Spark with Scala Project

Data Analysis using Spark with Scala transformation Functions:

1. Put the dataset (baby_names.csv) in the spark-bin-hadoop2.7 folder.

2. Store the dataset values in the local RDD.

```
val babyNames = sc.textFile("baby_names.csv")
```

3. Split the data into rows using comma(,) as delimiter and remove the first row which contains headings. Store the result in a new dataset.

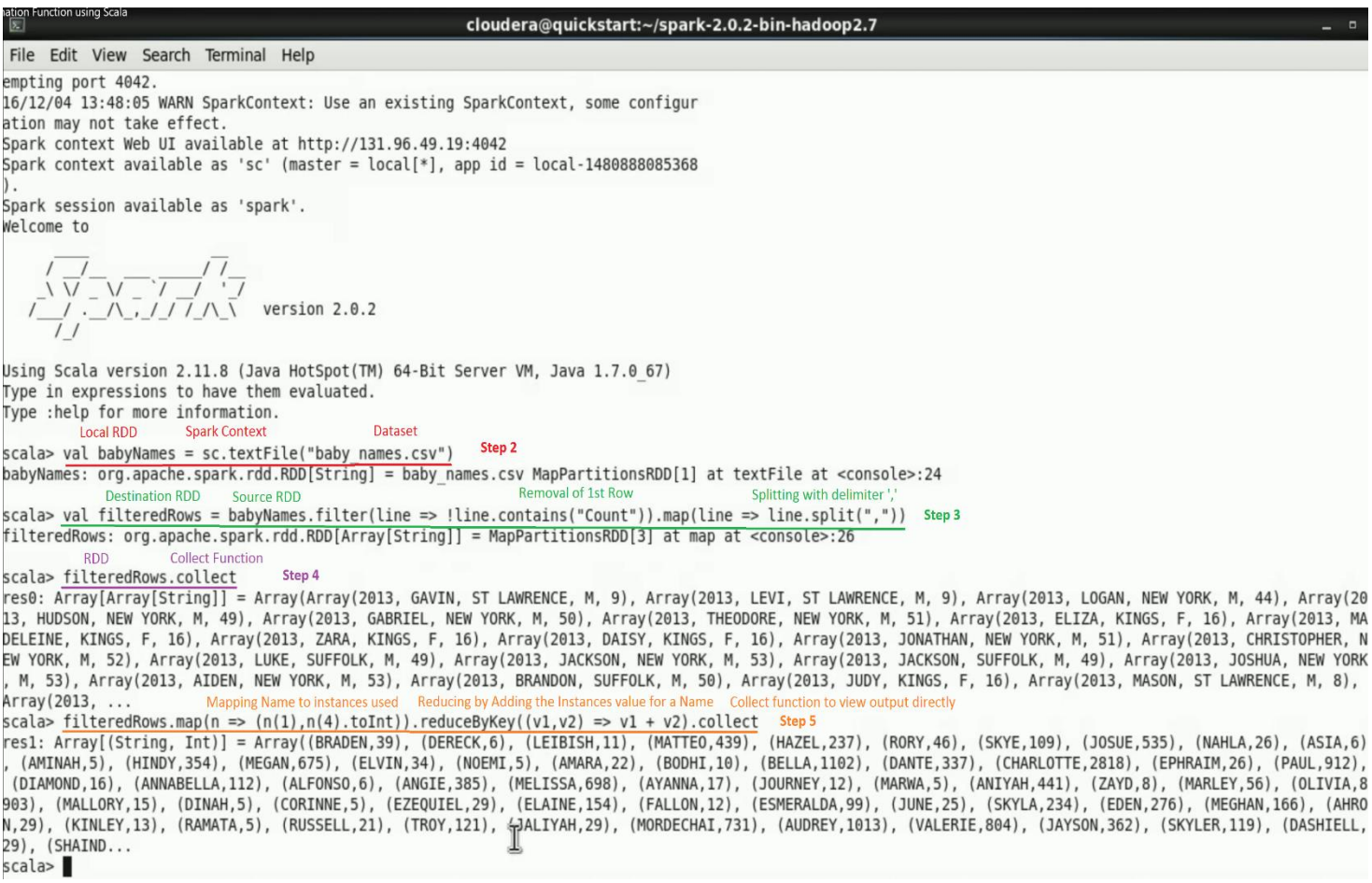
```
val filteredRows = babyNames.filter(line => !line.contains("Count")).map(line => line.split(","))
```

