

# 使用pgAdmin操作 Docker上的PostgreSQL

郭益華

# 目錄

- 1. 簡介說明
  - minnow下載安裝步驟
  - Docker下載安裝
- 2. 下載Docker上的PostgreSQL
- 3. 與Docker PostgreSQL互動
  - 方法(一) 從外部進到docker PostgreSQL psql 進行資料庫操作
  - 方法(二) 從外部直接操作docker PostgreSQL
  - 方法(三) 使用 pgAdmin 連線並結合 Makefile 操作資料庫

# 1. 簡介說明

# 實作說明

- 使用Docker所提供的PostgreSQL
- 使用本地端pgAdmin連線Docker上的PostgreSQL
- 使用本地端pgAdmin與Docker上的PostgreSQL互動
- 將相關指令寫入MakeFile

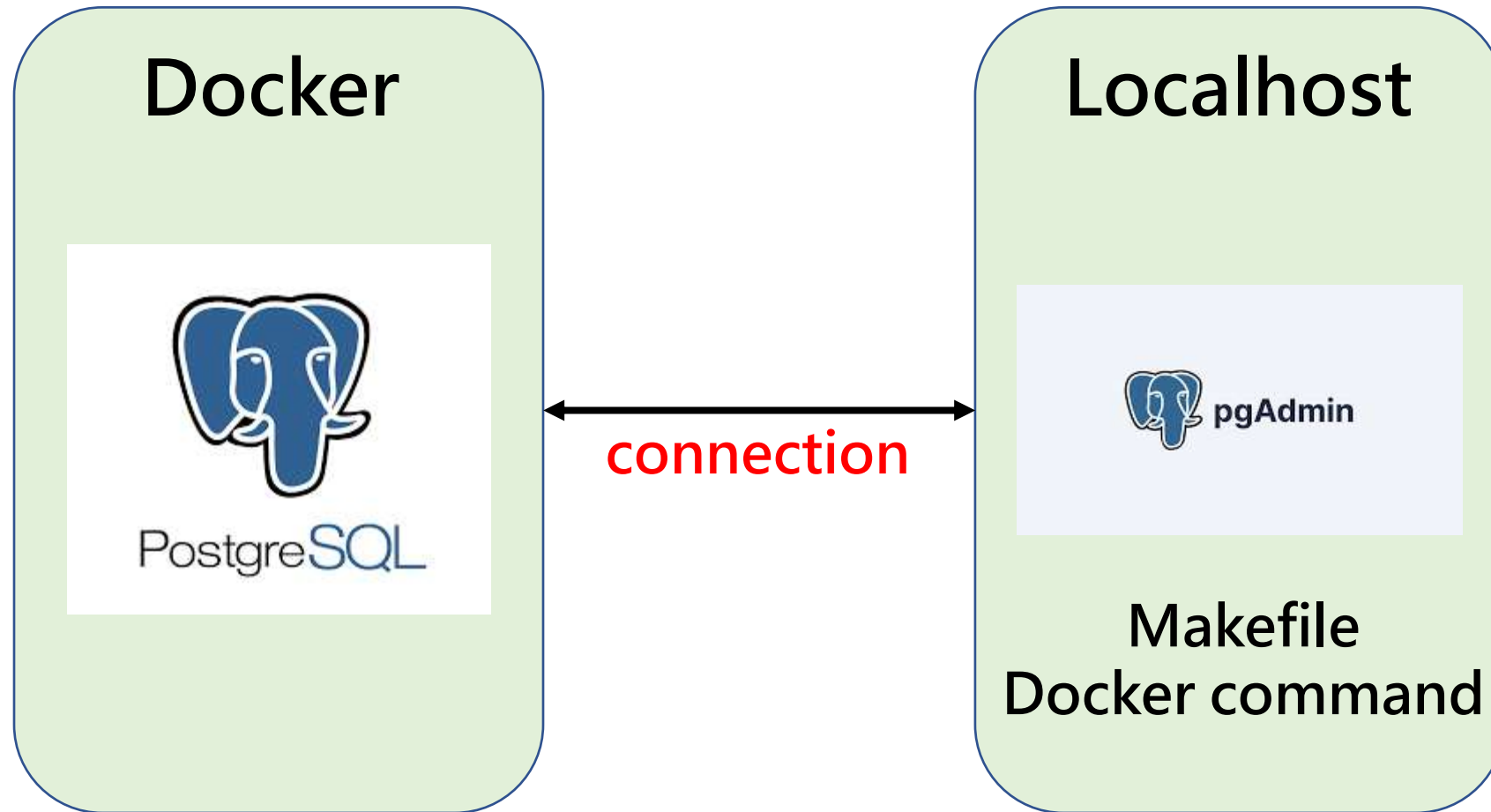
# 前置準備

- pgAdmin 下載
- <https://www.pgadmin.org/download/>
- minGW 下載
- <https://sourceforge.net/projects/mingw/postdownload>
- Docker 下載
- <https://docs.docker.com/desktop/install/windows-install/>

# 實務知識

- Docker 初探
- 使用powershell 連線至 Docker PostgreSQL
- 本地pgAdmin與Docker PostgreSQL連線
- MakeFile使用

# Flow



# pgAdmin介紹

- 為一可介面化操作與管理PostgreSQL的應用程式



# MinGW 介紹

- MinGW為可在**Windows環境**執行 Linux make 指令為主的一個工具程式，並且經由讀取一個叫做 MakeFile 的檔案，自動化建構軟體。
- 簡單說，就是可透過將**重複且複雜的指令**撰寫在MakeFile中，使用make指令去呼叫**重複複雜的指令**，這樣可省去大量編譯的時間


# 會使用到的 Docker 指令

- `docker ps`: 查看運作中的container
- `docker ps -a`: 查看所有container，不論是否運作中
- `docker images`: 查看images
- `docker pull <image_name>`: 下載images
- `docker run`: 執行啟動images
- `docker exec -it`: 操作images
- `docker stop`: 終止關閉運作

