## aaronjanse / dotfiles (Public)

<> Code ○ Issues ↑ Pull requests □ Security ✓ Insights

## dotfiles / configuration.nix aaronjanse add disabled code for zbak send 06231c1 · 4 years ago

389 lines (333 loc) · 9.65 KB

```
Code
         Blame
                                                                                                   <>
           { config, lib, pkgs, options, ... }:
    1
    2
    3
           let
    4
             theme = import ./theme.nix;
             secrets = import "${fetchGit {
    5
                 url = "ssh://git@github.com/aaronjanse/secrets.git";
    6
    7
                 rev = "16fe9eb16fb4e32d5b2b3e5d4d5cbc3ac8e0ae32";
             }}"
    8
    9
               { inherit pkgs; };
   10
           in
   11
   12
             /* Nix preferences */
   13
             nix = {
   14
   15
               package = pkgs.nixUnstable;
               extraOptions = ''
   16
                 experimental-features = nix-command flakes ca-references
   17
   18
                 builders-use-substitutes = true
   19
               '';
               distributedBuilds = true;
   20
   21
             };
   22
   23
             nix.trustedUsers = [ "ajanse" ];
   24
             /* XPS 13 hardware */
   25
   26
             boot.kernelParams = [ "mem sleep default=deep" ];
   27
             boot.supportedFilesystems = [ "zfs" ];
   28
             boot.zfs.requestEncryptionCredentials = [ "zzroot" ];
   29
             hardware.enableRedistributableFirmware = true;
   30
   31
             powerManagement.powertop.enable = true;
   32
   33
             # Disable sleep
             # # systemd.targets.sleep.enable = false;
   34
   35
             # # systemd.targets.suspend.enable = false;
             # # systemd.targets.hibernate.enable = false;
   36
                      1 4
```

```
# # systemu.targets.nyoriu-sieep.enable = Taise;
38
         boot.extraModulePackages = [ config.boot.kernelPackages.v4l2loopback ];
39
40
         hardware.opengl = {
41
           enable = true;
           extraPackages = with pkgs; [
42
             intel-media-driver # LIBVA_DRIVER_NAME=iHD
43
44
             vaapiIntel # LIBVA_DRIVER_NAME=i965 (older but works better for Firefox/Chromium)
45
             vaapiVdpau
             libvdpau-va-gl
46
47
            ];
           driSupport32Bit = true;
48
49
         };
50
         environment.systemPackages = with pkgs; [
51
           nixFlakes
52
           git
53
           vaapiIntel
54
           vaapi-intel-hybrid
55
           libva-full
           libva-utils
57
           intel-media-driver
           v4l-utils
58
59
            gnome3.adwaita-icon-theme
60
           vanilla-dmz
61
         ];
63
         # services.neo4j = {
64
         # enable = true;
             bolt.tlsLevel = "DISABLED";
            https.enable = false;
66
         #
            extraServerConfig = ''
67
         #
68
               apoc.export.file.enabled=true
         #
69
70
         # };
71
72
         services.postgresql = {
73
           enable = true;
           ensureUsers = [ ];
74
           initialScript = secrets.postgresqlInit;
75
76
         };
77
         /* Boot */
78
79
         boot.loader = {
80
81
           systemd-boot.enable = true;
82
           efi.canTouchEfiVariables = false;
83
            grub.enable = false;
84
         };
85
86
         boot.loader.grub.copyKernels = true;
87
88
         # boot.kernelPatches = let hashes = [
              "sha256-ffrN0y5Us/VJ1v6J0bfkSVD4PCm23wueI0JXjoRcOao="
```

```
"sha256-mh9MgRtLp4+sXjM6GHkAkyyvp5DP3yTx1+d/kJ3+KJQ="
 90
          #
               "sha256-1Jy1uv9x1c06/auY1aHN0SuyOsbIF1IcaK8/PuGmX8Q="
 91
          #
               "sha256-QE3NfIg60RxV528AbpNAcxIdTop1RNm5DgMNPn6AjHQ="
 92
          #
               "sha256-zyL1g2oNiTzYfS5qU0ehduGfzFfi2WIVLCQMg+XGBwE="
 93
          #
              "sha256-f79tof005HBFwLbddR9F0qqLsqGsHqMZM45uYZNcoec="
 95
              "sha256-JD0bi4sjsI9qRpHXxcPlGpiiwXMu2oPm/HJU8iQI1ag="
              "sha256-V+AGk7FGNiidd22ekt1GCvSh3Wo2bRrUWp6qgGmsKPI="
 96
          # ]; in builtins.genList (i: {
 97
             name = "fuse_passthrough ${toString (i+1)}/8";
98
              patch = pkgs.fetchpatch {
99
          #
               url = "https://lore.kernel.org/lkml/20210125153057.3623715-${toString (i+2)}-balsi
100
                sha256 = builtins.elemAt hashes i;
101
          #
          # };
102
          # }) 8;
103
104
          /* Maintenance */
105
106
          time.timeZone = "America/Los_Angeles";
107
108
          services.journald.extraConfig = "MaxRetentionSec=1week";
109
          boot.cleanTmpDir = true;
110
          boot.tmpOnTmpfs = true;
111
112
          systemd = {
113
114
            timers.zbak-snap = {
              wantedBy = [ "timers.target" ];
115
              partOf = [ "zbak-snap.service" ];
