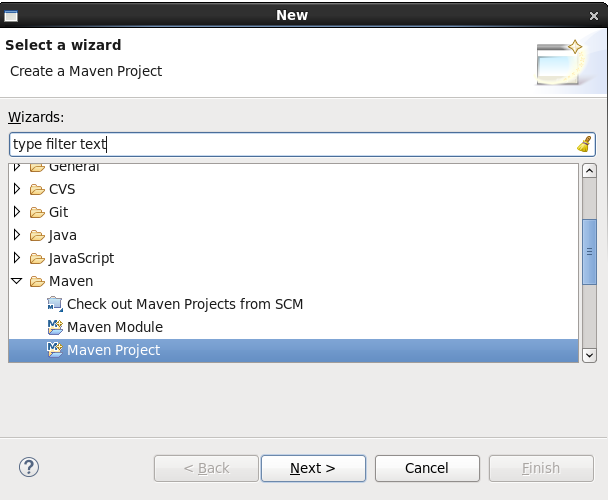
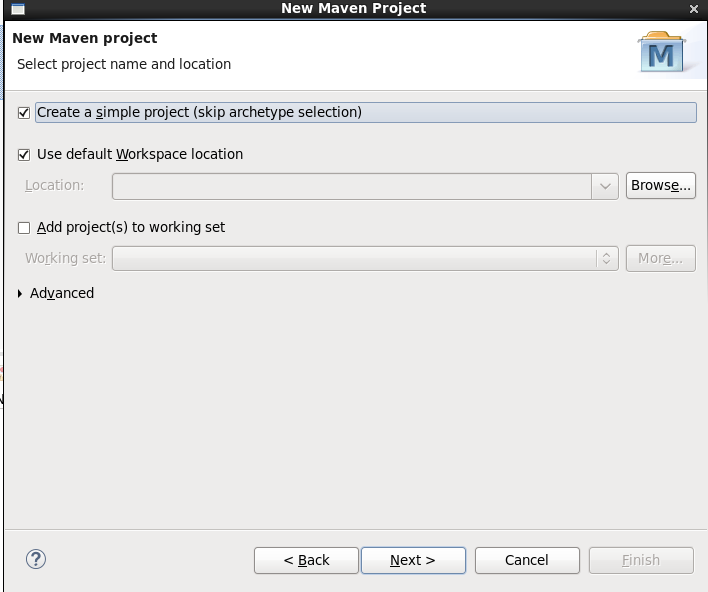
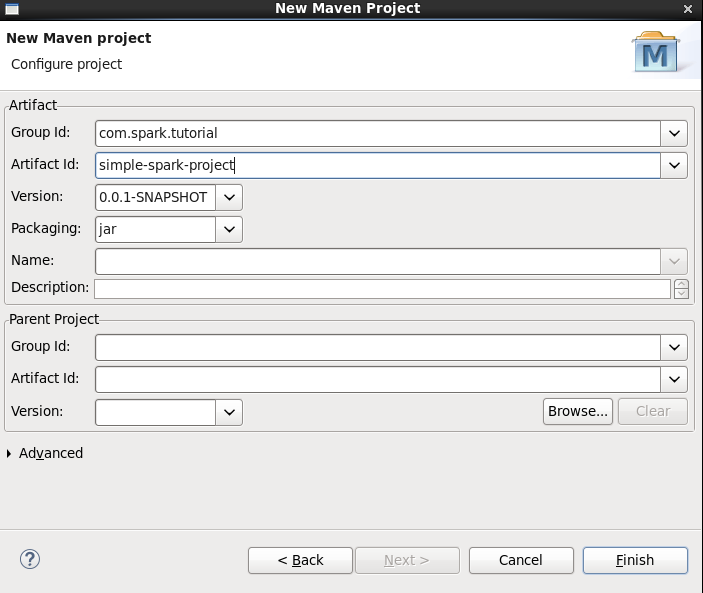
# Create Scala Project using Maven

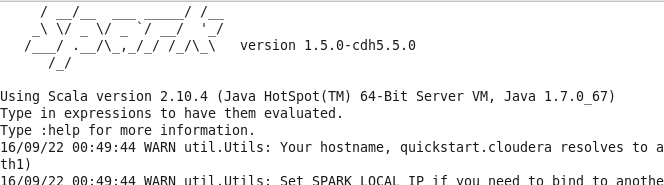






First check spark and scala version on which spark build upon

Once you start spark-shell you will get this info



Or type sc.version in spark-shell

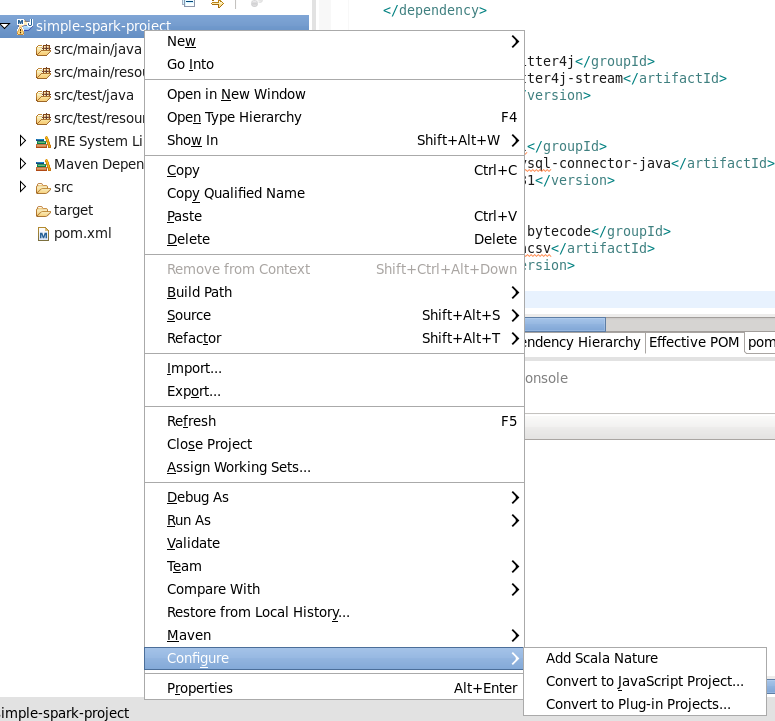


Note : You can find most of the jars under /usr/jars folder which are linked in other folder too like /usr/lib/spark/lib but we are going to use Maven to download the dependencies jars

