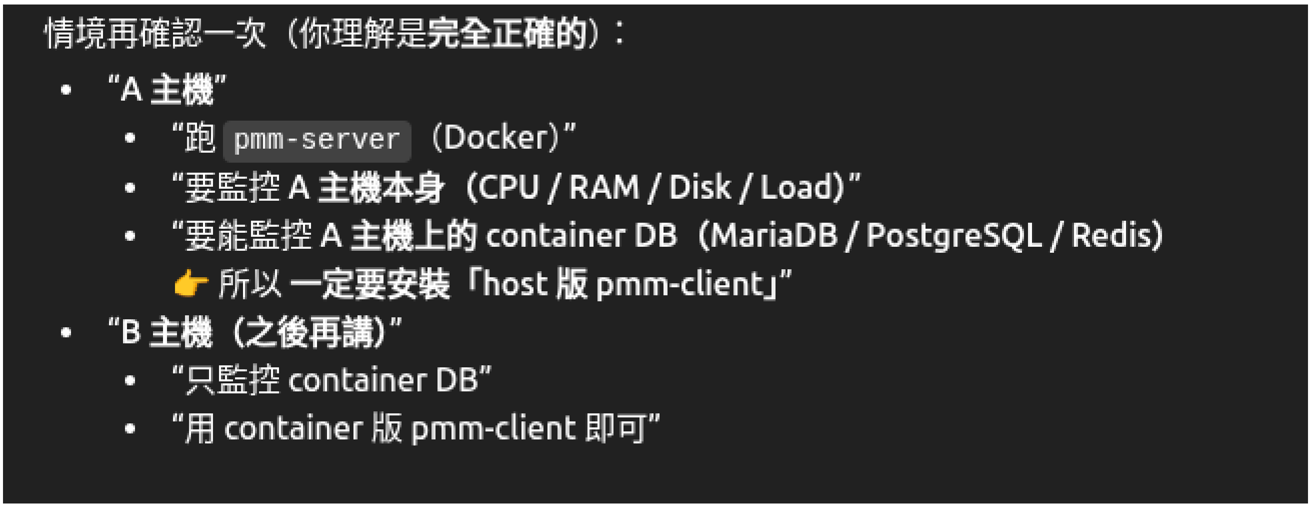
真實使用情境



情境再確認一次（你理解是**完全正確的**）：

- **“A 主機”**
  - “跑 `pmm-server` （Docker）”
  - **“要監控 A 主機本身（CPU / RAM / Disk / Load）”**
  - **“要能監控 A 主機上的 container DB（MariaDB / PostgreSQL / Redis)**
    👉 所以 **一定要安裝「host 版 pmm-client」**”
- **“B 主機（之後再講）”**
  - “只監控 container DB”
  - “用 container 版 pmm-client 即可”

```
[A Host]
 ├─ pmm-agent            ← 裝在 host（只裝一次）
 │   ├─ node_exporter    ← 主機 CPU / RAM / Disk
 │   ├─ mysqld_exporter / postgres_exporter / redis_exporter
 │   └─ 負責把資料送到 PMM Server
 │
 ├─ MariaDB container
 ├─ Redis container
 └─ PostgreSQL container

[PMM Server] (Docker)
 └─ 收 metrics + UI
```

A主機的問題 (234)

1. 安裝 pmm-agent

```
sduo yum install -y https://repo.percona.com/yum/percona-release-latest.noarch.rpm
sduo percona-release enable pmm3-client release
sduo yum install -y pmm-client
```

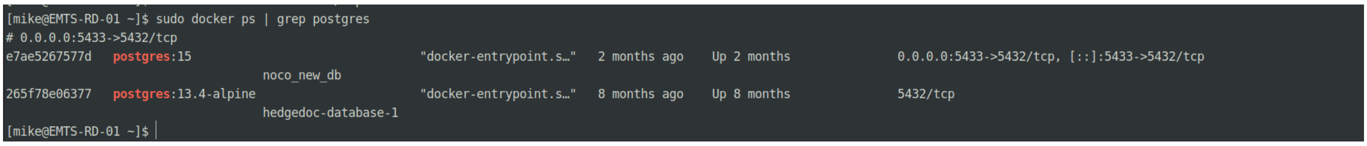
2. 因為 pmm-agent 的 7777 被佔用所以要去修改然後重新設定

```
sudo vi /usr/local/percona/pmm/config/pmm-agent.yaml

改port

sudo pmm-agent setup \
  --config-file=/usr/local/percona/pmm/config/pmm-agent.yaml \
  --server-address=192.168.199.234:18443 \
  --server-username=admin \
  --server-password='Aa123456' \
  --server-insecure-tls \
  --force

sudo systemctl restart pmm-agent
```

