

## Datasets:

```
https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv
https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-02.csv
https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-03.csv
https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-04.csv
https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-05.csv
https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-06.csv
```

## Create RDS Instance vg-rds-nyc-tlc

The screenshot shows the Amazon RDS console for the instance **vg-rds-nyc-tlc**. The left sidebar contains navigation links for Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Events, and Event subscriptions. The main content area displays the instance details under the 'Summary' tab. The instance is in the 'Available' state, using the 'db.t2.micro' class in the 'us-east-1d' region. The engine is 'MySQL Community'. The 'Connectivity & security' tab is selected, showing the endpoint 'vg-rds-nyc-tlc.c5dapn7s4k1r.us-east-1.rds.amazonaws.com' on port 3306, the availability zone 'us-east-1d', and the VPC 'vpc-0a63eea4ed0a92efd'. The security group is 'sg-0deb80682709791b5' and is active. The instance is publicly accessible.

Summary			
DB identifier	CPU	Status	Class
vg-rds-nyc-tlc	4.43%	Available	db.t2.micro
Role	Current activity	Engine	Region & AZ
Instance	0 Connections	MySQL Community	us-east-1d

Connectivity & security		
Endpoint & port	Networking	Security
Endpoint vg-rds-nyc-tlc.c5dapn7s4k1r.us-east-1.rds.amazonaws.com	Availability Zone us-east-1d	VPC security groups default (sg-0deb80682709791b5)
Port 3306	VPC vpc-0a63eea4ed0a92efd	Active
		Publicly accessible Yes

## Create EMR Cluster::

The screenshot shows the Amazon EMR console for the cluster **VG\_EMR\_NYC\_TLC\_20Feb**. The left sidebar contains navigation links for Amazon EMR, EMR Studio, EMR Serverless, EMR on EC2, Clusters, Notebooks, Git repositories, Security configurations, Block public access, VPC subnets, Events, EMR on EKS, and Virtual clusters. The main content area displays the cluster details under the 'Summary' tab. The cluster is in the 'Waiting' state. The configuration details include the release label 'emr-5.30.0', Hadoop distribution 'Amazon 2.8.5', applications 'HBase 1.4.13, Sqoop 1.4.7, Hive 2.3.6', and log URI 's3://aws-logs-568461604093-us-east-1/elasticmapreduce/'. The network and hardware details show the availability zone 'us-east-1c', subnet ID 'subnet-000b2980ac479239', and master instance type 'm4.xlarge'.

The new EMR console will become the default console on Feb 28, 2023. Switch to the new console. If you want, you can still switch back. Learn more

EMR Serverless is now GA. With EMR Serverless, get the benefits of Amazon EMR such as open source compatibility, latest versions and performance optimized runtime for popular frameworks along with easy provisioning, quick job startup, automatic capacity management, and simple cost controls. Get Started with EMR Serverless.

Clone Terminate AWS CLI export

Cluster: VG\_EMR\_NYC\_TLC\_20Feb Waiting Cluster ready to run steps.

Summary		Configuration details	
ID: j-1VHOJB1MBRX4T	Release label: emr-5.30.0		
Creation date: 2023-02-20 18:41 (UTC+5:30)	Hadoop distribution: Amazon 2.8.5		
Elapsed time: 7 minutes	Applications: HBase 1.4.13, Sqoop 1.4.7, Hive 2.3.6		
After last step completes: Cluster waits	Log URI: s3://aws-logs-568461604093-us-east-1/elasticmapreduce/		
Termination protection: Off Change	EMRFS consistent view: Disabled		
Tags: -- View All / Edit	Custom AMI ID: --		
Master public DNS: ec2-3-216-126-121.compute-1.amazonaws.com Connect to the Master Node Using SSH			

Application user interfaces		Network and hardware	
Persistent user interfaces: --	Availability zone: us-east-1c		
On-cluster user interfaces: Not Enabled Enable an SSH Connection	Subnet ID: subnet-000b2980ac479239		
	Master: Running 1 m4.xlarge		

Connect to the EMR:

Install MySQL drivers

Login as root using sudo -i

Download the datasets into root

```
wget -P /root/ https://nyc-tlc-upgrad.s3.amazonaws.com/yellow\_tripdata\_2017-01.csv
```

```
wget -P /root/ https://nyc-tlc-upgrad.s3.amazonaws.com/yellow\_tripdata\_2017-02.csv
```

Remove the header of the datasets

```
sed -i '1d' yellow_tripdata_2017-01.csv
```

```
sed -i '1d' yellow_tripdata_2017-02.csv
```

