Faculteit Bedrijf en Organisatie

Tools voor het opvolgen van systeem- en applicatielogs, voor containervirtualisatie: vergelijkende studie en proof of concept

Owen Van Damme

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Bert Van Vreckem
Co-promotor:
Bert Van Vreckem

Instelling: HOGENT

Academiejaar: 2020-2021

Tweede examenperiode

Faculteit Bedrijf en Organisatie

Tools voor het opvolgen van systeem- en applicatielogs, voor containervirtualisatie: vergelijkende studie en proof of concept

Owen Van Damme

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Bert Van Vreckem
Co-promotor:
Bert Van Vreckem

Instelling: HOGENT

Academiejaar: 2020-2021

Tweede examenperiode

Woord vooraf

Het voeren van een onderzoek en deze beschrijven in een bachelorproef is één van de laatste stappen die genomen moet worden wanneer een student toegepaste informatica wilt afstuderen. Ikzelf heb gekozen voor de afstudeerrichting netwerk- en systeembeheer omdat ik niet enkel geïnteresseerd ben in het maken van software maar ook een grote interesse heb voor hardware en de infrastructuur rondom computers en servers. Containervirtualisatie ligt binnen mijn interesse en toen ik zag dat omtrent containervirtualisatie een onderwerp beschikbaar was moest ik die gewoon onderzoeken.

Aangezien ik er tijdens dit onderzoek niet alleen voor stond zou ik hier toch enkele mensen willen bedanken.

Eerst zou ik Bert Van Vreckem willen bedanken die de rol van promotor en co-promotor op zich heeft genomen. Ik zou hem willen bedanken voor het onderwerp en mij de kans te geven dit onderwerp te onderzoeken. Ook wil ik hem bedanken voor alle feedback, alle antwoorden op mijn vragen en de algemene hulp die hij mij gegeven heeft.

Uiteraard wil ik mijn ouders bedanken om mij de steun te geven, zowel financieel als mentaal, om het mogelijk te maken voor mij om te studeren.

Ook wil ik mijn zus Zohra Van Damme bedanken, niet enkel voor alle mentale steun en afleiding tijdens mijn studies maar ook voor het verbeteren van de taalfouten in deze bachelorproef.

Als laatste wil ik mijn vriendin en al mijn vrienden bedanken. Sommige stonden mij bij voor technisch advies, anderen hielpen mij af en toe ontspannen en sommige deden beide.

Zonder de hulp van al deze personen was de bachelorproef nooit geweest wat ze is.

Samenvatting

Informatica is een wetenschap die altijd evolueert. Technologieën en programmeertalen die een jaar geleden nog de standaard waren kunnen nu volledig verouderd en achterhaald zijn. Zo goed als elk jaar komt er nieuwe en verbeterde hardware op de markt voor computers en servers. De infrastructuur rondom computers en servers blijft ook maar evolueren en verbeteren. De informatica, de achterliggende technologieën, de manier van werken binnen de computerwetenschappen en de filosofie erachter veranderen continu.

Het continu veranderende landschap van de computerwetenschappen is iets positief aangezien deze altijd verbeterd wordt. Het is echter een uitdaging voor universiteiten en hogescholen om uit dit veranderend landschap de belangrijke zaken te halen, deze te verwerken in cursussen en zo een student voor te bereiden op de bedrijfswereld waarin deze gaat terechtkomen. Daarom is het belangrijk voor een universiteit of hogeschool het eigen curriculum te herzien en bij te werken.

De bacheloropleiding toegepaste informatica van HOGENT gaat momenteel door veranderingen en één van die veranderingen is het curriculum zelf. Er wordt overwogen om containervirtualisatie en bijhorende best practices op te nemen in het curriculum. Één van die best practices is log management.

Er is dus een vraag gekomen vanuit HOGENT, en specifieker door lector Bert Van Vreckem om te onderzoeken welke log management tool, die logs kan verzamelen en verwerken uit containers, er geschikt is om op te nemen in het curriculum toegepaste informatica.

In dit onderzoek is eerst onderzocht welke log management tools er bestaan. Dan is aan de hand van een MoSCoW-analyse gevonden dat er vier log management tools geschikt zijn om op te nemen in het curriculum. Deze vier tools zijn de ELK-stack, de EFK-stack, Grafana Loki en Graylog. Deze vier tools zijn dan iets verder in detail bekeken en met

elkaar vergeleken. Uit de vergelijking is besloten om te kiezen voor de EFK-stack.

Van de EFK-stack zijn er verschillende proof of concept versies gemaakt. De ene versie tracht altijd een verbetering te zijn op de vorige versie. Uiteindelijk voldeed versie drie van de proof of concept aan alle requirements waardoor deze de laatste versie is.

Inhoudsopgave

Lijst van tabellen

1. Inleiding

De bacheloropleiding toegepaste informatica van HOGENT is momenteel door een heleboel veranderingen aan het gaan. Er verdwijnen enkele onderwerpen uit het curriculum en er worden nieuwe onderwerpen toegevoegd aan het curriculum. Één van de onderwerpen die overwogen wordt om op te nemen, is de log management van virtuele containers.

Tijdens dit onderzoek zullen eerst alle onderwerpen bekeken en beschreven worden die nodig zijn om te verstaan wat virtuele containers, logs en log management tools zijn. Er zal dan een opsomming en korte beschrijven volgen van alle log management tools.

Dan worden alle log management tools vergeleken aan de hand van requirements, om zo te onderzoeken welke tool het best geschikt is als leerstof voor de studenten toegepaste informatica. Deze vergelijking van de log management tools is nodig aangezien er niet mag vanuit gegaan worden dat de populairste tool ook de best geschikte tool is. Uiteindelijk zal er een proof of concept uitgewerkt worden. Deze moet bewijzen dat de log management tool kan opgezet worden en deze kan ook dienen als guide.

1.1 Probleemstelling

Wanneer er veranderingen in het curriculum toegepaste informatica worden overwogen, moeten verschillende zaken onderzocht worden. Daarom is er vanuit HOGENT, en specifieker door lector Bert Van Vreckem, de vraag gekomen om te onderzoeken welke log management tool het best geschikt is om eventueel op te nemen in het nieuwe curriculum.

Dit onderzoek zal gericht zijn op lectoren binnen de bacheloropleiding toegepaste informatica. Het uiteindelijk doel van het onderzoek is om de lectoren een beter beeld te geven

over alle log management tools die gebruikt kunnen worden voor het loggen van virtuele containers. Daarna volgt een vergelijking die duidelijk moet maken waarom er voor de proof of concept een specifieke tool is gekozen om verder uit te werken. Uiteindelijk volgt er de proof of concept, deze kan dienen als startpunt voor de lector of lectoren die een cursus opstellen over log management.

1.2 Onderzoeksvraag

Er gaat dus veel gepaard met het veranderen of het toevoegen van leerstof aan een curriculum en niet alles kan in één onderzoek bestudeerd en omschreven worden. Ook zijn er vragen of problemen die enkel opgelost kunnen worden na discussies tussen enkele of alle belanghebbende partijen, deze kunnen dus niet enkel opgelost worden door een onderzoek. Daarom zal dit onderzoek zich vooral focussen op het zoeken van het meest geschikte programma of applicatie, die log files uit een virtuele container verzameld, verwerkt en visualiseert rekening houdend met de unieke vereisten die zich voordoen. Er is dan een kans dat dit programma zou worden opgenomen in het curriculum toegepaste informatica.

Uiteindelijk is de onderzoeksvraag “Welke log management tool, die logs kan verzamelen en beheren uit virtuele containers, is geschrikt om aan te leren aan studenten in de bacheloropleiding toegepaste informatica?”. Na het uitzoeken van de meest geschikte log management tool zal deze uitgewerkt worden in een proof of concept.

1.3 Onderzoeksdoelstelling

Het uiteindelijke doel van deze bachelorproef is tweedelig. Enerzijds zal er een vergelijkende studie gemaakt worden tussen alle log management tools. De log management tools zullen met elkaar vergeleken worden aan de hand van requirements. Deze vergelijking zal geslaagd zijn wanneer er één of meerdere log management tools uit komen die geschikt zouden zijn om op te nemen in de bacheloropleiding toegepaste informatica.

Anderzijds moet er proof of concept gemaakt worden met de tool verkregen uit de vergelijkende studie. Deze proof of concept zal pas geslaagd zijn als de opstelling toelaat om logs te verzamelen uit verschillende containers, deze de logs op een centrale server, virtuele machine of virtuele container kan opslaan en er een complexe filtering op de logs kan uitgevoerd worden. Er moeten ook een soort van dashboards gemaakt kunnen worden. Dit alles moet natuurlijk kunnen gedaan worden door studenten op een laptop met dezelfde specificaties als de computervereisten opgelegd door HOGENT. De proof of concept zal ook moeten kunnen dienen als startpunt voor een eventuele cursus die over dit onderwerp gemaakt wordt. Het moet mogelijk zijn om aan de hand van de proof of concept een log management tool te kunnen installeren en configureren zonder gebruik te maken van externe bronnen.

1.4 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt alle informatie, nodig om te begrijpen wat virtuele containers en log management tools zijn, beschreven. Ook staat in dit hoofdstuk een opsomming van alle log management tools geschikt voor de log management van virtuele containers.

In Hoofdstuk 3 wordt beschreven welke fases het onderzoek zal doorlopen. De requirements staan in dit hoofdstuk voor het eerst beschreven.

In Hoofdstuk 4 worden de requirements nader verklaard. Ook worden in dit hoofdstuk alle log management tools met elkaar vergeleken en weergegeven in tabellen. Deze tabellen worden verder verklaard en beschreven.

In Hoofdstuk 5 wordt een proof of concept gemaakt van de log management tool gekozen in hoofdstuk 4. In de proof of concept worden de einddoelen toegelicht. Niet enkel de installatie en configuratie van de log management tool wordt beschreven, maar ook alle software nodig om de log management tool te installeren.

In Hoofdstuk 6 wordt voor een laatste keer een overzicht gegeven van de, in voorgaande hoofdstukken gevonden, resultaten.

2. Stand van zaken

2.1 Verschillende soorten virtualisatie

Vooraleer een studie kan gemaakt worden van welke log management tool het best gepast is voor gebruik in een containeromgevingen, moet eerst achterhaald worden wat containervirtualisatie juist is. Computers en servers bestaan uit hardware (denk aan processor, RAM, hard drives,...) en om deze allemaal in harmonie samen te laten werken is een besturingssysteem nodig (Tanenbaum & Bos, 2015). Een besturingssysteem bestaat uit meerdere onderdelen en één van die onderdelen is de kernel. De kernel voorziet het laagste niveau van controle over de hardware.

Specifiek voor servers zou het gunstig zijn om alle mogelijke functies die deze moet vervullen en de applicaties die nodig zijn om de server in staat te stellen deze functies te vervullen, zo veel als mogelijk te scheiden van elkaar. Dit kan bereikt worden door de verschillende functies van de server op fysiek verschillende servers te plaatsen. Hier is het nadeel dat er meer hardware moet gekocht worden voor de servers, wat kan zorgen voor een hogere kost en deze oplossing vergt ook meer fysieke ruimte. Dit is voor veel situaties niet de meest geschikte oplossing.

Een andere oplossing voor het voorgaande probleem zijn virtuele machines(Sloman & Chrisley, 2003). Waar voordien een server bestond uit hardware met daarop een besturingssysteem (ook wel het host besturingssysteem genoemd) zal er nu software aan toegevoegd worden. Die toegevoegde software wordt een hypervisor genoemd. Deze hypervisor zal zich dan gedragen als virtuele hardware waarop een volledig nieuw besturingssysteem kan geïnstalleerd worden met al zijn componenten, dit besturingssysteem is het guest besturingssysteem. De hypervisor kan zich meerdere keren voordoen als virtuele hardware waardoor er ook meerdere guest besturingssystemen kunnen geïnstalleerd worden. Deze

guest besturingssystemen gedragen zich als een normaal besturingssysteem, hierop kunnen ook programma's en applicaties geïnstalleerd worden en zo een guest besturingssysteem samen met alle geïnstalleerde programma's worden virtuele machines genoemd. Een virtuele machine laat toe alle gewenste functies van servers gescheiden van elkaar te laten draaien op dezelfde fysieke server. Dit heeft als voordeel dat er minder moet uitgegeven worden aan hardware, dat onderhoud van de verschillende servers makkelijker is. Het verhoogt ook de performantie, het falen van de ene functie heeft geen invloed op de werking van de andere functie aangezien deze virtueel op andere hardware staan en onderling van elkaar gescheiden zijn. Het nadeel van een virtuele machine is dat er steeds opnieuw een volledig besturingssysteem moet geïnstalleerd worden. Dit is een proces dat relatief lang kan duren, ook zijn virtuele machines minder efficiënt aangezien deze draaien op de virtuele hardware.

Om voorgenoemde problemen op te lossen zijn virtuele containers ontwikkeld (Chamberlain, 2018). Een virtuele container is een stuk software die alles bevat (libraries, configuration files, executables en de binary code) om een applicatie succesvol uit te voeren net zoals een virtuele machine. Waar een virtuele container verschilt met een virtuele machine is dat een virtuele container geen besturingssysteem bevat, het gebruikt de kernel die reeds aanwezig is op de server en past enkel de bestanden aan die moeten aangepast worden. Virtuele containers zullen dan net zoals virtuele machines alle verschillende functies van de server van elkaar scheiden. Het succes of falen van de ene applicatie heeft geen invloed op het succes of falen van een andere applicatie.

2.2 Verschillen tussen virtuele containers en virtuele machines

Zowel virtuele containers als virtuele machines bereiken dezelfde doelen. Beide verhogen ze de performantie van een server door alle nodige applicaties onderling van elkaar te scheiden. Ook vergemakkelijken ze het configureren en het onderhoud van een server, wanneer iets faalt is het makkelijker om te zien welk applicatie er precies faalt en te achterhalen wat er juist misgaat. Wanneer de server moet worden herstart voor een specifieke applicatie, om welke reden dan ook, kan enkel die virtuele container of virtuele machine herstart worden. Dit heeft als voordeel dat de andere functies van de server gewoon kunnen blijven draaien zonder verstoord te worden.

Ondanks virtuele containers en virtuele machines dezelfde doelen realiseren zijn er verschillen tussen beide. Een virtuele machine zal een volledig nieuw besturingssysteem (guest besturingssysteem) installeren bovenop het besturingssysteem van de server (host besturingssysteem) terwijl een virtuele container het besturingssysteem zal delen met de server. Dit zal ervoor zorgen dat een nieuwe virtuele container beduidend sneller gemaakt is dan een virtuele machine, met als gevolg dat een virtuele container aanzienlijk sneller operationeel zal zijn dan een virtuele machine. Een ander voordeel van het delen van het besturingssysteem van virtuele containers is dat een virtuele container minder plaats zal innemen op de harde schijf van een server dan virtuele machines, er zullen meer virtuele containers op een server kunnen draaien dan virtuele machines (dit is natuurlijk ook afhankelijk van de functies en applicaties die de virtuele containers en virtuele machines juist uitvoeren). Een tweede voordeel van het kleinere formaat van virtuele containers is

dat deze eenvoudiger en sneller gekopieerd en gedeeld kunnen worden. Als één van de functies moet verplaatst worden van de ene naar de andere server, of een bepaalde functie werd eerst in een testomgeving gemaakt en moet worden geïmplementeerd in een productie omgeving, zal dit allemaal veel sneller kunnen verlopen met virtuele containers.