3. 因為改 Port 所以 pmm-admin 指令需要修改增加 --pmm-agent-listen-port

```
sudo pmm-admin add mysql   --server-url=https://admin:Aa123456@192.168.199.234:18443   --server-insecure-tls   --pmm-agent-listen-port=17777   --username=資料庫帳號   --password='資料庫密碼'   --host=192.168.199.234   --port=3306   --service-name=mariadb-234   --query-source=slowlog
```

4. 同一台主機 不同容器的加入方式



```
[mike@EMTS-RD-01 ~]$ sudo docker ps | grep postgres
# 0.0.0.0:5433->5432/tcp
e7ae5267577d   postgres:15                       "docker-entrypoint.s…"   2 months ago   Up 2 months          0.0.0.0:5433->5432/tcp, [::]:5433->5432/tcp
                         noco_new_db
265f78e06377   postgres:13.4-alpine              "docker-entrypoint.s…"   8 months ago   Up 8 months          5432/tcp
                         hedgedoc-database-1
[mike@EMTS-RD-01 ~]$
```

5. 使用容器的 IP 來加入節點

```
sudo docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' noco_new_db

sudo docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' hedgedoc-database-1

sudo docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' mysql
```

6. 建立資料庫帳號

```
docker exec -it noco_new_db psql -U noco_appsmith -d app_data

CREATE USER pmm WITH PASSWORD 'StrongPMMpass!';

GRANT pg_monitor TO pmm;
```

7. noco-db 加入監控

```
sudo pmm-admin add postgresql \
  --server-url=https://admin:Aa123456@192.168.199.234:18443 \
  --server-insecure-tls \
  --pmm-agent-listen-port=17777 \
  --username=pmm \
  --password='StrongPMMpass!' \
  --host=172.17.0.6 \
  --port=5432 \
  --service-name=postgres-noco
```

8. hedgedoc 加入監控

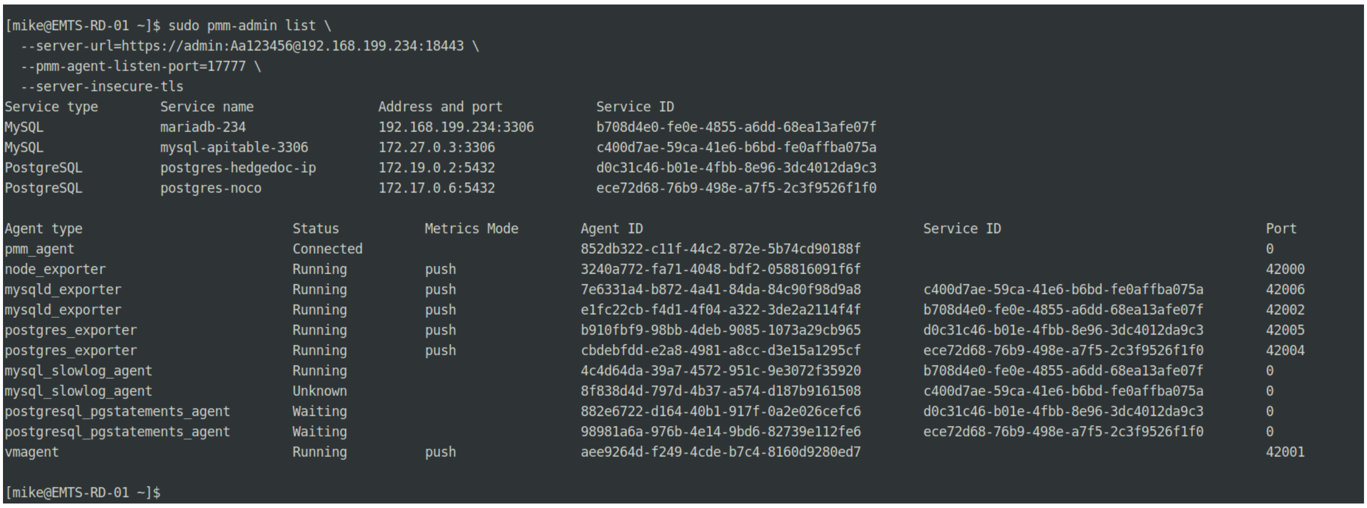
```
sudo pmm-admin add postgresql \
  --server-url=https://admin:Aa123456@192.168.199.234:18443 \
  --server-insecure-tls \
  --pmm-agent-listen-port=17777 \
  --username=pmm \
  --password='StrongPMMpass!' \
  --host=172.19.0.2 \
  --port=5432 \
  --service-name=postgres-hedgedoc-ip
```

