

Opdracht 2e zittijd

Enterprise Linux 18-19

Bachelor toegepaste informatica, HoGent Bedrijf en Organisatie

Dit document is bestemd voor studenten die doorverwezen werden naar de tweede zittijd voor dit opleidingsonderdeel. Je vindt hier alle praktische afspraken terug en een beschrijving van de opdracht.

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 van de eerste examenkans of tijdens de feedback heb je al met je begeleider afgesproken wat dat concreet voor jou betekent.

- 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 kleine extra opdracht waarvan de details verderop in dit document volgen.

Van de criteria waarvoor je je bekwaamheid wel aangetoond hebt, blijft je beoordeling uit eerste zittijd behouden. 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 voldoet aan de indicatoren in de evaluatiekaart.

2 Aanvullende opdracht: proof-of-concept Docker

De aanvullende opdracht bestaat er uit om een proefopstelling op te zetten met Docker.

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 proces/service binnen een container draait. Als je een LAMP-stack wil opzetten, heb je dus minstens twee containers nodig: één met de webserver en -applicatie, en één met een databank.

2.1 Opstelling

- Maak een nieuwe VM aan in je Vagrant-omgeving, gebaseerd op de base box bento/fedora29. Deze heeft immers een recentere versie van Docker dan op CentOS beschikbaar is.
- Installeer op deze VM Docker en Cockpit. Cockpit is een webinterface voor het beheer van CentOS/RedHat/Fedora-servers en heeft ook een module voor het monitoren van Docker-containers. Installeer dus zeker ook de geschikte plugin(s)! Zorg uiteraard dat de Docker en Cockpit services automatisch starten bij het booten. Automatiseer heel dit proces met Ansible, zodat je dit indien nodig snel opnieuw kan opzetten.

- Zet binnen deze VM de nodige Docker containers op voor je proof-of-concept. Probeer dit eerst manueel zodat je inzicht krijgt in het systeem. Een interessante tool mee te beginnen is Portainer. Dit geeft je een webinterface waarmee je de mogelijkheden van Docker-containers kan verkennen. Je kan kant-en-klare container images selecteren uit een lijst, die downloaden van Docker Hub (een publieke repository van Docker images, te vergelijken met Vagrant base boxes) en ze opstarten binnen jouw VM.
- Nodige onderdelen van het proof-of-concept:
 - Database-container met MariaDB of MySQL
 - Webapplicatie-container(s) met Drupal
 - Een monitoringsysteem voor het opvolgen van wat er gebeurt in de containers (bv. cAdvisor)
 - (Optioneel): Een load-balancer (bv. HAProxy) die het netwerkverkeer verdeelt over meerdere webapplicatie-containers
 - (Optioneel): Kan je er voor zorgen dat de Drupal site *persistent* is? D.w.z. dat als je de database- én webapplicatiecontainers vernietigt, je de site toch kan reproduceren? Je zal hiervoor wellicht *volumes* moeten gebruiken voor de storage van de database.
- De bedoeling is dat je, als je op je fysieke systeem in een webbrowser surft naar het IP-adres van de VM, je de Drupal-site 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 netwerkverkeer naar de VM doorstuurt naar de juiste container(s).
- Gebruik een load test tool om te simuleren dat gebruikers de website bezoeken. Toon via het monitoringsysteem aan dat je ziet dat het systeem intensiever gebruikt wordt (CPU, geheugen, HTTP-requests, enz.). Verhoog de load totdat het systeem overbelast wordt. Kan je via het monitoringsysteem aantonen waar de bottleneck zit?
- Automatiseer het gehele proof-of-concept met Docker compose.

Hou een laboverslag bij van heel het proces. Dat gaat van het opzoeken van informatie (welke websites, welke info was interessant voor de opdracht), tot instructies voor het reproduceren van je proof-of-concept en testrapport met screenshots. Al je documentatie en de code is beschikbaar in je Github repository vóór de algemene deadline voor het indienen van opdrachten in de tweede zitting (maandag 19 augustus 2019, 12:00u).

3 Evaluatie

Demonstreer het resultaat tijdens het evaluatiemoment van de derde examenperiode. Een rooster van het precieze moment dat we je verwachten wordt tijdig gepubliceerd op Chamilo.

Bij de beoordeling worden volgende criteria gehanteerd voor het bepalen van het bekwaamheidsniveau:

- **Nog niet bekwaam:** De opdracht is niet of slechts deels gerealiseerd, Docker werkt niet correct, de website kan niet getoond worden.
- **Bekwaam:** Het opzetten van het hostsysteem is volledig geautomatiseerd met Ansible. De student kan de werking tonen van een met Docker opgezette Drupal-site, met een database, en een monitoringsysteem. Via het monitoringsysteem is zichtbaar wanneer het systeem (over)belast wordt.
- **Gevorderd:** Ook het opzetten van de containers is geautomatiseerd met Docker compose.
- **Deskundig:** Naast het vorige niveau zijn extra's gerealiseerd, bijvoorbeeld een load-balancer of een persistente database. Eigen ideeën mogen ook, maar spreek dit best af met de lector!

Bij het bepalen van het examencijfer, valt deze opdracht onder dezelfde categorie als de hoofdopdracht. Om een bepaald bekwaamheidsniveau te behalen moet je voldoen aan de voorwaarden van zowel de gekozen hoofdopdracht als deze die hierboven zijn opgesomd.

Het laboverslag wordt beoordeeld onder de categorie "Laboverslagen", volgens dezelfde criteria als toegepast in de eerste examenkans.

4 Info over Docker en Ansible

Vernoemde tools (alfabetisch):

- cAdvisor: <https://prometheus.io/docs/guides/cadvisor/>
- Cockpit: <https://cockpit-project.org/>
- Docker: <https://www.docker.com/>
 - Docker compose: <https://docs.docker.com/compose/>
 - Docker hub (publieke repository voor specifieke container images): <https://hub.docker.com/>
- HAProxy: <http://www.haproxy.org/>
- Portainer: <https://www.portainer.io/>

Webserver load test tools (dit zijn voorbeelden, er bestaan er nog meerdere):

- Apache ab: <https://httpd.apache.org/docs/2.4/programs/ab.html>
- JMeter: <http://jmeter.apache.org/>
- Low Orbit Ion Cannon: <https://sourceforge.net/projects/loic/>