在 RHEL/CENTOS 上使用 PostgreSQL 資料庫

PostgreSQL

- PostgreSQL 是自由的關聯式資料庫伺服器(DBMS)，授權模式採取 BSD 授權許可最大好處是可以用於商業營運或修改成為自己的產品一部分而無須支付任何費用也不必擔心需要將修改過後的原始碼公開等眾多 GPL 授權軟體需要面對的問題。它具有商業資料庫絕大部分的功能，對於中小企業的應用十分的足夠。相較於 MySQL 這開放的資料庫系統，他有更完整的關聯式資料庫功能及優秀的穩定性。

為何要用 postgresql?

- 為何要用轉用 postgresql?
  - 我需要一個可以商業化的資料庫：PostgreSQL 採用授權較為寬鬆的 BSD 授權，對於商業公司而言不用擔心日後 PostgreSQL 的開發人員突然反悔改成其他授權模式的影響。
  - 一個強大的資料庫引擎：交易、觸發、內存程序等完整的關聯式資料庫機制，提供開發人員在開發過程中更方便的操作。
  - 一個取得容易，價格低廉的資料庫引擎：自由的 BSD 授權提供最便宜的成本。世界各地都有相關的 Mirror site。
  - 商用資料庫知名產品經常的更新版本及更新版本就收錢的風格令人不敢使用。
- MySQL 同樣是開放的資料庫管裡系統，為何選擇 PostgreSQL?
  - MySQL 本來是各很棒的選項，但是自從被 SUN 收購，而 SUN 又和 Oracle 合併後眾人從沒看好 Oracle 會好好善待它!!從 Oracle 對待 OpenSolaris 這各自由軟體專案的態度就可以想像以後 MySQL 可能的下場。
  - MySQL 在 5 版之前和 PostgreSQL 的定位並不衝突，MySQL 自從 4 版及 5 版加入觸發、內存程序、交易等功能後，雖然是各更完整的資料庫系統，但變得相對不穩定，且速度也因為更複雜的處理機制讓它疑似變慢了。

PostgreSQL 的安裝

postgresql 的套件安裝

- postgresql 的套件安裝
  - 安裝 postgresql server 及文字 client 端工具

```
[root@lab ~]# yum install postgresql postgresql-server
```

- 驗證你所安裝的套件

```
[root@lab ~]# rpm -qa | grep postgresql
postgresql-8.1.23-1.el5_6.1
postgresql-server-8.1.23-1.el5_6.1
postgresql-libs-8.1.23-1.el5_6.1
```

- 重新啟動 postgresql 服務

```
[root@lab ~]# /etc/init.d/postgresql restart
正在停止 postgresql 服務:                                 [失敗]
正在初始化資料庫:                                         [確定]
正在啟動 postgresql 服務:                                 [確定]
```

- postgresql 程式在 linux 上會以一個名稱為 postgres 的使用者帳戶執行，

```
[root@lab ~]# finger postgres
Login: postgres                         Name: PostgreSQL Server
Directory: /var/lib/pgsql               Shell: /bin/bash
Never logged in.
No mail.
No Plan.
```

