

## Assignment 25:

-----

**Give your own input in output-screenshot of report.**

**Task 1 As discussed in class integrate Spark Hive Task**

**Steps to integrate Spark hive:**

-----

2. Copy the hive-site.xml file from \$HIVE\_HOME/conf to \$SPARK\_HOME/conf
3. Add the following properties to hive-site.xml on spark side :
  - <property>
  - <name>hive.metastore.uris</name>
  - <value>thrift://localhost:9083</value>
  - <description>password for connecting to mysql server</description>
  - </property>
4. Write the code in Scala IDE to list the Databases in the hive. Source code is uploaded separately.
5. Make sure that your hadoop is started
6. Start hive metastore by executing hive —service metastore command.
7. Run the code from the IDE. You should be able to see all the hive databases.

**Output Screen Shot:**

-----

**Starting metastore:**

```
STARTUP_MSG: build = git://stakiar-MBP.local/Users/stakiar/Desktop/scratch-space/apache-hive -r 857a9fd8ad725a53bd95c1b2d6612f9
b1155f44d; compiled by 'stakiar' on Thu Nov 9 09:11:39 PST 2017
*****/
2018-06-07T18:16:45,098 INFO [main] org.apache.hadoop.hive.metastore.HiveMetaStore - Starting hive metastore on port 9083
2018-06-07T18:16:45,853 INFO [main] org.apache.hadoop.hive.metastore.HiveMetaStore - 0: Opening raw store with implementation cla
ss:org.apache.hadoop.hive.metastore.ObjectStore
2018-06-07T18:16:53,806 INFO [main] org.apache.hadoop.hive.metastore.HiveMetaStore - Added admin role in metastore
2018-06-07T18:16:53,812 INFO [main] org.apache.hadoop.hive.metastore.HiveMetaStore - Added public role in metastore
2018-06-07T18:16:53,852 INFO [main] org.apache.hadoop.hive.metastore.HiveMetaStore - No user is added in admin role, since config
is empty
2018-06-07T18:16:54,184 INFO [main] org.apache.hadoop.hive.metastore.HiveMetaStore - Starting DB backed MetaStore Server with Set
ting UGI enabled
2018-06-07T18:16:54,219 INFO [main] org.apache.hadoop.hive.metastore.HiveMetaStore - Started the new metaserver on port [9083]...
2018-06-07T18:16:54,219 INFO [main] org.apache.hadoop.hive.metastore.HiveMetaStore - Options.minWorkerThreads = 200
2018-06-07T18:16:54,219 INFO [main] org.apache.hadoop.hive.metastore.HiveMetaStore - Options.maxWorkerThreads = 1000
2018-06-07T18:16:54,220 INFO [main] org.apache.hadoop.hive.metastore.HiveMetaStore - TCP keepalive = true
```

## Hive database in the terminal:

```
hive> show databases;
OK
custom
default
Time taken: 0.04 seconds, Fetched: 2 row(s)
hive> █
```

## Output after running the code in Scala Ide

```
-----
4 object SparkHiveTest {
5
6 def main (args: Array[String]) : Unit = {
7
8     val sparkSession = SparkSession.builder.master("local").appName("Assig
9     val listOfDB = sparkSession.sqlContext.sql("show databases")
10    listOfDB.show(8, false)
11    println("test");
12 }
13 }

Problems Tasks Console
<terminated> SparkHiveTest$ [Scala Application] /usr/java/jdk1.8.0_151/bin/java (Jun 7, 2018, 6:18:48 PM)
18/06/07 18:19:28 INFO CodeGenerator: Code generated in 22.499053 ms
+-----+
|databaseName|
+-----+
|custom      |
|default     |
+-----+

test
18/06/07 18:19:28 INFO CodeGenerator: Code generated in 22.499053 ms
```

## 2. As discussed in class integrate Spark Hbase Task

### Steps:

- 
1. Write an API code in scala ide to create a new table in hbase. Source code is uploaded separately.
  2. Run the code in scala ide and check the hbase for the newly created table.
  3. Make sure to start the hbase shell using the below commands
- Start-hbase.sh  
Hbase shell

## Output Screenshots:

### List of tables before running the code:

```
hbase(main):001:0> list
TABLE
SparkHBasesTable
TRANSACTIONS
bulktable
clicks
4 row(s) in 0.3620 seconds

=> ["SparkHBasesTable", "TRANSACTIONS", "bulktable", "clicks"]
```

### List of tables after running the code:

#### Newly created table highlighted in red

```
hbase(main):001:0> list
TABLE
SparkHBasesTable
TRANSACTIONS
bulktable
clicks
4 row(s) in 0.3620 seconds

=> ["SparkHBasesTable", "TRANSACTIONS", "bulktable", "clicks"]
hbase(main):002:0> list
TABLE
SparkHBasesTable
SparkHBasesTable1
TRANSACTIONS
bulktable
clicks
5 row(s) in 0.0160 seconds

=> ["SparkHBasesTable", "SparkHBasesTable1", "TRANSACTIONS", "bulktable", "clicks"]
```

### Console output:

```
18/06/07 18:38:57 INFO HBaseAdmin: Created SparkHBasesTable1
creating table:SparkHBasesTable1
Data Entered In TableData Entered In TableData Entered In TableData Entered In TableData Entered In TableData
```

## Newly created table with column family, column and value

```
hbase(main):004:0> scan 'SparkHBasesTable1'
ROW          COLUMN+CELL
row1         column=cf:column, timestamp=1528376937645, value=value1
row10        column=cf:column, timestamp=1528376937716, value=value10
row2         column=cf:column, timestamp=1528376937673, value=value2
row3         column=cf:column, timestamp=1528376937679, value=value3
row4         column=cf:column, timestamp=1528376937683, value=value4
row5         column=cf:column, timestamp=1528376937688, value=value5
row6         column=cf:column, timestamp=1528376937694, value=value6
row7         column=cf:column, timestamp=1528376937701, value=value7
row8         column=cf:column, timestamp=1528376937705, value=value8
row9         column=cf:column, timestamp=1528376937711, value=value9
10 row(s) in 0.4000 seconds
```

## 3 As discussed in class integrate Spark Kafka