

# Open source tools voor het opzetten, monitoren, integreren van CI/CD en beheren van containers binnen Microsoft Azure: een vergelijkende studie en proof-of-concept.

Onderzoeksvoorstel Bachelorproef 2019-2020

Eggermont Rob<sup>1</sup>

## Samenvatting

Dit onderzoek gaat over de verschillende tools die gebruikt kunnen worden voor containers te beheren, te monitoren en te schalen in een Microsoft Azure omgeving. Dankzij de overvloed aan beschikbare en gebruikte tools geraakt men snel het overzicht kwijt. Dit onderzoek gaat daarom op zoek naar een juiste set van tools om de ontwikkeling, uitrol, monitoring en wijziging van containers een correcte workflow te geven om, op termijn, tijd en kosten te besparen.

## Sleutelwoorden

Applicatieontwikkeling (andere). Containers — CI/CD — Microsoft Azure

## Co-promotor

Joachim Dheedene<sup>2</sup> (delaware)

Contact: <sup>1</sup> rob.eggermont.y7223@student.hogent.be; <sup>2</sup> Joachim.Dheedene@delaware.pro;

## Inhoudsopgave

<b>1</b>	<b>Introductie</b>	<b>1</b>
1.1	Onderzoeksvraag	1
1.2	Onderliggende onderzoeksvragen	1
<b>2</b>	<b>Stand van zaken</b>	<b>1</b>
2.1	CI/CD	1
2.2	Monitoring	2
2.3	Security	2
<b>3</b>	<b>Methodologie</b>	<b>2</b>
<b>4</b>	<b>Verwachte resultaten</b>	<b>2</b>
<b>5</b>	<b>Verwachte conclusies</b>	<b>2</b>

## 1. Introductie

Met oog op de snel veranderende en groeiende wereld van de IT wordt er steeds meer gebruik gemaakt van containers. Deze containers zorgen voor een snelle, flexibele en schaalbare oplossing voor verschillende eisen die gesteld worden binnen de informatica wereld. Door de verschillende, kleine bouwstenen van deze infrastructuur gaat het overzicht echter snel verloren. Deze studie gaat op zoek naar open source tools die het opzetten en monitoren van containers in Microsoft Azure vereenvoudigd. Daarnaast bespreek dit onderzoek ook tools om het beheren van de containers te vereenvoudigen en CI/CD toepassingen te integreren in de workflow.

### 1.1 Onderzoeksvraag

Welke open source tools kunnen efficiënt worden gebruikt om containers op te zetten, te monitoren en te beheren binnen Microsoft Azure?

### 1.2 Onderliggende onderzoeksvragen

- Welke tools geven een duidelijk overzicht van de status van containers?
- Hoe pakken we problemen aan bij containers in productie?
- Hoe kunnen we, door gebruik te maken van CI/CD, containers aanpassen/updaten in productie?
- Welke open source tools en platformen kunnen we gebruiken voor containers in Microsoft Azure?

## 2. Stand van zaken

Vandaag de dag worden er meer en meer (open source) tools beschikbaar gesteld voor het beheren en onderhouden van containers. Waarschijnlijk de belangrijkste tool is Kubernetes. Dit onderzoek gaat op zoek naar hulpprogramma's die een aanvulling kunnen bieden aan Kubernetes (K8's). Er zijn immers veel Tools, bedoelt voor zowel het management als voor developers en systeembeheerders. Deze tools kunnen we onderverdelen in drie categorieën:

### 2.1 CI/CD

Continuous integration and continuous delivery ligt aan de basis voor developers. Het automatiseert het proces van code tot aflevering van de software (container in dit geval). Een eerste belangrijk onderdeel van CI/CD voor containers

zijn de Package Managers. Deze worden geclassificeerd onder CD (continuous delivery) en staan in voor het 'inpakken' van de code naar een containerimage. Sommigen van deze hulpprogramma's staan zelfs in voor het uitrollen van de gemaakte image. Enkele van deze tools zijn:

- Helm
- ksonnet and jsonnet
- Draft

Buiten de package management tools, zijn er verschillende andere tools die instaan voor continuous delivery zoals Weave Cloud, Spinnaker, Codefresh en Harness. Naast de tools voor continuous delivery zijn er die voor continuous integration. Deze stukjes software werden ontwikkeld om unit testen uit te voeren en plaatst de tool de geschreven software bij de rest van de code. Als alle testen slagen kan er aan deze tools gevraagd worden om de container image te maken en in de repository van de applicatie te plaatsen. Vele CI-tools hebben intussen ondersteuning toegevoegd om om deze container images uit te rollen in Kubernetes clusters. De meest gebruikte tools hiervoor zijn:

- Jenkins
- CircleCI
- Travis
- Gitlab

## 2.2 Monitoring

Om een goede werking van services te garanderen is een accurate monitoring belangrijker dan ooit. Monitoring software kan instaan voor rapportering naar de developer toe, maar ook naar andere software toe zodat de ontdekte fouten geautomatiseerd opgelost kunnen worden (indien mogelijk). Enkele voorbeelden zijn:

- Kubebox
- Kubernetes Operational View (Kube-ops-view)
- Kubetail
- Kubewatch
- Weave Scope
- Prometheus
- Searchlight
- cAdvisor
- Kube-state-metrics
- Sumo Logic App
- Dynatrace
- Kubernetes Dashboard

Alle geschreven code moet uiteraard grondig getest worden vooraleer deze uitgerold wordt naar de eindgebruiker. Deze testen kunnen vaak automatisch verlopen. Om dit te laten gebeuren worden er hulpprogramma's gebruikt zoals:

- Kube-Monkey
- K8s-testsuite
- Test-infra
- Sonobuoy
- PowerfulSeal

## 2.3 Security

Containers die in een Kubernetes cluster zitten komen vaak terecht op een shared of een shared-private cloud omgeving terecht. Het is dus van groot belang het netwerk tussen de containers (of pods in Kubernetes) te beveiligen. Dit wordt

verwezelijkt door een bepaald netwerkbeleid in te stellen op een cluster. De 4 meest voorkomende programma's zijn:

- Trireme
- Aporeto
- Twistlock
- Falco
- Sysdig Secure
- Kubesec.io

Dit zijn slechts enkele van de duizenden tools die beschikbaar zijn. Het overzicht is dus eenvoudig te verliezen terwijl er duidelijk nood is aan een 'gouden combinatie'.

## 3. Methodologie

Er zal een grondig onderzoek gevoerd worden naar de werking van de eerder genoemde applicaties. Dit op zowel het vlak van individuele werking als mogelijke samenwerking tussen de programma's onderling. De proof of concept zal bestaan uit een samenhang van tools die kan worden gebruikt van development tot monitoring en herstelling van containers in Kubernetes clusters. Door deze vergelijkende studie uit te voeren tussen de applicaties onderling zullen de voor- en nadelen van deze tools duidelijk worden. De applicaties zullen worden vergeleken op vlak van prestatie en onderlinge samenwerking (compatibiliteit). De belangrijkste parameter is de werking met Microsoft Azure AKS (Azure Kubernetes Service).

## 4. Verwachte resultaten

De verwachte resultaten van de proof of concept zullen de voordelen aantonen van de gekozen toolset voor het volledige ontwikkelproces van containers op Microsoft Azure binnen Kubernetes.

Hier beschrijf je welke resultaten je verwacht. Als je metingen en simulaties uitvoert, kan je hier al mock-ups maken van de grafieken samen met de verwachte conclusies. Benoem zeker al je assen en de stukken van de grafiek die je gaat gebruiken. Dit zorgt ervoor dat je concreet weet hoe je je data gaat moeten structureren.

## 5. Verwachte conclusies

Hier beschrijf je wat je verwacht uit je onderzoek, met de motivatie waarom. Het is **niet** erg indien uit je onderzoek andere resultaten en conclusies vloeien dan dat je hier beschrijft: het is dan juist interessant om te onderzoeken waarom jouw hypothesen niet overeenkomen met de resultaten.