



Literate Configuration Series

NixOS

by


Kuzey Koç

Contents

1 Imports	4
2 Global Variables	4
2.1 Package Overrides	4
2.1.1 ncmtcpp with visualizer	4
3 Main Configuration	5
4 Bootloader	5
4.1 Grub	5
5 External Home Directory	6
6 Hardware	6
7 Automatic Updates	7
8 Networking	7
8.1 Network Manager	8
8.2 Wireless	8
8.3 Firewall	8
8.4 NAT	8
8.5 DNS	9
9 Localization	9
9.1 Timezone	9
9.2 Internationalisation	9
9.3 Console Keymap	10
10 Users	10
11 Nix	11
11.1 Flakes	11
11.2 Unfree Packages	11
11.3 Package Overlays	11
11.3.1 mpv	11

11.3.2 weechat	12
12 Theme	12
13 Environment Variables	12
14 Filesystem Check	13
15 Packages	13
15.1 Editors	13
15.2 Virtualization	13
15.3 Wayland	14
15.4 Gaming	14
15.5 Music Production	14
15.6 Compiling Tools	16
15.7 \LaTeX	16
15.8 Development	17
15.8.1 Frontend	17
15.8.2 Backend	17
15.8.3 Mobile	17
15.8.4 Gamedev	18
15.9 DevOps	18
15.9.1 Kubernetes	18
15.9.2 Local Stack	19
15.10 Doom Emacs Specific	19
15.10.1 Dirvish Deps	19
15.10.2 LSP Deps	19
15.10.3 Other	20
15.11 Communication	20
15.12 Multimedia	21
15.13 Browsers	21
15.14 GUI	21
15.15 Suckless	22
15.16 Misc	22
16 Fonts	25

17 XDG	25
18 Virtualization	26
18.1 VirtualBox	26
18.2 Podman	26
18.3 Docker	26
18.4 Libvirt	27
18.5 Waydroid	27
19 Services	28
19.1 xorg	28
19.2 spice	28
19.3 ssh	28
19.4 cron	29
19.5 opensnitch	29
19.6 udisks2	30
19.7 gvfs	30
19.8 pipewire	30
19.9 ly	30
19.10 dnscrypt	30

 **WARNING:** This article is under construction. Please come back later

You can get the PDF version of this article [here](#)

1 Imports

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

2 Global Variables

Global variable definition starts with

```
let
```

Now here I define my important globals like username, hostname, static ip etc.

```
USERNAME = "savolla";  
HOSTNAME = "xkarna";  
HOME = "/home/savolla";  
ETHERNET_INTERFACE_NAME = "ens18";  
WIRELESS_INTERFACE_NAME = "wlp2s0";  
STATIC_IP_ADDRESS = "192.168.1.108";  
DEFAULT_GATEWAY = "192.168.1.1";
```

2.1 Package Overrides

2.1.1 ncmpcpp with visualizer

I use *ncmpcpp* for listening my music collection. It has a nice visualizer but the default nixos package does not include it for some reason. I override that package with visualizer support

```
ncmpcpp = pkgs.ncmpcpp.override {  
  visualizerSupport = true;  
  clockSupport = true;  
}; # ncmpcpp with visualizer
```

Global variables definition ends with

```
in
```

3 Main Configuration

The main configuration starts with

```
{
```

Like every nixos configuration it must import a *hardware-configuration.nix* like this

```
imports = [ ./hardware-configuration.nix ];
```

4 Bootloader

I was using *libvirt* for virtualization before *proxmox*. These are some kernel modules for nested virtualization (kubernetes). I commented them out since I'm now using Talos in my proxmox homelab.

I also use real-time kernel for better I/O performance on my system. It works well for gaming/music production etc.

```
boot = {
  # extraModprobeConfig = ''
  #   options kvm_intel nested=1
  #   options kvm_intel emulate_invalid_guest_state=0
  #   options kvm ignore_msrs=1
  # '';

  kernelPackages = pkgs.linuxPackages-rt_latest; # linux realtime
  ↪ kernel
```

4.1 Grub

I use *grub* instead *systemd-boot* which is the default in nixos. In order to use it I disabled *systemd-boot* completely and set some *grub* specific options. *saved* means that grub will remember your last choice on the next restart.

You can also limit the amount of configuration displayed in the grub menu with *configurationLimit* option. But I disabled that. with *OSProber* it will detect other OSes in your drive.

