Ask HN: What do you use to manage dotfiles?

91 points by polm23 on Feb 10, 2016 | hide | past | favorite | 93 comments

StreakyCobra on Feb 10, 2016 [–]

I use:

```
git init --bare $HOME/.myconf
alias config='/usr/bin/git --git-dir=$HOME/.myconf/ --work-tree=$HOME'
config config status.showUntrackedFiles no
```

where my ~/.myconf directory is a git bare repository. Then any file within the home folder can be versioned with normal commands like:

```
config status
config add .vimrc
config commit -m "Add vimrc"
config add .config/redshift.conf
config commit -m "Add redshift config"
config push
```

And so one…

No extra tooling, no symlinks, files are tracked on a version control system, you can use different branches for different computers, you can replicate you configuration easily on new installation.

> seliopou on Feb 10, 2016 [–]
>
> To complete the description of the workflow (for others), you can replicate your home directory on a new machine using the following command:
>
> ```
> git clone --separate-git-dir=~/.myconf /path/to/repo ~
> ```
>
> This is the best solution I've seen so far, and I may adopt it next time I get the itch to reconfigure my environment.
>
> > telotortium on Feb 11, 2016 [–]
> >
> > For posterity, note that this will fail if your home directory isn't empty. To get around that, clone the repo's working directory into a temporary directory first and then delete that directory,
> >
> > ```
> > git clone --separate-git-dir=$HOME/.myconf /path/to/repo $HOME/myconf-tmp
> > cp ~/myconf-tmp/.gitmodules ~  # If you use Git submodules
> > rm -r ~/myconf-tmp/
> > alias config='/usr/bin/git --git-dir=$HOME/.myconf/ --work-tree=$HOME'
> > ```
> >
> > and then proceed as before.
> >
> > > durdn on Feb 17, 2016 [–]
> > >
> > > I found out exactly the same problem :)

> durdn on Feb 17, 2016 [–]
>
> As promised I wrote a post about your setup. Thanks a lot for bringing the technique to my attention: https://developer.atlassian.com/blog/2016/02/best-way-to-sto...
>
> > StreakyCobra on Feb 22, 2016 [–]
> >
> > Really nice, well explained. I'll know where to point people to when I need to present this techique. Thanks!

> aprdm on Feb 10, 2016 [–]
>
> Can you write a blog post on your workflow? I would love to learn more on how you to use it :P
>
> > StreakyCobra on Feb 10, 2016 [–]
> >
> > There are probably already posts about this: I didn't invented it, I read it somewhere… but it was long time ago, I don't remember where.
> >
> > durdn on Feb 10, 2016 [–]
> >
> > I'm trying to reproduce it and documenting it as I go... ok I got it working now and I have notes. Stay tuned.

StreakyCobra on Feb 10, 2016 [–]

It's probably better if you do it then :-) I would probably skip some important parts trying to explain because I'm too much used to it already.

Siilwyn on Feb 13, 2016 [–]

Finally got mine working, it does need some figuring out especially the replication part. I documented the needed commands here: https://github.com/Siilwyn/my-dotfiles/tree/master/.my-dotfi...

durdn on Feb 17, 2016 [–]

I'll check how you did it. I wrote my notes and a couple of scripts down in a post: https://developer.atlassian.com/blog/2016/02/best-way-to-sto...

durdn on Feb 10, 2016 [–]

Working on it! I'll give you full credit for it and link to this thread for reference. Also let me know if you have a Twitter account you'd like me to reference.

StreakyCobra on Feb 10, 2016 [–]

Nice thanks. I don't have twitter. Also don't present me as the inventor of this technique because I'm not. I've read this long time ago on some dark corners of the internet :-) I'm just pointing it out.

durdn on Feb 10, 2016 [–]

Alright I'll be careful in my wording ;).

wooptoo on Feb 10, 2016 [–]

Why is this a bare repo as opposed to a normal one?

