

Agent Workflow 治理平台 · 部署架構

Docker Container 配置架構

compose 專案 workflow-hub：六個服務的角色、網路、儲存、秘密與互動關係



容器



SQL Server 2025



Web



Worker



MCP Server



磁碟區

來源：範例/12_部署設定與成果驗收/docker (docker-compose.yml 、 Dockerfile 、 init-db.sh)

實測環境：Docker Desktop · Windows 11 · Linux 容器 | 2026-09-29

簡報大綱

由全景到細節：先看整體，再逐一拆解每個容器，最後看它們如何協同運作



01

全景總覽

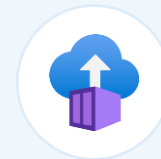
六個服務一覽、整體架構圖



02

映像與啟動

多階段建置、一個映像四種用途、啟動相依



03

各服務深入

資料庫、初始化工作、web、worker、MCP



04

互動流程

手動試跑、排程執行兩條序列



05

網路·儲存·秘密

bridge 網路、具名磁碟區、.env 與最小權限



06

維運與共存

健康檢查、主機連接埠配置、維運指令

六個服務一覽：4 個常駐、2 個一次性

compose 專案名稱 workflow-hub (docker/docker-compose.yml 的 name)

6

服務

4

常駐容器

2

一次性工作

2

映像

2

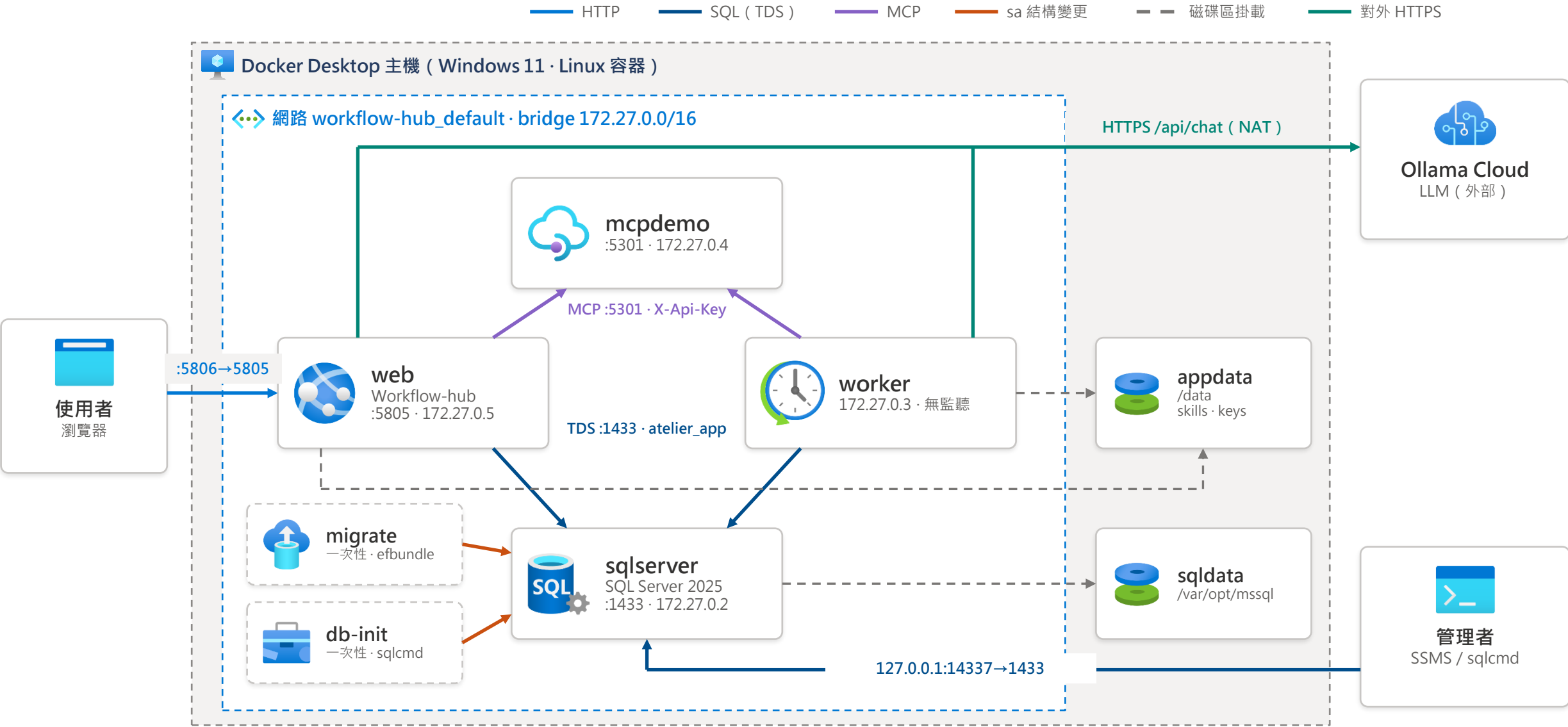
具名磁碟區

1

bridge 網路

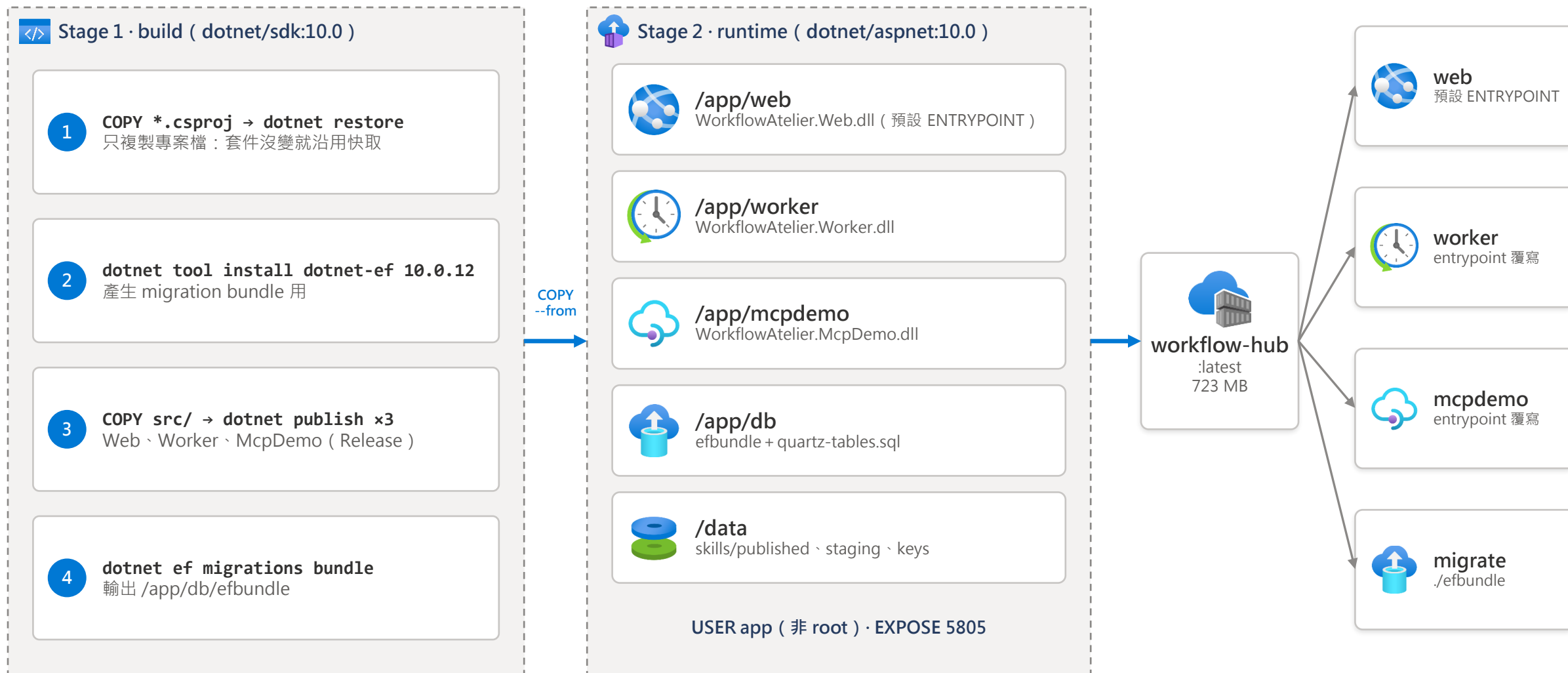
