

Git 簡介

Vincent Liu 2014/12/30

Agenda

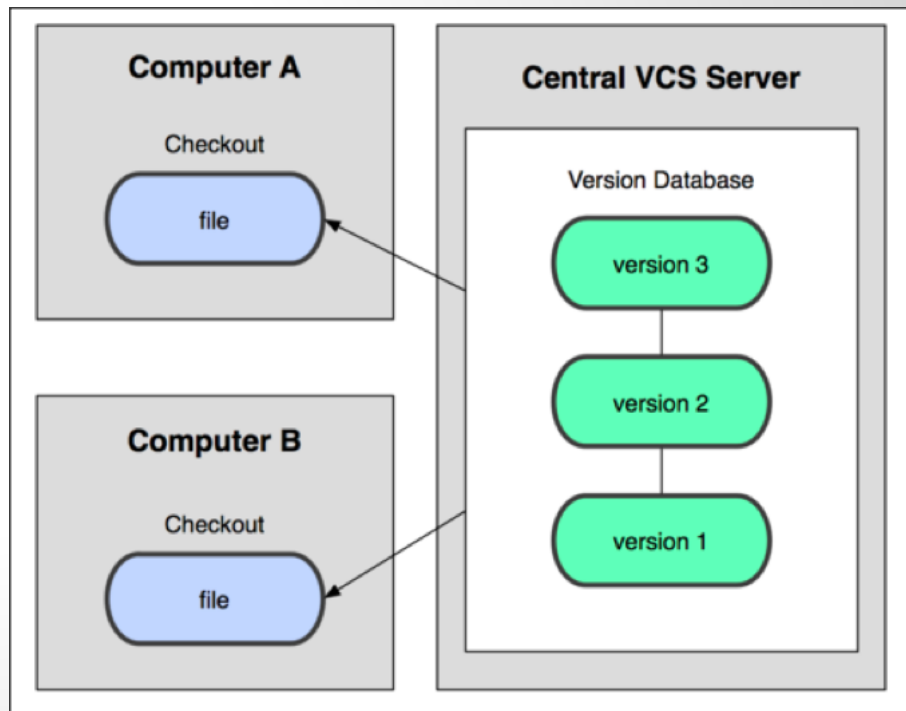
- 版本控制基本概念
- 版本控制系統類型:集中式與分散式
- Git Basics
- Git 常用指令
- Git branch
- 使用 branch 開發

為何需要版本控制系統

- 文件有不同版本需保存。
 - (可以用不同檔名或目錄名來保存)
- 文件需要分享。
 - (可以用網路分享)
- 文件存取要有權限。
 - (可以用系統權限設定)
- 需要知道誰修改了哪份文件的哪個部分。
- 文件需要共同編修。

