Odisee
DE CO-HOGESCHOOL

Bachelorproef

Studiegebied Industriële Wetenschappen en Technologie

Brecht Bekaert
Professionele Bachelor Elektronica - ICT
Afstudeerrichting ICT
Academiejaar 2020/2021

Implementatie van operationale tooling voor een kubernetes cluster in AWS

Ordina Belgium
Blarenberglaan 3 bus B
2800 Mechelen
België

Abstract

B. Bekaert

Het doel van deze bachelorproef is onderzoeken welke operationele tooling nodig is om een Kubernetes cluster operationeel te houden in AWS. Met operationele tooling wordt security, monitoring, logging en automatisatie tools bedoeld. Voor elke categorie van tools wordt de AWS-native tool en enkele third-party tools onderzocht. Voor elke tool wordt gekeken naar de configuratie, functionaliteiten, voor- en nadelen en de prijs. De conclusie is dat de AWS-native tools niet voldoende zijn om een Kubernetes cluster operationeel te houden. Voor de security tool is Prisma Cloud gekozen en voor de monitoring tool en logging tool is Datadog gekozen. Automatisatie is gerealiseerd met Terraform.

Op vraag van Ordina werd Azure DevOps gebruikt voor het automatiseren van de tools.

Trefwoorden: AWS, Kubernetes, Operationele tooling, Azure DevOps, Automation

Voorwoord

Deze bachelorproef vormt de afsluiter van mijn opleiding ‘Professionele Bachelor Elektronica-ICT’ met als afstudeerrichting ICT aan de Odisee Hogeschool te Gent.

Om mijn bachelorproef tot stand te brengen heb ik stagegelopen bij Ordina Belgium.

Als eerste wil ik graag mijn externe stagementor Geert Clissen, infrastructuur architect binnen Ordina Belgium bedanken. Ik kreeg een uitdagende opdracht waarbij ik zelf het plan van aanpak mocht uitwerken en dit op een zelfstandige manier uitvoeren. Op deze manier kreeg ik de kans om veel nieuwe zaken te ontdekken en uit te zoeken. Hierdoor kon ik veel kennis vergaren. Tijdens de stage kwamen we wekelijks digitaal samen en kreeg ik hulp, tips en bijsturing indien nodig. Geert stond altijd klaar als ik een probleem had.

Als tweede wil ik graag mijn interne stagementor Serge Fabre, Docent van de opleiding Elektronica-ICT bedanken. Serge heeft mij van begin tot einde begeleid om de stage en bachelorproef tot een goed einde te brengen. We kwamen tweewekelijks digitaal samen om de voortgang van de stage en bachelorproef te bespreken. Bij vragen of problemen kreeg ik ook direct hulp.

Als derde wil ik graag mijn ouders bedanken voor de steun tijdens deze periode. Ze stonden steeds klaar om mij te helpen.

Als laatste wens ik graag Geert Clissen, Serge Fabre, mijn ouders en Schauni Van De Velde te bedanken om de tijd te nemen om mijn bachelorproef volledig na te lezen.

Inhoudsopgave

Abstract 3
Voorwoord 4
Inhoudsopgave 5
Lijst met gebruikte figuren 6
Lijst met gebruikte codefragmenten 7
Lijst met gebruikte afkortingen 8
Voorstelling stagebedrijf 9
Inleiding 11
1 Amazon Web Services 12
1.1 Amazon EKS *12*
1.2 Amazon EC2 ..... *13*
1.3 Amazon S3 ..... *13*
2 Kubernetes 14
2.1 Concept *14*
2.2 Architectuur *14*
2.3 Werking *15*
2.4 Helm *16*
3 Marktonderzoek tools 17
4 Security Tools 27
4.1 AWS GuardDuty *27*
4.2 Qualys *29*
4.3 Prisma Cloud *32*
5 Monitoring Tools 35
5.1 AWS Cloudwatch *35*
5.2 Datadog *38*
5.3 New Relic *41*
6 Logging Tools 43
6.1 AWS Cloudwatch *43*
6.2 ELK Stack *45*
6.3 Datadog *46*
7 Automatisatie Tools 49
7.1 Cloudformation *49*
7.2 Terraform *49*
8 Toolselectie 51
9 Automatisatie van de cloud omgeving 53
9.1 Schema cloud omgeving *53*
9.2 Uitrol van de Kubernetes cluster *53*
9.3 Integratie van de tools *56*
9.4 Azure DevOps *57*
9.5 Prisma Cloud *60*
9.6 Datadog *61*
Besluit 62
Bibliografie 63
Bijlagen 66

