



實驗四

Deadline: 12/11 11:59pm

LAB4-1 Install and Test Docker

實驗目的

1. 安裝 Docker 及簡單測試

實驗步驟

1. (建議) 依照[官方網站教學](#)[1]安裝 Docker
2. 使用 Docker 安裝 image 指令安裝 lab4_image.tar，並把這 image 跑起來，裡面有“Lab4”文件，用 cat 指令把它印出來，並擷取圖片(4-1)如：

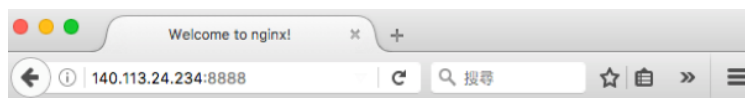
```
fonger@Grapes:~/Lab4$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
lab4                 latest             9ec9e8211f90       10 minutes ago     1.109 MB
ubuntu              latest             bea8f41ae3e8       12 days ago        128.1 MB
<none>              <none>             0d03e8a68b98       3 months ago       1.388 GB
nginx               latest             89732b811e7f       6 months ago       182.8 MB
<none>              <none>             94db582288db       12 months ago      1.355 GB
swarm               latest             6b40fe7724bd       13 months ago      15.6 MB
busybox             latest             c51f86c28340       13 months ago      1.109 MB
fonger@Grapes:~/Lab4$ sudo docker [redacted]
/ # ls
Lab4  bin  dev  etc  home  proc  root  sys  tmp  usr  var
/ # cat Lab4
Hi! This is final lab!
/ #
```

(指令：“docker help”有指令教學)

3. 至 Dock Hub 把 nginx image pull 下來，跑起來 (要做 port forwarding)

並修改此 image 裡面的“/usr/share/nginx/html/index.html”檔案，使自

己的學號顯示在網頁上面，並擷取圖片(4-2)：



Welcome to nginx!

My student id is XXXXX

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

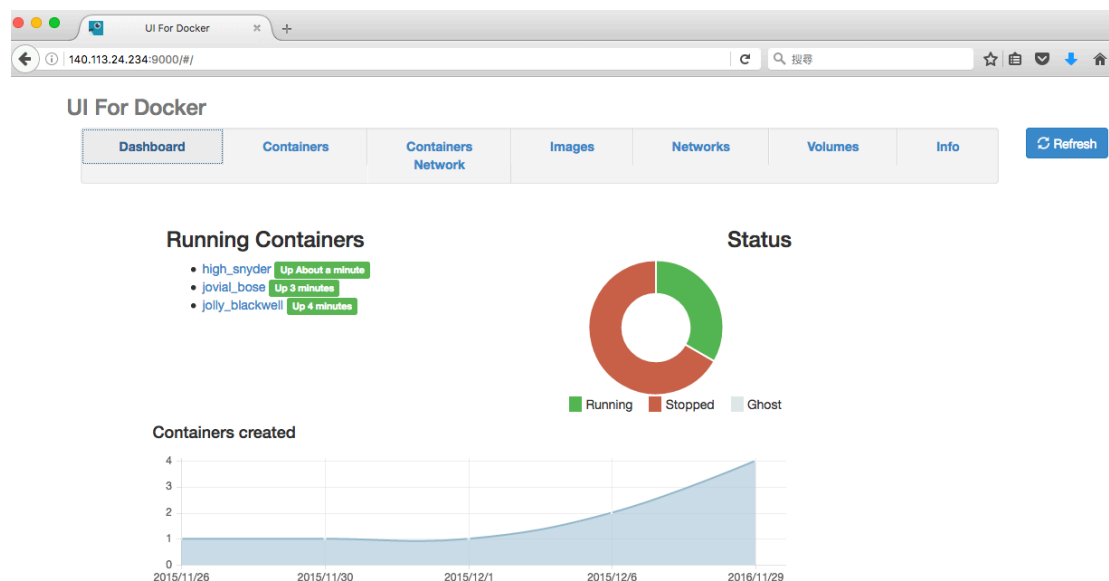
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

