Docker 簡介

台灣樹莓派 <sosorry@raspberrypi.com.tw>

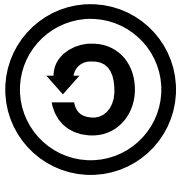
2016/11/26 @CYCU

CC (Creative Commons)

姓名標示 — 非商業性 — 相同方式分享

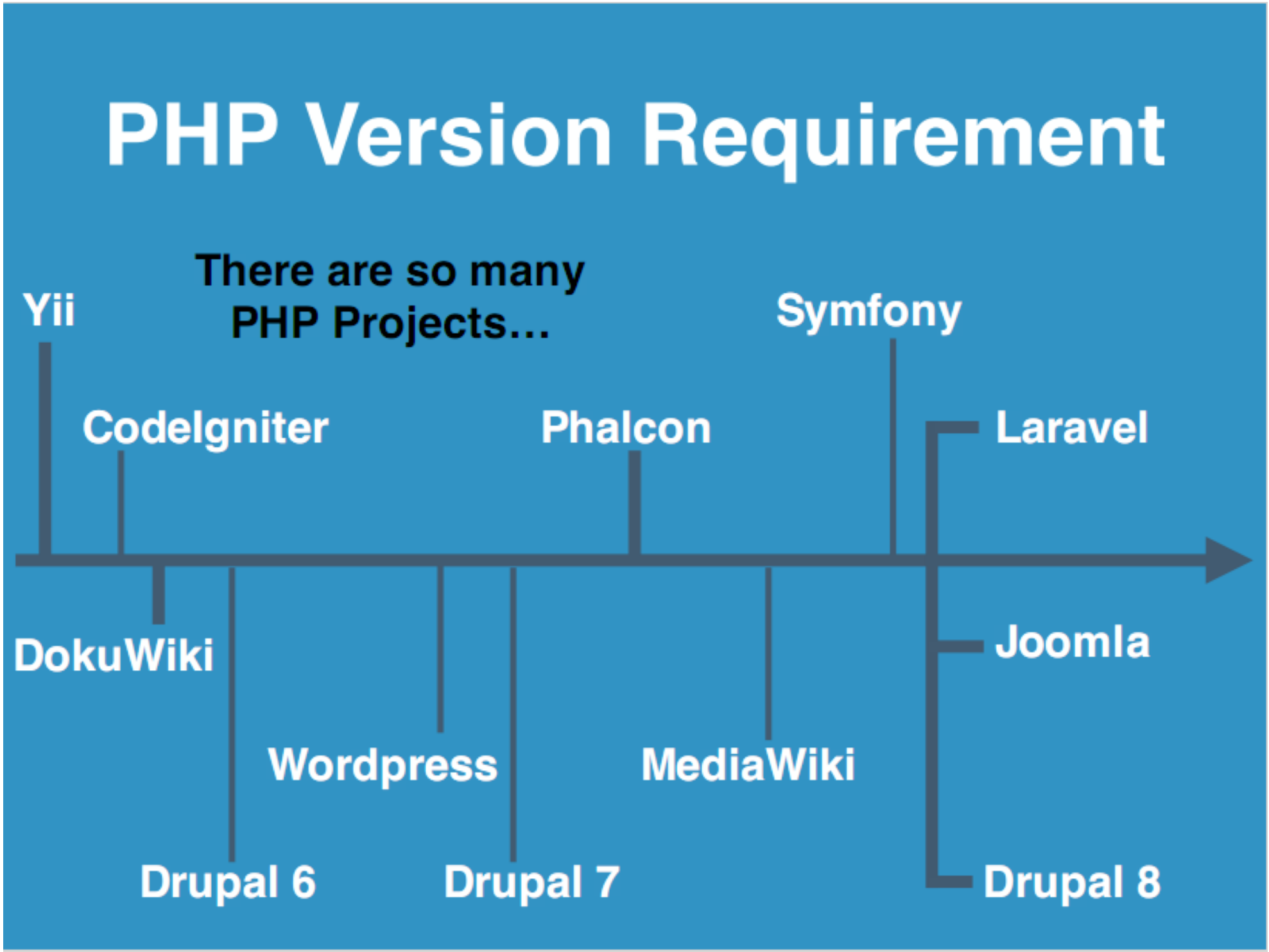
姓名標示 — 你必須給予 適當表彰、提供指向本授權條款的連結，以及 指出（本作品的原始版本）是否已被變更。你可以任何合理方式為前述表彰，但不得以任何方式暗示授權人為你或你的使用方式背書。

非商業性 — 你不得將本素材進行商業目的之使用。

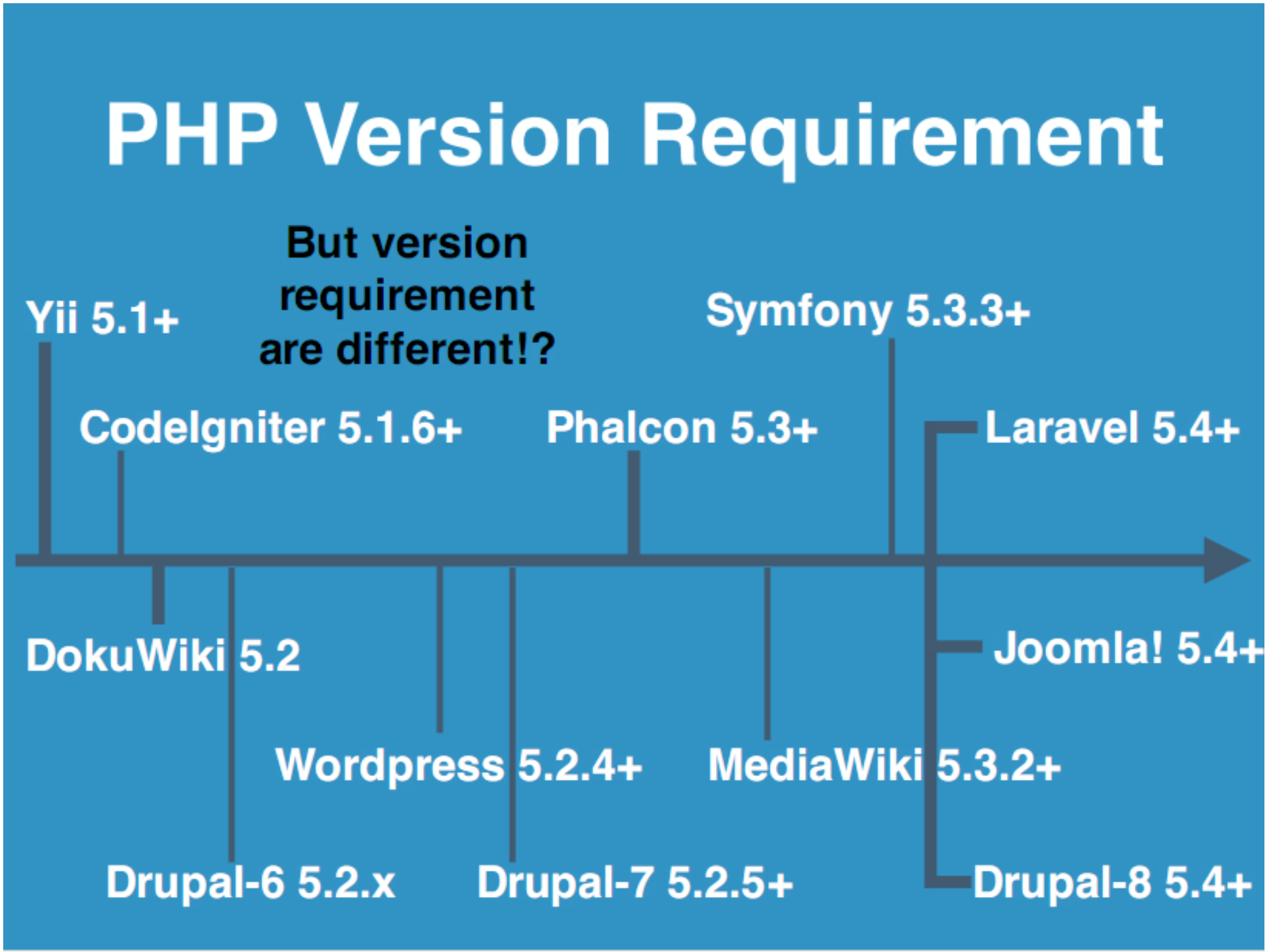


相同方式分享 — 若你重混、轉換本素材，或依本素材建立新素材，你必須依本素材的授權條款來散布你的貢獻物。

真實世界的問題：
不同的專案需要不同的環境設定
- 以 PHP 為例



http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker



http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker

Multifarious Extension

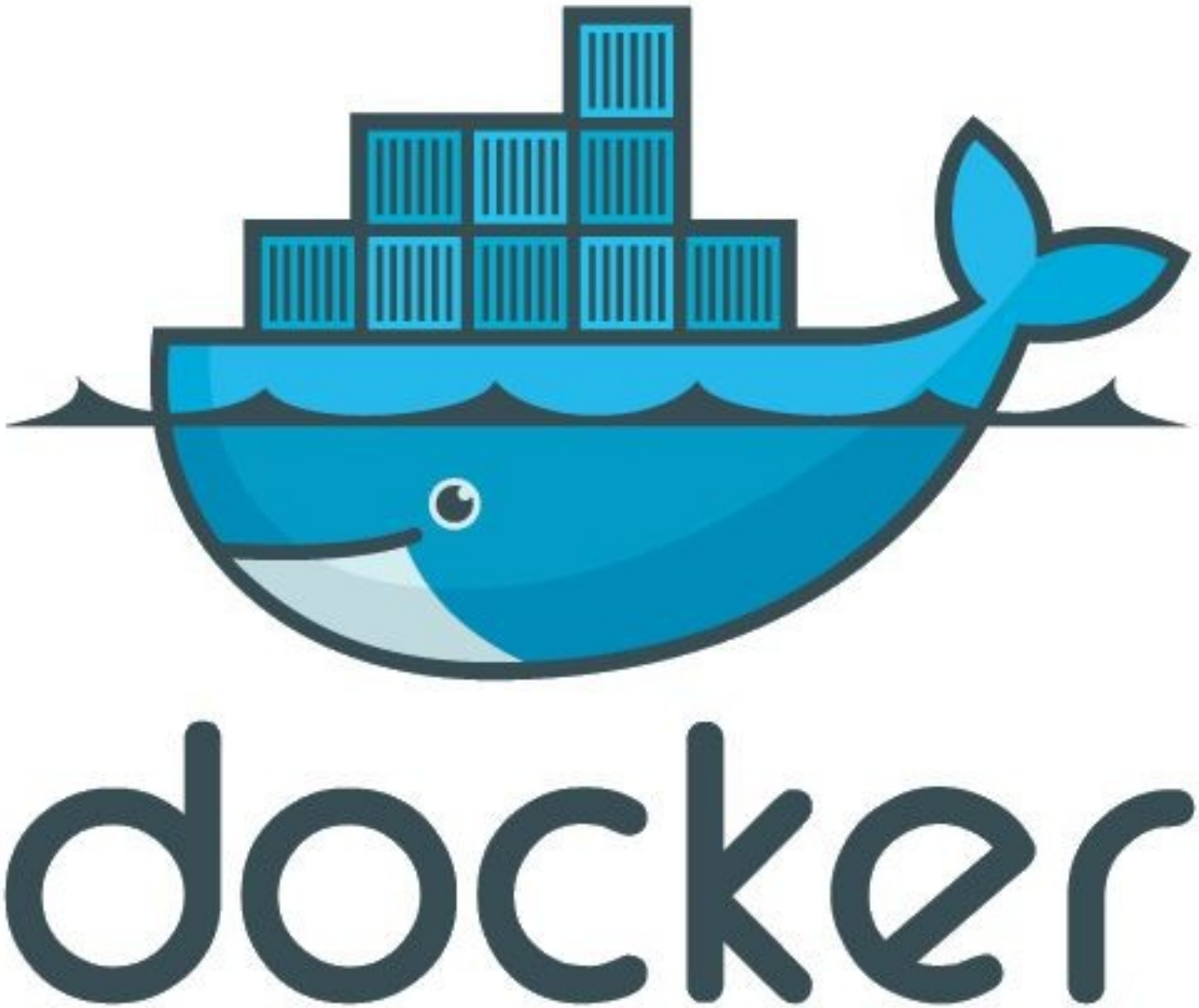
apxs2, bcmath, bz2, calendar, cgi, cli, ctype, dba, debug, dom, embed, exif, fileinfo, filter, fpm, ftp, gcov, gd, gettext, hash, iconv, icu, imap, intl, ipc, ipv6, json, kerberos, mbregex, mbstring, mcrypt, mhash, mysql, openssl, pcntl, pcre, pdo, pgsql, phar, posix, readline, session, soap, sockets, sqlite, tidy, tokenizer, xml_all, xmlrpc, zip, zlib

