git tutorial

王寿林10067372

Outline

- git基础
- gerrit使用

git基础

仓库

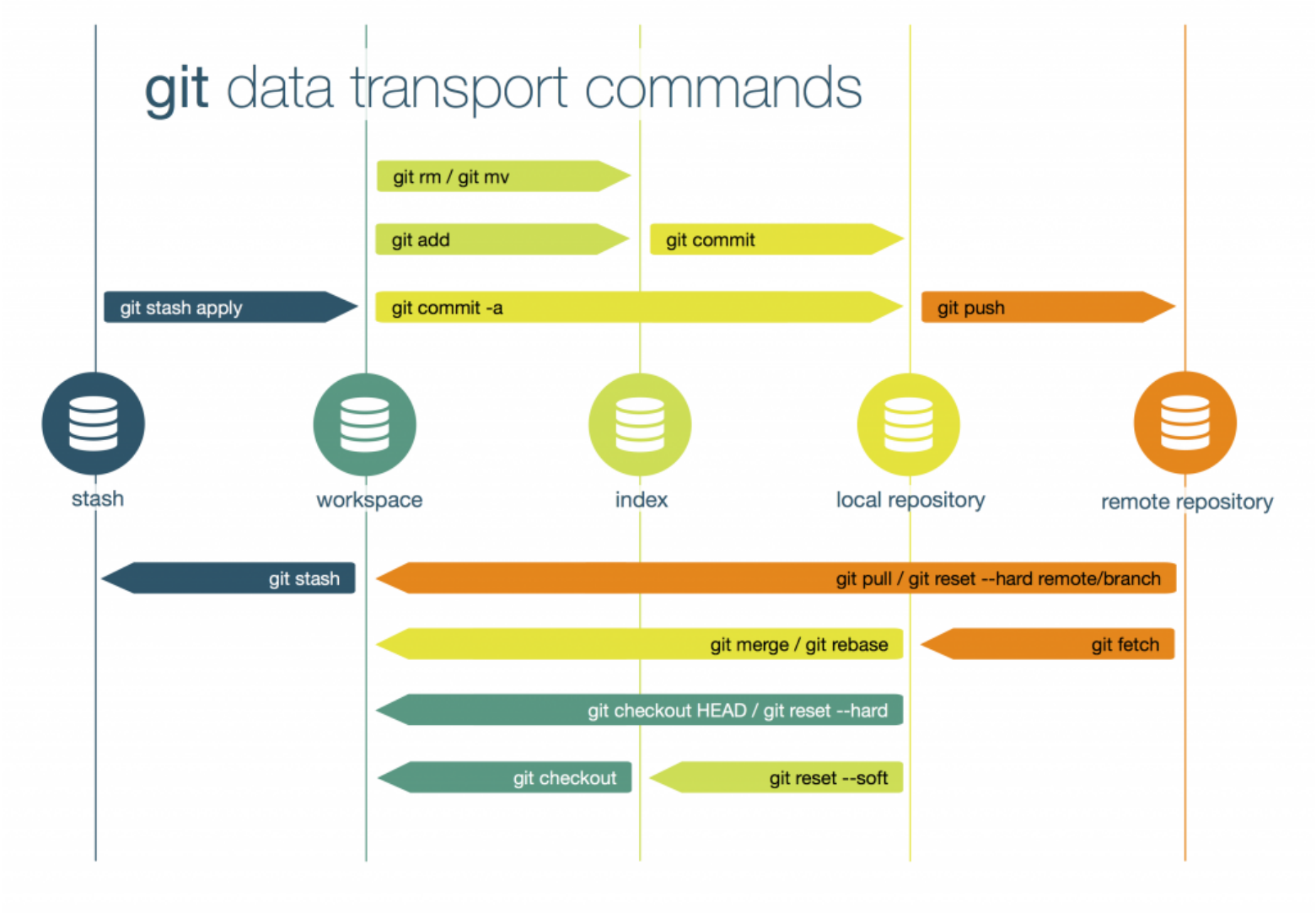
克隆一个远端库

```
# clone repo
git clone git@gitlab.zte.com.cn:10067372/gittutorial.git
```

