## Write Self-Contained Spark Applications in Eclipse

0. Download and configure Java 8

The Scala-IDE only supports Java 8, but the VM installed Java 11. Please follow the below steps to install and configure Java 8.

**$ sudo apt install openjdk-8-jdk**

**$ sudo update-alternatives --config java**

You will see both Java 8 and Java 11, please select Java 8 as the default Java version. Also do this for javac.

**$ sudo update-alternatives --config javac**

Now check the Java version, you would see that the version has been changed to Java 8.



1. Download Scala-IDE

Download from <http://scala-ide.org/download/sdk.html>.

If you use Mac OS, please download the Mac version.

If you work in the virtual machine, in your home folder, you can use the following command:

**$ wget http://downloads.typesafe.com/scalaide-pack/4.7.0-vfinal-oxygen-212-20170929/scala-SDK-4.7.0-vfinal-2.12-linux.gtk.x86\_64.tar.gz**

Note that the VM already has installed Eclipse. First rename it, and then uncompress the downloaded package.

**$ mv eclipse eclipse.mr**

**$ tar xvf scala-SDK-4.7.0-vfinal-2.12-linux.gtk.x86\_64.tar.gz**

2. Configure your compiler compliance level to 1.8 in the IDE as follows.

In Eclipse, go to menu Windows → Preferences, select Java, and expand it. Then select Compiler and change the compliance level to 1.8. Also guarantee that the Installed JREs is located at /usr/lib/jvm/java-8-openjdk-amd64.



3. Select File->New->Scala Project to create a Scala project. Name the project as “WordCountSpark”.

Right click the project, Properties->Java Build Path->Libraries->Add External JARs, go to the directory “/home/comp9313/spark/jars”, and add all jars into the project.

4. Create a new package “comp9313.lab6” in this project, and then right click the package and create a Scala object “WordCount”



Next, copy the following code into the file:

|  |
| --- |
| **package comp9313.lab6****import org.apache.spark.SparkContext****import org.apache.spark.SparkContext.\_****import org.apache.spark.SparkConf****object WordCount {** **def main(args: Array[String]) {** **val inputFile = args(0)** **val outputFolder = args(1)** **val conf = new SparkConf().setAppName("WordCount").setMaster("local")** **val sc = new SparkContext(conf)** **val input = sc.textFile(inputFile)** **val words = input.flatMap(line => line.split(" "))** **val counts = words.map(word => (word, 1)).reduceByKey(\_+\_)** **counts.saveAsTextFile(outputFolder)** **}****}** |

5. Right click the file WordCount.scala and double click Run as->Run Configurations->Scala Application. In the dialog, click the tab “Main”, and make input “comp9313.lab6.WordCount” as the “Main class”.



## Then configure the arguments for this project: make the arguments as “hdfs://localhost:9000/user/comp9313/input/pg100.txt hdfs://localhost:9000/user/comp9313/output”. Finally, click “Run”.

Start HDFS first, and upload pg100.txt to HDFS first!

You can also use a local file as the input, such as “file:///home/comp9313/pg100.txt”

## 6. If everything works normally, you will see the Spark running message in Eclipse console:



Wait until the program finishes and go to HDFS to check the results. You can debug and test your code in the IDE, which is easier than packaging your project and then submitting it to Spark.

7. You can try more examples given at: <http://spark.apache.org/examples.html>.