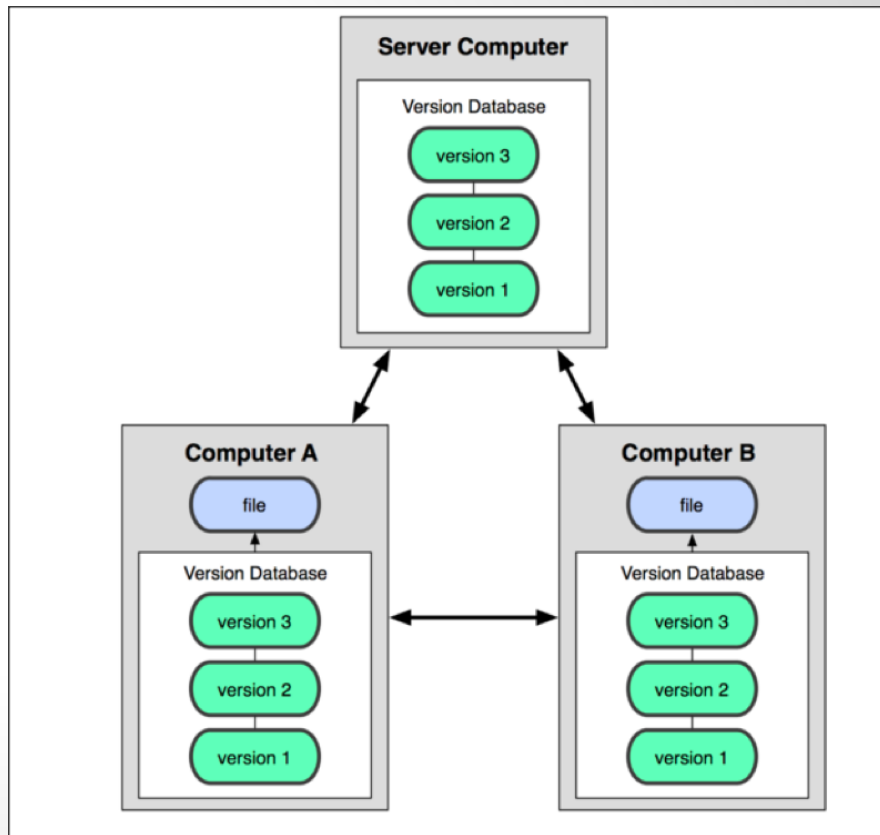
集中式版本管理系統

- 伺服器端才擁有完整資料。
- 幾乎所有動作都在伺服器端完成。
- 伺服器資料毀損，所有歷程會消失。
- 常見的 SVN 及 CVS 是代表的系統。



分散式版本管理系統

- 每個 Client 都有完整的資料。
- 幾乎所有的工作都可以在本地端完成。
- 適合多人異地平行開發。
- GIT 為代表系統。

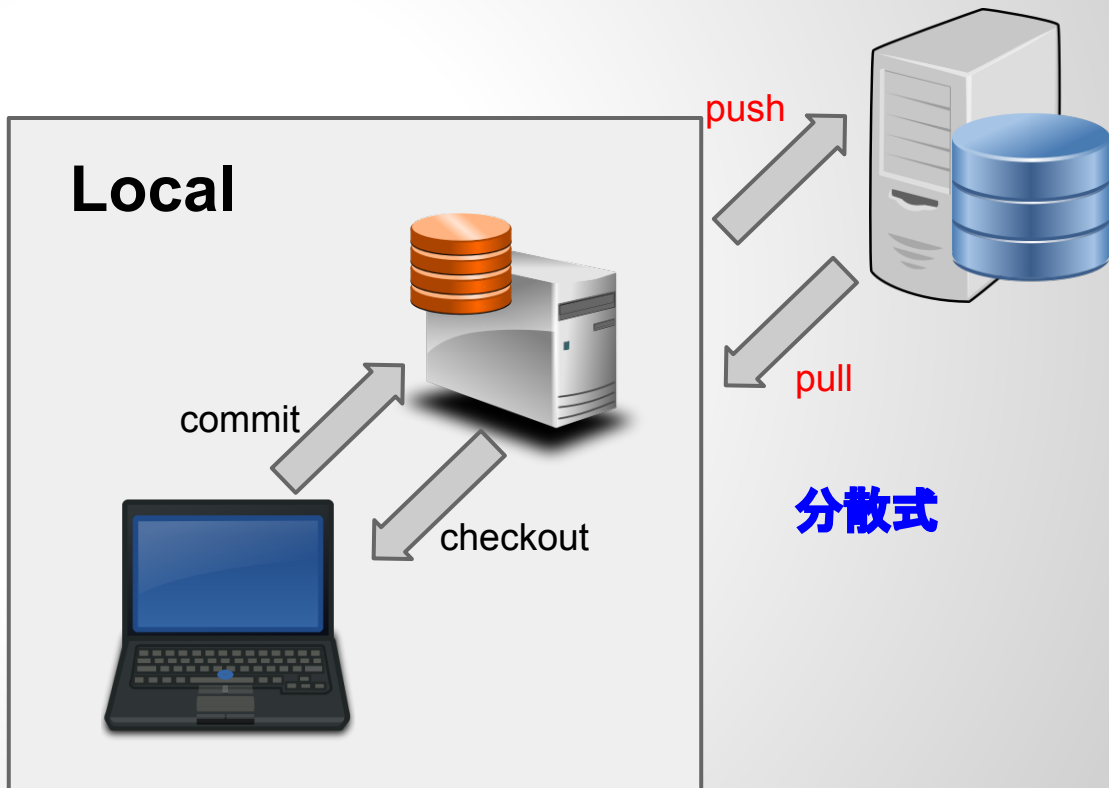


集中式與分散式圖示比較

集中式

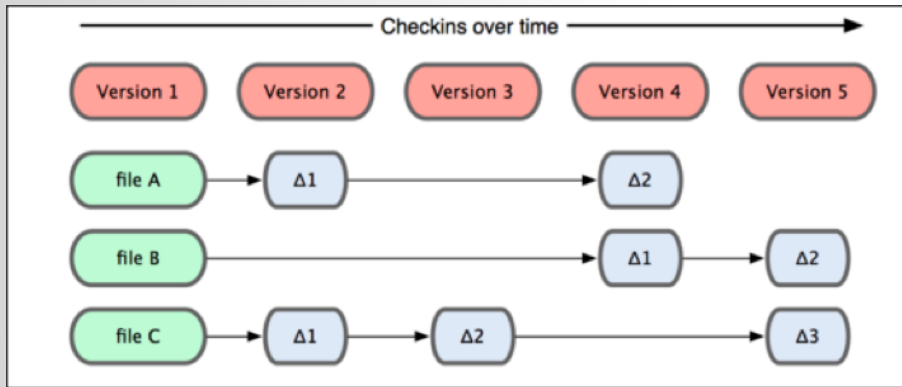


Local

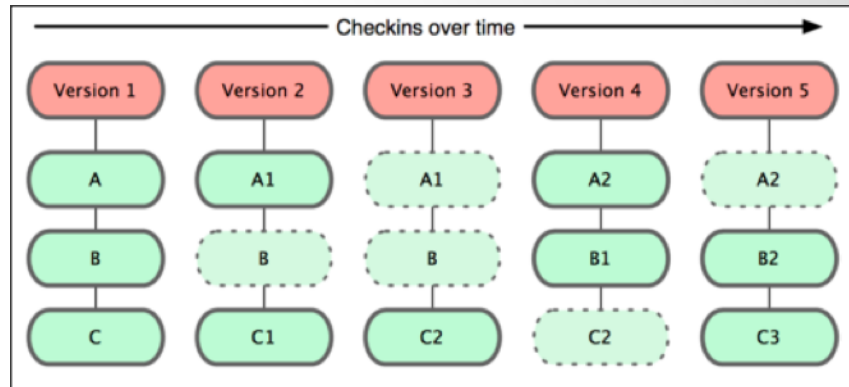


分散式

GIT Basics

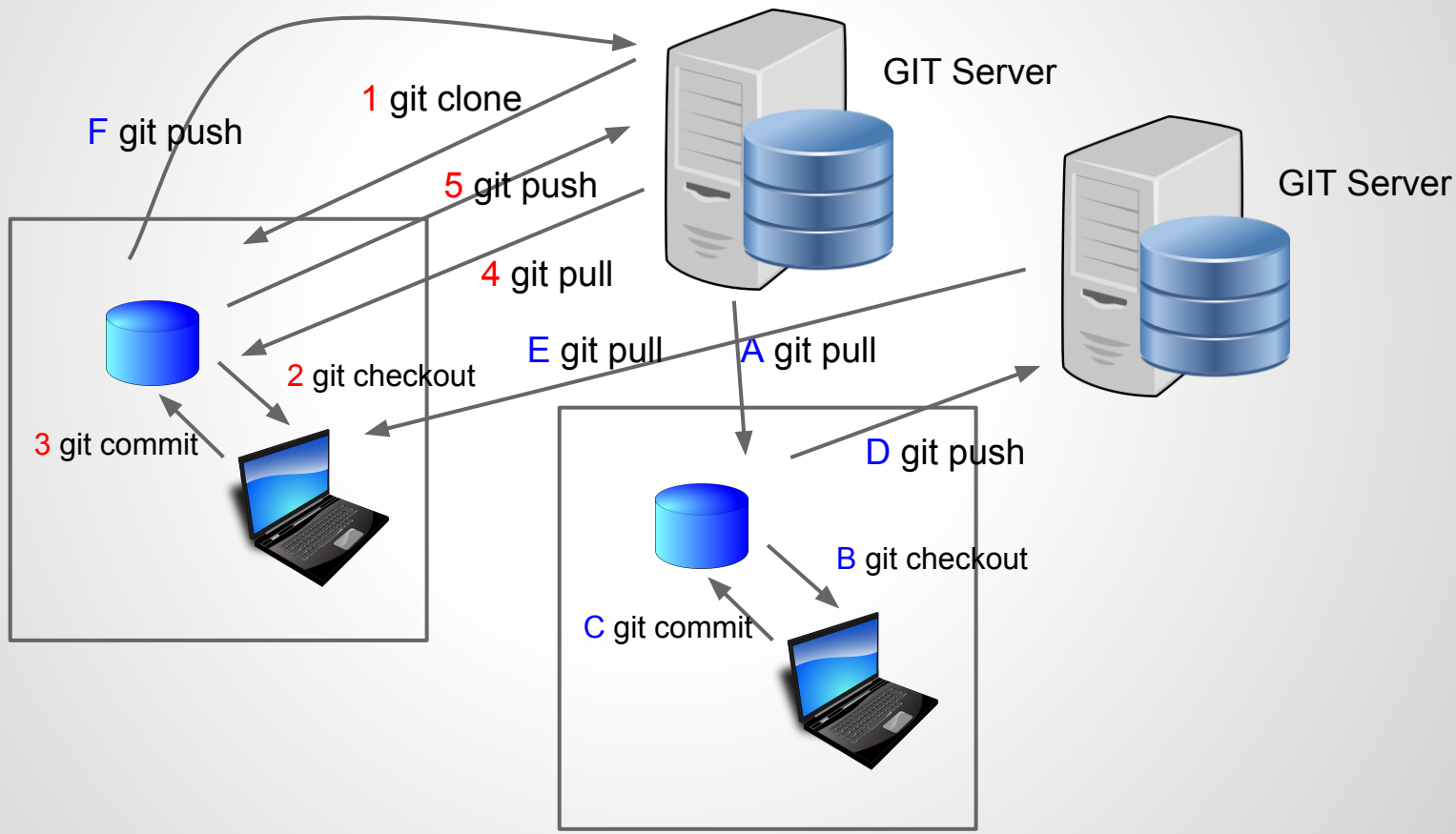


傳統版本控制系統記錄文件每次提交差異。



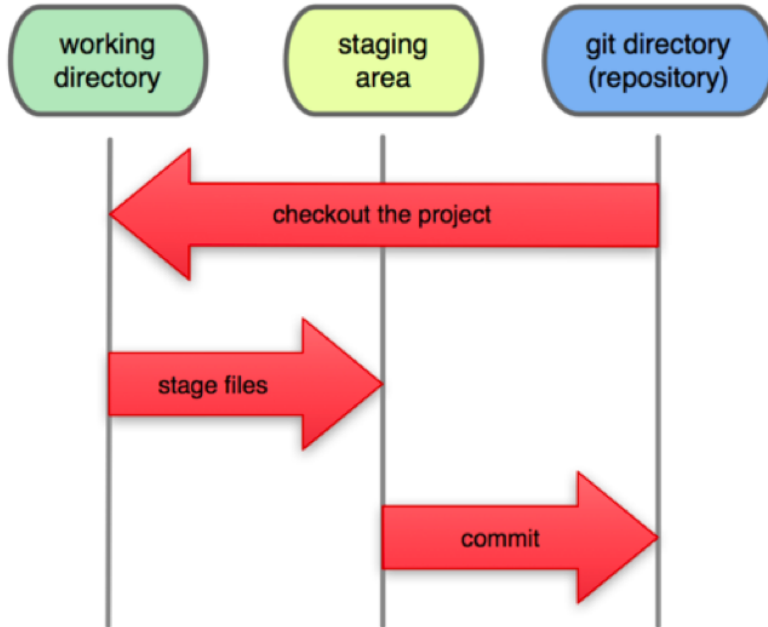
GIT 儲存每次提交文件的快照。

開發生命週期

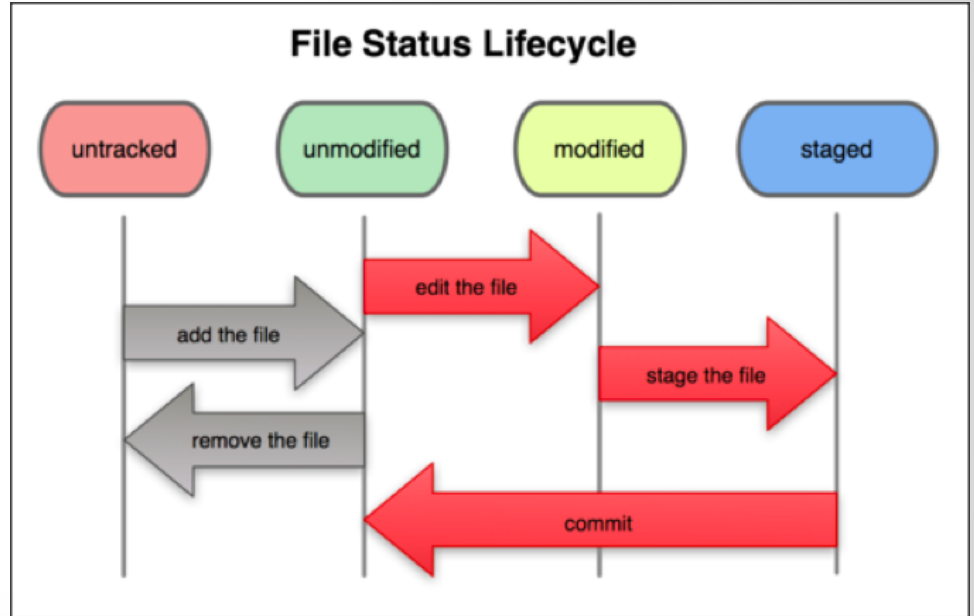


GIT Basics

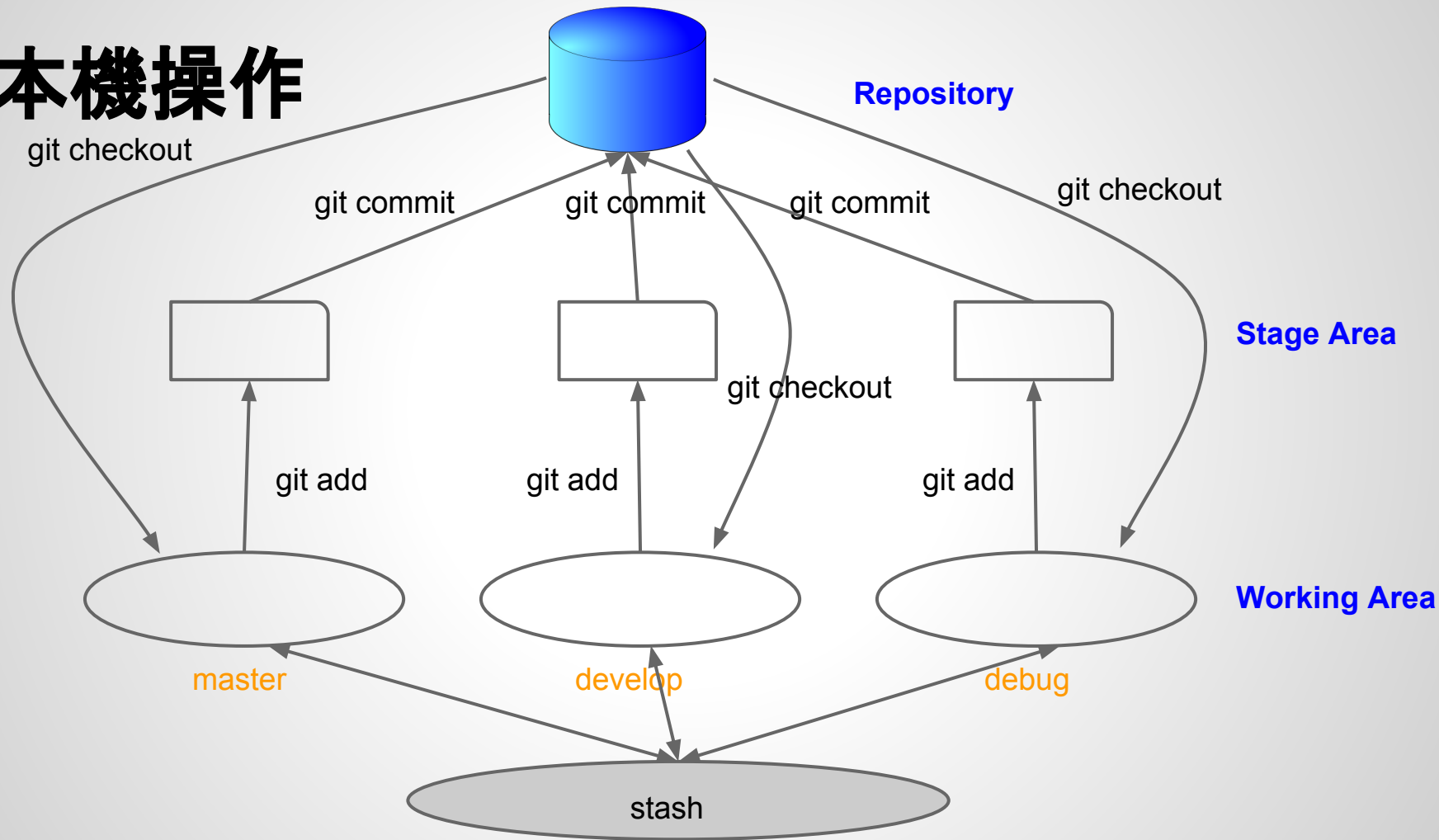
Local Operations



File Status Lifecycle



本機操作



GIT Stage

```
MacBook-Air:git-lessons wkliu$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   README.md
#       new file:   progit-en.252.pdf
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   "GIT \347\260\241\344\273\213.pdf"
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       progit.zh.pdf
MacBook-Air:git-lessons wkliu$
```

GIT 建立倉庫(Repository)

- git init 建立空的倉庫
- git status 檢視目前狀態

```
wkliu@ubuntu:~$ git init myrepo
Initialized empty Git repository in /home/wkliu/myrepo/.git/
wkliu@ubuntu:~$ cd myrepo
wkliu@ubuntu:~/myrepo$ git status
On branch master

Initial commit

nothing to commit (create/copy files and use "git add" to track)
```

GIT 複製已存在倉庫

- git clone

```
wkliu@ubuntu:~$ git clone https://github.com/wkliu/testrepo.git
Cloning into 'testrepo'...
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 0), reused 6 (delta 0)
Unpacking objects: 100% (6/6), done.
Checking connectivity... done.
```

GIT 常用指令

- git status 查看目前狀態

```
wkliu@ubuntu:~/CAG2$ git status
```

```
On branch fixbugs#6