# minGW下載安裝步驟



# 點入連結下載即可

Home / Browse Open Source / Software Development / Build Tools / MinGW - Minimalist GNU for Windows

 **MinGW - Minimalist GNU for Windows**  
A native Windows port of the GNU Compiler Collection (GCC)  
Brought to you by: [cstrauss](#), [earnie](#), [gressett](#), [keithmarshall](#)



★★★★★ 174 Reviews      Downloads: 6,104,563 This Week      Last Update: 2021-09-05


  

Windows

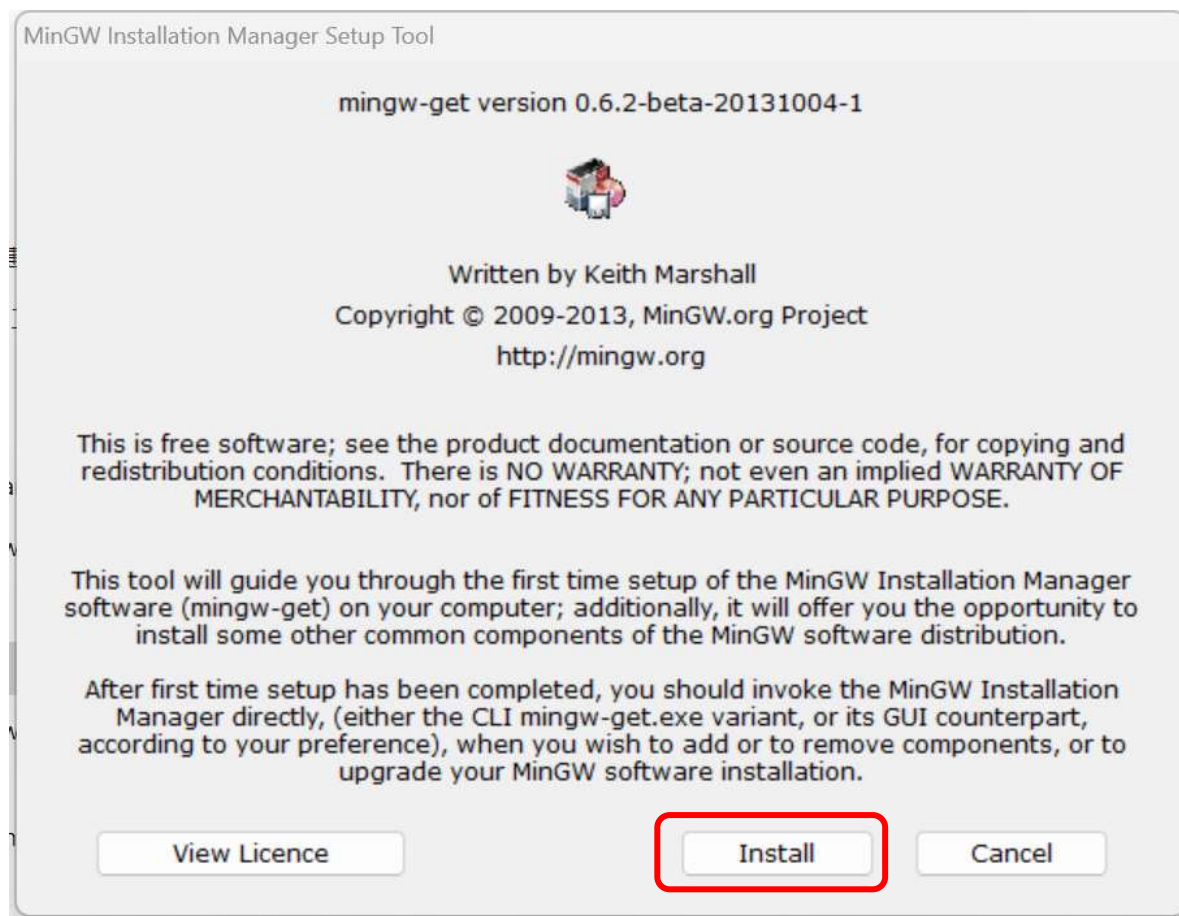
[Summary](#) | [Files](#) | [Reviews](#) | [Support](#) | [News](#) | [Wiki](#) | [Mailing Lists](#) | [Tickets](#) | [Git](#)

下載完成會得到這個檔案



<input checked="" type="checkbox"/>	 mingw-get-setup	2023/9/10 上午 09:00	應用程式	85 KB
-------------------------------------	---	--------------------	------	-------

