

# Ricochets

Hacking on <nixos> default channel for machine herd management, fun  
and non-profit

<https://functional.cafe/@yvan>

# The problem

Non-tech-savvy friends want to switch to Linux<sup>1</sup>, and they expect their computers to keep working for years<sup>2</sup>.

NixOS is great, **but there's no LTS**, and upgrading to the next stable version often requires writing a bit of Nix code.

---

<sup>1</sup>e.g., because of the EOL of Windows 10

<sup>2</sup>but I (hopefully) don't want to end up as their technical hotline

# The *magic* solution

**Use the default NixOS graphical installer<sup>3</sup>.** Then, open a terminal and paste the line that installs Ricochets<sup>4</sup>!

---

<sup>3</sup>choose GNOME or KDE, set your language/timezone/keymap, configure full-disk encryption, etc.

<sup>4</sup>it even includes a user-friendly TUI to help set up NixOS hardware customizations

# How does that work?

It keeps the `/etc/nixos/configuration.nix` that was generated but replaces the `<nixos>` channel<sup>5</sup> with a custom one<sup>6</sup>.

Please look up the sources at  
<https://git.sr.ht/~yvan-sraka/ricochets>

---

<sup>5</sup>I know Nix channels are a bit cursed

<sup>6</sup><https://ricochets.stoemp.club/main.tar.gz>

## How does it keep working?

I've set up some self-hosted<sup>7</sup> CI that auto-updates the channel but validates it against some example configs<sup>8</sup>.

The channel also serves as a placeholder to redefine options that could be deprecated in nixpkg<sup>9</sup>, e.g., `sound.enable` in `> 24.05`, `services.xserver.displayManager` in `> 25.05`, etc.<sup>10</sup>

---

<sup>7</sup>bye-bye GitHub Actions, hello IPv6

<sup>8</sup>using nixos-vm

<sup>9</sup>recall I don't want the user to have to edit their configuration.nix

<sup>10</sup>it's not a lot of backporting, not the full release notes, just the options that are used in the configuration.nix generated by the installer

# Can I do that?

The channel sets different defaults than nixpkgs: ApplImage, Flatpaks, envfs and nix-ld<sup>11</sup>.

So, we got some kind of super handy machine herd management, still without asking users<sup>12</sup> to add a module to their `configuration.nix`.

---

<sup>11</sup>and of course, automatic upgrade and garbage-collection

<sup>12</sup>or e.g., using some nix-manipulator-based automation

# Questions?

Feel free to ping me on Mastodon [@yvan@functional.cafe](https://functional.cafe/@yvan) ;)