

mudrii /  
nixlenovodot

&lt;&gt; Code

Issues

Pull requests

Actions

Projects

Security

Insights

## nixlenovodot / configuration.nix



mudrii small change in config

6d790a8 · 3 years ago



1409 lines (1308 loc) · 37.2 KB

Code

Blame



Raw



```
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
5  { inputs, config, options, pkgs, lib, callPackage, ... }:
6
7  let
8      unstable = import <unstable> {
9          config.allowUnfree = true;
10         /*
11             overlays = [
12                 (_: prev: {
13                     linuxPackagesFor = kernel:
14                         (prev.linuxPackagesFor kernel).extend (_: _: { ati_drivers_x11 = nul
15                     });
16             ];
17         */
18     };
19  in
20
21  {
22
23      imports =
24          [
25              # Include the results of the hardware scan.
26              ./hardware-configuration.nix
27              <home-manager/nixos>
28              # "${builtins.fetchTarball https://github.com/rycee/home-manager/arc
29              # ./home-manager-nelu.nix
30              ./home.nix
31              # ./containers.nix
32          ];
33      /*
34      imports = [
35          ./containers/gcpdrgn.nix
36          ./containers/gcpsndp.nix
37          ./containers/gcnion.nix
```

```
37     ./containers/gcpdrn.nix
38     ./containers/awsndp.nix
39     ./containers/awssion.nix
40 ];
41
42 containers = {
43   gcpdrn.autoStart = false;
44   gcpsndp.autoStart = false;
45   gcpion.autoStart = true;
46   awsndp.autoStart = false;
47   awssion.autoStart = true;
48 };
49 */
50
51
52 boot = {
53   supportedFilesystems = [ "ntfs" ];
54   #kernelPackages = pkgs.linuxPackages_latest;
55   # kernelPackages = unstable.linuxPackages_latest;
56   blacklistedKernelModules = [ "nouveau" ];
57   cleanTmpDir = true;
58   # extraModulePackages = with config.boot.kernelPackages; [ wireguard ]
59   # extraModulePackages = [ config.boot.kernelPackages.wireguard ];
60   # kernelParams = [ "nvidia-drm.modeset=1" ];
61
62   kernel.sysctl = {
63     "vm.swappiness" = 10;
64     "vm.vfs_cache_pressure" = 50;
65   };
66
67   loader = {
68     systemd-boot.enable = true;
69     efi.canTouchEfiVariables = true;
70
71     grub = {
72       enableCryptodisk = true;
73       # enable = true;
74       # devices = [ "nodev" ];
75       # efiInstallAsRemovable = true;
76       # efiSupport = true;
77       # useOSProber = true;
78     };
79   };
80 };
81
82 hardware = {
83   cpu.intel.updateMicrocode = true;
84   ledger.enable = true;
85   enableRedistributableFirmware = true;
86   # enableAllFirmware = true;
87
88   sane = {
89     enable = true;
```

```
90     extraBackends = with pkgs; [ hplipWithPlugin ];
91 };
92
93 bluetooth = {
94     enable = true;
95     package = pkgs.bluezFull;
96 };
97
98 pulseaudio = {
99     enable = true;
100     support32Bit = true;
101     package = pkgs.pulseaudioFull;
102 };
103
104 opengl = {
105     driSupport = true;
106     driSupport32Bit = true;
107 };
108
109 nvidia = {
110     # package = "config.boot.kernelPackages.nvidiaPackages.beta";
111     modesetting.enable = true;
112     # nvidiaPersistenced = false;
113     prime = {
114         sync.enable = true;
115         intelBusId = "PCI:0:2:0";
116         nvidiaBusId = "PCI:1:0:0";
117     };
118 };
119 };
120
121 networking = {
122     hostName = "p53-nixos";
123     enableIPv6 = false;
124     useDHCP = false;
125     # interfaces.enp0s31f6.useDHCP = true;
126     # nameservers = [ "8.8.8.8" "8.8.4.4" ];
127     # defaultGateway = "192.168.1.1";
128     #nameservers = [ "1.1.1.1" "1.0.0.1" ];
129     #resolvconf.enable = false;
130     # interfaces.enp0s31f6.ipv4.addresses = [{
131     #     address = "192.168.1.11";
132     #     prefixLength = 24;
133     # }];
134     networkmanager = {
135         enable = true;
136         #insertNameservers = [ "45.90.28.239" "45.90.30.239" ];
137         #dns = "none";
138         #unmanaged = [ "enp0s31f6" ];
139     };
140     # Enables wireless support via wpa_supplicant.
141     # wireless.enable = true;
142     # Configure network proxy if necessary
```

```
142 # configure network proxy if necessary
143 # proxy.default = "http://user:password@proxy:port/";
144 # proxy.noProxy = "127.0.0.1,localhost,internal.domain";
145 /*
146     nat = {
147         enable = true;
148         internalInterfaces = ["ve-+"];
149         externalInterface = "enp0s31f6";
150     };
151 */
152
153 # Open ports in the firewall.
154 firewall = {
155     # Or disable the firewall altogether.
156     enable = true;
157     allowedTCPPorts = [ 20 21 22 53 80 443 2022 ];
158     allowedUDPPorts = [ 53 ];
159     allowedTCPPortRanges = [ { from = 51000; to = 51999; } ];
160     allowPing = true;
161     trustedInterfaces = [ "docker0" "virbr0" ];
162 };
163 };
164
165 # TPM has hardware RNG
166 security = {
167     polkit.enable = true;
168     sudo = {
169         enable = true;
170         wheelNeedsPassword = false;
171     };
172
173     # Yubikey for Login
174     # nix-shell -p yubico-pam -p yubikey-personalization
175     # ykpersonalize -2 -ochal-resp -ochal-hmac -ohmac-lt64 -oserial-api-v1
176     # ykpamcfg -2 -v
177     /*
178         pam.yubico = {
179             enable = true;
180             debug = true;
181             mode = "challenge-response";
182             control = "required"; # for sudo
183         };
184     */
185
186     # Finger Print Auth //Not working 100% yet with fprint
187     /*
188         pam.services = {
189             login.fprintAuth = true;
190             xautolock.fprintAuth = true;
191         };
192     */
193
194     # Example how to use pam
195     /*
```

```
195     pam.services = [
196     { name = "gnome_keyring"
197     text = ''
198     auth      optional    ${gnome3.gnome_keyring}/lib/security/pam_gnome_
199     session   optional    ${gnome3.gnome_keyring}/lib/security/pam_gnome_
200     password  optional    ${gnome3.gnome_keyring}/lib/security/pam_gnome_
201     '' ;
202     }
203     ];
204     */
205 };
206
207 /*
208     powerManagement = {
209     enable = true;
210     powertop.enable = true;
211     cpuFreqGovernor = "ondemand"; # "powersave", "performance"
212     cpuFreqGovernor = "powersave"; # "ondemand", "performance"
213     };
214     */
215
216     fileSystems."/home/mudrii/mnt/nvme" = {
217     device = "/dev/nvme1n1p2";
218     fsType = "ext4";
219     options = [ "rw" "nofail" "noatime" "nodiratime" "discard" "auto" "ext4" ];
220     };
221
222     fileSystems."/home/mudrii/.bitmonero" = {
223     device = "/dev/nvme1n1p1";
224     fsType = "ext4";
225     options = [ "rw" "nofail" "noatime" "nodiratime" "discard" "auto" "ext4" ];
226     # options = [ "rw" "uid=1000" ];
227     };
228
229     programs = {
230     # seahorse app works with gnome-keyring
231     seahorse.enable = true;
232     vim.defaultEditor = true;
233     mtr.enable = true;
234     nm-applet.enable = true;
235
236     ssh = {
237     startAgent = false;
238     # forwardX11 = true;
239     # setXAuthLocation = true;
240     };
241
242     gnupg.agent = {
243     enable = true;
244     enableSSHSuport = true;
245     };
246
```

```

247     chromium = {
248         enable = true;
249     };
250
251     bash = {
252         enableCompletion = true;
253         # shellInit = "neofetch";
254         promptInit = ''
255             PS1="\n\[\033[1;32m\]\e[0;31m\[\[\e]0;\u@\h: \w\a\]\e[01;32m\u@\[\e
256         '';
257     };
258
259     fish = {
260         enable = true;
261         vendor = {
262             completions.enable = true;
263             config.enable = true;
264         };
265         # shellInit = "neofetch";
266         # functions = { fish_greeting = ""; };
267         /*
268             promptInit = ''
269                 any-nix-shell fish --info-right | source
270             '';
271         */
272     };
273
274     nano.nanorc = ''
275         unset backup
276         set nonewlines
277         set nowrap
278         set tabstospaces
279         set tabsize 4
280         set constantshow
281     '';
282 };
283
284 # List services that you want to enable:
285 virtualisation = {
286     /*
287         docker = {
288             enable = true;
289             enableNvidia = true;
290             autoPrune.enable = true;
291             enableOnBoot = true;
292         };
293     */
294     podman = {
295         enable = true;
296         #extraPackages = [ pkgs.gvisor ];
297         enableNvidia = true;
298         dockerCompat = true;
299     };

