- **cat:** Copies source paths to stdout.

  Usage: hdfs dfs -cat URI [URI …]
  Example: hdfs dfs -cat hdfs://<path>/file1 hdfs dfs -cat file:///file2 /user/hadoop/file3

- **chgrp:** Changes the group association of files. With -R, makes the change recursively by way of the directory structure. The user must be the file owner or the superuser.
  Usage: hdfs dfs -chgrp [-R] GROUP URI [URI …]

- **chmod:** Changes the permissions of files. With -R, makes the change recursively by way of the directory structure. The user must be the file owner or the superuser

  Usage: hdfs dfs -chmod [-R] <MODE[,MODE]… | OCTALMODE> URI [URI …]
  Example: hdfs dfs -chmod 777 test/data1.txt

- **chown:** Changes the owner of files. With -R, makes the change recursively by way of the directory structure. The user must be the superuser.

  Usage: hdfs dfs -chown [-R] [OWNER][:[GROUP]] URI [URI]
  Example: hdfs dfs -chown -R hduser2 /opt/hadoop/logs

- **copyFromLocal:** Works similarly to the put command, except that the source is restricted to a local file reference.
  Usage: hdfs dfs -copyFromLocal <localsrc> URI
  Example: hdfs dfs -copyFromLocal input/docs/data2.txt hdfs://localhost/user/rosemary/data2.txt

- **copyToLocal:** Works similarly to the get command, except that the destination is restricted to a local file reference.

  Usage: hdfs dfs -copyToLocal [-ignorecrc] [-crc] URI <localdst>
  Example: hdfs dfs -copyToLocal data2.txt data2.copy.txt

- **count:** Counts the number of directories, files, and bytes under the paths that match the specified file pattern.
  Usage: hdfs dfs -count [-q] <paths>
  Example: hdfs dfs -count hdfs://nn1.example.com/file1 hdfs://nn2.example.com/file2

- **cp:** Copies one or more files from a specified source to a specified destination. If you specify multiple sources, the specified destination must be a directory.

  Usage: hdfs dfs -cp URI [URI …] <dest>
  Example: hdfs dfs -cp /user/hadoop/file1 /user/hadoop/file2 /user/hadoop/dir

- **du:** Displays the size of the specified file, or the sizes of files and directories that are contained in the specified directory. If you specify the -s option, displays an aggregate summary of file sizes rather than individual file sizes. If you specify the -h option, formats the file sizes in a "human-readable" way.
  Usage: hdfs dfs -du [-s] [-h] URI [URI …]
  Example: hdfs dfs -du /user/hadoop/dir1 /user/hadoop/file1

- **dus:** Displays a summary of file sizes; equivalent to hdfs dfs -du –s.
  Usage: hdfs dfs -dus <args>

- **expunge:** Empties the trash. When you delete a file, it isn't removed immediately from HDFS, but is renamed to a file in the /trash directory. As long as the file remains there, you can undelete it if you change your mind, though only the latest copy of the deleted file can be restored.

  Usage: hdfs dfs –expunge

- **get:** Copies files to the local file system. Files that fail a cyclic redundancy check (CRC) can still be copied if you specify the –ignorecrcoption. The CRC is a common technique for detecting data transmission errors. CRC checksum files have the .crc extension and are used to verify the data integrity of another file. These files are copied if you specify the -crc option.

  Usage: hdfs dfs -get [-ignorecrc] [-crc] <src> <localdst>
  Example: hdfs dfs -get /user/hadoop/file3 localfile

- **getmerge:** Concatenates the files in src and writes the result to the specified local destination file. To add a newline character at the end of each file, specify the addnl option.
  Usage: hdfs dfs -getmerge <src> <localdst> [addnl]
  Example: hdfs dfs -getmerge /user/hadoop/mydir/ ~/result_file addnl

- **ls:** Returns statistics for the specified files or directories.
  Usage: hdfs dfs -ls <args>
  Example: hdfs dfs -ls /user/hadoop/file1

- **lsr:** Serves as the recursive version of ls; similar to the Unix command ls -R.
  Usage: hdfs dfs -lsr <args>
  Example: hdfs dfs -lsr /user/hadoop

- **mkdir:** Creates directories on one or more specified paths. Its behavior is similar to the Unix mkdir -p command, which creates all directories that lead up to the specified directory if they don't exist already.
  Usage: hdfs dfs -mkdir <paths>
  Example: hdfs dfs -mkdir /user/hadoop/dir5/temp

