Documentation: Deployment of Web app with a MongoDB and a Redis cache

November 29, 2024

Introduction

This document explains the inner working of the `compose.yml` for the core app and it's relationship with the mongoDB and a Redis cache. In a way that this docker compose has 3 diferent microservices with a total of 4 containes. This is because the mongoDB has a container in charge of the set-up.

Docker Compose Explanation

App container

core-app

- **Image:** This service uses a modified image that streams from a `python:3.12`. To this container we are also going to add the python flask files used to run the app as well as the html files of the web. An then execute the command `python3 app.py` to initialize the python server.
- **Port:** "6000:5000" to publish the port 5000 of the python server and map it to the port 6000 of the host.

MongoDB container

mongodb

- **Image:** This service uses the `mongo:latest` image.
- **Restart:** This is set to alwais, so that if the container were to fail it would automatically restart the containes.
- **Environment:** These variables configure the mongo server:
  - MONGO_INITDB_ROOT_USERNAME: sets te username to "*app*".

- MONGO_INITDB_ROOT_PASSWORD: sets the password to connect the db to "*secret*".
- ME_CONFIG_MONGODB_URL: Sets the url that other containers will use to connect with this mongo server to "*mongodb://app:secret@mongo_db:27017*"
- ME_CONFIG_BASICAUTH: This variable is set to false

mongo-setup

- **Image:** This service uses the `mongo:latest` image.
- **Links:** This links this container with the `mongodb` container.
- **Volumes:** "`./mongodb:/mongodb`" stores the data to import to the mongodb as well as bash file to make the `mongoimport`.
- **Command:** This executes the bash file with the `mongoimport` running the command `/mongodb/mongo_setup.sh`.
- This container is used to set up the mongoDB and make a `mongoimport` of the dataset we need for the web to show data. This `mongoimport` isn't cooperating at the moment.

We tried to make a clustered mongoDB but due to it malfuncioning we opted to try first with a regular mongo container.

Redis container

redis-cache

- **Image:** This service uses the `redis:latest` image.
- **Helthcheck:** For this particular service we are going to be running a helthcheck so to ensure that redis is upp before starting the redis server.
- **Command:** Initialicing aredis server with the command `redi-server`