

Onderwijsgroep
Professionele Opleidingen
Handelswetenschappen
en bedrijfskunde

SAP Deployment Automation
Thesis: CI/CD Deployment in Hybrid Cloud

Bachelorproef en Stageverslag aangeboden door
Denys Slyvka tot het behalen van de graad van
Bachelor in de Toegepaste Informatica

Begeleider Odisee: Yvan Rooseleer

Stagementor: Gert Van De Slijke

Academiejaar 2020 - 2021

Voorwoord

Dit document is bedoeld voor het behalen van een diploma van professionele bachelor in de Toegepaste Informatica binnen de hogeschool Odisee te Brussel. De voorbereiding en onderzoek van deze thesis is het hoofddoel van het derde en laatste jaar in de opleiding. Het onderwerp dat gekozen is voor de uitvoering binnen deze thesis leunt sterk naar persoonlijke interesses en wil tot ontwikkeling van eigen vaardigheden en kennis.

Dit verslag is het resultaat van een complete stage en een sterk verdiepend onderzoek naar eerder niet geleerde toepassingen. Het schrijven van de thesis duurde langer dan verwacht omdat een groot aantal gepresteerd materiaal gefilterd en duidelijk beschreven moest worden.

Deze thesis tracht voor een verbetering te verwezenlijken voor zowel klanten, als BC-teamleden voor bestaande en toekomstige SAP applicaties. Dit impliceert naar dagdagelijkse taken te automatiseren en migraties of back-ups van applicaties, vlotter te laten verlopen. Ook is het een interessant pakket aan leerstof dat verworven kan worden.

Ik wil iedereen die geholpen heeft tijdens de voorbereiding, uitvoering en verslaggeving van harte bedanken. Allereerst Gert Van De Slijke en Yvan Rooseleer, de personen die een grote rol hebben gespeeld in de uitwerking van dit project. Ik kon steeds bij beide partijen terecht met vragen of opmerkingen. Ook werd mijn evolutie door hen stipt gevolgd waardoor deze thesis goed gelukt is.

Vervolgens dank ik Patrick Van Den Broeck bij het helpen kiezen van een verdiepend onderwerp. Door de heer Van Den Broeck heb ik een sterke steun in de rug en motivatie gekregen voor het uitwerken van de eindproef.

Ten slotte dank aan Werner de Schryver, de andere stagiaire bij Flexso die ook gewerkt heeft aan een soortgelijke opdracht. Dankzij Werner heb ik beter inzicht gekregen in verschillende vormen van aanpak en kon ik sneller presteren omdat hij soms hielp bij het oplossen van complexe problemen.

Als laatste dank ik nog Rayen Nasra en Celien Otte voor de sterke steun tijdens deze periode. Ik kon steeds bij hen terecht in moeilijke momenten. Zij motiveerden mij elke stap in deze uitvoering.

Denys Slyvka
Juni 2021

Samenvatting

Deze thesis is een bondige verslaggeving over een verbetering van een langdradige en ingewikkelde installatie van SAP-ERP systemen. Door het resultaat dat behaald werd tijdens de uitvoering bestaat de kans tot een foute installatie niet meer. Het project heeft de werklust van meerdere consultants binnen Flexso sterk verlicht waardoor de mentale capaciteit gefocust kan worden op andere belangrijke taken in de job.

De uitwerking van de stage tool is gebeurd met verschillende technologieën. Deze worden binnen dit verslag geanalyseerd en beschreven:

- SAP Systemen
- Terraform
- Ansible
- Scripting
- Cloud infrastructuur
- Schematiseren

De hoofdpdracht van de stage heeft te maken met automatische *deployment* van een klassiek ERP-systeem zoals SAP door het benutten van nieuwe methodes, bv. tooling voor Infrastructure as Code (IaC).

De ervaringen van de stage hebben aanleiding gegeven tot verdere verdieping. Als verdiepend onderwerp, werd gekozen om een stap verder te zetten in de uitwerking van de stage tool door het toepassen van CI/CD en het uitvoeren van een installatie in een hybrid cloud omgeving. Het volgende wordt besproken in het verdiepen stuk van dit verslag:

- Ontwerp van diverse soorten cloud opstellingen
- Vergelijking tussen Azure en Google Cloud
- Methodes en tools voor provisioning
- Connectiviteitsoplossingen voor multi hybrid cloud
- Toepassing van continuous integration, delivery, *deployment*
- Scripting van Jenkins pipelines

De bovenstaande onderwerpen dienen als basis en achtergrond voor de uitwerking van volledig geautomatiseerde CI/CD *deployment* in een hybrid cloud omgeving.

Doorheen het verslag wordt ook vaak verwezen naar documenten in bijlage. Dit zijn documenten waar gedetailleerde beschrijvingen te vinden zijn over bepaald onderzoek of uitvoering.

Dit verslag beschrijft een gerealiseerd technologisch concept en is nuttig voor Flexso NV. Vanuit deze uitwerking is er een toegevoegde waarde die door Flexso gehanteerd wordt om de dagdagelijkse druk voor BC-teamleden sterk te verminderen.

Inhoudsopgave

SAP Deployment Automation	
Thesis: CI/CD Deployment in Hybrid Cloud	1
Voorwoord	2
Samenvatting	3
Inhoudsopgave	4
Begrippen	7
Inleiding	8
Opdrachtgever	9
Stageopdracht	10
Actieplan	11
Gantt-Chart	11
Stagevoorbereiding	11
Stage Uitvoering	12
SAP Automation Tool	13
Struikelblokken	14
Rapportering	15
Vooronderzoek	15
Experimenteren met mogelijke oplossingen	15
Selectie van Tools	15
Installatie Automatiseren	16
SFTP Server	16
HANA EXPRESS	16
HANA DATABASE	17
NETWEAVER ABAP	17
ABAP FNDN	17
JAVA ASE	18
Afwerking	18

Verdieping	19
Onderzoeksvraag	19
Aanpak	19
Soorten Cloud	20
Public Cloud	20
Private Cloud	21
Hybrid Cloud	22
Hybrid Multi Cloud	23
Advies	23
Azure VS Google Cloud Platform	24
Azure	24
Google	24
Virtualisering	25
Prijzen	25
Advies	26
Provisioning	27
Terraform	27
Ansible	28
Vagrant	28
Advies	28
Connectiviteit	29
SDN	29
SD-WAN	30
Advies	30
Volledige Automatisering	31
Continuous Integration	31
Continuous Delivery	31
Continuous Deployment	31

Advies	31
Jenkins	32
Installatie	32
Configuratie	32
Werking	32
Advies	32
Proof of Concept	33
Opzet	33
Proces	33
Reflectie	34
Conclusie	35
Bronnen	36

Begrippen

SAP: Systems Applications and Products

POC: Proof Of Concept

BC-team: Basis Components Team

SSH: Secure Shell

DRP: Disaster Recovery Plan

GCP: Google Cloud Platform

SFTP: Secure File Transfer Protocol

IaC: Infrastructure as Code

HCL: HashiCorp Configuration Language

CLI: Command Line Interface

ABAP: Advanced Business Application Programming

FNDN: Foundation

ASE: Adaptive Server Enterprise

HDBLCM: Hana Database Lifecycle Management

RHEL: Red Hat Enterprise Linux

GB: Gigabyte

RAM: Random Access Memory

TB: Terabyte

VCPU: Virtual Central Processing Unit

SWPM: Software Provisioning Manager

SPOF: Single Points Of Failures

Inleiding

Flexso is een bedrijf dat zich specialiseert in het ontwerpen van SAP-toepassingen voor verschillende business lijnen. Naast het maken van SAP software voor verschillende klanten, is Flexso ook sterk gericht op innovatie en vooruitgang in de markt.

De stageopdracht gaat over het uitrollen van een gestandaardiseerde template die je kan aanpassen met een geparametriseerd dashboard. Dit zou de *deployment* van de demo-omgevingen en projecten vereenvoudigen. Het framework dat dan wordt opgezet is dan ook later makkelijk herbruikbaar voor de opzet van nieuwe SAP omgevingen.

Om dit te realiseren moet allereerst gewerkt worden met ontwikkelaars en systeembeheerders aan een oplossing die eenvoudig en polyvalent is in gebruik. Verder wordt nagegaan welke typische routine installatie taken geautomatiseerd kunnen worden zodat deze minder tijd innemen van de dagdagelijkse taken van een systeembeheerder. Daarvoor voorziet Flexso testomgevingen in Azure naar gelang de behoefte voor het uitvoeren van de opdracht.

In vogelvlucht gaat de opdracht over inzicht en optimalisatie in de dagdagelijkse taken van een SAP systeembeheerder en actieve medewerking aan een innovatieve oplossing, gebaseerd op de nieuwste Cloud en SAP technologie (Ansible, Powershell, Bash-scripting, Terraform...).

Het verdiepend onderwerp is gefocust op een CI/CD aanpak van de installatie van SAP-applicaties in een hybrid cloud omgeving. Dit gebeurt door een onderzoek te doen over diverse soorten cloud opstellingen. Verder worden cloud providers zoals Azure en Google Cloud geanalyseerd op basis van methodes en tools voor provisioning. Vervolgens wordt er beschreven waarom connectiviteitsoplossingen voor multi hybrid cloud eerder niet bestonden en de reden dat een hybrid cloud concept nu wel mogelijk is. Ten slotte wordt een CI/CD pipeline in Jenkins toegepast. Hiermee wordt bewezen dat CI/CD een meerwaarde biedt aan de algemene structuur en architectuur van een hybrid cloud omgeving van eender welk bedrijf of organisatie. Verder in de paper wordt het onderzoek en uitvoering beschreven.

Alle bovenstaande onderwerpen worden in deze paper uitgebreid aangepakt en gedocumenteerd. SAP is een nieuwe en innovatieve technologie dat nu al een sterke invloed heeft op de business sector door het optimaliseren en visualiseren van bedrijfsprocessen.

