

Workstation Backup via Git

He does his configuration Management since years with git and without Symlinks. His Solution is more than simple:

create an empty bare git repo

```
git init --bare $HOME/.cfg
```

with this Step you have a repo inside your home directory `$HOME/.cfg`. Inside there are laying your git management files.

set username and email

```
git config --global user.name "Your Name"  
git config --global user.email "youremail@yourdomain.com"
```

create an alias

```
alias config="git --git-dir=$HOME/.cfg/ --work-tree=$HOME"
```

execute this **ONCE**

```
config config --local status.showUntrackedFiles no
```

if you execute it more than once, all your Home Files would be marked always as untracked. If you execute `config status`.

Now you can use the command `config` as you would do with `git`.

`config status` / `commit` / `pull` / `push` ... `merge`, `rebase`, `reset`...

If you have now a file or a folder which you want to add to your dotfiles, then simply add it:

```
config add .vim
config commit -m "My new VIM config"
config push
```

you can of course set an upstream URL to Github / Gitlab / Gitea etc. and push it

```
git clone --bare https://...../dotfiles.git $HOME/.cfg
```

codevoid has created the following ksh aliases (should also work for bash)

```
# config command
alias config='git --git-dir=$HOME/.cfg/ --work-tree=$HOME'

# I'm lazy, so just commit with some machine info
function dotfiles_autoupdate {
    config add -u && \
    config commit -m "Update $(date +"%Y-%m-%d %H:%M") $(uname -s)/$(uname -m)" && \
    config push
}
```

```
# please give me my dotfiles...
function dotfiles_init {
    git --no-replace-objects clone --bare \
        git@codevoid.de:dotfiles.git $HOME/.cfg
    config config --local status.showUntrackedFiles no
    config checkout -f
}
```

you can find this and more in his dotfiles: <https://codevoid.de/?q=/1/git/dotfiles/files.gph> Attention:

Do not commit passwords!!! This sounds logical but is sometimes not so simple. Many programs / config folder can possibly contain cached passwords (vim) and you should also consider to create sample files like .configfile.sample containing only "*****" instead the real password. But that's only one example. learn to commit with: --rebase, stash sometimes you change

something on a not up-to-date branch. Because of that he advises in general

```
config pull --rebase # instad with config pull
```

if there are some uncommitted changes in your git repo, you could do the following:

```
config stash  
config pull --rebase  
config stash apply
```

Guide by codeviod on [uugrn\(at\)mailman.uugrn.org](mailto:uugrn(at)mailman.uugrn.org), translated into english by me (tinfoil-hat)

Revision #1

Created 18 November 2021 21:13:55 by tinfoil-hat

Updated 18 November 2021 21:19:18 by tinfoil-hat