Instruction Manual for the Application Md Arafat Hossain Khan

arafat.math@gmail.com

December, 2020

Contents

Chapter 1

Instruction to run and test the application on AWS

Step 1: Data Collection

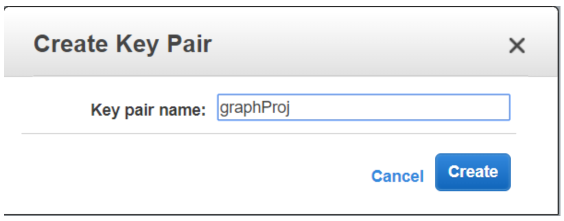
Go to https://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236. Select 'Filter Year' and 'Filter Period' from dropdown menu. Then from the 'Field Name' column, select only three columns ORIGIN, DEST and REPORTING_AIRLINE. Make sure that all other fields are unchecked. Click on 'Download' button. A zipped folder will be downloaded. Extract the csv file from this zipped folder.

Step 2: Log in to AWS

Go to https://aws.amazon.com/, log in to your account and go to 'AWS Management Console '.

Step 3: Create Key Pair

From EC2 Dashboard, create key pair.


Create Key Pair

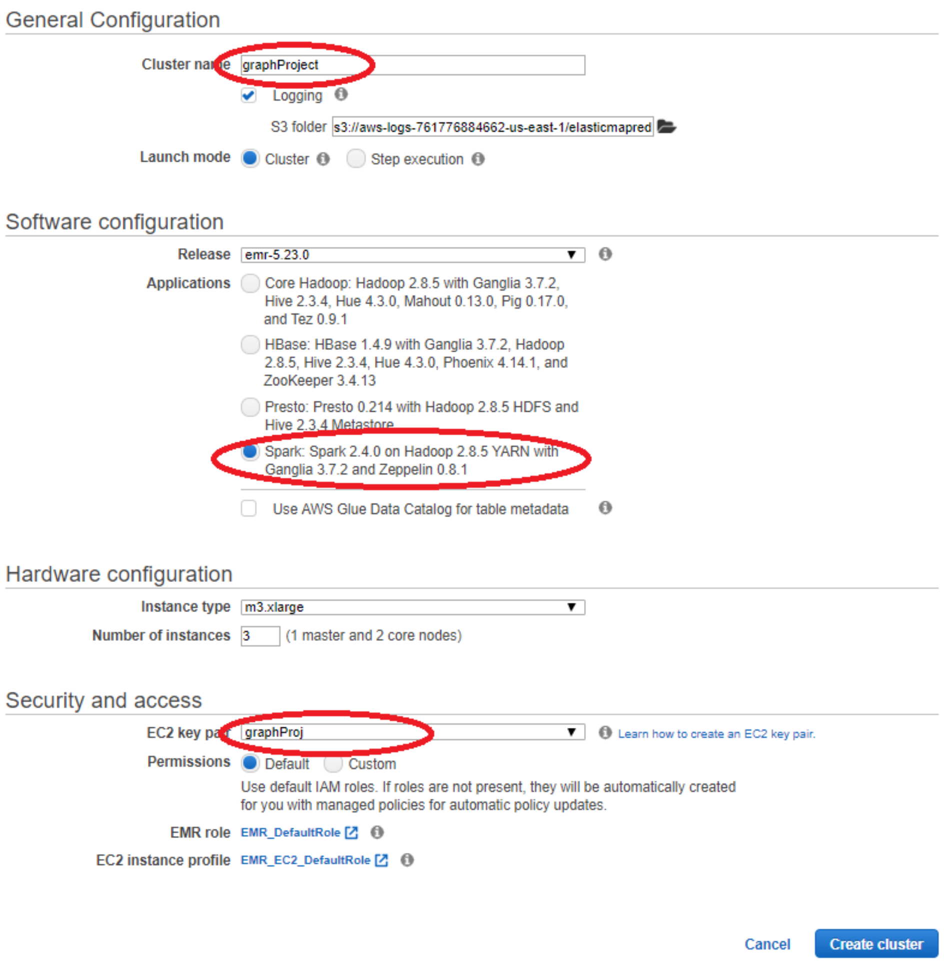
Key pair name: graphProj

Figure 1.1: Create Key Pair.

Save the 'graphProj.pem' file for later use if needed. But in this testing we will not use it locally.

Step 4: Create Cluster

Now follow the following setup to create a cluster from 'Amazon EMR'.



