
staps

Release 0.1.6

July 20, 2016

1	Overview	1
1.1	Installation	1
1.2	Documentation	1
1.3	Development	1
2	Installation	3
3	Usage	5
3.1	systemd	6
3.2	nginx	6
3.3	Apache	6
4	Contributing	7
4.1	Bug reports	7
4.2	Documentation improvements	7
4.3	Feature requests and feedback	7
4.4	Development	7
5	Authors	9
6	Changelog	11
6.1	0.1.6 (2016-07-20)	11
6.2	0.1.5 (2016-07-19)	11
6.3	0.1.4 (2016-07-19)	11
6.4	0.1.3 (2016-07-18)	11
6.5	0.1.2 (2016-05-25)	11
6.6	0.1.1 (2016-05-25)	11
6.7	0.1.0 (2016-03-03)	12
7	Indices and tables	13

Overview

docs	
tests	
package	

Simple Throw-Away Publish/Subscribe. Create channels simply by connecting one or mote websocket clients to a random URL path on a webserver with staps. Not meant to be used with webbrowsers but as a cheap way to let multiple websocket clients written outside the browser communicate with each other.

- Free software: BSD license

1.1 Installation

```
pip install staps
```

1.2 Documentation

<https://staps.readthedocs.io/en/latest/>

1.3 Development

To run the all tests run:

```
tox
```

Note, to combine the coverage data from all the tox environments run:

Windows	set PYTEST_ADDOPTS=--cov-append tox
Other	PYTEST_ADDOPTS=--cov-append tox

Installation

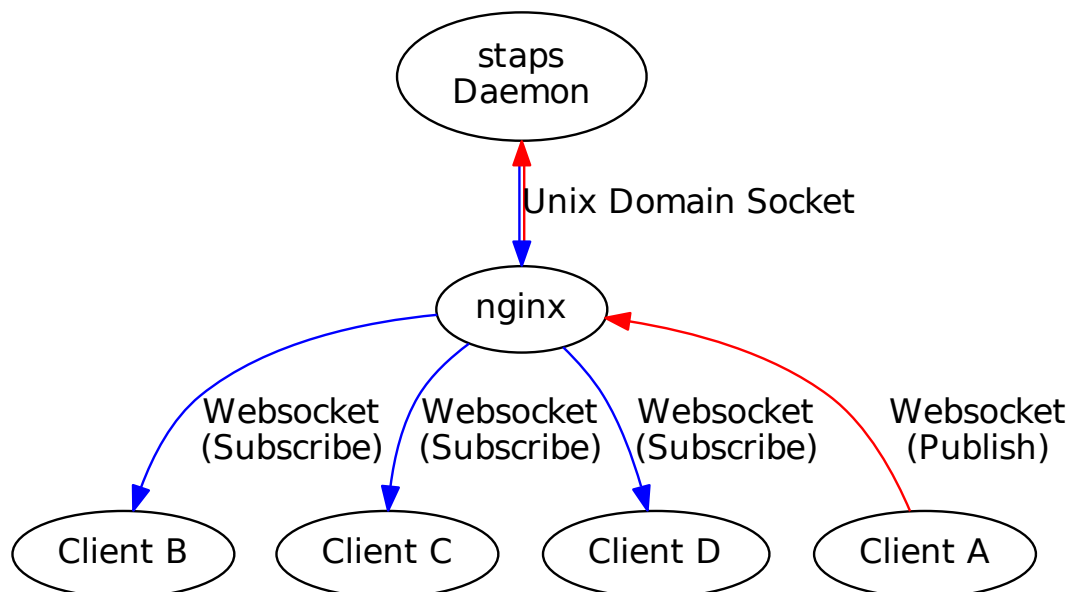
At the command line:

```
sudo pip install staps
```

Usage

staps provides publish-subscribe channels that are meant to be temporary. They can be created by simply connecting a websocket client to an URL. Each channel is identified by the URL path that is passed to staps. The first client to connect to a distinct path will create an anonymous pub/sub-channel. Each subsequent client that connects to the same URL will join the channel.

