FIELD NOTE / NIXOS

NixOS 01 : Instalasi dan Setup Nixpkgs di MacOS

Petruknisme · 11 Mar, 2022 · 4 min to read

Beberapa hari yang lalu kawan gue posting tentang NixOS dan Nixpkgs di sosmed. Gue tertarik dengan keunggulan yang ditawarkan oleh Nix ini yang salah satunya adalah *isolated environment* dan juga perihal sistem yang portable, jadinya kalau misal ganti device bisa langsung restore dan deploy sistem sesuai dengan yang pernah kita setup sebelumnya. Soal ini sih memudahkan banget, jadinya nggak perlu lagi tuh setup-setup os tiap ganti distro atau ganti device. Niat gue sih awalnya mau gantiin brew jadi nixpkgs.

Kalau dibayangin mungkin mirip sistemnya Ansible kali ya, yang tinggal jalanin script langsung OS nya terkonfigurasi sendiri. Nah bedanya, Nixpkgs ini dia bikin sistem volume baru untuk NixOS. Jadi berasa pake ansible tapi ada vm juga buat NixOS. Pokoknya gitu lah, kalau ntar udah bisa jelasin lebih detail, bagian ini bakal diedit wkkwkw

Gue sih belum paham-paham amat, tapi ya gue bermaksud buat bikin dokumentasi siapa tau ada yang butuh ye kan. Nah gue juga kelupaan buat screenshot atau copy log konsol. Jadi kemungkinan gue bakal kasih tau step by stepnya aja.

Note: Karena gue di sini pake MacOS, jadi gue cuman bakal jelasin gimana di MacOS. Kalau lo pake distro Linux, sila buat cek officialnya aja ya.

Install Nix

Pertama-tama elo mesti install Nix dulu biar bisa pakai fitur-fitur yang ada di NixOS. Instal nya gampang banget, tinggal jalanin command ini aja.

```
1 | sh <(curl https://nixos.org/nix/install) --daemon
```

Setelah itu, ntar bakal muncul popup untuk konfirmasi. Yang gue suka, nix ini transparan banget soal apa yang dia lagi lakuin dan apa yang mau diinstal. Jadi elo nggak perlu khawatir, semuanya bisa elo review ulang.

Saran gue, pilih multiuser pas proses pemilihan awal banget itu. Abis itu ikutin aja yang dia perluin pas instalasi. Gue jamin, elo bisa kok.

Install Nix-Darwin

Licensed under CC BY-SA 4.0

Kalau elo cuman butuh NixOS, step ini nggak perlu. Tapi gue sih butuh karena gue mau konfig MacOS via **configuration.nix** biar nantinya nggak perlu klak-klik lagi setting MacOS kalau pindah device. Kita bakal pake tools **nix-darwin** (https://github.com/LnL7/nix-darwin).

```
1 nix-build https://github.com/LnL7/nix-darwin/archive/master.tar.gz -A inst
```

Nah, kalau udah dijalanin tunggu sampe prosesnya selesai. Kalau elo mau instal secara default, elo bisa jalanin command ini

```
1 ./result/bin/darwin-installer
```

Tapi, gue sih nggak pake gituan. Jadinya instal manual pake yang namanya flakes dan nix nya gue config manual.

Abis beres nix-build, kita butuh config sebelum instalasi karena ada problem di MacOS salah satunya apfs dan synthetic.

```
1 $ echo -e "run\tprivate/var/run" | sudo tee -a /etc/synthetic.conf $ sudo
```

Ini dilakuin biar pas elo instalasi ntar nggak muncul error, jadi baiknya config dari awal.

Enable experimental-features

Karena kita bakal pake flakes yang status nya masih experimental, makanya perlu tuh tambahin ini di */etc/nix/nix.conf* sesuai dengan referensi dari wikinya NixOS **Flakes** (https://nixos.wiki/wiki/Flakes)

/etc/nix/nix.conf

```
1 experimental-features = nix-command flakes
```

Untuk sekarang, aktifin aja dulu ya. Nanti bakal dibahas dilain waktu soal flakes ini. Jadi nggak perlu khawatir.

Licensed under CC BY-SA 4.0

Config Nixpkgs

Seperti yang udah gue bilang sebelumnya di atas, tulisan ini bakal pake config yang udah jadi dan tinggal pake. Config ini terinspirasi dari **malob/nixpkgs** (https://github.com/malob/nixpkgs) yang gue dan **ri7nz/nixpkgs** (https://github.com/ri7nz/nixpkgs) kembangkan agar lebih sederhana. Feel free untuk pilih mana aja. Ditulisan ini sih gue pake config yang gue bikin sendiri.

