### 【Docker】自動分配網路用盡

failed to create network pub-sub\_pubsub: Error response from daemon: could not find an available, non-overlapping IPv4 address pool among the defaults to assign to the network

### □ 這個錯誤的意思是:

44 Docker 沒有可用的 IP 網段 來建立新的 network。

Docker 每次起新的 network, 會自動從預設的 IP range (例如 172.18.0.0/16) 分配一段子網(subnet)來給你的 docker-compose。

但因為你本機已經有太多 Docker network(或之前的 container network 沒清乾淨),導致 IP address range 已經用光了。

#### 解法,詳細往下看

### # 查詢目前所有container網段

docker network inspect \$(docker network Is -q) --format '{{.Name}}: {{(index .IPAM.Config 0).Subnet}}'

# 刪除無用network

docker network prune

#解法,指定網段

networks:

ngtest:

driver: bridge

pam:

- subnet: 172.20.0.0/24

### □ 解法 1:刪掉無用的 Docker Network

### 先看一下目前的 network:

docker network Is

### 會看到一堆像這樣:

NETWORK ID	NAME	DRIVER	SCOPE
1a2b3c4d5e	bridge	bridge	local
6f7g8h9i0j	pub-sub_pubsub	bridge	local

### 然後手動刪掉沒用的 network,例如:

docker network rm network\_name

### 例如:

docker network rm pub-sub\_pubsub

### 要一口氣刪很多:

docker network prune

# □解法 2:docker-compose.yml 手動指定 IP 範圍

如果你想避免這個問題,以後可以自己在 docker-compose.yml 裡面設定固定的子網段,例如:

networks:
pubsub-net:
driver: bridge
ipam:
config:

- subnet: 172.30.0.0/16

### 然後每個 service 指定 network:

services:

pubsub-emulator:

image: google/cloud-sdk:latest

networks: - nubsub-net

這樣 Docker 就不會自己亂分配 IP,會用你指定的 172.30.0.0/16 ,可以大大降低出錯的機會。

### □最快速推薦操作

直接下這兩行,清除掉無用網路 → 再重開 docker-compose:

docker network prune -f docker-compose up

### □小提醒

問題情境	說明	
通常出現在 Docker Desktop 很久沒清	每次起 network,Docker 自動分配,結果越堆越多	
docker network ls 看太多 bridge	可以確認哪些是 zombie network	
docker-compose down 不一定會清 network	要自己下 prune 或 network rm	

### 好問題!

你問的這個其實就是:

44 如何查目前 Docker 所有 network 的網段(subnet),確保自己新建的網段不會衝突?

這個是專業級的 docker networking 操作了,我來直接教你最正確、完整又快速的方法 □□□

□ 查詢目前所有 Docker network 網段 的方法

### 直接打:

docker network inspect \$(docker network Is -q) --format '{{.Name}}: {{(index .IPAM.Config 0).Subnet}}'

### 這條指令的意思是:

- docker network Is -q :列出所有 network 的 ID
- docker network inspect : 逐個去查 network 詳細資訊
- --format : 只輸出你要看的部分 (名字 + 子網範圍)

### □ 出來結果像這樣:

bridge: 172.17.0.0/16

host: none:

myproject\_default: 172.21.0.0/16 another-network: 172.22.0.0/16

如果某些 network 是空的(像 host 、 none ),那是正常的,因為它們沒有自己管理 IP。

### □ 查完之後怎麼重新分配?

- 看到目前用掉哪些子網了(例如 172.17.0.0/16、172.21.0.0/16)
- 選一個完全沒重疊的新網段來建你的 network,例如:
  - 0 172.28.0.0/24
  - 172.30.0.0/24
  - 0 192.168.100.0/24
  - 0 10.20.0.0/24
- 然後在你的 docker-compose.yml 裡設定你自己的網段!

(一般建議用 172.x.x.x / 192.168.x.x / 10.x.x.x 這些 private 網段)

## □進階版(更詳細看)

如果你想要看得更細,例如看到每個 container 的 IP,可以打:

docker network inspect network-name

### 例:

docker network inspect bridge docker network inspect pubsub-net

### 它會印出一個超大 JSON,裡面會列出:

- 網段 subnet
- gateway
- 已連接的 container 名稱和 IP 地址

### □小結論

你想做什麼	指令
看所有 network 網段	docker network inspect \$(docker network Is -q)format '{{.Name}}: {{(index .IPAM.Config 0).Subnet}}'
詳細看某個 network	docker network inspect network-name

⑤修訂版本 #2 ★由 treeman 建立於 28 �|�� 2025 13:23:10 ✔由 treeman 更新於 28 �|�� 2025 13:39:39