Assisted Practice 19: Working with Spark Structured Application

Problem Scenario: Create a Spark Structured streaming application to work with real-time data

Objective: In this demonstration, you will learn how to create a real-time Spark Structured streaming application.

Tasks to Perform:

1. Open the **"Webconsole"** and start the **"netcat"** with any port number
2. Log in to the PySpark shell in another **"Webconsole"**
3. Create a **"DataStreamReader"** by importing the necessary packages
4. Write a program to split each line with space and execute the code
5. Calculate the network word count using the groupby and count functions of real-time streaming data

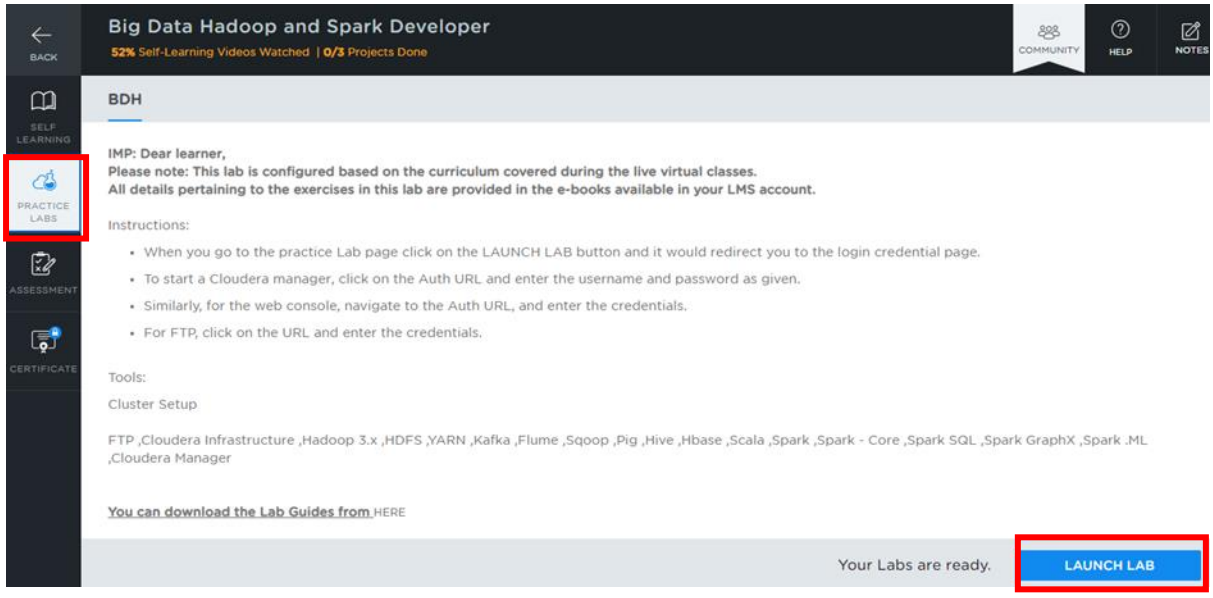
Steps to Perform:

Step 1: Log in to your LMS account

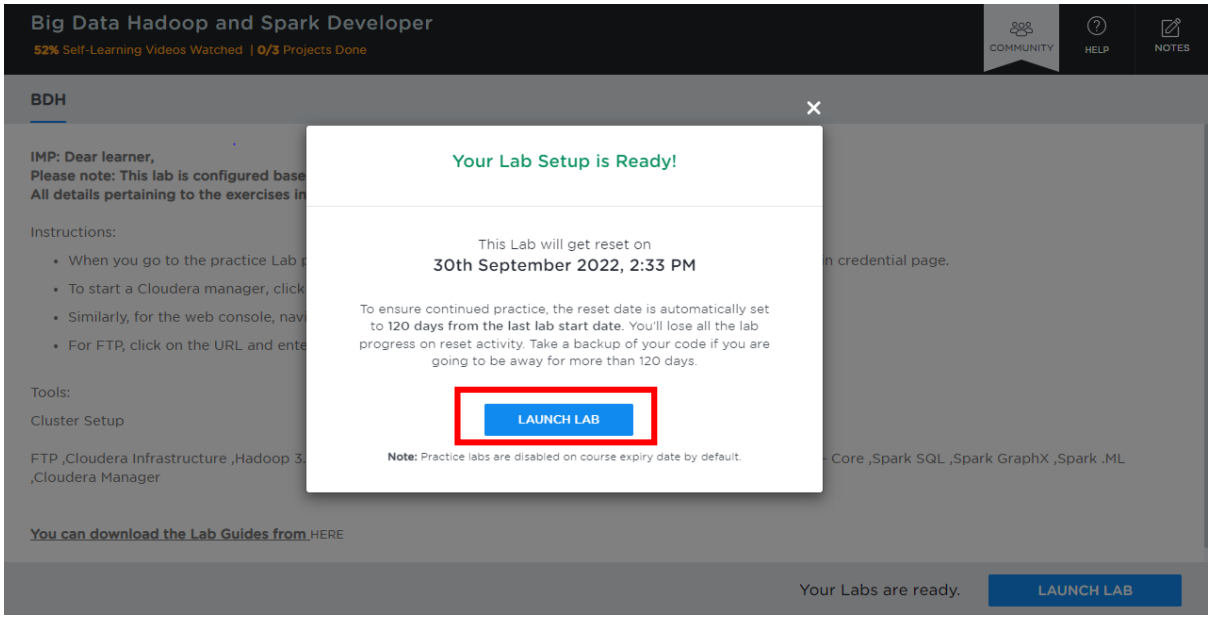
Step 2: Open the course **"Big Data Hadoop and Spark developer"**