Doorheen de thesis wordt verwezen naar verschillende bijlagen. Deze dienen als achtergrond-informatie en verdieping en zijn een aanvulling van de thesis.¹

¹ [Lijst met bijlagen](#)

Opdrachtgever

De gekozen opdrachtgever maakt deel uit van de Cronos groep en heeft een interessante opdracht waar veel diepgang en ontwikkeling kan in gemaakt worden.

Voor het tekenen van het stagecontract heb ik uitgebreid onderzoek uitgevoerd naar de activiteiten en waarden van Flexso. Dit moet deels vanuit de opleiding, maar zelf wou ik ook weten waar ik zou belanden. Verder in het jaar is nog een ander integriteitsonderzoek uitgevoerd waarin de waarden, normen en cultuur van zowel Flexso als de werknemers onder de loep genomen zijn.

Flexso is een bedrijf dat zich specialiseert in het toepassen van SAP-software voor verschillende business lijnen. Naast het maken van SAP software voor verschillende klanten is Flexso ook sterk gericht op innovatie en vooruitgang in de markt. Sinds 2004 is het bedrijf een officiële “full service SAP Gold Partner”. Dit betekent dat het stagebedrijf een specialisatie biedt in alle mogelijke vormen en lijnen van SAP zowel lokaal als in de cloud. Dankzij het partnerschap met SAP krijgt Flexso ook de mogelijkheid om als eerste nieuwe technologieën toe te passen om zo innovatieve SAP oplossingen te ontwerpen.

Sinds 2014 is Flexso ook erkend als “SAP Partner of Excellence” waardoor het bedrijf steun kan bieden in de SAP omgeving en software aan alle 140 actieve klanten in meer dan 40 landen.²

Naast het aanbieden van verschillende oplossingen aan de klanten denkt Flexso ook na over het tekort van IT medewerkers in de markt. Om dat probleem op te lossen organiseert het bedrijf sinds 2004 in samenwerking met VDAB 2 keer per jaar een SAP-consultancy opleiding van 4 maand waarbij een persoon zonder technische ervaring op het einde van de opleiding een job kan vinden binnen de IT-sector. Dankzij deze samenwerking zijn meer dan 300 personen opgeleid tot SAP consultant.³

Binnen Flexso wordt er sterk gelet op een horizontale omgeving zowel in de “hiërarchie” als de fysieke vormgeving van het gebouw. Het kantoor is gelegen op het gelijkvloers in een van de Cronos-gebouwen. Alle werknemers werken allemaal in open ruimtes zonder muren of afscheidingen. De directeur zit alsook te werken in de open ruimtes tussen de medewerkers. Deze manier van aanpak versterkt de band tussen de werknemers zelf en de directie. Het sociaal aspect is hier sterk van belang.⁴

Het stagebedrijf is een ideaal voorbeeld van een ‘stabiel IT bedrijf’ waar de focus niet enkel ligt op de cijfers en inkomsten, maar ook op het welzijn van de werknemers en stagiaires. Flexso toont aan dat ook een ‘groot’ bedrijf met meer dan 300 werknemers kan zorgen voor een familiale omgeving. Als startende IT-er is dat een ideale startplek waar makkelijk ervaring en kennis opgenomen kan worden.

² (Flexso, 2020)

³ (Goossens, 2020)

⁴ [Paper Ethiek Integriteit](#)

Stageopdracht

De focus van de stageopdracht is gericht op de vereenvoudiging van herhalende processen die vaak langer duren dan verwacht. De stage tool dient vooral een POC te zijn, het BC-team moet daarop verder kunnen bouwen en aanpassingen uitvoeren voor verschillende scenario's.

De initiële opdracht gaat over het uitrollen van een gestandaardiseerde template die je kan aanpassen met een geparametriseerd dashboard. Het framework dat wordt opgezet is later makkelijk herbruikbaar voor de opzet van nieuwe SAP omgevingen bij klanten.

Daarbij werd gedacht aan het automatisch uitrollen van servers via een dashboard. Tevens in de cloud of op locatie opzetten van nieuwe SAP-systemen a.d.h.v. een template. Daarnaast ook het automatiseren van Failover (als een server uitvalt dan kunnen andere servers overnemen⁵) en DRP (Disaster Recovery Plan is een gedocumenteerd plan om de IT-omgeving in een bedrijf te beschermen of herstellen van een technisch probleem⁶). De uitrol moet zowel op Linux als Windows Systemen werken

Tijdens het uitwerken van de opdracht werden enkele aanpassingen uitgevoerd aan de initiële opdracht. Een belangrijke aanpassing is het schrappen van Failover en DRP aangezien dat relatief makkelijk te implementeren is volgens mijn Flexso stagebegeleider. Een andere aanpassing is het ontwerpen van de automatisering binnen de Azure en Google Cloud platformen. Hoewel ik deels werk met Google Cloud is dat deel uitgevoerd door een andere stagiaire. Mijn focus ligt volledig op het ontwerpen van automatiseringen binnen de Azure cloud infrastructuur.

Om het BC-team een beter beeld te geven hoe de code in elkaar zit, zijn meerdere versies van SAP-systemen in kaart gebracht. Dit wordt gedaan om zo de verschillen binnen de code te kunnen aantonen voor toekomstige aanpassingen en uitbreidingen van andere types SAP-systemen.

Een laatste aanpassing is het schrappen van de Windows uitrol, deze is overbodig en ging niet gebruikt worden aangezien het grootste deel van de Flexso klanten liever Linux gebruiken.

De stageopdracht is heel wat veranderd doorheen de stageperiode, alle aanpassingen zijn de eerste dag besproken. Door het aantal aanpassingen ben ik zelf ook steeds meer beginnen denken aan een andere soort van aanpak. Zelf heb ik op een passieve manier geleerd om aanpassingen te accepteren en werk te laten vallen om een nieuwe aanpak te kunnen uitvoeren.

⁵ (Computerwoorden.nl, 2020)

⁶ (Cheops, 2020)

Actieplan

In dit gedeelte behandel ik de stagevoorbereiding en de uitwerking van een actieplan met behulp van een Gantt-Chart. Alle stappen die doorheen het jaar gemaakt zijn worden hier kort besproken.

Gantt-Chart

Het actieplan is een Gantt-Chart waar ik mijn vooruitgang van verschillende taken voor, tijdens en na de stage kan bekijken en aanpassen. In dit deel beschrijf ik kort hoe het plan ingedeeld is.⁷ Het gebruik van een Gantt-Chart zorgt voor een vlotte opvolging van de taken door visuele representatie. Zo kan ik in een ogenblik weten wat de situatie op het moment van analyse is.

Stagevoorbereiding

De stagevoorbereiding bevat alle stappen die uitgevoerd zijn voor het begin van het 2de semester en de stage. Tijdens deze periode zijn de volgende taken uitgevoerd;

- **Bedrijfsprofiel** - Het opstellen van mijn persoonlijke interesses, kerndoelen en keuzeprofiel.
- **Bedrijf** - CV opstellen, schrijven van een standaard motivatiebrief, versturen van mails met vraag voor stage en onderzoek van gevonden stagebedrijf Flexso.
- **Begeleidende docent** - Yvan Rooseleer contacteren en vragen om mijn begeleider te zijn.
- **Stage opdracht** - Analyseren van de stageopdracht, achtergrond en nut van de opdracht proberen achterhalen, indienen van het voorstel en laten keuren door de docenten van Odisee.
- **Stage contract** - Contractuele aspecten van de stage in orde brengen, handtekeningen van alle partijen verzamelen. Algemene afspraken en formaliteiten bespreken.
- **Verdiepend onderwerp** - Onderzoek naar een verdieping die gerelateerd is aan de stageopdracht. Overleggen met docenten over het onderwerp, raad vragen en onderzoek doen om een beter beeld te krijgen over de verschillende technologieën. Opmaken van een algemene map voor alle deliverables.

⁷ [Actieplan](#)

Stage Uitvoering

De volgende delen van het actieplan worden tijdens het uitvoeren van de stage steeds aangepast. Dit is ook het grootste deel van het plan waar in detail alle acties voor het voorbereiden, uitvoeren en afwerken van de stageopdracht en verdiepend onderwerp opgelijst worden.

- **Vooronderzoek** - Onderzoek naar SAP technologie, eerste gesprek met stagebegeleider aanvragen / uitvoeren, structuur opstellen in de bestanden en verdiepend onderwerp afbaken.
- **Stage** - Omgeving, collega's, team leren kennen, Azure en GCP introductie cursussen doornemen, verschillende SAP cursussen doornemen. Installaties van een tal SAP systemen manueel uitvoeren, gebruikte kennis van de manuele installaties toepassen tijdens de automatische uitrol. Terraform, Ansible, Powershell scripts schrijven voor de automatische installatie. Schrijven van documentatie.
- **Verdiepend onderwerp uitvoering** - Onderwerp onderzoeken, literatuurstudie, uitwerking van een werkend concept binnen Jenkins.
- **Afwerking** - Alle kennis en documentatie verzamelen, mappen opvullen met deliverables. Maken van een eindverslag en presentatie.

Het actieplan is doorheen het jaar een belangrijk document dat steeds aangepast wordt in geval van vooruitgang of tegenslag. Naast het actieplan is ook een logboek⁸ bijgehouden waar in detail elke dag van de stage beschreven is.

⁸ [Logboek](#)

SAP Automation Tool

Dit deel beschrijft kort de volledige samenwerking van alle gebruikte tools die gebruikt zijn in het finaal ontwerp. Verder in het verslag wordt elke technologie in detail beschreven en uitgelegd.

In het schema hieronder wordt het proces van een *SAP template deploy* in kaart gebracht. Dit is een visuele representatie van het eindproduct dat doorheen de hele periode werd ontworpen.