I also enable EFI support.

```
loader = {
  timeout = 5; # seconds
  systemd-boot.enable = false;
  grub = {
    enable = true;
    default = "saved";
    device = "nodev";
```



```

    # configurationLimit = 10;
    useOSProber = true;
    efiSupport = true;
    efiInstallAsRemovable =
        true; # otherwise /boot/EFI/BOOT/BOOTX64.EFI isn't
        ↪ generated
    extraEntriesBeforeNixOS = true;
};
    efi = { efiSysMountPoint = "/boot/efi"; };
};
};

```

5 External Home Directory

I keep my encrypted home directory separate in 1TB external SSD. When I boot my nixos machine I want it to be mounted under `/home/savolla`. Since it is an encrypted drive I also needed to setup some decryption operations.

```

environment.etc.crypttab.text = ''
    homecrypt UUID=5ba30f2e-b06a-4588-b594-70fb46ef16d9 none
    ↪ luks,timeout=120
'';

fileSystems."/home/savolla" = {
    device = "/dev/mapper/homecrypt";
    fsType = "ext4";
    options = [ "x-systemd.device-timeout=10" ];
    neededForBoot = false;
};

# Helpful to ensure USB storage is supported
boot.initrd.kernelModules = [ "usb_storage" ];
boot.kernelModules = [ "usb_storage" ];

```

This `timeout` attribute doesn't work. I'm still experimenting with these options. So it will probably change in the future.

6 Hardware

I enable 3D graphics support and Bluetooth support with the following

```

hardware = {
    graphics = {
        enable = true; # enable 3d acceleration
    }
}

```

```

    enable32Bit = true; # for older games
};
bluetooth = {
    enable = true; # enables support for Bluetooth
    powerOnBoot = true; # powers up the default BL
};
};

```

7 Automatic Updates

I generally don't prefer this but I keep it in my config anyway.

```

# automatic update
system = {
    autoUpgrade = {
        enable = false;
        dates = "weekly";
    };
};

```

8 Networking

My networking setup is a little bit messed up. I use **DNSEncrypt** to bypass censorship and other ISP shenanigans. So my dns is tied to DNSEncrypt a little bit..

```

networking = {
    hostName = "${HOSTNAME}";
    defaultGateway = "${DEFAULT_GATEWAY}";
    interfaces = {
        enp4s0 = {
            ipv4.addresses = [{
                address = "${STATIC_IP_ADDRESS}";
                prefixLength = 24;
            }];
        };
    };
    nameservers = [
        "127.0.0.1"
        ":::1"
    ]; # dnscrypt requires this. see
    ↪ https://nixos.wiki/wiki/Encrypted\_DNS

```


8.1 Network Manager

```
networkmanager = {  
  enable = true;  
  dns =  
    "none"; # prevent network manager from overriding dns settings  
    ↪ because dnscrypt will handle this part  
  unmanaged = [  
    "interface-name:ve-*"  
  ]; # for i2p container. network manager should not touch this  
};
```

8.2 Wireless

I don't have a laptop (yet) so I disable wireless option completely

```
# wireless.enable = true; # Enables wireless support via  
↪ wpa_supplicant.
```

8.3 Firewall

I sometimes self-host services in my home network and need a way to access these services from other devices. I use the following firewall settings. Currently I don't host anything but I keep these settings for the potential future use

```
firewall = {  
  enable = true; # always keep enabled  
  
  # open ports in the firewall.  
  allowedTCPPorts = [  
    # 3000  
  ];  
  allowedUDPPorts = [  
    # 9000  
  ];  
};
```

8.4 NAT

I need this because I sometimes expose in-vm services to my home network and I need to translate those ip addresses

```
nat = {  
  enable = true;  
  internalInterfaces = [ "ve-+" ];
```

```
externalInterface = "${ETHERNET_INTERFACE_NAME}";  
};
```

8.5 DNS

In nixos there is no `/etc/hosts` file. We have the following instead

```
extraHosts = ''  
    192.168.1.105 api.crc.testing  
    192.168.1.105 console-openshift-console.apps-crc.testing  
    192.168.1.105 oauth-openshift.apps-crc.testing  
    192.168.1.105 java-demo-java-demo.apps-crc.testing  
    192.168.1.105 argocd-sample-server-java-demo.apps-crc.testing  
    '';  
};
```

9 Localization

9.1 Timezone

```
time.timeZone = "Europe/Istanbul";
```

9.2 Internationalisation

```
i18n = {  
    defaultLocale = "en_US.UTF-8";  
    extraLocaleSettings = {  
        LC_ADDRESS = "tr_TR.UTF-8";  
        LC_IDENTIFICATION = "tr_TR.UTF-8";  
        LC_MEASUREMENT = "tr_TR.UTF-8";  
        LC_MONETARY = "tr_TR.UTF-8";  
        LC_NAME = "tr_TR.UTF-8";  
        LC_NUMERIC = "tr_TR.UTF-8";  
        LC_PAPER = "tr_TR.UTF-8";  
        LC_TELEPHONE = "tr_TR.UTF-8";  
        LC_TIME = "tr_TR.UTF-8";  
    };  
};
```

9.3 Console Keymap

```
console.keyMap = "trq";
```

10 Users

NixOS displays “message of the day” on console login. I disable it with

```
users = {  
  motdFile = null; # disable message of the day on tty
```

Setting some basic options for my user *savolla*

```
users."${USERNAME}" = {  
  isNormalUser = true;  
  home = "/home/savolla";  
  shell = pkgs.zsh;  
  description = "${USERNAME}";
```

Adding my user to important groups

```
extraGroups = [  
  "networkmanager" # wifi etc.  
  "wheel" # sudo  
  "input" # xorg  
  "video" # xorg  
  "audio" # pipewire  
  "libvirt" # virtualization  
  "docker" # run docker commands without sudo  
  "podman" # run podman commands without sudo  
  "vboxusers" # vbox guest additions and clipboard share  
  "kvm" # android emulation with kvm (faster)  
  "adbusers" # interact with android and emulators with adb  
  "systemd-journal" # watch system logs with `journalctl -f` without  
    ↪ sudo password  
];
```

I once tried to run *gparted* on wayland and I came with the following solution but it doesn't work anymore.. probably will delete it later..

```
packages = with pkgs;  
[  
  gparted # install this as a user package to prevent errors in  
    ↪ wayland  
];  
};
```



```
};
```

11 Nix

11.1 Flakes

I'm enabling flakes support

```
# nix config
nix = {
  settings = {
    experimental-features = [ "flakes" "nix-command" ];
  };
};
```

11.2 Unfree Packages

Nix does not allow unfree packages by default. You need to enable them explicitly

```
nixpkgs = {
  config = {
    allowUnfree = true;
    allowUnsupportedSystem = true;
    permittedInsecurePackages = [ "electron-27.3.11" ];
  };
};
```

11.3 Package Overlays

11.3.1 mpv

I usually like to watch YouTube videos using *mpv*. **quality-menu** lets me switch between video and audio qualities easily. **quack** on the other hand temporarily reduces video quality on video skimming.

```
overlays = [
  (self: super: {
    mpv = super.mpv.override {
      scripts = [
        # select youtube video quality from the player
        self.mpvScripts.quality-menu
        # temporarily reduce video and audio quality when skipping
        self.mpvScripts.quack
      ];
    };
  });
];
```

11.3.2 weechat

I use some addons in my *weechat* config such as **url_{hint}**, **colorize_{nicks}** and **weechat-notify-send**

```
weechat = super.weechat.override {
  configure = { availablePlugins, ... }: {
    scripts = with super.weechatScripts; [
      url_hint
      colorize_nicks
      weechat-notify-send
    ];
  };
}
```

12 Theme

Enabling global dark theme without a proper DE is always painful. In NixOS it is pain in the ass..

Qt Apps

```
qt.style = "adwaita-dark";
```

GTK Apps

```
# set variables for all users (including root)
environment.sessionVariables = {
  GTK_THEME = "Adwaita:dark"; # make sudo applications use dark
  ↪ theme
};
```

13 Environment Variables

I apply different settings in my *.xprofile* and *.profile* depending on the current nixos specialisation. So I use an env. variable *CURRENT_NIXOS_SPECIALISATION* for this. The default value is “vanila”. I also have “nvidia” and “musician”. See **Specialisations** section below

```
environment = {
  stub-ld.enable = true;
  variables = {
    CURRENT_NIXOS_SPECIALISATION = "vanila";
  };
};
```

14 Filesystem Check

I should probably enable file system checks again. But it sometimes takes too much time on boot.

```
fileSystems = {  
  "/" .noCheck = true;  
  "/boot/efi" .noCheck = true;  
};
```

