
collectd-haproxy Documentation

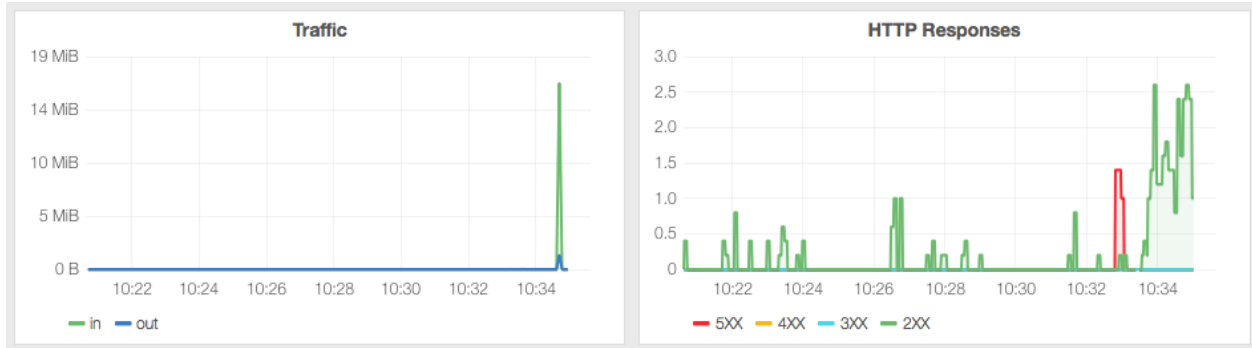
Release 1.2.1

William Glass

September 25, 2016

1	Installation	3
2	Configuration	5
2.1	Development	5
2.2	License	5
	Python Module Index	11

A plugin for `collectd` to gather metrics for a local `HAProxy` instance, with a focus on easy installation and configuration.



(graphs made with Grafana)

Installation

The project is on [PyPI](#) so installation is a simple `pip` command:

```
pip install collectd-haproxy
```

To install manually, first download the current tarball at [collectd-haproxy-1.2.1.tar.gz](#) then:

```
tar -zxvf collectd-haproxy-1.2.1.tar.gz
cd collectd-haproxy-1.2.1
python setup.py install
```

Note: If `collectd` is running with a `virtualenv` activated (under an isolated [supervisord](#) setup, for instance) make sure the `collectd-haproxy` package is installed and available in the `virtualenv`'s module path or `collectd` will run into an `ImportError`.

In order to function properly, the local HAProxy instance will need to have the “stats socket” enabled, details about how to do that can be found [here](#).

Configuration

The most basic configuration is to specify where the HAProxy socket file is located:

```
LoadPlugin "python"

<Plugin python>
  Import "collectd_haproxy"

  <Module haproxy>
    Socket "/var/run/haproxy.sock"
  </Module>
</Plugin>
```

For details on all of the options available, see the [Configuration docs](#).

2.1 Development

The code is hosted on [GitHub](#)

To file a bug or possible enhancement see the [Issue Tracker](#), also found on GitHub.

2.2 License

(c) 2015-2016 William Glass

collectd-haproxy is licensed under the terms of the MIT license. See the [LICENSE](#) file for more details.

2.2.1 Configuration

Configuring the collectd-haproxy plugin is done just like any other python-based plugin for collectd, for details see the [python plugin docs](#).

There are six available options (only the `Socket` option is required):

```
LoadPlugin "python"

<Plugin python>
  Import "collectd_haproxy"
```

```
<Module haproxy>
  Socket "/var/run/haproxy.sock"
  IncludeInfo true
  IncludeStats true
  IncludeFrontendStats true
  IncludeBackendStats true
  IncludeServerStats true
</Module>
</Plugin>
```

Socket (*required*)

This is the path where the HAProxy socket file is located, e.g. `/var/run/haproxy.sock`

IncludeInfo

A boolean value denoting whether or not to collect the “info” metrics that describe the state of the whole HAProxy process, metrics such as uptime seconds, current total connections, size of the run queue, etc.

Defaults to `true`

IncludeStats

Flag for whether to collect proxy “stats” metrics. These are detailed metrics for individual proxies in HAProxy, such as HTTP status 200 response count, bytes in/out, queued request count, etc. The full list of metrics available can be found in the [HAProxy ‘show stats’ docs](#).

More granular control of which stats to collectd is available via the [IncludeFrontendStats](#), [IncludeBackendStats](#) and [IncludeServerStats](#) options.

This option takes precedence over the more granular ones. That is, if `IncludeStats` is false, no proxy-level stats will be collected regardless of the other `Include*Stats` option values.

Defaults to `true`

IncludeFrontendStats

Granular flag for collecting stats for the “frontend” of a proxy, where connections are accepted.

Defaults to `true`

IncludeBackendStats

Granular flag for collecting aggregate stats for the “backend” of a proxy. Each proxy can have any number of individual servers in a backend, and these stats are aggregated across the whole lot.

Defaults to `true`

IncludeServerStats

Granular flag for collecting stats for individual servers that make up the backends of proxies.

Note: This doesn't include ways to filter specific servers, it merely determines whether stats at the individual server level are collected at all or not.

Defaults to `true`

2.2.2 Source Docs

`collectd_haproxy.plugin`

class `collectd_haproxy.plugin.HAProxyPlugin` (*collectd*)

Bases: `object`

The plugin class, workhorse that liasons between collectd and HAProxy.

Rather than being instantiated directly, the `register()` method should be used to make a plugin, since it both creates an instance *and* takes care of the proper callback registration.

HAProxy Plugin constructor

Since the collectd module is only available when the plugin is running in collectd's python process we use some dependency injection here.

