# Recommending music using Audioscrobbler dataset

> Big Data applications in the cloud
>
> **Surbhi Sonkiya** (EIT Digital Master School)

## Introduction

This document presents several ways of deploying a music recommender system. Firstly, it describes the steps to deploy it with spark on local machine and then progress to execute it on the docker container. Furthermore, it details the steps to store the data on HDFS and run the code in the master-slave cluster mode of spark using docker containers for spark master, spark slave, HDFS namenode and HDFS datanode.

## Run recommender and manage dependencies with SBT instead of Maven

### Trim dataset to top 1000 rows

Download the audioscrobbler dataset from provided webpage [1]. Extract top 10000 rows using below command, store it in a file and use it as the dataset henceforth.

```
head -n 10000 user_artist_data.txt > user_artist_data_10000.txt
```

Download music recommender code from provided webpage [2] . Build a new SBT project in IntelliJ and import the code. Convert pom to sbt as shown below.

### Translate pom.xml to build.sbt

```
name := "RecommenderBigData"
version := "0.1"
scalaVersion := "2.11.8"

libraryDependencies ++= Seq(
  "org.apache.spark" %% "spark-core" % "2.2.0",
  "org.apache.spark" %% "spark-sql" % "2.2.0",
  "org.apache.spark" %% "spark-mllib" % "2.0.1"
```

### Modify the music recommender code

After importing the code and resolving the dependencies using build.sbt, modify RunRecommender class to accomplish below requirements.

1. **Take user ID as argument**

Added below command in the main function to take input from user

```
val id = args(0).toInt
```

Modified function definition of below functions to pass the user input

```
  runRecommender.model(id, rawUserArtistData, rawArtistData, rawArtistAlias)
  runRecommender.recommend(id, rawUserArtistData, rawArtistData, rawArtistAlias)
```

Within function `def model` and `def recommend`, add below line

```
val userID = id
```

2. **Replace the path of the dataset with the path where dataset is downloaded on local machine**

```
    spark.sparkContext.setCheckpointDir("file:/home/surbhi/Downloads/profiledata_06-May-2005/")

    val base = "file:/home/surbhi/Downloads/profiledata_06-May-2005/"
```

Build the jar using below command in the IntelliJ terminal

```
sbt package
```

Execute `spark-submit` command in console of the local machine

```
$ <path-of-spark-directory>/bin/spark-submit --class com.cloudera.datascience.recommender.RunRecommender --master local[*] --deploy-mode client /home/surbhi/RecommenderBigData/target/scala-2.11/recommenderbigdata_2.11-0.1.jar 1000002
```

| **Options of `spark-submit`** | **Explanation** |
|---|---|
| --class | path of the class that has main method |
| --master local[*] | launch spark on localhost |
| --deploy-mode client | deploy locally on external client |
| /home/.../recommenderbigdata_2.11-0.1.jar | path of the jar file on local machine |
| 1000002 | value for user ID passed to main method |

**Output**

```
surbhi@surbhi-Lenovo-G50-80: ~/Downloads/spark-2.2.0-bin-hadoop2.7
        at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)
surbhi@surbhi-Lenovo-G50-80:~/Downloads/spark-2.2.0-bin-hadoop2.7$ ./bin/spark-s
ubmit --class com.cloudera.datascience.recommender.RunRecommender --master local
[*] --deploy-mode client /home/surbhi/RecommenderBigData/target/scala-2.11/recom
menderbigdata_2.11-0.1.jar 1000002
18/11/30 14:28:14 WARN NativeCodeLoader: Unable to load native-hadoop library fo
r your platform... using builtin-java classes where applicable
1000002 1 55
1000002 1000006 33
1000002 1000007 8
1000002 1000009 144
1000002 1000010 314
+---------+---------+-----------+-----------+
|min(user)|max(user)|min(artist)|max(artist)|
+---------+---------+-----------+-----------+
|  1000002|  1000072|          1|   10783758|
+---------+---------+-----------+-----------+

+-------+----------------+
|     id|            name|
+-------+----------------+
|1208690|Collective Souls|
|1003926| Collective Soul|
+-------+----------------+
```

```
surbhi@surbhi-Lenovo-G50-80: ~/Downloads/spark-2.2.0-bin-hadoop2.7
|   4061|      Marilyn Manson|
|2003588|                KoЯn|
+-------+--------------------+

0.4799129322570132
(0.8529914065992438,(5,1.0,1.0))
(0.8374948894505259,(5,1.0E-4,40.0))
(0.8052610532750009,(5,1.0E-4,1.0))
(0.8019728705214465,(30,1.0,1.0))
(0.7901670179826871,(5,1.0,40.0))
(0.7504540450716947,(30,1.0,40.0))
(0.6551478427484821,(30,1.0E-4,1.0))
(0.6214369405117961,(30,1.0E-4,40.0))
+--------------------+
|                name|
+--------------------+
|        Machine Head|
|                 AFI|
|       Alkaline Trio|
|         Propagandhi|
|The Presidents of...|
+--------------------+

surbhi@surbhi-Lenovo-G50-80:~/Downloads/spark-2.2.0-bin-hadoop2.7$
```