- 如果你仔細觀察會發現 postgres 使用者的家在 /var/lib/pgsql 目錄，目錄內 data 包涵了所有 postgresql 所有需要用到的檔案內容。包含所有的設定檔、紀錄檔及資料庫檔案都在裡面，備份時只要把整各打包備份即可。

```
[root@lab sql]# ls /var/lib/pgsql/ -la
總計 72
drwx------  4 postgres postgres 4096  4月 29 19:56 .
drwxr-xr-x 34 root     root     4096  4月 29 14:55 ..
drwx------  2 postgres postgres 4096  3月 31 01:23 backups
-rw-------  1 postgres postgres 1082  4月 29 19:56 .bash_history
-rw-r--r--  1 postgres postgres   85  3月 31 01:23 .bash_profile
drwx------ 11 postgres postgres 4096  4月 29 19:32 data
-rw-------  1 postgres postgres 2277  4月 29 18:29 pgstartup.log
-rw-------  1 postgres postgres  386  4月 29 18:04 .psql_history
-rw-------  1 postgres postgres 1417  4月 29 19:54 .viminfo
[root@lab sql]# ls /var/lib/pgsql/data/ -la
總計 148
```

```
drwx------ 11 postgres postgres  4096  4月 29 19:32 .
drwx------  4 postgres postgres  4096  4月 29 19:56 ..
drwx------  6 postgres postgres  4096  4月 29 23:36 base
drwx------  2 postgres postgres  4096  4月 29 23:43 global
drwx------  2 postgres postgres  4096  4月 29 14:56 pg_clog
-rw-------  1 postgres postgres  3330  4月 29 18:30 pg_hba.conf
-rw-------  1 postgres postgres  1460  4月 29 14:56 pg_ident.conf
drwx------  2 postgres postgres  4096  4月 29 14:56 pg_log
drwx------  4 postgres postgres  4096  4月 29 14:56 pg_multixact
drwx------  2 postgres postgres  4096  4月 29 14:56 pg_subtrans
drwx------  2 postgres postgres  4096  4月 29 14:56 pg_tblspc
drwx------  2 postgres postgres  4096  4月 29 14:56 pg_twophase
-rw-------  1 postgres postgres     4  4月 29 14:56 PG_VERSION
drwx------  3 postgres postgres  4096  4月 29 14:56 pg_xlog
-rw-------  1 postgres postgres 13804  4月 29 18:09 postgresql.conf
-rw-------  1 postgres postgres    57  4月 29 18:29 postmaster.opts
-rw-------  1 postgres postgres    45  4月 29 18:29 postmaster.pid
```

在資料庫系統上，建立使用者與資料庫

- 資料庫使用者
  - 資料庫使用者從概念上與系統上面的使用者是完全無關的。但是在 postgresql 資料庫使用者名稱(roles)在整個資料庫中預設會透過和 Linux 系統帳戶同樣的帳戶名稱認證，這是資料庫為了安全性所設計的驗證方式之一。
  - 一般在工作上，通常會為一個專案工作建立一個專用的資料庫帳戶及及資料庫，並將權限獨立給該資料庫使用者(roles)可以存取該資料庫並指定允許的資料庫存取範圍。避免因為單一資料庫使用者被入侵後影響整個資料庫管理系統。
- 在 SQL 指令中可以透過底下的指令建立資料庫使用者(roles)及資料庫。
  - 建立一個資料庫使用者(roles)名稱為 name 及密碼為 string 的使用者

```
CREATE USER name WITH PASSWORD 'string';
```

  - 建立一個資料庫 dbname

```
CREATE DATABASE dbname;
```

- 在剛裝好 postgresql 的時候，系統只有純文字的 SQL 管理工具(psql)可操作，並且設定好了一個有個資料庫使用者(roles)叫做 postgres ，他可以透過作業系統的系統的 postgres 使用者身份登入 postgresql 資料庫系統，並且取得最高管理者的權限。

- 切換成為 postgres 使用者，因為在 RHEL 中預設系統登入系統使用的為 ident sameuser 的認證模式，所以使用 Lnux OS postgres 使用者身份登入資料庫系統就可以得到最高資料庫系統的管理權限且過程中不需要密碼。這裡登入資料庫的方式是使用命令列的 psql 這這指令。

```
# root 切換成 postgres
[root@lab data]# su - postgres
# 登入  postgresql 資料庫，因為系統預設為 ident sameuser 的認證模式，所以不用密碼就可以登入
[root@lab pg_log]# su - postgres
-bash-3.2$ psql  template1
Welcome to psql 8.1.23, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

# 修改 postgre 的預設密碼，成為 1234
template1=# alter user postgres with password '1234';
ALTER ROLE
# 離開資料庫
template1=# \q
```

