

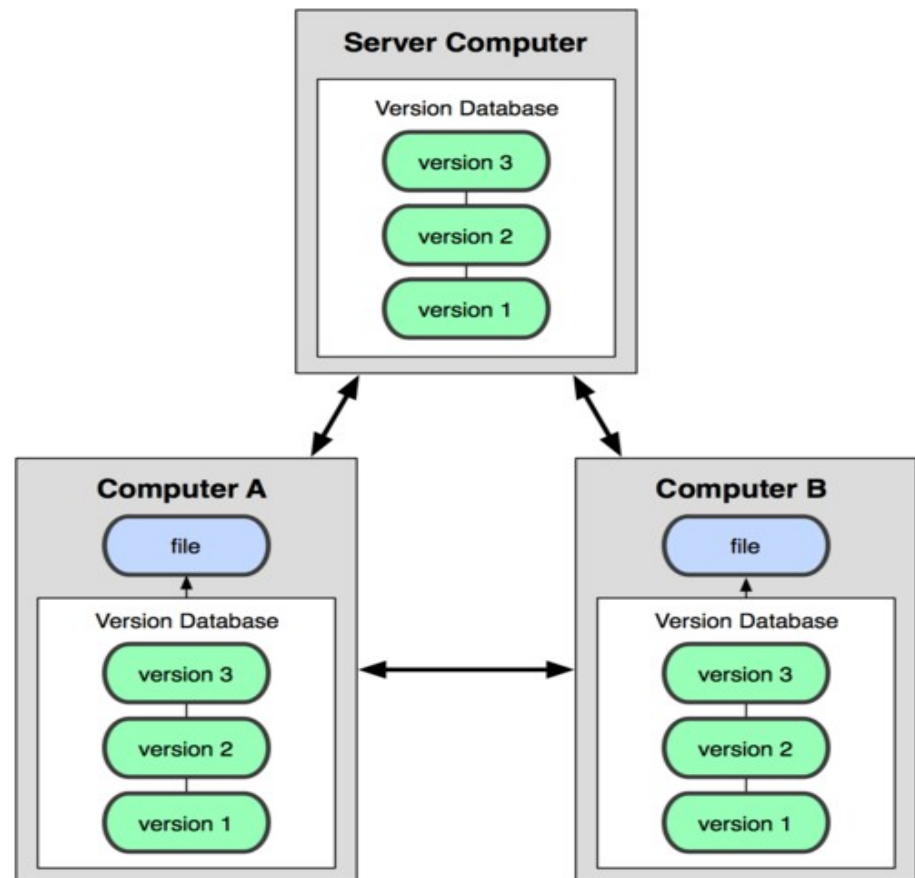
git

git

- 原理
- 分支
- 工作模式
- 工具

原理

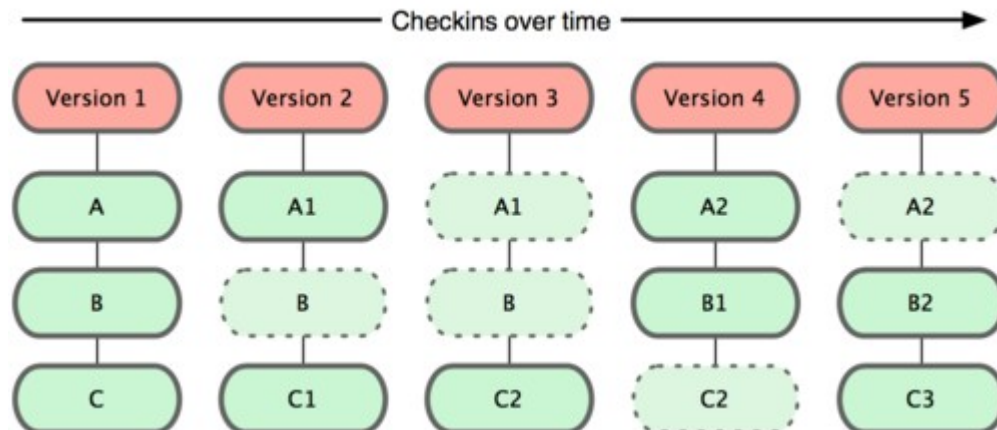
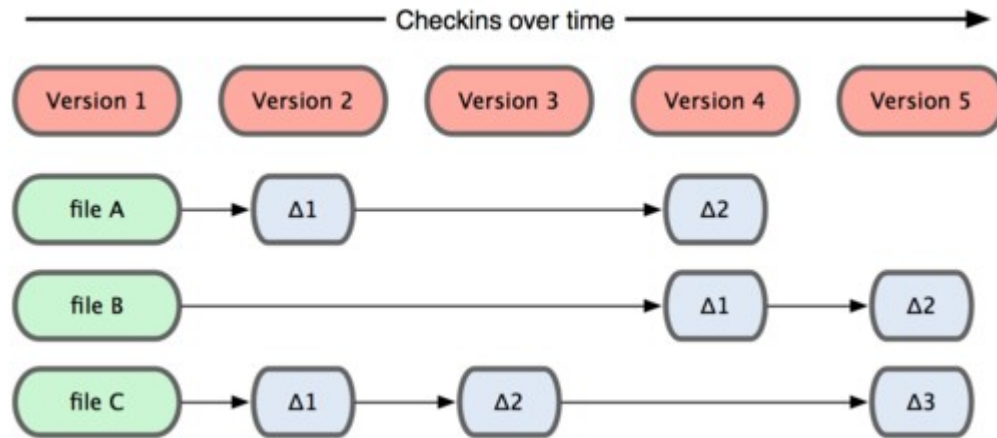
- 分布式
- 大部分操作在本机进行
- 大部分操作可离线



