**Docker Eindopdracht**

**Aude Sustronck**

studiegebied IWT

bachelor in de toegepaste informatica

Docker & Cloud Computing

academiejaar 2024-2025

Inhoud

[Applicatie 3](#_Toc186655816)

[Frontend 3](#_Toc186655817)

[Backend 3](#_Toc186655818)

[Docker 3](#_Toc186655819)

[Frontend Dockerfile 3](#_Toc186655820)

[Backend Dockerfile 4](#_Toc186655821)

[Docker-compose 4](#_Toc186655822)

[Google cloud 6](#_Toc186655823)

[Dockerhub 8](#_Toc186655824)

[Minikube 9](#_Toc186655825)

[Deployment naar google cloud 11](#_Toc186655826)

[Helm – environments 12](#_Toc186655827)

[CI/CD 15](#_Toc186655828)

[Secrets 15](#_Toc186655829)

[Dockerhub 15](#_Toc186655830)

[Google cloud 16](#_Toc186655831)

[Github 17](#_Toc186655832)

[Monitoring grafana-Prometheus 19](#_Toc186655833)

[Cloudflare 22](#_Toc186655834)

[Linken 26](#_Toc186655835)

# Applicatie

## Frontend

React applicatie van een wishlist, waar je een lijst krijgt van items, een item kan toevoegen en dit item kan verwijderen.

Afbeelding met tekst, schermopname, Lettertype, ontwerp

Automatisch gegenereerde beschrijving

## Backend

Een MongoDB en node API met GET, POST en DELETE.

# Docker

## Frontend Dockerfile

FROM node:18 AS build

WORKDIR /app

COPY package.json package-lock.json ./

RUN npm install

COPY . .

RUN npm run build

FROM nginx:alpine

COPY --from=build /app/build /usr/share/nginx/html

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]

In de terminal:

PS C:\Users\audes\OneDrive - Hogeschool VIVES\2024-2025\Docker\assignmentprivate-AudeSu\frontend> docker build -t frontend .

PS C:\Users\audes\OneDrive - Hogeschool VIVES\2024-2025\Docker\assignmentprivate-AudeSu\frontend> docker run -p 80:80 frontend

## Backend Dockerfile

FROM node:18

WORKDIR /app

COPY package.json package-lock.json ./

RUN npm install

COPY . .

EXPOSE 5000

CMD ["npm", "start"]

In de terminal:

PS C:\Users\audes\OneDrive - Hogeschool VIVES\2024-2025\Docker\assignmentprivate-AudeSu\backend> docker build -t backend .

PS C:\Users\audes\OneDrive - Hogeschool VIVES\2024-2025\Docker\assignmentprivate-AudeSu\backend> docker run -p 5000:5000 backend

## Docker-compose

services:

    frontend:

        build:

            context: ../frontend

        ports:

            - '80:80'

        depends\_on:

            - backend

    backend:

        build:

            context: ../backend

        ports:

            - '5000:5000'

        environment:

            - MONGO\_URI=mongodb://db:27017/wishlist

        depends\_on:

            - db

    db:

        image: mongo

        ports:

            - '27017:27017'

        volumes:

            - mongo-data:/data/db

volumes:

    mongo-data:

In de terminal:

PS C:\Users\audes\OneDrive - Hogeschool VIVES\2024-2025\Docker\assignmentprivate-AudeSu\docker> docker-compose -p wishlist up -d

Nu zou je dit moeten bekomen:

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Testen:

Frontend-url: <http://localhost:80>  
Backend-url: <http://localhost:5000/api/wishlist>  
Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

Afbeelding met tekst, schermopname, software

Automatisch gegenereerde beschrijving

## Google cloud

Maak een account aan op google cloud (je moet hier wel je rekening voor linken)  
klik linksboven op my first project en maak een nieuw project.  
Afbeelding met tekst, schermopname, lijn, Lettertype

Automatisch gegenereerde beschrijving

Afbeelding met tekst, schermopname, Lettertype, software

Automatisch gegenereerde beschrijving

Zorg dat je nu in het project wishlist zit.  
zoek in de zijbalk naar het onderdeel kubernetes engine.