服務	容器名稱	映像	角色	生命週期	對外埠
sqlserver	Workflow-hub-sqlserver	mssql/server:2025-latest	SQL Server 2025 Developer · 存放 AGENTDB	常駐 · healthcheck	127.0.0.1:14337→1433
migrate	workflow-hub-migrate-1	workflow-hub:latest (負責建置)	efbundle 建立 / 升級 AGENTDB 結構	一次性 (Exited 0)	—
db-init	workflow-hub-db-init-1	mssql/server:2025-latest	Quartz 14 張 QRTZ_ 表、專用登入 atelier_app	一次性 (Exited 0)	—
web	Workflow-hub	workflow-hub:latest	ASP.NET Core MVC + Web API · 手動試跑	常駐 · healthcheck	0.0.0.0:5806→5805
mcpdemo	Workflow-hub-mcpdemo	workflow-hub:latest	示範 MCP Server (訂單工具)	常駐	無 (內部 :5301)
worker	Workflow-hub-worker	workflow-hub:latest	Quartz 排程觸發 + 待執行工作派送	常駐	無

整體架構圖



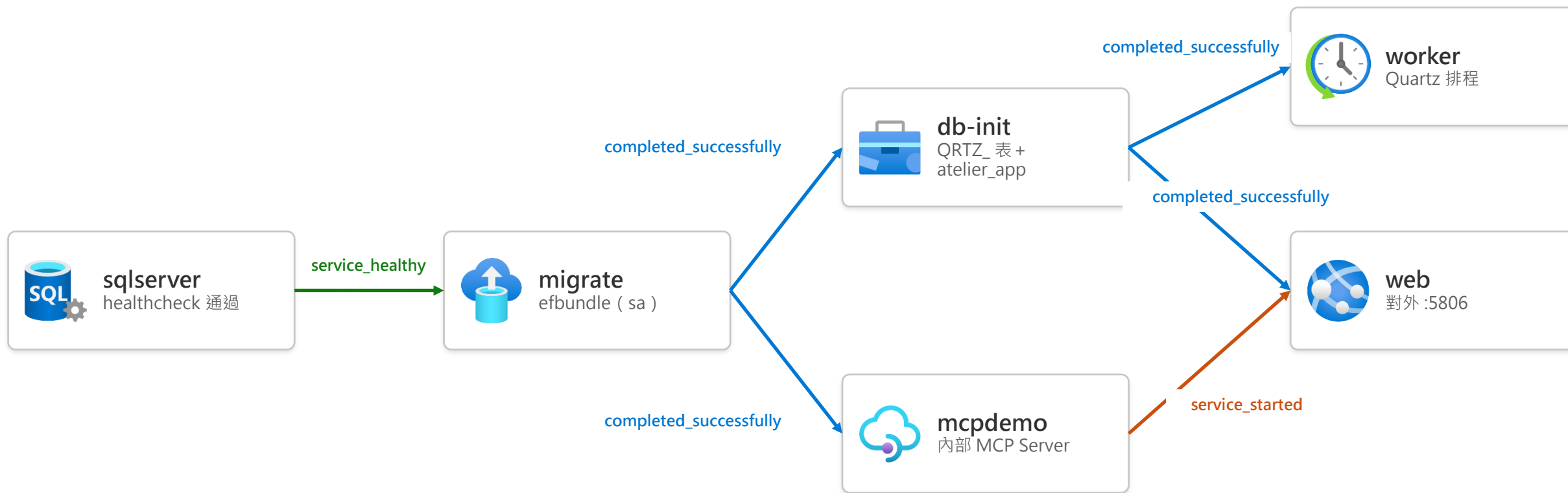
一個映像、四種用途：多階段 Dockerfile

docker/Dockerfile · 建置內容 (context) 為範例根目錄 · 產出 workflow-hub:latest (實測 723 MB)