原理

- 文件快照

- 完整文件
- 大部分操作在本机进行
- 快速恢复到任意版本
- 时刻保证数据完整 :SHA1



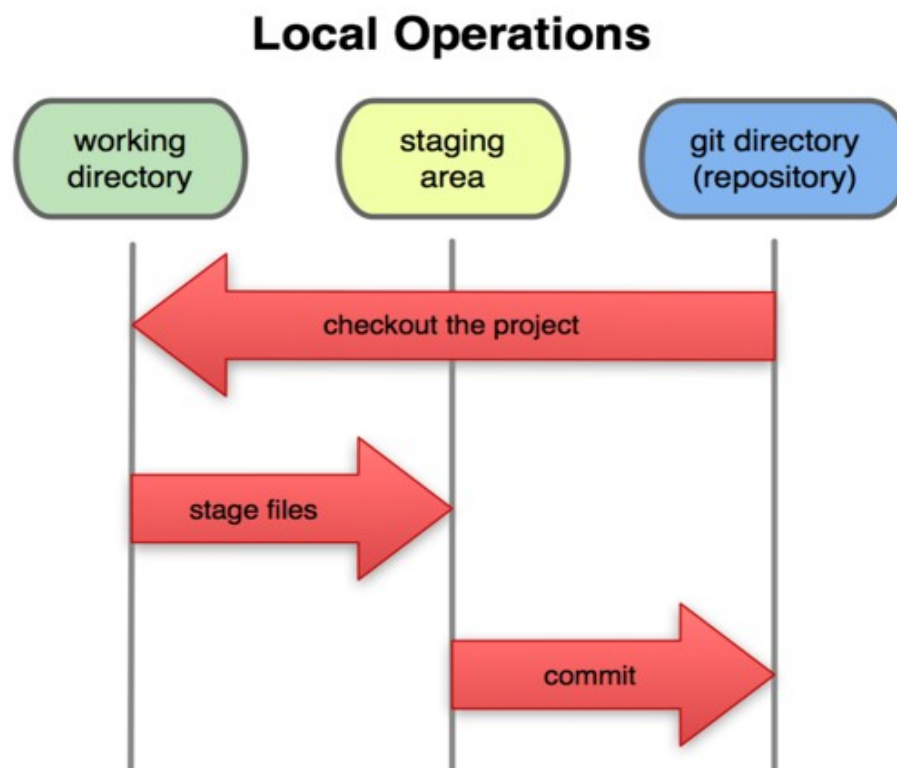
原理

- 文件状态

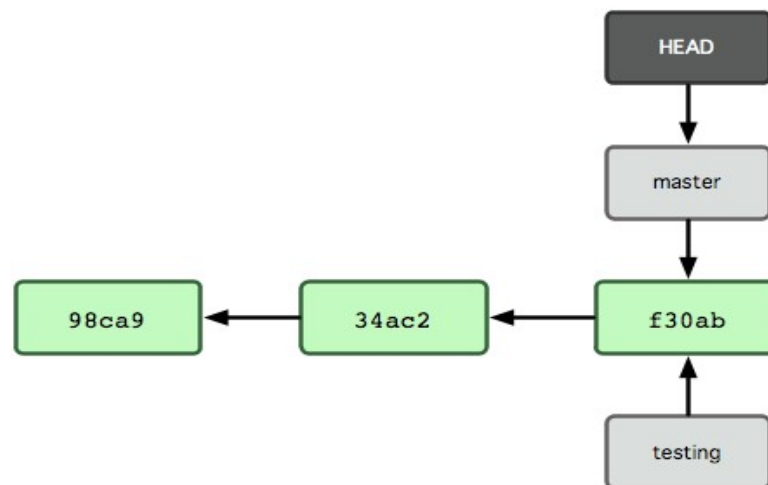
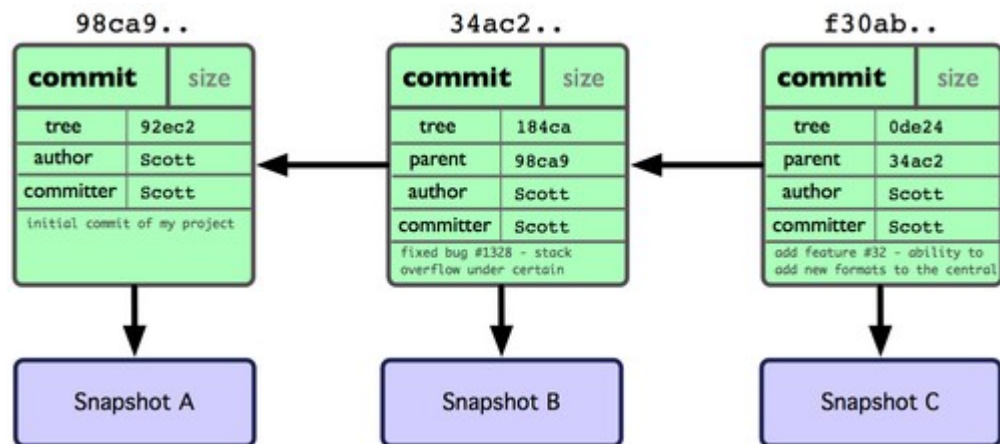
已提交

已暂存

已修改

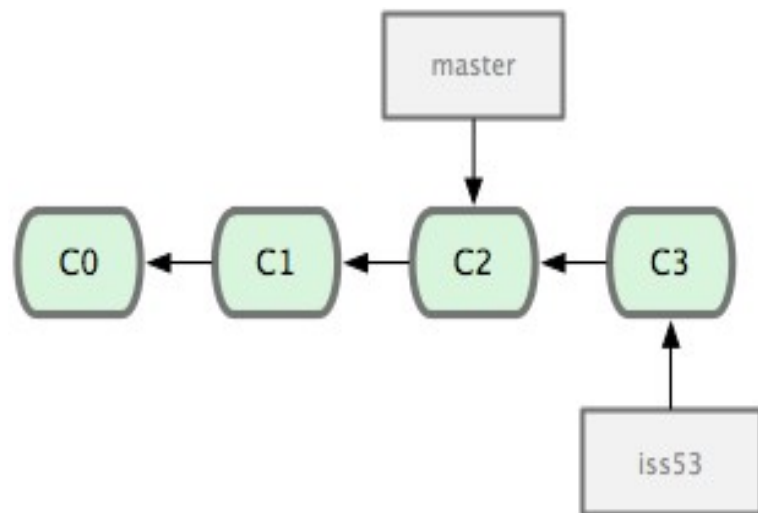
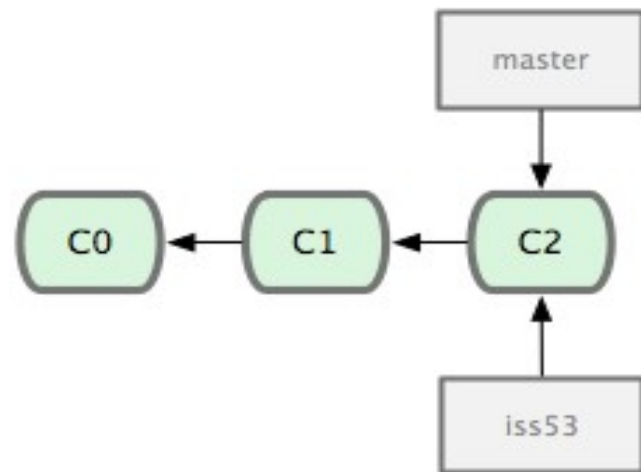