```

```
300
301     libvirt = {
302         enable = true;
303         qemu.runAsRoot = false;
304     };
305     /*
306         lxd = {
307             enable = true;
308         };
309     */
310     /*
311         virtualbox = {
312             host.enable = true;
313             host.enableExtensionPack = true;
314         };
315     */
316 };
317
318 services = {
319     #resolved.enable = false;
320     # localtime.enable = true;
321     # urxvtd.enable = true;
322     # vnstat.enable = true;
323     blueman.enable = true;
324     fwupd.enable = true;
325     fstrim.enable = true;
326     sysstat.enable = true;
327     gnome.gnome-keyring.enable = true;
328     fail2ban.enable = true;
329     #emacs.enable = true;
330     pcscd.enable = true; # needed for YubiKey
331     xmr-stak.cudaSupport = true;
332     lorri.enable = true;
333     #autorandr.enable = true;
334     upower.enable = true;
335     /*
336         nextdns = {
337             enable = true;
338             arguments = [
339                 "-config"
340                 "5954dd"
341             ];
342         };
343     */
344     /*
345         unbound = {
346             enable = true;
347             settings = {
348                 server = {
349                     interface = [ "127.0.0.1" ];
350                 };
351             };
352         };
353     */
354 }
```

```

352         forward-zone = {
353             name = ".";
354             forward-tls-upstream = "yes";
355             forward-addr = [
356                 "45.90.28.0#2ed827.dns1.nextdns.io"
357                 "2a07:a8c0::#2ed827.dns1.nextdns.io"
358                 "45.90.30.0#2ed827.dns2.nextdns.io"
359                 "2a07:a8c1::#2ed827.dns2.nextdns.io"
360             ];
361         };
362     };
363 };
364 */
365 vsftpd = {
366     enable = true;
367     # cannot chroot && write
368     # chrootlocalUser = true;
369     writeEnable = true;
370     localUsers = true;
371     userList = [ "mudrii" ];
372     userListEnable = true;
373     extraConfig = ''
374         pasv_enable=Yes
375         pasv_min_port=51000
376         pasv_max_port=51999
377     '';
378 };
379
380 logrotate = {
381     enable = true;
382 };
383 /*
384 # Finger Print unlock login
385 fprintd = {
386     enable = true;
387     package = unstable.pkgs.fprintd;
388 };
389 */
390 /*
391 clamav = {
392     daemon.enable = true;
393     updater.enable = true;
394 };
395 */
396 udev = {
397     packages = [
398         pkgs.yubikey-personalization # needed for YubiKey
399         pkgs.libu2f-host # needed for Yubikey
400     ];
401     path = [
402         pkgs.coreutils
403     ];
404     # extraConfig = ''

