

Container hosting Microsoft Azure

Afstudeerverslag

ilionx
Your partner in digital business

Auteur	| W.J.H. Damman
Begeleider	| P. van der Veen
Studentnummer	| 431093
Datum	| 12-6-2020
Versie	| 2.0

1. Voorwoord

Voor u ligt het afstudeeronderzoek Container hosting Microsoft Azure, welke is gemaakt ter afsluiting van de opleiding HBO-ICT Service Management van Saxion Hogeschool te Deventer. Dit onderzoek is uitgevoerd bij ilionx Groningen op de unit Cloud Solutions. Dit onderzoek is uitgevoerd tussen februari 2020 en juni 2020. In dit onderzoek is onderzocht welke containerplatform het beste bij een container gebaseerde applicatie past. Als beroepsproducten zijn een flyer en een whitepaper ontwikkeld. Deze producten moeten klanten leidraad bieden over hun keuze voor een container platform binnen Microsoft Azure.

Tijdens het onderzoek heb ik begeleiding gehad van mijn begeleider van ilionx Peter van der Veen. Hij heeft mij ondersteund bij het uitvoeren van dit onderzoek. Verder stelde hij kritische vragen en gaf richting aan het onderzoek. Ik wil hem bedanken voor zijn inzet en hulp.

Daarnaast wil ik graag mijn afstudeerbegeleider vanuit Saxion Hogeschool te Deventer Tonny Lievers bedanken. Hij heeft mij tweewekelijks voorzien van feedback over de verschillende documenten en voortgang van mijn afstudeerproces. Zijn kritische blik heeft mij verschillende nieuwe inzichten gegeven. Ik wil hem voor zijn hulp en inzet ook in het bijzonder bedanken. Ook mijn tweede beoordelaar Theo Lansink heeft mij voorzien van extra feedback, waarvoor mijn dank.

Tot slot wil ik ook alle overige medewerkers van ilionx, voornamelijk unit Cloud Solutions Groningen, bedanken voor hun inzet en ter ondersteuning van mijn afstudeeronderzoek. Ik heb mijn afstudeerperiode als een leuke en leerzame tijd ervaren binnen ilionx, zowel op kantoor als de bijeenkomsten via Microsoft Teams.

Wouter Damman

Assen, juni 2020

2. Samenvatting

De unit Cloud Solutions van ilionx Groningen liep tegen het probleem aan dat zij niet goed voorbereid waren op een vraag van een klant over de hosting van een container gebaseerde applicatie. Om in het vervolg goed voorbereid te zijn op dergelijke vragen, was de vraag van Cloud Solutions om hiervoor een whitepaper te ontwikkelen. Aan de hand hiervan moeten klanten geadviseerd worden over container hosting binnen Microsoft Azure. Hierbij is de volgende hoofdvraag geformuleerd: *"Wat is de meest passende geautomatiseerde infrastructuur oplossing voor mijn container gebaseerde applicatie binnen Microsoft Azure?"*

Cloud Solutions wilde ook onderzocht hebben hoe deze container infrastructuur automatisch geïmplementeerd kan worden. Aan de hand van literatuuronderzoek is kennis opgedaan over de verschillende container platformen binnen Microsoft Azure. In dit afstudeeronderzoek zijn alleen Container as a Service en Infrastructure as a Service onderzocht, en zijn Platform as a Service en Function as a Service niet onderzocht. De container platformen Virtual Machine met Docker, Azure Kubernetes Services en Azure Red Hat OpenShift zijn hierbij onderzocht. Vervolgens is een multicriteria-analyse gemaakt voor vergelijking van deze container platformen. Aan de hand van deze analyse zijn ook implementatiescenario's gemaakt. Door het gebruikte implementatiescenario komt uit de multicriteria-analyse naar voren dat Azure Kubernetes Services in dit onderzoek het beste passend is. Bij een andere klantvraag, met dus andere een implementatiescenario, is de kans aanwezig dat er een ander containerplatform beter passend is. De implementatiescenario's zijn vervolgens verder ontwikkeld naar workflows. In een Proof of Concept is één van de workflows uitgewerkt naar een playbook, waarin aangetoond is dat deze functioneel is. Het automatisch uitrollen van de containerinfrastructuur gebeurt door middel van Azure Resource Manager, waarbij gebruik wordt gemaakt van Azure DevOps.

In het verloop van het onderzoek kwam naar voren dat ook een flyer gemaakt moest worden, welke informeert over een Containerscan waarmee onderzocht wordt welke containerplatform het best aansluit op de container gebaseerde applicatie van een klant. Hiernaast is een whitepaper ontwikkeld waarin de lezer geïnformeerd en geadviseerd wordt over de verschillende container platformen binnen Microsoft Azure.

Met het afronden van de afstudeeropdracht beschikt de unit Cloud Solutions van ilionx Groningen over een flyer en whitepaper waarmee de klant in het vervolg beter kan worden geholpen met hun vraag over hosting van een container gebaseerde applicatie. Hiernaast heeft de unit aan de hand van dit onderzoek, en door de whitepaper en flyer, ook kennis opgedaan van container hosting en over het automatisch implementeren van infrastructuren door middel van Azure Resource Manager en Azure DevOps.

Er zijn naar aanleiding van dit onderzoek ook een aantal aanbevelingen beschreven welke mogelijk tot een vervolgonderzoek kunnen leiden. De eerste aanbeveling is het gebruik van Terraform voor het automatisch uitrollen van de infrastructuur in plaats van Azure Resource Manager. Hiernaast wordt aanbevolen om een vervolgonderzoek te doen naar het gebruik van Platform as a Service- en Function as a Service-platformen binnen Microsoft Azure, naast alleen de Container as a Service- en Infrastructure as a Service-platformen in dit onderzoek. Ten slotte wordt aanbevolen om het personeel van de unit Cloud Solutions van ilionx Groningen te voorzien van scholing op het gebied van container infrastructuren.

3. Inhoud

1. Voorwoord 2
2. Samenvatting 3
4. Figuren- en tabellenlijst 6
   4.1 Figurenlijst 6
   4.2 Tabellenlijst 6
5. Inleiding 8
   5.1 Probleemstelling 8
   5.2 De organisatie 8
6. Begrippenlijst 9
7. Methode van het onderzoek 11
   7.1 Probleemanalyse 11
   7.2 Behoefteanalyse 11
   7.3 Contextanalyse 11
   7.4 Literatuuronderzoek 12
   7.5 Ontwerpdocumentatie 12
8. Resultaten van het onderzoek 13
   8.1 Probleemanalyse 13
      8.1.1 Hoofd- en deelvragen 14
      8.1.2 Business requirements 14
   8.2 Behoefteanalyse 15
   8.3 Stakeholder inventarisatie 15
   8.4 Stakeholder selectie 16
      8.4.1 Requirements 16
   8.5 Contextanalyse 18
      8.5.1 Randvoorwaarden 18
   8.6 Literatuuronderzoek 19
      8.6.1 Deelvraag 1: Containers 19
      8.6.2 Deelvragen 2, 3 en 4: Onderzoek naar containerplatformen 20
      8.6.3 Deelvraag 5: Benodigdheden geautomatiseerde uitrol infrastructuur 30
   8.7 Ontwerpdocumentatie 32
      8.7.1 Functioneel ontwerp 32
      8.7.2 Technisch ontwerp 38
   8.8 Bedrijfsproducten 46
      8.8.1 Flyer 46
      8.8.2 Whitepaper 46
9. Evaluatie van het onderzoek 47
      9.1.1 Evaluatie hoofd- en deelvragen 47
      9.1.2 Evaluatie requirements 47
      9.1.3 Evaluatie randvoorwaarden 50
10. Conclusie van het onderzoek 51
11. Discussie 52
   11.1 Validiteit 52
   11.2 Resultaten interpreteren 52
   11.3 Beperkingen 52
   11.4 Implicaties 53
   11.5 Aanbevelingen en vervolgonderzoek 53
12. Nawoord 54
13. Bibliografie 55
14. Versiebeheer 59

15. Bijlage: het vooronderzoek 60
15.1 Probleemanalyse 60
15.1.1 Methode 60
15.1.2 Resultaten 61
15.1.3 Conclusie 63
15.2 Behoefteanalyse 64
15.2.1 Methode 64
15.2.2 Resultaten 65
15.3 Contextanalyse 66
15.3.1 Methode 66
15.3.2 Resultaten 67
15.3.3 Conclusie 70
15.4 Literatuuronderzoek 71
15.4.1 Methode 71
15.4.2 Resultaten 71
15.4.3 Conclusie 78
15.5 Bijlagen bij het vooronderzoek 79
15.5.1 Bijlagen probleemanalyse 79
15.5.2 Bijlagen contextanalyse 85
15.5.3 Bijlagen literatuuronderzoek 88
16. Bijlage: methode ontwerpdocumentatie 90

4. Figuren- en tabellenlijst

4.1 Figurenlijst

Figuur 1: Units Groningen 8
Figuur 2: Causaal veldmodel 13
Figuur 3: Influence- en interest stakeholdermatrix 16
Figuur 4: Stack VM en containers 19
Figuur 5: Voorbeeld e-commerceapplicatie 21
Figuur 6: ARM 30
Figuur 7: JSON script 31
Figuur 8: Workflow uitrollen voorbeeldapplicatie 36
Figuur 9: Workflow AKS 37
Figuur 10: Workflow ARO 37
Figuur 11: Workflow VM met Docker 38
Figuur 12: Deployment infrastructuur 39
Figuur 13: Te deployen infrastructuur 39
Figuur 14: Deployment applicatie 40
Figuur 15: Dockerfile 40
Figuur 16: Frontpage container gebaseerde applicatie 40
Figuur 17: AKS.deploy.json 41
Figuur 18: AKS deployment file 41
Figuur 19: Azure DevOps pipelines functie 42
Figuur 20: Azure Repos 42
Figuur 21: Aanspreken gedeelde variabelen 42
Figuur 22: Aanmaken resource group 42
Figuur 23: De twee fases binnen de pipeline 43
Figuur 24: REST API POST connectie ACR 44
Figuur 25: Build and push image to container registry 44
Figuur 26: Services manifest 45
Figuur 27: Deployment manifest 45

4.2 Tabellenlijst

Tabel 1: Hoofdvraag 14
Tabel 2: Deelvragen 14
Tabel 3: Business requirements 14
Tabel 4: Betrokken stakeholders 15
Tabel 5: Requirements opgedeeld per stakeholder 17
Tabel 6: Randvoorwaarden 18
Tabel 7: Kosten AKS 24
Tabel 8: Kosten ARO 26
Tabel 9: Kosten VM met Docker 29
Tabel 10: Beschikbaarheid in percentages 32
Tabel 11: Uitleg beoordeling 33
Tabel 12: Uitleg kleuren multicriteria-analyse 33
Tabel 13: Multicriteria-analyse container oplossingen 33
Tabel 14: Van klantvraag naar implementatiescenario 34
Tabel 15: Evaluatie hoofdvraag 47

Tabel 16: Evaluatie deelvragen 47
Tabel 17: Evaluatie requirements 49
Tabel 18: Evaluatie randvoorwaarden 50
Tabel 19: Versiebeheer 59
Tabel 20: Componenten SQL database 74
Tabel 21: Componenten MongoDB vs. CosmosDB 76
Tabel 22: Componenten Azure API Gateway 77
Tabel 23: Gebruikte zoektermen 88
Tabel 24: Gebruikte informatiebronnen en databanken 88

5. Inleiding

Voor het hosten van een container gebaseerde applicatie zijn verschillende mogelijkheden binnen Microsoft Azure, maar welke van deze is het beste in welke situatie? Cloud Solutions Groningen (GRO-Cloud Solutions), een afdeling van ilionx Groningen, wil haar klanten hierover infomeren door middel van een whitepaper. De whitepaper moet klanten leidraad bieden over op welke manier deze gehost kan worden en welke andere benodigdheden hierbij nodig zijn.

Aan de hand van een whitepaper wordt de klant geadviseerd over welk type hosting het beste bij hun container gebaseerde applicatie past. Naast dat GRO-Cloud Solutions een whitepaper wil ontwikkelen, moet voor verschillende situaties een workflow van het implementatieproces worden ontwikkeld. Aan de hand hiervan wil GRO-Cloud Solutions begrijpelijk aan de klant uitleggen kunnen hoe zijn applicatie geautomatiseerd geïmplementeerd wordt binnen Microsoft Azure. ilionx is dus in dit onderzoek de opdrachtgever.

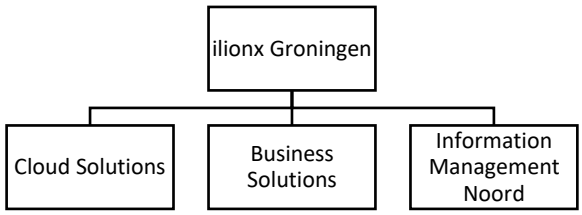
5.1 Probleemstelling

GRO-Cloud Solutions is op zoek naar een manier om haar klanten op een juiste manier te kunnen informeren over welke hostingmogelijkheden er zijn voor container gebaseerde applicaties. Deze vraag is naar voren gekomen nadat een klant van GRO-Cloud Solutions een vraag had over een container gebaseerde applicatie. GRO-Cloud Solutions was hier niet goed op voorbereid en beschikte niet over de juiste kennis om hier antwoord op te geven. Aan de hand van dit afstudeeronderzoek willen zij zelf meer kennis op doen over container gebaseerde applicaties, en daarnaast met behulp van een whitepaper haar klanten eenvoudig informeren over de verschillende mogelijkheden hiervoor binnen Microsoft Azure.

5.2 De organisatie

ilionx is een ICT-diensterverlener, welke is opgericht in 2002. ilionx helpt haar klanten op het gebied van artificial intelligence, management consultancy, business analytics, Cloud, mobile, security en professional services. Het bedrijf heeft ruim 900 medewerkers in dienst en is gevestigd in Groningen, Amsterdam, Utrecht, Zwolle, Maastricht, Den Bosch, Eindhoven en Rotterdam. De afstudeeropdracht heeft plaatsgevonden bij ilionx Groningen.

Binnen ilionx Groningen zijn weer verschillende units, welke zich met verschillende expertises van ilionx bezig houden (zie Figuur 1). De afstudeeropdracht vond plaats bij de divisie Cloud Solutions.



Figuur 1: Units Groningen

Cloud Solutions houdt zich bezig met de implementatie en beheer van zowel on-premise- als Cloud infrastructuren. Het doel van Cloud Solutions is de zorgen van haar klanten over ICT weg te nemen, zodat de klant zich bezig kan houden met zijn corebusiness.

6. Begrippenlijst

5W1H (+ laddering): Een methode waarbij veel vragen worden gesteld om het probleem beter te begrijpen. De vragen die worden gesteld zijn gebaseerd op de 5W1H methode (Wie, Wat, Waar, Wanneer, Waarom en Hoe).

Aggregatieniveaus: Niveau binnen de contextanalyse, bijvoorbeeld: het individuele stuk, het dossier of de zaak, het zaaktype of het onderwerp.

Application programming interface (API): Een API is een software-interface die het mogelijk maakt dat twee applicaties met elkaar kunnen communiceren.

Container as a Service (CaaS): Een Cloud servicemodel waarmee gebruikers containers, applicaties en clusters kunnen uploaden, organiseren, starten, stoppen, schalen en anderszins kunnen beheren.

Causaal veldmodel: Een causaal veldmodel geeft de oorzaak en gevolg tussen twee gebeurtenissen weer.

Command-line-interface (CLI): Een interface die werkt aan de hand van commandoregels. Voorbeelden hiervan zijn Command prompt, PowerShell en Bash.

Cloud computing: Via een netwerk op aanvraag beschikbaar stellen van hardware, software en gegevens.

Container gebaseerde applicatie: Een applicatie die geïsoleerd wordt gehost binnen een container.

Continuous Deployment (CD): Het CD-proces is een aanvulling op het Continuous Integration-proces. Bij CD wordt de geteste code in productie gezet.

Continuous Integration (CI): Het CI-proces, test en valideert de code wanneer wijzigingen optreden.

Debuggen: Debuggen is het opsporen en verhelpen van bugs in computerprogramma's en elektronische hardware.

Function as a Service (FaaS): FaaS verwijst naar een Cloud service die de ontwikkeling en het beheer van serverloze (serverless computing) applicaties mogelijk maakt.

Groningen Cloud Solutions (GRO-Cloud Solutions): GRO-Cloud Solutions is een unit binnen ilionx Groningen die zich bezig houdt met Cloud infrastructuur.

HashiCorp Configuration Language (HCL): Een configuratie taal gebouwd door HashiCorp, met als doel om een gestructureerde configuratietaal te bouwen. HCL kan gebruikt worden binnen een CLI, maar is specifiek gemaakt voor DevOps-tools en servers.

Hypervisor: Een hypervisor zorgt ervoor dat meerdere besturingssystemen tegelijkertijd op een hostcomputer kunnen draaien.

Infrastructure as a Service (IaaS): IaaS is een vorm van Cloud computing. De infrastructuur wordt afgenomen bij een Cloud provider.

JavaScript Object Notation (JSON): JSON is een gestandaardiseerd gegevensformaat.

Microservices: Microservices is een software-ontwikkelingstechniek, een variant van de structurele structuur van de servicegeoriënteerde architectuur, die een applicatie rangschikt als een verzameling los gekoppelde services.

Microsoft Azure: Microsoft Azure is een Cloud-platform van Microsoft.

MoSCoW methode: De MoSCoW methode is een manier van prioriteiten stellen binnen requirements (Must, Should, Could en Won't).

Orchestration-services: Orchestration zorgt ervoor dat applicaties geautomatiseerd, geconfigureerd en beheerd worden.

Platform as a Service (PaaS): PaaS wordt afgenomen door ontwikkelaars voor de ontwikkeling van een applicatie, zonder de complexiteit van het bouwen en onderhouden van de infrastructuur.

Pod: Een pod is een omhulsel voor één of meerdere containers. In één pod-plaats worden enkel containers die nauw met elkaar samenwerken en dezelfde systeembronnen zullen gebruiken geplaatst.

Proof of Concept (PoC): Realisatie van een bepaalde methode of idee om de haalbaarheid hiervan aan te tonen.

Representational State Transfer (REST): REST is een protocol voor een API en verzorgt de communicatie tussen de webservice en de applicatie.

SMART Methode: Een methode waarmee behoeftes concreet gemaakt worden. Dit wordt gedaan door te kijken of de behoefte Specifiek, Meetbaar, Acceptabel, Realistisch en Tijdsgebonden is.

Triangulatie: Het probleem, of onderdelen van het probleem, wordt vanuit verschillende perspectieven bekeken.

Virtual Machine (VM): Een VM is een computerprogramma dat een computer nabootst, waar andere programma's op kunnen worden uitgevoerd.

Whitepaper: Een whitepaper is een document dat beschrijft hoe een specifiek probleem kan worden opgelost.

Workflow: Een workflow is voorgedefineerd en haalbaar werkpatroon binnen een organisatie. Binnen dit onderzoek is een workflow een voorgedefineerde implementatie van een infrastructuur.

YAML-file: YAML is een door mensen leesbare gegevensformaat. Het wordt vaak gebruikt voor configuratiebestanden.

7. Methode van het onderzoek

Als methode is het ontwerponderzoek, die ontwikkeld is door Saxion Hogeschool [1], als leidraad genomen. Aan de hand van deze methode is op een wetenschappelijk verantwoorde manier onderzocht of er een oplossing is voor het probleem van GRO-Cloud Solutions.

Het ontwerponderzoek gaat ervanuit dat eerst een vooronderzoek wordt gedaan waarin wordt gekeken naar het probleem, de behoefte, de context en de literatuur. Vervolgens worden in de ontwerpdocumentatie het ontwerp en de oplossingen beschreven door middel van een functioneel- en technisch ontwerp. De onderdelen van het ontwerponderzoek maken elk gebruik van hun eigen methodes, welke hieronder staan beschreven.

7.1 Probleemanalyse

Het probleem van ilionx wordt vanuit meerdere perspectieven bekeken in de probleemanalyse (zie voor verdere verdieping hoofdstuk 15.1). Hierdoor wordt het duidelijk wat het onderliggende probleem van de onderzoeksvraag is. Dit wordt gedaan door interviews te houden aan de hand van de 5W1H methode met de business unit manager en lead consultant. Doordat hierbij triangulatie wordt gebruikt, wordt het probleem vanuit meerdere perspectieven belicht. Daarvanuit worden de hoofd- en deelvragen en business requirements opgesteld. Business requirements zijn specificaties van de belanghebbenden [2]. Deze business requirements worden vervolgens aan de hand van de SMART methode opgesteld en daarna geprioriteerd volgens de MoSCoW methode. Door middel van requirements traceability worden deze requirements vervolgens gekoppeld aan de stakeholders. Aan de hand van de afgenomen interviews in de probleemanalyse wordt ook deelvraag zes beantwoord.

7.2 Behoefteanalyse

In de behoefteanalyse (zie voor verdere verdieping hoofdstuk 15.2) wordt inzichtelijk gemaakt wie de stakeholders zijn bij het vraagstuk van ilionx. De verschillende behoeften van de stakeholders worden geïnventariseerd en geprioriteerd door middel van een stakeholderanalyse aan de hand van de verkregen informatie uit de interviews met de business unit manager en lead consultant (in de probleemanalyse beschreven). Hierna worden de user- en system-requirements opgesteld. User-requirements worden omschreven als hetgeen wat de oplossing moet bieden. Systeem-requirements worden omschreven als het gewenste gedrag van het systeem [3]. Hiernaast worden, net als in de probleemanalyse, de SMART- en MoSCoW methode en requirements traceability gebruikt.

7.3 Contextanalyse

In de contextanalyse (zie voor verdere verdieping hoofdstuk 15.3) wordt de specifieke context waarin de oplossingen van het vraagstuk geplaatst moet worden beschreven. Het gaat in de contextanalyse om randvoorwaarden die voortkomen uit de specifieke context. De eerste stap is om de verschillende aggregatieniveaus van de context uit te werken aan de hand van de resultaten van een afgenomen interview met de business unit manager. Dit wordt gedaan aan de hand van de Micro Meso Macro Methode, met daarbij het 7s-model. Hiernaast wordt een veldobservatie gedaan. Door het gebruik van deze methodes, wordt een duidelijk beeld geschetst van de huidige situatie. Vervolgens worden de randvoorwaarden opgesteld. Randvoorwaarden zijn beperkingen en voorwaarden die worden opgelegd door de context aan de mogelijke oplossing.

7.4 Literatuuronderzoek

De literatuur wordt geraadpleegd (zie voor verdere verdieping hoofdstuk 15.4) voor het vinden van antwoorden op de opgestelde deelvragen. Bij dit onderzoek is afgeweken van de hoofdmethode van Saxion, welke uit gaat van ontwerpprincipes. Het doel van dit literatuuronderzoek is namelijk het vergaren van kennis welke later in het onderzoek wordt gebruikt voor het maken van keuzes in de ontwerpdocumentatie (hoofdstuk 8.7), en niet het vinden van ontwerpprincipes. De zesde en zevende deelvraag kunnen niet in het literatuuronderzoek worden beantwoord, het antwoord op deze vragen wordt in de probleemanalyse en ontwerpdocumentatie geformuleerd. Deelvraag zeven kan ook niet worden beantwoord met het literatuuronderzoek, deze wordt beantwoord in de ontwerpdocumentatie. Het literatuuronderzoek wordt gedaan aan de hand van verschillende methodes; de zoekstrategie en sneeuwbalmethode zijn hiervoor gebruikt. Zie voor verdere uitwerking van de deze methoden hoofdstuk 15.5.3.

7.5 Ontwerpdocumentatie

De ontwerpdocumentatie (zie voor verdere verdieping hoofdstuk 16) gaat in op de oplossingen. Dit houdt in dat wordt uitgewerkt hoe de oplossing functioneel en technisch in elkaar zit. Hierbij worden verschillende methoden gebruikt. Als ontwerpbenadering is gekozen voor “prototyping” en als ontwikkelmethode is voor “trail and error” gekozen. In het functionele deel worden de verschillende onderzochte containerplatformen aan de hand van een multicriteria-analyse met elkaar vergeleken. Uit deze vergelijkingen zijn scenario’s te halen waardoor verschillende workflows kunnen worden ontwikkeld, wat wordt gedaan aan de hand van procesmodulering. Vervolgens wordt in het technische deel één van deze workflows uitgewerkt naar een playbook waarmee vervolgens in een Proof Of Concept (PoC) wordt aangetoond of de workflows functioneel zijn en werken. In de ontwerpdocumentatie wordt tevens deelvraag zeven beantwoord.

Vanuit de ontwerpdocumentatie zijn twee beroepsproducten ontwikkeld, namelijk een flyer en een whitepaper. Dit zijn producten die GRO-Cloud Solutions ontwikkeld wilde hebben als oplossing op het vraagstuk. Deze twee beroepsproducten zijn ontwikkeld aan de hand van de opgedane kennis in het vooronderzoek en de verschillende scenario’s in de ontwerpdocumentatie.

8. Resultaten van het onderzoek

In dit hoofdstuk worden de resultaten beschreven. De belangrijkste resultaten van het vooronderzoek en ontwerpdocumentatie zijn hieronder uitgewerkt. Voor verdere verdieping wordt naar hoofdstuk 15 verwezen.

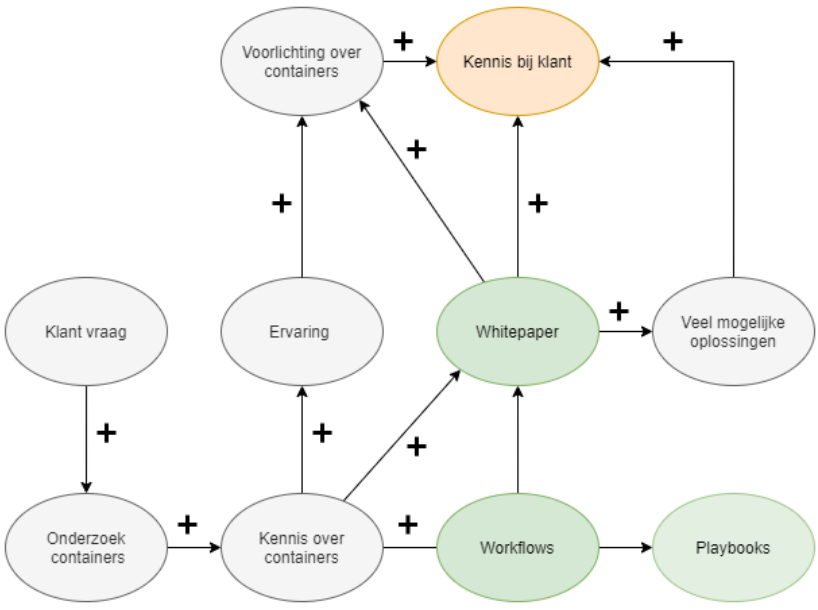
8.1 Probleemanalyse

In de probleemanalyse is de probleemstelling van de opdracht bepaald. Door middel van de probleemstelling zijn de hoofd- en deelvragen opgesteld en business requirements bepaald.

In de probleemanalyse kwam naar voren dat het vraagstuk vanuit ilionx is ontstaan doordat een klant van ilionx vragen had over e-commerce applicaties die ilionx zou gaan hosten. De applicatie draaide namelijk in een container en ilionx had nog niet veel met container gebaseerde applicaties gewerkt. Hier was het bedrijf niet goed op voorbereid. Op dat moment moest onderzocht worden hoe de vraag van de klant het best kon worden beantwoord om zo de klant goed te kunnen voorzien in zijn behoefte. Om voortaan goed voorbereid te zijn op vergelijkbare vragen van klanten, wil ilionx goed onderzocht en vastgelegd hebben wat de beste oplossingen voor container gebaseerde applicaties zijn.

Hiernaast wordt het ook steeds meer een trend om gebruik te maken van container gebaseerde applicaties. Om goed voorbereid te zijn op andere mogelijke vragen van de klanten, en om klanten goed voor te lichten, wordt aan de hand van dit onderzoek een whitepaper gemaakt. Het vraagstuk van ilionx omvat ook dat workflows moeten worden ontwikkeld om geautomatiseerd de verschillende mogelijke infrastructuren uit te rollen. Dit deel van het vraagstuk komt voort uit het feit dat ilionx goed voorbereid wil zijn op mogelijke vragen van klanten.

Om het vraagstuk te verduidelijken is een causaal veldmodel gemaakt. In dit model is te zien wat de oorzaken en gevolgen zijn van het vraagstuk. Zoals te zien is in Figuur 2, wordt eerst onderzoek gedaan naar containers en wordt de vraag van de klant in kaart gebracht. Vervolgens kunnen aan de hand van de opgedane kennis over containers een whitepaper, workflows en een playbook worden ontwikkeld. Het doel van het vraagstuk is om kennis bij de klant te krijgen (oranje), wat kan worden gedaan aan de hand van een whitepaper (groen).



Figuur 2: Causaal veldmodel

8.1.1 Hoofd- en deelvragen

Aan de hand van de probleemanalyse zijn de onderzoeksvraag en deelvragen opgesteld. Elke deelvraag (Tabel 2) beantwoordt een stukje van de hoofdvraag (Tabel 1). Hierdoor kan de hoofdvraag zo volledig mogelijk beantwoord worden.

Nr.	Hoofdvraag
1.	Wat is de meest passende geautomatiseerde infrastructuur oplossing voor mijn container gebaseerde applicatie binnen Microsoft Azure?

Tabel 1: Hoofdvraag

Nr.	Deelvraag
1.	Wat wordt onder container applicatie verstaan?
2.	Welke verschillende mogelijkheden zijn binnen Azure voor het hosten van een container applicatie?
3.	Wat zijn de kosten van deze verschillende oplossingen van Azure?
4.	Voorziet Azure in alle benodigde componenten?
5.	Wat is nodig voor het geautomatiseerd uitrollen van de infrastructuur binnen Azure?
6.	Op welke manier kan de oplossing het best aan de klant gepresenteerd worden?
7.	Hoe demonstreer ik het automatiseringsproces?

Tabel 2: Deelvragen

8.1.2 Business requirements

De eerder beschreven oorzaken van het vraagstuk en de hieruit geformuleerde hoofd- en deelvragen, zijn in Tabel 3 omgezet naar business requirements. De onderstaande requirements zijn opgesteld aan de hand van gehouden interviews en de resultaten van het 5W1H model (voor verdere verdieping, zie hoofdstuk 15.5.1).

Nr.	Prio.	Requirement
B1	Must	De whitepaper moet de unit GRO-Cloud Solutions meer kennis opleveren over containerhosting.
B2	Must	De whitepaper moet ervoor zorgen dat klanten van GRO-Cloud Solutions een leidraad hebben voor bij het kiezen van de juiste hosting voor hun container gebaseerde applicatie.
B3	Must	De workflows moeten de unit GRO-Cloud Solutions helpen bij het ontwikkelen van playbooks voor het automatisch uitrollen van container infrastructuren.

Tabel 3: Business requirements

8.2 Behoefteanalyse

In de behoefteanalyse zijn de stakeholders bepaald aan de hand van een stakeholderanalyse. De stakeholderanalyse geeft inzicht in alle stakeholders bij het vraagstuk. In een stakeholderanalyse wordt gekeken de belangen van de stakeholders en hoe deze belangen gebruikt moeten worden binnen het vraagstuk. De stakeholderanalyse bestaat uit twee stappen, namelijk stakeholder inventarisatie en stakeholder selectie.

8.3 Stakeholder inventarisatie

Om duidelijk te maken wie belang hebben en/of worden beïnvloed bij de uiteindelijk interventie, zijn interviews afgenomen met de business unit manager en lead consultant. Aan de hand van deze interviews is duidelijk geworden wie volgens hen de stakeholders en betrokken bij het vraagstuk zijn.

In het interview met business unit manager is naar voren gekomen dat er vier verschillende stakeholders betrokken zijn bij de uiteindelijke interventie, namelijk:

- De business unit manager;
- De lead consultant;
- De co-collega's van ilionx;
- De klanten.

