Using HDFS Commands

About This Lab

Objective: To become familiar with how files are added to and removed from HDFS, and how to view files in HDFS

Successful outcome: You will have added and deleted several files and folders in HDFS

File locations: `~/data`

Steps

<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. View the `hdfs dfs` Command</h4>

1. Open a Terminal window (if you do not have one open already).

2. From the command line, enter the following command to view the usage of hdfs dfs:

```
hdfs dfs
```

> *Note the usage contains options for performing file system tasks in HDFS, like copying files from a local folder into HDFS, retrieving a file from HDFS, copying and moving files around, and making and removing directories. In this lab, you will perform these commands and many others, to help you become comfortable with working with HDFS.*

<!-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. Create a Directory in HDFS</h4>

1. Enter the following `-ls` command to view the contents of the user's root directory in HDFS, which is `/user/[user-name]`:

```
hdfs dfs –ls
```

You do not have any files in `/user/[user-name]` yet, so no output is displayed.

2. Run the `-ls` command again, but this time specify the root HDFS folder:

```
hdfs dfs -ls /
```

The output should look something like:

<img width=“511” alt=“using-hdfs-commands-1” src="https://user-images.githubusercontent.com/21102559/41181153-c34727d0-6b3e-11e8-98a1-17377515b0c3.png">

> *Note: see how adding the `/` in the `-ls` command caused the contents of the root folder to display, but leaving off the `/` showed the contents of `/user/[user-name]`, which is the user root’s home directory on hadoop. If you do not provide the path for any hadoop commands, the user’s home on hadoop is assumed.*

3. Enter the following command to create a directory named `test` in HDFS:

```
hdfs dfs -mkdir test
```

4. Verify the folder was created successfully:

```
hdfs dfs –ls
```

Output:

```
Found 1 items
drwxr-xr-x  - root root 0   test
```

5. Create two subdirectories of `test`:

```
hdfs dfs -mkdir test/test1
hdfs dfs -mkdir -p test/test2/test3
```

6. Use the `-ls` command to view the contents of `/user/[user-name]`:

```
hdfs dfs –ls
```

> *Note you only see the test directory. To recursively view the contents of a folder, use `-ls -R`:*

```
hdfs dfs -ls –R
```

Output:

```
drwxr-xr-x  - root root 0   test
drwxr-xr-x  - root root 0   test/test1
drwxr-xr-x  - root root 0   test/test2
drwxr-xr-x  - root root 0   test/test2/test3
```

<!–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. Delete a Directory</h4>

1. Delete the `test2` folder (and recursively its subcontents) using the `-rm -R` command:

```
hdfs dfs -rm -R test/test2
```

2. Now run the `-ls -R` command:

```
hdfs dfs -ls –R
```

Output:

```
.Trash
.Trash/Current
.Trash/Current/user
.Trash/Current/user/[user-name]
.Trash/Current/user/[user-name]/test
.Trash/Current/user/[user-name]/test/test2
.Trash/Current/user/[user-name]/test/test2/test3 test
```

> *Note Hadoop created a* `.Trash` *folder for the root user and moved the deleted content there. The* `.Trash` *folder empties automatically after a configured amount of time.*

<!–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>4. Upload a File to HDFS</h4>

1. Now put a file into the test folder. Change directories to the `data` directory.

2. Notice this folder contains a file named `data.txt`:

```
tail data.txt
```

3. Run the following `-put` command to copy `data.txt` into the `test` folder in HDFS:

```
hdfs dfs -put data.txt test/
```

4. Verify the file is in HDFS by listing the contents of test:

```
hdfs dfs -ls test
```

The output should look like the following:

<!–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>5. Copy a File in HDFS</h4>

1. Now copy the `data.txt` file in test to another folder in HDFS using the `-cp` command:

```
hdfs dfs -cp test/data.txt test/test1/data2.txt
```

2. Verify the file is in both places by using the `-ls -R` command on test.

```
hdfs dfs -ls -R test
```

Output:

```
-rw-r--r-- 3 root root 1529355 test/data.txt
drwxr-xr-x - root root 0       test/test1
-rw-r--r-- 3 root root 1529355 test/test1/data2.txt
```

3. Now delete the `data2.txt` file using the `-rm` command:

```
hdfs dfs -rm test/test1/data2.txt
```

4. Verify the `data2.txt` file is in the `.Trash` folder.

<!–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>6. View the Contents of a File in HDFS</h4>

1. You can use the `-cat` command to view text files in HDFS.

2. Enter the following command to view the contents of data.txt:

```
hdfs dfs -cat test/data.txt
```

3. You can also use the `-tail` command to view the end of a file:

```
hdfs dfs -tail test/data.txt
```

> *Note the output this time is only the last 20 rows of `data.txt`.*

<!–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>7. Getting a File from HDFS</h4>

1. See if you can figure out how to use the `get` command to copy `test/data.txt` from HDFS into your local `/tmp` folder.

<!–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>8. The `getmerge` Command</h4>

1. Put the file `small_blocks.txt` into the `test` folder in HDFS. You should now have two files in test: `data.txt` and `small_blocks.txt`.

2. Run the following getmerge command:

```
hdfs dfs -getmerge test /tmp/merged.txt
```

3. What did the previous command do? Open the file `merged.txt` to see what happened?

<!–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>9. Specify the Block Size and Replication Factor</h4>

1. Put `data.txt` into `/user/[user-name]` in HDFS, giving it a blocksize of “1m” (one megabyte). The blocksize is defined using the `dfs.blocksize` property on the command line:

```
hdfs dfs -D dfs.blocksize=1m -put data.txt data.txt
```

2. Run the following `fsck` command on `data.txt`:

```
hdfs fsck /user/[user-name]/data.txt
```

3. How many blocks are there for this file?

Result

You should now be comfortable with executing the various HDFS commands, including creating directories, putting files into HDFS, copying files out of HDFS, and deleting files and folders.

<button type=“button”><a href="https://virtuant.github.io/hadoop-overview-spark-hwx/">Go Back</a></button>

<br>

<br>