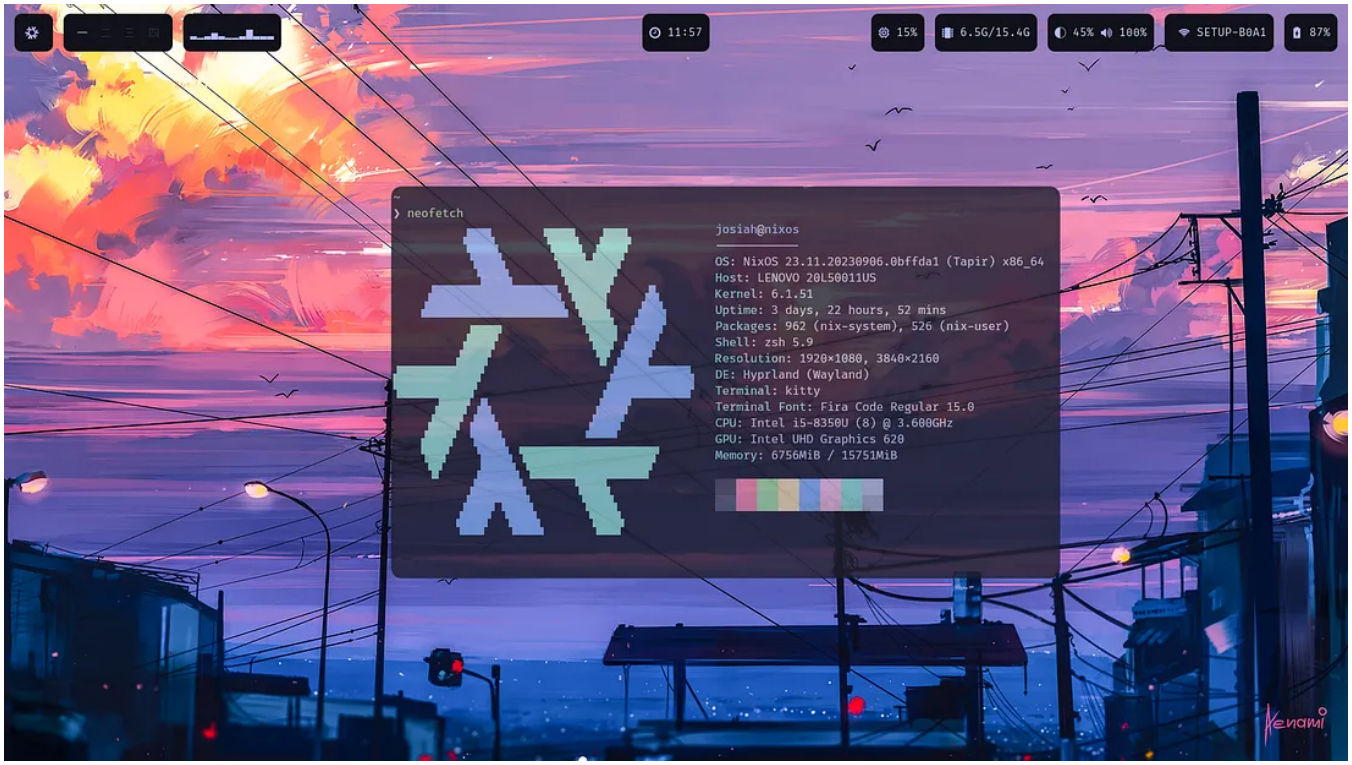
✨ Member-only story

Declaratively manage your Hyprland configuration with Home Manager

Josiah Brown · Follow

3 min read · Sep 15, 2023



My current desktop

If you read my last post about managing your dotfiles using Home-Manager, then you are probably curious about managing more than just some simple ZSH or Git

configurations. What if you could use it to generate your Hyprland configuration?

This article covers how to do that. However, you will need NixOS with Flakes enabled and Home Manager installed. If you don't have NixOS installed with Hyprland, follow this guide:

Installing NixOS with Hyprland!

Learn how to install NixOS with Hyprland, waybar, and other tools!

blog.stackademic.com

And this guide if you don't have Flakes enabled or Home Manager installed:

Managing your NixOS configuration with Flakes and Home Manager!

You can use the Nix language to manage packages and your dotfiles declaratively

medium.com

Now let's get started!