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

© Copyright 2022, Simplilearn. All rights reserved.

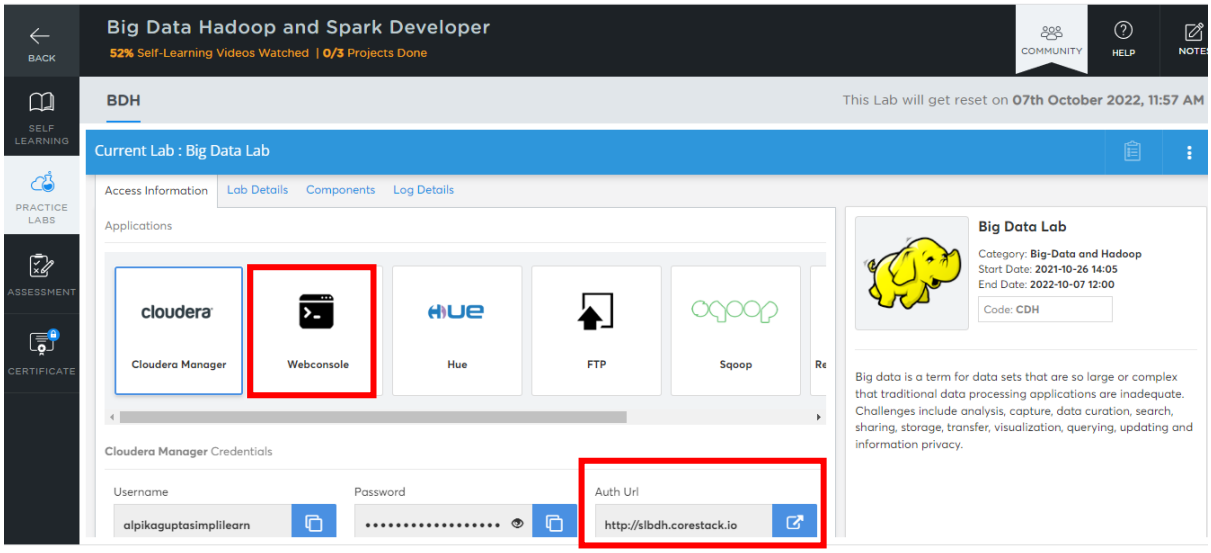


Step 4: Again, click on the **"LAUNCH LAB"** button

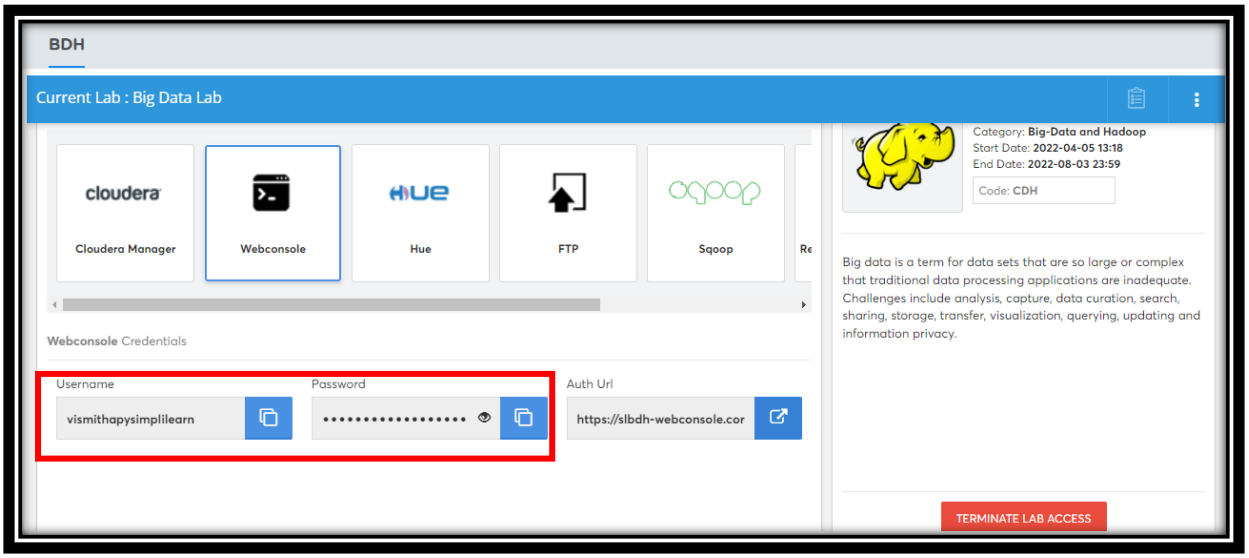


Step 5: Click on "**Webconsole**" and click on the **"Auth Url"**

© Copyright 2022, Simplilearn. All rights reserved.



