Passwords in the Terminal

What?! Another password manager - Why??

Jack Munday / IT-CD-PI

November 8th, 2024

I break my laptop a lot …

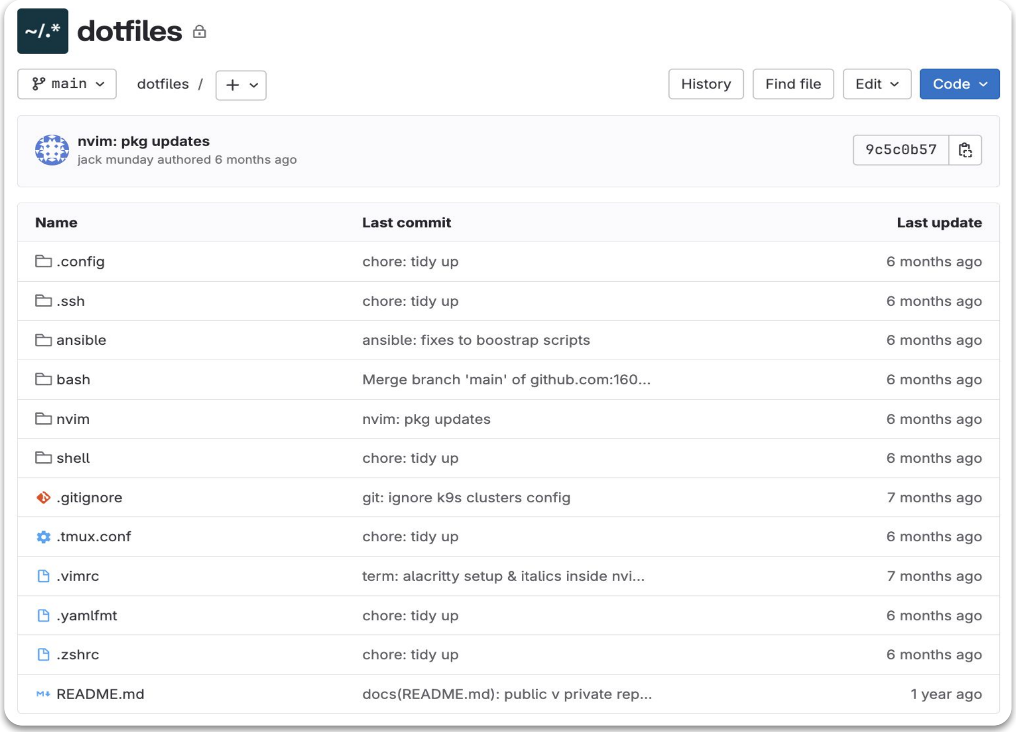
I’ve broken 4 laptops in the last 3 years.

Rice is often no match for a cup of tea under your screen.

Insurance has typically saved me :)

Repeatable paying for a new computer is far less irritating for me than having to actually set one up.

I have found most of the magic tricks for setting up a fresh laptop



dotfiles

nvim: pkg updates
jack munday authored 6 months ago

9c5c0b57

Name	Last commit	Last update
.config	chore: tidy up	6 months ago
.ssh	chore: tidy up	6 months ago
ansible	ansible: fixes to boostrap scripts	6 months ago
bash	Merge branch 'main' of github.com:160...	6 months ago
nvim	nvim: pkg updates	6 months ago
shell	chore: tidy up	6 months ago
.gitignore	git: ignore k9s clusters config	7 months ago
.tmux.conf	chore: tidy up	6 months ago
.vimrc	term: alacritty setup & italics inside nvi...	7 months ago
.yamlfmt	chore: tidy up	6 months ago
.zshrc	chore: tidy up	6 months ago
README.md	docs(README.md): public v private rep...	1 year ago

Every time you run typically find that:

a) **Something is broken**

b) **Something is missing**

If you are really lucky, then both!

Credentials (api & ssh keys) were typically always a pain.

