collectd-rabbitmq Documentation

Release 1.2.2

Mike Buzzetti

March 25, 2016

Contents

Contents:

CHAPTER 1

collectd-rabbitmq

"A collected plugin, written in python, to collect statistics from RabbitMQ."

- Free software: Apache license
- Documentation: https://collectd-rabbitmq.readthedocs.org.
- For the older single file version see https://github.com/NYTimes/collectd-rabbitmq/tree/0.1.1

1.1 Features

- Support queue, exchange, and node stats,

1.2 Configuration

This plugin supports a small amount of configuration options:

- *Username*: The rabbitmq user. Defaults to *guest*
- *Password*: The rabbitmq user password. Defaults to *guest*
- *Realm*: The http realm for authentication. Defaults to *RabbitMQ Management*
- *Scheme*: The protocol that the rabbitmq management API is running on. Defaults to *http*
- *Host*: The hostname that the rabbitmq server running on. Defaults to *localhost*
- *Port*: The port that the rabbitmq server is listening on. Defaults to *15672*
- *Ignore*: The queue to ignore, matching by Regex. See example.

See this example for further details.

1.3 Nodes

For each node the following statistics are gathered:

- disk_free_limit
- fd_total
- fd_used

- mem_limit
- mem_used
- proc_total
- proc_used
- processors
- run_queue
- sockets_total
- sockets_used

1.4 Queues

For each queue in each vhost the following statistics are gathered: _NOTE_: The / vhost name is sent as *default*

- **message_stats**
  - deliver_get
  - **deliver_get_details**
    - rate
  - get
  - **get_details**
    - rate
  - publish
  - **publish_details**
    - rate
  - redeliver
  - **redeliver_details**
    - rate
- messages
- **messages_details**
  - rate
- messages_ready
- **messages_ready_details**
  - rate
- messages_unacknowledged
- messages_unacknowledged_details * rate
- memory
- consumers

1.5 Exchanges

For each exchange in each vhost the following statistics are gathered: _NOTE_: The / vhost name is sent as *default*

- disk_free
- disk_free_limit
- fd_total
- fd_used
- mem_limit
- mem_used
- proc_total
- proc_used
- processors
- run_queue
- sockets_total
- sockets_used

1.6 Credits

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

CHAPTER 2

Installation

At the command line:

```
$ easy_install collectd-rabbitmq
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv collectd-rabbitmq
$ pip install collectd-rabbitmq
```

This plugins requires that the Collectd type database be updated. Each of these statistics have their own custom enter in collectd's type database. To add these types defined in `this example` run the following command:

```
$ cat config/types.db.custom >> /usr/share/collectd/types.db
```

Configuration

Example Configuration

See `this example config`.

```
TypesDB "/usr/local/share/types.db.custom"

LoadPlugin python
<Plugin python>
  LogTraces true
  Interactive false
  Import "collectd_rabbitmq.collectd_plugin"
  <Module "collectd_rabbitmq.collectd_plugin">

    Username "guest"
    Password "guest"
    Realm "RabbitMQ Management"
    Host "localhost"
    Port "15672"
  </Module>
</Plugin>
```

CHAPTER 4

Usage

To use collectd-rabbitmq in a project:

```
import collectd-rabbitmq
```

CHAPTER 5

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:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at https://github.com/NYTimes/collectd-rabbitmq/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.

5.1.2 Fix Bugs

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

5.1.3 Implement Features

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

5.1.4 Write Documentation

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

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/NYTimes/collectd-rabbitmq/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 :)

5.2 Get Started!

Ready to contribute? Here's how to set up *collectd-rabbitmq* for local development.

1. Fork the *collectd-rabbitmq* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/collectd-rabbitmq.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 collectd-rabbitmq
$ cd collectd-rabbitmq/
$ 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 collectd-rabbitmq tests
$ python setup.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.

5.3 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.md.
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/jimbydamonk/collectd-rabbitmq/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

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

CHAPTER 6

Credits

6.1 Development Lead

- Mike Buzzetti <mike.buzzetti@gmail.com>

6.2 Contributors

None yet. Why not be the first?

CHAPTER 7

History

CHAPTER 8

0.1.0 (2014-09-18)

- First public release.

CHAPTER 9

Indices and tables

- genindex
- modindex
- search