1 - SAP Automatische deploy

Alles begint vanuit de kant van de gebruiker. Voor het starten van een 'deploy' moet de gebruiker enkele cruciale parameters ingeven zoals namen, ID's en nummers van virtuele machines en/of SAP-systemen. Dit is ook het enige moment waar *user-input* nodig is tijdens het volledige verloop. In de gebruikershandleiding wordt in detail besproken welke stappen de gebruiker moet doornemen.⁹

Vervolgens gaat het juiste Terraform-script een tal van resources aanmaken binnen de infrastructuur van de gekozen cloud provider. Eens alle resources correct aangemaakt zijn zal Terraform verbinden met de aangemaakte Virtuele Machine via een SSH-verbinding, een veilige en geëncrypteerde verbinding door het internet. Eens de verbinding gelukt is worden alle pakketten en updates geïnstalleerd om de Linux omgeving voor te bereiden op het downloaden van het Ansible script vanuit Github¹⁰ en vervolgens het uitvoeren van het script met de Ansible toepassing.

Ansible zorgt voor het downloaden van de juiste SAP pakketten van de SFTP-server dat ook op maat gemaakt is en enkel dient voor SAP installatiesoftware. Na het downloaden worden alle gearchiveerde mappen geopend. Ten slotte gebruikt Ansible een parameterbestand van SAP. Deze dient om de installatie en configuratie van een SAP-systeem volledig autonoom te laten doorlopen met alle input van de gebruiker dat in het begin van het proces is meegegeven.

De SAP installatie duurt gemiddeld één uur, dit voor een heel basis instantie van een systeem zonder enige replicatie. Na het installeren kan de gebruiker inloggen op de VM of een SAP GUI tool gebruiken om de connectiviteit en werking van het systeem te controleren.

Dit was een representatie van het hele proces dat doorlopen wordt tijdens het automatisch deployen van virtuele machines. Het proces duurt gemiddeld 1 tot 2 uur afhankelijk van de grootte van de systemen en de configuratie. Meer hierover in de technische documentatie.¹¹

⁹ [Gebruikershandleiding](#)

¹⁰ [Github repository](#)

¹¹ [Technische documentatie](#)

Struikelblokken

Tijdens het maken van de stageopdracht zijn verschillende struikelblokken overwonnen. In de technische documentatie worden alle struikelblokken gedetailleerd beschreven.

Een eerste struikelblok is het kennen en kunnen onderscheiden van de verschillende SAP-systemen. De samenwerking ervan en het nut van elke versie. Om deze struikelblok te overwinnen heb ik een tal van cursussen over SAP installaties en algemene uitleg over de werking van SAP systemen gelezen. Er is beslist om verder te gaan met de stageopdracht en tijdens het uitvoeren vragen te stellen aan de stagebegeleider bij geval van twijfel of te weinig kennis over bepaalde software en/of pakketten. Dat blijkt een goede keuze te zijn aangezien geen diepe kennis van SAP nodig is voor het automatiseren van de installaties.

Azure is een tweede struikelblok. De stage is het eerste contact met de omgeving. De overvloed aan functies en mogelijkheden is overweldigend voor een beginnende stagiaire. Het uitwerken van een initieel POC is mogelijk gemaakt door het volgen van Azure Essentials cursussen om verschillende functies van Azure te begrijpen en te kunnen toepassen.

De kennis van Azure is zeker nuttig bij de uitwerking van Terraform. De nieuwe IaC taal is een gesloten boek waardoor de eerste uitwerking en toepassing van Terraform in de Azure omgeving moeilijk is verlopen. De werking van de software is onbekend. Er is snel besloten om enkele video's te bekijken en de officiële documentatie door te lezen om een betere inzicht te ontwikkelen over de fundamentele waarden en functies van de HCL declaratieve configuratie bestanden. Verder ging de installatie van Terraform moeilijk. Door een foute interpretatie van de stappen duurde de installatie langer dan verwacht.

Een andere grote struikel is het leren werken met Ansible en het uitschrijven van commando's voor de automatisering. Meerbepaald de werking van Ansible *pexpect*. Een manier om automatisch op prompts te antwoorden met Ansible code. Om *pexpect* te laten werken moeten verschillende pakketten geïnstalleerd worden. De commando's voor de installatie daarvan zijn niet te vinden in de officiële documentatie waardoor vele uren verloren zijn gegaan aan zoekwerk op het internet. Ten slotte is ondervonden dat Ansible *pexpect* niet nodig is voor het beantwoorden van de prompts wanneer er geen prompts getoond kunnen worden. Dat is mogelijk door het pushen van SAP configuratiebestanden **“.rsp”** en **“.params”** en deze met de SAP installatie media uit te voeren.

Naast het leren van de nieuwe talen is ook het beheren van een Linux-systeem soms moeilijk te begrijpen. Vooral de volgorde van schijven. Daarmee wordt bedoeld de logische en fysieke partities of mounts. Dat is deels omdat de schijven aangemaakt worden in de cloud en anders geïnterpreteerd kunnen worden door Linux zelf. Er zijn tot slot een tal van commando's gevonden die alle schijven en partities juist indelen en het systeem klaarmaken voor SAP opslag.

Dit zijn enkele grote struikelblokken tijdens het uitvoeren van de stageopdracht. Elke dag komt steeds een ander probleem boven dat eerst moet opgelost worden. Het aantal kleine 'issues' is te groot om vermeld te worden in dit stageverslag. Een oplistings hiervan is te vinden in het logboek.

Rapportering

De totale duur van de stage is 15 weken. In deze periode is de uitwerking van de stageopdracht verdeeld in kleine, behapbare delen. Dit deel beschrijft in detail de doelstellingen, moeilijkheden, oplossingen en afwerkingen van het automatiseringsproces.

Vooronderzoek

De eerste weken zijn toegewezen aan het gewoon worden van de stage en de collega's. Daarnaast ging een groot deel van de tijd naar het lezen van verschillende cursussen. De eerste cursus is van Azure Essentiels en bestaat uit 6 delen. Elk deel gaat over een specifieke groep functies binnen Azure, bv; Database, Compute, VM, vervolgens heeft de Flexso stagebegeleider een cursus over de werking van SAP genaamd "SAPTEC" gegeven. In deze cursus staat in detail beschreven hoe SAP werkt, welke hardware benodigdheden een systeem heeft en hoe deze worden verdeeld. De cursus is bedoeld om meer achtergrondinformatie te krijgen over SAP en het project evidenter te laten verlopen. De laatste cursus gaat over een installatie van een S4/HANA systeem, een specifieke database versie van SAP. De cursus vormt een stevige fundering van kennis over SAP installatie waarmee ik verder kan werken. Ten slotte is bijkomend onderzoek uitgevoerd naar eventuele technologieën die automatisering in de cloud mogelijk maakt.

Experimenteren met mogelijke oplossingen

Naast het lezen van de cursussen is ook een deel van de werktijd gegaan naar het oefenen en "ontdekken" van Azure. Zo is Terraform al op dag 2 ontdekt, alleen is dat binnen de Azure cloud CLI. Door het onderzoek dat eerder uitgevoerd is, probeer ik een Azure ARM *deployment* van een SAP HANA systeem uit te voeren.¹² Deze is gelukt. Dat bewijst dat Azure automated *deployments* mogelijk zijn en steeds meer toegepast kunnen worden op de markt. In deze stap is nog een Github repository aangemaakt geweest.¹³ Deze stap is belangrijk omdat hier het proces van de probleemoplossing begint. Verder in de uitwerking zal deze vaardigheid steeds meer gebruikt worden.

Selectie van Tools

Volgens de opdracht is Vagrant een tool die gebruikt wordt om de automatisering mogelijk te maken. Na uitgebreid onderzoek is gebleken dat Vagrant geen passende tool is en Terraform best geschikt is voor het voltooien van de opdracht.

¹² [AZURE ARM installatie](#)

¹³ [GitHub Repo](#)

Installatie Automatiseren

Na de voorbereiding en selectie van tools gaat het tweede en belangrijkste deel van de stageopdracht van start. In dit deel worden verschillende SAP toepassingen geautomatiseerd.

- **SFTP Server:** Maken van een server voor het opslaan van alle SAP installatiepakketten.
- **HANA EXPRESS:** Kleinste versie van SAP, goedkoop om te hosten in de cloud.
- **HANA DATABASE:** De eerste (grote) versie van een SAP toepassing dat ook een database is.
- **NETWEAVER ABAP:** SAP toepassing dat samen met HANA DATABASE werkt.
- **ABAP FNDN:** Vernieuwde versie van NETWEAVER ABAP.
- **JAVA ASE:** SAP met een Java kern en een ander type database (ASE).

SFTP Server

Voor het opslaan van SAP installatiebestanden heb ik een SFTP server aangemaakt. De server is een centrale plek waarvan alle downloads gebeuren. Door mijn betere kennis in Azure is het gelukt om de server snel op te zetten en alle SAP bestanden daarin op te slaan.

HANA EXPRESS

De keuze om HANA EXPRESS (de meest primitieve versie van SAP) te installeren komt door 2 redenen. Deze versie van SAP heeft lage 'resource' behoeftes. Daarom is er beslist om eerst de installatie hiervan te proberen om zo te leren werken met Terraform en Ansible. Voor de automatische installatie heb ik eerst een volledige manuele installatie uitgevoerd.¹⁴

Vervolgens start ik met een onderzoek naar de werking van Terraform in Azure. De bedoeling is het opstellen van een script dat lokaal uitgevoerd kan worden. Dat is mogelijk door het aanmaken van een Azure contributor, een speciaal account voor automatiseringen binnen Terraform. Vervolgens moeten alle uitgevoerde commando's die tijdens de manuele installatie ingevoerd zijn kunnen werken binnen Terraform en Ansible.