Uit studies blijkt dat er nagenoeg geen verschil is in uitvoeringstijd van een programma die rechtstreeks op een besturingssysteem draait of een programma die in een virtuele container draait (Beaurain, 2019).

Een groot nadeel dat een virtuele container heeft ten opzichte van een virtuele machine is dat een virtuele container enkel hetzelfde type van operating systeem kan hebben als dat het host operating systeem is. Wanneer er op een server Linux staat kan een virtuele container enkel maar een Linux besturingssysteem hebben (voor Linux kunnen de distributies ervan verschillen tussen de host en de container), er zal nooit een Windows container op een server kunnen gezet worden waar Linux op draait. Het guest besturingssysteem bij een virtuele machine is onafhankelijk van het host besturingssysteem.

2.3 Docker

Virtuele containers zijn een uiterst handige tool. Voor een virtuele container te deployen in een omgeving of opstelling is daar specifieke software voor nodig. Één van die programma's die gebruikt wordt voor het deployen, maken en aanpassen van virtuele containers is Docker.

Docker is één van de populairste programma's om te werken met virtuele containers en wordt na studies, zoals die van softwaretestinghelp (2021) en Lashawn (2021), bevonden als de beste virtuele container software optie.

Een Docker container start op aan de hand van een Dockerfile. Een Dockerfile bevat alle informatie nodig voor het maken van een Docker image (Yegulalp, 2019). Een Dockerfile is geschreven in een eenvoudige syntax. Het zal alle stappen doorlopen om bepaalde applicaties of software te installeren in een Docker container. Het kan zijn dat reeds iemand anders een Docker container heeft gemaakt met daarop applicaties geïnstalleerd. Deze persoon kan dan simpelweg de bekomen Docker image opslaan en online plaatsen.

Een Docker image kan beschreven worden als het recept voor een Docker container. Een image heeft reeds alle gewenste applicaties geïnstalleerd en geconfigureerd (Yegulalp, 2019). Zo kan het bijvoorbeeld dat iemand een mariadb Docker container wilt opstarten. Die persoon kan dan aan de hand van een Dockerfile de mariadb applicatie installeren in een container. Aangezien mariadb een populaire applicatie is kan het dat reeds iemand die Dockerfile gemaakt heeft en de Docker image online heeft geplaatst. Dan kan gewoon de Docker image gebruikt worden om de mariadb Docker container op te zetten.

Al deze Docker images zijn te vinden op dockerhub (Yegulalp, 2019). Dockerhub is een verzameling van Docker images.

Ook is Docker verkozen als nummer één “Most Loved” en nummer twee “Most wanted” platform in een bevraging van StackOverflow (2019).

2.4 Docker-Compose

Wanneer er meerder Docker containers gestart moeten worden kan dit gedaan worden met Docker Compose (Compose, g.d.). Docker Compose is een tool die aan de hand van een “docker-compose.yml” bestand gelijktijdig meerdere Docker containers kan starten met slechts één commando. Het YAML bestand bevat alles nodig om verschillende Docker containers gelijktijdig op te starten.

2.5 Een definitie van logs

Een log is vaak één maar regelmatig ook meerdere computer gegenereerde lijnen tekst die informatie bijhouden over gebeurtenissen en berichten die plaatsvinden in een besturingssysteem, een specifieke software, servers of andere toestellen (sumologic, g.d.). Zo kan er een onderscheid gemaakt worden in event logs en message logs.

Message logs zijn logs die berichten tussen verschillende gebruikers bijhoudt en opslaat (Wikipedia, g.d.). Dit type logs is echter minder interessant voor de studenten toegepaste informatica aangezien de communicatie tussen eindgebruikers geen beeld geeft op de gebeurtenissen die plaatsvinden in software of op een server.

Event logs houden gebeurtenissen bij die op een server plaatsvinden. Het kan bijvoorbeeld bijhouden welke gebruikers allemaal ingelogd hebben op een server, welke applicaties er gestart of gestopt zijn, of een persoon aanpassingen heeft gemaakt op een server en zoveel meer (sumologic, g.d.). Deze logs kunnen echter ook de status van een applicatie of service bijhouden. Het kan bijhouden wanneer een applicatie of service succesvol verloopt, of juist faalt. Logs kunnen ook meer duiding geven over waarom een bepaalde applicatie faalt of waar de fout zich precies bevindt.

2.6 Nut van log mangamenent

In de bedrijfswereld worden logs vooral gebruikt voor het opsporen van problemen in een netwerk. Logs kunnen duidelijkheid geven over de exacte server en applicatie waar iets fout is gegaan, ook kan het duidelijkheid geven over de reden waarom iets is mis gegaan. Het zou ook mogelijk zijn om proactief op te treden tegen fouten in een netwerk (sumologic, g.d.).

Logs kunnen ook gebruikt worden voor het controleren en verbeteren van de veiligheid van een systeem. De logs kunnen bijvoorbeeld aantonen hoeveel pogingen er zijn ondernomen

om in te loggen op een server, virtuele machine of virtuele container. Dit zou een indicatie kunnen zijn van een aanslag op die machine.

Er zijn ook nog specifiekere redenen waarvoor logs kunnen gebruikt worden. Zo kunnen logs gebruikt worden voor Internet woordenboeken te verbeteren (Bergenholtz & Johnsen, 2005) of kunnen web server logs gebruikt worden voor het verbeteren van de website design (Drott, 1998).

Logs en log files kunnen veel verschillende functies vervullen. Het is zeker interessant voor een student toegepaste informatica om aan te leren hoe er efficiënt aan log management kan gedaan worden.

2.7 Functie van log management tools

Log management tools bestaan meestal uit twee componenten. De eerste component is de aggregator. De aggregator ontvangt of verzamelt alle logs en log files uit verschillende bronnen en is verantwoordelijk voor het analyseren en visualiseren van deze logs. Wanneer de logs op een server staan die bereikbaar is voor de aggregator of de logs staan op de server waar de aggregator zelf op geïnstalleerd is, zijn er geen verdere acties meer nodig voor het installeren van een log management tool.

Wanneer de logs op een plaats staan die niet bereikbaar is door de aggregator zal een tweede component, de forwarder of shipper, moeten geïnstalleerd worden. Deze forwarder is een service die geïnstalleerd moet worden op de server, virtuele machine of virtuele container, waarvan de log files niet bereikbaar zijn door de aggregator. De forwarder is verantwoordelijk voor het verzamelen van de log files en deze door te sturen naar de aggregator.

Op deze manier kan een log management tool alle logs verzamelen van verschillende bronnen en deze op een centrale plaats bijhouden. Dit heeft als voordeel dat niet op alle verschillende servers afzonderlijk moet ingelogd worden om zo de log files te kunnen bekijken. Ook zal een log management tool ervoor zorgen dat er complexere zoekopdrachten kunnen uitgevoerd worden op de logs. Dit maakt het doorzoeken van de logs veel efficiënter.

2.8 Verschillende log management tools

2.8.1 Datadog

Datadog was oorspronkelijk enkel een cloud monitoring tool, maar hebben later log management toegevoegd. Datadog kan voor het verzamelen van de log files gebruik maken van zijn eigen agent of van een forwarder zoals Fluentd of Logstash.

Er is een gratis versie van Datadog maar daarin is container monitoring niet inbegrepen

en voor log management zelf is er enkel een free trial beschikbaar van veertien dagen (Datadog, g.d.).

2.8.2 Elastic

Elastic of Elasticsearch is een open-source en grotendeels gratis zoek- en analysemachine voor alle verschillende types van data en logs. Ook zal Elastic verantwoordelijk zijn voor het opslaan van deze data. Hoewel Elastic open-source is, wordt het verdeeld door Elastic. Dit is omdat Elastic ook functies aanbiedt die geen deel uitmaken van het open-source project en dan ook niet altijd gratis zijn. De betalende functies hebben dan vooral te maken met Elastic Cloud, dit laat de gebruiker toe om de Elastic service te deployen op eender welke cloud omgeving. De Elastic service kan echter zonder enige probleem op een eigen computer of server geïnstalleerd worden. Bepaalde specifieke functies in Elastic zelf zijn ook betalend (bijvoorbeeld de functies om email alerting of LDAP configuratie op te zetten) maar deze functies hebben geen invloed op het verzamelen, opslaan, analyseren en visualiseren van log files.

Het is hier echter belangrijk om op te merken dat Elastic enkel logs kan opslaan, doorzoeken en analyseren. Het is niet in staat om deze logs op te halen of te visualiseren. Om dit te verwezenlijken zal Elastic samen met twee andere services Logstash (verantwoordelijk voor het verzamelen van logs) en Kibana (verantwoordelijk voor het visualiseren van de logs) deel uitmaken van een ELK-stack of het zal met Fluentd (verantwoordelijk voor het verzamelen van de logs) en Kibana deel uitmaken van de EFK-stack. Zowel in de ELK-stack als in de EFK-stack werken de drie programma's in harmonie te samen om een volwaardig log managementtool te zijn. Het gebruik van drie verschillende programma's als één log management tool kan betekenen dat het installeren, gebruiken, configureren en onderhouden van deze log management tool te complex wordt. Het is eveneens mogelijk dat een ELK-stack of een EFK-stack veel resources nodig heeft om gebruikt te worden. Misschien vraagt het zelfs meer resources dan dat studenten toegepaste informatica tot hun beschikking hebben op hun laptops.

Enkele bekende klanten van Elastic zijn Adobe, Audi, Pfizer, Cisco, Rabobank en veel meer (Elastic, g.d.).

2.8.3 Fluentd

Fluentd vormt samen met Elasticsearch en Kibana een EFK-stack. Fluentd is in de EFK-stack niet enkel verantwoordelijk voor het verzamelen van de logs, maar het is ook verantwoordelijk voor het bewerken van de logs (bijvoorbeeld het parsen van logs, het toekennen en meegeven van bepaalde data aan logs en zoveel meer) en voor het versturen van de logs. Fluentd kan logs verzamelen van veel verschillende bronnen en kan deze logs versturen naar veel verschillende bestemmingen. Aangezien het uiteindelijke doel is om een centralized logging opstelling uit te werken zal er een Fluentd agent moeten geïnstalleerd worden op de server of computer waar de log files opstaan. Fluentd zal ook geïnstalleerd moeten worden op de server of computer waar Elasticsearch en Kibana

op geïnstalleerd staan. In een EFK-stack zullen alle logs uiteraard naar Elasticsearch doorgestuurd worden.

Het is mogelijk voor Fluentd om de logs naar andere services te sturen zoals Grafana Loki. Fluentd is eenflexibele software toepasbaar in veel verschillende opstellingen.

Fluentd is een open source en volledig gratis programma beheerd door de Cloud Native Computing Foundation (CNCF). Fluentd is geschreven in een combinatie van C language en Ruby. Ook beschikt het over meer dan vijfhonderd plugins waardoor Fluentd gemakkelijk en flexibel in gebruik zou moeten zijn. Het is trouwens mogelijk om zelf plugins te schrijven.

Naast Fluentd bestaat ook nog Fluent Bit. Fluent Bit heeft minder resources nodig om succesvol te kunnen draaien op een computer of server dan Fluentd en heeft meer features maar het is minder stabiel en minder goed ondersteund.

Fluentd heeft meer dan tweeduizend klanten waaronder: Atlassian, Microsoft, Amazon Web Serices (AWS) en Nintendo (Fluentd, g.d.).

Uit een bevraging gedaan door Datadog (2018) blijkt dat de Fluentd image het vierde meest gebruikte image is in Docker containers.

In het vervolg van dit onderzoek en voornamelijk in de methodologie zal niet meer naar Elasticsearch en Fluentd apart verwezen worden. Samen met Kibana zullen Elasticsearch en Fluentd gezien worden als één log management tool, de EFK-stack.

2.8.4 Grafana Loki

Grafana Loki of gewoon Loki is net zoals Elasticsearch een service die enkel de logs kan opslaan, analyseren en doorzoeken. Het heeft andere programma's nodig voor het verzamelen en visualiseren van logs. Loki maakt voor het verzamelen van de logs meestal gebruik van Promtail, maar dit kan even goed met Fluentd/Fluent-Bit. Voor het visualiseren van de logs maakt Loki gebruik van Grafana.

Loki is horizontally-scalable. Dit betekent dat de prestatie van Loki kan verbeterd of uitgebreid worden door er meer virtuele machines of containers van te maken. Waar bij verticale-scaling er meer resources zouden gegeven worden aan die specifieke virtuele machine of container. Loki is uitbreidbaar zonder dat deze service moet stoppen. Daardoor heeft Loki een high-availability. Loki zal de logs niet indexeren maar deze labels toekennen.

Loki is een project gestart door Grafana Labs in 2018 en is in hetzelfde jaar voor het eerst voorgesteld op KubeCon Seattle (Kaltschmidt, 2018). Loki is open source en gratis, maar er is ook een betalende cloud service en een betalende enterprise service (Grafana, g.d.).

2.8.5 GoAccess

GoAccess is een log management tool die enkel web logs (web logs zijn logs die afkomstig zijn van een webserver) kan verzamelen, analyseren en visualiseren. Dit log management tool is gemaakt voor snel te zijn en kan volledig gebruikt worden in de terminal. GoAccess kan een interessante log management tool zijn voor opstellingen die enkel bestaan uit webservers.

De toepassingen waar GoAccess in kan gebruikt worden zijn beperkt. Dit kan een nadeel zijn wanneer de opstelling complex is en uit meer bestaat dan enkel webservers. Echter zorgt dit ervoor dat GoAccess snel kan werken en niet onnodig ingewikkeld is in setup en gebruik.

GoAccess is open source en volledig gratis te gebruiken (GoAccess, g.d.).

2.8.6 Graylog

Graylog is een gratis open source log management tool die een betalende enterprise en cloud service aanbiedt. Graylog maakt voor de opslag van logs gebruik van Elasticsearch uit de ELK- en EFK-stack, maar in tegenstelling met de ELK- en EFK-stack zal Graylog alles in één service verpakken in plaats van drie verschillende services.

