jackdbd /
nix-config

Code Issues Pull requests Actions Projects Security Insigh

nix-config / nixos / hosts / x220 / **configuration.nix**

jackdbd add VSCodium 55f1b2e · 5 months ago

170 lines (145 loc) · 6.07 KB

Code Blame Raw

```
{
  allowed-unfree-packages,
  config,
  fh,
  inputs,
  lib,
  nil,
  nixos-hardware,
  permitted-insecure-pakages,
  pkgs,
  user,
  ...
}: {
  imports = [
    nixos-hardware.nixosModules.lenovo-thinkpad-x220
    ./hardware-configuration.nix
    {
      environment.systemPackages = [fh.packages.x86_64-linux.default];
    }
    {
      environment.systemPackages = [nil.packages.x86_64-linux.default];
    }
    ../../modules/android.nix
    ../../modules/bluetooth.nix
    ../../modules/dbt.nix
    ../../modules/fonts.nix
    ../../modules/nix.nix
    ../../modules/ollama.nix
    ../../modules/pipewire.nix
    ../../modules/printing.nix
    ../../modules/riscv.nix
    ../../modules/secrets.nix
    ../../modules/syncthing.nix
    ../../modules/tailscale.nix
    ../../modules/tarsnap.nix
    ../../modules/trezor.nix
    ../../modules/vscodium.nix
    ../../modules/vscodium.nix
    ../../modules/xserver.nix
  ];

  boot.initrd.luks.devices."luks-318ded24-f80a-41ea-96ec-c12aacb3f155".key
  boot.initrd.luks.devices."luks-8b9b15ff-cf4a-4e4a-8564-b577e7099437".key
  boot.initrd.luks.devices."luks-8b9b15ff-cf4a-4e4a-8564-b577e7099437".dev
  # Setup keyfile
  boot.initrd.secrets = {
    "/crypto_keyfile.bin" = null;
  };

  boot.loader.grub.device = "/dev/sda";
  boot.loader.grub.enable = true;
  boot.loader.grub.enableCryptodisk = true;
  boot.loader.grub.useOSProber = true;

  # Limit the number of NixOS generations to keep (and that show up in GRU
  # https://search.nixos.org/options?channel=23.11&show=boot.loader.grub.c
  boot.loader.grub.configurationLimit = 50;

  # Some Nix packages provide debugging symbols, but in general they don't
  # https://nixos.wiki/wiki/Debug_Symbols
  environment.enableDebugInfo = false;

  environment.homeBinInPath = true;

  # Pretty cool that NixOS allows to use different memory allocators from
  # one provided by libc. If you feel adventurous try scudo, a memory allo
  # based on LLVM Sanitizer's CombinedAllocator.
  # https://nixos.org/manual/nixos/stable/options#opt-environment.memoryAl
  # environment.memoryAllocator.provider = "scudo";

  # Environment variables set by Linux PAM
  # https://en.wikipedia.org/wiki/Linux_PAM
  # https://nixos.org/manual/nixos/stable/options#opt-environment.sessionV
  # environment.sessionVariables = {};

  environment.systemPackages = import ../../../lib/system-packages.nix {in

  # environment variables set at shell initialisation
  # https://nixos.org/manual/nixos/stable/options#opt-environment.variable
  # environment.variables = {};

  # introduced in NixOS 23.11
  # environment.xfce.excludePackages = with pkgs.xfce; [
  #   ristretto # image viewer.
  # ];

  hardware.bluetooth.enable = true;

  i18n.defaultLocale = "en_US.UTF-8";

  i18n.extraLocaleSettings = {
    LC_ADDRESS = "it_IT.UTF-8";
    LC_IDENTIFICATION = "it_IT.UTF-8";
    LC_MEASUREMENT = "it_IT.UTF-8";
    LC_MONETARY = "it_IT.UTF-8";
    LC_NAME = "it_IT.UTF-8";
    LC_NUMERIC = "it_IT.UTF-8";
    LC_PAPER = "it_IT.UTF-8";
    LC_TELEPHONE = "it_IT.UTF-8";
    LC_TIME = "it_IT.UTF-8";
  };

  networking.hostName = "x220-nixos";

  # To facilitate network configuration, we use NetworkManager.
  # https://nixpkgs-manual-sphinx-markedown-example.netlify.app/configurat
  networking.networkmanager.enable = true;

  # Instead of setting allowUnfree to true, I prefer explicitly list all t
  # unfree packages I am using.
  nixpkgs.config = {
    # allowUnfree = true;
    allowUnfreePredicate = pkg: builtins.elem (lib.getName pkg) allowed-ur
    permittedInsecurePackages = permitted-insecure-pakages;
  };

  # On XFCE, there is no configuration tool for NetworkManager by default:
  # enabling programs.nm-applet.enable, the graphical applet will be insta
  # and will launch automatically when the graphical session is started.
  programs.nm-applet.enable = true;

  services.pipewire.enable = true;
  services.printing.enable = true;
  services.syncthing.enable = true;
  services.tailscale.enable = true;
  services.tarsnap.enable = true;
  services.trezord.enable = true;
  services.xserver.enable = true;

  # Don't set sound.enable to true, as sound.enable is only meant for ALSA

  # I tried enabling automatic system upgrades and it broke my system. For
  # https://nixos.wiki/wiki/Automatic_system_upgrades
  # https://mynixos.com/nixpkgs/option/system.autoUpgrade.enable
  # https://www.reddit.com/r/NixOS/comments/yultt3/what_has_your_experienc
  system.autoUpgrade.enable = false;

  # This value determines the NixOS release from which the default
  # settings for stateful data, like file locations and database versions
  # on your system were taken. It‘s perfectly fine and recommended to leav
  # this value at the release version of the first install of this system.
  # Before changing this value read the documentation for this option
  # (e.g. man configuration.nix or on https://nixos.org/nixos/options.html
  # (e.g. man configuration.nix or on https://nixos.org/nixos/options.html
  # https://nixos.wiki/wiki/FAQ/When_do_I_update_stateVersion
  system.stateVersion = "23.05"; # Did you read the comment?

  time.timeZone = "Europe/Rome";

  # First, define extra groups. Then, declare users as members of those gr
  # https://superuser.com/a/1352988
  users.groups.skaters = {};
  users.users.${user} = import ../../users/jack.nix {inherit config pkgs;}

  virtualisation.docker = {
    enable = true;
    autoPrune = {
      enable = true;
      dates = "weekly";
    };

    # If you want to run the docker daemon in rootless mode, you need to s
    # either the socket path (using thr DOCKER_HOST environment variable)
    # CLI context using `docker context` explicitly.
    # https://docs.docker.com/engine/security/rootless/
    # https://docs.docker.com/engine/security/rootless/#client
    # rootless = {
    #   enable = true;
    #   setSocketVariable = true;
    # };
  };
}
```