awm

Thomas Bechtold <thomasbechtold@jpberlin.de>

Nov 06, 2020

CONTENTS:

awm is a collection of services to monitor websites.

CHAPTER

ONE

USAGE

A quick guide how to install and configure *awm*.

1.1 Installation

1.1.1 virtual environment

awm can be installed into a python virtual environment:

```
$ virtualenv venv
$ source venv/bin/activate
$ pip install git+https://github.com/toabctl/awm.git
```

The available service (*awm-crawler* and *awm-persister*) are now available in *$PATH* and can be executed.

1.1.2 RPM packages

There are also prebuilt RPM packages (currently openSUSE only) on the OpenBuildService available:

```
https://build.opensuse.org/project/show/home:tbechtold:awm
```

The RPM packages contain a system user, systemd service files and a configuration file in `/etc/awm/config.json`

1.2 Configure

awm-crawler and *awm-persister* need both a configuration file. The default path is `~/.config/awm/config.json`. Here's a example configuration.

```
{
    "kafka" : {
        "servers": "HOST:PORT",
        "topic_name": "awm-crawler",
        "ssl": {
            "enabled": true,
            "cafile": "./cacert",
            "certfile": "./certfile",
            "keyfile": "./keyfile",
```

(continues on next page)

(continued from previous page)

```
            "password": "SECRET"
        }
    },
    "persister": {
        "postgres": {
            "uri": "postgres://USERNAME:PASSWORD@HOST:PORT/DATABASE?sslmode=require"
        }
    },
    "crawler": {
        "interval": 5.0,
        "urls": {
            "https://toabctl.de": { "interval": 1.0, "regex": ".*html.*" },
            "https://aiven.io": {},
            "https://google.com": {}
        }
    }
}
```

Most of the *kafka* and *persister* options should be self-explanatory.

Note: the kafka topic configured with *topic_name* must already exist or kafka must be configured to automatically create new topics. *awm* will not create the topic.

Note: the database tables needed by *awm-persister* are automatically created but the database itself must already exist.

The *crawler* section contains the global check *interval*. It also contains a map of *urls*. Every url in that map will be periodically checked. There is also the possibility to do a regular expression check against the url response body. That's optional.

1.3 Start

With the RPM packages, *systemctl* can be used to start the services:

```
systemctl start awm-crawler
systemctl start awm-persister
```

CHAPTER

TWO

CONTRIBUTING

Please use github pull requests against:

```
https://github.com/toabctl/awm
```

Make sure the tests and linters are passing. This is done via TravisCI but can also be executed locally:

```
$ tox -epy38  # for unittests
$ tox -elint  # for linters (flake8, mypy)
$ tox -edocs  # for documentation build
```

CHAPTER

THREE

AWM-CRAWLER

3.1 CLI

```
usage: awm-crawler [-h] [-d] [-v] [-c CONFIG]

Periodically monitor website status and publish to kafka

optional arguments:
  -h, --help            show this help message and exit
  -d, --debug           set loglevel to DEBUG
  -v, --verbose         set loglevel to INFO
  -c CONFIG, --config CONFIG
                        path to the config file. Default:
                        /home/docs/.config/awm/config.json
```

3.2 Module

Periodically monitor website status and publish to kafka

awm.crawler.**main**()
main entry point for the persister service. This is used by the executable *awm-persister*

CHAPTER

FOUR

AWM-PERSISTER

4.1 CLI

```
usage: awm-persister [-h] [-d] [-v] [-c CONFIG]

Persist messages from kafka to the database

optional arguments:
  -h, --help            show this help message and exit
  -d, --debug           set loglevel to DEBUG
  -v, --verbose         set loglevel to INFO
  -c CONFIG, --config CONFIG
                        path to the config file. Default:
                        /home/docs/.config/awm/config.json
```

4.2 Module

Persist awm messages from a kafka topic in a database

awm.persister.**main**()
main entry point for the persister service. This is used by the executable *awm-persister*

CHAPTER

FIVE

CONFIG

5.1 Module

The config module is responsible to creating a config dict from an available configuration file. The configuration file needs to contain valid json.

awm.common.config.**get_config**(*config_path: pathlib.Path*) → Dict

Get a config dict from a configuration file The configuration file must be valid json

Parameters **config_path** (*Path*) – the path to the config file

Raises **AwmConfigError** – Raised when the file is not found or accessable or in an invalid format

Returns the configuration dict

Return type dict

CHAPTER

SIX

TODO

Some things that need to be done (unordered):

- more unittests
- functional tests
- config schema validation (jsonschema)
- config via env vars to override specific parameters from the config file
- systemd watchdog support in case the services run under systemd
- create kafka topic automatically or document/link avn client usage
- kafka producer/consumer schemas (karapace?)
- automatically publish on pypi when new git tags are pushed to github

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

a

INDEX