PGSync Documentation

Release 2.0.0

Tolu Aina

May 13, 2021

Contents:

PostgreSQL to Elasticsearch sync

PGSync is a middleware for syncing data from Postgres to Elasticsearch. It allows you to keep Postgres as your source of truth data source and expose structured denormalized documents in Elasticsearch.

Requirements

- Python 3.6+
- Postgres 9.6+
- Redis
- Elasticsearch 6.3.1+

Postgres setup

Enable logical decoding in your Postgres setting.

- You would also need to set up two parameters in your Postgres config postgresql.conf

  `wal_level = logical`

  `max_replication_slots = 1`

Installation

You can install PGSync from PyPI:

$ pip install pgsync

Config

Create a schema for the application named e.g **schema.json**

Example schema

Example spec

Environment variables

Setup required environment variables for the application

SCHEMA='/path/to/schema.json'

ELASTICSEARCH_HOST=localhost ELASTICSEARCH_PORT=9200

PG_HOST=localhost PG_USER=i-am-root # this must be a postgres superuser PG_PORT=5432 PG_PASSWORD=*****

REDIS_HOST=redis REDIS_PORT=6379 REDIS_DB=0 REDIS_AUTH=*****

Running

Bootstrap the database (one time only) $ bootstrap –config schema.json

Run pgsync as a daemon $ pgsync –config schema.json –daemon

CHAPTER 1

Installation

1.1 Stable release

To install pgsync, run this command in your terminal:

```
$ pip install pgsync
```

This is the preferred method to install pgsync, 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.

CHAPTER 2

Usage

To use pgsync in a project:

```
from pgsync import sync
sync.main()
```

CHAPTER 3

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:

3.1 Types of Contributions

3.1.1 Report Bugs

Report bugs at https://github.com/toluaina/pgsync/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.

3.1.2 Fix Bugs

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

3.1.3 Implement Features

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

3.1.4 Write Documentation

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

3.1.5 Submit Feedback

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

3.2 Get Started!

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

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

```
$ git clone git@github.com:toluaina/pgsync.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 pgsync
$ cd pgsync/
$ 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 pgsync 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.

3.3 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 3.6, 3.7, 3.8 and 3.9, and for PyPy. Check https://github.com/toluaina/pgsync/pulls and make sure that the tests pass for all supported Python versions.

3.4 Tips

To run a subset of tests:

```
$ py.test tests/test_base.py
```

3.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Github CI will then deploy to PyPI if tests pass.

CHAPTER 4

Credits

4.1 Development Lead

- Tolu Aina <toluaina@hotmail.com>

4.2 Contributors

Francois Deschenes chokosabe Densol92

CHAPTER 5

History

5.1 1.0.1 (2020-15-01)

- First release on PyPI.

5.2 1.0.1 (2020-01-01)

- RC1 release

5.3 1.1.0 (2020-04-13)

- Postgres multi schema support for multi-tennant applications
- Show resulting Query with verbose mode
- this release required you to re-bootstrap your database with
  - bootstrap -t
  - bootstrap

5.4 1.1.1 (2020-05-18)

- Fixed authentication with Redis
- Fixed Docker build

5.5 1.1.2 (2020-06-11)

- Sync multiple indices in the same schema
- Test for replication or superuser
- Fix PG_NOTIFY error when payload exceeds 8000 bytes limit

5.6 1.1.3 (2020-06-14)

- Bug fix when syncing multiple indices in the same schema

5.7 1.1.4 (2020-06-15)

- Only create triggers for tables present in schema

5.8 1.1.5 (2020-06-16)

- Bug fix when creating multiple triggers in same schema

5.9 1.1.6 (2020-07-31)

- Bug fix when tearing down secondary schema

5.10 1.1.7 (2020-08-16)

- Fix issue #29: SQLAlchemy err: Neither 'BooleanClauseList' object nor 'Comparator' object has an attribute '_orig'

5.11 1.1.8 (2020-08-19)

- Fix issue #30: Traceback AttributeError: id

5.12 1.1.9 (2020-08-26)

- Fix issue #33: Unable to set Redis port via environment variable.

5.13 1.1.10 (2020-08-29)

- Support Amazon RDS #16
- Optimize database reflection on startup
- Show elapsed time

5.14 1.1.11 (2020-09-07)

- Support specify Elasticsearch field data type

5.15 1.1.12 (2020-09-08)

- Add support for SSL TCP/IP connection mode

5.16 1.1.13 (2020-09-09)

- Show version details with –version argument
- Fixed airbnb examples docker build

5.17 1.1.14 (2020-10-07)

- Support Elasticsearch settings for adding mapping and analyzers

CHAPTER 6

Indices and tables

- genindex
- modindex
- search