

# NixOS Config

- [configuration.nix](#)
- [bemenu.nix](#)
- [hypridle.nix](#)
- [hyprland.nix](#)
- [hyprlock.nix](#)
- [kitty.nix](#)
- [neovim.nix](#)
- [waybar.nix](#)
- [zsh.nix](#)
- [tmux.conf](#)
- [nixvim.nix](#)
- [shared-home.nix](#)

# configuration.nix

```
{ config, lib, pkgs, ... }:  
  
{  
  imports =  
    [ ./hardware-configuration.nix ];  
  
  # Use the systemd-boot EFI boot loader.  
  boot.loader.systemd-boot.enable = true;  
  boot.loader.efi.canTouchEfiVariables = true;  
  
  nix.settings.experimental-features = [ "nix-command" "flakes" ];  
  nix.optimise.automatic = true;  
  nixpkgs.config.allowUnfree = false;  
  
  networking.hostName = "coracle";  
  networking.networkmanager.enable = true;  
  
  hardware.bluetooth.enable = true;  
  hardware.bluetooth.powerOnBoot = false;  
  
  time.timeZone = "America/New_York";  
  
  fonts.packages = with pkgs; [  
    noto-fonts  
    noto-fonts-cjk  
    noto-fonts-emoji  
    liberation_ttf  
    fira-code  
    fira-code-symbols  
    nerdfonts  
  ];  
  
  security.rtkit.enable = true;  
  security.pam.services.hyprlock = {};
```

```
services.printing.enable = true;
```

```
services.udisks2.enable = true;
```

```
services.pipewire = {
```

```
    enable = true;
```

```
    alsa.enable = true;
```

```
    alsa.support32Bit = true;
```

```
    pulse.enable = true;
```

```
};
```

```
services.flatpak.enable = true;
```

```
xdg.portal = {
```

```
    enable = true;
```

```
    extraPortals = [ pkgs.xdg-desktop-portal-hyprland pkgs.xdg-desktop-portal-gtk ];
```

```
};
```

```
services.openssh = {
```

```
    enable = true;
```

```
    ports = [ 22 ];
```

```
    settings = {
```

```
        PasswordAuthentication = false;
```

```
        PermitRootLogin = "prohibit-password";
```

```
    };
```

```
};
```

```
services.auto-cpufreq.enable = true;
```

```
powerManagement.enable = true;
```

```
users.mutableUsers = true;
```

```
users.defaultUserShell = pkgs.zsh;
```

```
users.users.root = {
```

```
};
```

```
users.users.game = {
```

```
    isNormalUser = true;
```

```
    createHome = true;
```

```
    extraGroups = [ "networkmanager" ];
```

```
};
```

```
users.users.code = {
```

```
    isNormalUser = true;
```

```
    createHome = true;
```

```
extraGroups = [ "networkmanager" "adbusers" "dialout" ];  
};
```

```
environment.systemPackages = [ ];
```

```
programs.zsh.enable = true;  
programs.hyprland.enable = true;  
programs.neovim = {  
  enable = true;  
  configure = {  
    customRC = "  
      set number relativenumber  
      filetype plugin indent on  
    ";  
  };  
  viAlias = true;  
  vimAlias = true;  
};
```

```
programs.tmux = {  
  enable = true;  
  extraConfig = import ./tmux.conf;  
};
```

```
programs.firejail = {  
  enable = true;  
};
```

```
programs.adb.enable = true;
```

```
virtualisation.containers.enable = true;
```

```
virtualisation = {  
  podman = {  
    enable = true;  
  };  
};
```

```
networking.firewall.enable = false;  
system.copySystemConfiguration = true;  
system.autoUpgrade = {  
  enable = true;  
  flake = "inputs.self.outPath";  
  flags = [
```

```
--update-input"  
"nixpkgs"  
"-L"  
];  
dates = "02:00";  
randomizedDelaySec = "45min";  
};  
  
system.stateVersion = "24.11";  
  
}
```

