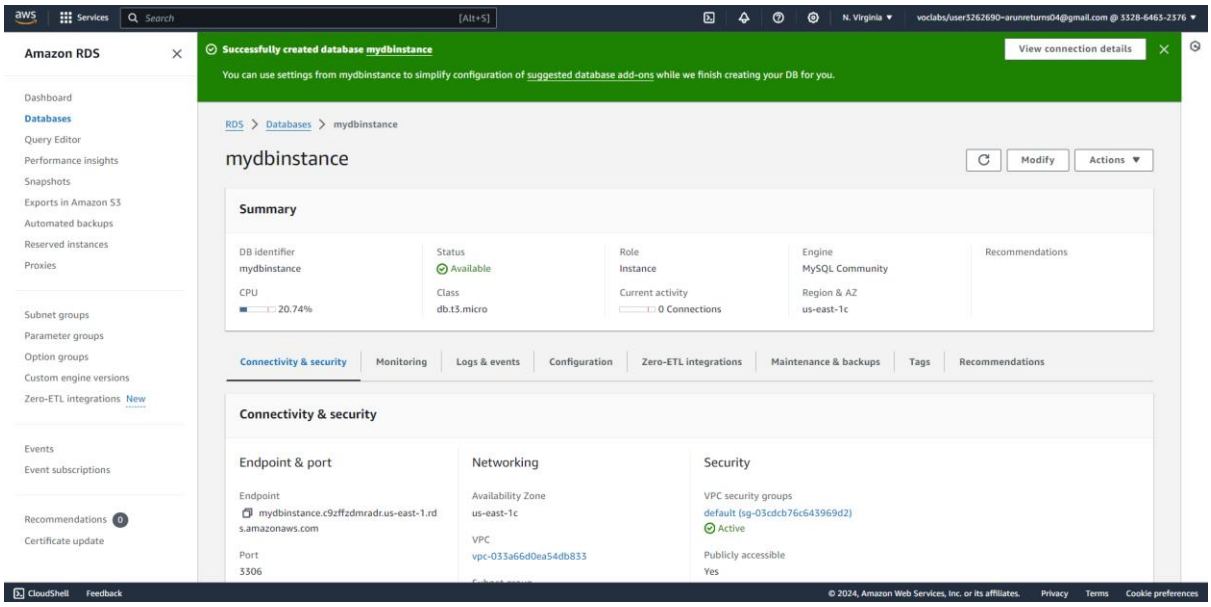
TASK-1:

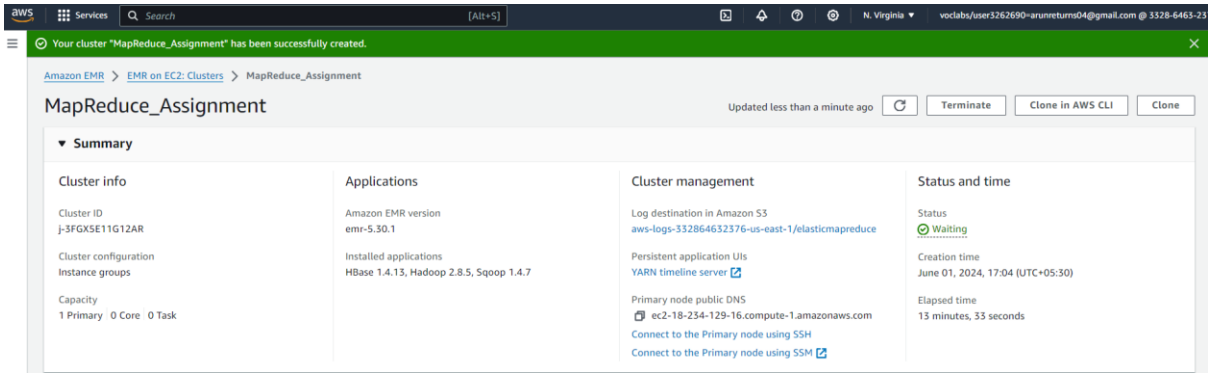
Create an RDS Instance in your AWS account and upload the data to the instance

