

# GOPASS CHEAT SHEET

Secure passwords with [gopass](#). It creates a folder tree, where encrypted files are the leaves.

```
gopass
  my-company
    pepe@my-company.com
  personal
    pepe@personal.com
```

## GPG Keys

### List secret keys

```
gpg -k
```

### Create new key (required)

```
gpg --full-generate-key
```

## Initialize gopass

### Autocomplete

```
echo "source <(gopass completion bash)" >> ~/.bashrc
```

### Initialize new password store (required)

```
gopass init
```

Note: backup your private key in an encrypted disk.

## Using gopass

### List passwords

```
gopass ls
```

## Creating passwords

Default store location [~/.password-store/](#)

```
gopass insert my-company/willy@email.com
```

## Generate random pass

```
gopass generate my-company/anothername@rmail.com
```

## Search secrets

```
gopass search @email.com
```

## Show password in console

```
gopass my-company/willy@email.com
```

## Copy password to clipboard

```
gopass -c my-company/willy@email.com
```

## Using stores

Stores (AKA mounts) let you group your passwords. Example: [personal](#), [company](#). Each one can live in a different repository, and you could share [company](#) with your peers.

## Initialize new store

Creates a new store located at [~/.password-store-my-company](#).

```
gopass init --store my-company
```

## Add git remote to store

```
gopass git remote add --store my-company origin git@gh
```

## Clone existing store

```
gopass clone git@gh.com/Woile/keys.git my-company --sy
```

## Synchronization

### Synchronize with remotes

```
gopass sync
```

### Synchronizing a single store

```
gopass sync --store my-company
```

## Team sharing

### Export public key

```
gpg -a --export willy@email.com > willy.pub.asc
```

### Check current recipients

```
gopass recipients
```

### Add public key into gopass

```
gpg --import < willy.pub.asc
```

```
gpg --list-keys
```

```
gopass recipients add willy@email.com
```

[source](#) | [presentation](#) | [santiwilly](#)