Lijst met gebruikte figuren

Figuur 1: Werking Amazon EKS....13
Figuur 2: Kubernetes architectuur [15]....16
Figuur 3: Werking van AWS GuardDuty [18]....18
Figuur 4: Werking van AWS Cloudwatch. [24]....21
Figuur 5: Werking van Elastic Stack [29]....23
Figuur 6: Werking van Cloudformation....24
Figuur 7: Werking van Terraform....25
Figuur 8: GuardDuty: Lijst met meldingen....27
Figuur 9: GuardDuty: Details van een melding....28
Figuur 10: Qualys: Lijst met kwetsbaarheden in AWS-omgeving....29
Figuur 11: Qualys: Detailweergave van een kwetsbaarheid....30
Figuur 12: Qualys: Stappenplan om kwetsbaarheid op te lossen....30
Figuur 13: Qualys: Overzicht kwetsbaarheden van image....31
Figuur 14: Qualys: Details van een kwetsbaarheid....31
Figuur 15: Prisma Cloud: Compliance dashboard....33
Figuur 16: Prisma Cloud: Visuele weergaven van namespaces en containers....34
Figuur 17: Prisma Cloud: Details van een image....34
Figuur 18: AWS Cloudwatch: Dashboard van EC2-instance....35
Figuur 19: AWS Cloudwatch: Visuele weergave van de cluster....36
Figuur 20: AWS Cloudwatch: Lijst van resources van de cluster....36
Figuur 21: AWS Cloudwatch: Details van namespace ‘default’....37
Figuur 22: Datadog: Events pagina....38
Figuur 23: Datadog: Dashboard Kubernetes: Overview....39
Figuur 24: Datadog: Metrics van een container....39
Figuur 25: Datadog: Details van een monitor....40
Figuur 26: New Relic: Kubernetes dashboard....41
Figuur 27: New Relic: Aanmaken van een nieuw alarm....42
Figuur 28: AWS Cloudwatch: Voorbeeld van log groups....43
Figuur 29: AWS Cloudwatch: Lijst van containers....44
Figuur 30: AWS Cloudwatch: Logs van specifieke container....44
Figuur 31: ELK Stack: Lijst met logs....45
Figuur 32: ELK Stack: Patronen in logs....46
Figuur 33: Datadog: Lijst van logs....47
Figuur 34: Datadog: Details van een log....47
Figuur 35: Datadog: Patronen in logs....48
Figuur 36: Terraform: Voorbeeld module....50
Figuur 37: Schema cloud omgeving....53
Figuur 38: Azure DevOps: Deploy Kubernetes on AWS - Pipeline....57
Figuur 39: Azure DevOps: Deploy tools on Kubernetes - Stages....58
Figuur 40: Azure DevOps: Deploy tools on Kubernetes – Pipeline....58
Figuur 41: Prisma Cloud: Lijst met vulnerabilities en gedrag van de tool....60
Figuur 42: Datadog: Monitor in alarm....61

Lijst met gebruikte codefragmenten

Codefragment 1: Cloudformation: Ekctl yaml file 49
Codefragment 2: Terraform: backend.tf 53
Codefragment 3: Terraform: var.tf 54
Codefragment 4: Terraform: provider.tf 54
Codefragment 5: Terraform: modules.tf 55

Lijst met gebruikte afkortingen

AWS	: Amazon Web Services
SLA	: Service Level Agreement
EKS	: Elastic Kubernetes Service
EC2	: Elastic Compute Cloud
S3	: Simple Storage Service
MDR	: Managed Detection & Response
VPC	: Virtual Private Cloud
CIDR	: Classless Inter-Domain Routing
EBS	: Elastic Block Storage

Voorstelling stagebedrijf

Over Ordina

Ordina is een lokale, onafhankelijke IT-dienstverlener in de Benelux. Ordina heeft meer dan 2650 medewerkers en is actief op 3 markten, namelijk financiële dienstverlening, overheid en industrie. Ze volgen de nieuwste technologieën op de voet en groeien mee met de markt.