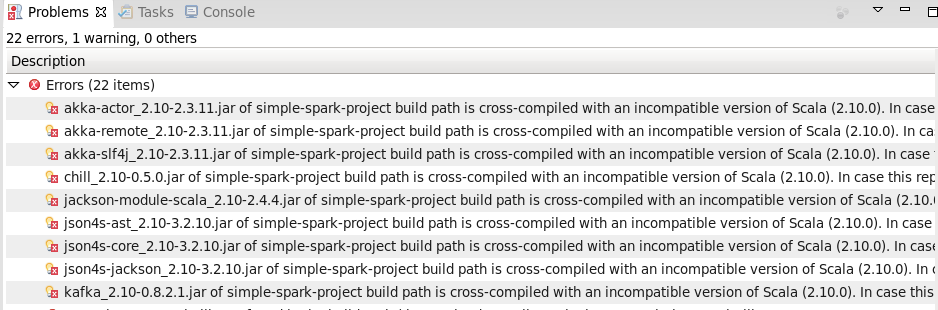
Now update the pom.xml as shown below and update project

|  |
| --- |
| <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>com.spark.tutorial</groupId>  <artifactId>simple-spark-project</artifactId>  <version>0.0.1-SNAPSHOT</version>  <dependencies>  <dependency>  <groupId>org.apache.spark</groupId>  <artifactId>spark-core\_2.10</artifactId>  <version>1.5.0</version>  </dependency>  <dependency>  <groupId>org.apache.spark</groupId>  <artifactId>spark-streaming-kafka\_2.10</artifactId>  <version>1.5.0</version>  </dependency>  <dependency>  <groupId>org.apache.spark</groupId>  <artifactId>spark-hive\_2.10</artifactId>  <version>1.5.0</version>  </dependency>  <dependency>  <groupId>com.databricks</groupId>  <artifactId>spark-csv\_2.10</artifactId>  <version>0.1</version>  </dependency>  <dependency>  <groupId>org.apache.spark</groupId>  <artifactId>spark-streaming\_2.10</artifactId>  <version>1.5.0</version>  <scope>provided</scope>  </dependency>  <dependency>  <groupId>org.apache.spark</groupId>  <artifactId>spark-streaming-twitter\_2.10</artifactId>  <version>1.5.0</version>  </dependency>    <dependency>  <groupId>org.twitter4j</groupId>  <artifactId>twitter4j-stream</artifactId>  <version>3.0.3</version>  </dependency>  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <version>5.1.31</version>  </dependency>  <dependency>  <groupId>au.com.bytecode</groupId>  <artifactId>opencsv</artifactId>  <version>2.4</version>  </dependency>  </dependencies>  </project> |

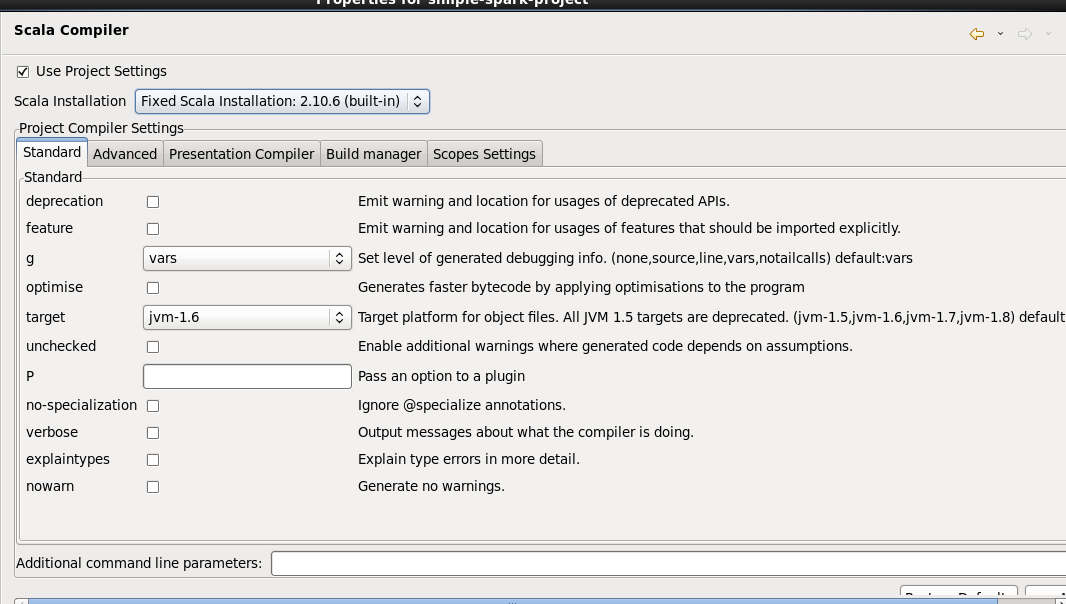
Right click on project and click on Configure>> Add scala nature



