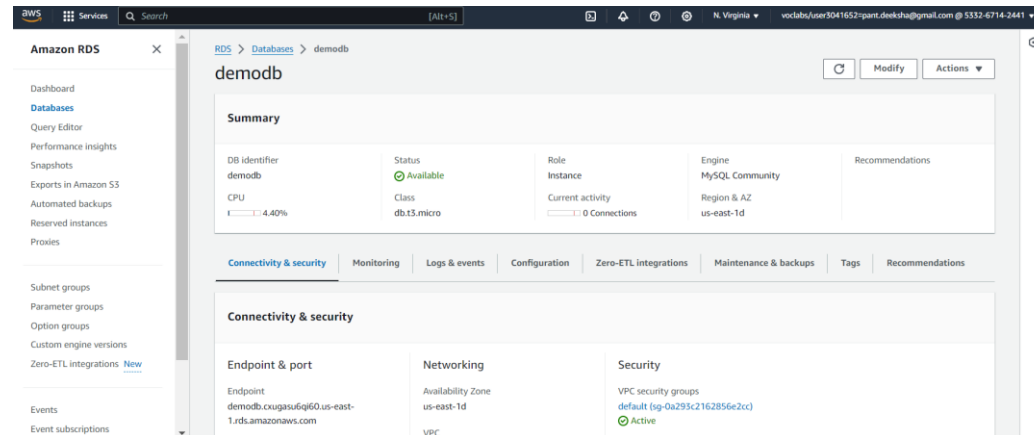


Task 1.

Create an RDS instance in AWS account and upload the data to the RDS instance.
Since the dataset is huge, uploaded the data from only two files
(i.e. yellow_tripdata_2017-01.csv & yellow_tripdata_2017-02.csv) from the dataset.
Created an appropriate schema before uploading the data sets to AWS RDS

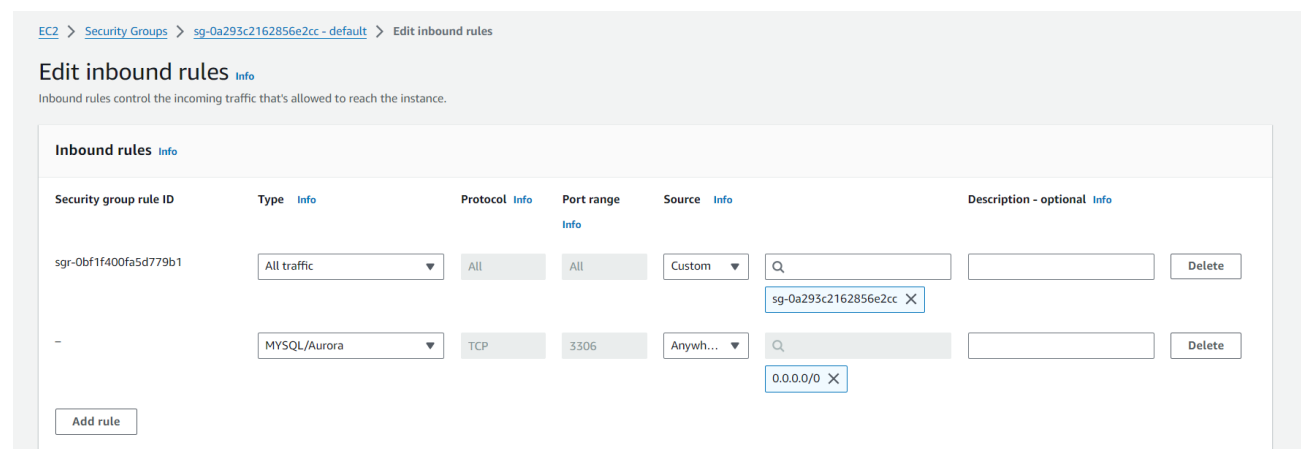
Created an RDS instance in AWS account



Database Name: demodb

User: admin

Updated the IP address under the Security Groups to access the RDS instance.
Provided access to the 3306 port to the required IP address.



Created EMR cluster & configured Security group->inbound rules

Name and applications - required Info
Name your cluster and choose the applications that you want to install on your cluster.

Name

Amazon EMR release Info
 A release contains a set of applications which can be installed on your cluster.