Het hoofdkantoor is gevestigd in Nieuwegein, Nederland en is opgericht in 1973. Voor België is het hoofdkantoor gelegen in Mechelen. Andere kantoren in België zijn gelegen in Gent, Lummen, Namen.

De consultants van Ordina helpen klanten bij het creëren van innovatieve business- en IT-toepassingen. [1]

Visie

Ordina heeft ook een visie: '**Ahead of change'**.

Het tempo van de digitalisering ligt hoog. Wat vandaag wordt bedacht, heeft morgen al impact.

Voor Ordina is IT de oplossing voor uitdagingen waar bedrijven en organisaties mee geconfronteerd worden.

- IT is de drijvende kracht voor innovatie
- IT biedt een antwoord op grote maatschappelijke vraagstukken.

Mensen moeten vertrouwen in goedwerkende en duurzame oplossingen. Op die manier helpt IT mensen om mee te zijn met de snel veranderde wereld.

Ordina helpt de klanten om verandering voor te blijven zodat ze voorbereid zijn op de uitdagingen van de toekomst. [2]

Kernwaarden

De kernwaarden van Ordina sluiten nauw aan bij hun strategie. Kernwaarden zijn de identiteit van de organisatie.

Ordina heeft 3 kernwaarden [3]:

- **We discover**: Vooroplopen in hun vak om klanten proactief te helpen om 'Ahead of change' te blijven.
- **We connect**: Samenwerken in teams en verantwoordelijkheid nemen voor het geleverde resultaat.
- **We accelerate**: De klant positief verrassen door een ambitieuze oplossing te bieden die het verschil kan maken in de toekomst.

Stageomgeving

De stage en de bachelorproef is uitgevoerd bij de dienst 'Business Platform Services'. In deze dienst zijn er 4 afdelingen.

- **Digital workplace** [4]: Ontwikkelen van een dynamische digitale werkplekken om de prestaties van medewerkers optimaal te houden
- **Robotic process automation** [5]**:** Automatiseren van processen waardoor het bedrijf een efficiëntie- en procesverbetering krijgt.
- **SAP Services** [6]**:** Ontwikkelen en beheren van SAP-applicaties.
- **Hybrid cloud & IT operations** [7]**:** Opzetten en beheren van Cloud omgevingen van de klant via verschillende partners.

Om precies te zijn is de stage en bachelorproef uitgevoerd bij de laatste afdeling (Hybrid cloud & IT operations)

Inleiding

In deze bachelorproef is onderzoek gedaan naar de benodigde tooling om een Kubernetes cluster optimaal operationeel te houden in AWS.

Kubernetes wordt de laatste jaren meer en meer gebruikt en dat brengt extra uitdagingen met zich mee. Hierdoor moet er tooling voorzien worden zodat een Kubernetes cluster op een eenvoudige manier operationeel gehouden kan worden. De tooling die onderzocht wordt kan onderverdeeld worden is 4 categorieën:

- Security
- Monitoring
- Logging
- Automatisatie

In bovenstaande categorieën wordt de AWS Tool en enkele third-party oplossingen onderzocht.

Voor elk van deze tools zullen volgende aspecten onderzocht worden

- Configuratie
- Functionaliteiten
- Voor- en nadelen
- Prijs

In het eerste hoofdstuk zal er kort gekeken worden naar Amazon Web Services. Hierbij worden de verschillende mogelijkheden van AWS besproken. In een tweede hoofdstuk zal Kubernetes toegelicht worden. In een derde hoofdstuk wordt het marktonderzoek besproken. Dit marktonderzoek omvat de 4 tooling categorieën. In hoofdstuk 4 tot 7 worden respectievelijk de security, monitoring, logging en automatisatietools uitgetest en besproken. In hoofdstuk 8 zal bij elke categorie een tool worden uitgekozen die later gebruikt zal worden in de Cloud omgeving. In hoofdstuk 9 zal de volledige infrastructuur automatisch opgezet worden met van elke categorie de uitgekozen tool.