分支



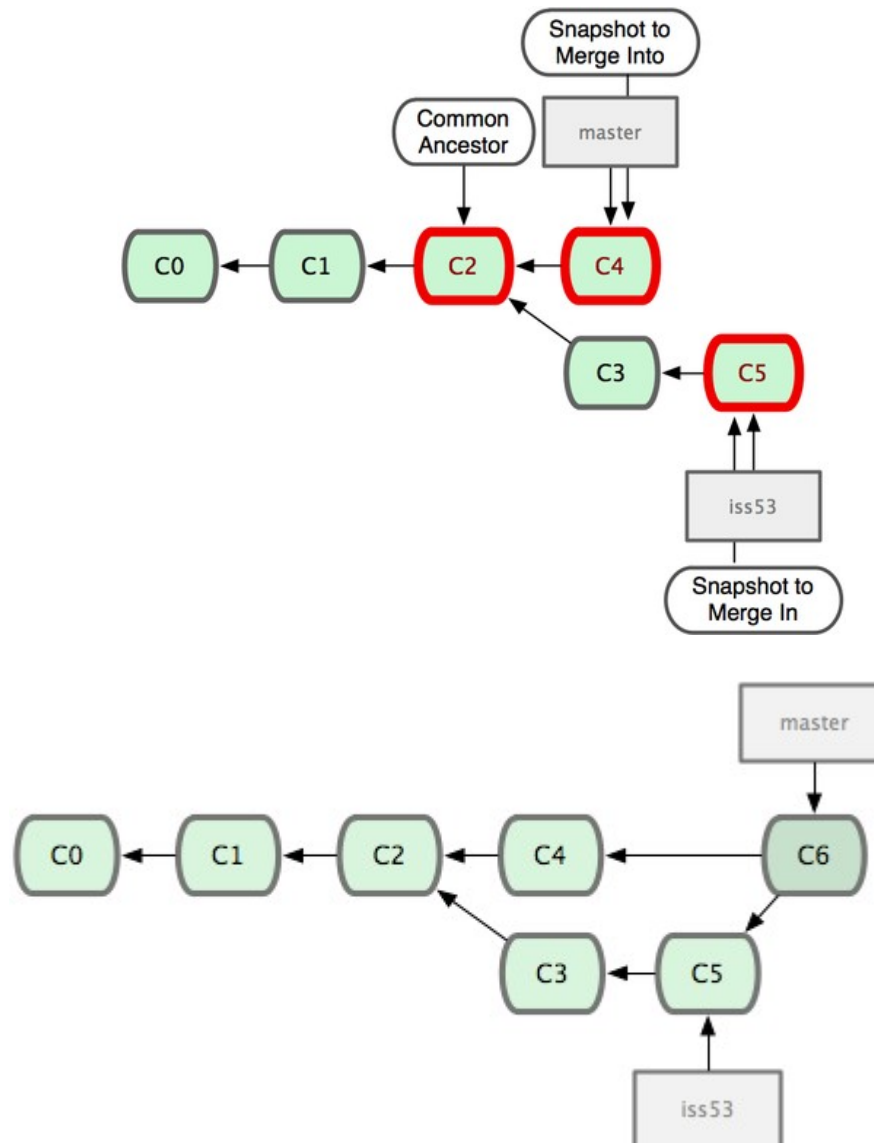
分支操作

- 新建分支



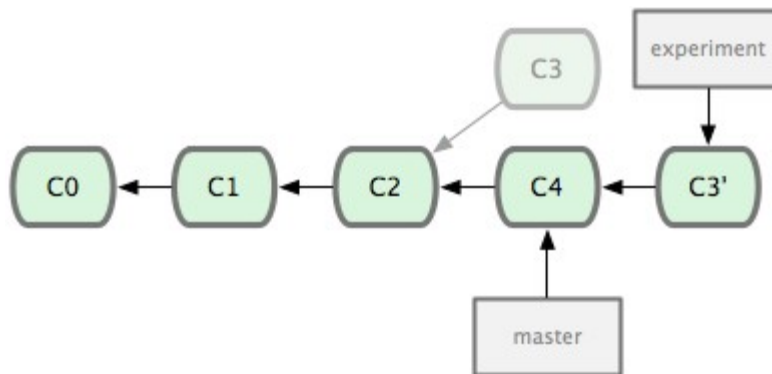
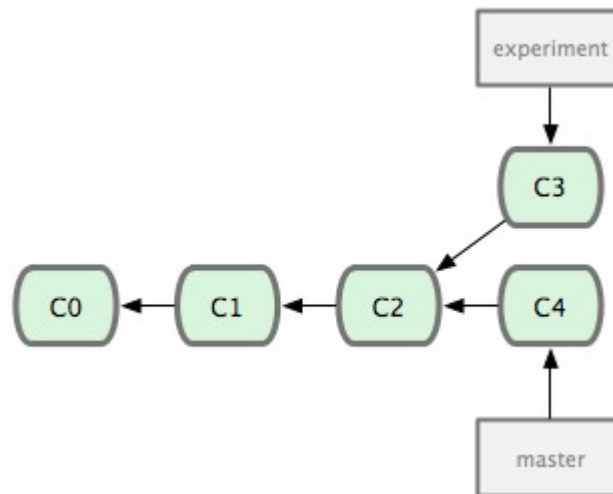
分支操作