```



```
404     extrakules = ''
405         ACTION=="add", SUBSYSTEM=="backlight", RUN+="${pkgs.coreutils}/bin/
406     '';
407 };
408
409
410 thinkfan = {
411     enable = true;
412     /*
413         levels = ''
414             (0,      0,      65)
415             (1,      58,      70)
416             (2,      60,      71)
417             (3,      62,      73)
418             (6,      66,      75)
419             (7,      70,      95)
420             (127,    90,      32767)
421         '';
422     */
423 };
424
425 undervolt = {
426     enable = true;
427     package = pkgs.undervolt;
428     temp = 97;
429     coreOffset = -125;
430 };
431
432 # Power button invokes suspend, not shutdown.
433 logind = {
434     extraConfig = "HandlePowerKey=suspend";
435     lidSwitch = "suspend";
436 };
437
438 # Generate thermal-conf.xml
439 # git clone https://github.com/intel/dptfxtract.git
440 # cd dptfxtract
441 # sudo acpidump > acpi.out
442 # acpixtract -a acpi.out
443 # sudo ./dptfxtract-static *.dat
444
445 thermald = {
446     enable = true;
447     configFile = builtins.toFile "thermal-conf.xml" ''
448         <!-- BEGIN -->
449         <ThermalConfiguration>
450         <Platform>
451             <Name> Auto generated </Name>
452             <ProductName>Standard PC (Q35 + ICH9, 2009)</ProductName>
453             <Preference>QUIET</Preference>
454             <ThermalZones>
455                 <ThermalZone>
456                     <Type>auto_zone_0</Type>
```

```

457         <TripPoints>
458             <TripPoint>
459                 <SensorType>B0D4</SensorType>
460                 <Temperature>100000</Temperature>
461                 <Type>Passive</Type>
462                 <CoolingDevice>
463                     <Type>B0D4</Type>
464                     <SamplingPeriod>0</SamplingPeriod>
465                 </CoolingDevice>
466             </TripPoint>
467         </TripPoints>
468     </ThermalZone>
469     <ThermalZone>
470         <Type>auto_zone_1</Type>
471         <TripPoints>
472             <TripPoint>
473                 <SensorType>SEN1</SensorType>
474                 <Temperature>99000</Temperature>
475                 <Type>Passive</Type>
476                 <CoolingDevice>
477                     <Type>B0D4</Type>
478                     <SamplingPeriod>1</SamplingPeriod>
479                 </CoolingDevice>
480             </TripPoint>
481         </TripPoints>
482     </ThermalZone>
483 </ThermalZones>
484 </Platform>
485 </ThermalConfiguration>
486 <!-- END -->
487 '';
488 };
489
490 chrony = {
491     enable = true;
492     servers = [
493         "0.sg.pool.ntp.org"
494         "1.sg.pool.ntp.org"
495         "2.sg.pool.ntp.org"
496         "3.sg.pool.ntp.org"
497     ];
498 };
499
500 /*
501     timesyncd = {
502         enable = true;
503         servers = [
504             "0.sg.pool.ntp.org"
505             "1.sg.pool.ntp.org"
506             "2.sg.pool.ntp.org"
507             "3.sg.pool.ntp.org"
508         ];

