

Java-Success.com

Prepare to fast-track, choose & go places with 800+ Java & Big Data Q&As with lots of code & diagrams.

[Home](#) [Why? ▾](#) [300+ Java FAQs ▾](#) [300+ Big Data FAQs ▾](#) [Courses ▾](#)[👤 Membership ▾](#) [Your Career ▾](#)[Home](#) > [bigdata-success.com](#) > [Tutorials - Big Data](#) > [TUT - Cloudera on Docker](#) > 15:

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

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

 Posted on [June 2, 2019](#)

CSV is a row based storage, and Parquet is **columnar** in nature, and it is designed from the ground up for efficient storage, compression and encoding, which gives better performance.

Run the cloudera/quickstart

This extends [Docker Tutorial: BigData on Cloudera quickstart via Docker](#).

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

Step 1: Run the container on a command line.

```
1 ~/projects/docker-hadoop]$ docker run --hostname=qu
2 --privileged=true -t -i -v /Users/arulkumarankumar
3 --publish-all=true -p 8888:8888 -p 80:80 -p 7180:71
```

Get parquet-tools

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

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

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

← → ↺ 🔒 https://repo.maven.apache.org/maven2/org/apache/parquet/parquet-tools/1.9.0/

org/apache/parquet/parquet-tools/1.9.0

../		
parquet-tools-1.9.0-javadoc.jar	2016-10-19 01:17	197543
parquet-tools-1.9.0-javadoc.jar.asc	2016-10-19 01:17	819
parquet-tools-1.9.0-javadoc.jar.md5	2016-10-19 01:17	32
parquet-tools-1.9.0-javadoc.jar.sha1	2016-10-19 01:17	40
parquet-tools-1.9.0-sources.jar	2016-10-19 01:17	41216
parquet-tools-1.9.0-sources.jar.asc	2016-10-19 01:17	819
parquet-tools-1.9.0-sources.jar.md5	2016-10-19 01:17	32
parquet-tools-1.9.0-sources.jar.sha1	2016-10-19 01:17	40
parquet-tools-1.9.0-tests.jar	2016-10-19 01:17	9162
parquet-tools-1.9.0-tests.jar.asc	2016-10-19 01:17	819
parquet-tools-1.9.0-tests.jar.md5	2016-10-19 01:17	32
parquet-tools-1.9.0-tests.jar.sha1	2016-10-19 01:17	40
parquet-tools-1.9.0.jar	2016-10-19 01:17	23897870
parquet-tools-1.9.0.jar.asc	2016-10-19 01:17	819
parquet-tools-1.9.0.jar.md5	2016-10-19 01:17	32
parquet-tools-1.9.0.jar.sha1	2016-10-19 01:17	40
parquet-tools-1.9.0.pom	2016-10-19 01:17	4754
parquet-tools-1.9.0.pom.asc	2016-10-19 01:17	819
parquet-tools-1.9.0.pom.md5	2016-10-19 01:17	32
parquet-tools-1.9.0.pom.sha1	2016-10-19 01:17	40

parquet-tools at Maven repository

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

Hive table over .csv

Step 4: Create a csv file “employee.csv” in the local file system.

TUT - Starting with Python

TUT - Kafka

TUT - Pig

TUT - Apache Storm

TUT - Spark Scala on Zeppelin

TUT - Cloudera

TUT - Cloudera on Docker

TUT - File Formats

TUT - Spark on Docker

TUT - Flume

TUT - Hadoop (HDFS)

TUT - HBase (NoSQL)

TUT - Hive (SQL)

TUT - Hadoop & Spark

TUT - MapReduce

TUT - Spark and Scala

TUT - Spark & Java

TUT - PySpark on Databricks

TUT - Zookeeper

800+ Java Interview Q&As

300+ Core Java Q&As

300+ Enterprise Java Q&As

150+ Java Frameworks Q&As

120+ Companion Tech Q&As

Tutorials - Enterprise Java

```
1 [root@quickstart /]# touch employee.csv
2 [root@quickstart /]# vi employee.csv
3
```

```
1 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 employ
```

Step 6: Connect to hive 2 via **beeline**. Beeline is a thin client that also uses the Hive JDBC driver but instead executes queries through HiveServer2, which allows multiple concurrent client connections and supports authentication.

```
1 [root@quickstart /]# beeline -u jdbc:hive2://quicl
2
```

Step 7: Create a database named “mydb”.

```
1 0: jdbc:hive2://quickstart.cloudera:10000> create
2
```

Step 8: An external table created over a folder in HDFS will not delete the file even if the table is dropped.

```
1 CREATE EXTERNAL TABLE IF NOT EXISTS mydb.tbl_employ
2 (first_name STRING,surname STRING, department STRI
3 ROW FORMAT DELIMITED
4 FIELDS TERMINATED BY ','
5 STORED AS TEXTFILE
6 LOCATION 'hdfs://quickstart.cloudera:8020/user/roo
7
```

Hive table over parquet

Step 9: An external table created over a folder in HDFS will not delete the file even if the table is dropped.

```
1 CREATE EXTERNAL TABLE IF NOT EXISTS mydb.tbl_parquet
2 (first_name STRING, surname STRING, department STRING)
3 STORED AS PARQUET
4 LOCATION 'hdfs://quickstart.cloudera:8020/user/root/parquet'
5
```

Insert data into tbl_parquet_employee

```
1 0: jdbc:hive2://quickstart.cloudera:10000>0: jdbc:
2
3
4 0: jdbc:hive2://quickstart.cloudera:10000>0: jdbc:
5 ....
6 +-----+-----+
7 | tbl_parquet_employee.first_name | tbl_parquet_e
8 +-----+-----+
9 | John                             | Samuel
10 | Peter                            | Smith
11 | Sean                             | Mendis
12 +-----+-----+
13 3 rows selected (0.58 seconds)
14 0: jdbc:hive2://quickstart.cloudera:10000>
```

List the parquet file

Step 10: Exit out with “ctrl+C”. List the file:

```
1 [root@quickstart /]# hdfs dfs -ls hdfs://quickstart
2 Found 1 items
3 -rwxrwxrwx    1 anonymous supergroup      540 2019
4
```

cat the parquet file with parquet-tools on HDFS

```
1 [root@quickstart /]# hadoop jar parquet-tools-1.9
2 ....
3 first_name = John
4 surname = Samuel
5 department = IT
6
7 first_name = Peter
8 surname = Smith
9 department = Finance
10
11 first_name = Sean
12 surname = Mendis
13 department = Marketing
14
15 [root@quickstart /]#
16
```

copy the parquet file to local file system

```
1 [root@quickstart /]# [root@quickstart /]# hdfs dfs
```

cat the parquet file with parquet-tools on local file system

```
1 [root@quickstart /]# hadoop jar parquet-tools-1.9
2 ....
3 first_name = John
4 surname = Samuel
5 department = IT
6
7 first_name = Peter
8 surname = Smith
9 department = Finance
10
11 first_name = Sean
12 surname = Mendis
13 department = Marketing
14
15 [root@quickstart /]#
16
```

Unlike “**hdfs**” commands that work only on hdfs, “**hadoop**” command will work on both “hdfs” and “local” file systems.

◀ 14: Docker Tutorial: Hive (via beeline) on Cloudera quickstart

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

Disclaimer

The contents in this Java-Success are copyrighted and from EmpoweringTech pty ltd. The EmpoweringTech pty ltd has the right to correct or enhance the current content without any prior notice. These are general advice only, and one needs to take his/her own circumstances into consideration. The EmpoweringTech pty ltd will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. Links to external sites do not imply endorsement of the linked-to sites. [Privacy Policy](#).