

# Build Your Own Cloud

by docker (linux)

# 0. 情境

- 使用者從自己的電腦 登入WebServer (Docker 1)
  - 上傳一個檔案
- 系統把工作送入ComputingNode (Docker 2) 的排程，
  - 計算中、等待中: view.php 顯示 “工作未完成”
  - 計算完畢: view.php 顯示 計算結果

# 1. 架構

PC1 (linux as host OS)

Docker 1: Ubuntu WebServer

upload.php  
exe.php  
view.php

Docker 2: Ubuntu ComputingNode

compute.php  
cron (排程)

Share Folder



Jobid\_1

File  
Computing  
Output.txt

(已完成計算)

Jobid\_2

File  
Computing  
Output.txt

(已完成計算)

Jobid\_3

File  
Computing

(等待/計算中)

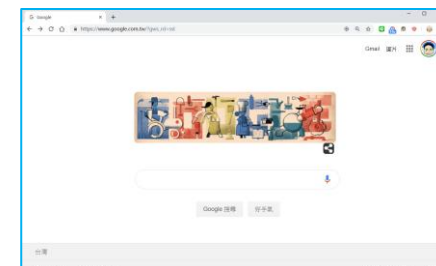
Jobid\_4

File

(等待/排隊中)

⋮

PC2 (任意作業系統)



瀏覽器

## 2. 部屬 Web server

- 2-1 docker pull
- 2-2 安裝環境
- 2-3 port forwarding 設定
- 2-4 測試

# 記下你的 virtual machine IP

```
root@ccllab-virtual-machine: ~ [146x47]
連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)

root@9f3c21465ab4:/var/www/html# read escape sequence
root@ccllab-virtual-machine:~# ifconfig
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    inet6 fe80::42:7bff:fe6c:18c0 prefixlen 64 scopeid 0x20<link>
    ether 02:42:7b:6c:18:c0 txqueuelen 0 (Ethernet)
    RX packets 13620 bytes 626632 (626.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 15393 bytes 55877108 (55.8 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.174.128 netmask 255.255.255.0 broadcast 192.168.
    inet6 fe80::6bc4:c064:75c5:499 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:af:c2:20 txqueuelen 1000 (Ethernet)
    RX packets 369763 bytes 307323147 (307.3 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 69617 bytes 6560327 (6.5 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

# 2-1 docker pull

```
docker run -i -t -d --name webServer --privileged=true -v /share -p 8080:80 ubuntu:18.04
```

docker attach webServer

執行二個指令

apt-get update

apt-get install vim apache2 php libapache2-mod-php wget unzip

Please select the geographic area in which you live. Subsequent configuration questions will narrow this down by presenting a list of cities, representing the time zones in which they are located.

1. Africa    3. Antarctica    5. Arctic    7. Atlantic    9. Indian    11. SystemV    13. Etc  
2. America    4. Australia    6. Asia    8. Europe    10. Pacific    12. US

Geographic area: 6

選6.Asia

Please select the city or region corresponding to your time zone.

1. Aden	16. Brunei	31. Hong_Kong	46. Kuala_Lumpur	61. Pyongyang	76. Tehran
2. Almaty	17. Chita	32. Hovd	47. Kuching	62. Qatar	77. Tel_Aviv
3. Amman	18. Choibalsan	33. Irkutsk	48. Kuwait	63. Qostanay	78. Thimphu
4. Anadyr	19. Chongqing	34. Istanbul	49. Macau	64. Qyzylorda	79. Tokyo
5. Aqtau	20. Colombo	35. Jakarta	50. Magadan	65. Rangoon	80. Toms
6. Aqtobe	21. Damascus	36. Jayapura	51. Makassar	66. Riyadh	81. Ujung_Pandang
7. Ashgabat	22. Dhaka	37. Jerusalem	52. Manila	67. Sakhalin	82. Ulaanbaatar
8. Atyrau	23. Dili	38. Kabul	53. Muscat	68. Samarkand	83. Urumqi
9. Baghdad	24. Dubai	39. Kamchatka	54. Nicosia	69. Seoul	84. Ust-Nera
10. Bahrain	25. Dushanbe	40. Karachi	55. Novokuznetsk	70. Shanghai	85. Vientiane
11. Baku	26. Famagusta	41. Kashgar	56. Novosibirsk	71. Singapore	86. Vladivostok
12. Bangkok	27. Gaza	42. Kathmandu	57. Omsk	72. Srednekolymsk	87. Yakutsk
13. Barnaul	28. Harbin	43. Khandyga	58. Oral	73. Taipei	88. Yangon
14. Beirut	29. Hebron	44. Kolkata	59. Phnom_Penh	74. Tashkent	89. Yekaterinburg
15. Bishkek	30. Ho_Chi_Minh	45. Krasnoyarsk	60. Pontianak	75. Tbilisi	90. Yerevan

Time zone: 73

選73.Taipei



# 測試Apache跟PHP

執行: `/etc/init.d/apache2 start`

```
root@5bb2adc45a28:/# /etc/init.d/apache2 start
* Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. S
et the 'ServerName' directive globally to suppress this message
*
root@5bb2adc45a28:/#
```

`vim /var/www/html/test.php`

按下i切換到輸入模式