```

```
509     };
510 */
511
512 openssh = {
513     enable = true;
514     permitRootLogin = "no";
515     passwordAuthentication = false;
516     kbdInteractiveAuthentication = false;
517     forwardX11 = true;
518     ports = [ 2022 ];
519 };
520
521 # Enable CUPS to print documents.
522 printing = {
523     enable = true;
524     # drivers = [ unstable.pkgs.epson-escpr ];
525     drivers = with pkgs; [ unstable.epson-escpr ];
526 };
527
528 avahi = {
529     enable = true;
530     hostName = config.networking.hostName;
531     ipv4 = true;
532     nssmdns = true;
533     publish = {
534         enable = true;
535         userServices = true;
536     };
537 };
538
539 tlp = {
540     enable = true;
541     # extraConfig = ''
542     settings = {
543         # Detailed info can be found https://linrunner.de/tlp/settings/ir
544
545         # Disables builtin radio devices on boot:
546         #   bluetooth
547         #   wifi - Wireless LAN (Wi-Fi)
548         #   wwan - Wireless Wide Area Network (3G/UMTS, 4G/LTE, 5G)
549         # DEVICES_TO_DISABLE_ON_STARTUP="bluetooth wifi"
550
551         # When a LAN, Wi-Fi or WWAN connection has been established, the s
552         #   bluetooth
553         #   wifi - Wireless LAN
554         #   wwan - Wireless Wide Area Network (3G/UMTS, 4G/LTE, 5G)
555         # DEVICES_TO_DISABLE_ON_LAN_CONNECT="wifi wwan"
556         # DEVICES_TO_DISABLE_ON_WIFI_CONNECT="wwan"
557         # DEVICES_TO_DISABLE_ON_WWAN_CONNECT="wifi"
558
559         # When a LAN, Wi-Fi, WWAN connection has been disconnected, the st
560         # DEVICES_TO_ENABLE_ON_LAN_DISCONNECT="wifi wwan"
561         # DEVICES_TO_ENABLE_ON_WIFI_DISCONNECT=""
```

```
562 # DEVICES_TO_ENABLE_ON_WWAN_DISCONNECT=""
563
564 # Set battery charge thresholds for main battery (BAT0) and auxili
565 START_CHARGE_THRESH_BAT0 = 30;
566 STOP_CHARGE_THRESH_BAT0 = 70;
567
568 # Control battery feature drivers:
569 NATACPI_ENABLE = 1;
570 TPACPI_ENABLE = 1;
571 TPSMAPI_ENABLE = 1;
572
573 # Defines the disk devices the following parameters are effective
574 DISK_DEVICES = "nvme0n1 nvme1n1";
575
576 # Set the "Advanced Power Management Level". Possible values range
577 # 1 - max power saving / minimum performance - Important: this se
578 # 128 - compromise between power saving and wear (TLP standard se
579 # 192 - prevents excessive head unloading of some HDDs
580 # 254 - minimum power saving / max performance (TLP standard sett
581 # 255 - disable APM (not supported by some disk models)
582 # keep - special value to skip this setting for the particular di
583 DISK_APM_LEVEL_ON_AC = "254 254";
584 DISK_APM_LEVEL_ON_BAT = "128 128";
585
586 # Set the min/max/turbo frequency for the Intel GPU. Possible val
587 # INTEL_GPU_MIN_FREQ_ON_AC=0
588 # INTEL_GPU_MIN_FREQ_ON_BAT=0
589 # INTEL_GPU_MAX_FREQ_ON_AC=0
590 # INTEL_GPU_MAX_FREQ_ON_BAT=0
591 # INTEL_GPU_BOOST_FREQ_ON_AC=0
592 # INTEL_GPU_BOOST_FREQ_ON_BAT=0
593
594 # Selects the CPU scaling governor for automatic frequency scaling
595 # For Intel Core i 2nd gen. ("Sandy Bridge") or newer Intel CPUs.
596 # powersave - recommended (kernel default)
597 # performance
598 # CPU_SCALING_GOVERNOR_ON_AC=powersave;
599 # CPU_SCALING_GOVERNOR_ON_BAT=powersave;
600
601 # Set Intel CPU energy/performance policy HWP.EPP. Possible values
602 # performance
603 # balance_performance
604 # default
605 # balance_power
606 # power
607 # for tlp-stat Version 1.3 and higher 'tlp-stat -p'
608 # CPU_ENERGY_PERF_POLICY_ON_AC=balance_performance;
609 # CPU_ENERGY_PERF_POLICY_ON_BAT=power;
610
611 # Set Intel CPU energy/performance policy HWP.EPP. Possible values
612 # performance
613 # balance_performance
```

```

614         #    default
615         #    balance_power
616         #    power
617         # Version 1.2.2 and lower For version 1.3 and higher this parameter
618         # CPU_HWP_ON_AC=balance_performance;
619         # CPU_HWP_ON_BAT=power;
620
621         # Define the min/max P-state for Intel Core i processors. Values are
622         CPU_MIN_PERF_ON_AC = 0;
623         CPU_MAX_PERF_ON_AC = 100;
624         CPU_MIN_PERF_ON_BAT = 0;
625         CPU_MAX_PERF_ON_BAT = 30;
626
627         # Disable CPU "turbo boost" (Intel) or "turbo core" (AMD) feature
628         CPU_BOOST_ON_AC = 1;
629         CPU_BOOST_ON_BAT = 0;
630
631         # Minimize number of used CPU cores/hyper-threads under light load
632         SCHED_POWERSAVE_ON_AC = 0;
633         SCHED_POWERSAVE_ON_BAT = 1;
634
635         # Set Intel CPU energy/performance policy EPB. Possible values are
636         #    performance
637         #    balance-performance
638         #    default (deprecated: normal)
639         #    balance-power
640         #    power (deprecated: powersave)
641         # Version 1.2.2 and lower For version 1.3 and higher this parameter
642         # ENERGY_PERF_POLICY_ON_AC=balance-performance;
643         # ENERGY_PERF_POLICY_ON_BAT=power;
644
645         # Timeout (in seconds) for the audio power saving mode (supports 1
646         SOUND_POWER_SAVE_ON_AC = 0;
647         SOUND_POWER_SAVE_ON_BAT = 1;
648
649         # Controls runtime power management for PCIe devices.
650         # RUNTIME_PM_ON_AC=on;
651         # RUNTIME_PM_ON_BAT=auto;
652
653         # Exclude PCIe devices assigned to listed drivers from runtime pow
654         # RUNTIME_PM_DRIVER_BLACKLIST="mei_me nouveau nvidia pcieport rade
655
656         # Sets PCIe ASPM power saving mode. Possible values:
657         #    default - recommended
658         #    performance
659         #    powersave
660         #    powersupersave
661         # PCIE_ASPM_ON_AC=default;
662         # PCIE_ASPM_ON_BAT=default;
663         #'';
664     };
665 };