- 這樣你就完成了資料庫的最高管理者 postgresql 的密碼修改。

更改 PostgreSQL 的登入認證模式

- 但是這樣的登入方式使用者只能在本機上工作，對於網頁程式的開發很不方便。我們可以透過修改登入的認證模式讓系統可以透過 TCP/IP 連線的方式登入系統，透過網路的分享方式可以讓本機或遠端的程式透過網路直接存取這台電腦的資料庫；要小心的是開放了網路連線也相對的把資料庫開放給全部的網路使用，所以在開放權限上需要進行適當的管控。
- pg_hba.conf 設定值說明 http://developer.postgresql.org/pgdocs/postgres/auth-pg-hba-conf.html
- 更改 postgresql 的認證模式增加 host 認證方式

```
[root@lab ~]# vim /var/lib/pgsql/data/pg_hba.conf
# 省略很多...
# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD
# "local" is for Unix domain socket connections only
# 開放本地端同資料庫使用者帳號的 linux 使用者可以不用密碼從本機登入系統
```

```
local   all         all                               ident sameuser
# IPv4 local connections: IPV4 本地端
host    all         all         127.0.0.1/32          ident sameuser
# IPv6 local connections: IPV6 本地端
host    all         all         ::1/128               ident sameuser

# IPv4 local connections:
# 可以透過 tcp/ip 從 127.0.0.1/32 及 110.111.69.0/24 登入
host    all         all         127.0.0.1/32          md5
host    all         all         110.111.69.0/24       md5
```

- 原來 Postgresql 的連線只開放本機 IP 可以存取，如果需要讓外面 IP 可以存取，需要打開預設的連線設定檔案 /var/lib/pgsql/data/postgresql.conf 將 listen_addresses = 'localhost' 修改為 listen_addresses = '*' 結果如下：

```
[root@lab ~]# vim /var/lib/pgsql/data/postgresql.conf
# - Connection Settings -

listen_addresses = '*'                  # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost', '*' = all
```

- 重新啟動 postgresql 服務

```
[root@lab data]# /etc/init.d/postgresql restart
正在停止 postgresql 服務:                                 [  確定  ]
正在啟動 postgresql 服務:                                 [  確定  ]
```

- 驗證 postgresql 服務是否已開啟(TCP port 5432)

```
[root@lab ~]# netstat -an | grep 5432
tcp        0      0 0.0.0.0:5432                0.0.0.0:*                   LISTEN
tcp        0      0 :::5432                     :::*                        LISTEN
unix  2      [ ACC ]     STREAM     LISTENING     6984191 /tmp/.s.PGSQL.5432
```

- psql 使用 host 的登入方式登入

```
[root@lab pg_log]# psql -U postgres -h 127.0.0.1
Password for user postgres:
Welcome to psql 8.1.23, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
```

```
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

postgres=# \q
```

- 設定不正確，可能會發生下面的錯誤。

```
[root@lab ~]# psql -U jangmt
psql: 嚴重錯誤:  Ident 驗證使用者"jangmt"失敗
```

- 登入系統預設會以使用者帳號登入系統，如果沒有此資料庫會產生錯誤警告。

```
[jangmt@lab ~]$ psql -U jangmt
psql: 嚴重錯誤:  資料庫"jangmt"不存在
```