- **moveFromLocal:** Works similarly to the put command, except that the source is deleted after it is copied.
  Usage: hdfs dfs -moveFromLocal <localsrc> <dest>
  Example: hdfs dfs -moveFromLocal localfile1 localfile2 /user/hadoop/hadoopdir

- **mv:** Moves one or more files from a specified source to a specified destination. If you specify multiple sources, the specified destination must be a directory. Moving files across file systems isn't permitted.
  Usage: hdfs dfs -mv URI [URI …] <dest>
  Example: hdfs dfs -mv /user/hadoop/file1 /user/hadoop/file2

- **put:** Copies files from the local file system to the destination file system. This command can also read input from stdin and write to the destination file system.
  Usage: hdfs dfs -put <localsrc> … <dest>
  Example: hdfs dfs -put localfile1 localfile2 /user/hadoop/hadoopdir; hdfs dfs -put –
  /user/hadoop/hadoopdir (reads input from stdin)

- **rm:** Deletes one or more specified files. This command doesn't delete empty directories or files. To bypass the trash (if it's enabled) and delete the specified files immediately, specify the -skipTrash option.
  Usage: hdfs dfs -rm [-skipTrash] URI [URI …]
  Example: hdfs dfs -rm hdfs://nn.example.com/file9

- **rmr:** Serves as the recursive version of –rm.
  Usage: hdfs dfs -rmr [-skipTrash] URI [URI …]
  Example: hdfs dfs -rmr /user/hadoop/dir

- **setrep:** Changes the replication factor for a specified file or directory. With –R, makes the change recursively by way of the directory structure.
  Usage: hdfs dfs -setrep <rep> [-R] <path>
  Example: hdfs dfs -setrep 3 -R /user/hadoop/dir1

- **stat:** Displays information about the specified path.
  Usage: hdfs dfs -stat URI [URI …]
  Example: hdfs dfs -stat /user/hadoop/dir1

- **tail:** Displays the last kilobyte of a specified file to stdout. The syntax supports the Unix -f option, which enables the specified file to be monitored. As new lines are added to the file by another process, tail updates the display.
  Usage: hdfs dfs -tail [-f] URI
  Example: hdfs dfs -tail /user/hadoop/dir1

- **test:** Returns attributes of the specified file or directory. Specifies –e to determine whether the file or directory exists; -z to determine whether the file or directory is empty; and -d to determine whether the URI is a directory.
  Usage: hdfs dfs -test -[ezd] URI
  Example: hdfs dfs -test /user/hadoop/dir1

- **text:** Outputs a specified source file in text format. Valid input file formats are zip and TextRecordInputStream.

Usage: hdfs dfs -text <src>
Example: hdfs dfs -text /user/hadoop/file8.zip

- **touchz:** Creates a new, empty file of size 0 in the specified path.
  Usage: hdfs dfs -touchz <path>
  Example: hdfs dfs -touchz /user/hadoop/file12

HADOOP ADMINISTRATION COMMANDS

Any Hadoop administrator worth his salt must master a comprehensive set of commands for cluster administration. The following list summarizes the most important commands, indicating what the command does as well as syntax and examples. Know them, and you will advance a long way along the path to Hadoop wisdom.

- **balancer:** Runs the cluster-balancing utility. The specified threshold value, which represents a percentage of disk capacity, is used to overwrite the default threshold value (10 percent). To stop the rebalancing process, press Ctrl+C.

  Syntax: hadoop balancer [-threshold <threshold>]
  Example: hadoop balancer -threshold 20

- **daemonlog:** Gets or sets the log level for each daemon (also known as a service). Connects to http://host:port/logLevel?log=name and prints or sets the log level of the daemon that's running at host:port. Hadoop daemons generate log files that help you determine what's happening on the system, and you can use the daemonlog command to temporarily change the log level of a Hadoop component when you're debugging the system. The change becomes effective 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

- **datanode:** Runs the HDFS DataNode service, which coordinates storage on each slave node. If you specify -rollback, the DataNode is rolled back to the previous version. Stop the DataNode and distribute the previous Hadoop version before using this option.
  Syntax: hadoop datanode [-rollback]
  Example: hadoop datanode –rollback

- **dfsadmin:** Runs a number of Hadoop Distributed File System (HDFS) administrative operations. Use the -help option to see a list of all supported options. The generic options are 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]]

- **mradmin:** Runs a number of MapReduce administrative operations. Use the -help option to see a list of all supported options. Again, the generic options are a common set of options that are supported by several commands. If you specify -refreshServiceAcl, reloads the service-level authorization policy file (JobTracker reloads the authorization policy file); -refreshQueues reloads the queue access control lists (ACLs) and state (JobTracker reloads the mapred-queues.xml file); -refreshNodes refreshes the hosts information at the JobTracker; -refreshUserToGroupsMappings refreshes user-to-groups mappings; -refreshSuperUserGroupsConfiguration refreshes superuser proxy groups mappings; and -help [cmd] displays help for the given command or for all commands if none is specified.