Er is beslist om een duidelijk onderscheid te houden tussen het *provisioning* van een systeem en het installeren van de SAP-software. Tijdens het overzetten van alle commando's is een probleem ontdekt bij het automatisch antwoorden op prompts tijdens de SAP installatie. De oplossing daarvoor ligt in Ansible *expect*. Hoewel *expect* een mogelijkheid biedt, is besloten een andere uitwerking toe te passen a.d.h.v. een *config* bestand waarin alle waarden die SAP moet hebben ingegeven kunnen worden.

Ten slotte kan het bestand door de installatie media (**HDBLCM**) van SAP ingelezen en geïnstalleerd worden zonder enige tussenkomst van de gebruiker. Ansible *expect* is geschrappt en ik ben beginnen zoeken naar de commando's waarmee ik de installatie automatisch kon uitvoeren.

¹⁴ [HANA Express manuele installatie](#)

HANA DATABASE

De installatie van HANA DATABASE gebeurt op een RHEL 8.2 systeem. Voor de installatie van het systeem is een minimum aan 160GB RAM, 1 TB aan schijfruimte en 12 vCPU's nodig. Deze specificaties hebben een hogere quota vereiste waardoor ik enkele dagen moet wachten op de rechten in Azure.

Na het krijgen van een juist aantal quota's kan ik nog steeds geen contributor aanmaken voor de automatische resource *deployment* met Terraform. Daardoor moet een andere aanpak toegepast worden omdat ik als student geen rechten kan krijgen voor het aanmaken van een contributor met de Azure abonnement dat genoeg quotas heeft. Door het bestuderen van de Azure documentatie is er een oplossing bedacht¹⁵ die de Azure CLI lokaal download. Vervolgens kan een gebruiker lokaal inloggen waardoor het Terraform script enkel het Azure **subscription ID** en **tenant ID** nodig heeft om de *deployment* te kunnen starten en uitvoeren.

Door het oplossen van alle problemen kan ik een RHEL systeem aanmaken met het Terraform script om zo de manuele installatie van HANA uit te voeren. Om commando's te kunnen uitvoeren in een 8.x systeem is een login van een Red Hat abonnement nodig. Deze is geleverd door een ander BC teamlid. Na het registreren kan de hele installatie van HANA doorlopen worden. Deze is identiek aan de HANA Express versie. Het enige verschil zijn de installatie bestanden die gedownload worden van de SFTP server. De Automatische installatie is ook identiek aan HANA Express, alleen duurt deze langer omdat het aantal functies ook hoger ligt.

NETWEAVER ABAP

Op hetzelfde systeem waarop HANA geïnstalleerd is voer ik een manuele installatie van Netweaver ABAP uit.¹⁶ Voor de installatie moeten eerst alle bestanden die nodig zijn gedownload worden met de SAP Download Manager. Deze zijn vervolgens geüpload naar de SFTP server. Na het downloaden van de bestanden op het RHEL systeem kan de SWPM installatie gestart worden. Tijdens de installatie worden enkele pakketten zoals de kernel bestanden niet gevonden. Deze moeten apart opgezocht en gedownload worden. Na het doorlopen van de hele installatie cyclus kon het **.params** bestand geüpload worden naar de SFTP server.

ABAP FNDN

Gelijk aan het proces van Netweaver ABAP voerde ik een manuele installatie van de FNDN versie uit. Daarvoor moeten andere bestanden gedownload worden met de download manager. Tijdens de installatie zijn ook andere bestanden nodig die apart opgezocht en geïnstalleerd moeten worden. De automatisering van het proces was gelijk aan die van Netweaver ABAP. De automatische installatie verliep foutief de eerste keren omdat het systeem een reboot nodig had. Dat probleem werd met de hulp van de BC teamleden snel opgelost.

¹⁵ [Azure CLI oplossing](#)

¹⁶ [Manuele installatie van NW ABAP 1709](#)

JAVA ASE

De installatie van JAVA ASE is heel sterk lijkend op de ABAP installaties. Het enige verschil tussen de versies zijn de installatie bestanden die gebruikt moeten worden. Naast de SAP JAVA bestanden zijn ook ASE database bestanden nodig. Het vinden van de juiste downloads neemt een groot deel van het manueel installatieproces in. Dankzij de hulp van de stagebegeleider zijn de laatste bestanden voor de JAVA ASE installatie gevonden. De manuele installatie van JAVA¹⁷ gaat vlot en zorgt voor weinig problemen.

Tijdens de uitwerking van de automatische installatie ontstaat een probleem waarbij de parameter bestanden niet ingelezen kunnen worden. Dat komt door de geëncrypteerde hash sleutel in het begin van het bestand dat niet gelezen kan worden. Daarvoor moet nog een bestand met de hash sleutel in dezelfde map geplaatst worden. Het bestand moet een juiste sleutel bevatten om de decryptie te kunnen uitvoeren. Deze worden normaal steeds samen aangemaakt.

Afwerking

De laatste opdrachten die ik krijg van de stagebegeleider focussen op het uitschrijven van een gebruikershandleiding voor iedereen dat zelf een deploy wil proberen. Daarnaast een technische documentatie schrijven voor het BC team waar in detail elke lijn code uitgelegd wordt.¹⁸¹⁹ Naast de documentatie is nog een onepager²⁰ gemaakt die de hele opdracht op 1 A4 blad uitlegt. Tot slot nog een finale presentatie²¹ voor Flexso waar alle deliverables besproken worden.

¹⁷ [Manuele installatie van JAVA ASE](#)

¹⁸ [Gebruikershandleiding](#)

¹⁹ [Technische documentatie](#)

²⁰ [Onepager](#)

²¹ [Flexso presentatie](#)

Verdieping

In dit deel wordt kort de aanpak en verdeling van de verdieping beschreven. Vanaf dit deel begint de verdieping en het onderzoek naar een verdiepend onderwerp.

De verdieping van de stageopdracht en verdiepend onderwerp bestaat uit het leren van nieuwe technologieën zoals Terraform en Ansible, maar ook het uitvoeren van literatuurstudie over het verdiepend onderwerp en de uitvoering van een concept dat gelinkt is aan de verdieping.

Onderzoeksvraag

Hoe wordt een geautomatiseerd *deployment* cyclus in een hybride cloud omgeving doorlopen en welke tools kunnen daarvoor gebruikt worden?

De onderzoeksvraag bestaat uit een combinatie van onderzoek en uitvoering en zet een “*stap verder*” in het leren over hybride cloud omgevingen en *deployment* cyclussen.

Aanpak

De aanpak van de bachelorproef wordt verdeeld in 2 delen. Deel 1 gaat over het verzamelen van achtergrondinformatie en het onderzoeken ervan. Deel 2 is de uitwerking van een Proof of Concept.

Onderzoek:

- **Soorten Cloud:** Analyse naar de bestaande soorten cloud.
- **Azure VS Google Cloud:** Vergelijking van Azure en Google Cloud .
- **Provisioning:** Vergelijking van Terraform, Ansible, Vagrant.
- **Connectiviteit:** Connectiviteitsmogelijkheden analyseren en vergelijken.
- **Volledige Automatisering:** Onderzoek naar CI/CD uitvoering.
- **Jenkins:** Analyse van een CI/CD tool dat gebruikt wordt in de PoC.

Uitwerking:

- **Proof of Concept:** Uitwerking van een werkend concept op basis van het geleverd onderzoek.

Na elk deel van het onderzoek wordt ook kort advies gegeven dat gebaseerd is op de resultaten van de analyse en verdiepend werkt.

Soorten Cloud

Het werken met de verschillende types cloud is een uitdaging geweest. Dit deel beschrijft de verschillende soorten cloud instanties en geeft een beter inzicht aan de PoC. Na het onderzoek wordt advies gegeven over de toepassing van de verschillende soorten cloud in elk scenario.

Doorheen de jaren kreeg de cloud markt een enorme boost. Vele bedrijven en particulieren hebben de kracht van cloud omgevingen snel begrepen. Het idee om “*on the fly*” RAM-geheugen toe te voegen of enkele CPU's toevoegen is een zeer interessant concept met een grote waarde in de toekomst. Er bestaan 4 soorten cloud modellen die vaak gebruikt worden.

Public Cloud

Een ‘public cloud’ is een term voor diensten die aangeboden worden door een derde bedrijf over het internet. Deze diensten worden aangeboden aan de consument onder een ‘*pay-as-you-go*’ model die hoge kosten voor het gebruik van de cloud beperkt omdat de gebruiker enkel betaald voor de diensten die effectief gebruikt worden. Naast de particuliere gebruikers is een public cloud ook een oplossing voor vele bedrijven die willen besparen op kosten dat anders betaald worden voor het aankopen en onderhouden van fysieke apparatuur. Daarnaast is cloud ook veel sneller en makkelijker configureerbaar dan een private of on-premise cloud.²²

De enige nadelen van een public cloud zijn veiligheid en snelheid. Omdat bedrijven die klant zijn van public cloud providers niet de volle rechten hebben tot op hardware niveau ontstaat er makkelijk een SPOF (Single Point Of Failure) dat voor serieuze problemen kan zorgen moest de cloud provider slachtoffer zijn van een hacker aanval. Dat gebeurt niet vaak omdat de beveiliging van de public cloud voldoende uitgewerkt is, maar de mogelijk bestaat nog steeds.

De meeste aanbieders van een public cloud doen hun best om servers aan te bieden in alle hoeken en kanten van de wereld. Zo wordt iedereen voorzien van normale snelheden van dataoverdracht. Vaak is dat voor vele bedrijven en particulieren meer als genoeg. Het wordt pas problematisch als een groot aantal data tegelijk overgebracht moet worden voor een type verwerking of berekening.

²² (Microsoft Azure, z.d.)

