Understanding Block Storage

In this Lab

Objective: To understand how data is partitioned into blocks and stored in HDFS

During this Lab: Perform the following steps

Lab Files: `~/data`

<img src="https://user-images.githubusercontent.com/558905/40613898-7a6c70d6-624e-11e8-9178-7bde851ac7bd.png" align=“left” width=“50” height=“50” title=“ToDo Logo” />
<h4>1. Put the File into HDFS</h4>

<br>
If not already done, open a Terminal.

Use less (or more) to view the contents of the `stocks.csv` file. Press `q` when you are finished to exit.

```
less stocks.csv
```

Try putting the file into HDFS with a block size of 30 bytes:

```
hdfs dfs -D dfs.blocksize=30 -put stocks.csv
```

> *Note that a size of 30 bytes is not a valid blocksize. The blocksize needs to be at least 1,048,576 according to the* `dfs.namenode.fs-limits.min-block-size property`

```
put: Specified block size is less than configured minimum value (dfs.namenode.fs-limits.min-block-size): 30 < 1048576
```

Try the put again, but use a block size of 2,000,000:

```
hdfs dfs -D dfs.blocksize=2000000 -put stocks.csv
```

> *Note that 2,000,000 is not a valid blocksize because it is not a multiple of 512 (the checksum size)*

Try the put again, but this time use 1,048,576 for the blocksize:

```
hdfs dfs -D dfs.blocksize=1048576 -put stocks.csv
```

This time the put command should have worked. Use ls to verify that the file is in HDFS in the `/user/[user-name]` folder:

```
hdfs dfs -ls
```

So now you should see something like this:

```
Found 1 items
-rw-r--r--  1 root root   3613198   stocks.csv
```

<!–STEP–>

<img src="https://user-images.githubusercontent.com/558905/40613898-7a6c70d6-624e-11e8-9178-7bde851ac7bd.png" align=“left” width=“50” height=“50” title=“ToDo Logo” />
<h4>2. View the Number of Blocks</h4>

<br>

Run the following command to view the number of blocks that were created for `stocks.csv`:

```
hdfs fsck /user/[user-name]/stocks.csv
```

There are four blocks. Look for the following line in the output:

```
Total blocks (validated):4 (avg. block size 903299 B)
```

<!–STEP–>

<img src="https://user-images.githubusercontent.com/558905/40613898-7a6c70d6-624e-11e8-9178-7bde851ac7bd.png" align=“left” width=“50” height=“50” title=“ToDo Logo” />
<h4>3. Find the Actual Blocks</h4>

<br>
Enter the same `fsck` command as before, but add the `-files` and `-blocks` options:

```
hdfs fsck /user/[user-name]/stocks.csv -files -blocks
```

> *Note the output contains the block IDs, which are coincidentally the names of the files on the DataNodes*

Run the command again, but this time add the -locations flag:

```
hdfs fsck /user/[user-name]/stocks.csv -files -blocks -locations
```

> *Note in the output that the IP address of the DataNode appear next to each block.*

Change directories to the following:

```
cd /hadoop/hdfs/data/current/BP-xxx/current/finalized/
```

Replace `BP-xxx` with the actual folder name. To finish this, use the `TAB` key to complete the filename once you have typed `B`. Then finish typing the rest of the directory path.

Try and find the folder that contains the blocks you are looking for and change directories into that folder. The easiest way is to look at the timestamps and find the most recently changed folder. You can use the `stat \*` command to view the contents of the directory, then use `ll` to list the contents of that directory.

```
stat *
cd <most recently created directory - for example, subdir0
>
ll
```

<!–STEP–>

<img src="https://user-images.githubusercontent.com/558905/40613898-7a6c70d6-624e-11e8-9178-7bde851ac7bd.png" align=“left” width=“50” height=“50” title=“ToDo Logo” />
<h4>Important</h4>

<br>
If the results of the `ll` command are additional subdirectories rather than block information (as shown in the next lab step), repeat the process above to once again find the newest directory created , change

to it, and list its contents.

Confirm that the actual blocks appear in this folder. Look for files that are exactly 1,048,576 bytes. These are three of the blocks.

```
-rw-r--r--  1   hdfs    hdfs    1048576 blk_1073742090
-rw-r--r--  1   hdfs    hdfs    8199    blk_1073742090_1266.meta
-rw-r--r--  1   hdfs    hdfs    1048576 blk_1073742091
-rw-r--r--  1   hdfs    hdfs    8199    blk_1073742091_1267.meta
-rw-r--r--  1   hdfs    hdfs    467470  blk_1073742093
-rw-r--r--  1   hdfs    hdfs    3663    blk_1073742093_1269.meta
```

> *Note that the fourth block is smaller: 467,470 bytes.*

You can view the contents of a block (although this is not a typical task in Hadoop). Here is the tail of the second block:

```
tail blk_1073741905
```

Output:

```
NYSE,XKK,2007-08-20,9.51,9.64,9.30,9.51,4700,7.17
NYSE,XKK,2007-08-17,9.30,9.99,9.26,9.57,3900,7.21
NYSE,XKK,2007-08-16,9.45,10.00,8.11,9.05,23400,6.82
```

```
NYSE,XKK,2007-08-15,9.51,9.51,9.18,9.35,4900,7.04
NYSE,XKK,2007-08-14,9.52,9.52,9.51,9.51,1100,7.17
NYSE,XKK,2007-08-13,9.60,9.60,9.56,9.56,3000,7.20
NYSE,XKK,2007-08-10,9.82,9.82,9.60,9.60,2500,7.23
NYSE,XKK,2007-08-09,9.83,9.87,9.82,9.82,4500,7.40
NYSE,XKK,2007-08-08,9.45,9.90,9.45,9.66,6000,7.28
NYSE,XKK,2007-08-07,9.25,9.50,9.25,9.40
```

> *Note: the last record in this file is not complete and spills over to the next block, a common occurrence in HDFS.*

Result

You are finished. Now you see more about hdfs!