CASE STUDY 5

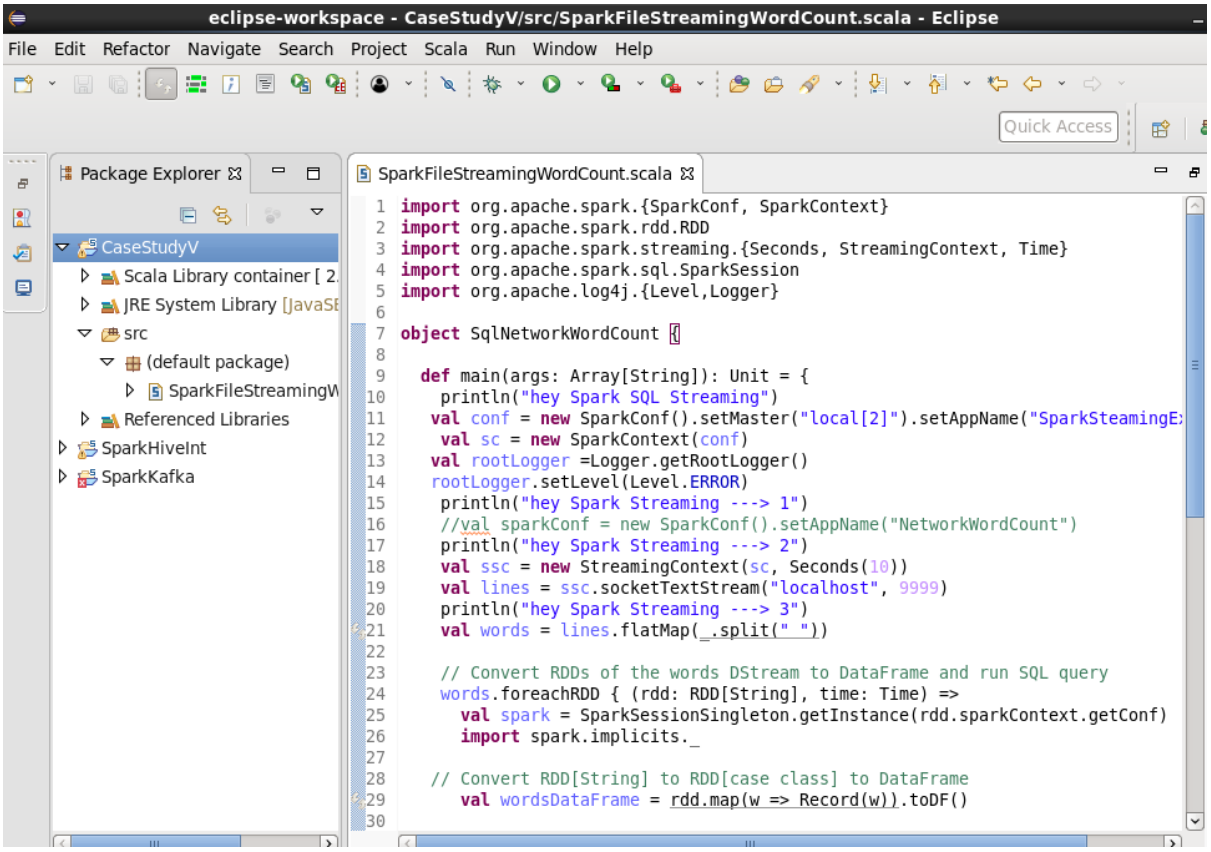
Task1: You have to create a Spark Application which streams data from a file on local directory on your machine and does the word count on the fly. The word should be done by the spark application in such a way that as soon as you drop the file in your local directory, your spark application should immediately do the word count for you.