- 合并 merge



分支操作

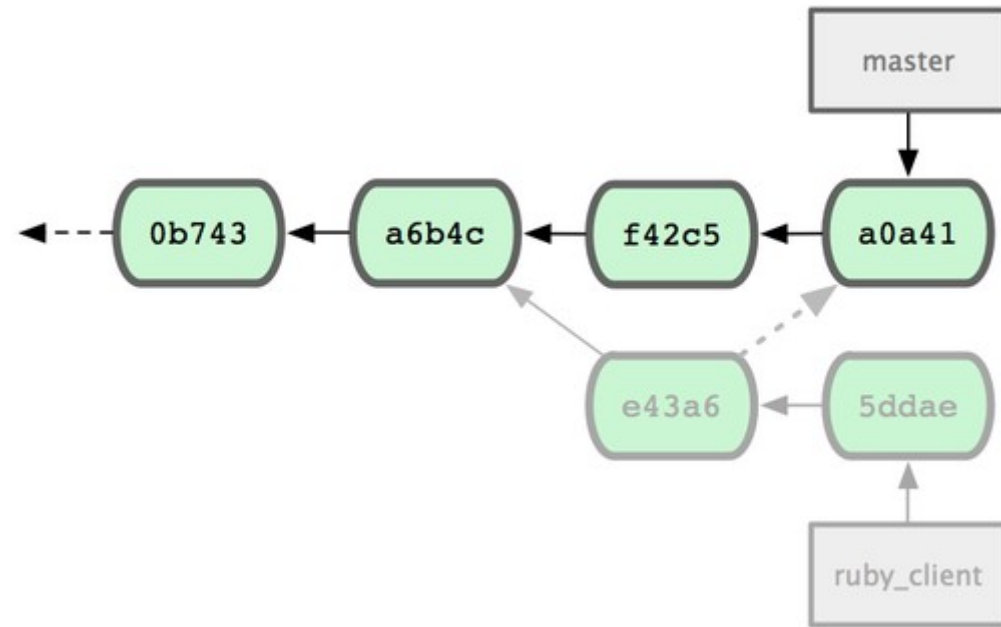
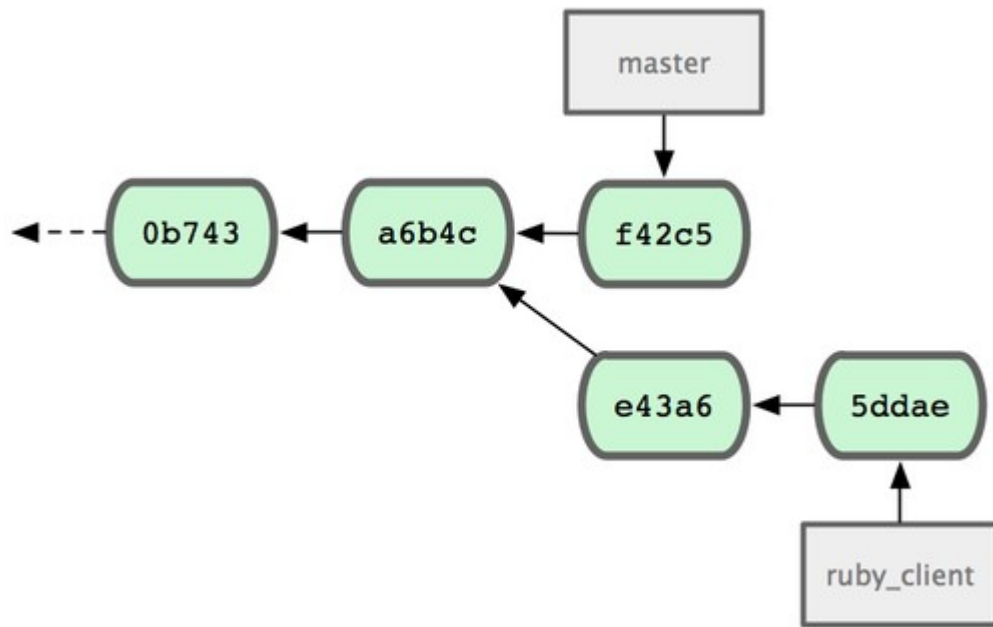
- 衍合 rebase