# bemenu.nix

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

```
{  
  programs.bemenu = {  
    enable = true;  
    settings = {  
      line-height = 28;  
      fn = "Fira Code";  
      fb = "#1e1e2e";  
      ff = "#cdd6f4";  
      nb = "#1e1e2e";  
      nf = "#cdd6f4";  
      tb = "#1e1e2e";  
      hb = "#1e1e2e";  
      tf = "#f38ba8";  
      hf = "#f9e2af";  
      af = "#cdd6f4";  
      ab = "#1e1e2e";  
    };  
  };  
};  
}
```

# hypridle.nix

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

{
  services.hypridle = {
    enable = true;
    settings = {
      general = {
        before_sleep_cmd = "hyprlock --immediate";
        after_sleep_cmd = "hyprctl dispatch dpms on";
        ignore_dbus_inhibit = false;
        lock_cmd = "hyprlock --immediate";
      };
    };
  };
}
```

# hyprland.nix

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

{
  wayland.windowManager.hyprland = {
    enable = true;
    settings = {
      "$browser" = "flatpak run --user io.gitlab.librewolf-community";
      "$mainMod" = "SUPER";
      "$menu" = "bemenu-run --fn Fira Code";
      "$terminal" = "kitty";

      monitor=",highrr,auto,auto";
      exec-once="waybar";

      general = {
        allow_tearing=false;
        border_size=2;
        "col.active_border"="rgba(33ccffee) rgba(00ff99ee) 45deg";
        "col.inactive_border"="rgba(595959aa)";
        gaps_in=5;
        gaps_out=20;
        layout="dwindle";
        resize_on_border=true;
      };
      animations = {
        "bezier" = "myBezier, 0.05, 0.9, 0.1, 1.05";
        "animation" = [
          "windows, 1, 7, myBezier"
          "windowsOut, 1, 7, default, popin 80%"
          "border, 1, 10, default"
          "borderangle, 1, 8, default"
          "fade, 1, 7, default"
          "workspaces, 1, 6, default"
        ];
      };
    };
  };
  enabled = true;
```



```
};  
decoration = {  
    rounding = 10;  
    active_opacity=1.0;  
    "col.shadow"="rgba(1a1a1aee)";  
    drop_shadow=true;  
    inactive_opacity=1.0;  
    shadow_range=4;  
    shadow_render_power=3;  
    blur = {  
        "enabled"="true";  
        "passes"="1";  
        "size"="3";  
        "vibrancy"="0.169600";  
    };  
};  
dwindle = {  
    preserve_split=true;  
    pseudotile=true;  
};  
master = {  
    new_status="master";  
};  
input = {  
    touchpad = {  
        natural_scroll=true;  
    };  
    follow_mouse=1;  
    kb_options="ctrl:nocaps";  
    sensitivity=0;  
};  
env = [  
    "XCURSOR_SIZE,38"  
    "HYPRCURSOR_SIZE,38"  
    "BEMENU_BACKEND,wayland"  
];  
  
bind = [  
    "$mainMod, Return, exec, $terminal"  
    "$mainMod, Q, killactive"
```

"\$mainMod SHIFT, E, exit"

"\$mainMod, W, exec, \$browser"

"\$mainMod SHIFT, Space, togglefloating"

"\$mainMod SHIFT, D, exec, \$menu"

"\$mainMod, F, fullscreen, 1"

"\$mainMod, E, togglegroup"

□"\$mainMod, S, changegroupactive"

"\$mainMod SHIFT, minus, movetoworkspacesilent, special"

"\$mainMod, minus, togglespecialworkspace, special"

"\$mainMod, h, movefocus, l"

"\$mainMod, l, movefocus, r"

"\$mainMod, k, movefocus, u"

"\$mainMod, j, movefocus, d"

"\$mainMod, 1, workspace, 1"

"\$mainMod, 2, workspace, 2"

"\$mainMod, 3, workspace, 3"

"\$mainMod, 4, workspace, 4"

"\$mainMod, 5, workspace, 5"

"\$mainMod, 6, workspace, 6"

"\$mainMod, 7, workspace, 7"

