16: Docker Tutorial: Apache Spark (spark-shell) & parquet-tools – csv to parquet on Cloudera quickstart

Posted on June 3, 2019

This extends Docker Tutorial: BigData on Cloudera quickstart via Docker.

Step 1: Run the container on a command line.

```
~/projects/docker-hadoop]$ docker run --hostname=qu
--privileged=true -t -i -v /Users/arulkumarankumar
--publish-all=true -p 8888:8888 -p 80:80 -p 7180:7
```

300+ Java Interview FAQs

300+ Java FAQs 🔥

16+ Java Key Areas Q&As

150+ Java Architect FAQs

80+ Java Code Quality Q&As

150+ Java Coding Q&As

300+ Big Data Interview FAQs

300+ Big Data FAQs 🔥

Tutorials - Big Data

TUT - ⏯ Starting Big Data

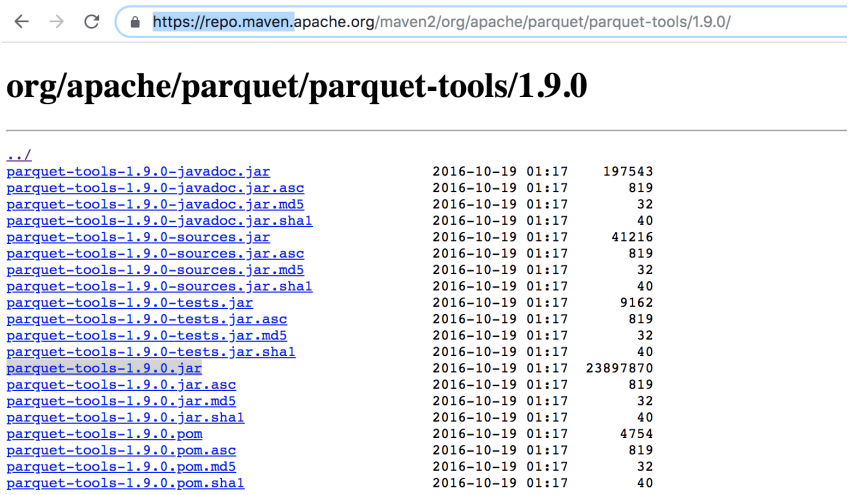
TUT - Starting Spark & Scala

Get parquet-tools

Step 2: Install **wget**. The "**uname -a**" gets you the info of the kernel.

```
[root@quickstart /]# sudo yum install wget
```

Step 3: Download "parquet-tools" from maven repository using wget.



parquet-tools at Maven repository

```
[root@quickstart /]# wget https://repo.maven.apach
```

Hive table over .csv

Step 4: Create a csv file "**employee.csv**" in the local file system.

```
[root@quickstart /]# touch employee.csv
[root@quickstart /]# vi employee.csv

```

```
John,Samuel,IT
```

```
2 Peter,Smith,Finance
3 Sean,Mendis,Marketing
```

Step 5: Copy this file onto HDFS file system.

```
1 [root@quickstart /]# hdfs dfs -copyFromLocal emplo
```

spark-shell

Step 6: Download Spark-csv from maven repository for Scala version 2.10.

```
1 [root@quickstart /]# wget https://repo.maven.apach
2
3 [root@quickstart /]# wget https://repo.maven.apach
4
```

Step 7: spark-shell is useful in interactive coding in Scala.

```
1 [root@quickstart /]# spark-shell --jars spark-csv_
2 ....
3 scala>
4
```

```
1 [root@quickstart /]# ls -ltr
2 ....
3 -rw-r--r--   1 root root   165361 Sep  5  2016 spa
4 -rw-r--r--   1 root root 23897870 Oct 19  2016 par
5 -rw-r--r--   1 root root    42400 Sep 19  2018 com
6 ...
7
```

Alternatively you can use **–packages**:

```
1 [root@quickstart /]# spark-shell --packages com.da
```

Step 8: Scala code to read from a csv file into a dataframe.

```
scala> val dfCsv = sqlContext.read.format("csv").
scala> dfCsv.show()
+-----+------+---------+
|   C0|    C1|       C2|
+-----+------+---------+
| John|Samuel|       IT|
|Peter| Smith|  Finance|
| Sean|Mendis|Marketing|
+-----+------+---------+


scala> dfCsv.write.parquet("hdfs:///user/root/spar
```

Step 9: Exit out of the spark-shell.

```
[root@quickstart /]# hdfs dfs -ls hdfs:///user/roo
Found 5 items
-rw-r--r--   1 root supergroup          0 2019-06-
-rw-r--r--   1 root supergroup        386 2019-06-
-rw-r--r--   1 root supergroup       1065 2019-06-
-rw-r--r--   1 root supergroup        760 2019-06-
3-bc74-140ead01143c.gz.parquet
-rw-r--r--   1 root supergroup        760 2019-06-
3-bc74-140ead01143c.gz.parquet
[root@quickstart /]#
```

parquet-tools

```
[root@quickstart /]# hadoop jar parquet-tools-1.9
af3-bc74-140ead01143c.gz.parquet

19/06/03 12:06:08 INFO hadoop.InternalParquetReco
C0 = John
C1 = Samuel
C2 = IT

C0 = Peter
C1 = Smith
C2 = Finance
```

```
[root@quickstart /]#

```

```
[root@quickstart /]# hadoop jar parquet-tools-1.9.(

19/06/03 12:12:23 INFO hadoop.InternalParquetRecor(
C0 = Sean
C1 = Mendis
C2 = Marketing

[root@quickstart /]#

```

‹ 15: Docker Tutorial: Hive & parquet-tools – csv to parquet on Cloudera quickstart

17: Docker Tutorial: sqoop import – on Cloudera quickstart ›

Disclaimer