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
modified:   apps/server/host/models/storage.js
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)
```

```
(use "git checkout -- <file>..." to discard changes in working directory)
```

```
modified:   apps/server/host/models/eventDenormalizer.js
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
apps/server/.DS_Store  
apps/server/._.DS_Store  
apps/server/host/.DS_Store
```

GIT 常用指令

- git log 查看已 committed 的歷史記錄。

```
commit 5072ff6f287f047cd806e4a9907eb47404e0023f
Author: SYC <sync@sync-A0D260.(none)>
Date: Tue Nov 18 16:20:31 2014 +0800
```

新增錯誤訊息多語系 Index

```
commit 048d29b5113bf17f22ae7c9817ffdbc212b721bb
Author: Timothy Huang <elsely@gmail.com>
Date: Tue Nov 18 12:11:55 2014 +0800
```

cloudsearch with environment CAGENV

cloudsearch upload with cagenv and delete with cagenv finished.
put the server-side formatcloudsearch datetime code from
bower_components utility to server utility

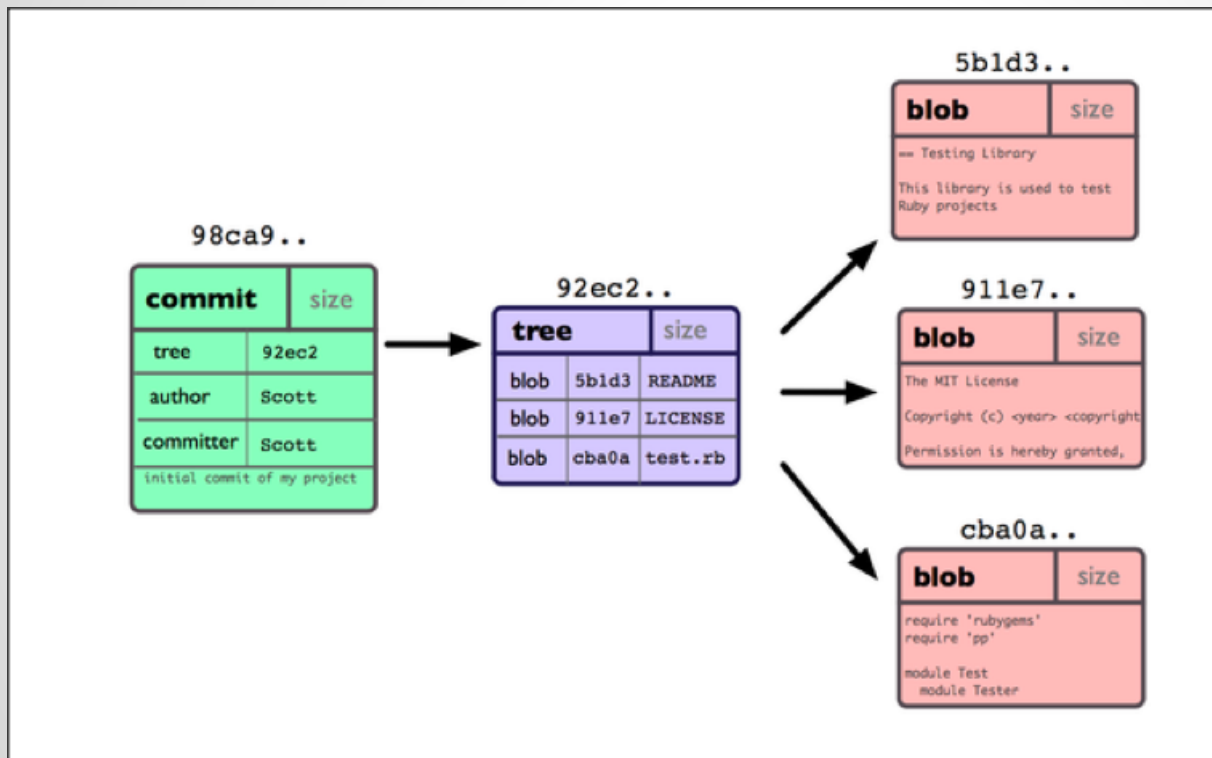
```
commit cb484455aa7b825145bc441b60a8286ffe81bc12
Author: wkliu <wkliu228@gmail.com>
Date: Tue Nov 18 04:21:08 2014 +0800
```

Solve the issue which clicked the contact us link on Service page, the

GIT Commit

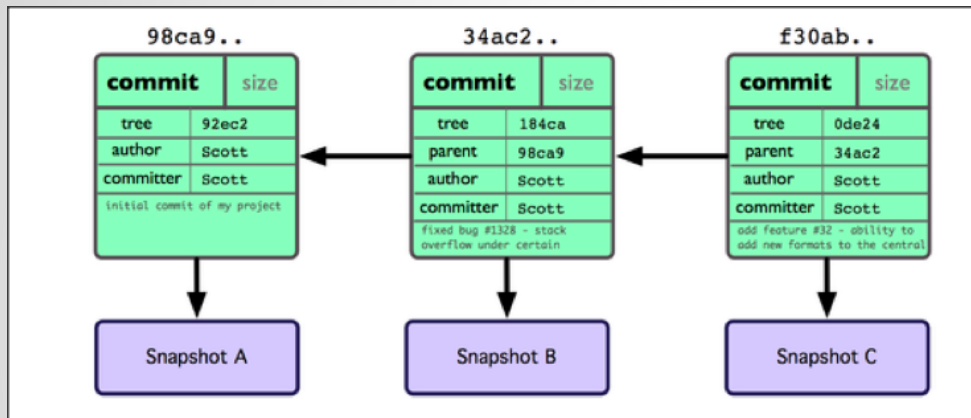
```
MacBook-Air:git-lessons wkliu$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   "GIT \347\260\241\344\273\213.pdf"
#
no changes added to commit (use "git add" and/or "git commit -a")
MacBook-Air:git-lessons wkliu$ git commit -a -m "Modified the slide to add Distributed workflows section"
[master 33249f5] Modified the slide to add Distributed workflows section
1 file changed, 0 insertions(+), 0 deletions(-)
```


GIT 分支(Branch)