http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker

該如何解決問題？

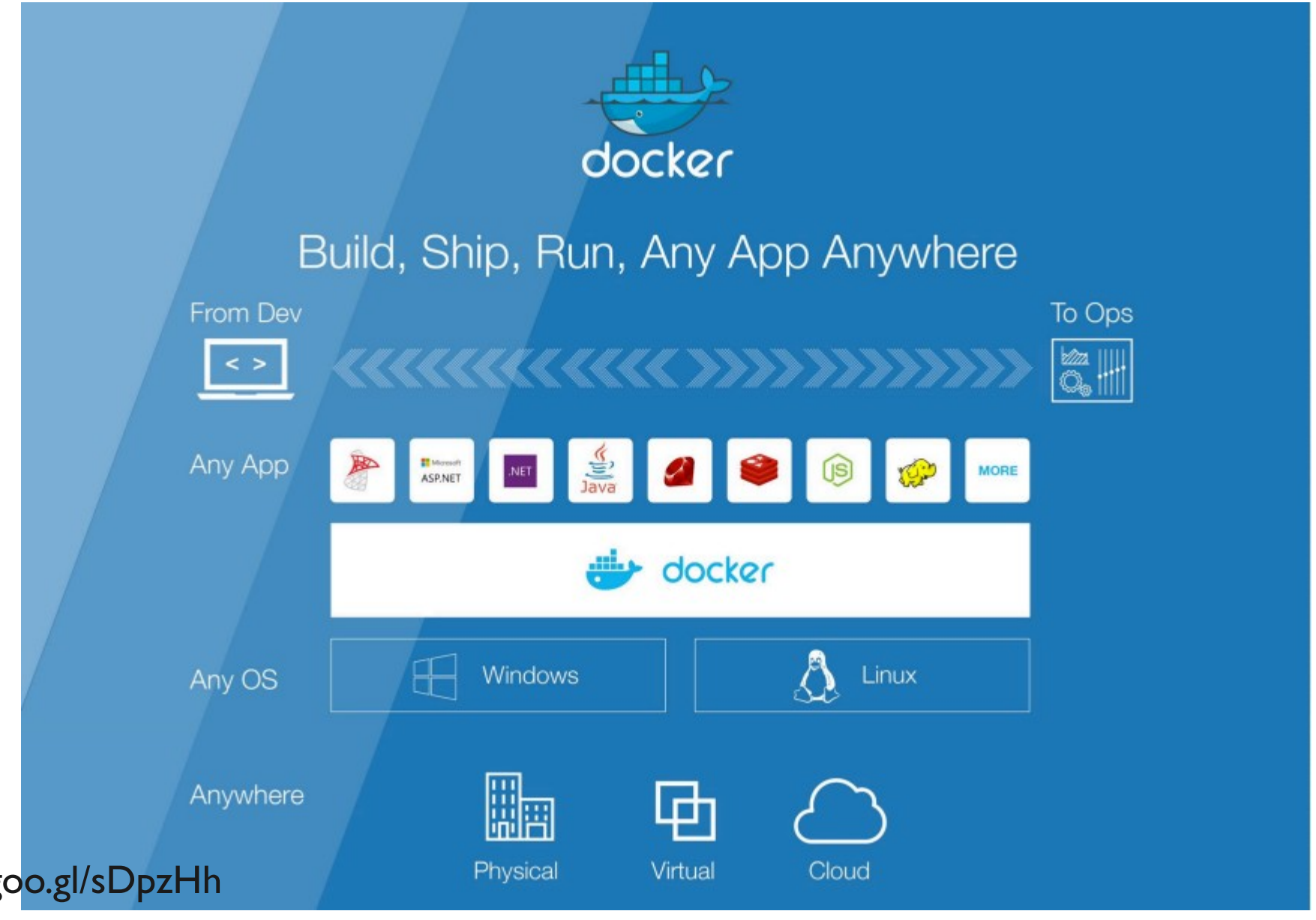
我們希望

- 能根據專案需求建立不同的環境
- 每一個專案環境都是獨立的，不互相干擾
- 環境上的套件或函式庫是可重製與可攜
- 能快速且輕量化的建置



https://www.ibm.com/blogs/bluemix/2015/06/ibm-containers-a-bluemix-runtime-leveraging-docker-technology

Docker 是什麼？



http://goo.gl/sDpzHh

Docker 是

- 基於 LXC(Linux Container) 的技術
- 不是 VM(Virtual Machine)
- 具封裝性 (Encapsulation)
- 具可攜性 (Portability)
- 輕量性 (Lightweight)

LXC

Linux Container 是一種作業系統層級的虛擬化技術 , 在共用的 Kernel 條件下 , 創造出應用程式的獨立沙箱執行環境

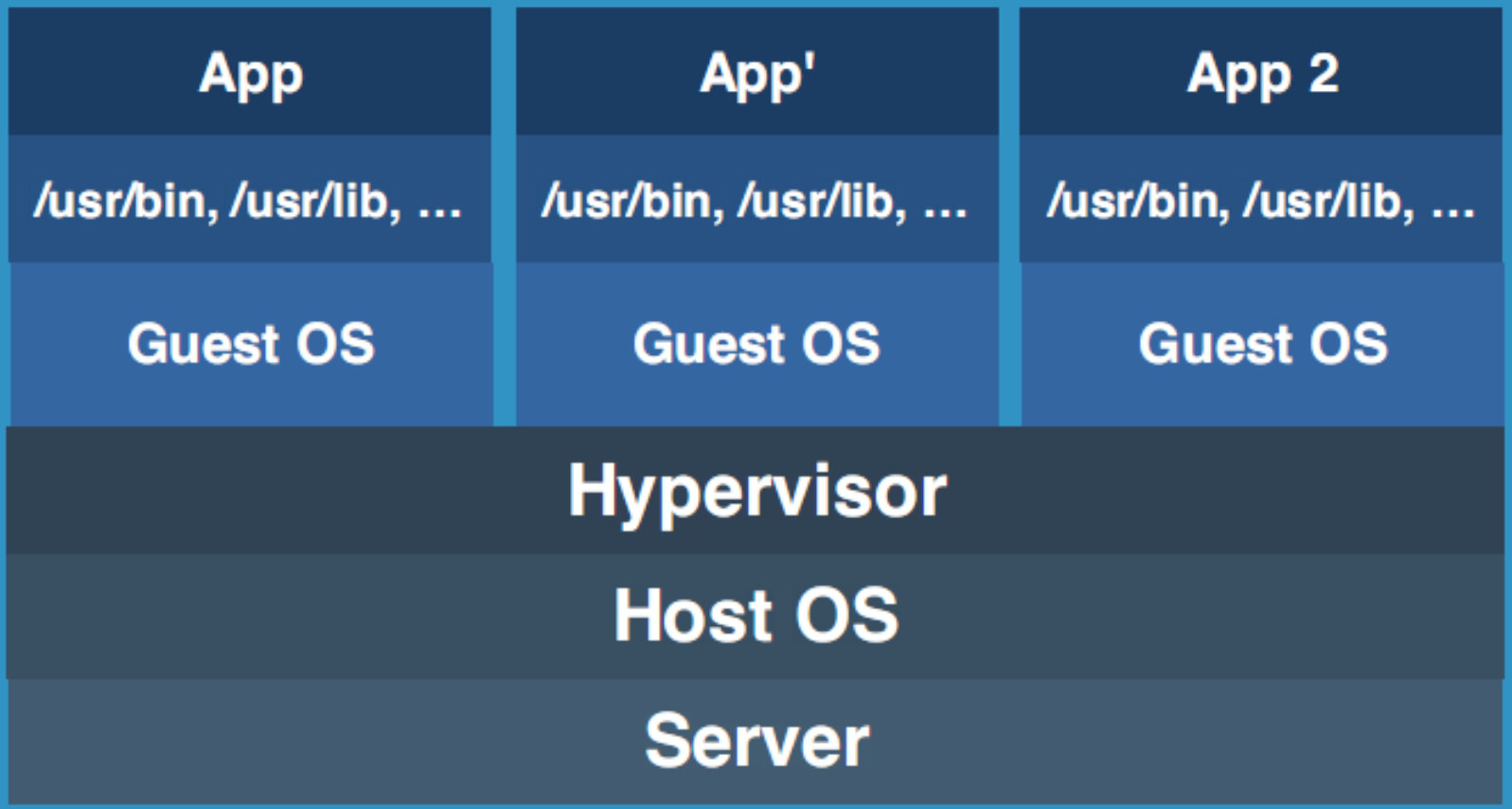
LXC

Kernel namespaces	Cgroups	Chroot
PID, mount, user, network, UTS, IPC	cpu, memory, disk I/O	File system

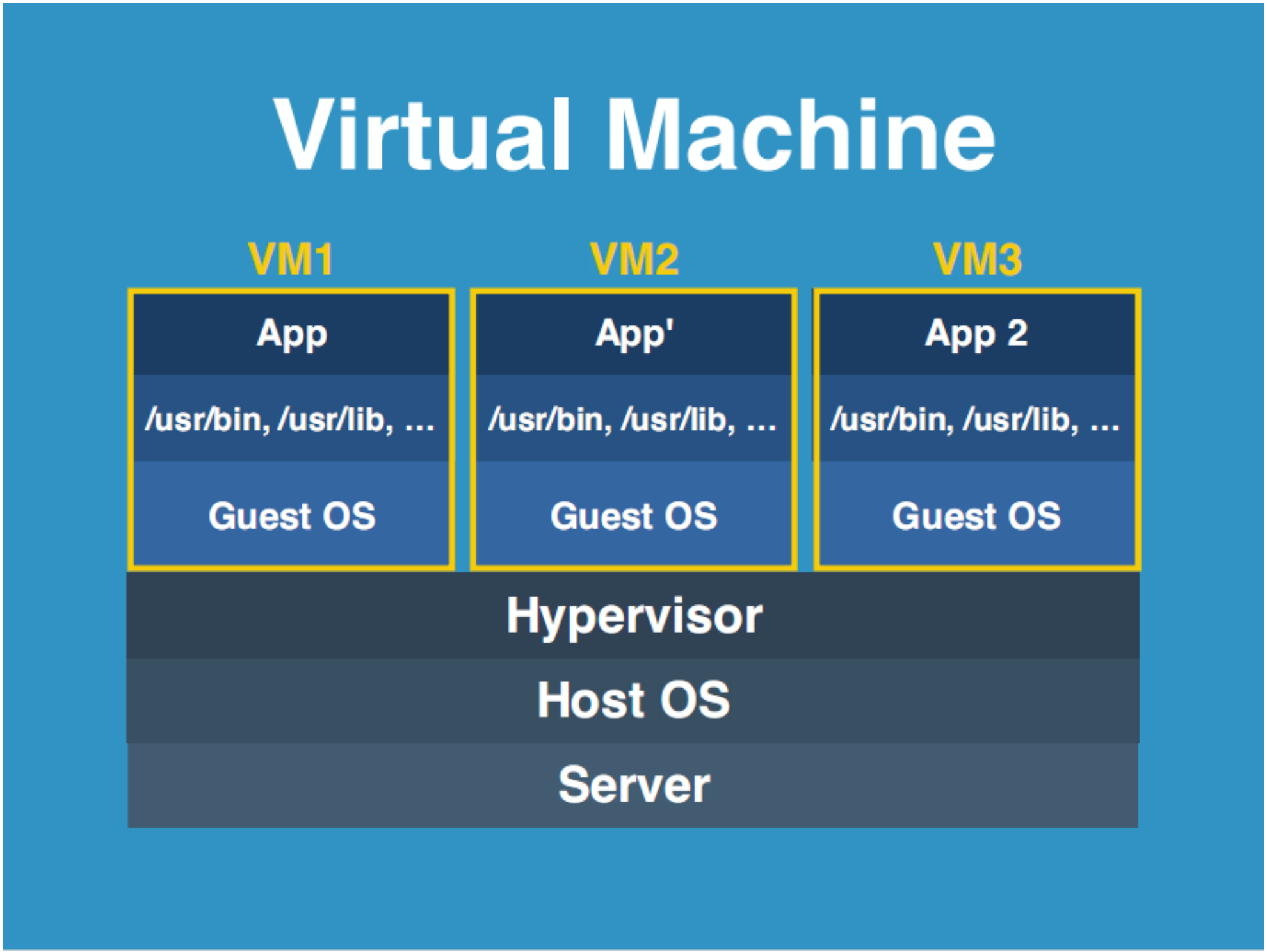
http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker

Container vs. Virtual Machine

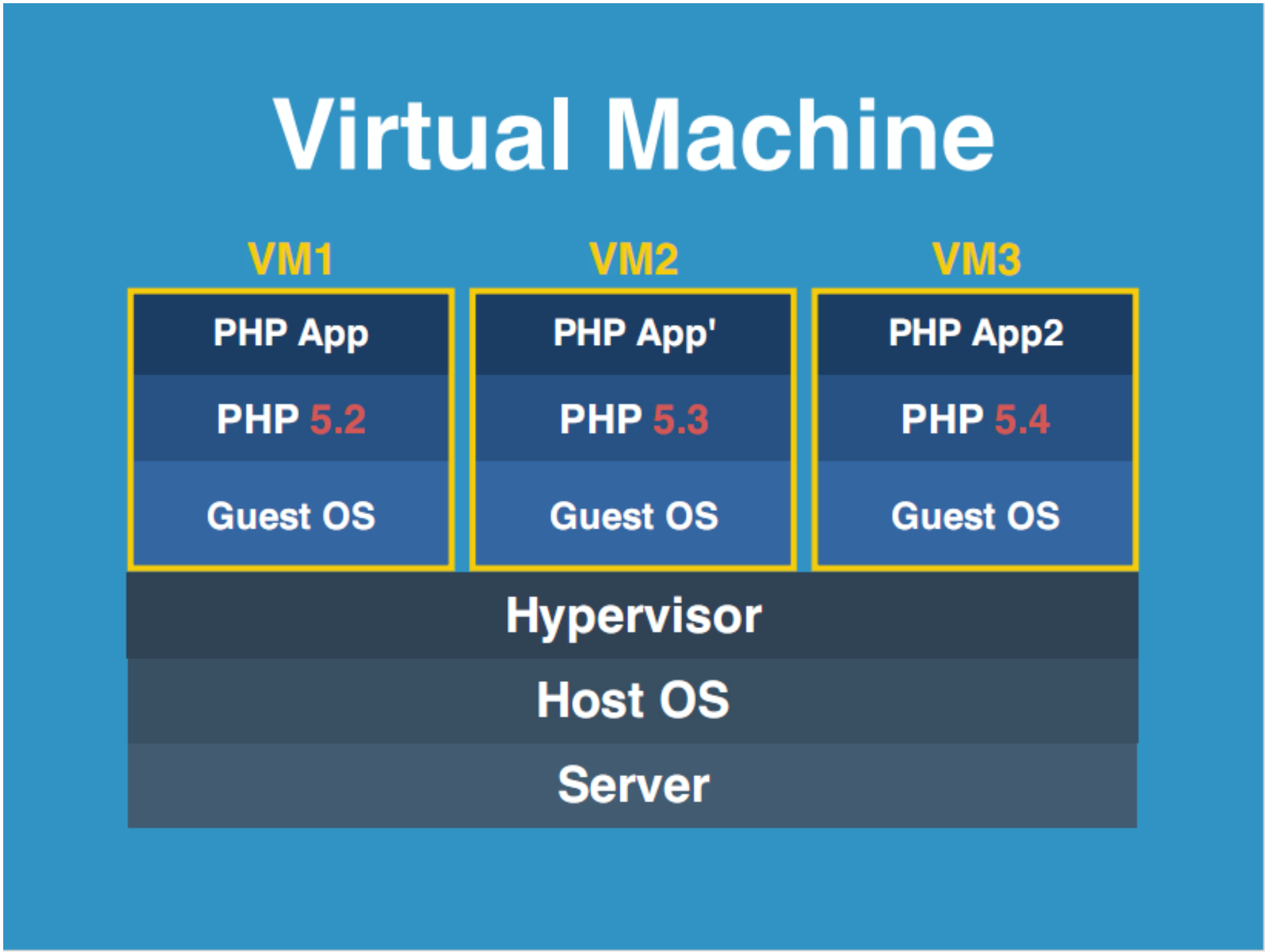
Virtual Machine



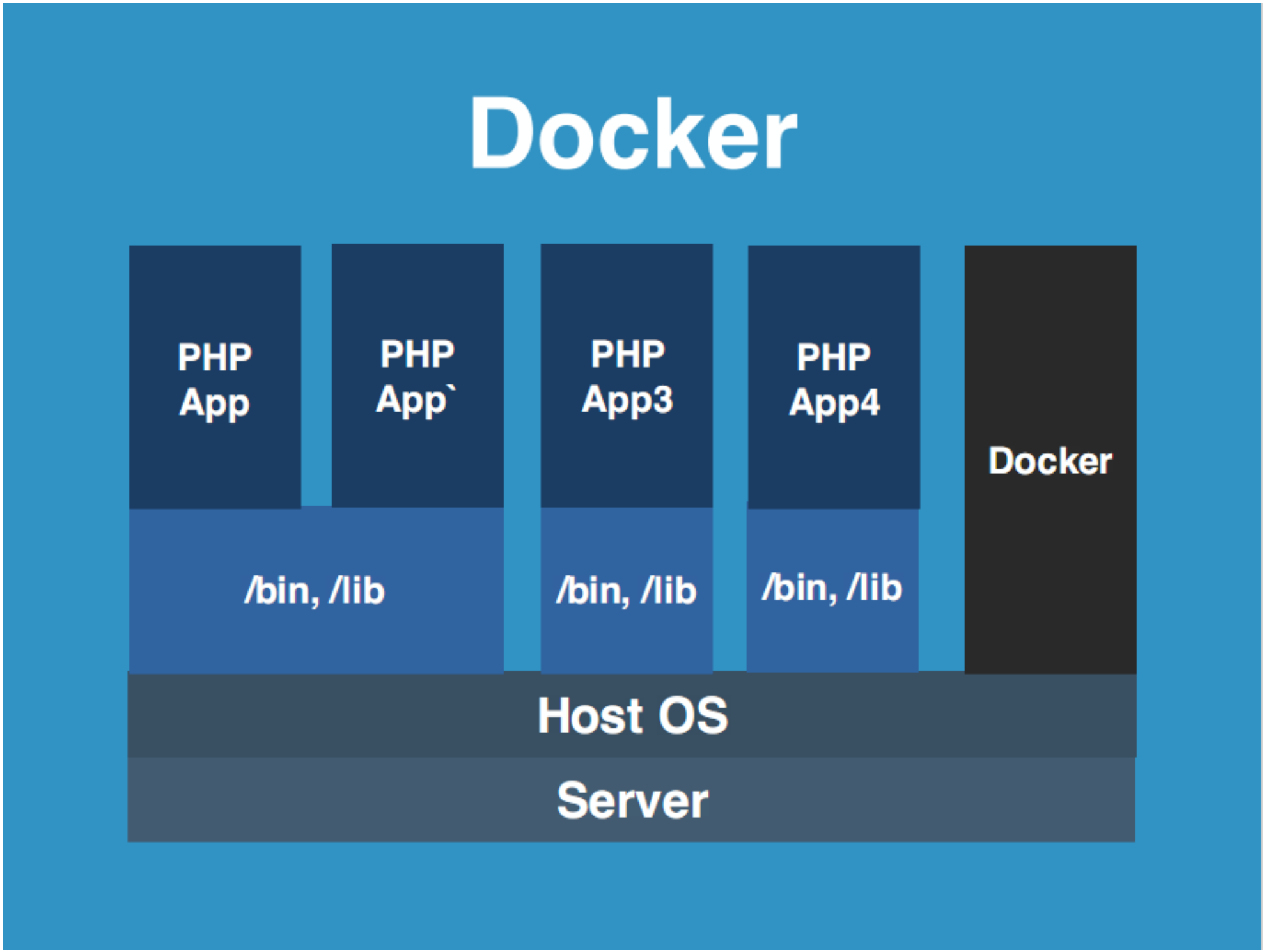
http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker



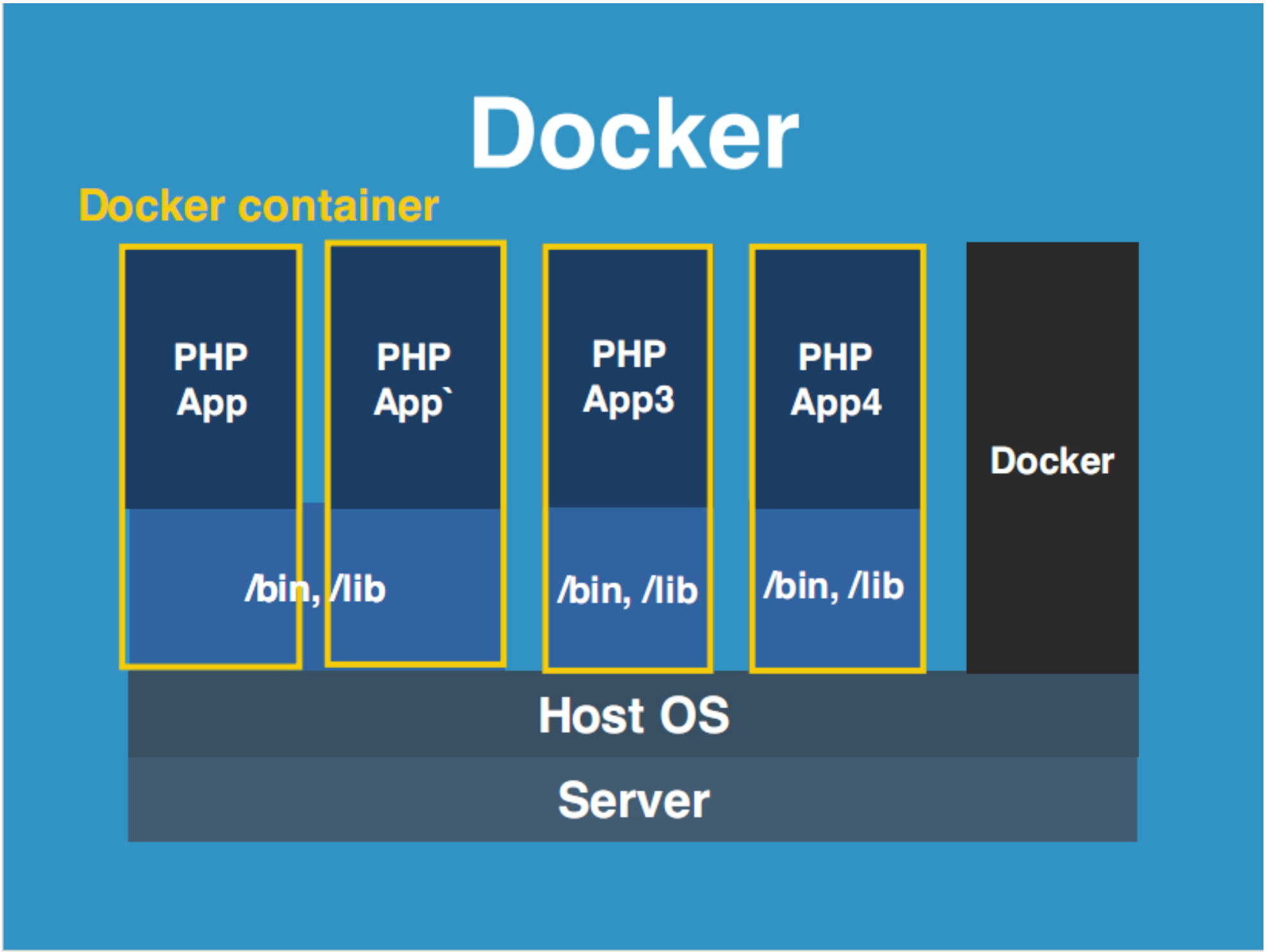
http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker



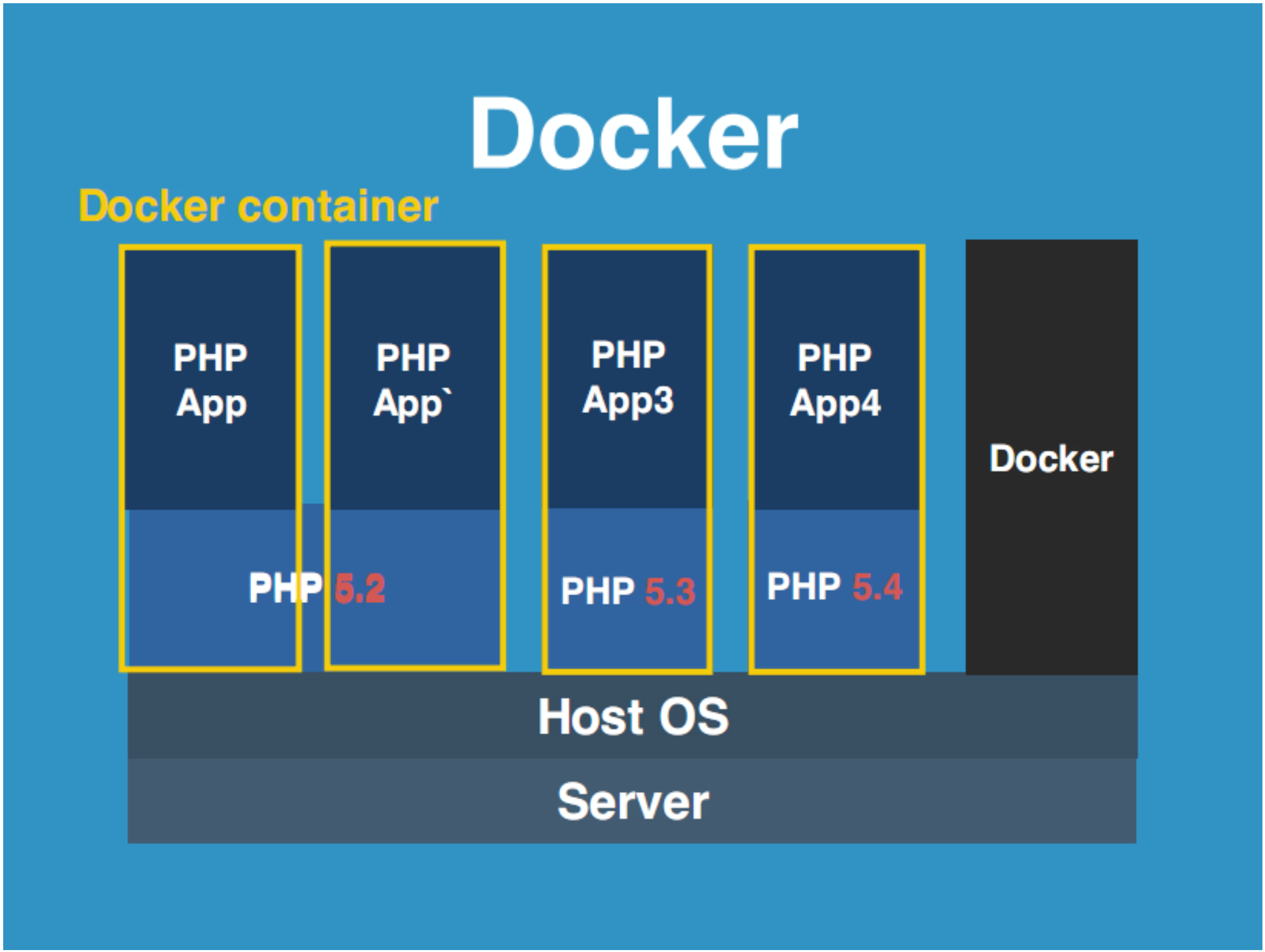
http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker



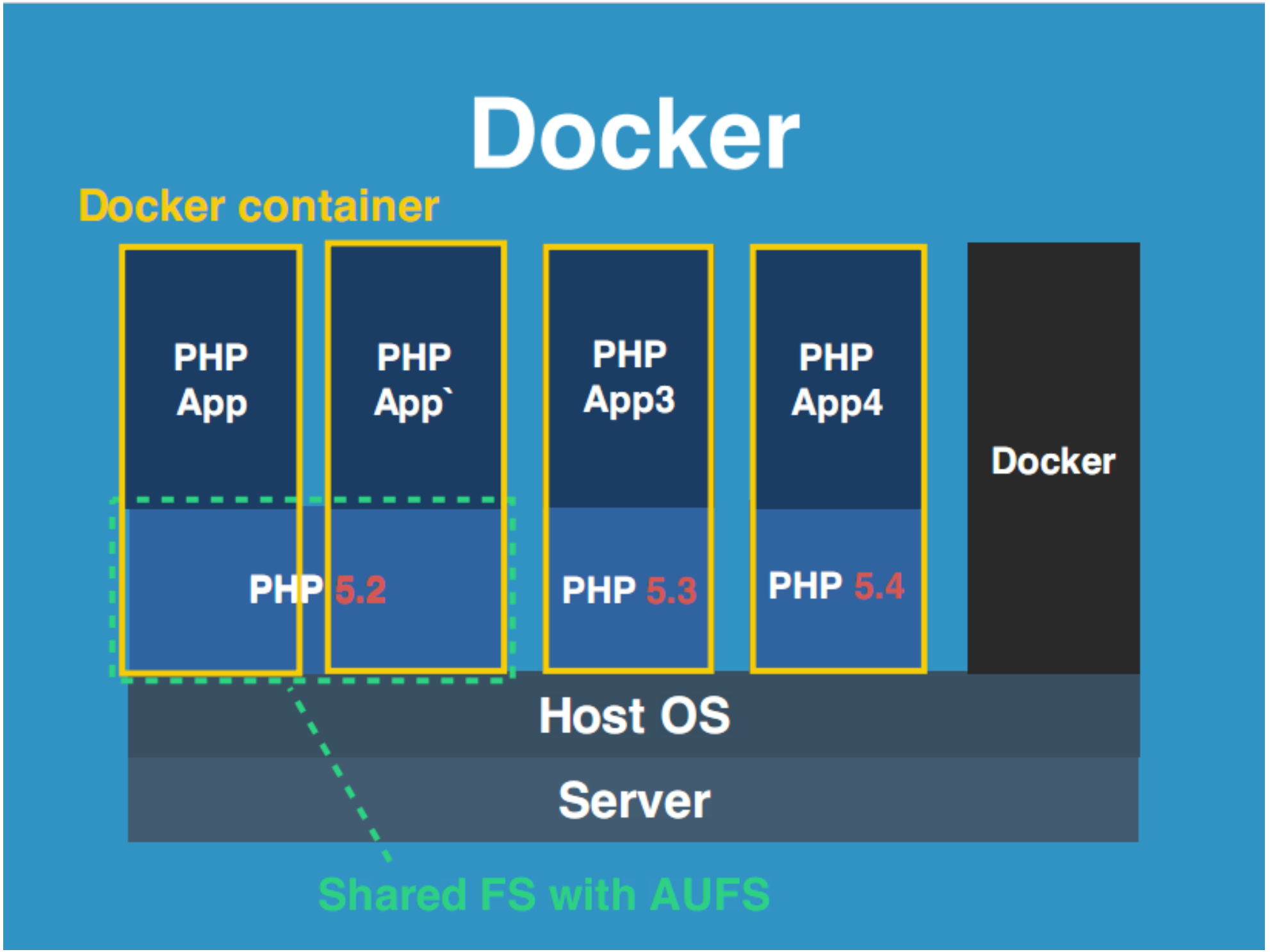
http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker



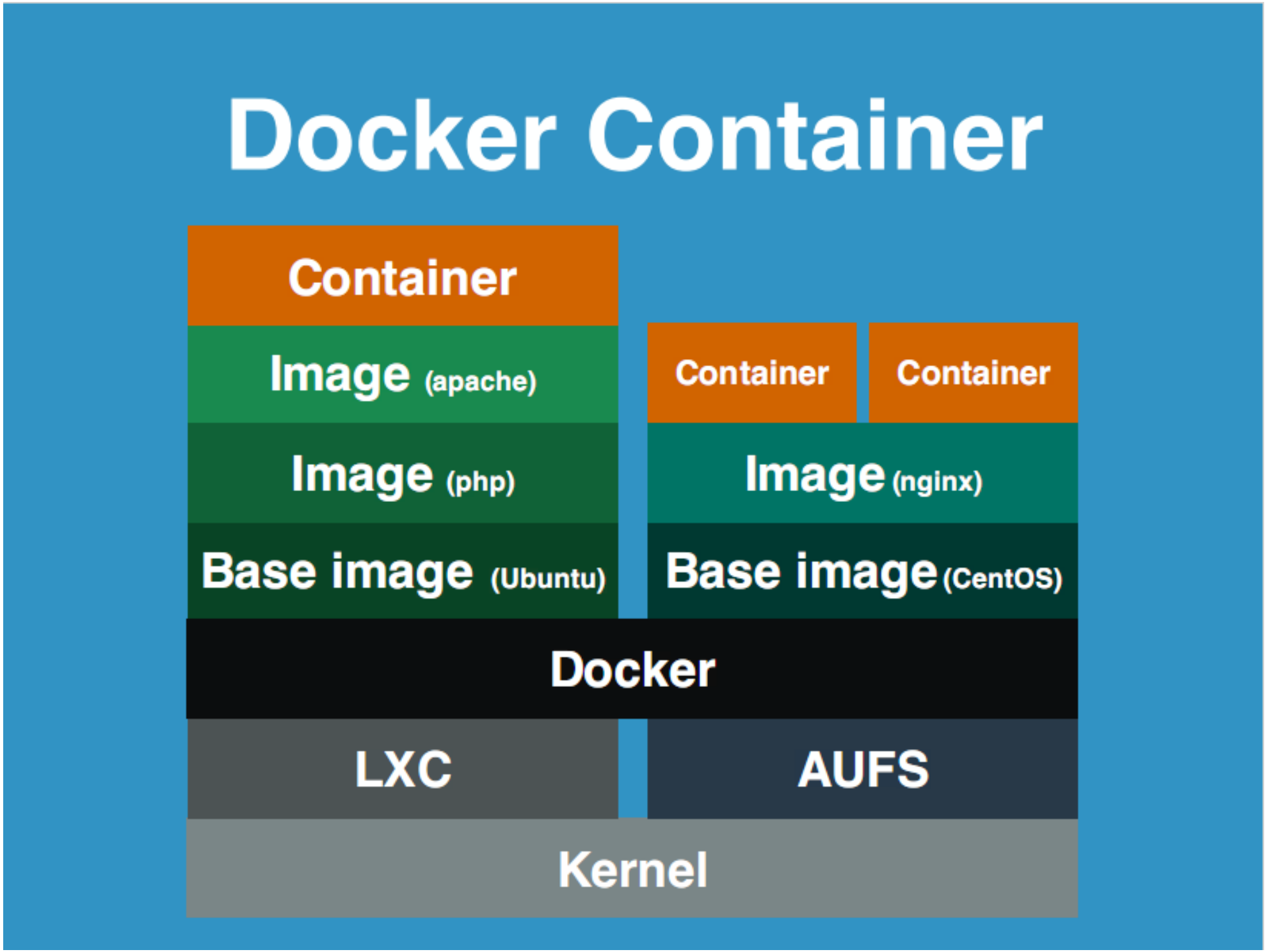
http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker



http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker



http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker



http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker

Docker 如何將 Container 標準化？

- Docker 用 aufs 檔案系統來設計一個可以層層堆疊的 Container 映象檔，將應用程式或相關函式庫與設定檔都打包進 Docker 映象檔
- 提供 Dockerfile 設定檔記錄每一個步驟包括參數
- Docker 公司也釋出 API，可以用來控制所有的 Container 相關指令

