



**Added apple silicon support**

Tyler Kelley authored 5 months ago

 config.nix 11.08 KiB

```
1  {
2      config,
3      pkgs,
4      host,
5      username,
6      options,
7      lib,
8      ...
9  }:
10 let
11     inherit (import ../variables.nix) keyboardLayout;
12 in
13 {
14     imports = [
15         ../hardware.nix
16         ../users.nix
17         ../../modules/amd-drivers.nix
18         ../../modules/apple-silicon-support
19         ../../modules/nvidia-drivers.nix
20         ../../modules/nvidia-prime-drivers.nix
21         ../../modules/intel-drivers.nix
22         ../../modules/vm-guest-services.nix
23         ../../modules/local-hardware-clock.nix
24     ];
25
26     boot = {
27         # Kernel
28         kernelPackages = lib.mkDefault pkgs.linuxPackages_zen;
29         # This is for OBS Virtual Cam Support
30         kernelModules = [ "v4l2loopback" ];
31         extraModulePackages = [ config.boot.kernelPackages.v4l2loopback ];
32         # Needed For Some Steam Games
33         kernel.sysctl = {
34             "vm.max_map_count" = 2147483642;
35         };
36         # Bootloader.
37         loader.systemd-boot.enable = true;
38         loader.efi.canTouchEfiVariables = true;
39         # Make /tmp a tmpfs
40         tmp = {
41             useTmpfs = false;
42             tmpfsSize = "30%";
43         };
44         # Appimage Support
45         binfmt.registrations.appimage = {
46             wrapInterpreterInShell = false;
47             interpreter = "${pkgs.appimage-run}/bin/appimage-run";
48             recognitionType = "magic";
49             offset = 0;
50             mask = '\xff\xff\xff\xff\x00\x00\x00\x00\xff\xff\xff';
51             magicOrExtension = '\x7fELF...AI\x02';
52         };
53         plymouth.enable = true;
54     };
55
56     # Styling Options
57     stylix = {
58         enable = true;
59         image = ../../config/wallpapers/beautifulmountainscape.jpg;
60         # base16Scheme = {
61             # base00 = "232136";
62             # base01 = "2a273f";
63             # base02 = "393552";
64             # base03 = "6e6a86";
65             # base04 = "908caa";
66             # base05 = "e0def4";
67             # base06 = "e0def4";
68             # base07 = "56526e";
69             # base08 = "eb6f92";
70             # base09 = "f6c177";
```

```
71 # base0A = "ea9a97";
72 # base0B = "3e8fb0";
73 # base0C = "9ccfd8";
74 # base0D = "c4a7e7";
75 # base0E = "f6c177";
76 # base0F = "56526e";
77 # };
78 polarity = "dark";
79 opacity.terminal = 0.8;
80 cursor.package = pkgs.bibata-cursors;
81 cursor.name = "Bibata-Modern-Ice";
82 cursor.size = 24;
83 fonts = {
84     monospace = {
85         package = pkgs.nerdfonts.override { fonts = [ "JetBrainsMono" ]; };
86         name = "JetBrainsMono Nerd Font Mono";
87     };
88     sansSerif = {
89         package = pkgs.montserrat;
90         name = "Montserrat";
91     };
92     serif = {
93         package = pkgs.montserrat;
94         name = "Montserrat";
95     };
96     sizes = {
97         applications = 12;
98         terminal = 15;
99         desktop = 11;
100        popups = 12;
101    };
102 };
103 };
104
105 # Extra Module Options
106 drivers.amdgpu.enable = false;
107 drivers.nvidia.enable = false;
108 drivers.nvidia-prime = {
109     enable = false;
110     intelBusID = "";
111     nvidiaBusID = "";
112 };
113 drivers.intel.enable = false;
114 vm.guest-services.enable = false;
115 local.hardware-clock.enable = false;
116 # Apple Hardware
117 hardware.asahi.useExperimentalGPUDriver = true;
118 hardware.asahi.peripheralFirmwareDirectory = ../modules/firmware;
119
120 # Enable networking
121 networking.wireless.iwd = {
122     enable = true;
123     settings.General.EnableNetworkConfiguration = true;
124 };
125 networking.hostName = host;
126 networking.timeServers = options.networking.timeServers.default ++ [ "pool.ntp.org" ];
127
128 # Set your time zone.
129 time.timeZone = "America/Chicago";
130
131 # Select internationalisation properties.
