



Professionele Bachelor Toegepaste Informatica



## Infrastructure solutions en Docker containers

Jorn Lasker

Promotoren:

Ferdi Hoobergs  
Peter Vaes

Essec-BNS  
Hogeschool PXL Hasselt



---

Bachelorpaper Academiejaar 2017-2018





Professionele Bachelor Toegepaste Informatica



## Infrastructure solutions en Docker containers

Jorn Lasker

Promotoren:

Ferdi Hoobergs  
Peter Vaes

Essec-BNS  
Hogeschool PXL Hasselt



---

Bachelorpaper Academiejaar 2017-2018

## Dankwoord

Tijdens deze stage heb ik de kans gekregen om te werken met verschillende technologieën en infrastructuur.

Ik maak graag van deze gelegenheid gebruik om enkele mensen te bedanken die het mogelijk hebben gemaakt deze stage te volbrengen.

In de eerste plaats wil ik mijn bedrijfspromotor, Ferdi Hoobergs, bedanken om mij deze kans te geven mijn stage te volbrengen bij Essec-BNS. Hij heeft mij goed ondersteund bij de opdrachten en steeds mijn vragen beantwoord met veel enthousiasme.

Graag wil ik ook Peter Vaes, mijn hogeschoolpromotor, bedanken om mij telkens bij te staan bij het maken van dit eindwerk. Gedurende de stage kon ik steeds terecht bij hem met vragen over de benodigde documentatie. Hij heeft telkens mijn vragen vol enthousiasme beantwoord en mij goede tips gegeven om het eindwerk tot een goed einde te brengen.

Daarnaast wil ik de collega's van BNS bedanken voor de samenwerking. Als ik met vragen zat in verband met bepaalde opdrachten, kon ik altijd rekenen op hun expertise. Dankzij hen heb ik veel bijgeleerd op een korte periode.

Ook wil ik de lectoren van Hogeschool PXL bedanken die mij tijdens mijn opleiding steeds begeleid en ondersteund hebben.

## Abstract

Essec-BNS is ontstaan na het samenvoegen van twee bedrijven, namelijk Essec en Business Network Systems. Hierdoor is de groep een totaalaanbieder op vlak van ICT en de meest veelzijdige groep inzake telecom, data en alarmering. Essec-BNS helpt bedrijven met installatie en ondersteuning van hun infrastructuur en bedrijfstelecommunicatie. Het bedrijf is gericht op een maximaal rendement en kostenefficiëntie.

Tijdens de stage komen er verschillende opdrachten aan bod. Het is de bedoeling in contact te komen met verschillende technologieën en hardware maar vooral ook de verschillende services van Essec-BNS te leren kennen. De meest voorkomende opdracht is het configureren en installeren van infrastructuur zoals laptops, desktops en servers. Deze worden voor bedrijfsklanten klaargemaakt door deze te configureren met de benodigde software zoals Windows 10, AV Defender, Office en enkele andere programma's. Voor het virtualiseren van servers wordt VMware ESXi gebruikt. Dit is virtualisatie software die op bare-metal servers gedraaid kan worden. Bepaalde servers moeten nog geassembleerd worden. Extra RAM of CPU komen dus ook aan bod bij deze assemblage. Naast deze opdrachten is het bemannen van de helpdesk ook een groot onderdeel geweest tijdens de stage. Via de helpdesk worden klanten geholpen met hun problemen in verband met infrastructuur en software.

Voor de onderzoeksopdracht wordt onderzocht of Docker-containers een businessopportunity zijn voor de Essec-groep. Om een conclusie te kunnen formuleren, wordt eerst onderzocht wat Docker is. Deze software zorgt ervoor dat applicaties in een container gestopt kunnen worden. De Docker-software wordt op een duidelijke manier omschreven evenals de voor- en nadelen van containers. Docker is niet de enige aanbieder voor containersoftware. Er bestaan alternatieven voor Docker, zoals rkt, die toegelicht zullen worden. Omdat Docker-containers de afgelopen jaren zo populair geworden zijn, heeft Microsoft een eigen soort container gemaakt. Deze containers noemt men Windows Server Containers en worden dan ook beschreven in deze opdracht. Deze containers werken op het Windows-OS en zijn niet helemaal hetzelfde als de Docker-containers. Nadat de Docker-software toegelicht is, wordt een SWOT-analyse over Essec-BNS gemaakt om na te gaan wat de opportuniteiten zijn. Deze analyse brengt meer duidelijkheid over de interne en externe factoren die invloed kunnen hebben op de keuze voor het al dan niet in gebruik nemen van Docker-containers. Samen met deze SWOT-analyse wordt een ROI-analyse opgesteld om te kijken of het financieel een goede oplossing zou zijn om Docker-containers te implementeren in het bedrijfsproces.

## Inhoudsopgave

Dankwoord .....	ii
Abstract .....	iii
Inhoudsopgave .....	iv
Lijst van gebruikte figuren .....	vii
Lijst van gebruikte tabellen .....	ix
Lijst van gebruikte afkortingen.....	x
Inleiding .....	1
I. Stageverslag.....	2
1 Bedrijfsvoorstelling.....	2
1.1 Situering .....	2
1.2 Partners .....	3
1.3 Organigram.....	4
1.4 Unique sellingpoints.....	4
1.4.1 Infrastructure Solutions.....	5
1.5 Bedrijfsgegevens .....	5
2 Stageopdracht .....	6
2.1 Probleemstelling.....	6
2.2 Doelstellingen.....	6
2.3 Technologieën en hardware.....	6
2.3.1 Sharepoint .....	7
2.3.2 VMware ESXi .....	7
2.3.3 vSphere Client.....	8
2.3.4 Windows 10 Media Creation Tool.....	8
2.3.5 Solarwinds N-Central.....	8
2.3.6 Citrix Receiver.....	9
2.3.7 Watchguard .....	9
2.3.8 VMware Airwatch.....	9
2.3.9 Remote Desktop Manager .....	10
2.3.10 Veeam Backup and replication.....	10
3 Uitwerking stageopdracht.....	11
3.1 Aanpak.....	11
3.1.1 Informatie over klant verzamelen .....	11
3.1.1.1 Sharepoint .....	11
3.1.1.2 Mondeling .....	11

3.1.2	Hardware-informatie verzamelen .....	11
3.1.2.1	Focuscenter .....	12
3.1.3	Software-informatie verzamelen .....	12
3.1.4	Implementatie .....	12
3.2	Resultaat.....	12
4	Besluit.....	13
4.1	Persoonlijke reflectie.....	13
II.	Onderzoekstopic.....	15
1	Onderzoeksvraag.....	15
2	Methode van onderzoek .....	15
3	Resultaten.....	15
3.1	Literatuurstudie.....	16
3.1.1	On-Premise versus Cloud computing .....	16
3.1.1.1	On-Premise .....	16
3.1.1.2	Cloud computing .....	16
3.1.2	Docker.....	19
3.1.2.1	Wat is Docker?.....	19
3.1.2.2	Wat zijn containers? .....	19
3.1.2.3	Container versus virtuele machine.....	20
3.1.3	Docker alternatieven .....	21
3.1.3.1	CoreOS rkt framework.....	21
3.1.3.2	LXC/LXD .....	22
3.1.3.3	OpenVZ/Virtuozzo .....	23
3.1.3.4	Vergelijkingstabel .....	23
3.1.4	Beheertools voor containers .....	23
3.1.4.1	Portainer.....	24
3.1.4.2	Shipyards.....	27
3.1.4.3	Rancher.....	29
3.1.4.4	Kitematic.....	31
3.1.4.5	Docker Datacenter.....	32
3.1.4.6	Vergelijking beheertools.....	37
3.1.4.7	Conclusie beheertools .....	38
3.1.5	Verskil tussen Linux en Windows containers .....	38
3.1.6	Integratie van shared storage (Nimble) .....	40
3.1.7	Containers in de cloud.....	40
3.1.7.1	Amazon Web Services (AWS) .....	40

3.1.7.2	Google Cloud .....	42
3.1.7.3	Microsoft Azure .....	43
3.1.7.4	Prijsvergelijking cloudproviders .....	44
3.1.8	Containers migreren naar de cloud.....	45
3.1.9	Back-up, Disaster Recovery en High Availability .....	48
3.1.10	SWOT .....	51
3.1.10.1	Wat is een SWOT-analyse?.....	51
3.1.10.2	Wat is het nut? .....	52
3.1.10.3	SWOT-analyse Essec-BNS .....	52
3.1.11	ROI .....	52
3.1.11.1	Wat is ROI? .....	52
3.1.11.2	ROI berekenen.....	53
4	Conclusie .....	54
4.1	Bespreking resultaten.....	54
4.2	Aanbevelingen.....	54
4.3	Persoonlijke reflectie.....	55
	Bibliografie .....	56



## Lijst van gebruikte figuren

Figuur 1 Gebouw Essec.....	2
Figuur 2 Sharepoint portal .....	7
Figuur 3 Werking Solarwinds N-Central .....	9
Figuur 4 VM vs container.....	20
Figuur 5 rkt-logo .....	21
Figuur 6 LXC-logo.....	22
Figuur 7 LXC-webpanel.....	22
Figuur 8 Logo Virtuozzo.....	23
Figuur 9 Swarm.....	24
Figuur 10 Logo Portainer .....	24
Figuur 11 Portainer dashboard.....	25
Figuur 12 Portainer templates .....	25
Figuur 13 Portainer containers.....	26
Figuur 14 Portainer Swarm.....	27
Figuur 15 Logo Shipyard.....	27
Figuur 16 Shipyard containers.....	27
Figuur 17 Shipyard info containers.....	28
Figuur 18 Shipyard nodes .....	28
Figuur 19 Logo Rancher.....	29
Figuur 20 Rancher stacks.....	29
Figuur 21 Rancher hosts.....	30
Figuur 22 Rancher containers.....	30
Figuur 23 Rancher registry .....	31
Figuur 24 Logo Kitematic.....	31
Figuur 25 Kitematic containers.....	31
Figuur 26 Kitematic Docker Hub.....	31
Figuur 27 Logo Docker Datacenter.....	32
Figuur 28 DDC dashboard.....	33
Figuur 29 DDC services .....	33
Figuur 30 DDC containers.....	34
Figuur 31 DDC nodes .....	35
Figuur 32 DDC DTR .....	36
Figuur 33 DTR .....	36
Figuur 34 DTR settings.....	37
Figuur 35 Windows Container vs Hyper-V container .....	39
Figuur 36 Logo Nimble.....	40
Figuur 37 Logo AWS .....	40
Figuur 38 AWS EC2 architectuur .....	41
Figuur 39 Logo Google Cloud .....	42
Figuur 40 GKE architectuur .....	43
Figuur 41 Logo Microsoft Azure .....	43
Figuur 42 Azure architectuur.....	44
Figuur 43 Host-based storage Docker .....	46
Figuur 44 Host-based shared storage Docker .....	46
Figuur 45 multi-host shared storage Docker.....	47
Figuur 46 Morpheus dashboard.....	49

Figuur 47 Morpheus Registry .....	49
Figuur 48 High Availability UCP .....	51
Figuur 49 High Availability DTR .....	51

## Lijst van gebruikte tabellen

Tabel 2 Voor- en nadelen On-Premise .....	16
Tabel 3 Voor- en nadelen Cloud Computing .....	19
Tabel 4 Voor- en nadelen Docker-containers.....	21
Tabel 5 Vergelijking containersoftware .....	23
Tabel 6 Vergelijking van beheertools .....	37
Tabel 7 Verschil Linux- en Windows-containers .....	38
Tabel 8 Verschil Docker-containers op Linux en Windows .....	39
Tabel 9 Prijsvergelijking cloudproviders.....	45
Tabel 10 SWOT-analyse Essec-BNS .....	52

## Lijst van gebruikte afkortingen

Afkorting	Betekenis
BNS	Business Network Systems
IP	Internet Protocol
IT	Information Technology
ICT	Information and Communication Technology
ROI	Return On Investment
SWOT	Strengths, weaknesses, opportunities, threats
AWS	Amazon Web Services
IaaS	Infrastructure-as-a-Service
PaaS	Platform-as-a-Service
SaaS	Software-as-a-Service
CaaS	Container-as-a-Service
CI	Continuous Integration
OS	Operating System
VM	Virtual Machine
CPU	Central Processing Unit
I/O	Input/Output
UID	User Identifier
VPS	Virtual Private Server
VE	Virtual Environment
CLI	Command Line Interface
UCP	Universal Control Pane
DTR	Docker Trusted Registry
LDAP	Lightweight Directory Access Protocol
AD	Active Directory
RBAC	Role-Based Access Control
API	Application Programming Interface
VPC	Virtual Private Cloud
SRE	Site Reliability Engineers
HA	High Availability
DR	Disaster Recovery

*Tabel 1 Lijst met gebruikte afkortingen*

## Inleiding

Iedere dag in een servicebedrijf brengt nieuwe uitdagingen met zich mee. Problemen in verband met infrastructuur zullen er altijd zijn en moeten op een structurele en adequate manier opgelost worden evenals het installeren en configureren van deze infrastructuur. De essentie van de stageopdracht is dan ook omgaan met deze problemen op een professionele manier en zoeken naar een duidelijke oplossing.

Dit eindwerk is opgedeeld in twee onderdelen: de stageopdracht en de onderzoeksopdracht. De stageopdracht was niet één bepaalde opdracht, maar diverse opdrachten om zo te leren hoe een correcte oplossing tot stand komt en hoe het eraan toe gaat in een IT-bedrijf. Om deze diverse opdrachten tot een goed einde te brengen, is het gebruik van verschillende technologieën en methodologieën essentieel. Naast het operationele deel voorzie ik ook tijd om een aantal installaties in verband met container technologie op te bouwen, die op langere termijn de Essec-groep moeten helpen om nog efficiënter om te gaan met softwareproblemen. De onderzoeksvraag luidt: “Zijn er *business opportunities* voor de Essec Group door gebruik te maken van Docker-containers?”.

In het eerste deel van het eindwerk wordt het stagebedrijf voorgesteld. Daarna volgt de toelichting van de stageopdracht aan de hand van een probleemstelling en de doelstellingen van deze opdracht. Vervolgens zullen de aanpak en de resultaten toegelicht worden.

In het tweede deel wordt de onderzoeksopdracht toegelicht aan de hand van de onderzoeksvraag. Een literatuurstudie zal meer informatie geven over Docker. Een SWOT-analyse en ROI-analyse zullen zorgen voor een resultaat en uiteindelijk een conclusie.

# I. Stageverslag

## 1 Bedrijfsvoorstelling

Essec-BNS is ontstaan na het samenvoegen van twee bedrijven, namelijk Essec NV en Business Network Systems. Hierdoor is de groep een totaalaanbieder op vlak van ICT en de meest veelzijdige groep inzake telecom, data en alarmering. In 2005 werd een nieuw bedrijf opgestart binnen de Essec-groep, namelijk IndigoCare. IndigoCare ontwierp het eerste oproepsysteem voor verplegers op basis van IP. In 2017 ontstond een samenwerking tussen Essec-BNS en MediaWave, een bedrijf dat doet aan *digital signage*. Naast de samenwerking met MediaWave is er later in het jaar 2017 nog een samenwerking ontstaan tussen Essec-BNS en HostYou. Dit is een bedrijf dat hosting en domeinregistratie aanbiedt. [1] [2]

Essec-BNS helpt bedrijven met installatie en ondersteuning van hun infrastructuur en bedrijfstelecommunicatie. Het bedrijf is gericht op een maximaal rendement en kostenefficiëntie. Daarnaast bieden ze ook totaaloplossingen aan voor oproepsystemen voor verplegers en verpleegsters.

BNS is al jaren een professionele partner bij wie bedrijven terecht kunnen voor ondersteuning voor het beheren van bestaande systemen of voor advies in verband met nieuwe oplossingen. BNS wil de klant laten genieten van een balans tussen zekerheid en flexibiliteit. Dit doen ze door te kiezen voor een hybride IT-oplossing. De klant kan kiezen waar applicaties gehost worden, dit kan in hun eigen datacenter, in het BNS datacenter of in wereldwijde datacenters. [3]

### 1.1 Situering

BNS bestaat uit twee afdelingen, namelijk 'Infrastructure solutions' en 'Business solutions'. 'Infrastructure solutions' biedt een hele waaier van diensten aan. Ze staan vooral in voor het installeren en configureren van de infrastructuur van de klant en support bieden aan klanten. Hiermee willen ze zich profileren als het verlengstuk van de interne IT-afdeling van een bedrijf en de beste netwerkoplossing aanbieden. Dit aan de hand van de behoeften en het budget van de klant. 'Business solutions' assisteert klanten bij het implementeren van software. Deze afdeling biedt een uitgebreid pakket aan diensten aan zoals analyses maken, consultancy, projectleiding, enzovoort. Samen met de mensen van BNS zal ik mijn stageopdracht volbrengen. Zij ondersteunen mij bij het uitgroeien tot een professioneel IT'er. [4] [5]

Het bedrijf is gelegen op een industrieterrein, namelijk Ravenshout, in de Schoebroekstraat 48, 3583 Beringen. Het is een gebouw vlak naast de E313 en is makkelijk herkenbaar van op afstand.



Figuur 1 Gebouw Essec

## 1.2 Partners

### Microsoft

Een van de belangrijkste softwarepartners van Essec-BNS is Microsoft. Hiermee bewijst Essec-BNS dat ze de vereiste kennis van Microsoft producten hebben. Door dit certificaat genieten ze van een breed pakket aan voordelen waaronder training en directe ondersteuning van Microsoft. Hiermee wordt de dienstverlening naar klanten toe verbeterd. [6]

### HP

De belangrijkste hardware partner is HP. Opnieuw toont Essec-BNS hiermee aan dat ze over de nodige kennis beschikken in verband met HP producten. [7]

### Citrix

Omdat Essec-BNS Citrix als partner heeft, zijn ze in staat om klanten beter van dienst te zijn door altijd op de hoogte te blijven van nieuwe ontwikkelingen. [8]

### Veeam

Essec-BNS maakt vooral gebruik van Veeam om hun Back-up-as-a-service aan te bieden. [9]

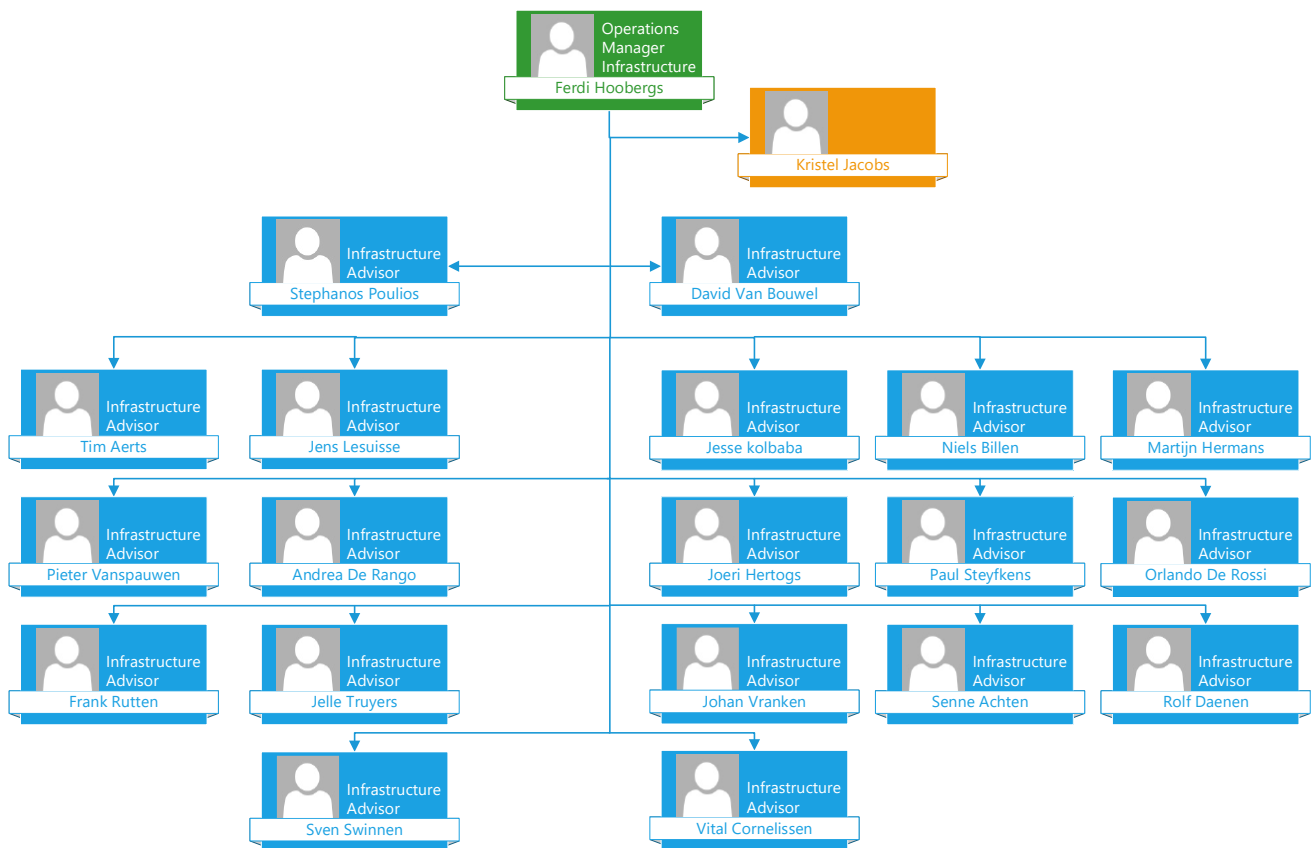
### VMware

Voor BNS is virtualisatie een stap richting de toekomst. De technologieën van Veeam bieden de zekerheid en veiligheid van stabiele virtuele infrastructures. [10]

### Watchguard

Dankzij Watchguard kan BNS verschillende producten leveren die het bedrijfsnetwerk kunnen verbeteren. Ze zijn dan ook Watchguard Gold Partner. [11]

## 1.3 Organigram



## 1.4 Unique sellingpoints

De Essec Group bestaat uit een groep bedrijven die elk een eigen specialisatie hebben. Zo is BNS gespecialiseerd in infrastructuur opzetten en onderhouden van een bedrijf. Dit gaat van het installeren van nieuwe infrastructuur tot het beheren van bestaande infrastructuur. BNS biedt een hybride oplossing aan waarmee een klant zelf kan kiezen waar bepaalde applicaties gehost worden. Dit wil zeggen dat een applicatie in het eigen datacenter gehost kan worden, maar ook in het datacenter van BNS of in een publieke cloud. Essec Telecom Center is gespecialiseerd in telefonie. Ze bieden een oplossing voor de interne telefonie. Zo is Essec-BNS een totaalaanbieder van oplossingen in verband met interne communicatie.

IndigoCare is gespecialiseerd in oproepsystemen voor verplegers. Zij hebben iCall ontwikkeld. Dit was het eerste IP-based oproepsysteem voor verplegers. IndigoCare biedt dus een oplossing voor ziekenhuizen, bejaardentehuizen en woonzorgcentra om verplegers, verpleegsters en patiënten op een efficiënte manier te helpen bij hun taken. [12]

HostYou is de meest recente aanwinst van de Essec Group. Dit bedrijf biedt oplossingen in verband met domeinnamen, webhosting en mailhosting aan. Hier kan een klant domeinnamen aankopen, websites en e-mail laten hosten. [13]

Met al deze gespecialiseerde bedrijven in de groep wil de Essec Group een totaalaanbieder zijn van ICT-oplossingen. Hierdoor ontstaat de meest veelzijdige groep inzake telecom, data en alarmering.



### 1.4.1 Infrastructure Solutions

Binnen BNS is de afdeling infrastructure solutions verantwoordelijk voor de infrastructuur van een klant. Ze zorgen voor nieuwe oplossingen bij de klant, maar ondersteunen ook klanten met hun bestaande systemen. De ondersteuning gebeurt via een helpdesk die BNS aanbiedt aan klanten, maar ook door outsourcing van werknemers. Zij ondersteunen de interne IT-afdeling van een klant om zo deze afdeling te versterken. Business Solutions assisteert bedrijven door nieuwe software te ontwikkelen en deze software te installeren. Om bedrijven zo goed mogelijk te assisteren, biedt Essec-BNS een aantal diensten aan. Ze zullen steeds trachten zoveel mogelijk te realiseren met standaard software en daar waar nodig kan men de “gaten” invullen met maatwerk.

