Assisted Practice 20: GraphX

Problem Scenario: Create a graph object to calculate the distance between different cities using GraphX

Objective: In this demonstration, you will solve a real-world problem by calculating the distance between the cities.

Tasks to Perform:

1. Open the Spark shell in **"Webconsole"** and import packages
2. Upload the **"vertices"** and **"edges"** data by specifying the path
3. Create a graph object from the vertices and edges array to calculate the distance between the cities and display the output

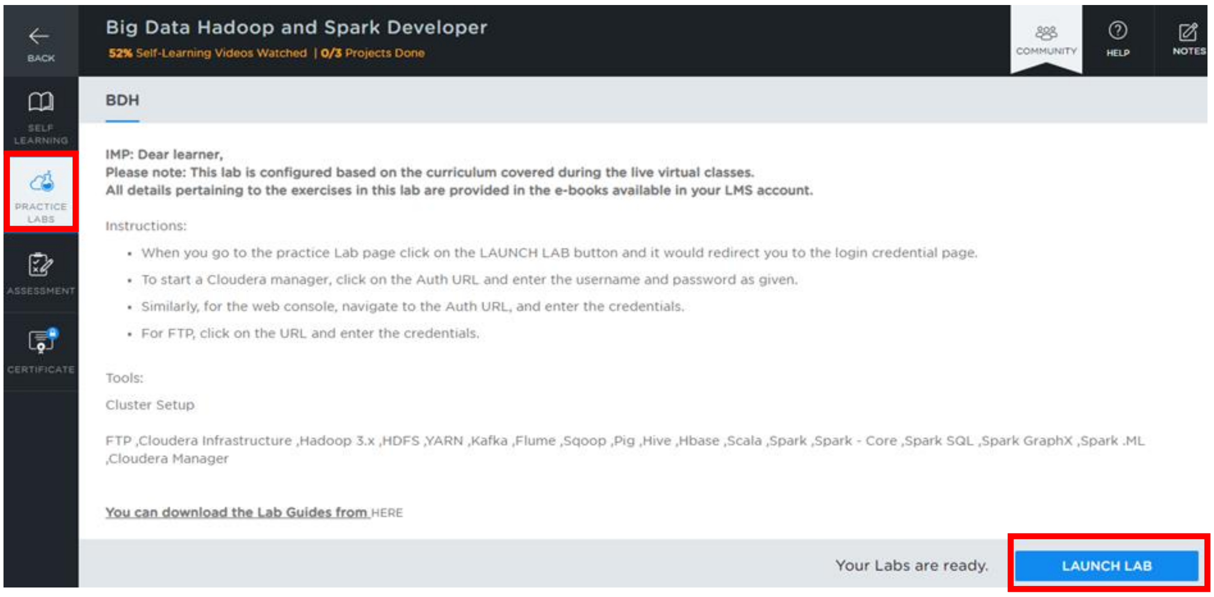
Steps to Perform:

Step 1: Download the text files with the names **"vertices"** and **"edges"** from the course resources section

Step 2: Log in to your LMS account

Step 3: Open the course **"Big Data Hadoop and Spark developer"**

Step 4: On the left side, click on the **"PRACTICE LABS"** tab and on the **"LAUNCH LAB"** button



Big Data Hadoop and Spark Developer

52% Self-Learning Videos Watched | 0/3 Projects Done

IMP: Dear learner,
Please note: This lab is configured based on the curriculum covered during the live virtual classes.
All details pertaining to the exercises in this lab are provided in the e-books available in your LMS account.

Instructions:

- When you go to the practice Lab page click on the LAUNCH LAB button and it would redirect you to the login credential page.
- To start a Cloudera manager, click on the Auth URL and enter the username and password as given.
- Similarly, for the web console, navigate to the Auth URL, and enter the credentials.
- For FTP, click on the URL and enter the credentials.

Tools:

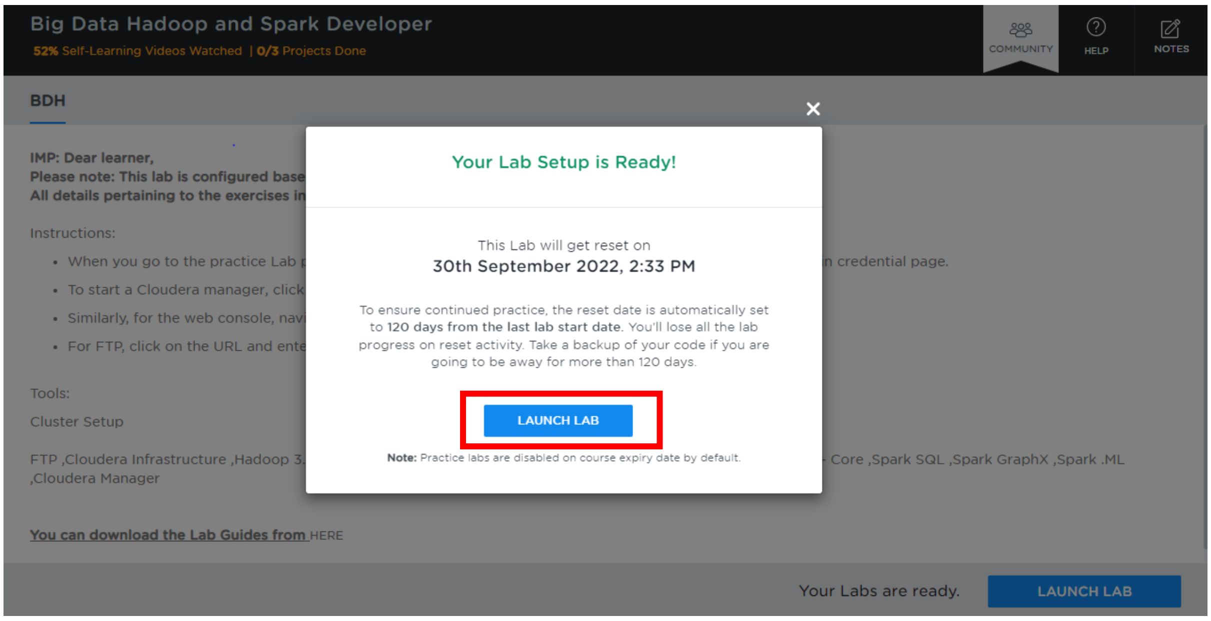
Cluster Setup

FTP ,Cloudera Infrastructure ,Hadoop 3.x ,HDFS ,YARN ,Kafka ,Flume ,Sqoop ,Pig ,Hive ,Hbase ,Scala ,Spark ,Spark - Core ,Spark SQL ,Spark GraphX ,Spark .ML ,Cloudera Manager

You can download the Lab Guides from HERE

Your Labs are ready.

Step 5: Again, click on the **“LAUNCH LAB”** button



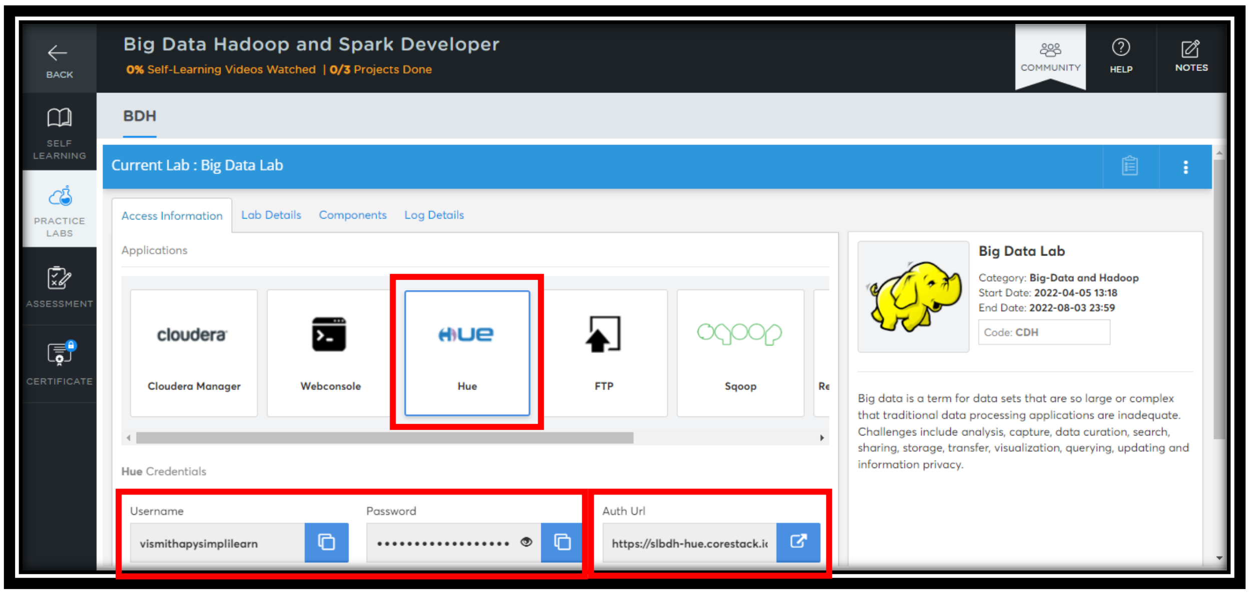
Your Lab Setup is Ready!

This Lab will get reset on
30th September 2022, 2:33 PM

To ensure continued practice, the reset date is automatically set to 120 days from the last lab start date. You'll lose all the lab progress on reset activity. Take a backup of your code if you are going to be away for more than 120 days.

Note: Practice labs are disabled on course expiry date by default.

