
s3tail Documentation

Release 0.2.1

Brad Robel-Forrest

January 07, 2017

1	s3tail	3
1.1	Features	3
1.2	Credits	4
2	Installation	5
2.1	Stable release	5
2.2	From sources	5
3	Usage	7
3.1	Configuration	7
3.2	Basic Console Example	8
3.3	Coding Example	8
3.4	GoAccess Example	8
4	Contributing	11
4.1	Types of Contributions	11
4.2	Get Started!	12
4.3	Pull Request Guidelines	12
4.4	Tips	13
5	Credits	15
5.1	Development Lead	15
5.2	Contributors	15
6	History	17
6.1	0.2.1 (2016-12-27)	17
6.2	0.2.0 (2016-12-27)	17
6.3	0.1.7 (2016-09-18)	17
6.4	0.1.6 (2016-09-12)	17
6.5	0.1.5 (2016-09-12)	17
6.6	0.1.4 (2016-09-11)	17
6.7	0.1.3 (2016-09-11)	18
6.8	0.1.2 (2016-09-07)	18
6.9	0.1.1 (2016-08-29)	18
6.10	0.1.0 (2016-08-25)	18
7	Indices and tables	19

Contents:

s3tail

S3tail is a simple tool to help access log files stored in an S3 bucket in the same way one might use the *nix `tail` command (with far fewer options, most notably the lack of `follow`).

- Free software: MIT license
- Documentation: <https://s3tail.readthedocs.io>.

Simplest install method is via `pip install s3tail` (see [installation](#) for other methods).

1.1 Features

S3tail downloads and displays the content of files stored in S3, optionally starting at a specific prefix. For example, the following will start dumping all the log file contents found for August the fourth in the order S3 provides from that prefix onward:

```
$ s3tail s3://my-logs/production-s3-access-2016-08-04
```

When `s3tail` is stopped or interrupted, it'll print a bookmark to be used to pick up at the exact spot following the last log printed in a previous run. Something like the following might be used to leverage this ability to continue tailing from a previous stopping point:

```
$ s3tail s3://my-logs/production-s3-access-2016-08-04
...
...a-bunch-of-file-output...
...
Bookmark: production-s3-access-2016-08-04-00-20-31-61059F36E0DBF36E:706
```

This can then be used to pick up at line 707 later on, like this:

```
$ s3tail s3://my-logs/production-s3-access-2016-08-04 \
  --bookmark production-s3-access-2016-08-04-00-20-31-61059F36E0DBF36E:706
```

Additionally, it's often useful to let `s3tail` track where things were left off and pick up at that spot without needing to copy and paste the previous bookmark. This is where “named bookmarks” come in handy. The examples above could have been reduced to these operations:

```
$ s3tail --bookmark my-special-spot s3://my-logs/production-s3-access-2016-08-04
...
^C
$ s3tail --bookmark my-special-spot s3://my-logs/production-s3-access
Starting production-s3-access-2016-08-04-02-22-32-415AE699C8233AC3
Found production-s3-access-2016-08-04-02-22-32-415AE699C8233AC3 in cache
```

```
Picked up at line 707
...
```

