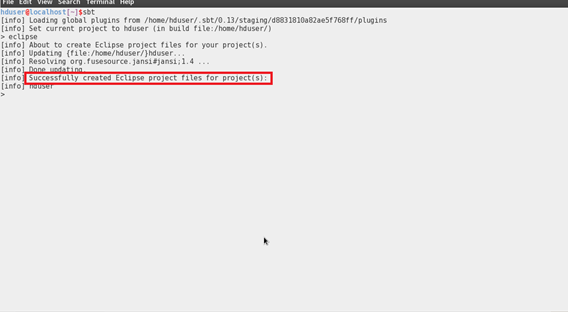
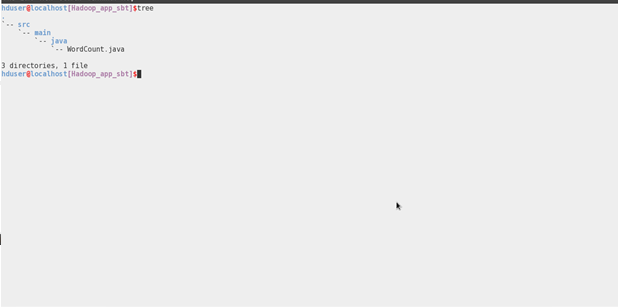
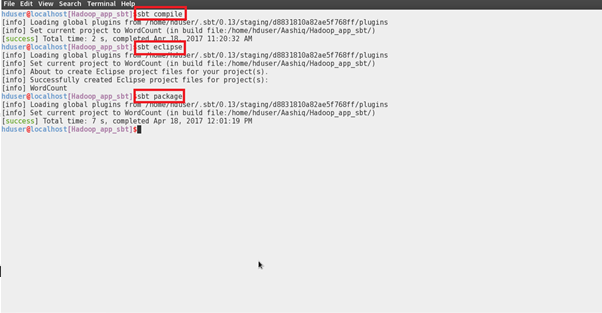
**Build and Deploy Hadoop word count program using SBT.**

**Creating the project:**

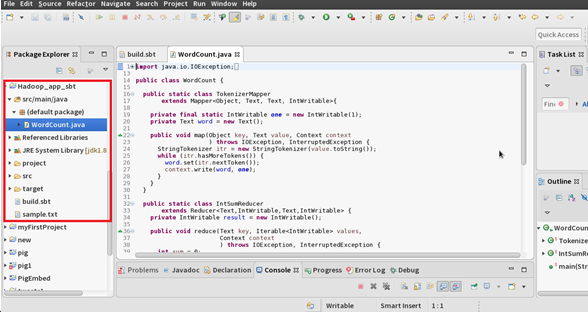




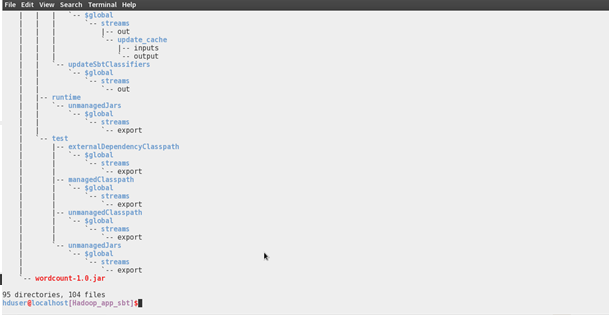
**Compiling the code and Building the Project:**



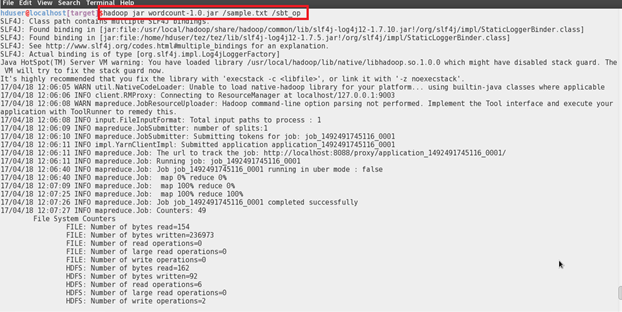
**Importing the Project into the Eclipse:**