http://www.ithome.com.tw/news/91847

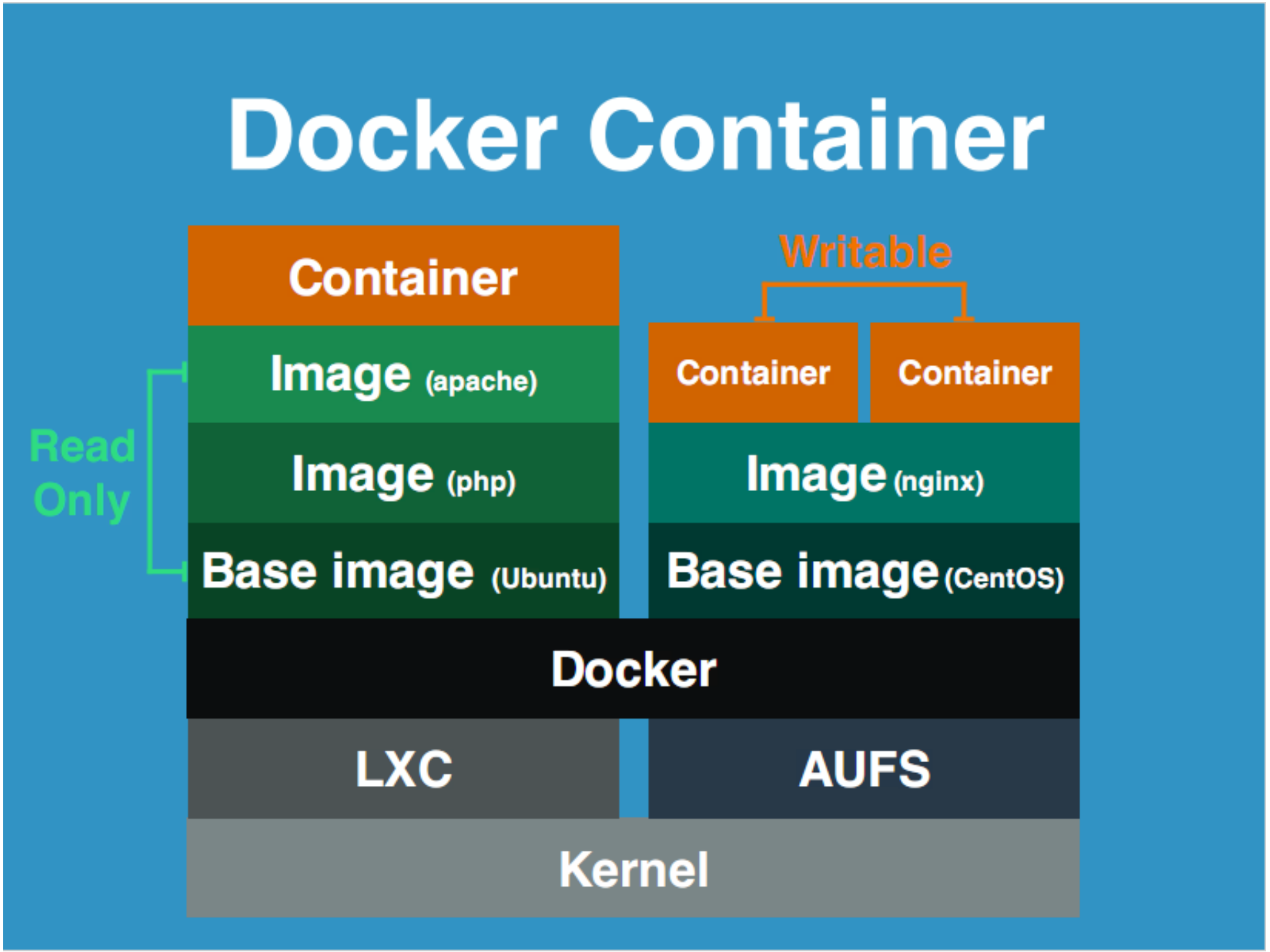
Images vs. Containers

Images

- 類似虛擬機器的映像檔 (image)
- 是一個唯讀 (Read Only) 的模板
- 通常包含檔案系統
- 用來建立 Container

Containers

- 可讀可寫
- 是 image 的實體 (running process)
- 通常由一個或多個 image 所建立
- 可看成簡易的 Linux 系統環境



http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker

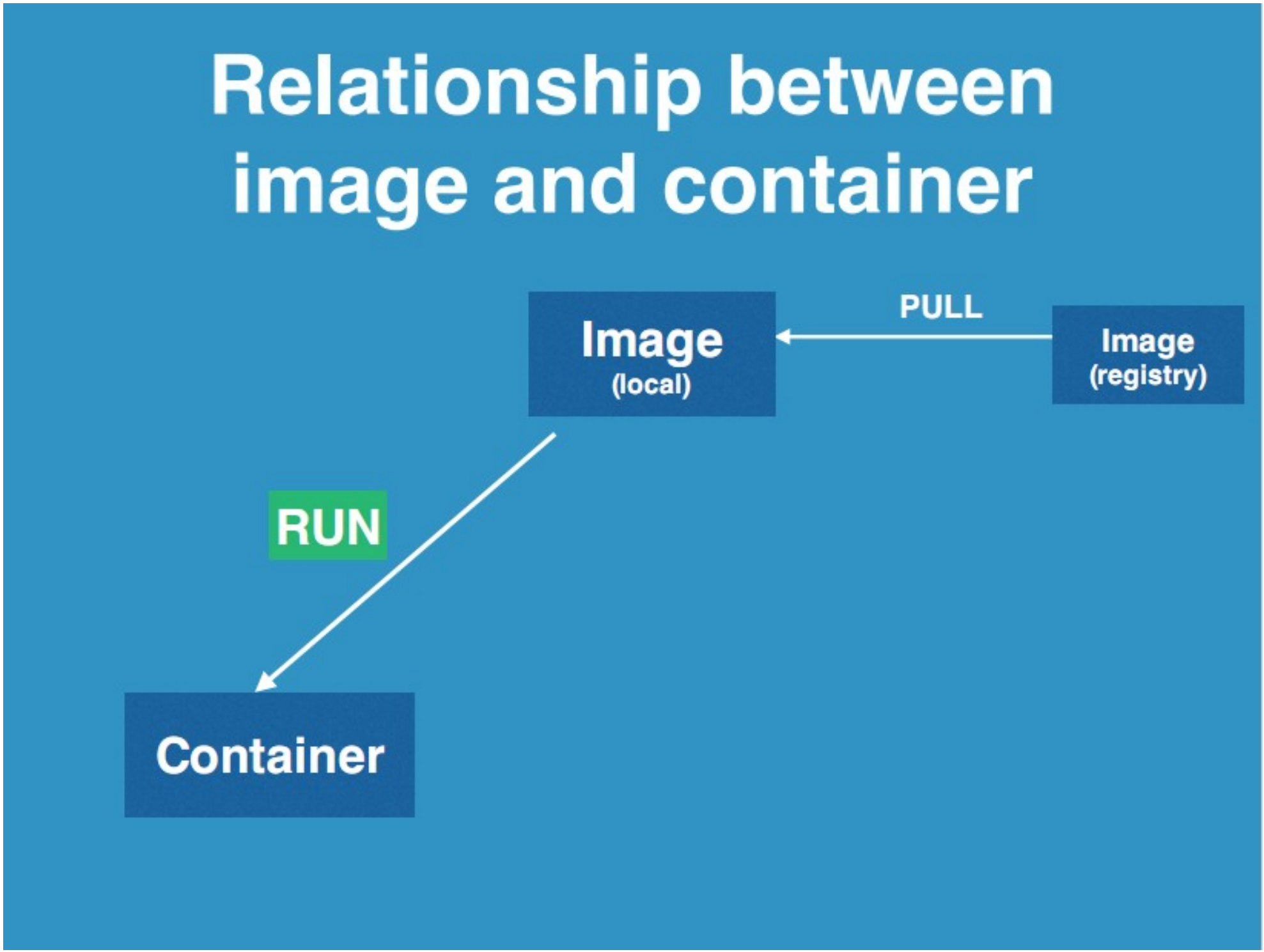
從操作看 Images 和 Containers 關係

Relationship between image and container

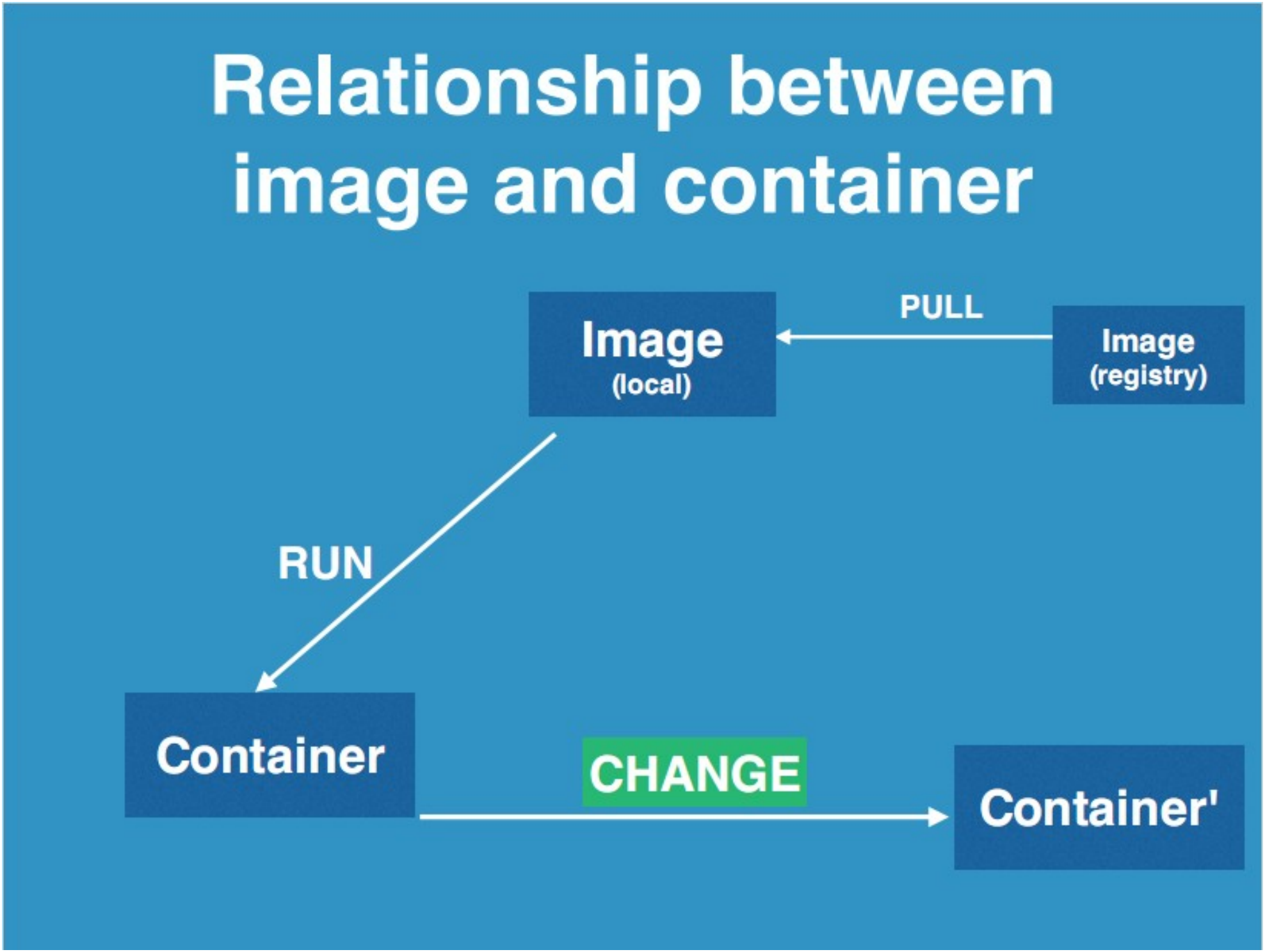
http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker

Relationship between image and container

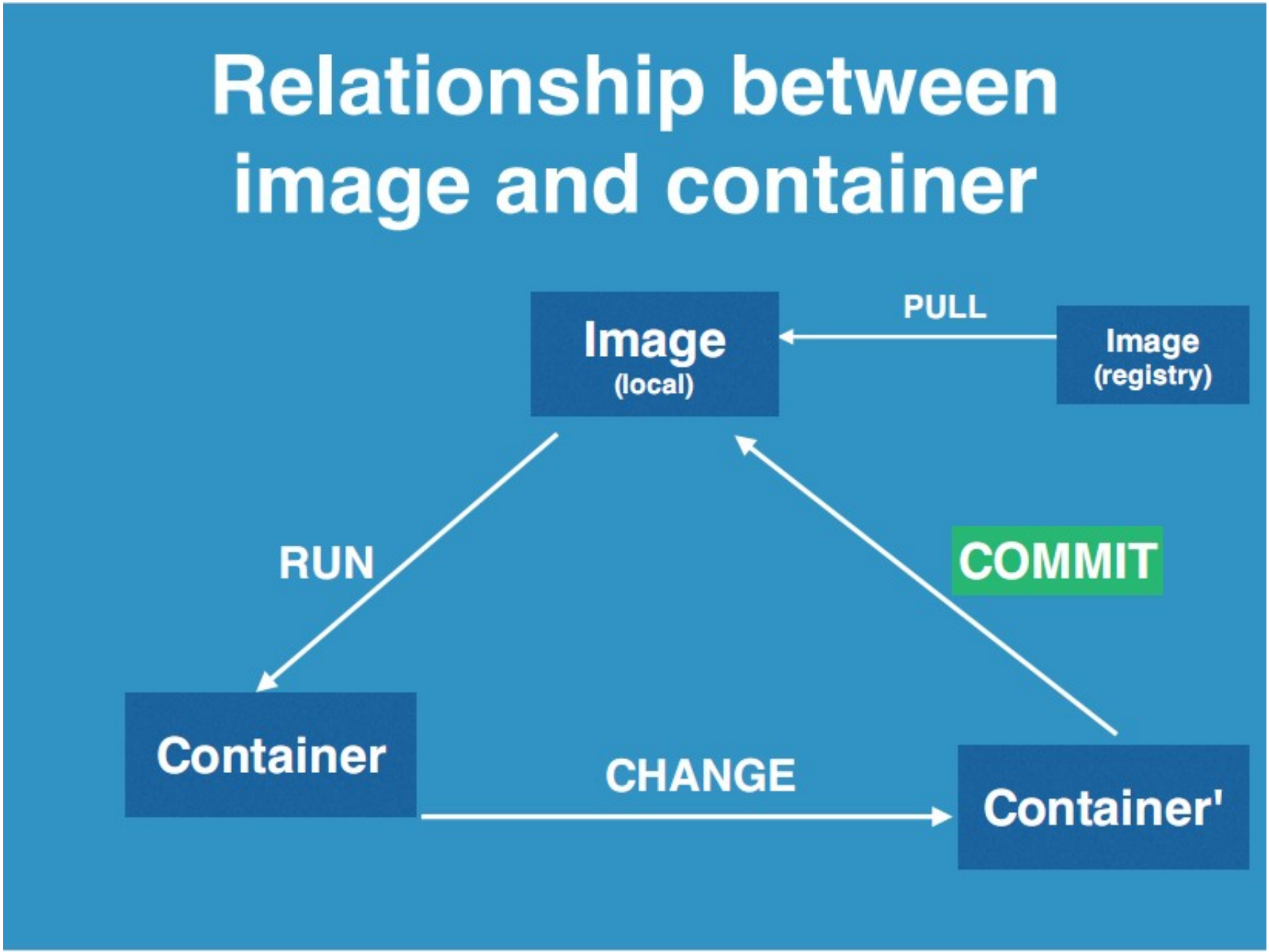
http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker



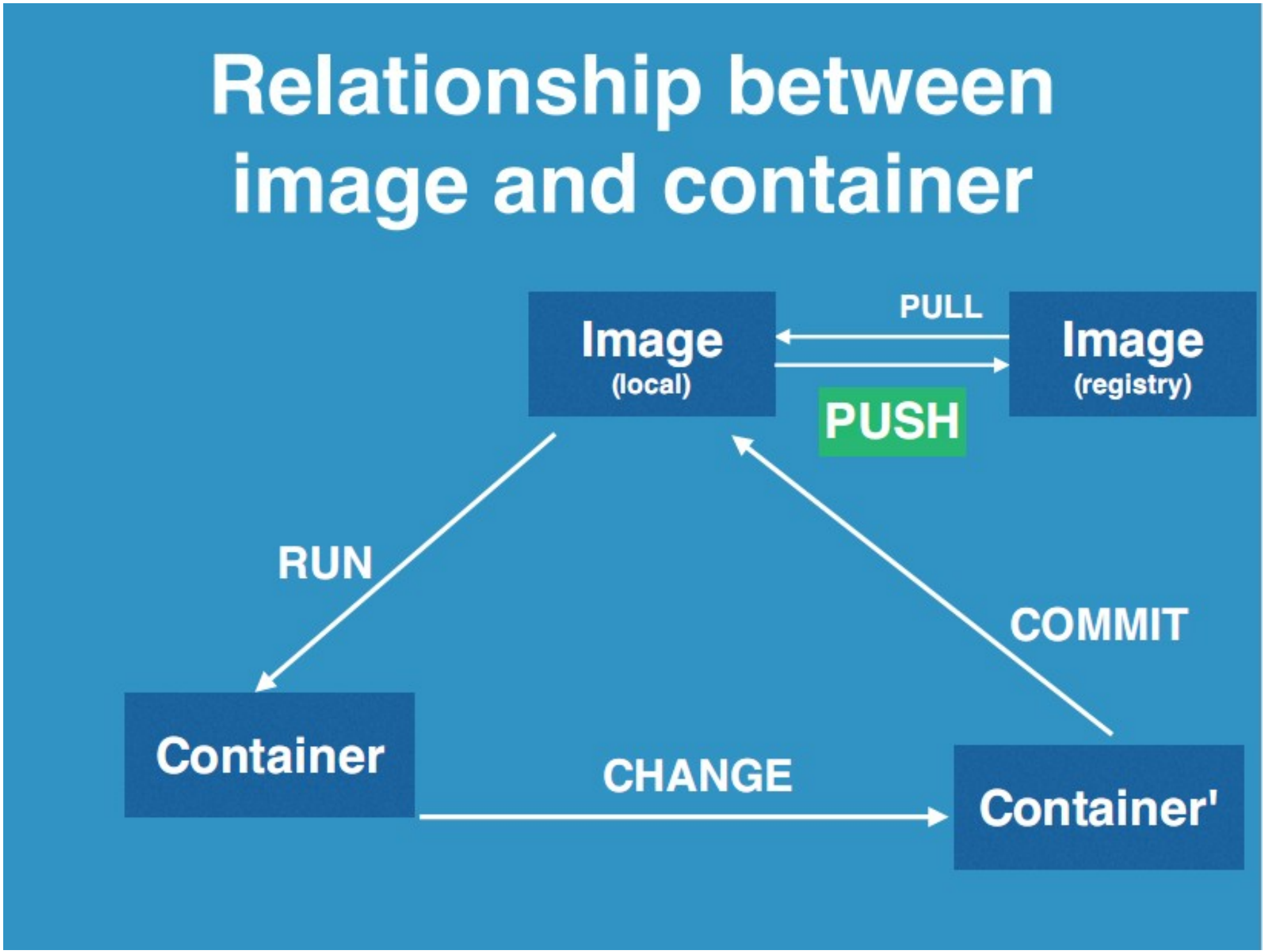
http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker



http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker



http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker



http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker

在 Raspberry Pi 上安裝與使用 Docker

安装

- 安装最新版 docker
- `$ curl -sSL https://get.docker.com | sh`
- `$ sudo usermod -aG docker pi`

