MLOps Lab2 Assignment

Task Summary

Create main component of your system from the assignment 1.

1. For this assignment you should create a baseline model for your system, create an API wrap around it (FastAPI, Seldon, BentoML, etc. but you can use other frameworks as well), put it in the docker container and make a demo how your API works.
2. Train a baseline model. You should organise your data for training in DB or object storage (this may be S3, minio or other storage). Create an Airflow job for putting new data to your storage (training dataset).

Note: all artefacts for lab results demo are located in the `MLOps/reranker_pipeline_demo` folder

Baseline Model

The baseline model is a sentence-level CrossEncoder based on the pre-trained `cross-encoder/ms-marco-MiniLM-L-6-v2` checkpoint. Its main task is pairwise relevance ranking — given a user query and a candidate document (or sentence), it estimates the relevance score.

In this project, the CrossEncoder is fine-tuned on custom training data consisting of `(query, document, label)` triples to adapt it to the domain-specific reranking task.

The trained model is wrapped into a **FastAPI service**, containerized with Docker, and deployed as a microservice. The system is integrated with **MinIO** for storing training data and uses **Airflow** to automate the ingestion of new data into the object store.

For the sake of convenience we will move all the files for the demo to the separated folder and create the next structure:

Project Description and Component Overview

'''

```
reranker_pipeline_demo/
├── clients/
│   ├── config.py
│   ├── openai_client_wrapper.py
│   ├── qdrant_client_wrapper.py
```

```
│   └── supabase_client_wrapper.py
├── dags/
│   └── data_pipeline_dag.py
├── logs/
├── models/
│   └── model_registry.py
├── plugins/
├── prompts/
│   ├── answer_prompt.txt
│   ├── fallback_answer_prompt.txt
│   ├── generate_queries.txt
│   └── router_prompt.txt
├── demo.py
├── docker-compose.yml
├── Dockerfile
├── Dockerfile.airflow
├── generate_training_data.py
├── load_reranker_data.py
├── parse_raw_batch_data.py
├── README.md
├── requirements.txt
├── reranker_fast_api_app.py
├── train_reranker.py
└── utils.py
```

This project implements a reranking pipeline for a RAG-based system. Its main objective is to train and deploy a reranker model that scores the relevance of document-query pairs and serves it via an API for inference tasks.

Core Components

- **Baseline Model**:
  The reranker is based on the pretrained model `cross-encoder/ms-marco-MiniLM-L-6-v2` from HuggingFace. It is fine-tuned on a custom dataset for binary relevance classification (relevant vs. irrelevant documents for a given query).
- **Training Pipeline**:
  - `generate_training_data.py`, `parse_raw_batch_data.py`, `load_reranker_data.py`: responsible for building the training dataset from external sources and preparing it for training.
  - `train_reranker.py`: performs fine-tuning of the CrossEncoder model using PyTorch/Transformers.
- **Model Serving**:

    - `reranker_fast_api_app.py` : exposes the trained reranker model through a FastAPI service.
    - `Dockerfile` : builds a production-ready container for model serving.
    - `docker-compose.yml` : orchestrates the deployment of services including FastAPI, Airflow, MinIO, and PostgreSQL.
- **Airflow Integration**:
    - `dags/data_pipeline_dag.py` : defines a DAG that automates the retraining pipeline, loading of new data, and pushing it to object storage (MinIO).
    - `Dockerfile.airflow` : builds a production-ready container for data scripts orchestration.
- **Supporting Modules**:
    - `clients/` : wrappers for interacting with OpenAI, Supabase, Qdrant, etc.
    - `prompts/` : prompt templates for generating queries or responses as part of the RAG workflow.
    - `model_registry.py` : utility for saving/loading trained model versions.

System Deployment

Navigate to the root folder of the reranker pipeline project

```
cd PycharmProjects/rag-news-assistant/reranker_pipeline_demo
```

Run a one-time initialization of the Airflow database (creates metadata DB and admin user)

```
docker compose up airflow-init
```

```
[+] Running 4/4
 ✔ Network reranker_pipeline_demo_default          Created              0.1s
 ✔ Volume "reranker_pipeline_demo_postgres_data"   Created              0.0s
 ✔ Container reranker_pipeline_demo-postgres-1     Created              0.1s
 ✔ Container reranker_pipeline_demo-airflow-init-1 Created              0.1s
Attaching to airflow-init-1
airflow-init-1  | /home/airflow/.local/lib/python3.8/site-packages/airflow/cli/commands/db_command.py:43 DeprecationWarning: `db init` is deprecated.  Use `db migrate` instead to migrate the db and/or airflow conne
ctions create-default-connections to create the default connections
airflow-init-1  | DB: postgresql+psycopg2://airflow:***@postgres/airflow
airflow-init-1  | [2025-07-11T02:32:56.587+0000] {migration.py:213} INFO - Context impl PostgresqlImpl.
airflow-init-1  | [2025-07-11T02:32:56.588+0000] {migration.py:216} INFO - Will assume transactional DDL.
airflow-init-1  | INFO  [alembic.runtime.migration] Context impl PostgresqlImpl.
airflow-init-1  | INFO  [alembic.runtime.migration] Will assume transactional DDL.
airflow-init-1  | INFO  [alembic.runtime.migration] Running stamp_revision  -> 405de8318b3a
airflow-init-1  | WARNI [airflow.models.crypto] empty cryptography key - values will not be stored encrypted.
airflow-init-1  | Initialization done
```

Start all services (MinIO, Reranker API, Airflow, PostgreSQL) in detached/background mode

```
docker compose up -d
```

```
[+] Building 2.1s (14/14) FINISHED                              docker:desktop-linux
 => [reranker-api internal] load build definition from Dockerfile          0.0s
 => => transferring dockerfile: 562B                                       0.0s
 => [reranker-api internal] load metadata for docker.io/library/python:3.  1.6s
 => [reranker-api auth] library/python:pull token for registry-1.docker.i  0.0s
 => [reranker-api internal] load .dockerignore                             0.0s
 => => transferring context: 2B                                            0.0s
```

```
[+] Running 7/7
 ✔ Volume "reranker_pipeline_demo_minio_data"              Created         0.0s
 ✔ Container reranker_pipeline_demo-minio-1                Started         0.6s
 ✔ Container reranker_pipeline_demo-reranker-api-1         Started         0.6s
 ✔ Container reranker_pipeline_demo-postgres-1             Healthy         1.4s
 ✔ Container reranker_pipeline_demo-airflow-init-1         Exited          6.4s
 ✔ Container reranker_pipeline_demo-airflow-scheduler-1    Started         0.6s
 ✔ Container reranker_pipeline_demo-airflow-webserver-1    Started         6.6s
```

List all running containers, their statuses, ports, and resource usage

```
docker compose ps
```

```
NAME                                      IMAGE                  COMMAND                  SERVICE             CREATED              STATUS                             PORTS
reranker_pipeline_demo-airflow-scheduler-1   apache/airflow:2.7.3   "/usr/bin/dumb-init …"   airflow-scheduler   36 seconds ago       Up 35 seconds                      8080/tcp
reranker_pipeline_demo-airflow-webserver-1   apache/airflow:2.7.3   "/usr/bin/dumb-init …"   airflow-webserver   36 seconds ago       Up 29 seconds (health: starting)   0.0.0.0:8080->8080/tcp
reranker_pipeline_demo-minio-1               minio/minio:latest     "/usr/bin/docker-ent…"   minio               36 seconds ago       Up 35 seconds                      0.0.0.0:9000-9001->9000-9001/tcp
reranker_pipeline_demo-postgres-1            postgres:13            "docker-entrypoint.s…"   postgres            About a minute ago   Up About a minute (healthy)        5432/tcp
```