Step 6: Click on **“Hue”** and click on the **“Auth Url”** to upload the files and copy the **“Username”** and the **“Password”** provided to log in to the **“Hue”**



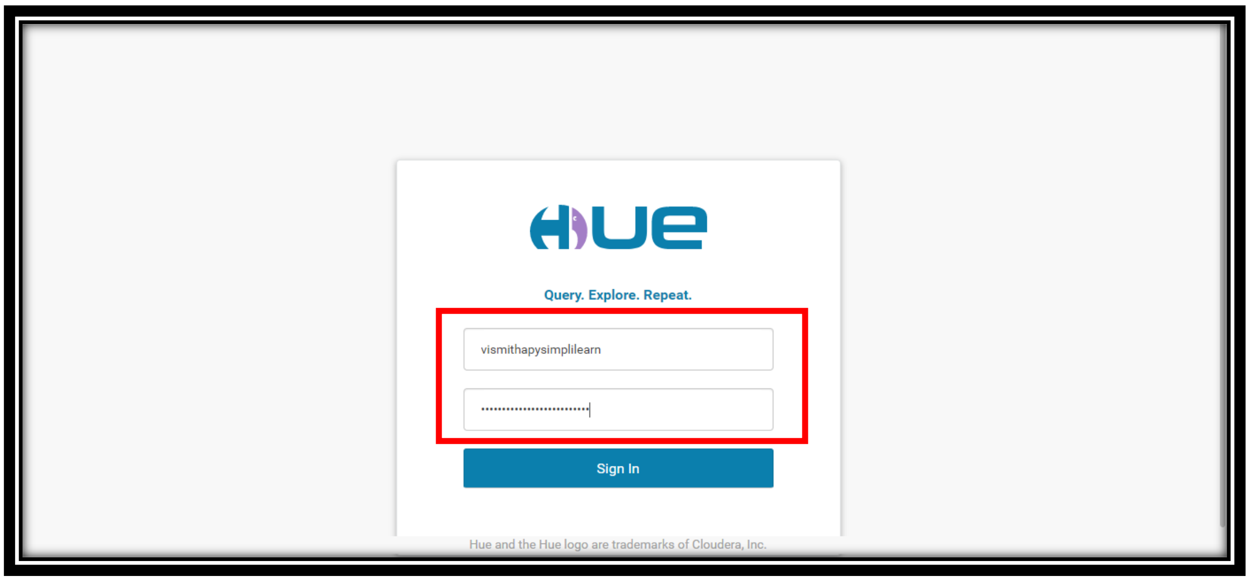
Big Data Lab

Category: Big-Data and Hadoop
Start Date: 2022-04-05 13:18
End Date: 2022-08-03 23:59
Code: CDH

Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate. Challenges include analysis, capture, data curation, search, sharing, storage, transfer, visualization, querying, updating and information privacy.

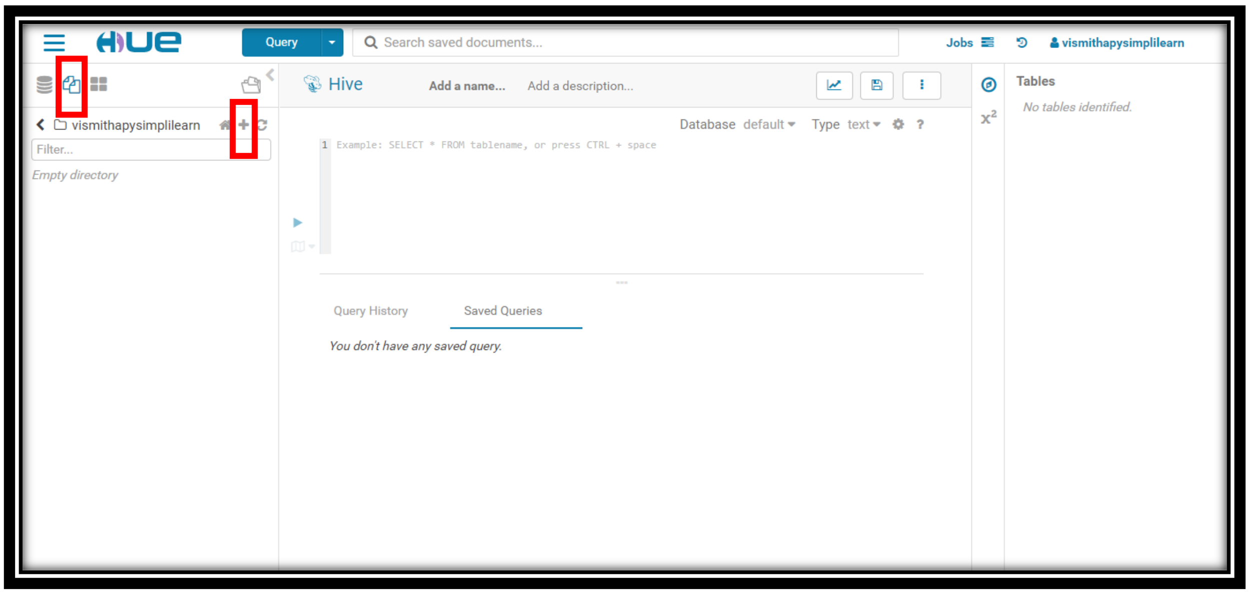
Username: vismithapysimplilearn
Password: ••••••••••••••••••
Auth Url: https://slbdh-hue.corestack.i