All clients in a channel are equal which means each one of them can publish messages while being subscribed to messages from other clients. Messages will not be echoed back to the client that published them.



staps is intended to be used as a daemon controlled by systemd. It listens for websocket connections on a unix domain socket. In order to make staps available over network a websocket-capable reverse-proxy like nginx is required.

Danger: There is no authentication or authorization. Everyone who knows the URL to a channel can join it and can snoop on published messages or can publish malicious messages themselves! URLs should be treated as confidential! Do not use staps with a webbrowser!

3.1 systemd

A service unit file for systemd is included with staps. To enable this unit copy the staps.service file to /etc/systemd/system/staps.service. Then run:

```
sudo systemctl enable staps.service
sudo systemctl start staps.service
```

The configuration file should be placed at /etc/staps/staps.conf or ~/.staps.conf:

```
[staps]
socket = /run/staps/staps.sock
mode = 0660
amqp = amqp://staps:@localhost/

[daemon]
user = staps
group = staps
dir = /var/lib/staps/
pid_file = /var/run/staps/staps.pid
umask = 0
```

3.2 nginx

Currently only nginx is supported as a frontend for staps. To have nginx handle all websocket connections at <http://example.com/<uuid4>> use the following example configuration. The regex for the location block matches only UUID4 paths and is the recommended way to generate throw-away pub/sub URLs.

```
map $http_upgrade $connection_upgrade {
    default upgrade;
    '' close;
}

upstream websocket {
    server unix:/run/staps/staps.sock;
}

server {
    ...

    location ~* "^[a-f0-9]{8}-[a-f0-9]{4}-4[a-f0-9]{3}-[89aAbB][a-f0-9]{3}-[a-f0-9]{12}$" {
        proxy_pass http://websocket;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;
    }

    ...
}
```

3.3 Apache

While Apache should be able to act as a frontend for staps it is currently not possible to use mod_proxy_wstunnel with unix domain sockets.

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

4.1 Bug reports

When [reporting a bug](#) please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.2 Documentation improvements

staps could always use more documentation, whether as part of the official staps docs, in docstrings, or even on the web in blog posts, articles, and such.

4.3 Feature requests and feedback

The best way to send feedback is to file an issue at <https://github.com/fladi/staps/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that code contributions are welcome :)

4.4 Development

To set up *staps* for local development:

1. Fork [staps](#) (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/staps.git
```

3. Create a branch for local development:

```
git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

4. When you're done making changes, run all the checks, doc builder and spell checker with `tox` one command:

```
tox
```

5. Commit your changes and push your branch to GitHub:

```
git add .
git commit -m "Your detailed description of your changes."
git push origin name-of-your-bugfix-or-feature
```

6. Submit a pull request through the GitHub website.

4.4.1 Pull Request Guidelines

If you need some code review or feedback while you're developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`)¹.
2. Update documentation when there's new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

4.4.2 Tips

To run a subset of tests:

```
tox -e envname -- py.test -k test_myfeature
```

To run all the test environments in *parallel* (you need to `pip install detox`):

```
detox
```

¹ If you don't have all the necessary python versions available locally you can rely on Travis - it will [run the tests](#) for each change you add in the pull request.
It will be slower though ...

Authors

- Michael Fladischer - <https://openservices.at/>

Changelog

6.1 0.1.6 (2016-07-20)

- Do not build universal wheel as Python2 is not supported.

6.2 0.1.5 (2016-07-19)

- Improved packaging to include missing templates.

6.3 0.1.4 (2016-07-19)

- Add HTTP handler to show usage information and autogenerate UUID4 URLs.
- Add CLI option to listen directly on TCP ports.

6.4 0.1.3 (2016-07-18)

- Improved documentation.

6.5 0.1.2 (2016-05-25)

- Include requirements.txt in source distribution.
- Reorganize imports using isort.

6.6 0.1.1 (2016-05-25)

- Include staps.conf in MANIFEST.in.
- Use preferred socket path in usage documentation.

6.7 0.1.0 (2016-03-03)

- First release on PyPI.

Indices and tables

- `genindex`
- `modindex`
- `search`