StreakyCobra on Feb 10, 2016 [–]

Because the working tree is already your home folder, you don't need to also have a copy of these files in ".myconf/".

idle_zealot on Feb 11, 2016 [–]

So why use .myconf/ at all? What is it doing?

telotortium on Feb 11, 2016 [–]

It contains the files that would normally be in .git (run `git help gitrepository-layout` for more details on the contents).

durdn on Feb 10, 2016 [–]

Honestly this is genius! I hadn't thought of doing it like that, thank you! I had resorted to the usual .cfg folder and a helper script to link/update everything.

shabda on Feb 10, 2016 [–]

Could you explain how this works? Won't this be putting the .bashrc in $HOME/.myconf/.bashrc, which won't get picked up by bash? (And similar for other dotfiles)

unwind on Feb 10, 2016 [–]

No, since the config alias runs git with the option "--work-tree=$HOME", which tells it that the working directory is your home directory root, i.e. where config files (used to) live.

Anyway it's the proper root for config files, since if you use a .config directory (as seems to be the modern choice) *that* needs to live in your home directory of course.

nindalf on Feb 10, 2016 [–]

You could add a symlink in your $HOME folder

qznc on Feb 10, 2016 [–]

"branches for different computers" sounds tedious if most changes are for every computer.

StreakyCobra on Feb 10, 2016 [–]

I have a "master" branch, and some "computer" branches. When changes are required for all computer, I do it in "master", and then update each branch by doing a "git merge master".

kavehmz on Feb 10, 2016 [–]

You mean, you are also adding pushing your .ssh dir to a remote repo?

durdn on Feb 10, 2016 [–]

Not at all. You can 'git add' (or in the case of the technique above 'config add') only the files and folders that are safely stored in a repository. Because of the 'ShowUntrackedFiles' flag, git/config won't always show folders you don't want to track, which would be annoying.

kavehmz on Feb 10, 2016 [–]

Perfect then,

abricot on Feb 10, 2016 [–]

I'm confused - you can add files that are in a parent directory?

StreakyCobra on Feb 10, 2016 [–]

It's because we specify to git that the working tree is the home folder. For it the versionning doesn't happen in ".myconf", but directly in the home folder

berdario on Feb 10, 2016 [–]

Ansible and my own python script.

I use ansible, to template my gitconfig for different unix machines, and to install software that might be referenced in a dotfile

https://github.com/berdario/dotfiles/blob/master/setup/roles...

https://github.com/berdario/dotfiles/blob/master/setup/feynm...

(I have a separate branch for windows, but I found out that branches are not a good solution for this, since unlike feature branches, they'll never be truly merged... and unlike maintainance branches, they'll never stop being touched due to being out of maintanance)

And I use my own script, to also support Windows (since ansible supports windows targets, but cannot be used from Windows)... I defined this table with the destination for the symlinks (or, in the case of .ghci the destination where to copy it, since symlinking it wouldn't work)

https://github.com/berdario/dotfiles/blob/master/deploy.py#L...

mintplant on Feb 10, 2016 [–]

GNU stow and git, per this tutorial:

http://brandon.invergo.net/news/2012-05-26-using-gnu-stow-to...

FireBeyond on Feb 10, 2016 [–]

I do love stow. My only hurdle (and it's not unique to stow is the chicken-and-egg of a 'setup.sh' that builds out my Mac the way I like (creating directories, installing packages etc), when my SSH keys are in stow's SSH folder.

s_kilk on Feb 10, 2016 [–]

I just rebuild them by hand whenever I need to.

It helps weed out the crap I've accumulated since the last machine rebuild, and makes sure I don't end up with an ever-growing hairball of dotfile madness.

jb1991 on Feb 10, 2016 [–]

Agreed, backup is great, but only if you are backing up the right stuff. Sometimes, *not* backing something up is the right call.

gjulianm on Feb 10, 2016 [–]

I think no one has mentioned rcm[1]. I just maintain a private git repository cloned in .dotfiles in each system I own, and use rcm to set up symbolic links properly. It works pretty well with directories and lets you choose between creating and populating it with symlinks, or just symlinking the whole directory. For example, I can symlink the full .vim directory (including git submodules) and only link some files inside the .ssh directory (link the config file to my .dotfiles repo but leave ssh keys alone).

