

Docker and openSUSE workshop 101

黃俊宏 sakana / Max

<https://goo.gl/Jui4oW>



Who is Max?

- Study Area 小雑工
- GNOME Foundation Member
- GNOME.Asia Committee Member
- openSUSE members

工商服務

- openSUSE x StudyArea x docker.Taipei x moby.Taipei
 - StudyArea
 - FB <https://www.facebook.com/sataiwan/>
 - openSUSE Taiwan
 - FB <https://www.facebook.com/groups/opensuse.zh/>
 - openSUSE KKTIX <http://opensuse-tw.kktix.cc/>
 - docker.Taipei
 - FB <https://www.facebook.com/groups/docker.taipei/>
 - Moby.Taipei
 - FB <https://www.facebook.com/groups/moby.taipei>



What is openSUSE project?

openSUSE是一個由SUSE所贊助的社群。

以推廣Linux 為宗旨，openSUSE.org計畫讓每個人可以免費而且輕易的使用世界上最好用的Linux 發行版本--SUSE Linux。同時，openSUSE也提供Linux開發人員和其支持者使用Linux所需的一切。

openSUSE計畫的目標：

- 打造SUSE Linux 成為最容易取得和使用的Linux發行版本，並成為最廣泛使用的開放原始碼平台。
- 提供良好的合作環境使SUSE Linux不論對新手老手而言都是世界上最好的Linux發行版本。
- 藉由大幅度地簡化和開放開發過程使 SUSE Linux成為程式設計師和Linux駭客們最喜愛的工作平台。

openSUSE 特色

- YaST

- openSUSE提供YaST(Yet another Setup Tool)實現方便的系統管理。YaST的功能強大,幾乎可以配置系統的各個方面:包括軟體管理、硬體設置、網路設置、系統服務、安全設置等諸多功能。這些功能的引入,大大簡化了配置系統的過程。

- OBS

- Build Service編譯服務,提供了openSUSE、Fedora、Ubuntu、Archlinux ...的全面編譯支援。也為openSUSE帶來了數不清的應用程式套件。
- <https://software.opensuse.org/search>

- openSUSE Tumbleweed

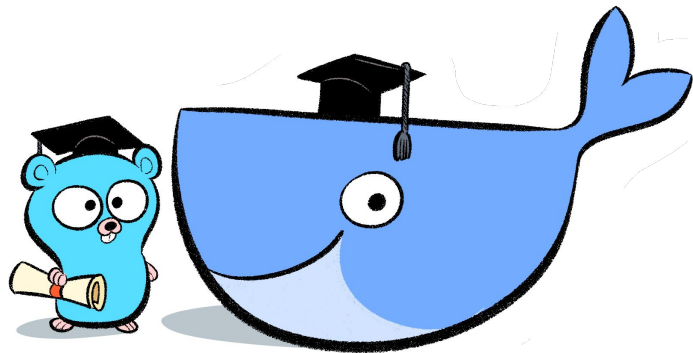
- 新的Tumbleweed 計劃旨在提供一個「滾動升級」的openSUSE 版本,滾動升級的版本即不再像openSUSE的常規版本(Leap)受限於版本開發周期,可以始終使用最新穩定版本的全部軟體。該計劃適合想使用最新的,又是穩定版的軟體的openSUSE 用戶。

