

GIT

版本控制之王

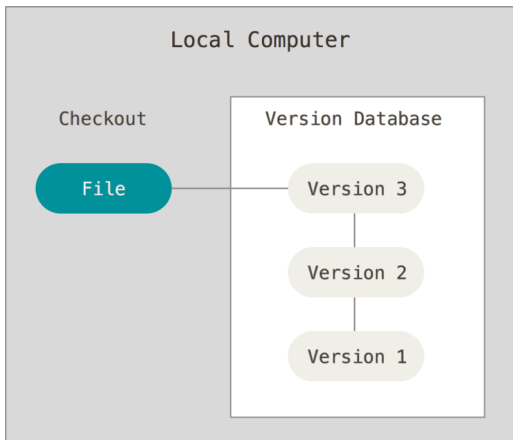
范 輝

Rust Engineer
BeiJing, China



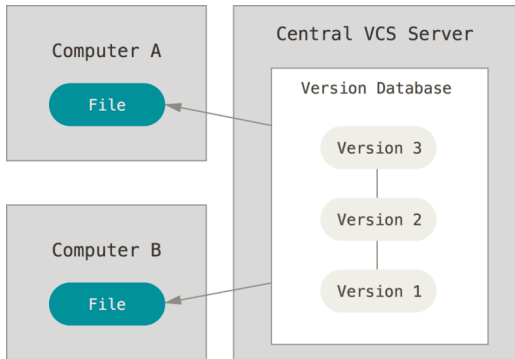
December 31, 2020

第一代 VCS



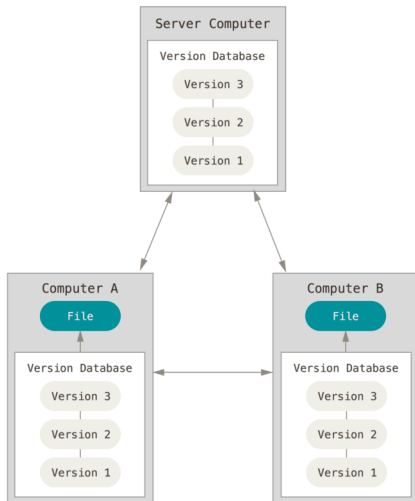
- 单点架构
- 本地存储

第二代 VCS



- C/S 架构
- 中心化存储
- 典型代表: CVS/Svn

第三代 VCS



- P2P 架构
- 分布式存储
- 典型代表: Git



- 为自由之信仰而生
- 出身名门，天生不凡
- 实力碾压一切牛鬼蛇神

Git之父

Linus Torvalds

- Linux之父
- 世界级开源领袖
- 人类史上十大黑客之一



Git 的目标

- Speed
- Simple design
- Strong support for non-linear development (thousands of parallel branches)
- Fully distributed
- Able to handle large projects like the Linux kernel efficiently (speed and data size)

Git 的优势

- 业界最流行方案，主流 IDE 默认集成
- 运行速度比 Svn 快数倍，甚至数百倍
- 分布式架构，支持离线完成绝大多数工作
- 极其轻量的本地分支
- 便捷的分支合并流程
- 强大的数据安全保证
- 创新的缓存区设计
- 免费且开源
- ...

Git PK Svn

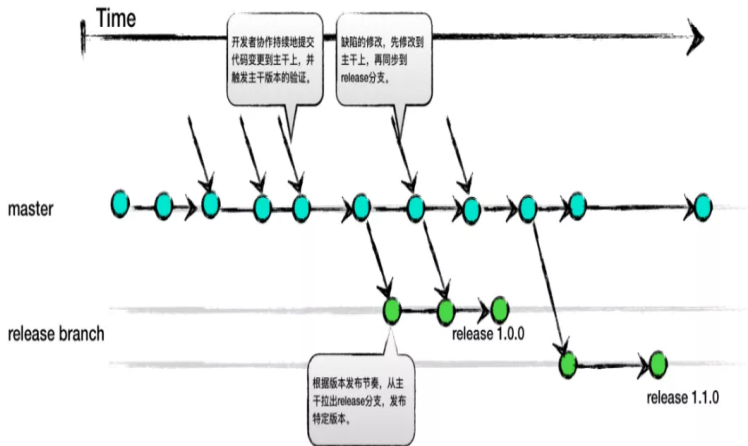
Operation		Git	SVN	
Commit Files (A)	Add, commit and push 113 modified files (2164+, 2259-)	0.64	2.60	4x
Commit Images (B)	Add, commit and push a thousand 1 kB images	1.53	24.70	16x
Diff Current	Diff 187 changed files (1664+, 4859-) against last commit	0.25	1.09	4x
Diff Recent	Diff against 4 commits back (269 changed/3609+, 6898-)	0.25	3.99	16x
Diff Tags	Diff two tags against each other (v1.9.1.0/v1.9.3.0)	1.17	83.57	71x
Log (50)	Log of the last 50 commits (19 kB of output)	0.01	0.38	31x
Log (All)	Log of all commits (26,056 commits – 9.4 MB of output)	0.52	169.20	325x
Log (File)	Log of the history of a single file (array.c – 483 revs)	0.60	82.84	138x
Update	Pull of Commit A scenario (113 files changed, 2164+, 2259-)	0.90	2.82	3x
Blame	Line annotation of a single file (array.c)	1.91	3.04	1x

