

Git-ify Your (digital) Life

Git-based tools to ease your life

January 17, 2014 | Torbjörn Klatt <t.klatt@fz-juelich.de> | JSC Internal Seminar

Overview

- Git *a short review*
- etckeeper *keep your system's configs*
- vcsh *version your \$HOME*
- mr *my / multiple repositories*
- git-annex *so meta!*
- bup *backup with Git*
- gcrypt *GPG-encrypted Git repositories*
- gitodo *cmd-based ToDo List Manager in a Git repo*
- Tipps *I can't resist ...*

Git-ify Your (digital) Life

Part I: Git

January 17, 2014 | Torbjörn Klatt <t.klatt@fz-juelich.de>

Version Control System *Git*

A short overview

- decentralized / distributed
alike Mercurial/hg or Bazar in contrast to CVS or Subversion
- works on deltas (*diffs*, *patches*) instead of whole files
- non-linear history
branching and *merging* is easy and performant
- cryptological verification of revisions
each revision (*commit*) has a unique SHA-1 hash computed from diff + meta info
- no need for a server / everything is locally available
because of first point

Git-ify Your (digital) Life

Part II: etckeeper

January 17, 2014 | Torbjörn Klatt <t.klatt@fz-juelich.de>

etckeeper – Keep Your System's Configurations

1

- creates a Git (or Mercurial/Bazaar/Darcs) repo for `/etc`
- uses additional meta-file for remembering permissions for each file
DVCS usually don't track file owner info; only executable bit
- uses *pre-commit* hooks to fix file permissions
- hooks itself into package managers (e.g. *apt*, *zypper*) to auto-commit `/etc` before and after package changes
- manual commits also possible

¹ Windows users: sleep or think of moving to Linux

etckeeper – Keep Your System's Configurations

Example 1

Initialization and switching setup

```
etckeeper init
# after some time
cd /etc && git log --oneline
> 5bb2977 daily autocommit
> cdd9c8c yast update
> 9b76558 I added some cron jobs
> 711446f initial commit

# on April first
git checkout april_first_joke_etc
etckeeper init
# day later
git checkout master
etckeeper init
```

etckeeper– Keep Your System's Configurations

Example 2

Get difference between two system's configs

```
git remote add my-other-host ssh://my-other-host/etc
git fetch my-other-host
git diff my-other-host/master group | head
> diff --git a/group b/group
> index 0242b84..b5e4384 100644
> --- a/group
> +++ b/group
> @@ -5,21 +5,21 @@ sys:x:3:
> adm:x:4:joey
> tty:x:5:
> disk:x:6:
> -lp:x:7:cupsys
> +lp:x:7:
```


Git-ify Your (digital) Life

Part III: vcsh

January 17, 2014 | Torbjörn Klatt <t.klatt@fz-juelich.de>

vcsh – Version Control System for (your) \$HOME

version .profile, .{bash,zsh,vim}rc, ... — without pollution

- separate Git repositories for dotfiles
without polluting \$HOME with .git directories
- easily migrate your personalized environment to other hosts
clone your .vim repository on new host to have it synchronized
- allows for different branches for different hosts
e.g. “tklatt-zamws”, “myself-laptop”, “su-myservers”
- *vcsh* is a single Shell script

vcsh – Version Control System for (your) \$HOME

Example

One repository for your Vim config, another for Zsh

```
vcsh init vim
vcsh vim add ~/.vimrc ~/.vim
vcsh vim commit -m "Initial commit of my Vim configuration"
vcsh vim remote add origin git@my-server.net:vim-repo
vcsh vim push -u origin master

vcsh init zsh
vcsh zsh add ~/.zsh ~/.zshrc ~/.zshenv
vcsh zsh commit -m "Initial commit of my Zsh configuration"
vcsh zsh remote add origin git@my-server.net:zsh-repo
vcsh zsh push -u origin master
```

