

# Win10 Docker 環境建置

拉回 Docker image, container mysql 設定

# win10 Docker 環境 前置作業

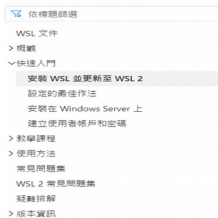
## 0. 跟部門主管申請公司個人 gmail 連線 gcr docker im

## 1. 安裝 docker desktop

<https://docs.docker.com/desktop/windows/install/>

## 2.” 步驟 4 - 下載 Linux 核心更新套件” 安裝

<https://docs.microsoft.com/zh-tw/windows/wsl/install-win10>



## 步驟 4 - 下載 Linux 核心更新套件

1. 下載最新套件：

- WSL2 Linux 核心更新套件 (適用於 x64 電腦) 的

① 注意

如果您使用的是 ARM64 電腦，請改為下載 **ARM64 套件**。如果您不確定您有何種類型的電腦，請開啟命令提示字元或 PowerShell，然後輸入：`systeminfo | find "System Type"`。警告：在非英文的 Windows 版本中，您可能必須修改搜尋文字，例如，在德文中，`systeminfo` 會是 `systeminfo | find "Systemtyp`。

2. 請執行在上一個步驟中下載的更新套件。(按兩下以執行 - 系統會提示您提供更高的權限, 請選取 [是] 以核准此項操作。)

一旦完成安裝，就會移至下一個步驟。在安裝新的 Linux 發行版本時，將 WSL 2 設定為預設版本。(如果您想將現有的 Linux 安裝設定為 WSL 1，請略過此步驟)。

① 注意

如欲詳細資訊，請參閱 Windows 命令列部落格上的關於更新 WSL2 Linux 核心的變更一文。

此夏面有所助益嗎？

 Yes  No

## 本文内容

Windows 測試人員的難

### 手動安裝步驟

### 步驟 1 - 啟用 Windows

也照 2. 檢查結果 MAC

的需求

### 步驟 3 - 啟用虛擬機器功能

#### 步驟 4 - 下載 Linux 核心

| 更新软件

步驟 5 - 將 WSL 2 設定為預設版本

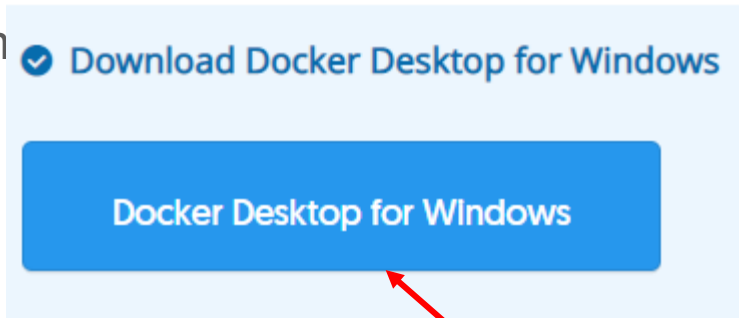
### 步驟 6 - 安裝您選擇的

CHICK 發行版本  
 國際 Macintosh 校對機

用)

將發行版本設定為 WSL  
或 WSL 2

### 空想難難排解



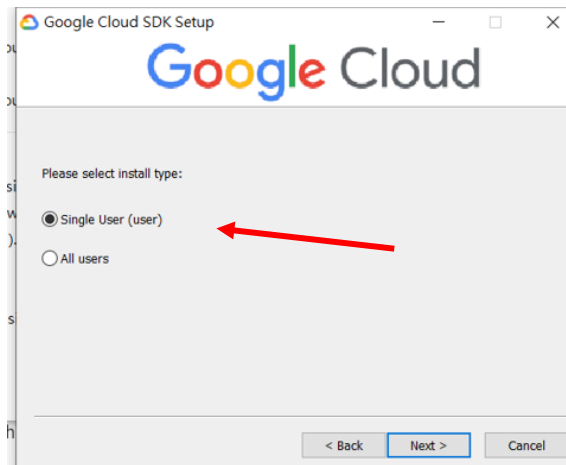
### 3. 安裝 google cloud SDK

<https://cloud.google.com/sdk/docs/install>

<https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe>



安裝過程中 / 點擊 Single User 下載安裝會跳到 CMD 視窗 / 會再跳到瀏覽器 點擊個人公司 gmail 帳號 . 點擊右下角 ' 允許 ' / 會再次跳回 CMD 視窗 / 輸入 1/ region 輸入 34

