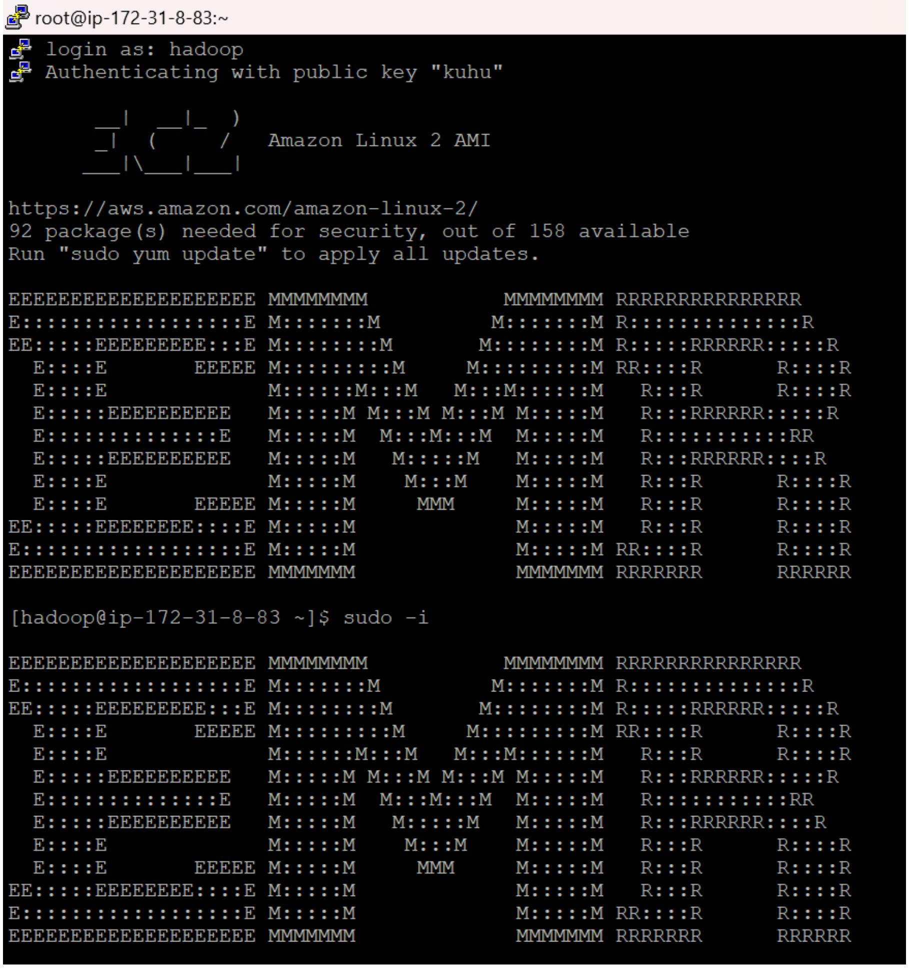
Data ingestion from RDS to HDFS using Sqoop

We followed the below steps to complete the above task:

1. First, we create a EMR cluster containing apps Hadoop and Sqoop and then logged into the EMR cluster and switched to root user by running **sudo -i** command.



