



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

Git Basics for TI Staff

Tuesday, 8th September 2015 16:00 - 17:30

BFH - Room HG 4.08, Quellgasse, Biel

Daniel Baumann <daniel.baumann@bfh.ch>

IT System Engineer, Team Infrastructure Services

Overview

Part 1

- ▶ Introduction
- ▶ Git Quickstart

Part 2

- ▶ Git Core Concepts
- ▶ Questions and Answers

Overview

Part 1

- ▶ Introduction
- ▶ Git Quickstart

Part 2

- ▶ Git Core Concepts
- ▶ Questions and Answers

Notes

- ▶ main focus is quick start and some essential concepts, rather than learning to type commands
- ▶ slides contain more information/stuff than is covered now
- ▶ not covering Git installation, configuration, or GUI stuff, see:
 - Linux: <https://linux.bfh.ch/software/git>
 - Windows: <https://linux.bfh.ch/windows>
 - OS X: <https://linux.bfh.ch/osx>

Introduction

Introduction (1/3)

What?

- ▶ Version or Revision Control is any kind of practice that tracks and provides control over changes.

Why Version Control?

- ▶ see who did what, why and when
- ▶ be able to revert stuff
- ▶ extremely important when working alone on non-trivial things over a timespan of more than a couple of days
- ▶ impossible to work without a version control system (VCS) when working in teams

What is Git?

- ▶ Distributed Version Control System
- ▶ Stupid Content Tracker (KISS)
- ▶ defacto industry standard

Introduction (2/3)

What makes Git special?

- ▶ fast
- ▶ does not need a server
- ▶ cloning with full history
- ▶ it is quite hard to loose data
- ▶ relatively straight forward yet powerful
- ▶ multiple ways to get the same thing

What people say about Git who did not get it yet

- ▶ I don't need version control
- ▶ I don't need „distributed“
- ▶ I don't need „complexity“

Introduction (3/3)

Real Disadvantages of Git

- ▶ available physical memory equals max. single filesize in a repository
- ▶ almost too much documentation/bikeshedding available
- ▶ if you do it wrong from the beginning, the learning curve can be a bit steep

Git Repository Hosting at BFH

git.bfh.ch

- ▶ documentation at <https://linux.bfh.ch/services/git/>, configuration available for reuse
- ▶ Debian 8 (jessie) Container with git, gitweb, gitolite3, irker
- ▶ initial key generation by user, import via ticket, after that selfservice for users via ssh

Git Repository Hosting at BFH

git.bfh.ch

- ▶ documentation at <https://linux.bfh.ch/services/git/>, configuration available for reuse
- ▶ Debian 8 (jessie) Container with git, gitweb, gitolite3, irker
- ▶ initial key generation by user, import via ticket, after that selfservice for users via ssh

„Rant“ about GitLab

- ▶ not reasonably packagable (~200 Gems, ~half of them forked)
- ▶ security situation
- ▶ ruby; watching the other alternatives (~~gitotious~~, Gogs, Kallithea, ...)

Git Repository Hosting at BFH

git.bfh.ch

- ▶ documentation at <https://linux.bfh.ch/services/git/>, configuration available for reuse
- ▶ Debian 8 (jessie) Container with git, gitweb, gitolite3, irker
- ▶ initial key generation by user, import via ticket, after that selfservice for users via ssh

„Rant“ about GitLab

- ▶ not reasonably packagable (~200 Gems, ~half of them forked)
- ▶ security situation
- ▶ ruby; watching the other alternatives (git~~ori~~ous, Gogs, Kallithea, ...)