啟動順序：depends_on 條件串成的相依圖

docker compose up -d --build 之後，compose 依條件逐一啟動；每次 up 都會先重跑 migrate 與 db-init



● **service_healthy** 等 healthcheck 回報 healthy (sqlserver：每 10 秒 sqlcmd SELECT 1，最多 20 次)

● **service_completed_successfully** 等一次性容器以結束碼 0 結束；失敗則後續服務不啟動

● **service_started** 只等容器已啟動，不檢查就緒 (web 對 mcpdemo)

sqlserver：平台唯一的狀態中心

容器 Workflow-hub-sqlserver · hostname workflow-hub-sql · mcr.microsoft.com/mssql/server:2025-latest

 容器設定

項目	設定
版本	SQL Server 2025 · MSSQL_PID=Developer
對外埠	127.0.0.1:14337 → 1433 (只開放本機)
資料	具名磁碟區 sqldata → /var/opt/mssql
密碼	MSSQL_SA_PASSWORD=\${SA_PASSWORD:?...}
healthcheck	sqlcmd -Q "SELECT 1"
檢查頻率	interval 10s · timeout 5s · retries 20 · start_period 20s
重啟	restart: unless-stopped
時區	TZ=Asia/Taipei

 AGENTDB (實測 35 張資料表)

平台資料表 (EF Core migration)

工作流、執行紀錄、技術包、MCP 伺服器與工具呼叫、排程、稽核、Identity 帳號

Quartz 系統表 QRTZ_ (14 張)

Job、Trigger、Cron、鎖定；Worker 重啟後從這裡還原排程

登入與權限



sa

只給 migrate、db-init
結構變更 (DDL)



atelier_app

web、worker 使用
db_datareader +
db_datawriter

migrate 與 db-init : 一次性的資料庫初始化

兩者都設定 restart: "no" , 完成後以 Exited (0) 結束 · 可重複執行 (冪等)

migrate · workflow-hub:latest

```
working_dir: /app/db
entrypoint: ./efbundle --connection "$MIGRATE_CONNECTION"
# MIGRATE_CONNECTION : Server=sqlserver;Database=AGENTDB;User Id=sa
```

- 1 等待 sqlserver healthy
depends_on: service_healthy
- 2 套用 EF Core migration
全新資料庫 : 9 個 migration
- 3 已是最新 → 不變更
重跑安全 (冪等)
- 4 以結束碼 0 結束
觸發 db-init 與 mcpdemo

db-init · mssql/server:2025-latest (借用 sqlcmd)

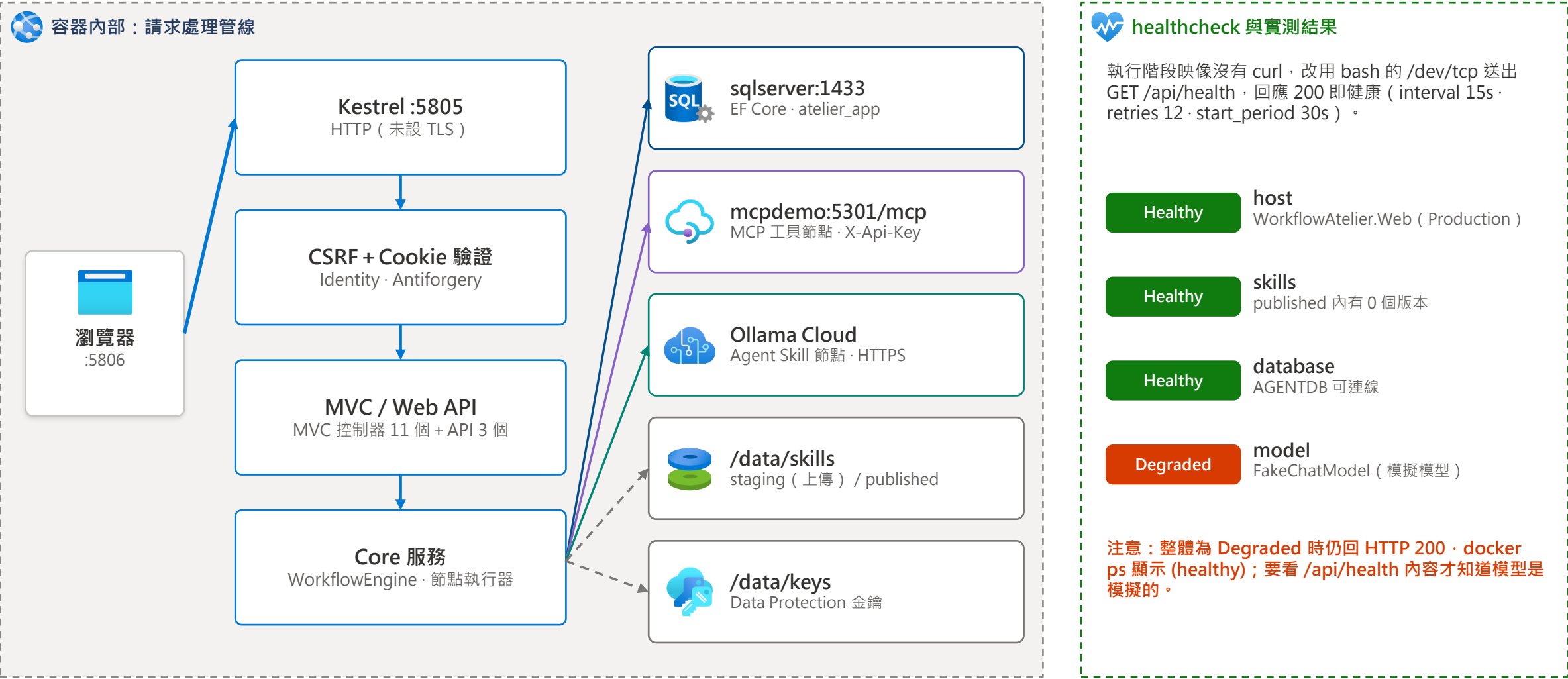
唯讀 bind mount (:ro)