Provides real-time metrics for all running containers

```
docker stats
```

```
CONTAINER ID   NAME                                      CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O         PIDS
b90d4425cef0   reranker_pipeline_demo-minio-1            0.00%     217.9MiB / 7.654GiB   2.78%     12.4kB / 4.19kB   38.7MB / 41kB     16
a59f4315353b   reranker_pipeline_demo-reranker-api-1     0.38%     544.2MiB / 7.654GiB   6.94%     93.7MB / 1.85MB   2.55MB / 269MB    40
59545d336792   reranker_pipeline_demo-postgres-1         1.10%     31.53MiB / 7.654GiB   0.40%     1.53MB / 2.08MB   3.58MB / 1.59MB   12
57949c87472a   reranker_pipeline_demo-airflow-scheduler-1   2.56%     401MiB / 7.654GiB     5.12%     780kB / 941kB     3.47MB / 39.6MB   73
c27c3a0bd25d   reranker_pipeline_demo-airflow-webserver-1   0.11%     574.4MiB / 7.654GiB   7.33%     933kB / 396kB     0B / 103MB        9
```

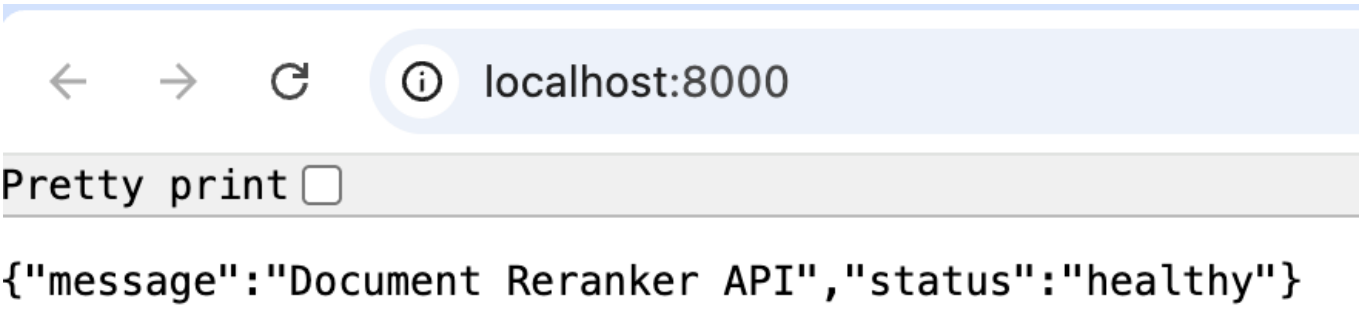
Containers statuses can also be seen in Docker Desktop

The following services are launched in separate containers:

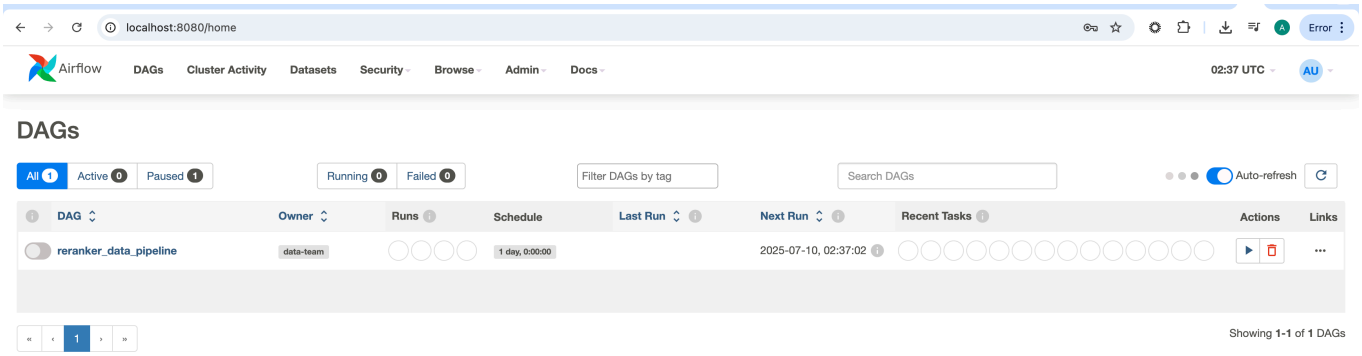
- Reranker API (FastAPI) — available at http://localhost:8000
- Airflow UI — available at http://localhost:8080, login: `admin/admin`
- MinIO Object Store — http://localhost:9001, login: `minioadmin/minioadmin`
- PostgreSQL — used by Airflow backend and logs

