Compute Cluster

We have a relatively modest Hadoop cluster for this course, based on Cloudera *(http://www.cloudera.com/)* express: 6 nodes, 96 cores, 660GB memory, 11TB storage.

Connecting Remotely

The goal here is to connect to `gateway.sfucloud.ca` by SSH.

Option 1: just get it working

You will be connecting to the cluster a lot: you may want to get things set up more nicely to make your life easier later. But, this should at least *work*.

You generally just need to SSH to `gateway.sfucloud.ca` (substituting whatever SSH method you use on your computer):

```
[yourcomputer]$ ssh <USERID>@gateway.sfucloud.ca
[gateway]$
```

Once you're connected to the Hadoop gateway, you can start running `spark-submit` (and `hdfs`) commands.

If you need access to the web frontends in the cluster, you can do the initial SSH with a much longer command including a bunch of port forwards:

```
ssh -L 50070:master.sfucloud.ca:50070 -L 8088:master.sfucloud.ca:8088 <USERID>@gateway.sf
```

Option 2: the slick way

Create an SSH key *(https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys--2)* (if you don't have one already) so you can log in without a password. Then copy your public key into `.ssh/authorized_keys` on the server (with `ssh-copy-id` or by copying to `~/.ssh/authorized_keys`).

Create (or add to) the `~/.ssh/config` file on your computer. With this config, you can simply `ssh gateway.sfucloud.ca` to connect. (bonus: tab-completion)

```
Host gateway.sfucloud.ca
  User <USERID>
  LocalForward 8088 master.sfucloud.ca:8088
  LocalForward 50070 master.sfucloud.ca:50070
```

With this configuration, port forwards will let you connect (in a limited unauthenticated way) to the web interfaces:

- HDFS namenode: http://localhost:50070/ *(http://localhost:50070/)*
- YARN application master: http://localhost:8088/ *(http://localhost:8088/)*

Copying Files

You will also frequently need to copy files to the cluster:

```
[yourcomputer]$ scp code.py <USERID>@gateway.sfucloud.ca:
```

Or whatever your preferred SCP/SFTP method is.

Spark Applications

In order to get your environment set up correctly, you need to start **every session** with this command:

```
module load 353
```

That sets environment variables enabling the newest version of Spark, and tells it to use Python 3 to run jobs, not 2.

Since you'll likely forget this occasionally, I suggest starting Spark jobs with the provided SparkSkeleton which will fail quickly when you forget.

Once you have the code there, you can start jobs as usual with `spark-submit`, and they will be sent to the cluster:

```
spark-submit code.py ...
```

Cleaning Up

If you have unnecessary files sitting around (especially large files created as part of an assignment), please clean them up with a command like this:

```
hdfs dfs -rm -r output*
```

It is possible that you have jobs running and consuming resources without knowing: maybe you created an infinite loop or otherwise have a job burning memory or CPU. You can list jobs running on the cluster like this:

```
yarn application -list
```

And kill a specific job:

```
yarn application -kill <APPLICATION_ID>
```

Updated Fri Aug. 28 2020, 11:21 by ggbaker.