```

```
000
667     dbus = {
668         enable = true;
669         packages = [ pkgs.fwupd ];
670     };
671
672     locate = {
673         enable = true;
674         locate = unstable.pkgs.mlocate;
675         localuser = null; # mlocate does not support this option so it must
676         # interval = "daily";
677         interval = "hourly";
678         pruneNames = [
679             ".git"
680             "cache"
681             ".cache"
682             ".cpcache"
683             ".aot_cache"
684             ".boot"
685             "node_modules"
686             "USB"
687         ];
688         prunePaths = options.services.locate.prunePaths.default ++ [
689             "/dev"
690             "/lost+found"
691             "/nix/var"
692             "/proc"
693             "/run"
694             "/sys"
695             "/usr/tmp"
696         ];
697     };
698     /*
699     # Monitor plug n play
700     # https://github.com/phillipberndt/autorandr/blob/v1.0/README.md#hov
701     autorandr = {
702         enable = true;
703     };
704     */
705     xserver = {
706         enable = true;
707         # autorun = false;
708         # videoDrivers = [ "intel" ];
709         # videoDrivers = [ "nvidiaBeta" ];
710         videoDrivers = [ "nvidia" ];
711         layout = "us";
712         xkbOptions = "eurosign:e";
713         dpi = 90;
714
715         libinput = {
716             enable = true;
717             touchpad = {
718                 disableWhileTyping = true;
```

```
719     naturalScrolling = false;
720     additionalOptions = ''
721         Option "PalmDetection" "True"
722     '';
723 };
724
725 };
726
727 desktopManager = {
728     xterm.enable = false;
729 };
730
731 displayManager = {
732     /*
733     sessionCommands = ''
734         ${pkgs.xorg.xrdb}/bin/xrdb -merge <<EOF
735         Xft.dpi: 192
736         EOF
737     '';
738     */
739     /*
740     setupCommands = ''
741         ${pkgs.xorg.xrandr}/bin/xrandr --setprovideroutputsource modeset
742         ${pkgs.xorg.xrandr}/bin/xrandr --auto
743     '';
744     */
745     defaultSession = "none+i3";
746     #autoLogin.enable = true;
747     #autoLogin.user = "mudrii";
748
749     lightdm = {
750         enable = true;
751         #greeter.enable = false; # for outo login
752         greeter.enable = true;
753     };
754 };
755
756 windowManager = {
757     i3.enable = true;
758     i3.package = pkgs.i3-gaps;
759     i3.extraPackages = with pkgs; [
760         dmenu #application launcher most people use
761         i3lock #default i3 screen locker
762         # i3status # gives you the default i3 status bar
763         # i3blocks #if you are planning on using i3blocks over i3status
764         # i3status-rust
765         i3-gaps
766         i3lock-fancy
767         xautolock
768         # rofi
769         numlockx
770         # conky
```

```
771         # rxvt_unicode
772         # rxvt_unicode-with-plugins
773         # (lowPrio urxvt_perls)
774         # (lowPrio urxvt_font_size)
775         acpilight
776         glxinfo
777         pavucontrol
778         networkmanager_dmenu
779         arandr
780         escrotum
781         obs-studio
782         libva-utils
783         networkmanagerapplet
784         gimp
785     ];
786 };
787 };
788 };
789
790 # Select internationalisation properties.
791 i18n = {
792     defaultLocale = "en_US.UTF-8";
793     supportedLocales = [ "en_US.UTF-8/UTF-8" ];
794 };
795
796 console = {
797     # font = "Powerline";
798     font = "Lat2-Terminus16";
799     keyMap = "us";
800 };
801
802 fonts = {
803     fontDir.enable = true;
804     enableGhostscriptFonts = true;
805
806     fonts = with pkgs; [
807         corefonts
808         # inconsolata
809         # unifont
810         ubuntu_font_family
811         # symbola
812         nerdfonts
813         freefont_ttf
814         powerline-fonts
815         # font-awesome
816         font-awesome_4
817         dejavu_fonts
818         google-fonts
819         noto-fonts
820         # noto-fonts-cjk
821         # noto-fonts-emoji
822         # liberation_ttf
823         # fira-code
```



```
824     # fira-code-symbols
825     # mplus-outline-fonts
826     # dina-font
827     # proggyfonts
828     # emoji-one
829     # twemoji-color-font
830     # mononoki
831 ];
832 };
833
834 # Enable sound.
835 sound.enable = true;
836
837 # Set your time zone.
838 time.timeZone = "Asia/Singapore";
839
840 # environment.pathsToLink = [ "/libexec" ]; # links /libexec from deriv
841 # environment.systemPackages = with pkgs; [ xorg.xbacklight ];
842
843 environment = {
844     etc."sysconfig/lm_sensors".text = ''
845         # Generated by sensors-detect on Tue Aug  7 10:54:09 2018
846         # This file is sourced by /etc/init.d/lm_sensors and defines the mod
847         # be loaded/unloaded.
848         #
849         # The format of this file is a shell script that simply defines vari
850         # HWMON_MODULES for hardware monitoring driver modules, and optional
851         # BUS_MODULES for any required bus driver module (for example for I2
852
853         HWMON_MODULES="coretemp"
854     '';
855
856     # YubiKey SSH and GPG support
857
858     shellInit = ''
859         export GPG_TTY="$(tty)"
860         gpg-connect-agent /bye
861         export SSH_AUTH_SOCK="/run/user/$UID/gnupg/S.gpg-agent.ssh"
862         # export SSH_AUTH_SOCK=$(gpgconf --list-dirs agent-ssh-socket)
863     '';
864
865     variables = {
866         # Preferred applications
