



Verdieping Support

Adam support

SEM 3

Inleiding

Om te laten zien hoe ik mij heb verdiept in Support zal ik refereren naar onderdelen in mijn capita selecta maar ook in mijn proftaak verslag die betrekking hebben tot support. Ik raad ook aan om het verslag met de capita selecta in te zien zodat u weet waar ik nog meer aan gewerkt heb.

Onderwerp	Bladzijden
De opdracht	3
Kubernetes	4 t/m 5
Docker	6 t/m7
WebsiteLink	8
Loadbalancer	9

De opdracht

Tijdens de proftaak kreeg ik de opdracht om ervoor te zorgen dat de lokalen website zich emigreert naar vsphere en dat deze schaalbaar gedeployed word.

Hiervoor wil ik gebruik maken van Docker en Kubernetes.

Docker en Kubernetes zijn populaire tools die gebruikt kunnen worden om een applicatie schaalbaar te maken.

Een van de belangrijkste voordelen van het gebruik van Docker is dat u hiermee een toepassing en de bijbehorende afhankelijkheden in één enkele container kunt verpakken. Dit maakt het eenvoudig om de applicatie in elke omgeving te implementeren en uit te voeren, omdat alle benodigde componenten zich in de container bevinden.

Kubernetes is een krachtige tool voor het op grote schaal beheren van gecontaineriseerde applicaties. Hiermee kunt u meerdere containers op een server cluster implementeren en beheren, met functies zoals taakverdeling, automatische schaling en zelfherstel.

Door Docker te gebruiken om een applicatie in een container te verpakken en Kubernetes om de implementatie en schaling van de container applicatie te beheren, kunt u een applicatie eenvoudig schaalbaar maken. Als u bijvoorbeeld een applicatie heeft die een plotselinge toename van het verkeer ondervindt, kan Kubernetes automatisch extra containers laten draaien om de belasting aan te kunnen, zodat de applicatie responsief en beschikbaar blijft voor gebruikers.

Bovendien kan het gebruik van Docker en Kubernetes het gemakkelijker maken om updates en nieuwe functies uit te rollen naar een applicatie, omdat je de container eenvoudig kunt bijwerken en opnieuw kunt implementeren met behulp van Kubernetes. Hierdoor kunt u snel en efficiënt wijzigingen aanbrengen in uw applicatie zonder dat u zich zorgen hoeft te maken over de onderliggende infrastructuur.

Over het algemeen zijn Docker en Kubernetes krachtige tools die u kunnen helpen een applicatie schaalbaar te maken, ervoor te zorgen dat deze steeds meer verkeer aankan en beschikbaar blijft voor gebruikers.

Kubernetes

Om de services te installeren moeten twee dingen gebeuren.

1. Kubernetes installatie
2. Yaml files

Kubernetes installatie

In deze installatie zal er 1 master zijn met twee worker nodes.
3 machine waar Ubuntu op draait vanaf versie 20.04

Installatie master node

Stap 1 Download k3s

```
curl -sfL https://get.k3s.io | sh -
```

Stap 2 check of de installatie is gewerkt

```
sudo kubectl get node
```

installatie voor worker node

Stap 1 Genereer een token vanuit de master node

```
sudo cat /var/lib/rancher/k3s/server/node-token
```

Stap 2 verbindt de worker node met de master node

```
curl -sfL https://get.k3s.io | K3S_TOKEN="JOUWTOKEN" K3S_URL="https://[Your IP-adres]:6443" K3S_NODE_NAME="Naammagjezelfkiezen" sh -
```

Stap 3 Check de resultaten ga terug naar de master node en type
kubectl get node

Tot zover de Kubernetes installatie.

Yaml files

Er zijn twee yaml files die gedeployed dienen te worden.

1. Webserver
2. Nodeport

Webserver yaml file :

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx

namespace: default

spec:

replicas: 6

selector:

matchLabels:

app: nginx De web service draaid op nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: hoiikbenadam/basyx-website8 (Dit is de gebruikte image)

imagePullPolicy: Always

ports:

- containerPort: 80

name: web

protocol: TCP

readinessProbe:

httpGet:

port: web

path: /

Om ervoor te zorgen dat de website van buitenaf bereikbaar is moet de interne kubernetes IP adres geforward worden na een ip adres die op het web gevonden kan worden.

Nodepoort yaml file:

apiVersion: v1

kind: Service

metadata:

name: nginx-nodeport

namespace: default

spec:

type: NodePort

selector:

app: nginx

ports:

- name: web

port: 80

targetPort: web

nodePort: 31111

Docker

De opdrachtgever wilt graag dat de infrastructuur automatisch uitrolbaar is en ook schaalbaar. Dit betekent dat we kubernetes nodig hebben echter vereist kubernetes image's om vervolgens daarvan containers te deployen en te reguleren. Het is dus belangrijk om van de webserver een image te maken

1. Maak een ubuntu-desktop omgeving aan
2. Instaleer docker
<https://pimylifeup.com/raspberry-pi-docker/>
3. Type `sudo nano index.html`
(Hierin zal de html van de website worden ingevoerd)
4. Type `sudo nano touch dockerfile`

