

# 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](#) › 21:

Docker Tutorial: Apache Spark (spark-submit) in Scala on Cloudera quickstart

## 21: Docker Tutorial: Apache Spark (spark-submit) in Scala on Cloudera quickstart

 Posted on [June 8, 2019](#)

Extends [20: Docker Tutorial: Apache Spark-submit in Java – on Cloudera quickstart](#), and [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 7180:7
```

### 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

# Install Java 8

**Step 2:** Install Java 8. The “cloudera/quickstart” comes with Java 7.

```
1 [root@quickstart /]# sudo yum install java-1.8.0-openjdk
2
```

Check the version:

```
1 [root@quickstart /]# java -version
2 openjdk version "1.8.0_212"
3 OpenJDK Runtime Environment (build 1.8.0_212-b04)
4 OpenJDK 64-Bit Server VM (build 25.212-b04, mixed mode)
5 [root@quickstart /]#
6
```

## update-alternatives

```
1 [root@quickstart /]# which java
2 /usr/bin/java
3 [root@quickstart /]# ls -ltr /usr/bin/java
4 lrwxrwxrwx 1 root root 22 Jun  8 05:42 /usr/bin/java -> /usr/lib/jvm/java-1.8.0-openjdk.x86_64/bin/java
5 [root@quickstart /]#
6
```

If you have multiple versions or installations of Java, you can list them as shown below:

```
1 [root@quickstart /]# update-alternatives --config java
2
3 There is 1 program that provides 'java'.
4
5   Selection    Command
6   -----
7   *+ 1         /usr/lib/jvm/jre-1.8.0-openjdk.x86_64/bin/java
8
9 Enter to keep the current selection[+], or type selection to change
10
11
```

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



## Install scala & sbt

We will install Scala **2.10.7** from <https://www.scala-lang.org/download/2.10.7.html> , which uses Java 8.

### Step 3: Install Scala via **curl**.

```
1 [root@quickstart /]# curl -O -L https://downloads.
```

Copy to “/opt” folder and untar.

```
1 [root@quickstart /]# cp scala-2.10.7.tgz /opt
2 [root@quickstart /]# cd /opt
3 [root@quickstart opt]# tar xzf scala-2.10.7.tgz
4 [root@quickstart opt]# rm -f scala-2.10.7.tgz
5 [root@quickstart opt]# ls -ltr
6 total 16
7 drwxr-xr-x 5 cloudera cloudera 4096 Aug 1
8 drwxr-xr-x 4 cloudera cloudera 4096 Aug 2
9 drwxr-xr-x 4 cloudera-scm cloudera-scm 4096 Apr 6
10 drwxrwxr-x 9 1001 1001 4096 Nov 3
11 [root@quickstart opt]#
12
```

### Step 4: Install sbt via **curl**.

Go to <https://www.scala-lang.org/download/2.10.7.html> and click on “Download SBT”.

```
1 [root@quickstart opt]# curl -O -L https://piccolo.
2
```

Untar the tgz file.

```
1 [root@quickstart opt]# tar xzf sbt-1.2.8.tgz
2 [root@quickstart opt]# rm -f sbt-1.2.8.tgz
3 [root@quickstart opt]# ls -ltr
```

```
4 total 20
5 drwxr-xr-x 5 cloudera cloudera 4096 Aug 1
6 drwxr-xr-x 4 cloudera cloudera 4096 Aug 2
7 drwxr-xr-x 4 cloudera-scm cloudera-scm 4096 Apr 6
8 drwxrwxr-x 9 1001 1001 4096 Nov 3
9 drwxrwxr-x 5 1000 1000 4096 Dec 30
10 [root@quickstart opt]#
11
```

## Update ~/.bashrc

**Step 5:** If you are a root user update the ~/.bashrc file, and if you are a any other user update the ~/.bash\_profile so that “scala” and “sbt” command can be run from any folder.

```
1 [root@quickstart opt]# vi ~/.bashrc
2
1 # .bashrc
2
3 # User specific aliases and functions
4
5 alias rm='rm -i'
6 alias cp='cp -i'
7 alias mv='mv -i'
8
9 # Source global definitions
10 if [ -f /etc/bashrc ]; then
11     . /etc/bashrc
12 fi
13
14 SCALA_HOME=/opt/scala-2.10.7
15 SBT_HOME=/opt/sbt
16
17 export PATH=$PATH:$SCALA_HOME/bin:$SBT_HOME/bin
18
```

### Activate:

```
1 [root@quickstart opt]# source ~/.bashrc
2
```

Check scala command prompt:

```
1 [root@quickstart opt]# scala
2 Welcome to Scala version 2.10.7 (OpenJDK 64-Bit Se
3 Type in expressions to have them evaluated.
4 Type :help for more information.
5
6 scala>
```

Check sbt command prompt:

```
1 [root@quickstart opt]# sbt
2 [info] Updated file /opt/project/build.properties:
3 [info] Loading project definition from /opt/project
4 [info] Updating ProjectRef(uri("file:/opt/project/
5 [info] Done updating.
6 [info] Set current project to opt (in build file:/o
7 [info] sbt server started at local:///root/.sbt/1.0
8 sbt:opt>
```

## Create Scala project structure

**Step 6:** Unlike maven archetype:generate, sbt does not create the basic project structure. We can create the sbt project structure with a shell script.