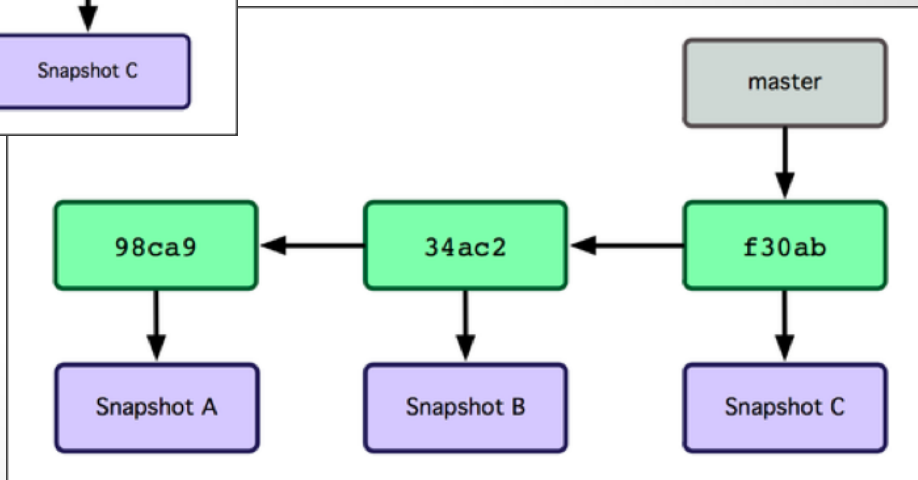
GIT commit 的資料結構示意圖

GIT Branch



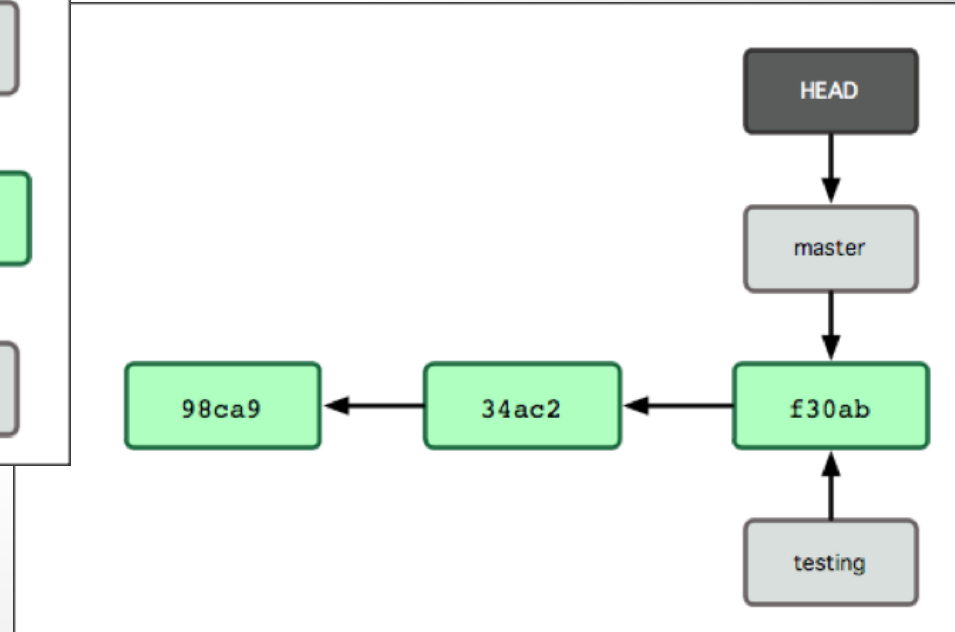
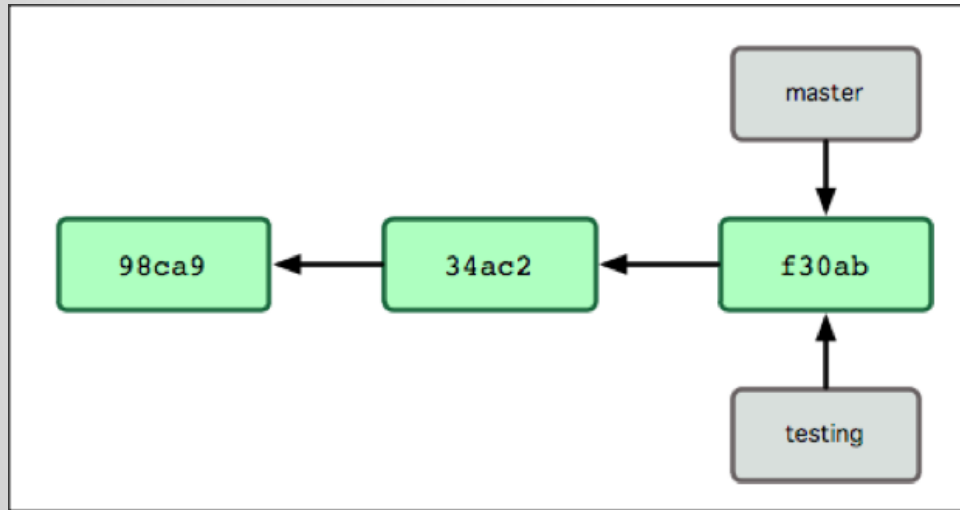
多次 Commit 後的圖示

預設分支 **master**



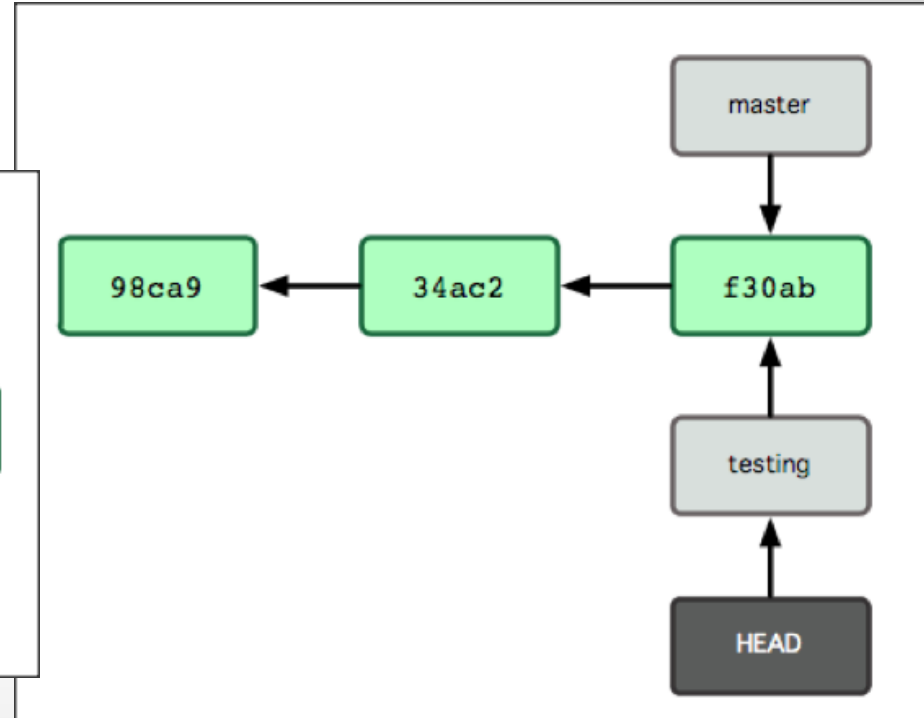
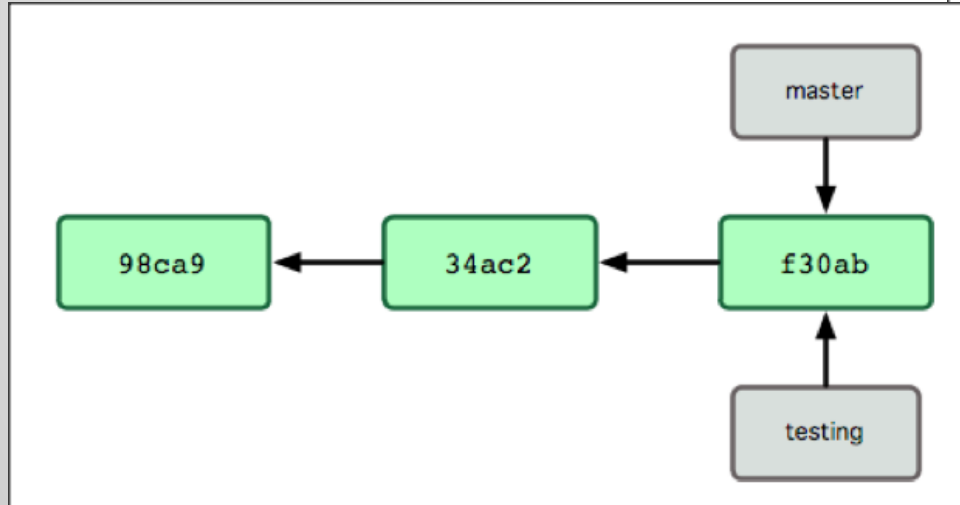
GIT Branch

- git branch testing



GIT Branch

- git checkout -b testing
- git branch testing and git checkout testing



GIT 分支(Branch)

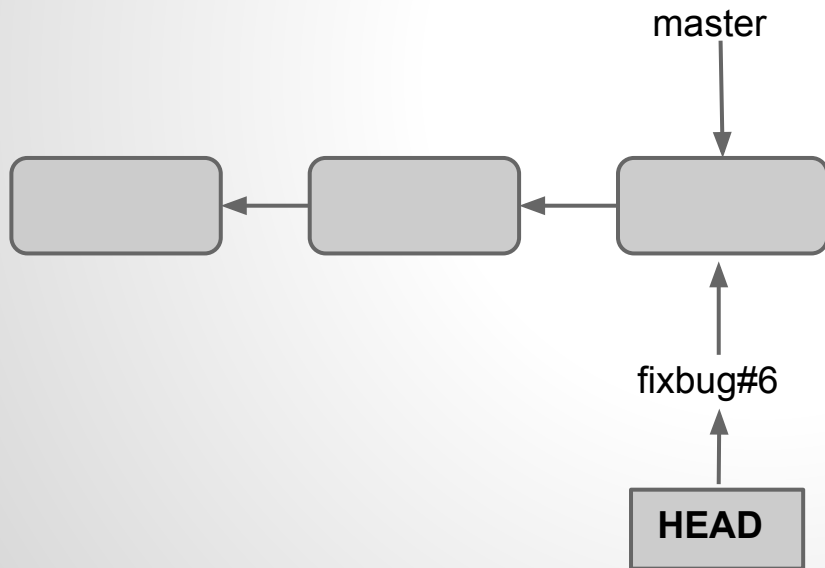
- git branch 查看目前所在分支與目前擁有分支。

```
wkliu@ubuntu:~/CAG2$ git branch
* fixbugs#6
  i18n
  master
```

