

Guix-tox

A functional version of tox

Cyril Roelandt
cyril@redhat.com

January 30, 2016



- Cyril Roelandt
- OpenStack developer at Red Hat since 2013
- GNU Guix developer



1 Tox

2 Issues with virtualenv

- Only Python packages can be installed
- Reproducibility
- One package manager per language

3 GNU Guix

- Functional package manager
- Guix environment
- Solving the issues of virtualenv

4 guix-tox

- Goal
- Demo
- Drawbacks



Tox:

- Automate/standardize testing of Python software
- Run tests with various configurations
- Widely used
- Uses virtualenv



Various user-defined configurations:

```
$ tox -l  
py34 # Run the test suite with Python 3.4  
py27 # Run the test suite with Python 2.7  
pypy # Run the test suite with PyPy  
pep8 # Find stylistic issues  
docs # Build the doc
```

Running tests:

```
$ tox -e pep8
pep8 create: /home/cyril/openstack/python-novaclient/.tox/pep8
pep8 installdeps: -r/home/cyril/openstack/python-novaclient/requirements
.txt,
-r/home/cyril/openstack/python-novaclient/test-requirements.txt
pep8 runtests: PYTHONHASHSEED='1957133822'
pep8 runtests: commands[0] flake8
./novaclient/v2/shell.py:4918:10: E211 whitespace before '('
print(_("Server supported API versions:"))
ERROR: InvocationError: '/home/cyril/openstack/python-novaclient/.tox/
pep8/bin/flake8'
ERROR: pep8: commands failed
```

Tox creates a virtualenv for each configuration:

```
$ ls .tox/  
log pep8
```

Users may ask tox to completely recreate a given environment:

```
$ tox -r --e pep8
```

Table of contents

- 1 Tox
- 2 Issues with virtualenv
 - Only Python packages can be installed
 - Reproducibility
 - One package manager per language
- 3 GNU Guix
 - Functional package manager
 - Guix environment
 - Solving the issues of virtualenv
- 4 guix-tox
 - Goal
 - Demo
 - Drawbacks



Issue:

```
(libffi-test)$ pip install cffi
x86_64-linux-gnu-gcc ...
c/_cffi_backend.c:13:17: fatal error: ffi.h: No such file or directory
#include <ffi.h>
                ^
compilation terminated.
error: command 'x86_64-linux-gnu-gcc' failed with exit status 1
```

Fix:

```
# apt-get install libffi-dev
```

Or:

```
# yum install ...
```

Or...

```
# $distro_pkg_manager install $missing_pkg
```



What version of libffi will be used?

- Debian Stable/Testing/Unstable (+ Debian patches)
- Fedora X (+ Fedora patches)
- Package from another distribution
- "Vanilla" version manually installed from a tarball

-> Different behaviours



Table of contents

- 1 Tox
- 2 Issues with virtualenv
 - Only Python packages can be installed
 - **Reproducibility**
 - One package manager per language
- 3 GNU Guix
 - Functional package manager
 - Guix environment
 - Solving the issues of virtualenv
- 4 guix-tox
 - Goal
 - Demo
 - Drawbacks



Environment variables can change the behaviour of a program:

```
$ LANG=C python3 -c "print('é')"  
Unable to decode the command from the command line:  
UnicodeEncodeError: 'utf-8' codec can't encode character '\udcc3' in  
  position 7:  
surrogates not allowed  
  
$ LANG=en_US.UTF-8 python3 -c "print('é')"  
é
```

Two developers may witness different behaviours depending on the value of "LANG".



External commands may be used even though the virtual environment has no control over them:

```
$ cat tox.ini
[tox]
envlist=py34

[testenv]
whitelist_externals=/bin/echo
commands=
    echo "Starting tests"
    python test.py
```

The "echo" command might not be the exact same on all machines. A valid tox.ini might even use more complex commands instead of "echo"...

