syslog-ng Apache Kafka destination

Release 0.1.11

Julien Anguenot

Aug 23, 2017

Contents

1 syslog-ng-mod-python Apache Kafka destination **3**

2 librdkafka installation **5**
- 2.1 DEB packages via apt . . . 5
- 2.2 From source . . . 6

3 syslog-ng installation **7**

4 syslog-ng Kafka destination **9**
- 4.1 Stable release . . . 9
- 4.2 From sources . . . 9
- 4.3 Configure . . . 9

5 syslogng_kafka **13**
- 5.1 syslogng_kafka package . . . 13

6 Contributing **15**
- 6.1 Types of Contributions . . . 15
- 6.2 Get Started! . . . 16
- 6.3 Pull Request Guidelines . . . 17
- 6.4 Tips . . . 17

7 Releasing **19**

8 History **21**
- 8.1 0.1.11 (2017-08-23) . . . 21
- 8.2 0.1.10 (2017-08-09) . . . 21
- 8.3 0.1.9 (2017-07-28) . . . 21
- 8.4 0.1.8 (2017-07-28) . . . 21
- 8.5 0.1.7 (2017-07-27) . . . 21
- 8.6 0.1.6 (2017-07-04) . . . 22
- 8.7 0.1.5 (2017-07-03) . . . 22
- 8.8 0.1.4 (2017-06-30) . . . 22
- 8.9 0.1.3 (2017-06-30) . . . 22
- 8.10 0.1.2 (2017-06-29) . . . 22
- 8.11 0.1.1 (2017-06-29) . . . 22
- 8.12 0.1.0 (2017-06-28) . . . 22

9 Credits **23**
9.1 Development Lead 23
9.2 Contributors 23

10 Indices and tables **25**

Python Module Index **27**

Contents:

CHAPTER 1

syslog-ng-mod-python Apache Kafka destination

syslogng_kafka provides a Python module for syslog-ng 3.7 allowing one to filter and forward syslog messages to Apache Kafka brokers.

The implementation leverages confluent-kafka which uses the awesome librdkafka library providing reliability and high performance.

Please read the doc as in most cases a 'pip install' won't work as they are particular requirements that are currently not met by mainstream Linux distributions.

CHAPTER 2

librdkafka installation

syslogng_kafka depends on the confluent-kafka lib:

confluent-kafka requires librdkafka >= 0.11.0 which is currently not installed with mainstream Linux distribution.

librdkafka is a C library implementation of the Apache Kafka protocol, containing both Producer and Consumer support. It was designed with message delivery reliability and high performance in mind, current figures exceed 1 million msgs/second for the producer and 3 million msgs/second for the consumer.

DEB packages via apt

Confluent maintains apt repositories that provide packages for Debian-based Linux distributions such as Debian and Ubuntu and the installation is documented here

First install Confluent's public key, which is used to sign the packages in the apt repository:

Below are extracts from that page if you are in a hurry:

```
$ wget -qO - http://packages.confluent.io/deb/3.3/archive.key | sudo apt-key add -
```

Add the repository to your /etc/apt/sources.list:

```
$ sudo add-apt-repository "deb [arch=amd64] http://packages.confluent.io/deb/3.3␣
↪stable main"
```

Pin down the version of librdkafka to 0.11.x

```
$ sudo vim /etc/apt/preferences.d/confluent-librdkafka
```

Then add the content below:

```
Package: librdkafka1
Pin: origin "packages.confluent.io"
Pin: version 0.11.*
Pin-Priority: 550
```

```
Package: librdkafka-dev
Pin: origin "packages.confluent.io"
Pin: version 0.11.*
Pin-Priority: 550

Package: librdkafka++1
Pin: origin "packages.confluent.io"
Pin: version 0.11.*
Pin-Priority: 550
```

Then update and install:

```
$ sudo apt-get update
$ sudo apt-get install librdkafka1 librdkafka-dev
```

Note, we need to install the *-dev* package so that pip can compile confluent-kafka

From source

Alternatively, you can install librdkafka from source using the script included in this repository:

```
$ sudo bash tools/bootstrap-librdkafka.sh ${LIBRDKAFKA_VERSION} /usr/local
$ sudo ldconfig -vvv
```

This actual script is used by the Travis CI integration tests

CHAPTER 3

syslog-ng installation

You will need syslog-ng >= 3.7.x

If your favorite Linux distribution does not provide a recent enough version read this

In a nutshell, below is an example for Ubuntu / Debian

Add the repo keys:

```
$ wget -qO -  http://download.opensuse.org/repositories/home:/laszlo_budai:/syslog-ng/
↪xUbuntu_16.04/Release.key | sudo apt-key add -
```