4. We can see a part of the RDD using the 'collect' function.

```
filteredRows.collect
```

5. We will next map the 1st value of each array (i.e. Names) with the 4th value of each array (i.e. Instances of name use). We will reduce the mapped data by adding the 'instances' value, keeping the 'Names' value as reference.

```
filteredRows.map(n => (n(1),n(4).toInt)).reduceByKey((v1,v2) => v1 + v2).collect
```



Spark Standalone Cluster Deployment with Scala Program:

1. Prerequisites : Installation of Spark, Scala, Java and SBT.

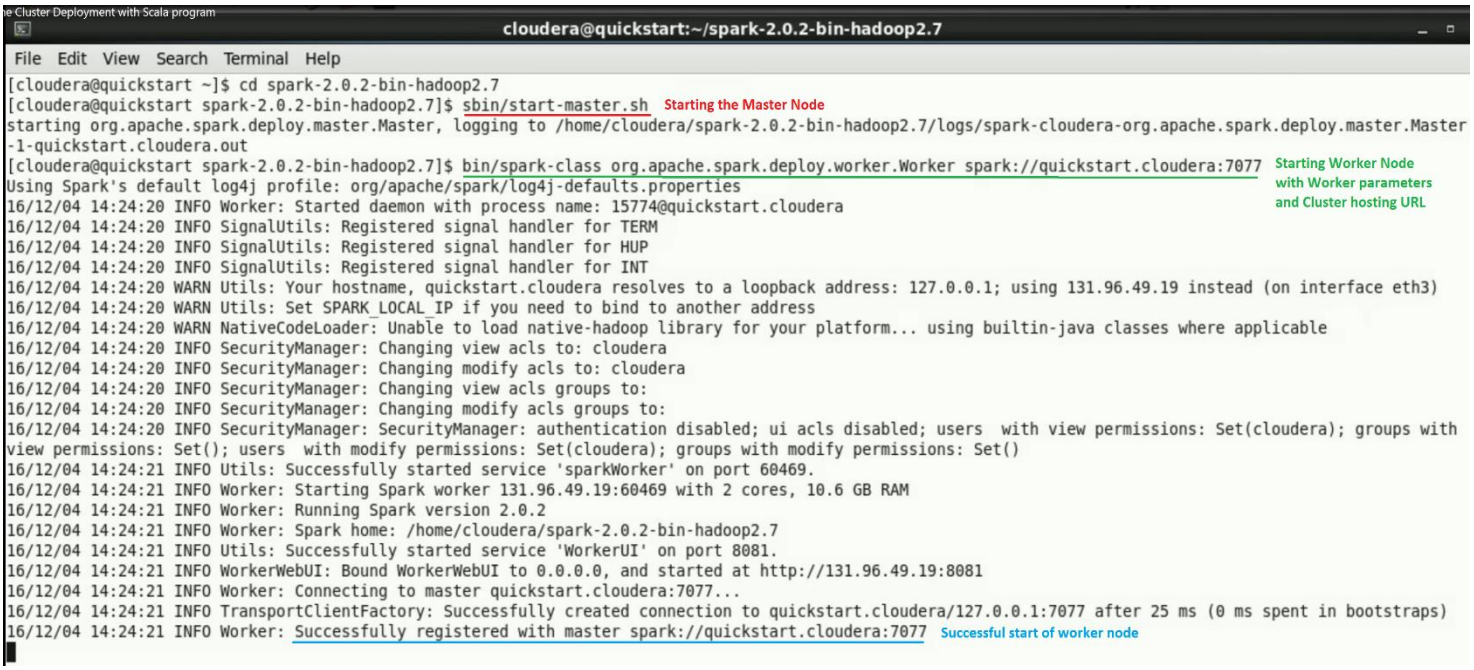
2. Copy the 'spark-cluster-master' to the folder 'spark-2.0.2-bin-hadoop2.7'.