Afbeelding met Lettertype, tekst, logo, symbool

Automatisch gegenereerde beschrijving

Klik vervolgens op enable  
Afbeelding met tekst, Lettertype, schermopname

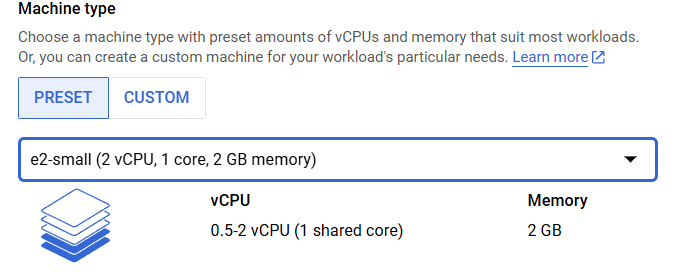
Automatisch gegenereerde beschrijving

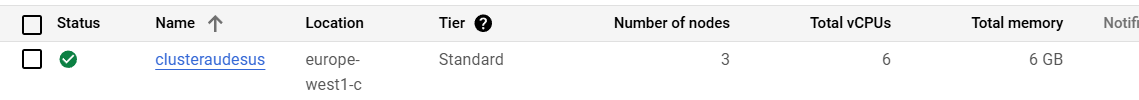
Na even wachten kan je nu bij cluster op create klikken om een kubernetes cluster aan te maken.  
Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

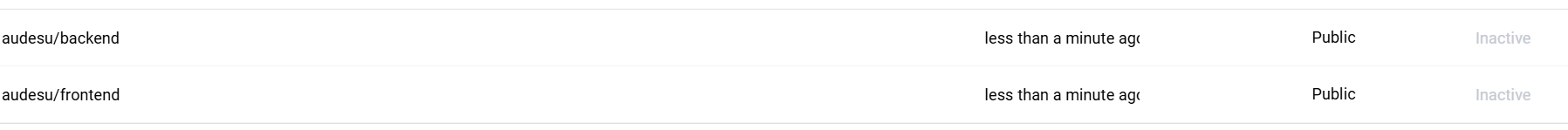
Als je rechtsboven de optie krijgt om te switchen naar een standaard cluster is het aangeraden om dit te doen.  
Geef vervolgens een naam in voor je cluster bv: (clusteraudesus).  
en zet de zone op zonal en kies voor een europe-west-…  
Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

Klik vervolgens in de zijbalk op default-pool je kan hier de naam veranderen als je dit wenst.  
Klik vervolgens links op nodes.  
Laat alle instellingen staan en verander machine type naar e2-small  


Klik vervolgens op create, dit kan even duren.  


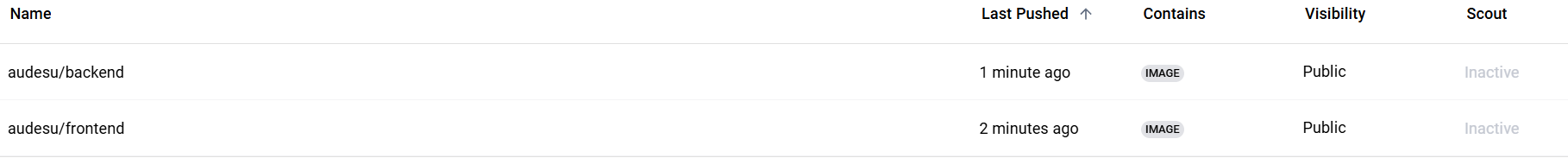
## Dockerhub

Log in op dockerhub  
en ga vervolgens via my profile naar repositories.  
maak hier 2 nieuwe repositories aan genaamd frontend en backend.  


