Assisted Practice 16.1: Create DataFrame Using PySpark to Process Records

Problem Scenario: Create a PySpark DataFrame to filter 10 complete records from a real-world retail business dataset

Objective: In this demonstration, you will use a PySpark DataFrame to read the data from **"HDFS"** and filter only complete orders.

Dataset Name: "order_parquet"

Dataset Description: This dataset is about the order details, which have order_id, order_date, order_customer_id, and order_status in it with 68883 rows × 4 columns.

Tasks to Perform:

1. Download the dataset from the course resource section and upload it into the HDFS using "Hue"
2. Login into the Webconsole and open the PySpark shell
3. Import functions as F from "pyspark.sql"
4. As a PySpark DataFrame, read the **order_parquet** data from HDFS
5. Filter complete orders to show 10 records
6. Display the complete order list

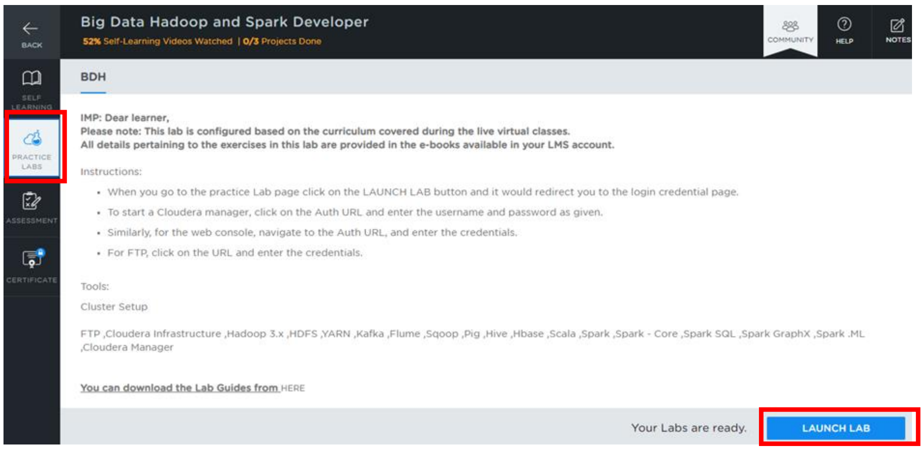
Steps to Perform:

Step 1: Download the dataset with the name **"order_parquet"** 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 click 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:

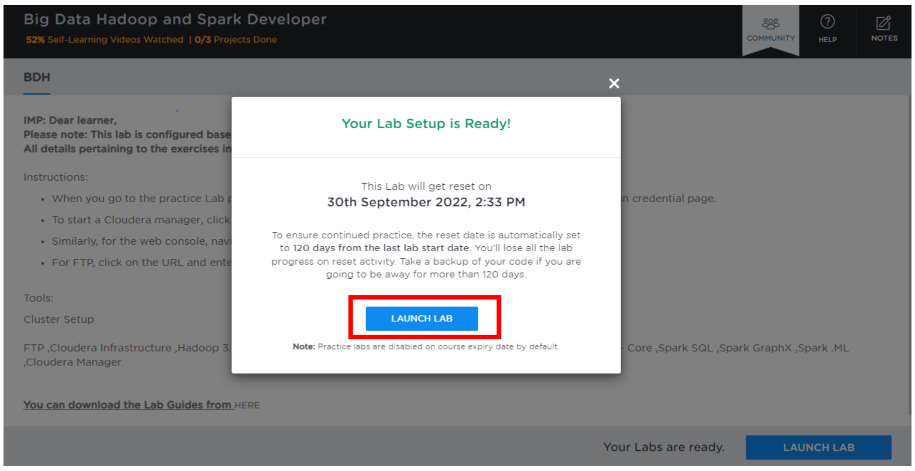
Cluster Setup

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



Big Data Hadoop and Spark Developer

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

IMP: Dear learner,
Please note: This lab is configured base
All details pertaining to the exercises in

Instructions:

- When you go to the practice Lab p … n credential page.
- To start a Cloudera manager, click
- Similarly, for the web console, navi
- For FTP, click on the URL and ente

Tools:

Cluster Setup

FTP ,Cloudera Infrastructure ,Hadoop 3. … Core ,Spark SQL ,Spark GraphX ,Spark .ML ,Cloudera Manager

You can download the Lab Guides from HERE

Your Lab Setup is Ready!

