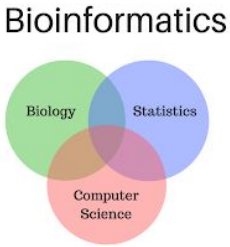
生物資訊環境系統設置基礎教學
Bioinformatics System Environment Implementation

2020/07/30 (Thur.)
Speaker: Kent Chen (GBST, Bioinformaticsian)

https://t.ly/Qddy

What is Bioinformatics



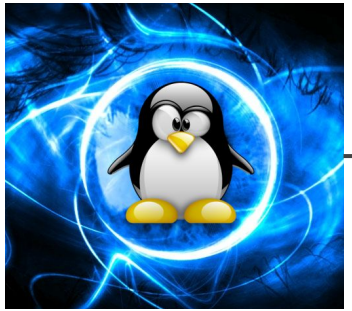
- 生物資訊學利用應用數學、**資訊學、統計學和電腦科學**的方法**研究生物學的問題**。生物資訊學的研究材料和結果就是各種各樣的生物學資料，其研究工具是電腦，研究方法包括對生物學資料的搜尋、處理及利用。目前主要的研究方向有：**序列比對、序列組裝、基因辨識、基因重組、蛋白質結構預測、基因表現、蛋白質反應的預測，以及建立進化模型。** (from Wikipedia)

- Different work of **Bioinformatician** & **Biostatician**
  - “Biostatistics” is the science of designing, conducting, **analyzing and interpreting studies aimed at improving public health and medicine**.
  - “Bioinformatics” is the science of developing and **applying computational algorithms and analysis methodologies to big biological data such as genetic sequences.**
    (https://publichealth.gwu.edu/departments/biostatistics-and-bioinformatics)

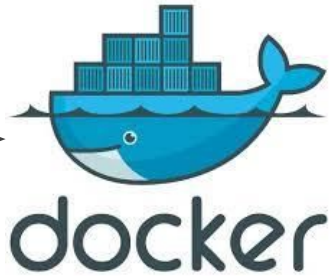
Outline

Be a Bioinformatician,
you Should Be Learning Linux as much as you can.
By KentChen

Linux



Docker



Conda

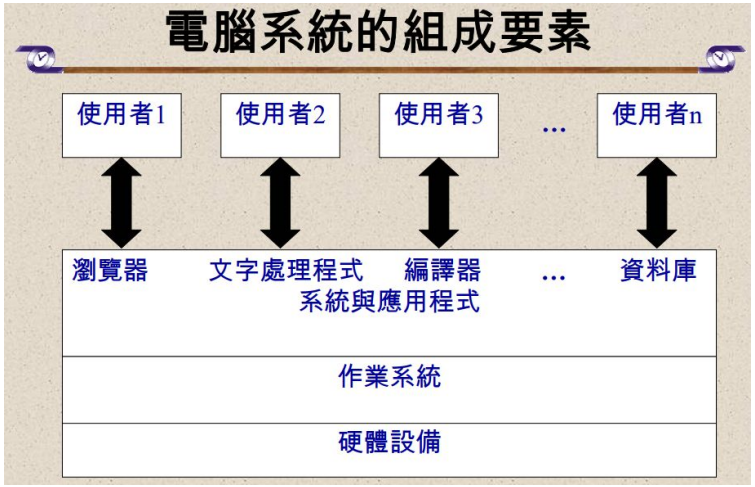
CONDA®

Linux

Docker → Conda

What is Linux

- 就和Windows and MacOS一樣，它就是一個作業系統(OS)
- 每個作業系統一定會有一個**核心 (kernal)**，其上層會搭建許多系統軟體
  - 編譯環境 (把程式碼轉成讓電腦看的懂的東西)
  - 應用程式介面 (API)
  - 使用者介面 (AUI): shell / GUI (Graphical User Interface)



Linux Ubuntu GUI

Pros & Cons of Linux OS

Pros

- 穩定的系統
- 免費或少許費用 (open source)
- 安全性、漏洞的修補
- **多工、多使用者***
- **使用者與群組的規劃**
- 相對比較不耗資源的系統

Cons

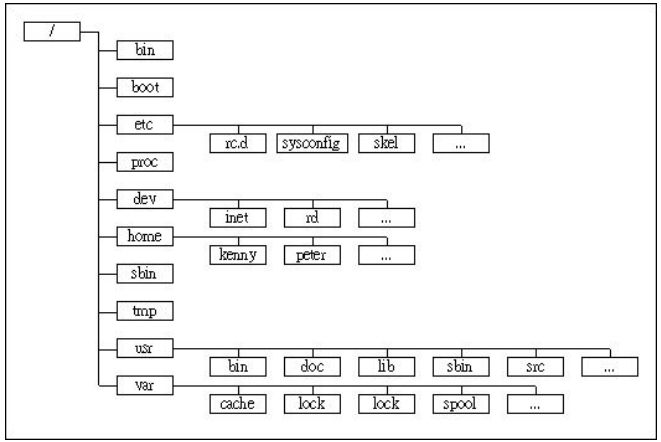
- 沒有特定的支援廠商
- 圖形介面作的還不夠好
  - 所以 command 還是很重要
- 無法使用 microsoft office