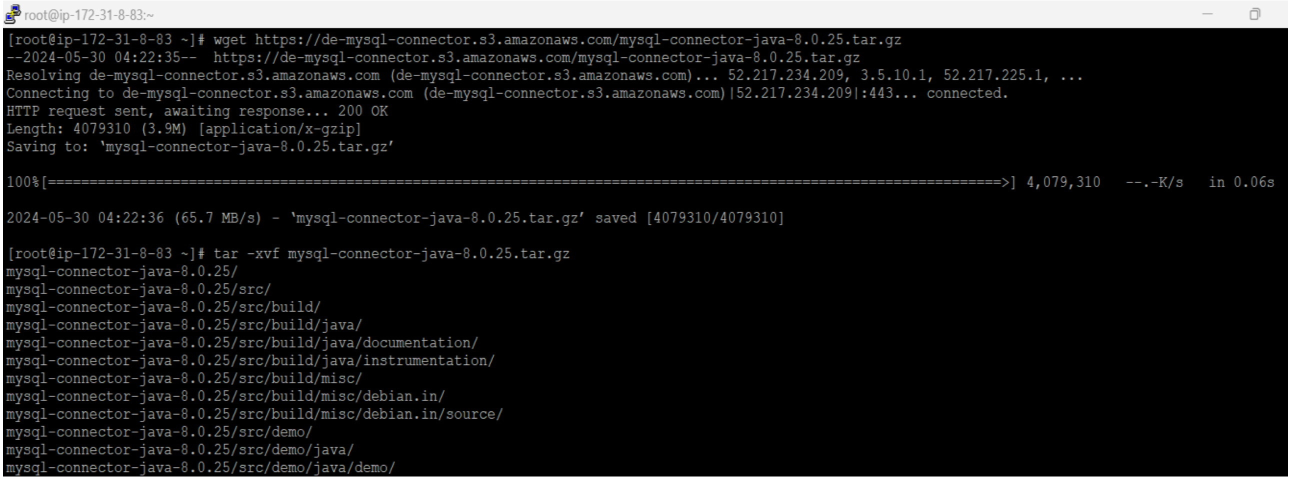
root@ip-172-31-8-83:~

```
login as: hadoop
Authenticating with public key "kuhu"

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
92 package(s) needed for security, out of 158 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-8-83 ~]$ sudo -i

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR
```

2. Next, we need to run the below commands to install the MySQL connector jar file:

   wget [**https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz**](https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz)

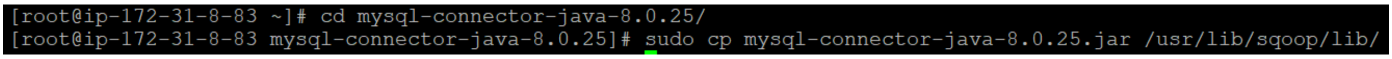
   tar -xvf mysql-connector-java-8.0.25.tar.gz



root@ip-172-31-8-83:~

```
[root@ip-172-31-8-83 ~]# wget https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz
--2024-05-30 04:22:35--  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)... 52.217.234.209, 3.5.10.1, 52.217.225.1, ...
Connecting to de-mysql-connector.s3.amazonaws.com (de-mysql-connector.s3.amazonaws.com)|52.217.234.209|: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   --.-K/s   in 0.06s

2024-05-30 04:22:36 (65.7 MB/s) - ‘mysql-connector-java-8.0.25.tar.gz’ saved [4079310/4079310]

[root@ip-172-31-8-83 ~]# tar -xvf mysql-connector-java-8.0.25.tar.gz
mysql-connector-java-8.0.25/
mysql-connector-java-8.0.25/src/
mysql-connector-java-8.0.25/src/build/
