Managing Your Userland With Home Manager

NixCon NA 2024

Your Speaker

About Marijan Petricevic - `https://marijan.pro`

👨‍💻 Informatics Degree, Software Engineer (Contractor)
Nix/NixOS user and contributer since 2019

💪 Interests/ Experience:
Software Architecture, Regression-free Development, Complexity Control, Haskell, C++

2023	NIXCADEMY EXPERT EDUCATION
2022	Platonic.Systems
2020	CYBERUS TECHNOLOGY
2017 - 2019	KIT Karlsruher Institut für Technologie FZI

Home Manager

Overview

2 What Is Home Manager?

3 Typical User Environment Configuration

4 The Home Manager Workflow

5 Installation

6 Tutorial

What Is Home Manager?

- Declarative configuration of user packages, services, and dotfiles
- Complementary to NixOS configurations
- Started by @rycee in 2017
- ~3000 configuration options for ~450 modules

Typical User Environment Configuration

- Imperative package installation
  - sudo apt-get install pkg1 pkg2 ... pkg300
  - Different distribution -> different name or version?
- Manual install of plugins/extensions/themes *per app*
- Create config files at specific locations
- Adapt config files to your needs/system type (server, laptop, ...)
- Manual service maintenance

Drawbacks or inconformities with the typical approach are ...

- A lot of manual work
- Lack of coupling
- Lack of config dependency management
- Managing the same env over different distros is hard ...
- Imperative changes can't be rolled back

Home Manager to the Rescue

1. Same on all distros/macOS
2. Pinning
3. Express dependencies between snippets in Nix lang
4. Package installation + config in one step
5. Rollback possible

The Home Manager Workflow

1. Edit your Home Manager configuration

2. Activate the new configuration with a single command

Activate The Configuration

```
$ home-manager switch
```

Installation

- From scratch with an initial config

Activate The Configuration

```
$ nix run home-manager -- init --switch
```

- With an existing config

Activate The Configuration

```
$ nix run home-manager -- switch \
  --flake github:applicative-systems/nixcon-home-manager \
  -b backup
```

Home Manager Configuration Example

Home Manager Configuration

```
{ pkgs, lib, config, ... }:
{
  home.packages = [ pkgs.ripgrep ];
  home.file.".config/git/ignore".text = ''
    result
    .DS_Store
  '';
  programs.git = {
    enable = true;
    userName = "NixCon";
    userEmail = "hello@nixcon.org";
  };
  programs.lazygit.enable = config.programs.git.enable;
}
```

Rollbacks

If we don't like our changes we can switch back to a previous generation:

Commands to inspect previous states and rollback

```
$ home-manager generations
2024-03-10 12:01 : id 8 ->
    ↪ /nix/store/d2...yd-home-manager-generation
2024-03-10 12:01 : id 7 ->
    ↪ /nix/store/j1...30-home-manager-generation
2024-03-10 11:55 : id 6 ->
    ↪ /nix/store/07...0w-home-manager-generation
2024-03-10 11:53 : id 5 ->
    ↪ /nix/store/jr...az-home-manager-generation
$ /nix/store/07..0w-home-manager-generation/activate
Starting home manager activation
. . .
```

Standalone Installation vs. NixOS/nix-darwin Module

- Standalone is per-user without root rights.

- Alternative: Home Manager user config can be integrated into NixOS/nix-darwin system config

NixOS Configuration

```
{
  imports = [ home-manager.nixosModules.home-manager ];
  users.users.nixcon.isNormalUser = true;
  home-manager.users.nixcon = { pkgs, ... }: {
    home.stateVersion = "23.11";
    programs.bash.enable = true;
  };
}
```

In the tutorial we are going to ...

- ...install Home Manager on an Ubuntu VM using Flakes from scratch
- ...create an example config

The final configuration can be found here:
`github.com/applicative-systems/nixcon-home-manager`

How to Find Configuration Options

- `https://nix-community.github.io/home-manager`
- man home-configuration.nix
- Configuration collection `https://nixos.wiki/wiki/Configuration_Collection`
- `https://home-manager-options.extranix.com`

Stay up2date with the Blog: `https://nixcademy.com`

NIXCADEMY
EXPERT EDUCATION