```
<?php
echo phpinfo();
?>
```

按下ESC離開編輯模式

輸入:wq!存檔

`cd /var/www/html;`

`chmod 777 test.php;`

打開windows瀏覽器輸入

`http://你的 virtualMachine IP:8080/test.php`

按ctrl+P+Q離開容器

PHP Version 7.2.17-0ubuntu0.18.04.1

php

System	Linux 5bb2adc45a28 4.9.125-linuxkit #1 SMP Fri Sep 7 08:29:28 UTC 2018 x86_64
Build Date	Apr 15 2019 14:12:38
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php7.2/apache2
Loaded Configuration File	/etc/php7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php7.2/apache2/conf.d
Additional .ini files parsed	/etc/php7.2/apache2/conf.d/10-opcache.ini, /etc/php7.2/apache2/conf.d/10-redis.ini, /etc/php7.2/apache2/conf.d/20-calendar.ini, /etc/php7.2/apache2/conf.d/20-ctype.ini, /etc/php7.2/apache2/conf.d/20-enchant.ini, /etc/php7.2/apache2/conf.d/20-ffi.ini, /etc/php7.2/apache2/conf.d/20-fileinfo.ini, /etc/php7.2/apache2/conf.d/20-ftp.ini, /etc/php7.2/apache2/conf.d/20-gettext.ini, /etc/php7.2/apache2/conf.d/20-iconv.ini, /etc/php7.2/apache2/conf.d/20-ldap.ini, /etc/php7.2/apache2/conf.d/20-openssl.ini, /etc/php7.2/apache2/conf.d/20-phar.ini, /etc/php7.2/apache2/conf.d/20-posix.ini, /etc/php7.2/apache2/conf.d/20-readline.ini, /etc/php7.2/apache2/conf.d/20-shmop.ini, /etc/php7.2/apache2/conf.d/20-sockets.ini, /etc/php7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php7.2/apache2/conf.d/20-sysvsem.ini, /etc/php7.2/apache2/conf.d/20-sysvshm.ini, /etc/php7.2/apache2/conf.d/20-tokenizer.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API20170718.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	ftp, ftps, compress, zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:  
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies  
with Zend OPcache v7.2.17-0ubuntu0.18.04.1, Copyright (c) 1999-2018, by Zend Technologies

zendengine

Configuration

apache2handler

Apache Version	Apache/2.4.29 (Ubuntu)
Apache API Version	2010211
Server Administrator	webmaster@localhost
Hostname/Port	***.***.***.***:80
User/Group	
Max Request	
Timeouts	
Virtual Serve	
Server Root	

Loaded Modules

core mod\_so mod\_watchdog http\_core mod\_log\_config mod\_logio mod\_version mod\_unwind mod\_access\_compat mod\_alias mod\_auth\_basic mod\_auth\_core mod\_authn\_file mod\_authz\_core mod\_authz\_host mod\_authz\_user mod\_authz\_md mod\_deflate mod\_dir mod\_env mod\_ext\_mime mod\_mime mod\_negotiation mod\_php mod\_php7 mod\_rewrite mod\_setenvif mod\_status

Filesystem

Process Manager

Monitor Status

看到這個畫面表示成功了

把範例程式 放到 container 的 /var/www/html 資料夾

- 登入 webServer container :

```
cd /var/www/html
```

```
wget http://120.126.17.185/labWebsite/courses/ITM088\_2021/week04\_docker\_cloud.zip
```

```
unzip week04_docker_cloud.zip
```

```
mv cloudsystem_docker cloudsystem
```

```
chmod 777 -Rf cloudsystem
```

```
cd cloudsystem
```

```
mv webserver/* /var/www/html/cloudsystem/
```

```
chmod -Rf 777 /share
```

更改 exe.php 程式碼，改成你的 virtual machine IP

不是 container 的 IP

```
1 <?php
2
3 ini_set('display_errors','1');
4 error_reporting(E_ALL);
5
6
7 $dirname = time().".".rand(0,100);
8 $dirPath = "/share/".$dirname."/";
9
10 $cmd = "mkdir $dirPath;chmod -Rf 777 $dirPath";
11 shell_exec($cmd);
12 if($_FILES["uploadFile"]["error"]> 0 ){
13     echo "ERROR.\n";
14 }
15 else{
16     #echo $_FILES["uploadFile"]["name"]."\n";
17     $fname = $dirPath."/input.txt";
18     #echo $fname."\n";
19     move_uploaded_file($_FILES["uploadFile"]["tmp_name"], $fname);
20     shell_exec("chmod -Rf 777 $dirPath");
21 }
22
23 echo "<script type='text/javascript'>window.location.href='http://Your Virtual Machine IP:8080/cloudsystem/view.php?jobId=".$dirname."'</script>";
24
25 ?>
```

改 virtual machine 的 ip

# 3. 部屬 計算節點

- 3-1 docker pull
- 3-2 相關設定

# 3-1 docker pull

```
docker run -i -t -d --name computingNode --privileged=true --volumes-from webServer ubuntu:18.04
```

(上週的slide)

step2. 掛載 myDB 內的 /dbdata

```
docker run -it --volumes-from myDB --name myMachine1 ubuntu
```

```
docker run -it --volumes-from myDB --name myMachine2 ubuntu
```

```
docker run -it --volumes-from myMachine1 --name myMachine2 ubuntu
```

==> myMachine2 也可以透過掛載 myMachine1 來共用空間

myDB、myMachine1 都 不需在執行狀態

docker attach **computingNode**

```
apt-get update
```

```
apt-get install vim php cron
```

```
Please select the geographic area in which you live. Subsequent configuration questions will narrow this
down by presenting a list of cities, representing the time zones in which they are located.
```

```
  1. Africa   3. Antarctica  5. Arctic   7. Atlantic  9. Indian   11. SystemV  13. Etc
  2. America  4. Australia  6. Asia     8. Europe    10. Pacific  12. US
Geographic area: 6
```

選6.Asia

```
Please select the city or region corresponding to your time zone.
```

```
  1. Aden      16. Brunei    31. Hong_Kong  46. Kuala_Lumpur  61. Pyongyang  76. Tehran
  2. Almaty    17. Chita     32. Hovd       47. Kuching        62. Qatar       77. Tel_Aviv
  3. Amman     18. Choibalsan 33. Irkutsk    48. Kuwait         63. Qostanay    78. Thimphu
  4. Anadyr    19. Chongqing 34. Istanbul   49. Macau          64. Qyzylorda   79. Tokyo
  5. Aqtau     20. Colombo   35. Jakarta    50. Magadan        65. Rangoon     80. Toms
  6. Aqtobe    21. Damascus  36. Jayapura   51. Makassar       66. Riyadh      81. Ujung_Pandang
  7. Ashgabat  22. Dhaka     37. Jerusalem  52. Manila         67. Sakhalin    82. Ulaanbaatar
  8. Atyrau    23. Dili      38. Kabul      53. Muscat         68. Samarkand   83. Urumqi
  9. Baghdad   24. Dubai     39. Kamchatka  54. Nicosia        69. Seoul       84. Ust-Nera
  10. Bahrain  25. Dushanbe  40. Karachi    55. Novokuznetsk  70. Shanghai    85. Vientiane
  11. Baku     26. Famagusta 41. Kashgar    56. Novosibirsk   71. Singapore   86. Vladivostok
  12. Bangkok  27. Gaza      42. Kathmandu  57. Omsk           72. Srednekolymsk 87. Yakutsk
  13. Barnaul  28. Harbin    43. Khandyga   58. Oral           73. Taipei      88. Yangon
  14. Beirut   29. Hebron    44. Kolkata    59. Phnom_Penh     74. Tashkent     89. Yekaterinburg
  15. Bishkek  30. Ho_Chi_Minh 45. Krasnoyarsk 60. Pontianak     75. Tbilisi      90. Yerevan
Time zone: 73
```

選73.Taipei，完成後可以按ctrl+P+Q離開容器

# 4. 測試範例

## WebServer:

```
cd /var/www/html/cloudsystem/computer/;
```

```
mv compute.php /share
```

- upload.php 為一個上傳介面
- exe.php 會把上傳的檔案存放到 /share
- view.php 每3秒刷新一次看上傳的檔案是否計算完畢，需要get參數
- <http://你的IP:8080/cloudsystem/view.php?jobId=1556695153.33>
- 若還沒顯示Queuing，若完成輸出md5編碼



## ComputingNode

`docker attach computingNode`

(請根據你前面的命名打)

```
mkdir /cloudsystem
```

```
mv /share/compute.php /cloudsystem
```

```
chmod -Rf 777 /cloudsystem
```

```
crontab -e
```

- 按下i進入編輯模式
- 輸入 `*/1 * * * * /usr/bin/php /cloudsystem/compute.php`
- 按下ESC輸入:wq!

