

①

不用預設 bridge. 要建立一個 bridge

```
# docker network create -d bridge mybr (name)
# ls
```

```
# docker run -it --name di --network mybr busybox /bin/sh
di/# ping dz dz
```

②

```
# docker run -itd --name myweb -p 80:80 httpd:latest
```

```
# curl 127.0.0.1 看到頁面不能訪問
```

```
# docker exec -it myweb /bin/bash
```

安裝

```
# apt update
```

```
# apt install mariadb-server
```

```
# mysql_secure_installation
```

```
# mysql_secure_installation
enter
:
```

```
# mysql -uroot -p123456
```

```
> create database testdb;
```

```
> show databases;
```

```
> use testdb;
```

```
> create table test (name varchar(50) not null, phone varchar(10));
```

```
> insert into test (name, phone) values ("tom", "123");
("amy", "1234");
```

```
> select * from test;
```

```
> exit
```

```
# apt install php ...
```

```
#
```

Docker-compose : 管理、維護多個容器

使用 yml 配置檔, 配置多個容器, 要在配置檔的資料夾裡啟動、關閉

```
# mkdir test-dockercompose
# gedit docker-compose.yml
```

web: → 容器名

image 鏡像

ports:

8080:80

volumes:

本地磁碟資料夾 映射到網頁伺服器裡面的容目錄裡面

```
# docker-compose up -d
          啟動 背景執行
```

看容器

```
# docker-compose ps
```

服務關閉

```
# docker-compose down
```

Test-1

```
# mkdir composetest
# cd composetest
# gedit app.py
```

```
# gedit requirements.txt
          flask 網頁框架
```

```
# gedit Dockerfile
          基礎容器
```

pip ... 安裝 這裡的套件

```
# gedit docker-compose.yml
```

2個容器 { web: build: — 去執行 Dockerfile
 redis 資料庫, 記憶像是儲存資料

```
# docker-compose up
# docker ps

firefox search 127.0.0.1:5000
```

Test 2 mysql + apache - php

```
# mkdir compose-test-2
# cp -r mydbdate compose-test-2 持久化. 把資料保存在本地端
# cd compose-test-2
# gedit index.php

index.php { $servername = "hello-mysql"
          連接資料庫, 把裡面的東西顯示出來
```

```
# gedit docker-compose.yml

apache-php 網頁伺服器
depends_on:  想要啟動容器, 一定要先啟動指定容器
mysql      資料庫
container_name: "hello-mysql"

networks:  用內建網路, 通訊只能用ip
           用配好的bridge 就可以使用名字
```

```
# docker-compose up

firefox search 127.0.0.1:8000/index.php
```

Test 3: 可用名字 ping 到對方

讓一個容器持續運作, 可以讓他保持睡眠