-  **init-db.sh** → /init/init-db.sh
-  **app-login.sql** → /init/app-login.sql
-  **../sql/quartz-tables.sql** → /init/quartz-tables.sql

- 1 Quartz 系統表
sqlcmd -v DatabaseName=AGENTDB -i quartz-tables.sql
已存在就略過
- 2 專用登入 atelier_app
CREATE LOGIN / ALTER LOGIN (密碼輪替)
AGENTDB : db_datareader + db_datawriter
- 3 安全細節
sa 密碼放 SQLCMDPASSWORD · 不出現在命令列
-f 65001 : 以 UTF-8 讀含中文的腳本

web : 平台網站與 Web API (容器 Workflow-hub)

ASP.NET Core 10 MVC + Web API · Kestrel :5805 · 主機 0.0.0.0:5806 · ASPNETCORE_ENVIRONMENT=Production



worker : Quartz 排程與工作派送 (容器 Workflow-hub-worker)

.NET 泛型主機 · 不開任何埠 · DOTNET_ENVIRONMENT=Production (泛型主機讀 DOTNET_ 前綴)

四個 HostedService



ScheduleSyncWorker

每 15 秒

WorkflowSchedules → Quartz Job / Cron Trigger (依時區、misfire 策略)



Quartz Scheduler

持久化 QRTZ_

觸發 ScheduledRunJob : 只建立「待執行」紀錄 (預定時間防重複)



RunDispatcherWorker

每 5 秒

原子領取待執行紀錄 → WorkflowEngine 執行 ; 啟動時中斷者標記「結果不明」



PlatformMonitorWorker

每 30 秒

呼叫共用健康檢查並寫日誌

對外連線與掛載



sqlserver:1433

AGENTDB (atelier_app)
Quartz 持久化儲存 QRTZ_



mcpdemo:5301/mcp
排程工作流的 MCP 工具節點



Ollama Cloud (HTTPS)

Agent Skill 節點

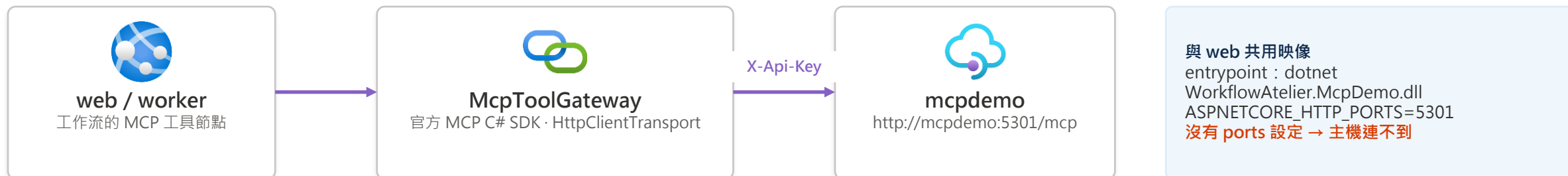


appdata → /data

讀 skills/published (與 web 共用)

mcpdemo : 只在內部網路提供的示範 MCP Server

容器 Workflow-hub-mcpdemo · ModelContextProtocol.AspNetCore · 無狀態 Streamable HTTP · 不發布埠



金鑰怎麼來 : CredentialRef → 環境變數

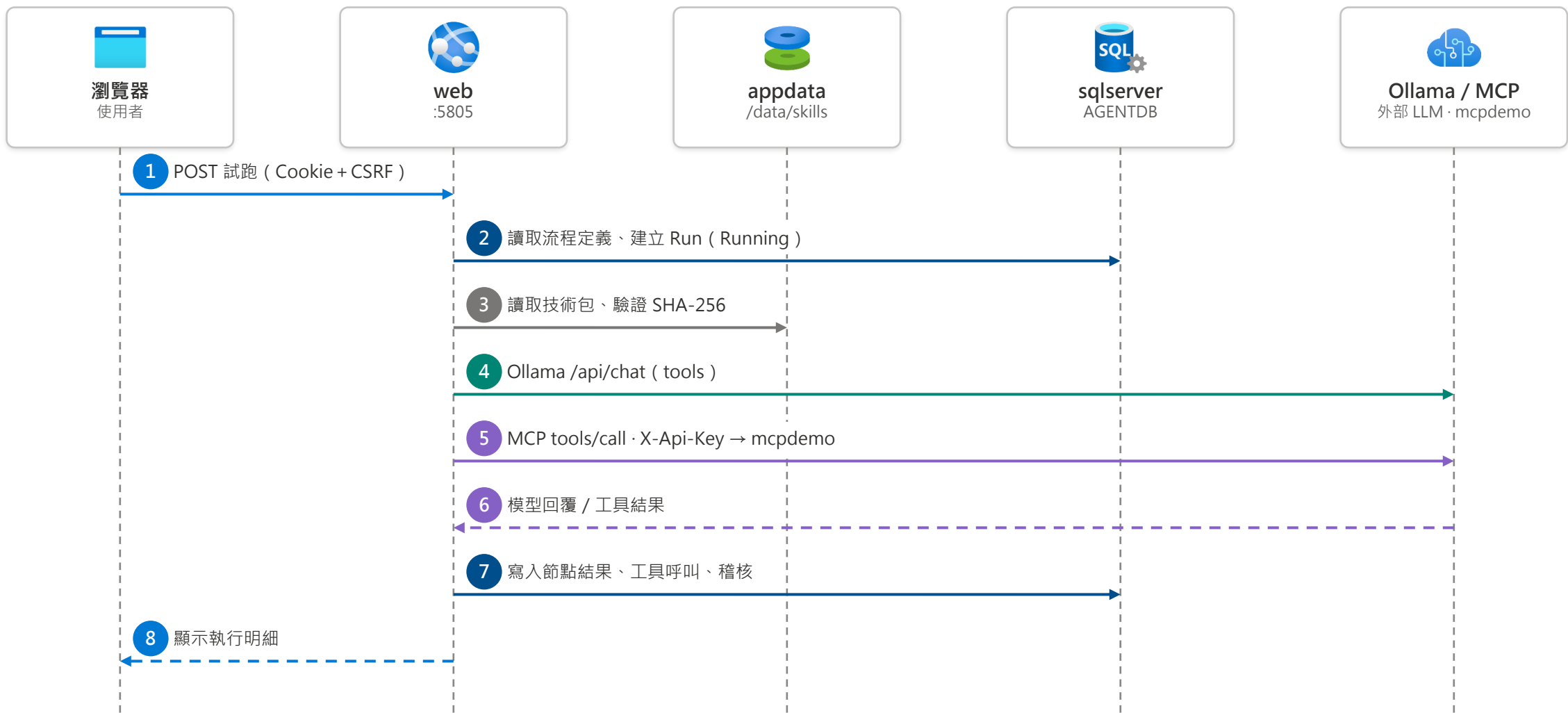
- 1 管理頁登錄端點**
Endpoint = http://mcpdemo:5301/mcp
CredentialRef = MCP_DEMO_ORDERS_KEY (只存名稱)
- 2 執行時解析**
config["MCP_DEMO_ORDERS_KEY"] → 實際金鑰
找不到就失敗 : 「找不到認證參照...的設定值」
- 3 放進請求標頭**
AdditionalHeaders : X-API-Key
連線逾時 15 秒 · Streamable HTTP

