table_name)
csv_dir_path = '/home/hadoop/files/file'
csv_files = glob.glob(csv_dir_path + '/*.csv')

# open CSV file and read the data
for file in csv_files:
        with open(file, 'r') as csvfile:
                reader = csv.DictReader(csvfile)
                for row in reader:
# defining row key and column values for each row
                        row_key = row['ID']
                        column_values = {
                                f'{column_family}:VendorID': row['VendorID'],
                                f'{column_family}:tpep_pickup_datetime':
row['tpep_pickup_datetime'],
                                f'{column_family}:tpep_dropoff_datetime':
row['tpep_dropoff_datetime'],
                                f'{column_family}:passenger_count': row['passenger_count'],
                                f'{column_family}:trip_distance': row['trip_distance'],
                                f'{column_family}:RatecodeID': row['RatecodeID'],
                                f'{column_family}:store_and_fwd_flag': row['store_and_fwd_flag'],
                                f'{column_family}:PULocationID': row['PULocationID'],
                                f'{column_family}:DOLocationID': row['DOLocationID'],
                                f'{column_family}:payment_type': row['payment_type'],
                                f'{column_family}:fare_amount': row['fare_amount'],
                                f'{column_family}:extra': row['extra'],
                                f'{column_family}:mta_tax': row['mta_tax'],
                                        f'{column_family}:tip_amount': row['tip_amount'],
                                        f'{column_family}:tolls_amount': row['tolls_amount'],
                                        f'{column_family}:improvement_surcharge':
row['improvement_surcharge'],
                                        f'{column_family}:total_amount': row['total_amount'],
                                        f'{column_family}:congestion_surcharge':
row['congestion_surcharge'],
                                        f'{column_family}:airport_fee': row['airport_fee']
                                        }
# inserting the row into the HBase table
                              table.put(row_key.encode('utf-8'), column_values)
# Close the connection
connection.close()
```

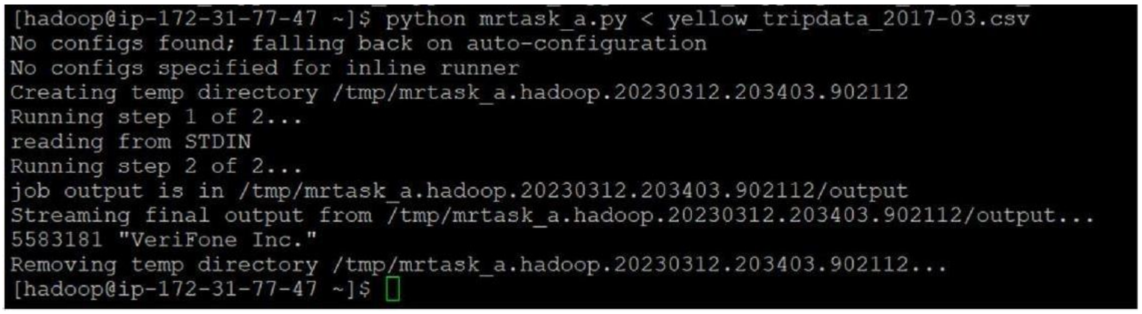
Task 4

Task 4. Write MapReduce codes to perform the tasks using the files you've downloaded on your EMR Instance:

a) Which vendors have the most trips, and what is the total revenue generated by that vendor?

Code File reference: mrtask_a.py

Code execution screenshot:

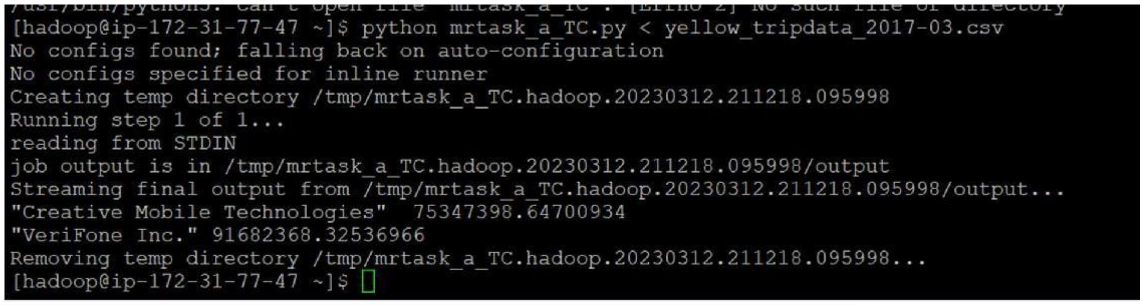


```
[hadoop@ip-172-31-77-47 ~]$ python mrtask_a.py < yellow_tripdata_2017-03.csv
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_a.hadoop.20230312.203403.902112
Running step 1 of 2...
reading from STDIN
Running step 2 of 2...
job output is in /tmp/mrtask_a.hadoop.20230312.203403.902112/output
Streaming final output from /tmp/mrtask_a.hadoop.20230312.203403.902112/output...
