

lontivero /
my-nixos-configuration

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

my-nixos-configuration / configuration.nix



lontivero virtualization

456c4de · 6 months ago



405 lines (354 loc) · 10.3 KB

```
1  # Edit this configuration file to define what should be installed on
2  # your system. Help is available in the configuration.nix(5) man page
3  # and in the NixOS manual (accessible by running 'nixos-help').
4
```

my-nixos-configuration / configuration.nix

↑ Top

Code

Blame



Raw



```
9      ./hardware-configuration.nix
10     inputs.home-manager.nixosModules.home-manager
11     ./ssh.nix
12     ./tmux.nix
13     # ./picom.nix
14     ./networking.nix
15 ];
16
17 nixpkgs.config.allowUnfree = true;
18
19 # Manage the virtualisation services
20 virtualisation = {
21     libvirtd = {
22         enable = true;
23         qemu = {
24             swtpm.enable = true;
25             ovmf.enable = true;
26             ovmf.packages = [ pkgs.OVMFFull.fd ];
27         };
28     };
29     spiceUSBRedirection.enable = true;
30 };
31 services.spice-vdagentd.enable = true;
32
33 # Use the systemd-boot EFI boot loader.
34 boot.loader.systemd-boot.enable = true;
35 boot.loader.efi.canTouchEfiVariables = true;
36
37 # Plymouth boot splash screen
```

```
38 boot.plymouth.enable = true;
39
40 # For nvidia test
41 # boot.kernelPackages = pkgs.linuxPackages_latest;
42
43 # Set your time zone.
44 time.timeZone = "America/Argentina/Buenos_Aires";
45
46 # Select internationalisation properties.
47 # i18n.defaultLocale = "en_US.UTF-8";
48 # console = {
49 #   font = "Lat2-Terminus16";
50 #   keyMap = "us";
51 # };
52
53 # Enable the X11 windowing system.
54 services.xserver.enable = true;
55
56 hardware.opengl.enable = true;
57
58 # do not install what I dont want
59 # services.gnome.core-utilities.enable = false;
60
61 # Enable the GNOME Desktop Environment.
62 services.displayManager.defaultSession = "none+i3";
63 services.xserver.desktopManager.xterm.enable = false;
64
65 documentation.man.generateCaches = true;
66
67 environment.pathsToLink = [ "/libexec" "/share/fish" ];
68 services.xserver.windowManager.i3 = {
69   enable = true;
70   package = pkgs.i3-gaps;
71   extraPackages = with pkgs; [
72     dmenu
73     i3status
74     i3lock
75     i3blocks
76   ];
77   configFile = "/etc/i3.conf";
78 };
79
80 environment.etc."i3.conf".text = pkgs.callPackage ./i3-config.nix {};
81
82 # Do not suspend when close the laptop
83 services.logind.lidSwitch = "ignore";
84
85 # Configure keymap in X11
86 services.xserver.xkb.layout = "latam,us";
87 services.xserver.xkb.options = "eurosign:e, compose:menu, grp:alt_space_
88
89 # Enable CUPS to print documents.
```

```
90     # services.printing.enable = true;
91
92     # Enable sound.
93     sound.enable = true;
94     hardware.pulseaudio.enable = true;
95
96     # Enable touchpad support (enabled default in most desktopManager).
97     # services.xserver.libinput.enable = true;
98
99     # Define a user account. Don't forget to set a password with 'passwd'.
100    # users.mutableUsers = false;
101    users.defaultUserShell = pkgs.fish;
102    users.users.lontivero = {
