

## Logsene Command-line Interface

Enables searching Logsene log entries from the command-line. Currently supports OS X and Linux.

## Installation

```
npm install logsene-cli -g
```

or

```
git clone https://github.com/sematext/logsene-cli.git
cd logsene-cli && npm install
npm link
```

To test, simply run:

```
npm test
```

## Logsene

Logsene is a centralized log management solution. You basically upload your logs in bulk or real-time from all your servers.

Those logs are then aggregated by time and shown in the Logsene web application, where you have the ability to search for specific terms, narrow down time ranges, filter fields, setup alerts, ...

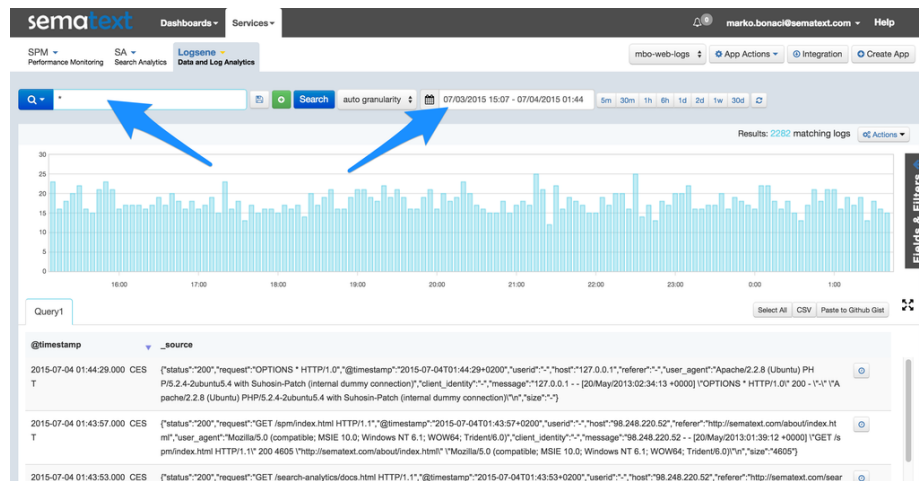


Figure 1:

A quick rundown of most notable features of Logsene:

- All your logs accessible in one place
- Control who sees which data
- Be up and running within minutes – there is nothing to install or maintain
- Log Alerts & Anomaly Detection
- Saved Searches
- Scheduled Reporting
- CoreOS Log Collection
- Docker container monitoring
- REST API

## Logsene CLI

Logsene CLI gives you capability to search through your logs from the command-line, which brings the awesome benefit of being able to pipe results to `awk`, `sed`, `cut`, `sort`, `head` and friends from the \*nix command-line.

Imagine a situation where you suspected that your site were under a DoS attack. You'd be interested in quickly finding out the top offenders. Here's a one-liner that shows top originating IP addresses in the last 3 hours (also shows how to use the `-f` switch to specify which field(s) to return - field `host`, in this example):

```
$ logsene search -t 3h -f host | sort | uniq -c | sort -r | head -n20
```

```
(master) ⚡% logsene search -t 3h -f host | sort | uniq -c | sort -r | head -n20
42 127.0.0.1
21 5.63.151.190
12 66.249.75.170
11 198.211.107.216
5 96.44.151.3
5 222.129.236.222
5 123.24.183.243
4 98.251.61.191
4 24.24.242.234
4 126.194.20.224
3 177.98.224.187
2 78.248.176.49
2 72.14.199.31
2 66.249.75.31
2 49.249.30.188
2 212.60.85.80
2 208.115.113.92
2 208.115.113.82
2 180.76.6.17
2 178.84.251.33
(master) ⚡% 
```

Figure 2:

You can find more useful examples in the blog post that announced the release of Logsene CLI.

## Logsene CLI Session

We define L-CLI session as a set of commands issued by the user, with no more than 30m between them. Every session has a set of configuration parameters that control the way L-CLI behaves. E.g. which Sematext account is used (`--api-key`); which Logsene application is used (`--app-key`); is tracing information going to be displayed (`-trace`).

