CloudLab- Kickstarter

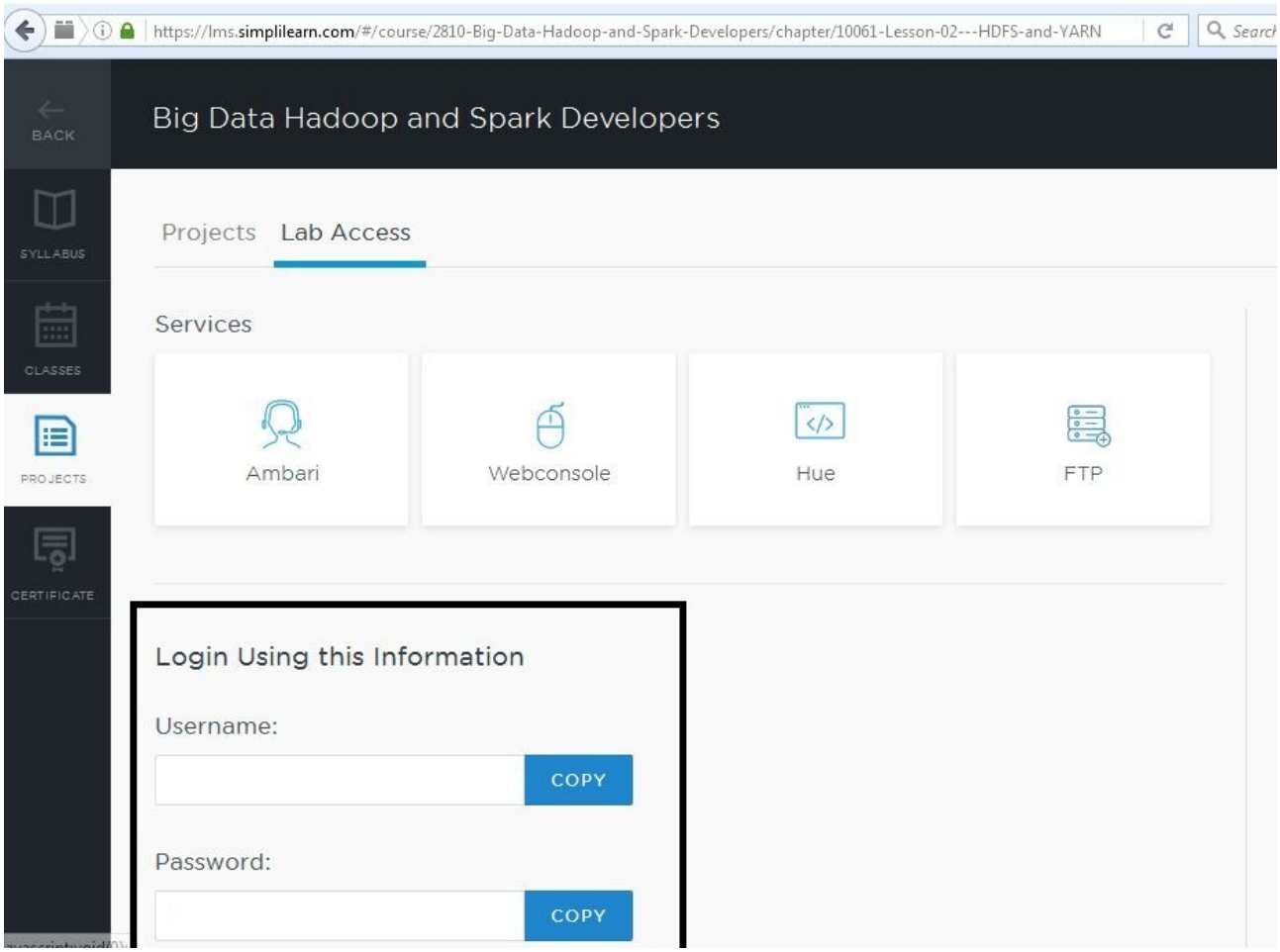
Our Big Data Hadoop course has a massive learner-base, with participants from different work experience years and background. Installation is a vital part of Hadoop, and installing its components separately is quite a tedious process, and is quite demanding for learners with limited experience in installations and administration.