Private Cloud

Dit is een intern netwerk dat enkel gebruikt kan worden door personen die daarvoor bevoegd zijn, zoals werknemers van een bedrijf. Vaak wordt een public cloud ook wel een “corporate cloud” genoemd aangezien deze vaak gebruikt wordt door grote bedrijven. Public cloud heeft verschillende voordelen naast de standaard functies dat ook in een public cloud voorkomen. Enkele daarvan zijn eigen beheer, elasticiteit, meer controle en eigen insteek. Dat zorgt ervoor dat bedrijven het niveau van diepgang op vlak van beveiliging, back-ups, snelheid van werking, kunnen kiezen. Daardoor is ook de IT-afdeling van een bedrijf verantwoordelijk voor het beheer: onderhoud, monitoring, kost en andere aspecten die verbonden zijn aan het beheer van een cloud infrastructuur.

In een private cloud zijn 2 types van service mogelijk, **IaaS**; maakt een bedrijf mogelijk om een eigen infrastructuur op te bouwen en modelleren. Daarnaast ook het beheren van het gebruik.

Een ander type is **PaaS**; dat geeft bedrijven de kans om zelf een provider te zijn van een platform voor hun klanten. Dat is vaak handig voor bedrijven die een soort software pakket aanbieden aan hun klanten. Zo kan de klant kiezen om deze op eigen server te laten draaien of in de (private) cloud van de softwareleverancier zelf.²³

Het grootste voordeel van dit soort cloud is beveiliging. Een bedrijf kan zelf kiezen hoe ver de beveiliging van de public cloud kan gaan. Daarnaast is een private cloud volledig afgeschermd van een public cloud. Zo kunnen geen gevoelige gegevens van bedrijven gestolen worden in geval van een lek in de datacenters van een public cloud. De nadelen van dit type cloud is vooral de kost en het onderhoud. De IT-afdeling moet uitgebreide kennis hebben van zowel cloud als systeem en netwerkbeheer om deze het best te kunnen beheren.²⁴

²³ (Microsoft Azure, z.d.)

²⁴ (Eswelt, 2019)

Hybrid Cloud

Een hybrid cloud is een verzameling van on-premise, public en private cloud instanties. Een bedrijf kan een hybrid cloud als uitgebreide IT infrastructuur gebruiken voor het beheren van gevoelige data in een eigen datacenter in de private cloud binnen de hybride omgeving. Door de vlotte samenwerking met cloud providers zoals Azure en GCP kunnen de resources van een private cloud snel uitgebreid worden waardoor de verwerking van een groter aantal gevoelige gegevens sneller kan verlopen. Het gebruik van een hybride omgeving biedt een pakket van voordelen aan;²⁵

- **Beveiliging en regulatie:** Gebruik van beveiligde private omgevingen voor gevoelige data. Voor het opslaan en verwerken van niet gevoelige data kan een public cloud gebruikt worden. Deze vermindert de kost een stuk en verwijderd de moeite van het opzetten van een eigen on premise omgeving.
- **Uitbreiding:** Cloud instanties uitbreiden in gevallen van verhoogd internetverkeer. De uitbreiding kan zowel permanent als tijdelijk toegepast worden waardoor de kost steeds lager blijft dan het opzetten van een fysieke on premise server.
- **Verouderde verwerkingen versnellen:** Verbeteren van prestatie en snelheid van verouderde applicaties door op maat resources aan te wijzen. Dit verbetert niet enkel de verwerkingstijd, maar ook de gebruikerservaring
- **Betere omgeving voor developers:** De integratie van DevOps maakt het makkelijker voor developers om te werken aan projecten in grotere teams op verschillende locaties.
- **Meer controle:** Er is meer visuele controle over de verschillende resources. Door het gebruik van dashboards kunnen problemen snel geïdentificeerd en opgelost worden.
- **Back-ups:** In geval van storing kunnen resources elkaar autonoom vervangen waardoor de downtime voor applicaties en services minimaal blijven.

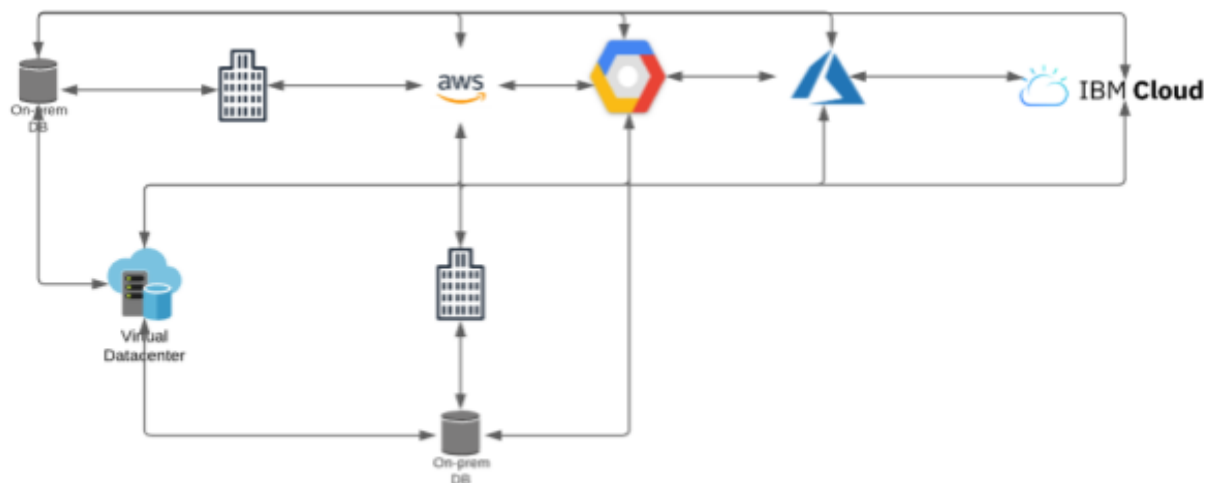
Bovenstaande aspecten dienen als voorbeeld van de vele voordelen waardoor een hybride omgeving superieur is aan het apart gebruik van alle technologieën. Vele bedrijven zijn begonnen met migratie naar een hybride omgeving omdat daar de toekomst van IT zich bevindt.

²⁵ (Vennam, 2021)

Hybrid Multi Cloud

Een multi cloud omgeving bevat cloud diensten van meerdere providers . De samenwerking van alle cloud providers maakt het makkelijker voor bedrijven om de juiste resources te vinden voor de beste prijs. Dit zowel op korte als lange termijn. IBM institute for Business toont aan dat een volledige hybrid multi cloud aanpak de omzet van bedrijven 2.5 keer hoger is dan een aanpak met een cloud vendor.²⁶

De onderstaande afbeelding geeft een visuele representatie van de werking van een multi cloud omgeving in combinatie met een private data center. Elke locatie van een bedrijf heeft een directe verbinding met een eigen on-premise data center of database dat gebruikt wordt voor het opslaan van gevoelige data. Deze zijn vervolgens gelinkt aan elkaar en een virtueel datacenter dat ook deel uitmaakt van de private cloud omgeving. Het virtueel datacenter is zoals de locaties van het bedrijf verbonden met alle cloud instanties van elke provider. Deze architectuur voorziet een makkelijke samenspraak tussen de resources van een bedrijf. (ook andere providers kunnen daarbij horen)



2 - Hybrid Multi Cloud Schema

Advies

Startende kleine bedrijven, maar ook grote uitgewerkte multinationals moeten zeker kijken naar de mogelijkheden van een hybrid (multi) cloud. Voor kleinere bedrijven, zelfstandige of particuliers is een simpele hybrid cloud omgeving een betere en goedkopere oplossing. Zo kunnen kosten voor het beheer van het intern netwerk exponentieel verminderd worden. Bij gebrek aan kennis kunnen IT bedrijven die gespecialiseerd zijn in cloud consultatie daarbij helpen.

Multinationals moeten in elk geval zeker kijken naar de voordelen van een hybrid multi cloud omgeving omdat de besparingen en tijdwinst die daarmee mogelijk zijn een groot voordeel zullen zijn voor het bedrijf, de werknemers en klanten. Hybrid cloud is de toekomst waar de huidige IT infrastructuur naartoe streeft. Door de verdere ontwikkeling van AI zal hybrid cloud een voordeel brengen voor professionele en particuliere doelen.

²⁶ (Comfort, Dolph, Robinson, Kesterson-Townes, & Marshall, 2020)

Azure VS Google Cloud Platform

In dit deel worden de verschillen tussen Azure en Google Cloud platform geanalyseerd. Vervolgens wordt advies gegeven over de toepassing en gebruik van deze cloud omgevingen.

Door de snelle evolutie van technologie en steeds grotere vereisten van uitgebreide applicaties kijken bedrijven steeds meer naar een cloud oplossing dat niet enkel vele kosten kan besparen maar ook voldoet aan andere aspecten die even belangrijk zoals schaalbaarheid en beveiliging. Vele cloud providers weten dat cloud de volgende grote stap in de technologische evolutie zou zijn. Daarom ontstaan steeds meer bedrijven in de cloud-computing markt met oplossingen voor ieders zin. Dit deel van het onderzoek gaat over Google Compute Cloud en Microsoft Azure, dit zijn de 2 cloud providers dat gebruikt werden tijdens het uitvoeren van de stage en verdiepend onderzoek. De 2 bedrijven hebben ook sterk geïnvesteerd in dit deel van de markt waardoor het analyseren van de korte, maar forse groei een interessant aspect kan zijn voor een beter begrip te kunnen vormen over de mogelijkheden tot evolutie, uitbreiding, aanpassing en concurrentie in de zeer competitieve en hoog technologische markt.