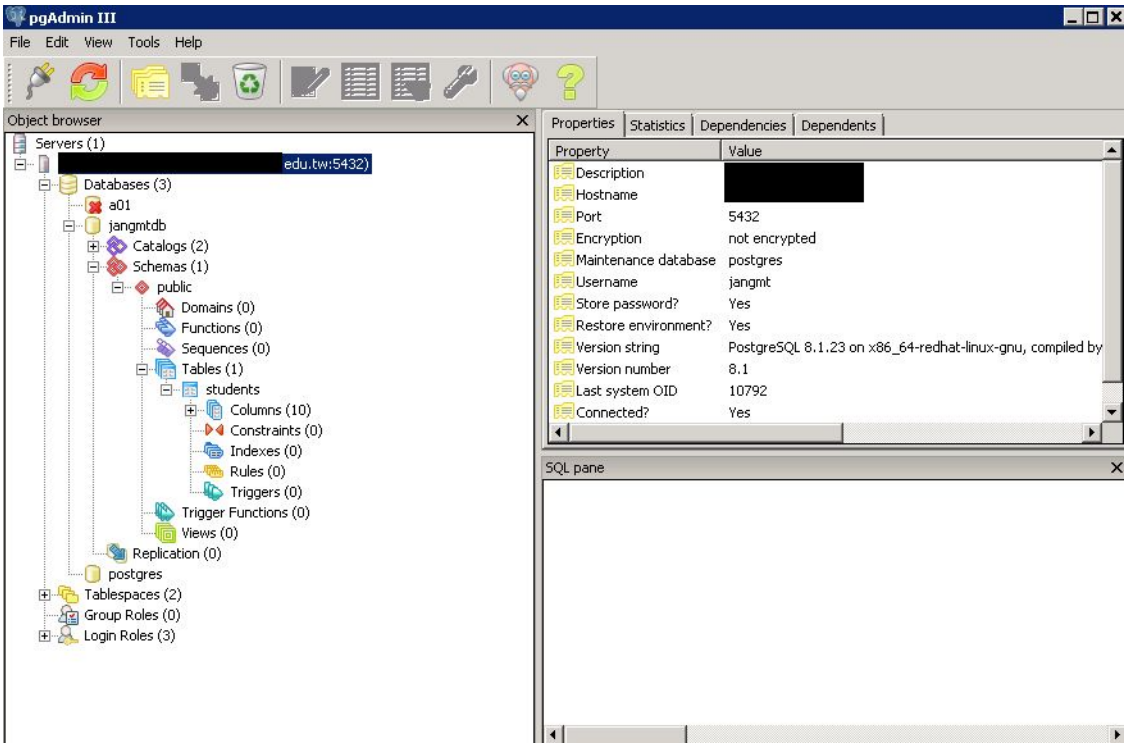
- 在認證過程中如果發生問題，請觀看記錄檔 /var/lib/pgsql/data/pg_log/postgresql-Fri.log 可以獲得類似的下面的訊息紀錄。

```
[root@lab pg_log]# tail /var/lib/pgsql/data/pg_log/postgresql-Fri.log  -n 20
LOG:  下一個交易 ID：662，下一個 OID：16394
LOG:  next MultiXactId: 1; next MultiXactOffset: 0
LOG:  資料庫系統待命
LOG:  transaction ID wrap limit is 2147484146, limited by database "postgres"
LOG:  could not connect to Ident server at address "127.0.0.1", port 113: 連線被拒絕
嚴重錯誤:  Ident 驗證使用者"postgres"失敗
錯誤:  資料庫"sqladmin"不存在
LOG:  transaction ID wrap limit is 2147484146, limited by database "postgres"
LOG:  transaction ID wrap limit is 2147484146, limited by database "postgres"
LOG:  收到快速關閉的要求
LOG:  正在關閉
LOG:  資料庫系統已關閉
LOG:  正在關閉 logger
LOG:  資料庫系統於 2011-04-29 18:29:29 CST 被關閉
LOG:  檢查點記錄於 0/398728
LOG:  redo 記錄於 0/398728，undo 記錄於 0/0，關閉 TRUE
LOG:  下一個交易 ID：674，下一個 OID：16394
LOG:  next MultiXactId: 1; next MultiXactOffset: 0
LOG:  資料庫系統待命
```

```
LOG:  transaction ID wrap limit is 2147484146, limited by database "postgres"
```

安裝 PostgreSQL 管理工具

- PhpPgAdmin 是一套使用 php 程式編寫，透過 web 介面來管理 postgresql 資料庫的資料庫管理工具。它和 mysql 上面 phpmyadmin 很類似，事實上很多概念是從那裡參考而來。但是加入了很多 postgresql 特有的功能。在安裝過程中如果有問題可以參考官方網站上面的 FAQ 有很多解決的方式及已知的問題。
- FAQ http://phppgadmin.sourceforge.net/doku.php?id=faq_docs
- 另外一套比較知名的管理工具是 pgAdmin III 這是一套圖形化介面的管理工具。提供更多友善的設計，如：圖形化的查詢設計師，對 TSearch 全文搜索引擎 的支持 .....等眾多商業化軟體才有的功能..