You will get below errors after adding Add Scala Nature

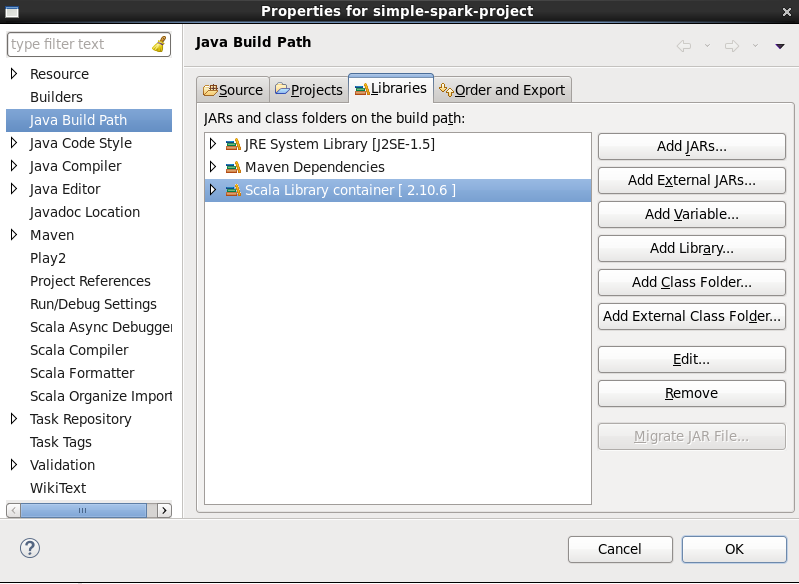


Right click on project >>Properties>>Scala Compiler

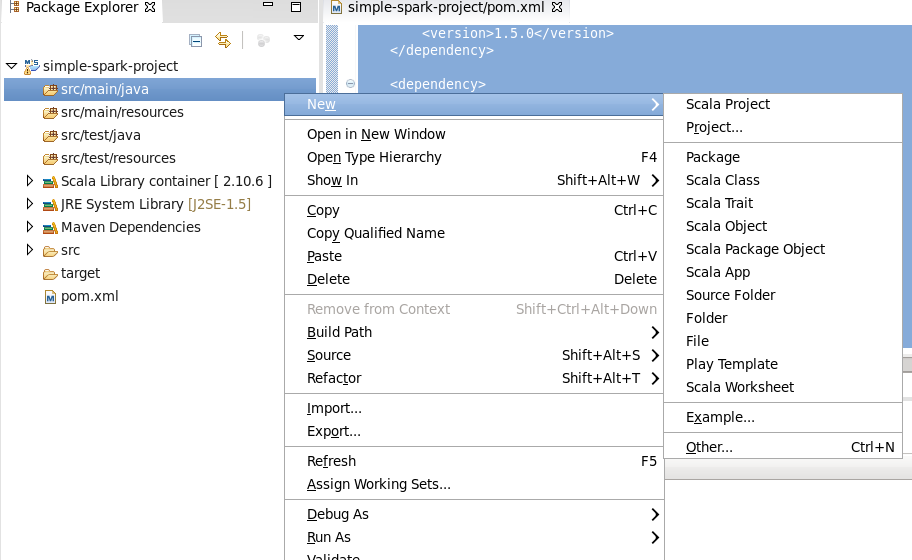


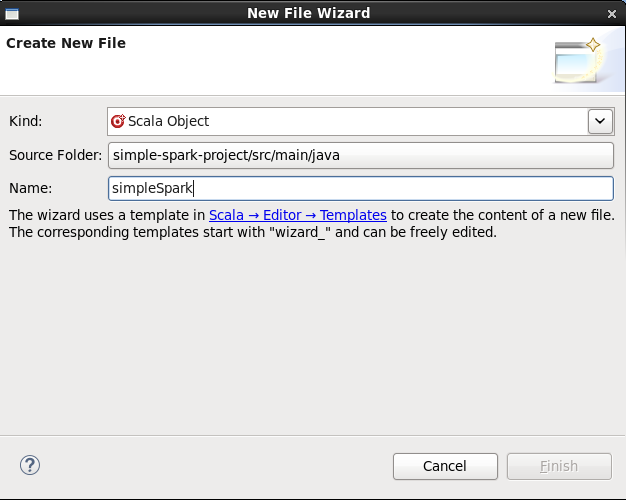
And choose 2.10 version of scala as our spark version build on 2.10 version

Remove the scala libarary from project as these are coming from spark configured under maven



Now create new Scala object

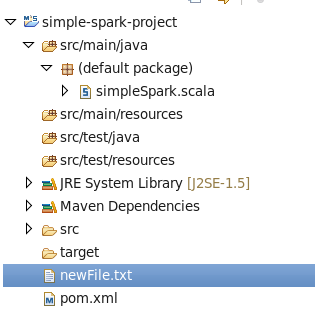




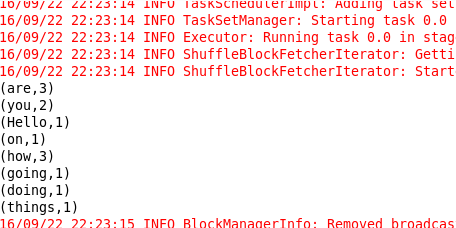
|  |
| --- |
| package org.apache.spark  import org.apache.spark.SparkContext  import org.apache.spark.SparkContext.\_  import org.apache.spark.SparkConf  import org.apache.spark.rdd.RDD.rddToPairRDDFunctions  object simpleSpark {  def main(args : Array[String]){  //println("hi1")  //val conf = new SparkConf().setAppName("Simple Application").setMaster("local[2]").set("spark.executor.memory", "1g")  val conf = new SparkConf().setAppName("Simple Application").setMaster("local")  val sc = new SparkContext(conf)  val myFile = sc.textFile("newFile.txt")  val wordspair =myFile.flatMap(row =>row.split(" ")).map(x=>(x,1)).reduceByKey(\_+\_)  wordspair.foreach(println)  wordspair.saveAsTextFile("myoutput");    }  } |

Create newFile.txt in project

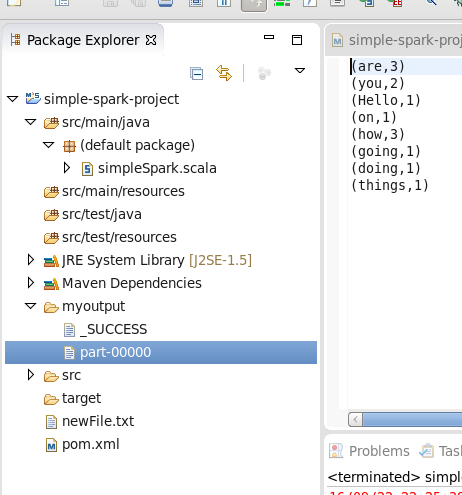
|  |
| --- |
| Hello how are you  how are things going on  how are you doing |