Why Bioinformatician Need To Learn Linux

- 生物資訊的工具時常需要吃大量的記憶體,**相對於Windows系統，穩定且快速**
- 絕大部分的生物資訊工具是**在Linux上開發的**
- Linux的**多工處理**
- 使用bash shell非常容易處理大量資料
    - shell是命令直譯器，用於解析和執行命令。它對用 戶隱藏了操作系統底層 kernel 的複雜性，是兩者間的橋梁。

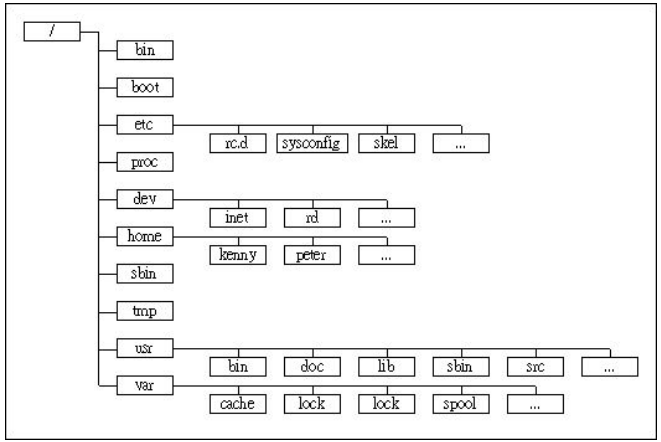
Linux 檔案系統架構

- /bin, /sbin, /usr/bin：主要放置一般使用者可以操作的指令
- /dev：放置 device 裝置檔案，包括滑鼠鍵盤等
- /etc：主要放置系統檔案
- **/home, /root：主要是一般帳戶的家目錄, /root 為系統管理者的家目錄**
- /usr：全名為 unix software resource 縮寫，放置系統相關軟體、服務
- /tmp：全名為 temporary，放置暫存檔案



Linux 檔案系統架構

- /bin, /sbin, /usr/bin：主要放置一般使用者可以操作的指令
- /dev：放置 device 裝置檔案，包括滑鼠鍵盤等
- /etc：主要放置系統檔案
- **/home, /root：主要是一般帳戶的家目錄, /root 為系統管理者的家目錄**
- /usr：全名為 unix software resource 縮寫，放置系統相關軟體、服務
- /tmp：全名為 temporary，放置暫存檔案



Terminal

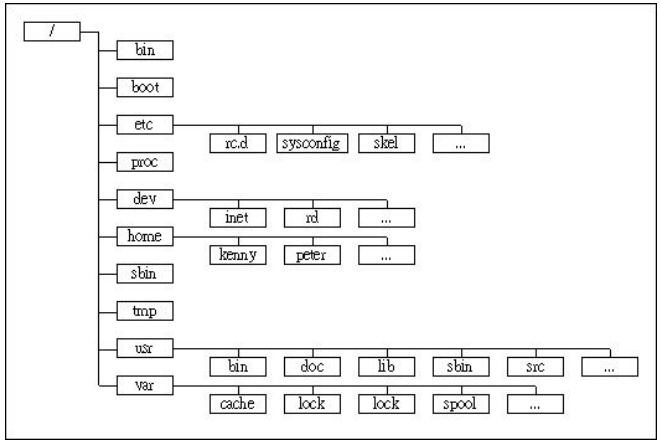
```
(base) kentchen@KentdeMacBook-Pro ~ $tree -L 1 /
/
├── Applications
├── Library
├── Preboot
├── Preboot\ 1
├── Secomba
├── System
├── Users
├── Volumes
├── bin
├── cores
├── dev
├── etc -> private/etc
├── home -> /System/Volumes/Data/home
├── opt
├── private
├── sbin
├── tmp -> private/tmp
├── usr
└── var -> private/var
```

Terminal

```
/Users/kentchen/
├── ACT
├── Applications
├── Box\ Sync
├── Desktop
├── Docker
├── Documents
├── Downloads
├── Dropbox
├── Google\ Drive\ F
├── GoogleDrive -> /
├── Library
├── MEGAsync\ Downlo
├── Movies
├── Music
├── OneDrive\ -\ @\2
├── Pictures
├── Public
├── Test
├── VirtualBox\ VMs
├── cloud_backup.sh
├── github
├── igv
├── miniconda3
├── mnt
├── opt
├── perl5
└── wekafiles
```

Linux 檔案系統架構

- /bin, /sbin, /usr/bin：主要放置一般使用者可以操作的指令
- /dev：放置 device 裝置檔案，包括滑鼠鍵盤等
- /etc：主要放置系統檔案
- **/home, /root：主要是一般帳戶的家目錄, /root 為系統管理者的家目錄**
- /usr：全名為 unix software resource 縮寫，放置系統相關軟體、服務
- /tmp：全名為 temporary，放置暫存檔案



Terminal

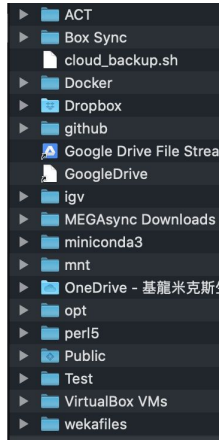
```
(base) kentchen@KentdeMacBook-Pro ~ $tree -L 1 /
/
├── Applications
├── Library
├── Preboot
├── Preboot\ 1
├── Secomba
├── System
├── Users
├── Volumes
├── bin
├── cores
├── dev
├── etc -> private/etc
├── home -> /System/Volumes/Data/home
├── opt
├── private
├── sbin
├── tmp -> private/tmp
├── usr
└── var -> private/var
```

