Holdup

Release 5.1.0

Ionel Cristian Mărieș

Apr 11, 2024

CONTENTS

CHAPTER

ONE

OVERVIEW

docs
tests
package

A tool to wait for services and execute command. Useful for Docker containers that depend on slow to start services (like almost everything).

- Free software: BSD 2-Clause License

1.1 Installation

Currently holdup is only published to PyPI and hub.docker.com.

To install from PyPI:

```
pip install holdup
```

It has no dependencies except the optional PostgreSQL check support, which you'd install with:

```
pip install 'holdup[pg]'
```

You can also install the in-development version with:

```
pip install https://github.com/ionelmc/python-holdup/archive/master.zip
```

1.1.1 Alternate installation (Docker image)

Example:

```
docker run --rm ionelmc/holdup tcp://foobar:1234
```

Note that this will have some limitations:

- executing the `command` is pretty pointless because holdup will run in its own container
- you'll probably need extra network configuration to be able to access services
- you won't be able to use *docker run* inside a container without exposing a docker daemon in said container

1.2 Usage

usage: holdup [-h] [-t SECONDS] [-T SECONDS] [-i SECONDS] [-n] service [service ...] [– command [arg [arg ...]]]

Wait for services to be ready and optionally exec command.

positional arguments:

service
A service to wait for. Supported protocols: "tcp://host:port/", "path:///path/to/something", "unix:///path/to/domain.sock", "eval://expr", "pg://user:password@host:port/dbname" ("postgres" and "postgresql" also allowed), "http://urn", "https://urn", "https+insecure://urn" (status 200 expected for http*). Join protocols with a comma to make holdup exit at the first passing one, eg: "tcp://host:1,host:2" or "tcp://host:1,tcp://host:2" are equivalent and mean *any that pass*.

command
An optional command to exec.

optional arguments:

-h, --help	show this help message and exit
-t SECONDS, --timeout SECONDS	Time to wait for services to be ready. Default: 60.0
-T SECONDS, --check-timeout SECONDS	Time to wait for a single check. Default: 1.0
-i SECONDS, --interval SECONDS	How often to check. Default: 0.2
-v, --verbose	Verbose mode.
--verbose-passwords	Disable PostgreSQL/HTTP password masking.
-n, --no-abort	Ignore failed services. This makes *holdup* return 0 exit code regardless of services actually responding.
--insecure	Disable SSL Certificate verification for HTTPS services.
--version	display the version of the holdup package and its location, then exit.

Example:

```
holdup tcp://foobar:1234 -- django-admin ...
```

1.3 Documentation

https://python-holdup.readthedocs.io/

1.4 Development

To run all the tests run:

```
tox
```

Note, to combine the coverage data from all the tox environments run:

Windows	`set PYTEST_ADDOPTS=--cov-append` `tox`
Other	`PYTEST_ADDOPTS=--cov-append tox`

CHAPTER

TWO

REFERENCE

2.1 holdup.checks

class holdup.checks.**AnyCheck**(*checks*)

__init__(*checks*)

display(*, *verbose*, **kwargs*)

run(*options*)

class holdup.checks.**Check**

display(*, *verbose*, **kwargs*)

display_definition(**kwargs*)

error = None

is_passing(*options*)

run(*options*)

property status

class holdup.checks.**EvalCheck**(*expr*)

__init__(*expr*)

display_definition(**_*)

run(*_*)

class holdup.checks.**HttpCheck**(*url*)

__init__(*url*)

display_definition(*, *verbose_passwords*)

run(*options*)

class holdup.checks.**PathCheck**(*path*)

__init__(*path*)

display_definition(**_*)

run(_)

class holdup.checks.**PgCheck**(*connection_string*)

__init__(*connection_string*)

display_definition(*, *verbose_passwords*, *_password_re=re.compile(':[^@:]+@')*)

run(*options*)

class holdup.checks.**TcpCheck**(*host*, *port*)

__init__(*host*, *port*)

display(*, *verbose*, **_)

run(*options*)

class holdup.checks.**UnixCheck**(*path*)

__init__(*path*)

display_definition(**_)

run(*options*)

CHAPTER

THREE

CONTRIBUTING

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

3.1 Bug reports

When 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.