Ga nu terug naar je frontend en backend map op je project en voor volgende compass uit.

docker build --build-arg REACT\_APP\_BACKEND\_URL=http://127.0.0.1:5000 -t audesu/frontend .

docker push audesu/frontend

docker build -t audesu/backend .   
docker push audesu/backend  


## Minikube

Vul de .yaml files verder aan of maak nieuwe als dit nodig is in je k8s directory.  
run vervolgens minikube start.

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

Run nu volgende commando’s:  
kubectl apply -f services.yaml  
kubectl apply -f deployment.yaml  
Afbeelding met tekst, software, Lettertype, schermopname

Automatisch gegenereerde beschrijving

Nu kan je kubectl get svc uitvoeren en zal je dit zien.

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

De external ip staat nog op pending, dit is omdat je met minikube werkt.  
los dit op door minikube tunnel uit te voeren.  
Alles zal nu nog lokaal runnen op 127.0.0.1 natuurlijk  
Afbeelding met tekst, schermopname, Lettertype, software

Automatisch gegenereerde beschrijving  
nu kan je op een andere terminal opnieuw kubectl get svc uitvoeren.  
Afbeelding met tekst, schermopname, Lettertype, software

Automatisch gegenereerde beschrijving

Als je nu naar de extern IP’s gaat zal je je website bereiken.

Frontend: <http://127.0.0.1/>

Backend: <http://127.0.0.1:5000/api/wishlist>

Afbeelding met tekst, schermopname, Lettertype, ontwerp

Automatisch gegenereerde beschrijving  
Afbeelding met tekst, schermopname, software, Lettertype

Automatisch gegenereerde beschrijving

## Deployment naar google cloud

Eerst moet je gcloud installeren  
vervolgens moet je hier op inloggen.  
vervolgens zal gcloud vragen welk project je wil gebruiken, kies hier voor jou gemaakt project.

Voer vervolgens dit commando uit:  
gcloud container clusters get-credentials clusteraudesus --zone europe-west1-c --project wishlist-446412  
clusteraudesus = cluster name  
europe-west1-c = zone  
wishlist-446412 = project-id

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Voer volgende commando’s uit om alles opnieuw te builden.

Afbeelding met tekst, Lettertype, software, schermopname

Automatisch gegenereerde beschrijving

Via kubectl get svc kan je nu je poorten vinden voor je backend en frontend.  
  
Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Nu dat je de external ip hebt van je backend kan je een nieuwe image bouwen voor je frontend met deze backend.  
docker build --build-arg REACT\_APP\_BACKEND\_URL=http://34.140.230.157:5000 -t audesu/frontend:online .

Testen

Frontend: <http://34.140.50.48/>  
backend: <http://34.140.230.157:5000/api/wishlist>

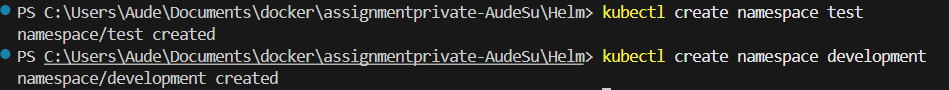
Afbeelding met tekst, software, schermopname, Multimediasoftware

Automatisch gegenereerde beschrijving  
Afbeelding met Multimediasoftware, software, Grafische software, schermopname

Automatisch gegenereerde beschrijving

## Helm – environments

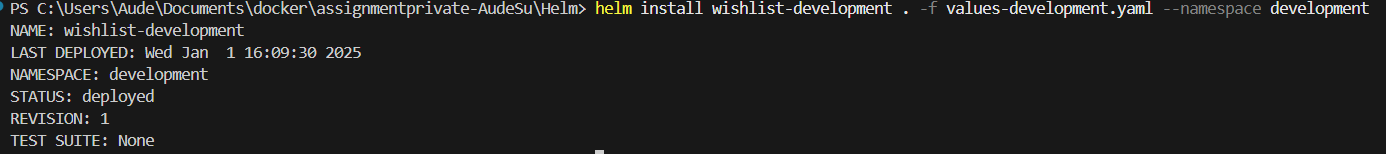
Eerst maken we 2 namespaces aan (groepen)