## Deploy on Docker

Clone the P7h github [3] repository on the local machine. Navigate inside the cloned folder and execute commands on the terminal.

```
$ sudo docker build -t p7hb/docker-spark .
```

The `docker build` command builds Docker images from the Dockerfile downloaded as part of P7h github repository.

```
$ sudo docker pull p7hb/docker-spark
```

Check the location of spark directory in the docker container after docker container is launched.

```
$ whereis spark
```

**In RunRecommender class: modify the path of the dataset with the path of the data volume that will be mounted on the docker container.**

```
spark.sparkContext.setCheckpointDir("/home/")
val base = "/home/"
```

Build the jar using below command in the IntelliJ terminal

```
sbt package
```

Execute `docker pull` command pulls Docker image from a registry.

```
$ sudo docker run -it -p 4040:4040 -p 8082:8080 -p 8081:8081 -v
/home/surbhi/RecommenderBigData/target/scala-2.11:/home -h spark --name=spark
p7hb/docker-spark
```

The docker run command creates a writeable container layer over the specified image and then starts using specific command [4].

| **Options of `docker run`** | **Explanation** |
|---|---|
| -it | interactive mode |
| -p | publish ports. Left side is ports of local machine and right side is ports of container. |
| -v | mount volume from given path of the local machine to the path of the container. Both the paths are separated by a colon (:) in between. |
| -h | container host name |
| --name | assign name to the container |

To launch spark-submit inside the running container, first enter inside the running container using `docker exec` command.

```
$ sudo docker exec -it <name-of-the-container> bash
```

Execute `spark-submit` command inside docker container

```
<path-of-spark-directory>/bin/spark-submit --class
com.cloudera.datascience.recommender.RunRecommender --master local[*]
/home/recommenderbigdata_2.11-0.1.jar 1000002
```

**Output**

```
surbhi@surbhi-Lenovo-G50-80: ~/Downloads/docker-spark-master
Using default tag: latest
latest: Pulling from p7hb/docker-spark
Digest: sha256:92854d2edbb569b90274b104fa1d7f0f76f5e5ea5df35f8dd601e7cde9bbb692
Status: Downloaded newer image for p7hb/docker-spark:latest
surbhi@surbhi-Lenovo-G50-80:~/Downloads/docker-spark-master$ sudo docker run -it
 -p 4040:4040 -p 8082:8080 -p 8081:8081 -v /home/surbhi/RecommenderBigData/targe
t/scala-2.11:/home -h spark --name=spark p7hb/docker-spark
root@spark:~# whereis spark
spark: /usr/local/spark
root@spark:~# /usr/local/spark/bin/spark-submit --class com.cloudera.datascience
.recommender.RunRecommender --master local[*] /home/recommenderbigdata_2.11-0.1.
jar 1000002
1000002 1 55
1000002 1000006 33
1000002 1000007 8
1000002 1000009 144
1000002 1000010 314
+---------+---------+-----------+-----------+
|min(user)|max(user)|min(artist)|max(artist)|
+---------+---------+-----------+-----------+
|  1000002|  1000072|          1|   10783758|
+---------+---------+-----------+-----------+
```

```
surbhi@surbhi-Lenovo-G50-80: ~/Downloads/docker-spark-master
|1000660|             Soulwax|
|   1413|   The Dandy Warhols|
+-------+--------------------+

0.4284541956964573
(0.829804569486633,(30,1.0,1.0))
(0.814772266770665,(5,1.0E-4,40.0))
(0.8106639208846173,(5,1.0,1.0))
(0.7920120153834406,(5,1.0,40.0))
(0.785504955519975,(5,1.0E-4,1.0))
(0.7610902289453347,(30,1.0,40.0))
(0.7006150732302384,(30,1.0E-4,40.0))
(0.660593889145396,(30,1.0E-4,1.0))
+--------------------+
|                name|
+--------------------+
|The Jesus and Mar...|
|       Cocteau Twins|
|           Morrissey|
|Nick Cave and the...|
|         Syd Barrett|
+--------------------+

root@spark:~#
```

