LABSHEET-1

HDFS File System Commands

.

Using the command below, we can get a list of FS Shell commands:

$ hadoop fs -help

user@ubuntu1:~$ hadoop fs -help

Example: hadoop fs [generic options]

[-appendToFile ...]

[-cat [-ignoreCrc] ...]

[-checksum ...]

[-chgrp [-R] GROUP PATH...]

[-chmod [-R] PATH...]

[-chown [-R] [OWNER][:[GROUP]] PATH...]

[-copyFromLocal [-f] [-p] ...]

[-copyToLocal [-p] [-ignoreCrc] [-crc] ...]

[-count [-q] ...]

[-cp [-f] [-p | -p[topax]] ...]

[-createSnapshot []]

[-deleteSnapshot]

[-df [-h] [...]]

[-du [-s] [-h] ...]

[-expunge]

[-get [-p] [-ignoreCrc] [-crc] ...]

[-getfacl [-R]]

[-getfattr [-R] {-n name | -d} [-e en]]

[-getmerge [-nl]]

[-help [cmd ...]]

[-ls [-d] [-h] [-R] [...]]

[-mkdir [-p] ...]

[-moveFromLocal ...]

[-moveToLocal]

[-mv ...]

[-put [-f] [-p] ...]

[-renameSnapshot]

[-rm [-f] [-r|-R] [-skipTrash] ...]

[-rmdir [--ignore-fail-on-non-empty]

...]

[-setfacl [-R] [{-b|-k} {-m|-x }]|[--set]]

[-setfattr {-n name [-v value] | -x name}]

[-setrep [-R] [-w] ...]

[-stat [format] ...]

[-tail [-f]]

[-test -[defsz]]

[-text [-ignoreCrc] ...]

[-touchz ...]

[-Example [cmd ...]]

Most of the commands that we use on an HDFS environment are listed as above, from this thorough list of commands we will take a look at some of the most important commands with examples. Let us take a look into the commands with examples:

1. mkdir:

is used to create a directory on an HDFS environment.

Options:

–p	mention not to fail if the directory already exists.

Syntax:

$ hadoop fs -mkdir [-p]

example:

$ hadoop fs -mkdir /user/hadoop/

$ hadoop fs -mkdir /user/data/

In order to create subdirectories, the parent directory must exist. If the condition is not met then, 'No such file or directory' message appears

2. ls:

it is used for listing the directories present under a specific directory in an HDFS system. The –lsr command may be used for the recursive listing of the directories and files under a specific folder.

options:

–d	The option is used to list the directories as plain files
–h	The option is used to format the sizes of files into a human-readable manner than just number of bytes
–R	The option is used to recursively list the contents of directories

Syntax:

$ hadoop fs -ls [-d] [-h] [-R]

Example:

$ hadoop fs -ls /

$ hadoop fs -lsr /

The command above will match the specified file pattern, and directory entries are of the form (as shown below)

Output:

permissions - userId groupId sizeOfDirectory(in bytes) modificationDate(yyyy-MM-dd HH:mm) directoryName''

3. put:

This command is used to copy files from the local file system to the HDFS filesystem. This command is similar to –copyFromLocal command. This command will not work if the file already exists unless the –f flag is given to the command. This overwrites the destination if the file already exists before the copy

Option:

–p	The flag preserves the access, modification time, ownership and the mode

Syntax:

$ hadoop fs -put [-f] [-p] ...

Example:

$ hadoop fs -put sample.txt /user/data/

4. get:

This command is used to copy files from HDFS file system to the local file system, just the opposite to put command.

Syntax:

$ hadoop fs -get [-f] [-p]

Example:

$ hadoop fs -get /user/data/sample.txt workspace/

5. cat:

This command is similar to the UNIX cat command and is used for displaying the contents of a file on the console.

Example:

$ hadoop fs -cat /user/data/sampletext.txt

6. cp:

This command is similar to the UNIX cp command, and it is used for copying files from one directory to another directory within the HDFS file system.