Azure

Microsoft Azure heeft een breed portfolio van diensten. De 4 belangrijkste zijn Back-up & DRP, Webhosting, Active Directory en IoT. Elk heeft een brede diepgang dat beschikbaar is, maar niet geconfigureerd moet worden om de basis diensten te kunnen gebruiken. Azure komt vaak voor in professionele omgevingen²⁷, dat komt het groot aantal diensten, waardoor bedrijven niet moeten zoeken naar ander providers. Alles is beschikbaar op 1 platform, dat maakt de leercurve ook kleiner. Azure voorziet ook een eenvoudige integratie met on-premise toepassingen zoals datacenters van bedrijven. Dit maakt de stap tot migratie naar een hybrid cloud vlotter.

Google

Google heeft een beperkt portfolio van diensten. Omdat het aantal diensten niet overbodig is kan de Cloud provider focussen op het leveren van kwalitatieve, volledige en uitgebreide functionaliteit in elke dienst zonder hoge kosten. Het platform is daardoor bedoeld voor individuele personen en kleine tot middelgrote bedrijven. Een beperkt gamma aan diensten, duidelijke begeleiding en documentatie schieten Google Cloud naar de top 3 cloud providers in 2021 volgens het "Magic Quadrant" van Gartner.²⁸

²⁷ (Acronis, z.d.; Acronis, z.d.)

²⁸ (Jones, 2020)

Virtualisering

De virtualisering op beide platformen wordt getoond in de *Machine-Type tabel*. De tabel is gebaseerd op de laatste cijfers van Google en Azure zelf. Voor de vergelijking werden de verschillende ‘pakketten’ die aangeboden worden door de providers verdeeld onder 6 types;²⁹

- Algemeen - Een basis platform voor kleine projecten.
- Geheugen - RAM-geheugen voor snelle orkestratie van processen.
- Rekenkracht - CPU en SSD voor snelle berekeningen.
- Opslag - M.2 SSD voor een snel en betrouwbaar opslagmedium.
- Grafisch - GPU voor het trainen van AI en ML, grafische verwerking.
- High PE. - TTL Hardware voor optimale virtualisering op alle vlakken.

Naast de 6 types heeft GCP nog een mogelijkheid voor het maken van een eigen configuratie waarbij de gebruiker zelf kan kiezen hoeveel hardwarecomponenten er nodig zijn voor het ondersteunen van een virtuele machine. De prijs van de configuratie wordt berekend o.b.v. de gekozen componenten.

Prijzen

De prijzen van Azure en Google zijn heel competitief. Door een analyse te maken op de bovenvermelde types en gelijkaardige aanbiedingen, kunnen 2 grafieken opgesteld worden die de prijs vergelijken. Beide grafieken tonen de kosten die verbonden zijn aan een bepaald pakket, de 1^{ste} grafiek wijst op de kost per uur, de 2^{de} grafiek op de kost per maand. Google heeft een lagere kost in vergelijking met Azure, dat komt door de verschillende kortingen die automatisch worden toegepast op het ‘pay as you go’ model. Eens een instantie gebruikt wordt voor een bepaalde tijd, worden die kortingen toegepast. Zonder de kortingen zou de prijs hoger zijn dan die van Azure³⁰. In conclusie is Google goedkoper op lange termijn.



3 - Prijsvergelijking per uur

4 - Prijsvergelijking per maand

²⁹ [Tabel virtualisering](#)

³⁰ (Jones, 2020)

Advies

Het is aangeraden om Azure te kiezen indien het gebruik van de resources bedoeld is voor een langere periode. In geval van een kort gebruik, bijvoorbeeld in het geval van een “*spike*” in het aantal gebruikers die tegelijk een applicatie gebruiken, kan Google ingezet worden met een snellere en goedkopere *deployment* in vergelijking met Azure.

Voor startende ondernemers is het zeker aangeraden om te beginnen met GCP, dat platform is namelijk bedoeld voor professionele starters en biedt dan ook hogere kortingen aan.

Voor grotere bedrijven en multinationals is het gebruik van Azure een voordeel omdat deze ook een grote prijs vermindering aanbiedt bij gebruikt van een hoog aantal resources met een vast contract van minimum 2 jaar.

Door de snelle evolutie van cloudinfrastructuur en grote concurrentie is het aangeraden om eerst uitgebreid onderzoek te doen over de aanbiedingen van alle providers.

Provisioning

Voor het provisionen van infrastructuur, netwerken en systemen zijn tools zoals Terraform, Ansible en Vagrant marktleiders. Zij bieden de meest uitgebreide oplossingen aan. In dit deel worden deze besproken. Vervolgens wordt advies gegeven over de toepassing van de tools in verschillende scenario's

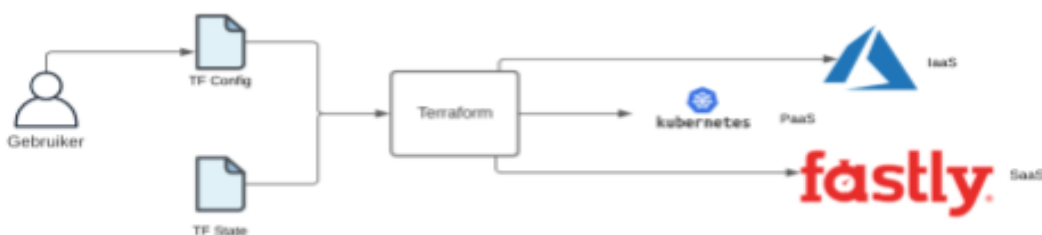
Terraform

Terraform is een *open-source* IaaS software tool waarmee een groot aantal cloud resources beheerd kunnen worden.³¹ Het nut en gebruik ervan werd gedurende de hele stageperiode en verdieping gebruikt voor het deployen van resources binnen de Azure omgeving.

Voor het maken van resources in de cloud worden **tf config** bestanden gebruikt. In deze bestanden wordt in HCL taal declaratief geschreven welke resources aangemaakt, veranderd of verwijderd moeten worden. Om de veranderingen uit te voeren zal Terraform eerst de staat van de cloud ophalen, zoeken naar bestaande resources, en vervolgens vergelijken welke aanpassingen er aangemaakt zijn. De volgende functies zijn funderingen van van Terraform;

- Init: Terraform starten
- Refresh: Tf config vergelijken met de cloud infrastructuur.
- Plan: Geschreven configuratie vergelijken met de bestaande configuratie.
- Apply: Plan toepassen in de cloud. Veranderingen uitvoeren.
- Destroy: Alle aangemaakte resources binnen een project volledig verwijderen.

Terraform is niet enkel een IaaS beheer, naast cloud en on-premise installaties kan Terraform ook PaaS en SaaS beheren. Individueel beheer van de 3 services is een zekere functie, maar wat Terraform zo uniek maakt is de mogelijkheid tot het beheer van alle 3 tegelijk.



5 - Terraform werking

In het bovenstaand schema wordt 1 gebruiker getoond die het config bestand beheert. Terraform voorziet ook een **enterprise** versie. Deze geeft meerdere gebruikers de kans om samen te werken aan een infrastructuur. Terraform is kortom de ideale tool als het komt op het beheren (provisioning) van cloud infrastructuur. Het is eenvoudig om te leren en biedt geen extra kosten. Verder in deze thesis worden ook andere tools besproken en de verschillen daartussen.

³¹ (HashiCorp, z.d.)

Ansible

Ansible is een handige tool voor het provisionen van systemen en het automatiseren van herhalende taken zoals configuratie, *deployment* en intern beheer van verkeer. De technologie gebruikt modules om de verschillende taken uit te voeren en maakt het mogelijk om alle taken die manueel uitgevoerd worden binnen een systeem te automatiseren. De taal die gebruikt wordt voor het schrijven van de scripts is YAML, een eenvoudig leesbare taal waarvoor weinig tot geen kennis nodig is om de verschillende commando's van elkaar te kunnen onderscheiden en begrijpen. Een voorbeeldscenario waarin Ansible perfect gebruikt kan worden is bij het deployen van meerdere diensten of databases tegelijk. De stappen voor *deployment* kunnen hergebruikt worden, de verschillen tussen de *deployments* kunnen met verschillende "rollen" aangetoond worden. Een *playbook* voert elke rol een voor een uit tot alle commando's succesvol afgelopen zijn.³²

Een sterk punt van ansible is **idempotentie**, in geval dat een script 10 stappen heeft en enkel stap 4 is verschillend van het huidig systeem, dan wordt enkel een aanpassing uitgevoerd op stap 4 en niet alle andere stappen in het script.^{33 34}

Vagrant

Vagrant is ook een product van Hashicorp en biedt naast terraform een bredere verdieping in het beheren van het netwerk verkeer, HTTP tunneling en synchronisatie van gedeelde mappen tussen verschillende virtuele machines. De tool biedt een oplossing voor 3 soorten jobs;³⁵

- **Developers:** Met 1 commando kan een hele development omgeving opgezet worden voor een developer. Zo is de werkomgeving van iedere werknemer binnen een bedrijf gelijk.
- **Operators:** Een omgeving voor het testen van verschillende scripts voor het beheer van de infrastructuur. Zo kunnen vele SSH-prompts vermeden worden.
- **Designers:** Opzetten van een hele omgeving waarbij alle applicaties geconfigureerd worden naar de juist instellingen, zodat geen tijd verloren wordt door niet werkende toepassingen.

Algemeen is Vagrant een handige tool voor het klaarmaken en configureren van meerdere toepassingen binnen een systeem. Daardoor kan de focus liggen op het werken en niet troubleshooten van issues.

Advies

Wat onthouden moet worden is dat Terraform dient voor het opzetten van infrastructuur, Ansible voor het configureren van de virtuele machines en Vagrant voor de installatie van toepassingen en werkomgevingen. De technologieën kunnen door elkaar gebruikt worden alleen is dat niet aangeraden omdat de code te ingewikkeld zal worden en niet maximaal zal kunnen presteren.

³² (StarAgile, 2020)

³³ (Ansible, z.d.)

³⁴ (Sebastian, 2020)

³⁵ (Hashicorp, z.d.)

