nickjanus /
nixos-config

Code Issues Pull requests Actions Projects Security Insigh

nixos-config / configuration.nix

Nick Janus updates to 23.11 df0d0b9 · last year

449 lines (404 loc) · 11.5 KB

Code Blame Raw

```
# 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’).

{ lib, config, pkgs, ... }:

let
  parameters = import ./parameters.nix;

  dbus-sway-environment = pkgs.writeTextFile {
    name = "dbus-sway-environment";
    destination = "/bin/dbus-sway-environment";
    executable = true;

    text = ''
      dbus-update-activation-environment --systemd WAYLAND_DISPLAY XDG_CUR
      systemctl --user stop pipewire pipewire-media-session xdg-desktop-po
      systemctl --user start pipewire pipewire-media-session xdg-desktop-p
    '';
  };

  configure-gtk = pkgs.writeTextFile {
    name = "configure-gtk";
    destination = "/bin/configure-gtk";
    executable = true;
    text = let
      schema = pkgs.gsettings-desktop-schemas;
      datadir = "${schema}/share/gsettings-schemas/${schema.name}";
    in ''
      export XDG_DATA_DIRS=${datadir}:$XDG_DATA_DIRS
      gnome_schema=org.gnome.desktop.interface
      gsettings set $gnome_schema gtk-theme 'Dracula'
      '';
  };

  security.rtkit.enable = true;
  baseServices = {
  baseServices = {
    dbus.enable = true;
    timesyncd.enable = true;
    pipewire = {
      enable = true;
      alsa.enable = true;
      alsa.support32Bit = true;
      pulse.enable = true;
    };
  };

  basePackages = with pkgs; [
    ack
    bat
    bc
    bind
    binutils
    calibre
    google-chrome
    cmus
    direnv
    dpkg
    efibootmgr
    efivar
    eza
    file
    firefox
    fwupd
    fzf
    git
    glances # system pane of glass
    gnumake
    gnupg
    gnupg1compat
    gptfdisk
    gtop
    htop
    imagemagick7
    jq
    libnotify
    lsof
    neovim
    nfs-utils
    nix-index
    openssh
    parted
    patchelf
    pavucontrol
    s3cmd
    screen
    silver-searcher
    spotify
    sysstat
    teensy-loader-cli
    terraform
    tldr # man page-like examples
    tmux
    tree
    unzip
    vlc
    wget
    xxd
    yq
    zoxide
    (
      pkgs.writeTextFile {
        name = "startsway";
        destination = "/bin/startsway";
        executable = true;
        text = ''
          #! /usr/bin/env bash

          # first import environment variables from the login manager
          systemctl --user import-environment DISPLAY WAYLAND_DISPLAY PATH

          # then start the service
          exec systemctl --user start sway.service
        '';
      }
    )
  ];

in {
  imports =
    [ # Include the results of the hardware scan.
      (./hardware-configurations + "/${parameters.machine}.nix")
      # Machine specific config
      (
        import (./machines + "/${parameters.machine}.nix") {
          inherit lib;
          inherit config;
          inherit pkgs;
          inherit baseServices;
          inherit basePackages;
          inherit parameters;
        }
      )
    ];


  fonts = {
    fontDir.enable = true;
    enableGhostscriptFonts = true;
    packages = with pkgs; [
      corefonts  # Micrsoft free fonts
      fira # monospaced
      fira # monospaced
      inconsolata  # monospaced
      open-dyslexic
      powerline-fonts
      ubuntu_font_family
      unifont # some international languages
    ];
  };

  # Select internationalisation properties.
  console = {
    font = "Lat2-Terminus16";
    keyMap = "us";
  };
  i18n = {
    defaultLocale = "en_US.UTF-8";
  };

  # Set your time zone.
  time.timeZone = "America/New_York";

  nixpkgs.config = {
    allowUnfree = true;
    permittedInsecurePackages = [
   ];
  };

  environment = {
    etc = {
      "sway/config".source = import ./sway.nix { inherit config; inherit p
