GIT WTF?

Björn Guth

Fachschaft Mathematik/Physik/Informatik der RWTH Aachen

May 22, 2014

Inhalt

Git WTF?

- Tool zur Versionskontrolle
- erleichtert kollaboratives Arbeiten
- einfacher Weg zu externen Backups von Projekten
- weiteres Argument für LaTeX und gegen MS-Office / Open Office / Libreoffice

Vorteile von Git gegenüber SVN

- Git ist schneller als SVN[1]
- Git auch lokal und ohne Verbindung zu einem Server
- Es gibt in Git Branches
- SVN Repositories können in Git enigebunden werden
- man kann mit Git genauso arbeiten, wie mit SVN

[1] zumindest laut `http://git-scm.com/about/small-and-fast`

Vorteile von Git gegenüber Dropbox

- Git ist open source, Dropbox nicht.
- Dorpbox hat so gut wie keine Versionskontrolle
  - Versionskontrolle nur 30 Tage
  - alles darüber hinaus kostet Geld
- Git hat .gitignore
- Dropbox ist ein zentralisiertes System
- Dropbox ist an einen Ordner gebunden
- viel Spaß mit den unterschiedlichen Quota der Dropboxuser
- Google Drive ist ähnlich einzuordnen wie Dropbox

Vorteile von Git gegenüber Dropbox

Bildquelle: Wikipedia (`https://upload.wikimedia.org/wikipedia/commons/thumb/0/04/National_Security_Agency.svg/718px-National_Security_Agency.svg.png`)

Git als Versionskontrolle in Gruppenarbeiten

- Generell gibt es drei Möglichkeiten, Git in deiner Gruppenarbeit zu nutzen:
  1. Plain per ssh von Rechner zu Rechner
  2. mit Gitolite
  3. mit einen externen Hoster

ssh direkt von Rechner zu Rechner

- eine Person erstellt mit ein Repository
- alle anderen clonen dieses
- mit `git remote add NAME ‘‘ADRESSE ZUM REPOSITORY’’` werden alle anderen als Quellen hinzugefügt
- nun können commits von anderen gepullt werden

Vor- und Nachteile

Vorteile:

- dezentrales Netz
- alle Daten nur lokal bei euch

Nachteile:

- auf jeden Rechner muss ein ssh-Daemon laufen
- sinnvoller Weise müsstet ihr euch dafür einen zusätzlichen User anlegen
- wechselnde Netzwerkadressen bereiten Probleme

Gitolite zur Rechteverwaltung

- Gitolite erledigt Rechteverwaltung bei Git-Repositories
- muss zusätzlich zu Git installiert und eingerichtet werden
- weitere Verwaltung sehr einfach über ein Git-Repository
- bei Interesse kann ich mehr zeigen

Vor- und Nachteile

Vorteile:

- alle Daten sind lokal bei euch
- Rechteverwaltung sinnvoll geregelt

Nachteile:

- Einrichtung nicht trivial
- als zentralistisches System ausgelegt[a]

[a] kann aber auch ähnlich wie plain ssh verwendet werden

externe Hoster

- es gibt Anbieter, die Git-Repositories zur verfügung stellen.
- als Beispiel seien hier mal genannt:
  - github.com[2]
  - gitorious.org
  - bitbucket.org[3]
- sehr leicht zu managen
- sind in der Nutzung genauso wie Gitolite

[2] bietet auch einige sehr gute Tutorials zu Git

[3] bietet Git- und Murcurial-Repositories und bei Anmeldung mit einer Univeritäts-Mail-Adresse private Repositorie mit Zugriff durch soviele, wie man will.

Vor- und Nachteile

Vorteile:

- leichte Einrichtung
- vergleichsweise geringe Downtime

Nachteile:

- Daten werden bösen Unternehmen in den Rachen geworfen
- Solange man kein Geld bezahlt, sind die Repositories public

Installation

- Linux:
  - Mit der Paketverwaltung eurer Distribution das Paket `git` installieren.
  - Debian & Ubuntu: `sudo apt-get install git`
  - Ansonsten: `http://git-scm.com/download/linux`
- Windows:
  - `https://msysgit.github.io/`
- Mac OS:
  - `http://git-scm.com/download/mac`

Konfiguration

- Globale Konfiguration in `~/.gitconfig`
- wichtige globale Konfiguratonen:
  - `git config --global user.name NAME`
  - `git config --global user.email EMAIL@ADREESSE`
- im Projekt in `./.git/config`

wichtigste Befehle

- basic:
  - `git init`
  - `git clone ‘‘ADRESSE ZUM REPOSITORY’’`
  - `git add DATEI`
  - `git status`
  - `git commit -m ‘‘ÄNDERUNGSBESCHREIBUNG’’`
  - `git commit --amend`
  - `git push ADRESSE BRANCHNAME`
  - `git pull`
  - `git log`

ein eigener Versuch

- Clont das Repository dieser Präsi
- `git clone kiss12@137.226.113.201:~/git-vortrag`
- schaut mal nach, wie viele commits ich schon gemacht habe

weiter wichtige Befehle

- more advanced:
  - `git branch BRANCH`
  - `git checkout BRANCH`
  - `git checkout BRANCH1`
    `git merge BRANCH2`
  - `git diff`
  - `git branch -d BRANCH`