Na het invullen van alle files voor helm (deployment, service, values, vlues-test, values-…)  
helm install wishlist-test . -f values-test.yaml --namespace test  
Afbeelding met tekst, software, Lettertype, Multimediasoftware

Automatisch gegenereerde beschrijving

helm install wishlist-development . -f values-development.yaml --namespace development

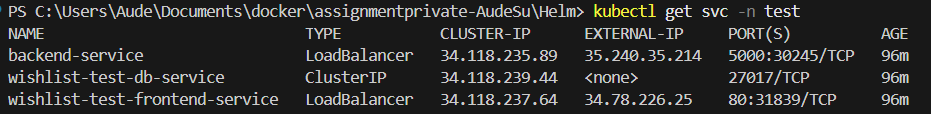


Stel dat je nog iets veranderd hebt en de environment wil updaten kan dit zo:  
helm upgrade wishlist-development . -f values-development.yaml -n development

testen – test environment

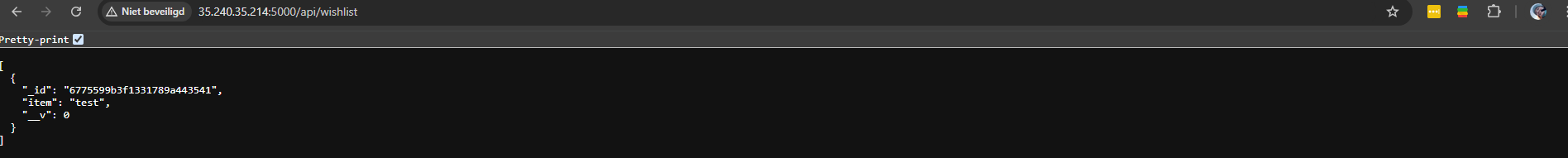
frontend test: <http://34.78.226.25/>  
backend test: <http://35.240.35.214:5000/api/wishlist>

Afbeelding met tekst, schermopname, Lettertype

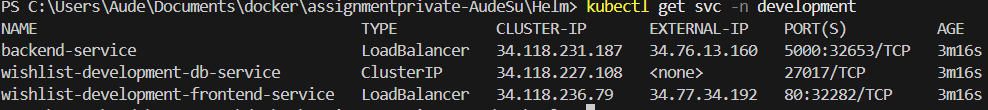
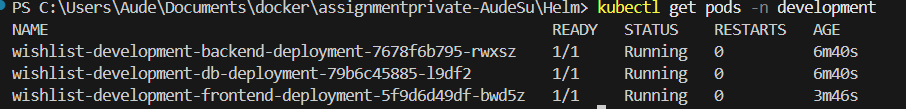
Automatisch gegenereerde beschrijving  
  
Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

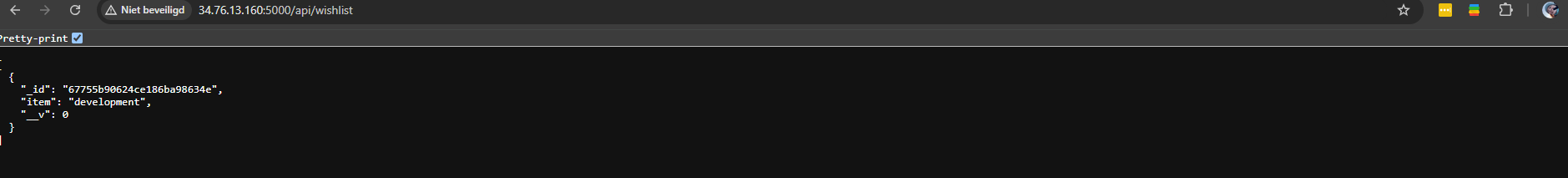
Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving  


Testen – test development

Frontend development: <http://34.77.34.192/>  
backend development: <http://34.76.13.160:5000/api/wishlist>  
  
  


Afbeelding met schermopname, tekst, software, Multimediasoftware

Automatisch gegenereerde beschrijving  


## CI/CD

Maak op je github een tweede repository aan (dockerwishlist)

En voer volgend commando uit:  
git remote add wishlist-remote <https://github.com/AudeSu/dockerwishlist.git>

## Secrets

Nu gaan we verschillende variabelen aanmaken die door github gebruikt zullen worden.