Copy the files to Hadoop FS

```
hadoop fs -put /root/yellow_tripdata_2017-01.csv /user/root/yellow_trip_data1
```

```
hadoop fs -put /root/yellow_tripdata_2017-02.csv /user/root/yellow_trip_data2
```

Connect to RDS instance:

```
mysql -h vg-rds-nyc-tlc.c5dapn7s4k1r.us-east-1.rds.amazonaws.com -u admin -p
```

Create database nyc\_tlc

```
[root@ip-172-31-11-67 ~]# mysql -h vg-rds-nyc-tlc.c5dapn7s4k1r.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 8.0.28 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_tlc;
Query OK, 1 row affected (0.01 sec)

MySQL [(none)]> use nyc_tlc
Database changed
MySQL [nyc_tlc]> 
```

Create a temp table without auto increment rowkey

```
Create table nyc_tlc.yellow_tripdata_temp(  
VendorID int,  
tpep_pickup_datetime datetime,  
tpep_dropoff_datetime datetime,  
passenger_count int,  
trip_distance double,  
RatecodeID int,  
store_and_fwd_flag varchar(150),  
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);
```

Create table yellow\_pathdata

```
Create table nyc_tlc.yellow_tripdata(  
rowkey MEDIUMINT NOT NULL AUTO_INCREMENT,  
VendorID int,  
tpep_pickup_datetime datetime,  
tpep_dropoff_datetime datetime,  
passenger_count int,  
trip_distance double,  
RatecodeID int,  
store_and_fwd_flag varchar(150),  
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,  
PRIMARY KEY (rowkey)  
);
```

#### Database changed

```
MySQL [nyc_tlc]> Create table nyc_tlc.yellow_tripdata(
-> rowkey MEDIUMINT NOT NULL AUTO_INCREMENT,
-> VendorID int,
-> tpep_pickup_datetime datetime,
-> tpep_dropoff_datetime datetime,
-> passenger_count int,
-> trip_distance double,
-> RatecodeID int,
-> store_and_fwd_flag varchar(150),
-> 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,
-> PRIMARY KEY (rowkey)
-> );
```

**Query OK, 0 rows affected (0.04 sec)**

```
MySQL [nyc_tlc]> show tables;
```

```
+-----+
| Tables_in_nyc_tlc |
+-----+
| yellow_tripdata   |
+-----+
```

**1 row in set (0.00 sec)**

Using Scoop Export to load data from csv into temp table.

```
sqoop export --connect jdbc:mysql://vg-rds-nyc-tlc.c5dapn7s4k1r.us-east-
1.rds.amazonaws.com:3306/nyc_tlc \
--table yellow_tripdata_temp \
--username admin --password admin123 \
--export-dir /user/root/yellow_trip_data1 \
--fields-terminated-by ',' --lines-terminated-by '\n'
```

```
sqoop export --connect jdbc:mysql://vg-rds-nyc-tlc.c5dapn7s4k1r.us-east-
1.rds.amazonaws.com:3306/nyc_tlc \
--table yellow_tripdata_temp \
--username admin --password admin123 \
--export-dir /user/root/yellow_trip_data2 \
--fields-terminated-by ',' --lines-terminated-by '\n'
```

### Database changed

```
MySQL [nyc_tlc]> show tables;
```

```
+-----+  
| Tables_in_nyc_tlc |  
+-----+  
| yellow_tripdata    |  
+-----+
```

**1 row in set (0.01 sec)**

```
MySQL [nyc_tlc]> select count(*) from yellow_tripdata;
```

```
+-----+  
| count(*) |  
+-----+  
| 18880595 |  
+-----+
```

```
[root@ip-172-31-11-104 ~]# wc -l yellow_tripdata_2017-01.csv
```

**9710820** yellow\_tripdata\_2017-01.csv

```
[root@ip-172-31-11-104 ~]# wc -l yellow_tripdata_2017-02.csv
```

**9169775** yellow\_tripdata\_2017-02.csv

File	Count
yellow_tripdata_2017-01.csv	9710820
yellow_tripdata_2017-02.csv	9169775
<b>Total</b>	<b>18880595</b>

Inserting into yellow\_tripdata which has an auto increment rowkey:

```
INSERT INOT yellow_tripdata
```

```
(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,
```

```
improvement_surcharge,
```

```
total_amount,  
congestion_surcharge,  
airport_fee double)  
Select 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,  
improvement_surcharge,  
total_amount,  
congestion_surcharge,  
airport_fee double  
From yellow_tripdata_temp;
```