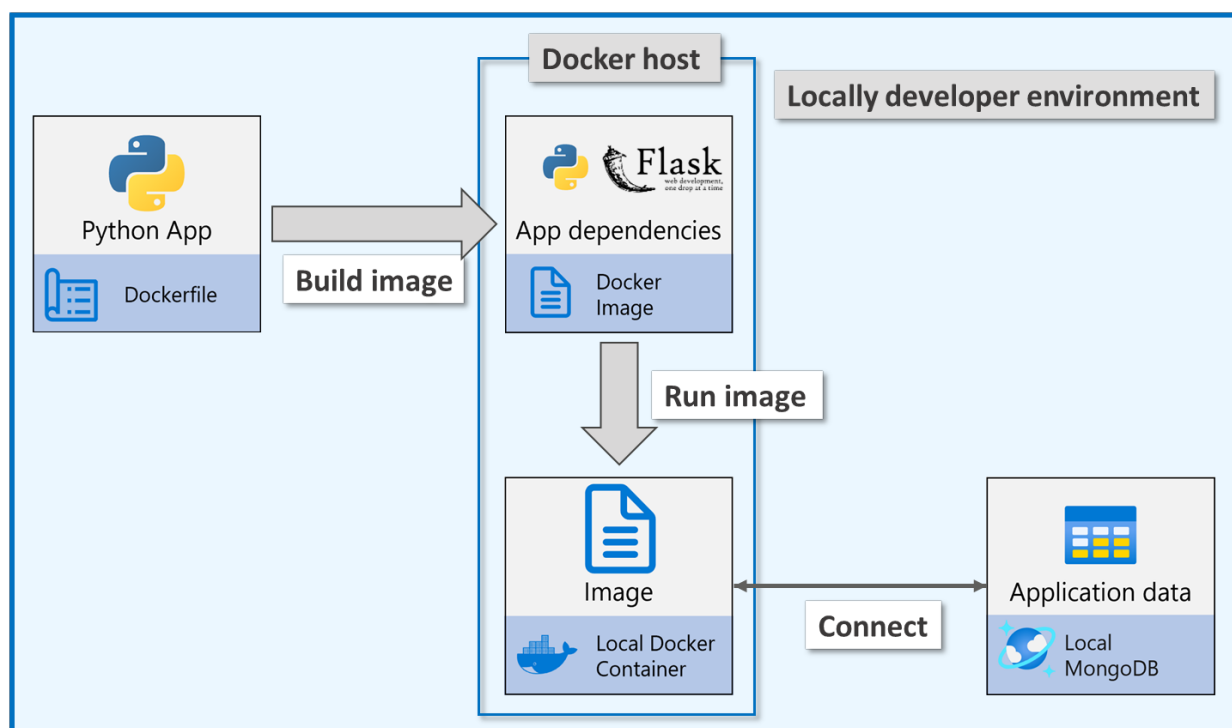


# Docker環境建置過程

我們的網站是由Docker來架設，

首先先編寫一個Dockerfile。



Dockerfile是一種文本文件，其中包含了一個image的所有指令，從而讓Docker引擎可以自動構建一個包含所需應程序的映像。也可以說它是一個使用container的說明書，依照環境與使用者需求編寫指令。

Image與container 完成後，這一個完整的docker host就可以與其中的flask程式做溝通，以達成在本地端的瀏覽器上呈現我們的網站。

```

frankwu@MSI:/mnt/c/Users/Frank/website_v6$ cd /mnt/c/Users/Frank/website_v7
frankwu@MSI:/mnt/c/Users/Frank/website_v7$ docker build -t bdse29image2 -f ./Dockerfile.txt .
[+] Building 88.0s (14/14) FINISHED
=> [internal] load build definition from Dockerfile.txt 0.1s
=> => transferring dockerfile: 42B 0.1s
=> [internal] load .dockerignore 0.1s
=> => transferring context: 2B 0.1s
=> [internal] load metadata for docker.io/library/python:3.9-slim-buster 1.9s
=> [auth] library/python:pull token for registry-1.docker.io 0.0s
=> [1/8] FROM docker.io/library/python:3.9-slim-buster@sha256:1c5091a9ba3001c8e8182f56a740f26e5b328b7aa3f2c1f7d9 0.0s
=> [internal] load build context 0.4s
=> => transferring context: 70.33kB 0.4s
=> CACHED [2/8] WORKDIR /app 0.0s
=> CACHED [3/8] COPY app.py . 0.0s
=> [4/8] COPY templates/ templates/ 0.0s
=> [5/8] COPY static/ static/ 0.1s
=> [6/8] COPY model/ model/ 5.6s
=> [7/8] COPY data/ data/ 0.0s
=> [8/8] RUN pip install --upgrade pip && pip install flask && pip install pandas && pip install jo 73.3s
=> exporting to image 6.3s
=> => exporting layers 6.3s
=> => writing image sha256:055b11b821b4909143ffb3b9e1bd5be7842bc7ddb5b6985ac002714eca6b9874 0.0s
=> => naming to docker.io/library/bdse29image2 0.0s
frankwu@MSI:/mnt/c/Users/Frank/website_v7$ docker container run -dit --name bdse29web2 -p 8082:5000 --network bdse29netw
ork bdse29image2
1309f5e4ad4a84879baad09d58ff902f4265a39c258990a828f1d00f951b1648
frankwu@MSI:/mnt/c/Users/Frank/website_v7$ |