Future for git.bfh.ch

- ▶ short-term: cgit instead of gitweb
- ▶ mid-term: integration with request-tracker and review board
- ▶ key generation via browser, submission via IDM at selfhelp.bfh.ch
- ▶ 2016: two factor auth support with yubi key

Git Quickstart

Git Basic Usage (1/2)

Download a repository from the server to your computer

- ▶ `git clone git@git.example.org:[...]/${repository}.git`

Update an existing repository with changes from the server to your computer

- ▶ `git pull`

Check what has changed in the repository on your computer

- ▶ `git status`

Check how it has changed in the repository on your computer

- ▶ `git diff`

Git Basic Usage (2/2)

Save your changes as a commit

- ▶ `git commit -a -m "My commit message."`

Push your commit(s) from your computer to the server

- ▶ `git push`

Show the list of your commits

- ▶ `git log`

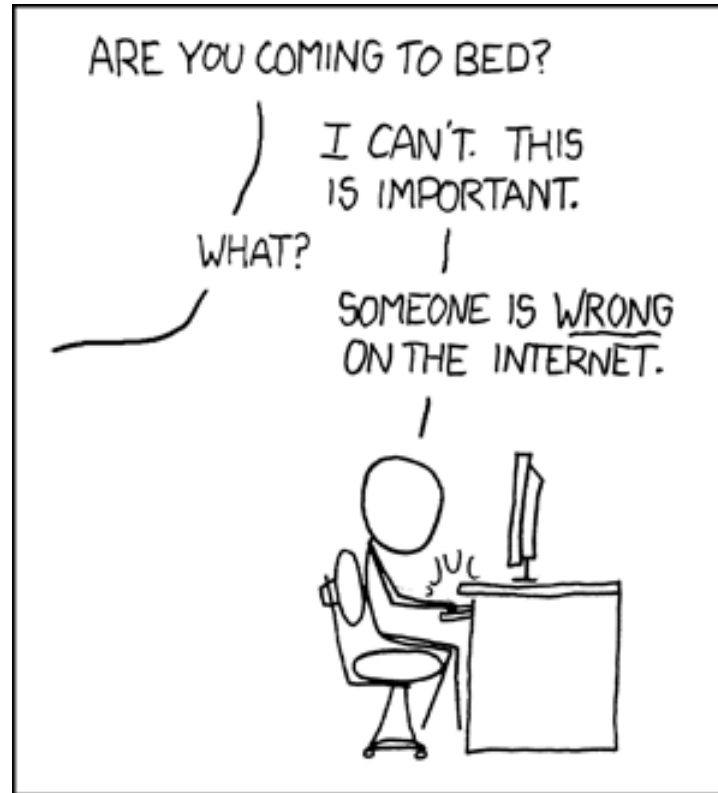
Git Core Concepts

Git Core Concepts

- ▶ **staging area**
- ▶ **commit and hashsums:** you cannot change commits, you can create new ones (and garbage-collect the old ones)
- ▶ **fastforward / non-fastforward**
- ▶ **known solutions for problem A can be used for problem B by first making problem B into A**
- ▶ **branches:** master
- ▶ **references:** explicit and relative; branch, tags and HEAD
- ▶ **remotes and origin**
- ▶ **merge and rebase**

Duty Calls

<https://xkcd.com/386/>



Tooltip: *What do you want me to do? LEAVE? Then they'll keep being wrong!*

Rebase versus Merge

- ▶ don't listen to the Internet, people are wrong
(or have very, very, veeeery specific workflows like kernel.org
and are in now way representative or good role models)
- ▶ if you don't, you're creating a mess for you and everyone else.
- ▶ always do `git pull --rebase` (preferably set it in git config to be the default)
- ▶ always do fast-forward merges (aka rebase), never non-fastforward merges (aka merges)

Questions and Answers

Further Information

Installation, Configuration, Usage, Ressources...

- ▶ <https://linux.bfh.ch/software/git>

BFH Git Services

- ▶ <https://linux.bfh.ch/services/git>

Public Mailinglist for Help

- ▶ English or German: bfh-linux-users@lists.bfh.ch

this was only a quick intro today...

- ▶ ..would you like to have a more focused, indepth, second part?
- ▶ Or a questions & answers session? Or individual help?

Next up...

Git Advanced for TI Staff

▶ **Some more Git commands,
Best Practice, and Distributed Workflows**

▶ **Thursday, 10th September; 09:00-10:30**

Room HG 4.09, Quellgasse, Biel

Thank You for Your Attention.

♥ Source Code is freely available

```
git clone git://git.bfh.ch/git/staff/bad9/other/talks.git
```