It's safe to rerun s3tail sessions when working with piped commands searching for data in the stream (e.g. `grep`). S3tail keeps files in a local file system cache (for 24 hours by default) and will always read and display from the cache before downloading from S3. This is done in a best-effort background thread to avoid impacting performance. The file cache is stored in the user's HOME directory, in an `.s3tailcache` subdirectory, where the file names are the S3 keys hashed with SHA-256. These can be listed through the use of the `--cache-lookup` option:

```
$ s3tail --cache-lookup s3://my-logs/production-s3-access-2016-08-04

my-logs/production-s3-access-2016-08-04-23-20-40-9935D31F89E5E38B
=> NOT IN CACHE
my-logs/production-s3-access-2016-08-04-23-20-45-D76C63A0478F829B
=> NOT IN CACHE
my-logs/production-s3-access-2016-08-04-23-20-51-C14A8D0980A9F562
=> NOT IN CACHE
...
my-logs/production-s3-access-2016-08-04-23-24-02-C9DF441E6B14EFBB
=> /Users/brad/.s3tailcache/05/0536db5ed3938c0b7fb8d2809bf8b4eb1a686ba14c9dc9b09aafc20780ef0528
my-logs/production-s3-access-2016-08-04-23-24-10-E9E55E9019AA46D0
=> /Users/brad/.s3tailcache/d1/d1c8b060d7c9a59c6387fc93b7a3d42db09ce90df2ed4eb71449e88e010ab4a8
my-logs/production-s3-access-2016-08-04-23-24-58-28FE2F9927BCBEA3
=> /Users/brad/.s3tailcache/46/46de81db7cd618074a8ff24cef938dca0d8353da3af8ccc67f517ba8600c3963
```

Check out [usage](#) for more details and examples (like how to leverage GoAccess to generate beautiful traffic reports!).

1.2 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

Installation

2.1 Stable release

To install s3tail, run this command in your terminal:

```
$ pip install s3tail
```

This is the preferred method to install s3tail, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for s3tail can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/bradrf/s3tail
```

Or download the `tarball`:

```
$ curl -OL https://github.com/bradrf/s3tail/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


Usage

```

$ s3tail --help

Usage: s3tail [OPTIONS] S3_URI

  Begins tailing files found at [s3://]BUCKET[/PREFIX]

Options:
  --version                Show the version and exit.
  -c, --config-file PATH  Configuration file [default:
                          /Users/brad/.s3tailrc]
  -r, --region [us-east-1|us-west-1|us-gov-west-1|ap-northeast-2|ap-northeast-1|sa-east-1|eu-central-1]
                          AWS region to use when connecting
  -b, --bookmark TEXT     Bookmark to start at (key:line or a named
                          bookmark)
  -l, --log-level [debug|info|warning|error|critical]
                          set logging level
  --log-file FILENAME     write logs to FILENAME
  --cache-hours INTEGER   Number of hours to keep in cache before
                          removing on next run (0 disables caching)
  --cache-lookup          Report if s3_uri keys are cached (showing
                          pathnames if found)
  -h, --help              Show this message and exit.

```

3.1 Configuration

Follow the instructions provided by the Boto Python interface to AWS: http://boto.cloudhackers.com/en/latest/boto_config_tut.html

Optionally, following can be configured to override the defaults by editing a configuration file. Normally, this file stores bookmark information, but can also include a section for setting command line options.

An example might look like this (usually lives in the executing user's HOME directory as `.s3tailrc`):

```

[bookmarks]
barf = production/s3/collab-production-s3-access-2016-09-11-02-26-19-718F6332DA1867B6:2935
last-look = production/s3/collab-production-s3-access-2016-09-18-21-27-17-79EB845D49F9F7E9:1611

[options]
cache_hours = 1
cache_path = /Users/brad/.s3tailcache
log_level = warn

```

Option descriptions:

- `cache_hours`: Any integer describing the number of hours to keep items in the cache before they are discarded (can be a value of zero to disable the cache entirely).
- `cache_path`: The full pathname to a directory for storing cached files when downloading from S3.
- `log_file`: The full pathname to a file for writing all log output (only logs from s3tail; content extracted from S3 files is always written to standard output (STDOUT)).
- `log_level`: Any one of `debug`, `info`, `warning`, `error`, or `critical`.
- `region`: The AWS region for accessing S3 (see http://docs.aws.amazon.com/general/latest/gr/rande.html#s3_region).

Any options specified on the command line itself always will have preference over those stated in the configuration file.