Git-ify Your (digital) Life

Part IV: mr

January 17, 2014 | Torbjörn Klatt <t.klatt@fz-juelich.de>

mr – my / multiple repositories

One command to rule them all

- Problem: a bunch of *vcsh* repos are not very handy
- iterates over list of repos and runs same command on each
- can handle Git, git-svn and *vcsh* repos equally
- provides *bootstrap* command to setup/clone an environment on new host
- integrates well with *vcsh* (*mr* config directory can be a *vcsh* repo itself)
- a single Perl script

Example

```
vcsh list
> vim zsh git ssh bin
mr update      # runs 'git pull' or 'git clone' for each
# downloads named .mrconfig and clones all repos in there
mr bootstrap https://my-server.net/.mrconfig
```

Git-ify Your (digital) Life

Part V: git-annex

January 17, 2014 | Torbjörn Klatt <t.klatt@fz-juelich.de>

git-annex – Version Files Without Their Contents

So meta! ²

- saves meta info (i.e. name, size) of files without their contents
- saves actual files read-only in `.git/annex/objects`
symlinks them to original/real location
- keeps track of which remote has which files
each remote identified by UUID
- designed for flaky connections
uses rsync for data transfer

²Windows users: Wake up!

git-annex – Version Files Without Their Contents

I mean, so really meta!

- written in Haskell
- allows for special remotes
 - Amazon S3 / Glacier
 - WebDAV
 - rsync
 - the web (`http(s)://`, `ftp://`, `archive.org`, `arxiv.org/[format]/[ID]`, etc.)
 - podcast feeds
 - XMPP
 - simple directories
- example collection of some conference proceedings (slides + video recordings)
`https://github.com/RichiH/conference_proceedings`

git-annex – Version Files Without Their Contents

Example Szenario: The Archivist

- annex all files
- actual files offline in special remotes on USB drives, tapes, etc.
- having full info about name, size and location of all files in one place at hand

Example

```
git annex whereis
> whereis my_cool_big_file (1 copy)
>   7570b02e-15e9-11e0-adf0-9f3f94cb2eaa  -- backup drive
> whereis other_file (3 copies)
>   0c443de8-e644-11df-acbf-f7cd7ca6210d  -- here (laptop)
> 62b39bbe-4149-11e0-af01-bb89245a1e61  -- usb drive
> 7570b02e-15e9-11e0-adf0-9f3f94cb2eaa  -- backup drive
```

git-annex – Version Files Without Their Contents

Example Szenario: The Nomad

- keep copies of data online (on internet)
- sync several local devices for occasional backup
- add data locally while on the road
- sync data to online remotes while at Internet café or friend's place
- drop local copies, still have them online and knowing exactly where
- perfect for photos while traveling

Git-ify Your (digital) Life

Part VI: bup

January 17, 2014 | Torbjörn Klatt <t.klatt@fz-juelich.de>

bup – Git for LARGE Files

- recap: Git is designed for plaintext files
binary files are just a huge blob for Git; no diff possible
- uses Git object trees and replaces hashing and packing algorithms
- designed for space-saving incremental backups
- backups can be FUSE mounted
- can be a special remote for *git-annex*
- `bup web`: browse backup trees in web browser
- written in Python

bup – Git for LARGE Files

Example

```
bup init
> Initialized empty Git repository in /root/.bup/
bup index /etc
bup save --name zamws-etc /etc
> Reading index: 6340, done.
> Saving: 100.00% (31381/31381k, 6340/6340 files), done.
bup index /home                                     # took a few seconds
bup save --name zamws-home /home                     # took about 3min
> Reading index: 203502, done.
> Saving: 100.00% (14743111/14743111k, 203502/203502 files), done.
du -sh /etc /home $BUP_DIR
> 49M      /etc
> 15G      /home
> 9.2G     /root/.bup
```

Git-ify Your (digital) Life

Part VII: gcrypt

January 17, 2014 | Torbjörn Klatt <t.klatt@fz-juelich.de>

gcrypt – GPG-encrypted Git remotes

- implements a git-remote-handler to deal with `gcrypt::remotes` transport via `rsync`, `sftp` or `git`
- remote repository is GPG-encrypted for one or multiple participants
- each pack is encrypted with a symmetric key stored in a asymmetric encrypted manifest file
- can be a special remote for *git-annex*
- Hint: use it as a remote for your *etckeeper*'s repo
- Remark: You might want to use Joey “joeyh” Hess' fork of `gcrypt` ³