1: https://robots.thoughtbot.com/rcm-for-rc-files-in-dotfiles-r...

gmmeyer on Feb 11, 2016 [–]

I've tried the other ones, but rcm is by far the most user friendly. I maintain a public and a private repo, and link them both into rcm. It works wonderfully.

profsnuggles on Feb 10, 2016 [–]

Git and Xstow. I have a small shell script that parses the xstow.ini file and creates all the directories I have listed under the [keep-dirs] directive in order to prevent it from deleting empty directories or replacing them with links.

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

#Read the keep directories from xstow.ini
ini="$(<'xstow.ini')"
IFS=$'\n' && ini=( ${ini} )
ini=( ${ini[*]/\    =/=} )  # remove tabs before =
ini=( ${ini[*]/=\   /=} )   # remove tabs after =
ini=( ${ini[*]/\ =\ /=} )   # remove anything with a space around =

#for each keep dir make sure it exists in the home dir
for i in ${ini[@]}
do
      if [[ $i =~ ^\ *dir ]]
      then
              eval $i
              mkdir $dir
      fi
done
```

celadevra_ on Feb 10, 2016 [–]

Emacs Org-mode's org-babel functionality. I have a few org-mode files storing all the dotfiles, put under version control. I can update and deploy them from within Emacs.

> profsnuggles on Feb 10, 2016 [–]
>
> This is the most interesting system I've heard about for dotfiles. How do you organize the org files? Having a single org file with all my dotfiles in it could be interesting but probably unwieldy. It could be interesting to use code block expansion for keeping secrets out of the dotfile source.
>
> I keep my emacs init in org and I can't believe I've never thought of this.
>
> > mih on Feb 10, 2016 [–]
> >
> > Interesting approach! Think you can provide us with a small working example, especially the 'deploy' part?
> >
> > > profsnuggles on Feb 10, 2016 [–]
> > >
> > > This is what I came up with over lunch. I'm going to try keeping all my dotfiles in one org file. I have an init hook that runs org-bable-tangle to re-export all the dotfiles after saving.
> > >
> > > ```
> > > ** Meta
> > >    If you place the following code into your emacs init when saving the
> > >    ~/.dotfiles.org file the dotfiles will all be exported.
> > >
> > >   #+BEGIN_SRC emacs-lisp :tangle yes
> > >   (defun dotfiles-hook ()
> > >   "If the current buffer is '~/.dotfiles.org' the code-blocks are
> > >   tangled."
> > >   (when (equal (buffer-file-name)
> > >   (expand-file-name (concat (getenv "HOME")
> > >   "/.dotfiles.org")))
> > >   (org-babel-tangle)))
> > >
> > >   (add-hook 'after-save-hook 'dotfiles-hook)
> > >   #+END_SRC
> > >
> > > ** bashrc
> > >   #+BEGIN_SRC conf :tangle ~/.bashrc
> > >   export PATH=$HOME/bin:$PATH
> > >   #+END_SRC
> > >
> > > ** tmux
> > >   #+BEGIN_SRC conf :tangle ~/.tmux.conf
> > >   unbind C-b
> > >   set -g prefix C-t
> > >   bind C-t send-prefix
> > >   #+END_SRC
> > > ```
> > >
> > > > mih on Feb 10, 2016 [–]

Thanks. Upvoted your solution, but I see some problems with this approach for my workflow:

1. I run emacs-server, so the init file is run only during restarts OR when I manually load it from emacsclient.

2. Re-exporting all dotfiles seems rather redundant. I prefer to selectively export individual dotfiles only when I have made changes to it. That's why I was interested in the 'deploy' part.

I am not familiar enough with Babel, but it definitely seems to offer a more centralized way of managing my dotfiles. It's going into TODO.

jb1991 on Feb 10, 2016 [–]

fascinating. never heard of doing something like this, though i use emacs all day long (but not org mode -- didn't even know org mode could do this).

stevekemp on Feb 12, 2016 [–]

You might enjoy this then:

https://github.com/skx/dotfiles/tree/master/.emacs.d

In there you'll find ~/.emacs/init.el which parses ~/.emacs/init.md which is a markdown file containing both documentation and executable Lisp.

There are other files in separate directories/packages, but I like the idea of placing everything in one readable file:

https://github.com/skx/dotfiles/blob/master/.emacs.d/init.md

Kototama on Feb 10, 2016 [–]

Git. Look for "homegit" on this page: http://chneukirchen.org/blog/archive/2013/01/a-grab-bag-of-g...

knlje on Feb 10, 2016 [–]

I have all my dotfiles in Dropbox and create symlinks to those. I require some 'quasi-secrets' in my configs and therefore I do not use Github.

Brajeshwar on Feb 10, 2016 [–]

Heard good things about stow[1]. Haven't moved to it personally. I'm still stuck with manual symlinks.

1. https://www.gnu.org/software/stow/

tony on Feb 10, 2016 [–]

I've probably spent hundreds of hours across all my configs over the years. In the old days, I'd rsync config files manually. Having frustrating times where I had to start everything over again.

