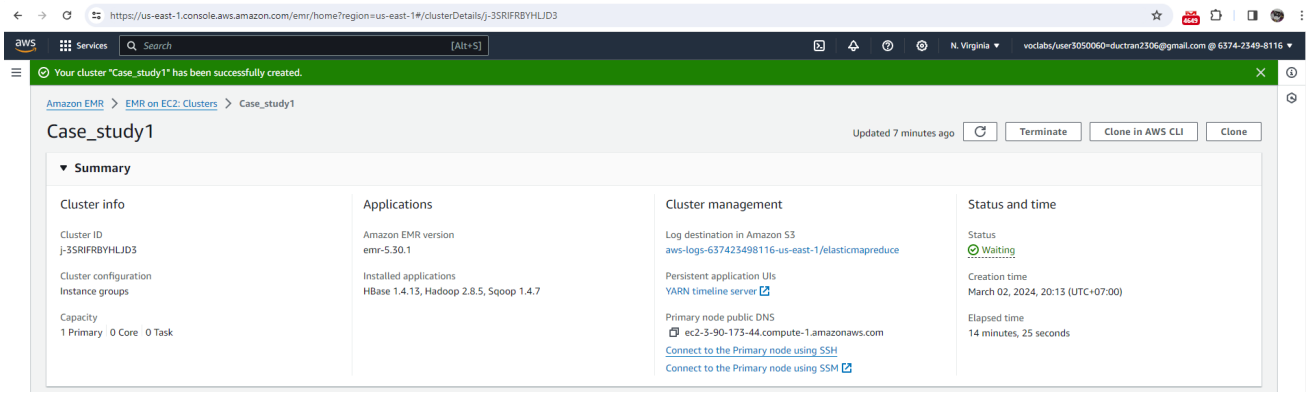
MapReduce Case Study Task 2

Create EMR instance



Ingest the data from RDS into HBase table using Sqoop

1) Create HBase database

Create table name "yellow_taxis_hbase" with a column family "tlc_trip"

```
Shell > create 'yellow_taxis_hbase', 'tlc_trip'
```

```
[root@ip-172-31-47-183 ~]# hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hbase/lib/client-facing-thirdparty/slf4j-reload4j-1.7.33.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Reload4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.15-amzn-0.1, rUnknown, Fri Jun 23 16:31:13 UTC 2023
Took 0.0030 seconds
hbase:001:0> create 'yellow_taxis', 'tlc_trip';
Created table yellow_taxis
Took 1.4504 seconds
=> Hbase::Table - yellow_taxis
```

2) Setup mysql connection

First, download MySQL connection jar file:

```
wget https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz
```

```
[hadoop@ip-172-31-42-104 ~]$ wget https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz
--2024-03-02 07:04:23--  https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz
Resolving de-mysql-connector.s3.amazonaws.com (de-mysql-connector.s3.amazonaws.com)... 52.217.229.49, 52.217.138.97, 52.217.89.124, ...
Connecting to de-mysql-connector.s3.amazonaws.com (de-mysql-connector.s3.amazonaws.com)|52.217.229.49|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4079310 (3.9M) [application/x-gzip]
Saving to: 'mysql-connector-java-8.0.25.tar.gz'

100%[=============================================================================================================================>] 4,079,310   20.1MB/s   in 0.2s

2024-03-02 07:04:24 (20.1 MB/s) - 'mysql-connector-java-8.0.25.tar.gz' saved [4079310/4079310]
```

Extract the file:

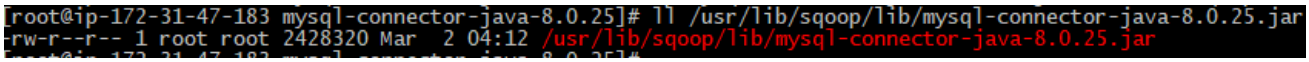
```
tar -xvf mysql-connector-java-8.0.25.tar.gz
```

Copy the jar file to Sqoop library:

```
cd mysql-connector-java-8.0.25/
sudo cp mysql-connector-java-8.0.25.jar /usr/lib/sqoop/lib/
```

