# Create Spark Scala project using SBT

Here we will see how to setup Spark Scala project using sbt , here I am building spark project in windows environment and later going to move jar file to linux file system where actual spark service is running

Create working directory for new project

For my example it is D:\ShalajS\bigdata-spark-scala\simple-spark

Under working directory create src/main/scala directory structure

Check the scala and spark version in remote linux machine by issuing command spark-shell

|  |
| --- |
| [root@mac127 ~]# spark-shell  Setting default log level to "WARN".  To adjust logging level use sc.setLogLevel(newLevel).  Welcome to  \_\_\_\_ \_\_  / \_\_/\_\_ \_\_\_ \_\_\_\_\_/ /\_\_  \_\ \/ \_ \/ \_ `/ \_\_/ '\_/  /\_\_\_/ .\_\_/\\_,\_/\_/ /\_/\\_\ version 1.6.0  /\_/  Using Scala version 2.10.5 (Java HotSpot(TM) 64-Bit Server VM, Java 1.7.0\_67)  Type in expressions to have them evaluated.  Type :help for more information.  Spark context available as sc (master = yarn-client, app id = application\_1484320332413\_0004).  SQL context available as sqlContext. |

So our spark version is 1.6.0 and which is using scala 2.10.5 version

Now we need to create build.sbt under our project directory with name, version, scalaVersion

Also update build.sbt with libraryDepencies for spark

|  |
| --- |
| name := "simple-spark-scala"  version := "1.0"  scalaVersion := "2.10.5"  libraryDependencies += "org.apache.spark" % "spark-core\_2.10" % "1.6.0" |

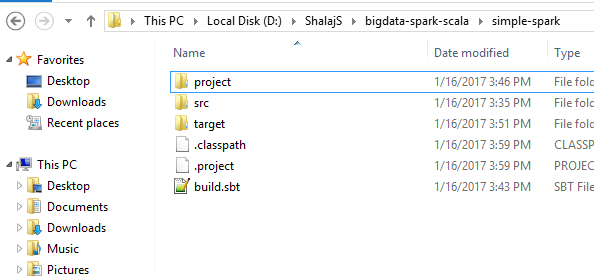
Here 1.6.0 is spark version

Run ***sbt eclipse*** to build eclipse project for scala, it will download necessary jar files