Terminal

```
/Users/kentchen/
├── ACT
├── Applications
├── Box\ Sync
├── Desktop
├── Docker
├── Documents
├── Downloads
├── Dropbox
├── Google\ Drive\ F
├── GoogleDrive -> /
├── Library
├── MEGAsync\ Downlo
├── Movies
├── Music
├── OneDrive\ -\ @\2
├── Pictures
├── Public
├── Test
├── VirtualBox\ VMs
├── cloud_backup.sh
├── github
├── igv
├── miniconda3
├── mnt
├── opt
├── perl5
└── wekafiles
```

=

GUI



How to install Linux in Windows

- **Install by Virtual Machine: (beginner recommended)**
  - How to Install Linux on Windows 10 | Learn In One Video
  - 下載並安裝 VirtualBox
  - 下載Ubuntu作業系統
    - 桌面版 (with GUI)
    - 伺服器版 (command only)

Bash Command 基本操作

1. ls：list，查看檔案及子目錄

列出基本資料夾資料：

```
ls
```

列出詳細資料和隱藏資料：

```
// -l 列出詳細資料 -a 列出隱藏資料
$ ls -la
```

列出部分檔案：

```
// 列出為 .js 的檔案
$ ls *.js
```

Bash Command 基本操作

2. pwd：print work directory，印出目前工作目錄

```
$ pwd
// /Users/happycoder/Desktop/projects/HappyCoder
```

3. cd：change directory，移動進入資料夾

移動到目前資料夾下的 examples 資料夾：

```
$ cd ./examples
```

移動到家目錄：`~`：

```
$ cd ~
```

移動到上一層目錄 `..`：

```
$ cd ..
```

移動到根目錄 `/`：

```
$ cd /
```

Bash Command 基本操作

4. mkdir：make directory，創建新資料夾

```
$ mkdir examples
```

5. cp：copy，複製檔案

先將字串 TEST 存入 README.md 文件中

```
$ echo "TEST" > README.md
```

```
$ cp README.md
```

6. mv：move (rename) files，移動檔案或是重新命名檔案

移動檔案：

```
$ mv README.md /examples/README.md
```

重新命名

```
$ mv README.md README_MV.md
```

Bash Command 基本操作

7. rm：remove file，刪除檔案

```
$ rm README.md
```

刪除目前資料夾下副檔名為 .js 檔案：

```
$ rm *.js
```

刪除資料夾和所有檔案：

```
$ rm -f examples
```

Bash Command 基本操作

9. cat：將文件印出在終端機上

```
$ cat README.md
```

10. tail：顯示檔案最後幾行內容

```
$ tail README.md
```

持續顯示更新內容，常用於 web server 看 log debug 使用：

```
$ tail -f README.md
```

$ head README.md

Bash Command 基本操作

8. touch：用來更新已存在文件的 timestamp 時間戳記或是新增空白檔案

```
$ touch README.md
```

1. nano：在終端機編輯文字檔案

編輯或是新增文字檔案：

```
$ nano README.md
```

啟動編輯完後可以使用 Ctrl + X 離開，Ctrl + V 移動到上一頁，Ctrl + Y 移動到下一頁，Ctrl + W 搜尋文字內容

2. vim：在終端機編輯文字檔案

```
$ vim README.md
```

啟動後，使用 i 進入編輯，esc 離開編輯模式，`:q` 不儲存離開，`:wq` 儲存離開，`:q!` 強制離開

Bash Command 進階操作
(搜尋資料夾並擷取特定字串)

```
find /User/kentchen/Dropbox -maxdepth 1 -name "*.txt" -exec grep -i ‘bacillus’ {} \;
```

- /User/kentchen/Dropbox: 選擇要尋找的路徑
- -maxdepth 1: 僅搜尋第一層的內容
- *.txt: 要尋找的檔案類型
- -exec 找到符合的檔案後接著執行那個指令
- grep -i: 抓取關鍵字, 且不分大小寫
- bacillus 檔案內容有這個keyword的檔名

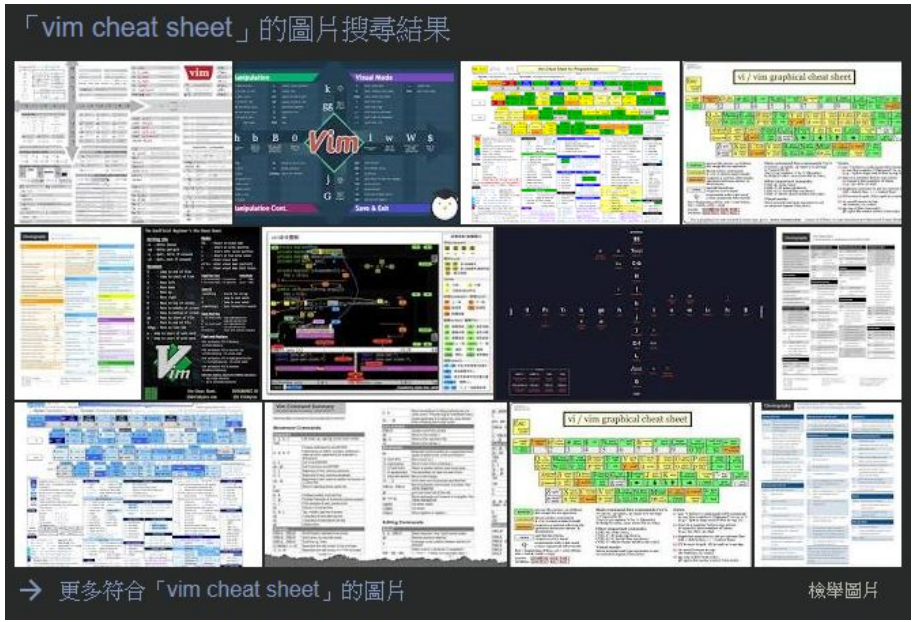
Bash Command 進階操作
(對大量資料進行擷取並處理)