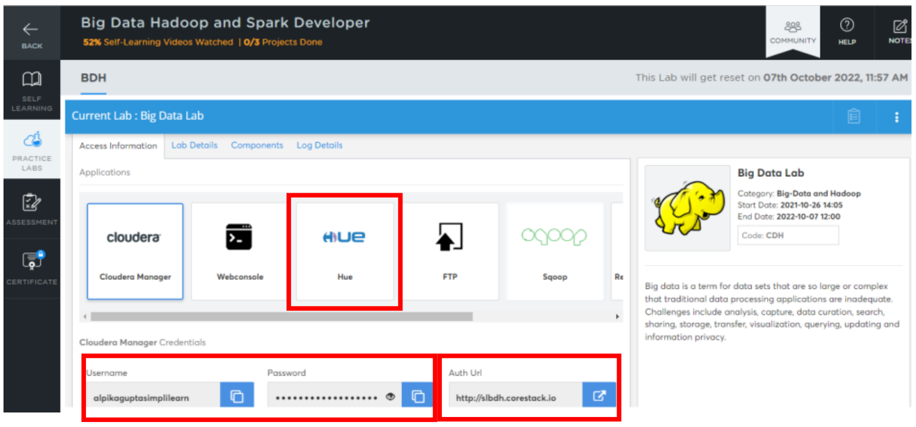
This Lab will get reset on
30th September 2022, 2:33 PM

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.

Your Labs are ready.

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



Big Data Hadoop and Spark Developer
52% Self-Learning Videos Watched | 0/3 Projects Done

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

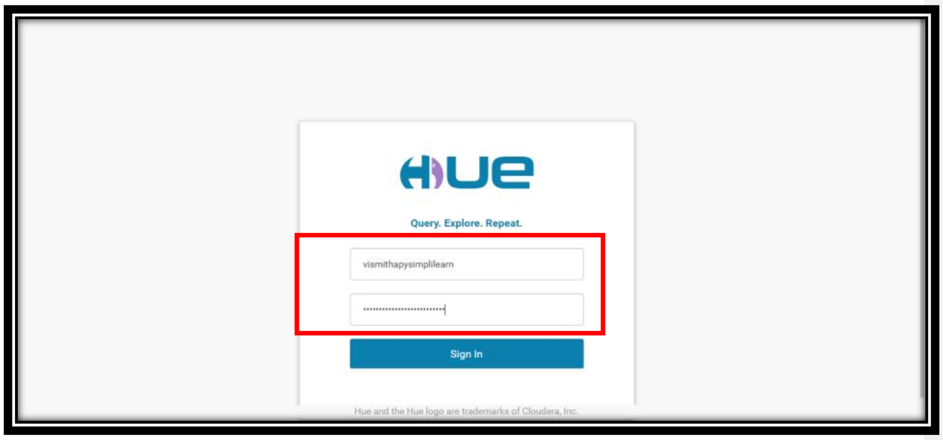
Big Data Lab
Category: Big-Data and Hadoop
Start Date: 2021-10-26 14:05
End Date: 2022-10-07 12:00
Code: CDH

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.