Sol: Step1: open a scala project and create a scala application

➔ Program for scala application

```
import org.apache.spark.{SparkConf, SparkContext}
import org.apache.spark.rdd.RDD
import org.apache.spark.streaming.{Seconds, StreamingContext, Time}
import org.apache.spark.sql.SparkSession
import org.apache.log4j.{Level,Logger}

object SqlNetworkWordCount {

  def main(args: Array[String]): Unit = {
   println("hey Spark SQL Streaming")
  val conf = new SparkConf().setMaster("local[2]").setAppName("SparkSteamingExample")
   val sc = new SparkContext(conf)
  val rootLogger =Logger.getRootLogger()
  rootLogger.setLevel(Level.ERROR)
   println("hey Spark Streaming ---> 1")
   //val sparkConf = new SparkConf().setAppName("NetworkWordCount")
   println("hey Spark Streaming ---> 2")
   val ssc = new StreamingContext(sc, Seconds(10))
   val lines = ssc.socketTextStream("localhost", 9999)
   println("hey Spark Streaming ---> 3")
   val words = lines.flatMap(_.split(" "))

   // Convert RDDs of the words DStream to DataFrame and run SQL query
   words.foreachRDD { (rdd: RDD[String], time: Time) =>
     val spark = SparkSessionSingleton.getInstance(rdd.sparkContext.getConf)
     import spark.implicits._

  // Convert RDD[String] to RDD[case class] to DataFrame
     val wordsDataFrame = rdd.map(w => Record(w)).toDF()

   // Creates a temporary view using the DataFrame
     wordsDataFrame.createOrReplaceTempView("words")
  // Do word count on table using SQL and print it
     val wordCountsDataFrame =
       spark.sql("select word, count(*) as total from words group by word")
```

```
    println(s"========= $time =========")
    wordCountsDataFrame.show()
  }
  ssc.start()
  ssc.awaitTermination()
}
/** Case class for converting RDD to DataFrame */
case class Record(word: String)
/** Lazily instantiated singleton instance of SparkSession */
object SparkSessionSingleton {
@transient private var instance: SparkSession = _
  def getInstance(sparkConf: SparkConf): SparkSession = {
    if (instance == null) {
      instance = SparkSession
        .builder
        .config(sparkConf)
        .getOrCreate()
    }
    instance
  } } }
```



SparkFileStreamingWordCount.scala

```scala

    // Creates a temporary view using the DataFrame
      wordsDataFrame.createOrReplaceTempView("words")
   // Do word count on table using SQL and print it
      val wordCountsDataFrame =
        spark.sql("select word, count(*) as total from words group by word")
      println(s"========= $time =========")
      wordCountsDataFrame.show()
    }
    ssc.start()
    ssc.awaitTermination()
  }



 case class Record(word: String)
  /** Lazily instantiated singleton instance of SparkSession */
  object SparkSessionSingleton {
@transient private var instance: SparkSession = _
    def getInstance(sparkConf: SparkConf): SparkSession = {
      if (instance == null) {
        instance = SparkSession
          .builder
          .config(sparkConf)
          .getOrCreate()
      }
      instance
    }  }

}
```

Step2: Start nc server

➔ nc –lk 9999

```
hs_err_pid4625.log  test.txt
install             Videos
jarKafkaUtils
[acadgild@localhost ~]$ nc -lk 9999
hey bigdata
bigdata word word
hey hey tech tech
acadgild
acadgild acadgild
^C
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$
```

Step3: Run the program

eclipse-workspace - CaseStudyV/src/SparkFileStreamingWordCount.scala - Eclipse

File Edit Refactor Navigate Search Project Scala Run Window Help

Quick Access

Problems Tasks Console

```
<terminated> SqlNetworkWordCount$ [Scala Application] /usr/java/jdk1.8.0_151/bin/java (May 28, 2018, 8
18/05/28 08:56:40 INFO BlockManagerMasterEndpoint: Registering block manager 192.168.
18/05/28 08:56:40 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(dri
18/05/28 08:56:40 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver,
hey Spark Streaming ---> 1
hey Spark Streaming ---> 2
hey Spark Streaming ---> 3
========= 1527478010000 ms =========
+-------+-----+
|   word|total|
+-------+-----+
|   word|    2|
|bigdata|    2|
|    hey|    3|
|   tech|    2|
+-------+-----+

========= 1527478020000 ms =========
+----+-----+
|word|total|
```

```
m words group by word")

Session */

on = {
```

eclipse-workspace - CaseStudyV/src/SparkFileStreamingWordCount.scala - Eclipse

File Edit Refactor Navigate Search Project Scala Run Window Help

Quick Acc

Problems Tasks Console

```
<terminated> SqlNetworkWordCount$ [Scala Application] /usr/java/jdk1.8.0_151/bin/java (May 28, 2018, 8
|word|total|
+----+-----+
+----+-----+

========= 1527478070000 ms =========
+----+-----+
|word|total|
+----+-----+
+----+-----+

========= 1527478080000 ms =========
+--------+-----+
|    word|total|
+--------+-----+
|acadgild|    3|
+--------+-----+

========= 1527478090000 ms =========
+----+-----+
```

```
m words group b

Session */

on = {
```

