**🔎 Checklist kiểm tra DB (Docker Postgres)**

**1. Kiểm tra container DB có chạy chưa**

docker ps

👉 Phải thấy container db (theo docker-compose.yml). Nếu không có → start lại:

docker compose up -d db

**2. Kiểm tra log DB**

docker logs db

👉 Tìm dòng kiểu:  
database system is ready to accept connections → DB đã ok.

**3. Kết nối thử bằng psql (bên trong container)**

docker exec -it db psql -U postgres -d app

(hoặc -d ${POSTGRES\_DB} theo .env)

👉 Nếu login được → gõ lệnh test:

\dt -- liệt kê bảng

SELECT now(); -- test query

**4. Kiểm tra migrations đã apply chưa**

Trong psql:

SELECT \* FROM schema\_migrations;

👉 Có bản ghi = migration đã chạy.

**5. Kiểm tra DB từ service API (readiness endpoint)**

curl http://localhost:8080/readiness

👉 Nếu API trả { "ok": true } (hoặc tương tự) → DB kết nối được.

**⚡ Quick step gọn nhất**

docker compose up -d db

docker exec -it db psql -U postgres -d app -c "SELECT now();"

Bổ sung **.env.example** và **docker-compose.yml** (có DB, migrate, Adminer tùy chọn) + lệnh kiểm tra nhanh DB.

**.env**

# App

APP\_ENV=dev

PORT=8080

# Postgres (Docker)

POSTGRES\_USER=postgres

POSTGRES\_PASSWORD=postgres

POSTGRES\_DB=app

POSTGRES\_PORT=5432

# Kết nối DB từ các service trong compose (host = db)

DATABASE\_URL=postgres://postgres:postgres@db:5432/app?sslmode=disable

# (tuỳ chọn) Kết nối DB từ máy host

HOST\_DATABASE\_URL=postgres://postgres:postgres@localhost:5432/app?sslmode=disable

**docker-compose.yml (ngắn gọn, chuẩn dev)**

version: "3.9"

services:

db:

image: postgres:16

container\_name: db

environment:

POSTGRES\_USER: ${POSTGRES\_USER}

POSTGRES\_PASSWORD: ${POSTGRES\_PASSWORD}

POSTGRES\_DB: ${POSTGRES\_DB}

ports:

- "${POSTGRES\_PORT}:5432" # host:container

volumes:

- pgdata:/var/lib/postgresql/data

healthcheck:

test: ["CMD-SHELL", "pg\_isready -U ${POSTGRES\_USER} -d ${POSTGRES\_DB}"]

interval: 3s

timeout: 3s

retries: 30

migrate:

image: migrate/migrate:4

depends\_on:

db:

condition: service\_healthy

volumes:

- ./db/migrations:/migrations:ro

environment:

DATABASE\_URL: ${DATABASE\_URL}

entrypoint: ["/bin/sh","-c"]

command: >

"migrate -path=/migrations -database ${DATABASE\_URL} up"

restart: "no"

api:

build:

context: .

dockerfile: ./build/Dockerfile

env\_file: .env

depends\_on:

db:

condition: service\_healthy

migrate:

condition: service\_completed\_successfully

environment:

PORT: ${PORT}

DATABASE\_URL: ${DATABASE\_URL}

APP\_ENV: ${APP\_ENV}

ports:

- "${PORT}:${PORT}"

# Tùy chọn: Adminer để xem DB trên trình duyệt

adminer:

image: adminer

depends\_on:

db:

condition: service\_healthy

ports:

- "8081:8080"

environment:

# đăng nhập: System=PostgreSQL, Server=db, User=postgres, Pass=postgres, DB=app

ADMINER\_DEFAULT\_SERVER: db

volumes:

pgdata:

**Cách dùng nhanh**

# 1) tạo .env từ mẫu

cp .env.example .env

# 2) khởi động DB + migrate + API

docker compose up --build -d

# 3) kiểm tra DB sẵn sàng

docker exec -it db psql -U ${POSTGRES\_USER:-postgres} -d ${POSTGRES\_DB:-app} -c "SELECT now();"

# 4) mở Adminer (tùy chọn)

# http://localhost:8081 (Server=db, User=postgres, Pass=postgres, DB=app)

**Gợi ý thêm (script check nhanh)**

Tạo scripts/check\_db.sh:

#!/usr/bin/env bash

set -e

docker ps --format "table {{.Names}}\t{{.Status}}"

docker logs db --tail=20

docker exec -it db psql -U "${POSTGRES\_USER:-postgres}" -d "${POSTGRES\_DB:-app}" -c "SELECT now();"

chmod +x scripts/check\_db.sh && ./scripts/check\_db.sh

Lưu ý Windows: nếu ACE muốn mount SQL seed riêng, dùng path tương đối trong repo (đã cấu hình ở ./db/migrations) để tránh lỗi quyền thư mục.