"\$mainMod, 8, workspace, 8"

"\$mainMod, 9, workspace, 9"

"\$mainMod, 0, workspace, 10"

"\$mainMod SHIFT, 1, movetoworkspace, 1"

"\$mainMod SHIFT, 1, workspace, 1"

"\$mainMod SHIFT, 2, movetoworkspace, 2"

"\$mainMod SHIFT, 2, workspace, 2"

"\$mainMod SHIFT, 3, movetoworkspace, 3"

"\$mainMod SHIFT, 3, workspace, 3"

"\$mainMod SHIFT, 4, movetoworkspace, 4"

"\$mainMod SHIFT, 4, workspace, 4"

"\$mainMod SHIFT, 5, movetoworkspace, 5"

"\$mainMod SHIFT, 5, workspace, 5"

"\$mainMod SHIFT, 6, movetoworkspace, 6"

"\$mainMod SHIFT, 6, workspace, 6"

"\$mainMod SHIFT, 7, movetoworkspace, 7"

"\$mainMod SHIFT, 7, workspace, 7"

"\$mainMod SHIFT, 8, movetoworkspace, 8"

"\$mainMod SHIFT, 8, workspace, 8"

"\$mainMod SHIFT, 9, movetoworkspace, 9"

```

"$mainMod SHIFT, 9, workspace, 9"
"$mainMod SHIFT, 0, movetoworkspace, 10"
"$mainMod SHIFT, 0, workspace, 10"
"$mainMod SHIFT, h, movewindoworgroup, l"
"$mainMod SHIFT, j, movewindoworgroup, d"
"$mainMod SHIFT, k, movewindoworgroup, u"
"$mainMod SHIFT, l, movewindoworgroup, r"
"$mainMod, mouse_down, workspace, e+1"
"$mainMod, mouse_up, workspace, e-1"
", XF86AudioPlay, exec, playerctl play-pause"
", XF86AudioNext, exec, playerctl next"
", XF86AudioPrev, exec, playerctl previous"
", XF86Tools, exec, playerctl play-pause"
", XF86Favorites, exec, playerctl next"
", XF86Bluetooth, exec, playerctl previous"
", XF86MonBrightnessUp, exec, brightnessctl set +10%"
", XF86MonBrightnessDown, exec, brightnessctl set --min-value=1 10%-"
", XF86AudioRaiseVolume, exec, wpctl set-volume -l 1.5 @DEFAULT_SINK@ 5%+ && wpctl set-mute
@DEFAULT_SINK@ 0"
", XF86AudioLowerVolume, exec, wpctl set-volume @DEFAULT_SINK@ 5%- && wpctl set-mute
@DEFAULT_SINK@ 0"
", XF86AudioMute, exec, wpctl set-mute @DEFAULT_SINK@ toggle"
", XF86AudioMicMute, exec, wpctl set-mute @DEFAULT_SOURCE@ toggle"
"$mainMod&Alt_L, k, exec, hyprctl keyword monitor ,preferred,auto,1,transform,0"
"$mainMod&Alt_L, k, exec, hyprctl keyword input:touchdevice:transform 0"
"$mainMod&Alt_L, k, exec, hyprctl keyword input:tablet:transform 0"
"$mainMod&Alt_L, h, exec, hyprctl keyword monitor ,preferred,auto,1,transform,1"
"$mainMod&Alt_L, h, exec, hyprctl keyword input:touchdevice:transform 1"
"$mainMod&Alt_L, h, exec, hyprctl keyword input:tablet:transform 1"
"$mainMod&Alt_L, j, exec, hyprctl keyword monitor ,preferred,auto,1,transform,2"
"$mainMod&Alt_L, j, exec, hyprctl keyword input:touchdevice:transform 2"
"$mainMod&Alt_L, j, exec, hyprctl keyword input:tablet:transform 2"
"$mainMod&Alt_L, l, exec, hyprctl keyword monitor ,preferred,auto,1,transform,3"
"$mainMod&Alt_L, l, exec, hyprctl keyword input:touchdevice:transform 3"
"$mainMod&Alt_L, l, exec, hyprctl keyword input:tablet:transform 3"
];
bindm = [
"$mainMod, mouse:272, movewindow"
"$mainMod, mouse:273, resizewindow"
];