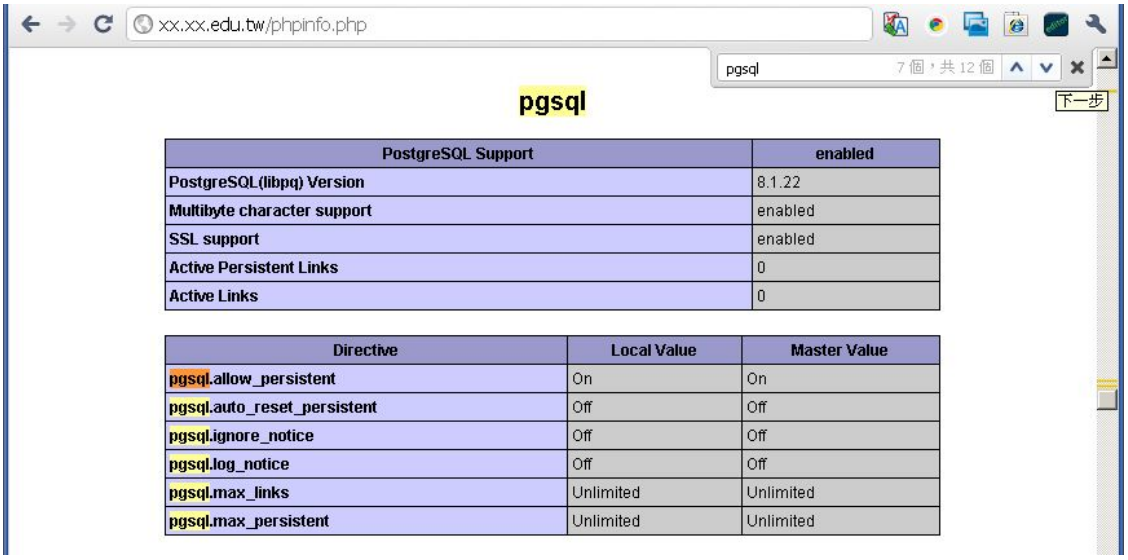
- 安裝 phpmyadmin 前請先確認你的 linux 已經有 apache + php 的功能支援，如果沒有可以透過 yum 指令快速的安裝這些套件。

```
yum -y install httpd httpd-devel httpd-manual
yum -y install php-mysql php-mbstring \
php-soap php-xml php-mcrypt php-pear php-cli php-devel php-gd
```

- 並可以透過 php 的 phpinfo(); 函示確認 php 是否已經有支援 postgresql 的存取。

```
[root@lab html]# vim /var/www/html/phpinfo.php
<? phpinfo(); ?>
```

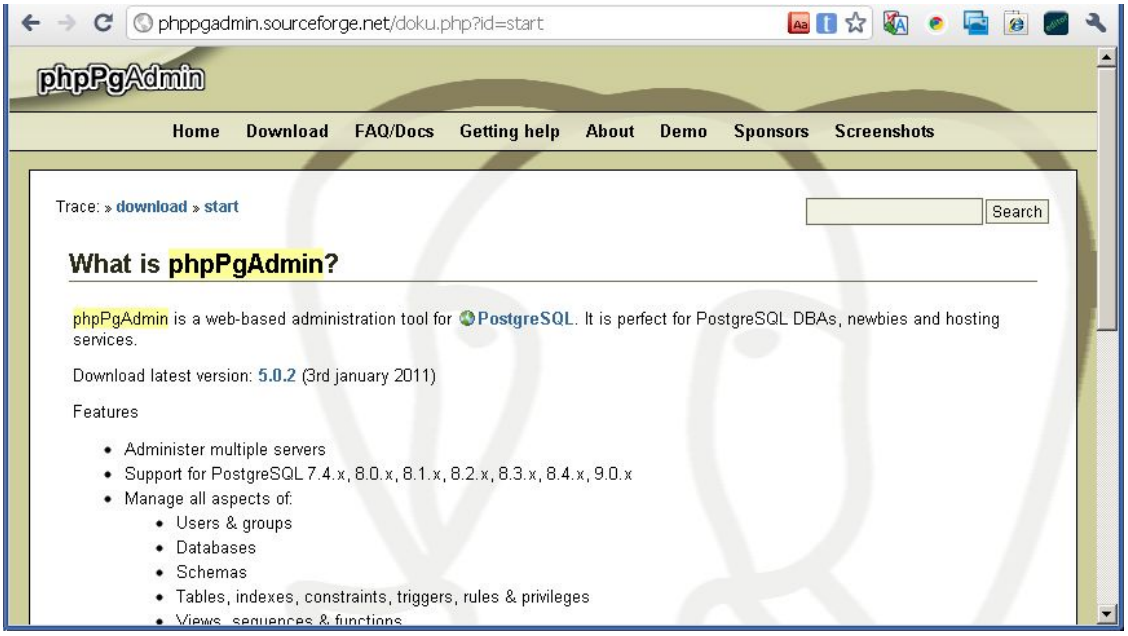
- 當網頁出現 psql 的函示庫，表示該 php 有支援 postgresql 的存取。



