海克力斯：
PostgreSQL 生態共榮圈

鍾明達，張友謙，孫菖鴻

講者介紹

鍾明達

一位系統開發的工程師

張友謙

已經在Coscup講PG很多次了.

孫菖鴻

我是Eagle(老鷹)，是一個PHP與PostgreSQL 的使用者，喜歡分享與研究PostgreSQL的技術。

前言

今年某一個晚上...

一如往常閒聊最近資訊產業相關事情,
資訊安全已經是家常便飯。不管在雲端或實體Server上安裝軟體都需要一些控管。

對於系統開發及維運來說，
監控系統佔有相當的重要性。

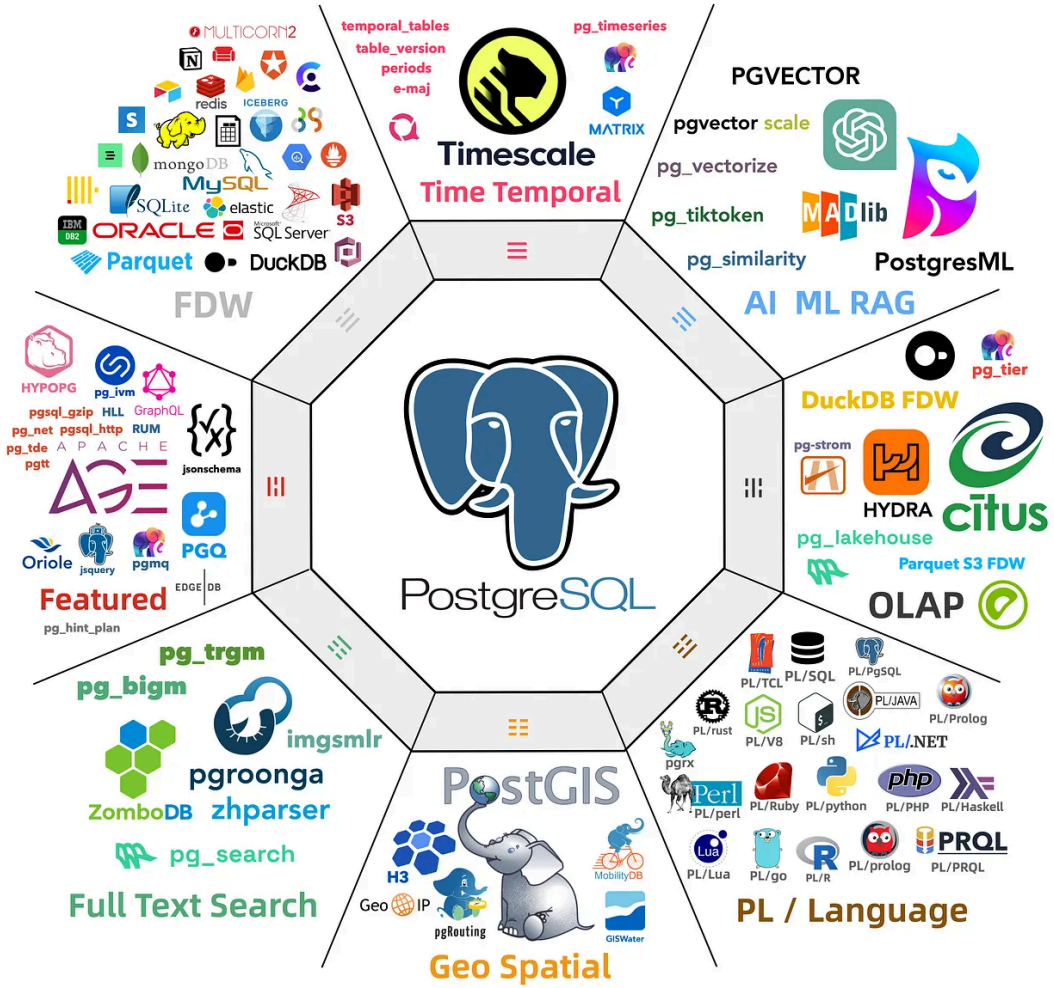
那有什麼方式可以簡化，但又可以兼具監控的本質？

最後...
不如動手做，把想法放去資料庫吧！？
於是，有了下面 PostgreSQL 展開...