Het gehele interview met de business unit manager is te vinden in hoofdstuk 15.5.1.1.1.

Hiernaast kwam in het interview met de lead consultant kwam naar voren dat vier verschillende stakeholders betrokken zijn bij de uiteindelijke interventie:

- De business unit manager;
- De lead consultant;
- De sales GRO-Cloud Solutions;
- De klanten (als doelgroep).

Het gehele interview met de lead consultant is te vinden in hoofdstuk 15.5.1.1.2.

In de stakeholderanalyse kwam naar voren dat de in Tabel 4 beschreven stakeholders invloed hebben op dit onderzoek.

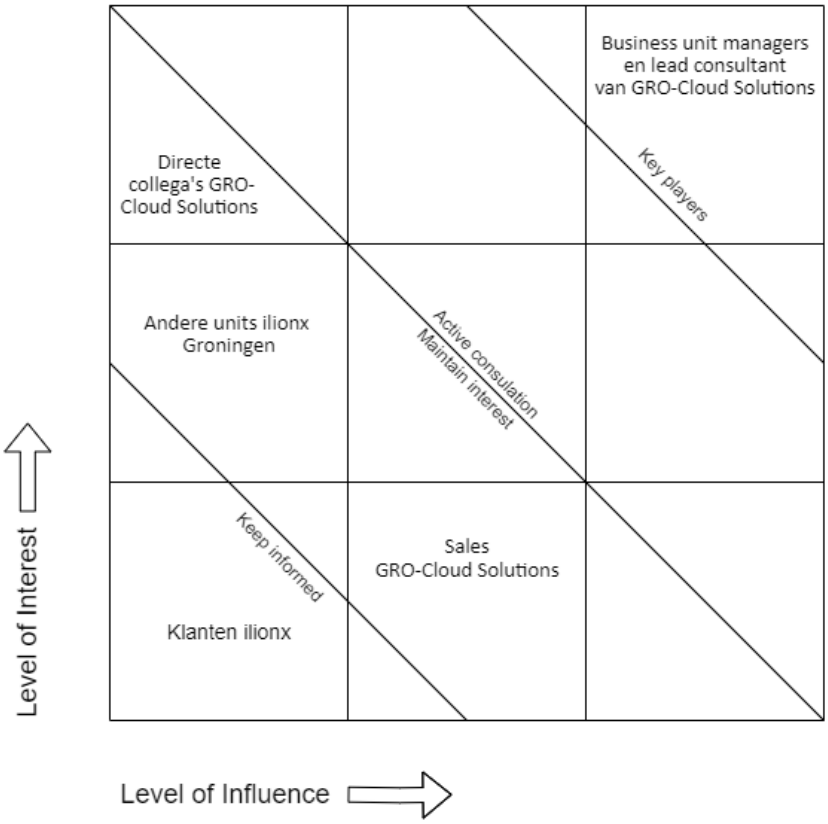
Stakeholder[1]	**Omschrijving**
Business unit manager GRO-Cloud Solutions	De business unit manager stuurt de unit aan. Hij is uiteindelijk eindverantwoordelijk voor de resultaten van de unit.
Lead consultant GRO-Cloud Solutions	De lead consultant heeft contact met de klant over de verschillende diensten die ilionx kan leveren. De lead consultant vertaalt de wens van de klant naar het uiteindelijke product dat de klant gaat gebruiken.
Directe collega's van GRO-Cloud Solutions	De unit waar de opdracht wordt uitgevoerd is GRO-Cloud Solutions. Directe collega's krijgen uiteindelijk te maken met de interventie.
Sales GRO-Cloud Solutions	Sales heeft invloed op de teksten die naar de klanten gaan, maar heeft geen invloed op de inhoud hiervan.
Ander units ilionx Groningen	Ook andere units binnen ilionx gaan gebruik maken van de opgedane kennis uit dit onderzoek.
Klanten van ilionx	De klanten nemen producten en diensten af van ilionx.

Tabel 4: Betrokken stakeholders

[1] De stakeholders en personen zijn geanonimiseerd onder afdelingsnamen. In deze tabel worden ook andere units binnen ilionx aangehaald. Met deze units worden software development en business intelligence bedoeld.

8.4 Stakeholder selectie

Nadat de verschillende stakeholders waren geïnventariseerd, zijn zij ingedeeld in de influence- en interest stakeholdermatrix (zie Figuur 3). Hierdoor is duidelijk geworden wat de invloed en/of interesse van het onderzoek is van de stakeholder. Hoe meer invloed de stakeholder heeft, hoe meer naar rechts de stakeholder in het model wordt geplaatst. Hoe hoger de stakeholder in de matrix wordt geplaatst, hoe interessanter die stakeholder is voor dit onderzoek.



Figuur 3: Influence- en interest stakeholdermatrix

Met de “Key players” die te zien zijn in Figuur 3 (business unit manager en lead consultent van GRO-Cloud Solutions), is een interview gehouden. In dit interview zijn de eisen en wensen van de stakeholders naar voren gekomen. Deze eisen en wensen zijn omgezet naar requirements.

8.4.1 Requirements

De in Tabel 5 beschreven requirements, zijn naar voren gekomen uit de afgenomen interviews met de verschillende stakeholders, namelijk de business unit manager en lead consultant. De requirements zijn geformuleerd en geprioriteerd aan de hand van de SMART- en MoSCoW methode. Bij dit kwalitatieve onderzoek was het lastig om “Meetbaar” van de SMART methode per requirement goed te formuleren, hierdoor zijn een aantal requirements met een voetnoot gespecificeerd.

Nr.	F/NF	Prio.[2]	Requirement
Stakeholder: business unit manager			
U1	NF	Must	De uitkomsten van het onderzoek moeten worden verwerkt in een whitepaper.
U2	F	Must	De whitepaper moet de klant duidelijkheid[3] geven over de hosting van hun container gebaseerde applicatie.
U3	NF	Must	De whitepaper moet worden geschreven voor de doelgroep[4] die niet noodzakelijk een diepgaande IT-kennis heeft.
U4	NF	Must	In de whitepaper moet duidelijk[3] zijn wat de voordelen van container gebaseerde applicaties hosting ten opzichte van traditionele applicaties hosting is.
U5	F	Must	Door middel van het onderzoek moet GRO-Cloud Solutions worden geadviseerd over het gebruik van de hosting van container gebaseerde applicaties.
U6	NF	Could	Bij het onderzoek naar container gebaseerde applicaties moet rekening worden gehouden met DevOps.
U7	NF	Could	Bij het onderzoek naar container gebaseerde applicaties moet rekening worden gehouden met het Continuous Integration proces (CI).
U8	NF	Could	Bij het onderzoek naar container gebaseerde applicaties moet rekening worden gehouden met het Continuous Deployment proces (CD).
Stakeholder: lead consultant			
U9	F	Must	GRO-Cloud Solutions moet aan de hand van dit onderzoek in staat zijn om een container infrastructuur technisch op te bouwen.
U10	NF	Must	GRO-Cloud Solutions moet aan de hand van dit onderzoek meer kennis opdoen van containers.
U11	NF	Must	GRO-Cloud Solutions moet na afloop van dit onderzoek meer duidelijkheid[3] hebben gekregen over hetgeen met containers gedaan kan worden.
U12	F	Must	Ter oplossing van het vraagstuk moet een whitepaper worden gemaakt.
U13	NF	Must	De whitepaper moet impliceren dat GRO-Cloud Solutions kennis heeft opgedaan van container hosting.
U14	NF	Must	De whitepaper moet een leidraad vormen voor de hosting van de container gebaseerde applicatie van de klant.
U15	F	Must	In de whitepaper moeten de verschillende mogelijke oplossingen van container hosting worden beschreven.
U16	F	Must	De in de whitepaper beschreven mogelijkheden moeten worden omgezet naar workflows voor de afdeling GRO-Cloud Solutions.
U17	F	Must	Aan de hand van de workflows moet duidelijk[3] worden hoe GRO-Cloud Solutions automatisch container applicatie infrastructuren technisch kan implementeren.
U18	F	Must	Om aan te tonen dat de werking van de workflows correct is en functioneert, wordt één workflow omgezet naar één playbook.
S1	F	Must	Het playbook dient binnen Azure DevOps uitgevoerd te kunnen worden.
S2	F	Must	Het playbook dient geschreven te worden door middel van Azure Resource Manager.

Tabel 5: Requirements opgedeeld per stakeholder

[2] De requirements met een lagere MoSCoW-prioriteit worden alleen uitgevoerd bij voldoende tijd.
[3] Met "duidelijkheid" wordt het overbrengen van kennis bedoeld. Dit is niet anders kwantificeerbaar volgens de SMART methode.
[4] Hiermee worden alle mogelijke lezers van de whitepaper bedoeld.

8.5 Contextanalyse

In de contextanalyse is de context waarin de uiteindelijke oplossing terecht komt beschreven. Dit is gedaan aan de hand van verschillende aggregatieniveaus, waaruit vervolgens randvoorwaarden zijn opgesteld. De aggregatieniveaus die zijn gebruikt in de contextanalyse zijn de macro, meso en microniveaus. Per aggregatieniveau wordt naar een ander perspectief van de context gekeken. Hieronder zijn de verschillende niveaus samengevat.

Vanuit het macroniveau is gekeken naar de klanten van GRO-Cloud Solutions (voor verdere uitwerking zie hoofdstuk 15.3.2.2.1). Hierin werd duidelijk dat GRO-Cloud Solutions verantwoordelijk is voor de klanten in de noordelijke-provincies van Nederland en zich niet focust op een specifieke branche. Hierdoor is het van belang dat de whitepaper in het Nederlands wordt geschreven. Verder heeft GRO-Cloud Solutions als doel om de zorgen over ICT weg te halen bij de klant.

Vanuit het mesoniveau is gekeken naar GRO-Cloud Solutions (zie voor verdere uitwerking hoofdstuk 15.3.2.2.2). Naast Cloud Solutions in Groningen, bestaat deze unit ook in Utrecht en Maastricht. Deze verschillende units hebben elkaar niet nodig en hanteren elk een verschillende werkwijze. Elke unit is op zijn eigen manier vormgegeven, om zo op verschillende manieren klanten in Nederland te kunnen helpen. De unit GRO-Cloud Solutions werkt standaard met Microsoft Azure. Deze werkwijze komt voort uit dat binnen ilionx voornamelijk met Microsoft producten wordt gewerkt en ilionx hier dus de meeste kennis van heeft. Het bedrijf is niet Microsoft-only, maar werkt wel voornamelijk met deze producten. Omdat GRO-Cloud Solutions in de toekomst gaat werken met de oplossing, moet deze dus toegespitst zijn op zijn werkwijze. De wens van GRO-Cloud Solutions is om eerst te onderzoeken hoe een container gebaseerde oplossing in een CaaS-platform gehost kan worden. Platformen zoals PaaS en FaaS hoeven dus niet meegenomen te worden in het onderzoek.

Het microniveau ging in op de afdeling GRO-Cloud Solutions, wat is gedaan aan de hand van het 7s-model welke beschreven staat in hoofdstuk 15.3.2.2.3. Dit model geeft inzicht in Strategie, Systemen, Gedeelde waarden, Personeel, Structuur, Stijl en Vaardigheden (vertaald naar het Nederlands). Uit deze verschillende onderdelen zijn randvoorwaarden gehaald.

8.5.1 Randvoorwaarden

Uit de aggregatieniveaus die zijn onderzocht, zijn in Tabel 6 de volgende randvoorwaarden opgesteld.

Nr.	Aggregatieniveau	Randvoorwaarde
R1	Macroniveau	De oplossingen moeten in het Nederlands worden geschreven.
R2	Mesoniveau	De oplossingen moeten toegespitst zijn op de werkwijze van GRO-Cloud Solutions.
R3	Mesoniveau	De oplossing moet beperkt worden tot containerplatformen.
R4	Microniveau	Het uitgangspunt voor GRO-Cloud Solutions is dat er gewerkt wordt met Microsoft Azure.
R5	Microniveau	Voor de automation tools moet gebruik worden gemaakt van Azure DevOps met Azure Resource Manager (ARM).

Tabel 6: Randvoorwaarden

8.6 Literatuuronderzoek

In het literatuuronderzoek is de literatuur geraadpleegd voor het vinden van antwoorden op een aantal onderzoeksvragen. Het zoeken in de literatuur is gedaan aan de hand van verschillende gevalideerde methodes (hoofdstuk 15.4.1), om zo een valide en betrouwbaar literatuuronderzoek uit te voeren. Van de opgestelde deelvragen uit Tabel 2, zijn de deelvragen één, twee, drie, vier en vijf beantwoord aan de hand van het literatuuronderzoek.

8.6.1 Deelvraag 1: Containers

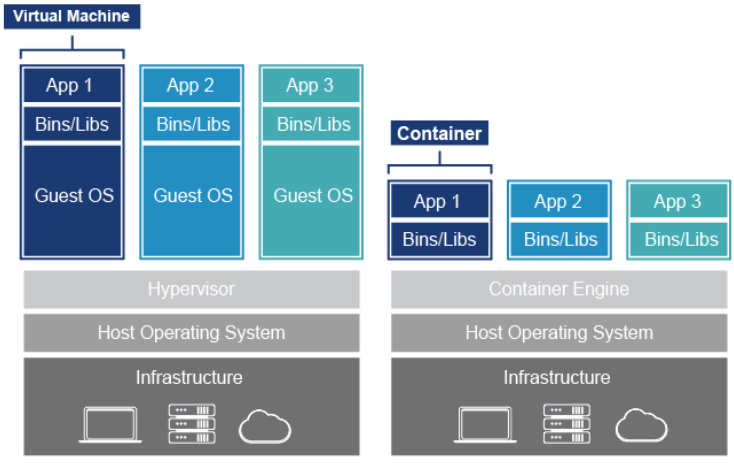
Voor het correct beantwoorden van deze deelvraag moet eerst worden uitgelegd wat een container precies is. Aan de hand hiervan kan duidelijk worden gemaakt wat precies onder een container applicatie wordt verstaan.

Containers

Containers zijn geïsoleerde en portable operating environments waarmee een applicatie (app) kan worden gescheiden van de hardware waarop deze wordt uitgevoerd (hosting). Een container gebaseerde applicatie is een applicatie die wordt ontwikkeld met alle benodigdheden om zelfstandig te kunnen worden uitgevoerd op een container engine. De applicatie is niet afhankelijk van het type host waarop het wordt uitgevoerd.

De isolatie van een container zorgt voor een goede scheiding van werkzaamheden. Ontwikkelaars richten zich op de code van de app, terwijl een Cloud- of Infrastructuurteam zich richt op de implementatie en het beheer van de hosting, zonder zich druk te hoeven maken over app-details [4]. Deze vorm van werken wordt DevOps [5] genoemd. DevOps is een combinatie van ontwikkeling (Dev) en activiteiten zoals implementatie en beheer (Ops). DevOps is een samenstelling van mensen, processen en technologieën om doorlopend van waarde te kunnen zijn voor de klant.

Vaak worden containers vergeleken met een Virtuele Machine (VM). Een VM is een gastbesturingssysteem (Linux/Windows) dat wordt uitgevoerd op een hostbesturingssysteem (Microsoft Hyper-V of VMware ESXi) en krijgt zo gevirtualiseerd toegang tot de onderliggende hardware. Net zoals bij een VM, kan een container een app samen met bibliotheken en andere afhankelijkheden verpakken. Hierdoor wordt het een geïsoleerde omgeving waarin een softwareservice wordt uitgevoerd [4].



Figuur 4: Stack VM en containers

Zoals in Figuur 4 [6] zien is, is het verschil tussen een VM en containers dat de virtualisatie op een andere laag van de stack plaatsvindt. Bij VM's draaien door middel van een hypervisor verschillende VM's op één fysieke server. Bij containers wordt echter het besturingssysteem, in plaats van de hardware, gevisualiseerd. Dit wordt gedaan door middel van een container engine.

8.6.2 Deelvragen 2, 3 en 4: Onderzoek naar containerplatformen

De deelvragen twee, drie en vier zijn samengevoegd in één hoofdstuk omdat deze veel overeenkomsten hebben met elkaar. In dit hoofdstuk is eerst onderzocht welke verschillende mogelijkheden er zijn om een container gebaseerde applicatie te hosten binnen Microsoft Azure. De mogelijke container hosting oplossingen zijn daarna onderzocht op verschillende componenten (hoofdstuk 15.4.2.2.2).

8.6.2.1 Container oplossingen binnen Microsoft Azure

Microsoft Azure voorziet in meerdere oplossingen voor het hosten van container gebaseerde applicaties. Microsoft Azure heeft drie oplossingen beschikbaar, namelijk Azure Kubernetes Service (AKS), Azure Red Hat OpenShift (ARO) en Virtual Machine (VM) met Docker.

Azure Kubernetes Service
AKS is een managed container orchestration-service op het CaaS-platform. Deze service is gebaseerd op de open source techniek Kubernetes. Een organisatie kan gebruik maken van AKS om container gebaseerde applicaties in een cluster van container hosts te implementeren, schalen en beheren [7].

Azure Red Hat OpenShift
ARO biedt een hybride Cloud omgeving op het CaaS-platform, welke bestaat uit het bouwen en schalen van container gebaseerde applicaties. Het platform is opgebouwd rond een kern van Docker containers en wordt beheerd door Kubernetes clusters met onderliggend de Linux gebaseerde infrastructuur [8].

Virtual Machine met Docker
Deze optie maakt gebruik van een VM op het IaaS-platform van Microsoft Azure [9]. De oplossing maakt gebruik van een VM met Ubuntu Linux waarop Docker is geïnstalleerd [10]. Docker wordt gebruikt om de verschillende applicaties binnen een container uit te voeren. Containers zijn binnen Docker geïsoleerd van elkaar en bundelen hun eigen applicatie, tools, bibliotheken en configuratiebestanden.

8.6.2.2 Overige container oplossingen binnen Microsoft Azure

Naast de bovenstaande oplossing heeft Microsoft Azure nog twee container platformen die niet in dit onderzoek zijn meegenomen, namelijk Azure Container Instances (ACI) en Azure App Services (AAS). Deze twee oplossingen worden aangeboden als een PaaS-platform, welke niet verwerkt hoeven te worden in dit onderzoek (zie 8.5.1, Tabel 5, randvoorwaarde drie).

Azure Container Instances
ACI is een PaaS-omgeving van Microsoft Azure waar containers kunnen worden geïmplementeerd zonder dat een onderliggende infrastructuur gemaakt of beheerd hoeft te worden. ACI die zowel Linux als Windows containers ondersteunt, elimineert de noodzaak voor een ontwikkelaar om VM's in te richten of een container orkestratie platform, zoals Kubernetes, te configureren om daar uiteindelijk containers te implementeren en uit te voeren. In plaats daarvan kan een organisatie met ACI een nieuwe container opstarten, via de Azure Portal of de Command Line Interface (CLI), en Microsoft levert en schaalt automatisch de onderliggende rekenresources.

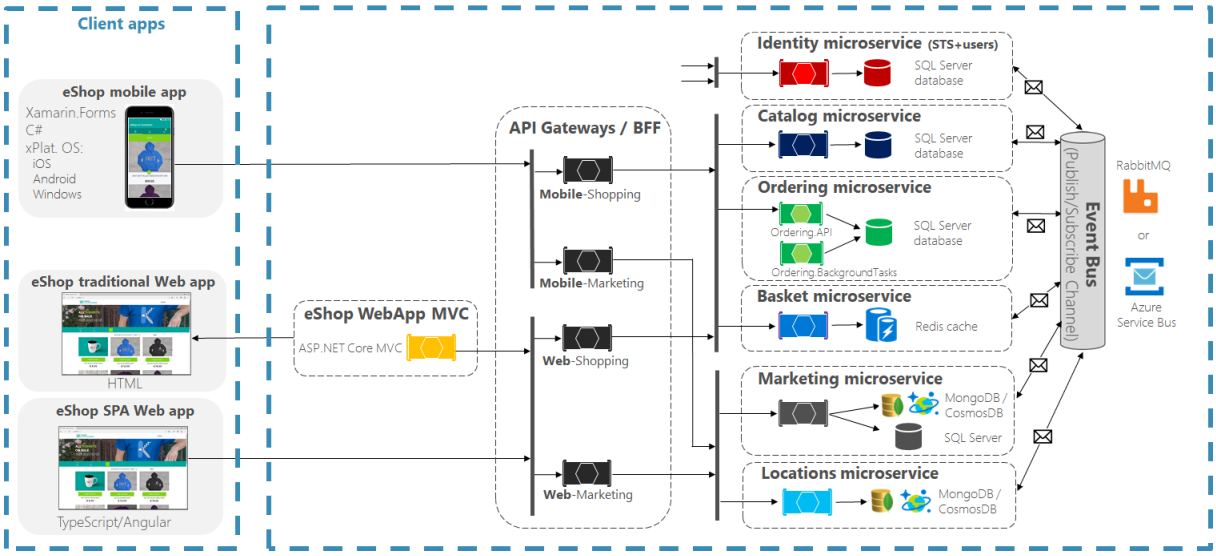
Azure App Services
Microsoft AAS is een volledig beheerde PaaS-omgeving welke is ontstaan door het integreren van Microsoft Azure-websites, mobiele services en BizTalk-services. AAS biedt de gebruiker verschillende mogelijkheden tot het hosten van Web Apps, Mobile Apps, API Apps en Logic Apps. Met AAS is het dus mogelijk verschillende soorten apps te hosten binnen het platform, zonder dat rekening moet worden gehouden met beheer van een ICT-infrastructuur.

8.6.2.3 Benodigdheden container gebaseerde applicatie

Hieronder staat in Figuur 5 een voorbeeld van een container gebaseerde applicatie. De applicatie is een e-commerceapplicatie en is ontwikkeld door Microsoft [11]. Deze applicatie wordt aangehaald om te laten zien dat een container gebaseerde applicatie meer nodig heeft dan alleen container hosting.

In dit hoofdstuk zijn de generieke benodigdheden voor het hosten van een container gebaseerde applicatie onderzocht. De voorbeeld container gebaseerde applicatie maakt gebruik van:

1. Application Programming Interface (API) zorgt ervoor dat de juiste acties worden uitgevoerd in de microservices.
2. De Event Bus wordt gebruikt om te zorgen dat de verschillende microservices informatie met elkaar kunnen uitdelen.
3. De applicatie maakt verder nog gebruik van verschillende databases, welke ervoor zorgen dat de gegevens van de container gebaseerde applicatie worden opgeslagen.



Figuur 5: Voorbeeld e-commerceapplicatie

De voorbeeldapplicatie is gebruikt om onderzoek te doen naar de verschillende componenten die nodig zijn voor een container gebaseerde applicatie. Deze componenten zijn gespecificeerd in hoofdstuk 15.4.2.2.3. Met het gebruiken van dit voorbeeld kan een vergelijking gemaakt worden over wat de impact op prijs, beschikbaarheid en betrouwbaarheid tussen de verschillende oplossingen is.

8.6.2.4 Meerwaarde van Microsoft Azure

Door het Cloudplatform Microsoft Azure te gebruiken voor het hosten van de container gebaseerde applicatie, biedt Microsoft Azure verschillende mogelijkheden om de productiviteit van de afnemer te verhogen.

Doordat Microsoft Azure voorziet in alle benodigdheden voor het hosten van container gebaseerde applicatie, kan elke vorm van applicatie binnen Microsoft Azure gehost worden. Daarnaast is het binnen Microsoft Azure mogelijk om een container gebaseerde applicatie tijdelijk bij te schalen op een PaaS-platform, wanneer het gebruikte containerplatform niet voldoende resources heeft [12].

Verder zorgt Microsoft Azure voor de hosting, beheer en updates van de containerplatformen, waardoor de afnemer hierover ontzorgd wordt. Bij het containerplatform Redhat OpenShift wordt de afnemer ook nog ontzien van het configureren en beheren van een back-up, wat door een goede samenwerking van Microsoft Azure en Redhat tot stand is gekomen (dit wordt uitgevoerd door Microsoft Azure) [13].

Daarnaast speelt Microsoft Azure een toonaangevende rol in de bescherming en privacy van de verschillende infrastructuur benodigdheden. Doordat de Microsoft Azure Cloud de capaciteit heeft, om veel geld te investeren in de veiligheid en privacy van de data [14]. Met het gebruik van Microsoft Azure voor de hosting van de container gebaseerde applicatie, worden ook een deel van deze zorgen bij de afnemer weggenomen.

8.6.2.5 Uitwerking componenten per mogelijke container oplossing

Hieronder staan de mogelijke containeroplossingen van Microsoft Azure uitgewerkt.

8.6.2.5.1 Azure Kubernetes Service

Hieronder wordt AKS door middel van de opgestelde componenten onderzocht en uitgelegd aan de hand van de testapplicatie.

Beschikbaarheid

Microsoft Azure streeft ernaar om een beschikbaarheid van minimaal 99.95% te halen [15]. De agent node van AKS wordt onder de motorkap gehost in een VM en valt hierdoor onder het Service Level Agreement (SLA) van de VM's. De SLA garandeert een uptime van 99.95% [16].

Geconcludeerd kan worden dat de uptime gelijk is aan het laagste uptime percentage. Dit zou dus betekenen dat AKS een beschikbaarheid heeft van 99.95%. Hieruit kan worden opgemaakt dat de e-commerce voorbeeldapplicatie mogelijk per jaar 4 uur en 22 minuten seconden niet beschikbaar is voor de eindgebruiker [17].

Betrouwbaarheid

Voor de consistentie en continuïteit van container gebaseerde applicaties kan standaard worden gewerkt met VM Scale Set. Hiermee wordt zowel de beschikbaarheid als de betrouwbaarheid verhoogd. Alleen bij een Azure regio storing is het AKS-cluster dan niet beschermd tegen uitval.

Om bij mogelijke fouten in het AKS-cluster snel weer online te zijn, kan worden gewerkt met een gespiegeld cluster over meerdere regio's. Hiermee kan ervoor worden gezorgd dat bij het wegvallen van regio X, regio Y het overneemt. Hierdoor merkt een bezoeker van de webshop niets tot nauwelijks iet van het weg vallen van het AKS-cluster in regio X [18]. Een gespiegeld AKS-cluster over twee of meerdere regio's is in feite een back-up van het AKS-cluster en verhoogt de kans op een hogere continuïteit van de applicatie.

Schaalbaarheid

AKS is goed schaalbaar. Het is namelijk mogelijk om een AKS-cluster te schalen over meerdere Availability Sets (AS), Availability Zones (AZ) en Regio Pairs (RP). Het schalen van een AKS-cluster wordt nadat het is ingeschakeld, automatisch uitgevoerd door Azure. Dit wordt gedaan doordat het AKS-cluster dan niet op één VM wordt gehost, maar met een cluster dat gebruik maakt van VM Scale Sets voor clusterknooppunten [19]. Voor een oplossing in meerdere regio's moet gebruik worden gemaakt van Azure Traffic Manger om zo het verkeer over meerdere AKS-clusters te verdelen [18].

De container gebaseerde applicatie kan worden geschaald door Kubernetes Horizontal Pod Autoscaler. Deze schaalt automatisch het aantal pods (containers) op basis van waargenomen CPU-gebruik [20]. Daarnaast kan nog gebruik gemaakt worden van de virtual nodes instelling binnen AKS. Deze schaalt containers bij door het gebruik van ACI.

Veiligheid

De veiligheid van de container gebaseerde applicatie en van het AKS-cluster is een belangrijk onderdeel. Kubernetes bevat zelf onderdelen als netwerkbeleid en secrets. Netwerkbeleid van Kubernetes gaat in op de toegangsbeperking van API's en het Kubernetes cluster [21]. Met Kubernetes secrets kan gevoelige informatie opgeslagen en beheerd worden (zoals wachtwoorden, OAuth-tokens en SSH-sleutels) [22].

Microsoft Azure heeft naast de standaard beveiligingsfeatures van Kubernetes extra beveiliging toegevoegd op punten zoals netwerkbeveiligingsgroepen en georkestreerde clusterupgrades. Deze beveiligingscomponenten worden gecombineerd om ervoor te zorgen dat het AKS-cluster de nieuwste beveiligingsupdates en Kubernetes-releases uitvoert. Verder zit in AKS een integratie met Azure Active Directory, waarmee gemakkelijk toegang verleend kan worden over delen van het AKS-cluster aan geautoriseerde medewerkers binnen de organisatie [23].

Kosten en prijsmodel
Voor de hosting wordt het volgende knooppunt gebruikt: "D8 V3: 8 vCPU('s), 32GB RAM, 200GB tijdelijk opslag". De voorbeeldapplicatie komt te draaien op vier VM's, en het OS komt te draaien op een standaard SSD met 256GB (E15- 256GB). Dit alles wordt afgenomen in "Regio: EU-West". De kosten van de omgeving zijn te zien in Tabel 7. Microsoft biedt ook de mogelijkheid om de omgeving te reserveren voor één of drie jaar. De prijzen zijn berekend aan de hand van de prijzen calculator van Microsoft Azure [24].

Deployment	**Aantal**	**Azure AKS Maandelijkse betaling**	**Azure AKS 1 jaar gereserveerd (~32% korting)**	**Azure AKS 3 jaar gereserveerd (~57% korting)**
D8 V3: 8 vCPU('s), 32GB RAM, 200GB tijdelijke opslag	4	€ 1.181,97 p.m.	€752,79 p.m.	€518,54 p.m.
E15- 256 GB	4	€64,77 p.m.	Niet beschikbaar[5]	Niet beschikbaar[5]
Totaalbedrag		€1.246,73 p.m.	€817,56 p.m.	€583,31 p.m.

Tabel 7: Kosten AKS

Features/mogelijkheden
De belangrijke onderdelen van Kubernetes zijn [25]:
- Geautomatiseerde planning;
- Zelfherstellende mogelijkheden;
- Geautomatiseerd uitrollen en terugdraaien;
- Horizontaal schalen en load balancing;
- Automatisch schaalbare infrastructuur.

Voor- en nadelen
Voordelen van AKS:
- Schaalbaarheid en molariteit;
- Rijk ecosysteem van add-ons;
- Monitoring beschikbaar vanuit Microsoft Azure monitoring op Kubernetes-cluster en containerniveaus;
- Brede ondersteuning van de Kubernetes community.

Nadelen van AKS:
- Kubernetes kan code niet rechtstreeks uitvoeren, alleen via een container.
- Als open source dienst biedt Microsoft AKS geen serviceovereenkomst met financiële compensatie. Microsoft streeft naar een beschikbaarheid van minimaal 99,95% voor de Kubernetes API-server. De beschikbaarheid van de agentknooppunten in het cluster wordt gedekt door VM SLA.
- Problemen oplossen kan ingewikkeld zijn omdat Kubernetes grotendeels via de CLI wordt aangestuurd.

[5] Alleen maandelijks te factureren

8.6.2.5.2 Azure Red Hat OpenShift

Hieronder wordt ARO door middel van de opgestelde componenten onderzocht en uitgelegd aan de hand van de testapplicatie.

Beschikbaarheid
Microsoft Azure garandeert volgens de SLA van ARO een uptime van ten minsten 99.9 % [26]. Dit betekent dat de container gebaseerde applicatie mogelijk per jaar 8 uur, 46 minuten niet beschikbaar kan zijn voor de eindgebruiker [27].

Betrouwbaarheid
ARO maakt gebruik van Kubernetes. Hierdoor wordt gebruik gemaakt van etcd-gegevens, welke per uur worden geback-upt en vervolgens standaard zes uur bewaard. Mochten deze gegevens hersteld moeten worden, zal er sprake zijn van downtime.

Naast de back-up worden upgrades uitgevoerd om Common Vulnerabilities and Exposures (CvE's) te voorkomen. Daarnaast voert Azure onderhoud uit aan netwerk en netwerkopslag. Mede door de bovenstaande maatregelen wordt de consistentie en continuïteit van ARO gegarandeerd [28].