867         EDITOR = "nvim";
868         BROWSER = "qutebrowser";
869     };
870
871     enableDebugInfo = true;
872
873     systemPackages = with pkgs; [
874         efibootmgr
875         efivar
```

```
876      chkrootkit
877      lynis
878      # nixfmt
879      nix-index
880      nixpkgs-fmt
881      nixpkgs-lint
882      nix-prefetch
883      nix-du
884      nix-top
885      nix-linter
886      nixpkgs-review
887      nixpkgs-pytools
888      nix-update
889      nix-tree
890      nixos-generators
891      nvd
892      rnix-lsp
893      unstable.nix-simple-deploy
894      niv
895      graphviz
896      (lowPrio nix-prefetch-git)
897      nix-prefetch-scripts
898      #qemu
899      #qemu-utils
900      qemu_full
901      #unstable.virt-manager
902      #unstable.virt-viewer
903      virt-manager
904      virt-viewer
905      virt-top
906      # undervolt
907      acpica-tools
908      patchelf
909      binutils
910      cryptsetup
911      dstat
912      unstable.duf
913      duff
914      gptfdisk
915      parted
916      mc
917      nnn
918      # vifm
919      # lf
920      trash-cli
921      traceroute
922      whois
923      which
924      nmap
925      wget
926      curl
927      speedtest-cli
928      unstable.xournal
```

```
928         unstable.neovim
929     # micro
930     commonsCompress
931     libarchive
932     archiver
933     # p7zip
934     unzip
935     unrar
936     zstd
937     unstable.cpu-x
938     lsof
939     acpi
940     pciutils
941     usbutils
942     powertop
943     tree
944     jdupes
945     #ag
946     silver-searcher
947     gtop
948     iftop
949     atop
950     nethogs
951     nmon
952     iotop
953     sysstat
954     tcpdump
955     atool
956     rsync
957     gnutils
958     bind
959     mkpasswd
960     openssl
961     unstable.cfssl
962     file
963     lshw
964     lm_sensors
965     inxi
966     s-tui
967     stress-ng
968     tpacpi-bat
969     # tlp
970     cpufrequitls
971     msr-tools
972     nvtop
973     sshfs
974     vdpauinfo
975     dmidecode
976     fwupd
977     # fwupdate
978     ncdu
979     smartmontools
980     pass
```

```
981         encfs
982         linuxPackages.perf
983         networkmanager
984         wirelesstools
985         # blueman
986         # (lowPrio bluez)
987         # bluez-tools
988         mtr
989         nftables
990         psmisc
991         ripgrep-all
992         du-dust
993         # exa
994         srm
995         tcpdump
996         ethtool
997         bmon
998         (lowPrio inetutils)
999         socat
1000        iptables
1001        iperf
1002        gping
1003        nload
1004        nvme-cli
1005        ncurses
1006        protonvpn-cli
1007        # unstable.protonvpn-gui
1008        openvpn
1009        wireguard-tools
1010        # wireshark-cli
1011        # wireshark
1012        # aircrack-ng
1013        gopass
1014        xmrig
1015        xmr-stak
1016        unstable.linux-wifi-hotspot
1017        unstable.dnsmasq
1018        # unstable.solaar # logitech mous unify receiver config
1019        #nextdns
1020    ];
1021
1022    shellAliases = {
1023        ans = "ansible";
1024        ansp = "ansible-playbook";
1025        ansv = "ansible-vault";
1026        ccat = "pygmentize -f terminal256 -g -P style=monokai";
1027        cp = "cp -i";
1028        diff = "diff --color=auto";
1029        dmesg = "dmesg --color=always | lless";
1030        egrep = "egrep --color=auto";
1031        fgrep = "fgrep --color=auto";
1032        gcl = "gcloud";
```

```

1033     grep = "grep --color=auto";
1034     gadc = "git add -A; and git commit";
1035     gad = "git add .";
1036     gcm = "git commit";
1037     gdf = "git diff";
1038     gl = "git log";
1039     glg = "git log --color --graph --pretty --oneline";
1040     gpl = "git pull";
1041     gps = "git push";
1042     gst = "git status";
1043     kcl = "kubectl";
1044     kms = "kubens";
1045     ktx = "kubectx";
1046     la = "exa -alg --group-directories-first -s=type --icons";
1047     # la="ls -lha --color=auto --group-directories-first";
1048     lless = "set -gx LESSOPEN '|pygmentize -f terminal256 -g -P style=mc';";
1049     # ll="ls -lah";
1050     # ls="ls --color=auto";
1051     ll = "exa -la";
1052     ls = "exa";
1053     mv = "mv -i";
1054     nixcl = "sudo nix-store --optimise -v && sudo nix-collect-garbage -c";
1055     nixup = "sudo nix-channel --update && sudo nixos-rebuild switch";
1056     nixts = "sudo nix-channel --update && sudo nixos-rebuild test";
1057     nixg = "git --git-dir=$HOME/src/nixlenovodot/ --work-tree=/etc/nixos";
1058     py = "python";
1059     ping = "ping -c3";
1060     ps = "ps -ef";
1061     pvpn = "sudo protonvpn";
1062     pul = "pulumi";
1063     # rm = "rm -i";
1064     # rmf = "rm -rf";
1065     rm = "trash-put";
1066     unrm = "trash-restore";
1067     rmcl = "trash-empty";
1068     rml = "trash-list";
1069     sudo = "sudo -i";
1070     suf = "su --shell=/usr/bin/fish";
1071     ter = "terraform";
1072     vdir = "vdir --color=auto";
1073 };
1074 };
1075
1076 users = {
1077     mutableUsers = false;
1078     users.irutsu = {
1079         isNormalUser = true;
1080         home = "/home/irutsu";
1081         shell = pkgs.fish;
1082         description = "irutsu";
1083         extraGroups = [ "wheel" "audio" "video" "networkmanager" ];
1084         # mkpasswd -m sha-512 password
1085         hashedPassword = "$6$/RP5F11e$Hvcukkdtd/iRkM99IINrNT05r/xOPTiRzK40021";