Step 7: Paste the "**Username**" and the "**Password**" on the login window and click on Sign In

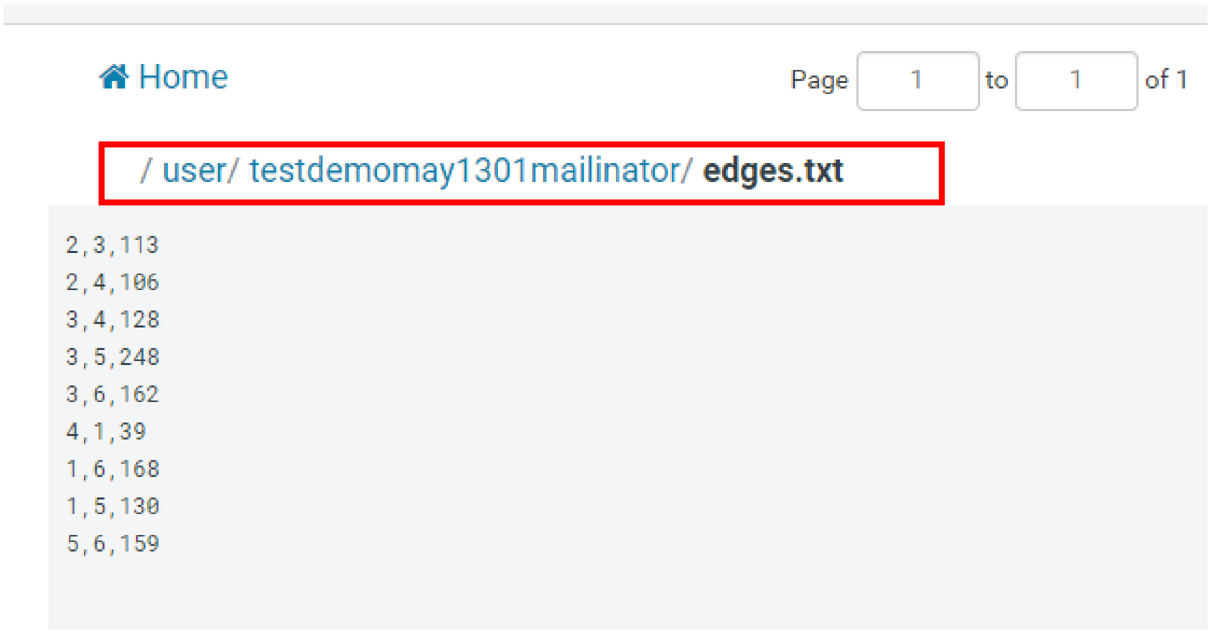


vismithapysimplilearn
••••••••••••••••••••••••••

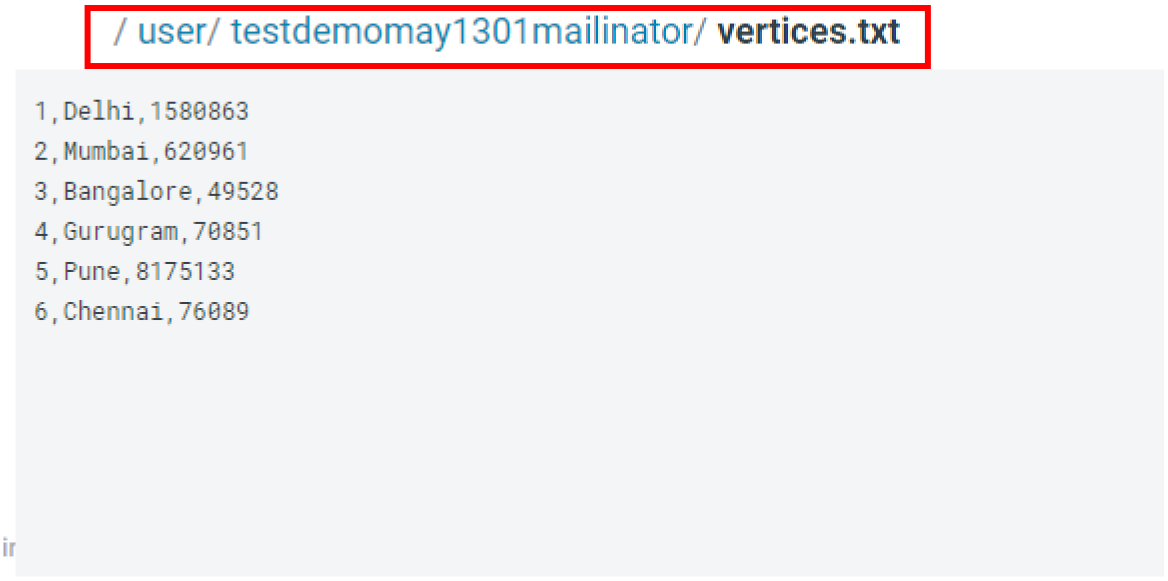
Step 8: Click on the **"HDFS"** icon and on the **"+"** symbol to upload the dataset