- Airflow Scheduler & Webserver — for running and monitoring DAGs

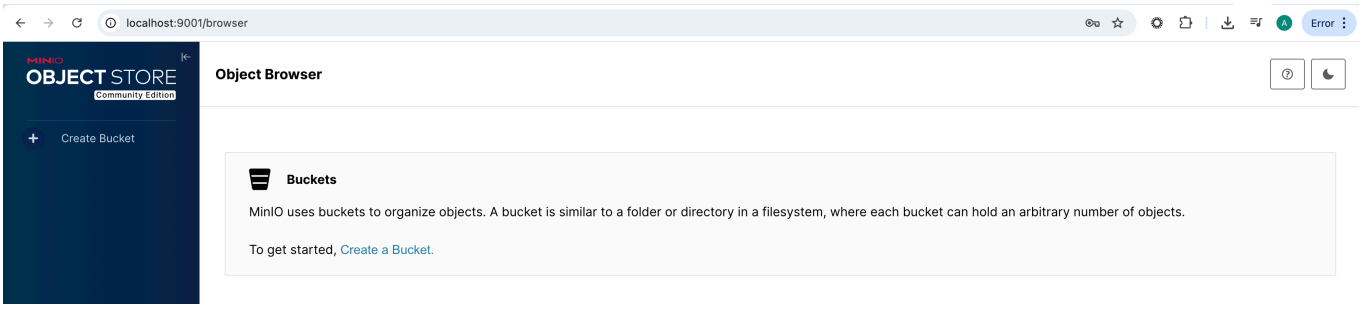
Reranker API



Airflow UI



MinIO Object Store



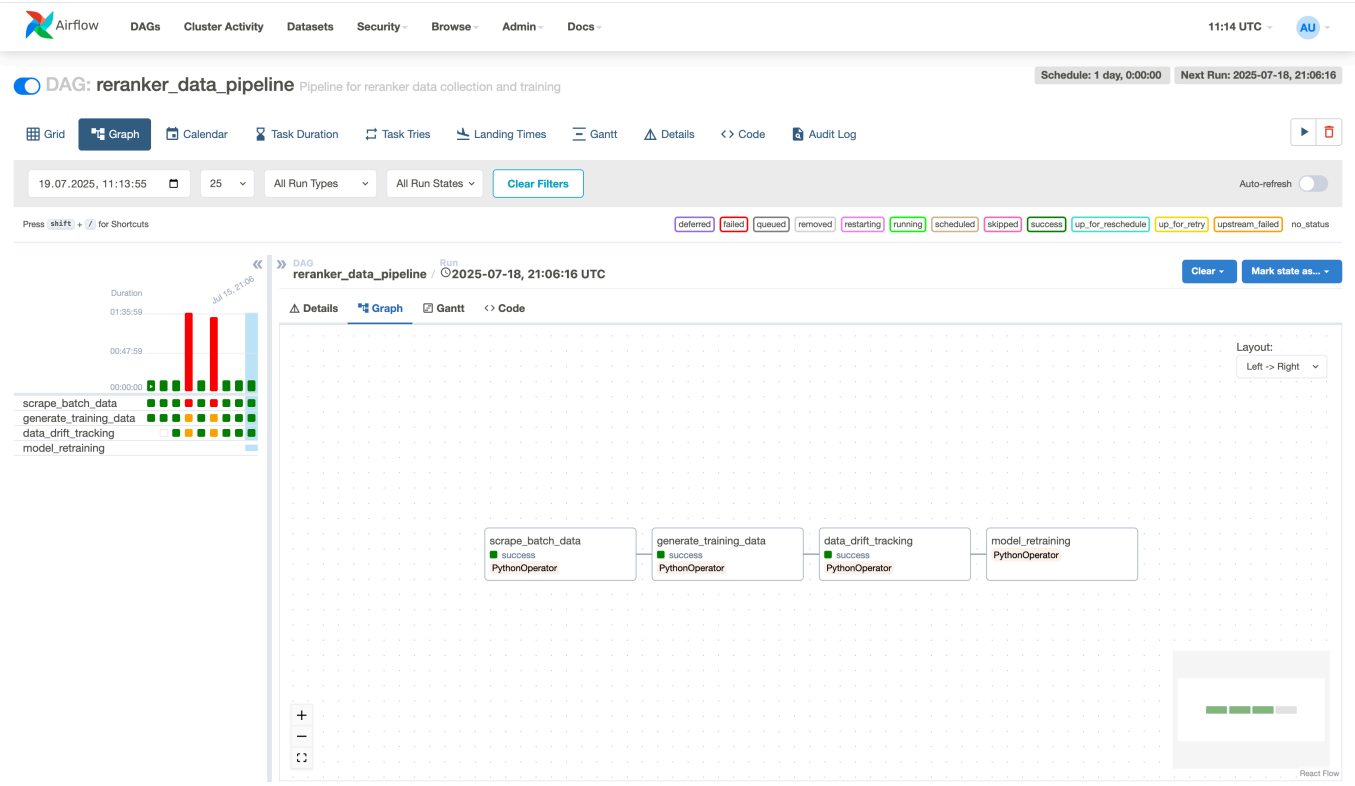
Just commands that have been used 100500 times to debug and restart everything

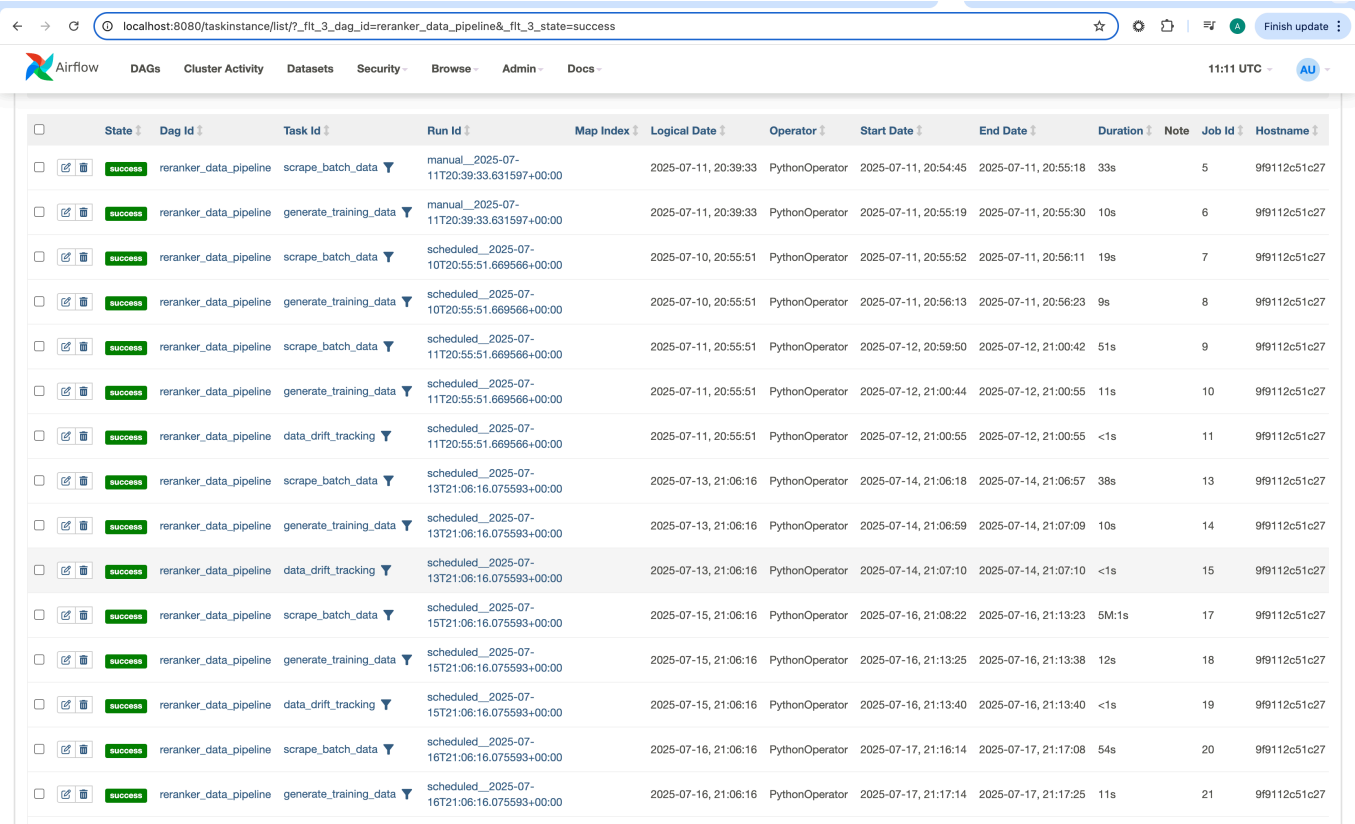
```
docker compose down

docker compose build reranker-api

docker compose up -d
```

```
docker compose down -v

docker compose build --no-cache

docker compose up airflow-init
```

