

# Linux 與 虛擬化 & GIT

Brian Luan  
2015/07/21

# 關於我

- 現職：
  - 大數軟體有限公司 顧問
  - 網鈺數位科技 技術經理
- 曾任職
  - 新創 - MIGO entertainment
  - IBM/ORACLE 資料解決方案
- 大數學堂 <http://course.largitdata.com/>
- 粉絲頁 <https://www.facebook.com/largitdata>





# 輿情分析 & 輿論監控





卡提諾論壇  
CK101.com

阿宅



iBeauty  
愛漂亮

女人



DONG  
動網

男人



媽媽經

媽媽



NIUS  
妞新聞

女人

# 使用者為中心的大數據實務



# 雲端 與 虛擬化介紹



# 雲端運算的概念



# 雲端的分類

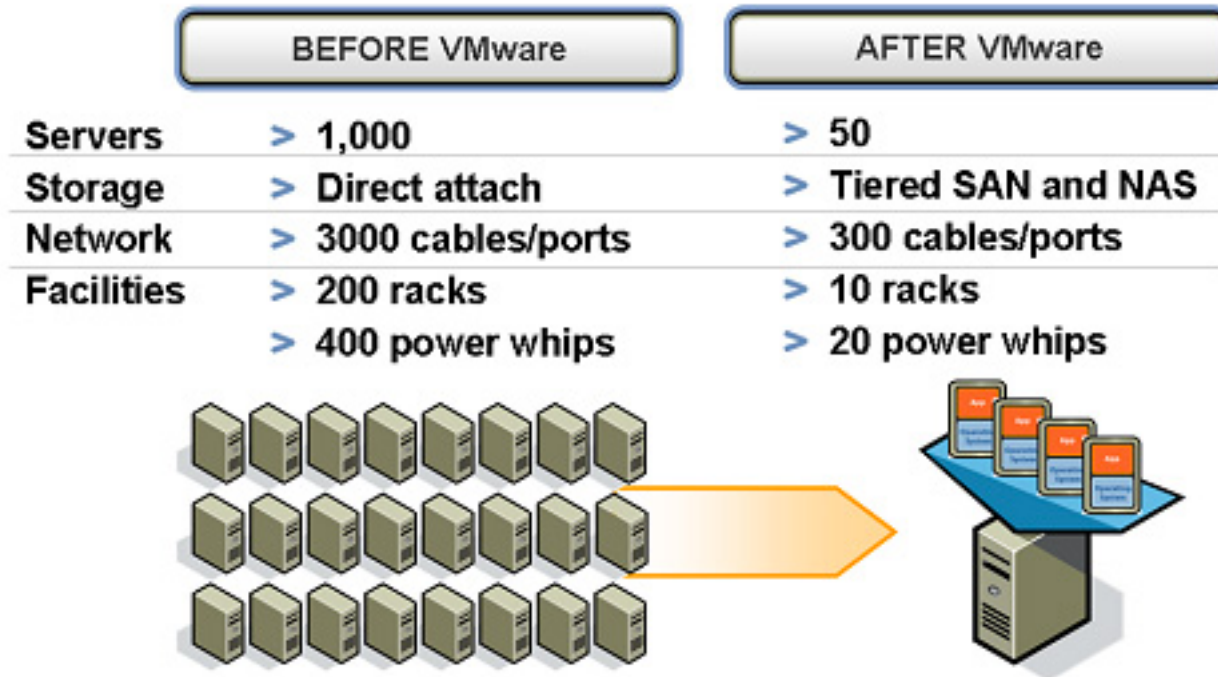


## PaaS: Platform as a Service

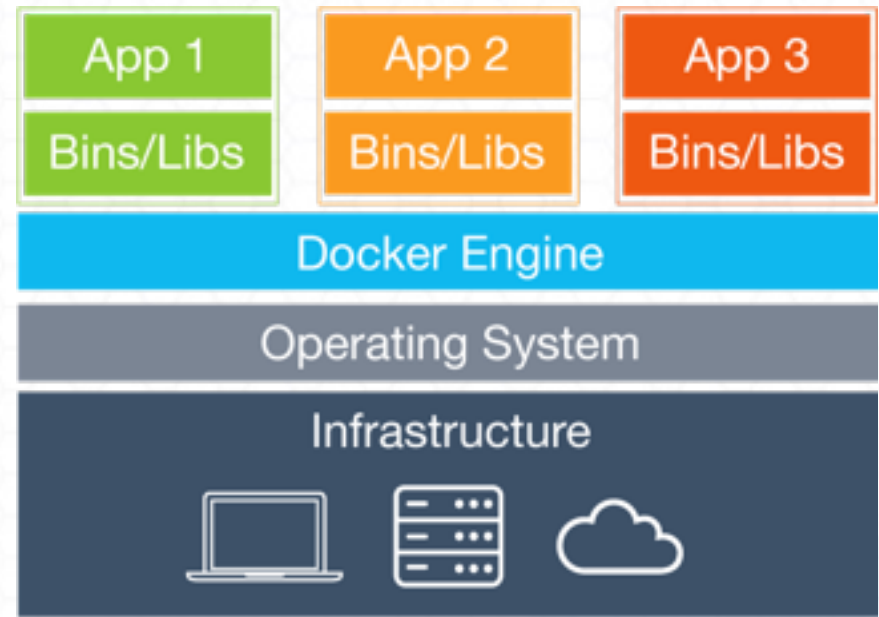
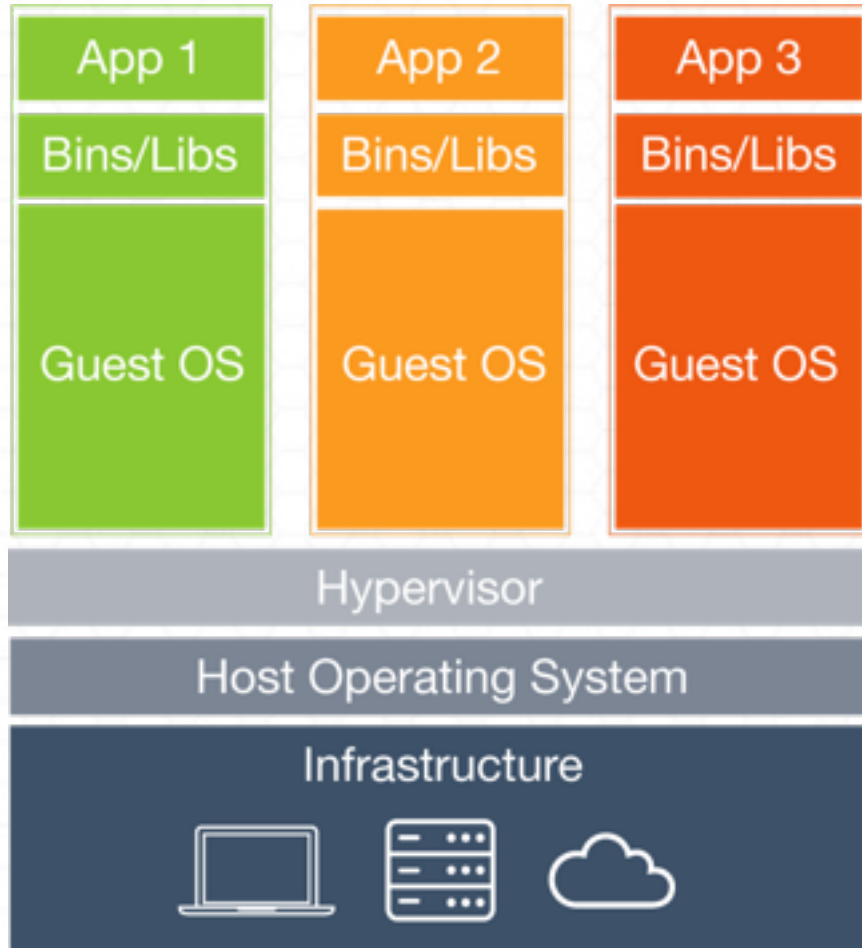