查询远端库

```
# show remote repo
git remote -v
```

基本操作


git data transport commands
git rm / git mv
git add
git commit
git stash apply
git commit -a
git push
stash
workspace
index
local repository
remote repository
git stash
git pull / git reset --hard remote/branch
git merge / git rebase
git fetch
git checkout HEAD / git reset --hard
git checkout
git reset --soft

一个简单的GIT流程

```
# edit file
vim GIT.md

# check changes
git status

# add tracked file to staging
git add GIT.md

# commit file to local repo
git commit -m 'new file GIT.md'

# show log
git log

# push to remote repo
git push origin master:master
```

查看变更

```
# show local diff
git diff

# show cache diff
git diff --cached

# show commit diff
git diff COMMITID

# show diff between two commits
git diff COMMITID1 COMMITID2
```

撤销工作区变化

```
# edit file
vim GIT.md

# revert local file
git checkout -- GIT.md
```

撤销暂存区变化

```
# edit file
vim GIT.md

# add file to staging
git add -u

# revert add
git reset GIT.md
```

撤销本地仓库提交

```
# edit file
vim GIT.md

# add and commit file
git commit -a -m 'message'

# revert commit and add from all spaces
git reset --hard PREVIOUS_COMMINTID
```

```
# revert commit and add only from local repo and index
```

```
git reset --soft --mixed PREVIOUS_COMMINTID
```

查看提交日志

```
# get all commit logs
git log

# get 4 commit logs
git log -4

# get commit stat
git log --stat

# get commit detail
git log -p

# get pretty logs
git log --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) \
    %C(bold blue)<%an>%Creset' --abbrev-commit --
```

删除文件

```
# delete a file
git rm GIT.md
git commit -m 'message'
```

冲突解决-方法1

```
# pull remote repo and conflict:GIT.md
git pull

# resolve conflict
git checkout --ours GIT.md
cp GIT.md GIT.ours.md
git checkout --theirs GIT.md

# merge GIT.ours.md GIT.md
vimdiff GIT.ours.md GIT.md

# add and commit file
git commit -a -m 'resolve conflict message'

# pull again
git pull

# push chg to remote repo
git push
```

冲突解决-方法2

```
# merge by tool
git mergetool
```

分支--创建、删除与查看

```
# create a new branch called feature branch
git branch feature/bugfix

# switch to feature branch, if it is not existed, create it
git checkout -b feature/bugfix [start_point]

# delete feature branch
git branch -D feature/bugfix

# list all branch
git branch -a
```

分支--切换与合并

```
# switch branch
git checkout feature/bufix

# merge feature branch
git checkout -b feature/bugfix
vim GIT.md

git checkout master
git merge feature/bugfix
```

基于分支的工作流(1)

```
# create new branch to fix bug
git checkout -b feature/bugfix

# push bugfix to remote
git push origin feature/bugfix

# git push origin bugfix
vim files
git commit -a -m 'message'
```

基于分支的工作流(2)

```
# merge bugfix
git checkout master
git merge feature/bugfix

# push to remote
git push

# delete temp branch
git branch -D feature/bugfix

# delete remote branch
git push origin :feature/bugfix
```

merge VS rebase

```
D---E---F---G master
    \
     A---B---C topic
```

merge流程会产生一个额外的提交

```
D---E---F---G--- master
    \            \
     A---B---C---H topic
```

rebase尽量不要在master分支上进行

```
D---E---F---G master
            \
             A'---B'---C' topic
```

fetch

```
git checkout feature

#edit

git commit -am 'edit somethins'

git fetch origin

git rebase origin/master
```