Step 9: Select the downloaded text files and upload it to **"HDFS."** In addition, by right-clicking, copy the paths from the text files that has been uploaded



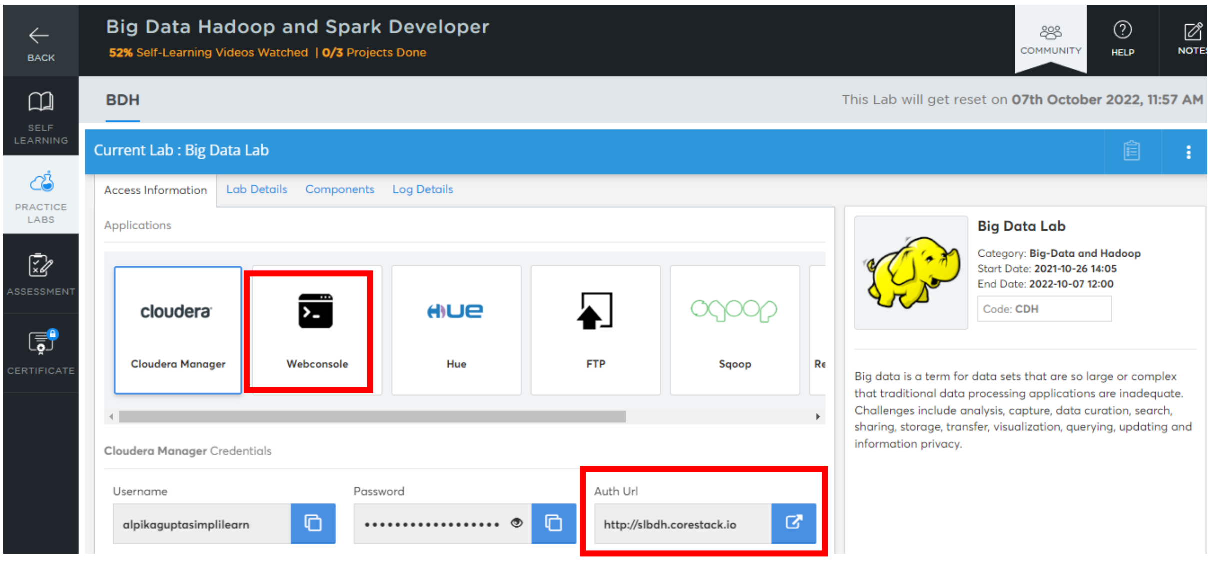
```
2,3,113
2,4,106
3,4,128
3,5,248
3,6,162
4,1,39
1,6,168
1,5,130
5,6,159
```



/ user/ testdemomay1301mailinator/ **vertices.txt**

```
1,Delhi,1580863
2,Mumbai,620961
3,Bangalore,49528
4,Gurugram,70851
5,Pune,8175133
6,Chennai,76089
```

Step 10: Go back to the lab window and click on the "**Webconsole**" and click on the **"Auth Url"**



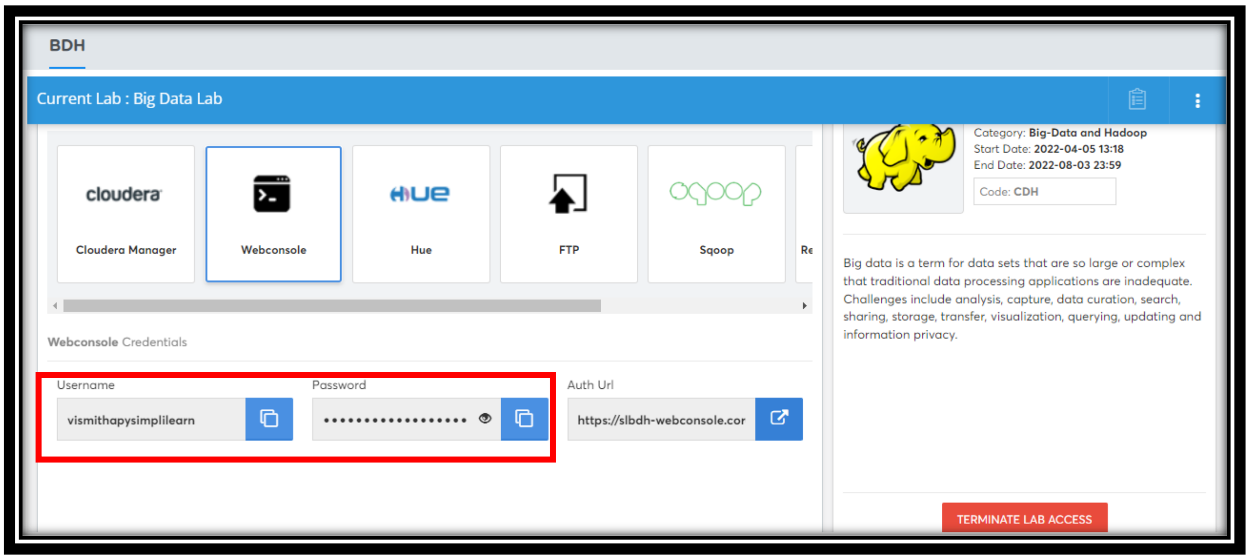
BACK
SELF LEARNING
PRACTICE LABS
ASSESSMENT
CERTIFICATE
Big Data Hadoop and Spark Developer
52% Self-Learning Videos Watched | 0/3 Projects Done
COMMUNITY
HELP
NOTES
BDH
This Lab will get reset on 07th October 2022, 11:57 AM
Current Lab : Big Data Lab
Access Information
Lab Details
Components
Log Details
Applications
Cloudera Manager
Webconsole
Hue
FTP
Sqoop
Cloudera Manager Credentials

