

GitHub ❤️ ~/

Why would I want my dotfiles on GitHub?

- **Backup, restore**, and **sync** the prefs and settings for your toolbox. Your dotfiles might be the most important files on your machine.
- **Learn** from the community. Discover new tools for your toolbox and new tricks for the ones you already use.
- **Share** what you've learned with the rest of us.

Get started with a bootstrap

If you're just starting out, before you go symlinking everything in ~/*, you may want to check out a few bootstrap projects that take the heavy lifting out of the process. Here are a few of our favorites:

- **Abdullah's dotfiles**
- **Abhishek Keshri's Dotfiles** has configurations for zsh with custom **Powerlevel9k** prompt, vim, git, tmux, compiz etc. It also has a branch for **Termux**. Comes with a handy setup script.
- **Adam Eivy's dotfiles** are focused on Automation (no manual install/config) for ZSH and macOS with a friendly bot to guide your way.
- **Aditya Pillai's Jumpstart** is a *user-tailored customizable* bootstrap installation script for macOS (formerly Mac OS X). Jumpstart is a fully-fledged tool that quickly installs a development setup to your macOS computer and replaces the need for manually setting up dotfiles. Jumpstart implements the best practices and setups for all-around development environment.
- **Anders Ballegaard's dotfiles** Mostly custom aliases I find useful for network engineering, and Python development. Oh and it updates automatically
- **Andrew Schwartzmeyer's dotfiles** use GNU Stow (like xero's) for symlink management, and `git-subtree` for repository integration.
- **Artem Sapegin's dotfiles** with custom ZSH and Terminal/iTerm themes and useful aliases for web

developers.

- **Ashish Bhatia's dotfiles** focus on Android development, reverse engineering and blockchain (crypto currencies) development on macOS.
- **Awesome dotfiles** contains articles, tools and other resources to get you up to speed with dotfiles.
- **bashdot** A minimalist dotfile management framework, written 100% in bash.
- **Ben Alman's dotfiles** support different configurations per OS, linking, copying and environment setup.
- **Bradford Dabbs's dotfiles** uses a simple shell script to setup **bash-it** and **vim-plug** along with the solarized color scheme and aliases for **Bro NSM**.
- **Darryl Abbate's dotfiles** can be installed with a **single curl command** for absolutely effortless setup on a fresh macOS machine.
- **Diki Ananta's dotfiles** is focused on Window Manager users, especially for **i3**, and Web developers. It has various configs for standard of window manager.
- **dotphiles** are a community driven framework of dotfiles, for the usual terminal apps and shells, designed to work across multiple platforms and degrade for older versions of software or OS, allowing you to use the same settings on all your machines.
- **Dries Vints' dotfiles** leverages **Brew** and **mackup** to setup an entire macOS environment.
- **Eduardo Lundgren's dotfiles**, the first JavaScript-based dotfiles powered by Grunt.
- **F-dotfiles** is an opinionated dotfiles organization scheme based on GNU Stow. Highest priorities are ease of maintenance and deployment on both Linux and macOS.
- **Felix Volny's dotconfig** is a one-script install to setup a complete JavaScript development environment on MacOS, including tmux, neovim, ZSH, and iterm2 configuration.
- **Huy Vo's dotfiles**, a lightweight yet powerful dotfiles for MacOS users. It uses GNU Stow to manage the symlinks,

ZSH as the shell environment and Nvim + Tmux for the coding (along with other useful tools).

- **Ian Langworth's dotfiles** are designed to work on any platform and be kept in sync across different hosts. With plenty of useful shell aliases and command defaults, it features **a simple install script** to install itself and stay up-to-date.
- **John Wyatt's Chef Dotfile Manager Tutorial** - This is a complete bootstrap to get a Chef dotfile manager running on a Linux system with examples for packages, templates, and setting up the Atom text editor + Atom packages + Atom configuration.
- **Jeff Coffler's dotfiles** has a bootstrap script that symlinks and doesn't require "." (hidden file) in the repo. The repo itself can live anywhere.
- **jh3y's kody** is a dotfiles runner/manager written with node inspired by Zach Holman's popular dotfiles.
- **Jonas Devlieghere's dotfiles** for both macOS and Linux has a little bit of everything for compiler development in the terminal.
- **kutsan's dotfiles** includes ongoing configuration files for various interfaces and text-based command-line applications such as vim, ZSH, tmux, ranger, mutt, newsboat and more.
- **Mathias Bynens' dotfiles** includes a bootstrap script that rsyncs your repo to your home folder. Mathias' **macOS defaults script** is legendary.
- **Matt Smith's dotfiles** includes a one-liner install, fish, vscode, mac desktop app installs via brew cask, os x customizations inspired by Mathias, etc. No dotfiles framework, just shell scripts to set everything up.
- **mihaliak's dotfiles** for macOS focused on Web development
- **Mohit Singh's dotfiles** has a lightweight setup for JavaScript development environment on macOS.
- **neenasade's dotfiles** utilize a template based theming system, with the ability to switch between themes without