Now run the program



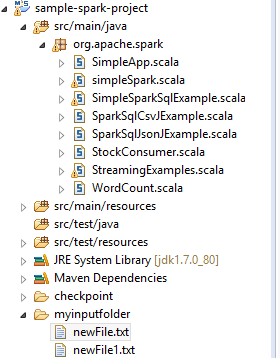
And check the output directory that we created in program



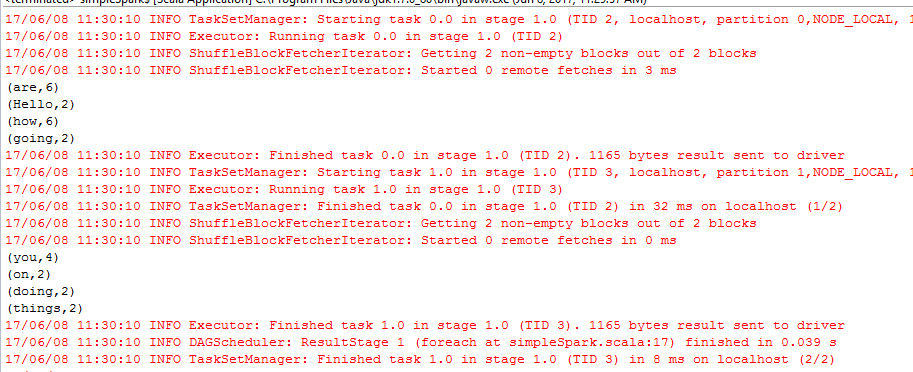
Or we can read multiple files from a folder

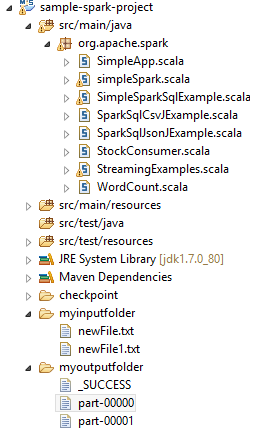
|  |
| --- |
| **package** org.apache.spark  **import** org.apache.spark.SparkContext  **import** org.apache.spark.SparkContext.\_  **import** org.apache.spark.SparkConf  **import** org.apache.spark.rdd.RDD.rddToPairRDDFunctions  **object** simpleSpark {  **def** main(args : Array[*String*]){  //println("hi1")  //val conf = new SparkConf().setAppName("Simple Application").setMaster("local[2]").set("spark.executor.memory", "1g")  **val** conf = **new** SparkConf().setAppName("Simple Application").setMaster("local")  **val** sc = **new** SparkContext(conf)  **val** myFile = sc.textFile("myinputfolder")  **val** wordspair =myFile.flatMap(row =>row.split(" ")).map(x=>(x,1)).reduceByKey(\_+\_)  wordspair.foreach(println)  wordspair.saveAsTextFile("myoutputfolder");    }  } |

Now create myinputfolder and put two file with same content (newfile.txt,newfile1.txt)