BNS biedt een datacenter ter beschikking voor klanten die hun applicaties in de cloud willen hosten. Wanneer een klant liever zijn applicatie in het eigen datacenter, maakt BNS infrastructuur klaar om deze applicatie te hosten. Ook bieden ze VPS aan om de omgeving van klanten volledig te scheiden van andere omgevingen.

Back-ups zijn heel belangrijk om data te behouden en te restoren in geval van dataverlies. Essec-BNS biedt vanuit het eigen datacenter een back-up-oplossing voor grote hoeveelheden data.

Met alle diensten die BNS aanbiedt, willen ze zich profileren als een verlangstuk van de interne IT-afdeling van een bedrijf of als vervanger wanneer er geen IT-afdeling aanwezig is. Hierdoor kunnen ze proactief meedenken om zo tot een oplossing te komen die aan de behoefte van de klant voldoet en die binnen een bepaald budget valt.

### 1.5 Bedrijfsgegevens

Eind 2014 fuseerde Essec Telecom Center en Business Network Systems tot Essec-BNS. Hierdoor werden ze een totaalaanbieder inzake telecommunicatie en netwerken. Essec Telecom Systems bestaat uit ongeveer 32 werknemers die gespecialiseerd zijn op vlak van telecommunicatie. Business Network Systems bestaat uit ongeveer 30 werknemers. In totaal werken bij de Essec Group ongeveer 130 mensen. In 2016 behaalde men een winst van 757.000 euro. [14]

## 2 Stageopdracht

Dit hoofdstuk stelt de stageopdracht voor. Eerst volgt een korte samenvatting over het probleem of de problemen, namelijk de probleemstelling. Daarna zullen de doelstellingen van de stageopdracht uitgelegd worden, gevolgd door de gebruikte technologieën tijdens de stage. Deze technologieën worden kort samengevat om een idee te krijgen waarvoor ze gebruikt worden.

### 2.1 Probleemstelling

Iedere dag in een servicebedrijf brengt nieuwe uitdagingen met zich mee. Dagelijks ontstaan nieuwe problemen waarvoor een structurele oplossing gevonden moet worden. Een adequate aanpak is hiervoor van groot belang.

Business Network Systems is een bedrijf dat een waaier van diensten aanbiedt. Ze helpen met de installatie en configuratie van de infrastructuur en ondersteunen klanten wanneer ze problemen hebben met hun infrastructuur. Hiervoor biedt BNS een helpdesk aan die dagelijks de problemen van de klanten oplost in verband met hun infrastructuur.

De installatie van nieuwe infrastructuur kan tijdrovend zijn wanneer men niet de juiste expertise bezit. BNS helpt klanten met hun infrastructuur door deze te installeren en configureren. Hierdoor is de infrastructuur gebruiksklaar en heeft BNS een goed overzicht over de systemen van hun klanten.

De stageopdracht is dus een brede opdracht waar meerdere probleemstellingen aan gekoppeld kunnen worden. Het is in het belang van de klant dat deze problemen worden opgelost en daarom zijn de klanten de stakeholders van deze verschillende opdrachten.

### 2.2 Doelstellingen

Het doel van de stage is meewerken binnen een IT-bedrijf. Tijdens de stage worden verschillende opdrachten voorzien waardoor alle aspecten van het IT-bedrijf aan bod komen. Door deze opdrachten uit te voeren, is het de bedoeling meer inzicht te krijgen hoe het eraan toe gaat in het bedrijf en leren werken met verschillende technologieën. Dankzij deze manier van werken, kom ik met verschillende technologieën en verschillende soorten hardware in contact.

Aan het einde van de stage zijn verschillende technologieën, hardware en methodologieën gekend waardoor ik een betere kijk krijg op hoe een servicebedrijf functioneert. Deze ervaring kan later gebruikt worden in het verdere leven als IT'er.

### 2.3 Technologieën en hardware

Tijdens de stage voer ik verschillende opdrachten uit en kom ik in contact met verschillende technologieën en verschillende soorten hardware.

Informatie is een belangrijk onderdeel voor de opdracht. Daarom wordt gebruik gemaakt van een platform, namelijk Sharepoint om informatie over klanten te delen met collega's. Dit platform bevat alle informatie over alle klanten van Essec-BNS. Dankzij dit platform kan nieuwe infrastructuur correct geïnstalleerd en geconfigureerd worden en kunnen problemen op een adequate manier aangepakt worden.

Voor het virtualiseren van servers wordt VMware ESXi gebruikt. Dit is de tegenhanger van Microsoft Hyper-V. Virtuele machines worden aangemaakt via de webclient van VMware of via de lokale client, namelijk 'vSphere Client'. Om laptops opnieuw te installeren, wordt gebruik gemaakt van de Windows 10 Media Creation Tool. Om deze laptops te beschermen tegen virussen en malware,

gebruikt Essec-BNS “Security Manager” van Solarwinds N-Central. Met deze applicatie kan ook hardware gescand worden. Citrix-servers worden gebruikt om klanten een virtuele desktop of virtuele applicaties aan te bieden. Back-ups zijn belangrijk om data te behouden. Daarom is Essec-BNS partner van Veeam. Deze software biedt gerichte back-up en *recovery*-mogelijkheden aan.

Klanten kopen nieuwe hardware die op voorhand geconfigureerd moet worden. Hierdoor is het contact met verschillende soorten hardware zoals laptops, desktops, switchen, firewalls, NAS, printers en servers, onvermijdelijk. De laptops, desktops, switchen en servers zijn meestal van het merk HP en als firewall gebruikt men Watchguard.

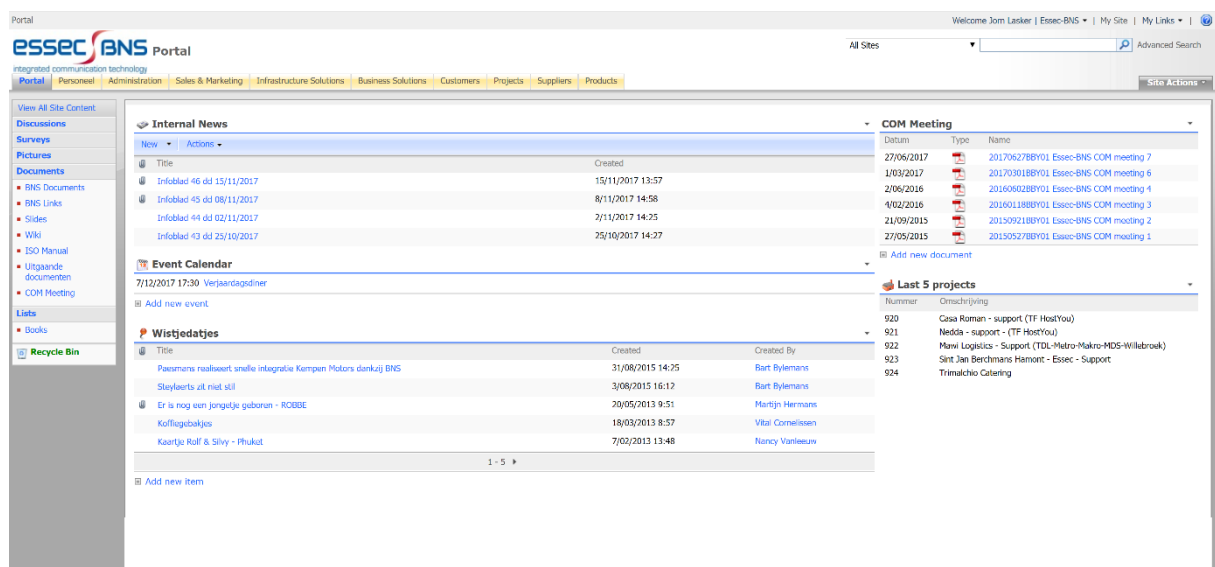
De werknemers van BNS worden ingezet bij klanten om daar de interne IT-afdeling te ondersteunen. Om alles in goede banen te leiden, maakt men gebruik van een planning die online beschikbaar is. In deze planning staat welke werknemer waar naartoe moet. Wanneer klanten problemen hebben en een ticket aanmaken, worden deze ook beschikbaar in deze tool.

De gebruikte technologieën en hardware worden hieronder besproken.

### 2.3.1 Sharepoint

Sharepoint is een platform van Microsoft voor het opzetten van een website waarop informatie gedeeld kan worden en om online samen te werken binnen een bedrijf. Een belangrijk concept bij Sharepoint is dat van de bibliotheken. In deze bibliotheken kunnen documenten worden opgeslagen en beschikbaar gesteld voor andere werknemers binnen een organisatie. Niet alleen documenten kunnen gedeeld worden, ook bepaalde fora, enquêtes, taken en agenda’s kunnen gedeeld worden.

Hieronder een voorbeeld van de startpagina van de portal van Essec-BNS.



Figuur 2 Sharepoint portal

### 2.3.2 VMware ESXi

VMware ESXi is een hypervisor die geïnstalleerd kan worden op bare-metal servers. Dit betekent dat deze hypervisor direct op de hardware geïnstalleerd kan worden zonder dat een onderliggend OS nodig is. ESX integreert zijn eigen kernel en gebruikt deze kernel. Deze aangepaste kernel begint te werken wanneer de Linux kernel de rest van de hardware opstart.

De vmkernel communiceert rechtstreeks met CPU en geheugen. Hiervoor gebruikt het een *scan-before-execute* (SBE) om speciale instructies in verband met CPU te behandelen en de *system resource allocation table* (SRAT) om toegewezen geheugen te vinden. Andere hardware wordt aangesproken via modules. Sommige van deze modules stammen af van de Linux-kernel. [15]

## Deployment

VMware ESXi kan op verschillende manieren geïnstalleerd worden. Zo kan het via het installatiebestand, via scripts of door gebruik te maken van een PXE boot. Deze scripts kunnen taken uitvoeren op de host zoals de configuratie van verschillende netwerken. ESXi kan geïnstalleerd worden via een bootable USB-stick. Deze USB-stick wordt aangesloten op de server en deze kan dan booten van de USB-stick. [16]

### 2.3.3 vSphere Client

Met deze client kunnen virtuele machines beheerd worden op de hypervisor die op de server draait. Via de vSphere Client kunnen virtuele machines aangemaakt, verwijderd, gemonitord en aangepast worden.

### 2.3.4 Windows 10 Media Creation Tool

Dit is een tool ontwikkeld door Microsoft dat ervoor zorgt dat een Windows ISO onmiddellijk gedownload kan worden zonder dat een *product key* nodig is. Deze ISO kan dan gebruikt worden voor het herinstalleren van computers of om te upgraden naar Windows 10.

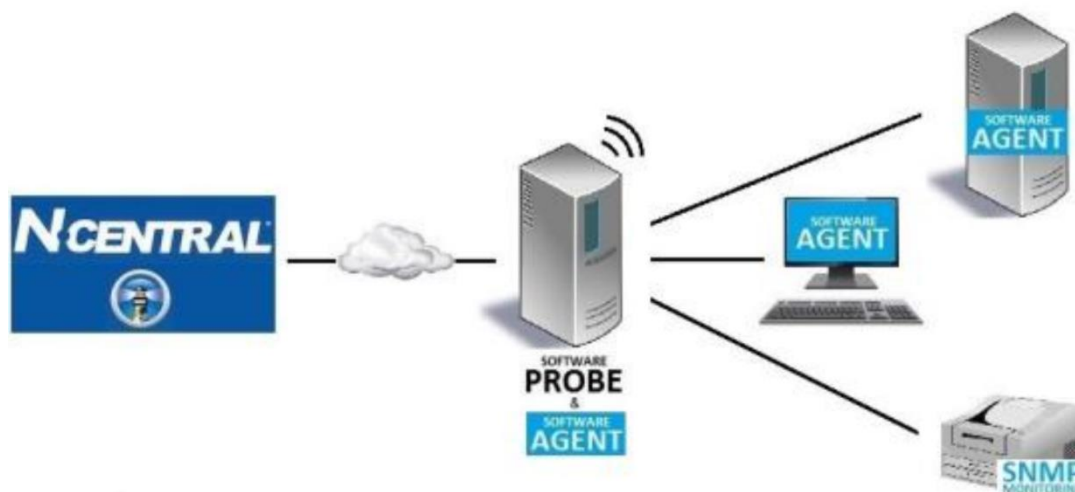
Deze tool bevat bestandformaten die geoptimaliseerd zijn voor downloadsnelheid, opties om USB's aan te maken en conversie naar het ISO-formaat. [17]

### 2.3.5 Solarwinds N-Central

Solarwinds N-Central is een platform waarop infrastructuur beheerd kan worden. *Agents* en *probes* zorgen dat informatie binnenkomt op het platform waardoor de systemen en apparatuur gemonitord kunnen worden en indien nodig aangepast worden. Enkele *features* van dit platform zijn het monitoren van Windows, Mac, Linux, VMware en netwerktoestellen, patchen uitvoeren, de installatie van een antivirus, back-up en herstel, remote controle,...

#### Architectuur en werking

Om toestellen te monitoren met Solarwinds N-Central wordt gebruik gemaakt van *agents* en *probes*. Deze *probes* worden meestal geïnstalleerd op een Windows-server en zijn verantwoordelijk voor de netwerkinfrastructuur zoals routers, firewall, switchen en printers. De *agents* zijn kleine programma's die geïnstalleerd moeten worden op toestellen van klanten.



Figuur 3 Werking Solarwinds N-Central

In een domein kan bepaalde software gedownload worden voor een Solarwind N-Central-*probe*. Deze software kan dan geïnstalleerd worden op een server in de omgeving van de klant en een *discovery* initialiseren om zo de toestellen binnen het AD-domein te importeren. Wanneer deze toestellen geïmporteerd zijn, kan de *agent* automatisch uitgerold worden naar de Windows- en Mac-systemen. Andere toestellen zoals printers en netwerktoestellen kunnen ook geïmporteerd worden om hierover informatie op te slaan.

### 2.3.6 Citrix Receiver

Citrix Receiver is een product gebundeld met Citrix Xenapp en XenDesktop. Hiermee kunnen gebruikers connecteren met verschillende virtualisatieservices van Citrix. Receiver wordt vooral gebruikt om gebruikers te connecteren met XenDesktop en XenApp desktops en applicaties, maar het kan ook applicaties afleveren via Microsoft App-V, links naar websites doorgeven en individuele documenten beschikbaar stellen.

Receiver kan gebruikt worden met Citrix StoreFront waardoor gebruikers toegang hebben tot eigen persoonlijke applicaties. Dit kan allemaal met een normale user interface op eender welk apparaat of OS.

Citrix Receiver connecteert de systemen van gebruikers met XenApp- of XenDesktop-omgevingen via het HDX-protocol.

### 2.3.7 Watchguard

De firewall-oplossing die gebruikt wordt bij Essec-BNS is van Watchguard. Watchguard is al meer dan twintig jaar een producent van *cybersecurity*-technologie. Ze leveren oplossingen die makkelijk te beheren en uit te rollen zijn. Watchguard is leider in netwerkbeveiliging, *secure Wi-Fi* en *network intelligence*-producten en services. Hiermee beschermen ze meer dan 80000 kleine tot middelgrote bedrijven rond de hele wereld. [18]

### 2.3.8 VMware Airwatch

VMware Airwatch is een technologie die IT-administrators helpt om mobiele toestellen, applicaties en data uit te rollen, de beveiliging ervan verzorgen en het onderhoud van deze toestellen in handen nemen. Er wordt een applicatie rechtstreeks op het *end device* geïnstalleerd, de Airwatch Agent, maar ook *backend*-software waardoor systeembeheerders *policies* kunnen aanmaken en uitrollen.

Het belangrijkste product van Airwatch is het enterprise mobility management software. Deze software bestaat uit verschillende features:

- MDM: Mobile Device management → voor de configuratie, beveiliging en installatie van policies op smartphones, tablets en PC's.
- MAM: Mobile application management → voor de installatie van policies en de beveiliging op smartphones en tablets.

De EMM-suite is beschikbaar voor on-premise deployment of als cloudservice.

### Werking Airwatch MDM

Wanneer een systeembeheerder gebruik wil maken van VMware Airwatch MDM moet hij op elk toestel dat hij wil beheren een applicatie installeren. Deze applicatie, AirWatch Agent, zorgt ervoor dat de toestellen en de MDM-server met elkaar kunnen communiceren. Wanneer de applicatie geïnstalleerd is en de gebruiker zich heeft geauthentiseerd (*enrollment*), kan de systeembeheerder, met behulp van de *agent*, de verschillende toestellen beheren. [19]

### 2.3.9 Remote Desktop Manager

Dankzij Remote Desktop Manager kunnen connecties, wachtwoorden en *credentials* veilig gedeeld worden tussen verschillende gebruikers op één platform. Met deze software kunnen verschillende type connecties gemaakt worden zoals RDP, SSH, VNC, Telnet en meer. Ook kunnen wachtwoorden opgeslagen en beheerd worden, waarna deze gedeeld kunnen worden met andere leden van een team. [20]

### 2.3.10 Veeam Backup and replication

Deze software wordt gebruikt om back-ups te maken. Veeam biedt een oplossing aan voor VMware vSphere-omgevingen en Hyper-V-omgevingen en dit zonder gebruik te maken van *agents*. Met Veeam Backup and Recovery kan een gefaalde virtuele machine binnen twee minuten terug werken. Ook bestanden en mappen kunnen terug gehaald worden.

Met de replicatie-*features* kunnen virtuele machines gerepliceerd worden *on-site* voor *high availability* en *off-site* voor *disaster recovery*. Ze bieden een cloudoplossing om replicaties *off-site* op te slaan zodat er geen hoge kosten zijn voor een *disaster recovery* site.

Replica's van virtuele machines kunnen getest en geverifieerd worden of deze nog terug te halen zijn of niet. [21]

## 3 Uitwerking stageopdracht

In het derde hoofdstuk wordt de aanpak van de verschillende opdrachten toegelicht. Vervolgens worden de resultaten beschreven.

### 3.1 Aanpak

De stage bestaat uit verschillende opdrachten die samen één doel hebben: meedraaien als systeem- en netwerkbeheerder binnen het operationele team van Essec-BNS. Tijdens deze stage heb ik vooral infrastructuur geconfigureerd en geïnstalleerd. Voordat deze infrastructuur geïnstalleerd en geconfigureerd kan worden, is het belangrijk om eerst achtergrondinformatie op te zoeken. Deze informatie kan van groot belang zijn bij de installatie van bepaalde software op de systemen.

Omdat de stage uit verschillende opdrachten bestaat, is er niet een bepaalde aanpak die gevolgd wordt voor alle opdrachten.

#### 3.1.1 Informatie over klant verzamelen

##### 3.1.1.1 SharePoint

Sharepoint is een platform van Microsoft dat dient als raamwerk om informatie te delen via een website en de samenwerking binnen een bedrijf makkelijker maakt. Een belangrijk concept bij Sharepoint is dat van de bibliotheken. In deze bibliotheken worden documenten opgeslagen en beschikbaar gesteld voor andere werknemers binnen een organisatie. Naast documenten is het ook mogelijk om fora, enquêtes, taken en agenda's te delen.

Bovendien biedt Essec-BNS dit portaal aan om informatie met betrekking tot de klanten bij te houden. Hier staat alle informatie die nodig is om het netwerk en de systemen van een bepaalde klant te analyseren. Wanneer een klant nieuwe infrastructuur aankoopt, verandert de documentatie. Deze documentatie wordt gebruikt voor de installatie en configuratie van nieuwe of bestaande infrastructuur. Zo weet men welke inloggegevens ingevoerd moeten worden wanneer de klant nieuwe infrastructuur aankoopt.

##### 3.1.1.2 Mondeling

Informatie kan ook mondeling geleverd worden. Om extra informatie te verkrijgen over bepaalde klanten, kan dit mondeling overgebracht worden via collega's. Ze hielpen me bij het verkrijgen van deze informatie.

#### 3.1.2 Hardware-informatie verzamelen

Naast de informatie over de klant, is het ook belangrijk informatie te hebben over de infrastructuur. Wanneer een klant nieuwe infrastructuur aankoopt en deze geconfigureerd moet worden, is het belangrijk te weten wat voor soort infrastructuur het is en wat er mee gedaan moet worden. Wanneer een klant problemen heeft met bestaande infrastructuur, is het ook belangrijk te weten welke infrastructuur de klant bezit om zo tot een oplossing te komen.

Omdat niet alle hardware op dezelfde manier geconfigureerd en geïnstalleerd wordt, kunnen collega's helpen door extra informatie te voorzien. Anderzijds is er veel informatie te vinden op het internet of in de bijgevoegde handleidingen van de infrastructuur.

### 3.1.2.1 Focuscenter

Focuscenter is een andere benaming voor de tool genaamd 'Solarwinds N-Central'. Een beschrijving van deze tool staat beschreven in het vorige hoofdstuk.

Focuscenter is de tool die Essec-BNS gebruikt om de infrastructuur van klanten te beheren. In deze tool komen alle toestellen die beheerd worden door Essec-BNS. Dit gebeurt aan de hand van een *agent* die geïnstalleerd wordt op de toestellen van klanten. Deze *agent* stuurt informatie over het toestel naar Focuscenter. Zo is het mogelijk om alle interne hardware te monitoren.

### 3.1.3 Software-informatie verzamelen

Klanten willen dat hun systemen op voorhand geïnstalleerd en geconfigureerd zijn. Ze willen dus dat bepaalde software geïnstalleerd wordt op hun toestellen. De software wordt geïnstalleerd aan de hand van de behoeften van de klant. Wanneer een klant een bepaald contract heeft, wordt een extra programma geïnstalleerd dat voor de monitoring van het toestel zorgt.

Wanneer een klant problemen heeft met bepaalde software die al op de systemen draait, is het belangrijk te weten welke software dit is. Op deze manier kan op een structurele manier gezocht worden naar de juiste oplossing voor het probleem van de klant.

Informatie over bepaalde software is terug te vinden via Sharepoint. Op Sharepoint zijn documenten beschikbaar die extra informatie bevatten over de software van klanten.

### 3.1.4 Implementatie

Met de vergaarde informatie kan de opdracht vervolgens uitgevoerd worden. Als het gaat om het installeren van infrastructuur, kan deze geïnstalleerd en geconfigureerd worden. De hardware wordt nagekeken en indien nodig geassembleerd. Wanneer de hardware klaar is voor installatie, wordt de benodigde software geïnstalleerd en geconfigureerd om deze gebruiksklaar te maken voor de klant.

Wanneer een klant problemen heeft met bestaande infrastructuur of software, wordt eerst gekeken of het probleem *remote* opgelost kan worden. Als dit geval is, probeert men het probleem op te lossen aan de hand van de vergaarde informatie. Contact met de klant is van essentieel belang, want zij willen weten wat er net gebeurd is. Wanneer men het probleem niet van op afstand kan oplossen, wordt een medewerker naar de klant gestuurd om het probleem *on-site* op te lossen.

## 3.2 Resultaat

Het doel van deze stage was om ervaring op te doen en meedraaien binnen een IT-servicebedrijf. In het begin was dit niet makkelijk omdat het nog niet duidelijk was wat ik ging doen tijdens deze stage. De opdracht was in het begin nog niet duidelijk.

Tijdens de stage was werken met hardware een belangrijk onderdeel. Deze hardware moest geconfigureerd en geïnstalleerd worden voor de klant. Dankzij deze configuratie en installatie, heeft de klant werkende infrastructuur die gemonitord wordt door Essec-BNS. Ik heb tijdens mijn stage veel laptops geïnstalleerd met Windows 10. Deze laptops worden door de klanten gebruikt en worden gemonitord door Essec-BNS. Hierdoor heeft men een overzicht van welke toestellen er in gebruik zijn bij klanten.