restarting programs.

- **Nelson Estevão's dotfiles** uses a modular repository structure. It features configuration files for popular software like neovim, ZSH and i3wm. Just delete the directories that you are not interested and run `install.sh`.
- **Nick Plekhanov's dotfiles** features properly customized ZSH and iTerm environments, along with Atom editor and Webstorm IDE. As a bonus, included is a set of useful aliases for web developers.
- **Nikita Sobolev's dotfiles** contains Python, Node, and Elixir configurations for macOS alongside with the most user-friendly command line tools for the developer happiness.
- **Overbryd's dotfiles** features a 0-100 macOS bootstrap and topical setup maintained with a simple Makefile
- **Paul Miller's dotfiles** feature greatly customized ZSH with auto-completion and syntax highlighting, a bunch of useful Git extras and colourful themes for macOS Terminal and Sublime Text.
- **posquit0's dotfiles** contains awesome configurations for CLI commands and X environments, along with powerfully customized Vim, ZSH and Tmux environments for nerds.
- **ridhwaans's dotfiles** has dotfiles and a ZSH environment setup built for macOS High Sierra and WSL (Windows 10 Creators Update with Ubuntu 17.10 (Artful Armadillo))
- **Rosco Kalis' dotfiles** feature Fish shell configuration with custom completions, as well as comprehensive package management, repository management and Hammerspoon configuration.
- **Voku's dotfiles** dotfiles for Bash (Linux) / ZSH (Linux) / Git Bash (Windows) / Cygwin (Windows) / Bash on Ubuntu on Windows.
- **xero's dotfiles** are managed with **GNU Stow**, a free, portable, lightweight symlink farm manager.
- **Yan Pritzker's dotfiles** bundles an opinionated set of Vim plugins and ZSH setup all tuned for using **Solarized** on macOS.

- **Zach Holman's dotfiles** features topical organization, auto sourcing ZSH files, easy ZSH completion extensions, and a local bin folder for executables. The included Rakefile will symlink anything ending in .symlink to your ~ folder.

Go further with a framework

For a lot of us, a big chunk of our ~ folder is devoted to our shell and our text editor. Until you're ready to roll your own setup, these projects make customization safe and easy.

Shells

- **Liquid Prompt** is a full-featured and carefully designed adaptive prompt for Bash and ZSH.

Bash

- **bash-it** is a “shameless ripoff of oh-my-zsh,” but for bash.

Fish

- **Fisher** A package manager for the **fish shell**.
- **Oh My Fish** is a fish-shell framework inspired by oh-my-zsh.

ZSH

Frameworks

- **Antibody** is a shell plugin manager made from the ground up thinking about performance.
- **antigen-hs** is an antigen-inspired ZSH plugin manager that tries to do work statically and only on manual invocation, minimizing the ZSH startup time. Antigen-hs is much more minimalistic and emphasizes convention over configuration more strongly than antigen.
- **antigen** is a framework for using plugins and themes in your ZSH configuration. It will automatically clone repositories containing the plugins you're using without you having to manually create submodules or clones, and supports using **oh-my-zsh** plugins and themes as well as ones published as separate repositories.
- **chezmoi** makes it easy to manage your dotfiles across multiple machines, securely.
- **dotzsh** strives to be platform and version independent,

some functionality may be lost when running under older versions of ZSH, but it should degrade cleanly and allow you to use the same setup on multiple machines of differing OS's without problems.

