

Docker - 運用 docker compose 組合 container

docker compose 為使用多個container 成為一個完整服務工具

單一container

node.js+npm為例

```
docker run -it --rm -v $PWD:/source -w /source node:14-alpine
```

使用yml寫下docker-compose.yml

檔名必須是 docker-compose.yml, docker-compose.yaml, compose.yml, compose.yaml 其中一個

```
#使用第幾版的設定檔
version: "3.8"

#定義 Service 的區塊，一個 service 設定可以用來啟動多個container
services:
  # 定義一個叫 npm 的 service
  npm:
    image: node:14-alpine
    stdin_open: true
    tty: true
    working_dir: /source
    volumes:
      - ../source
#image => IMAGE參數
#stdin_open => --interactive
#tty => --tty
#working_dir => --workdir
#volumes => --volume 為 使用volumn同步程式
```

定義好後，docker-compose run

```
docker-compose run --rm npm npm -v
```

多環境測試

```
#docker-compose.yml
version: "3.8"

services:
  php71: &basic
    image: php:7.1-alpine
    stdin_open: true
    tty: true
    working_dir: /source
    volumes:
      - ../source
    command: vendor/bin/phpunit
  php72:
    <<: *basic
    image: php:7.2-alpine
  php73:
    <<: *basic
    image: php:7.3-alpine
  php74:
    <<: *basic
    image: php:7.4-alpine
  php80:
    <<: *basic
    image: php:8.0.0beta4-alpine

#執行
#各別執行
docker-coompose run --rm php71
docker-coompose run --rm php72
docker-coompose run --rm php73
docker-coompose run --rm php74
docker-coompose run --rm php80

#一次全部執行
docker-compose up
docker-comose ps #看結果
```

連結多個container

多個container無法同時綁定終端機的輸入在每個container上，通常需要觀察log或背景執行

```
# Terminal 1, 使用 MYSQL_ROOT_PASSWORD 與 MYSQL_DATABASE
docker run --rm -it --name db -e MYSQL_ROOT_PASSWORD=pass -e MYSQL_DATABASE=some perco
na
```

```
# Terminal 2
docker run --rm -it --link db -p 8080:80 phpmyadmin

#可以轉成docekr compose
#docker-compose.yml
version: "3.8"
services:
  database:
    image: percona
    environment:
      MYSQL_ROOT_PASSWORD: pass
      MYSQL_DATABASE: some
  phpmyadmin:
    image: phpmyadmin
    ports:
      - 8080:80
    links:
      - database:db

#enviroment => --env
#ports => -p
#links => --link
```

執行命令

```
#直接前景執行，當中斷的時候，會停止所有container
#docker-compose up

#背景執行
docker-compose up -d

#查看log
docker-compose logs -f
```

連結更多container

加入wordpress服務

```
#直接修改docekr-compose.yml file!
version: "3.8"

services:
  database:
    image: percona
    environment:
```

```
    MYSQL_ROOT_PASSWORD: pass
    MYSQL_DATABASE: some

phpmyadmin:
  image: phpmyadmin
  ports:
    - 8081:80
  links:
    - database:db

wordpress:
  image: wordpress
  environment:
    WORDPRESS_DB_HOST: db
    WORDPRESS_DB_USER: root
    WORDPRESS_DB_PASSWORD: pass
    WORDPRESS_DB_NAME: some
  ports:
    - 8082:80
  links:
    - database:db
```

執行命令

```
#背景執行
docker-compose up -d

#查看 wordpress 的 log
docker-compose logs wordpress

#透過網址連到wordpress服務
http://localhost:8082
```