



NixOS: reproducible, declarative, reliable

A gentle introduction

Thomas Oulevey

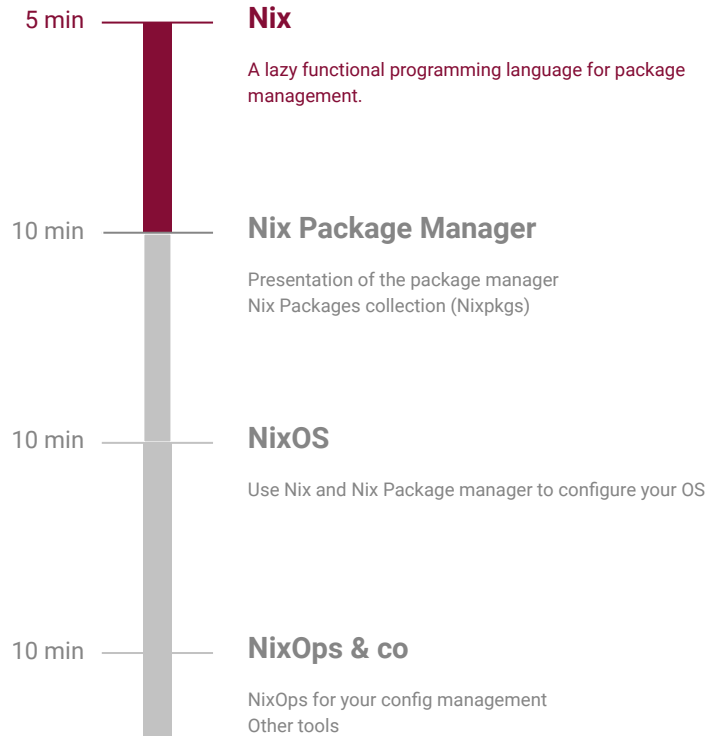
CERN // Beams department // Controls Software and Services

Linux Distribution and Ecosystem Evolution in Control Systems Workshop

ICALEPCS 2021

2021-10-14

Nix



How it started ?

```
$ curl -O https://nixos.org/nix/install
```

```
$ echo "review the script quickly :)"
```

```
$ sh install
```

```
$ nix-env -iA <my favorite package>
```

Nix: a simple example

[link](#)

```
$ cat default.nix
{ pkgs ? import <nixpkgs> {} }:

pkgs.python3Packages.buildPythonApplication {

  pname = "icalepcs21";
  src = ./.;
  version = "0.0.1";

  propagatedBuildInputs = [ pkgs.python3Packages.flask ];

}

$ nix-build
[...]
/nix/store/hsqm4hw3jwymq3h53y6ixhwqqy854h8p-icalepcs21-0.0.1
```

Nix









- A “Purely functional package manager” started in 2003
 - No side effects, such as destructively updating or deleting files used by other packages.
- package == derivation
- i686, x86_64 and arm64 architectures support
- Linux and MacOS support
- Focus on reproducibility and portability of the builds
- Installed in `/nix/store/`
- Nix uses a sandbox to isolate builds
- helper define all steps for pkg management `unpackPhase, patchPhase, [...], installPhase`

Nix “unfree”

- A word on unfree
- https://nixos.wiki/wiki/FAQ/How_can_I_install_a_proprietary_or_unfree_package%3F
- Useful for oracle clients, vscode etc...

Nix: root.cern.ch example

[link](#)

 veprbl Merge pull request #136016 from veprbl/pr/root_6_24_04 ...	
..	
 5.nix	root5: fix for gcc10
 default.nix	root: build with -Dimt=ON
 disable_libc_dicts_root5.patch	root5: fix build
 purify_include_paths_root5.patch	root5: fix build
 setup-hook.sh	root: fix setupHook to work with "set -u"
 sw_vers.patch	root: 6.22.08 -> 6.24.00
 sw_vers_root5.patch	root5: init at 5.34.36

nix-shell

- Give user a shell by setting up the necessary environment variables to develop on a derivation.

```
$ nix-shell
```

```
[...]
```

```
Installing collected packages: icalepcs21
```

```
  Running setup.py develop for icalepcs21
```

```
Successfully installed icalepcs21-0.0.1
```

```
Finished executing setuptoolsShellHook
```

```
[nix-shell:~/src/icalpecs21]$ echo $propagatedBuildInputs
```

```
/nix/store/zlffsy67mc9g5a3vkzvw5mpjwb89h07z-python3.9-Flask-2.0.1
```

```
/nix/store/i1m8r7mv8h47wr850cdsxksy22lv6gsz-python3-3.9.6
```