# 為什麼虛擬化？



# 虛擬機器 v.s. 虛擬環境



# 虛擬環境

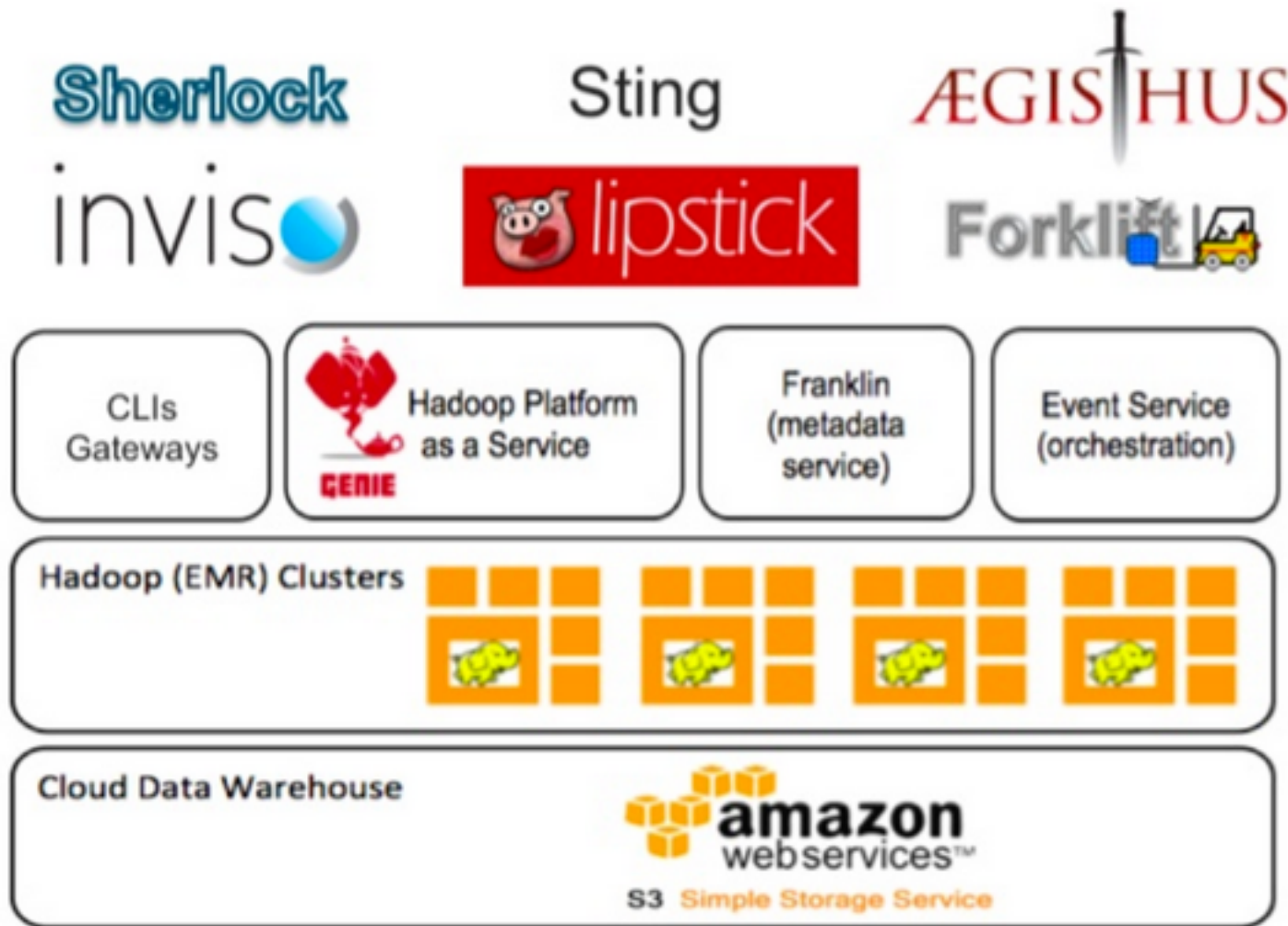


ch**root**



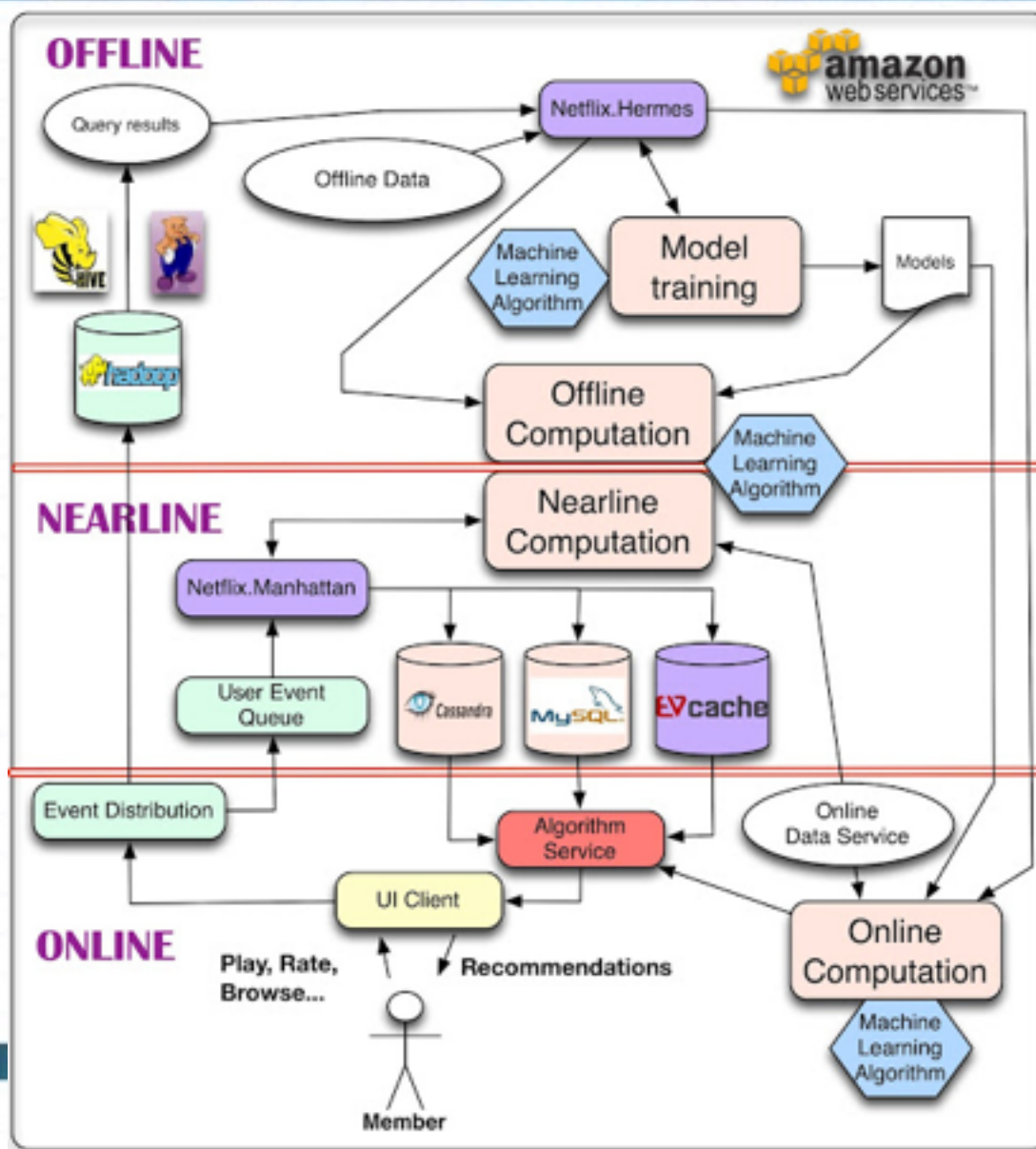
Rocket

# 最佳實務 - Netflix

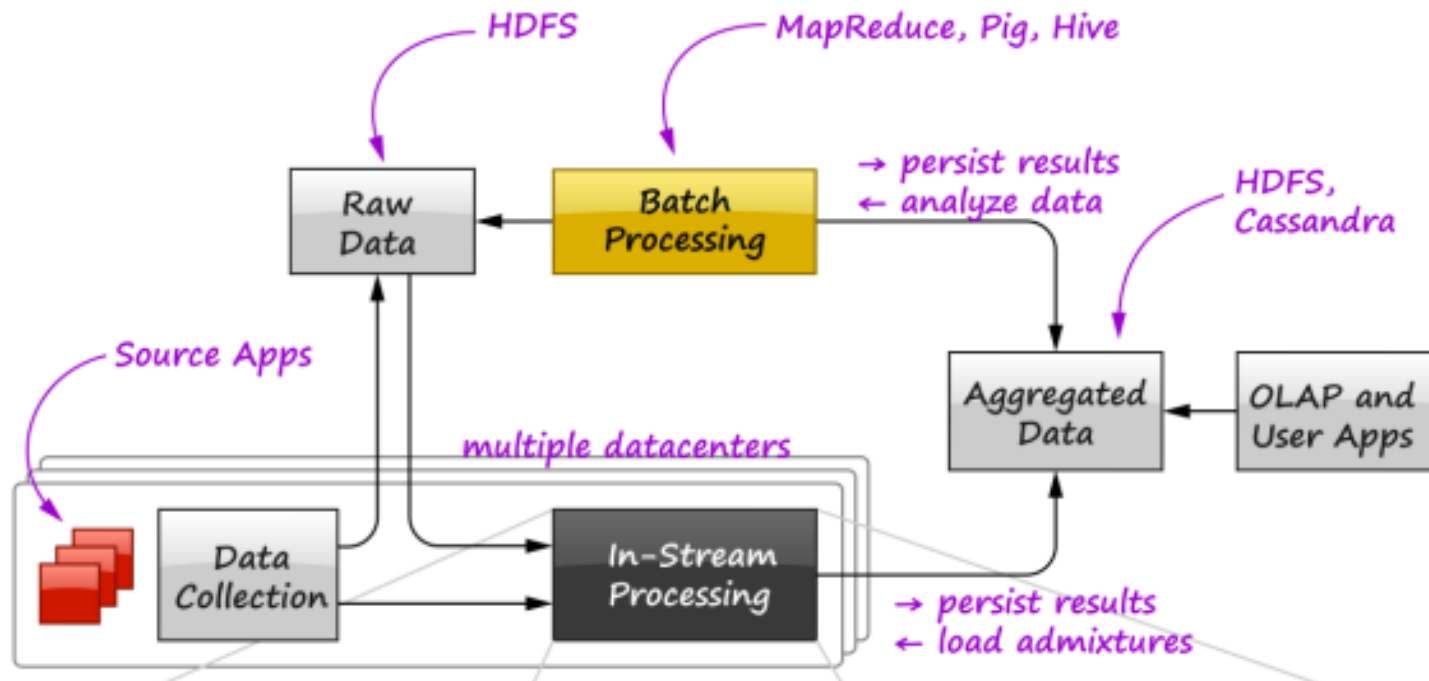




