

從零開始的 Docker 美好世界

Intro Docker by Hot Pot

Contents of this template

Here's what you'll find in this **Slidesgo** template:

- A slide structure based on a healthcare center presentation template, which you can easily adapt to your needs. For more info on how to edit the template, please visit **Slidesgo School** or read our **FAQs**.
- An assortment of pictures/illustrations that are suitable for use in the presentation can be found in the **alternative resources** slide.
- A **thanks slide**, which you must keep so that proper credits for our design are given.
- A **resources** slide, where you'll find links to all the elements used in the template.
- **Instructions for use**.
- Final slides with:
 - The **fonts and colors** used in the template.
 - A selection of **illustrations**. You can also customize and animate them as you wish with the online editor. Visit **Storyset by Freepik** to find more.
 - More **infographic resources**, whose size and color can be edited.
 - Sets of **customizable icons** of the following themes: general, business, avatar, creative process, education, help & support, medical, nature, performing arts, SEO & marketing, and teamwork.

You can delete this slide when you're done editing the presentation.

Table of Contents

01

Virtualization

介紹虛擬化技術

02

First Container

第一個 Docker Container

03

Hot Pot !

從火鍋理解 Docker
Command

04

First Compose

第一個 Docker Compose

William Mou

<https://www.mou.tw>



01

Virtualization

虛擬化技術



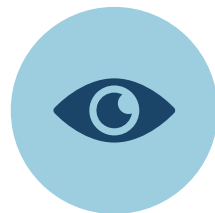


VM

慢

肥

少



Container

快

輕

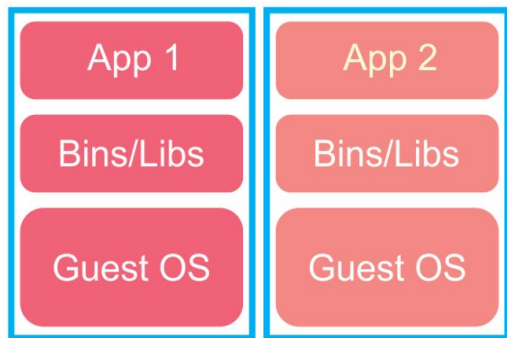
多

啟動

容量

數量

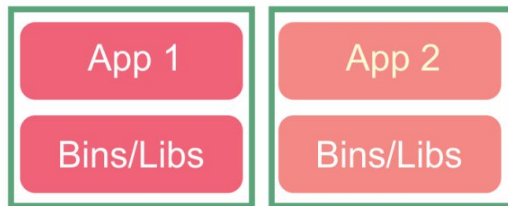
上面框框起來的 => **Guest**
下面沒被框起來 => **Host**



Hypervisor

Host OS

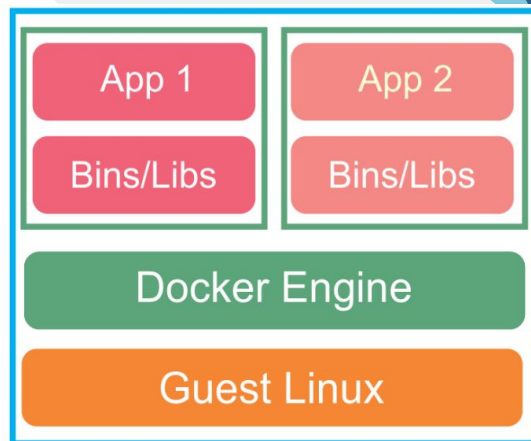
Virtual Machines



Docker Engine

Host Linux

Docker (Linux)



Docker Engine

Guest Linux

Hypervisor

Host OS

Docker (Non-Linux)

虛擬戶的好處

01 開發環境

快速獲得乾淨且統一的開發環境

02 環境隔離

Guest 與 Host 的環境在某層被隔離



降低成本 03

一台 Host 上可以有很多 Guest
小資學生最愛

部屬快速 04

方便複製 Guest
給不同單位使用

Docker 更多優點



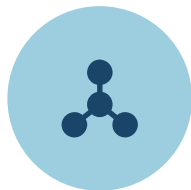
資源分配佳

VM 通常需要給定
固定的資源



直接硬體訪問

可以直接操作硬體
如 NV 的顯卡等



冗餘少

不用重複安裝
Guest OS



“Although containers themselves have almost no overhead, Docker is not without performance gotchas.

