

# Opdracht 2e zittijd

## Enterprise Linux 17-18

### Bachelor toegepaste informatica, HoGent Bedrijf en Organisatie

In dit document worden de afspraken opgesomd voor studenten die doorverwezen werden naar de tweede zittijd.

## Contents

<b>1 Algemene afspraken</b>	<b>1</b>
1.1 Aanvullende opdracht: webserver met Docker	1
1.1.1 Opstelling	2
1.2 Evaluatie	2
1.3 Contact tijdens 2e semester, zomervakantie	2
1.4 Info over Docker en Ansible	2

## 1 Algemene afspraken

In de tweede zittijd dien je je bekwaamheid aan te tonen voor alle criteria waarvoor je dat nog niet gedaan hebt. Tijdens het evaluatiemoment of de feedback heb je al afgesproken wat dat concreet voor jou betekent. Je kan ook in je Github-repository een stand van zaken vinden ivm je evaluatie, met onderaan het document een overzicht van nog af te handelen zaken.

- Wie de **hoofdopdracht** nog niet afgewerkt heeft, moet dat alsnog doen
- Idem voor wie de **actualiteitsopdracht** niet heeft gerealiseerd
- Wie ook de derde **troubleshooting-opdrachten** niet met succes heeft kunnen oplossen krijgt voor de tweede zittijd opnieuw een opgave gebaseerd op de laatste (zelfde opstelling, maar met andere configuratiefouten)
- Iedereen krijgt een extra opdracht waarvan de details verderop in dit document volgen.

Van de criteria waarvoor je je bekwaamheid wel aangetoond hebt, wordt de beoordeling overgenomen in de tweede zittijd. Het staat je wél vrij om ook aan deze taken verder te werken of te verbeteren om toch een hogere score te bekomen.

Stel bijvoorbeeld dat je voor de hoofdopdracht “Bekwaam” hebt, dan kan je dit dus nog verhogen door ook de indicatoren voor een hogere beoordeling te realiseren. Hetzelfde geldt voor de rapportering: door die te verbeteren kan je ook voor deze criteria een hogere score bekomen, mits je de indicatoren realiseert.

### 1.1 Aanvullende opdracht: webserver met Docker

De aanvullende opdracht bestaat er uit om een webserver op te zetten met Docker en die te integreren in de opstelling die je nu hebt. Het opzetten wordt uiteraard volledig geautomatiseerd met Ansible.

- In de SME-opdracht wordt dit de interne webserver (hostnaam pr010)
- In de HA-opdracht maak je een nieuwe opstelling volgens de beschrijving hieronder en voer je ook de performantietests uit en vergelijk je die met de LAMP-opstelling.

Docker is een succesvolle implementatie van zgn. containervirtualisatie. In deze vorm van virtualisatie hergebruiken de VMs (containers) de OS-kernel van het hostsysteem. Daardoor zijn containers ook een stuk kleiner en draaien ze met minder overhead. Bij containers geldt typisch dat je slechts één service binnen een container draait. Als je een LAMP-stack wil opzetten, heb je dus minstens twee containers nodig: één met Apache, en één met MariaDB.

### 1.1.1 Opstelling

- Maak een nieuwe VM aan, gebaseerd op base box bertvv/fedora27 (Fedora heeft recentere versies van Docker dan CentOS).
- Installeer Docker op deze VM via Ansible
- Zet binnen deze VM de nodige Docker containers op. Probeer dit eerst manueel zodat je inzicht krijgt in het systeem en automatiseer pas daarna.
  - Database-container met MariaDB of MySQL
  - Web-container(s) met Apache of Nginx, waarop een website draait die de database aanspreekt (bv. Wordpress of Drupal)
  - (Optioneel): Een monitoringsysteem voor het opvolgen van wat er gebeurt in de containers
  - (Optioneel): Een load-balancer (HAProxy) die het netwerkverkeer verdeelt over de containers
- De bedoeling is dat het resultaat is dat je door naar de hostnaam of IP-adres van de VM de website te zien krijgt. Containers zijn typisch op een intern netwerk aangesloten dat niet zomaar van buitenaf toegankelijk is. Je zal dan port-forwardingregels moeten opstellen die verkeer naar de VM doorstuurt naar de containers.

## 1.2 Evaluatie

Maak een labo-rapport op en zorg dat het beschikbaar is op Github. Toon het resultaat tijdens het evaluatiemoment van de derde examenperiode.

- **Nog niet bekwaam:** De opdracht is niet of slechts deels gerealiseerd, de website kan niet getoond worden.
- **Bekwaam:** De website kan getoond worden. Het opzetten van het hostsysteem is volledig geautomatiseerd met Ansible. Het opzetten van de containers is geautomatiseerd, maar niet noodzakelijk met Ansible (maar bijvoorbeeld met Docker compose).
- **Gevorderd:** Ook het opzetten van de containers is geautomatiseerd met Ansible
- **Deskundig:** Naast het vorige niveau zijn extra's gerealiseerd, bijvoorbeeld (eigen ideeën mogen ook, maar spreek dit best af met de lector!):
  - Performantievergelijking met een LAMP-stack
  - Meerdere webcontainers met een load-balancer (HAProxy)
  - Een monitoringsysteem

## 1.3 Contact tijdens 2e semester, zomervakantie

Het is toegelaten om contact op te nemen met de lector tijdens het 2e semester. Als je ergens vast zit, moet je wél een gedetailleerde beschrijving geven van wat precies het probleem is en wat je zelf al geprobeerd hebt om het op te lossen. Geef exacte commando's en eventuele foutboodschappen, geen vage beschrijvingen. Wat ook niet geapprecieerd wordt zijn vragen die neerkomen op "hier is de code, het werkt niet, zoek zelf uit wat precies en waarom". Pas zelf eerst bottom-up troubleshooting toe en maak er een verslag van dat je mee stuurt.

## 1.4 Info over Docker en Ansible

- <https://ansible.github.io/ansible-container-demo/>
- [http://docs.ansible.com/ansible-container/getting\\_started.html](http://docs.ansible.com/ansible-container/getting_started.html)

- <https://github.com/ansible/ansible-container>
- [http://docs.ansible.com/ansible/latest/guide\\_docker.html](http://docs.ansible.com/ansible/latest/guide_docker.html)
  - [http://docs.ansible.com/ansible/latest/list\\_of\\_cloud\\_modules.html#docker](http://docs.ansible.com/ansible/latest/list_of_cloud_modules.html#docker)