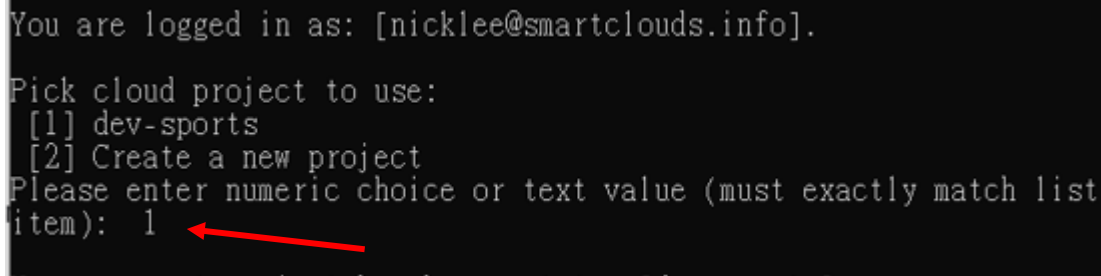


點擊右下角 '允許'

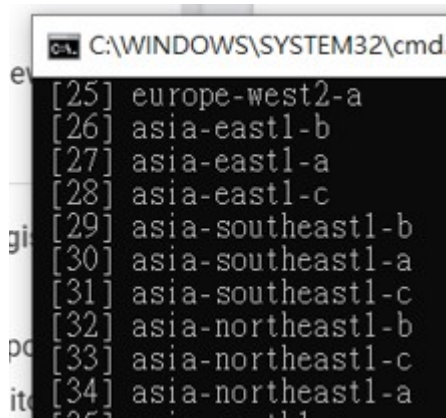


點擊 Single User

輸入 1



region 輸入 34



## 4 CMD 視窗 登入 gcloud 權限

CMD 視窗 輸入 **gcloud auth configure-docker /**

EX:

C:\Users\user\project>**gcloud auth configure-docker**

## 5. 將後端 php Docker image 拉回本機端

# 要拉回後端的 Docker image 要先開啟 VPN 連線 //

# 請 php 後端 傳送檔案 yml+.env 兩個檔案

ps: 如果 **QA 組本地端已有** yml+.env 兩個檔案則更新

yml 檔 image: **gcr.... 連結**即可

存檔在 project 目錄 //.env 檔 port 號改成 **3307**

MYSQL\_EXPOSE\_PORT=**3307**

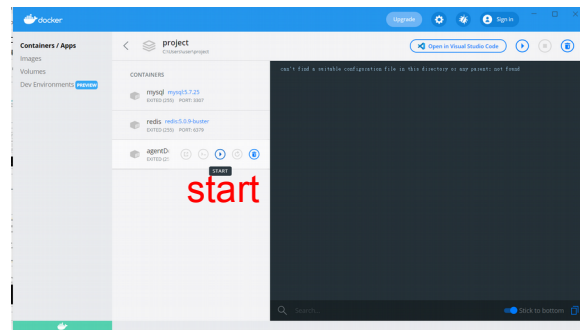
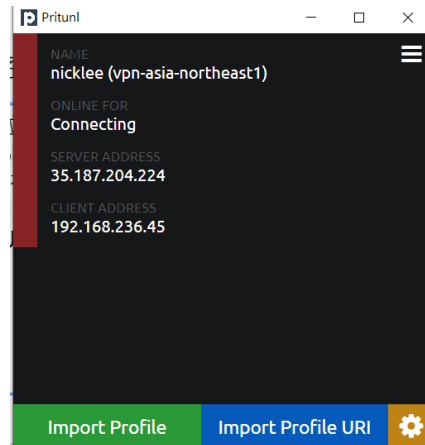
# 拉回 Docker image CMD 指令