Het voordeel van alles als één service te hebben is dat de installatie gemakkelijker kan zijn, misschien is de configuratie ook eenvoudiger. Ook kan Graylog specifieke functies aanbieden die er niet zijn in de ELK- of EFK-stack. Het nadeel is dat enkel de Graylog API kan gebruikt worden en niet de Elasticsearch API, met als gevolg dat de ELK- of EFK-stack functies heeft die Graylog niet heeft. Aangezien Graylog geen gebruik zal maken van Kibana zal de visualisatie van de logs ook beperkter zijn (Graylog, g.d.).

2.8.7 Humio

Humio is een enterprise-grade log management tool. Met Humio kan honderden TeraBytes aan logs opgeslagen worden per dag, ook kan Humio logs verwerken en visualiseren in minder dan een seconde en kan het data vijf- tot vijftienmaal comprimeren waardoor er minder hardware nodig is om logs op te slaan, of deze langer kan bijgehouden worden.

Humio heeft een gratis versie maar deze is op veel verschillende manieren beperkt en is enkel te gerbuiken via hun cloud infrastructure (SaaS) en de self hosted versie heeft geen gratis versie. De focus wordt vooral gelegd op de betalende versies (Humio, g.d.).

2.8.8 LogDNA

LogDNA is een cloud gebaseerde log management tool. Deze kan niet op een eigen computer of server geïnstalleerd worden. Het heeft een gratis versie maar slechts één gebruiker kan deze gebruiken en slaat de logs niet op. Er zijn ook betalende versies maar

zelfs de duurste versie houdt de logs maar bij voor maximum dertig dagen.

Verder is LogDNA een standaard log management tool die de nadruk legt op security, gemakkelijk gebruik van de tool en een vermindering in kosten voor bedrijven aangezien deze geen eigen infrastructuur moeten voorzien om LogDNA te gebruiken (LogDNA, g.d.).

2.8.9 Loggly

Loggly is een log management tool van Solarwinds. Het is ook een cloud based log management tool een installatie op eigen infrastructuur is niet mogelijk. Loggly kan logs verzamelen en analyseren van alle mogelijke bronnen en wordt goed ondersteund door verschillende log forwarders zoals Fluentd en Logstash.

Er is een gratis versie beschikbaar maar deze is vrij beperkt. Het heeft ook betalende versies maar deze zijn ook allemaal beperkt in gebruikers (behalve de duurste versie).

Loggly heeft enkele grote klanten zoals: Lenovo, Pizza Hut, EA, Samsung en nog veel meer (Loggly, g.d.).

2.8.10 LOGIQ

LOGIQ is opnieuw een volledig cloud based log management tool. Het legt de focus op eenvoud in gebruik.

LOGIQ heeft een volledig gratis versie met enkele limitaties. Ook betaalde versies zijn beschikbaar met een betaling op maandelijkse basis (LOGIQ, g.d.).

2.8.11 Logstash

Logstash is een forwarder die alle logs verzamelt en doorstuurt. Samen met Elasticsearch en Kibana vormt het de ELK-stack. In een opstelling waar het doel is om de het analyseren van de logs op een centrale server, container of virtuele machine te zetten zal Logstash twee keer geïnstalleerd moeten worden.

Logstash zal moeten geïnstalleerd worden op de server, virtuele container of virtuele machine waar de log files op staan en waar ze vandaan komen. Daar zal Logstash verantwoordelijk zijn voor het verzamelen van deze logs en ze door te sturen naar de server, virtuele container of virtuele machine waar Elastichsearch en Kibana op geïnstalleerd staan.

Daar zal dan de tweede installatie van Logstash te vinden zijn. In die server, virtuele container of virtuele machine zal het dan de taak zijn van Logstash om de inkomende logs van de andere Logstash service of meerdere Logstash services te verzamelen en deze dan door te sturen naar Elasticsearch.

Logstash kan ook een forwarder zijn voor andere programma's op voorwaarde dat daar een plugin voor bestaat. Er bestaan meer dan tweehonderd plugins voor Logstash (Logstash, g.d.).

Logstash heeft ook een versie die minder resource intensief is, genaamd Log Beats. Logstash is net zoals de twee andere componenten van de ELK-stack open source, volledig gratis en het wordt beheerd door Elastic. Dit wil zeggen dat alle drie de services zonder problemen zouden moeten kunnen samenwerken. Logstash biedt echter geen enterprise versie aan (Logbeats, g.d.).

In het vervolg van dit onderzoek en voornamelijk in de methodologie zal niet meer naar Elasticsearch en Logstash apart verwezen worden. Samen met Kibana zullen Elasticsearch en Logstash gezien worden als één log management tool, de ELK-stack.

2.8.12 Logz.io

Logz.io is in essentie de ELK-stack maar dan aangeboden als een volledige cloud service. Het is alles wat de ELK-stack is, door hun opgezet en geconfigureerd op hun cloud omgeving. Hierdoor zou de setup minimaal en eenvoudig moeten zijn.

Over het algemeen kan Logz.io hetzelfde als een ELK-stack met nog een paar extra features die zij zelf voorzien hebben. Er is een gratis versie met beperkingen en meerdere betalende versies (Logz.io, g.d.).

2.8.13 Promtail

Promtail is de shipper vooral gebruikt in combinatie met Grafana Loki. Het is verantwoordelijk voor de log files te ontdekken, labels aan log streams toe te kennen en de logs te verzenden naar Grafana Loki.

Promtail wordt beheerd door Grafana maar is volledig gratis te gebruiken (Promtail, g.d.).

In het vervolg van deze proef en in de methodologie zal er niet verder verwezen worden naar Promtail. Telkens als naar Grafana Loki verwezen wordt moet er aangenomen worden dat Promtail als shipper gebruikt wordt.

2.8.14 Scalyr

Scalyr is een cloud log management en monitoring tool. Het spitst zich toe op het verwerken van veel data op een snelle manier en het zou eenvoudig moeten zijn in gebruik.

Er is geen gratis versie van Scalyr maar wel een gratis trial versie voor dertig dagen (Scalyr, g.d.).

2.8.15 Sematext

Sematext is een cloud service die monitoring en log management aanbiedt. Het is voornamelijk een cloud service omdat sematext ook een enterprise on premise service ter beschikking heeft. Enkel de cloud service heeft een gratis versie met limitaties, de andere versies hebben een 14 dagen gratis trial.

Sematext maakt gebruik van de Elasticsearch API en Kibana voor het opslaan, analyseren en visualiseren van logs en kan deze logs ontvangen van een forwarder die logs kan shippen naar Elastichsearch (Logstash en Fluentd). Het is een ELK-stack in één verpakking in plaats van drie verschillende services. Ook voegt Sematext enkele functies toe, door hun ontwikkeld.

Sematext is een ELK- of EFK-stack die een eenvoudigere setup zou moeten hebben dan de ELK- of EFK-stack en support voorziet voor hun gebruikers en klanten.

Enkele bekende klanten van Sematext zijn: Dell, Facebook, Instagram, BBC, Walmart en veel meer(Sematext, g.d.).

2.8.16 Splunk

Splunk is een cloud en on-premises service dat tracht een oplossing te zijn voor alles dat met data te maken heeft. Dit gaat dan over het verzamelen van logs, monitoren van infrastructuur, het beveiligen van data en zoveel meer. Log management is slechts een klein deel van Splunk maar log management is perfect mogelijk met Splunk. Aangezien Splunk veel meer kan dan enkel log management, kan het als een totaaloplossing fungeren voor bedrijven die ook aan monitoring willen doen.

Splunk heeft een gratis trial maar geen gratis versie, alle versies van Splunk zijn betalend (Splunk, g.d.).

2.8.17 Sumo logic

Sumo Logic is een log management en monitoring tool die enkel aangeboden wordt als cloud service. Het kan niet enkel log files verzamelen, analyseren en visualiseren maar het kan ook infrastructuren monitoren.

Er is een gratis versie van Sumo Logic beschikbaar met enkele limitaties. Uiteraard zijn er verschillende betalende versies (Sumologic, g.d.).

2.8.18 Extras

Er zijn ook nog andere log management tools zoals Pandora, rizhiyi en Tencent Cloud. Deze tools worden niet verder overwogen of onderzocht aangezien hun websites niet beschikbaar zijn in een Engelse versie. Er kan niet achterhaald worden wat deze tools

juist zijn en hoe ze werken tenzij de lezer de specifieke taal verstaat waarin de websites beschikbaar zijn.

3. Methodologie

In dit hoofdstuk wordt er in detail bekeken aan welke requirements de log management tools specifiek moeten voldoen. Omdat niet alle requirements even belangrijk zijn gaan deze verdeeld worden over verschillende groepen met een verschillend belang volgens de MoSCoW analyse. Concreet wil dit zeggen dat er vier groepen gaan zijn waarover de requirements verdeeld worden.

De eerste groep is “Must have”. Dit is een groep van requirements waaraan de log management tools moeten voldoen. Als een log management tool niet voldoet aan de requirements uit deze groep zullen ze niet verder overwogen worden.

De tweede categorie is “Should have”. De requirements uit deze groep zijn belangrijk en zouden een meerwaarde bieden aan het eindresultaat maar zijn niet noodzakelijk. Log management tools die aan deze requirements voldoen zullen uiteraard als meer waarschijnlijk bevonden worden om uiteindelijk gekozen te worden.

De derde groep is “Could have”. De requirements uit deze groep zouden een grote meerwaarde kunnen zijn maar worden eerder gezien als een luxe en niet belangrijk. De laatste categorie is “Will not have”. In deze categorie bevinden zich de requirements waar in deze opstelling geen rekening zal gehouden worden. Ze kunnen echter wel de doorslag geven tussen verschillende log management tools en de reden zijn waarom voor de ene gekozen wordt om uit te werken en niet voor de andere.

Verder zal in dit hoofdstuk ook nog besproken worden welke stappen allemaal zullen genomen worden doorheen de proef om uit te komen bij de beste log management tool voor logs te verzamelen, opslaan, analyseren en visualiseren uit virtuele containers.

3.1 Must have

De log management tool moet beschikken over een gratis versie of een trial versie die lang genoeg is.

De log management tool moet beschikken over een gratis versie of moet een trial versie hebben van minstens acht maanden. Er is hier gekozen voor minstens acht maanden aangezien de leerstof kan gegeven worden in het tweede semester waardoor de tool lang genoeg gratis is voor een volledig semester en een herexamen periode. Beter zou natuurlijk zijn dat het een onbeperkte tijd gratis is dan kan de leerstof gegeven worden in eender welk semester.

De log management tool moet zowel systeem- als applicatielogs kunnen verwerken.

Het log management tool moet in staat zijn om elk type logs te verzamelen. Het moet systeem logs kunnen verzamelen van alle soorten virtuele containers en het moet mogelijk zijn om logs te verzamelen van andere applicaties.

De minimum nodige resources die de log management tool nodig heeft om operationeel te zijn, mogen niet hoger zijn de specificaties van de laptops van de studenten toegepaste informatica in HOGENT.

De nodige resources om een log management tool te kunnen gebruiken mogen niet meer zijn dan de studenten hun laptops aankunnen. HOGENT heeft voor elk van zijn richtingen een laptop die ze aanraden om te kopen. Deze zal dan ook gezien worden als de laptop die elke student toegepaste informatica gebruikt.

3.2 Should have

De log management tool moet een on premise service aanbieden.

Het zou interessant zijn dat er een on premise optie is. Het installeren van deze log management tools in virtuele containers is altijd een goede oefening voor studenten. Ook kan er door het installeren van de software een beter beeld en begrip over de werking van de software vergaard worden.

De log management tool beschikt over een gratis versie zonder enige limitaties.

Alhoewel er met een trial versie of een gratis versie perfect een log management tool kan opgezet worden, zou het kunnen dat deze versie bepaalde functies blokkeert die interessant kunnen zijn voor studenten om aan te leren. Een volledig open en ongelimiteerde versie van de software verzekert dat de student alles over log management kan aangeleerd worden.

3.3 Could have

De log management tool heeft reeds een bestaande container image.

Het zou een goede bonus zijn als er reeds een Docker image bestaat voor de log management software. Dit maakt de installatie van de software eenvoudiger waardoor meer gefocust kan worden op de log management tool zelf. Tevens is dit interessant voor bedrijven, hoe gemakkelijker en hoe sneller iets te integreren is in een bedrijf hoe hoger de kans dat dit dan ook zal gebeuren.

De log management tool beschikt over de mogelijkheid om aan andere type monitoring te doen.

Als de log management software de mogelijkheid zou bieden om andere monitoring te verwezenlijken is dit uiteraard ook een pluspunt. Er zou dan kunnen gekeken worden naar het combineren van leerstof van monitoring en logging. De monitoring mogelijkheden zullen echter niet verder onderzocht worden aangezien dit geen deel uitmaakt van dit onderzoek.

3.4 Will not have

De log management tool heeft een cloud service ter beschikking.

De log management software die ook een cloud service kan aanbieden zou zeker een goede bonus zijn. Dit kan de software interessant maken voor bedrijven waar de student aan het werk kan gaan later. In deze studie zal het minder een rol spelen, de focus wordt gelegd op het installeren en configureren van de software omdat dit een grote meerwaarde is voor studenten en daarom is een on premise software meer gegeerd.

3.5 Fase 1: Vergelijkende studie

Tijdens de vergelijking is het de bedoeling om alle log management tools, die gevonden zijn in de literatuurstudie, met elkaar te vergelijken aan de hand van de MoSCoW-analyse. Het zal eerst de bedoeling zijn om per categorie elke requirement beter af te bakenen. Daarna zal voor elk log management tool achterhaald worden of deze voldoet aan de requirements.

Wanneer een log management tool niet voldoet aan één van de requirements uit de “must have” categorie zal deze niet verder worden opgenomen in het onderzoek, deze zullen dan ook niet meer bekeken worden voor de requirements in de daarop volgende categorieën.

In de “Should have” categorie zullen de resterende log management tools opnieuw met elkaar vergeleken worden aan de hand van de opgestelde requirements. Als een log management tool niet voldoet aan een of meerdere van deze requirement zal deze, net

zoals In de “Must have” categorie niet verder worden opgenomen in dit onderzoek.

Als na de eerste twee categorieën twee tot vijf log management tools overblijven zal de “Could have” categorie dienen om de resterende log management tools te rangschikken van meest naar minst geschikt. Indien er na de eerste twee categorieën toch meer dan vijf log management tools overblijven zal de “Could have” categorie ook als een filter dienen.

De “Will not have” categorie zal in dit onderzoek eerder gezien worden als een extra. De requirements en daarachter liggende functies zullen niet verder uitgewerkt worden in deze opstelling.

Indien de analyse geen duidelijkheid schept over het meest geschikte log management tool zullen de resterende tools meer in detail met elkaar vergeleken worden. Alle voor- en nadelen van deze tools zullen worden onderzocht om toch een uiteindelijke rangschikking te kunnen maken. In de volgende fase zal dat wat de hoogst gerangschikte log management tool een proof of concept gemaakt worden.