```
/etc/init.d/cron start
```

`compute.php` 把讀到的檔案內容做md5編碼並寫入檔案

# Crontab 排程管理員

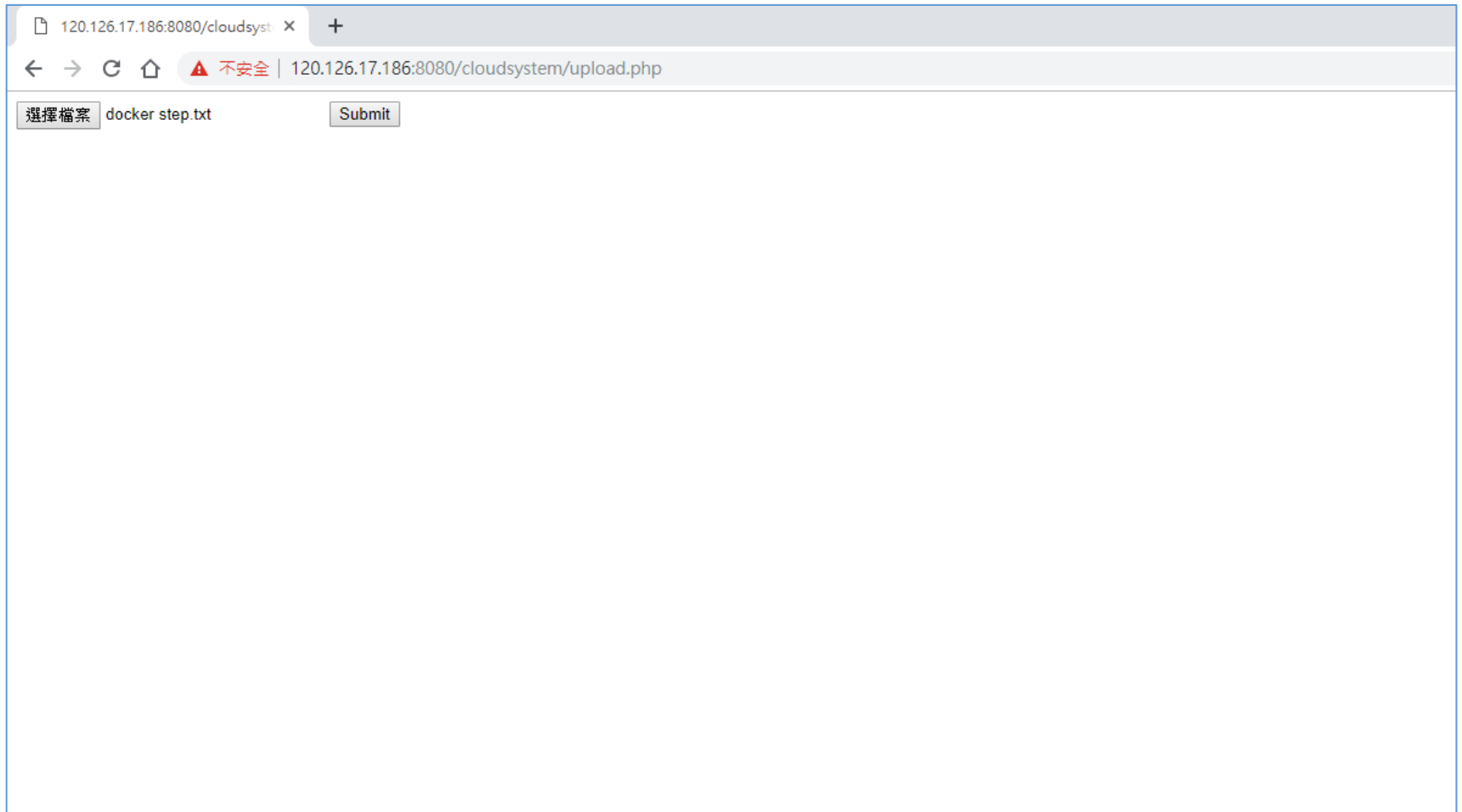
- 執行的程式執行的檔案都要是絕對路徑，且權限要為777

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
*/1 * * * * /usr/bin/php /cloudsystem/compute.php
~
~
~
```

- 打開瀏覽器輸入
  - <http://你的IP:8080/cloudsystem/upload.php>
- 上傳一個文字檔，按送出

computer 會把你上傳的檔案內容利用md5編碼存起來  
在view.php會把編碼過的內容顯示出來  
也可以直接到share裡面看上傳的資料跟處理過的資料

# upload.php



# 作業 mini project 1

請根據前面的範例，

- 提供一種計算服務
- 新增二個計算節點(總共三個計算節點)
- 製作一個web版工作管理員
- 監看目前有多少工作 執行完成、執行中(標明正在哪一個節點中計算)、排隊中
- 顯示 CPU/記憶體 使用率 (指令: `top -bn 1 -i -c`) ==> 計算節點的數據
- 允許使用者把還未執行的工作刪除
- 加分項:
  - 計算節點當機偵測、處理方法、工作重新分配、剔除計算節點、...
  - 跨電腦
- 4/19 11:00前，跟老師討論選題
- demo: 5/3 上課期間
- 每組8-10min、製作投影片、實機 demo
- e-learning 上傳 deadline: 5/3 9:10