C:\Users\user\project>**docker-compose up -d**

# 本機進入 Docker image bash 端 CMD 指令

C:\Users\user\project>**docker exec -it agentDemo bash**

// 若無法進入容器 , 請先用 **docker desktop** 啟動容器



## 6. 本機容器 container 資料庫連線設定

# 檢視 PHP 專案指令 // 輸入 php bin/hyperf.php

bash-5.0#**php bin/hyperf.php**

# 編輯 php 後端連線容器裡 mysql database 設定檔指令 // 輸入 vi .env

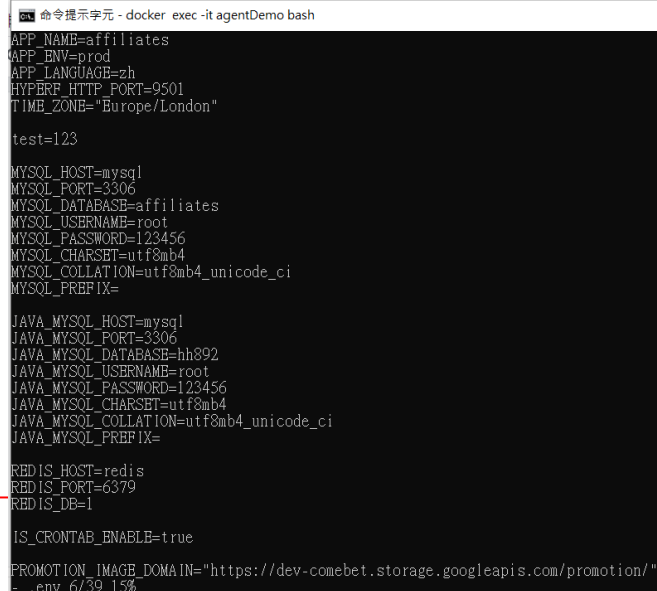
bash-5.0#**vi .env**

# 編輯 php 檔連線資料庫設定

phpDB //affiliates , javaDB / hh892

# 要先按 **O** 再開始編輯, 游標才不會消失

# 編輯完後, 先按 **Esc** 再按 **:wq** 存檔跳出



```
命令提示字元 - docker exec -it agentDemo bash
APP_NAME=affiliates
APP_ENV=prod
APP_LANGUAGE=zh
HYPERF_HTTP_PORT=9501
TIME_ZONE="Europe/London"

