

# 课题：课程概述

1. 为什么要版本管理
2. Git之于SVN的优点
3. 课程大纲



**此事有转折 并非那么简单**

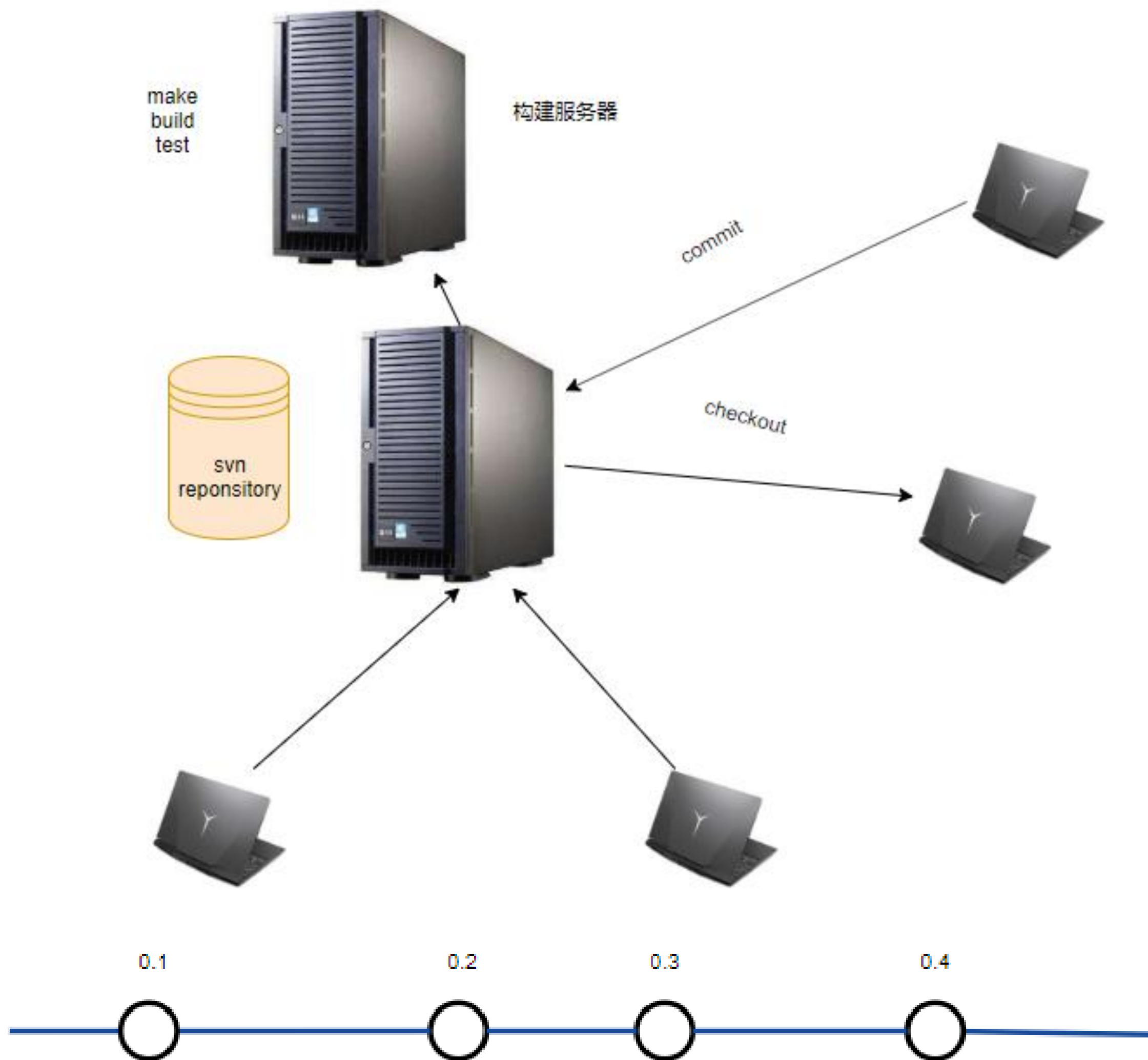
# 为什么要用版本管理

名称	修改日期	类型	大小
cihub	2019/6/30 1:44	文件夹	
golang	2019/6/30 1:44	文件夹	
luci	2019/6/30 1:44	文件夹	
orcaman	2019/6/30 1:44	文件夹	
rtmp_proxy	2019/6/30 1:44	文件夹	
satori	2019/6/30 1:44	文件夹	