Docker’s NAT also introduces overhead for workloads with high packet rates.

These features represent a tradeoff between ease of management and performance and should be considered on a case-by-case basis.”

–2014 Study by IBM





02

First Container

第一個 Docker container



Docker 的說明網站？

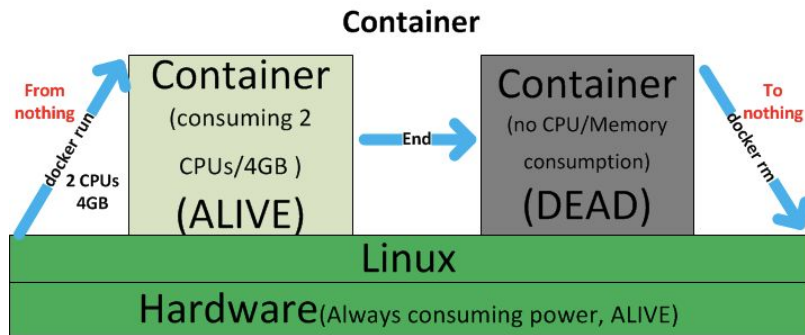
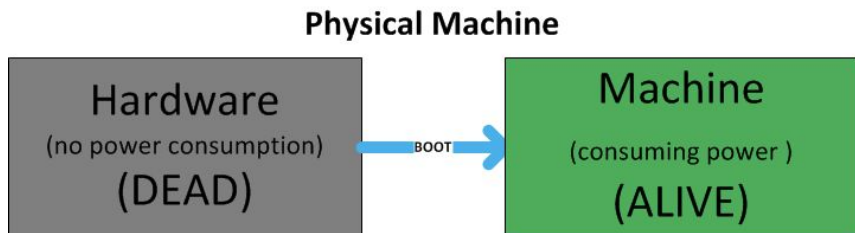
用 Docker 架設一個！

```
docker run -d -p 80:80 docker/getting-started
```

Then chick it -> <http://127.0.0.1:80>



Container 的生命週期



抓 pull

從網路上抓一個
別人寫好的環境下來

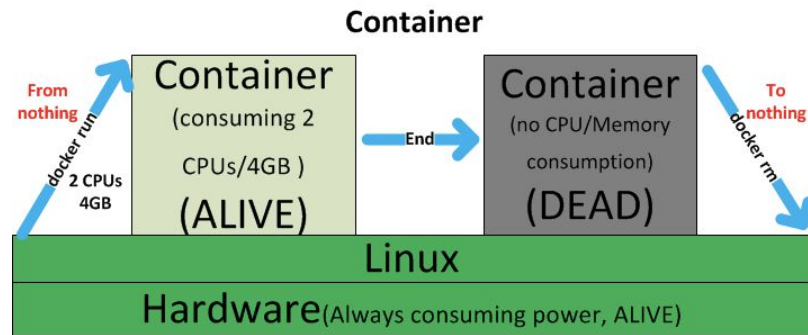
開 create

把這個環境啟動
並且設定 Port Forwarding

跑 start

執行別人寫好的程式
(網站伺服器)

剛剛的指令做了什麼



如何管理？

外面有 Host，裡面有 Guest
不同的作業系統該如何管理？

好不容易設定好一個容器
該如何分享給別人？

希望他人好使用
又不希望他人弄壞？

03 Hot Pot

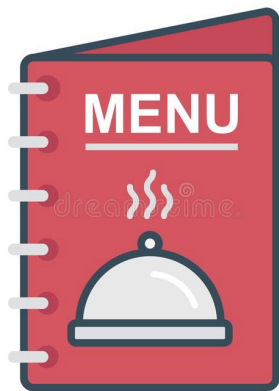
從火鍋理解 Docker 架構



你是一個火鍋迷

自幼自調鍋底、自訂醬料

**如果有一家火鍋店
可以
在菜單加上“自訂”鍋底**



菜單

- 養生蒙古鍋
- 大腸臭臭鍋
- 海鮮鍋

鍋底

- 養生蒙古鍋
 - 人蔘
 - 山藥
 - 紅棗



加購火鍋料

- 甜菜根
- 青椒
- ...



