wait-for-it Documentation

Release 2.2.2

Travis Clarke

Sep 11, 2023

Contents

Version 2.2.2

Wait for service(s) to be available before executing a command.

`wait-for-it` is a script that will wait on the availability of one or more TCP services (i.e. `host:port`) before executing a user-defined command. It is useful for synchronizing the spin-up of interdependent services, such as linked docker containers.

> Since v2.0.0, `wait-for-it` will return the exit code of the executed command(s).

CHAPTER 1

Installation

```
$ pip install wait-for-it
```

CHAPTER 2

Demo

usage demo

CHAPTER 3

Usage

```
Usage: wait-for-it [OPTIONS] [COMMANDS]...

  Wait for service(s) to be available before executing a command.

Options:
  -h, --help               Show this message and exit.
  -v, --version            Show the version and exit.
  -q, --quiet              Do not output any status messages
  -p, --parallel           Test services in parallel rather than in serial
  -t, --timeout seconds    Timeout in seconds, 0 for no timeout  [default: 15]
  -s, --service host:port  Services to test, in one of the formats: ':port',
                           'hostname:port', 'v4addr:port', '[v6addr]:port' or
                           'https://...'
```

CHAPTER 4

Examples

Test to see if we can access port 80 on www.google.com, and if it is available, echo the message **google is up**:

```
$ wait-for-it \
--service www.google.com:80 \
-- echo "google is up"
```

```
[*] Waiting 15 seconds for www.google.com:80
[+] www.google.com:80 is available after 0 seconds
google is up
```

You can set your own timeout with the `-t` or `--timeout` option. Setting the timeout value to **0** will disable the timeout:

```
$ wait-for-it \
--service www.google.com:80 \
--timeout 0 \
-- echo "google is up"
```

```
waiting for www.google.com:80 without a timeout
www.google.com:80 is available after 0 seconds
google is up
```

Multiple services can be tested by adding additional `-s` or `--service` options:

```
$ wait-for-it \
--service www.google.com:80 \
--service www.bing.com:80 \
--service www.duckduckgo.com:80 \
-- echo "google, bing, and duckduckgo are up"
```

```
[*] Waiting 15 seconds for www.google.com:80
[+] www.google.com:80 is available after 0 seconds
[*] Waiting 15 seconds for www.bing.com:80
```

(continues on next page)

(continued from previous page)

```
[+] www.bing.com:80 is available after 0 seconds
[*] Waiting 15 seconds for www.duckduckgo.com:80
[+] www.duckduckgo.com:80 is available after 0 seconds
google, bing, and duckduckgo are up
```

By adding the `-p` or `--parallel` option, `wait-for-it` can do the same in parallel rather than serial:

```
$ wait-for-it \
--parallel \
--service www.google.com:80 \
--service www.bing.com:80 \
--service www.duckduckgo.com:80 \
-- echo "google, bing, and duckduckgo are up"
```

```
[*] Waiting 15 seconds for www.bing.com:80
[*] Waiting 15 seconds for www.duckduckgo.com:80
[*] Waiting 15 seconds for www.google.com:80
[+] www.bing.com:80 is available after 0 seconds
[+] www.duckduckgo.com:80 is available after 0 seconds
[+] www.google.com:80 is available after 0 seconds
google, bing, and duckduckgo are up
```

Status message output can be suppressed with the `-q` or `--quiet` option:

```
$ wait-for-it \
--quiet \
--service www.google.com:80 \
-- echo "google is up"
```

```
google is up
```

CHAPTER 5

Related

- vishnubob/wait-for-it

CHAPTER 6

License