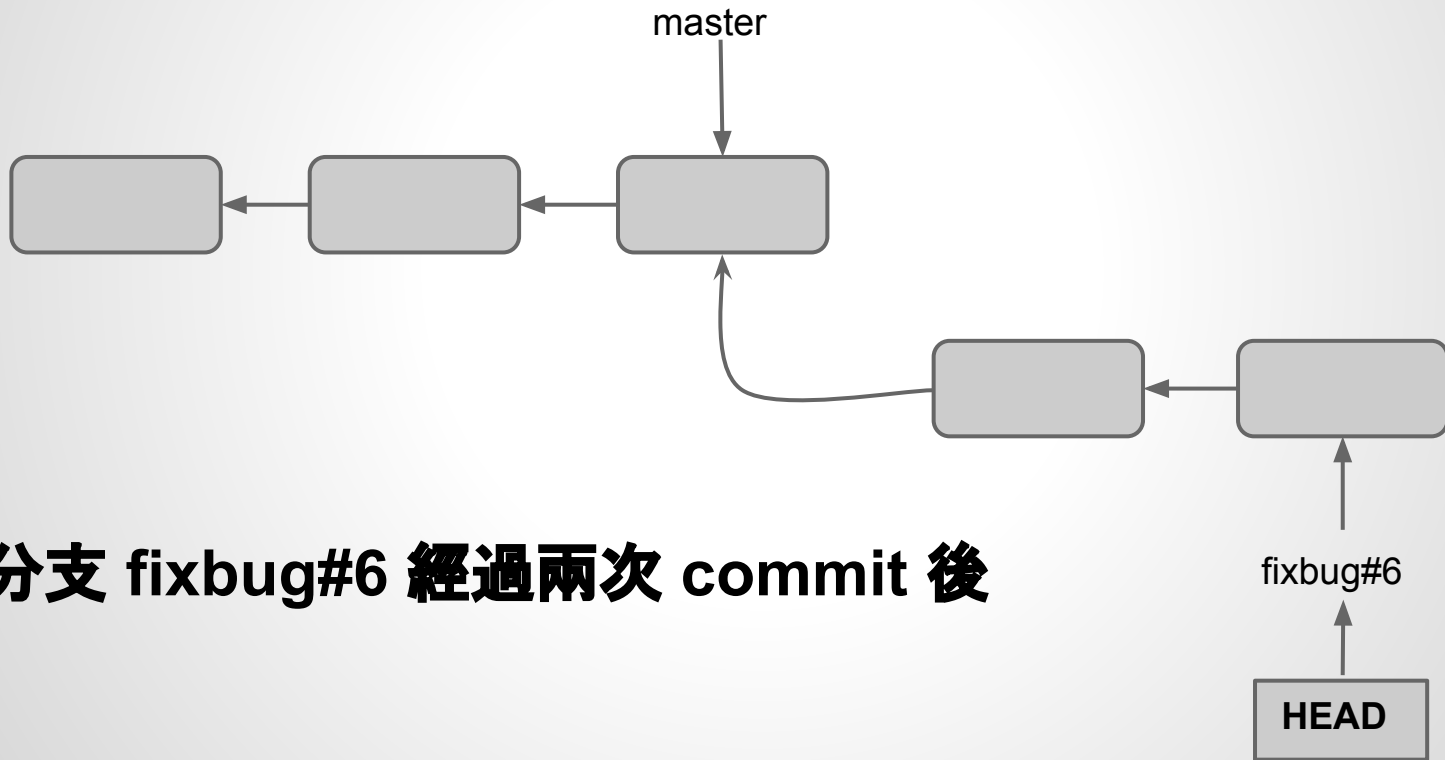
- git branch -d branch_name 刪除分支。
 - 不能刪除目前所在分支。
 - 不能刪除未合併的分支。(可改用 -D 強制刪除)

