

## Task 1:

Setting up an RDS instance, establishing a database, creating tables within the RDS instance, and importing CSV files into the RDS instance. Additionally, initiating the creation of an EMR cluster.

**Step1:** Initiating an EMR cluster named **MapReduce-EMR** with a root file system size of 40GB:

The screenshot shows the AWS Management Console for the 'MapReduce-EMR' cluster. The cluster is in a 'Waiting' status. The page includes a summary section with the following details:

- Cluster info:** Cluster ID: j-2CH3HJSZ1Y0EB, 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, Hive 2.3.6, Hue 4.6.0, Spark 2.4.5, Sqoop 1.4.7.
- Cluster management:** Log destination in Amazon S3: aws-logs-339713178912-us-east-1/elasticmapreduce, Persistent application UIs: Spark History Server, YARN timeline server, Tez UI, Primary node public DNS: ec2-3-234-243-60.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: February 05, 2024, 16:22 (UTC+05:30), Elapsed time: 12 minutes, 32 seconds.

The screenshot shows the AWS Management Console for the 'MapReduce-EMR' cluster. The page includes sections for 'Cluster logs', 'Cluster termination', and 'Network and security'.

- Cluster logs:** Archive log files to Amazon S3: Turned on, Amazon S3 location: s3://aws-logs-339713178912-us-east-1/elasticmapreduce/, Encryption for logs: Turned off.
- Cluster termination:** Termination option: Manually terminate cluster, Termination protection: Turned on, Idle time: -.
- Network and security:** Network: Virtual Private Cloud (VPC): vpc-034aad4fc7cf5b380, Subnet(s) and Availability Zone(s) (AZ): subnet-0069360a41588e284, Security configuration: Security configuration: None, EC2 key pair: NewKey2, Permissions: Service role for Amazon EMR: EMR\_DefaultRole, EC2 instance profile: EMR\_EC2\_DefaultRole, Custom automatic scaling role: Not configured.

## Step 2: Creating an RDS instance **mapreduce-rds** with MySQL, making it publicly accessible and enabling MySQL port in the security group:

The screenshot shows the AWS Management Console for an RDS instance named 'mapreduce-rds' in the 'us-east-1' region. The instance is a MySQL Community Edition instance, currently 'Available' with 0 connections. The CPU usage is at 29.24%. The instance is located in the 'us-east-1a' Availability Zone. The endpoint is 'mapreduce-rds.cv9g4e4op16.us-east-1.rds.amazonaws.com'. The VPC security group is 'default (sg-085c695a4671eac5f)' and is 'Active'.

Summary				
DB identifier mapreduce-rds	Status Available	Role Instance	Engine MySQL Community	Recommendations
CPU 29.24%	Class db.t3.micro	Current activity 0 Connections	Region & AZ us-east-1a	

Connectivity & security | Monitoring | Logs & events | Configuration | Zero-ETL integrations | Maintenance & backups | Tags | Recommendations

Connectivity & security		
Endpoint & port Endpoint mapreduce-rds.cv9g4e4op16.us-east-1.rds.amazonaws.com	Networking Availability Zone us-east-1a	Security VPC security groups default (sg-085c695a4671eac5f) Active

## Step 3: Logging in to EMR using Putty:

```
hadoop@ip-172-31-70-33:~
login as: hadoop
Authenticating with public key "NewKey2"
Last login: Mon Feb  5 11:18:34 2024

 _ _ | _ _ | _ _ )
 _ _ | ( _ _ | _ _ /
 _ _ | \ _ _ | _ _ |

https://aws.amazon.com/amazon-linux-2/
94 package(s) needed for security, out of 161 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRRRRRRRRRRR
E::::::::::::::::::::E M::::::::M           M::::::::M R:::::::::R
EE::::::::EEEEEEEE::E M::::::::M           M::::::::M R::::RRRRRR:::R
  E::::E      EEEEE M::::::::M           M::::::::M RR::::R      R::::R
  E::::E      M::::M M::M M::M M::M M::M M::M R::R      R::::R
  E::::EEEEEEEEEE M::::M M::M M::M M::M M::M R::RRRRRR:::R
  E::::::::::::E M::::M M::M M::M M::M M::M R:::::::::RR
  E::::EEEEEEEEEE M::::M M::M M::M M::M M::M R::RRRRRR:::R
  E::::E      M::::M M::M 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

[hadoop@ip-172-31-70-33 ~]$ sudo yum update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.6 kB 00:00:00
14 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
```