Git 的应用场景

- 版本控制（代码、文档等）
- 多人协作
- 项目管理（GitHub/GitLab）
- 兼容 Svn
- CI/CD
- ...

TBD

Trunk-Based Development



TBD

优缺点

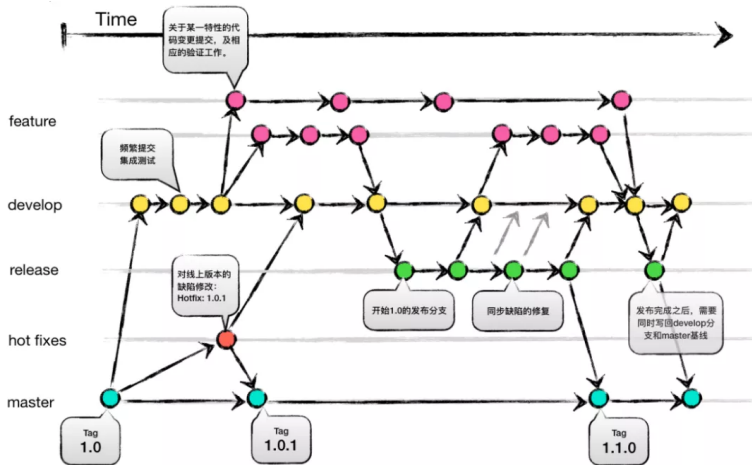
优点

- 分支少
- 操作简单

缺点

- 并行开发支持度低
- 仅适用于简单版本控制场景

Git-Flow



Git-Flow

优缺点

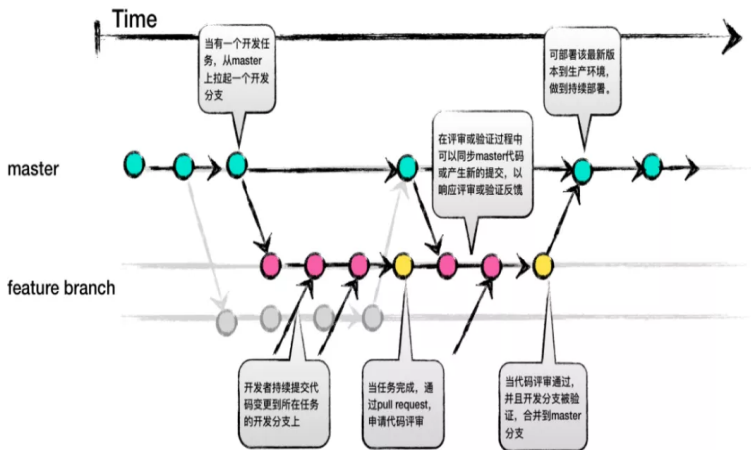
优点

- 分支职责清晰
- 完善的并行开发支持
- 适用于大项目大团队场景

缺点

- 分支多
- 代码冲突概率大

GitHub-Flow



GitHub-Flow

优缺点

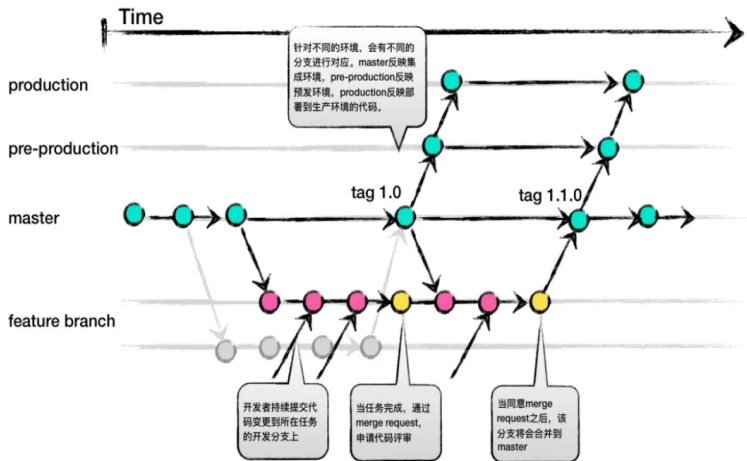
优点

- 分支逻辑简单
- 较高的并行开发支持