Schaalbaarheid
De ARO is goed schaalbaar, namelijk op hostniveau en op containerniveau (pod). Op het hostniveau kan een cluster worden gecreëerd van meerdere hosts, wat kan worden gedaan door middel van Azure CLI. Door het gebruik van een ARO worden de verschillende container gebaseerde applicaties verspreid over het cluster, om zo het gebruik te verdelen over meerdere servers. Verder kan een ARO-cluster gespiegeld worden over meerdere Azure regio’s. Hierdoor is bij een regio storing de container gebaseerde applicatie nog steeds bereikbaar [29].

Daarnaast kan een container gebaseerde applicatie worden bijgeschaald door Kubernetes Horizontal Pod Autoscaler te gebruiken. Deze schaalt automatisch het aantal pods op basis van het waargenomen CPU- en geheugengebruik [30].

Veiligheid
ARO wordt beveiligd door Red Hat. OpenShift bevat naast de standaard Kubernetes beveiliging (netwerkbeleid en secrets) nog een hele lijst met extra veiligheidsmaatregelen. Red Hat OpenShift is draait namelijk op Red Hat Enterprise Linux. Deze vorm van Linux heeft extra beveiligingsmogelijkheden. Deze speciale versie van Linux uitgevoerd met:

- SELinux;
- Proces-, netwerk- en opslagscheiding;
- Statefull en stateless inspectie firewall;
- Proactieve bewaking van capaciteitslimieten (CPU, schijf, geheugen, etc.);
- Inbraakdetectie (bestanden, poorten, achterdeuren, etc.);
- Beveiliging frameworks;
- RPM-verificatie en is bijgewerkt op kwetsbaarheden;
- Versleutelde communicatie (SSH, SSL, etc.).

Verder wordt de risicobeoordeling en veiligheidsadvies verzorgd door het productbeveiligingsteam van Red Hat [31].

Kosten en prijsmodel
Voor de hosting moeten standaard al verschillende onderdelen worden aangeschaft. Zo bestaat een standaard installatie van ARO uit [32]:

1. Vier applicatie nodes;
2. Drie master nodes;
3. Drie infrastructuur nodes.

Voor de applicatie nodes is gekozen voor instance type “D8 V3: 8 vCPU(‘s), 32GB RAM” met een opslag disk met het type: “P10: 128GiB”. Daarnaast moet naast het bedrag voor de applicatie nodes, ook nog voor de licentie van AHO worden betaald. Voor de master en infrastructuur nodes wordt het instance type “D4s V3: 4 vCPU(‘s), 16GB RAM” met een opslag disk met het type “P3: 16GiB” gebruikt. Dit alles wordt afgenomen in “Regio: EU-West”. De kosten van de omgeving zijn te zien in Tabel 8. Microsoft biedt ook de mogelijkheid om de omgeving te reserveren voor één of drie jaar. De prijzen zijn berekend aan de hand van de prijzen calculator van Microsoft Azure [24].

Deployment	**Aantal**	**Azure AKS Maandelijkse betaling**	**Azure AKS 1 jaar gereserveerd (~32% korting)**	**Azure AKS 3 jaar gereserveerd (~57% korting)**
D8 V3: 8 vCPU(‘s), 32GB”	4	€1.053,92 p.m.	€690,66 p.m.	€455,58 p.m.
Red Hat OpenShift licentie	1	€3.747,83 p.m.	€2.248,80 p.m.	Niet beschikbaar[6]
D4s V3: 4 vCPU(‘s), 16GB RAM	6	€790,44 p.m.	€518,22 p.m.	€341,66 p.m.
P10: 128GiB	4	€66,49 p.m.	Niet beschikbaar[5]	Niet beschikbaar[5]
P3: 16GiB	6	€6,68 p.m.	Niet beschikbaar[5]	Niet beschikbaar[5]
Totaalbedrag		€5.665,36 p.m.	€3.530,84 p.m.	€3.119,20 p.m.

Tabel 8: Kosten ARO

Features/mogelijkheden
De belangrijkste features in ARO zijn [33]:

- Kubernetes-based Event-driven Auto-scaling (KEDA);
- SELinux;
- ODO Command Line Interface;
- OpenShift Service Mesh.

Voor- en nadelen
Voordelen van ARO:

- Veiligheid;
- Schaalbaarheid;
- Ondersteuning Red Hat en Microsoft.

Nadelen van ARO:

- Hoge kosten;
- Met de Gui wordt ten opzichte van de CLI weinig gedaan.

[6] Prijs van één jaar gereserveerd

8.6.2.5.3 Virtual Machine met Docker

Hieronder wordt VM met Docker door middel van de opgestelde componenten onderzocht en uitgelegd aan de hand van de testapplicatie.

Beschikbaarheid

Het SLA van Microsoft Azure geeft drie verschillende uptime garanties [16]:

1. Voor VM's waarbij twee of meer instances zijn geïmplementeerd in twee of meer availability zones binnen dezelfde Azure-regio, garandeert Microsoft Azure tenminste 99,99% van de tijd connectiviteit.
2. Voor alle VM's waarbij twee of meer instances zijn geïmplementeerd binnen dezelfde availability sets of in dezelfde toegewezen hostgroep, garandeert Microsoft Azure ten minste 99,95% connectiviteit.
3. Voor Single-Instance Virtuele Machines die gebruikmaken van Premium SSD of Ultra Disk voor alle Operating System-schijven en Gegevensschijven, garandeert Microsoft Azure ten minste 99,9% connectiviteit.

Voor de container gebaseerde applicatie wordt gekozen voor meerdere VM's over de verschillende availability zones binnen een regio. Hierdoor komt de maximale downtime uit op 52 minuten en 35 seconden per jaar [34].

Betrouwbaarheid

De VM waarop Docker, en dus de container gebaseerde applicatie, draait kan worden geback-upt door middel van Azure Backup-service. Met deze backup-service kan aangeven worden hoe vaak de VM moet worden geback-upt en wat moet worden geback-upt. Met Azure Backup-services kunnen naast back-ups ook snapshots worden genomen van de VM status [35]. De combinatie van back-ups en snapshots zorgen ervoor dat op het moment van een crash van de container gebaseerde applicatie, snel kan worden hersteld naar een moment dat de applicatie nog correct wekte.

Azure Backup-service wordt bij het component kosten en prijsmodel meegenomen om zo een betrouwbaar beeld te vormen van de kosten voor een operationele container gebaseerde applicatie [36].

Schaalbaarheid

Om een schaalbare container gebaseerde applicatie binnen Docker te krijgen, zijn extra generieke componenten nodig van Microsoft Azure. Deze extra generieke componenten zijn VM Scale Sets [37] en een load balancer [38] van Microsoft Azure.

VM Scale Sets zorgen ervoor dat bij een bepaalde belasting van de VM's automatisch een nieuwe VM wordt opgestart. Deze VM is een exacte kopie van de al bestaande VM. Op de VM's draait een Docker Swarm [39]. Dit is een cluster van Docker hosts die samenwerken om verschillende containers te hosten. Een Docker Swarm verdeelt zijn gebruik over de verschillende hosts. De nieuw toegevoegde VM vanuit VM Scale Sets wordt toegevoegd aan de Docker Swarm en hierdoor wordt het gebruik van de al eerder bestaande Docker hosts verminderd.

Docker is niet in staat automatisch de applicatie container te schalen. Dit moet handmatig gebeuren via de CLI. Docker is niet in staat dit te doen omdat het geen orchestrator is zoals Kubernetes. Het gebruik van een derde partij als orchestrator (zoals VMware orchestrator of Kubernetes) zal het wel mogelijk maken om de applicatie containers bij te schalen [40, 41]. Bij gebruik van meerdere dezelfde applicatiecontainers is het gebruik van een applicatie loadbalancer nodig om het gebruik van de applicatie te verdelen over de verschillende containers.

Veiligheid
De veiligheid van de container gebaseerde applicatie die binnen de Docker wordt gehost, is afhankelijk van de container hosting (Docker) en de hosting van de VM (IaaS-platform van Microsoft), hierdoor zijn de volgende mogelijkheden voor het beveiligen van de VM's [42]:

- Azure Firewall: Beveiligt het Azure Virtueel Netwerk.
- Microsoft Antimalware voor Azure: Beschermt VM's tegen schadelijke bestanden, ad-ware en andere bedreigingen.
- Azure Schijfversleuteling voor IaaS-VM's: Biedt Azure Disk Encryptie aan, wat gebruik maakt van de industriestandaard BitLocker voor Windows en de dm-crypt-functie voor Linux.
- Azure Backup (zie betrouwbaarheid).
- Virtueel Azure Netwerk: Zorgt ervoor dat het netwerk niet toegankelijk is voor andere Azure klanten doordat het virtuele netwerk geïsoleerd is van alle andere virtuele Azure netwerken.
- Azure Security Center: Geeft inzicht en controle over de Azure Resources.

Naast deze bovenstaande mogelijkheden binnen Azure, bestaat de mogelijkheid om de VM alleen toegankelijk te maken voor geautoriseerde gebruikers en/of via een beveiligde verbinding (VPN) [43].

Kosten en prijsmodel
De kosten van een VM met Docker worden berekend voor het gebruik van de voorbeeld container gebaseerde applicatie. Hierbij wordt uitgegaan dat standaard vijf VM's draaien in één Docker Swarm en dat deze kan worden bijgeschaald met nog eens drie Docker VM's. Verder worden de containers dubbel uitgevoerd binnen de Docker Swarm en worden deze containers geloadbalanced door middel van een Azure load balancer. De beveiliging van de VM's wordt geregeld door middel van een Azure Firewall.

De hosting wordt gedaan in "Regio: Europa-West". De VM's bestaan uit het type instance "D8 V3: 8 vCPU('s), 32GB RAM" met een opslag disk met het type: "E15: 256GiB". Voor de tijdelijke VM's die worden uitgevoerd door middel van VM Scale sets wordt gebruik gemaakt van het type instance "D8 V3: 8 vCPU('s), 32GB, 64 GB tijdelijke opslag". De load balancer wordt berekend als een standaard load balancer met 10 load balancer-regels en verwerkt de gegevens van 1000 GiB. Deze 1000 GiB geldt ook voor de Azure Firewall. De kosten van de omgeving zijn te zien in Tabel 9. Microsoft biedt ook de mogelijkheid om de omgeving te reserveren voor één of drie jaar. De prijzen zijn berekend aan de hand van de prijzen calculator van Microsoft Azure [24].

Diensten	Aantal	VM met Docker Maandelijkse betaling	VM met Docker 1 jaar gereserveerd (~32% korting)	VM met Docker 3 jaar gereserveerd (~57% korting)
VM: D8 V3: 8 vCPU('s), 32GB	5	€1.752,- p.m.	€1.115,84 p.m.	€768,62 p.m.
Disk: E15: 256GiB	5	€96,- p.m.	Niet beschikbaar[5]	Niet beschikbaar[5]
Scale Sets D8 V3: 8 vCPU('s), 32GB, 64 GB tijdelijke opslag	3	€499,56 p.m.[7]	€670,01 p.m.[8]	€460,99 p.m.[8]
Load balancer	1	€59,75 p.m.	Niet beschikbaar[5]	Niet beschikbaar[5]
Azure Firewall	1	€928,50 p.m.	Niet beschikbaar[5]	Niet beschikbaar[5]
Totaalbedrag		€3.336,01 p.m.	€2.870,30 p.m.	€2.314,06 p.m.

Tabel 9: Kosten VM met Docker

Features/mogelijkheden
De belangrijkste mogelijkheden van VM met Docker zijn [44]:
- Geïsoleerde omgevingen voor het beheren van de applicaties;
- Docker Swarm;
- Routing Mesh;
- Veiligheidsmanagement;
- Wendbaarheid van applicaties;
- Productiviteit voor ontwikkelaars;
- Operationele efficiëntie.

Voor- en nadelen
Voordelen van Docker in een VM:
- Vrijheid van beheer;
- Snelle applicatie-implementatie;
- Docker is goed gedocumenteerd;
- Docker heeft openbare containerregisters.

Nadelen van Docker in een VM:
- Schaalbaarheid Docker;
- Docker heeft een beperkte monitoring.

[7] Prijs is berekend op het gebruik van 362 uur
[8] Prijs is berekend op het gebruik van 720 uur

8.6.3 Deelvraag 5: Benodigdheden geautomatiseerde uitrol infrastructuur

Het automatisch uitrollen van een infrastructuur kan door middel van een playbook. Een playbook is infrastructure by code; door een playbook uit te voeren wordt de infrastructuur die gescript is in het playbook uitgerold binnen een Cloud provider.

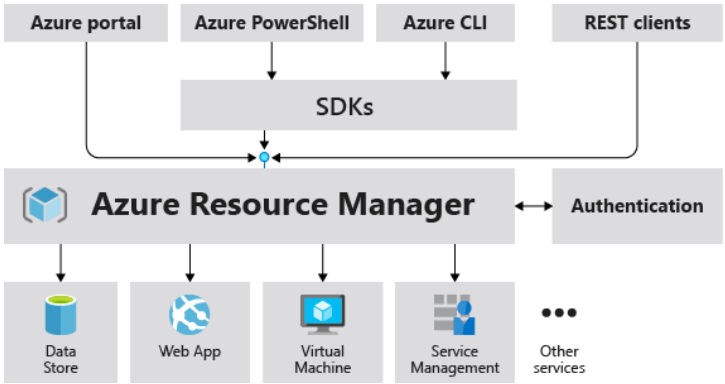
Een playbook kan met verschillende automation tools worden gemaakt en uitgerold. Het maken en uitrollen van een playbook binnen Microsoft Azure kan via de volgende automation tools: Ansible, Terraform, Azure Resource Manager (ARM) en Azure DevOps.

Voor het playbook die voor GRO-Cloud Solutions gemaakt moet worden, wordt gebruik gemaakt van ARM. Het playbook is ontwikkeld met ARM en wordt uitgevoerd door middel van een Azure DevOps CD-pipeline, welke is geïmplementeerd binnen Microsoft Azure. De keuze hiervoor komt voort uit de requirements S1 en S2 (zie Tabel 5) en uit de opgestelde randvoorwaarde nummer vier uit de contextanalyse (zie Tabel 6).

8.6.3.1 Azure Resource Manger

ARM is een deployment- en beheersservice voor Microsoft Azure. ARM biedt een beheerlaag waarmee resources binnen Azure kunnen worden aangemaakt, geüpdatet en verwijderd. ARM gebruikt beheerfuncties zoals toegangscontrole, vergrendelingen en tags, om resources na implementatie te beveiligen en te ordenen.

Als een geautoriseerde gebruiker een verzoek verzendt vanuit één van de Azure-hulpprogramma's, ontvangt ARM deze aanvraag. ARM verifieert en machtigt het verzoek en stuurt de aanvraag naar de Azure-service, waarbij de gevraagde actie wordt uitgevoerd (zie Figuur 6) [45].



Figuur 6: ARM

8.6.3.1.1 ARM templates

ARM wordt aangesproken door middel van ARM templates. In deze templates wordt gedefinieerd wat moet worden uitgerold binnen Microsoft Azure. Zoals te zien is in Figuur 7, worden templates geschreven in JSON (JavaScript Object Notation). Dit is een gestandaardiseerd gegevensformaat. De ARM template maakt gebruik van declarative syntax, waarmee wordt aangegeven welke resources met welke eigenschappen geïmplementeerd moeten worden, zonder dat dit op logische manier gescript is [46].

```JSON
"resources": [
  {
    "type": "Microsoft.Storage/storageAccounts",
    "apiVersion": "2016-01-01",
    "name": "mystorageaccount",
    "location": "westus",
    "sku": {
      "name": "Standard_LRS"
    },
    "kind": "Storage",
    "properties": {}
  }
]
```

Figuur 7: JSON script

8.6.3.2 Azure DevOps

Azure DevOps is door Microsoft ontwikkeld om functionaliteit te bieden voor het testen en deployen van applicaties en infrastructuren. Azure DevOps bestaat uit de volgende onderdelen: Azure Boards, Azure Pipelines, Azure Repos, Azure Test Plans, Azure Artifacts en Extensions Marketplace. Deze onderdelen kunnen als één gezamenlijke of als afzonderlijke oplossing worden afgenomen om bestaande workflows aan te vullen. Azure DevOps wordt voornamelijk gebruikt als CI-/CD-proces, echter is voor het automatisch uitrollen van de ARM-templates alleen het CD-proces nodig. Het CI-proces wordt gebruikt voor het testen en builden van de applicatie [47].

De Azure DevOps CD-Pipeline kan worden gebruikt om de infrastructuur en de applicatie uit te rollen binnen Microsoft Azure. Voor de infrastructuur zou er dan gebruik worden gemaakt van ARM-templates die worden opgehaald uit een repository (bijvoorbeeld Azure Repos). Daarna kunnen deze templates de infrastructuur uitrollen binnen Microsoft Azure [48]. Het CD-proces voor de applicatie wordt automatisch getriggerd doordat het CI-proces een artifact heeft aangemaakt. Dit artifact bestaat uit bestanden en gegevens die nodig zijn voor het uitrollen van de container gebaseerde applicatie op de containerhosting [47].

8.7 Ontwerpdocumentatie

De ontwerpdocumentatie gaat in op hoe de uiteindelijke oplossingen tot stand komen. De uiteindelijke oplossingen moeten voldoen aan de opgestelde requirements en randvoorwaarden uit het vooronderzoek. De business requirements staan Tabel 3 beschreven, de user- en system requirements staan in Tabel 5 en de randvoorwaarden staan in Tabel 6 beschreven.
Tevens wordt in de ontwerpdocumentatie deelvraag zeven beantwoord.

De ontwerpdocumentatie bestaat uit twee delen, namelijk een functioneel- en een technisch ontwerp. In het functioneel ontwerp worden de onderzochte container oplossingen uit het literatuuronderzoek door middel van een multicriteria-analyse met elkaar vergeleken. Hierdoor is in één overzicht te zien wat de sterke en zwakke punten van de verschillende containerplatformen zijn. Door middel van die informatie worden implementatiescenario's bedacht. Deze implementatiescenario's worden in de whitepaper gebruikt om een keuzeboom voor de klant te ontwikkelen.

De implementatiescenario's worden in het functioneel ontwerp ook verder uitgewerkt naar workflows. Om aan te tonen dat de workflows werken, wordt een workflow in het technisch ontwerp omgezet naar een playbook, zodat in het Proof of Concept (PoC) kan worden aangetoond dat het een werkende oplossing is.

8.7.1 Functioneel ontwerp

In het functioneel ontwerp worden de onderzochte container oplossingen uit het literatuuronderzoek door middel van een multicriteria-analyse met elkaar vergeleken. Aan de hand van de informatie uit de multicriteria-analyse kunnen implementatiescenario's worden bedacht en deze kunnen vervolgens uitgewerkt worden naar workflows.

8.7.1.1 Multicriteria-analyse

De containeroplossingen die in het literatuuronderzoek zijn onderzocht worden nu met elkaar vergeleken door middel van een multicriteria-analyse, welke te zien is in Tabel 13. In de multicriteria-analyse is te zien welk containerplatform het best scoort op verschillende componenten. Bij deze analyse worden de gegevens uit het literatuuronderzoek gebruikt en wordt per component de onderstaande beoordeling gebruikt. Om duidelijker te af te lezen op welke componenten de verschillende containerplatformen scoren, is gebruik gemaakt van kleuren. In Tabel 12 is een legenda van de verschillende kleuren te zien.

Beschikbaarheid
In de multicriteria-analyse wordt beschikbaarheid beoordeeld op het beschikbaarheidspercentage. Hiervoor is gekozen omdat dit een standaard is in de ICT. Hoe lager het percentage, hoe lager de beschikbaarheid van de container oplossing is. Zie Tabel 10 voor mogelijke beschikbaarheidspercentages [49].

Beschikbaarheid	**Aantal negens**	**Downtime in één jaar tijd**			
		Dagen	**Uren**	**Minuten**	**Seconden**
90%	Eén	36	12	34	55
99%	Twee	3	15	39	29
99.9%	Drie		8	45	56
99.99%	Vier			52	35
99.999%	Vijf			5	15
99.9999%	Zes				31

Tabel 10: Beschikbaarheid in percentages

Kosten
De beoordeling van de kosten wordt in de multicriteria-analyse gedaan door middel van de maandelijkse prijs zonder korting[9]. Azure heeft namelijk de mogelijkheid een dienst voor één of drie jaar te reserveren en hierdoor wordt de maandelijkse prijs lager [50].

Andere beoordelingen
De beoordelingen op betrouwbaarheid, schaalbaarheid, veiligheid en features/mogelijkheden worden in de multicriteria-analyse gedaan door middel van plusjes (+). Hoe meer plusjes de component heeft, hoe beter het containerplatform op dit onderdeel scoort (zoals te zien is in Tabel 11). Deze beoordeling is in overleg met de stakeholders (lead consultant en business unit manager) gedaan, met behulp van de opgedane kennis in het literatuuronderzoek (in hoofdstuk 8.6.2).

Plusjes	**Betekenis**
+	Slecht
++	Gemiddeld
+++	Goed

Tabel 11: Uitleg beoordeling

Kleur	**Betekenis**
Groen	Het containerplatform scoort goed op dit onderdeel.
Oranje	Het containerplatform scoort minder op dit onderdeel dan andere platformen.
Rood	Het containerplatform scoort het laagst op dit onderdeel.

Tabel 12: Uitleg kleuren multicriteria-analyse

	Beschikbaarheid	**Betrouwbaarheid**	**Schaalbaarheid**	**Veiligheid**	**Kosten**	**Features/mogelijkheden**
Azure Kubernetes Services	99.95%	++	+++	++	€1.246,73	++
Azure Red Hat OpenShift	99.9%	+++	+++	+++	€5.665.36	+++
Virtual Machine met Docker	99.99%	++	+	+	€3.336,01	+

Tabel 13: Multicriteria-analyse container oplossingen

[9] De prijs is berekend aan de hand van de voorbeeldapplicatie in het literatuuronderzoek

Uit de gegevens in Tabel 13 kunnen de volgende conclusies worden getrokken:

- Azure Kubernetes Services: Over het algemeen geen grote nadelen. AKS heeft zowel positieve als negatieve eigenschappen, en in vergelijking met de andere opties het meest voordelig en stabiel.
- Azure Red Hat OpenShift: Meest veilige oplossing door het gebruik van de kennis van Red Hat op Security. Hierdoor is het wel erg prijzig en dus ook het meest nadelig bij de prijsvergelijking.
- Virtual Machine met Docker: Hoogste beschikbaarheid volgens SLA. Om dit mogelijk te maken moet veel beheer worden uitgevoerd. Door de slechte schaalbaarheid van Docker eindigt deze optie tussen de twee bovenstaande opties in (uiteindelijk het meest neutraal).

Alle drie de platformen blinken uit in verschillende componenten. Het is dus ook sterk afhankelijk van de container gebaseerde applicatie welke type hosting gebruikt kan worden.

8.7.1.2 Implementatiescenario's

Uit de multicriteria-analyse kunnen implementatiescenario's gehaald worden. Een implementatiescenario is een mogelijke vraag vanuit de klant over een containeroplossing. Aan de hand van de onderzochte componenten kan een keuze worden gemaakt voor een bepaalde containeroplossing.

Deze implementatiescenario's kunnen in de whitepaper worden gebruikt om een beslisboom voor de klant te ontwikkelen. Hieronder staan vier verschillende klantvragen. Elke vraag heeft zijn eigen componenten met verschillende index zwaarte. Hierdoor leidt elke klantvraag tot een ander implementatiescenario (met de container gebaseerde applicatie wordt de voorbeeldapplicatie uit het literatuuronderzoek in hoofdstuk 8.6.2.3 bedoeld).

Nr.	Klantvraag	Implementatiescenario
1.	"*Een ilionx klant heeft een container gebaseerde applicatie die moet worden gehost, de hosting moet voldoen aan: een goede beschikbaarheid, goede schaalbaarheid en de kosten moeten laag zijn.*"	De oplossing die hierbij zou passen is AKS. Deze heeft niet de hoogste beschikbaarheid, maar is wel goed schaalbaar en heeft de laagste kosten.
2.	"*Een ilionx klant heeft een container gebaseerde applicatie. De klant host de applicatie nu on-premise binnen een docker omgeving, maar wil deze graag hosten binnen Microsoft Azure.*	De oplossing die hierbij zou passen is VM's met Docker. Voor de migratie van de applicatie zou gebruik gemaakt kunnen worden van een "lift-and-shift".
3.	"*Een ilionx klant heeft een container gebaseerde applicatie die veel persoonsgegevens verwerkt. De applicatie hosting moet goed beveiligd zijn maar de kosten morgen niet meer zijn dan €4.000 per maand*"	De oplossing die hier bij zou passen is AKS. Deze is niet het veiligst, maar valt binnen het budget van de klant.
4.	"*Een ilionx klant heeft een container gebaseerde applicatie die moet worden gehost, de hosting moet voldoen aan: een goede schaalbaarheid, goede betrouwbaarheid en de hosting moet veilig zijn.*"	De oplossing die hierbij zou passen zou ARO zijn. Deze services is goed schaalbaar, betrouwbaar en is goed beveiligd.

Tabel 14: Van klantvraag naar implementatiescenario

In de bovenstaande tabel (Tabel 14) worden de verschillende componenten uit de multicriteria-analyse gebruikt. Aan deze componenten wordt steeds een andere index zwaarte (waarde) gehangen. Hierdoor komt steeds een ander containerplatform naar voren als beste keuze.

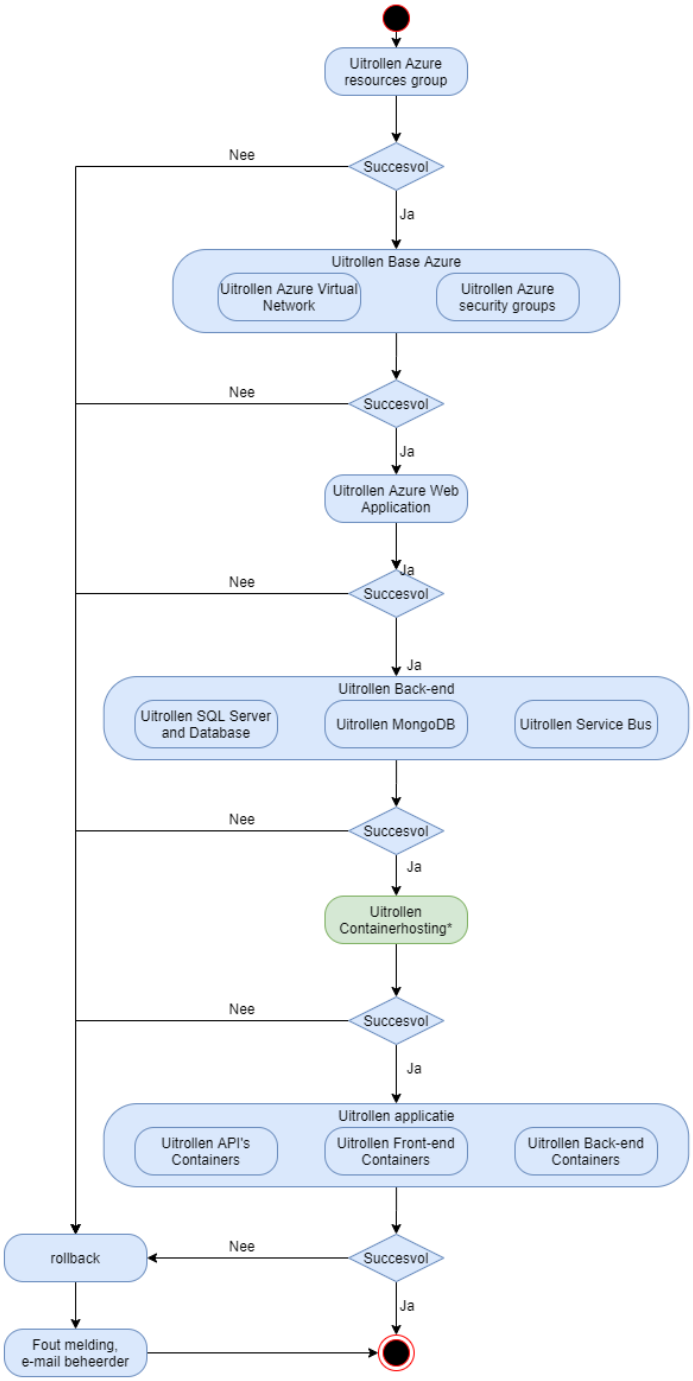
Naast de index zwarte heeft ook de applicatie die moet worden gehost veel invloed op de uiteindelijke containeroplossing. De applicatie bepaalt namelijk de benodigde resources. Het gebruik van veel of weinig resources bepaalt de uiteindelijke kosten voor de hosting van de applicatie.

Naast de container hosting heeft de container gebaseerde applicatie nog andere benodigdheden nodig (zie hoofdstuk 15.4.2.2.3). Deze services hebben ook invloed op de verschillende componenten. Een slecht beschikbare database zorgt ervoor dat de applicatie ook slecht bereikbaar is terwijl de hosting misschien wel gewoon werkt.

8.7.1.3 Workflows

De implementatiescenario's uit Tabel 14 zijn hieronder uitgewerkt naar workflows. Een workflow geeft een procesmatige weergave van de implementatie van een containeroplossingen. De onderstaande workflow in Figuur 8 geeft de uitrol weer van een voorbeeldapplicatie uit het literatuuronderzoek. In deze workflow wordt "uitrol containerhosting" benoemd. Dit proces is per containeroplossing hieronder uitgewerkt.

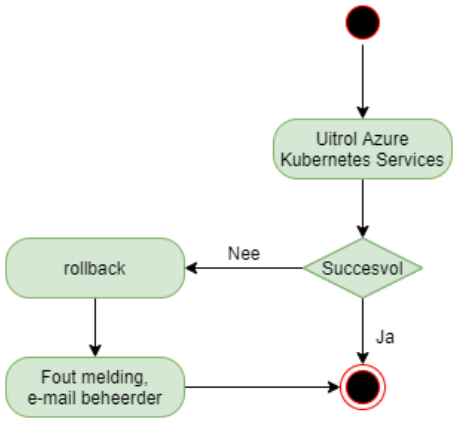
Het uitrollen van de voorbeeldapplicatie bestaat uit zes processen: base, firewall, back-end, uitrollen containerhosting, uitrollen applicatie en uitrollen front-end. Per proces wordt gecontroleerd of het succesvol is of niet. Bij een niet succesvol proces, wordt het gehele proces teruggedraaid en wordt de beheerder op de hoogte gesteld door middel van een e-mail.



Figuur 8: Workflow uitrollen voorbeeldapplicatie

8.7.1.3.1 Workflow Azure Kubernetes Services

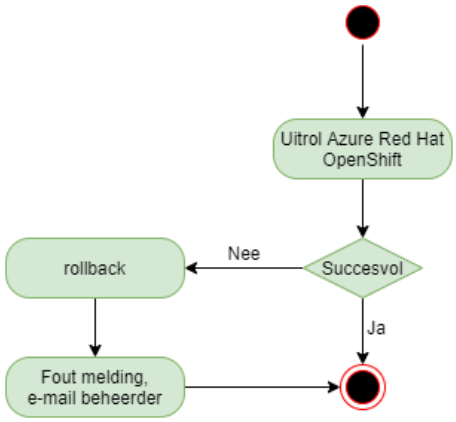
In Figuur 9 staat de workflow van AKS. De workflow bestaat uit één proces, namelijk “Uitrol Azure Kubernetes Services”. Dit geautomatiseerde proces is een service van Microsoft Azure. Door het proces van Microsoft Azure met de juiste parameters aan te spreken, wordt een AKS-cluster opgezet.



Figuur 9: Workflow AKS

8.7.1.3.2 Workflow Azure Red Hat OpenShift

In Figuur 10 staat de workflow van ARO. De workflow bestaat uit één proces, namelijk “Uitrol Azure Red Hat OpenShift”. Dit geautomatiseerde proces is een service van Microsoft Azure. Door het proces van Microsoft Azure met de juiste parameters aan te spreken, wordt een ARO-cluster opgezet.

