Password-store keeps your passwords (or any other sensitive information) saved in GnuPG encrypted files organized in `~/.password-store`. For more information about GPG, consult the GNU Privacy Handbook.

Getting started

To get started, install `pass` and generate a keypair.

```
$ brew install pass
$ gpg --gen-key
$ gpg --list-keys
```

Back up the keypair and store it in a safe place.

```
$ gpg --export-secret-keys --armor <fingerprint> > privkey.asc
$ gpg --export --armor <fingerprint> > pubkey.asc
```

Start using pass

```
$ pass init <fingerprint>
```

Each entry is its own file, so you can store whatever text information you'd like, eg. usernames, email addresses, answers to secret questions, two factor auth backup codes, etc. Read the man page for a complete description of its features.

A particularly nice feature is the ability to keep your password store in a git repository.

Managing your password-store with git

Initialize a new bare repository on your server.

```
server $ git init --bare ~/.password-store
```

Make your local password store a git respository and add a remote URL that points to your server.

```
$ pass git init
$ pass git remote add origin user@server:~/.password-store
$ pass git push
```

Using our password store on a new host is easy now.

Import your keypair.

```
$ gpg --import pubkey.asc
$ gpg --allow-secret-key-import --import privkey.asc
```

Trust them if necessary.

```
$ gpg --edit-key <fingerprint>
```

Clone your repository to `~/.password-store`.

```
$ git clone user@server:~/.password-store
```

At this point you can use `pass` on each host and manually synch them with `pass git push` and `pass git pull`. To delete your password store, just `rm -rf ~/.password-store`.