Pertama, clone dulu repo dari config gue

```
// with SSH

git clone git@github.com:aancw/nixpkgs ~/.config/nixpkgs

// OR with HTTP
git clone https://github.com/aancw/nixpkgs ~/.config/nixpkgs
```

Kalau udah beres, masuk ke direktori *~/.config/nixpkgs*

```
cd ~/.config/nixpkgs
```

Config yang gue buat terdiri dari beberapa struktur file dan folder

Project structure

```
❯ tree .
.
├── README.md
├── darwin.nix
├── default.nix
├── flake.lock
├── flake.nix
├── flake.nix.simple
├── home
│   ├── default.nix
│   ├── packages.nix
│   └── shells.nix
```

Licensed under CC BY-SA 4.0

```
13  ├── nixpkgs.nix
14  └── result
```

- darwin.nix ini isinya config buat ngatur MacOS kita yang dibutuhkan oleh **nix-darwin** (https://github.com/LnL7/nix-darwin), sila cek github buat config-config tambahan yang dibutuhkan. Di sini lo bisa ubah shell yang digunakan, font yang lo pake, aktifin konfig dari shell yang elo pake, keyboard binding, dan sebagainya. Gue yakin elo bisa pahami pas liat konfignya
- default.nix ini isinya penggunaan flake sebagai projek utama pengganti legacy Nix
- flake.lock file ini tergenerate otomatis ketika kita jalanin *nix flake lock — update-input nixpkgs* yang gunanya buat nge lock dependensi
- flake.nix ini isinya konfigurasi flake kita, ada beberapa yang lo bisa ubah. Di config gue, lo cukup ubah soal environment intel/m1. Lo sesuaikan aja user yang dipakai, hostname dan network adapter.



```
# Current Macbook Pro M1
darwinConfigurations = rec {
  M1 = darwinSystem {
    system = "aarch64-darwin";
    modules = nixDarwinCommonModules ++ [
      {
        users.primaryUser = "petruknisme";
        networking.computerName = "Chernobyl";
        networking.hostName = "Chernobyl";
        networking.knownNetworkServices = [
          "Wi-Fi"
          "USB 10/100/1000 LAN"
        ];
      }
    ];
  };

  INT = darwinSystem {
    system = "x86_64-darwin";
    modules = nixDarwinCommonModules ++ [
      {
        users.primaryUser = "petruknisme";
        networking.computerName = "Chernobyl";
        networking.hostName = "Chernobyl";
        networking.knownNetworkServices = [
          "Wi-Fi"
          "USB 10/100/1000 LAN"
        ];
      }
    ];
  };
};
```

Default config flake.nix

Licensed under CC BY-SA 4.0

- folder home/ ini isinya konfig yang dipakai oleh **home-manager** (https://github.com/nix-community/home-manager), untuk konfig bisa referensi ke **https://nix-community.github.io/home-manager/index.html#sec-install-nix-darwin-module** (https://nix-community.github.io/home-manager/index.html#sec-install-nix-darwin-module). Beberapa config berasal dari **malob/nixpkgs** (https://github.com/malob/nixpkgs)

Build Nix

Setelah semua selesai dikonfigurasi, langkah selanjutnya kita build Nix nya dengan command:

```
nix build .#darwinConfigurations.[NAME].system
```

Untuk name, diganti dengan M1 atau INT sesuai dengan platform MacOS nya. Kalau pakai MacOS arm pilih M1, sedangkan kalau pakai MacOS intel pilih INT.

Apply Config

Setelah dibuild, kita perlu melakukan apply config. Ini berguna untuk menerapkan konfigurasi yang sudah kita buat dari file-file.nix sebelumnya.

