# HDFS Commands

## Overview

The File System (FS) shell includes various shell-like commands that directly interact with the Hadoop Distributed File System (HDFS) as well as other file systems that Hadoop supports, such as Local FS, HFTP FS, S3 FS, and others.

The FS shell is invoked by:

```
bin/hdfs dfs <args>
```

All FS shell commands take path URIs as arguments. The URI format is scheme://autority/path.

For HDFS the scheme is hdfs, and for the Local FS the scheme is file. The scheme and authority are optional. If not specified, the default scheme specified in the configuration is used. An HDFS file or directory such as /parent/child can be specified as hdfs://namenodehost/parent/child or simply as /parent/child (given that your configuration is set to point to hdfs://namenodehost).

Most of the commands in FS shell behave like corresponding Unix commands. Differences are described with each of the commands. Error information is sent to stderr and the output is sent to stdout.

1. **mkdir**

Usage: hdfs dfs -mkdir <paths>

Takes path uri's as argument and creates directories. The behavior is much like unix mkdir -p creating parent directories along the path.

Example:

- hdfs dfs -mkdir /user/username/dir1

- hdfs dfs -mkdir hdfs://bigdata:8020/user/username/dir2

2. **put**

Usage: hdfs dfs -put hdfsdata.txt /user/username/dir1

Copy single src, or multiple srcs from local file system to the destination file system. Also reads input from stdin and writes to destination file system.

Example:

- hdfs dfs -put hdfsdata2.txt  hdfsdata3.txt /user/username/dir1

- hdfs dfs -put - /user/username/dir1/hdfsdata5.txt
  Reads the input from stdin.
  CTRL + D to terminate

3. **cat**

Usage: hdfs dfs -cat URI [URI …]

Copies source paths to stdout.

Example:

- hdfs dfs -cat /user/username/dir1/hdfsdata5.txt

• hdfs dfs -cat file:/home/username/file1/file1.txt /user/username/dir1/hdfsdata5.txt

It will show both the local and hdfs file contents.

4. **get**
Usage: hdfs dfs -get [-ignorecrc] [-crc] <src> <localdst>

Copy files to the local file system. Files that fail the CRC check may be copied with the -ignorecrc option. Files and CRCs may be copied using the -crc option.

Cyclic redundancy checking is a method of checking for errors in data that has been transmitted on a communications link (accidental changes to raw data).

Example:
• hdfs dfs -get /user/username/dir1/hdfsdata5.txt hdfsdata_to_local_file1.txt

5. **rm**
Usage: hdfs dfs -rm [-skipTrash] URI [URI …]

Delete files specified as args. Only deletes files. If the -skipTrash option is specified, the trash, if enabled, will be bypassed and the specified file(s) deleted immediately. This can be useful when it is necessary to delete files from an over-quota directory. Refer to rmr for recursive deletes.

Example:
• hdfs dfs -rm /user/username/dir1/hdfsdata4.txt._COPYING_

Here we delete the file hdfsdata4.txt._COPYING_

6. **copyFromLocal**
Usage: hdfs dfs -copyFromLocal <localsrc> URI

Similar to *put* command, except that the source is restricted to a local file reference.

Example:
• hdfs dfs -copyFromLocal file1/file1.txt /user/username/dir2

Copy the local file file1.txt to hdfs in Directory dir2, the destination file name also will be the same as file1.txt

7. **copyToLocal**
Usage: hdfs dfs -copyToLocal [-ignorecrc] [-crc] URI <localdst>

Similar to *get* command, except that the destination is restricted to a local file reference.

Example:
• hdfs dfs -copyToLocal /user/username/dir2/file1.txt file1/new.txt

8. **cp**
Usage: hdfs dfs -cp URI [URI …] <dest>

Copy files from source to destination. This command allows multiple sources as well in which case the destination must be a directory.

Example:
• hdfs dfs -cp /user/username/dir1/data6.txt /user/username/dir2/newfile.txt

• hdfs dfs -cp /user/username/dir1/data6.txt /user/username/dir1/data7.txt /user/username/dir2

We can copy multiple files at same time, but in this case we need to only provide the directory names.

9. **moveFromLocal**
Usage: dfs -moveFromLocal <localsrc> <dst>

Similar to *put* command, except that the source localsrc is deleted after it's copied.

Example:
• hdfs dfs -moveFromLocal file1/file1.txt /user/username/dir2/movedfile.txt

10. **mv**
Usage: hdfs dfs -mv URI [URI …] <dest>

Moves files from source to destination. This command allows multiple sources as well in which case the destination needs to be a directory. Moving files across file systems is not permitted.