Agenda

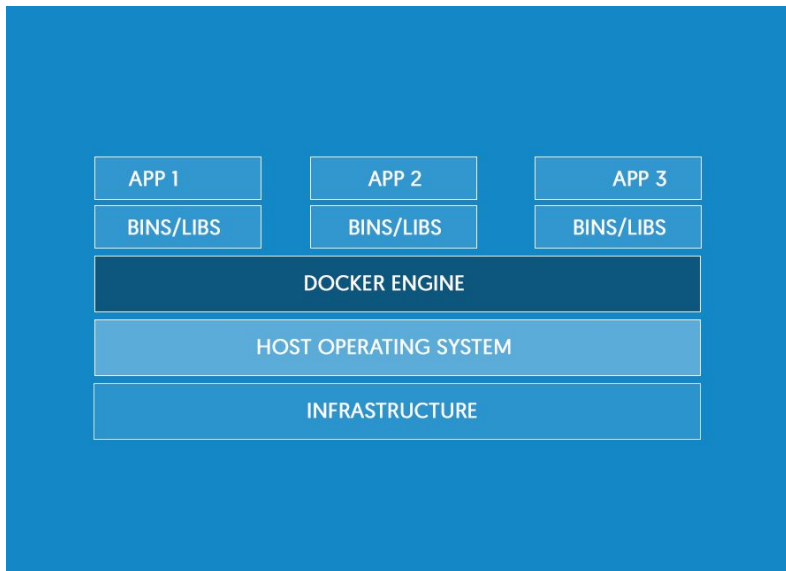
- Docker 介紹
- 安裝與設定Docker 方式
- Docker 基礎操作
- 執行Docker容器
- 建立Docker image
- YaST Docker 模組介紹

Docker 介紹

- 什麼是 Docker?
 - Docker 專案誕生於2013年，基於Go語言實作，目標是實作輕量級的作業系統虛擬化解決方案。Docker 的基礎是 Linux 容器(LXC)等技術。by [Docker —— 從入門到實踐](#)
 - Docker是一個開放原始碼軟體專案，讓應用程式布署在軟體容器下的工作可以自動化進行，藉此在Linux作業系統上，提供一個額外的軟體抽象層，以及作業系統層虛擬化的自動管理機制 [1]。by [wikipedia](#)

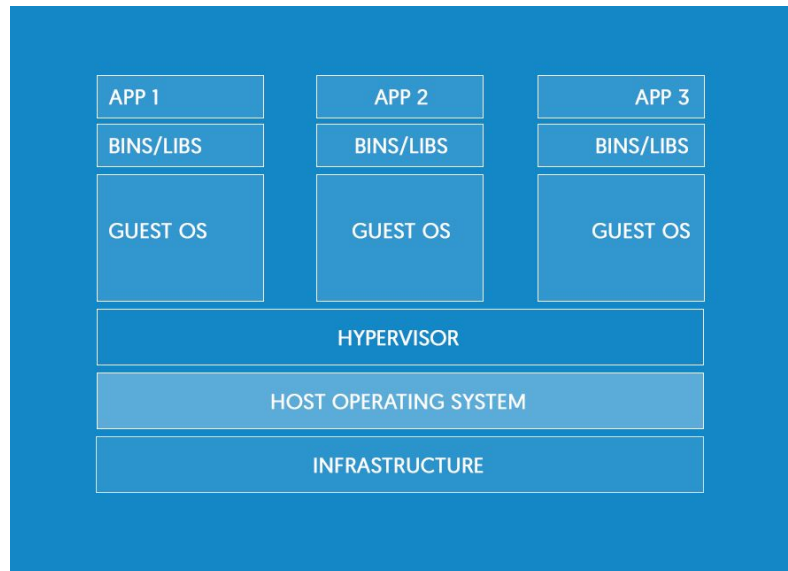


Docker vs VM



CONTAINERS

Containers include the application and all of its dependencies --but **share the kernel with other containers**, running as isolated processes in user space on the host operating system. Docker containers are not tied to any specific infrastructure: they run on any computer, on any infrastructure, and in any cloud.