## Deploy on several nodes (1 Master + N Slaves)

Clone the gettyimage github [5] repository on the local machine. Navigate inside the cloned folder and modify `docker-compose.yml` the file as below.

**Modify docker-compose.yml file**

```
master:
  image: gettyimages/spark:2.2.0-hadoop-2.7
  command: bin/spark-class org.apache.spark.deploy.master.Master -h master
  hostname: master
  environment:
    MASTER: spark://master:7077
...
  volumes:
    - ./conf/master:/conf
    - ./data:/tmp/data
worker:
  image: gettyimages/spark:2.2.0-hadoop-2.7
  command: bin/spark-class org.apache.spark.deploy.worker.Worker spark://master:7077
  hostname: worker
...
  ports:
    - 8081:8081
  volumes:
    - ./conf/worker:/conf
    - ./data:/tmp/data
```

**In RunRecommender class: modify the path of the dataset with the path of the data volume that will be mounted on the docker container**

```
spark.sparkContext.setCheckpointDir("/tmp/data/")
val base = "/tmp/data/"
```

Build the jar using below command in the IntelliJ terminal

```
sbt package
```

Copy datasets and jar file inside `data` folder. This folder is present inside the cloned repository.

**For 1 Master + 1 Slave**

```
$ sudo docker-compose up
```

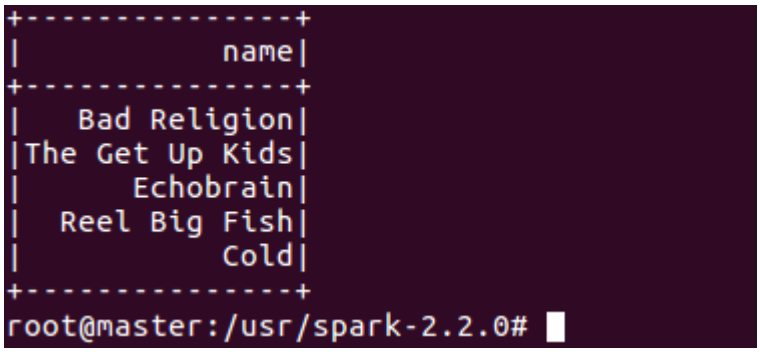
Enter inside the running `master` docker container

```
$ sudo docker exec -it <name-of-the-master-container> bash
```

Execute `spark-submit` in **client mode** inside the master docker container

```
<path-of-spark-directory>/bin/spark-submit --class
com.cloudera.datascience.recommender.RunRecommender --deploy-mode client
/tmp/data/recommenderbigdata_2.11-0.1.jar 1000002
```

**Output**




Execute `spark-submit` in **cluster mode** inside the master docker container

```
<path-of-spark-directory>/bin/spark-submit --class
com.cloudera.datascience.recommender.RunRecommender --supervise --master
spark://master:6066 --deploy-mode cluster /tmp/data/recommenderbigdata_2.11-0.1.jar
1000002
```

| **More options of `spark-submit` explored** | **Explanation** |
|---|---|
| --supervise | to monitor driver program from the master node and reset it in case it dies. |
| --master spark://master:6066 | master URL for the cluster; REST API of spark is at port 6066 |
| --deploy-mode cluster | deploy spark driver on worker nodes in cluster mode |