```
1 [root@quickstart ~]# cd ~
2 [root@quickstart ~]# pwd
3 /root
4 [root@quickstart ~]# mkdir projects
5 [root@quickstart ~]# cd projects/
6 [root@quickstart projects]# mkdir my-app
7 [root@quickstart projects]# cd my-app
8 [root@quickstart my-app]# vi mkdirs4sbt.sh
9
```

The “mkdirs4sbt.sh”

```
1 #!/bin/sh
2 mkdir -p src/{main,test}/{java,resources,scala}
3 mkdir lib project target
4
```

```
5 # create an initial build.sbt file
6 echo 'name := "my-app"'
7 version := "1.0"
8 scalaVersion := "2.10.7"' > build.sbt
9
```

```
1 [root@quickstart my-app]# chmod 755 mkdirs4sbt.sh
2 [root@quickstart my-app]# ./mkdirs4sbt.sh
3
```

```
1 [root@quickstart my-app]# tree
2 .
3 |— build.sbt
4 |— lib
5 |— mkdirs4sbt.sh
6 |— project
7 |— src
8 |   |— main
9 |       |— java
10 |       |— resources
11 |       |— scala
12 |   |— test
13 |       |— java
14 |       |— resources
15 |       |— scala
16 |— target
17
18 12 directories, 2 files
19 [root@quickstart my-app]#
20
```

## Add spark dependency in build.sbt

**Step 7:** To write Spark code spark-core api library is required.

```
1 [root@quickstart my-app]# vi build.sbt
2
```

```
1 name := "my-app"
2 version := "1.0"
3 scalaVersion := "2.10.7"
4
5 libraryDependencies += "org.apache.spark" %% "spark-core" % "2.4.0"
6
```

## Create the Spark job in Scala

**Step 8:** Create the package “com.mycompany.app”.

```
1 [root@quickstart my-app]# mkdir -p src/main/scala/com/mycompany/app
```

**Step 9:** Create “SimpleSpark.scala”.

```
1 [root@quickstart my-app]# vi src/main/scala/com/mycompany/app/SimpleSpark.scala
```

```
1 package com.mycompany.app
2
3 import org.apache.spark.SparkContext
4 import org.apache.spark.SparkContext._
5 import org.apache.spark.SparkConf
6
7
8 object SimpleSpark {
9     def main(args: Array[String]) {
10         val conf = new SparkConf().setAppName("SimpleSpark")
11         val sc = new SparkContext(conf)
12         val data = List("John", "Peter", "Samuel")
13         val rdd = sc.parallelize(data)
14         rdd.foreach(println)
15     }
16 }
17
```

## Compile & Package with sbt

**Step 10:** Package it with “sbt”

```
1 [root@quickstart my-app]# sbt package
2 ....
3
```

```
1 [root@quickstart my-app]# ls -ltr target/scala-2.10.4
2 total 12
3 drwxr-xr-x 5 root root 4096 Jun  8 09:06 resolution-cache
4 drwxr-xr-x 3 root root 4096 Jun  8 09:06 classes
5 -rw-r--r-- 1 root root 2334 Jun  8 09:06 my-app_2.10.4.jar
6 [root@quickstart my-app]#
7
```

# spark-submit to run the spark job

**Step 11:** Run the Spark job in the jar file via Spark-submit command.

## Local client mode

```
1 [root@quickstart my-app]# spark-submit \
2 --class com.mycompany.app.SimpleSpark \
3 --master local \
4 --deploy-mode client \
5 target/scala-2.10/my-app_2.10-1.0.jar
6
```

```
1 ....
2 John
3 Peter
4 Samuel
5 ....
6
```

## Local cluster mode

```
1 [root@quickstart my-app]# spark-submit \
2 --class com.mycompany.app.SimpleSpark \
3 --master yarn \
4 --deploy-mode client \
5 target/scala-2.10/my-app_2.10-1.0.jar
6
```

◀ 20: Docker Tutorial: Apache Spark (spark-submit) in Java on Cloudera quickstart

22: Docker Tutorial: Apache Spark (spark-submit) in Python 2.6 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](https://www.java-success.com)