Now run this job

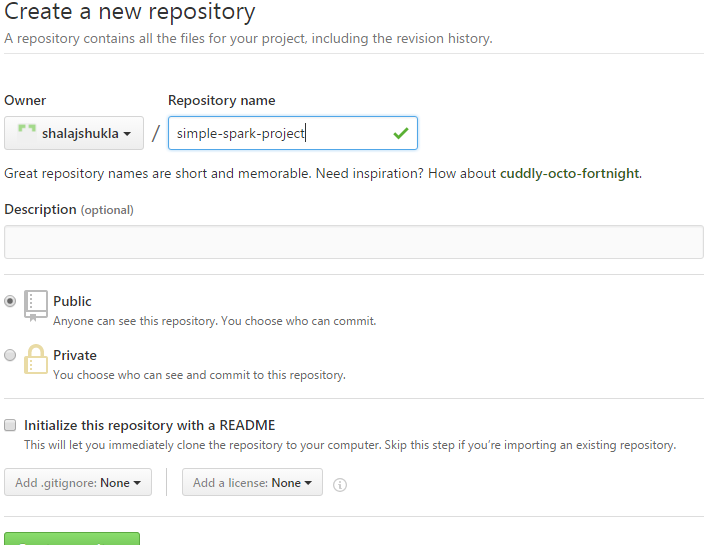




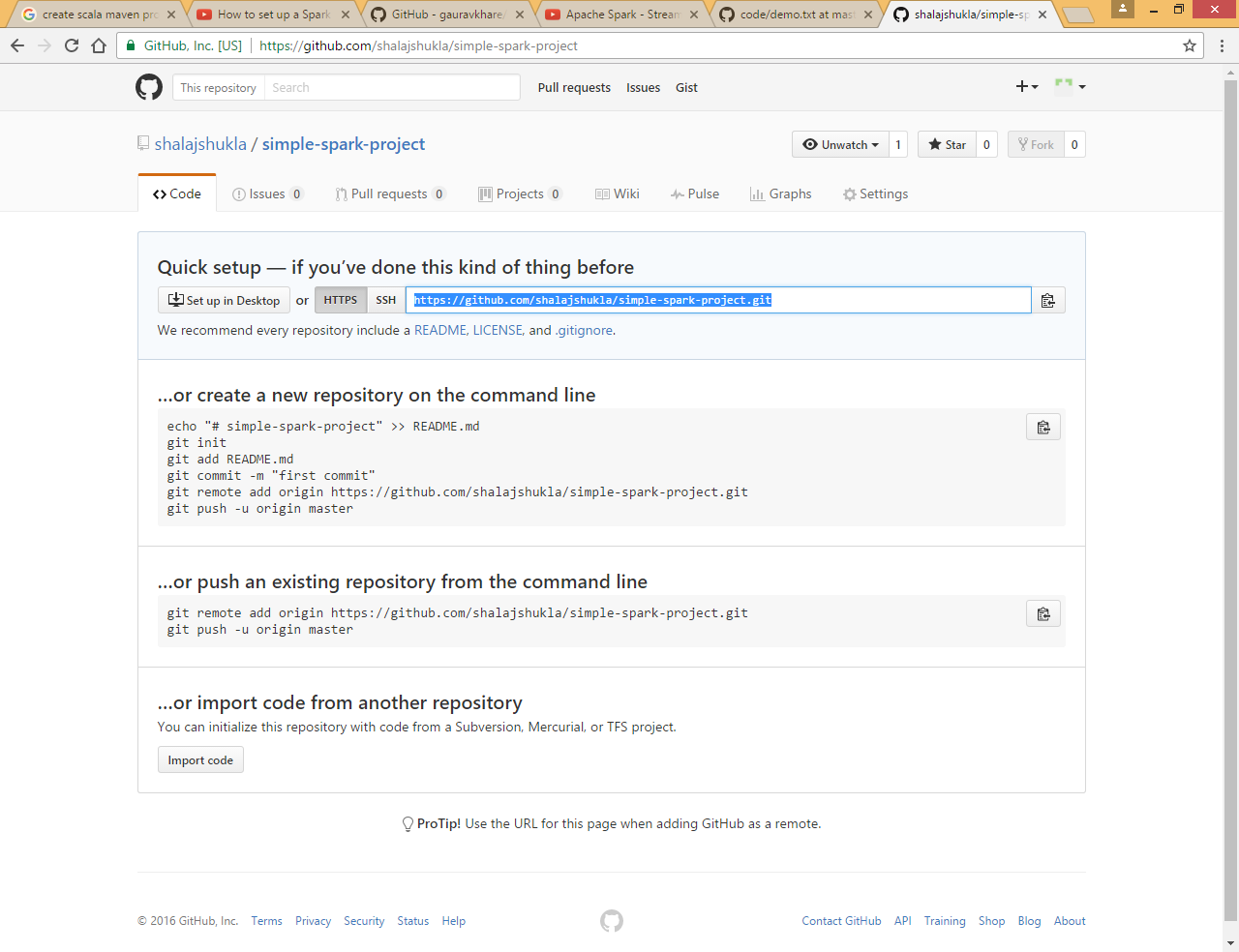
Now we will have two part files under myoutputfolder

