

1. **First, log in into your EMR instance (using hadoop, then switch user to root using `sudo -i`) and complete the initial steps of setup. Now you need to run the following command to install the MySQL connector jar file.**

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

```
tar -xvf mysql-connector-java-
```

```
8.0.25.tar.gz cd mysql-connector-
```

```
java-8.0.25/
```

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

2. **To ingest data from mySQL RDS to HBase table:**

```
sqoop import --connect jdbc:mysql://mydbinstance.cxueuenwslg.us-east-1.rds.amazonaws.com/taxi_records --username admin --password rajatramesshgarg --table trip_log --hbase-table trip_log_hbase --column-family cf1 --hbase-create-table --hbase-row-key tpep_pickup_datetime,tpep_dropoff_datetime --hbase-bulkload --split-by payment_type
```

command explanation:

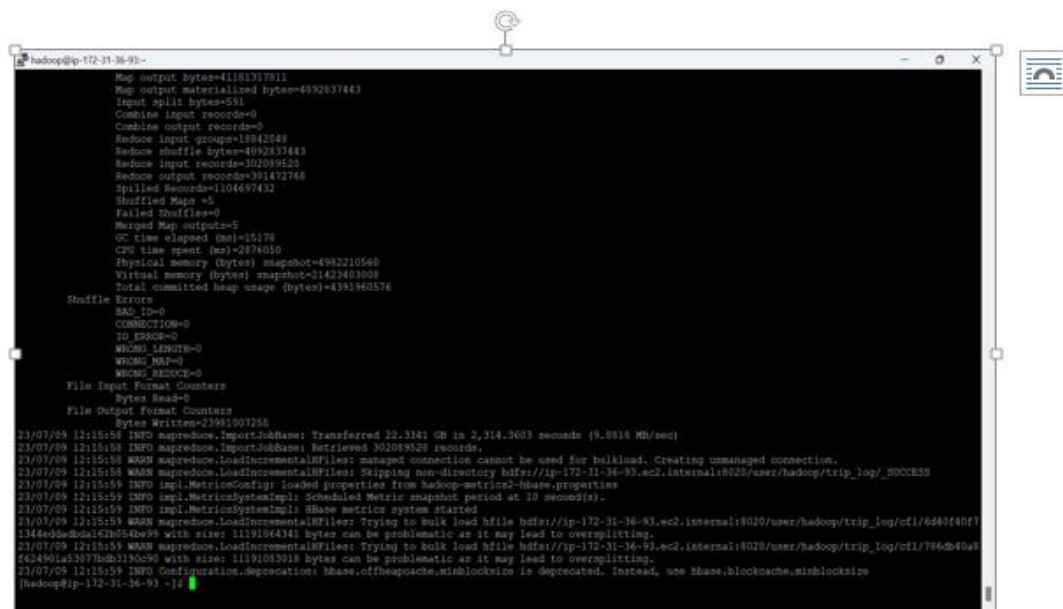
This is a `sqoop` command that imports data from a MySQL database table `trip_log` into an HBase table `trip_log_hbase`. Here is what each option in the command does:

- `--connect`: specifies the JDBC connection string for the MySQL database.
- `--username`: specifies the username to use when connecting to the MySQL database.
- `--password`: specifies the password to use when connecting to the MySQL database.
- `--table`: specifies the name of the MySQL table to import data from.
- `--hbase-table`: specifies the name of the HBase table to import data into.
- `--column-family`: specifies the name of the column family in HBase where the imported data will be stored.
- `--hbase-create-table`: creates an HBase table if it does not exist.
- `--hbase-row-key`: specifies one or more columns from the MySQL table that will be used as the row key in HBase.

- `--hbase-bulkload`: uses HBase bulk load feature for faster data loading.
- `--split-by`: specifies a column from the MySQL table that will be used to split data into multiple HBase regions.

In this command, data is imported from the MySQL table `trip_log` into an HBase table named `trip_log_hbase`. The column family in HBase where the imported data will be stored is named `cf1`. The row key in HBase is composed of two columns from the MySQL table:

`tpep_pickup_datetime` and `tpep_dropoff_datetime`. The data is loaded into HBase using bulk load feature for faster loading. The data is split into multiple regions based on the column `payment_type`.



```

hadoop@ip-172-31-36-93:~$
map output bytes=41181317811
Map output materialized bytes=4832837443
Input split bytes=581
Combine input records=0
Combine output records=0
Reduce input groups=1842048
Reduce shuffle bytes=4892437443
Reduce input records=302089520
Reduce output records=301472768
Spilled Records=1104687432
Shuffled Maps=5
Failed Shuffles=0
Merged Map outputs=5
GC time elapsed (ms)=15178
CPU time spent (sec)=2474050
Physical memory (bytes) snapshot=4982270560
Virtual memory (bytes) snapshot=3142340800
Total committed heap usage (bytes)=4391960576

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=2396107255
23/07/29 12:15:55 INFO mapreduce.ImportJobBase: Transferred 22.3341 GB in 2,314.3693 seconds (9.8818 MB/sec)
23/07/29 12:15:58 INFO mapreduce.ImportJobBase: Retrieved 302089520 records.
23/07/29 12:15:58 WARN mapreduce.LoadIncrementalHFiles: Managed connection cannot be used for bulkload. Creating unmanaged connection.
23/07/29 12:15:58 WARN mapreduce.LoadIncrementalHFiles: Skipping non-directory hdfs://ip-172-31-36-93.ec2.internal:8020/user/hadoop/trip_log/_SUCCESS
23/07/29 12:15:59 INFO impl.MetricConfig: Loaded properties from hadoop-metrics2-hbase.properties
23/07/29 12:15:59 INFO impl.MetricSystemImpl: Scheduled Metric snapshot period at 10 second(s).
23/07/29 12:15:59 INFO impl.MetricSystemImpl: HBase metrics system started
23/07/29 12:15:59 WARN mapreduce.LoadIncrementalHFiles: Trying to bulk load hfile hdfs://ip-172-31-36-93.ec2.internal:8020/user/hadoop/trip_log/cf1/6463f40f71344edd8bd61c2b554be99 with size: 11191064341 bytes can be problematic as it may lead to oversplitting.
23/07/29 12:15:59 WARN mapreduce.LoadIncrementalHFiles: Trying to bulk load hfile hdfs://ip-172-31-36-93.ec2.internal:8020/user/hadoop/trip_log/cf1/796d040a8634901a53077b0d119cc99 with size: 11191082018 bytes can be problematic as it may lead to oversplitting.
23/07/29 12:15:59 INFO Configuration.deprecation: hbase.cdfheapcache.mishlocksize is deprecated. Instead, use hbase.blockcache.mishlocksize
(hadoop@ip-172-31-36-93) ~$

```

```
hadoop@ip-172-31-36-83:~$
at org.jruby.Main.run(Main.java:108)
at org.jruby.Main.main(Main.java:188)
log4j:ERROR Either File or DataPattern options are not set for appender [DRFAS].
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
Version 1.4.13, r0kknown, Fri Apr 17 15:11:24 UTC 2020

hbase(main):001:0> list
TABLE
trip_log_hbase
1 row(s) in 0.2730 seconds

=> ["trip_log_hbase"]
hbase(main):002:0> describe trip_log_hbase
NameError: undefined local variable or method 'trip_log_hbase' for #<Object#717508eff>

hbase(main):003:0> describe 'trip_log_hbase'
Table trip_log_hbase is ENABLED
trip_log_hbase
COLUMN FAMILIES DESCRIPTION
NAME => 'cf1', BLOCKFILTER => '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 0.3030 seconds

hbase(main):004:0> count 'trip_log_hbase'
Current count: 1000, rows 2017-01-01 00:04:50 2017-01-01 00:14:24
Current count: 2000, rows 2017-01-01 00:10:49 2017-01-01 00:32:23
Current count: 3000, rows 2017-01-01 00:14:09 2017-01-01 00:15:47
Current count: 4000, rows 2017-01-01 00:17:05 2017-01-01 00:21:41
Current count: 5000, rows 2017-01-01 00:17:44 2017-01-01 00:24:27
Current count: 6000, rows 2017-01-01 00:22:20 2017-01-01 00:34:07
Current count: 7000, rows 2017-01-01 00:24:57 2017-01-01 00:27:01
Current count: 8000, rows 2017-01-01 00:27:31 2017-01-01 01:12:08
Current count: 9000, rows 2017-01-01 00:30:06 2017-01-01 00:54:24
Current count: 10000, rows 2017-01-01 00:32:41 2017-01-01 00:41:09
Current count: 11000, rows 2017-01-01 00:35:22 2017-01-01 01:06:32
Current count: 12000, rows 2017-01-01 00:37:56 2017-01-01 00:45:44
Current count: 13000, rows 2017-01-01 00:40:17 2017-01-01 00:52:45
Current count: 14000, rows 2017-01-01 00:42:58 2017-01-01 01:15:32
Current count: 15000, rows 2017-01-01 00:45:37 2017-01-01 00:59:34
Current count: 16000, rows 2017-01-01 00:48:00 2017-01-01 01:04:19
Current count: 17000, rows 2017-01-01 00:50:32 2017-01-01 01:04:34
Current count: 18000, rows 2017-01-01 00:53:06 2017-01-01 00:59:28
```

