

2023

Docker

DESK AGENT AI
TADRAŁA, PIOTR P.P.

O-PP-CMK

CONTENTS

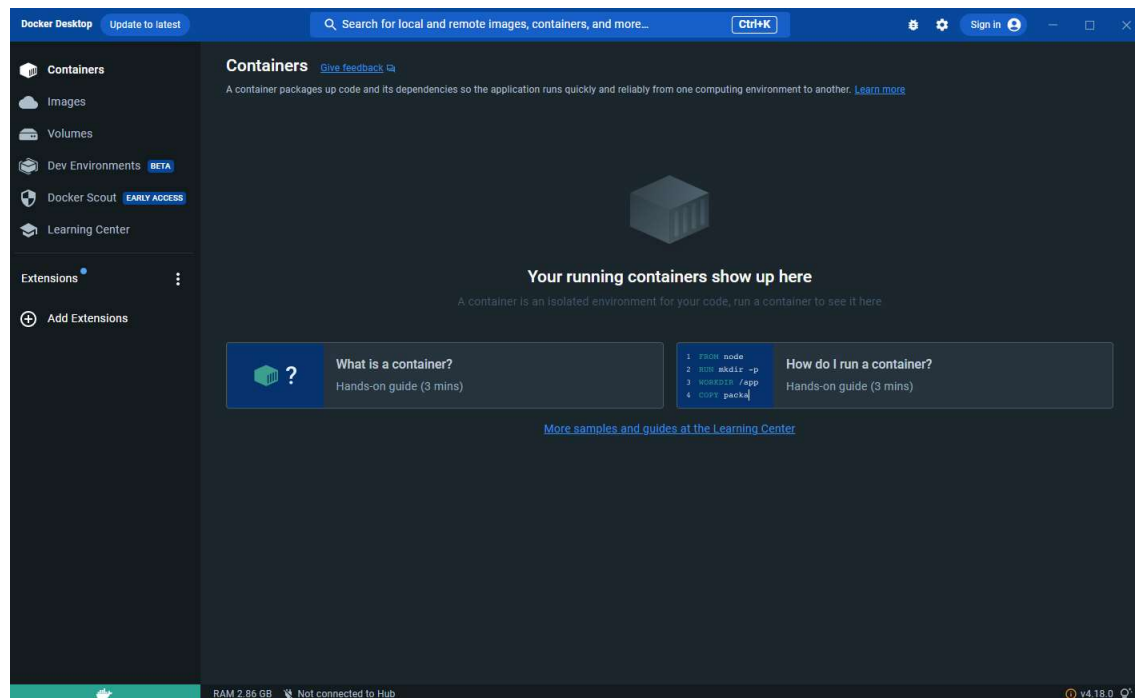
Inleiding	2
Vorbereiding	2
Dockerfile	3
Image Build	4
Container	5
Eindresultaat.....	6

INLEIDING

Onze Desk Agent-applicatie is ontworpen om te worden uitgevoerd in Docker-images. Dit is volledig nieuw voor mij, en ik wil daarom mijn beperkte kennis naar het juiste niveau tillen voor dit project. In dit document wordt mijn leerproces beschreven.

VOORBEREIDING

Docker kan volledig worden aangestuurd vanuit de command line interface (CLI), en uiteindelijk wil ik dat ook gaan doen. Voor nu heb ik echter ook de desktopapplicatie gedownload om extra ondersteuning te hebben, totdat ik alles soepel vanuit de CL kan beheren.