- 方便与 CI/CD 系统集成

缺点

- 分支管理不够严谨

GitLab-Flow



GitLab-Flow

优缺点

优点

- 分支管理较严谨
- 较高的并行开发支持
- 方便与 CI/CD 系统集成

缺点

- 分支与部署环境过度耦合
- 是一种折中方案，复杂度较高

总览

GitLab 提供免费的社区版安装包

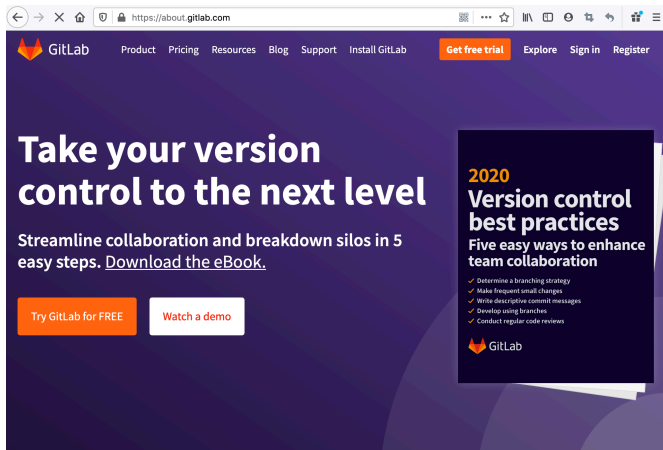
GitHub 世界范围内最知名的 Git 协作平台

BitBuck 清爽简洁，有一定的用户规模

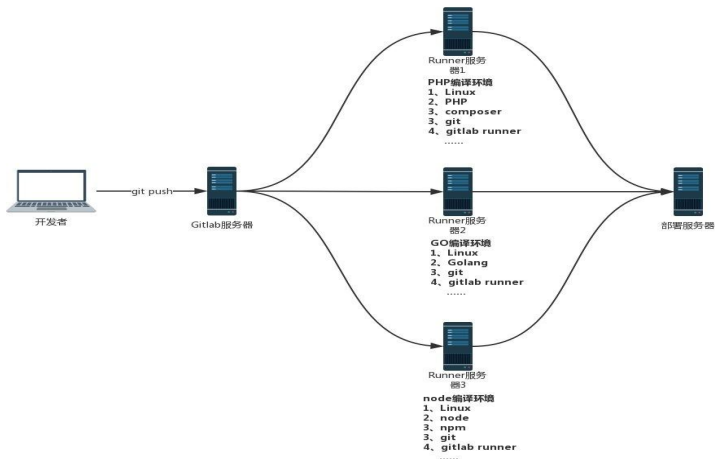
GitEE 国内最优秀的协作平台，基于 GitLab 开发

Coding 国内的另一知名协作平台，腾讯系

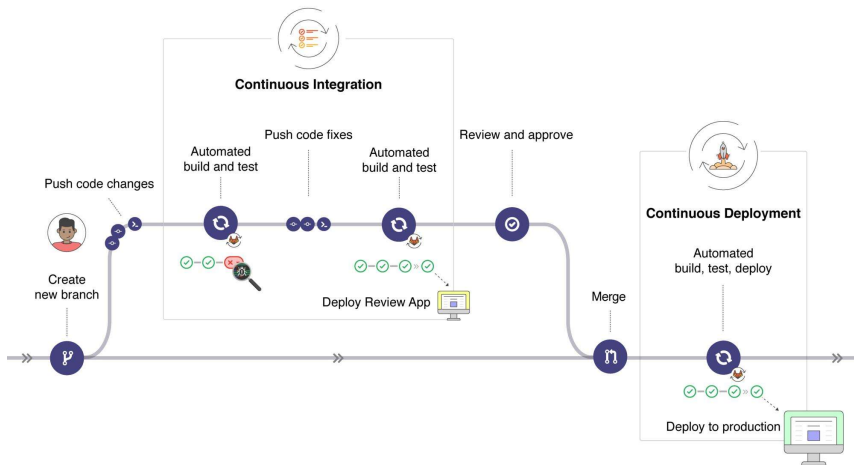
GitLab



GitLab CI/CD (一)



GitLab CI/CD (二)



