|  |
| --- |
|  |
| MLOPS assignment |
| Diabetes application |

|  |
| --- |
| Dominic Ho 3MCT  13-12-2022 |

Inhoud

[Chosen dataset 2](#_Toc121774347)

[AI Model 2](#_Toc121774348)

[About 2](#_Toc121774349)

[FastAPI 3](#_Toc121774350)

[Environment 3](#_Toc121774351)

[Schema 4](#_Toc121774352)

[Routes 5](#_Toc121774353)

[Database with Docker 6](#_Toc121774354)

[Connect database 7](#_Toc121774355)

[Database tables 8](#_Toc121774356)

[Repository and queries 9](#_Toc121774357)

[Docker deployment 10](#_Toc121774358)

[Github Container Registry 11](#_Toc121774359)

[Push image to Github 11](#_Toc121774360)

[Kubernetes 12](#_Toc121774361)

[Create deployment, add service, port-forward and scale it 12](#_Toc121774362)

[Secret 14](#_Toc121774363)

# Chosen dataset

<https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset>

## AI Model

I will keep it simple and use a Logistic regression classifier, with hyperparameter tuning I will have around 77 – 82 accuracy.

# About

My fictional company is making an application that speeds up diabetes diagnoses.  
A normal blood sugar/glucose test will take a few hours (taking the blood sample and comparing changes after different tests)

After every test, a blood sample must be sent to the lab which then can take a few days to get the test results back. We want to speed up diagnosis in the lab with our AI model.

(if possible we could also skip the lab part with a mobile application that nurses or doctors can use directly, but I don’t know what happens with the samples in the lab. I assume the blood gets analyzed with (expensive) machines )

# FastAPI

## Environment

I made a new Fastapi **environment** with Poetry

Afbeelding met tekst

Automatisch gegenereerde beschrijving

## Schema

Added a schema **Patient**

Afbeelding met tekst

Automatisch gegenereerde beschrijving

## Routes

Added **routes** with **API** **rerouting**

Afbeelding met tekst

Automatisch gegenereerde beschrijving

## Database with Docker

Setting up **database** with a **docker-compose.yaml** file

Login to database with **.env** file

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Afbeelding met tekst

Automatisch gegenereerde beschrijving

## Connect database

**Connect** to database with **database.py**

Afbeelding met tekst

Automatisch gegenereerde beschrijving

## Database tables

Create database tables with **patient\_model**

Afbeelding met tekst

Automatisch gegenereerde beschrijving

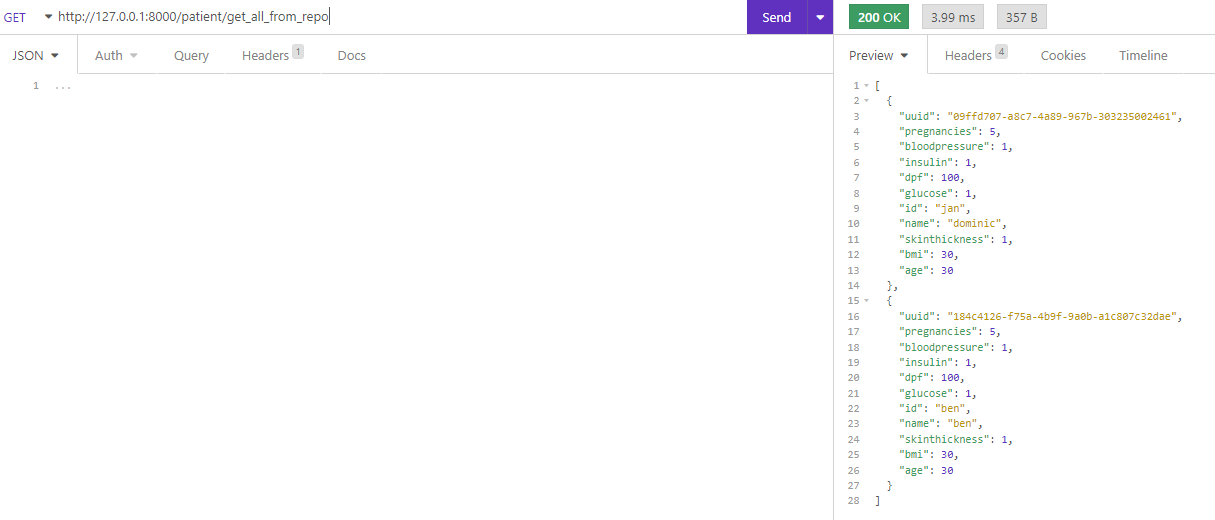
Register Patient Schema as an ORM-ready object

Afbeelding met tekst

Automatisch gegenereerde beschrijving

## Repository and queries

Make a repository and add queries from repository



Afbeelding met tekst

Automatisch gegenereerde beschrijving

## Docker deployment

Docker deployment with **Dockerfile**, we also add a **service** to our docker-compose.yaml

Afbeelding met tekst

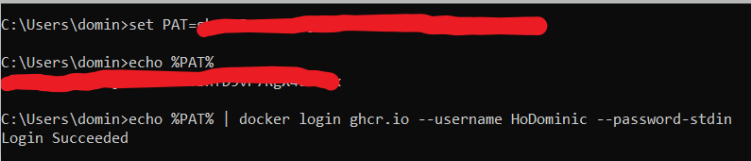
Automatisch gegenereerde beschrijving

Afbeelding met tekst

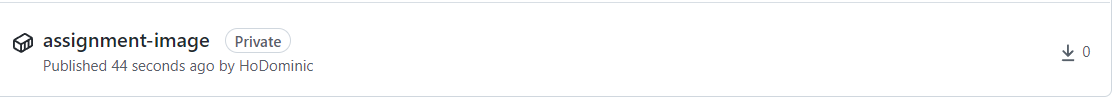
Automatisch gegenereerde beschrijving

## Github Container Registry

Generate a token from Github and login with Github account



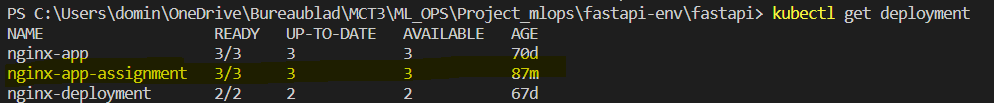
## Push image to Github

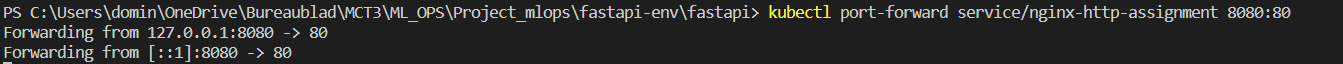


# Kubernetes

To scale out our API accross multiple machines, and get them updated in a perfect manner, we use **Kubernetes**

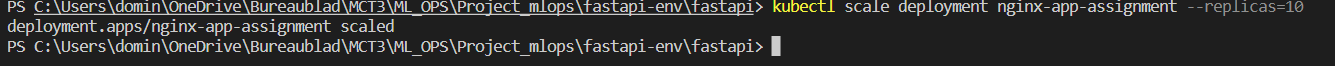
### **Create deployment, add service, port-forward and scale it**





Afbeelding met tekst

Automatisch gegenereerde beschrijving



We need a **cluster-role-binding.yaml** and **service-account.yaml** and a **token** to access the **Kubernetes** **dashboard**

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Afbeelding met tekst

Automatisch gegenereerde beschrijving

## Secret

Create secret from Docker Registry and deploy API with Kubernetes

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Afbeelding met tekst

Automatisch gegenereerde beschrijving