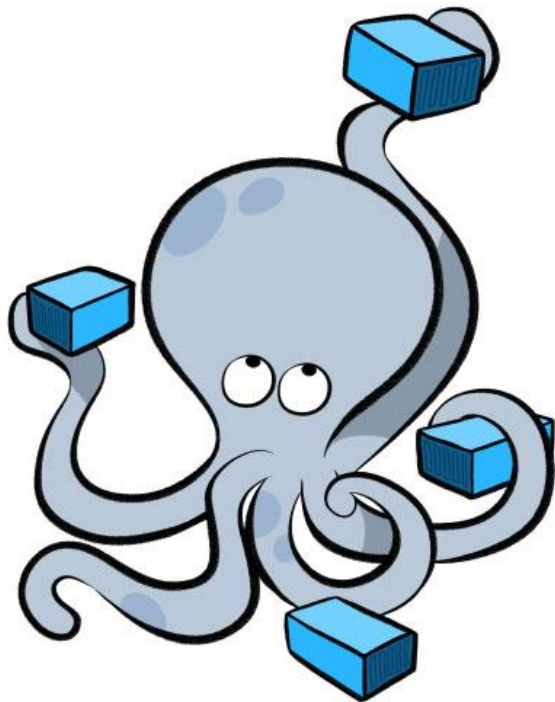


VIRTUAL MACHINES

Virtual machines include the application, the necessary binaries and libraries, and an **entire guest operating system** -- all of which can amount to **tens of GBs**.

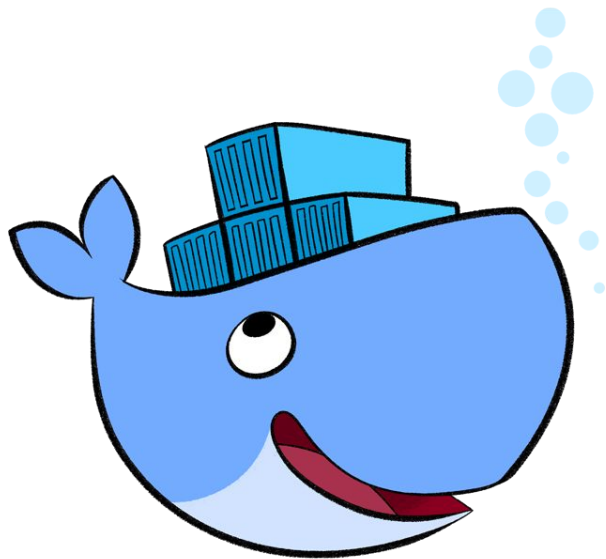
Docker 特性

- 輕量
 - Container (容器) 不需要消耗太多記憶體, 只需要程序所需的用量。
- 快速
 - 相較於VM, 啟動Container就像啟動一般程序一樣快, 只需要幾秒鐘的時間。
- 易於分享與重現環境
 - Dockerfile 可以基於已經建構好的 Docker image, 建立新的容器, 並確保每次的Container 擁有同樣的環境與行為。