Example:

$ hadoop fs -cp /user/data/sample1.txt /user/hadoop1

$ hadoop fs -cp /user/data/sample2.txt /user/test/in1

7. mv:

This command is similar to the UNIX mv command, and it is used for moving a file from one directory to another directory within the HDFS file system.

Example:

$ hadoop fs -mv /user/hadoop/sample1.txt /user/text/

8. rm:

This command is similar to the UNIX rm command, and it is used for removing a file from the HDFS file system. The command –rmr can be used to delete files recursively.

Options:

–rm	Only files can be removed but directories can't be deleted by this command
–rm r	Recursively remove directories and files
–skipTrash	used to bypass the trash then it immediately deletes the source
–f	mention that if there is no file existing
–rR	used to recursively delete directories

Syntax:

$ hadoop fs -rm [-f] [-r|-R] [-skipTrash]

Example:

$ hadoop fs -rm -r /user/test/sample.txt

9. getmerge:

This is the most important and the most useful command on the HDFS filesystem when trying to read the contents of a MapReduce job or PIG job's output files. This is used for merging a list of files in a directory on the HDFS filesystem into a single local file on the local filesystem.

Example:

$ hadoop fs -getmerge /user/data

10. setrep:

This command is used to change the replication factor of a file to a specific count instead of the default replication factor for the remaining in the HDFS file system. It is a directory then the command will recursively change the replication factor of all the residing files in the directory tree as per the input provided.

Options:

–w	used to request the command to wait for the replication to be completed

–R	used to accept for backward capability and has no effect

Syntax:

$ hadoop fs -setrep [-R] [-w]

Example:

$ hadoop fs -setrep -R /user/hadoop/

11. touchz:

This command can be used to create a file of zero bytes size in HDFS filesystem.

Example:

$ hadoop fs -touchz URI

12. test:

This command is used to test an HDFS file's existence of zero length of the file or whether if it is a directory or not.

options:

–d	used to check whether if it is a directory or not, returns 0 if it is a directory
–e	used to check whether they exist or not, returns 0 if the exists
–f	used to check whether there is a file or not, returns 0 if the file exists
–s	used to check whether the file size is greater than 0 bytes or not, returns 0 if the size is greater than 0 bytes
–z	used to check whether the file size is zero bytes or not. If the file size is zero bytes, then returns 0 or else returns 1.

Example:

$ hadoop fs -test -[defsz] /user/test/test.txt

13. expunge:

This command is used to empty the trash available in an HDFS system.

Syntax:

$ hadoop fs –expunge

Example:

user@ubuntu1:~$ hadoop fs –expunge

17/10/15 10:15:22 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.

14. appendToFile:

This command appends the contents of all the given local files to the provided destination file on the HDFS filesystem. The destination file will be created if it is not existing earlier.

Syntax:

$ hadoop fs -appendToFile

Example:

user@ubuntu1:~$ hadoop fs -appendToFile derby.log data.tsv /in/appendfile

user@ubuntu1:~$ hadoop fs -cat /in/appendfile

Sun Oct 15 14:41:10 IST 2017 Thread[main,5,main] Ignored duplicate property derby.module.dataDictionary in jar:file:/home/user/Downloads/apache-hive-0.14.0-bin/lib/hive-jdbc-0.14.0-standalone.jar!/org/apache/derby/modules.properties

Sun Oct 15 14:41:10 IST 2017 Thread[main,5,main] Ignored duplicate property derby.module.lockManagerJ1 in jar:file:/home/user/Downloads/apache-hive-0.14.0-bin/lib/hive-jdbc-0.14.0-standalone.jar!/org/apache/derby/modules.properties

Sun Oct 15 14:41:10 IST 2017 Thread[main,5,main] Ignored duplicate property derby.env.classes.dvfJ2 in jar:file:/home/user/Downloads/apache-hive-0.14.0-bin/lib/hive-jdbc-0.14.0-standalone.jar!/org/apache/derby/modules.properties