# 最佳實務 - Netflix

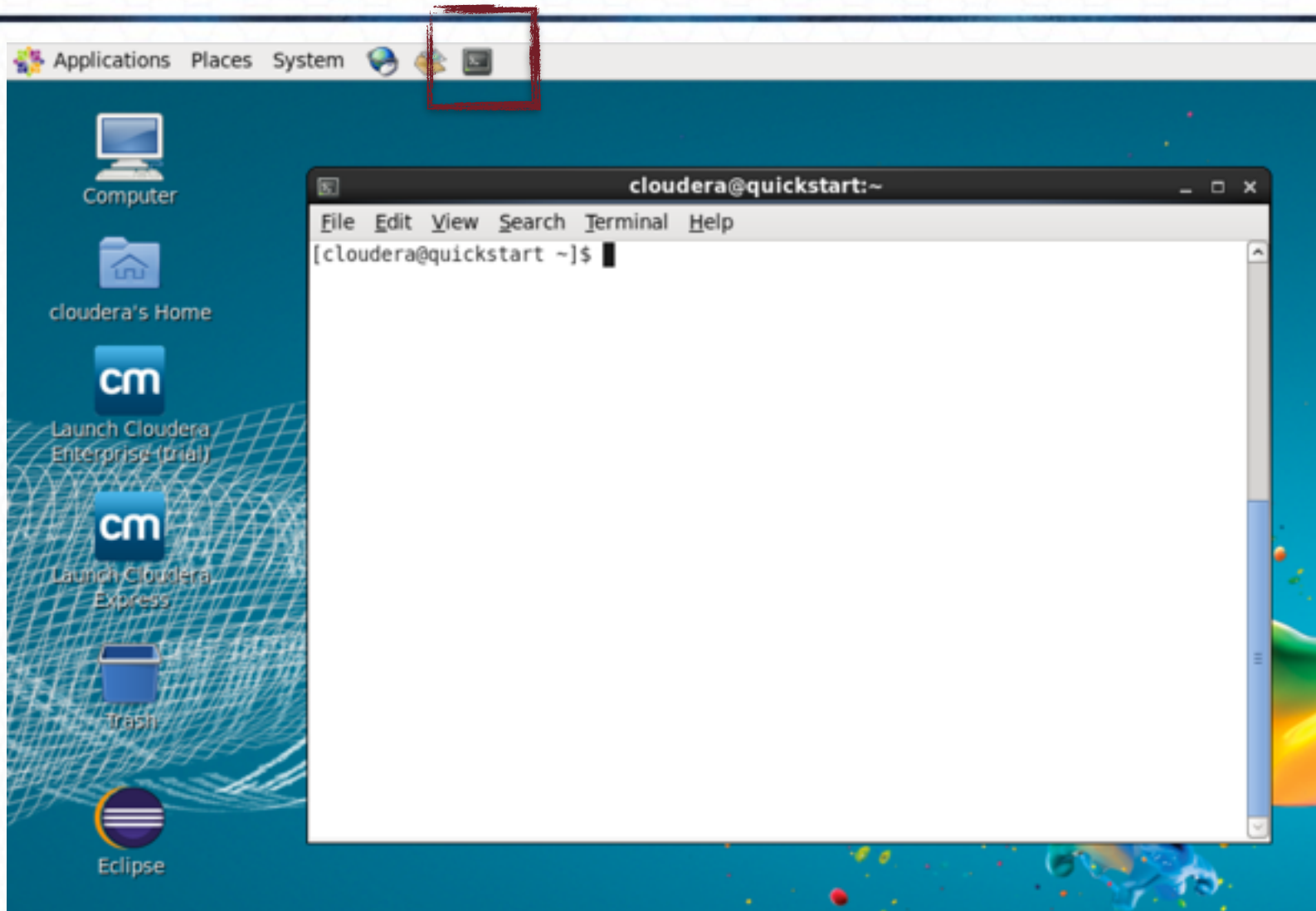


# 資料架構 - overview



# LINUX 操作

# 第一次玩linux就上手





# TAB 鍵是你最好的朋友

- 命令補全
- 檔案補齊
- TIPS: 雙擊TAB => 列出相同可補齊的指令

# CTRL + C 中斷目前程式

- 輸入了錯誤的指令...
- 跑了一個會持續運行的程式...
- 跑了一個「跑不停」的程式...