Editing your flake.nix

Unlike some of the other programs that Home Manager can work with natively, Hyprland requires a dedicated flake input. It will look something like this:

```
# flake.nix
{
  inputs = {
    # ...

    hyprland.url = "github:hyprwm/Hyprland";
  };
}
```

If you go to the GitHub repository for the Hyprland project, you will see in the root directory a flake.nix file of its own. That is what this input is referencing. You can see the same with the other inputs that you provide.

But an input without an output is a bit useless, so let's add it.

```
# flake.nix
{
  inputs = {
    # ...
  };
  outputs = {nixpkgs, home-manager, hyprland, ...}@inputs: {
    nixosConfiguration = {
      hostName = nixkpkgs.lib.nixosSystem {
        modules = [
          # ...
          hyprland.nixosModules.default
          { programs.hyprland.enable = true; }
          home-manager.nixosModules.home-manager
           {
            # ...
            home-manager.extraSpecialArgs = { inherit inputs; };
           }
        ];
      };
    };
  };
}
```

Ok, so let’s break this down. First, we give the outputs access to the Hyprland input from before. Then we define the NixOS configuration for the user. That's where we set Hyprland as a module and enable it.

Since we have Home Manager defined as a NixOS module, we place the basic configuration for that here. Inside that are the special arguments where we are inheriting the inputs, which contain the Hyprland input. This allows Home Manager to configure Hyprland.

Go ahead and rebuild your flake now. (`sudo nixos-rebuild switch --flake .#`)

Editing home.nix

The final step is adding the Hyprland dotfile configuration to your home.nix file. Since the Hyprland configuration can be lengthy, I highly recommend creating a new folder with a file called `default.nix` inside of it. But first, there is one thing you need to do in the root home.nix file.

```
# home.nix
{ inputs, config, pkgs, ... }:
{
  imports = [
    inputs.hyprland.homeManagerModules.default
    ./hypr # points to ./hypr/default.nix
  ];
}
```

In the `./hypr/default.nix` file (remember to use relative paths), add this:

```
# default.nix
{ config, inputs, pkgs, ... }:
{
  wayland.windowManager.hyprland = {
    enable = true;
    systemdIntegration = true;
    extraConfig = ''
      # copy your existing hyprland.conf file here
    '';
  };
}
```

In the `extraConfig` section, just copy your entire hyprland.conf file and paste it there. The only thing you might have to change if you have it enabled is the blur settings. It will look something like this:

```
extraConfig = ''
  # ...

  decortation {
    blur {
      enable = true
      # ...
    }
   # ...
  }
'';
```

Rebuild your flake again. I recommend restarting Hyprland (the default key bind for this is `SUPER + m`). When you log back in you should hopefully have no errors.

And that's it! The only downside to this method (if Home Manager is a NixOS module and not standalone), is that you will have to rebuild your entire system every time you want to save some changes to the Hyprland settings.

Here is the GitHub Gist for this article

And here are my Dotfiles where you can find how I manage my system.

Wayland Linux Dotfiles Nix

Follow

Written by Josiah Brown

359 Followers

Building nerdy stuff | Linux | AWS | Docker | Python | & everything else

More from Josiah Brown

Josiah Brown

Managing your NixOS configuration with Flakes and Home Manager!

You can use the Nix language to manage packages and your dotfiles declaratively

· 4 min read · Sep 8, 2023

4

 Josiah Brown in Stackademic

Installing NixOS with Hyprland!

Learn how to install NixOS with Hyprland, waybar, and other tools!

✨ · 7 min read · Aug 18, 2023

42

Josiah Brown

How to create a simple torrent stack with Docker compose

Good ole torrenting. If you're reading this you probably already understand what torrenting is. Usually seen as a bad thing by the...