3.6 Fase 2: Proof of concept

Het einddoel van de proof of concept is om aan te tonen dat de, in de vorige fase verkregen log management tool, in staat is om log files te verzamelen uit verschillende virtuele containers. Het zal deze log files uiteraard ook moeten kunnen opslaan, analyseren en visualiseren. Wanneer dit lukt op een computer met dezelfde hardware specificaties als die aangeraden door HOGENT is dit een geslaagde proof of concept.

Er zullen meerdere versies zijn van de proof of concept. De eerste versie zal zich focussen op het installeren en configureren van de log management tool. De versies erna zullen dan meer focussen op het introduceren van best practices en wat er allemaal mogelijk is met een specifieke log management tool.

Wanneer later zou beslist worden om log management op te nemen in het curriculum toegepaste informatica dan zouden deze proof of concepts kunnen gebruikt worden als een gids of startpunt om de log management tool uiteindelijk aan te leren aan de studenten.

4. De vergelijkende studie

Het doel van de vergelijking is om uit te zoeken welke log management tool het meest geschikt is om op te nemen in het curriculum van toegepaste informatica.

Voor elke categorie uit de MoSCoW analyse zullen alle requirements overlopen worden en waar nodig zullen er specifieke limieten afgebakend worden. Dan zal er per categorie een tabel gemaakt worden met daarin elke, nog in aanmerking komende, log management tool en of ze al dan niet voldoen aan de opgestelde requirements.

Elke tabel zal dan verder verklaard worden waar nodig. De log management tools die niet meer in aanmerking komen zullen vanaf dat punt ook niet meer worden besproken.

Daarna zullen alle geschikte log management tools met elkaar vergeleken worden om te achterhalen welke het beste is om op te nemen in het curriculum toegepaste informatica.

4.1 Must have

De log management tools moeten voldoen aan al de vooropgestelde requirements uit deze categorie. Wanneer de tool niet voldoet aan één van deze requirements is deze niet geschikt en wordt ze niet verder onderzocht.

4.1.1 Afbakening requirements

De log management tool moet beschikken over een gratis versie of een trial versie die lang genoeg is.

De log management tool die gekozen wordt om op te nemen in het curriculum toegepaste informatica mag niet gepaard gaan met extra kosten aan HOGENT of de studenten. Om deze reden moet de log management tool zeker een gratis versie hebben of een trial versie hebben die minstens acht maanden geldig is. De trial versie moet minstens acht maanden duren want als de log management tool nodig is voor leerstof in het tweede semester is deze nog lang genoeg geldig voor de herexamens. Ook moet het mogelijk zijn voor éénzelfde student om meerdere keren de trial versie te starten. Het kan altijd zijn dat een student een vak meerdere jaren moet opnemen en voor die student moet het uiteraard ook gratis blijven.

In de tabel zal deze requirement de naam "**Gratis versie**" krijgen.

De log management tool moet zowel systeem- als applicatielogs kunnen verwerken.

De log management tool moet over de mogelijkheid beschikken om alle type log files uit alle verschillende soorten virtuele containers te kunnen verzamelen, opslaan, analyseren en visualiseren. Dit wil zeggen dat niet enkel uit apache en nginx containers log files moeten kunnen verzameld worden maar ook uit MySQL, MariaDB, Redis, Node en alle andere soorten virtuele containers.

In de tabel zal deze requirement de naam "**Log types**" krijgen.

De minimum resources die de log management tool nodig heeft om operationeel te zijn mogen niet hoger zijn de specificaties van de laptops van de studenten toegepaste informatica in HOGENT.

HOGENT heeft voor de bachelor toegepaste informatica computervereisten opgelegd. Aangezien het hebben van een laptop met die requirements een must is voor het modeltraject van de bachelor toegepaste informatica zal deze laptop aanschouwd worden als de laptop die elke student heeft. De minimum requirements van de log management tools zullen niet hoger mogen liggen dan de specificaties van de laptops.

De minimum computervereisten voor de bachelor toegepaste informatica zijn (HOGENT, g.d.):

- Processor: Intel i7 met een minimumcache van 6 MB (of gelijkaardige AMD processor).
- Werkgeheugen (RAM): 16GB (of meer).
- Harde schijf: 512GB SSD.
- Scherm: 15 inch full HD.
- Grafische kaart (GPU): dedicated met 4GB videogeheugen.

De grootste bottleneck voor het uitvoeren van een log management tool op een laptop zal het werkgeheugen zijn. Om deze reden zal in vergelijking enkel naar het werkgeheugen gekeken worden.

In de tabel zal deze requirement de naam "**Computer requirements**" krijgen.

Naam van de log management tool	Gratis versie	Log types	Computer requirements
Datadog	Gratis met beperkingen	Alle log types	voldoen
EFK-stack	Gratis*	Alle log types	voldoen
ELK-stack	Gratis*	Alle log types	voldoen
Grafana Loki	Gratis	Alle log types	voldoen
GoAccess	Gratis	Enkel web logs	voldoen
Graylog	Gratis	Alle log types	voldoen
Humio	Betalend	Alle log types	voldoen
LogDNA	Gratis met beperkingen	Alle log types	voldoen
Loggly	Gratis met beperkingen	Alle log types	voldoen
LOGIQ	Gratis met beperkingen	Alle log types	voldoen
Logz.io	Gratis met beperkingen	Alle log types	voldoen
Scalyr	Betalend	Alle log types	voldoen
Sematext	Betalend	Alle log types	voldoen
Splunk	Betalend	Alle log types	voldoen
Sumo logic	Gratis met beperkingen	Alle log types	voldoen

Tabel 4.1: Vergelijking 'Must have' categorie resultaten.

4.1.2 Bespreking van de vergelijking

Log management tools met Gratis* in de "**Gratis versie**" kolom in Tabel 4.1 hebben ook beperkingen en zijn daarom niet volledig gratis. Desondanks worden deze tools gezien als volledig gratis aangezien de betalende functies, functies zijn die sowieso niet zouden uitgewerkt worden in dit onderzoek.

Zoals te zien is in Tabel 4.1 voldoen vijf log management tools niet aan één of meerdere requirements. Deze log management tools zijn GoAccess, Humio, Scalyr, Sematext en Splunk. Deze tools zullen daarom niet verder opgenomen worden in de vergelijking.

De log management tools die wel voldoen aan alle requirements zijn: Datadog, EFK-stack, ELK-stack, Grafana Loki, Graylog, LogDNA, Loggly, LOGIQ, Logz.io en Sumo logic. Deze log management tools zullen verder worden onderzocht.

4.2 Should have

Net zoals in de vorige categorie zullen de log management tools moeten voldoen aan alle opgestelde requirements, als dit niet het geval is zal de log management tool niet verder worden opgenomen in de vergelijking en dit onderzoek.

Naam van de log management tool	On premis	Volledig gratis
Datadog	Enkel cloud service	Gratis met beperkingen
EFK-stack	On premise service	Gratis zonder beperking
ELK-stack	On premise service	Gratis zonder beperking
Grafana Loki	On premise service	Gratis zonder beperking
Graylog	On premise service	Gratis zonder beperking
LogDNA	Enkel cloud service	Gratis met beperkingen
Loggly	Enkel cloud service	Gratis met beperkingen
LOGIQ	Enkel cloud service	Gratis met beperkingen
Logz.io	Enkel cloud service	Gratis met beperkingen
Sumo logic	Enkel cloud service	Gratis met beperkingen

Tabel 4.2: Vergelijking 'Should have' categorie resultaten.

4.2.1 Afbakening requirements

De log management tool moet een on premise service aanbieden.

De reden voor het opstellen van deze requirement is tweevoudig. Enerzijds is het installeren van de log management tool een handige oefening voor de studenten en kan dit een beter inzicht geven op de werking van een log management tool en zoals in Tabel 4.1 te zien is, is dit allemaal perfect mogelijk op de laptops van de studenten. De andere reden is dat de kennis voor het installeren van de tool on premise de student meer gegeerd kan maken voor de toekomstige werkgever. Het kan zijn dat bedrijven zelf beschikken over een server farm en dan niet willen dat de tool beheerd wordt door een ander bedrijf of al de informatie van hun logs op een ander bedrijf zijn servers staat. Deze on premise service moet uiteraard gratis zijn.

In de tabel zal deze requirement de naam "**On premise**" krijgen.

De log management tool beschikt over een gratis versie zonder enige limitaties.

Uit de Tabel 4.1 blijkt dat er van sommige log management tools de gratis versie beperkingen heeft. Het zou beter zijn als de tool gratis is zonder enige beperking. Dit zou dan willen zeggen dat logs oneindig lang kunnen bijgehouden worden (wat interessant kan zijn om later in de proof of concept enkele best practices te introduceren). Ook maakt het dan niet uit hoeveel logs er worden doorgestuurd naar de log management tool. Hierdoor wordt vermeden dat er onverwachte kosten de kop opsteken.

In de tabel zal deze requirement de naam "**Volledig gratis**" krijgen

4.2.2 Bespreking van de vergelijking

Zoals te zien is in Tabel 4.2 zijn de log management tools die enkel een cloud service aanbieden niet gratis zonder enige beperking. Deze tools zijn: Datadog, LogDNA, Loggly, LOGIQ, logz.io en Sumo logic. Deze worden niet verder vergeleken en opgenomen in dit

Naam van de log management tool	Container image	Monitoring
EFK-stack	Image beschikbaar	Monitroing mogelijk
ELK-stack	Image beschikbaar	Monitroing mogelijk
Grafana Loki	Image beschikbaar	Monitroing mogelijk
Graylog	Image beschikbaar	Monitroing niet mogelijk

Tabel 4.3: Vergelijking 'Could have' categorie resultaten.

onderzoek.

De log management tools die voldoen aan alle requirements zijn: EFK-stack, ELK-stack, Grafana Loki en Graylog. Deze tools zullen verder worden onderzocht.

4.3 Could have

Aangezien er na de vergelijking in de vorige categorie slechts vier log management tools overblijven zullen de requirements uit deze en volgende categorieën eerder dienen om de tools te rangschikken. Ondanks de resultaten van komende requirements blijft elke tool overwogen worden en vergeleken worden.

4.3.1 Afbakening requirements

De log management tool heeft reeds een bestaande container image.

Aangezien Docker met images werkt voor het opzetten van virtuele containers zou het handig zijn om te achterhalen of een log management tool reeds een bestaande image heeft. Dit zou de installatie en het opzetten van de log management tool container of containers veel eenvoudiger en sneller maken.

In de tabel zal deze requirement de naam "**Container image**" krijgen.

De log management tool beschikt over de mogelijkheid om aan andere type monitoring te doen.

Wanneer overwogen wordt om log management tool in het curriculum te introduceren, kan de kans dat deze weldegelijk wordt opgenomen vergroten door meer te kunnen dan enkel log management. Als elementen van een bepaalde tool kunnen gebruikt worden om andere monitoring te verwezenlijken is dit een bonus.

In de tabel zal deze requirement de naam "**Monitoring**" krijgen.

Naam van de log management tool	Cloud service
EFK-stack	Geen cloud service beschikbaar
ELK-stack	Geen cloud service beschikbaar
Grafana Loki	Geen cloud service beschikbaar
Graylog	Cloud service beschikbaar

Tabel 4.4: Vergelijking 'Will not have' categorie resultaten.

4.3.2 Bespreking van de vergelijking

Uit Tabel 4.3 kan vastgesteld worden dat EFK-stack, ELK-stack en Grafana Loki een voordeel hebben ten opzichte van Graylog. Aan de hand van deze requirements lijken deze drie tools beter geschikt te zijn om op te nemen in de bacheloropleiding toegepaste informatica en hebben een hogere kans om uiteindelijk gekozen te worden.

4.4 Will not have

Net zoals in de vorige categorie is deze categorie eerder voor het rangschikken en het vergelijken van de voor- en nadelen van de log management tools en niet om de tool uit te sluiten van het verdere onderzoek.

4.4.1 Afbakening requirements

De log management tool heeft een cloud service ter beschikking.

Een log management tool die beschikt over een cloud service heeft meer kans om gebruikt te worden door bedrijven. De studenten aanleren hoe ze deze tools juist configureren kan ze beter voorbereiden op hun toekomstige werk, wat een groot doel is van de bachelor opleiding toegepaste informatica.

In de tabel zal deze requirement de naam "**Cloud service**" krijgen.

4.4.2 Bespreking van de vergelijking

In tegenstelling tot Tabel 4.3 kan geconcludeerd worden uit Tabel 4.4 dat het beter zou zijn om uiteindelijk voor Graylog te kiezen aangezien deze als enige aan de requirement voldoet. De laatste twee categorieën bieden geen duidelijkheid over welke tool nu juist de beste zou zijn.

4.5 Uiteindelijke keuze

Nu kan er specifiek en meer in detail gekeken worden naar deze vier log management tools.

Zo is te zien dat de ELK-stack één van de oudere log management tools is. Het is bewezen dat deze werkt, stabiel en goed ondersteund is maar de ELK-stack blijkt dan niet optimaal te zijn voor virtuele containers.

Graylog heeft dan als voordeel dat het alles is wat de ELK-stack is, maar dan met een gemakkelijkere installatie.

Grafana Loki is de meest recent ontwikkelde log management tool, de Docker image is het meest gedownload en de Grafana visualisatie zou superieur zijn aan de Kibana visualisatie.

De EFK-stack zou alle voordelen moeten hebben die de ELK-stack heeft maar Fluentd zou beter geschikt moeten zijn om te gebruiken in een omgeving met virtuele containers en zou minder resources nodig hebben.

Uiteindelijk zijn alle vier deze log management tools geschikt om mee verder te gaan in deze bachelorproef. Desondanks zal er voor het vervolg van de bachelorproef gekozen worden om verder te werken met de EFK-stack. De keuze voor de EFK-stack is tweevoudig.

De eerste reden dat voor de EFK-stack is gekozen is Docker. Docker heeft standaard een paar log mechanismes ter beschikking. Zo'n mechanisme heet een Docker logging driver. Één van die standaard logging drivers is Fluentd. Dit zal tijdens het uitwerken van de proof of concept ervoor zorgen dat er geen agent moet geïnstalleerd worden op de containers. Dit maakt de installatie en configuratie voor de EFK-stack veel eenvoudiger dan voor de andere log management tools.

De tweede reden dat er voor de EFK-stack is gekozen is de flexibiliteit van Fluentd. Fluentd kan niet enkel gebruikt worden voor logs te sturen naar Elasticsearch maar Fluentd kan ook logs sturen naar Grafana (als alle juiste plugins geïnstalleerd zijn).

5. Proof of concept

Deze proof of concept zal dienen als bewijs dat de gekozen log management tool kan geïnstalleerd, geconfigureerd en uitgevoerd kan worden op een computer of laptop met gelijkaardige requirements als die vooropgesteld door HOGENT voor de bacheloropleiding toegepaste informatica.

