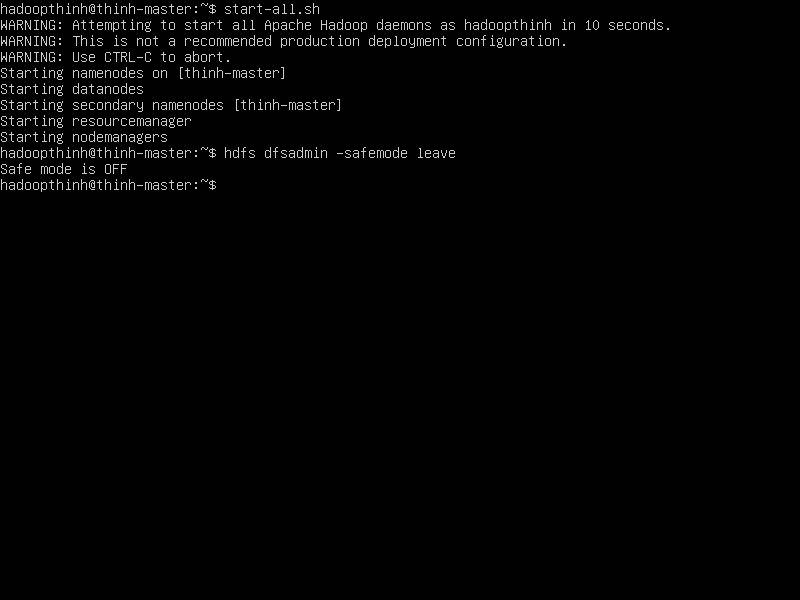
Đỗ Hoàng Thịnh – 20133122

**Spark Tutorial**

**1. Set up**

**1.1. Master**



**1.2. Slave**



**1.3. Spark**



Text

Description automatically generated

**2. Apache Spark – Core Programing**

**2.1. Create input file**

$vim intput.txt

$cat input.txt



**2.2. Run on Spark**

- put file on hdfs

$hdfs dfs -mkdir spark

$hdfs dfs -cp input.txt spark/

$spark-shell

scala>val inputfile = sc.textFile("./spark/input.txt")

scala>val counts = inputfile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(\_+\_);

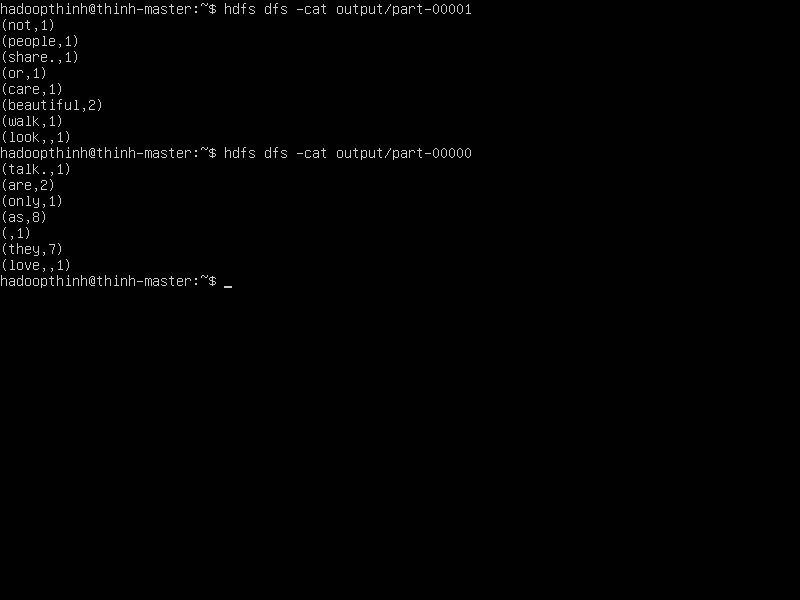
scala>counts.toDebugString

scala> counts.cache()

scala>counts.saveAsTextFile("output")