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

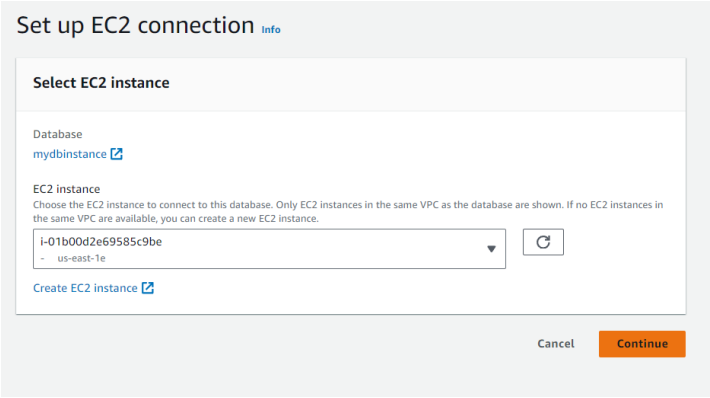
1) Creation of RDS instance in AWS



2) Creation of EMR instance with bundled applications such as Hadoop, Hbase, Sqoop



3) To connect RDS with EMR instance, we have to click on "Action" button on RDS instance menu and then "Set up an EC2 Instance".



4) Login to RDS through EMR instance using command:

```
mysql -h mydbinstance.c9zffzdmradr.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
```

```
[hadoop@ip-172-31-61-251 ~]$  mysql -h mydbinstance.c9zffzdmradr.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 20
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)]> |
```

5) Creation of Database "taxi_records" and table "trip_log"

```
CREATE DATABASE taxi_records;
USE taxi_records;

CREATE TABLE trip_log (VendorID INT, tpep_pickup_datetime VARCHAR(50), tpep_dropoff_datetime
VARCHAR(50), Passenger_count INT, Trip_distance FLOAT, RatecodeID INT, store_and_fwd_flag
VARCHAR(2), 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 );
```

```
MySQL [(none)]> CREATE DATABASE taxi_records
    -> ;
Query OK, 1 row affected (0.01 sec)
```

```
MySQL [(none)]> use taxi_records;
Database changed
MySQL [taxi_records]> CREATE TABLE trip_log (VendorID INT, tpep_pickup_datetime VARCHAR(50), tpep_dropoff_datetime VARCH
AR(50), Passenger_count INT, Trip_distance FLOAT, RatecodeID INT, store_and_fwd_flag VARCHAR(2), PULocationID INT, DOLoc
ationID INT, payment_type INT, fare_amount FLOAT, extra FLOAT, mta_tax FLOAT, tip_amount FLOAT, tolls_amount FLOAT, impr
ovement_surcharge FLOAT, total_amount FLOAT, Airport_fee FLOAT );
Query OK, 0 rows affected (0.04 sec)
```

6) Downloading required csv files from internet in local using command

```
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-172-31-61-251 ~]$ wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv
_tripdata_2017-02.csv
--2024-06-01 20:58:04--  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)... 16.182.104.57, 52.216.10.11, 52.216.44.65
, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|16.182.104.57|: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 45.4MB/s   in 21s

2024-06-01 20:58:24 (41.9 MB/s) - 'yellow_tripdata_2017-01.csv' saved [914029540/914029540]

[hadoop@ip-172-31-61-251 ~]$ wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-02.csv
--2024-06-01 20:58:24--  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)... 3.5.28.211, 16.182.32.209, 52.216.215.113
, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|3.5.28.211|: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 47.5MB/s   in 20s

2024-06-01 20:58:44 (41.5 MB/s) - 'yellow_tripdata_2017-02.csv' saved [863487050/863487050]
```

7) To load data in mysql table we have to login and then run sql command:

```
mysql -h mydbinstance.c9zffzdmradr.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
USE taxi_records;
LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-01.csv'
INTO TABLE trip_log
FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES;

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

SELECT COUNT(*) FROM taxi_records.trip_log;
```

```
MySQL [(none)]> use taxi_records;
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 [taxi_records]> LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-01.csv'
    -> INTO TABLE trip_log
    -> FIELDS TERMINATED BY ','
    -> LINES TERMINATED BY '\n'
    -> IGNORE 1 LINES;
Query OK, 9710820 rows affected, 65535 warnings (2 min 36.05 sec)
Records: 9710820  Deleted: 0  Skipped: 0  Warnings: 9710820

MySQL [taxi_records]> LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-02.csv'
    -> INTO TABLE trip_log
ATED    -> FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES;

Query OK, 9169775 rows affected, 65535 warnings (2 min 56.83 sec)
Records: 9169775  Deleted: 0  Skipped: 0  Warnings: 9169775
```

```
MySQL [taxi_records]> SELECT COUNT(*) FROM taxi_records.trip_log;
+----------+
| COUNT(*) |
+----------+
| 18880595 |
+----------+
1 row in set (50.88 sec)
```