Listing Files

This is how you inspect HDFS to see what it contains. Use these commands to find files and their associated metadata.

`hadoop fs -ls ./example`	List files in a directory
`hadoop fs -ls ./example/matthew*`	List files matching a pattern
`hadoop fs -ls -h ./example/`	List files with human-friendly file sizes (eg 1.3M vs 1331325)
`hadoop fs -ls -R ./example/`	Recursively list files in this directory AND all child directories.
`hadoop fs -ls -d ./example/`	List files, but with directories shown as files. So in this case it will show info about the example folder itself.

Uploading/Downloading Files

Moving files between HDFS and the local filesystem and back.

`hadoop fs -put ./localfile.txt ./example/`	Upload a file from your local machine to a specific directory on HDFS.
`hadoop fs -put -f ./localfile.txt ./example/`	Upload a file and overwrite any existing file on HDFS.
`hadoop fs -put -l ./localfile.txt ./example/`	Upload a file and set a replication factor of 1 (you probably shouldn't ever really use this).
`hadoop fs -get ./example/remotefile.txt ./`	Download a file from HDFS to your local machine.
`hadoop fs -get -p ./example/remotefile.txt ./`	Download a file from HDFS to your local machine, preserving metadata (eg modified time).
`hadoop fs -get ./example/*.txt ./`	Download a set of files that match a pattern to your local machine.

Reading & Writing Files

Reading file contents without downloading the file itself.

`hadoop fs -text ./example/file.txt`	Print the contents of a file to the terminal, decompressing if nessecery.
`hadoop fs -cat ./example/*.txt`	Print to the terminal the contents of all files that match the provided pattern. Note - this will NOT decompress like 'text' will.
`hadoop fs [-cat,-text] -ignoreCrc ./example/*.txt`	As above, but disable the verification checksum.
`hadoop fs -appendToFile ./localfile.txt ./example/remotefile.txt`	Append the contents of a local file to a file on HDFS. This is only supported in Hadoop versions 2.1.1+

File Management

Organize your files with these commands.

`hadoop fs -mv ./example/f1.txt ./example/f2.txt`	Move a file to a different file/directory (omit the filename to name it the same)
`hadoop fs -cp ./example/f1.txt ./example/f2.txt`	copy a file to a different file/directory (omit the filename to name it the same)
`hadoop fs -rm ./example/f1.txt`	Delete a file (sends it to the trash)
`hadoop fs -rm -skipTrash ./example/f1.txt`	Actually delete the file. No trash.
`hadoop fs -rm -r ./example/directory`	Recursively delete a directory and it's contents
`hadoop fs -touchz ./example/somefile`	Create a zero-length file (great for creating _SUCCESS files).
`hadoop fs -touchz ./example/somefile`	Create a zero-length file (great for creating _SUCCESS files).

HDFS Administration

Check on important stuff that is less about your files and more about HDFS.

`hadoop fs -df -h ./example`	Show capacity and used space of the filesystem. Will show partition space remaining if you have partitions.
`hadoop fs -du -h ./example/*.txt`	show the amount of space used by matching files
`hadoop fs -expunge`	Empty the Trash (useful if you -rm without -skipTrash)
`hadoop fs -chown owner:group ./example`	Change ownership of a file (use -R for the directory)
`hadoop fs -chmod 0700 ./example/file.txt`	Change the mode of the file (eg to 0700)
`hadoop fs -checksum ./example/*.txt`	Fetch checksum information for the matching files (requires a datanode roundtrip, slow and intensive).