# 點選 mingw-get-setup 檔案

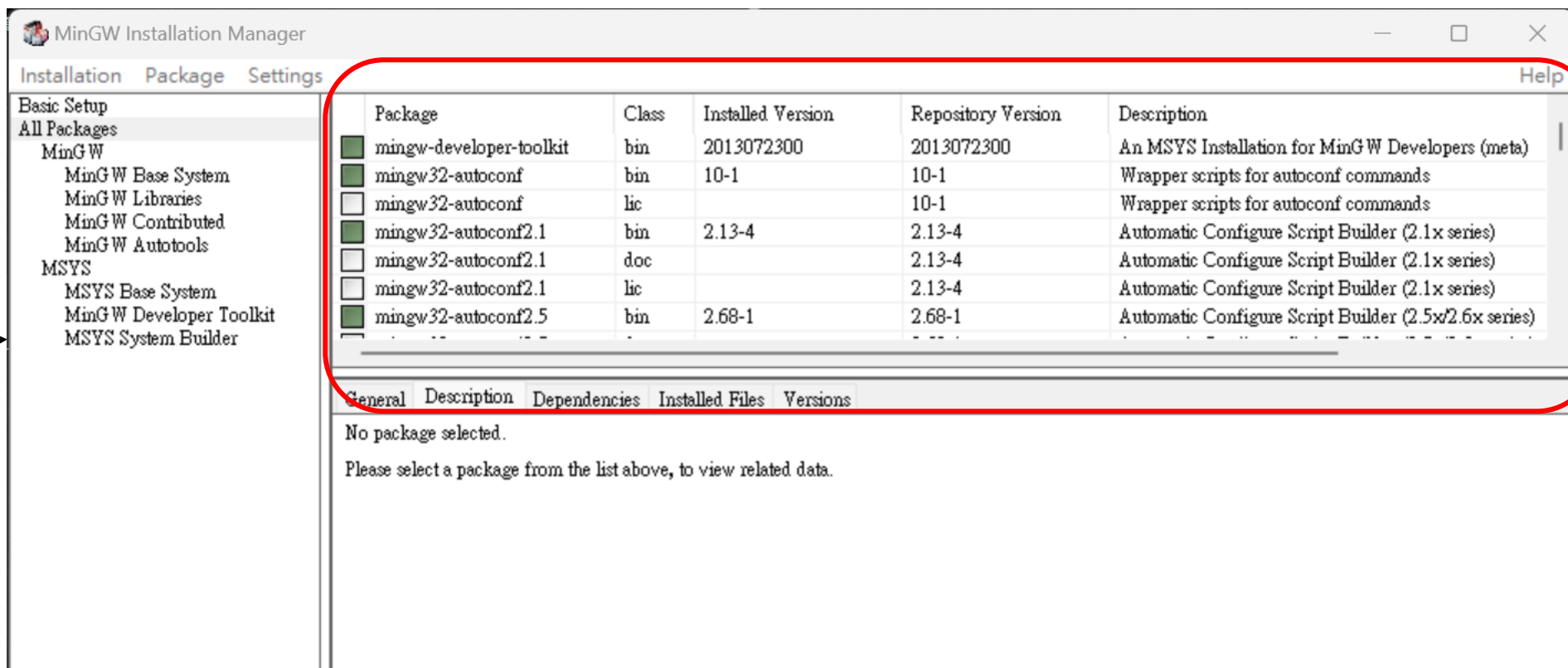


點選Install安裝後會產生此檔案

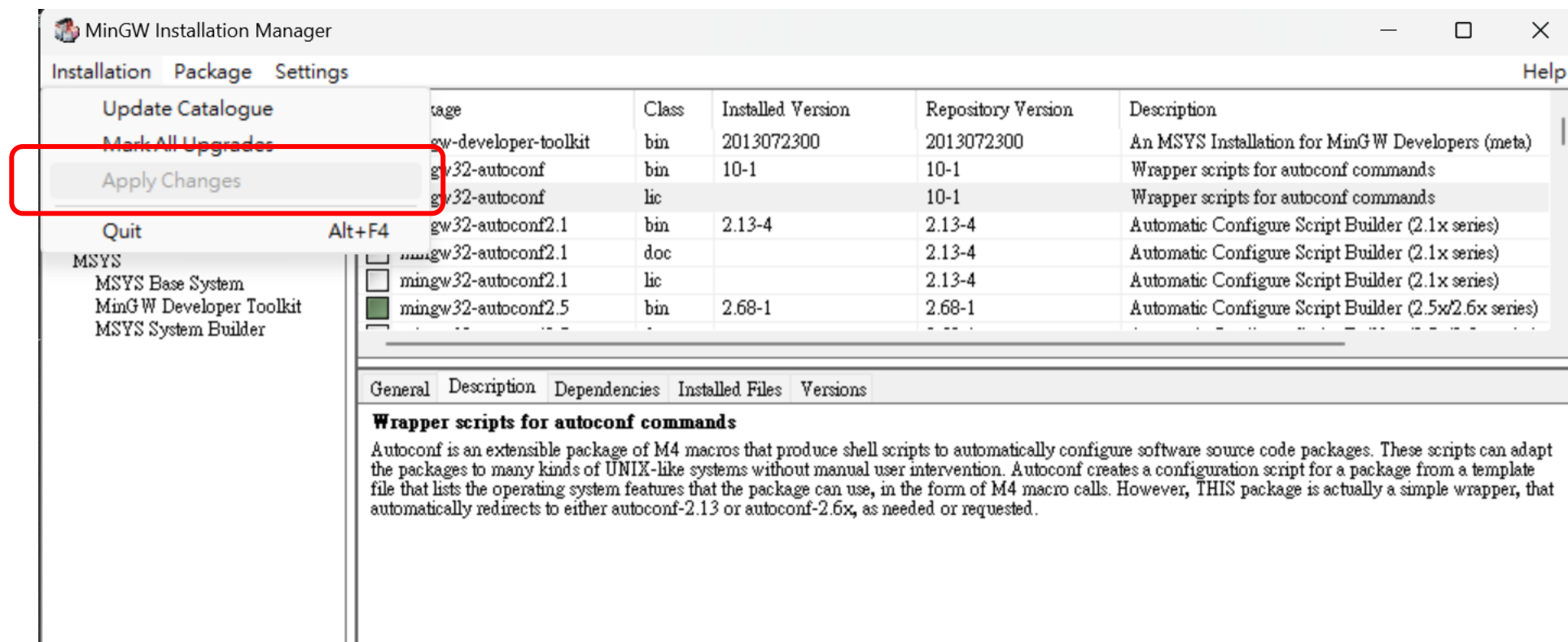


# 點選 MinGW Installer

可全部勾選



# 右側狗選完點選上方Installation中的Apply Changes 即可



# 設定環境變數(1/6)

安裝後會有一個MinGW的資料夾

For Windows安裝檔_EndNote20_WIN	2022/3/26 上午 08:27	檔案資料夾
Intel	2023/9/11 下午 06:29	檔案資料夾
<input checked="" type="checkbox"/> MinGW	2023/9/10 下午 01:21	檔案資料夾
mingw64	2018/3/20 下午 10:02	檔案資料夾
MSI	2023/9/11 下午 06:30	檔案資料夾

