

Casus

Eén van onze Data Scientists heeft een nieuw ML model opgeleverd, in een Jupyter Notebook. Als Data Engineering-team nemen we deze in beheer. Eén van onze senior Data Engineers heeft alvast een raamwerk neergezet voor een Python-script waarin we het model kunnen gaan hosten. We zitten nu in de testfase: het gaat dus nog niet over schaalbaarheid, maar over het controleren of het model *containerized* goed gaat werken.

Het model voorspelt de kosten die we als verzekeraar gaan maken bij een patiënt die in het ziekenhuis opgenomen wordt.

Globaal stappenplan

Om het Python-script in een pakketje op te nemen waarin alle afhankelijkheden aanwezig zijn, gaan we een container aanmaken op basis van de benodigde Python-versie. In die container installeren we vervolgens de benodigde modules, en kopiëren de benodigde (Python)-scripts. Vervolgens testen we of de container werkt.

Afhankelijk van hoe makkelijk je dit afgaat, kun je vervolgens kiezen of je het script van de Data Scientist zelf wilt gaan vertalen. Dit maakt voor de “flow” en het grote plaatje van MLOps niet uit: wanneer je zelf de vertaling niet maakt, kun je het meegeleverde voorbeeldscript gebruiken.

Uitwerking

Inloggen op VM

Log in op de VM op het meegeleverde RDP-bestand. Het is een Windows 11-VM met daarop Docker reeds geïnstalleerd.

Loginnaam: student

Wachtwoord: SigmaDataLearning.nl

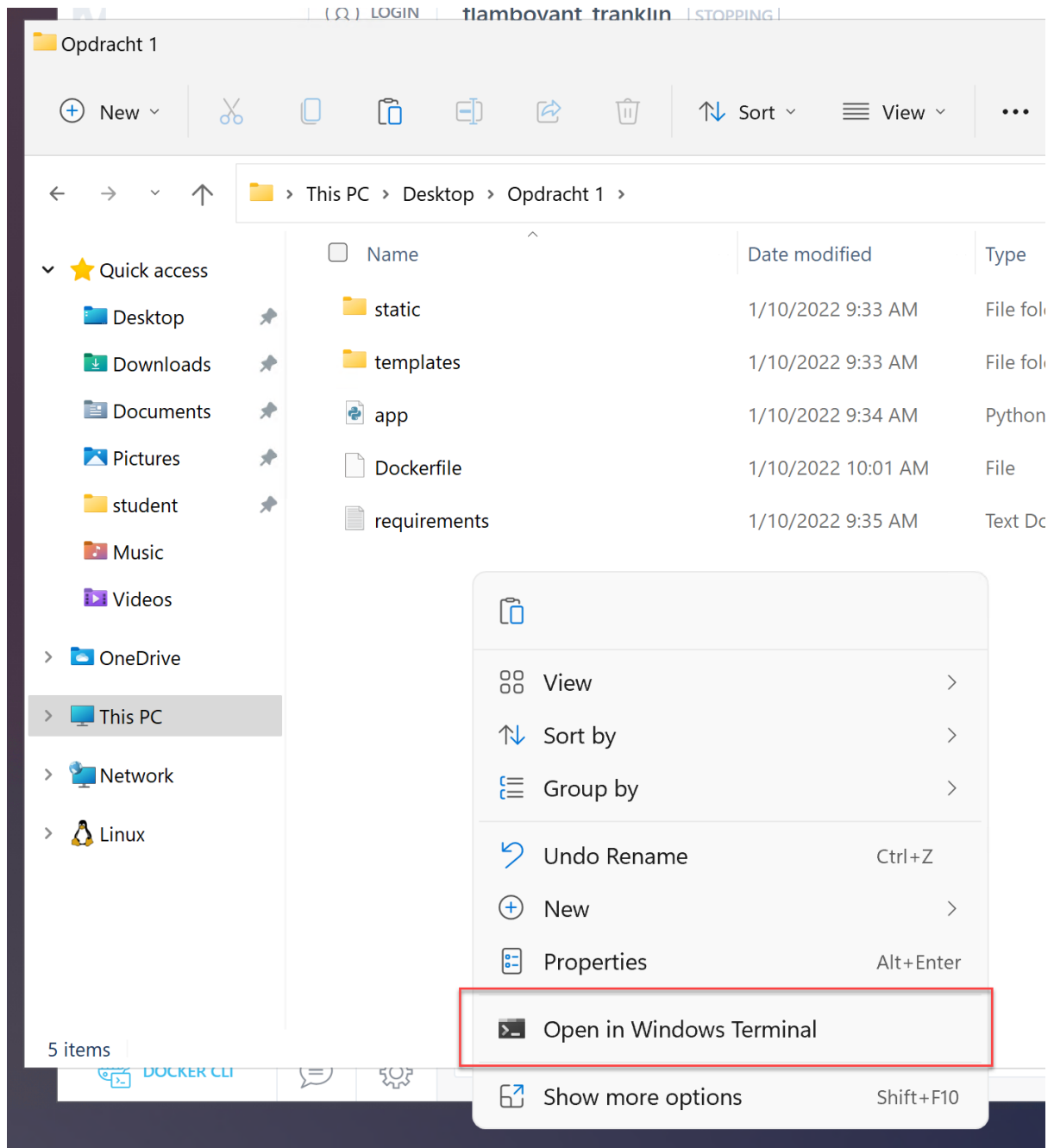
Maken van een container

Een container wordt gemaakt op basis van een *Dockerfile*: een recept waarin staat welke basis een container gebruikt en wat dit image moet gaan doen. Iemand uit het IT-team heeft al een Dockerfile voor je opgezet die simpelweg een website aan je kan tonen.