一旦分支中的提交对象发布到公共仓库，就千万不要对该分支进行衍合操作

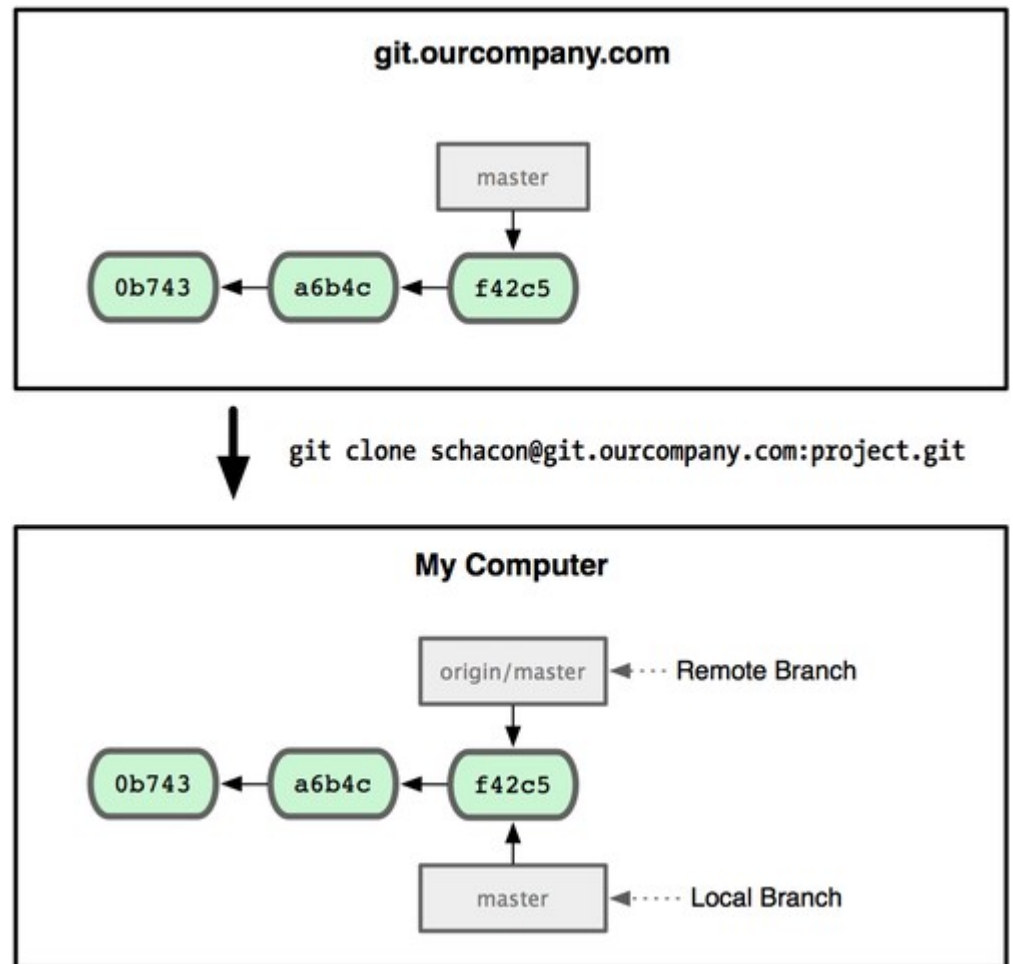
分支操作

- 拣选 cherry-pick



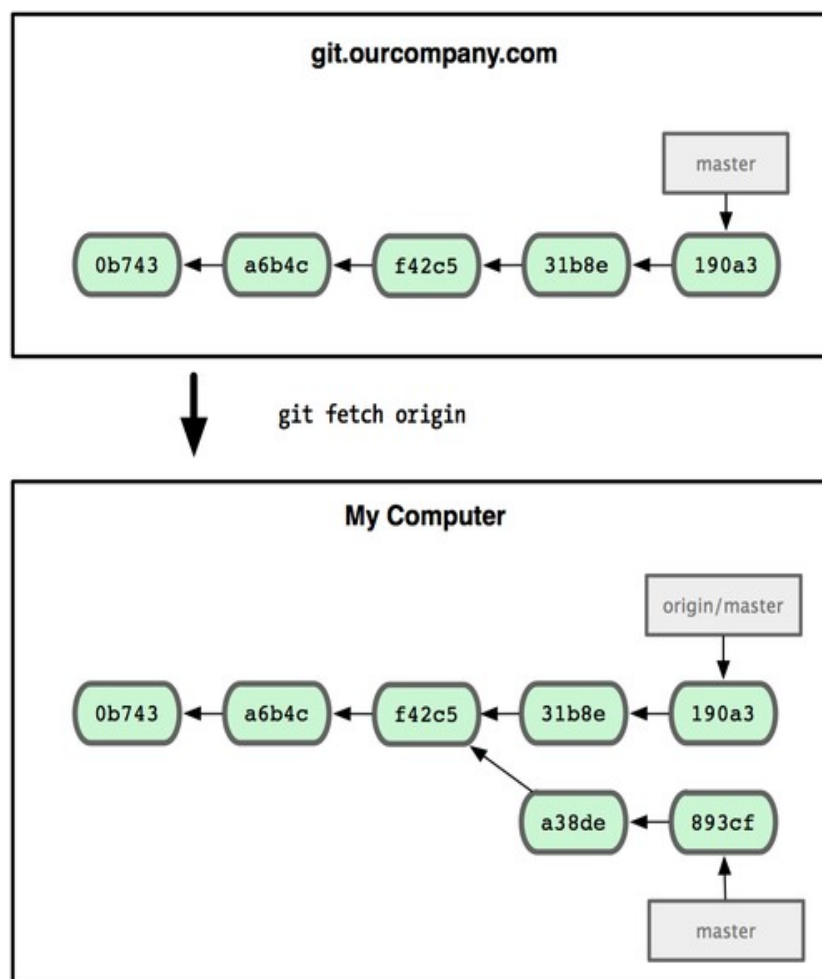
远程分支

- 本地分支跟踪远程分支
- 网络交互时才会更新



远程分支

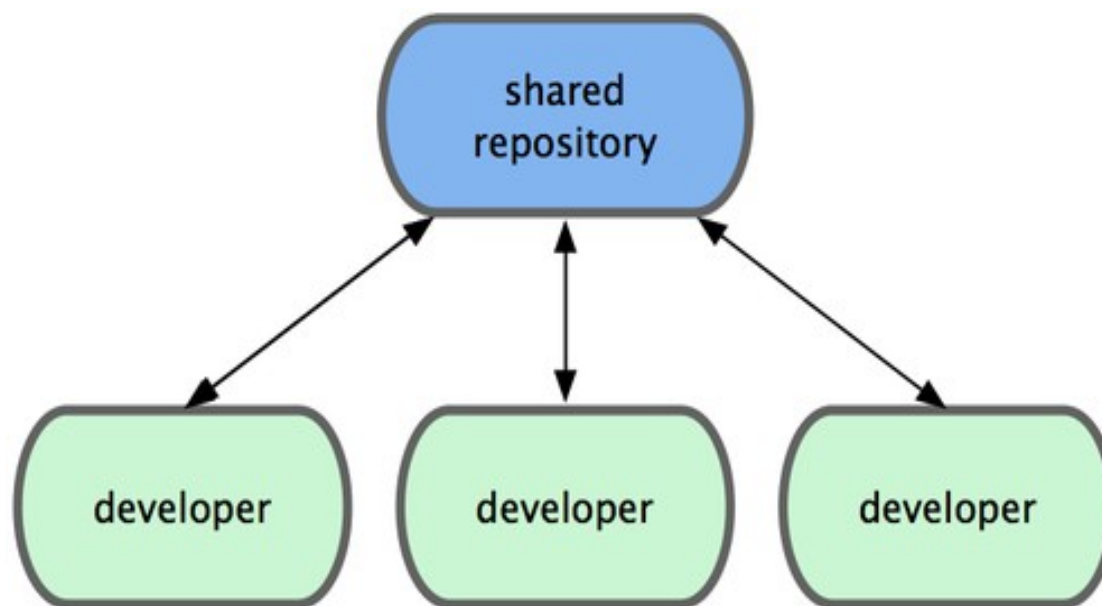
- Fetch
- Pull: fetch + merge
- push



Fetch

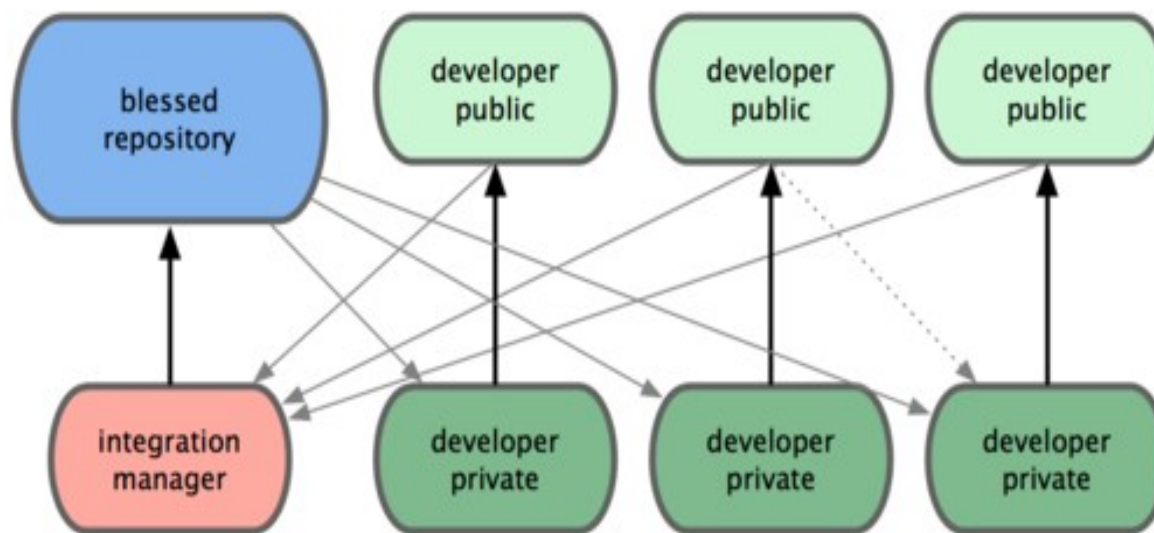
工作模式

- 集中式工作流