如何單獨擷取”Feature”欄位”，並得知每一種feature共有多少數量呢？

Prokka_ID	Source	Feature	Start	End	Strand	Unitig	EC_number	Gene	Product	refID	BLASTP
BGHBDAGM	Prodigal:2.6	CDS	80	325	+	gnl|X|BGHBDAGM_1			hypothetical p	Q5DQT8	Q5DQT8|Q5I
BGHBDAGM	Prodigal:2.6	CDS	403	921	+	gnl|X|BGHBDAGM_1			hypothetical p	D0CG66	D0CG66|D0C
BGHBDAGM	Prodigal:2.6	CDS	1131	1430	+	gnl|X|BGHBDAGM_1			hypothetical p	A0A0E8PKM	A0A0E8PKM
BGHBDAGM	Prodigal:2.6	CDS	1996	2352	+	gnl|X|BGHBDAGM_1		higB2_1	Putative toxin	A0A009QCJ(	A0A009QCJ(
BGHBDAGM	Prodigal:2.6	CDS	2345	2647	+	gnl|X|BGHBDAGM_1		higA1_1	Antitoxin Hig	Q5DQT4	Q5DQT4|Q5I
BGHBDAGM	Prodigal:2.6	CDS	2640	2849	+	gnl|X|BGHBDAGM_1			hypothetical p	A0A0K9ATF	A0A0K9ATF
BGHBDAGM	Prodigal:2.6	CDS	3194	3553	+	gnl|X|BGHBDAGM_1			hypothetical p	-	-
BGHBDAGM	Prodigal:2.6	CDS	3630	4034	-	gnl|X|BGHBDAGM_1			hypothetical p	A0A334ZHD	A0A334ZHD

1. 下載此連結的檔案 (annotation.txt)：https://t.ly/RfTW
2. 進入此檔案所在的資料夾並檢視裡面有什麼檔案 ($ cd /Users/kentchen/Documents & ls -l)
3. 擷取"Feature"欄位 ($ cut -f3 annotation.txt)
4. 計算每一種feature共有多少數量 ($ cut -f3 annotation | sort | uniq -c)

Linux command cheat sheet
Vim cheat sheet



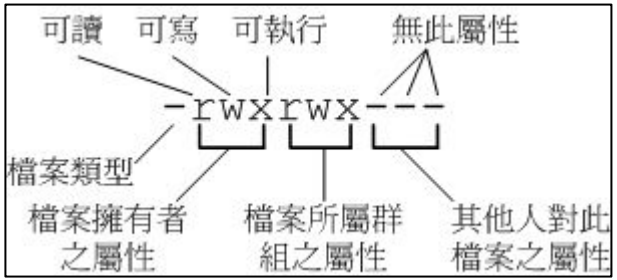
檔案權限設定

在 Linux 系統中，每一個 Linux 檔案都具有四種存取權限：

1. 可讀取 `(r, Readable)`，用數字 4 表示
2. 可寫入 `(w, writable)`，用數字 2 表示
3. 可執行：`(x, eXecute)`，用數字 1 表示
4. 無權限(-)，用數字 0 表示

系統管理者依據使用者需求來設定檔案權限，若我們想檢視檔案權限可以使用 `$ ls -l` 來查看

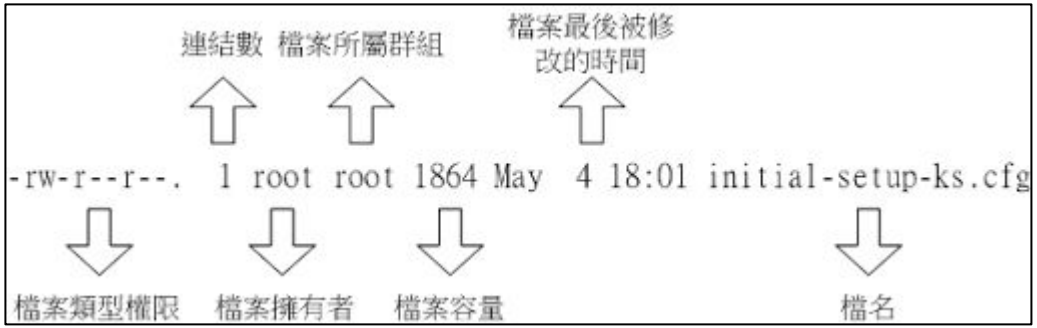
檔案權限設定



在 Linux 系統中，每一個 Linux 檔案都具有四種存取權限：

1. 可讀取 (r, Readable)，用數字 4 表示
2. 可寫入 (w, writable)，用數字 2 表示
3. 可執行：(x, eXecute)，用數字 1 表示
4. 無權限(-)，用數字 0 表示