Naast de installatie van laptops, moesten servers nog geassembleerd worden. De onderdelen van de servers moeten nog in de server geplaatst worden. Wanneer klanten een server bestellen, is het mogelijk om extra geheugen te plaatsen of een extra voeding. Zelfs een extra processor kan



toegevoegd worden. Op deze infrastructuur kan een hypervisor geïnstalleerd worden waar de virtuele machines op draaien. Klanten moeten op deze manier niets zelf doen aan hun infrastructuur. Het resultaat hiervan is dat klanten werkende infrastructuur krijgen die meteen gebruikt kan worden zonder dat ze er zelf iets aan moeten veranderen. Deze infrastructuur zorgt ervoor dat klanten hun dagelijkse taken kunnen uitvoeren en makkelijk kunnen werken met systemen en applicaties.

Naast werkende infrastructuur is het ook belangrijk dat de toepassingen op deze infrastructuur werken naar behoren. Hiervoor is er de servicedesk die problemen oplost van klanten. Deze problemen kunnen van verschillende aard zijn. Problemen kunnen te maken hebben met hardware zoals printers of ze kunnen te maken hebben met de software die gebruikt wordt binnen het bedrijf van de klant. Wanneer een klant belt om een probleem te melden, probeert men dit zo snel mogelijk van op afstand op te lossen. Als er geen oplossing geboden kan worden van op afstand, wordt een werknemer de baan op gestuurd om dit probleem *on-site* op te lossen. Ik heb zelf bij de servicedesk gewerkt en heb hier heel veel bijgeleerd over systemen en toepassingen die klanten gebruiken. Ik heb steeds naar een goede oplossing gezocht voor de problemen die klanten hadden en heb zo veel ervaring opgedaan. Het resultaat van het werken aan de servicedesk is dat klanten kunnen bellen om vragen te stellen of met problemen en dat de werknemers aan de servicedesk dit oplossen zodat de klant zijn werk kan hervatten.

## 4 Besluit

### 4.1 Persoonlijke reflectie

Het doel van deze stage was ervaring opdoen en meedraaien binnen een IT-servicebedrijf. Dankzij deze stage ben ik in contact gekomen met verschillende technologieën en heb ik geleerd hoe ik problemen moet formuleren en hoe ik tot een adequate oplossing kan komen.

Tijdens deze stage heb ik veel nieuwe dingen geleerd in verband met hardware en software. Ik ben mee op interventie geweest bij klanten om ondersteuning te bieden bij verschillende opdrachten, ik heb meegewerkt aan de helpdesk om klanten verder te helpen met hun problemen en ik heb hardware geïnstalleerd en geconfigureerd voor klanten.

De opleiding aan de hogeschool PXL heeft me veel geleerd en heeft me geholpen tijdens deze stage. Ik heb veel kennis opgedaan de afgelopen drie jaar die ik goed heb kunnen gebruiken tijdens de opdrachten die ik heb uitgevoerd tijdens de stage. Hierdoor kon ik voor bepaalde problemen meteen een oplossing bieden. Ik heb ook ondervonden dat de opleiding niet elk onderwerp behandeld heeft. Zo heb ik gemerkt dat werken met Office 365 van groot belang is in bedrijven. Ik heb dan ook veel bijgeleerd over Office 365 toen ik aan de helpdesk zat. Ook de communicatie en omgang met klanten kan men niet leren via een opleiding. Dit heb ik tijdens de stage geleerd dankzij de helpdesk. Op de helpdesk komen dagelijks verschillende *calls* binnen met verschillende soorten problemen. Het is de bedoeling om het probleem zo goed mogelijk voor ogen te zien en een oplossing te vinden. Alleen door goed te communiceren met de klant kan een oplossing gevonden worden.

Dankzij deze stage ben ik als persoon en professional gegroeid op verschillende aspecten. Dankzij deze stage voel ik mij meer zelfzeker als het gaat over contact leggen met medewerkers en klanten. Omdat ik met nieuwe technologieën en methodieken heb gewerkt, heb ik veel bijgeleerd. Deze technologieën en methodieken hebben gelijkenissen met die dat aangeleerd worden op school, maar verschillen op enkele vlakken ook. Zo heb ik leren werken met Exchange en oudere Windows Server-versies. Ik heb ook de kans gekregen om verschillende taken uit te voeren en zo een beeld te krijgen van de werkomgeving binnen Essec-BNS.

De stage was een positieve ervaring voor mij. De verschillende opdrachten die ik heb uitgevoerd hebben me veel bijgeleerd. Deze opdrachten waren telkens verschillend waardoor ik elke dag met andere problemen in contact ben gekomen en ervaring heb kunnen opdoen. Ik ben heel tevreden met wat ik tijdens deze stage heb geleerd en ik ben de werknemers van Essec-BNS dan ook dankbaar dat ze mij deze kans hebben gegeven en mij geholpen hebben bij problemen.

## II. Onderzoekstopic

### 1 Onderzoeksvraag

Docker is de afgelopen jaren in populariteit gestegen. De Docker-containers worden steeds vaker gebruikt om te optimaliseren en te vereenvoudigen. Deze containers zijn makkelijker te onderhouden dan gehele virtuele machines. De onderzoeksvraag luidt: “Zijn er *business opportunities* door het gebruik van Docker?”. Momenteel maakt Essec-BNS nog geen gebruik van Docker-containers.

Tijdens de stage wordt onderzocht wat Docker is en hoe men dit kan implementeren. De voor- en nadelen spelen een grote rol in deze beslissing. Daarom worden deze ook opgenomen in het onderzoek samen met nog enkele beheertools en *cloudbased* opportuniteiten.

Daarna wordt een SWOT-analyse opgemaakt van Essec-BNS om zo de opportuniteiten te bekijken en een mogelijke strategie te kiezen. Vervolgens wordt de ROI-analyse opgesteld om na te gaan of de investering in Docker een positieve uitkomst heeft.

### 2 Methode van onderzoek

Om het onderzoek zo goed mogelijk te laten verlopen en de resultaten zo duidelijk mogelijk voor te stellen, bestaat het onderzoek uit drie onderdelen. Het eerste onderdeel is een literatuurstudie. Deze schets een achtergrond van de technologie. De voor- en nadelen van Docker worden beschreven alsook de verschillende beheertools en de verschillende soorten containers. Ook de integratie van storage in Docker-containers zal onderzocht worden. Verder worden de *cloudbased* opportuniteiten in kaart gebracht. Deze achtergrondinformatie is van groot belang voor de beslissing genomen kan worden in verband met het gebruik van Docker.

Daarna bestaat de mogelijkheid een ROI-analyse te maken. Deze moet uitwijzen of het een besparing is om te werken met Docker. Om tot deze conclusie te komen, voorziet Docker een speciale calculator.

Als laatste wordt een SWOT-analyse opgesteld om de opportuniteiten en bedreigingen in kaart te brengen als een overschakeling naar Docker zou gebeuren. Met deze analyse kunnen de interne en externe factoren opgesomd worden. Tenslotte kan een beslissing genomen worden op basis van deze uitkomst.

### 3 Resultaten

In dit hoofdstuk staan de bereikte resultaten beschreven. Ten eerste is er een uitgebreide literatuurstudie over Docker. Vervolgens komt de ROI-analyse aan bod. Tot slot zal de SWOT-analyse uitgelegd en toegepast worden.

## 3.1 Literatuurstudie

### 3.1.1 On-Premise versus Cloud computing

#### 3.1.1.1 On-Premise

De software wordt op de systemen van het bedrijf geïnstalleerd. Hierdoor hebben de werknemers fysieke toegang tot deze soft- en hardware. Ze hebben dan ook meteen de controle over de configuratie, het management en de beveiliging van de systemen en de data. Servers kunnen gevirtualiseerd worden en software kan uitgerold worden op deze virtuele servers, maar de fysieke hosts zijn nog steeds aanwezig in het bedrijf of in het datacenter van het bedrijf. Fysieke toegang tot de servers is nog steeds mogelijk. Dit wil dus ook zeggen dat het bedrijf zelf zorgt voor het bijkomend management zoals de stroom, koeling, netwerk,...

Deze methode wordt aangezien als de traditionele manier van werken met systemen en software. Als een bedrijf met deze methode werkt, is het meestal zo dat elke server en/of eindgebruiker een licentie moet hebben om deze systemen/software te gebruiken. [19]

Voordelen	Nadelen
Complete controle over systemen en data door fysieke toegang	Dure systemen en software om alles te laten draaien
Data kan beveiligd worden door het bedrijf zelf. Ze kunnen dan hun eigen standaarden gebruiken om de data veilig te houden.	Gelimiteerde opslagcapaciteit. Om de opslag uit te breiden moeten er dan weer kosten gedaan worden
Het bedrijf is zeker dat kritieke infrastructuur binnen het bedrijf is en dat deze veilig is.	Bijblijven met de nieuwe features en technologieën
Het bedrijf kan vaste werknemers aanduiden om de systemen te managen, een disaster recovery plan op te stellen en dit uit te voeren en om back-ups uit te voeren.	
Er is minimale vertraging in de connectie met de systemen en software omdat de data dichtbij is.	

Tabel 2 Voor- en nadelen On-Premise

#### 3.1.1.2 Cloud computing

Het aanleveren van *computing services* (servers, databases, netwerken, storage, software, etc) door gebruik te maken van het internet, wordt *cloud computing* genoemd. De bedrijven die *cloud computing* aanbieden worden ook wel *cloudproviders* genoemd. Meestal worden gebruikers van deze cloud gefactureerd aan de hand van verbruik van *resources*. [20]

Enkele voorbeelden hiervan:

- AWS
- Microsoft Azure
- Google
- IBM

Met *cloud computing* moet een gebruiker niet speciaal op een bepaalde plaats zijn om zijn/haar data te gebruiken. Zolang de gebruiker toegang heeft tot het internet, kan men de nodige data gebruiken. Dit wil zeggen dat als een bedrijf zijn systemen in de cloud heeft gezet, de werknemers van op afstand kunnen werken. Dit kan vanuit het bedrijf zelf zijn, maar ook van thuis.

Het idee van *cloud computing* is om alle data, die normaal lokaal staat op de pc of data die on-premise staat, online in de cloud te brengen zodat de lokale systemen minder belast worden of zodat er geen lokale systemen meer nodig zijn. Hierdoor worden alle bestanden, applicaties en data online beschikbaar en is het mogelijk om via een geconnecteerd toestel toegang te hebben tot data. [21]

*Cloud computing* is niet één bepaald stuk technologie. Het is een verzameling van systemen. Cloud omvat meestal de volgende drie services: IaaS, SaaS en PaaS

### 1. Infrastructure as a Service (IaaS)

De infrastructuur die een bedrijf normaal *on-premise* heeft, wordt nu gehost door de provider. De servers, storage en netwerkhardware, maar ook de virtualisatie of hypervisor worden gehost. Bij deze hosting komt vaak nog een gamma aan services. Deze services zijn *policy-driven*. Dit wil zeggen dat gebruikers een keuze hebben in welke mate men wil automatiseren en het beheren van *resources*. [22]

Enkele voorbeelden hiervan:

- Monitoring
- Security
- Load balancing
- Clustering
- Back-up
- Recovery
- Replication

### 2. Software as a Service (SaaS)

Dit is een *software distribution*-model waarbij een *cloud provider* applicaties host en deze beschikbaar stelt voor gebruikers via het internet.

Hierdoor moeten gebruikers de applicatie niet meer op hun eigen toestel of in het datacenter installeren van het bedrijf, maar kan men via het internet deze applicatie gebruiken. Hierdoor wordt de kost voor aanschaffen van hardware, onderhoud van systemen, *licensing*, installatie en support, geëlimineerd. [23]

### 3. Platform as a Service (PaaS)

De provider zorgt voor de hard- en software tools over het internet. Dit wordt vooral gebruikt voor *software development*. De provider host de hardware en de software op zijn eigen infrastructuur en zorgt er dus voor dat users geen *in-house* hardware en software moeten installeren voor het ontwikkelen en runnen van een nieuwe applicatie.

PaaS vervangt niet de hele IT infrastructuur. Men vertrouwt op PaaS voor belangrijke services zoals het hosten van een applicatie of *Java development*. De provider bouwt een betrouwbare en optimale

omgeving waarop gebruikers applicaties kunnen installeren. Hierdoor kunnen de gebruikers zich focussen op het ontwikkelen en runnen van applicaties in plaats van zich bezig te houden met de onderliggende infrastructuur. [24]

Met de opkomst van containers is er een nieuwe service die cloud-providers aanbieden, namelijk CaaS. CaaS staat voor 'Container-as-a-Service'. Providers bieden een compleet *framework* aan voor het uitrollen en beheren van containers, applicaties en clusters.

Het CaaS-model is ontworpen om ontwikkelaars en IT-departementen te helpen bij het ontwikkelen, runnen en beheren van applicaties die men in een container heeft gestoken. In dit model worden de containers en clusters aangeboden als een service. Deze service kan dan On-Premise of in de cloud uitgerold worden.

Aan de basis van het CaaS-systeem ligt het *container orchestration*-platform. Dit platform is ontworpen voor het beheer van clusters en het uitrollen van containers. Enkele populaire voorbeelden zijn: [25]

- Google Kubernetes
- Docker Swarm
- Rackspace Carina
- Apache Mesos
- Openstack nova-docker

Enkele voorbeelden van cloud-providers die het CaaS-model aanbieden:

- Google
- AWS (EC2 Container Service)
- Microsoft Azure
- Rackspace
- IBM

In deze tabel worden de voor- en nadelen van het gebruik van *cloud computing*.

Voordelen	Nadelen
Software kan op elk toestel gebruikt worden. Dit door gebruik te maken van een app of via een webbrowser	Security: gevoelige data kan zomaar gelekt worden. Deze data kan dan verkeerd gebruikt worden of verwijderd worden
Gebruikers kunnen hun data makkelijk overbrengen van het ene toestel naar het andere, op voorwaarde dat deze geconnecteerd zijn met het internet	Infrastructuur, die gehost wordt door een provider, kan blootgesteld worden aan natuurrampen, interne bugs en stroompannes
Kosten besparen: men moet niet meer alle infrastructuur zelf kopen en ergens plaatsen	Omdat veel users tegelijk de data kunnen gebruiken dankzij hetzelfde portaal, kunnen er fouten gemaakt worden en misverstanden zich voordoen. Dit heeft dan effect op het hele systeem

Groei van capaciteit is geen probleem. Als een bedrijf groeit kan men makkelijk extra infrastructuur aanvragen. Ook als men minder infrastructuur nodig heeft, kan dit makkelijk geregeld worden	
Een bedrijf betaalt enkel voor de resources die het gebruikt	

Tabel 3 Voor- en nadelen Cloud Computing

### 3.1.2 Docker

#### 3.1.2.1 Wat is Docker?

Docker is een open source platform waarmee men een applicatie in een container kan plaatsen. Een container kan makkelijk verplaatst worden van het ene platform naar het andere. De enige voorwaarde is dat elk systeem dat Docker containers wil gebruiken, gebruik maakt van de Docker software, namelijk Docker Engine.

Developers gebruiken Docker om problemen te voorkomen bij het samenwerken. Soms werkt een bepaalde applicatie wel op de ene computer, maar niet op een andere computer van een collega. Om dit probleem op te lossen, kan men gebruik maken van Docker. De containers zorgen ervoor dat een applicatie op elk platform werkt.

De mensen die het systeem moeten onderhouden, de operators, gebruiken Docker om applicaties uit te rollen en om deze te onderhouden. De verschillende applicaties kunnen dan naast elkaar in werking gesteld worden op één machine. Omdat Docker-containers zo klein zijn in grootte, verbetert dit de dichtheid per machine. Dat wil zeggen dat meer containers per machine geactiveerd kunnen worden. In tegenstelling tot virtuele machines die veel meer opslagcapaciteit nodig hebben.

Docker wordt ook gebruikt door grote bedrijven die software ontwikkelen om zo sneller nieuwe features te ontwikkelen en uit te rollen. Om dit sneller te laten verlopen, gebruikt men vaak *Continuous Integration*. Dit is een techniek waarbij ontwikkelaars hun code op een gezamenlijke *repository* uploaden en dit meerdere malen per dag. Hier wordt meestal een *pipeline* aan gekoppeld om code testen en uit te rollen. Deze *pipeline* zorgt dat dit proces sneller verloopt.

Docker verschilt van virtuele machines in de zin dat Docker werkt met Linux-containers gebaseerd op functionaliteiten binnen de Linux kernel, cgroups en namespaces. Docker voorkomt de *overhead* die ontstaat bij het gebruik van meerdere virtuele systemen met elk een eigen OS. Dit zijn de extra resources die een virtuele machine nodig heeft om te werken. Omdat Docker containers deze *overhead* voorkomt, bespaart een bedrijf niet alleen resources, maar applicaties gaan ook sneller werken omdat de noodzaak aan een hypervisor (Hyper-V, VMWare,...) wegvalt. Dit wil dus zeggen dat Docker containers op *bare-metal* gedraaid kunnen worden. Dit is een niet-gevirtualiseerde server waarop een OS geïnstalleerd is.

De technologie werkt lokaal, binnen (virtuele) datacenteromgeving en in de cloud. [26]

#### 3.1.2.2 Wat zijn containers?

Een container *image* is een klein, alleenstaand, uitvoerbaar *package* waarin software zit. Dit *package* bevat alles om de software te laten werken: code, *runtime*, *tools*, *libraries*, *settings*,... . Omdat de software in een container zit, zal deze altijd op dezelfde manier werken, ongeacht de omgeving

waarin de software geïntegreerd is. Dit kan een Linux of een Windows omgeving zijn. Deze containers isoleren software zodat deze de omgeving niet kent. Hierdoor is het mogelijk meerdere containers uit te rollen in één omgeving zonder dat deze elkaars bestaan kennen.

De containers zijn *lightweight* omdat ze op één machine werken en ze delen dan ook het OS van deze machine. De *images* worden gemaakt van *filesystem layers*. Deze *filesystem layers* zorgen ervoor dat het systeem weet hoe data opgeslagen en verkregen wordt. Zonder een *filesystem* zou informatie, opgeslagen op een *storage medium*, één grote blok data zijn. Het systeem zou dan niet weten waar een stuk data begint en eindigt.

Omdat de container zo *lightweight* is, houdt men meer opslagruimte over en de images worden sneller gedownload.

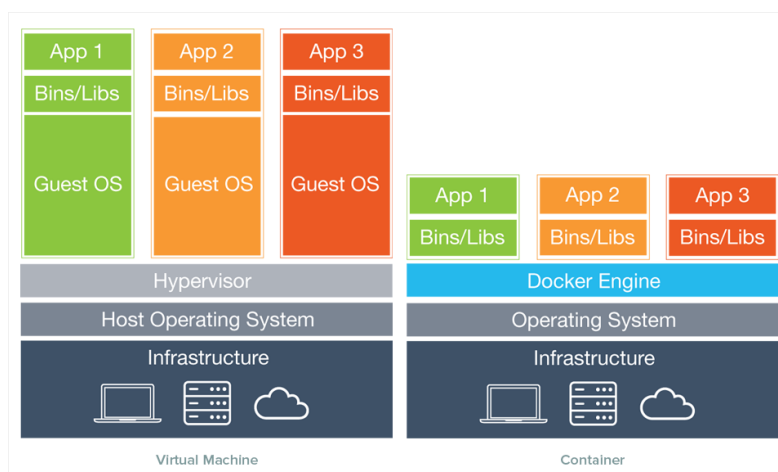
De containers runnen op de meeste Linux distributies, Microsoft Windows, in virtuele machines, op bare-metal en in de cloud. [27]

### 3.1.2.3 Container versus virtuele machine

Containers en virtuele machines lijken heel hard op elkaar in de zin dat ze ongeveer dezelfde *resources* isoleren. Ze functioneren anders; containers virtualiseren het OS, virtuele machines virtualiseren hardware. Hierdoor zijn containers efficiënter en makkelijk te verplaatsen.

Containers zijn een abstractie van de applicatie laag. Ze bundelen code en *dependencies* samen. Meerdere containers kunnen samen op één machine geactiveerd worden en delen de kernel met andere containers. Deze containers hebben hun eigen processen, geïsoleerd van de rest. Containers nemen minder opslagcapaciteit in tegenover virtuele machines omdat ze niet alle hardware virtualiseren.

Virtuele machines zijn een abstractie van de fysieke hardware. Ze veranderen één server in meerdere (virtuele) servers. De hypervisor van de machine zorgt ervoor dat meerdere virtuele machines aangemaakt kunnen worden op één fysieke machine. Elke VM heeft een kopie van het OS. Op deze VM kunnen dan applicaties geïnstalleerd worden. Hierdoor kan een VM veel plaats innemen op de *storage*. [28] [29]



Figuur 4 VM vs container



Hieronder een tabel met de voor- en nadelen het gebruik van Docker-containers.

Voordelen	Nadelen
Container < Virtuele machine	Aantal containers kan snel oplopen
Laten werken van containers verbruikt minder resources	Tijdrovend voor het beheer van vele containers
Tijd voor het uitrollen van een applicatie daalt	Minder geïsoleerd dan VM's door het delen van een OS
Kostenbesparend	Docker werkt alleen op de nieuwste versies van Windows

Tabel 4 Voor- en nadelen Docker-containers

### 3.1.3 Docker alternatieven

Docker is het meest populaire platform voor het creëren van containers. Docker heeft een standaard gemaakt voor het uitrollen van Linux-applicaties aan de hand van voorafgaande *open source*-projecten. De bedoeling was om deze containers in verschillende omgevingen te runnen. Docker heeft nog andere *open source*-projecten die de basis zijn van het Docker-platform. [30]

Omdat Docker zo populair is, willen bedrijven meteen werken met Docker. Hier moeten ze eerst goed onderzoek naar doen of Docker wel geschikt is voor hen. Alternatieven voor Docker bieden soms meer aan en daarom kan een bedrijf opteren om met deze platformen te werken in plaats van met Docker. Hieronder enkele alternatieven met wat uitleg.

#### 3.1.3.1 CoreOS rkt framework

Een eerste alternatief voor Docker is CoreOS rkt. Dit framework is ontworpen om de security en de werking van de containeromgeving te verbeteren. Docker start nieuwe containers met een *root*-proces waardoor de container gevoelig is voor malware. Hierdoor kan een hacker *root*-privileges krijgen en staat de container met andere woorden open om te hacken. In tegenstelling tot Docker start CoreOS rkt een container op zonder een *root*-proces. Hierdoor kan de container niet blootgesteld worden aan malware en is de data veilig. [31]



Figuur 5 rkt-logo

Het opzetten en uitrollen van een rkt container steunt op de CoreOS Linux distributie. Om dit allemaal in goede banen te leiden, wordt vaak gebruik gemaakt van Kubernetes. Dit is een *open source*-platform dat zorgt voor het automatisch uitrollen, schalen en in werking stellen van containers.

De CoreOS-distributie is klein en geoptimaliseerd voor het hosten van containers. Deze distributie wordt enkel gebruikt voor het maken van containers. Het is niet nodig andere features te installeren op deze distributie omdat deze zo geconcentreerd is op het maken van containers.

Om de containers van CoreOS rkt te onderhouden, te monitoren en te beveiligen heeft men een managementsysteem ontwikkeld, Fleet. Deze zorgt voor een automatische deployment en recovery van containers en zat in het CoreOS operating system of kon apart gedownload worden. Deze tool wordt niet meer verder ontwikkeld en men raadt aan om Kubernetes in de plaats te gebruiken. [32]

Hoewel CoreOS een goed alternatief is voor Docker, zijn er toch nog enkele nadelen mee gemoeid. Zo heeft Fleet een limiet op vlak van schaalbaarheid. Fleet voorziet een platform om containers uit te rollen, maar de machineclusters mogen maximum honderd nodes bevatten of duizend services draaien. Als een bedrijf toch op grotere schaal containers zou willen aanmaken, raadt CoreOS aan om Kubernetes te gebruiken. Dit is een *open source managementtool* van Google om applicaties in containers te plaatsen. [33]

CoreOS biedt zelf ook een Enterprise-level oplossing aan, namelijk Tectonic. Dit is een commercieel platform dat CoreOS en Kubernetes combineert om zo een gelijkaardige situatie als bij Google te creëren op vlak van infrastructuur. Dit platform zorgt voor betere security en schaalbaarheid die Fleet niet kan bieden aan grote bedrijven met veel containers. [34]

### 3.1.3.2 LXC/LXD

Het principe van LXC is hetzelfde als dat van Docker: het operating system virtualiseren. Hierdoor kunnen meerdere geïsoleerde Linux-systemen werken op één host. Deze containers delen dan éénzelfde Linux-kernel.