15. tail:

This command is used to show the last 1KB of the file.

option:

–f	used to the show appended data as the file grows

Syntax:

$ hadoop fs -tail [-f]

Example:

user@tri03ws-386:~$ hadoop fs -tail /in/appendfile

Sun Oct 15 14:41:10 IST 2017:

Booting Derby version The Apache Software Foundation - Apache Derby - 10.10.1.1 - (1458268): instance a816c00e-0149-e638-0064-0000093808b8 on database directory /home/user/metastore_db with class loader sun.misc.Launcher$AppClassLoader@3485def8

Loaded from file:/home/user/Downloads/apache-hive-0.14.0-bin/lib/derby-10.10.1.1.jar

java.vendor=Oracle Corporation

java.runtime.version=1.7.0_65-b32

user.dir=/home/user

os.name=Linux

os.arch=amd64

os.version=3.13.0-39-generic

derby.system.home=null

Database Class Loader started - derby.database.classpath=''

16. Stat:

This command is used to print the statistics about the file/directory in the specified format. Format accepts file size in blocks (%b), the group name of the owner (%g) and the file name (%n), block size (%o), replication (%r), the username of the owner (%u), modification date (%y, %Y)

Syntax:

$ hadoop fs -stat [format]

Example:

user@tri03ws-386:~$ hadoop fs -stat /in/appendfile

2014-11-26 04:57:04

user@tri03ws-386:~$ hadoop fs -stat %Y /in/appendfile

1416977824841

user@tri03ws-386:~$ hadoop fs -stat %b /in/appendfile

20981

user@tri03ws-386:~$ hadoop fs -stat %r /in/appendfile

1

user@tri03ws-386:~$ hadoop fs -stat %o /in/appendfile

134217728

17. setfattr:

This command sets an extended attribute name and value for a file or directory on the HDFS filesystem.

Options:

–n	used to provide the extended attribute name
–x	used to remove the extended attribute, file or directory
–v	used to provide the extended attribute value.

There are 3 different encoding methods available for the value.

- If the argument is enclosed in double quotes, then the value is the string inside the quotes
- If the argument is prefixed with 0x or 0X, then it is taken as a hexadecimal number
- If the argument begins with 0s or 0S, then it is taken as a Base64 encoding

Syntax:

$ hadoop fs -setfattr {-n name [-v value] | -x name}

18. df:

This command is used to show the capacity, free and used space available on the HDFS filesystem. If the filesystem has multiple partitions and if there is no path is mentioned to any specific partition, then the status of the root partition will be displayed for us to know.

Option:

–h	used to format the sizes of the files in a human-readable manner rather than the number of bytes.

Syntax:

$ hadoop fs -df [-h] [...]

19. du:

This command is used to show the amount of space in bytes that have been used by the files that match the specified file pattern. Even without the –s option, this only shows the size summaries one level deep in the directory.

Options:

–s	used to show the size of each individual file that matches the pattern, shows the total (summary) size
–h	used to format the sizes of the files in a human-readable manner rather than the number of bytes.

Syntax:

$ hadoop fs -du [-s] [-h]

20. count:

This command is used to count the number of directories, files, and bytes under the path that matches the provided file pattern.

Syntax:

$ hadoop fs -count [-q]

Output:

The output columns are as follows:

1. DIR_COUNT FILE_COUNT CONTENT_SIZE FILE_NAME
2. QUOTA REMAINING_QUOTA SPACE_QUOTA REMAINING_SPACE_QUOTA
3. DIR_COUNT FILE_COUNT CONTENT_SIZE FILE_NAME

21. chgrp:

This command is used to change the group of a file or a path.

Syntax:

$ hadoop fs -chgrp [-R] groupname

22. chmod:

This command is used to change the permissions of a file, this command works similar to LINUX's shell command chmod with a few exceptions.