9. mysql-apitable 加入監控

```
sudo pmm-admin add mysql \
  --server-url=https://admin:Aa123456@192.168.199.234:18443 \
  --server-insecure-tls \
  --pmm-agent-listen-port=17777 \
  --username=pmm \
  --password='StrongPMMpass!' \
  --host=172.27.0.3 \
  --port=3306 \
  --service-name=mysql-apitable-3306
```

10. 驗證

```
sudo pmm-admin list \
  --server-url=https://admin:Aa123456@192.168.199.234:18443 \
  --pmm-agent-listen-port=17777 \
  --server-insecure-tls
```



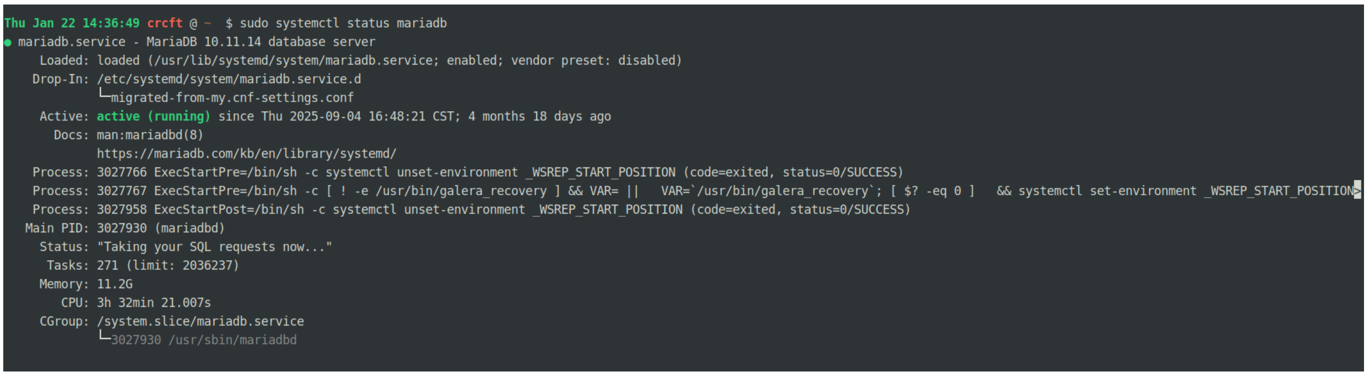
```
[mike@EMTS-RD-01 ~]$ sudo pmm-admin list \
  --server-url=https://admin:Aa123456@192.168.199.234:18443 \
  --pmm-agent-listen-port=17777 \
  --server-insecure-tls
Service type        Service name                Address and port            Service ID
MySQL               mariadb-234                 192.168.199.234:3306        b708d4e0-fe0e-4855-a6dd-68ea13afe07f
MySQL               mysql-apitable-3306         172.27.0.3:3306             c400d7ae-59ca-41e6-b6bd-fe0affba075a
PostgreSQL          postgres-hedgedoc-ip        172.19.0.2:5432             d0c31c46-b01e-4fbb-8e96-3dc4012da9c3
PostgreSQL          postgres-noco               172.17.0.6:5432             ece72d68-76b9-498e-a7f5-2c3f9526f1f0

Agent type                      Status          Metrics Mode        Agent ID                                    Service ID                                  Port
pmm_agent                       Connected                           852db322-c11f-44c2-872e-5b74cd90188f                                                    0
node_exporter                   Running         push                3240a772-fa71-4048-bdf2-058816091f6f                                                    42000
mysqld_exporter                 Running         push                7e6331a4-b872-4a41-84da-84c90f98d9a8        c400d7ae-59ca-41e6-b6bd-fe0affba075a        42006
mysqld_exporter                 Running         push                e1fc22cb-f4d1-4f04-a322-3de2a2114f4f        b708d4e0-fe0e-4855-a6dd-68ea13afe07f        42002
postgres_exporter               Running         push                b910fbf9-98bb-4deb-9085-1073a29cb965        d0c31c46-b01e-4fbb-8e96-3dc4012da9c3        42005
postgres_exporter               Running         push                cbdebfdd-e2a8-4981-a8cc-d3e15a1295cf        ece72d68-76b9-498e-a7f5-2c3f9526f1f0        42004
mysql_slowlog_agent             Running                             4c4d64da-39a7-4572-951c-9e3072f35920        b708d4e0-fe0e-4855-a6dd-68ea13afe07f        0
mysql_slowlog_agent             Unknown                             8f838d4d-797d-4b37-a574-d187b9161508        c400d7ae-59ca-41e6-b6bd-fe0affba075a        0
postgresql_pgstatements_agent   Waiting                             882e6722-d164-40b1-917f-0a2e026cefc6        d0c31c46-b01e-4fbb-8e96-3dc4012da9c3        0
postgresql_pgstatements_agent   Waiting                             98981a6a-976b-4e14-9bd6-82739e112fe6        ece72d68-76b9-498e-a7f5-2c3f9526f1f0        0
vmagent                         Running         push                aee9264d-f249-4cde-b7c4-8160d9280ed7                                                    42001

[mike@EMTS-RD-01 ~]$
```