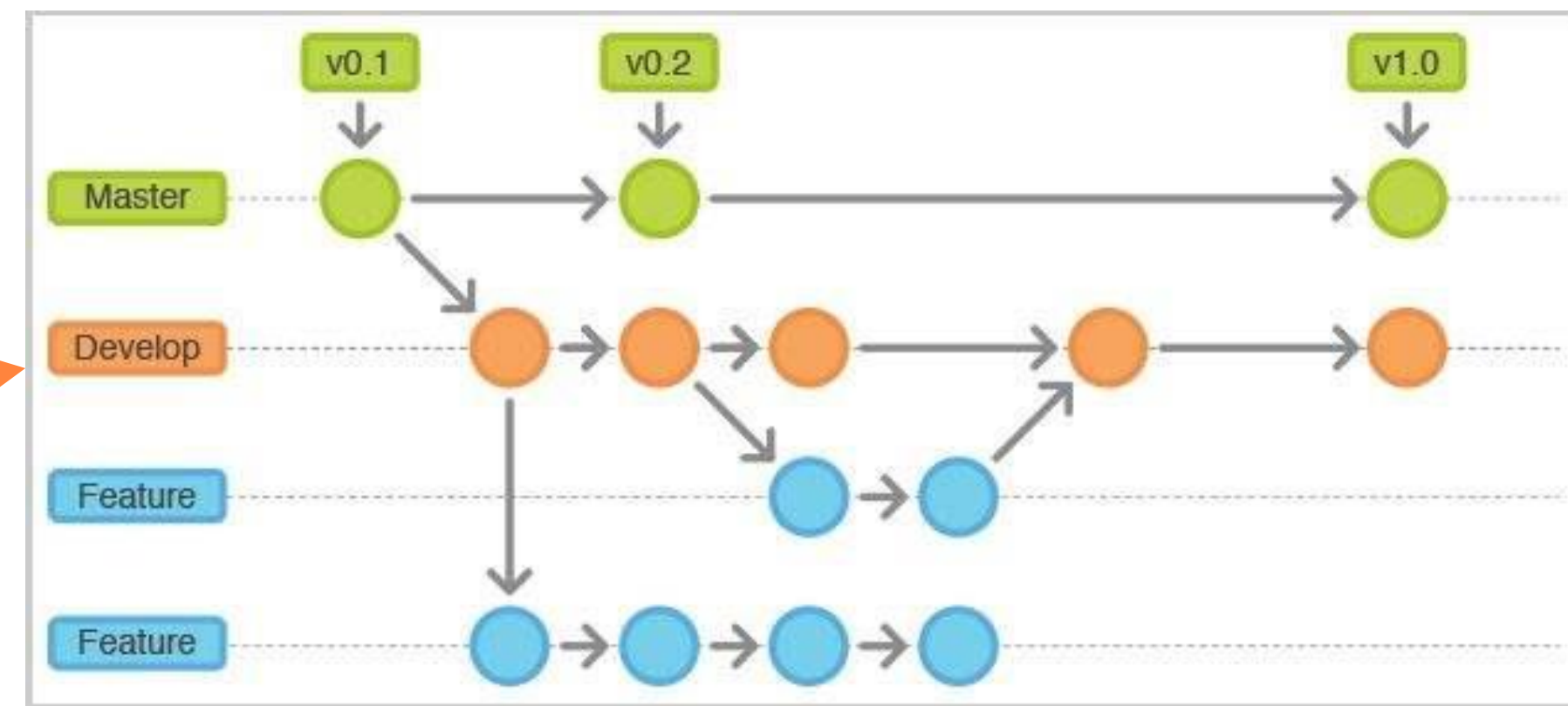
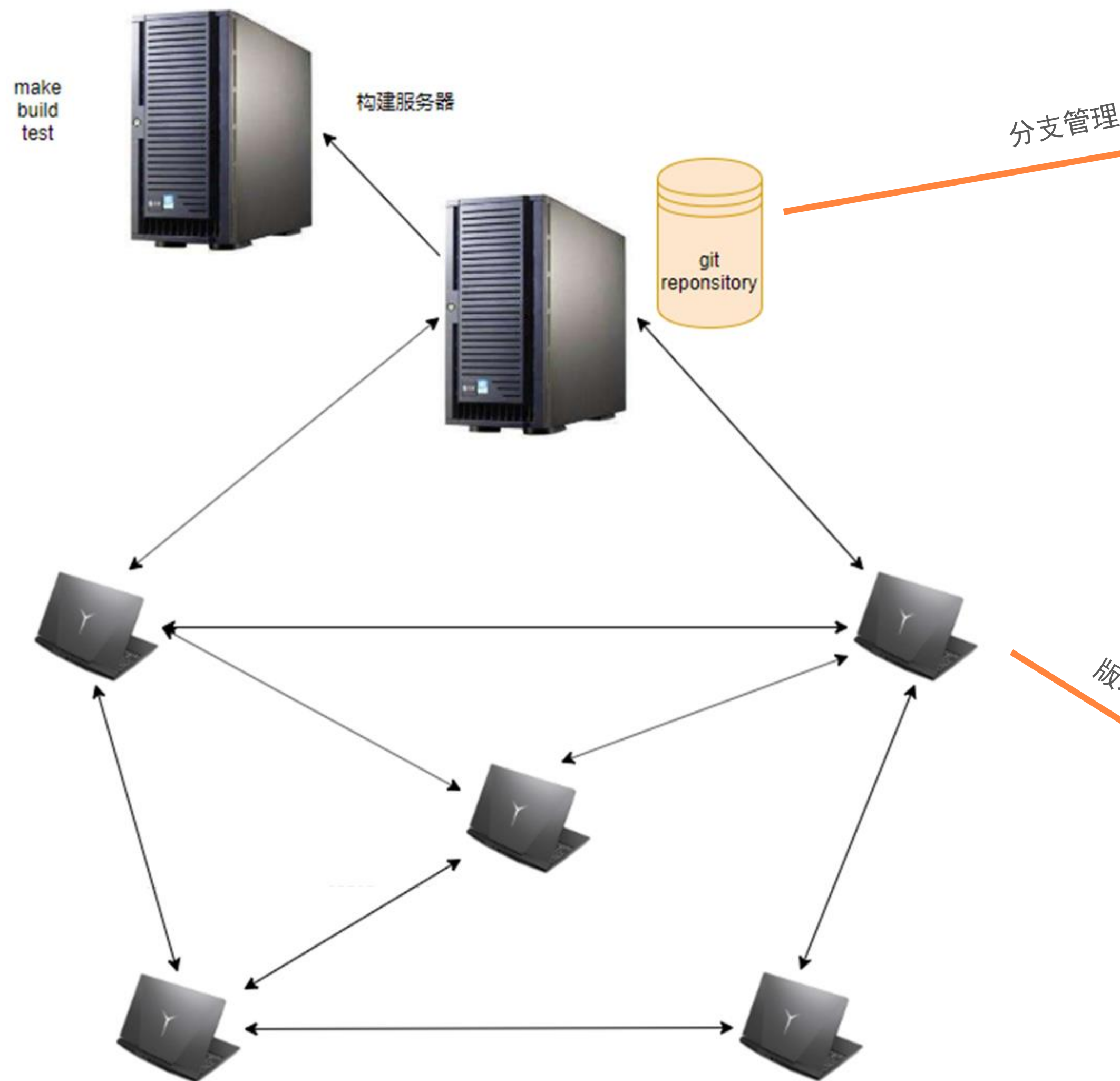
名称	修改日期	类型	大小
mcu_proxy	2019/6/30 1:44	文件夹	
mcu_proxy - 7.15	2019/8/20 10:20	文件夹	
mcu_proxy - 7.20 - mcu	2019/8/20 10:22	文件夹	
mcu_proxy - 7.25	2019/8/20 10:22	文件夹	
mcu_proxy - 7.30	2019/8/20 10:22	文件夹	
mcu_proxy - 8.02	2019/8/20 10:24	文件夹	
mcu_proxy - 8.02 - failed	2019/8/20 10:24	文件夹	
mcu_proxy - 8.05 - reverted	2019/8/20 10:24	文件夹	
mcu_proxy - PR201913498	2019/8/20 10:20	文件夹	