Figuur 6 LXC-logo

LXC voorziet geen virtuele machine, maar een virtuele omgeving. Deze omgeving heeft dan zijn eigen CPU, geheugen, I/O, netwerk, enzovoort. Deze virtuele omgeving wordt voorzien door cgroups en namespaces die de Linux-kernel van de host aanbiedt.

LXC-containers kunnen op twee manieren opgezet worden: met *privileged* configuratie of met *unprivileged* configuratie.

Een container zonder privileges, genaamd een *unprivileged* container, wordt aangezien als veiliger. Dit omdat deze containers een betere isolatie hebben. Deze betere isolatie komt tot stand door het mappen van de root-UID in de container naar een non-root UID op de host waar de container op draait. Het is dan ook moeilijker voor een hacker om uit de container te breken en zo de host te beschadigen. Dit wil zeggen dat wanneer een hacker toch kan uitbreken uit de container en terecht komt in de host, hij of zij geen root-rechten heeft op het host-systeem. [35]

Om LXC-containers makkelijk te beheren is een grafische interface handig. Hiervoor is dan ook een web-based tool ontwikkeld, namelijk LXC Web Panel. Met deze tool kan men via een grafische interface de lokale containers beheren. Dit is de belangrijkste feature: het beheer van lokale containers. Hierdoor krijgt men een duidelijk overzicht:



Figuur 7 LXC-webpanel

- Welke containers staan op 'Running'?
- Extra informatie over een bepaalde container
- Het netwerk van LXC-containers (dit kan men aanpassen)
- Users

### 3.1.3.3 OpenVZ/Virtuozzo

OpenVZ is een container-based virtualisatie voor Linux. Deze software creëert meerdere beveiligde en geïsoleerde containers, VPSs of VEs, op één fysieke machine. Hierdoor kan men de servers beter benutten en er zeker van zijn dat er geen conflicten ontstaan tussen applicaties.

Elke container werkt zoals een *stand-alone* server. Zo kan een container opnieuw opstarten, gebruikers, IP-adressen, geheugen, processen, bestanden, applicaties en configuratiefiles hebben.

OpenVZ is de basis van Virtuozzo. Virtuozzo biedt een virtualisatie oplossing aan en is geoptimaliseerd voor hosters en biedt hypervisors, cloud storage, support en managementtools aan. [36]



Figuur 8 Logo Virtuozzo

OpenVZ gebruikt één enkele gepatchte Linux kernel en kan dus ook alleen Linux OS gebruiken. Alle OpenVZ-containers delen deze architectuur en kernerversie. Dit kan een nadeel zijn wanneer bepaalde applicaties een andere kernerversie nodig hebben. Met OpenVZ heeft elke container ook zijn eigen filesystem.

Met OpenVZ is het mogelijk om container live te migreren. Dit betekent dat een container verplaatst kan worden van de ene fysieke server naar de andere zonder dat de container afgesloten moet worden. Dit wordt ook *checkpointing* genoemd. *Checkpointing* is het proces waarbij een container in *freeze-mode* gezet wordt, waarna zijn status opgeslagen wordt in een bestand op een bepaalde schijf. Het bestand kan overgeplaatst worden op een andere machine. Wanneer het bestand is overgeplaatst, kan de container hersteld worden (*unfrozen*). Bij *checkpointing* is de vertraging minimaal. Meestal lijkt het een vertraging door andere omstandigheden. [37]

### 3.1.3.4 Vergelijkingstabel

Deze tabel geeft een vergelijking weer tussen de verschillende container-software. [38]

	Docker	Rkt	LXC	OpenVZ/Virtuozzo
Virtualisatie technologie	OS	OS, Hypervisor	OS	OS, Hypervisor
Full-system container	Nee	Nee	Ja	Ja
Applicatie container	Ja	Ja	Nee	Nee
Container formaat	Docker container	Appc, Docker container	Linux Container	Virtuozzo container
Ondersteunde platformen	Linux, Windows, macOS, Microsoft Azure, AWS	Linux, Windows, macOS	Linux	Linux (Virtuozzo Linux, RHEL7)

Tabel 5 Vergelijking containersoftware

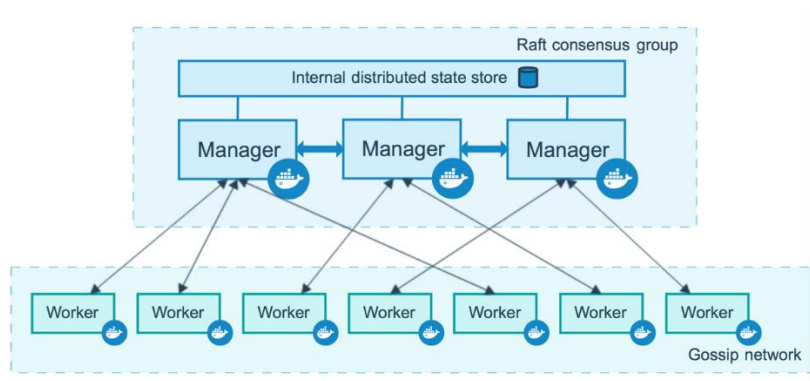
### 3.1.4 Beheertools voor containers

In de vorige hoofdstukken werd uitgelegd wat Docker is, wat containers zijn en de voor- en nadelen hiervan. Om deze containers te kunnen beheren, heeft een bedrijf nood aan tools. Deze tools dienen om het beheer van containers zo vlot mogelijk te laten verlopen.

In grote bedrijven werkt men vaak met clusters van machines. Op deze machines kan men dan gebruik maken van containers. Docker heeft een ingebouwde mode, Swarm, die voor deze clusters zorgt. Een Swarm is een cluster van machines die allemaal Docker Engine geïnstalleerd hebben. Deze bestaat uit één of meer nodes. Deze nodes kunnen virtuele servers of fysieke servers zijn. Om Swarm te gebruiken, moet elke node in de cluster in Swarm-mode staan.

In een Swarm zijn twee soorten nodes: de manager en de werker. Om een applicatie uit te rollen in een Swarm, moet een service verstuurd worden naar de managernode. De manager zendt dan taken naar de werkers. De managers zorgen ervoor dat de Swarm de gewenste toestand behoudt. Hiervoor selecteren ze een leider die alles in goede banen leidt.

De werkers krijgen taken van de manager en voeren deze dan uit. Een managernode is eigenlijk vergelijkbaar met een werkernode. Ze runnen beide dezelfde services, maar je kan een managernode zo configureren dat deze alleen maar manager-taken uitvoert. Zo wordt dit een manager-only node. Elke werker rapporteert de status van zijn toegewezen taak aan de manager zodat deze de gewenste toestand kan onderhouden en beheren. [39] [40]



Figuur 9 Swarm

Om deze clusters te beheren, is het gebruik van grafische tools aangeraden. Er bestaan ook andere tools, maar grafische tools hebben een groter gebruiksgemak. Eerst volgen er enkele voorbeelden van grafische beheertools en hun features. Daarna zal een vergelijking en een conclusie opgemaakt worden. Deze vergelijking en conclusie zullen helpen bij het kiezen van de juiste tool.

### 3.1.4.1 Portainer

Portainer is een beheertool die ervoor zorgt dat een systeemadministrator makkelijk Docker hosts, Swarms en containers kan beheren. Portainer is bedoeld om snel een container uit te rollen en deze te beheren aan de hand van een grafische interface. Hierdoor moet een systeemadministrator niet alle Docker-commando's uit het hoofd leren. [41]

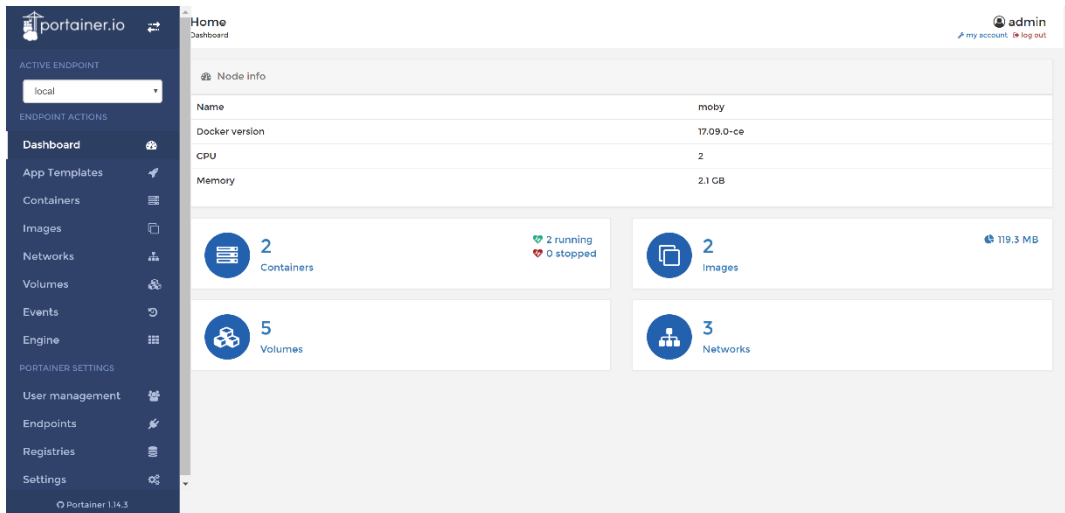


Figuur 10 Logo Portainer

Bij het inlogscherf van Portainer krijgt men een keuze: beheer een lokale Docker-host, waar de Portainer-container ook op draait of beheer een Docker-instantie die niet lokaal draait. Dit kan een host zijn die *On-Premise* werkt of een container die in de cloud staat. Als men een *remote* host wil beheren, moet men hier het IP-adres van hebben om zo met de Docker-host te connecteren.

## Dashboard

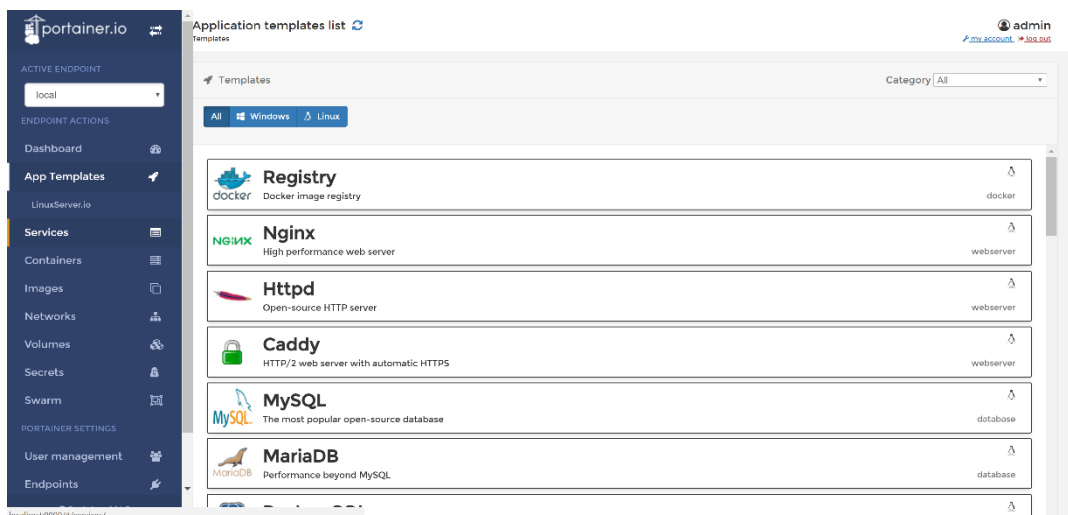
In het dashboard vinden we een overzicht van de Docker-host of Swarm-cluster. Via het dashboard is navigatie naar andere pagina's mogelijk. Het dashboard geeft een overzicht van alle benodigde elementen van de opstelling. Alle containers en images worden weergegeven in het dashboard.



Figuur 11 Portainer dashboard

## App Templates

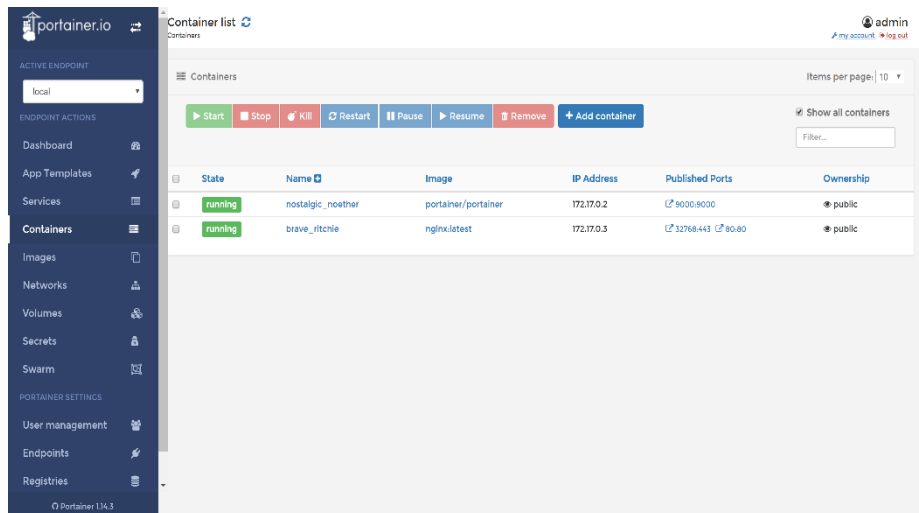
Portainer voorziet enkele *templates* om containers te downloaden en te starten. Deze methode om een container op te starten is makkelijker omdat alles grafisch wordt voorgesteld. Deze templates zijn de meest gebruikte containers. Bij de templates wordt een onderscheid gemaakt tussen Linux-containers en Windows-containers.



Figuur 12 Portainer templates

## Containers

Het tabblad "Containers" geeft de containers van de Docker Engine weer. Hier is het mogelijk enkele basisacties te ondernemen in verband met de containers, zoals het toevoegen van containers. Als de image die nodig is niet bij de App Templates staat, kan dit image gespecificeerd worden. Wanneer een bedrijf zelf een *repository* heeft op een andere locatie, bijvoorbeeld op een bepaalde *storage*, dan kan deze *repository* toegevoegd worden en kan er ook een image van deze *repository* gedownload worden.



Figuur 13 Portainer containers

Om een container nog beter te beheren, is het mogelijk om de statistieken te bekijken van de container. Hierin worden grafieken voorgesteld waarmee men de containers kan monitoren.

De mogelijkheid om rechtstreeks aanpassingen te doen in de container bestaat ook. Je kan via Portainer rechtstreeks met de CLI van de container verbinden. Hierdoor kan een systeemadministrator rechtstreeks in de container werken.

### Images

Een container wordt aangemaakt door een image te downloaden. Wanneer via de Docker CLI een nieuwe container aangemaakt wordt met het docker run-commando, wordt de image gedownload. Deze image kunnen we terugvinden met de Docker CLI via het commando *docker images*. Als Portainer gebruikt wordt om een nieuwe container aan te maken, komt de image in een lijst te staan. In deze lijst staat informatie over de images.

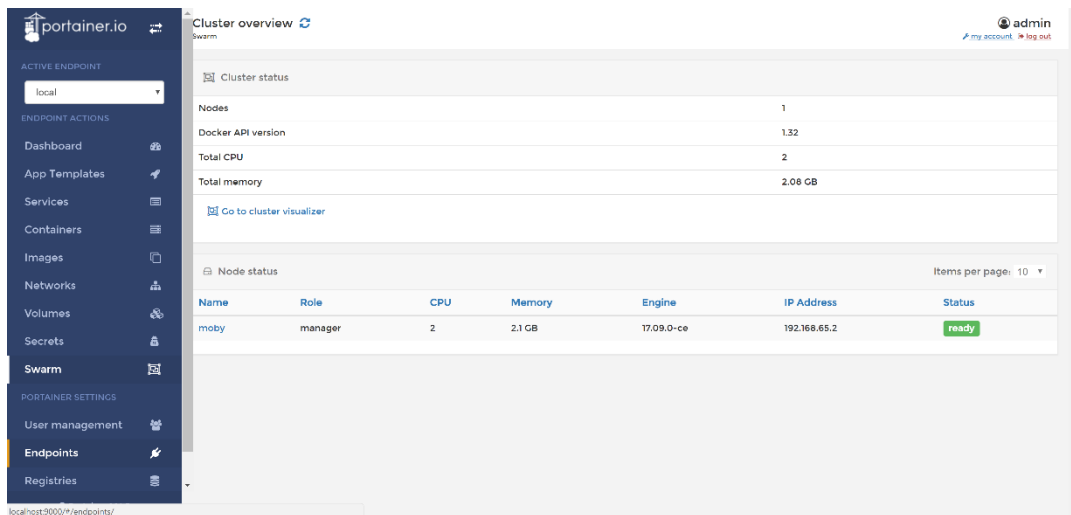
### Networks en volumes

In deze sectie kan de systeembeheerder de netwerken beheren. Wanneer bepaalde containers niet in hetzelfde netwerk horen, kan de systeembeheerder extra netwerken toevoegen en zo alle containers verbinden met de gewenste netwerken.

Bij volumes kan de systeembeheerder de opslagplaatsen beheren. Hier kan een systeembeheerder bepaalde volumes van de host *mounten* in een container. Wanneer een systeembeheerder een bepaald volume van de host *mount* in de container, wordt de data naar deze plaats gekopieerd op de host.

### Swarm

Wanneer een systeembeheerder een Docker Swarm heeft gemaakt, wil hij graag al de nodes in de Swarm kunnen beheren. Met Portainer kan dit allemaal op een gecentraliseerde plaats. Het tabblad 'Swarm' bevat informatie over de nodes in een Swarm. Hij kan dan elke node in de Swarm apart beheren.



Figuur 14 Portainer Swarm

### 3.1.4.2 Shipyard

Shipyard lijkt heel erg op Portainer. Ze hebben allebei ongeveer dezelfde functionaliteiten. In tegenstelling tot Portainer heeft Shipyard wel een eigen infrastructuur. Dit wil zeggen dat Shipyard tijdens de installatie enkele containers opstart om zo de webinterface in werking te stellen. Portainer verschilt hierin omdat deze tool maar één container opstart en meteen werkt. Met Portainer bespaart men extra resources. [42]



Figuur 15 Logo Shipyard

## Containers

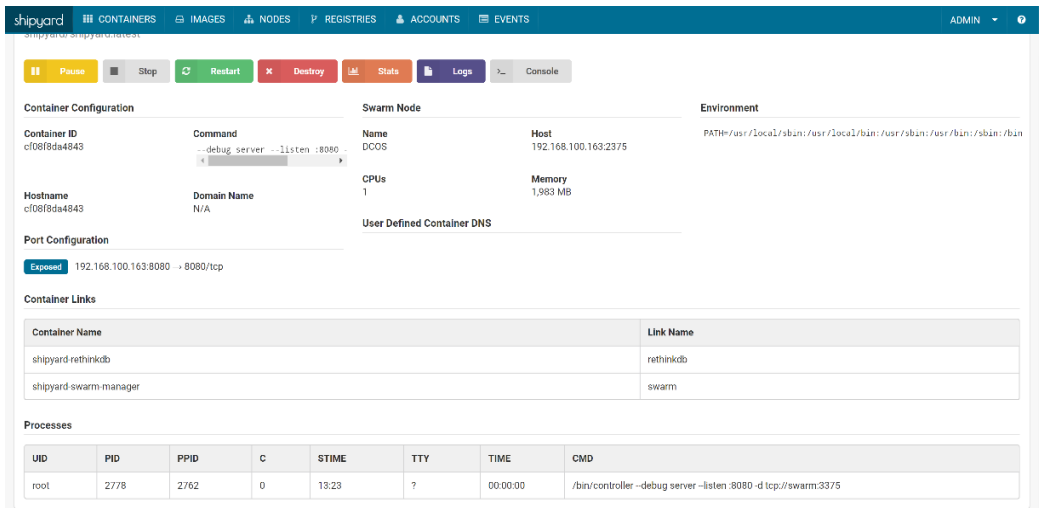
Bij het tabblad 'containers' worden alle lopende containers weergegeven. Via de gegeven interface is het mogelijk om meer gedetailleerde informatie op te vragen van bepaalde containers.

The screenshot shows the Shipyard 'CONTAINERS' tab. It features a table with columns for Id, Node, Name, Image, Status, Created, and Actions. There are buttons for 'Refresh' and 'Deploy Container' at the top left, and a search bar at the top right.

Id	Node	Name	Image	Status	Created	Actions
b34bee5753d0	DCOS	testUbuntu	ubuntu	Exited (0) 6 minutes ago	2017-10-05 11:28:35 +0200	[Q] [X]
cf08f8da4843	DCOS	shipyard-controller	shipyard/shipyard:latest	Up 5 minutes	2017-10-05 11:19:34 +0200	[Q] [X]
9e156b2e4c23	DCOS	shipyard-swarm-agent	swarm:latest	Up 5 minutes	2017-10-05 11:19:24 +0200	[Q] [X]
700be4d503d7	DCOS	shipyard-swarm-manager	swarm:latest	Up 5 minutes	2017-10-05 11:19:22 +0200	[Q] [X]
26fc156bd492	DCOS	shipyard-proxy	shipyard/docker-proxy:latest	Up 5 minutes	2017-10-05 11:19:14 +0200	[Q] [X]
4ad9f901e5d2	DCOS	shipyard-certs	alpine	Up 5 minutes	2017-10-05 11:19:05 +0200	[Q] [X]
ce7e15f8a7ec	DCOS	shipyard-discovery	microbox/etcd:latest	Up 5 minutes	2017-10-05 11:19:04 +0200	[Q] [X]
6f74888ad142	DCOS	shipyard-rethinkdb	rethinkdb	Up 5 minutes	2017-10-05 11:18:55 +0200	[Q] [X]

Figuur 16 Shipyard containers

Als er extra informatie gewenst wordt over een bepaalde container, is dit mogelijk door op het vergrootglas te klikken naast de gewenste container. Hier zijn dezelfde features beschikbaar als bij Portainer.



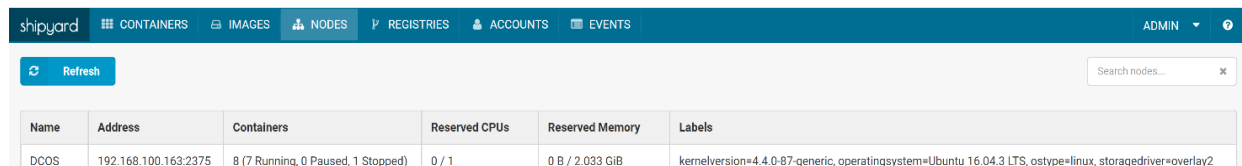
Figuur 17 Shipyard info containers

## Images

In de categorie 'Images' worden alle images weergegeven die gedownload zijn van het internet of van de lokale *repository*. Het is mogelijk om hier nieuwe images te downloaden. Hiervoor moet enkel een naam opgegeven worden.

## Nodes

Nodes zijn computers of virtuele machines binnen een cluster. Onder nodes worden de verschillende nodes van een cluster weergegeven. Een limitatie is dat alleen nodes bekeken kunnen worden die in éénzelfde cluster zitten. Dit wil zeggen dat andere clusters niet beheerd kunnen worden en dat er per cluster een nieuwe Shipyard-instantie geïnstalleerd dient te worden.



Figuur 18 Shipyard nodes

## Registries

Bedrijven kunnen er voor kiezen om zelf een *registry* bij te houden voor de images die ze gebruiken. Wanneer een bedrijf ervoor kiest om een eigen *registry* te gebruiken, moet men deze toevoegen aan deze tool. Shipyard verbindt dan met deze *registry* die op een ander locatie opgeslagen is. Zolang een connectie gemaakt kan worden naar deze *registry*, kan men images downloaden en uploaden.

