18: Docker Tutorial: sqoop export – on Cloudera quickstart

Posted on June 5, 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
```

Connect to MySQL DB

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 2: Hive & Impala store the metadata in a database like MySQL. The Hive **metastore service** connects to the metastore Database to store metadata. You can connect to the mysql RDBMS as shown below:

```
[root@quickstart /]# mysql -uroot -pcloudera
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cm                 |
| firehose           |
| hue                |
| metastore          |
| mysql              |
| nav                |
| navms              |
| oozie              |
| retail_db          |
| rman               |
| sentry             |
+--------------------+
12 rows in set (0.01 sec)


```

Create a DB & a table

Step 3: Create a new database and switch to that database.

```
mysql> create database test;
mysql> use test;
Database changed
mysql>

```

Step 4: Create a new table.

```
mysql> CREATE TABLE employee (firstname varchar(10
surname varchar(100) not null,
```

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

```
department varchar(100) not null
);

```

Create a file on HDFS

Step 5: Create a file on the local file system.

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

```

```
John, Samuel, IT
Peter, Smith, Finance
Sean, Mendis, Marketing
```

Step 6: Put to HDFS folder "/user/root/".

```
[root@quickstart /]# hdfs dfs -put employee.csv  /

```

List the folder on HDFS.

```
[root@quickstart /]# hdfs dfs -ls /user/root/
Found 1 items
-rw-r--r--   1 root supergroup         63 2019-06-

```

Export HDFS file into MySQL table

Step 7: Export HDFS data into an empty table in MySQL RDBMS.

```
[root@quickstart /]# sqoop export --connect jdbc:my
--username root --password cloudera \
--table employee  \
--export-dir /user/root/employee.csv --validate

```

Connect to MySQL

Step 8: Connect to MySQL table and check if the data inserted via sqoop export.

```
[root@quickstart /]# mysql -uroot -pcloudera

```

```
mysql> SELECT * FROM test.employee;
+-----------+---------+------------+
| firstname | surname | department |
+-----------+---------+------------+
| John      |  Samuel |  IT        |
| Peter     |  Smith  |  Finance   |
| Sean      |  Mendis |  Marketing |
+-----------+---------+------------+
3 rows in set (0.00 sec)

mysql>

```

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

19: Docker Tutorial: Apache Spark SQL – on Cloudera quickstart ›

Disclaimer