

Nix, Nixpkgs & NixOS

2024-06-14

Guilhem Saurel

Available at

<https://homepages.laas.fr/gsaurel/nix.pdf>

<https://gitlab.laas.fr/gsaurel/talks> : nix.md

Under License



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

Discussions

#nix:laas.fr

Introduction

- Linters:
 - faciliter l'écriture des sources
 - standardiser la forme des sources

- Linters:
 - faciliter l'écriture des sources
 - standardiser la forme des sources

- git & CI:
 - gérer le passé, le présent, et l'avenir des sources
 - faciliter les modifications des sources

- Linters:
 - faciliter l'écriture des sources
 - standardiser la forme des sources

- git & CI:
 - gérer le passé, le présent, et l'avenir des sources
 - faciliter les modifications des sources

- Build: transformer les sources en fichiers utilisables

- **Linters:**
 - faciliter l'écriture des sources
 - standardiser la forme des sources

- **git & CI:**
 - gérer le passé, le présent, et l'avenir des sources
 - faciliter les modifications des sources

- **Build:** transformer les sources en fichiers utilisables

- **Packaging:** mettre ces fichiers au bon endroit
 - dépendances
 - distributions

Cahier des charges

- source
- binaires

- source
- binaires
- cache

Mêmes sources + recette => mêmes résultats

Mêmes sources + recette => mêmes résultats

- OS (eg. macos / linux)
- CPU (eg. x86_64 / arm64)
- Compilateur (eg. gcc / clang)
- Environnement
- Options automatiques

=> Isolation

Mêmes sources + recette => mêmes résultats

- OS (eg. macos / linux)
- CPU (eg. x86_64 / arm64)
- Compilateur (eg. gcc / clang)
- Environnement
- Options automatiques

=> Isolation

Gains

- Si ça a été compilé une fois, pas besoin de recompiler
- Si ça marche sur un ordinateur, ça marche sur un autre

Limites

- dépendances: nombre de paquets déjà disponibles

Ref. Own your CI with Nix. FOSDEM, Hufschmitt (2024)

La solution

Ref. The purely functional software deployment model. PhD,
Dolstra (2006)

Impératif

lancer la commande:

```
echo 'enable = true' >> /etc/toto.conf
```

Déclaratif

Définir que /etc/toto.conf soit enable = true

Simplement modifier n'importe quel paramètre

- projet
- user
- OS

`/nix/store` est read-only

«Réaliser» un paquet n'agit que sur ce paquet

```
stdenv.mkDerivation rec {  
  pname = "hello";  
  version = "2.12.1";  
  
  src = fetchurl {  
    url = "mirror://gnu/hello/hello-${version}.tar.gz";  
    hash = "sha256-jZkUKv2SV28wsM18tCqNxoCZmLxdYH2...=";  
  };  
  
  doCheck = true;  
})
```

```
nix-repl> pkgs.hello  
«derivation /nix/store/img71klf2wzx867q5ldhf0zqc3wl0m  
hello-2.12.1.drv»
```

```
nix-repl> pkgs.hello.outPath  
"/nix/store/bw9z0jxp5qcm7jfp4vr6ci9qynjyaaip-  
hello-2.12.1"
```

```
$ tree /nix/store/bw9z0jxp...-hello-2.12.1
```

```
bw9z0jxp...-hello-2.12.1/
```

```
├── bin/
│   └── hello*
├── share/
│   ├── info/
│   │   └── hello.info
│   ├── locale/
│   │   └── [...]
│   └── man/
│       └── man1/
│           └── hello.1.gz
```

```
$ ldd /nix/store/bw9z0jxp...-hello-2.12.1/bin/hello  
linux-vdso.so.1 (0x00007fab657f1000)  
libc.so.6 => /nix/store/35p...-glibc-2.39-5/lib/libc.so  
/nix/store/35p...-glibc-2.39-5/lib/ld-linux-x86-64.so.2
```

- 100k paquets
- 7k contributeurs
- 3k mainteneurs
- 640k commits
- 20 ans

- dépôt git + flake.nix
- entrées: autres flakes
- sorties:
 - paquets
 - applications
 - tests
 - dev shells
 - NixOS configurations
 - Home-Manager configurations

```
{ hello }:  
hello.overrideAttrs (_: {  
  postPatch = ''  
    substituteInPlace src/hello.c \  
      --replace world LAAS  
  '';  
  doCheck = false;  
})
```

```
{
  description = "hello LAAS";
  inputs.nixpkgs.url = "github:NixOS/nixpkgs";
  outputs =
    { nixpkgs, ... }:
    let
      system = "x86_64-linux";
      pkgs = nixpkgs.legacyPackages.${system};
      hello-laas = pkgs.callPackage ./package.nix {};
    in
    {
      packages.${system}.default = hello-laas;
      devShells.${system}.default = pkgs.mkShell {
        inputsFrom = [ hello-laas ];
      };
    };
};
```

NixOS

```
{
  imports = [ ./hardware-configuration.nix ];
  boot.loader.systemd-boot.enable = true;
  networking.hostName = "loon";
  networking.networkmanager.enable = true;
  time.timeZone = "Europe/Paris";
  console.keyMap = "fr-bepo";
  users.users.gsaurel = {
    shell = pkgs.fish;
    isNormalUser = true;
    description = "Guilhem Saurel";
    extraGroups = [ "networkmanager" "wheel" ];
  };
  environment.systemPackages = [ pkgs.git pkgs.vim ];
  services.openssh.enable = true;
}
```