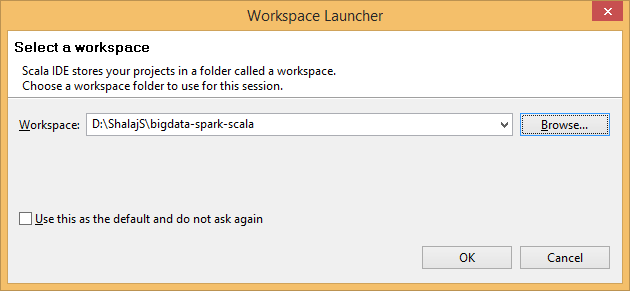


Project structure look like below

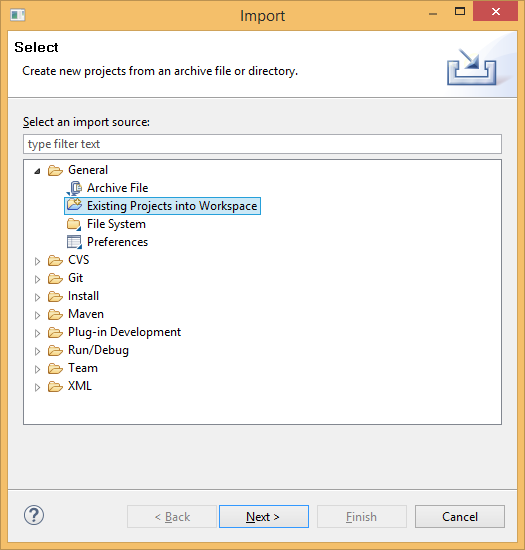


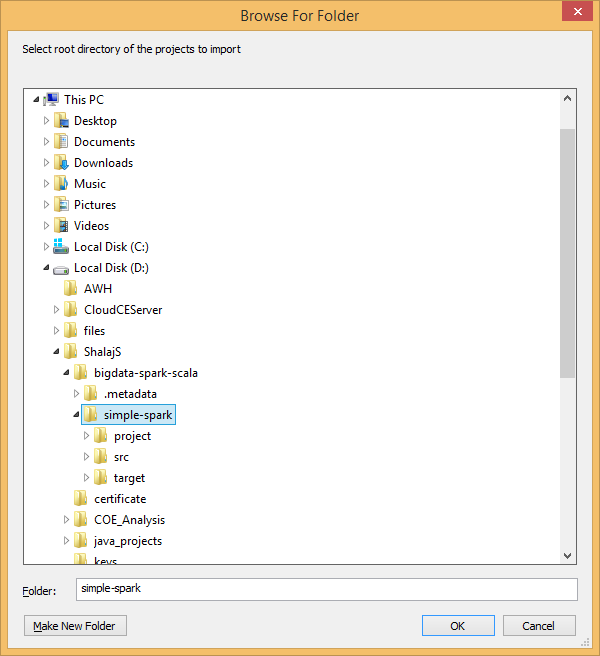
Now open eclipse

Select bigdata-spark-scala as workspace



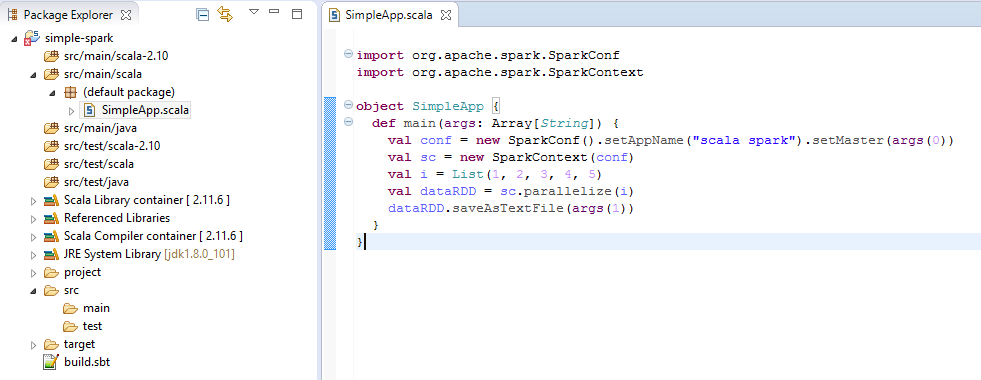
Now import project simple-spark, go to File>>Import…



