ERIC BAILEY

DOTFILES

Contents

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⟩

  ⟨*Configure vm.swappiness* 5⟩

  ⟨*Add the coretemp kernel module* 6⟩

  ⟨*Link some extra paths* 7⟩

  ⟨*Configure the file systems* 8⟩

  ⟨*Configure Bluetooth* 10⟩

  ⟨*Configure home-manager* 11⟩

  ⟨*Configure networking* 12⟩

  ⟨*Configure Nix* 13⟩

  ⟨*Configure overlays* 14⟩

  ⟨*Configure sudo* 15⟩

  ⟨*Configure services* 16⟩

  ⟨*Configure user* 17⟩

```
}
```

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/c4e26a93-c0c9-4680-bb79-8c0ca47df96c";
    keyFile = "/dev/mapper/cryptkey";
  };

  cryptswap = {
    device = "/dev/disk/by-uuid/565c0358-110e-4279-ba59-619cb2cc1ebf";
    keyFile = "/dev/mapper/cryptkey";
  };
};
```

https://docs.oracle.com/en/operating-systems/oracle-linux/
7/admin/section_h4v_3yt_tr.html

⟨*Configure vm.swappiness 5*⟩≡   (1)

```
boot.kernel.sysctl = {
  "vm.swappiness" = 1;
};
```

⟨*Add the coretemp kernel module 6*⟩≡   (1)

```
boot.kernelModules = [
  "coretemp"
];
```

⟨*Link some extra paths 7*⟩≡   (1)

```
environment.homeBinInPath = true;
environment.pathsToLink = [
  "/lib/aspell"
  "/share/emacs/site-lisp"
  "/share/fish"
  # FIXME: "/share/icons"
];
```

btrfs clean boot?

⟨*Configure the file systems* 8⟩≡   (1) 9▷

```
fileSystems."/" = {
  device = "/dev/disk/by-uuid/71cfcb8a-13a9-49f0-b034-a9c8841be07b";
  fsType = "btrfs";
  options = [ "subvol=@" "rw" "noatime" "compress=zstd" "ssd" "space_cache" ];
};

fileSystems."/home" = {
  device = "/dev/disk/by-uuid/71cfcb8a-13a9-49f0-b034-a9c8841be07b";
  fsType = "btrfs";
  options = [ "subvol=@home" "rw" "noatime" "compress=zstd" "ssd" "space_cache" ];
};

fileSystems."/nix" = {
  device = "/dev/disk/by-uuid/71cfcb8a-13a9-49f0-b034-a9c8841be07b";
  fsType = "btrfs";
  options = [ "subvol=@nix" "rw" "noatime" "compress=zstd" "ssd" "space_cache" ];
};

fileSystems."/var" = {
  device = "/dev/disk/by-uuid/71cfcb8a-13a9-49f0-b034-a9c8841be07b";
  fsType = "btrfs";
  options = [ "subvol=@var" "rw" "noatime" "compress=zstd" "ssd" "space_cache" ];
};

fileSystems."/boot" = {
  device = "/dev/disk/by-uuid/54D0-C859";
  fsType = "vfat";
};
```

⟨*Configure the file systems* 8⟩+≡   (1) ◁8

```
fileSystems."/mnt/music" = {
  device = "192.168.1.147:/volume1/homes/eric/music";
  fsType = "nfs";
  options = [ "noatime" "noauto" "rw" "x-systemd.automount" ];
};
```

⟨*Configure Bluetooth* 10⟩≡   (1)

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

⟨*Configure home-manager* 11⟩≡   (1)

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

⟨*Configure networking* 12⟩≡   (1)

```
networking.hostName = "MSP-EBAILEY01";

networking.interfaces.wlp1s0.useDHCP = true;
networking.useDHCP = false; # NOTE: Deprecated, so set it false.
```

⟨*Configure Nix* 13⟩≡   (1)
```
nix = {
  nixPath = [
    "nixos-config=/etc/nixos/configuration.nix"
    "nixpkgs-overlays=/etc/nixos/overlays"
  ];
  trustedUsers = [ "root" username ];
};
```

⟨*Configure overlays* 14⟩≡   (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* 15⟩≡   (1)
```
security.sudo = {
  enable = true;
  extraConfig = "
    ${username} ALL=(ALL) NOPASSWD: ALL
  ";
};
```

⟨*Configure services* 16⟩≡   (1)
```
services.blueman.enable = true;

services.fwupd.enable = true;

# TODO: services.lorri.enable = true;

services.tlp.enable = true;

services.xserver.displayManager.lightdm.autoLogin = {
  enable = true;
  user = username;
};

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

⟨*Configure user* 17⟩≡ (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* 18⟩≡

```
{ lib, pkgs, ... }:
let
  nixpkgs-unstable = import (import ../../nix/sources.nix).nixpkgs-unstable {
    config.allowUnfree = true;
  };
in
{

  ⟨Import various program configs 19⟩

  ⟨Configure the primary email account 20⟩

  ⟨Install some user packages 21⟩

  _module.args.pkgs = lib.mkForce pkgs;

}
```

⟨*Import various program configs* 19⟩≡   (18)
```
imports = [
  ../../config/bash.nix
  ../../config/bat.nix
  ../../config/browserpass.nix
  ../../config/bugwarrior.nix
  ../../config/direnv.nix
  ../../config/dunst
  ../../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.nix
  ../../config/man.nix
  ../../config/nixpkgs
  ../../config/rebar3.nix
  ../../config/taskwarrior
  ../../config/xmonad
];
```

> Configure Vim

⟨*Configure the primary email account* 20⟩≡   (18)
```
accounts.email.accounts.primary = {
  address = "e.bailey@sportradar.com";
  gpg.key = "EFD6F1EDC84D2FA935E38570462054AB8B682702";
  primary = true;
  realName = "Eric Bailey";
};
```

⟨*Install some user packages* 21⟩≡   (18) 22 ▷
```
home.packages = with pkgs; (
  [
    aws-iam-authenticator
    ec2instanceconnectcli
```

⟨*Install some user packages* 21⟩+≡   (18) ◁21 23 ▷
```
    clac
    dolphin
    lorri
    mtr
    progress
    pv
```

Clementine[1] is a great music player.

