

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.

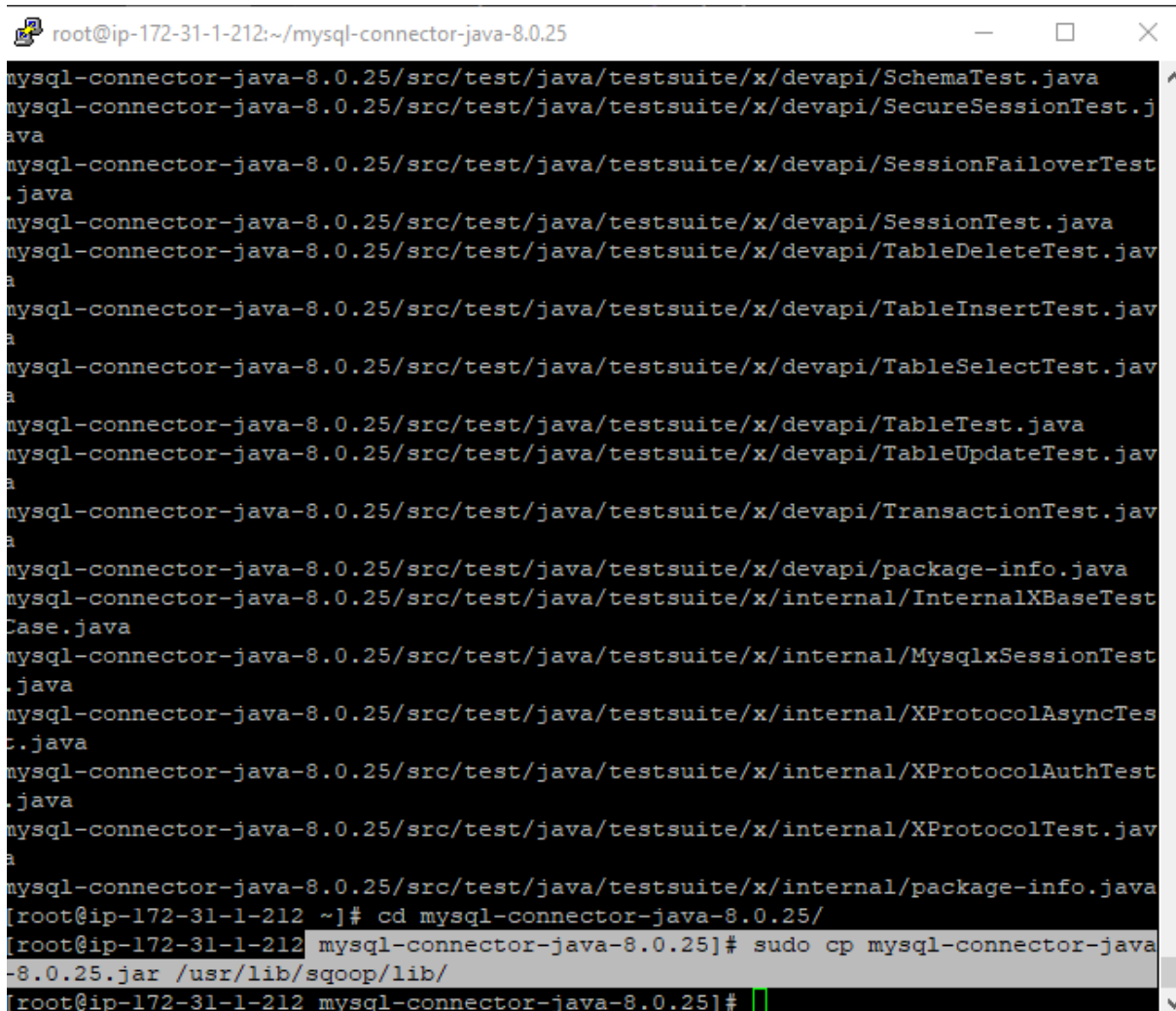
Sudo -i

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/

A terminal window screenshot from an EMR instance. The title bar shows 'root@ip-172-31-1-212:~/mysql-connector-java-8.0.25'. The terminal content shows a list of files in the 'src/test/java/testsuite/x/devapi/' directory, including SchemaTest.java, SecureSessionTest.java, SessionFailoverTest.java, SessionTest.java, TableDeleteTest.java, TableInsertTest.java, TableSelectTest.java, TableTest.java, TableUpdateTest.java, TransactionTest.java, package-info.java, InternalXBaseTestCase.java, MysqlxSessionTest.java, XProtocolAsyncTest.java, XProtocolAuthTest.java, XProtocolTest.java, and another package-info.java. Below this list, the user enters 'cd mysql-connector-java-8.0.25/' and then 'sudo cp mysql-connector-java-8.0.25.jar /usr/lib/sqoop/lib/'. The prompt changes to '[root@ip-172-31-1-212 mysql-connector-java-8.0.25]#', indicating a successful root shell session.

```
mysql-connector-java-8.0.25/src/test/java/testsuite/x/devapi/SchemaTest.java
mysql-connector-java-8.0.25/src/test/java/testsuite/x/devapi/SecureSessionTest.j
ava
mysql-connector-java-8.0.25/src/test/java/testsuite/x/devapi/SessionFailoverTest
.java
mysql-connector-java-8.0.25/src/test/java/testsuite/x/devapi/SessionTest.java
mysql-connector-java-8.0.25/src/test/java/testsuite/x/devapi/TableDeleteTest.jav
a
mysql-connector-java-8.0.25/src/test/java/testsuite/x/devapi/TableInsertTest.jav
a
mysql-connector-java-8.0.25/src/test/java/testsuite/x/devapi/TableSelectTest.jav
a
mysql-connector-java-8.0.25/src/test/java/testsuite/x/devapi/TableTest.java
mysql-connector-java-8.0.25/src/test/java/testsuite/x/devapi/TableUpdateTest.jav
a
mysql-connector-java-8.0.25/src/test/java/testsuite/x/devapi/TransactionTest.jav
a
mysql-connector-java-8.0.25/src/test/java/testsuite/x/devapi/package-info.java
mysql-connector-java-8.0.25/src/test/java/testsuite/x/internal/InternalXBaseTest
Case.java
mysql-connector-java-8.0.25/src/test/java/testsuite/x/internal/MysqlxSessionTest
.java
mysql-connector-java-8.0.25/src/test/java/testsuite/x/internal/XProtocolAsyncTes
t.java
mysql-connector-java-8.0.25/src/test/java/testsuite/x/internal/XProtocolAuthTest
.java
mysql-connector-java-8.0.25/src/test/java/testsuite/x/internal/XProtocolTest.jav
a
mysql-connector-java-8.0.25/src/test/java/testsuite/x/internal/package-info.java