mysql-connector-java-8.0.25/src/build/java/
mysql-connector-java-8.0.25/src/build/java/documentation/
mysql-connector-java-8.0.25/src/build/java/instrumentation/
mysql-connector-java-8.0.25/src/build/misc/
mysql-connector-java-8.0.25/src/build/misc/debian.in/
mysql-connector-java-8.0.25/src/build/misc/debian.in/source/
mysql-connector-java-8.0.25/src/demo/
mysql-connector-java-8.0.25/src/demo/java/
mysql-connector-java-8.0.25/src/demo/java/demo/
```

```
cd mysql-connector-java-8.0.25/

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



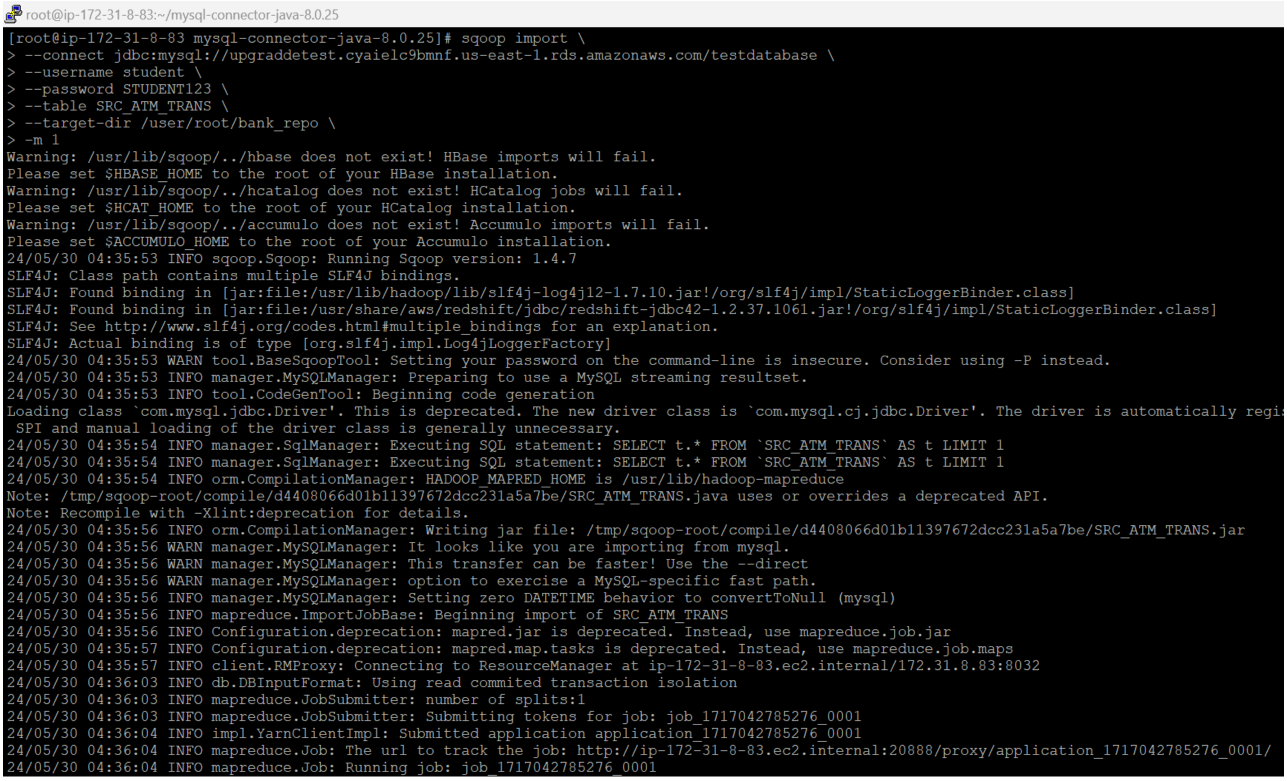
```
[root@ip-172-31-8-83 ~]# cd mysql-connector-java-8.0.25/
[root@ip-172-31-8-83 mysql-connector-java-8.0.25]# sudo cp mysql-connector-java-8.0.25.jar /usr/lib/sqoop/lib/
```

3. Next, for ingesting data from AWS RDS's MySQL database instance to the EMR cluster, we ran the below command:

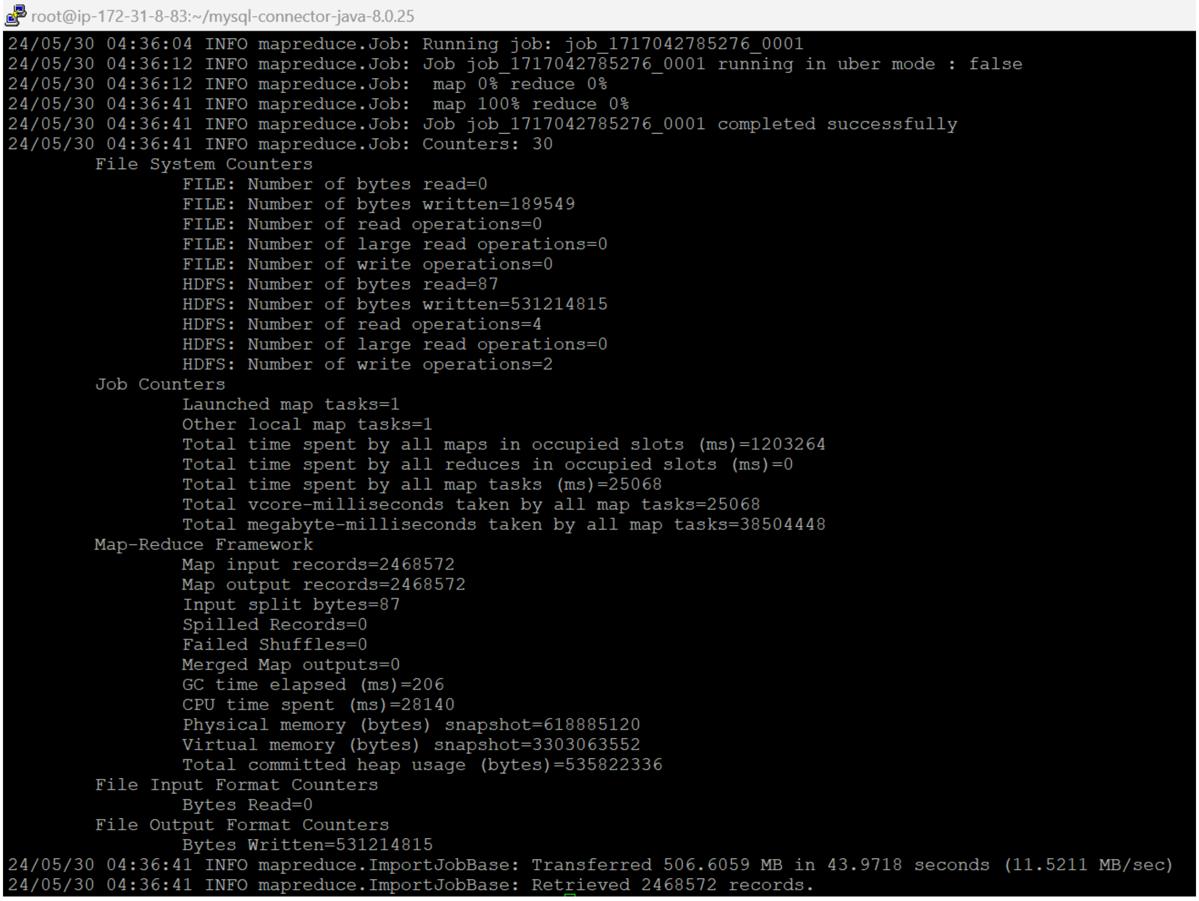
```
sqoop import \
--connect jdbc:mysql://upgraddetest.cyaielc9bmnf.us-east-1.rds.amazonaws.com/testdatabase \
--username student \
--password STUDENT123 \
--table SRC_ATM_TRANS \
--target-dir /user/root/bank_repo \
-m 1
```

Explanation for the above command is as follows:

- **sqoop import –** is a command used for importing data
- **--connect –** specifies the JDBC string of the MySQL database
