Assisted Practice 12.4: Encapsulation

Problem Scenario: Write a program to demonstrate encapsulation using classes, objects, and methods.

Objective: In this demonstration, we will learn how to perform encapsulation.

Expected Output:

Accessing the protected member created in the parent class: 20

Accessing the modified protected member outside the class: 30

Accessing a protected member of the child class: 30

Accessing a protected member of the parent class: 20

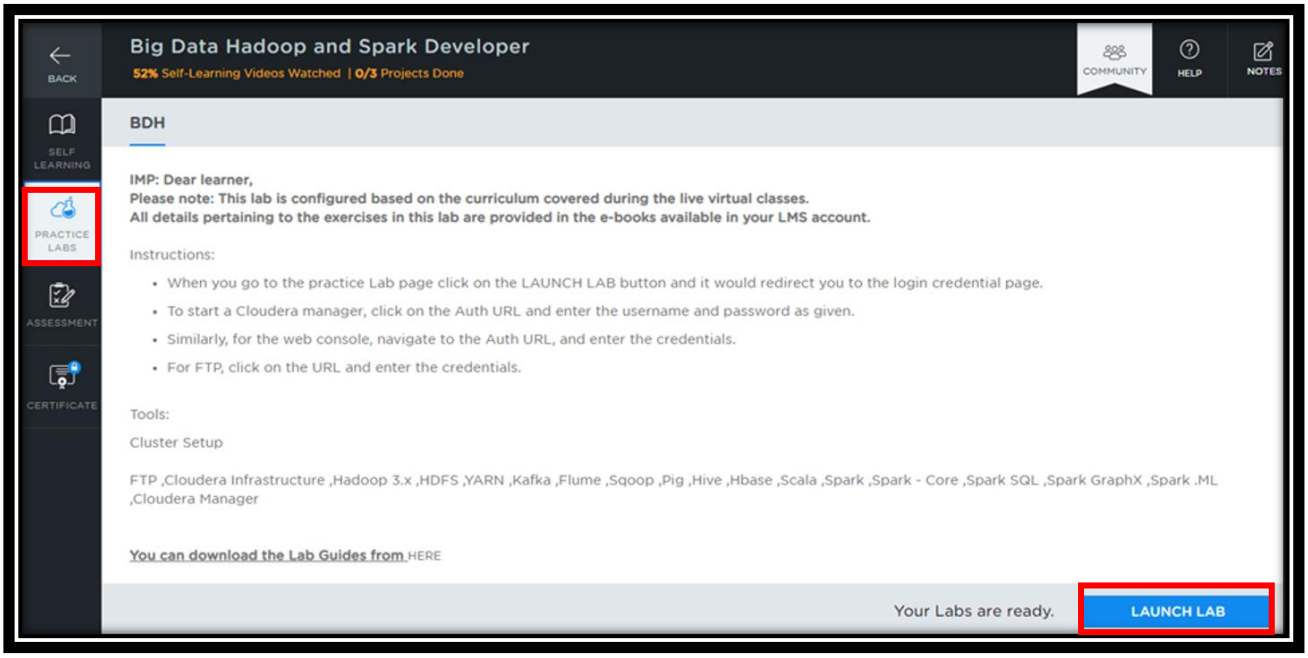
Steps to Perform:

Step 1: Log in to your LMS account

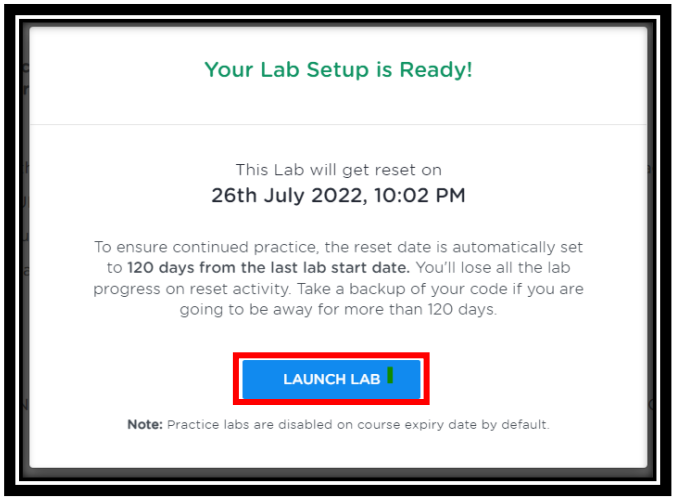
Step 2: Open the course "**Big data Hadoop and Spark developer**".