# 投票評分

- 全班投票票選評分，各組按分數排名
- 給挖礦獎勵
- 5000, 3000, 1000

過往作業



# 雲端系統DOCKER報告

第8組

B0429017 陳 蕾

B0429034 林睿文

B0429040 王昀筠



SERVICE PURPOSE



## SERVICE PURPOSE

---

資料查詢、抓取、處理

使用者搜尋關鍵字及數量

即可從icon網站下載多張圖片

自動將所有下載的圖片壓縮成zip檔，供使用者下載。

# SERVICE PURPOSE

輸入圖片關鍵字  
和數目

icon 圖片打包

關鍵字 milk

總數目 50

送出

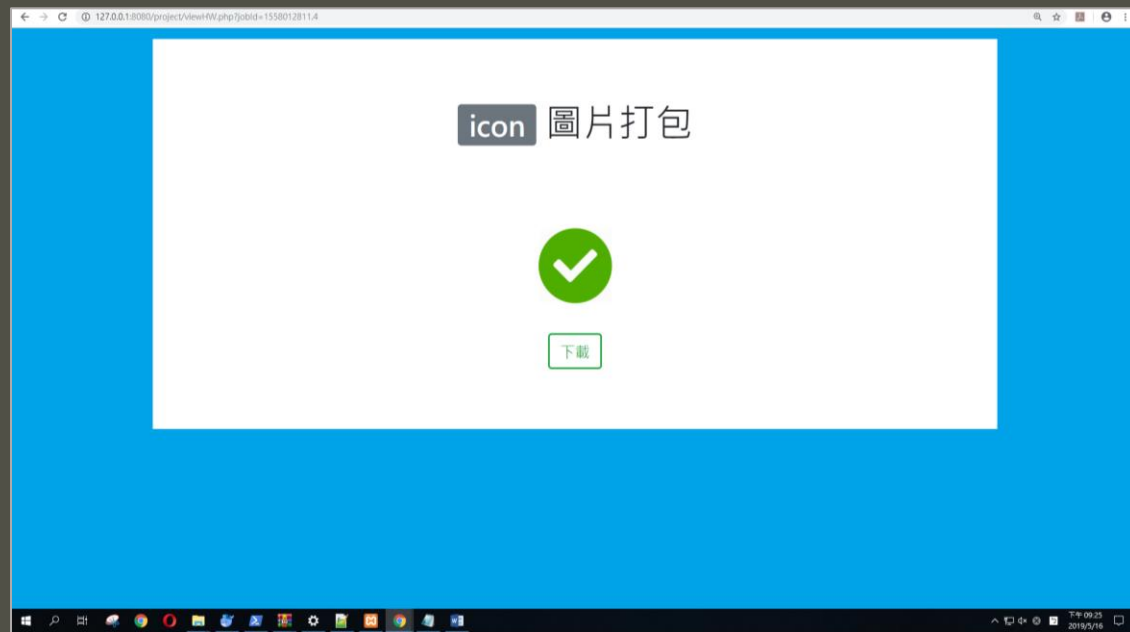
# SERVICE PURPOSE

---

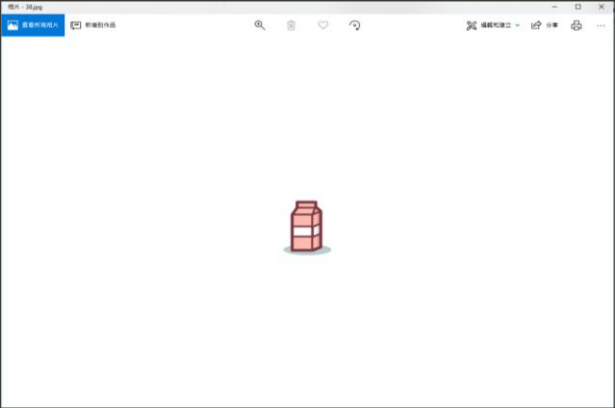
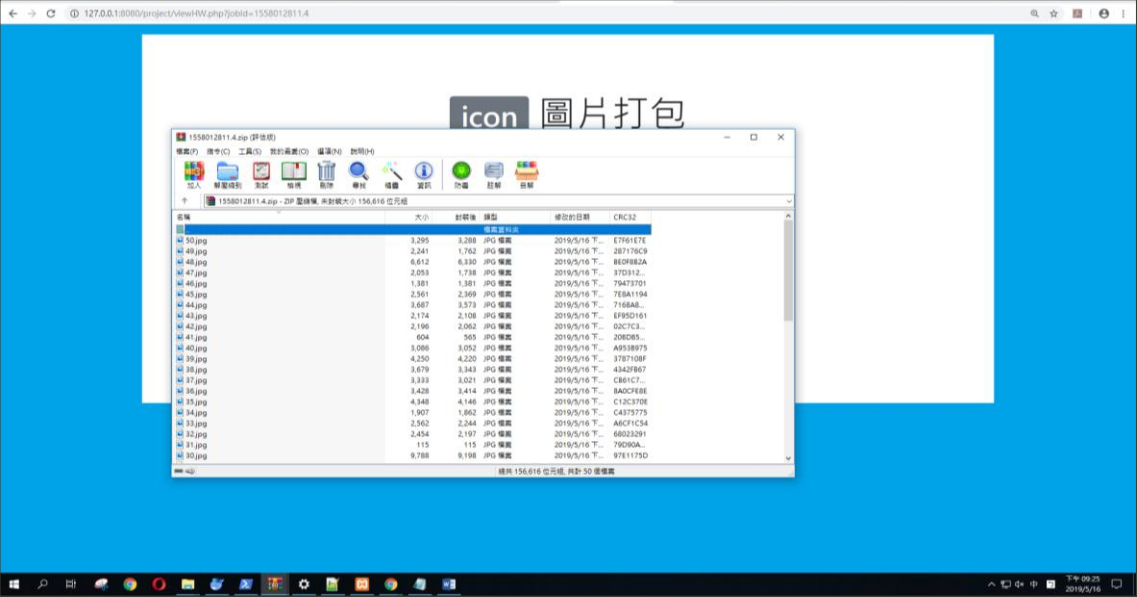