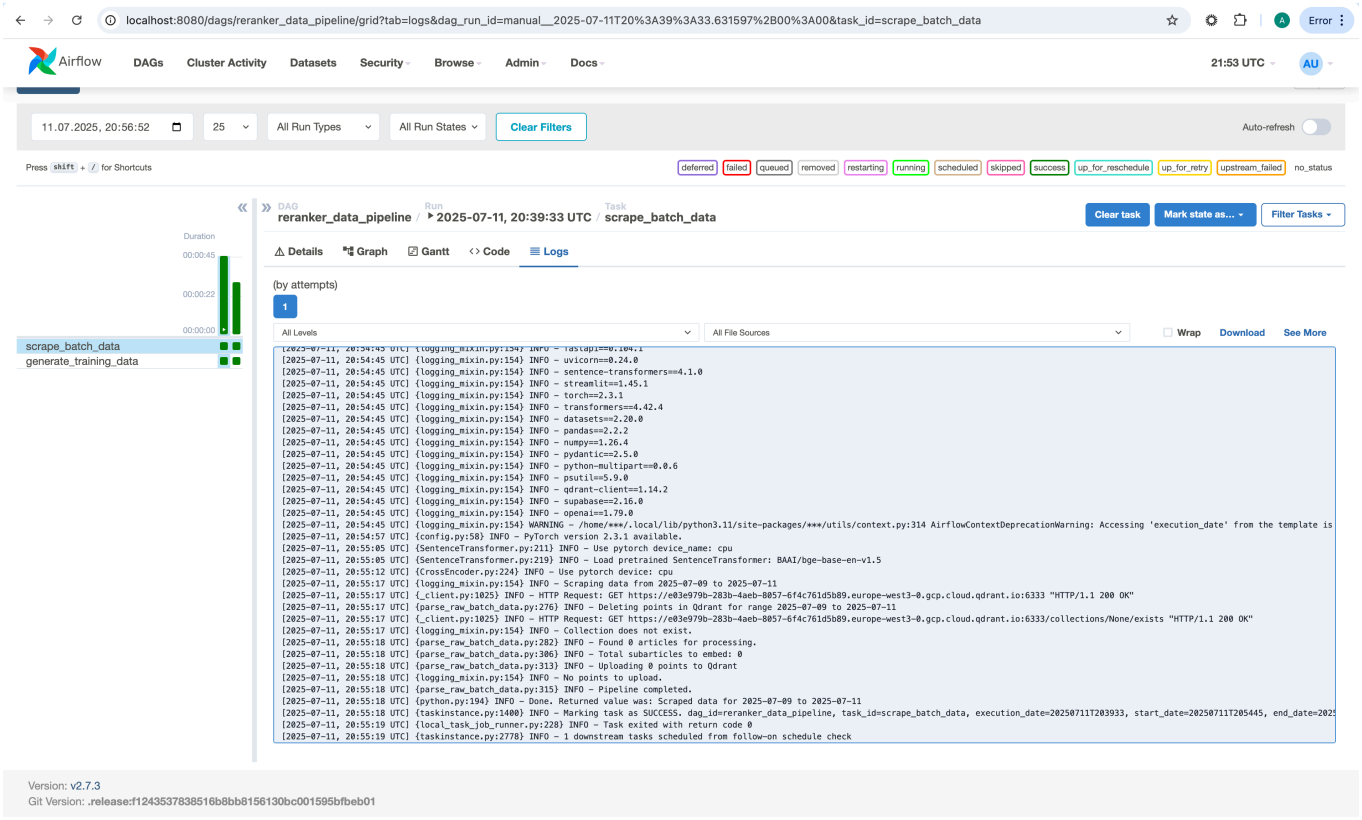
```
docker compose up -d
```

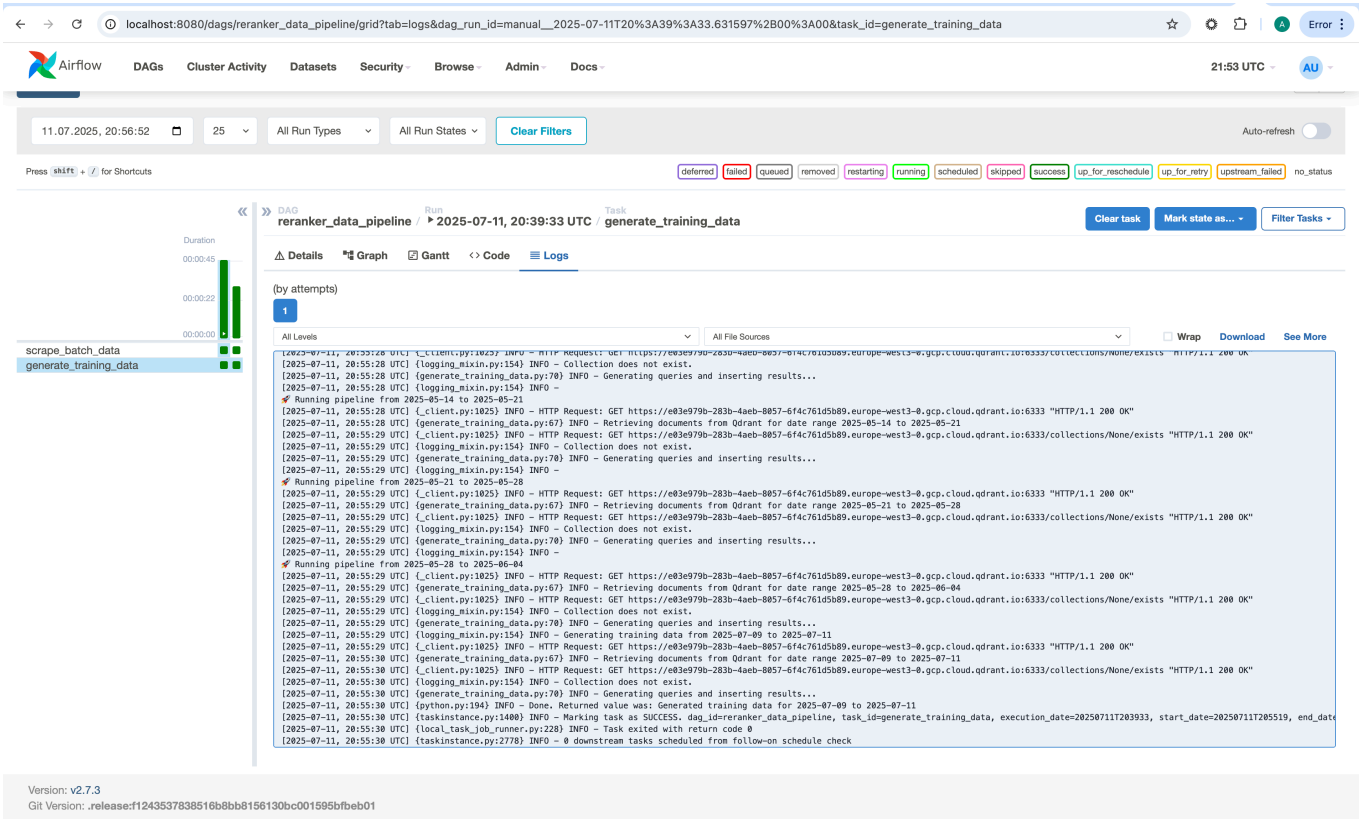
Data Pipeline Logs





State	Dag Id	Task Id	Run Id	Map Index	Logical Date	Operator	Start Date	End Date	Duration	Note	Job Id	Hostname
success	reranker_data_pipeline	scrape_batch_data	manual__2025-07-11T20:39:33.631597+00:00		2025-07-11, 20:39:33	PythonOperator	2025-07-11, 20:54:45	2025-07-11, 20:55:18	33s		5	9f9112c51c27
success	reranker_data_pipeline	generate_training_data	manual__2025-07-11T20:39:33.631597+00:00		2025-07-11, 20:39:33	PythonOperator	2025-07-11, 20:55:19	2025-07-11, 20:55:30	10s		6	9f9112c51c27
success	reranker_data_pipeline	scrape_batch_data	scheduled__2025-07-10T20:55:51.669566+00:00		2025-07-10, 20:55:51	PythonOperator	2025-07-11, 20:55:52	2025-07-11, 20:56:11	19s		7	9f9112c51c27
success	reranker_data_pipeline	generate_training_data	scheduled__2025-07-10T20:55:51.669566+00:00		2025-07-10, 20:55:51	PythonOperator	2025-07-11, 20:56:13	2025-07-11, 20:56:23	9s		8	9f9112c51c27
success	reranker_data_pipeline	scrape_batch_data	scheduled__2025-07-11T20:55:51.669566+00:00		2025-07-11, 20:55:51	PythonOperator	2025-07-12, 20:59:50	2025-07-12, 21:00:42	51s		9	9f9112c51c27
success	reranker_data_pipeline	generate_training_data	scheduled__2025-07-11T20:55:51.669566+00:00		2025-07-11, 20:55:51	PythonOperator	2025-07-12, 21:00:44	2025-07-12, 21:00:55	11s		10	9f9112c51c27
success	reranker_data_pipeline	data_drift_tracking	scheduled__2025-07-11T20:55:51.669566+00:00		2025-07-11, 20:55:51	PythonOperator	2025-07-12, 21:00:55	2025-07-12, 21:00:55	<1s		11	9f9112c51c27
success	reranker_data_pipeline	scrape_batch_data	scheduled__2025-07-13T21:06:16.075593+00:00		2025-07-13, 21:06:16	PythonOperator	2025-07-14, 21:06:18	2025-07-14, 21:06:57	38s		13	9f9112c51c27