General Configuration

Cluster name: graphProject
[✓] Logging
S3 folder: s3://aws-logs-761776884662-us-east-1/elasticmapred
Launch mode: [✓ Cluster] Step execution

Software configuration

Release: emr-5.23.0
Applications:
- Core Hadoop: Hadoop 2.8.5 with Ganglia 3.7.2, Hive 2.3.4, Hue 4.3.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.1
- HBase: HBase 1.4.9 with Ganglia 3.7.2, Hadoop 2.8.5, Hive 2.3.4, Hue 4.3.0, Phoenix 4.14.1, and ZooKeeper 3.4.13
- Presto: Presto 0.214 with Hadoop 2.8.5 HDFS and Hive 2.3.4 Metastore
- [✓ Spark: Spark 2.4.0 on Hadoop 2.8.5 YARN with Ganglia 3.7.2 and Zeppelin 0.8.1]

Use AWS Glue Data Catalog for table metadata

Hardware configuration

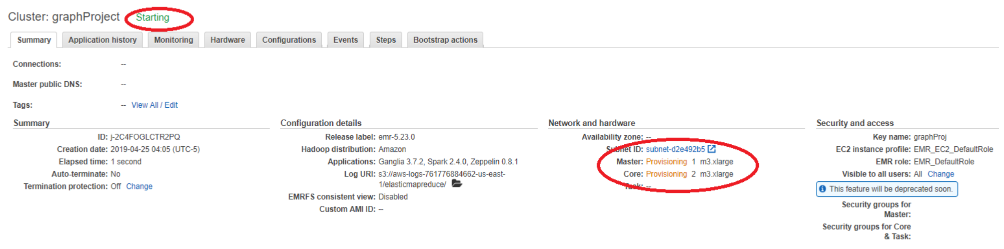
Instance type: m3.xlarge
Number of instances: 3 (1 master and 2 core nodes)

Security and access

EC2 key pair: graphProj
Permissions: [✓ Default] Custom
Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.
EMR role: EMR_DefaultRole
EC2 instance profile: EMR_EC2_DefaultRole

Figure 1.2: Create Cluster.

At the beginning the window will show the following,



Cluster: graphProject Starting

Connections: --
Master public DNS: --
Tags: -- View All / Edit

Summary
ID: j-2C4FOGLCTR2PQ
Creation date: 2019-04-25 04:05 (UTC-5)
Elapsed time: 1 second
Auto-terminate: No
Termination protection: Off Change

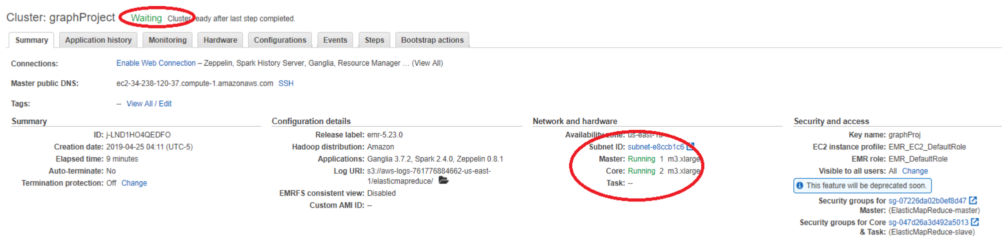
Configuration details
Release label: emr-5.23.0
Hadoop distribution: Amazon
Applications: Ganglia 3.7.2, Spark 2.4.0, Zeppelin 0.8.1
Log URI: s3://aws-logs-761776884662-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --

Network and hardware
Availability zone: --
Subnet ID: subnet-d2e492b5
Master: Provisioning 1 m3.xlarge
Core: Provisioning 2 m3.xlarge
Task: --

Security and access
Key name: graphProj
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Visible to all users: All Change
This feature will be deprecated soon.
Security groups for Master:
Security groups for Core & Task:

Figure 1.3: Creating Cluster.

When the cluster will be ready, the window will show the following,


Cluster: graphProject Waiting Cluster ready after last step completed.