132 i18n.defaultLocale = "en_US.UTF-8";
133
134 i18n.extraLocaleSettings = {
135     LC_ADDRESS = "en_US.UTF-8";
136     LC_IDENTIFICATION = "en_US.UTF-8";
137     LC_MEASUREMENT = "en_US.UTF-8";
138     LC_MONETARY = "en_US.UTF-8";
139     LC_NAME = "en_US.UTF-8";
140     LC_NUMERIC = "en_US.UTF-8";
```

```
141 LC_PAPER = "en_US.UTF-8";
142 LC_TELEPHONE = "en_US.UTF-8";
143 LC_TIME = "en_US.UTF-8";
144 };
145
146 programs = {
147   firefox.enable = false;
148   starship = {
149     enable = true;
150     settings = {
151       add_newline = false;
152       buf = {
153         symbol = " ";
154       };
155       c = {
156         symbol = " ";
157       };
158       directory = {
159         read_only = " ";
160       };
161       docker_context = {
162         symbol = " ";
163       };
164       fossil_branch = {
165         symbol = " ";
166       };
167       git_branch = {
168         symbol = " ";
169       };
170       golang = {
171         symbol = " ";
172       };
173       hg_branch = {
174         symbol = " ";
175       };
176       hostname = {
177         ssh_symbol = " ";
178       };
179       lua = {
180         symbol = " ";
181       };
182       memory_usage = {
183         symbol = " ";
184       };
185       meson = {
186         symbol = " ";
187       };
188       nim = {
189         symbol = " ";
190       };
191       nix_shell = {
192         symbol = " ";
193       };
194       nodejs = {
195         symbol = " ";
196       };
197       ocaml = {
198         symbol = " ";
199       };
200       package = {
201         symbol = " ";
202       };
203       python = {
204         symbol = " ";
205       };
206       rust = {
207         symbol = " ";
208       };
209       swift = {
210         symbol = " ";
```

```
211         };
212         zig = {
213             symbol = " ";
214         };
215     };
216 };
217 dconf.enable = true;
218 seahorse.enable = true;
219 fuse.userAllowOther = true;
220 mtr.enable = true;
221 gnupg.agent = {
222     enable = true;
223     enableSSHSupport = true;
224 };
225 virt-manager.enable = true;
226 thunar = {
227     enable = true;
228     plugins = with pkgs.xfce; [
229         thunar-archive-plugin
230         thunar-volman
231     ];
232 };
233 };
234
235 nixpkgs.config.allowUnfree = true;
236
237 users = {
238     mutableUsers = true;
239 };
240
241 environment.systemPackages = with pkgs; [
242     vim
243     wget
244     killall
245     eza
246     git
247     cmatrix
248     lolcat
249     htop
250     brave
251     libvirt
252     lxqt.lxqt-policykit
253     lm_sensors
254     unzip
255     unrar
256     libnotify
257     v4l-utils
258     ydotool
259     duf
260     ncdu
261     wl-clipboard
262     pciutils
263     ffmpeg
264     socat
265     cowsay
266     ripgrep
267     lshw
268     bat
269     pkg-config
270     meson
271     hyprpicker
272     ninja
273     brightnessctl
274     virt-viewer
275     swappy
276     appimage-run
277     yad
278     inxi
279     playerctl
280     nh
```

```
281     nixfmt-rfc-style
282     libvirt
283     swww
284     grim
285     slurp
286     file-roller
287     swaynotificationcenter
288     imv
289     mpv
290     gimp
291     pavucontrol
292     tree
293     neovide
294     greetd.tuigreet
295 ];
296
297 fonts = {
298     packages = with pkgs; [
299         noto-fonts-emoji
300         noto-fonts-cjk
301         font-awesome
302         symbola
303         material-icons
304     ];
305 };
306
307 environment.variables = {
308     ZANEYOS_VERSION = "2.2";
309     ZANEYOS = "true";
310 };
311
312 # Extra Portal Configuration
313 xdg.portal = {
314     enable = true;
315     wlr.enable = true;
316     extraPortals = [
317         pkgs.xdg-desktop-portal-gtk
318         pkgs.xdg-desktop-portal
319     ];
320     configPackages = [
321         pkgs.xdg-desktop-portal-gtk
322         pkgs.xdg-desktop-portal-hyprland
323         pkgs.xdg-desktop-portal
324     ];
325 };
326
327 # Services to start
328 services = {
329     xserver = {
330         enable = false;
331         xkb = {
332             layout = "${keyboardLayout}";
333             variant = "";
334         };
335     };
336     greetd = {
337         enable = true;
338         vt = 3;
339         settings = {
340             default_session = {
341                 # Wayland Desktop Manager is installed only for user ryan via home-manager!