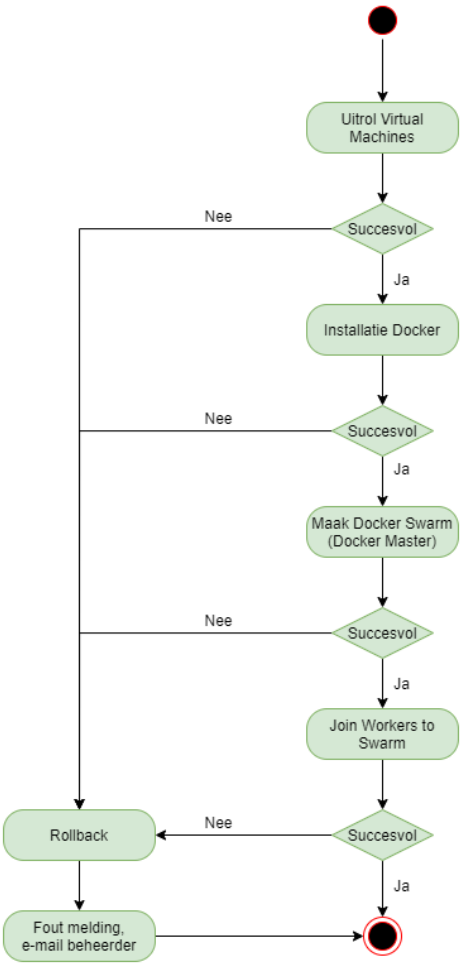


Figuur 10: Workflow ARO

8.7.1.3.3 Workflow Virtual Machine met Docker

In Figuur 11 staat de workflow van VM met Docker. De workflow gaat ervan uit dat meerdere VM’s worden uitgerold. Dit is nodig om een Docker Swarm aan te maken. Een Docker Swarm bestaat namelijk uit minimaal één master-node en meerdere worker-nodes.

In de workflow worden eerst de VM’s uitgerold. Zodra dit voltooid is, wordt de installatie van docker uitgevoerd. Daarna worden één of meerdere nodes gepromoot naar master-nodes. Vervolgens kunnen de overige nodes de Docker Swarm joinen als worker-node. Als al het bovenstaande succesvol is verlopen, komt een Docker Swarm beschikbaar waarop containers kunnen worden gehost.



Figuur 11: Workflow VM met Docker

8.7.2 Technisch ontwerp

In het technisch ontwerp wordt het functionele concept omgezet naar een technische oplossing. Eén van de ontworpen workflows uit het functioneel ontwerp wordt omgezet naar een playbook. Dit is toegepast in een PoC. Met deze PoC moet kunnen worden aangetoond of de workflows ook in de praktijk toepasbaar zijn. Hiermee is tevens deelvraag zeven beantwoord.

In deze PoC wordt de workflow van AKS uitgewerkt naar een playbook. Er is voor AKS gekozen omdat deze uit implementatiescenario één (zie Tabel 14) als best naar voren is gekomen. In dit playbook wordt door middel van ARM een volwaardige infrastructuur gedeployd. Deze is voorzien van AKS, Azure Container Registry (ACR), SQL-database, Azure Services Bus en Resources Group met een Virtual network en Netwerk Security Group (NSG). Zie voor verdere uitwerking van de ARM-templates hoofdstuk 8.7.2.3.

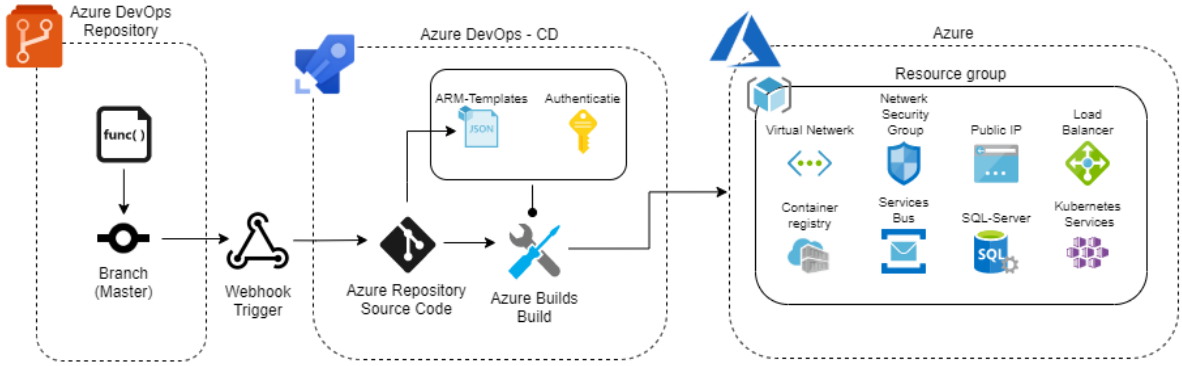
Op deze infrastructuur wordt vervolgens een container gebaseerde applicatie uitgerold. Hierbij is niet de voorbeeldapplicatie waarmee het onderzoek is gedaan gebruikt gezien de technische moeilijkheidsgraad en de beschikbare tijd om dit te kunnen ontwikkelen. Hiervoor is een alternatieve applicatie gebruikt die technisch gemakkelijker in gebruik is, waarmee de opdrachtgever akkoord is gegaan. Met deze alternatieve applicatie is aantoonbaar of de workflows functioneel zijn. Zie voor verdere details over de applicatie in hoofdstuk 8.7.2.2.

8.7.2.1 Pipeline ontwerp

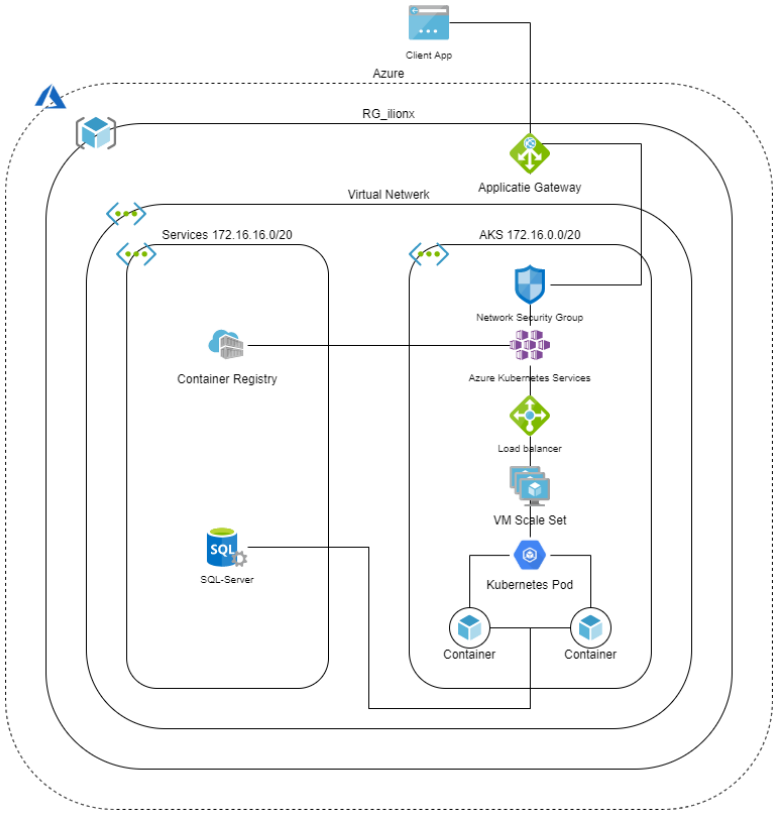
Voordat verder wordt ingegaan op de pipelines, die de infrastructuur en applicatie uitrollen, wordt eerst het ontwerp van de pipelines uitgewerkt. De playbook bestaat uit twee pipelines; één pipeline die de infrastructuur uitrolt en één pipeline die de applicatie uitrolt.

8.7.2.1.1 Infrastructuur pipeline

Om de implementatie van de infrastructuur inzichtelijk te maken is een deployment infrastructuur, zoals te zien is in Figuur 12, gemaakt. De technische inrichting van de pipeline wordt verder toegelicht in hoofdstuk 8.7.2.4.2. De uitwerking van de infrastructuur binnen Azure wordt inzichtelijk gemaakt in Figuur 13, waarin de verschillende uitgerolde componenten en de verschillende verbindingen hiertussen, staan beschreven.



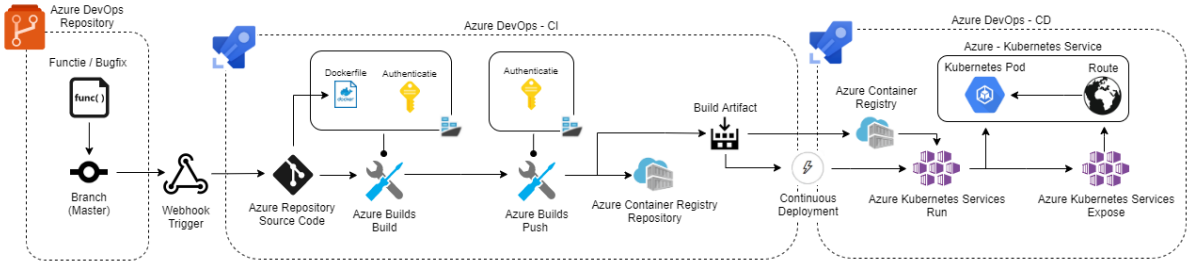
Figuur 12: Deployment infrastructuur



Figuur 13: Te deployen infrastructuur

8.7.2.1.2 Applicatie pipeline

Om de implementatie van de applicatie inzichtelijk te maken, is de deployment applicatie in Figuur 14 opgesteld. De technische inrichting van de pipeline wordt verder toegelicht in hoofdstuk 8.7.2.4.4.



Figuur 14: Deployment applicatie

8.7.2.2 Applicatie

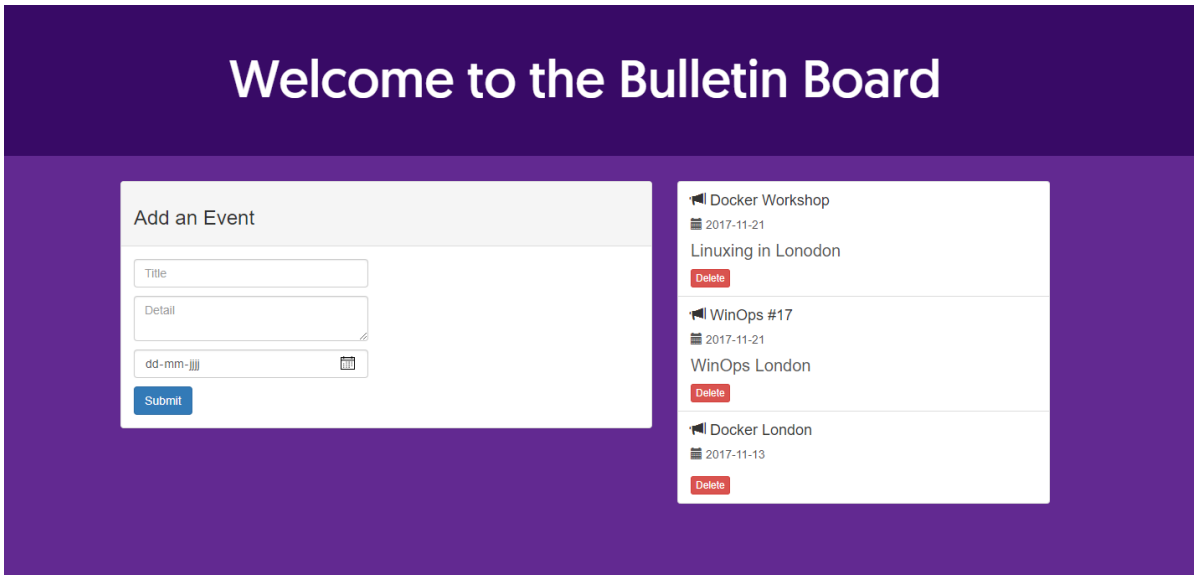
De gebruikte container gebaseerde applicatie in het PoC wijkt af van de voorbeeldapplicatie in Figuur 5, welke gebruikt is om de verschillende componenten te testen in het literatuuronderzoek. Er is gekozen voor het niet gebruiken van deze voorbeeldapplicatie door advies vanuit ilionx, omdat de applicatie te uitgebreid is om te gebruiken voor een PoC.

De applicatie die nu wordt gebruikt voor de PoC is de “bulletin-board-app” [51]. Deze applicatie is ontworpen als container gebaseerde applicatie en is hierdoor standaard voorzien van een Dockerfile (zie Figuur 15). Met de Dockerfile kan de applicatie straks worden gebuild tot containerimage. Deze containerimage wordt uiteindelijk opgehaald uit de container registry en wordt uitgevoerd binnen Kubernetes.

```
1  FROM node:current-slim
2
3  WORKDIR /usr/src/app
4  COPY package.json .
5  RUN npm install
6
7  EXPOSE 8080
8  CMD [ "npm", "start" ]
9
10 COPY . .
```

Figuur 15: Dockerfile

De container gebaseerde applicatie is een single frontpage applicatie, waarop standaard drie events te zien zijn (Zie Figuur 16). Door middel van een formulier (via Add dan Event) zijn hier nieuwe events aan toe te voegen. De applicatie maakt geen gebruik van externe diensten van Microsoft Azure. Gegevens worden in de container opgeslagen waardoor de moeilijkheidsgraad van het deployen van de applicatie in het PoC wordt verlaagd. Hierdoor is meer tijd over voor het automatisch uitrollen van het gehele playbook.

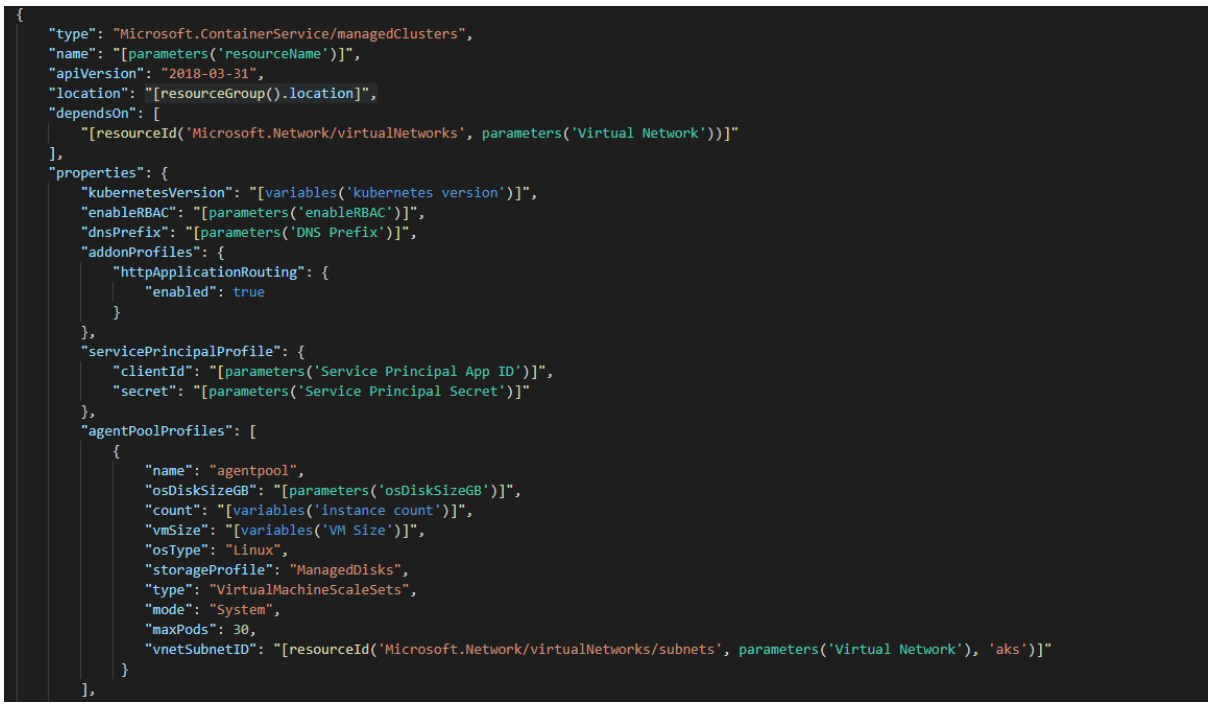


Figuur 16: Frontpage container gebaseerde applicatie

8.7.2.3 ARM-templates

De infrastructuur wordt uitgerold door middel van ARM-templates. ARM-templates zijn geschreven met de JavaScript Object Notation (JSON). In ARM-templates staat gescript welke infrastructuur binnen Microsoft Azure moet worden uitgerold.

Voor elke ARM-template is een "*.deploy.json"[10] en een "*.Deploy_parameters.json"[10] gemaakt. In het "*.deploy.json"[10] file worden de verschillende resources gedefinieerd, waarbij gebruik wordt gemaakt van parameters. Aan de hand van deze parameters kunnen bijvoorbeeld de resource group, locatie en naam van de resources makkelijk worden aangepast. In Figuur 17 is een deel van de deployment te zien van het "AKS.deploy.json".



```
{
    "type": "Microsoft.ContainerService/managedClusters",
    "name": "[parameters('resourceName')]",
    "apiVersion": "2018-03-31",
    "location": "[resourceGroup().location]",
    "dependsOn": [
        "[resourceId('Microsoft.Network/virtualNetworks', parameters('Virtual Network'))]"
    ],
    "properties": {
        "kubernetesVersion": "[variables('kubernetes version')]",
        "enableRBAC": "[parameters('enableRBAC')]",
        "dnsPrefix": "[parameters('DNS Prefix')]",
        "addonProfiles": {
            "httpApplicationRouting": {
                "enabled": true
            }
        },
        "servicePrincipalProfile": {
            "clientId": "[parameters('Service Principal App ID')]",
            "secret": "[parameters('Service Principal Secret')]"
        },
        "agentPoolProfiles": [
            {
                "name": "agentpool",
                "osDiskSizeGB": "[parameters('osDiskSizeGB')]",
                "count": "[variables('instance count')]",
                "vmSize": "[variables('VM Size')]",
                "osType": "Linux",
                "storageProfile": "ManagedDisks",
                "type": "VirtualMachineScaleSets",
                "mode": "System",
                "maxPods": 30,
                "vnetSubnetID": "[resourceId('Microsoft.Network/virtualNetworks/subnets', parameters('Virtual Network'), 'aks')]"
            }
        ],
```

Figuur 17: AKS.deploy.json

De parameters die zijn gedefinieerd in de "*.deploy.json"[10], worden verwerkt in de "*.Deploy_parameters.json"[10]. De paramaters staan apart om zo gemakkelijk bij een volgende deployment te kunnen worden aangepast. In Figuur 18 staat een voorbeeld van de paramaters van van "AKS.deploy.json" gedefinieerd in "AKS.deploy_ parameters.json".



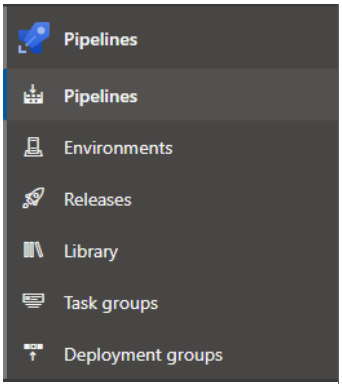
```
    "parameters": {
        "resourceName": {
          "value": "AKSilionx"
        },
        "networkPlugin": {
          "value": "azure"
        },
        "DNS Prefix": {
          "value": "ilionxAKSCluster"
        },
        "enablePrivateCluster": {
          "value": false
        },
```

Figuur 18: AKS deployment file

[10] * = naam van de resources bijvoorbeeld "AKS.deploy.json" en "AKS.deploy_parameters.json"

8.7.2.4 Inrichting Azure DevOps

De pipeline wordt gerealiseerd met Azure DevOps. Dit gebeurt door middel van het onderdeel pipelines. Deze functie, zoals weergegeven in Figuur 19, maakt het mogelijk om door middel van YAML-files een infrastructuur of applicatie te deployen.

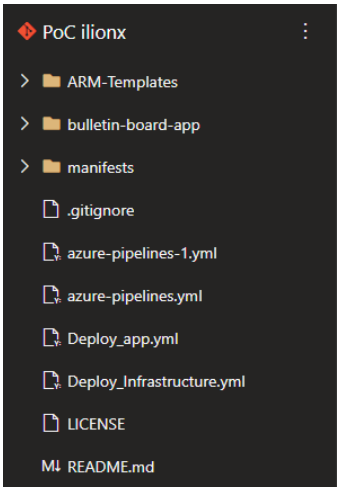


Figuur 19: Azure DevOps pipelines functie

Bij het gebruik van de pipeline maakt Azure DevOps gebruik van een agent. De agent zorgt ervoor dat de pipeline wordt doorlopen. Binnen dit project wordt een agent gebruikt met 'Ubuntu-18.04' [52].

8.7.2.4.1 Openhalen code uit Azure Repos

Naast het gebruik van een agent om de pipeline te kunnen doorlopen, moet ook verschillende code worden opgehaald voor zowel de deployment van de infrastructuur als voor de applicatie. Deze code is opgeslagen in een repository binnen Azure Repos (zie Figuur 20).



Figuur 20: Azure Repos

Azure Repos zorgt naast de opslag ook voor het mergen van nieuwe code met de al bestaande code. Dit zorgt ervoor dat er een automatisch versiebeheer beschikbaar is. Verder ondersteunt Azure Repos ook verschillende branches die weer een afsplitsing zijn van de master repository [53].

8.7.2.4.2 Infrastructuur pipeline

De pipeline deployment infrastructuur zorgt voor de implementatie van de infrastructuur binnen Microsoft Azure. Voordat de pipeline begint met uitrollen van de infrastructuur, worden eerst de gedeelde variabelen opgehaald (zie Figuur 21). Hierin staat de naam van de resource group, regio, abonnement id en abonnementsnaam gedefinieerd. Door deze onderdelen in variabelen te zetten, kan de pipeline snel worden gedeployd in een andere resource group, regio of Azure abonnement.

```
variables:
- group: SharedVariablesPoC
```

Figuur 21: Aanspreken gedeelde variabelen

Het eerste onderdeel van de pipeline is het aan maken van een resource group. Dit gebeurt door middel van een 'Bash-script' (zie Figuur 22). Binnen deze resource group worden de verschillende infrastructuurcomponenten geïmplementeerd.

```
- task: AzureCLI@2
  displayName: 'Deploy Resource group'
  inputs:
    azureSubscription: '$(AzureConnect)'
    scriptType: 'bash'
    scriptLocation: 'inlineScript'
    inlineScript: |
      az deployment sub create \
        --name $RGNAME \
        --location $LOCATION \
        --template-uri "https://raw.githubusercontent.com/Azure/azure-docs-json-samples/master/azure-resource-manager/emptyRG.json" \
        --parameters rgName=$RGNAME rgLocation=$LOCATION
```

Figuur 22: Aanmaken resource group

Nadat de resource group succesvol is geïmplementeerd, worden de verschillende infrastructuur componenten door middel van ARM-templates uitgerold. Dit proces bestaat per infrastructuur component steeds uit twee fases (zie Figuur 23):

- Fase 1: De deployment van de infrastructuurcomponent door middel van het ARM-template en het parameter file.
- Fase 2: Door middel van “ARM Outputs” worden de gedefinieerde outputs in het ARM-template als variable in de pipeline geplaatst [54].

Deze fases worden per infrastructuur component herhaald totdat de gehele infrastructuur is geïmplementeerd binnen Microsoft Azure. Zodra dat succesvol voltooid is, wordt de pipeline afgesloten en is de infrastructuurpipeline afgerond.

```
- task: AzureResourceManagerTemplateDeployment@3
  displayName: 'Deploy AKS'
  inputs:
    deploymentScope: 'Resource Group'
    azureResourceManagerConnection: '$(AzureConnect)'
    subscriptionId: '$(SubId)'
    action: 'Create Or Update Resource Group'
    resourceGroupName: '$(RgName)'
    location: '$(Location)'
    templateLocation: 'Linked artifact'
    csmFile: 'ARM-Templates/AKS_Deploy.json'
    csmParametersFile: 'ARM-Templates/AKS_Deploy_parameters.json'
    deploymentMode: 'Incremental'

- task: ARM Outputs@5
  displayName: 'Import outputs AKS to pipeline'
  inputs:
    ConnectedServiceNameSelector: 'ConnectedServiceNameARM'
    ConnectedServiceNameARM: '$(AzureConnect)'
    resourceGroupName: '$(RgName)'
    whenLastDeploymentIsFailed: 'fail'
```

Figuur 23: De twee fases binnen de pipeline

8.7.2.4.3 Registeren serviceconnecties

Na een succesvolle implementatie van de verschillende infrastructuurcomponenten, moet voor de implementatie van de applicatie serviceconnecties worden gelegd naar de geïmplementeerde infrastructuur componenten. Een serviceconnectie is het verlenen van toestemming aan Azure DevOps voor het gebruik van een services binnen Microsoft Azure [55]. Het leggen van deze serviceconnecties kan op twee manieren:

1. Handmatig via de portal van Azure DevOps [55];
2. Via de Representational state transfer (REST) API van Azure DevOps [56].

Het aanmaken van een serviceconnectie binnen het portal kan door het aanmelden in een pop-scherm en hier de credentials van Microsoft Azure in te voeren. Daarna is het mogelijk de resource te selecteren waarmee een serviceconnectie moet worden gelegd. Zodra dit is voltooid, is succesvol een serviceconnectie gelegd met de service van Microsoft Azure en kan deze worden gebruikt binnen een pipeline.

Het is ook mogelijk om deze serviceconnectie via een de REST API van Azure DevOps aan te maken. In dit onderzoek is het momenteel al gelukt om een serviceconnectie te leggen met ACR door middel van een REST API call, zoals te zien is in Figuur 24[11]. Om de gehele pipeline automatisch te laten verlopen door middel van deze REST API calls zou ook de serviceconnectie van AKS moeten worden aangemaakt via deze REST API. Dit is tijdens dit onderzoek niet gelukt door gebrek aan tijd en omdat de techniek van de REST API binnen Azure DevOps nog nieuw is, waardoor er nog weinig bronnen zijn. In dit onderzoek is geprobeerd om aan de hand van "trail and error" (zie hoofdstuk 16) deze REST API calls te schrijven, waarvan er dus al één gelukt is en de tweede nog in ontwikkeling is.

```
$AzureDevOpsPAT = "[redacted]"
$OrganizationName = "[redacted]"
$ProjectName = "Afstuderen"
$nameACR = "[redacted]"

$AzureDevOpsAuthenicationHeader = @{Authorization = 'Basic ' + [Convert]::ToBase64String([Text.Encoding]::ASCII.GetBytes(":$($AzureDevOpsPAT)")) }

$UriOrga = "https://dev.azure.com/$($OrganizationName)/$($ProjectName)/"
$uriAccount = $UriOrga + "_apis/serviceendpoint/endpoints?api-version=5.1-preview.2"
$uriAccount1 = $UriOrga + "_apis/build/authorizedresources?api-version=5.1-preview.1"

$ServicesConnection = "{
        ""authorization"": {
          ""scheme"": ""ServicePrincipal"",
          ""parameters"": {
            ""loginServer"": ""$nameACR.azurecr.io"",
            ""servicePrincipalId"": ""[redacted]"",
            ""tenantId"": ""[redacted]"",
            ""serviceprincipalkey"": ""[redacted]""
          }
        },
        ""description"": """",
        ""name"": ""$nameACR"",
        ""type"": ""dockerregistry"",
        ""url"": ""https://$nameACR.azurecr.io"",
        ""isShared"": false,
        ""owner"": ""library"",
        ""data"": {
          ""registryId"": ""/subscriptions/[redacted]/resourceGroups/RG_AKS_ilionx/providers/Microsoft.ContainerRegistry/registries/$nameACR"",
          ""registrytype"": ""ACR"",
          ""spnObjectId"": """",
          ""subscriptionId"": ""[redacted]"",
          ""subscriptionName"": ""test""
        }
}"
```

Figuur 24[11]: REST API POST connectie ACR

8.7.2.4.4 Applicatie pipeline

In de applicatie pipeline wordt de applicatie uitgerold binnen de eerder geïmplementeerde infrastructuur. De pipeline bestaat uit twee fases met totaal drie stappen. In het eerste deel wordt de applicatie gebuild en gepusht naar de ACR (zie Figuur 25), in het tweede deel van de pipeline wordt de applicatie uitgerold binnen Kubernetes. Hieronder wordt uitgelegd wat deze verschillende fases inhouden.

```
- task: Docker@2
  displayName: Build and push an image to container registry
  inputs:
    command: buildAndPush
    repository: $(imageRepository)
    dockerfile: $(dockerfilePath)
    containerRegistry: $(dockerRegistryServiceConnection)
    tags: |
      $(tag)
```

Figuur 25: Build and push image to container registry

Applicatie builden

Het builden van een applicatie image gebeurt door middel van Docker. Voor deze taak is eerst de serviceconnectie met ACR nodig, vervolgens wordt met een taak de Dockerfile van de applicatie uit Azure Repos gehaald. Deze Dockerfile definieert hoe de container eruit moet gaan zien en welke onderdelen nodig zijn. Bij het builden worden een image naam en image tag meegeven die in het volgende proces gebruikt worden [57] [58].

[11] In deze figuur zijn access codes van Microsoft Azure weggelaten, omdat deze verwijzen naar de tenant van ilionx.

Applicatie pushen
Nadat de applicatie images gebuild is door het gebruik van een Dockerfile, wordt deze opgeslagen binnen ACR. Deze wordt dan in de release stage van de applicatie weer gebruikt. Voor het opslaan van de image wordt het Docker push commando gebruikt. Met dit commando worden de applicatie images over de serviceconnectie naar de ACR gestuurd [57] [58].

Applicatie release
De release van de applicatie binnen Kubernetes gebeurt door middel van deployment manifest (zie Figuur 27) en een services manifest (zie Figuur 26). Voor het uitvoeren van de twee manifesten is een services connectie nodig met AKS en ACR. Deze stage wordt automatisch getriggerd zodra de bovenstaande stage automatisch is afgesloten.

In het deployment manifest staat de naam van de container, waar de container images te vinden zijn in de ACR, en wat voor poort de applicatie bereikbaar moet zijn. In het servicemanifest staat wat voor type service moet worden aangemaakt, voor welke applicatie en op welke poort dit moet [59].

```yaml
apiVersion : apps/v1beta1
kind: Deployment
metadata:
  name: pocilionx
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: pocilionx
    spec:
      containers:
        - name: pocilionx
          image: acrilionx.azurecr.io/pocilionx
          ports:
          - containerPort: 8080
```

Figuur 27: Deployment manifest

```yaml
apiVersion: v1
kind: Service
metadata:
    name: pocilionx
spec:
    type: LoadBalancer
    ports:
    - port: 8080
    selector:
        app: pocilionx
```

Figuur 26: Services manifest

Zodra de deployment en het servicemanifest succesvol zijn uitgevoerd, is de applicatie beschikbaar op het publieke IP-adres en de gedefinieerde poort van AKS (zie Figuur 16).

8.8 Bedrijfsproducten

De onderstaande bedrijfsproducten zijn een deel van de oplossing van het vraagstuk van GRO-Cloud Solutions. Deze bedrijfsproducten bestaan uit een flyer en een whitepaper, welke hieronder zijn uitgelegd.

8.8.1 Flyer

Er is een Containerscan ontwikkeld waarmee een container gebaseerde applicatie op een juiste manier kan worden geïmplementeerd binnen Microsoft Azure. Deze containerscan is toegevoegd als aparte bijlage bij dit afstudeerverslag (Zie “Bijlage_Flyer_Containerscan.pdf”). De Containerscan biedt vijf speerpunten waaraan de hosting van de container gebaseerde applicatie moet voldoen. Aan de hand van deze Containerscan wordt een adviesrapport opgesteld. Dit rapport bevat concrete aanbevelingen voor het implementatietraject in de vorm van een stappenplan, een overzicht van de benodigde infrastructuur componenten, een architectuurtekening en een deployment workflow.