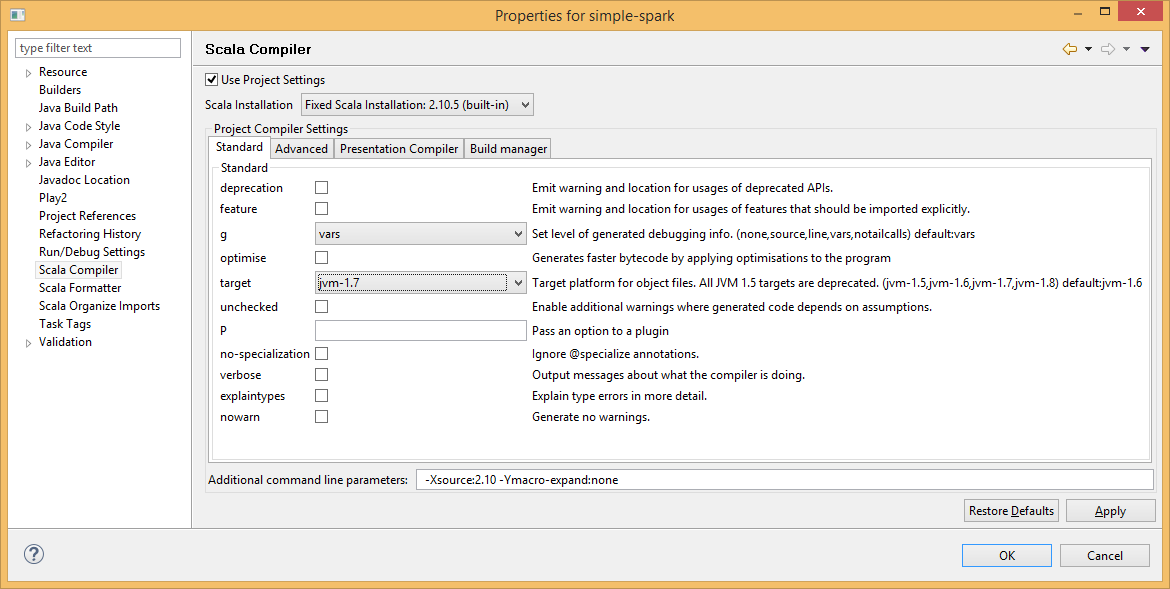


Now create simpleApp.scala object under src/main/scala

|  |
| --- |
| **import** org.apache.spark.SparkConf  **import** org.apache.spark.SparkContext  **object** SimpleApp {  **def** main(args: Array[*String*]) {  **val** conf = **new** SparkConf().setAppName("scala spark").setMaster(args(0))  **val** sc = **new** SparkContext(conf)  **val** i = List(1, 2, 3, 4, 5)  **val** dataRDD = sc.parallelize(i)  dataRDD.saveAsTextFile(args(1))  }  } |

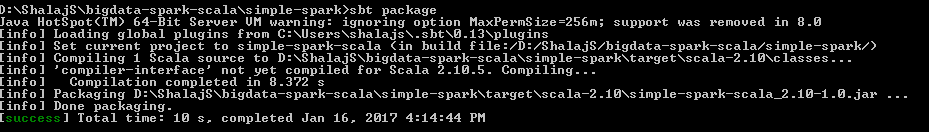


Some time you face some issue regarding compilation and JRE level so change it accordingly

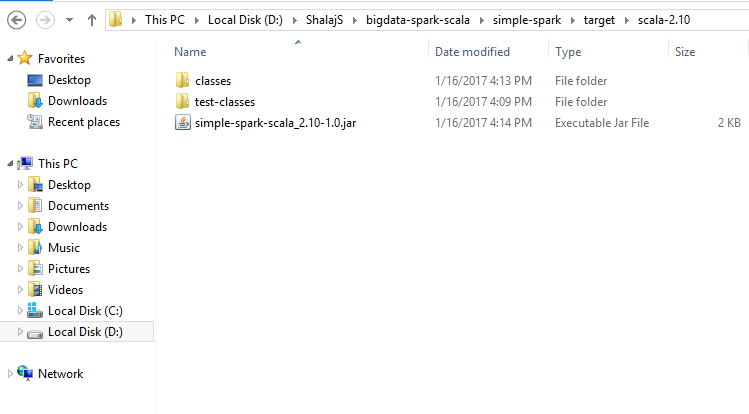


Here I am using 2.10 scala version to build the file as we need to build it on 2.10 level and java version is 1.7

Now run command ***sbt package*** to build the jar file



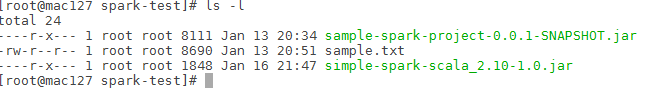
You will find jar file under target directory



Now you can ship that jar file to remote linux machine and run spark job

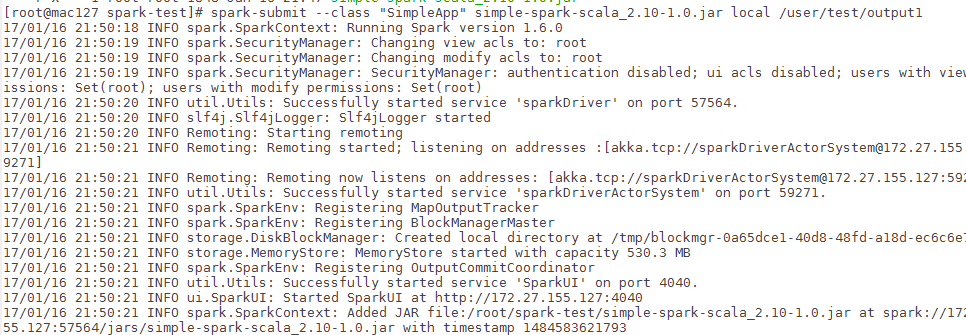


Check if file is copied to destination

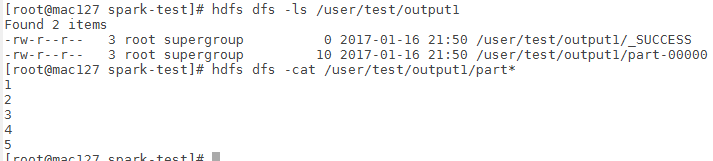


Now run below command to submit the job in local mode (command should be run on same directory where we placed jar file as we didn’t specify full path of jar file)

