

# Beginnen met Docker



**Develop faster.  
Run anywhere.**

# Programma



- Biografie
- Scope sessie
- Soorten gastsystemen
- Installatie Docker
- Images en containers
- Dockerhub
- Soorten images
- Locatie Docker projecten
- Stappen maken Docker container
- Docker compose
- Vragen
- Extra: handige docker commando's



# Biografie



- Ongeveer 20 jaar open source gebruiker: begonnen met PHP en MySQL
- Ongeveer 15 jaar geleden Linux gaan gebruiken door op Pogoplug Arch Linux te zetten
- Door goede WIKI Arch op thuisserver gaan gebruiken en kennis Arch/Linux verder uitgebreid.
- Ongeveer 5 jaar geleden voor thuisserver overgestapt naar Debian
- Ongeveer 10 jaar Linux Mint op desktop als hoofdbesturingssysteem
- Auteur van artikelen in Linux Magazine
- Actief lid binnen NLLGG sinds september 2018
- In 2021 gestart met plaatsen Nederlandstalige informatie over Linux en Open Source op <https://www.hetnetwerk.org>

# Scope sessie



1. Installeren Docker onder Linux
2. Kennis maken met Docker via de opdrachtregel
3. Uitleg van de belangrijkste begrippen
4. Er leiden veel wegen naar Rome. Dat geldt ook voor Docker. In deze sessie volgen we 1 route.
5. Maken van een Docker container met de Apache webserver.



# Soorten gastsystemen



1. Virtualisatie (b.v. VirtualBox)
2. Linux Containers (LXC en LXD)
3. Container platforms (Docker, Kubernetes, Podman)



# Installatiemogelijkheden Docker



- Docker Desktop for Linux
- Via de Docker APT repositories
- Handmatige installatie en handmatige updates
- Via een script



# Installatie Docker via Script

- Werkt eigenlijk op alle Linux varianten
- Root of sudo rechten nodig
- Formeel alleen voor testdoeleinden

```
curl -fsSL https://get.docker.com -o get-docker.sh  
sudo sh get-docker.sh
```

- Daarna huidige gebruiker toevoegen aan Docker groep:

```
sudo usermod -aG docker $USER ==> uit en inloggen
```



# Images en containers



In eigen woorden:

Een **image** is te vergelijken met een programma dat je downloadt. Je kunt in een image geen wijzigingen aanbrengen.



Een **container** is een image dat je start. De instellingen van de container zorgen ervoor dat je gegevens kunt opslaan en raadplegen.



# Dockerhub



**Dockerhub** (<https://hub.docker.com>) is een website waarop veel Docker *images staan.*



The screenshot shows the Docker Hub homepage with a blue header. The main heading is "Develop faster. Run anywhere." Below it, a search bar contains "Search Docker Hub" and a "ctrl+K" button. The page is divided into several sections:

- Trusted content:** Docker Official Image, Verified Publisher, Sponsored OSS.
- Categories:** API Management, Content Management System, Data Science, Databases & Storage, Languages & Frameworks, Integration & Delivery, Internet of Things, Machine Learning & AI, Message Queues, Monitoring & Observability, Networking, Operating Systems, Security.
- Spotlight:** Three featured articles: "Build up to 39x faster with Docker Build Cloud", "LLM Everywhere: Docker and Hugging Face", and "Take action on prioritized insights".
- Machine Learning & AI:** Four featured Docker images: tensorflow/tensorflow, pytorch/pytorch, langchain/langchain, and ollama/ollama.



# Soorten images – 1 -



Search bar containing 'wordpress|' with icons for search, close, help, refresh, and menu. Buttons for 'Sign In' and 'Sign up' are on the right.

1 - 25 of 10.000 results for wordpress.

Best Match ▾



 **wordpress**  ==> **Docker Official Image** ↓1B+ · ☆5.6K Pulls: 615,613 Last week

Updated 5 hours ago

The WordPress rich content management system can utilize plugins, widgets, and themes.

**CONTENT MANAGEMENT SYSTEM**

[Learn more](#)



 **bitnamicharts/wordpress**  ==> **Verified publisher** ↓1M+ · ☆1 Pulls: 14,901 Last week

By [VMware](#) · Updated 6 hours ago

Bitnami Helm chart for WordPress

**CONTENT MANAGEMENT SYSTEM** **INTEGRATION & DELIVERY** **SECURITY**

[Learn more](#)

 **secoresearch/wordpress**  ==> **Sponsored Open Source Software** ↓100K+ · ☆0 Pulls: 1,165 Last week

By [secoresearch](#) · Updated 22 days ago

Apache+PHP+Varnish+WordPress

**CONTENT MANAGEMENT SYSTEM** **SECURITY** **WEB SERVERS**

[Learn more](#)



# Soorten images – 2 -

dockerhub

Q httpd

Explore / Official Images / httpd



**httpd** Docker Official Image · ↓1B+ · ☆4.8K

The Apache HTTP Server Project

WEB SERVERS

Overview Tags

## Quick reference

- Maintained by:  
[the Docker Community](#)
- Where to get help:  
[the Docker Community Slack](#), [Server Fault](#), [Unix & Linux](#), or [Stack Overflow](#)

