psf-infra Documentation

Release 0.0.2

PSF Infrastructure Team

Mar 23, 2017

Contents

Welcome to the Python Infrastructure Team Documentation Index

CHAPTER 1

Services

Warehouse

Warehouse is deployed continuously. Every push to master triggers a new build which will get picked up the next time the deployment procedure is run on the Warehouse servers. The process is:

1. For every push to master, except those automatically generated by the release process, trigger a new test run which will ensure nothing has broken. If the test run completes successfully, then a new build job is triggered in Jenkins.
2. Jenkins will generate and tag a new version for the next release of Warehouse, following a version scheme of `YY.MM.NN`, where `NN` is an incrementing serial number.
3. Jenkins will generate a new Python source distribution and Wheel of the latest release.
4. Jenkins will generate a new Debian package of the latest version, bundling Warehouse and all of its dependencies into a single virtual environment which, when installed, will end up in `/opt/warehouse`.
5. If generating both the Python packages and the Debian package was successful then Jenkins will publish the Python packages to PyPI, the Debian packages to an internal apt repository, and push the tagged version to GitHub.
6. Chef will periodically (every 30 minutes) check the internal apt repository for an updated package and will update to the latest version if needed.

Environment / Dependencies

- PyPy
- PostgreSQL 9.2+ (Hosted by OSUOL)
- Elasticsearch

Configuration

Warehouse is configured using a YAML file which the cookbook will write to `/opt/warehouse/etc/config.yml`.

Debian Packages and Virtual Environments

The Warehouse deployment uses Debian packaging as a means of delivery to the application servers. This allows us to easily generate a build artifact and then deploy that built artifact to the application server.

Using a modified dh-virtualenv the build process for the Debian package creates a new virtual environment, installs Warehouse and all of its dependencies into that virtual environment, and then packages the resulting environment into a single debian package.

This setup was chosen because it offers the best isolation from build time failures. It also moves as much of the process into a one time build process instead of needing to execute a pip install every 30 minutes to check for updated requirements.

CHAPTER 2

Let's get started

- genindex
- search
- getting-started
- nodes
- roles
- generating-these-docs