# 檔案系統操作

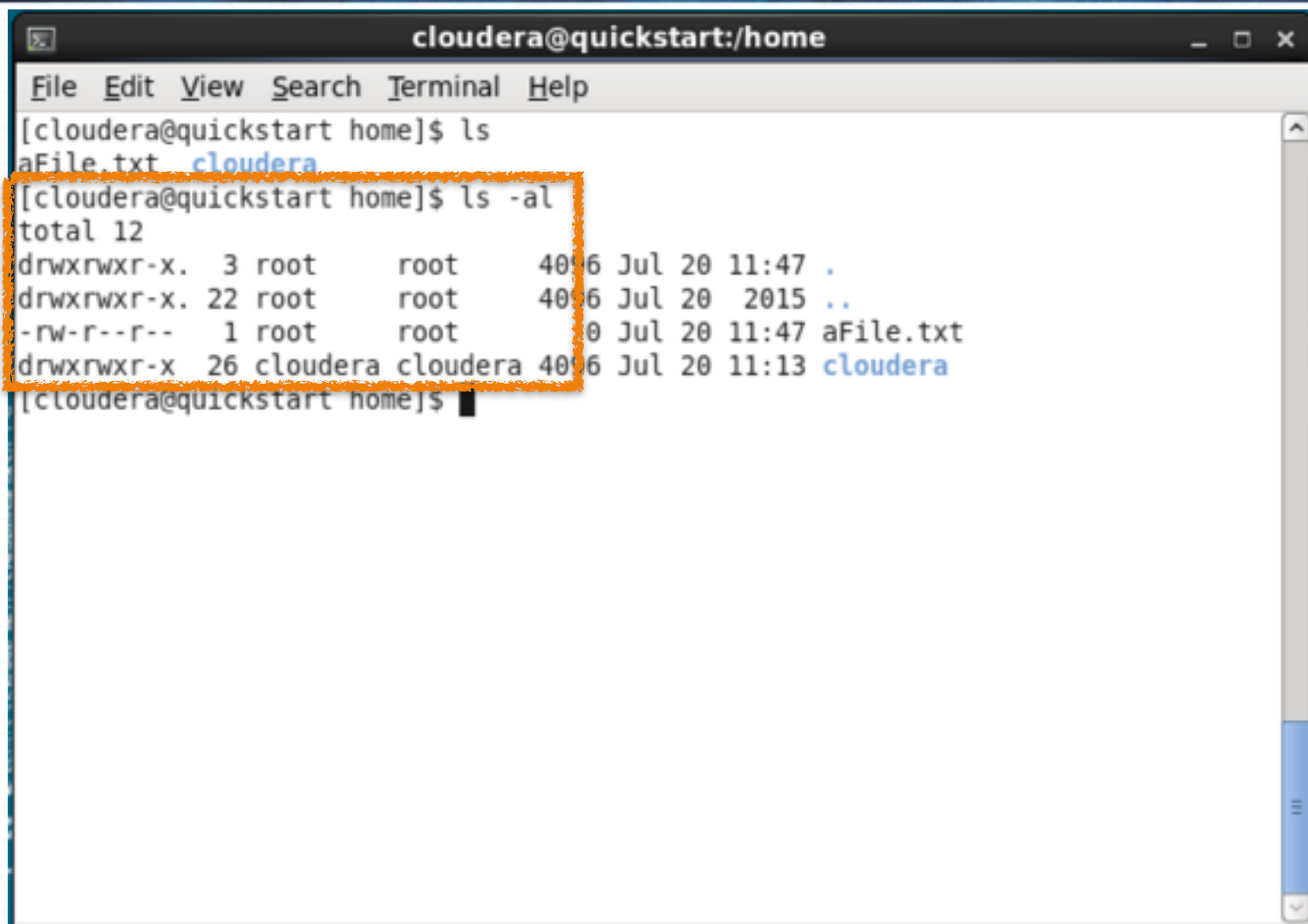
- Linux是一個大型檔案系統 - 資料夾 + 檔案
  - “cd” 指令讓你在資料夾移動
  - “ls -l” 指令可列出當前資料夾的檔案
  - “pwd” 指令可獲取當前目錄路徑
  - “touch 檔名.txt” 指令建立檔案
  - “mkdir 資料夾” 指令建立資料夾
  - “rm 檔案.txt” 刪除檔案
  - “rm -rf 資料夾” 刪除資料夾
  - “mv 檔案.txt 目標位置/新檔案.txt” 移動檔案
  - “cat 檔案.txt” 列出檔案內容