系統管理者依據使用者需求來設定檔案權限，若我們想檢視檔案權限可以使用 `$ ls -l` 來查看



1. chmod：修改檔案權限

將權限設為 `rw-rw-r--`：

```
$ chmod 664 README.md
```

將檔案的使用者和群組加入執行權限

```
$ chmod ug+x README.md
```

2. chown：修改檔案擁有者與群組

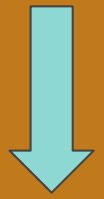
```
$ chown www-data:www-data README.md
```

基本 BASH shell script 寫法

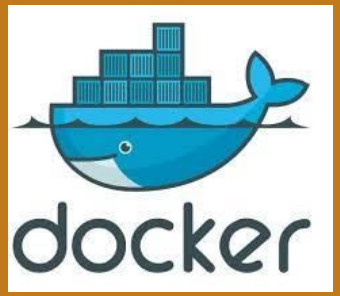
```
#! /bin/bash
echo “Hello World”
date
hostname
sleep 60
echo “Bye”
```

Play yourself

- 目前所在的位置資料夾該如何得知?
- Linux作業系統中, cd ../../ 命令的作用是？
- 請創建一個檔案 (test.txt)，內容第一行是”GBST”，第二行是”I Love Bioinformatics”，儲存離開並將此檔案移到/Users/{user name}/BFX_training資料夾底下
- 修改test.txt文件權限為 rwxr-xr--
- 請使用find搜尋/Users/{user name}/BFX_training資料夾，並抓出含有bioinformatics關鍵字的列



Docker

Linux →  → Conda

Why Docker is important

相信只要是IT公司一定會遇到以下情境：

情境一：

開發人員(RD)：我的程式寫好並打包了，麻煩幫我安裝到客戶的電腦。

維運人員(SE)：好的，已安裝完畢。

這情境當然是最好的情境，可惜現實總是殘酷的...

情境二：

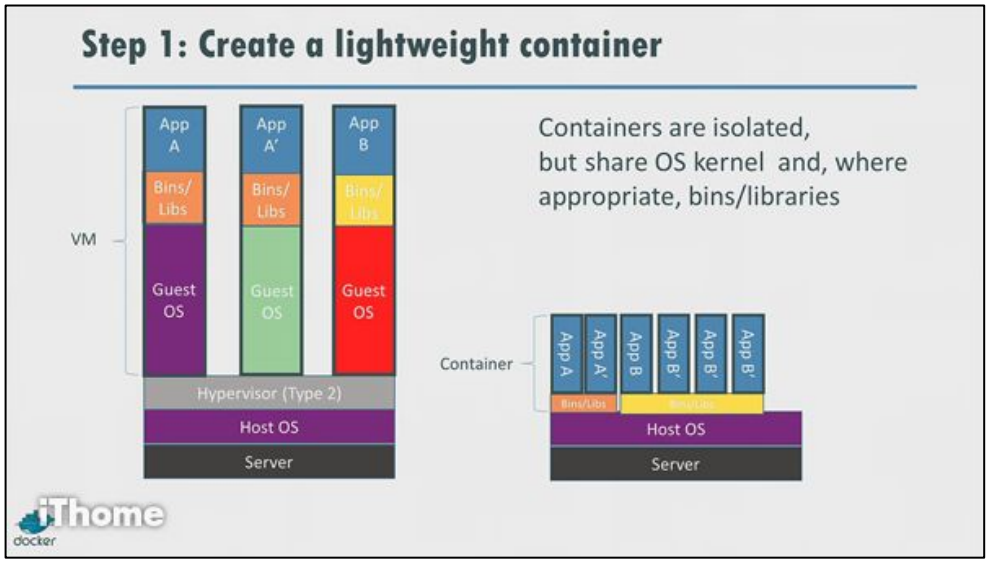
維運人員(SE)：我把你給的安裝包放在客戶的電腦安裝，怎麼跑不起來。

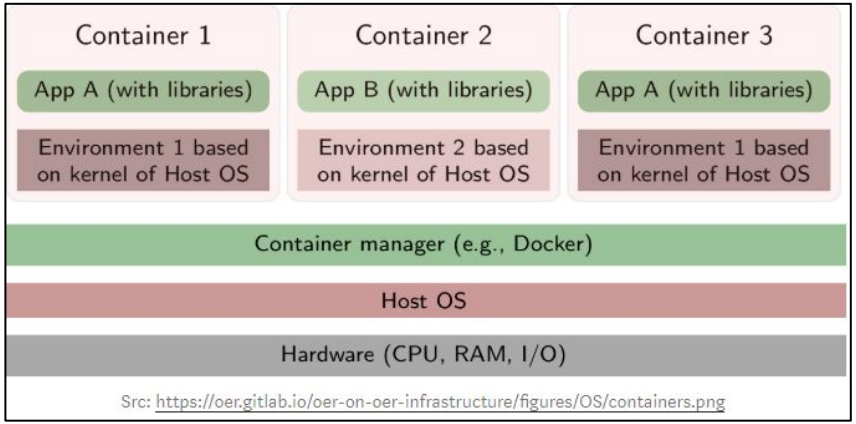
開發人員(RD)：怎麼可能！在我的電腦執行沒問題呀！

這情境大家一定並不陌生，有可能開發人員可能漏包程式檔案，也有可能是維運人員遺漏參數設定，最慘的情境是客戶的電腦做了特殊的設定讓所有人白忙一場...。許多公司往往會消耗大量的人力和時間在解決開發與維運之間的鴻溝。

