docker-compose-db-shared.yml

version: "3.7"

services:

postgres-system:

image: postgres:15

container\_name: postgres-system

restart: always

environment:

# (Khuyến nghị) Dùng user/db riêng cho metadata của Airflow

- POSTGRES\_USER=airflow

- POSTGRES\_PASSWORD=airflow

- POSTGRES\_DB=airflow\_meta

ports:

- "5434:5432"

volumes:

- postgres-data:/var/lib/postgresql/data # <-- Sử dụng named volume

networks:

- shared-network # <-- Gán service vào mạng chung

networks:

shared-network: # <-- Định nghĩa rằng đây là mạng bên ngoài

external: true

volumes:

postgres-data:

name: airflow\_postgres\_data # Đặt tên cụ thể cho volume để tránh trùng lặp

docker-compose-airflow-local.yml

version: '3.8'

services:

redis:

image: redis:latest

container\_name: airflow-redis

networks:

- shared-network

expose:

- 6379

restart: always

airflow-webserver:

build: .

platform: linux/amd64

container\_name: airflow-webserver

user: airflow

volumes:

- ./:/opt/airflow/

environment:

- AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_CONN=${AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_CONN}

- AIRFLOW\_\_CORE\_\_EXECUTOR=CeleryExecutor

- AIRFLOW\_\_WEBSERVER\_\_SECRET\_KEY=${AIRFLOW\_\_WEBSERVER\_\_SECRET\_KEY}

- AIRFLOW\_\_CORE\_\_FERNET\_KEY=${AIRFLOW\_\_CORE\_\_FERNET\_KEY}

- AIRFLOW\_\_WEBSERVER\_\_WORKERS=2

- AIRFLOW\_\_CELERY\_\_BROKER\_URL=redis://redis:6379/0

- AIRFLOW\_\_CELERY\_\_RESULT\_BACKEND=redis://redis:6379/0

- AIRFLOW\_HOME=/opt/airflow

- AWS\_ACCESS\_KEY\_ID=${AWS\_ACCESS\_KEY\_ID}

- AWS\_SECRET\_ACCESS\_KEY=${AWS\_SECRET\_ACCESS\_KEY}

- AWS\_DEFAULT\_REGION=ap-southeast-1

ports:

- "8080:8080"

depends\_on:

- redis

networks:

- shared-network

restart: always

command: ["airflow", "webserver"]

airflow-scheduler:

build: .

platform: linux/amd64

container\_name: airflow-scheduler

user: airflow

volumes:

- ./:/opt/airflow/

environment:

- AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_CONN=${AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_CONN}

- AIRFLOW\_\_CORE\_\_EXECUTOR=CeleryExecutor

- AIRFLOW\_\_CELERY\_\_BROKER\_URL=redis://redis:6379/0

- AIRFLOW\_\_CELERY\_\_RESULT\_BACKEND=redis://redis:6379/0

- AIRFLOW\_HOME=/opt/airflow

- AWS\_ACCESS\_KEY\_ID=${AWS\_ACCESS\_KEY\_ID}

- AWS\_SECRET\_ACCESS\_KEY=${AWS\_SECRET\_ACCESS\_KEY}

- AWS\_DEFAULT\_REGION=ap-southeast-1

depends\_on:

- redis

networks:

- shared-network

restart: always

command: ["airflow", "scheduler"]

airflow-worker:

build: .

platform: linux/amd64

container\_name: airflow-worker

user: airflow

volumes:

- ./:/opt/airflow/

environment:

- AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_CONN=${AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_CONN}

- AIRFLOW\_\_CORE\_\_EXECUTOR=CeleryExecutor

- AIRFLOW\_\_CELERY\_\_BROKER\_URL=redis://redis:6379/0

- AIRFLOW\_\_CELERY\_\_RESULT\_BACKEND=redis://redis:6379/0

- AIRFLOW\_HOME=/opt/airflow

- AWS\_ACCESS\_KEY\_ID=${AWS\_ACCESS\_KEY\_ID}

- AWS\_SECRET\_ACCESS\_KEY=${AWS\_SECRET\_ACCESS\_KEY}

- AWS\_DEFAULT\_REGION=ap-southeast-1

depends\_on:

- redis

networks:

- shared-network

restart: always

command: ["airflow", "celery", "worker"]

nginx:

build:

context: ./nginx

dockerfile: Dockerfile

restart: always

ports:

- 80:80

environment:

- NGINX\_PORT=${NGINX\_PORT}

- NGINX\_HOST=${NGINX\_HOST}

networks:

- shared-network

volumes:

- ./nginx/templates:/etc/nginx/templates

depends\_on:

airflow-webserver:

condition: service\_started

networks:

shared-network:

external: true

nginx/templates/default.conf.template

server {

listen ${NGINX\_PORT};

server\_name ${NGINX\_HOST};

location / {

proxy\_set\_header Host $http\_host;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_redirect off;

proxy\_buffering off;

proxy\_pass http://airflow-webserver:8080;

}

}

.env

DATABASE\_URL\_MONITOR=postgresql://airflow:airflow@postgres-system:5432/monitor

DATABASE\_URL\_REDSHIFT=redshift+psycopg2://xxxx/{{DATABASE\_NAME}}?sslmode=require

NGINX\_HOST=localhost

NGINX\_PORT=8080

AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_CONN=postgresql+psycopg2://airflow:airflow@postgres-system:5432/airflow\_meta

AIRFLOW\_\_WEBSERVER\_\_SECRET\_KEY=

AIRFLOW\_\_CORE\_\_FERNET\_KEY=

Dockerfile

FROM apache/airflow:2.10.4-python3.10

USER airflow

# copy file requirement to container

COPY requirement.txt /opt/airflow/requirement.txt

# upgrade pip và install library

RUN python -m pip install --upgrade pip && \

pip install --no-cache-dir -r /opt/airflow/requirement.txt

RUN airflow db init && \

airflow users create -u admin -p admin -r Admin -e admin@example.com -f Admin -l User

docker-compose -f docker-compose-airflow-local.yml --env-file=./.env up -d --build

Trong airflow-webserver docker exec

airflow db init

airflow users create -u admin -p admin -r Admin -e admin@example.com -f Admin -l User

Link: https://gemini.google.com/app/ef5eece2bc128177

Docker for api-ui monitor

version: "3.7"

services:

api:

container\_name: mapping-mtk-api

restart: always

build:

context: ./api

dockerfile: Dockerfile

environment:

- PRODUCTION=1

- DATABASE\_URL=${DATABASE\_URL}

- S3\_BUCKET\_NAME=${S3\_BUCKET\_NAME}

- S3\_REGION=${S3\_REGION}

- DATABASE\_REDSHIFT\_URL=${DATABASE\_REDSHIFT\_URL}

- AWS\_ACCESS\_KEY\_ID=${AWS\_ACCESS\_KEY\_ID}

- AWS\_SECRET\_ACCESS\_KEY=${AWS\_SECRET\_ACCESS\_KEY}

- COGNITO\_CLIENT\_ID=${COGNITO\_CLIENT\_ID}

- COGNITO\_USER\_POOL\_ID=${COGNITO\_USER\_POOL\_ID}

- COGNITO\_CLIENT\_SECRET=${COGNITO\_CLIENT\_SECRET}

- REGION=${REGION}

- REDSHIFT\_HOST=${REDSHIFT\_HOST}

- REDSHIFT\_PORT=${REDSHIFT\_PORT}

- REDSHIFT\_USER=${REDSHIFT\_USER}

- REDSHIFT\_PASSWORD=${REDSHIFT\_PASSWORD}

- REDSHIFT\_DBNAME=${REDSHIFT\_DBNAME}

- FB\_API\_VERSION=${FB\_API\_VERSION}

- FB\_APP\_ID=${FB\_APP\_ID}

- FB\_APP\_SECRET=${FB\_APP\_SECRET}

- FB\_REDIRECT\_URI\_BASE=${FB\_REDIRECT\_URI\_BASE}

volumes:

- ./api:/home/dll/api

networks:

- postgres-system\_postgres-system-network

ui:

container\_name: mapping-mtk-ui

build:

context: ./ui

dockerfile: Dockerfile

environment:

- NGINX\_PORT=80

- NGINX\_HOST=mtmonitor.xyz

- VITE\_API\_BASE\_URL=${VITE\_API\_BASE\_URL}

ports:

- 3333:80

volumes:

- ./ui:/home/dll/ui

networks:

- postgres-system\_postgres-system-network

depends\_on:

api:

condition: service\_started

networks:

postgres-system\_postgres-system-network:

external: true

volumes:

api:

ui: