IPpy Documentation

Release 0.3.7

Shivam Mathur

Oct 02, 2019

Contents

Contents:

IPpy

Parallel testing of IP addresses and domains in python. Reads IP addresses and domains from a CSV file and gives two lists of accessible and inaccessible ones.

1.1 About

- Compatible with both Python 2 and 3.
- Testing of IPs and domains is done in parallel.
- By default there are 4 Workers.
- All Workers work on an input Queue and a output Queue.

1.2 Modes

- verbose - if true, ping output will be displayed.
- output - `json` or `csv`

1.3 Support

- Windows, Linux and macOS are supported.
- Supports both IPv4 and IPv6 IPs, and domain names.

```
# Examples
127.0.0.1
::1
localhost
```

1.4 Install

```
$ pip install ippy
```

1.5 Usage

```python
# Create IPpy instance
ippy_obj = ippy.Ippy()

# Set config - verbose, output, num_workers
# verbose - True or False
# output - csv or json
ippy_obj.set_config(True, 'csv', 4)

# Set Input File
ippy_obj.set_file(file='ip_list.csv')

# Run IPpy
ippy_obj.run()

# Get Results
output = ippy_obj.result()
print(output)
```

1.6 Tests

To run the tests, first install tox

```
$ pip install tox
```

then run tox from the project root directory.

```
$ tox
```

Installation

2.1 Stable release

To install IPpy, run this command in your terminal:

```
$ pip install ippy
```

This is the preferred method to install IPpy, 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.

2.2 From sources

The sources for IPpy can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/shivammathur/ippy
```

Or download the tarball:

```
$ curl  -OL https://github.com/shivammathur/ippy/tarball/master
```

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

```
$ python setup.py install
```

CHAPTER 3

Usage

To use IPpy in a project:

```python
import ippy
```

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:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/shivammathur/ippy/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.

4.1.2 Fix Bugs

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

4.1.3 Implement Features

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

4.1.4 Write Documentation

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

4.1.5 Submit Feedback

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

4.2 Get Started!

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

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

   ```
   $ git clone git@github.com:your_name_here/ippy.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 ippy
   $ cd ippy/
   $ 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 the tests, including testing other Python versions with tox:

   ```
   $ python setup.py test or py.test
   $ tox
   ```

   To get 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.

4.3 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 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/shivammathur/ippy/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_ippy
```

Indices and tables

- genindex
- modindex
- search