# SERVICE PURPOSE

---



# SERVICE PURPOSE



WORKFLOW



# WORKFLOW

---

## Linux as host OS

Docker1: Ubuntu WS2

home.php  
exeHW.php  
viewHW.php  
downloadZipHW.php  
hwView.php  
remove.php



Docker2: Ubuntu CS2

getHtmlHW.php



Share Folder

Jobid\_1    input.txt  
              computing.txt  
              output.txt

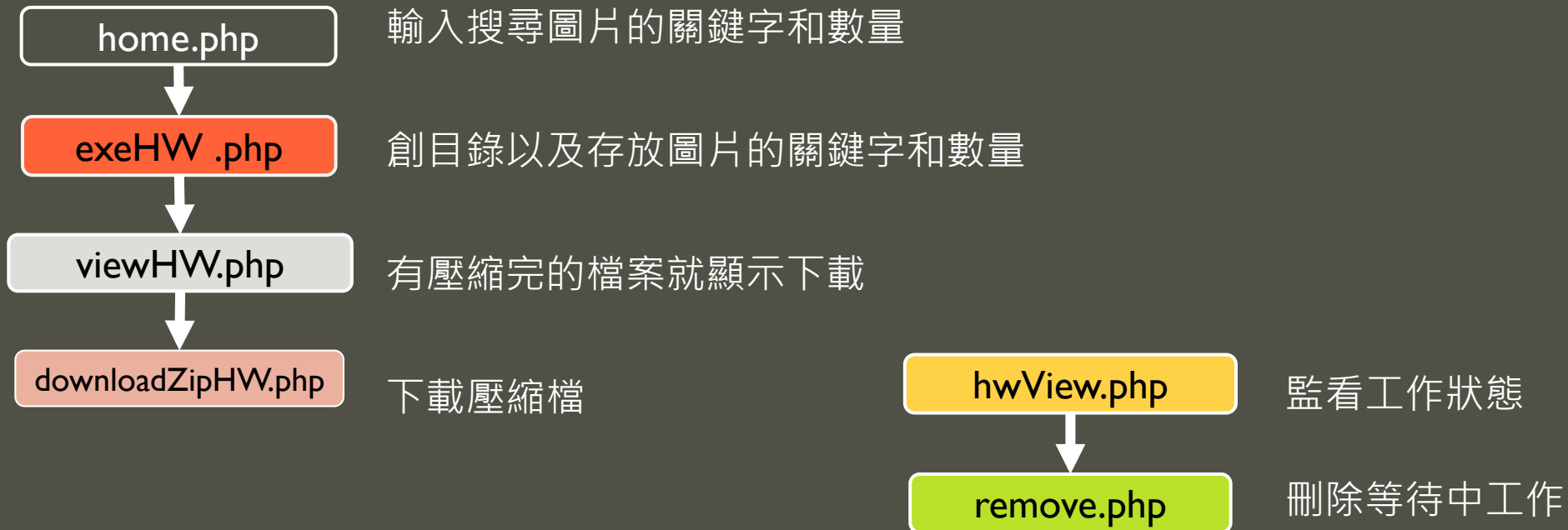
Jobid\_2    input.txt  
              computing.txt



# WORKFLOW

---

## webServer



# SYSTEM MIGRATION

---

## Export匯出

`docker export -o output.tar name`

**output** : 匯出之後的.tar的名字

**name** : 想匯出的container名字

## Import匯入

`docker import input.tar name`

**input** : 想匯入的.tar的名字

**name** : 匯入之後的image名字

## 會入之後會是Image最後再Run來啟動

`docker run -t -i -d --name=newname name:latest bin/bash`

**name** : 想匯入的image名字

**name** : 匯入之後的container名字

MULTIPLE USES



# MULTIPLE USES

多個人同時執行工作，平台使用queue決定執行工作的先後順序，因此可達成多人使用的環境。

icon 工作狀態

	排隊中	執行中	已完成	刪除工作
1558112135.3			V	
1558120983.54			V	
1558120987.85		V		
1558120992.12	V			刪除
1558120994.31	V			刪除
1558120996.65	V			刪除
1558120998.52	V			刪除
1558121006.56	V			刪除
1558121013.63	V			刪除
1558121017.73	V			刪除
1558121019.91	V			刪除
1558121022.30	V			刪除
1558121026.38	V			刪除