- **oh-my-zsh** is a community-driven framework for managing your ZSH configuration. It bundles 40+ plugins and 80+ themes.
- **Prezto** is a configuration framework for ZSH. It's a lightweight alternative to oh-my-zsh with sane defaults, aliases, functions, auto completion, prompt themes and dozens of well documented modules.
- **slimzsh** is a small starter framework for ZSH that features the pure prompt, syntax highlighting and tab completion.
- **toasty-zsh** is a simplistic ZSH plugin and configuration system that aims to provide a "sane defaults" set for any ZSH config and is optimized for system-wide installation
- **zgen** is a lightweight plugin manager for ZSH inspired by Antigen. The goal is to have a minimal overhead when starting up the shell because nobody likes waiting. Zgen will also automatically handle cloning repositories for plugins you're using without you manually maintaining submodules or clones, and can use oh-my-zsh formatted plugins and themes.
- **zplug** is a next-generation plugin manager for ZSH. There is more to a ZSH plugin manager than increasing its speed. Also, there is nothing that zplug cannot do.

ZSH Resources

- **awesome-zsh-plugins** is a list of ZSH plugins, themes and completions compatible with Zsh frameworks like **antigen**, **oh-my-zsh** and **zgen**.
- **zsh-quickstart-kit** A quickstart for zgen. It includes zgen setup, a starter list of plugins, the **zsh-users/zsh-completions** collection, and the **bullet-train** theme.

Editors

Emacs

- **Cask** is a package manager for Emacs.
- **Prelude** is an enhanced Emacs 24 distribution that should

make your experience with Emacs both more pleasant and more powerful.

- **Spacemacs** is a Emacs 24 distribution that builds on Evil-mode with ports of popular Vim plugins to closer emulate a Vim environment.
- **use-package** is a declaration macro for simplifying your `.emacs`

Vim

- **dotvim** is a community driven framework for vim.
- **Janus** is a distribution of plugins and mappings for Vim, Gvim and MacVim.
- **Neobundle** is a next generation Vim plugin manager.
- **Pathogen** is a plugin manager for Vim.
- **spf13-vim** is Steve Francia's Vim distribution of vim plugins and resources for Vim, Gvim and MacVim.
- **vim-plug** is a simple plugin manager that supports parallel installations / upgrades.
- **Vundle** is short for *Vim Bundle* and is a plugin manager for Vim. It works with Pathogen compatible vim plugins.

General-purpose dotfile utilities

- **Bonclay** is a simple dotfiles manager. Well, technically, it is a backup/restore/sync tool magiggy.
- **Config Curator** defines your dotfile installation in a simple but powerful config file. Sync files, directories, and symlinks across multiple environments with this ludicrous-speed Node.js successor to the original **Ruby version**.
- **Dotbot** is a lightweight standalone tool to bootstrap dotfiles, making it easy to have a “one click” installation/upgrade process for your dotfiles.
- **dotdrop** makes the management of dotfiles between different hosts easy. It allows to store your dotfiles on git and automagically deploy different versions on different setups.
- **dotfiler** is inspired by homesick and **Zach Holman's dotfiles**, made using principle of KISS.
- **dotfiles.sh** by Trevor King. Dotfiles.sh manages multiple **dotfile repositories** so you can keep your public and

private configuration separate. It also supports locally patching your dotfiles before symlinking to adapt to the local machine. Dotfile repositories may be fetched via either Git or (where Git is not available) wget.

- **dotgit** by Kobus van Schoor. A comprehensive and versatile dotfiles manager which helps to synchronize your dotfiles between multiple computers and devices.
- **Dotsync** utility for syncing dotfiles between multiple machines from a Git repo or push using rsync.
- **dotty** by Vibhav Pant. *dotty* uses a simple JSON-formatted file to symlink and/or copy your dotfiles. It can also execute shell commands and check for directories, and create them accordingly.
- **Ellipsis** is a package manager for dotfiles.
- **exogenesis** by Lucas Dohmen. *exogenesis* uses a simple YAML-formatted file to symlink your dotfiles and install npms/bundlers.
- **fresh** is a tool to source dotfiles from others into your own. It supports shell configuration (aliases, functions, etc.) as well as config files (e.g. *ackrc* and *gitconfig*). Think of it as *Bundler for your dotfiles*.
- **Ghar** by Brandon Philip. Ghar is a standalone Python script for managing Git repos symlinked into your home.
- **GNU Stow** is a symlink farm manager, useful for automatically (and safely) linking your dotfiles folder into your home directory.
- **Homely** helps you script your dotfile installation using Python. If you're getting frustrated by the limitations of pure shell scripts, then this is the tool for you. Homely also has a clever Automatic Cleanup feature and good tutorials.
- **Home Manager** by Robert Helgesson is a system built for managing **NixOS** user environments using the **Nix** package manager and the **Nixpkgs** libraries.
- **Homemaker** by Alex Yatskov. Homemaker is a standalone tool written in Golang to manage both common and machine-specific dotfile settings. Homemaker can be