#### Step 4: Fetching necessary CSV files from the internet and saving them locally:

Use 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-172-31-70-33:~$ wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv
--2024-02-05 11:28:45-- 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)... 54.231.235.145, 5
4.231.236.161, 3.5.19.1, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|54.231.235.145|:
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 38.7MB/s in 24s

2024-02-05 11:29:10 (35.8 MB/s) - 'yellow_tripdata_2017-01.csv' saved [914029540/914029540]

[hadoop@ip-172-31-70-33 ~]$ wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-02.csv
--2024-02-05 11:29:35-- 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)... 52.216.57.161, 52.216.186.235, 52.217.123.225, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|52.216.57.161|: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 36.1MB/s in 23s

2024-02-05 11:29:58 (35.9 MB/s) - 'yellow_tripdata_2017-02.csv' saved [863487050/863487050]
```

#### Step 5: Establishing a connection between an EMR cluster and an RDS instance, along with executing MySQLWorkbench commands:

Use command:

```
mysql -h mapreduce-rds.cvg0g4e4op16.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
```

```
hadoop@ip-172-31-70-33:~$ mysql -h mapreduce-rds.cvg0g4e4op16.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 32
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)]>
```

Use command:

**create database <database\_name>;** : To create new database  
**show databases;** : To view list of databases  
**use <database\_name>;** : To go to the database  
**show tables;** : To view list of tables in a database

```
hadoop@ip-172-31-70-33:~  
[hadoop@ip-172-31-70-33 ~]$ mysql -h mapreduce-rds.cvg0g4e4op16.us-east-1.rds.am  
azonaws.com -P 3306 -u admin -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MySQL connection id is 32  
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_taxi;  
Query OK, 1 row affected (0.00 sec)  
  
MySQL [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| nyc_taxi |  
| performance_schema |  
| sys |  
+-----+  
5 rows in set (0.00 sec)  
  
MySQL [(none)]> use nyc_taxi;  
Database changed  
MySQL [nyc_taxi]> show tables;  
Empty set (0.00 sec)  
  
MySQL [nyc_taxi]> █
```

To create new table in database **nyc\_taxi**, use command:

```
create table NYC_TRIPS  
(  
  VendorID INT,  
  tpep_pickup_datetime DATETIME,  
  tpep_dropoff_datetime DATETIME,  
  passenger_count INT,  
  trip_distance FLOAT(10,2),  
  RatecodeID INT,  
  store_and_fwd_flag VARCHAR(1),  
  PULocationID VARCHAR(50),  
  DOLocationID VARCHAR(50),  
  payment_type INT,  
  fare_amount FLOAT(10,2),  
  extra FLOAT(10,2),  
  mta_tax FLOAT(10,2),  
  tip_amount FLOAT(10,2),  
  tolls_amount FLOAT(10,2),  
  improvement_surcharge FLOAT(10,2),  
  total_amount FLOAT(10,2),  
  Airport_fee FLOAT(10,2)  
);
```

```
hadoop@ip-172-31-70-33:~
MySQL [nyc_taxi]> create table NYC_TRIPS
-> (
-> VendorID INT,
-> tpep_pickup_datetime DATETIME,
-> tpep_dropoff_datetime DATETIME,
-> passenger_count INT,
-> trip_distance FLOAT(10,2),
-> RatecodeID INT,
-> store_and_fwd_flag VARCHAR(1),
-> PULocationID VARCHAR(50),
-> DOLocationID VARCHAR(50),
-> payment_type INT,
-> fare_amount FLOAT(10,2),
-> extra FLOAT(10,2),
-> mta_tax FLOAT(10,2),
-> tip_amount FLOAT(10,2),
-> tolls_amount FLOAT(10,2),
-> improvement_surcharge FLOAT(10,2),
-> total_amount FLOAT(10,2),
-> Airport_fee FLOAT(10,2)
-> );
Query OK, 0 rows affected, 9 warnings (0.02 sec)

MySQL [nyc_taxi]> show tables;
+-----+
| Tables_in_nyc_taxi |
+-----+
| NYC_TRIPS           |
+-----+
1 row in set (0.00 sec)