## Accounts

Hier kunnen nieuwe accounts toegevoegd worden. Wanneer men een nieuwe account aanmaakt, kan de administrator kiezen welke rechten deze nieuwe gebruiker krijgt. Een gebruiker kan bijvoorbeeld enkel leesrechten hebben. De gebruiker kan dan geen nieuwe containers aanmaken en ook geen images downloaden.



### 3.1.4.3 Rancher

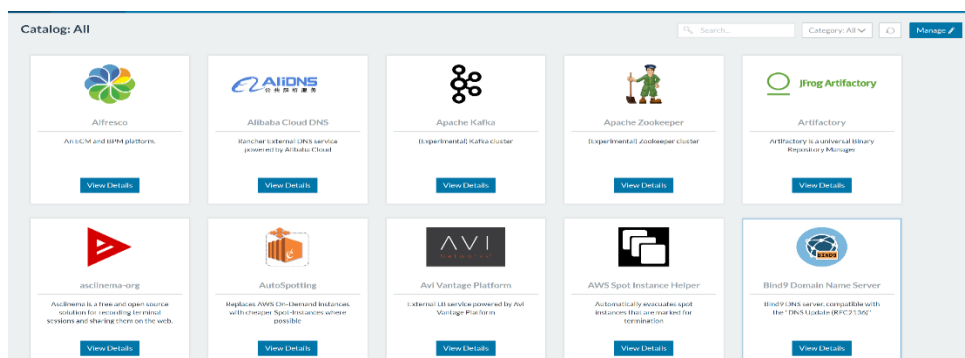
Rancher is een *open source*-project dat een heel platform voorziet voor het uitrollen van Docker-containers in productie. Rancher ondersteunt ook Mesos en Kubernetes. Dit zijn systemen om applicaties automatisch uit te rollen, te schalen en te beheren. Deze groeperen verschillende containers die samen een applicatie vormen om zo het beheer op een efficiëntere manier te doen. Kubernetes en Mesos worden niet ondersteund door de voorgaande tools.



Figuur 19 Logo Rancher

Rancher heeft nog een aantal features die de andere tools niet hebben, zoals het instellen van *high availability*. Ook een *healthcheck* zit ingebouwd in Rancher. Deze monitort hosts en containers en geeft informatie over de staat van de host. [43]

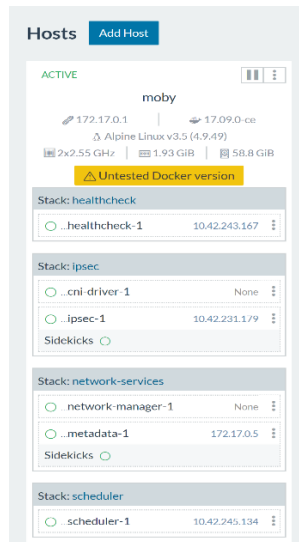
Stacks kunnen toegevoegd worden waardoor deze tool veel geavanceerder is dan de voorgaande tools. Zo is het bijvoorbeeld mogelijk om een Portainer-container te draaien bovenop de Rancher-server. Rancher voorziet een hele catalogus met verschillende tools die bovenop de Rancher-server geïnstalleerd kunnen worden.



Figuur 20 Rancher stacks

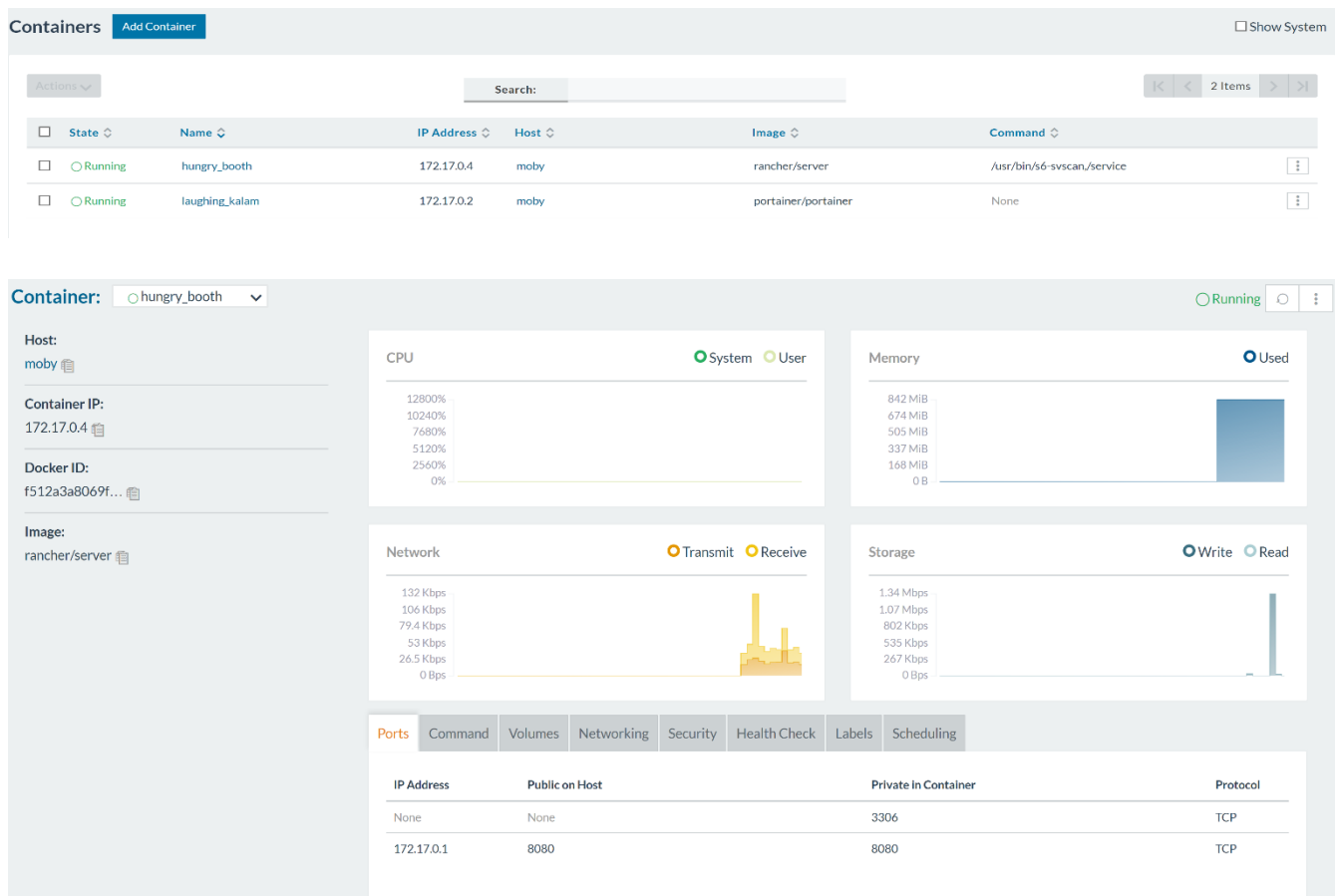
Met Rancher kunnen verschillende soorten hosts toegevoegd worden aan de server. Zo is het mogelijk om machines toe te voegen die in de cloud aanwezig zijn. Enkele voorbeelden zijn Microsoft Azure, Amazon AWS, ...

Deze hosts worden dan toegevoegd en kunnen gemonitord worden.



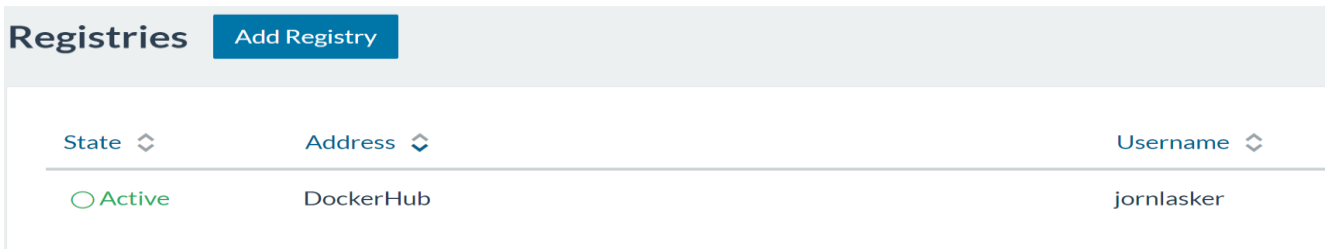
Figuur 21 Rancher hosts

Containers kunnen bekeken worden in een ander tabblad. Ook extra informatie over een bepaalde container kan worden weergegeven als dit gewenst is.



Figuur 22 Rancher containers

Om images op te slaan, is een *registry* nodig. Rancher biedt een functie aan die ervoor zorgt dat een bedrijf kan connecteren met een *registry* op de DockerHub, een Quay-*registry* of een eigen *registry*. Een voorwaarde om de eerste twee opties te gebruiken, is dat men een account heeft op deze platformen.



Figuur 23 Rancher registry

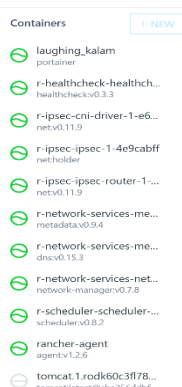
### 3.1.4.4 Kitematic

Kitematic is een *open source*-project dat het proces van het uitrollen en beheren van lokale containers op Mac en Windows makkelijker maakt. Dit omdat Kitematic werkt met een grafische interface. Deze software moet worden geïnstalleerd op de machine waar de containers op uitgerold worden. Dit wil zeggen dat men met Kitematic geen fysieke machines of virtuele machines van op afstand kan beheren en dus geen containers kan uitrollen op deze machines. [44]



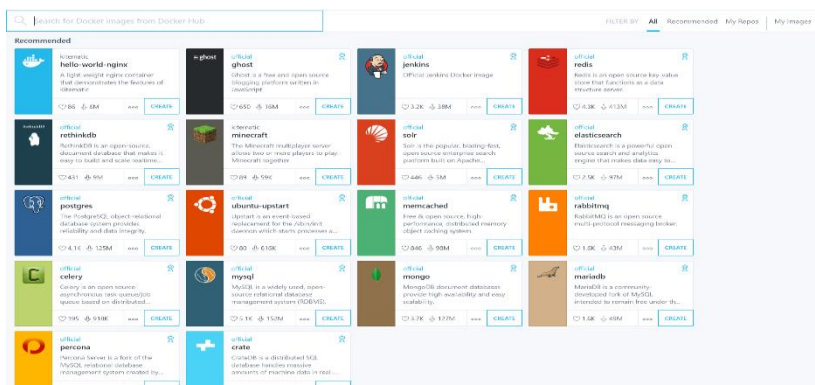
Figuur 24 Logo Kitematic

Wanneer Kitematic geïnstalleerd is op Mac of Windows, komt men meteen terecht in het centrale beheer van de containers. De software herkent meteen containers op de host en geeft een lijst weer van deze containers.



Figuur 25 Kitematic containers

Kitematic is verbonden met de Docker Hub waardoor elke container gedownload kan worden naar een toestel. Ook als een bedrijf zelf een *repository* heeft op DockerHub, met hun eigen images, kunnen deze gedownload worden.



Figuur 26 Kitematic Docker Hub

Wanneer men een container heeft gedownload en opgestart, kan men naar de logs kijken, een interactieve sessie starten met de container en veranderingen aanbrengen in de configuratie van de container.

Deze tool is handig voor ontwikkelaars om lokaal op hun machine te gebruiken. Hiermee kunnen ze goed hun applicatie testen. Omdat de tool zo klein is en weinig functionaliteiten heeft, zijn er betere opties om in de productie-omgeving te gebruiken. Deze tool kan niet gebruikt worden als men een grote cluster heeft waar veel containers op geïnstantieerd zijn. Men zou dan op elke node deze software moeten installeren. Het beheer van de cluster is niet mogelijk met deze software.

### 3.1.4.5 Docker Datacenter

Docker Datacenter is een tool voor het beheer en *deployment* van containers en services. Dit project is ontwikkeld door Docker om bedrijven te helpen met hun eigen Docker-platform. Docker beschrijft de Docker Datacenter tool als “an integrated, end-to-end platform for agile application development and management at any scale”.

Docker Datacenter zorgt voor een soort *container-as-a-service* binnen een bedrijf. IT-afdelingen binnen een bedrijf kunnen Docker-images opbouwen en opslaan. Zo kunnen ontwikkelaars applicaties ontwikkelen en testen op hun eigen machine in een container. Systeembeheerders kunnen daarna de applicaties doorsturen naar de gewenste servers om ze in productie uit te rollen. Docker Datacenter bevat twee componenten, namelijk Docker Universal Control Plane en Docker Trusted Registry. [45]



Figuur 27 Logo Docker Datacenter

### Docker UCP

Docker UCP is een beheertool voor clusters, ontwikkeld door Docker en is geïntegreerd in de Docker Enterprise Edition. Deze tool kan geïnstalleerd worden on-premise of in de cloud. Met deze tool kunnen Docker Swarms en applicaties beheerd worden via één centrale interface.

Met Docker kunnen duizenden fysieke machines of virtuele machines samen gebundeld worden tot een cluster, een Swarm genaamd. In een Swarm bestaat de mogelijkheid een applicatie te schalen. Dit wil zeggen dat men de applicatie gaat uitrollen in meerdere containers. Dankzij UCP wordt het schalen, uitrollen en beheren van containers eenvoudiger en containers en clusters kunnen gemonitord worden binnen dezelfde grafische interface.

Het beheer van *computing resources* kan gedaan worden in Docker UCP. Hiermee kunnen netwerken, volumes en nodes beheerd en gemonitord worden. Ook applicaties kunnen veelzijdig beheerd en gemonitord worden.

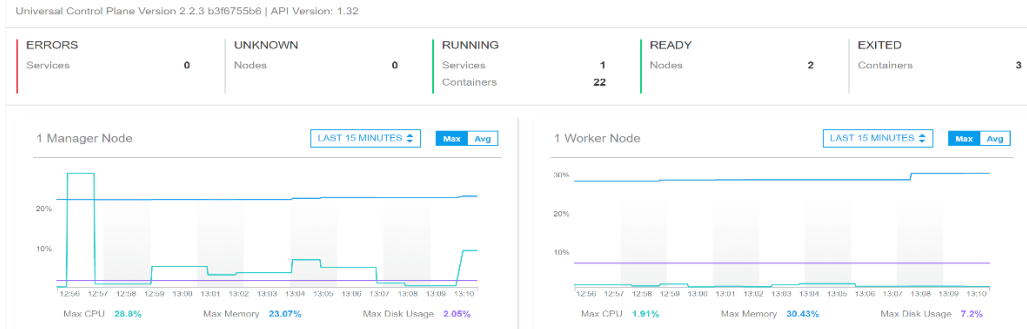
Docker UCP heeft een ingebouwd authenticatiemechanisme, maar geeft ook de mogelijkheid om met LDAP en AD in te loggen. Samen met deze inlogmethoden biedt het ook RBAC aan. Met RBAC kan de administrator beslissen wie de containers en applicaties kan beheren en veranderingen kan aanbrengen.

UCP integreert Docker Trusted Registry (DTR) waardoor Docker-images opgeslagen kunnen worden achter de firewall van het bedrijf. De images zijn opgeslagen op een veilige plaats en kunnen enkel aangepast worden door bevoegde personen.[46]

Hieronder enkele features van Docker UCP:

## Dashboard

In het dashboard wordt een algemeen overzicht gegeven van een cluster. Het aantal containers op de verschillende hosts wordt weergegeven, grafieken in verband met CPU kunnen geraadpleegd worden en informatie over RAM en storage is ook beschikbaar.



Figuur 28 DDC dashboard

## Service

Onder services is het mogelijk om nieuwe services te lanceren of lopende services te bekijken. Wanneer een service wordt aangemaakt, is het mogelijk deze meteen te configureren via de interface. Zo kan men bijvoorbeeld een webserver opzetten die meteen werkt.

The interface shows the configuration for an nginx service. The SERVICE DETAILS section includes Name (nginx) and TASK TEMPLATE Image (nginx:latest). The ENDPOINT SPEC section includes Mode (VIP), PORTS (Internal Port 8080), Protocol (top, http), Publish Mode (Ingress, Host), and Public Port (8080). Below the configuration is a table of services:

Status	Name	Image	Mode	Last Updated	Last Error
● 1/1	nginx	nginx:latest@sha256:004ac1...	Replicated	a minute ago	No errors

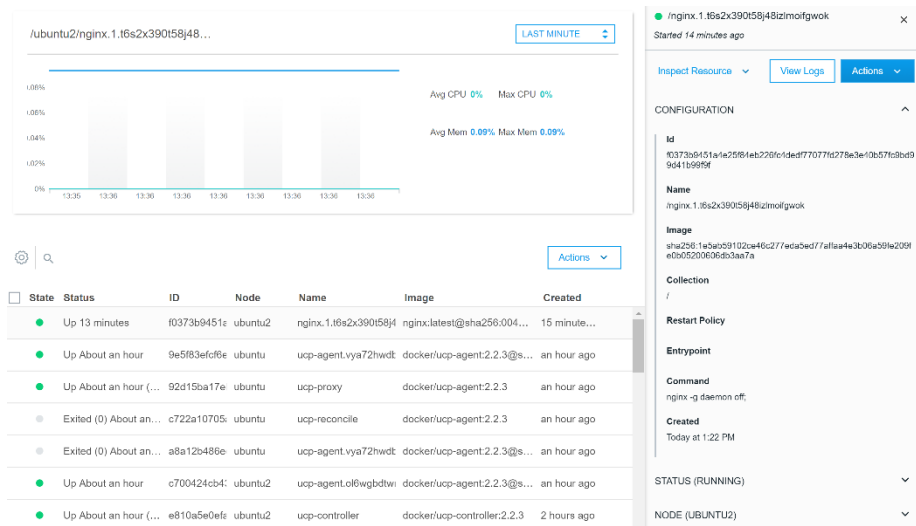
Figuur 29 DDC services

## Containers

Onder het tabblad 'Containers' zijn de containers te zien die gestart zijn. Voor meer gedetailleerde informatie over een bepaalde container, kan men op deze container klikken en wordt de extra informatie getoond.

Wanneer aanpassingen gedaan moeten worden in de container, is er de mogelijkheid de CLI van de container te gebruiken.

Ook kunnen nog andere acties ondernomen worden, zoals containers verwijderen, de logs bekijken,...



Figuur 30 DDC containers

## Images

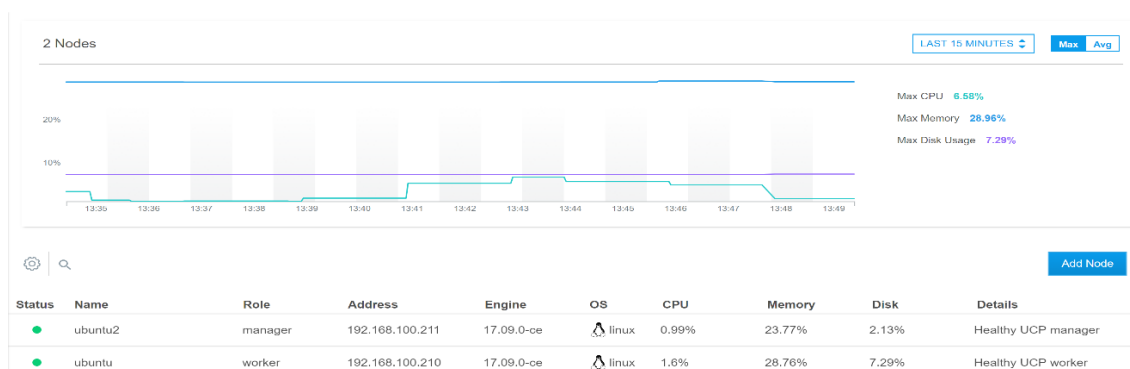
Zoals de titel aangeeft, kunnen de images bekeken worden die al zijn gedownload. Via dit tabblad is het ook mogelijk om nieuwe images te downloaden. Deze images worden gedownload van Docker Hub of van een DTR die aangemaakt kan worden. Om toegang te hebben tot deze *repositories* is authenticatie vereist.

Wanneer er een bestand beschikbaar is waarmee een image opgebouwd kan worden, is het mogelijk dit bestand in te laden en een image te creëren. Dit bestand moet in de vorm van een .tar zijn.

## Nodes

Een cluster bestaat uit verschillende nodes. Deze nodes zijn zichtbaar in het tabblad 'Nodes'. Hier wordt onder andere informatie gegeven over de status van de node, het gebruik van CPU, of het een manager of een worker is, ... .

Ook nieuwe nodes kunnen toegevoegd worden. Er moet gekozen worden of deze node een manager of een worker wordt en welk besturingssysteem deze node heeft. Daarna wordt een script gegenereerd dat uitgevoerd moet worden op de node die toegevoegd moet worden aan de cluster.



Node Type\*

windows  linux

Node Role\* ?

Manager  Worker

USE A CUSTOM LISTEN ADDRESS ?

USE A CUSTOM ADVERTISE ADDRESS ?

Nodes are added to the cluster by using Docker's 'swarm join' command. Use the following command on the engine you want to connect to the cluster.

```
docker swarm join --token SWMTKN-1-4c0q7tyks5uznzkqbnfaod6iw9m0jl1irvbsrrtz1o1vjhnhkhku-1aj2er13mwip40au8n1vxkpih 192.168.100.211:2377
```

Figuur 31 DDC nodes

## Netwerken en Volumes

Onder 'Networks' worden alle netwerken weergegeven van de nodes. Hier bestaat de mogelijkheid om nieuwe netwerken toe te voegen en oude netwerken te verwijderen. Bij 'Volumes' worden alle volumes weergegeven die gekoppeld zijn aan containers. Extra informatie over deze volumes kan verkregen worden door erop te klikken en de mogelijkheid is er om nieuwe volumes toe te voegen.

## Docker Trusted Registry

DTR is een opslagplaats voor images die meegeleverd wordt bij Docker Datacenter. Deze opslagplaats kan achter een firewall geplaatst worden waardoor images veilig zijn en beheerd kunnen worden.

Net zoals Docker UCP kan DTR *on-premise* of in de cloud geïnstalleerd worden. Omdat automatisering zo belangrijk wordt, kan DTR gebruikt worden in een CI-pipeline om applicaties continu te bouwen, te versturen en in gebruik te nemen.

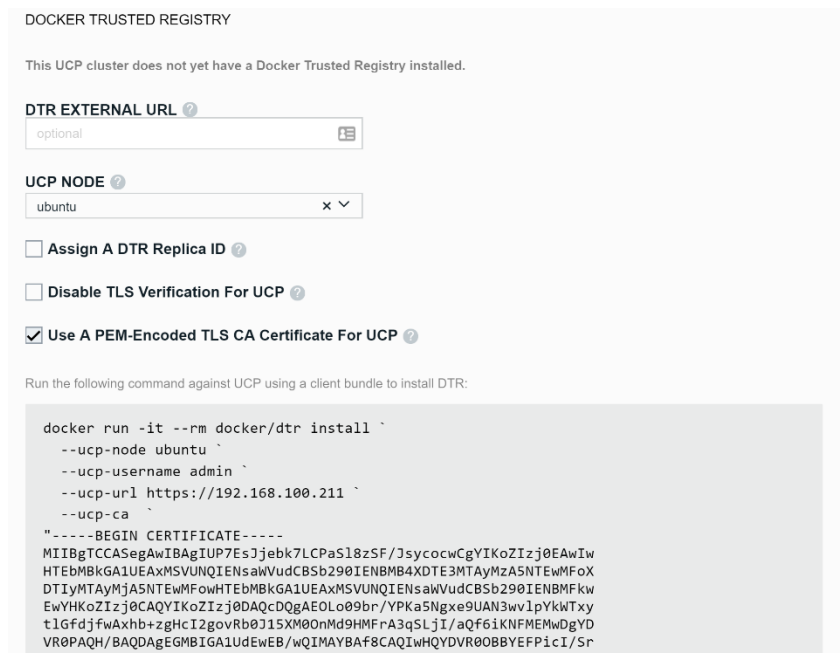
Voor het gebruik van DTR heeft Docker een grafische interface ontwikkeld waar geautoriseerde gebruikers images kunnen beheren. Wanneer een gebruiker toegang heeft tot deze images, kan hij/zij informatie verkrijgen over hoe de image is opgebouwd, hoe de Dockerfile is opgebouwd. Met security scanning heeft de gebruiker de mogelijkheid om een lijst te verkrijgen van de geïnstalleerde software binnen het image.