test=123

MYSQL_HOST=mysql
MYSQL_PORT=3306
MYSQL_DATABASE=affiliates
MYSQL_USERNAME=root
MYSQL_PASSWORD=123456
MYSQL_CHARSET=utf8mb4
MYSQL_COLLATION=utf8mb4_unicode_ci
MYSQL_PREFIX=

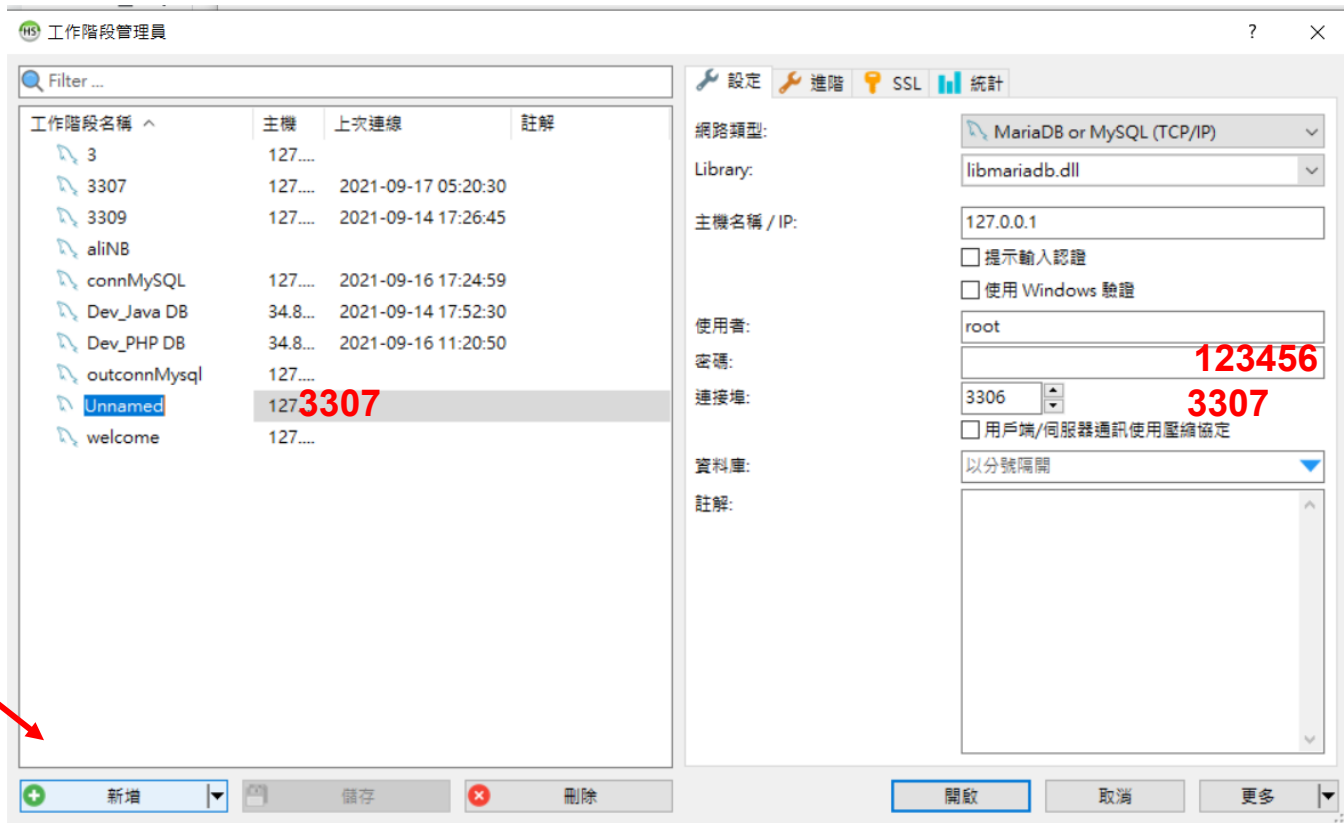
JAVA_MYSQL_HOST=mysql
JAVA_MYSQL_PORT=3306
JAVA_MYSQL_DATABASE=hh892
JAVA_MYSQL_USERNAME=root
JAVA_MYSQL_PASSWORD=123456
JAVA_MYSQL_CHARSET=utf8mb4
JAVA_MYSQL_COLLATION=utf8mb4_unicode_ci
JAVA_MYSQL_PREFIX=

REDIS_HOST=redis
REDIS_PORT=6379
REDIS_DB=1

IS_CRONTAB_ENABLE=true

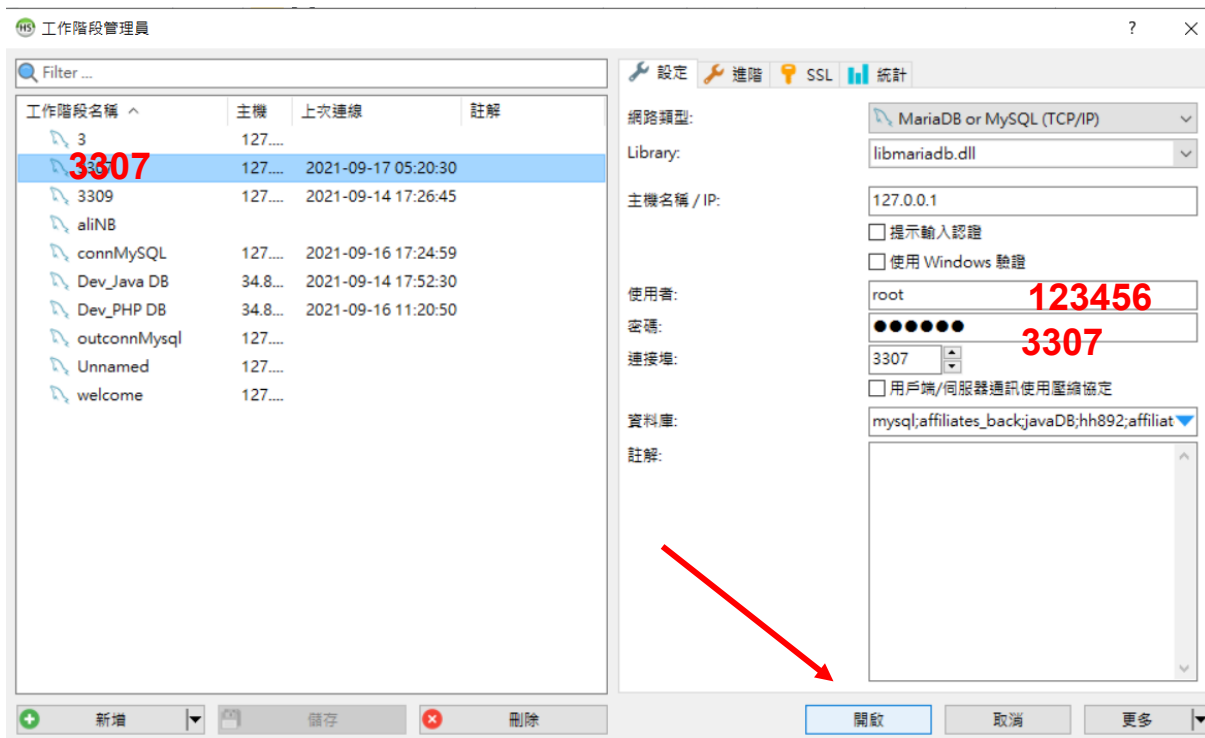
PROMOTION_IMAGE_DOMAIN="https://dev-comebet.storage.googleapis.com/promotion/"
- .env 6/39 15%
```

