HDFS Commands Guide

Overview

All HDFS commands are invoked by the `bin/hdfs` script. Running the hdfs script without any arguments prints the description for all commands.

Usage: `hdfs [SHELL_OPTIONS] COMMAND [GENERIC_OPTIONS] [COMMAND_OPTIONS]`

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

COMMAND_OPTIONS	Description
`--config --loglevel`	The common set of shell options. These are documented on the Commands Manual page.
GENERIC_OPTIONS	The common set of options supported by multiple commands. See the Hadoop Commands Manual for more information.
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.

User Commands

Commands useful for users of a hadoop cluster.

`classpath`

Usage: `hdfs classpath`

Prints the class path needed to get the Hadoop jar and the required libraries

`dfs`

Usage: `hdfs dfs [COMMAND [COMMAND_OPTIONS]]`

Run a filesystem command on the file system supported in Hadoop. The various COMMAND_OPTIONS can be found at File System Shell Guide.

fetchdt

Usage: `hdfs fetchdt [--webservice <namenode_http_addr>] <path>`

COMMAND_OPTION	Description
`--webservice` *https_address*	use http protocol instead of RPC
fileName	File name to store the token into.

Gets Delegation Token from a NameNode. See fetchdt for more info.

fsck

Usage:

```
hdfs fsck <path>
       [-list-corruptfileblocks |
       [-move | -delete | -openforwrite]
       [-files [-blocks [-locations | -racks]]]
       [-includeSnapshots]
       [-storagepolicies] [-blockId <blk_Id>]
```

COMMAND_OPTION	Description
path	Start checking from this path.
`-delete`	Delete corrupted files.
`-files`	Print out files being checked.
`-files -blocks`	Print out the block report
`-files -blocks -locations`	Print out locations for every block.
`-files -blocks -racks`	Print out network topology for data-node locations.
`-includeSnapshots`	Include snapshot data if the given path indicates a snapshottable directory or there are snapshottable directories under it.
`-list-corruptfileblocks`	Print out list of missing blocks and files they belong to.
`-move`	Move corrupted files to /lost+found.
`-openforwrite`	Print out files opened for write.
`-storagepolicies`	Print out storage policy summary for the blocks.
`-blockId`	Print out information about the block.

Runs the HDFS filesystem checking utility. See fsck for more info.

getconf

Usage:

```
hdfs getconf -namenodes
hdfs getconf -secondaryNameNodes
hdfs getconf -backupNodes
hdfs getconf -includeFile
hdfs getconf -excludeFile
```

```
hdfs getconf -nnRpcAddresses
hdfs getconf -confKey [key]
```

COMMAND_OPTION	Description
`-namenodes`	gets list of namenodes in the cluster.
`-secondaryNameNodes`	gets list of secondary namenodes in the cluster.
`-backupNodes`	gets list of backup nodes in the cluster.
`-includeFile`	gets the include file path that defines the datanodes that can join the cluster.
`-excludeFile`	gets the exclude file path that defines the datanodes that need to decommissioned.
`-nnRpcAddresses`	gets the namenode rpc addresses
`-confKey` [key]	gets a specific key from the configuration

Gets configuration information from the configuration directory, post-processing.

groups

Usage: `hdfs groups [username ...]`

Returns the group information given one or more usernames.

lsSnapshottableDir

Usage: `hdfs lsSnapshottableDir [-help]`

COMMAND_OPTION	Description
`-help`	print help

Get the list of snapshottable directories. When this is run as a super user, it returns all snapshottable directories. Otherwise it returns those directories that are owned by the current user.

jmxget

Usage: `hdfs jmxget [-localVM ConnectorURL | -port port | -server mbeanserver | -service service]`

COMMAND_OPTION	Description
`-help`	print help
`-localVM` ConnectorURL	connect to the VM on the same machine
`-port` *mbean server port*	specify mbean server port, if missing it will try to connect to MBean Server in the same VM
`-service`	specify jmx service, either DataNode or NameNode, the default

Dump JMX information from a service.

oev

Usage: `hdfs oev [OPTIONS] -i INPUT_FILE -o OUTPUT_FILE`

Required command line arguments:

COMMAND_OPTION	Description
`-i`,`--inputFile` *arg*	edits file to process, xml (case insensitive) extension means XML format, any other filename means binary format
`-o`,`--outputFile` *arg*	Name of output file. If the specified file exists, it will be overwritten, format of the file is determined by -p option

Optional command line arguments:

COMMAND_OPTION	Description
`-f`,`--fix-txids`	Renumber the transaction IDs in the input, so that there are no gaps or invalid transaction IDs.
`-h`,`--help`	Display usage information and exit
`-r`,`--ecover`	When reading binary edit logs, use recovery mode. This will give you the chance to skip corrupt parts of the edit log.
`-p`,`--processor` *arg*	Select which type of processor to apply against image file, currently supported processors are: binary (native binary format that Hadoop uses), xml (default, XML format), stats (prints statistics about edits file)
`-v`,`--verbose`	More verbose output, prints the input and output filenames, for processors that write to a file, also output to screen. On large image files this will dramatically increase processing time (default is false).

Hadoop offline edits viewer.

`oiv`

Usage: `hdfs oiv [OPTIONS] -i INPUT_FILE`

Required command line arguments:

COMMAND_OPTION	Description
`-i`,`--inputFile` *arg*	edits file to process, xml (case insensitive) extension means XML format, any other filename means binary format

Optional command line arguments:

COMMAND_OPTION	Description
`-h`,`--help`	Display usage information and exit
`-o`,`--outputFile` *arg*	Name of output file. If the specified file exists, it will be overwritten, format of the file is determined by -p option
`-p`,`--processor` *arg*	Select which type of processor to apply against image file, currently supported processors are: binary (native binary format that Hadoop uses), xml (default, XML format), stats (prints statistics about edits file)

Hadoop Offline Image Viewer for newer image files.

`oiv_legacy`

Usage: `hdfs oiv_legacy [OPTIONS] -i INPUT_FILE -o OUTPUT_FILE`

COMMAND_OPTION	Description
`-h`,`--help`	Display usage information and exit
`-i`,`--inputFile` *arg*	edits file to process, xml (case insensitive) extension means XML format, any other filename means binary format
`-o`,`--outputFile` *arg*	Name of output file. If the specified file exists, it will be overwritten, format of the file is determined by -p option

Hadoop offline image viewer for older versions of Hadoop.

`snapshotDiff`

Usage: `hdfs snapshotDiff <path> <fromSnapshot> <toSnapshot>`

Determine the difference between HDFS snapshots. See the HDFS Snapshot Documentation for more information.

`version`

Usage: `hdfs version`

Prints the version.

Administration Commands

Commands useful for administrators of a hadoop cluster.

balancer

Usage:

```
hdfs balancer
      [-threshold <threshold>]
      [-policy <policy>]
      [-exclude [-f <hosts-file> | <comma-separated list of hosts>]]
      [-include [-f <hosts-file> | <comma-separated list of hosts>]]
      [-idleiterations <idleiterations>]
```

COMMAND_OPTION	Description
`-policy` <policy>	`datanode` (default): Cluster is balanced if each datanode is balanced. `blockpool`: Cluster is balanced if each block pool in each datanode is balanced.
`-threshold` <threshold>	Percentage of disk capacity. This overwrites the default threshold.
`-exclude -f` <hosts-file> | <comma-separated list of hosts>	Excludes the specified datanodes from being balanced by the balancer.
`-include -f` <hosts-file> | <comma-separated list of hosts>	Includes only the specified datanodes to be balanced by the balancer.
`-idleiterations` <iterations>	Maximum number of idle iterations before exit. This overwrites the default idleiterations(5).

Runs a cluster balancing utility. An administrator can simply press Ctrl-C to stop the rebalancing process. See Balancer for more details.