5583181 "VeriFone Inc."
Removing temp directory /tmp/mrtask_a.hadoop.20230312.203403.902112...
[hadoop@ip-172-31-77-47 ~]$
```

Comment: VeriFone Inc. holds the highest number of trips in the yellow_tripdata_2017-03.csv dataset, totaling **5,583,181 trips.**

Code File reference:

mrtask_a_TC.py Code execution screenshot:



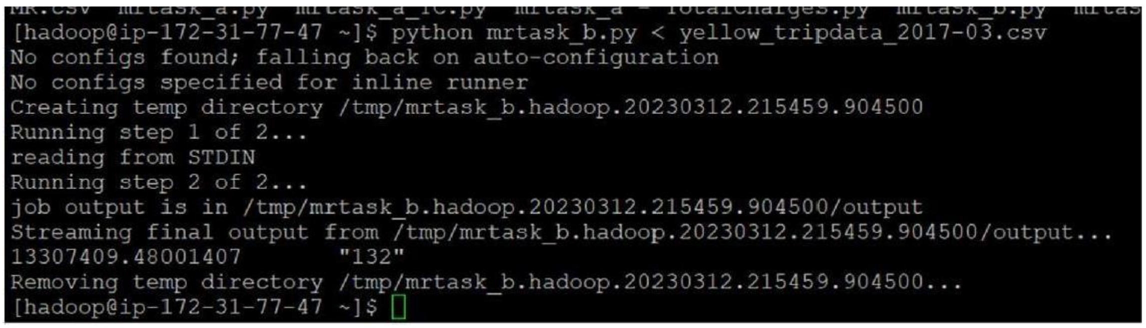
```
[hadoop@ip-172-31-77-47 ~]$ python mrtask_a_TC.py < yellow_tripdata_2017-03.csv
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_a_TC.hadoop.20230312.211218.095998
Running step 1 of 1...
reading from STDIN
job output is in /tmp/mrtask_a_TC.hadoop.20230312.211218.095998/output
Streaming final output from /tmp/mrtask_a_TC.hadoop.20230312.211218.095998/output...
"Creative Mobile Technologies"  75347398.64700934
"VeriFone Inc." 91682368.32536966
Removing temp directory /tmp/mrtask_a_TC.hadoop.20230312.211218.095998...
[hadoop@ip-172-31-77-47 ~]$
```

Comment: VeriFone Inc. has generated a total revenue of **91,682,368.32.**

TASK 4

Code File reference: mrtask_b.py

Code execution screenshot:



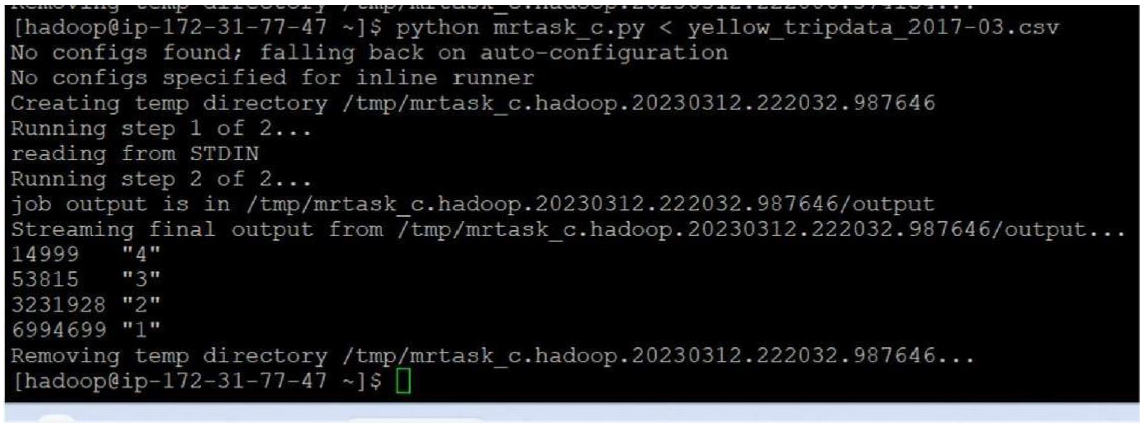
```
MR.csv  mrtask_a.py  mrtask_a_rc.py  mrtask_a - TotalCharges.py  mrtask_b.py  mrtas
[hadoop@ip-172-31-77-47 ~]$ python mrtask_b.py < yellow_tripdata_2017-03.csv
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_b.hadoop.20230312.215459.904500
Running step 1 of 2...
