

ERIC BAILEY

DOTFILES

Contents

<i>Machines</i>	5
<i>SRUXPS</i>	5
<i>NixOS configuration</i>	5
<i>home-manager configuration</i>	8
 <i>Modules</i>	11
<i>Common</i>	11
<i>Packages</i>	12
<i>Location</i>	14
<i>NixOS</i>	16
<i>Darwin</i>	19
 <i>Software configuration</i>	23
<i>Bash</i>	23
<i>bat</i>	23
<i>Browserpass</i>	24
<i>bugwarrior</i>	24
<i>direnv</i>	25
<i>Emacs</i>	26
<i>Init File</i>	28
<i>Firefox</i>	32
<i>fish</i>	32
<i>Fish abbreviations</i>	33

<i>fzf</i>	40
<i>Git</i>	41
<i>GPG</i>	46
<i>htop</i>	46
<i>i3</i>	46
<i>i3status</i>	50
<i>jq</i>	51
<i>kitty</i>	51
<i>man</i>	53
<i>nixpkgs</i>	53
<i>rebar3</i>	53
<i>Taskwarrior</i>	54
<i>Config</i>	54
<i>on-exit Git hook</i>	56
 <i>Glossary</i>	 59
 <i>Acronyms</i>	 61
 <i>Bibliography</i>	 63

Machines

SRUXPS

NixOS configuration

The SRUXPS configuration is an expression that takes, at least, `config`, `lib`, and `pkgs`.

```
<machines/sruxps/configuration.nix 1>≡
{ config, lib, pkgs, ... }:
let
  username = "e.bailey";
in
{

  <Import hardware config and modules 2>

  <Set the location to MSP 3>

  <Configure the LUKS devices 4>

  <Link some extra paths 5>

  <Configure the file systems 6>

  <Configure Bluetooth 7>

  <Configure home-manager 8>

  <Configure networking 9>

  <Configure Nix 10>

  <Configure overlays 11>

  <Configure sudo 12>

  <Configure services 13>

  <Configure user 14>

}
```

Import the hardware config and other Nix *modules*.

(Import hardware config and modules 2)≡ (1)

```
imports = [
  <nixos-hardware/dell/xps/13-9380>
  ./hardware-configuration.nix
  ../../nix
  ../../modules/common.nix
  ../../modules/location.nix
  ../../modules/nixos.nix
  ../../modules/packages.nix
  <home-manager/nixos>
];
```

Describe the location module

(Set the location to MSP 3)≡ (1)

```
airportCode = "MSP";
```

Configure the *LUKS* devices.

(Configure the LUKS devices 4)≡ (1)

```
boot.initrd.luks.devices = {
  cryptkey.device = "/dev/disk/by-uuid/603b64c6-8544-4b43-9b6a-7d8a08091514";
  cryptroot.device = "/dev/disk/by-uuid/a81783fe-31ec-4762-a845-4b5be1900e61";
  cryptroot.keyFile = "/dev/mapper/cryptkey";
  cryptswap = {
    device = "/dev/disk/by-uuid/565c0358-110e-4279-ba59-619cb2cc1ebf";
    keyFile = "/dev/mapper/cryptkey";
  };
};
```

(Link some extra paths 5)≡ (1)

```
environment.pathsToLink = [
  "/lib/aspell"
  "/share/emacs/site-lisp"
  "/share/fish"
];
```

(Configure the file systems 6)≡ (1)

```
fileSystems."/ " = {
  device = "/dev/disk/by-uuid/3e6e5ef8-7c9d-4759-94bd-44ac093add8a";
  fsType = "ext4";
};

fileSystems."/boot" = {
  device = "/dev/disk/by-uuid/3898-907E";
  fsType = "vfat";
};
```

(Configure Bluetooth 7)≡ (1)

```
hardware.bluetooth = {
  enable = true;
  extraConfig = "
    [General]
    Enable=Source,Sink,Media,Socket
  ";
};
```

(Configure home-manager 8)≡ (1)

```
home-manager.useUserPackages = true;
home-manager.users."${username}" = args:
  import ./home.nix (args // { inherit pkgs; });
```

```
<Configure networking 9>≡ (1)
networking.hostName = "MSP-EBAILEY01";
```

```
<Configure Nix 10>≡ (1)
nix = {
  nixPath = [
    "nixos-config=/etc/nixos/configuration.nix"
    "nixpkgs-overlays=/etc/nixos/overlays"
  ];
  trustedUsers = [ "root" username ];
};
```

```
<Configure overlays 11>≡ (1)
nixpkgs.overlays =
  let
    path = <nixpkgs-overlays>;
  in
    with builtins;
    map (n: import (path + ("/" + n)))
      (
        filter (
          n: match ".*\\.nix" n ≠ null
          || pathExists (path + ("/" + n + "/default.nix"))
        )
        (attrNames (readDir path))
      )
    ++ (
      with (import <nurpkgs> {}).overlays; [
        nur
        git
        node
      ]
    );
```

```
<Configure sudo 12>≡ (1)
security.sudo = {
  enable = true;
  extraConfig = "
    ${username} ALL=(ALL) NOPASSWD: ALL
  ";
};
```

```
<Configure services 13>≡ (1)
services.bluedn.enable = true;

services.fwupd.enable = true;

# TODO: services.lorri.enable = true;

services.xserver.monitorSection = "
  DisplaySize 406 228
";
```

```

<Configure user 14>≡ (1)
users.mutableUsers = false;
users.users."${username}" = {
  name = username;
  hashedPassword = lib.fileContents (./. + "/secrets/${username}.hashedPassword");
  group = "users";
  extraGroups = [
    "audio"
    "disk"
    "docker"
    "http"
    "networkmanager"
    "systemd-journal"
    "video"
    "wheel"
  ];
  createHome = true;
  uid = 1001;
  home = "/home/${username}";
  shell = "/etc/profiles/per-user/${username}/bin/fish";
};

```

home-manager configuration

```

<machines/sruxps/home.nix 15>≡
{ lib, pkgs, ... }:

{

  <Import various program configs 16>

  <Configure the primary email account 17>

  <Install some user packages 18>

  _module.args.pkgs = lib.mkForce pkgs;

}

```



```

<Import various program configs 16>≡ (15)
imports = (import <nurpkgs> {}).home-manager.modules ++ [
  ../../config/bash.nix
  ../../config/bat.nix
  ../../config/browserpass.nix
  ../../config/bugwarrior.nix
  ../../config/direnv.nix
  ../../config/emacs
  ../../config/firefox.nix
  ../../config/fish
  ../../config/fzf.nix
  ../../config/git
  ../../config/gpg.nix
  ../../config/htop.nix
  ../../config/i3
  ../../config/jq.nix
  ../../config/kitty
  ../../config/man.nix
  ../../config/nixpkgs
  ../../config/rebar3.nix
  ../../config/taskwarrior
];

```

Configure Vim

```

<Configure the primary email account 17>≡ (15)
accounts.email.accounts.primary = {
  address = "e.bailey@sportradar.com";
  gpg.key = "EFD6F1EDC84D2FA935E38570462054AB8B682702";
  primary = true;
  realName = "Eric Bailey";
};

```

```

<Install some user packages 18>≡ (15) 19>
home.packages = with pkgs; (
  [
    aws-iam-authenticator
    ec2instanceconnectcli

```

Clementine¹ is a great music player.

¹ <https://www.clementine-player.org/>

```

<Install some user packages 18>+≡ (15) <18 22>
  clementine

```

I haven't used Dhall in a while, but it's still interesting.

```

<* 20>≡ 21>
  dhall
  dhall-json

```

I might need Docker Compose² again someday, but not lately.

² <https://docs.docker.com/compose/>

```

<* 20>+≡ <20 25>
  docker-compose

```

```

<Install some user packages 18>+≡ (15) <19 23>
  ghc

```

iw³ is useful for troubleshooting wireless issues, like in Trondheim.

³ <https://wireless.wiki.kernel.org/en/users/documentation/iw>

```

<Install some user packages 18>+≡ (15) <22 24>
  iw
  jdk

```

`naal`⁴ is my tool for non-interactive AWS Azure login.

⁴ <https://github.com/yurrriq/naal>

```
<Install some user packages 18>+≡ (15) <23 26>
  naal
  networkmanager-openconnect
```

It would be cool to play with `Next`⁵ some lazy day.

