Hadoop Commands Guide

Overview

All of the Hadoop commands and subprojects follow the same basic structure:

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

FIELD	Description
shellcommand	The command of the project being invoked. For example, Hadoop common uses `hadoop`, HDFS uses `hdfs`, and YARN uses `yarn`.
SHELL_OPTIONS	Options that the shell processes prior to executing Java.
COMMAND	Action to perform.
GENERIC_OPTIONS	The common set of options supported by multiple commands.
COMMAND_OPTIONS	Various commands with their options are described in this documention for the Hadoop common sub-project. HDFS and YARN are covered in other documents.

Shell Options

All of the shell commands will accept a common set of options. For some commands, these options are ignored. For example, passing `---hostnames` on a command that only executes on a single host will be ignored.

SHELL_OPTION	Description
`--buildpaths`	Enables developer versions of jars.
`--config confdir`	Overwrites the default Configuration directory. Default is `$HADOOP_HOME/etc/hadoop`.

SHELL_OPTION	Description
`--daemon mode`	If the command supports daemonization (e.g., `hdfs namenode`), execute in the appropriate mode. Supported modes are `start` to start the process in daemon mode, `stop` to stop the process, and `status` to determine the active status of the process. `status` will return an LSB-compliant result code. If no option is provided, commands that support daemonization will run in the foreground. For commands that do not support daemonization, this option is ignored.
`--debug`	Enables shell level configuration debugging information
`--help`	Shell script usage information.
`--hostnames`	When `--workers` is used, override the workers file with a space delimited list of hostnames where to execute a multi-host subcommand. If `--workers` is not used, this option is ignored.
`--hosts`	When `--workers` is used, override the workers file with another file that contains a list of hostnames where to execute a multi-host subcommand. If `--workers` is not used, this option is ignored.
`--loglevel loglevel`	Overrides the log level. Valid log levels are FATAL, ERROR, WARN, INFO, DEBUG, and TRACE. Default is INFO.
`--workers`	If possible, execute this command on all hosts in the `workers` file.

Generic Options

Many subcommands honor a common set of configuration options to alter their behavior:

GENERIC_OPTION	Description
`-archives <comma separated list of archives>`	Specify comma separated archives to be unarchived on the compute machines. Applies only to job.
`-conf <configuration file>`	Specify an application configuration file.
`-D <property>=<value>`	Use value for given property.
`-files <comma separated list of files>`	Specify comma separated files to be copied to the map reduce cluster. Applies only to job.
`-fs <file:///> or <hdfs://namenode:port>`	Specify default filesystem URL to use. Overrides 'fs.defaultFS' property from configurations.
`-jt <local> or <resourcemanager:port>`	Specify a ResourceManager. Applies only to job.
`-libjars <comma seperated list of jars>`	Specify comma separated jar files to include in the classpath. Applies only to job.

Hadoop Common Commands

All of these commands are executed from the `hadoop` shell command. They have been broken up into User Commands and Administration Commands.

User Commands

Commands useful for users of a hadoop cluster.

archive

Creates a hadoop archive. More information can be found at Hadoop Archives Guide.

checknative

Usage: `hadoop checknative [-a] [-h]`

COMMAND_OPTION	Description
`-a`	Check all libraries are available.
`-h`	print help

This command checks the availability of the Hadoop native code. See Native Libaries for more information. By default, this command only checks the availability of libhadoop.

classpath

Usage: `hadoop classpath [--glob |--jar <path> |-h |--help]`

COMMAND_OPTION	Description
`--glob`	expand wildcards
`--jar` *path*	write classpath as manifest in jar named *path*
`-h`, `--help`	print help

Prints the class path needed to get the Hadoop jar and the required libraries. If called without arguments, then prints the classpath set up by the command scripts, which is likely to contain wildcards in the classpath entries. Additional options print the classpath after wildcard expansion or write the classpath into the manifest of a jar file. The latter is useful in environments where wildcards cannot be used and the expanded classpath exceeds the maximum supported command line length.

conftest

Usage: `hadoop conftest [-conffile <path>]...`

COMMAND_OPTION	Description
`-conffile`	Path of a configuration file or directory to validate
`-h`, `--help`	print help

Validates configuration XML files. If the `-conffile` option is not specified, the files in `${HADOOP_CONF_DIR}` whose name end with .xml will be verified. If specified, that path will be verified. You can specify either a file or directory, and if a directory specified, the files in that directory whose name end with `.xml` will be verified. You can specify `-conffile` option multiple times.

The validation is fairly minimal: the XML is parsed and duplicate and empty property names are checked for. The command does not support XInclude; if you using that to pull in configuration items, it will declare the XML file invalid.

credential

Usage: `hadoop credential <subcommand> [options]`

COMMAND_OPTION	Description
create *alias* [-provider *provider-path*] [-strict] [-value *credential-value*]	Prompts the user for a credential to be stored as the given alias. The *hadoop.security.credential.provider.path* within the core-site.xml file will be used unless a `-provider` is indicated. The `-strict` flag will cause the command to fail if the provider uses a default password. Use `-value` flag to supply the credential value (a.k.a. the alias password) instead of being prompted.
delete *alias* [-provider *provider-path*] [-strict] [-f]	Deletes the credential with the provided alias. The *hadoop.security.credential.provider.path* within the core-site.xml file will be used unless a `-provider` is indicated. The `-strict` flag will cause the command to fail if the provider uses a default password. The command asks for confirmation unless `-f` is specified
list [-provider *provider-path*] [-strict]	Lists all of the credential aliases The *hadoop.security.credential.provider.path* within the core-site.xml file will be used unless a `-provider` is indicated. The `-strict` flag will cause the command to fail if the provider uses a default password.

Command to manage credentials, passwords and secrets within credential providers.

The CredentialProvider API in Hadoop allows for the separation of applications and how they store their required passwords/secrets. In order to indicate a particular provider type and location, the user must provide the *hadoop.security.credential.provider.path* configuration element in core-site.xml or use the command line option `-provider` on each of the following commands. This provider path is a comma-separated list of URLs that indicates the type and location of a list of providers that should be consulted. For example, the following path: `user:///,jceks://file/tmp/test.jceks,jceks://hdfs@nn1.example.com/my/path/test.jceks`

indicates that the current user's credentials file should be consulted through the User Provider, that the local file located at `/tmp/test.jceks` is a Java Keystore Provider and that the file located within HDFS at `nn1.example.com/my/path/test.jceks` is also a store for a Java Keystore Provider.

When utilizing the credential command it will often be for provisioning a password or secret to a particular credential store provider. In order to explicitly indicate which provider store to use the `-provider` option should be used. Otherwise, given a path of multiple providers, the first non-transient provider will be used. This may or may not be the one that you intended.

Providers frequently require that a password or other secret is supplied. If the provider requires a password and is unable to find one, it will use a default password and emit a warning message that the default password is being used. If the `-strict` flag is supplied, the warning message becomes an error message and the command returns immediately with an error status.

Example: `hadoop credential list -provider jceks://file/tmp/test.jceks`

distch

Usage: `hadoop distch [-f urilist_url] [-i] [-log logdir] path:owner:group:permissions`

COMMAND_OPTION	Description
-f	List of objects to change
-i	Ignore failures
-log	Directory to log output

Change the ownership and permissions on many files at once.

distcp

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

dtutil

Usage: `hadoop dtutil [-keytab` *keytab_file* `-principal` *principal_name* `]` *subcommand* `[-format (java|protobuf)] [-alias` *alias* `] [-renewer` *renewer* `]` *filename…*

Utility to fetch and manage hadoop delegation tokens inside credentials files. It is intended to replace the simpler command `fetchdt`. There are multiple subcommands, each with their own flags and options.

For every subcommand that writes out a file, the `-format` option will specify the internal format to use. `java` is the legacy format that matches `fetchdt`. The default is `protobuf`.

For every subcommand that connects to a service, convenience flags are provided to specify the kerberos principal name and keytab file to use for auth.

SUBCOMMAND	Description

`print`
`[-alias` *alias* `]`
filename `[` *filename2* `...]`
	Print out the fields in the tokens contained in *filename* (and *filename2* …).
	If *alias* is specified, print only tokens matching *alias*. Otherwise, print all tokens.

SUBCOMMAND	Description
`get` *URL* `[-service` *scheme* `]` `[-format (java|protobuf)]` `[-alias` *alias* `]` `[-renewer` *renewer* `]` *filename*	Fetch a token from service at *URL* and place it in *filename*. *URL* is required and must immediately follow `get`. *URL* is the service URL, e.g. *hdfs://localhost:9000*. *alias* will overwrite the service field in the token. It is intended for hosts that have external and internal names, e.g. *firewall.com:14000*. *filename* should come last and is the name of the token file. It will be created if it does not exist. Otherwise, token(s) are added to existing file. The `-service` flag should only be used with a URL which starts with `http` or `https`. The following are equivalent: *hdfs://localhost:9000/* vs. *http://localhost:9000* `-service` *hdfs*
`append` `[-format (java|protobuf)]` *filename filename2* `[` *filename3* `...]`	Append the contents of the first N filenames onto the last filename. When tokens with common service fields are present in multiple files, earlier files' tokens are overwritten. That is, tokens present in the last file are always preserved.
`remove -alias` *alias* `[-format (java|protobuf)]` *filename* `[` *filename2* `...]`	From each file specified, remove the tokens matching *alias* and write out each file using specified format. *alias* must be specified.
`cancel -alias` *alias* `[-format (java|protobuf)]` *filename* `[` *filename2* `...]`	Just like `remove`, except the tokens are also cancelled using the service specified in the token object. *alias* must be specified.
`renew -alias` *alias* `[-format (java|protobuf)]` *filename* `[` *filename2* `...]`	For each file specified, renew the tokens matching *alias* and write out each file using specified format. *alias* must be specified.

fs

This command is documented in the File System Shell Guide. It is a synonym for `hdfs dfs` when HDFS is in use.

gridmix

Gridmix is a benchmark tool for Hadoop cluster. More information can be found in the Gridmix Guide.

jar

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

Runs a jar file.