# Push Code to Git

To share this code on github create a new repository in your git account

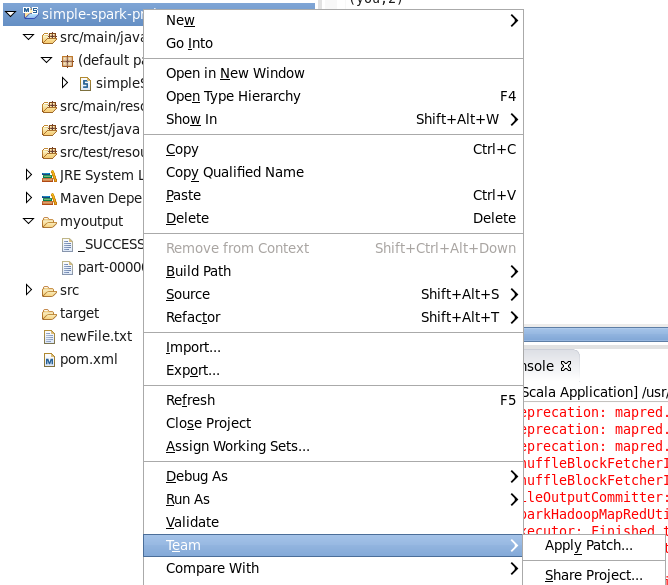


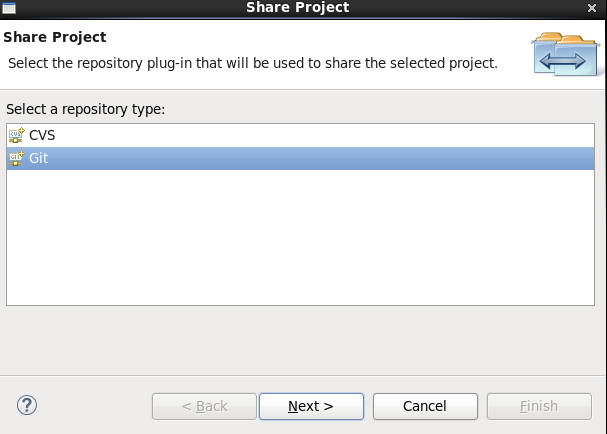
Copy the address



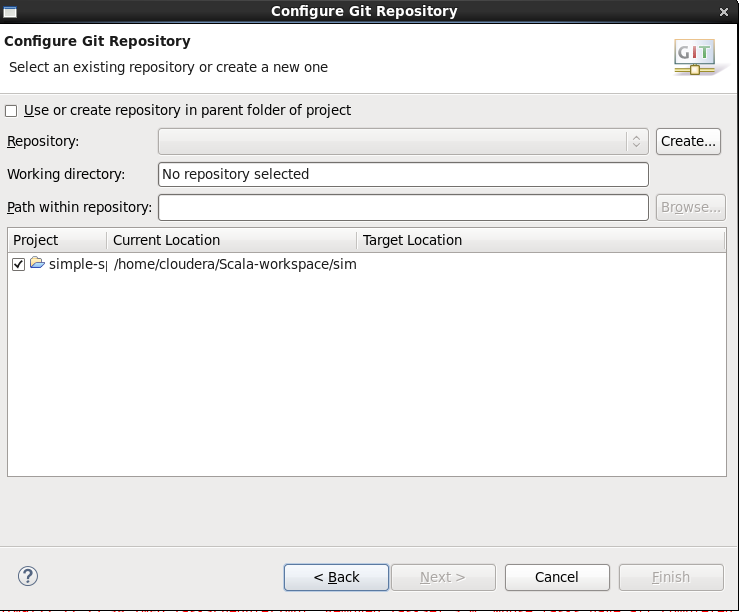
Now right click on project

And choose Team >>Share project

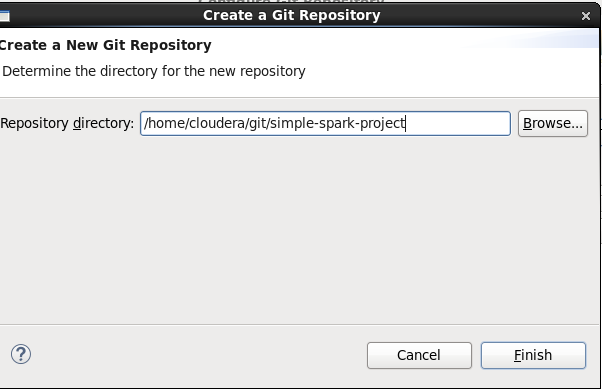




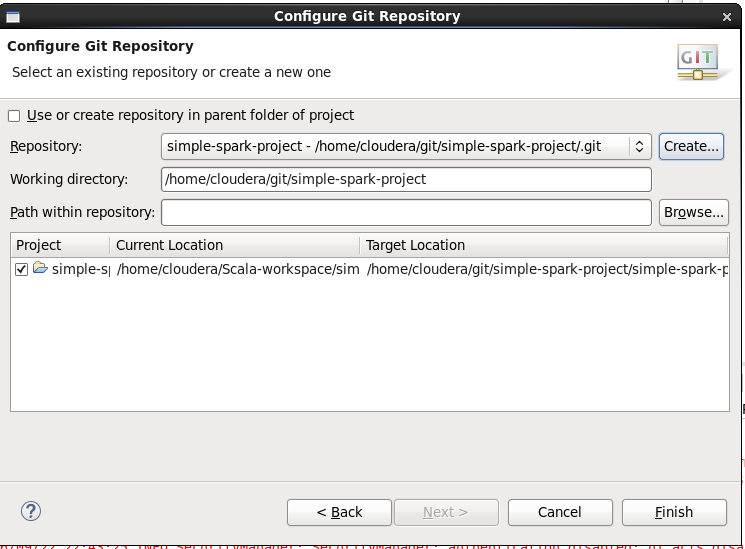
Choose Git and click on Next



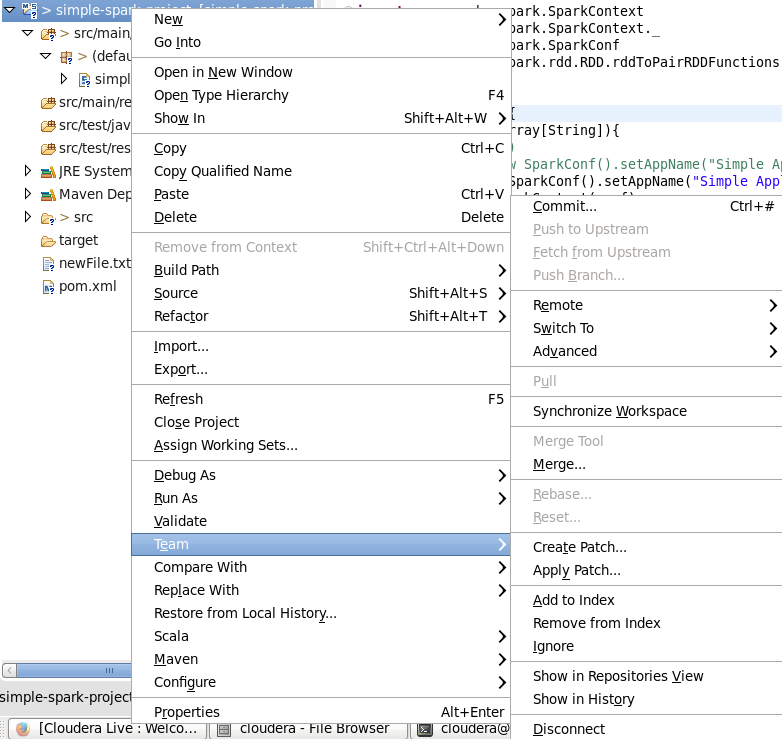
Create a local repository for your project