```

```
1086     openssh.authorizedKeys.keys = [ "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQ= mudrii@nixlenovodot"; ];
1087 packages = with pkgs; [
1088     home-manager
1089 ];
1090 };
1091
1092 users.mudrii = {
1093     isNormalUser = true;
1094     home = "/home/mudrii";
1095     shell = pkgs.fish;
1096     description = "mudrii";
1097     extraGroups = [ "wheel" "docker" "lp" "audio" "video" "tty" "input"
1098 # mkpasswd -m sha-512 password
1099     hashedPassword = "$6$ewXNcoQRNG$czTic9vE8CGH.eo4mabZsHVRdmTjtJF4SdDr";
1100     openssh.authorizedKeys.keys = [ "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQ= mudrii@nixlenovodot"; ];
1101     packages = with pkgs; [
1102         home-manager
1103         unstable.bpytop
1104         unstable.glances
1105         neofetch
1106         poppler_utils
1107         elinks
1108         w3m
1109         ffmpeg-full
1110         ffmpegthumbnailer
1111         exif
1112         exiftool
1113         mupdf
1114         screen
1115         keychain
1116         unstable.minio-client
1117         google-cloud-sdk-gce
1118         unstable.awscli
1119         # unstable.clojure
1120         # clojure-lsp
1121         # pulumi-bin
1122         # unstable.pulumi-bin
1123         unstable.gitAndTools.gitFull
1124         unstable.gitAndTools.gh
1125         unstable.git-crypt
1126         unstable.git-lfs
1127         unstable.terraform
1128         unstable.terraform-lsp
1129         unstable.tflint
1130         unstable.kubernetes
1131         unstable.kubernetes-helm
1132         # unstable.kubeseal
1133         # unstable.helmfile
1134         # unstable.helmsman
1135         # unstable.kind
1136         # unstable.kube3d
1137         # unstable.skopeo
```

```
1138     # unstable.dive
1139     # unstable.lens
1140     # unstable.docker-machine-kvm2
1141     # unstable.minikube
1142     # unstable.fluxctl
1143     # unstable.argo
1144     # unstable.argocd
1145     # unstable.kustomize
1146     # unstable.k9s
1147     # unstable.velero
1148     # unstable.go
1149     # unstable.xmind
1150     # teams
1151     signal-desktop
1152     # kubectx
1153     dep
1154     /*
1155     (unstable.terraform.withPlugins(p: with p; [
1156         archive
1157         aws
1158         external
1159         google
1160         helm
1161         kubernetes
1162         local
1163         null
1164         random
1165         template
1166     ]))
1167     */
1168     # (lowPrio unstable.python39Full)
1169     # python3Full
1170     # python3Full
1171     # python39Packages.poetry
1172     # unstable.python38Packages.pynvim
1173     # poetry
1174     (
1175         python39.withPackages (
1176             ps: with ps; [
1177                 #poetry
1178                 pip
1179                 powerline
1180                 pygments
1181                 yubico-client
1182                 pygments-markdown-lexer
1183                 xstatic-pygments
1184                 # pylint
1185                 # pandas
1186                 # pycuda
1187                 # numpy
1188                 # opencv4
1189                 # scipy
1190                 # matplotlib
```

```
1190         # matplotlib
1191         # jupyter
1192         pynvim
1193         pyopenssl
1194     ]
1195 )
1196 )
1197 aspell
1198 aspellDicts.en
1199 asciinema
1200 unstable.direnv
1201 psensor
1202 thunderbird
1203 # libreoffice
1204 onlyoffice-bin
1205 qutebrowser
1206 bitwarden
1207 bitwarden-cli
1208 shotcut
1209 # unstable.amfora
1210 unstable.lagrange
1211 unstable.vscode
1212 # vscodium
1213 unstable.sublime4
1214 unstable.sublime-merge
1215 # unstable.insomnia
1216 slack
1217 vlc
1218 gpview
1219 ripgrep
1220 ripgrep-all
1221 tldr
1222 procs
1223 fd
1224 # zathura
1225 mpv
1226 pv
1227 feh
1228 # weechat
1229 gcc
1230 gnumake
1231 gnupg
1232 pinentry-curses
1233 pinentry-qt
1234 paperkey
1235 pam_u2f
1236 libu2f-host
1237 libu2f-server
1238 gpa
1239 imagemagick
1240 spotify
1241 bookworm
1242 ltrace
```



```
1243      /*
1244      (unstable.tor-browser-bundle-bin.override {
1245      mediaSupport = true;
1246      pulseaudioSupport = true;
1247      })
1248      */
1249      unstable.KeepassXC
1250      openconnect_openssl
1251      networkmanager-openconnect
1252      unstable.element-desktop
1253      unstable.ansible
1254      graphviz
1255      scrcpy
1256      unstable.pcmanfm
1257      #mucommander
1258      unstable.hexchat
1259      unstable.grsync
1260      unstable.luckybackup
1261      unstable.drawio
1262      unstable.youtube-dl
1263      yubikey-manager
1264      yubico-piv-tool
1265      yubikey-manager-qt
1266      yubikey-personalization
1267      yubikey-personalization-gui
1268      yubioath-desktop
1269      yubico-pam
1270      unstable.monero
1271      unstable.monero-gui
1272      #unstable.xmrig
1273      #unstable.xmr-stak
1274      unstable.ledger
1275      unstable.ledger-web
1276      unstable.ledger-live-desktop
1277      unstable.audacity
1278      unstable.nodejs
1279      unicode-paracode
1280      system-config-printer
1281      play-with-mpv
1282      jetbrains.idea-ultimate
1283      viber
1284    ];
1285  };
1286 };
1287
1288 nix = {
1289   # package = pkgs.nixUnstable;
1290   package = pkgs.nixFlakes;
1291   useSandbox = true;
1292   autoOptimiseStore = true;
1293   readOnlyStore = false;
1294   allowedUsers = [ "@wheel" "mudrii" ];
```

```

1295     trustedUsers = [ "@wheel" "mudrii" ];
1296
1297     extraOptions = ''
1298         keep-outputs = true
1299         keep-derivations = true
1300         # preallocate-contents = false
1301         experimental-features = nix-command flakes
1302     '';
1303
1304     gc = {
1305         automatic = true;
1306         dates = "weekly";
1307         options = "--delete-older-than 7d --max-freed $((64 * 1024**3))";
1308         # dates = "Mon *-*-* 06:00:00";
1309     };
1310
1311     optimise = {
1312         automatic = true;
1313         dates = [ "weekly" ];
1314     };
1315
1316 };
1317
1318 nixpkgs = {
1319     config = {
1320         pulseaudio = true;
1321         allowBroken = true;
1322         allowUnfree = true;
1323         permittedInsecurePackages = [
1324             "python3.9-poetry-1.1.12"
1325         ];
1326         packageOverrides = pkgs: {
1327             # unstable = import <nixpkgs-unstable> {
1328             #     config = config.nixpkgs.config;
1329             # };
1330             nur = import (builtins.fetchTarball "https://github.com/nix-commur
1331                 inherit pkgs;
1332             };
1333         };
1334     };
1335     /*
1336     overlays = [
1337         (self: super: {
1338
1339             element-desktop = super.element-desktop.overrideAttrs (old: rec {
1340                 version = "1.7.27";
1341                 src = pkgs.fetchFromGitHub {
1342                     owner = "vector-im";
1343                     repo = "element-desktop";
1344                     rev = "v${version}";
1345                     sha256 = "0rgsc2cc1v6gjsklwvsjlqq9a8j9j80h9ac0jkvf9lhq33f3c57k";
1346                 };
1347             });