- Mise à jour atomiques

- Mise à jour atomiques
- Rollbacks atomiques

- Mise à jour atomiques
- Rollbacks atomiques
- Garbage collector

Communauté: projets

```
home = {  
  packages = [ ... ];  
  home.file = {  
    ".config/dfc/dfcrc".source =  
      ./config/dfc/dfcrc;  
  };  
  sessionVariables.PAGER = "vim -c PAGER -";  
  programs.atuin = {  
    enable = true;  
    flags = [ "--disable-up-arrow" ];  
  };  
  programs.git = {  
    enable = true;  
    delta.enable = true;  
    lfs.enable = true;  
    userName = "Guilhem Saurel";  
    userEmail = "guilhem.saurel@laas.fr";  
  };  
};
```

```
$ echo "use flake" >> .envrc  
$ direnv allow
```

<https://github.com/nix-community/nix-direnv>

Cache binaire

<https://www.cachix.org/>

```
disko.devices.disk.my-disk = {  
  device = "/dev/sda";  
  type = "disk";  
  content = {  
    type = "gpt";  
    partitions = {  
      ESP = {  
        type = "EF00";  
        size = "500M";  
        content = {  
          type = "filesystem";  
          format = "vfat";  
          mountpoint = "/boot";  
        };  
      };  
    };  
  };  
};
```

Installation automatique

<https://github.com/nix-community/nixos-anywhere>

Gestion des secrets par utilisateur et par machine

<https://github.com/Mic92/sops-nix>

Framework CI/CD

<https://github.com/Mic92/sops-nix>

Formatte tous les langages

<https://github.com/numtide/treefmt-nix>

steps:

- uses: actions/checkout@v4
- uses: DeterminateSystems/nix-installer-action@v1
- uses: DeterminateSystems/magic-nix-cache-action@v
- run: nix flake check
- run: nix build

Fédération de dépôts personnels

<https://nur.nix-community.org/>

- flake & devShells
- shell.nix

- flake & devShells
- shell.nix
- cachix/devenv
- numtide/devshell
- jetify-com/devbox
- flox/flox

- flake & devShells
- shell.nix
- cachix/devenv
- numtide/devshell
- jetify-com/devbox
- flox/flox
- direnv

Collection de modules pour différents devices

<https://github.com/NixOS/nixos-hardware/>

- <https://mynixos.com/>
- <https://github.com/snowfallorg/nixos-conf-editor>
- <https://github.com/snowfallorg/nix-software-center>

- nixos
- VMs
- containers
- isos
- macos
- freebsd
- android
- projets par langage

Communauté: Communication

- <https://github.com/NixOS/>
- <https://github.com/nix-community/>

<https://discourse.nixos.org/>

#space:nixos.org

Officielle

- <https://nix.dev/>
- <https://wiki.nixos.org/>
- <https://nixos.org/learn/>: Nix{,pkgs,OS} manual
- <https://nixos.org/guides/nix-pills/>

Communauté

- <https://zero-to-nix.com/>
- <https://nixos-and-flakes.thiscute.world/>

- <https://search.nixos.org/>
- <https://home-manager-options.extranix.com/>
- <https://noogle.dev/>
- <https://code.tvl.fyi/about/nix/nix-1p>

Communauté: entreprises

- Determinate Systems
- Flox
- Tweak / Modus Create
- Numtide
- Cachix
- Hercules-CI

- Anduril

Communauté: projets connexes

Ref. Functional package management with guix. arXiv, Courtès
(2013)

<https://guix.gnu.org/>

Nouvelle implémentation, en Rust

<https://tvix.dev/>

Fork de NixCPP

<https://lix.systems/>

Un alternative à l'écosystème Nix

<https://aux.computer/en/>

Communauté: humains

...

Attention

Apprendre un nouveau langage difficile

Toute la chaîne doit être connue

- ccache
- sccache

Explosion de la taille de la communauté

- Assemblée Constitutante
- <https://nixpkgs.zulipchat.com/>

Conclusion

- De nouveaux paradigmes
- Un large pannel d'applications pratiques
- Des tas de problèmes résolus
- Beaucoup d'activités et d'engouements

=> Une aventure qui vaut le coup, allez-y !

- De nouveaux paradigmes
- Un large pannel d'applications pratiques
- Des tas de problèmes résolus
- Beaucoup d'activités et d'engouements

=> Une aventure qui vaut le coup, allez-y !

Mais prenez votre temps.

Questions ?

Links

- <https://homepages.laas.fr/gsaurel/home-manager.pdf> (enregistrement)
- <https://homepages.laas.fr/gsaurel/nix-rob.pdf>

References

- Courtès, Ludovic. 2013. “Functional Package Management with Guix.” *arXiv Preprint arXiv:1305.4584*.
- Dolstra, Eelco. 2006. “The Purely Functional Software Deployment Model.” PhD thesis, Utrecht University. <https://edolstra.github.io/pubs/phd-thesis.pdf>.
- Hufschmitt, Théophane. 2024. *Own Your CI with Nix*. <https://fosdem.org/2024/schedule/event/fosdem-2024-2282-own-your-ci-with-nix/>.