圖片來源：https://www.chilling.tw/article/86538

PostgreSQL 只是資料庫嗎？

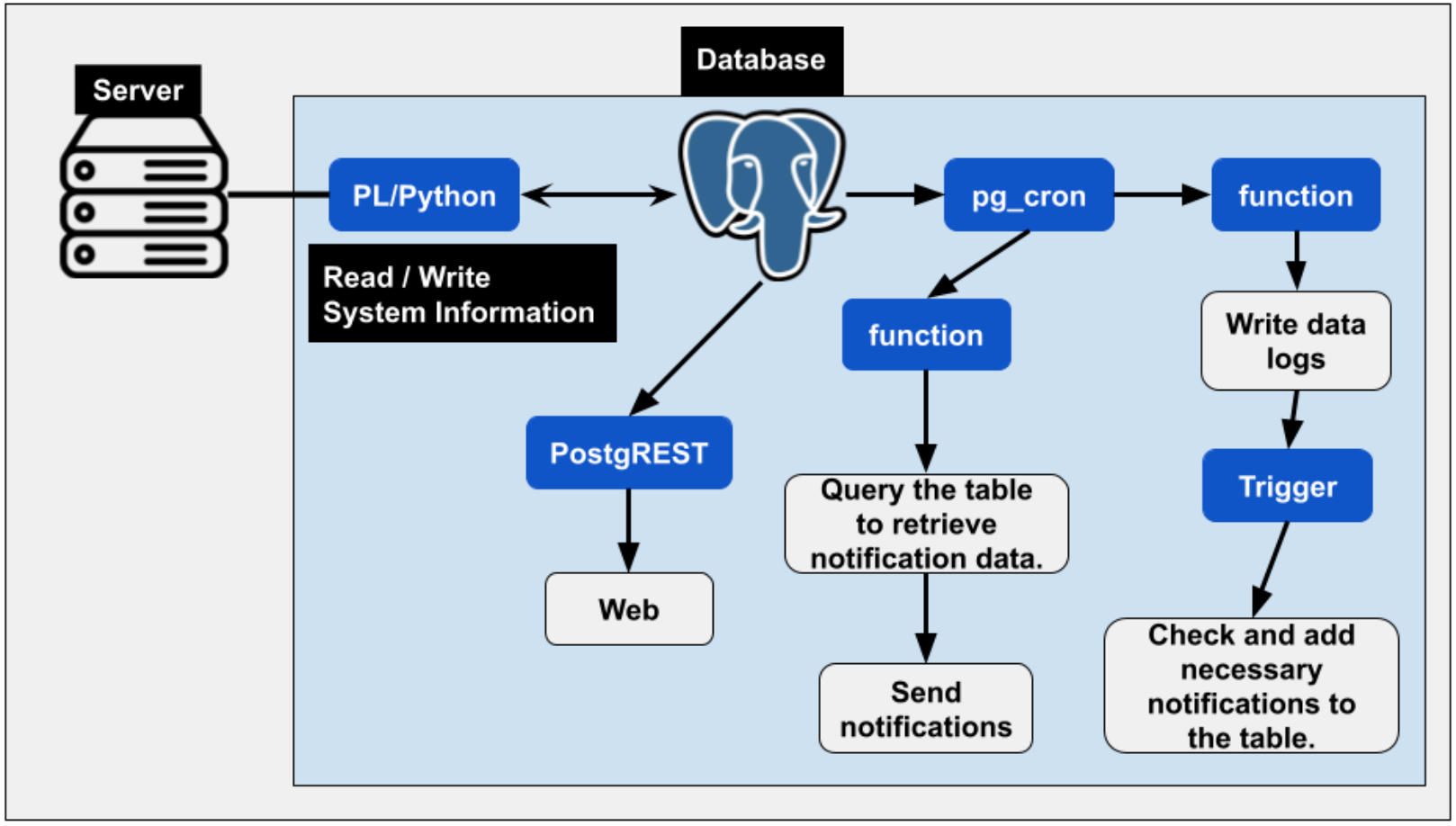
它可能超乎你的想像...
在你工作日常隨處可見，只是你沒發現



圖片來源：https://medium.com/@fengruohang/postgres-is-eating-the-database-world-157c204dcfc4

話說回來...