Cloudera Manager Credentials

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

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

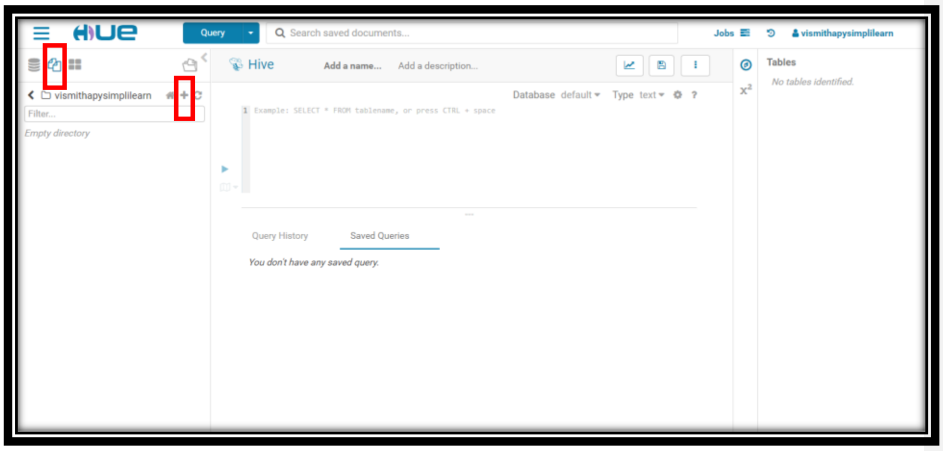


Query. Explore. Repeat.

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

Hue and the Hue logo are trademarks of Cloudera, Inc.

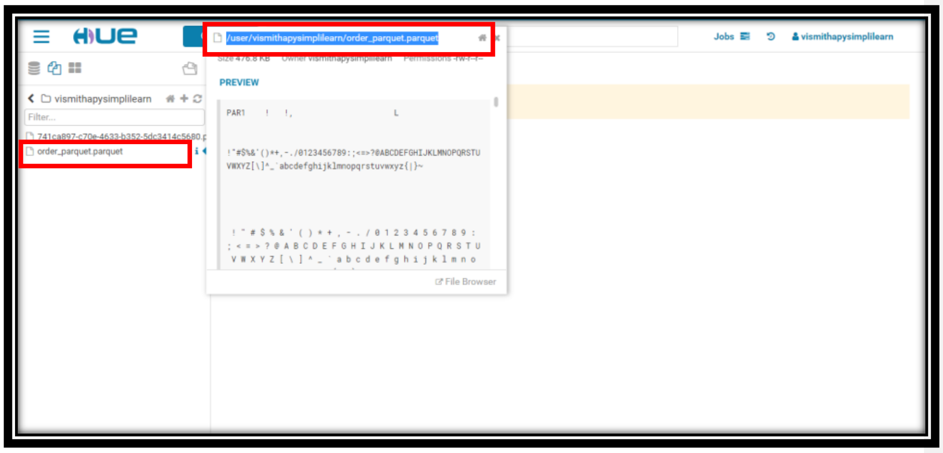
Step 8: Click on **“HDFS”** icon and click on the **“+”** symbol to upload the dataset



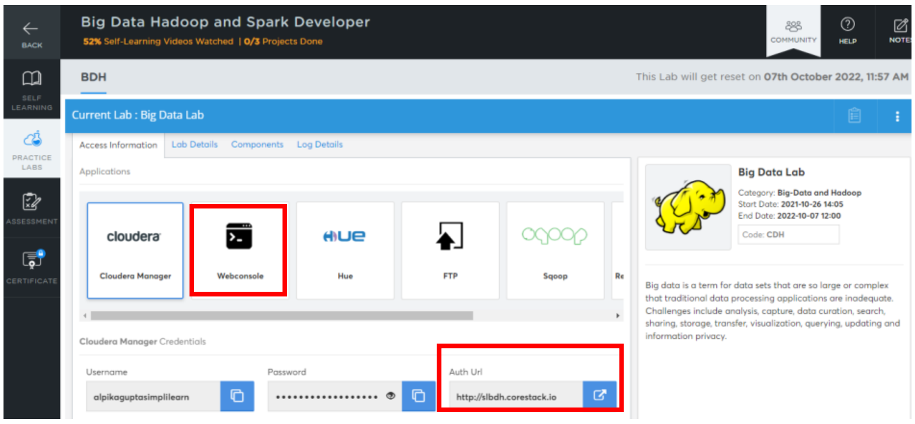
Step 9: Select the downloaded dataset file and upload it to **HDFS.** In addition, by right-clicking, copy the path from the dataset that has been uploaded.

Commented [SB1]: Is double quotes required?

