Splunk® Enterprise Admin Manual 7.3.1

Administrative CLI commands

Generated: 9/19/2019 8:33 am

Administrative CLI commands

This topic discusses the administrative CLI commands, which are the commands used to manage or configure your Splunk server and distributed deployment.

For information about accessing the CLI and what is covered in the CLI help, see the previous topic, Get help with the CLI. If you're looking for details about how to run searches from the CLI, see About CLI searches in the *Search Reference*.

Your Splunk role configuration dictates what actions (commands) you can execute. Most actions require you to have Splunk admin privileges. Read more about setting up and managing Splunk users and roles in the About users and roles topic in the *Admin Manual*.

Splunk CLI command syntax

The general syntax for a CLI command is this:

```
./splunk <command> [<object>] [[-<parameter>] <value>]...
```

Note the following:

- Some commands don't require an object or parameters.
- Some commands have a default parameter that can be specified by its value alone.
- Some commands can take extra parameters like `-uri` or `-auth`. See the "Universal parameters" section of Get help with the CLI.

Commands, objects, and examples

A **command** is an action that you can perform. An **object** is something you perform an action on.

Command	Objects	Examples
add	exec, forward-server, index, licenser-pools, licenses, master, monitor, oneshot, saved-search, search-server, tcp, udp, user	**1.** Adds monitor directory and file inputs to source `/var/log`. `./splunk add monitor /var/log/`
		2. Adds another master to the list of instances the searchhead searches across.

Command	Objects	Examples
		`./splunk add cluster-master https://127.0.0.1:8089 -secret testsecret -multisite false'`
anonymize	source	**1.** Replaces identifying data, such as usernames and IP addresses, in the file located at `/tmp/messages`. `./splunk anonymize file -source /tmp/messages`
		2. Anonymizes `Mynames.txt` using name-terms, a file containing a list of common English personal names. `./splunk anonymize file -source /tmp/messages -name_terms $SPLUNK_HOME/bin/Mynames.txt`
apply	cluster-bundle	**1.** Makes validated bundle active on peers. `./splunk apply cluster-bundle`
		2. Skip-validation is an optional argument to skip bundle validation on the master and peers. `./splunk apply cluster-bundle --skip-validation`
check-integrity	NONE	**1.** Verifies the integrity of an index with the optional parameter `verbose`. `./splunk check-integrity -index $SPLUNK_HOME/var/lib/splunk/defaultdb/ [-<verbose> ]`
		2. Verifies the integrity of a bucket with the optional parameter `verbose`. `./splunk check-integrity -bucketPath $SPLUNK_HOME/var/lib/splunk/defaultdb/db/ [-<verbose> ]`
clean	all, eventdata, globaldata, inputdata, userdata, kvstore	**1.** Removes data from Splunk installation. `eventdata` refers to exported events indexed as raw log files. `./splunk clean eventdata`

Command	Objects	Examples
		2. `globaldata` refers to host tags and source type aliases. `./splunk clean globaldata`

cmd	btool, classify, locktest, locktool, parsetest, pcregextest, regextest, searchtest, signtool, walklex	**1.** Runs the `splunk btool inputs list` command string with various environment variables set. Run `splunk envvars` to see which environment variables are set.
		`./splunk cmd btool inputs list`
		2. Shows contents of the bin directory.
		`./splunk cmd /bin/ls`

create	app	**1.** Builds myNewApp from a template. `./splunk create app myNewApp -template sample_app`
createssl	NONE	
diag	NONE	

disable	app, boot-start, deploy-client, deploy-server, dist-search, index, listen, local-index, maintenance-mode, perfmon, webserver, web-ssl, wmi	**1.** Disables the maintenance mode on peers in indexer clustering. Must be invoked at the master.
		`'./splunk disable maintenance-mode'`
		2. Disables the logs1 collection.
		`./splunk disable eventlog logs1`

display	app, boot-start, deploy-client, deploy-server, dist-search, jobs, listen, local-index	**1.** Displays status information, such as enabled/disabled, for all apps.
		`./splunk display app`
		2. Displays status information for the unix app.
		`./splunk display app unix`

edit	app, cluster-config, shcluster-config, exec, index, licenser-localslave,	**1.** Edits the current clustering configuration. `./splunk edit cluster-config -mode slave -site site2`

Command	Objects	Examples
	licenser-groups, monitor, saved-search, search-server, tcp, udp, user	**2.** Edits monitored directory inputs in `/var/log` and only reads from the end of this file. `./splunk edit monitor /var/log -follow-only true`
enable	app, boot-start, deploy-client, deploy-server, dist-search, index, listen, local-index, maintenance-mode, perfmon, webserver, web-ssl, wmi	**1.** Sets the maintenance mode on peers in indexer clustering. Must be invoked at the master. `'./splunk enable maintenance-mode'`
		2. Enables the `col1` collection. `./splunk enable perfmon col1`
export	eventdata, user data	**1.** Exports data out of your Splunk server into `/tmp/apache_raw_404_logs.` `./splunk export eventdata -index my_apache_data -dir /tmp/apache_raw_404_logs -host localhost -terms "404 html"`
fsck	repair, scan, clear-bloomfilter	
help	NONE	
import	userdata	**1.** Imports user accounts data from directory `/tmp/export.dat.` `./splunk import userdata -dir /tmp/export.dat`
install	app	**1.** Installs the app from foo.tar to the local Splunk server. `./splunk install app foo.tar`
		2. Installs the app from foo.tgz to the local Splunk server. `./splunk install app foo.tgz`

