Hadoop Commands Guide

Table of contents

1. Overview

All Hadoop commands are invoked by the bin/hadoop script. Running the Hadoop script without any arguments prints the description for all commands.

```
Usage: hadoop [--config confdir] [COMMAND] [GENERIC_OPTIONS] [COMMAND_OPTIONS]
```

Hadoop has an option parsing framework that employs parsing generic options as well as running classes.

COMMAND_OPTION	Description
`--config confdir`	Overwrites the default Configuration directory. Default is ${HADOOP_HOME}/conf.
`GENERIC_OPTIONS`	The common set of options supported by multiple commands.
`COMMAND` `COMMAND_OPTIONS`	Various commands with their options are described in the following sections. The commands have been grouped into User Commands and Administration Commands.

1.1. Generic Options

The following options are supported by dfsadmin, fs, fsck and job. Applications should implement Tool to support GenericOptions.

GENERIC_OPTION	Description
`-conf <configuration file>`	Specify an application configuration file.
`-D <property=value>`	Use value for given property.
`-fs <local|namenode:port>`	Specify a namenode.
`-jt <local|jobtracker:port>`	Specify a job tracker. Applies only to job.
`-files <comma separated list of files>`	Specify comma separated files to be copied to the map reduce cluster. Applies only to job.
`-libjars <comma seperated list of jars>`	Specify comma separated jar files to include in the classpath. Applies only to job.
`-archives <comma separated list of archives>`	Specify comma separated archives to be unarchived on the compute machines. Applies only to job.

2. User Commands

Commands useful for users of a Hadoop cluster.

2.1. archive

Creates a Hadoop archive. More information see the Hadoop Archives Guide.

```
Usage: hadoop archive -archiveName NAME <src>* <dest>
```

COMMAND_OPTION	Description
`-archiveName NAME`	Name of the archive to be created.
`src`	Filesystem pathnames which work as usual with regular expressions.
`dest`	Destination directory which would contain the archive.

2.2. distcp

Copy file or directories recursively. More information can be found at DistCp Guide.

```
Usage: hadoop distcp <srcurl> <desturl>
```

COMMAND_OPTION	Description
`srcurl`	Source Url
`desturl`	Destination Url

2.3. fs

Runs a generic filesystem user client.

```
Usage: hadoop fs [GENERIC_OPTIONS] [COMMAND_OPTIONS]
```

The various COMMAND_OPTIONS can be found at File System Shell Guide.

2.4. fsck

Runs a HDFS filesystem checking utility. See Fsck for more info.

```
Usage: hadoop fsck [GENERIC_OPTIONS] <path> [-move | -delete |
-openforwrite] [-files [-blocks [-locations | -racks]]]
```

COMMAND_OPTION	Description
`<path>`	Start checking from this path.
`-move`	Move corrupted files to /lost+found
`-delete`	Delete corrupted files.
`-openforwrite`	Print out files opened for write.
`-files`	Print out files being checked.
`-blocks`	Print out block report.
`-locations`	Print out locations for every block.
`-racks`	Print out network topology for data-node locations.

2.5. jar

Runs a jar file. Users can bundle their Map Reduce code in a jar file and execute it using this command.

```
Usage: hadoop jar <jar> [mainClass] args...
```

The streaming jobs are run via this command. For examples, see Hadoop Streaming.

The WordCount example is also run using jar command. For examples, see the MapReduce Tutorial.

2.6. job

Command to interact with Map Reduce Jobs.

```
Usage: hadoop job [GENERIC_OPTIONS] [-submit <job-file>] |
[-status <job-id>] | [-counter <job-id> <group-name>
<counter-name>] | [-kill <job-id>] | [-events <job-id>
<from-event-#> <#-of-events>] | [-history [all] <historyFile>]
| [-list [all]] | [-kill-task <task-id>] | [-fail-task
<task-id>] | [-set-priority <job-id> <priority>]
```

COMMAND_OPTION	Description
`-submit <job-file>`	Submits the job.
`-status <job-id>`	Prints the map and reduce completion percentage and all job counters.

`-counter <job-id> <group-name> <counter-name>`	Prints the counter value.
`-kill <job-id>`	Kills the job.
`-events <job-id> <from-event-#> <#-of-events>`	Prints the events' details received by jobtracker for the given range.
`-history [all] <historyFile>`	-history <historyFile> prints job details, failed and killed tip details. More details about the job such as successful tasks and task attempts made for each task can be viewed by specifying the [all] option.
`-list [all]`	-list all displays all jobs. -list displays only jobs which are yet to complete.
`-kill-task <task-id>`	Kills the task. Killed tasks are NOT counted against failed attempts.
`-fail-task <task-id>`	Fails the task. Failed tasks are counted against failed attempts.
`-set-priority <job-id> <priority>`	Changes the priority of the job. Allowed priority values are VERY_HIGH, HIGH, NORMAL, LOW, VERY_LOW