116
              timerConfig.OnCalendar = "*:0/15";
117
118
            };
            services.zbak-snap = {
119
120
              serviceConfig.Type = "oneshot";
              path = [ pkgs.zbak ];
121
              script = ''
122
123
                zbak snap zzroot/code --keep 7d24h4f
               '';
124
            };
125
126
            # timers.zbak-send = {
127
              wantedBy = [ "timers.target" ];
128
                partOf = [ "zbak-send.service" ];
129
                timerConfig.OnCalendar = "*:10,40";
130
131
            # };
            # services.zbak-send = {
132
               serviceConfig.Type = "oneshot";
133
               path = [ pkgs.zbak ];
134
                script = ''
135
            #
                  [ $(cat /sys/class/power_supply/BAT0/status) = "Full" ] && \
136
            #
                    zbak send --name ssd500gb \
137
            #
                               --from zzroot/code \
138
            #
            #
                               --to root@100.112.12.44:rpool/code \
139
140
                               --keep 6m4w7d
141
```

```
# };
142
143
          };
144
          /* Security */
145
146
147
          services.openssh.enable = true;
148
          services.gnome.gnome-keyring.enable = true;
149
150
          programs.adb.enable = true;
151
152
          security.sudo = {
153
             enable = true;
154
155
             wheelNeedsPassword = true;
156
          };
          users.mutableUsers = true;
157
158
          users.users.ajanse = {
             isNormalUser = true;
159
             uid = 1000;
160
             group = "users";
161
162
             extraGroups = [
               "wheel"
163
               "lp" # for bluetooth
164
               "video"
165
               "audio"
166
               "vboxusers"
167
               "adbusers"
168
               "libvirtd"
169
170
               "kvm"
               "adbusers"
171
               "docker"
172
               "bluetooth"
173
174
             ];
175
             createHome = true;
             home = "/home/ajanse";
176
             openssh.authorizedKeys.keys = [
177
               "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAILBA0RNuRdCTQgK15gzpqMBHH9XBbkvE2z7orygz67jv" #
178
179
             1;
180
          };
181
182
          users.users.enclave = {
             isNormalUser = true;
183
             uid = 1001;
184
             group = "users";
185
             extraGroups = [ "wheel" ];
186
187
             createHome = true;
             home = "/home/enclave";
188
189
          };
190
          /* Shell */
191
192
193
          users.defaultUserShell = pkgs.fish;
          environment.nathsTolink = [ "/share/zsh" "/share/fish" ]:
```

```
195
          environment.variables.MOZ_X11_EGL = "1";
          environment.variables.EDITOR = "${pkgs.kakoune}/bin/kak";
196
197
          environment.etc."fish/config.fish".text = ''
            set REPO DIR $HOME/school/sp21-s367
198
            set SNAPS DIR $HOME/school/snaps-sp21-s367
199
200
            set GOPATH $HOME/.go
201
202
            alias enter="sudo ip netns exec wguard su - enclave"
203
            alias ls=exa
204
205
          /* Networking */
206
207
208
          networking = {
            hostName = "xps-ajanse";
209
210
            networkmanager.enable = true;
211
            nameservers = pkgs.lib.mkForce [ "193.138.218.74" ]; # mullvad
            iproute2.enable = true;
212
213
            firewall = {
              enable = true;
214
              allowedTCPPorts = [ 80 ];
215
216
              allowedUDPPorts = [ 51820 41641 ];
217
              trustedInterfaces = [ "wg0" ];
218
            };
219
            hosts = {
220
              "192.168.1.249" = [ "BRW707781875760.local" ];
              "172.31.98.1" = [ "aruba.odyssys.net" ];
221
222
              "127.0.0.1" = [ "localhost.dev" "local.metaculus.com" ];
223
            };
          };
224
225
226
          services.tailscale.enable = true;
227
228
          services.rpcbind.enable = true;
229
          services.nfs.server.enable = true;
230
          networking.wireguard.enable = true;
231
232
          networking.wireguard.interfaces.wg0 = secrets.wireguard;
233
          /* Sound */
234
235
          sound.enable = true;
236
237
          hardware.pulseaudio = {
238
            enable = true;
            daemon.config = { flat-volumes = "no"; };
239
240
          };
241
          /* Bluetooth */
242
243
244
          hardware.bluetooth.enable = true;
245
          services.blueman.enable = true;
          programs.dconf.enable = true;
246
```

```
247
          services.dbus.packages = [ pkgs.blueman pkgs.foliate ];
          hardware.pulseaudio = {
248
            extraModules = [ pkgs.pulseaudio-modules-bt ];
249
            package = pkgs.pulseaudioFull;
250
251