      "fish/functions/fish_user_key_bindings.fish".source = ./fish_functio
      "environment".source = ./etc-environment;
    };
  };

  # List services that you want to enable:
  programs = {
    corectrl.enable = true;
    ssh.forwardX11 = false;
    ssh.startAgent = true;
    fish = {
      enable = true;
      shellInit = ''
        set EDITOR ${pkgs.neovim}/bin/nvim
        set VISUAL ${pkgs.neovim}/bin/nvim
        set PAGER "${pkgs.less}/bin/less -R"

        fish_vi_key_bindings
        direnv hook fish | source
        zoxide init fish | source
      '';
      interactiveShellInit = ''
        startsway
      '';
      shellAbbrs = {
        ergodox-update = "sudo teensy-loader-cli --mcu=atmega32u4 -v -w";
        grab = "grim -g (slurp)";
        xprimary = "xrandr --output (xrandr --listactivemonitors | grep 25
        ls = "eza --group-directories-first --color=auto";
      };
    };
    sway = {
      enable = true;
      extraPackages = with pkgs; [
        bemenu
        kanshi # autorandr replacement
        kitty
        gnome3.adwaita-icon-theme # icons for various applications
        grim # screen cap
        i3status
        light
        mako
        rxvt-unicode
        slurp # for screenshot selection
        swaylock
        swayidle
        wdisplays # arandr equivalent
        wl-clipboard
        xwayland
      ];
      wrapperFeatures.gtk = true;
    };
    neovim = {
      enable = true;
      viAlias = true;
      vimAlias = true;
      configure = {
        customRC = ''
          syntax enable

          """ My settings:
          " tab with two spaces
          set nobackup
          set noswapfile
          set tabstop=2
          set shiftwidth=2
          set softtabstop=2
          set expandtab
          set smartcase
          set autoindent
          set nostartofline
          set hlsearch      " highlight search terms
          set incsearch     " show search matches as you type

          set mouse=a
```

```
set cmdheight=2

set wildmenu
set showcmd

set number
set cursorline
set ruler
set backspace=indent,eol,start " Allows backspace on these chara
set clipboard=unnamedplus

set spell

" Folding
set foldmethod=syntax

""" Keep all folds open when a file is opened
augroup OpenAllFoldsOnFileOpen
    autocmd!
    autocmd BufRead * normal zR
augroup END

""" Don't open folds with block motions
set foldopen-=block

" Map esc to exit terminal mode
:tnoremap <Esc> <C-\><C-n>

" php
autocmd Filetype php setlocal tabstop=4 shiftwidth=4 softtabstop

" ruby
autocmd FileType ruby compiler ruby

" go
autocmd BufNewFile,BufRead *.go setlocal noexpandtab tabstop=4 s

" yaml
autocmd FileType yaml setlocal ts=2 sts=2 sw=2 expandtab

filetype plugin on    " Enable filetype-specific plugins

" indentLine
let g:indentLine_char = '⁞'

" neosolarized
let g:neosolarized_contrast = "normal"
set background=dark
colorscheme solarized

" Those types
if has("user_commands")
  command! -bang -nargs=? -complete=file E e<bang> <args>
```

```
          command! -bang -nargs=? -complete=file W w<bang> <args>
          command! -bang -nargs=? -complete=file Wq wq<bang> <args>
          command! -bang -nargs=? -complete=file WQ wq<bang> <args>
          command! -bang Wa wa<bang>
          command! -bang WA wa<bang>
          command! -bang Q q<bang>
          command! -bang QA qa<bang>
          command! -bang Qa qa<bang>
        endif

        " Relative numbering
        function! NumberToggle()
          if(&relativenumber == 1)
            set nornu
            set number
          else
            set rnu
          endif
        endfunc

        " Toggle between normal and relative numbering.
        nnoremap <leader>r :call NumberToggle()<cr>