Nix Package Manager

5 min	Nix A lazy functional programming language for package management.
10 min	Nix Package Manager Presentation of the package manager Nix Packages collection (Nixpkgs)
10 min	NixOS Use Nix and Nix Package manager to configure your OS
10 min	NixOps & co NixOps for your config management Other tools

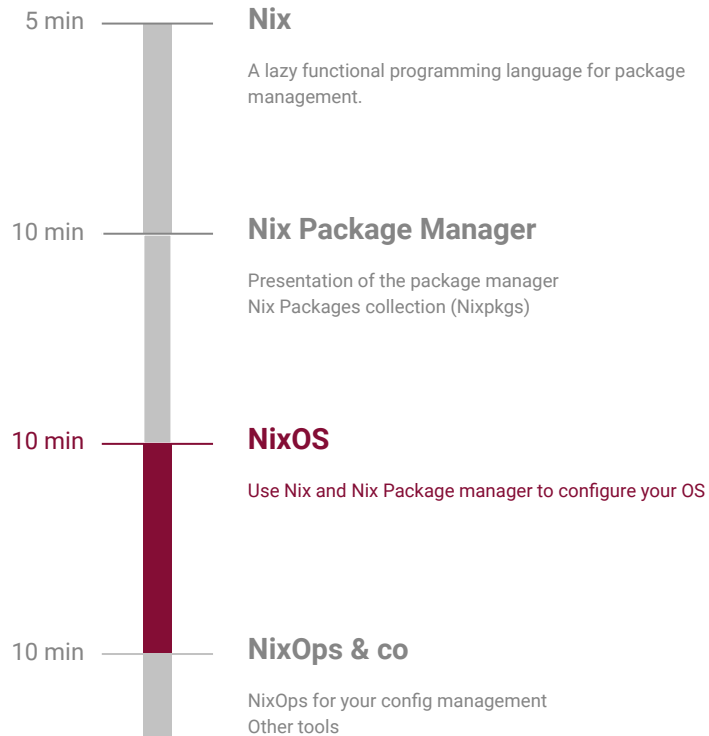
Nix Package Manager

- [Nixpkgs](#) git repository which contains definitions O(80k) packages.
- git commit hash of a particular revision can be used as a snapshot
- alternatives which are configured through overlays.
- All used artifacts are cached (identified by sha256)
- <https://search.nixos.org/packages>

Nix Package Manager

action	CentOS/Fedora	Debian/Ubuntu	Nix
<i>update package list</i>	yum clean all	apt update	nix-channel --update
<i>search</i>	yum search <query>	apt search <query>	nix search <query>
<i>install</i>	yum install <package>	apt install <package>	nix-env -i <package>
<i>upgrade installed</i>	yum update	apt upgrade	nix-env -u
<i>remove</i>	yum remove <package>	apt remove <package>	nix-env -e <package>
<i>undo last operation</i>	nix-env --rollback
<i>list installed</i>	rpm -qa	dpkg -l	nix-env -q

NixOS



NixOS

“NixOS is a **Linux distribution** based on the Nix package manager and build system. It supports reproducible and declarative system-wide configuration management as well as **atomic upgrades and rollbacks**, although it can additionally support imperative package and user management. In NixOS, all components of the distribution — including the kernel, installed packages and system configuration files — are **built by Nix from pure functions** called Nix expressions.”

[NixOS wiki](#)

Installation

- Installation guide : https://nixos.wiki/wiki/NixOS_Installation_Guide
- The best way to evaluate is playing with virtual machines on common cloud providers right ? :)
 - [Nixos-generators](#)
 - `nixos-generate -f {openstack, azure, amazon, gce, vmware, virtualbox}`

Few notes

- Packages root installs are system-wide.
System-wide packages are in `/run/current-system/sw/`
Modify the configuration file and apply changes with `nixos-rebuild switch`

1. Add to `/etc/nixos/configuration.nix`:

```
environment.systemPackages = with pkgs; [  
  wget # let's assume wget was already present  
  emacs  
];
```

