部署文档

1 软硬件依赖

- 2vCPUs | 4GiB | s7.large.2 CentOS 7.6 64bit
- Docker / Docker Compose
- Python 3.11（强烈建议用conda创建虚拟环境）
- openGauss 3.x（容器）
- OpenAI API Key（用于 Embedding/LLM）

2 openGauss 部署

1. 拉取docker镜像

```
docker pull enmotech/opengauss:latest
```

Tips：建议服务器上面挂个代理，或者本地拉取（我本地有代理）导入服务器

2. 创建opengauss容器并启动

```
docker run -d --name opengauss \
  -e GS_PASSWORD='Enmo@123' \
  -e GAUSSHOME=/usr/local/opengauss \
  -e LD_LIBRARY_PATH=/usr/local/opengauss/lib \
  -e PATH=/usr/local/opengauss/bin:$PATH \
  -p 5432:5432 enmotech/opengauss:3.1.0
```

后续开机只需要启动就好

```
docker start opengauss
```

3. 创建表，我这里用的gsql

```
export GAUSSHOME=/usr/local/opengauss
export LD_LIBRARY_PATH=$GAUSSHOME/lib:$LD_LIBRARY_PATH
export PATH=$GAUSSHOME/bin:$PATH
gsql -d postgres -U gaussdb -W Enmo@123
```

3 WhyHow部署

1. 下载安装程序

```
git clone https://gitcode.com/paradox/whyhow_opengauss.git
cd knowledge-graph-studio
pip install -r requirements.txt
```

2. 配置环境变量

```
cp .env.sample .env
```

```
WHYHOW__EMBEDDING__OPENAI__API_KEY=<你的openai api key>
WHYHOW__GENERATIVE__OPENAI__API_KEY=<你的openai api key>

WHYHOW__OPENGAUSS__HOST=<数据库的host>
WHYHOW__OPENGAUSS__PORT=<数据库docker映射出来的端口>5432
WHYHOW__OPENGAUSS__DATABASE=<数据库名称>
WHYHOW__OPENGAUSS__USER=<数据库用户名>
WHYHOW__OPENGAUSS__PASSWORD=<数据库密码>
WHYHOW__OPENGAUSS__ECHO_SQL=<是否打印 SQL 语句>

# e.g.
# WHYHOW__OPENGAUSS__HOST=127.0.0.1
# WHYHOW__OPENGAUSS__PORT=5432
# WHYHOW__OPENGAUSS__DATABASE=postgres
# WHYHOW__OPENGAUSS__USER=gaussdb
# WHYHOW__OPENGAUSS__PASSWORD=Enmo@123
# WHYHOW__OPENGAUSS__ECHO_SQL=true
```

3. 运行API服务器

```
uvicorn whyhow_api.main:app --host 0.0.0.0 --port 8000 --reload
```

Tips：如果服务器断连记着先杀死进程再重新运行

```
pkill -f "uvicorn .*whyhow_api.main:app" || true
```