mcpdemo 內部

- ① 金鑰中介軟體**
X-API-Key ≠ ApiKey → 401 Unauthorized
- ② MapMcp("/mcp")**
無狀態 HTTP (SessionMode = Stateless)
- ③ list_pending_orders**
列出待處理訂單 (limit 1 ~ 50)
- ④ get_order**
依訂單編號查詢 · 例如 A001

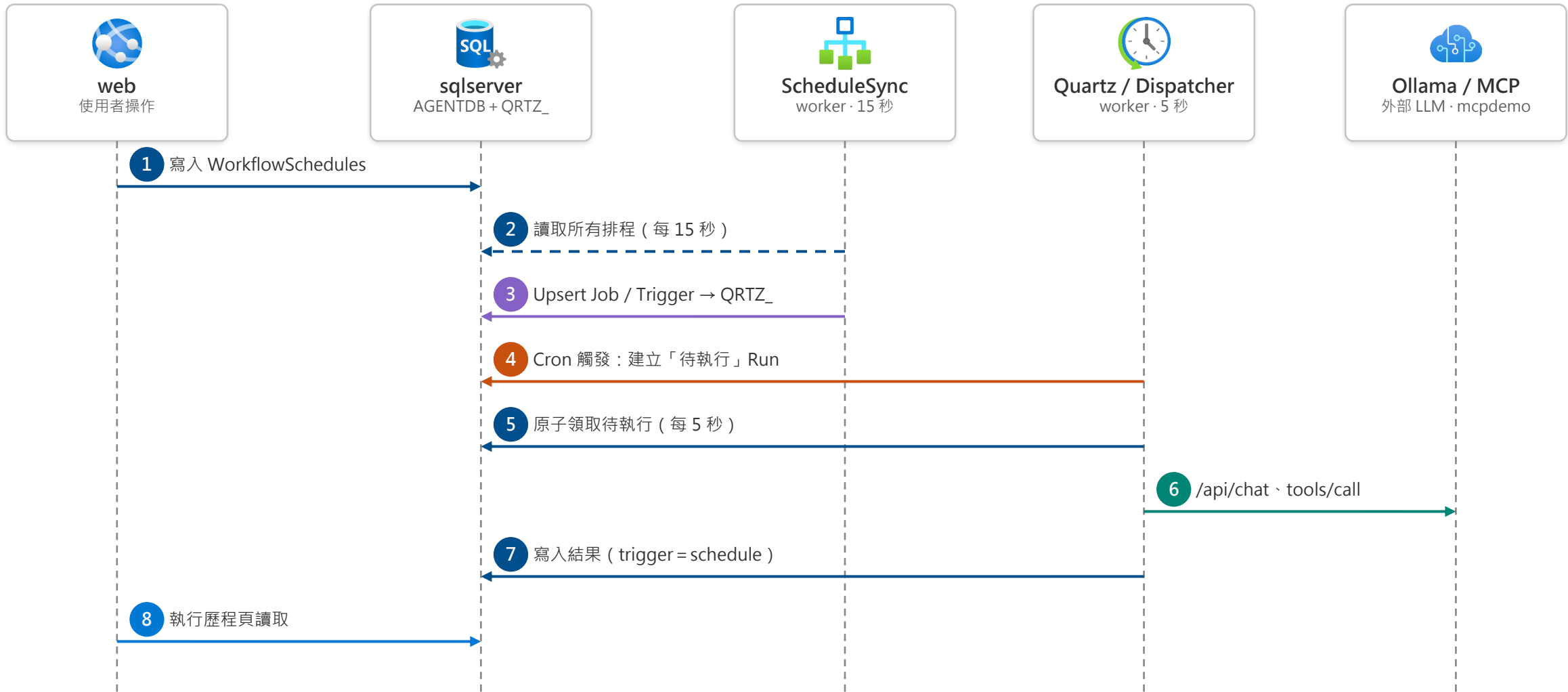
序列圖①：使用者在 web 手動試跑 workflow

手動執行在 web 容器內完成；排程執行則由 worker 負責（下一頁）



