ETL TECHNICAL WALKTHROUGH

Presentation by Chananan Srasri

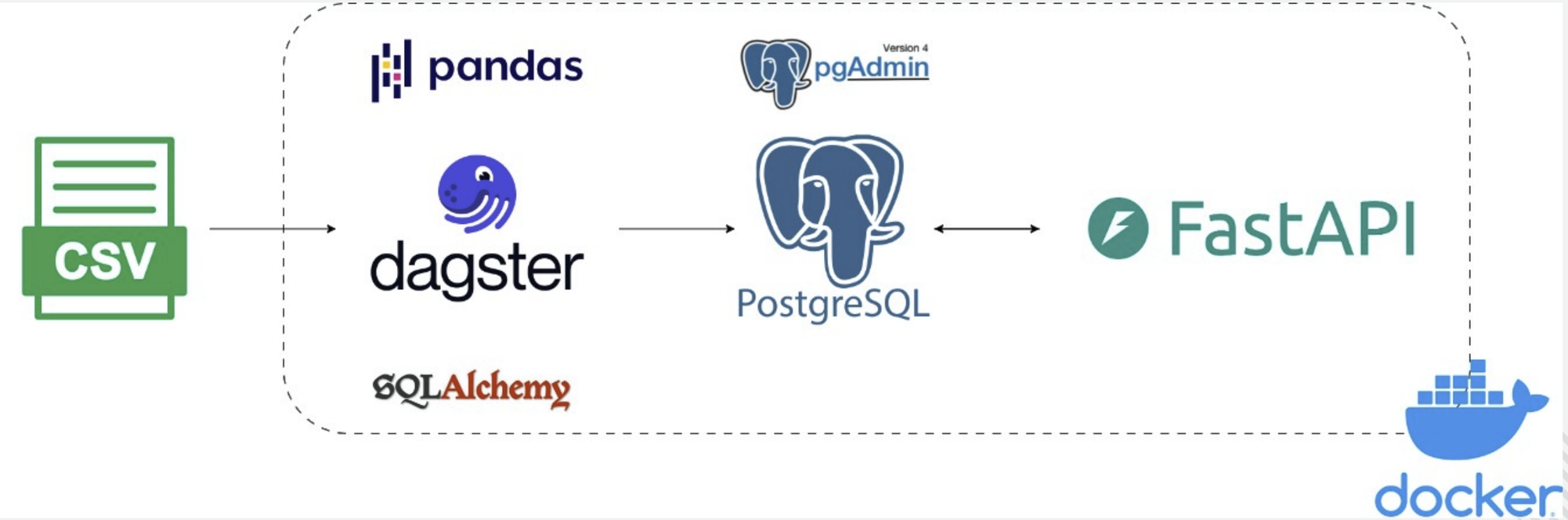
Thinking Machine Interview

Data Engineer

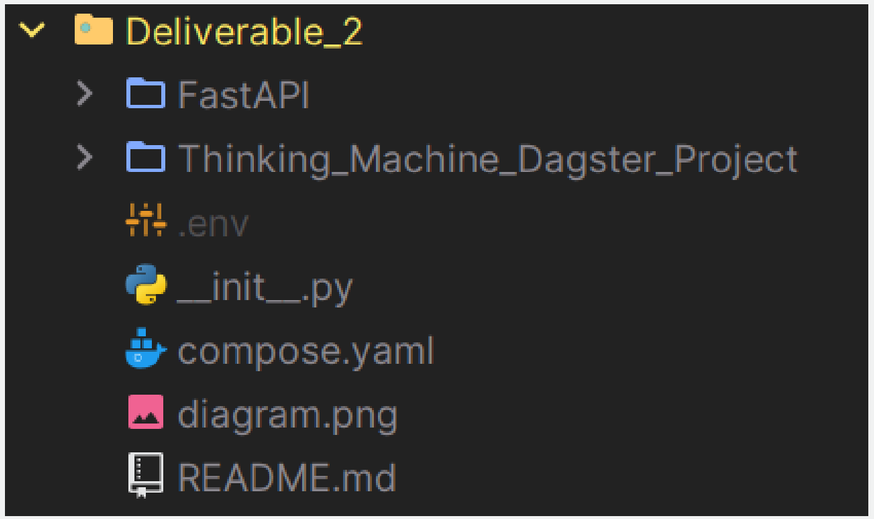
INTRODUCTION

This ETL project is built based on **Python** mainly with some SQL elements. The project aims to **clean the CSV data** and load it into the database. Also, FastAPI, a web service, returns the check-in data associated with a given user.