### Dockerhub

Ga naar dockerhub en ga naar je account settings.  
Klik vervolgens op personal access tokens  


Klik op generate new token  
vul een naam in en zet de permissies op read, write, delete  
Afbeelding met tekst, schermopname, Lettertype, software

Automatisch gegenereerde beschrijving

Klik nu op generate, Nu zie je jou token:  
dckr\_pat\_f6V1WJofyimtI1lFEVt\_fr8Ewgs

### Google cloud

Ga naar je google cloud pagina, en zoek naar IAM & admin en selecteer service accounts  
Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

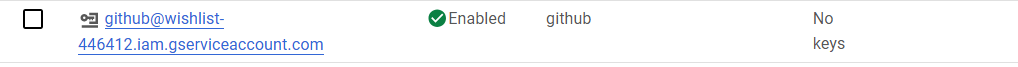
Klik vervolgens op create new service account en geef een naam  
Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

Kies vervolgens voor de rol kubernetes engine admin, zodat je full access hebt, en klik vervolgens op done.Afbeelding met tekst, schermopname, Lettertype, software

Automatisch gegenereerde beschrijving

Klik nu op je nieuwe user.



Klik vervolgens op het tablad keys en selecteer add key en create new key.  
Afbeelding met tekst, Lettertype, software, schermopname

Automatisch gegenereerde beschrijving

Hou de selectie op json  
Afbeelding met tekst, Lettertype, schermopname, wit

Automatisch gegenereerde beschrijving

Nu is de key gedownload op je computer.

### Github

Ga nu naar je github repository.  
ga naar je settings en onder secrets and variables klik je op actions  
Afbeelding met tekst, schermopname, Lettertype, ontwerp

Automatisch gegenereerde beschrijving

Klik vervolgens op new repository secret en geef hier een naam en je token, doe dit voor alle keys die je zelf nodig hebt.

Ga nu terug naar je project en in de root maak .github/workflows aan met daarin een yaml file: deployment.yaml, vul deze file aan zodat hij eerst op google cloud inlogged en het juiste project aanspreekt, vervolgens build je een nieuwe image van je frontend backend en push je deze naar je dockerhub.  
installeer vervolgens helm en upgrade de environment die je wilt updaten.

Testen

Verander nu iets aan je html pagina.  


Nu ga je git add . uitvoeren  
git commit -m “nieuwe update”  
git push wishlist-remote main  
Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Je kan op github zien als je workflow correct uitgevoerd is.  
binnen je repository ga je naar actions en daar kan je dit opvolgen.  
Afbeelding met tekst, Lettertype, schermopname

Automatisch gegenereerde beschrijving

Als je nu naar   
<https://34.77.34.192/> gaat  
zou je de verandering moeten zien  
Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

## Monitoring grafana-Prometheus

Maak een nieuwe namespace aan dit is best practice.

Kubectl create namespace monitor  


Voeg vervolgens prometheus toe aan je project.  
helm repo add prometheus-community <https://prometheus-community.github.io/helm-charts>



Doe nu het zelfde voor grafana:  
helm repo add grafana https://grafana.github.io/helm-charts  


Voer vervolgens helm repo update uit:  
Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Nu moeten we grafana en prometheus installeren  
helm install prometheus prometheus-community/prometheus -n monitor  
helm install grafana grafana/grafana -n monitor

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Voer vervolgens kubectl -n monitor edit svc grafana uit, en zet type op LoadBalancer zodat er een external IP is.

