

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 Rob¹

Samenvatting

Dit onderzoek gaat over de verschillende tools die gebruikt kunnen worden om containers te beheren, te monitoren en te schalen in een Microsoft Azure Kubernetes omgeving. Dankzij de overvloed aan beschikbare 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 in Kubernetes een correcte workflow te geven om, op termijn, tijd en kosten te besparen.

Sleutelwoorden

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

Co-promotor

Joachim Dheedene² (delaware)

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

Inhoudsopgave

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

1. Introductie

Met oog op de snel veranderende en groeiende wereld van de IT wordt er steeds meer gebruikgemaakt van containers. 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 vereenvoudigt. Daarnaast bespreekt 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 in termen van snelheid en verbruikte resources om containers op te zetten, te monitoren en te beheren binnen Microsoft Azure?

1.2 Onderliggende onderzoeksvragen

- Welke tools geven een bruikbare informatie over de status en de gezondheid van containers?
- Hoe zorgen we ervoor dat containers op de gewenste manier blijven werken in productie omgevingen wanneer er zich fouten voordoen?
- 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. Eén van de meest geanticipeerde tools is Kubernetes. Dit onderzoek gaat op zoek naar hulpprogramma's die een aanvulling kunnen bieden aan Kubernetes (K8's). Er zijn immens veel tools, bedoeld voor zowel het management als voor ontwikkelaars en systeembeheerders. Deze tools kunnen we onderverdelen in drie categorieën:

2.1 CI/CD

Continuous integration and continuous delivery ligt aan de basis voor ontwikkelaars. 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. (Anita; 2018a) 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. Alle developers werken in éénzelfde repository. Bij elke check-in wordt de ingevoerde code gecontroleerd door een automatisch gestart proces. Dit proces behoort tot CI. Als alle taken 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 deze container images uit te rollen in Kubernetes clusters.

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, waar mogelijk, geautomatiseerd opgelost kunnen worden. (Anita, 2018b) 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:

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 verwezenlijkt door een bepaald netwerkbeleid in te stellen op een cluster. Er zijn 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 onderzoek gevoerd worden op basis van snelheid en gebruikte resources voor de 5 meest populaire tools per categorie. 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 een Kubernetes cluster. Daarnaast wordt aangetoond dat de bekomen resultaten beter zijn dan die voor een andere kandidaat toolset. De resultaten zullen bestaan uit zowel de snelheid van de gekozen tools als de gebruikte resources. Als extra kunnen het aantal fouten die voorkomen gebruikt worden als doorslaggevende factor.

5. Verwachte conclusies

Na het uitvoeren van dit onderzoek verwachten we, bij benadering, een perfecte toolset voor het ontwikkelen en uitrollen van containers. Daarnaast breiden we deze toolset uit met programma's en bijhorende dashboard die containers monitoren op verschillende soorten problemen zoals te weinig geheugen, netwerkproblemen, ...

Referenties

- Anita, B. (2018a). Top 11 Continuous Delivery Tools for Kubernetes (Part 1). Verkregen van <https://www.weave.works/blog/continuous-delivery-tools-for-kubernetes-part-1>
- Anita, B. (2018b, maart 23). Top 11 Continuous Delivery Tools for Kubernetes (Part 2). Verkregen van <https://dzone.com/articles/top-11-continuous-delivery-tools-for-kubernetes-pa>