Daniel Stoddart

A blog by a full stack web developer from Philadelphia.

Managing dotfiles with chezmoi

Dotfiles. Don't leave home without 'em

Over the past five years or so, I've realised the benefits and power of keeping the most important and configured dotfiles on my system under version control.

Simply put, managing your dotfiles with version control keeps historical versions of your configuration files so that if something goes wrong or you accidentally overwrite a crucial setting, you can easily and trivially revert to a previous state of the file.

Doing this offers many benefits:

- Backup and recovery
- Consistent configuration across multiple systems
- Tracking changes
- Ease of migration
- Collaboration and sharing
- Customizability
- Automation

Version controlling your dotfiles adds a layer of robustness and flexibility to managing your system configuration. Dotfile management gives you the combined benefit of a consistent environment everywhere with an undo command and a restore from backup.

A few days ago, I discovered chezmoi, a command-line tool designed to manage dotfiles across different machines. It works by creating a repository of your dotfiles and handling the synchronisation and deployment of these files to your systems. This makes it easier to keep your configurations in sync, apply updates, and share settings across multiple machines.

It turns out that chezmoi is an amazing piece of work by a group of developers who deeply understand git integration and workflow (more on that later). But before I could fully appreciate this new development, I had to embark on a series of false starts and wound up in some dead ends while trying to optimise my workflow.

Trial and error

In the past, to achieve dotfile version control I had tried several methods. The first time, I used an install script. Doing it that way was somewhat awkward because it created symlinks and I needed to follow up and edit file permissions to make them executable. It also lacked idempotence, because if I ran the script twice by mistake it was possible to erase the original dotfiles before the script ran again.

That setup didn't last long. I remembered GNU Stow and decided that it might be a solution, since Stow is cross-platform and has very simple git integration. However, after using Stow to manage my dotfiles for a few years, I experienced the following shortcomings and limitations:

- Stow doesn't integrate fully with version control systems like git. And again, just like the previous method I used, it relies on symlinks to manage the dotfiles. Any versioning or tracking has to be done manually using git, so I had to manage my repository separately.
- Stow doesn't offer templating or encryption features. If you need to manage sensitive information (e.g., API keys) or customize dotfiles for different machines, you need to handle these separately.
- Stow is primarily a symlink manager and works well on UNIX-like systems but requires extra effort to set up and maintain on different platforms, like macOS or Windows.
- Because of its nature (symbolic link management), Stow sometimes struggles with more complex directory structures or special files. It doesn't manage secrets such as files that shouldn't be symlinked, dotfiles that require specific ownership, or edge cases where symlinking isn't ideal.
- While Stow is simple and effective at what it was designed to do, ultimately it's a lower-level tool and requires manual intervention. Using it means you're going to need to handle your git repository and manage per-machine config outside of Stow itself.
- To use Stow, you need to manually organize your dotfiles into appropriate directories and ensure that the symbolic links are correctly created. This setup can be tedious, especially for large configurations.

chezmoi to the rescue!

I needed a better solution with less maintenance overhead. So I was intrigued when I stumbled upon the chezmoi project after a perfunctory search.

chezmoi is an application written in Go and focussed exclusively on home directory management. "chezmoi" means "at my house" in French, and is pronounced /ʃeɪ mwa/ (shay-moi). It allows you to manage your configuration files across multiple machines

running diverse operating systems, and solves the problems inherent in solutions that revolve around using bare git repos or symlinking. Some of the features include:

- Templates
- Password manager support (most of the commonly used apps)
- Importing files from archives
- Full file encryption for secrets
- Running scripts

Best of all, updating your dotfiles on a given machine is a single command:

```
> chezmoi update
```

Unlike other dotfile management solutions, chezmoi is a single binary with many install methods and no bootstrap requirements. It supports private files, whole file encryption, password manager integration, and custom variables in templates. With chezmoi you can set up your personal environment on a new machine with just two short commands, with minimal dependencies.

After installing chezmoi, follow the quick start guide to make the initial commit:

```
> chezmoi init
> chezmoi add ~/.zshrc
> chezmoi cd
> git commit -m 'init'
> git remote add # Set remote to your repo
> git branch -M main
> git push origin main
```

Now your ~/.zshrc is under version control. You can go ahead and add your most important dotfiles, like ~/.bashrc, ~/.bash_profile, ~/.profile, ~/.zshenv, ~/.tmux.conf, ~/.vimrc (or in my case, ~/.config/nvim/init.lua), ~/.ssh/config, ~/.gitconfig, ~/.gitignore_global, ~/.pythonrc, ~/.npmrc, ~/.curlrc, ~/.wgetrc, &c.

chezmoi is designed to work closely with git and other version control systems. It integrates version control commands directly into your workflow, simplifying both management and deployment. There’s built-in templating, making it easy to customize

your dotfiles for different machines, environments, or users. It also includes support for encryption, so you can safely manage secrets within your dotfiles repository using tools like GPG.

The package is cross-platform and designed to work seamlessly on Linux, macOS, and Windows. This makes it much more flexible if you manage dotfiles across different operating systems, and chezmoi is a bit more sophisticated in the way it manages edge cases. It allows you to configure certain files to be symlinked, copied, or ignored. It also supports conditional logic, so you can define different behaviours depending on the system or environment.

chezmoi is a higher-level tool with more built-in capabilities for managing dotfiles. It has more comprehensive documentation and a larger feature set designed specifically for dotfile management, including encrypted secrets, machine-specific configurations, and better git integration. The setup process is mostly automated. You can initialise your dotfiles repository directly from your home directory and let chezmoi handle the file management, ensuring that files are placed in the correct locations without manual intervention.

Summary

chezmoi provides a more powerful, all-in-one solution for versioning, cross-platform usage, templating, and securely managing sensitive information. This makes it far superior for complex and modern dotfile management, especially in environments where you need version control, customization, and encryption.