B主機的問題 (134)

用容器安裝 pmm-client:3 並且註冊好遠端的 pmm-server (腳本:pmm-client.sh)

進入容器設相關資料庫服務

本地資料庫的加入方式如下



```
Thu Jan 22 14:36:49 crcft @ ~  $ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.11.14 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
    Drop-In: /etc/systemd/system/mariadb.service.d
             └─migrated-from-my.cnf-settings.conf
     Active: active (running) since Thu 2025-09-04 16:48:21 CST; 4 months 18 days ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 3027766 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
    Process: 3027767 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= ||   VAR=`/usr/bin/galera_recovery`; [ $? -eq 0 ]   && systemctl set-environment _WSREP_START_POSITION>
    Process: 3027958 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
   Main PID: 3027930 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 271 (limit: 2036237)
     Memory: 11.2G
        CPU: 3h 32min 21.007s
     CGroup: /system.slice/mariadb.service
             └─3027930 /usr/sbin/mariadbd
```

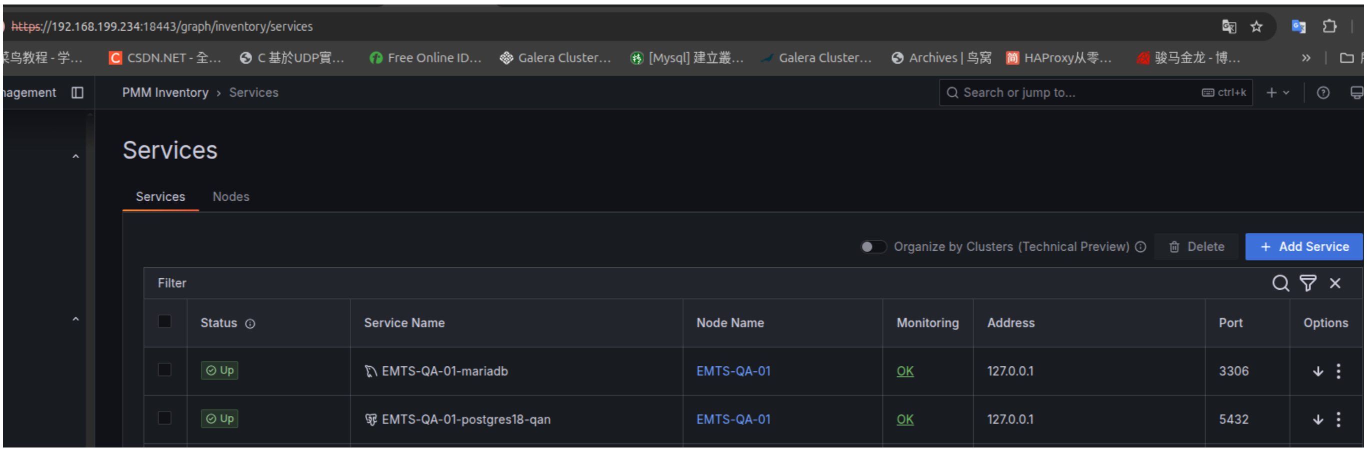
```
sudo docker exec -it pmm-client bash -lc \
"pmm-admin add mysql \
  --username=資料庫帳號 \
  --password='資料庫密碼' \
  --host=127.0.0.1 \
  --port=3306 \
  --service-name='EMTS-QA-01-mariadb' \
  --query-source=perfschema"
```

