

```

File Edit View Search Terminal Help
empting port 4042.
16/12/04 13:48:05 WARN SparkContext: Use an existing SparkContext, some configur
ation may not take effect.
Spark context Web UI available at http://131.96.49.19:4042
Spark context available as 'sc' (master = local[*], app id = local-1480888085368
).
Spark session available as 'spark'.
Welcome to

  version 2.0.2

Using Scala version 2.11.8 (Java HotSpot(TM) 64-Bit Server VM, Java 1.7.0_67)
Type in expressions to have them evaluated.
Type :help for more information.

Local RDD      Spark Context      Dataset
scala> val babyNames = sc.textFile("baby_names.csv")      Step 2
babyNames: org.apache.spark.rdd.RDD[String] = baby_names.csv MapPartitionsRDD[1] at textFile at <console>:24

Destination RDD      Source RDD      Removal of 1st Row      Splitting with delimiter ','
scala> val filteredRows = babyNames.filter(line => !line.contains("Count")).map(line => line.split(","))      Step 3
filteredRows: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[3] at map at <console>:26

RDD      Collect Function
scala> filteredRows.collect      Step 4
res0: Array[Array[String]] = Array(Array(2013, GAVIN, ST LAWRENCE, M, 9), Array(2013, LEVI, ST LAWRENCE, M, 9), Array(2013, LOGAN, NEW YORK, M, 44), Array(20
13, HUDSON, NEW YORK, M, 49), Array(2013, GABRIEL, NEW YORK, M, 50), Array(2013, THEODORE, NEW YORK, M, 51), Array(2013, ELIZA, KINGS, F, 16), Array(2013, MA
DELEINE, KINGS, F, 16), Array(2013, ZARA, KINGS, F, 16), Array(2013, DAISY, KINGS, F, 16), Array(2013, JONATHAN, NEW YORK, M, 51), Array(2013, CHRISTOPHER, N
EW YORK, M, 52), Array(2013, LUKE, SUFFOLK, M, 49), Array(2013, JACKSON, NEW YORK, M, 53), Array(2013, JACKSON, SUFFOLK, M, 49), Array(2013, JOSHUA, NEW YORK
, M, 53), Array(2013, AIDEN, NEW YORK, M, 53), Array(2013, BRANDON, SUFFOLK, M, 50), Array(2013, JUDY, KINGS, F, 16), Array(2013, MASON, ST LAWRENCE, M, 8),
Array(2013, ...
Mapping Name to instances used      Reducing by Adding the Instances value for a Name      Collect function to view output directly
scala> filteredRows.map(n => (n(1),n(4).toInt)).reduceByKey((v1,v2) => v1 + v2).collect      Step 5
res1: Array[(String, Int)] = Array((BRADEN,39), (DERECK,6), (LEIBISH,11), (MATTEO,439), (HAZEL,237), (RORY,46), (SKYE,109), (JOSUE,535), (NAHLA,26), (ASIA,6)
, (AMINAH,5), (HINDY,354), (MEGAN,675), (ELVIN,34), (NOEMI,5), (AMARA,22), (BODHI,10), (BELLA,1102), (DANTE,337), (CHARLOTTE,2818), (EPHRAIM,26), (PAUL,912),
(DIAMOND,16), (ANNABELLA,112), (ALFONSO,6), (ANGIE,385), (MELISSA,698), (AYANNA,17), (JOURNEY,12), (MARWA,5), (ANIYAH,441), (ZAYD,8), (MARLEY,56), (OLIVIA,8
903), (MALLORY,15), (DINAH,5), (CORINNE,5), (EZEQUIEL,29), (ELAINE,154), (FALLON,12), (ESMERALDA,99), (JUNE,25), (SKYLA,234), (EDEN,276), (MEGHAN,166), (AHRO
N,29), (KINLEY,13), (RAMATA,5), (RUSSELL,21), (TROY,121), (JALIJAH,29), (MORDECHAI,731), (AUDREY,1013), (VALERIE,804), (JAYSON,362), (SKYLER,119), (DASHIELL,
29), (SHAIND...