103        isNormalUser = true;
104        # hashedPassword = "$6$NDuMuWF2P3Z3aDSl$cai6jw.X8jvmaSHxRqbbzEtEVW7T/
105        extraGroups = [ "wheel" "networkmanager" "libvirt" ]; # Enable 'sudo'
106    };
107
108    # needed for vscode
109    services.gnome.gnome-keyring.enable = true;
110    home-manager.useGlobalPkgs = true;
111    home-manager.users.lontivero = { pkgs, ... }: {
112        home.stateVersion = "22.05";
113        gtk = {
114            enable = true;
115            theme = {
116                name = "Nordic";
117                package = pkgs.nordic; # gnome3.gnome_themes_standard;
118                # name = "Adwaita";
119            };
120        };
121        programs = {
122            man.generateCaches = true;
123            #bat = {
124            #    enable = true;
125            #};
126            fish = {
127                enable = true;
128                shellInit = ''
129                    set fish_color_autosuggestion brblack
130                    if status is-interactive
131                        and not set -q TMUX
132                            exec tmux
133                        end
134                '';
135                shellAliases = {
136                    gdiff = "git diff";
137                    gl = "git prettylog";
138                    gs = "git status";
139                    cat = "bat -p";
140                    grep = "rg";
141                    mkdir = "mkdir -p";
```

```

142     };
143     shellAbbrs = {
144         n = "nvim";
145     };
146     functions = {
147         mkdcd = {
148             description = "Make a directory and enter it";
149             body = "mkdir -p $argv[1]; and cd $argv[1]";
150         };
151         clip = {
152             description = "xclip selection";
153             body = "xclip -selection c $argv";
154         };
155         qr = {
156             description = "Encode data as QR code";
157             body = "qrencode -o - | feh -";
158         };
159     };
160 };
161 direnv = {
162     enable = true;
163 };
164 git = {
165     enable = true;
166     package = pkgs.gitAndTools.gitFull;
167     userName = "Lucas Ontivero";
168     userEmail = "lucasontivero@gmail.com";
169     aliases = {
170         remotes = "remote -v";
171         prettylog = "log --graph --pretty=format:'%Cred%h%Creset -%C(yel";
172         root = "rev-parse --show-toplevel";
173         fixup = "!git log -n 10 --pretty=format:'%h %s' --no-merges | fz";
174     };
175     extraConfig = {
176         # branch.autosetuprebase = "always";
177         color.ui = true;
178         core.askPass = ""; # needs to be empty to use terminal for ask p
179         credential.helper = "store"; # want to make this more secure
180         github.user = "lontivero";
181         init.defaultBranch = "master";
182         rebase = {
183             autosquash = true;
184         };
185         pull = {
186             ff = "only";
187         };
188         sendemail = {
189             smtpserver = "smtp.gmail.com";
190             smtpuser = "lucasontivero";
191             smtpencryption = "tls";
192             smtpserverport = 587;
193         };
194     };