8.8.2 Whitepaper

Voor de geïnteresseerde klanten van ilionx over containerhosting is een whitepaper ontwikkeld. Hiermee wordt de klant geïnformeerd en geadviseerd over de verschillende mogelijkheden van hosting binnen Microsoft Azure. Dit zijn Virtual Machine met Docker, Azure Kubernetes Services en Azure Red Hat OpenShift. Deze whitepaper is ook toegevoegd als aparte bijlage bij dit afstudeerverslag (Zie “Bijlage_Whitepaper_Containerhosting.pdf”). Verder wordt in deze whitepaper uitgelegd wat een container gebaseerde applicatie is en welke benodigdheden nodig zijn voor het hosten van een container. In de whitepaper is de multicriteria-analyse toegevoegd, waarin per containerplatform de verschillende eigenschappen met elkaar zijn vergeleken, om zo een overzichtelijk beeld te schetsen van de verschillende container gebaseerde oplossingen voor de klant voor het geven van een onderbouwd en betrouwbaar advies.

9. Evaluatie van het onderzoek

Na afloop van het onderzoek is hieronder geëvalueerd of in het onderzoek antwoord is gegeven op de opgestelde hoofd- en deelvragen. Verder wordt geëvalueerd of het ontwerp voldoet aan de opgestelde requirements uit de probleem- en behoefteanalyse en de randvoorwaarden uit de contextanalyse. Deze evaluatie is uitgevoerd door te controleren of elke requirement is toegepast en of deze ook op een juiste wijze is toegepast in het onderzoek.

9.1.1 Evaluatie hoofd- en deelvragen

In dit onderzoek zijn de volgende hoofd- en deelvragen opgesteld (Tabel 15 en Tabel 16). Deze vragen zijn in dit onderzoek in verschillende hoofdstukken onderzocht en uitgewerkt. In de evaluatie wordt verwezen naar waar de hoofd- en deelvragen zijn beantwoord.

Nr.	Hoofdvraag	Beantwoord
1.	Wat is de meest passende geautomatiseerde infrastructuur oplossing voor mijn container gebaseerde applicatie binnen Microsoft Azure?	Is beantwoord aan de hand van de uitwerking van de deelvragen en door middel van de whitepaper.

Tabel 15: Evaluatie hoofdvraag

Nr.	Deelvraag	Beantwoord
1.	Wat wordt onder container applicatie verstaan?	Wordt beantwoord in hoofdstuk 8.6.1 en 15.4.3.1
2.	Welke verschillende mogelijkheden zijn binnen Azure voor het hosten van een container applicatie?	Worden beantwoord in hoofdstuk 8.6.2 en 15.4.3.2
3.	Wat zijn de kosten van deze verschillende oplossingen van Azure?	
4.	Voorziet Azure in alle benodigde componenten?	
5.	Wat is nodig voor het geautomatiseerd uitrollen van de infrastructuur binnen Azure?	Wordt beantwoord in hoofdstuk 8.6.3 en 15.4.3.3
6.	Op welke manier kan de oplossing het best aan de klant gepresenteerd worden?	Wordt beantwoord in hoofdstuk 8.1
7.	Hoe demonstreer ik het automatiseringsproces?	Wordt beantwoord in hoofdstuk 8.7

Tabel 16: Evaluatie deelvragen

9.1.2 Evaluatie requirements

In de evaluatie van de requirements is geëvalueerd of alle producten voldoen aan de vooraf opgestelde requirements. De evaluatie is uitgewerkt in Tabel 17, deze tabel beschrijft de volgende onderdelen:

- Nummer van de requirement;
- De opgestelde requirement;
- Of de requirement verwerkt is in het onderzoek, aangeven met Ja (groen);
- Uitleg over hoe en waar de requirement verwerkt is.

Nr.	Requirement	Verwerkt (Ja/Nee)	Uitleg
U1	De uitkomsten van het onderzoek moeten worden verwerkt in een whitepaper.	Ja	De whitepaper is geschreven op basis van de gevonden resultaten in het literatuuronderzoek en functioneel ontwerp.
U2	De whitepaper moet de klant duidelijkheid[12] geven over de hosting van hun container gebaseerde applicatie.	Ja	De whitepaper gaat in op de verschillende soorten hosting binnen Microsoft Azure en de verschillende benodigdheden.
U3	De whitepaper moet worden geschreven voor de doelgroep[13] die niet noodzakelijk een diepgaande IT-kennis heeft.	Ja	De whitepaper is zo geschreven zodat deze voor een leek te begrijpen is.
U4	In de whitepaper moet duidelijk[12] zijn wat de voordelen van container gebaseerde applicaties hosting ten opzichte van traditionele applicaties hosting is.	Ja	In de whitepaper wordt uitgelegd wat het verschil is tussen een VM en een container.
U5	Door middel van het onderzoek moet GRO-Cloud Solutions worden geadviseerd over het gebruik van de hosting van container gebaseerde applicaties.	Ja	In de whitepaper wordt op basis van de gemaakte multicriteria-analyse een advies gegeven over welke hosting het best bij welke container gebaseerde applicatie.
U6	Bij het onderzoek naar container gebaseerde applicaties moet rekening worden gehouden met DevOps.	Ja	In het literatuuronderzoek en de in whitepaper wordt rekening gehouden met DevOps. Hierbij is uitgelegd wat de relatie is tussen containers en DevOps.
U7	Bij het onderzoek naar container gebaseerde applicaties moet rekening worden gehouden met het Continuous Integration proces (CI).	Ja	In het literatuuronderzoek en in de whitepaper wordt rekening gehouden met CI. Hierbij is uitgelegd wat de relatie is tussen containers en CI.
U8	Bij het onderzoek naar container gebaseerde applicaties moet rekening worden gehouden met het Continuous Deployment proces (CD).	Ja	In het literatuuronderzoek en de whitepaper wordt rekening gehouden met CD. Hierbij is uitgelegd wat de relatie is tussen containers en CD.
U9	GRO-Cloud Solutions moet aan de hand van dit onderzoek in staat zijn om een container infrastructuur technisch op te bouwen.	Ja	GRO-Cloud Solutions is door dit onderzoek in staat om een container infrastructuur technisch op te bouwen.
U10	GRO-Cloud Solutions moet aan de hand van dit onderzoek meer kennis opdoen van containers.	Ja	GRO-Cloud Solutions kan terugvallen op de opgedane kennis in dit onderzoek, hierdoor is de kennis over containers bij personeel van unit vergroot.

[12] “Duidelijkheid” is het overbrengen van kennis. Dit is niet anders kwantificeerbaar volgens SMART methode.
[13] Hiermee worden alle mogelijke lezers van de whitepaper bedoeld.

U11	GRO-Cloud Solutions moet na afloop van dit onderzoek meer duidelijkheid[12] hebben gekregen over hetgeen wat er met containers gedaan kan worden	Ja	GRO-Cloud Solutions heeft door het uitgevoerde literatuuronderzoek een beter beeld gekregen van de verschillende mogelijkheden van containers.
U12	Ter oplossing van het vraagstuk moet een whitepaper worden gemaakt.	Ja	Eén van de ontwikkelde beroepsproducten is een whitepaper.
U13	De whitepaper moet impliceren dat GRO-Cloud Solutions kennis heeft opgedaan van container hosting.	Ja	Uit feedback over de whitepaper van het personeel van de unit, blijkt dat de unit GRO-Cloud Solutions kennis heeft opgedaan van container hosting.
U14	De whitepaper moet een leidraad vormen voor de hosting van de container gebaseerde applicatie van de klant.	Ja	De whitepaper geeft de lezer een leidraad over de verschillende keuzes omtrent containerhosting.
U15	In de whitepaper moeten de verschillende mogelijke oplossingen van container hosting worden aangeraden.	Ja	De whitepaper geeft advies over drie verschillende containerplatformen, zonder dat hierbij een specifiek platform wordt geadviseerd.
U16	De in de whitepaper aangeraden mogelijkheden moeten worden omgezet naar workflows voor de afdeling GRO-Cloud Solutions.	Ja	In de ontwerpdocumentatie zijn de containerplatformen omgezet naar workflows.
U17	Aan de hand van de workflows moet duidelijk[12] worden hoe GRO-Cloud Solutions automatisch container applicaties infrastructuren technisch kan implementeren.	Ja	De workflows geven duidelijk weer welke stappen per containerplatform genomen moeten worden om deze technisch te implementeren.
U18	Om aan te tonen dat de werking van de workflows correct is en functioneert, wordt één workflow omgezet naar één playbook.	Ja	In het technisch ontwerp en PoC is een playbook ontwikkeld die één workflow van AKS kan uitrollen.
S1	Het playbook dient binnen Azure DevOps uitgevoerd te kunnen worden.	Ja	Het playbook wordt binnen Azure DevOps uitgevoerd.
S2	Het playbook dient geschreven te worden door middel van Azure Resource Manager.	Ja	De infrastructuur wordt uitgerold door middel van ARM templates.

Tabel 17: Evaluatie requirements

9.1.3 Evaluatie randvoorwaarden

In de evaluatie van de randvoorwaarden is geëvalueerd of alle producten voldoen aan de vooraf opgestelde randvoorwaarden. De evaluatie gebeurt in Tabel 18, deze tabel beschrijft de volgende onderdelen:

- Nummer van randvoorwaarde;
- De opgestelde randvoorwaarde;
- Of de producten aan de randvoorwaarde voldoen, aangeven met Ja (groen);
- Uitleg over hoe de randvoorwaarde verwerkt is.

Nr.	**Randvoorwaarde**	**Verwerkt (Ja/Nee)**	**Uitleg**
R1	De oplossingen moeten in het Nederlands worden geschreven.	Ja	Het afstudeerverslag, de flyer en de whitepaper zijn in het Nederlands geschreven.
R2	De oplossingen moeten toegespitst zijn op de werkwijze van GRO-Cloud Solutions.	Ja	De oplossingen sluiten aan bij de denk- en werkwijze van GRO-Cloud Solutions.
R3	De oplossing moet beperkt worden tot containerplatformen.	Ja	De oplossing gaat in op de verschillende containerplatformen. FaaS- en PaaS-platformen worden wel benoemd, maar niet verder onderzocht in dit onderzoek.
R4	Het uitgangspunt voor GRO-Cloud Solutions is dat gewerkt wordt met Microsoft Azure.	Ja	De oplossing is toegespitst op het gebruik van Microsoft Azure.
R5	Voor de automation tools moet gebruik worden gemaakt van Azure DevOps met Azure Resource Manager (ARM).	Ja	Voor het playbook wordt gebruik gemaakt van Azure DevOps en ARM.

Tabel 18: Evaluatie randvoorwaarden

10. Conclusie van het onderzoek

In dit onderzoek is gezocht naar een antwoord op de vraag: *"Wat is de meest passende geautomatiseerde infrastructuur oplossing voor mijn container gebaseerde applicatie binnen Microsoft Azure?"* Hiervoor is onderzoek gedaan naar de verschillende mogelijkheden van het hosten van een container gebaseerde applicatie binnen Microsoft Azure aan de hand van verschillende analyses en literatuuronderzoek.

Bij de vergelijking, door middel van een multicriteria-analyse, van de container platformen Virtual Machine met Docker, Azure Kubernetes Services en Azure Red Hat OpenShift, kwam Azure Kubernetes Services het beste uit de test. Dat Azure Kubernetes Services het beste uit de test kwam, komt door het gebruikte implementatiescenario in dit onderzoek. Bij een andere klantvraag, met dus andere een implementatiescenario, is de kans aanwezig dat een ander containerplatform beter passend is. De opdrachtgever wilde ook onderzocht hebben hoe een container infrastructuur automatisch geïmplementeerd kan worden. Voor het automatisch uitrollen van een infrastructuur is aan de hand van een workflow, een playbook ontwikkeld. Dit playbook maakt gebruik van Azure Resource Manger en Azure DevOps. Het playbook rolt aan de hand hiervan de infrastructuur en de applicatie uit. Het automatisch uitrollen is tijdens dit onderzoek niet volledig gelukt. Het automatisch leggen van serviceconnecties is binnen Azure DevOps nog een handmatige actie en kan worden geautomatiseerd door middel van een REST API. Omdat dit nog een nieuwe techniek is, is dit tijdens het onderzoek niet gelukt door het gebrek aan tijd en bronnen. De serviceconnectie met Azure Container Registry door middel van een REST API is al wel gelukt. Om het playbook volledig automatisch te laten uitrollen, zou er nog een REST API call moeten worden geschreven voor de serviceconnectie van Azure Kubernetes Services. Hierdoor worden in het Proof of Concept de serviceconnecties nog handmatig gelegd en bestaat het automatiseringsproces nog uit twee verschillende Azure DevOps pipelines (in plaats van één pipeline die alles uitvoert).

Ten slotte zijn er twee beroepsproducten, een whitepaper en flyer, ontwikkeld. In de whitepaper wordt de lezer geïnformeerd en geadviseerd over de verschillende container platformen binnen Microsoft Azure. De flyer, waarvan later in het onderzoek naar voren kwam dat deze nodig was, informeert over een Containerscan waarmee onderzocht wordt wat voor containerplatform het best aansluit op de container gebaseerde applicatie van de klant. Aan de hand van deze twee beroepsproducten wordt de klant van ilionx geadviseerd over de verschillende benodigdheden en het best passende containerplatform bij hun container gebaseerde applicatie. Met deze adviezen kan de klant een zelfgekozen container infrastructuur samenstellen. De hoofdvraag is hiermee tevens beantwoord. Verder is aan alle opgestelde requirements en randvoorwaarden in dit onderzoek voldaan.

11. Discussie

In dit hoofdstuk wordt ingegaan op de conclusies van het onderzoek door het koppelen van resultaten aan verwachtingen, beperkingen en implicaties van het onderzoek. De validiteit van het onderzoek wordt ook beargumenteerd. Hiernaast worden suggesties voor het vervolgonderzoek besproken.

11.1 Validiteit

In dit onderzoek zijn voor het uitvoeren van analyses, het doen van literatuuronderzoek en voor de ontwerpdocumentatie bewezen methodieken gebruikt. Deze methodieken komen voort uit het ontwerponderzoek van Saxion Hogeschool [1] welke als leidraad is genomen bij dit onderzoek. Bij het literatuuronderzoek is voornamelijk gebruik gemaakt van valide websites en bronnen. Hierbij zijn voornamelijk bronnen van de Microsoftwebsite gebruikt. Dit is gedaan omdat Microsoft het product heeft gemaakt, en dus het meest betrouwbaar is als bron omdat zij hierover de meeste kennis hebben. Op basis van het gebruik van deze methodieken kan gesteld worden dat bij een herhaling van een dergelijk onderzoek, de uitkomsten hetzelfde zouden zijn en dat daarmee de resultaten van dit onderzoek valide zijn.

11.2 Resultaten interpreteren

De vraag van ilionx was om een whitepaper te maken om aan de hand daarvan in de toekomst gemakkelijker klanten te kunnen informeren over hun container gebaseerde applicatie. In dit onderzoek is deze whitepaper daadwerkelijk ontwikkeld. Hiernaast zijn verschillende workflows gemaakt, waarvan één verder is ontwikkeld naar een playbook. Deze workflow functioneert, waarmee aangetoond wordt dat de workflows hoogstwaarschijnlijk functioneren, wat overeen komt met de verwachting GRO-Cloud Solutions.

11.3 Beperkingen

Er zijn echter wel een aantal beperkingen aan het uitgevoerde onderzoek. Ten eerste leverde het literatuuronderzoek geen ontwerpprincipes op, wat wel in het ontwerponderzoek van Saxion Hogeschool [1] staat beschreven. Dit is niet gedaan omdat het doel van het literatuuronderzoek in dit onderzoek alleen het opdoen van kennis voor het maken van de whitepaper en flyer was. Dit heeft in dit onderzoek dus voor weinig beperkingen gezorgd omdat de oplossing niet vanuit ontwerpprincipes gemaakt hoefde te worden.

Ten tweede is geen inspirerend voorbeeld gebruikt, welke ook in de methode [1] van Saxion Hogeschool staat beschreven. Hier is voor gekozen omdat dit geen nieuwe inzichten zou geven en tevens optioneel is in de methode van Saxion Hogeschool.

In dit onderzoek is misschien te eenzijdig; voornamelijk bronnen gebruikt van de Microsoft website. Dit is gedaan omdat binnen ilionx voornamelijk met Microsoft producten wordt gewerkt en ilionx hier dus al de meeste kennis over beschikt. De informatie is dus niet van vele verschillende bronnen verzameld, wat voor enige beperking kan hebben gezorgd.

Door ilionx was besloten dat één workflow getest moest worden aan de hand van een playbook, waarbij moet worden aangetoond dat deze workflow functioneert. Het is echter wel zo dat het functioneren van deze ene workflow, niet bewijst dat de andere workflows ook werken.

Hiernaast is in dit alleen naar CaaS- en IaaS-platformen, en niet naar PaaS- en Faas-platformen onderzoek gedaan omdat dit volgens randvoorwaarde drie niet nodig was. Het kan dus zijn dat als deze platformen wel waren onderzocht, de resultaten anders zouden zijn.

Ten slotte werd met Azure DevOps tegen problemen aangelopen in relatie tot het automatisch registeren van service connecties met Microsoft Azure. Dit komt waarschijnlijk doordat Azure DevOps nog erg nieuw en nog volop in ontwikkeling is. Het kan dus zijn dat als dit onderzoek een paar jaar later zou worden uitgevoerd, er niet meer tegen een dergelijk probleem aangelopen wordt. De workaround die hiervoor is bedacht, is het gebruiken van een REST API. Het is nog niet helemaal duidelijk of dit volledig gaat functioneren. Zo niet, zou de implementatie bestaan uit twee verschillende pipelines met een handmatige actie daar tussen, in plaats van één pipeline die volledige automatisch werkt.

11.4 Implicaties

Door het onderzoek naar container gebaseerde applicaties is GRO-Cloud Solutions in staat om op een juiste manier een klant te voorzien van goed advies op het gebied van container hosting, aan de hand van een flyer met de Containerscan en de whitepaper. Daarnaast heeft GRO-Cloud Solutions meer kennis opgedaan van container hosting en hierdoor in staat om een container infrastructuur automatisch uit te rollen binnen Microsoft Azure.

11.5 Aanbevelingen en vervolgonderzoek

Hieronder zijn, aan de hand van dit onderzoek, een aantal aanbevelingen beschreven welke mogelijk tot een vervolgonderzoek kunnen leiden.

De eerste aanbeveling is het gebruik van Terraform voor het automatisch uitrollen van de infrastructuur in plaats van ARM [60]. Deze aanbeveling wordt gedaan omdat Terraform platform-onafhankelijk is, waardoor alleen één programmeertaal geleerd hoeft te worden om meerdere Cloudproviders aan te spreken. Naast dat Terraform platform-onafhankelijk is, zijn de Terraform templates ook beter modulair, is de code makkelijker te valideren en de leesbaarheid van de Terraform templates veel groter dan die van ARM. Deze drie punten hebben vooral te maken met dat Terraform gebruik maakt van HashiCorp Configuration Language (HCL), wat in tegenstelling is met ARM welke gebruik maakt van JSON-format in een duidelijke taal. Dit bovenstaande is een advies voor een vergelijkbaar onderzoek, maar in dit specifieke onderzoek bij ilionx is dit niet van toepassing geweest omdat dit een Microsoft onderneming is.

Er wordt hiernaast aanbevolen om een vervolgonderzoek te doen naar het gebruik van PaaS- en FaaS-platformen binnen Microsoft Azure, naast alleen de CaaS- en IaaS-platformen zoals is gedaan in dit onderzoek. Als er ook onderzoek wordt gedaan naar andere platformen, geeft dit een betere en betrouwbaardere weergave van de verschillende mogelijkheden voor het hosten van een container gebaseerde applicatie binnen Microsoft Azure.

Ten slotte wordt aanbevolen om het personeel van GRO-Cloud Solutions te voorzien van scholing op het gebied van container infrastructuren. Als het personeel op dit gebied beter geschoold is, wordt er verwacht dat beter advies kan worden geven aan klanten over het hosten van hun container gebaseerde applicatie.

12. Nawoord

Ik heb het schrijven van dit afstudeerverslag als erg leerzaam ervaren. Ik ben erg tevreden met het eindresultaat en ik hoop dat ik met mijn onderzoek veel mensen kan bereiken zodat ik mijn opgedane kennis met hen kan delen.

Ik heb bij dit onderzoek ook tegenslagen ervaren, zoals het plots thuiswerken was even wennen. Hiernaast vond ik het lastig door mijn dyslexie om een goed lopend verslag te schrijven. Ook was het niet altijd even makkelijk om bepaalde zaken te vinden of uit te voeren. Uiteindelijk is alles gelukt, met hulp en ondersteuning uit mijn omgeving. Ik heb geleerd dat als ik doorzet, veel kan bereiken.

Wouter Damman

Assen, juni 2020

13. Bibliografie

[1] K. Haveman, E. Hageraats en S. v. d. Linden, „Ontwerponderzoek," Saxion, Enschede, 2016.

[2] A. Cox, „Business Requirements vs Functional Requirements? Who Cares?," netmind, September 2017. [Online]. Available: https://netmind.net/business-vs-functional-requirements-who-cares/. [Geopend 20 mei 2020].

[3] EenBlogjeOm, „business, user en system requirements," 28 Januari 2018. [Online]. Available: http://www.raamstijn.nl/eenblogjeom/index.php/requirements/4255-business-user-en-system-requirements.

[4] Google cloud, „Wat zijn containers en welke voordelen bieden ze?," z.d.. [Online]. Available: https://cloud.google.com/containers?hl=nl. [Geopend 17 Maart 2020].

[5] Microsoft Azure, „Wat is DevOps?," z.d. -f. [Online]. Available: https://azure.microsoft.com/nl-nl/overview/what-is-devops/. [Geopend 17 Maart 2020].

[6] ecloudvalley, „CONTAINERS," ecloudvalley, z.d.. [Online]. Available: https://www.ecloudvalley.com/solutions/containers/. [Geopend 7 mei 2020].

[7] M. Rouse, „Azure Kubernetes Service (AKS)," z.d Augustus 2018. [Online]. Available: https://searchcloudcomputing.techtarget.com/definition/Azure-Kubernetes-Service-AKS. [Geopend 2 april 2020].

[8] Red Hat Openshift, „What is OpenShift," 8 mei 2019. [Online]. Available: https://www.openshift.com/learn/what-is-openshift. [Geopend 3 april 2020].

[9] Microsoft Azure, „Azure IaaS (infrastructuur als een dienst)," z.d. -b. [Online]. Available: https://azure.microsoft.com/nl-nl/overview/what-is-azure/iaas/. [Geopend 3 april 2020].

[10] opensource.com, „opensource.com," z.d. z.d. z.d.. [Online]. Available: https://opensource.com/resources/what-docker. [Geopend 3 april 2020].

[11] Microsoft, „eShopOnContainers," 25 Februari 2020. [Online]. Available: https://github.com/dotnet-architecture/eShopOnContainers. [Geopend 2 April 2020].

[12] jurian, „Public Cloud - Microsoft Azure," handson, z.d.. [Online]. Available: https://www.doherty.co.uk/blog/azure-vs-on-premise . [Geopend 4 juni 2020].

[13] Doherty, „Azure vs on-premise: Why you should choose the cloud," doherty, z.d. [Online]. Available: https://www.doherty.co.uk/blog/azure-vs-on-premise. [Geopend 4 juni 2020].

[14] D. Patterson, „Why Microsoft spends over $1 billion on cybersecurity each year," techrepublic, 14 Februari 2018. [Online]. Available: https://www.techrepublic.com/article/why-microsoft-spends-over-1-billion-on-cybersecurity-each-year/. [Geopend 4 juni 2020].

[15] Microsoft Azure, „SLA for Azure Kubernetes Service (AKS)," juni 2018. [Online]. Available: https://azure.microsoft.com/en-us/support/legal/sla/kubernetes-service/v1_0/. [Geopend 7 april 2020].

[16] Microsoft Azure, „SLA voor Virtual Machines," januari 2020. [Online]. Available: https://azure.microsoft.com/nl-nl/support/legal/sla/virtual-machines/v1_9/. [Geopend 7 april 2020].

[17] uptime, „Uptime and downtime with 99.5 % SLA," z.d.. [Online]. Available: https://uptime.is/99.5. [Geopend 7 april 2020].

[18] Microsoft Azure, „Aanbevolen procedures voor bedrijfscontinuïteit en disaster recovery in Azure Kubernetes Service (AKS)," 28 november 2018. [Online]. Available: https://docs.microsoft.com/nl-nl/azure/aks/operator-best-practices-multi-region. [Geopend 7 april 2020].

[19] Microsoft Azure, „Meerdere knooppuntgroepen voor een cluster maken en beheren in Azure Kubernetes Service (AKS)," 10 Maart 2020. [Online]. Available: https://docs.microsoft.com/nl-nl/azure/aks/use-multiple-node-pools. [Geopend 7 april 2020].

[20] kubernetes, „Horizontal Pod Autoscaler," 28 maart 2020. [Online]. Available: https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/. [Geopend 7 april 2020].

[21] Kubernetes, „Securing a Cluster," 28 oktober 2019. [Online]. Available: https://kubernetes.io/docs/tasks/administer-cluster/securing-a-cluster/. [Geopend 7 april 2020].

[22] Kubernetes, „Secrets," 26 Maart 2020. [Online]. [Geopend 7 april 2020].

[23] Microsoft Azure, „Beveiligingsconcepten voor toepassingen en clusters in Azure Kubernetes Service (AKS)," 1 maart 2019. [Online]. Available: https://docs.microsoft.com/nl-nl/azure/aks/concepts-security.

[24] Microsoft Azure, „Prijscalculator," z.d. -d. [Online]. Available: https://azure.microsoft.com/nl-nl/pricing/calculator/. [Geopend 6 april 2020].

[25] A. Diwan, „Kubernetes: Twelve Key Features," 14 december 2017. [Online]. Available: https://medium.com/@abhaydiwan/kubernetes-introduction-and-twelve-key-features-cdfe8a1f2d21. [Geopend 7 april 2020].

[26] Microsoft Azure, „SLA for Azure Red Hat OpenShift," Microsoft Azure, mei 2019. [Online]. Available: https://azure.microsoft.com/en-us/support/legal/sla/openshift/v1_0/. [Geopend 9 april 2020].

[27] uptime, „Uptime and downtime with 99.9 % SLA," uptime, z.d.. [Online]. Available: https://uptime.is/99.9. [Geopend 2020 april 9].

[28] Microsoft Azure, „Veelgestelde vragen over Azure Red Hat OpenShift," Microsoft Azure, 4 november 2019. [Online]. Available: https://docs.microsoft.com/nl-nl/azure/openshift/openshift-faq. [Geopend 9 april 2020].

[29] Microsoft Azure, „Een Azure Red Hat OpenShift-cluster schalen," Microsoft Azure, 6 mei 2019. [Online]. Available: https://docs.microsoft.com/nl-nl/azure/openshift/tutorial-scale-cluster. [Geopend 9 april 2020].

[30] Red hat OpenShift, „Automatically scaling pods," Red hat OpenShift, z.d.. [Online]. Available: https://docs.openshift.com/container-platform/4.3/nodes/pods/nodes-pods-autoscaling.html. [Geopend 9 april 2020].

[31] Red Hat OpenShift, „Security information," Red Hat OpenShift, z.d.. [Online]. Available: https://www.openshift.com/policy/security/. [Geopend 9 april 2020].

[32] Microsoft Azure, „Azure Red Hat OpenShift pricing," Microsoft Azure, z.d.. [Online]. Available: https://azure.microsoft.com/en-us/pricing/details/openshift/. [Geopend 9 april 2020].

[33] N. Jain, „What are the New Features of OpenShift 4?," WHIZLABS, 11 oktober 2019. [Online]. Available: https://www.whizlabs.com/blog/openshift-4-new-features/. [Geopend 9 april 2020].

[34] Uptime , „Uptime and downtime with 99.99 % SLA," Uptime , z.d.. [Online]. Available: https://uptime.is/99.99. [Geopend 10 april 2020].

[35] Microsoft Azure, „Een back-up maken van een Azure VM vanuit de VM-instellingen," Microsoft Azure, 13 juni 2019. [Online]. Available: https://docs.microsoft.com/nl-nl/azure/backup/backup-azure-vms-first-look-arm. [Geopend 11 april 2020].

[36] Microsoft Azure, „Een overzicht van Azure VM-back-up," Microsoft Azure, 13 september 2019. [Online]. Available: https://docs.microsoft.com/nl-nl/azure/backup/backup-azure-vms-introduction. [Geopend 11 april 2020].

[37] Microsoft Azure, „Schaalsets voor virtuele machines," Microsoft Azure, z.d.. [Online]. Available: https://azure.microsoft.com/nl-nl/services/virtual-machine-scale-sets/. [Geopend 14 april 2020].

[38] Microsoft Azure, „Wat is Azure Load Balancer?," Microsoft Azure, 14 januari 2020. [Online]. Available: https://docs.microsoft.com/nl-nl/azure/load-balancer/load-balancer-overview. [Geopend 14 april 2020].

[39] Docker, „Swarm mode overview," Docker, z.d.. [Online]. Available: https://docs.docker.com/engine/swarm/. [Geopend 14 april 2020].

[40] Docker Forum, „Autoscaling in docker swarm," Docker Forum, Januari 2018. [Online]. Available: https://forums.docker.com/t/autoscaling-in-docker-swarm/44353. [Geopend 14 april 2020].

[41] Docker, „Orchestration," Docker, z.d.. [Online]. Available: https://docs.docker.com/get-started/orchestration/. [Geopend 14 april 2020].

[42] Microsoft Azure, „Overzicht van azure Virtual Machines-beveiliging," Microsoft Azure, 2 oktober 2019. [Online]. Available: https://docs.microsoft.com/nl-nl/azure/security/fundamentals/virtual-machines-overview. [Geopend 14 april 2020].

[43] Paul Schnackenburg, „Tips & Tricks To Secure and Optimize Azure VMs," virtualizationreview, 09 mei 2018. [Online]. Available: https://virtualizationreview.com/Articles/2018/09/05/Tips-and-Tricks-To-Secure-and-Optimize-Azure-VMs.aspx?Page=2. [Geopend 14 april 2020].

[44] Docker , „Docker overview," Docker , z.d.. [Online]. Available: https://docs.docker.com/get-started/overview/. [Geopend 14 april 2020].

[45] Microsoft Azure, „What is Azure Resource Manager?," Microsoft Azure, 25 maart 2020. [Online]. Available: https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/overview. [Geopend 16 april 2020].

[46] Microsoft Azure, „What are ARM templates?," Microsoft Azure, 6 april 2020. [Online]. Available: https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/overview. [Geopend 14 april 2020].

[47] J. Cool, „Introducing Azure DevOps," Microsoft Azure, 10 september 2018. [Online]. Available: https://azure.microsoft.com/nl-nl/blog/introducing-azure-devops/. [Geopend 7 mei 2020].

[48] A. Charot, „Continuously deploy ARM Templates using Azure DevOps," medium, 25 februari 2019. [Online]. Available: https://medium.com/faun/continuously-deploy-arm-templates-using-azure-devops-1855513c7d56. [Geopend 2020 mei 7].

[49] uptime, „uptime," z.d.. [Online]. Available: https://uptime.is/. [Geopend 11 mei 2020].

[50] Microsoft Azure, „eserveringen," Microsoft Azure, z.d.. [Online]. Available: https://azure.microsoft.com/nl-nl/reservations/. [Geopend 15 april 2020].

[51] S. Scherer, „dockersamples/Node-bulletin-board," Github, 3 februari 2020. [Online]. Available: https://github.com/dockersamples/node-bulletin-board/tree/master/bulletin-board-app. [Geopend 3 mei 2020].

[52] S. Danielson, „Azure Pipelines agents," 20 November 2019. [Online]. Available: https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/agents?view=azure-devops&tabs=browser. [Geopend 1 December 2019].

[53] Microsoft Azure, „Azure Repos," Microsoft Azure, z.d.. [Online]. Available: https://azure.microsoft.com/en-us/services/devops/repos/. [Geopend 2020 mei 26].