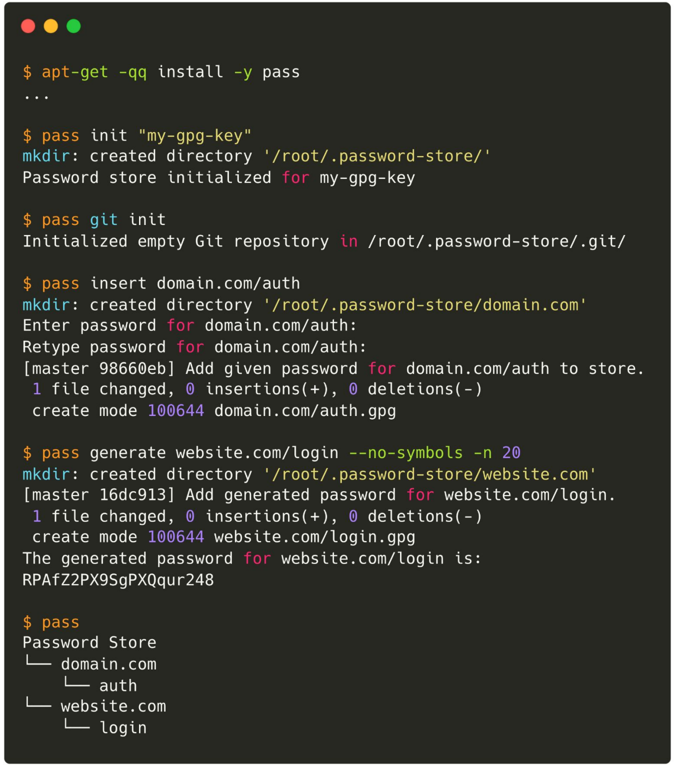
Pass lets you manage credentials as encrypted files

- Each credential is a gpg encrypted file.
- `~/.password-store` is commitable with the rest of `~/.dotfiles`.

  Modifying a password is an atomic action that is tracked in git.
- Multiline data is supported using `--multiline` or `-m`.

  SSH and API key support ✅
- Passwords can be generated using `/dev/urandom`.
- Passwords are easily accessible from scripts.

`pass -c domain.com/auth`



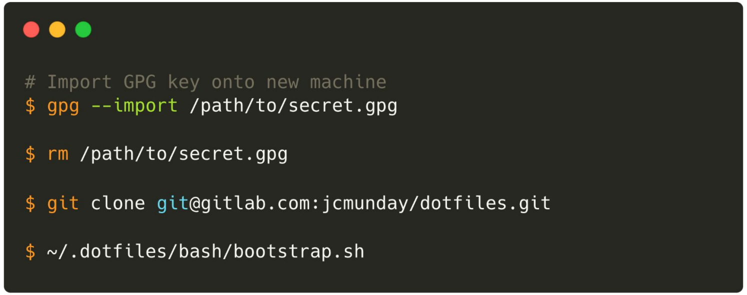
```
$ apt-get -qq install -y pass
...

$ pass init "my-gpg-key"
mkdir: created directory '/root/.password-store/'
Password store initialized for my-gpg-key

$ pass git init
Initialized empty Git repository in /root/.password-store/.git/

$ pass insert domain.com/auth
mkdir: created directory '/root/.password-store/domain.com'
Enter password for domain.com/auth:
Retype password for domain.com/auth:
[master 98660eb] Add given password for domain.com/auth to store.
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 domain.com/auth.gpg

$ pass generate website.com/login --no-symbols -n 20
mkdir: created directory '/root/.password-store/website.com'
[master 16dc913] Add generated password for website.com/login.
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 website.com/login.gpg
The generated password for website.com/login is:
RPAfZ2PX9SgPXQqur248

$ pass
Password Store
└── domain.com
    └── auth
└── website.com
    └── login
```

For the next accident…

Setting up a new machine then requires:

- **Importing GPG key from a third-party backup.** *This can not be git.*
- **Cloning .dotfiles, which now includes both config and credentials files.**
- **Running setup scripts / ansible playbooks with pass ansible plugin**

These decode credentials / ssh keys and place in their proper directories.



```
# Import GPG key onto new machine
$ gpg --import /path/to/secret.gpg

$ rm /path/to/secret.gpg

$ git clone git@gitlab.com:jcmunday/dotfiles.git

$ ~/.dotfiles/bash/bootstrap.sh
```