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.

```
~/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

Install Java 8

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

```
[root@quickstart /]# sudo yum install java-1.8.0-op
```

Check the version:

```
[root@quickstart /]# java -version
openjdk version "1.8.0_212"
OpenJDK Runtime Environment (build 1.8.0_212-b04)
OpenJDK 64-Bit Server VM (build 25.212-b04, mixed r
[root@quickstart /]#
```

update-alternatives

```
[root@quickstart /]# which java
/usr/bin/java
[root@quickstart /]# ls -ltr /usr/bin/java
lrwxrwxrwx 1 root root 22 Jun  8 05:42 /usr/bin/jav
[root@quickstart /]#
```

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

```
[root@quickstart /]# update-alternatives --config

There is 1 program that provides 'java'.

  Selection    Command
-----------------------------------------------
*+ 1           /usr/lib/jvm/jre-1.8.0-openjdk.x86_

Enter to keep the current selection[+], or type se
```

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

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**.

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

Copy to "/opt" folder and untar.

```
[root@quickstart /]# cp scala-2.10.7.tgz /opt
[root@quickstart /]# cd /opt
[root@quickstart opt]# tar xfz scala-2.10.7.tgz
[root@quickstart opt]# rm -f scala-2.10.7.tgz
[root@quickstart opt]# ls -ltr
total 16
drwxr-xr-x 5 cloudera     cloudera     4096 Aug  1
drwxr-xr-x 4 cloudera     cloudera     4096 Aug 21
drwxr-xr-x 4 cloudera-scm cloudera-scm 4096 Apr  6
drwxrwxr-x 9         1001         1001 4096 Nov  3
[root@quickstart opt]#
```

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

Go to https://www.scala-lang.org/download/2.10.7.html and click on "**Download SBT**".

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

Untar tha tgz file.

```
[root@quickstart opt]# tar xfz sbt-1.2.8.tgz
[root@quickstart opt]# rm -f sbt-1.2.8.tgz
[root@quickstart opt]# ls -ltr
```

```
total 20
drwxr-xr-x 5 cloudera     cloudera     4096 Aug  1
drwxr-xr-x 4 cloudera     cloudera     4096 Aug 27
drwxr-xr-x 4 cloudera-scm cloudera-scm 4096 Apr  6
drwxrwxr-x 9         1001         1001 4096 Nov  3
drwxrwxr-x 5         1000         1000 4096 Dec 30
[root@quickstart opt]#
```

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.

```
[root@quickstart opt]# vi ~/.bashrc
```

```
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

SCALA_HOME=/opt/scala-2.10.7
SBT_HOME=/opt/sbt

export PATH=$PATH:$SCALA_HOME/bin:$SBT_HOME/bin
```

Activate:

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

Check scala command prompt:

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

scala>

```

Check sbt command prompt:

```
[root@quickstart opt]# sbt
[info] Updated file /opt/project/build.properties:
[info] Loading project definition from /opt/projec
[info] Updating ProjectRef(uri("file:/opt/project/
[info] Done updating.
[info] Set current project to opt (in build file:/
[info] sbt server started at local:///root/.sbt/1.
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.

```
[root@quickstart ~]# cd ~
[root@quickstart ~]# pwd
/root
[root@quickstart ~]# mkdir projects
[root@quickstart ~]# cd projects/
[root@quickstart projects]# mkdir my-app
[root@quickstart projects]# cd my-app
[root@quickstart my-app]# vi mkdirs4sbt.sh

```

The "**mkdirs4sbt.sh**"

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

```

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

```

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

```

```
[root@quickstart my-app]# tree
.
├── build.sbt
├── lib
├── mkdirs4sbt.sh
├── project
├── src
│   ├── main
│   │   ├── java
│   │   ├── resources
│   │   └── scala
│   └── test
│       ├── java
│       ├── resources
│       └── scala
└── target

12 directories, 2 files
[root@quickstart my-app]#

```

Add spark dependency in build.sbt

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

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

```

```
name := "my-app"
version := "1.0"
scalaVersion := "2.10.7"

libraryDependencies += "org.apache.spark" %% "sparl

```

Create the Spark job in Scala

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

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

Step 9: Create “SimpleSpark.scala”.

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

```
package com.mycompany.app

import org.apache.spark.SparkContext
import org.apache.spark.SparkContext._
import org.apache.spark.SparkConf


object SimpleSpark {
  def main(args: Array[String]) {
    val conf = new SparkConf().setAppName("Simple
    val sc = new SparkContext(conf)
    val data = List("John", "Peter", "Samuel")
    val rdd = sc.parallelize(data)
    rdd.foreach(println)
  }
}

```

Compile & Package with sbt

Step 10: Package it with “sbt”

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

```

```
[root@quickstart my-app]# ls -ltr target/scala-2.1
total 12
drwxr-xr-x 5 root root 4096 Jun  8 09:06 resolutio
drwxr-xr-x 3 root root 4096 Jun  8 09:06 classes
-rw-r--r-- 1 root root 2334 Jun  8 09:06 my-app_2.
[root@quickstart my-app]#

```

spark-submit to run the spark job

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

Local client mode

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

```

```
....
John
Peter
Samuel
....

```

Local cluster mode

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

```

‹ 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