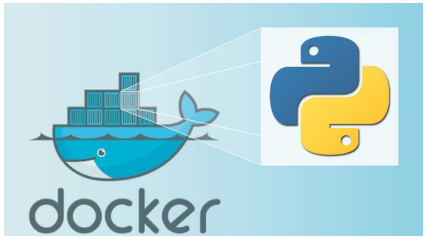
What is Docker

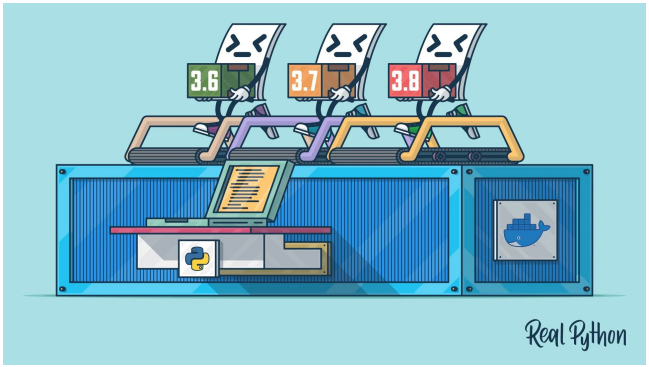
輕量級虛擬化Container技術則是一種以應用程式為中心的虛擬化技術
不需要安裝Guest OS

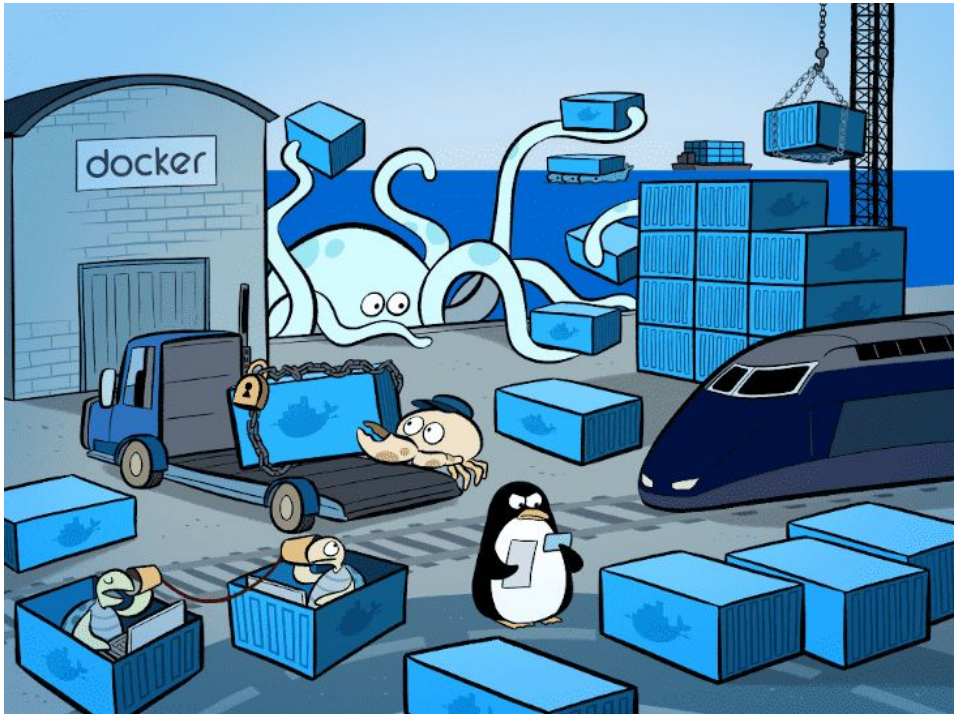


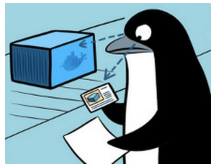


What is Docker









Docker vs VM

Virtual Machine:

- 需要安裝作業系統
- VM 裡面的作業系統開機需要花一點時間開機
- 完全的把系統的硬體資源隔離
- 佔用硬碟的容量較大

Docker:

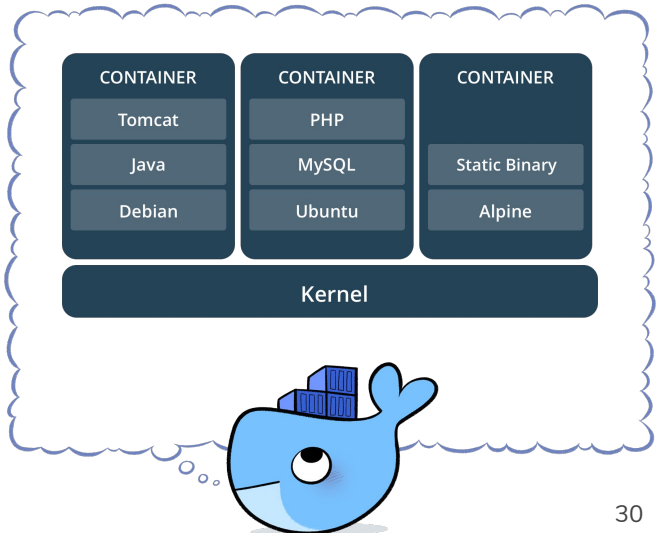
- 直接從 Docker Hub Pull 作業系統的 Image
- 不用開機，啟動速度比 VM 快
- 底層還是使用作業系統的 Kernel
- 佔用硬碟的容量較小
- 配合Linux的多工，非常方便

