Lab Report: Docker

Student information

- Student name: Xander Van der Linden
- Student code: 202292316

Assignment description

In dit Docker labo, moest ik een aantal Docker Images maken en optimalizeren voor een simpele web applicatie. Het uiteindelijke doel was om een Node.js API in een container te stoppen zodat deze bruikbaar was via poort 3000. De data werd opgeslaan aan de hand van volumes en werd geconnecteerd met een SQLite en MySQL Database. Ook werden deze images naar de Docker hub gepusht.

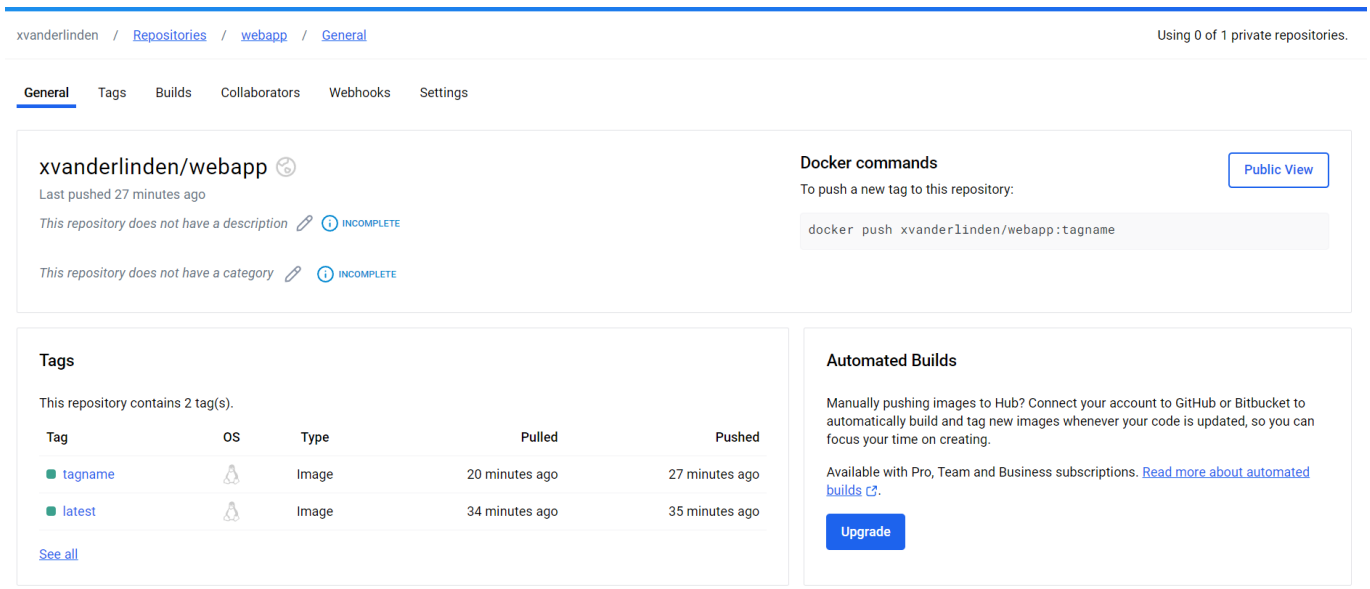
Proof of work done

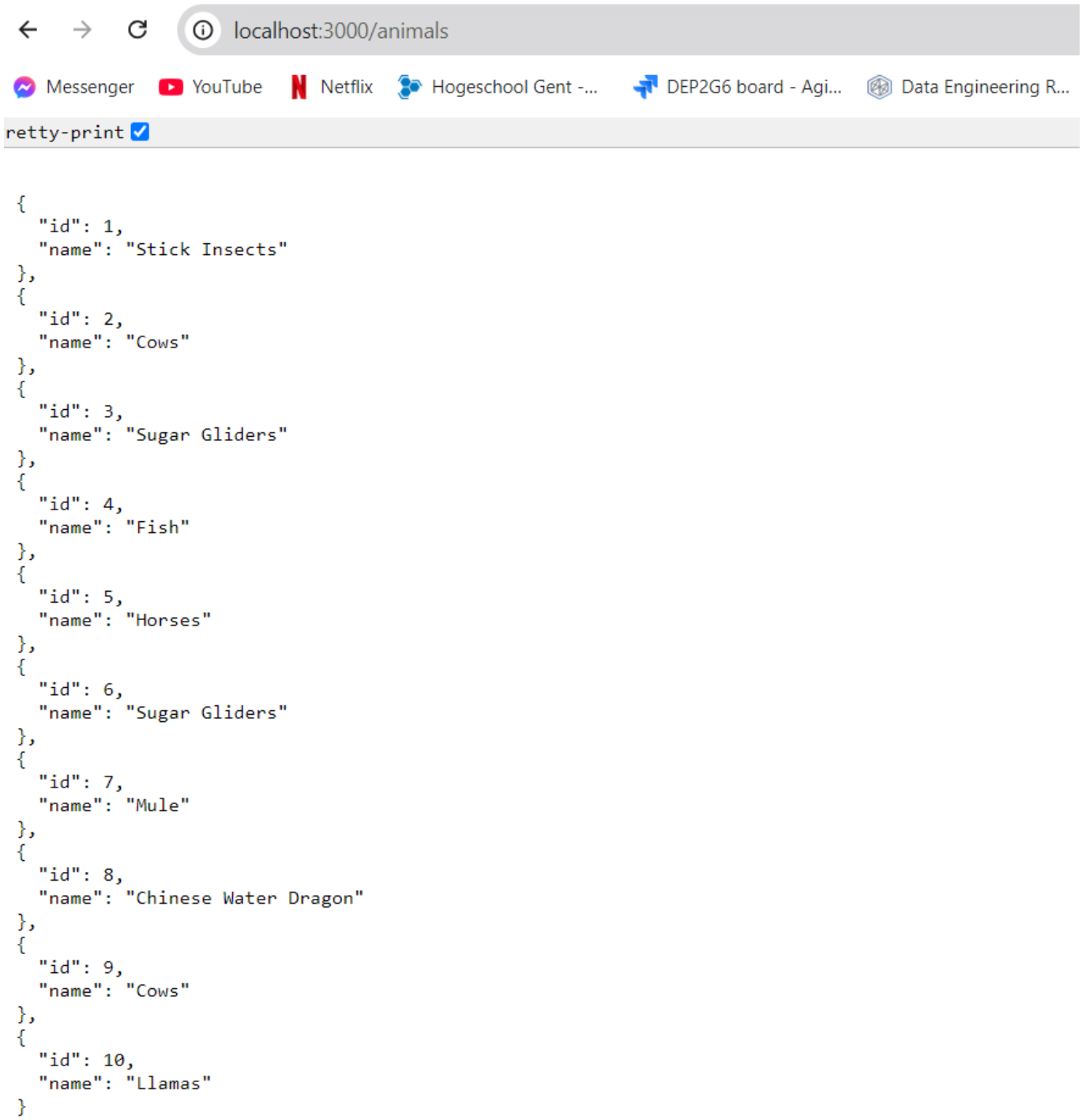
```
PS C:\Users\Xander\mlops-2425-xvanderlinden\resources\02-dockerlab\webapp> docker compose run test
>>
time="2024-10-12T21:53:10+02:00" level=warning msg="C:\\Users\\Xander\\mlops-2425-xvanderlinden\\resources\\02-dockerlab\\webapp\\docker-compose.
yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Creating 2/0
 ✓ Container webapp-database-1  Running                                                                    0.0s
 ✓ Container webapp-webapp-1    Created                                                                    0.0s
[+] Running 1/1
 ✓ Container webapp-webapp-1  Started                                                                      0.4s
yarn run v1.22.22
$ mocha tests/animals.spec.js


  Animals
    ✓ should 200 and return all animals (38ms)
    ✓ should 200 and return a single animal
    ✓ should 404 and return an error when the animal is not found


  3 passing (55ms)