De proof of concept bestaat uit meerdere versies, dit komt door het iteratief verbeteren van de configuraties en opstelling van de proef. Op het einde van elke versie wordt achterhaald of aan alle einddoelen voldaan is. Wanneer dit niet het geval is zal er een nieuwe iteratie gestart worden.

Deze proof of concept is uitgewerkt op een computer met volgende hardware:

- Processor: Intel i7 4790 @ 3.60GHz
- Moederbord: MSI Z87-G45
- RAM: 16GB DDR3
- Grafische kaart: AMD Radeon RX 5700 XT
- Besturingssysteem: Microsoft Windows 10 Pro

Voor het opzetten van Linux containers is er echter een systeem nodig met een Linux besturingssysteem. Om deze reden wordt deze proof of concept volledig gedaan in een virtuele machine. Dit representeert ook beter de situatie die zich bij studenten zal voordoen, aangezien veel studenten een laptop hebben met een Microsoft Windows besturingssysteem. De virtuele machine beschikt over volgende specificaties:

- Processor: 4
- RAM: 8192 MB
- Video Memory: 16MB

- Besturingssysteem: Ubuntu (64-bit)
- Adapter 1: Bridged Adapter

De volledige opstelling zal gemaakt worden in een virtuele machine waarvan de specificaties significant lager zijn dan de computervereisten die HOGENT oplegt.

Een belangrijke opmerking: tijdens deze proof of concept wordt een aantal keer verwezen naar het host systeem. Ongeacht of de opstelling gemaakt wordt op een Linux machine of in een virtuele machine die Linux draait, wordt met de host steeds het Linux systeem bedoeld. Er zal nooit iets in Microsoft Windows moeten gedaan worden.

5.1 Requirements

Voor de proof of concept succesvol kan uitgevoerd worden moet eerst aan een paar requirements voldaan worden. Deze requirements zijn voor alle versies van deze proof of concept hetzelfde daarom worden deze eerst beschreven. Ook maakt de installatie ervan geen deel uit van dit onderzoek maar wordt hier toch beschreven voor de volledigheid.

5.1.1 Docker installatie

De installatie van Docker kan met een paar simpele commando's. De gebruikte commando's zijn hieronder beschreven.

```
$ curl -fsSL https://get.docker.com/ -o get-docker.sh
$ sudo sh get-docker.sh
$ rm get-docker.sh
$ sudo usermod -aG docker <username>
```

1. ***<username>*** : Dit moet de naam zijn van de user. Dit commando zorgt ervoor dat later niet alle commando's van Docker als sudo moeten worden uitgevoerd.

5.1.2 Docker Compose installatie

Het installeren van Docker Compose is net zoals de installatie van Docker met slechts enkele commando's gedaan.

```
$ sudo curl -L "https://github.com/docker/compose/
   releases/download/1.26.0/docker-compose-$(uname -s
   )-$(uname -m)" -o /usr/local/bin/docker-compose
$ sudo chmod +x /usr/local/bin/docker-compose
```

5.2 Einddoel proof of concept

Het einddoel van de proof of concept is wanneer er uit meerdere virtuele containers tegelijk logs kunnen verzameld, opgeslaan, geanalyseerd en gevisualiseerd worden. Dit is uiteraard de bedoeling om met een log management tool te doen meer specifiek zal dit in deze opstelling gebeuren met de EFK-stack (reden voor de keuze van deze tool kan gevonden worden in fase 1: De vergelijking).

Het zal mogelijk moeten zijn om complexe filtering te kunnen toepassen op logs van alle virtuele containers uit deze opstelling. Dit toont aan dat alle logs gecentraliseerd zijn en er analyses op kunnen uitgevoerd worden. Er moet ook een dashboard kunnen gemaakt worden met daarin nuttige statistieken. Ook zullen er best practices moeten geïntroduceerd worden. Zo moet het duidelijk zijn van waar de logs juist afkomstig zijn en moeten er voorzieningen genomen worden om na een bepaalde tijd de logs te verwijderen om te voorkomen dat een harde schijf vol komt te staan met enkel logs.

Als laatste uitbereiding kan gekeken worden of het mogelijk is om logs te verzamelen van het host systeem zelf. Dit kan een perfecte opstelling zijn om na te gaan of het verzamelen van logs uit virtuele containers en het verzamelen van logs uit gewone servers gecombineerd kan worden. Het is ook een ideale manier om custom parsing te bekijken en na te gaan of het introduceren van de best practices werkt.

De installatie van alle containers en alle andere benodigdheden moet gebeuren met Docker en Docker Compose. Dit moet ervoor zorgen dat de installatie en configuratie zo eenvoudig en minimalistisch mogelijk blijft.

5.3 Proof of concept versie 1

Dit is een eerste iteratie van de proof of concept. Als deze voldoet aan alle requirements beschreven in het einddoel dan zal dit de enige iteratie zijn, als dit niet het geval is zullen er nog meer iteraties volgen.

5.3.1 Bestandsstructuur aanmaken

Het is eerst belangrijk om de juist bestandsstructuur te maken. Als hier een fout optreedt, kunnen alle volgende versies ook falen. Eerst zal in stappen uitgelegd worden welke bestanden er moeten aangemaakt worden, op het einde volgen de commando's waarmee dit kan gedaan worden.

1. Open de geprefereerde command line interface (CLI).
2. Maak een folder aan met de naam "EFK".
3. Verander de current directory naar de EFK folder.
4. Maak nu een folder "fluentd".
5. Verander de current directory naar de fluentd folder.

6. Maak nu een folder met de naam “conf”.

Gebruikte commando’s:

```
$ mkdir EFK
$ cd EFK
$ mkdir fluentd
$ cd fluentd
$ mkdir conf
```

5.3.2 Docker container installatie

docker-compose.yml

Een Compose file is een YAML bestand waarin alles beschreven staat om één of meerdere Docker containers op te starten met slechts 1 commando. Eerst zullen de te nemen stappen beschreven worden, daarna volgen de gebruikte commando’s en verdere uitleg over de bestanden.

1. Maak in de “EFK” folder een bestand aan genaamd “docker-compose.yml”
2. Open dit bestand met de geprefereerde editor.
3. Plaats onderstaande code in dit bestand.

```
$ touch docker-compose.yml
$ nano docker-compose.yml
```

Het bestand moet er als volgt uitzien.

```
version: "3"

services:
  web:
    image: httpd
    ports:
      - "80:80"
    links:
      - fluentd
    logging:
      driver: "fluentd"
      options:
        fluentd-address: localhost:24224
        tag: httpd

  fluentd:
    build: ./fluentd
    volumes:
      - ./fluentd/conf:/fluentd/etc
```

```
    links:
      - "elasticsearch"
    ports:
      - "24224:24224"
      - "24224:24224/udp"

  elasticsearch:
    image: elasticsearch:7.12.1
    expose:
      - 9200
    ports:
      - "9200:9200"
    environment:
      - "discovery.type=single-node"

  kibana:
    image: kibana:7.12.1
    links:
      - "elasticsearch"
    ports:
      - "5601:5601"
```

1. ***version: "3"*** : Dit is de versie van de Docker Compose file format.
2. ***services:*** : Dit zijn alle services/containers die zullen opstarten.
3. ***web:*** : Dit is de naam van de service/container.
   (a) ***image: httpd*** : Dit is de naam van de image die Docker moet gebruiken.
   (b) ***ports:*** : Dit zijn de poorten die een service/container nodig heeft om te kunnen werken.
   (c) ***links:*** : Dit is een extra alias waarvoor een server bereikbaar kan zijn, hier zal fluentd bereikbaar zijn vanuit web.
   (d) ***logging:*** : Docker heeft verschillende drivers beschikbaar, één daarvan is fluentd en deze zal dan ook gebruikt worden. De tag is een naam die zal worden meegegeven met alle logs komende van die container/server. Deze een nuttige naam geven is belangrijk aangezien deze later in de fluentd configuratie bestanden zal gebruikt worden.
4. ***fluentd:***
   (a) ***build:*** : Het bestand dat ter beschikking wordt gesteld tijdens het builden van de container (hier zal Docker zoeken naar een Docker file).
   (b) ***volumes:*** : Het pad van een file op de host die gekopieerd wordt naar het pad in de container.
5. ***elasticsearch:***
   (a) ***expose:*** : Dit maakt duidelijk aan Docker dat er op deze poort geluisterd wordt.
   (b) ***environment:*** : Voegt een environment variable toe (dit was nodig om versie 7.12.1 werkende te krijgen).
6. ***kibana:***

Dockerfile

Deze Dockerfile is verantwoordelijk voor het installeren van de Fluentd Elastichsearch plugin. Er is namelijk een plugin nodig in Fluentd om het sturen van logs naar Elasticsearch mogelijk te maken.

1. Maak in “EFK” > “fluentd” een bestand met de naam “Dockerfile”.
2. Open dit bestand met de geprefereerde editor.
3. Plaats onderstaande code in dit bestand.

```
$ touch Dockerfile
$ nano Dockerfile
```

Het bestand moet er als volgt uitzien.

```
# fluentd/Dockerfile
FROM fluent/fluentd:v1.6-debian-1
USER root
RUN ["gem", "install", "fluent-plugin-elasticsearch",
    "--no-document", "--version", "5.0.3"]
```

1. ***FROM*** : Dit is de base image die gebruikt moet worden om de container te builden.
2. ***USER*** : Dit is de user die zal gebruikt worden om de container te builden.
3. ***RUN*** : Dit is het uit te voeren commando, hier het installeren van de plugin.

5.3.3 Fluentd configuratie

fluent.conf

Dit bestand is verantwoordelijk voor de juiste configuratie van Fluentd.

1. Maak in “EFK” > “fluentd” > “conf” een bestand met de naam “fluent.conf”.
2. Open dit bestand met de geprefereerde editor.
3. Plaats onderstaande code in dit bestand.

```
$ touch Dockerfile
$ nano Dockerfile
```

Het bestand moet er als volgt uitzien.

```
# fluentd/conf/fluent.conf
<source>
  @type forward
  port 24224
</source>
<match *.**>
  @type copy
  <store>
```

```
    @type elasticsearch
    host elasticsearch
    port 9200
    index_name fluentd
    type_name fluentd
  </store>
  <store>
    @type stdout
  </store>
</match>
```

1. ***<source>*** : Dit maakt duidelijk waar Fluentd de logs moet zoeken, hier is dit forward aangezien de logs uit de containers komen en dit de centrale EFK server is.
2. ***<match *.**>*** : De ster tekens naast match is de naam van de tag (eerder gedefinieerd). Het gaat een specifieke tag matchen en zal dan een copy maken, één om naar Elasticsearch door te sturen en één gewoon naar de CLI. Voor de copy die naar Elasticsearch gestuurd wordt krijgen de logs een index naam, deze is belangrijk voor de Elastichsearch configuratie.
3. ***index_name*** : Een index is een verzameling van logs en de fields waaruit een log bestaat.

De mappenstructuur zou er als volgt moeten uitzien.

- EFK
  - docker-compos.yml
  - fluentd
    - conf
      - fluent.conf
    - Dockerfile

5.3.4 Opstarten Docker containers

Als alles geconfigureerd is kunnen nu alle Docker containers gemaakt en opgestart worden. Indien dit de eerste keer is kan het downloaden van de images en de installatie enkele minuten duren (afhankelijk van de internetsnelheid). Dit proces kan gestart worden met slechts één commando aangezien er met Docker Compose gewerkt wordt.

1. Start vanuit de EFK folder de containers.

```
$ docker-compose up
```

De installatie kan enkele minuten duren. Ter controle kan eerst gekeken worden of de Docker containers aangemaakt zijn, dit kan gedaan worden door het onderstaande commando in te voeren.

```
$ docker ps
```

Nadat de containers up and running zijn kan gecontroleerd worden of deze correct werken. Navigeer, in een browser naar keuze, naar http://localhost:9200. Daar zou een JSON configuratie zichtbaar moeten zijn met als onderste lijn: “tagline: “You Know, for Search”” Als dit het geval is werkt Elasticsearch zoals het moet. Navigeer vervolgens naar http://localhost:5601. Daar zou de Home pagina van Elastic moeten laden. Als dit het geval is wilt dit zeggen dat Kibana werkt zoals het moet en deze kan communiceren met Elasticsearch.

Nadat alles werkt moeten er enkele logs gemaakt worden. Dit kan op twee manieren.

De eerste manier is door op de host machine in de geprefereerde browser te navigeren naar http://localhost:80. Op deze webpagina zou de tekst “It works!” moeten verschijnen. Herlaad deze pagina een tiental keer en per keer de pagina wordt herladen wordt er een log aangemaakt.

De tweede manier is door het onderstaande commando uit te voeren in de CLI. Elke keer dit commando uitgevoerd wordt zal er een nieuwe log gemaakt worden.

```
$ curl http://localhost:80
```

5.3.5 Elasticsearch configuratie

Zoals te zien is op de http://localhost:5601 web interface staan er nog nergens logs om te analyseren of deze te kunnen visualiseren in de vorm van een dashboard. Elasticsearch moet nog verder geconfigureerd worden om de logs weer te kunnen geven. Voor alle functies van Elasticsearch en Kibana te kunnen gebruiken zal een index pattern moeten gemaakt worden.

Index management

Eerst is het belangrijk om na te gaan of de logs binnenkomen. Dit gebeurt volledig op de web interface van Elasticsearch/Kibana die te vinden is op volgende locatie: http://localhost:5601.

1. Open het linker pop-out menu.
2. Navigeer in het menu naar “Management” > “Stack Management”.
3. Navigeer naar “Data” > “Index management” (in de menu links op de webpagina).
4. Nu zou er een lijst moeten verschijnen met daarin een index met de naam fluentd.

Als dit het geval is werkt alles zoals het moet.

Index pattern

Het aanmaken van een index pattern gebeurt volledig op http://localhost:5601 en volgende stappen moeten ondernomen worden.

1. Open het linker pop-out menu.
2. Navigeer daarin naar “Management” > “Stack Management”.

3. In het menu links op de webpagina navigeer naar “Kibana” > “Index Patterns”.
4. Klik daar op de blauwe rechthoekige knop met daarin “+ Create index pattern” geschreven.
5. Plaats in index pattern name “f*”, dit zal alle indexen die starten met een “f” samen nemen.
6. Klik op “Next step >” en vervolgens op “Create index pattern”.

Nu zou er een index pattern moeten gemaakt zijn.

Discover

Complexe filtering en analyse van de logs wordt op de Discover pagina van Elasticsearch gedaan.

1. Open het linker pop-out menu.
2. Navigeer naar “Analytics” > “Discover”.
3. Als dit een nieuwe installatie is van de EFK-stack zou het kunnen dat hier nog niets te zien is, dit is volkomen normaal.
4. Op de linker kant van deze webpagina staat een witte rechthoek met daarin “ * ”. Klik hierop.
5. Selecteer “f*”.