建議開發時使用 branch

- `git checkout -b fixbug#6`

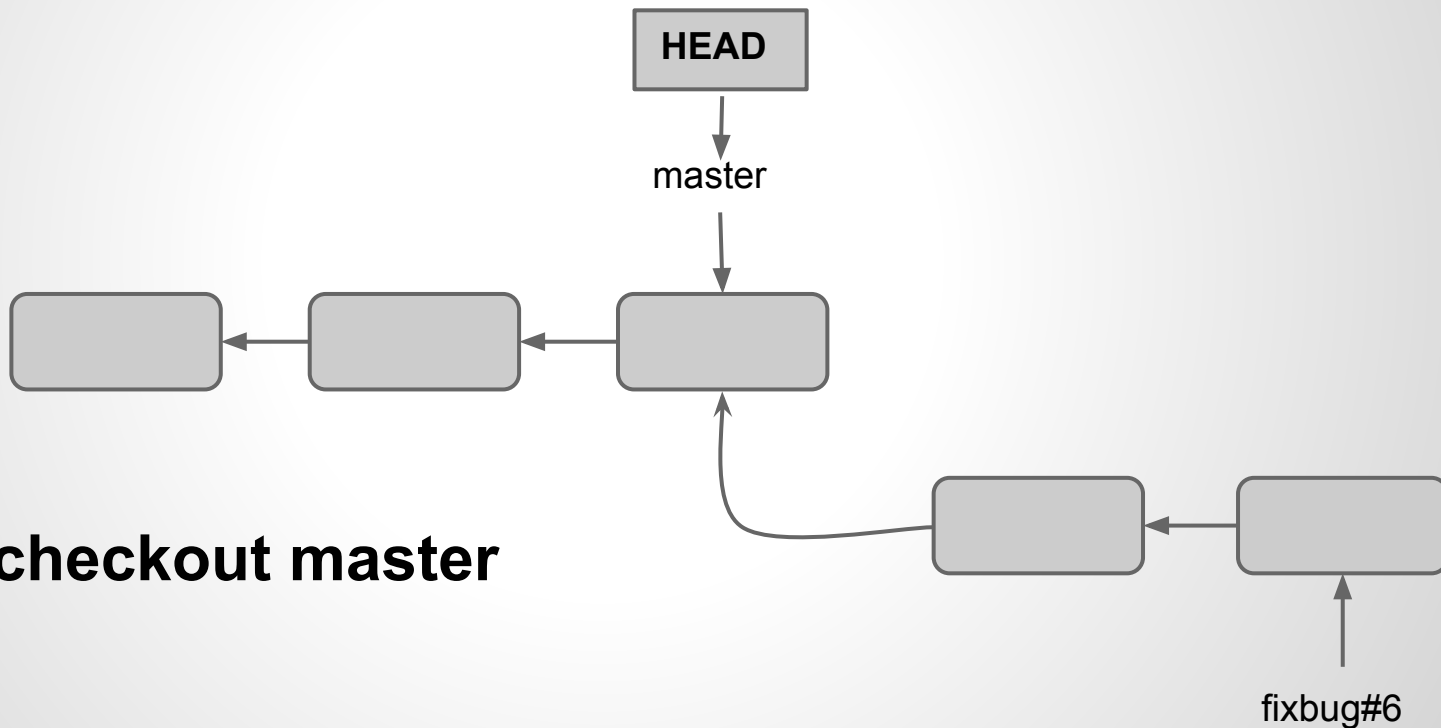


使用 branch 開發



在分支 `fixbug#6` 經過兩次 `commit` 後

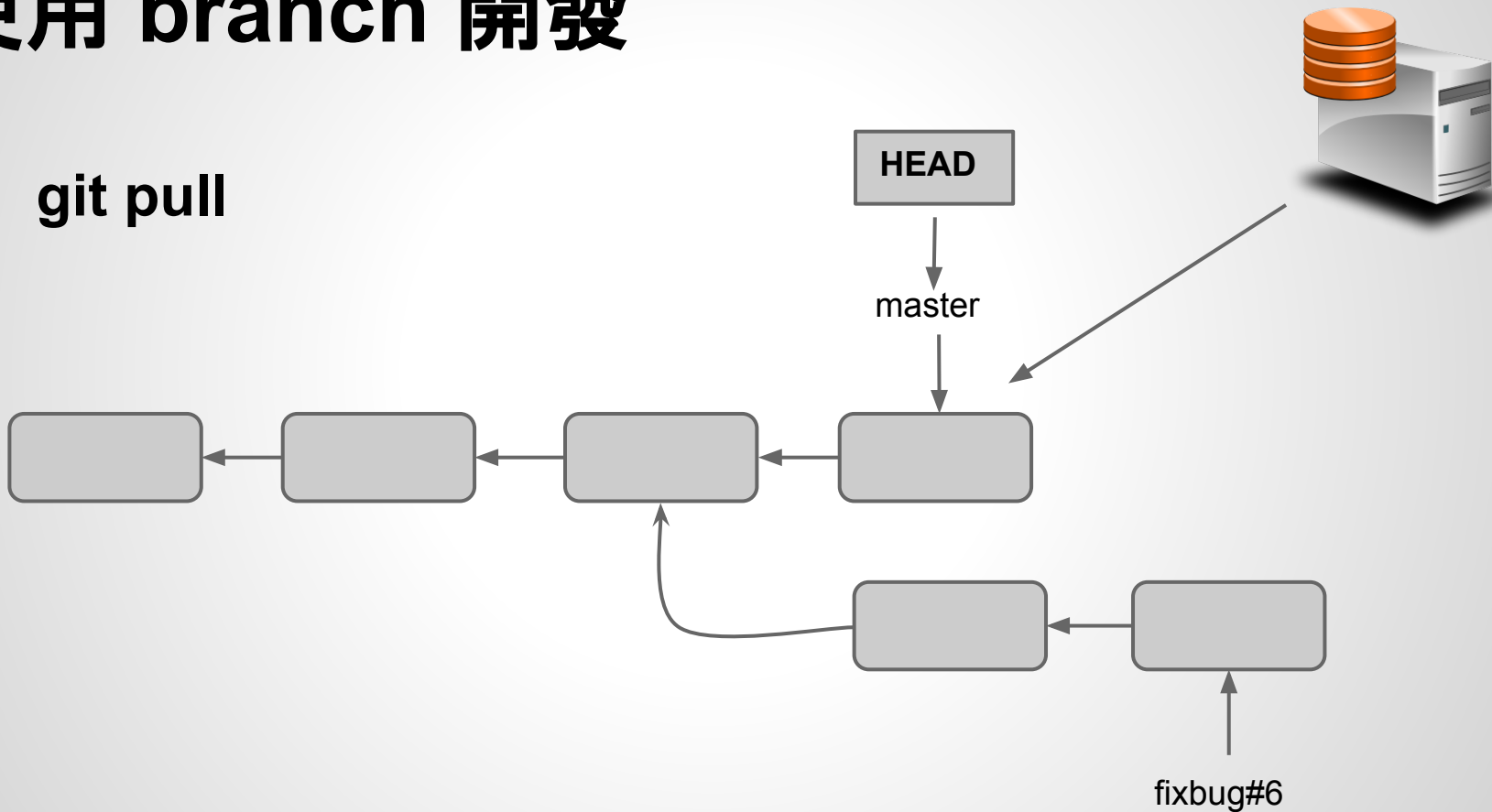
使用 branch 開發



- **git checkout master**

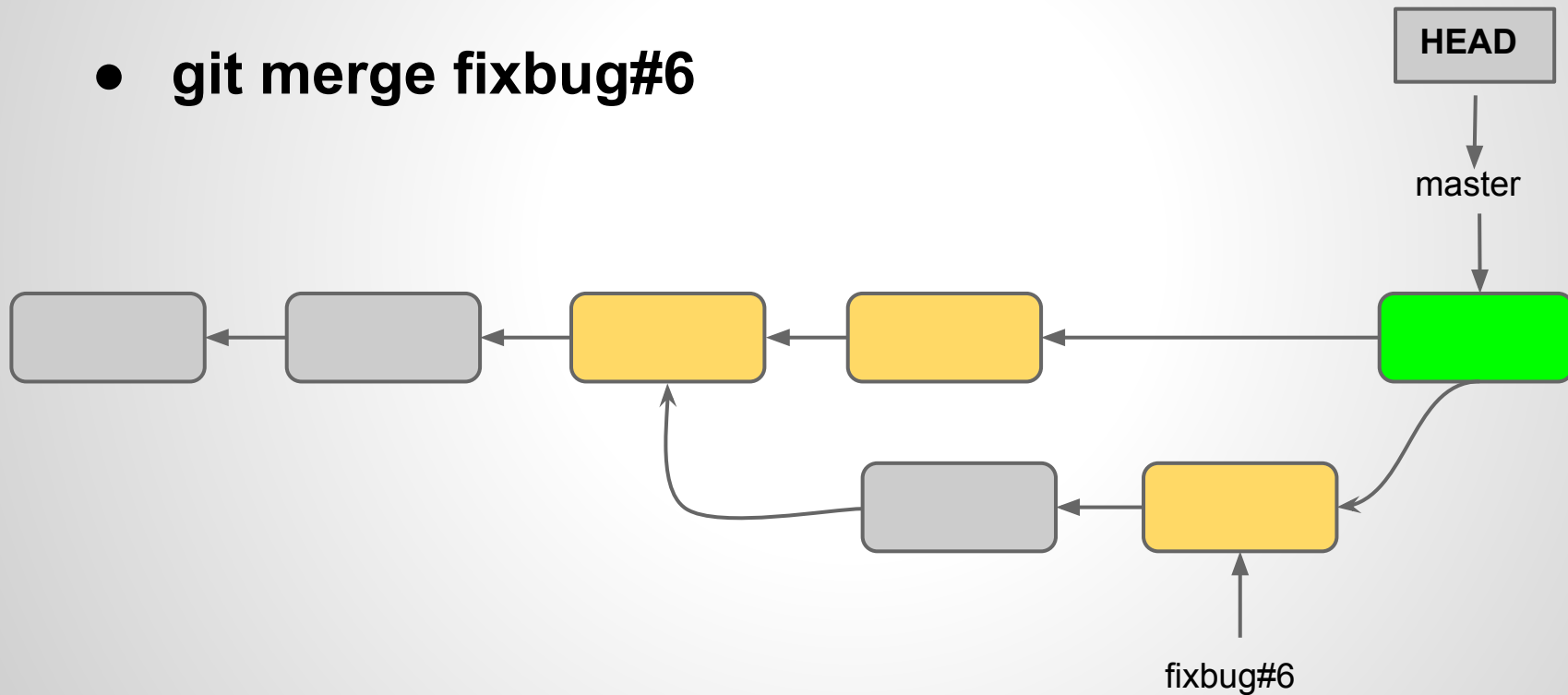
使用 branch 開發

- git pull



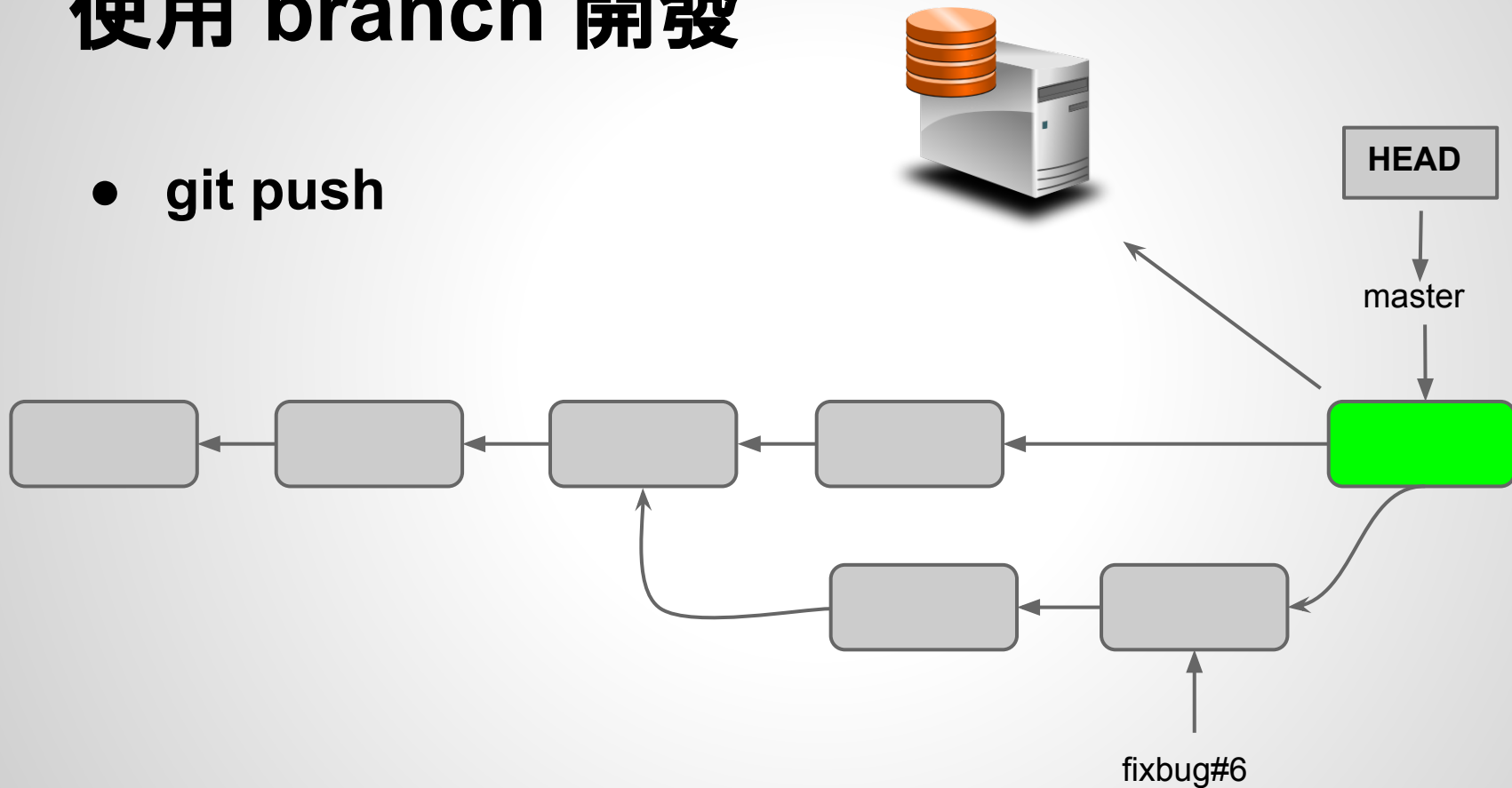
使用 branch 開發

- **git merge fixbug#6**



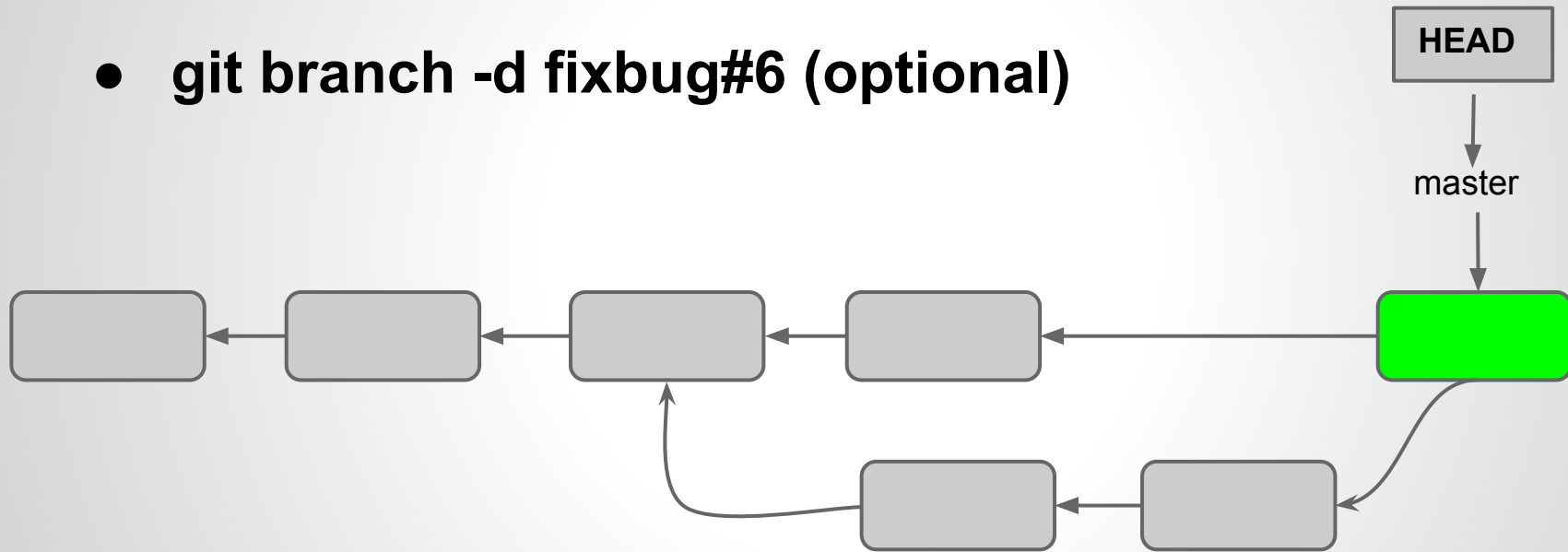
使用 branch 開發

- git push

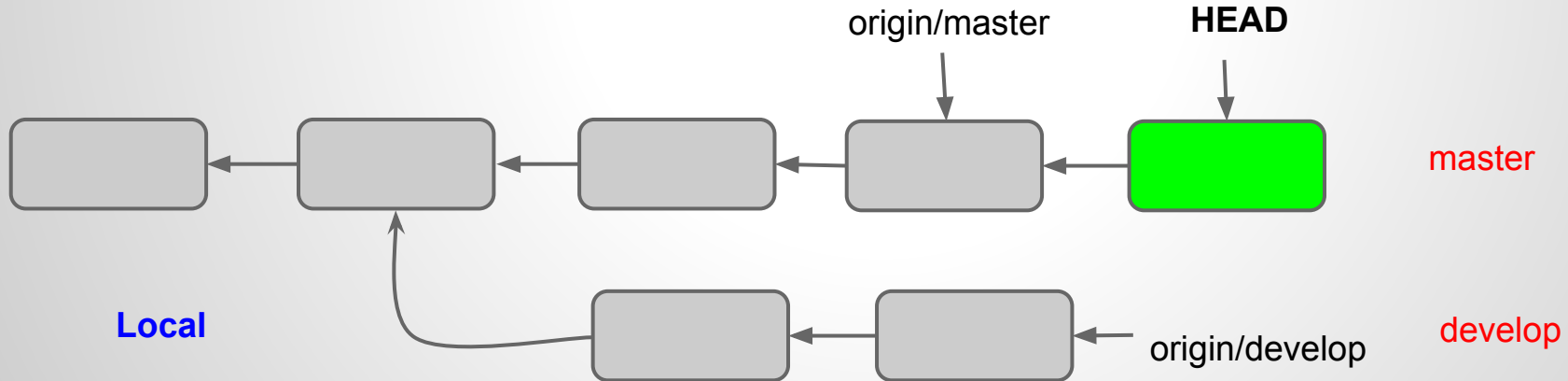
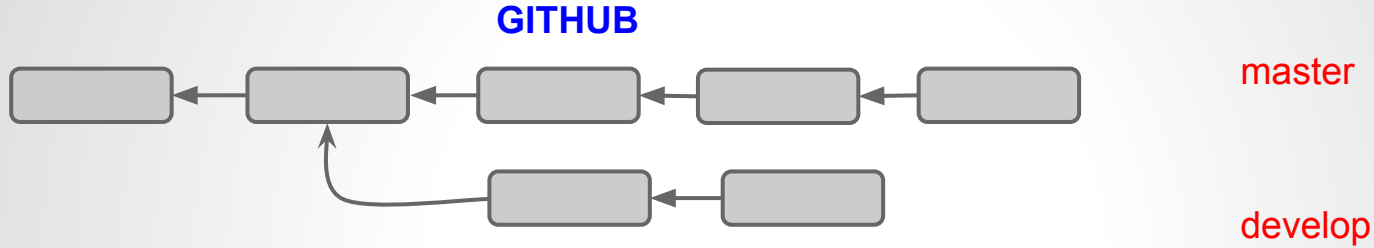


使用 branch 開發

- `git branch -d fixbug#6` (optional)



Remote Branches



Create Remote Branch

```
MacBook-Air:git-lessons wkliu$ git branch
for104
* master
MacBook-Air:git-lessons wkliu$ git checkout for104
Switched to branch 'for104'
MacBook-Air:git-lessons wkliu$ ls
GIT 简介.pdf  README.md
MacBook-Air:git-lessons wkliu$ git push vincent for104:git104
Username for 'https://github.com': vincentliu228
Password for 'https://vincentliu228@github.com':
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/vincentliu228/lessons.git
* [new branch]      for104 -> git104
MacBook-Air:git-lessons wkliu$
```

The diagram illustrates the relationship between the local branch 'for104' and the remote branch 'git104'. A red arrow labeled 'Repo' points to the 'for104' branch in the terminal output. Another red arrow labeled 'local branch' points to the 'for104' branch in the terminal output. A third red arrow labeled 'Remote Branch' points to the 'git104' branch in the terminal output. The 'git push' command is highlighted with a red box, and the 'for104' branch is also highlighted with a red box.

Remote Branches