Done in 0.32s.
```





Evaluation criteria

Evaluation criteria

- ☑ **Show that you created a Docker image for the API**
- ☑ **Show that you can start the API using the SQLite database**
- ☑ **Show that you can start the API using the MySQL database**
- ☑ **Show that you can access the API on port 3000**
- ☑ **Show that you optimized the Docker image size**
- ☑ **You've used an Alpine version of Node.js**
- ☑ **You've copied and installed the dependencies in a separate layer**
- ☑ **Show all running containers in the Portainer dashboard**

- ☑ **Show that all tests are passing**
- ☑ **Show that you pushed the Docker image to Docker Hub and that you can pull it from Docker Hub**
- ☑ **Show that you wrote an elaborate lab report in Markdown and pushed it to the repository**
- ☑ **Show that you updated the cheat sheet with the commands you need to remember**

Issues

"none."

Reflection

What was difficult?

Het moeilijkste was de juiste Docker compose file attributen vinden voor te verbinden met de 2 verschillende soorten databases.

What was easy?

Een image maken voor de node.js API ging zeer vlot.

What did you learn?

Ik weet nu hoe ik op een effictieve manier docker images maak en dees zo optimaal mogelijk deploy.

What would you do differently?

Ik zou eerst de volledige opdracht lezen voor ik docker files begin te maken aangezien er heel wat aanpassingen moesten gebeuren bij later delen van de opdracht.

Resources

- Docker documentation
- Docker Compose documentation
- Node.js Docker best practices