[54] K. Schollaart, „ARM outputs," Visual Studio Marketplace, 3 maart 2019. [Online]. Available: https://marketplace.visualstudio.com/items?itemName=keesschollaart.arm-outputs. [Geopend 10 mei 2020].

[55] Microsoft Build, „Service connections," Microsoft Build, 20 mei 2020. [Online]. Available: https://docs.microsoft.com/en-us/azure/devops/pipelines/library/service-endpoints?view=azure-devops&tabs=yaml. [Geopend 26 mei 2020].

[56] stackoverflow, „VSTS create service connection via API," stackoverflow, 15 september 2018. [Online]. Available: https://stackoverflow.com/questions/52335657/vsts-create-service-connection-via-api. [Geopend 26 mei 2020].

[57] M. Developer, Regisseur, *Getting started with CI/CD & Azure Container Service (AKS) powered by VSTS.* [Film]. 2018.

[58] Azuredevopslabs, „Create a Docker Container Registry in Azure," 28 November 2019. [Online]. Available: https://azuredevopslabs.com/labs/java/builddocker/. [Geopend 5 December 2019].

[59] kubernetes, „Deployments," kubernetes, 18 maart 2020. [Online]. Available: https://kubernetes.io/docs/concepts/workloads/controllers/deployment/. [Geopend 26 mei 2020].

[60] K. Marttila, „Comparing Azure ARM and Terraform," Medium, 20 maart 2019. [Online]. Available: https://medium.com/@kari.marttila/comparing-azure-arm-and-terraform-d2b38975c7ea. [Geopend 29 mei 2020].

[61] H. Oberink, „5W1H methode," z.d.. [Online]. Available: https://okam.nl/5w1h-methode/. [Geopend 17 Februari 2020].

[62] L. interview, „Laddering interview,” 21 Januari 2017. [Online]. Available: https://communicatiekc.com/2017/01/21/laddering-interview/. [Geopend 11 Maart 2020].

[63] K. Schermer, „De effectieve projectgroep,” November 2015. [Online]. Available: http://hoadd.noordhoff.nl/sites/7864/_assets/7864d19.pdf. [Geopend 2 Februari 2020].

[64] H. Smals, „Triangulatie - Communicatie KC,” z.d.. [Online]. Available: https://communicatiekc.com/triangulatie/. [Geopend 17 Februari 2020].

[65] „Soorten interviews,” 8 september 2015. [Online]. Available: https://www.scribbr.nl/onderzoeksmethoden/soorten-interviews/. [Geopend 2 Februari 2020].

[66] R. Pijlman en M. v. Oosterhout, „SMART doelen stellen,” z.d. [Online]. Available: http://www.carrieretijger.nl/functioneren/management/leidinggeven/doelen-stellen/smart. [Geopend 18 februari 2020].

[67] Marketingtermen.nl, „MoSCoW-methode,” z.d.. [Online]. Available: https://www.marketingtermen.nl/begrip/moscow-methode. [Geopend 18 Februari 2020].

[68] E. Schuurmans, „In 5 stappen een praktische stakeholderanalyse,” 2016 Augustus 29. [Online]. Available: https://www.leanenkwaliteit.nl/blog/5-stappen-praktische-stakeholderanalyse/. [Geopend 20 Febuari 2020].

[69] T. Morphy, „Stakeholder-matrix,” 2 Januari 2010. [Online]. Available: https://www.stakeholdermap.com/stakeholder-matrix.html. [Geopend 20 Februari 2020].

[70] EenBlogjeOm, „Functionele vs. niet-functionele requirements,” 29 December 2017. [Online]. Available: http://www.raamstijn.nl/eenblogjeom/index.php/requirements/569-functionele-requirements-vs-niet-functoinele-requirements.

[71] „Micro meso macro - managementmodellensite,” z.d. z.d. z.d.. [Online]. Available: https://managementmodellensite.nl/micro-meso-macro/#.XmYlMXOg-Ul. [Geopend 09 Maart 2020].

[72] A. Doelen en A. Weber, Organiseren en managen, Het 7S model toegepast, Groningen: Noordhoff Uitgevers, 2010, p. 250.

[73] Business unit manager, Interviewee, *Interview contextanalyse.* [Interview]. 12 Maart 2020.

[74] Hogeschool Rotterdam, „Handleiding zoekstrategie deskresearch,” z.d.. [Online]. Available: https://www.hogeschoolrotterdam.nl/contentassets/097de8020b354103a95d701f5b947b18/zoekstrategieformulier-uitgebreid-v2.2.pdf. [Geopend 6 April 2020].

[75] Rijksuniversiteit groningen, „Informatievaardigheden: Zoekmethoden,” 17 September 2019. [Online]. Available: https://libguides.rug.nl/c.php?g=531668&p=3637472. [Geopend 17 Maart 2020].

[76] A. Belo, „Moving from VMs to Containers,” 16 mei 2019. [Online]. Available: https://www.armory.io/blog/moving-from-vms-to-containers/. [Geopend 25 Maart 2020].

[77] S. McCarty, „So, What Does a Container Engine Really Do Anyway?,” 13 juli 13. [Online]. Available: http://crunchtools.com/so-what-does-a-container-engine-really-do-anyway/. [Geopend 26 Maart 2020].

[78] focuson-it, „Uptime is niet hetzelfde als beschikbaarheid,” 13 december 2018. [Online]. Available: https://www.focuson-it.nl/uptime-paessler/. [Geopend 6 april 2020].

[79] M. Rouse, „Microsoft SQL Server,” juni 2019. [Online]. Available: https://searchsqlserver.techtarget.com/definition/SQL-Server. [Geopend 3 april 2020].

[80] Microsoft Azure, „Azure SQL Database,” z.d. -c. [Online]. Available: https://azure.microsoft.com/nl-nl/services/sql-database/#product-overview.

[81] Microsoft Azure, „Prijzen van Azure SQL Database,” z.d. -e. [Online]. Available: https://azure.microsoft.com/nl-nl/pricing/details/sql-database/managed/. [Geopend 6 april 2020].

[82] Microsoft Azure, „SLA voor Azure SQL Database,” juli 2019 . [Online]. Available: https://azure.microsoft.com/nl-nl/support/legal/sla/sql-database/v1_4/. [Geopend 6 april 2020].

[83] Microsoft Azure, „Automatische back-ups,” 13 december 2019. [Online]. Available: https://docs.microsoft.com/nl-nl/azure/sql-database/sql-database-automated-backups?tabs=single-database. [Geopend 6 april 2020].

[84] Microsoft Azure, „Hyperscale-servicelaag,” 1 oktober 2019. [Online]. Available: https://docs.microsoft.com/nl-nl/azure/sql-database/sql-database-service-tier-hyperscale. [Geopend 6 april 2020].

[85] Microsoft Azure, „Een overzicht van azure SQL Database-beveiligingsmogelijkheden,” 14 mei 2019. [Online]. Available: https://docs.microsoft.com/nl-nl/azure/sql-database/sql-database-security-overview#security-management. [Geopend 6 april 2020].

[86] M. VARNAS, „Azure SQL Managed Instances – Pros and Cons,” z.d.. [Online]. Available: https://red9.com/pros-cons-azure-sql-managed-instances/. [Geopend 6 april 2020].

[87] L. Shiff en W. Rowe, „bmc,” 5 maart 2018. [Online]. Available: https://www.bmc.com/blogs/sql-vs-nosql/. [Geopend 3 april 2020].

[88] Y. Perry, „Azure Availability Zones: An In-Depth Look,” 5 september 2019. [Online]. Available: https://cloud.netapp.com/blog/azure-availability-zones-an-in-depth-look. [Geopend 6 april 2020].

[89] Microsoft Azure, „SLA voor Azure Cosmos DB,” mei 2019. [Online]. Available: https://azure.microsoft.com/nl-nl/support/legal/sla/cosmos-db/v1_3/. [Geopend 6 april 2020].

[90] MongoDB, „Microsoft Azure,” z.d. -a. [Online]. Available: https://docs.atlas.mongodb.com/reference/microsoft-azure/. [Geopend 6 april 2020].

[91] Microsoft Azure, „Online back-up en on-demand gegevensherstel in Azure Cosmos DB,” 21 mei 2019. [Online]. Available: https://docs.microsoft.com/nl-nl/azure/cosmos-db/online-backup-and-restore. [Geopend 6 april 2020].

[92] Microsoft Azure, „Beveiliging in Azure Cosmos DB - overzicht,” 10 maart 2020. [Online]. Available: https://docs.microsoft.com/nl-nl/azure/cosmos-db/database-security.

[93] mongodb, „Deploy MongoDB on Microsoft Azure,” z.d. -a. [Online]. Available: https://www.mongodb.com/cloud/atlas/azure-mongodb. [Geopend 6 april 2020].

[94] S. Jaybhay, „Features of MongoDB,” 6 augustus 2019. [Online]. Available: https://medium.com/@expertwithsagarjaybhay/features-of-mongodb-fb8900b743c4. [Geopend 6 april 2020].

[95] Microsoft Azure, „Azure Cosmos DB,” z.d. -a. [Online]. Available: https://azure.microsoft.com/nl-nl/services/cosmos-db/#featured. [Geopend 6 April 2020].

[96] Richard Cookson , „The Pros and Cons of MongoDB,” 8 augustus 2019. [Online]. Available: https://www.virtual-dba.com/pros-and-cons-of-mongodb/.

[97] trivento, „trivento,” z.d.. [Online]. Available: https://trivento.nl/apis-ontraadseld-en-hoe-werken-apis/. [Geopend 2 April 2020].

[98] Microsoft Azure, „SLA voor API Management,” december 2019. [Online]. Available: https://azure.microsoft.com/nl-nl/support/legal/sla/api-management/v1_2/. [Geopend 6 april 2020].

[99] Microsoft Azure, „Noodherstel implementeren met back-up en herstellen van services in Azure API Management,” 3 februari 2020. [Online]. Available: https://docs.microsoft.com/nl-nl/azure/api-management/api-management-howto-disaster-recovery-backup-restore. [Geopend 6 april 2020].

[100] Microsoft Azure, „Beveiligingsbesturingselementen voor API-beheer,” 23 september 2019. [Online]. Available: https://docs.microsoft.com/nl-nl/azure/api-management/api-management-security-controls. [Geopend 6 april 2020].

[101] Microsoft Azure, „API Management,” z.d. -g. [Online]. Available: https://azure.microsoft.com/nl-nl/services/api-management/#features. [Geopend 6 april 2020].

[102] Microsoft Azure, „Service Bus,” z.d -h. [Online]. Available: https://azure.microsoft.com/nl-nl/services/service-bus/. [Geopend 6 april 2020].

[103] „Messaging that just works,” z.d.. [Online]. Available: https://www.rabbitmq.com/. [Geopend 6 april 2020].

[104] M. Faizan, „Azure Service Bus vs RabbitMQ for Enterprise applications,” juni 2018. [Online]. Available: https://stackoverflow.com/questions/43913747/azure-service-bus-vs-rabbitmq-for-enterprise-applications. [Geopend 6 april 2020].

[105] M. Eelants, „7S MODEL MCKINSEY,” z.d. z.d. z.d.. [Online]. Available: https://www.marketingmodellen.com/7s-model-van-mckinsey/. [Geopend 11 Maart 2020].

[106] managementmodellensite.nl, „Checklist 7S-model,” z.d. z.d. z.d.. [Online]. Available: https://managementmodellensite.nl/7s-model-checklist/#.Xmn133Og-Uk. [Geopend 11 Maart 2020].

[107] E. hageraats, H. K en S. V. Linden, „Reader Ontwerponderzoek V2,” Saxion, Enschede, 2016.

[108] Van Dale, „Van Dale,” Van Dale, z.d.. [Online]. Available: https://www.vandale.nl/gratis-woordenboek/nederlands/betekenis/trial-and-error. [Geopend 12 mei 2020].

[109] B. Janse, „Multicriteria-analyse (MCA),” 2018. [Online]. Available: https://www.toolshero.nl/besluitvorming/multicriteria-analyse-mca/. [Geopend 17 Maart 2020].

[110] Wikipedia, „Unified Modeling Language,” Wikipedia, 11 mei 2020. [Online]. Available: https://nl.wikipedia.org/wiki/Unified_Modeling_Language. [Geopend 12 mei 2020].

[111] M. Rouse, „Azure Container Instances,” z.d. november 2018. [Online]. Available: https://searchcloudcomputing.techtarget.com/definition/Azure-Container-Instances. [Geopend 2 april 2020].

[112] M. Rouse, „workflow,” techtarget, oktober 2014. [Online]. Available: https://searchcio.techtarget.com/definition/workflow. [Geopend 16 april 2020].

[113] Microsoft Azure, „SLA for Azure Kubernetes Service (AKS),” maart 2020. [Online]. Available: https://azure.microsoft.com/en-us/support/legal/sla/kubernetes-service/v1_1/. [Geopend 7 april 2020].

[114] uptime, „Uptime and downtime with 99.95 % SLA,” z.d.. [Online]. Available: https://uptime.is/99.95. [Geopend 7 april 2020].

14. Versiebeheer

Versie	Datum	Aanpassingen
0.1	11-02-2020	Opzet document
0.2	23-03-2020	Invoegen methode en resultaten probleem- en behoefteanalyse
0.3	21-04-2020	Invoegen methode en resultaten contextanalyse en literatuuronderzoek
0.4	06-05-2020	Aanpassen en uitbreiden resultaten literatuuronderzoek
0.5	13-05-2020	Invoegen methode en resultaten ontwerpdocumentatie
0.6	19-05-2020	Aanpassen en uitbreiden ontwerpdocumentatie
0.7	25-05-2020	Feedback stakeholders ilionx en Tonny Lievers invoegen
0.8	29-05-2020	Verwerken feedback Invoegen samenvatting, inleiding, evaluatie, discussie en nawoord
0.9	31-05-2020	Controle spelling en grammatica
1.0	02-06-2020	Inleveren conceptversie
1.1	04-06-2020	Verbeteren feedback concept versie
1.2	08-06-2020	Inleveren conceptversie ilionx
1.3	09-06-2020	Verbeteren feedback stagebegeleider ilionx
1.4	10-06-2020	Verbeteren feedback business unit manager
1.5	11-06-2020	Aanpassen details technisch ontwerp
2.0	12-06-2020	Definitief document

Tabel 19: Versiebeheer

15. Bijlage: het vooronderzoek

In het vooronderzoek zullen verschillende zaken verhelderd worden. Zo wordt in de probleemanalyse (hoofdstuk 15.1) gekeken naar het probleem van de opdrachtgever, waardoor duidelijk zal worden wat het probleem is van de opdrachtgever. Vervolgens worden in de behoefteanalyse (hoofdstuk 15.2) de verschillende stakeholders geïdentificeerd en wordt hiernaast inzichtelijk gemaakt wat de eisen en wensen zijn van deze stakeholders. Door middel van de contextanalyse (hoofdstuk 15.3) wordt daarna duidelijk in welke omgeving de uiteindelijke oplossing wordt gebruikt en aan wat voor voorwaarden deze moet voldoen om een kans van slagen te hebben. Ten slotte wordt het literatuuronderzoek (hoofdstuk 15.4) uitgevoerd over dit onderwerp.

15.1 Probleemanalyse

In de probleemanalyse wordt het probleem van de opdrachtgever concreet gemaakt. Dit wordt gedaan zodat het duidelijk wordt of de vraag van het bedrijf door middel van het onderzoek kan worden opgelost. Hiernaast wordt gekeken of deze vraag geschikt is om via het ontwerponderzoek opgelost te worden.

In de probleemanalyse worden verschillende methodes aangehaald. Deze methodes worden gebruikt om gegevens te verzamelen, resultaten te noteren en analyses uit te voeren. Door het probleem vanuit meerdere perspectieven te bekijken, wordt het duidelijker wat het onderliggende probleem van de onderzoeksvraag is. Aan de hand van deze verschillende methodes moet ook duidelijk worden of de vooraf gedefinieerde hoofd- en deelvragen bij dit onderzoek passen. De hoofd- en deelvragen worden hieronder ook verder uitgewerkt. Verder wordt in deze probleemanalyse duidelijk wat de business requirements van de interventies zijn, en of een specificatie van wat de oplossing is bewerkstelligd moet worden.

15.1.1 Methode

Er wordt van verschillende methodes gebruik gemaakt om zo de verschillende gegevens te verzamelen die nodig zijn om te controleren of de juiste hoofd- en deelvragen zijn opgesteld. Voor een valide- en betrouwbaar onderzoek wordt alleen gebruik gemaakt van bewezen methodes.

5W1H methode
Om een duidelijke probleemvorming te kunnen genereren, moeten een aantal vragen worden gesteld. Dit wordt in dit onderzoek gedaan door de 5W1H [61] met hierbij laddering [62] toe te passen. De methode 5W1H is gebaseerd op vijf W-vragen (Wie, Wat, Waar, Wanneer en Waarom) en één H-vraag (Hoe). De laddering wordt toegepast door steeds de vraag “Waarom?” te stellen wanneer een antwoord wordt verschaft over een bepaald onderwerp.
Aan de hand van de antwoorden van de business unit manager en lead consultant van ilionx op deze vragen, kunnen de hoofd- en deelvragen gecontroleerd worden op juistheid en hierna beantwoord worden aan de hand van dit onderzoek.

Causaal veldmodel
Het probleem heeft te maken met verschillende invloeden. Om deze in kaart te brengen wordt er gebruik gemaakt van een causaal model [63]. In een causaal model worden verschillende oorzaken en gevolgen met elkaar in verband gebracht. Door het gebruik van dit model wordt het probleem verhelderd.

Triangulatie
Om niet maar vanuit één standpunt naar het probleem te kijken, wordt er gebruik gemaakt van triangulatie [64]. Bij triangulatie wordt vanuit meerdere standpunten naar het probleem gekeken. Hierbij is het mogelijk dat de verschillende personen het probleem verschillend van elkaar beschrijven. Door het gebruik van triangulatie wordt de validiteit van dit onderzoek vergroot.

Interviewmethode
In de probleemanalyse worden verschillende personen geïnterviewd om een breed en compleet beeld te krijgen van het probleem. De interviews worden afgenomen door middel van de semigestructureerde interviewmethode [65]. Deze methode voorziet kwalitatief onderzoek door af te wisselen tussen de vooropgestelde vragen en de tijdens het interview nieuw bedachte vragen.

SMART methode
Deze methode wordt gebruikt om behoeftes (eisen en wensen) zo concreet mogelijk te formuleren. Dit wordt gedaan door te kijken of de behoefte Specifiek, Meetbaar, Acceptabel, Realistisch en Tijdsgebonden is. Door requirements op deze manier te formuleren, kan achteraf gemakkelijk bepaald worden of de requirement is opgenomen in de gerealiseerde oplossing [66].

MoSCoW methode
De MoSCoW methode wordt gebruikt om de verschillende requirements te prioriteren. MoSCoW kent de volgende prioriteringen: Must-have (vereist), Should-have (hoge prioriteit), Could (mocht er tijd over zijn) of Would (geen prioriteit, toekomst). Door het toekennen van deze prioriteringen aan de requirements, zorgt het voor duidelijkheid van de eisen van de opdrachtgever. Op deze manier kan sneller worden gewerkt, namelijk door de minder belangrijkere requirements achterwege te laten [67].

Requirements traceability
Door middel van requirements traceability worden requirements gekoppeld aan de verschillende stakeholders. Hierdoor kan achteraf worden gekeken of er voldaan is aan de eisen en wensen van de verschillende stakeholders. Ook kan zo gecontroleerd worden op tegenstijdigheden van requirements door de verschillende stakeholders.

15.1.2 Resultaten
De onderzoeksresultaten worden in dit hoofdstuk uitgewerkt. Deze resultaten worden in de conclusie gebruikt om te controleren of de opgestelde hoofd- en deelvragen valide zijn en om de business requirements te formuleren.

15.1.2.1 Perspectieven probleemstelling
De opgestelde vragen uit de 5W1H methode zijn uitgewerkt in een interview, die te vinden is in hoofdstuk 15.5.1.1. Dit interview is afgenomen met de business unit manager en een lead consultant. Uit overleg met de begeleider kwam namelijk naar voren dat voornamelijk de business unit manager en de lead consultant met het probleem te maken hebben. Een volledige stakeholderanalyse wordt uitgewerkt in de behoefteanalyse. Bij het interview met de business unit manager, ook de opdrachtgever, worden de grote lijnen van het probleem duidelijk. Bij het interview met de lead consultant wordt specifieker gekeken naar het probleem die de klant ervaart.

15.1.2.1.1 Perspectief business unit manager

Uit het perspectief van de business unit manager werd duidelijk dat het probleem is ontstaan door het gebrek aan kennis over containers. De business unit manager denkt dat aan de hand van een onderzoek naar automated containerization, een oplossing kan worden gevonden voor het gebrek aan kennis hiervan binnen de unit GRO-Cloud Solutions. Het onderzoek moet zich richten op het gebruik van container gebaseerde applicaties binnen Microsoft Azure. De business unit manager gaf in het interview aan dat de getrachte oplossing moet worden gevonden in een whitepaper, maar mocht tijdens het onderzoek wat anders naar voren komen mag dit ook worden voorgesteld. De oplossing moet inzicht geven over wat container gebaseerde applicaties zijn, wat het oplevert voor de klant en welke oplossingen er zijn voor de hosting van deze containers. Het automated deel in het onderzoek moet worden uitgewerkt in workflows. Workflows vormen de basis voor het automatisch kunnen uitrollen van infrastructuren of applicaties binnen een Cloud omgeving.

Het bovenstaande perspectief is gebaseerd op de uitwerking van het 5W1H model, welke terug te vinden is in hoofdstuk 15.5.1.1.1. Het interview is afgenomen op 24 februari 2020.

15.1.2.1.2 Perspectief lead consultant

Het perspectief van de lead consultant op het vraagstuk is dat er op dit moment onvoldoende kennis is over het gebruik van containers binnen de unit. Hierdoor kan de unit GRO-Cloud Solutions niet goed uitspraken doen over wat de beste oplossingen zijn voor het gebruik van containers. Door het doen van onderzoek naar deze containers, moet naar voren komen wat de beste oplossingen hiervoor zijn. Naast dat het kennisniveau binnen de ilionx aan de hand van dit onderzoek omhoog zou moeten gaan, moet ook een whitepaper ontwikkeld worden. Deze whitepaper moet aan de klant inzicht geven over wat containers voor hun organisatie kan betekenen. De klant kan de whitepaper als leidraad gebruiken om zo tot een passende oplossing te komen voor hun container gebaseerde applicatie. In de whitepaper moet rekening worden gehouden met beschikbaarheid, betrouwbaarheid, schaalbaarheid, veiligheid en de kosten van de oplossing. Verder moet voor de unit workflows worden gemaakt. Deze workflows vormen de basis van een container infrastructuur welke automatisch kan worden uitgerold door middel van een automatization tool.

Het bovenstaande perspectief is gebaseerd op de uitwerking van het 5W1H model, welke terug te vinden is in hoofdstuk 15.5.1.1.2. Het interview is afgenomen op 27 februari 2020.

15.1.2.2 Oorzaken van het probleem

Het vraagstuk is nu bekeken vanuit verschillende perspectieven. Hier worden de verschillende invalshoeken van waaruit het vraagstuk kan worden bekeken, om de oorzaak van het vraagstuk te verduidelijken, uitgewerkt. Daarnaast zijn de oorzaken van het vraagstuk gebaseerd op interviews met de business unit manager en lead consultant, welke terug te vinden zijn in hoofdstuk 15.5.1.1.1 en 15.5.1.1.2.

Het vraagstuk is naar voren gekomen doordat een klant van ilionx vragen had over een e-commerce applicaties die ilionx gaat hosten. De applicatie draaide namelijk in een container. Ilionx had nog niet veel met container gebaseerde applicaties gewerkt. Op dat moment moest heel urgent onderzocht worden hoe de vraag van de klant het best kon worden beantwoord om zo de klant zo goed mogelijk te voorzien in zijn behoefte. Hier was het bedrijf niet goed op voorbereid. Om voortaan goed voorbereid te zijn op vergelijkbare vragen van klanten, wil ilionx goed onderzocht en vastgelegd hebben wat de beste oplossingen voor container gebaseerde applicaties zijn.

Hiernaast wordt het ook steeds meer een trend om gebruik te maken van container gebaseerde applicaties. Om goed voorbereid te zijn op mogelijke andere vragen van de klanten en om klanten goed voor te lichten, wordt aan de hand van dit onderzoek een whitepaper gemaakt.

Het vraagstuk van ilionx omvat ook dat er workflows moeten worden gemaakt om geautomatiseerd de verschillende mogelijke infrastructuren uit te rollen. Dit deel van het vraagstuk komt voort uit het feit dat ilionx goed voorbereid wil zijn op mogelijke vragen van klanten. Een workflow is een blueprint van de infrastructuur waardoor de infrastructuur snel kan worden gerealiseerd.

15.1.2.2.1 Causaal veldmodel

Om de verschillende invloeden op het vraagstuk duidelijk in beeld te brengen, is er een causaal veldmodel gemaakt. In dit model zijn de oorzaken en gevolgen aan het vraagstuk gekoppeld. In Figuur 2 staat een duidelijk overzicht van de verschillende oorzaken en gevolgen van het vraagstuk. Het model is gebaseerd op de verschillende perspectieven die staan beschreven in hoofdstuk 15.1.2.1 en oorzaken in hoofdstuk 15.1.2.2.

15.1.3 Conclusie

Aan de hand van interviews met de business unit manager en de lead consultant van ilionx zijn de onderstaande hoofd- en deelvragen opgesteld. Hierbij zijn ook de business requirements opgesteld aan de hand van de resultaten van de 5W1H methode.

15.1.3.1 Hoofd- en deelvragen

De resultaten van de probleemanalyse zijn ter onderbouwing van de hoofd- en deelvragen. Naast de hoofdvraag in Tabel 1, zijn de verschillende deelvragen in Tabel 2 opgesteld. Elke deelvraag beantwoordt een stukje van de hoofdvraag, waardoor deze zo volledig mogelijk beantwoord kan worden.

15.1.3.2 Business requirements

De eerder beschreven oorzaken van het vraagstuk en de hieruit geformuleerde hoofd- en deelvragen, zijn in Tabel 3 omgezet naar business requirements. De onderstaande requirements zijn opgesteld aan de hand van gehouden interviews en de resultaten van het 5W1H model (15.5.1.1.1 en 15.5.1.1.2).

15.2 Behoefteanalyse

In de behoefteanalyse wordt inzichtelijk gemaakt wie de stakeholders (belanghebbenden) zijn bij het vraagstuk van ilionx. Hiernaast worden deze verschillende behoeftes van de stakeholders geïnventariseerd en geprioriteerd.

In de behoefteanalyse wordt gebruik gemaakt van verschillende methodes die worden gebruikt om gegevens te verzamelen, resultaten te noteren en analyses uit te voeren. Eerst worden door middel van deze methodes de stakeholders geïnventariseerd. Hierna zal tussen alle stakeholders een selectie plaatsvinden voor de diegenen die van belang zijn voor het vraagstuk. De volgende stap is het bepalen van de behoeften van die geselecteerde stakeholders. De behoeften van deze stakeholders vormen de eisen en wensen van het vraagstuk. Deze eisen en wensen worden omgezet naar requirements en worden vervolgens geprioriteerd.

15.2.1 Methode

Om de verschillende gegevens te verzamelen die nodig zijn om de behoeften van de verschillende stakeholders te bepalen, wordt gebruik gemaakt van verschillende methodes. Voor een valide- en betrouwbaar onderzoek wordt alleen gebruik gemaakt van gevalideerde methoden. Het is belangrijk dat de gebruikte methodes aansluiten bij de specifieke toepassing.

Hieronder worden een aantal methoden opnieuw gebruikt die ook zijn gebruikt in de probleemanalyse. Dit zijn de SMART- en de MoSCoW methodes en requirements traceability. Voor uitleg van deze methodes, zie 15.1.1.

Stakeholderanalyse
De stakeholderanalyse geeft inzicht van alle stakeholders bij dit vraagstuk. In een stakeholderanalyse wordt gekeken naar wat de belangen van de stakeholders zijn en hoe deze gebruikt moeten worden bij het vraagstuk. De stakeholderanalyse bestaat uit meerdere stappen, welke hieronder zijn uitgelegd:

Stap 1: Identificeer de stakeholders
In de eerste stap van de stakeholderanalyse worden alle partijen in kaart gebracht die met de uiteindelijke interventie te maken krijgen. Dit is aan de hand van een interview door middel van het 5W1H model (met de business unit manager en lead consultant) in de probleemanalyse gedaan [68].

Stap 2: Bepalen invloed en belangen van de stakeholders
Nadat alle stakeholders bekend zijn, wordt onderzocht wat de invloeden en belangen van de stakeholders op de uiteindelijk interventie zijn. Dit wordt gedaan door middel van de influence- en interest stakeholdermatrix. Alle stakeholders worden binnen deze matrix geplaatst, zowel horizontaal (level of influence) als verticaal (level of interest). Verder heeft de matrix nog een diagonale indeling welke bestaat uit vier verschillende classificaties: key players, active consulation, maintain interest en keep informed [69].

Nadat stakeholderanalyse is uitgevoerd, is bekend welke stakeholders belang hebben bij de interventie en aan welke requirements de meeste waarde moet worden gehecht.

Interviews
Voor de behoefteanalyse zijn de business unit manager en lead consultant geïnterviewd. De interviews zijn afgenomen aan de hand van een semigestructureerde interviewmethode in combinatie met de 5W1H methode van de probleemanalyse (zie 15.1.1). Aan de hand van deze interviews zijn de stakeholders hun behoeften geïdentificeerd.

User- en systeem-requirements (functioneel en non-functioneel)
De verschillende behoeften van de stakeholders worden omgezet naar requirements en worden daarna weer verder onderverdeeld in user- en systeem-requirements. User-requirements worden omschreven als hetgeen wat de oplossing moet bieden. Systeem-requirements worden omschreven als het gewenste gedrag van het systeem [3].

De user- en system-requirements worden daarna weer verder ingedeeld als functioneel en non-functioneel is [70]:

- Functioneel is een “iets” wat het item moet doen (aangeduid met F);
- Non-functioneel is een kwaliteitseis aan het item (aangeduid met NF).

15.2.2 Resultaten

Zie hoofdstuk 8.3 en 8.4 voor de uitwerking van de stakeholderanalyse en hoofdstuk 8.4.1 voor de uitwerking van de requirements van de stakeholders.

15.3 Contextanalyse

In de contextanalyse wordt de specifieke context waarin de oplossingen van het vraagstuk geplaatst moet worden beschreven. Het gaat in de contextanalyse om randvoorwaarden die voortkomen uit de specifieke context. Dit zijn meestal financiële- of technische beperkingen. Hiernaast kunnen beperkingen ook voorkomen uit de doelstelling van een organisatie en/of door bijvoorbeeld culturele invloeden.

De eerste stap is om de verschillende aggregatieniveaus van de context uit te werken, waardoor een duidelijk beeld wordt geschetst van de huidige situatie. Hierna worden de randvoorwaarden opgesteld uit de gedefinieerde context.

15.3.1 Methode

Om de verschillende gegevens te verzamelen die nodig zijn om de randvoorwaarden van de opdracht te bepalen, wordt gebruik gemaakt van verschillende methoden. Voor een valide- en betrouwbaar onderzoek wordt alleen gebruik gemaakt van valide methoden.

Micro Meso Macro methode

Een veel gebruikte methode om de contextanalyse op te stellen is de Micro Meso Macro methode [71]. De methode geeft inzicht in de gehele organisatie en de omgeving. De drie dimensies bevinden zich allemaal in hun eigen omgeving. Deze verschillende aggregatieniveaus geven elke een andere kijken op de organisatie. Voor deze methode is gekozen omdat door het gebruik hiervan op verschillende manieren naar de organisatie kan worden gekeken. De omgevingen waarin gekeken worden zijn:

- Macroniveau (algemene omgeving);
- Mesoniveau (markt omgeving);
- Microniveau (directe omgeving).