[1] https://www.clementine-player.org/

⟨*Install some user packages* 21⟩+≡   (18) ◁22 25 ▷
```
    clementine # FIXME: Unfree
```

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

⟨* 24⟩≡   30 ▷
```
    dhall
    dhall-json
```

I might need Docker Compose[2] again someday, but not lately.

[2] https://docs.docker.com/compose/

⟨*Install some user packages* 21⟩+≡   (18) ◁23 26 ▷
```
    docker-compose
```

⟨*Install some user packages* 21⟩+≡   (18) ◁25 27▷
```
    ghc
```

iw[3] is useful for troubleshooting wireless issues, like in Trondheim.

⟨*Install some user packages* 21⟩+≡   (18) ◁26 28▷
```
    iw
    jdk
```
```
https://github.com/Azure/kubelogin
```

⟨*Install some user packages* 21⟩+≡   (18) ◁27 29▷
```
    nixpkgs-unstable.kubectx
    kubelogin
    k9s
    stern
```

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

⟨*Install some user packages* 21⟩+≡   (18) ◁28 31▷
```
    naal
    networkmanager-openconnect
```

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

⟨* 24⟩+≡   ◁24
```
    next
```

⟨*Install some user packages* 21⟩+≡   (18) ◁29 32▷
```
    openconnect
```

PulseAudio Volume Control[6] is useful for configuring Bluetooth audio.

⟨*Install some user packages* 21⟩+≡   (18) ◁31 33▷
```
    pavucontrol
    renderizer
    scc
    lm_sensors
    nfs-utils
    libnotify
    dunst
    paper-icon-theme
```

Try to improve battery life.

⟨*Install some user packages* 21⟩+≡   (18) ◁32 34▷
```
    powertop
```

More Trondheim wireless debugging...

⟨*Install some user packages* 21⟩+≡   (18) ◁33
```
    wirelesstools
    nixpkgs-unstable.zoom-us
  ] ++ (
    with nodePackages; [
      aws-azure-login
    ]
  )
);
```

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

4 https://github.com/yurrriq/naal

5 https://next.atlas.engineer/

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

Modules

Common

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

⟨*modules/common.nix* 35⟩≡   36 ▷
```
{ pkgs, ... }:

{
```

Configure garbage collection (`gc`) and my personal Cachix[7]-based   [7] https://cachix.org
`binaryCaches`.

- https://yurrriq.cachix.org

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

⟨*modules/common.nix* 35⟩+≡   ◁ 35 37 ▷
```
  nix = {
    binaryCaches = [
      "https://yurrriq.cachix.org"
      "https://yurrriq-nur-packages.cachix.org"
    ];
    binaryCachePublicKeys = [
      "yurrriq.cachix.org-1:evpJ5wKluf7QOCcv69VkIxCOtHgubrqXlZpp3JAXLBE="
      "yurrriq-nur-packages.cachix.org-1:7kbjuGBUZcWf876g2cdelmIQXrXzOhpMVBqYOyyAv70="
    ];
    gc = {
      automatic = true;
      options = "--delete-older-than 30d";
    };
  };
```

Install the Nerd Fonts[8] patched Iosevka[9].   [8] https://www.nerdfonts.com
⟨*modules/common.nix* 35⟩+≡   ◁ 36 38 ▷   [9] http://be5invis.github.io/Iosevka
```
  fonts = {
    enableFontDir = true;
    fonts = with pkgs; [
      (nerdfonts.override { withFont = "Iosevka"; })
    ];
  };
```

Enable system-wide Bash completion.
⟨*modules/common.nix* 35⟩+≡   ◁ 37 39 ▷
```
  programs.bash.enableCompletion = true;
```

⟨*modules/common.nix* 35⟩+≡   ◁38
```
  }
```

Packages

⟨*modules/packages.nix* 40⟩≡   41 ▷
```
  { lib, pkgs, ... }:
  let
    nixpkgs-unstable = import (import ../nix/sources.nix).nixpkgs-unstable {
      config.allowUnfree = true;
    };
  in
  {
```

⟨*modules/packages.nix* 40⟩+≡   ◁40 42 ▷
```
    environment.systemPackages = with pkgs; (
      [
        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
        prettyping
        ripgrep
        shellcheck
        shfmt
        silver-searcher # TODO: find helm-rg solution
        slack
        spotify
        stow
```