# 版本控制



1. 版本管理有一个中央服务器，可以保存所有代码、文档
2. 每一次的修改都可以提交到版本库，修改有记录，可追踪
3. 不害怕某个同事离职了，代码没有入库
4. 本地的代码流失后，可以从版本库检出
5. 多人协作，每个同事完成的工作提交到版本库，方便进行集成
6. 当我们要开发需求或修复PR时，可以从版本库上拉出分支管理
7. 在大的企业，每次提交都可能触发一次构建，实时检查代码的质量
8. 如果构建失败了，可以自动revert掉某次提交

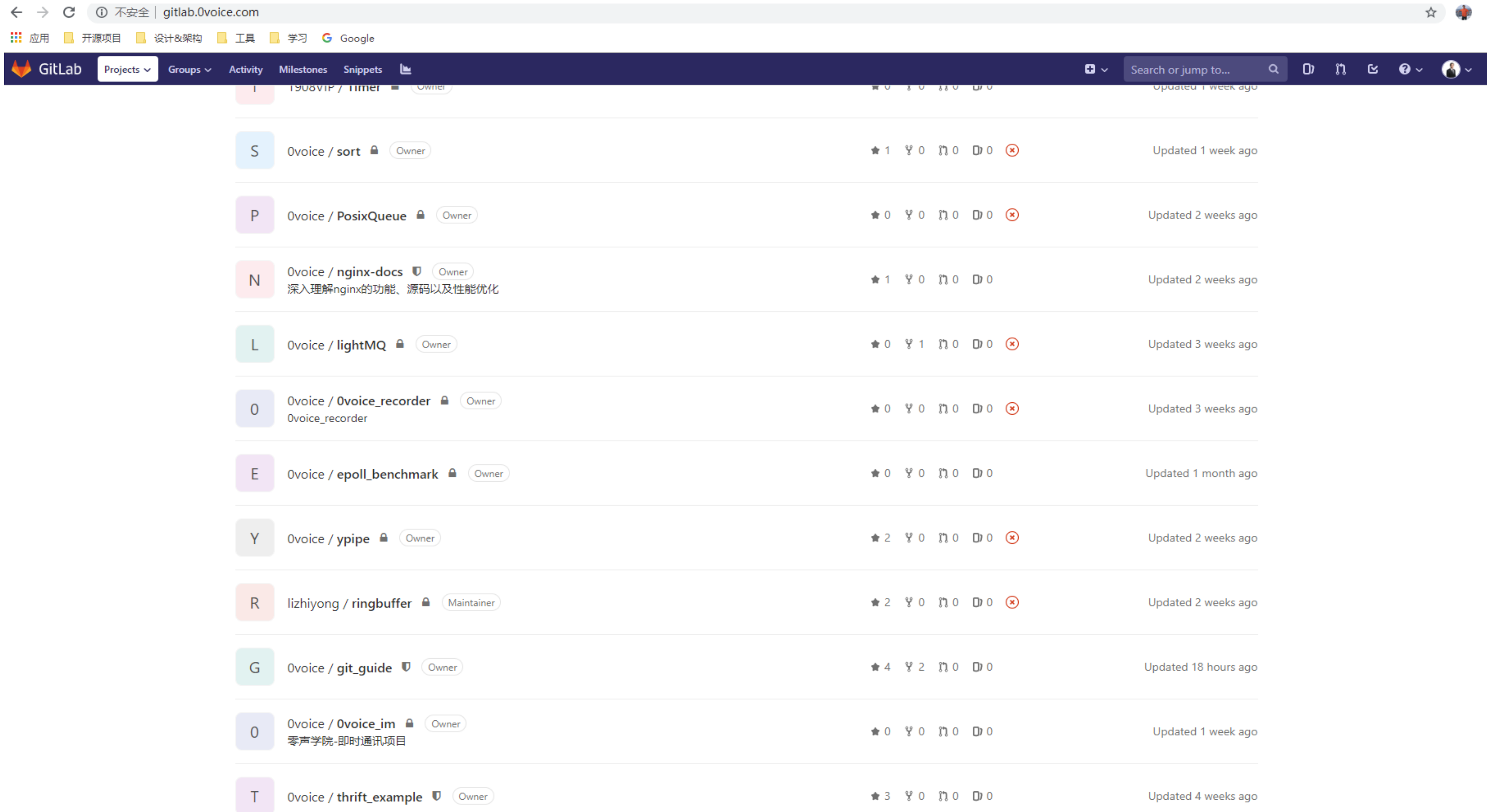
# 分布式与集中式的对比



```
nginx: master - gitk
File Edit View Help
* master remotes/origin/branches/default remotes/origin/master Geo: fixed handling of
  Upstream: implemented $upstream_bytes_sent.
  Resolver: report SRV resolve failure if all A resolves failed.
  Copy regex unnamed captures to cloned subrequests.
  Version bump.
  release-1.15.7 tag
  release-1.15.7 nginx-1.15.7-RELEASE
  Updated OpenSSL used for win32 builds.
  Negative size buffers detection.
  Mp4: fixed possible pointer overflow on 32-bit platforms.
  Limit req: "delay=" parameter.
  Limit req: fixed error message wording.
  Upstream: revised upstream response time variables.
  Upstream: removed unused ngx_http_upstream_t.timeout field.
  Core: ngx_explicit_memzero().
  Core: free shared memory on cycle initialization failure.
SHA1 ID: ce4a23d144762cfa27c0e4b13f74cada2f7486a8 Row 1 / 6483
Find commit containing:
Search
Diff Old version New version Lines of context: 3
Author: Maxim Dounin <mdounin@mdounin.ru> 2018-12-14 23:11:06
Committer: Maxim Dounin <mdounin@mdounin.ru> 2018-12-14 23:11:06
Parent: cb4dd56771c1af082bf3e810436712b4f48f2cf2 (U)
Branches: master, remotes/origin/branches/default,
remotes/origin/master
Follows: release-1.15.7
Precedes:
Geo: fixed handling of AF_UNIX client addresses
Previously, AF_UNIX client addresses were handled
to unexpected results.
----- src/http/modules/ngx_http_geo_module.c
index c11bafa6..153b6aaf 100644
```



