

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

Big Data Hadoop and Spark Developer

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

COMMUNITY HELP NOTES

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:

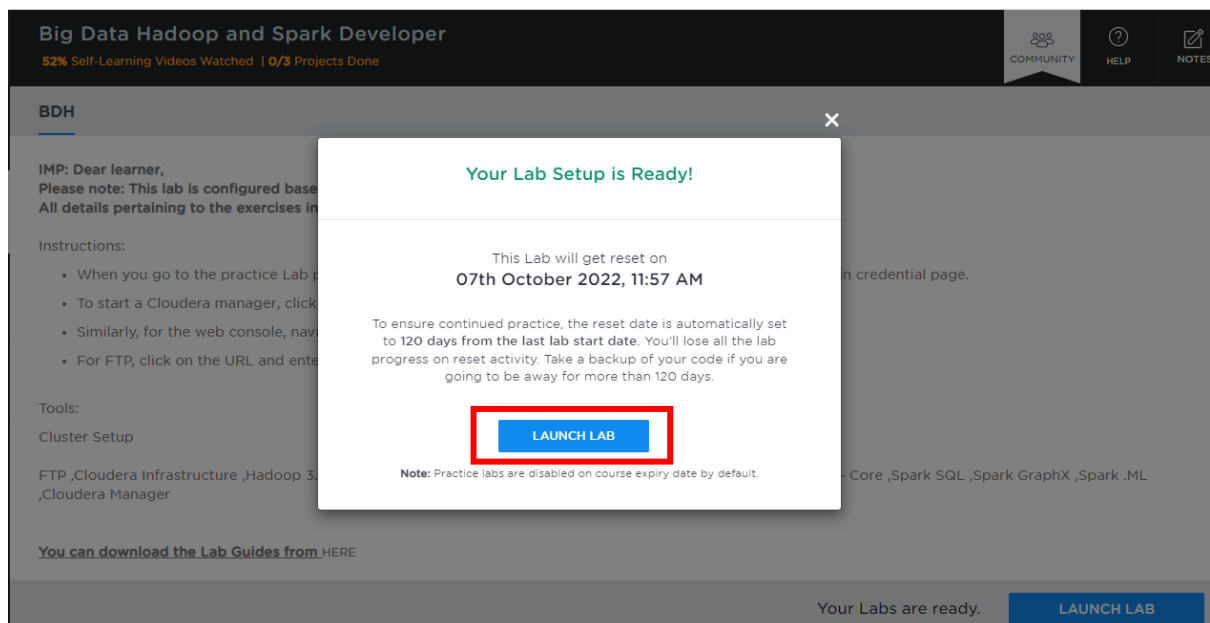
Cluster Setup

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

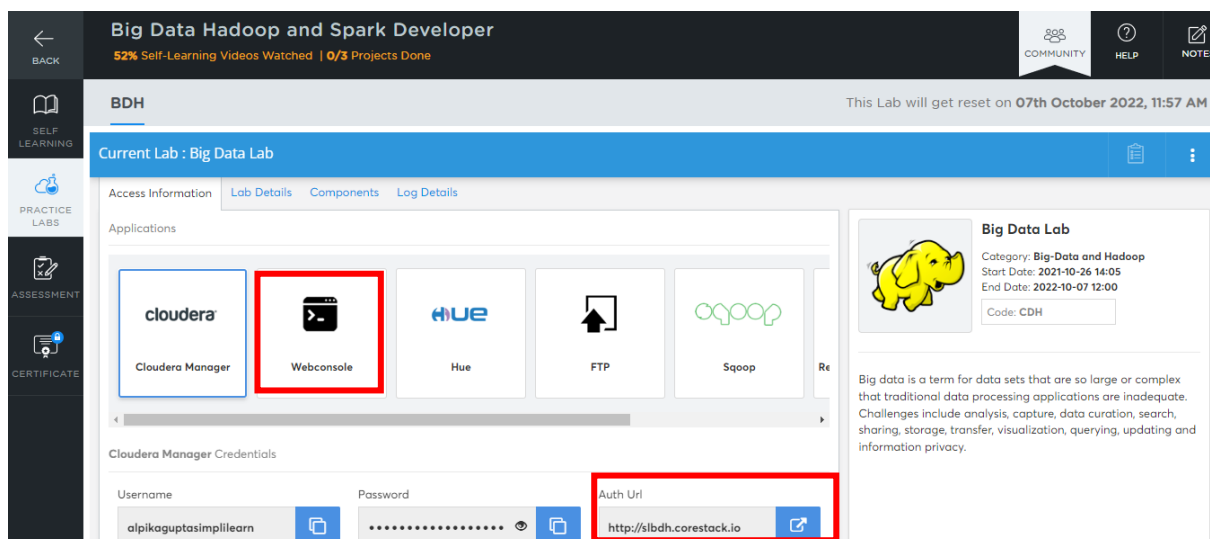
[You can download the Lab Guides from HERE](#)

Your Labs are ready. **LAUNCH LAB**

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.

```
bdh-cluster2-edgenode10 login: testdemomay1301mailinator
Password:
Last login: Tue May 31 07:33:55 on pts/28

=====
*                               :
=====

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

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

Command:

vi map.py

```
[testdemomay1301mailinator@bdh-cluster2-edgenode10 ~]$ vi map.py
[testdemomay1301mailinator@bdh-cluster2-edgenode10 ~]$
```

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.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
```

```

22/03/30 12:42:29 INFO yarn.Client:
  client token: N/A
  diagnostics: N/A
  ApplicationMaster host: ip-10-0-32-145.ec2.internal
  ApplicationMaster RPC port: 45061
  queue: root.users.bhavanavasudevsimplilearn
  start time: 1648644144765
  final status: UNDEFINED
  tracking URL: http://ip-10-0-21-22.ec2.internal:8088/proxy/application_1640258093152_34221/
  user: bhavanavasudevsimplilearn
22/03/30 12:42:30 INFO yarn.Client: Application report for application_1640258093152_34221 (state: RUNNING)
22/03/30 12:42:31 INFO yarn.Client: Application report for application_1640258093152_34221 (state: RUNNING)
22/03/30 12:42:32 INFO yarn.Client: Application report for application_1640258093152_34221 (state: RUNNING)
22/03/30 12:42:33 INFO yarn.Client: Application report for application_1640258093152_34221 (state: RUNNING)
22/03/30 12:42:34 INFO yarn.Client: Application report for application_1640258093152_34221 (state: RUNNING)
22/03/30 12:42:35 INFO yarn.Client: Application report for application_1640258093152_34221 (state: FINISHED)
22/03/30 12:42:35 INFO yarn.Client:
  client token: N/A
  diagnostics: N/A
  ApplicationMaster host: ip-10-0-32-145.ec2.internal
  ApplicationMaster RPC port: 45061
  queue: root.users.bhavanavasudevsimplilearn
  start time: 1648644144765
  final status: SUCCEEDED
  tracking URL: http://ip-10-0-21-22.ec2.internal:8088/proxy/application_1640258093152_34221/
  user: bhavanavasudevsimplilearn
22/03/30 12:42:35 INFO util.ShutdownHookManager: Shutdown hook called
22/03/30 12:42:35 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-394edbf9-726f-4e66-8133-244a39160de7
22/03/30 12:42:35 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-61b4ecab-61bb-44c7-8f53-15cbefddd2a3
bhavanavasudevsimplilearn@ip-10-0-42-218 ~$

```

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