

Cloud platformen en software release management vergeleken

Onderzoeksvorstel Bachelorproef 2019-2020

Kenzie Coddens¹

Samenvatting

In deze paper wordt er gekeken naar de verschillende cloud platformen en manieren om software release management te implementeren. Deze vraag komt er vanuit een specifieke use case die voorgesteld is door Aucxis. Aucxis heeft reeds een Azure omgeving en wenst om de alternatieven te bekijken. In deze paper wordt er een duidelijke vergelijking gemaakt tussen de verschillende cloud platformen. Ook worden er experimenten uitgevoerd omtrent verschillende probleempunten en uitdagingen.

Sleutelwoorden

Systeembeheer. IaaS — TaaS — Cloud testing — Testing — Software release management

Co-promotor

Raf Boon² (Aucxis)

Contact: ¹ kenzie.coddens.y1678@student.hogent.be; ² Raf.Boon@aucxis.com;

Inhoudsopgave

1	Introductie	1
2	Literatuurstudie	1
2.1	Conventional Software Testing Vs. Cloud Testing	1
2.2	Software Testing Based on Cloud Computing	2
2.3	Benchmarking in the Cloud: What It Should, Can, and Cannot Be	2
2.4	When to Migrate Software Testing to the Cloud?	2
3	Methodologie	2
4	Verwachte resultaten	3
4.1	Vergelijking van platformen	3
4.2	Proof of concept	3
4.3	beveiliging experiment	3
5	Verwachte conclusies	3
5.1	Vergelijking van platformen	3
5.2	Proof of concept	3
5.3	beveiliging experiment	3
	Referenties	3

1. Introductie

Software release management kan op verschillende manieren geïmplementeerd worden. Dit zijn vaak complexe en zeer use case specifieke omgevingen. Ook testen en debuggen van software en infrastructuur zijn belangrijk voor het afleveren van kwalitatieve producten. Een vraag die hierbij opduikt is of dit op lokale infrastructuur moet gebeuren of op cloud platformen die IaaS (Infrastructure as a Service), TaaS (Testing as a Service) of SaaS (Software as a Service) aanbieden.

Der bestaan al tientallen papers over de performantie, flexibiliteit, enz. in een cloud omgeving. Ook over software release management zijn er al talloze papers geschreven. Des ondanks is het toch nog interessant om dit voor een specifieke use case te bekijken. Aucxis heeft al een software release omgeving op Azure. Ze hebben echter geen onderzoek gedaan naar andere cloud platformen of oplossingen. Deze paper is in de eerste plaats bedoelt om voor Aucxis een duidelijk beeld te scheppen over de mogelijkheden.

Deze paper zal proberen in detail duidelijkheid te scheppen over wanneer een cloud platform een goede keuze is en wanneer niet, wat de beste tools zijn, hoe het zit met de gebruiksvriendelijkheid en de prijs. Ook de performantie is niet onbelangrijk. Deze paper zal zich ook afvragen hoe data privacy kan gecontroleerd en geïmplementeerd worden. Op het einde van deze paper zal er een conclusie gemaakt worden over welke optie het best past binnen de use case van Aucxis.

2. Literatuurstudie

2.1 Conventional Software Testing Vs. Cloud Testing

Dit artikel (Katherine & Alagarsamy, 2012) snijdt oppervlakkig aan wat pijnpunten kunnen zijn voor testen van software in een cloud platform. Hier gebruiken ze een web applicatie als voorbeeld. Het artikel stelt een aantal punten voor, waarop getest kan worden. Dit zijn de traditionele test cases. Functionaliteit testen, gebruiksvriendelijkheid testen, interface testen, compatibiliteitstesten, performantie testen en tot slot veiligheid en privacy testen. Het artikel stelt een aantal uitdagingen voor bij lokale omgevingen. Dit gaat over de kost, over het onderhoud ervan, hoe eenzijdig

een lokale omgeving is en voor ieder project een nieuwe omgeving gebouwd moet worden en het feit dat het geen accurate weergave is van de werkelijke omgevingen waarin de software zal draaien.