工作模式

- 集成管理员 workflow



项目维护者可以推送数据到公共仓库 **blessed repository**。

贡献者克隆此仓库，修订或编写新代码。

贡献者推送数据到自己的公共仓库 **developer public**。

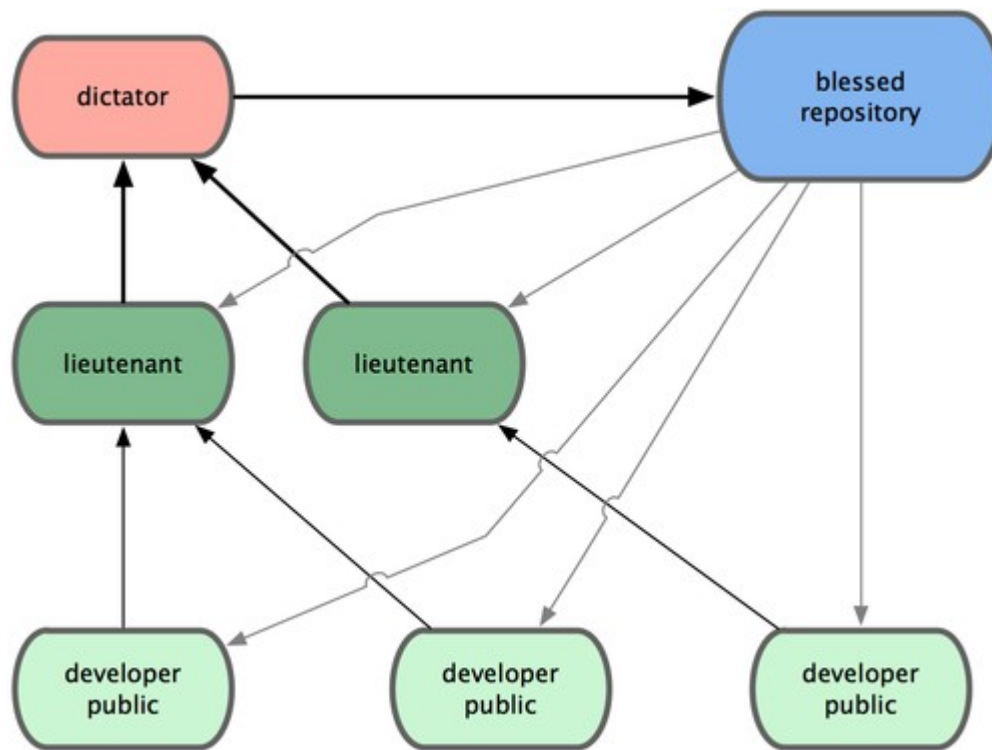
贡献者给维护者发送邮件，请求拉取自己的最新修订。

维护者在本地 **integration manger** 仓库中，将贡献者的仓库加为远程仓库，合并更新并做测试。

维护者将合并后的更新推送到主仓库 **blessed repository**。

工作模式

- 司令官与副官 workflow