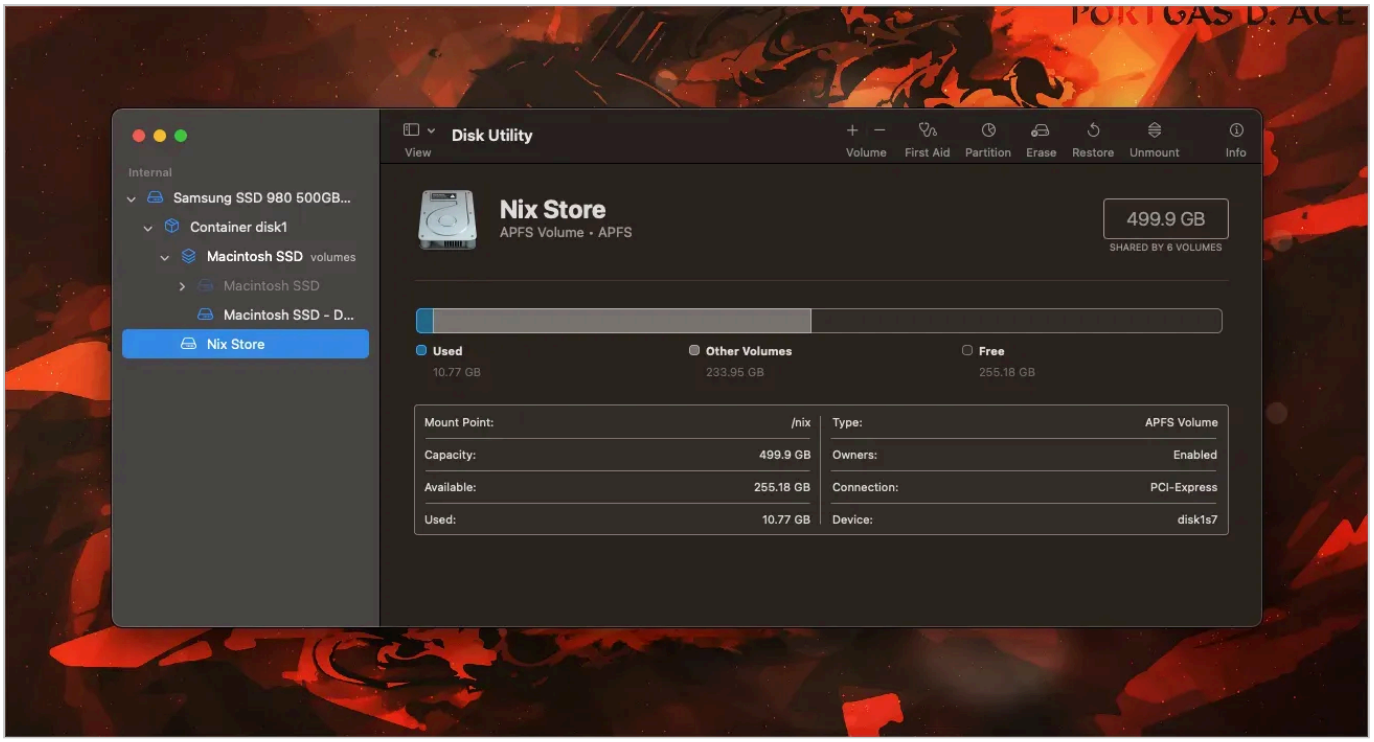
```
./result/sw/bin/darwin-rebuild switch --flake .#[NAME]
```

Ganti [NAME] dengan M1 atau INT tanpa [].