⁵ <https://next.atlas.engineer/>

```
<* 20>+≡ <21>
  next
```

```
<Install some user packages 18>+≡ (15) <24 27>
  openconnect
```

`PulseAudio Volume Control`⁶ is useful for configuring Bluetooth audio.

⁶ <https://freedesktop.org/software/pulseaudio/pavucontrol/>

```
<Install some user packages 18>+≡ (15) <26 28>
  pavucontrol
  renderizer
  scc
```

More Trondheim wireless debugging...

```
<Install some user packages 18>+≡ (15) <27>
  wirelesstools
  zoom-us
] ++ (
  with nodePackages; [
    aws-azure-login
  ]
)
);
```

Modules

Common

This module is a lambda that takes at least `pkgs` in its arguments.

```
<modules/common.nix 29>≡ 30>
{ pkgs, ... }:

{
```

Configure garbage collection (`gc`) and my personal Cachix⁷-based `binaryCaches`. ⁷ <https://cachix.org>

- <https://yurrriq.cachix.org>
- <https://yurrriq-nur-packages.cachix.org>

```
<modules/common.nix 29>+≡ <29 31>
nix = {
  binaryCaches = [
    "https://yurrriq.cachix.org"
    "https://yurrriq-nur-packages.cachix.org"
  ];
  binaryCachePublicKeys = [
    "yurrriq.cachix.org-1:evpJ5wKluf7Q0Ccv69VkJxCOtHgubrqX1Zpp3JAXLBE="
    "yurrriq-nur-packages.cachix.org-1:7kbjuGBUZcWf876g2cdeImIQXrXz0hpMVBqY0yyAv70="
  ];
  gc = {
    automatic = true;
    options = "-delete-older-than 30d";
  };
};
```

Installed the Nerd Fonts patched Iosevka .

```
<modules/common.nix 29>+≡ <30 32>
fonts = {
  enableFontDir = true;
  fonts = with pkgs; [
    (nerdfonts.override { withFont = "Iosevka"; })
  ];
};
```

Add Nerd Fonts link

Add Iosevka link

Enable system-wide Bash completion.

```
<modules/common.nix 29>+≡ <31 33>
programs.bash.enableCompletion = true;
```

```

<modules/common.nix 29>+≡
}
<32

```

Packages

```

<modules/packages.nix 34>≡
{ lib, pkgs, ... }:
35>

```

```

{

```

```

<modules/packages.nix 34>+≡
environment.systemPackages = with pkgs; (
<34 36>
[
  aspell
  aspellDicts.en
  awscli
  bind
  cachix
  coreutils
  # TODO: cquery
  curl
  expect
  gawk
  gcc
  gnumake
  gnused
  gnutar
  # TODO: graphviz
  gzip
  httpie
  # TODO: http-prompt
  # TODO: idris
  moreutils
  neofetch
  nix
  noweb
  pandoc
  prettytyping
  ripgrep
  shellcheck
  shfmt
  silver-searcher # TODO: find helm-rg solution
  slack
  spotify
  stow

```

```

<modules/packages.nix 34>+≡                                <35 37>
(
    texlive.combine {
        inherit (texlive) scheme-small
            datetime
            dirtytalk
            fmtcount
            framed
            fvextra
            hardwrap
            ifplatform
            latexmk
            minted
            titlesec
            todonotes
            tufte-latex
            xetex
            xstring
            ;
        inherit noweb;
    }
)

<modules/packages.nix 34>+≡                                <36 38>
    tree
    unzip
    vim
    watch
    # wipe
    yq
]

<modules/packages.nix 34>+≡                                <37 39>
) ++ (
    with haskellPackages; [
        cabal-install
        cabal2nix
        hpack
        stylish-haskell
    ]
]

<modules/packages.nix 34>+≡                                <38 40>
) ++ (
    with nodePackages; [
        # TODO: bash-language-server
    ]
]

<modules/packages.nix 34>+≡                                <39 41>
) ++ (
    with python3Packages; [
        pygments
    ]
]

```

```

<modules/packages.nix 34>+≡                                     <40 42>
    ) ++ lib.optionals stdenv.isDarwin (
      [
        clementine
        m-cli
        sourcetree
        onyx
      ] ++ (
        with chunkwm; [
          core
          ffm
          tiling
        ]
      )
    )

<modules/packages.nix 34>+≡                                     <41 43>
    ) ++ lib.optionals stdenv.isLinux [
      gnome3.networkmanagerapplet
      hicolor-icon-theme
      keybase-gui
      libreoffice
      qpdfview
      playerctl
      psmisc

<modules/packages.nix 34>+≡                                     <42 44>
    (
      signal-desktop.override {
        spellcheckerLanguage = "en_US";
      }
    )

<modules/packages.nix 34>+≡                                     <43 45>
    thunderbird
    volumeicon
    xclip
    xorg.xbacklight
  ];

<modules/packages.nix 34>+≡                                     <44
}

```

Location

```

<modules/location.nix 46>≡                                     47>
  { config, lib, ... }:

<modules/location.nix 46>+≡                                     <46 48>
  let
    airportCode = config.airportCode;
  in
  {

```

```

<modules/location.nix 46>+≡                                     <47 49>
  options.airportCode = lib.mkOption {
    default = "MSP";
    type = lib.types.enum [ "ATL" /* "LHR" */ "LJU" "MSP" "TRD" ];
  };

```

```

<modules/location.nix 46>+≡                                     <48 50>
  config = lib.mkMerge [

```

The Sportradar office in Minneapolis.

```

<modules/location.nix 46>+≡                                     <49 51>
  (
    lib.mkIf (airportCode == "MSP") {
      location = {
        latitude = 44.98;
        longitude = -93.27;
      };
      time.timeZone = "America/Chicago";
    }
  )

```

Cook Out on Ponce de Leon Ave.

```

<modules/location.nix 46>+≡                                     <50 52>
  (
    lib.mkIf (airportCode == "ATL") {
      location = {
        latitude = 33.77;
        longitude = -84.37;
      };
      time.timeZone = "America/New_York";
    }
  )

```

The Sportradar office in London.

```

<modules/location.nix 46>+≡                                     <51 53>
  (
    lib.mkIf (airportCode == "LHR") {
      location = {
        latitude = 51.51;
        longitude = -0.09;
      };
      time.timeZone = "Europe/London";
    }
  )

```

The Sportradar office in Ljubljana.

```

<modules/location.nix 46>+≡                                     <52 54>
  (
    lib.mkIf (airportCode == "LJU") {
      location = {
        latitude = 46.09;
        longitude = 14.55;
      };
      time.timeZone = "Europe/Ljubljana";
    }
  )

```

The Sportradar office in Trondheim.

```

<modules/location.nix 46>+≡                                     <53 55>
(
  lib.mkIf (airportCode == "TRD") {
    location = {
      latitude = 63.43;
      longitude = 10.40;
    };
    time.timeZone = "Europe/Oslo";
  }
)

<modules/location.nix 46>+≡                                     <54>
];

}

```

NixOS

```

<modules/nixos.nix 56>≡                                         57>
{ pkgs, ... }:

{

  <modules/nixos.nix 56>+≡                                     <56 58>
    imports = with (import <nurpkgs> {}).modules; [
      pass
      tomb
    ];

  <modules/nixos.nix 56>+≡                                     <57 59>
    boot = {
      cleanTmpDir = true;
      loader = {
        systemd-boot.enable = true;
        efi.canTouchEfiVariables = true;
      };
    };

  <modules/nixos.nix 56>+≡                                     <58 60>
    i18n = {
      # consoleFont = "latarcyrheb-sun32";
      consoleFont = "Lat2-Terminus16";
      consoleKeyMap = "us";
      defaultLocale = "en_US.UTF-8";
    };

  <modules/nixos.nix 56>+≡                                     <59 61>
    location.provider = "manual";

```



```

<modules/nixos.nix 56>+≡                                     <60 62>
    networking.networkmanager.enable = true;

```

```

<modules/nixos.nix 56>+≡                                     <61 63>
    nixpkgs.config.allowUnfree = true;

```

```

<modules/nixos.nix 56>+≡                                     <62 64>
    programs = {
        pass = {
            enable = true;
            genphrase = true;
            git-helper = true;
            otp = true;
            update = true;
        };
        tomb = {
            enable = true;
            resize = true;
            slam = true;
        };
    };

