SPARK STANALONE APPLICATION IN LOCAL MACHINE USING COMMAND LINE METHOD

---

1. Install Spark in local machine from http://spark.apache.org/downloads.html
   Useful commands:
   wget http://d3kbcqa49mib13.cloudfront.net/spark-1.1.0-bin-hadoop2.4.tgz
   gzip -d spark-1.1.0-bin-hadoop2.4.tgz
   tar xvf spark-1.1.0-bin-hadoop2.4.tar

2. Check the Installation by launching the spark shell

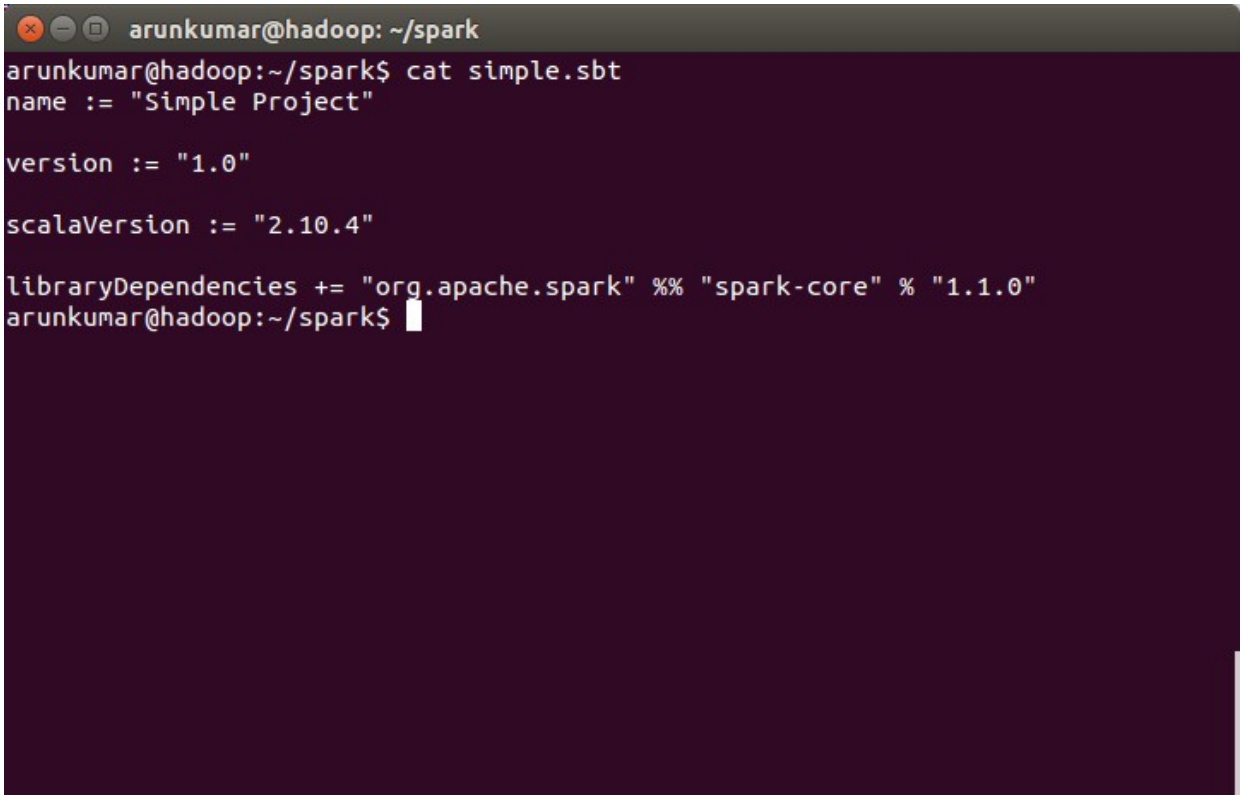
```
arunkumar@hadoop: ~/spark
arunkumar@hadoop:~$ cd spark
arunkumar@hadoop:~/spark$ bin/spark-shell
Spark assembly has been built with Hive, including Datanucleus jars on classpath
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 1.1.0
      /_/

Using Scala version 2.10.4 (OpenJDK 64-Bit Server VM, Java 1.7.0_65)
Type in expressions to have them evaluated.
Type :help for more information.
14/11/14 15:03:32 WARN Utils: Your hostname, hadoop resolves to a loopback addre
ss: 127.0.1.1; using 10.0.0.6 instead (on interface wlan0)
14/11/14 15:03:32 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
14/11/14 15:03:33 WARN NativeCodeLoader: Unable to load native-hadoop library fo
r your platform... using builtin-java classes where applicable
Spark context available as sc.

scala>
```