4. 至 Docker Hub 把" uifd/ui-for-docker"[2]此 image pull 下來，並跑起來，

這是一個別人做的 Rest API 可以使用 http 對 containers 進行操作，並

擷取此圖(4-3)：



參考文件

1. Docker 官方網站: <https://www.docker.com/>
2. UI-for-docker : <https://github.com/kevana/ui-for-docker>

Deadline: 12/11 11:59pm



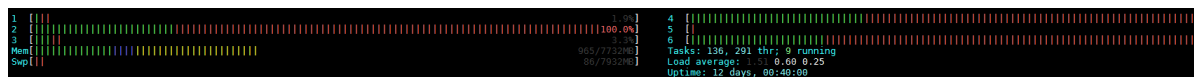
LAB4-2 Docker Resource Control

實驗目的

1. 使用 docker 提供的 resource control command 來限制各個 container 之硬體使用。

實驗步驟

1. 限制一個 container 只能使用雙數號的 cores 並跑 pi.c，並在執行中擷取一張 htop 上 cpu 的狀態(4-4)，如：



2. 請執行 container 並執行“sysbench”，來觀測執行時間，並且完成下表

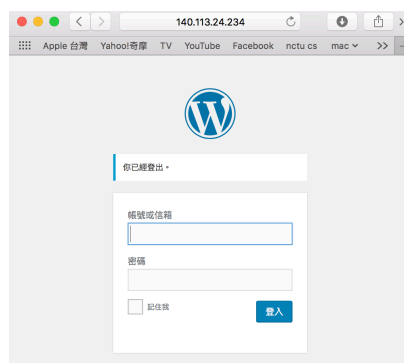
(cmd: sysbench --test=threads --num-threads=**N** --thread-yields=8000 --thread-locks=100 run)

(n 為整台機器的總核心執行緒數，e.g. 4C8T --> n=8)

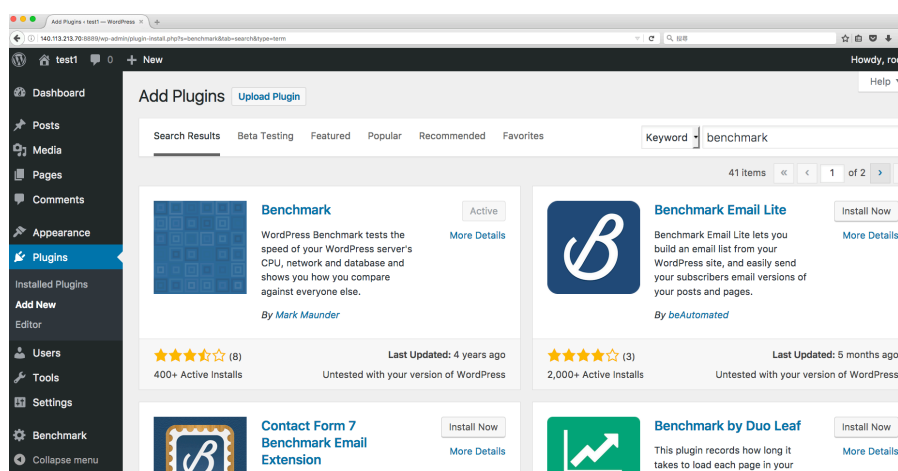
Container 1	Container 2	Execution Time
能使用所有 cores	能使用所有 cores	
只能使用單數號 cores	只能使用雙數號 cores	

並且簡單敘述觀察到的情況，並簡單寫出可能的原因。

3. 練習安裝 wordpress 在一個 container 上，可以參考[教學文件](#)，成功後可以在自己的 browser 上面輸入自己 host-IP 加上自己設定的 port，會出現 wordpress 設定畫面，設定完後就可以登入 (如下圖)