Table of contents

- 1 Tox
- 2 Issues with virtualenv
 - Only Python packages can be installed
 - Reproducibility
 - One package manager per language
- 3 GNU Guix
 - Functional package manager
 - Guix environment
 - Solving the issues of virtualenv
- 4 guix-tox
 - Goal
 - Demo
 - Drawbacks



Current trend: one language => one package manager

- Python -> pip
- Perl -> cpan
- Javascript -> npm
- ...



Table of contents

- 1 Tox
- 2 Issues with virtualenv
 - Only Python packages can be installed
 - Reproducibility
 - One package manager per language
- 3 GNU Guix
 - Functional package manager
 - Guix environment
 - Solving the issues of virtualenv
- 4 guix-tox
 - Goal
 - Demo
 - Drawbacks



- Based on Nix
- Written in GNU Guile (Scheme)
- Per-user package management
- Rollbacks
- Functional



Functional package manager:

- As in "functional programming"
- A package is a "function"
- Parameters: version, tarball URL, inputs...
- Return value: directory in /gnu/store



Example: the "hello" package

```
(define-public hello
  (package
    (name "hello")
    (version "2.10")
    (source
      (origin
        (method url-fetch)
        (uri (string-append "mirror://gnu/hello/hello-" version
          ".tar.gz"))
        (sha256
          (base32
            "0ssi1wpaf7p1aswqqjwigppsg5fyh99vdlb9kz17c9lng89ndq1i")))))
    (build-system gnu-build-system)
    (synopsis "Hello, GNU world: An example GNU package")
    (description
      "GNU Hello prints the message \"Hello, world!\" and then exits. It
      serves as an example of standard GNU coding practices. As such, it
      supports
      command-line arguments, multiple languages, and so on.")
    (home-page "http://www.gnu.org/software/hello/")
    (license gpl3+)))
```



Example: installing "hello"

```
$ guix package -i hello
The following package will be installed:
  hello      2.10      /gnu/store/6b2sx521shms932sldmj2kv1skk246ky-
  hello -2.10
...
16 packages in profile

$ which hello
/home/cyril/.guix-profile/bin/hello

$ ls -l /home/cyril/.guix-profile/bin/hello
/home/cyril/.guix-profile/bin/hello ->
/gnu/store/6b2sx521shms932sldmj2kv1skk246ky-hello-2.10/bin/hello
```



Table of contents

- 1 Tox
- 2 Issues with virtualenv
 - Only Python packages can be installed
 - Reproducibility
 - One package manager per language
- 3 GNU Guix
 - Functional package manager
 - **Guix environment**
 - Solving the issues of virtualenv
- 4 guix-tox
 - Goal
 - Demo
 - Drawbacks



- Create a virtual environment (just like virtualenv)
- The user profile is left untouched
- Run a given command
- Interactive mode



```
$ guix environment --ad-hoc --pure which
```




```
$ guix environment --ad-hoc --pure which  
  
$ which guix  
which: no guix in (/gnu/store/scra07w6z2hnmackqj851rqznd7sznvm--which  
-2.21/bin)
```



```
$ guix environment --ad-hoc --pure which  
  
$ which guix  
which: no guix in (/gnu/store/scra07w6z2hnmackqj851rqznd7sznm-which  
-2.21/bin)  
  
$ echo $PATH  
/gnu/store/scra07w6z2hnmackqj851rqznd7sznm-which-2.21/bin
```



```
$ guix environment --ad-hoc --pure which  
  
$ which guix  
which: no guix in (/gnu/store/scra07w6z2hnmackqj851rqznd7sznvm-which  
-2.21/bin)  
  
$ echo $PATH  
/gnu/store/scra07w6z2hnmackqj851rqznd7sznvm-which-2.21/bin  
  
$ exit  
exit
```



```
$ guix environment --ad-hoc --pure python-babel python
```



```
$ guix environment --ad-hoc --pure python-babel python
$ echo $PYTHONPATH
/gnu/store/7k8z2a1bqhj47hq4r4iz54cmb5y3hl7d-python-babel-1.3/lib/python3
.4/site-packages:/gnu/store/9iannxg7mfz6misi4nf1q8avdjrlzr-python
-3.4.3/lib/python3.4/site-packages
```



```
$ guix environment --ad-hoc --pure python-babel python
$ echo $PYTHONPATH
/gnu/store/7k8z2a1bqhj47hq4r4iz54cmb5y3hl7d-python-babel-1.3/lib/python3
.4/site-packages:/gnu/store/9iannxg7mfz6misi4nf1q8avdjrlzr-python
-3.4.3/lib/python3.4/site-packages
$ exit
exit
```