功能\平台	容器	虛擬主機
開機載入速度	秒為單位	分鐘為單位
硬碟容量	MB 為單位	GB 為單位
相容性	接近原生	受限於虛擬技術支援性
單一主機同時運行	最高可到數千個	最多可到數十個
隔離性	完全隔離	完全隔離

How To Install Docker

- **如果是Windows的使用者，Docker安裝可能受限於以下幾點**
  - Windows 10 64-bit: Pro, Enterprise, or Education (Build 16299 or later).
  - For Windows 10 Home should be version 2004 or higher

- **如果是Mac的使用者，Docker的安裝是超乎想像的簡單**
  - 下載 install package，並安裝完成
  - open terminal, and press “docker run hello-world”



- **[Docker hub]: 雲端儲存空間，存放許多好用的docker images**
  - Docker Hub
- **[Docker cheat sheet]**
  - https://1drv.ms/b/s!Aiwtjhj5fofrk8tQBTZ6wZzRpR0yqQ

Docker 重點名詞解釋

- Layer: 層，一組唯讀的文件或命令，描述如何在容器下設置底層系統。層 (Layer) 構建在彼此之上，每個層代表對文件系統的更改
- **Image: 映像檔，一個不可變的層 (Layer)，形成容器的基礎**
- **Container: 容器，可以作為獨立執行應用程式的映像檔實例。容器具有可變層，該可變層位於映像檔的頂部並且與底層相分離**
- Registry: 註冊伺服器，用於散佈 Docker 映像檔的存儲系統
- Repository: 倉庫，一組相關相同應用程式但不同版本的 Docker 映像檔儲存倉庫

倉庫 (repository) 公司

映像檔 (images) 事業群

容器 (containers) 部門

基本Docker指令

開發 Docker 容器相關：

- `docker create [image]`: Create a new container from a particular image.
- `docker login`: 登錄 Docker Hub 或私有的 Docker 註冊伺服器
- `docker build`: 建立一個新的映像檔 doc
- **`* docker pull [image]`: 從倉庫取得所需要的映像檔**
- **`* docker push [username/image]`: 把自己建立的映像檔上傳到Docker Hub中來共享**
- `docker search [term]`: Search the Docker Hub repository for a particular term.
- `docker tag [source] [target]`: 為目標映像檔新增標籤或匿名 (alias)

基本Docker指令

使用 Docker 工具:

- `docker history [image]` : Display the history of a particular image.
- **`* docker images`: 顯示所有儲存在本機的映像檔 doc**
- `docker inspect [object]` : Display low-level information about a particular Docker object.
- **`* docker ps`: 列出當前正在運行的所有容器 doc**
- `docker version`: 顯示系統上當前安裝的 Docker 版本
- **`* docker kill [container]`: Kill a particular container. doc**
- `docker kill $(docker ps -q)` : Kill all containers that are currently running.
- **`* docker rm [container]`: 移除本地端未執行的容器 doc**
- `docker rm $(docker ps -a -q)` :移除本地端所有未執行的容器
- **`* docker rmi [image]` 移除本地端的映像檔 doc**
- `docker container prune` 移除所以未執行的容器 doc

基本Docker指令

執行 Docker 容器相關:

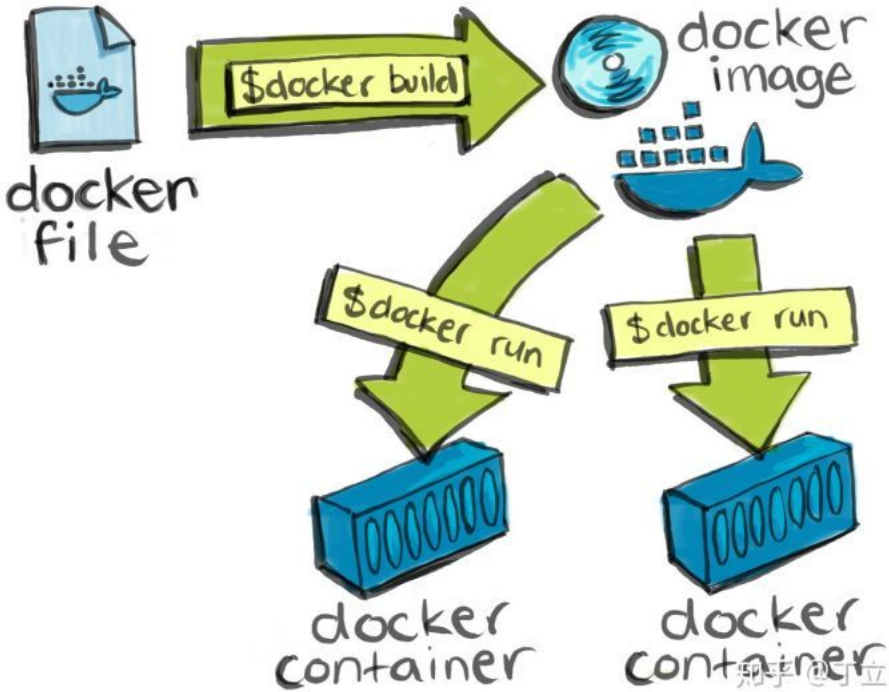
- `docker start [container]` : 啟動一個已經停止的容器
- `docker stop [container]` : 停止一個執行中的容器
- `docker exec -it [container] [command]` : Run a shell command inside a particular container.
- `docker run [image]` 啟動容器 doc
- `docker run -it -image [image] [container] [command]` : 建立並啟動容器，並在內部執行指令
- **`* docker run -it --rm -image [image] [container] [command]`: 建立並啟動容器，並在內部執行指令，指令完成後自動移除此容器**
- `docker pause [container]` : Pause all processes running within a particular container.

```
$ sudo docker run --rm -it \
-v /export/EC1680U/kentchen:/export/EC1680U/kentchen \
-v /mnt/NFS/EC2480U-P/scratch/QCResults/fastq:/mnt/NFS/EC2480U-P/scratch/QCResults/fastq \
-v `pwd`:`pwd` \
-w `pwd` kentchendocker/trinityrnaseq:latest \
<your_command>
```