Task2: In this part, you will have to create a Spark Application which should do the following

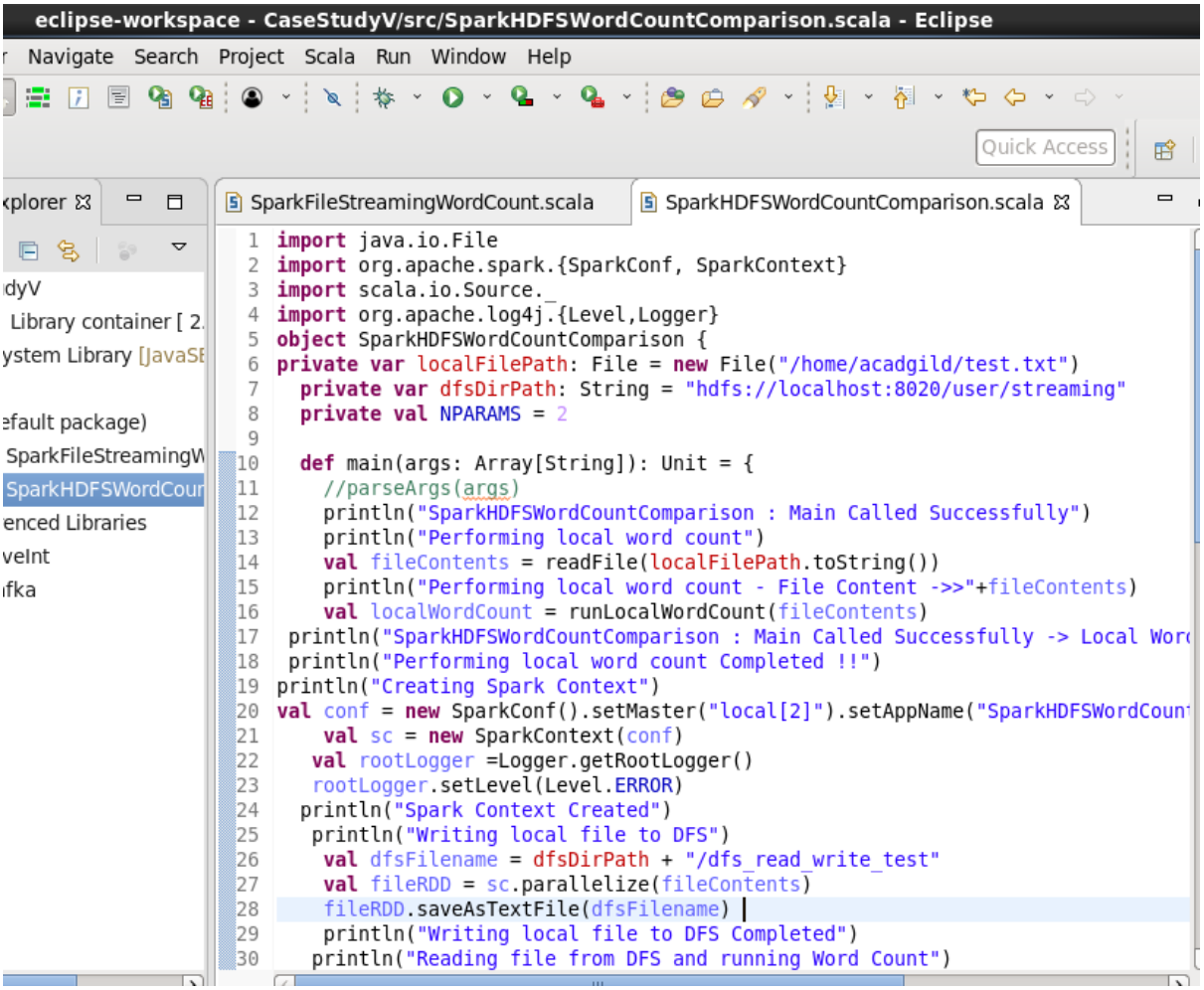
1. Pick up a file from the local directory and do the word count