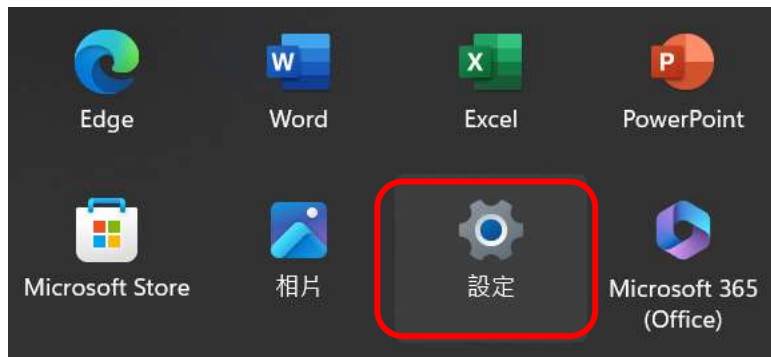
點選進入找到bin資料夾，將此路徑複製

<input type="checkbox"/> 名稱	修改日期	類型	大小
<input checked="" type="checkbox"/> bin	2018/3/20 下午 10:02	檔案資料夾	
etc	2018/3/20 下午 09:47	檔案資料夾	
include	2018/3/20 下午 10:02	檔案資料夾	
lib	2018/3/20 下午 10:02	檔案資料夾	
libexec	2018/3/20 下午 09:19	檔案資料夾	
licenses	2018/3/20 下午 10:02	檔案資料夾	

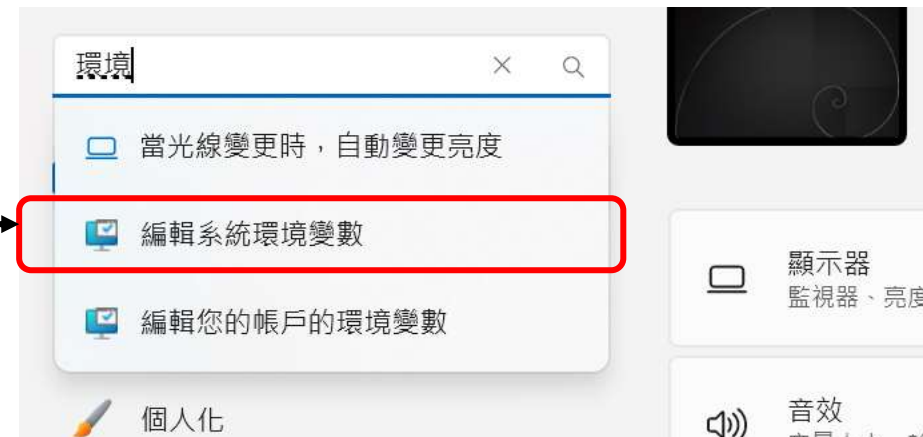


# 設定環境變數(2/6)

移至設定



搜尋 環境變數 點選



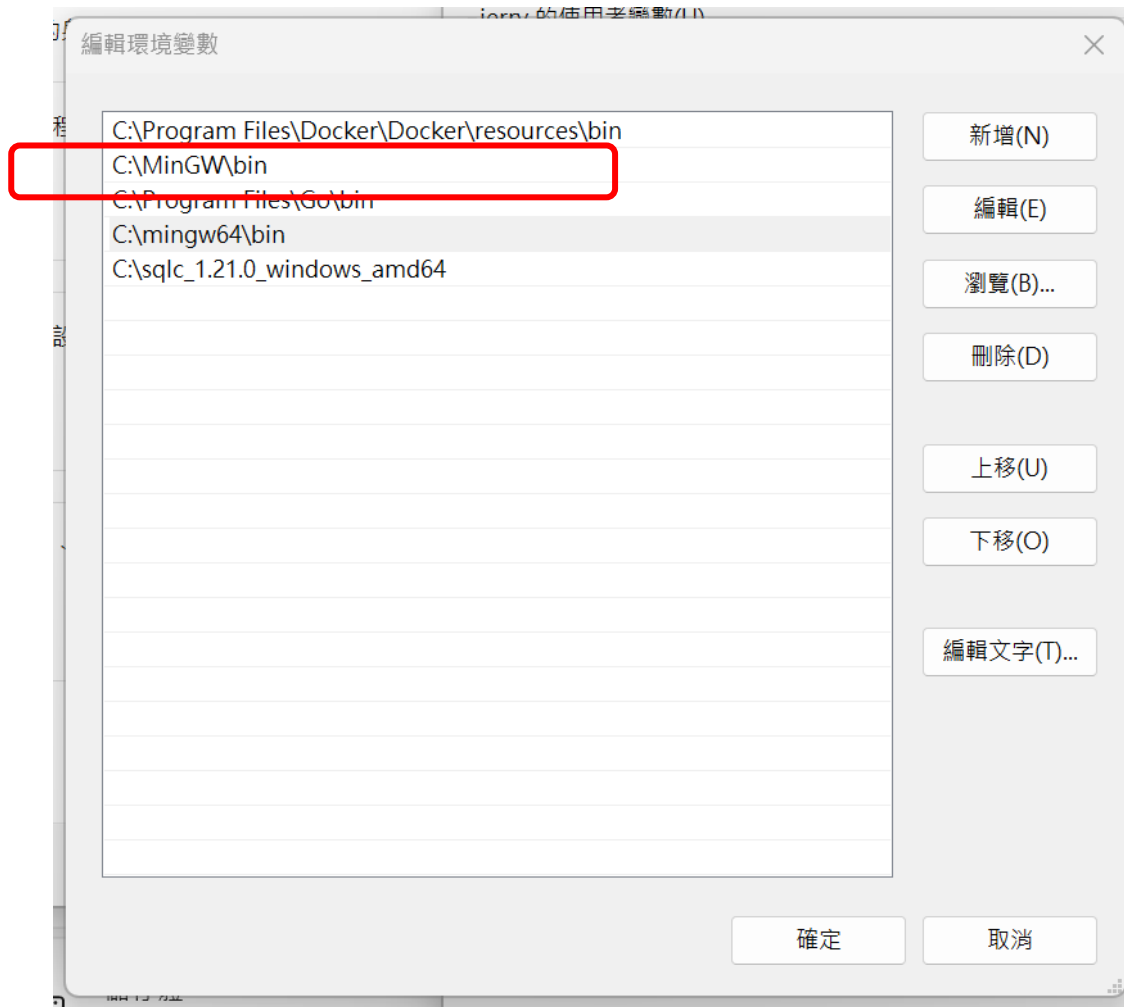
# 設定環境變數(3/6)