Verder legt het artikel kort uit wat voor mogelijke cloud oplossingen er op dat moment zijn. Eerst moet er onderscheid gemaakt worden in hoe een cloud platform uitgerold kan worden. Er bestaat enerzijds de publieke cloud (Google Cloud Platform, AWS, Azure, DigitalOcean) en anderzijds een privé cloud. De privé cloud is een lokale opstelling die beschikbaar is over het internet naar andere gebruikers. Ook bestaat er iets zoals een hybride cloud. Hierna wordt er dan nog onderscheid gemaakt tussen welke services deze platformen kunnen aanbieden. Dit artikel beschrijft er drie. SaaS (service as a Service), PaaS (Platform as a Service), IaaS (Infrastructure as a Service). Het artikel maakt toch een onderscheid van het traditionele testen. Aangezien het in de cloud mogelijk is om de applicatie te testen op load, stress en capaciteit.

Tot slot stelt dit artikel belangrijke uitdagingen aan het licht. Zo is de beveiliging van de data die ontvangen, verstuurd of bewaard worden op een cloud platform belangrijk. Ook zijn er over alle platformen heen weinig tot geen standaarden vastgelegd over zowel de performantie van de systemen als de beschikbaarheid op vlak van aanbod. Het zijn juist deze uitdagingen die belangrijk kunnen zijn voor de onderzoeksvragen.

2.2 Software Testing Based on Cloud Computing

In dit artikel (Jun & Meng, 2011) wordt er opnieuw in detail beschreven wat de verschillende platform mogelijkheden zijn zoals IaaS, PaaS, SaaS. Het artikel probeert ook een definitie te geven aan testen op cloud platformen. Tevens geeft het artikel ook een aantal redenen waarom cloud testen een stuk beter zou zijn dan het testen in lokale omgevingen. Deze komen grotendeels overeen met het vorige artikel. In dit artikel wordt er ook besproken dat beveiliging een groot probleem kan zijn. Juist zoals het vorige artikel wordt er gesteld dat het een echte uitdaging is om test datasets in de cloud te gebruiken aangezien deze meestal afkomstig zijn van een klant. Het artikel bespreekt ook een aantal mogelijkheden om met de cloud te verbinden en testomgevingen te configureren. Het artikel bespreekt vooral virtualisatie.

Dit artikel bevestigt deels het vorige artikel. Het geeft wat detail en inzicht in cloud testen. Dit artikel sluit aan met de onderzoeksvragen en geeft richting in probleemgebieden.

2.3 Benchmarking in the Cloud: What It Should, Can, and Cannot Be

Zomaar willekeurig testen of experimenten uitvoeren is meestal geen goed idee. Er is nood aan een goed gedefinieerde methode om deze testen uniform uit te voeren. Dit artikel (Folkerts e.a., 2013) beschrijft in extreem detail hoe een cloud platform het best getest kan worden. Er wordt beschreven wat de valkuilen zijn bij performantietesten van een cloud platform. Zo wordt het testen van een lokale omgeving vergeleken met het testen van een cloud omgeving.

Dit is een hele uitdaging aangezien de hardware van een cloud platform meestal verschilt en niet hetzelfde is. Het artikel beschrijft het testen van een cloud platform aan de hand van een aantal use cases. Het artikel gebruikt hiervoor use cases die schaalbaar zijn en in pieken benaderd worden. Ook beschrijft het artikel dat het belangrijk is om goed te definiëren wat er allemaal getest moet worden over de verschillende platformen heen.

Dit artikel biedt een gedetailleerd inzicht in het opstellen van benchmarks voor cloud omgevingen en zal een belangrijke leidraad vormen voor het opstellen van de experimenten.

2.4 When to Migrate Software Testing to the Cloud?

Wanneer moet er gedacht worden om naar een cloud omgeving te migreren? Dit artikel (Parveen & Tilley, 2010) beschrijft vanaf wanneer het nuttig is om naar de cloud te migreren. Ook beschrijft het artikel kort wat de ervaring was bij een migratie. Dit artikel is interessant omdat het kort een inzicht geeft in wanneer het nuttig en efficiënt is om naar een cloud te migreren. Dit is interessant omdat dit aansluit bij de probleemstelling of een cloud omgeving voor testen nu zoveel beter kan zijn dan een lokale omgeving.

3. Methodologie

Een groot deel van de onderzoeksvragen zullen beantwoord worden door onderzoekswerk en vergelijkingen. Zo zal deze paper in detail bespreken welke cloud platformen er bestaan en wat de mogelijk plannen (tarieven en voor gedefinieerde configuraties) zijn. De verschillende platformen zullen op een duidelijke manier naast elkaar gelegd worden en vergeleken worden. Ook zal er gekeken worden naar software release tools. Op basis hiervan zal er dan een keuze gemaakt worden welke platformen er in aanmerking komen voor een proof of concept.