reading from STDIN
Running step 2 of 2...
job output is in /tmp/mrtask_b.hadoop.20230312.215459.904500/output
Streaming final output from /tmp/mrtask_b.hadoop.20230312.215459.904500/output...
13307409.48001407       "132"
Removing temp directory /tmp/mrtask_b.hadoop.20230312.215459.904500...
[hadoop@ip-172-31-77-47 ~]$
```

Remarks: Location 132 generated most revenue in the file tested.

What are the different payment types used by customers and their count? The final results should be in a sorted format.

Code File reference: mrtask_c.py

Code execution screenshot:



```
[hadoop@ip-172-31-77-47 ~]$ python mrtask_c.py < yellow_tripdata_2017-03.csv
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_c.hadoop.20230312.222032.987646
Running step 1 of 2...
reading from STDIN
Running step 2 of 2...
job output is in /tmp/mrtask_c.hadoop.20230312.222032.987646/output
Streaming final output from /tmp/mrtask_c.hadoop.20230312.222032.987646/output...
14999   "4"
53815   "3"
3231928 "2"
6994699 "1"
Removing temp directory /tmp/mrtask_c.hadoop.20230312.222032.987646...
[hadoop@ip-172-31-77-47 ~]$
```

Remark: Credit card is mostly used followed by cash, then no charge and lowest one observed is dispute

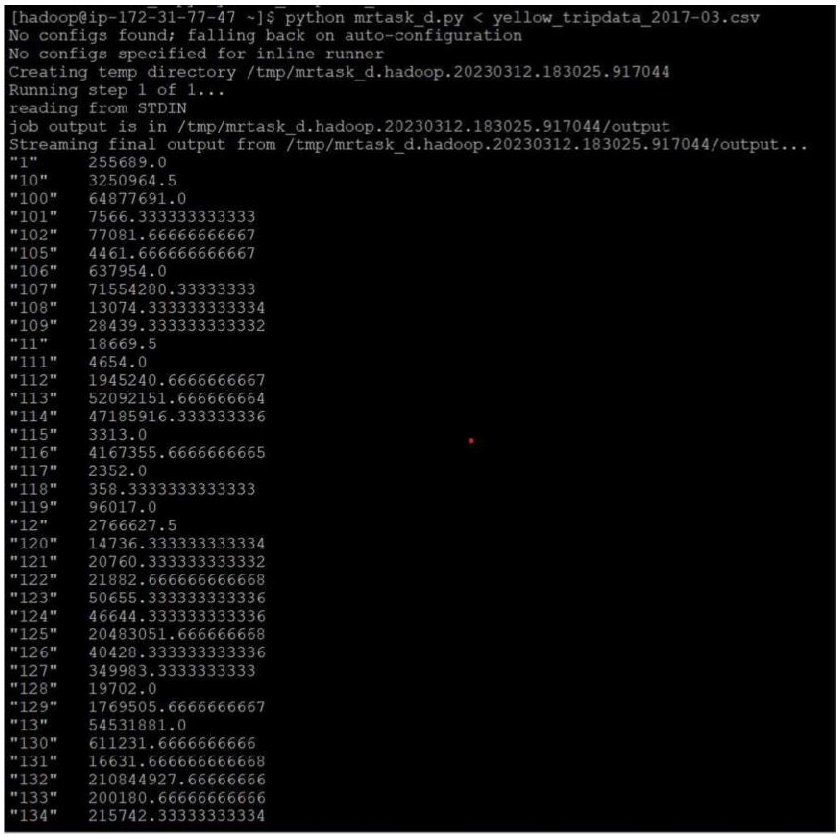
1= Credit card, 2= Cash, 3= No charge,4= Dispute, 5= Unknown, 6= Voided trip

TASK 4

What is the average trip time for different pickup locations?

