Assisted Practice 16.2: UDF with DataFrame

Problem Scenario: Create a DataFrame and a Python function to convert it into UDF

Objective: In this demonstration, you will create a built-in function using UDF to convert the first letter of every word into uppercase.

Tasks to Perform:

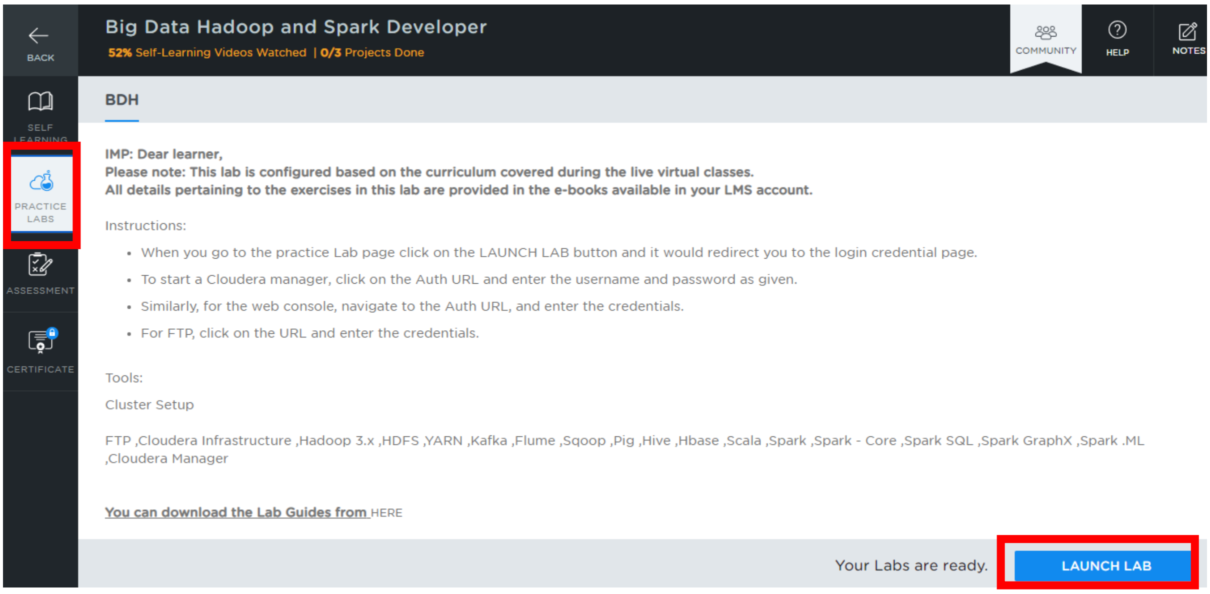
1. Import required packages and create a DataFrame with two columns (S_No, Name)
2. Create a function convertCase() which will convert the first letter of every word into a capital letter
3. Convert a Python function to a PySpark UDF
4. Apply the convertUDF function to a DataFrame column as a built-in function

Steps to Perform:

Step 1: Log in to your LMS account

Step 2: Open the course “**Big Data Hadoop and Spark developer**”

Step 3: 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

BDH

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:

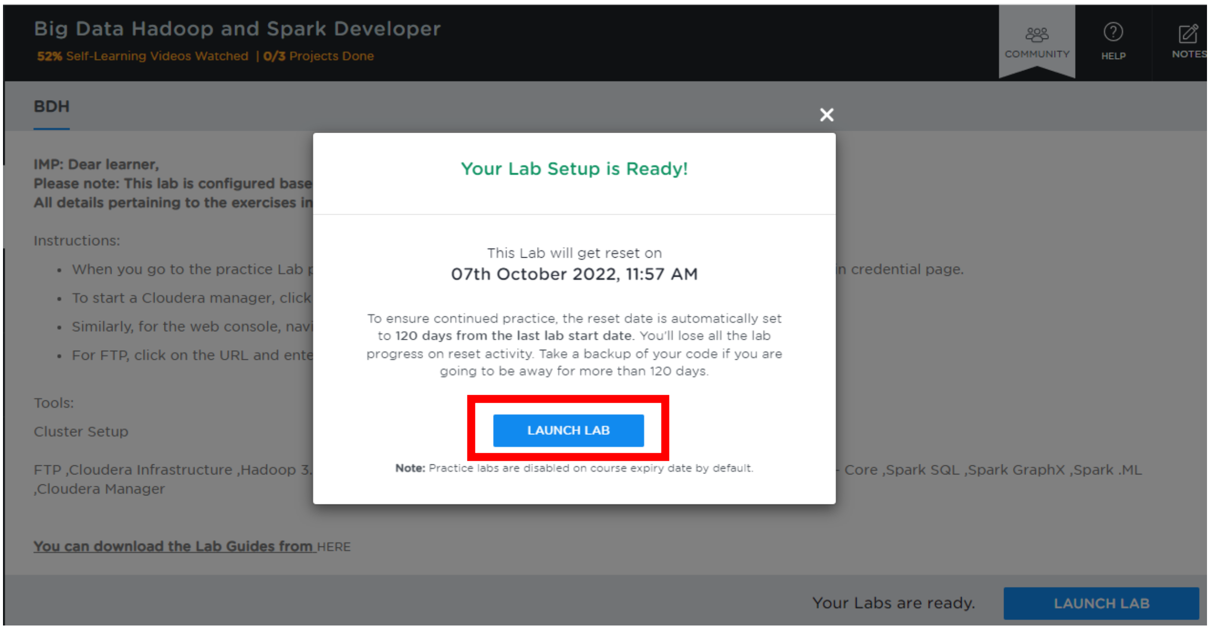
Cluster Setup

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 4: Again, click on the "**LAUNCH LAB**" button



Your Lab Setup is Ready!

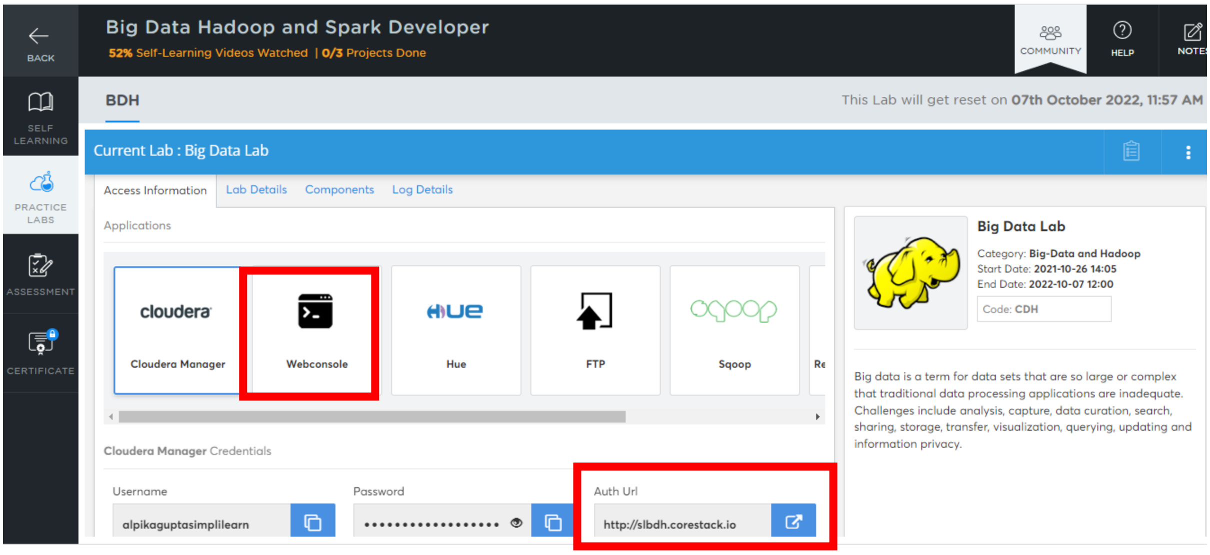
This Lab will get reset on
07th October 2022, 11:57 AM

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.

Big Data Hadoop and Spark Developer
52% Self-Learning Videos Watched | 0/3 Projects Done
BDH
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 Labs are ready.

