Step1: Create the application for Network Word Count in eclipse

Create Object NetworkWordCount

* Import spark, streaming, StreamingContext packages
* Create a object NetworkWorkCount
* Create a main function
* Creates SparkConf with application name NeworkWordCount
* Creating StreamingContext with batch interval of 10 second
* Create socketTextSteam with IP as localhost and port as 9999 and return as lines
* Using flatMap, split lines to workds
* Use map and reduceByKey to calculate workd count for each word
* Print the word and word count
* Start streaming context

Code is as below:

import org.apache.spark.\_

import org.apache.spark.streaming.\_

object NetworkWordCount {

def main(args:Array[String]) {

val SparkConf = new SparkConf().setAppName("NetworkWordCount").setMaster("local[2]")

// Create a local StreamingContext with batch interval of 10 second

val ssc = new StreamingContext(sparkConf, Seconds(10))

/\* Create a DStream that will connect to hostname and port, like localhost 9999. As stated earlier, DStream will get created from StreamContext, which in return is created from SparkContext. \*/

val lines = ssc.socketTextStream("localhost",9999)

// Using this DStream (lines) we will perform transformation or output operation.

val words = lines.flatMap(\_.split(" "))

val wordCounts = words.map(x => (x, 1)).reduceByKey(\_ + \_)

wordCounts.print()

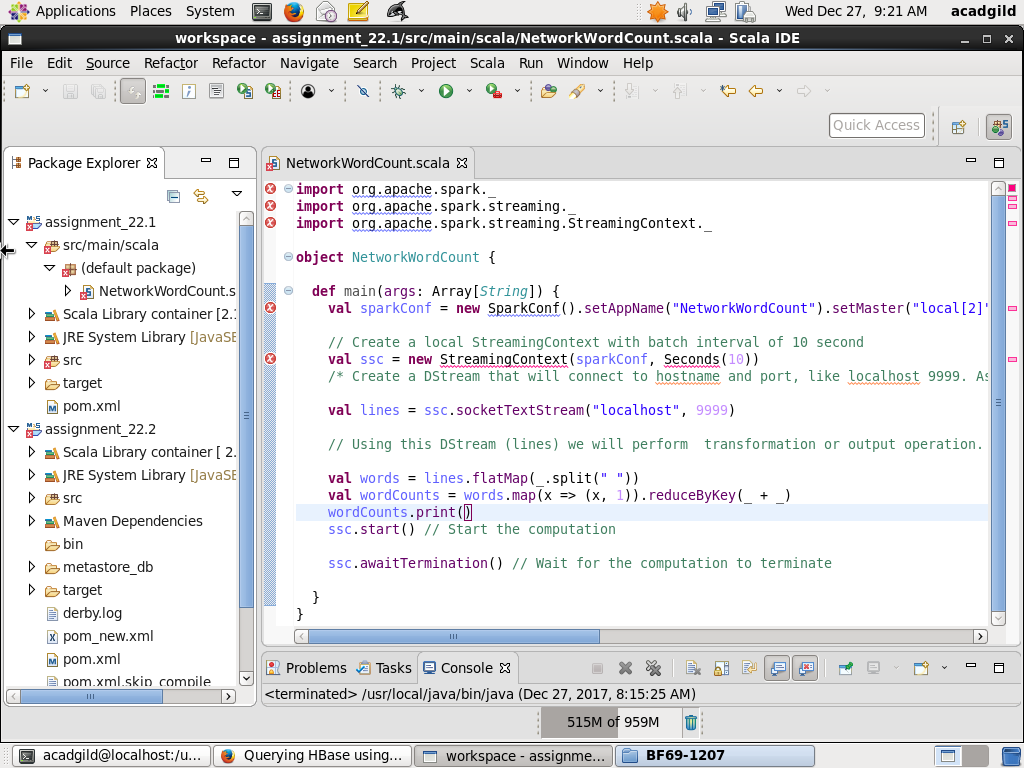
ssc.start() // Start the computation

ssc.awaitTermination() // Wait for the computation to terminate

}

}

Screenshot is as below:

* 

Step2: Create a maven project

Create a maven project and use pom.xml which will have all the dependencies. Pom.xml file is as