**Output**

Spark 2.2.0 **Spark Master at spark://master:7077**

**URL:** spark://master:7077
**REST URL:** spark://master:6066 *(cluster mode)*
**Alive Workers:** 1
**Cores in use:** 2 Total, 2 Used
**Memory in use:** 3.0 GB Total, 2.0 GB Used
**Applications:** 1 Running, 3 Completed
**Drivers:** 1 Running, 1 Completed
**Status:** ALIVE

**Workers**

| Worker Id | Address | State | Cores | Memory |
|---|---|---|---|---|
| worker-20181130151929-172.17.0.3-8881 | 172.17.0.3:8881 | ALIVE | 2 (2 Used) | 3.0 GB (2.0 GB Used) |

**Running Applications**

| Application ID | Name | Cores | Memory per Executor | Submitted Time | User | State | Duration |
|---|---|---|---|---|---|---|---|
| app-20181130154128-0003 (kill) | com.cloudera.datascience.recommender.RunRecommender | 1 | 1024.0 MB | 2018/11/30 15:41:28 | root | RUNNING | 4.1 min |

**Running Drivers**

| Submission ID | Submitted Time | Worker | State | Cores | Memory | Main Class |
|---|---|---|---|---|---|---|
| driver-20181130154125-0001 (kill) | 2018/11/30 15:41:25 | worker-20181130151929-172.17.0.3-8881 | RUNNING | 1 | 1024.0 MB | com.cloudera.datascience.recommender.RunRecommender |

Spark 2.2.0 **stdout log page for driver-20181130154125-0001**

Back to Master

Showing 5556 Bytes: 0 - 5556 of 5556

```
+-------+-----------------+
|1000343|    Skunk Anansie|
|   1168|         Babybird|
|   4629|          Unknown|
|1001172|       60ft Dolls|
|   1413|The Dandy Warhols|
+-------+-----------------+

0.4168996394338809
(0.7993388363165866,(5,1.0,1.0))
(0.7917516064800765,(5,1.0E-4,1.0))
(0.7854812230295221,(30,1.0,1.0))
(0.7816360999702199,(5,1.0E-4,40.0))
(0.7786191929078892,(30,1.0,40.0))
(0.7769624973134955,(5,1.0,40.0))
(0.6701574892236919,(30,1.0E-4,1.0))
(0.6670718157567418,(30,1.0E-4,40.0))
+-------------+
|         name|
+-------------+
| Howard Shore|
|     Bon Jovi|
|Guns N' Roses|
|   Jamiroquai|
|        Wham!|
+-------------+
```

Load New

### For 1 Master + 2 Slaves

Navigate inside the cloned folder and modify `docker-compose.yml` the file as below. Note: comment the ports for worker.

```
  master:
    image: gettyimages/spark:2.2.0-hadoop-2.7
    command: bin/spark-class org.apache.spark.deploy.master.Master -h master
    hostname: master
    environment:
      MASTER: spark://master:7077
  ...
    volumes:
```

```
      - ./conf/master:/conf
      - ./data:/tmp/data
  worker:
    image: gettyimages/spark:2.2.0-hadoop-2.7
    command: bin/spark-class org.apache.spark.deploy.worker.Worker spark://master:7077
    hostname: worker
  ...
   # ports:
   #   - 8081:8081
    volumes:
      - ./conf/worker:/conf
      - ./data:/tmp/data
```

