

# Flakes

Les Nix Flakes sont des dossiers (généralement des repo git) ayant à leur racine un fichier `flake.nix`. Ce fichier indique des "inputs" et des "outputs". Ainsi il permet de prendre des expressions Nix en entrée et donner toutes sortes de sorties.

Par exemple, on peut construire un flake qui:

- Setup un environnement de dev
- Build le paquet
- Crée une image docker minimales
- Build et effectue des tests sur le résultat
- Crée des configurations NixOS
- Crée des overlays de nixpkgs

et bien d'autres.

Un flake a également l'avantage par rapport aux channels, d'être vraiment reproductible.

Lorsqu'un flake est créé et lancé pour la première fois, un fichier `flake.lock` est automatiquement créé. Ce fichier va spécifier chaque dépendance au commit prêt. Contrairement aux channels de Nixpkgs qui sont spécifiés par "release" et non par commit.

Les flakes bénéficient aussi de très bons outils (flake-utils et les commandes nix). Les flake-utils donnent des outils puissants pour créer des flakes et les commandes nix sont des outils très puissants et simple pour *utiliser* ces flakes.

Ainsi par exemple :

- `nix build` va build la dérivation se trouvant dans l'output nommé `defaultPackage.<system>` ou `packages.<system>.default`
- `nix shell` va entrer dans un shell où le package/dérivation dit "default" est accessible dans le `$PATH`
- `nix run` va lancer le binaire du package "default" qui correspond au nom de la dérivation
- `nix develop` qui va donner tous les outils de développement pour build le paquet et développer dans le `$PATH` (va d'abord chercher après `devShells.<system>.default` ou `devShell.<system>` avant de chercher pour le package default.
- `nix flake show` donne la liste de tous les outputs d'un flake
- `nix flake info` donne des métadonnées générales sur le flake
- `nix flake new` crée un nouveau flake
- `nix flake update` met à jour le fichier `nix.lock`
- `nix flake check` qui effectue les checks sur le flake
- `nix profile install` pour installer le defaultPackage du flake dans son profile
- `nix profile list` pour voir la liste des flakes installés

- `nix profile remove <id>` pour supprimer un flake de son profile

Je vais expliquer plus en détail comment ça fonctionne dans la partie pratique.

Il est aussi possible de créer des alias et d'utiliser `nixpkgs` via les flakes. Les alias peuvent être défini grâce au *flake registery*. Par exemple le flake `nixpkgs` est un alias pour `github:NixOS/nixpkgs/nixpkgs-unstable`. Et imaginons que je veut installer cowsay temporairement via un flake je peux simplement faire.

```
nix shell nixpkgs#cowsay
```

---

Revision #4

Created 10 July 2023 21:01:06 by SnowCode

Updated 12 July 2023 12:15:13 by SnowCode