blame

查看文件内容每个部分的修改者

```
git blame
```

reflog

查看HEAD的指针变化

```
git reflog

git reset --hard reflog_id
```

stash And clean

暂存当前的修改

```
git stash

# doing somethings

git pop
```

清除本地未追踪的文件

```
git clean -f -d
```

配置

```
# username and email
git config --global user.name "shoulinwang"
git config --global user.email shoulinwang@gmail.com

# color
git config --global color.ui true
```

```
# alias
```

```
git config --global alias.ci commit
git config --global alias.co checkout
git config --global alias.st status
git config --global alias.br branch
git config --global alias.lg "log --color --graph --pretty=format:'%Cred%h%Creset \
    -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit"
```

忽略

仓库根目录新建`.gitignore`文件，将不需要加入到仓库的目录和文件配置在此文件中

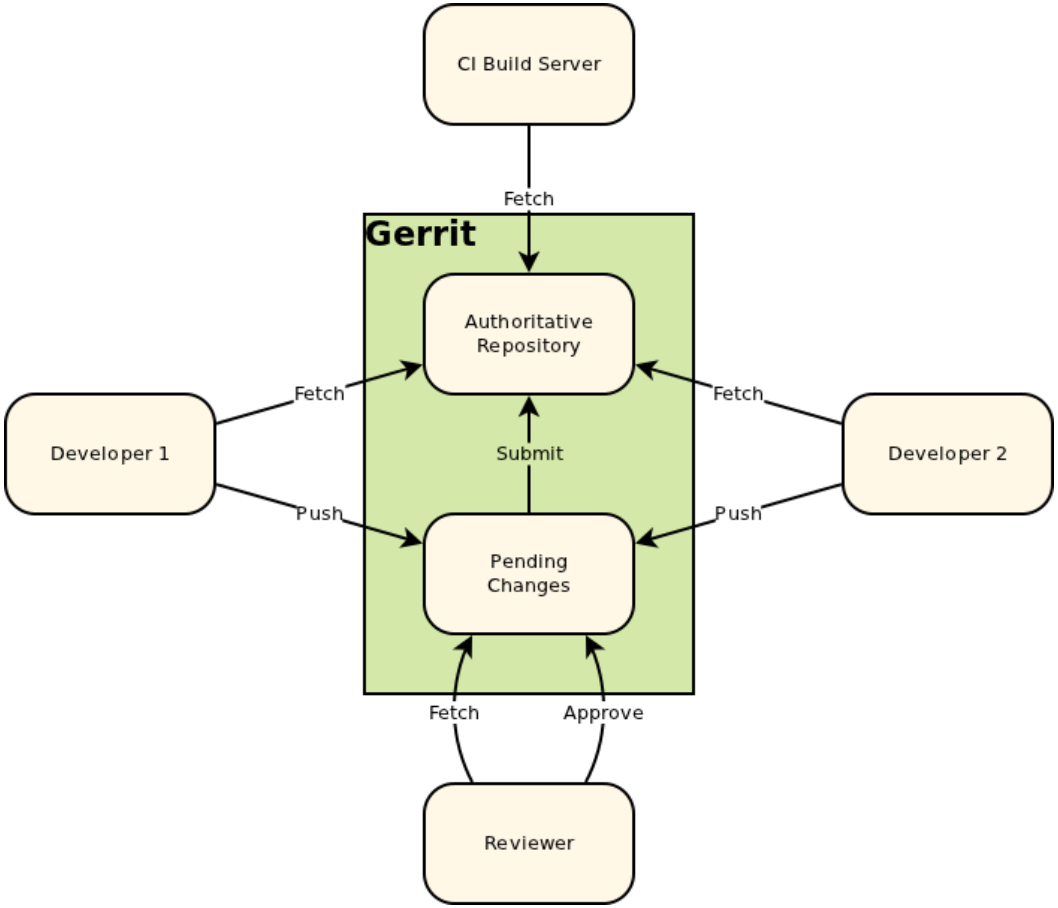
```
$ cat .gitignore
cscope.*
*.o
*.obj
/build/
/target/
/tcfs/
```