3. To start the architecture, start the Master Node. Go to spark-2.0.2-bin-hadoop2.7. Then start the 'start-master' shell script.

```
sbi/start-master.sh
```

4. Then we will start the Worker Node with the 'spark-class' script with worker parameters and the URL of the Cluster Host.

```
bin/spark-class org.apache.spark.deploy.worker.Worker spark://quickstart.cloudera:7077
```



5. After viewing the Success message, go to Web Browser and launch the IP Address:localhost:8080 to launch the Spark UI. The UI will give info like Number of Workers, Memory in Use, Applications Data, Worker Address,etc.

6. Compile the Bash Profile.

```
. ~/.bash_profile
```

7. Compile the program that is available in the path 'spark-cluster-master/spark-cluster/src/main/scala/SparkPi.scala' by using sbt to make a jar file that we will use while implementing the program on the cluster.

```
cd spark-cluster

sbt package
```

8. The deployment of program is done through 'spark-submit' script with certain parameters like Program we are running, URL of Cluster being hosted and path of the jar file.

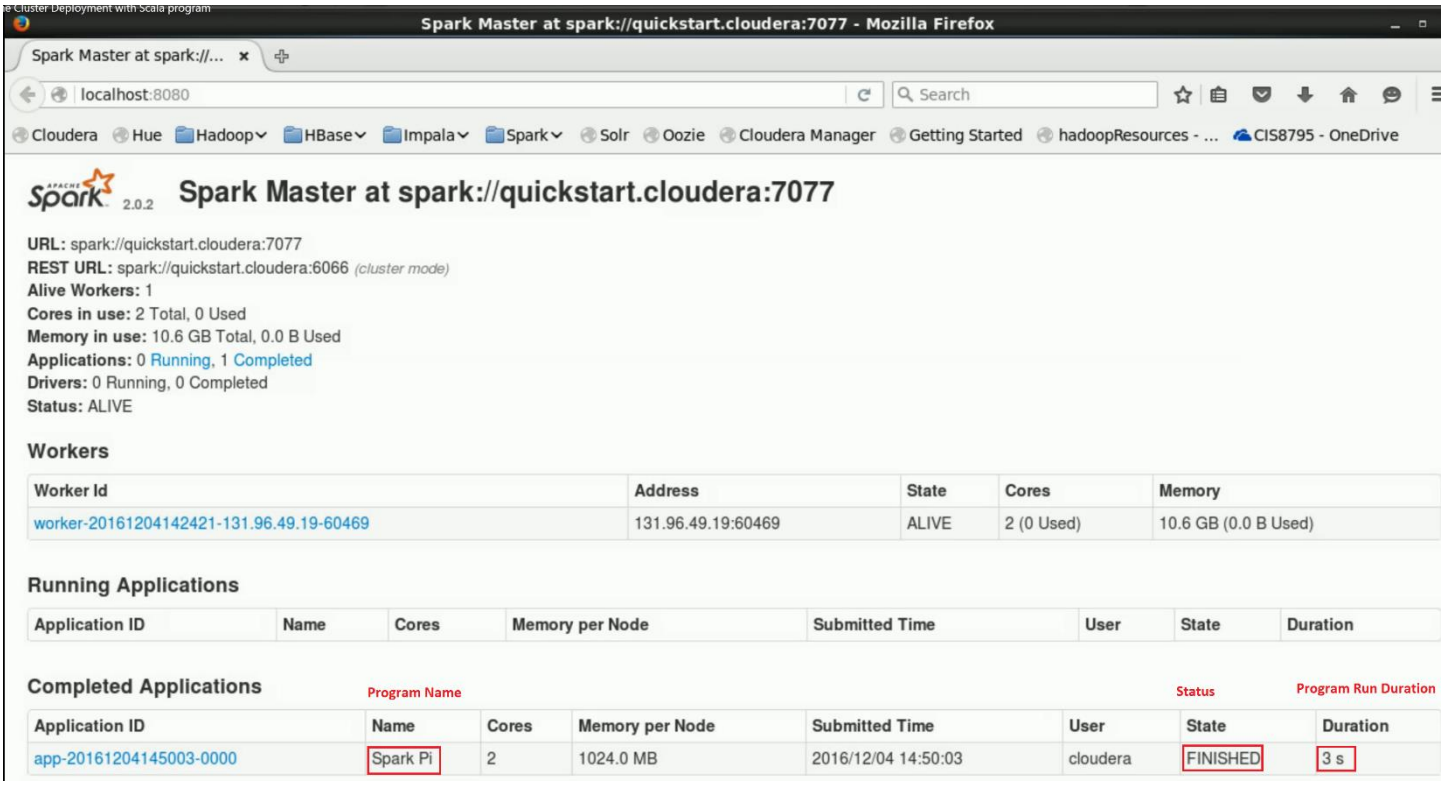
~/spark-2.0.2-bin-hadoop2.7/bin/spark-submit --class "SparkPi" --master spark://quickstart.cloudera:7077 ./target/scala-2.10/spark-sample_2.10-1.0.jar

```
cloudera@quickstart:~/spark-2.0.2-bin-hadoop2.7/spark-course-master/spark-cluster
File  Edit  View  Search  Terminal  Help
[cloudera@quickstart ~]$ cd spark-2.0.2-bin-hadoop2.7
[cloudera@quickstart spark-2.0.2-bin-hadoop2.7]$ ls
baby_names.csv  conf  examples  LICENSE   logs       NOTICE  R          RELEASE  spark-course-master  yarn
bin             data  jars      licenses  names.csv  python  README.md  sbin     work
[cloudera@quickstart spark-2.0.2-bin-hadoop2.7]$ cd spark-course-master/
[cloudera@quickstart spark-course-master]$ ls
