mpickering / configuration.nix

Created 6 years ago • Report abuse

Code Revisions 1 Stars 1

configuration.nix

```
# Edit this configuration file to define what should be installed on
# your system.  Help is available in the configuration.nix(5) man page
# and in the NixOS manual (accessible by running ‘nixos-help’).

{ config, pkgs, ... }:

{



  environment.variables.EDITOR = "vim";

  imports =
    [ # Include the results of the hardware scan.
      ./hardware-configuration.nix
    ];

  # Use the systemd-boot EFI boot loader.
  boot.loader.systemd-boot.enable = true;
  boot.loader.efi.canTouchEfiVariables = true;

  boot.kernelPackages = pkgs.linuxPackages_4_19;

  nix.nixPath = [ "nixpkgs=/home/matt/nixpkgs" "nixos-config=/etc/nixos/configuration.n

  # networking.hostName = "nixos"; # Define your hostname.
  # networking.wireless.enable = true;  # Enables wireless support via wpa_supplicant.

  # Configure network proxy if necessary
  # networking.proxy.default = "http://user:password@proxy:port/";
  # networking.proxy.noProxy = "127.0.0.1,localhost,internal.domain";

  # Select internationalisation properties.
  # i18n = {
  #   consoleFont = "Lat2-Terminus16";
  #   consoleKeyMap = "us";
  #   defaultLocale = "en_US.UTF-8";
  # };

  # Set your time zone.
  # time.timeZone = "Europe/Amsterdam";

  # List packages installed in system profile. To search, run:
```

```
# $ nix search wget
environment.systemPackages = with pkgs;
    [
firefox
htop
networkmanagerapplet

      wget
neovim-with-packages
git
tmux
haskellPackages.hlint
      fzf
      haskellPackages.ghcid
      haskellPackages.fast-tags
      #haskell-ide
      cabal2nix

      haskell.packages.ghc863.ghc
      haskell.packages.ghc863.cabal-install

      ghc-head-from
   ];

networking.networkmanager.enable = true;
virtualisation.docker.enable = true;

services.xserver.enable = true;
services.xserver.desktopManager.plasma5.enable = true;


services.xserver.layout = "gb";
services.xserver.videoDrivers = ["intel"];

#services.xserver = {
#  windowManager.xmonad = {
#    enable = true;
#    enableContribAndExtras = true;
#    extraPackages = haskellPackages: [
#      haskellPackages.xmonad-contrib
#      haskellPackages.xmonad-extras
#      haskellPackages.xmonad
#    ];
#  };
#  enable = true;
#  layout = "gb";
#  autorun = false;
#};

# Some programs need SUID wrappers, can be configured further or are
# started in user sessions.
# programs.mtr.enable = true;
```

```
  # programs.gnupg.agent = { enable = true; enableSSHSupport = true; };

  # List services that you want to enable:

  # Enable the OpenSSH daemon.
  services.openssh.enable = true;

  # Open ports in the firewall.
  # networking.firewall.allowedTCPPorts = [ ... ];
  # networking.firewall.allowedUDPPorts = [ ... ];
  # Or disable the firewall altogether.
  # networking.firewall.enable = false;

  # Enable CUPS to print documents.
  # services.printing.enable = true;

  # Enable sound.
  sound.enable = true;
  hardware.pulseaudio.enable = true;

  # Enable the X11 windowing system.
  # services.xserver.enable = true;
  # services.xserver.layout = "us";
  # services.xserver.xkbOptions = "eurosign:e";

  # Enable touchpad support.
  # services.xserver.libinput.enable = true;

  # Enable the KDE Desktop Environment.
  # services.xserver.displayManager.sddm.enable = true;
  # services.xserver.desktopManager.plasma5.enable = true;

  nix.trustedUsers = [ "root" "matt" ];

  # Define a user account. Don't forget to set a password with ‘passwd’.
  users.extraUsers.matt = {
    createHome = true;
    extraGroups = ["wheel" "video" "audio" "disk" "networkmanager"];
    group = "users";
    home = "/home/matt";
    isNormalUser = true;
    uid = 1000;
  };

  nixpkgs.config.allowUnfree = true;
  nixpkgs.overlays = [ (import /home/matt/overlay.nix) ];

  # This value determines the NixOS release with which your system is to be
  # compatible, in order to avoid breaking some software such as database
  # servers. You should change this only after NixOS release notes say you
  # should.
  system.stateVersion = "18.09"; # Did you read the comment?
```

```

     nix.trustedBinaryCaches = [
  "http://hydra.cryp.to"
  "http://hydra.nixos.org"
  ];


     nix.binaryCaches = [
        "https://cache.nixos.org"
        "https://ocharles.cachix.org"
        "https://r-ryantm.cachix.org"
        "https://mpickering.cachix.org"
        "https://chessai.cachix.org"
        ];
     nix.binaryCachePublicKeys = [
        "ocharles.cachix.org-1:tZc7pKI8if0igUsr6QJD9GaM1pddllatd4W+8IQoH0I="
        "r-ryantm.cachix.org-1:gkUbLkouDAyvBdpBX0JOdIiD2/DP1ldF3Z3Y6Gqcc4c="
        "mpickering.cachix.org-1:COxPsDJqqrggZgvKG6JeH9baHPue8/pcpYkmcBPUbeg="
        "chessai.cachix.org-1:KWZ25BN1sJQ5PXT4bY/ye2KpaVXG2+tO8N76sxsp7YQ="
        ];
     nix.extraOptions = ''
        binary-caches-parallel-connections = 5
      '';

}
```