
Nix for Python developers

Who am I?

- Sysadmin/Ops mostly (Currently OpenStack ops @IBM)
 - Experience with anything Linux
 - Automation, containers
 - Python
 - Dirt bikes
 - github.com/jozko
 - [@jskrablin](https://twitter.com/jskrablin)
 - jozko@zomg.si
-

Nix - what the...

- Purely functional package manager
 - Uses Nix expressions
 - simple functional programming language
 - input -> function -> output
 - build input -> build function -> build output
 - Think JSON like syntax + functions
-

Nix and software packages

- Package installs are atomic
- Install path = cryptographically derived hash
- `/nix/store`

```
Jozkos-MacBook-Pro:~ jozko$ ls -l /nix/store | head
total 156488
dr-xr-xr-x  5 jozko  wheel      170 Jan  1 1970 0077k5r69fv52cdmrhgkszg1h2y278mx-libatomic_ops-7.4.4
-r--r--r--  1 jozko  wheel     2732 Jan  1 1970 0071xdgh6m2pzzrlci54lh1x4y508aq4-show-0.6.tar.gz.drv
dr-xr-xr-x  5 jozko  wheel      170 Jan  1 1970 0085y4izs70asxzkarkalpxh0fmzx4xq-openssl-1.0.2k-dev
-r--r--r--  1 jozko  wheel     2127 Jan  1 1970 00b3z5q1xww9k744xjdyw7z2jxpwlpqn-add-cfmachport.patch
-r--r--r--  1 jozko  wheel     6258 Jan  1 1970 00fzap09mdidkmr2znddcnpjc3snz8m1-edit-distance-0.2.2.1.drv
-r--r--r--  1 jozko  wheel     5350 Jan  1 1970 00hf61akkrkzmviwf86aqrshy3h4nmf-tasty-hunit-0.9.2.drv
-r--r--r--  1 jozko  wheel     3871 Jan  1 1970 00ji2cf1lvviggfypa22wx8zfsgwi6jg-python3.5-cssselect-0.9.1.drv
-r--r--r--  1 jozko  wheel     2731 Jan  1 1970 00xha2xv13r4yy0rwjxa1a25ac3hvvz0s-openSP-1.5.2.tar.gz.drv
-r--r--r--  1 jozko  wheel    17076 Jan  1 1970 011by9hmnhpsm4fagl8fswrkz7fn5ays-libsecurity_sd_cspdl-osx-10.7.5.drv
```

Nix and software packages

- Change an input - change hash
 - you get a 'new package'
 - Will not overwrite previous package
 - Generations - every change is a new generation
 - Can upgrade and rollback per package/entire env
 - Multi user friendly (NixOS by default, requires extra work elsewhere)
 - <https://gist.github.com/joepie91/043a51a7b70be5f50f1d>
 - Multi version friendly
-

Nix and software packages

- It doesn't interfere with system packages at all
- Runs on Linux/macOS
- Installing and testing

```
$ #as a regular sudo user
```

```
$ curl https://nixos.org/nix/install | sh
```

```
$ docker run --rm -it nixos/nix bash
```

Nix and software packages

Uninstalling

```
$ rm -rf /nix
```

```
$ rm -rf ~/.nix*
```

Then reload shell session(s)

Nix vs apt/yum/etc...

Try to:

- manage multiple Python versions install
 - install package as non root user
 - upgrade Python install on CentOS (don't do that, really)
 - modify apt/yum package
-

Nix head first

- Nix packages -> bunch of nix expressions
- see <https://github.com/NixOS/nixpkgs>
- channels point to specific package locations (think apt repo)

```
$ nix-channel --update
```

```
$ nix-env -qa
```

```
$ nix-env -i curl
```

Nix head first

```
$ nix-env --upgrade curl
```

```
$ nix-env --rollback
```

```
$ nix-env --set-flag keep true curl
```

```
$ nix-env -qc #show what can be upgraded
```

```
$ nix-env --list-generations #delete old/switch
```

```
$ nix-env --help
```

Nix head first

- One gotcha - can and will use disk space

```
$ nix-collect-garbage #will remove nix-shell envs!
```

- Modify package - modify Nix expression for package
- Use local nixpkgs checkout/expression and then

```
$ nix-env -i modified_package -I ~/Code/nixpkgs
```

- Or set NIX_PATH
-

Nix with Python - nix-shell

- nix-shell - what is that now?
 - builds an isolated environment using Nix expressions
 - Python libs in potentially mixed envs are no-no
 - Py3 + Py2 + Py3libs + Py2libs == kaboom
 - No such problems using nix-shell
 - Can git it, email to a friend or print using dot matrix printer
-

Nix with Python - nix-shell

- See some samples - <https://github.com/jozko/nixpie>
- Try it if the wifi allows it :)

00-PyRedis

01-CustomPyRedis

02-Python-2.7-scrapy

03-Python-3.5-scrapy

04-OverrideRedis

Nix with Python - nix-shell

- Can be used to build and run env in prod as well
 - *Gotchas:*
 - Not completely isolated - can use something from std env
 - Use `--pure` to build it/use in CI/etc...
 - No `--pure` with devs - they need their dev tools :)
 - MacOS has it's problems - mainly tests (cffi) and include paths
-

Nix with Python - nix-shell

- By default, the unstable channel is used
- Nice to have same expressions versions
- bit.ly/pin-nixpkgs

05-PinPyRedis

```
Jozkos-MacBook-Pro:05-PinPyRedis jozko$ nix-shell .  
downloading 'https://nixos.org/channels/nixos-17.03/nixexprs.tar.xz'... [2108/8686 KiB, 2004.5 KiB/s]  
unpacking 'https://nixos.org/channels/nixos-17.03/nixexprs.tar.xz'...  
error: fetchsvnssh does not support md5 anymore, please use sha256 or sha512  
(use '--show-trace' to show detailed location information)
```

(it worked after updating the pyredis checksum :) Gotcha - it may build for some time.

But why all this work?

- It's a lot of work in the beginning
 - A reproducible development env
 - Will not implode on some other machine
 - Testing apps with different lib/py versions?
 - Testing apps with different Nginx/Gunicorn/whatever versions?
 - CI/testing runs with all above included
 - Did I mention the dot matrix printers?
-

But why all this work?

- If you like some thrill
- Try running it in a production env?
 - not really the scope of this talk
 - it can be done
- <https://nixos.org/nix/manual/#sec-nix-shell>

06-RunNixShellPython

- In the long term -> NixOS
-

Gotchas?

- nixpkgs versions - should be kept consistent
- Need to pin nixpkgs versions - see example
- Packages attribute names in nix-shell != packages names in nix-env
- MacOS build/test failures

```
$ nix-env -qaP | grep redis
```

```
#nix-shell or nix-env -Ai    #nix-env -i
```

```
nixpkgs.redis                redis-3.2.8
```

Sources and sorcerers

- <http://datakurre.pandala.org/2015/10/nix-for-python-developers.html>
 - <https://ariya.io/2016/06/isolated-development-environment-using-nix>
 - <https://nixos.org/wiki/Python>
 - <http://python-on-nix.readthedocs.io/en/latest/tutorial.html>
 - <https://garbas.si/2015/reproducible-development-environments.html>
 - <https://nixos.org/nix/>
 - <http://nixos.org/nixpkgs/manual/#python>
 - Thnx @iElectric github.com/domenkozar
 - Thnx @otobrglez github.com/otobrglez
 - See <http://bit.ly/nixpie> for code samples
-