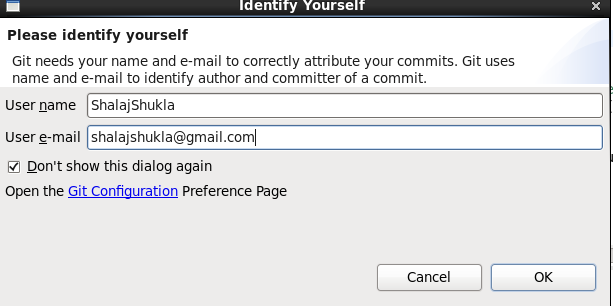


Now click on finish

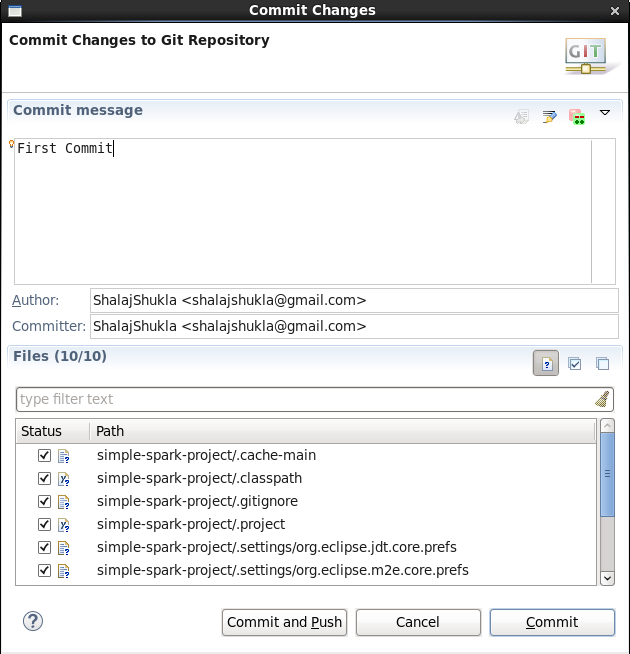


Now right click on project and choose Team>>commit

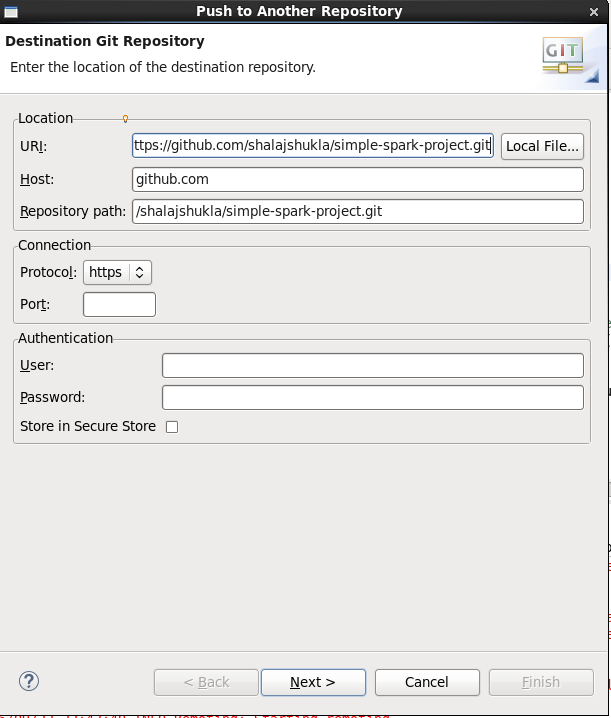




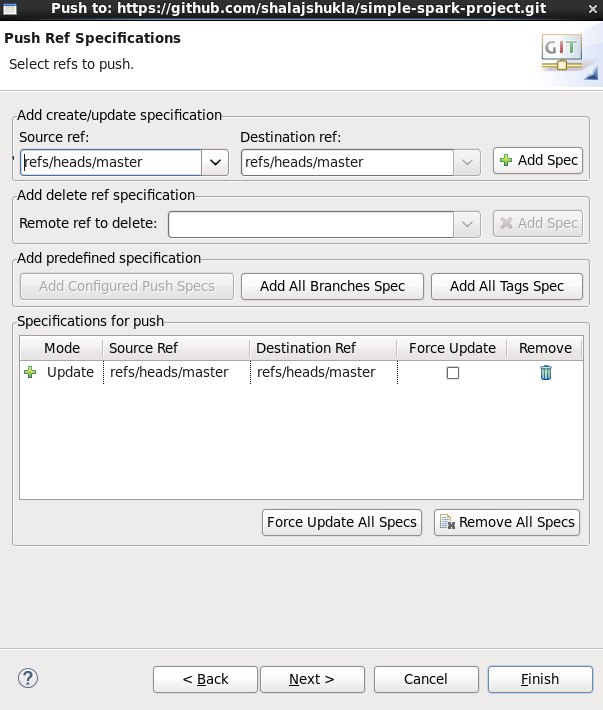
Click OK



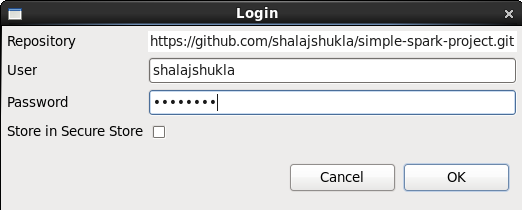
Select files and click Commit and Push



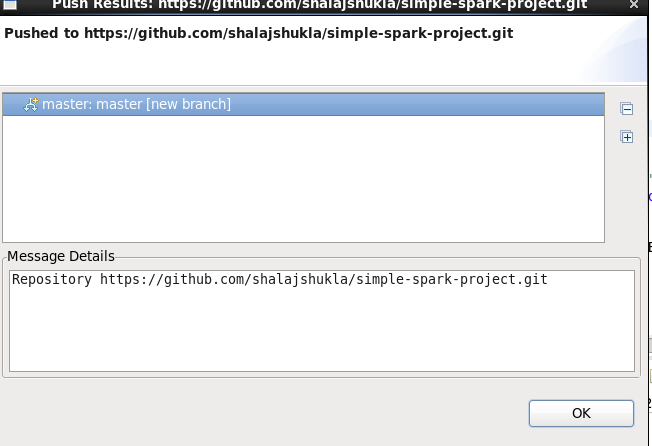
Now past the path from git account to URI and click on Next