3.2 Documentation improvements

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

3.3 Feature requests and feedback

The best way to send feedback is to file an issue at https://github.com/ionelmc/python-holdup/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 code contributions are welcome :)

3.4 Development

To set up *python-holdup* for local development:

1. Fork python-holdup (look for the "Fork" button).
2. Clone your fork locally:

```
git clone git@github.com:YOURGITHUBNAME/python-holdup.git
```

3. Create a branch for local development:

```
git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

4. When you're done making changes run all the checks and docs builder with one command:

```
tox
```

5. 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
```

6. Submit a pull request through the GitHub website.

3.4.1 Pull Request Guidelines

If you need some code review or feedback while you're developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`).
2. Update documentation when there's new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

3.4.2 Tips

To run a subset of tests:

```
tox -e envname -- pytest -k test_myfeature
```

To run all the test environments in *parallel*:

```
tox -p auto
```

CHAPTER

FOUR

AUTHORS

- Ionel Cristian Mărieș - https://blog.ionelmc.ro
- Mithun Ayachit - https://github.com/mithun
- Dan Ailenei - https://github.com/Dan-Ailenei

CHAPTER

FIVE

CHANGELOG

5.1 5.1.0 (2024-04-12)

- Fixed buggy handling when http checks are specified with a port.
- Changed User-Agent header and stripped port from Host header for http checks.
- Refactored a bunch of code into a separate `holdup.checks` module.

5.2 5.0.0 (2024-04-11)

- Added a static binary in the Github release (built with Pyinstaller on Alpine, as a static bin).
- Dropped support for Python 3.7 and added in Python 3.12 in the test suite.

5.3 4.0.0 (2023-02-14)

- Added support for psycopg 3 (now the `holdup[pg]` extra will require that). The old psycopg2 is still supported for now.
- Dropped support for Python 3.6 and added in Python 3.11 in the test suite.

5.4 3.0.0 (2022-03-20)

- Dropped support for Python 2.
- Switched CI from Travis to GitHub Actions.
- Fixed bugs with password masking (it wasn't working for postgresql URIs).

5.5 2.0.0 (2021-04-08)

- Added support for password masking (`--verbose-passwords` to disable this feature).
- Overhauled checks display a bit, output might be slightly different.
- Added support for basic and digest HTTP authentication.
- Published Docker image at https://hub.docker.com/r/ionelmc/holdup (Alpine based).

5.6 1.9.0 (2021-01-11)

- Added a `--version` argument.
- Changed verbose output to mask passwords in postgresql checks.

5.7 1.8.1 (2020-12-16)

- Add support for PostgreSQL 12+ clients (strict integer type-checking on `connect_timeout`). The float is now converted to an integer.

5.8 1.8.0 (2019-05-28)

- Added a PostgreSQL check. It handles the `the database system is starting up` problem. Contributed by Dan Ailenei in PR #6.
- Changed output so it's more clear and more brief:
  - arguments (checks) are quoted when printed,
  - "any" checks give exact info about what made it pass,
  - repetitive information is removed.
- Simplified the internals for the "AnyCheck".

5.9 1.7.0 (2018-11-24)

- Added support for skipping SSL certificate verification for HTTPS services (the `--insecure` option and `https+insecure` protocol). Contributed by Mithun Ayachit in PR #2.

5.10 1.6.0 (2018-03-22)

- Added verbose mode (*-v* or `--verbose`).
- Changed default timeout to 60s (from 5s).

5.11 1.5.0 (2017-06-07)

- Added an `eval://expression` protocol for weird user-defined checks.

5.12 1.4.0 (2017-03-27)

- Added support for HTTP(S) check.

5.13 1.3.0 (2017-02-21)

- Add support for "any" service check (service syntax with comma).

5.14 1.2.1 (2016-06-17)

- Handle situation where internal operations would take more than planned.

5.15 1.2.0 (2016-05-25)

- Added a file check.

5.16 1.1.0 (2016-05-06)

- Removed debug print.
- Added `--interval` option for how often to check. No more spinloops.

5.17 1.0.0 (2016-04-22)

- Improved tests.
- Always log to stderr.

5.18 0.1.0 (2016-04-21)

- First release on PyPI.

CHAPTER

SIX

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

h

INDEX

Symbols

A

C

D

E

H

I

M

P

R

S

T

U