Complete isolation from the rest of the system using containers:

```
$ guix environment --ad-hoc --pure coreutils  
$ ls /usr/bin/make  
/usr/bin/make
```



Complete isolation from the rest of the system using containers:

```
$ guix environment --ad-hoc --pure coreutils
$ ls /usr/bin/make
/usr/bin/make

$ guix environment --ad-hoc --pure --container coreutils
# ls /usr/bin/make
ls: cannot access /usr/bin/make: No such file or directory
```



Table of contents

- 1 Tox
- 2 Issues with virtualenv
 - Only Python packages can be installed
 - Reproducibility
 - One package manager per language
- 3 GNU Guix
 - Functional package manager
 - Guix environment
 - Solving the issues of virtualenv
- 4 guix-tox
 - Goal
 - Demo
 - Drawbacks



- Handling multiple versions of a single package (like virtualenv)
- Handling !Python packages (compilers, lib*-dev)
- Better isolation from the rest of the system (environment variables, unspecified commands...)



Table of contents

- 1 Tox
- 2 Issues with virtualenv
 - Only Python packages can be installed
 - Reproducibility
 - One package manager per language
- 3 GNU Guix
 - Functional package manager
 - Guix environment
 - Solving the issues of virtualenv
- 4 guix-tox
 - Goal
 - Demo
 - Drawbacks



- Keep using tox
- Replace virtualenv with GNU Guix
- Code : <https://git.framasoft.org/Steap/guix-tox>



Table of contents

- 1 Tox
- 2 Issues with virtualenv
 - Only Python packages can be installed
 - Reproducibility
 - One package manager per language
- 3 GNU Guix
 - Functional package manager
 - Guix environment
 - Solving the issues of virtualenv
- 4 guix-tox
 - Goal
 - **Demo**
 - Drawbacks



```
$ cd python-keystoneclient
$ GUIX_TOX_EXTRA=openssl guix-tox --env=guix -epy27
...
py27 Guix env: guix environment --ad-hoc --pure python2-pbr python2-
babel
python2-iso8601 python2-debtcollector python2-netaddr python2-oslo.
config
python2-oslo.i18n python2-oslo.serialization python2-oslo.utils
python2-prettytable python2-requests python2-six python2-stevedore
python2-hacking python2-coverage python2-discover python2-fixtures
python2-keyring python2-lxml python2-mock python2-oauthlib python2-
oslotest
python2-pycrypto python2-requests-mock python2-sphinx python2-
testrepository
python2-testresources python2-testtools python2-webob python2-setuptools
python2-pbr openssl python-2 --exec=python setup.py testr --slowest
--testr-args=
...
Ran 1131 (+937) tests in 26.290s (+22.746s)
PASSED (id=42, skips=4)
...
```

Table of contents

- 1 Tox
- 2 Issues with virtualenv
 - Only Python packages can be installed
 - Reproducibility
 - One package manager per language
- 3 GNU Guix
 - Functional package manager
 - Guix environment
 - Solving the issues of virtualenv
- 4 guix-tox
 - Goal
 - Demo
 - Drawbacks



- All of PyPI is not available in GNU Guix (but "guix import works pretty well)
- Some features are not supported
- Guix-tox is alpha software (but one can use "guix-tox -env=virtualenv")
- Only available on GNU/Linux



Join us on Freenode: #guix !