```

首先建立一個資料夾，用以放置Dockerfile文件、python flask程式與模型資料。

開啟命令提示字元，輸入WSL進入Linux虛擬環境。WSL (Windows Subsystem for Linux) 它是一個適用於Linux的Windows子系統

輸入`cd /mnt/c/Users/XXX/website_v7`，進入Dockerfile所在的目錄中

輸入`docker image build -t bdse29image2 -f ./Dockerfile.txt .`

這串指令是透過Dockerfile來建置映像檔。docker引擎會按照dockerfile步驟自動建置映像。

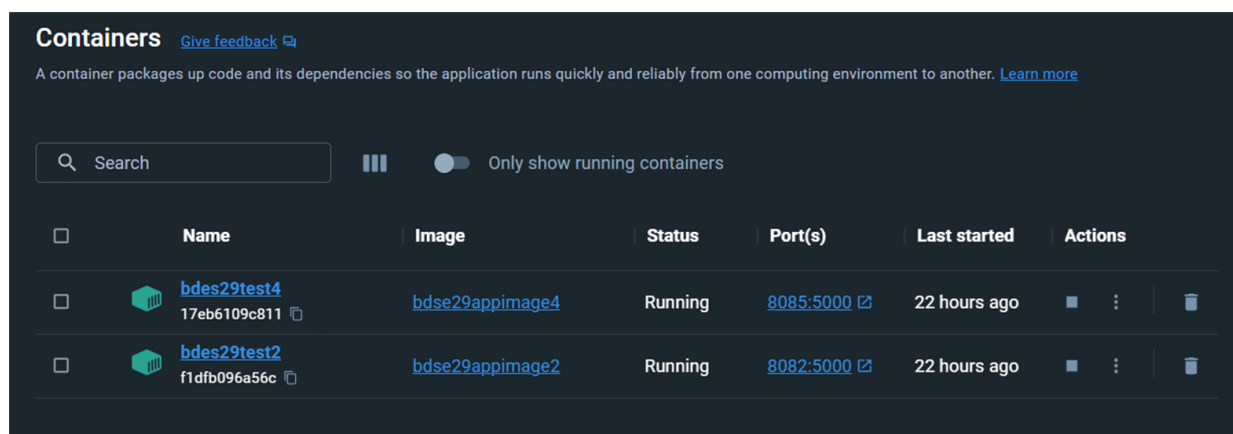
輸入`docker container run -dit --name bdse29test4 -p 8082:5000 bdse29image2`

-d：參數表示啟動一個後台執行的容器，也就是說容器會在背景執行而不會顯示 log 訊息在終端機上。

-i：參數表示開啟標準輸入（stdin），讓使用者可以跟容器互動。

-t：參數表示開啟一個偽終端（pseudo-TTY），這樣才能進行輸入輸出的互動操作。

-p 8082:5000：參數表示將主機的 8082 port 對應到容器的 5000 port，而我們的flask設定port=5000，讓瀏覽器可以透過 <http://localhost:8082> 訪問我們的網站。