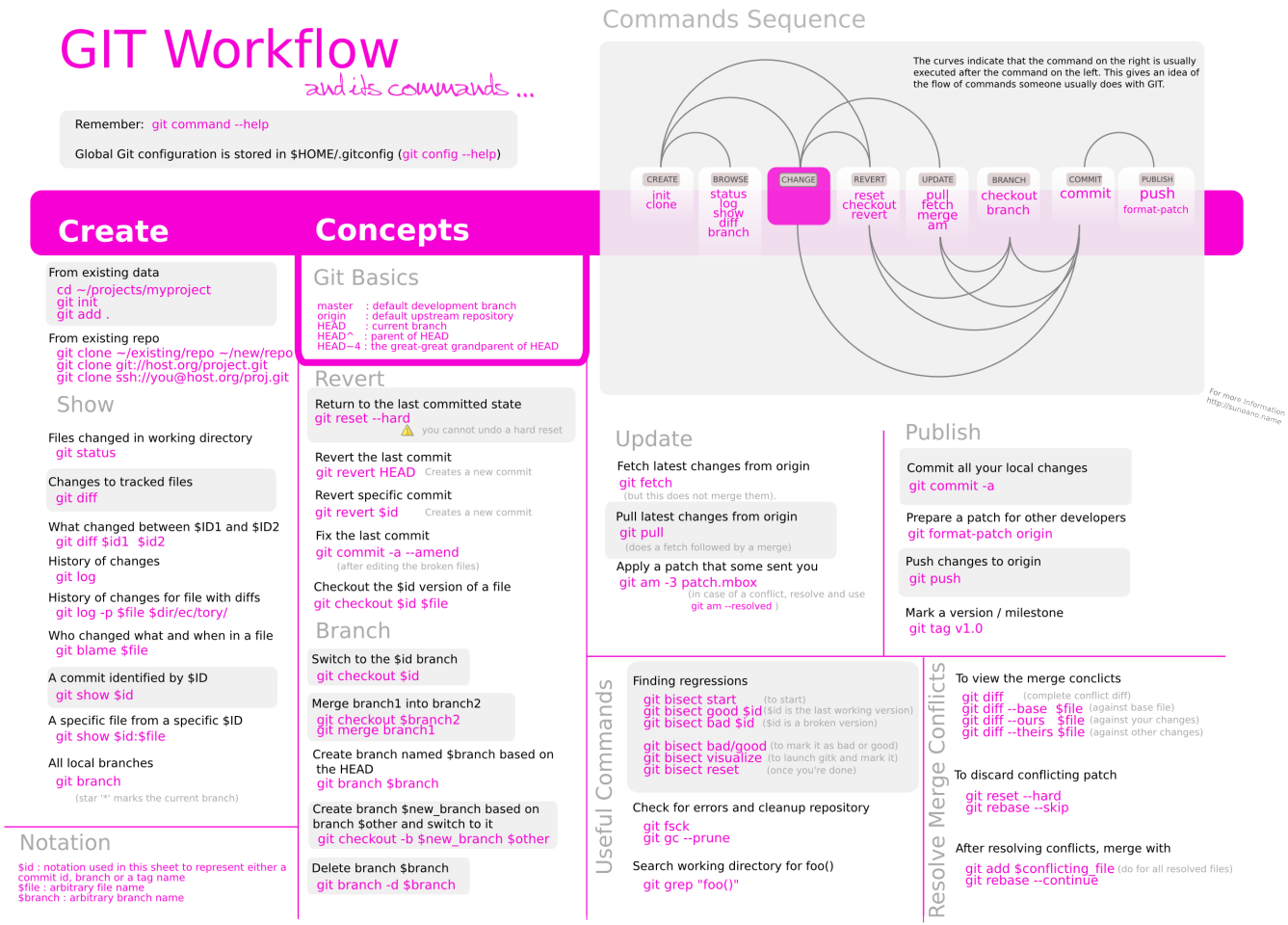
ein weiterer Versuch

- Erstellt einen neuen Branch
- pusht ihn ins Repository
- hat es geklappt?

Weitere Hilfe

- http://git-scm.com/doc
- https://help.github.com/
- http://wiki.ubuntuusers.de/Git
- http://www.markus-gattol.name/misc/mm/si/content/git_workflow_and_cheat_sheet.png

Weitere Hilfe



Bildquelle: Markus Gattol (`http://www.markus-gattol.name/misc/mm/si/content/git_workflow_and_cheat_sheet.png`)

Danke für die Aufmerksamkeit[5]

- Wenn ihr noch Fragen habt, wäre jetzt der richtige Zeitpunkt[4]

[4] Später geht aber auch!

[5] auch wenn es sehr komandozeilenlastig war[6]

[6] Ich hoffe aber ich konte trotzdem ein paar von euch überzeugen Git zu nutzen