# 檔案系統

- 起始目錄在 /，我們也叫它root
- 每個資料夾、檔案都有它：
  - 使用者權限
  - 操作權限



# 檔案權限管理



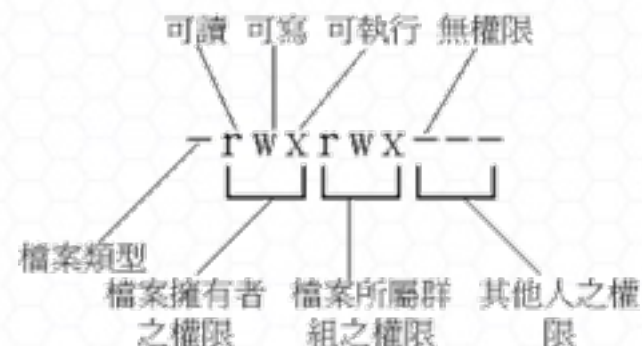
A terminal window titled "cloudera@quickstart:/home" with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
[cloudera@quickstart home]$ ls
aFile.txt  cloudera
[cloudera@quickstart home]$ ls -al
total 12
drwxrwxr-x.  3 root    root    4096 Jul 20 11:47 .
drwxrwxr-x. 22 root    root    4096 Jul 20 2015 ..
-rw-r--r--   1 root    root      0 Jul 20 11:47 aFile.txt
drwxrwxr-x  26 cloudera cloudera 4096 Jul 20 11:13 cloudera
[cloudera@quickstart home]$
```

The output of the `ls -al` command is highlighted with an orange box. The entry for the `cloudera` directory shows permissions `drwxrwxr-x`, owner `cloudera`, group `cloudera`, size `4096`, and timestamp `Jul 20 11:13`.

# 檔案權限管理

- “chown brian:brian 檔.txt”
  - 將使用者與群組都改為brian
- “chmod u+x 檔.txt”
- “chmod g-x 檔.txt”
- “chmod o+x 檔.txt”

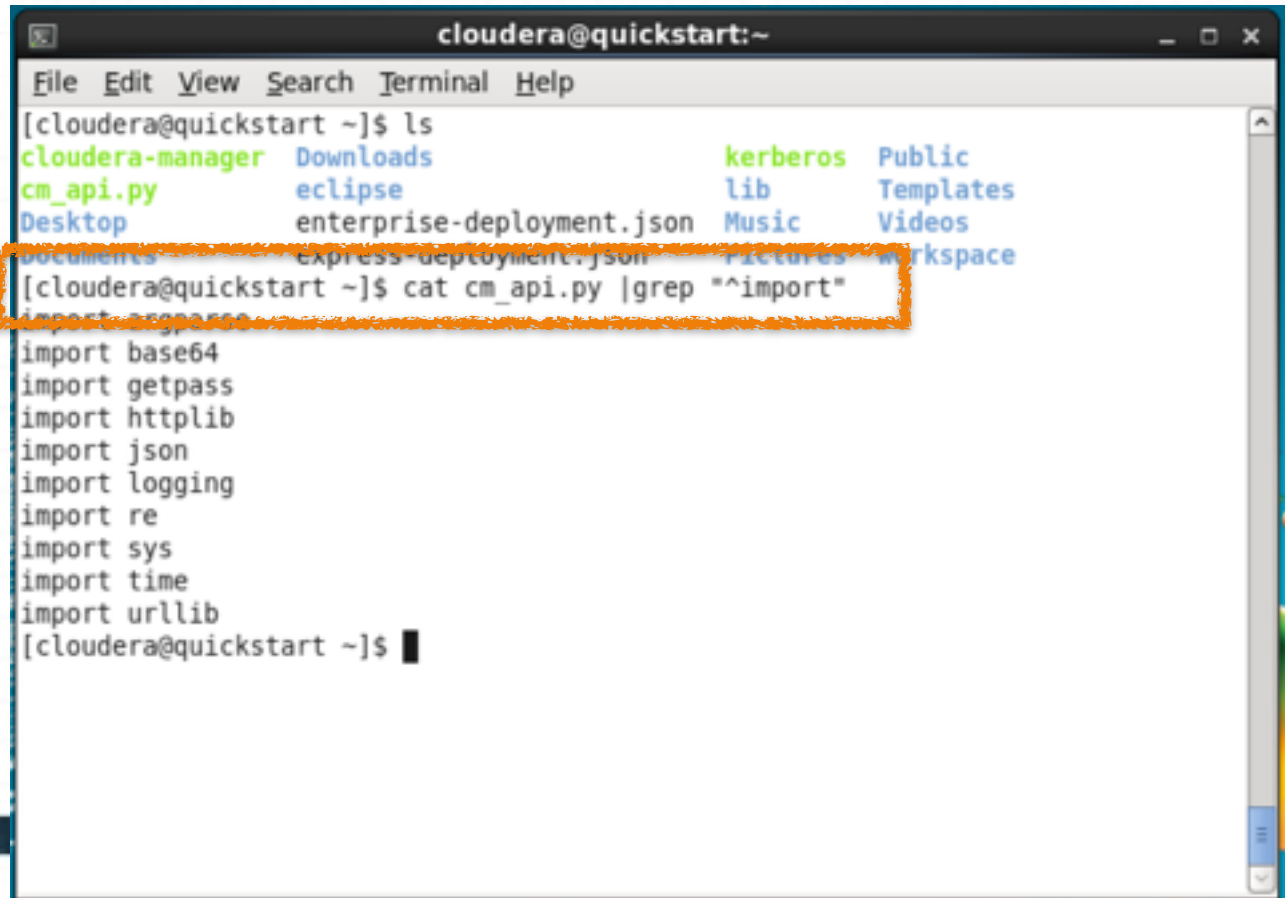


# 作業程序 與 網路

- “top” 可以列出「作業程序」的執行狀態
  - ctrl + c 可跳出
- “ps” 可以列出正在運型的作業程序
- “netstat -na” 可以列出對外服務的ip與port
- ifconfig 可列出所有的網路介面

# 其它不可不學指令 - grep

- 以“正規表達式”過濾關鍵字

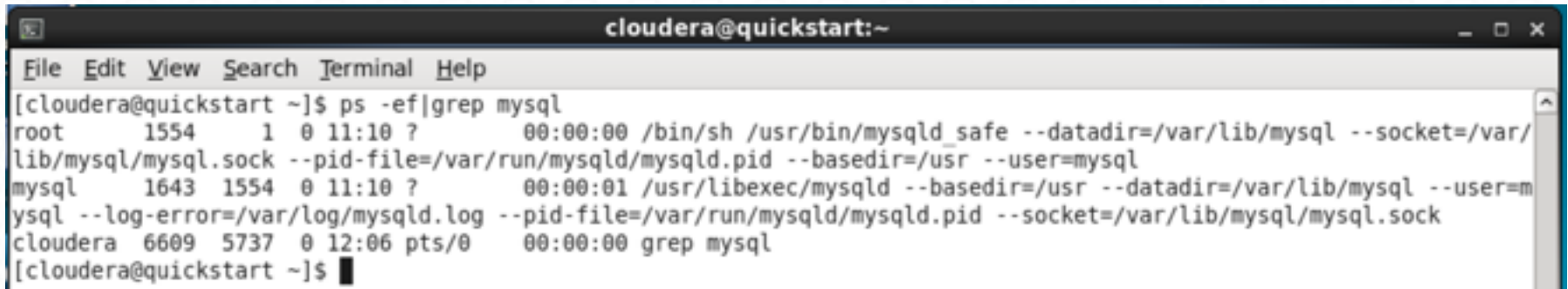


A terminal window titled "cloudera@quickstart:~" showing a file listing and a grep command. The file listing shows "cm\_api.py" and "enterprise-deployment.json". The grep command is "cat cm\_api.py | grep '^import'", which is highlighted with an orange box. The output shows the first few lines of the file, all starting with "import".

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ ls  
cloudera-manager Downloads kerberos Public  
cm_api.py eclipse lib Templates  
Desktop enterprise-deployment.json Music Videos  
documents express-deployment.json pictures workspace  
[cloudera@quickstart ~]$ cat cm_api.py | grep "^import"  
import argparse  
import base64  
import getpass  
import httplib  
import json  
import logging  
import re  
import sys  
import time  
import urllib  
[cloudera@quickstart ~]$
```



