CPSC 4330 Big Data Analytics

In-Class Exercise 1 – Using HDFS

Getting Started

1. Sign up for a Docker ID from https://hub.docker.com/signup, if you don't have a docker account yet.

2. Download and install Docker on your computer. You can download Docker from https://docs.docker.com/get-started/.

3. Start Docker and log in using your docker account.

4. After Docker is completely started, open a command prompt or bash window, and run the command

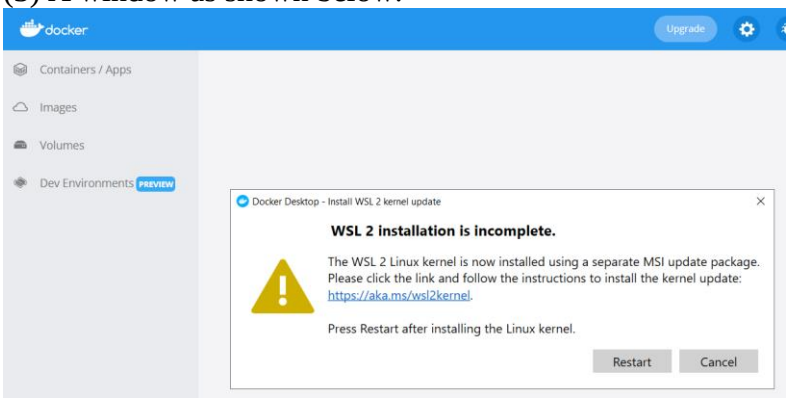
```
docker pull linli2021/testimage:fullimage
```

to pull the class image from Docker Hub.

Note: for Windows user, if you see following issue(s), try the suggested solutions below.
(1) The docker daemon is not running
The error message will be something like "error during connect…. the docker client must be run elevated to connect… The docker daemon is not running".
Solution: Start "task manager" in Windows as an administrator, go to the service tab, and find the service for docker. Start the service and make sure the service is running.

(2) An error message like "Error response from daemon: open \\.\pipe\docker_engine_linux: The system cannot find the file specified".
Solution: This message means that the docker is not completely started yet. If you see "docker engine starting" in Docker, it means that the docker is still starting.

(3) A window as shown below.



Solution: Follow the instructions in the prompt window to install the kernel update.

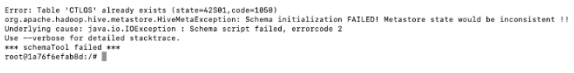
5. To run the image in a container (a runnable instance of an image) for the 1st time, run the command

```
docker run -it --name bda linli2021/testimage:fullimage bash
```

"-it" means using an interactive mode.
"--name" is to give a name to the container. In this sample command, the name of the container is "bda" (you can change to a different name if you like).
"linli2021/testimage:fullimage" is the image name and the tag.
"bash" means to use bash command.

When the container starts, it prompts "Re-format filesystem in Storage Directory root= /tmp/hadoop-root/dfs/name; location= null ? (Y or N)". Type "N" choosing not to re-format filesystems.

Then you will see messages in the screenshot below. The error messages do not disturb using the VM. It will take a few seconds (or even a few minutes if your computer is slow) to finish starting the VM.



6. If you want to stop the container, type "exit" in the docker terminal.

To start an *exited* container in an interactive mode, run the command

```
docker start -i bda
```

where "bda" is the container name that you set in step 5.

Exercise

In this exercise, you will begin to get acquainted with the Hadoop tools. You will manipulate files in HDFS, the Hadoop Distributed File System.

Hadoop is already installed, configured, and running on your virtual machine. Most of your interaction with the system will be through a command-line wrapper called *hadoop*. If you run this program with no arguments, it prints a help message. To try this, run the following command in a terminal window.

```
#hadoop
```

The *hadoop* command is subdivided into several subsystems. For example, there is a subsystem for working with files in HDFS and another for launching and managing MapReduce processing jobs.

Step 1: Exploring HDFS

The subsystem associated with HDFS in the Hadoop wrapper program is called FsShell. This subsystem can be invoked with the command *hadoop fs*.

1. In the docker terminal window, enter:

   ```
   # hadoop fs
   ```

   You should see a help message describing all the commands associated with the FsShell subsystem.

2. Enter

   ```
   #hadoop  fs  –ls   /
   ```

   This shows you the contents of the root directory in HDFS. There are two entries initially (/tmp and /user).

   Note that the directory structure in HDFS has nothing to do with the directory structure of the local filesystem; they are completely separate namespaces.

Step 2: Uploading Files

Besides browsing the existing filesystem, another important thing you can do with FsShell is to upload new data into HDFS.