4 min read · Nov 3, 2023

12

Josiah Brown

Using Obsidian to manage Code snippets

Organize and find your code snippets using Obsidian

4 min read · Aug 11, 2023

55 1

See all from Josiah Brown

Recommended from Medium

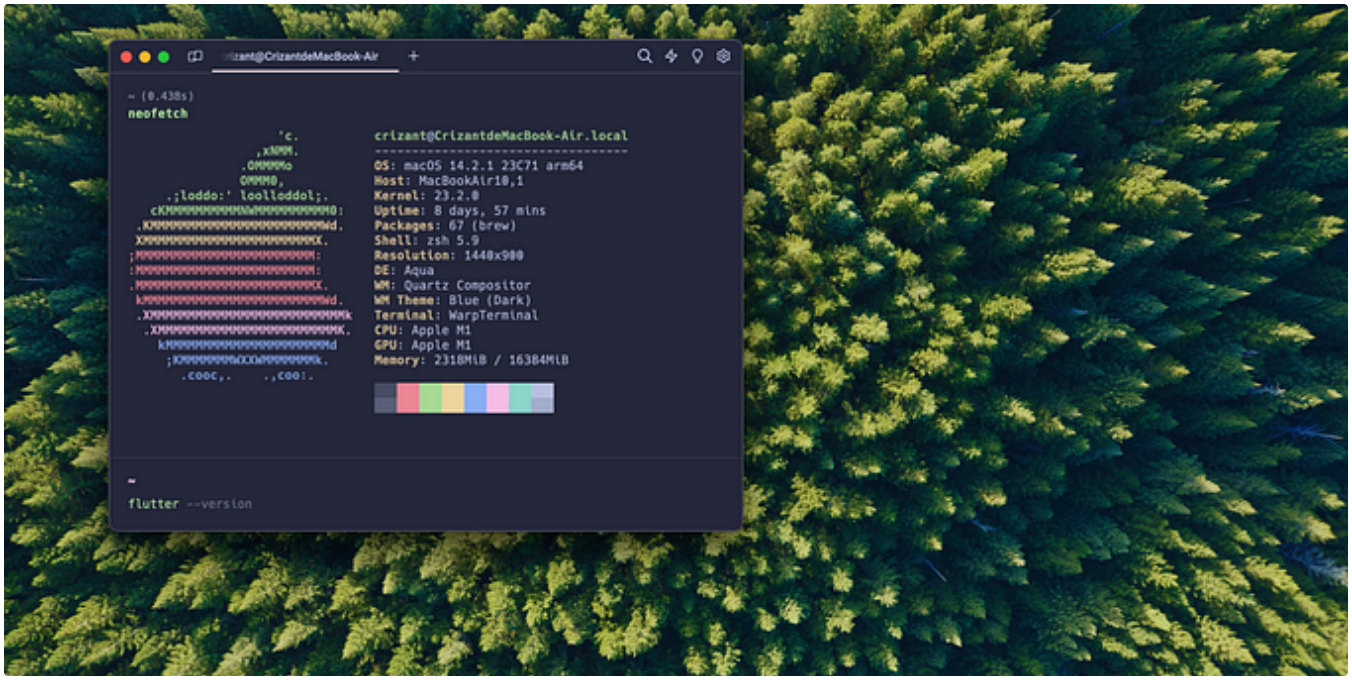
Michael Bao in Unixification

Is That Linux? No, It's SketchyBar and macOS

Stylising the macOS bar to be stunning and functional with SketchyBar.

✨ · 7 min read · Nov 28, 2023

125 1



Crizant Lai in CodeX

How I Setup My MacBook for Development (2024 Version)

Unlocking the Potential of the Cutting-Edge MacBook Setup in 2024.