TECHNOLOGY STACK AND DIAGRAM



PROJECT STRUCTURE



This project will consist of 2 main services

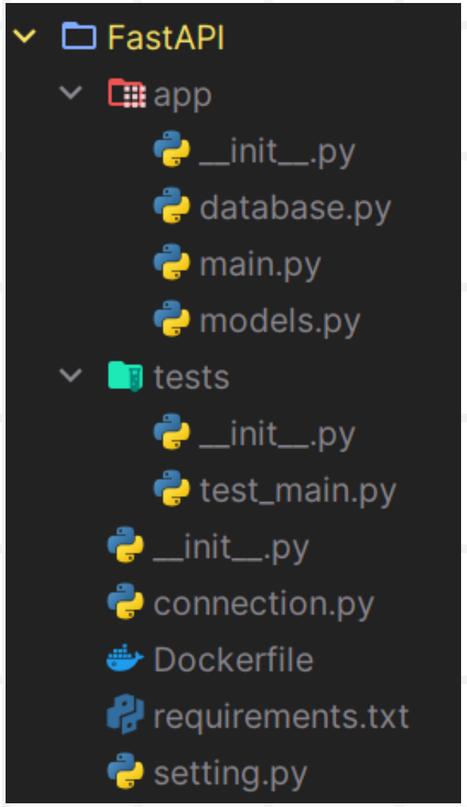
- FastAPI
- Dagster

Along with compose.yaml, README, .env

PROJECT STRUCTURE

In FastAPI's directory

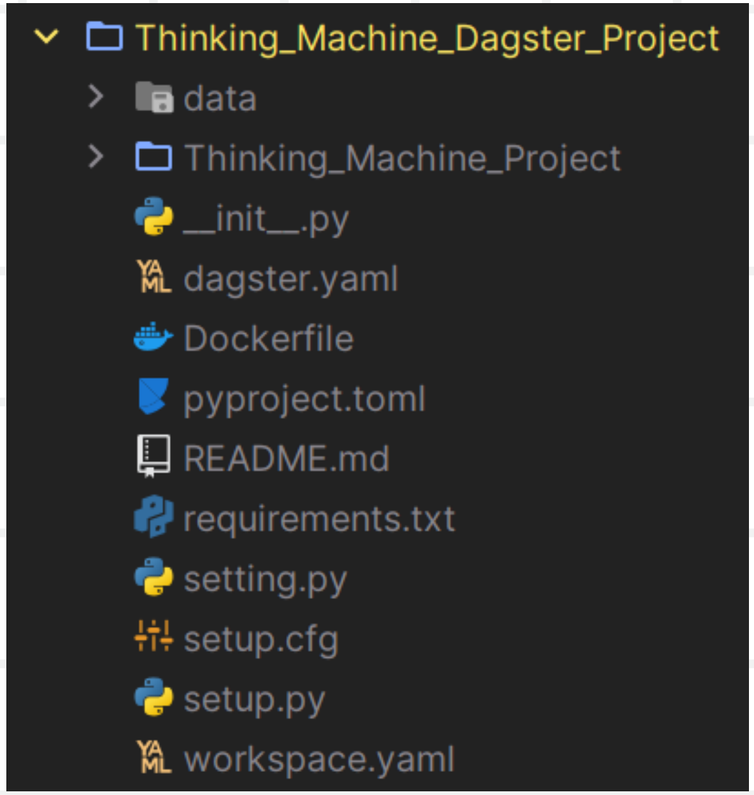
- **app**: store main app
- **tests**: store test for the main app
- **connection**.py and **setting**.py is for SQLAlchemy connection and environment variables
- **Dockerfile**: to create image for this app



PROJECT STRUCTURE

In Dagster's directory

- **data**: input and output
- **_Project**: Definition, Assets with tests
- **dagter.yaml**, **workspace.yaml**: config for Dagster
- **Dockerfile**: to create image for this app



FEATURES SUPPORTED

INGESTION

Ingest the source csv file into database

CONTAINERIZED

All app will included in docker compose

DAGSTER

A pipeline orchestration tool to manage assets

FEATURES SUPPORTED

PGADMIN4

To easily manage or query the result table

FASTAPI

As a web service that returns the data to user

UNITTEST

Easily test the FastAPI and the result table

THOUGHT PROCESS

DATA ORCHESTRATION TOOLS

There are many data orchestration tools nowadays.
But the one I use as a daily driver is Dagster.
Even though the version I've been using is 1.0.17.
But the recent version is 1.7.0. So I decided to try something new before I finish my migration project for my present company.

DATABASE OF CHOICE

DuckDB is trending right now. I want to try it for this project too.
But I can't deny the fact like previous data orchestration tools choice. PostgreSQL is also more mature and well-documented.
So it's easier to find some solution than DuckDB for sure.

THOUGHT PROCESS

API FRAMEWORK

FastAPI is fast and easy to deploy. It’s packed with a built-in test function to easily create the unit test without extra tools.

SQLALCHEMY ORM MODEL

SQLAlchemy ORM model is great when it’s paired with FastAPI. So I decided to use it with Dagster’s asset too. Which resulted me a hard time creating a timestamp column with the default value.

THOUGHT PROCESS

TIMEZONE CLEANING

It’s taken me a while to transform the Russian timestamp to the UTC timestamp. Then I found dateparse library to save my day

DOCKER WORKDIR

A workdir like /code might work fine when it’s only running a container. However, it can’t work if you’re trying to run python in your container. The code is also a python module which is conflict to our workdir.

THOUGHT PROCESS

DOCKER DESKTOP BUG

Can’t build a docker new service for unknown reasons. Turn out it’s a bug from the docker desktop and solved by upgrading or downgrading the docker desktop version

IMPLEMENTATION CONSIDERATIONS

SCHEDULE JOB

- If the source data has changed periodically. We must create a job to define a schedule insert periodically too.
- Need to add unique keys and create_at, modified_at column to track the record further.

UNIT TEST

- create a fully function unit test after understanding the data's boundary like max possible hour for each project.
- consider great expectations as a data quality tools.

IMPLEMENTATION CONSIDERATIONS

DEPLOY TO A CLOUD SERVICE

- setup other services for production deployment such as dagster-daemon to take care of sensors, backfill, schedule, and run queue
- add config for additional dagster instances
- create a proper schema in a database to store Dagster information
- might create an asset factory if this same type of work (ingest CSV files) to make this asset-creating process re-usable.

THANK YOU