Bij deze opdracht worden de drie niveaus beschreven met de volgende insteek:

- Macroniveau: klanten GRO-Cloud Solutions;
- Mesoniveau: ilionx Groningen;
- Microniveau: afdeling GRO-Cloud Solutions intern.

7s-model

Het 7s-model van McKinsey [72] geeft inzicht in hetgeen wat er binnen de organisatie afspeelt. Met dit model wordt de micro omgeving onderzocht. Het 7s-model is een interne analyse, hierdoor worden de zwakke- en sterke punten van GRO-Cloud Solutions bepaald. Het 7s model bevat zeven dimensies die allemaal in samenhang met elkaar staan, namelijk:

- Strategy (strategie);
- Structure (structuur);
- Systems (systemen);
- Staff (personeel);
- Style (managementstijl);
- Shared values (gemeenschappelijke waarden);
- Skills (sleutelvaardigheden).

Interviews
Door een interview met de business unit manager af te nemen, wordt duidelijk hoe op de afdeling gewerkt wordt. Aan de hand van de verkregen informatie uit dit interview kan het 7s-model worden uitgewerkt. Hierdoor wordt in kaart gebracht wat de externe partijen van GRO-Cloud Solutions en hun soort klanten zijn die uiteindelijk met de interventie gaan werken. Door het afnemen van het interview worden dus de macro-, micro- en meso omgevingen onderzocht.

Veldobservatie
Er wordt een observatie van het gedrag en van de werkwijze binnen GRO-Cloud Solutions gedaan. Door te observeren van hetgeen wat op de afdeling gebeurt, wordt duidelijk wat de verschillende werkwijzen van deze afdeling zijn. Hierdoor wordt de meso omgeving ook in kaart gebracht.

15.3.2 Resultaten
In dit hoofdstuk staan de uitwerkingen van de contextanalyse. Door het beschrijven van de resultaten wordt hiermee de context van de opdracht in kaart gebracht. Dit wordt vervolgens in 15.3.3 omgezet naar de randvoorwaarden.

15.3.2.1 Aspecten context
De oplossingen die het onderzoek zal bieden, zullen oplossingen zijn die zowel adviezen als technische inbreng geeft aan de unit GRO-Cloud Solutions.

Hierbij is het belangrijk om goed in kaart te brengen welke verschillende randvoorwaarden uit de contextanalyse komen. Dit is niet alleen gebaseerd op GRO-Cloud Solutions zelf, maar alle betrokkenen binnen de context zijn van de toepassing op de oplossing.

15.3.2.2 Aggregatieniveaus
Door het gebruik van verschillende aggregatieniveaus kan de context van het onderzoek worden gedefinieerd. In dit onderzoek wordt de context gedefinieerd met behulp van de onderstaande aggregatieniveaus:

- Macroniveau: hierbij wordt gekeken naar de klanten van de afdeling GRO-Cloud Solutions.
- Mesoniveau: hierbij wordt gekeken naar de externe partijen van de unit GRO-Cloud Solutions.
- Microniveau: hierbij wordt gedetailleerd gekeken naar de afdeling GRO-Cloud Solutions.

Aan de hand van de uitwerking van de bovenstaande aggregatieniveaus, wordt een duidelijk beeld gecreëerd van de omgeving waarin uiteindelijk de oplossing gebruikt gaat worden en aan welke randvoorwaarden hierbij voldaan moet worden.

15.3.2.2.1 Context – macroniveau
Om de context op macroniveau inzichtelijk te maken, heeft er een interview met de business unit manager plaatsgevonden. In dit interview is gefocust op wat voor type klanten GRO-Cloud Solutions heeft (de branche), hoeveel een klant van GRO-Cloud Solutions heeft in te brengen bij het ontwerp en realisatie van de infrastructuur en wat vanuit klanten de randvoorwaarden zijn aan de infrastructuur.

De klanten van GRO-Cloud Solutions zitten verspreid over heel Noord-Nederland (Friesland, Groningen, Drenthe en een deel van Overijssel), hierdoor is het van belang dat de whitepaper in het Nederlands wordt geschreven. GRO-Cloud Solutions focust zich niet op een bepaald soort branche en heeft dus klanten zoals overheidsinstellingen, zorginstellingen en banken.

GRO-Cloud Solutions heeft als doel de zorgen over IT weg te halen bij de klant. Mede hierdoor hoeft de klant niet veel in te brengen in de oplossingen. De business unit manager omschrijft het als volgt: "*Wat wij doen is het hele zorgstuk van IT wegnemen. De klant in de regierol en wij adviseren, implementeren en beheren en leveren "nachtrust". De klant hoeft zich niet druk te maken over IT, de klant moet zich alleen maar druk maken over zijn core business."* [73]

15.3.2.2.2 Context – mesoniveau

De context op mesoniveau wordt ook inzichtelijk gemaakt door middel van een interview met de business unit manager. Aan de hand van dit interview is de context op mesoniveau gedefinieerd. De context op mesoniveau gaat in op ilionx Groningen en de andere Cloud Solutions.

GRO-Cloud Solutions is een unit binnen ilionx in Groningen. De unit houdt zich voornamelijk bezig met het adviseren, implementeren en beheren van infrastructuren in de Cloud maar ook on-premise. Naast Cloud Solutions in Groningen, bestaat deze unit ook in Utrecht en Maastricht. Deze verschillende units hebben elkaar niet nodig en hanteren elk een verschillende werkwijze. Elke unit is op zijn eigen manier vormgegeven, om zo op verschillende manieren klanten te helpen. Het komt wel vaak voor dat de hulp van de andere units wordt ingeroepen omdat zij meer kennis hebben van bepaalde zaken.

De unit GRO-Cloud Solutions werkt standaard met Microsoft Azure. Deze werkwijze komt voort uit dat binnen ilionx voornamelijk met Microsoft producten wordt gewerkt en het meest kennis van is binnen de unit. Het bedrijf is niet Microsoft-only, maar werkt wel voornamelijk met deze producten. Verder is in minder mate ook kennis van AWS. Mocht een klant aankloppen bij GRO-Cloud Solutions die is aangesloten bij een andere vestiging van Cloud Solutions, is het beter om deze klant door te sturen naar deze andere vesteging. Omdat GRO-Cloud Solutions in de toekomst gaat werken met de oplossing, moet deze dus toegespitst zijn op zijn werkwijze. De wens van GRO-Cloud Solutions is om eerst te onderzoeken hoe een container gebaseerde oplossing in een CaaS-platform gehost kan worden. Platformen zoals PaaS en FaaS hoeven dus niet meegenomen te worden in het onderzoek.

Verder werkt GRO-Cloud Solutions veel samen met andere units binnen de vestiging Groningen. Het is hierbij wel zo dat zij niet afhankelijk zijn van deze andere units en dus prima op eigen kracht de behoefte van de klant kan voorzien [73].

15.3.2.2.3 Context – microniveau

Het 7s-model is bedacht door McKinsey om de prestatie van een onderneming te kunnen analyseren. Het 7s-model wordt daarom in dit onderzoek gebruikt om GRO-Cloud Solutions te bestuderen en om vervolgens aan de hand van deze resultaten randvoorwaarden op te stellen.

Het 7s-model is uitgewerkt door middel van een interview met de business unit manager [73]. De uitwerking en uitleg van het 7s-model zijn te vinden in 15.5.2.1. Hieronder staat per dimensie beschreven hoe de business unit manager van GRO-Cloud Solutions over de verschillende dimensies van dit model denkt.

Strategie

De dimensie strategie (strategy) beschrijft de lange termijn doelen van de afdeling GRO-Cloud Solutions. GRO-Cloud Solutions wil het aankomende jaar zich richten op vier verschillende punten: marktpositie, waarde propositie, goed werkgeverschap en unit-/vestiging-resultaat. Deze vier punten moeten in het aankomende jaar sterk worden verbeterd door hier in de unit veel aandacht en tijd aan te besteden. Verder gaat de unit sterk uit van eigen kracht en streeft altijd naar het leveren van een nog betere kwaliteit.

Systemen
De dimensie systemen (systems) omvat alle mogelijke systemen, zoals procedures, werkwijzen, voorschriften, communicatiestromen en afspraken binnen GRO-Cloud Solutions. De communicatie binnen GRO-Cloud Solutions gaat op verschillende manieren, maar voornamelijk via Microsoft Teams. Hiernaast zijn veel overleggen, zowel wekelijks met de teams als eenmaal in de vier weken met de hele unit. Ook zijn er lunchbijeenkomsten die klachtgericht zijn. De business unit manager wil ook dat de lead consultant ook met andere units praat. Na afloop van het gesprek willen de business unit managers de uitkomsten hiervan ook meteen in actie zetten. Belangrijkste systemen voor GRO-Cloud Solutions zijn Microsoft (MS) Azure, MS Teams, Office365 en MS Windows 10. Om als team op het spoor te blijven, worden de volgende "leading princpels" gehanteerd: wij gunnen elkaar wat, wij werken naast de klanten ook met elkaar samen en er moet een veilige omgeving zijn (het respecteren van elkaar).

Gedeelde waarden
De dimensie gedeelde waarden (shared values) zijn waarden die centraal staan binnen GRO-Cloud Solutions. Deze waarden zijn de waarden van ilionx. Binnen ilionx zijn er vijf kernwoorden: samen, doelgericht, dichtbij, bevlogen en innovatief. Deze vijf kernwaarden zijn heel sterk in de organisatie aanwezig. De cultuur van ilionx (GRO-Cloud Solutions) wordt als volgt omschreven: *"Onze cultuur laat zich samenvatten als een no nonsens cultuur waarin persoonlijke ontwikkeling, uitdagende opdrachten en innovatieve projecten centraal staan. Wij zijn van nature nieuwsgierig, vinden samen succes behalen belangrijk en hebben passie voor ons vak. Ondanks dat we bijzondere dingen steeds meer normaal vinden, zijn we trots op wat we met onze klanten bereiken en maken we graag het verschil. Onze lokale ilionx-vestigingen nemen een prominente plaats in de manier waarop we georganiseerd zijn. We ondernemen op het laagste niveau waarbij een collegiale sfeer en voor elkaar klaarstaan cruciale onderdelen zijn van ons werkplezier."*

Fundamentele waarden die hierbij horen zijn: persoonlijke- en samenwerkende benadering, lokale aanwezigheid, technologische kennis, branche ervaring en kennis van verschillende technologieën.

Personeel
Bij de dimensie personeel (staf) wordt een beeld gecreëerd van het personeel en management binnen GRO-Cloud Solutions. Bij Cloud Solutions Groningen werken veertig medewerkers. Deze medewerkers zijn onderverdeeld in consultants, engineers en beheerders. Zij hebben de volgende specialismen: Cloud, werkplek en packaging. Nieuw hierbij komt nog security. Binnen de unit is behoefte aan mensen met kennis over package, storage en VMware.

Structuur
De dimensie structuur (structure) gaat in op de structuur van GRO-Cloud Solutions. Binnen de unit GRO-Cloud Solutions zijn er business unit managers. Zij praten met de lead consultants en consultants. Op hun beurt praten zij met de engineers en beheerders. De consultants geven adviezen. De engineers zijn bezig met het implementeren. Het beheer onderhoudt het product. Al deze werknemers met verschillende taken moeten dus goed met elkaar kunnen overleggen. Ondertussen zijn de business unit managers continu op de hoogte van alles wat er binnen de unit speelt. Besluitvorming wordt zowel centraal als decentraal genomen. Dit wordt ook wel ondernemerschap genoemd; neem verantwoordelijkheid voor hetgeen wat je doet.

Stijl
De dimensie stijl (style) staat voor de vorm van management binnen GRO-Cloud Solutions. Het management van GRO-Cloud Solutions bestaat uit twee business unit managers. Zij hanteren een erg meewerkende en flexibele stijl van leidinggeven. Ze vinden het belangrijk dat de werknemer goed in zijn vel zit en zich op zijn gemak voelt met de klanten die hij heeft. De business unit managers proberen dit zo goed mogelijk na te streven. Wanneer er een probleem is, wordt er ook altijd hulp aangeboden. De stijl is niet altijd even effectief, wat vaak komt door miscommunicatie. Door deze stijl te hanteren is het team coöperatief, maar is er ook sprake van enige competitie. Dit zorgt ervoor dat het een heel goed en effectief team is. De effecten hiervan zijn dat de medewerkers zich aan elkaar optrekken om zo betere prestaties te leveren.

Vaardigheden
Bij de dimensie vaardigheden (skills) wordt gekeken naar het onderscheidend vermogen van GRO-Cloud Solutions. De vaardigheden die binnen de unit vertegenwoordigd worden zijn: werkplek beheer, Cloud en packaging service. De unit GRO-Cloud Solutions staat binnen ilionx bekend om de werkplek beheer, Virtual Desktop Infrastructure (VDI) en packaging. De vaardigheden van de werknemer wordt bij GRO-Cloud Solutions elke acht weken beoordeeld in een individueel gesprek met de business unit manager. Verder bepalen de medewerkers zelf welke certificeringen zij willen behalen voor verdere (persoonlijke) groei.

15.3.3 Conclusie
In het bovenstaande hoofdstuk zijn de verschillende aggregatieniveaus onderzocht. Hieruit zijn veel verschillende gegevens naar voren gekomen. Deze gegevens worden vervolgens omgezet naar randvoorwaarden. Randvoorwaarden zijn regels waaraan de oplossing moet voldoen.

15.3.3.1 Randvoorwaarden
De randvoorwaarden staan in Tabel 6. Deze opgestelde randvoorwaarden zijn opgesteld om de oplossingen in te perken.

15.4 Literatuuronderzoek

In het literatuuronderzoek wordt de literatuur geraadpleegd voor het vinden van antwoorden op de onderzoeksvragen. Het zoeken in de literatuur gebeurt aan de hand van de verschillende valide methodes, om zo een valide en betrouwbaar literatuuronderzoek uit te voeren.

15.4.1 Methode

In de probleemanalyse zijn in hoofdstuk 15.1.3.1 de hoofd- en deelvragen opgesteld. De deelvragen (Tabel 2) beantwoorden ieder een gedeelte van de hoofdvraag (Tabel 1). De zesde en zevende deelvraag kan niet door middel van het literatuuronderzoek worden beantwoord, het antwoord op deze vraag wordt in de probleemanalyse en Ontwerpdocumentatie geformuleerd. De overige deelvragen worden binnen het literatuuronderzoek beantwoord.

Hieronder staan een aantal zoekmethoden opgesomd. Deze zijn ieder uitgewerkt om zo uiteindelijk te kunnen concluderen welke methode het beste past bij het literatuuronderzoek van dit onderzoek.

15.4.1.1 Zoekstrategie

Om op een juiste en gestructureerde manier literatuuronderzoek te verrichten, wordt een zoekstrategie gebruikt [74]. In de zoekstrategie worden de volgende punten behandeld:

- Wat voor informatie is er nodig?
- Waar komt de informatie vandaan?
- Hoe is de informatie beoordeeld?

De bovenstaande punten zijn uitgewerkt en uitgewerkt in een zoekstrategie, welke terug te vinden is in hoofdstuk 15.5.3.

15.4.1.2 Sneeuwbalmethode

De sneeuwbalmethode [75] is één manier om geschikte literatuur te vinden. Door het raadplegen van de literatuurlijst van een bron, kunnen meer interessante bronnen worden gevonden over het onderwerp. Door daarna weer de literatuurlijst van deze nieuwe gevonden bron te raadplegen, kunnen nog weer meer relevante bronnen gevonden worden. Door de sneeuwbalmethode kan snel en relatief gemakkelijk veel nieuwe en relevante literatuur worden verzameld. Een nadeel hiervan kan zijn dat er terug wordt gezocht in de tijd en daardoor oudere literatuur gevonden wordt.

15.4.2 Resultaten

In dit hoofdstuk worden de deelvragen beantwoord. Dit is gedaan door middel van literatuuronderzoek aan de hand van verschillende bovenstaande methodes.

15.4.2.1 Deelvraag 1: Containers

Voor het correct beantwoorden van deze deelvraag moet eerst worden uitgelegd wat een container precies is. Aan de hand van de informatie over wat een container is, kan duidelijk worden gemaakt wat precies onder een container applicatie wordt verstaan.

Containers zijn geïsoleerde en portable operating environments waarmee een applicatie (app) kan worden gescheiden van de hardware waarop deze wordt uitgevoerd (hosting). Door middel van de splitsing van app en de hosting, kan een container gebaseerde applicatie eenvoudig en consistent worden geïmplementeerd. Er wordt steeds meer gebruik gemaakt van containers, dit komt voornamelijk voort uit de volgende voordelen:

1. Een applicatieontwikkelaar kan zijn applicatie makkelijker testen met containers.
2. Het loskoppelen van de infrastructuur en het gemakkelijk toevoegen- en verwijderen van containers applicaties zorgt ervoor dat updates snel, veelvoudig en eenvoudig doorgevoerd kunnen worden ongeacht op welke infrastructuur.

Door het gebruik van containers is het voor een applicatieontwikkelaar makkelijker om een applicatie testen. Dit komt door dat de infrastructuur is losgekoppeld van de applicatie zelf. Door dat de infrastructuur is losgekoppeld van de applicatie is de applicatieontwikkelaar niet meer afhankelijk van de IT-infrastructuur medewerkers om een bijpassende omgeving op te zetten.

Het gebruik van containers door middel van een CI-/CD-pipeline (Continuous Integration, Continuous Delivery) zorgt voor frequentere updates voor de container gebaseerde applicaties en herhaalbare implementatieprocessen te creëren. Dit is mogelijk vanwege het licht en wendbare karakter van een container, omdat elke container veel minder code bevat dan wanneer een volledige server moet worden bijgewerkt. Het gebruik van een container gebaseerde oplossing heeft ook als voordeel dat verschillende omgevingen (e.g. productie, test) vrijwel altijd identiek zijn, dit komt omdat een applicatieontwikkelaar werkt vanuit een bestaande container en deze vervangt met een nieuwer versie. Hiermee is de werking van de applicatie in de verschillende omgevingen gelijk [76].

De isolatie van een container zorgt voor een goede scheiding van werkzaamheden. Ontwikkelaars richten zich op de code van de app, terwijl een Cloud- of Infrastructuurteam zich richt op de implementatie en het beheer van de hosting, zonder zich druk te hoeven maken over app-details [4]. Deze vorm van werken wordt DevOps [5] genoemd. DevOps is een combinatie van ontwikkeling (Dev) en activiteiten zoals implementatie en beheer (Ops). DevOps is een samenstelling van mensen, processen en technologieën om doorlopend van waarde te kunnen zijn voor de klant.

Vaak worden containers vergeleken met een VM. Een VM is een gastbesturingssysteem (Linux/Windows) dat wordt uitgevoerd op een hostbesturingssysteem (Microsoft Hyper-V of VMware ESXi) en krijgt zo gevirtualiseerd toegang tot de onderliggende hardware. Net zoals bij een VM, kan een container een app samen met bibliotheken en andere afhankelijkheden verpakken. Hierdoor wordt het een geïsoleerde omgeving waarin een softwareservice wordt uitgevoerd [4]. Zoals in Figuur 4 [6] te zien is, is het verschil tussen een VM en containers dat de virtualisatie op een andere laag van de stack plaatsvindt. Bij VM's draaien door middel van een hypervisor verschillende VM's op één fysieke server. Bij containers wordt echter het besturingssysteem, in plaats van de hardware, gevisualiseerd. Dit wordt gedaan door middel van een container engine.

Een container engine voorziet in de basis drie dingen:

1. Het zorgt voor de communicatie tussen de eindgebruiker en container door middel van een API en/of user interface.
2. Het verzorgt het beschikbaar stellen van de container image(s) vanuit een container image repository en/of harde schijf.
3. Als laatste zorgt de container engine ervoor dat van de container images een configuratie (config.json/config.yml) bestand wordt gemaakt. Deze file kan worden uitgevoerd door de container engine doordat deze doorgestuurd wordt naar de kernel van het besturingssysteem waarop de container engine draait [77].

15.4.2.2 Deelvragen 2, 3 en 4: Onderzoek naar containerplatformen

In dit hoofdstuk worden deelvraag twee, drie en vier beantwoord. Eerst moet worden onderzocht welke verschillende mogelijkheden er zijn om een container gebaseerde applicatie te hosten binnen Microsoft Azure. De mogelijke container hosting oplossingen worden daarna onderzocht op de in hoofdstuk 15.4.2.2.2 gedefinieerde componenten. Dit wordt gedaan door literatuuronderzoek naar de verschillende componenten.

Hieronder wordt uitgewerkt welke verschillende mogelijkheden er zijn binnen Azure voor het hosten van een container applicatie en wat de kosten hiervan zijn.

15.4.2.2.1 Container oplossingen binnen Microsoft Azure

Zie hoofdstuk 8.6.2.1 voor de uitwerking.

15.4.2.2.2 Gebruikte componenten

De hierboven benoemde container oplossingen worden in het literatuuronderzoek onderzocht op verschillende componenten (eigenschappen). In dit hoofdstuk wordt per component uitgelegd wat het inhoudt en hoe het wordt beoordeeld.

Hoe goed is de beschikbaarheid van de oplossing?
Beschikbaarheid wordt als volgt gedefinieerd: “Beschikbaarheid geeft aan in hoeverre een ICT-dienst, -systeem of -component toegankelijk is voor de gebruiker” [78].

Wat is de betrouwbaarheid van de oplossing?
Bij betrouwbaarheid wordt gekeken naar de consistentie en continuïteit. Er wordt hier gekeken naar hoe de data beveiligd is en of data bij verlies snel kan worden teruggezet.

Hoe schaalbaar is de oplossing?
Schaalbaarheid is hoe goed er bij een piekmoment kan worden bijgeschaald om zo de container gebaseerde applicatie beschikbaar te houden voor de eindgebruiker. Schalen van een services kan worden gedaan op twee manieren:

1. Bij verticaal schalen worden de capaciteiten van het systeem vergroot om zo meer aan te kunnen.
2. Bij horizontaal schalen wordt de belasting verdeeld over een extra server. Dus in plaats van een hele zware server, twee gewone servers met een load balancer ertussen.

Bij schaalbaarheid wordt nu gekeken naar hoe goed de container hosting horizontaal kan worden geschaald. Het verticaal schalen van een systeem zorgt namelijk voor downtime omdat het hele systeem daar voor moet worden uit gezet. Om de schaalbaarheid te onderzoeken wordt gekeken naar de volgende punten:

- In hoeverre is de oplossing automatisch schaalbaar? Als de oplossing niet schaalbaar blijkt, hoe is de oplossing dan wel schaalbaar?
- Welke mogelijkheden heeft de oplossing voor het automatisch schalen van container applicaties?

Hoe veilig is de oplossing?
Voor het component veiligheid wordt onderzocht welke mogelijkheden de container oplossing biedt om de veiligheid van de containers en gegevens te waarborgen. Er wordt bij veiligheid gekeken naar de standaard beveiliging van de oplossingen zonder dat er wat is ingesteld.

Welke kosten heeft de oplossing?
Dit component gaat in op de kosten van de container oplossing zonder extra Azure services en hoe duur is het om de test applicatie te laten draaien.

Hoe werkt het prijsmodel?
Bij afname van services bij Azure worden vaak verschillende prijsmodellen gebruikt. Een prijsmodel is een manier om de dienst af te nemen. Per mogelijke oplossing worden de verschillende prijsmodellen onderzocht.

Welke features/mogelijkheden heeft de oplossing?
De mogelijke features/mogelijkheden van de container oplossingen worden onderzocht en uitgelegd.

Welke voor- en nadelen heeft de oplossing?
De voor- en nadelen worden per oplossing onderzocht.

15.4.2.2.3 Container gebaseerde applicatie
Zie hoofdstuk 8.6.2.3 voor de uitwerking.

15.4.2.2.3.1 Databases
De data van een container gebaseerde applicatie moet ergens worden opgeslagen, wat vaak gebeurt in een database. Hiervoor zijn verschillende mogelijkheden beschikbaar binnen Microsoft Azure. De voorbeeldapplicatie maakt gebruik van Microsoft SQL database en van MongoDB of CosmosDB.

Microsoft SQL database
Microsoft SQL Server is een relationeel databasebeheersysteem ontwikkeld door Microsoft. Binnen Microsoft Azure is dit de Azure SQL Database [79]. Deze wordt aangeprezen als een intelligente- en schaalbare Cloud databaseservice die de breedste compatibiliteit met SQL Server-engines zou moeten bieden [80].

De voorbeeldapplicatie maakt gebruik van vier SQL databases, welke gehost kunnen worden op een SQL server instance. Voor deze instance wordt gekozen voor een G5 database met acht virtuele kernen (Vcore) van 40,8 GB aan geheugen [81]. De uitgekozen databases is in Tabel 20 uitgewerkt aan de hand van de gedefinieerde componenten om te bekijken wat de invloed is op de applicatie.

Component	**Azure SQL database G5 8Vcore 40.8GB**
Beschikbaarheid	De database heeft een beschikbaarheidsgarantie van minimaal 99.99% [82].
Betrouwbaarheid	De data op de SQL databases kan worden beveiligd op veel manieren (zie ook veiligheid). Verder maakt Azure SQL database automatisch database back-ups die worden bewaard voor de duur van de geconfigureerde bewaarperiode [83].
Schaalbaarheid	Door middel van hyperscale kan de database verder schalen dan de standaard limieten van het gekozen [84].
Veiligheid	De beveiliging van de SQL databases kan op vier manieren worden toegepast [85]: 1. Netwerkbeveiliging; 2. Autorisatie; 3. Controle- en bedreigingsdetectiemogelijkheden; 4. Informatiebescherming en versleuteling.
Kosten en prijsmodel	De database kost per maand €1.316,94. De database kan worden afgenomen onder het pay-per-use model, maar kan ook worden vast gelegd voor één of drie jaar [81].
Features/mogelijkheden	Azure SQL Database heeft de volgende mogelijkheden [80]: - Migratie van on-premises SQL naar Cloud SQL; - Ingebouwde machine learning; - Dashboard voor schaalbaarheid en beschikbaarheid.
Voor– en nadelen	Voordelen [86]: - Infrastructuur beheerd door Azure; - Back-ups beheerd door Azure; - Lift & shift-migraties mogelijk. Nadelen [86]: - Sommige functies zijn niet beschikbaar (SQL on- premises).

Tabel 20: Componenten SQL database

15.4.2.2.3.2 MongoDB of Azure CosmosDB

De voorbeeldapplicatie maakt naast een SQL database ook nog gebruik van een non-relational data structuur database (NoSQL). Hiervoor wordt in de voorbeeldapplicatie MongoDB of Azure CosmosDB gebruikt. Een NoSQL database is een database die de data niet vooraf voorziet van een (semi)structuur [87].

Beide databases worden onderzocht op de gedefinieerde componenten (zie hoofdstuk 15.4.2.2.2). De resultaten hiervan worden in Tabel 21 hieronder uitgewerkt.

Component	MongoDB M10 2GB ram en 32 GB opslag	Azure CosmosDB
Beschikbaarheid	De beschikbaarheid van een M10 cluster is 99.99% doordat dit cluster wordt geplaatst in een Azure Availability Zone (AZ) [88].	De beschikbaarheid van Azure CosmosDB van 99.99% [89].
Betrouwbaarheid	De betrouwbaarheid hangt af van de implementatie, op de manier met de verschillende AZ is de kans groot dat altijd een server online is, maar dit ligt dus sterk aan de vorm van implementatie. Back-ups worden daarentegen niet geregeld en moeten dus handmatig of met een Azure back-up worden ingesteld [90].	Azure CosmosDB maakt automatisch regelmatig back-ups van de gegevens. De automatische back-ups worden genomen zonder dat dit gevolgen heeft voor de prestaties of beschikbaarheid van de databasebewerkingen. Alle back-ups worden afzonderlijk opgeslagen in een opslagservice. Deze back-ups worden wereldwijd gerepliceerd voor tolerantie tegen regionale rampen [91].
Schaalbaarheid	Na deployment kan het cluster in capaciteit worden geüpgraded [90].	Het biedt kant-en-klare wereldwijde distributie binnen een onbeperkt aantal Azure-regio's [89].
Veiligheid	Ook de veiligheid van de database moet zelf worden ingesteld. Dit moet worden gedaan door de systeembeheerders zelf.	De beveilig van de Azure CosmosDB wordt gedaan op vier manieren [92]: 1. Netwerkbeveiliging 2. Autorisatie 3. Controle- en bedreigingsdetectiemogelijkheden 4. Informatiebescherming en versleuteling
Kosten en prijsmodel	Het uitgekozen cluster wordt betaald per uur. De kosten zijn €0,13, dit komt uit op een maandbedrag van €95.16 per maand [93].	Azure CosmosDB kan worden afgenomen voor prijs per maand of vaste prijs bij een afname van één of drie jaar. De kosten van een maand Azure CosmosDB voor een omgeving van 10 x 100 RU en schrijven in meerdere regio's ligt op €108,15 per maand [24].
Features/mogelijkheden	MongoDB heeft de volgende mogelijkheden [94]: - Cluster MongoDB; - Load balancing.	Azure CosmosDB heeft de volgende mogelijkheden [95]: - Wereldwijde distributie; - Toegang via API.
Voor – en nadelen	Voordelen [96]: - Prestatie; - Gemakkelijkheid; - Flexibiliteit; - Schaalbaarheid. Nadelen [96]: - Indexering.	Voordelen: - Prestatie; - Wordt veel geregeld door Microsoft; - Lage latentie. Nadelen: - Bij zwaar gebruik schieten de kosten omhoog.

Tabel 21: Componenten MongoDB vs. CosmosDB

15.4.2.2.3.3 API gateway

Een API zorgt ervoor dat er op meerdere manieren kan worden gepraat met de back-end. “Een API maakt communicatie en het uitwisselen van informatie mogelijk tussen verschillende (software systemen) [97].” De als voorbeeld genomen container gebaseerde applicatie maakt ook gebruik van API’s om opdrachten van de client door te sturen naar de back-end. Deze API’s kunnen op twee manier worden gehost:

1. De API in een container: Met deze optie zouden de API’s net als de microservices worden gehost binnen een container oplossing. Een voordeel hiervan is dat het in eigen beheer is.
2. De API hosten binnen Azure API Management: De API wordt gehost binnen Microsoft Azure. Het beheren van de API’s kan zowel via de Azure portal of met de CLI. Een voordeel hiervan is dat alle API’s in één overzicht te vinden zijn. Daarnaast wordt de hard- en software beheerd, gelicenceerd en beveiligd door Microsoft Azure. Hierdoor valt een hoop beheer weg ten opzichte van een API in een container.

Waar uiteindelijk de API’s worden gehost heeft beperkte invloed op de container gebaseerde applicatie. De verschillen zullen vooral merkbaar zijn in de beschikbaarheid, veiligheid en kosten. Wat de verschillen zijn tussen Azure API Gateway en hosting in een container, hangt sterk af van de gekozen oplossing voor het hosten van containers. Mede hierdoor is in Tabel 22 alleen Azure API Gateway onderzocht op de verschillende componenten. Container hosting van Microsoft Azure wordt onderzocht in hoofdstuk 8.6.2.5.

Component	**Azure API Gateway standard**
Beschikbaarheid	De beschikbaarheid van API Gateway standard van 99.99% [98].
Betrouwbaarheid	API’s worden door Microsoft geback-upt, en dertig dagen bewaard. Hierdoor is er de mogelijkheid om dertig dagen terug te kunnen in de tijd [99].
Schaalbaarheid	Zit niet bij het standaard pakket in, dit wordt automatisch door Azure gedaan vanaf het premium tier pakket [98].
Veiligheid	API Gateway brengt verschillende mogelijkheden voor beveiliging met zich mee op het gebied van [100]: - Netwerk; - Controle en logging; - Authenticatie; - Gegevensbeveiliging; - Configuratiebeheer.
Kosten en prijsmodel	De kosten voor een API Gateway standaard zijn per maand €1105,- [24].
Features/mogelijkheden	Azure API Gateway biedt de volgende mogelijkheden [101]: - Resources beveiliging; - Versnellen bedrijfsactiviteiten; - Verbetering van API; - Omzetten webservices tot REST API.
Voor – en nadelen	Voordelen: 1. Microsoft regelt de hosting, back-up en beveiliging van de API’s. 2. Veel mogelijkheden voor het beveiligen van de verschillende API’s. Nadelen: 1. Hoge kosten.