2.7. pipes

Runs a pipes job.

```
Usage: hadoop pipes [-conf <path>] [-jobconf <key=value>, <key=value>, ...] [-input <path>] [-output <path>] [-jar <jar file>] [-inputformat <class>] [-map <class>] [-partitioner <class>] [-reduce <class>] [-writer <class>] [-program <executable>] [-reduces <num>]
```

COMMAND_OPTION	Description
`-conf <path>`	Configuration for job
`-jobconf <key=value>, <key=value>, ...`	Add/override configuration for job
`-input <path>`	Input directory
`-output <path>`	Output directory
`-jar <jar file>`	Jar filename

`-inputformat <class>`	InputFormat class
`-map <class>`	Java Map class
`-partitioner <class>`	Java Partitioner
`-reduce <class>`	Java Reduce class
`-writer <class>`	Java RecordWriter
`-program <executable>`	Executable URI
`-reduces <num>`	Number of reduces

2.8. queue

command to interact and view Job Queue information

`Usage : hadoop queue [-list] | [-info <job-queue-name> [-showJobs]] | [-showacls]`

COMMAND_OPTION	Description
`-list`	Gets list of Job Queues configured in the system. Along with scheduling information associated with the job queues.
`-info <job-queue-name> [-showJobs]`	Displays the job queue information and associated scheduling information of particular job queue. If -showJobs options is present a list of jobs submitted to the particular job queue is displayed.
`-showacls`	Displays the queue name and associated queue operations allowed for the current user. The list consists of only those queues to which the user has access.

2.9. version

Prints the version.

`Usage: hadoop version`

2.10. CLASSNAME

Hadoop script can be used to invoke any class.

Runs the class named CLASSNAME.

```
Usage: hadoop CLASSNAME
```

3. Administration Commands

Commands useful for administrators of a Hadoop cluster.

3.1. balancer

Runs a cluster balancing utility. An administrator can simply press Ctrl-C to stop the rebalancing process. For more details see Rebalancer.

```
Usage: hadoop balancer [-threshold <threshold>]
```

COMMAND_OPTION	Description
`-threshold <threshold>`	Percentage of disk capacity. This overwrites the default threshold.

3.2. daemonlog

Get/Set the log level for each daemon.

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

COMMAND_OPTION	Description
`-getlevel <host:port> <name>`	Prints the log level of the daemon running at <host:port>. This command internally connects to http://<host:port>/logLevel?log=<name>
`-setlevel <host:port> <name> <level>`	Sets the log level of the daemon running at <host:port>. This command internally connects to http://<host:port>/logLevel?log=<name>

3.3. datanode

Runs a HDFS datanode.

```
Usage: hadoop datanode [-rollback]
```

COMMAND_OPTION	Description
`-rollback`	Rollsback the datanode to the previous version. This should be used after stopping the datanode

	and distributing the old Hadoop version.

3.4. dfsadmin

Runs a HDFS dfsadmin client.

```
Usage: 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]]
```

COMMAND_OPTION	Description
`-report`	Reports basic filesystem information and statistics.
`-safemode enter | leave | get | wait`	Safe mode maintenance command. Safe mode is a Namenode state in which it 1. does not accept changes to the name space (read-only) 2. does not replicate or delete blocks. Safe mode is entered automatically at Namenode startup, and leaves safe mode automatically when the configured minimum percentage of blocks satisfies the minimum replication condition. Safe mode can also be entered manually, but then it can only be turned off manually as well.
`-refreshNodes`	Re-read the hosts and exclude files to update the set of Datanodes that are allowed to connect to the Namenode and those that should be decommissioned or recommissioned.
`-finalizeUpgrade`	Finalize upgrade of HDFS. Datanodes delete their previous version working directories, followed by Namenode doing the same. This completes the upgrade process.
`-printTopology`	Print a tree of the rack/datanode topology of the cluster as seen by the NameNode.
`-upgradeProgress status | details | force`	Request current distributed upgrade status, a detailed status or force the upgrade to proceed.
`-metasave filename`	Save Namenode's primary data structures to

	<filename> in the directory specified by hadoop.log.dir property. <filename> will contain one line for each of the following 1. Datanodes heart beating with Namenode 2. Blocks waiting to be replicated 3. Blocks currrently being replicated 4. Blocks waiting to be deleted
`-setQuota <quota> <dirname>...<dirname>`	Set the quota <quota> for each directory <dirname>. The directory quota is a long integer that puts a hard limit on the number of names in the directory tree. Best effort for the directory, with faults reported if 1. N is not a positive integer, or 2. user is not an administrator, or 3. the directory does not exist or is a file, or 4. the directory would immediately exceed the new quota.
`-clrQuota <dirname>...<dirname>`	Clear the quota for each directory <dirname>. Best effort for the directory. with fault reported if 1. the directory does not exist or is a file, or 2. user is not an administrator. It does not fault if the directory has no quota.
`-restoreFailedStorage true | false | check`	This option will turn on/off automatic attempt to restore failed storage replicas. If a failed storage becomes available again the system will attempt to restore edits and/or fsimage during checkpoint. 'check' option will return current setting.
`-help [cmd]`	Displays help for the given command or all commands if none is specified.

