Tutorial 2: Working with HDFS

CN7031 - Big Data Analytics

Dr Amin Karami (a.karami@uel.ac.uk), October 2023

LEARNING OUTCOMES:

- **Understand HDFS Operations:** Gain a comprehensive understanding of HDFS operations.
- **Proficient HDFS Administration:** Develop skills in administering and managing HDFS.
- **Problem-Solving and Troubleshooting HDFS Issues:** Solve complex problems related to HDFS, including appending content to files, setting time-to-live values, and working with block locations.

Tasks:

1- Create two directories called "data" and “data_copy” in HDFS

```
hdfs dfs -mkdir -p /data

hdfs dfs -mkdir -p /data_copy
```

-p: creates parent directories if they do not exist

2- Upload a file from the local file system to the "/data" directory in HDFS

```
hdfs dfs -put -f /home/cloudera/eclipse/about.html /data
```

-f: overwrites the destination if it already exists

3- List the contents of the "/data" directory in HDFS

```
hdfs dfs -ls /data
```

4- Copy a file from one location in HDFS to another location within HDFS

```
hdfs dfs -cp -f /data/about.html /data_copy/about_2.html
```

5- Delete a file from HDFS

```
hdfs dfs -rm -f /data_copy/about_2.html
```

-f: ignores non-existent files and does not prompt for confirmation

-r: deletes directories recursively

6- Display the content of a file stored in HDFS

```
hdfs dfs -cat /data/about.html
```

7- Set the replication factor of a file in HDFS to 2.

```
hdfs dfs -setrep 2 /data/about.html
```

-w: wait until the replication is done

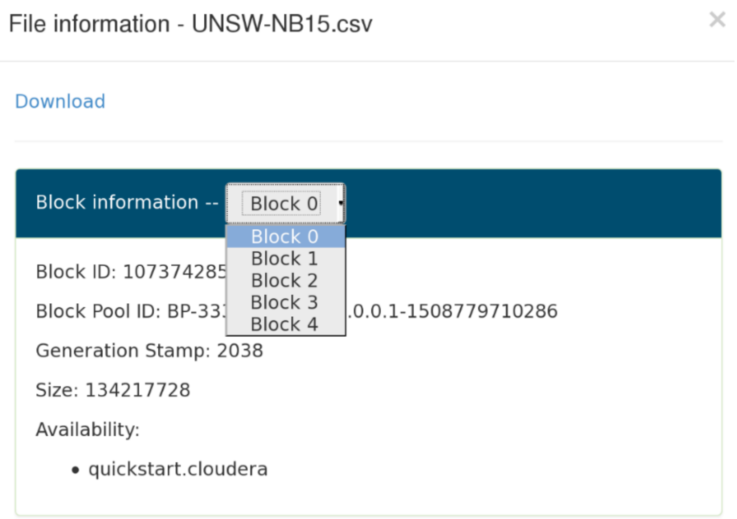
-R: set the replication factor of all files and subdirectories

8- Transfer/Put the “UNSW-NB15.csv” (~600MB) data to HDFS

Firstly, download it from here, and locate it in the Desktop. Then, go through the following syntaxes:

```
hdfs dfs -mkdir -p /data/DataAnalysis/

hdfs dfs -put -f /home/cloudera/Desktop/UNSW-NB15.csv
/data/DataAnalysis/
```



File information - UNSW-NB15.csv

Block ID: 107374285

Block Pool ID: BP-33 .0.0.1-1508779710286

Generation Stamp: 2038

Size: 134217728

Availability:

- quickstart.cloudera

9- View the disk usage of a directory named "data" in HDFS

```
hdfs dfs -du -h /data
```

-du: display the disk usage of a directory

-h: display the sizes in a human-readable format (e.g. 1K, 234M, 2G, etc.)

10- Set a storage policy named "hot" for a directory in HDFS

```
hdfs storagepolicies -setStoragePolicy -policy hot -path /data/DataAnalysis/
```

The "hot" storage policy is typically used for files that are frequently accessed and require low-latency access times. This can be useful when you want to optimize the performance of frequently accessed files in HDFS.

- COLD: This policy is used for files that are rarely accessed and can tolerate high-latency access times.
- WARM: This policy is used for files that are semi-frequently accessed and require moderate-latency access times.
- ALL_SSD: This policy is used for files that require the highest level of performance and are stored on solid-state drives (SSD).
- ONE_SSD: This policy is used for files that require high performance and are stored on a single SSD.
- HOT_AND_COLD: This policy is used for files that have both frequently and rarely accessed data, and require both low and high-latency access times.

11- Check the number of blocks for a file, block size, block sequence, block names and block locations.

```
hadoop fsck /data/DataAnalysis -files -blocks -locations
```

"fsck": stands for "file system consistency check"

12- Set a time-to-live (TTL) value to 30 days for files in "data" directory

```
hdfs dfs -setfattr -n user.hdfs.ttl -v 2592000000 /data
```

-n: the name of the attribute being set is "user.hdfs.ttl"

-v: the value being set

note: The value is specified in milliseconds, so 30 days is equal to 30 * 24 * 60 * 60 * 1000 milliseconds.

We start by multiplying 30 days by 86,400 (24 * 60 * 60) seconds per day:

30 x 86,400 = 2,592,000 seconds

Next, we multiply that result by 1,000 milliseconds per second to get the milliseconds:

2,592,000 x 1,000 = 2,592,000,000 milliseconds

13- Get and print the time-to-live (TTL) value for “data” folder

```
hdfs dfs -getfattr -d /data
```

-d: display the values of all extended attributes associated with the file or directory specified ("/data" in this case)

14- At the end, clean up HDFS

```
hdfs dfs -rm -r -f /data
```

```
hdfs dfs -rm -r -f /data_copy
```