Docker 用途介紹

觀念

1. 建立映像檔(image)來做出特定的開發環境
2. image上可以有不同互不影響的容器(containers)
3. 每個container都是一個process

目的

1. 做出一個隔離的開發環境
2. 方便交付給他人做開發與測試

常用的命令

1. 打包images
2. run images(generate a container)
3. 將資料從container拷貝出來
4. 資料備份與共享

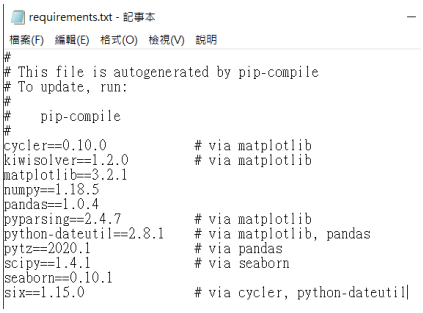
打包的流程

1. 在docker hub上找一個基礎的image做基底
2. 將想要複製的資料夾結構寫進Dockerfile裡面
3. 如果有自己想要額外安裝的套件，寫在requirements.txt裡面
4. 用docker指令build自己的image
5. 例如你想要在python:3這個image做為基底，建立一個工作目錄，叫做my_workspace，且將所有資料夾的資料都從本機帶入image內，且同時要額外安裝某些套件(寫在requirements.txt內)，那你可以在Dockerfile中這樣書寫:

```
FROM python:3

WORKDIR /usr/src/my_workspace

COPY ./ ./

RUN pip install --no-cache-dir -r requirements.txt

EXPOSE 8888
```

6. requirements.txt的位置要跟Dockerfile同一層
   資料夾結構

   requirements.txt內容



7. 以docker build指令建立自己的image

   > docker build -t <image_name:tag> .

   ex: docker build -t my_first_image:0.0.1 .

8. 這樣之後每個基於此image所建立的container，都會有my_workspace這個開發資料夾，且所有套件的版本都會相同。

Run images(generate a container)的流程

1. 找到你想run的image id(可能是在dockerhub pull下來的或自己build的)
2. 以docker指令run，如果是要建立一個jupyterlab的開發環境就要記得產生container連外的街口(port)

```
docker run -it -p 82:8888 6de62981d855 /bin/bash \
-c "jupyter lab --ip='*' --port=8888 --no-browser --allow-root"
```

ps: 自己把\ -c的"\"去掉，且不要斷行。

意思是，你要在在某個image(image id=6de62981d855)上，產生一個container，而且要有想要在container中開啟jupyterlab開發環境，於是產生一個對外接口(8888)連到本機接口(82)，這樣我們在

本地網頁輸入localhost:82就可以連到container內部的開發環境(jupyterlab)

將container檔案拷貝回本機

1. 找到container ID
2. 以docker cp指令把檔案帶回來(也可以把本機資料拷貝到container)
3. 例如你想要把這某container(0936cf2de3b6)中，/usr/src/my_workspace所有檔案都拷貝回本地路徑C:\Users\使用者名稱\Desktop\volume\data_2下，你可以在本地的終端機輸入以下指令:

```
docker cp 0936cf2de3b6:/usr/src/my_workspace C:\Users\電腦使用者名稱\Desktop\volume\data_2
```

資料備份與共享

> 在docker上跑數據時，若刪除了container，則裡面所有產生的資料一般都會一起消失；且往往會碰到輸出數據需要保存或是不同container數據需要共享或交換的情形，docker中有volume的機制，可以將數據以掛載的方式，達成備份與共享。以下是範例:

1. run一個container，名稱叫做container1，且指定本地的路徑做為volume mapping的地址，掛載到container內。

   ```
   docker run --rm -it -v C:\Users\使用者名稱\Desktop\volume_test:/usr/src/my_workspace/volume \
   --name container1 6de62981d855 /bin/bash
   ps:
   1. 6de62981d855是image_id
   2. \ --name ...是換行，執行的時候把\刪掉，不要斷行
   ```

2. 產生第二個container，名稱叫做container2，並且利用--volumes-from的指令，將兩個container的資料作連動，共同指向本地的資料夾。

   ```
   docker run --rm -it --volumes-from container1 --name container2 6de62981d855 /bin/bash
   ps:以上的/usr/src/my_workspace/volume是絕對路徑
   ```