2. Then in the same Spark Application, write the code to put the same file on HDFS.

3. Then in same Spark Application, do the word count of the file copied on HDFS in step 2

4. Lastly, compare the word count of step 1 and 2. Both should match, other throw an error

Sol: Step1: create a scala application in the previous scala project



Places System Mon May 28, 9:20 AM

eclipse-workspace - CaseStudyV/src/SparkHDFSWordCountComparison.scala - Eclipse

ctor Navigate Search Project Scala Run Window Help

Quick Access

e Explorer

StudyV
ala Library container [2.
E System Library [JavaSE
c
(default package)
SparkFileStreamingW
SparkHDFSWordCour
ferenced Libraries
HiveInt
Kafka

SparkFileStreamingWordCount.scala | SparkHDFSWordCountComparison.scala

```
42
43      if (localWordCount == dfsWordCount) {
44        println(s"Success! Local Word Count ($localWordCount) " +
45          s"and DFS Word Count ($dfsWordCount) agree.")
46      } else {
47        println(s"Failure! Local Word Count ($localWordCount) " +s"and DFS Word (
48       }
49  }
50    private def printUsage(): Unit = {
51      val usage: String = "DFS Read-Write Test\n" +
52        "\n" +"Usage: localFile dfsDir\n" + "\n" +
53        "localFile - (string) local file to use in test\n" +
54        "dfsDir - (string) DFS directory for read/write tests\n"
55      println(usage)
56    }
57   private def readFile(filename: String): List[String] = {
58      val lineIter: Iterator[String] = fromFile(filename).getLines()
59      val lineList: List[String] = lineIter.toList
60      lineList
61    }
62   def runLocalWordCount(fileContents: List[String]): Int = {
63      fileContents.flatMap(_.split(" "))
64        .flatMap(_.split("\t"))
65        .filter(_.nonEmpty)
66        .groupBy(w => w)
67        .mapValues(_.size)
68        .values
69        .sum
70    }
71 }
```