Add the repo sources:

```
$ vim /etc/apt/sources.list.d/syslog-ng-obs.list
```

```
deb  http://download.opensuse.org/repositories/home:/laszlo_budai:/syslog-ng/xUbuntu_
↪16.04 ./
```

Pin down version:

```
$ vim /etc/apt/preferences.d/syslog-ng
```

```
Package: syslog-ng-core
Pin: origin "download.opensuse.org"
Pin: version 3.7.*
Pin-Priority: 550

Package: syslog-ng-mod-python
Pin: origin "download.opensuse.org"
Pin: version 3.7.*
Pin-Priority: 550
```

Finally update and install:

```
$ apt-get update
$ apt-get install syslog-ng-core syslog-ng-mod-python
```

Note, syslog-ng-mod-python has been introduced in syslog-ng 3.7.x

CHAPTER 4

syslog-ng Kafka destination

Stable release

To install syslog-ng Kafka driver, run this command in your terminal:

```
$ pip install syslogng_kafka
```

This is the preferred method to install syslog-ng Kafka driver, 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.

From sources

The sources for syslog-ng Kafka driver can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/anguenot/syslogng_kafka
```

Or download the tarball:

```
$ curl  -OL https://github.com/anguenot/syslogng_kafka/tarball/master
```

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

```
$ pip install -e .
```

Configure

First, let's make sure that your syslog-ng instance can accept messages.

Start by editing the main configuration file:

```
$ sudo vim /etc/syslog-ng/syslog-ng.conf
```

Below is an example opening TCP and UDP port 514 on all interfaces:

```
[...]
source s_src {

    system();

    internal();

    tcp(ip(0.0.0.0) port(514));

    udp(ip(0.0.0.0) port(514));

};
[...]
```

Configure the syslog-ng Apache Kafka destination:

```
$ vim /etc/syslog-ng/conf.d/kafka.conf
```

Sample driver configuration with every possible options. See below for documentation:

```
destination syslog_to_kafka {
    python(
        class("syslogng_kafka.kafkadriver.KafkaDestination")
            on-error("fallback-to-string")
            options(
                hosts("localhost:9092,localhost:9182")
                topic("syslog")
                partition("10")
                msg_key("src_ip")
                programs("firewall,nat")
                broker_version("0.8.2.1")
                verbose("True")
                display_stats("True")
                producer_config("{'client.id': 'sylog-ng-01', 'retry.backoff.ms': 100,
↪ 'message.send.max.retries': 5, 'queue.buffering.max.kbytes': 50240, 'default.topic.
↪config': {'request.required.acks': 1, 'request.timeout.ms': 5000, 'message.timeout.
↪ms': 300000}, 'queue.buffering.max.messages': 100000, 'queue.buffering.max.ms':␣
↪1000, 'statistics.interval.ms': 15000, 'socket.timeout.ms': 60000, 'retry.backoff.ms
↪':100,}")
                )
    );
};
log {
    source(s_src);
    destination(syslog_to_kafka);
};
```

The available options are:

- *hosts*: Kafka *bootstrap.servers*. One or multiple coma separated
- *topic*: Topic to produce message to
- *partition* (optional): Partition to produce to, elses uses the configured partitioner.

- *msg_key* (optional): Message key
- *programs* (optional): filter messages by syslog program. One or multiple coma separeted
- *broker_version* (optional): default is '0.9.0.1'
- *verbose (optional): if wether or not to print messages in logs. False by default
- *display_stats (optional): if wether or not to print broker statistics in logs. False by default
- *producer_config* (optional): The supported configuration values are dictated by the underlying librdkafka C library. For the full range of configuration properties please consult librdkafka's documentation: https://github.com/edenhill/librdkafka/blob/master/CONFIGURATION.md

** DO NOT USE *value-pairs* as indicated in syslog-ng documentation as you will get huge memory leaks...**

Restart the syslog-ng service:

```
$ service syslog-ng restart
```

To start the service in the foreground and see errors:

```
$ syslog-ng -F
```

Ensure your syslog-ng server is ready to get messages:

```
$ netstat -tanpu | grep syslog
tcp        0      0 0.0.0.0:514             0.0.0.0:*               LISTEN      11297/
↪syslog-ng
udp        0      0 0.0.0.0:514             0.0.0.0:*                           11297/
↪syslog-ng
```

CHAPTER 5

syslogng_kafka

syslogng_kafka package

Submodules

syslogng_kafka.kafkadriver module

syslogng_kafka.log module

A library that provides a custom logger for the *KafkaDestination* object.

syslogng_kafka.util module

Util library for the kakfa driver.

`syslogng_kafka.util.`**`date_str_to_timestamp`**(*date_str*)
Convert '%b %d %H:%M:%S' date string format to UNIX timestamp in local time assuming current year.

Parameters **`date_str`** – string in '%b %d %H:%M:%S' format.

Returns a string containing the UNIX timestamp

`syslogng_kafka.util.`**`parse_firewall_msg`**(*msg*)
Parse a syslog message from the firewall program into a python dictionary.

Parameters **`msg`** – firewall msg from syslog

Returns a dictionary of firewall related key value pairs

`syslogng_kafka.util.`**`parse_nat_msg`**(*msg*)
Parse a syslog message from the nat program into a python dictionary.

Parameters **`msg`** – nat msg from syslog

Returns a dictionary of nat related key value pairs

syslogng_kafka.util.**parse_str_list**(*list_str*)
Parse a string containing comma separated values and return a list of strings.

Parameters **list_str** – a string containing a comma separated list of strings

Returns a list of string Python builtin object.

Module contents

CHAPTER 6

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:

Types of Contributions

Report Bugs

Report bugs at https://github.com/ilanddev/syslogng_kafka/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.

Fix Bugs

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

Implement Features

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

Write Documentation

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

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/ilanddev/syslogng_kafka/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 :)

Get Started!

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

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

```
$ git clone https://github.com/ilanddev/syslogng_kafka.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 syslogng_kafka
$ cd syslogng_kafka/
$ 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:

```
$ make lint
$ make test
$ make test-all
```

   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.

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 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/ilanddev/syslogng_kafka/pull_requests and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

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

CHAPTER 7

Releasing

We using bumpversion to manage the releases.

Install bumpversion using pip:

```
$ pip install bumpversion
```

Run a dry run to make sure all is looking good:

```
$ bumpversion --dry-run --verbose $CURRENT_VERSION --new-version=$NEW_VERSION
```

Perform the actual release:

```
$ bumpversion $CURRENT_VERSION --new-version=$NEW_VERSION
```

Push the changes and actual tag:

```
$ git push

$ git push --tags origin
```

Then publish the archive on PyPI:

```
$ make release
```

Note, it requires valid server login in *~/.pypirc*

CHAPTER 8

History

0.1.11 (2017-08-23)

- *display_stats* options to turn on and off broker statistics

0.1.10 (2017-08-09)

- NSX edge *nat* program pre-processing

0.1.9 (2017-07-28)

- Handle *LogMessage* vs syslog-ng *values-pair* because it badly leaks if one do...
- Make 3.7.x the supported version for now because of *LogMessage* issues.

0.1.8 (2017-07-28)

- Delivery and stats callback refactoring.

0.1.7 (2017-07-27)

- Update confluent-kafka dependency to version 0.11.0: https://github.com/confluentinc/confluent-kafka-python/releases/tag/v0.11.0

0.1.6 (2017-07-04)

- Disable *delivery.report.ony.error* on callbacks because of a bug in *confluent-kafka*: https://github.com/confluentinc/confluent-kafka-python/issues/84 Let's revisit when 0.11 is released.

0.1.5 (2017-07-03)

- provide a global *on_delivery* callback in the *Producer()* config dict better for memory consumptions vs per message callback.

0.1.4 (2017-06-30)

- make *send* more robust

0.1.3 (2017-06-30)

- catch *UnicodeEncodeError* in *send()*

0.1.2 (2017-06-29)

- catch *UnicodeDecodeError* in delivery callback as it can be thrown by *err.str()*

0.1.1 (2017-06-29)

- add util to produce syslog messages in *tools* sub-folder
- remove useless *KeyboardInterrupt*
- reduce timeout of *flush()* from 30 to 5 seconds
- more tests

0.1.0 (2017-06-28)

- First release on PyPI.

CHAPTER 9

Credits

Development Lead

- Julien Anguenot <julien@anguenot.org>

Contributors

- Brett Snyder <bsnyder@iland.com>

CHAPTER 10

Indices and tables

- genindex
- modindex
- search

Python Module Index

S

syslogng_kafka, 14
syslogng_kafka.log, 13
syslogng_kafka.util, 13

Index

D

date_str_to_timestamp() (in module syslogng_kafka.util), 13

P

parse_firewall_msg() (in module syslogng_kafka.util), 13
parse_nat_msg() (in module syslogng_kafka.util), 13
parse_str_list() (in module syslogng_kafka.util), 13

S

syslogng_kafka (module), 14
syslogng_kafka.log (module), 13
syslogng_kafka.util (module), 13