```

```
}; #Settings block
extraConfig = "
    bind=$mainMod,R,submap,resize
    submap=resize
    binde=,l,resizeactive,150 0
    binde=,h,resizeactive,-150 0
    binde=,k,resizeactive,0 -150
    binde=,j,resizeactive,0 150

    bind=,escape,submap,reset
    submap=reset

    $mode-system = (k) lock | (e) exit | (s) poweroff | (r) reboot
    bind = $mainMod, x, submap, $mode-system
    submap = $mode-system
    bind = , k, exec, hyprlock --immediate
    bindl = , k, submap, reset
    bind = , e, exit
    bind = , r, exec, systemctl reboot
    bind = , s, exec, systemctl poweroff
    bind = , escape, submap, reset
    submap = reset
";
};
}
```

# hyprlock.nix

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

{
  programs.hyprlock = {
    enable = true;
    settings = {
      general = {
        disable_loading_bar = true;
        grace = 300;
        hide_cursor = true;
        no_fade_in = false;
      };

      background = [
        {
          path = "screenshot";
          blur_passes = 3;
          blur_size = 8;
        }
      ];

      input-field = [
        {
          size = "200, 50";
          position = "0, -80";
          monitor = "";
          dots_center = true;
          fade_on_empty = false;
          font_color = "rgb(202, 211, 245)";
          inner_color = "rgb(91, 96, 120)";
          outer_color = "rgb(24, 25, 38)";
          outline_thickness = 5;
          placeholder_text = "<span foreground='##cad3f5'>Password...</span>";
          shadow_passes = 2;
        }
      ]
    }
  }
}
```

```
};
```

```
};
```

```
};
```

```
}
```

# kitty.nix

```
{ config, pkgs, lib, ... }:  
  
{  
  programs.kitty = lib.mkForce {  
    enable = true;  
    settings = {  
      confirm_os_window_close = 0;  
      dynamic_background_opacity = true;  
      background_opacity = "0.5";  
      background_blur = 5;  
      font_family = "Noto Sans";  
    };  
  };  
};  
}
```

# neovim.nix

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

{
  programs.neovim = {
    enable = true;
    extraConfig = "
      set number relativenumber
    ";
    viAlias = true;
    vimAlias = true;
    defaultEditor = true;
  };
}
```



# waybar.nix

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

{
  programs.waybar = {
    enable = true;
    settings = {
      mainBar = {
        layer = "top";
        position = "top";
        modules-left = [ "hyprland/workspaces" "hyprland/submap" ];
        modules-center = [ "mpris" ];
        modules-right = [ "backlight" "privacy" "wireplumber" "disk" "load" "memory" "battery" "tray" "clock" ];

        "hyprland/workspaces" = {
          all-outputs = true;
          on-click = "activate";
        };
      };
    };
  };

  [ "wireplumber" = {
    "format" = "{icon} {volume}%";
    "format-muted" = "[]";
    "on-click" = "set-mute @DEFAULT_SINK@ toggle";
    "max-volume" = 150;
    "scroll-step" = 0.2;
  ];

  [ format-icons = [ "[]" "[]" "[]" ];
  ];

  [ disk = {
    format = "[] {specific_free:0.0f}GB";
  ];

  [ unit = "GB";
  ];

  [ };

  "mpris" = {
    "format" = "{player_icon} {title}";
    "format-paused" = "{status_icon} <i>{title}</i>";
    "player-icons" = {
      "default" = "[]";
      "mpv" = "[]";
    };
  };
}
```