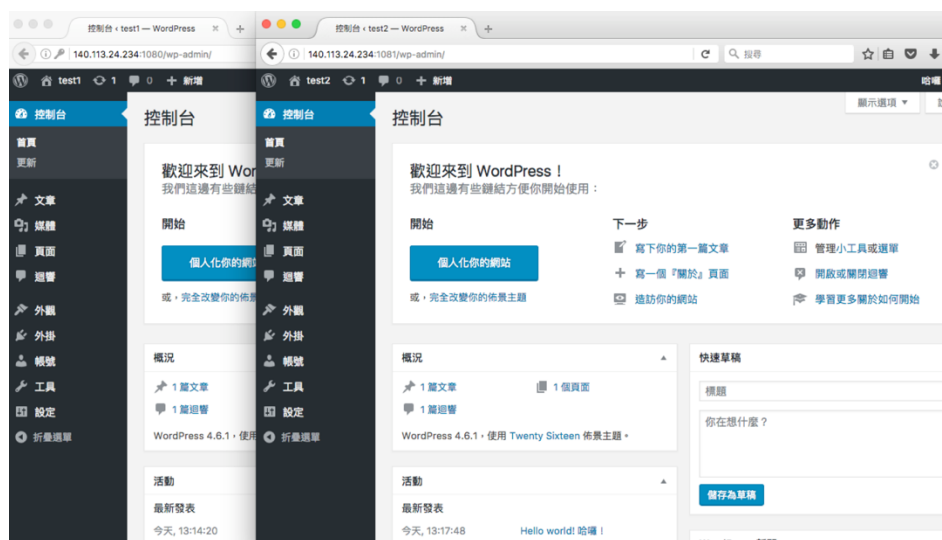
登入後去左邊欄位 plugins 安裝 benchmark 如下圖:



之後把自己安裝好的 container commit 成一個 image，之後要開新的

wordpress container 不需要重新安裝，就可以快速跑起來（下圖是我同

時開兩個 wordpress 的範例）





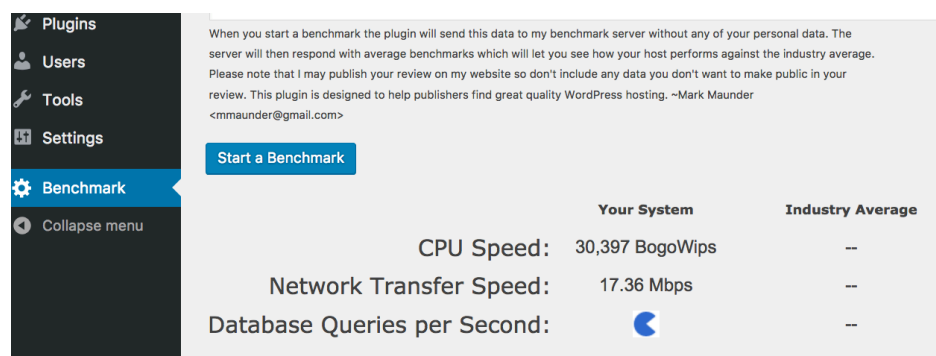
再來也在 vmware or virtual box 上也跑一個 VM，上面安裝 wordpress，

之後也把灌好這 VM 創成一個 template，未來安裝新的 instance 就不

需重新安裝 wordpress。

之後請分別跑 4 個以及 10 個 docker/VM instance，並同時跑剛剛裝的

benchmark plugin 比較資源使用量，以完成下表



	4 wordpress instance	10 wordpress instance
Docker	Cpu/mem/IO	
VM		

CPU/MEM/IO usage 可以從 Lab1 寫的 monitor 去做觀察(亦可以從 http

以及 iotop 取出資料)，觀察到所有的 benchmark 都跑完，1~2 秒取一

次資料做平均。

參考文件

1. <https://docs.docker.com/engine/reference/run/>
2. <https://tw.wordpress.org/>

上傳檔案

- Lab4.pdf