Cloudera Quickstart Docker Image

For this class we will be using a modified version of Cloudera's Quickstart Docker image, available here: https://hub.docker.com/r/gdancik/cloudera

To run Hadoop, we will use the command below (in linux, including a \ tells the shell that the command continues onto the next line):

```
docker run --hostname=quickstart.cloudera --privileged=true -it \
-p 8888:8888 -p 8088:8088 -p 8042:8042 gdancik/cloudera \
/usr/bin/docker-quickstart
```

This will create a new container and run the *docker-quickstart* script, which starts various services required for Hadoop (such as Hadoop data node, Hadoop name node, Hive, and more). The hostname and extended privileges are required for Hadoop. The *-p* argument maps a port from the container to the local host. This allows us to access web services running inside the container, such as Hue (port 8888), the YARN Web API (port 8088), and the YARN Node manager (port 8042). When creating the container, Hadoop requires setting the hostname and that the container has extended privileges, which is granted in the command above.

More details about running the Cloudera Hadoop in a docker container can be found here: https://hub.docker.com/r/cloudera/quickstart/

Running PySpark

Installing and running PySpark locally through Python

To download PySpark for use outside of docker, see https://pypi.org/project/pyspark/

Running regular python, you will need to create a Spark Context by executing the following commands:

```
import pyspark
sc = pyspark.SparkContext(appName="myAppName")
```

Running pyspark inside docker

Simply type *pyspark* at the command prompt. However, you must execute the command below (from the terminal) so that *PySpark* will use Python 3 instead of Python 2. To automatically execute this command when creating a container, add it to your */root.bashrc* file. The command is below:

```
export PYSPARK_PYTHON=python3
```

If autocomplete does not work, run the following from within PySpark:

```
import rlcompleter, readline
readline.parse_and_bind(“tab: complete”)
```