MySQL [nyc_taxi]>
```

**Step 6:** Within the RDS instance, importing the datasets 'yellow\_tripdata\_2017-01.csv' and 'yellow\_tripdata\_2017-02.csv' into the pre-existing 'NYC\_TRIPS' table within the 'nyc\_taxi' database in MySQL:

Use commands:

LOAD DATA LOCAL INFILE '/home/hadoop/yellow\_tripdata\_2017-01.csv' INTO TABLE NYC\_TRIPS FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES;

LOAD DATA LOCAL INFILE '/home/hadoop/yellow\_tripdata\_2017-02.csv' INTO TABLE NYC\_TRIPS FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES;

```
hadoop@ip-172-31-70-33:~
-> );
Query OK, 0 rows affected, 9 warnings (0.02 sec)

MySQL [nyc_taxi]> show tables;
+-----+
| Tables_in_nyc_taxi |
+-----+
| NYC_TRIPS           |
+-----+
1 row in set (0.00 sec)

MySQL [nyc_taxi]> LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-01.csv' INTO TABLE NYC_TRIPS FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES;
Query OK, 9710820 rows affected, 65535 warnings (2 min 8.57 sec)
Records: 9710820 Deleted: 0 Skipped: 0 Warnings: 9710820
MySQL [nyc_taxi]> LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-02.csv' INTO TABLE NYC_TRIPS FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES;
Query OK, 9169775 rows affected, 65535 warnings (2 min 10.11 sec)
Records: 9169775 Deleted: 0 Skipped: 0 Warnings: 9169775

MySQL [nyc_taxi]>
```

To check data in database, use command:

Select \* from NYC\_TRIPS LIMIT 10; (To view data in table)

```
hadoop@ip-172-31-70-33:~
MySQL [nyc_taxi]> Select * from NYC_TRIPS LIMIT 10;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 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 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 2017-01-01 00:32:05 | 2017-01-01 00:37:48 | 1 | 1.20 | 1 | N | 140 | 236 | 2 | 6.50 | 0.50 | 0.50 | 0.00 | 0.00 |
| 1 | 2017-01-01 00:43:25 | 2017-01-01 00:47:42 | 2 | 0.70 | 1 | N | 237 | 140 | 2 | 5.00 | 0.50 | 0.50 | 0.00 | 0.00 |
| 1 | 2017-01-01 00:49:10 | 2017-01-01 00:53:53 | 2 | 0.80 | 1 | N | 140 | 237 | 2 | 5.50 | 0.50 | 0.50 | 0.00 | 0.00 |
| 1 | 2017-01-01 00:36:42 | 2017-01-01 00:41:09 | 1 | 1.10 | 1 | N | 41 | 42 | 2 | 6.00 | 0.50 | 0.50 | 0.00 | 0.00 |
| 1 | 2017-01-01 00:07:41 | 2017-01-01 00:18:16 | 1 | 3.00 | 1 | N | 48 | 263 | 2 | 11.00 | 0.50 | 0.50 | 0.00 | 0.00 |
| 1 | 2017-01-01 00:20:52 | 2017-01-01 00:24:59 | 2 | 0.70 | 1 | N | 236 | 262 | 2 | 5.00 | 0.50 | 0.50 | 0.00 | 0.00 |
| 1 | 2017-01-01 00:33:49 | 2017-01-01 00:42:38 | 2 | 1.60 | 1 | N | 236 | 238 | 1 | 8.00 | 0.50 | 0.50 | 1.85 | 0.00 |
| 1 | 2017-01-01 00:48:22 | 2017-01-01 00:52:15 | 2 | 0.60 | 1 | N | 238 | 239 | 1 | 5.00 | 0.50 | 0.50 | 1.25 | 0.00 |
| 1 | 2017-01-01 00:57:12 | 2017-01-01 01:06:28 | 2 | 1.00 | 1 | N | 239 | 48 | 1 | 7.50 | 0.50 | 0.50 | 1.75 | 0.00 |
| 1 | 2017-01-01 00:10:25 | 2017-01-01 00:29:06 | 1 | 1.00 | 1 | N | 246 | 48 | 2 | 12.00 | 0.50 | 0.50 | 0.00 | 0.00 |
10 rows in set (0.02 sec)
```

Select count(\*) from NYC\_TRIPS; (To get count of number of rows added)

```
hadoop@ip-172-31-70-33:~
MySQL [nyc_taxi]> Select count(*) from NYC_TRIPS;
+-----+
| count(*) |
+-----+
| 18880595 |
+-----+
1 row in set (38.08 sec)

MySQL [nyc_taxi]>
```