success	reranker_data_pipeline	generate_training_data	scheduled__2025-07-13T21:06:16.075593+00:00		2025-07-13, 21:06:16	PythonOperator	2025-07-14, 21:06:59	2025-07-14, 21:07:09	10s		14	9f9112c51c27
success	reranker_data_pipeline	data_drift_tracking	scheduled__2025-07-13T21:06:16.075593+00:00		2025-07-13, 21:06:16	PythonOperator	2025-07-14, 21:07:10	2025-07-14, 21:07:10	<1s		15	9f9112c51c27
success	reranker_data_pipeline	scrape_batch_data	scheduled__2025-07-15T21:06:16.075593+00:00		2025-07-15, 21:06:16	PythonOperator	2025-07-16, 21:08:22	2025-07-16, 21:13:23	5M:1s		17	9f9112c51c27
success	reranker_data_pipeline	generate_training_data	scheduled__2025-07-15T21:06:16.075593+00:00		2025-07-15, 21:06:16	PythonOperator	2025-07-16, 21:13:25	2025-07-16, 21:13:38	12s		18	9f9112c51c27
success	reranker_data_pipeline	data_drift_tracking	scheduled__2025-07-15T21:06:16.075593+00:00		2025-07-15, 21:06:16	PythonOperator	2025-07-16, 21:13:40	2025-07-16, 21:13:40	<1s		19	9f9112c51c27
success	reranker_data_pipeline	scrape_batch_data	scheduled__2025-07-16T21:06:16.075593+00:00		2025-07-16, 21:06:16	PythonOperator	2025-07-17, 21:16:14	2025-07-17, 21:17:08	54s		20	9f9112c51c27
success	reranker_data_pipeline	generate_training_data	scheduled__2025-07-16T21:06:16.075593+00:00		2025-07-16, 21:06:16	PythonOperator	2025-07-17, 21:17:14	2025-07-17, 21:17:25	11s		21	9f9112c51c27

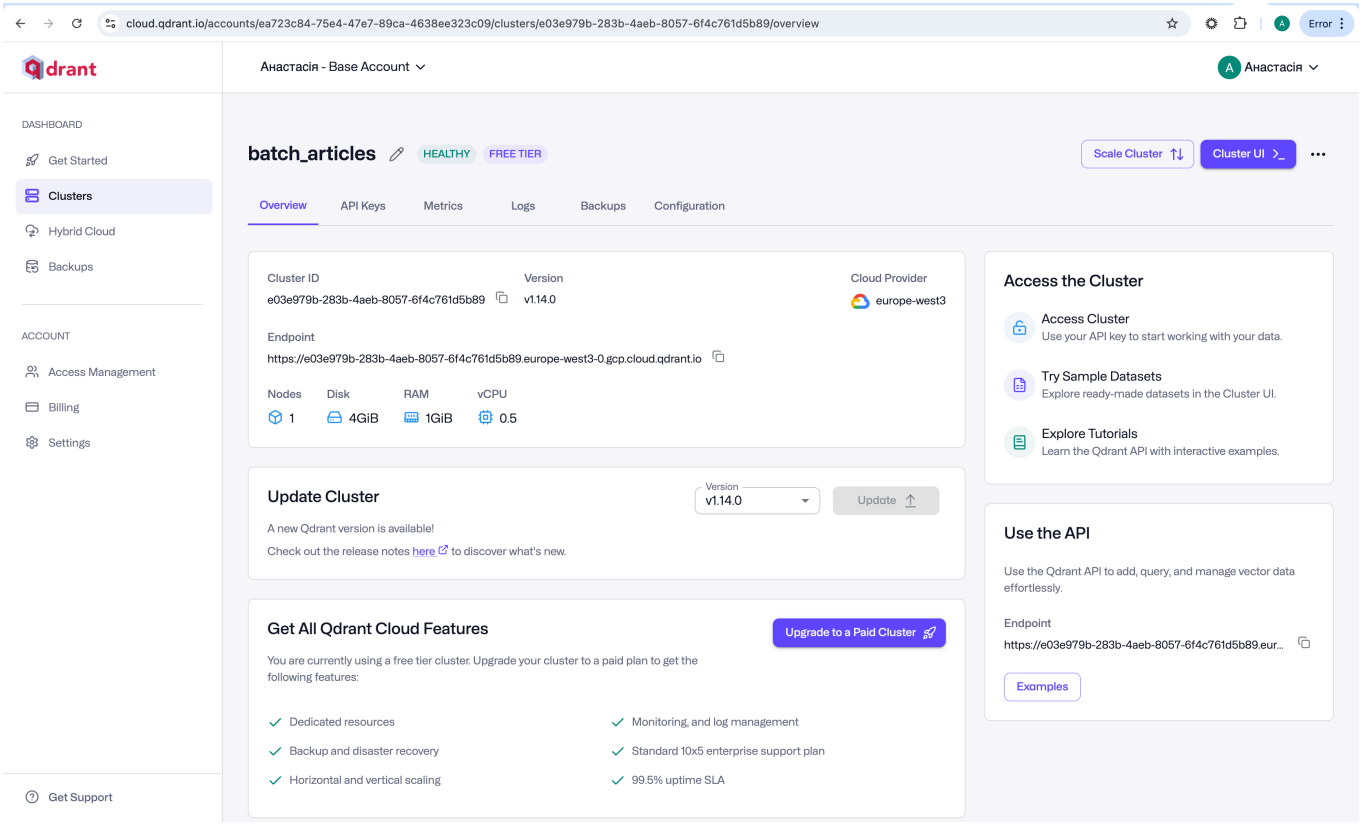




Data Storages Overview

In this project, two types of storages were used to support the RAG pipeline — Qdrant and Supabase. By separating responsibilities — unstructured retrieval to Qdrant and structured storage to Supabase — the system was made more modular and easier to maintain or scale.

Qdrant was selected for storing dense vector embeddings and enabling fast semantic similarity search.