scala>

```

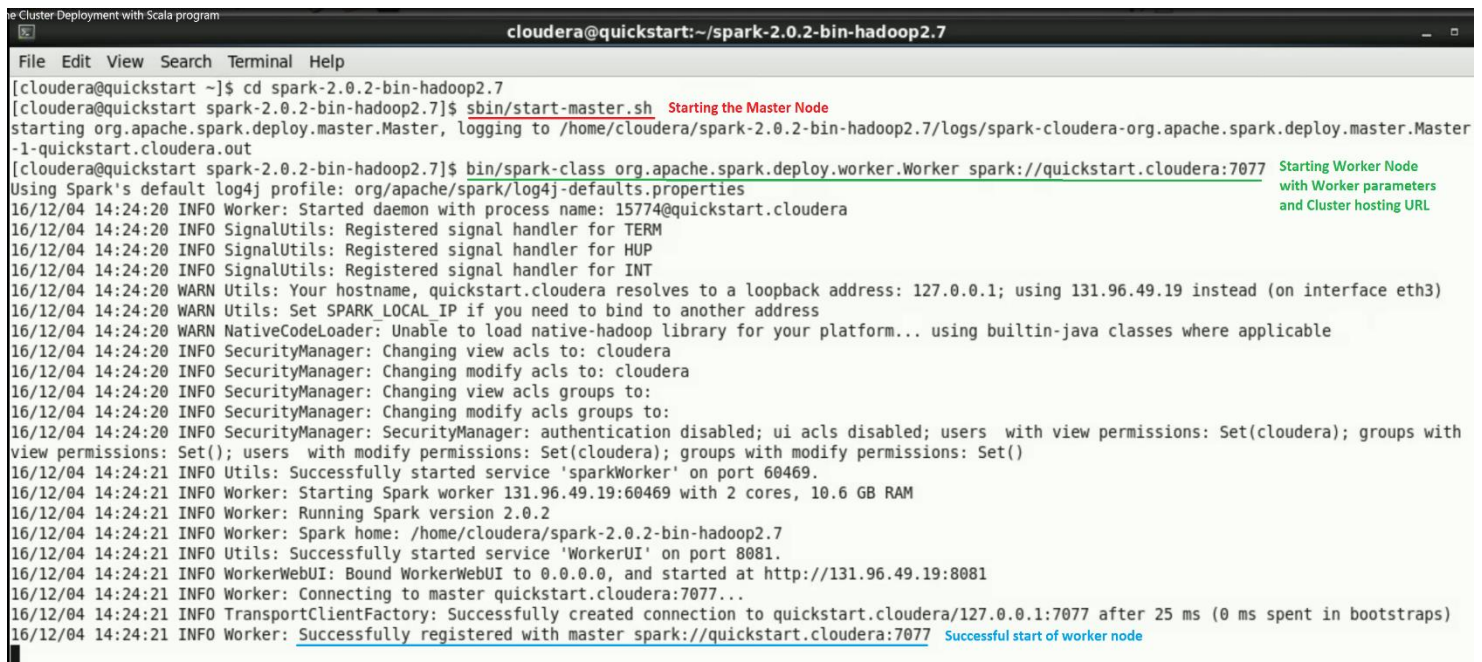
Spark Standalone Cluster Deployment with Scala Program:

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
```



The screenshot shows a terminal window titled "cloudera@quickstart:~/spark-2.0.2-bin-hadoop2.7". The terminal displays the following commands and output:

```
[cloudera@quickstart ~]$ cd spark-2.0.2-bin-hadoop2.7
[cloudera@quickstart spark-2.0.2-bin-hadoop2.7]$ sbi/start-master.sh Starting the Master Node
starting org.apache.spark.deploy.master.Master, logging to /home/cloudera/spark-2.0.2-bin-hadoop2.7/logs/spark-cloudera-org.apache.spark.deploy.master.Master-1-quickstart.cloudera.out
[cloudera@quickstart spark-2.0.2-bin-hadoop2.7]$ bin/spark-class org.apache.spark.deploy.worker.Worker spark://quickstart.cloudera:7077 Starting Worker Node with Worker parameters and Cluster hosting URL
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
16/12/04 14:24:20 INFO Worker: Started daemon with process name: 15774@quickstart.cloudera
16/12/04 14:24:20 INFO SignalUtils: Registered signal handler for TERM
16/12/04 14:24:20 INFO SignalUtils: Registered signal handler for HUP
16/12/04 14:24:20 INFO SignalUtils: Registered signal handler for INT
16/12/04 14:24:20 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 14:24:20 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
16/12/04 14:24:20 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
16/12/04 14:24:20 INFO SecurityManager: Changing view acls to: cloudera
16/12/04 14:24:20 INFO SecurityManager: Changing modify acls to: cloudera
16/12/04 14:24:20 INFO SecurityManager: Changing view acls groups to:
16/12/04 14:24:20 INFO SecurityManager: Changing modify acls groups to:
16/12/04 14:24:20 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()
16/12/04 14:24:21 INFO Utils: Successfully started service 'sparkWorker' on port 60469.