⟨*modules/packages.nix* 40⟩+≡   ◁41 43▷
```
    (
      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* 40⟩+≡   ◁42 44▷
```
      tree
      unzip
      vim
      watch
      # wipe
      nixpkgs-unstable.yq
    ]
```

⟨*modules/packages.nix* 40⟩+≡   ◁43 45▷
```
  ) ++ (
    with haskellPackages; [
      cabal-install
      cabal2nix
      hpack
      # FIXME: stylish-haskell
    ]
```

⟨*modules/packages.nix* 40⟩+≡   ◁44 46▷
```
  ) ++ (
    with nodePackages; [
      # TODO: bash-language-server
    ]
```

⟨*modules/packages.nix* 40⟩+≡   ◁45 47▷
```
  ) ++ (
    with python3Packages; [
      pygments
    ]
```

⟨*modules/packages.nix* 40⟩+≡   ◁46 48▷
```
    ) ++ lib.optionals stdenv.isDarwin (
      [
        clementine
        m-cli
        sourcetree
        onyx
      ] ++ (
        with chunkwm; [
          core
          ffm
          tiling
        ]
      )
```

⟨*modules/packages.nix* 40⟩+≡   ◁47 49▷
```
    ) ++ lib.optionals stdenv.isLinux [
      gnome3.networkmanagerapplet
      hicolor-icon-theme
      keybase-gui
      libreoffice
      qpdfview
      playerctl
      psmisc
```

⟨*modules/packages.nix* 40⟩+≡   ◁48 50▷
```
      (
        nixpkgs-unstable.signal-desktop.override {
          spellcheckerLanguage = "en_US";
        }
      )
```

⟨*modules/packages.nix* 40⟩+≡   ◁49 51▷
```
      thunderbird
      volumeicon
      xclip
      xorg.xbacklight
    ];
```

⟨*modules/packages.nix* 40⟩+≡   ◁50
```

  }
```

Location

⟨*modules/location.nix* 52⟩≡   53▷
```
  { config, lib, ... }:
```

⟨*modules/location.nix* 52⟩+≡   ◁52 54▷
```
  let
    airportCode = config.airportCode;
  in
  {
```

⟨*modules/location.nix* 52⟩+≡   ◁53 55▷
```
    options.airportCode = lib.mkOption {
      default = "MSP";
      type = lib.types.enum [ "ATL" /* "LHR" */ "LJU" "MSP" "TRD" ];
    };
```

⟨*modules/location.nix* 52⟩+≡   ◁54 56▷
```
    config = lib.mkMerge [
```

The Sportradar office in Minneapolis.

⟨*modules/location.nix* 52⟩+≡   ◁55 57▷
```
      (
        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* 52⟩+≡   ◁56 58▷
```
      (
        lib.mkIf (airportCode == "ATL") {
          location = {
            latitude = 33.77;
            longitude = -84.37;
          };
          time.timeZone = "America/New_York";
        }
      )
```

The Sportradar office in London.

⟨*modules/location.nix* 52⟩+≡   ◁57 59▷
```
      (
        lib.mkIf (airportCode == "LHR") {
          location = {
            latitude = 51.51;
            longitude = -0.09;
          };
          time.timeZone = "Europe/London";
        }
      )
```

The Sportradar office in Ljubljana.

⟨*modules/location.nix* 52⟩+≡   ◁58 60▷
```
      (
        lib.mkIf (airportCode == "LJU") {
          location = {
            latitude = 46.09;
            longitude = 14.55;
          };
          time.timeZone = "Europe/Ljubljana";
        }
      )
```

The Sportradar office in Trondheim.

⟨*modules/location.nix* 52⟩+≡   ◁59 61▷
```
    (
      lib.mkIf (airportCode == "TRD") {
        location = {
          latitude = 63.43;
          longitude = 10.40;
        };
        time.timeZone = "Europe/Oslo";
      }
    )
```

⟨*modules/location.nix* 52⟩+≡   ◁60
```
    ];

  }
```

NixOS

⟨*modules/nixos.nix* 62⟩≡   63▷
```
  { pkgs, ... }:

  {
```

⟨*modules/nixos.nix* 62⟩+≡   ◁62 64▷
```
    imports = with (import <nurpkgs> { }).modules; [
      pass
      tomb
    ];
```

⟨*modules/nixos.nix* 62⟩+≡   ◁63 65▷
```
    boot = {
      cleanTmpDir = true;
      loader = {
        systemd-boot.enable = true;
        efi.canTouchEfiVariables = true;
      };
    };
```

⟨*modules/nixos.nix* 62⟩+≡   ◁64 66▷
```
    # console.font = "latarcyrheb-sun32";
    console.font = "Lat2-Terminus16";
    console.keyMap = "us";
    i18n.defaultLocale = "en_US.UTF-8";
```

⟨*modules/nixos.nix* 62⟩+≡   ◁65 67▷
```
    location.provider = "manual";
```

⟨*modules/nixos.nix* 62⟩+≡   ◁66 68▷
```
    networking.networkmanager.enable = true;
```

⟨*modules/nixos.nix* 62⟩+≡   ◁67 69▷

```
nixpkgs.config.allowUnfree = true;
```

⟨*modules/nixos.nix* 62⟩+≡   ◁68 70▷

```
programs = {
  pass = {
    enable = true;
    genphrase = true;
    git-helper = true;
    otp = true;
    update = true;
  };
  tomb = {
    enable = true;
    resize = true;
    slam = true;
  };
};
```

⟨*modules/nixos.nix* 62⟩+≡   ◁69 71▷

```
services = {
  fstrim.enable = true;

  logind.lidSwitch = "hibernate";

  kbfs.enable = true;

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

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

⟨*modules/nixos.nix* 62⟩+≡   ◁70 72▷

