

# Monitoring-tools voor Docker en Kubernetes

## Onderzoeksvoorstel Bachelorproef 2020-2021

Olivier Troch<sup>1</sup>

### Samenvatting

Omdat binnen toegepaste informatica op HoGent geen installatie en gebruik is van monitoring tools, zullen aan de hand van deze bachelorproef verschillende monitoring tools binnen een omgeving getest worden om deze later te gebruiken voor studenten omtrent monitoring bij te leren. Door het willen invoegen van monitoring van containers (e.g. Docker) binnen deze opleiding moet op zoek gegaan worden naar een, bij voorkeur, open source monitoring tool. Om dit te onderzoeken zal er een reproduceerbaar 'proof-of-concept' worden opgesteld waarin meerdere, verschillende, monitoring tools zullen worden getest. In detail zal er in dit onderzoek voornamelijk gekeken worden naar het voortdurend meten van allerlei performantiemetrieken (CPU, I/O, enz.) en wat deze resultaten betekenen. Ook worden de verschillende framework-lagen vergeleken per monitoringtool. Er wordt verwacht om aan de hand van dit onderzoek een werkend en reproduceerbaar 'proof-of-concept' van geschikte open source monitoring-tools op te zetten, dat nadien kan worden gebruikt in het lessenpakket om studenten een basis te geven omtrent monitoring binnenin een docker omgeving. Aan de hand van dit werk zullen toekomstige studenten kunnen bijleren over monitoring.

### Sleutelwoorden

Systeembeheer — Docker — Monitoring

### Co-promotor

Bert Van Vreckem<sup>2</sup> (HoGent)

Contact: <sup>1</sup> olivier.troch.w2257@student.hogent.be; <sup>2</sup> bert.vanvreckem@hogent.be;

## Inhoudsopgave

1	Introductie	1
2	State-of-the-art	1
3	Methodologie	2
4	Verwachte resultaten	2
5	Verwachte conclusies	2
	Referenties	2

## 1. Introductie

Containertechnologie vindt in recordtempo zijn weg naar de data-omgeving van de ondernemingen (Cole, 2016). Het gemak waarmee containerplatformen zoals Docker kunnen worden ingezet, suggereert dat ze de dominantste architectuur zijn en zullen blijven voor deze en de volgende generatie services en microservices. De uitdaging om een goede monitoring van containers op te stellen is belangrijk. Zoals te verwachten, zullen traditionele monitoringsplatforms die vooral gebaseerd zijn op virtualisatie, niet voldoende zijn. Containers zijn zeer vluchtig in ontstaan en net zo snel in het verdwijnen. Die snelheid is vaak te danken aan een geautomatiseerd proces waarbij een minimale input vereist is. Deze containers bevinden zich tussen host -en applicatielaag, wat het moeilijk maakt om in detail te zien hoe ze zich gedragen en of ze efficiënt gebruik maken van hulpbronnen (e.g. CPU, RAM, etc.) en goede systeem prestaties leveren. De valkuil voor

vele bedrijven is het overschatten van een eenvoudige hostmonitoring voor een container. Dit idee wordt echter al snel uit de weg geruimd, omdat het aantal containers snel begint te vergoten wat een traditioneel hostmonitoring niet kan bijhouden. Binnen HoGent is er voorlopig geen opleidingsonderdeel omtrent monitoring, specifiek op een Docker omgeving in een Kubernetes orkestratie. Daar dit een zeer interessant en relevant onderwerp is, is het een goede zaak om deze leerstof bij te leren aan de toekomstige studenten van HoGent. De onderzoeksvraag komt in dit geval uit van een docent tewerkgesteld in HoGent. Omdat er nog geen intern onderzoek heeft plaatsgevonden, is deze onderzoeksvraag dus ontstaan. De doelstelling van dit onderzoek is bepalen welke monitoringtool hiervoor geschikt is en welke haalbaar is voor de studenten.

Zo bekomen we volgende onderzoeksvragen:

- Welke monitoringtools zijn geschikt?
- Wat zijn de belangrijkste verschillen tussen de gekozen tools?
- Waar moet de gekozen tool inzicht op geven?
- Welk proof-of-concept is genoeg om de leerstof te verstaan?