Tabel 22: Componenten Azure API Gateway

15.4.2.2.3.4 Event bus

De applicatie die wordt gebruikt is een microservice applicatie. Elke microservices draait een eigen container. Om er voor te zorgen dat al deze verschillende microservices met elkaar kunnen communiceren, wordt bij dergelijke applicaties gebruik gemaakt van een messaging service. Er zijn verschillende soorten messaging services beschikbaar. Voor deze applicatie wordt aangeraden te werken met Azure Service Bus [102] of met RabbitMQ [103].

Omdat het uiteindelijke resultaat van een onderzoek naar deze twee messaging services geen invloed heeft op het uiteindelijk resultaat van het onderzoek of de werking van de applicatie, wordt standaard gekozen voor Azure Service Bus. De keuze is hiervoor gemaakt omdat Azure Service Bus kan worden afgenomen als een dienst van Azure en dus geen beheer vereist. Dit zou echter wel het geval zijn geweest bij RabbitMQ. Verder blijkt uit een post op stackoverflow en een vergelijking [104] dat er bijna geen verschil zit tussen deze twee messaging services.

15.4.2.2.4 Uitwerking componenten per mogelijke container oplossing

Zie hoofdstuk 8.6.2.5 voor de uitwerking.

15.4.2.2.5 Overige container oplossingen binnen Microsoft Azure

Zie hoofdstuk 8.6.2.2 voor de uitwerking.

15.4.2.3 Deelvraag 5: Benodigdheden geautomatiseerde uitrol infrastructuur

Zie hoofdstuk 8.6.3 voor de uitwerking.

15.4.3 Conclusie

Hieronder is de conclusie per deelvraag uitgewerkt.

15.4.3.1 Conclusie deelvraag 1

Containers zijn geïsoleerde en portable operating environments waarmee een applicatie (app) kan worden gescheiden van de hardware waarop deze wordt uitgevoerd (hosting). Een container gebaseerde applicatie is een applicatie die wordt ontwikkeld met alle benodigdheden om zelfstandig te kunnen worden uitgevoerd op een container engine. De applicatie is niet afhankelijk van het type host waarop het wordt uitgevoerd.

15.4.3.2 Conclusie deelvragen 2, 3 en 4

AKS, ARO en VM met Docker zijn de verschillende mogelijkheden binnen Azure voor het hosten van een container applicatie. Hierbij zijn nog twee andere opties niet meegenomen in het onderzoek omdat deze Faas- en PaaS-platformen zijn en dus niet van toepassing waren. In het voorbeeld getoonde applicatie is aangetoond dat alle generieke componenten die een container gebaseerde applicatie nodig heeft, aanwezig zijn binnen Azure. In hoofdstuk 7.2.2.4. Zijn ook de kosten berekend aan de hand van de voorbeeldapplicatie die redundant zijn uitgevoerd. Hieruit blijkt dat de kosten van AKS het laagst, en de kosten voor ARO het hoogst zijn.

15.4.3.3 Conclusie deelvraag 5

Het automatisch uitrollen van een infrastructuur binnen Microsoft Azure kan door middel van ARM en Azure DevOps. Binnen Azure DevOps kan een CD-pipeline worden aangemaakt waarin het ARM-template wordt aangesproken en uitgevoerd. Een ARM template is een JSON bestand waarin gescripte resources zijn gedefinieerd.

15.5 Bijlagen bij het vooronderzoek

Hieronder zijn alle bijlagen van het vooronderzoek bijgevoegd.

15.5.1 Bijlagen probleemanalyse

Als bijlage van de probleemanalyse zijn de uitwerkingen van de 5W1H vragen bijgevoegd.

15.5.1.1 5W1H vragen

Hieronder staan de vragen die gesteld zijn voor het 5W1H methode. De vragen zijn gesteld in een interview/gesprek.

Wie-vragen:

- Wie zijn er betrokken bij het ontstaan van het probleem en/of vraagstuk?
- Wie zijn stakeholders?
- Wie is verantwoordelijk voor wat?
- Wie wil dit probleem/vraagstuk opgelost zien?
- Op wie heeft dit project invloed?
- Wie beslist wat er gaat gebeuren?
- Wie is betrokken bij het oplossen?

Wat-vragen:

- Wat is het probleem?
- Wat zijn de mogelijke oorzaken?
- Wat willen we bereiken met de oplossing?
- Wat zijn de grenzen waarbinnen de oplossingen gevonden moet worden?
- Wat is de beoogde impact van de oplossing?
- Wat moet het project opleveren?

Waar-vragen:

- Waar doet het net gespecificeerde probleem zich voor?
- Waar moet de oplossing worden gebruikt?

Wanneer-vragen:

- Wanneer doet het probleem zich voor?
- Wanneer is de vraag ontstaan?
- Wanneer moet het probleem opgelost zijn?

Hoe-vragen:

- Hoe moet het probleem worden opgelost?
- Hoe is het probleem ontstaan?
- Hoe wordt eraan getoond dat de oplossing juist is?

15.5.1.1.1 5W1H business unit manager

Hieronder staan de vragen van het interview van de probleem- en behoefteanalyse beschreven. Deze vragen zijn opgesteld aan de hand van het 5W1H methode. Aan de hand van een interview met de business unit manager zijn deze vragen beantwoord. Dit interview vond plaats op 25 februari 2020 tussen 13:00 en 14:00. In verband met de AVG-wetgeving zijn de persoonsgegevens geanonimiseerd.

Wie zijn er betrokken bij het ontstaan van het probleem en/of vraagstuk?
Het vraagstuk is ontstaan bij een klant. Deze klant heeft een e-commerce applicatie die in een container draait. Vanuit deze vraag zijn wij samen met een consultant na gaan denken over waar de technische verantwoordelijkheid ligt.

Wie zijn stakeholders?
De stakeholders van het vraagstuk zijn:

- Ik, de business unit manager;
- De lead consultant;
- De co-collega's van ilionx;
- De klanten.

Dit zijn de stakeholders omdat containers dingen zijn waarin de applicatie draait en die applicatie wordt geprogrammeerd door een andere unit.

Wie is verantwoordelijk voor wat binnen de divisie Cloud Solutions?
Dat ben ik als business unit manager. Maar wanneer wij onze businessstrategie uitzetten, betrekken wij altijd de lead architecten erbij omdat die uiteindelijk de strategie moeten uitvoeren.

Wie wil dit probleem/vraagstuk opgelost zien?
De businessunit Cloud Solutions.

Op wie heeft dit project invloed?
De uitkomst van het project heeft invloed op onze unit en de klant. Het project is voor de unit een stukje Research and Development (R&D), waar wij als unit normaal niet veel tijd voor hebben. Wij hebben zo dit onderzoek als referentiemateriaal.
De klant heeft straks baat bij de gemaakte whitepaper, hierdoor wordt de klant voorgelicht over container gebaseerde applicaties.

Wie beslist wat er gaat gebeuren?
Wij als ilionx hebben een visie die door alle units wordt overgenomen en vertaald wordt naar de strategie van de unit. Maar hoe gek het ook klinkt, bij onze unit bepaald iedere werknemer zelf wat het beste is voor de klant. Hiernaast is de werknemer ook zelf verantwoordelijk voor het eindresultaat.

Wie is betrokken bij het oplossen?
In het begin zijn alleen de lead consultant en ik als business unit manager betrokken. Later als de whitepaper geschreven wordt, is iedereen in deze unit betrokken. Dit komt omdat het een opkomend fenomeen is en wij er allemaal mee te maken krijgen.

Wat is het probleem?
Er is geen probleem, maar een vraagstuk. Deze komt voort uit een bestaande uitdaging bij een klant.

Wat zijn de mogelijke oorzaken?
De opkomst van container gebaseerde applicatie is de grootste oorzaak.

Wat willen wij bereiken met de oplossing?
Wij willen het onderzocht hebben en wij willen weten hoe het werkt. Hiernaast willen wij uiteindelijk gezamenlijk met andere units van ilionx een klant zo goed mogelijk kunnen voorlichten. Het uiteindelijke doel is om de klant te kunnen binnenhalen.

Wat zijn de grenzen waarbinnen de oplossingen gevonden moet worden?
Er zijn een aantal grenzen. Het eerste is dat er een oplossing gevonden moet worden binnen de huidige leveranciers die wij nu gebruiken. Als je het zou kunnen beperken tot Microsoft en de derde partijen die containerplatformen leveren aan Microsoft, dan is dat heel mooi zijn. Het financiële aspect is hier geen grens, wij kijken naar wat het oplevert niet naar wat het kost. Verder zit er nog een beperking aan de tijd die er is voor het onderzoek. Het onderzoek moet voor het einde van de afstudeerperiode afgerond zijn.

Wat is de beoogde impact van de oplossing?
De impact van de oplossing ligt binnen de unit, maar ook zeker bij de klant. De unit kan uit jouw onderzoek kennis op doen. Voor de klant is het referentiemateriaal en houvast van wat wij kunnen bieden.

Wat moet het project opleveren?
Wij willen tegen een klant kunnen zeggen dat wij kennis hebben van containers en dit is onze whitepaper.

Waar doet het net gespecificeerde probleem zich voor?
Niet, het vraagstuk kwam naar voren bij een klant, en wij willen nu voorkomen dat het een echt probleem wordt.

Waar moet de oplossing worden gebruikt
Binnen ilionx, de unit Cloud Solutions en bij de klant.

Wanneer doet het probleem zich voor?
Niet, het vraagstuk kwam naar voren bij een klant, en wij willen nu voorkomen dat het een echt probleem wordt (idem).

Wanneer is de vraag ontstaan?
Het vraagstuk is ontstaan bij een klant. Deze klant heeft een e-commerce applicatie die in een container draait (idem).

Wanneer moet het probleem opgelost zijn?
Het zou fijn zijn dat het vraagstuk beantwoord is aan het einde van de stageperiode, maar het is niet gezegd dat dat met een whitepaper kan. Hierover kunnen wij niet echt een uitspraak doen.

Hoe moet het probleem worden opgelost?
Op dit moment moet het probleem worden opgelost met een whitepaper, maar misschien is het wel zo dat jij tijdens je onderzoek wel wat anders tegenkomt wat veel duidelijker is.

Hoe is het probleem ontstaan?
Het vraagstuk is ontstaan bij een klant. Deze klant heeft een e-commerce applicatie die in een container draait (idem).

Hoe wordt er aangetoond dat de oplossing juist is?
Doordat duidelijk in de whitepaper staat wat container applicaties opbrengen, wat de voordelen ervan zijn en waarom ze gebruikt moeten worden.

15.5.1.1.2 5W1H lead consultant

Hieronder staan de vragen van het interview van de probleem- en behoefteanalyse beschreven. Deze vragen zijn opgesteld aan de hand van het 5W1H methode. Aan de hand van een interview met de een lead consultant zijn deze vragen beantwoord. Dit interview vond plaats op 27 februari 2020 tussen 14:00 en 15:00. In verband met de AVG-wetgeving zijn de persoonsgegevens geanonimiseerd.

Wie zijn er betrokken bij het ontstaan van het probleem en/of vraagstuk?
De klanten, de verkoper, lead consultant, business unit manager en developers van andere units.

Wie zijn stakeholders?
De doelgroep is de klant (als doelgroep). De stakeholders zijn de lead consultant, business unit manager en in mindere mate de sales (of er niets in staat wat ilionx kan schaden bij de klant).

Wie is verantwoordelijk voor wat binnen de divisie Cloud Solutions?
Uiteindelijk bepaalt toch de business unit manager het. Alleen leunen zij sterk op de collega's in de unit natuurlijk. En voornamelijk de lead consultant en de kennishouders van een bepaald product, daar baseren zijn natuurlijk hun beslissing op.

Wie wil dit probleem/vraagstuk opgelost zien?
De unit, en zou dan kunnen zeggen dat sales daar ook achter zit samen met de lead consultant, die moeten dat verkopen bij de klanten.

Op wie heeft dit project invloed?
In eerste instantie de engineers, de consultants die het moeten uitvoeren. Het zou hierbij zo kunnen zijn dat die mensen nog de technieken moeten leren

Wie beslist wat er gaat gebeuren?
Uiteindelijk bepaalt toch de business unit manager het. Alleen leunen zij sterk op de collega's in de unit natuurlijk. En voornamelijk de lead consultant en de kennishouders van een bepaald product, daar baseren zijn natuurlijk hun beslissing op. (Idem)

Wie is betrokken bij het oplossen?
Ik denk dat het handig is om te informeren bij mensen die het al gebruiken, dus al containers gebruiken. Waarom worden bepaalde keuzes gemaakt? En misschien worden die keuzes wel gemaakt omdat ze niet anders weten? Maar daarna vragen ze ook aan developers en lead architects die die al wel kennis hebben van containers, maar niet in detail wat de mogelijkheden zijn. Ik denk dat je ook Microsoft zelf als bron kan gebruiken.

Wat is het probleem?
Nou, ik denk dat dit een beetje een kip ei verhaal is. Als wij als Cloud-/infrastructuurspecialisten goed weten wat je met containers kan je wat voor oplossingen zijn. Dan kunnen wij developers informeren. Dan kunnen zij het gaan ontwikkelen op basis van een slimme architectuur. Dan kunnen zij daar rekening mee houden. Op dit moment is het nog andersom, maar de verrassing krijgen van dit is de applicatie en deze containers zijn gemaakt. En zo zit het in elkaar. Wij moeten hierbij de oplossing zoeken om toch alles werkend te krijgen. Ik denk dat je dat liever omdraait, dat je vanuit de infrastructuur wil adviseren wat een handige manier is om een applicatie op te bouwen. Dus dat het handig is om te draaien.

Wat zijn de mogelijke oorzaken?
Ik denk dat het vooral het opkomen van DevOps is, omdat de mogelijkheden voor een developer gewoon veel uitgebreider zijn. Zij kunnen veel makkelijker resources zelf aanvragen. Maar dat betekent niet dat zij er ook echt verstand van hebben. Ze doen het dus waarschijnlijk uit noodzaak dat zij gewoon snel aan de slag willen en dus niet wachten op een infrastructuur.

Wat willen wij bereiken met de oplossing?
Kennis delen over hoe je best een container applicaties infrastructuur technisch kan opbouwen en zo dus advies geven en dus ook voor een traject gemaakt kan zitten bij het ontwikkelen van applicaties zodat je wel invloed hebt over hoe de uiteindelijke applicatie opgebouwd wordt. Dat lijkt mij ook gewoon beter dan wat de mogelijkheden zijn, en daar de mensen ook voor trainen. Hiernaast kan de klant ook adviseren over de opties.

Wat zijn de grenzen waarbinnen de oplossingen gevonden moet worden?
Microsoft Azure mits de kosten niet heel veel hoger zijn dan andere Cloudoplossingen en mogelijkheden niet heel veel minder. Dus dit is een kwestie van snel toetsen op dezelfde oplossing of binnen een andere Cloud vergelijkbare kosten heeft dan wel wat duurder is. En ja, het moet wel een proven technology zijn, dus in de markt veel gebruikt worden.

Wat is de beoogde impact van de oplossing?
Dat de kennis verhoogd wordt en dat het duidelijk is wat je hiermee kan.

Wat moet het project opleveren?
Kennis over containers en een whitepaper voor de klant.

Waar doet het net gespecificeerde probleem zich voor?
Nee ik denk over de hele vestiging wel want er zijn over de vestiging Groningen wel veel meer containers gebruikt. Alleen op een niet gestructureerde manier beheert, en misschien gebruiken de andere units ook niet de meest optimale omgeving. Dus ik denk dat wij onszelf ook wel als interne klant kunnen beschouwen.

Waar moet de oplossing worden gebruikt?
Binnen ilionx voor onze eigen dienstverlening vanuit Groningen dus de afdelingen business intelligence science en application development. Hiernaast ook voor de klanten van ilionx om die goed advies te kunnen geven over welke container hosting het best gebruikt kan worden.

Wanneer is de vraag ontstaan?
Vijf maanden geleden, toen kwam de klantvraag binnen, en toen was de vraag hoe doen wij dat? In een markt wordt er veel meer gebruikt, maar bij ons is het helemaal actueel geworden op het moment dat wij met een offerte moesten komen voor deze vraag van de klant.

Wanneer moet het probleem opgelost zijn?
In principe lossen wij het nu ook altijd op, maar hierbij is de vraag of wij de beste keuzes maken. Door middel van dit onderzoek willen wij een keuzerichtlijn maken voor de oplossingen die beschikbaar zijn zodat bij de volgende klantvraag wij hun probleem sneller kunnen oplossen. Er zit dus voor het probleem geen eindoplossing aan, maar de whitepaper moet af zijn voordat jij afgestudeerd bent.

Hoe moet het probleem worden opgelost?
Door middel van een onderzoek naar containers binnen Microsoft Azure. Naar het onderzoek moet er voor de klant een whitepaper worden gemaakt over mogelijke oplossingen.

Hoe is het probleem ontstaan?
Vijf maanden geleden, toen kwam de klantvraag binnen, en toen was de vraag hoe doen wij dat? In een markt wordt er veel meer gebruikt, maar bij ons is het helemaal actueel geworden op het moment dat wij met een offerte moesten komen voor deze vraag van de klant (Idem).

Hoe wordt eraan getoond dat de oplossing juist is?
Ten eerste is er de toetsing van de reactie bij de klant, maar ook of een niet bestaande klant, dan klant bij ilionx wordt. Als de klant kies voor de oplossing dan weet je dat je het goed hebt gedaan. En of dat dan ligt aan de whitepaper is dan natuurlijk de vraag. En ik denk dat het handig is om te toetsen naar mensen die dat gebruiken, dus binnen de unit die al containers gebruiken bij bijvoorbeeld development. Aan hen kan je dan ook feedback vragen, en als dit positief is, is het dus geslaagd. Sales zou wel feedback kunnen geven over de opmaak maar niet over de inhoud. De reactie van klanten en developers zijn beter.

15.5.2 Bijlagen contextanalyse

Als bijlage van de probleemanalyse zijn de uitweringen van het 7s-model bijgevoegd.

15.5.2.1 Uitwerking 7s-model

Het 7s-model is uitgewerkt door middel van een interview met de business unit manager. Hieronder staat per dimensie beschreven wat deze inhoudt, en welke vragen er zijn gesteld om de dimensie te beantwoorden.
De uitleg van de verschillende dimensies is gebaseerd op het 7s-model van McKinsey [105]. De bijhorende vragen zijn van een managentmodellensite [106].

Strategie
De dimensie strategie (strategy) beschrijft de lange termijn doelen van de afdeling GRO-Cloud Solutions en de manier waarop deze doelen bereikt dienen te worden.

Wat is de strategie van GRO-Cloud Solutions?

- Marktpositie: Onze marktpositie helder krijgen binnen de noordelijke provincies. Op een hoger niveau bij IT en Business aan tafel komen.
- Waarde propositie: Duidelijk verwoorden intern en extern. Meer projectmatig-/dienst-/ opdracht-gericht werken.
- Goed werkgeverschap: Werknemerstevredenheidsonderzoek, acht van de tien collega's vinden ilionx een goede werkgever, werk om je vingers bij af te likken.
- Unit/vestiging-resultaat: o.a. 80% van de projecten moet als gevolg een beheercontract hebben, kwalitatieve groei, kwantitatieve groei.

Hoe is GRO-Cloud Solutions van plan deze doelstellingen te bereiken?
Door het investeren van tijd in de bovenstaande punten.

Hoe gaat GRO-Cloud Solutions om met de concurrentiedruk?
Uitgaan van eigen kracht en een nog betere kwaliteit leveren.

Systemen
De dimensie systemen (systems) omvat alle mogelijke systemen, zoals procedures, werkwijzen, voorschriften, communicatiestromen en afspraken binnen GRO-Cloud Solutions.

Hoe wordt er binnen GRO-Cloud Solutions gecommuniceerd?
Op een hele boel verschillende manieren. Vooral via Microsoft Teams. Hiernaast zijn er veel overleggen, zowel wekelijks met de teams als eenmaal in de vier weken met de hele unit. Ook zijn er lunchbijeenkomsten die klachtgericht zijn. Wij willen ook dat de lead consultants ook bij andere units praten. Na afloop van het gesprek willen wij de uitkomsten hiervan ook meteen in actie zetten.

Wat zijn belangrijke systemen die de organisatie gebruikt?
Microsoft Azure, MS Teams, Office365 en Microsoft 10.

Welke interne regels en processen gebruikt het team om op het goede spoor te blijven?
Voor deze regio hebben wij drie "leading principles" (regels), namelijk: wij gunnen elkaar wat, werken naast de klanten ook met elkaar samen en het moet een veilige omgeving zijn (iedereen moet zichzelf kunnen zijn).

Gedeelde waarden
De dimensie gedeelde waarden (shared values) is de verbinding tussen alle andere dimensies. De gedeelde waarden zijn waarden die centraal staan binnen GRO-Cloud Solutions. Dit zijn bijvoorbeeld: kernwaarden, bedrijfscultuur en bedrijfsvisie.

Wat zijn kernwaarden van GRO-Cloud Solutions?
- Samen;
- Doelgericht;
- Dichtbij;
- Bevlogen;
- Innovatief.

Hoe sterk zijn de kernwaarden?
Sterk, ze zijn door heel ilionx bekend.

Wat is de cultuur binnen GRO-Cloud Solutions?
Onze cultuur laat zich samenvatten als een no-nonsens cultuur waarin persoonlijke ontwikkeling, uitdagende opdrachten en innovatieve projecten centraal staan. Wij zijn van nature nieuwsgierig, vinden samen succes behalen belangrijk en hebben passie voor ons vak. Ondanks dat we bijzondere dingen steeds meer normaal vinden, zijn we trots op wat we met onze klanten bereiken en maken we graag het verschil. Onze lokale ilionx-vestigingen nemen een prominente plaats in de manier waarop we georganiseerd zijn. We ondernemen op het laagste niveau waarbij een collegiale sfeer en voor elkaar klaarstaan cruciale onderdelen zijn van ons werkplezier.

Wat zijn de fundamentele waarden waarmee GRO-Cloud Solutions is opgebouwd?
- Door onze persoonlijke en op samenwerking gerichte benadering, welke niet voor, maar samen met de klant ontstaande langdurige op vertrouwen gebaseerde relaties.
- Door onze lokale aanwezigheid bieden we de beste kennis dichtbij, kunnen we snel schakelen en zijn de lijnen met onze klanten en partners kort.
- We onderscheiden ons door de juiste combinatie van diepgaande technologie-kennis en relevante branche ervaring.
- Door sterke partnerships met onze technologiepartners beschikken we over kennis uit de bron en zijn we in staat om verschillende technologieën te integreren.

Personeel
Bij de dimensie personeel (staf) wordt een beeld gecreëerd over personeel en management. Er wordt gekeken hoe GRO-Cloud Solutions omgaat met het personeel.

Welke posities of specialisaties zijn vertegenwoordigd binnen GRO-Cloud Solutions?
De posities zijn consultants, engineers en beheer. Specialismen zijn Cloud, werkplek en packaging. Nieuw hierbij komt security.

Welke posities moeten worden ingevuld?
Het zit het meer in de aantallen denk ik. Er is te weinig kennis over package, storage en VMware. Wij dekken wel alles in kennis.

Structuur
De dimensie structuur (structure) gaat in op de structuur van GRO-Cloud Solutions, hiermee wordt onder meer gekeken naar de verdeling van taken, functies, bevoegdheden en verantwoordelijkheden. Verder wordt gekeken naar factoren zoals de structuur van de afdeling.

Hoe is GRO-Cloud Solutions verdeeld en hoe is de hiërarchie in elkaar?
Je hebt business unit managers, zij praten met de lead consultants en consultants. Op hun beurt praten zij met de engineers en zij praten weer met het beheer. De consultants geven adviezen. De engineers zijn bezig met het implementeren. Het beheer onderhoudt deze acties weer. De mensen met de verschillende taken moeten dus goed met elkaar kunnen overleggen.

Hoe worden de activiteiten gecoördineerd binnen GRO-Cloud Solutions?
Eerst wordt door pre-sales gekeken of de klant bij ilionx past. Een opdracht wordt binnengehaald, je gaat naar een consultant, die kijkt of hij er wat mee kan. Ondertussen zijn de business unit managers continu op de hoogte.

Wordt binnen GRO-Cloud Solutions besluitvorming centraal of decentraal genomen?
Het is een mix van centraal van decentraal; dit wordt ook wel ondernemerschap genoemd, neem je verantwoordelijkheid voor hetgeen wat je doet.

Stijl
De dimensie stijl (style) staat voor de vorm van management binnen GRO-Cloud Solutions. Onder management binnen het bedrijf vallen de handelingen van de manager die voor een bepaald gedrag zorgen bij de medewerkers.

Hoe meewerkend is de stijl van leidinggeven?
Erg meewerkend en flexibel. Het is belangrijk dat het personeel goed in hun vel zit en op zijn gemak is met de klanten die hij heeft. Wij zorgen er dan ook altijd voor dat dit zo is. Wanneer iets niet lukt wordt ook altijd hulp aangeboden, etc.

Hoe effectief is het leiderschap?
Dit is niet altijd heel effectief, wat komt door miscommunicatie.

Hebben medewerkers binnen GRO-Cloud Solutions de neiging competitief of coöperatief te zijn? Wat zijn daar de effecten van?
Coöperatief, maar toch is er altijd wel enige competitie. Dit zorgt ervoor dat het een heel goed team is. De effecten hiervan is dat medewerkers zich aan elkaar optrekken om betere prestaties te leveren.

Vaardigheden
Bij de dimensie vaardigheden (skills) wordt gekeken naar het onderscheidend vermogen van GRO-Cloud Solutions. Dit kunnen één of meerdere dingen zijn.

Wat zijn de sterkste vaardigheden vertegenwoordigd binnen GRO-Cloud Solutions?
Wij zijn goed in werkplek, Cloud en Packaging Service.

Wat staat GRO-Cloud Solutions bekend om?
Wij staan bekend om onze werkplek, VDI en packaging. De andere dingen van ons kent ilionx op andere locaties niet omdat wij niet veel reclame voor onszelf maken.

Hoe worden de vaardigheden gemonitord en beoordeeld?
Eén maal in de acht weken heb ik met iedereen een gesprek. Collega's bepalen zelf welke certificeringen en opleidingen zij willen doen, welke vaardigheden zij willen hebben.

15.5.3 Bijlagen literatuuronderzoek

Als bijlage van het literatuuronderzoek zijn de uitweringen van de zoekstrategie, welke informatie nodig is, waar deze informatie vandaan komt en hoe de informatie beoordeeld is, bijgevoegd.

15.5.3.1 Zoekstrategie

In dit hoofdstuk wordt de zoekstrategie uitgelegd. Aan de hand van deze zoekstrategie zijn de verschillende literatuurvragen beantwoord.

15.5.3.2 Wat voor informatie is nodig?

In dit literatuuronderzoek worden verschillende deelvragen beantwoord. In Tabel 23 staat per deelvraag beschreven welke vraag er beantwoord is en welke zoektermen hierbij gebruikt zijn.

Deelvraag	**Zoekterm Nederlands**	**Zoekterm Engels**
Wat wordt onder container applicatie verstaan?	Wat zijn containers? Wat zijn container applicaties? Wat zijn container applicatie architecturen?	What are containers? What are container applications? What are container application architectures?
Welke verschillende mogelijkheden zijn binnen Azure voor het hosten van een container applicatie?	Container hosting Microsoft Azure.	Container hosting Microsoft Azure.
Voorziet Azure in alle benodigde componenten?	Microsoft Azure Databases Microsoft Azure API hosting Microsoft Azure Service Bus	Microsoft e-commerce application GitHub
Wat zijn de kosten van deze verschillende oplossingen van Azure?	Wat zijn de kosten van container oplossing[14]?	Costs container solution[14]?
Wat is nodig voor het geautomatiseerd uitrollen van de infrastructuur binnen Azure?	Hoe werkt Azure Resource Manager? Wat is JSON? Wat is Azure DevOps?	How does Azure Resource Manger work? What is JSON? What is Azure DevOps?

Tabel 23: Gebruikte zoektermen

15.5.3.3 Waar komt de informatie vandaan?

De informatie is gevonden in verschillende informatiebronnen en databanken. In Tabel 24 staat een lijst met informatiebronnen en databanken die geraadpleegd zijn.

Informatiebronnen en databanken
Google.nl
Scholar.google.nl
Hbo-kennisbank.nl
Saxionbibliotheek.nl
Ieee.org

Tabel 24: Gebruikte informatiebronnen en databanken

[14] Container oplossing wordt in de vraag veranderd voor de te onderzoeken productnaam

15.5.3.4 Hoe is de informatie beoordeeld?

De informatie die is gevonden worden weerlegd met de volgende criteria:

- Is de gevonden informatie relevant?
- Wat is de betrouwbaarheid van de informatie?
- Is de gevonden informatie te verifiëren?
- Is de gevonden bron valide?

Verder werd bij een bron altijd de sneeuwbalmethode toegepast om zo nog meer informatie te vinden of om de informatie te bevestigen van de gevonden bron. De sneeuwbalmethode wordt uitgelegd in 15.4.1.2. Nadat de bron is gevalideerd, werd deze en zijn informatie opgenomen in het onderzoek.

16. Bijlage: methode ontwerpdocumentatie

Het ontwerponderzoek gaat ervan uit dat de oplossing wordt gevonden door het gebruik van een ontwerpbenadering en ontwikkelmethode. Naast de ontwerpbenadering en ontwikkelmethode worden er in het functioneel- en technisch ontwerp nog andere methoden gebruikt, welke hieronder worden toegelicht.

Ontwerpbenadering
Als ontwerpbenadering wordt “prototyping” gebruikt. Deze ontwerpbenaderingsmethode is gekozen omdat de methode uitgaat van continu overleg tussen stakeholders. Dit sluit goed aan bij de inhoud van de whitepaper in overleg moet met de verschillende stakeholders. De methode prototyping biedt verder de mogelijkheid om de systeem ontwikkelmethode “trial and error” te gebruiken [107].

Ontwikkelmethode
Voor het ontwikkelen van de oplossing, wordt er gebruik gemaakt van de “trial and error” methode. Trial and error wordt volgens de Van Dale gedefinieerd als: “methode waarbij je alle mogelijkheden probeert tot je op de oplossing stuit” [108]. Hiervoor is gekozen omdat er zo op een stapsgewijze manier de whitepaper, flyer en het PoC ontwikkeld kan worden. Bij het ontwikkelen van de twee producten zullen er steeds fouten naar voren komen, waarna deze moeten worden opgelost voordat er weer met nieuwe dingen kan worden gestart [107].

Multicriteria-analyse
Een multicriteria-analyse is een methode om uit meerdere oplossingen, de beste oplossing te kiezen voor een specifiek probleem [109]. In deze analyse worden meerdere container oplossingen met elkaar vergeleken. Door vergelijking van de verschillende mogelijkheden kan de keuze voor de beste oplossing worden gemaakt.

Deze analyse wordt dit als volgt toegepast (de context is al beschreven in de contextanalyse): de onderzochte informatie dat in het literatuuronderzoek is gevonden wordt per component beoordeeld en ingevuld in de analyse. Uiteindelijk kan uit deze analyse worden opgemaakt welke container hosting optie het beste past bij de verschillende container gebaseerde applicaties.

Procesmodulering
Voor de ontwikkeling van de workflows binnen het functioneel ontwerp wordt er gebruik gemaakt van procesmodelleren. Voor het procesmodelleren wordt gebruik gemaakt van Unified Modeling Language (UML). Door het gebruik van UML kunnen er beschrijvingen, statische en dynamische processen worden gemaakt. Eén van de UML technieken is een activiteitendiagram. Met deze techniek zijn de workflows gemaakt. Een activiteitendiagram laat de toestanden van het systeem zien gedurende het gebruik ervan en hoe de verschillende toestanden in elkaar overlopen. [110]