## Supported tags and respective Dockerfile links

- [2.4.62](#) [2.4](#) [2](#) [latest](#) [2.4.62-bookworm](#) [2.4-bookworm](#) [2-bookworm](#) [bookworm](#)
- [2.4.62-alpine](#) [2.4-alpine](#) [2-alpine](#) [alpine](#) [2.4.62-alpine3.20](#) [2.4-alpine3.20](#) [2-alpine3.20](#) [alpine3.20](#)



Image namen worden  
bijvoorbeeld:  
- httpd-latest  
- httpd-2.4.62-alpine

==> Beschikbare varianten van  
images

# Locatie Docker projecten



- Makkelijkst is om alle Docker projecten onder 1 subdirectory te plaatsen van bijvoorbeeld je home directory:
- `mkdir ~/docker`



# Stappen maken Docker container

- 1 -



NLGG  
Nederlandse Linux Gebruikers Groep

- Maak een subdirectory (*mkdir ~/docker/apache*)
- Ga naar deze subdirectory (*cd ~/docker/apache*)
- Maak een configuratiefile **dockerfile**
- Bouw de container
- Maak een subdirectory voor data en neem data op
- Start, test en verwijder de container
- Maak een *compose.yaml* bestand
- Start de container m.b.v. *compose.yaml* bestand



# Stappen maken Docker container

## – 2 -: Dockerfile



- Dockerfile is configuratiebestand voor maken containers
- Elementen onder andere:
  - **FROM** ==> welke Docker image moet gebruikt worden
  - **COPY** ==> hiermee kopieer je bestanden naar het image
  - **RUN** ==> hiermee kun je linux commando's uitvoeren, bijvoorbeeld het toevoegen van software of het aanmaken van een directory
- Dockerfile voor Apache webserver (1 regel):
  - `FROM httpd:latest`



# Stappen maken Docker container

## – 3 -: Container bouwen



- Commando:

```
docker build -t apache:v1 .
```

- **build** ==> opdracht voor maken image
- **-t** ==> hiermee kun je de tag **v1** meegeven. Een tag wordt voorafgegaan door een dubbele punt
- **apache** ==> de naam die je aan de container geeft



# Stappen maken Docker container – 4 -: Datadirectory aanmaken en opnemen data



- Aanmaken directory

- `mkdir apache_data`

- Een test website in de data directory opnemen

- `echo "<html><body><h1>Demo data directory</h1></body></html>" > apache_data/index.html`





# Stappen maken Docker container

## – 5 -: Starten en testen container 1



- Commando:
  - ```
docker run -d -p 8080:80 --name=mijnapache --restart=always -v "${PWD}/apache_data:/usr/local/apache2/htdocs" apache:v1
```
- **-d** ==> detached mode; na ingeven heb je prompt
- **-p 8080:80** ==> poort 80 container is bereikbaar via 8080 host
- **--name=mijnapache** ==> naam van de container
- **--restart=always** ==> container altijd starten na herstarten host



# Stappen maken Docker container

## – 6 -: Starten container 2



- Commando:
  - ```
docker run -d -p 8080:80 --name=mijnapache --restart=always -v "${PWD}/apache_data:/usr/local/apache2/htdocs" apache:v1
```
- **-v "\${PWD}/apache\_data:/usr/local/apache2/htdocs"**  
==> Vanuit de huidige directory is de subdirectory `apache_data` in de container beschikbaar in de directory `/usr/local/apache2/htdocs`
- **apache:v1** ==> image *apache* met tag *v1*



# Stappen maken Docker container – 7 - : Testen en verwijderen container



- Testen in browser:

Meestal  
localhost

- `http://ipadresdockermachine:8080`

- `docker stop mijnapache` ==> stopt de Docker container met de naam *mijnapache*

- `docker rm mijnapache` ==> verwijdert de Docker container met de naam *mijnapache*



# Docker compose – 1 -



- In een `compose.yaml` leg je de configuratie van een Docker container vast
- Vervolgens kun je met één relatief kort commando de Docker container starten.



# Docker compose – 2 -



compose.yaml voor Apache webserver:

- services: ==> configuratie van de containers
- mijnapache: ==> configuratie mijnapache
- image: apache:v1 ==> image dat gebruikt wordt
- restart: always ==> altijd herstarten
- ports:
- 8080:80 ==> poort 80 container is poort 8080 host
- volumes:
- `${PWD}/apache_data:/usr/local/apache2/htdocs`



# Docker compose – 3 -

- Apache webserver met Docker compose starten:
  - `docker compose up -d`
- Apache webserver met Docker compose stoppen:
  - `docker compose down`



# Vragen / opmerkingen?



# Extra: handige docker commando's



- `docker images` ==> laat zien welke Docker images op de computer staan
- `docker ps` ==> laat zien welke Docker processen er lopen
- `docker logs mijnapache` ==> geeft de logs van de Docker container *mijnapache* weer
- `docker image rm apache:v1` ==> verwijdert het Docker image met de naam *apache* en de tag *v1*