Example:
• hdfs dfs -mv /user/username/dir2/movedfile.txt /user/username/dir1/movedfile.txt

We can move multiple file also at the same time, in this case we need to provide the name of the directory only
A syntax of the command is given below.

• hdfs dfs -mv /user/username/file1 /user/username/file2 /user/username/directoryname

11. **ls**
Usage: hdfs dfs -ls <args>

For a file returns stat on the file with the following format:
permissions number_of_replicas userid groupid filesize modification_date modification_time filename

For a directory it returns list of its direct children as in unix. A directory is listed as:
permissions userid groupid modification_date modification_time dirname

Example:
- hdfs dfs -ls /user/username/dir1

12. **lsr**
Usage: hdfs dfs -ls -R <args>

Recursive version of ls. Similar to Unix ls -R.

Example:
- hdfs dfs -ls -R /user/username/dir1

13. **getmerge**
Usage: hdfs dfs -getmerge <src> <localdst> [addnl]

Takes a source directory and a destination file as input and concatenates files in src into the destination local file. Optionally add nl can be set to enable adding a newline character at the end of each file.

Example:
- hdfs dfs -getmerge /user/username/dir2/data6.txt file1/file1.txt

14. **rmr**
Usage: hdfs dfs -rm -r [-skipTrash] URI [URI …]

Recursive version of delete. The rmr command recursively deletes the directory and any content under it. If the -skipTrash option is specified, the trash, if enabled, will be bypassed and the specified file(s) deleted immediately. This can be useful when it is necessary to delete files from an over-quota directory.

Example:
- hdfs dfs -rm -r /user/username/dir3

15. **setrep**
Usage: hdfs dfs -setrep [-R] <path>

Changes the replication factor of a file. -R option is for recursively increasing the replication factor of files within a directory.
Example:

- hdfs dfs -setrep 4 /user/username/dir1/data6.txt

Replication 4 set: /user/username/dir1/data6.txt

16. **count**
Usage: hdfs dfs -count [-q] <paths>

Count the number of directories, files and bytes under the paths that match the specified file pattern.

The output columns with -count are:
DIR_COUNT, FILE_COUNT, CONTENT_SIZE FILE_NAME

The output columns with -count -q are:

QUOTA, REMAINING_QUATA, SPACE_QUOTA, REMAINING_SPACE_QUOTA, DIR_COUNT, FILE_COUNT, CONTENT_SIZE, FILE_NAME

Example:

- hdfs dfs -count /user/username/dir1

- hdfs dfs -count -q /user/username/dir1

17. **du**

Usage: hdfs dfs -du [-s] [-h] URI [URI …]

Displays sizes of files and directories contained in the given directory or the length of a file in case it's just a file.

Options:

- The -s option will result in an aggregate summary of file lengths being displayed, rather than the individual files.

- The -h option will format file sizes in a "human-readable" fashion (e.g 64.0m instead of 67108864)

Example:

- hdfs dfs -du /user/username/dir1

- hdfs dfs -du -s /user/username/dir1

- hdfs dfs -du -h /user/username/dir1

18. **stat**

Usage: hdfs dfs -stat URI [URI …]

Returns the stat information on the path.

Example:

- hdfs dfs -stat /user/username/dir1

19. **tail**

Usage: hdfs dfs -tail [-f] URI

Displays last kilobyte of the file to stdout.

Example:

- hdfs dfs -tail /user/username/dir1/hdfsdata.txt

20. **touchz**

Usage: hdfs dfs -touchz URI [URI …]

Create a file of zero length.

Example:

- hdfs dfs -touchz /user/username/dir1/create.txt

Admin Commands:

21. **chgrp**
Usage: hdfs dfs -chgrp [-R] GROUP URI [URI …]
Change group association of files. With -R, make the change recursively through the directory structure. The user must be the owner of files, or else a super-user.

22. **chmod**
Usage: hdfs dfs -chmod [-R] <MODE[,MODE]... | OCTALMODE> URI [URI …]
Change the permissions of files. With -R, make the change recursively through the directory structure. The user must be the owner of the file, or else a super-user.

23. **chown**
Usage: hdfs dfs -chown [-R] [OWNER][:[GROUP]] URI [URI ]
Change the owner of files. With -R, make the change recursively through the directory structure. The user must be a super-user.

For Additional Information on above 3 commands, Refer

https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsPermissionsGuide.html