15 Packages

15.1 Editors

```
environment.systemPackages = with pkgs; [  
  
  # editors  
  vim # fallback text editor  
  neovim # better vim  
  # emacs-pgtk # transparency works in wayland  
  emacs-gtk # true transparency works with this one on xorg  
  vscode # just in case ide  
  jetbrains.webstorm # vscode sucks sometimes
```

15.2 Virtualization

```
# virtualization  
vagrant # declarative virtual machines  
quickemu # installed for installing macos sonoma (for react-native  
↳ dev)  
quickgui # gui for quickemu  
guestfs-tools # bunch of tools with virt-sparsify  
virglrenderer # allows a qemu guest to use the host GPU for  
↳ accelerated 3D rendering  
virt-viewer # viewer for qemu  
spice-vdagent # shared clipboard between qemu guests and host  
qemu # all supported architectures like arm, mips, powerpc etc.  
distrobox # run other distros using docker  
edk2 # for osx-kvm (tianocore uefi)  
edk2-uefi-shell # for osx-kvm (tianocore uefi)  
OVMFFull  
virtiofsd # share file system between host and guests
```


15.3 Wayland

```
# wayland related
waybar # status bar for wayland
rofi-wayland # application launcher
wl-clipboard # wayland clipboard
slurp # region select. combine it with grim to select region for
↳ screenshot
grim # screenshot utility
gammastep # redshift/sct alternative for wayland
swaybg # set wallpapers in wayland
ydotool # xdotool for wayland
```

15.4 Gaming

I don't play games anymore but I once built a sane gaming setup on NixOS. There you go if you need it.

```
# gaming stuff
lutris # install and launch windows and linux games
mangohud # display fps, temperature etc.
cabextract # installed this to install Age of Empires Online
↳ (prefix that was created by Kron4ek)
bottles-unwrapped # powerful wine thing
unigine-valley # test GPU drivers
protonup # proton-ge
wineWowPackages.staging # wine staging version
winetricks # install windows dlls with this
retroarchFull # retroarch + cores
ryujinx # switch emulator
antimicrox # map ps4 controller keys to nintendo switch and others
nsz # .nsz to .nsp nintendo switch game convertor for ryujinx
↳ emulator
input-remapper # map mouse movement to joystick. (play ryujinx
↳ games with mouse and keyboard)
qjoypad # play ryujinx games with mouse and keyboard
sc-controller # emulate joysticks on linux (to play switch games
↳ using mouse and keyboard)
```

15.5 Music Production

I was using NixOS for music production but since I started using Proxmox I migrated to Windows 10.

```
## music

# daw
reaper
sonic-pi
faust # dsp language
faust2jaqt # faust dependency
supercollider_scel # supercollider with emacs extension scel

# guitar stuff
gxplugins-lv2 # guitar amps, pedals, effects
neural-amp-modeler-lv2 # you'll download guitar tones for this
    ↳ below
guitarix # a virtual guitar amplifier for Linux running with JACK
tuxguitar # guitar pro for linux
musescore # sheet happens
tonelib-jam # 3d tab editor (paid)
tonelib-gfx # good guitar amp
tonelib-metal # all in one guitar rig
proteus # NAM

# VST (open source)
# distrho # not in packages anymore
calf # high quality music production plugins and vsts
eq10q
lsp-plugins # collection of open-source audio plugins
x42-plugins # collection of lv2 plugins by Robin Gareus
x42-gmsynth
dragonfly-reverb
FIL-plugins
geonkick

# utils
wineasio # for playing Rocksmith 2014 Remastered
alsa-scarlett-gui # focusrite scarlett solo gui
yabridge # use windows vsts on linux wine is requirement here
yabridgectl # yabridge control utility
scarlett2 # update firmware of focusrite scarlett devices
tenacity # audacity fork
klick # cli metronom
qpwgraph
qjackctl # reduce latency
helvum # modern jack ui
```

15.6 Compiling Tools

NixOS is not pretty when it comes to “compiling” from source but I was using *crosstool-ng* to build toolchains for BeagleBoneBlack and Raspberry Pi. These are the requirements

```
# compiling
stdenv # build-essentials
help2man # for crosstool-ng dep
gnumake # make for all
autoconf # for crosstool-ng dep
audit # for crosstool-ng dep
automake # for crosstool-ng dep
gcc # for crosstool-ng dep
flex # for crosstool-ng dep
file
bison
ncurses
freetype # to be able to compile suckless utils
```