3.5. mradmin

Runs MR admin client

`Usage: hadoop mradmin [` GENERIC_OPTIONS `] [-refreshServiceAcl] [-refreshQueues] [-refreshNodes] [-help [cmd]]`

COMMAND_OPTION	Description
`-refreshServiceAcl`	Reload the service-level authorization policies. Jobtracker will reload the authorization policy file.

`-refreshQueues`	Reload the queues' configuration at the JobTracker. Most of the configuration of the queues can be refreshed/reloaded without restarting the Map/Reduce sub-system. Administrators typically own the *conf/mapred-queues.xml* file, can edit it while the JobTracker is still running, and can do a reload by running this command. It should be noted that while trying to refresh queues' configuration, one cannot change the hierarchy of queues itself. This means no operation that involves a change in either the hierarchy structure itself or the queues' names will be allowed. Only selected properties of queues can be changed during refresh. For example, new queues cannot be added dynamically, neither can an existing queue be deleted. If during a reload of queue configuration, a syntactic or semantic error in made during the editing of the configuration file, the refresh command fails with an exception that is printed on the standard output of this command, thus informing the requester with any helpful messages of what has gone wrong during the edit/reload. Importantly, the existing queue configuration is untouched and the system is left in a consistent state. As described in the *conf/mapred-queues.xml* section, the *<properties>* tag in the queue configuration file can also be used to specify per-queue properties needed by the scheduler. When the framework's queue configuration is reloaded using this command, this scheduler specific configuration will also be reloaded , provided the scheduler being configured supports this reload. Please see the documentation of the particular scheduler in use.
`-refreshNodes`	Refresh the hosts information at the jobtracker.
`-help [cmd]`	Displays help for the given command or all commands if none is specified.

3.6. jobtracker

Runs the MapReduce job Tracker node.

```
Usage: hadoop jobtracker [-dumpConfiguration]
```

COMMAND_OPTION	Description
`-dumpConfiguration`	Dumps the configuration used by the JobTracker alongwith queue configuration in JSON format into Standard output used by the jobtracker and exits.

3.7. namenode

Runs the namenode. For more information about upgrade, rollback and finalize see Upgrade and Rollback.

```
Usage: hadoop namenode [-format] | [-upgrade] | [-rollback] | [-finalize] | [-importCheckpoint] | [-checkpoint] | [-backup]
```

COMMAND_OPTION	Description
`-regular`	Start namenode in standard, active role rather than as backup or checkpoint node. This is the default role.
`-checkpoint`	Start namenode in checkpoint role, creating periodic checkpoints of the active namenode metadata.
`-backup`	Start namenode in backup role, maintaining an up-to-date in-memory copy of the namespace and creating periodic checkpoints.
`-format`	Formats the namenode. It starts the namenode, formats it and then shut it down.
`-upgrade`	Namenode should be started with upgrade option after the distribution of new Hadoop version.
`-rollback`	Rollsback the namenode to the previous version. This should be used after stopping the cluster and distributing the old Hadoop version.
`-finalize`	Finalize will remove the previous state of the files system. Recent upgrade will become permanent. Rollback option will not be available anymore. After finalization it shuts the namenode down.
`-importCheckpoint`	Loads image from a checkpoint directory and saves it into the current one. Checkpoint directory is read from property fs.checkpoint.dir

	(see Import Checkpoint).
`-checkpoint`	Enables checkpointing (see Checkpoint Node).
`-backup`	Enables checkpointing and maintains an in-memory, up-to-date copy of the file system namespace (see Backup Node).

3.8. secondarynamenode

Note:
The Secondary NameNode has been deprecated. Instead, consider using the Checkpoint Node or Backup Node.

Runs the HDFS secondary namenode. See Secondary NameNode for more info.

```
Usage: hadoop secondarynamenode [-checkpoint [force]] |
[-geteditsize]
```

COMMAND_OPTION	Description
`-checkpoint [force]`	Checkpoints the Secondary namenode if EditLog size >= fs.checkpoint.size. If -force is used, checkpoint irrespective of EditLog size.
`-geteditsize`	Prints the EditLog size.

3.9. tasktracker

Runs a MapReduce task Tracker node.

```
Usage: hadoop tasktracker
```