3. 這樣在container中，在my_workspace/volume這個資料夾下產生的數據就可以和container2共享，在container2產生的資料也能和container1共享，而且即使這兩個container都被移除了，資料也不會消失，因為已經存在本機資料夾(C:\Users\使用者名稱\Desktop\volume_test)了。

實作安裝docker及建立container

1. 到 控制台開-程式集-開啟或關閉Windows功能啟動hyper-v(不能啟動找IT)。
2. 輸入指令將資料夾作連動，這樣之後docker就可以裝在D槽，避免C槽塞滿。

```
mklink /j C:\Program Files\Docker D:\Docker\Program Files\Docker
mklink /j C:\ProgramData\Docker D:\Docker\ProgramData\Docker
mklink /j C:\Users\使用者名稱\AppData\Local\Docker C:\Docker\AppData\Local\Docker
mklink /j C:\ProgramData\DockerDesktop D:\Docker\ProgramData\DockerDesktop
```

3. 到官網下載docker安裝檔，並安裝。
4. 進到某個資料夾(看你想要之後image裡面有甚麼資料夾的結構)下，撰寫Dockerfile，範例可以參考上面。
5. run build命令。

```
docker build -t my_first_image:0.0.1 .
```

6. 以下面指令找到建立的image的id。

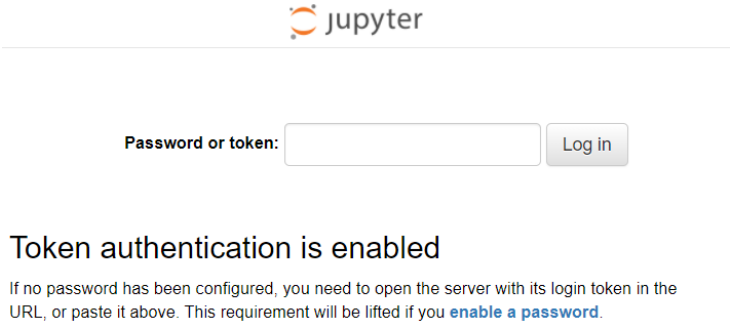
```
docker images
```

7. 以下面指令以該image為基礎，產生container。

```
想要叫出jupyterlab開發:
docker run -it -p 82:8888 6de62981d855 /bin/bash \
-c "jupyter lab --ip='*' --port=8888 --no-browser --allow-root"

ps: 自己把\ -c的"\"去掉，且不要斷行。

無需叫出jupyterlab:
docker run -it -p 82:8888 6de62981d855 /bin/bash
```

8. 若是開啟jupyterlab的模式，則打開網頁輸入localhost:82，並且在剛剛的終端機中找token，將token貼到Password or token的欄位就可以進入jupyterlab開發了。

進入localhost:82後:



token位置:


To access the server, open this file in a browser:
file:///root/.local/share/jupyter/runtime/jpserver-1-open.html
Or copy and paste one of these URLs:
http://09d30246030c:8888/lab?token=7e3f6e5f479aca4483e6935e744adf613fa14bd6571da599
or http://127.0.0.1:8888/lab?token=7e3f6e5f479aca4483e6935e744adf613fa14bd6571da599

資料備份與共享

1. 分別在終端機下，以下指令，如此兩個container(1和2)在/usr/src/my_workspace/volume位置下的資料就會被保存和共用，不會因為container被刪除而資料消失。

   ```
   docker run --rm -it -v C:\Users\使用者名稱\Desktop\volume_test:/usr/src/my_workspace/volume \
   --name container1 6de62981d855 /bin/bash

   ps:以上的\ --name container1 6de62981d855 /bin/bash自己把"\"去掉，寫成一行，不要斷行。

   docker run --rm -it --volumes-from container1 --name container2   6de62981d855 /bin/bash
   ```

2. 由於/usr/src/myworkspace/volume是絕對路徑，要事先知道container的資料夾結構，但基本上都會是 usr/src/...

Reference:

1. https://jchu.cc/2016/04/19-docker.html
2. https://blog.gtwang.org/linux/docker-commands-and-container-management-tutorial/
3. https://hackmd.io/@bluewings1211/SJkLOW9_l?type=view#What-is-a-container