top - 23:25:14 up 3:21, 0 users, load average: 0.01, 0.01, 0.00

Tasks: 11 total, 1 running, 10 sleeping, 0 stopped, 0 zombie

%Cpu(s): 1.6 us, 1.5 sy, 0.2 ni, 96.3 id, 0.3 wa, 0.0 hi, 0.1 si, 0.0 st

KiB Mem : 985076 total, 79468 free, 661108 used, 244500 buff/cache

KiB Swap: 998396 total, 727548 free, 270848 used. 164312 avail Mem

遇到的困難與解決辦法



## 遇到的困難與解決辦法

---



**docker** 的**container** 沒有安裝程式需要的指令  
安裝指令後即可執行。



**瀏覽器渲染html**

直接看抓下來的html，而不看原碼。



在**linux**中無法使用**php**抓**html**

改用**command line** 下指令即可解決。

# 自建雲專題

第9組    B0429008 李珮瑄  
            B0429030 鍾岳蓉  
            M0729021 吳淳羽

# Service purpose

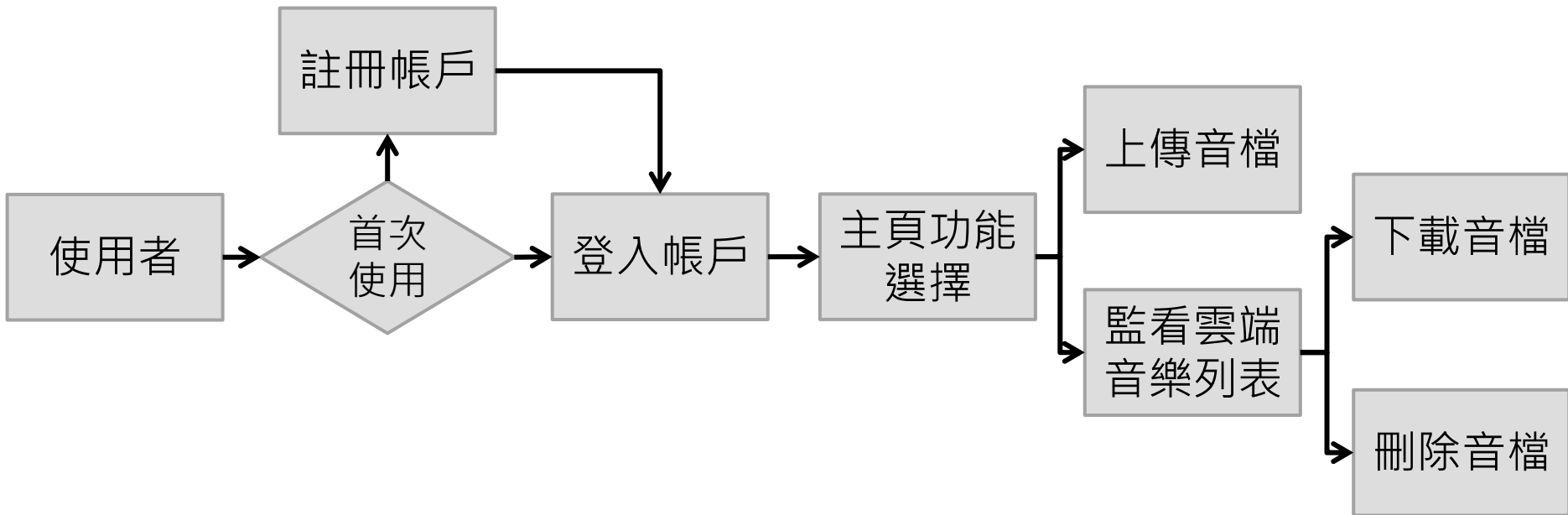
40

- 提供使用者建立屬於自己的音樂下載列表，除了使用者上傳的音樂，也可以取得平台提供的經過變音處理的音樂。



# Workflow

41



# Implementation environment

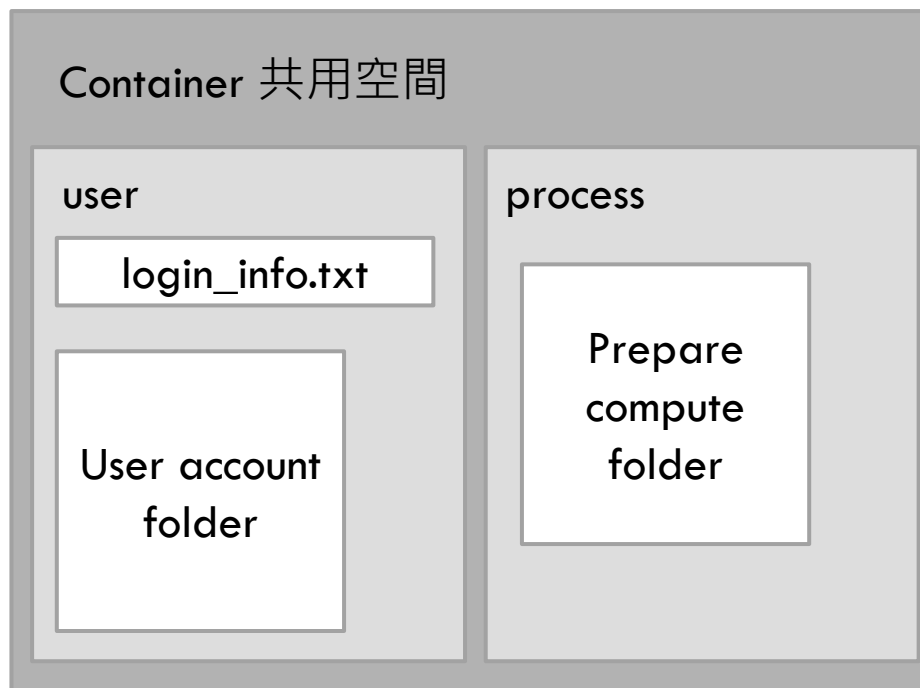
42

- 在Vmware建置Ubuntu16.04
- 在作業系統上的docker建置Ubuntu18.04
- Docker上建2個container
  - WebServer
    - apache2
  - computingNode
    - Cron
    - Ffmpeg
    - Python3.6.4
    - pydub