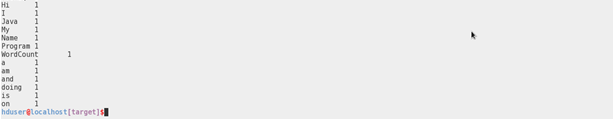
**Jar Created-Tree Diagram:**



**Running the Jar File:**



**Output of the WordCount Program:**



**Build.sbt.txt**

name := "WordCount"

version := "1.0"

scalaVersion := "2.11.6"

scalacOptions += "-target:jvm-1.8"

publishMavenStyle := true

//dependencies for Hadoop program

libraryDependencies += "org.apache.hadoop" % "hadoop-mapreduce-client-core" % "2.7.1" libraryDependencies += "org.apache.hadoop" % "hadoop-common" % "2.7.1"

javacOptions in (Compile, compile) ++= Seq("-source", "1.8", "-target", "1.8", "-g:lines")

mainClass in (Compile,run) := Some("WordCount")

//specifying fully qualified path of main class crossPaths := false

autoScalaLibrary := false //runs a pure java program

resolvers += Resolver.file("Frozen IVY2 Cache Dependences", file("/home/luis/.ivy2/cache")) (Resolver.ivyStylePatterns) ivys "/home/luis/.ivy2/cache/[organisation]/[module]/ivy-[revision].xml" artifacts "/home/luis/.ivy2/cache/[organisation]/[module]/[type]s/[module]-[revision].[type]"

**plugins.sbt.txt:**

addSbtPlugin("com.typesafe.sbteclipse" % "sbteclipse-plugin" % "4.0.0")

addSbtPlugin("com.eed3si9n" % "sbt-assembly" % "0.13.0")

addSbtPlugin("org.xerial.sbt" % "sbt-pack" % "0.5.1")

**wordcount.java:**

|  |
| --- |
|  |
|  | import java.util.StringTokenizer; |
|  |  |
|  |  |
|  | import org.apache.hadoop.conf.Configuration; |
|  | import org.apache.hadoop.fs.Path; |
|  | import org.apache.hadoop.io.IntWritable; |
|  | import org.apache.hadoop.io.Text; |
|  | import org.apache.hadoop.mapreduce.Job; |
|  | import org.apache.hadoop.mapreduce.Mapper; |
|  | import org.apache.hadoop.mapreduce.Reducer; |
|  | import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; |
|  | import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; |
|  |  |
|  | public class WordCount { |
|  |  |
|  | public static class TokenizerMapper |
|  | extends Mapper<Object, Text, Text, IntWritable>{ |
|  |  |
|  | private final static IntWritable one = new IntWritable(1); |
|  | private Text word = new Text(); |
|  |  |
|  | public void map(Object key, Text value, Context context |
|  | ) throws IOException, InterruptedException { |
|  | StringTokenizer itr = new StringTokenizer(value.toString()); |
|  | while (itr.hasMoreTokens()) { |
|  | word.set(itr.nextToken()); |
|  | context.write(word, one); |
|  | } |
|  | } |
|  | } |
|  |  |
|  | public static class IntSumReducer |
|  | extends Reducer<Text,IntWritable,Text,IntWritable> { |
|  | private IntWritable result = new IntWritable(); |
|  |  |
|  | public void reduce(Text key, Iterable<IntWritable> values, |
|  | Context context |
|  | ) throws IOException, InterruptedException { |
|  | int sum = 0; |
|  | for (IntWritable val : values) { |
|  | sum += val.get(); |
|  | } |
|  | result.set(sum); |
|  | context.write(key, result); |
|  | } |
|  | } |
|  |  |
|  | public static void main(String[] args) throws Exception { |
|  | Configuration conf = new Configuration(); |
|  | Job job = Job.getInstance(conf, "word count"); |
|  | job.setJarByClass(WordCount.class); |
|  | job.setMapperClass(TokenizerMapper.class); |
|  | job.setCombinerClass(IntSumReducer.class); |
|  | job.setReducerClass(IntSumReducer.class); |
|  | job.setOutputKeyClass(Text.class); |
|  | job.setOutputValueClass(IntWritable.class); |
|  | FileInputFormat.addInputPath(job, new Path(args[0])); |
|  | FileOutputFormat.setOutputPath(job, new Path(args[1])); |
|  | System.exit(job.waitForCompletion(true) ? 0 : 1); |
|  | } |
|  | } |