

git – the stupid content tracker

sECuRE beim NoName e.V.

12. Juni 2008

Vorteile gegenüber subversion

- git ist schnell!
- git ist stabil
- git kann besser mergen
- git kann einfacher branchen
- git hat tags (svn nicht)
- git kommt mit manpages (*#\$\$%! svn help)

Nachteile gegenüber subversion

- git in trac ist Masochismus (performance = 0), ansonsten nur gitweb verfügbar
- WebDAV nicht wirklich unterstützt, dafür ssh
- Manche Leute sagen, es sei schwer zu benutzen

Systemweite config

- Praktisch, wenn man immer unter dem selben Namen auftritt
- `git config --global user.name "Heinz Muster"`
- `git config --global user.email "heinz@kernel.org"`
- `git config --global color.diff auto` (wichtig: **auto**, nicht `true`. sonst mag `gitk` nicht mehr)
- Pro Repository: `git config user.name "Mr. X"`

Wie mache ich...?

- `git init` (oder `git-init`)
- Programme schreiben/Content generieren
- `git add .`
- `git diff`
- `git status`
- `git commit -a` (-a für all files)
- `git log`

Und remote?

- ssh host
- mkdir project && cd project
- git init
- exit
- git push host:project/.git refs/heads/master
- (git config remote.origin.url host:project/.git, dann geht's nach dem ersten push später mit "git push")
- andere Menschen: git clone host:project/.git
- zum Updaten: git pull

Wie tagge ich?

- `git-tag 0.1-ALPHA`
- das war ja einfach. und so schnell...
- `git push --tags`
- falls man tags nachholen will: `git-tag 0.1-ALPHA 20edd81`

Wie revertete ich?

- "git reset --hard HEAD^", hard = verwirft Änderungen, soft = behält Änderungen
- "git reset --hard HEAD 2", ebenso HEAD 3, etc...

Wie branche ich?

- `git-checkout -b ssl`
- ganz normal arbeiten...
- Änderungen von \$remote übernehmen: `git pull host:project/.git`
- Bei conflicts die files mit `"git log -p --merge"` (patch für den letzten merge) anzeigen
- Konflikt beheben, danach `"git commit -a"`
- Branch in master integrieren: `"git checkout master"` und `"git merge ssl"`
- Branch veröffentlichen: `"git push host:project/.git refs/heads/test"`

Mehrere Entwickler

- SSH-zugriff via "git-shell" einschränken
- Private/Public-Key-Authentifizierung, noch cooler als SSL+HTTP+WebDAV (außer auf Debian)
- <http://eagain.net/blog/2007/03/22/howto-host-git.html> – mit Script zum Einschränken auf bestimmte Directories
- git-daemon, damit man git://-URLs nutzen kann (als Nutzer, in der Regel nutzt man git-daemon read-only)
- gitweb

Von SVN konvertieren

- Textfile git-svn.users mit " michael = Michael Stapelberg <michael+git@stapelberg.de>"
- `git-svn clone -A git-svn.users --no-metadata file:///tmp/mxallowd`
- `rm -rf mxallowd/.git/svn`
- `git push ...`

Cool stuff

- "git-format-patch [revision]" erzeugt 0001-my-commit-msg.patch mit E-Mail-Headern
- "git-bisect start <bad> <good>", danach "git-bisect good" oder "git-bisect bad" bis man den Bug hat
- Woher der Untertitel kommt: "This is a stupid (but extremely fast) directory content manager. It doesn't do a whole lot, but what it does do is track directory contents efficiently. "