# Database

43

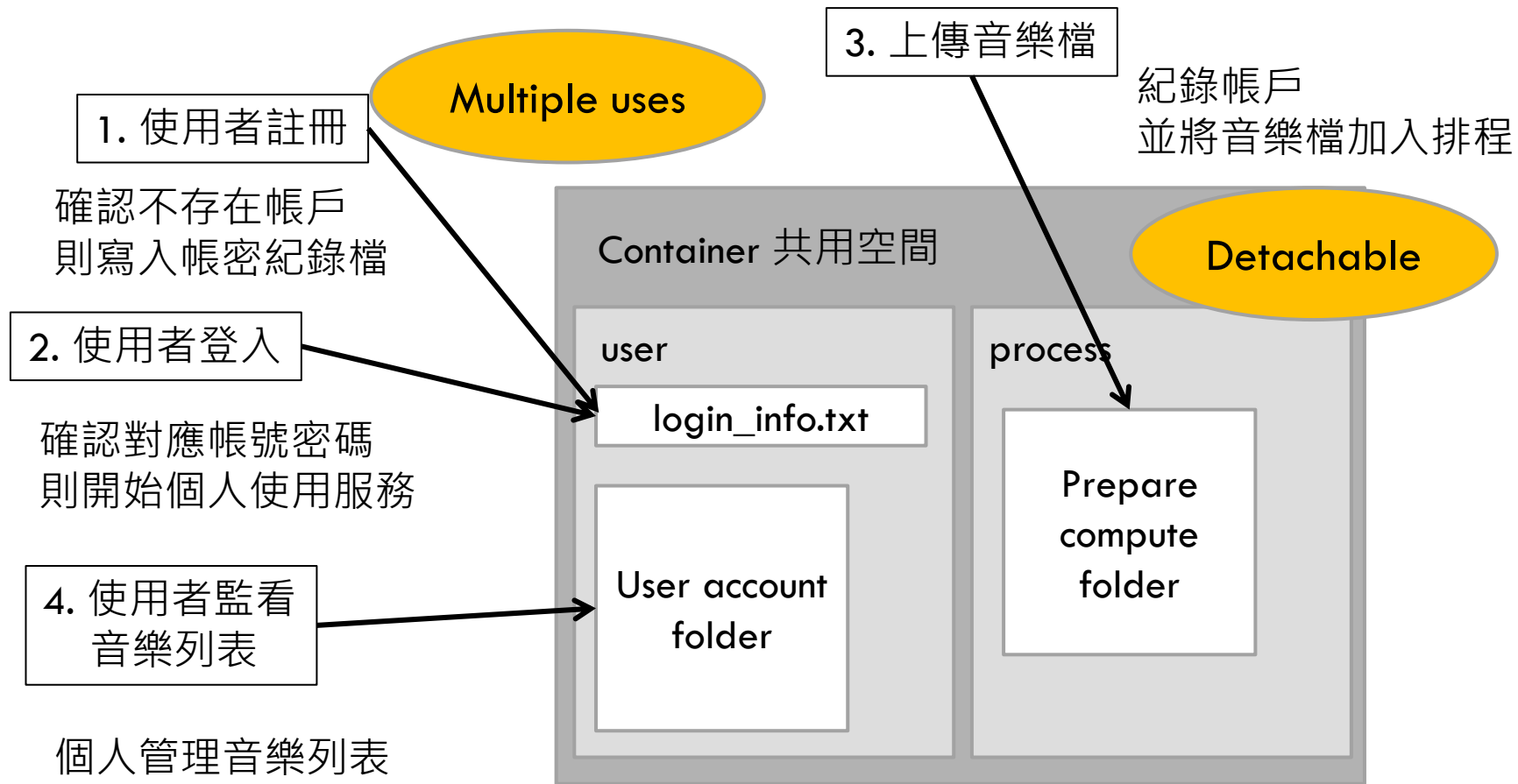
用於使用者  
註冊的紀錄以及  
個人資料的保存



用於將網頁  
上傳的資料  
依據時間  
進行排程

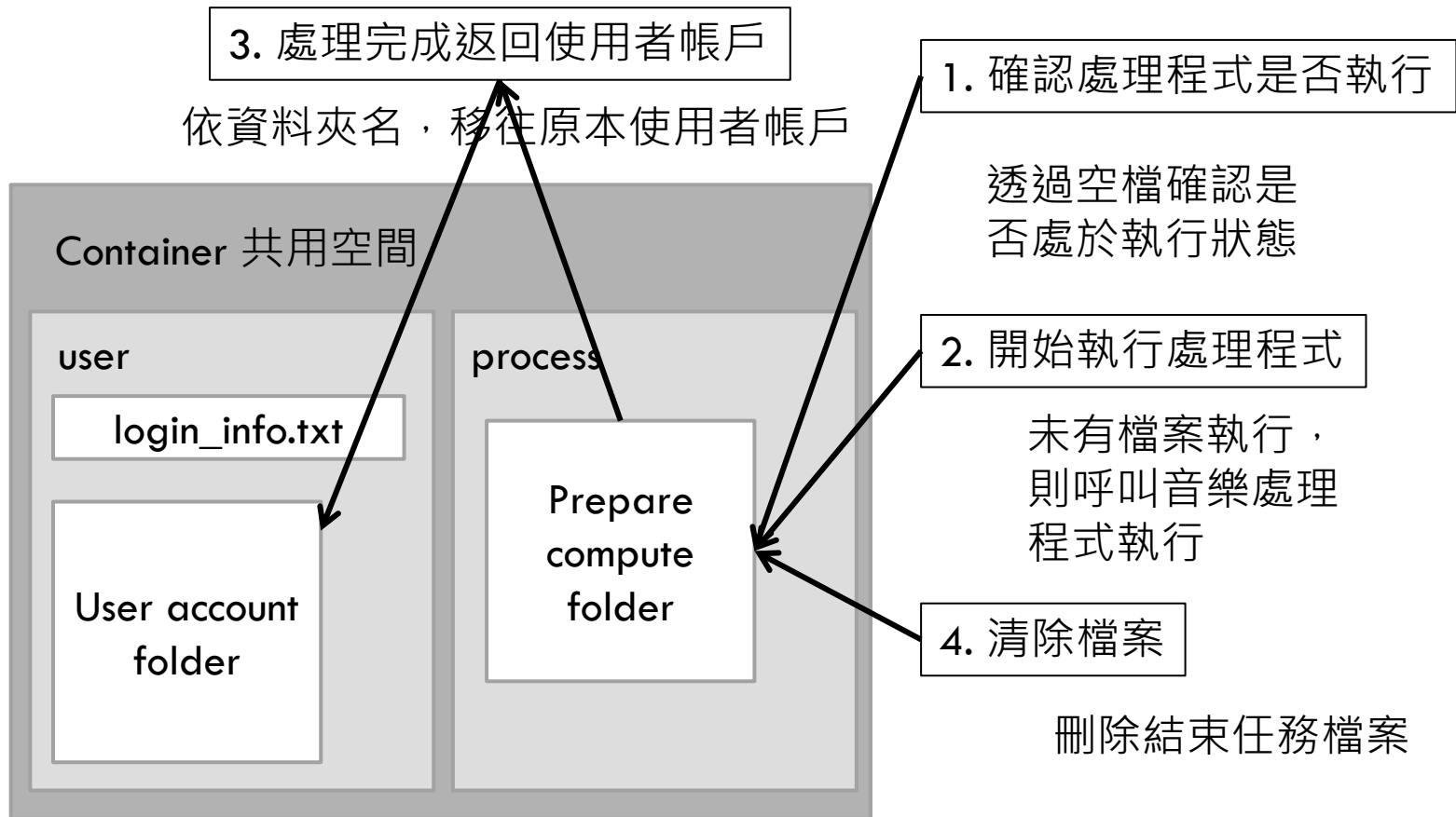
# Container-WebServer

44



# Container-computingNode

45



# 雲端服務

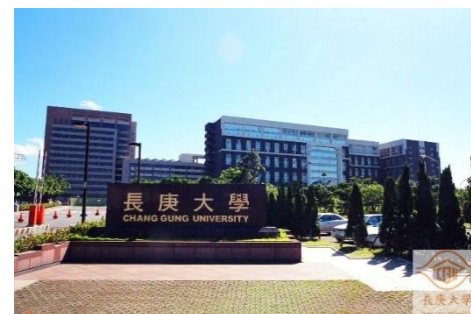
## 自建雲專題

B0429004 溫佳翔

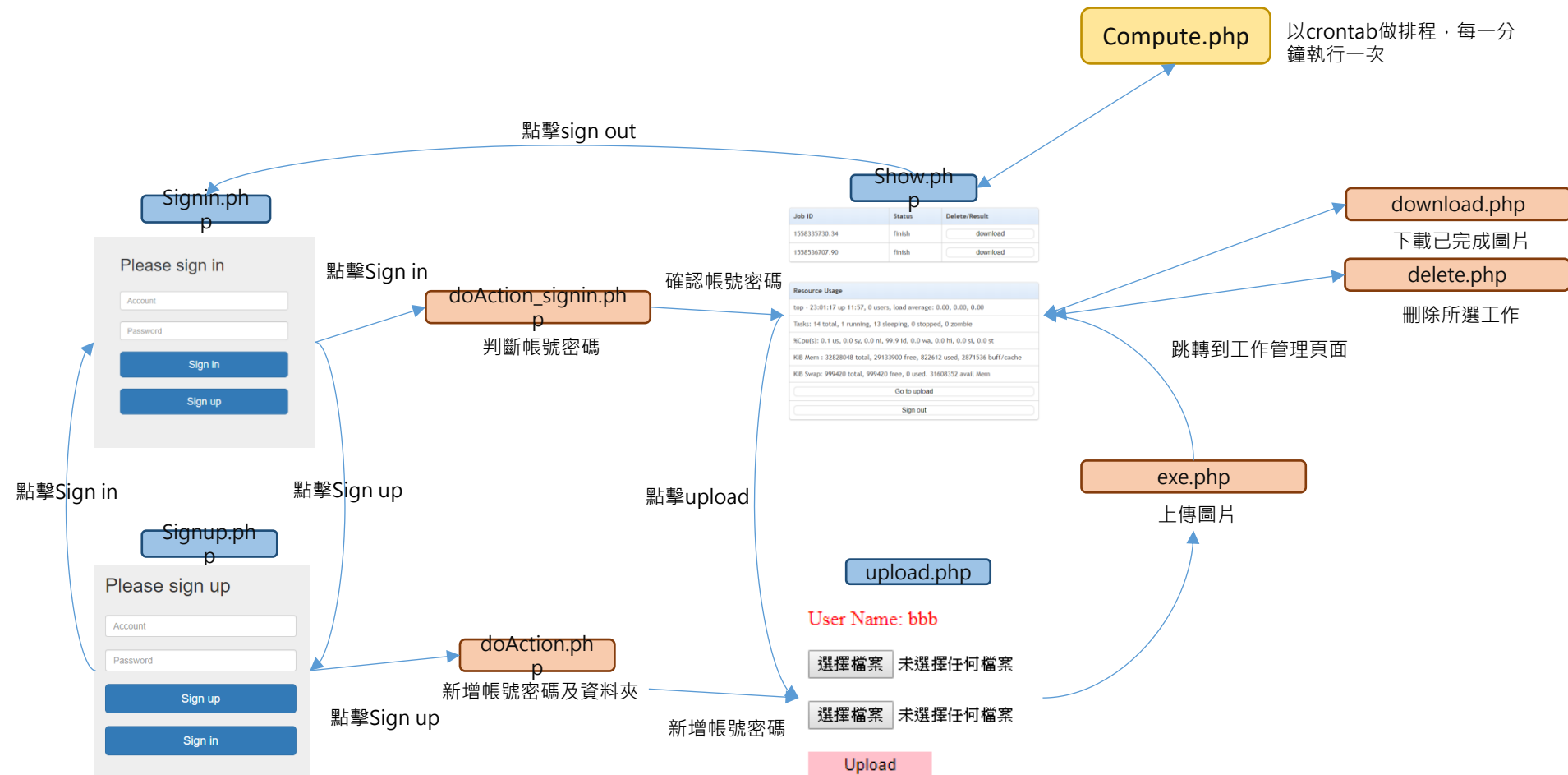
B0429056 吳承軒

# 1. Service purpose

- 使用者可上傳兩張圖片（限制JPG格式），雲端會將第二張以浮水印的方式印在第一張圖片的右下角。



## 2. Workflow





# 上傳頁面

User Name: aaa

選擇檔案 a\_2.jpg



選擇檔案 下載.jpg




長庚大學

Chang Gung University

Upload

# 管理介面

Job ID	Status	Delete/Result
1558539168.78	finish	<a href="#">download</a>
1558540954.19	running	
1558540961.59	queueing	<a href="#">delete</a>

Resource Usage
top - 00:03:07 up 12:59, 0 users, load average: 0.00, 0.01, 0.00
Tasks: 14 total, 1 running, 13 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.1 us, 0.0 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 32828048 total, 29014080 free, 835104 used, 2978864 buff/cache