- **--username –** specifies the username to connect to the MySQL database
- **--password** – specifies the password to connect to the MySQL database
- **--table –** specifies the MySQL table name from where the data will be imported
- **--target-dir** – specifies the directory to where the data will be imported
- **-m 1** – specifies the number of mappers



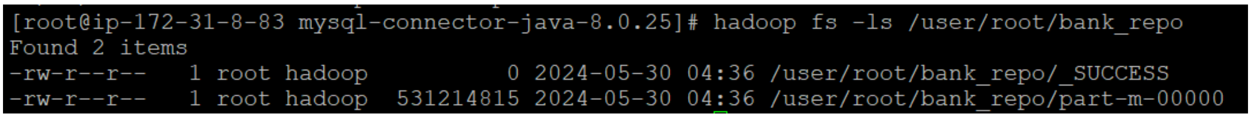
```
[root@ip-172-31-8-83 mysql-connector-java-8.0.25]# sqoop import \
> --connect jdbc:mysql://upgraddetest.cyaielc9bmnf.us-east-1.rds.amazonaws.com/testdatabase \
> --username student \
> --password STUDENT123 \
> --table SRC_ATM_TRANS \
> --target-dir /user/root/bank_repo \
> -m 1
Warning: /usr/lib/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/lib/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
24/05/30 04:35:53 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/share/aws/redshift/jdbc/redshift-jdbc42-1.2.37.1061.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.Log4jLoggerFactory]
24/05/30 04:35:53 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
24/05/30 04:35:53 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
24/05/30 04:35:53 INFO tool.CodeGenTool: Beginning code generation
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is automatically regi
 SPI and manual loading of the driver class is generally unnecessary.