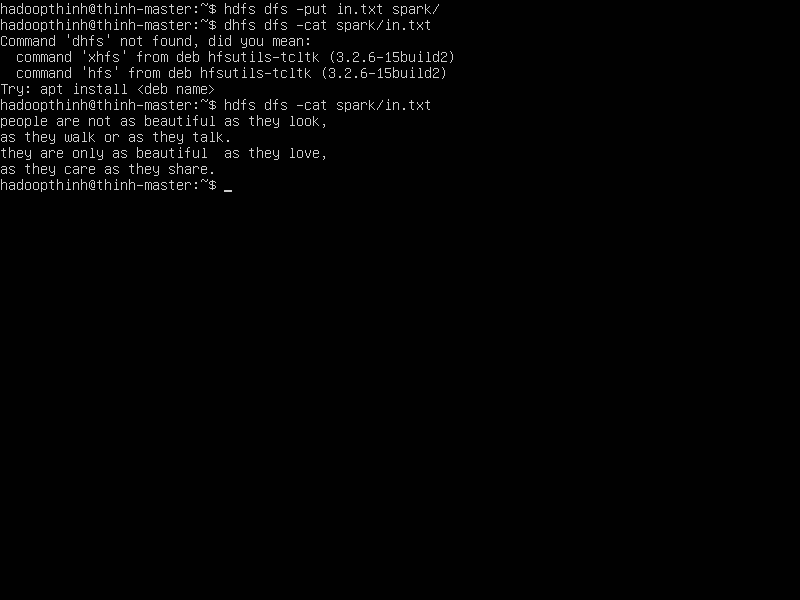


**2.3. OutPut**



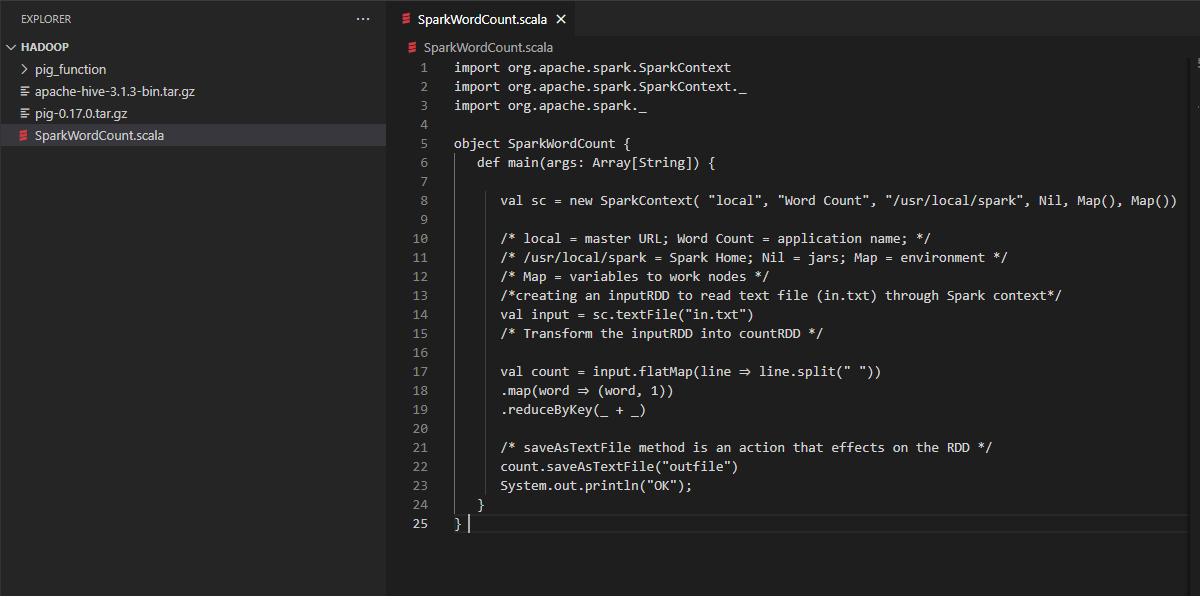
**2. Apache Spark – Deployment**

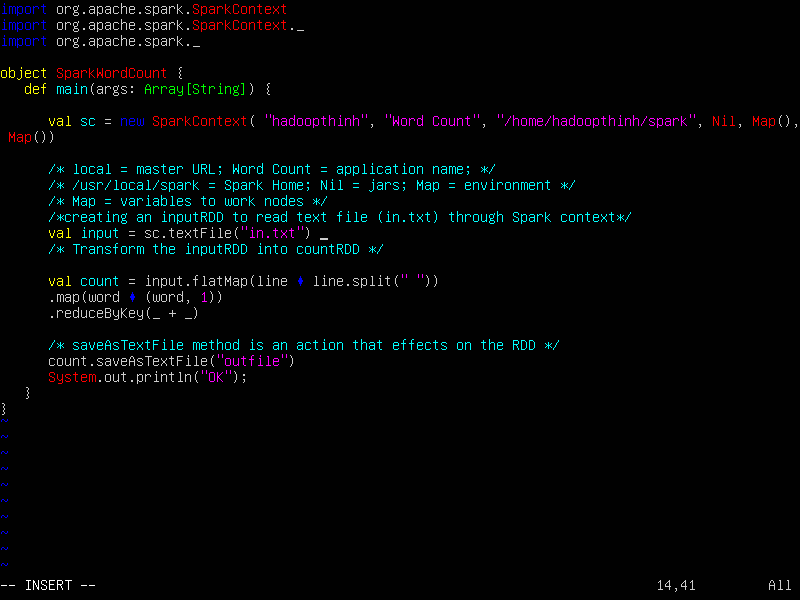
**2.1. Create file in.txt**

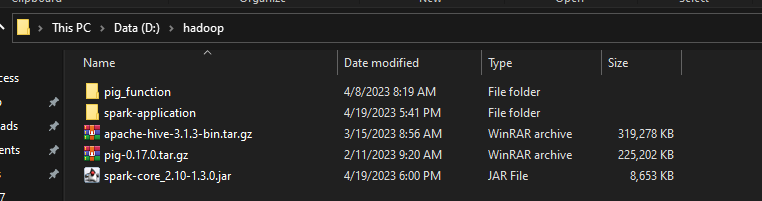


**2.2. SparkWordCount.scala & spark-core\_2.10-1.3.0.jar**

- Create and download



