docker 建立

dockerfile建立

如果後端一開始在visual studio 2022 建了一個dockerfile 後來又改了路徑 可以重新設定dockerfile

可以使用 Visual Studio Code 重新生成 Dockerfile。下面是一個基本的步驟：

1. 在 Visual Studio Code 中打開你的專案文件夾。
2. 在左側的側邊欄中，選擇你的專案文件夾，以確保 Visual Studio Code 正在查看你的專案。
3. 在菜單中，選擇“擴展”（Extensions）並搜尋安裝名為 “Docker” 的擴展。點擊安裝。
4. 安裝完成後，點擊左側側邊欄的 “Docker” 圖標。
5. 在 Docker 擴展中，你應該能夠看到一個選項叫做 “Add Dockerfiles to Workspace”。點擊它。
6. 這將提示你選擇你的應用程式類型。選擇後，Visual Studio Code 將為你的應用程式生成一個 Dockerfile。
7. 完成後，你可以在你的專案文件夾中看到新生成的 Dockerfile。

8. 根據你的需求對生成的 Dockerfile 進行調整，確保它與你的應用程式兼容並滿足你的要求。

分別在 前後端利用 docker build -t XXXXX(自己建立一個標籤)

打包後端image執行

必須進到 xxx.csproj 這個檔案的目錄下 在執行

```
docker build -t yang_backend -f ./leaveSystem_vue/Dockerfile .
```

打包前端image執行

```
docker build -t yang_frontend -f ./leavesystem/Dockerfile .
```

建立docker資料庫

資料庫docker掛載必須先建立docker的資料庫 再將自己的資料庫備份還原到docker內的資料庫，因為docker是隔離環境

```
docker run -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=!QAZ2wsx#EDC' -p 1

docker cp E:\SSMS\MSSQL16.MSSQLSERVER\MSSQL\Backup\leaveSystem.b

進入docker內部的資料庫
docker exec -it leavesystem_vue-sqlserver-1 /opt/mssql-tools/bin
```

E:\SSMS\MSSQL16.MSSQLSERVER\MSSQL\Backup\leaveSystem.bak (看你自己的資料庫檔案在哪裡)

```
還原資料庫
RESTORE DATABASE leaveSystem FROM DISK = '/tmp/leaveSystem.bak'
go

查詢資料庫
use leaveSystem
go

查詢資料表
select * from dbo.Leave
go
```

執行docker-compose

在有docker-compose文件的目錄下執行

```
docker-compose up -d
```

PS:分開運行的話

要分別運行 Docker 容器而不是使用 `docker-compose`，您可以使用 `docker run` 命令單獨啟動每個容器。以下是基於您提供的 `docker-compose` 文件的 `docker run` 命令的一些範例。

首先，確保您有一個網路設置好，以便容器可以互相通信：

```
docker network create --driver bridge --subnet=192.168.48.0/24 leavesystem_vue_default
```

接下來，對於每個服務，使用 `docker run` 啟動容器：

1. **SQL Server 容器**:

```
docker run -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=!QAZ2wsx#EDC' \
    --network=leavesystem_vue_default --ip=192.168.48.2 \
    --name sqlserver -p 1433:1433 \
    -v sqlserver_data:/var/opt/mssql \
    -d mcr.microsoft.com/mssql/server:2022-latest
```

1. **Backend 容器**:

確保您已經建立了後端映像 `yang_backend`。

```
docker run -e 'DATABASE_HOST=sqlserver' -e 'DATABASE_USER=SA'
\
    -e 'DATABASE_PASSWORD=!QAZ2wsx#EDC' -e 'DATABASE_NAME=lea
veSystem' \
    --network=leavesystem_vue_default --ip=192.168.48.3 \
    --name backend -p 5163:8080 \
    -v backend_data_protection_keys:/root/.aspnet/DataProtect
ion-Keys \
    -d yang_backend
```

1. **Frontend 容器**:

確保您已經建立了前端映像 `yang_frontend`。

```
docker run --network=leavesystem_vue_default --ip=192.168.48.
4 \
    --name frontend -p 8080:8080 \
    -d yang_frontend
```

請注意，使用這種方法時，您需要手動管理容器之間的依賴性和啟動順序。例如，您可能需要先運行 SQL Server 容器，等它完全啟動並運行後，再運行 Backend 容器，最後是 Frontend 容器。

此外，確保您在運行 `docker run` 命令之前已經建立了必要的 Docker 映像，並且映像名稱與您在命令中使用的名稱匹配。如果您在本地建立映像，使用 `docker build` 命令並提供正確的路徑到您的 Dockerfile。

docker-compose 內容詳細說明

Docker Compose 文件的一部分，用於定義和運行多個容器化應用程序的服務。它包括三個主要服務：`sqlserver`、`backend`、和 `frontend`。下面是對每個服務配置的詳細解釋：

共同設定

- **version: "3.8"**：指定了使用的 Docker Compose 文件版本，這裡是 3.8 版本。
- **services**：定義了在此 Docker Compose 配置中將要運行的所有服務（容器）。
- **volumes**：定義了數據卷，用於數據持久化。
- **networks**：定義了網絡設定，用於容器間通信。

SQL Server 服務 (`sqlserver`)

- **image**：指定使用的 Docker 鏡像，這裡是 Microsoft SQL Server 2022 的最新版。
- **user**：以 root 用戶身份運行容器。
- **environment**：設定環境變量，包括接受最終用戶授權協議（EULA）和設定 SA 帳號的密碼。
- **ports**：將容器的 1433 端口映射到宿主機的同一端口，用於數據庫訪問。
- **volumes**：將數據卷掛載到容器的 `/var/opt/mssql` 目錄，用於數據持久化。
- **networks**：設定網絡，並指定固定的 IPv4 地址。

後端服務 (`backend`)

- **image**：指定使用的後端 Docker 鏡像。
- **user**：以 root 用戶身份運行容器。
- **environment**：設定後端應用所需的環境變量，包括數據庫連接信息。
- **ports**：映射容器內部端口到宿主機端口，以便外部訪問後端服務。
- **volumes**：用於數據保護密鑰的持久化存儲。
- **depends_on**：指定後端服務依賴於 `sqlserver` 服務。
- **networks**：設定網絡，並指定固定的 IPv4 地址。

前端服務 (`frontend`)

- **image**：指定使用的前端 Docker 鏡像。
- **ports**：映射容器內部端口到宿主機端口，以便外部訪問前端應用。
- **depends_on**：指定前端服務依賴於 `backend` 服務。
- **networks**：設定網絡，並指定固定的 IPv4 地址。

網絡設定 (`leavesystem_vue_default`)

- **driver**：使用橋接模式來設定 Docker 網絡。
- **ipam**：IP 地址管理（IPAM），設定子網和容器的 IP 地址。