Setting up the Protein database

Requirements:

1. PostgreSQL server instance
2. Conda (Anaconda/ Miniconda)
3. docker-compose

1. Set up the code base by cloning the repository:

   ```
   git clone https://github.com/BackofenLab/protein-graph-database.gitcd protein-graph-database/
   ```

2. Create credentials.yml file in the root directory of the cloned repository. The credentials file in the tests folder can be used as a template.

   ```
   ln -s tests/credentials.test.yml credentials.yml
   ```

3. Create a conda environment ‘pgdb’ to run the code. Libraries installed by the environment include
   - py2neo - client library and toolkit for working with Neo4j
   - psycopg2 -PostgreSQL database adapter
   - neomodel - An Object Graph Mapper (OGM) for the neo4j graph database, built on the neo4j_driver

   ```
   make env
   ```

4. Build docker containers for web and database

   ```
   docker build -f docker/web/Dockerfile -t <repository>/pgdb-web:<version> .
   ```

   ```
   docker build -f docker/database/Dockerfile -t <repository>/pgdb-database:<version> .
   ```

5. Start the containers*

   Docker-compose.yml (Version 2) defines the 2 services :

- database- runs from the pgdb-database container on ports 7474 (neo4j graph database), 7687 (bolt protocol).
- web- runs from the container pgdb-web on port 5000.

```
docker-compose up
```

*in case the terminal returns an error “Not enough disk on space”, run **docker system prune** to free up volumes.

6. Stop the containers

   Stop the service using Ctrl+C and docker compose down to stop and remove containers.

   ```
   docker-compose down
   ```

Note:
When using an IDE, start the neo4j database service with docker-compose run, and run the server from the IDE.

```
docker-compose run --publish 7474:7474 -p 7687:7687  database
```