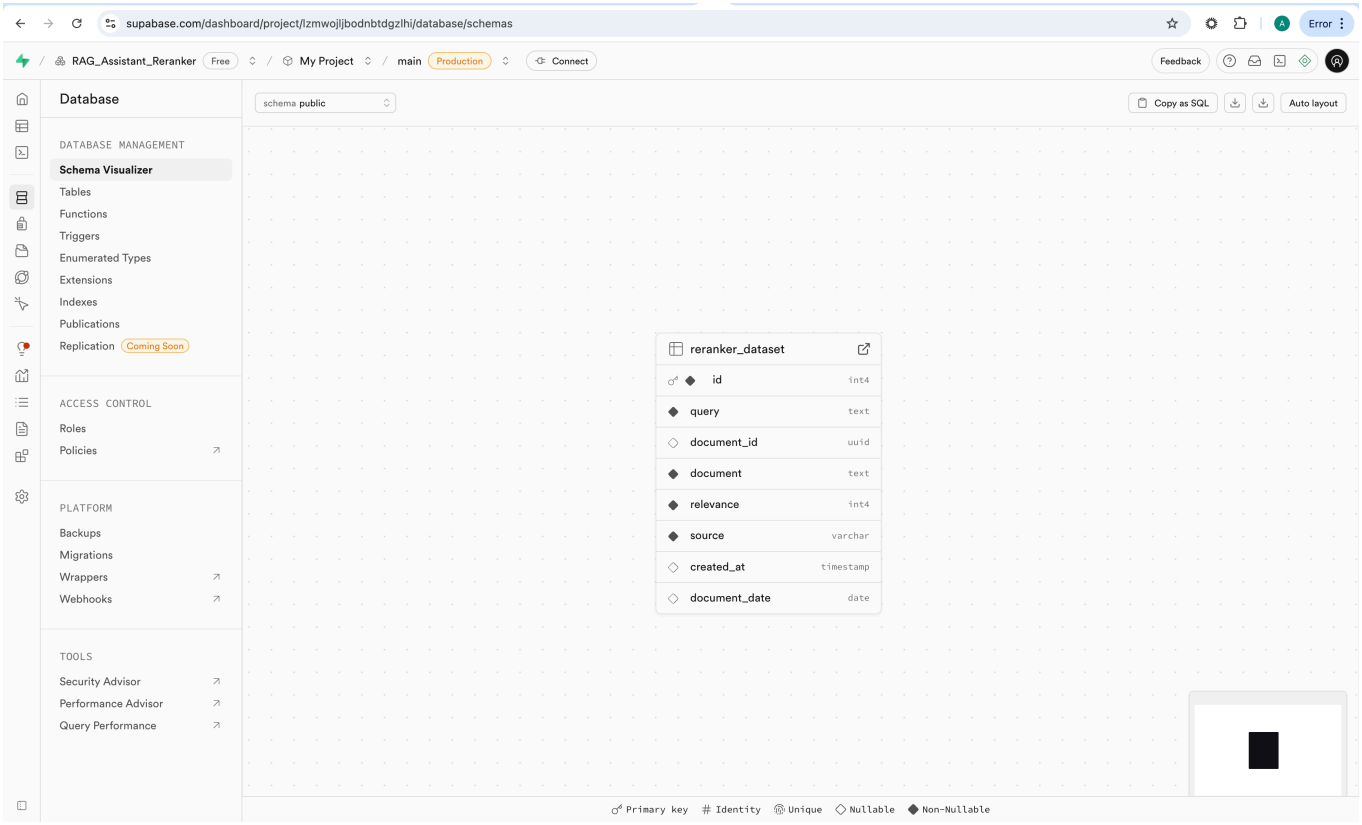


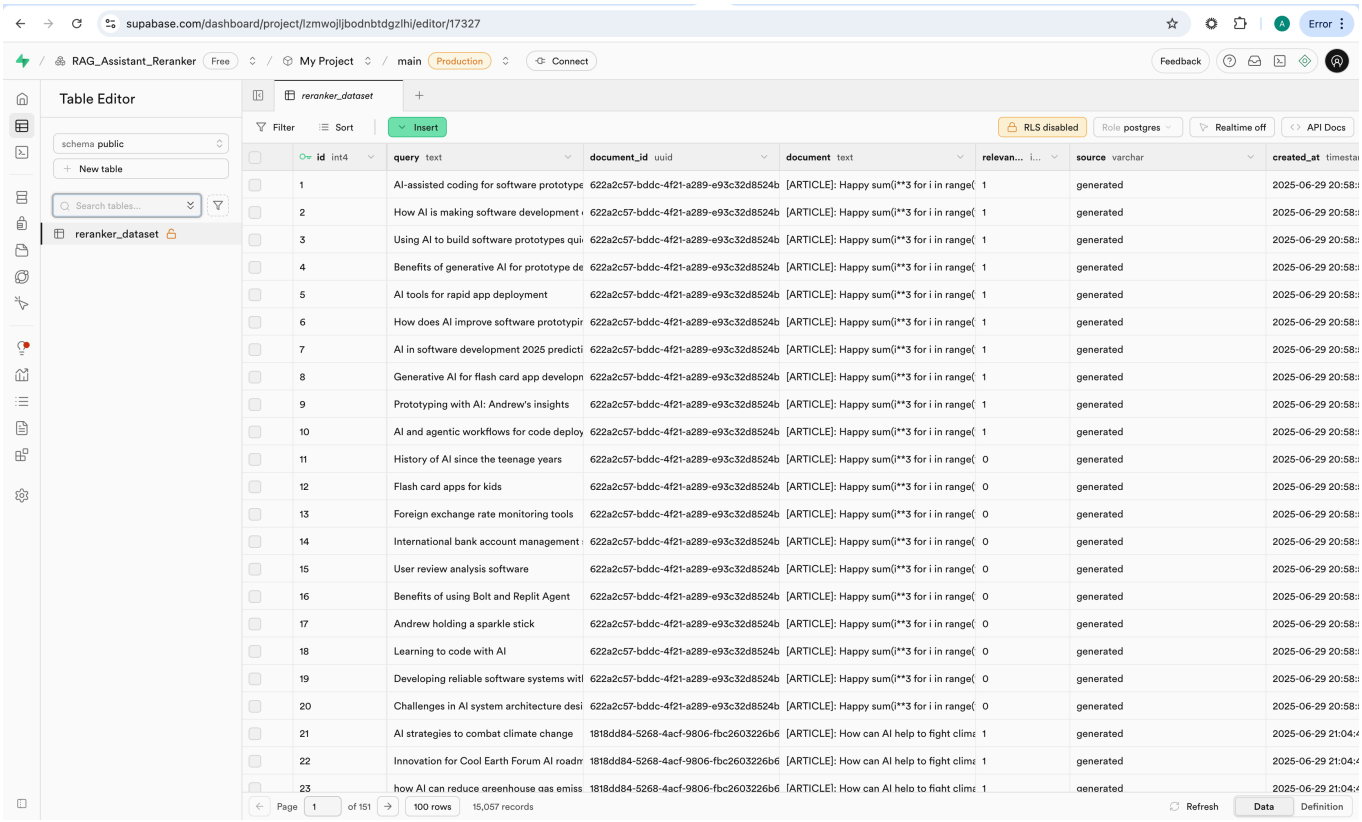
This Qdrant collection stores vectorized documents along with metadata in the payload.

```
Qdrant Point
├── id : string
├── vector : list[float]
└── payload
    ├── title : string
    ├── type : string
    ├── date : string
    ├── date_int : int
    ├── url : string
    ├── text : string
    ├── image
    │   ├── image_url : string
    │   ├── image_caption : string
    │   └── image_embedding : list[float]
    └── combined_text : string
```

- Text embeddings are used for semantic search.
- Metadata fields help with filtering or advanced ranking.
- Image-related fields allow for multimodal querying or reranking.

Supabase, built on PostgreSQL, was used to store structured information such as annotated samples of training data.