- 底下的範例不使用官方預設的 phpmyadmin 套件安裝，如果你想使用官方預設把包好的套件，請自行用 yum 安裝程式。

```
[root@lab html]# yum search phpmyadmin
phpMyAdmin.noarch : Web based MySQL browser written in php
```

- 從官方網站下載最新版本的 phppgadmin http://phppgadmin.sourceforge.net



- 解開放到目錄 /var/www/html 內

```
[root@lab html]# ls /var/www/html/ -l
總計 1072
-rw-r--r--  1 root root      17  4月 29 18:50 phpinfo.php
drwxr-xr-x 12 root root    4096  1月  4 03:23 phpPgAdmin-5.0.2
```

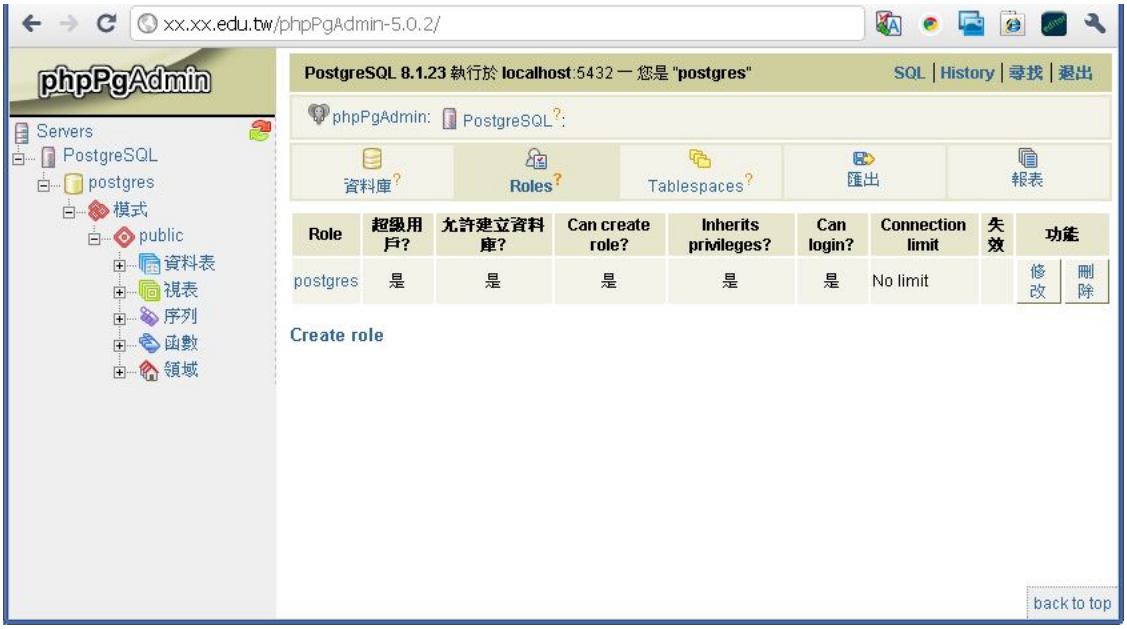
- 請修改 phpPgAdmin 設定檔

```
[root@lab html]# vim /var/www/html/phpPgAdmin-5.0.2/conf/config.inc.php


        // Hostname or IP address for server.  Use '' for UNIX domain socket.
        // use 'localhost' for TCP/IP connection on this computer
        // 底下欄位加入 localhost 內容，指定此程式連接到放在本機 host 上
        $conf['servers'][0]['host'] = 'localhost';


        //把底下這行最後的 true 更換為 false
        //$conf['extra_login_security'] = true;
        $conf['extra_login_security'] = false;
```

