Terrarium Documentation

Release dev

Kyle Gibson, Wes Winham

Mar 15, 2017

Contents

CHAPTER 1

Installation

Requirements

- CPython 2.7 or 2.6
- virtualenv (only the latest 2 versions are officially supported and tested) 15.1.x, 15.0.x

Note: As Python 2.6 is no longer being maintained, terrarium's support for 2.6 is officially deprecated and will be removed in a future version.

Install using pip

```
$ pip install terrarium
```

Upgrade using pip

```
$ pip install -U terrarium
```

CHAPTER 2

Quick Start

```
terrarium [options] COMMAND [requirements files...]
```

See `terrarium --help` for a complete list of options and commands.

Creating a new environment

The following example will create a new virtual environment named `env` that includes the packages defined in `requirements.txt`

```
$ terrarium --target env install requirements.txt
```

Replacing an existing environment

The following example demonstrates how terrarium can be used to replace an existing activated virtual environment with a different set of packages.

```
$ terrarium --target env install requirements.txt
$ source env/bin/activate
$ terrarium install other_requirements.txt
```

Note: The environment that was replaced is renamed to `env.bak`, and can be restored using `terrarium revert`.

Note: After installing the `other_requirements`, it is not necessary to run `deactivate` or `activate` to begin using the new environment.

CHAPTER 3

User Guide

```
terrarium [options] COMMAND [requirements files...]
```

See `terrarium --help` for a complete list of options and commands.

Creating a new environment

The following example will create a new virtual environment named `env` that includes the packages defined in `requirements.txt`

```
$ terrarium --target env install requirements.txt
```

Replacing an existing environment

The following example demonstrates how terrarium can be used to replace an existing activated virtual environment with a different set of packages.

```
$ terrarium --target env install requirements.txt
$ source env/bin/activate
$ terrarium install other_requirements.txt
```

Note: The environment that was replaced is renamed to `env.bak`, and can be restored using `terrarium revert`.

Note: After installing the `other_requirements`, it is not necessary to run `deactivate` or `activate` to begin using the new environment.

Saving and using environment archives

Terrarium provides options for archiving and compressing a freshly installed and built environment, either locally or remotely (via Amazon S3).

When these options are used, terrarium will first check if the environment has already been saved. In that case, terrarium will download the environment archive instead of downloading and building each individual package specified in the requirements files.

Storing terrarium environments locally

Storing terrarium environments locally (or on a shared network disk) can be achieved using the `--storage-dir` option.

```
$ terrarium --target env --storage-dir path/to/environments install requirements.txt
```

After building a fresh environment from the requirements in `requirements.txt`, terrarium will archive and compress the environment. Finally, the compressed version is then copied to the path specified by `--storage-dir`.

Storing terrarium environments on Cloud Storage Services (S3, GCS)

Terrarium also supports storing and retrieving archives stored on these storage services:

- Amazon Web Service - S3
- Google Cloud Platform - Google Cloud Storage

Amazon S3

The following options are only available if `boto` is installed.

- `--s3-bucket`
- `--s3-access-key`
- `--s3-secret-key`
- `--s3-max-retries`

Google Cloud Storage

The following options are only available if `gcloud` is installed.

- `--gcs-bucket`
- `--gcs-client-email`
- `--gcs-secret-key`
- `--gcs-max-retries`

Note: Each of the above options can be specified using environment variables, e.g. `S3_BUCKET`, `GCS_BUCKET` instead of being passed in as a parameter.

Tips

Using an alternative index server

If you're using an index server other than PyPI (perhaps an index server with internal-only packages), then you need to be able to tell terrarium to use that index URL. Terrarium does not have the `-i` (`--index-url`) option that pip has, so how do you indicate the index URL? Well, you may recall that pip requirements files can also contain command-line options... So add a line like this to one of your requirements files:

```
--index-url http://internal-index-server.corp/index
```

You can add a line like the above to an existing requirements file that has a list of packages or you could add it to a separate requirements file and then add that to the terrarium command-line.

```
$ terrarium --target testenv install internal-index-server.txt requirements.txt
```

CHAPTER 4

Development

Installing requirements

Using pip

```
$ pip install -r requirements/docs.txt -r requirements/testing.txt
```

Building the documentation locally

1. Install the documentation requirements:

   ```
   $ pip install -r requirements/docs.txt
   ```

2. Change directory to `docs` and run `make html`:

   ```
   $ cd docs
   $ make html
   ```

3. Load HTML documentation in a web browser of your choice:

   ```
   $ firefox docs/_build/html/index.html
   ```

Running tests

1. Install the development requirements:

   ```
   $ pip install -r requirements/testing.txt
   ```

2. Run `nosetests` in the project root.

```
$ nosetests
```

To run all tests against all supported versions of python, use `tox`.

Running tests with tox

`tox` allows us to use one command to run tests against all versions of python that we support.

Setting up tox

1. Decide how you want to manage multiple python versions.
   (a) System level using a package manager such as `apt-get`. This approach will likely require adding additional `apt-get` sources in order to install alternative versions of python.
   (b) Use pyenv to manage and install multiple python versions. After installation, see the pyenv command reference.
2. Install `tox`.

```
$ pip install tox
```

3. Configure tox.

Running tox

Now that you have `tox` setup, you just need to run the command `tox` from the project root directory.

```
$ tox
```

Getting involved

The terrarium project welcomes help in any of the following ways:

- Making pull requests on github for code, tests and documentation.
- Participating on open issues and pull requests, reviewing changes

Pull Request Checklist

To have the best chance at an immediate merge, your pull request should have:

- A passing Travis-CI build. If it fails, check the console output for reasons why.
- New unit tests for new features or bug fixes.
- New documentation in `docs` for any new features. You do want people to know how to use your new stuff, right?

Release process

1. Update CHANGELOG.
2. Bump the version number in __init__.py on master.
3. Tag the version.
4. Push to PyPI.

CHAPTER 5

Release Notes

1.0.1

- Support latest 2 versions of virtualenv

1.0.0

- Support for Google Cloud Storage, in addition to Amazon S3
- Preserve order of requirements

1.0.0-rc.6

- shlex comments=True breaks remote source requirements

1.0.0-rc.5

- Fixed some spelling and grammar in the README
- Added “Tips” Section to README
- Permit inline comments in requirements files
- Added tox.ini for tox
- Deprecate python 2.5
- Update travis configuration to use tox
- Update documentation to use sphinx
- Update version number in only one place
- Moved README content to sphinx docs

1.0.0-rc.4

- Add support for pip 1.3.1
- Add command to restore backup
- Added –require-download option

1.0.0-rc.3

- Don't print S3 secret key in verbose mode / debug level
- Update requirements to include support for virtualenv 1.8.4 and others
- Fixed mac compatibility issues because of bsd tar
- Improvements for tests
- Pip 1.2 compatibility
- Handle comments in requirements files
- Output when boto not installed is confusing
- Output on normal install makes it feel like terrarium hung
- –storage-dir should default to environment variable TERRARIUM_STORAGE_DIR
- Gracefully handle permission failures when creating environment backup

1.0.0-rc.2

- Cannot call rmtree on a symbolic link
- Dangling symlink on extract

1.0.0-rc.1

- boto.S3Connection should be boto.s3.connection.S3Connection
- Terrarium needs to fix paths after upload
- Implement options as described in README spec