容器資料庫的加入方式

簡單弄個PG資料庫

```
services:
  postgres18:
    image: postgres:18
    container_name: postgres18
    restart: always
    environment:
      POSTGRES_USER: myuser
      POSTGRES_DB: mydb
      POSTGRES_PASSWORD: mypassword
    ports:
      - "5432:5432"
    volumes:
      - pg18_data:/var/lib/postgresql
    command:
        # 開 pg_stat_statements
      - "postgres"
      - "-c"
      - "shared_preload_libraries=pg_stat_statements"
      - "-c"
      - "pg_stat_statements.track=all"
      - "-c"
      - "track_activity_query_size=2048"

volumes:
  pg18_data:
```

```
# 建立 extension（每個 DB 要建一次）
sudo docker exec -it postgres18 psql -U myuser -d mydb -c \
"CREATE EXTENSION IF NOT EXISTS pg_stat_statements;"

# 把 PG18 加進 PMM
sudo docker exec -it pmm-client pmm-admin add postgresql \
  --service-name=EMTS-QA-01-postgres18-qan \
  --host=127.0.0.1 \
  --port=5432 \
  --username=資料庫帳號 \
  --password='資料庫密碼' \
  --database=mydb \
  --query-source=pgstatements
```



□	Status	Service Name	Node Name	Monitoring	Address	Port	Options
□	Up	EMTS-QA-01-mariadb	EMTS-QA-01	OK	127.0.0.1	3306	↓ ⋮
□	Up	EMTS-QA-01-postgres18-qan	EMTS-QA-01	OK	127.0.0.1	5432	↓ ⋮

MSSQL 的設定

sudo docker network create pmm-net 2>/dev/null || true

1. 先建立 MSSQL Server 容器

```
sudo docker run -d \
  --name mssql \
  --restart unless-stopped \
  --network pmm-net \
  -e 'ACCEPT_EULA=Y' \
  -e 'MSSQL_SA_PASSWORD=YourStr0ng!Passw0rd' \
  -e "MSSQL_PID=Express" \
  -p 2433:1433 \
  mcr.microsoft.com/mssql/server:2022-latest
```

2. 建立 PMM 專用帳號

```
sudo docker exec -i mssql bash -lc 'set +H
/opt/mssql-tools18/bin/sqlcmd \
  -S localhost -U sa -P "YourStr0ng!Passw0rd" \
  -C \
  -Q "
USE master;
IF EXISTS (SELECT 1 FROM sys.server_principals WHERE name = '\''pmm'\'')
BEGIN
  DROP LOGIN pmm;
END;

CREATE LOGIN pmm WITH PASSWORD = '\''X9!aZ7#Qp2@Mssql'\'', CHECK_POLICY =
ON, CHECK_EXPIRATION = OFF;
CREATE USER pmm FOR LOGIN pmm;
ALTER SERVER ROLE sysadmin ADD MEMBER pmm;
"
'
```