baby_names.csv  baby_names.json  baby_names_reduced.csv  README.md  spark-cluster  spark-ml  spark-sql  spark-streaming  spark-streaming-tests
[cloudera@quickstart spark-course-master]$ cd spark-cluster/
[cloudera@quickstart spark-cluster]$ . ~/.bash_profile   Compilation of Bash Profile for successful execution of SBT
[cloudera@quickstart spark-cluster]$ sbt package   SBT compilation to create JAR file of the program about to be deployed
[info] Loading project definition from /home/cloudera/spark-2.0.2-bin-hadoop2.7/spark-course-master/spark-cluster/project
[info] Set current project to Spark Sample (in build file:/home/cloudera/spark-2.0.2-bin-hadoop2.7/spark-course-master/spark-cluster/)
[success] Total time: 0 s, completed Dec 4, 2016 2:40:54 PM   JAR file created successfully
[cloudera@quickstart spark-cluster]$ ~/spark-2.0.2-bin-hadoop2.7/bin/spark-submit --class "SparkPi" --master spark://quickstart.cloudera:7077 ./target/scala-
2.10/spark-sample_2.10-1.0.jar   Program Deployment in Standalone Cluster
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
16/12/04 14:50:01 INFO SparkContext: Running Spark version 2.0.2
16/12/04 14:50:02 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
16/12/04 14:50:02 WARN Utils: Your hostname, quickstart.cloudera resolves to a loopback address: 127.0.0.1; using 131.96.49.19 instead (on interface eth3)
16/12/04 14:50:02 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
16/12/04 14:50:02 INFO SecurityManager: Changing view acls to: cloudera
16/12/04 14:50:02 INFO SecurityManager: Changing modify acls to: cloudera
16/12/04 14:50:02 INFO SecurityManager: Changing view acls groups to:
16/12/04 14:50:02 INFO SecurityManager: Changing modify acls groups to:
16/12/04 14:50:02 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view permissions: Set(cloudera); groups with
view permissions: Set(); users  with modify permissions: Set(cloudera); groups with modify permissions: Set()
```