Use `yarn jar` to launch YARN applications instead.

jnipath

Usage: `hadoop jnipath`

Print the computed java.library.path.

kerbname

Usage: `hadoop kerbname principal`

Convert the named principal via the auth_to_local rules to the Hadoop user name.

Example: `hadoop kerbname user@EXAMPLE.COM`

kdiag

Usage: `hadoop kdiag`

Diagnose Kerberos Problems

key

Usage: `hadoop key <subcommand> [options]`

COMMAND_OPTION	Description
create *keyname* [-cipher *cipher*] [-size *size*] [-description *description*] [-attr *attribute=value*] [-provider *provider*] [-strict] [-help]	Creates a new key for the name specified by the *keyname* argument within the provider specified by the `-provider` argument. The `-strict` flag will cause the command to fail if the provider uses a default password. You may specify a cipher with the `-cipher` argument. The default cipher is currently "AES/CTR/NoPadding". The default keysize is 128. You may specify the requested key length using the `-size` argument. Arbitrary attribute=value style attributes may be specified using the `-attr` argument. `-attr` may be specified multiple times, once per attribute.
roll *keyname* [-provider *provider*] [-strict] [-help]	Creates a new version for the specified key within the provider indicated using the `-provider` argument. The `-strict` flag will cause the command to fail if the provider uses a default password.
delete *keyname* [-provider *provider*] [-strict] [-f] [-help]	Deletes all versions of the key specified by the *keyname* argument from within the provider specified by `-provider`. The `-strict` flag will cause the command to fail if the provider uses a default password. The command asks for user confirmation unless `-f` is specified.
list [-provider *provider*] [-strict] [-metadata] [-help]	Displays the keynames contained within a particular provider as configured in core-site.xml or specified with the `-provider` argument. The `-strict` flag will cause the command to fail if the provider uses a default password. `-metadata` displays the metadata.
-help	Prints usage of this command

Manage keys via the KeyProvider. For details on KeyProviders, see the Transparent Encryption Guide.

Providers frequently require that a password or other secret is supplied. If the provider requires a password and is unable to find one, it will use a default password and emit a warning message that the default password is being used. If the `-strict` flag is supplied, the warning message becomes an error message and the command returns immediately with an error status.

NOTE: Some KeyProviders (e.g. org.apache.hadoop.crypto.key.JavaKeyStoreProvider) do not support uppercase key names.

NOTE: Some KeyProviders do not directly execute a key deletion (e.g. performs a soft-delete instead, or delay the actual deletion, to prevent mistake). In these cases, one may encounter errors when creating/deleting a key with the same name after deleting it. Please check the underlying KeyProvider for details.

kms

Usage: `hadoop kms`

Run KMS, the Key Management Server.

trace

View and modify Hadoop tracing settings. See the Tracing Guide.

version

Usage: `hadoop version`

Prints the version.

CLASSNAME

Usage: `hadoop CLASSNAME`

Runs the class named `CLASSNAME`. The class must be part of a package.

envvars

Usage: `hadoop envvars`

Display computed Hadoop environment variables.

Administration Commands

Commands useful for administrators of a hadoop cluster.

daemonlog

Usage:

```
hadoop daemonlog -getlevel <host:port> <classname> [-protocol (http|https)]
hadoop daemonlog -setlevel <host:port> <classname> <level> [-protocol (http|https)]
```

COMMAND_OPTION	Description
`-getlevel` *host:port classname* [-protocol (httphttps)]	Prints the log level of the log identified by a qualified *classname*, in the daemon running at *host:port*. The `-protocol` flag specifies the protocol for connection.
`-setlevel` *host:port classname level* [-protocol (httphttps)]	Sets the log level of the log identified by a qualified *classname*, in the daemon running at *host:port*. The `-protocol` flag specifies the protocol for connection.

Get/Set the log level for a Log identified by a qualified class name in the daemon dynamically. By default, the command sends a HTTP request, but this can be overridden by using argument `-protocol https` to send a HTTPS request.

Example:

```
$ bin/hadoop daemonlog -setlevel 127.0.0.1:9870 org.apache.hadoop.hdfs.server.namenode.NameNode DE
$ bin/hadoop daemonlog -getlevel 127.0.0.1:9871 org.apache.hadoop.hdfs.server.namenode.NameNode DE
```

Note that the setting is not permanent and will be reset when the daemon is restarted. This command works by sending a HTTP/HTTPS request to the daemon's internal Jetty servlet, so it supports the following daemons:

- Common
  - key management server
- HDFS
  - name node
  - secondary name node
  - data node
  - journal node
  - HttpFS server
- YARN
  - resource manager
  - node manager
  - Timeline server

Files

etc/hadoop/hadoop-env.sh

This file stores the global settings used by all Hadoop shell commands.

etc/hadoop/hadoop-user-functions.sh

This file allows for advanced users to override some shell functionality.

~/.hadooprc

This stores the personal environment for an individual user. It is processed after the hadoop-env.sh and hadoop-user-functions.sh files and can contain the same settings.