4 初始化数据库，创建表

```
-- 基础表（统一 UUID 主键）
CREATE TABLE IF NOT EXISTS users (
  id UUID PRIMARY KEY,
  email VARCHAR(255) UNIQUE,
  username VARCHAR(255),
  firstname VARCHAR(255),
  lastname VARCHAR(255),
  api_key VARCHAR(64) UNIQUE NOT NULL,
  providers JSON,
  active BOOLEAN NOT NULL DEFAULT TRUE,
  created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
  updated_at TIMESTAMPTZ NOT NULL DEFAULT NOW()
);

CREATE TABLE IF NOT EXISTS workspaces (
  id UUID PRIMARY KEY,
  name VARCHAR(128) UNIQUE NOT NULL,
  description TEXT,
  user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
  created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
  updated_at TIMESTAMPTZ NOT NULL DEFAULT NOW()
);

CREATE TABLE IF NOT EXISTS graphs (
  id           uuid PRIMARY KEY,
  schema_id    uuid NULL,
  workspace_id uuid NOT NULL REFERENCES workspaces(id) ON DELETE CASCADE,
  created_by   uuid NOT NULL REFERENCES users(id) ON DELETE CASCADE,
```

```
  public       boolean NOT NULL DEFAULT false,
  name         text NOT NULL,
  created_at   timestamptz NOT NULL DEFAULT now(),
  updated_at   timestamptz NOT NULL DEFAULT now()
);

CREATE INDEX IF NOT EXISTS idx_graphs_created_by   ON graphs(created_by);
CREATE INDEX IF NOT EXISTS idx_graphs_workspace_id ON graphs(workspace_id);
CREATE INDEX IF NOT EXISTS idx_graphs_created_at   ON graphs(created_at DESC);

CREATE TABLE IF NOT EXISTS rules (
  id           uuid PRIMARY KEY,
  workspace_id uuid        NULL REFERENCES workspaces(id) ON DELETE CASCADE,
  created_by   uuid        NOT NULL REFERENCES users(id) ON DELETE CASCADE,
  name         text,
  body         json,
  created_at   timestamptz NOT NULL DEFAULT now()
);

CREATE TABLE IF NOT EXISTS schemas (
  id           uuid PRIMARY KEY,
  workspace_id uuid NOT NULL,
  created_by   uuid NOT NULL,
  name         text,
  body         jsonb,
  created_at   timestamptz DEFAULT now()
);

CREATE TABLE IF NOT EXISTS documents (
  id          uuid PRIMARY KEY,
  created_by  uuid        NOT NULL REFERENCES users(id) ON DELETE CASCADE,
  status      varchar(32) NOT NULL DEFAULT 'uploaded',
  metadata    jsonb,
  created_at  timestamptz NOT NULL DEFAULT now(),
  updated_at  timestamptz NOT NULL DEFAULT now()
);

CREATE TABLE IF NOT EXISTS document_workspaces (
  document_id  uuid NOT NULL REFERENCES documents(id) ON DELETE CASCADE,
  workspace_id uuid NOT NULL REFERENCES workspaces(id) ON DELETE CASCADE,
  created_at   timestamptz NOT NULL DEFAULT now(),
  PRIMARY KEY (document_id, workspace_id)
);

CREATE TABLE IF NOT EXISTS chunks (
  id            uuid PRIMARY KEY,
  document_id   uuid        NULL REFERENCES documents(id) ON DELETE CASCADE,
  workspaces    uuid[]      NOT NULL,           -- 参与的 workspace 列表
  data_type     text        NOT NULL,           -- 'string' / 'object'
  content       text,
  content_obj   json,
  embedding     json,
  tags          json         NOT NULL DEFAULT '{}'::json,
  user_metadata json         NOT NULL DEFAULT '{}'::json,
  metadata      json         NOT NULL DEFAULT '{}'::json,
  created_by    uuid         NOT NULL REFERENCES users(id) ON DELETE CASCADE,
  created_at    timestamptz  NOT NULL DEFAULT now(),
  updated_at    timestamptz  NOT NULL DEFAULT now()
```

```
);

CREATE TABLE IF NOT EXISTS nodes (
  id          uuid PRIMARY KEY,
  graph_id    uuid        NOT NULL,
  name        text        NOT NULL,
  label       text        NOT NULL,
  properties  json        NOT NULL DEFAULT '{}'::json,
  chunks      uuid[]      NOT NULL DEFAULT '{}',    -- 关键：默认空数组
  created_by  uuid        NOT NULL REFERENCES users(id) ON DELETE CASCADE,
  created_at  timestamptz NOT NULL DEFAULT now(),
  updated_at  timestamptz NOT NULL DEFAULT now()
);

CREATE TABLE IF NOT EXISTS triples (
  id            uuid PRIMARY KEY,
  graph_id      uuid        NOT NULL,
  head_node_id  uuid,
  tail_node_id  uuid,
  relation_name text        NOT NULL,
  properties    json        NOT NULL DEFAULT '{}'::json,
  chunks        uuid[]      NOT NULL DEFAULT '{}',  -- 关键：默认空数组
  created_by    uuid        NOT NULL REFERENCES users(id) ON DELETE CASCADE,
  embedding     json,
  created_at    timestamptz NOT NULL DEFAULT now(),
  updated_at    timestamptz NOT NULL DEFAULT now()
);


CREATE TABLE IF NOT EXISTS tasks (
  id           CHAR(36)                NOT NULL,
  user_id      UUID                    NOT NULL,
  title        VARCHAR(255)            NOT NULL,
  description  TEXT,
  status       VARCHAR(50)             NOT NULL DEFAULT 'pending',
  created_at   TIMESTAMPTZ             DEFAULT now(),
  updated_at   TIMESTAMPTZ             DEFAULT now(),
  CONSTRAINT tasks_pkey PRIMARY KEY (id),
  CONSTRAINT tasks_user_id_fkey
      FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
)
WITH (orientation = row, compression = no);

CREATE TABLE IF NOT EXISTS queries (
  id         uuid PRIMARY KEY,
  user_id    uuid        NOT NULL REFERENCES users(id) ON DELETE CASCADE,
  graph_id   uuid,
  status     varchar(32) NOT NULL DEFAULT 'pending',
  name       text,
  payload    json,
  created_at timestamptz NOT NULL DEFAULT now(),
  updated_at timestamptz NOT NULL DEFAULT now()
);
```