```

```

<modules/nixos.nix 56>+≡                                     <63 65>
    services = {
        logind.lidSwitch = "hibernate";

        kbfs.enable = true;

        redshift = {
            enable = true;
            temperature.night = 2300;
        };

        unclutter = {
            enable = false; # FIXME
            extraOptions = [ "exclude-root" "ignore-scrolling" ];
            # FIXME
            threshold = 1;
            timeout = 1;
        };

```

```

<modules/nixos.nix 56>+≡
xserver = {
    enable = true;

    autorun = true;

    desktopManager = {
        gnome3.enable = false;
        xterm.enable = false;
        default = "none";
    };

    displayManager.lightdm.enable = true;

    layout = "us";

    libinput = {
        enable = true;
        accelSpeed = "1.0";
        disableWhileTyping = true;
        naturalScrolling = false;
        tapping = true;
    };

    videoDrivers = [
        "intel"
    ];

    windowManager = {
        default = "i3";
        i3.enable = true;
    };

    xkbOptions = "ctrl:nocaps,compose:ralt";
};

<64 66>

<modules/nixos.nix 56>+≡
hardware.pulseaudio = {
    enable = true;
    extraModules = [ pkgs.pulseaudio-modules-bt ];
    package = pkgs.pulseaudioFull;
};
sound.enable = true;

<modules/nixos.nix 56>+≡
system.stateVersion = "19.09";

<66 68>

<modules/nixos.nix 56>+≡
virtualisation.docker.enable = true;

<67 69>

<modules/nixos.nix 56>+≡
}
<68>

```

Darwin

```

<modules/darwin.nix 70>≡ 71▷
  { pkgs, ... }:
```

```

  {
```

```

<modules/darwin.nix 70>+≡ <70 72>
    nixpkgs.config.allowUnfree = true;
```

```

<modules/darwin.nix 70>+≡ <71 73>
  services.chunkwm = {
    enable = true;
    extraConfig = "
      chunkc core::log_file /usr/local/var/log/chunkwm.log
      chunkc core::log_level info
      chunkc tiling::rule -owner "Cisco AnyConnect Secure Mobility Client" -state float &
      chunkc tiling::rule -owner Emacs -except "^$" -state tile &
    ";
    package = pkgs.chunkwm.core;
    plugins = {
      dir = "/run/current-system/sw/bin/chunkwm-plugins/";
      list = [ "ffm" "tiling" ];
      "tiling".config = "
        chunkc set desktop_padding_step_size      0
        chunkc set desktop_gap_step_size          0
        chunkc set global_desktop_mode             bsp
        chunkc set global_desktop_offset_top       0
        chunkc set global_desktop_offset_bottom    0
        chunkc set global_desktop_offset_left      0
        chunkc set global_desktop_offset_right     0
        chunkc set global_desktop_offset_gap       0
        chunkc set bsp_spawn_left                  1
        chunkc set bsp_optimal_ratio                1.618
        chunkc set bsp_split_mode                   optimal
        chunkc set bsp_split_ratio                  0.66
        chunkc set window_focus_cycle              all
        chunkc set mouse_follows_focus              1
        chunkc set window_region_locked            1
      ";
    };
  };
};
```

```

<modules/darwin.nix 70>+≡
services.skhd = {
  enable = true;
  package = pkgs.skhd;
  skhdConfig = "
    alt - space : /Applications/kitty.app/Contents/MacOS/kitty -single-instance -d ~

    ctrl + alt + cmd - up    : chunkc tiling::window -warp north
    ctrl + alt + cmd - right : chunkc tiling::window -warp east
    ctrl + alt + cmd - down  : chunkc tiling::window -warp south
    ctrl + alt + cmd - left  : chunkc tiling::window -warp west

    shift + alt + cmd - up    : chunkc tiling::window -focus north
    shift + alt + cmd - right : chunkc tiling::window -focus east
    shift + alt + cmd - down  : chunkc tiling::window -focus south
    shift + alt + cmd - left  : chunkc tiling::window -focus west

    ctrl + alt + cmd - e : chunkc tiling::desktop -equalize
    ctrl + alt + cmd - f : chunkc tiling::window -toggle float
    ctrl + alt + cmd - m : chunkc tiling::window -toggle fullscreen

    shift + cmd - 1 : chunkc tiling::window -send-to-desktop 1
    shift + cmd - 2 : chunkc tiling::window -send-to-desktop 2
    shift + cmd - 3 : chunkc tiling::window -send-to-desktop 3
    shift + cmd - 4 : chunkc tiling::window -send-to-desktop 4
    shift + cmd - 5 : chunkc tiling::window -send-to-desktop 5
    shift + cmd - 6 : chunkc tiling::window -send-to-desktop 6
  ";
};

```

Set `AppleKeyboardUIMode = 3` to enable full keyboard access, e.g. tab in dialogs. Disable press-and-hold, in favor of key repeat.

Dig through <https://github.com/mathiasbynens/dotfiles/blob/master/.macos>

```

<modules/darwin.nix 70>+≡
system.defaults.NSGlobalDomain = {
  AppleKeyboardUIMode = 3;

  ApplePressAndHoldEnabled = false;
  InitialKeyRepeat = 10;
  KeyRepeat = 1;

  NSAutomaticDashSubstitutionEnabled = false;
  NSAutomaticQuoteSubstitutionEnabled = false;
  NSAutomaticSpellingCorrectionEnabled = true;

  NSNavPanelExpandedStateForSaveMode = true;
  NSNavPanelExpandedStateForSaveMode2 = true;
};

```

```

<modules/darwin.nix 70>+≡
system.defaults.dock = {
  autohide = true;
  mru-spaces = false;
  orientation = "right";
  showhidden = true;
};

```

```
<modules/darwin.nix 70>+≡ <75 77>  
  system.defaults.finder = {  
    AppleShowAllExtensions = true;  
    FXEnableExtensionChangeWarning = false;  
    QuitMenuItem = true;  
  };
```

```
<modules/darwin.nix 70>+≡ <76 78>  
  system.defaults.trackpad.Clicking = true;
```

```
<modules/darwin.nix 70>+≡ <77>  
}
```


Software configuration

Bash

Enable the Bash module, as well as autojump.

```
<config/bash.nix 79>≡
{ ... }:

{

  programs.bash = {
    enable = true;
    enableAutojump = true;
  };

}
```

bat

Enable the `bat`⁸ module.

⁸ <https://github.com/sharkdp/bat>

```
<config/bat.nix 80>≡ 81▷
{ ... }:

{

  programs.bat = {
    enable = true;
    config = {

      For the pager, use less.
      <config/bat.nix 80>+≡ <80 82>
        pager = "less -FR";

      Set the style to changes.
      <config/bat.nix 80>+≡ <81 83>
        style = "changes";

      Use the Monokai Extended theme.
      <config/bat.nix 80>+≡ <82 84>
        theme = "Monokai Extended";

      <config/bat.nix 80>+≡ <83
    };
  };

}
```

Browserpass

```

<config/browserpass.nix 85>≡
{ pkgs, ... }:

{

    home.packages = with pkgs; [
        browserpass
    ];

    programs.browserpass = {
        enable = true;
        browsers = [ "firefox" ];
    };

}

```

bugwarrior

```

<config/bugwarrior.nix 86>≡ 87▷
{ config, pkgs, ... }:

{

<config/bugwarrior.nix 86>+≡ <86 88>
    home.packages = with pkgs; [
        bugwarrior
        pass
    ];

<config/bugwarrior.nix 86>+≡ <87 89>
    imports = [
        ./taskwarrior
    ];

```


⟨config/bugwarrior.nix 86⟩+≡

◁88 90▷

```
xdg.configFile."bugwarrior/bugwarriorrc".text = "
[general]
targets = sportradar_jira
taskrc = ${config.home.homeDirectory}/.taskrc
inline_links = False
annotation_links = True
annotation_comments = True
legacy_matching = False
log.level = DEBUG
log.file = ${config.home.homeDirectory}/log/bugwarrior.log
annotation_length = 80

[sportradar_jira]
service = jira
jira.base_uri = https://jira.sportradar.ag
jira.username = e.bailey
jira.password = @oracle:eval:pass jira.sportradar.ag
jira.query = ((assignee = e.bailey OR reporter = e.bailey) OR (summary ~ e.bailey OR description ~ e.bailey OR c
jira.version = 8
jira.add_tags = work
jira.description_template = {{jiraid}}: {{jirasummary}}
";
```

⟨config/bugwarrior.nix 86⟩+≡

◁89

```
}
```

direnv

⟨config/direnv.nix 91⟩≡

```
{ ... }::

{

  programs.direnv = {
    enable = true;
    enableZshIntegration = false;
  };

}
```

Emacs

```

<config/emacs/default.nix 92>≡
{ config, lib, pkgs, ... }:

{

  imports = [
    ./packages.nix
  ];

  home.file.".emacs.d/init.el".source = ./init.el;

  home.sessionVariables = rec {
    EDITOR = "emacsclient -nw -a \"\"";
    GIT_EDITOR = EDITOR;
    VISUAL = "emacsclient -cna \"\"";
  };

  programs.emacs.enable = true;

  programs.fish.shellAliases = lib.mkIf (config.programs.fish.enable) rec {
    e = "emacsclient -na \"\"";
    ec = e + " -c";
    et = "emacsclient -nw -a \"\"";
  };

  services.emacs.enable = ! pkgs.stdenv.isDarwin;

}

```

```

<config/emacs/packages.nix 93>≡
{ ... }:

{

  programs.emacs.extraPackages = epkgs: with {
    elpa = epkgs.elpaPackages;
    melpa = epkgs.melpaPackages;
    melpaStable = epkgs.melpaStablePackages;
    # org = epkgs.orgPackages;
  }; (

```

```

<config/emacs/packages.nix 93>+≡
  with elpa; [
    mmm-mode
    rainbow-mode
  ]

```

94▷

◁93 95▷

<config/emacs/packages.nix 93>+≡

<94 96>

```

) ++ (
  with melpa; [
    avy
    better-defaults
    # clj-refactor
    # clojure-mode
    # TODO: company-lsp
    # TODO: cquery
    crux
    dhall-mode
    direnv
    dockerfile-mode
    editorconfig
    elixir-mode
    emojiify
    enh-ruby-mode
    # FIXME: erlang
    fill-column-indicator
    fish-mode
    gap-mode
    go-mode
    graphviz-dot-mode
    haskell-mode
    helm-ag
    # TODO: hindent
    hl-todo
    htmlize
    idris-mode
    j-mode
    kubernetes-tramp
    # TODO: lsp-haskell
    # TODO: lsp-mode
    # TODO: lsp-ui
    magit
    markdown-mode
    multiple-cursors
    nix-mode
    paredit
    rainbow-delimiters
    robe
    rust-mode
    rvm
    smex
    tuareg
    use-package
    whitespace-cleanup-mode
    yaml-mode
  ]
) ++ (

```

```

<config/emacs/packages.nix 93>+≡
  with melpaStable; [
    ess
  ]
);

}

```

Init File

```

<config/emacs/init.el 97>≡
  (server-start)

<config/emacs/init.el 97>+≡
  (setq server-window 'pop-to-buffer-same-window)

<config/emacs/init.el 97>+≡
  (column-number-mode 1)

<config/emacs/init.el 97>+≡
  (add-to-list 'exec-path "/run/current-system/sw/bin")

<config/emacs/init.el 97>+≡
  (menu-bar-mode 0)

  (tool-bar-mode 0)

<config/emacs/init.el 97>+≡
  (let ((font (if (eq system-type 'darwin)
                  "Iosevka-16"
                  "Iosevka-11"))))
    (set-face-attribute 'default t :font font)
    (set-frame-font font nil t))

<config/emacs/init.el 97>+≡
  (require 'package)

<config/emacs/init.el 97>+≡
  (setq-default indent-tabs-mode nil
                 package-archives nil
                 package-enable-at-startup nil)

<config/emacs/init.el 97>+≡
  (package-initialize)

  (eval-when-compile
    (require 'use-package))

  (setq-default use-package-always-defer t
                 use-package-always-ensure t)

```

<https://stackoverflow.com/a/18330742>

```
<config/emacs/init.el 97>+≡ <105 107>
(let ((backup-directory (concat user-emacs-directory "backup")))
  (unless (file-exists-p backup-directory)
    (make-directory backup-directory t))
  (setq backup-directory-alist '((" " . ,backup-directory))))

(setq auto-save-default      t
      auto-save-interval    200
      auto-save-timeout     20
      backup-by-copying     t
      delete-by-moving-to-trash t
      delete-old-versions   t
      kept-new-versions     6
      kept-old-versions     2
      make-backup-files     t
      vc-make-backup-files  t
      version-control       t)
```

<https://anirudhsasikumar.net/blog/2005.01.21.html> Copyright

©2004-2011 Anirudh Sasikumar. All rights reserved.

```
<config/emacs/init.el 97>+≡ <106 108>
(define-minor-mode sensitive-mode
  "For sensitive files like password lists.
  It disables backup creation and auto saving.

  With no argument, this command toggles the mode.
  Non-null prefix argument turns on the mode.
  Null prefix argument turns off the mode."
  ;; The initial value.
  nil
  ;; The indicator for the mode line.
  " □"
  ;; The minor mode bindings.
  nil
  (if (symbol-value sensitive-mode)
      (progn
        ;; disable backups
        (set (make-local-variable 'backup-inhibited) t)
        ;; disable auto-save
        (if auto-save-default
            (auto-save-mode -1)))
      ;; resort to default value of backup-inhibited
      (kill-local-variable 'backup-inhibited)
      ;; resort to default auto save setting
      (if auto-save-default
          (auto-save-mode 1)))))
```

```
<config/emacs/init.el 97>+≡ <107 109>
(dolist (pattern '("^\\(/dev/shm/\\|/tmp/\\)"
                  "\\.(enc\\|pgp\\|hashedPassword\\|)$"))
  (add-to-list 'auto-mode-alist (cons pattern 'sensitive-mode)))
```

```
<config/emacs/init.el 97>+≡ <108 110>
(setq custom-file "~/.emacs.d/private/local/custom.el")
```

```

<config/emacs/init.el 97>+≡                                     <109 111>
  (load-theme 'wombat)

<config/emacs/init.el 97>+≡                                     <110 112>
  (global-set-key (kbd "C-x C-k") 'kill-this-buffer)
  (global-set-key (kbd "s-u") 'revert-buffer)

<config/emacs/init.el 97>+≡                                     <111 113>
  (setq c-default-style      "k&r"
        c-basic-offset      4
        emacs-lisp-mode-hook '(fci-mode paredit-mode
                                rainbow-delimiters-mode)
        js-indent-level      2
        text-mode-hook       '(text-mode-hook-identify))

<config/emacs/init.el 97>+≡                                     <112 114>
  (setq-default fill-column 80)

<config/emacs/init.el 97>+≡                                     <113 115>
  (org-babel-do-load-languages
   'org-babel-load-languages
   '((shell . t)))

<config/emacs/init.el 97>+≡                                     <114 116>
  (use-package avy
    :demand
    :config
    (global-set-key (kbd "C-;") 'avy-goto-char)
    (global-set-key (kbd "C-'"') 'avy-goto-char-2)
    (global-set-key (kbd "M-g f") 'avy-goto-line))

<config/emacs/init.el 97>+≡                                     <115 117>
  (use-package crux
    :demand
    :config (global-set-key (kbd "C-a") 'crux-move-beginning-of-line))

<config/emacs/init.el 97>+≡                                     <116 118>
  (use-package editorconfig
    :ensure t
    :config
    (editorconfig-mode 1))

<config/emacs/init.el 97>+≡                                     <117 119>
  (use-package emojiify
    :demand
    :config
    (global-emojiify-mode)
    (global-emojiify-mode-line-mode))

<config/emacs/init.el 97>+≡                                     <118 120>
  (use-package helm
    :demand
    :config
    (global-set-key (kbd "M-s-f") 'helm-do-grep-ag)
    (global-set-key (kbd "M-s-f") 'helm-do-grep-ag))

<config/emacs/init.el 97>+≡                                     <119 121>
  (use-package hl-todo
    :demand
    :config (global-hl-todo-mode t))

```

```

<config/emacs/init.el 97>+≡ <120 122>
  (use-package magit
    :demand
    :config (global-magit-file-mode t))

<config/emacs/init.el 97>+≡ <121 123>
  (use-package multiple-cursors
    :demand
    :config (global-set-key (kbd "C-S-c C-S-c") 'mc/edit-lines))

<config/emacs/init.el 97>+≡ <122 124>
  (use-package nix-mode
    :mode ("\\.nix\\'"))

<config/emacs/init.el 97>+≡ <123 125>
  (use-package noweb-mode
    :load-path "/run/current-system/sw/share/emacs/site-lisp"
    :mode ("\\.nw\\'")
    :demand)

<config/emacs/init.el 97>+≡ <124 126>
  (use-package robe
    :demand
    :mode ("\\.rb\\'"))

<config/emacs/init.el 97>+≡ <125 127>
  (use-package rvm
    :ensure t)

<config/emacs/init.el 97>+≡ <126 128>
  (use-package smex
    :demand
    :config
    (global-set-key (kbd "M-x") 'smex)
    (global-set-key (kbd "M-X") 'smex-major-mode-commands)
    (global-set-key (kbd "C-c C-c M-x") 'execute-extended-command))

<config/emacs/init.el 97>+≡ <127 129>
  (use-package tuareg
    :mode ("\\.ml\\'" "\\.mli\\'"))

<config/emacs/init.el 97>+≡ <128
  (use-package whitespace-cleanup-mode
    :demand
    :config (global-whitespace-cleanup-mode t))

```

Firefox

```

<config/firefox.nix 130>≡
{ pkgs, ... }:

{

  programs.firefox = {
    enable = true;
    extensions = with (import <nur> { inherit pkgs; }).repos.rycee.firefox-addons; [
      https-everywhere
      privacy-badger
    ];
  };

}

```

fish

```

<config/fish/default.nix 131>≡
{ lib, pkgs, ... }:

{

  imports = [
    ./abbrs.nix
    ./aliases.nix
  ];

  home = {
    packages = with pkgs; [
      autojump
    ];
    sessionVariables = {
      SHELL = "fish";
      TERMINAL = "kitty";
    };
  };

  programs.fish = let
    inherit (lib.strings) fileContents;
  in
  {
    enable = true;
    interactiveShellInit = fileContents ./interactiveShellInit.fish;
    promptInit = "
      ${fileContents ./sushi/fish_prompt.fish}
      ${fileContents ./sushi/fish_right_prompt.fish}
    ";
    shellInit = fileContents ./shellInit.fish;
  };

}

```


Fish abbreviations

“abbr manages abbreviations - user-defined words that are replaced with longer phrases after they are entered.

For example, a frequently-run command like `git checkout` can be abbreviated to `gco`. After entering `gco` and pressing `Space` or `Enter`, the full text `git checkout` will appear in the command line.”⁹

<config/fish/abbrs.nix 132>≡

```
{ ... }:
```

```
{
```

```
    programs.fish.shellAbbrs = {
```

```
        <Kubernetes fish abbreviations 133>
```

```
        <Nix fish abbreviations 134>
```

```
        <ripgrep fish abbreviations 135>
```

```
    };
```

```
}
```

⁹ abbr - manage fish abbreviations.
URL <https://fishshell.com/docs/current/cmds/abbr.html>

`kubectl` IS VERY CUMBERSOME to type all the time. Using fish abbreviations, save some precious keystrokes. It’s definitely *worth the time*.

<Kubernetes fish abbreviations 133>≡ (132)

```
kc = "kubectl";
```

```
kcd = "kubectl drain -delete-local-data -ignore-daemonsets";
```

```
kcnp = "kubectl get pods -field-selector=spec.nodeName=";
```

```
kn = "kubens";
```

```
kns = "kubens";
```

```
kt = "kubetail";
```

```
kx = "kubectx";
```

IT’S USEFUL TO RUN `nix build` with various flags. Why not save some keystrokes for those tasks, too?

<Nix fish abbreviations 134>≡ (132)

```
nb = "nix build";
```

```
nbd = "nix build -dry-run";
```

```
nbn = "nix build -no-link";
```

```
nbo = "nix build -o";
```

I’M NOT QUITE USED TO all the `ripgrep` flags yet. I’m also a lazy typist, so define some memorable abbreviations.

“Searches case insensitively if the pattern is all lowercase. Search case sensitively otherwise.”¹⁰

<ripgrep fish abbreviations 135>≡ (132) 136▷

```
rg = "rg -S";
```

<ripgrep fish abbreviations 135>+≡ (132) <135 137▷

```
rga = "rg -hidden -iglob !.git";
```

```
rgf = "rg -F";
```

¹⁰ Andrew Gallant. `rg` - man page.
URL <https://www.mankier.com/1/rg>

“Never print the file path with the matched lines”, and “[s]uppress line numbers.”¹¹

¹¹ Andrew Gallant. `rg` - man page.
URL <https://www.mankier.com/1/rg>

```
<ripgrep fish abbreviations 135>+≡ (132) <136 138>
  rgin = "rg -IN";
```

```
λ rg -IN name: helmfile.d/*.yaml | sort -u
- name: cert-manager
- name: cluster-autoscaler
- name: elasticsearch
- name: elasticsearch-client
- name: elasticsearch-curator
- name: elasticsearch-exporter
- name: elasticsearch-master
- name: external-dns
- name: fluentd-elasticsearch
- name: hubble
- name: kibana
- name: kube-resource-report
- name: metrics-server
- name: nginx-ingress-core
- name: prometheus-operator
```

```
<ripgrep fish abbreviations 135>+≡ (132) <137
  rgn = "rg -no-heading";
```

```
<config/fish/aliases.nix 139>≡
{ ... } :

{

  programs.fish.shellAliases = rec {
    gpg = "gpg2";

    k = "clear";

    l = "ls -color=auto -Glah";
    ll = "ls -color=auto -Glh";
    ls = "ls -color=auto -G";

    # Old Darwin habits
    pbcopy = "xclip -sel clipboard";
    pbpaste = "${pbcopy} -o";
  };
}
```

<config/fish/shellInit.fish 140>≡

```
for p in /run/current-system/sw/bin ~/bin
  if not contains $p $fish_user_paths
    set -U fish_user_paths $p $fish_user_paths
  end
end

set -U fish_user_paths /run/wrappers/bin $fish_user_paths

function fish_title
  echo "$PWD | $_" | sed "s|$HOME|~|g"
end
```

<config/fish/interactiveShellInit.fish 141>≡

142>

```
function clone
  function __update
    test -d $argv[1]; and cd $argv[1]; and git fetch -all; and git pull
  end

  function __usage
    echo "Usage: clone [username] [repository] [[destination]]"
  end

  set -local num_args (count $argv)

  if test $num_args -ge 2
    set -local user $argv[1]
    set -local repo $argv[2]

    if test $num_args -eq 2
      set dest ~/src/$user/$repo
    else if test $num_args -eq 3
      set dest $argv[3]/$user/$repo
    else
      __usage
    end

    echo $dest

    git clone git@github.com:$user/$repo.git $dest; or __update $dest; or __usage
  else
    __usage
  end
end
```

```

<config/fish/interactiveShellInit.fish 141>+≡                                <141 143>
function latest -d 'Print the latest tag (on GitHub) for a given user and repo.'
    # TODO: __usage

    set -local num_args (count $argv)

    if test $num_args -eq 2
        set -local user $argv[1]
        set -local repo $argv[2]
        http https://api.github.com/repos/$user/$repo/tags | jq '.[0].name'
    end
end

<config/fish/interactiveShellInit.fish 141>+≡                                <142 144>
command -sq fluidsynth; and function playmidi
    fluidsynth -i ~/lib/arachno-soundfont/Arachno\ SoundFont\ -\ Version\ 1.0.sf2 $argv
end

<config/fish/interactiveShellInit.fish 141>+≡                                <143 145>
command -sq kitty; and function icat
    kitty +kitten icat $argv
end

<config/fish/interactiveShellInit.fish 141>+≡                                <144 146>
command -sq kubectl; and begin
    # TODO: Add option to print server versions too.
    function k8s::versions
        printf "kubectl %s\n" (command kubectl version -client -short)
        printf "helm %s\n" (command helm version -client -short)
        command helmfile -version
        printf "kops %s\n" (command kops version)
    end
end

<config/fish/interactiveShellInit.fish 141>+≡                                <145 147>
# FIXME: functions rvm >/dev/null 2>&1; and rvm default

<config/fish/interactiveShellInit.fish 141>+≡                                <146 148>
set fish_greeting

<config/fish/interactiveShellInit.fish 141>+≡                                <147>
command -sq task; and command -sq jq; and function tj \
    -d 'Open the Jira ticket associated with a Taskwarrior task'
    test (count $argv) -ne 1; and return
    open (task $argv[1] export | jq -r '.[0].jiraurl')
end