The screenshot displays a GitHub repository interface. At the top, a summary bar shows '4 commits', '2 branches', '0 releases', and '1 contributor'. Below this, the repository name 'lessons / +' is visible next to a 'branch: master' dropdown. A 'Switch branches/tags' modal is open, featuring a search input 'Find or create a branch...' and two tabs: 'Branches' and 'Tags'. Under the 'Branches' tab, a list of branches is shown, with 'git104' highlighted in blue and 'master' marked with a checkmark. The background shows a commit history table with columns for commit message, latest commit hash, and time ago. On the right sidebar, links for 'Issues', 'Pull Requests', 'Wiki', 'Pulse', 'Graphs', and 'Settings' are listed.

4 commits 2 branches 0 releases 1 contributor

branch: master lessons / +

Switch branches/tags

Find or create a branch...

Branches Tags

git104

✓ master

Commit Message	Latest Commit	Time Ago
ed prepared it...	2bfc92a19e	20 hours ago
104 colleague.		
duce the class and related prepared it...		4 hours ago

Issues 0

Pull Requests 0

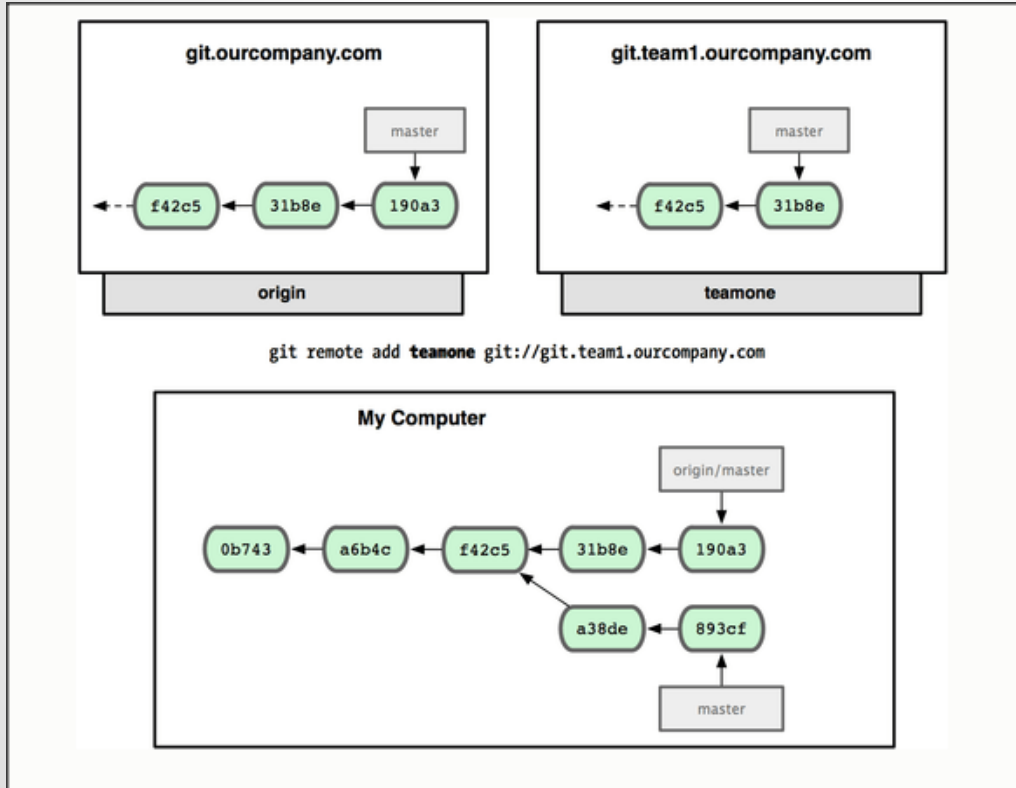
Wiki

Pulse

Graphs

Settings

Remote Branches



git remote

```
MacBook-Air:git-lessons wkliu$ git remote
```

```
origin  
vincent
```

```
MacBook-Air:git-lessons wkliu$ git remote -v
```

```
origin https://github.com/wkliu/git-lessons.git (fetch)  
origin https://github.com/wkliu/git-lessons.git (push)  
vincent https://github.com/vincentliu228/lessons.git (fetch)  
vincent https://github.com/vincentliu228/lessons.git (push)
```

```
MacBook-Air:git-lessons wkliu$
```

```
MacBook-Air:git-lessons wkliu$ git remote remove vincent
```

```
MacBook-Air:git-lessons wkliu$ git remote
```

```
origin
```

```
MacBook-Air:git-lessons wkliu$ git remote add vincent https://github.com/vincentliu228/lessons.git
```

```
MacBook-Air:git-lessons wkliu$ git remote -v
```

