

Advanced GIT

Who? Martin Miksanik

When? December 1, 2016

Mozno zaroven testovat na: *[http : //gitlab.mmik.cz](http://gitlab.mmik.cz)*

Uvod

Základní operace

Clone / Fetch

Prace se stromem

Add / RM

Commit / Reset

Fast-forward

Push / Pull

Rebase vs Merge

Stash

Adresovani

Kooperace

Mere/pull/push requesty

Drobneni projektu

CI

Gitlab

Zaver

Úvod

- Proc se to ucit, kdyz to jde naklikat?
- Obří množství funkcí/parametru
- Široké možnosti konfigurace - diff, whitespace atd.
- Jednoduše nastavitelné hooky
- Snadná CI
- <https://git-scm.com/>
- `git help`

Základní operace

- Clone / Fetch
- Add / rm
- Commit / Reset
- Push / Pull
- Checkout
- Branch
- Merge / Rebase
- Revert
- Stash
- ...
- Bare git repozitář
- Repository s index a WORK_TREE

Clone / Fetch

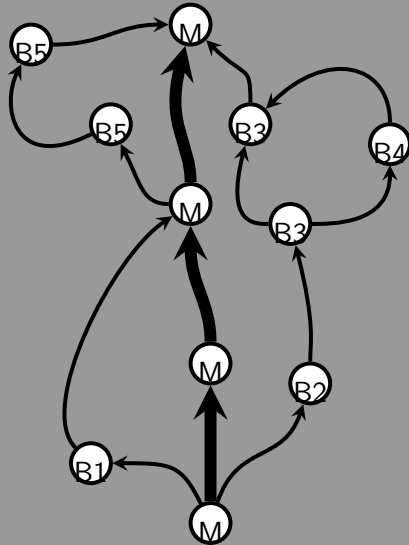
Clone `git clone https://github.com/euroska/slides_git.git`

- Stáhne vzdálené repository (fetch)
- Nastaví remote origin
- Nastavi index (branche, HEAD)

Fetch `git fetch origin -a`

- Stáhne vsechny zmeny ze vzdalenyh adresaru, ale neaplikuje je
- Vzdalene repozitare je mozno vyuzivat (git rebase origin/master)

Strom



Prace se stromem

- Co je radek?
- Co je soubor?
- Co je slozka?
- Co je commit?
- Co je index?
- Co je branch?
- Co je tag?
- Co je reference?

Add / RM

Add

- Podepise zmeny a prida je k commitu (soft)
- Lzou pridať i soubory z ignore s parametrem `–force` (nastavení atd.)

RM

- Odebere soubory z repozitáře
- Lze odebrat mekce s parametrem `–cached`

Commit / Reset

Commit Podepise zmeny, prida zpravu a informace o uzivateli

Reset Odebere soubor/zmenu ze zmen k commitnutí, případně ruší celé commity

- Soft
- Hard
- HEAD~N
- Rychly nástroj pro squash(ovani) commitu

Fast-forward

- Nahraje po sobe jdouci zmeny (push/pull)
- Pouziva se hlavne tam, kde se automaticky provadi zmeny
- Zjednodusuje prehlednost zmen (mene commitu)

Push / Pull

Stáhne/nahraje změny z vzdáleného repozitáře

- Metadata
- Stream objektu
- Lisi se prakticky jen, jake se volaji hooky

Rebase vs Merge

- Lisi se v poradi aplikovani mergu
- Stejne co se tyce vysledneho stavu zdrojovych kodu
- Vyrazne jine pro code review
- Velky rozdil pri automatickych/schvalovanych push/pull/merge requestech

Stash

- Pred pull / push jeste nez se chce udelat opravdovy commit
- Pokud se potrebuje aktualizovat, jsou rozpracovane zmeny ktere se nebudou commitovat (gitlab)

Adresovani

- `refs/heads/master`
- `refs/tags/V0.0.1`
- `refs/remotes/origin/master`

Zakladni domluva - diktat

- Jeden tomu veli, protoze vice lidi se uz nemusi shodnout :)
- Kde se spravujou ukoly, reference z commitu, closes atd.
- Problem s case insenzitive soubory - aneb prokliname windows
- Kdo muze mergovat do masteru - merge/push/pull requesty, kazdy sam, atd.
- Politika vytvoreni/mazani branchu
- Fork a k cemu je dobry
- Co na to code-review

Merge/pull/push requesty

- Jedna se prakticky o tutez vec, jen je ji mozno vyresit N moznostma.
- FastForward vs Merge
- Co nejmensi pocet commitu, idealne squash(nute) do jednoho komentare - generovani changelogu
- Co nejmene primergovani, idealne jen rebasovat master
- Automatika nemuze zafungovat vzdy - rebase, merge, push -force

Drobnosti projektu

- Používat kompozery, instalatory, stahovatory atd. (npm, bower, composer, pip, galaxy atd.)
- Pozor na git submodule
- Jednotlivé moduly jde používat na více místech a netreba kopírovat kód

CI

- GitLab-Ci - test
- Jenkins
- TeamCity
- mnoho dalsich

GitLab

.gitlab-ci.yml

<https://docs.gitlab.com/ce/ci/yaml/README.html>

stages:

- test

job1:

stage: test

script:

- python test.py
- python -m unittest

Testovací projekt na: <http://gitlab.mmik.cz/test/python>

- Variables
- Environments
- Runners

Zaver

Gitlab neni jediny a neni to jedina uzitecna sluzba:

- Gitis
- GitHub
- Jira stash

Stejne ale vetsina veci jde resit i jednoduse pres zakladni git server