```

(config/fish/sushi/fish_prompt.fish 149)≡

```
function fish_prompt
  set -l code $status

  if set -q IN_NIX_SHELL
    set symbol "λ"
  else
    set symbol "λ"
  end

  if git::is_repo
    set -l branch (git::branch_name ^/dev/null)
    set -l ref (git show-ref -head -abbrev | awk '{print substr($0,0,7)}' | sed -n 1p)

    if git::is_stashed
      echo -n -s (white)"^"(off)
    end

    echo -n -s (red)" "(off)

    if git::is_dirty
      printf (white)"*"(off)
    end

    if command git symbolic-ref HEAD > /dev/null ^/dev/null
      if git::is_staged
        printf (cyan)"$branch"(off)
      else
        printf (yellow)"$branch"(off)
      end
    else
      printf (dim)"$ref"(off)
    end

    for remote in (git remote)
      set -l behind_count (echo (command git rev-list $branch..$remote/$branch ^/dev/null | wc -l | tr -d " ")
      set -l ahead_count (echo (command git rev-list $remote/$branch..$branch ^/dev/null | wc -l | tr -d " ")

      if test $ahead_count -ne 0; or test $behind_count -ne 0; and test (git remote | wc -l) -gt 1
        echo -n -s " "(orange)$remote(off)
      end

      if test $ahead_count -ne 0
        echo -n -s (white)" +"$ahead_count(off)
      end

      if test $behind_count -ne 0
        echo -n -s (white)" -"$behind_count(off)
      end
    end

    echo -n -s (red)" "(off)
  end

  if test "$code" = 0
    echo -n -s (red)"$symbol"(off)
  end
end
```

```

        else
            echo -n -s (dim)"$symbol"(off)
        end
    end
end

```

<config/fish/sushi/fish_right_prompt.fish 150>≡

151▷

```

# ----- [ Colors ]

```

```

function orange
    set_color -o ee5819
end

```

```

function yellow
    set_color -o b58900
end

```

```

function red
    set_color -o d30102
end

```

```

function cyan
    set_color -o 2aa198
end

```

```

function white
    set_color -o fdf6e3
end

```

```

function dim
    set_color -o 4f4f4f
end

```

```

function off
    set_color -o normal
end

```

```

<config/fish/sushi/fish_right_prompt.fish 150>+≡          <150 152>
# ----- [ Git ]

function git::is_repo
    test -d .git; or command git rev-parse -git-dir >/dev/null ^/dev/null
end

function git::branch_name
    git::is_repo; and begin
        command git symbolic-ref -short HEAD ^/dev/null;
        or command git show-ref -head -s -abbrev | head -n1 ^/dev/null
    end
end

function git::is_dirty
    git::is_repo; and not command git diff -no-ext-diff -quiet -exit-code
end

function git::is_staged
    git::is_repo; and begin
        not command git diff -cached -no-ext-diff -quiet -exit-code
    end
end

function git::is_stashed
    git::is_repo; and begin
        command git rev-parse -verify -quiet refs/stash >/dev/null
    end
end

```

```

<config/fish/sushi/fish_right_prompt.fish 150>+≡          <151 153>
# ----- [ AWS ]

function aws::current_profile
    if set -q AWS_PROFILE
        printf "$AWS_PROFILE"
    else if set -q AWS_DEFAULT_PROFILE
        printf "$AWS_DEFAULT_PROFILE"
    end
end

```

```

<config/fish/sushi/fish_right_prompt.fish 150>+≡          <152 154>
# ----- [ Kubernetes ]

function k8s::current_context
    command kubectl config current-context
end

function k8s::current_namespace
    command kubectl config view -minify -o jsonpath='{.contexts[0].context.namespace}'
end

function k8s::current_user
    command kubectl config view -minify -o jsonpath='{.users[0].name}' \
        | string replace "@sportradar.com" ""
end

```

```

<config/fish/sushi/fish_right_prompt.fish 150>+≡
# ----- [ Right Prompt ]
function fish_right_prompt
    set -l cwd (basename (prompt_pwd))

    if git::is_repo
        set root_folder (command git rev-parse --show-toplevel ^/dev/null)
        set parent_root_folder (dirname $root_folder)
        set cwd (echo $PWD | sed -e "s|$parent_root_folder/||")
    end

    printf (yellow)"(" (dim)$cwd (yellow)" " (off)

    set -l aws_profile (aws::current_profile)
    test -n "$aws_profile";
    and printf (dim)"{" (yellow)(aws::current_profile)(dim)" " (off)

    command -sq kubectl; and begin
        if [ "e.bailey" = (k8s::current_user) ]
            printf (dim)"[" (off)
        else
            printf (red)"[" (off)
        end

        printf (yellow)(k8s::current_context)(off)

        set -l k8s_namespace (k8s::current_namespace)
        test -n "$k8s_namespace";
        and printf (dim)"/" (yellow)"$k8s_namespace" (off)

        if [ "e.bailey" = (k8s::current_user) ]
            printf (dim)"]" (off)
        else
            printf (red)"]" (off)
        end
    end

    printf (dim)(date +%H(yellow): (dim)%M(yellow): (dim)%S)(off)
end

```

fzf

```

<config/fzf.nix 155>≡
{ ... }:

{

    programs.fzf = {
        enable = true;
        enableZshIntegration = false;
    };

}

```


Git

```

<config/git/default.nix 156>≡
{ config, lib, pkgs, ... }:

{
  imports = [
    ./aliases.nix
    ./config.nix
    ./packages.nix
  ];

  programs.git = {
    enable = true;
    ignores = [
      "**/"
      ".DS_Store"
    ];
    lfs.enable = true;
  } // (
    with config.accounts.email.accounts.primary; {
      signing.key = gpg.key;
      userEmail = address;
      userName = realName;
    }
  );
}

```

```

<config/git/packages.nix 157>≡
{ lib, pkgs, ... }:

{
  home.packages = with pkgs; (
    [
      diff-pdf
      git
      git-lfs
      github-cli
      kdiff3
      nix-prefetch-git
      nix-prefetch-github
      sops
    ] ++ (
      lib.optionals pkgs.stdenv.isLinux [
        # git-cola
      ]
    ) ++ (
      with gitAndTools; [
        git-crypt
        # gitflow
        hub
        lab
      ]
    ) ++ (
      with nodePackages; [
        codeowners
        diff-so-fancy
      ]
    )
  );
}

```

```

<config/git/config.nix 158>≡
{ config, pkgs, ... }:

{

  programs.git.extraConfig = {

    color = {
      diff-highlight = {
        oldNormal = "red bold";
        oldHighlight = "red bold 52";
        newNormal = "green bold";
        newHighlight = "green bold 22";
      };
      diff = {
        meta = 227;
        frag = "magenta bold";
        commit = "227 bold";
        old = "red bold";
        new = "green bold";
        whitespace = "red reverse";
      };
      status = {
        added = "green";
        changed = "yellow";
        untracked = "cyan";
      };
    };

    commit.template = "${config.xdg.dataHome}/git/commit.template";

    core.pager = "diff-so-fancy | less -tabs=4 -RFX";

    credential = {
      helper = "${pkgs.gitAndTools.pass-git-helper}/bin/pass-git-helper";
      useHttpPath = true;
    };

    diff = {
      sopsdiffer = {
        textconv = "sops -d";
      };
    };

    difftool = {
      pdfdiffer.cmd = "diff-pdf -view \"$LOCAL\" \"$REMOTE\"";
      prompt = false;
      trustExitCode = true;
    };

    rerere.enabled = true;

    ui.color = true;
  };
}

```

```
xdg.configFile."pass-git-helper/git-pass-mapping.ini" = {  
    source = ./git-pass-mapping.ini;  
};  
  
xdg.dataFile."git/commit.template" = {  
    source = ./commit.template;  
};  
  
}
```