找到Path點選編輯



# 設定環境變數(4/6)

將先前複製的路徑加入



# 設定環境變數(5/6)

 libstdc++-6.dll	2018/3/20 下午 09:20	應用程式擴充	1,397 KB
 libwinpthread-1.dll	2018/3/20 下午 09:20	應用程式擴充	51 KB
<input checked="" type="checkbox"/>  mingw32-make	2018/3/20 下午 10:02	應用程式	215 KB
 nm	2018/3/20 下午 08:23	應用程式	1,042 KB
 objcopy	2018/3/20 下午 08:23	應用程式	1,206 KB

將bin資料夾中的mingw32-make  
更改檔名為 make



# 設定環境變數(6/6)

開啟powershell測試  
查看版本: `make -v`

```
PS C:\Users\jerry> make -v
GNU Make 3.82.90
Built for i686-pc-mingw32
Copyright (C) 1988-2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

# Docker下載安裝

# 點選連結進入官網下載即可

[Home](#) / [Manuals](#) / [Docker Desktop](#) / [Install Docker Desktop](#) / [Install on Windows](#)

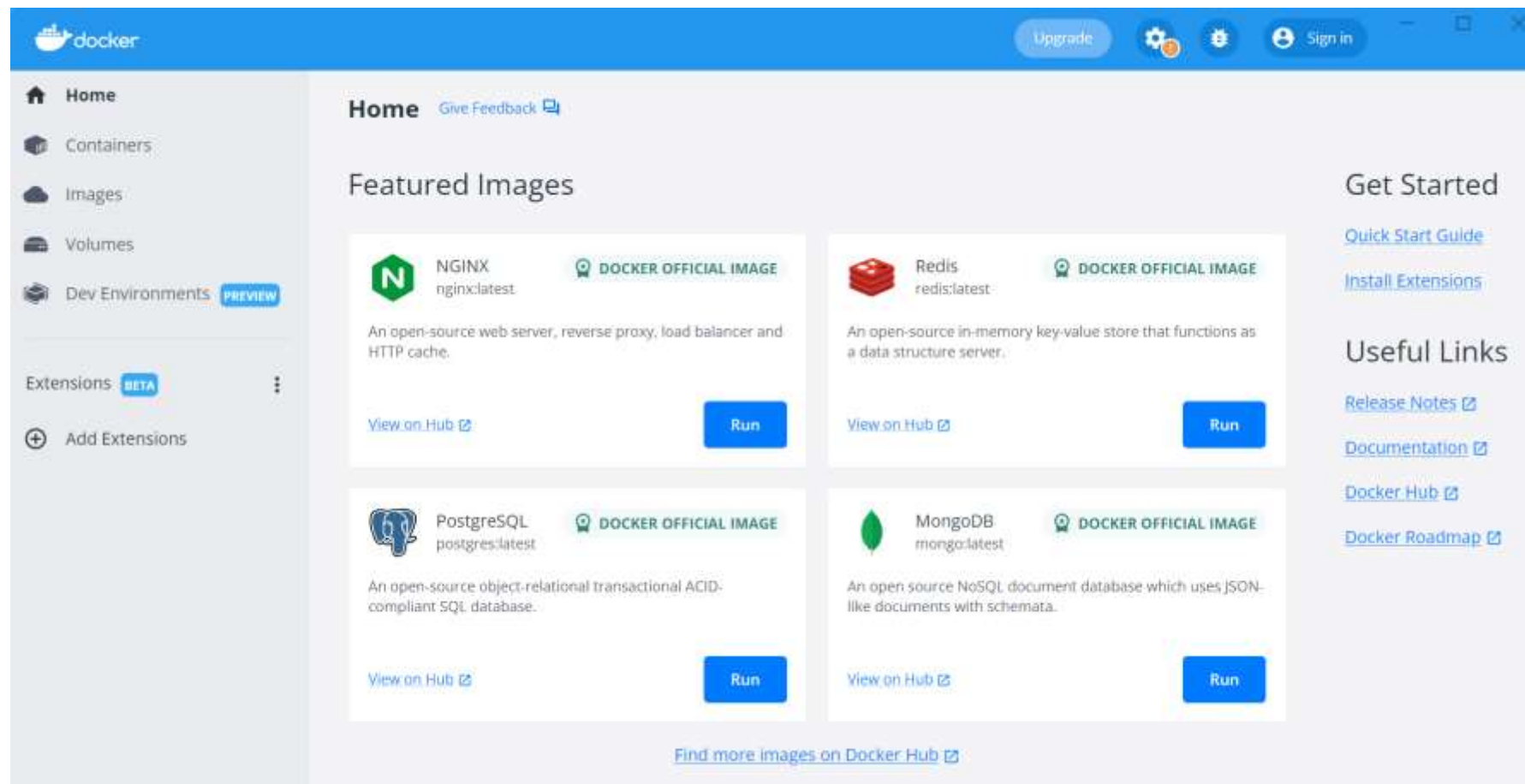
## Install Docker Desktop on Windows

This page contains the download URL, information about system requirements, and instructions on how to install Docker Desktop for Windows.

[Docker Desktop for Windows](#)

For checksums, see [Release notes](#)

# 安裝完成後開啟畫面





## 2. 下載 Docker 上的 PostgreSQL

# 下載Image - postgres:14-alpine