海克力斯是什麼？

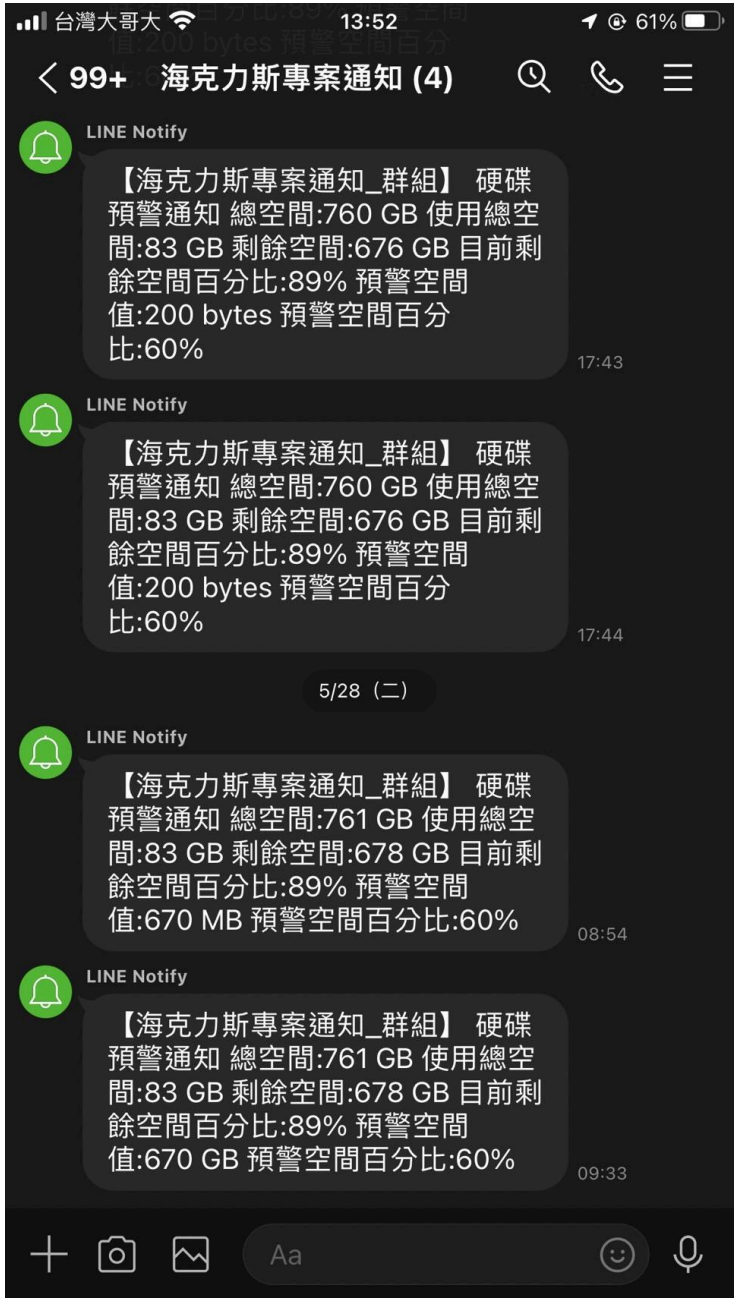

Server
Database
PL/Python
Read / Write
System Information
pg_cron
function
Write data
logs
Trigger
Check and add
necessary
notifications to
the table.
function
Query the table
to retrieve
notification data.
Send
notifications
PostgREST
Web

先看結果！

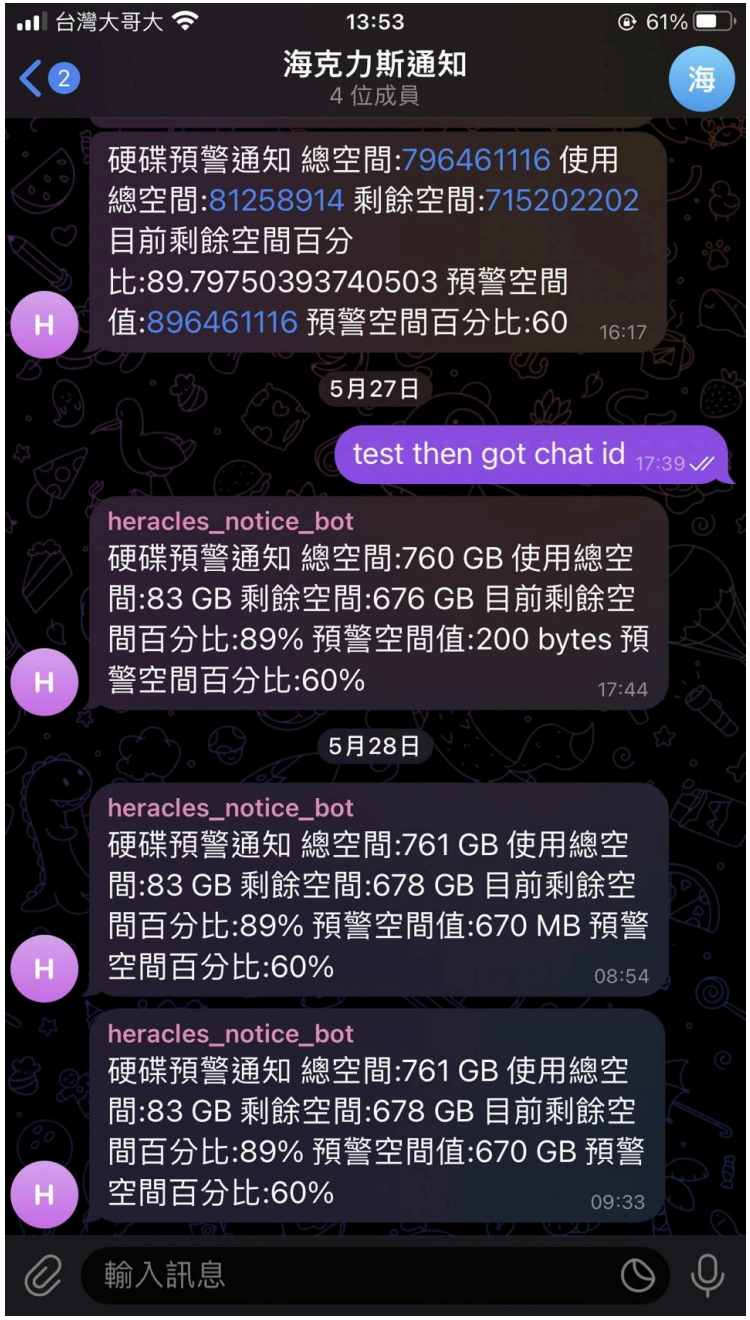
Slack


5 月 28 日週二
Disk_notify 應用程式 早上 8:54
硬碟預警通知 總空間:761 GB 使用總空間:83 GB 剩餘空間:678 GB 目前剩餘空間百分比:89% 預警空間值:670 MB 預警空間百分比:60%
Disk_notify 應用程式 上午 9:33
硬碟預警通知 總空間:761 GB 使用總空間:83 GB 剩餘空間:678 GB 目前剩餘空間百分比:89% 預警空間值:670 GB 預警空間百分比:60%