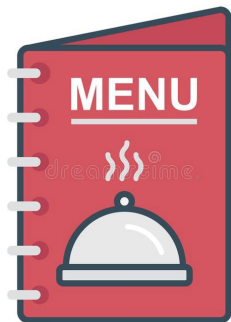
鍋底+火鍋料

- 甜菜根
- 青椒

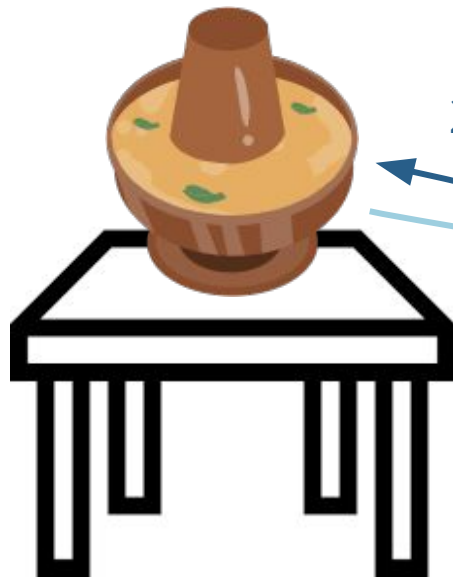


新菜單！

- 養生蒙古鍋
- 大腸臭臭鍋
- 海鮮鍋
- 蔬菜蒙古鍋 (News)



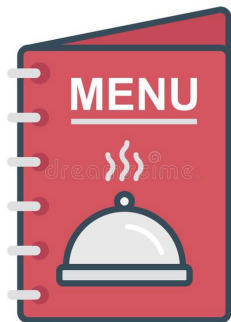
1. 上桌



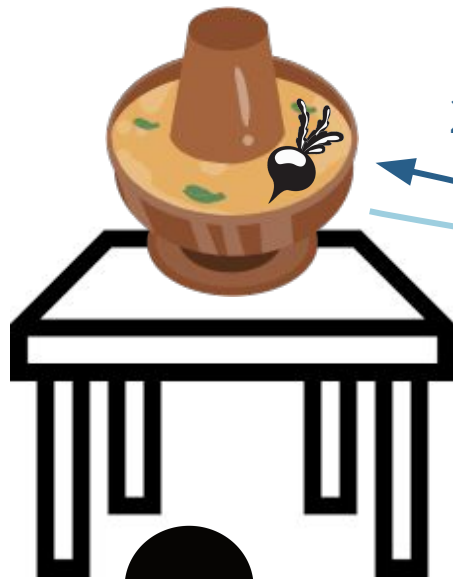
2. 開火



2-1. 熄火



1. 上桌

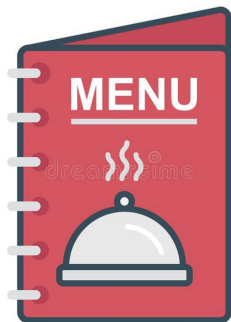


2. 開火

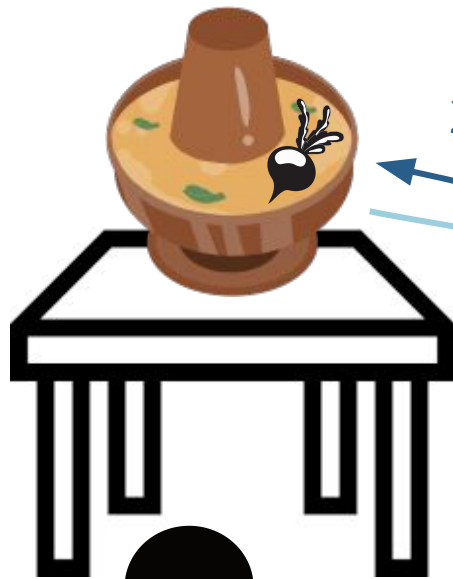
2-1. 熄火

3. 加料





1. 上桌



2. 開火

2-1. 熄火

3. 加料

4. 自訂新菜單

- 養生蒙古鍋
- 蔬菜蒙古鍋
(News)