1. Open de map “Opdracht 1”, die je op het bureaublad van de VM vindt
Je ziet hier de volgende bestanden en mappen:

Bestand	Betekenis
Dockerfile	Hierin staat het “recept” wat er nodig is om een container te maken
requirements.txt	Hierin staat welke versies van welke modules het Python-script nodig heeft. Dit is niet iets dat specifiek voor containers is, maar wel voor Python
App.py	Dit is het Python-script dat we overal willen kunnen laten uitvoeren
static\Style.css templates\home.html	Om een website netjes te laten zien, wordt wat styling aan toegevoegd. Deze bestanden worden gebruikt door de website, maar kun je verder negeren

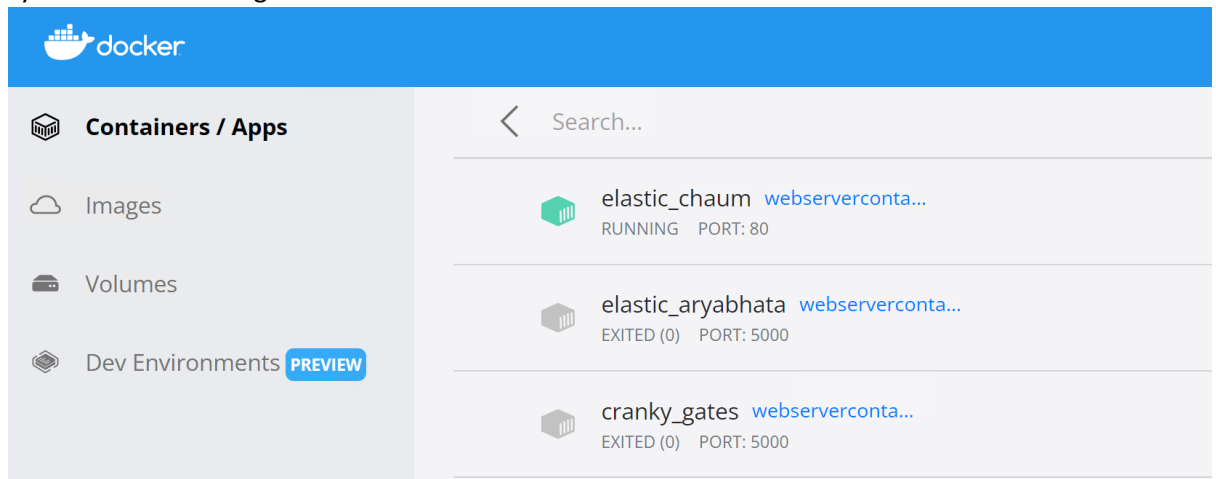
2. Bekijk de eerste drie bestanden in Notepad++ of Visual Studio Code (beide staan geïnstalleerd op de VM), en kijk of je herkent wat hierboven beschreven wordt
3. Open de Dockerfile, en bekijk de beschrijving goed. Snap je wat hier gebeurt?
4. In Windows Verkenner, binnen de map Opdracht 1, rechtsklik en kies voor “Windows Terminal”.



Er opent zich nu een Windows Terminal-venster (een opdrachtprompt) binnen de huidige map.

5. Laat Docker nu een container "bouwen". Om deze container te kunnen identificeren, geven we deze het naamkaartje (de "tag") *webservercontainer* mee:
`docker build . -t webservercontainer`
6. De container die nu gemaakt is, bevat alle informatie die nodig is om op elke mogelijke container-host deze Python-applicatie te hosten. Voer de container nu op de VM uit met het volgende commando:
`docker run -d -p 5000:5000 webservercontainer`
 Binnen enkele seconden ben je terug op de commandline, en draait je applicatie.
7. Open nu de webbrowser, en ga naar het webadres <http://localhost:5000/>. Je ziet hier een tekst met meer uitleg over het *docker run* commando wat je zojuist hebt uitgevoerd.

8. Open nu de applicatie *Docker Desktop*. Hier zie je welke containers er momenteel op je systeem staan en uitgevoerd worden.



De namen die je in het zwart ziet zijn overigens willekeurig gekozen – in het blauw erachter zie je welke blauwdruk er gebruikt wordt.

9. Stop en verwijder de containers uit deze lijst. Bekijk ook de lijst met *images*, maar verwijder deze niet!