Connections: Enable Web Connection – Zeppelin, Spark History Server, Ganglia, Resource Manager ... (View All)
Master public DNS: ec2-34-238-120-37.compute-1.amazonaws.com SSH
Tags: -- View All / Edit

Summary
ID: j-LND1HO4QEDFO
Creation date: 2019-04-25 04:11 (UTC-5)
Elapsed time: 9 minutes
Auto-terminate: No
Termination protection: Off Change

Configuration details
Release label: emr-5.23.0
Hadoop distribution: Amazon
Applications: Ganglia 3.7.2, Spark 2.4.0, Zeppelin 0.8.1
Log URI: s3://aws-logs-761776884662-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --

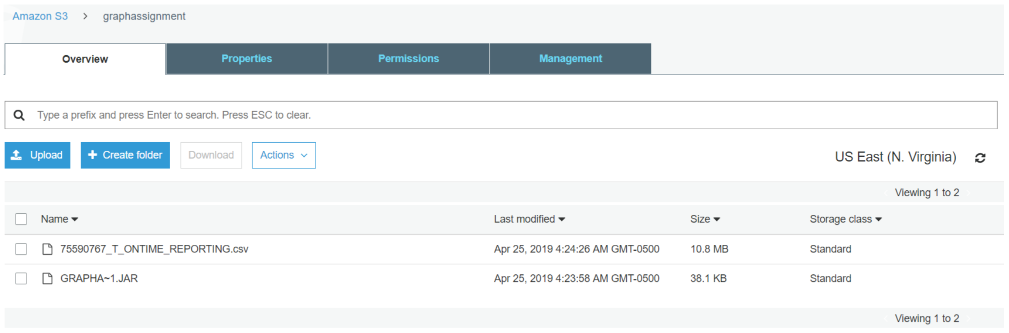
Network and hardware
Availability zone: us-east-1a
Subnet ID: subnet-e8ccb1c6
Master: Running 1 m3.xlarge
Core: Running 2 m3.xlarge
Task: --

Security and access
Key name: graphProj
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Visible to all users: All Change
This feature will be deprecated soon.
Security groups for Master: sg-07226da02b0ef8d47 (ElasticMapReduce-master)
Security groups for Core & Task: sg-047d26a3d492a5013 (ElasticMapReduce-slave)

Figure 1.4: Cluster Created.

Step 5: S3 Bucket

Now, upload the csv file you downloaded and the jar file I uploaded on eLearning.



Name	Last modified	Size	Storage class
75590767_T_ONTIME_REPORTING.csv	Apr 25, 2019 4:24:26 AM GMT-0500	10.8 MB	Standard
GRAPHA~1.JAR	Apr 25, 2019 4:23:58 AM GMT-0500	38.1 KB	Standard

Figure 1.5: S3 Bucket and File Upload.

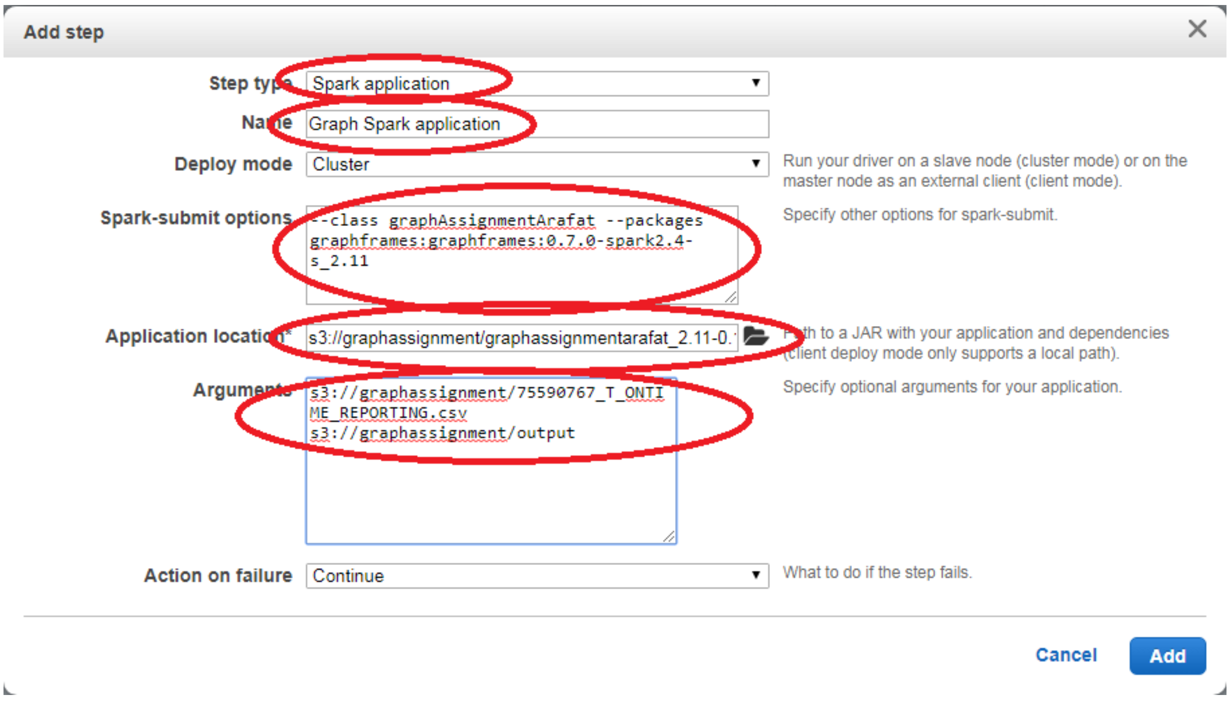
Step 6: Add Step for Execution and Provide Input Output File Path as Argument