鍋底 Image



1. 可從菜單上獲得
2. 可饕客臨時建立
3. 定好之後就唯獨
4. 開鍋後會變成火鍋

火鍋

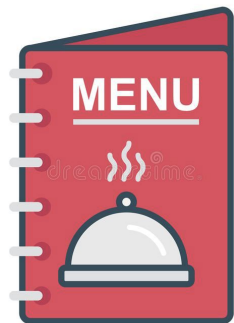
Container



1. 由鍋底建立
2. 可加菜、開火/關火
3. 沒存成菜單就消失
4. 一般客人都對火鍋操作

菜單

Docker Hub

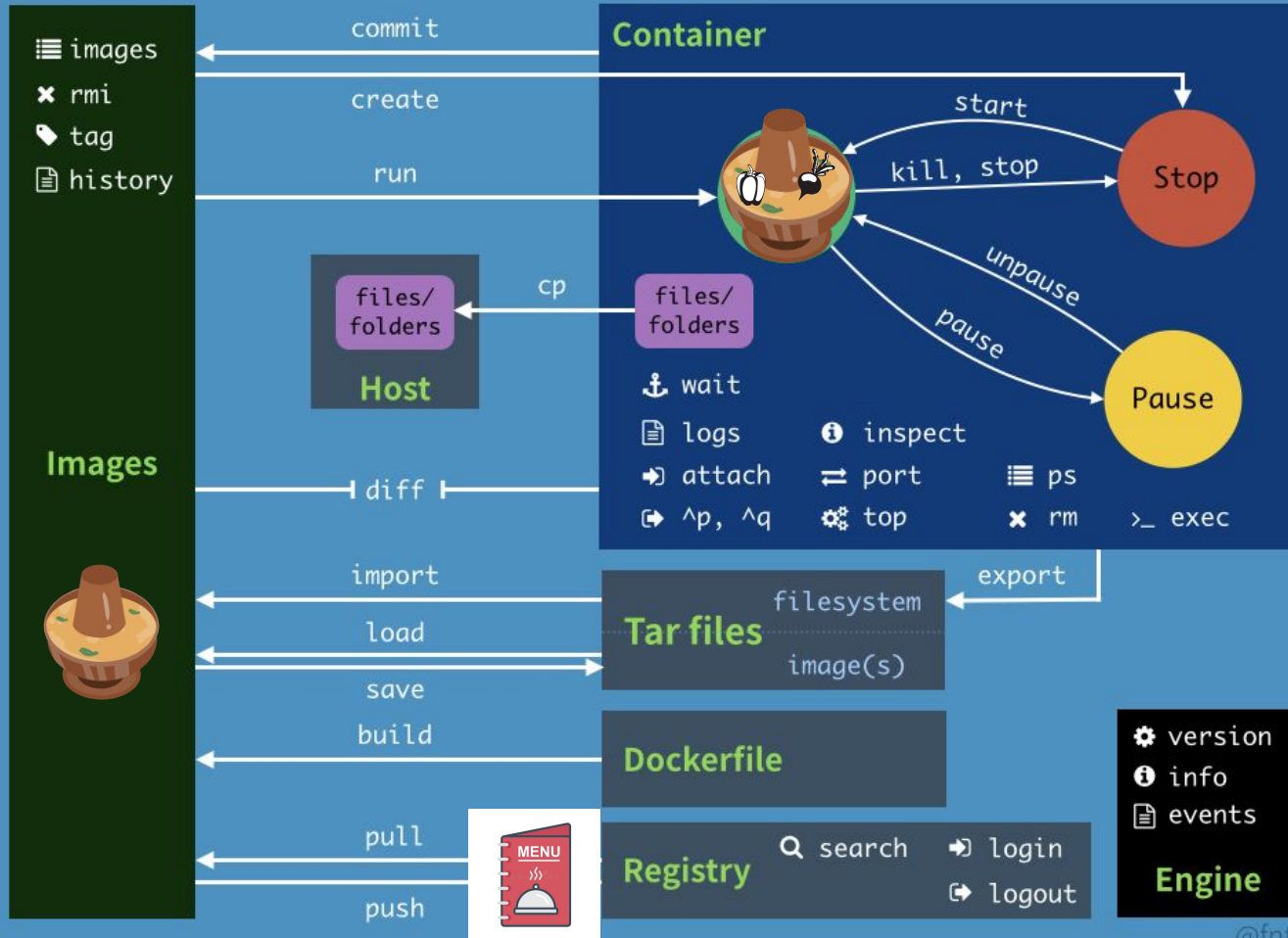


1. 儲存眾多鍋底
2. 所有人都可看
3. 店家/饕客建立



鍋底 Image
火鍋 Container
菜單 Docker Hub

Docker Commands Diagram



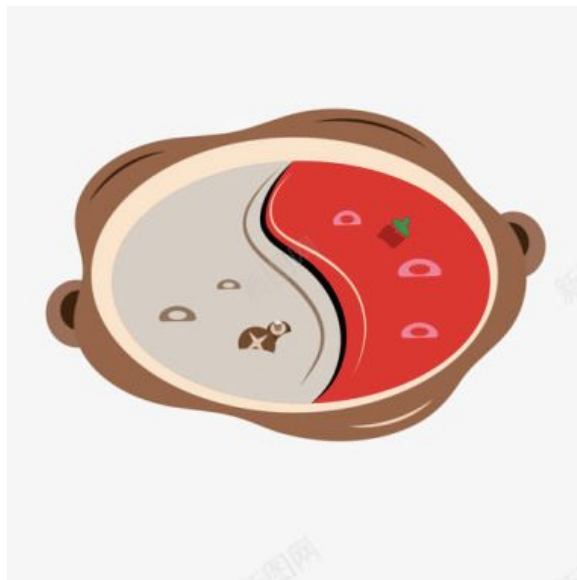
開始把火鍋變成 Docker 吧！

https://hackmd.io/-HXLcTaeRzy8kOJCp_v2kA

04

First Compose

老闆，開一鍋鴛鴦鍋！





安裝 Docker Compose

<https://docs.docker.com/compose/>

試試看 自架設 GitLab

- <https://docs.gitlab.com/omnibus/docker/#install-gitlab-using-docker-compose>

Install GitLab using Docker Compose

With [Docker Compose](#) you can easily configure, install, and upgrade your Docker-based GitLab installation:

1. [Install Docker Compose](#).
2. Create a `docker-compose.yml` file (or [download an example](#)):

```
web:
  image: 'gitlab/gitlab-ee:latest'
  restart: always
  hostname: 'gitlab.example.com'
  environment:
    GITLAB_OMNIBUS_CONFIG: |
      external_url 'https://gitlab.example.com'
      # Add any other gitlab.rb configuration here, each on its own line
  ports:
    - '80:80'
    - '443:443'
    - '22:22'
  volumes:
    - '$GITLAB_HOME/config:/etc/gitlab'
    - '$GITLAB_HOME/logs:/var/log/gitlab'
    - '$GITLAB_HOME/data:/var/opt/gitlab'
```

3. Make sure you are in the same directory as `docker-compose.yml` and start GitLab:

```
docker-compose up -d
```


Resources

Docker Command

- <https://joshhu.gitbooks.io/dockercommands/content/Containers/ContainersBasic.html>

Instructions for use (free users)

In order to use this template, you must credit [Slidesgo](#) by keeping the Thanks slide.

You are allowed to:

- Modify this template.
- Use it for both personal and commercial purposes.

You are not allowed to:

- Sublicense, sell or rent any of Slidesgo Content (or a modified version of Slidesgo Content).
- Distribute this Slidesgo Template (or a modified version of this Slidesgo Template) or include it in a database or in any other product or service that offers downloadable images, icons or presentations that may be subject to distribution or resale.
- Use any of the elements that are part of this Slidesgo Template in an isolated and separated way from this Template.
- Delete the “Thanks” or “Credits” slide.
- Register any of the elements that are part of this template as a trademark or logo, or register it as a work in an intellectual property registry or similar.

For more information about editing slides, please read our FAQs or visit Slidesgo School:

<https://slidesgo.com/faqs> and <https://slidesgo.com/slidesgo-school>

Thanks!

Do you have any questions?

contact@mou.tw

+886 920 992 660

www.mou.tw