24/05/30 04:35:54 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `SRC_ATM_TRANS` AS t LIMIT 1
24/05/30 04:35:54 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `SRC_ATM_TRANS` AS t LIMIT 1
24/05/30 04:35:54 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-root/compile/d4408066d01b11397672dcc231a5a7be/SRC_ATM_TRANS.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
24/05/30 04:35:56 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-root/compile/d4408066d01b11397672dcc231a5a7be/SRC_ATM_TRANS.jar
24/05/30 04:35:56 WARN manager.MySQLManager: It looks like you are importing from mysql.
24/05/30 04:35:56 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
24/05/30 04:35:56 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
24/05/30 04:35:56 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
24/05/30 04:35:56 INFO mapreduce.ImportJobBase: Beginning import of SRC_ATM_TRANS
24/05/30 04:35:56 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
24/05/30 04:35:57 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
24/05/30 04:35:57 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-8-83.ec2.internal/172.31.8.83:8032
24/05/30 04:36:03 INFO db.DBInputFormat: Using read commited transaction isolation
24/05/30 04:36:03 INFO mapreduce.JobSubmitter: number of splits:1
24/05/30 04:36:03 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1717042785276_0001
24/05/30 04:36:04 INFO impl.YarnClientImpl: Submitted application application_1717042785276_0001
24/05/30 04:36:04 INFO mapreduce.Job: The url to track the job: http://ip-172-31-8-83.ec2.internal:20888/proxy/application_1717042785276_0001/
24/05/30 04:36:04 INFO mapreduce.Job: Running job: job_1717042785276_0001
```


root@ip-172-31-8-83:~/mysql-connector-java-8.0.25

```
24/05/30 04:36:04 INFO mapreduce.Job: Running job: job_1717042785276_0001
24/05/30 04:36:12 INFO mapreduce.Job: Job job_1717042785276_0001 running in uber mode : false
24/05/30 04:36:12 INFO mapreduce.Job:  map 0% reduce 0%
24/05/30 04:36:41 INFO mapreduce.Job:  map 100% reduce 0%
24/05/30 04:36:41 INFO mapreduce.Job: Job job_1717042785276_0001 completed successfully
24/05/30 04:36:41 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=189549
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=87
                HDFS: Number of bytes written=531214815
                HDFS: Number of read operations=4
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Other local map tasks=1
                Total time spent by all maps in occupied slots (ms)=1203264
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=25068
                Total vcore-milliseconds taken by all map tasks=25068
                Total megabyte-milliseconds taken by all map tasks=38504448
        Map-Reduce Framework
                Map input records=2468572
                Map output records=2468572
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=206
                CPU time spent (ms)=28140
                Physical memory (bytes) snapshot=618885120
                Virtual memory (bytes) snapshot=3303063552
                Total committed heap usage (bytes)=535822336
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=531214815
24/05/30 04:36:41 INFO mapreduce.ImportJobBase: Transferred 506.6059 MB in 43.9718 seconds (11.5211 MB/sec)
24/05/30 04:36:41 INFO mapreduce.ImportJobBase: Retrieved 2468572 records.