DTR heeft een ingebouwde *securityscanner* waarmee achterhaald kan worden welke versie van software gebruikt is in een bepaald image. De scanner scant elke laag van de image en geeft de resultaten laag per laag terug. Hierdoor creëert de scanner een volledig plaatje van wat er uitgerold wordt in productie. Een extra feature van deze scanner is dat deze de bekomen informatie vergelijkt met een database waar kwetsbaarheden in staan. Mogelijke kwetsbaarheden voor de applicatie kunnen dan opgelost worden.

Samen met DTR wordt Notary geïnstalleerd. Hiermee kunnen images gesigneerd en geverifieerd worden aan de hand van bepaalde sleutels.

DTR kan geïnstalleerd worden via de interface van UCP. In de interface moet gekozen worden op welke node men de DTR wil installeren. Enkele andere instellingen worden ook weergegeven waar uit gekozen kan worden. Nadat alle benodigde informatie is ingevuld, wordt een script gegenereerd

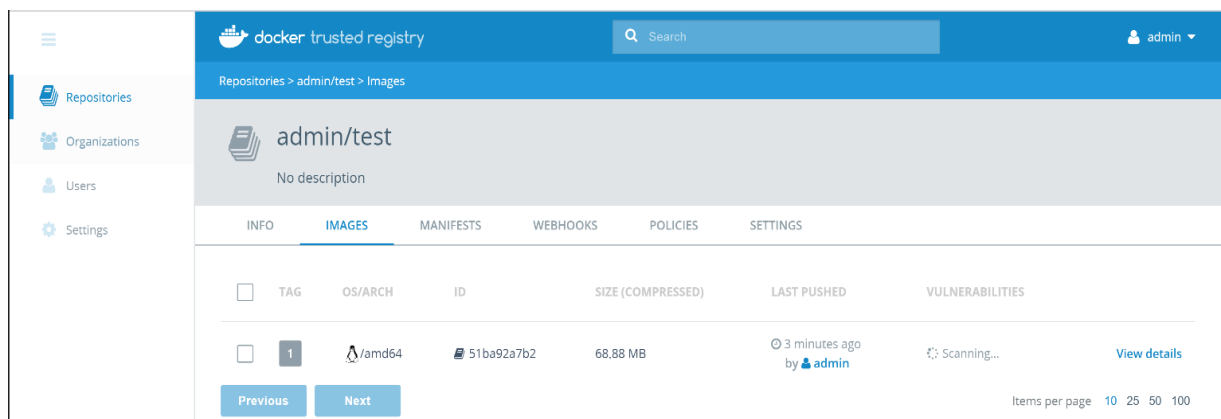
dat uitgevoerd moet worden op de node waar de opslagplaats voor images, DTR, geïnstalleerd moet worden. Dit script zorgt ervoor dat er een nieuwe stack wordt toegevoegd aan Docker Datacenter.



Figuur 32 DDC DTR

Wanneer DTR geïnstalleerd is op de gewenste node, kan de webinterface bereikt worden via het IP-adres of de URL die eraan toegewezen is. Op deze grafische interface kan men dan inloggen met de UCP-credentials. Via deze interface is het mogelijk om nieuwe repositories aan te maken en kan men images van deze repositories downloaden.

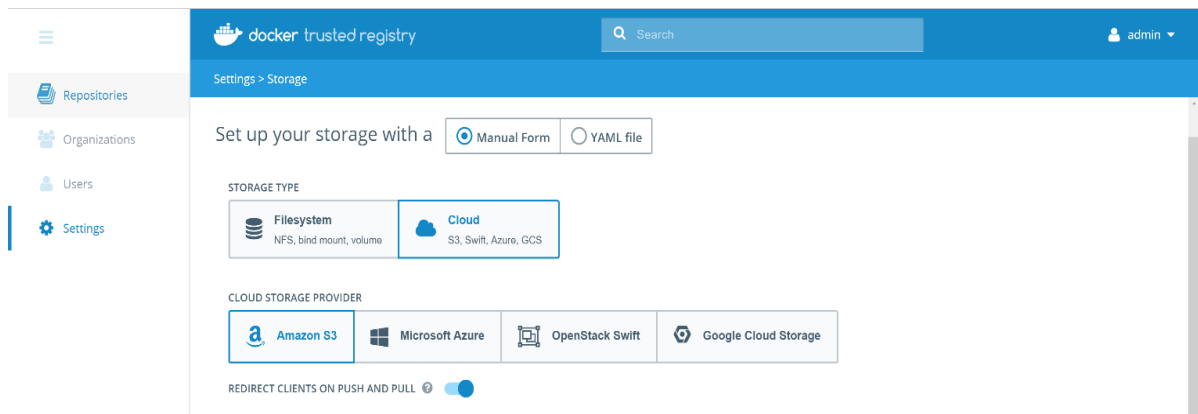
Voordat men een image upload naar de repository, kan men deze een signatuur meegeven zodat dit een trusted image is.



Figuur 33 DTR

Voor het opslaan van images bestaan er verschillende mogelijkheden. Er is de keuze tussen lokaal images opslaan of opslaan in de cloud. Bij deze tweede optie heeft men de keuze uit verschillende cloud-providers, zoals Amazon S3, Microsoft Azure, OpenStack Swift en Google Cloud Storage. Om van deze opties gebruik te maken, is een account vereist.





Figuur 34 DTR settings

### 3.1.4.6 Vergelijking beheertools

In volgende tabel wordt een vergelijking gemaakt tussen de verschillende beheertools. Hierbij komen technische onderdelen aan bod, maar ook eigen bevindingen.

	Portainer	Shipyard	Rancher	Kitematic	Docker Datacenter
Makkelijke installatie?	✓	✓	✓	✓	✓
Container console	✓	✓	✓	✓	✓
Private repository integratie	✓	✓	✓	✓	✓
Beheer van netwerken (bestaande beheren, nieuwe aanmaken)	✓	✗	✓	✗	✓
Beheer van volumes	✓	✗	✓	✗	✓
Beheer van verschillende Docker hosts	✓	✗	✓	✗	✓
Authenticatie met Active Directory (ingesteld via de GUI)	✓	✗	✓	✗	✓
Beheren van andere clustermethoden (bijvoorbeeld Mesos, Kubernetes)	✗	✗	✓	✗	✗
High Availability setup	✗	✗	✓	✗	✓

Tabel 6 Vergelijking van beheertools

### 3.1.4.7 Conclusie beheertools

Zoals de vergelijking aangeeft, zijn Rancher en Docker Datacenter de meest complete tools. Beide tools implementeren een groot aanbod aan functies. Het verschil tussen beiden is dat met Rancher ook andere clustermethoden beheerd kunnen worden terwijl dit met Docker Datacenter niet gaat. Docker Datacenter maakt gebruik van de eigen clustermethode, namelijk Swarm.

Voor een bedrijf dat alleen met Docker wil werken en dus Swarm als clustermethode, is Docker Datacenter de beste oplossing omdat deze tool echt gericht is op Docker-instanties. Wanneer men ervoor kiest om een andere clustermethode te gebruiken, kan men gebruik maken van Rancher. Bij Rancher heeft men nog andere opties om clusters te beheren.

Een alternatief voor Docker Datacenter is Portainer. Deze tool was de eerste die ik had geïnstalleerd en deze was ook makkelijk om mee te werken. Het was meteen duidelijk waar alles geconfigureerd kon worden en de documentatie was ook in orde.

Omdat Essec-BNS nog geen Docker gebruikt, zou ik aanraden om Docker Datacenter te gebruiken dat bij de Docker Enterprise Edition komt. Deze tool is ideaal voor een bedrijf dat pas begint met Docker. Als men later wat meer ervaring heeft met Docker is het nog altijd mogelijk om over te stappen naar een andere tool zoals Rancher om een andere clustermethode te gebruiken.

### 3.1.5 Verschil tussen Linux en Windows containers

Windows-containers en Linux-containers lijken heel erg op elkaar. Wanneer men gebruik maakt van Docker op een Windows-toestel, kunnen de containers op dezelfde manier beheerd worden als op een Linux toestel. Dit wil zeggen dat men gebruik kan maken van Docker Compose, Docker Swarm, Docker Machine,... . Windows-containers kunnen ook gebruikt worden zonder Docker. Deze containers kunnen enkel op Windows-machines aangemaakt en gestart worden.

Hoewel deze containers heel erg op elkaar lijken, zijn er toch enkele verschillen in implementatie tussen beide. [47]

	Linux containers	Windows containers
Systeem processen	Syscalls	DLL
OS-architectuur	/	Compute service abstraction level
Union file system	Wel UFS	Geen UFS

Tabel 7 Verschil Linux- en Windows-containers

Wanneer we kijken naar het verschil tussen Docker-containers op Windows en Linux, zien we enkele andere verschillen. [48]

	Docker-containers op Linux	Docker-containers op Windows
Installatie	Docker Engine  Management tools	MobyLinuxVM  Subnet om met localhost te communiceren  NAT om containers onderling te laten communiceren
OS	Docker kan op de meeste versies van Linux geïnstalleerd worden	Windows Server 2016  Windows 10 Pro

Tabel 8 Verschil Docker-containers op Linux en Windows

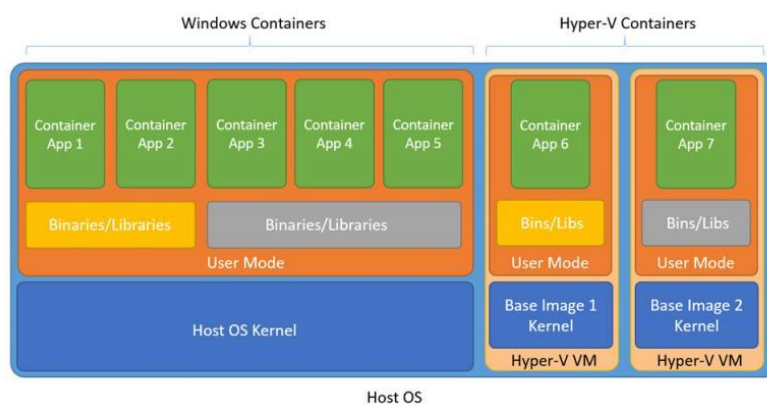
Binnen de familie van de Windows-containers is er nog een onderscheid tussen twee bepaalde vormen van containers. Er is de Windows Server Container en de Hyper-V Container. Ze verschillen op vlak van isolatie en performantie. [49]

### Windows container

Een Windows Server Container deelt de kernel met de host van de container en met alle andere containers van de host. Omdat de containers een kernel delen, hebben alle containers dezelfde versie van de kernel nodig om te werken en dezelfde configuratie van de kernel. Deze containers bieden niet de benodigde beveiliging.

### Hyper-V container

Deze containers werken binnen een geoptimaliseerde virtuele machine. Hierdoor wordt de kernel van de host niet gedeeld met andere containers. Deze containers zorgen voor dezelfde beveiliging als een normale virtuele machine. Omdat deze containers geen kernel delen, kunnen ze elk een andere versie van de kernel draaien en een andere configuratie aannemen.



Figuur 35 Windows Container vs Hyper-V container

### 3.1.6 Integratie van shared storage (Nimble)

Containers worden vaak gebruikt om applicaties te testen omdat ze zo makkelijk opnieuw op te bouwen zijn. Wanneer een ontwikkelaar een nieuwe *feature* of een update wil doorvoeren in de applicatie, kan hij de container stoppen, de updates uitvoeren en de container opnieuw starten. Dat is geen probleem wanneer men met containers werkt die geen data moeten opslaan, maar wat gebeurt er als een applicatie wel data moet opslaan en deze data niet verloren mag gaan wanneer de container gestopt en verwijderd wordt?



Figuur 36 Logo  
Nimble

Docker heeft hiervoor een oplossing: werken met volumes. Deze volumes kunnen data opslaan die men nog steeds kan gebruiken wanneer een container gestopt en verwijderd is. Deze volumes kunnen folders van de host zijn die *gemount* worden in de container. Een bepaalde map in de container verwijst dan naar een map op de host. In deze map kan data opgeslagen en beschikbaar gesteld worden op de host.

Essec-BNS werkt met NimbleStorage. Docker en Nimble zijn partners en Docker heeft een *plugin* ontworpen voor het gebruik van NimbleStorage in een Docker-container. Als een bedrijf data wil behouden die gegenereerd wordt binnen de container, kunnen ze gebruik maken van de HPE Nimble Storage Docker Volume plugin. Met deze plugin is het mogelijk om volumes te beheren via Docker API of CLI. [50]

Hieronder staan enkele features van deze plugin:

- Creëer Docker-volumes (achterliggend zijn dit Nimble-volumes) om data op te slaan op een efficiënte manier
- Kloon Docker-volumes zodat er meerdere kopieën zijn voor de ontwikkelaars en testers
- Importeer bestaande Nimble-volumes van bestaande omgevingen in een Docker-omgeving
- Importeer klonen van Nimble-snapshots in Docker
- Maak Docker-volumes gebaseerd op Nimble beschikbaar voor elke node die containers draait
- Nimble-volumes verwijderen uit Docker
- Filesystem-standaarden instellen voor Docker-volumes die gebaseerd zijn op Nimble

### 3.1.7 Containers in de cloud

Cloud is tegenwoordig een *hot topic* in de IT-wereld. Bedrijven verhuizen hun infrastructuur naar de cloud om verschillende redenen zoals *security, high availability, redundancy, management,...* . Al deze onderdelen worden overgenomen door de cloudprovider waardoor een bedrijf minder werkkraft nodig heeft om al deze infrastructuur te beheren. Hoe minder werkkraft men nodig heeft om de infrastructuur te beheren, hoe meer men hierop kan besparen.

Samen met cloud zijn containers ook een *hot topic*. Er zijn verschillende cloudproviders die services in verband met containers aanbieden.

#### 3.1.7.1 Amazon Web Services (AWS)

AWS biedt een service aan voor het beheren van containers, namelijk Amazon EC2 Container Service. Dit is een snelle, schaalbare container management service die het makkelijk maakt om Docker-containers te runnen, stoppen en beheren die draaien op een cluster van Amazon instanties. ECS geeft een bedrijf de mogelijkheid om applicaties uit te rollen

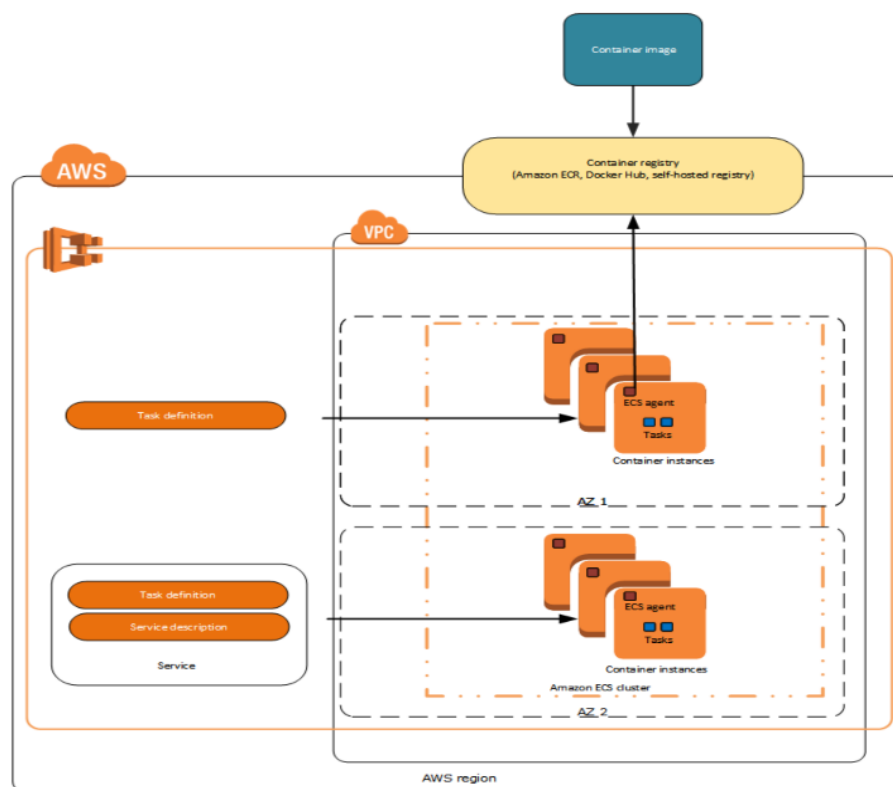


Figuur 37 Logo AWS

in containers met *API-calls*. Ook is het mogelijk om de status van de cluster te bekijken en andere features van Amazon EC2 te gebruiken.

Amazon ECS kan gebruikt worden om te plannen waar containers geplaatst moeten worden in een cluster en dit gebaseerd op de resources die men nodig heeft, de *policies* omtrent isolatie en de vereisten in verband met beschikbaarheid. Met ECS is het dus niet nodig zelf een *cluster-managementtool* en/of een *configuration-managementtool* te voorzien.

Amazon ECS is een service die het makkelijker maakt om applicaties in containers te draaien op een manier dat ze altijd beschikbaar zijn. Deze containers zijn dan verdeeld over meerdere *availability zones* in een bepaalde regio. Een ECS-cluster kan gemaakt worden in een bestaande of een nieuwe *virtual private cloud* (VPC). Nadat een cluster is opgezet, is het mogelijk om taken en services te definiëren welke containers op welke node van de cluster moeten draaien. De images voor deze containers worden gedownload van *registries*. Deze *registries* kunnen zich binnen of buiten de AWS infrastructuur bevinden. [51]



Figuur 38 AWS EC2 architectuur

Hieronder enkele features van Amazon EC2 Container Service: [52]

### Compatibiliteit met Docker

ECS ondersteunt Docker waardoor het mogelijk is om Docker-containers te draaien en te beheren in de cluster. Elke instantie van de cluster bevat een service, namelijk Docker-daemon. Met deze service is het mogelijk om applicaties in Docker-containers te stoppen en ze uit te rollen op de instanties van de cluster.

## Clusters beheren

Wanneer het aankomt op het managen van een cluster kan het soms ingewikkeld zijn om zelf de beheerinfrastructuur te installeren, beheren en schalen naar gewenste grootte. ECS neemt al deze moeilijkheden weg en zorgt voor een simpel beheer van de cluster. Het enige wat een gebruiker moet doen is het starten van de nodes en de containers uitrollen die men wil. ECS zorgt daarna voor het management van de cluster.

## Taken definiëren

Met de *Task Defenition* van ECS kunnen containers makkelijk aangemaakt worden. Dit gebeurt in een JSON-file waarin men het aantal containers kan meegeven, welke *Docker-repository* gebruikt moet worden om het image af te halen, wat de *requirements* zijn in verband met geheugen en CPU, welke *shared* volumes gebruikt moeten worden en hoe de containers gelinkt zijn aan elkaar.

## Controle met behulp van APIs

ECS voorziet een aantal simpele APIs waarmee de service uitgebreid kan worden. Hiermee kunnen clusters aangemaakt en verwijderd worden, taken kunnen geregistreerd worden, Docker-containers kunnen gestart en gestopt worden en informatie over de cluster kan weergegeven worden.

## Plannen

Om te voldoen aan de behoefte aan resources en beschikbaarheid van de cluster, heeft ECS enkele *schedulers* geïmplementeerd. Hiermee is het mogelijk om bepaalde taken en *batch-jobs* te plannen om later uit te voeren.

## Repositories

ECS kan gebruikt worden met elke *repository* of een toegankelijke *Docker-registry* of met de DockerHub. Deze *repository* moet enkel meegegeven worden in de *Task Definition* en de juiste image zal worden gedownload.

## Beveiliging

Voor elke taak kan een bepaalde *IAM-role* toegekend worden. Dit is een beveiligingssysteem dat op basis van identiteit van de gebruiker toegang verleend.

### 3.1.7.2 Google Cloud

Google heeft een product genaamd Google Container Engine. Dit is een beheerde omgeving voor het uitrollen van applicaties die in containers gestopt zijn.



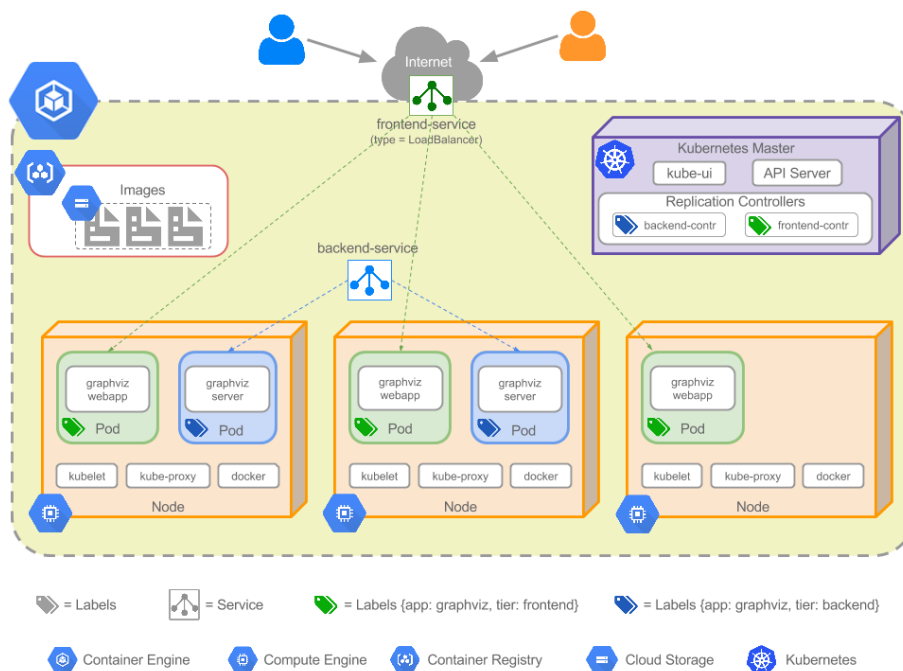
Figuur 39 Logo Google Cloud

Container Engine zorgt ervoor dat applicaties en services snel ontwikkeld, uitgerold, geüpdatet en beheerd kunnen worden. Container Engine kan ook gebruikt worden wanneer een applicatie data moet behouden. Hierdoor is het mogelijk om *persistent storage* aan een container te koppelen. Ook databanken kunnen in aparte containers binnen een cluster aangemaakt worden. Wanneer een bedrijf containers wil uitrollen, moet ze enkel de *compute*-componenten, geheugen en de storage kiezen die de applicatie nodig heeft en Container Engine zorgt voor deze resources en beheert de onderliggende *resources* automatisch.

Voor het monitoren van een cluster en de *resources* wordt gebruik gemaakt van Google *Site Reliability Engineers* (SREs). *Liveness checks* detecteren en vervangen applicaties die niet meer werken. Deze checks maken gebruik van een bepaalde replicatietechniek en automatische reparaties van containers. Hierdoor worden services altijd beschikbaar en ondervindt een gebruiker van de applicatie geen hinder bij het gebruik van de applicatie.

In tijden wanneer een service veel aanvragen krijgt, zorgt Container Engine voor *autoscaling*. Hierdoor blijft de service beschikbaar wanneer de aanvragen beginnen te pieken. Als de piek afgelopen is, is het mogelijk om terug te schalen naar minder containers om zo geld te besparen op *resources*.

Google Container Engine maakt gebruik van Kubernetes voor het beheren van clusters. Deze tool werd ontwikkeld door Google en wordt nu beheerd door een *community*. Met Kubernetes kan men zelf bepaalde dingen bepalen, namelijk het beheren van de monitoring, de logging en *continuous integration*. Wanneer een bedrijf beslist om niet meer te werken met Google Container Engine kunnen ze de *workloads* die draaien bij Google exporteren en ergens anders plaatsen waar Kubernetes ondersteund is. [53]



Figuur 40 GKE architectuur

### 3.1.7.3 Microsoft Azure

De containerservice van Microsoft Azure heet Azure Container Service (AKS). Deze service maakt het makkelijk om clusters aan te maken, te configureren en te beheren. Hierdoor kan een gebruiker zijn ervaring gebruiken om container-based applicaties uit te rollen op Microsoft Azure. Door AKS te gebruiken, worden de *enterprise-features* van Azure beschikbaar gesteld en *portability* van applicaties is verzekerd dankzij Kubernetes en Docker.



