Session 25 – Big Data System Integration

Assignment 1

Task 1: Integrate Spark with Hive

On following the steps specified in the document:

Step 1:

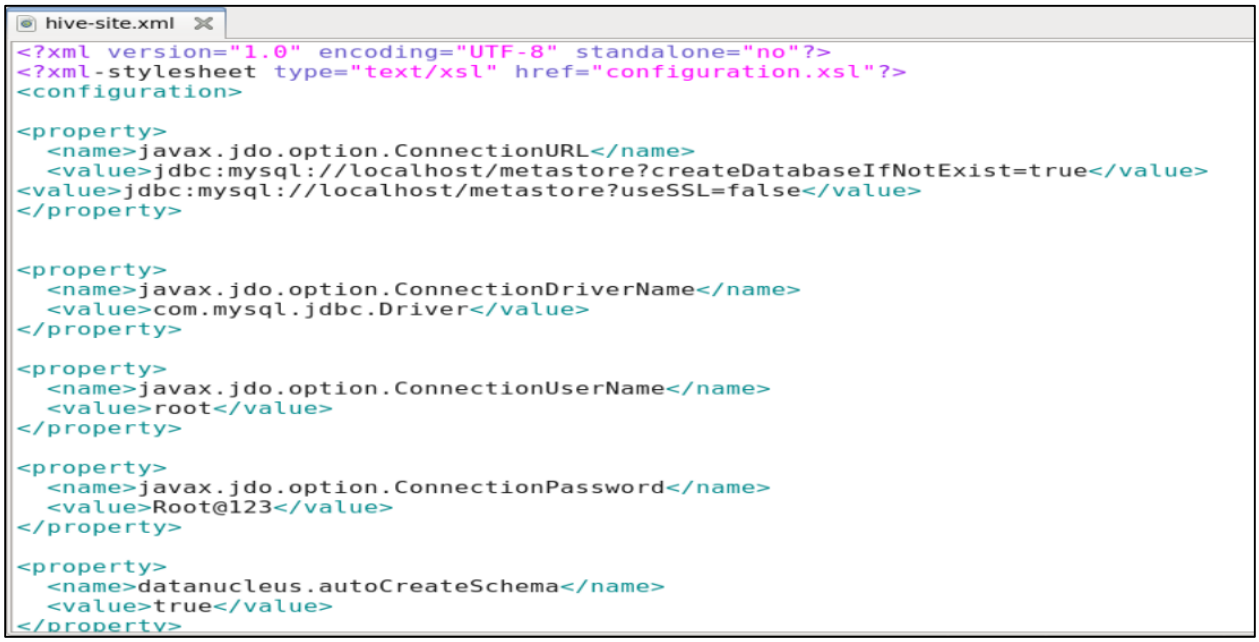
In reference with the above screenshot,

1: Copying hive-site.xml from hive home to spark home

2: hive-site.xml is in the spark home folder.

3: editing the hive-site xml to add the property related to hive metastore.

Step 2:



Editing the hive-stie.xml and adding the below property.

<property>

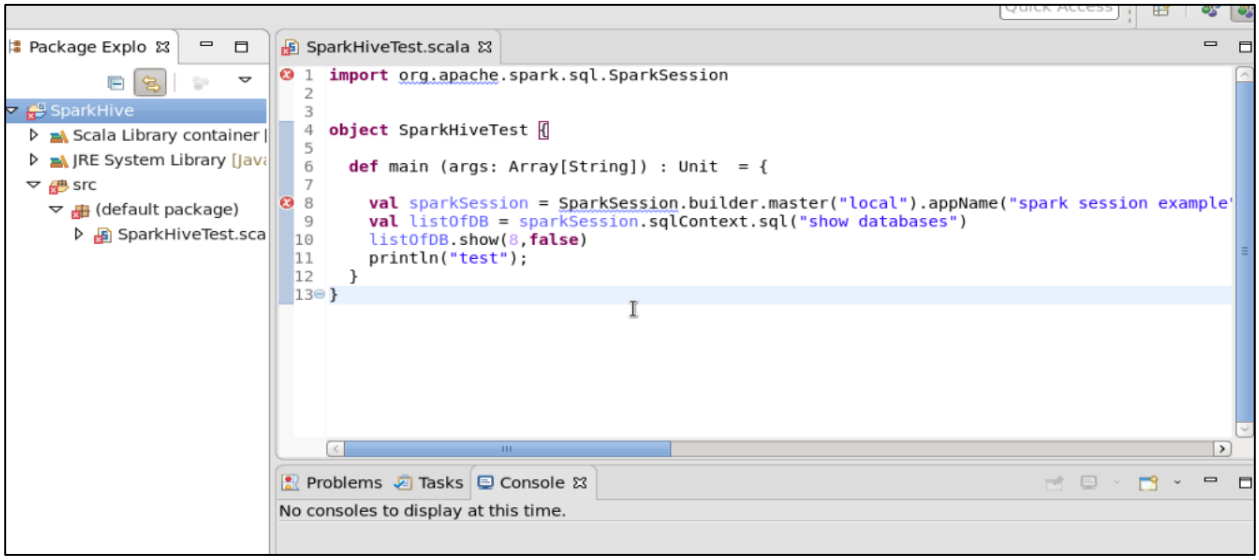
<name>hive.metastore.uris</name>

<value>thrift://localhost:9083</value>

<description>password for connecting to mysql server</description>

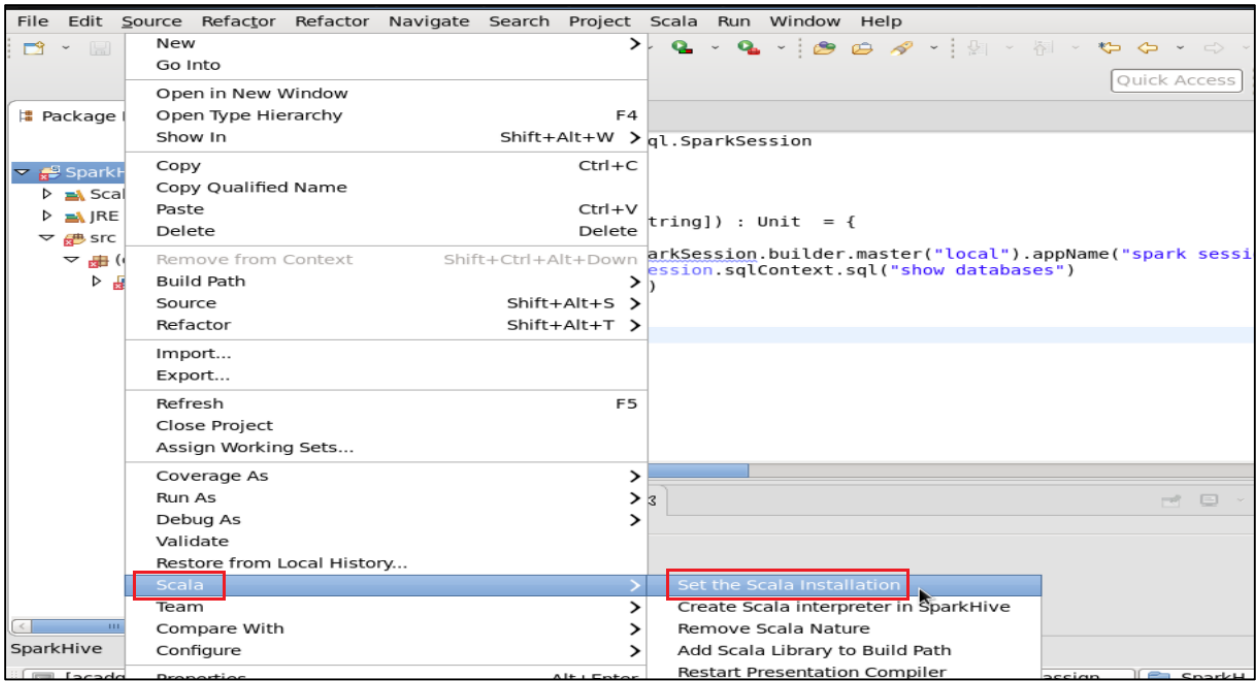
</property>

Step 3:

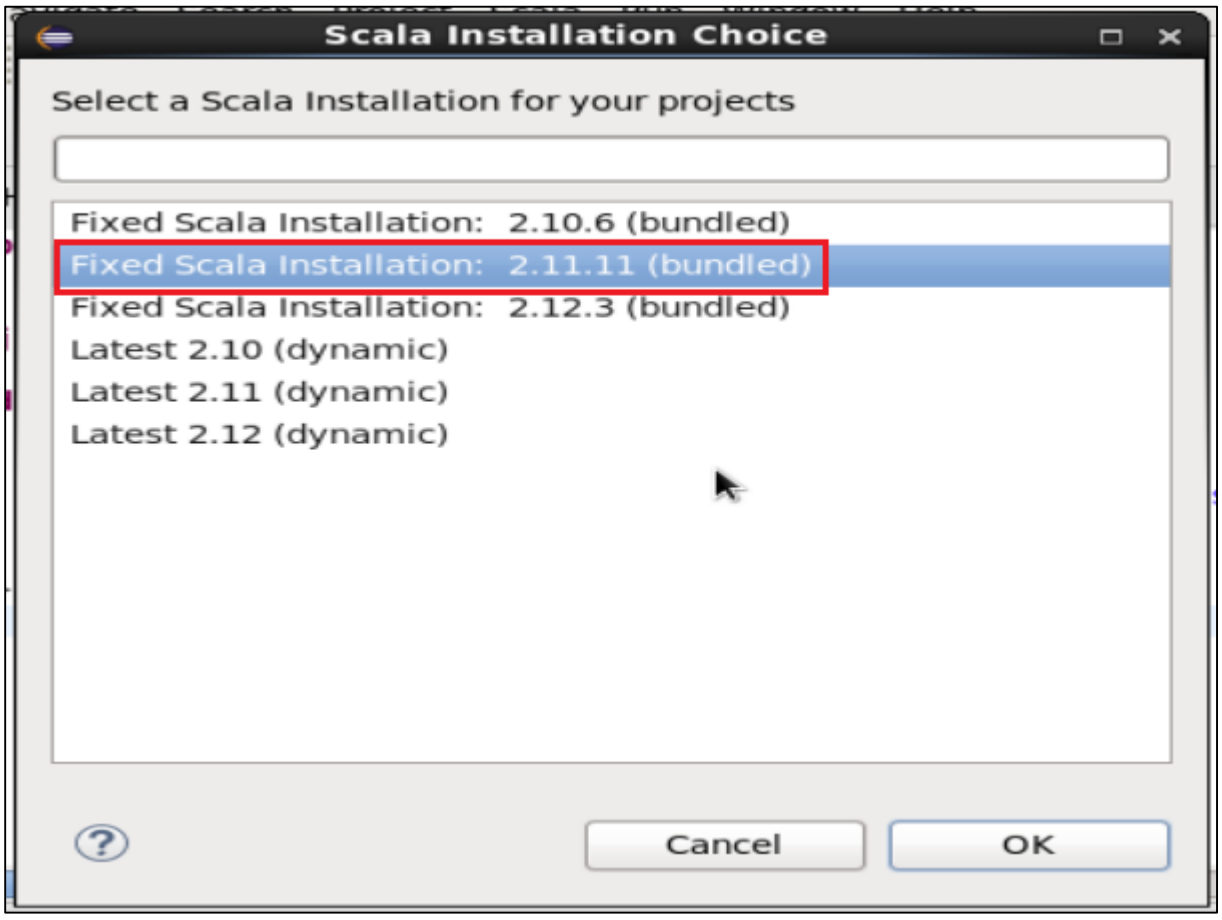


Step 4:

Adding the given code to Scala IDE. Since none of the required libraries are added, the project is with errors in the IDE.

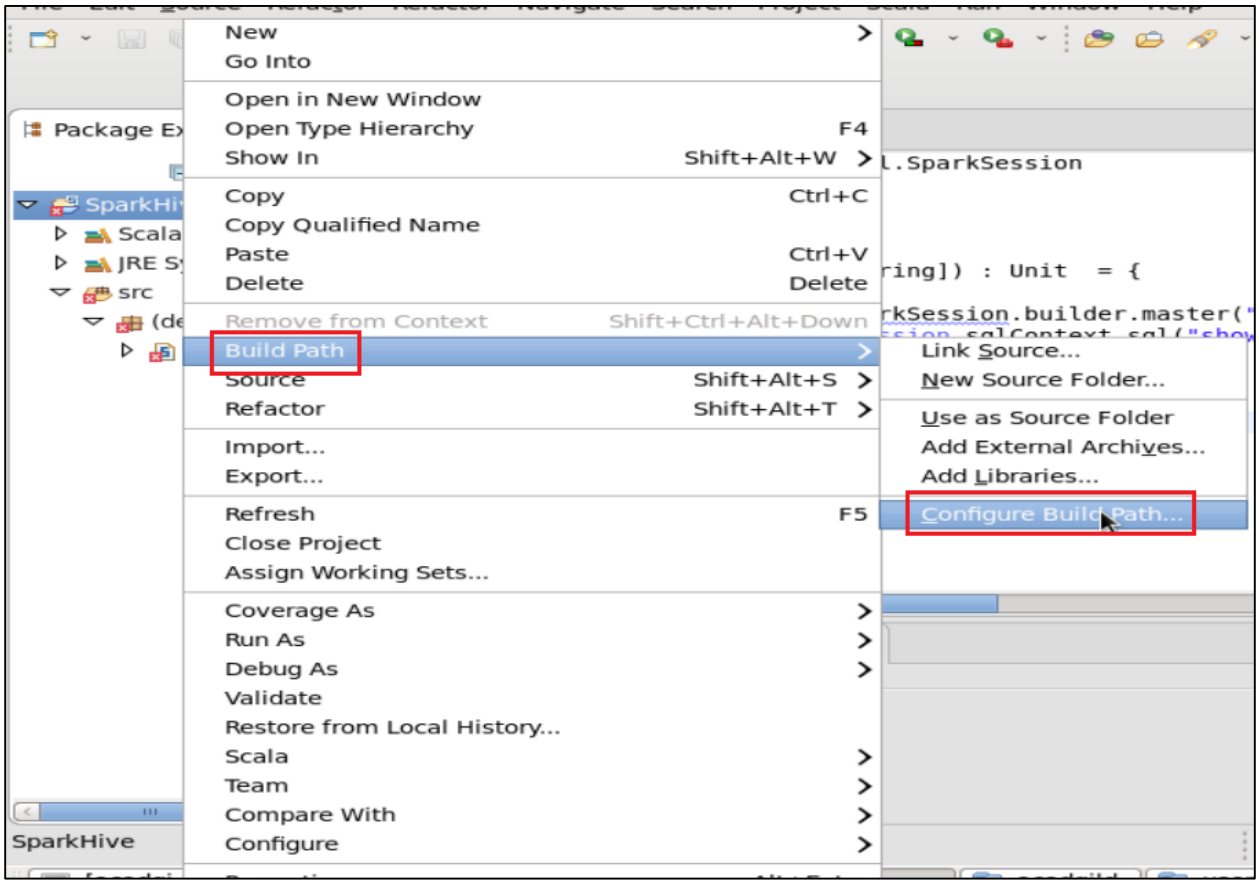


Right click on the Project in Package Explorer, go to Scala -> Set the Scala Intallation.



Select the specified Scala version (2.11) and click on OK.

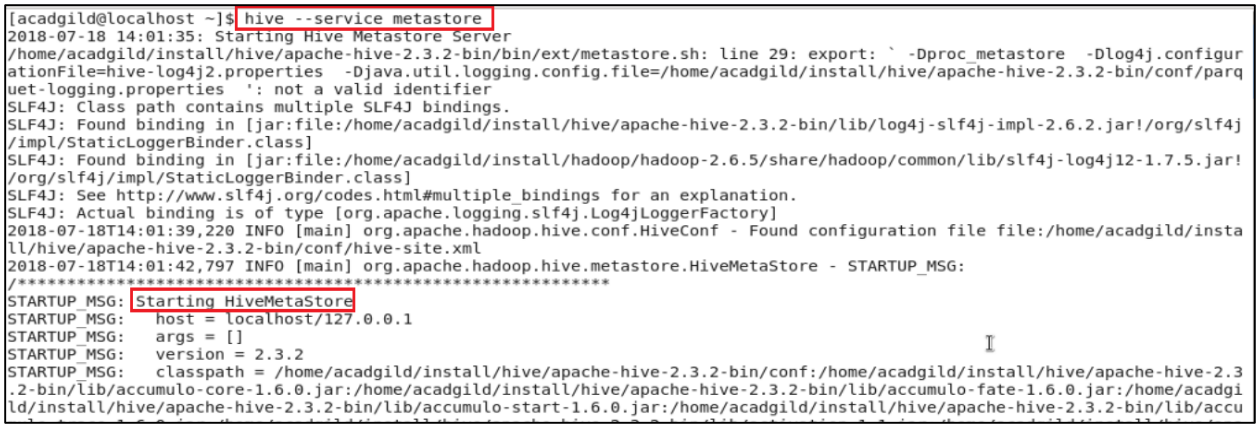
Step 5:

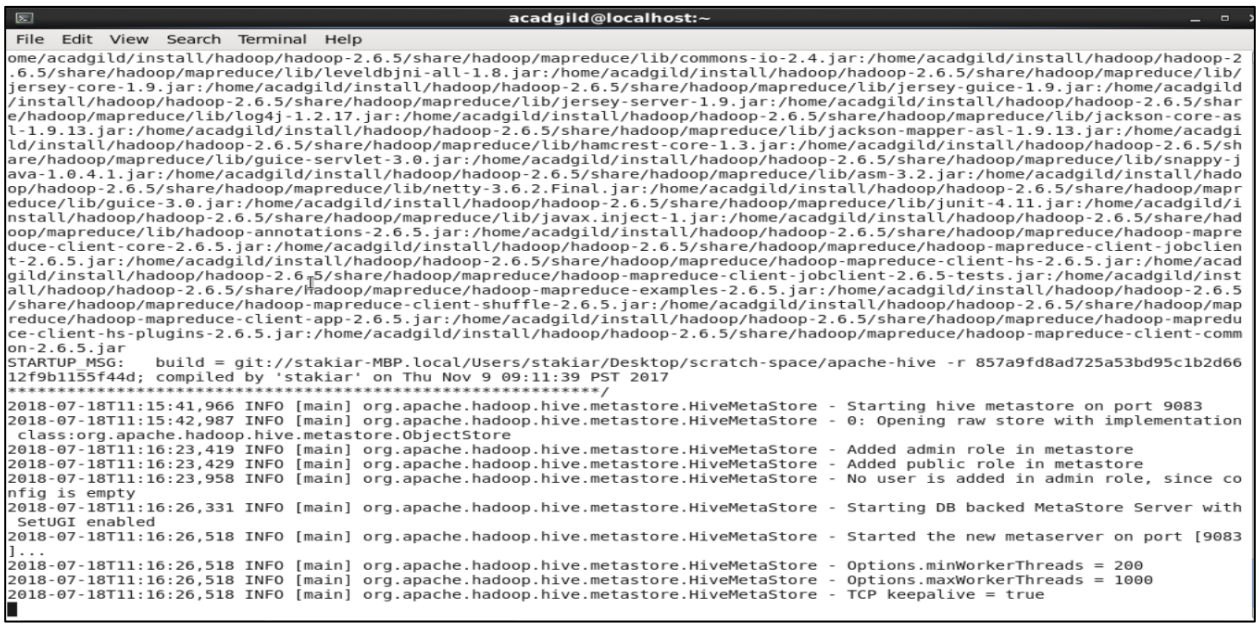


To add required jars, right click on the Project in Package Explorer, got to Build Path -> Configure Build Path…

Navigate to the jars location of Spark($SPARK_HOME/jars) and Hive($HIVE_HOME/lib) and select all the jars into the build path. Click on Apply and Close. The Project is built and all the errors are removed from the code.

Step 6:



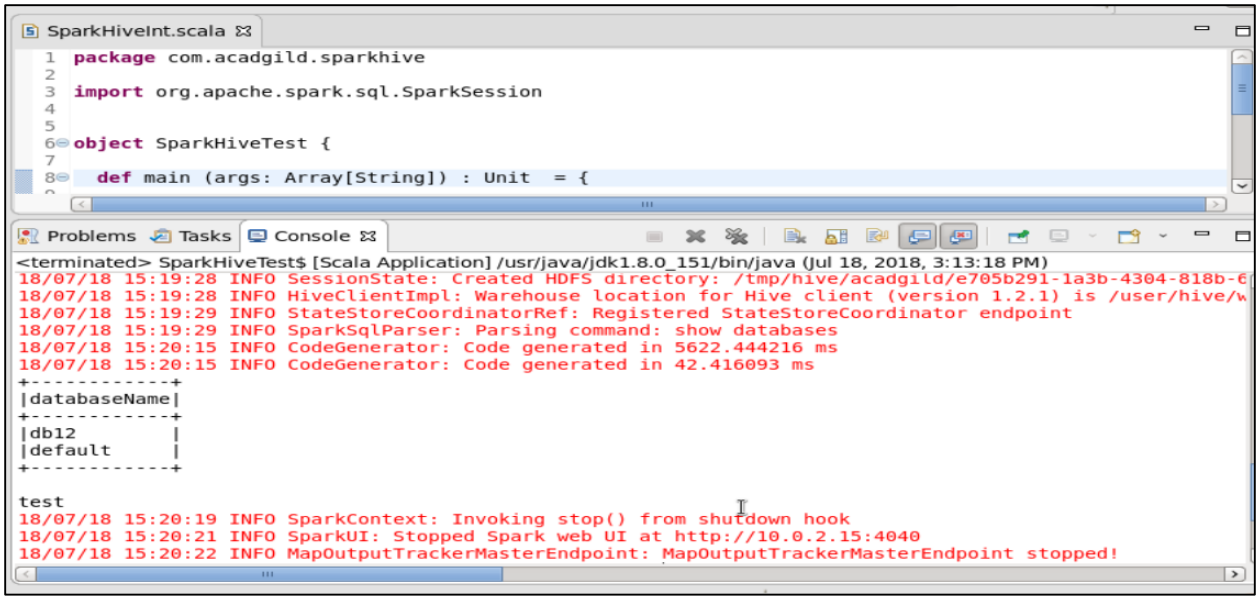


Hive Metastore is started using the following command.

hive --service metastore

As seen in the above screen shot, Hive metastore is successfully started.

Step 7:



On running the given script, we can see that the list of databases created in Hive are displayed.

Task 2: Integrate Spark with HBase

Following a similar procedure as with Hive, below steps are performed.

Step 1:

With respect to the screeshot above,

1: Ensuring that hadoop services are up.

2: Starting the hbase services.

Step 2:

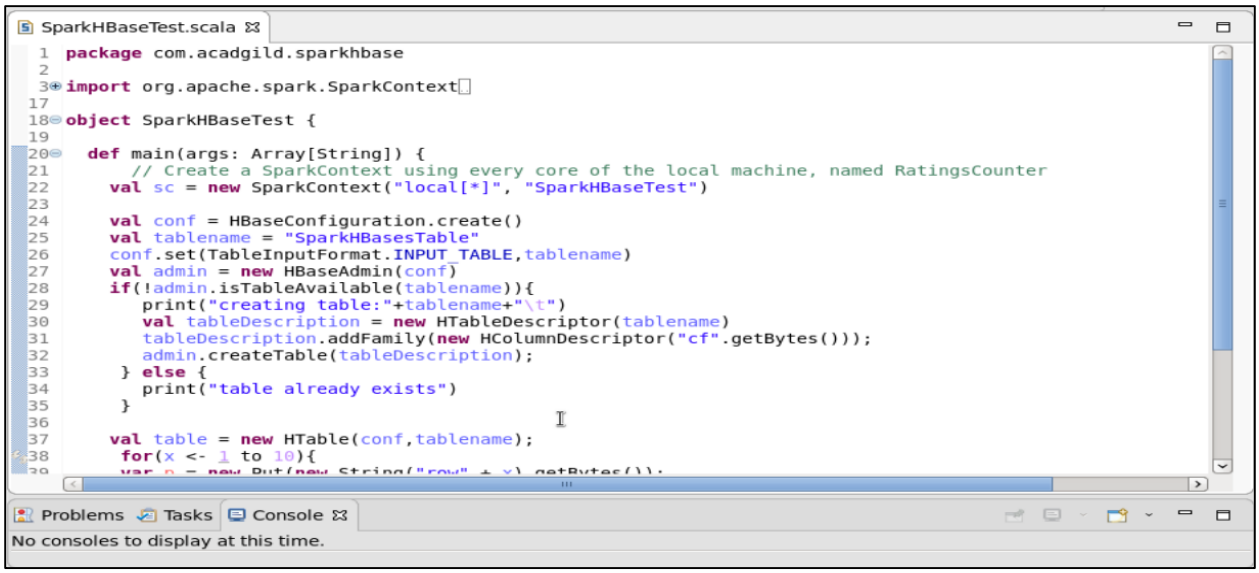
The given code is created as a project in Scala IDE.

Step 3:

Scala installation is set 2.11

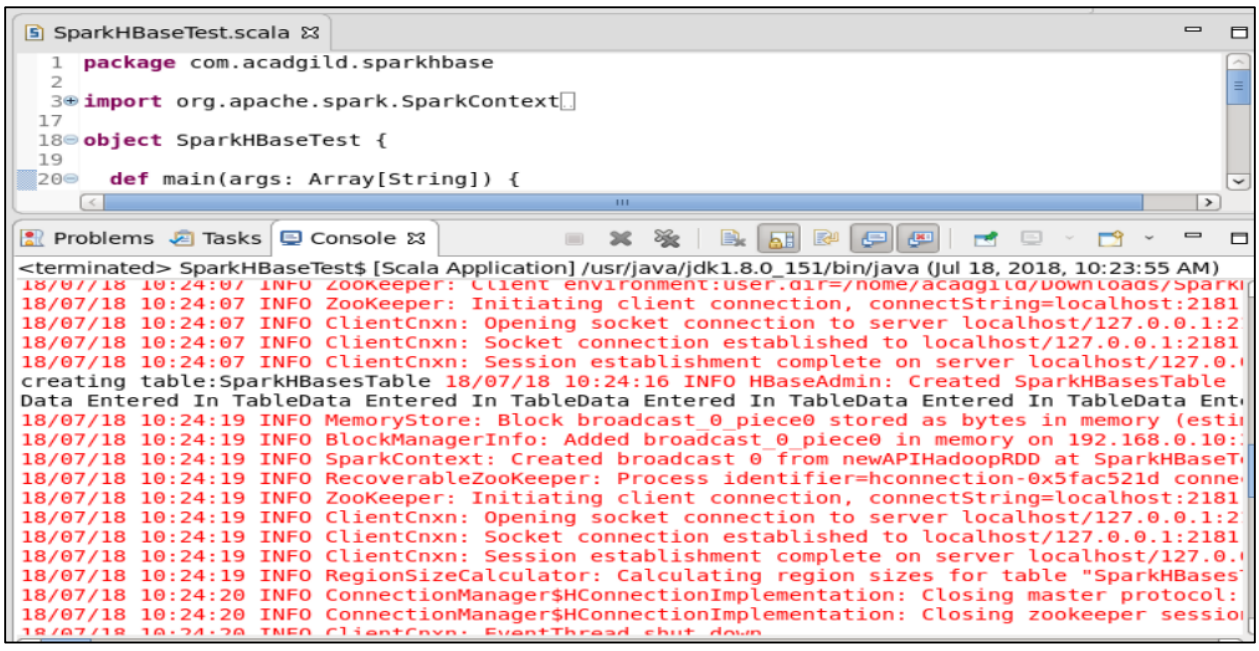
Step 4:

All the necessary jars related to Spark and Hbase are included in the Build Path.

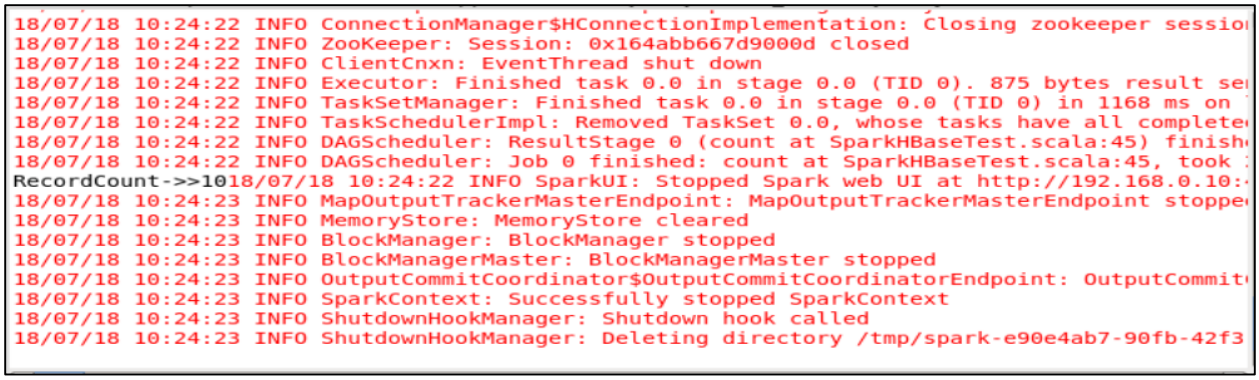


Step 5:

The table 'SparkHBasesTable' is created in hbase and data is inserted into the table. The same is specified in the log as shown below



The Record Count of the table is also displayed.


18/07/18 10:24:22 INFO ConnectionManager$HConnectionImplementation: Closing zookeeper sessio
18/07/18 10:24:22 INFO ZooKeeper: Session: 0x164abb667d9000d closed
18/07/18 10:24:22 INFO ClientCnxn: EventThread shut down
18/07/18 10:24:22 INFO Executor: Finished task 0.0 in stage 0.0 (TID 0). 875 bytes result se
18/07/18 10:24:22 INFO TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 1168 ms on
18/07/18 10:24:22 INFO TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all complete
18/07/18 10:24:22 INFO DAGScheduler: ResultStage 0 (count at SparkHBaseTest.scala:45) finish
18/07/18 10:24:22 INFO DAGScheduler: Job 0 finished: count at SparkHBaseTest.scala:45, took
RecordCount->>1018/07/18 10:24:22 INFO SparkUI: Stopped Spark web UI at http://192.168.0.10:
18/07/18 10:24:23 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stoppe
18/07/18 10:24:23 INFO MemoryStore: MemoryStore cleared
18/07/18 10:24:23 INFO BlockManager: BlockManager stopped
18/07/18 10:24:23 INFO BlockManagerMaster: BlockManagerMaster stopped
18/07/18 10:24:23 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommit
18/07/18 10:24:23 INFO SparkContext: Successfully stopped SparkContext
18/07/18 10:24:23 INFO ShutdownHookManager: Shutdown hook called
18/07/18 10:24:23 INFO ShutdownHookManager: Deleting directory /tmp/spark-e90e4ab7-90fb-42f3