342                 user = username;
343                 # .wayland-session is a script generated by home-manager, which links to the current wayland compositor(sway)
344                 # with such a vendor-no-locking script, we can switch to another wayland compositor without modifying greetd
345                 # command = "$HOME/.wayland-session"; # start a wayland session directly without a login manager
346                 command = "${pkgs.greetd.tuigreet}/bin/tuigreet --time --cmd Hyprland"; # start Hyprland with a TUI login manager
347             };
348         };
349     };
350     smartd = {
```

```
351     enable = false;
352     autodetect = true;
353 };
354 libinput.enable = true;
355 fstrim.enable = true;
356 gvfs.enable = true;
357 openssh.enable = true;
358 flatpak.enable = false;
359 printing = {
360     enable = true;
361     drivers = [
362         # pkgs.hplipWithPlugin
363     ];
364 };
365 gnome.gnome-keyring.enable = true;
366 avahi = {
367     enable = true;
368     nssmdns4 = true;
369     openFirewall = true;
370 };
371 ipp-usb.enable = true;
372 syncthing = {
373     enable = false;
374     user = "${username}";
375     dataDir = "/home/${username}";
376     configDir = "/home/${username}/.config/syncthing";
377 };
378 pipewire = {
379     enable = true;
380     alsa.enable = true;
381     alsa.support32Bit = true;
382     pulse.enable = true;
383 };
384 rpcbind.enable = false;
385 nfs.server.enable = false;
386 };
387 systemd.services.flatpak-repo = {
388     path = [ pkgs.flatpak ];
389     script = ''
390         flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.flatpakrepo
391     '';
392 };
393 hardware.sane = {
394     enable = true;
395     extraBackends = [ pkgs.sane-airscan ];
396     disabledDefaultBackends = [ "escl" ];
397 };
398
399 # Extra Logitech Support
400 hardware.logitech.wireless.enable = false;
401 hardware.logitech.wireless.enableGraphical = false;
402
403 # Bluetooth Support
404 hardware.bluetooth.enable = true;
405 hardware.bluetooth.powerOnBoot = true;
406 services.bluedman.enable = true;
407 hardware.pulseaudio.enable = false;
408
409 # Security / Polkit
410 security.rtkit.enable = true;
411 security.polkit.enable = true;
412 security.polkit.extraConfig = ''
413     polkit.addRule(function(action, subject) {
414         if (
415             subject.isInGroup("users")
416             && (
417                 action.id == "org.freedesktop.login1.reboot" ||
418                 action.id == "org.freedesktop.login1.reboot-multiple-sessions" ||
419                 action.id == "org.freedesktop.login1.power-off" ||
420                 action.id == "org.freedesktop.login1.power-off-multiple-sessions"
```

```
421         )
422     )
423     {
424         return polkit.Result.YES;
425     }
426 })
427 '';
428 security.pam.services.swaylock = {
429     text = ''
430         auth include login
431     '';
432 };
433
434 # Optimization settings and garbage collection automation
435 nix = {
436     settings = {
437         auto-optimise-store = true;
438         experimental-features = [
439             "nix-command"
440             "flakes"
441         ];
442         substituters = [ "https://hyprland.cachix.org" ];
443         trusted-public-keys = [ "hyprland.cachix.org-1:a7pgxzMz7+chwVL3/pzj6jIBMioiJM7ypFP8PwtkuGc=" ];
444     };
445     gc = {
446         automatic = true;
447         dates = "weekly";
448         options = "--delete-older-than 7d";
449     };
450 };
451
452 # Virtualization / Containers
453 virtualisation.libvirtd.enable = true;
454 virtualisation.podman = {
455     enable = true;
456     dockerCompat = true;
457     defaultNetwork.settings.dns_enabled = true;
458 };
459
460 # OpenGL
461 hardware.graphics = {
462     enable = true;
463     enable32Bit = false;
464 };
465
466 console.keyMap = "${keyboardLayout}";
467
468 # Open ports in the firewall.
469 # networking.firewall.allowedTCPPorts = [ ... ];
470 # networking.firewall.allowedUDPPorts = [ ... ];
471 # Or disable the firewall altogether.
472 # networking.firewall.enable = false;
473
474 # This value determines the NixOS release from which the default
475 # settings for stateful data, like file locations and database versions
476 # on your system were taken. It's perfectly fine and recommended to leave
477 # this value at the release version of the first install of this system.
478 # Before changing this value read the documentation for this option
479 # (e.g. man configuration.nix or on https://nixos.org/nixos/options.html).
480 system.stateVersion = "23.11"; # Did you read the comment?
481 }
482
```