100



# 课程目录

- 前言
- 第1章 Git基础
  - 1.1 课程概述
  - 1.2 Linux上安装git
  - 1.3 Mac OS上安装git
  - 1.4 Windows上安装git
  - 1.5 配置git环境
  - 1.6 获取帮助
- 第2章 git常用命令
  - 2.1 创建仓库
  - 2.2 搭建git服务器
  - 2.3 git的基础原理
  - 2.4 提交修改并push到远程仓库
  - 2.5 合并commit为一个
  - 2.6 修改commit的message
  - 2.7 查看commit的信息
  - 2.8 版本回退
  - 2.9 删除文件
  - 2.10 查看不同提交的指定文件的修改
  - 2.11 多客户端之间的同步
  - 2.12 处理突发事件
  - 2.13 指定不需要git管理的文件
  - 2.14 如何解决项目之间的依赖
  - 2.15 如何备份git仓库到本地
  - 2.16 git的基础常用命令
- 第3章 git分支与tag
  - 3.1 创建与合并分支
  - 3.2 解决冲突
  - 3.3 分支管理策略
  - 3.4 Bug分支
  - 3.5 Feature分支
  - 3.6 多人协作
  - 3.7 创建标签
  - 3.8 操作标签
  - 3.9 标准化的开发工作流
- 第4章 全球最大的开源项目管理基地github
  - 4.1 初识 GitHub
  - 4.2 加入 GitHub
  - 4.3 Git 速成
  - 4.4 向GitHub 提交代码
  - 4.5 启用issue跟踪需求和缺陷
  - 4.6 如果保证团队开发质量
- 第5章 企业里为什么喜欢使用gitlab
  - 5.1 搭建gitlab
  - 5.2 gitlab权限管理
  - 5.3 gitlab的codereview
  - 5.4 团队知识管理方法

# 下 节 课 再 见



联系Lee老师



课程顾问微信



课程咨询微信：2207032995