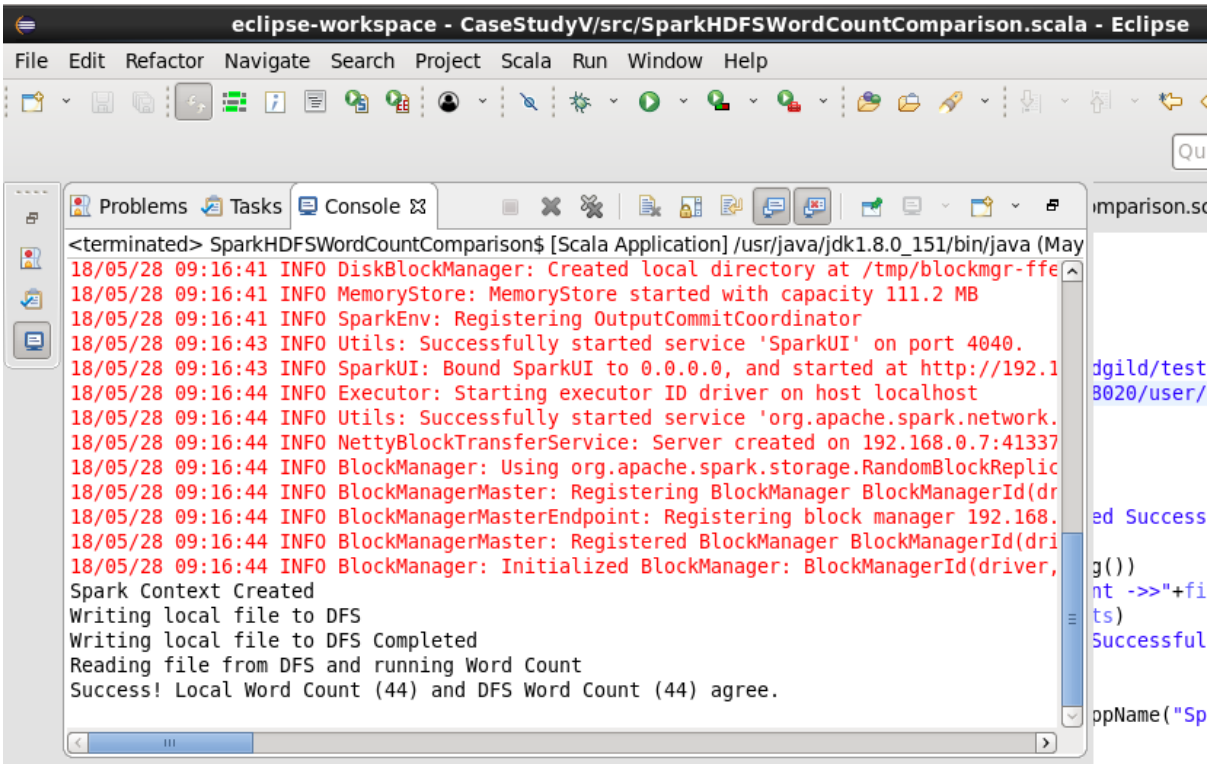
Step2: Run the program

File Edit Refactor Navigate Search Project Scala Run Window Help

Quick Ac

Problems Tasks Console

```
<terminated> SparkHDFSWordCountComparison$ [Scala Application] /usr/java/jdk1.8.0_151/bin/java (May
SparkHDFSWordCountComparison : Main Called Successfully
Performing local word count
Performing local word count - File Content ->>List( case class Record(word: String) ,
SparkHDFSWordCountComparison : Main Called Successfully -> Local Word Count is ->>44
Performing local word count Completed !!
Creating Spark Context
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-1.2.6/lib/slf4j-
SLF4J: Found binding in [jar:file:/home/acadgild/install/spark/spark-2.2.1-bin-hadoop
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
18/05/28 09:16:35 INFO SparkContext: Running Spark version 2.2.1
18/05/28 09:16:37 WARN NativeCodeLoader: Unable to load native-hadoop library for you
18/05/28 09:16:38 WARN Utils: Your hostname, localhost.localdomain resolves to a loop
18/05/28 09:16:38 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another addre
18/05/28 09:16:38 INFO SparkContext: Submitted application: SparkHDFSWordCountComp ar
18/05/28 09:16:38 INFO SecurityManager: Changing view acls to: acadgild
18/05/28 09:16:38 INFO SecurityManager: Changing modify acls to: acadgild
```


eclipse-workspace - CaseStudyV/src/SparkHDFSWordCountComparison.scala - Eclipse
File Edit Refactor Navigate Search Project Scala Run Window Help
Problems Tasks Console
<terminated> SparkHDFSWordCountComparison$ [Scala Application] /usr/java/jdk1.8.0_151/bin/java (May
18/05/28 09:16:41 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-ffe
18/05/28 09:16:41 INFO MemoryStore: MemoryStore started with capacity 111.2 MB
18/05/28 09:16:41 INFO SparkEnv: Registering OutputCommitCoordinator
18/05/28 09:16:43 INFO Utils: Successfully started service 'SparkUI' on port 4040.
18/05/28 09:16:43 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://192.1
18/05/28 09:16:44 INFO Executor: Starting executor ID driver on host localhost
18/05/28 09:16:44 INFO Utils: Successfully started service 'org.apache.spark.network.
18/05/28 09:16:44 INFO NettyBlockTransferService: Server created on 192.168.0.7:41337
18/05/28 09:16:44 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplic
18/05/28 09:16:44 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(dr
18/05/28 09:16:44 INFO BlockManagerMasterEndpoint: Registering block manager 192.168.
18/05/28 09:16:44 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(dri
18/05/28 09:16:44 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver,
Spark Context Created
Writing local file to DFS
Writing local file to DFS Completed
Reading file from DFS and running Word Count
Success! Local Word Count (44) and DFS Word Count (44) agree.