For controlling those settings, we use `config set` and `config get` commands. For convenience reasons, you don't have to deal with API and APP keys manually. L-CLI automatically retrieves both keys on each session start, as users login (`--api-key`) and choose Logsene application (`--app-key`). L-CLI then writes those parameters to the session configuration store and reuses them on each subsequent command, until the session times out.

```
(master) % logsene search
No active sessions. Please log in using your Sematext account:
Enter your username: marko.bonaci@sematext.com
Enter your password:
Successfully logged in and retrieved API key.
1. mbo-cli
2. mbo-web-logs
Choose a Logsene app that you want to work with: 2
Successfully established Logsene mbo-web-logs application session.
2015-07-03T10:47:24.478068+00:00      222.187.0.221  NOTICE user      httpd: httpd 04
https://www.google.com/ Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3) AppleWebKit/537.31
- - [02/July/2015:23:17:01 +0200] GET /products/mlt/23221/ HTTP/1.1 200 5811 https://www.g
, like Gecko) Chrome/26.0.1410.65 Safari/537.31 5811      httpd
```

Figure 3:

The session primitives were introduced in order to enable frictionless multi-user experience, where all users may possibly be accessing L-CLI from the same box (while being SSHd into it), using the same Sematext account and possibly even the same Logsene application.

## Commands

`logsene search`

**Usage:** `logsene search [query] [OPTIONS]`

where OPTIONS may be:

- `-q <query>` Query string (`-q` parameter can be omitted)
- `-f <fields>` OPTIONAL Fields to return (defaults to all fields)
- `-t <interval>` OPTIONAL datetime, duration or range (defaults to last hour)

<code>-s &lt;size&gt;</code>	OPTIONAL Number of matches to return (default to 200)
<code>-o &lt;offset&gt;</code>	OPTIONAL Number of matches to skip from the beginning (defaults to 0)
<code>-op AND</code>	OPTIONAL Overrides default OR operator between multiple query terms
<code>--json</code>	OPTIONAL Returns log entries in JSON instead of TSV format
<code>--sep</code>	OPTIONAL Sets the separator between start and end of time ranges

#### Examples:

`logsene search`

`returns` last 1h of log entries

`note:` default return limit of 200 hits is always in effect unless you `explicitly` change it with the `-s` switch (where `-s` without params `disables` the limit altogether)

`logsene search -q ERROR`

`returns` last 1h of log entries that contain the term ERROR

`logsene search ERROR`

`equivalent` to the previous example

`logsene search UNDEFINED SEGFAULT`

`returns` last 1h of log entries that have either of the terms

`note:` default operator is OR

`logsene search SEGFAULT Segmentation -op AND`

`returns` last 1h of log entries that have both terms

`note:` convenience parameter `--and` has the same effect

`logsene search -q "Server not responding"`

`returns` last 1h of log entries that contain the given phrase

`logsene search "rare thing" -t 1y8M4d8h30m2s`

`returns` all the log entries that contain the phrase "rare thing" reaching back to 1 year 8 months 4 days 8 hours 30 minutes and 2 seconds

`note:` when specifying duration, any datetime designator character can be `omited` (shown in the following two examples)

`note:` months must be specified with uppercase M (distinction from minutes)

`note:` minutes (m) `are` the default, so "m" can be omitted

`logsene search -t 1h30m`

`returns` all the log entries from the last 1,5h

`logsene search -t 90`

`equivalent` to the previous example (default time unit is minute)

`logsene search -t 2015-06-20T20:48`

`returns` all the log entries that were logged after the provided datetime

**note:** allowed formats listed at the bottom of this help message

**logsene** search -t "2015-06-20 20:28"

**returns** all the log entries that were logged after the provided datetime

**note:** if a parameter contains spaces, it must be enclosed in quotes

**logsene** search -t 2015-06-16T22:27:41/2015-06-18T22:27:41

**returns** all the log entries between the two provided timestamps

**note:** date range must either contain forward slash between datetimes,  
or a different range separator must be specified (next example)

**logsene** search -t "2015-06-16T22:27:41 TO 2015-06-18T22:27:41" --sep " TO "

**same** as previous command, except it sets the custom string separator that  
**denotes** a range

**note:** default separator is the forward slash (as per ISO-8601)

**note:** if a parameter contains spaces, it must be enclosed in quotes

**logsene** search -t "last Friday at 13/last Friday at 13:30"

**it** is also possible to use "human language" to designate datetime

**note:** it may be used only in place of datetime. Expressing range is not  
**possible** (e.g. "last friday between 12 and 14" is not allowed)

**note:** may yield unpredictable datetime values

**logsene** search -q ERROR -s 20

**returns** at most 20 log entries (within the last hour) **with** the term ERROR

**logsene** search ERROR -s 50 -o 20

**returns** chronologically sorted hits 21st to 71st (offset is 20)

**note:** default sort order is ascending (latest entries at the bottom)

**logsene** search --help

**outputs** this usage information

Allowed datetime formats:

YYYY[-]MM[-]DD(T, )[HH[:MM[:SS]]]

e.g.