```
PS C:\Users\jerry> docker pull postgres:14-alpine
14-alpine: Pulling from library/postgres
7264a8db6415: Already exists
6ff36a0c8b9b: Already exists
41485c1d4f30: Already exists
a6a18be65c77: Pull complete
6da10a7bbed1: Pull complete
ab7e93c0ebb7: Pull complete
fdfea4fc215a: Pull complete
79def4ab9423: Pull complete
Digest: sha256:8da7e8e94d4bcc28da8f222a4e259b6f7df2dc62640d104f81762678dc1e2200
Status: Downloaded newer image for postgres:14-alpine
docker.io/library/postgres:14-alpine
```

指令: `docker pull <image_name>`  
image\_name: postgres:14-alpine

# 查看當前存在的images

```
PS C:\Users\jerry> docker images
REPOSITORY    TAG                IMAGE ID           CREATED           SIZE
sqlc/sqlc     latest            e5c3b6c57c08      5 days ago      98.2MB
postgres      12-alpine         57609cf5d1df      4 weeks ago     230MB
postgres      14-alpine         955d825eef13      4 weeks ago     235MB
```

指令: docker images

# 執行啟動images

run：標準的Docker建立Container並執行指令

--name postgres14：run指令的文字參數，指定這個Container的名字為postgres14。

-p 5433:5432：run指令的數值參數，把主機的5433 port所有流量轉發到postgres14這個Container的5432 port。

-e：環境參數，可設定使用者及密碼。

-d：run指令的無數值參數，背景執行。

```
PS C:\Users\jerry> docker run --name postgres14 -p 5433:5432 -e POSTGRES_USER=root -e POSTGRES_PASSWORD=1234 -d postgres:14-alpine
a122411c10bffb245efc35f9bdc0d17a4a46fe38fb05237f226fadc5e6c816a7
PS C:\Users\jerry> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	N
AMES						
a122411c10bf	postgres:14-alpine	"docker-entrypoint.s..."	25 seconds ago	Up 24 seconds	0.0.0.0:5433->5432/tcp	p
ostgres14						

docker ps:查看運作中的container

# 3.與Docker PostgreSQL互動

方法(一) 從外部進到docker PostgreSQL psql 進行資料庫操作

方法(二)從外部直接操作docker PostgreSQL

方法(三) 使用 pgAdmin 連線並結合 Makefile 操作資料庫

# 方法(一) 從外部進到docker PostgreSQL psql 進行資料庫操作

```
PS C:\Users\jerry> docker exec -it postgres14 psql -U root
psql (14.9)
Type "help" for help.

root=# select now();
               now
-----
2023-09-12 05:28:20.516228+00
(1 row)

root=# createdb --username=root --owner=root hello_sql 建立名為 hello_sql的資料庫
root=# dropdb hello_sql 刪除 hello_sql的資料庫
root=# exit
Use \q to quit.
root=# \q \q 離開指令
PS C:\Users\jerry>
```

指令:

`docker -it <container name> psql -U <username>`

-U: 指定使用者

Ex: `docker exec -it postgres14 psql -U root`

# 方法(二)從外部直接操作docker postgresQL

```
PS C:\Users\jerry> docker exec -it postgres14 createdb --username=root --owner=root hello_sql
PS C:\Users\jerry> docker exec -it postgres14 dropdb hello_sql
PS C:\Users\jerry>
```

指令:

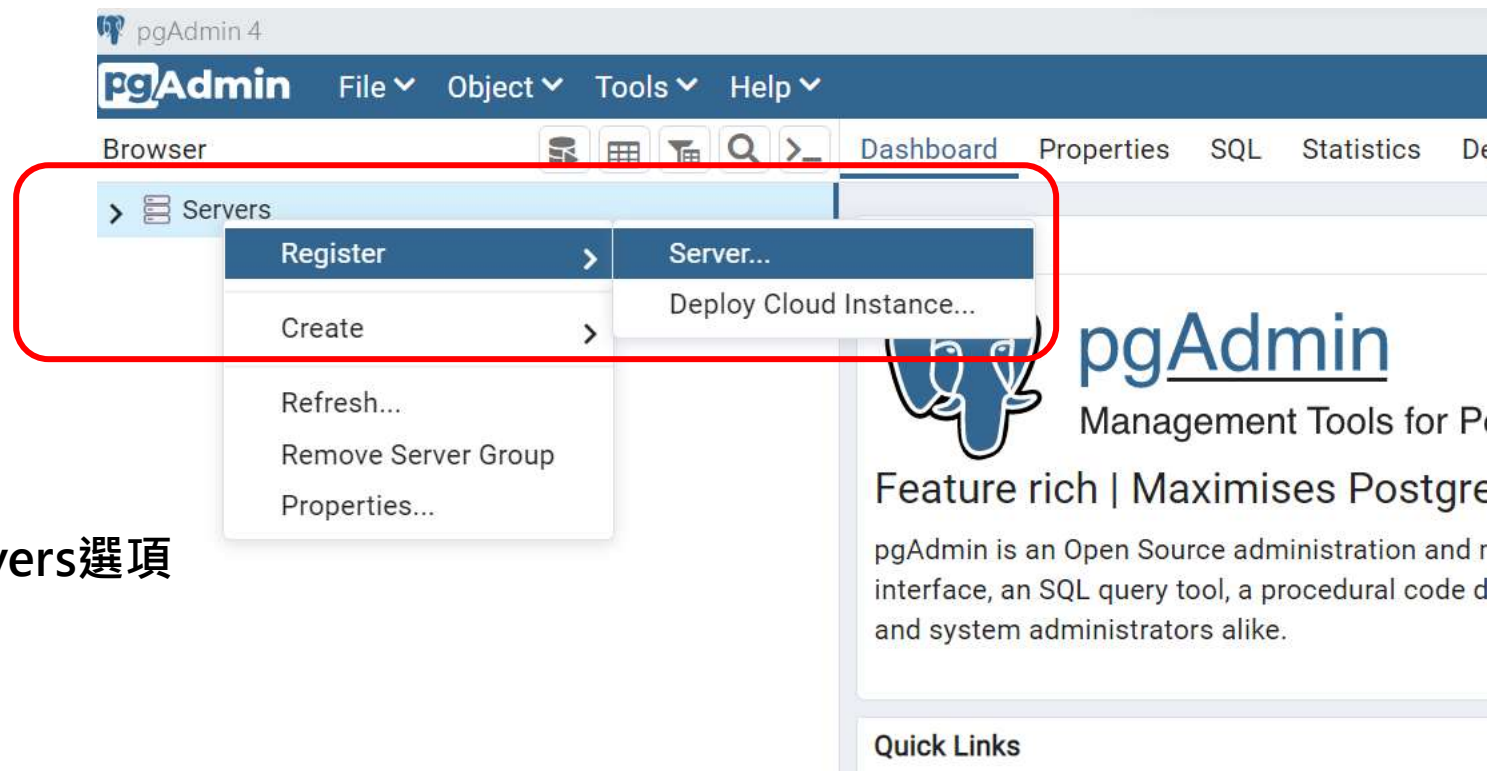
建立資料庫:

`docker exec -it postgres14 createdb -username=root -owner hello_sql`

刪除資料庫:

`docker exec -it postgres14 dropdb hello_sql`

# 方法(三) 使用 pgAdmin 連線並結合 Makefile 操作資料庫



- 滑鼠右鍵點選Servers選項
- Register
- Server



# 設定General & Connection

The image displays two screenshots of the 'Register - Server' dialog box, illustrating the configuration steps for a server.

**Left Screenshot (General Tab):**

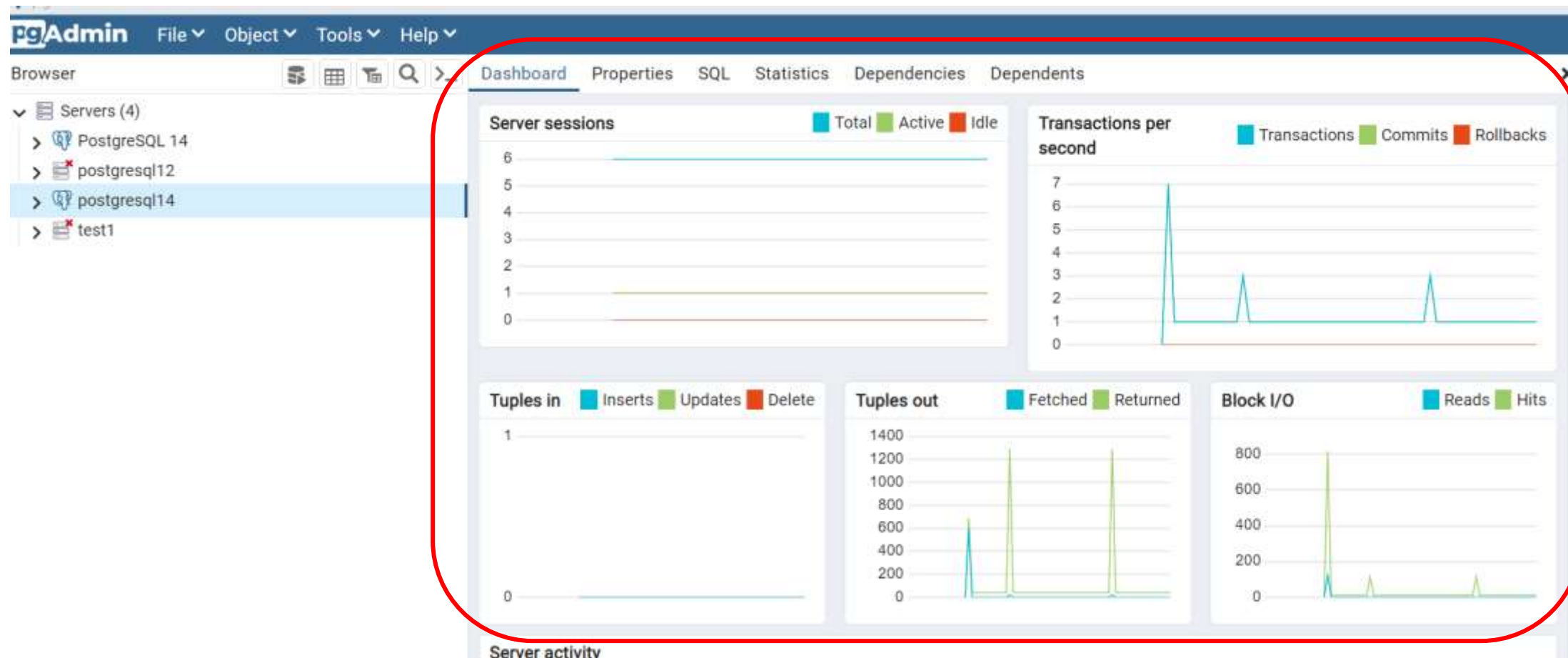
- Name:** postgresql14 (highlighted with a red box)
- Server group:** Servers
- Background:** ☐
- Foreground:** ☐
- Connect now?:** ☒
- Comments:** (empty text area)