Execute below docker commands to start docker containers and scale up to two workers instead of one.

```
$ sudo docker-compose up
$ sudo docker-compose scale worker=2
```

```
surbhi@surbhi-Lenovo-G50-80:~/Downloads/docker$ sudo docker-compose scale worker
=2
[sudo] password for surbhi:
WARNING: The scale command is deprecated. Use the up command with the --scale fl
ag instead.
Starting docker_worker_1 ... done
Creating docker_worker_2 ... done
surbhi@surbhi-Lenovo-G50-80:~/Downloads/docker$
```

```
surbhi@surbhi-Lenovo-G50-80: ~/Downloads/docker
worker_2  | 18/11/30 14:27:50 INFO handler.ContextHandler: Started o.s.j.s.Servl
etContextHandler@17fc9a28{/logPage/json,null,AVAILABLE,@Spark}
worker_2  | 18/11/30 14:27:50 INFO handler.ContextHandler: Started o.s.j.s.Servl
etContextHandler@5ec9f23{/,null,AVAILABLE,@Spark}
worker_2  | 18/11/30 14:27:50 INFO handler.ContextHandler: Started o.s.j.s.Servl
etContextHandler@5606d932{/json,null,AVAILABLE,@Spark}
worker_2  | 18/11/30 14:27:50 INFO handler.ContextHandler: Started o.s.j.s.Servl
etContextHandler@526ff0b6{/static,null,AVAILABLE,@Spark}
worker_2  | 18/11/30 14:27:50 INFO handler.ContextHandler: Started o.s.j.s.Servl
etContextHandler@593812fc{/log,null,AVAILABLE,@Spark}
worker_2  | 18/11/30 14:27:50 INFO ui.WorkerWebUI: Bound WorkerWebUI to 0.0.0.0,
 and started at http://localhost:8081
worker_2  | 18/11/30 14:27:50 INFO worker.Worker: Connecting to master master:70
77...
worker_2  | 18/11/30 14:27:50 INFO handler.ContextHandler: Started o.s.j.s.Servl
etContextHandler@7323585a{/metrics/json,null,AVAILABLE,@Spark}
worker_2  | 18/11/30 14:27:51 INFO client.TransportClientFactory: Successfully c
reated connection to master/172.17.0.2:7077 after 50 ms (0 ms spent in bootstrap
s)
master_1  | 18/11/30 14:27:51 INFO master.Master: Registering worker 172.17.0.4:
8881 with 2 cores, 3.0 GB RAM
worker_2  | 18/11/30 14:27:51 INFO worker.Worker: Successfully registered with m
aster spark://master:7077
```

Enter docker-spark container

```
$ sudo docker exec -it <name-of-the-master-container> bash
```

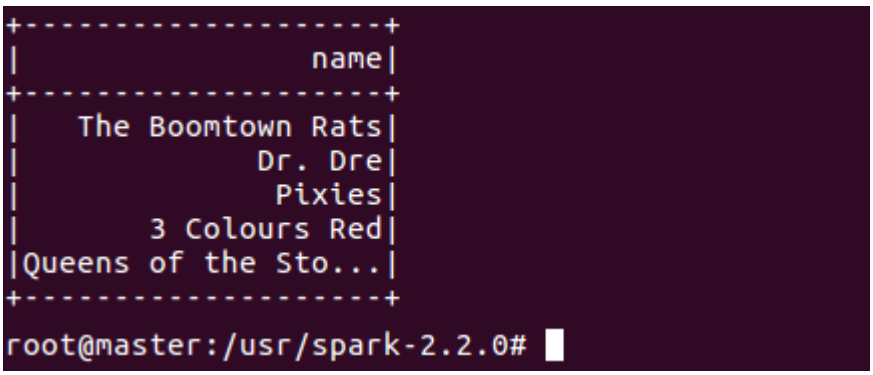
Execute `spark-submit` locally inside master docker container

```
<path-of-spark-directory>/bin/spark-submit --class
com.cloudera.datascience.recommender.RunRecommender /tmp/data/recommenderbigdata_2.11-
0.1.jar 1000002
```

Execute `spark-submit` in **client mode** inside the master docker container

```
<path-of-spark-directory>/bin/spark-submit --class
com.cloudera.datascience.recommender.RunRecommender --deploy-mode client
/tmp/data/recommenderbigdata_2.11-0.1.jar 1000002
```

**Output**




Execute `spark-submit` in **cluster mode** inside the master docker container

```
<path-of-spark-directory>/bin/spark-submit --class
com.cloudera.datascience.recommender.RunRecommender --supervise --master
spark://master:6066 --deploy-mode cluster /tmp/data/recommenderbigdata_2.11-0.1.jar
1000002
```