Now, as the cluster is running and the required files are uploaded we are ready to add a step to execute the application and providing the data file path and output path. Following table and the figure demonstrates the setup,

```
--class graphAssignmentArafat --packages graphframes:graphframes
    :0.7.0-spark2.4-s_2.11

s3://graphassignment/graphassignmentarafat_2.11-0.1.jar


s3://graphassignment/75590767_T_ONTIME_REPORTING.csv
s3://graphassignment/output
```



Add step

Step type	Spark application	
Name	Graph Spark application	
Deploy mode	Cluster	Run your driver on a slave node (cluster mode) or on the master node as an external client (client mode).
Spark-submit options	--class graphAssignmentArafat --packages graphframes:graphframes:0.7.0-spark2.4-s_2.11	Specify other options for spark-submit.
Application location*	s3://graphassignment/graphassignmentarafat_2.11-0.	Path to a JAR with your application and dependencies (client deploy mode only supports a local path).
Arguments	s3://graphassignment/75590767_T_ONTIME_REPORTING.csv s3://graphassignment/output	Specify optional arguments for your application.
Action on failure	Continue	What to do if the step fails.

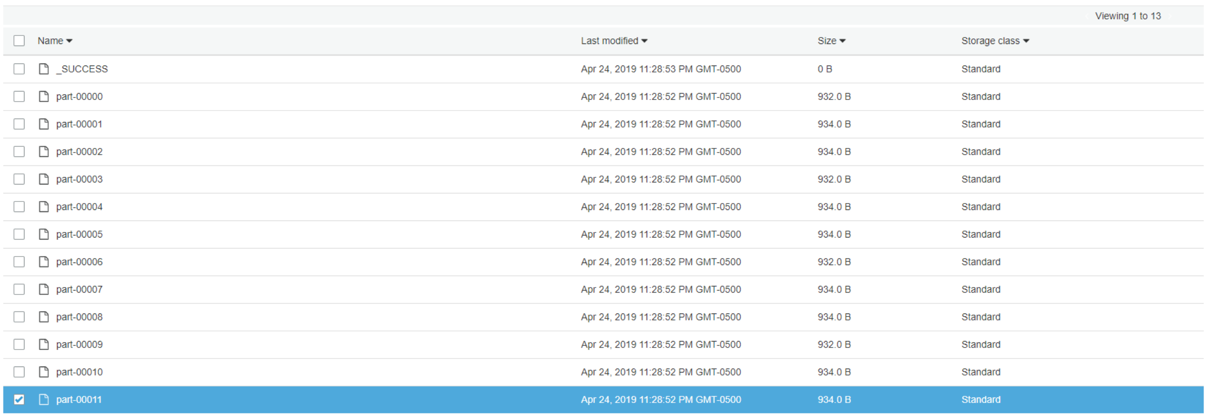
Figure 1.6: Add Step.

After the step is added the status of the execution will start. This program takes around 1.5 hours to execute.

Step 7: Check Output

You will see a folder named 'output' in your bucket on Amazon S3 once the program is completed. There are multiple files in the folder. These files can be downloaded to read the output. In my case there were

Now, finally, we are ready to see the output. Here, the output files can be opened using Notepad++ or Wordpad application. It is important to note that the output files should not be opened using Notepad. Because the formatting will better be demonstrated in Notepad++ or Wordpad. Notepad may not be able to handle Unix newlines. Chapter 3 shows the truncated sample output.