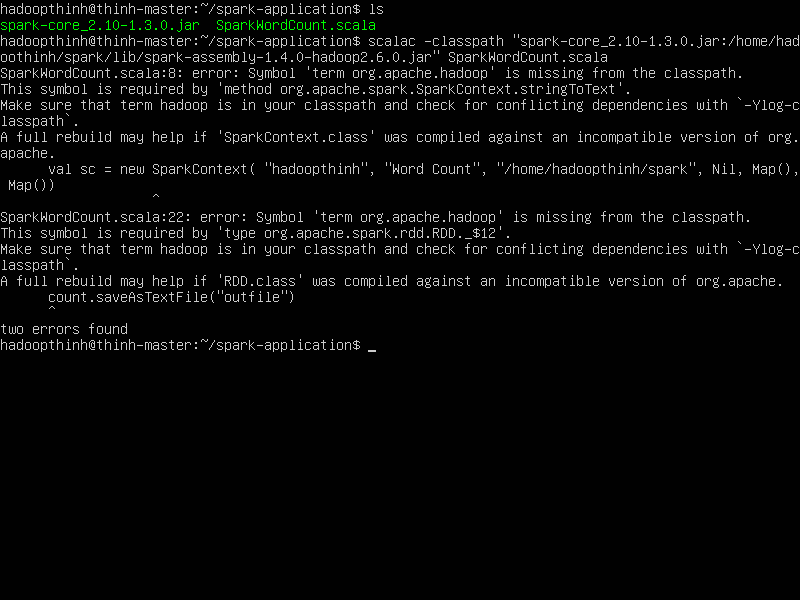




- Put in /home/hadoopthinh/spark-application



**2.3. Run**



**3.Advanced Spark Programing**

scala> val broadcastVar = sc.broadcast(Array(1, 2, 3))

scala> val accum = sc.longAccumulator("0")

scala> sc.parallelize(Array(1, 2, 3, 4)).foreach(x => accum.add(x))

scala> accum.value

