

# 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](#) > 19:

Docker Tutorial: Apache Spark SQL – on Cloudera quickstart

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

 Posted on [June 7, 2019](#)

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

**Step 1:** Run the container on a command line.

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

### Create a .csv in HDFS

### 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:** Create a csv file “employee.csv” in the local file system.

```
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 3:** Copy this file onto HDFS file system.

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

## Start spark-shell & create tables

**Step 4:** Start spark-shell.

```
1 [root@quickstart /]# spark-shell
```

**Step 5:** Create tables via Spark SQL.

```
1 scala> sqlContext.sql("CREATE TABLE employee (first
```

## Load data

**Step 6:** Load data from HDFS into the table via Spark SQL.

```
1 scala> sqlContext.sql("LOAD DATA INPATH 'hdfs:///u
```

## Create a Dataframe

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



**Step 7:** Create a Dataframe via Spark SQL, and show the results.

```
1 scala> val employeeDf = sqlContext.sql("SELECT * FROM employee")
2 employeeDf: org.apache.spark.sql.DataFrame = [first_name, surname, department]
3
4 scala> employeeDf.show()
5 +-----+-----+-----+
6 |first_name|surname|department|
7 +-----+-----+-----+
8 |      John|Samuel |         IT|
9 |     Peter|Smith  |    Finance|
10 |      Sean|Mendis |Marketing|
11 +-----+-----+-----+
12
13 scala>
```

## Filter by name

**Step 8:** Using the “filter” on a Dataframe. Please note “===” is used as opposed to “==” or “=”.

```
1 scala> employeeDf.filter(employeeDf("first_name") === "John")
2 +-----+-----+-----+
3 |first_name|surname|department|
4 +-----+-----+-----+
5 |      John|Samuel |         IT|
6 +-----+-----+-----+
7
8
9 scala>
```

## Save Dataframe as a table

```
1 scala> employeeDf.write.saveAsTable("employee_2")
```

◀ 18: Docker Tutorial: sqoop export – on Cloudera quickstart

20: Docker Tutorial: Apache Spark (spark-submit) in Java 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](#)

© 2022 [java-success.com](#)