Op de linker kant van deze webpagina zijn ook alle fields te zien waarop gefilterd kan worden. Deze fields zijn: _index, _id, _score, _type, container_id, container_name, log en message. Wanneer er enkel interesse is in het bekijken van bepaalde fields kan er met de muis over deze field bewogen worden en dan op de blauwe “+” knop geklikt worden die tevoorschijn komt. Zo kunnen de logs vertoond worden in kolommen naar keuze.

Deze fields kunnen ook gebruikt worden om de logs te filteren. Dit kan gedaan worden door op de “+ Add filter” knop te klikken, een field te selecteren, een operator te kiezen en dan een value in te typen. In deze versie is dit niet echt nuttig aangezien alle logs van dezelfde plaats komen en hetzelfde zijn. Ook kan gezien worden dat de tijd wanneer de logs zijn aangemaakt er gewoon als tekst tussen staat en niet in een date format. Hierdoor kan er minder filtering gedaan worden en zijn de opties voor een dashboard te maken ook beperkt.

5.3.6 Kibana configuratie

Kibana zal vooral gebruikt worden voor het visualiseren van de logs in dashboards. Zoals eerder vermeld zullen de opties voor het opstellen van dashboards beperkt zijn door een beperkt aanbod aan fields. Dashboards worden als volgt gemaakt.

1. Open het linker pop-out menu.
2. Navigeer naar “Analytics” > “Dashboard”.
3. Als dit de eerste keer is dat een dashboard word gemaakt zal er in het midden van de webpagina een grote blauwe rechthoek staan met daarin “+ Create new dashboard” geschreven, klik op deze knop.

4. Klik op de knop “+ Create panel”.
5. In dit geval zal enkel een “Lens” kunnen gemaakt worden, voor alle andere opties zijn er niet genoeg of niet de juiste fields.
6. Op deze webpagina moet opnieuw “ * ” veranderd worden naar “f*”.
7. Nu kan er een combinatie gekozen worden tussen de fields die gevisualiseerd moeten worden en met welk type grafiek dit gedaan moet worden. De opties zijn eindeloos, elke mogelijke combinatie overlopen is niet mogelijk. In de huidige opstelling zijn er weinig nuttige grafieken te maken.
8. Sleep de “Records” field in het grote open vlak in het midden van het scherm.
9. Klik onder Suggestions op de meest rechtse optie (dit zou een nummer moeten zijn, in dit geval het aantal logs).
10. Klik vervolgens op “Save and return”.
11. Klik op “Save”.
12. Geef het dashboard een naam en klik op “Save”.

Nu is succesvol een eerste Dashboard gemaakt.

5.3.7 Eindsituatie

Er is een volledig werkend proof of concept. De EFK-stack is up and running en het is mogelijk om logs te verzamelen, op te slaan, te analyseren en te visualiseren uit virtuele containers.

Echter is niet voldaan aan alle vooropgestelde einddoelen. De installatie van de log management tool is goed maar aan de configuratie moet nog gewerkt worden. De fields waar filtering op toegepast kan worden, moeten nog uitgebreid worden. De indexering kan beter, er moet een cleanup strategie geïntroduceerd worden en er moeten meerdere containers opgezet worden waarvan logs verzameld kunnen worden. Ook worden er nog geen logs verzameld van het host systeem. De tijd in de “time” field moet de tijd zijn wanneer de log gemaakt is, nu is die tijd de tijd wanneer de log is binnengekomen in Elasticsearch. Hierdoor kunnen geen tijd gerelateerde grafieken en zoekopdrachten verwezenlijkt worden.

5.4 Proof of concept versie 2

In deze tweede versie van de proof of concept is het de bedoeling om alle of enkele van de tekortkomingen te verbeteren uit de vorige versie. De configuraties en instellingen die identiek zijn aan de vorige opstelling worden niet herhaald of opnieuw uitgelegd. Daarvoor wordt verwezen naar de vorige versie.

5.4.1 Verwijderen Docker containers

Als deze proof of concept wordt gedaan na het uitvoeren van de vorige versie moeten de reeds gemaakte virtuele containers gestopt en verwijderd worden. Dit kan met volgende commando's.

```
$ docker stop efk_web_1 efk_fluentd_1 efk_kibana_1
   efk_elasticsearch_1
$ docker rm efk_web_1 efk_fluentd_1 efk_kibana_1
   efk_elasticsearch_1
```

5.4.2 Bestandsstructuur aanmaken

Deze moet niet aangepast worden en blijft hetzelfde als die uit de vorige versie.

5.4.3 Docker container installatie

Docker-compose.yml

Deze wordt op dezelfde manier gedaan als in de eerste versie, enkel de inhoud verschilt.

```
version: "3"

services:
  web:
    image: httpd
    ports:
      - "80:80"
    links:
      - fluentd
    logging:
      driver: "fluentd"
      options:
        fluentd-address: localhost:24224
        tag: httpd.access
  web2:
    image: httpd
    ports:
      - "81:80"
    links:
      - fluentd
    logging:
      driver: "fluentd"
      options:
        fluentd-address: localhost:24224
        tag: httpd2.access
```

```
fluentd:
  build: ./fluentd
  volumes:
    - ./fluentd/conf:/fluentd/etc
  links:
    - "elasticsearch"
  ports:
    - "24224:24224"
    - "24224:24224/udp"

elasticsearch:
  image: elasticsearch:7.12.1
  expose:
    - 9200
  ports:
    - "9200:9200"
  environment:
    - "discovery.type=single-node"

kibana:
  image: kibana:7.12.1
  links:
    - "elasticsearch"
  ports:
    - "5601:5601"
```

1. ***web1:***
   (a) ***tag:*** : Hier is de naam van de tag aangepast.
2. Er is een extra web2 bijgekomen, deze dient als een tweede container om te achterhalen of meerdere containers kunnen gelogd worden. Ook wordt hiermee achterhaald hoe de tags werken en of elke tag een verschillende ***<match>*** blok kan hebben.
   (a) ***port:*** : Voor het hostsysteem wordt een andere (ongebruikte) port toegekend.
   (b) ***tag:*** : Er wordt een unieke naam gegeven.

Dockerfile

Identiek aan de Dockerfile uit de eerste versie.

5.4.4 Fluentd configuratie

fluent.conf

Deze wordt op dezelfde manier gedaan als in de eerste versie, enkel de inhoud verschilt.

```
# fluentd/conf/fluent.conf
<source>
```

```
  @type forward
  port 24224
</source>

<filter httpd*.**>
  @type parser
  key_name log
  <parse>
    @type apache2
  </parse>
</filter>

<match httpd.access>
  @type copy
  <store>
    @type elasticsearch
    host elasticsearch
    port 9200
    include_timestamp true
    index_name hg-httpd.access
    type_name fluentd
  </store>
  <store>
    @type stdout
  </store>
</match>

<match httpd2.access>
  @type copy
  <store>
    @type elasticsearch
    host elasticsearch
    port 9200
    include_timestamp true
    index_name hg-httpd2.access
    type_name fluentd
  </store>
  <store>
    @type stdout
  </store>
</match>

<match *.**>
  @type copy
  <store>
    @type elasticsearch
```

```
    host elasticsearch
    port 9200
    index_name fluentd
    include_timestamp true
    include_tag_key true
    type_name fluentd
  </store>
  <store>
    @type stdout
  </store>
</match>
```

1. ***<filter httpd*.**>*** : Deze filter zal alle logs filteren die binnenkomen met een tag die start met httpd. Deze filter is verantwoordelijk voor het parsen van de logs die een tag hebben die start met httpd.
   (a) ***@type parser*** : Dit is het type van de filter.
   (b) ***key_name log*** : Het deel van de log waar de parsing moet op worden toegepast.
   (c) ***<parse>***
      i. ***@type apache2*** : het type parsing dat moet worden toegepast op de log, hier is een apache2 parsing nodig.
2. ***<match httpd.access>*** : Een match blok met een specifieke tag. Dit is zodat elke tag een eigen index naam kan krijgen.
   (a) ***include_timestamp true*** : Dit zal ervoor zorgen dat de "time" field in Elasticsearch dezelfde tijd zal zijn als de tijd dat in de log staat.
   (b) ***index_name hg-httpd2.access*** : Dit is de index naam voor de logs. Als er logs zijn die tezamen moeten gezien worden moet er een patroon in de naam van de index aanwezig zijn. Daarom zullen hier alle index namen beginnen met "hg" (hg staat voor hogeschool gent).
3. ***<match httpd2.access>*** : Op de naam van de tag en de index naam na, is deze configuratie identiek al de vorige match blok.
4. ***<match *.**>*** : Deze blok zal de overblijvende logs naar Elasticsearch sturen. Dit zijn dan meestal logs van Fluentd zelf.

5.4.5 Opstarten Docker containers

Het opstarten en nagaan of alle virtuele containers opgestart zijn en services naar behoren werken , is hetzelfde als in de proof of concept versie één. Nu zal er enkel een extra web server gestart worden.

Bij het maken van logs moet nu niet enkel naar http://localhost:80 gesurft worden en deze pagina een paar keer herladen maar moet dit ook gedaan worden voor http://localhost:81. Onderstaande commando's kunnen ook enkele keren herhaald worden in een CLI.

```
$ curl http://localhost:80
$ curl http://localhost:81
```

5.4.6 Elasticsearch configuratie

Net zoals in de vorige versie van de proof of concept zal er een index pattern moeten gemaakt worden voordatde logs kunnen gevisualiseerd en geanalyseerd worden.

Index management

Opnieuw is het belangrijk om te kijken of alle logs wel binnenkomen in Elasticsearch.

1. Open het linker pop-out menu.
2. Navigeer in het menu naar “Management” > “Stack Management”.
3. Navigeer naar “Data” > “Index management” (in het menu links op de webpagina).
4. Nu zou er een lijst moeten verschijnen met daarin drie indexen genaamd: hg-httpd.access, hg-httpd2.access en fluentd (het kan zijn dat de fluentd index er niet tussen staat).

Als dit het geval is kan een index pattern gemaakt worden.

Index pattern

Het maken van een index pattern zal nu licht verschillen met de vorige versie aangezien we dit keer een timefield definiëren en meegeven. Ook zal er een andere pattern gedefinieerd worden aangezien de indexen een andere naam hebben gekregen.

1. Open het linker pop-out menu.
2. Navigeer daarin naar “Management” > “Stack Management”.
3. In het menu links op de webpagina navigeer naar “Kibana” > “Index Patterns”.
4. Klik daar op de blauwe rechthoekige knop met daarin “+ Create index pattern” geschreven.
5. Plaats in index pattern name hg*, dit zal alle indexen die starten met “hg” samennemen.
6. Klik op “Next step >”.
7. In het drop down menu timefield selecteer @timestamp en klik vervolgens op “Create index pattern”.

Nu zou er een index pattern moeten gemaakt zijn voor alle indexen die starten met hg en er moet een time field gedefinieerd zijn.

Discover

Complexe filtering en analyse van de logs wordt op de Discover pagina van Elasticsearch gedaan.

1. Open het linker pop-out menu.
2. Navigeer naar “Analytics” > “Discover”.
3. Als dit een nieuwe installatie is van de EFK-stack zou het kunnen dat hier nog niets te zien is, dit is volkomen normaal.
4. Op de linker kant van deze webpagina staat een witte rechthoek met daarin “ * ”.

Klik hierop.
5. Selecteer “hg*”. Als dit niet helpt, moet rechtsboven op het scherm de tijd aangepast worden.

Opnieuw kunnen in de linker kolom van de webpagina alle fields gezien worden waarop gefilterd kan worden. Aangezien er een parser is gedefinieerd in de fluent.conf file zijn er nu andere fields te zien. Deze fields zijn: _id, _index, _score, _type, @timestamp, code, host, method en path. Hier is te zien dat @timestamp een nieuwe field is. Timestamp zal ervoor zorgen dat er correcter kan gefilterd worden op datums. Ook zal het ervoor zorgen dat er betere grafieken kunnen gemaakt worden in Kibana. Host, method en path zijn fields die er door de parser zijn bijgekomen.

5.4.7 Kibana configuratie

Het maken van een Lens in Kibana is identiek als in de vorige versie, alleen gaan er nuttigere grafieken kunnen gemaakt worden. Interessanter is om te kijken naar TSVB. Dit is een visualisatie optie van Kibana die een geavanceerdere analyse toelaat van tijd gebaseerde data en logs.

1. Open het linker pop-out menu.
2. Navigeer naar “Analytics” > “Dashboard”.
3. Als dit de eerste keer is dat een dashboard wordt gemaakt zal er in het midden van de webpagina een grote blauwe rechthoek staan met daarin “+ Create new dashboard” geschreven, klik op deze knop.
4. Klik op de knop “+ Create panel”.
5. Selecteer “TSVB”.
6. Hier kunnen allemaal verschillende instellingen aangepast worden. Het is belangrijk om op te merken dat onder “Panel options” > “Data” > “Index pattern” de juiste index pattern geselecteerd is.
7. Klik vervolgens op “Save and return”.
8. Klik op “Save”.
9. Geef het dashboard een naam en klik op “Save”.

Opnieuw zijn de mogelijkheden hier zo goed als eindeloos.

5.4.8 Eindsituatie

In vergelijking met de vorige versie van de proof of concept zijn er duidelijke verbeteringen. Het toevoegen van een extra virtuele container toont aan dat de EFK-stack gelijktijdig logs kan verzamelen van verschillende bronnen en deze op een centrale plaats kan verzamelen. Het geeft ook een beter beeld op een complexere configuratie van Fluentd en Elasticsearch en wat er allemaal mogelijk is met de configuratie ervan. In de vorige versie was er slechts één index, dit had als gevolg dat er geen selectie kon gemaakt worden in welke logs verwijderd worden, het was alles of niets. Het introduceren van meerdere indexes laat toe om de logs beter te kunnen onderscheiden van elkaar, de index patterns kunnen anders

geconfigureerd worden en nu worden ook niet alle logs verwijderd wanneer één index verwijderd wordt. Het toevoegen van de timestamp heeft ook een positieve impact gehad op deze opstelling.

Niet alle veranderingen zijn echter positief. Door de logs te parsen in Fluentd zijn alle fields verdwenen die meer informatie hadden over de virtuele containers (container_id en container_name fields). De informatie over de virtuele containers is, in deze opstelling, interessanter dan de fields die erbij komen door de logs te parsen.

Ook zijn nog niet alle einddoelen bereikt. Er moet nog steeds een cleanup strategie geïmplementeerd worden en er worden ook nog steeds geen logs verzameld van het host systeem zelf.

5.5 Proof of concept versie 3

Deze versie tracht alle problemen op te lossen uit de vorige versie. Ook zullen er nog enkele best practices geïntroduceerd worden en is het de bedoeling om logs van het host systeem te verzamelen. Dit wordt gedaan omdat alle logs van de virtuele containers worden opgeslagen in "var/lib/docker/containers". Er is een mogelijkheid om op die manier alle logs te verzamelen en niet enkel via de Docker logging driver. Ook kan hiermee worden aangetoond dat logs van verschillende soorten bronnen tegelijkertijd kunnen gelogd worden.