          };
252
253
          services.pcscd.enable = true;
254
          services.udev.packages = [
255
            pkgs.yubikey-personalization
            pkgs.libu2f-host
256
257
            pkgs.android-udev-rules
          1;
258
259
260
          # systemd.user.sockets.gpg-agent-ssh = {
              wantedBy = [ "sockets.target" ];
261
             listenStreams = [ "%t/gnupg/S.gpg-agent.ssh" ];
262
              socketConfig = {
263
                FileDescriptorName = "ssh";
264
          #
                Service = "gpg-agent.service";
265
          #
                SocketMode = "0600";
266
                DirectoryMode = "0700";
267
          #
268
          # };
269
          # };
270
          # environment.shellInit = ''
271
              export GPG TTY="$(tty)"
272
273
             ${pkgs.gnupg}/bin/gpg-connect-agent /bye
              export SSH_AUTH_SOCK="/run/user/$UID/gnupg/S.gpg-agent.ssh"
274
          # '';
275
276
277
          # services.yubikey-agent.enable = true;
278
          programs = {
279
280
            ssh.startAgent = false;
281
            gnupg.agent.enable = true;
             gnupg.agent.enableSSHSupport = true;
282
             gnupg.agent.pinentryFlavor = "gtk2";
283
284
          };
285
          /* User Interface */
286
287
          services.xserver = {
288
289
            enable = true;
290
            displayManager.gdm = {
              enable = true;
291
292
              wayland = true;
293
            };
            desktopManager.gnome.enable = true;
294
295
          };
296
          programs.light.enable = true;
297
298
           /* II-DDT */
```

```
\ . UTDLT .\
277
300
301
          console = {
302
            earlySetup = true;
            font = "sun12x22";
303
304
            colors = theme.colors16;
305
          };
306
          services.xserver.dpi = 192;
          services.xserver.displayManager.sessionCommands = ''
307
            ${pkgs.xorg.xrdb}/bin/xrdb -merge <<EOF</pre>
308
309
               Xcursor.theme: Vanilla-DMZ
              Xcursor.size: 32
310
            EOF
311
            ${pkgs.xorg.xrandr}/bin/xrandr --output eDP-1 --scale 1.25x1.25
312
313
          environment.variables = {
314
            GDK SCALE = "1.5";
315
            GDK DPI SCALE = "0.75";
316
317
            QT_AUTO_SCREEN_SCALE_FACTOR = "1";
318
          };
319
          /* Peripherials */
320
321
322
          services.xserver.xkbOptions = "compose:caps";
          services.xserver.libinput = {
323
324
            enable = true;
325
            touchpad = {
326
               tappingDragLock = false;
327
               naturalScrolling = true;
328
            };
329
          };
330
           systemd.user.services.xinput-config = {
            description = "configure xinput for xps";
331
332
            wantedBy = [ "graphical-session.target" ];
333
             partOf = [ "graphical-session.target" ];
             script = ''
334
               ${pkgs.xorg.xinput}/bin/xinput set-prop "SynPS/2 Synaptics TouchPad" "libinput Accel
335
               ${pkgs.xorg.xinput}/bin/xinput set-prop "SynPS/2 Synaptics TouchPad" "libinput Tappi
336
               ${pkgs.xorg.xinput}/bin/xinput set-prop "SysPS/2 Synaptics TouchPad" "libinput Disab
337
             '';
338
339
          };
340
341
          /* Printing */
342
343
          services.printing.enable = true;
344
          services.printing.drivers = [ pkgs.brlaser pkgs.mfcl2740dwlpr pkgs.mfcl2740dwcupswrapper
          services.avahi.enable = true;
345
          services.avahi.nssmdns = true;
346
347
          /* Default Fonts */
348
349
350
          fonts = {
351
             fontDir.enable = true;
```

```
352
            fonts = with pkgs;[
353
              corefonts
354
              inconsolata
355
              terminus_font
              dejavu_fonts
356
357
              ubuntu_font_family
358
              source-code-pro
              source-sans-pro
359
360
              source-serif-pro
361
              roboto-mono
              roboto
362
363
              overpass
364
              libre-baskerville
              font-awesome
365
366
              julia-mono
            ];
367
368
          };
369
370
          /* Virtualization */
371
372
          boot.binfmt.emulatedSystems = [ "aarch64-linux" ];
373
374
          # virtualisation.libvirtd.enable = true;
375
          # virtualisation.cri-o = {
          # enable = true;
376
377
              storageDriver = "vfs";
378
          # };
379
          # virtualisation.virtualbox.host = {
380
          # enable = true;
381
          # enableExtensionPack = false; # takes a long time to build
382
          # };
383
          # virtualisation.docker = {
384
          # enable = true;
          # storageDriver = "zfs";
385
386
          # };
387
388
          system.stateVersion = "19.03";
389
        }
```