Code File reference: mrtask_d.py

Code execution screenshot:



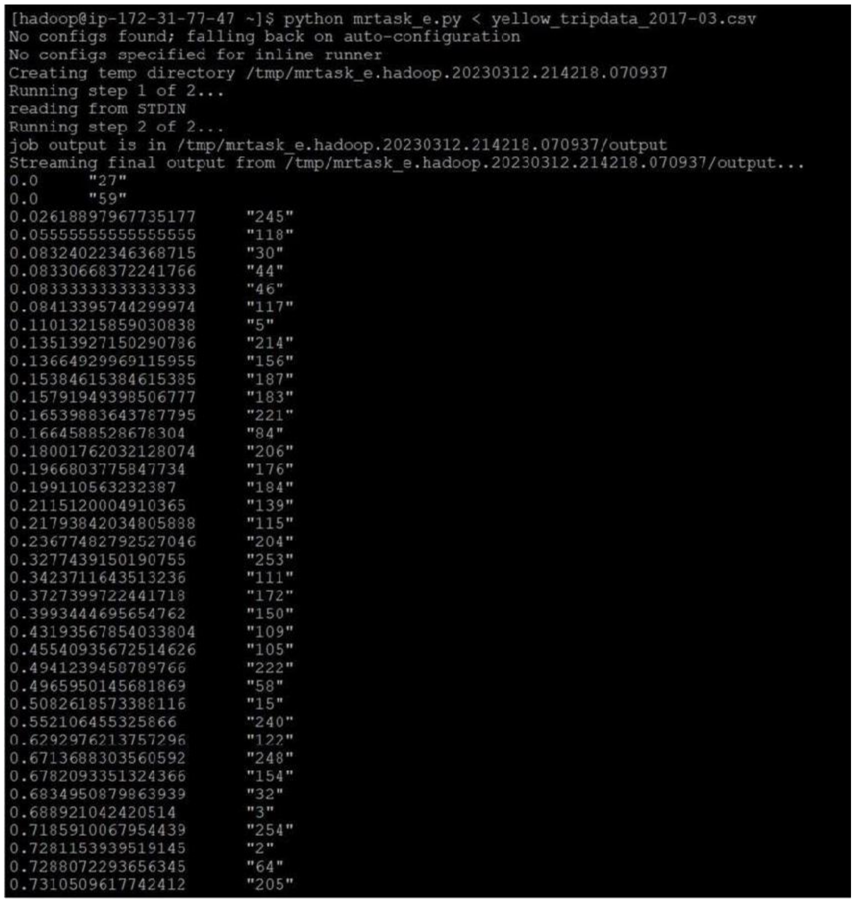
```
[hadoop@ip-172-31-77-47 ~]$ python mrtask_d.py < yellow_tripdata_2017-03.csv
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_d.hadoop.20230312.183025.917044
Running step 1 of 1...
reading from STDIN
job output is in /tmp/mrtask_d.hadoop.20230312.183025.917044/output
Streaming final output from /tmp/mrtask_d.hadoop.20230312.183025.917044/output...
"1"     255689.0
"10"    3250964.5
"100"   64877691.0
"101"   7566.333333333333
"102"   77081.66666666667
"105"   4461.666666666667
"106"   637954.0
"107"   71554280.33333333
"108"   13074.333333333334
"109"   28439.333333333332
"11"    18669.5
"111"   4654.0
"112"   1945240.6666666667
"113"   52092151.666666664
"114"   47185916.333333336
"115"   3313.0
"116"   4167355.6666666665
"117"   2352.0
"118"   358.3333333333333
"119"   96017.0
"12"    2766627.5
"120"   14736.333333333334
"121"   20760.333333333332
"122"   21882.666666666668
"123"   50655.333333333336
"124"   46644.333333333336
"125"   20483051.666666668
"126"   40428.333333333336
"127"   349983.3333333333
"128"   19702.0
"129"   1769505.6666666667
"13"    54531881.0
"130"   611231.6666666666
"131"   16631.666666666668
"132"   210844927.66666666
"133"   200180.66666666666
"134"   215742.33333333334
```

Remarks: Running map reduce job on yellow_tripdata_2017-03.csv. Above screenshot has average trip time in seconds for each location.

Calculate the average tips to revenue ratio of the drivers for different locations in sorted format.

TASK 4

Code File reference: mrtask_e.py



```
[hadoop@ip-172-31-77-47 ~]$ python mrtask_e.py < yellow_tripdata_2017-03.csv
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_e.hadoop.20230312.214218.070937
Running step 1 of 2...
reading from STDIN
Running step 2 of 2...
job output is in /tmp/mrtask_e.hadoop.20230312.214218.070937/output
Streaming final output from /tmp/mrtask_e.hadoop.20230312.214218.070937/output...