[root@ip-172-31-1-212 ~]# cd mysql-connector-java-8.0.25/
[root@ip-172-31-1-212 mysql-connector-java-8.0.25]# sudo cp mysql-connector-java
-8.0.25.jar /usr/lib/sqoop/lib/
[root@ip-172-31-1-212 mysql-connector-java-8.0.25]#
```

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

sqoop import \

--connect jdbc:mysql://database-1.cjd8k4oc9drm.ap-south-1.rds.amazonaws.com/taxi_records

\

--username admin \

--password user1234 \

```
--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`. Let me know if you have any other questions

```
root@ip-172-31-1-212:~/mysql-connector-java-8.0.25
guration, Class<?>...) method has been deprecated since it is easy to use incorr
ectly. Most users should rely on addDependencyJars(Job) instead. See HBASE-8386
for more details.
23/11/12 12:08:14 WARN mapreduce.TableMapReduceUtil: The addDependencyJars(Confi
guration, Class<?>...) method has been deprecated since it is easy to use incorr
ectly. Most users should rely on addDependencyJars(Job) instead. See HBASE-8386
for more details.
23/11/12 12:08:15 INFO zlib.ZlibFactory: Successfully loaded & initialized nativ
e-zlib library
23/11/12 12:08:15 INFO compress.CodecPool: Got brand-new compressor [.deflate]
23/11/12 12:08:15 INFO client.RMProxy: Connecting to ResourceManager at ip-172-3
1-1-212.ap-south-1.compute.internal/172.31.1.212:8032
23/11/12 12:08:16 INFO db.DBInputFormat: Using read committed transaction isolati
on
23/11/12 12:08:16 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN
('payment_type'), MAX('payment_type') FROM `trip_log`
23/11/12 12:09:01 INFO db.IntegerSplitter: Split size: 156562; Num splits: 4 fro
m: -350 to: 625901
23/11/12 12:09:01 INFO mapreduce.JobSubmitter: number of splits:4
23/11/12 12:09:02 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_16
99782747383_0002
23/11/12 12:09:02 INFO impl.YarnClientImpl: Submitted application application_16
99782747383_0002
23/11/12 12:09:02 INFO mapreduce.Job: The url to track the job: http://ip-172-31
-1-212.ap-south-1.compute.internal:20888/proxy/application_1699782747383_0002/
23/11/12 12:09:02 INFO mapreduce.Job: Running job: job_1699782747383_0002
23/11/12 12:09:10 INFO mapreduce.Job: Job job_1699782747383_0002 running in uber
mode : false
23/11/12 12:09:10 INFO mapreduce.Job: map 0% reduce 0%
23/11/12 12:10:06 INFO mapreduce.Job: map 25% reduce 0%
23/11/12 12:11:00 INFO mapreduce.Job: map 50% reduce 0%
23/11/12 12:11:55 INFO mapreduce.Job: map 75% reduce 0%
```