**Right Screenshot (Connection Tab):**

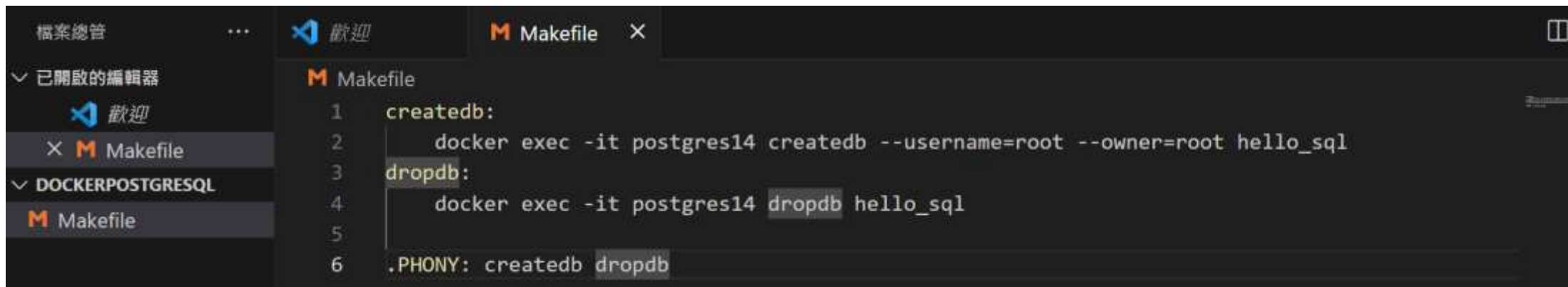
- Host name/address:** 127.0.0.1 (highlighted with a red box)
- Port:** 5433
- Maintenance database:** root
- Username:** root
- Kerberos authentication?:** ☐
- Password:** ....
- Save password?:** ☐
- Role:** (empty text field)
- Service:** (empty text field)

An arrow points from the 'Name' field in the General tab to the 'Host name/address' field in the Connection tab, indicating the flow of configuration.

# Dashboard有流量代表連線成功



# 撰寫Makefile簡化指令



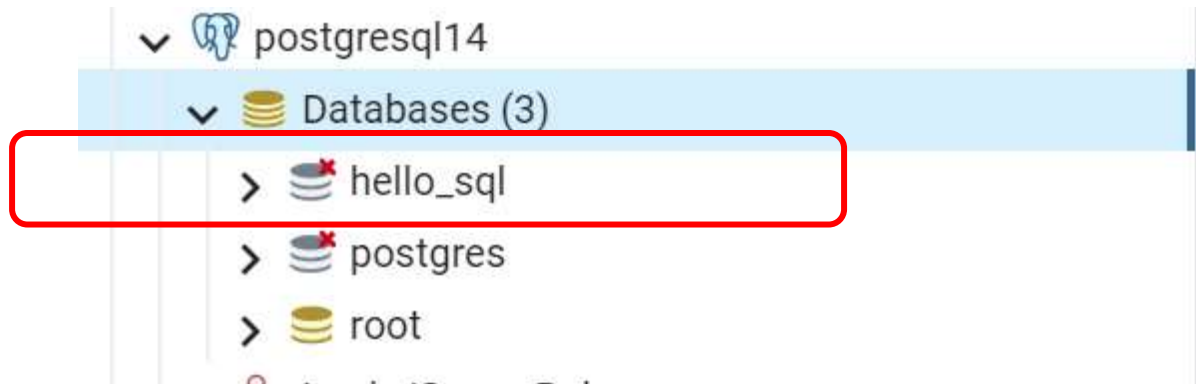
The screenshot shows the Visual Studio Code interface with a Makefile open. The left sidebar shows the '已開啟的編輯器' (Opened Editors) section with 'Makefile' selected. The main editor area displays the following content:

```
1 createdb:
2     docker exec -it postgres14 createdb --username=root --owner=root hello_sql
3 dropdb:
4     docker exec -it postgres14 dropdb hello_sql
5
6 .PHONY: createdb dropdb
```

- 於vscode中建立Makefile檔案
- 自定義簡化指令，EX: dropdb
- 並將docker的指令寫入，EX: docker exec -it postgres14 dropdb hello\_sql

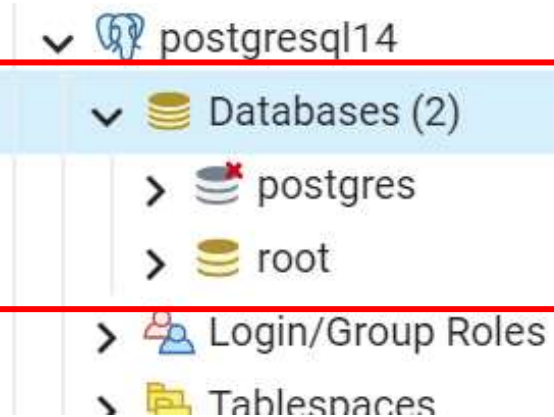
# 測試 make createdb

```
PS C:\Users\jerry\Desktop\master course\dataEngineer\goPostgreK8s\dockerPostgreSQLlocal筆記\dockerPostgreSQL> make creat  
edb  
docker exec -it postgres14 createdb --username=root --owner=root hello_sql  
PS C:\Users\jerry\Desktop\master course\dataEngineer\goPostgreK8s\dockerPostgreSQLlocal筆記\dockerPostgreSQL>
```



# 測試 make dropdb

```
PS C:\Users\jerry\Desktop\master course\dataEngineer\goPostgreK8s\dockerPostgreSQLlocal筆記\dockerPostgreSQL> make dropdb  
docker exec -it postgres14 dropdb hello_sql  
PS C:\Users\jerry\Desktop\master course\dataEngineer\goPostgreK8s\dockerPostgreSQLlocal筆記\dockerPostgreSQL>
```



確認刪除

# 停止docker 運作

```
PS C:\Users\jerry> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
a122411c10bf   postgres:14-alpine "docker-entrypoint.s..." 31 minutes ago Up 31 minutes  0.0.0.0:5433->5432/tcp    postgres14

PS C:\Users\jerry> docker stop postgres14
postgres14

PS C:\Users\jerry> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
PS C:\Users\jerry> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAME
a122411c10bf   postgres:14-alpine "docker-entrypoint.s..." 32 minutes ago Exited (0) 18 seconds ago    post
gres14
b3f218ec345c   postgres:12-alpine "docker-entrypoint.s..." 47 hours ago   Exited (0) 18 hours ago     post
gres12
PS C:\Users\jerry>
```

docker ps: 查看運行中的container

docker stop <container name>: 指定要停止運作的container

docker ps -a: 查看存在的container 不論是否運作

**End**