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>assignment\_22.1</groupId>

<artifactId>network\_word\_count</artifactId>

<version>0.0.1-SNAPSHOT</version>

<build>

<sourceDirectory>src/main/scala</sourceDirectory>

<resources>

<resource>

<directory>sr/main/scalac</directory>

<excludes>

<exclude>\*\*/\*.java</exclude>

</excludes>

</resource>

</resources>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.5.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<plugin>

<groupId>org.scala-tools</groupId>

<artifactId>maven-scala-plugin</artifactId>

<version>2.15.2</version>

<executions>

<execution>

<goals>

<goal>compile</goal>

</goals>

</execution>

</executions>

</plugin>

</plugins>

</build>

<dependencies>

<!-- https://mvnrepository.com/artifact/org.apache.spark/spark-core\_2.11 -->

<dependency>

<groupId>org.apache.spark</groupId>

<artifactId>spark-core\_2.11</artifactId>

<version>2.1.0</version>

</dependency>

<dependency>

<groupId>org.apache.spark</groupId>

<artifactId>spark-streaming\_2.11</artifactId>

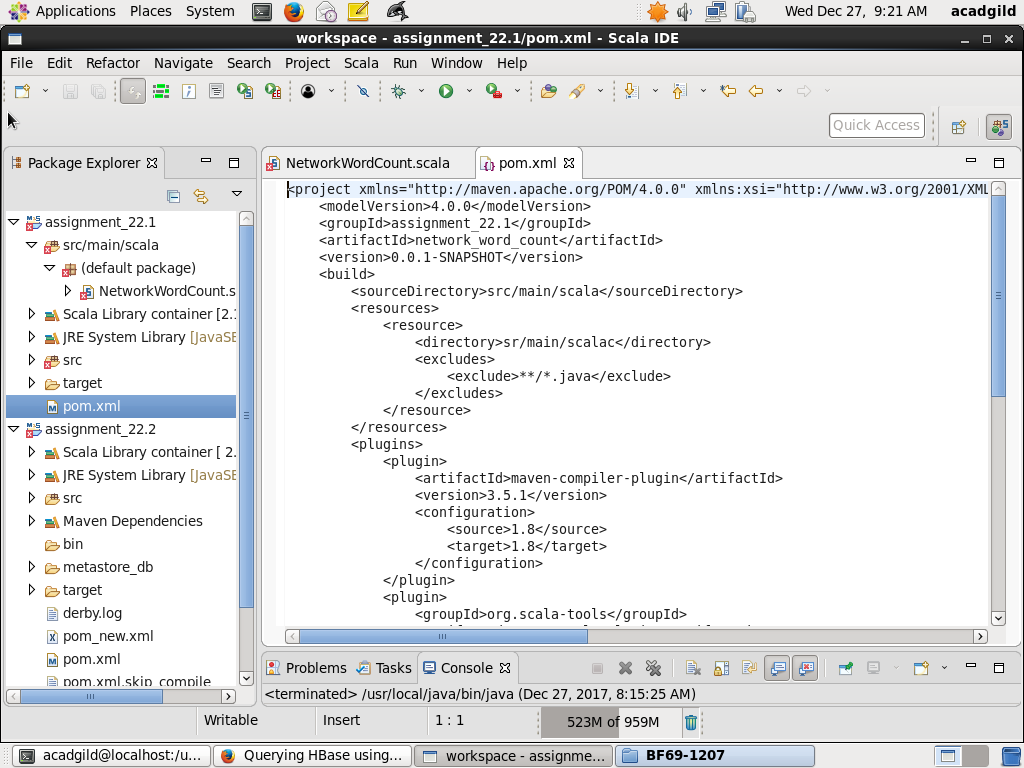
<version>2.1.0</version>

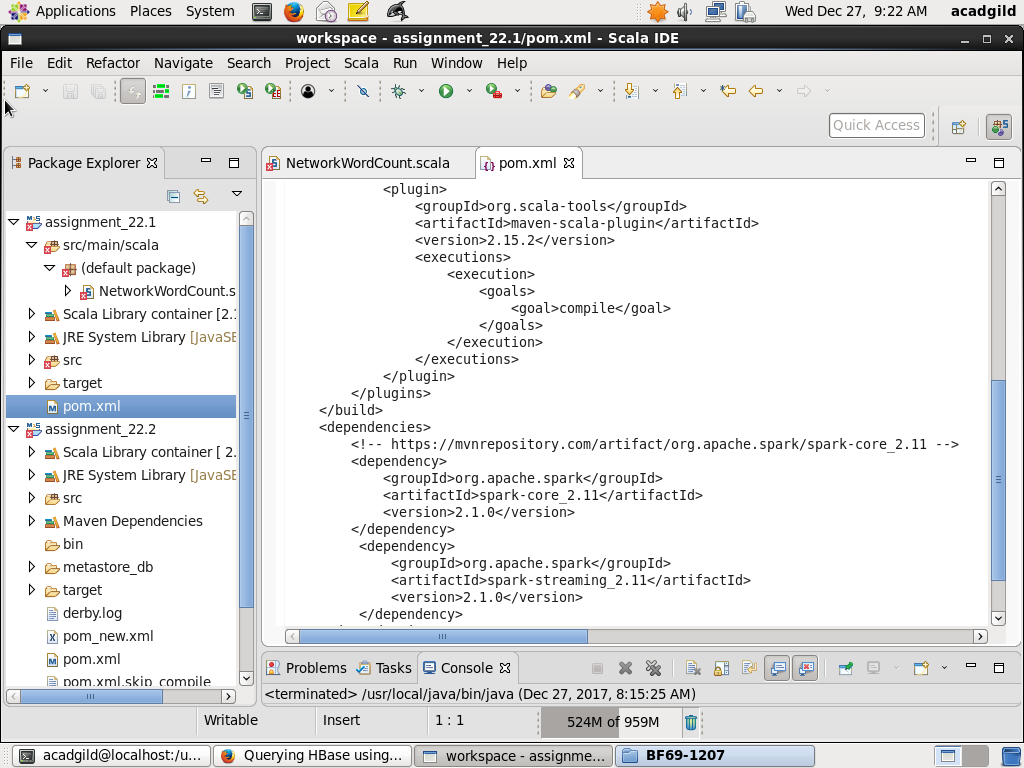
</dependency>

</dependencies>

</project>

Screenshots are as below:

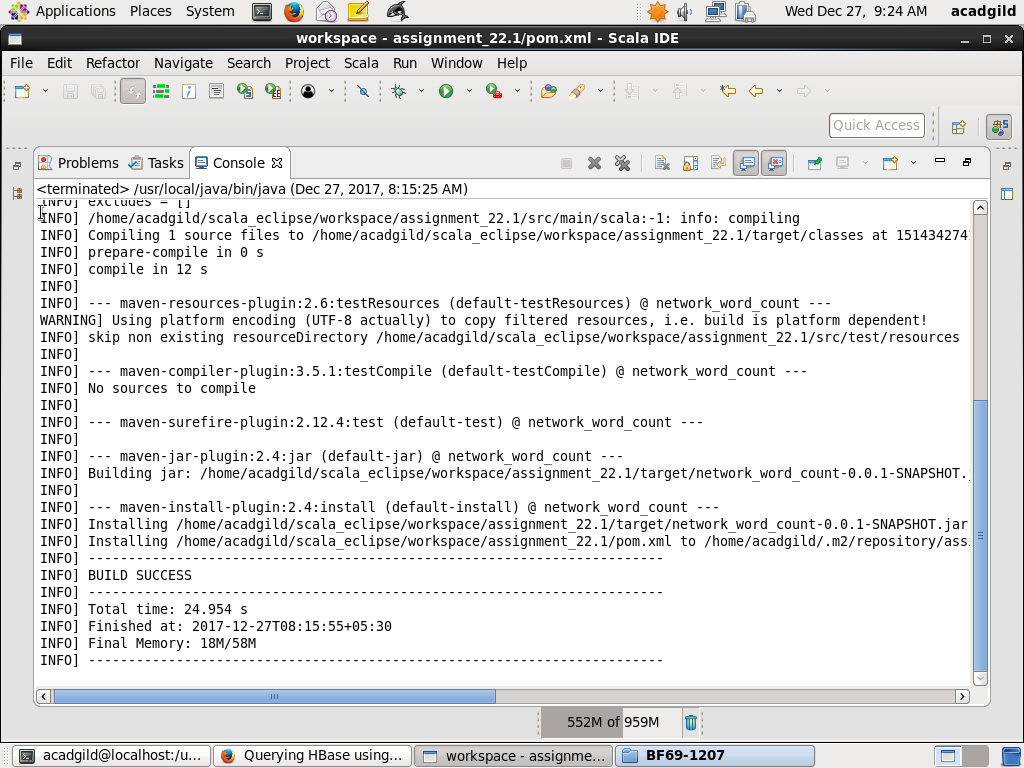




Step3: Compile the code

Using eclipse, create “maven clean” and “maven install”