3. 啟動 MSSQL Exporter（Prometheus）

建立 env 檔

```
sudo mkdir -p /data/mssql-exporter

sudo tee /data/mssql-exporter/env >/dev/null <<'EOF'
SERVER=mssql
PORT=1433
USERNAME=pmm
PASSWORD=X9!aZ7#Qp2@Mssql
ENCRYPT=true
TRUST_SERVER_CERTIFICATE=true
EXPOSE=4000
```

```
EOF

sudo chmod 600 /data/mssql-exporter/env
```

啟動 awaragi/mssql-exporter container

```
sudo docker rm -f mssql-exporter 2>/dev/null || true
sudo docker pull awaragi/prometheus-mssql-exporter:latest

sudo docker run -d \
  --name mssql-exporter \
  --restart unless-stopped \
  --network pmm-net \
  --env-file /data/mssql-exporter/env \
  -p 4000:4000 \
  awaragi/prometheus-mssql-exporter:latest
```

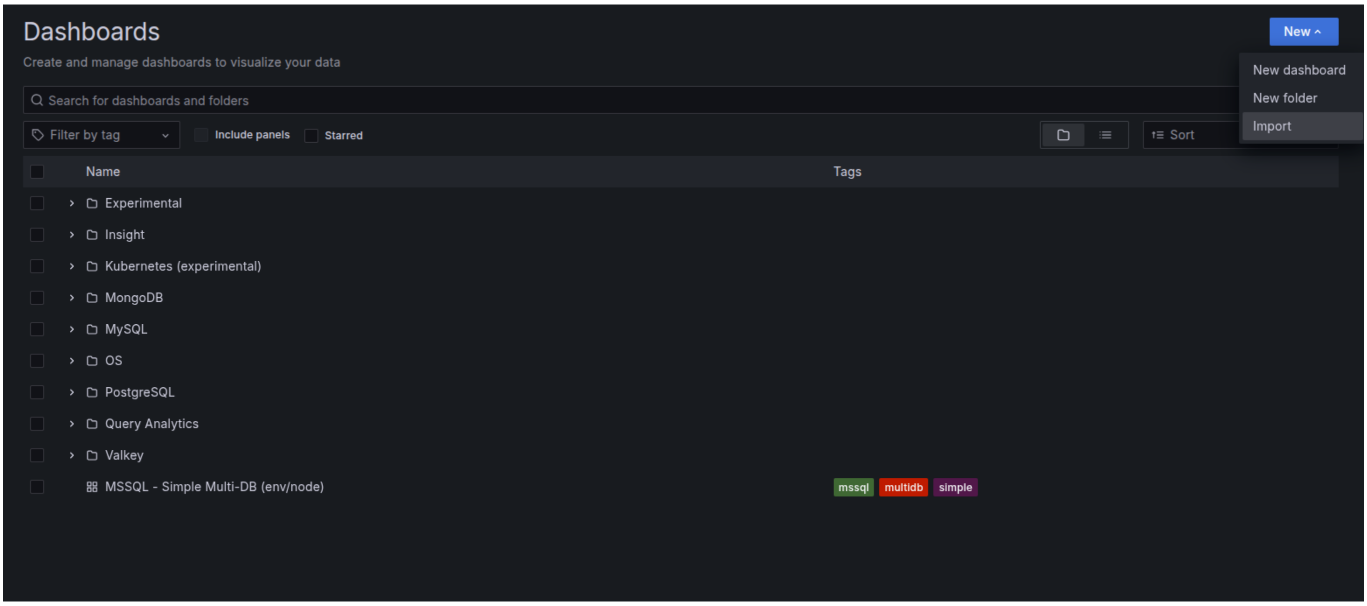
4. 加進 PMM（External exporter）

```
sudo docker exec -it pmm-client pmm-admin remove external "EMTS-QA-01-mssql" 2>/dev/null || true

NODE_ID=$(sudo docker exec -it pmm-client pmm-admin status \
  | awk -F': ' '/Node ID/ {print $2}' | tr -d '\r')

echo "Node ID = $NODE_ID"


sudo docker exec -it pmm-client pmm-admin add external \
  --service-name="EMTS-QA-01-mssql" \
  --service-node-id="$NODE_ID" \
  --scheme="http" \
  --metrics-path="/metrics" \
  --listen-port=4000 \
  --environment="qa" \
  --custom-labels="dbtype=mssql,env=qa,node=EMTS-QA-01" \
  --metrics-mode="pull"
```