Note that the `blockpool` policy is more strict than the `datanode` policy.

cacheadmin

Usage: `hdfs cacheadmin -addDirective -path <path> -pool <pool-name> [-force] [-replication <replication>] [-ttl <time-to-live>]`

See the HDFS Cache Administration Documentation for more information.

crypto

Usage:

```
hdfs crypto -createZone -keyName <keyName> -path <path>
hdfs crypto -help <command-name>
hdfs crypto -listZones
```

See the HDFS Transparent Encryption Documentation for more information.

datanode

Usage: `hdfs datanode [-regular | -rollback | -rollingupgrace rollback]`

COMMAND_OPTION	Description
`-regular`	Normal datanode startup (default).
`-rollback`	Rollback the datanode to the previous version. This should be used after stopping the datanode and distributing the old hadoop version.
`-rollingupgrade` rollback	Rollback a rolling upgrade operation.

Runs a HDFS datanode.

dfsadmin

Usage:

```
hdfs dfsadmin [GENERIC_OPTIONS]
      [-report [-live] [-dead] [-decommissioning]]
      [-safemode enter | leave | get | wait]
      [-saveNamespace]
      [-rollEdits]
      [-restoreFailedStorage true |false |check]
      [-refreshNodes]
      [-setQuota <quota> <dirname>...<dirname>]
      [-clrQuota <dirname>...<dirname>]
      [-setSpaceQuota <quota> <dirname>...<dirname>]
      [-clrSpaceQuota <dirname>...<dirname>]
      [-setStoragePolicy <path> <policyName>]
      [-getStoragePolicy <path>]
      [-finalizeUpgrade]
      [-rollingUpgrade [<query> |<prepare> |<finalize>]]
      [-metasave filename]
      [-refreshServiceAcl]
      [-refreshUserToGroupsMappings]
      [-refreshSuperUserGroupsConfiguration]
      [-refreshCallQueue]
      [-refresh <host:ipc_port> <key> [arg1..argn]]
      [-reconfig <datanode |...> <host:ipc_port> <start |status>]
      [-printTopology]
      [-refreshNamenodes datanodehost:port]
      [-deleteBlockPool datanode-host:port blockpoolId [force]]
      [-setBalancerBandwidth <bandwidth in bytes per second>]
      [-allowSnapshot <snapshotDir>]
      [-disallowSnapshot <snapshotDir>]
      [-fetchImage <local directory>]
      [-shutdownDatanode <datanode_host:ipc_port> [upgrade]]
      [-getDatanodeInfo <datanode_host:ipc_port>]
      [-triggerBlockReport [-incremental] <datanode_host:ipc_port>]
      [-help [cmd]]
```

COMMAND_OPTION	Description
`-report [-live] [-dead] [-decommissioning]`	Reports basic filesystem information and statistics. Optional flags may be used to filter the list of displayed DataNodes.
`-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.
`-saveNamespace`	Save current namespace into storage directories and reset edits log. Requires safe mode.
`-rollEdits`	Rolls the edit log on the active NameNode.

