Using HDFS

Objective: Begin to get acquainted with Hadoops file system. And manipulate files in HDFS, the Hadoop Distributed File System.

Exercise directory: `~/data`

HDFS paths: `/user/[user-name]`

In this tutorial you will begin to get acquainted with Hadoop. You will manipulate files in HDFS,
the Hadoop Distributed File System. We will walk through many of the common of the basic Hadoop
Distributed File System (HDFS) commands you will need to manage files on HDFS.
The particular datasets we will utilize to learn HDFS file management are from San Francisco salaries.

We will download data onto our local filesystems. The commands are tailored for mac and linux users.

Open a Terminal on AWS or your Local Machine

Using either Putty or your SSH client.

> *Note: if you're on AWS or VMware or Azure, insert your appropriate*

> *IP address in place of 127.0.0.1. Azure users will need to replace port 2222 with 22*

<h4>1. Copy and Paste Sample Data</h4>

Either get the several of latest years from san francisco or get some earlier ones from us:

```
    wget https://transcal.s3.amazonaws.com/public/export/san-francisco-2017.csv
    wget https://transcal.s3.amazonaws.com/public/export/san-francisco-2016.csv
```

Manipulate the Data

- Create a Directory in HDFS, Upload a file and List Contents
- Find Out Space Utilization in a HDFS Directory
- Download Files From HDFS to Local File System
- Explore Two Advanced Features
- Use Help Command to Access Hadoop Command Manual

<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>

Let’s learn by writing the syntax. You will be able to copy and paste the following example commands into your terminal. Let’s login under hdfs user, so we can give `root` user permission to perform file operations:

```
sudo su hdfs
```

We will use the following command to run filesystem commands on the file system of hadoop:

```
hdfs dfs [command_operation]
```

This affects the permissions of the folder or file. Controls who has read/write/execute privileges. We will give root access to read and write to the user directory. Later we will perform an operation in which we send a file from our local filesystem to hdfs.

```
hdfs dfs -chmod 777 /user
hdfs dfs -chmod 777 /user/[user-name]
```

>  *If the directory isn’t there, then do a* `hdfs dfs -mkdir /user/student` *and the re-chmod it.*

> *Warning: in production environments, setting the folder with the permissions above is not a good idea because anyone can read/write/execute files or folders.*

Type the following exit command, so we can switch back to the base user. We can perform the remaining file operations under the user folder since the permissions were changed.

```
exit
```


<h4>3. Make a HDFS Directory</h4>

Takes the path URI's as an argument and creates a directory or multiple directories:

```
hdfs dfs -mkdir <paths>
```

Do the following:

```
hdfs dfs -mkdir salaries
```

 What just happened? Do you know were `salaries` is? If not, do a `hdfs dfs -ls -R` and look for the `salaries` directory.

<h4>4. Now Copy Data into HDFS</h4>

Copies single src file or multiple src files from local file system to the Hadoop Distributed File System (HDFS).

```
   hdfs dfs -put <local-src> ... <HDFS_dest_path>
```

Do the following:

```
   hdfs dfs -put [source file].csv   /user/[user-name]/sal
aries/[source file].csv
   hdfs dfs -put [source file 2].csv /user/[user-name]/sal
aries/[source file 2].csv
```


<h4>5. Now List the Directory</h4>

Lists the contents of a directory - for a file, returns stats of a file.

```
hdfs dfs  -ls  <args>
```

Do the following:

```
hdfs dfs -ls /user/[user-name]
hdfs dfs -ls /user/[user-name]/salaries
```

You can also do the commands in shorhand as the `hadoop` user:

```
sudo su [user-name]
cd
```

So now you the following:

```
hdfs dfs -ls /
hdfs dfs -ls salaries
```

And now exit the `hadoop` user:

```
exit
```

<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. Find Out Space Utilization in A HDFS Directory</h4>

Displays size of files and directories contained in the given directory or the size of a file if its just a file.

```
   hdfs dfs -du URI
```

Now do:

```
   hdfs dfs -du  /user/[user-name]/ /user/[user-name]/sala
ries/[source file].csv
```


<h4>7. Now Download files from HDFS to the Local File System</h4>

This copies/downloads files from HDFS to the local file system:

```
   hdfs dfs -get <hdfs_src> <localdst>
```

So do it:

```
    hdfs dfs -get /user/[user-name]/salaries/[source file].
csv /home/
```


<h4>8. Advanced Features</h4>

Takes a source directory file or files as input and concatenates files in src into the local destination file. It concatenates files in the same directory or from multiple directories as long as we specify their location and outputs them to the local file system, as can be seen in the usage below.

Let’s concatenate the San Francisco salaries from two separate directories and output them to our local filesystem. Our result will be the salaries from one year which are appended below the last row of the other.

```
    hdfs dfs -getmerge <src> <localdst> [addnl]
    hdfs dfs -getmerge <src1> <src2> <localdst> [addnl]
```

> *Note: can also be set to enable adding a newline on end of each file:*

```
    hdfs dfs -getmerge /user/[user-name]/salaries/ /root/ou
tput.csv
```

Merges the files in your input files to output.csv in the root directory of the local filesystem. In our example, the first file contained about 120,000+ rows and the second file contained almost 30,000 rows. This file operation is important because it will save you time from having to manually concatenate them.

<h4>9. Recursive Copying</h4>

Copy file or directories recursively, all the directory's files and subdirectories to the bottom of the directory tree are copied.

It is a tool used for large inter/intra-cluster copying:

```
    hdfs dfs -cp <src-url> <dest-url>
```

Now do:

```
    hdfs dfs -cp /user/[user-name]/salaries/ /user/[user-na
me]/[some other name]
```

-cp: copies and all their contents to [some other dir]

Now, verify the files or directories successfully copied to the destination folder:

```
hdfs dfs -ls /user/[user-name]/[some other dir]/
```

Visual result of distcp file operation. Notice that both src1 and src2 directories and their contents were copied to the dest directory.

<h4>10. Other Commands</h4>

There are several other commands associated with the FsShell subsystem which let you perform most common filesystem manipulations: `rm, rm -r (recursive rm), mv, cp, mkdir`, etc.

For example:

```
hdfs dfs -cat glossary | tail -n 50
```

This prints the last 50 lines of the glossary to your 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.

Help

Help command opens the list of commands supported by Hadoop Data File System (HDFS):

```
hdfs dfs -help
```

Results

Congratulations! We just learned to use commands to manage our San Francisco dataset in HDFS. We learned to create, upload and list the the contents in our directories. We also acquired the skills to download files from HDFS to our local file system and explored a few advanced features of HDFS file management using the command line.

<button type=“button”>Go Back</button>