Viewing 1 to 13

Name	Last modified	Size	Storage class
_SUCCESS	Apr 24, 2019 11:28:53 PM GMT-0500	0 B	Standard
part-00000	Apr 24, 2019 11:28:52 PM GMT-0500	932.0 B	Standard
part-00001	Apr 24, 2019 11:28:52 PM GMT-0500	934.0 B	Standard
part-00002	Apr 24, 2019 11:28:52 PM GMT-0500	934.0 B	Standard
part-00003	Apr 24, 2019 11:28:52 PM GMT-0500	932.0 B	Standard
part-00004	Apr 24, 2019 11:28:52 PM GMT-0500	934.0 B	Standard
part-00005	Apr 24, 2019 11:28:52 PM GMT-0500	934.0 B	Standard
part-00006	Apr 24, 2019 11:28:52 PM GMT-0500	932.0 B	Standard
part-00007	Apr 24, 2019 11:28:52 PM GMT-0500	934.0 B	Standard
part-00008	Apr 24, 2019 11:28:52 PM GMT-0500	934.0 B	Standard
part-00009	Apr 24, 2019 11:28:52 PM GMT-0500	932.0 B	Standard
part-00010	Apr 24, 2019 11:28:52 PM GMT-0500	934.0 B	Standard
✓ part-00011	Apr 24, 2019 11:28:52 PM GMT-0500	934.0 B	Standard

Figure 1.7: Output Folder.

Chapter 2

Development

2.1 Development Stage on Databricks

First I developed the code on Databricks. In this stage I explicitly coded everything and and used the had coded path variables of the data files. The code is as follows,

```
import org.apache.spark.sql.functions._
import org.apache.spark.graphx._
import org.apache.spark.rdd.RDD
import org.graphframes._
val airport = spark.read.option("header","true").option("inferSchema","true").csv("/FileStore/tables/84936234
      _T_ONTIME_REPORTING.csv")
val graphTable = airport.groupBy("ORIGIN","DEST").count().orderBy(desc("count")).toDF("Origin","Dest","count")

----------------I SKIPPED THE LONG MIDDLE PORTION OF THE CODE --------------

val motifs = g.find("(A)-[]->(B); (B)-[]->(C); !(A)-[]->(C)")
var ABCList = motifs.select("A.name", "B.name", "C.name").toDF("A", "B", "C")
var Q6 = "\n\n6. Triplets [A, B, C] such that, at least one flight from A to B, and from B to C, but no flight from
      A to C.\n"
var resultABC = ABCList.rdd.map(_.toString())
Q6 += resultABC.collect().deep.mkString(",")
// -------------- Final Answer ----------------------- //
var Answer = Q1+Q2+Q3+Q4+Q5+Q6
println(Answer)
```

The output was as expected.

2.2 Development Stage on IntelliJ IDEA

2.2.1 Versions

First I created an empty project named `graphAssignmentArafat` where I used the following,

```
name := "\graphAssignmentArafat"
version := "0.1"
scalaVersion := "2.11.8"
val sparkVersion = "2.2.1"
```

2.2.2 Dependencies

For the project I also had to use the following dependencies,

```scala
import org.apache.spark.sql.functions._
import org.apache.spark.graphx._
import org.apache.spark.rdd.RDD
import org.apache.spark.sql.SQLContext
import org.graphframes._
import org.apache.spark.{SparkConf, SparkContext}
```

`org.apache.spark.sql.functions` has been imported for utilizing the functions available for DataFrame operations.

`org.apache.spark.graphx._` has been imported to utilize the GraphX library. This is a new component in Spark for graphs and graph-parallel computation. In addition, GraphX includes a growing collection of graph algorithms and builders to simplify graph analytics tasks.

A Resilient Distributed Dataset (RDD), the basic abstraction in Spark. That is why we need `org.apache.spark.rdd.RDD` for our spark application where RDD is in the core.

GraphFrames is a package for Apache Spark which provides DataFrame-based Graphs. Specially GraphFrames also support new algorithms:

1. Breadth-first search (BFS): Find shortest paths from one set of vertices to another
2. Motif finding: Search for structural patterns in a graph.

We will utilize Motif finding in our application. That is why we need `org.graphframes._` to be imported.