Application bundle

Spark

Core Hadoop

HBase

Presto

Custom

☐ Flink 1.10.0
☐ HCatalog 2.3.6
☐ Hue 4.6.0
☐ MXNet 1.5.1
☐ Phoenix 4.14.3
☐ Spark 2.4.5
☐ Tez 0.9.2

☐ Ganglia 3.7.2
☒ Hadoop 2.8.5
☐ JupyterHub 1.1.0
☐ Mahout 0.13.0
☐ Pig 0.17.0
☒ Sqoop 1.4.7
☐ Zeppelin 0.8.2

☐ HBase 1.4.13
☐ Hive 2.3.6
☐ Livy 0.7.0
☐ Oozie 5.2.0
☐ Presto 0.232
☐ TensorFlow 1.14.0
☐ ZooKeeper 3.4.14

HBase storage settings
 Choose the storage layer for your data stored in HBase. The HDFS option uses the HBase default location for the root directory.
☒ Hadoop distributed file system (HDFS)

Summary Info

Name and applications - required

Name
 MapReduce_EMR

Amazon EMR release
 emr-5.30.1

Application bundle
 Custom (HBase 1.4.13, Hadoop 2.8.5, Sqoop 1.4.7)

Cluster configuration - required

Instance groups
 Primary (m4.xlarge)

Networking - required

VPC
[vpc-01b085492...](#)

Subnet

Amazon EMR > EMR on EC2 Clusters > MapReduce_EMR

Updated less than a minute ago Terminate Clone in AWS CLI Clone

MapReduce_EMR

Summary

Cluster info Cluster ID j-QYH0W5DZM0K 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-533267142441-us-east-1/elasticmapreduce Persistent application URL VARN timeline server Primary node public DNS ec2-18-232-95-243.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 29 February 2024 11:35 (UTC+04:00) Elapsed time 12 minutes, 39 seconds
---	--	---	--

Properties | Bootstrap actions | Instances (hardware) | Steps | Applications | Configurations | Monitoring | Events | Tags (0)

Cluster logs Info

Archive log files to Amazon S3
 Turned on

Amazon S3 location
[s3://aws-logs-533267142441-us-east-1/elasticmapreduce](#)