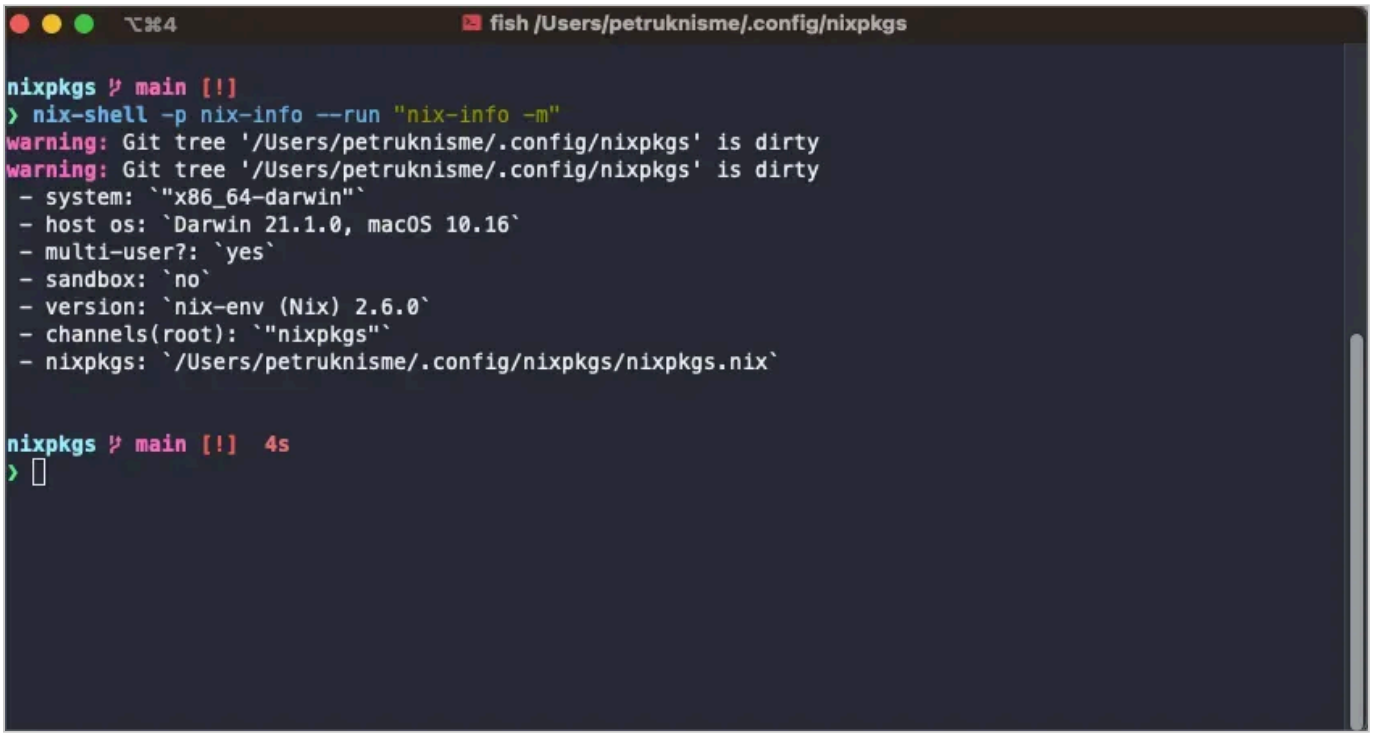
Verifikasi Instalasi dan setup

Kalau semua udah berhasil build, gue saranin untuk restart laptop. Untuk verifikasi nya bisa cek di Disk Utility, mestinya ada Nix-Store