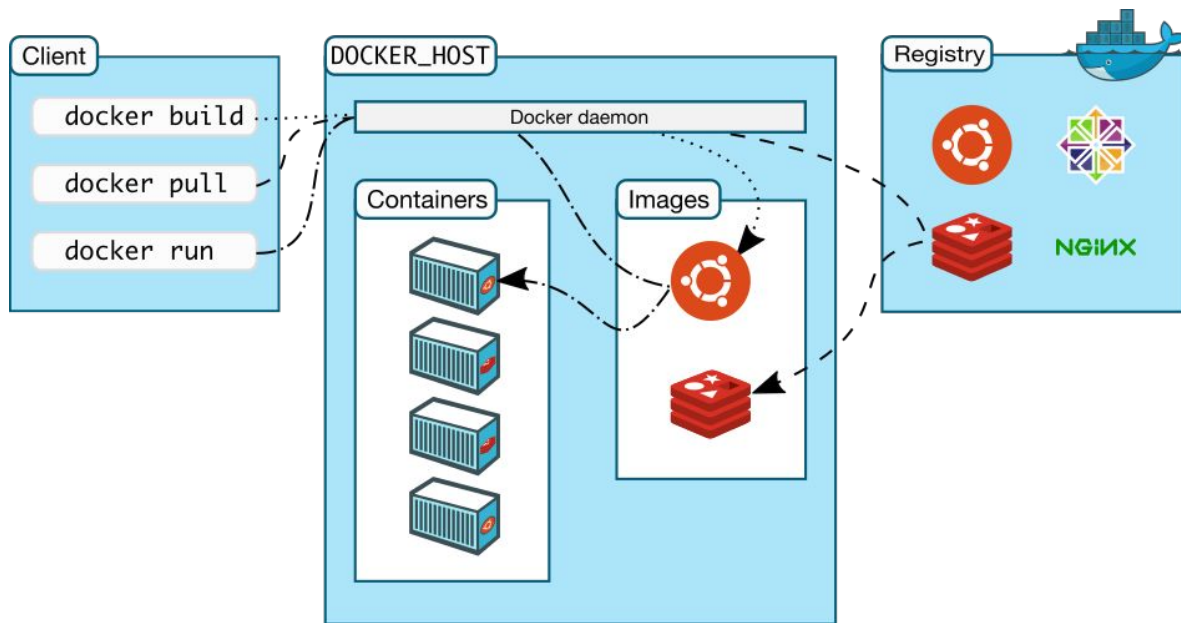
Docker basic

- 映像檔(Image)
 - 用來建立容器的唯讀的範本。
- 容器(Container)
 - 基於映像檔, 使用 docker run 來執行的程序或是服務就稱為 container。可以想成VM的 instance, 以隔離的方式共享主機的核心。
- 倉庫(Registry)
 - 倉庫是 docker image 的集合場所。
 - 公開倉庫 (Public)
 - [DockerHub](https://docs.docker.com/engine/understanding-docker/)
 - 私有倉庫 (Private)
 - [Docker Registry](https://docs.docker.com/engine/understanding-docker/)



Docker 架構

- Docker 採用 Client - Server 架構。
- Client 透過 socket 或是 API 與 Docker daemon 溝通，建立或是取得 image 並執行 container。
- 如果要取得的 image 本地端沒有就會到 Registry 搜尋與取得。



Hands on 預備

如何確認有安裝 Docker 環境

```
# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
------------	-----	----------	---------	--------------

如果沒有.....

- 確認是否啟動 docker
- 連安裝都沒有請前往 <https://docs.docker.com/engine/installation/>
 - Windows - Docker for Windows (Win 10 Pro + 64bits)
 - Mac - Docker for Mac (OS X 10.10.3 Yosemite or newer)
 - Linux - 直接安裝

Docker 官方練習資源

- play-with-docker.com

Docker 映像檔管理

- Lab: 下載映像檔並觀察
 - docker pull
 - docker images
- Lab: 映像檔查詢與取得
 - docker search
 - docker pull

動手實作吧

https://github.com/sakanamax/SA_dockerReading/blob/master/workshop/docker_workshop_101_1_20170915.pdf

Demo / Hands on Lab

Docker 映像檔管理

- Lab: 建立 docker 映像檔-修改現有映像檔
 - docker ps
 - docker run
 - with --rm
 - without --rm
 - docker commit

動手實作吧

https://github.com/sakanamax/SA_dockerReading/blob/master/workshop/docker_workshop_101_2_20170915.pdf

Demo / Hands on Lab

Docker 映像檔管理

- Lab: 建立 docker 映像檔-使用Dockerfile
 - Dockerfile 撰寫
 - docker build
- Lab: 映像檔管理 - Tag 與刪除映像檔
 - docker tag
 - docker rmi
 - docker rm

動手實作吧

https://github.com/sakanamax/SA_dockerReading/blob/master/workshop/docker_workshop_101_3_20170915.pdf

Demo / Hands on Lab

Reference

- O'Reilly - Docker Cookbook
- Blog 文章 [sakananote](#)
- 整理的心智圖 [Mindmap](#)
 - https://github.com/sakanamax/SA_dockerReading
 - 心智圖推薦使用 [freemind](#)
- Github
 - <https://github.com/sakanamax>
- Docker hub
 - <https://hub.docker.com/u/sakana/>

Thanks You