

在群暉 NAS 上使用「 Docker 」跑新版 python

Docker 簡介 (維基百科)

Docker [編輯]

維基百科，自由的百科全書



建議將 **Kubernetes in docker** 併入此條目或章節。 (討論)

Docker 是一個開放原始碼軟體，是一個開放平台，用於開發應用、交付 (shipping) 應用、執行應用。 Docker 允許用戶將基礎設施 (Infrastructure) 中的應用單獨分割出來，形成更小的顆粒 (容器) ，從而提高交付軟體的速度。 [1]

Docker 容器與虛擬機器類似，但二者在原理上不同。容器是將作業系統層虛擬化，虛擬機器則是虛擬化硬體，因此容器更具有可攜式性、高效地利用伺服器。容器更多的用於表示軟體的一個標準化單元。由於容器的標準化，因此它可以無視基礎設施 (Infrastructure) 的差異，部署到任何一個地方。另外， Docker 也為容器提供更強的業界的隔離相容。 [2]

Docker 利用 Linux 核心中的資源分離機制，例如 **cgroups**，以及 Linux 核心命名空間 (namespaces)，來建立獨立的容器 (containers)，這可以在單一 Linux 實體下運作，避免啟動一個虛擬機器造成的額外負擔 [3]。Linux 核心對命名空間的支援完全隔離了工作環境中應用程式的視野，包括行程樹、網路、用戶 ID 與掛載檔案系統，而核心的 **cgroup** 提供資源隔離，包括 **CPU**、**記憶體**、**block I/O** 與網路。從 0.9 版本起， Docker 在使用抽象虛擬是經由 **libvirt** 的 **LXC** 與 **systemd - nspawn** 提供介面的基礎上，開始包括 **libcontainer** 函式庫做為以自己的方式開始直接使用由 Linux 核心提供的虛擬化的設施，依據行業分析公司「451 研究」：「 Docker 是有能力打包應用程式及其虛擬容器，可以在任何 Linux 伺服器上執行的依賴性工具，這有助於實現靈活性和可攜式性，應用程式在任何地方都可以執行，無論是公用雲、私有雲、單機等。」 [4]。

目錄 [隱藏]

1 基礎架構

1.1 Docker 引擎

1.2 Docker 註冊中心

1.3 物件

2 擴充架構

2.1 Docker Compose

Docker



原作者 Solomon Hykes

開發者 Docker, Inc.

初始版本 2013年3月13日

穩定版本 19.03.13 (2020年9月17日，14天前)

預覽版本 19.03.13-beta2 (2020年8月4日，58天前)