- 修改完成後，直接輸入剛剛更改的 postgresql 最高管理者的帳號及密碼就可以登入資料庫管理系統開始工作。



建立使用者及匯入資料

- 在上面部分已經完成最高管理者登入網頁管理介面的設定，對於一個專案或工作來說絕對不要用最高管理員來寫程式，因為 SQL injection 在程式上面的防堵常會因為程式設計師的不小心而發生漏洞，造成不可預期的狀況。所以底下將說明如何建立一個普通受限制的使用者及資料庫，並且匯入些資料。
- 在 shell 中雖然有提供 createdb 及 createuser 這些指令，但在使用上並不會比單純的 psql 配合 sql 語法來的簡單好用。所以底下以 psql + sql 語法建立資料庫及作相關的操作。

建立使用者

- -c 參數可以直接在命令列下 SQL 敘述句
    - -e 參數是將 SQL 的執行指令過程全部列出來

```
[root@lab sql]# psql -h localhost -U postgres -W -e -c "CREATE USER jangmt WITH PASSWORD '密碼' ;"
Password for user postgres:
CREATE USER jangmt WITH PASSWORD '密碼' ;
CREATE ROLE
```

建立資料庫

- 建立資料庫前一定要確定已經建立使用者，否則 OWNER 後面的這個使用者身份會發生語法錯誤。

```
[root@lab sql]# psql -h localhost -U postgres -W -e -c "CREATE DATABASE jangmtDB OWNER jangmt ENCODING 'UTF8';"
Password for user postgres:
CREATE DATABASE jangmtDB OWNER jangmt ENCODING 'UTF8';
CREATE DATABASE
```

匯入資料

- -f 參數可以讀取檔案 sql.txt 內的 SQL 敘述輸入到資料庫系統內

```
[root@lab sql]# psql -h localhost -U jangmt -d jangmtdb -W -e -f sql.txt
Password for user jangmt:

# 底下開使用 SQL 敘述，如需測試可以複製成為檔案 sql.txt 在測試時使用
CREATE TABLE STUDENTS (
  學號 numeric(15,5) default NULL,
  班級座號 varchar(9) default NULL,
  姓名 varchar(7) default NULL,
  出生年月日 date default NULL,
  身分証號碼 varchar(11) default NULL,
  住址 varchar(33) default NULL,
  家長 varchar(9) default NULL,
  電話 varchar(15) default NULL,
  科別 varchar(10) default NULL,
  畢業國中 varchar(22) default NULL
);
CREATE TABLE
INSERT INTO STUDENTS VALUES (911001.00000, '10101', '王于穎', '1984-03-05 00:00:00', 'C100000012',
'基隆市安樂區安和一街 4 巷 4-3 號 4F', '王世傑', '02-24310667', '商業經營科', '基隆市市立安樂國中畢業');
INSERT 0 1
INSERT INTO STUDENTS VALUES (911002.00000, '10102', '王彗如', '1980-09-08 00:00:00', 'F200000026',
```

```
'台北縣瑞芳鎮一坑路 426 號', '王文淵', '02-24971835', '商業經營科', '台北縣縣立瑞芳國中畢業');
INSERT 0 1
INSERT INTO STUDENTS VALUES (911003.00000, '10103', '王琇榆', '1984-08-28 00:00:00', 'F200000035',
'基隆市暖暖區源遠路 292 巷 1-5 號 1F', '王進豐', '02-24570828', '商業經營科', '基隆市市立暖暖國中畢業');
INSERT 0 1
INSERT INTO STUDENTS VALUES (911004.00000, '10104', '朱勝真', '1984-08-28 00:00:00', 'F100000042',
'台北縣瑞芳鎮逢甲路 337 號', '朱水順', '02-24970773', '商業經營科', '台北縣縣立瑞芳國中修業');
INSERT 0 1
```

- 執行 sql select 指令,確認資料都有匯入系統內。

```
[root@lab sql]# psql -h localhost -U jangmt -d jangmtdb -W -e -c 'select 學號,班級座號,姓名 from students;'
Password for user jangmt:
select 學號,班級座號,姓名 from students;
     學號      | 班級座號 |  姓名
--------------+----------+--------
 911001.00000 | 10101    | 王于穎
 911002.00000 | 10102    | 王彗如
 911003.00000 | 10103    | 王琇榆
 911004.00000 | 10104    | 朱勝真
(4 行)
```