Version

- `$ sudo docker version`

```
 Client:
  Version:      1.12.3
  API version:  1.24
  Go version:   go1.6.3
  Git commit:   6b644ec
  Built:        Wed Oct 26 19:06:36 2016
  OS/Arch:      linux/arm
 Server:
  Version:      1.12.3
  API version:  1.24
  Go version:   go1.6.3
```

Search

- `$ sudo docker search raspbian`

```
NAME                      DESCRIPTION                          STARS
resin/rpi-raspbian        Base image for the Raspberry Pi      196
sdhibit/rpi-raspbian      Base raspbian image for ARM ...      44
jsurf/rpi-raspbian        raspbian jessie base image  ...      3
…
```

Search

- `$ sudo docker search raspbian` keyword

```
NAME                        DESCRIPTION                                     STARS
resin/rpi-raspbian          Base image for the Raspberry Pi                 196
sdhibit/rpi-raspbian        Base raspbian image for ARM ...                 44
jsurf/rpi-raspbian          raspbian jessie base image  ...                 3
…
```

Pull Base Image

- `$ sudo docker pull resin/rpi-raspbian`

```
Using default tag: latest
latest: Pulling from resin/rpi-raspbian
614910dbc340: Pull complete
5d152067e785: Pull complete
fb055dd17164: Pull complete
c90a0f777307: Pull complete
2da42988b278: Pull complete
522f3a08010d: Pull complete
018f5d347be4: Pull complete
6a5aa9ebbc85: Pull complete
...
```

Pull Base Image

- `$ sudo docker pull resin/rpi-raspbian` image name

```
Using default tag: latest
latest: Pulling from resin/rpi-raspbian
614910dbc340: Pull complete
5d152067e785: Pull complete
fb055dd17164: Pull complete
c90a0f777307: Pull complete
2da42988b278: Pull complete
522f3a08010d: Pull complete
018f5d347be4: Pull complete
6a5aa9ebbc85: Pull complete
...
```

List Images

- `$ sudo docker images`

```
REPOSITORY              TAG     IMAGE ID      CREATED         SIZE
resin/rpi-raspbian      latest  a649b6a8a7cf  1 minute ago  117.2 MB
```

Create and Start a Container

- `$ sudo docker run -i -t resin/rpi-raspbian /bin/bash`

```
root@7aa71c1a817f:/#
```

- `$ sudo docker ps -a`

```
CONTAINER ID  IMAGE                COMMAND                  CREATED         STATUS         NAMES
7aa71c1a817f  resin/rpi-raspbian   "/usr/bin/entry.sh /b"   2 minutes ago   Up 2 minutes   ecstatic_booth
```

Create and Start a Container

- `$ sudo docker run -i -t resin/rpi-raspbian /bin/bash`

```
root@7aa71c1a817f:/#
```

interactive image name execute

- `$ sudo docker ps -a` all

```
CONTAINER ID  IMAGE                COMMAND                  CREATED        STATUS        NAMES
7aa71c1a817f  resin/rpi-raspbian   "/usr/bin/entry.sh /b"   2 minutes ago  Up 2 minutes  ecstatic_booth
```

Exit Container

- # exit

```
exit
```

- $ sudo docker ps -a

```
CONTAINER ID  IMAGE                COMMAND                  CREATED         STATUS
7aa71c1a817f  resin/rpi-raspbian   "/usr/bin/entry.sh /b"   8 minutes ago   Exited (0) 8 seconds ago
```

Daemon Mode and Attach a Container

- `$ sudo docker run -it -d resin/rpi-raspbian /bin/bash`

```
72d6f54a6aaa8433650258d208b0f533673b3a4379887625ded467daa8587b30
```

- `$ sudo docker ps -a`

```
CONTAINER ID  IMAGE                COMMAND                  CREATED         STATUS
72d6f54a6aaa  resin/rpi-raspbian   "/usr/bin/entry.sh /b"   7 seconds ago   Up 2 seconds
7aa71c1a817f  resin/rpi-raspbian   "/usr/bin/entry.sh /b"   14 minutes ago  Exited (0) 5 minutes ago
```

- `$ sudo docker exec -it 72d6f54a6aaa /bin/bash`

```
root@72d6f54a6aaa:/#
```

Daemon Mode and Attach a Container

daemon mode

- `$ sudo docker run -it -d resin/rpi-raspbian /bin/bash`

```
72d6f54a6aaa8433650258d208b0f533673b3a4379887625ded467daa8587b30
```

- `$ sudo docker ps -a`

```
CONTAINER ID  IMAGE                COMMAND                  CREATED         STATUS
72d6f54a6aaa  resin/rpi-raspbian   "/usr/bin/entry.sh /b"   7 seconds ago   Up 2 seconds
7aa71c1a817f  resin/rpi-raspbian   "/usr/bin/entry.sh /b"   14 minutes ago  Exited (0) 5 minutes ago
```

- `$ sudo docker exec -it 72d6f54a6aaa /bin/bash`

```
root@72d6f54a6aaa:/#
```

container id

Remove a Container

- `$ sudo docker stop 72d6f54a6aaa`

```
72d6f54a6aaa
```

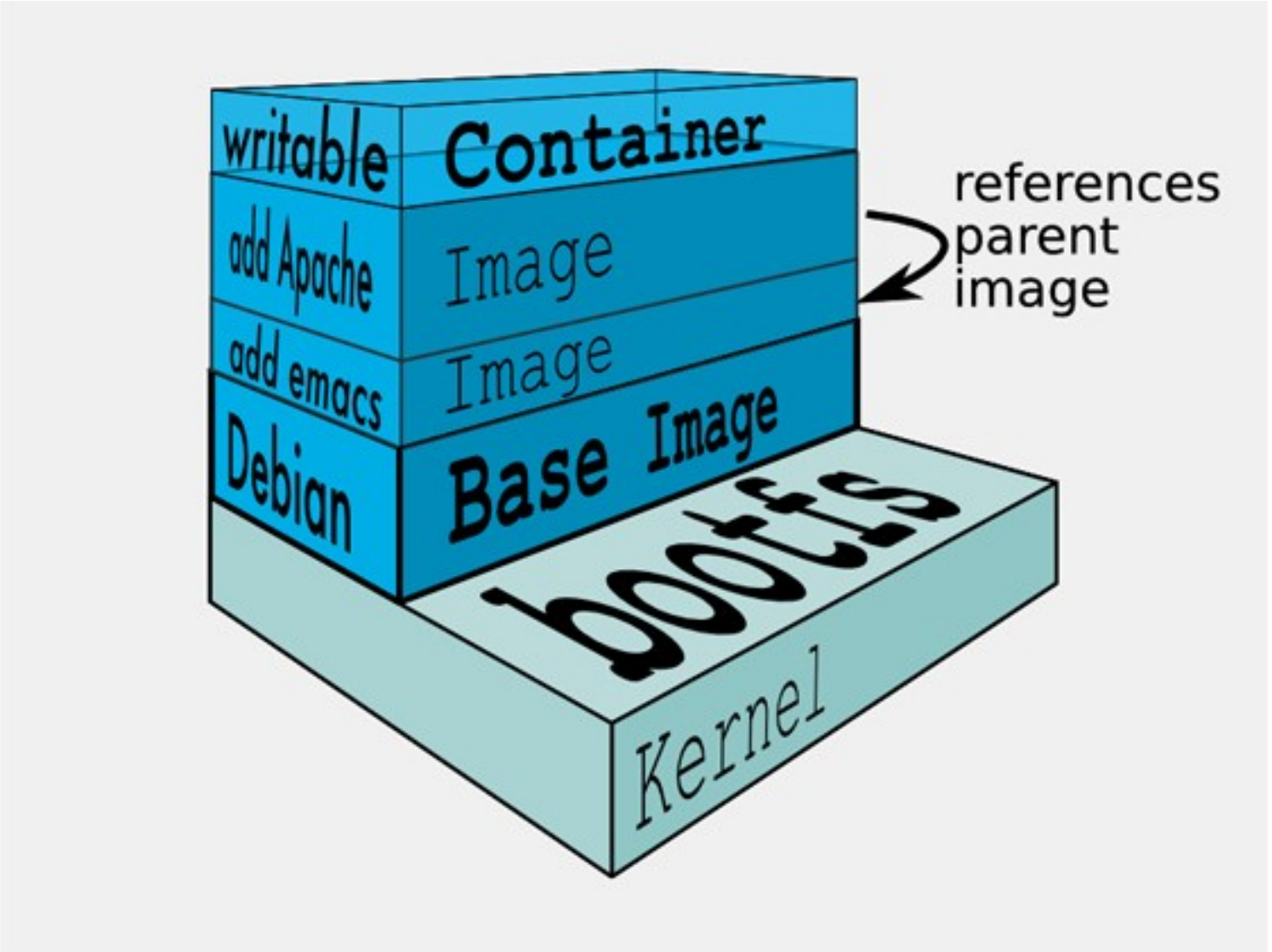
- `$ sudo docker rm 72d6f54a6aaa`

```
72d6f54a6aaa
```

- `$ sudo docker ps -a`

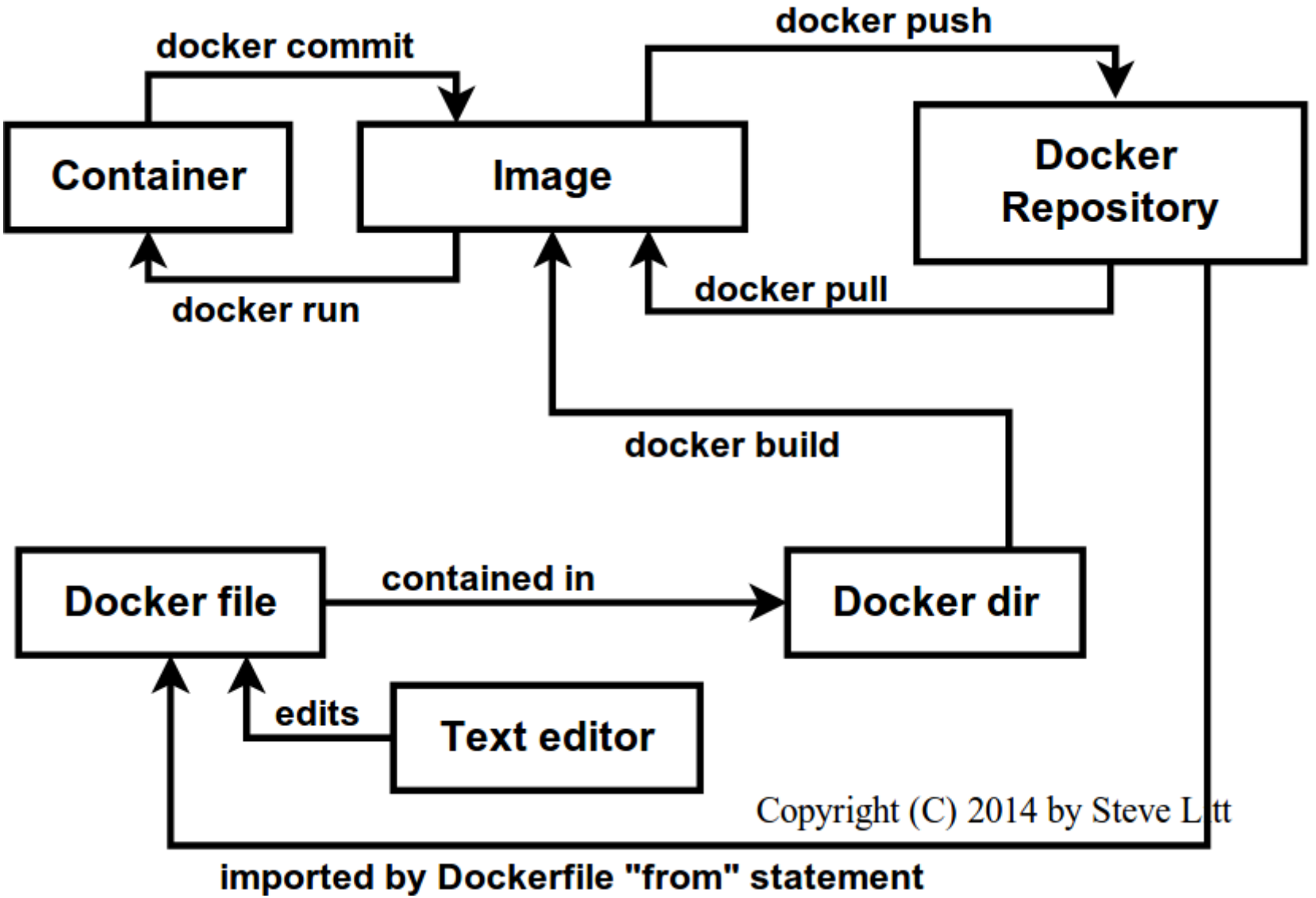
```
CONTAINER ID  IMAGE                COMMAND                  CREATED        STATUS
7aa71c1a817f  resin/rpi-raspbian   "/usr/bin/entry.sh /b"   8 minutes ago  Exited (0) 8 seconds ago
```