5.5.1 Verwijderen Docker containers

Het verwijderen van de containers is identiek aan de vorige versie.

5.5.2 Bestandsstructuur aanmaken

Om de opstelling van de EFK-stack zo eenvoudig mogelijk te houden is er een op maat gemaakte logfile voorzien in de bijlage B. Om alle onderstaande code te laten werken zonder aanpassingen zal deze log file op een specifieke plaats moeten gezet worden.

1. Maak in de map EFK een nieuwe map met de naam "testLog".
2. Plaats in die map de log file met naam "test.log".

```
$ mkdir testLog
```

Één log uit deze log file ziet er als volgt uit:

```
[2021-05-20 18:22:10,125]  INFO HOGENT GENT This is a
    test log
```

1. Alles tussen de "[]" is de datum en tijd.

2. INFO zal als field naam “level” krijgen. In deze logfile kan het INFO, WARN of ERROR zijn.
3. HOGENT zal als field naam “organization” krijgen. Deze kan HOGENT of – zijn.
4. GENT zal als field naam “city” krijgen. Deze kan de waarde GENT, AALST of – krijgen.
5. This is a test log zal als field naam “message” krijgen. Deze kan een klein beetje verschillen tussen de logs.

Deze log file is gemaakt om later met complexere filtering te kunnen oefenen en testen en te achterhalen wat er juist mogelijk is met die filtering.

5.5.3 Docker container installatie

Docker-compose.yml

```
version: "3"

services:
  web:
    image: httpd
    ports:
      - "80:80"
    links:
      - fluentd
    logging:
      driver: "fluentd"
      options:
        fluentd-address: localhost:24224
        tag: httpd.access
  web2:
    image: httpd
    ports:
      - "81:80"
    links:
      - fluentd
    logging:
      driver: "fluentd"
      options:
        fluentd-address: localhost:24224
        tag: httpd2.access

  fluentd:
    build: ./fluentd
    volumes:
      - ./fluentd/conf:/fluentd/etc
      - ./testLog:/fluentd/log/testLog
```

```
    links:
      - "elasticsearch"
    ports:
      - "24224:24224"
      - "24224:24224/udp"

  elasticsearch:
    image: elasticsearch:7.12.1
    expose:
      - 9200
    ports:
      - "9200:9200"
    environment:
      - "discovery.type=single-node"

  kibana:
    image: kibana:7.12.1
    links:
      - "elasticsearch"
    ports:
      - "5601:5601"
```

1. Het enige verschil met de vorige versie is dat er in Fluentd een extra volume is toegevoegd. Het eerste deel is waar de log file opgeslagen staat op de host, het tweede deel is waar die moet opgeslagen worden in de virtuele container.

Dockerfile

Dockerfile is hetzelfde als de vorige versies. Hier zijn geen aanpassingen aan nodig.

5.5.4 Fluentd configuratie

fluent.conf

```
# fluentd/conf/fluent.conf
<source>
  @type forward
  port 24224
</source>

<source>
  @type tail
  read_from_head true
  tag host.log
  path /fluentd/log/testLog/*.log
  <parse>
```

```
    @type regexp
    expression /^\S(?<timestamp>[^]]+)\S\s*(?<level>[^
      ]+)\s*(?<organization>[^ ]+)\s*(?<city>[^ ]+)\s
      *(?<message>.+)/
    time_key timestamp
    time_format %Y-%m-%d %H:%M:%S,%L
  </parse>
</source>

<match host.log>
  @type elasticsearch
  host elasticsearch
  port 9200
  include_timestamp true
  logstash_format true
  logstash_prefix hg-host.log
  type_name fluentd
</match>

<match httpd.access>
  @type copy
  <store>
    @type elasticsearch
    host elasticsearch
    port 9200
    include_timestamp true
    logstash_format true
    logstash_prefix hg-httpd.access
    type_name fluentd
  </store>
  <store>
    @type stdout
  </store>
</match>

<match httpd2.access>
  @type copy
  <store>
    @type elasticsearch
    host elasticsearch
    port 9200
    include_timestamp true
    logstash_format true
    logstash_prefix hg-httpd2.access
    type_name fluentd
  </store>
```

```
  <store>
    @type stdout
  </store>
</match>

<match *.**>
  @type copy
  <store>
    @type elasticsearch
    host elasticsearch
    port 9200
    index_name fluentd
    include_timestamp true
    include_tag_key true
    type_name fluentd
  </store>
  <store>
    @type stdout
  </store>
</match>
```

1. ***<source>*** : Er is een nieuwe source bijgevoegd. Dit is om de log file van het host systeem te kunnen lezen en doorsturen.
2. ***@type tail*** : Aangezien het reeds om een bestaande file gaat is het "tail" type nodig.
3. ***read_from_head true*** : Zorgt ervoor dat de volledige file van in het begin gelezen wordt en niet enkel de nieuwe binnenkomende logs.
4. ***path /fluentd/log/testLog/*.log*** : Het pad naar het eerder gedefinieerde volume (in de Docker-compose.yml).
5. ***<parse>*** : Het gaat hier om een volledig verzonnen log file met een verzonnen format, er zal een op maat gemaakte parsing voorzien moeten worden. Daarom dat deze parser type regexp heeft.
6. ***expression ...*** : Dit is de regular expression. Fluentd gebruikt Ruby regex.
7. ***time_key timestamp %Y-%m-%d %H:%M:%S,%L*** : De naam van de field die de tijd en datum bevat.
8. ***time_format*** : Deze formatting wordt gedaan volgens de "Time#strftime" format.
9. ***<match host.log>*** : Toegevoegd zodanig dat deze logs ook in Elasticsearch terechtkomen.
10. ***logstash_format true*** : Deze format laat toe om de datum toe te voegen aan de index. Dit wordt ook in de andere ***<match>*** blokken aangepast.
11. ***logstash_prefix hg-host.log*** : De naam die voor de datum van de index moet staan. Ook dit wordt in alle andere ***<match>*** blokken aangepast.

Ook is hier te zien dat de ***<filter>*** blok er niet meer tussenstaat. Dit is omdat de extra parsing in de vorige versie geen verbetering was. Daarom wordt die voor deze versie van de proof of concept verwijderd.

De mappenstructuur zou er als volgt moeten uitzien.

- EFK
  - docker-compos.yml
  - fluentd
    - conf
      - fluent.conf
    - Dockerfile
- testLog
  - test.log

5.5.5 Opstarten Docker containers

Opstarten van de virtuele containers blijft opnieuw hetzelfde. Het controleren of de containers correct zijn opgestart blijft ook hetzelfde als in de vorige versies. En het aanmaken van logs voor beide webservers blijft ook het zelfde als in de vorige versie. De logs van de host zullen automatisch worden ingelezen.

5.5.6 Elasticsearch configuratie

Door de nieuwe manier van indexeren kan er een cleanup strategie geïmplementeerd worden. Dit zal alle indexen (en de daarbij horende logs) na een bepaalde tijd verwijderen. De stappen om dit te verwezenlijken worden hier verder beschreven

Index management

Het controleren van de indexes gebeurt op dezelfde manier als in de vorige versie van de proof of concept. Enkel zullen de logs er nu uitzien als: <naam_index>-<jaar>.<maand>.<dag>. Deze dag is de dag dat de logs zijn aangemaakt en niet wanneer deze zijn ingelezen en ontvangen in Elasticsearch.

Index pattern

Een index pattern maken is identiek aan de vorige versie van de proof of concept.

Index template

Een index template laat toe automatisch bepaalde instellingen en toepassingen toe te voegen aan een index.

1. Open het linker pop-out menu.
2. Navigeer daarin naar “Management” > “Stack Management”.
3. In het menu links op de webpagina navigeer naar “Data” > “Index Management”.
4. Onder de titel “Index Management” kan “Index Templates” aangeklikt worden.
5. Klik op de blauwe “+ Create template” knop.
6. Vul hier de naam voor de template in en vul de Index pattern in, gemaakt in de vorige

stappen.
7. Klik op “Next” en herhaal dit tot er een de knop “Create template” verschijnt, druk vervolgens ook op die knop.

Index lifecycle policy

Met een index lifecycle policy kan er ingesteld worden wat er moet gebeuren met een index. Hier kan dan ingesteld worden na hoeveel tijd een bepaalde index (en de daarbij horende logs) verwijderd moeten worden. Het zal hiervoor rekenen vanaf de dag een log ingelezen is in Elasticsearch, niet perse de dag dat de log is aangemaakt.

1. Open het linker pop-out menu.
2. Navigeer daarin naar “Management” > “Stack Management”.
3. In het menu links op de webpagina navigeer naar “Data” > “Index lifecycle policies”.
4. Klik op de blauwe “+ Create policy” knop.
5. Geef de policy een naam onder “Policy name”.
6. Onder “Hot phase” druk op het vuilbak icoontje, nu moet “Delete phase” verschijnen.
7. Nog steeds onder “Hot phase” zet “Use recommended defaults” uit en vervolgens zet “Enable rollover” ook uit.
8. In “Delete phase” kan nu aangeduid worden na welke tijd van creatie een index verwijderd moet worden.
9. Klik op de blauwe “Save policy” knop.
10. Zoek in de lijst naar de nieuwe policy.
11. Klik naast de policy naam op “Action” > “Add policy to index template”.
12. Vul onder “Index template” de naam in van de eerder gemaakte index template.
13. Klik op de blauwe “Add policy” knop.

Het is aangeraden om alle indexes te verwijderen (dit kan gedaan worden onder “Data“ > “Index” management) en de Fluentd virtuele container opnieuw op te starten.

```
$ docker restart efk_fluentd_1
```

Discover

Op de discover pagina van Elasticsearch kan vastgesteld worden dat er meer fields zijn waarop gefilterd kan worden en dat de fields, met info over de virtuele containers, er ook opnieuw tussen staan. Nu kunnen hier ook complexere zoekopdrachten op de logs uitgevoerd worden. Zo kan er gezocht worden naar alle logs die als city field AALST hebben, maar niet HOGENT als organisation field.

5.5.7 Kibana configuratie

Deze configuratie is hetzelfde als in de tweede versie van de proof of concept, daarom zal hier niet opnieuw op ingegaan worden.

5.5.8 Eindsituatie

Alle vooropgestelde einddoelen zijn in deze versie van de proof of concept behaald. Er kunnen logs verzameld worden van verschillende locaties. Deze logs kunnen aan de hand van complexe zoekopdrachten doorzocht worden op één centrale plaats. Ook kunnen de logs gevisualiseerd worden in verschillende en duidelijke grafieken. De machine zal nooit te vol komen te staan met logs aangezien er een cleanup strategie geïmplementeerd is.

Aangezien alle einddoelen bereikt zijn zullen er geen verdere versies van de proof of concept gemaakt worden. Wat uiteraard niet wil zeggen dat er niet meer mogelijk is met de EFK-stack maar dit valt buiten de scope van dit onderzoek en deze opstelling.

6. Conclusie

In dit onderzoek werd een antwoord gegeven op de vraag “Welke log management tool, die logs kan verzamelen en beheren uit virtuele containers, is geschikt om aan te leren aan studenten in de bacheloropleiding toegepaste informatica?”.

Om een antwoord te vinden op deze vraag werd eerst een opsomming gemaakt van elke log management tool en wat ze uniek maakt. Daarna werden alle log management tools onderworpen aan requirements waaraan deze moeten voldoen. Als de tools niet voldeden aan de opgestelde requirements werden ze niet verder overwogen. Uit de resterende log management tools werd dan de meest geschikte tool gekozen. Deze log management tool werd dan uiteindelijk uitgewerkt in een proof of concept waar deze aan enkele einddoelen moet voldoen.

Uit de vergelijking was duidelijk dat er vier log management tools in aanmerking kwamen om als nieuwe leerstof op te nemen. Deze vier tools waren: ELK-stack, EFK-stack, Grafana Loki en Graylog. Uiteindelijk is voor de EFK-stack gekozen, omdat Fluentd, één van de services waar de EFK-stack uit bestaat, beter is geïntegreerd in Docker (de software die gebruikt wordt voor het maken van de virtuele containers). Ook is Fluentd flexibeler dan de andere log management tools, Fluentd kan van meer log management tools deel uitmaken dan enkel de EFK-stack.

Voor deze log management tool werd uitgewerkt zijn er einddoelen gedefinieerd. Na het maken van de proof of concept werd de bekomen opstelling altijd getoetst aan deze einddoelen. Als een proof of concept niet voldeed aan deze einddoelen werd er een nieuwe versie gemaakt van deze proof of concept. Er zijn drie versies gemaakt van de proof of concept, de ene versie trachtte steeds een verbetering te zijn in vergelijking met de andere. De laatste versie voldeed uiteindelijk aan alle voorop gestelde einddoelen.

Ook kan deze proof of concept als startpunt dienen voor een cursus die over log management zou gaan.

A. Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

A.1 Introductie

Het uiteindelijke hoofddoel van dit onderzoek is om het meest geschikte logging-tool of tools te vinden voor Docker en Kubernetes om aan studenten Toegepaste Informatica Systeem- en Netwerkbeheer te kunnen aanleren in een nieuw curriculum.

Eerst moet achterhaald worden wat de limieten (ook wel requirements genoemd) zijn die zich voordoen wanneer er iets moet worden aangeleerd aan studenten. Er moet bijvoorbeeld rekening gehouden worden met de hardware die studenten ter beschikking hebben. Dit is nodig om tijdens de literatuurstudie alle correcte informatie te hebben om een juiste vergelijking te kunnen maken tussen de verschillende tools die gebruikt worden voor het loggen van systemen en applicaties in de context van containervirtualisatie, oftewel voor Docker en Kubernetes (Rad e.a., 2017) (Burns e.a., 2019). Eens er een keuze is gemaakt, kan deze worden uitgewerkt in een proof of concept. De proof of concept test of de literatuurstudie correct was of mogelijk is, indien de literatuurstudie geen duidelijkheid schept over de meest geschikte tool, kan de proof of concept duidelijkheid scheppen over welke tool het beste is.

A.2 State-of-the-art

Wanneer alles correct verloopt zal het opvolgen van systeem- en applicatielogs in de bedrijfswereld nooit de hoogste prioriteit krijgen. Want wanneer niets misloopt heeft het opvolgen van deze logs een lage return on investment (ROI) (Friedlob en Plewa Jr (1996)). Tot er iets misloopt met het systeem of de applicatie. Daarom is het belangrijk om studenten die in deze bedrijfswereld terechtkomen reeds op te leiden in logging.