· 5 min read · Dec 31, 2023

417 6

Lists

General Coding Knowledge

20 stories · 768 saves

Staff Picks

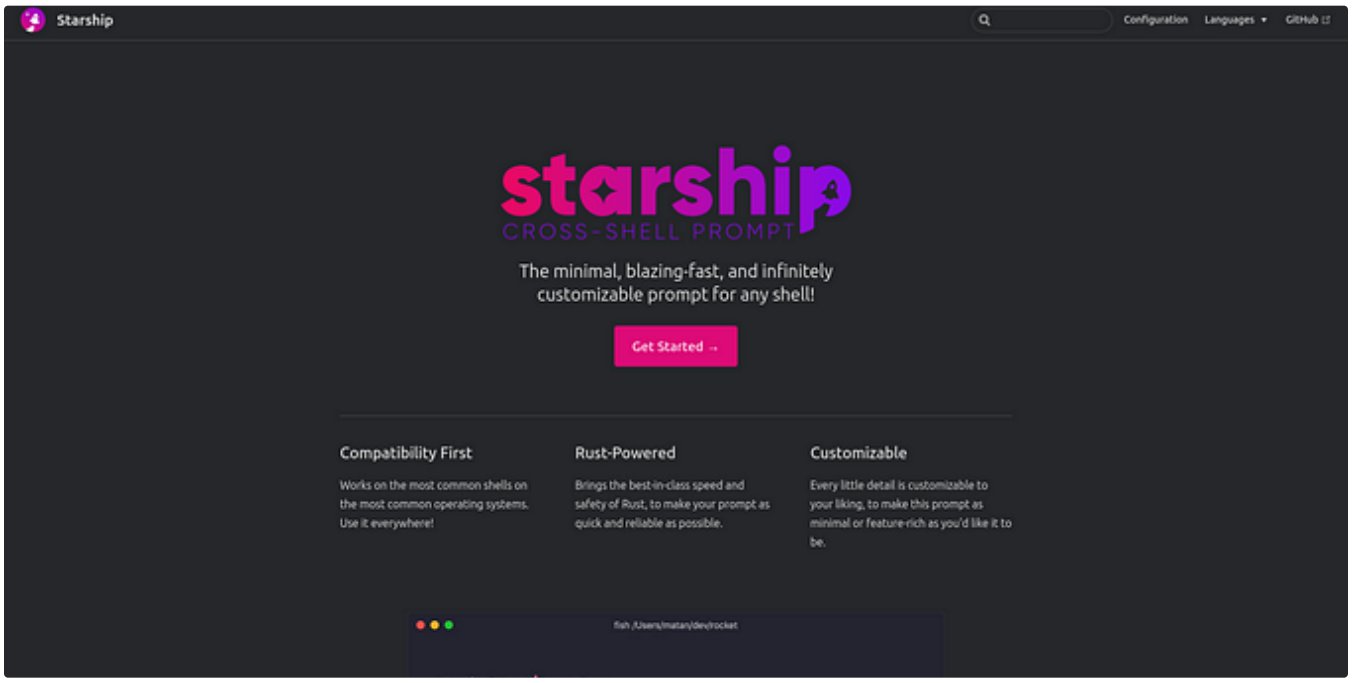
550 stories · 616 saves

Josiah Brown

Managing your NixOS configuration with Flakes and Home Manager!

