HƯỚNG DẪN TRIỂN KHAI & SỬ DỤNG THIEN-MILVUS-VECTORDB

1. Tải và khởi chạy Docker

1. Clone source THIEN-MILVUS-VECTORDB về local.
2. Chạy Docker Compose:
   docker compose up -d
3. Kiểm tra container đã chạy:
   docker ps

2. Milvus-CLI và thiết lập môi trường

1. Vào Milvus-CLI.
2. Kích hoạt máy ảo
   venv\Scripts\activate
3. Cài đặt thư viện:
   pip install -r requirements.txt

Lưu ý: Nếu gặp lỗi DLL của `ujson` cần cài thêm VSCode kèm C++ Build Tools vì `ujson` sử dụng C++ để build.

3. Cấu trúc thư mục chính

main_api/
├── main_api.py
├── main_api_header.py
├── milvus_until.py
├── milvus_until_api.py
├── milvus_utils_api_online.py
├── milvus_untils_api_25_7.py

Các endpoint chính:

- collection: Xử lý các tác vụ collection như tạo, xóa, tạo index, ...
- data: Xử lý CRUD dữ liệu trong collection.

4. Mô tả các file Milvus chính

milvus_until.py

- Phiên bản cũ nhất, hỗ trợ CRUD đầy đủ qua CLI.

- Phù hợp cho DevOps quản lý database.

milvus_until_api.py

- Phiên bản cũ hỗ trợ Web API CRUD cơ bản.
- Dễ thao tác nhưng chưa mở rộng tính năng.

milvus_utils_api_online.py

- Tối ưu cho việc tạo collection, bổ sung nhiều metadata hỗ trợ RAG.
- Hỗ trợ auto UUID, random ID hoặc cấu hình ID tăng dần.
- Metadata giúp tìm kiếm ANN chuẩn hơn.
- Auto UUID chỉ mang tính định danh, không tìm kiếm trực tiếp bằng UUID.
- Muốn search: phải dùng auto ID hoặc ID tăng dần để trỏ trực tiếp vào data.
- Vector data không thể chỉnh sửa trực tiếp vì embedding đa chiều (hơn 700 dimensions).
    - Muốn update: phải re-embedding input mới và gán lại ID cũ.
- Phiên bản này triển khai qua Zillcloud.

milvus_untils_api_25_7.py

- Bản mới nhất, config ngày 25/07/2025.
- Hỗ trợ chạy local (standalone).
- Nếu cần production chịu tải khoảng 100 request/giây, cần triển khai cluster, Kubernetes và Kafka queue.

5. API chính

- main_api.py: API chính thức, đã cấu hình CORS.
- main_api_header.py: Bổ sung Token Auth để phục vụ thương mại hóa hoặc tăng tính bảo mật.

6. Khởi chạy API

- Chạy bằng Uvicorn:

```
uvicorn main_api:app --host 0.0.0.0 --port 8001 --reload
```

- Truy cập web api
  http://localhost:8001/docs#/

7. Yêu cầu bắt buộc về dữ liệu

- Dữ liệu trước khi insert bắt buộc phải được clean 100%.
- Các công ty lớn như Meta, AWS đều clean trước khi chunk và insert.
- Nếu không clean dữ liệu:

    - Việc chunk dữ liệu từ PDF sẽ dễ lỗi.
    - Tìm kiếm sẽ không chính xác.
    - Chi phí vận hành tăng cao do vector dư thừa.

Ví dụ lỗi thường gặp:

- Chunk PDF không clean dẫn đến vector chứa dữ liệu chồng chéo.
- Khi query như "Công ty ABC kinh doanh lĩnh vực gì?" dễ trả về nội dung dư thừa, dài dòng, gây lãng phí token.
- Mô hình RAG không tối ưu buộc phải tăng top K để tìm kiếm đủ dữ liệu, làm chi phí tăng và prompt nặng hơn.

Kết luận: Muốn RAG và ANN hoạt động tốt, cần clean dữ liệu, chunk đúng chuẩn, bổ sung metadata rõ ràng. Điều này giúp tiết kiệm chi phí và nâng cao độ chính xác tìm kiếm.

8. Tham khảo thêm

Tài liệu README.md đã mô tả chi tiết. Vui lòng đọc kỹ trước khi triển khai.