

SPARK framework introduction

Go to the Bigdata User login

Pwd - hadoop

Now search for Oracle virtual box

Double Click Hadoop

Go to login,

Login - Ubuntu

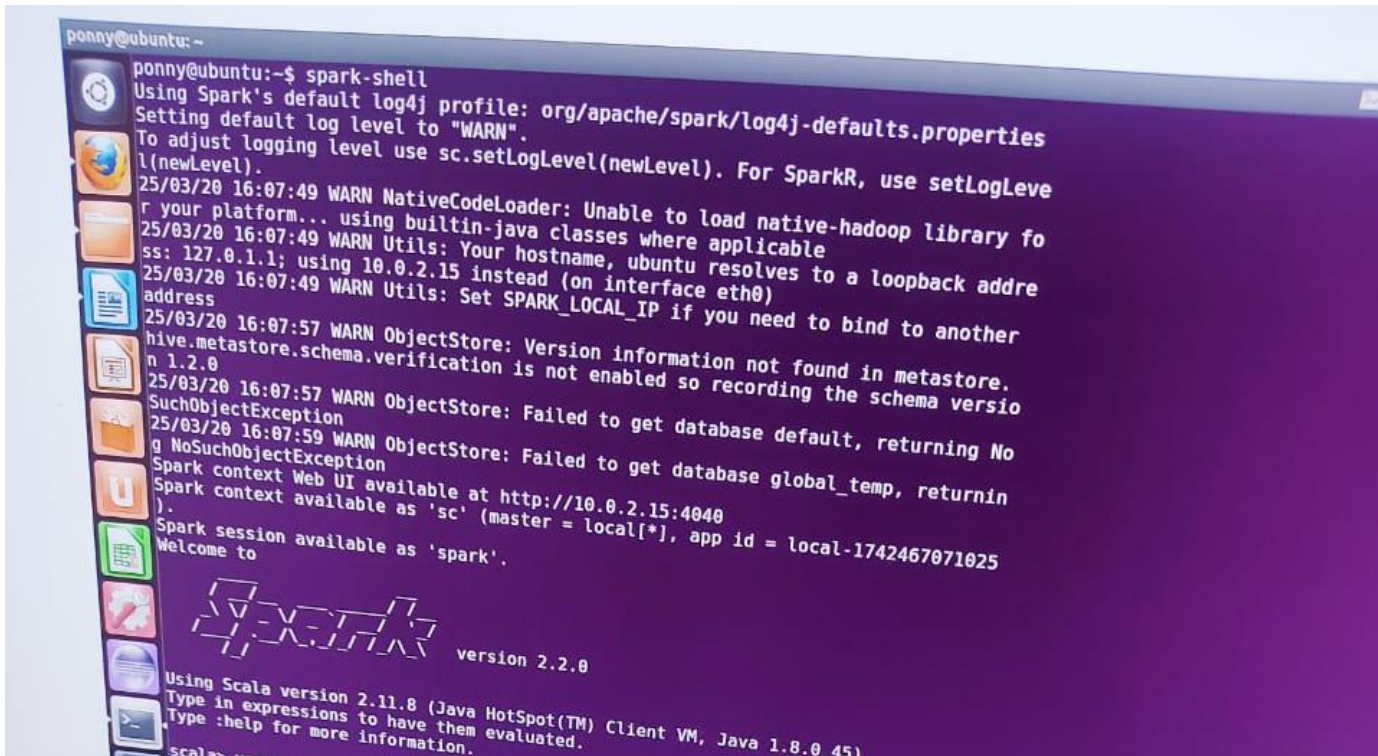
Passwd - vitcc123

Experiment 2: User Defined Function using Scala in spark framework

Open terminal

Type the command: spark-shell

1. Follow the commands given in the screenshot --



```
ponny@ubuntu:~$ spark-shell
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/03/20 16:07:49 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
25/03/20 16:07:49 WARN Utils: Your hostname, ubuntu resolves to a loopback address: 127.0.0.1; using 10.0.2.15 instead (on interface eth0)
25/03/20 16:07:49 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
25/03/20 16:07:57 WARN ObjectStore: Version information not found in metastore. hive.metastore.schema.validation is not enabled so recording the schema version 1.2.0
25/03/20 16:07:57 WARN ObjectStore: Failed to get database default, returning NoSuchObjectException
25/03/20 16:07:59 WARN ObjectStore: Failed to get database global_temp, returning NoSuchObjectException
Spark context Web UI available at http://10.0.2.15:4040
Spark context available as 'sc' (master = local[*], app id = local-1742467071025)
Welcome to

      _ _ _ _ _
     / _ _ _ _ \   version 2.2.0
    / _ _ _ _ \
   / _ _ _ _ \
  / _ _ _ _ \
 / _ _ _ _ \
/_ _ _ _ _ \

Using Scala version 2.11.8 (Java HotSpot(TM) Client VM, Java 1.8.0_45)
Type in expressions to have them evaluated.
Type :help for more information.
scala>
```

2.

```
scala> import spark.implicits._
import spark.implicits._

scala> val cols = Seq("sno","name")
<console>:26: error: not found: value Seq
    val cols = Seq("sno","name")
                ^

scala> val cols = Seq("sno","name")
cols: Seq[String] = List(sno, name)

scala> val data = Seq(("1", "gowtham"),
  | ("2", "nandini"),
  | ("3", "saravana")
  | )
data: Seq[(String, String)] = List((1,gowtham), (2,nandini), (3,saravana))

scala> val df = data.toDF(cols:_)
df: org.apache.spark.sql.DataFrame = [sno: string, name: string]

scala> df.show(false)
+----+-----+
|sno|name|
+----+-----+
|1|gowtham|
|2|nandini|
|3|saravana|
+----+-----+
```

3.

```
scala> df.show(false)
+----+-----+
|sno|name|
+----+-----+
|1|gowtham|
|2|nandini|
|3|saravana|
+----+-----+

scala> val customUDF = udf(Ucase)
<console>:26: error: not found: value Ucase
    val customUDF = udf(Ucase)
                        ^

scala> val Ucase = (strQuote:String) => {
  | val dt = strQuote.split(" ")
  | dt.map(f=> f.substring(0,1).toUpperCase + f.substring(1,f.length)).mkString(" ")
  | }
Ucase: String => String = <function1>

scala> val customUDF = udf(Ucase)
customUDF: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function1>,StringType,Some(List(StringType)))

scala> df.select(col("sno"), customUDF(col("name")).as("name")).show(false)
+----+-----+
|sno|name|
+----+-----+
|1|Gowtham|
|2|Nandini|
|3|Saravana|
+----+-----+

scala>
```