Vervolgens voer je dit uit, kubectl edit secret grafana -n monitor -o yaml  
nu kan je de username en password aanpassen  
de user staat automatisch op admin (het is encrypted) dus kopieer gwn de username voor het password dit maakt het makkelijker.  
Afbeelding met tekst, Lettertype, schermopname

Automatisch gegenereerde beschrijving  
voer vervolgens dit commando uit om de aanpassingen door te voeren:  
kubectl rollout restart deployment grafana -n monitor

Grafana-url: <http://34.38.12.206/login>

Nu kan je inloggen met   
username: admin   
password: admin  
Afbeelding met software, Multimediasoftware, tekst, Grafische software

Automatisch gegenereerde beschrijving

Ga vervolgens naar connections > add new connection en zoek prometheus en selecteer prometheus.  
Afbeelding met tekst, Multimediasoftware, schermopname

Automatisch gegenereerde beschrijving  
klik vervolgens op add data source.

Vul deze link in bij de connection  
<http://prometheus-server.monitor.svc.cluster.local:80>  
  
prometheus-server : kan je vinden door kubectl get svc -n monitor  
Klik nu op save & test

Nu kan je rechtsboven op het plusje klikken en import dashboard doen  
Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

je kan hier nu een ID geven dit is je eigen keuze, veel gebruikte ID’s zijn:  
1860 Prometheus Monitoring Dashboard  
315 Kubernetes Cluster Monitoring  
10856 Kubernetes Node Exporter Full  
Afbeelding met tekst, schermopname, klok

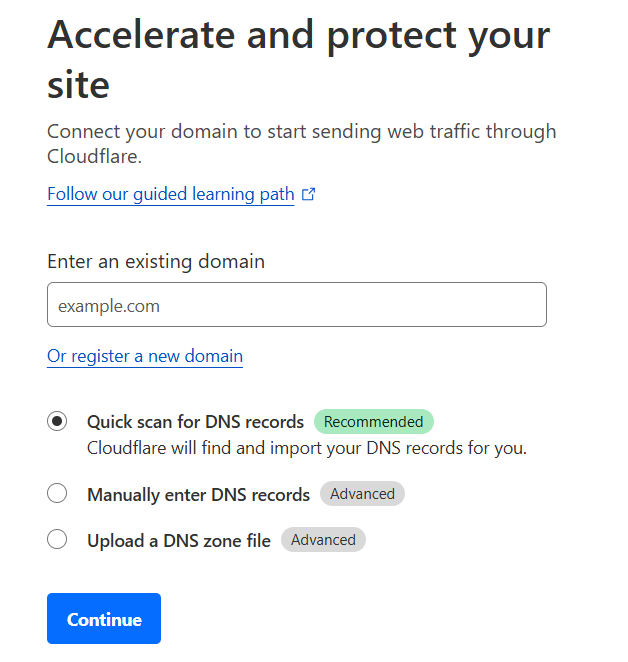
Automatisch gegenereerde beschrijving

## Cloudflare

Op aanraden van medestudenten heb ik een goedkoop domain gekocht op One.com

Audes.online

Ga nu naar cloudflare en log hier op in met google.

Vervolgens geef je je domain naam in en klik je op continue.  


Kies vervolgens voor de free versie en klik continue  
Afbeelding met tekst, Lettertype, nummer, lijn

Automatisch gegenereerde beschrijving

Geef vervolgens alle records in die je vind op one.com en voor de A-record geef je het juist ip adres mee, klik vervolgens op continue to activation  
Afbeelding met tekst, schermopname, software, Lettertype

Automatisch gegenereerde beschrijving

Vervolgens krijg je 2 namespaces die je moet toevoegen aan je Domain.  
je doet dit bij DNS-instellingen  
Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Na even wachten zou je pagina nu moeten verschijnen.  
<http://audes.online/>

Ga nu naar zero trust > networks > tunnels  
Afbeelding met tekst, software, Webpagina, nummer

Automatisch gegenereerde beschrijving

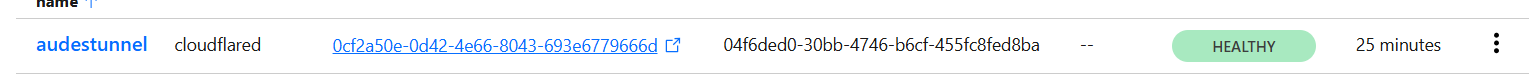
Klik vervolgens op add tunnel en selecteer cloudflared  
Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

Geef een naam en save tunnel.