Image 和 Container 的關係



https://flipboard.com/@blowfishyeh/docker-d7lg2im2z

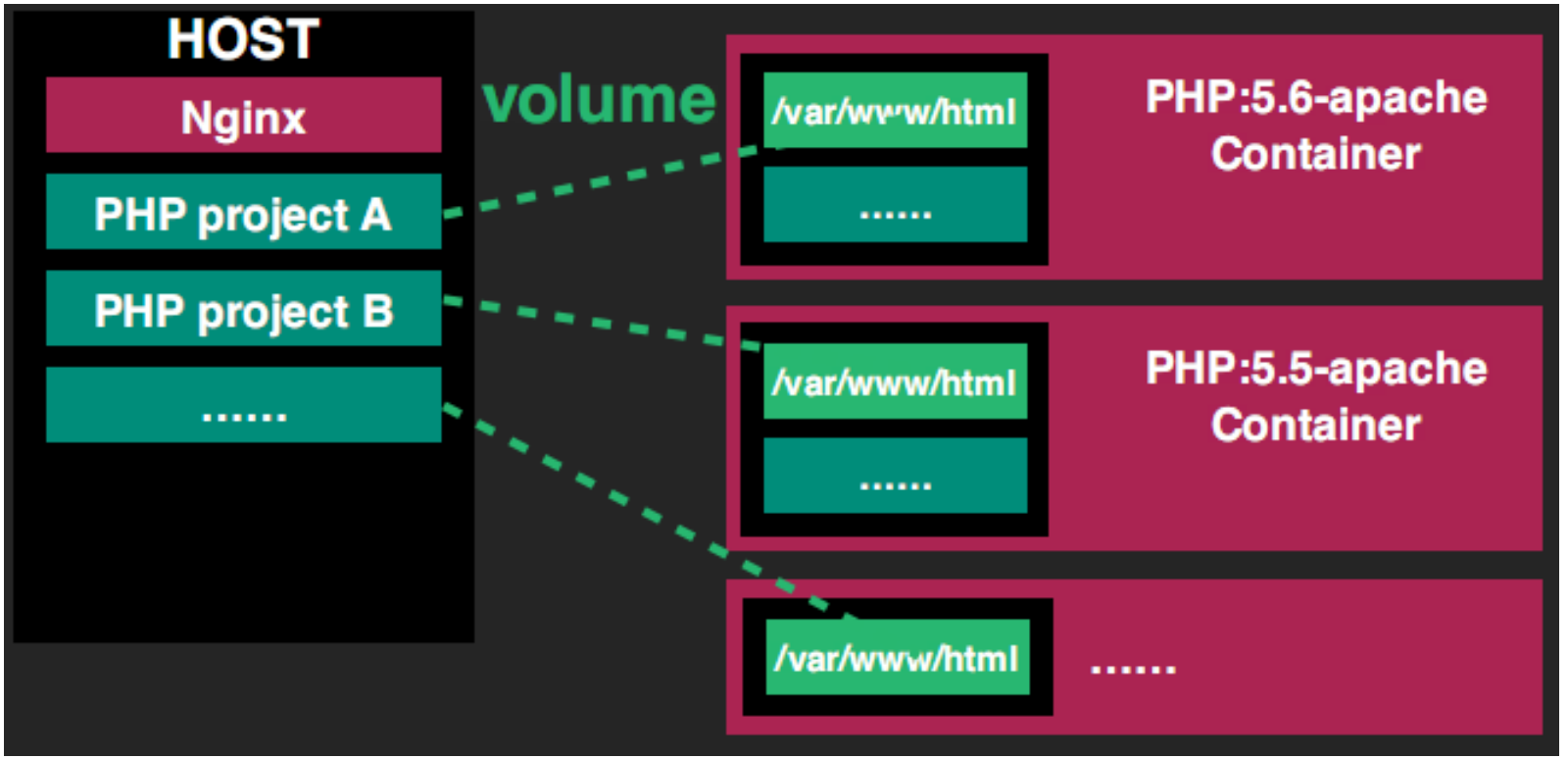
Image 和 Container 的狀態圖



http://www.troubleshooters.com/linux/docker/docker_newbie.htm

實戰

- 本次目標：
  - 在 Pi 安裝多個 PHP 版本，並對應到不同的專案



http://www.slideshare.net/ruoshiling/how-to-deploy-php-projects-with-docker

步驟

- 在 Pi 拉一個 base image 下來
- 建立新 Container(ap25)，並安裝 Apache2+PHP5
- 建立新 Container(ap27)，並安裝 Apache2+PHP7
- 在 Pi 建立兩個目錄，分別對應到 a2p5 和 a2p7 的 www
- 正確結果是在不同 URL 顯示不同的 PHP 版本

Pull Base Image

- `$ sudo docker pull resin/rpi-raspbian`

```
Using default tag: latest
latest: Pulling from resin/rpi-raspbian
Digest: sha256:4aadf300bd195e88bcd1c5096b9a80551a9b2f4bf9c59cec34
Status: Image is up to date for resin/rpi-raspbian:latest
```

- 如果前面已經拉下過會先檢查版本，如果已經是最新版會顯示`up to date`

建立新 Container 名稱為 apache2_php5

- ```
  $ sudo docker run -it -d  \
  --name=apache2_php5  \
  -v /home/pi/www-php5:/var/www/html  \
  -p 2005:80  \
  resin/rpi-raspbian /bin/bash
  ```
  ```

建立新 Container 名稱為 apache2_php5

- ```
  $ sudo docker run -it -d  \
  --name=apache2_php5  \
  -v /home/pi/www-php5:/var/www/html  \
  -p 2005:80  \
  resin/rpi-raspbian /bin/bash
  ```

- --name: 自訂名稱
- -v:shared folder,[host]:[container]
- -p:port mapping,[host]:[container]
  ```

連接已經啟動的 Container(apache2_php5)

- ```
  $ sudo docker exec -it  \
  apache2_php5 /bin/bash
  ```
  ```

連接已經啟動的 Container

- `$ sudo docker exec -it \`
  `apache2_php5 /bin/bash`

  exec: 執行命令 (例如 /bin/bash)

  apache2_php5: 自訂名稱 ,docker ps -a 可查

安裝 PHP5 和 Apache2

- `# sudo apt-get update`
- `# sudo apt-get install nano apache2 php5 libapache2-mod-php5`

- 啟動 Apache
- `# apachectl start`

建立新 Container 名稱為 apache2_php7

- ```
  $ sudo docker run -it -d  \
  --name=apache2_php7  \
  -v /home/pi/www-php7:/var/www/html  \
  -p 2007:80  \
  resin/rpi-raspbian /bin/bash
  ```
  ```

連接已經啟動的 Container(apache2_php5)

- `$ sudo docker exec -it  \`
  `apache2_php7 /bin/bash`

安装 PHP7 和 Apache2

- `# sudo apt-get update`
- `# sudo apt-get install nano`

安裝 PHP7 和 Apache2

- 新增APT來源
- # nano /etc/apt/sources.list
- deb http://repozytorium.mati75.eu/raspbian jessie-backports main contrib non-free
- # sudo gpg --keyserver pgpkeys.mit.edu --recv-key CCD91D6111A06851
- # sudo gpg --armor --export CCD91D6111A06851 | sudo apt-key add -
- # sudo apt-get update
- # apt-get install apache2 php7.0 libapache2-mod-php7.0
- 啟動Apache
- # apachectl start

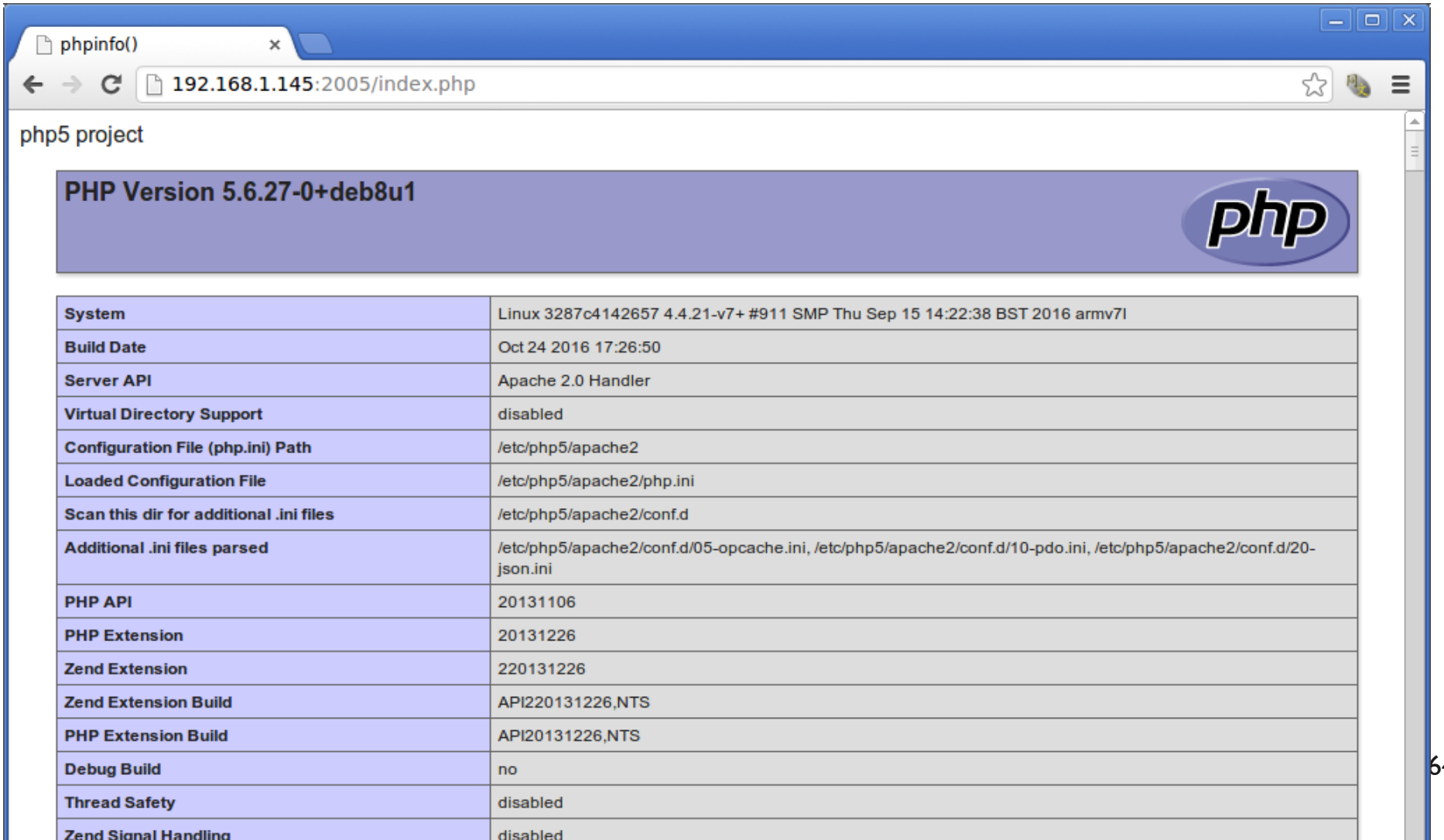
在 Pi 建立各別對應的專案

- 需求：
  - apache2_php5 對應目錄 /home/pi/www-php5
  - apache2_php7 對應目錄 /home/pi/www-php7
- 步驟：
  - `$ mkdir /home/pi/www-php5 /home/pi/www-php7`
  - `$ echo "<?php echo 'php5 project'; phpinfo(); ?>" > /home/pi/www-php5/index.php`
  - `$ echo "<?php echo 'php7 project'; phpinfo(); ?>" > /home/pi/www-php7/index.php`

https://www.stewright.me/2016/03/turn-raspberry-pi-3-php-7-powered-web-server/

確認 Apache2+PHP5

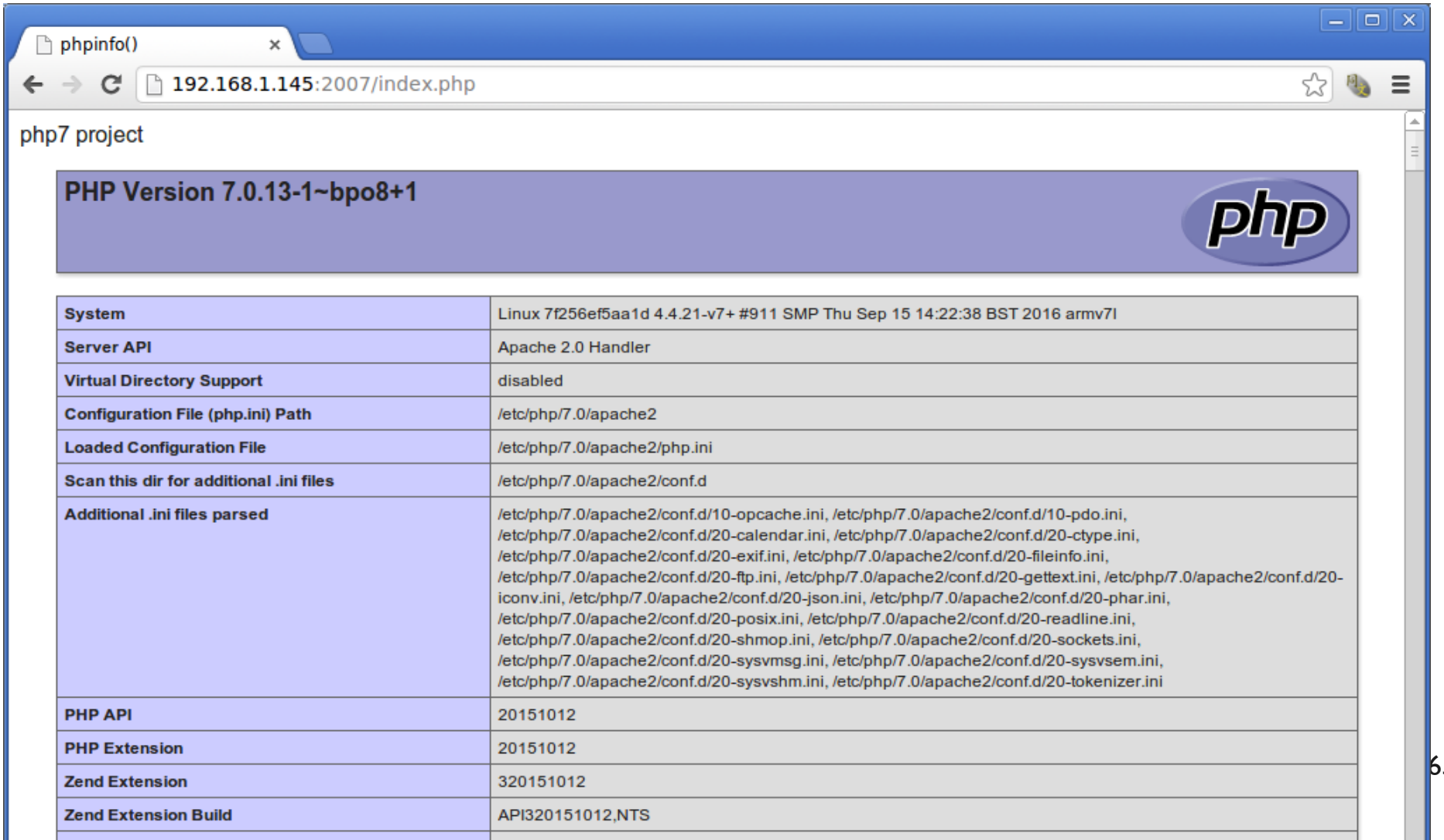
- 網址 :http://RPI_IP:2005/index.php



System	Linux 3287c4142657 4.4.21-v7+ #911 SMP Thu Sep 15 14:22:38 BST 2016 armv7l
Build Date	Oct 24 2016 17:26:50
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-json.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled

確認 Apache2+PHP7

- 網址 :http://RPI_IP:2007/index.php



System	Linux 7f256ef5aa1d 4.4.21-v7+ #911 SMP Thu Sep 15 14:22:38 BST 2016 armv7l
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvsem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,NTS

BONUS

將 Container 匯出給別人使用

Export Container to tar Image

- $ sudo docker ps -a

```
CONTAINER ID  IMAGE                COMMAND                  PORTS                   NAMES
7f256ef5aa1d  resin/rpi-raspbian   "/usr/bin/entry.sh /b"   0.0.0.0:2007->80/tcp    apache2_php7
3287c4142657  resin/rpi-raspbian   "/usr/bin/entry.sh /b"   0.0.0.0:2005->80/tcp    apache2_php5
```

- 將 apache2_php7 匯出成 image(預設 tar 檔)
- $ sudo docker export apache2_php7 > apache2_php7.tar

在另一台主機使用 (從 A 到 B)

- 用 scp 傳到另一台主機
- A$ scp apache2_php7.tar pi@x.x.x.x:/home/pi

- 在 B 還原 tar
- B$ cat apache2_php7.tar | docker import - apache2_php7

建立 Container 名稱為 B_apache2_php7

- B$ sudo docker images

```
REPOSITORY    TAG     IMAGE ID      CREATED              SIZE
apache2_php7  latest  376569993863  About a minute ago  208.9 MB
```

- B$ sudo docker run -it -d \
  --name=B_apache2_php7 \
  -v /home/pi/www-php7:/var/www/html \
  -p 2007:80 \
  apache2_php7 /bin/bash

在 B 連接 Container(B_apache2_php7)

- B$ sudo docker exec -it \
  B_apache2_php7 /bin/bash

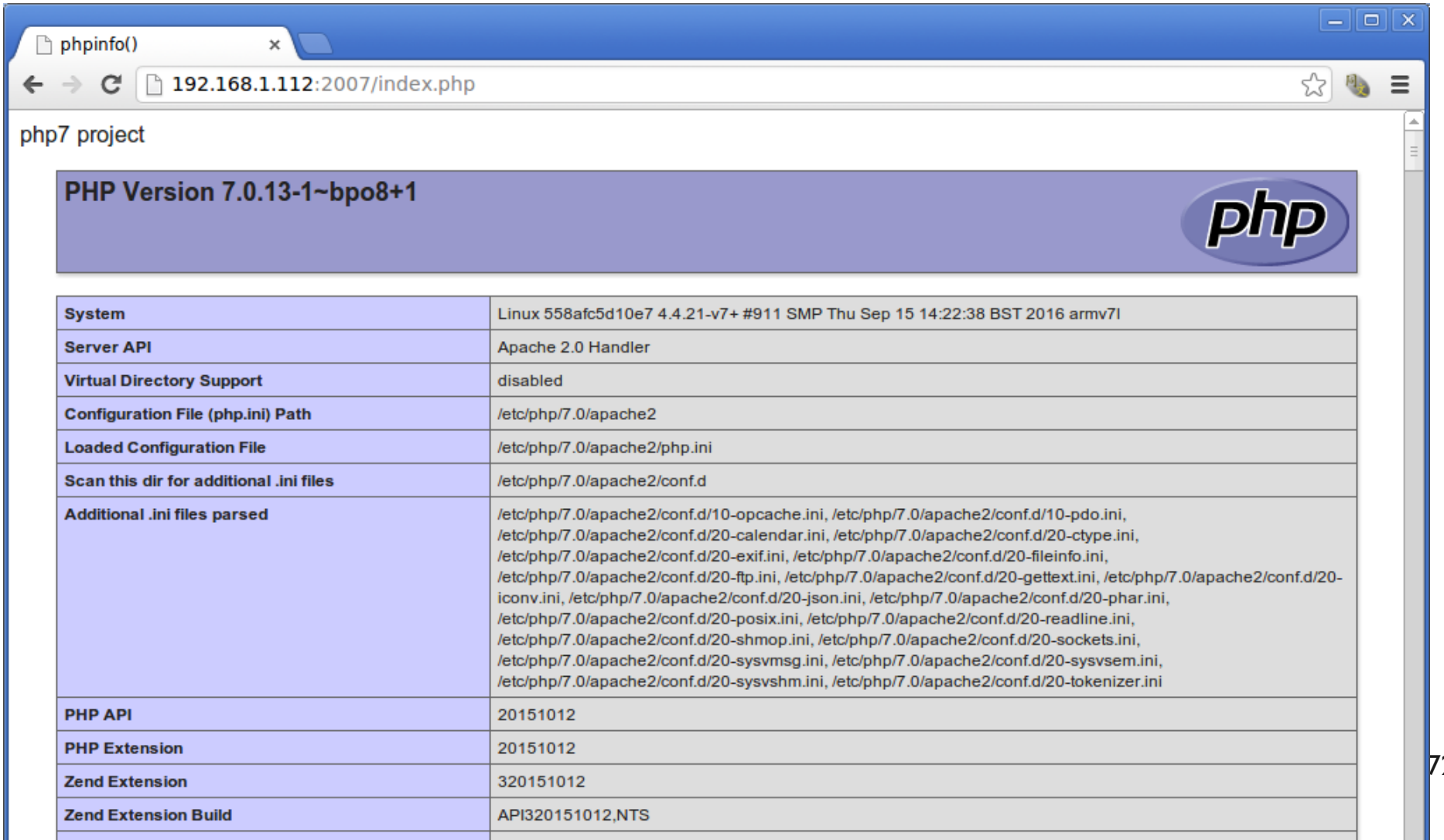
- 啟動 apache
  # apachectl start

在 B 建立對應的專案

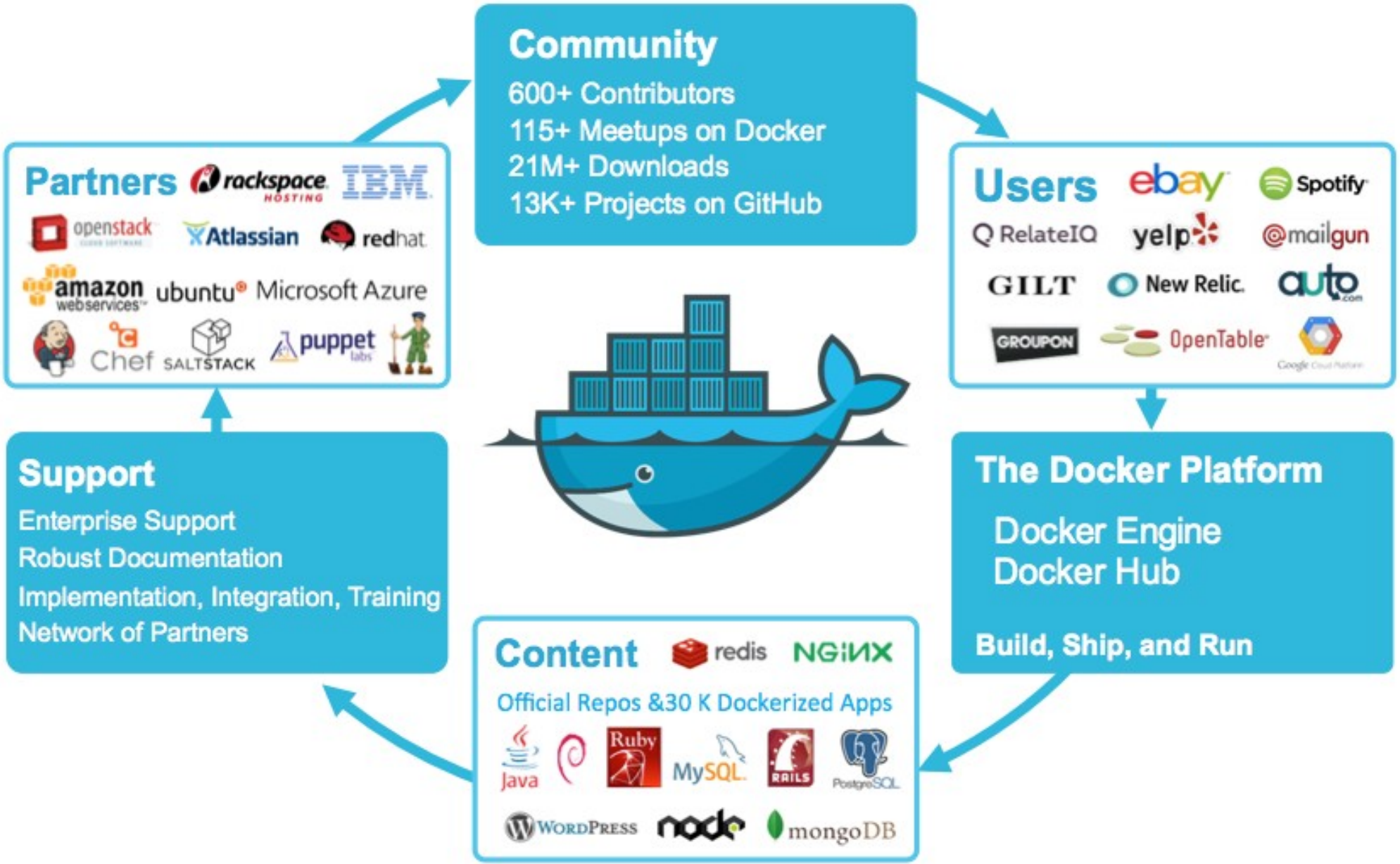
- B$ mkdir /home/pi/www-php7
- B$ echo "<?php echo 'php7 project'; phpinfo(); ?>" > /home/pi/www-php7/index.php

連上 B 看是否啟動成功

- 網址 :http://RPI_B_IP:2007/index.php

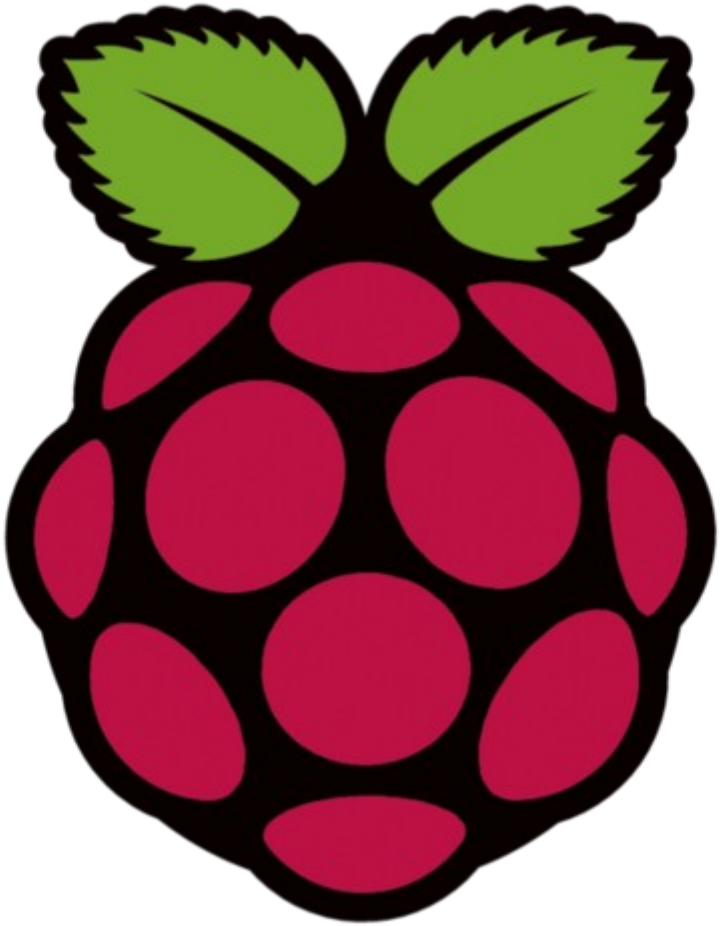


System	Linux 558afc5d10e7 4.4.21-v7+ #911 SMP Thu Sep 15 14:22:38 BST 2016 armv7l
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvsem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,NTS



https://blog.docker.com/2014/09/docker-closes-40m-series-c-led-by-sequoia/

Raspberry Pi Rocks the World



Thanks