LINE


台灣大哥大
13:52
61%
99+
海克力斯專案通知 (4)
LINE Notify
【海克力斯專案通知_群組】 硬碟預警通知 總空間:760 GB 使用總空間:83 GB 剩餘空間:676 GB 目前剩餘空間百分比:89% 預警空間值:200 bytes 預警空間百分比:60%
17:43
LINE Notify
【海克力斯專案通知_群組】 硬碟預警通知 總空間:760 GB 使用總空間:83 GB 剩餘空間:676 GB 目前剩餘空間百分比:89% 預警空間值:200 bytes 預警空間百分比:60%
17:44
5/28（二）
LINE Notify
【海克力斯專案通知_群組】 硬碟預警通知 總空間:761 GB 使用總空間:83 GB 剩餘空間:678 GB 目前剩餘空間百分比:89% 預警空間值:670 MB 預警空間百分比:60%
08:54
LINE Notify
【海克力斯專案通知_群組】 硬碟預警通知 總空間:761 GB 使用總空間:83 GB 剩餘空間:678 GB 目前剩餘空間百分比:89% 預警空間值:670 GB 預警空間百分比:60%
09:33
Aa

Telegram


台灣大哥大
13:53
61%
海克力斯通知
4 位成員
海
硬碟預警通知 總空間:796461116 使用總空間:81258914 剩餘空間:715202202 目前剩餘空間百分比:89.79750393740503 預警空間值:896461116 預警空間百分比:60
16:17
5月27日
test then got chat id
17:39
heracles_notice_bot
硬碟預警通知 總空間:760 GB 使用總空間:83 GB 剩餘空間:676 GB 目前剩餘空間百分比:89% 預警空間值:200 bytes 預警空間百分比:60%
17:44
5月28日
heracles_notice_bot
硬碟預警通知 總空間:761 GB 使用總空間:83 GB 剩餘空間:678 GB 目前剩餘空間百分比:89% 預警空間值:670 MB 預警空間百分比:60%
08:54
heracles_notice_bot
硬碟預警通知 總空間:761 GB 使用總空間:83 GB 剩餘空間:678 GB 目前剩餘空間百分比:89% 預警空間值:670 GB 預警空間百分比:60%
09:33
輸入訊息

以下進入實作環節！

OS: CentOS Stream release 9, DB: PostgreSQL 16.2

- PostgreSQL 安裝
- PL/Python 安裝
- pg_cron 安裝設定
- TimescaleDB 安裝設定
- 建立資料模型
- 建立 PL/Python3 函式
- 輸入資料
- 建立 PL/Pgsql 函式、Trigger

- 建立 pg_cron 排程
- 透過 TimescaleDB
  建立 Hypertable
- PostgREST 建立 Web API

PostgreSQL 安裝

(*依個人使用作業系統不同，請參考官方安裝說明)

https://www.postgresql.org/download/

PL/Python3 安装

OS:

```
# dnf install postgresql16-plpython3
```

DB:

```
# create extension plpython3u;
```

pg_cron 安裝設定

OS:

```
# dnf install pg_cron_16
```

postgresql.conf

```
shared_preload_libraries = 'pg_cron'
cron.database_name = 'coscup2024'
cron.timezone = 'Asia/Taipei'
```

OS:

```
# systemctl restart postgresql-16.service
```

DB:

```
# create extension pg_cron;
```

TimescaleDB 安装設定

OS:

```
# dnf install timescaledb_16 timescaledb-tools

# timescaledb-tune
--pg-config=/usr/pgsql-16/bin/pg_config

# shared_preload_libraries =
'pg_cron,timescaledb'
```

DB:

```
create extension timescaledb;
```

目前安裝的 extension

DB:

\dx

名稱	版本	Schema	描述
pg_cron	1.6	pg_catalog	Job scheduler for PostgreSQL
plpgsql	1.0	pg_catalog	PL/pgSQL procedural language
plpython3u	1.0	pg_catalog	PL/Python3U untrusted procedural language
timescaledb	2.15.3	public	Enables scalable inserts and complex queries for time-series data (Apache 2 Edition)

(4 筆資料)

建立資料模型

```
create schema s0727;
```

以下開始建立 9 tables

table: s0727.config

```
CREATE TABLE s0727.config (
 id bigint GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
 line_token text,
 slack_url text,
 telegram_url text,
 telegram_group_id text,
 is_notify bool DEFAULT false
);
```

table: s0727.servers

```
CREATE TABLE s0727.servers (
 id bigint GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
 name text,
 ip cidr DEFAULT '127.0.0.1',
 remark text
);
```

table: s0727.hardware_type

```
CREATE TABLE s0727.hardware_type (
 id bigint GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
 name text
);
```

table: s0727.hardware

```
CREATE TABLE s0727.hardware (
 id bigint GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
 type_id bigint REFERENCES s0727.hardware_type(id),
 server_id bigint REFERENCES s0727.servers(id),
 name text,
 hardware_info json,
 remark text
);
```

table: s0727.monitor_config

```
CREATE TABLE s0727.monitor_config (
 id bigint GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
 hardware_id bigint REFERENCES s0727.hardware(id),
 monitor_setting json,
 remark text
);
```

table: s0727.monitor_events

```
CREATE TABLE s0727.monitor_events (
 id bigint GENERATED ALWAYS AS IDENTITY,
 monitor_config_id bigint,
 event_remark text,
 created_at timestamptz NOT NULL DEFAULT CURRENT_TIMESTAMP
);
```

table: s0727.disk_data

```
CREATE TABLE s0727.disk_data (
 id bigint GENERATED ALWAYS AS IDENTITY,
 hardware_id bigint,
 total_used_size bigint not null,
 created_at timestamptz not NULL DEFAULT CURRENT_TIMESTAMP
);
```

table: s0727.notify

```
CREATE TABLE s0727.notify (
 id bigint GENERATED ALWAYS AS IDENTITY,
 event_id bigint,
 created_at timestamptz NOT NULL DEFAULT CURRENT_TIMESTAMP
);
```

```
table: s0727.canned_messages
CREATE TABLE s0727.canned_messages (
 id bigint GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
 type_id bigint REFERENCES s0727.hardware_type(id),
 message text
);
```