  Syntax: hadoop mradmin [GENERIC_OPTIONS] [-refreshServiceAcl] [-refreshQueues] [-refreshNodes] [-refreshUserToGroupsMappings] [-refreshSuperUserGroupsConfiguration] [-help [cmd]]
  Example: hadoop mradmin -help –refreshNodes

- **jobtracker:** Runs the MapReduce JobTracker node, which coordinates the data processing system for Hadoop. If you specify -dumpConfiguration, the configuration that's used by the JobTracker and the queue configuration in JSON format are written to standard output.
  Syntax: hadoop jobtracker [-dumpConfiguration]
  Example: hadoop jobtracker –dumpConfiguration

- **namenode:** Runs the NameNode, which coordinates the storage for the whole Hadoop cluster. If you specify -format, the NameNode is started, formatted, and then stopped; with -upgrade, the NameNode starts with the upgrade option after a new Hadoop version is distributed; with -rollback, the NameNode is rolled back to the previous version (remember to stop the cluster and distribute the previous Hadoop version before using this option); with -finalize, the previous state of the file system is removed, the most recent upgrade becomes permanent, rollback is no longer available, and the NameNode is stopped; finally, with -importCheckpoint, an image is loaded from the checkpoint directory (as specified by the fs.checkpoint.dir property) and saved into the current directory.
  Syntax: hadoop namenode [-format] | [-upgrade] | [-rollback] | [-finalize] | [-importCheckpoint]
  Example: hadoop namenode –finalize

- **Secondary namenode:** Runs the secondary NameNode. If you specify -checkpoint, a checkpoint on the secondary NameNode is performed if the size of the EditLog (a transaction log that records every change that occurs to the file system metadata) is greater than or equal to fs.checkpoint.size; specify -force and a checkpoint is performed regardless of the EditLog size; specify –geteditsize and the EditLog size is printed.
  Syntax: hadoop secondarynamenode [-checkpoint [force]] | [-geteditsize]
  Example: hadoop secondarynamenode –geteditsize

- **tasktracker:** Runs a MapReduce TaskTracker node.
  Syntax: hadoop tasktracker
  Example: hadoop tasktracker

THE HADOOP DFSADMIN COMMAND OPTIONS

The dfsadmin tools are a specific set of tools designed to help you root out information about your Hadoop Distributed File system (HDFS). As an added bonus, you can use them to perform some administration operations on HDFS as well.

Option	**What It Does**
-report	Reports basic file system information and statistics.
-safemode enter | leave | get | wait	Manages *safe* mode, a NameNode state in which changes to the name space are not accepted and blocks can be neither replicated nor deleted. The NameNode is in safe mode during start-up so that it doesn't prematurely start replicating blocks even though there are already enough replicas in the cluster.
-refreshNodes	Forces the NameNode to reread its configuration, including the dfs.hosts.exclude file. The NameNode decommissions nodes after their blocks have been replicated onto machines that will remain active.
-finalizeUpgrade	Completes the HDFS upgrade process. DataNodes and the NameNode delete working directories from the previous version.
-upgradeProgress status | details | force	Requests the standard or detailed current status of the distributed upgrade, or forces the upgrade to proceed.
-metasave filename	Saves the NameNode's primary data structures to *filename* in a directory that's specified by the hadoop.log.dir property. File *filename*, which is overwritten if it already exists, contains one line for each of these items: a) DataNodes that are exchanging heartbeats with the NameNode; b) blocks that are waiting to be replicated; c) blocks that are being replicated; and d) blocks that are waiting to be deleted.
-setQuota <quota> <dirname>…<dirname>	Sets an upper limit on the number of names in the directory tree. You can set this limit (a long integer) for one or more directories simultaneously.
-clrQuota <dirname>…<dirname>	Clears the upper limit on the number of names in the directory tree. You can clear this limit for one or more directories simultaneously.
-restoreFailedStorage true | false | check	Turns on or off the automatic attempts to restore failed storage replicas. If a failed storage location becomes available again, the system attempts to restore edits and the fsimage during a checkpoint. The check option returns the current setting.
-help [cmd]	Displays help information for the given command or for all commands if none is specified.