Turn on encryption for logs
 Turned off

Cluster termination Info

Termination option
 Manually terminate cluster

Termination protection
 Off

Primary Node DNS: ec2-18-232-95-243.compute-1.amazonaws.com

Connected RDS instance to EC2

Amazon RDS

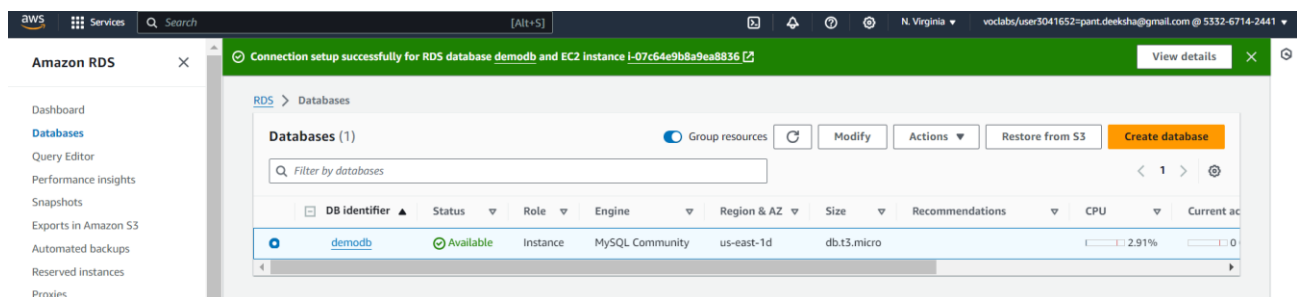
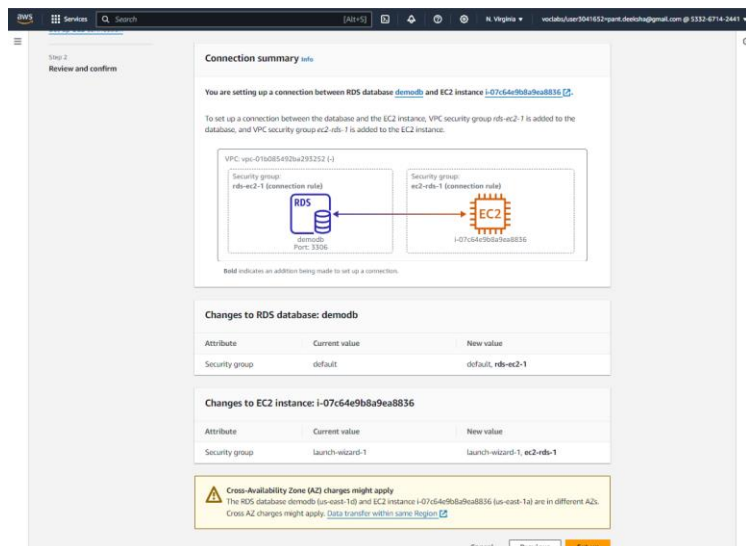
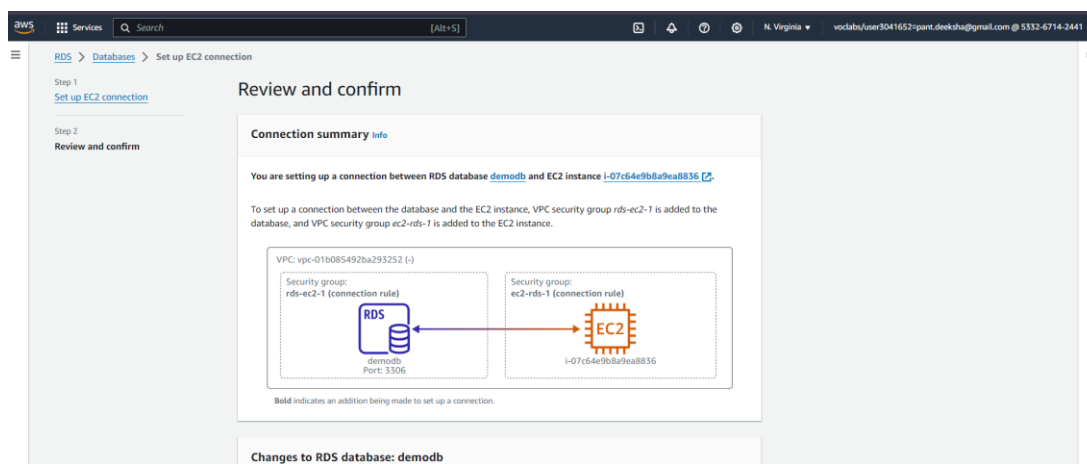
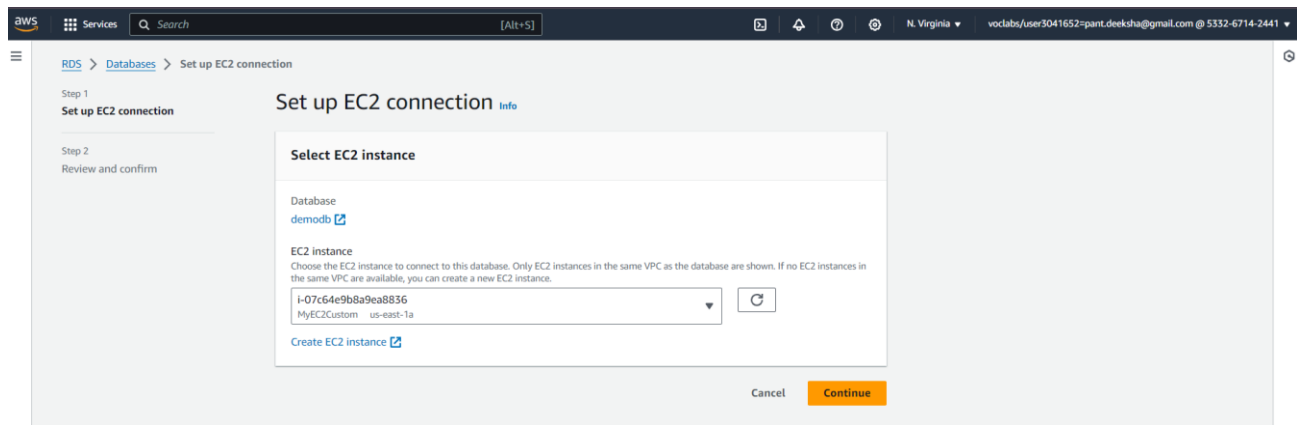
Dashboard
Databases
 Query Editor
 Performance insights
 Snapshots
 Exports in Amazon S3
 Automated backups
 Reserved instances
 Proxies