# mysql 作業程序是不是正在運行？



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ ps -ef|grep mysql  
root      1554      1  0 11:10 ?        00:00:00 /bin/sh /usr/bin/mysqld_safe --datadir=/var/lib/mysql --socket=/var/  
lib/mysql/mysql.sock --pid-file=/var/run/mysqld/mysqld.pid --basedir=/usr --user=mysql  
mysql     1643   1554  0 11:10 ?        00:00:01 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --user=m  
ysql --log-error=/var/log/mysqld.log --pid-file=/var/run/mysqld/mysqld.pid --socket=/var/lib/mysql/mysql.sock  
cloudera  6609   5737  0 12:06 pts/0    00:00:00 grep mysql  
[cloudera@quickstart ~]$
```

# 檔案權限

- 資料夾
- 檔案

# 文字編輯器



## ■ 指令列操作

vi \$filename	建立新檔案, 編輯現有檔案
:w	保存
:q!	不保存並離開
:wq	保存並離開
a	在目前的位置後增加字元
i	在目前的位置增加字元
x	刪除字元
/sql	搜尋關鍵字: <b>sql</b>
n	移動到下一個關鍵字



# **GIT**基礎簡介

# 版本控制系統

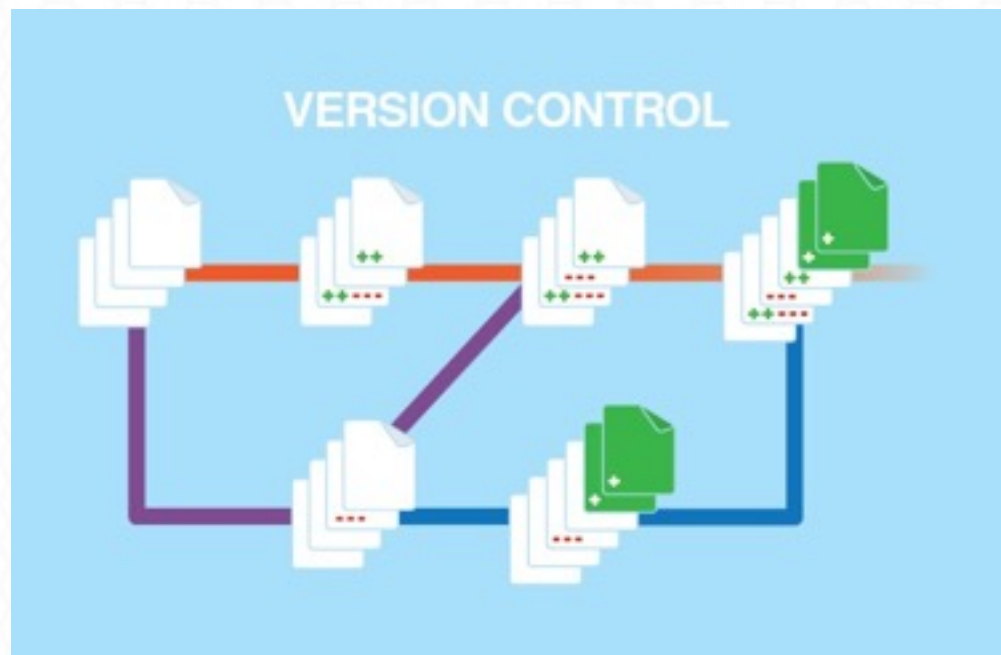
- VCS (Version Control System)

- CVS

- SVN (Yahoo)

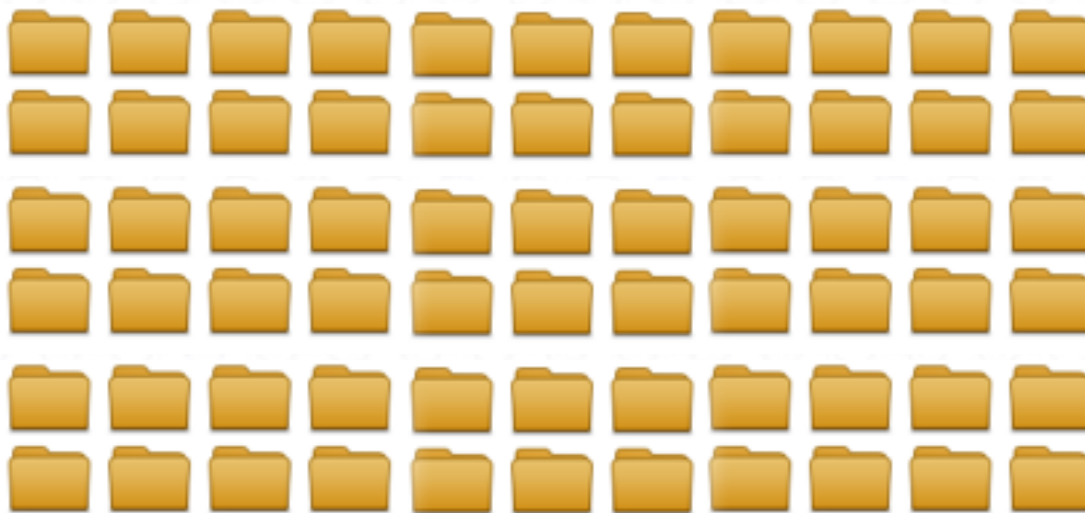
- Perforce (趨勢)

- GIT (KKBOX)



# 為什麼要做版本控制？

- 假使你的客戶一直修改需求，你怎麼管理每個版本？



## 多人共同工作

- 我們來共同開發一個網站吧，你負責A、B、C，我負責開發B、C、D；如果同時間必須更改到同一個檔案？該怎麼合併檔案？





# 使用版本控制系統管理工作

- 版本控制
  - Version Control, Revision Control, Source Control
- 每個專案下面都有Repository
  - Repository: 程式碼的倉庫，Commit 的集合
  - 包含修改時間跟敘述等基本資訊
- 版本控制都必須自行下Commit 指令後，才會生成新版本 (Version)

# GIT 發明者



- 為方便開發Linux Kernel 所設計的
- 原先是設計一組API 讓人可以快速方便 Commit 程式碼

# 使用**GIT**管理版本

---

- 快速開新分支 (Fast Branch)
- 本地端Commit (Local Commit)
- 提供許多工具瀏覽Commit 紀錄
- 方便與多人協作 (Distributed VCS)
- 可以用GitHub 快速分享程式碼

# Centralized VCS

- 需要網路才能Commit

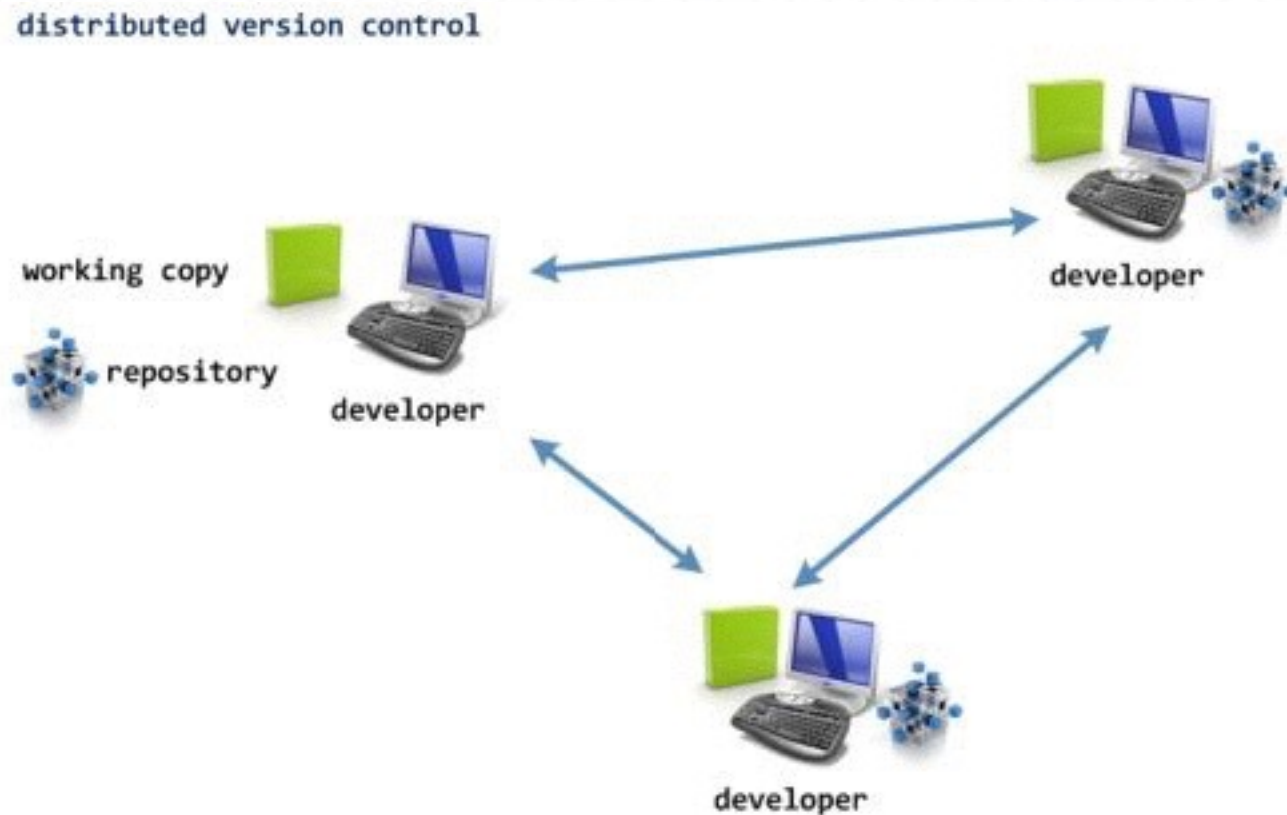
centralized version control





# Distributed VCS

- 每個人都保有一份完整的Code Commit



# 開始使用**GIT**

# 安裝GIT

- Windows

- <http://msysgit.github.com/>

- MAC

- `sudo port install git-core +svn +doc +bash_completion +gitweb`

- Linux

- `apt-get install git` (Ubuntu)

- `yum install git-core` (Centos)