Figuur 41 Logo Microsoft Azure

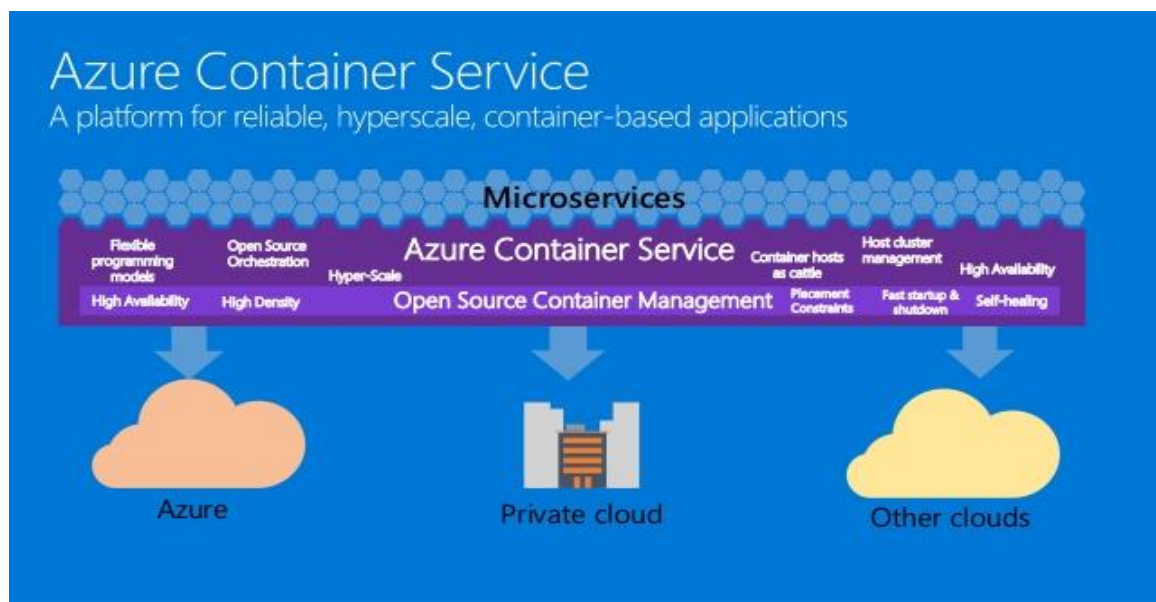
Met Kubernetes moet een gebruiker zich geen zorgen meer maken over planning en het uitrollen van applicaties. Een cluster kan opgezet worden in enkele klikken. Monitoring en cluster management

kan allemaal beheerd worden via Kubernetes, die ook zorgt voor automatische upgrades en beschikt over een ingebouwde console.

In Azure is het mogelijk om gebruikte clusters te verplaatsen naar een andere omgeving waar ook Kubernetes gebruikt wordt als *managementtool*. Het gebruik van een eigen *registry* is mogelijk, maar Azure biedt ook een *registry* aan, namelijk Azure Container Registry.

Wanneer een applicatie veel aanvragen krijgt van gebruikers, moet de applicatie schaalbaar zijn. Dankzij AKS kan men snel en efficiënt schalen om zo maximaal gebruik te maken van *resources* zonder dat de applicatie offline gaat. Om dit te kunnen uitvoeren, isoleert men de applicatie van *failures* in de infrastructuur. De onderliggende infrastructuur wordt dan transparant geschaald waardoor het de stijgende aanvragen kan beantwoorden.

Azure voorziet Kubernetes als cluster-managementtool in AKS. Ze bieden ook de mogelijkheid om een andere *orchestrator* te gebruiken, zoals Docker of DC/OS. Eigen managementtools en monitoring tools kunnen dus ook toegevoegd worden. [54]



Figuur 42 Azure architectuur

### 3.1.7.4 Prijsvergelijking cloudproviders

De onderstaande tabel geeft een prijsvergelijking weer tussen de verschillende cloudproviders. Deze vergelijking is tot stand gekomen door de verschillende calculators van de verschillende cloudproviders te gebruiken. Bij elke provider is ongeveer dezelfde configuratie toegepast.

	AWS	Google	Azure
Instanties	T2.xlarge	n1-standard-4	D3 v2
Geheugen	16 GB	15 GB	14 GB
CPU	4	4	4



Nodes	10	10	10
Uptime	100%	100%	100%
Prijzen	\$0,188/uur * 730 * 10 = \$1376,20 →€1181,15/maand	(\$0,161 * 730 * 10) + \$109,50 - \$106,29 = \$1178,51 →€1011,87/maand	€0,197 * 730 * 10 = €1438,10/maand

Tabel 9 Prijsvergelijking cloudproviders

### 3.1.8 Containers migreren naar de cloud

Containers kunnen aangemaakt worden *on-premise* om dan gebruikt te worden voor bepaalde bedrijfsprocessen. Maar wat als het bedrijf kiest om deze containers toch te verplaatsen naar de cloud? Containers die geen data moeten opslaan, kunnen makkelijk naar de cloud verplaatst worden. Het is mogelijk om de image van deze container een nieuwe tag te geven, deze te pushen naar een bepaalde *repository* en daarna opnieuw te downloaden op de virtuele machine in de cloud. De container kan dan opnieuw opgestart worden met de nieuwe image.

Wanneer een container wel data moet bijhouden, is het een goede oplossing om deze data apart op te slaan, dus niet in de container zelf. Dit omdat de data die in de container geschreven wordt niet blijft bestaan als de container afgesloten wordt. De data gaat dan verloren.

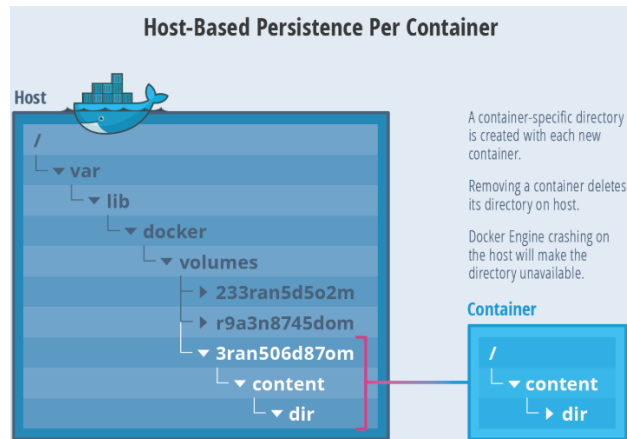
#### Manieren om data te behouden

Om deze data te kunnen behouden, is het mogelijk om gebruik te maken van *Host-Based Persistence*. Hierbij speelt de host waarop de container staat een grote rol. De host bevat bepaalde schijven of mappen waarin data kan worden opgeslagen. Deze schijven of mappen worden *gemount* in de container. De container kan dan gegevens schrijven naar deze map. De data is dan ook meteen zichtbaar op de host op de schijf of in de map waar de data naar geschreven is. De data blijft dus beschikbaar wanneer de container wordt afgesloten. Containers kunnen volumes delen met elkaar waardoor data beschikbaar wordt op meerdere containers.

Er zijn 3 manieren om gebruik te maken van host-based persistence, namelijk: *implicit per-container storage*, *explicit shared storage* en *shared multi-host storage*. Deze lijken heel hard op elkaar, maar verschillen toch op bepaalde vlakken. [55]

#### 1. Implicit per-container storage

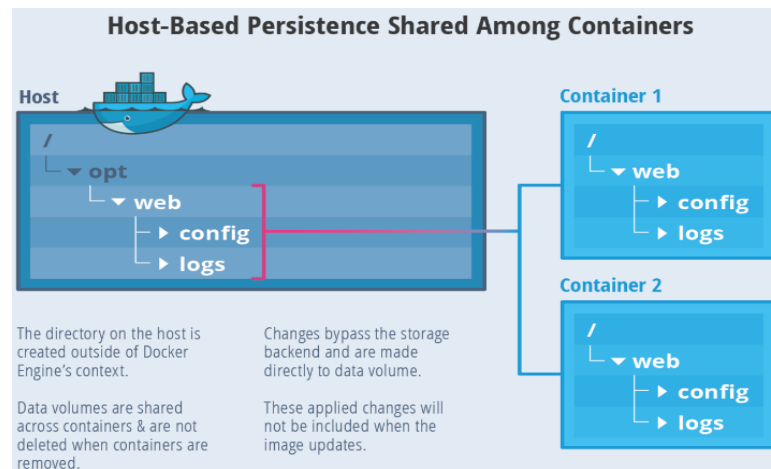
Deze manier maakt een *storage-sandbox* voor een bepaalde container. Er wordt een map aangemaakt op de host onder */var/lib/docker/volumes* wanneer de container wordt aangemaakt. Als de container verwijderd wordt, zal de map ook verwijderd worden door de Docker Engine. De data in deze *sandbox* is niet beschikbaar voor andere containers, enkel voor degene die het heeft aangevraagd.



Figuur 43 Host-based storage Docker

## 2. Explicit shared storage (data volumes)

Via deze methode is het mogelijk om data te delen tussen verschillende containers op dezelfde host. Een bepaalde locatie op de host wordt beschikbaar gesteld voor de containers. Deze locatie wordt *gemount* in de containers. Dit is nuttig wanneer meerdere containers *read-write* toegang nodig hebben op dezelfde map. Containers die bijvoorbeeld een Apache webserver draaien, kunnen logs in dezelfde directory opslaan waardoor het makkelijker wordt deze logs te beheren.



Figuur 44 Host-based shared storage Docker

Omdat de map buiten de context van de Docker Engine wordt aangemaakt, blijft deze data bestaan wanneer de containers gestopt worden en zelfs wanneer de Docker Engine gestopt wordt. Omdat deze map geen deel uitmaakt van de Docker Engine storage backend, blijft de data bestaan.

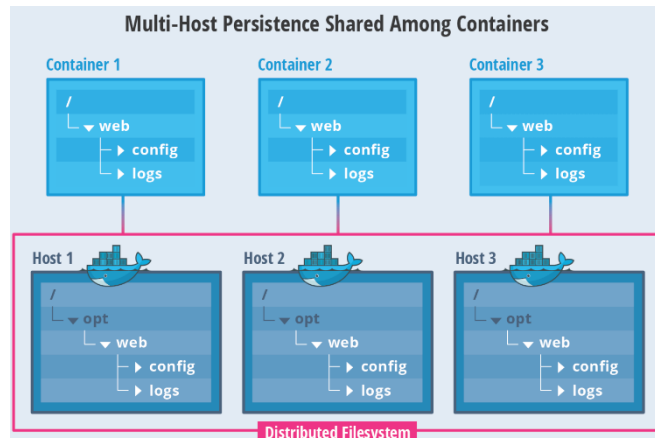
Deze techniek is de meest populaire techniek om data op te slaan en wordt daarom gerefereerd naar datavolumes in Docker.

## 3. Shared multi-host storage

Bovenstaande technieken bieden gevarieerde voordelen aan, maar toch hebben ze één groot nadeel: ze maken de containers *non-portable*. Wanneer de container van host verandert,

zal de data blijven staan op de host waar de container de data gegenereerd heeft. Hierdoor ontstaat een bepaalde band tussen de host en de container.

Bedrijven rollen applicaties in containers vaak uit in clusters waar meerdere hosts in het spel zijn. Deze situatie vraagt om storage dat beschikbaar is voor alle hosts in de cluster en die daarna ter beschikking gesteld wordt voor containers door gebruik te maken van een bepaalde *namespace*.



Figuur 45 multi-host shared storage Docker

Door gebruik te maken van een consistente benaming en een gehele *namespace* hebben alle containers toegang tot de gedeelde storage, ongeacht op welke host van de cluster de container draait.

### Migreren naar de cloud

Containers kunnen best gebruikt worden zonder dat ze data opslaan binnen de container zelf. Hiervoor bestaan de bovenstaande oplossingen. Om containers naar de cloud te migreren, is het mogelijk om de aangepaste images naar een eigen *repository* of *registry* te pushen met een bepaalde tag. Men kan dan een nieuwe virtuele machine opstarten in de cloud en hier de gewenste image downloaden. De container kan dan gestart worden en zal meteen ook werken.

Wat de datavolumes betreft, kunnen deze in een ZIP-file gestoken worden en dan gekopieerd worden naar de nieuwe virtuele machine in de cloud.

Een echte live migratie met volumes is er nog niet, maar Jelastic biedt wel een manier aan om draaiende containers te migreren van cloudprovider naar cloudprovider. Jelastic maakt gebruik van CRIU en P.Haul. Dit is een mechanisme gebaseerd op Python dat draait bovenop CRIU. Hiermee kunnen omgevingen in *freeze*-mode gezet worden en de data waarmee de container op dat moment werkt, wordt op een schijf bijgehouden. Hierdoor kan een applicatie verplaatst worden. Op de nieuwe locatie kan de omgeving terug actief gemaakt worden in de staat waarin hij gefreezeed werd. Alle inkomende aanvragen in deze periode worden in een wachtrij geplaatst waardoor gebruikers maar een kleine interruptie ondervinden in hun connectie. [56]

### 3.1.9 Back-up, Disaster Recovery en High Availability

Data is een belangrijk element in een bedrijf. Daarom is het belangrijk een goede back-up-methode te kiezen en een *disaster recovery*-plan op te stellen. Om ervoor te zorgen dat een applicatie of systeem niet of toch zo weinig mogelijk offline gaat, is het van essentieel belang om de infrastructuur in *high availability* te configureren zodat applicaties blijven werken.

#### Back-up

Het is mogelijk om een back-up te maken van Docker-containers in hun huidige staat. Dit kan op een manuele manier of via een tool genaamd Morpheus. Deze tool kan nog veel meer dan alleen back-ups maken van containers, maar enkel de back-up-feature is hier van belang.

Om een back-up te maken op een manuele manier kan men gebruik maken van ingebouwde Docker-commando's. De gegenereerde data van de container kan men best opslaan buiten de container op een aparte plaats. Het is mogelijk om de back-up-methode te gebruiken die men voordien gebruikte. Hieronder staan de stappen die uitgevoerd moeten worden om een back-up van een container te maken en deze te verplaatsen naar een andere host:

1. Docker commit -p [CONTAINER-ID] [NAME] → hiermee wordt een container gepauzeerd en de nieuwe image wordt lokaal opgeslagen.
2. Docker login → inloggen op de Docker Hub met eigen credentials.
3. Docker tag [CONTAINER-ID] [NAME[:TAG]] → het nieuwe image moet een tag krijgen voor het naar de Docker Hub gepushed kan worden.
4. Docker push [NAME[:TAG]] → push het nieuwe image naar de repository.

Op deze manier moet men elke image één voor één uploaden naar de Docker Hub. Hierdoor heeft men een back-up in de DockerHub en kunnen deze images altijd gedownload worden op nieuwe hosts waar de Docker Engine op draait.

Een back-up van images van containers die aan staan, kunnen met een zelf geschreven script opgeslagen worden op de lokale host. De code van dit script staat hieronder.

```
for i in `docker ps | awk '{ print $1|" $2 }' | grep -v CONTAINER `;
do
    docker commit `echo $i | cut -d '|' -f1` `echo $i | cut -d '|' -f2`-`date +"%m%d%y_%H%M%S"`;
done
```

Dit script maakt eerst een lijst van alle lopende containers en print dan de eerste en tweede kolom af. Daarna neemt hij de kolom met de *header* "CONTAINER". Dit gebeurt in een *loop* zodat daarna voor elke lopende container een nieuw image gemaakt kan worden met een eigen naam, samengevoegd met de huidige datum en tijd als tag.

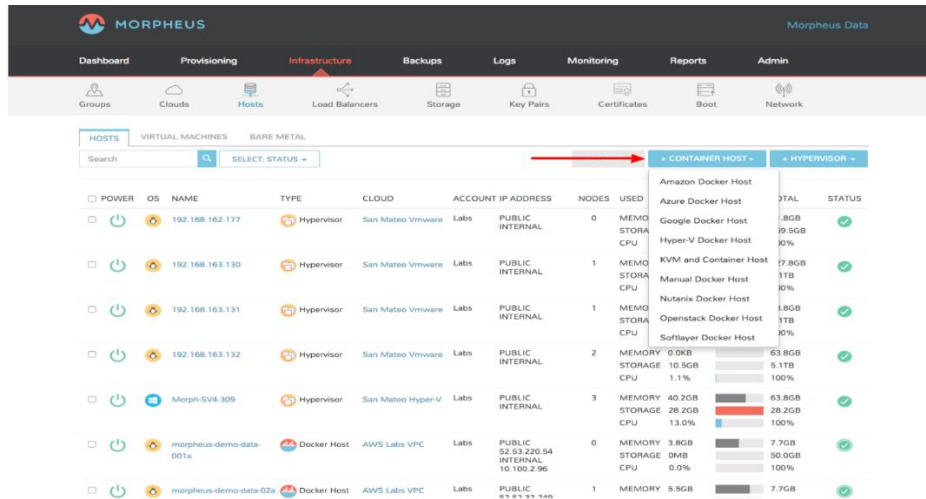
Een .tar-bestand kan ook aangemaakt worden om zo de image naar een andere host te verplaatsen. Dit gebeurt aan de hand van het 'save'-commando. (docker save -o [opslagplaats tar-file] [container-naam]).

Nadat de image verplaatst is naar de nieuwe host, kan deze ingeladen worden met het 'load'-commando.

Wanneer er liever met een grafische tool een back-up gemaakt wordt, is het mogelijk om een tool te gebruiken, namelijk Morpheus. Morpheus is sneller en efficiënter in het maken van een back-up. Deze tool maakt een back-up van de container en van de datavolumes. Hierdoor moet een systeemadministrator zich geen zorgen maken over de back-up van de images en de data. [57]

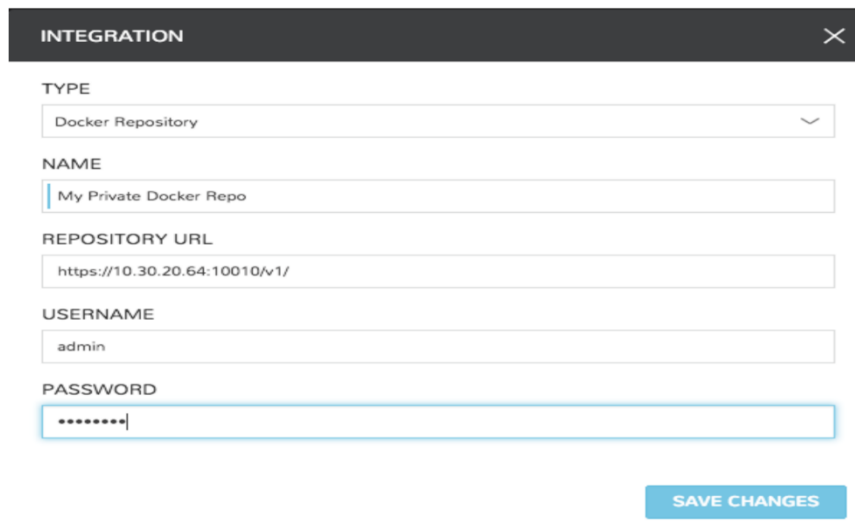
Om een back-up te maken met Morpheus moet men enkele stappen doorlopen. Deze stappen worden allemaal grafisch voorgesteld in de Morpheus-tool. [58]

1. Een Docker-host toevoegen aan een cloudprovider of een *on-premise* server. Hier is de keuze mogelijk tussen bekende cloudproviders of een eigen datacenter. De gegevens van de cloudprovider en de nieuwe host moeten ingevuld worden voordat de Docker-host aangemaakt kan worden.



Figuur 46 Morpheus dashboard

2. Toevoegen van Docker Registry aan public of private clouds. De *repository* voor het opslaan van back-ups moet hier geconfigureerd worden. Dit kan de DockerHub zijn, maar ook een *privé-repository* van het bedrijf zelf.



Figuur 47 Morpheus Registry

3. Het beheren van back-ups. Back-ups kunnen aangemaakt, bekeken en verwijderd worden. Om het allemaal te automatiseren, is er de keuze om de back-ups te plannen op bepaalde momenten.

## Disaster Recovery

Een disaster recovery plan beschrijft wat men moet doen bij een onvoorziën incident dat de bedrijfsprocessen verstoort. Dit plan moet ervoor zorgen dat de kans en de impact van zo een incident beperkt blijft. Het is een soort draaiboek dat stap voor stap beschrijft wie wat moet doen in geval van een incident. Hierdoor ontstaat de garantie dat iedereen weet wat hij/zij moet doen. Dit plan wordt opgesteld met de bedoeling het incident zo snel mogelijk op te lossen en zo snel mogelijk terug operationeel te zijn. [59]

Voor bedrijven die met Docker UCP werken, heeft Docker een disaster recovery plan opgesteld om de UCP-cluster te herstellen. Wanneer de helft of meer manager-nodes van een UCP-cluster down zijn en niet hersteld kunnen worden, verliest het systeem quorum en kan het alleen hersteld worden via het plan dat Docker-UCP heeft opgesteld. Met dit plan is het niet gegarandeerd dat alle configuraties en lopende services hersteld worden. Een bedrijf moet zichzelf beschermen tegen deze gebeurtenis door de cluster in *high availability* op te stellen en regelmatig back-ups te nemen.

Volgens het plan is het mogelijk om een nieuwe cluster aan te maken op één van de overgebleven manager-nodes. Hiermee wordt een nieuwe single-manager cluster opgebouwd voor het herstellen van de staat van de cluster. Daarna moet een back-up genomen worden van één van de overgebleven managers. Na de back-up moet UCP gedeïnstalleerd worden op alle nodes van de cluster die nog werken. Nu is het mogelijk om een *restore* te doen op een manager node die men heeft kunnen redden. Als er nog nodes zijn die in de oude cluster zitten, moeten deze manueel verwijderd worden uit de oude cluster en toegevoegd worden in de nieuwe cluster. [60]

Voor de hardware in een datacenter, heeft een bedrijf een eigen disaster recovery plan. Dit kan gebruikt worden om zo de fysieke servers en virtuele machines te herstellen.

## High Availability

High availability zorgt ervoor dat een applicatie zo weinig mogelijk downtime ondergaat. Vaak worden clusters van machines gemaakt om zo een groep machines te hebben die redundant zijn. Als een bepaalde machine dan niet meer functioneert, kan een andere machine zijn taak overnemen waardoor de *downtime* van een applicatie zo laag mogelijk blijft. Het proces waarbij een machine de taak op zich neemt van een niet-functionele machine, noemt men failover.

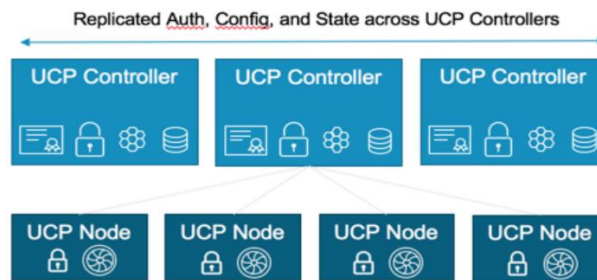
Om een cluster van machines te maken, kan men gebruik maken van software. In het geval van Docker-containers is het mogelijk om via Docker Datacenter en UCP een cluster van nodes te maken waardoor *high availability* gegarandeerd wordt. Docker Datacenter zorgt voor de HA van de nodes terwijl *container-rescheduling* zorgt voor de HA van containers. Deze *rescheduling* wordt als parameter meegegeven bij het aanmaken van een nieuwe container.

## HA op de infrastructuur

Docker Datacenter is een combinatie van UCP en DTR. HA voor de infrastructuur begint bij het instellen van UCP. Hierbij is het mogelijk om meerdere UCP controllers te configureren die allemaal op een andere host werken. Aan de hand van een *key value store* worden de controllers geüpdatet met de laatste informatie over de cluster. Elke controller is een identieke kopie van de originele controller. Dit zorgt ervoor dat wanneer een controller niet meer werkt, een andere zijn werk kan overnemen met identiek dezelfde configuratie.