Nu kom je een pagina met een token die je nodig hebt in je code  
eyJhIjoiOGQxNjA1OTJmOTFhMTlhZjgxZmUzMGUwZTViNDRlMzkiLCJ0IjoiMDRmNmRlZDAtMzBiYi00NzQ2LWI2Y2YtNDU1ZmM4ZmVkOGJhIiwicyI6Ik1qWTNOV1F5TkRBdE5HTTJZeTAwWmpOakxXSmtaVE10TUdJMk4yWTNPVEkwTmpFNCJ9

Nu maak je in de root van je project een map aan cloudflare  
daarin maak je opnieuw zoals bij helm de files aan, na je dit gedaan hebt run je volgend commando:  
helm install cloudflare . -f values.yaml

Nu zou je dit moeten zien bij je tunnel  


Als je nu op de 3 bolletjes klikt en op configre kan je bij public hostname alle subdomains toevoegen.  
Afbeelding met tekst, schermopname, lijn, Lettertype

Automatisch gegenereerde beschrijving

Ik heb de echte productie website gedeployed, de grafana gedeployed en de test website gedeployed om fout te tonen.

Extra:

Omdat ik deze build structuur gebruik  
docker build --build-arg REACT\_APP\_BACKEND\_URL=http://127.0.0.1:5000 -t audesu/frontend .

Is het nodig om de runnende backend nog eens te linken aan mijn runnende frontend.  
ik doe dit door een nieuwe image te maken op mijn dockerhub:  
  
docker build --build-arg REACT\_APP\_BACKEND\_URL=https://backend.audes.online -t audesu/frontend:cloudprod .

docker push audesu/frontend:cloudprod

en in mijn k8s/deployment file verander ik de gebruikte image naar audesu/frontend:cloudprod  
als ik dit terug verander naar online zal dit terug zonder cloudflare runnen.  
Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

nu run ik dit commando: kubectl apply -f deployment.yaml

nu zal je zien dat de echte website nu zal werken:  
<https://audes.online/>  
Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

<https://backend.audes.online/api/wishlist>

Afbeelding met Multimediasoftware, software, tekst, schermopname

Automatisch gegenereerde beschrijving

Als ik dit niet zou doen krijg ik errors zoals degene die je ziet bij mijn testsite  
<https://frontend-test.audes.online/>

Afbeelding met tekst, software, Multimediasoftware, Computerpictogram

Automatisch gegenereerde beschrijving

Dit kan ik opnieuw oplossen via volgende methode:  
docker build --build-arg REACT\_APP\_BACKEND\_URL= <https://backend-test.audes.online> -t audesu/frontend:cloud .

docker push audesu/frontend:cloud

nu in mijn helm/values-test.yaml de tag veranderen naar cloud  
Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

vervolgens dit commando runnen:  
helm upgrade wishlist-test . -f values-test.yaml -n test

en het zal werken maar ik laat dit nu staan zodat ik deze fout kan aantonen.

## Linken

Productie website frontend: <http://34.140.50.48/>

Productie website backend: <http://34.140.230.157:5000/api/wishlist>

Development website frontend (CI/CD): <http://34.77.34.192/>

Development website backend: <http://34.76.13.160:5000/api/wishlist>

test website frontend: <http://34.78.226.25/>

test website backend: <http://35.240.35.214:5000/api/wishlist>

website grafana overzicht: <http://34.38.12.206/login>

Cloudflare productie website frontend: <https://audes.online/>

Cloudflare productie website backend: <https://backend.audes.online/api/wishlist>

Cloudflare test website frontend (met fout): <https://frontend-test.audes.online/>

Cloudflare test website backend: <https://backend-test.audes.online/api/wishlist>

Cloudflare Grafana overzich: <https://grafana.audes.online/>