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

Step 8: Open a **"Webconsole"** and start **"netcat"** with any port number

Command:

nc -lk 4499

© Copyright 2022, Simplilearn. All rights reserved.

```
bdh-cluster2-edgenode10 login: testdemomay1301mailinator
Password:
Last login: Tue May 24 10:54:11 on pts/7
     | |              | |
  ___| | ___  _   _  __| | ___ _ __ __ _
 / __| |/ _ \| | | |/ _` |/ _ \ '__/ _` |
| (__| | (_) | |_| | (_| |  __/ | | (_| |
 \___|_|\___/ \__,_|\__,_|\___|_|  \__,_|
=================================================
  *                        :

Password for testdemomay1301mailinator@BDH-ENV.GNE4-RUTX.CLOUDERA.SITE:
[testdemomay1301mailinator@bdh-cluster2-edgenode10 ~]$ nc -lk 4499
```

Step 9: Open a PySpark shell in another "**Webconsole"** by running the command mentioned below:

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 10: Create a **"DataStreamReader"** by importing the necessary packages using the command below:

Note: The port number should be the same as mentioned in Step 8 as Terminal 1.

Command:

© Copyright 2022, Simplilearn. All rights reserved.

```
from pyspark.sql import functions as F
lines = spark \
    .readStream \
    .format("socket") \
    .option("host", "localhost") \
    .option("port", 4499) \
    .load()
```

```
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:4044
Spark context available as 'sc' (master = local[*], app id = local-1653390901723).
SparkSession available as 'spark'.
>> from pyspark.sql import functions as F
>> lines = spark \
..     .readStream \
..     .format("socket") \
..     .option("host", "localhost") \
..     .option("port", 4499) \
..     .load()
22/05/24 11:15:37 WARN sources.TextSocketSourceProvider: The socket source should not be used for production applications! It does not support recovery.
```

Step 11: Split each line with space using function F and explode the same using the below command:

Command:

```
words = lines.select(
    F.explode(
        F.split(lines.value, " ")
    ).alias("word")
)
```

```
>>> words = lines.select(
...     F.explode(
...         F.split(lines.value, " ")
...     ).alias("word")
... )
```

© Copyright 2022, Simplilearn. All rights reserved.

Step 12: Calculate the network word count using the groupby and count functions

Note: Word count output is initially empty.

```
wordCounts = words.groupBy("word").count()
query = wordCounts.writeStream \
.outputMode("complete") \
.format("console") \
.start()
```

```
>>> wordCounts = words.groupBy("word").count()
>>>
>>> query = wordCounts.writeStream \
...     .outputMode("complete") \
...     .format("console") \
...     .start()
22/05/24 11:17:08 WARN streaming.StreamingQueryManager: Temporary checkpoint lo
e2eb983c-9d00-4f5a-847d-8e441c478f8f. If it's required to delete it under any c
ue. Important to know deleting temp checkpoint folder is best effort.
```

Note: To pause the running code, below mentioned commands are used:

Command:

```
import time
time.sleep(35)
ssc.stop()
```

```
-------------------------------------------
Batch: 0
-------------------------------------------
+----+-----+
|word|count|
+----+-----+
+----+-----+
```

Step 13: Return to terminal 1, where the "**netcat** "is running, and type the data, which will be reflected in real-time in terminal 2

© Copyright 2022, Simplilearn. All rights reserved.

Command:

Sample data: Hello Hello

This is my first program

```
ip-10-0-42-218 login: vismithapysimplilearn
Password:
Last login: Wed Apr  6 08:52:57 on pts/262
[vismithapysimplilearn@ip-10-0-42-218 ~]$ nc -lk 4567
Hello Hello
This is my first program
```

Step 14: Go to terminal 2, and the real-time data you entered in terminal 1 will display as output

```
...     .start()
>>> 22/04/06 11:43:31 WARN sources.TextSocketSourceProvider: The socket source should not be used for production applications! It does not support recovery.
-------------------------------------------
Batch: 0
-------------------------------------------
+----+-----+
|word|count|
+----+-----+
+----+-----+

-------------------------------------------
Batch: 1
-------------------------------------------
+-----+-----+
| word|count|
+-----+-----+
|Hello|    2|
+-----+-----+

-------------------------------------------
Batch: 2
-------------------------------------------
+-------+-----+
|   word|count|
+-------+-----+
|program|    1|
|     is|    1|
|  Hello|    2|
|     my|    1|
|   This|    1|
|  first|    1|
+-------+-----+
```

© Copyright 2022, Simplilearn. All rights reserved.