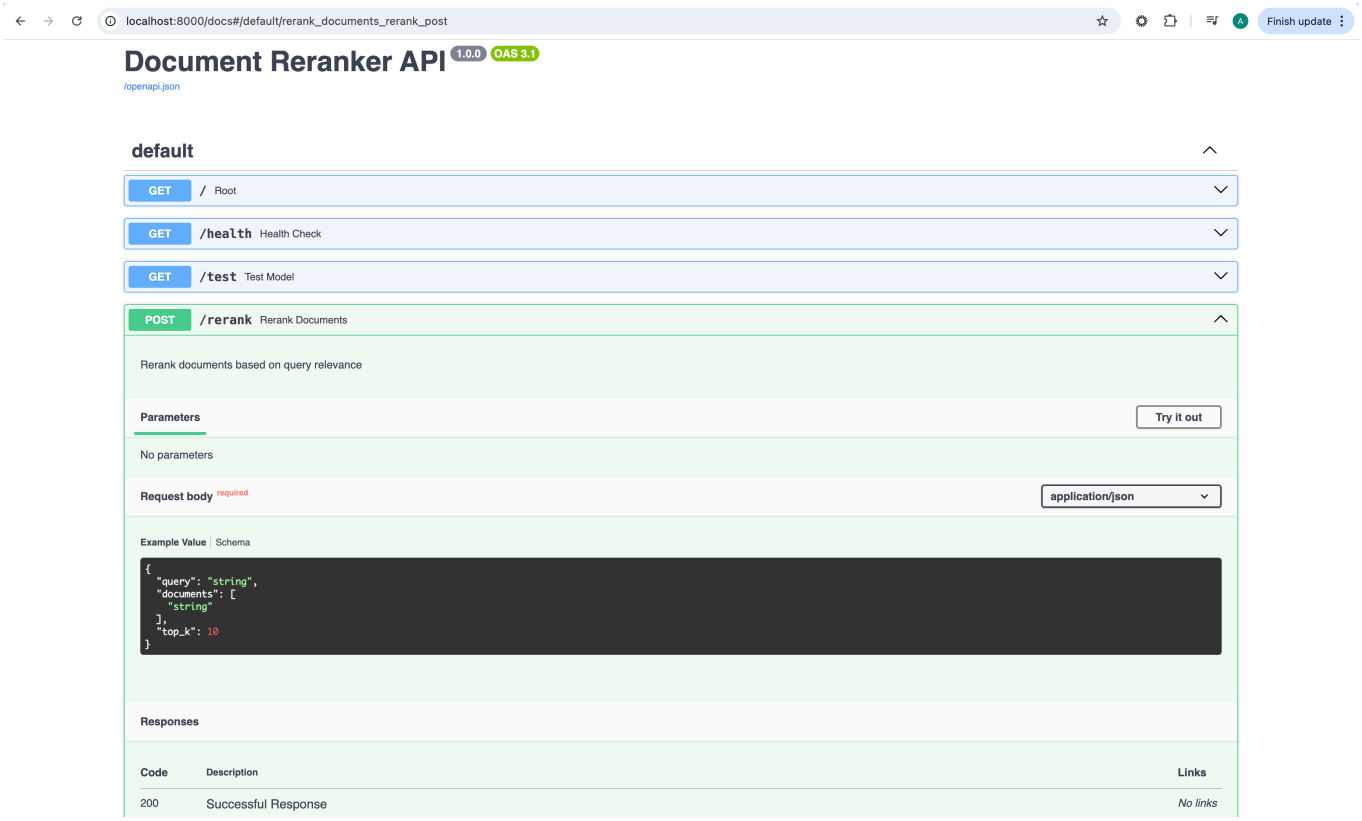


Model API Demo Preparings

Check if API is actually available

```
docker compose logs reranker-api
```

```
reranker-api-1  | INFO:     Started server process [1]
reranker-api-1  | INFO:     Waiting for application startup.
reranker-api-1  | INFO:reranker_fast_api_app:No trained model found, using base model
Downloading config.json: 100%|██████████| 385/385 [00:00<00:00, 114kB/s]
Downloading (…)173ca8f8edc65152fae7: 100%|██████████| 133M/133M [00:13<00:00, 9.97MB/s]
reranker-api-1  | Some weights of BertForSequenceClassification were not initialized from the model checkpoint at microsoft/MiniLM-L12-H384-uncased and are newly initialized: ['classifier.weight', 'classifier.bias'
]
reranker-api-1  | You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
Downloading tokenizer_config.json: 100%|██████████| 2.00/2.00 [00:00<00:00, 8.37kB/s]
Downloading vocab.txt: 232kB [00:00, 2.73MB/s]
Downloading (…)cial_tokens_map.json: 100%|██████████| 112/112 [00:00<00:00, 493kB/s]
reranker-api-1  | INFO:sentence_transformers.cross_encoder.CrossEncoder:Use pytorch device: cpu
reranker-api-1  | INFO:reranker_fast_api_app:Model loaded successfully
reranker-api-1  | INFO:     Application startup complete.
reranker-api-1  | INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
```



Specific point requests to check everything one more time

```
/Users/anastasiiamazur/PycharmProjects/rag-news-assistant/rag_assistant3.11/bin/python /Users/anastasiiamazur/PycharmProjects/rag-news-assistant/api_test.py
INFO:__main__:Run http://localhost:8000/test: {'status': 'success', 'test_score': 0.48815932869911194}
INFO:__main__:Run http://localhost:8000/rerank: {'results': [{'document': 'Deep learning models require large datasets', 'score': 0.48848041892051697, 'original_index': 0, 'rank': 1}, {'document': 'Neural
INFO:__main__:Run http://localhost:8000/score: {'scores': [0.4879770278930664]}

Process finished with exit code 0
```

Create virtual environment to install `requests` library and run demo from terminal

```
python3 -m venv venv
source venv/bin/activate
pip install requests
```