16/12/04 14:24:21 INFO Worker: Starting Spark worker 131.96.49.19:60469 with 2 cores, 10.6 GB RAM
16/12/04 14:24:21 INFO Worker: Running Spark version 2.0.2
16/12/04 14:24:21 INFO Worker: Spark home: /home/cloudera/spark-2.0.2-bin-hadoop2.7
16/12/04 14:24:21 INFO Utils: Successfully started service 'WorkerUI' on port 8081.
16/12/04 14:24:21 INFO WorkerWebUI: Bound WorkerWebUI to 0.0.0.0, and started at http://131.96.49.19:8081
16/12/04 14:24:21 INFO Worker: Connecting to master quickstart.cloudera:7077...
16/12/04 14:24:21 INFO TransportClientFactory: Successfully created connection to quickstart.cloudera/127.0.0.1:7077 after 25 ms (0 ms spent in bootstraps)
16/12/04 14:24:21 INFO Worker: Successfully registered with master spark://quickstart.cloudera:7077 Successful start of worker node
```

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
```

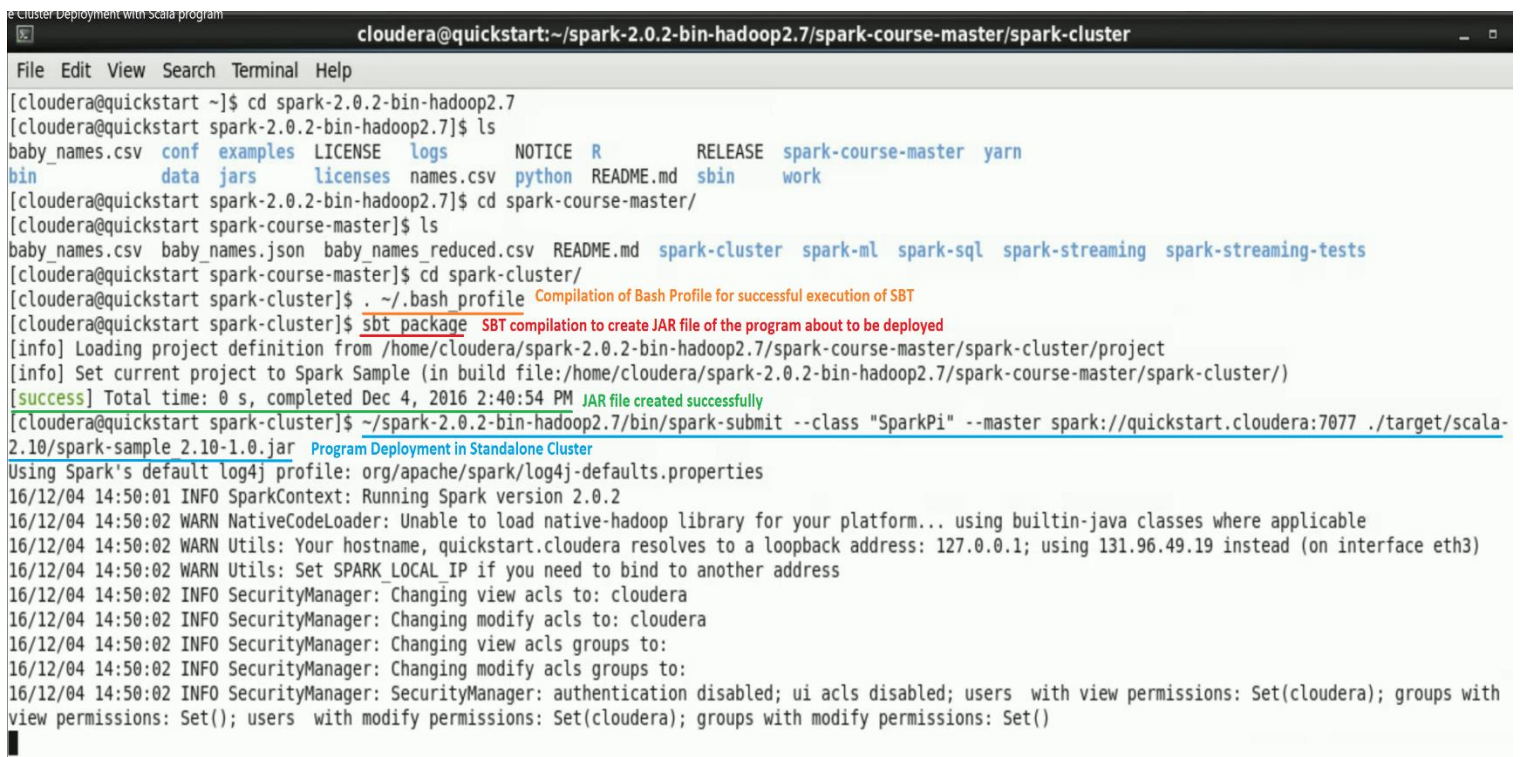
7. Compile the program that is available in the path 'spark-cluster-master/spark-cluster/src/main/scala/SparkPi.scala' by using sbt to make a jar file that we will use while implementing the program on the cluster.