Commented [AG2R1]: no


/user/vismithapysimplilearn/order_parquet.parquet
PREVIEW
vismithapysimplilearn
Filter...
741ca897-c70e-4633-b352-5dc3414c5680.p
order_parquet.parquet
Jobs
File Browser

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



Big Data Hadoop and Spark Developer

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

BDH

This Lab will get reset on 07th October 2022, 11:57 AM

Current Lab : Big Data Lab

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.

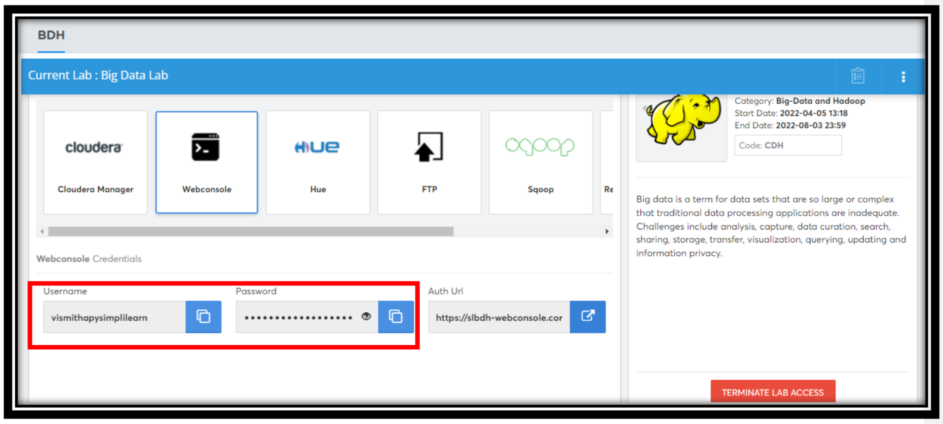
Cloudera Manager Credentials

Username: alpikaguptasimplilearn

Password: ••••••••••••••••••

Auth Url: http://slbdh.corestack.io

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



BDH

Current Lab : 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.

Webconsole Credentials

Username: vismithapysimplilearn

Password: ••••••••••••••••••

Auth Url: https://slbdh-webconsole.cor

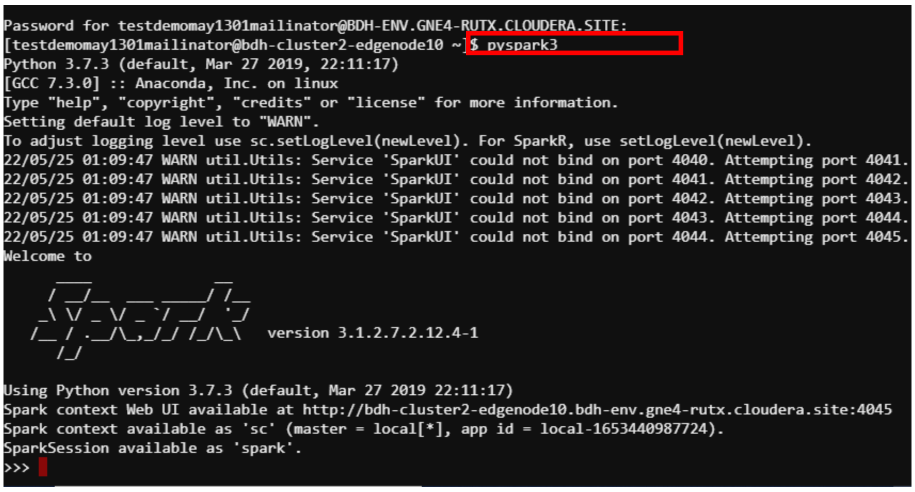
Step 12: Paste the **Username** and the **Password** on the console and click on enter.

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

Step 13: Enter the **“PySpark”** shell by running the below command.