2.2.3 build.sbt file

I have created the following `build.sbt` file,

```scala
/*
Author: Md Arafat Hossain Khan, PhD Candidate
Department: Mathematics
*/

name := "graphAssignmentArafat"

version := "0.1"

scalaVersion := "2.11.8"

// Extra Dependencies

val sparkVersion = "2.2.1"

// Refer to https://mvnrepository.com/

libraryDependencies ++= Seq(
"org.apache.spark" %% "spark-core" % sparkVersion,
"org.apache.spark" %% "spark-graphx" % sparkVersion,
"org.apache.spark" %% "spark-sql" % sparkVersion,
"org.apache.spark" %% "spark-mllib" % sparkVersion % "runtime",
"org.apache.spark" %% "spark-streaming" % sparkVersion % "provided",
"org.apache.spark" %% "spark-hive" % sparkVersion % "provided"
)

// Instruction for graphframes is available at https://spark-packages.org/package/graphframes/graphframes
resolvers += "Spark Packages Repo" at "http://dl.bintray.com/spark-packages/maven"
libraryDependencies += "graphframes" % "graphframes" % "0.7.0-spark2.4-s_2.11"

// https://mvnrepository.com/artifact/org.apache.spark/spark-graphx
// libraryDependencies += "org.apache.spark" %% "spark-graphx" % "2.2.1"
// https://mvnrepository.com/artifact/org.apache.spark/spark-sql
// libraryDependencies += "org.apache.spark" %% "spark-sql" % "2.2.1"
// https://mvnrepository.com/artifact/org.apache.spark/spark-mllib
// libraryDependencies += "org.apache.spark" %% "spark-mllib" % "2.2.1" % "runtime"
// https://mvnrepository.com/artifact/org.apache.spark/spark-streaming
// libraryDependencies += "org.apache.spark" %% "spark-streaming" % "2.2.1" % "provided"
// https://mvnrepository.com/artifact/org.apache.spark/spark-hive
// libraryDependencies += "org.apache.spark" %% "spark-hive" % "2.2.1" % "provided"
```

2.2.4 Scala Class

I created one class names graphAssignmentArafat. In this class I used the input and output file paths as arguments. So, while using the class on AWS or Databricks I kept the option for choosing the desired csv file as an input as well as its path. Also the output has been saved in a folder named output. The class looks like the follwoing,

```
/*
Author: Md Arafat Hossain Khan, PhD Candidate
Department: Mathematics
*/

import org.apache.spark.sql.functions._
import org.apache.spark.graphx._
import org.apache.spark.rdd.RDD
import org.apache.spark.sql.SQLContext
import org.graphframes._
import org.apache.spark.{SparkConf, SparkContext}

object graphAssignmentArafat {
def main(args: Array[String]): Unit = {
if (args.length != 2) {
println("Usage: graphAssignmentArafat InputDir OutputDir")
}
// create Spark context with Spark configuration
val sc = new SparkContext(new SparkConf().setAppName("Spark graphAssignmentArafat"))
val sqlContext = new SQLContext(sc)
import sqlContext.implicits._

val Q0 = "This project was created by\n"+
"Md Arafat Hossain Khan\n"+
"PhD Candidate\n"+
"Department of Mathematics\nUniversity of Texas at Dallas"

///////////////// Code Start //////////////////////////

----------------I SKIPPED THE LONG MIDDLE PORTION OF THE CODE --------------

///////// -------- Code End --------- //////////////////

// Prepare output string
var answerRDD = sc.parallelize(Seq(Answer))

///////////////// Code End ///////////////////////////
answerRDD.map(r => r +"\n").saveAsTextFile(args(1))
}
}
```