|  |
| --- |
| spark-submit --class "SimpleApp" simple-spark-scala\_2.10-1.0.jar local /user/test/output1 |

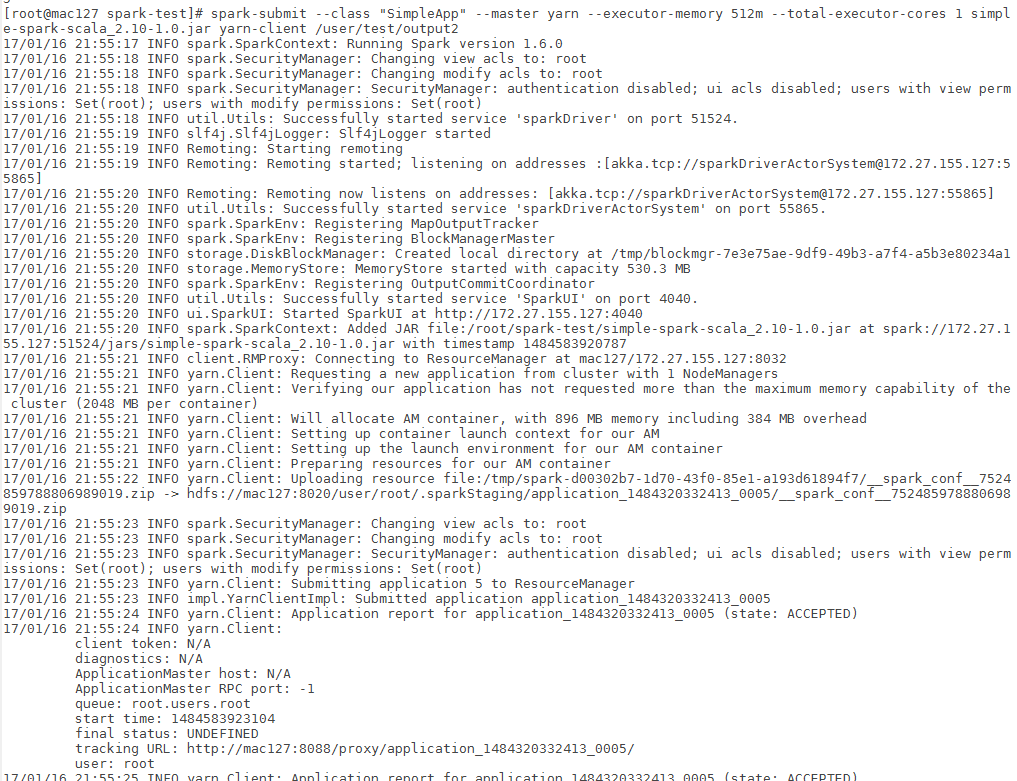


Check the output directory

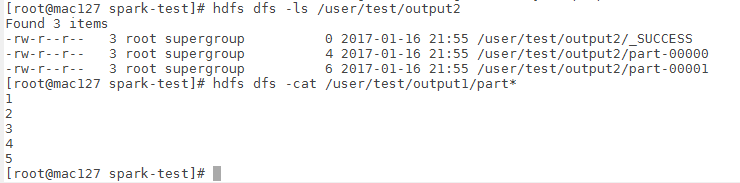


To run in yarn mode execute below command

|  |
| --- |
| spark-submit --class "SimpleApp" --master yarn --executor-memory 512m --total-executor-cores 1 simple-spark-scala\_2.10-1.0.jar yarn-client /user/test/output2 |



Now check the output directory



You can also see the history in resource manager webui

Just open RM webUI page http://<IP>:8088