³ <https://github.com/joeyh/git-remote-gcrypt>

gcrypt – GPG-encrypted Git remotes

Example

```
git init
git add my_secret_file
git commit -m "secret file"
git remote add secret-server gcrypt::git@my-server.net:secret-repo
git push secret-server master
git clone git@my-server.net:secret-repo
ls -lA secret-repo
> -rw----- 1 t.klatt users 303 Jan 15 09:24 0153f2b0...ea5f861d
> -rw----- 1 t.klatt users 1.4K Jan 15 09:24 91bd0c09...4881aa0a
> drwx----- 1 t.klatt users 138 Jan 15 09:25 .git
gpg -d 91bd0c09...4881aa0a
> fc564bef...94c3ff80 refs/heads/master
> pack :SHA256:0153f2b0...ea5f861d w+bxes2v...1MCkGi8+
> repo :id:3lmzxTGoXJVMHPtfa0Tf
```


Git-ify Your (digital) Life

Part VIII: gitodo

January 17, 2014 | Torbjörn Klatt <t.klatt@fz-juelich.de>

gitodo – ToDo List Manager with Git

- simple commandline-based ToDo list management
- Todos are managed in a single Git repository
- items are stored in separate text files (in YAML format)
- supports prioritization, deadlines and *highcal* export
- single portable Ruby script (was a Shell script some time ago)

gitodo – ToDo List Manager with Git

Example

```
gitodo new # --> $EDITOR opens for writing a new ToDo item
```

```
cat $GITODO_DATA/i0010
```

```
> what: Still awake?
```

```
> dead: 2013-12-10 23:59
```

```
> warn: 1
```

```
>
```

```
> Go to sleep! Now!
```

```
gitodo
```

S	Pri	Deadline	ID	Subject
-	-	-	-	-
>	-3		2	Christmas presents
> D	0	22:00	9	Remove garbage from your bed
> C	0	23:30	8	Go to sleep -- yes, it's important
>	0	2013-12-10 23:59	10	Still awake?
>	0	2013-12-24	7	Remove SVN from all computers

Git-ify Your (digital) Life

Part IX: Tipps

January 17, 2014 | Torbjörn Klatt <t.klatt@fz-juelich.de>

Tipps

I can't resist ...

- portable GUI for Git (browsing and actions): *git-cola*⁴
- Zsh
 - very powerful built-in completion for most programs (Git: incl. selecting branches/tags)
 - prompt-integrated info about current VCS working copy
 - highly customizable prompt (left and right)
 - can mimic Bash, Ksh, tcsh (never tried it myself)

⁴ <https://github.com/git-cola/git-cola>

Finally ...

You cannot time travel in real-life.
But you can in your digital life, selectively!

Project Links

`etckeeper` <https://github.com/agimenez/etckeeper>

`vcsh` <https://github.com/RichiH/vcsh>

`mr` <https://github.com/joeyh/myrepos>

`git-annex` <https://git-annex.branchable.com/>

`bup` <https://github.com/bup/bup>

`gcrypt` <https://github.com/blake2-ppc/git-remote-gcrypt>

`gitodo` <https://github.com/vain/gitodo>

Sources

- This talk is heavily inspired by Richard “*RichiH*” Hartman’s talk at *Linuxtag 2013*⁵
- official and unofficial documentation of named tools
- (long-term) experiments with named tools

⁵ <http://www.linuxtag.org/2013/de/program/mittwoch-22-mai-2013.html?eventid=147>

Thank you for your interest!

Questions?

(now or later)

Building Container (16.3v)

Room 2005

Tel 96452

eMail t.klatt@fz-juelich.de

PGP-Key 0x126181D2

Fingerprint 48C5 67E6 26C3 787F 6E8E
3A20 492E 6DF2 1261 81D2