Assisted Practice 14: Deployment of PySpark Job

Problem Scenario: Andrew has been hired by a bank as a Data Engineer. Andrew's primary responsibility is to compare the performance of a Spark application in both cluster and client mode. Andrew is required to submit the same job in both modes and compare the results.

Objective: In this demonstration, you will learn how to deploy a PySpark job in client and cluster mode.

Tasks to Perform:

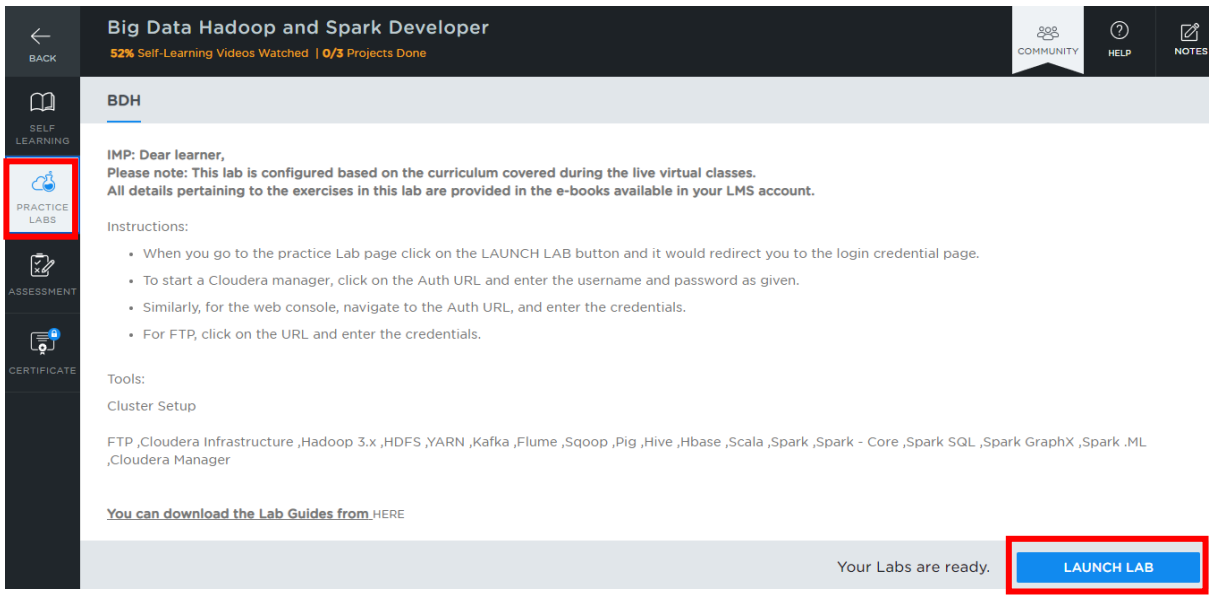
1. Create a Python file in the Web Console using the vi editor
2. Import the required libraries and create a Spark session to initialize the code
3. Submit the job in Client-mode using the spark-submit --deploy-mode client map.py command
4. Submit the job in Cluster-mode using the spark-submit --deploy-mode cluster map.py command

Steps to Perform:

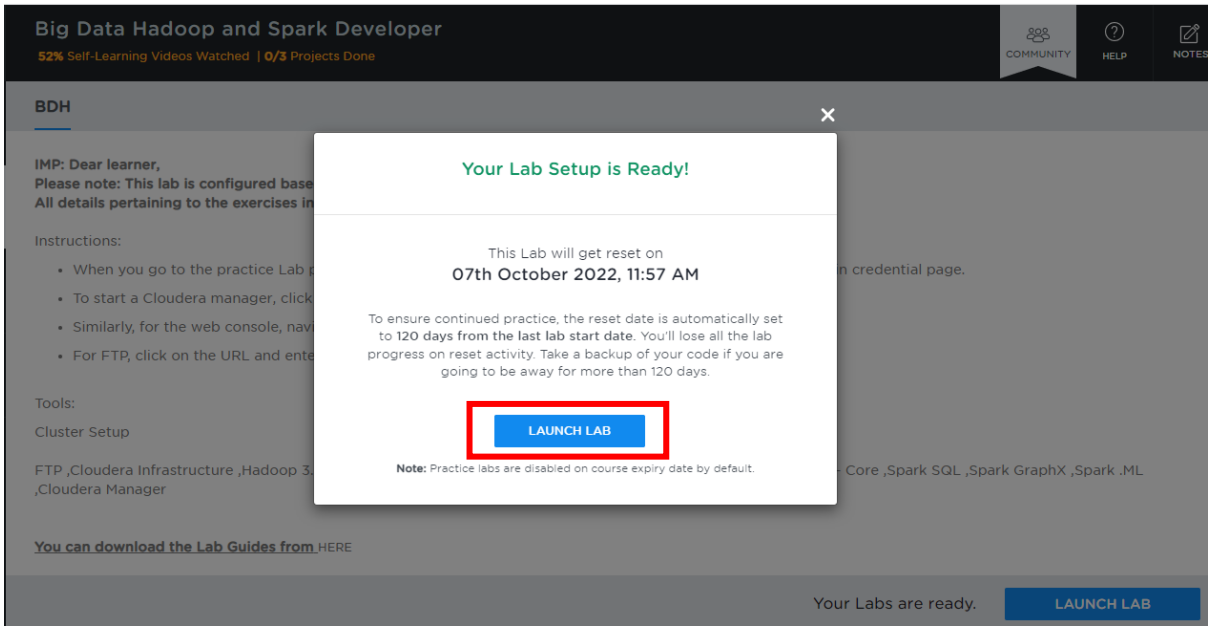
Step 1: Log in to your LMS account

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

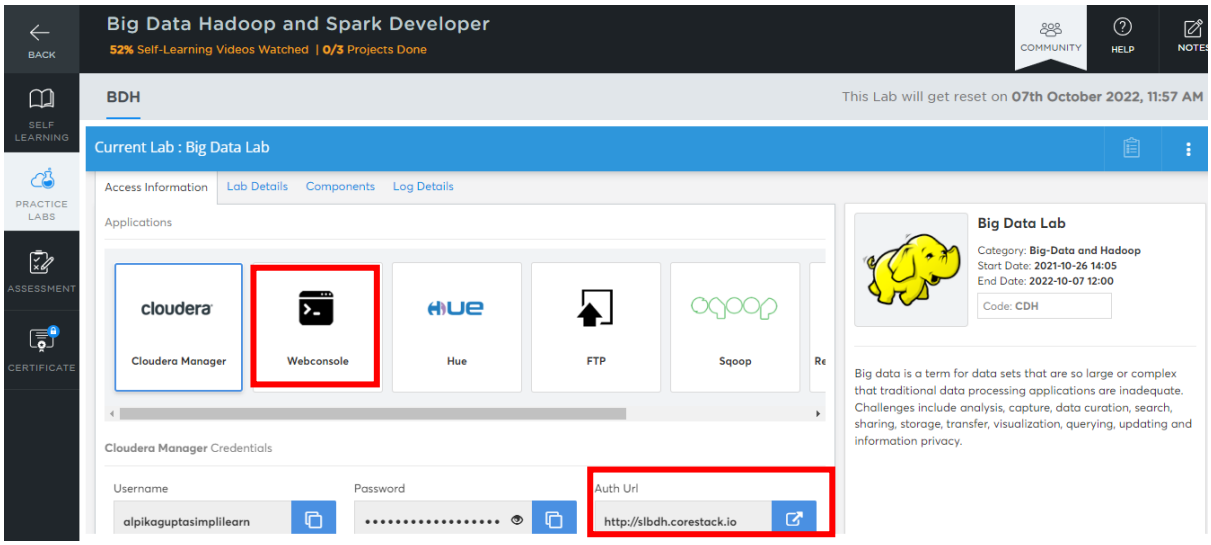
Step 3: On the left side, click on the "**PRACTICE LABS**" tab and click on the "**LAUNCH LAB**" button



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



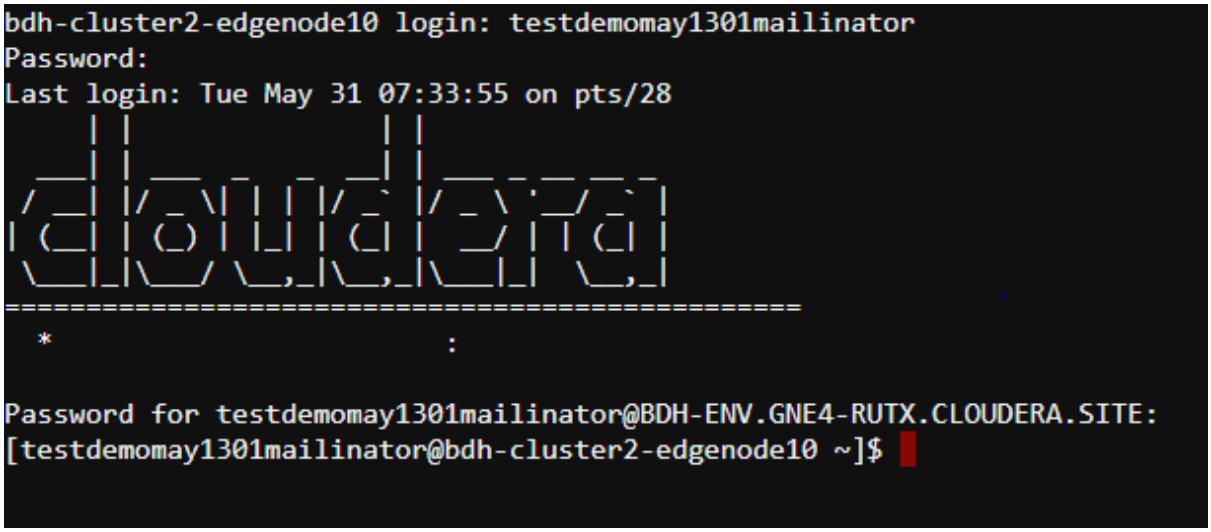
Step 5: Click on "**Webconsole**" and click on the "**Auth Url**"



Step 6: Copy the "**Username**" and the "**Password**" provided to log in to the web console

Step 7: Paste the "**Username**" and the "**Password**" on the console and click on enter

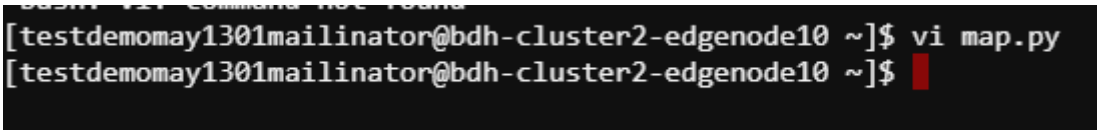
Note: The password will not be visible when pasted on the console.