Connectiviteit

In dit deel wordt beschreven waarom het concept van hybrid cloud vroeger niet mogelijk was. Vervolgens wordt advies gegeven over de toepassing van de connectiviteit.

Voor de verschillende 'remote offices' met elkaar te kunnen verbinden werden doorheen de jaren verschillende soorten connectie mogelijkheden mogelijk gemaakt: MPLS, broadband en LTE zijn daar enkele voorbeelden van. MPLS is een laag 2 communicatieprotocol dat vroeger gebruikt werd om verschillende bedrijfstakken met elkaar en het centrale datacenter te verbinden³⁶. De technologie was zeer handig wanneer het internetverkeer aan de lage kant stond, echter met de komst van cloud is het verkeer sterk toegenomen waardoor enkele problemen ontstonden bij het gebruik van MPLS;

- Hoge kosten
- Single Point Of Failure
- Beperkte bandbreedte
- Lange laadtijden



6 - MPLS connectiviteit

Door de verhoging van internetverkeer werd MPLS te duur voor het transport van grote hoeveelheden data. Er werd gepolst naar andere manieren voor het transport van grote hoeveelheden data op een snelle, autonome en goedkopere manier. SDN voor LAN netwerken en SD-WAN voor WAN netwerken zijn waardevolle technieken om deze problemen op te lossen zonder grote veranderingen aan het netwerk.

SDN

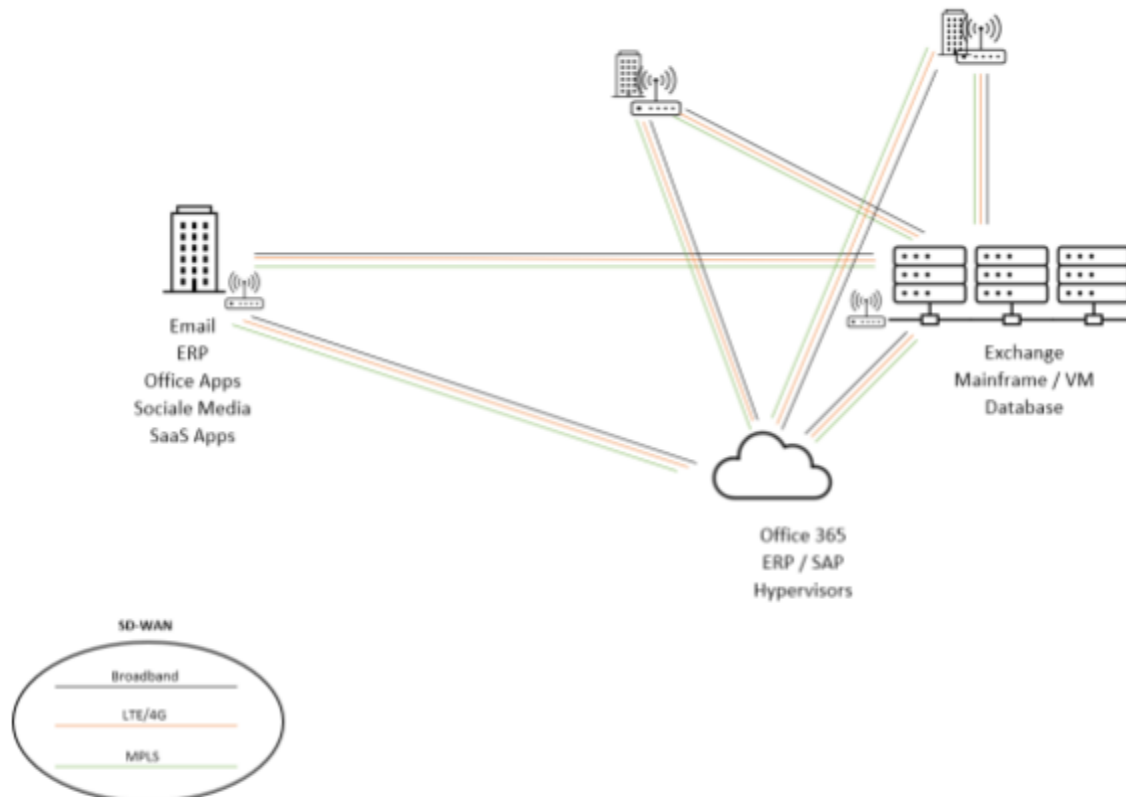
Software-Defined Networking is een van de meest gebruikte technologieën voor het beter beheren van een LAN omgeving, of een datacenter architectuur van een service provider. SDN splitst de *control plane* die verantwoordelijk is voor de apparaten binnen het netwerk en de *data plane* welke de data tussen de netwerkapparaten verstuurd. De technologie beheert routers en switches op een gecentraliseerde manier. Daardoor is het mogelijk om, met een correct werkende SDN structuur, de verschillende netwerk apparaten te configureren, parametriseren en automatiseren vanuit een 1 centrale plek in het netwerk.³⁷

³⁶ (Mushroom Networks, z.d.)

³⁷ (Cooney, 2019)

SD-WAN

Software-Defined Wide Area Networking is een manier om verschillende WAN links tussen branches en datacenters van grote bedrijven op een intelligente en proactieve manier te gebruiken. De technologie gebruikt Broadband, MPLS en 4G door elkaar en baseert zich op de noden van de applicatie. Een correct ingestelde SD-WAN routing kan vaak voorkomende taken elimineren, daarnaast biedt het ook vele besparingen en vermindert de complexiteit van het netwerk. SD-WAN is in 2021 nog steeds een relevante oplossing om WAN verbindingen te beheersen op een flexibele en autonome wijze.³⁸



7 - SD-WAN Netwerken

Advies

Door SD-WAN is de mogelijkheid tot het toepassen van hybrid cloud sterk gestegen. De lage kosten en mogelijkheid tot het gebruik van meerdere kanalen voor data transfer maken ook mogelijk om hybride multi cloud oplossingen toe te passen. Het is sterk aangeraden om te onderzoeken welke transportmogelijkheden bedrijven met een focus in SD-WAN aanbieden.

³⁸ (Banks, 2014)

Volledige Automatisering

In dit deel van de thesis wordt CI/CD kort besproken. Dit is relevant voor de uitwerking van het POC. Vervolgens wordt advies gegeven over de toepassing.

Automatisering speelt een steeds grotere rol in de wereld van IT, meer bepaald DevOps. Door het maken van een volledige automatisering met gebruik van continuous integration, delivery en *deployment* wordt het automatisch updaten van code, websites of applicaties een stuk makkelijker.

Continuous Integration

Continuous integration voorziet dat de code waar developers binnen het bedrijf aan werken gelijk blijft door deze eerst te testen op bugs, versie, veiligheid, kwaliteit, ...³⁹ Het is belangrijk dat alle testen gebeuren voor de code gepusht wordt naar de productieomgeving waar deze toegepast wordt. Na het pushen kunnen developers de aanpassingen ophalen van het centraal platform.

Continuous Delivery

Deze stap volgt nauwkeurig op de vorige en voorziet een snelle en foutloze levering van code in de productieomgeving. Zo kan deze toegepast worden in *deployment* of kan hier verder op gebouwd worden door de developers.⁴⁰

Continuous Deployment

Na de voorbije controles kan de software of code toegepast worden in een omgeving. Een voorbeeld is het uitvoeren van een Terraform script in de cloud. Nadat het script getest is en correct werkt, wordt eerst geanalyseerd welke code binnen het script veranderd is. Deze waarden worden gepusht naar de cloud en aangepast. Meer hierover in het deel Proof of Concept.

Advies

CI/CD is een zeer handige tool voor het automatisch updaten van een infrastructuur, software of applicaties. Zowel voor professionals als particulieren is het gebruik van deze technologie een stap vooruit. Door de concurrentie van verschillende CI/CD software zijn vele toepassingen gratis te gebruiken voor het ontwerpen van simpele pipelines. Azure pipelines, Jenkins, Gitlab, Kubernetes zijn maar enkele (populaire) voorbeelden.

Op het internet kan veel documentatie en uitleg gevonden worden over de bovenstaande tools. Verder in deze thesis staat ook uitgelegd hoe CI/CD ontwikkeld kan worden binnen Jenkins.

³⁹ (Mansystems, z.d.)

⁴⁰ (Drewel, 2020)

Jenkins

Dit deel gaat over een CI/CD platform dat gebruikt werd voor de uitwerking van een POC. Vervolgens wordt advies gegeven over de toepassing.

Jenkins is een standaard tool die gebruikt wordt voor continuous integration, deployment en delivery. Doorheen de jaren is Jenkins een zeer stabiel platform geworden dat vaak “go-to” is in het veld van DevOps. Door een makkelijke installatie en configuratie voorziet Jenkins een eenvoudige mogelijkheid voor te starten met CI/CD.

Installatie

De Jenkins installatie is mogelijk op verschillende soorten systemen; Docker, Linux, Windows, MacOS, Tijdens de installatie is het belangrijk om te letten in welke map alle gegevens worden bewaard. Bij het gebruik van “containers” moet de gebruiker er ook op letten dat alle gegevens bewaard worden na het afsluiten van de Jenkins toepassing.⁴¹

Configuratie

De configuratie van een Jenkins platform is een stuk eenvoudiger door de verschillende plug-ins die gedownload kunnen worden. Zo kunnen de login gegevens voor verschillende websites en cloud instanties bewaard worden binnen de Jenkins applicatie. Daardoor kunnen de verschillende pipelines sneller en veiliger uitgevoerd worden.

Werking

De pipeline van Jenkins werkt met een **Groovy** script. Binnen het script moet vermeld worden welke tool Jenkins moet gebruiken voor de installatie. Daarnaast is het belangrijk om te vermelden welke credentials die tijdens de configuratie bewaard werden gebruikt moeten worden voor de login te kunnen voltooiën. Tot slot worden stappen aangemaakt, in elke stap wordt een functie beschreven die normaal manueel uitgevoerd wordt. Het is zeker belangrijk om na elke stap een controle uit te voeren indien dat nodig is, vooraleer de pipeline verder kan met de installatie.