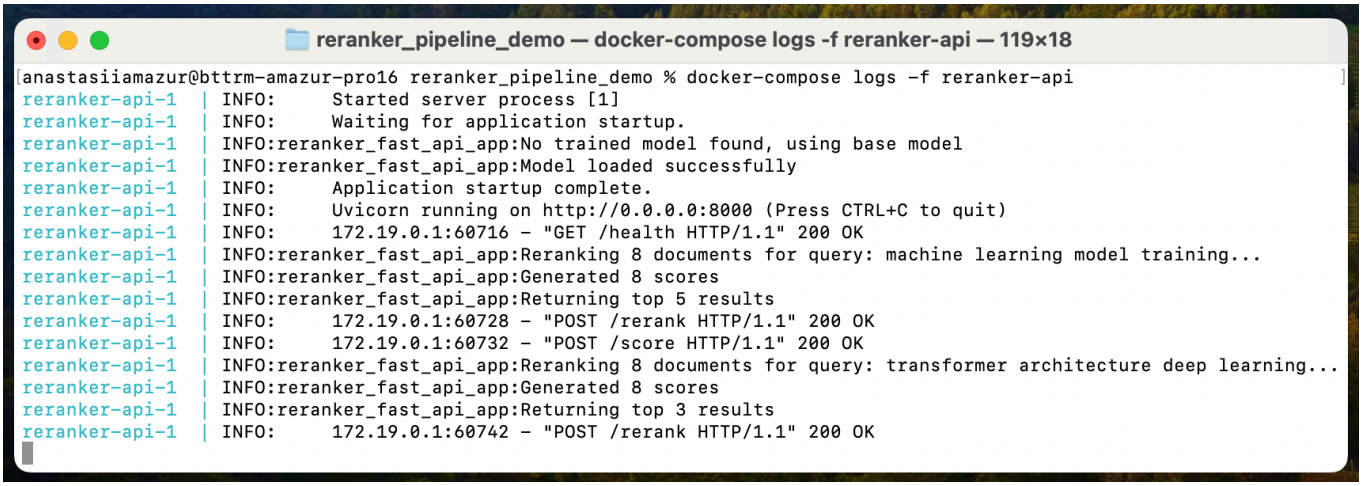
Setup realtime logs checker

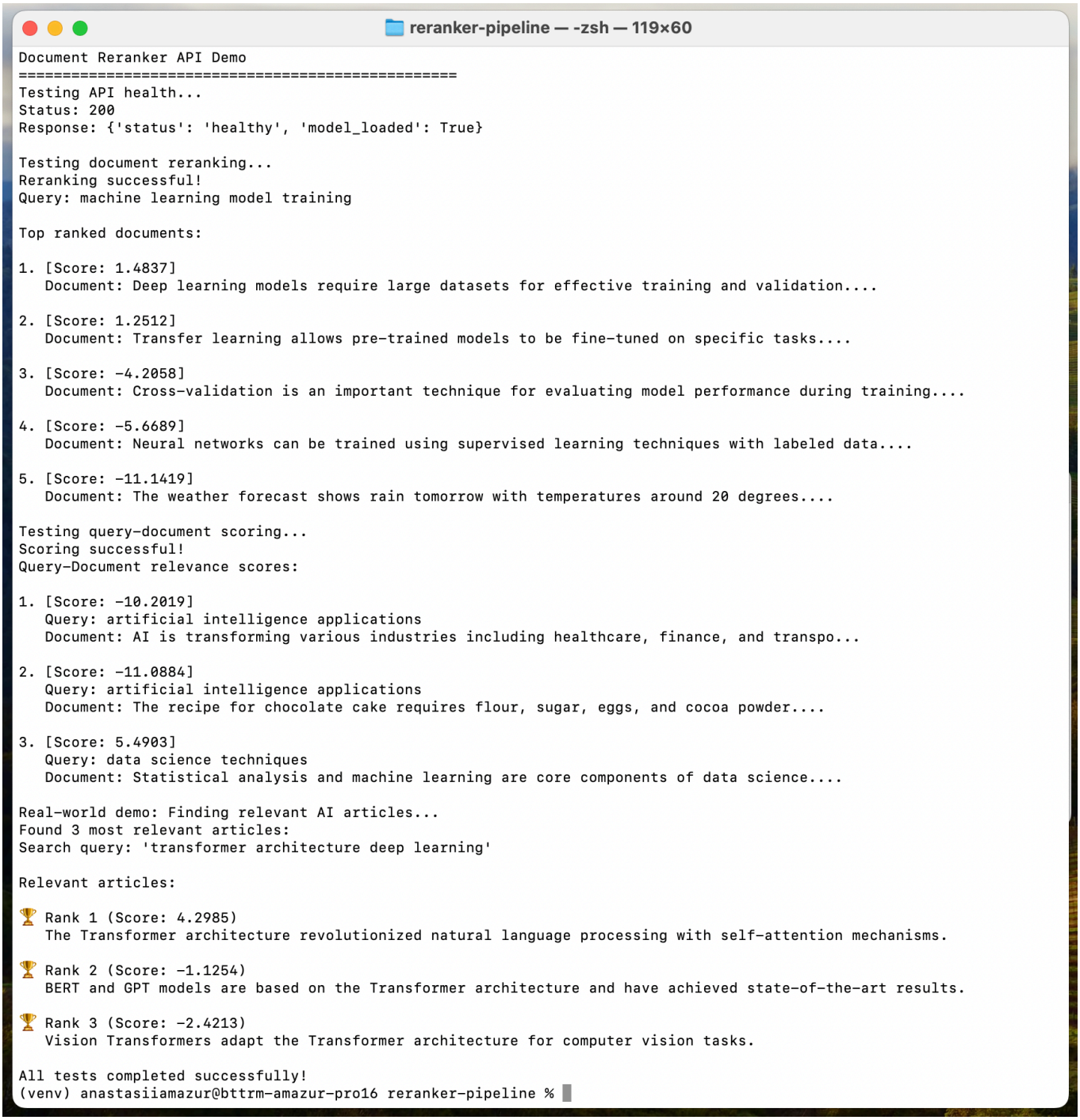
```
docker-compose logs -f reranker-api
```

Model API Demo

Run demo from venv

```
python demo.py
```


reranker_pipeline_demo — docker-compose logs -f reranker-api — 119×18
anastasiiamazur@bttrm-amazur-pro16 reranker_pipeline_demo % docker-compose logs -f reranker-api
reranker-api-1 | INFO: Started server process [1]
reranker-api-1 | INFO: Waiting for application startup.
reranker-api-1 | INFO:reranker_fast_api_app:No trained model found, using base model
reranker-api-1 | INFO:reranker_fast_api_app:Model loaded successfully
reranker-api-1 | INFO: Application startup complete.
reranker-api-1 | INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
reranker-api-1 | INFO: 172.19.0.1:60716 - "GET /health HTTP/1.1" 200 OK
reranker-api-1 | INFO:reranker_fast_api_app:Reranking 8 documents for query: machine learning model training...
reranker-api-1 | INFO:reranker_fast_api_app:Generated 8 scores
reranker-api-1 | INFO:reranker_fast_api_app:Returning top 5 results
reranker-api-1 | INFO: 172.19.0.1:60728 - "POST /rerank HTTP/1.1" 200 OK
reranker-api-1 | INFO: 172.19.0.1:60732 - "POST /score HTTP/1.1" 200 OK
reranker-api-1 | INFO:reranker_fast_api_app:Reranking 8 documents for query: transformer architecture deep learning...
reranker-api-1 | INFO:reranker_fast_api_app:Generated 8 scores
reranker-api-1 | INFO:reranker_fast_api_app:Returning top 3 results
reranker-api-1 | INFO: 172.19.0.1:60742 - "POST /rerank HTTP/1.1" 200 OK


reranker-pipeline — -zsh — 119×60
Document Reranker API Demo
==
Testing API health...
Status: 200
Response: {'status': 'healthy', 'model_loaded': True}
Testing document reranking...
Reranking successful!
Query: machine learning model training
Top ranked documents:
1. [Score: 1.4837]
Document: Deep learning models require large datasets for effective training and validation....
2. [Score: 1.2512]
Document: Transfer learning allows pre-trained models to be fine-tuned on specific tasks....
3. [Score: -4.2058]
Document: Cross-validation is an important technique for evaluating model performance during training....
4. [Score: -5.6689]
Document: Neural networks can be trained using supervised learning techniques with labeled data....
5. [Score: -11.1419]
Document: The weather forecast shows rain tomorrow with temperatures around 20 degrees....
Testing query-document scoring...
Scoring successful!
Query-Document relevance scores:
1. [Score: -10.2019]
Query: artificial intelligence applications
Document: AI is transforming various industries including healthcare, finance, and transpo...
2. [Score: -11.0884]
Query: artificial intelligence applications
Document: The recipe for chocolate cake requires flour, sugar, eggs, and cocoa powder....
3. [Score: 5.4903]
Query: data science techniques
Document: Statistical analysis and machine learning are core components of data science....
Real-world demo: Finding relevant AI articles...
Found 3 most relevant articles:
Search query: 'transformer architecture deep learning'
Relevant articles:
🏆 Rank 1 (Score: 4.2985)
The Transformer architecture revolutionized natural language processing with self-attention mechanisms.
🏆 Rank 2 (Score: -1.1254)
BERT and GPT models are based on the Transformer architecture and have achieved state-of-the-art results.
🏆 Rank 3 (Score: -2.4213)
Vision Transformers adapt the Transformer architecture for computer vision tasks.
All tests completed successfully!
(venv) anastasiiamazur@bttrm-amazur-pro16 reranker-pipeline %