**Output**

```
surbhi@surbhi-Lenovo-G50-80: ~/Downloads/docker
root@master:/usr/spark-2.2.0# ./bin/spark-submit --class com.cloudera.datascienc
e.recommender.RunRecommender --supervise --deploy-mode cluster --master spark://
master:6066 /tmp/data/recommenderbigdata_2.11-0.1.jar 1000002
Running Spark using the REST application submission protocol.
18/11/30 16:06:22 INFO rest.RestSubmissionClient: Submitting a request to launch
 an application in spark://master:6066.
18/11/30 16:06:23 INFO rest.RestSubmissionClient: Submission successfully create
d as driver-20181130160622-0001. Polling submission state...
18/11/30 16:06:23 INFO rest.RestSubmissionClient: Submitting a request for the s
tatus of submission driver-20181130160622-0001 in spark://master:6066.
18/11/30 16:06:23 INFO rest.RestSubmissionClient: State of driver driver-2018113
0160622-0001 is now RUNNING.
18/11/30 16:06:23 INFO rest.RestSubmissionClient: Driver is running on worker wo
rker-20181130154921-172.17.0.4-8881 at 172.17.0.4:8881.
18/11/30 16:06:23 INFO rest.RestSubmissionClient: Server responded with CreateSu
bmissionResponse:
{
  "action" : "CreateSubmissionResponse",
  "message" : "Driver successfully submitted as driver-20181130160622-0001",
  "serverSparkVersion" : "2.2.0",
  "submissionId" : "driver-20181130160622-0001",
  "success" : true
}
root@master:/usr/spark-2.2.0#
```

## Spark Master at spark://master:7077

**URL:** spark://master:7077
**REST URL:** spark://master:6066 *(cluster mode)*
**Alive Workers:** 2
**Cores in use:** 4 Total, 0 Used
**Memory in use:** 6.0 GB Total, 0.0 B Used
**Applications:** 0 Running, 2 Completed
**Drivers:** 0 Running, 1 Completed
**Status:** ALIVE

### Workers

| Worker Id | Address | State | Cores | Memory |
|---|---|---|---|---|
| worker-20181130154740-172.17.0.3-8881 | 172.17.0.3:8881 | ALIVE | 2 (0 Used) | 3.0 GB (0.0 B Used) |
| worker-20181130154921-172.17.0.4-8881 | 172.17.0.4:8881 | ALIVE | 2 (0 Used) | 3.0 GB (0.0 B Used) |

### Running Applications

| Application ID | Name | Cores | Memory per Executor | Submitted Time | User | State | Duration |
|---|---|---|---|---|---|---|---|

### Running Drivers

| Submission ID | Submitted Time | Worker | State | Cores | Memory | Main Class |
|---|---|---|---|---|---|---|

### Completed Applications

| Application ID | Name | Cores | Memory per Executor | Submitted Time | User | State | Duration |
|---|---|---|---|---|---|---|---|
| app-20181130155557-0001 | com.cloudera.datascience.recommender.RunRecommender | 4 | 1024.0 MB | 2018/11/30 15:55:57 | root | FINISHED | 4.2 min |


## Store data in HDFS

Download docker-compose.yml file from github repository [6]. Modify as below.

```
version: '2'
services:
  namenode:
    image: bde2020/hadoop-namenode:1.1.0-hadoop2.8-java8
    container_name: namenode
    volumes:
      - ./data/namenode:/hadoop/dfs/name
    environment:
      - CLUSTER_NAME=test
    env_file:
      - ./hadoop.env
...
  datanode:
    image: bde2020/hadoop-datanode:1.1.0-hadoop2.8-java8
    depends_on:
      - namenode
    volumes:
      - ./data/datanode:/hadoop/dfs/data
    env_file:
      - ./hadoop.env
...
  spark-master:
    image: bde2020/spark-master:2.1.0-hadoop2.8-hive-java8
    container_name: spark-master
...
    volumes:
      - ./conf/master:/conf
    env_file:
      - ./hadoop.env
  spark-worker:
```

```
    image: bde2020/spark-worker:2.1.0-hadoop2.8-hive-java8
    depends_on:
      - spark-master
...
    volumes:
    - ./conf/worker:/conf
    env_file:
      - ./hadoop.env
  hue:
    image: bde2020/hdfs-filebrowser:3.11
...
```

Download hadoop.env file from github repository [6]. The first line of the document tells the location and port number exposed for hdfs.

```
CORE_CONF_fs_defaultFS=hdfs://namenode:8020
...
```

**Modify the path of the dataset with the path of the data volume mounted on the HDFS**

```
spark.sparkContext.setCheckpointDir("hdfs://namenode:8020/user/root/data/")
val base = "hdfs://namenode:8020/user/root/data/"
```

Create a new folder called datanode inside the data folder (created previously). Copy the new jar and dataset files inside the newly created datanode folder.

