


Git, bilibili, 尚硅谷

2022.11.3



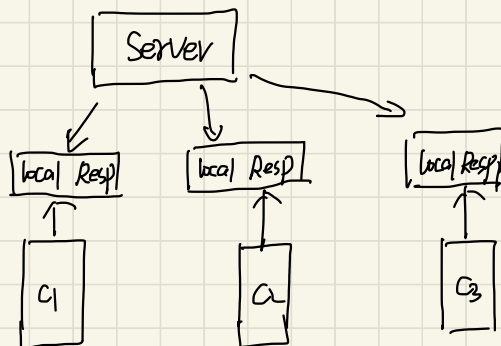
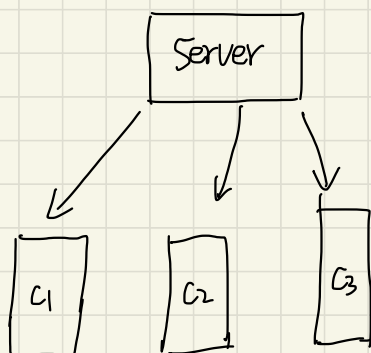
Git: 分布式版本控制工具

集中式版本控制工具

↳ everything is local: 描述其分布式特性。

版本控制: 记录文件内容变化, 以便之后用来查阅特定版本修订情况的系统

集中式版本控制工具 VS 分布式版本控制工具



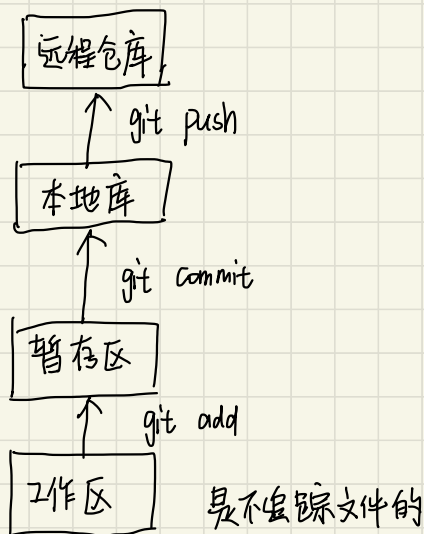
致命问题就是 SPoF

SPoF: Single point of Failure

单点故障

GitHub: 2008年上线:

Git工作机制:



GitLab 和 GitHub 和 Gitee (码云) 区别

GitLab, GitLab, 是局域网上的版本控制系统, 一般是公司内使用

GitHub 和 Gitee 是互联网上的版本控制系统,

查看本地库状态: `git status`

track

将文件添加到暂存区: `git add`, 添加到暂存区 `git` 才开始追踪文件

将文件从暂存区到本地库中 `git commit -m " " 文件名`

↓ `commit -m` 是添加描述信息

查看 `git` 历史版本: `git log`, `git log` 中有这样一句描述:

HEAD → dev-branch



这个是指针的意思, 指向哪个版本

`Commit "xxxx"`

↑ 有40位字符串, 这个是版本号

`git reflog`: 查看版本信息

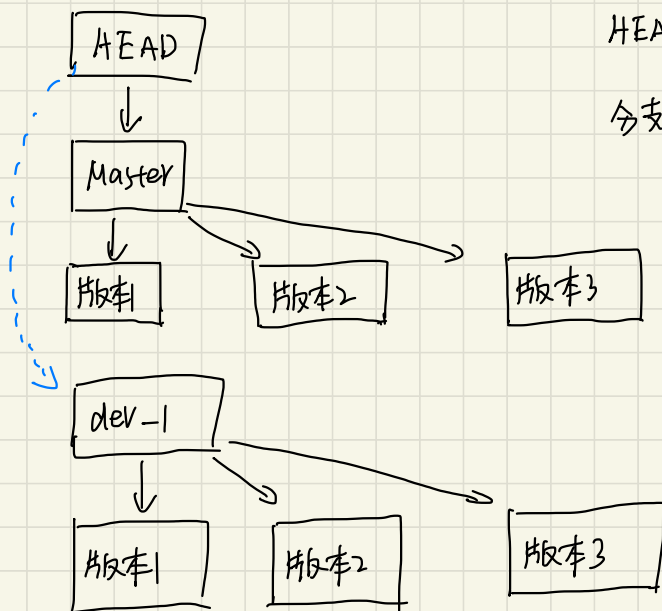
`git log`: 查看版本详细信息

版本穿梭: 让 `HEAD` 指针指向其他版本号,

`HEAD` 指针 和 分支 以及 分支 版本 之间关系:



HEAD指针和分支以及分支版本之间关系:



HEAD指针指向当前分支,

分支可以选择 reset 到任何版本

分支操作:

git branch 创建分支

git branch -v 查看分支

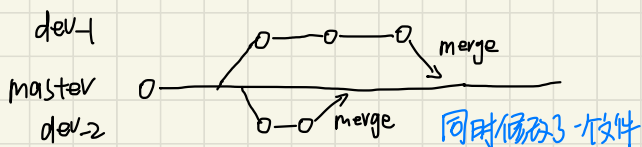
git checkout 分支名, 切换分支

git merge 分支名, 将别的分支 merge 到当前分支

代码冲突:

合并分支时, 两个分支在同一个文件的同一位置有两套不同修改

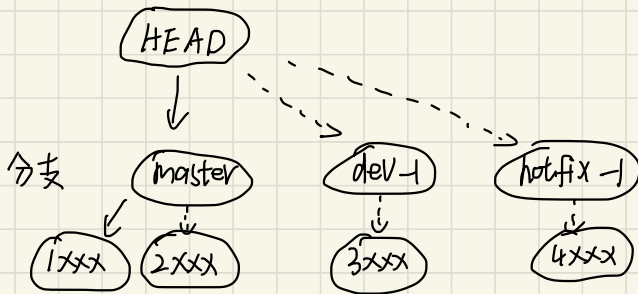
Git无法确定使用哪个分支时,



dev-2 merge -> master

① git是以行为单位来记录文件的变化:

② git实际是就是两个指针 HEAD + 版本指针



clone: 是指将 Remote 中文件 clone 到 local Resp 中

push: 将 local Resp $\xrightarrow{\text{push}}$ Remote Resp

pull: 将 Remote Resp $\xrightarrow{\text{pull}}$ local Resp

fork: 将 Remote Resp $\xrightarrow{\text{fork}}$ Remote Resp

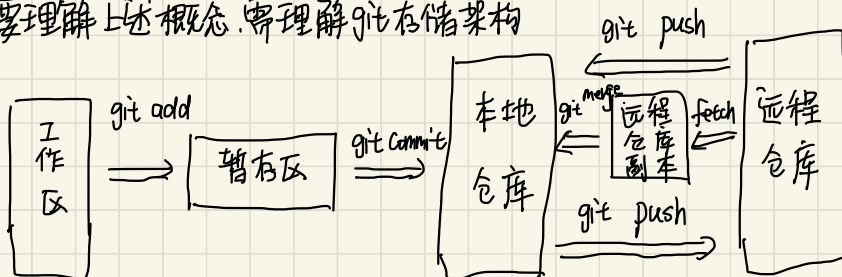
↙ 不是同一个 ↘

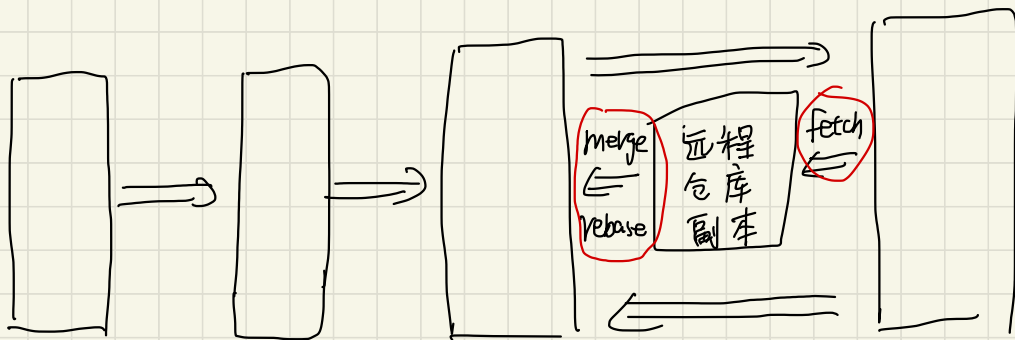
fork 是又的意思, 分岔意思

git fetch 和 git pull 区别:

↓ 取来, 拿来,

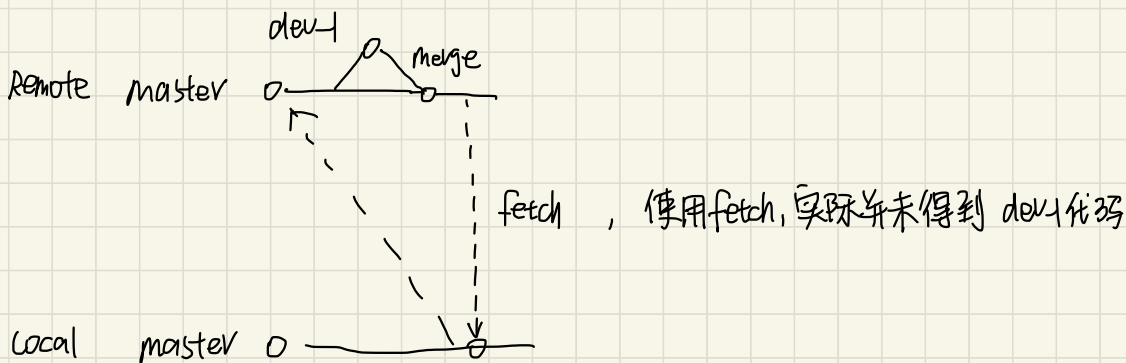
要理解上述概念, 需理解 git 存储架构





远程仓库副本：可理解为本地远程仓库缓存、

使用 fetch 时，代码并未合并到本地仓库、



$\text{git pull} = \text{git fetch} + \text{git merge}$

或

$\text{git fetch} + \text{git rebase}$

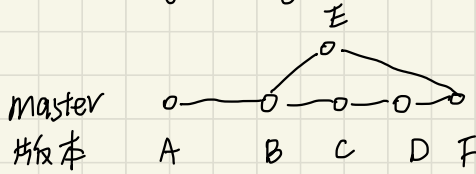
线上实操一下：fetch 并未在当前工作区中拉取到 remote Resp 中代码

其实这里是可以理解为：存在暂存区的概念

fetch 取出来，并未 merge 到当前 local Resp 中

git merge 和 git rebase 区别:

先来说 merge, merge 是合并的意思:



E 想要 merge 进 master 分支

master 会产生 新的节点

新的版本

Merge 会保留所有 commit 历史时间

master 主要是其他分支 merge.