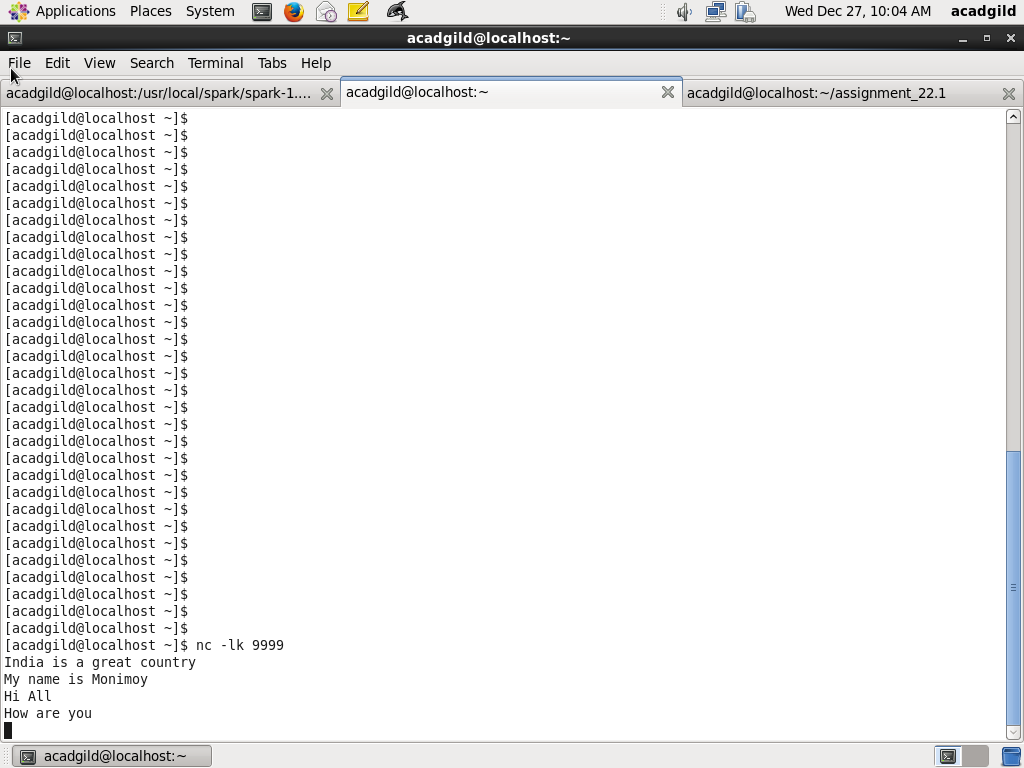
Screenshot is as below



Step4: Start NetCat

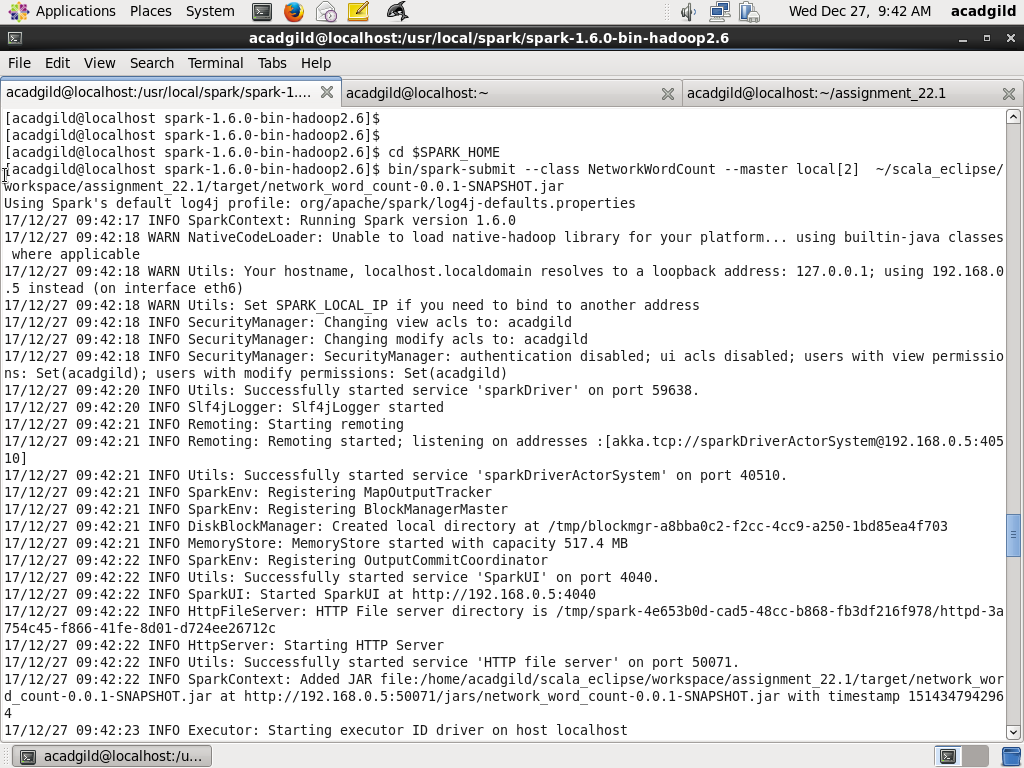
Use the command below to start netcat on port 9999

nc –lk 9999



Step4: Use spark-submit to deploy the application on spark

Use the spark-submit command below to deploy application on spark