GitLab Issue

Tanuki-Group

Issue Boards

Label: Discussion Add list

performance 8

- Issues data-when payload is using #41838 SL2 bug
- Issues inside Changes-tab renaming file SL2 backend bug
- Issue' using project name at point of 79 SL2 backend bug
- Slow branch picker when creating a merge request Discussion SP2 UX ready awaiting feedback backend customer frontend merge requests performance
- Search is slow (non-Elasticsearch version) AP2 Discussion backend frequently duplicated performance search
- Show a message when loading the issues / merge requests dashboard without filters Deliverable Discussion UX ready frontend issues merge requests performance
- Controller Projects::IssuesController#show executes more than 100 SQL queries AP2 Discussion backend database performance
- Controller Projects::MergeRequestsController#approvals executes more than 100 SQL queries AP2 Discussion backend database

elasticsearch 3

- ElasticSearch "Exact Term" broken when used with special characters. Discussion SP2 backend bug customer elasticsearch
- Elasticsearch with AWS still getting throttled due to instance profile credentials Discussion backend bug customer elasticsearch regression
- Elasticsearch developer documentation Discussion backend elasticsearch

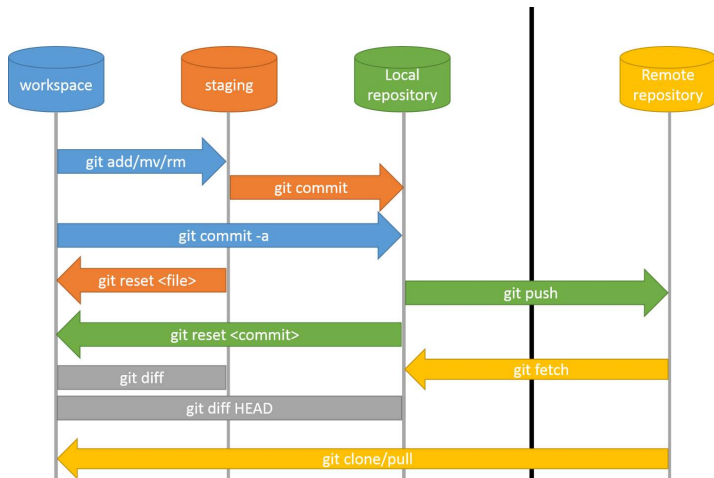
api 1

- API to return all issues / merge requests of all subgroups in the groups list Discussion api backend direction issues merge requests subgroups

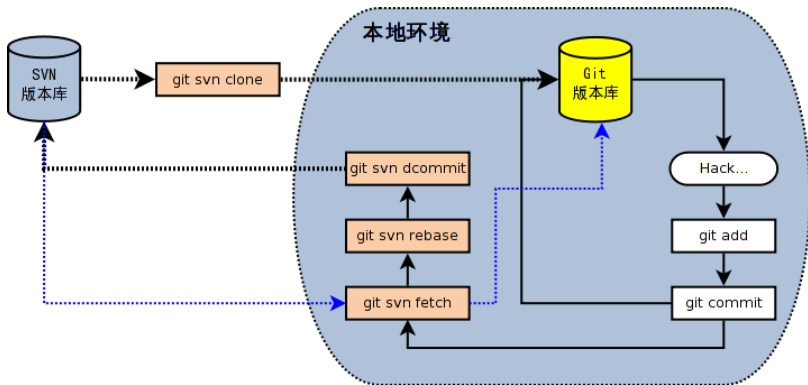
Labels

- SL2
- bug
- backend
- customer
- frontend
- merge requests
- performance
- search
- frequently duplicated
- database
- regression

基本操作



操作 Svn



其它常用操作（一）

```
# 查看分支
git branch -a
# 切换至指定分支
git chekcout <BranchName>
# 基于当前分支创建并切换至新分支
git checkout -b <BranchName>
# 合并整个目标分支至当前分支
git merge --no-ff <BranchName>
# 合并指定的单次提交至当前分支
git cherry-pick <CommitID>
# 删除本地分支
git branch -D <BranchName>
# 重命名本地分支
git branch -m <OldName> <NewName>
```

其它常用操作（二）

```
# 添加远程仓库地址
git remote add <RemoteName> <RemoteAddr>
# 修改远程仓库地址
git remote set-url <RemoteName> <RemoteAddr>
# 删除远程仓库地址
git remote remove <RemoteName>
# 删除远程分支
git push --prune <RemoteName> <RemoteBranch>
# 创建新标签
git tag <TagName> <BaseCommitID>
# 删除标签
git tag --delete <TagName>
# 推送本地标签至远程仓库
git push --tags
```

