

Docker

Slides voor Docker workshop van het
IT-lab



Meevolgen op:

<https://hogent-it-lab.github.io/docker-workshop/slides>

Wat is Docker?

- Virtualisatie op een ander niveau: containerisatie
- Gebruikt onderliggend Linux kernel
- Opzetten van applicaties
- Worden geïsoleerd van het hostsysteem (sandbox)

Waarom containerisatie en Docker?

- Bundelen van software en requirements/dependencies!
- Isoleren van draaiende applicaties
- Makkelijk verschillende containers op een systeem naast elkaar
- Verdelen van resources hostsysteem -> minder verspilling!



Docker versus virtuele machine

- Wat is nu eigenlijk het verschil met een virtuele machine (VM)?
- Virtuele machine: emuleert volledige computer (virtuele hardware!)
- (Docker) container: emuleert op applicatie-niveau



Bouwstenen van Docker

- Docker Image
- Docker Container

Docker Image

- Beschrijving van alles dat nodig is voor een applicatie
- Het recept of de blauwdruk
- Gebaseerd op een Dockerfile (~ ingrediënten)
- Kan je zelf heel custom maken!

Dockerfile

```
FROM node:20-alpine    # Kies jouw image
WORKDIR /app
COPY . .                # Kopieer de broncode

# Installeer de dependencies en run de software

RUN yarn install --production
CMD ["node", "./src/index.js"]
```

Docker Container

- Een draaiende instantie van een image
- Geïsoleerde sandbox (los van hostsysteem)
- Kan je customizen met variabelen
- Vaak ga je een container opspinnen van een bestaande image!

Docker - structuur

Port bindings

- Elke container heeft een IP-adres (zelf ingesteld of automatisch toegewezen)
- Elke container zit in een (of meerdere) Docker netwerk (zelf ingesteld of automatisch toegewezen)
- Concept van port binding: een poort van een container koppelen aan een poort van het hostsysteem

Port bindings - visueel

 bg:100%

Volume binding

- Mappen/bestanden van jouw Docker container binden aan het hostsysteem -> **bind mount**
- Nuttig voor data-persistentie
- Biedt mogelijkheden voor het maken van backups
- Live aanpassingen maken (development en testen!)

Docker installeren

- Algemene stappen (voor Linux)
- Voor deze demo: werken in een virtuele machine (Ubuntu desktop)
- Ook heel eenvoudig mogelijk op MacOS
- Windows mogelijkheden: WSL en Docker desktop (Opgelet!!)
- Voorkeur werken in Linux omgeving!

Docker gebruiken - commando's

- Beheren van containers!
- `docker ps`
- `docker run`
- `docker stop`
- `docker rm`
- `docker rmi`

Docker gebruiken - commando's

- Extra's en handige zaken:
- `docker exec -ti`
- `docker system prune -a`

Docker gebruiken - Docker Compose

- Docker run commando's: handig maar omslachtig...
- Wat met meerdere containers tegelijk opstarten?
- Oplossing: Docker Compose!

Docker demo - Minecraft server

- Eigen Minecraft server opstarten in een container
- Volume binding voor de data van de server
- Speel Minecraft op een server die je zelf in beheer hebt!

Docker networking

- Standaard: Docker regelt vanalles zelf under the hood
- Je kan zelf (interne) Docker netwerken declareren
- Nut? Isolatie van netwerken, overzicht, veiligheid,...

Nuttige links - Docker

- [Docker documentatie](#)
- [Docker installatie](#)
- [Docker Compose documentatie](#)
- [Docker Networking documentatie](#)

Nuttige links - Docker

- [Composerize](#) - `docker run` commando naar docker-compose
- [DockerHub](#)
- [Docker cheat sheet](#)