## 7. 新增本機端 heidis MySQL 連線容器 mysql 連線設定 / 開啟 3307 連線





# 開啟 3307 連線



# 新增資料庫 phpDB //affiliates , javaDB / hh892

3307\ - HeidiSQL 11.3.0.6295

檔案 編輯 搜尋 查詢 工具 前往 說明



資料庫篩選 資料表篩選 主機: 127.0.0.1 查詢

資料庫 (6) 變數 狀態 處理程序 指令統計

資料庫 大小 資料項 最後... 資料表 檢視 函

affiliates 192.0 KiB 9 202... 9 0 0

affiliates\_back

ck1

編輯(K) Alt+Enter

刪除(L)... Shift+Del

清空資料表(M)...

運行常式(N)...

建立新的(O) 資料庫(U)

清除資料分頁篩選(P)

匯出資料為 SQL 腳本(S)

維護(Q)

在伺服器尋找文字(R) Shift+Ctrl+F

批次表格編輯器(T)

全部展開(U)

全部摺疊(V)

物件樹類型選項(W)

列印(X) Ctrl+P

重新整理(Y) F5

Connection properties

中斷連線(Z)

資料表(V)

資料表複本(W)

檢視(X)

Stored procedure

Stored function

觸發器(Y)

事件(Z)

建立資料庫...

名稱(N):

affiliates

排序規則(O):

utf8mb4\_unicode\_ci

伺服器預設: latin1\_swedish\_ci

確定

取消

CREATE 代碼:

CREATE DATABASE `affiliates` /\*!40100 COLL

修訂資料庫...

名稱(N):

hh892

排序規則(O):

utf8mb4\_unicode\_ci

伺服器預設: latin1\_swedish\_ci

確定

取消

CREATE 代碼:

CREATE DATABASE `hh892` /\*!40100 COLLATE `

## 8.php 後端更新 image 後拉回後端

# php 後端有更新 image 時, 要請 php 後端傳送新的 .yml 檔 或傳送 image 連結

[gcr.io/dev-sports/comebet/php-affiliates:0aa810bcd98703a607745da098a7494ec3b41814](https://gcr.io/dev-sports/comebet/php-affiliates:0aa810bcd98703a607745da098a7494ec3b41814)

# 將 gcr 連結 貼到 .yaml 檔 image: [gcr.io/dev-sports/comebet/php-affiliates:0aa810bcd9.....](https://gcr.io/dev-sports/comebet/php-affiliates:0aa810bcd98703a607745da098a7494ec3b41814)

# 拉回 Docker image CMD 指令重新 compose C:\Users\user\project>**docker-compose up**

```
C:\Users\user\project> docker-compose up
1 # syntax version.
2 version: "3.7"
3 services:
4   mysql:
5     image: mysql:5.7.25
6     container_name: mysql
7     ports:
8       - $(MYSQL_EXPOSE_PORT):3306
9     environment:
10      MYSQL_ROOT_PASSWORD: $(MYSQL_PASSWORD)
11     networks:
12       - default
13     volumes:
14       - $(MYSQL_DATA):/var/lib/mysql
15   redis:
16     container_name: redis
17     image: redis:5.0.9-buster
18     ports:
19       - $(REDIS_EXPOSE_PORT):6379
20     stdin_open: true
21   php:
22     networks:
23       - default
24   agentDemo:
25     image: gcr.io/dev-sports/comebet/php-affiliates:0aa810bcd98703a607745da098a7494ec3b41814 #這與php image位置
26     container_name: agentDemo
27     ports:
28       - $(EXPOSE_PORT_AGENT):9501
29     networks:
30       - default
31   networks:
32     default:
33       name: demo
34       driver: bridge
```

## 9 cmd 視窗 bash 跑 php 指令

# 本機進入 Docker image bash 端 CMD 指令

```
C:\Users\user\project>docker exec -it agentDemo bash
```

# 將 php 兩個 sh 檔放到 agentDemo 裡面

```
C:\Users\user\project\agentDemo\
```

# 複製 agentDemo 檔案到容器根目錄


```
bash-5.0# cp /agentDemo/* ./
```


# 本機進入 Docker image bash 端 跑 php 運算指令

```
bash-5.0# ./run_all_stat.sh 2021-09
```

使用者 > user > project > agentDemo

名稱

 run\_all\_stat.sh

 run\_customer\_by\_day.sh