```

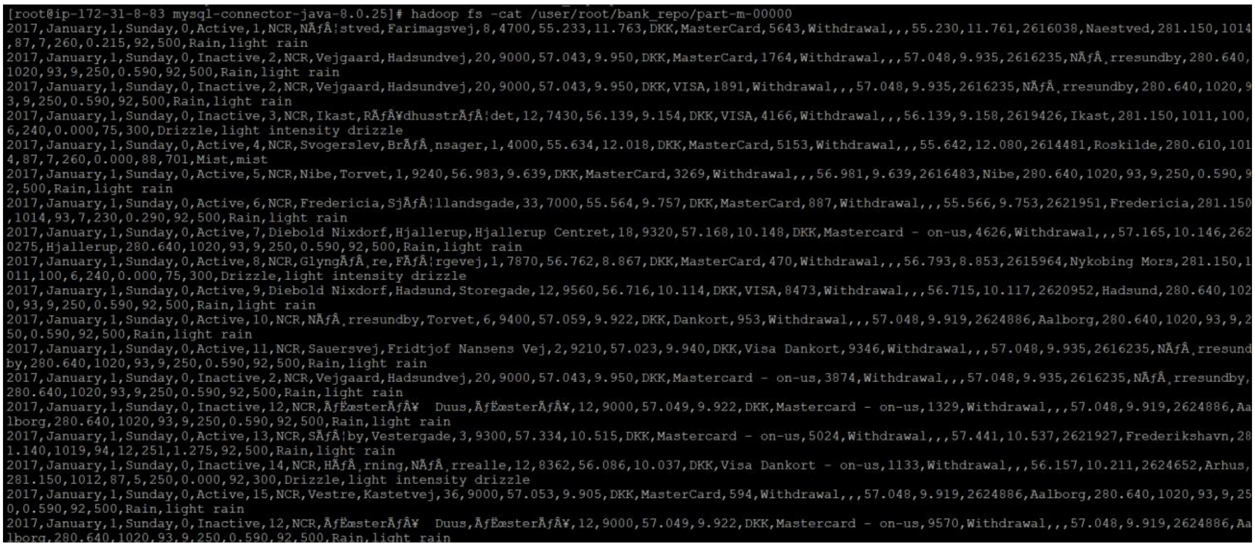
We can see from the above screenshot's last line: 2468572 records have been retrieved.

4. Next, we ran the command **hadoop fs -ls /user/root/bank_repo** which has two files: 1. The success file which indicates the import was successful and Mapreduce job ran correctly 2. The file where all the data from RDS table got stored (Only 1 file got created because only 1 mapper ran and all the data got stored in this 1 file)



```
[root@ip-172-31-8-83 mysql-connector-java-8.0.25]# hadoop fs -ls /user/root/bank_repo
Found 2 items
-rw-r--r--   1 root hadoop          0 2024-05-30 04:36 /user/root/bank_repo/_SUCCESS
-rw-r--r--   1 root hadoop  531214815 2024-05-30 04:36 /user/root/bank_repo/part-m-00000
```

5. Next, we ran the command **hadoop fs -cat /user/root/bank_repo/part-m-00000** to see the list of data that got imported from RDS to HDFS.



```
[root@ip-172-31-8-83 mysql-connector-java-8.0.25]# hadoop fs -cat /user/root/bank_repo/part-m-00000
2017,January,1,Sunday,0,Active,1,NCR,NÃƒÂ¦stved,Farimagsvej,8,4700,55.233,11.763,DKK,MasterCard,5643,Withdrawal,,,55.230,11.761,2616038,Naestved,281.150,1014
,87,7,260,0.215,92,500,Rain,light rain
2017,January,1,Sunday,0,Inactive,2,NCR,Vejgaard,Hadsundvej,20,9000,57.043,9.950,DKK,MasterCard,1764,Withdrawal,,,57.048,9.935,2616235,NÃƒÂ¸rresundby,280.640,
1020,93,9,250,0.590,92,500,Rain,light rain
2017,January,1,Sunday,0,Inactive,2,NCR,Vejgaard,Hadsundvej,20,9000,57.043,9.950,DKK,VISA,1891,Withdrawal,,,57.048,9.935,2616235,NÃƒÂ¸rresundby,280.640,1020,9
3,9,250,0.590,92,500,Rain,light rain
2017,January,1,Sunday,0,Inactive,3,NCR,Ikast,RÃƒÂ¥dhusstrÃƒÂ¦det,12,7430,56.139,9.154,DKK,VISA,4166,Withdrawal,,,56.139,9.158,2619426,Ikast,281.150,1011,100,
6,240,0.000,75,300,Drizzle,light intensity drizzle
2017,January,1,Sunday,0,Active,4,NCR,Svogerslev,BrÃƒÂ¸nsager,1,4000,55.634,12.018,DKK,MasterCard,5153,Withdrawal,,,55.642,12.080,2614481,Roskilde,280.610,101
4,87,7,260,0.000,88,701,Mist,mist
2017,January,1,Sunday,0,Active,5,NCR,Nibe,Torvet,1,9240,56.983,9.639,DKK,MasterCard,3269,Withdrawal,,,56.981,9.639,2616483,Nibe,280.640,1020,93,9,250,0.590,9
2,500,Rain,light rain
2017,January,1,Sunday,0,Active,6,NCR,Fredericia,SjÃƒÂ¦llandsgade,33,7000,55.564,9.757,DKK,MasterCard,887,Withdrawal,,,55.566,9.753,2621951,Fredericia,281.150