匯出資料

- pg_sump 可以將資料庫依需求匯出，以供備份或轉移系統使用。

```
[root@lab sql]# pg_dump -f sql_dump.txt -h localhost -U jangmt -W jangmtdb
```

```
密碼：


# 把備份檔內容 sql_dump.txt 呈現
[root@lab sql]# cat sql_dump.txt
--
-- PostgreSQL database dump
--
SET client_encoding = 'UTF8';
SET check_function_bodies = false;
SET client_min_messages = warning;
--
-- Name: SCHEMA public; Type: COMMENT; Schema: -; Owner: postgres
--
COMMENT ON SCHEMA public IS 'Standard public schema';
SET search_path = public, pg_catalog;
SET default_tablespace = '';
SET default_with_oids = false;
--
-- Name: students; Type: TABLE; Schema: public; Owner: jangmt; Tablespace:
--
CREATE TABLE students (
    "學號" numeric(15,5),
    "班級座號" character varying(9),
    "姓名" character varying(7),
    "出生年月日" date,
    "身分証號碼" character varying(11),
    "住址" character varying(33),
    "家長" character varying(9),
    "電話" character varying(15),
    "科別" character varying(10),
    "畢業國中" character varying(22)
);
-- ...資料太多予以省略....
```

shell script 建立使用者資料

- 如果要透過 shell script 方便建立資料庫的使用者帳號及資料庫，可以使用 psql 指令並用配合 postgres 這個超級使用者登入系統來執行必要得 SQL 敘述。因為在命令列的工具 psql 並沒有提供密碼預設輸入的功能，只能透過鍵盤互動輸入。增加了安全姓，相對也造成自動化處理上的麻煩。
- 所以在這裡我使用了 pg_hba.conf 的 ident sameuser 認證方式，讓只要是 postgres 使用者執行的 psql 都不用輸入密碼即可自動執行 SQL 敘述，來執行自動化的作業。

- 切換到 postgres 帳號及測試 psql 登入免密碼。

```
[root@lab tmp]# su - postgres
-bash-3.2$ psql
Welcome to psql 8.1.23, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

postgres=# \q
```

- 上面的範例沒有問題的話請用編輯器建立底下的 shell script

```
-bash-3.2$ cat ./pssqladd.sh
#!/bin/bash
# get argc
sqlaccount=$1
sqlpassword=$2

# check $1 and $2
if !(test -z $sqlaccount;) then
  if !(test -z $sqlpassword;) then
        echo "account is $sqlaccount , DB is $sqlaccount , password is $sqlpassword"
        # sql
        echo "CREATE USER $sqlaccount WITH PASSWORD '$sqlpassword';
CREATE DATABASE $sqlaccount OWNER $sqlaccount ENCODING 'UTF8'; " > /tmp/pgsqladd.txt
        echo "psql -f /tmp/pgsqladd.txt" | bash
  else
        echo 'error!!! need DB user password'
        echo 'usage:./pssqladd.sh DBusername DBuserpassword'
```

```
  fi
else
        echo 'error!!! need DB user name'
        echo 'usage:./pssqladd.sh DBusername DBuserpassword'
fi
```

- 記得賦予此 shell 可執行權或使用 bash pssqladd.sh 方式執行 shell script

```
-bash-3.2$ chmod +x pssqladd.sh
```

- 執行 script ,建立 a01 帳號及 a01 資料庫,密碼為 a01test

```
-bash-3.2$ ./pssqladd.sh a01 a01test
account is a01 , DB is a01 , password is a01test
CREATE ROLE
CREATE DATABASE
```

- 驗證

- 以上為 postgresql 簡單的安裝及使用，如果需要對資料庫的 SQL 語言有更進一步的認識可以參考結構化查詢語言(SQL)的書籍或是相關的課程 http://www.lccnet.com.tw/zsergb026-test/data/data1.html 。

參考資料

- http://wwwmaster.postgresql.org
- http://www.postgresql.org/
- http://phppgadmin.sourceforge.net/
- http://postgresql-chinese.blogspot.com/2007/02/postgresql-bin.html

- http://forum.yoursun.com.tw/forum_posts.asp?TID=1380