Additionally, also copy the new jar inside the `<path-of-the-directory>/conf/master` folder (this folder is also created previously and it should be present in the in the same path as the data folder).

Execute below docker commands to start docker containers and scale up to two workers instead of one.

```
$ sudo docker-compose up
$ sudo docker-compose scale spark-worker=2
```

Enter datanode docker container

```
$ sudo docker exec -it docker_datanode_1 bash
```

Execute below commands inside datanode docker container

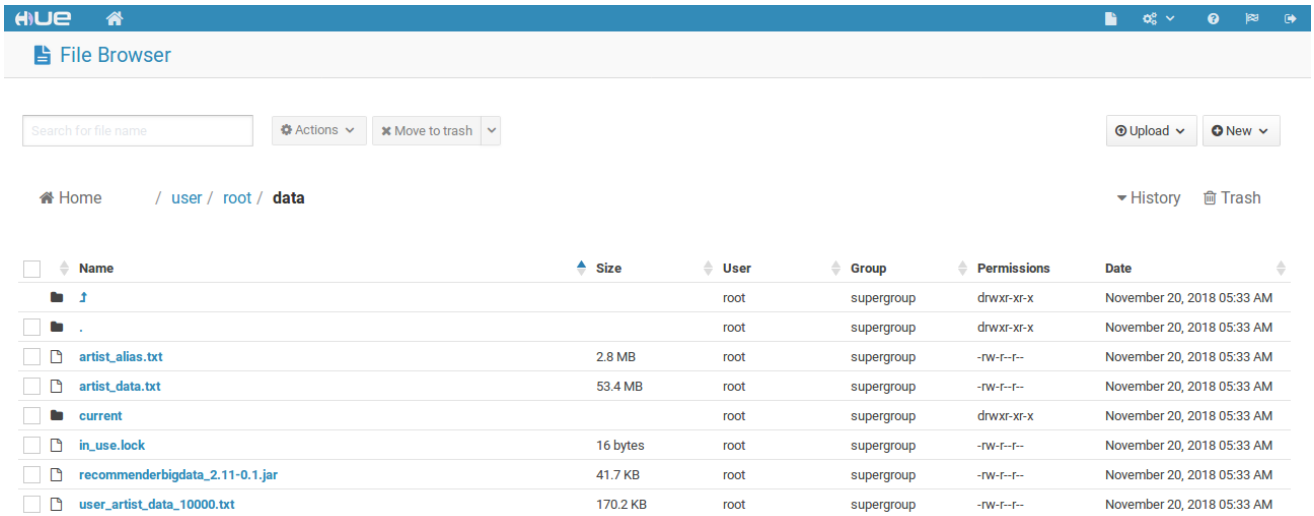
```
$ hadoop fs -mkdir -p /user/root
$ hadoop fs -put /hadoop/dfs/data /user/root/data
```

| Hadoop Options | Explanation |
|---|---|
| hadoop fs | hadoop fs instead of hdfs dfs because hadoop fs can point to any file systems like local, HDFS etc. unlike hdfs dfs. |
| -mkdir -p | `-mkdir` is to create directory if does not exist, and `-p` helps to overwrite if already exists. |
| -put | copy from container path (/user/root/data) to hdfs path (/hadoop/dfs/data) |

Path to see hdfs files in Hue file browser: do `ifconfig` from terminal and use the **docker0 ip address**

```
http://172.17.0.1:8088/filebrowser
```

**Output**




Enter master docker container

```
$ sudo docker exec -it spark-master bash
```

Execute `spark-submit` in **client mode** inside the master docker container

```
<path-of-spark-directory>/bin/spark-submit --class
com.cloudera.datascience.recommender.RunRecommender /conf/recommenderbigdata_2.11-
0.1.jar 1000002
```