建立 PL/Python3 函式

讀取硬碟空間總共多大

```
CREATE OR REPLACE FUNCTION s0727.disk_total_size()
 RETURNS numeric
AS $$
  import os
  from decimal import Decimal
  return Decimal(os.popen("cd / | df --output=size | awk
'{if(NR>1) sum+=$1} END {print sum}'").read()) * 1024
$$ LANGUAGE plpython3u;
```

硬碟空間總共使用了多少

```
CREATE OR REPLACE FUNCTION s0727.disk_total_used_size()
 RETURNS numeric
AS $$
  import os
  from decimal import Decimal
  return Decimal(os.popen("cd / | df --output=used | awk
'{if(NR>1) sum+=$1} END {print sum}'").read()) * 1024
$$ LANGUAGE plpython3u;
```

讀取現在硬碟資訊並且轉成JSON格式

```
CREATE OR REPLACE FUNCTION s0727.disk_info()
 RETURNS text
AS $$
  import os
  import json
  # 執行 df 命令並獲取輸出
  output = os.popen("cd / | df").read().split('\n')
  output.pop()
  # 獲取列的標題
  keys = output[0].split()
  # 將列的標題轉換為英文
  keys = ["FileSystem", "1K_blocks", "Used", "Available", "UsePercentage", "MountedOn"]
  # 創建一個空列表來存儲結果
  result = []
  # 遍歷每一行
  for line in output[1:]:
   # 分割行並創建一個字典
   values = line.split()
   row_dict = dict(zip(keys, values))
   # 將字典添加到結果列表中
   result.append(row_dict)
   # 返回 JSON 格式的結果
  return json.dumps(result)
$$ LANGUAGE plpython3u;
```

將 JSON 內容作為 view 呈現

```
create view s0727.disk_info_view as
select *
  from json_to_recordset(s0727.disk_info()::json)
    as x(
        "FileSystem" text
        , "1K_blocks" text
        , "Used" text
        , "Available" text
        , "UsePercentage" text
        , "MountedOn" text
       );
```

可以查檢一下 `s0727.disk_info_view`

```
select sum("1K_blocks"::int) as total_size
     , sum("Used"::int) as total_used_size
     , sum("Available"::int) as total_free_size
  from s0727.disk_info_view;
```

total_size	total_used_size	total_free_size
796460316	148786647	647673664

Slack 訊息通知

```
CREATE OR REPLACE FUNCTION s0727.slack_notify(message
text, url text)
  RETURNS VOID
AS $$
  import os
  os.system(f"curl -X POST -H 'Content-type:
application/json' --data '{{\"text\":\"{message}\"}}'
{url}")
$$ LANGUAGE plpython3u;
```

LINE 訊息通知

```
CREATE OR REPLACE FUNCTION s0727.line_notify(message text,
token text)
  RETURNS VOID
AS $$
  import os
  os.system(f"curl -X POST -H 'Authorization: Bearer {token}'
-F 'message={message}'
https://notify-api.line.me/api/notify")
$$ LANGUAGE plpython3u;
```

Telegram 訊息通知

```
CREATE OR REPLACE FUNCTION s0727.telegram_notify(message
text, url text, telegram_group_id text)
  RETURNS VOID
AS $$
  import os
  os.system(f"curl -X POST -H 'Content-type:
application/json' --data
'{{\"chat_id\":\"{telegram_group_id}\",\"text\":\"{message}\"
}}' {url}")
$$ LANGUAGE plpython3u;
```

輸入資料

設定發送通知平台

```
insert into s0727.config
(line_token,slack_url,telegram_url,telegram_group_id,is_no
tify)
values ('LINE Token'
,'Slack URL'
,'Telegram URL'
,'Telegram group id'
,false);
```

被監控的硬體元件

```
insert into s0727.hardware_type (name) values ('disk');
```

被監控的Server

```
insert into s0727.servers (name) values ('海克力斯測試機');
```

建立硬體資訊清單

```
insert into s0727.hardware (type_id,name,hardware_info,server_id)
select id
     , '資料庫硬碟'
     , ('{"disk_type":"ssd","disk_size" : ' || s0727.disk_total_size() ||
'}')::json
     , (select id from s0727.servers where name = '海克力斯測試機')
  from s0727.hardware_type
 where name = 'disk';
```

監控設定

```
insert into s0727.monitor_config (hardware_id,monitor_setting)
select id
     , '{"percentage":60,"size":200}'::json
  from s0727.hardware
 where id = 1;
```

設定訊息格式

```
insert into s0727.canned_messages (type_id,message)
select id
     , '硬碟預警通知 總空間:{total_size} 剩餘空間:{free_size} 預警空間值:{size}'
  from s0727.hardware_type
 where name = 'disk';
```

建立

PL/Pgsql 函式、Trigger

硬碟剩餘空間總共多少

```
CREATE OR REPLACE FUNCTION s0727.disk_total_free_size()
 RETURNS numeric
AS $$
BEGIN
```

```
RETURN (SELECT s0727.disk_total_size() -
s0727.disk_total_used_size());
END;
$$ LANGUAGE plpgsql;
```

