Release Management with Devpi

Stephan Erb | @ErbStephan | 22. July 2015
Release Management
Release Management

“zZZzzZ”

Developer
Anti-Example
(yes, that happened...)

**Original:**  pawesome-1.0.tar.gz

**Bugfix:**   pawesome-1.0.tar.gz

pawesome-1.0+golden_copy.tar.gz
Our Approach
(move towards Immutable Infrastructure)

Immutable Packages

- create once, store forever
- package versions derived from git
- packaged as wheels
- dependencies pinned in requirements.txt
Packaging Example
(powered by setuptools_scm)

$ git clone https://github.com/StephanErb/pawesome.git
$ git tag

1.0
1.1

$ python setup.py bdist_wheel
$ ls dist/

pawesome-1.2.dev2+n962232f-cp27-none-linux_x86_64.whl
Devpi Package Server
(from Holger Krekel et al.)

Elevator Pitch:
“Host your own python packages and install them like any other package from pypi.python.org”

http://doc.devpi.net
Devpi by Example

$ devpi use https://mydevpi/user/index
$ devpi login user
$ devpi upload pawesome-1.0-cp27-none-linux_x86_64.whl

$ pip install pawesome -i https://mydevpi/user/index
TechTeam/generic/: pawesome-1.2.3 metadata and description

Example demonstration of devpi and setuptools_scm

**author**  Stephan Erb
**author_email** stephan.erb@blue-yonder.com

<table>
<thead>
<tr>
<th>File</th>
<th>Tox results</th>
<th>History</th>
</tr>
</thead>
<tbody>
<tr>
<td>pawesome-1.2.3-cp27-none-linux_x86_64.whl</td>
<td></td>
<td>Uploaded to TechTeam/generic by TechTeam Today at 14:19</td>
</tr>
<tr>
<td>Size: 2 KB</td>
<td>Type: Python Wheel</td>
<td></td>
</tr>
<tr>
<td>Python: 2.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pawesome-1.2.3.tar.gz</td>
<td></td>
<td>Uploaded to TechTeam/generic by TechTeam Today at 14:19</td>
</tr>
<tr>
<td>Size: 11 KB</td>
<td>Type: Source</td>
<td></td>
</tr>
</tbody>
</table>

Example project for the EuroPython talk Release Management with Devpi

The following command sequence will upload code and documentation to the given Devpi server. Packages are uploaded as wheels and source distributions. Upload settings are defined in setup.cfg.

```
pip install -r requirements.txt

devpi use http://mydevpi/myuser/myindex
devpi login myuser

devpi upload --with-docs # builds before uploading
```
Release Workflow

[Diagram showing the workflow with various blocks and arrows connecting them, leading to the label 'Devpi']
Release Workflow

Constraints:
- multiple teams
- binary packages
- internal & external dependencies
Devpi Usage
Devpi by Example

$ devpi use https://mydevpi/user/index
$ devpi login user
$ devpi upload pawesome-1.0-cp27-none-linux_x86_64.whl

$ pip install pawesome -i https://mydevpi/user/index
Basic Index Layout

- Teams as **Devpi users**
- Operating systems as (non-volatile) **indices**

(needed until wheel issue #144 and pip issue #2875 are resolved)
Aggregation Indices

Team A

Team B

Platform

$ pip install pawesome -i https://mydevpi/platform/Debian7
Open Source Packages

- Company-wide whitelist
- Filled via devpi-builder (https://github.com/blue-yonder/devpi-builder)
Consumer Indices

Providers

Aggregation

Consumers
Some Statistics

Used on
- Jenkins, Desktops, Servers

Devpi Usage
- ~10,000 artifacts uploaded onto 350 indices
- ~17.5TB downloaded over 1 year

Devpi Hosts
- 4 cores, 4gb ram
Host Setup
(power by replication)

Hginx

Replica

Master
High Availability
(power by replication)

Reverse Proxy

Replica

Master

Host A

Host B
Executive Summary

• **Wheels** with version numbers from **git**

• Upload to **non-volatile** indices
• One **index per operating system**

• Users for **internal teams** and external software

• **Replicas** for high-availability
Some Lessons
(learned the hard way)

Provide a test server!
• you have to cleanup mistakes on the prod server
• consequence of immutable (non-volatile) indices

Use a client wrapper!
• proper command line usage
• enforce policies and prevent common mistakes
• server-client compatibility
Thank You!
If you think of
• literature when you hear Kafka
• mythology when you hear Cassandra
• animals when you hear Zookeeper
... then have a nice day.

If you think of distributed systems, then join us!

www.blue-yonder.com
References
Websites

Devpi Upstream
• https://bitbucket.org/hpk42/devpi
• https://github.com/devpi/

Devpi plumber, builder, and cleaner
• https://github.com/blue-yonder

New Feature: “Platform tags in Wheels”
• https://bitbucket.org/pypa/wheel/issues/144/
• https://github.com/pypa/pip/issues/2875