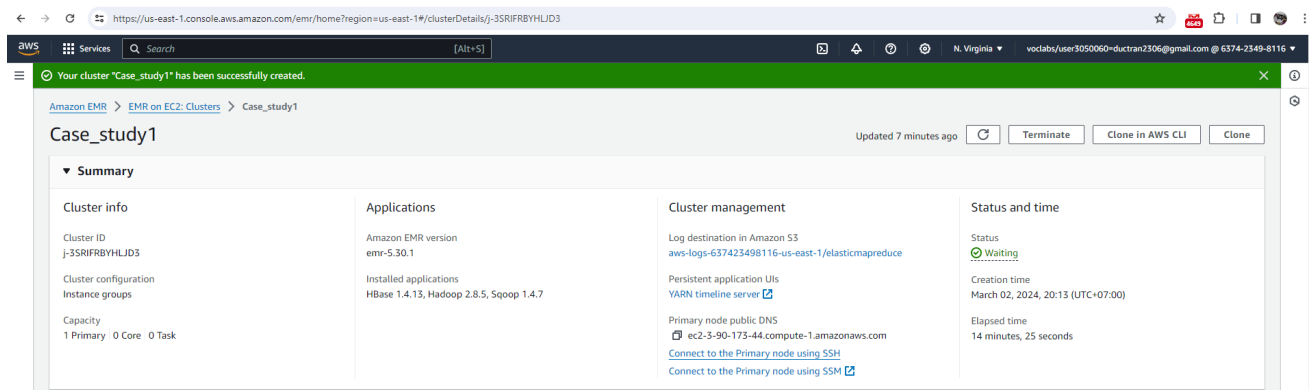


MapReduce Case Study Task 2

Create EMR instance



Ingest the data from RDS into HBase table using Sqoop

1) Create HBase database

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

```
Shell > create 'yellow_taxi_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> create 'yellow_taxi', 'tlc_trip';
Created table yellow_taxi
Took 1.4504 seconds
=> Hbase::Table - yellow_taxi
```

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 successfully copied

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

```
root@ip-172-31-47-183 mysql-connector-java-8.0.25]# ll /usr/lib/sqoop/lib/mysql-connector-java-8.0.25.jar  
-rw-r--r-- 1 root root 2428320 Mar  2 04:12 /usr/lib/sqoop/lib/mysql-connector-java-8.0.25.jar  
root@ip-172-31-47-183 mysql-connector-java-8.0.25]#
```

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
```

```

HDFS: Number of bytes read=1034
HDFS: Number of bytes written=2802717531
HDFS: Number of read operations=25
HDFS: Number of large read operations=0
HDFS: Number of write operations=5
Job Counters
  Killed map tasks=1
  Launched map tasks=8
  Launched reduce tasks=1
  Other local map tasks=3
  Total time spent by all maps in occupied slots (ms)=53099648
  Total time spent by all reduces in occupied slots (ms)=141554880
  Total time spent by all map tasks (ms)=1939576
  Total time spent by all reduce tasks (ms)=1474530
  Total vuore-milliseconds taken by all map tasks=1939576
  Total vuore-milliseconds taken by all reduce tasks=1474530
  Total megabyte-milliseconds taken by all map tasks=3979189736
  Total megabyte-milliseconds taken by all reduce tasks=4529756160
Map-Reduce Framework
  Map input records=18880595
  Map output records=302089520
  Map output bytes=17540426193
  Map output materialized bytes=5197558875
  Input split bytes=1034
  Combine input records=0
  Combine output records=0
  Reduce input groups=18870891
  Reduce shuffle bytes=5197558875
  Reduce input records=302089520
  Reduce output records=301934256
  Spilled Records=1070426280
  Shuffled Maps=0
  Failed Shuffles=0
  Merged Map outputs=8
  GC time elapsed (ms)=25457
  CPU time spent (ms)=3524930
  Physical memory (bytes) snapshot=7348060160
  Virtual memory (bytes) snapshot=31154596736
  Total committed heap usage (bytes)=4239531488
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=0
File Output Format Counters
  Bytes Written=2802717531
24/03/02 09:11:56 INFO mapreduce.ImportJobBase: Transferred 36.1022 GB in 2,503.1181 seconds (10.6782 MB/sec)
24/03/02 09:11:56 INFO mapreduce.ImportJobBase: Retrieved 302089520 records.
24/03/02 09:11:56 WARN mapreduce.LoadIncrementalHFiles: managed connection cannot be used for bulkload. Creating unmanaged connection.
24/03/02 09:11:56 WARN mapreduce.LoadIncrementalHFiles: Skipping non-directory hdfs://ip-172-31-42-104.ec2.internal:8020/user/root/tlc_trip/_SUCCESS
24/03/02 09:11:56 INFO impl.MetricConf: loaded properties from hadoop-metrics2-hbase.properties
24/03/02 09:11:56 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
24/03/02 09:11:56 INFO impl.MetricsSystemImpl: HBase metrics system started
24/03/02 09:11:57 WARN mapreduce.LoadIncrementalHFiles: Trying to bulk load hfile hdfs://ip-172-31-42-104.ec2.internal:8020/user/root/tlc_trip/tlc_trip/48634e97cdda4d6292a311838a2ffb30 with size: 11129148516 bytes can be problematic as i
t may lead to oversplitting.
24/03/02 09:11:57 WARN mapreduce.LoadIncrementalHFiles: Trying to bulk load hfile hdfs://ip-172-31-42-104.ec2.internal:8020/user/root/tlc_trip/tlc_trip/48634e97cdda4d6292a311838a2ffb30 with size: 11129157343 bytes can be problematic as i
t may lead to oversplitting.
24/03/02 09:11:57 INFO Configuration.deprecation: hbase.offheapcache.minblocksize is deprecated. Instead, use hbase.blockcache.minblocksize

```

Explanation 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_taxi_hbase'  
count 'yellow_taxi_hbase'
```

```
hbase(main):001:0> describe 'yellow_taxi_hbase'  
Table yellow_taxi_hbase is ENABLED  
yellow_taxi_hbase  
COLUMN FAMILIES DESCRIPTION  
(NAME => 't1c_trip', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE  
=> '65536', REPLICATION_SCOPE => '0')  
1 row(s) in 0.3890 seconds
```