```

```

1348
1349     /*
1350     fwup = super.fwup.overrideAttrs (old: {
1351     src = super.fetchFromGitHub {
1352     owner = "fhunleth";
1353     repo = "fwup";
1354     rev = "v1.8.3";
1355     sha256 = "0p3kp1kai5zrgagjzhd41gl84gfqk04qnq1d1dnf0ckvhsfdq9vb";
1356     };
1357     buildInputs = with super; [
1358     libarchive
1359     libconfuse
1360     which
1361     xdelta
1362     ];
1363     });
1364     */
1365     /*
1366     youtube-dl = super.youtube-dl.overrideAttrs (old: rec {
1367     pname = "youtube-dl";
1368     version = "2020.11.11-3";
1369     postInstall = "";
1370     src = pkgs.fetchurl {
1371     url = "https://github.com/blackjack4494/yt-dlc/archive/${version}.tar.gz";
1372     sha256 = "116azzzj0df3bv99zwn0rsarirw56knbq7xqn1fs8v4ilslqp7v4";
1373     };
1374     });
1375     */
1376     #    })
1377     #    ];
1378 };
1379
1380 # This value determines the NixOS release with which your system is to be
1381 # compatible, in order to avoid breaking some software such as database
1382 # servers. You should change this only after NixOS release notes say you
1383 # should.
1384 system = {
1385     copySystemConfiguration = true;
1386     stateVersion = "22.05"; # Did you read the comment?
1387     autoUpgrade = {
1388         enable = true;
1389         dates = "weekly";
1390         allowReboot = false;
1391         # dates = "Sun *-*-* 04:00:00";
1392     };
1393 };
1394 /*
1395     system = {
1396     autoUpgrade = {
1397     enable = true;
1398     allowReboot = true;
1399     flake = self.outPath;

```

```
1400     flags = [  
1401         "--recreate-lock-file"  
1402         "--no-write-lock-file"  
1403         "-L" # print build logs  
1404     ];  
1405     dates = "daily";  
1406 };  
1407 }:  
1408 */  
1409 }
```