其它常用操作（三）

```
# 基于指定的起点，重新整理提交记录
git rebase -i <BranchName>
# 将所有未提交的更改收入隐藏区(垃圾桶)
git stash
# 清理所有未纳入版本库管理的文件与目录
git clean -fdx
# 查看近期所有分支上的操作日志
git reflog
# 查看内容有变动的文件列表
git diff --name-only <BranchName>
# 子模块递归更新
git submodule update --init --recursive
# 定义指令别名
git config --global alias.<Alias> <SubCMD>
```

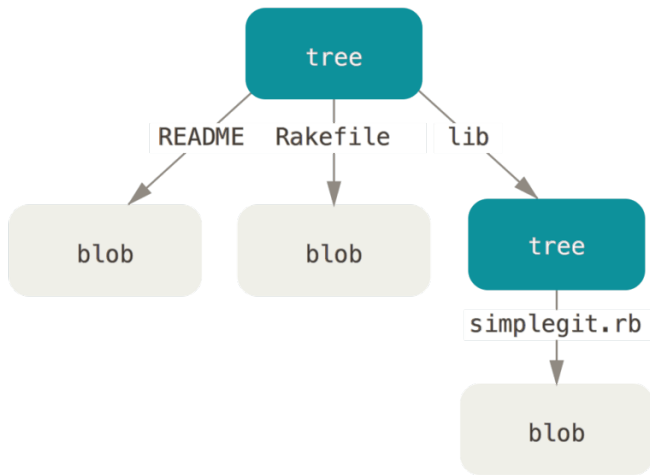
数据存储格式（一）

- 数据存储的核心是一个 KV 数据库
- 数据单位通过 SHA1 散列值进行寻址
- 整体结构类似于 UNIX 文件系统布局

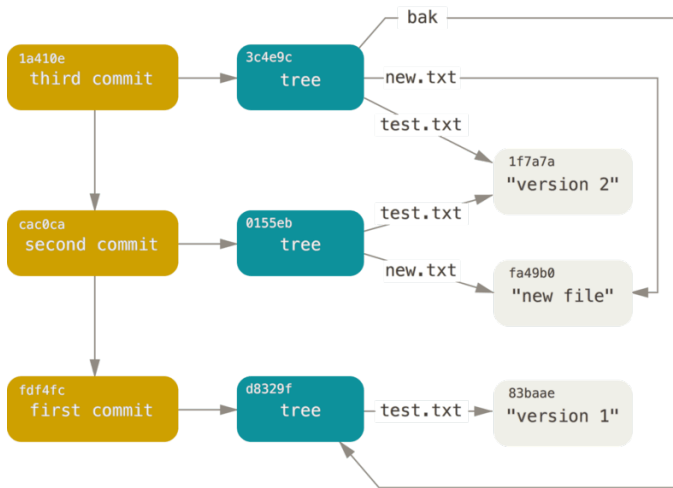
```
$ cd .git/objects && find . -type f
```

```
./1f/7a7a472abf3dd9643fd615f6da379c4acb3e3a  
./83/baae61804e65cc73a7201a7252750c76066a30  
./d6/70460b4b4aece5915caf5c68d12f560a9fe3e4
```

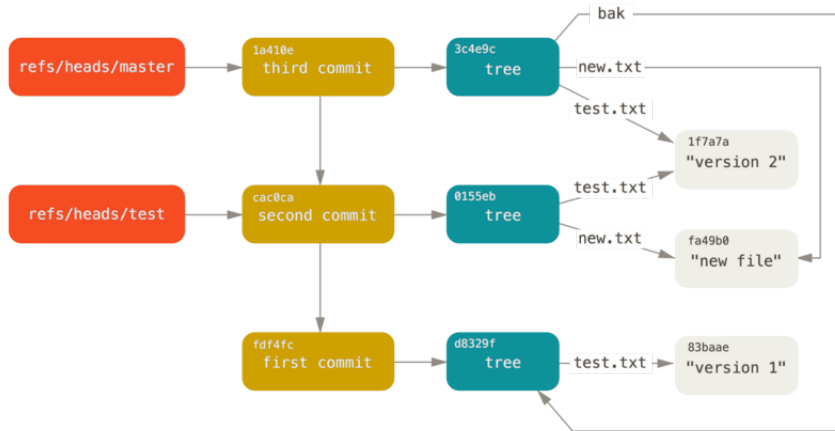
数据存储格式（二）



数据存储格式（三）



分支与标签的真相



- 愉悦的团队合作
- 巨大的效率提升

Come On !

进一步学习

Pro Git

<https://git-scm.com/book/zh/v2>

Git 源码

<https://github.com/git/git>

Git 第三方开发库

<https://libgit2.org>