```
    xserver = {
      enable = true;

      autorun = true;

      displayManager = {
        lightdm.enable = true;
        session = [
          {
            name = "home-manager";
            manage = "window";
            start = "
              ${pkgs.runtimeShell} $HOME/.hm-xsession &
              waitPID=$!
            ";
          }
        ];
      };

      layout = "us";

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

      videoDrivers = [
        "intel"
      ];

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

⟨*modules/nixos.nix* 62⟩+≡   ◁71 73▷

```
  hardware.pulseaudio = {
    enable = true;
    extraModules = [ pkgs.pulseaudio-modules-bt ];
    package = pkgs.pulseaudioFull;
  };
  sound.enable = true;
```

⟨*modules/nixos.nix* 62⟩+≡   ◁72 74▷

```
  system.stateVersion = "20.03";
```

⟨*modules/nixos.nix* 62⟩+≡   ◁73 75▷

```
  virtualisation.docker.enable = true;
```

⟨*modules/nixos.nix* 62⟩+≡   ◁74

```
}
```

Darwin

⟨*modules/darwin.nix* 76⟩≡   77 ▷
```
{ pkgs, ... }:

{
```

⟨*modules/darwin.nix* 76⟩+≡   ◁ 76 78 ▷
```
  nixpkgs.config.allowUnfree = true;
```

⟨*modules/darwin.nix* 76⟩+≡   ◁ 77 79 ▷
```
  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* 76⟩+≡   ◁78 80▷

```
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
  ";
};
```

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

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

⟨*modules/darwin.nix* 76⟩+≡   ◁79 81▷

```
system.defaults.NSGlobalDomain = {
  AppleKeyboardUIMode = 3;

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

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

  NSNavPanelExpandedStateForSaveMode = true;
  NSNavPanelExpandedStateForSaveMode2 = true;
};
```

⟨*modules/darwin.nix* 76⟩+≡   ◁80 82▷

```
system.defaults.dock = {
  autohide = true;
  mru-spaces = false;
  orientation = "right";
  showhidden = true;
};
```

⟨*modules/darwin.nix* 76⟩+≡   ◁81 83▷
```
    system.defaults.finder = {
      AppleShowAllExtensions = true;
      FXEnableExtensionChangeWarning = false;
      QuitMenuItem = true;
    };
```

⟨*modules/darwin.nix* 76⟩+≡   ◁82 84▷
```
    system.defaults.trackpad.Clicking = true;
```

⟨*modules/darwin.nix* 76⟩+≡   ◁83
```
  }
```

Software configuration

Bash

Enable the Bash module, as well as autojump.

⟨*config/bash.nix* 85⟩≡

```
{ ... }:

{

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

}
```

bat

Enable the bat[10] module.

[10] https://github.com/sharkdp/bat

⟨*config/bat.nix* 86⟩≡   87 ▷

```
{ ... }:

{

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

For the pager, use `less`.

⟨*config/bat.nix* 86⟩+≡   ◁ 86 88 ▷

```
    pager = "less -FR";
```

Set the style to `changes`.

⟨*config/bat.nix* 86⟩+≡   ◁ 87 89 ▷

```
    style = "changes";
```

Use the Monokai Extended theme.

⟨*config/bat.nix* 86⟩+≡   ◁ 88 90 ▷

```
    theme = "Monokai Extended";
```

⟨*config/bat.nix* 86⟩+≡   ◁ 89

```
    };
  };

}
```

Browserpass

⟨*config/browserpass.nix* 91⟩≡

```
{ pkgs, ... }:

{

  home.packages = with pkgs; [
    browserpass
  ];

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

}
```

bugwarrior

⟨*config/bugwarrior.nix* 92⟩≡   93 ▷

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

{
```

⟨*config/bugwarrior.nix* 92⟩+≡   ◁ 92 94 ▷

```
  home.packages = with pkgs; [
    bugwarrior
    pass
  ];
```

⟨*config/bugwarrior.nix* 92⟩+≡   ◁ 93 95 ▷

```
  imports = [
    ./taskwarrior
  ];
```

⟨*config/bugwarrior.nix* 92⟩+≡ ◁94 96▷

```
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.baley OR description ~ e.bailey OR c
  jira.version = 8
  jira.add_tags = work
  jira.description_template = {{jiraid}}: {{jirasummary}}
";
```

⟨*config/bugwarrior.nix* 92⟩+≡ ◁95

```
}
```

direnv

⟨*config/direnv.nix* 97⟩≡

```
{ ... }:

{

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

}
```

Emacs

⟨*config/emacs/default.nix* 98⟩≡

```
{ 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* 99⟩≡   100 ▷

```
{ ... }:

{

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

⟨*config/emacs/packages.nix* 99⟩+≡   ◁99 101 ▷

```
    with elpa; [
      mmm-mode
      rainbow-mode
    ]
```

⟨*config/emacs/packages.nix* 99⟩+≡

```
) ++ (
  with melpa; [
    avy
    better-defaults
    clj-refactor
    clojure-mode
    # TODO: company-lsp
    # TODO: cquery
    crux
    dhall-mode
    direnv
    dockerfile-mode
    editorconfig
    elixir-mode
    emojify
    enh-ruby-mode
    erlang
    fill-column-indicator
    fish-mode
    frames-only-mode
    gap-mode
    go-mode
    graphviz-dot-mode
    haskell-mode
    helm-ag
    # TODO: hindent
    hl-todo
    htmlize
    idris-mode
    j-mode
    lfe-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* 99⟩+≡   ◁101
```
    with melpaStable; [
      ess
    ]
  );


  }
```

