Containerization As A Tool For Learning

Henning W.

May 24, 2021

1 Introduction

Choice of application: Some kind of social media platform thing. Due to large time constraints, some suboptimal solutions will have be taken. I usually would start off the project by making CI/CD with a server. These factors have been discarded entirely. This means, that there won't be any online capabilities in this project. Instead the entire application will be containerized, in an attempt at making the local solution easier to run. Another suboptimal solution is, that the database containers and the containers which utilize them have been split into two separate container files. You'd usually want both of them to share the same docker-compose file. The problem lies with migration strategies, and what I like calling "A chicken or egg" complication. Both data processing and my backend are reliant on the database. Further more, my backend is reliant on the data-processing. Because of this, the containers will need to be run in chunks. The standard solution to this is somewhat suboptimal in itself. It involves files which lets other processes run first before executing itself.

Since the project requires a larger amount of data, the project has been created a bit backwards. I've looked at some datasets, and then slowly formulated data models from them.

2 Goals

This section is dedicated to the following assignment definition question:

Define functional and non-functional requirements to your project

2.1 Implemented

-

2.2 Planned

- Gathering of datasets
- Create diagrams for database
- data_population: Processing of datasets to usable format
- data_population: Datasets inserted into Mongo with Cython

- data_population: connection tests
- data_population: logging
- backend: Postgres, Redis, Mongo connectors and tests
- backend: Generate Postgres entries from Mongo database
- backend: Redis caching
- frontend: Meilisearch functionality implemented

2.3 Stretch Goals

- Apply solution to Digital Ocean droplet
  - Create and setup server on digital ocean
  - docker-compose deployment version
  - CI with github actions.
  - github actions secrets with docker-compose
  - SSH SCP appleboy execution of docker-compose solution
- data_population: documentation with Sphinx
- backend: Documentation rustdocs

2.4 Scrapped

3 Installation