讀取硬碟剩餘空間百分比

```
CREATE OR REPLACE FUNCTION s0727.disk_free_percentage()
 RETURNS double precision
AS $$
BEGIN
RETURN (SELECT s0727.disk_total_free_size()::float /
s0727.disk_total_size()::float * 100);
END;
$$ LANGUAGE plpgsql;
```

建立寫入硬碟數值的 function

```
CREATE OR REPLACE FUNCTION s0727.add_disk_data()
 RETURNS void
AS $$
BEGIN
   BEGIN
       INSERT INTO s0727.disk_data (hardware_id, total_used_size)
       SELECT id
            , s0727.disk_total_used_size()
         FROM s0727.hardware h;
   EXCEPTION WHEN others THEN
       -- 在這裡處理錯誤，例如，您可以選擇記錄錯誤信息
       -- RAISE NOTICE 'An error occurred: %', SQLERRM;
       RETURN;
   END;
END;
$$ LANGUAGE plpgsql;
```

取得還未發送的 events

```
CREATE OR REPLACE FUNCTION s0727.get_unnotified_events()
RETURNS TABLE(id bigint, monitor_config_id bigint,
event_remark text, created_at timestamptz)
AS $$
BEGIN
 RETURN QUERY
 SELECT e.id
      , e.monitor_config_id
      , e.event_remark
      , e.created_at
 FROM s0727.monitor_events e
 LEFT JOIN s0727.notify n
   ON e.id = n.event_id
WHERE n.event_id IS NULL;
END;
$$ LANGUAGE plpgsql;
```

發訊息函式

```
CREATE OR REPLACE FUNCTION s0727.heracles_fn_notify()
 RETURNS void
AS $$
DECLARE
  local_config_row s0727.config%ROWTYPE;
  local_message_text text;
  local_total_size numeric;
  local_used_size numeric;
  local_free_size numeric; --- 剩於空間
  local_free_size_human_read float; --- 剩於空間，轉型
  local_free_percentage float; --- 剩於空間百分比
  local_free_percentage_human_read float; --- 剩於空間百分比，轉型
  local_monitor_setting_json json;
  local_size numeric;
  local_percentage float; --- 預警空間百分比
  local_percentage_human_read float; --- 預警空間百分比，轉型
BEGIN
  SELECT * INTO local_config_row
    FROM s0727.config
   WHERE is_notify is true;
IF local_config_row.is_notify THEN
  SELECT "message" INTO local_message_text
    FROM s0727.canned_messages
   WHERE type_id = 1;
  local_total_size := s0727.disk_total_size();
  local_used_size := s0727.disk_total_used_size();
  local_free_size := s0727.disk_total_free_size();
  local_free_size_human_read = round(local_free_size::double precision);
  local_free_percentage := s0727.disk_free_percentage();
  local_free_percentage_human_read = round(local_free_percentage::double precision);
  SELECT monitor_setting INTO local_monitor_setting_json
    FROM s0727.monitor_config
   WHERE hardware_id = 1;
  local_size := (local_monitor_setting_json->>'size')::numeric;
  local_percentage := (local_monitor_setting_json->>'percentage')::double precision;
```

```
  local_percentage_human_read = round(local_percentage);

  local_message_text := REPLACE(local_message_text, '{total_size}',
  pg_size_pretty(local_total_size));

  local_message_text := REPLACE(local_message_text, '{used_size}',
  pg_size_pretty(local_used_size));

  local_message_text := REPLACE(local_message_text, '{free_size}',
  pg_size_pretty(local_free_size));

  local_message_text := REPLACE(local_message_text, '{free_percentage}',
  local_free_percentage_human_read::text || '%');

  local_message_text := REPLACE(local_message_text, '{size}', pg_size_pretty(local_size));

  local_message_text := REPLACE(local_message_text, '{percentage}',
  local_percentage_human_read::text || '%');

  IF local_config_row.slack_url IS NOT NULL THEN
    PERFORM s0727.slack_notify(local_message_text, local_config_row.slack_url);
  END IF;
  IF local_config_row.line_token IS NOT NULL THEN
    PERFORM s0727.line_notify(local_message_text, local_config_row.line_token);
  END IF;
  IF local_config_row.telegram_url IS NOT NULL THEN
    PERFORM s0727.telegram_notify(local_message_text,local_config_row.telegram_url,local_config_row.telegram_group_id);
  END IF;
END IF;
END;
$$ LANGUAGE plpgsql;
```

未發送的發送並且寫到 heracles_fn_notify

```
CREATE OR REPLACE FUNCTION s0727.send_and_record_notify()
RETURNS VOID
AS $$
DECLARE
```

```
 event_row RECORD;
BEGIN
 FOR event_row IN SELECT *
                  FROM s0727.get_unnotified_events()
 LOOP
   PERFORM s0727.heracles_fn_notify();
   INSERT INTO s0727.notify (event_id)
   VALUES (event_row.id);
 END LOOP;
END;
$$ LANGUAGE plpgsql;
```

