
CloudWatch Alarms Manager Documentation

Release 0.1.0

Quentin Rousseau

May 17, 2018

Contents

1	CloudWatch Alarms Manager	3
1.1	Installation	3
1.2	Requirements	3
1.3	CWAM	3
1.4	CLI	4
2	Installation	7
2.1	Stable release	7
2.2	From sources	7
3	Usage	9
4	Contributing	11
4.1	Types of Contributions	11
4.2	Get Started!	12
4.3	Pull Request Guidelines	13
4.4	Tips	13
5	Indices and tables	15

Contents:

CloudWatch Alarms Manager

Easy way to create default CloudWatch Alarms.

CWAM is creating default alarms associated with default metrics for different kind of AWS resources.

1.1 Installation

Install using pip:

```
$ pip install cwam
```

1.2 Requirements

- Python 2.6, 2.7, 3.3, 3.4, or 3.5
- An AWS account

1.3 CWAM

1.3.1 Resources supported

- ELB
- ALB
- RDS
- Kinesis

- ElasticCache

1.3.2 Templates samples

- <https://github.com/instacart/cwam/blob/master/templates/alb.template.yml>
- <https://github.com/instacart/cwam/blob/master/templates/rds.template.yml>
- <https://github.com/instacart/cwam/blob/master/templates/kinesis.template.yml>
- https://github.com/instacart/cwam/blob/master/templates/elastic_cache.template.yml

1.3.3 Human interaction

At any time, a human can modify an alarm value created by **CWAM**. To make sure **CWAM** is not overriding that value again, the alarm description field needs to be updated with a string different from `Created by Script`.

1.4 CLI

1.4.1 CLI Authentication

Via environment variables:

```
$ export AWS_ACCESS_KEY_ID="aws_access_key_id"
$ export AWS_SECRET_ACCESS_KEY="aws_access_secret_key"
$ export AWS_SESSION_TOKEN="aws_session_token"
$ export AWS_DEFAULT_REGION="us-east-1"
$ cwam --conf ~/.cwam/conf.yml elb create -t /path/to/template.yml
```

Via (`--conf/-c`) option:

Edit `~/.cwam/conf.yml`

```
aws:
  aws_access_key_id: aws_access_key_id
  aws_access_secret_key: aws_access_secret_key
  aws_session_token: aws_session_token
  aws_default_region: aws_default_region
```

```
$ cwam --conf ~/.cwam/conf.yml elb create -t /path/to/template.yml
```

Via CLI options:

```
$ cwam ---aws-access-key-id aws_access_key_id \
--aws-access-secret-key aws_access_secret_key \
--aws-session-token aws_session_token \
--aws-default-region us-east-1 elb create -t /path/to/template.yml
```

1.4.2 Subcommands

Documentation

- <https://cwam.readthedocs.io>

History

View the [changelog](#)

Authors

- [Quentin Rousseau](#)

License

2.1 Stable release

To install CloudWatch Alarms Manager, run this command in your terminal:

```
$ pip install cwam
```

This is the preferred method to install CloudWatch Alarms Manager, 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 CloudWatch Alarms Manager can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/instacart/cwam
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/instacart/cwam/tarball/master
```

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

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


CHAPTER 3

Usage

To use CloudWatch Alarms Manager in a project:

```
import cwam
```


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/instacart/cwam/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

CloudWatch Alarms Manager could always use more documentation, whether as part of the official CloudWatch Alarms Manager 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/instacart/cwam/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 *cwam* for local development.

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

```
$ git clone git@github.com:your_name_here/cwam.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 cwam
$ cd cwam/
$ 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 cwam 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/kwent/cwam/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_cwam
```


CHAPTER 5

Indices and tables

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