如果是使用原本的bridge網路,docker間進行通訊是不能用容器名稱進行通訊

如果docker處在不同的bridge網路,這些docker是不能進行通訊的

如果docker是建立在自己創立的bridge網路環境下,docker之間是可以直接用docker名稱進行通訊

docker inspect 容器ID : 顯示容器詳細資訊

進入容器 輸入ifconfig

若是沒有則sudo apt install net-tools

輸入ip route show 或是 route -n 查看路由器

若是無法ping 則apt install iputils-ping

docker network create - -driver bridge 橋接器名稱 : 建立橋接網路

docker network ls : 查看網路

用自己所建立的bridge網路所創造出的容器,可以用容器名稱進行通訊

若是用預設的bridge網路,只能用ip進行通訊

docker network rm : 刪除網路

如果使用host網路,則docker看到的網路空間會跟原本HOST主機是一樣的

缺點:docker 很容易受到外在的威脅

好處是如果這個docker是用作防火牆,入侵偵測,就可以用來保護HOST主機安 全

利用自己的橋接網路建立mysql

docker run -d --network mybridge --name hello-mysql -e MYSQL\_ROOT\_PASSWORD=123456 -p 3306:3306 mysql

進入mysql

先進入容器

mysql -u root -p

mysql

查看資料庫

show databases;

創建testdb資料庫

create database testdb;

切換到testdb資料庫

use testdb;

建立表格

create table addrbook(nqme varchar(50) not null, phone varchar(10));

插入資料

insert into addrbook(name, phone) values ("tom", "123456");

檢視表格

select \* from addrbook;

實際上不需要打開3306 因為客戶端可以透過80進入容器進行內部溝通且需要進行持久化的動作

mkdir mydbdata

docker run -d --network mybridge --name hello-mysql -e MYSQL\_ROOT\_PASSWORD=123456 -v /root/mydbdata:/var/lib/mysql mysql

確認本機80port是否有被占用

netstat -tunlp | grep 80

利用自己的橋接網路建立myphp

docker run -d -p 80:80 --network mybridge -v /root/myphp:/var/www/html --name hello-php radys/php-apache:7.4

注意是否有鏡像標籤latest沒有要去找對應標籤並加在後面