Init File

⟨*config/emacs/init.el* 103⟩≡   104 ▷
```
  (server-start)
```

⟨*config/emacs/init.el* 103⟩+≡   ◁103 105 ▷
```
  (setq server-window 'pop-to-buffer-same-window)
```

⟨*config/emacs/init.el* 103⟩+≡   ◁104 106 ▷
```
  (column-number-mode 1)
```

⟨*config/emacs/init.el* 103⟩+≡   ◁105 107 ▷
```
  (add-to-list 'exec-path "/run/current-system/sw/bin")
```

⟨*config/emacs/init.el* 103⟩+≡   ◁106 108 ▷
```
  (menu-bar-mode 0)

  (tool-bar-mode 0)
```

⟨*config/emacs/init.el* 103⟩+≡   ◁107 109 ▷
```
  (let ((font "Iosevka-10"))
    (set-face-attribute 'default t :font font)
    (set-frame-font font nil t))
```

⟨*config/emacs/init.el* 103⟩+≡   ◁108 110 ▷
```
  (require 'package)
```

⟨*config/emacs/init.el* 103⟩+≡   ◁109 111 ▷
```
  (setq-default frames-only-mode t
                indent-tabs-mode nil
                package-archives nil
                package-enable-at-startup nil)
```

⟨*config/emacs/init.el* 103⟩+≡   ◁110 112 ▷
```
  (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* 103⟩+≡   ◁111 113▷
```
(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

⟨*config/emacs/init.el* 103⟩+≡   ◁112 114▷
```
(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* 103⟩+≡   ◁113 115▷
```
(dolist (pattern '("^\\(/dev/shm/\\|/tmp/\\)"
                   "\\.\\(enc\\|gpg\\|hashedPassword\\)$"))
  (add-to-list 'auto-mode-alist (cons pattern 'sensitive-mode)))
```

⟨*config/emacs/init.el* 103⟩+≡   ◁114 116▷
```
(setq custom-file "~/.emacs.d/private/local/custom.el")
```

⟨*config/emacs/init.el* 103⟩+≡   ◁115 117▷
```
(load-theme 'wombat)
```

⟨*config/emacs/init.el* 103⟩+≡   ◁116 118▷
```
(global-set-key (kbd "C-x C-k") 'kill-this-buffer)
(global-set-key (kbd "s-u") 'revert-buffer)
```

⟨*config/emacs/init.el* 103⟩+≡   ◁117 119▷
```
(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* 103⟩+≡   ◁118 120▷
```
(setq-default fill-column 80)
```

⟨*config/emacs/init.el* 103⟩+≡   ◁119 121▷
```
(org-babel-do-load-languages
 'org-babel-load-languages
 '((shell . t)))
```

⟨*config/emacs/init.el* 103⟩+≡   ◁120 122▷
```
(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* 103⟩+≡   ◁121 123▷
```
(use-package crux
  :demand
  :config (global-set-key (kbd "C-a") 'crux-move-beginning-of-line))
```

⟨*config/emacs/init.el* 103⟩+≡   ◁122 124▷
```
(use-package editorconfig
  :ensure t
  :config
  (editorconfig-mode 1))
```

⟨*config/emacs/init.el* 103⟩+≡   ◁123 125▷
```
(use-package emojify
  :demand
  :config
  (global-emojify-mode)
  (global-emojify-mode-line-mode))
```

⟨*config/emacs/init.el* 103⟩+≡   ◁124 126▷
```
(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* 103⟩+≡   ◁125 127▷
```
(use-package hl-todo
  :demand
  :config (global-hl-todo-mode t))
```

⟨*config/emacs/init.el* 103⟩+≡   ◁126 128▷
```
(use-package magit
  :demand
  :config (global-magit-file-mode t))
```

⟨*config/emacs/init.el* 103⟩+≡   ◁127 129▷
```
(use-package multiple-cursors
  :demand
  :config (global-set-key (kbd "C-S-c C-S-c") 'mc/edit-lines))
```

⟨*config/emacs/init.el* 103⟩+≡   ◁128 130▷
```
(use-package nix-mode
  :mode ("\\.nix\\'"))
```

⟨*config/emacs/init.el* 103⟩+≡   ◁129 131▷
```
(use-package noweb-mode
  :load-path "/run/current-system/sw/share/emacs/site-lisp"
  :mode ("\\.nw\\'")
  :demand)
```

⟨*config/emacs/init.el* 103⟩+≡   ◁130 132▷
```
(use-package robe
  :demand
  :mode ("\\.rb\\'"))
```

⟨*config/emacs/init.el* 103⟩+≡   ◁131 133▷
```
(use-package rvm
  :ensure t)
```

⟨*config/emacs/init.el* 103⟩+≡   ◁132 134▷
```
(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* 103⟩+≡   ◁133 135▷
```
(use-package tuareg
  :mode ("\\.ml\\'" "\\.mli\\'"))
```

⟨*config/emacs/init.el* 103⟩+≡   ◁134
```
(use-package whitespace-cleanup-mode
  :demand
  :config (global-whitespace-cleanup-mode t))
```

Firefox

⟨*config/firefox.nix* 136⟩≡

```nix
{ pkgs, ... }:

{

  programs.firefox = {
    enable = true;
    extensions = with (import <nur> { inherit pkgs; }).repos.rycee.firefox-addons; [
      browserpass
      browserpass-otp
      darkreader
      https-everywhere
      lastpass-password-manager
      privacy-badger
    ];
    profiles = {
      default = {
        settings = {
          # http://kb.mozillazine.org/About:config_entries
          "browser.ctrlTab.recentlyUsedOrder" = false;
          "browser.newtabpage.activity-stream.asrouter.userprefs.cfr.addons" = false;
          "browser.newtabpage.activity-stream.asrouter.userprefs.cfr.features" = false;
          "browser.newtabpage.activity-stream.feeds.section.topstories" = false;
          "browser.newtabpage.activity-stream.feeds.snippets" = false;
          # FIXME: "browser.newtabpage.activity-stream.improvesearch.topSiteSearchShortcuts.havePinned" = null;
          "browser.newtabpage.activity-stream.section.highlights.includePocket" = false;
          "browser.newtabpage.activity-stream.showSearch" = false;
          # FIXME: "browser.newtabpage.pinned" = "[]";
          # FIXME: "browser.search.defaultenginename" = "DuckDuckGo";
          # FIXME: "browser.search.defaulturl" = "https://duckduckgo.com/?q=";
          "browser.search.hiddenOneOffs" = "Google,Bing,Amazon.com,eBay,Twitter,Wikipedia (en)";
          # FIXME: git "browser.search.selectedEngine" = "DuckDuckGo";
          "browser.search.suggest.enabled" = false;
          "browser.startup.page" = 3;
          "browser.tabs.unloadOnLowMemory" = true;
          "browser.urlbar.placeholderName" = "DuckDuckGo";
          "extensions.activeThemeID" = "firefox-compact-dark@mozilla.org";
          "font.size.variable.x-western" = 12;
          "signon.rememberSignons" = false;
        };
      };
    };
  };

}
```

fish

⟨*config/fish/default.nix* 137⟩≡

```
{ 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."[11]

⟨*config/fish/abbrs.nix* 138⟩≡
```
{ ... }:

{

  programs.fish.shellAbbrs = {
    ⟨AWS Azure login abbreviation 139⟩
    ⟨Kubernetes fish abbreviations 140⟩
    ⟨Nix fish abbreviations 141⟩
    ⟨ripgrep fish abbreviations 142⟩
  };

}
```

[11] abbr - manage fish abbreviations. URL https://fishshell.com/docs/current/cmds/abbr.html

NAAL[12] IS COOL AND ALL, BUT...

... it's a bad habit too. the `aal` abbeviation makes being not so sneaky a bit less annoying.

⟨*AWS Azure login abbreviation* 139⟩≡   (138)
```
aal = "aws-azure-login —no-prompt";
```

[12] https://github.com/yurrriq/naal

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

⟨*Kubernetes fish abbreviations* 140⟩≡   (138)
```
kc = "kubectl";
kcd = "kubectl drain —delete-local-data —ignore-daemonsets";
kcnp = "kubectl get pods —field-selector=spec.nodeName=";
kcx = "kubectl —context";
kgno = "kubectl get nodes";
kn = "kubens";
kns = "kubens";
kt = "stern";
kx = "kubectx";
```

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

⟨*Nix fish abbreviations* 141⟩≡   (138)
```
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."[13]

[13] Andrew Gallant. rg - man page. URL https://www.mankier.com/1/rg

⟨*ripgrep fish abbreviations* 142⟩≡   (138) 143 ▷
```
rg = "rg -S";
```

⟨*ripgrep fish abbreviations* 142⟩+≡   (138) ◁142 144 ▷
```
rga = "rg –hidden –iglob !.git";
rgf = "rg -F";
```

"Never print the file path with the matched lines", and "[s]uppress line numbers."[14]

[14] Andrew Gallant. rg - man page. URL https://www.mankier.com/1/rg

⟨*ripgrep fish abbreviations* 142⟩+≡   (138) ◁143 145 ▷
```
rgin = "rg -IN";
```

```
λ rg -IN name: helmfile.d/*.yaml | sort -u
- name: cert-manager
- name: cluster-autoscaler
- name: elastalert
- 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* 142⟩+≡   (138) ◁144
```
rgn = "rg –no-heading";
```

⟨*config/fish/aliases.nix* 146⟩≡
```
{ ... }:

{

  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 147⟩≡

```
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 148⟩≡   149 ▷

```
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* 148⟩+≡   ◁148 150▷

```
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* 148⟩+≡   ◁149 151▷

```
command -sq aws; and command -sq jq; and \
function describe-cert -d 'List the domains for a given ACM certificate'
    test (count $argv) -ne 1; and return
    aws acm describe-certificate --certificate-arn $argv[1] |
    jq -r '.Certificate | .SubjectAlternativeNames[]'
end
```

⟨*config/fish/interactiveShellInit.fish* 148⟩+≡   ◁150 152▷

```
command -sq fluidsynth; and function playmidi
    fluidsynth -i ~/lib/arachno-soundfont/Arachno\ SoundFont\ -\ Version\ 1.0.sf2 $argv
end
```

⟨*config/fish/interactiveShellInit.fish* 148⟩+≡   ◁151 153▷

```
command -sq kitty; and function icat
    kitty +kitten icat $argv
end
```

⟨*config/fish/interactiveShellInit.fish* 148⟩+≡   ◁152 154▷

```
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

    function kcterm -d 'Terminate a Kubernetes node'
        test (count $argv) -ne 1; and return
        kubectl get node -o jsonpath='{.spec.providerID}' $argv[1] |
        cut -d'/' -f5 |
        xargs aws ec2 terminate-instances --instance-ids
    end
end
```