        " gitgutter settings
        let g:gitgutter_max_signs = 2000

        " fzf settings
        nnoremap <silent> <leader>f :FZF<cr>
        nnoremap <silent> <leader>F :Lines<cr>
        let g:fzf_colors =
        \ { 'fg':      ['fg', 'Normal'],
        \ 'bg':      ['bg', 'Normal'],
        \ 'hl':      ['fg', 'Comment'],
        \ 'fg+':     ['fg', 'CursorLine', 'CursorColumn', 'Normal'],
        \ 'bg+':     ['bg', 'CursorLine', 'CursorColumn'],
        \ 'hl+':     ['fg', 'Statement'],
        \ 'info':    ['fg', 'PreProc'],
        \ 'prompt':  ['fg', 'Conditional'],
        \ 'pointer': ['fg', 'Exception'],
        \ 'marker':  ['fg', 'Keyword'],
        \ 'spinner': ['fg', 'Label'],
        \ 'header':  ['fg', 'Comment'] }

        " Clean up artifacts in neovim, see https://github.com/neovim/ne
        let $VTE_VERSION="100"
      '';
      packages.myVimPackage = with pkgs.vimPlugins; {
        start = [
          ale
          fugitive
          fzf-vim
          fzfWrapper
```

```
            indentLine
            NeoSolarized
            solarized
            tabular
            vim-fish
            vim-gh-line
            vim-gitgutter
            vim-json
            vim-jinja
            vim-nix
            vim-go
            zoxide-vim
          ];
        };

      };
    };
  };

  # enable screensharing in sway
  xdg.portal = {
    enable = true;
    wlr.enable = true;
    # gtk portal needed to make gtk apps happy
    extraPortals = [ pkgs.xdg-desktop-portal-gtk ];
  };

  systemd = {
    user = {
      targets = {
        sway-session = {
          description = "Sway compositor session";
          documentation = [ "man:systemd.special(7)" ];
          bindsTo = [ "graphical-session.target" ];
          wants = [ "graphical-session-pre.target" ];
          after = [ "graphical-session-pre.target" ];
        };
      };
      services = {
        sway = {
          description = "Sway - Wayland window manager";
          documentation = [ "man:sway(5)" ];
          bindsTo = [ "graphical-session.target" ];
          wants = [ "graphical-session-pre.target" ];
          after = [ "graphical-session-pre.target" ];
          # We explicitly unset PATH here, as we want it to be set by
          # systemctl --user import-environment in startsway
          environment.PATH = lib.mkForce null;
          serviceConfig = {
            Type = "simple";
            ExecStart = ''
              ${pkgs.sway}/bin/sway
            '';
            '';
            Restart = "on-failure";
            RestartSec = 1;
            TimeoutStopSec = 10;
          };
        };
        kanshi = {
          description = "Kanshi output autoconfig ";
          wantedBy = [ "graphical-session.target" ];
          partOf = [ "graphical-session.target" ];
          serviceConfig = {
            ExecStart = ''
              ${pkgs.kanshi}/bin/kanshi
            '';
            RestartSec = 5;
            Restart = "always";
          };
        };
      };
    };
  };

  users.extraUsers.nick = {
    home = "/home/nick";
    description = "Nick Janus";
    extraGroups = [
      "corectrl"
      "docker"
      "lp"
      "networkmanager"
      "scanner"
      "video"
      "wheel"
    ];
    isNormalUser = true;
    openssh.authorizedKeys.keys = [
      "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFYFRUEBS2XeaW6sgNhbguZ3500VhdD
      "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDArVbqOscUP0BaduqIRpUdOGYkByV7
      "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGFOJR+pJwtAch8btM5dVsbybQrH/lY
    ];
    uid = 1000;
  };
  users.defaultUserShell = "/run/current-system/sw/bin/fish";

  virtualisation.docker.enable = true;
}
```