# 建立 Repository

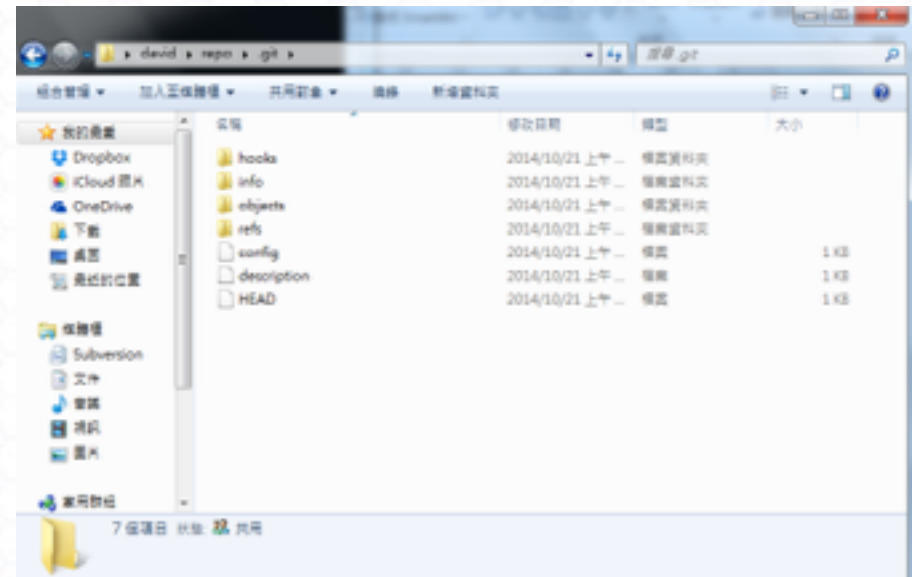
- 檢查 GIT 版本

- `git --version`

- 建立 Repository

- `git init repo`

- 檢視repo 下面的 .git





# 於目錄中建立Repository

- 於目錄repo2 建立repository
  - ▣ mkdir repo2
  - ▣ cd repo2
  - ▣ git init
- 目錄名字不重要，可以任意更改

# 刪除Repository

---

- Windows
  - 點選刪除.git 即可
- Linux, Mac
  - `rm -r .git`

## 小測驗

---

- 建立一個新的Repository於目錄test 下
- 檢視test 下的.git 包含的內容
- 刪除test 的Repository



提交更新 (**Commit**)

# 提交變化至Repository

- 於repo (含有.git)下建立 檔案 readme.txt
- 新增檔案至Staging Area
  - git add read.me.txt
    - 檔案新增或更新不會自動提交
    - 可以選擇性的提交檔案
  - git commit
    - # 後面接的是 comment
    - 提交變化至Repository



# 設定提交者基本資訊

- 使用git config 設定git
- 設定提交者資訊
  - ▣ git config --global user.name "david"
  - ▣ git config --global user.email [david@largitdata.com](mailto:david@largitdata.com)
- 新增修改
  - ▣ 修改readme.txt 的內容
  - ▣ git commit -a (提交變更)
- 查詢提交
  - ▣ git log

# 什麼時候該提交變更？

- 能不能想到好的註解 (comment)
  - ▣ 修改目的 (簡短且具意義)
- 更簡短的Commit 寫法
  - ▣ `git commit -a -m "add comment"`
    - -a: 提交所有(all)變更
    - -m: 變更訊息

# Staging area

- 控制該提交什麼檔案

- 顯示git 提交狀況

- ▣ git status

- 新增readme2.txt

- ▣ git status

On branch master

Untracked files:

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

readme2.txt

nothing added to commit but untracked files present (use "git add" to track)

# 提交實作

- 提交新增檔案readme2.txt
  - git status
  - git add readme2.txt
  - git commit -m "add readme2.txt"
- 取消新增至staging area 的檔案
  - git reset HEAD readme2.txt

# 瀏覽提交紀錄

- 檢視所有提交紀錄

- git log

- 使用 commit 後的hash 字串當唯一識別子
    - 為多人共同協作，所以必須取唯一Hash

- Git checkout to latest change

- Git checkout master

- Git log



# 回復舊有版本

- 回復之前版本

- ▣ git checkout 4bac
- ▣ 不用打完整Hash
- ▣ 發現之前新增檔案已消失
- ▣ 使用git log 檢視commit 紀錄

- 回到主線版本 (Master)

- ▣ git checkout master
- ▣ 使用git log 檢視commit 紀錄

- 回到上一版本 (HEAD~1)

- ▣ git checkout HEAD~1

# 比較版本差異

## ■ 使用git diff 比較差異

