A gentle introduction to Nix

Bryan Honof—Software Engineer @ Tweag
Why do we Nix?!
Why do we I Nix?!
nixos-rebuild --rollback switch
Intermezzo: Functional Programming

https://serokell.io/blog/introduction-to-functional-programming
https://www.manning.com/books/get-programming-with-haskell
https://parentheticallyspeaking.org/articles/how-not-to-teach-recursion/
Side effects and pure functions

myList = [1,2,3]
myList.reverse()
newList = myList.reverse()
What is `newList` equal to?

```python
myList = [1, 2, 3]
myList.reverse()
newList = myList.reverse()
```
Ruby

myList = [1,2,3]
myList.reverse()  # Does not change myList
newList = myList.reverse()

> [3,2,1]
```javascript
myList = [1,2,3]
myList.reverse()
newList = myList.reverse()
> [1,2,3]
```
```python
myList = [1,2,3]
myList.reverse()
newList = myList.reverse()

> None
```
First-class functions

fn sayHi() {
    return "Hi, ";
}

fn greet(hiFn, s) {
    print(hiFn() + s);
}

greet(sayHi, "Config Management Camp!");

> Hi, Config Management Camp!
Immutability

myList = [1,2,3]
myList.reverse()
newList = myList.reverse()
myList = [1,2,3]
myList.reverse()
newList = myList.reverse()
myList = [1,2,3]
myList = myList.reverse()
newList = myList.reverse()
myList = [1, 2, 3]
myList = myList.reverse()
newList = myList.reverse()
myList = [1,2,3]  
newList = myList.reverse()
Recursion

fn recursiveFun(a) {
    if a <= 0
        return 0;
    return a + recursiveFun(a-1);
}

recursiveFun(5);

> 15
fn recursiveFun(a) {
    if a <= 0
        return 0;
    return a + recursiveFun(a-1);
}

recursiveFun(5);

> 15
fn recursiveFun(a) {
    if a <= 0
        return 0;
    return a + recursiveFun(a-1);
}

recursiveFun(5);

> 15
What is Nix?
What is Nix?

It's complicated...
https://twitter.com/theprincessxena/status/1360214709790728192?s=20&t=F5A1283qndkuq4uU1aRE6Q
+ some horrible edits by myself
The Package set IS not IS IS IS IS IS IS IS
The Language
Nix
The OS
The Package set

https://twitter.com/theprincessxena/status/1360214709790728192?s=20&t=F5A1283qndkuq4uU1aRE6Q
+ some horrible edits by myself
Nix, the package manager

https://github.com/vim/vim
Not too different

$ sudo apt install emacs

$ nix profile install nixpkgs#emacs
Not too different

$ sudo apt install emacs

$ nix profile install nixpkgs#emacs
But it can do some tricks

$ vim
vim: command not found

$ nix shell nixpkgs#vim
...

$ vim --version
VIM - Vi IMproved 9.0 (2022 Jun 28, compiled Jan 01 1980 00:00:00)
$ nix run nixpkgs#vim -- --version
VIM - Vi IMproved 9.0 (2022 Jun 28, compiled Jan 01 1980 00:00:00)
...

$ nix develop nixpkgs#vim 
...

$ unpackPhase && cd source 
...

$ vim
vim: command not found

$ make && make test 
...

$ ./src/vim --version
VIM - Vi IMproved 9.0 (2022 Jun 28, compiled Jan 01 1980 00:00:00)
Included patches: 1-609
Compiled by bryan@X1 
...

$ nix store gc
... store paths deleted, ... MiB freed
Nix, the language

https://nix.dev/tutorials/nix-language
Nix, the programming language

https://nix.dev/tutorials/nix-language
fun = { pkgs, myList }:
pkgs.lib.reverseList myList
...
fun { pkgs = pkgs;
    myList = [ 1 2 3 ]; }
> [ 3 2 1 ]
{ lib, fetchFromGitHub }:
rec {
  version = "9.0.0609";

  src = fetchFromGitHub {
    owner = "vim";
    repo = "vim";
    rev = "v${version}";
    hash = "sha256-...=";
  };
...

... hardeningDisable = [ "fortify" ];

postPatch = ''
...
'';
}


Nix, the OS

https://www.tweag.io/blog/2022-10-11-stable-narhashes/
https://xeiaso.net/talks/nixos-pain-2021-11-10
https://blog.wesleyac.com/posts/the-curse-of-nixos
https://ops.functionalalgebra.com/nixos-by-example/
users.users.bryan = {
    isNormalUser = true;
    home = "/home/bryan";
    description = "Bryan Honof";
    extraGroups = [ "wheel" "networkmanager" ];
    openssh.authorizedKeys.keys = [ "..." ];
};
services.prometheus.exporters.node = {
    enable = true;
    port = 9100;
    enabledCollectors = [ "logind" "systemd" ];
    disabledCollectors = [ "textfile" ];
    openFirewall = true;
};
Nix, the package set

https://github.com/NixOS/nixpkgs
Nix, the mechanisms

https://www.tweag.io/blog/2022-07-14-taming-unix-with-nix/
https://shopify.engineering/what-is-nix
https://nixos.org/guides/nix-pills/
Where to ask for help

https://nixos.org/community/
https://discourse.nixos.org/
https://matrix.to/#/#community:nixos.org
https://discord.com/invite/RbvHtG
THANK YOU!

Let's make something great together.