5. 導入到UI上

因為UI並沒有支援 MSSQL 但是 我們的 exporter 有正常執行 所以他會知道

接下來把 mssql.json Import 進去就好了

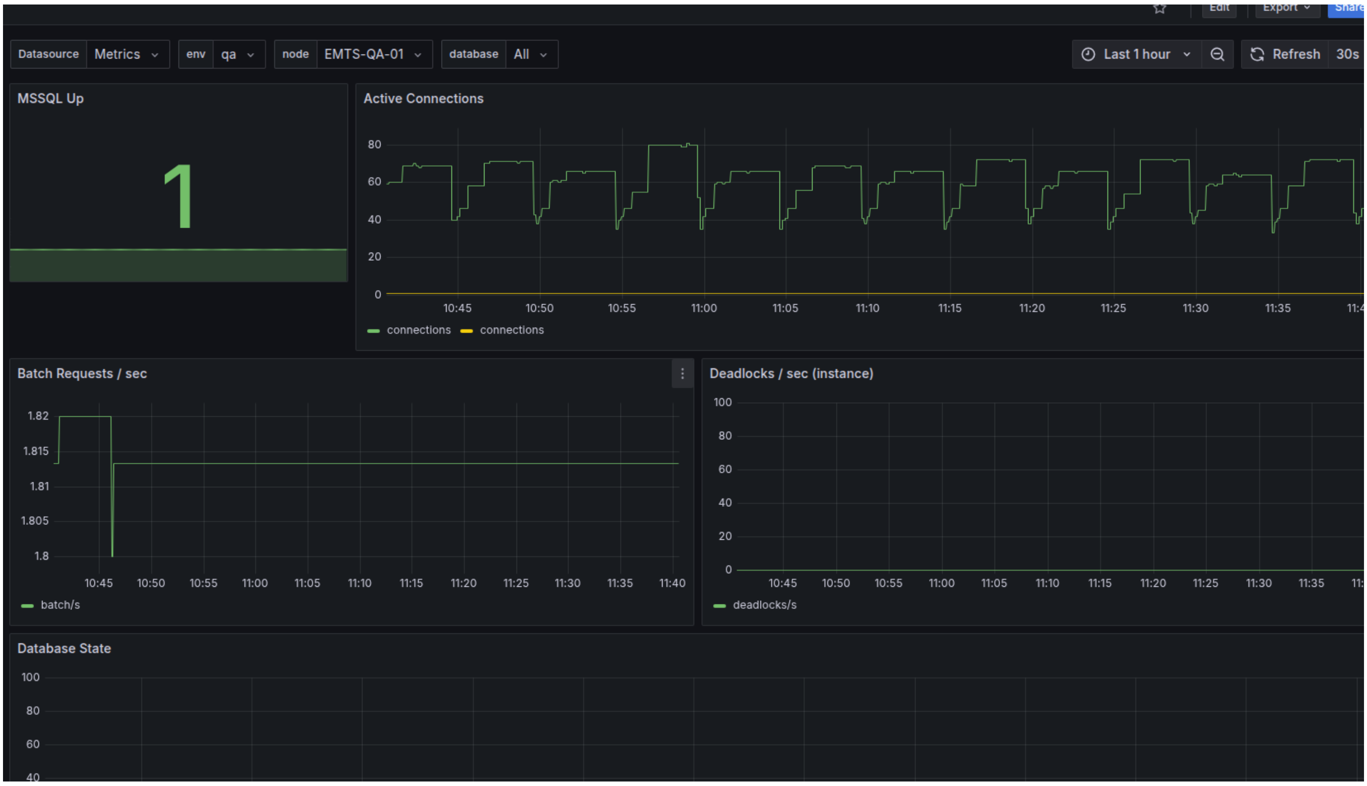


Dashboards

Create and manage dashboards to visualize your data

Name	Tags
Experimental	
Insight	
Kubernetes (experimental)	
MongoDB	
MySQL	
OS	
PostgreSQL	
Query Analytics	
Valkey	
MSSQL - Simple Multi-DB (env/node)	mssql multidb simple

6. 完成



Datasource Metrics · env qa · node EMTS-QA-01 · database All · Last 1 hour · Refresh 30s