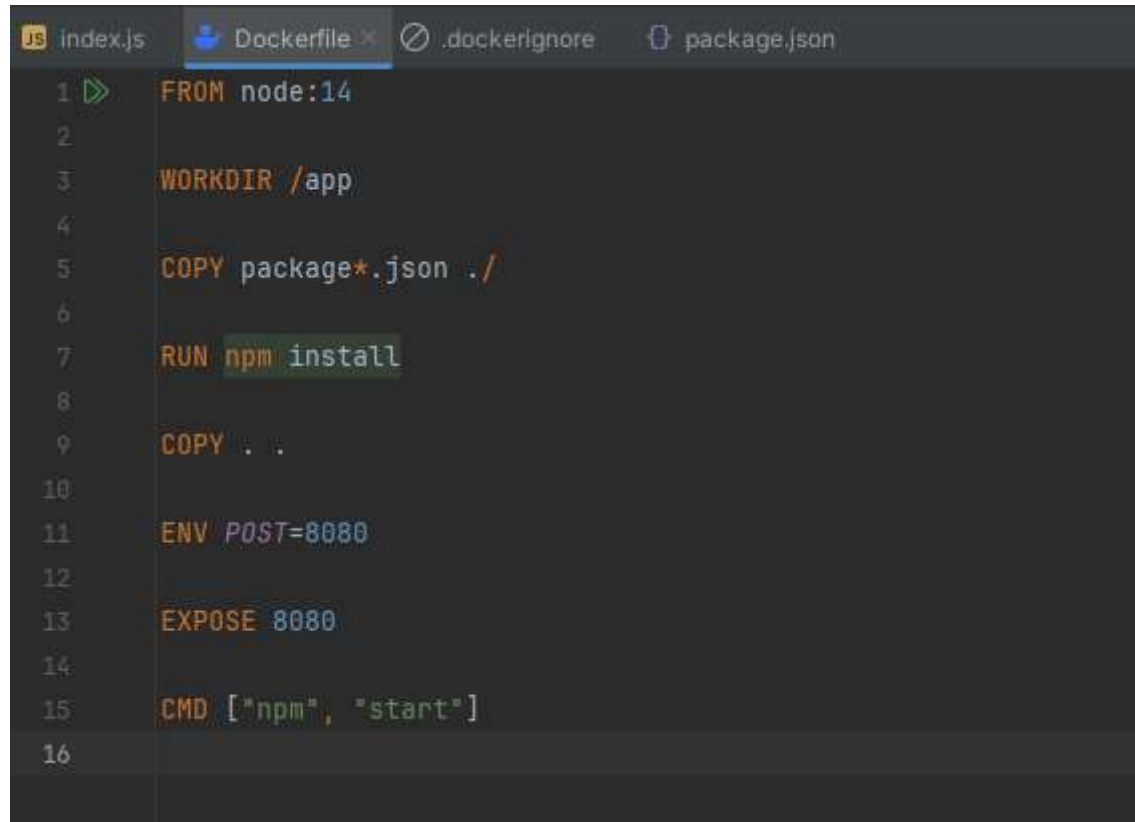


Vervolgens heb ik een app nodig die ik kan containerizen. Hiervoor heb ik een eenvoudige Node.js-applicatie gemaakt die luistert naar GET-requests op poort 8080.

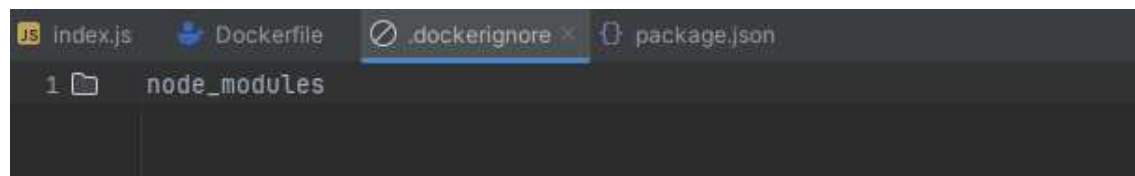
```
1  const express : e | => core.Express = require('express');
2
3  const app : any | Express = express();
4  const port : string | number = process.env.PORT || 8080;
5
6  app.get('/', (request : Request<P, ResBody, ReqQuery, LocalsObj> , response : Response<ResBody, LocalsObj> ) : void => {
7    response.json( body: {message: 'Docker response'});
8  })
9  app.listen(port, callback: () => console.log('Listening on ${port}'));
10
```

DOCKERFILE

Om een applicatie te dockerizen, heb je in eerste instantie een Dockerfile nodig. In dit bestand geef je alle parameters en commando's aan die nodig zijn voor het bouwen van de image. Hiermee heb ik de meeste moeite gehad, maar dankzij de hulp van ChatGPT kon ik alle foutmeldingen oplossen. Daarnaast heb ik een .dockerignore file aangemaakt om onnodige bestanden niet te kopiëren. Het concept van .dockerignore is hetzelfde als dat van .gitignore en was dus niet nieuw voor mij.



```
JS index.js Dockerfile x .dockerignore package.json
1 FROM node:14
2
3 WORKDIR /app
4
5 COPY package*.json ./
6
7 RUN npm install
8
9 COPY . .
10
11 ENV POST=8080
12
13 EXPOSE 8080
14
15 CMD ["npm", "start"]
16
```



```
JS index.js Dockerfile .dockerignore x package.json
1 node_modules
```