You can use the Nix language to manage packages and your dotfiles declaratively

· 4 min read · Sep 8, 2023



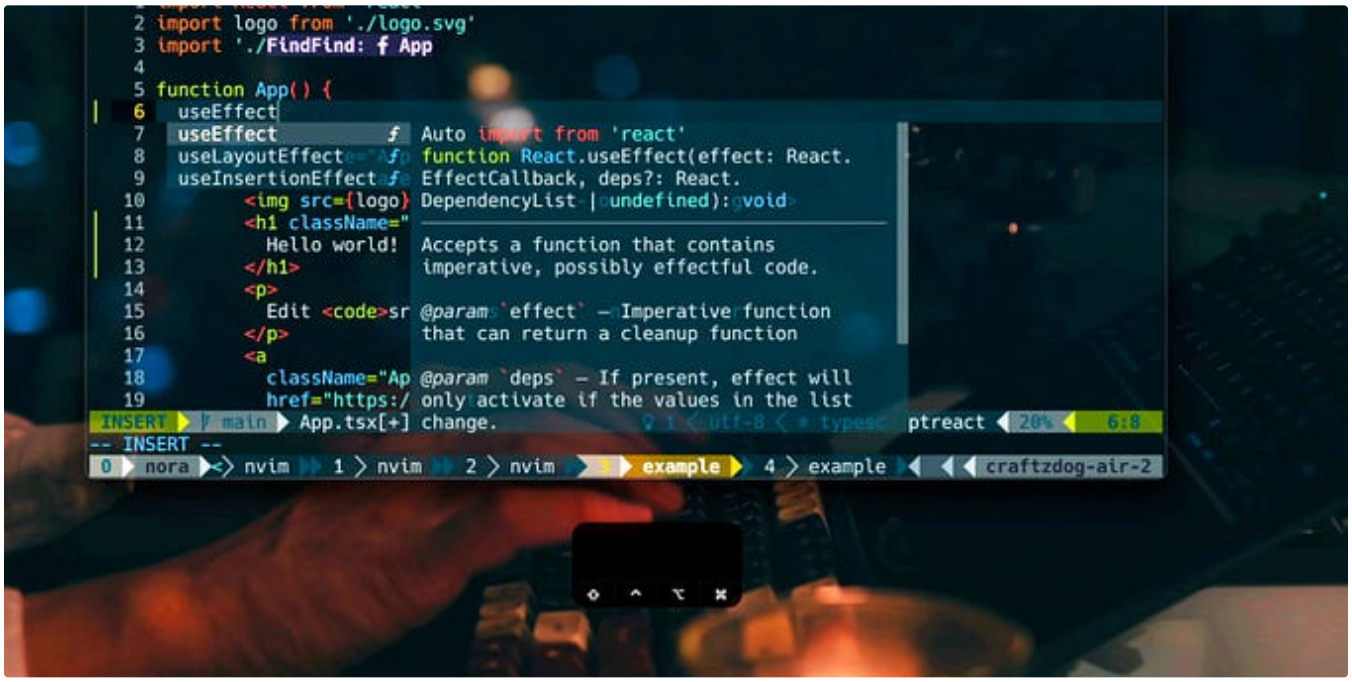
Rajdeep Singh in The Linux

What is a starship shell, and how is it configured?

Starship is a minimal, blazing-fast, customizable prompt shell built with rust language.

✨ · 5 min read · Sep 9, 2023

2



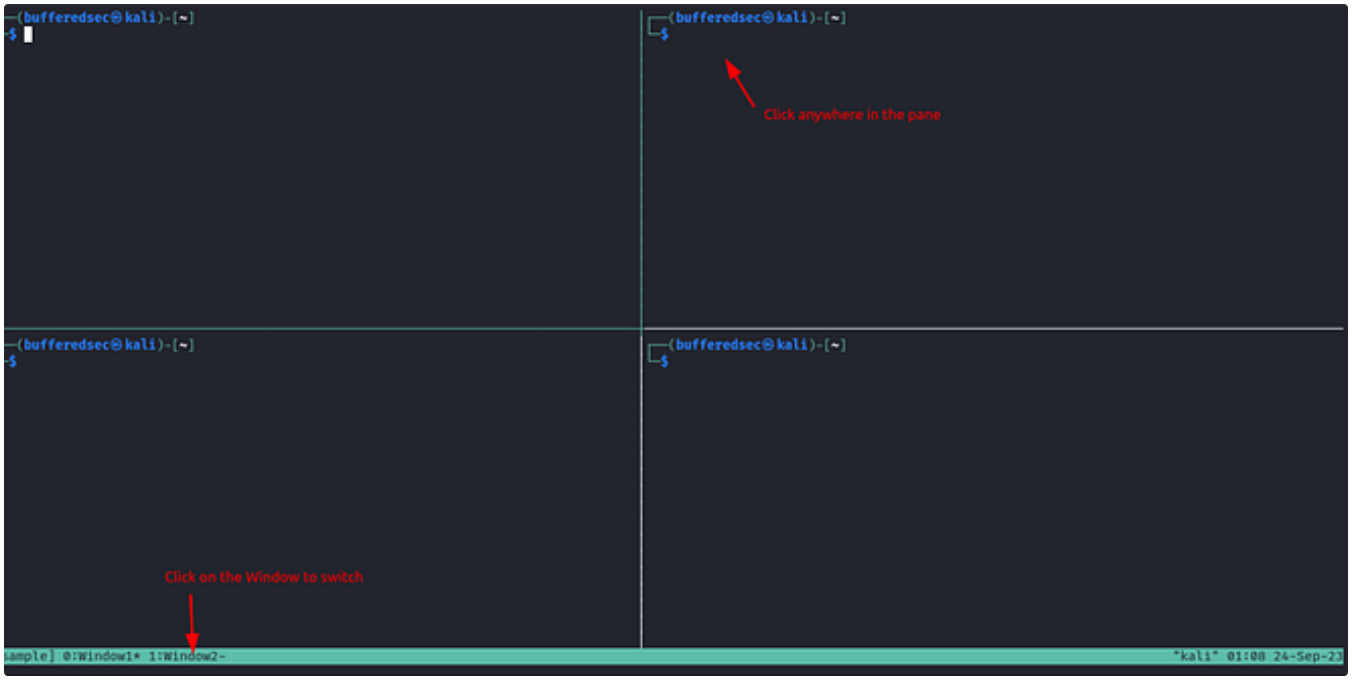
Kevin Feng

Overhauling my Neovim LSP and lazy.nvim configurations

Another Neovim blog post? I'm starting to understand why ThePrimeagen's GitHub states that he's located at the 9th Ring of Vim.

14 min read · Jul 26, 2023

66 1



Bhavik Nahar

Customize TMUX To Use It Effectively!

This article will go through customizing Tmux so you can use it effectively. It will go through detailed steps on how to change the...

6 min read · Sep 23, 2023

See more recommendations