Trigger：建立檢查硬碟使用情況之 function，接著設定 trigger 呼叫 function 檢查硬碟使用情況

```
CREATE OR REPLACE FUNCTION s0727.check_disk_usage()
RETURNS TRIGGER AS $$
DECLARE
monitor_setting_json json;
total_used_size bigint;
size bigint;
percentage bigint;
hardware_type_name text;
monitor_config_id bigint;
BEGIN
-- Get the total used size from the new disk data
total_used_size := NEW.total_used_size;
-- Get the monitor setting
SELECT monitor_setting INTO monitor_setting_json
 FROM s0727.monitor_config
WHERE hardware_id = NEW.hardware_id;

SELECT id INTO monitor_config_id
 FROM s0727.monitor_config
WHERE hardware_id = NEW.hardware_id;

size := (monitor_setting_json->>'size')::bigint;
percentage := (monitor_setting_json->>'percentage')::bigint;

-- Check if the total used size exceeds the size or percentage in the monitor setting
IF total_used_size >= size OR s0727.disk_free_percentage() >= percentage THEN
-- Get the hardware type name
SELECT hardware_type.name INTO hardware_type_name
  FROM s0727.monitor_config
  JOIN s0727.hardware
    ON monitor_config.hardware_id = hardware.id
```

```
  JOIN s0727.hardware_type ON hardware_type.id = hardware.type_id
 WHERE hardware.id = NEW.hardware_id;
 -- Insert a new event into the monitor_events table
INSERT INTO s0727.monitor_events (monitor_config_id, event_remark)
VALUES (monitor_config_id, hardware_type_name);
END IF;

RETURN NEW;
EXCEPTION
WHEN OTHERS THEN
  RAISE NOTICE '錯誤訊息：%', SQLERRM;
  RETURN NULL;
END;
$$ LANGUAGE plpgsql;
```

設定 Trigger

```
CREATE TRIGGER disk_data_inserted
AFTER INSERT ON s0727.disk_data
FOR EACH ROW
EXECUTE FUNCTION s0727.check_disk_usage();
```

建立 pg_cron 排程

```
select
cron.schedule('*/1 * * * *', $$select
s0727.add_disk_data()$$);

select
cron.schedule('10 15 * * *', $$select
s0727.send_and_record_notify()$$);
```

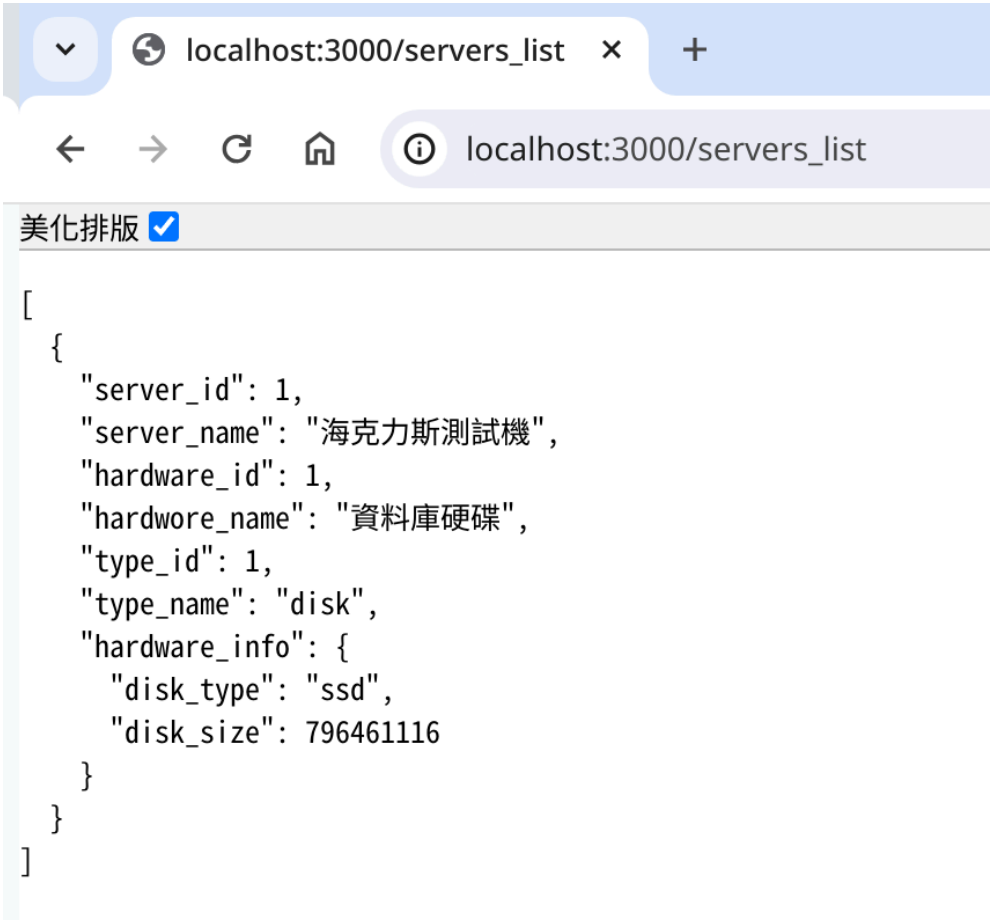
透過 TimescaleDB
建立 Hypertable

請參閱
https://www.youtube.com/watch?v=lz2PPBu_1EI

PostgREST 建立 Web API

mywebapi.conf

```
db-uri =
"postgres://heracles_viewer:heracles_viewer_test@localhost:5432
/heracles"
db-schemas = "s0727"
db-anon-role = "heracles_viewer"
```




```
[
  {
    "server_id": 1,
    "server_name": "海克力斯測試機",
    "hardware_id": 1,
    "hardwore_name": "資料庫硬碟",
    "type_id": 1,
    "type_name": "disk",
    "hardware_info": {
      "disk_type": "ssd",
      "disk_size": 796461116
    }
  }
]
```