Command	Objects	Examples

list	cluster-buckets, cluster-config, cluster-generation, cluster-peers, deploy-clients, excess-buckets, exec, forward-server, index, inputstatus, licenser-groups, licenser-localslave, licenser-messages, licenser-pools, licenser-slaves, licenser-stacks, licenses, jobs, master-info, monitor, peer-info, peer-buckets, perfmon, saved-search, search-server, tcp, udp, user, wmi	**1.** Lists all active monitored directory and file inputs. This displays files and directories currently or recently monitored by splunkd for change.
		`./splunk list monitor`
		2. Lists all licenses across all stacks.
		`./splunk list licenses`

login,logout	NONE	

offline	NONE	**1.** Used to shutdown the peer in a way that does not affect existing searches. The master rearranges the primary peers for buckets, and fixes up the cluster state in case the enforce-counts flag is set.
		`./splunk offline`
		2. Because the `--enforce-counts` flag is used, the cluster is completely fixed up before this peer is taken down.
		`./splunk offline --enforce-counts`

package	app	**1.** Packages the stubby app and returns its uri. `./splunk package app stubby`
rebalance	cluster-data	**1.** Rebalances data for all indexes.

Command	Objects	Examples
		`./splunk rebalance cluster-data -action start`
		2. Rebalances data for a single index using the optional `-index` parameter. `./splunk rebalance cluster-data -action start -index $SPLUNK_HOME/var/lib/splunk/defaultdb/`
		3. Rebalances data using the optional `-max_runtime` parameter to limit the rebalancing activity to 5 minutes. `./splunk rebalance cluster-data start -max_runtime interval_: 5`
rebuild	NONE	
refresh	deploy-clients	**1.** Reloads your deployment server, in entirety or by server class. `./splunk reload deploy-server`
reload	ad, auth, deploy-server, exec, index, listen, monitor, registry, tcp, udp, perfmon, wmi	**2.** Reloads my_serverclass. `./splunk reload deploy-server -class my_serverclass`
		3. Reloads a specific index configuration. To reload all indexes, do not include an index name. `./splunk reload index [index_name]`
remove	app, cluster-peers, excess-buckets, exec, forward-server, index, jobs, licenser-pools, licenses, monitor, saved-search, search-server, tcp, udp, user	**1.** Removes the cluster master from the list of instances the searchhead searches across. Uses testsecret as the secret/pass4SymmKey. `'./splunk remove cluster-master https://127.0.0.1:8089 -secret testsecret'`
		2. Removes the Unix app. `./splunk remove app unix`
rollback	cluster-bundle	Rolls back your Splunk Web configuration bundle to your previous version. From the

Command	Objects	Examples
		master node, run this command: `./splunk rollback cluster-bundle`
rolling-restart	cluster-peers, shcluster-members	

rtsearch	app, batch, detach, earliest_time, header, id, index_earliest, index_latest, max_time, maxout, output, preview, rt_id, timeout, uri, wrap	**1.** Runs a real-time search that does not line-wrap for individual lines.
		`./splunk rtsearch 'error' -wrap false`
		2. Runs a real-time search. Use `rtsearch` exactly as you use the traditional search command.
		`./splunk rtsearch 'eventtype=webaccess error | top clientip'`

search	app, batch, detach, earliest_time, header, id, index_earliest, index_latest, latest_time, max_time, maxout, output, preview, timeout, uri, wrap	**1.** Uses the wildcard as the search object. Triggers an asynchronous search and displays the job id and ttl for the search.
		`./splunk search '*' -detach true`
		2. Uses `eventtype=webaccess error` as the search object. Does not line wrap for individual lines that are longer than the terminal width.
		`./splunk search 'eventtype=webaccess error' -wrap 0`

set	datastore-dir, deploy-poll, default-hostname, default-index, minfreemb, servername, server-type, splunkd-port, web-port, kvstore-port	**1.** Sets the force indexing ready bit.
		`./splunk set indexing-ready`
		2. Sets `bologna:1234` as the deployment server to poll updates from.
		`./splunk set deploy-poll bologna:1234`

show	config, cluster-bundle-status, datastore-dir, deploy-poll,	**1.** Shows current logging levels. `./splunk show log-level`

Command	Objects	Examples
	default-hostname, default-index, jobs, minfreemb, servername, splunkd-port, web-port, kvstore-port	**2.** Shows which deployment server Splunk Enterprise is configured to poll from. `./splunk show deploy-poll`
spool	NONE	
start,stop,restart	splunkd, splunkweb	
status	splunkd, splunkweb	
validate	index	**1.** Uses main as the index to validate. Verifies index paths specified in `indexes.conf`. `./splunk validate index main`
version	NONE	

Exporting search results with the CLI

You can use the CLI to export large numbers of search results. For information about how to export search results with the CLI, as well as information about the other export methods offered by Splunk Enterprise, see Export search results in the *Search Manual*.

Troubleshooting with the CLI

The Splunk CLI also includes tools that help with troubleshooting. Invoke these tools using the CLI command `cmd`:

```
./splunk cmd <tool>
```

For the list of CLI utilities, see Command line tools for use with Support in the *Troubleshooting Manual*.