Databases (1) Group resources Modify

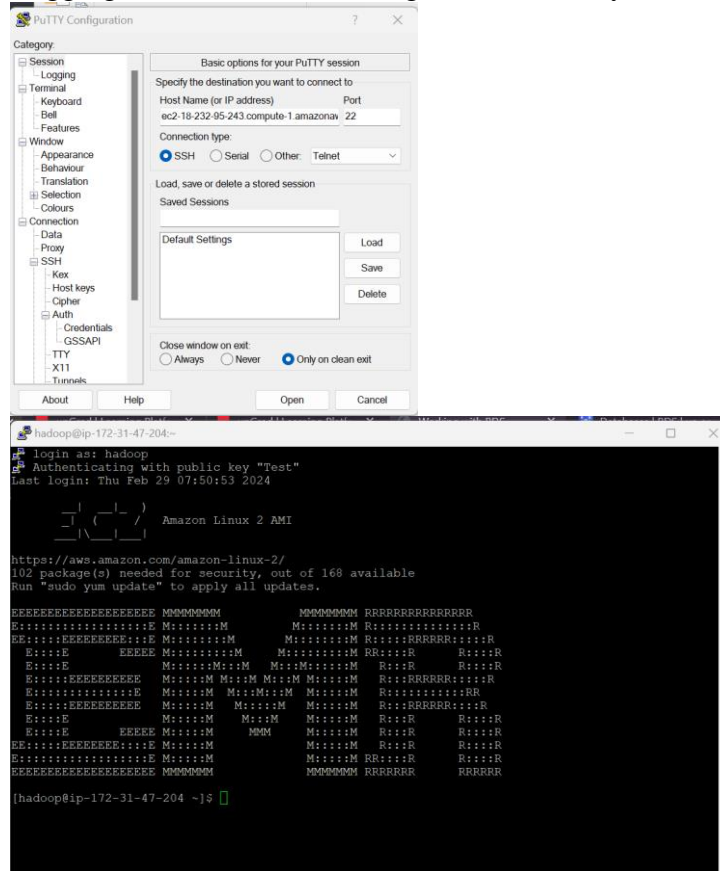
DB identifier	Status	Role	Engine
demodb	Available	Instance	MySQL

Actions

- Restore from S3
- Create database
- Quick Actions - New
 - Convert to Multi-AZ deployment
 - Stop temporarily
 - Reboot
 - Delete
 - Set up EC2 connection
 - Set up Lambda connection
 - Create read replica
 - Create Aurora read replica
 - Create Blue/Green Deployment - new



Logging to EMR Cluster through SSH via Putty

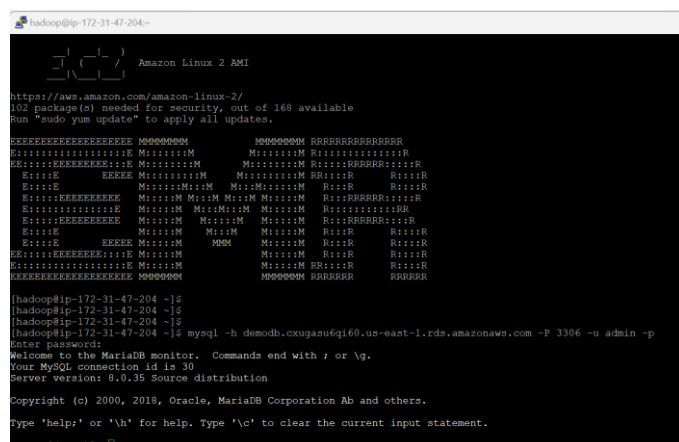


Connecting to RDS:

Command used:

```
mysql -h demodb.cxugasu6qi60.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
```