```

};
"status-icons" = {
    "paused" = "⏸";
};
max-length = 50;
"ignored-players" = [ "firefox" ];
};
"clock" = {
    "tooltip-format" = "<tt><small>{calendar}</small></tt>";
    "format" = "⏰ {:%H:%M %m/%d/%Y}";
};
"calendar" = {
    "mode" = "year";
    "mode-mon-col" = 3;
    "weeks-pos" = "right";
    "on-scroll" = 1;
    "format" = {
        "months" = "<span color='#ffead3'><b>{ }</b></span>";
        "days" = "<span color='#ecc6d9'><b>{ }</b></span>";
        "weeks" = "<span color='#99ffdd'><b>W{ }</b></span>";
        "weekdays" = "<span color='#ffcc66'><b>{ }</b></span>";
        "today" = "<span color='#ff6699'><b><u>{ }</u></b></span>";
    };
};
};
"actions" = {
    "on-click-right" = "mode";
    "on-click-forward" = "tz_up";
    "on-click-backward" = "tz_down";
    "on-scroll-up" = "shift_up";
    "on-scroll-down" = "shift_down";
};
"backlight" = {
    "device" = "intel_backlight";
    "format" = "{icon}";
    "format-icons" = ["⬆" "⬇" "⬆" "⬇" "⬆" "⬇" "⬆" "⬇"];
};
"battery" = {
    "states" = {
        "warning" = 30;
        "critical" = 15;
    };
};

```

```

};

"format-icons" = [ "⏻" "🔋" "🔌" "🔌" "🔌" "🔌" "🔌" "🔌" "🔌" "🔌" ];

"format" = "{icon} {capacity}%";

# "format-charging" = "⏻ {capacity}%";

# "format-plugged" = "🔌 {capacity}%";

"format-alt" = "{time}";

};

};

};

style = "

@import "/home/.home-manager/mocha.css";

* {

    font-family: Fira Code;
    font-size: 17px;
    min-height: 0;
}

#waybar {

    background: transparent;
    color: @text;
    margin: 5px 5px;
}

#workspaces {

    border-radius: 1rem;
    margin: 5px;
    background-color: @surface0;
    margin-left: 1rem;
}

#workspaces button {

    color: @lavender;
    border-radius: 1rem;
    padding: 0.4rem;
}

#workspaces button.active {

    color: @sky;
    border-radius: 1rem;

```

```
}
```

```
#workspaces button:hover {  
  color: @sapphire;  
  border-radius: 1rem;  
}
```

```
#custom-music,  
#backlight,  
#privacy,  
#network,  
#wireplumber,  
#disk,  
#load,  
#memory,  
#battery,  
#tray,  
#clock {  
  background-color: @surface0;  
  padding: 0.5rem 1rem;  
  margin: 5px 0;  
}
```

```
#clock {  
  color: @blue;  
  border-radius: 0px 1rem 1rem 0px;  
  margin-right: 1rem;  
}
```

```
#battery {  
  color: @green;  
}
```

```
#battery.charging {  
  color: @green;  
}
```

```
#battery.warning:not(.charging) {  
  color: @red;  
}
```

```
#backlight {
  color: @yellow;
  margin-left: 1rem;
  border-radius: 1rem 0px 0px 1rem;
}

#wireplumber {
  color: @maroon;
}

#tray {
  border-radius: 0px 0px 0px 0px
}
";
};
}
```

## mocha.css

```
@define-color rosewater #f5e0dc;
@define-color flamingo #f2cdcd;
@define-color pink #f5c2e7;
@define-color mauve #cba6f7;
@define-color red #f38ba8;
@define-color maroon #eba0ac;
@define-color peach #fab387;
@define-color yellow #f9e2af;
@define-color green #a6e3a1;
@define-color teal #94e2d5;
@define-color sky #89dceb;
@define-color sapphire #74c7ec;
@define-color blue #89b4fa;
@define-color lavender #b4befe;
@define-color text #cdd6f4;
@define-color subtext1 #bac2de;
@define-color subtext0 #a6adc8;
@define-color overlay2 #9399b2;
@define-color overlay1 #7f849c;
@define-color overlay0 #6c7086;
```