Er zijn veel artikels en onderzoeken online te vinden die verschillende logging-tools vergelijken. Een voorbeeld hiervan is het onderzoek van Gheorghe (g.d.). Deze vergelijkingen hebben echter geen conclusie zoals we uit dit onderzoek verwachten aangezien hier het beste logging-tool gezocht wordt voor Docker en Kubernetes. Dit onderzoek onderscheidt zich ook van andere onderzoeken door de unieke requirements. Deze artikels zijn echter een goed vertrekpunt, aangezien er hierdoor al veel logging-tools gevonden kunnen worden. Een deel hiervan zal niet in aanmerking komen als best passende logging-tool, maar dit kan enkel bepaald worden na aftoetsing met de requirements.

Ook kunnen er artikels gevonden worden die algemene monitoring tools vergelijken zoals deze van Sissons (g.d.). Hier is het probleem dat niet alle monitoring tools geschikt zijn voor het opvolgen van systeem- en applicatielogs. Voor dit onderzoek worden die artikels dus niet opgenomen.

A.3 Methodologie

Het onderzoek zal kunnen opgedeeld worden in vier grote delen.

Eerst moet onderzocht worden met welke requirements er rekening moet worden gehouden en welke de belangrijkste zijn. Het achterhalen van de requirements kan gebeuren aan de hand van interviews met belanghebbenden (co-promotor en andere lectoren). De bevonden requirements kunnen dan, eventueel in samenspraak met de belanghebbenden, geanalyseerd worden met de MoSCoW priotisation technique (Korolev, g.d.). Dit zal nodig zijn om te achterhalen met welke requirements er het meest rekening moet worden gehouden. In de literatuurstudie moet dan onderzocht worden welke logging tools geschikt zijn voor het gebruik in de context van containervirtualisatie. Alsook waarin deze logging-tools verschillen van traditionele logging. Ook zal de literatuurstudie eventuele uitdagingen onthullen die gepaard gaan met het implementeren van logging-tools in een containervirtualisatie context.

Tijdens het tweede deel van het onderzoek zullen de producten gevonden in de literatuurstudie afgetoetst worden aan de requirements met als doel het beste product vinden.

Het derde deel bestaat dan uit de proof of concept. Dit zal de test zijn of de logging-tools wel echt voldoen aan de requirements en of het mogelijk is deze op te zetten.

Het vierde deel zal dan een besluit zijn. Is de proof of concept gelukt of zijn er tekortkomingen? Is er een logging-tool voor Docker en Kubernetes die voldoet aan alle

requirements?

A.4 Verwachte resultaten

Een grootschalige toepassing van een logging tool zal niet mogelijk zijn door de beperkingen van de hardware van studenten. Dit wil echter niet zeggen dat het onmogelijk zal zijn om het onderwerp aan te leren aan de studenten. De gemiddelde laptops zullen een kleinschalige opstelling volledig aankunnen en het installeren en gebruiken van de logging tool zou geen probleem mogen vormen.

De literatuurstudie zal echter minder duidelijk zijn. Het gebruiken van een ELK stack zal complexer zijn omdat er drie componenten moeten geïnstalleerd worden maar deze wordt wel het meest gebruikt in de bedrijfswereld. Cloud hosted tools zullen waarschijnlijk ook geen optie zijn aangezien deze nog een grotere complexiteit hebben en deze meestal gepaard gaan met extra kosten, wat moet worden vermeden.

Ook de proof of concept zal succesvol kunnen gemaakt worden, indien de literatuurstudie correct verlopen is. Deze zal tevens kunnen dienen als een handleiding voor lectoren om een cursus uit op te stellen.

A.5 Verwachte conclusies

Er wordt verwacht dat de ELK stack, ondanks een grotere complexiteit, als meest geschikte tool zal eindigen. Er zal geen probleem zijn om deze tool te installeren en te laten draaien op de laptops van studenten. Tevens zijn er geen extra kosten aan verbonden en is het een lokaal gehoste tool. Ook is de ELK stack vaak de enige tool die in artikels vermeld wordt voor het analyseren van de logfiles en lijkt het erop dat deze ook het meest gebruikt wordt.

Het monitoren van logfiles is een interessant en uitgebreid onderdeel in de ICT die op dit moment geen deel is van het curriculum Toegepaste Informatica. Na dit onderzoek zal duidelijk zijn dat het een meerwaarde is aan de studierichting om dit onderdeel er wel aan toe te voegen.

B. Volledige log file

Deze bijlage bevat de volledige, zelf verzonnen en gemaakte log file die gebruikt wordt in hoofdstuk 5.5. De parsing in het fluentd.conf bestand zal alleen maar werken op logs met deze format.

B.1 log file

```
[2021-05-18 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-18 11:22:10,125]  INFO HOGENT AALST This is
   not a test log
[2021-05-18 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 18:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
   not a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
   a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
   a test log
```

```
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
   a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   not a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 18:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is a
    test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is a
    test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is a
    test log
[2021-05-19 20:12:10,125]  WARN HOGENT GENT This is a
    test log
[2021-05-19 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-19 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-19 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 18:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is a
    test log
[2021-05-20 09:45:33,125]  WARN HOGENT GENT This is a
    test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is a
    test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is a
    test log
[2021-05-20 09:45:33,125]  ERROR HOGENT GENT This is
   a test log
[2021-05-20 09:45:33,125]  ERROR HOGENT AALST This is
    a test log
```

```
[2021-05-20 09:45:33,125]  ERROR HOGENT GENT This is
   not a test log
[2021-05-20 09:45:33,125]  ERROR HOGENT GENT This is
   a test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is
   not a test log
[2021-05-20 11:22:10,125]  INFO HOGENT - This is a
   test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 18:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is
   not a test log
[2021-05-20 20:12:10,125]  WARN HOGENT - This is a
   test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is a
    test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is a
    test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is
   not a test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 18:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
   a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
   a test log
[2021-05-20 20:12:10,125]  WARN HOGENT - This is a
   test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
   a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
```

```
[2021-05-20 11:22:10,125]  INFO - AALST This is a
   test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 18:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 20:12:10,125]  WARN HOGENT - This is a
   test log
[2021-05-20 09:45:33,125]  WARN HOGENT AALST This is
   a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
   a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
   a test log
[2021-05-20 09:45:33,125]  ERROR HOGENT AALST This is
    a test log
[2021-05-20 09:45:33,125]  ERROR HOGENT AALST This is
    a test log
[2021-05-20 09:45:33,125]  ERROR HOGENT AALST This is
    not a test log
[2021-05-17 09:45:33,125]  ERROR HOGENT AALST This is
    a test log
[2021-05-18 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-18 11:22:10,125]  INFO HOGENT AALST This is
   not a test log
[2021-05-18 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 18:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 11:22:10,125]  INFO - AALST This is a
   test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
   not a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
   a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
   a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
   a test log
```

```
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   not a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 18:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is a
    test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is a
    test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is a
    test log
[2021-05-19 20:12:10,125]  WARN HOGENT GENT This is a
    test log
[2021-05-19 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-19 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-19 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 18:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
    test log
[2021-05-20 20:12:10,125]  WARN - GENT This is a test
    log
[2021-05-20 09:45:33,125]  WARN HOGENT GENT This is a
    test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is a
    test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is a
    test log
[2021-05-20 09:45:33,125]  ERROR HOGENT GENT This is
   a test log
[2021-05-20 09:45:33,125]  ERROR HOGENT AALST This is
    a test log
[2021-05-20 09:45:33,125]  ERROR - - This is not a
   test log
```

```
[2021-05-20 09:45:33,125]  ERROR HOGENT GENT This is
  a test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
   test log
[2021-05-20 11:22:10,125]  INFO HOGENT - This is not
  a test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
   test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
   test log
[2021-05-20 18:22:10,125]  INFO HOGENT GENT This is a
   test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
   test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is
  not a test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is a
   test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is a
   test log
[2021-05-20 20:12:10,125]  WARN HOGENT GENT This is a
   test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
   test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
   test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is
  not a test log
[2021-05-20 11:22:10,125]  INFO HOGENT GENT This is a
   test log
[2021-05-20 18:22:10,125]  INFO HOGENT AALST This is
  a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
  a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
  a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
  a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
  a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
  a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
  a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
  a test log
```

```
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 18:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 11:22:10,125]  INFO HOGENT AALST This is
   a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
   a test log
[2021-05-20 09:45:33,125]  WARN HOGENT AALST This is
   a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
   a test log
[2021-05-20 20:12:10,125]  WARN HOGENT AALST This is
   a test log
[2021-05-20 09:45:33,125]  ERROR HOGENT AALST This is
    a test log
[2021-05-20 09:45:33,125]  ERROR HOGENT AALST This is
    a test log
[2021-05-20 09:45:33,125]  ERROR HOGENT AALST This is
    not a test log
[2021-05-17 09:45:33,125]  ERROR HOGENT AALST This is
    a test log
```

Bibliografie

Beaurain, J. (2019). *Design en implementatie van een container-based deployment systeem voor drones.* (masterscriptie). UGent. Faculteit Ingenieurswetenschappen en Architectuur.

Bergenholtz, H. & Johnsen, M. (2005). Log files as a tool for improving internet dictionaries. *HERMES-Journal of Language and Communication in Business*, (34), 117–141.

Burns, B., Beda, J. & Hightower, K. (2019). *Kubernetes: up and running: dive into the future of infrastructure*. O'Reilly Media.

Chamberlain, D. (2018). *Containers vs. Virtual Machines (VMs): What's the Difference?* https://blog.netapp.com/blogs/containers-vs-vms/ (accessed: 9.4.2021)

Compose, D. (g.d.). *Overview of Docker Compose*. https://docs.docker.com/compose/ (accessed: 3.5.2021)

Datadog. (g.d.). *Modern Log Management & Analytics*. https://www.datadoghq.com/product/log-management/ (accessed: 7.5.2021)

Datadog. (2018). *8 surprising facts about real Docker adoption*. https://www.datadoghq.com/docker-adoption/#6 (accessed: 7.5.2021)

Drott, M. C. (1998). Using web server logs to improve site design. *Proceedings of the 16th annual international conference on Computer documentation*, 43–50.

Elastic. (g.d.). *The heart of the free and open Elastic Stack*. https://www.elastic.co/elasticsearch/ (accessed: 7.5.2021)

Fluentd. (g.d.). *What is Fluentd?* https://www.fluentd.org/architecture (accessed: 7.5.2021)

Friedlob, G. T. & Plewa Jr, F. J. (1996). *Understanding return on investment*. John Wiley & Sons.

Gheorghe, R. (g.d.). *20+ Best Log Management Tools for Monitoring, Analytics & More: Pros & Cons Comparison [2021]*. https://sematext.com/blog/best-log-management-tools/#toc-19-syslog-ng-18 (accessed: 10.2.2021)

GoAccess. (g.d.). *What is it?* https://goaccess.io/ (accessed: 7.5.2021)

Grafana. (g.d.). *Grafana Loki*. https://grafana.com/oss/loki/ (accessed: 7.5.2021)

Graylog. (g.d.). *Graylog enterprise*. https://www.graylog.org/products/enterprise (accessed: 7.5.2021)

HOGENT. (g.d.). *Op zoek naar een nieuwe laptop voor je opleiding?* https://www.hogent.be/student/een-vlotte-start/laptopaanbod/ (accessed: 13.5.2021)

Humio. (g.d.). *Streaming observability*. https://www.humio.com/log-management#features (accessed: 7.5.2021)

Kaltschmidt, D. (2018). *KubeCon schedule*. https://kccna18.sched.com/event/GrXC/on-the-oss-path-to-full-observability-with-grafana-david-kaltschmidt-grafana-labs (accessed: 7.5.2021)

Korolev, S. (g.d.). *MoSCoW Method: How to Make the Best of Prioritization*. https://railsware.com/blog/moscow-prioritization/ (accessed: 10.2.2021)

Lashawn, A. (2021). *Comparing Top Container Software Options for 2021*. https://scoutapm.com/blog/container-service-tools (accessed: 30.4.2021)

Logbeats. (g.d.). *Lightweight data shippers*. https://www.elastic.co/beats/ (accessed: 8.5.2021)

LogDNA. (g.d.). *Logging is shifting left*. https://www.logdna.com/product (accessed: 7.5.2021)

Loggly. (g.d.). *Visualize, Analyze, Inspect, and Solve.* https://www.loggly.com/product/ (accessed: 8.5.2021)

LOGIQ. (g.d.). *What is LOGIQ?* https://logiq.ai/what-is-logiq/ (accessed: 8.5.2021)

Logstash. (g.d.). *Centralize, transform & stash your data*. https://www.elastic.co/logstash (accessed: 8.5.2021)

Logz.io. (g.d.). *LOG MANAGEMENT*. https://logz.io/platform/log-management/ (accessed: 9.5.2021)

Promtail. (g.d.). *Promtail*. https://grafana.com/docs/loki/latest/clients/promtail/ (accessed: 7.5.2021)

Rad, B. B., Bhatti, H. J. & Ahmadi, M. (2017). An introduction to docker and analysis of its performance. *International Journal of Computer Science and Network Security (IJCSNS)*, *17*(3), 228.

Scalyr. (g.d.). *LOG MANAGEMENT Blazing-fast log management for engineering and operations teams*. https://www.scalyr.com/product/ (accessed: 9.5.2021)

Sematext. (g.d.). *Cloud Log Management & Analysis*. https://sematext.com/logsene/ (accessed: 9.5.2021)

Sissons, G. (g.d.). *Comparing 10 Docker Container Monitoring Solutions for Rancher*. https://rancher.com/comparing-10-container-monitoring-solutions-rancher/ (accessed: 18.12.2020)

Sloman, A. & Chrisley, R. (2003). Virtual machines and consciousness. *Journal of consciousness studies*, *10*(4-5), 133–172.

softwaretestinghelp. (2021). *Top 10 Best Container Software In 2021*. https://www.softwaretestinghelp.com/container-software/ (accessed: 30.4.2021)

Splunk. (g.d.). *Splunk Log Observer*. https://www.splunk.com/en_us/software/log-observer.html (accessed: 9.5.2021)

StackOverflow. (2019). *Developer Survey Results 2019*. https://insights.stackoverflow.com/survey/2019 (accessed: 2.5.2021)

Sumologic. (g.d.). *Faster monitoring and troubleshooting*. https://www.sumologic.com/ (accessed: 9.5.2021)

sumologic. (g.d.). *DevOps and Security Glossary Terms*. https://www.sumologic.com/glossary/log-file/#:~:text=A%20log%20file%20is%20a,application%2C%20server%20or%20another%20device. (accessed: 3.5.2021)

Tanenbaum, A. S. & Bos, H. (2015). *Modern operating systems*. Pearson.

Wikipedia. (g.d.). *Log file*. https://en.wikipedia.org/wiki/Log_file#:~:text=In%20computing%2C%20a%20log%20file,to%20a%20single%20log%20file. (accessed: 3.5.2021)

Yegulalp, S. (2019). *What is Docker? The spark for the container revolution*. https://www.infoworld.com/article/3204171/what-is-docker-the-spark-for-the-container-revolution.html (accessed: 1.5.2021)