Lab 1

Introduction

This is a warm-up lab. In this lab, you will log into NYU's Peel Hadoop Cluster and try out basic HDFS and Spark/Scala commands.

Objectives

- Familiarize yourself with NYU's Hadoop cluster.
- Practice basic HDFS commands.
- Familiarize yourself with the Spark shell.

Preparation

You need to run this lab in NYU's Peel Hadoop Cluster. If you enrolled in this course before February 1, an account has been created on your behalf. Please check your email for the account information that was sent to you on February 2. Otherwise, please request an account as soon as possible.

Your tasks

Step 1: Connect to the Peel cluster

Please follow the instructions on the Peel Cluster User Guide to connect to the Peel cluster.

Step 2: Try out HDFS commands

Please try issuing these commands:

- `hadoop fs -ls /` – to see the contents of the top-level directory in HDFS

- `hadoop fs -ls` – to see the contents of your user directory
- `hadoop fs -ls /user/yourNetID` – same as above
- `hadoop fs -mkdir lab1` – to create a new directory named `lab1` in your user directory
- `hadoop fs -ls` – to verify that you now have a directory called `lab1`

After that, try creating a text file in your scratch directory (say, `/scratch/yourNetID/data.txt`). Your scratch directory is a good staging area for Big Data files because the quota is 5TB. Note that files in the scratch directory that are not accessed for 60 days will be automatically deleted. By contrast, files in your home directory will not be deleted, but the quota is only 50GB. You can find the specifications of various storage systems in the Peel Cluster here.

Next, try putting the file to HDFS by issuing:

- `hadoop fs -put /scratch/yourNetID/data.txt lab1` – to put `data.txt` to `lab1`
- `hadoop fs -ls lab1` – to verify that you have put `data.txt` to `lab1`
- `hadoop fs -cat lab1/data.txt` – to verify the content of `lab1/data.txt`

Finally, try removing the directory:

- `hadoop fs -rm -r lab1` – to remove directory `lab1`
- `hadoop fs -ls` – to verify that you have successfully removed the directory called `lab1`

You don't have to follow the exact procedure, and you are encouraged to try out more commands (*e.g.*, `get`, `mv`, `tail`). Here is a great reference on HDFS commands.

Step 3: Try out the Spark shell

Type one of the following commands to start the Spark shell - you shouldn't see any errors (warnings can be ignored):

- `spark-shell --deploy-mode client` or
- `spark-shell --deploy-mode client --driver-java-options=-Dscala.color` – to enable syntax coloring (syntax coloring is turned on by default in Spark 3.0, but the Peel Cluster runs Spark 2.4)

After some output from the shell, you should see a `scala>` prompt. Then, try the following commands (`[TAB]` means the Tab key):

- `:help`
- `sc.[TAB]` – to see the commands available in the Spark Context (the REPL has tab-completions)
- `sc.version` – to get the version of Spark that is running in the shell
- `val myConstant: Int = 2437`
- `myConstant`
- `myConstant.[TAB]`
- `myConstant.to[TAB]`
- `myConstant.toFloat`
- `myConstant` – note that myConstant has not changed; it's still an Int
- `myConstant.toFloat.toInt`
- `val myString = myConstant.toString` – note the type inferred for myString
- `:type myString` – use the :type command to see the type that is inferred for myString
- `:q` – to quit the Spark shell (alternatively, you can press Ctrl-D)

Again, you don't have to follow the exact procedure, and you are encouraged to try out more commands.

Submission

To receive full credit, please compress all of the following items into a single file named `YourName_NetID_lab1.zip` and upload it to NYU Brightspace:

- Screenshots to show that you have connected to the Peel cluster and tried out the HDFS and Spark/Scala commands.

Tips

Please use Discord if you experience any difficulties. The graders and I will help you get your environment working.

Don’t procrastinate. The Hadoop cluster tends to get crowded near the due date.