原始碼庫 github.com/docker/docker-ce

程式語言 Go

作業系統 Linux、Windows、macOS

系統平台 x86-64、ARM、s390x、ppc64le

類型 作業系統層虛擬化

授權條款 可執行檔：免費增值軟體即服務

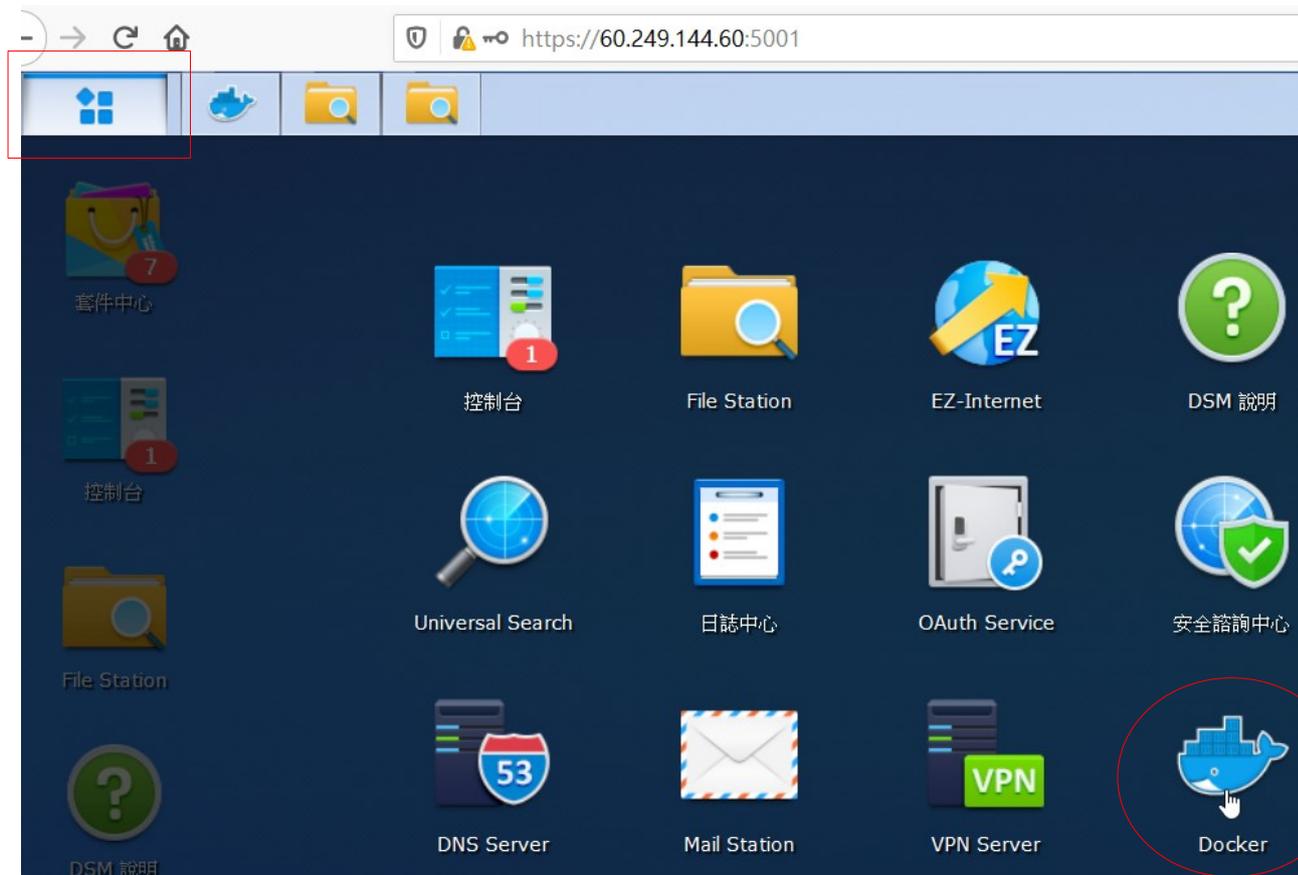
原始碼：Apache 授權條款 2.0

網站 www.docker.com

Docker 套件下的 Python

- 在 docker 套件下安裝新版 python 直譯器
 - 跑 python 程式的另一選擇，因為群暉的「官方」python3 套件只支援到 3.5.1 版
- 在群暉 NAS 上的 docker 套件下試著執行 python 程式
 - SSH 連線：沒切換至 bash shell
 - 無法完全控制路徑設定 (移動路徑、檔案讀取或存檔都非預期)
 - 這可能阻礙了工作排程表的自動化實現 (程式可以跑，但過程或結果並非預期)
 - SSH 連線：假設 container 已經在跑，進入帳號後執行下面指令
 - `sudo docker container exec -it python1 bash`
 - 切換至 bash shell 後，各項 linux 指令的操作和 python 程式的執行都 OK
 - docker 套件下的圖形化介面
 - 在群暉 docker 套件下的 python 視窗可以順利執行 python 程式

準備進入 Docker(之後跑 container)



Python Container (image 已下載)

Docker Desktop interface showing container management. The 'python1' container is highlighted with a red circle around the start/stop toggle.