2.3 Creating jar file

I used the command line to create the jar file. Following is the glimpse of the command line,

```
C:\Users\Md Arafat H Khan\Dropbox\Academics - University of Texas at Dallas\CS 6307.18s Introduction to Big Data
      Management and Analytics for non CS-Majors - Spring 2019\Assignment Solution\Assignment 03\
      graphAssignmentArafat>sbt
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=256m; support was removed in 8.0
[info] Loading global plugins from C:\Users\Md Arafat H Khan\.sbt\1.0\plugins
[info] Loading project definition from C:\Users\Md Arafat H Khan\Dropbox\Academics - University of Texas at Dallas\
      CS 6307.18s Introduction to Big Data Management and Analytics for non CS-Majors - Spring 2019\Assignment
      Solution\Assignment 03\graphAssignmentArafat\project
[info] Loading settings from build.sbt ...
[info] Set current project to graphAssignmentArafat (in build file:/C:/Users/Md%20Arafat%20H%20Khan/Dropbox/
      Academics%20-%20University%20of%20Texas%20at%20Dallas/CS%206307.18s%20Introduction%20to%20Big%20Data%20
      Management%20and%20Analytics%20for%20non%20CS-Majors%20-%20Spring%202019/Assignment%20Solution/Assignment
      %2003/graphAssignmentArafat/)
[info] sbt server started at local:sbt-server-cbb1816e2de0852a07f7
sbt:graphAssignmentArafat> compile
```

```
[info] Compiling 1 Scala source to C:\Users\Md Arafat H Khan\Dropbox\Academics - University of Texas at Dallas\CS
      6307.18s Introduction to Big Data Management and Analytics for non CS-Majors - Spring 2019\Assignment Solution
      \Assignment 03\graphAssignmentArafat\target\scala-2.11\classes ...
[warn] there was one deprecation warning; re-run with -deprecation for details
[warn] one warning found
[info] Done compiling.
[success] Total time: 27 s, completed Apr 25, 2019 3:30:09 AM
sbt:graphAssignmentArafat> package
[info] Packaging C:\Users\Md Arafat H Khan\Dropbox\Academics - University of Texas at Dallas\CS 6307.18s
      Introduction to Big Data Management and Analytics for non CS-Majors - Spring 2019\Assignment Solution\
      Assignment 03\graphAssignmentArafat\target\scala-2.11\graphassignmentarafat_2.11-0.1.jar ...
[info] Done packaging.
[success] Total time: 2 s, completed Apr 25, 2019 3:30:19 AM
sbt:graphAssignmentArafat>
```

`\graphAssignmentArafat\target\scala-2.11\` folder contains the jar file named `graphassignmentarafat_2.11-0.1.jar`.

Then follow the process of Chapter 1 to execute on AWS.

Chapter 3

Trancated Sample Output

The actual output is much bigger. I am presenting here the sample output read by Notepad++ after downloading from AWS.

```
1. Find the total number of airports (vertices) and the total flights
    connecting them.
Total number of airports :: 295
Total number of flights :: 464205

2. Find the in-degree and out-degree of each airport.
in-degree :: (ATL,151),(ORD,148),(DEN,124),(MSP,112),(DFW,106),(IAH
    ,103),(DTW,99),(PHX,89),(SLC,88),(EWR,85),(SFO,79),(LAX,79),(LAS
    ,75),(MCO,74),(SEA,71),...
out-degree :: (ATL,151),(ORD,148),(DEN,124),(MSP,112),(DFW,106),(IAH
    ,103),(DTW,99),(PHX,89),(SLC,88),(EWR,85),(SFO,79),(LAX,79),(LAS
    ,75),(MCO,74),(SEA,71),...

3. Find the top 10 airports with the highest in-degree and out-degree
    values.
in-degree :: (ATL,151),(ORD,148),(DEN,124),(MSP,112),(DFW,106),(IAH
    ,103),(DTW,99),(PHX,89),(SLC,88),(EWR,85)
out-degree :: (ATL,151),(ORD,148),(DEN,124),(MSP,112),(DFW,106),(IAH
    ,103),(DTW,99),(PHX,89),(SLC,88),(EWR,85)

4. Find out how many triangles can be formed around Dallas Fort Worth
    International (DFW) airport.
Total number of triangles formed around DFW :: 1235

5. Find out the top 10 airports with the highest page ranks.
Top 10 airports with the highest page ranks :: (ATL
    ,10.846702996395441),(ORD,10.100217228379133),(MSP
    ,7.753265350768704),(DEN,7.512767694389999),(IAH
    ,6.458574807471953),(DFW,5.980831089994658),(DTW
    ,5.927876189482903),(SLC,5.816666616332601),(PHX
    ,4.9464141894103575),(SFO,4.629206834210625)
```

```

6. Triplets [A, B, C] such that, at least one flight from A to B, and
    from B to C, but no flight from A to C.
[MCO,SFO,MCO],[PBI,ORD,MIA],[ORF,CLT,SAT],[FLL,TPA,MHT],[IND,DEN,SDF
    ],[CMH,ATL,RST],[CMH,MSP,BMI],[MKE,LAS,PIT],[OKC,SEA,BOS],[MSN,ORD
    ,ROA],...
```