YYYY-MM-DD HH:mm:ss

YYYY-MM-DDTHH:mm

YYYY-MM-DDHH:mm

YYYYMMDDTHH:mm

YYYYMMDD HH:mm

YYYY-MM-DD

YYYYMMDD

YYYY-MM-DD HHmm

YYYYMMDD HHmm

YYYY-MM-DDTHHmm

YYYYMMDDTHH:mm  
 YYYYMMDDTHHmm  
 YYYYMMDDTHH:mm  
 YYYY-MM-DDTHH:mm:ss  
 YYYYMMDDHHmm:ss

note: date part may be separated from time by T (ISO-8601) or space  
 note: if datetime contains a space, it must be enclosed in double quotes

Allowed duration format:

[Ny] [NM] [Nd] [Nh] [Nm] [Ns]

e.g.

1y2M8d22h8m48s

note: uppercase M must be used for months, lowercase m for minutes

note: if only a number is specified, it defaults to minutes

Allowed range formats

range can be expressed in all datetime/duration combinations:

datetime/datetime

datetime/(+|-)duration

duration/(+|-)duration

duration/datetime

note: / is default range separator; + or - sign is duration direction

note: duration must begin with either + or - when used in end of range position

The following table shows how ranges are calculated, given the different input parameters

-t parameter	range start	range end
2016-06-24T18:42	timestamp	now
2016-06-24T18:42/2016-06-24T18:52:30	timestamp	timestamp
2016-06-24T18:42/+1d	timestamp	timestamp + duration
2016-06-24T18:42/-1d	timestamp - duration	timestamp
2h30m8s	now - duration	now
2h/+1h	now - duration1	start + duration2
2h/-1h	now - duration1 - duration2	now - duration1
5d10h25/2016-06-24T18:42	now - duration	timestamp

note: all allowable datetime formats are also permitted when specifying ranges

note: disallowed range separators:

Y, y, M, D, d, H, h, m, S, s, -, +, P, p, T, t

Allowed "human" formats (all in local time):

- 10 minutes ago
- yesterday at 12:30pm
- last night (night becomes 19:00)
- last month
- last friday at 2pm
- 3 hours ago
- 2 weeks ago at 17
- wednesday 2 weeks ago
- 2 months ago
- last week saturday morning (morning becomes 06:00)

note: "human" format can only be used instead of date-time

note: it is not possible to express duration with "human" format (e.g. "from 2 to 3 this m

note: it is recommended to avoid human format, as it may yield unexpected results

logsene config set

Usage: logsene config set [OPTIONS]

where OPTIONS may be:

- api-key <apiKey>
- app-key <appKey>
- default-size <size>
- range-separator <sep>
- trace <true|false>

It is not necessary to explicitly set api-key nor app-key.

Logsene CLI will ask you to log in and choose Logsene application

if keys are missing from the configuration

Examples:

logsene config set --api-key 11111111-1111-1111-1111-111111111111  
sets the api key for the current session

logsene config set --app-key 22222222-2222-2222-2222-222222222222  
sets Logsene application key for the current session

logsene config set --default-size 3000  
sets default number of hits returned for the current session (overrides the default 20)

logsene config set --range-separator T0  
sets default separator of two datetimes for time ranges (default is /, as per ISO6801)

logsene config set --trace [true]  
activates tracing for the current session (true can be omitted)

```
logsene config set --trace false
  deactivates tracing for the current session
```

```
logsene config get
```

Usage: logsene config get [OPTION] Where OPTION may be:

```
--api-key
--app-key
--app-name
--default-size (sets the default number of hits returned for the current session)
--range-separator (used to separate start and end of a time range)
--trace
--all (return listing of all params from the current user's session)
```