## 2. State-of-the-art

Hoewel er vele onderzoeken zijn naar goede monitoringtools voor Docker, zijn het vooral onderzoeken waarbij ook de betalende tools vergeleken worden (Ribenzaft, 2020) en (Cirelly, 2020). Deze onderzoeken

specificeren zich ook niet op één specifieke 'proof-of-concept' maar leggen vooral de voor- en nadelen van elke tool uit. Hoewel het boek van Alex Williams (Cole, 2016) ook niet echt een 'proof-of-concept' heeft, komt het onderzoek toch dichterbij in de buurt van wat het verwachte resultaat is van deze bachelorproef. Hierin wordt ook over een aantal andere onderwerpen gesproken omtrent monitoring die zeker interessant zijn. Daar een deel van dit boek ook een aantal tools beschrijft en vergelijkt is dit zeker relevant aan de onderzoeksvraag. De bedoeling van deze bachelorproef is het onderzoeken van zowel zelf-gehoste open source-oplossingen als commerciële cloud-gebaseerde services te bekijken, dit boek kaart ook deze opties aan. Volgens het boek van Alex Williams (Cole, 2016) is de conclusie dat de keuze sterk afhangt van de resultaten die je wenst te bereiken die bij uw werklast passen, eventueel met een combinatie van extra tools. Door opzoekingswerk en verzameling van interne informatie worden de vereisten bepaald, zo kan er gekozen worden voor de juiste monitoringtools. Tot dusver vind ik er geen exacte kopie van de onderzoeksvraag, wat deze bachelorproef uniek maakt en interessanter.

### 3. Methodologie

Allereerst zal een MoSCoW requirements-analyse opgesteld worden aan de hand van een interview met mogelijke belanghebbende. Voor dit onderzoek zal vooral de mening van docenten die dit onderwerp zullen voordragen aan de studenten belangrijk zijn. Hierna zal een literatuurstudie opgesteld worden waarin volgende onderwerpen zullen afgehandeld worden:

- Concept monitoring, soorten monitoring, taken van monitoring
- Architectuur monitoringsystemen in de context van containers/Kubernetes
- Overzicht aanbod producten/tools in dit marktsegment

Vervolgens zullen de gevonden producten/tools afgetoetst worden aan de requirements die voordien zijn opgesteld door de belanghebbende. Nadien wordt een bestaande 'proof-of-concept' gebruikt (Cedric, 2019) waarop de meest belovende producten toegepast worden. Indien er in deze 'proof-of-concept' tekortkomingen zijn voor het bekomen van de vooraf bepaalde requirements, zal deze verwerkt worden tot het punt waar er deze tekortkomingen miniem zijn.

### 4. Verwachte resultaten

Uit deze studie, naar een gepaste monitoringtool, wordt verwacht dat aan de hand van de experimenten, het duidelijk wordt welke tool een basis kan zijn voor een deel van het opleidingsonderdeel waarop beslist wordt wat toekomstige studenten kunnen gebruiken. Het opstellen van een goede proof-of-concept is uiterst belangrijk voor het bereiken van goede resultaten.

### 5. Verwachte conclusies

Het experiment zou een goede basis moeten hebben om zo de essentiële onderdelen van monitoring aan te leren aan studenten toegepaste informatica. Met een basis wordt vooral bedoeld dat de 'proof-of-concept' makkelijk reproduceerbaar is en dat de 'beste' monitoringtool en zijn functionaliteiten allemaal uit te voeren zijn. Bovendien is het de bedoeling dat de 'proof-of-concept' alle deelaspecten van monitoring bevat met inzicht op, in geval van dit onderzoek, zicht op het voortdurend meten van allerlei performantiemetrieken. Indien het experiment aan deze eisen voldoet, mogen we concluderen dat dit een geslaagd onderzoek is.

### Referenties

- Cedric, D. (2019). Kubernetes Educatieve Mogelijkheden En Opstellingen. HoGent.
- Cirelly, J. (2020, september 20). 7 Best Docker Monitoring Tools. <https://www.itprc.com/docker-monitoring-tools/>
- Cole, A. (2016, september 29). The Importance of Container Monitoring. <https://www.stratoscale.com/blog/importance-container-monitoring/>
- Ribenzaft, R. (2020, maart 9). Top Six Open Source Tools for Monitoring Kubernetes and Docker. <https://devops.com/top-six-open-source-tools-for-monitoring-kubernetes-and-docker/>