
Terrarium Documentation

Release dev

Kyle Gibson, Wes Winham

Mar 01, 2019

Contents

1	Installation	1
1.1	Requirements	1
1.2	Install using pip	1
1.3	Upgrade using pip	1
2	Quick Start	3
2.1	Creating a new environment	3
2.2	Replacing an existing environment	3
3	User Guide	5
3.1	Creating a new environment	5
3.2	Replacing an existing environment	5
3.3	Saving and using environment archives	6
3.4	Tips	7
4	Development	9
4.1	Building the documentation locally	9
4.2	Running tests	9
4.3	Getting involved	9
4.4	Release process	10
5	Release Notes	11

1.1 Requirements

- CPython 2.7
- [virtualenv](#) (only the latest 2 versions are officially supported and tested)

Note: As Python 2.7 is no longer being maintained, terrarium’s support for 2.7 is officially deprecated and will be removed in a future version.

1.2 Install using pip

```
$ pip install terrarium
```

1.3 Upgrade using pip

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



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

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

2.1 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
```

2.2 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.


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

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

3.1 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
```

3.2 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.

3.3 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.

3.3.1 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`.

3.3.2 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.

3.4 Tips

3.4.1 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
```


4.1 Building the documentation locally

1. Install `tox`

```
$ pip install tox
```

2. Use `tox`

```
$ tox -e docs
```

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

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

4.2 Running tests

1. Install `tox`

```
$ pip install tox
```

2. Use `tox`

```
$ tox
```

4.3 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

4.3.1 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?

4.4 Release process

1. Update `CHANGELOG`.
2. Bump the version number in `__init__.py` on master.
3. Tag the version.
4. Generate source and wheel distributions: `python setup.py sdist bdist_wheel`
5. Upload to PyPI: `twine upload dist/*`

1.2.0

- Support virtualenv 16 and 15
- Added wheel distribution to PyPI.
- Improved packaging (setup.cfg) and tox configuration
- Announced deprecation for Python 2.7

1.1.0

- Dropped support for virtualenv<15
- Dropped support for python 2.6
- Added support for pip>=10
- Added support for latest version of virtualenv

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