⟨*config/fish/interactiveShellInit.fish* 148⟩+≡   ◁153 155▷

```
# FIXME: functions rvm >/dev/null 2>&1; and rvm default
```

⟨*config/fish/interactiveShellInit.fish* 148⟩+≡   ◁154 156▷

```
set fish_greeting
```

⟨*config/fish/interactiveShellInit.fish* 148⟩+≡   ◁155
```
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* 157⟩≡

```
function fish_prompt
    set -l code $status

    if set -q IN_NIX_SHELL
        set symbol "λΠ "
    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)
```

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

⟨*config/fish/sushi/fish_right_prompt.fish* 158⟩≡   159 ▷

```
# ────────────────────── [ 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* 158⟩+≡

```fish
# ───────────────────────── [ 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* 158⟩+≡

```fish
# ───────────────────────── [ 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* 158⟩+≡

```fish
# ───────────────────── [ 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* 158⟩+≡   ◁161

```fish
# ───────────────────── [ 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* 163⟩≡

```nix
{ ... }:

{

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

}
```

Git

⟨*config/git/default.nix* 164⟩≡

```
{ 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* 165⟩≡

```
{ 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* 166⟩≡

```
{ 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;
    };

    pull.ff = "only";

    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* 167⟩≡

```nix
{ 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 {
    ap = "add -p";
    bm = "branch --merged";
    bnm = "branch --no-merged";
    ca = "commit --amend";
    cam = "${ca} -m";
    can = "${ca} --no-edit";
    cans = "${can} -S";
    cas = "${ca} -S";
    casm = "${cas} -m";
    cm = "commit -m";
    cnm = "commit -nm";
    co = "checkout";
    cob = "${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* 168⟩≡
```
  { 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 = 28800;
      enableSshSupport = true;
      maxCacheTtl = 28800;
    };

  }
```

htop

⟨*config/htop.nix* 169⟩≡
```
  { ... }:

  {

    programs.htop.enable = true;

  }
```

i3

⟨*config/i3/default.nix* 170⟩≡
```
  { config, lib, pkgs, ... }:

  {
```

⟨*config/i3/default.nix* 170⟩+≡

```
    home.packages = with pkgs; [
      flameshot
      i3lock
      rofi
      rofi-pass
    ];
```

To use i3status-rust:

```
{ pkgs, ... }:

{

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

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

}
```

To use the custom status bar with Spotify metadata :   ⌐ link

```
{ pkgs, ... }:

{

  home.file.".local/bin/i3status".source = ./i3status;

  xsession.windowManager.i3.bars = [
    {
      # fonts = [ ... ];
      statusCommand = "${config.home.homeDirectory}/.local/bin/i3status";
      # colors = { ... };
    }
  ];
}
```

⟨*config/i3/default.nix* 170⟩+≡

```
    services.picom.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
```

Use Solarized Dark `colors`.[15]

⟨*config/i3/default.nix* 170⟩+≡   ◁172 174▷

```nix
        colors = rec {
          background = "#001e26"; # "#002b36";
          focused = {
            border = "#001e26"; # "#002b36";
            background = "#586e75";
            text = "#fdf6e3";
            indicator = "#268bd2";
            childBorder = ""; # "#285577";
          };
          focusedInactive = {
            border = "#001e26"; # "#002b36";
            background = "#073642";
            text = "#839496";
            indicator = "#073642";
            childBorder = ""; # "#5f676a";
          };
          unfocused = {
            border = "#001e26"; # "#002b36";
            background = "#073642";
            text = "#839496";
            indicator = "#073642";
            childBorder = ""; # "#222222";
          };
          urgent = {
            border = "#001e26"; # "#002b36";
            background = "#dc322f";
            text = "#fdf6e3";
            indicator = "#001e26"; # "#002b36";
            childBorder = ""; # "#900000";
          };
        };
        barColors = {
          inherit (colors) background;
          focusedWorkspace = {
            border = "#b58900";
            background = "#b58900";
            text = "#001e26"; # "#002b36";
          };
          inactiveWorkspace = {
            border = "#073642";
            background = "#001e26"; # "#002b36";
            text = "#839496";
          };
          separator = "#586e75";
          statusline = "#839496";
          urgentWorkspace = {
            border = "#dc322f";
            background = "#dc322f";
            text = "#fdf6e3";
          };
        };
```

⟨*config/i3/default.nix* 170⟩+≡   ◁173

```nix
        modifier = "Mod4";
    in
    {
      bars = [
        {
          fonts = [ "Iosevka Term" ];
          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+grave" = "move scratchpad";
        "${modifier}+Shift+minus" = "move scratchpad";
        "${modifier}+grave" = "scratchpad show";
        "${modifier}+minus" = "scratchpad show";
        "${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;
      startup = [
        {
          command = "volumeicon";
          notification = false;
        }
      ];
    };
  };

}
```

i3status

Use some Solarized Dark colors for good, bad, anddegraded.[16]

⟨*config/i3status/config* 175⟩≡   176 ▷
```
  general {
      colors = true
      color_good = "#859900"
      color_bad = "#dc322f"
      color_degraded = "#cb4b16"
      interval = 5
      output_format = "i3bar"
  }
```

⟨*config/i3status/config* 175⟩+≡   ◁175 177 ▷
```
  order += "path_exists VPN"

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

I usually don't care about SSID.

