

Docker workshop

Je kunt de Docker workshop op verschillende manieren uitvoeren:

Optie 1: Op je eigen laptop met behulp van Vagrant en VirtualBox.

Optie 2: In je browser met behulp van Katacoda playgrounds. Het voordeel hierbij is dat je niets hoeft te installeren op je laptop en alles kunt uitvoeren in je browser.

Indien je geen software wilt of kunt installeren op je laptop (of problemen hebt met de installatie), kun je het beste kiezen voor optie 2, Katacoda playgrounds.

Optie 1: Docker workshop op je laptop met Vagrant en Virtualbox

Voorwaarden:

- Installeer [Oracle VM VirtualBox](#)
- Installeer [Vagrant](#)
Vagrant is een open-source tool van het maken en onderhouden van virtuele machines voor onder andere VirtualBox en Hyper-V.
- Installeer [GIT](#)
GIT is een gedistribueerde softwareversie beheersysteem.

Nadat je alle software hebt geïnstalleerd, maak je een nieuwe map aan waarvan je middels de Windows command prompt verder werkt.

Vagrant

Je gaat nu een virtual machine aanmaken met Oracle Linux 7 waarin Docker wordt geïnstalleerd. Voor het aanmaken van de virtual machine maken we gebruik van Vagrant.

Voer het volgende uit:

```
git clone https://github.com/oracle/vagrant-boxes
cd vagrant-boxes/DockerEngine
vagrant up
vagrant ssh
```

Met het commando `vagrant up`, wordt een nieuwe VirtualBox machine aangemaakt met Oracle Linux 7. In diezelfde stap wordt de Docker Engine geïnstalleerd. Met het commando `vagrant ssh` log je in op de hiervoor geïnstalleerde virtuele machine. Met `vagrant halt` stop je de virtuele machine en met `vagrant destroy` verwijder je de virtuele machine.

Workshop

Je kunt kiezen uit een aantal workshops.

- **RequestCounter Node application in Docker Run Time on Linux.**

Als eerste moeten we git installeren in de virtual machine.

Voer de volgende stappen uit:

- i. `vagrant ssh`
- ii. `sudo -s`
- iii. `yum install git -y`

Open nu het volgende document: <https://github.com/lucasjellema/2019-fontys-business-and-IT-agility-through-microservice-architecture/blob/master/2-docker/README.MD>.

Voer de stappen die hierin beschreven staat uit op je hiervoor geïnstalleerde virtual machine. In plaats van de stappen uit te voeren op Katacoda playgrounds, voer je de stappen uit in je virtual machine.

- **Je kunt ook een aantal scenario's doorlopen van Katacoda.**

Ga naar: <https://www.katacoda.com/courses/docker>.

De volgende scenario's kun je uitvoeren in je eigen omgeving:

- Deploy Static HTML Website as Container
- Building Container Images
- Dockerizing Node.js applications
- Create Data Containers
- Creating Networks Between Containers using Links
- Creating Networks Between Containers using Networks
- Persisting Data Using Volumes
- Manage Container Log Files
- Ensuring Container Uptime with Restart Policies
- Load Balancing containers
- See Container Metrics with Docker Stats
- Run Docker From Rootless Users
- Formatting PS Output

Op Katacoda zijn nog veel meer scenario's beschikbaar waarmee je verder aan de slag kunt. Kijk bijvoorbeeld eens bij:

- <https://www.katacoda.com/courses/docker-security>
- <https://www.katacoda.com/courses/docker-production>
 - Deploy ELK stack and aggregate container logs
 - Launch private Docker Registry
 - Start Docker containers on boot with system
- <https://www.katacoda.com/courses/docker-performance>
 - Test network performance
 - Test IO performance
- <https://www.katacoda.com/courses/container-runtimes>

- **Kijk op de Linux Academy. Daar zijn ook een aantal Docker workshops beschikbaar.**

Bijvoorbeeld:

- Learn Docker by doing
- Docker – Deep Dive
- Using Volumes in Docker Containers
- Creating your own Docker image
- Using Docker Secrets to manage sensitive data
- Tagging and pushing images to DockerHub
- Docker Networking
- Docker Bench Lab
- Container Logging

Optie 2: Katacoda playgrounds

Ga naar de volgende site: <https://www.katacoda.com/courses/docker>

De volgende scenario's kun je uitvoeren in Katacoda:

- Deploy Static HTML Website as Container
- Building Container Images
- Dockerizing Node.js applications
- Create Data Containers
- Creating Networks Between Containers using Links
- Creating Networks Between Containers using Networks
- Persisting Data Using Volumes
- Manage Container Log Files
- Ensuring Container Uptime with Restart Policies
- Load Balancing containers
- See Container Metrics with Docker Stats
- Run Docker From Rootless Users
- Formatting PS Output

Op Katacoda zijn nog veel meer scenario's beschikbaar waarmee je verder aan de slag kunt. Kijk bijvoorbeeld eens bij:

- <https://www.katacoda.com/courses/docker-security>
- <https://www.katacoda.com/courses/docker-production>
 - Deploy ELK stack and aggregate container logs
 - Launch private Docker Registry
 - Start Docker containers on boot with system
- <https://www.katacoda.com/courses/docker-performance>
 - Test network performance
 - Test IO performance
- <https://www.katacoda.com/courses/container-runtimes>

Kijk op de **Linux Academy**. Daar zijn ook een aantal Docker workshops beschikbaar. Bijvoorbeeld:

- Learn Docker by doing
- Docker – Deep Dive
- Using Volumes in Docker Containers
- Creating your own Docker image
- Using Docker Secrets to manage sensitive data
- Tagging and pushing images to DockerHub
- Docker Networking
- Docker Bench Lab
- Container Logging