COMMAND_OPTION	Description
`-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.
`-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.
`-setQuota` <quota> <dirname>…<dirname>	See HDFS Quotas Guide for the detail.
`-clrQuota` <dirname>…<dirname>	See HDFS Quotas Guide for the detail.
`-setSpaceQuota` <quota> <dirname>… <dirname>	See HDFS Quotas Guide for the detail.
`-clrSpaceQuota` <dirname>…<dirname>	See HDFS Quotas Guide for the detail.
`-setStoragePolicy` <path> <policyName>	Set a storage policy to a file or a directory.
`-getStoragePolicy` <path>	Get the storage policy of a file or a directory.
`-finalizeUpgrade`	Finalize upgrade of HDFS. Datanodes delete their previous version working directories, followed by Namenode doing the same. This completes the upgrade process.
`-rollingUpgrade` [<query>|<prepare>| <finalize>]	See Rolling Upgrade document for the detail.
`-metasave` filename	Save Namenode's primary data structures to *filename* in the directory specified by hadoop.log.dir property. *filename* is overwritten if it exists. *filename* will contain one line for each of the following 1. Datanodes heart beating with Namenode 2. Blocks waiting to be replicated 3. Blocks currently being replicated 4. Blocks waiting to be deleted
`-refreshServiceAcl`	Reload the service-level authorization policy file.
`-refreshUserToGroupsMappings`	Refresh user-to-groups mappings.
`-refreshSuperUserGroupsConfiguration`	Refresh superuser proxy groups mappings
`-refreshCallQueue`	Reload the call queue from config.
`-refresh` <host:ipc_port> <key> [arg1..argn]	Triggers a runtime-refresh of the resource specified by <key> on <host:ipc_port>. All other args after are sent to the host.
`-reconfig` <datanode |…> <host:ipc_port> <start|status>	Start reconfiguration or get the status of an ongoing reconfiguration. The second parameter specifies the node type. Currently, only reloading DataNode's configuration is supported.
`-printTopology`	Print a tree of the racks and their nodes as reported by the Namenode
`-refreshNamenodes` datanodehost:port	For the given datanode, reloads the configuration files, stops serving the removed block-pools and starts serving new block-pools.
`-deleteBlockPool` datanode-host:port blockpoolId [force]	If force is passed, block pool directory for the given blockpool id on the given datanode is deleted along with its contents, otherwise the directory is deleted only if it is empty. The command will fail if datanode is still serving the block pool. Refer to refreshNamenodes to shutdown a block pool service on a datanode.
`-setBalancerBandwidth` <bandwidth in bytes per second>	Changes the network bandwidth used by each datanode during HDFS block balancing. <bandwidth> is the maximum number of bytes per second that will be used by each datanode. This value overrides the dfs.balance.bandwidthPerSec parameter. NOTE: The new value is not persistent on the DataNode.
`-allowSnapshot` <snapshotDir>	Allowing snapshots of a directory to be created. If the operation completes successfully, the directory becomes snapshottable. See the HDFS Snapshot Documentation for more information.
`-disallowSnapshot` <snapshotDir>	Disallowing snapshots of a directory to be created. All snapshots of the directory must be deleted before disallowing snapshots. See the HDFS Snapshot Documentation for more information.
`-fetchImage` <local directory>	Downloads the most recent fsimage from the NameNode and saves it in the specified local directory.
`-shutdownDatanode` <datanode_host:ipc_port> [upgrade]	Submit a shutdown request for the given datanode. See Rolling Upgrade document for the detail.
`-getDatanodeInfo` <datanode_host:ipc_port>	Get the information about the given datanode. See Rolling Upgrade document for the detail.
`-triggerBlockReport [-incremental]` <datanode_host:ipc_port>	Trigger a block report for the given datanode. If 'incremental' is specified, it will be otherwise, it will be a full block report.
`-help` [cmd]	Displays help for the given command or all commands if none is specified.

Runs a HDFS dfsadmin client.

`haadmin`

Usage:

```
    hdfs haadmin -checkHealth <serviceId>
    hdfs haadmin -failover [--forcefence] [--forceactive] <serviceId> <serviceId>
    hdfs haadmin -getServiceState <serviceId>
    hdfs haadmin -help <command>
    hdfs haadmin -transitionToActive <serviceId> [--forceactive]
    hdfs haadmin -transitionToStandby <serviceId>
```

COMMAND_OPTION	Description
`-checkHealth`	check the health of the given NameNode
`-failover`	initiate a failover between two NameNodes
`-getServiceState`	determine whether the given NameNode is Active or Standby
`-transitionToActive`	transition the state of the given NameNode to Active (Warning: No fencing is done)
`-transitionToStandby`	transition the state of the given NameNode to Standby (Warning: No fencing is done)

See HDFS HA with NFS or HDFS HA with QJM for more information on this command.

journalnode

Usage: `hdfs journalnode`

This comamnd starts a journalnode for use with HDFS HA with QJM.

mover

Usage: `hdfs mover [-p <files/dirs> | -f <local file name>]`

COMMAND_OPTION	Description
-f <local file>	Specify a local file containing a list of HDFS files/dirs to migrate.
-p <files/dirs>	Specify a space separated list of HDFS files/dirs to migrate.

Runs the data migration utility. See Mover for more details.