Container Name	Image	CPU	RAM	Status
mongodb	mongo:4.1.13			停止
oddrationalle-docker-shadowso...	oddrationalle/docker-shadowsocks:latest	0%	13 MB	執行中 Up for 5 days
python1	python:3.8			停止

(1) 準備啟動

Docker Desktop 'Terminal' tab for the 'python1' container. A terminal prompt '>>>' is visible.

(3) 終端機：預設

Docker Desktop 'Terminal' tab for the 'python1' container. A terminal prompt 'root@python1:/#' is visible.

(4) 新增 bash shell

Docker Desktop 'Overview' tab for the 'python1' container. It shows system information like CPU and RAM usage.

Item	Value
執行時間	1 分前
桌面捷徑	已停用
CPU 優先權	中
記憶體上限	1 GB
執行指令	python3

Resource	Usage
CPU 使用率	0%
RAM 使用率	3 MB

Environment Variable	Value
PATH	/usr/local/bin:/usr/local/s...
LANG	C.UTF-8
GPG_KEY	E3FF2839C048B25C084...
PYTHON_VERSION	3.8.6
PYTHON_PIP_VERSION	20.2.3

(2) 進入 container 後的概況

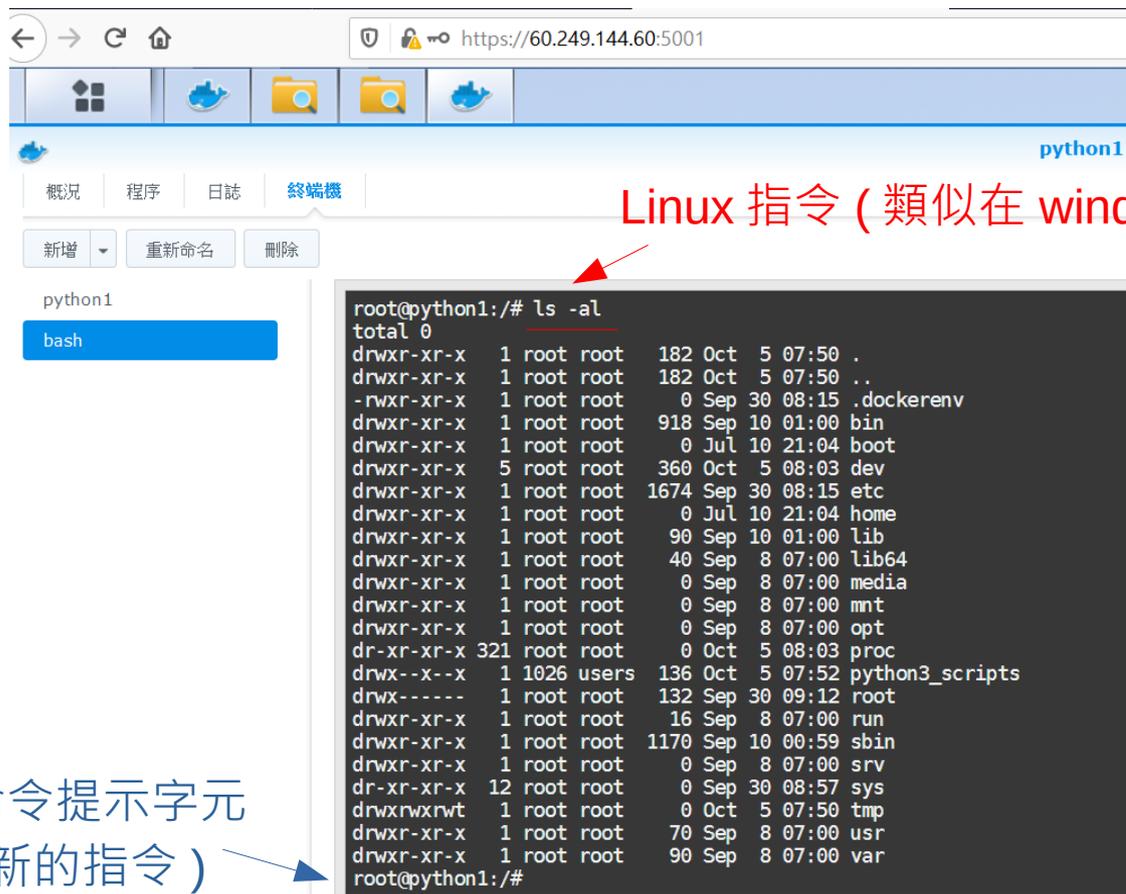
Docker Desktop 'Edit' tab for the 'python1' container. It shows path settings for 'homes/Administrator/python3_scripts' and '/python3_scripts/'.