localhost:3000/view_disk_data

美化排版


```
[{"hardware_id":1,"hardware_name":"資料庫硬碟","hardware_info":{"disk_type":"ssd","disk_size" : 815575363584},"total_used_size":152139320320,"record_time":"2024-07-27T15:45:13.848889+08:00"},
 {"hardware_id":1,"hardware_name":"資料庫硬碟","hardware_info":{"disk_type":"ssd","disk_size" : 815575363584},"total_used_size":152129612800,"record_time":"2024-07-27T15:48:04.827208+08:00"},
 {"hardware_id":1,"hardware_name":"資料庫硬碟","hardware_info":{"disk_type":"ssd","disk_size" : 815575363584},"total_used_size":152108411904,"record_time":"2024-07-27T16:01:00.019539+08:00"},
 {"hardware_id":1,"hardware_name":"資料庫硬碟","hardware_info":{"disk_type":"ssd","disk_size" : 815575363584},"total_used_size":152099060736,"record_time":"2024-07-27T16:08:06.278246+08:00"},
 {"hardware_id":1,"hardware_name":"資料庫硬碟","hardware_info":{"disk_type":"ssd","disk_size" : 815575363584},"total_used_size":152091700224,"record_time":"2024-07-27T16:12:00.014601+08:00"},
 {"hardware_id":1,"hardware_name":"資料庫硬碟","hardware_info":{"disk_type":"ssd","disk_size" : 815575363584},"total_used_size":152091974656,"record_time":"2024-07-27T16:13:00.017347+08:00"},
 {"hardware_id":1,"hardware_name":"資料庫硬碟","hardware_info":{"disk_type":"ssd","disk_size" : 815575363584},"total_used_size":152081099776,"record_time":"2024-07-27T16:14:00.015021+08:00"}]
```


localhost:3000/disk_info_view

美化排版


```
[{"FileSystem":"devtmpfs","1K_blocks":"4096","Used":"0","Available":"4096","UsePercentage":"0%","MountedOn":"/dev"},
 {"FileSystem":"tmpfs","1K_blocks":"7948456","Used":"104904","Available":"7843552","UsePercentage":"2%","MountedOn":"/dev/shm"},
 {"FileSystem":"tmpfs","1K_blocks":"3179384","Used":"18668","Available":"3160716","UsePercentage":"1%","MountedOn":"/run"},
 {"FileSystem":"efivarfs","1K_blocks":"268","Used":"187","Available":"76","UsePercentage":"72%","MountedOn":"/sys/firmware/efi/efivars"},
 {"FileSystem":"/dev/mapper/cs_sakimio-root","1K_blocks":"282977280","Used":"29412968","Available":"253564312","UsePercentage":"11%","MountedOn":"/"},
 {"FileSystem":"/dev/nvme0n1p2","1K_blocks":"2031616","Used":"1249172","Available":"782444","UsePercentage":"62%","MountedOn":"/boot"},
 {"FileSystem":"/dev/mapper/cs_sakimio-home","1K_blocks":"497683016","Used":"118032132","Available":"379650884","UsePercentage":"24%","MountedOn":"/home"},
 {"FileSystem":"/dev/nvme0n1p1","1K_blocks":"1046512","Used":"7652","Available":"1038860","UsePercentage":"1%","MountedOn":"/boot/efi"},
 {"FileSystem":"tmpfs","1K_blocks":"1589688","Used":"148","Available":"1589540","UsePercentage":"1%","MountedOn":"/run/user/1000"}]
```

建立一個使用者 記得給密碼 postgrest需要用到

```
CREATE USER heracles_viewer WITH PASSWORD
'heracles_viewer_test';
```

我這邊先給他管理員權限

```
ALTER ROLE heracles_viewer SUPERUSER;
```

記得把連線到資料庫的權線給 heracles_viewer 使用者

```
grant connect on database coscup2024 to heracles_viewer;
```

並把 s0727 schema使用權 給 heracles_viewer 使用者

```
grant usage on schema s0727 to heracles_viewer;
```

```
---------- create viewer ----------
建立伺服器清單的view
create view s0727.servers_list as
select s.id as server_id
    , s."name" as server_name
    , h.id as hardware_id
    , h."name" as hardwore_name
    , ht.id as type_id
    , ht."name" as type_name
    , h.hardware_info
 from s0727.servers s
 join s0727.hardware h
   on s.id  = h.server_id
 join s0727.hardware_type ht
   on h.type_id = ht.id;
```

建立硬碟資料 view

```
create view s0727.view_disk_data as
select h.id as hardware_id
     , h."name" as hardware_name
     , h.hardware_info
     , dd.total_used_size
     , dd.created_at as record_time
 from s0727.disk_data dd
 join s0727.hardware h
   on h.id  = dd.hardware_id;
```

建立硬碟分割磁區 view

```
create view s0727.disk_info_view as
select *
  from json_to_recordset(s0727.disk_info()::json)
 as x("FileSystem" text, "1K_blocks" text, "Used" text, "Available"
text, "UsePercentage" text, "MountedOn" text);
```

謝謝大家