9. The program SparkPi program uses the Monte-Carlo alogorithm to calculate the approximate value of PI by running 200,000 simulations of the algorithm. After program implementation we can see output among the display like : "Pi is roughly 3.14048".

```
16/12/04 14:50:05 INFO TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 734 ms on 131.96.49.19 (2/2)
16/12/04 14:50:05 INFO DAGScheduler: ResultStage 0 (reduce at SparkPi.scala:16) finished in 2.054 s
16/12/04 14:50:05 INFO TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
16/12/04 14:50:05 INFO DAGScheduler: Job 0 finished: reduce at SparkPi.scala:16, took 2.313812 s
Pi is roughly 3.14048   Output Displayed Accurately after running many simulations
16/12/04 14:50:05 INFO SparkUI: Stopped Spark web UI at http://131.96.49.19:4040
16/12/04 14:50:06 INFO StandaloneSchedulerBackend: Shutting down all executors
16/12/04 14:50:06 INFO CoarseGrainedSchedulerBackend$DriverEndpoint: Asking each executor to shut down
16/12/04 14:50:06 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
16/12/04 14:50:06 INFO MemoryStore: MemoryStore cleared
16/12/04 14:50:06 INFO BlockManager: BlockManager stopped
16/12/04 14:50:06 INFO BlockManagerMaster: BlockManagerMaster stopped
16/12/04 14:50:06 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
16/12/04 14:50:06 INFO SparkContext: Successfully stopped SparkContext
16/12/04 14:50:06 INFO ShutdownHookManager: Shutdown hook called
16/12/04 14:50:06 INFO ShutdownHookManager: Deleting directory /tmp/spark-eacdf38f-4662-4bcb-9591-6c97d18c7e11
[cloudera@quickstart spark-cluster]$
```

10. On going back to the Spark Cluster UI, we can see under the "Completed Applications" section that the "SparkPi" application has been successfully completed and the time duration it took for completion.



11. We can stop the Worker Node by hitting "Ctrl+c" in the terminal running the "spark-class" script.

12. Master node can be stopped by running the 'stop-master.sh'.

```
sbin/stop-master.sh
```

```
16/12/04 14:50:06 INFO Worker: Asked to kill executor app-20161204145003-0000/0   On pressing (Ctrl + C) in the worker node terminal, worker node is killed
16/12/04 14:50:06 INFO ExecutorRunner: Runner thread for executor app-20161204145003-0000/0 interrupted
16/12/04 14:50:06 INFO ExecutorRunner: Killing process!
16/12/04 14:50:06 INFO Worker: Executor app-20161204145003-0000/0 finished with state KILLED exitStatus 0
16/12/04 14:50:06 INFO Worker: Cleaning up local directories for application app-20161204145003-0000
16/12/04 14:50:06 INFO ExternalShuffleBlockResolver: Application app-20161204145003-0000 removed, cleanupLocalDirs = true
^C16/12/04 14:56:31 ERROR Worker: RECEIVED SIGNAL INT
16/12/04 14:56:31 INFO ShutdownHookManager: Shutdown hook called
16/12/04 14:56:31 INFO ShutdownHookManager: Deleting directory /tmp/spark-4caaa0ba-27aa-4cf9-8e67-b55761eb2c23
[cloudera@quickstart spark-2.0.2-bin-hadoop2.7]$ sbin/stop-master.sh   Calling the shell script which stops Mater Node
stopping org.apache.spark.deploy.master.Master   Master Node stopped successfully
[cloudera@quickstart spark-2.0.2-bin-hadoop2.7]$
```

Refer the video "Spark Standalone Cluster Deployment with Scala program" (https://youtu.be/nx_v721rc9A) for more clarity.