(Note: The course name reflects depending on the program purchased)

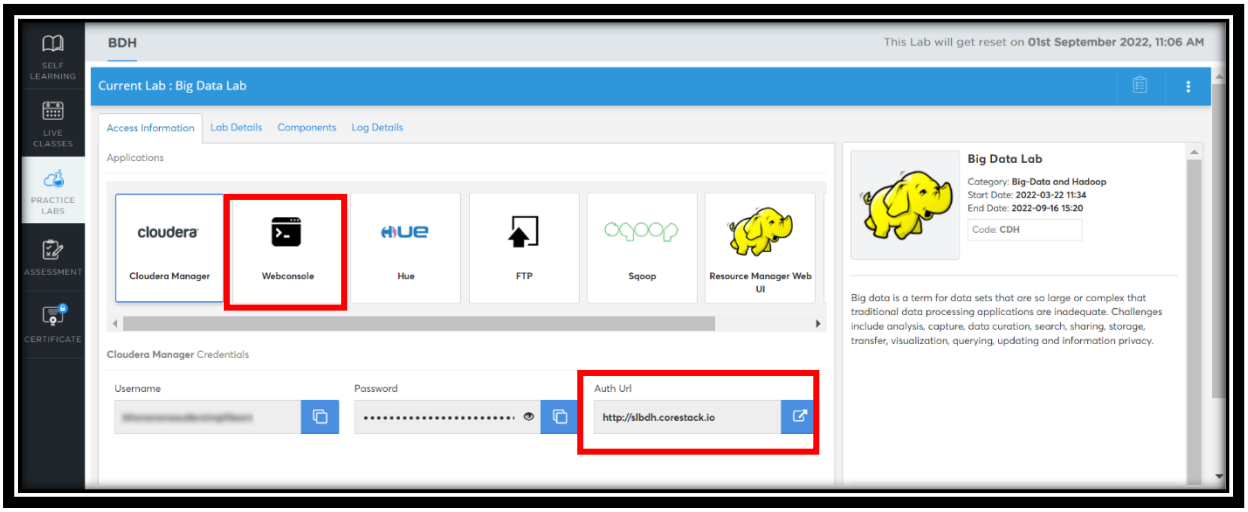
Step 3: On the left side, click on the "**PRACTICE LABS**" tab and click on the "**LAUNCH LAB**" button.



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.

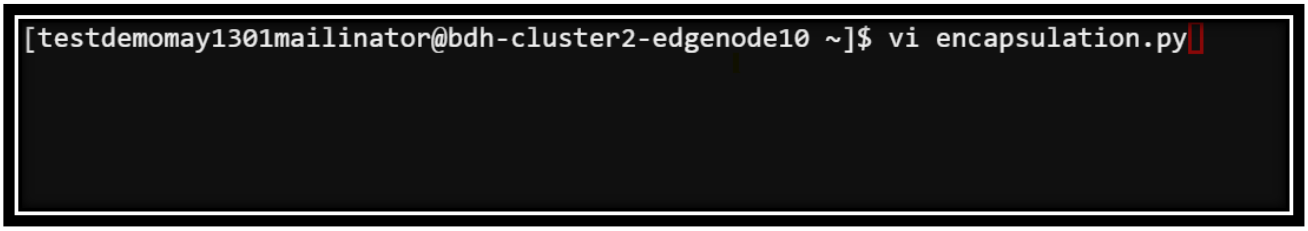
Step 7: Paste the "**Username**" and the "**Password**" on the console and click on enter.

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

Step 8: Create a python file**.**

Command:

vi encapsulation.py



The below screen appears:

Step 9: Perform the tasks.

9.1 Create a parent class with protected members.

Command:

```
class Parent:

    def __init__(self):

            self._number = 20
```

9.2 Create a child class that extracts the value of the protected members in the parent class.

Command:

```
class Child(Parent):

    def __init__(self):

            Parent.__init__(self)

            print("Accessing the protected member created in the parent class: ",self._number)
```

9.3 Modify the protected member in the derived class.

Command:

self._number = 30

print("Accessing the modified protected member outside the class: ",

self._number)

9.4 Create the objects of the parent and the child class.

Command:

obj1 = Child()

obj2 = Parent()

9.5 Print the protected member using the objects.

Command:

print("Accessing a protected member of the child class: ", obj1._number)

print("Accessing a protected member of the parent class: ", obj2._number)

```
class Parent:
        def __init__(self):
                self._number = 20
class Child(Parent):
        def __init__(self):

                Parent.__init__(self)
                print("Accessing the protected member created in the parent class: ",
                        self._number)

                self._number = 30
                print("Accessing the modified protected member outside the class: ",
                        self._number)


obj1 = Child()
obj2 = Parent()

print("Accessing a protected member of the child class: ", obj1._number)
print("Accessing a protected member of the parent class: ", obj2._number)

~
```

9.6 Run the code.

Command:

python3 encapsulation.py

```
[testdemomay1301mailinator@bdh-cluster2-edgenode10 ~]$ python3 encapsulation.py
Accessing the protected member created in the parent class:  20
Accessing the modified protected member outside the class:  30
Accessing a protected member of the child class:  30
Accessing a protected member of the parent class:  20
```