IMAGE BUILD

Zodra mijn Dockerfile klaar was, kon ik het image builden. Na enkele pogingen en wijzigingen in de Dockerfile is het me gelukt om een werkend image te builden

```
λ docker build -t test-image:latest .
[+] Building 102.2s (10/10) FINISHED
=> [internal] load build definition from Dockerfile                                0.1s
=> => transferring dockerfile: 504B                                              0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 34B                                                  0.0s
=> [internal] load metadata for docker.io/library/node:14                      1.9s
=> [1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd846 85.2s
=> => resolve docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461 0.0s
=> => sha256:b253aeafeaa7e0671bb60008df01de101a38a045ff7bc656e3b0fbfc7c05cca5 7.86MB / 7.86MB          5.1s
=> => sha256:3d2201bd995cccf12851a50820de03d34a17011dcbb9ac9fdf3a50c952cbb131 10.00MB / 10.00MB        5.5s
=> => sha256:1d12470fa662a2a5cb50378dc8c8ea228c1735747db410bbbf8e2d9144b5452 7.51kB / 7.51kB         0.0s
=> => sha256:2ff1d7c41c74a25258bfa6f0b8adb0a727f84518f55f65ca845ebc747976c408 50.45MB / 50.45MB       19.0s
=> => sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa 776B / 776B           0.0s
=> => sha256:2cfa3fbb0b6529ee4726b4f599ec27ee557ea3dea7019182323b3779959927f 2.21kB / 2.21kB         0.0s
=> => sha256:1de76e268b103d05fa8960e0f77951ff54b912b63429c34f5d6adfd09f5f9ee2 51.88MB / 51.88MB      24.3s
=> => sha256:d9a8df5894511ce28a05e2925a75e8a4acb0634c39ad734fd8ba8e23d1b1569 191.85MB / 191.85MB    56.8s
=> => sha256:6f51ee005deac0d99898e41b8ce60ebf250ebe1a31a0b03f613aec6bbc9b83d8 4.19kB / 4.19kB        20.3s
=> => extracting sha256:2ff1d7c41c74a25258bfa6f0b8adb0a727f84518f55f65ca845ebc747976c408 2.0s
=> => sha256:5f32ed3c3f278edda4fc571c880b5277355a29ae8f52b52cdf865f058378a590 35.24MB / 35.24MB     39.4s
=> => extracting sha256:b253aeafeaa7e0671bb60008df01de101a38a045ff7bc656e3b0fbfc7c05cca5 0.4s
=> => extracting sha256:3d2201bd995cccf12851a50820de03d34a17011dcbb9ac9fdf3a50c952cbb131 0.3s
=> => sha256:0c8cc2f24a4dc6b64e602e086fc9446b0a541e8acd9ad72d2e90df3ba22f158b3 2.29MB / 2.29MB       25.6s
=> => extracting sha256:1de76e268b103d05fa8960e0f77951ff54b912b63429c34f5d6adfd09f5f9ee2 4.7s
=> => sha256:0d27a8e861329007574c6766fba946d48e20d2c8e964e873de352603f22c4ceb 450B / 450B           25.8s
=> => extracting sha256:d9a8df5894511ce28a05e2925a75e8a4acb0634c39ad734fd8ba8e23d1b1569 22.1s
=> => extracting sha256:6f51ee005deac0d99898e41b8ce60ebf250ebe1a31a0b03f613aec6bbc9b83d8 0.0s
=> => extracting sha256:5f32ed3c3f278edda4fc571c880b5277355a29ae8f52b52cdf865f058378a590 3.1s
=> => extracting sha256:0c8cc2f24a4dc6b64e602e086fc9446b0a541e8acd9ad72d2e90df3ba22f158b3 0.2s
=> => extracting sha256:0d27a8e861329007574c6766fba946d48e20d2c8e964e873de352603f22c4ceb 0.0s
=> [internal] load build context                                                0.0s
=> => transferring context: 3.54kB                                              0.0s
=> [2/5] WORKDIR /usr/src/app                                                  8.0s
=> [3/5] COPY package*.json ./                                                0.2s
=> [4/5] RUN npm install                                                       5.1s
=> [5/5] COPY . .                                                             0.2s
=> exporting to image                                                         1.1s
=> => exporting layers                                                         0.9s
=> => writing image sha256:4f7c3f8eff40c57fd3df599cb841acbdd536a0735a62d066b00930d07f76479 0.0s
=> => naming to docker.io/test-image:latest                                   0.0s
```