Username: alpikaguptasimplilearn
Password: ••••••••••••••••••
Auth Url: http://slbdh.corestack.io

Big Data Lab
Category: Big-Data and Hadoop
Start Date: 2021-10-26 14:05
End Date: 2022-10-07 12:00
Code: CDH

Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate. Challenges include analysis, capture, data curation, search, sharing, storage, transfer, visualization, querying, updating and information privacy.

Step 11: Copy the "**Username**" and the "**Password**" provided to log in to the **"Webconsole"**


BDH
Current Lab : Big Data Lab
Cloudera Manager
Webconsole
Hue
FTP
Sqoop
Category: Big-Data and Hadoop
Start Date: 2022-04-05 13:18
End Date: 2022-08-03 23:59
Code: CDH
Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate. Challenges include analysis, capture, data curation, search, sharing, storage, transfer, visualization, querying, updating and information privacy.
Webconsole Credentials
Username
vismithapysimplilearn
Password
Auth Url
https://slbdh-webconsole.cor
TERMINATE LAB ACCESS

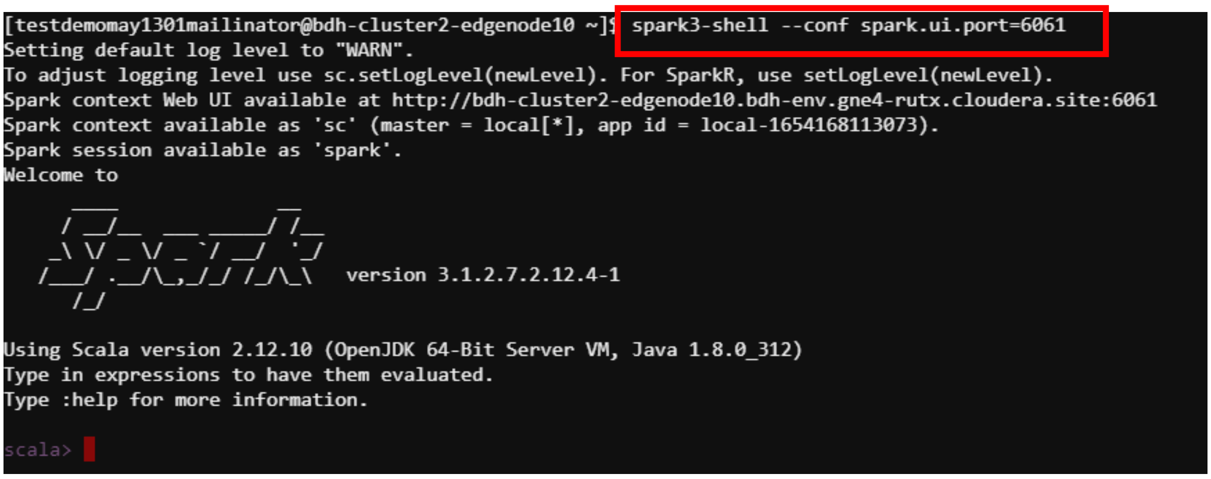
Step 12: Paste the "**Username**" and the "**Password**" on the console and click on Enter

Note: The password will not be visible when pasted on the console.

Step 13: Enter the below command to open the **"spark-shell"**:

Command:

spark3-shell –conf spark.ui.port=6061

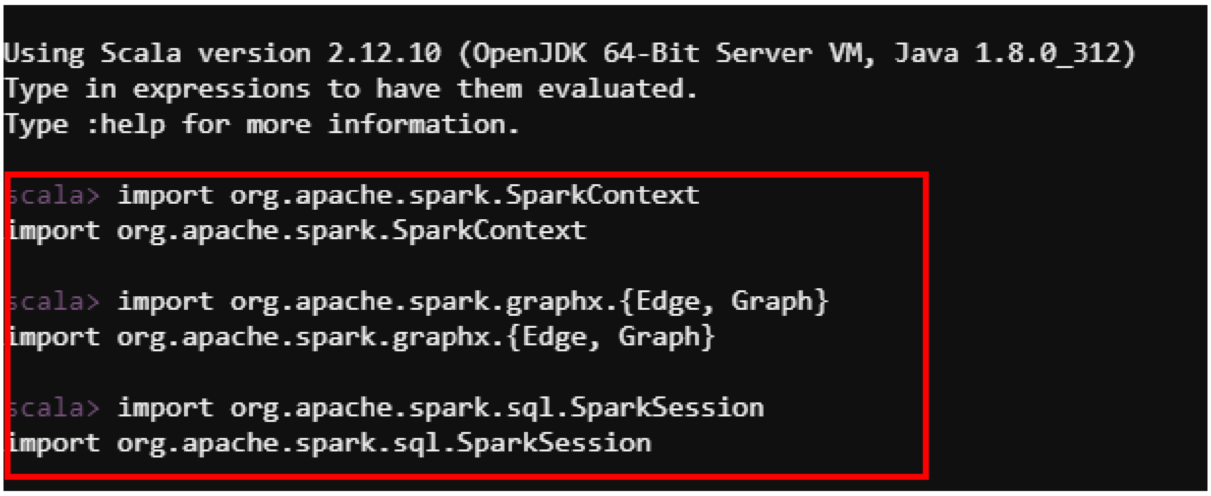


```
[testdemomay1301mailinator@bdh-cluster2-edgenode10 ~]$ spark3-shell --conf spark.ui.port=6061
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://bdh-cluster2-edgenode10.bdh-env.gne4-rutx.cloudera.site:6061
Spark context available as 'sc' (master = local[*], app id = local-1654168113073).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.2.7.2.12.4-1
      /_/

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_312)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

Step 14: Import the required packages

Command:

```
import org.apache.spark.SparkContext

import org.apache.spark.graphx.{Edge, Graph}

import org.apache.spark.sql.SparkSession
```



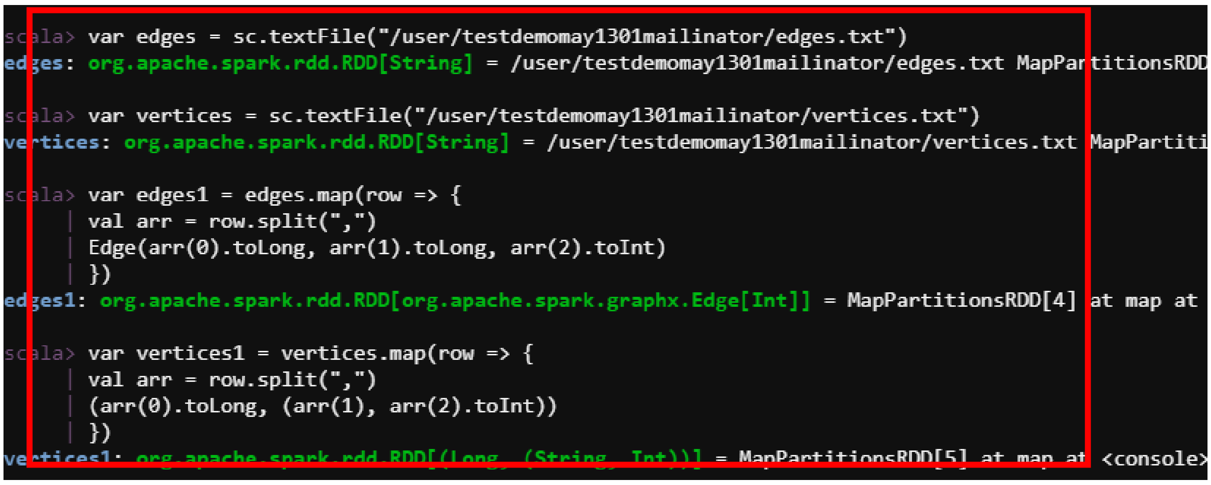
```
Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_312)
Type in expressions to have them evaluated.
Type :help for more information.

scala> import org.apache.spark.SparkContext
import org.apache.spark.SparkContext

scala> import org.apache.spark.graphx.{Edge, Graph}
import org.apache.spark.graphx.{Edge, Graph}

scala> import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.SparkSession
```

Step 15: Read the **“vertices”** and **“edges”** data by specifying the paths to the files that were uploaded in HDFS

Command:

```
var edges = sc.textFile("/user/testdemomay1301mailinator/edges.txt")

var vertices = sc.textFile("/user/testdemomay1301mailinator/vertices.txt")

var edges1 = edges.map(row => {

val arr = row.split(",")

Edge(arr(0).toLong, arr(1).toLong, arr(2).toInt)

})

var vertices1 = vertices.map(row => {
```

```
val arr = row.split(",")

(arr(0).toLong, (arr(1), arr(2).toInt))