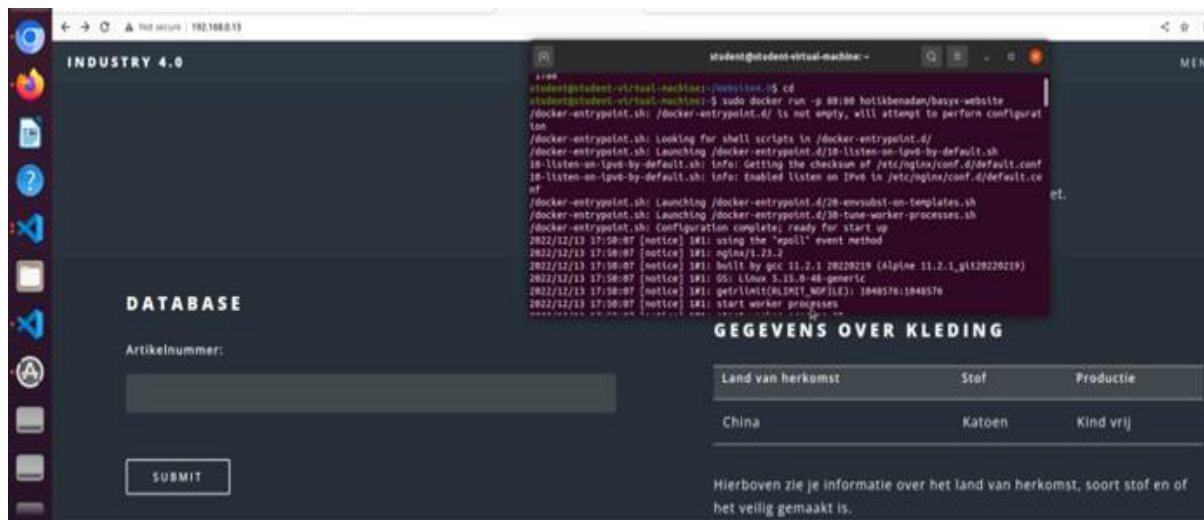
```
Type FROM nginx:alpine
COPY ./index.html /usr/share/nginx/html/index.html
COPY ./elements.html /usr/share/nginx/html/elements.html
COPY ./generic.html /usr/share/nginx/html/generic.html
COPY assets /usr/share/nginx/html/assets
COPY images /usr/share/nginx/html/images
```

De bovenstaande regels zijn op te sommen in drie delen. Het eerste deel is het **COPY** commando dit kopieert de bestanden van de host naar de docker-image. Het tweede gedeelte is het **./ 'Bestand'** segment dit gedeelte duidt aan op het bestand dat zich op de host bevindt. Het derde gedeelte is de **path** die het bestand in de Docker container zal hebben. Het laatste gedeelte is een referentie naar de gebruikte bestand.

5. Type `sudo docker build -t basyxwebsite8`
6. Maak een account aan op de docker hub
Dit is het account waar de image naar toe wordt geupload kubernetes zal het dan via dockerhub pullen.
Ga naar terminal en type `docker login` en vul de gegevens in.
7. Type `docker tag html-image hoiikneadam/basyxwebsite8`
Dus de naam van de image met de gebruiksnaam die je op dockerhub gebruikt.
8. `docker push hoiikbenadam/basyxwebsite8`
9. Nu om te checken of de image correct werkt
1. Maak een nieuwe Ubuntu desktop machine
2. Download docker

3. Type `sudo docker run -p 80:80 hoiikbenadam/basyx-website8`

Resultaat



Docker Hub

Omdat we de image in docker hub hebben geupload diend dit metween twee doelen.

1. Uiteraard is de website nu makkelijk te deployen en schaalbaar te maken.
2. Maar het is ook een hele goede back want mocht er iets gebeuren dan kan er altijd een nieuwe container gedeployed worden.

Door simple **sudo docker run -p 80:80 hoiikbenadam/basyx-website8** uit te voeren is de website al local op uw machine bereikbaar.

Website link

Om de website overal bereikbaar te maken zijn twee dingen nodig.

1. Domain name

2. Poortforwarding

1. De domeinnaam is nodig zodat de gebruiker gemakkelijk de website kan vinden
gelukkig heeft een klasgenoot Ihsan al een domeinnaam gekocht.
2. Open firewall → NAT → Interface kies WAN → Address Family: IPv4 → Protocol: TCP →
Destination port range: HTTP → Single host 192.168.0.10 (Router ip address) →
Redirect target port: Other 31111 (De kubernetes ip poort) → Description : http → NAT
reflection : User system default → Rule NAT http → save

Link website : Basyx.ihsansoft.nl

Loadbalancer

In mijn capita selecta heb ik een website gemaakt waar ik hacks op kan uitvoeren. Om mogelijke downtime op de website te vermijden heb ik een load balancer gemaakt. Dit zorgt ervoor dat er meerdere instanties van dezelfde website worden gemaakt met een andere inwendige ip adres.

Wanneer de websites overladen wordt door bijvoorbeeld hacks dan wordt een loadbalancer toegepast. Deze loadbalancer maakt een nieuwe website aan en geeft het een aparte interne ip kubernetes ip adres. Echter ziet de gebruiker hier niets van terug.

```
kubectl expose deployment dvwa --port=80 --target-port=80 \
--name=dvwa-service -- type=LoadBalancer
```

```
dvwa-deployment-5b08806978-mr1jv 1/1 Running 0 52s
root@master-virtual-machine:~# kubectl describe services dvwa-deployment
Name:          dvwa-deployment
Namespace:     default
Labels:        <none>
Annotations:   <none>
Selector:      app=dvwa
Type:          ClusterIP
IP Family Policy: SingleStack
IP Families:   IPv4
IP:            10.43.11.139
IPs:           10.43.11.139
Port:          <unset> 80/TCP
TargetPort:    80/TCP
Endpoints:     10.42.0.19:80,10.42.0.20:80,10.42.0.21:80 + 3 more...
Session Affinity: None
Events:        <none>
```

Wanneer gekeken wordt naar de regel **Endpoints** dan zie je dat er 6 interne ip adressen zijn aangemaakt voor dezelfde website.