Licensed under CC BY-SA 4.0



Nix Store Volume

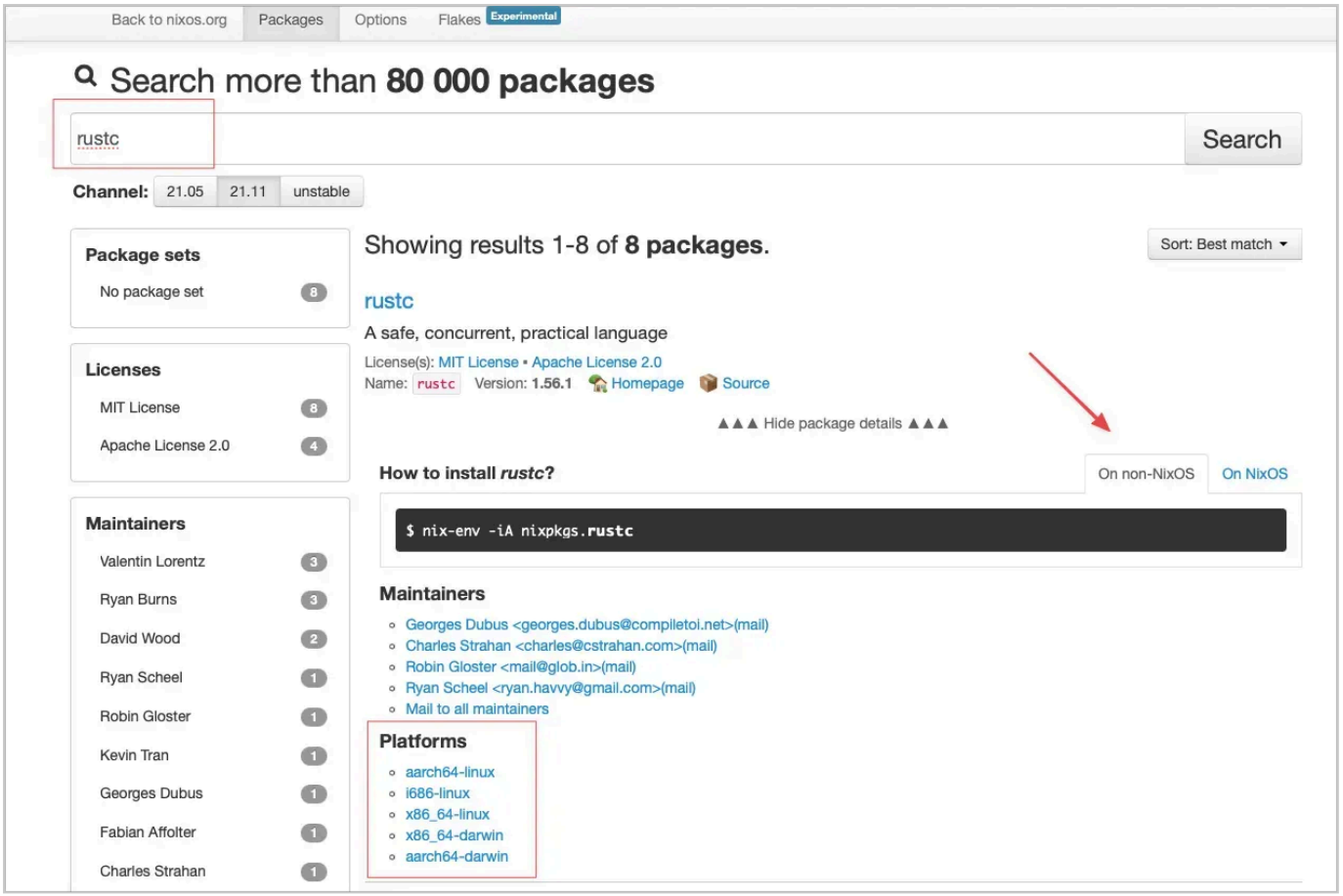


Nix udah berhasil diinstall beserta nixpkgs dan lain-lainnya.

Install Aplikasi

Kalau misal butuh app, tinggal tambahin nama app itu di packages.nix. Sebelumnya cari dulu nama app nya di search.nixos.org. Pastiin platform nya ada yang -darwin dan sesuaiin sama platform yang dimiliki.

Licensed under CC BY-SA 4.0

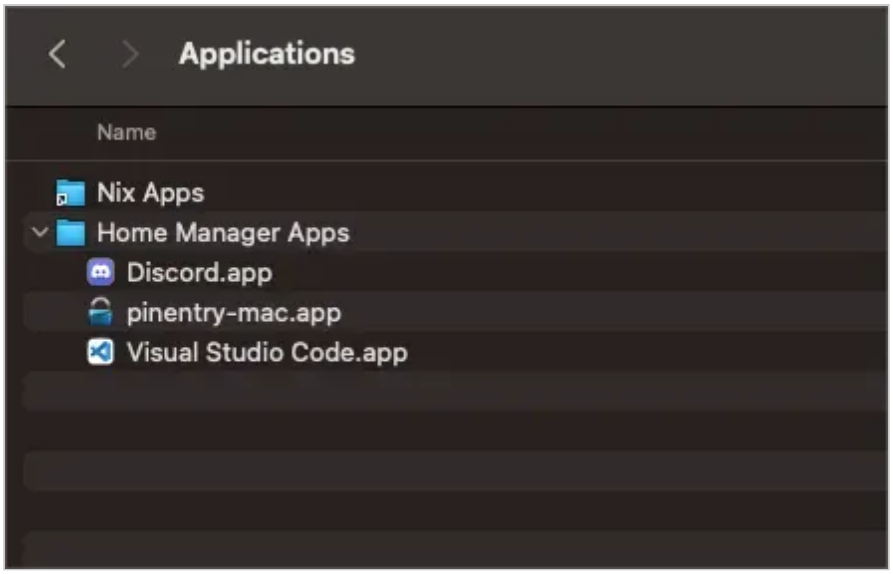


Abis itu ditambahin package nya di packages.nix, kita mesti re-apply config.

```
./result/sw/bin/darwin-rebuild switch --flake .#[NAME]
```

Tunggu sampe selesai, kalau ga ada error artinya berhasil install. Kalau itu app dari.dmg dia akan muncul di /Users/namauser/Applications/Home Manager Apps/

Licensed under CC BY-SA 4.0



Home Manager Apps

Dah, gitu aja. Moga berguna. Ntar gue update lagi kalau udah senggang ya

---

Originally published on **Medium** (https://medium.com/@petruknisme/nixos-instalasi-dan-setup-nixpkgs-di-macos-80bca52629b9).

NixOS # macOS # Tutorial

Licensed under CC BY-SA 4.0