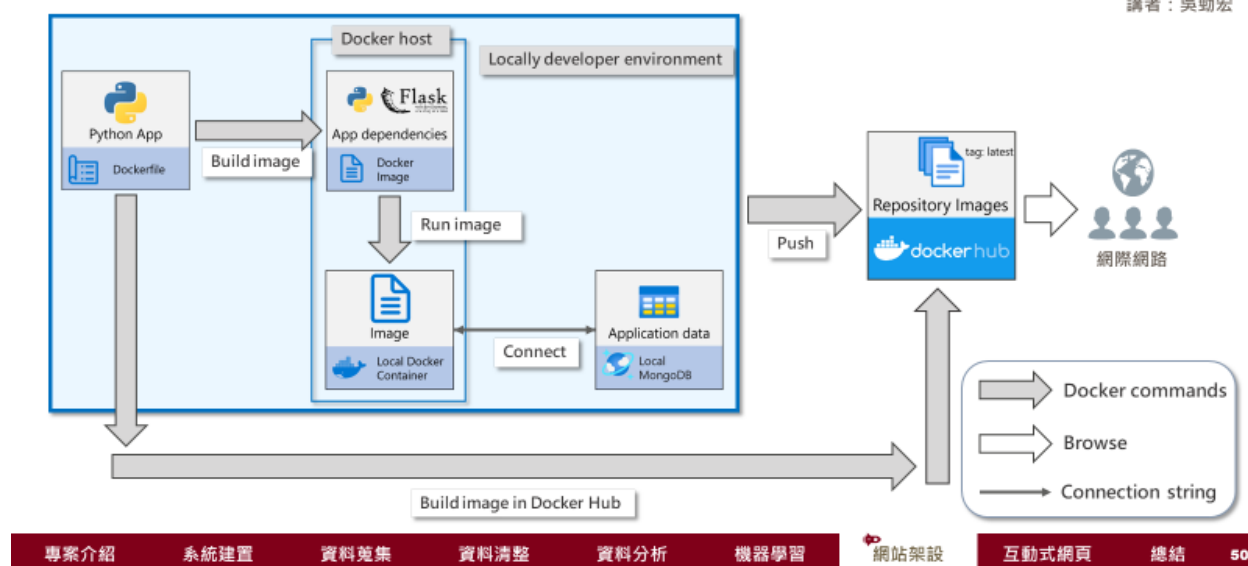
Docker Desktop是運行在Windows和Mac上的一個應用程式，用於管理Docker容器的平台。它允許用戶快速地在自己的本地機器上設置Docker環境，並且可以管理所有Docker資源，包括Docker容器、映像檔和網路等。

Docker Desktop提供了一個簡單、方便的方式來管理Docker容器和映像檔，使得開發人員和系統管理員能夠更輕鬆地使用Docker來開發、測試和部署應用程式。

## Docker Hub



講者：吳勁宏



網際網路的部分，一開始我們想透過上傳到docker hub的式來實現。

Docker hub是docker 官方提供的雲端平台，可以存放的映像。並且透過租用網域達到呈現網站的效果。

commit 用來將容器轉換為映像檔： `docker commit [container-name] [your-dockerhub-username]/[repository-name]:[tag]` `docker commit bdes29test4 frankwu2023/bdse29appimage4:latest`

push用來上傳映像檔： `docker push [your-dockerhub-username]/[repository-name]:[tag]`  
`docker push frankwu2023/bdse29appimage4:latest`

```
ngrok

Announcing ngrok-rs: The ngrok agent as a Rust crate: https://ngrok.com/rust

Session Status      online
Account             FrankWu (Plan: Free)
Update              update available (version 3.2.2, Ctrl-U to update)
Version             3.2.2
Region              Japan (jp)
Latency             48ms
Web Interface       http://0.0.0.0:4040
Forwarding           https://37af-125-227-255-79.ngrok-free.app -> http://bdse29web2:5000

Connections          ttl      opn      rt1      rt5      p50      p90
                    5        0        0.01     0.01     0.02     0.03

HTTP Requests
-----
GET /static/startbootstrap-modern-business-gh-pages/assets/favicon-32x32.png 200 OK
GET /static/startbootstrap-modern-business-gh-pages/js/scripts.js             200 OK
GET /static/startbootstrap-modern-business-gh-pages/css/styles.css             200 OK
GET /static/img/USA_flight_delay_600x400.png                                  200 OK
GET /                                                                           200 OK
```

我們同時也使用ngrok來架設我們的網站， ngrok是一個免費的開放原始碼的工具，它的原理是建立一個反向代理，將本地伺服器連接給公共網路，並將對這個公共網路的請求轉發到本地伺服器上。指定本地伺服器端口和ngrok的端口。在運行命令之後，ngrok會生成一個公共網址，用於存取本地伺服器的應用程式。在瀏覽器中打開這個網址，就可以預覽和測試本地的應用程式。

隨後我透過官方提供的docker指令將ngrok製作成容器並放入docker虛擬網路中，透過設定好的端口將ngrok容器(bdse29ngrok)與我們網頁所在的容器(bdse29web2)連接。