Intégration continue et Nix

Ryan Lahfa

30 juin 2020

L'intégration continue

Jenkins

Jenkins est le poids lourd des serveurs d'intégration continue, très utilisé en entreprise, particulièrement dans les contextes où on voit des fonctionnalités spéciales (Docker, KVM, etc.)

Jenkins

Jenkins est le poids lourd des serveurs d'intégration continue, très utilisé en entreprise, particulièrement dans les contextes où on voit des fonctionnalités spéciales (Docker, KVM, etc.)

En revanche, déployer proprement un Jenkins n'est pas chose aisée, en TP d'application de NixOS, vous trouverez dans `5_jenkins`, un fichier à améliorer pour déployer Jenkins en quelques lignes, pour le reste: https://nixos.org/nixos/options.html#jenkins

Hydra

Hydra est l'équivalent "NixOS" de Jenkins, plus fonctionnel et actuellement gère l'ensemble des paquets de NixOS, vous pouvez voir sur https://hydra.nixos.org toutes les constructions en cours, reproducible à travers `curl` et `nix`, avec des mécanismes de sandboxing intéressant.

Hydra

Hydra est l'équivalent "NixOS" de Jenkins, plus fonctionnel et actuellement gère l'ensemble des paquets de NixOS, vous pouvez voir sur https://hydra.nixos.org toutes les constructions en cours, reproducible à travers `curl` et `nix`, avec des mécanismes de sandboxing intéressant.

Par ailleurs, notez que l'écosystème Nix possède nixpkgs, l'un des plus grands référentiels de paquet, en effet: 44 019 paquets, c'est le premier sur deux classements au moins, d'après https://repology.org/ avec l'AUR et les Gentoo Ports qui suivent à côté. Clairement, Docker n'en a pas autant. Et pourtant, c'est autant d'images Docker de combinaisons de paquets, c'est 2^{44019} au moins d'images Docker réalisables, hein. Je vous laisse regarder le nombre.

TP: GitHub Actions

L'objectif c'est de reprendre: https://github.com/mangaki/zero pour lui rajouter un GitHub Actions.

(a) Forker Mangaki Zero
(b) Supprimer le .github/workflow/main.yml et le publish aussi.
(c) Vous pouvez conserver les fichiers Nix.

Maintenant, il faut écrire le GitHub Action et à la fin, on va rajouter un test si on y arrive, vous n'avez pas à rajouter du caching, c'est bonus, ne faites pas une build matricielle, juste le minimum.

Cachix

À présent, qu'est ce qu'est Cachix ?

Classiquement, un CI va mettre en cache les artéfacts de construction, mais bien souvent, en raison du fractionnement des gestionnaire de paquets, en raison des piètres systèmes de cache (hein, le système de layer de Docker?), il est très difficile d'avoir un cache intelligent.

Avec https://cachix.org/, vous pouvez (si vous êtes open source) gratuitement bénéficier d'un cache magique en quelques secondes, en plus de cela, vous pouvez lui faire confiance (système de clef publique/privée qui signe un cache binaire, protège contre les supply attacks).

Cachix et Mangaki Zero

Vous remarquerez que Mangaki Zero utilise Cachix pour stocker les artéfacts de construction, et produit donc des temps de tests complètement incomparable avec des technologies Docker de GitLab CI (trop lente).

Cachix et Mangaki Zero

Vous remarquerez que Mangaki Zero utilise Cachix pour stocker les artéfacts de construction, et produit donc des temps de tests complètement incomparable avec des technologies Docker de GitLab CI (trop lente).

Si vous n'êtes toujours pas convaincu, lisez https://nixos.org/nixos/nix-pills/why-you-should-give-it-a-try.html.