Docker Compose

Day	Wednesday
Tasks	Run docker compose for tweet_collector and mongoDB containers

🎯 Goal: run docker compose consisting of two containers

- Docker compose is where usefulness/beauty/convenience of docker actually starts to shine.
  - You are rarely going to want to build just a single service/container for something.
- Docker compose is a tool for defining and running multi-container docker applications.
- We define all containers/services in one file:
  - YAML file ("yet another markup language", "YAML ain't markup language")
  - nested, similar to JSON (but it's more readable)
- Inside the docker-compose.yml file you define *services,* each service is equivalent to one `docker run` line / should contain all information that `docker run` contains:
  - image (if building from the Dockerfile)
  - prebuilt image name (if building from a prebuilt image)
  - port mapping
  - volumes to mount
  - environment variables
  - + how they depend on each other

```
version: '3'

services: # here we define all our docker containers

  tweet_collector: # 1st container
    build: tweet_collector/ # Dockerfile is in the tweet_collector folder
    volumes:
    - ./tweet_collector/:/app # local_path:container_path
    depends_on:
    - mongodb

  mongodb: # 2nd container
    image: mongo # we run it directly
    ports:
    - 27017:27017 # local port : container port

  # etl_job: # 3rd container

  # postgres: # 4th container
```

```
project_folder/ # we run docker compose from this folder
|-- docker-compose.yml # this is the file we wrote above
|-- tweet_collector/
|   |-- Dockerfile
|   |-- requirements.txt
|   |-- tweet_collector.py
|   |-- config.py
|-- etl_job/
    |-- Dockerfile
    |-- requirements.txt
    |-- etl.py
```