<config/git/aliases.nix 159>≡

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

```
{
```

```
  programs.fish.shellAbbrs = lib.mkIf (config.programs.fish.enable) {
```

```
    g = "git";
```

```
    gd = "git d";
```

```
    gdc = "git dc";
```

```
    gs = "git st";
```

```
  };
```

```
  programs.git.aliases = rec {
```

```
    bm = "branch -merged";
```

```
    bnm = "branch -no-merged";
```

```
    ca = "commit -amend";
```

```
    cam = "${ca} -m";
```

```
    can = "${ca} -no-edit";
```

```
    cans = "${can} -S";
```

```
    cas = "${ca} -S";
```

```
    casm = "${cas} -m";
```

```
    cm = "commit -m";
```

```
    co = "checkout";
```

```
    cob = "${co} -b";
```

```
    cs = "crypt status";
```

```
    cse = "crypt status -e";
```

```
    csm = "commit -S -m";
```

```
    csu = "crypt status -u";
```

```
    d = "diff";
```

```
    dc = "diff -cached";
```

```
    ds = "diff -stat";
```

```
    ffco = "flow feature checkout";
```

```
    ffr = "flow feature rebase";
```

```
    ffs = "flow feature start";
```

```
    frf = "flow release finish";
```

```
    frfs = "flow release finish -s";
```

```
    frs = "flow release start";
```

```
    r = "reset";
```

```
    rb = "rebase";
```

```
    rba = "rebase -abort";
```

```
    rbc = "rebase -continue";
```

```
    rbi = "rebase -interactive";
```

```
    rbs = "rebase -skip";
```

```
    rest = "reset";
```

```
    rh = "reset -hard";
```

```
    sa = "stash apply";
```

```
    sk = "stash -keep-index";
```

```
    sl = "stash list";
```

```
    sp = "stash pop";
```

```
    spa = "stash -patch";
```

```
    ss = "stash save";
```

```
    st = "status -s";
```

```
    stat = "status";
```

```
    tree = "log -all -graph -oneline";
```

```
  };
```

```
}
```

GPG

<config/gpg.nix 160>≡

```
{ config, ... }:  
  
{  
  
  programs.gpg = {  
    enable = true;  
    settings = {  
      default-key = config.accounts.email.accounts.primary.gpg.key;  
      keyid-format = "long";  
      no-emit-version = true;  
    };  
  };  
  
  services.gpg-agent = {  
    enable = true;  
    defaultCacheTtl = 600;  
    enableSshSupport = true;  
    maxCacheTtl = 3600;  
  };  
  
}
```

htop

<config/htop.nix 161>≡

```
{ ... }:  
  
{  
  
  programs.htop.enable = true;  
  
}
```

i3

<config/i3/default.nix 162>≡

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

163▷

```

<config/i3/default.nix 162>+≡
    home.packages = with pkgs; [
        flameshot
        i3lock
        rofi
        rofi-pass
    ];

```

<162 164>

o use `i3status-rust`:

```
gs, ... }:
```

```

me.packages = with pkgs; [
    font-awesome-ttf
    i3status-rust

```

```

ession.windowManager.i3.bars = [
{
    fonts = [ "Iosevka" "FontAwesome 12" ];
    statusCommand = "${pkgs.i3status-rust}/bin/i3status-rs ${config.xdg.configHome}/i3/status.toml";
    # colors = { ... };
}

```

```

<config/i3/default.nix 162>+≡
    services.compton.enable = true;

    xdg.configFile."i3status/config".source = ../i3status/config;

    xdg.dataFile."i3/matrix.png".source = ./matrix.png;

    xsession.enable = true;

    xsession.windowManager.i3 = {
        enable = true;
        config =
            let

```

<163 165>

T

Use Solarized Dark colors.¹²

`<config/i3/default.nix 162>+≡`

```

colors = rec {
  background = "#002b36";
  focused = {
    border = "#002b36";
    background = "#586e75";
    text = "#fdf6e3";
    indicator = "#268bd2";
    childBorder = ""; # "#285577";
  };
  focusedInactive = {
    border = "#002b36";
    background = "#073642";
    text = "#839496";
    indicator = "#073642";
    childBorder = ""; # "#5f676a";
  };
  unfocused = {
    border = "#002b36";
    background = "#073642";
    text = "#839496";
    indicator = "#073642";
    childBorder = ""; # "#222222";
  };
  urgent = {
    border = "#002b36";
    background = "#dc322f";
    text = "#fdf6e3";
    indicator = "#002b36";
    childBorder = ""; # "#900000";
  };
};

barColors = {
  inherit (colors) background;
  focusedWorkspace = {
    border = "#b58900";
    background = "#b58900";
    text = "#002b36";
  };
  inactiveWorkspace = {
    border = "#073642";
    background = "#002b36";
    text = "#839496";
  };
  separator = "#586e75";
  statusline = "#839496";
  urgentWorkspace = {
    border = "#dc322f";
    background = "#dc322f";
    text = "#fdf6e3";
  };
};

```

`<164 166>`

¹² <https://github.com/tobiaszwaszak/i3wm/blob/9c097f3/config>


```

<config/i3/default.nix 162>+≡
modifier = "Mod4";
in
{
  bars = [
    {
      fonts = [ "Iosevka" ];
      statusCommand = "${pkgs.i3status}/bin/i3status";
      colors = barColors;
    }
  ];
  inherit colors;
  fonts = [ "Iosevka 12" ];
  keybindings = lib.mkOptionDefault {
    "XF86MonBrightnessDown" = "exec xbacklight -dec 10";
    "XF86MonBrightnessUp" = "exec xbacklight -inc 10";
    "XF86AudioMute" = "exec -no-startup-id amixer set Master 1+ toggle";
    "XF86AudioLowerVolume" = "exec -no-startup-id amixer set Master 5%-";
    "XF86AudioRaiseVolume" = "exec -no-startup-id amixer set Master 5%+";
    "XF86AudioPrev" = "exec playerctl previous";
    # "XF86AudioPlayPause" = "exec playerctl play-pause";
    "XF86AudioPlay" = "exec playerctl play-pause";
    # "XF86AudioPause" = "exec playerctl play-pause";
    "XF86AudioNext" = "exec playerctl next";
    "${modifier}+Escape" = "
      exec i3lock -i ${config.xdg.dataHome}/i3/matrix.png
    ";
    "${modifier}+Shift+minus" = "move scratchpad";
    "${modifier}+minus" = "scratchpad show";
    "${modifier}+Tab" = "exec -no-startup-id rofi -show window";
    "${modifier}+d" = "
      exec -no-startup-id "rofi -combi-modi run,window,drun -show combi -modi combi"
    ";
    "${modifier}+space" = "
      exec -no-startup-id "rofi -combi-modi run,window,drun -show combi -modi combi"
    ";
    "Print" = "exec flameshot gui";
  };
  inherit modifier;
};
};
}

```

<165

i3status

Use some Solarized Dark colors for good, bad, and degraded.¹³

¹³ <https://github.com/tobiaszwaszak/i3wm/blob/9c097f3/i3status.conf>

```
<config/i3status/config 167>≡ 168>
general {
    colors = true
    color_good = "#859900"
    color_bad = "#dc322f"
    color_degraded = "#cb4b16"
    interval = 5
    output_format = "i3bar"
}
```

```
<config/i3status/config 167>+≡ <167 169>
order += "path_exists VPN"
```

```
path_exists VPN {
    format = "□"
    format_down = ""
    path = "/proc/sys/net/ipv4/conf/tun0"
}
```

```
order += "wireless _first_"
```

```
wireless _first_ {
    format_up = "%ssid"
}
```

```
order += "load"
```

```
load {
    format = "%1min %5min %15min"
    format_above_threshold = "□ %1min %5min %15min □"
    max_threshold = 9
}
```

```
order += "battery all"
```

```
battery all {
    format = "%status %percentage %remaining"
```

Show the battery percentage without decimals.

```
<config/i3status/config 167>+≡ <168 170>
integer_battery_capacity = true
```

Use the last full capacity, instead of the design capacity. Refer to the [man page](#)¹⁴ for more details.