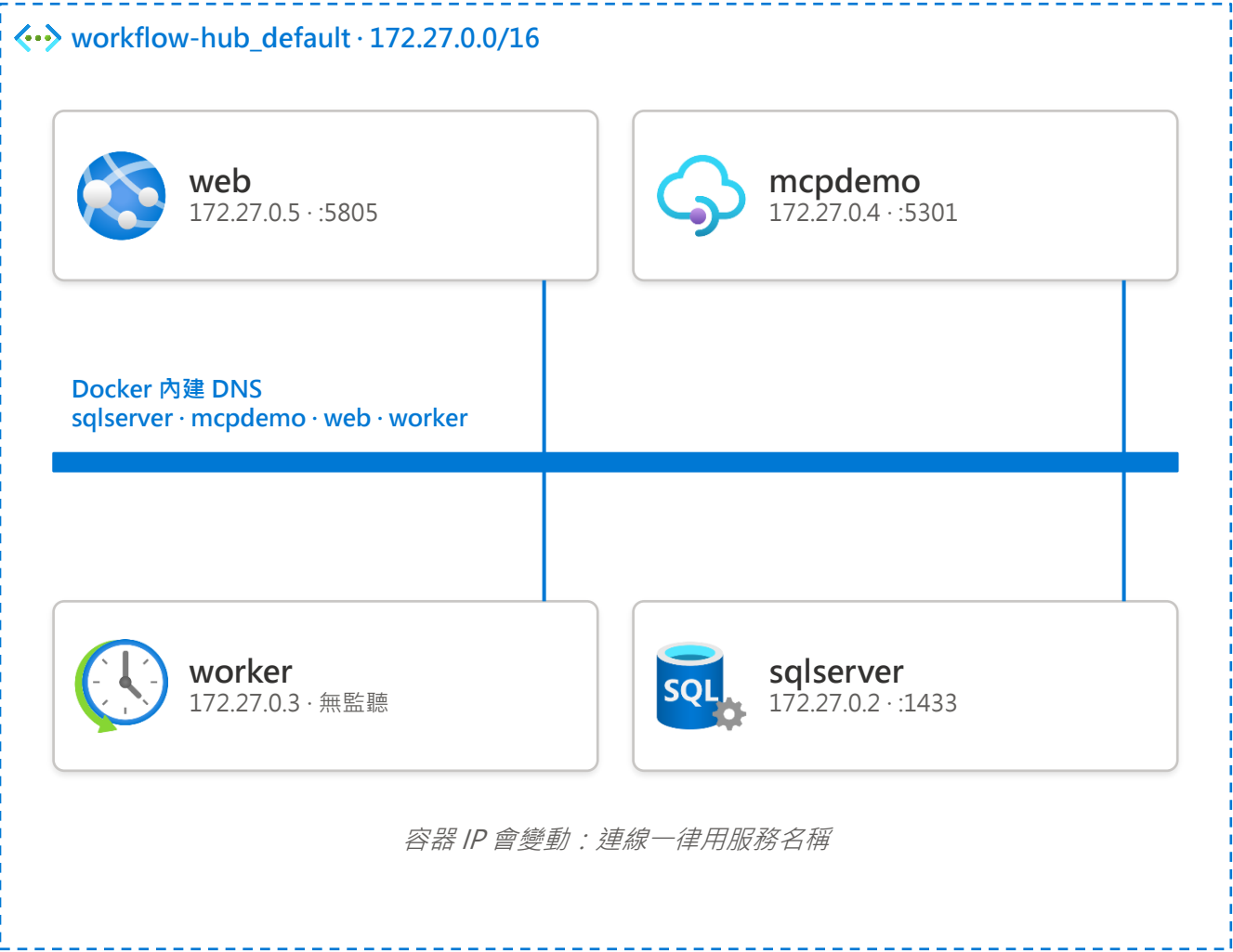
序列圖②：排程觸發到 worker 執行

web 與 worker 不直接呼叫彼此：兩者只透過 AGENTDB 交換狀態（解耦）



網路與連接埠：只開放必要的入口

compose 自動建立 bridge 網路 workflow-hub_default；服務之間以「服務名稱」做 DNS 解析



主機綁定	容器	可達範圍
0.0.0.0:5806	web :5805	所有網路介面 (區網可連)
127.0.0.1:14337	sqlserver :1433	只有本機 (SSMS)
(無)	mcpdemo :5301	只限 compose 網路
(無)	worker	不監聽任何埠