Door een uitgewerkt en duidelijk interface is het makkelijk om de vooruitgang van een pipeline te bekijken. Ook kunnen details zoals duurtijd van elke stap of de console output bekeken worden.

Advies

Jenkins is een van de CI/CD tools dat pipelines kan aanmaken. Het is zeker niet verplicht om zich te focussen op de uitwerking binnen Jenkins. Probeer andere CI/CD tools en analyseer zelf de voor- en nadelen van elke tool. Door zich niet te focussen op een tool wordt de kennis van CI/CD en andere tools uitgebreid. Zo ontstaat een bredere visie van het marktaanbod en kan beter bepaald worden welke tool gebruikt kan worden voor een specifiek type deployment.

⁴¹ (Jenkins, z.d.)

Proof of Concept

In dit deel wordt kort besproken hoe de proof of concept voor een CI/CD deployment werkt en toegepast kan worden in een hybrid cloud omgeving.

Opzet

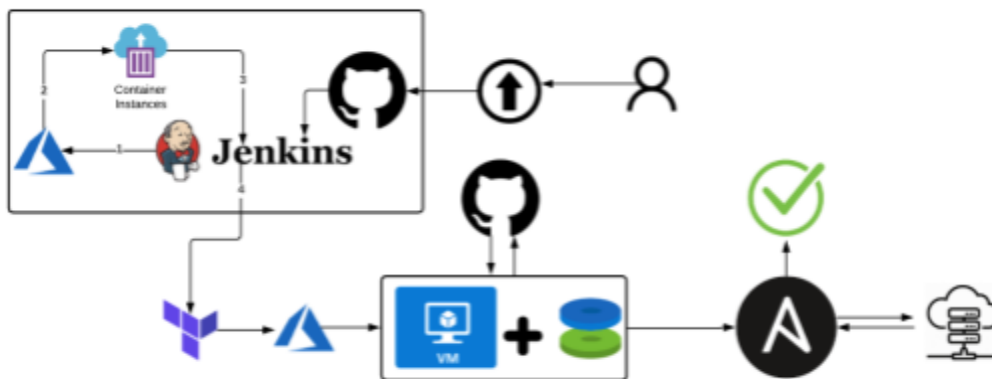
De uitwerking van het POC is bedoeld als een verdieping op het bestaand stageproject. De bedoeling van deze uitwerking is het leren toepassen van een CI/CD tool en workflow door zelf onderzoek te doen naar de nodige technologieën en de configuraties daarvan.

Proces

Het proces begint met een Github push van de developer. Na het pushen van het document stuurt Github een ping naar Jenkins die vervolgens de veranderingen in de code analyseert en controleert. Vervolgens wordt de verbinding met azure vastgesteld. Het script wordt in een vooraf aangemaakte Azure container geplaatst. Jenkins voert een laatste controle uit en start de terraform initialisering.

Het script wordt door Terraform doorlopen, er wordt gekeken naar verschillen tussen de nieuwe code en de situatie van de resource binnen de cloud. Indien de code correct is opgesteld, vraagt Terraform bevestiging voor het uitvoeren van een **apply**. Ten slotte worden aanpassingen verwerkt in de resources van de cloud. Hier is geen Jenkins tussenkomst nodig, omdat Terraform zelf controle doet, een extra controle is redundant en overbodig.

Voor een beter inzicht te krijgen van de uitwerking kan de Jenkins CI/CD documentatie in bijlage daarbij helpen^{42,43}. Ook is de code beschikbaar gesteld op Github⁴⁴ of in bijlage⁴⁵



8 - CI/CD Hana Express Azure Deployment

⁴² [Jenkins CI/CD Documentatie](#)

⁴³ [Ubuntu Docker Jenkins configuratie](#)

⁴⁴ [Github Repository](#)

⁴⁵ [Repository Zip](#)

Reflectie

Door de stage en bachelorproef heb ik geleerd om volledig zelfsturend te werken en processen in eigen handen te nemen zonder te wachten op een mogelijke oplossing die door iemand anders aangeboden wordt. Het project heeft mij een duidelijk inzicht gegeven op de wereld na de studies en hoe ik daar in kan presteren zonder mezelf te overwerken. Doorheen de maanden kreeg ik ook terechte tips van de stagebegeleiders en collega's over hoe mijn mentaliteit op de werkvloer later zou moeten zijn.

Naast het zelfstandig aanpakken van de uitwerking, heb ik ook geleerd om probleemgericht te denken waardoor ik het probleem centraal stel en meerdere manieren van aanpak probeer tot er een vorm van duidelijkheid is die helpt bij de probleemoplossing.

Ik heb alle doelen die bereikt moeten na het einde van de stage en de bachelorproef met volle zekerheid bereikt en soms zelf overtroffen. Daardoor heb ik ook een hoger niveau van kennis en vaardigheden bereikt.

Conclusie

De uitgewerkte tool voor de automatische deployment van een klassiek ERP-systeem zoals SAP heeft een opvallend vlotte en foutloze werking. Dankzij transparante en beheerbare deployments kan de installatie makkelijk gemonitored worden. Ook is het mogelijk om deze volledig autonoom te laten verlopen zonder enige tussenkomst van de mens.

De uitwerking zal ook gebruikt worden door het BC-team dat verantwoordelijk is voor installaties, migraties en algemeen support van SAP-systemen in het dagdagelijks leven. Daardoor biedt de tool een toegevoegde waarde aan het stagebedrijf.

De verdieping is gelukt door het overnemen van de stageopdracht en deze verder uit te werken naar een volwaardig continuous integration en deployment pipeline. Zo kan de Infrastructure as Code structuur eerst meermaals gecontroleerd en uiteindelijk volledig uitgerold worden in de productieomgeving. Het nut van deze uitwerking verlost de administrators volledig van herhalende en langdradige processen die vroeger een *'bottle-neck'* vormden in de job doelen.

Dit document biedt een overzicht van alle toepassingen en adviezen en verwijst tenslotte ook naar de bijlagen waar in breder detail alle uitwerking en notities beschreven staan.

Het project en verdieping bewijst nogmaals dat automatisering van herhalende processen steeds meer toegepast worden in verschillende branches van IT. Het gebruik van continuous integration en deployment, pipelines, infrasturucure as code en tools voor automatische configuraties van software in hybride cloud omgevingen, vormen een stevig fundament voor automatisering.

Bronnen

Ansible. (z.d.). *Glossary*. Opgehaald van docs.ansible.com:

https://docs.ansible.com/ansible/latest/reference_appendices/glossary.html

Cheops. (2020). *Wat is een disaster recovery plan (DRP)?* Opgehaald van Cheops:

<https://www.cheops.com/glossary/wat-is-een-disaster-recovery-plan-drp/>

Comfort, J., Dolph, B., Robinson, S., Kesterson-Townes, L., & Marshall, A. (2020). *The hybrid*. IBM Institute for Business Value.

Computerwoorden.nl. (2020). *Betekenis Fail-over*. Opgehaald van Betekenisdefinitie:

<https://www.betekenis-definitie.nl/Fail-over>

Drewel, K. (2020, Juli 20). *Continuous Integration, Continuous Delivery en Continuous Deployment (CI/CD) uitgelegd*. Opgehaald van true.nl: <https://www.true.nl/blog/ci-cd-uitgelegd/>

Eswelt. (2019, Mei 01). *Public vs. Private Cloud, wat moet je als bedrijf kiezen?* Opgehaald van Eswelt.nl: <https://www.eswelt.nl/blog/2019/05/public-vs-private-cloud>

Flexso. (2020). *Over Ons*. Opgehaald van Flexso: <https://www.flexso.be/nl/over-flexso/>

Goossens, G. (2020, Augustus 28). *Flexso trains more than 300 job seekers to start a career as an SAP Consultant*. Opgehaald van Flexso:

<https://www.flexso.be/en/blog/start-your-career-as-an-sap-consultant/>

Hashicorp. (z.d.). *Introduction to Vagrant*. Opgehaald van vagrantup.com:

<https://www.vagrantup.com/intro>

Hashicorp. (z.d.). *Vagrant vs. Terraform*. Opgehaald van vagrantup.com:

<https://www.vagrantup.com/intro/vs/terraform>

Jenkins. (z.d.). *User Handbook*. Opgehaald van jenkins.io: <https://www.jenkins.io/doc/book/installing/>

Mansystems. (z.d.). *Continuous Integration / Continuous Delivery*. Opgehaald van mansystems.com:

https://www.mansystems.com/smart-digital-factory/tooling/ci-cd?gclid=Cj0KCQjw--GFBhDeARIsACH_kdb8DMYVdFd2T0n8MUXFbKe6oTGJTvFGH0QovFkrA-XhYFC3tWwnUX8aAgJIEALw_wcB

Microsoft Azure. (z.d.). *What a public cloud?* Opgehaald van azure.microsoft.com:

<https://azure.microsoft.com/en-us/overview/what-is-a-public-cloud/>

Microsoft Azure. (z.d.). *What is a private cloud?* Opgehaald van azure.microsoft.com:

<https://azure.microsoft.com/en-us/overview/what-is-a-private-cloud/>

Sebastian. (2020, Oktober 12). *Ansible: Idempotent Playbooks*. Opgehaald van dev.to:
<https://dev.to/admantium/ansible-idempotent-playbooks-4e67>

StarAgile. (2020, Juli 28). *What Is Ansible?: Uses, Benefits, Architecture*. Opgehaald van StarAgile:
<https://staragile.com/blog/what-is-ansible-in-devops>

Vennam, S. (2021, Mei 12). *Hybrid cloud*. Opgehaald van ibm.com:
<https://www.ibm.com/cloud/learn/hybrid-cloud>