一般的开发者在自己的特性分支上工作，并不定期地根据主干分支（**dictator** 上的 **master**）衍合。

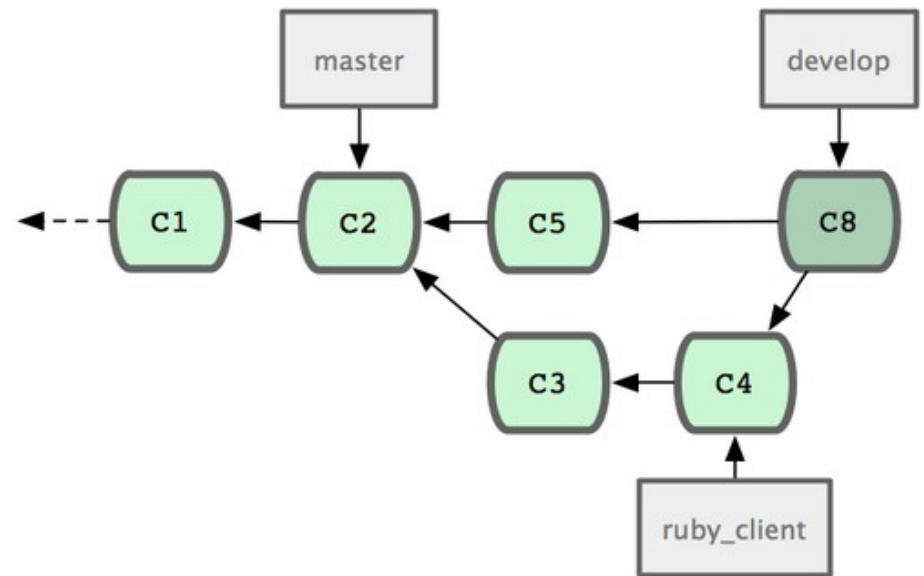
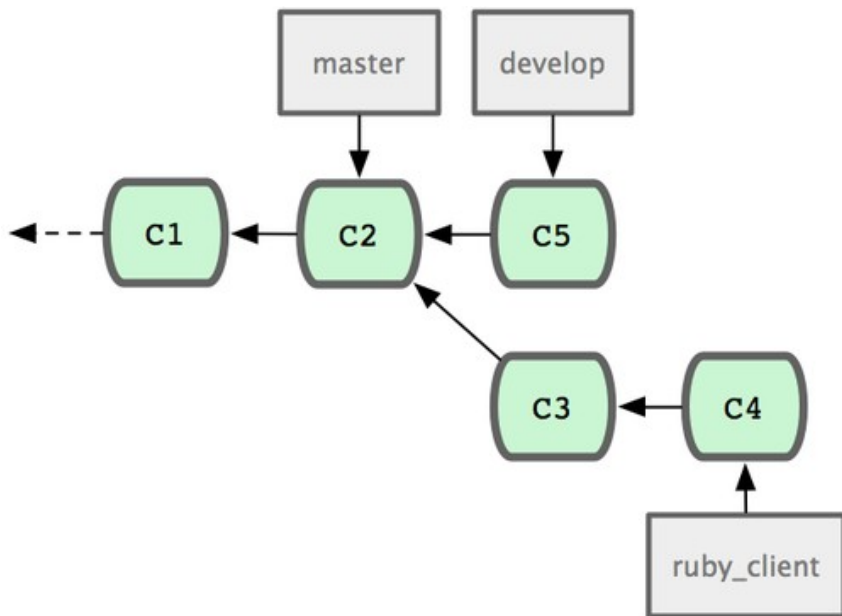
副官（**lieutenant**）将普通开发者的特性分支合并到自己的 **master** 分支中。

司令官（**dictator**）将所有副官的 **master** 分支并入自己的 **master** 分支。

司令官（**dictator**）将集成后的 **master** 分支推送到共享仓库 **blessed repository** 中，以便所有其他开发者以此为基础进行衍合。