Command:

pyspark3



```
Password for testdemomay1301mailinator@BDH-ENV.GNE4-RUTX.CLOUDERA.SITE:
[testdemomay1301mailinator@bdh-cluster2-edgenode10 ~]$ pyspark3
Python 3.7.3 (default, Mar 27 2019, 22:11:17)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/05/25 01:09:47 WARN util.Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
22/05/25 01:09:47 WARN util.Utils: Service 'SparkUI' could not bind on port 4041. Attempting port 4042.
22/05/25 01:09:47 WARN util.Utils: Service 'SparkUI' could not bind on port 4042. Attempting port 4043.
22/05/25 01:09:47 WARN util.Utils: Service 'SparkUI' could not bind on port 4043. Attempting port 4044.
22/05/25 01:09:47 WARN util.Utils: Service 'SparkUI' could not bind on port 4044. Attempting port 4045.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.2.7.2.12.4-1
      /_/

Using Python version 3.7.3 (default, Mar 27 2019 22:11:17)
Spark context Web UI available at http://bdh-cluster2-edgenode10.bdh-env.gne4-rutx.cloudera.site:4045
Spark context available as 'sc' (master = local[*], app id = local-1653440987724).
SparkSession available as 'spark'.
>>>
```

Step 14: Import the function **F** from **pyspark.sql**

Commented [SB3]: double quotes required?

Commented [AG4R3]: Nowe can remove that

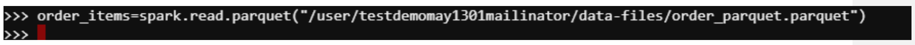
Command:

from pyspark.sql import functions as F

Step 15: Read the dataset as shown below, specifying the path of the dataset

Command:

order_items=spark.read.parquet
("/user/testdemomay1301mailinator/data-files/order_parquet.parquet")



```
>>> order_items=spark.read.parquet("/user/testdemomay1301mailinator/data-files/order_parquet.parquet")
>>>
```

Step 16: Get data for completed orders in the dataset by using the command below

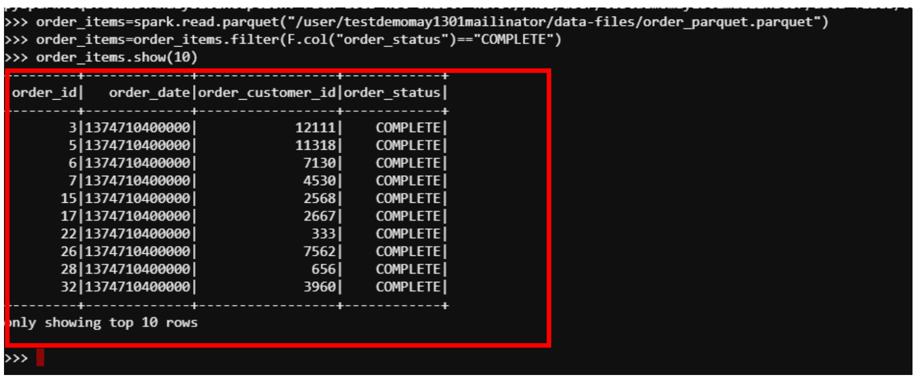
Command:

order_items=order_items.filter(F.col("order_status")=="COMPLETE")

Step 17: Display the complete order list using below command:

Command:

order_items.show(10)



```
>>> order_items=spark.read.parquet("/user/testdemomay1301mailinator/data-files/order_parquet.parquet")
>>> order_items=order_items.filter(F.col("order_status")=="COMPLETE")
>>> order_items.show(10)
--------+-------------+-----------------+------------+
 order_id|   order_date|order_customer_id|order_status|
--------+-------------+-----------------+------------+
        3|1374710400000|            12111|    COMPLETE|
        5|1374710400000|            11318|    COMPLETE|
        6|1374710400000|             7130|    COMPLETE|
        7|1374710400000|             4530|    COMPLETE|
       15|1374710400000|             2568|    COMPLETE|
       17|1374710400000|             2667|    COMPLETE|
       22|1374710400000|              333|    COMPLETE|
       26|1374710400000|             7562|    COMPLETE|
       28|1374710400000|              656|    COMPLETE|
       32|1374710400000|             3960|    COMPLETE|
--------+-------------+-----------------+------------+
only showing top 10 rows

>>> 
```