configured in TOML, YAML or JSON.

- **Homeshick** by Anders Ingemann is like Homeshick but written in bash. Great to combine with **myrepos**.
- **Homesick**, by Josh Nichols. Homeshick makes it easy to symlink and clone dotfiles repos.
- **myrepos** is a tool to manage all your version control repositories at once.
- **Pearl** is a brand new revolutionary package manager that allows to automatically activate dotfiles whenever shells or editors start via a smart hook mechanism. Dotfiles are treated as packages that coexist together seamlessly and can be fully controlled and synced across different systems. There is a wide range of packages already available in the **Official Pearl Hub**.
- **rcm** is a set of well-documented shell scripts that help manage your dotfiles. It is easily installable on macOS with the homebrew package manager, but works on all Unix operating systems.
- **shallow-backup** - `shallow-backup` lets you easily create lightweight backups of installed packages, applications, fonts and dotfiles, and automatically push them to a remote Git repository.
- **shprofile** - `shprofile` manages a set of shell profiles which can be enabled at any time. Script execution is done within the current shell session, so scripts can modify the current shell environment.
- **stowsh** is a clone of GNU Stow written in bash with minimal dependencies.
- **vcsh** by Richard “RichiH” Hartmann. `vcsh` manages all your dotfiles in Git without the need for symlinks. Any number of Git repositories will co-exist in parallel in your `$HOME` without interfering with each other. Advanced use cases with different branches for different systems are supported by default. An extensive hook system lets you customize your repositories. `vcsh` includes batch push, pull, and status commands which operate on all your repositories at once.

- **yadm** by Tim Byrne. *yadm* is a dotfile management tool with 3 main features: Manages files across systems using a single Git repository. Provides a way to use alternate files on a specific OS or host. Supplies a method of encrypting confidential data so it can safely be stored in your repository.
- The **Super User Spark** is a declarative domain specific language that allows you to specify the deployment of your dotfiles. It also includes a compiler and interpreter for the language.
- **yadoma** by Valentin Haenel. *yadoma* is a simple dotfile management tool that uses YAML as configuration syntax and the pile-of-symlinks approach. It can manage multiple hosts, supports variable symlink targets and is infinitely hackable.

Tutorials

- Abdullah wrote a **How to manage dotfiles securely**, a step by step guide to encrypt your dotfiles using your gpg keys.
- Lars Kappert wrote a **tutorial** about getting started using a dotfiles repository.
- Anish Athalye wrote a **guide on dotfiles management** highlighting organizational approaches, installation tools, and general tips and tricks.
- Wes Bos has a series of **free videos** introducing ZSH, oh-my-zsh, and z.
- Dries Vints wrote **an extensive blog post** on why and how he created **his dotfiles**.
- Michael J. Smalley wrote a **popular blog post** with an associated **git repo** that includes a dotfiles installer script.
- Rosco Kalis wrote an article about **setting up dotfiles**, and a second one specifically about his **automated repository management**.
- Dmitry Geurkov wrote an article about his tool called **dotcentral**.

Don't ignore your .gitignore

GitHub has a great [collection of .gitignore templates](#) for a wide range of languages and editors. We recommend Simon Whitaker's [gitignore-boilerplates](#) to help you manage them.

Embrace submodules / subtrees

Consider using [Git submodules](#) as you start to add 3rd party frameworks, scripts, and plugins. Submodules make managing dotfile dependencies so much easier.

If you get fed up with submodules, many people prefer [git-subtree](#), which lets you merge subtrees (other repositories) into one Git repository, and later split and push changes back out.