```

```
195     };
196     alacrity = {
197         enable = true;
198
199         settings = {
200             env.TERM = "xterm-256color";
201         };
202     };
203     i3status = {
204         enable = true;
205
206         general = {
207             colors = true;
208             color_good = "#8C9440";
209             color_bad = "#A54242";
210             color_degraded = "#DE935F";
211         };
212
213         modules = {
214             ipv6.enable = false;
215             "wireless _first_".enable = false;
216             "battery all".enable = false;
217         };
218     };
219     neovim = {
220         enable = true;
221         viAlias = true;
222         vimAlias = true;
223         withNodeJs = true;
224
225         plugins = with pkgs; [
226             vimPlugins.vim-cue
227             vimPlugins.vim-fugitive
228             vimPlugins.which-key-nvim
229             vimPlugins.nvim-whichkey-setup-lua
230
231             vimPlugins.nvim-comment
232             vimPlugins.vim-misc
233             vimPlugins.telescope-nvim
234
235             vimPlugins.nvim-treesitter
236             vimPlugins.nvim-treesitter-textobjects
237
238             # vimPlugins.coc-nvim
239             vimPlugins.nvim-lspconfig
240             vimPlugins.vim-fish
241             vimPlugins.vim-airline
242             vimPlugins.vim-airline-themes
243             vimPlugins.vim-gitgutter
244
245             vimPlugins.vim-markdown
246             vimPlugins.vim-nix
```

```
247     vimPlugins.vimwiki
248 ];
249
250     # extraConfig = (import ./vim-config.nix) { inherit sources; };
251     extraConfig = pkgs.callPackage ./vim-config.nix {};
252 vscode = {
253     enable = true;
254     extensions = with pkgs.vscode-extensions; [
255         # Some example extensions...
256         dracula-theme.theme-dracula
257         vscodevim.vim
258         yzhang.markdown-all-in-one
259     ];
260 };
261 };
262 };
263
264 # List packages installed in system profile. To search, run:
265 # $ nix search wget
266
267 environment.systemPackages = with pkgs; [
268     neovim # Do not forget to add an editor to edit configuration.nix! The
269     curl
270     wget
271     firefox
272     htop
273     dmenu
274     xclip
275     bc
276     qrencode
277     ffmpeg
278     mplayer
279     rxvt_unicode
280     alacritty
281
282     git
283     git-crypt
284     lazygit
285
286     unzip
287     feh
288     tmux
289     fzf
290     jq
291     ripgrep
292     weechat
293     rofi
294     nitrogen
295     ranger
296     tree
297     watch
298     bat
299     file
```

```
...
300 killall
301 patchelf
302 direnv
303
304 dunst
305 viewnior
306 mpd
307 mpc_cli
308 acpi
309 brightnessctl
310 scrot
311 libnotify
312
313 lxappearance
314 networkmanagerapplet
315
316 mc
317 sshfs
318 graphviz
319 pandoc
320 gnuplot
321
322 virt-manager
323 spice spice-gtk
324 spice-protocol
325 win-virtio
326 win-spice
327 ];
328
329 # Solves problem for binaries that cannot find the interpreter
330 system.activationScripts.ldso = lib.stringAfter [ "usrbinenv" ] ''
331     mkdir -m 0755 -p /lib64
332     ln -sf ${pkgs.glibc.out}/lib64/ld-linux-x86-64.so.2 /lib64/ld-linux->
333     mv -f /lib64/ld-linux-x86-64.so.2.tmp /lib64/ld-linux-x86-64.so.2 # at
334     '';
335
336 environment.variables = {
337     EDITOR = "nvim";
338     TERMINAL = "alacritty";
339     BROWSER = "firefox";
340     DOTNET_ROOT = "${pkgs.dotnet-sdk_7}";
341 };
342
343
344 programs = {
345     dconf = {
346         enable = true;
347     };
348     fish = {
349         enable = true;
350     };
351 };
```

```
352
353     fonts = {
354         fontDir.enable = true;
355         enableGhostscriptFonts = true;
356         packages = with pkgs; [
357             powerline-fonts
358             ubuntu_font_family
359             liberation_ttf
360             (nerdfonts.override { fonts = [ "FiraCode" "Monoid" ]; })
361         ];
362     };
363
364     # Some programs need SUID wrappers, can be configured further or are
365     # started in user sessions.
366     # programs.mtr.enable = true;
367     programs.gnupg.agent = {
368         enable = true;
369         enableSSHSuport = true;
370         # pinentryFlavor = "tty";
371     };
372
373     nix = {
374         package = pkgs.nixFlakes;
375         settings.auto-optimise-store = true;
376         gc = {
377             automatic = true;
378             options = "--delete-older-than 30d";
379         };
380         extraOptions = ''
381             keep-outputs = true
382             experimental-features = nix-command flakes
383         '';
384     };
385
386     # List services that you want to enable:
387
388     # Enable the OpenSSH daemon.
389     # services.openssh.enable = true;
390
391     # Open ports in the firewall.
392     # networking.firewall.allowedTCPPorts = [ ... ];
393     # networking.firewall.allowedUDPPorts = [ ... ];
394     # Or disable the firewall altogether.
395     # networking.firewall.enable = false;
396
397     # This value determines the NixOS release from which the default
398     # settings for stateful data, like file locations and database versions
399     # on your system were taken. It's perfectly fine and recommended to lea
400     # this value at the release version of the first install of this system.
401     # Before changing this value read the documentation for this option
402     # (e.g. man configuration.nix or on https://nixos.org/nixos/options.html
403     system.stateVersion = "21.11"; # Did you read the comment?
```