15.7 \LaTeX

I use Emacs’s org-mode feature A LOT for writing and exporting my documents to various formats like pdf, html, slides etc. I have my own pdf export template. The following packages are the dependencies for this template

```
# latex
texliveFull # full latex environment for pdf exports (doom emacs)
texlivePackages.booktabs # Publication quality tables in LaTeX
texlivePackages.fvextra # Extensions and patches for fancyvrb (for
↳ syntax highlighting)
texlivePackages.xcolor # Driver-independent color extensions for
↳ LaTeX and pdfLaTeX
texlivePackages.fontspec # Advanced font selection in XeLaTeX and
↳ LuaLaTeX
texlivePackages.microtype # Subliminal refinements towards
↳ typographical perfection
texlivePackages.titlesec # Select alternative section titles
texlivePackages.minted # syntax highlighting
latexminted # just in case
texlivePackages.librebaskerville # main font
libre-baskerville # above font does not work
texlivePackages.plex # sans and mono fonts
```


15.8 Development

```
# development
gh # github cli
tts # coqui-ai TTS (works with cpu)
drawio # draw rldb, uml diagrams etc.
# android-studio # for testing react native apps in emulators
lua # other python
ruby # language
pkg-config # needed for building ruby files
shfmt # doom emacs's dep for bash file formatter to work
spring-boot-cli # vite for java (spring boot)
```

15.8.1 Frontend

```
# frontend development
nodePackages.prettier # prettier code formatter for js/ts
figma-linux # frontend development
firefox-devedition # bin edition because nix tries to compile
↳ `firefox-devedition` and fails
pnpm # better npm?
jpegoptim # optimize jpeg
optipng # optimize png
libwebp # convert images to webp
# responsively-app # responsive design helper
```

15.8.2 Backend

```
# backend development
insomnia # make api calls easily (postman alternative)
dbeaver-bin # database awesomeness
```

15.8.3 Mobile

```
# mobile development
genymotion # for testing react native apps in emulators
react-native-debugger # official react-native debugger
sdkmanager # manage android sdk versions
```

```
jdk # for JAVA installation. needed for android sdk and other apps  
↳ (this installs the latest version of jdk)  
nodePackages_latest.eas-cli # build expo apk and dmg  
bundletool # convert .abb to .apk
```

15.8.4 Gamedev

```
# game development  
love # awesome 2d game engine written in lua  
godot_4 # 3d and 2d game engine
```

15.9 DevOps

```
# devops  
camunda-modeler # business modeling tool  
eclipses.eclipse-java # for camunda and spring boot  
terraform # iac  
ansible # cac  
awscli2 # aws cli tools  
argocd # control argocd instances from commandline
```

15.9.1 Kubernetes

```
# minikube # kubernetes testing and learning environment  
openshift # kubernetes for stake holders  
crc # locally install openshift  
kubernetes-helm # kubernetes package manager  
talosctl # control talos clusters  
k9s # lazy kubernetes  
kubectl # control your k8s cluster from your local machine  
kubernetes-helm # kubernetes package manager  
docker-compose # poor man's kubernetes
```

15.9.2 Local Stack

```
localstack # local aws
terraform-local # use terraform with localstack
```

15.10 Doom Emacs Specific

I use Doom Emacs for almost anything. programming, writing, blogging, terminal multiplexing and even music listening. But to do all those things it needs some dependencies.

15.10.1 Dirvish Deps

```
# doom emacs dirvish
vips # display images in emacs buffer
poppler-utils # view pdfs first page
ffmpegthumbnailer # extract thumbnails from videos
mediainfo # for displaying audio metadata
epub-thumbnailer # for displaying epub covers
```

15.10.2 LSP Deps

1. C/C++

```
# c/c++ (clang lsp)
ccls
glslang
clang
clang-tools
```

2. Shell

```
# shell
shellcheck
```

3. Rust

```
## rust
cargo
rustc
rust-analyzer
```


4. Nix

```
# nix
nixfmt-classic # formatting
nil # language server
```

5. Python

```
python311Packages.isort
python311Packages.pytest
python311Packages.nose2
python311Packages.nose2pytest
python313Packages.diagrams
pipenv
black # python-mode code formatter
python313Packages.pyflakes # python-mode import reordering
python313Packages.nose2 # python-mode tests
```

6. Web HTML, CSS, JS etc.

```
# web
html-tidy
stylelint
jsbeautifier
rustywind # for tailwind lsp
```

7. Docker

```
dockfmt # docker file formatting
```

15.10.3 Other

```
ispell # emacs spell checking dep
ripgrep # doom emacs dep
```

15.11 Communication

```
ferdium # communication
gajim # xmpp client for linux
nicotine-plus # pure piracy
```

15.12 Multimedia

```
jamesdsp # equalizer for pipewire  
mpd # music player daemon  
mpc # control mpd from terminal  
ncmcpp # custom ncmcpp with visualizer. see let/in on top
```

15.13 Browsers

```
firefox # normal browser  
chromium # ungoogled chrome (needed for react-native debugger)  
tor-browser # just in case  
librewolf # paranoid browser
```

15.14 GUI

```
kdePackages.kdenlive # open source video editing software  
# anydesk # proprietary remote control  
libreoffice-qt6 # open .docx  
pcmanfm # file manager  
blender # 3d design  
wireshark # network analyzer  
xournalpp # draw shapes using your wacom tablet  
zathura # pdf reader  
rustdesk # open source remote control  
flameshot # screenshot utility for xorg  
anki # the best spaced repetition tool  
bleachbit # system cleanup  
  
# security  
keepassxc # password manager  
cryptomator  
  
# image processing  
gimp-with-plugins # open source photoshop  
krita # digital art in linux? also comfyui integration using  
↳ comfyui plugins  
inkscape-with-extensions # svg and logo design
```

15.15 Suckless

```
# suckless

# st
(st.overrideAttrs { src = ../../suckless/st-0.9.2; })

# slstatus
(slstatus.overrideAttrs {src = ../../suckless/slstatus;})

# dmenu
(dmenu.overrideAttrs { src = ../../suckless/dmenu-5.3; })
```

15.16 Misc

```
wget # download things
gcolor3 # color palettes for web dev
# simplescreenrecorder # screen recorder for xorg (disabled because
↳ using ffmpeg)
screenkey # show key presses on screen (screencast)
xcolor # color picker for xorg
appimage-run # run appimages on nixos
libcaca # ascii art viewer
wipe # securely wipe directories and files on hdd/ssd
git # version control
unrar # non-free but needed
koreader # awesome book reader
dunst # notification daemon
opensnitch-ui # enable interactive notifications for application
↳ firewall
gowall # change colorschemes of any wallpaper
peek # record desktop gifs
graphviz # org-mode graph generation dependency
plantuml-c4 # org-mode graph generation
kitty # fallback terminal
# teams-for-linux # microsoft teams for linux (unofficial)
pinentry-gtk2 # password prompt for gnupg
ffmpeg-full # needed for ncmpcpp cover art display and bunch of
↳ other things
libnotify # dunst dep
xcalib # invert colors of x
xorg.xev # find keysims
```

```
nixos-generators # generate various images from nixos config
↳ (qcow2)
dmg2img # convert apple's disk images to .img files. (needed for
↳ installing hackintosh on qemu)
pandoc # emacs's markdown compiler and org-mode dep
socat # serial communication with quickemy headless hosts without
↳ ssh
btrfs-progs # you need this for nixos os-prober detect other oses
↳ like fedora which uses btrfs by default
arp-scan # scan local ips
iptraf-ng # watch network traffic in tui
fd # doom emacs dep
logseq # note taking tool
protonvpn-cli_2 # vpn (this does not work anymore)
protonvpn-cli # vpn
lilypond-unstable-with-fonts # music notation (for doom emacs
↳ org-mode)
inetutils # for whois command
binwalk # check files
jq # needed for my adaptive bluelight filter adjuster
ranger # tui file manager
zip # archiving utility
p7zip # great archiving tool
# ollama-cuda # use local llms. installing this via systemPackages
↳ to set "models" dir to my home
tree-sitter # parser for programming
lazygit # git but fast
cmake # installed for vterm to compile
pavucontrol # pipewire buffer size and latency settings can be done
↳ from there
libtool # installed for compiling vterm
undollar # you copy and paste code from internet? you simply need
↳ it
gperftools # improve memory allocation performance for CPU (need
↳ for ai apps use CPU instead of GPU)
gtk3 # emacs requires this
# nvidia-container-toolkit # needed for docker use nvidia
w3m # image display for terminal
gdu # scan storage for size
tmux # life saver
smartmontools # check health of ssd drives
gsmartcontrol # check harddrive health
htop # better system monitor
yazi # new ranger
xorg.xinit # for startx command to work
xorg.libxcb # fix steam "glXChooseVisual" error
picom # xorg compositor
```


xsel # x clipboard
xdotool # simulate keyboard and mouse events
scrcpy # mirror android phone to pc
nsxiv # image viewer
yt-dlp # youtube video downloader + you can watch videos from mpv
↳ using this utility
tshark # scan local network
colordiff # to display colored output (installed for tshark)
unp # unpack any archive
mermaid-cli # for org-babel mermaid diagrams support (doom emacs)
xorg.libxshmfence # appimage-run requires it for some appimages
↳ like Mechvibes
libgen-cli # download books from libgen
udiskie # auto mount hotplugged block devices
ntfs3g # make udiskie mount NTFS partitions without problems
lxappearance # style gtk applications
nodejs # for emacs to install lsp packages
mpv # awesome media player (overlaid!)
syncthing # sync data between devices
imagemagick # for mp4 to gif conversion and other stuff
pulseaudio # installed for pactl to work. was trying to record
↳ screen with ffmpeg and pipewire. needed pactl
uv # vital python package. solves all those python version and
↳ dependency problems
unclutter-xfixes # hide mouse cursor after a time period
tldr # too long didn't read the manual
tabbed # suckless tabbed
emacsPackages.nov # to make nov.el work
xsct # protect your eyes (blue light filter) (disabled because
↳ using redshift)
fftw # fastest fourier transform for ncpcpp
vlc # play dvds .VOB
hugo # generate static site
feh # set wallpapers
mp3blaster # auto tag mp3 files using mp3tag tool
zstd # extract .zst files
isoimagewriter # balena etcher alternative
neofetch # system info
weechat # irc client (overlaid!)
sysstat # get system statistics (used for tmux status bar cpu
↳ usage)
eza # ls alternative
starship # cross shell (very cool)
alsa-utils # for amixer and setting volume via scripts and
↳ terminal
busybox # bunch of utilities (need)
sxhkd # simple x hotkey daemon

```

adw-gtk3 # for adwaita-dark theme
adwaita-qt # make qt applications use dark theme
adwaita-icon-theme # pretty icons (objective)
newsboat # rss/atom reader
transmission_4-gtk # torrent application
xd # i2p torrenting
unzip # mandatory
hdparm # remove disks safely from terminal
scrot # for emacs's org-mode screen shot capability
bat # cat but better
fzf # fuzzy finder for terminal
xclip # clipboard for xorg
sxhkd # simple x11 hotkey daemon
# lxqt.lxqt-policykit # authentication agent
lxde.lxsession # session manager
xorg.xf86videoqxl # trying to improve scaling in spice
gnupg # encryption and stuff
stress # simulate high cpu load for testing
];

```

16 Fonts

I use Fira Code in my pdf documents and iosevka in terminal

```

# fonts
fonts.packages = with pkgs; [
  nerd-fonts.fira-code
  nerd-fonts.iosevka-term
  nerd-fonts.iosevka
];

```

17 XDG

I'm having issues with xdg even though I set the required settings below (needs more work)

```

xdg.portal = {
  enable = true;
  extraPortals = [ pkgs.xdg-desktop-portal-gtk ];
};

```

18 Virtualization

I stopped using these virtualization settings after I started using Proxmox as my type 1 hypervisor. But I'll keep these setting for future reference

18.1 VirtualBox

```
virtualbox = { # virtualbox cannot be built with linux-rt kernel.  
  ↪ hence disabled  
  host = {  
    enable = true;  
  };  
  guest = {  
    enable = true;  
    vboxsf = true;  
    dragAndDrop = true;  
    clipboard = true;  
  };  
};
```

18.2 Podman

```
podman = {  
  enable = true;  
  # dockerCompat = true;  
};
```

18.3 Docker

I change my docker data root directory to prevent bloating my root directory.

```
docker = {  
  enable = true;  
  enableOnBoot = false;  
  daemon.settings = { data-root = "${HOME}/resource/docker"; };  
  autoPrune = {  
    enable = true;  
    dates = "weekly";  
  };  
};
```

I was using rootless docker but I faced issues when running containers that require Nvidia drivers. So I disabled this feature on my side.

```

rootless = {
  enable = true;
  setSocketVariable = false;
  daemon.settings = {
    runtimes = {
      nvidia = {
        path =
          ↪ "${pkgs.nvidia-docker}/bin/nvidia-container-runtime";
      };
    };
  };
};