```
order += "wireless _first_"

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

⟨*config/i3status/config* 175⟩+≡   ◁176 178 ▷
```
  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* 175⟩+≡   ◁177 179 ▷
```
      integer_battery_capacity = true
```

Use the last full capacity, instead of the design capacity. Refer to the man page[17] for more details.

⟨*config/i3status/config* 175⟩+≡   ◁178 180 ▷
```
      last_full_capacity = true
```

⟨*config/i3status/config* 175⟩+≡   ◁179 181▷

```
    low_threshold = 10
    status_bat = ""
    status_chr = "⚡"
    status_full = ""
    status_unk = ""
    threshold_type = "time"
}
```

```
order += "tztime singapore"
```

```
tztime singapore {
    format = "  %a %H:%M"
    hide_if_equals_localtime = true
    locale = "en_SG.utf8"
    timezone = "Asia/Singapore"
}
```

```
order += "tztime london"
```

```
tztime london {
    format = "  %a %H:%M"
    hide_if_equals_localtime = true
    locale = "en_GB.utf8"
    timezone = "Europe/London"
}
```

⟨*config/i3status/config* 175⟩+≡   ◁180 182▷

```
order += "tztime trondheim"
```

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

⟨*config/i3status/config* 175⟩+≡   ◁181

```
order += "tztime local"
```

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

jq

⟨*config/jq.nix* 183⟩≡

```
{ ... }:

{

  programs.jq.enable = true;

}
```

⟨*config/kitty.nix* 184⟩≡
```
{ pkgs, ... }:

{

  programs.kitty = {
    enable = true;
    extraConfig = "
      include theme.conf
    ";
    font = {
      name = "Iosevka Term";
      package = (pkgs.nerdfonts.override { withFont = "Iosevka"; });
    };
    keybindings = {
      "kitty_mod+enter" = "new_window_with_cwd";
      "kitty_mod+k" = "
        combine : clear_terminal scrollback active : send_text normal \x0c
      ";
    };
    settings = {
      editor = "emacsclient -nw -a """;
      font_size = 24;
      kitty_mod = "ctrl+shift";
      scrollback_lines = -1;
      shell = ".";
      term = "xterm";
    };
  };

  xdg.configFile."kitty/theme.conf".source = builtins.fetchurl {
    url = "https://raw.githubusercontent.com/dexpota/kitty-themes/c4bee86ce8372d3426620e27086d6afa6873a98c/themes/S
    sha256 = "0vwjcyd0asqrbs4437k2b80lqlanl79q03kk4pjnfdf379jklrm4";
  };

}
```

man

⟨*config/man.nix* 185⟩≡
```
{ ... }:

{

  programs.man.enable = true;

}
```

nixpkgs

⟨*config/nixpkgs/default.nix* 186⟩≡
```
{ ... }:

{

  nixpkgs.config = import ./nixpkgs-config.nix;

  xdg.configFile."nixpkgs/config.nix".source = ./nixpkgs-config.nix;

}
```

⟨*config/nixpkgs/nixpkgs-config.nix* 187⟩≡
```
{
  allowBroken = false;
  allowUnfree = true;
  allowUnsupportedSystem = false;
}
```

rebar3

⟨*config/rebar3.nix* 188⟩≡
```
{ ... }:

{

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

}
```

Taskwarrior

Config

⟨*config/taskwarrior/default.nix* 189⟩≡   190 ▷
```
{ config, lib, pkgs, ... }:

{
```

SINCE HOME-MANAGER GENERATES A READ-ONLY `.taskrc` FILE, and
using contexts[18] requires Taskwarrior to be able to modify the it, use
the following hacky workaround.

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

⟨*config/taskwarrior/default.nix* 189⟩+≡   ◁189 191▷

```
home.sessionVariables = {
  TASKRC = "~/.taskrc-dirty";
};
```

DEFINE short Taskwarrior shell aliases.

⟨*config/taskwarrior/default.nix* 189⟩+≡   ◁190 192▷

```
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";
};
```

[18] https://taskwarrior.org/docs/
context.html

CONFIGURE TASKWARRIOR, using home-manager[19].

⟨*config/taskwarrior/default.nix* 189⟩+≡   ◁191 193▷

```
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. [20]

⟨*config/taskwarrior/default.nix* 189⟩+≡   ◁192

```
xdg.dataFile."task/hooks/on-exit-git.sh" = {
  executable = true;
  source = ./on-exit-git.sh;
};

}
```

[20] See next subssection.

on-exit Git hook

FIRST, use a fairly portable Bash shebang, and be safe.[21]

⟨*config/taskwarrior/on-exit-git.sh* 194⟩≡   195 ▷

```
#! /usr/bin/env bash

set -eufo pipefail
```

Run in debug mode[22], if the environment variable DEBUG is nonempty.

⟨*config/taskwarrior/on-exit-git.sh* 194⟩+≡   ◁194 196 ▷

```
if [ -n "${DEBUG:-}" ]; then
    set -x
fi
```

[21] `-e`xit immediately upon failure, treat `-u`nset variables as an error, disable `-f`ile globbing, and fail if any part of a pipeline fails (`-o pipefail`).

[22] i.e. echo commands

PARSE the command line arguments.

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

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

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

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

⟨*config/taskwarrior/on-exit-git.sh* 194⟩+≡   ◁195 197 ▷

```
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* 194⟩+≡   ◁196 198 ▷

```
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` `exit`s 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* 194⟩+≡   ◁197 199 ▷

```
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* 194⟩+≡   ◁198 200▷

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

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

⟨*config/taskwarrior/on-exit-git.sh* 194⟩+≡   ◁199

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

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

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."[23] 6

[23] 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