2. Run :

```
$ sudo nixos-rebuild switch
```

configuration.nix

```
NixOS (Generation 24 NixOS 18.03.132597.9df3c53f38b, Linux Kernel 4.14.47, Bu
NixOS (Generation 25 NixOS 18.03.132597.9df3c53f38b, Linux Kernel 4.14.47, Bu
NixOS (Generation 26 NixOS 18.03.132597.9df3c53f38b, Linux Kernel 4.14.47, Bu
NixOS (Generation 27 NixOS 18.03.132597.9df3c53f38b, Linux Kernel 4.14.47, Bu
NixOS (Generation 28 NixOS 18.03.132597.9df3c53f38b, Linux Kernel 4.14.47, Bu
NixOS (Generation 29 NixOS 18.03.132597.9df3c53f38b, Linux Kernel 4.14.47, Bu
NixOS (Generation 30 NixOS 18.03.132597.9df3c53f38b, Linux Kernel 4.14.47, Bu
NixOS (Generation 31 NixOS 18.03.132597.9df3c53f38b, Linux Kernel 4.14.47, Bu
NixOS (Generation 32 NixOS 18.03.132597.9df3c53f38b, Linux Kernel 4.14.47, Bu
NixOS (Generation 33 NixOS 18.03.132597.9df3c53f38b, Linux Kernel 4.14.47, Bu
EFI Default Loader
Reboot Into Firmware Interface
```


configuration.nix

```
environment.systemPackages = [
  pkgs.awscli
  pkgs.nixops
  pkgs.terraform-full
  pkgs.tmux
];

nix.gc.automatic = true;
nix.gc.dates = "daily";
nix.gc.options = '--max-freed "$((30 * 1024**3 - 1024 * $(df -P -k /nix/store | tail -n 1 | ${pkgs.gawk}/bin/awk '{ print $4 }'))))"';

services.openssh.enable = true;

# Temporary hack until we have proper users/roles.
services.openssh.extraConfig = ''
  AcceptEnv AWS_ACCESS_KEY_ID AWS_SECRET_ACCESS_KEY FASTLY_API_KEY GIT_AUTHOR_NAME GIT_AUTHOR_EMAIL GIT_COMMITTER_NAME GIT_COMMITTER_EMAIL
'';

boot.loader.grub.enable = true;
boot.loader.grub.device = "nodev";

fileSystems."/ " = {
  fsType = "ext4";
  device = "/dev/disk/by-label/nixos";
};

fileSystems."/scratch" = {
  autoFormat = true;
  fsType = "ext4";
  device = "/dev/nvme1n1";
};
```

Reproducible



Reproducible

Nix builds packages in isolation from each other. This ensures that they are reproducible and don't have undeclared dependencies, so **if a package works on one machine, it will also work on another.**

- My random notes
 - Very useful to distribute sdk
 - Your build will work the same on different Operating Systems
 - `$ nix-shell -p icalepcs21`
 - Big investment in package management for all teams

Declarative



Declarative

Nix makes it **trivial to share development and build environments** for your projects, regardless of what programming languages and tools you're using.

- My random notes
 - “virtualenv” of everything
 - `nix-shell` can be used to have contained build environment.
 - Trivial to build OS images
 - Trivial to cross compile

Reliable



Reliable

Nix ensures that installing or upgrading one package **cannot break other packages**. It allows you to **roll back to previous versions**, and ensures that no package is in an inconsistent state during an upgrade.

- My random notes
 - Rollback is very useful
 - Rebooting in a known working configuration and the integration with grub is super powerful

NixOps & CO

5 min	Nix A lazy functional programming language for package management.
10 min	Nix Package Manager Presentation of the package manager Nix Packages collection (Nixpkgs)
10 min	NixOS Use Nix and Nix Package manager to configure your OS
10 min	NixOps & co NixOps for your config management Other tools

NixOPS & other tools

- NixOps is a tool for deploying to NixOS machines in a network or the cloud.
 - Same language
 - User guide : <https://releases.nixos.org/nixops/nixops-1.5/manual/manual.html>
 - No clear winner in the space yet e.g: <https://github.com/DBCDDK/morph>
- Hydra : <https://github.com/NixOS/hydra>
- Nix-GUI <https://github.com/nix-gui/nix-gui> : editing derivation with a GUI

Questions ?

BONUS

- CoreOS & Fedora Silverblue ; atomic updates and user space containers

References

- LHCb experiment take on Nix :
https://cds.cern.ch/record/2700235/files/10.1051_epjconf_201921405005.pdf
- Nix for software deployment in high energy physics :
<https://www.youtube.com/watch?v=Ee8k97Rx3DA>