```

18.4 Libvirt

```

libvirtd = {
  enable = true;
  qemu = {
    package = pkgs.qemu_kvm;
    runAsRoot = true;
    swtpm.enable = true;

    # enable file share between host/guest
    vhostUserPackages = with pkgs; [ virtiofsd ];

    ovmf = {
      enable = true;
      packages = [
        (pkgs.OVMF.override {
          secureBoot = true;
          tpmSupport = true;
        }).fd
      ];
    };
  };
};

```

18.5 Waydroid

Android emulation on Linux? Sounds cool!

```
waydroid.enable = true;
```


19 Services

19.1 xorg

I use NixOS as a Proxmox vm. I enable “qxl” for SPICE. You don’t need this if you’re running NixOS on baremetal

```
# enable xorg
xserver = {
  enable = true;
  xkb = { layout = "tr"; };
  videoDrivers = [ "qxl" ]; # increase SPICE performance

# display manager
displayManager.lightdm.enable = true;

# window manager
windowManager = {
  dwm = {
    enable = true;
    package = pkgs.dwm.overrideAttrs {
      src = ../../suckless/dwm-6.5;
    };
  };
};
};
```

19.2 spice

Since I use NixOS as a Proxmox vm, I sometimes connect to it via SPICE client *virt-viewer*. These services increase performance in the viewer and also let me share clipboard etc.

```
spiceUSBRedirection.enable = true;
spice-vdagentd.enable = true;
qemuGuest.enable = true;
```

19.3 ssh

```
openssh = {
  enable = true;
  startWhenNeeded = true;
};
```

19.4 cron

```
cron = {
  enable = true;
  systemCronJobs = [
    # keep your ssd healthy by trimming it hourly (redhat
    ↪ suggestion)
    "0 * * * * root fstrim -av >> /var/log/fstrim.log 2>&1"
  ];
};
```

19.5 opensnitch

I use **opensnitch** as my **application firewall**. It is the closest alternative to **Simplewall** on windows. I also set some default rules below

```
opensnitch = {
  enable = true;
  rules = {
    systemd-timesyncd = {
      name = "systemd-timesyncd";
      enabled = true;
      action = "allow";
      duration = "always";
      operator = {
        type = "simple";
        sensitive = false;
        operand = "process.path";
        data = "${lib.getBin
        ↪ pkgs.systemd}/lib/systemd/systemd-timesyncd";
      };
    };
    systemd-resolved = {
      name = "systemd-resolved";
      enabled = true;
      action = "allow";
      duration = "always";
      operator = {
        type = "simple";
        sensitive = false;
        operand = "process.path";
        data = "${lib.getBin
        ↪ pkgs.systemd}/lib/systemd/systemd-resolved";
      };
    };
  };
};
```

```
};
```

19.6 udisks2

Mount disks without sudo (requires udiskie)

```
udisks2 = {  
    enable = true;  
};
```

19.7 gvfs

enable android file system mount in pcmanfm

```
gvfs = {  
    enable = true;  
};
```

19.8 pipewire

```
pipewire = {  
    enable = true;  
    alsa.enable = true;  
    alsa.support32Bit = true;  
    pulse.enable = true;  
    jack.enable = true;  
};
```

19.9 ly

I used *ly* as my tui login manager but it does a **horrible** job in managing my session. I replaced it with *lightdm* but I'll leave it here for future reference

```
displayManager = {  
    ly = {  
        enable = true;  
    };  
};
```

19.10 dnscrypt

I encrypt my dns instead of using slow VPNs. see [dnscrypt-proxy](#) for more information

```

dnscrypt-proxy2 = {
  enable = true;
  settings = {
    ipv6_servers = true;
    require_dnssec = true;
    # Add this to test if dnscrypt-proxy is actually used to
    ↪ resolve DNS requests
    query_log.file = "/var/log/dnscrypt-proxy/query.log";
    sources.public-resolvers = {
      urls = [
        ↪ "https://raw.githubusercontent.com/DNSCrypt/dnscrypt-resolvers/
        ↪ "https://download.dnscrypt.info/resolvers-list/v3/public-resolv
      ];
    cache_file = "/var/cache/dnscrypt-proxy/public-resolvers.md";
    minisign_key =
      "RWQf6LRCGA9i53mlyec04IzT51TGPpvWucNSch1CBM0QTaLn73Y7GF03";
  };
};
};
};

```