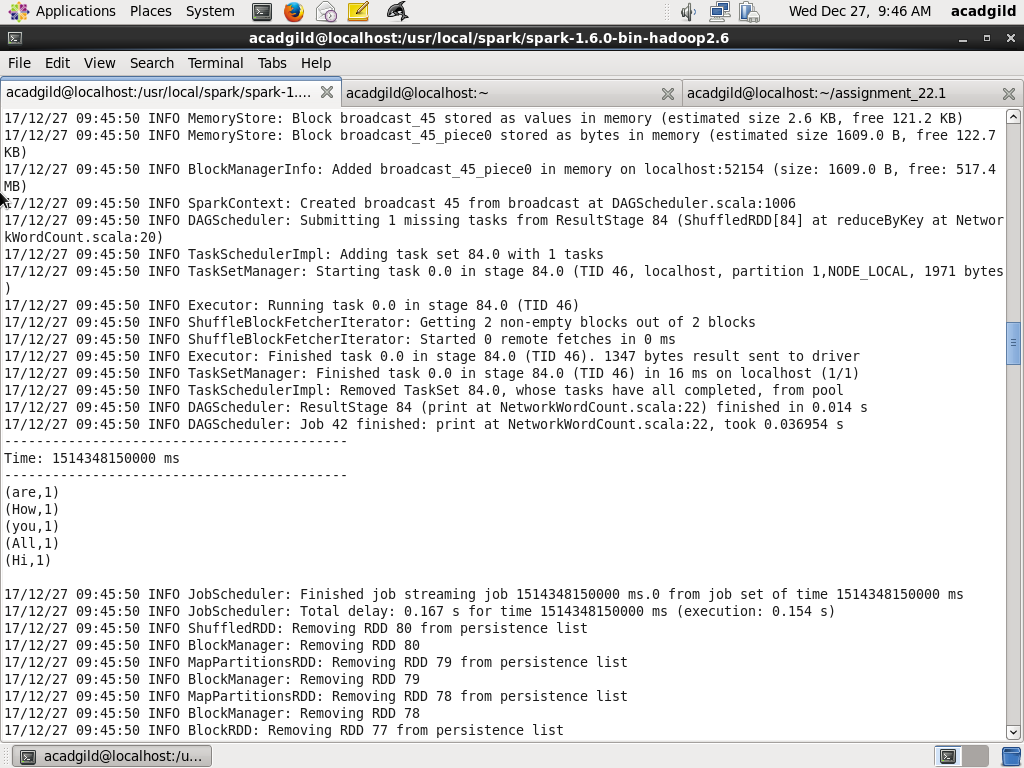


Step6: Word Count Output

Enter the line on the terminal where nc is run and check the word count

Following screenshot of wordcount for:

Hi all how are you



Following is the screenshot of wordcount for line entered:

India defeated England by 5 wickets