0.0     "27"
0.0     "59"
0.02618897967735177     "245"
0.05555555555555555     "118"
0.08324022346368715     "30"
0.08330668372241766     "44"
0.08333333333333333     "46"
0.08413395744299974     "117"
0.11013215859030838     "5"
0.13513927150290786     "214"
0.13664929969115955     "156"
0.15384615384615385     "187"
0.15791949398506777     "183"
0.16539883643787795     "221"
0.1664588528678304      "84"
0.18001762032128074     "206"
0.1966803775847734      "176"
0.199110563232387       "184"
0.2115120004910365      "139"
0.21793842034805888     "115"
0.23677482792527046     "204"
0.3277439150190755      "253"
0.3423711643513236      "111"
0.3727399722441718      "172"
0.3993444695654762      "150"
0.43193567854033804     "109"
0.45540935672514626     "105"
0.4941239458789766      "222"
0.4965950145681869      "58"
0.5082618573388116      "15"
0.552106455325866       "240"
0.6292976213757296      "122"
0.6713688303560592      "248"
0.6782093351324366      "154"
0.6834950879863939      "32"
0.688921042420514       "3"
0.7185910067954439      "254"
0.7281153939519145      "2"
0.7288072293656345      "64"
0.7310509617742412      "205"
```

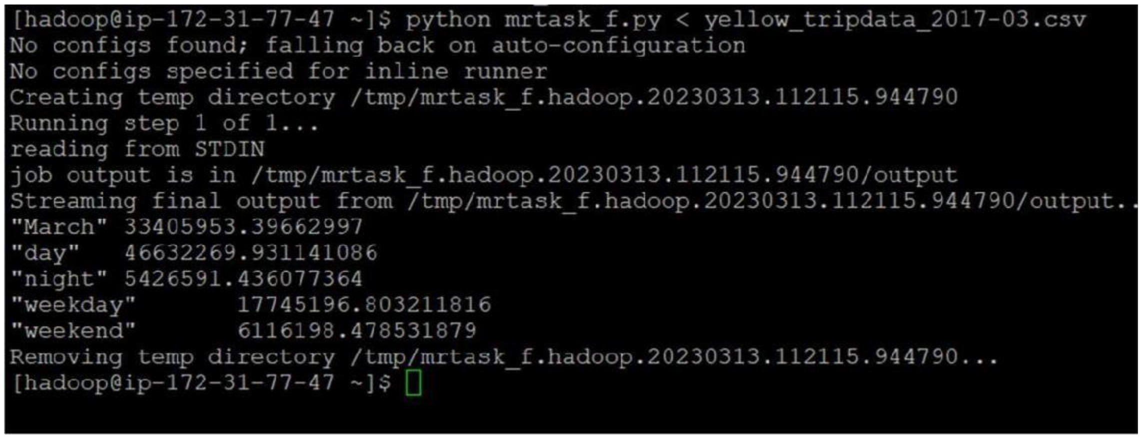
Comment: The screenshot above was taken during the testing of the code designed to calculate the average tip-to-revenue ratio for drivers in various locations.

Furthermore, the analysis aims to understand how revenue changes over time by calculating the average trip revenue per month. This analysis is carried out by considering the time of day (day versus night) and the day of the week (weekday versus weekend) as factors for comparison.

TASK 4

Code File reference: mrtask_f.py

Code execution screenshot:



```
[hadoop@ip-172-31-77-47 ~]$ python mrtask_f.py < yellow_tripdata_2017-03.csv
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mrtask_f.hadoop.20230313.112115.944790
Running step 1 of 1...
reading from STDIN
job output is in /tmp/mrtask_f.hadoop.20230313.112115.944790/output
Streaming final output from /tmp/mrtask_f.hadoop.20230313.112115.944790/output..
"March" 33405953.39662997
"day"   46632269.931141086
"night" 5426591.436077364
"weekday"       17745196.803211816
"weekend"       6116198.478531879
Removing temp directory /tmp/mrtask_f.hadoop.20230313.112115.944790...
[hadoop@ip-172-31-77-47 ~]$
```

Observations/Findings:

1. The dataset predominantly consists of data for the month of March.

2. The average revenue for trips in March can be calculated.

3. On average, trip revenue during daytime hours exceeds that of nighttime hours. We assume nighttime to be between 11 PM and 5 AM.

4. The dataset suggests that the average trip revenue during weekdays surpasses that of weekends, with Saturday and Sunday considered as weekends.