I have a lot to say on the subject.

1. Like other users here, git is a great way version your files. Not just that, it handles the issue you have with keeping the configs of various systems in sync.

1b. It doesn't have to be GitHub, but understand pushing to *some* remote gives you a backup, and a way to keep the latest configs you have in sync across multiple machines.

2. As a rule of thumb, the more POSIX compliant you are, the more cross-compatible your dot-configs will be. In my case, a great deal of my config works superbly across Ubuntu, FreeBSD and OS X with no modification whatsoever.

3. dotfiles (https://pypi.python.org/pypi/dotfiles) is very helpful for building those initial symlinks.

4. Tangentially related is PATH's. Definitely be sure you're not accidentally appending multiple one's over again or omitting ones you want to search. For this, I recommend a pathappend function like one used at http://superuser.com/a/753948.

5. As for managing vim / neovim, I'm coming to the realization the amount of time I've spent trying to configure completion / fix tiny things over the years probably makes me lose the net benefit vim has given me. Too bad there is no intellij for the CLI. In any event, I keep a vim config at https://github.com/tony/vim-config which I document extensively. It has quite a lot of bells and whistles, but lazy loads and checks the system before installing certain plugins. It should work with neovim too.

I keep my central dot-configs (along with its submodules) at https://github.com/tony/.dot-config. Its permissively licensed, so feel free to copy whatever you'd like.

vok5 on Feb 10, 2016 [–]

I have everything in ~/dotfiles which is a git repository here: https://github.com/dmarcoux/dotfiles. Basically, I use GNU Stow to symlink what I need. I have master which contains common config and one branch per computer. Everything is explained in the README, in more details if you want do know more. It's not perfect yet, I still have some small irritants, but I'm quite happy with this setup.

kiesel on Feb 11, 2016 [–]

I use homeshick (https://github.com/andsens/homeshick), a - as I understand - rewrite of homesick (ruby) in bash.

It needs to be sourced in .{bash,z}shrc and has features like tracking files from multiple repos (so called "castles"), auto-linking, auto-update every X days.

We also use it in our dev team to share some config (and ~/bin) files, works fine.

RazorX on Feb 11, 2016 [–]

I have a modular family of git repos with a 'dotfiles' one that pulls in the dependencies via bower and handles various bootstrapping and install steps. I also have one for my system config which works similarly. Vim, zsh, and tmux have their own standalone repos with one step install and updating scripts. They are written as normal plugins that load other plugins.

I want to avoid vendor lock-in to something like chef for this. The idea is that everything is either defined in a tool-agnostic config file, and bootstrapping / installing / updating the dependencies is handled by simple shell scripts. Down the road I can always swap out the tooling (bower, config_curator, archutil) without updating code in my repos since state is defined as data.

I don't like syslinks or putting ~/ under git as I don't want my working tree to affect my dotfiles until I run an "install" command.

https://github.com/rxrc

https://github.com/rxrc/dotfiles

jamescun on Feb 10, 2016 [–]

I see a trend of people maintaining a GitHub repo called `dotfiles` for their public configurations, myself included for zsh/tmux/vim/git. I haven't found a satisfactory way to sync secrets between machines other than via sneakernet.

https://github.com/jamescun/dotfiles

> rjbwork on Feb 10, 2016 [–]
>
> Have you tried syncthing perchance?

> jimmcslim on Feb 10, 2016 [–]
>
> ansible-vault?

rcconf on Feb 10, 2016 [–]

My entire machine is setup using Ansible, Homebrew and Homebrew Cask. It works pretty well.

https://github.com/arianitu/setup-my-environment

https://github.com/arianitu/dotfiles

simi_ on Feb 10, 2016 [–]

I keep all my dotfiles in a repo at ~/cfg, and have a script to perform tasks such as creating symlinks (e.g. ~/.emacs.el -> ~/cfg/emacs.el) and installing brew, antigen, etc.

https://github.com/andreis/cfg

> qznc on Feb 10, 2016 [–]
>
> Same here
>
> https://github.com/qznc/dot

chrisseaton on Feb 10, 2016 [–]

I try not to deviate from configuration default so that I don't need to manage any dot files.

riquito on Feb 10, 2016 [–]

I use a git repository, cloned in .myconfigs, with a script that create a symlink for any file in it (apart from .git and a couple more)

https://github.com/riquito/configs

The usage is