After entering this command, entered the password



Created a database in RDS:

```

hadoop@ip-172-31-47-204:~
E:::::E M:::M M:::M M:::M R:::RR
E:::::EEEEEEEE M:::M M:::M M:::M R:::RRRRR:::R
E:::E M:::M M:::M M:::M R:::R R:::R
E:::E EEEEE M:::M M M:::M R:::R R:::R
E:::::EEEEEEEE:::E M:::M M:::M R:::R R:::R
E:::::EEEEEEEE:::E M:::M M:::M RR:::R R:::R
EEEEEEEEEEEEEEEE MMMMMM MMMMMM RRRRRR RRRRR

[hadoop@ip-172-31-47-204 ~]$
[hadoop@ip-172-31-47-204 ~]$
[hadoop@ip-172-31-47-204 ~]$
[hadoop@ip-172-31-47-204 ~]$ mysql -h demodb.cxugasu6qi60.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 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)]> create database NYC_yellowtaxi;
Query OK, 1 row affected (0.01 sec)

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

MySQL [(none)]>

```

Created a table structure in RDS:

```

create table yellow_tripdata (
VendorID INT,
tpep_pickup_datetime VARCHAR(255),
tpep_dropoff_datetime VARCHAR(255),
Passenger_count INT,
Trip_distance FLOAT,
RatecodeID INT,
store_and_fwd_flag VARCHAR(50),
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
);

```

```
hadoop@ip-172-31-47-204:~
MySQL [NYC_yellowtaxi]>
MySQL [NYC_yellowtaxi]> create table yellow_tripdata (
-> VendorID INT,
-> tpep_pickup_datetime VARCHAR(255),
-> tpep_dropoff_datetime VARCHAR(255),
-> Passenger_count INT,
-> Trip_distance FLOAT,
-> RatecodeID INT,
-> store_and_fwd_flag VARCHAR(50),
-> 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
-> );
Query OK, 0 rows affected (0.03 sec)
MySQL [NYC_yellowtaxi]>
```

Downloaded the necessary data on Hadoop local file system.

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-172-31-47-204:~
Bye
[hadoop@ip-172-31-47-204 ~]$
[hadoop@ip-172-31-47-204 ~]$
[hadoop@ip-172-31-47-204 ~]$
[hadoop@ip-172-31-47-204 ~]$ wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv
--2024-02-29 08:15:38-- 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)... 16.182.41.185, 54.231.169.177, 3.5.28.190, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)[16.182.41.185]: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 23.0MB/s in 39s

2024-02-29 08:16:18 (22.3 MB/s) - 'yellow_tripdata_2017-01.csv' saved [914029540/914029540]

[hadoop@ip-172-31-47-204 ~]$
[hadoop@ip-172-31-47-204 ~]$
[hadoop@ip-172-31-47-204 ~]$ wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-02.csv
--2024-02-29 08:16:47-- 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)... 52.217.102.36, 52.217.227.105, 52.216.211.177, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)[52.217.102.36]: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 23.7MB/s in 38s

2024-02-29 08:17:25 (21.5 MB/s) - 'yellow_tripdata_2017-02.csv' saved [863487050/863487050]

[hadoop@ip-172-31-47-204 ~]$
[hadoop@ip-172-31-47-204 ~]$
[hadoop@ip-172-31-47-204 ~]$ ls
yellow_tripdata_2017-01.csv yellow_tripdata_2017-02.csv
[hadoop@ip-172-31-47-204 ~]$
```

Loaded data in MySQL table with following commands:

```
LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-01.csv'
INTO TABLE yellow_tripdata
FIELDS TERMINATED BY ','
```

```
MySQL [NYC_yellowtaxi]>
MySQL [NYC_yellowtaxi]>
MySQL [NYC_yellowtaxi]> LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-01.csv'
-> INTO TABLE yellow_tripdata
-> FIELDS TERMINATED BY ','
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES;
Query OK, 9710820 rows affected, 65535 warnings (2 min 4.06 sec)
Records: 9710820 Deleted: 0 Skipped: 0 Warnings: 9710820
```

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

Query OK, 9169775 rows affected, 65535 warnings (2 min 6.43 sec)
Records: 9169775 Deleted: 0 Skipped: 0 Warnings: 9169775
```

[illegible]