Step 5: 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

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

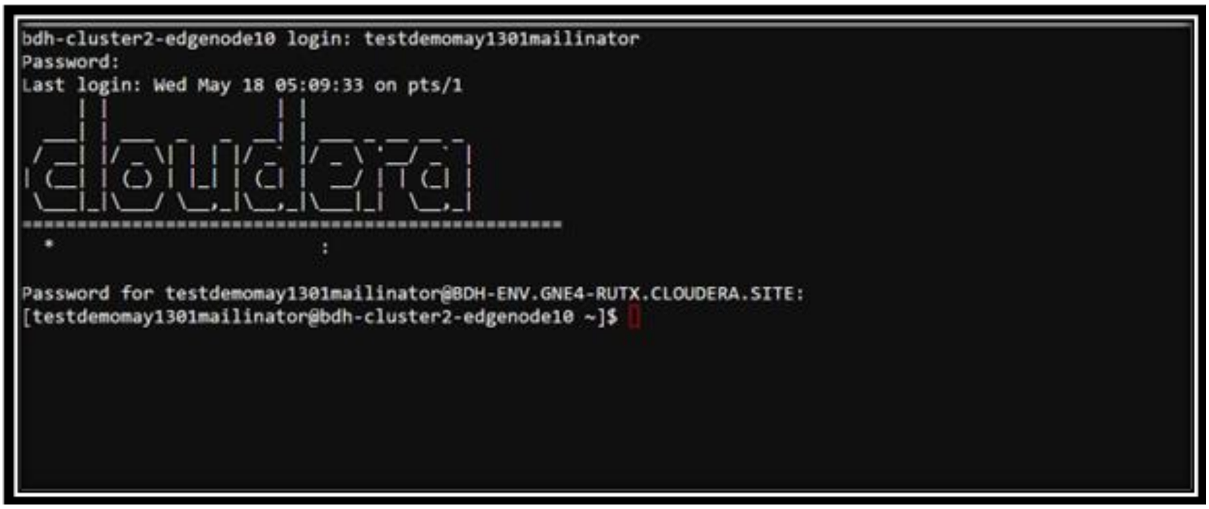
Code: CDH

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 6: Copy the **Username** and the **Password** provided to log in to the Webconsole

Step 7: 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.

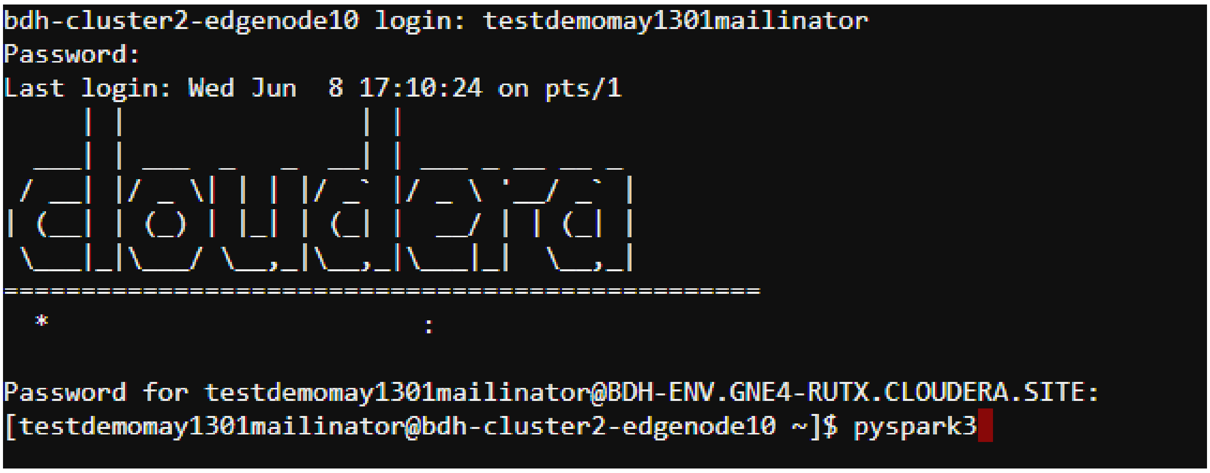


```
bdh-cluster2-edgenode10 login: testdemomay1301mailinator
Password:
Last login: Wed May 18 05:09:33 on pts/1
==================================================
 *                        :

Password for testdemomay1301mailinator@BDH-ENV.GNE4-RUTX.CLOUDERA.SITE:
[testdemomay1301mailinator@bdh-cluster2-edgenode10 ~]$
```