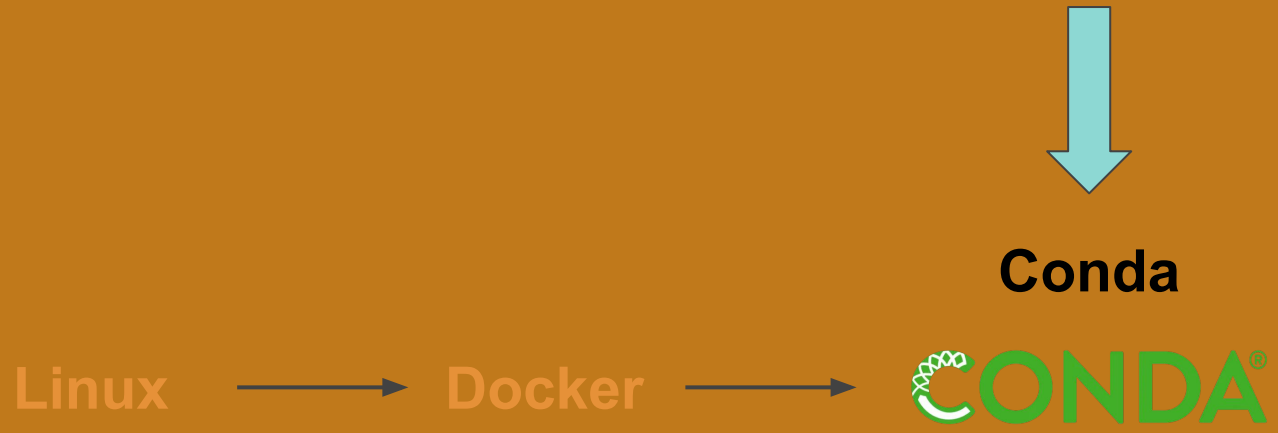
Docker實際操作範例

郭韋廷Docker操作範例

影片網址 : http://goo.gl/rO2NIX



https://zhuanlan.zhihu.com/p/80528411


Conda
Linux
Docker
CONDA

What is Conda

Anaconda 是一家美國的軟體公司, 該公司製作了一套Anaconda Python Distribution

- 便於安裝許多流行的科學、數學、工程、數據分析的 Python 模組
- 開源和免費
- 跨平台支持:Linux、Windows、Mac
- 支持 Python 版本切換, 方便建立不同的虛擬開發環境

註:環境 (environment) 是 Anaconda 中虛擬的概念, 在某個環境中可以指定某個版本的Python 軟體和相關的套件, 並且**可快速在不同環境中切換**, **環境間不會互相干擾**。

Powerful of Docker + Conda

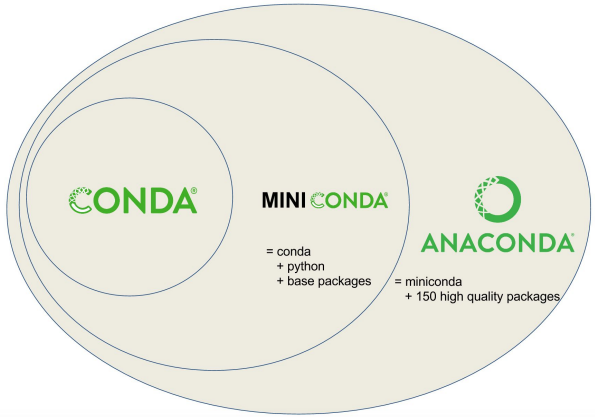
https://towardsdatascience.com/conda-pip-and-docker-ftw-d64fe638dc45

How to install miniconda

https://docs.conda.io/en/latest/miniconda.html

$ wget https://repo.anaconda.com/miniconda/Miniconda3-latest-MacOSX-x86_64.sh

$ bash Miniconda3-latest-MacOSX-x86_64.sh



Conda 基本指令

Pre-configuration

- $ conda config --add channels defaults
- $ conda config --add channels bioconda
- $ conda config --add channels conda-forge

Tool installation

- $ conda create --name <environment>
- $ conda activate <environment>
- $ conda install <tool>

What is the “**tool name**” in conda channel ???
Try to Google It !

Conclusions

- How to be a Bioinformatician
  - Biology, Programming and Statistics
- How to start in Bioinformatics
  - Basic Linux bash skill
- How to build an undisturbed Bioinformatics environment
  - VM / Docker (system build)
  - Conda (bioinformatics tool package)
  - Basic programming skill to pipe the tools