Deze was ook zichtbaar in de desktopapp.

Refresh to see disk usage 1 images

Last refresh: 37 minutes ago

<input type="checkbox"/>	Name	Tag	Status	Created	Size	Actions
<input type="checkbox"/>	test-image 4f7c3f8eff4	latest	Unused	11 minutes ago	916.28 MB	<div></div>

Enkele mislukte pogingen:

```
λ docker build -t test-image .
[+] Building 139.7s (9/11)
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 345B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 34B 0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest 0.7s
=> [internal] load build context 0.0s
=> => transferring context: 3.38kB 0.0s
=> [1/7] FROM docker.io/library/ubuntu:latest@sha256:8eab65df33a6de2844c9aefd19efe8ddb87b7df5e9185a4ab73af936 0.0s
=> CACHED [2/7] WORKDIR /app 0.0s
=> CACHED [3/7] COPY package*.json ./ 0.0s
=> [4/7] RUN apt-get update && apt-get install -y curl 26.4s
=> ERROR [5/7] RUN curl -sL https://deb.nodesource.com/setup_8.x | bash - 112.5s
```

```
λ docker build -t test-image .
[+] Building 22.4s (10/11)
=> [internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 236B 0.0s
=> [internal] load .dockerignore 0.1s
=> => transferring context: 34B 0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest 0.6s
=> [1/7] FROM docker.io/library/ubuntu:latest@sha256:8eab65df33a6de2844c9aefd19efe8ddb87b7df5e9185a4ab73af936 0.0s
=> [internal] load build context 0.1s
=> => transferring context: 689B 0.0s
=> CACHED [2/7] WORKDIR /app 0.0s
=> CACHED [3/7] COPY package*.json ./ 0.0s
=> CACHED [4/7] RUN apt-get update && apt-get install -y curl 0.0s
=> [5/7] RUN apt-get install nodejs -y 20.9s
=> ERROR [6/7] RUN npm install 0.6s
```

```
λ docker build -t test-image .
[+] Building 2.8s (9/11)
=> [internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 233B 0.0s
=> [internal] load .dockerignore 0.1s
=> => transferring context: 34B 0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest 0.6s
=> [1/7] FROM docker.io/library/ubuntu:latest@sha256:8eab65df33a6de2844c9aefd19efe8ddb87b7df5e9185a4ab73af936 0.0s
=> [internal] load build context 0.1s
=> => transferring context: 3.27kB 0.0s
=> CACHED [2/7] WORKDIR /app 0.0s
=> CACHED [3/7] COPY package*.json ./ 0.0s
=> CACHED [4/7] RUN apt-get update 0.0s
=> ERROR [5/7] RUN apt-get install nodejs 1.9s
```

CONTAINER

Het laatste wat nog overgebleven was, was het aanmaken van een container en het runnen van mijn image. Met ``$docker images`` kon ik het juiste ID ophalen en vervolgens kon ik met een extra port forwarding parameter de image runnen:

```
λ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
test-image latest 4f7c3f8efff4 46 seconds ago 916MB

C:\repos\DockerCursus (dockercursus@1.0.0)
λ docker run -p 5000:8080 4f7c3f8efff4

> dockercursus@1.0.0 start /usr/src/app
> node src/index.js

Listening on 8080
```

<input type="checkbox"/>	Name	Image	Status	Port(s)	Last started	Actions
<input type="checkbox"/>	 elated_engelbart f09a00329739	test-image:latest	Running	5000:8080	7 seconds ago	  

EINDRESULTAAT

Het resultaat hiervan is een NODE.js-applicatie die in een Docker-container draait.