Step 8: Login into the PySpark shell using the below command:

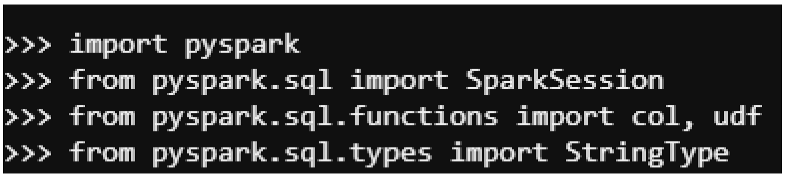
Command: pyspark3



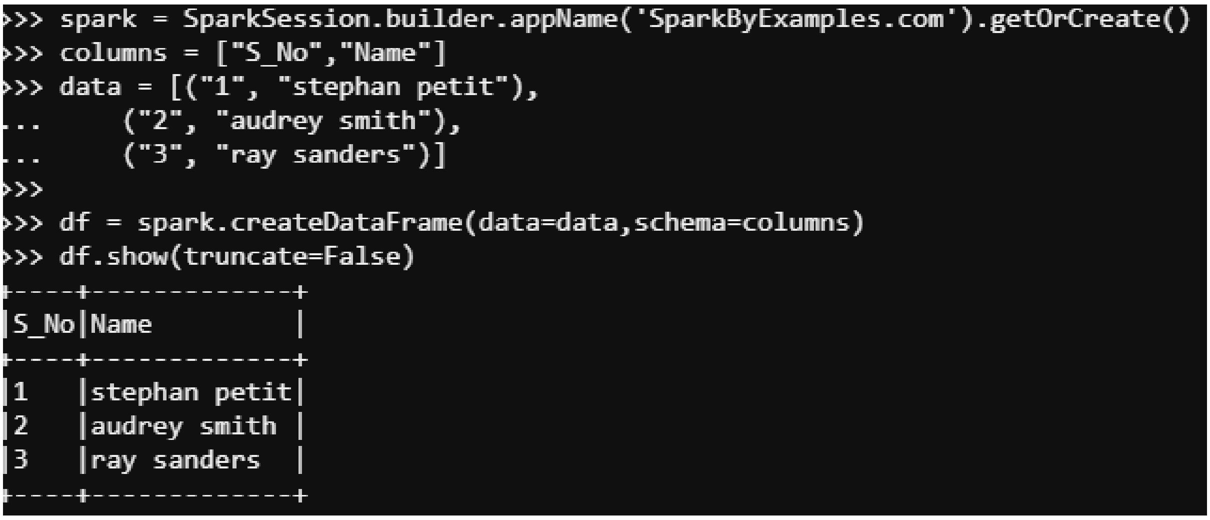
```
bdh-cluster2-edgenode10 login: testdemomay1301mailinator
Password:
Last login: Wed Jun  8 17:10:24 on pts/1
     | |               | |
  ___| | ___  _   _  __| | ___ _ __ __ _
 / __| |/ _ \| | | |/ _` |/ _ \ '__/ _` |
| (__| | (_) | |_| | (_| |  __/ | | (_| |
 \___|_|\___/ \__,_|\__,_|\___|_|  \__,_|
=================================================
  *                          :

Password for testdemomay1301mailinator@BDH-ENV.GNE4-RUTX.CLOUDERA.SITE:
[testdemomay1301mailinator@bdh-cluster2-edgenode10 ~]$ pyspark3
```

Step 9: Import the required libraries and create a DataFrame with two columns (S_No, Name)