```
hadoop@ip-172-31-1-212:~/mysql-connector-java-8.0.25
FILE: Number of bytes read=6110436348
FILE: Number of bytes written=9476725751
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=511
HDFS: Number of bytes written=0
HDFS: Number of read operations=8
HDFS: Number of large read operations=0
HDFS: Number of write operations=0
Job Counters
  Failed reduce tasks=4
  Killed map tasks=1
  Launched map tasks=4
  Launched reduce tasks=4
  Other local map tasks=4
  Total time spent by all maps in occupied slots (ms)=56102592
  Total time spent by all reduces in occupied slots (ms)=1116768
  Total time spent by all map tasks (ms)=1168804
  Total time spent by all reduce tasks (ms)=11633
  Total vcore-milliseconds taken by all map tasks=1168804
  Total vcore-milliseconds taken by all reduce tasks=11633
  Total megabyte-milliseconds taken by all map tasks=1795282944
  Total megabyte-milliseconds taken by all reduce tasks=35736576
Map-Reduce Framework
  Map input records=18880595
  Map output records=320970115
  Map output bytes=32113646988
  Map output materialized bytes=3365369987
  Input split bytes=511
  Combine input records=0
  Spilled Records=899298135
```

```

hadoop@ip-172-31-36-93:~
  at org.jruby.Main.run(Main.java:208)
  at org.jruby.Main.main(Main.java:198)
log4j:ERROR Either File or DatePattern options are not set for appender [DREAS].
Hbase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
Version 1.4.13, rUnknown, Fri Apr 17 15:18:24 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:0x7f0d8eff>

hbase(main):003:0> describe 'trip_log_hbase'
Table trip_log_hbase is ENABLED
trip_log_hbase
COLUMN FAMILIES DESCRIPTION
(NAME => 'cf1', 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 0.1030 seconds

hbase(main):004:0> count 'trip_log_hbase'
Current count: 1000, row: 2017-01-01 00:06:50 2017-01-01 00:14:24
Current count: 2000, row: 2017-01-01 00:18:49 2017-01-01 00:32:23
Current count: 3000, row: 2017-01-01 00:14:09 2017-01-01 00:15:47
Current count: 4000, row: 2017-01-01 00:17:05 2017-01-01 00:33:41
Current count: 5000, row: 2017-01-01 00:19:44 2017-01-01 00:29:27
Current count: 6000, row: 2017-01-01 00:22:20 2017-01-01 00:35:07
Current count: 7000, row: 2017-01-01 00:24:57 2017-01-01 00:27:01
Current count: 8000, row: 2017-01-01 00:27:31 2017-01-01 01:12:08
Current count: 9000, row: 2017-01-01 00:38:06 2017-01-01 00:56:24
Current count: 10000, row: 2017-01-01 00:32:41 2017-01-01 00:41:09
Current count: 11000, row: 2017-01-01 00:35:22 2017-01-01 01:06:12
Current count: 12000, row: 2017-01-01 00:37:56 2017-01-01 00:45:49
Current count: 13000, row: 2017-01-01 00:40:27 2017-01-01 00:52:43
Current count: 14000, row: 2017-01-01 00:42:58 2017-01-01 01:15:32
Current count: 15000, row: 2017-01-01 00:45:37 2017-01-01 00:59:36
Current count: 16000, row: 2017-01-01 00:48:00 2017-01-01 01:04:19
Current count: 17000, row: 2017-01-01 00:50:32 2017-01-01 01:04:34
Current count: 18000, row: 2017-01-01 00:53:06 2017-01-01 00:59:26

```