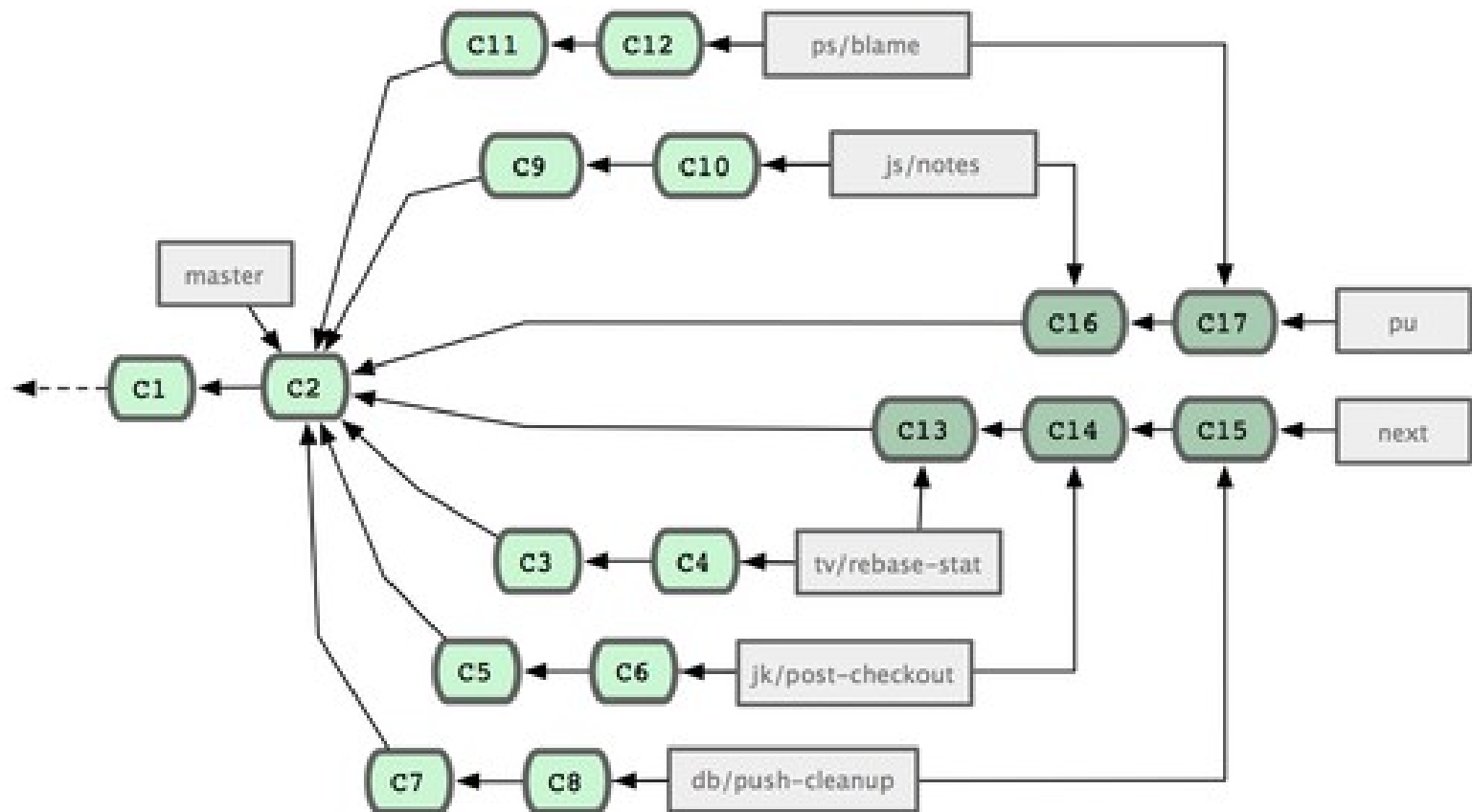
工作模式

- 小型团队 分支



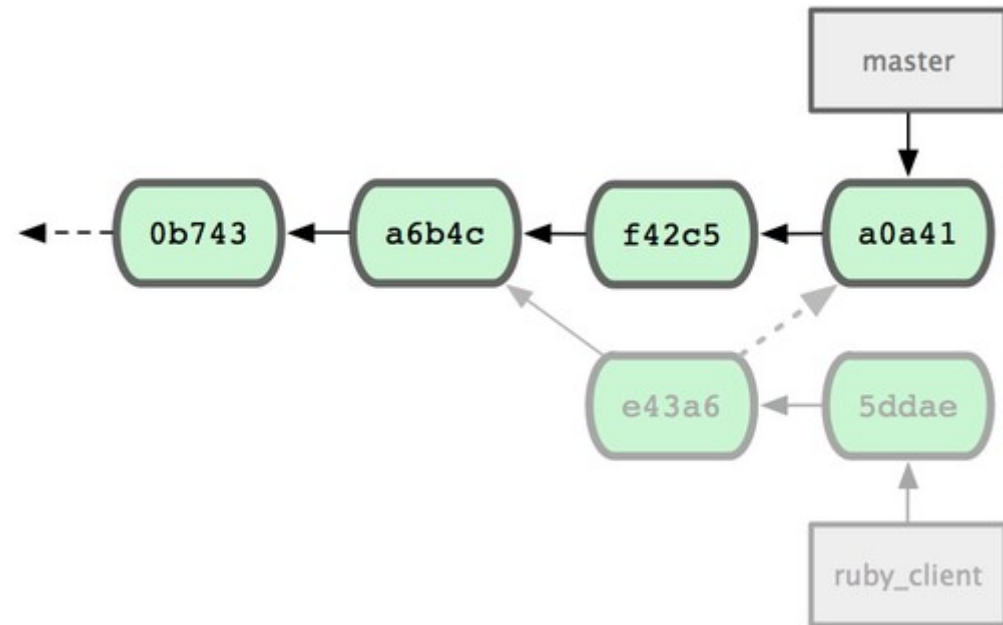
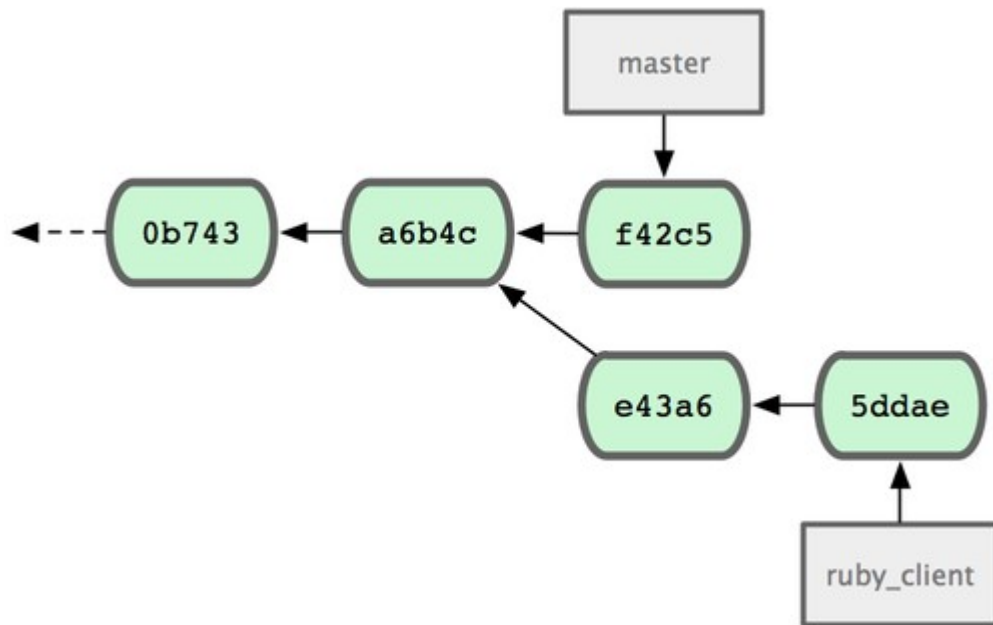
工作模式

- 大型团队 分支



工作模式

- cherry-pick/rebase



常用工具

- 储藏 :**stash**
- 打标签 :**tag**
- 改变最近一次提交 :**amend**
- 修改多个提交 :**rebase -i** 压制 / 重排 / 拆分
- **Filter-branch**: 批量修改提交