Option:

–R	Used to modify the files recursively and it is the only option that is being supported currently

The is the same as the mode used for the shell command. The letters that are recognized are 'rwxXt'.

This is the mode specified in 3 or 4 digits. The first maybe 0 or 1 to turn the sticky bit OFF or ON respectively. Unlike the shell command, it is not at all possible to specify only part of the mode.

Syntax:

$ hadoop fs -chmod [-R] PATH

23. chown:

This command is used to change the owner and group of a file. This command is similar to the shell's chown command with a few exceptions.

If only the owner of the group is specified then only the owner of the group is modified via this command. The owner and group names may only consist of digits, alphabets and any of the characters mentioned here [-_./@a-zA-Z0-9]. The names thus specified are case sensitive as well.

It is better to avoid using '.' to separate username and the group just the way LINUX allows it. If the usernames have dots in them and if you are using a local file system, you might see surprising results since the shell command chown is used for the local file alone.

Option:

–R	Modifies the files recursively and is the only option that is being supported currently

Syntax:

$ hadoop fs -chown [-R] [OWNER][:[GROUP]] PATH

Now that we have understood Hadoop distributed file commands (HDFS) we will learn frequently used Hadoop administration commands.

Hadoop admin commands

24. balancer:

This command is used to run the cluster-balancing utility.

Syntax:

hadoop balancer [-threshold <threshold>]

Example:

hadoop balancer -threshold 20

25. Datanode:

This command is used to run the HDFS DataNode service, which coordinates storage on each slave node. Before using the -rollback you need to stop the DataNode and distribute the earlier version of Hadoop.

Option:

-rollback	The DataNode is rolled back to the previous version.

Syntax:

hadoop datanode [-rollback]

Example:

hadoop datanode –rollback

26. dfsadmin:

This command is used to run a number of Hadoop Distributed File System (HDFS) administrative operations.

Options:

-help	This option is used to see a list of all supported options.
GENERIC_OPTIONS	It is a common set of options supported by several commands

Syntax:

hadoop dfsadmin [GENERIC_OPTIONS] [-report] [-safemode enter | leave | get | wait] [-refreshNodes] [-finalizeUpgrade] [-upgradeProgress status | details | force] [-metasave filename][-setQuota<quota><dirname>…<dirname>][-clrQuota <dirname>…<dirname>] [-restoreFailedStorage true|false|check] [-help [cmd]]

27. Secondary namenode:

This command is used to run the secondary NameNode.

Options:

-checkpoint	a checkpoint on the secondary NameNode is performed if the size of the EditLog is greater than or equal to fs.checkpoint.size
-force	a checkpoint is performed regardless of the EditLog size;
–geteditsize	EditLog size is displayed

Syntax:

hadoop secondarynamenode [-checkpoint [force]] | [-geteditsize]

Example:

hadoop secondarynamenode –geteditsize

28. tasktracker:

This command is used to run a MapReduce TaskTracker node.

Syntax:

hadoop tasktracker

Example:

hadoop tasktracker

29. jobtracker:

This command is used to run the MapReduce JobTracker node, which coordinates the data processing system for Hadoop.

Option:

-dumpConfiguration	Used by the JobTracker and the queue configuration in JSON format are written to standard output.

Syntax:

hadoop jobtracker [-dumpConfiguration]

Example:

hadoop jobtracker –dumpConfiguration

30. daemonlog:

This command is used to get or set the log level for each daemon. The changes reflect only when the daemon restarts.

Syntax:

hadoop daemonlog -getlevel <host:port> <name>; hadoop daemonlog -setlevel <host:port> <name> <level>

Example:

Hadoop daemonlog -getlevel 10.250.1.15:50030 org.apache.hadoop.mapred.JobTracker; hadoop daemonlog -setlevel 10.250.1.15:50030 org.apache.hadoop.mapred.JobTracker DEBUG