

---

# **Underwear Documentation**

***Release 0.1.0***

**Matt Makai**

July 26, 2015



<b>1</b>	<b>Underwear</b>	<b>3</b>
1.1	What Problem Does Underwear Solve? . . . . .	3
1.2	Quick Start . . . . .	3
1.3	License and Documentation . . . . .	4
1.4	About the Name “Underwear” . . . . .	4
1.5	Features . . . . .	4
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
3.1	SSH keys . . . . .	7
<b>4</b>	<b>Contributing</b>	<b>9</b>
4.1	Types of Contributions . . . . .	9
4.2	Get Started! . . . . .	10
4.3	Pull Request Guidelines . . . . .	10
4.4	Tips . . . . .	11
<b>5</b>	<b>Credits</b>	<b>13</b>
5.1	Development Lead . . . . .	13
5.2	Contributors . . . . .	13
<b>6</b>	<b>Roadmap</b>	<b>15</b>
6.1	0.5.0 . . . . .	15
6.2	0.4.4 . . . . .	15
6.3	0.4.3 . . . . .	15
<b>7</b>	<b>History</b>	<b>17</b>
7.1	0.4.3 (2014-01-02) . . . . .	17
7.2	0.4.2 (2013-12-30) . . . . .	17
7.3	0.4.0 (2013-12-26) . . . . .	17
7.4	0.3.0 (2013-12-24) . . . . .	17
7.5	0.2.0 (2013-12-16) . . . . .	17
7.6	0.1.0 (2013-12-16) . . . . .	17
<b>8</b>	<b>Indices and tables</b>	<b>19</b>



Contents:



---

## Underwear

---

Underwear is a library for easily deploying any Python-powered web application to one or more Linux servers.

### 1.1 What Problem Does Underwear Solve?

Despite the advent of configuration management tools such as [Puppet](#), [Chef](#), [Ansible](#), and [Salt](#), it remains difficult to deploy a web application because you have to first learn one of those tools and then write scripts in the tool's domain-specific language.

Underwear makes deploying to a traditional Linux server stack as easy as deploying to Heroku by providing a pre-packaged, easily configurable library. Deployments can be executed simply by installing Underwear with [pip](#), specifying the IP addresses of the server(s) to deploy to, then running a couple of commands.

### 1.2 Quick Start

After these ten steps your app should be completely configured, secured, and deployed to remote servers.

*These instructions will be simplified in future releases as the library is further developed.*

1. Install Underwear to your [virtualenv](#):

```
pip install underwear
```

2. Add 'underwear' to `INSTALLED_APPS` in your Django project. *Underwear currently only supports Django (Flask and Bottle will be added in the 0.5 release).*
3. Create a deploy directory under your Django project:

```
mkdir deploy; cd deploy
```

4. Download the Fabric file for uploading SSH keys, YAML configuration template, and hosts file to your project under the `deploy/` directory:

```
wget https://raw.githubusercontent.com/makaimc/underwear/master/underwear/deploy/fabfile.py
wget https://raw.githubusercontent.com/makaimc/underwear/master/underwear/deploy/underwear.yml
wget https://raw.githubusercontent.com/makaimc/underwear/master/underwear/deploy/hosts
```

5. Fill in blank variables fields in **fabfile.py**.
6. Execute the Fabric script (the script will prompt you for the non-root password):

```
fab bootstrap_ansible
```

7. Fill in blank fields and modify desired settings in the **underwear.yml** file so that Underwear knows where to install your web application.
8. Fill in the **hosts** file with your servers' IP addresses. For example, the file may look like the following:

```
[webservers]
192.168.1.1

[dbservers]
192.168.1.2
```

8. Deploy your application:

```
python manage.py deploy
```

9. Underwear will output each step in the deployment process to standard output as it happens.
10. Access your application from the domain name you provided in the underwear.yml file.

## 1.3 License and Documentation

Underwear is free software under the MIT license.

Additional Underwear documentation can be found on [Read the Docs](#).

## 1.4 About the Name “Underwear”

This library goes underneath your WSGI application to help you quickly deploy it to traditional servers. Otherwise no one knows you're using it, and you can swap it out at any time for another deployment process when you outgrow the standard LAMP-stack setup.

There also weren't any projects on GitHub by the name of underwear that had more than two stars and “underwear” was an open library name on PyPi :)

## 1.5 Features

- Automated deployments to a web & database server
- Configurable via a simple YAML template



---

# Installation

---

At the command line:

```
$ easy_install underwear
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv underwear  
$ pip install underwear
```



---

## Usage

---

To use Underwear in a project:

```
import underwear
```

### 3.1 SSH keys

Ansible runs over SSH, so we need a way to bootstrap SSH connections through a non-root user.

One way to automate these first few steps is with Fabric. The `fabfile.py.template` contains one public function, `bootstrap_ansible`. `bootstrap_ansible` calls the other functions to create a non-root user with sudo privileges, upload public keys for deployment, and lock down root from logging in.

Copy `fabfile.py.template` to `fabfile.py`, fill in the commented fields at the top of the script, then run the script with:

```
fab bootstrap_ansible
```

Right now the script will prompt you for the password the non-root user should be created with. I'll automate that manual step away later.



---

## 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/makaimc/underwear/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” is open to whoever wants to implement it.

#### 4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

#### 4.1.4 Write Documentation

Underwear could always use more documentation, whether as part of the official Underwear 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/makaimc/underwear/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 *underwear* for local development.

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

```
$ git clone git@github.com:your_name_here/underwear.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 underwear
$ cd underwear/
$ 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 underwear tests
$ python setup.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.

## 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, and 3.3, and for PyPy. Check [https://travis-ci.org/makaimc/underwear/pull\\_requests](https://travis-ci.org/makaimc/underwear/pull_requests) and make sure that the tests pass for all supported Python versions.

## 4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_underwear
```





---

**Credits**

---

## 5.1 Development Lead

- Matt Makai <[matthew.makai@gmail.com](mailto:matthew.makai@gmail.com)>

## 5.2 Contributors

None yet. Why not be the first?



---

# Roadmap

---

This document proposes how Underwear will evolve over the course of 2014.

### 6.1 0.5.0

- Add support for deploying Flask apps.

### 6.2 0.4.4

- Clean up ugly Django management command integration.
- Create unit tests for core deployment flow.

### 6.3 0.4.3

- Add new flags to configure how a Django app should be deployed.



---

## History

---

### 7.1 0.4.3 (2014-01-02)

- Adding toggle for whether South should be used with a Django project (`django_use_south`) and a toggle for whether a fixtures file should be loaded (`django_fixtures_file`).

### 7.2 0.4.2 (2013-12-30)

- Added toggle to flip whether or not SSL should be used with Nginx (`deploy_ssl`).

### 7.3 0.4.0 (2013-12-26)

- Release now allows for 2 server web app deployments (one web server and one database server).
- Fixes issues with Supervisor template and Django environment variables.
- Includes templates required to tell the library where to deploy the application.

### 7.4 0.3.0 (2013-12-24)

- First release on PyPI.

### 7.5 0.2.0 (2013-12-16)

- Added initial working Ansible scripts for deployment.

### 7.6 0.1.0 (2013-12-16)

- Initial codebase with Python package



---

## Indices and tables

---

- *genindex*
- *modindex*
- *search*