```
cd spark-cluster
```

```
sbt package
```

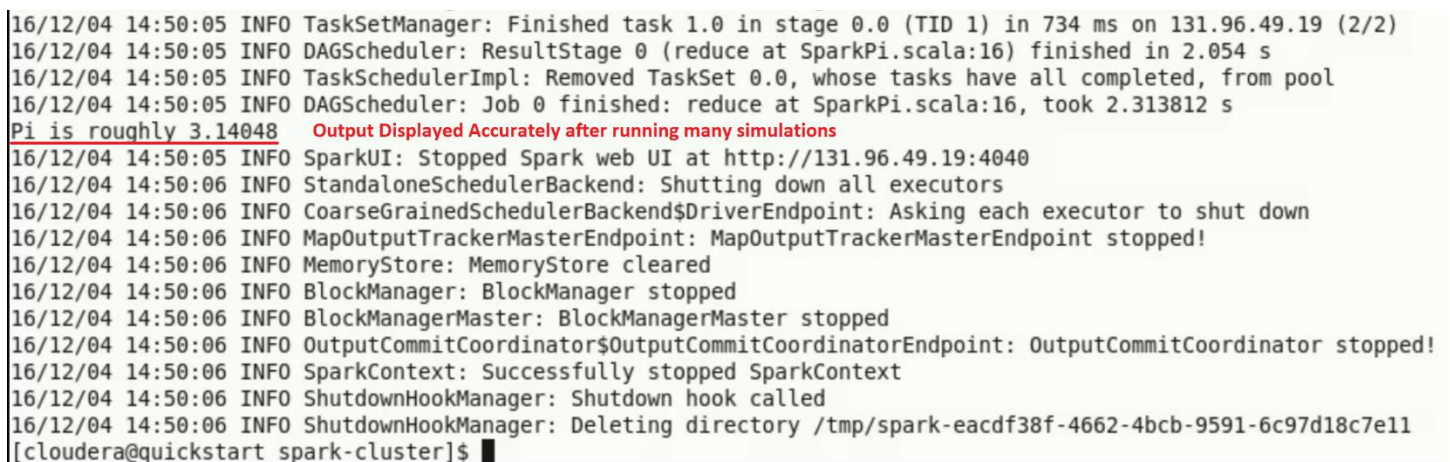
8. The deployment of program is done through 'spark-submit' script with certain parameters like Program we are running, URL of Cluster being hosted and path of the jar file.

```
~/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 algorithm 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.

Cluster Deployment with Scala program

Spark Master at spark://quickstart.cloudera:7077 - Mozilla Firefox

Spark Master at spark://... x

localhost:8080

Cloudera Hue Hadoop HBase Impala Spark Solr Oozie Cloudera Manager Getting Started hadoopResources - ... CIS8795 - OneDrive

Spark Master at spark://quickstart.cloudera:7077

URL: spark://quickstart.cloudera:7077
 REST URL: spark://quickstart.cloudera:6066 (cluster mode)
 Alive Workers: 1
 Cores in use: 2 Total, 0 Used
 Memory in use: 10.6 GB Total, 0.0 B Used
 Applications: 0 Running, 1 Completed
 Drivers: 0 Running, 0 Completed
 Status: ALIVE

Workers

Worker Id	Address	State	Cores	Memory
worker-20161204142421-131.96.49.19-60469	131.96.49.19:60469	ALIVE	2 (0 Used)	10.6 GB (0.0 B Used)

Running Applications

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

Completed Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
app-20161204145003-0000	Spark Pi	2	1024.0 MB	2016/12/04 14:50:03	cloudera	FINISHED	3 s

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.