Parameters `collectd` (*module*) – The collectd module.

name = 'haproxy'

classmethod `register` (*collectd*)

Registers the plugin's callbacks with the given collectd module.

Since the collectd module is only available when the plugin is running in collectd's python process we use some dependency injection here.

Parameters `collectd` (*module*) – The collectd module.

configure (*config*)

The 'configure' collectd callback for the plugin.

Iterates over the config object's `children` attribute and sets any applicable attributes on the plugin instance.

Parameters `config` (*collect.Config*) – The collectd Config instance. Passed in automatically by collectd itself.

initialize ()

The 'initialize' collectd callback for the plugin.

This callback fires after the 'config' one but before the 'read' one gets added to the loop.

Instantiates a `collectd.Values` for each known metric (these are used to dispatch actual values to collectd) as well as sets up the `HAProxySocket` for fetching the values.

read ()

The 'read' collectd callback for the plugin.

Simple method that calls `collect_info()` and/or `collect_stats()` based on the configuration.

collect_info()

Method for sending HAProxy “info” metrics to collectd.

Iterates over the metric names and values provided by the socket and dispatches each known one to collectd.

collect_stats()

Method for sending HAProxy “info” metrics to collectd.

Iterates over the metric names and values provided by the socket, checking that the metric is a known one and taking care of numeric coercion before dispatching to collectd.

collectd_haproxy.connection

class collectd_haproxy.connection.HAProxySocket(*collectd, socket_file_path*)

Bases: object

Class used for interacting with an HAProxy control socket.

Provides two methods for generating metrics, one for the “info” metrics that give process-wide details and the “stats” metrics for individual proxies/servers.

The HAProxySocket constructor.

Since the collectd module is only available when the plugin is running in collectd’s python process we use some dependency injection here.

Parameters

- **collectd** (*module*) – The collectd module.
- **socket_file_path** (*str*) – Full path to HAProxy’s socket file.

send_command(*command*)

Sends a given command to the HAProxy socket.

Collects the response (it can arrive in chunks) and then calls the `process_command_response` method on the result.

Parameters **command** (*str*) – The command to send, e.g. “show stat”

process_command_response(*command, response*)

Takes an HAProxy socket command and its response and either raises an appropriate exception or returns the formatted response.

Parameters

- **command** (*str*) – The command that was run.
- **response** (*str*) – The full response string from running the command.

gen_info()

Generator that yields (name, value) tuples for HAProxy info.

These values represent stats for the whole HAProxy process.

gen_stats(*include_frontends, include_backends, include_servers*)

Generator that yields (name, values) for individual proxies.

Each tuple has two items, the proxy and a dictionary mapping stat field names to their respective values.

Parameters

- **include_frontends** (*bool*) – Whether or not to include FRONTEND aggregate stats.

- **include_backends** (*bool*) – Whether or not to include BACKEND aggregate stats.
- **include_servers** (*bool*) – Whether or not to include individual server stats.

`collectd_haproxy.compat`

`collectd_haproxy.compat.iteritems` (*dictionary*)

Helper function for iterating over (key, value) dict tuples.

In Python 3 the “`iteritems()`” method went away and “`items()`” became an iterator method.

Parameters **dictionary** (*dict*) – The dictionary to iterate over.

`collectd_haproxy.compat.coerce_long` (*string*)

Function for coercing a string into a long (“10.4” -> 10.4).

In Python 3 the long and int types were unified, so no more “long”.

Parameters **string** (*str*) – The string value to coerce.

2.2.3 Release Notes

1.0.0

- Initial public release

1.0.1

- Some small updates to the documentation.

1.1.0

- Added some missing timing metrics, namely `qtime`, `ctime`, `rtime` and `ttime`.
- Switched to a new documentation theme.
- Added a badge for the pypi version to the README file, links to the pypi page.
- Fixed a style failure with an import not at the top of a file.
- Added a coverage report step to the Travis CI tests.

1.1.1

- Fixed bug where `collectd` module namespace collision would prevent installation.

1.2.0

- Fixed up python 3.5 support (hopefully)
- Test coverage now at 100%

1.2.1

- Add support for 2.6

C

`collectd_haproxy.compat`, 9
`collectd_haproxy.connection`, 8
`collectd_haproxy.plugin`, 7

C

`coerce_long()` (in module `collectd_haproxy.compat`), 9
`collect_info()` (`collectd_haproxy.plugin.HAProxyPlugin` method), 7
`collect_stats()` (`collectd_haproxy.plugin.HAProxyPlugin` method), 8
`collectd_haproxy.compat` (module), 9
`collectd_haproxy.connection` (module), 8
`collectd_haproxy.plugin` (module), 7
`configure()` (`collectd_haproxy.plugin.HAProxyPlugin` method), 7

G

`gen_info()` (`collectd_haproxy.connection.HAProxySocket` method), 8
`gen_stats()` (`collectd_haproxy.connection.HAProxySocket` method), 8

H

`HAProxyPlugin` (class in `collectd_haproxy.plugin`), 7
`HAProxySocket` (class in `collectd_haproxy.connection`), 8

I

`initialize()` (`collectd_haproxy.plugin.HAProxyPlugin` method), 7
`iteritems()` (in module `collectd_haproxy.compat`), 9

N

`name` (`collectd_haproxy.plugin.HAProxyPlugin` attribute), 7

P

`process_command_response()` (`collectd_haproxy.connection.HAProxySocket` method), 8

R

`read()` (`collectd_haproxy.plugin.HAProxyPlugin` method), 7

`register()` (`collectd_haproxy.plugin.HAProxyPlugin` class method), 7

S

`send_command()` (`collectd_haproxy.connection.HAProxySocket` method), 8