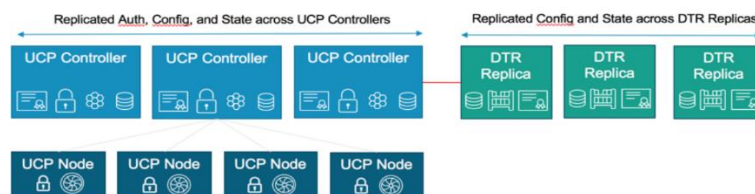
Een cluster met N controllers kan  $(N-1)/2$  niet-werkende nodes tolereren. Dit wil dus zeggen dat een cluster met vijf nodes twee niet-werkende nodes tolereert. Wanneer een cluster minder nodes bevat dan het minimum, breekt de cluster af. Hoe meer controllers toegevoegd worden, hoe groter de

foutmarge mag zijn. Dit brengt wel met zich mee dat meer hosts nodig zijn en dat de snelheid van de cluster verlaagd wordt. [61]



Figuur 48 High Availability UCP

Samen met UCP is het gebruik van DTR aangeraden. DTR maakt gebruik van replica's om HA te garanderen. Deze gebruikt dezelfde formule als UCP voor tolerantie van fouten.



Figuur 49 High Availability DTR

Via UCP is het mogelijk om containers zo in te stellen dat ze opstarten op een andere node wanneer één van de nodes van de cluster niet meer werkt. Wanneer een container aangemaakt wordt binnen de cluster zijn er verschillende instellingen die hiervoor kunnen zorgen. Dit voorbeeld werkte via UCP, maar via andere tools zoals bijvoorbeeld Rancher is het ook mogelijk om HA-clusters op te stellen.

### 3.1.10 SWOT

In dit onderdeel wordt beschreven wat een SWOT-analyse is en waarvoor deze gebruikt kan worden. Daarna wordt deze analyse toegepast op het stagebedrijf.

#### 3.1.10.1 Wat is een SWOT-analyse?

Een SWOT-analyse is een bedrijfskundig model waarmee men langs systematische weg inzicht verschaft in de positieve en negatieve uitdagingen die er voor een organisatie zullen ontstaan. Tijdens deze analyse onderzoekt men de belangrijkste bestaansmiddelen en toekomstige ontwikkelingen.

Een bestaansmiddel is bijvoorbeeld de relatie met de klant, ervaring van medewerkers, investeringscapaciteit, ... . Het kan dus menselijk, fysiek, financieel of technologisch van aard zijn. Een ontwikkeling is bijvoorbeeld marktgroei, concurrentiesituatie, ... . het heeft zijn oorsprong buiten de organisatie.

De bestaansmiddelen worden tijdens deze analyse verdeeld in sterkten en zwakten. De ontwikkelingen worden verdeeld in kansen en bedreigingen. De essentie van deze analyse is dat men sterkten en zwakten systematisch confronteert met kansen en bedreigingen en hieruit strategische conclusies trekt. [62]

De afkorting SWOT is overgenomen uit het Engels en staat voor:

- Strengths
- Weaknesses
- Opportunities
- Threats

### 3.1.10.2 Wat is het nut?

Door het maken van een SWOT-analyse krijgt een organisatie een beeld van de situatie waarin de organisatie nu verkeert of waar ze in terecht zal komen. Het is een krachtig middel om nieuwe markten te ontdekken of de organisatie te verbeteren. Omdat de belangrijkste uitdagingen zichtbaar zijn, kan een organisatie haar strategische koers bepalen en indien nodig aanpassen. [62]

### 3.1.10.3 SWOT-analyse Essec-BNS

In de volgende tabel staat de SWOT-analyse voor Essec-BNS. Deze heb ik opgesteld aan de hand van eigen ervaringen binnen het bedrijf.

Sterktes		Zwaktes	
<ul style="list-style-type: none"><li>- Veel kennis en ervaring</li><li>- Breed assortiment</li><li>- Bekendheid in de regio</li></ul>		<ul style="list-style-type: none"><li>- Weinig actief op sociale media</li><li>- Minder kennis van Linux t.o.v. Windows</li></ul>	
Kansen		Bedreigingen	
<ul style="list-style-type: none"><li>- Snel groeiende technologische ontwikkelingen (processen sneller uitvoeren)</li><li>- Meer afzet</li></ul>		<ul style="list-style-type: none"><li>- Mogelijke toetreding van nieuwe concurrenten</li><li>- Concurrentie heeft meer kennis van Docker</li><li>- Hoge macht van afnemers</li></ul>	

Tabel 10 SWOT-analyse Essec-BNS

### 3.1.11 ROI

In dit onderdeel van het eindwerk wordt de ROI berekend door gebruik te maken van Docker. Om deze ROI te berekenen, wordt gebruik gemaakt van een speciale calculator, aangeboden door Docker.

#### 3.1.11.1 Wat is ROI?

ROI staat voor 'Return on investment' en geeft het rendement op een bepaalde investering weer. Wanneer een bepaalde investering een verlies oplevert, zal de ROI negatief zijn. Wanneer de investering winst oplevert, zal de ROI positief zijn. De ROI wordt berekend door de nettowinst te delen door de boekwaarde van de totale activa. De ROI van een deelproject kan berekend worden door de specifiek opbrengst voor dat project te delen door de specifieke investering.



Om de ROI te berekenen, is het mogelijk een formule te gebruiken. Deze formule ziet er als volgt uit:

$$\text{ROI} = (\text{OPBRENGSTEN DOOR INVESTERING} - \text{KOSTEN VAN INVESTERING}) / \text{KOSTEN VAN INVESTERING}$$

### 3.1.11.2 ROI berekenen

Naar aanleiding van een meeting met collega's, kon ik de ROI niet berekenen. Omdat de Essec-groep uit meerdere bedrijven bestaat, heb ik met enkele collega's van de verschillende afdelingen een vergadering gehad in verband met de ROI. Hierbij zijn enkele punten naar boven gekomen waardoor de ROI nog niet meteen berekend kan worden.

Eén van deze punten was dat men niet meteen zag hoe men Docker kon gebruiken in productie. Om applicaties en websites te ontwikkelen, wil men wel gebruik maken van Docker omdat op deze manier het ontwikkelen veel sneller gaat. De applicaties kunnen op verschillende machines getest worden in Docker-containers. De ontwikkeling van de applicaties gaat veel sneller omdat een container op enkele seconden is opgestart.

Ook het gebruik van resources was een punt. Een host die verschillende containers draait met zware toepassingen kan meer resources verbruiken dan deze toepassingen te draaien op verschillende virtuele machines.

Wanneer een toepassing andere dependencies nodig heeft dan een andere toepassing, is het misschien wel mogelijk om een container aan te bieden met deze dependencies. De klant kan dan zijn/haar toepassing in deze container draaien wanneer het gaat om shared hosting.

Omdat de Essec-groep veel klanten heeft, zijn er ook verschillende soorten toepassingen en heel veel toepassingen. Om deze allemaal om te zetten naar containers is mogelijk, maar deze containers moeten ook beheerd worden. Als je met zo veel containers moet werken, kan het na verloop van tijd verwarrend zijn om deze allemaal te beheren.

Het besluit van de vergadering was dus dat men graag Docker wil gebruiken en dat ik hen heb aangezet om hier meer informatie over op te zoeken. Vooral in de ontwikkelfase willen ze dit integreren omdat dit op lange termijn een besparing in tijd zal zijn. Dankzij deze tijdsbesparing kan men meer werk op kortere tijd uitvoeren.

## 4 Conclusie

Dit hoofdstuk bestaat uit de bespreking van de onderzoeksresultaten en enkele aanbevelingen over het gebruik van Docker-containers. Om te eindigen is er nog een persoonlijke reflectie.

### 4.1 Bespreking resultaten

De onderzoeksvraag die bij deze stage hoort is: “zijn er *business opportunities* voor Essec-BNS door gebruik te maken van Docker-containers?”.

Uit het onderzoek blijkt dat het gebruik van Docker-containers heel wat voordelen heeft. Het proces voor het uitrollen van software gaat hierdoor sneller wat een besparing in tijd kan betekenen. Niet alleen de tijd voor het uitrollen van deze software wordt korter, ook de kosten in verband met resources kunnen vermindert worden. Als men kijkt naar de implementatie van Docker, kan deze software vooral gebruikt worden bij webapplicaties en client-server applicaties.

BNS schrijft verschillende soorten applicaties waarvoor Docker gebruikt kan worden. Zo kunnen bijvoorbeeld de applicaties die via een webbrowser gebruikt worden, ontwikkeld worden in een Docker-container en zo op de server geplaatst worden.

Docker kan niet enkel op Linux-machines gebruikt worden, maar sinds de nieuwe versie van Windows server, namelijk Windows Server 2016, kunnen Docker-containers ook gebruikt worden op het Windows-platform. Naast de Docker-containers kunnen ook Windows Containers gebruikt worden in deze nieuwe versie. Omdat Essec-BNS vooral met Windows Server werkt, kan men deze nieuwe feature toepassen voor bepaalde applicaties.

Het management van deze containers is niet altijd even makkelijk als er heel veel containers aangemaakt worden. Daarvoor bestaan er verschillende tools die gebruikt kunnen worden om deze containers te beheren. Uit het onderzoek is gebleken dat Docker Datacenter en Rancher de meest volledige tools zijn om deze containers te beheren.

Uit de SWOT-analyse blijkt dat BNS beschikt over voldoende kennis en dat men bekend is in de regio. Dit kan in hun voordeel spelen als men Docker gaat gebruiken. Bij zwaktes is te zien dat men minder kennis heeft van Linux tegenover Windows, maar dit is geen probleem meer sinds de nieuwe versie van Windows Server. Ook om de docker commando's te gebruiken, heeft men niet een uitgebreide kennis van Linux nodig.

Uit deze resultaten blijkt dat Docker makkelijk geïmplementeerd kan worden nadat men genoeg research heeft gedaan. De overschakeling van virtuele machines naar Docker-containers is een grote stap en moet goed gedocumenteerd en in kaart gebracht worden. Wanneer men weet welke applicaties in containers gestopt kunnen worden, kan BNS ervoor kiezen om deze stap te maken.

### 4.2 Aanbevelingen

Omdat er momenteel nog twijfels zijn over het gebruik van Docker zou ik Essec-BNS en HostYou zeker aanraden om de ontwikkeling van applicaties sneller te laten verlopen door Docker-containers te gebruiken in *development*. Omdat men nog niet vertrouwd is met Docker, raad ik aan om dit zeker eens te proberen binnen het bedrijf, vooral de ontwikkelaars. Docker voorziet een community-editie die de ontwikkelaars gratis kunnen gebruiken en zo projecten makkelijk op verschillende computers draaien.

Nadat men wat meer ervaring heeft met Docker, kan men stap voor stap overschakelen van virtuele machines naar Docker-containers in productie.

Het is ook mogelijk om een VPS aan te bieden aan klanten waar de Docker Engine op staat zodat klanten deze server kunnen gebruiken en hun eigen containers kunnen aanmaken op hun VPS. Zo kan een klant steeds zelf beslissen of hij containers wil gebruiken of virtuele machines.

### 4.3 Persoonlijke reflectie

Tijdens de stage heb ik onderzocht of het gebruik van Docker-containers een opportuniteit zou zijn voor Essec-BNS. Omdat ik tijdens mijn opleiding al geleerd had over Docker-containers, wist ik al redelijk veel over dit onderwerp.

De literatuurstudie was het eerste onderdeel waarmee ik begon. De informatie die ik vond over Docker-containers kwam bekend voor van de lessen Linux expert. Hoe meer ik begon op te zoeken, hoe meer nieuwe informatie ik verzamelde. Deze nieuwe informatie heeft me veel bijgeleerd hoe Docker gebruikt en geïmplementeerd kan worden. Tijdens deze literatuurstudie heb ik gemerkt dat ik bepaalde informatie niet goed begreep en deze ook niet goed kon verwoorden. Ook vond ik het moeilijk om goede bronnen te onderscheiden van minder betrouwbare bronnen. Er is zoveel informatie beschikbaar op het internet dat het niet altijd makkelijk was om deze informatie in eigen woorden te omschrijven.

Naast de literatuurstudie heb ik ook een SWOT-analyse gemaakt voor Essec-BNS. Hiervoor moest ik eerst opzoeken wat dit is om deze analyse tot een goed einde te brengen. Het was niet vanzelfsprekend om deze analyse te maken omdat hier de sterke en zwakke punten genoteerd moeten worden. Vooral het vinden van zwakke punten vond ik moeilijk.

Het laatste onderdeel van het onderzoek was een ROI-analyse. Hiervoor heb ik een meeting gehad met enkele medewerkers. Uit de vergadering is gebleken dat ik de ROI niet kon berekenen omdat men nog met enkele vragen zat. Docker voorziet een speciale calculator om deze ROI te berekenen en deze heb ik niet kunnen gebruiken.

Ik ben heel tevreden over de resultaten van de literatuurstudie en de SWOT-analyse. Deze onderdelen heb ik zo uitgebreid mogelijk uitgelegd. Bij de literatuurstudie heb ik zelf gebruik gemaakt van Docker om bepaalde informatie te verifiëren, zoals het gebruik van de beheertools. Er zijn nog wat twijfels om Docker te gebruiken in productie, maar dankzij dit onderzoek heb ik ervoor gezorgd dat men gaat bekijken of Docker al gebruikt kan worden in ontwikkeling en later in productie. Dit onderzoek heeft met andere woorden een aanzet gegeven om Docker verder te onderzoeken en eventueel te gebruiken.

## Bibliografie

- [1] „Over Essec,” Essec-BNS, [Online]. Available: <http://www.essec.be/essec.html>.
- [2] „Essec Group neemt participatie in Digital Signage bureau Mediawave,” Essec-BNS, [Online]. Available: <http://www.essec.be/nl-essec-bns-voegt-digital-signage-toe-aan-communicatiemix.html>.
- [3] „Uw partner voor hybride IT-oplossingen,” Essec-BNS, [Online]. Available: <http://mybns.com/bedrijf.html>.
- [4] „Infrastructure Solutions,” Essec-BNS, [Online]. Available: <http://mybns.com/infrastructure-solutions.html>.
- [5] „Business Solutions,” Essec-BNS, [Online]. Available: <http://mybns.com/business-solutions.html>.
- [6] „Microsoft Gold Certified Partner,” Essec-BNS, [Online]. Available: <http://mybns.com/microsoft-gold-certified-partner.html>.
- [7] „HP Gold Preferred Partner,” Essec-BNS, [Online]. Available: <http://mybns.com/hp-gold-preferred-partner.html>.
- [8] „Citrix Silver Solution Advisor,” Essec-BNS, [Online]. Available: <http://mybns.com/citrix-silver-solution-advisor.html>.
- [9] „Symantec Partner,” Essec-BNS, [Online]. Available: <http://mybns.com/symantec-partner.html>.
- [10] „Veeam Silver ProPartner,” Essec-BNS, [Online]. Available: <http://mybns.com/veeam-silver-propartner.html>.
- [11] „Cisco Authorized Reseller,” Essec-BNS, [Online]. Available: <http://mybns.com/cisco-authorized-reseller.html>.
- [12] „ Full IP Call station,” [Online]. Available: <http://www.indigocare.com/full-ip-nursecall-usa.html>.
- [13] „Home,” [Online]. Available: <https://www.hostyou.be/en/home>.
- [14] „Jaarrekeningen raadplegen,” [Online]. Available: <https://cri.nbb.be/bc9/web/catalog?execution=e2s1>.
- [15] „VMware ESXi,” [Online]. Available: [https://en.wikipedia.org/wiki/VMware\\_ESXi](https://en.wikipedia.org/wiki/VMware_ESXi).
- [16] „ESXi,” [Online]. Available: <https://www.vmware.com/products/esxi-and-esx.html>.
- [17] „Use Windows 10 Media Creation Tool to Create Installation Media or Upgrade PC,” [Online]. Available: <http://www.thewindowsclub.com/windows-10-media-creation-tool-create-installation-media-upgrade>.
- [18] „About watchguard,” [Online]. Available: <https://www.watchguard.com/wgrd-about>.

- [19] M. Rouse, „VMware AirWatch,” [Online]. Available: <http://searchmobilecomputing.techtarget.com/definition/VMware-AirWatch>.
- [20] [Online]. Available: <https://remotedesktopmanager.com/>.
- [21] [Online]. Available: [https://www.veeam.com/veeam\\_backup\\_9\\_5\\_datasheet\\_ds.pdf](https://www.veeam.com/veeam_backup_9_5_datasheet_ds.pdf).
- [22] F. Stroud, „on-premises,” [Online]. Available: <https://www.webopedia.com/TERM/O/on-premises.html>. [Geopend 2 10 2017].
- [23] „What is cloud computing?,” Microsoft Azure, [Online]. Available: <https://azure.microsoft.com/en-in/overview/what-is-cloud-computing/>. [Geopend 4 10 2017].
- [24] „Cloud computing,” Investopedia, [Online]. Available: <https://www.investopedia.com/terms/c/cloud-computing.asp>. [Geopend 05 10 2017].
- [25] M. Rouse, „Infrastructure as a Service,” TechTarget, [Online]. Available: <http://searchcloudcomputing.techtarget.com/definition/Infrastructure-as-a-Service-iaaS>. [Geopend 06 10 2017].
- [26] M. Rouse, „Software as a Service,” TechTarget, [Online]. Available: <http://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service>.
- [27] M. Rouse, „Platform as a Service,” TechTarget, [Online]. Available: <http://searchcloudcomputing.techtarget.com/definition/Platform-as-a-Service-PaaS>.
- [28] F. Stroud, „Container-as-a-Service (CaaS),” Webopedia, [Online]. Available: <http://www.webopedia.com/TERM/C/caas-containers-as-a-service.html>.
- [29] „What is Docker,” Docker, [Online]. Available: <https://www.docker.com/what-docker>.
- [30] „What is a container?,” Docker, [Online]. Available: <https://www.docker.com/what-container>.
- [31] „Containers vs VMs: Which is better in the Data Center?,” sdxCentral, [Online]. Available: <https://www.sdxcentral.com/cloud/containers/definitions/containers-vs-vm/>.
- [32] „Introduction to Containers: Concept, Pros and Cons, Orchestration, Docker, and Other Alternatives,” [Online]. Available: <https://medium.com/flow-ci/introduction-to-containers-concept-pros-and-cons-orchestration-docker-and-other-alternatives-9a2f1b61132c>.
- [33] „Introduction to Containers: Concept, Pros and Cons, Orchestration, Docker, and Other Alternatives,” [Online]. Available: <https://medium.com/flow-ci/introduction-to-containers-concept-pros-and-cons-orchestration-docker-and-other-alternatives-9a2f1b61132c>.
- [34] „When to use Docker alternatives rkt and LXD,” TechTarget, [Online]. Available: <http://searchitoperations.techtarget.com/tip/When-to-use-Docker-alternatives-rkt-and-LXD>.
- [35] „Is CoreOS' fleet container management system right for your business?,” TechTarget, [Online]. Available: <http://searchmicroservices.techtarget.com/feature/Is-CoreOS-fleet-container-management-system-right-for-your-business>.

- [36] „Is CoreOS' fleet container management system right for your business?,” TechTarget, [Online]. Available: <http://searchmicroservices.techtarget.com/feature/Is-CoreOS-fleet-container-management-system-right-for-your-business>.
- [37] „Introducing Tectonic - a Kubernetes + CoreOS Platform,” CoreOS, [Online]. Available: <https://tectonic.com/blog/announcing-tectonic/>.
- [38] „Linux Containers,” ArchLinux, [Online]. Available: [https://wiki.archlinux.org/index.php/Linux\\_Containers](https://wiki.archlinux.org/index.php/Linux_Containers).
- [39] „OpenVZ,” OpenVZ, [Online]. Available: [https://openvz.org/Main\\_Page](https://openvz.org/Main_Page).
- [40] „OpenVZ,” Wikipedia, [Online]. Available: <https://en.wikipedia.org/wiki/OpenVZ>.
- [41] „Alternatives to Docker: an overview of the container platform,” 1and1, [Online]. Available: <https://www.1and1.com/digitalguide/server/know-how/docker-alternatives-at-a-glance/>.
- [42] „How nodes work,” Docker, [Online]. Available: <https://docs.docker.com/engine/swarm/how-swarm-mode-works/nodes/>.
- [43] „Swarm mode key concepts,” Docker, [Online]. Available: <https://docs.docker.com/engine/swarm/key-concepts/#what-is-a-swarm>.
- [44] „Portainer,” Portainer, [Online]. Available: <https://portainer.io/>.
- [45] „Shipyards,” Shipyards, [Online]. Available: <https://shipyards-project.com/>.
- [46] „Overview,” Rancher , [Online]. Available: <https://rancher.com/docs/rancher/latest/en/>.
- [47] „Kitematic user guide,” Docker , [Online]. Available: <https://docs.docker.com/kitematic/userguide/>.
- [48] F. Stroud, „Docker Datacenter,” Webopedia, [Online]. Available: <https://www.webopedia.com/TERM/D/docker-datacenter.html>.
- [49] „Universal Control Plane overview,” Docker, [Online]. Available: <https://docs.docker.com/datacenter/ucp/2.2/guides/>.
- [50] „Deep dive into Windows Server Containers and Docker – Part 2 – Underlying implementation of Windows Server Containers,” Xebia, [Online]. Available: <http://blog.xebia.com/deep-dive-into-windows-server-containers-and-docker-part-2-underlying-implementation-of-windows-server-containers/>.
- [51] B. Monteiro, „The Similarities and Differences Between Windows and Linux Containers,” Rancher, [Online]. Available: <http://rancher.com/the-similarities-and-differences-between-windows-and-linux-containers/>.
- [52] „Windows Containers,” Microsoft, [Online]. Available: <https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/>.

- [53] „HPE Nimble Storage Docker Volume Plugin,” Docker, [Online]. Available: <https://store.docker.com/plugins/nimble?tab=description>.
- [54] „Amazon EC2 Container Service,” Amazon, [Online]. Available: <https://aws.amazon.com/ecs/>.
- [55] „Product Details,” Amazon Web Services, [Online]. Available: <https://aws.amazon.com/ecs/details/>.
- [56] „Container Engine,” [Online]. Available: <https://cloud.google.com/container-engine/>.
- [57] „Azure Container Service (AKS),” [Online]. Available: <https://azure.microsoft.com/en-us/services/container-service/>.
- [58] „MANAGING PERSISTENCE FOR DOCKER CONTAINERS,” [Online]. Available: <https://thenewstack.io/methods-dealing-container-storage>.
- [59] „Live Containers Migration Across Data Centers: AWS and Azure Integration,” [Online]. Available: <https://blog.jelastic.com/2015/12/14/live-containers-migration-across-data-centers-aws-and-azure-integration/>.
- [60] „Pushing and Pulling to and from Docker Hub,” [Online]. Available: <https://ropenscilabs.github.io/r-docker-tutorial/04-Dockerhub.html>.
- [61] D. O. Reilly, „How to Create a Docker Backup With Morpheus,” [Online]. Available: <https://www.morpheusdata.com/blog/2017-03-02-how-to-create-a-docker-backup-with-morpheus>.
- [62] L. Degraer, „Wat is een Disaster Recovery Plan en welke stappen zijn er voor nodig?,” [Online]. Available: <https://blog.aurelium.be/wat-is-disaster-recovery-plan-welke-stappen>.
- [63] „Backups and disaster recovery,” [Online]. Available: <https://docs.docker.com/datacenter/ucp/2.2/guides/admin/backups-and-disaster-recovery/#ucp-and-swarm>.
- [64] V. Saraswat, „Setting up HA on your DDC Deployment,” [Online]. Available: <https://blog.docker.com/2016/05/high-availability-architecture-docker-datacenter/>.
- [65] „SWOT-analyse - Wiki,” [Online]. Available: <https://www.frankvanormondt.nl/advies/innovatie/wiki/swot-analyse/>.
- [66] [Online]. Available: <https://remotedesktopmanager.com/>.