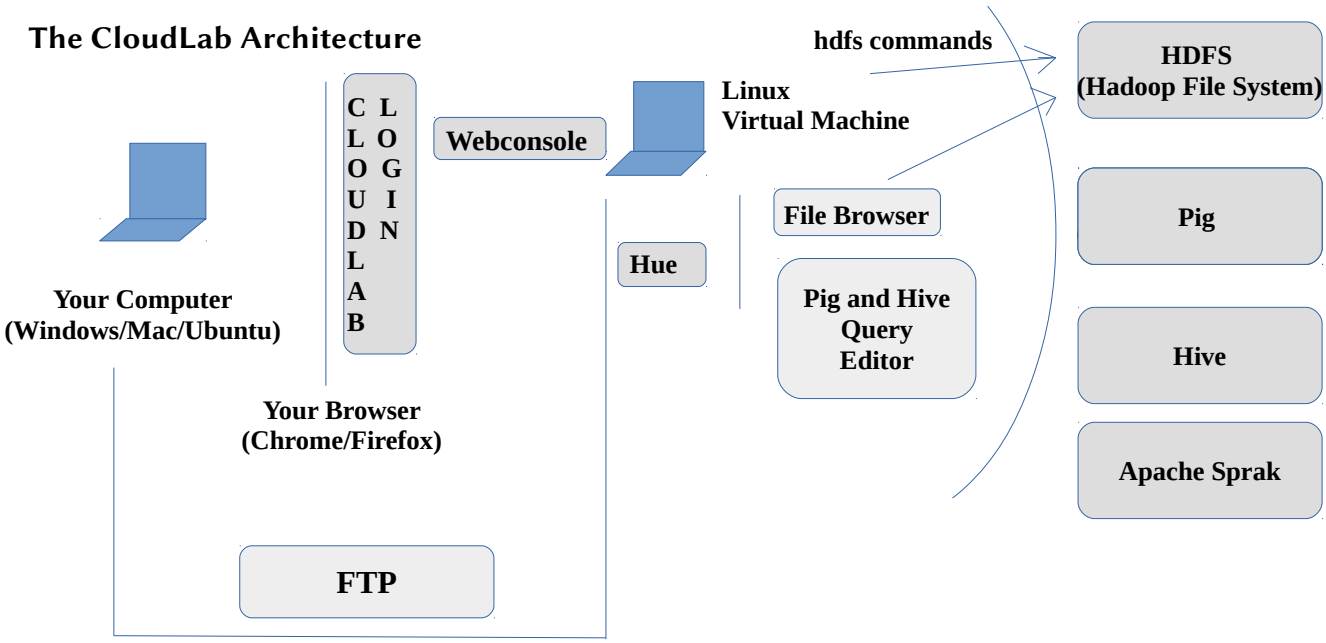
To help all our participants get past this roadblock and get straight to work, we have now created a fully functional multinode Hadoop cluster- CloudLab. Simplilearn's CloudLab will cater to the developer needs of Big Data Hadoop and helps you work with different Hadoop components like Pig, Hive, Sqoop, and Apache Spark.

Here's an introduction to CloudLab. Feel free to use this as a ready reckoner for all the Hadoop practice on the Lab.

Accessing CloudLab

On your LMS, head to the Projects>Lab Access tab and you can find CloudLab services, and the credentials required for logging in.





CloudLab is based on a Linux Virtual machine, on which Hadoop is installed, and hence its different services. You can access this Virtual Machine, by logging in to WebConsole. Create a file using the 'vi' command and you can see the file using the -ls command- just like any other Linux machine.
To move files to and from the Linux system you can use our FTP service. For a more detailed insight, refer the Community link below,

http://community.simplilearn.com/threads/what-is-ftp-and-hue.15024/

Services in CloudLab

- **Webconsole**- The Linux terminal of the system on which Hadoop is installed. Use the -ls command to access the local files, and the hdfs commands to access the HDFS files. The copyFromLocal command copies the file on the Linux machine to the HDFS.

- **FTP**- Move files from your computer to the Linux machine.

- **Hue**- Use the File Browser in Hue to get a User interface for the files in HDFS, and the Query editors for Pig and Hive for writing Pig and Hive queries.

- **Ambari**- Login to Ambari to view the Hadoop services running, and all the other cluster characteristics (Note: The jps command does not work in CloudLab. In its absence, you can use the Ambari console to view the cluster characteristics)

Here are the introductory commands for different services on CloudLab.