https://github.com/github/gitignore (github.com/github/gitignore)

gerrit使用

gerrit简介

refs/for/<branch_name>



git-review安装

使用pip安装

```
pip install git-review==1.25
```

pip环境配置

- windows下pip环境配置(https://wiki.zte.com.cn/display/CNNM/0060_00_02+Windows)
- linux下pip环境配置(https://wiki.zte.com.cn/display/CNNM/0060_00_01+Linux)

git-review使用

初始化本地环境

```
git review -s
```

提交评审

```
git review [BRANCHNAME] [--reviewers xxx@zte.com.cn yyy@zte.com.cn]
```

重新修改评审

```
git review -d change[,patchid]

#edit somethings
git commit -a --amend

git review [BRANCHNAME]
```

查看所有未关闭的评审

```
git review -l
```

gerrit-flows

本工作流建议禁用使用pull指令，使用fetch和rebase指令替代

首次变更review流程(1)

创建自己的本地工作分支并切换到自己的工作分支

```
git checkout -b feature/yourself_branch_name
```

修改文件,将本地修改提供到本地仓库

```
git commit -am 'for feature'
git commit -am 'for feature1'
git commit -am 'for feature2'
```

将本地多次提交合并成一个提交进行review

```
git rebase -i origin/master

# 编辑如下文档，只保留最上面的commit为pick，其它编辑为f
pick acdace4 test1
f    fa15458 test2
f    2f89fc6 test3
```

首次变更review流程(2)

将远端修改rebase到本地仓库

```
git fetch
git rebase origin/master
```

如果rebase过程中有冲突,需要解决冲突

```
git mergetool
git rebase --continue

# 冲突解决不下去了，可以终止此次rebase
git rebase --abort
```

提交review流程

```
git review [BRANCHNAME]
```

再次变更review流程(1)

切换到变更分支

```
git review -d chgid
```

编辑文件，再次提交变更

```
# edit

git commit -a --amend
```

再次变更review流程(2)

将远端修改rebase到本地仓库

```
git fetch
git rebase origin/master
```

如果rebase过程中有冲突,需要解决冲突

```
git mergetool
git rebase --continue

# 冲突解决不下去了，可以终止此次rebase
git rebase --abort
```

再次发起reivew, 并返回工作分支

```
git review -f [BRANCHNAME]
```

丢弃review

登录到gerrit网络，找到自己想要到丢弃的review，点击Abandon按钮
丢弃过review后再次review提交时，会报如下错误

```
git review [BRANCHNAME]
# 出现如下错误
remote:
remote: Processing changes: refs: 1, done
To ssh://id@gerrit.zte.com.cn:29418/xx/xxxx
 ! [remote rejected] HEAD -> refs/publish/master (no new changes)
  error: failed to push some refs to 'ssh://id@gerrit.zte.com.cn:29418/xx/xxxx'
```

遇到上述问题后，需要删除原来提交中的Change-Id

```
git commit --amend
# 将提交中的下行Change-Id删除掉
Change-Id: If5774e6af6499aa04433f0e9c769a8f965f27fa7
```

注意:Abandon后，再次commit后，一定要将新的commit和abandon的commit做提交合作(git rebase -i origin/master)

More information

Reference Material

- Learn Git Branching (https://learngitbranching.js.org/?locale=zh_CN)
- git manual (http://git-scm.com/book/zh/v2)
- Top 10 Git Tutorials for Beginners (http://sixrevisions.com/resources/git-tutorials-beginners/)
- 19 git tips for everyday use (http://www.alexkras.com/19-git-tips-for-everyday-use/)
- 10 tips git next level (http://www.sitepoint.com/10-tips-git-next-level/)
- git-review manual (https://docs.openstack.org/infra/git-review/)

Thank you

王寿林10067372