Note that, when both -p and -f options are omitted, the default path is the root directory.

namenode

Usage:

```
  hdfs namenode [-backup] |
          [-checkpoint] |
          [-format [-clusterid cid ] [-force] [-nonInteractive] ] |
          [-upgrade [-clusterid cid] [-renameReserved<k-v pairs>] ] |
          [-upgradeOnly [-clusterid cid] [-renameReserved<k-v pairs>] ] |
          [-rollback] |
          [-rollingUpgrade <downgrade |rollback> ] |
          [-finalize] |
          [-importCheckpoint] |
          [-initializeSharedEdits] |
          [-bootstrapStandby] |
          [-recover [-force] ] |
          [-metadataVersion ]
```

COMMAND_OPTION	Description

`-backup`	Start backup node.
`-checkpoint`	Start checkpoint node.
`-format [-clusterid cid] [-force] [-nonInteractive]`	Formats the specified NameNode. It starts the NameNode, formats it and then shut it down. -force option formats if the name directory exists. -nonInteractive option aborts if the name directory exists, unless -force option is specified.
`-upgrade [-clusterid cid] [-renameReserved` <k-v pairs>`]`	Namenode should be started with upgrade option after the distribution of new Hadoop version.
`-upgradeOnly [-clusterid cid] [-renameReserved` <k-v pairs>`]`	Upgrade the specified NameNode and then shutdown it.
`-rollback`	Rollback the NameNode to the previous version. This should be used after stopping the cluster and distributing the old Hadoop version.
`-rollingUpgrade` <downgrade|rollback|started>	See Rolling Upgrade document for the detail.
`-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 save it into the current one. Checkpoint dir is read from property fs.checkpoint.dir
`-initializeSharedEdits`	Format a new shared edits dir and copy in enough edit log segments so that the standby NameNode can start up.
`-bootstrapStandby`	Allows the standby NameNode's storage directories to be bootstrapped by copying the latest namespace snapshot from the active NameNode. This is used when first configuring an HA cluster.
`-recover [-force]`	Recover lost metadata on a corrupt filesystem. See HDFS User Guide for the detail.
`-metadataVersion`	Verify that configured directories exist, then print the metadata versions of the software and the image.

Runs the namenode. More info about the upgrade, rollback and finalize is at Upgrade Rollback.

nfs3

Usage: `hdfs nfs3`

This comamnd starts the NFS3 gateway for use with the HDFS NFS3 Service.

portmap

Usage: `hdfs portmap`

This comamnd starts the RPC portmap for use with the HDFS NFS3 Service.

secondarynamenode

Usage: `hdfs secondarynamenode [-checkpoint [force]] | [-format] | [-geteditsize]`

COMMAND_OPTION	Description
`-checkpoint` [force]	Checkpoints the SecondaryNameNode if EditLog size >= fs.checkpoint.size. If `force` is used, checkpoint irrespective of EditLog size.
`-format`	Format the local storage during startup.
`-geteditsize`	Prints the number of uncheckpointed transactions on the NameNode.

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

storagepolicies

Usage: `hdfs storagepolicies`

Lists out all storage policies. See the HDFS Storage Policy Documentation for more information.

zkfc

Usage: `hdfs zkfc [-formatZK [-force] [-nonInteractive]]`

COMMAND_OPTION	Description
`-formatZK`	Format the Zookeeper instance
`-h`	Display help

This comamnd starts a Zookeeper Failover Controller process for use with HDFS HA with QJM.

Debug Commands

Useful commands to help administrators debug HDFS issues, like validating block files and calling recoverLease.

`verify`

Usage: `hdfs debug verify [-meta <metadata-file>] [-block <block-file>]`

COMMAND_OPTION	Description
`-block` *block-file*	Optional parameter to specify the absolute path for the block file on the local file system of the data node.
`-meta` *metadata-file*	Absolute path for the metadata file on the local file system of the data node.

Verify HDFS metadata and block files. If a block file is specified, we will verify that the checksums in the metadata file match the block file.

`recoverLease`

Usage: `hdfs debug recoverLease [-path <path>] [-retries <num-retries>]`

COMMAND_OPTION	Description
[`-path` *path*]	HDFS path for which to recover the lease.
[`-retries` *num-retries*]	Number of times the client will retry calling recoverLease. The default number of retries is 1.

Recover the lease on the specified path. The path must reside on an HDFS filesystem. The default number of retries is 1.