1. MySQL

The MySQL service runs in 172.31.54.174. Credentials are as below,
Username: labuser
Password: simplilearn

Use the below command to connect to MySQL

```
mysql -h 172.31.54.174 -u labuser -p simplilearn
```

If prompted for password again, type 'simplilearn' and hit Enter.

2. Pig and Hive

Simply type 'pig' or 'hive' to invoke the respective interactive shells, on the Webconsole. Use the Hue Query editor for User Interface for the Pig and Hive services.

3. Sqoop

Here's the very first Sqoop command you can try,

```
sqoop import --connect jdbc:mysql://172.31.54.174/db_name --driver
com.mysql.jdbc.Driver --username labuser --password simplilearn
--table table_name --m 1 --as-textfile --target-dir path/results
```

Replace the db_name, table_name and path/results variables and you're all good to go!

4. Beeline Hive

Type beeline to invoke the beeline shell. Once you enter, type the following connect command, to connect to Beeline,

```
!connect jdbc:hive2://172.31.56.102:10000
```

Use the below credentials to login,
Username: beeline
Password: simplilearn

5. Scala

To connect to the Scala interactive shell, set the following environment variables as below,

```
export SCALA_HOME=/usr/local/share/scala
export PATH=$PATH:$SCALA_HOME/bin
bash
```

And then you'll be able to run Scala, by using the 'scala' command.

6. Apache Spark

You can work with the Python shell of Apache Spark, by using the 'pyspark' command and Scala shell of Apache Spark by using the 'spark-shell' command.

7. Map Reduce

As we do not have an IDE yet, with our CloudLab, preparing and running a Java program is a roundabout process. However, for the curious, here is the step by step process as to how you can run the Map Reduce program with CloudLab.
Preparing a .jar will involve including the classpath (path where Java development dependencies can be found).
As many users use CloudLab, the relevant paths, need to be set manually to run jar files on CloudLab.

If you prepare your .jar on Eclipse in your local machine, the classpath file points to a path that looks like "C:\\Program Files\...". As this cannot be found on CloudLab, your Main class cannot be located.
To avoid this error, kindly bundle your jar on CloudLab itself and run the same.

Here are the steps for executing a MapReduce java program.

1. Make the directory java_resources using 'mkdir' command on the Webconsole and copy all the dependencies (.jar) there- client-core.jar, client-commons.jar, logging.jar

2. Create the .java file- **first.java**. (or upload the .java file on to your Linux system using FTP.)

3. Run below commands, to Set the following paths on CloudLabs,

```
export JAVA_HOME=$JAVA_HOME:/usr/jdk64/jdk1.8.0_60/bin
export PATH=$PATH:/usr/jdk64/jdk1.8.0_60/bin
```

4. Follow these steps to bundle your .java file into a .jar.

- Compile your program including the .jar dependencies
```
javac -classpath ":java_resources/client-core.jar:java_resources/client-commons.jar" MyJavaFile.java
```
(add paths of other dependencies if any)

- Add main class to the manifest
```
echo Main-Class: first >manifest.txt
```

- Create the jar bundle
```
jar cvfm first.jar manifest.txt *.class
```

- Using the Hadoop environment run your program,
```
hadoop jar first.jar first
```

Also, please mention the path as simply 'file.txt', as the Home path is already set within your Linux environment.
Starting a path with '/' will lead to access issues, as you do not have access to root directory.

Please note:
Also, if you face any version mismatch or errors due to dependencies client common and client core jars (or any other packages you might have used), please upload those jar files, note down their paths on your Linux system, using the pwd command, and add it to the .classpath file in the jar.

Pitfall!

Permission denied error on trying to create a directory.

On Webconsole, you are within a Linux environment and a '/' means the root directory in Linux. If you are trying to create a directory with as mkdir /simplilearn, you will not be able to create that, as you do not have the required permissions.

Kindly remove the '/' and use the following command structure to create a directory,

```
mkdir dir1

mkdir dir1/dir2
```

Alright then! Now you're all equipped, we wish you all the very best for your hands-on practice and projects.
If you hit any roadblocks, look out for the same in the Simplitalk community, so you can find your answers right away!

Do reach us on the Help & Support section, with a screenshot/traceback/log, if necessary, if you happen to face more hurdles.

All the very best!