¹⁴ https://i3wm.org/i3status/manpage.html#_battery

```
<config/i3status/config 167>+≡ <169 171>
last_full_capacity = true
```

```

<config/i3status/config 167>+≡
    low_threshold = 10
    status_bat = ""
    status_chr = "4"
    status_full = ""
    status_unk = ""
    threshold_type = "time"
}

order += "tztime trondheim"

tztime trondheim {
    format = "%a %H:%M"
    hide_if_equals_localtime = true
    locale = "nb_NO.UTF-8"
    timezone = "Europe/Oslo"
}

order += "tztime local"

tztime local {
    format = "%a %d.%m %H:%M:%S"
}

```

jq

```

<config/jq.nix 172>≡
{ ... }::

{

    programs.jq.enable = true;

}

```

kitty

```

<config/kitty/default.nix 173>≡
{ ... }::

{

    programs.kitty = {
        enable = true;
        config = {
            editor = "emacsclient -nw -a """;
            scrollbarlines = -1;
            term = "xterm";
        };
    };

}

```

```

<config/kitty/kitty.conf 174>≡ 175>
font_family Iosevka
bold_font Iosevka Medium
italic_font Iosevka Italic
bold_italic_font Iosevka Medium Italic

<config/kitty/kitty.conf 174>+≡ <174 176>
font_size 12.0
scrollback_lines -1

<config/kitty/kitty.conf 174>+≡ <175 177>
foreground #839496
background #002b36

color0 #073642
color1 #dc322f
color2 #859900
color3 #b58900
color4 #268bd2
color5 #d33682
color6 #2aa198
color7 #eee8d5
color8 #002b36
color9 #cb4b16
color10 #586e75
color11 #657b83
color12 #839496
color13 #6c71c4
color14 #93a1a1
color15 #fdf6e3

<config/kitty/kitty.conf 174>+≡ <176 178>
editor emacsclient -nw -a ""

<config/kitty/kitty.conf 174>+≡ <177 179>
term xterm

<config/kitty/kitty.conf 174>+≡ <178>
macos_option_as_alt yes

kitty_mod ctrl+shift

map kitty_mod+enter new_window_with_cwd
map kitty_mod+k combine : clear_terminal scrollback active : send_text normal \x0c

```

man

```
⟨config/man.nix 180⟩≡
{ ... }:

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

nixpkgs

```
⟨config/nixpkgs/default.nix 181⟩≡
{ ... }:

{
    nixpkgs.config = import ./nixpkgs-config.nix;
    xdg.configFile."nixpkgs/config.nix".source = ./nixpkgs-config.nix;
}
```

```
⟨config/nixpkgs/nixpkgs-config.nix 182⟩≡
{
    allowBroken = false;
    allowUnfree = true;
    allowUnsupportedSystem = false;
}
```

rebar3

```
⟨config/rebar3.nix 183⟩≡
{ ... }:

{
    xdg.configFile."rebar3/rebar.config".text = "
        {plugins, [rebar3_hex]}.
    ";
}
```

*Taskwarrior**Config*

```

<config/taskwarrior/default.nix 184>≡
{ config, lib, pkgs, ... }:

{

```

185>

SINCE HOME-MANAGER GENERATES A READ-ONLY `.taskrc` FILE, and using `contexts`¹⁵ requires Taskwarrior to be able to modify the it, use the following hacky workaround.

¹⁵ <https://taskwarrior.org/docs/context.html>

Set `$TASKRC` to `~/.taskrc-dirty`, which must contain at least `include ~/.taskrc`.

```

<config/taskwarrior/default.nix 184>+≡
    home.sessionVariables = {
        TASKRC = "~/.taskrc-dirty";
    };

```

<184 186>

DEFINE short Taskwarrior shell aliases.

```

<config/taskwarrior/default.nix 184>+≡
    programs.fish.shellAliases = lib.mkIf (config.programs.fish.enable) rec {
        p = "task ls limit:page";
        pp = tbd;
        t = "task limit:page";
        ta = "task add";
        tbd = "task burndown.daily";
        te = "env VISUAL=$EDITOR task edit";
        tl = "task list";
        tm = "task mod";
    };

```

<185 187>

CONFIGURE TASKWARRIOR, using `home-manager`¹⁶.

¹⁶ <https://github.com/rycee/home-manager>

`<config/taskwarrior/default.nix 184>+≡`

`<186 188>`

```

programs.taskwarrior = {
  enable = true;
  colorTheme = "solarized-dark-256";
  config = {
    context.other = "jiraurl.none or -work";
    context.work = "jiraurl.any or +work";
    uda = {
      jiracreatedts = {
        label = "Created At";
        type = "date";
      };
      jiradescription = {
        label = "Jira Description";
        type = "string";
      };
      jiraestimate = {
        label = "Estimate";
        type = "numeric";
      };
      jirafixversion = {
        label = "Fix Version";
        type = "string";
      };
      jiraid = {
        label = "Jira Issue ID";
        type = "string";
      };
      jiraissuetype = {
        label = "Issue Type";
        type = "string";
      };
      jirastatus = {
        label = "Jira Status";
        type = "string";
      };
      jirasummary = {
        label = "Jira Summary";
        type = "string";
      };
      jiraurl = {
        label = "Jira URL";
        type = "string";
      };
    };
  };
};

```

LINK the on-exit Git hook.

ref section

```
<config/taskwarrior/default.nix 184>+≡
xdg.dataFile."task/hooks/on-exit-git.sh" = {
  executable = true;
  source = ./on-exit-git.sh;
};

}
```

on-exit Git hook

FIRST, use a fairly portable Bash shebang, and be safe.¹⁷

```
<config/taskwarrior/on-exit-git.sh 189>≡
#!/usr/bin/env bash
```

```
set -eufo pipefail
```

Run in debug mode¹⁸, if the environment variable `DEBUG` is nonempty.

¹⁷ `-exit` immediately upon failure, treat `-Unset` variables as an error, disable `-file` globbing, and fail if any part of a pipeline fails (`-o pipefail`).

¹⁸ i.e. echo commands

```
<config/taskwarrior/on-exit-git.sh 189>+≡
if [ -n "${DEBUG:-}" ]; then
  set -x
fi
```

PARSE the command line arguments.

Don't include the `api` version, `rc` file, or Taskwarrior version.

N.B. The positional arguments are formatted as `foo:bar`, where `foo` is the name of the argument, and `bar` is the value.

```
i="${1#api:}"
rc="${4#rc:}"
rsion="${6#version:}"
```

Parse and store the values of the arguments, `args`, `command`, and `data` directory.

```
<config/taskwarrior/on-exit-git.sh 189>+≡
args="${2#args:}"
command="${3#command:}"
data="${5#data:}"
```

THROUGHOUT THIS SCRIPT, run `git` as if it were started in the Taskwarrior `data` directory, i.e. `git -C "$data"`.

```
<config/taskwarrior/on-exit-git.sh 189>+≡
if git -C "$data" diff -quiet; then
  if [ -n "${DEBUG:-}" ]; then
    echo 'No changes to commit'
  fi
  exit 0
```

If there are no changes, `exit` successfully, printing an informative message if running in debug mode. `git diff` `exits` with status code `0` if there are no differences between the working tree and the index.

If present, stage the changes or die trying.

```
<config/taskwarrior/on-exit-git.sh 189>+≡
elif ! git -C "$data" add -A; then
  echo 'Failed to add files to the index'
  exit 100
```


Try to commit the changes, or else.

`<config/taskwarrior/on-exit-git.sh 189>+≡`

```
elif ! git -C "$data" commit -qm "$command: ${args##task $command}"; then
    echo 'Failed to record changes to the repository'
    exit 101
```

Quietly store the current contents of the index in a new commit, along with a log message describing the changes.

`<193 195>`

If running in debug mode, print a brief summary of the commit.

`<config/taskwarrior/on-exit-git.sh 189>+≡`

```
elif [ -n "${DEBUG:-}" ]; then
    git -C "$data" log -oneline -1
fi
```

`<194`

Glossary

module “Each NixOS module is a file that handles one logical aspect of the configuration, such as a specific kind of hardware, a service, or network settings.”¹⁹ [6](#)

¹⁹ <https://nixos.org/nixos/manual/index.html#sec-writing-modules>

Acronyms

LUKS Linux Unified Key Setup [6](#)

Bibliography

abbr - manage fish abbreviations. URL <https://fishshell.com/docs/current/cmds/abbr.html>.

Andrew Gallant. rg - man page. URL <https://www.mankier.com/1/rg>.

Rafael J. Wysocki. CPU Performance Scaling – The Linux Kernel documentation. URL <https://www.kernel.org/doc/html/v4.19/admin-guide/pm/cpufreq.html#cpu-performance-scaling-in-linux>.

To-Do

Describe the location module	6
Configure Vim	9
Add Nerd Fonts link	11
Add Iosevka link	11
Dig through https://github.com/mathiasbynens/dotfiles/ blob/master/.macos	20
ref section	56