,1014,93,7,230,0.290,92,500,Rain,light rain
2017,January,1,Sunday,0,Active,7,Diebold Nixdorf,Hjallerup,Hjallerup Centret,18,9320,57.168,10.148,DKK,Mastercard - on-us,4626,Withdrawal,,,57.165,10.146,262
0275,Hjallerup,280.640,1020,93,9,250,0.590,92,500,Rain,light rain
2017,January,1,Sunday,0,Active,8,NCR,GlyngÃƒÂ¸re,FÃƒÂ¦rgevej,1,7870,56.762,8.867,DKK,MasterCard,470,Withdrawal,,,56.793,8.853,2615964,Nykobing Mors,281.150,1
011,100,6,240,0.000,75,300,Drizzle,light intensity drizzle
2017,January,1,Sunday,0,Active,9,Diebold Nixdorf,Hadsund,Storegade,12,9560,56.716,10.114,DKK,VISA,8473,Withdrawal,,,56.715,10.117,2620952,Hadsund,280.640,102
0,93,9,250,0.590,92,500,Rain,light rain
2017,January,1,Sunday,0,Active,10,NCR,NÃƒÂ¸rresundby,Torvet,6,9400,57.059,9.922,DKK,Dankort,953,Withdrawal,,,57.048,9.919,2624886,Aalborg,280.640,1020,93,9,2
50,0.590,92,500,Rain,light rain
2017,January,1,Sunday,0,Active,11,NCR,Sauersvej,Fridtjof Nansens Vej,2,9210,57.023,9.940,DKK,Visa Dankort,9346,Withdrawal,,,57.048,9.935,2616235,NÃƒÂ¸rresund
by,280.640,1020,93,9,250,0.590,92,500,Rain,light rain
2017,January,1,Sunday,0,Inactive,2,NCR,Vejgaard,Hadsundvej,20,9000,57.043,9.950,DKK,Mastercard - on-us,3874,Withdrawal,,,57.048,9.935,2616235,NÃƒÂ¸rresundby,
280.640,1020,93,9,250,0.590,92,500,Rain,light rain
2017,January,1,Sunday,0,Inactive,12,NCR,ÃƒËœsterÃƒÂ¥  Duus,ÃƒËœsterÃƒÂ¥,12,9000,57.049,9.922,DKK,Mastercard - on-us,1329,Withdrawal,,,57.048,9.919,2624886,Aa
lborg,280.640,1020,93,9,250,0.590,92,500,Rain,light rain
2017,January,1,Sunday,0,Active,13,NCR,SÃƒÂ¦by,Vestergade,3,9300,57.334,10.515,DKK,Mastercard - on-us,5024,Withdrawal,,,57.441,10.537,2621927,Frederikshavn,28
1.140,1019,94,12,251,1.275,92,500,Rain,light rain
2017,January,1,Sunday,0,Inactive,14,NCR,HÃƒÂ¸rning,NÃƒÂ¸rrealle,12,8362,56.086,10.037,DKK,Visa Dankort - on-us,1133,Withdrawal,,,56.157,10.211,2624652,Arhus,
281.150,1012,87,5,250,0.000,92,300,Drizzle,light intensity drizzle
2017,January,1,Sunday,0,Active,15,NCR,Vestre,Kastetvej,36,9000,57.053,9.905,DKK,MasterCard,594,Withdrawal,,,57.048,9.919,2624886,Aalborg,280.640,1020,93,9,25
0,0.590,92,500,Rain,light rain
2017,January,1,Sunday,0,Inactive,12,NCR,ÃƒËœsterÃƒÂ¥  Duus,ÃƒËœsterÃƒÂ¥,12,9000,57.049,9.922,DKK,Mastercard - on-us,9570,Withdrawal,,,57.048,9.919,2624886,Aa
lborg,280.640,1020,93,9,250,0.590,92,500,Rain,light rain
```