1. How to copy files from file system on your computer (host machine) to docker (VM).
   - If you are on Mac, open a terminal, run

     ```
     docker cp  /Users/linli/Documents/data   bda:/hadoop-data/
     ```

where "/Users/linli/Documents/data" is the data directory on my computer (substitute it with your own file directory), "bda:/hadoop-data/" means the "hadoop-data" folder in the "bda" container. If you have a different container name, substitute "bda" with your container name. If you want to copy files to a different folder in your docker image, substitute "hadoop-data" with your preferred folder name.

- If you are on Windows, open a command terminal, run

```
docker cp  C:\Users\lin\Documents\data  bda:/hadoop-data
```

See the instruction above for mac users to find how to modify the two command arguments following "docker cp".

- To copy files from docker (VM) to your computer (host machine), use the same command "docker cp", but the 1st argument is the container name and the file directory on VM (e.g. bda:/hadoop-data/ex1.txt), and the 2nd argument is the file directory on your computer (e.g. /Users/linli/Documents/data).

2. Download the file "shakespeare.tar.gz" from Canvas to your computer (host machine).

3. Copy the file "shakespeare.tar.gz" from your computer to Docker. For example,

```
docker cp  /Users/linli/Documents/data/shakespeare.tar.gz     bda:/hadoop-data/
```

4. On docker, enter the directory where the file "shakespeare.tar.gz" is located. Run the command

```
#cd hadoop-data
```

5. Unzip *shakespeare.tar.gz* by running:

```
# tar zxvf shakespeare.tar.gz
```

This creates a directory named *shakespeare/* containing several files on your local filesystem.

6. Create a folder on HDFS for organizing all data used for exercises.

```
# hadoop  fs -mkdir  /exercise
```

7. Copy the files in "*shakespeare*" into HDFS

```
# hadoop fs –put  shakespeare  /exercise
```

This copies the local *shakespeare* directory and its contents into a remote, HDFS directory named /exercise/shakespeare .

8. List the contents of your HDFS directory to see if the files are copied successfully

```
#hadoop fs  -ls  /exercise/shakespeare
```

9. We will also need a sample web server log file, which we will put into HDFS for use in future exercises. This file is currently compressed using GZip. Rather than extract the file to the local disk and then upload it, we will extract and upload in one step. First, create a directory in HDFS in which to store it:

```
#hadoop fs  -mkdir  /exercise/weblog
```

10. Download the file "access_log.gz" from Canvas, and copy it to Docker.

```
docker cp  /Users/linli/Documents/data/access_log.gz    bda:/hadoop-data/
```

11. Extract and upload the file in one step. The *–c* option to *gunzip* uncompresses to standard output, and the dash (-) in the *hadoop fs –put* command takes whatever is being sent to its standard input and places that data in HDFS.

```
# gunzip -c access_log.gz | hadoop  fs  -put  -  /exercise/weblog/access_log
```

12. Run the *hadoop fs –ls /exercise/weblog* command to verify that the log file is in your HDFS directory.

13. The access log file is large – around 500 MB. Create a smaller version of this file, consisting only of its first 5000 lines, and store the smaller version in HDFS. You can use the smaller version for testing in subsequent exercises.

```
# hadoop fs  -mkdir  /exercise/testlog
#gunzip -c  access_log.gz  | head  -n  5000 |  hadoop fs  -put  -  /exercise/testlog/test_access_log
```

Step 3: Viewing and Manipulating Files

Now let's view some of the data you just copies into HDFS.

1. Enter:

```
# hadoop fs  -ls  /exercise/shakespeare
```

This lists the contents of the */exercise/shakespeare* HDFS directory, which consists of the files *comedies*, *glossary*, *histories*, *poems*, and *tragedies*.

2. The *glossary* file included in the compressed file you began with is not strictly a work of Shakespeare, so let's remove it:

```
#hadoop fs  –rm  /exercise/shakespeare/glossary
```

Note that you could leave this file in place if you so wished. If you did, then it would be included in subsequent computations across the works of Shakespeare, and would skew your results slightly. As with many real-world big data problems, you make trade-offs between the labor to purify your input data and the precision of your results.

3. Enter

```
#hadoop fs  -cat  /exercise/shakespeare/histories  | tail  -n  50
```

This prints the last 50 lines of the document "histories" to your docker terminal. This command is handy for viewing the output of MapReduce programs. Very often, an individual output file of a MapReduce program is very large, making it inconvenient to view the entire file in the terminal. For this reason, it's often a good idea to pipe the output of the *fs –cat* command into *head, tail, more,* or *less.*

4. To download a file to work with on the local filesystem on Docker, use the *fs –get* command. This command takes two arguments: an HDFS path and a local path. It copies the HDFS contents into the local directory (on Docker) where you run the command:

```
#hadoop fs  -get  /exercise/shakespeare/poems  shakepoems.txt

#cat shakepoems.txt
```