s3workers Documentation

Release 0.3.0

Brad Robel-Forrest

Contents

Contents:

s3workers

Helper to simplify concurrent access to object scanning in AWS S3 buckets.

- Free software: MIT license
- Documentation: https://s3workers.readthedocs.io.

Simplest install method is via `pip install s3workers` (see installation for other methods).

1.1 Features

S3workers provides faster list and delete operations on S3 buckets by opening up simultaneous connections to issue distinct sets of shared prefix queries. Effectively, this splits up the query space into 36 independent queries (26 alpha and 10 numeric prefixes). For example, a request to list all objects in the `myfancy/` bucket would result in concurrent list queries to S3 for everything from `myfancy/a...` through `myfancy/b...` and everything from `myfancy/0...` through `myfancy/9...`, all at the same time, reporting and collating the results locally.

1.1.1 Selection

The default output of s3workers is to simply list (or delete) all objects found at the prefix requested. However, often it is advantageous to restrict the output to only those matching certain criteria. The `--select` option provides the ability for evaluating matches using any normal Python operators or builtins against one or more of the following variables provided to the selector for each object found:

- `name`: The full S3 key name, everything *except* the bucket name (string)
- `size`: The number of bytes as used by the S3 object (integer).
- `md5`: The MD5 hash of the S3 object (string).
- `last_modified`: The timestamp indicating the last time the S3 object was changed (string).

1.1.2 Reduction

In cases where aggregation of some kind is desired, s3workers provides the ability to execute reduction logic against an accumulator value. For example, to produce a sum of the size of all selected S3 objects or to even group the size according to MD5 values. See the usage output for examples. In all cases, the same variables provided by selection are also provided when reducing.

1.1.3 Usage

For more help and examples of selection and reduction abilities, check out the usage.

1.2 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

Installation

2.1 Stable release

To install s3workers, run this command in your terminal:

```
$ pip install s3workers
```

This is the preferred method to install s3workers, 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 s3workers can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/bradrf/s3workers
```

Or download the tarball:

```
$ curl  -OL https://github.com/bradrf/s3workers/tarball/master
```

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

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

Usage

3.1 To use s3workers from the command line (CLI)

```
$ s3workers --help

Usage: s3workers [OPTIONS] COMMAND S3_URI

  Perform simple listing, collating, or deleting of many S3 objects at the
  same time.

  Examples:

      List empty objects:
          s3workers list --select 'size == 0' s3://mybucket/myprefix

      Report total of all non-empty objects:
          s3workers list --select 'size > 0' --reduce 'accumulator += size' s3://
→mybucket/myprefix

      Total size group by MD5:
          s3workers list --accumulator '{}' --reduce 'v=accumulator.get(md5,0)+size;␣
→accumulator[md5]=v' s3://mybucket/myprefix

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|ap-southeast-1|ca-central-1|ap-southeast-2|us-west-2|us-east-
→2|ap-south-1|cn-north-1|eu-west-1|eu-west-2]
                                AWS region to use when connecting
  -l, --log-level [debug|info|warning|error|critical]
                                set logging level
  --log-file FILENAME           write logs to FILENAME
```

```
  --concurrency INTEGER       set number of workers processing jobs
                              simultaneously  [default: 36]
  --select TEXT               provide comparisons against object name,
                              size, md5, or last_modified to limit
                              selection
  --reduce TEXT               provide reduction logic against the
                              accumulator value for all selected objects
  --accumulator TEXT          provide a different initial accumulation
                              value for the reduce option  [default: 0]
  -h, --help                  Show this message and exit.
```

3.2 To use s3workers in a project

```python
import boto
import s3workers

manager = s3workers.Manager(3)

bucket = boto.connect_s3().get_bucket('mybucket')
progress = s3workers.S3KeyProgress()

def key_dumper(key):
    progress.write('%s %10d %s %s', key.last_modified, key.size, key.md5, key.name)

job = s3workers.S3ListJob(bucket, 'myprefix', None, key_dumper, progress.report)
manager.add_work(job)

manager.start_workers()
manager.wait_for_workers()
```

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/s3workers/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

s3workers could always use more documentation, whether as part of the official s3workers 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/s3workers/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 *s3workers* for local development.

1. Fork the *s3workers* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/s3workers.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 s3workers
$ cd s3workers/
$ 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 s3workers 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/s3workers/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_s3workers
```

Indices and tables

- genindex
- modindex
- search