**Output**

```
18/11/30 16:45:38 INFO datasources.FileScanRDD: Reading File path: hdfs://namenode:8020/user/root/data/artist_data.txt, range: 45118407-5596357
5, partition values: [empty row]
18/11/30 16:45:39 INFO executor.Executor: Finished task 2.0 in stage 6198.0 (TID 28593). 1881 bytes result sent to driver
18/11/30 16:45:39 INFO scheduler.TaskSetManager: Finished task 2.0 in stage 6198.0 (TID 28593) in 688 ms on localhost (executor driver) (1/3)
18/11/30 16:45:39 INFO executor.Executor: Finished task 0.0 in stage 6198.0 (TID 28591). 1956 bytes result sent to driver
18/11/30 16:45:39 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 6198.0 (TID 28591) in 801 ms on localhost (executor driver) (2/3)
18/11/30 16:45:39 INFO executor.Executor: Finished task 1.0 in stage 6198.0 (TID 28592). 1912 bytes result sent to driver
18/11/30 16:45:39 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 6198.0 (TID 28592) in 803 ms on localhost (executor driver) (3/3)
18/11/30 16:45:39 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 6198.0, whose tasks have all completed, from pool
18/11/30 16:45:39 INFO scheduler.DAGScheduler: ResultStage 6198 (show at RunRecommender.scala:184) finished in 0.803 s
18/11/30 16:45:39 INFO scheduler.DAGScheduler: Job 457 finished: show at RunRecommender.scala:184, took 0.806106 s
18/11/30 16:45:39 INFO codegen.CodeGenerator: Code generated in 3.715593 ms
+-------------+
|         name|
+-------------+
|         Kiss|
|  Lemon Jelly|
|    The Clash|
|Alkaline Trio|
|  Duran Duran|
+-------------+
```

Execute `spark-submit` in **cluster mode** inside the master docker container

```
<path-of-spark-directory>/bin/spark-submit --class
com.cloudera.datascience.recommender.RunRecommender --master spark://spark-master:6066
--supervise --deploy-mode cluster
hdfs://namenode:8020/user/root/data/recommenderbigdata_2.11-0.1.jar 1000002
```

### Output

Spark 2.1.2-SNAPSHOT **stdout log page for driver-20181130170731-0001**

Back to Master

Showing 5533 Bytes: 0 - 5533 of 5533

```
+-------+-------------------+
|    719|     Ella Fitzgerald|
|     59|      Massive Attack|
|    976|             Nirvana|
|1000569|Queens of the Sto...|
|1001172|          60ft Dolls|
+-------+-------------------+

0.48997221309669337
(0.8316764726294403,(30,1.0,1.0))
(0.8115280728551643,(30,1.0,40.0))
(0.7996759651383106,(5,1.0E-4,1.0))
(0.7952232861714673,(5,1.0,40.0))
(0.7934593450247976,(5,1.0,1.0))
(0.7901095411817244,(5,1.0E-4,40.0))
(0.6647281007278888,(30,1.0E-4,40.0))
(0.6494350439016447,(30,1.0E-4,1.0))
+-----------------+
|             name|
+-----------------+
|              DMX|
|      The Specials|
|     Guns N' Roses|
|Jefferson Airplane|
|         Warren G|
+-----------------+
```

Load New

## Acknowledgement

The author would like to express heartfelt gratitude for the extensive support and valuable time provided by Andrés Muñoz Arcentales and Sonsoles López Pernas (PhD students, UPM) in achieving the results of the project.

## References

[1] https://storage.googleapis.com/aas-data-sets/profiledata_06-May-2005.tar.gz, "Audioscrobbler dataset"

[2] https://github.com/sryza/aas/blob/master/ch03-recommender/src/main/scala/com/cloudera/datascience/recommender/RunRecommender.scala, "Recommender Code"

[3] https://github.com/P7h/docker-spark, "Docker github"

[4] https://docs.docker.com/engine/reference/commandline/run/#parent-command, "Docker documentation"

[5] https://github.com/gettyimages/docker-spark, "Docker-compose github"

[6] https://github.com/big-data-europe/docker-hadoop-spark-workbench/blob/master/, "Docker-compose and hadoop github"

[7] https://github.com/sryza/aas.git, "Advanced Analytics with Spark"

[8] https://github.com/sryza/aas/tree/master/ch03-recommender, "Chapter 3: Recommender"