3. Install sbt-scala from http://www.scala-sbt.org/
   It is a interactive build tool to compile and create create jar from scala code
   Useful commands:
   wget https://dl.bintray.com/sbt/debian/sbt-0.13.6.deb
   sudo dpkg --install sbt-0.13.6.deb

4. Go to http://spark.apache.org/docs/latest/quick-start.html#standalone-applications and read about running standalone applications

5. create simple.sbt with the following content

```
name := "Simple Project"
version := "1.0"
scalaVersion := "2.10.4"
libraryDependencies += "org.apache.spark" %% "spark-core" % "1.1.0
```



6. Create a directory structure as follows:
   src/main/scala/
   Useful command : mkdir -p src/main/scala

   We have to write all the scala code in this directory.

```
arunkumar@hadoop: ~/spark
arunkumar@hadoop:~/spark$ tree src/
src/
└── main
    └── scala
        ├── SimpleApp.scala
        ├── WC Length.scala
        ├── WC Length.scala~
        ├── WC.scala
        └── WC.scala~

2 directories, 5 files
arunkumar@hadoop:~/spark$
```

7. Use the below template word count scala programs

```
arunkumar@hadoop: ~/spark
arunkumar@hadoop:~/spark$ cat src/main/scala/WC.scala
import org.apache.spark.SparkContext
import org.apache.spark.SparkContext._
import org.apache.spark.SparkConf

object WC {
  def main(args: Array[String]) {
    val logFile = "/home/arunkumar/SparkWorkSpace/CharCount.txt"
    val conf = new SparkConf().setAppName("Word Count")
    val sc = new SparkContext(conf)
    val logData = sc.textFile(logFile)
    val counts = logData.flatMap(line => line.split(" "))
                 .map(word => (word, 1))
                 .reduceByKey(_ + _)
counts.saveAsTextFile("/home/arunkumar/SparkWorkSpace/CharCount20")
  }
}
arunkumar@hadoop:~/spark$
```

8. Build package using 'sbt package' commad. At first time, it takes a while to build the package.

```
arunkumar@hadoop: ~/spark
arunkumar@hadoop:~/spark$ sbt package
[info] Set current project to Simple Project (in build file:/home/arunkumar/spar
k/)
[info] Compiling 1 Scala source to /home/arunkumar/spark/target/scala-2.10/class
es...
[info] Packaging /home/arunkumar/spark/target/scala-2.10/simple-project_2.10-1.0
.jar ...
[info] Done packaging.
[success] Total time: 9 s, completed Nov 14, 2014 3:31:29 PM
arunkumar@hadoop:~/spark$
```

```
arunkumar@hadoop: ~/spark
arunkumar@hadoop:~/spark$ tree target/scala-2.10
target/scala-2.10
├── classes
│   ├── CountLines.class
│   ├── CountLines$.class
│   ├── SimpleApp$$anonfun$1.class
│   ├── SimpleApp$$anonfun$2.class
│   ├── SimpleApp.class
│   ├── SimpleApp$.class
│   ├── WC$$anonfun$1.class
│   ├── WC$$anonfun$2.class
│   ├── WC$$anonfun$3.class
│   ├── WC.class
│   ├── WC$.class
│   ├── WL$$anonfun$1.class
│   ├── WL$$anonfun$2.class
│   ├── WL$$anonfun$3.class
│   ├── WL.class
│   └── WL$.class
└── simple-project_2.10-1.0.jar

1 directory, 17 files
arunkumar@hadoop:~/spark$
```

```
arunkumar@hadoop: ~/spark
arunkumar@hadoop:~/spark$ ls target/scala-2.10
classes  simple-project_2.10-1.0.jar
arunkumar@hadoop:~/spark$
```

9. Run the application in standalone mode by using the below command

bin/spark-submit --class "WC" --master local target/scala-2.10/simple-project_2.10-1.0.jar

```
arunkumar@hadoop: ~/spark
arunkumar@hadoop:~/spark$ bin/spark-submit  --class "WC" --master local  target/
scala-2.10/simple-project_2.10-1.0.jar
Spark assembly has been built with Hive, including Datanucleus jars on classpath
14/11/14 15:37:47 WARN Utils: Your hostname, hadoop resolves to a loopback addre
ss: 127.0.1.1; using 10.0.0.6 instead (on interface wlan0)
14/11/14 15:37:47 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
14/11/14 15:37:49 WARN NativeCodeLoader: Unable to load native-hadoop library fo
r your platform... using builtin-java classes where applicable
arunkumar@hadoop:~/spark$
```

10. Check the output in the location specified in the program

```
arunkumar@hadoop: ~/spark
arunkumar@hadoop:~/spark$ cat ~/SparkWorkSpace/CharCount20/part-00000
(Deer,2)
(Bear,2)
(Car,3)
(River,2)
arunkumar@hadoop:~/spark$
```