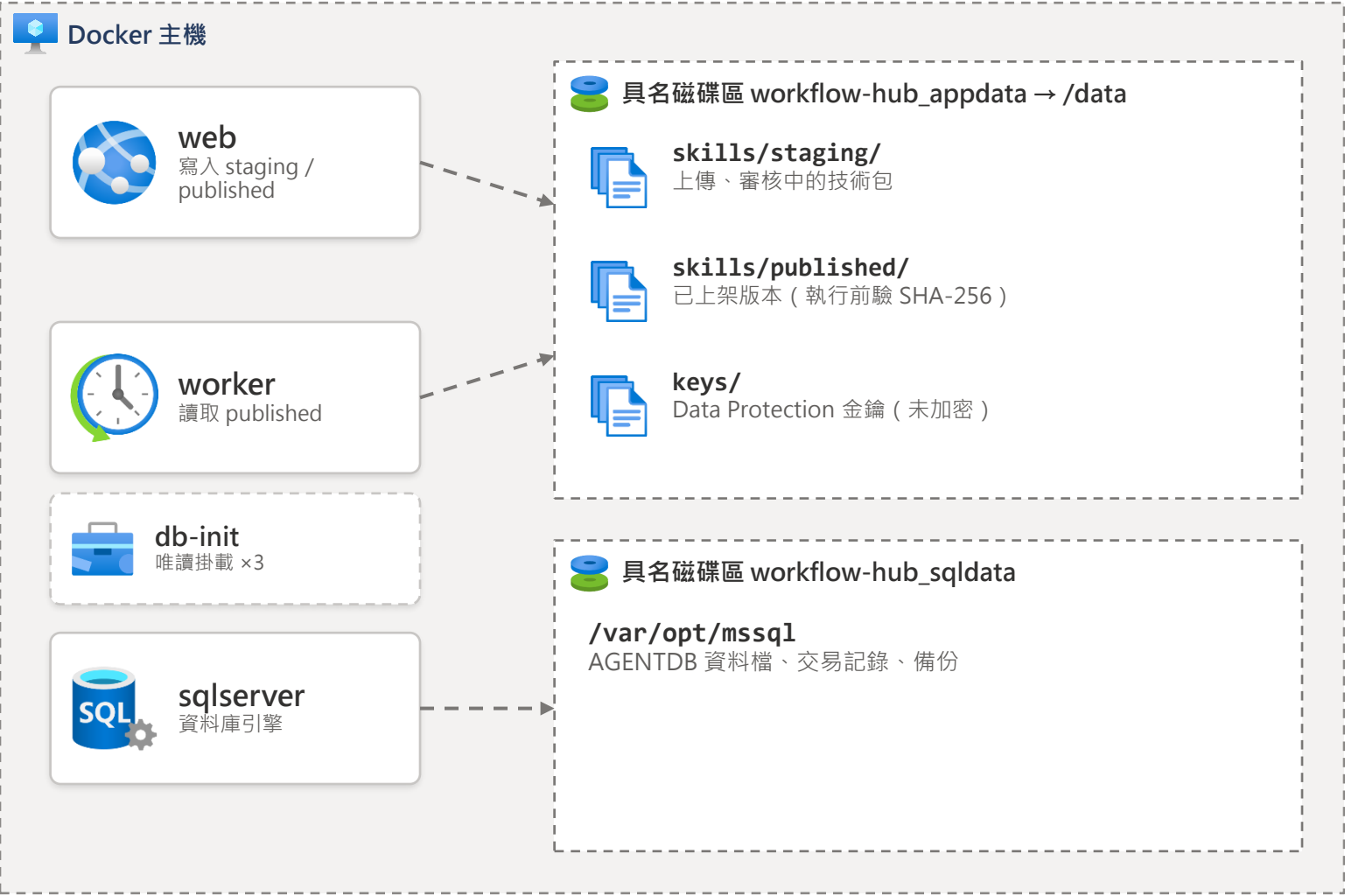
已知限制：沒有 TLS
目前以 HTTP 提供服務；對外前以反向代理 (IIS / Nginx 等) 終止 TLS 或在 Kestrel 設定憑證



連接埠選擇的原因
主機 5805 被 rag15-web 使用 → web 用 5806；14333 ~ 14336 已被其他專案的 SQL Server 使用 → 14337

儲存：兩個具名磁碟區 + 三個唯讀掛載

容器可隨時重建，狀態只存在磁碟區；web 與 worker 必須看到同一份技術包



唯讀 bind mount (db-init)

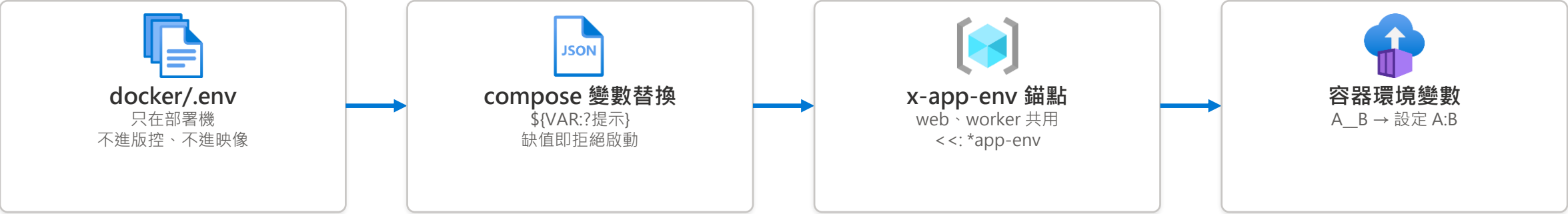
主機檔案	容器路徑
docker/init-db.sh	/init/init-db.sh
docker/app-login.sql	/init/app-login.sql
sql/quartz-tables.sql	/init/quartz-tables.sql

備份原則

- **同時備份** sqldata 與 appdata 要同一時間點，否則雜湊或路徑對不上
- **金鑰保管** Linux 容器無 DPAPI，keys 為明文，備份檔需加密保存
- **勿用 down -v** 會連同具名磁碟區一起刪除

設定與秘密：.env → compose 變數 → 環境變數

秘密只放 docker/.env (.gitignore 、.dockerignore 都排除) ；範本 docker/.env.example 只有變數名稱



變數 (.env)	用途	sqlserver	migrate	db-init	web	worker	mcpdemo
SA_PASSWORD	sa 密碼 (結構變更)	●	●	●			
APP_DB_PASSWORD	atelier_app 密碼			●	●	●	
MCP_DEMO_ORDERS_KEY	MCP 金鑰 (呼叫端 = 驗證端)				●	●	● ApiKey
OLLAMA_API_KEY	Ollama Cloud (空白→模擬模型)				●	●	
ADMIN_EMAIL / PASSWORD	第一次啟動建立管理者				●		
WEB_PORT	主機對外埠 (預設 5806)				●		

健康檢查、重啟策略與故障行為

設定錯誤就不啟動、外部依賴缺席就降級、執行中斷就交給人查核

服務	healthcheck	重啟策略	故障時的行為
sqlserver	sqlcmd SELECT 1 · 10s×20	unless-stopped	migrate 等到 healthy 才開始
migrate	—	no	失敗 → db-init、mcpdemo、web、worker 都不啟動
db-init	—	no	失敗 → web、worker 不啟動
web	bash /dev/tcp → /api/health · 15s×12	unless-stopped	設定缺漏即拒絕啟動；Degraded 仍回 200
worker	(無)	unless-stopped	無 QRTZ_ 表即啟動失敗；中斷的執行標記「結果不明」
mcpdemo	(無)	unless-stopped	缺 ApiKey 不啟動；工具節點失敗會記錄在執行紀錄