Ook zal deze paper methodes beschrijven en testen door middel van experimenten wat betreft het behouden van data privacy. Deze paper zal bijvoorbeeld een experiment uitvoeren met een proxy (een tunnel met encryptie naar het datacenter) om de gebruiksvriendelijkheid hiervan te testen.

Na het onderzoek zal er een proof of concept opgezet worden met beste alternatief. Er zal getracht worden om de huidige omgeving van Auctis zo goed mogelijk te benaderen in functionaliteit. De bedoeling is om dezelfde tools te gebruiken om de omgevingen te monitoren (bijvoorbeeld: een Docker image voor dezelfde configuratie en Telegraf en Grafana voor monitoring). Ook zal er getracht worden om dezelfde testen te gebruiken. Dit alles zal over een bepaalde periode draaiende gehouden worden waarna alle resultaten gebundeld zullen worden. Hierbij zitten ook een aantal subjectieve waarnemingen aangezien er ook onderzoek zal gedaan worden naar gebruiksvriendelijkheid. In deze proof of concept zal er een alternatief platform vergeleken worden met het huidige systeem.

4. Verwachte resultaten

4.1 Vergelijking van platformen

Er wordt verwacht dat de huidige cloud omgeving vanuit de use case van Aucxis de beste oplossing is, zeker op vlak van gebruiksvriendelijkheid en efficiëntie. Ondanks dit wordt er verwacht dat de alternatieven een even goede oplossing zullen aanbieden. Deze zullen waarschijnlijk niet de meest gebruiksvriendelijkste of goedkoopste oplossingen zijn. Voor de tools voor software release management wordt er verwacht dat er gelijkaardige alternatieven aan de huidige tools vanuit de use case gevonden worden.

4.2 Proof of concept

Er wordt verwacht dat er een werkende alternatieve omgeving opgezet zal worden die de huidige functionaliteit benaderd. Er wordt verwacht dat de performantie van dit platform op z'n minst gelijkaardig is aan de huidige use case. Er wordt verwacht dat de gebruiksvriendelijkheid en de kost verbeteren.

4.3 beveiliging experiment

Voor dit experiment zijn er gemengde verwachtingen. Vooral op het vlak van tijdrovende configuraties. Er wordt verwacht dat een proxy het meest flexibel is en het meest gebruiksvriendelijk, zeker als het vergeleken wordt met encryptie of andere tools.

5. Verwachte conclusies

5.1 Vergelijking van platformen

Het is moeilijk om een conclusie te voorspellen over welk cloud platform het beste uit de vergelijkingen zal komen. Ook is het moeilijk te voorspellen wat de beste tools zullen zijn. Er wordt wel verwacht dat er een degelijk alternatief voor de huidige oplossing gevonden zal worden.

5.2 Proof of concept

De conclusie zal voor dit experiment zeer duidelijk zijn. Deze zal afhangen van de werking van het alternatief. Is het alternatief sneller, gebruiksvriendelijker, enz. dan zal deze conclusie positief zijn. Anders niet.

5.3 beveiliging experiment

Om de data privacy te garanderen wordt er verwacht dat een proxy de beste en meest doelmiddelke oplossing zal zijn. Het valt moeilijk te zeggen of andere tools of methodes beter zullen presteren.

Referenties

- Folkerts, E., Alexandrov, A., Sachs, K., Iosup, A., Markl, V. & Tosun, C. (2013). Benchmarking in the cloud: what it should, can, and cannot be. *Springer-Verlag Berlin Heidelberg*. TPCTC 2012, LNCS 7755.
- Jun, W. & Meng, F. (2011). Software testing based on cloud computing. 978-0-7695-4539-4.

- Katherine, M. & Alagarsamy, D. K. (2012). Conventional software testing vs cloud testing. *International Journal Of Scientific & Engineering Research*, 3(9).
- Parveen, T. & Tilley, S. (2010). When to migrate software testing to the cloud? *Third International Conference on Software Testing, Verification, and Validation Workshops*. DOI 10.1109/ICSTW.2010.77 IEEE.