```
git clone git@github.com:username/configs.git ~/.myconfigs
cd ~/.myconfigs
./reinstall.sh
```

Whenever I update the repository, maybe adding files, I run this in the other computers:

```
cd ~/.myconfigs
git pull --ff-only
./reinstall.sh
```

which simply refresh the symlinks (I should remove stale symlinks now that I think about it, for removed configurations - never happened yet)

arc0re on Feb 10, 2016 [–]

Just git and Github.I have two folders, a .dotfiles (https://github.com/Arc0re/dotfiles) that contains stuff like .emacs (for each OS), .bashrc/.zshrc, etc, which I symlink into my $HOME folder, and an elisp (https://github.com/Arc0re/mac-elisp) folder that contains my Emacs themes and plugins.

charlieegan3 on Feb 10, 2016 [–]

I have a GitHub repo with a script that sets everything up. https://github.com/charlieegan3/dotfiles

Seems to be quite a common pattern: https://github.com/search?o=desc&q=dotfiles&s=stars&type=Rep...

trengrj on Feb 10, 2016 [–]

I have a little Makefile that symlinks everything (WIP) https://github.com/trengrj/dotfiles/blob/master/Makefile.

I was previously using an automated Ruby script but found it inflexible and so have switched to a hand coded Makefile.

ssh0 on Feb 12, 2016 [–]

I use bash based dotfiles manager "dot".

* https://github.com/ssh0/dot

And I have dotfiles repository at GitHub. Some files are hosted in Dropbox (for API keys and etc.).

my dotfiles:

* https://github.com/ssh0/dotfiles

srijanshetty on Feb 10, 2016 [–]

I use vcsh and my, I even wrote a blog post about it: https://srijanshetty.in/technical/vcsh-mr-dotfiles-nirvana/

> tfn on Feb 11, 2016 [–]
>
> Me too! Here's my blog-post about it: http://blog.tfnico.com/2014/03/managing-dot-files-with-vcsh-...
>
> I also interviewed the developer some years ago here: http://episodes.gitminutes.com/2013/06/gitminutes-13-richard...

> ge0 on Feb 10, 2016 [–]
>
> Vcsh is great, surprised more people haven't heard of it or used it

> joshuata on Feb 11, 2016 [–]
>
> Thank you so much for that post! I used it extensively when I set up vcsh and my. It took a few hours to get everything working, but the flexibility is unbelievable.

pedrospdc on Feb 10, 2016 [–]

I use Homesick (https://github.com/technicalpickles/homesick).

It's basically a dotfile manager. Symbolically links your stuff and runs scripts.

> kiesel on Feb 11, 2016 [–]
>
> There's a fork called homeshick which is a plain-bash rewrite of homesick, so it doesn't have the ruby dependencies (for those who care about that). There might be few differences, though.

georgewsinger on Feb 10, 2016 [–]

https://github.com/georgewsinger/drop-dot/blob/master/README...

girvo on Feb 10, 2016 [–]

I don't use anything, to be honest. My good friend wrote and uses this however: https://github.com/ncraike/dither

sgtpep on Feb 10, 2016 [–]

Just git.