3.2 Basic Console Example

```
$ s3tail s3://my-logs/production-s3-access-2016-08-04
```

3.3 Coding Example

To use the `s3tail.S3Tail` class in a project:

```
from s3tail import S3Tail
from configparser import ConfigParser

def process_line(num, line):
    print '%d: %s' % (num, line)

config = ConfigParser() # stores the bookmarks
tail = S3Tail(config, 'my-logs', 'production-s3-access-2016-08-04', process_line)
tail.watch()
tail.cleanup()

print 'stopped at bookmark ' + tail.get_bookmark()
```

3.4 GoAccess Example

A great use for s3tail is as a data provider to the amazing [GoAccess](#) utility that can provide beautiful visualization of traffic logs.

First, build [GoAccess](#) with the ability track incremental progress in a local database. The following works when building on Ubuntu Trusty:

```
$ wget http://tar.goaccess.io/goaccess-1.0.2.tar.gz
$ apt-get install libgeoip-dev libncursesw5-dev libtokyocabinet-dev libz-dev libbz2-dev
$ ./configure --enable-geoip --enable-utf8 --enable-tcb=btree --with-getline
$ make
```

```
$ make install
```

Next, build a configuration file for `GoAccess`. The `log-format` should match nicely with the [S3 Log Format](#). Many `GoAccess` configuration options are available, but the following works quite well (e.g. placed in `~/.goaccessrc_s3`):

```
date-format %d/%b/%Y
time-format %H:%M:%S %z
log-format %^ %v [%d:%t] %h %^ %^ %^ %^ "%m %U %H" %s %^ %b %^ %L %^ "%R" "%u" %~
agent-list true
4xx-to-unique-count true
with-output-resolver true
load-from-disk true
keep-db-files true
```

Periodically, run something like the following to download and analyze traffic reported into an S3 bucket. Through the use of `s3tail`'s named bookmark (`goaccess-traffic` in the example below), each successive run will pick up where `s3tail` left off on the previous run, continuing to read and feed logs into `GoAccess`:

```
$ s3tail --log-file /var/log/s3tail.log -b goaccess-traffic my-logs/production-s3-access-2016-08-04
  goaccess -p ~/.goaccessrc_s3 -o ~/report.json
```

At any time, `GoAccess` can view the current dataset via its wonderful CLI, generate a self-contained HTML report, or make use of the live preview provided via a websocket (e.g. <http://rt.goaccess.io/> is a live demo)!

```
$ goaccess -p ~/.goaccessrc_s3
```

Contributing

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

You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at <https://github.com/bradrf/s3tail/issues>.

If you are 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.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

4.1.4 Write Documentation

s3tail could always use more documentation, whether as part of the official s3tail docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/bradrf/s3tail/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 contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *s3tail* for local development.

1. Fork the *s3tail* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/s3tail.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv s3tail
$ cd s3tail/
$ python setup.py develop
```

4. Create a branch for local development:

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

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 s3tail tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. 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
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.

2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/bradrf/s3tail/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_s3tail
```


Credits

5.1 Development Lead

- Brad Robel-Forrest <brad@bitpony.com>

5.2 Contributors

None yet. Why not be the first?

History

6.1 0.2.1 (2016-12-27)

- Documentation.

6.2 0.2.0 (2016-12-27)

- Add gunzip for *.gz files found (based only on extension name for now).
- Save configuration using ConfigStruct w/ overridable values.

6.3 0.1.7 (2016-09-18)

- Fix incorrect final bookmark when no more logs to read from key.

6.4 0.1.6 (2016-09-12)

- Documentation.

6.5 0.1.5 (2016-09-12)

- Documentation.

6.6 0.1.4 (2016-09-11)

- Fix bug in prefix matching when using named bookmarks.
- Added timestamps to logs.

6.7 0.1.3 (2016-09-11)

- Added “named” bookmarks to pick up automatically from last position when possible.
- Added option to disable cache entirely.

6.8 0.1.2 (2016-09-07)

- Better perf when reading from cache.
- Improved docs.

6.9 0.1.1 (2016-08-29)

- Refactor into classes and provide some minimal docs.

6.10 0.1.0 (2016-08-25)

- First release on PyPI.

Indices and tables

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