

Docker - 了解 Docker build 指令

Docker image 是如何架設起來的？

Docker 採用 union 檔案系統來儲存image

union檔案系統的特色如下：

1. 分層 (layer) 式的檔案系統
2. 支援把 [檔案系統的修改] 作為 commit，讓多個 commit 一層層堆疊
3. 支援把多個目錄掛載到同一個虛擬檔案系統下

什麼是UnionFS？

聯合檔案系統（Union File System），它可以把多個目錄(也叫分支)內容聯合掛載到同一個目錄下，而目錄的物理位置是分開的。UnionFS允許只讀和可讀寫目錄並存，就是說可同時刪除和增加內容。

docker的映象rootfs，和layer的設計

任何程式執行時都會有依賴，無論是開發語言層的依賴庫，還是各種系統lib、作業系統等，不同的系統上這些庫可能是不一樣的，或者有缺失的。

為了讓容器執行時一致，docker將依賴的作業系統、各種lib依賴整合打包在一起（即映象），然後容器啟動時，作為它的根目錄（根檔案系統rootfs），使得容器程序的各種依賴呼叫都在這個根目錄裡，這樣就做到了環境的一致性。

Docker 是如何應用這些特性呢？

image裡的檔案都是不可以修改的，但將image執行的container是可以修改檔案的，透過修改後的container加上commit指令可以將container修改+image合併成新的image

```
#Terminal 1
docker run --rm -it --name some busybox

#進入建立檔案
```

```

echo new commit > file
cat file

#Terminal 2, commit檔案系統成為新的image
#some為container的名稱, myimage為commite完後image的名稱
docker commit some myimage

#執行新的image
docker run --rm -it --name new myimage

#檢查剛剛建立的檔案是否存在
cat file

#docker commit => 可以把container上對檔案系統的修改commit成新的image

```

docker pull image時，每一行digest都是commit，每行都可以當image使用

```

docker pull mysql

Using default tag: latest
latest: Pulling from library/mysql
a10c77af2613: Pull complete
b76a7eb51ffd: Pull complete
258223f927e4: Pull complete
2d2c75386df9: Pull complete
63e92e4046c9: Pull complete
f5845c731544: Pull complete
bd0401123a9b: Pull complete
3ef07ec35f1a: Pull complete
c93a31315089: Pull complete
3349ed800d44: Pull complete
6d01857ca4c1: Pull complete
4cc13890eda8: Pull complete
Digest: sha256:aeecae58035f3868bf4f00e5fc623630d8b438db9d05f4d8c6538deb14d4c31b
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest

docker run --rm -it a10c77af2613

```

實際打包一個有vim的image，完成之後就可以編輯Container裡面的設定或程式碼了

```

# Terminal 1
docker run --rm -it --name some alpine

# 進去確認並安裝vim
apk add --no-cache vim

```

```
# Terminal 2
docker commit some vim_image

# 使用新的vim_image 執行container
docker run --rm -it --name new vim_image

# 在container內執行vim
vim
```

Dockerfile檔案可以讓客製化image變成"自動化"，可以藉由Dockerfile檔來描述與執行自動化建置

Docker 的原理很單純，裡面描述多個RUN跟COPY等指令。執行完成後，docker會立刻commit，在執行下一個指令docker基於前一個commit再創造一個container，直到全部都執行完畢，代表image建置完成

dockerfile就是不斷執行run跟commit！可以達到環境即程式的目標

建立一個新檔案dockerfile

```
FROM alpine
RUN apk add --no-cache vim
```

docker build指令，dockerfile跟當前目錄要相同，之後執行dockerfile

```
# Build 一個新image，完成後tag為vim
docker build -t vim .

# 使用新的image執行container
docker run --rm -it --name new vim
```