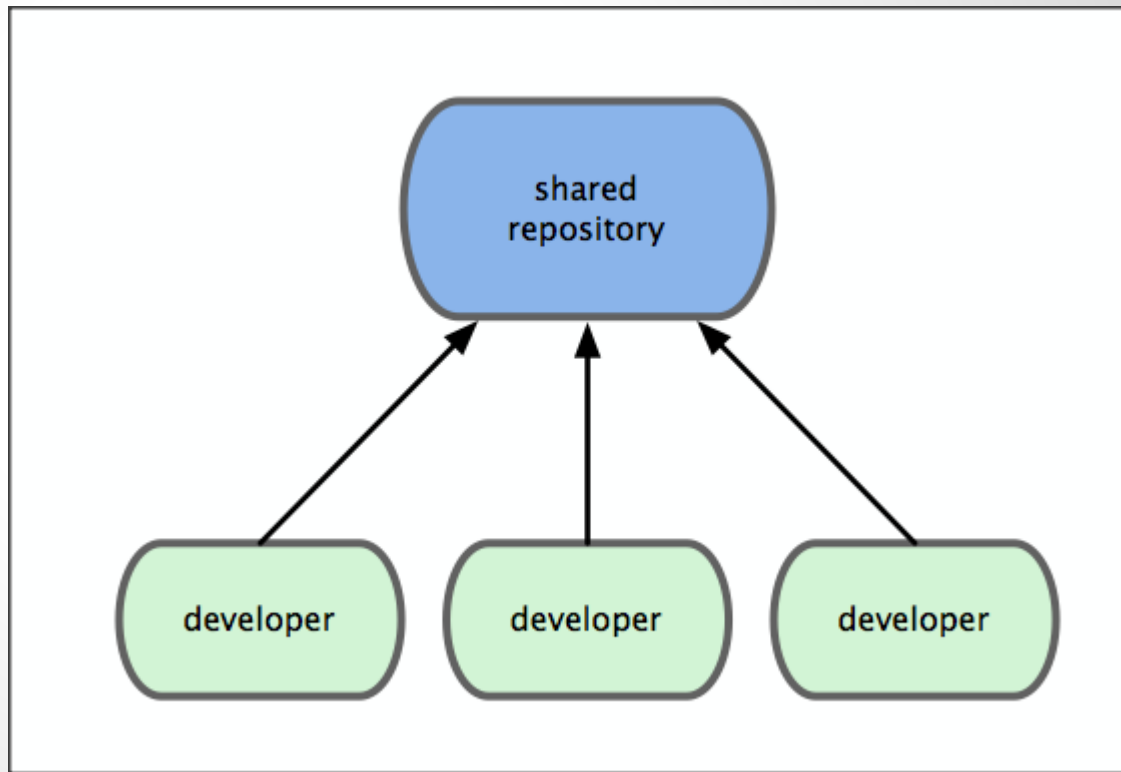
```
origin https://github.com/wkliu/git-lessons.git (fetch)  
origin https://github.com/wkliu/git-lessons.git (push)  
vincent https://github.com/vincentliu228/lessons.git (fetch)  
vincent https://github.com/vincentliu228/lessons.git (push)
```

```
MacBook-Air:git-lessons wkliu$
```

協同開發的方法

1. 集中式

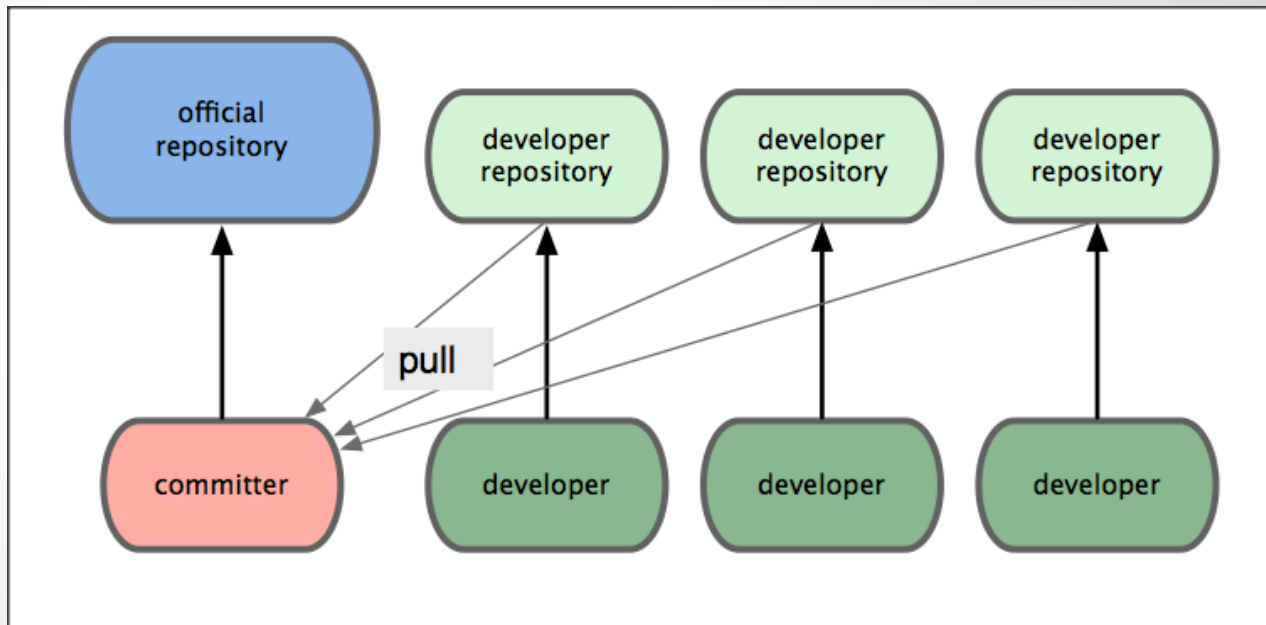
每個開發者都有 push 的權限



協同開發的方法

2. 管理者整合

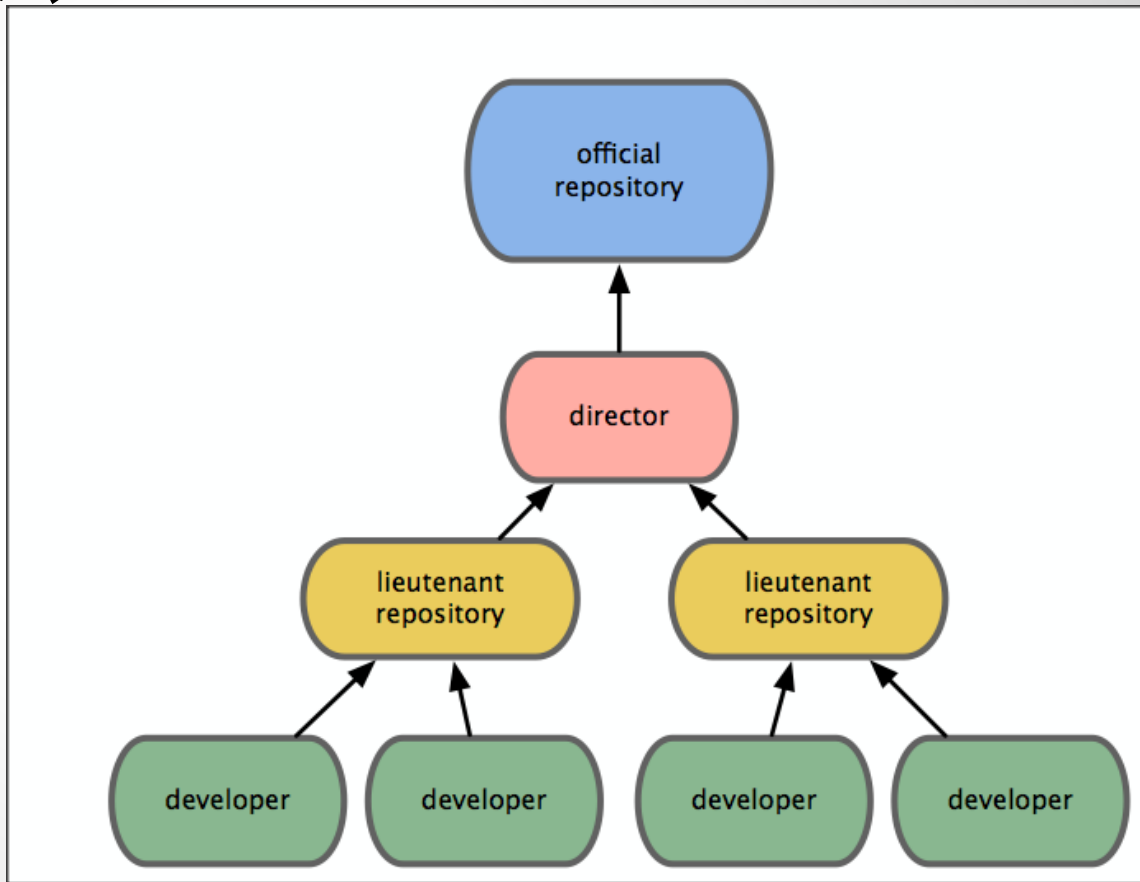
僅有少數管理者有權限 Push, 其餘開發者提出 Pull request。



協同開發的方法

3. 多層權限管控

分成不同子模組，每個模組有不同的管理者，最後由主控管理者負責 push。



GIT Flow