KiB Swap: 999420 total, 999420 free, 0 used. 31593244 avail Mem
<a href="#">Go to upload</a>
<a href="#">Sign out</a>

## 雲端第3組

B0429006 邱尹俞

B0429038 江和寰

B0429057 徐伯愷

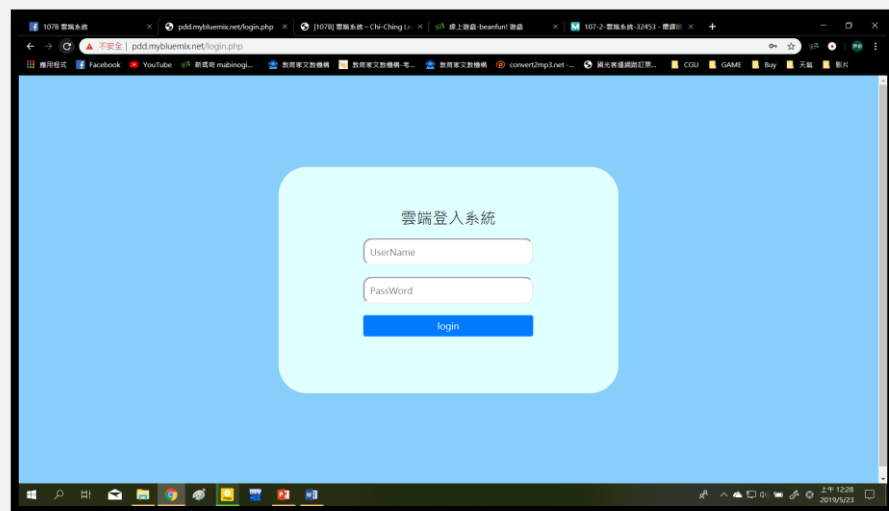
## 目的

- 提供使用者擁有自己的雲端(帳號密碼)
- 提供使用者上傳檔案雲端
- 幫助使用者對資料做一個加密的動作，並且可以下載
- 並也透過我們的網站利用相同密碼解密，取得原始檔案
- 使用者離開雲端依然可以繼續執行工作，下次登入即可下載

## 實作方法

- 我們使用cloud foundry的技術來實作這次的專題，再安裝cflocal的套件讓我們的程式能夠在local端的Docker中執行。
- 因為Cloud foundry是RUN在Docker中更精簡的環境，因此無法執行System service，例如Cron，所以我們利用Bluemix裡面提供的Cloud Function來做我們程式的排程功能，我們的設計是每分鐘Post到compute.php裡面來讀取檔案並且加密後再把執行完的結果寫入新的檔案中。
- 我們是以RC4(Riverst Cipher 4)這個演算法來達到這次Project的加密功能，我們會選擇RC4演算法而不是用其他加密演算法來實作這次的功能，是因為RC4的特性：即為使用同一個密碼加密兩次後會等於原始的檔案。
- 我們的使用者介面利用了許多CSS的樣式來美化我們的UI。

登入



## 上傳檔案

System Info

logout

選擇檔案

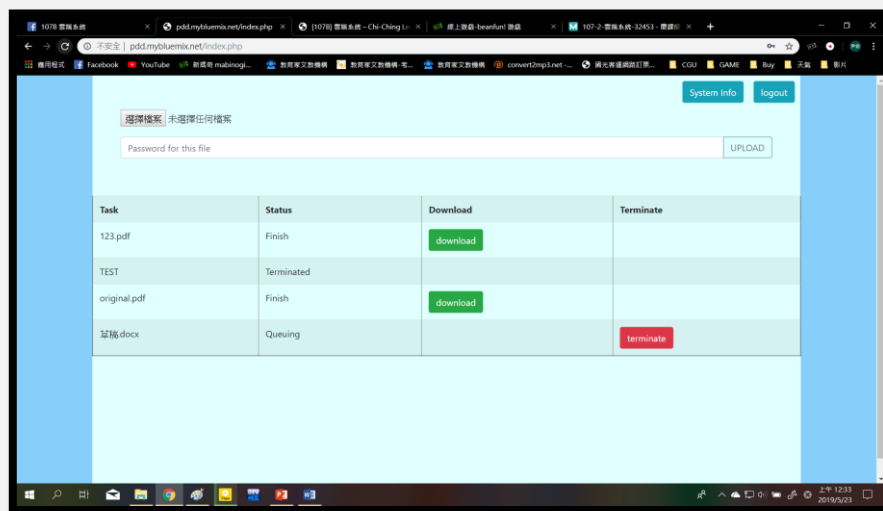
未選擇任何檔案

Password for this file

UPLOAD

Task	Status	Download	Terminate
123.pdf	Finish	<div>download</div>	
TEST	Terminated		
original.pdf	Finish	<div>download</div>	

## 功能(下載、中止)





加密後的檔案，無法開啟