(0) 設定好對應路徑

群暉作業系統的目錄

Python container 內的目錄

在 Python Container 內下指令



The screenshot shows a Docker container named 'python1' with a terminal window open. The terminal prompt is 'root@python1:/#'. The command 'ls -al' has been executed, resulting in a detailed listing of the root directory. The output shows various system files and directories, including '.dockerenv', 'bin', 'boot', 'dev', 'etc', 'home', 'lib', 'lib64', 'media', 'mnt', 'opt', 'proc', 'python3_scripts', 'root', 'run', 'sbin', 'srv', 'sys', 'tmp', 'usr', and 'var'. The permissions, owner, and group for each file are also listed.

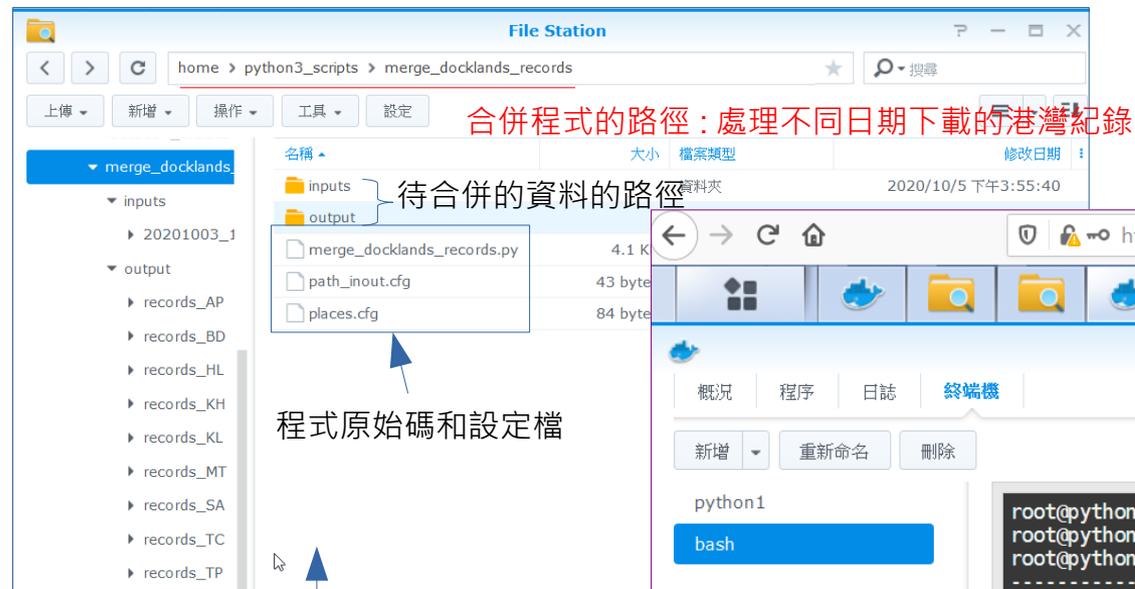
```
root@python1:/# ls -al
total 0
drwxr-xr-x 1 root root 182 Oct 5 07:50 .
drwxr-xr-x 1 root root 182 Oct 5 07:50 ..
-rwxr-xr-x 1 root root 0 Sep 30 08:15 .dockerenv
drwxr-xr-x 1 root root 918 Sep 10 01:00 bin
drwxr-xr-x 1 root root 0 Jul 10 21:04 boot
drwxr-xr-x 5 root root 360 Oct 5 08:03 dev
drwxr-xr-x 1 root root 1674 Sep 30 08:15 etc
drwxr-xr-x 1 root root 0 Jul 10 21:04 home
drwxr-xr-x 1 root root 90 Sep 10 01:00 lib
drwxr-xr-x 1 root root 40 Sep 8 07:00 lib64
drwxr-xr-x 1 root root 0 Sep 8 07:00 media
drwxr-xr-x 1 root root 0 Sep 8 07:00 mnt
drwxr-xr-x 1 root root 0 Sep 8 07:00 opt
dr-xr-xr-x 321 root root 0 Oct 5 08:03 proc
drwx-x-x 1 1026 users 136 Oct 5 07:52 python3_scripts
drwx----- 1 root root 132 Sep 30 09:12 root
drwxr-xr-x 1 root root 16 Sep 8 07:00 run
drwxr-xr-x 1 root root 1170 Sep 10 00:59 sbin
drwxr-xr-x 1 root root 0 Sep 8 07:00 srv
dr-xr-xr-x 12 root root 0 Sep 30 08:57 sys
drwxrwxrwt 1 root root 0 Oct 5 07:50 tmp
drwxr-xr-x 1 root root 70 Sep 8 07:00 usr
drwxr-xr-x 1 root root 90 Sep 8 07:00 var
root@python1:/#
```