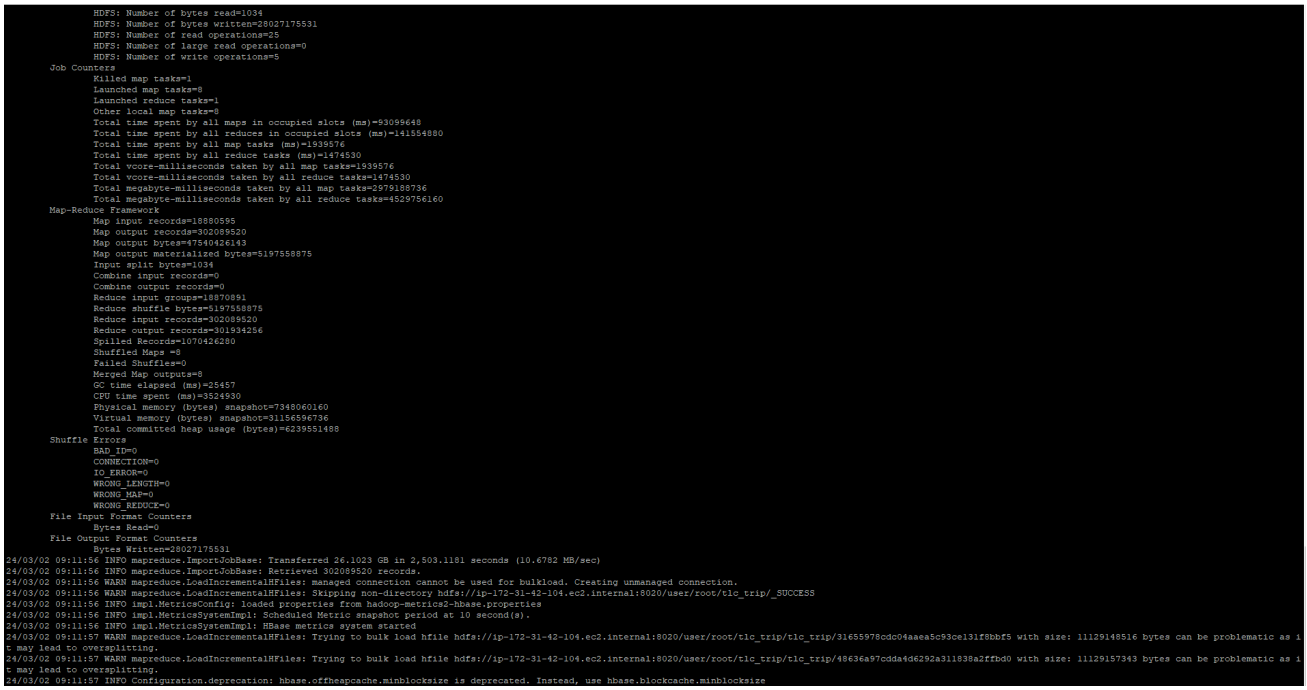
Check if the jar file has succesfully copied

```
ll /usr/lib/sqoop/lib/mysql-connector-java-8.0.25.jar
```



3) Import CSV

```
sqoop import -Dorg.apache.sqoop.splitter.allow_text_splitter=true \
--connect jdbc:mysql://mysql-db.cx2egsy0slxq.us-east-1.rds.amazonaws.com/yellow_taxis
\
--username admin -P --table tlc_trip -m 8 \
--hbase-table yellow_taxis_hbase  \
--column-family tlc_trip \
--hbase-create-table \
--hbase-row-key tpep_pickup_datetime,tpep_dropoff_datetime,PULocationID \
--split-by PULocationID
```



Explaination of the import command:

--connect : specifies connection string to RDS MySQL jdbc:mysql://[hostname]:[port number default is 3306]/[database name]

--username : user name connect to database.

-P : prompts for the password interactively.

-m : specifies the number of map tasks to use for the import job.

--hbase-table : specifies the name of HBase table.

--column-family : specifies the name of the column family.

--hbase-create-table : Sqoop will create the table if it does not exist.

--hbase-row-key : Specifies the row key for the table. In this case the row key is the composite of 3 columns.

--split-by : specifies the column by which the data should be split for parallel import.

-Dorg.apache.sqoop.splitter.allow_text_splitter=true : allow the use of the text splitter for splitting data. This is typically used to enable text-based splitting when importing data.

4) Check the result

```
describe 'yellow_taxis_hbase'
count 'yellow_taxis_hbase'
```