```
C:\Users\david\repo>git diff 4bac master
diff --git a/readme2.txt b/readme2.txt
new file mode 100644
index 0000000..d6b0a2b
--- /dev/null
+++ b/readme2.txt
@@ -0,0 +1 @@
+456d
\ No newline at end of file
```

## 小測驗

---

- 試者在test 下面commit 幾個檔案，並使用git log 觀察提交紀錄與變化，使用git diff 找到變化之處，最後使用git checkout 回復上一個版本

# 建立分支

# 建立分支

- 如果今天老闆要求你開發新的功能，但你為了不想影響線上系統，而要做開發測試，這時就必須建立分支。





# 建立新分支

- 切換到主幹線 (Master)
  - `git checkout master:`
  - Master 通常又名 Trunk
  - 主要是放置到 Production 環境
- 建立並切換分支
  - `git branch new_feature`
  - `git branch`
  - `git checkout new_feature`
  - `git branch`

# 檢視分支所在位置

---

- 檢視分支所在位置
  - ▣ `git branch`
- 建立新分支並切換至新分支
  - ▣ `git checkout -b cool_feature`

# 删除分支

---

- 删除分支
  - `git checkout master`
  - `git branch -D new_feature`

# 合并分支

# 合併分支

- 當所有人的新功能都已經開發到一段落，接著必須將新功能上線，便開始需要將所有之前開出去的分支做合併動作。
- 如果大家改的都是不同檔案？
- 如果大家會改到同個檔案？
  - 如何解決衝突(Conflict)?



# 合併實例 (沒有衝突)

- 新增檔案的合併方法

- ▣ 於new\_feature 新增檔案index.py (寫入部分內容)

- ▣ git checkout master

- ▣ git merge new\_feature

# 合併實例 (衝突)

- 新增檔案的合併方法

- 於cool\_feature 新增檔案index.py (寫入部分內容)

- git checkout master

- git merge cool\_feature

- 遇到衝突!

- 修改檔案

- 去除標示

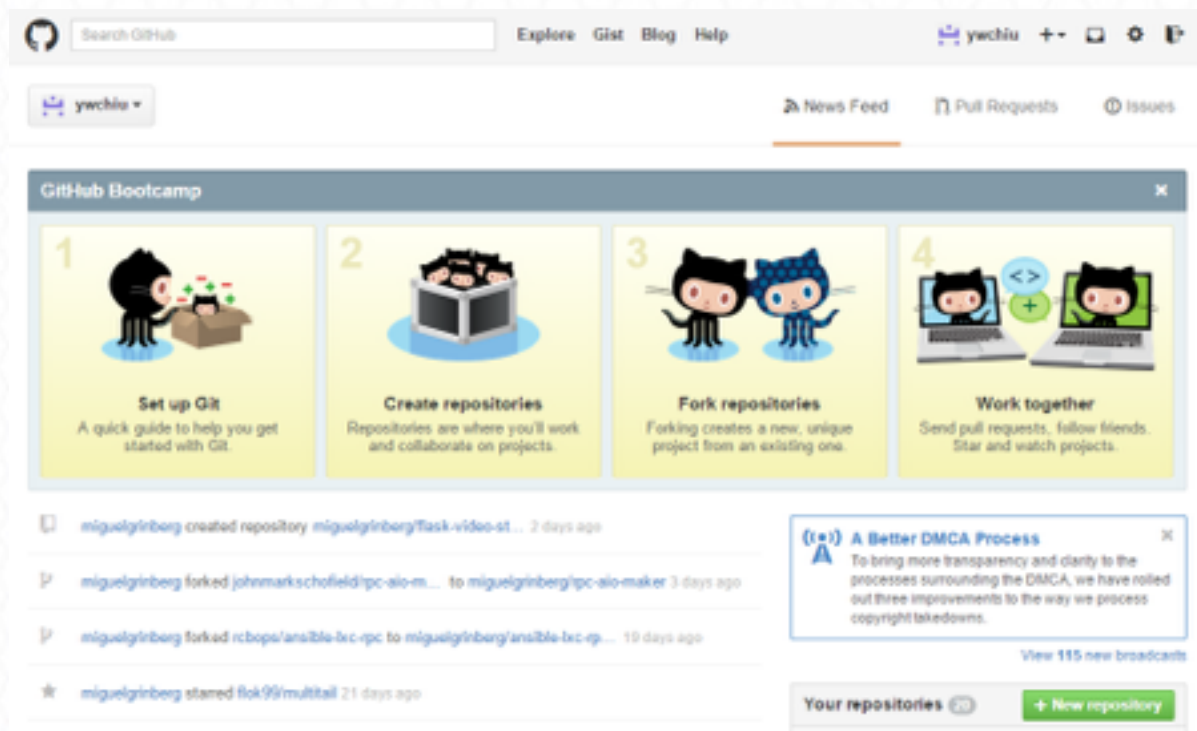
- Git commit



# 遠端儲存庫

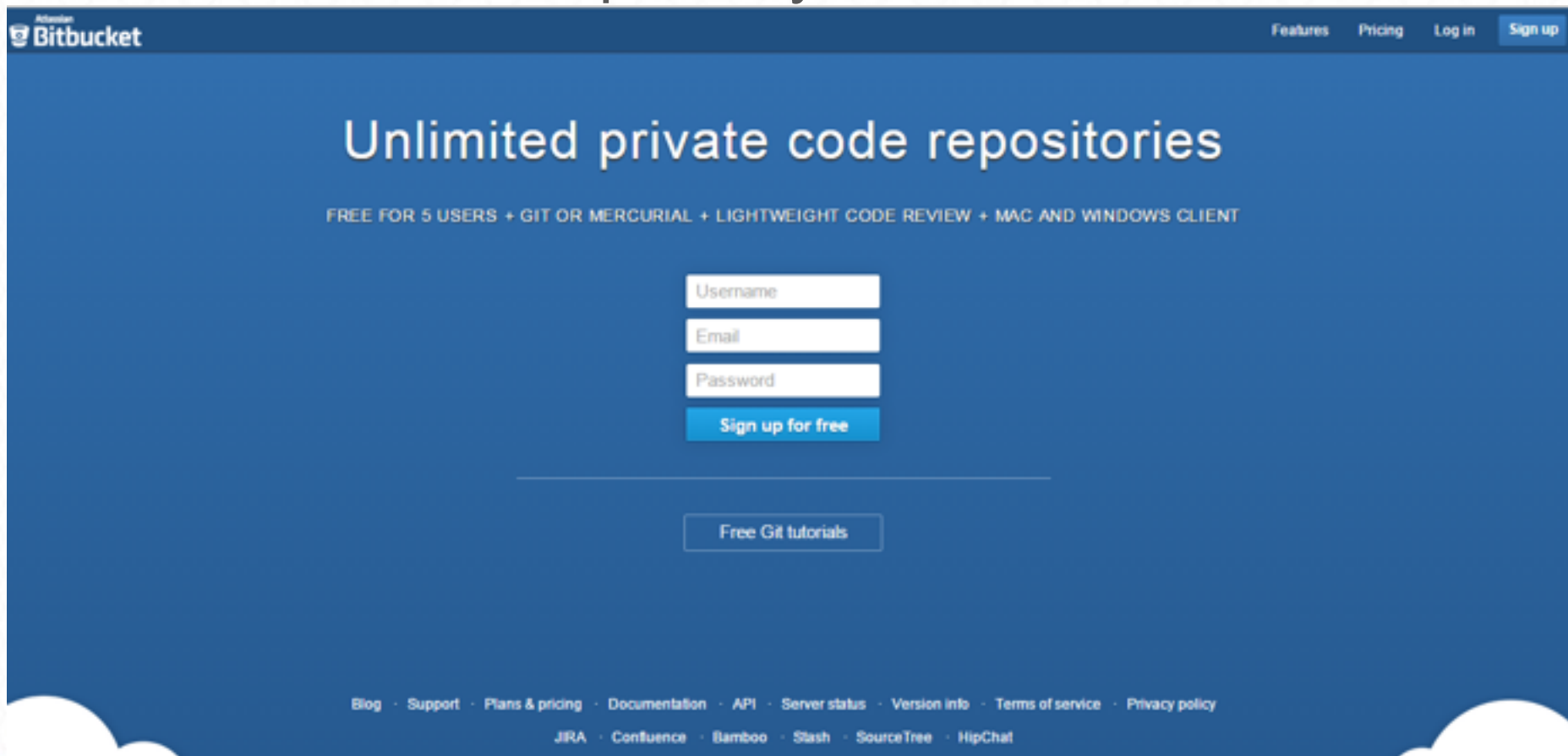
# 使用github

- 免費的open source repository  
(私密repository需要付費)



# Bitbucket (推薦使用)

- 可以無限放私密repository



The image is a screenshot of the Bitbucket website's sign-up page. The page has a dark blue header with the Bitbucket logo on the left and links for 'Features', 'Pricing', 'Log in', and 'Sign up' on the right. The main content area is a lighter blue and features the heading 'Unlimited private code repositories' in white. Below this heading is a sub-headline in all caps: 'FREE FOR 5 USERS + GIT OR MERCURIAL + LIGHTWEIGHT CODE REVIEW + MAC AND WINDOWS CLIENT'. In the center of the page is a white sign-up form with three input fields labeled 'Username', 'Email', and 'Password'. Below these fields is a blue button with the text 'Sign up for free'. Underneath the button is a horizontal line, and below that is a button labeled 'Free Git tutorials'. At the bottom of the page, there is a footer with a list of links: 'Blog', 'Support', 'Plans & pricing', 'Documentation', 'API', 'Server status', 'Version info', 'Terms of service', 'Privacy policy', 'JIRA', 'Confluence', 'Bamboo', 'Slash', 'SourceTree', and 'HipChat'.

Bitbucket

Features Pricing Log in Sign up

## Unlimited private code repositories

FREE FOR 5 USERS + GIT OR MERCURIAL + LIGHTWEIGHT CODE REVIEW + MAC AND WINDOWS CLIENT

Username

Email

Password

Sign up for free

Free Git tutorials

Blog · Support · Plans & pricing · Documentation · API · Server status · Version info · Terms of service · Privacy policy

JIRA · Confluence · Bamboo · Slash · SourceTree · HipChat



# Git remote add repository

---

- `git clone https://github.com/user/repo.git`
- `git remote add origin https://github.com/user/repo.git`

# Start from scratch

- `mkdir /path/to/your/project`
- `cd /path/to/your/project`
- `git init`
- `git remote add origin git@bitbucket.org:ywchiu/rbasic.git`
- `echo "David Chiu" >> contributors.txt`
- `git add contributors.txt`
- `git commit -m 'Initial commit with contributors'`
- `git push -u origin master`

# Existing Project

---

- `cd /path/to/my/repo`
- `git remote add origin git@bitbucket.org:ywchiu/rbasic.git`
- `git push -u origin --all` # pushes up the repo and its refs for the first time
- `git push -u origin --tags` # pushes up any tags

- git pull

# 流程測驗

- 當你發現你的同事有個branch 上面有錯誤，而這個問題很簡單就可以解決，你該怎麼做？
  - 解決bug, 提交變更,送至遠端repo
- 如果有人發現你打錯字, 最好的解解決方式
  - 在master 修改錯字就好
- 如果你今天做的一半後，發現有個更好的做法，請問該怎麼做？
  -



# 流程測驗

---

- 如果你的網站已經上production，而目前有使用者報告你的網站有問題，請問該如何修復該問題？
- 你被主管要求要開一個新功能，請問該如何開始第一步？

The background features a light blue hexagonal grid pattern. Overlaid on this is a large, stylized graphic of concentric circles and arcs in shades of blue and white, resembling a target or a complex geometric design. A solid dark blue horizontal bar runs across the top of the slide.

**THANK YOU**