Linux 指令 (類似在 windows 下 dir /s 指令)

輸出結果

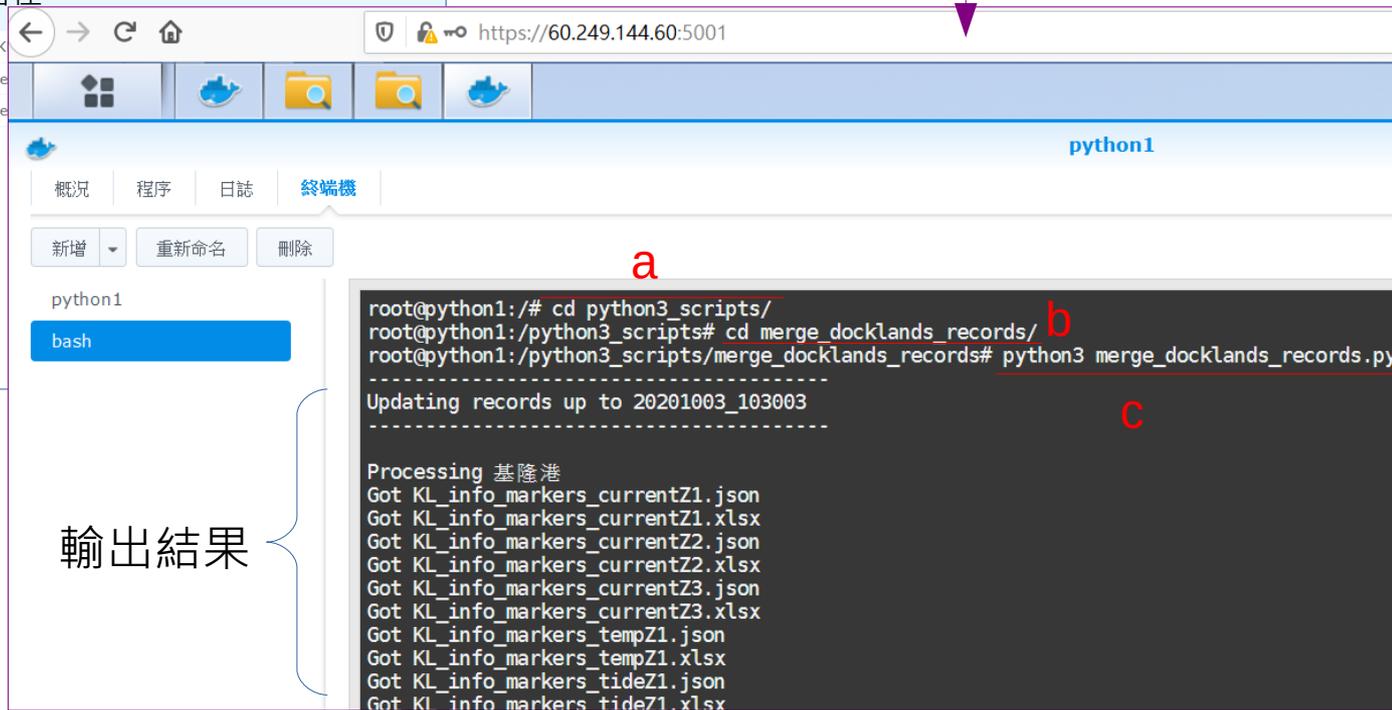
回到命令提示字元
(等待新的指令)