```
@define-color surface2 #585b70;  
@define-color surface1 #45475a;  
@define-color surface0 #313244;  
@define-color base #1e1e2e;  
@define-color mantle #181825;  
@define-color crust #11111b;
```

# zsh.nix

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

{
  programs.zsh = {
    enable = true;
    enableCompletion = true;
    autosuggestion.enable = true;
    syntaxHighlighting.enable = true;
  };
}
```

# tmux.conf

```
"

set -g mouse on
set-window-option -g mode-keys vi


unbind C-B
set -g prefix C-Space
bind C-Space send-prefix


bind R source-file ~/.tmux.conf \; display "Reloaded!"


set-option -g history-limit 50000


bind C-j select-pane -D
bind C-k select-pane -U
bind C-h select-pane -L
bind C-l select-pane -R
bind j select-pane -D
bind k select-pane -U
bind h select-pane -L
bind l select-pane -R


bind r switch-client -T RESIZE


bind -T RESIZE k resize-pane -U \; switch-client -T RESIZE
bind -T RESIZE j resize-pane -D \; switch-client -T RESIZE
bind -T RESIZE h resize-pane -L \; switch-client -T RESIZE
bind -T RESIZE l resize-pane -R \; switch-client -T RESIZE


bind -T RESIZE K resize-pane -U 5 \; switch-client -T RESIZE
bind -T RESIZE J resize-pane -D 5 \; switch-client -T RESIZE
bind -T RESIZE H resize-pane -L 5 \; switch-client -T RESIZE
bind -T RESIZE L resize-pane -R 5 \; switch-client -T RESIZE


bind s split-window -c "#{pane_current_path}"
bind v split-window -h -c "#{pane_current_path}"
```



```
bind Space new-window -c "#{pane_current_path}"
```

```
bind C-Space new-window -c "#{pane_current_path}"
```

```
unbind x
```

```
bind c confirm-before -p "kill-pane #W? (y/n)" kill-pane
```

```
bind N previous-window
```

```
bind y switch-client -T copy-mode
```

```
""
```

# nixvim.nix

```
let
nixvim = import (builtins.fetchGit {
  url = "https://github.com/nix-community/nixvim";
});
in
{
  imports = [
    nixvim.homeManagerModules.nixvim
  ];
  programs.nixvim = {
    enable = true;
    defaultEditor = true;
    vimdiffAlias = true;
    viAlias = true;
    vimAlias = true;
    opts = {
      number = true;
      relativenumber = true;
      shiftwidth = 2;
    };
    plugins = {
      lualine.enable = true;
      bufferline.enable = true;
      nvim-tree.enable = true;
      telescope = {
        enable = true;
        extensions = {
          file-browser.enable = true;
        };
      };
      keymaps = {
        "<C-p>" = {
          action = "git_files";
          options = {
            desc = "Telescope Git Files";
          };
        };
      };
    };
  };
}
```

```
  };
  "<leader>fg" = "live_grep";
};

  };
  indent-blankline.enable = true;
  cmp.enable = true;
  lsp = {
enable = true;
servers = {
  nixd.enable = true;
};
  };
  };
  colorschemes.gruvbox.enable = true;
};
}
```

# shared-home.nix

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

{
  imports = [
    ./bemenu.nix
    ./hypridle.nix
    ./hyprland.nix
    ./hyprlock.nix
    ./kitty.nix
    # ./neovim.nix
    ./nixvim.nix
    ./waybar.nix
    ./zsh.nix
  ];

  home.stateVersion = "24.05"; # Please read the comment before changing.

  home.packages = [
    pkgs.grim
    pkgs.slurp
    pkgs.playerctl
    pkgs.brightnessctl
    pkgs.wl-clipboard
    pkgs.qrencode
    pkgs.libarchive
  ];

  home.file = { };

  home.sessionVariables = { };

  xdg.userDirs = {
    enable = true;
    createDirectories = true;
  };
}
```

```
programs.home-manager.enable = true;
programs.xplr.enable = true;
programs.imv.enable = true;
programs.bat.enable = true;
programs.zathura = {
  enable = true;
  extraConfig = "
    set selection-clipboard clipboard
    set sandbox strict
  ";
};

services.playerctld.enable = true;

}
```