



# Nix: maîtrisez les sources de vos paquets

Capitole du Libre 2024

2024-11-15

Guilhem Saurel





#### Cette présentation

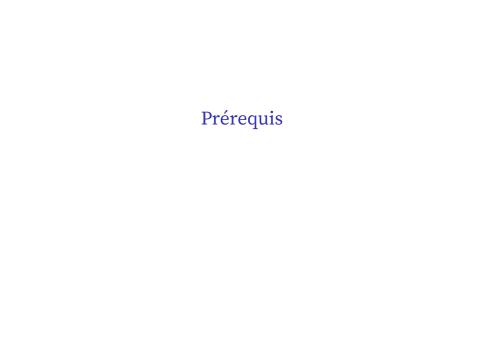
# Disponible à

#### Source

#### Sous license



https://creativecommons.org/licenses/by-sa/4.0/



# Paquet

### Dans le monde UNIX:

un logiciel



#### Dans le monde UNIX:

- un logiciel
- sa configuration
- sa documentation
- son installation
- ses dépendances



### Nix

C'est (entre autres) un gestionnaire de paquets

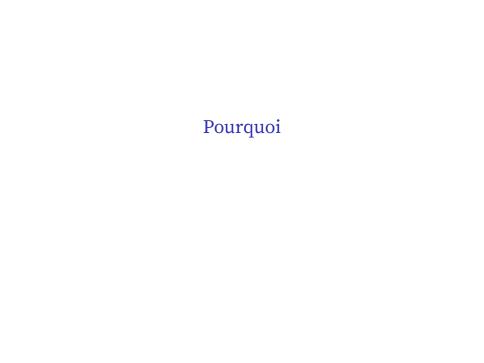


#### Nix

C'est (entre autres) un gestionnaire de paquets

The Purely Functional Software Deployment Model

— Eelco Dolstra, PhD Thesis, 2006





# Pour installer des logiciels

- firefox
- vlc
- libreoffice



# Pour installer des logiciels

- firefox
- vlc
- libreoffice
- linux
- systemd



# Pour installer des logiciels

- firefox
- vlc
- libreoffice
- linux
- systemd
- gcc
- pytorch
- nextcloud
- angular
- nerdfonts



Quelle version de de quel logiciel ?



- Quelle version de de quel logiciel ?
- Récupérée depuis quelle source ?



- Quelle version de de quel logiciel ?
- Récupérée depuis quelle source ?
- Compilée de quelle manière ?



- Quelle version de de quel logiciel ?
- Récupérée depuis quelle source?
- Compilée de quelle manière ?
- Avec quelles dépendances ?



### Pour modifier ce que vous voulez

- Utiliser une version particulière
- Ajouter des patchs
- Modifier certains paramètres de compilation



### Pour modifier ce que vous voulez

- Utiliser une version particulière
- Ajouter des patchs
- Modifier certains paramètres de compilation
- Ajouter des modules noyau
- Activer des services systemd



# Modulaire, Versionné & Reproductible

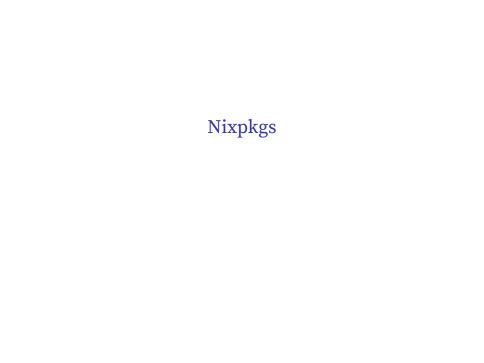
- configuration as code
- rollbacks
- sandbox



### Modulaire, Versionné & Reproductible

- configuration as code
- rollbacks
- sandbox

pour un paquet, ou un environnement, ou un OS





# Un monorépo

- https://github.com/NixOS/nixpkgs
- repology: 100k paquets



# Un monorépo

- https://github.com/NixOS/nixpkgs
- repology: 100k paquets
- ajoutez le votre: PR



#### NixOS stable & unstable

- une rolling release: nixos-unstable
- une version stable tous les 6 mois: nixos-24.11



#### NixOS stable & unstable

- une rolling release: nixos-unstable
- une version stable tous les 6 mois: nixos-24.11
- master



### custom forks / branches

bring your own master



#### custom PR

applyPatches!





# Encore plus de paquets

- 264 dépôts indépendants
- 4852 paquets en plus



#### Et les votres!

https://github.com/nix-community/NUR/blob/master/READ ME.md#how-to-add-your-own-repository

**NUR** 



# Chaque paquet à sa source

un dépôt git avec un flake.nix (et un flake.lock)



### Chaque paquet à sa source

un dépôt git avec un flake.nix (et un flake.lock)

#### outputs

- paquets
- applications
- devShells
- Modules NixOS
- tests
- templates
- overlays



#### Flake inputs

- autres flakes:
  - nixpkgs/nixos-unstable
  - github:stack-of-tasks/pinocchio
- urls classiques
  - https://github.com/NixOS/nixpkgs/pull/354231.patc



# Dépôts de flakes

- Github / Gitlab / ...
- https://flakehub.com/
- https://flakestry.dev/
- https://github.com/cafkafk/rime

Flakes



### Merci:)

#### Contact



Figure 1: Guilhem Saurel / @nim65s

#### Slides

https://homepages.laas.fr/gsaurel/talks/
nix-sources.pdf

#### Voir aussi

- "Home-manager", CDL 2023
- "Nix et NixOS: fonctionnement technique", demain 11:30, A202
- "NixOS embarqué dans la stratosphère", demain 15:00, B00



### flake.nix

```
inputs = {
  nixpkgs.url = "nixpkgs/nixos-unstable";
  flake-parts.url = "github:hercules-ci/flake-parts";
  pin = {
    url = "https://github.com/NixOS/nixpkgs/pull/354231.patch";
    flake = false:
 };
};
outputs = inputs@{ nixpkgs, ... }:
inputs.flake-parts.lib.mkFlake { inherit inputs; } {
  systems = [ "x86_64-linux" "aarch64-linux" "aarch64-darwin" ];
  perSystem = { pkgs, system, ... }: {
    _module.args.pkgs = import ./patched-nixpkgs.nix {
      inherit nixpkgs system;
      patches = [ inputs.pin ];
    };
    packages.default = pkgs.python3.withPackages (p: [p.pinocchio]);
 };
};
```

# patched-nixpkgs.nix

```
{ nixpkgs, patches, system, ... }:
let
  super = import nixpkgs { inherit system; };
in
import (super.applyPatches {
  inherit patches;
  name = "patched nixpkgs";
  src = nixpkgs;
}) { inherit system; }
```