在 container 環境下執行 python 程式



(1) 準備待測的 python 程式
(無法在群暉官方套件下執行)

(2) 在 Command Line Interface
下 linuxc 指令並執行 python 程式



測試 ssh+container 的 python 程式

facebook_articles.py - D:\Python_projects\facebook_articles\facebook_articles.py...

File Edit Format Run Options Window Help

```
1 from facebook_scraper import get_posts
2
3 #for post in get_posts('hwangchinyeong', pages=1):
4 #    print(post['text'])
5 #    print('-----')
6
7 ...
8 for post in get_posts('hwangchinyeong', pages=1):
9
10     tmp_filename = post['time'].isoformat()
11     filename = tmp_filename.replace(':', '') + '.txt'
12
13     with open(filename, mode='wt', encoding='utf-8') as fp_w:
14
15         fp_w.write(post['text'])
16         print('Got ' + tmp_filename)
17     ...
18 |
19 #fb_id = 'hwangchinyeong'
20 #fb_id = '2020blueseas'
21 fb_id = '威騰有限公司-1438424896388286'
22
23 with open(fb_id + '.txt', mode='wt', encoding='utf-8') as fp_w:
24
25     for post in get_posts(fb_id, pages=3):
26
27         try:
28             article_time = post['time'].isoformat()
29         except:
30             continue
31
32         fp_w.write(article_time)
33
34         fp_w.write('\n' + '-----' + '\n')
35
36         fp_w.write(post['text'])
37         fp_w.write('\n\n')
38
39         print('Got ' + article_time)
40
```

左邊的 python 腳本可以下載臉書的文章，並將某用戶的文章存成一個文字檔。

程式用到了 facebook-scraper 這個第三方 library，而這 library 在群暉的官方 python 套件是無法安裝的，相較下，docker 套件下的新版 python 則可以正常安裝它。

左邊 python 腳本存成的檔案會和主程式放在同一目錄，可是在接下來的測試中，卻發現生成檔案的位置不符合預期，這行為和 windows 環境下執行的結果不一致。