```
cd ~
git init
git remote add origin https://sgtpep@github.com/sgtpep/dotfiles.git
git fetch
git checkout -ft origin/master
git config status.showUntrackedFiles no
```

atmosx on Feb 13, 2016 [–]

I a Syncthing directory called "Data/" where my personal files are. Then I have a script which creates symlinks between ~/.dir and Sync/Data/dir and that's it, simple but works.

YesThatTom2 on Feb 10, 2016 [–]

Obsessive Compulsive Directory http://wiki.eater.org/ocd

It is a very simple system for keeping your dotfiles (and other files) in Git.

gtf21 on Feb 10, 2016 [–]

git + symlinks (https://github.com/gfarrell/dotfiles). I use a makefile to set everything up

thiht on Feb 10, 2016 [–]

```
cd ~
git init
```

And add this to a .gitignore:

```
# Ignore everything
*

# Except the dotfiles I explicitely want to share
!.vimrc
!.tmux.conf
# ...
```

skeoh on Feb 11, 2016 [–]

Fun fact! `cd` is equivalent to `cd ~` -- that is to say that the directory is optional and defaults to $HOME.

yuvadam on Feb 10, 2016 [–]

dotfiles repo on Github + GNU Stow

AndrewVos on Feb 10, 2016 [–]

Git repos + https://github.com/AndrewVos/slink

Fizzadar on Feb 10, 2016 [–]

Git! https://github.com/Fizzadar/dotfiles

jtfairbank on Feb 10, 2016 [–]

https://github.com/majnemer/davesdots

thekaleb on Feb 10, 2016 [–]

My ~/.config directory is a git repository. I use some environment variables for utilities (like vim) that do not support the XDG spec.

ejrgoiejgeoi on Feb 10, 2016 [–]

https://github.com/RichiH/vcsh + myrepos

hiyer on Feb 10, 2016 [–]

https://github.com/lra/mackup + dropbox

peterbond9 on Feb 11, 2016 [–]

lawpoop on Feb 11, 2016 [–]

I put several of my dotfiles as gists on github's gist site, in addition to storing my home directory in a repo.

cJ0th on Feb 12, 2016 [–]

I only semi-regularly tar my dotfiles and dotfolders and then move the archive to an external drive.

bandrami on Feb 10, 2016 [–]

tar, scp, and a VPS server I've had for longer than I can remember (it's still running Lenny, if that helps -- its OpenSSL was *too old* to be vulnerable to Heartbleed...). Every new install, I scp the tarball and extract it to my home directory.

0x142857 on Feb 10, 2016 [–]

nobody mentioned mackup? https://github.com/lra/mackup

oconnor663 on Feb 11, 2016 [–]

git and https://github.com/buildinspace/peru

yoshuaw on Feb 10, 2016 [–]

git + symlinks

https://github.com/yoshuawuyts/dotfiles

ejrgoiejgeoi on Feb 10, 2016 [–]

mr + vcsh