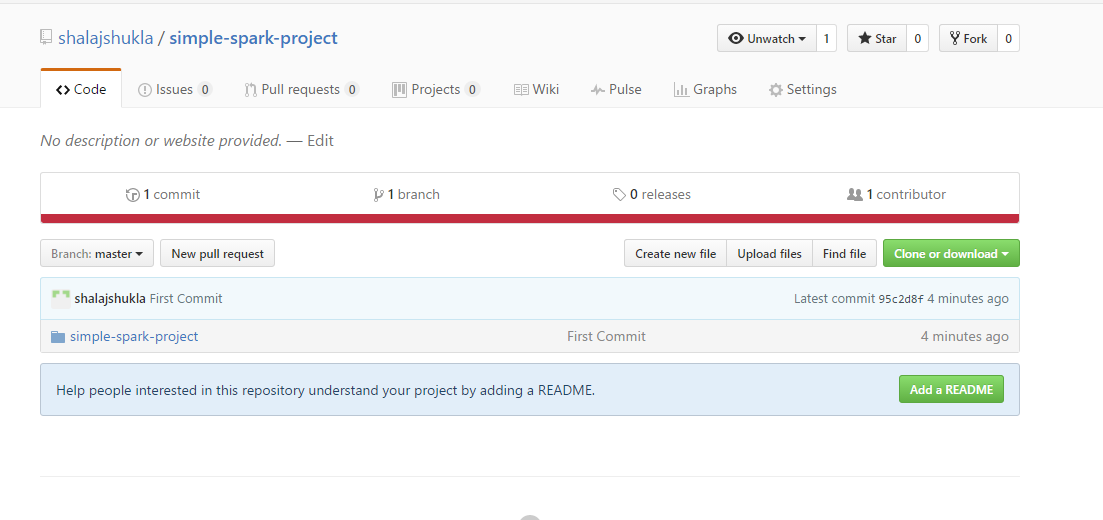
Select Master branch and click Add Spec and click Finish



Provide git credential and click OK



Now check Git Account



# Submit the job from command line

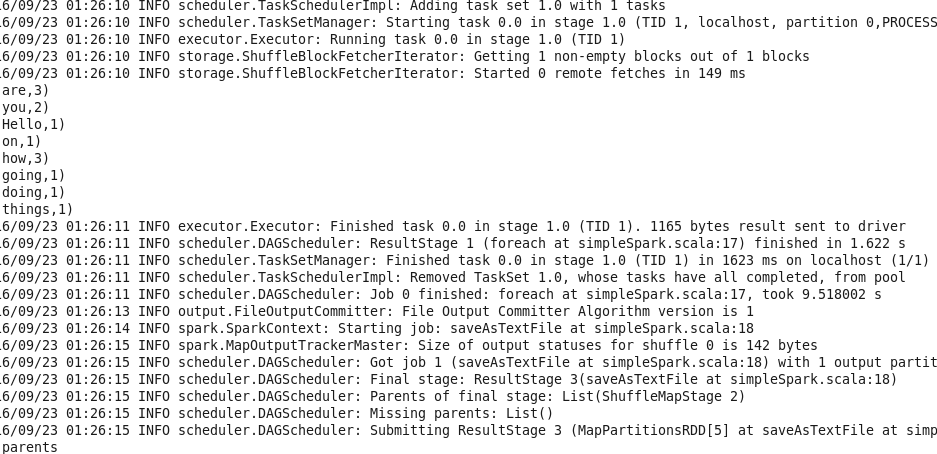
First Create a jar file namely SimpleSparkProject.jar and put it under some folder

Put newFile.txt file on root of Hadoop file system using below command

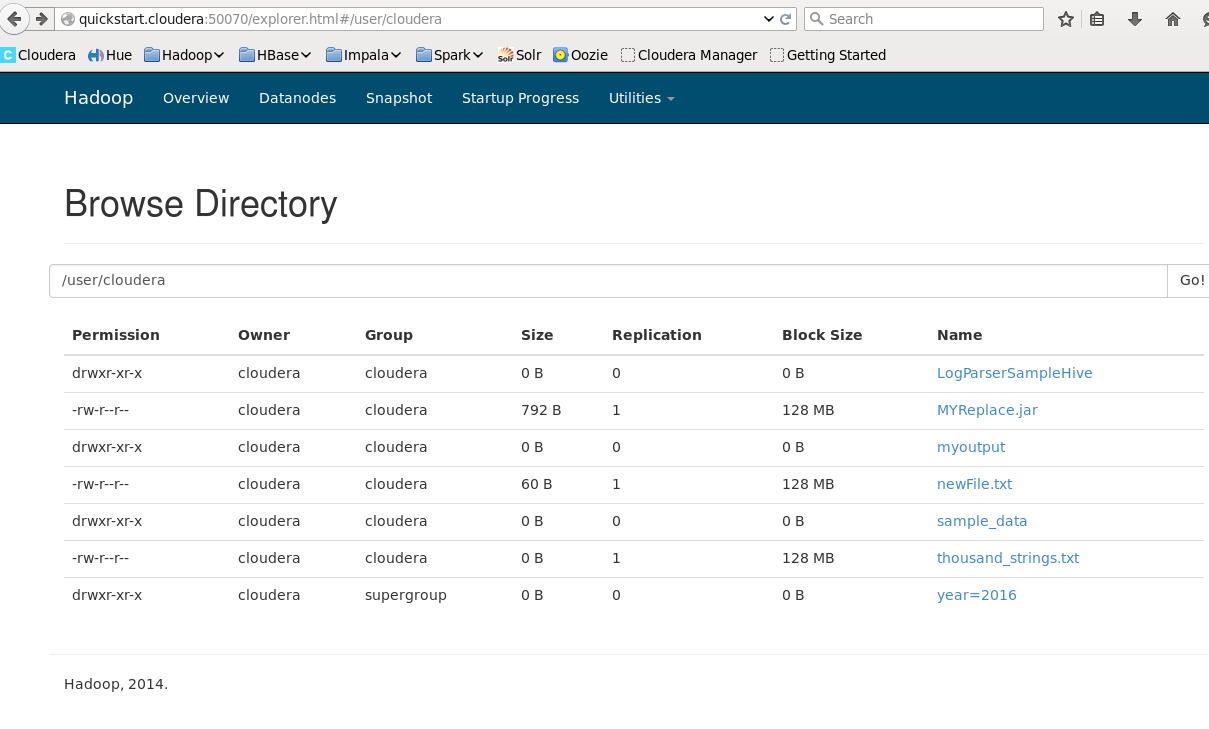
|  |
| --- |
| hdfs dfs -put newFile.txt ./ |

Now go to that folder in terminal and issue below command

|  |
| --- |
| spark-submit --class simpleSpark --master local SimpleSparkProject.jar |



Now go to Hadoop NameNode UI and browse file system from utilities tab



Click on myoutput folder