ssh 連線跑程式 (工作目錄沒指定)

```
OpenSSH SSH client
C:\Users\Atrisk>ssh Administrator@60.249.144.60
Administrator@60.249.144.60's password:
Administrator@WTGroup:~$ sudo docker container start python1
Password:
python1
Administrator@WTGroup:~$ sudo docker container exec python1 ls -x
bin boot dev etc home lib lib64 media mnt opt proc python3_scripts
root run sbin srv sys tmp usr var
Administrator@WTGroup:~$ sudo docker container exec python1 ls python3_scripts/facebook_articles/
facebook_articles.py
Administrator@WTGroup:~$ sudo docker container exec python1 python3 python3_scripts/facebook_articles/
facebook_articles.py
Got 2020-09-03T09:49:26
Got 2020-08-12T22:43:55
Got 2020-07-27T22:19:06
Got 2020-07-10T03:25:41
Got 2020-07-01T09:41:18
Got 2020-06-30T04:02:02
Got 2020-06-24T05:06:40
Got 2020-06-19T23:46:38
Got 2020-06-11T07:35:12
Got 2020-05-19T03:45:14
Administrator@WTGroup:~$ sudo docker container exec python1 ls python3_scripts/facebook_articles/
facebook_articles.py
Administrator@WTGroup:~$ sudo docker container exec python1 ls -x
bin boot dev etc home lib
lib64 media mnt opt proc python3_scripts
root run sbin srv sys tmp
usr var 威騰有限公司-1438424896388286.txt
Administrator@WTGroup:~$
```

透過 ssh 連上群暉主機並啟動 python1 container

在 container 下跑 python 程式時，程式要顯示的訊息可以正常產生，但是新產生的檔案卻沒和主程式處在同一目錄

原來新生成的檔案被存到 container 的根目錄去了

ssh 連線跑程式 (移動到工作目錄)

命令提示字元
改變了

```
OpenSSH SSH client
Administrator@WTGroup:~$ sudo docker container exec -it python1 bash
root@python1:/# cd python3_scripts/facebook_articles/
root@python1:/python3_scripts/facebook_articles# ls
facebook_articles.py
root@python1:/python3_scripts/facebook_articles# python3 facebook_articles.py
Got 2020-09-03T09:49:26
Got 2020-08-12T22:43:55
Got 2020-07-27T22:19:06
Got 2020-07-10T03:25:41
Got 2020-07-01T09:41:18
Got 2020-06-30T04:02:02
Got 2020-06-24T05:06:40
Got 2020-06-19T23:46:38
Got 2020-06-11T07:35:12
Got 2020-05-19T03:45:14
root@python1:/python3_scripts/facebook_articles# ls
facebook_articles.py 威騰有限公司-1438424896388286.txt
root@python1:/python3_scripts/facebook_articles# cat 威騰有限公司-1438424896388286.txt
2020-09-03T09:49:26
-----
大老級加大砲級委員蒞臨
給了許多意見
回去好好加油，再拼期末

2020-08-12T22:43:55
-----
第一次的嘗試，希望一切順利

2020-07-27T22:19:06
```

切換到 bash shell 後
才有 cd 這個移動目錄
的 linux 指令可用

新生成的檔案終於出現在預期的目錄了

敲 cat 指令顯示新
生成檔案內的內容

將 docker 納入任務排程

(2) 下了三道 docker 命令



概況 程序 日誌 終端機

啟動 停止 重新啟動 強制停止

python_facebook-scraper_only

執行時間: 29 分鐘前
桌面捷徑: 已停用
CPU 優先權: 低
記憶體上限: 32 MB
執行指令: bash

(1) 設定 container

連接埠設定 儲存空間 容器間連結 網路

檔案/資料夾	掛載路徑	類別
homes/Administrator...	/home/	rw

使用者定義指令碼

- A docker container start python_facebook-scraper_only 啟動容器
- B docker container exec python_facebook-scraper_only sh -c "cd /home ; python3 facebook_articles.py ; exit" 用 sh -c 語法下 Linux 指令
- C docker container stop python_facebook-scraper_only 停止容器



控制台

新增 編輯 執行 動作 儲存 設定

(3) 以 root 權限執行任務排程

已啟動	擁有者	應用程式	任務名稱	動作	下次執行時間
<input checked="" type="checkbox"/>	root	使用者定義指令碼	Test docker pyth...	執行: docker co...	2020-10-08 0...

擔心若用 Administrator 的話可能會被要求輸入密碼
· 如此便無法完成任務排程自動化