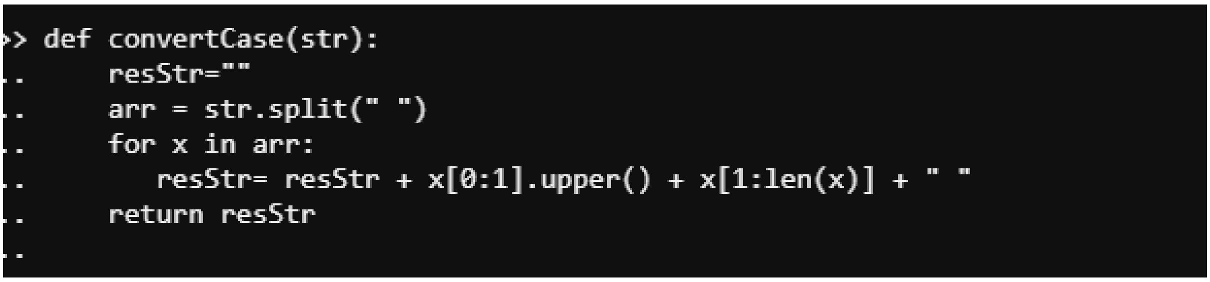


```
>>> import pyspark
>>> from pyspark.sql import SparkSession
>>> from pyspark.sql.functions import col, udf
>>> from pyspark.sql.types import StringType
```



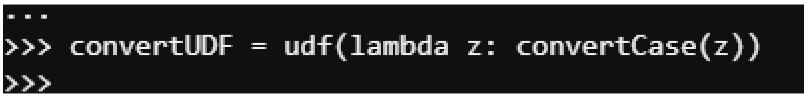
```
>>> spark = SparkSession.builder.appName('SparkByExamples.com').getOrCreate()
>>> columns = ["S_No","Name"]
>>> data = [("1", "stephan petit"),
...     ("2", "audrey smith"),
...     ("3", "ray sanders")]
>>>
>>> df = spark.createDataFrame(data=data,schema=columns)
>>> df.show(truncate=False)
+----+-------------+
|S_No|Name         |
+----+-------------+
|1   |stephan petit|
|2   |audrey smith |
|3   |ray sanders  |
+----+-------------+
```

Step 10: Creates a function convertCase() which takes a string parameter and converts the first letter of every word to capital letter



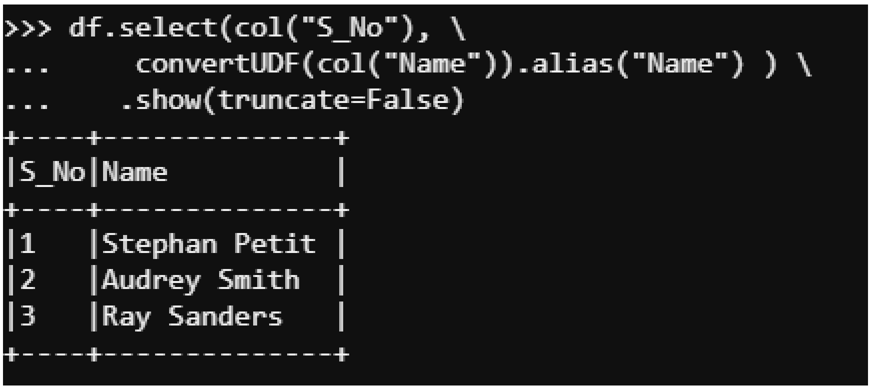
```
>>> def convertCase(str):
...     resStr=""
...     arr = str.split(" ")
...     for x in arr:
...        resStr= resStr + x[0:1].upper() + x[1:len(x)] + " "
...     return resStr
...
```

Step 11: Convert the function convertCase() to convertUDF



```
...
>>> convertUDF = udf(lambda z: convertCase(z))
>>>
```

Step 12: Now, you can use convertUDF on a DataFrame column as a regular built-in function



```
>>> df.select(col("S_No"), \
...     convertUDF(col("Name")).alias("Name") ) \
...    .show(truncate=False)
+----+--------------+
|S_No|Name          |
+----+--------------+
|1   |Stephan Petit |
|2   |Audrey Smith  |
|3   |Ray Sanders   |
+----+--------------+
```