啟動前：快速失敗
DeploymentChecks 檢查連線字串、SkillsRoot 絕對路徑；錯了就不啟動



執行中：降級而不中斷
無 Ollama 金鑰 → 模擬模型 (Degraded) ；
published 不存在 → 503

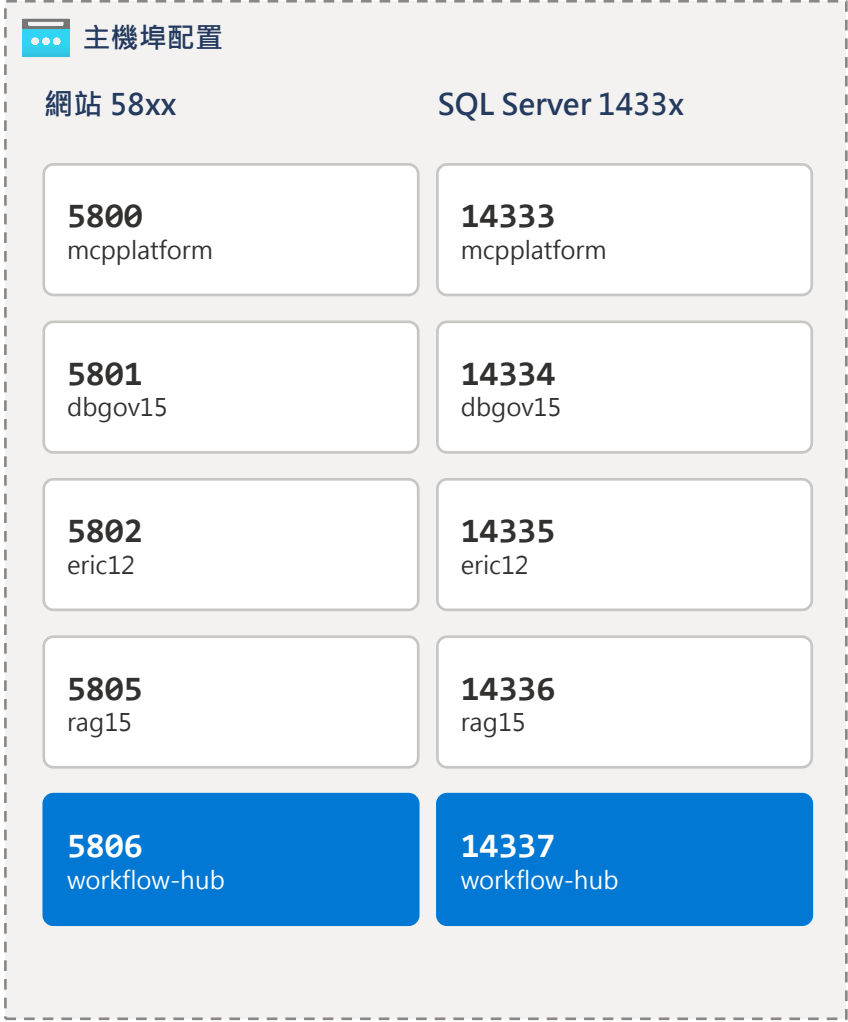


中斷後：交給人查核
重啟時 Running 的排程執行 → 「結果不明」，不自動重跑

主機共存：同一台 Docker Desktop 上的 compose 專案

實測 docker ps：各專案各自一個 bridge 網路，彼此隔離；只有發布到主機的埠會互相衝突

compose 專案	網站埠	SQL Server 埠	執行中的容器
n8n	127.0.0.1:5678	—	n8n
n8n-docker	0.0.0.0:5679	—	n8n-ubike
mcpplatform	127.0.0.1:5800	127.0.0.1:14333	web、demoapis、sqlserver
dbgov15	127.0.0.1:5801	127.0.0.1:14334	web、worker、sqlserver
eric12	0.0.0.0:5802	127.0.0.1:14335	api、sqlserver
rag15	127.0.0.1:5805	127.0.0.1:14336	web、sqlserver
workflow-hub (本專案)	0.0.0.0:5806	127.0.0.1:14337	web、worker、mcpdemo、sqlserver



維運指令速查

在範例根目錄 (12_部署設定與成果驗收) 執行

部署與升級 (已實測)

```
# 1. 準備秘密 ( docker/.env 不進版控 )
Copy-Item docker\.env.example docker\.env

# 2. 建置映像並依相依順序啟動
docker compose -f docker/docker-compose.yml up -d --build

# 3. 冒煙測試 : HTTP 200 才算成功
pwsh deploy\smoke.ps1 -WebUrl http://localhost:5806
```

檢查與排錯

```
# 狀態 : 一次性工作應為 Exited (0)
docker compose -f docker/docker-compose.yml ps -a

# 追蹤日誌 ( web / worker / migrate / db-init )
docker compose -f docker/docker-compose.yml logs -f worker

# 網路成員與 IP
docker network inspect workflow-hub_default

# 健康檢查明細 ( Healthy / Degraded )
curl http://localhost:5806/api/health
```



docker ps 顯示 healthy ≠ 全部正常
再看 /api/health 的 model 項目是否 Degraded



備份 sqldata + appdata
同一時間點；金鑰為明文，備份檔要加密



down -v 會刪除資料

docker compose down 保留磁碟區，加 -v 則一併刪除

總結

六個容器，一個狀態中心



一個映像、四種用途

web、worker、mcpdemo、migrate 共用同一映像，以 entrypoint 區分



啟動順序由條件保證

healthy → 建結構 → 建 QRTZ_ 與登入 → 常駐服務



狀態只在磁碟區

sqldata + appdata 同時備份；容器可隨時重建



AGENTDB 是唯一的協調點

web 與 worker 不互相呼叫，排程與執行狀態都經由資料庫交換



最小權限與最小暴露

sa 只給一次性工作；只開 5806 與本機 14337；秘密只在 .env



可觀測、可降級

/api/health 分項回報；中斷的執行交給人查核