1) Tạo thư mục & file migrations

Trong root project:

mkdir -p db/migrations

db/migrations/000001\_create\_core\_tables.up.sql

-- Enable extension for UUID

CREATE EXTENSION IF NOT EXISTS pgcrypto;

-- devices

CREATE TABLE IF NOT EXISTS devices (

id uuid PRIMARY KEY DEFAULT gen\_random\_uuid(),

name text NOT NULL,

metadata jsonb NOT NULL DEFAULT '{}'::jsonb,

created\_at timestamptz NOT NULL DEFAULT now(),

updated\_at timestamptz NOT NULL DEFAULT now()

);

-- readings

CREATE TABLE IF NOT EXISTS readings (

id bigserial PRIMARY KEY,

device\_id uuid NOT NULL REFERENCES devices(id) ON DELETE CASCADE,

value double precision NOT NULL,

recorded\_at timestamptz NOT NULL DEFAULT now()

);

CREATE INDEX IF NOT EXISTS idx\_readings\_device\_time ON readings(device\_id, recorded\_at DESC);

-- plans

CREATE TABLE IF NOT EXISTS plans (

id bigserial PRIMARY KEY,

device\_id uuid NOT NULL REFERENCES devices(id) ON DELETE CASCADE,

threshold double precision NOT NULL,

rule text NOT NULL DEFAULT 'gt', -- gt/lt/eq...

active boolean NOT NULL DEFAULT true,

created\_at timestamptz NOT NULL DEFAULT now()

);

CREATE INDEX IF NOT EXISTS idx\_plans\_device\_active ON plans(device\_id, active);

-- alerts

CREATE TABLE IF NOT EXISTS alerts (

id bigserial PRIMARY KEY,

device\_id uuid NOT NULL REFERENCES devices(id) ON DELETE CASCADE,

reading\_id bigint REFERENCES readings(id) ON DELETE SET NULL,

plan\_id bigint REFERENCES plans(id) ON DELETE SET NULL,

status text NOT NULL DEFAULT 'open', -- open|serviced

message text NOT NULL,

created\_at timestamptz NOT NULL DEFAULT now(),

serviced\_at timestamptz

);

CREATE INDEX IF NOT EXISTS idx\_alerts\_device\_status ON alerts(device\_id, status);

db/migrations/000001\_create\_core\_tables.down.sql

DROP TABLE IF EXISTS alerts;

DROP TABLE IF EXISTS plans;

DROP TABLE IF EXISTS readings;

DROP TABLE IF EXISTS devices;

(Tùy chọn) db/migrations/000002\_seed\_minimal.up.sql

INSERT INTO devices (name) VALUES ('device-1');

INSERT INTO plans (device\_id, threshold, rule)

SELECT id, 50.0, 'gt' FROM devices WHERE name='device-1';

(Tùy chọn) db/migrations/000002\_seed\_minimal.down.sql

DELETE FROM plans WHERE device\_id IN (SELECT id FROM devices WHERE name='device-1');

DELETE FROM devices WHERE name='device-1';

2) Chạy migrations (không cần cài gì thêm)

Cách A — Dùng image migrate (khuyến nghị trên Windows)

DB đang mở cổng 5432 ra host; dùng host.docker.internal để container migrate truy cập host (Docker Desktop).

# Lần 1: tạo schema

docker run --rm -v "%cd%/db/migrations":/migrations \

migrate/migrate:4 -path=/migrations \

-database "postgres://admin:Vietanh4791@host.docker.internal:5432/main-1?sslmode=disable" up

# (tùy chọn) áp dụng seed

docker run --rm -v "%cd%/db/migrations":/migrations \

migrate/migrate:4 -path=/migrations \

-database "postgres://admin:Vietanh4791@host.docker.internal:5432/main-1?sslmode=disable" up

PowerShell có thể dùng ${PWD} thay cho %cd%.

Nếu API/DB chạy trong cùng compose và bạn không expose 5432 ra host, ta chạy migrate trong network compose (em sẽ viết giúp nếu anh cần).

Cách B — Dùng psql trong container (cực gọn, không versioning)

Nhanh để test, nhưng không ghi nhận version migrations.

# Apply schema

docker exec -i wh-ma-db psql -U admin -d main-1 < db/migrations/000001\_create\_core\_tables.up.sql

# (tùy chọn) seed

docker exec -i wh-ma-db psql -U admin -d main-1 < db/migrations/000002\_seed\_minimal.up.sql

3) Kiểm tra kết quả

docker exec -it wh-ma-db psql -U admin -d main-1 -c "\dt"

docker exec -it wh-ma-db psql -U admin -d main-1 -c "select \* from devices limit 5;"

Kỳ vọng: thấy 4 bảng; nếu seed thì có device-1.

4) Readiness test lại

curl -s http://localhost:8080/readiness

Kỳ vọng: {"ok":true,"db":"up"}

5) (Tùy chọn) Makefile lệnh ngắn

Anh muốn em tạo Makefile với:

make migrate-up

make migrate-down

make seed

…để bấm 1 lệnh là chạy được, em sẽ soạn ngay.

1. **Cài migrate CLI**

go install github.com/golang-migrate/migrate/v4/cmd/migrate@latest

export PATH=$PATH:$(go env GOPATH)/bin

migrate -version # kiểm tra, ra "dev" cũng ok

1. **Tạo thư mục**

mkdir -p db/migrations

1. **Sinh file migration đầu tiên**

migrate create -ext sql -dir db/migrations -seq init\_schema

→ sinh ra 000001\_init\_schema.up.sql và .down.sql.

1. **Viết schema trong up.sql**

* Bảng devices
* Bảng readings (FK device)
* Bảng plans (FK device, threshold)
* Bảng alerts (FK device, reading)

Trong down.sql → drop 4 bảng ngược lại.

1. **(Optional) Seed dữ liệu mẫu**

* Migration 000002\_seed.up.sql để insert 1 device + 1 plan.
* Migration 000002\_seed.down.sql để xoá seed.

1. **Chạy migration**

export DATABASE\_URL="postgres://admin:Vietanh4791@localhost:5432/main-1?sslmode=disable"

migrate -path db/migrations -database "$DATABASE\_URL" up

1. **Kiểm tra DB**

docker compose exec -e PGPASSWORD='Vietanh4791' db \

psql -U admin -d main-1 -c "\dt"

→ phải thấy các bảng.

1. **Rollback khi cần**

migrate -path db/migrations -database "$DATABASE\_URL" down 1

migrate -path db/migrations -database "$DATABASE\_URL" version