})
```

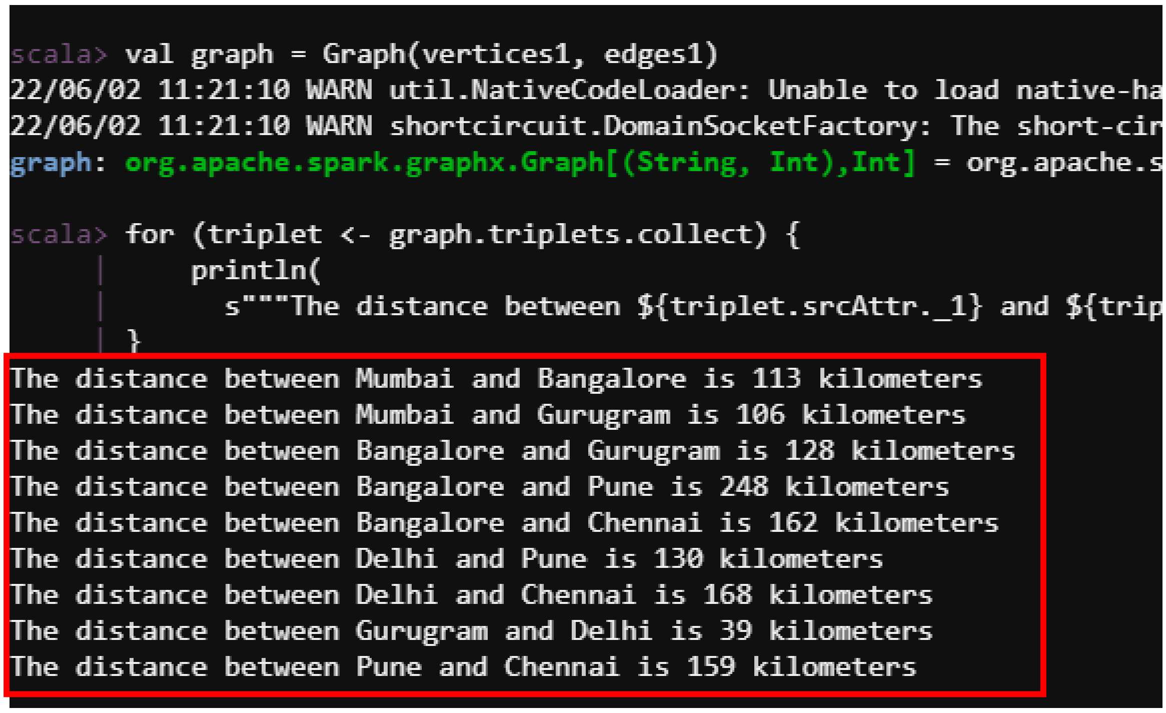


```
scala> var edges = sc.textFile("/user/testdemomay1301mailinator/edges.txt")
edges: org.apache.spark.rdd.RDD[String] = /user/testdemomay1301mailinator/edges.txt MapPartitionsRDD

scala> var vertices = sc.textFile("/user/testdemomay1301mailinator/vertices.txt")
vertices: org.apache.spark.rdd.RDD[String] = /user/testdemomay1301mailinator/vertices.txt MapPartiti

scala> var edges1 = edges.map(row => {
     | val arr = row.split(",")
     | Edge(arr(0).toLong, arr(1).toLong, arr(2).toInt)
     | })
edges1: org.apache.spark.rdd.RDD[org.apache.spark.graphx.Edge[Int]] = MapPartitionsRDD[4] at map at

scala> var vertices1 = vertices.map(row => {
     | val arr = row.split(",")
     | (arr(0).toLong, (arr(1), arr(2).toInt))
     | })
vertices1: org.apache.spark.rdd.RDD[(Long, (String, Int))] = MapPartitionsRDD[5] at map at <console>
```

Step 16: Create a graph object from the **"vertices1"** and **"edges1"** array to calculate the distance between the cities and display the output

Command:

```
val graph = Graph(vertices1, edges1)

for (triplet <- graph.triplets.collect) {

  println(

    s"""The distance between ${triplet.srcAttr._1} and ${triplet.dstAttr._1}
is ${triplet.attr} kilometers""")

}
```



```
scala> val graph = Graph(vertices1, edges1)
22/06/02 11:21:10 WARN util.NativeCodeLoader: Unable to load native-ha
22/06/02 11:21:10 WARN shortcircuit.DomainSocketFactory: The short-cir
graph: org.apache.spark.graphx.Graph[(String, Int),Int] = org.apache.s

scala> for (triplet <- graph.triplets.collect) {
     |      println(
     |        s"""The distance between ${triplet.srcAttr._1} and ${trip
     | }
The distance between Mumbai and Bangalore is 113 kilometers
The distance between Mumbai and Gurugram is 106 kilometers
The distance between Bangalore and Gurugram is 128 kilometers
The distance between Bangalore and Pune is 248 kilometers
The distance between Bangalore and Chennai is 162 kilometers
The distance between Delhi and Pune is 130 kilometers
The distance between Delhi and Chennai is 168 kilometers
The distance between Gurugram and Delhi is 39 kilometers
The distance between Pune and Chennai is 159 kilometers
```