Step 8: Create a Python file using the below command:

Command:

vi map.py



Step 9: Import the necessary libraries and enter the following code:

Command:

```
    sc= SparkSession \
   .builder \
   .appName("Simplilearn Examples") \
   .getOrCreate() \
   .sparkContext

words = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday"]

wordsRDD = sc.parallelize(words)

wordsRDD = wordsRDD.map(lambda word: (word, len(word)))

for word in wordsRDD.collect():
   print(word)
```

```
from pyspark import SparkContext
from pyspark.sql import SparkSession

sc= SparkSession \
    .builder \
    .appName("Simplilearn Examples") \
    .getOrCreate() \
    .sparkContext

words = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]

wordsRDD = sc.parallelize(words)

wordsRDD = wordsRDD.map(lambda word: (word, len(word)))

for word in wordsRDD.collect():
    print(word)

~
~
~
~
```

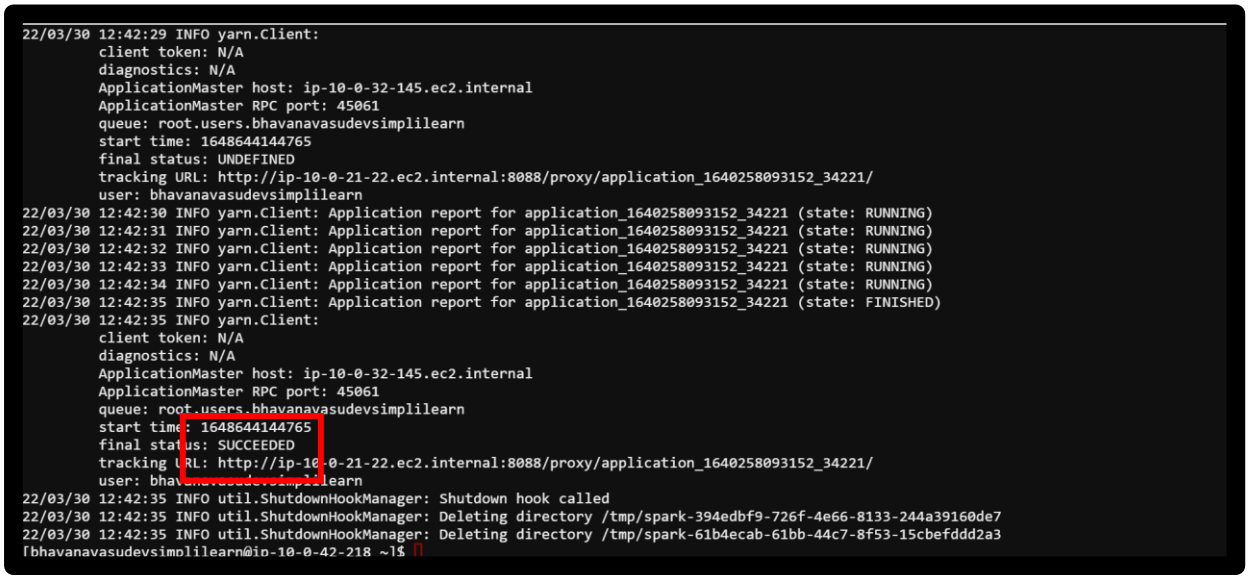
Step 10: Submit the job in Client-mode
Command:

```
spark3-submit --conf spark.ui.port=6065 --deploy-mode client
map.py
```

```
22/06/02 11:36:23 INFO scheduler.DAGSche
22/06/02 11:36:23 INFO scheduler.TaskSch
22/06/02 11:36:23 INFO scheduler.DAGSche
('Sunday', 6)
('Monday', 6)
('Tuesday', 7)
('Wednesday', 9)
('Thursday', 8)
('Friday', 6)
('Saturday', 8)
22/06/02 11:36:23 INFO server.AbstractCo
22/06/02 11:36:23 INFO ui.SparkUI: Stopp
22/06/02 11:36:23 INFO spark.MapOutputTr
22/06/02 11:36:23 INFO memory.MemoryStor
22/06/02 11:36:23 INFO storage.BlockMana
```

Step 11: Submit the job in Cluster-mode

Command:
spark3-submit –conf spark.ui.port=6065 --deploy-mode cluster map.py
or
spark3-shell --master yarn



You will be able to see the status on cluster mode as **“SUCCEEDED”** which indicates that the task is completed.