De eindsituatie van de bachelorproef zal een Kubernetes cluster zijn met de nodige operationele tooling uitgerold op een AWS-omgeving.

1 Amazon Web Services

AWS is een cloud platform van een Amerikaanse e-commercegigant Amazon. Het is sinds 2010 jaarlijks verkozen tot marktleider volgens Gartner ’s Magic Quadrant for Cloud Infrastructure as a Service. [8]

In deze bachelorproef zal gebruik gemaakt worden van volgende toepassingen.

- Amazon Elastic Kubernetes Service (AWS EKS)

Met behulp van EKS zal de master node van de Kubernetes cluster opgezet worden.

- Amazon Elastic Compute Cloud (AWS EC2)

Met behulp van EC2 zullen er verschillende worker nodes toegevoegd worden aan de cluster.

- Bestandsopslag (AWS S3)

S3-buckets wordt gebruikt voor storage van applicaties in de Kubernetes cluster.

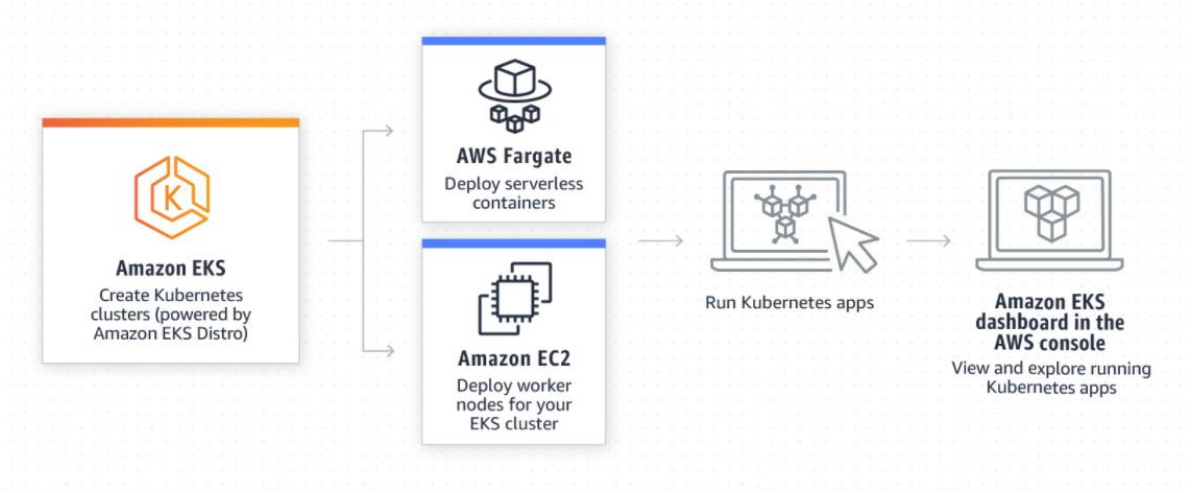
1.1 Amazon EKS

Amazon Elastic Kubernetes Service (EKS) is een beheerde service waarmee er een Kubernetes kan gebruiken zonder zelf alles te installeren en beheren. Met EKS zal de control plane beheerd worden door AWS. Er kan dan zelf bepaald worden of de worker nodes via EC2 of Fargate gedeployed worden.

AWS EKS heeft 3 grote voordelen. Een eerste voordeel is beschikbaarheid. EKS zal de Kubernetes control plane op verschillende availability zones plaatsen zodat er geen downtime mogelijk is. Ook zal het automatisch de ‘unhealthy’ control plane vervangen met een nieuwe control plane. EKS garandeert een 99,95% uptime SLA. Een tweede voordeel is schaalbaarheid. Met EKS managed node groups zal ervoor zorgen dat er nodes toegevoegd worden als de ‘load’ op de Kubernetes cluster toeneemt. Deze node groups kunnen uiteraard beperkt worden zodat de kosten niet te hoog oplopen. Een derde voordeel is security. EKS zal automatisch de laatste security patches toepassen op de cluster control plane en worker nodes in de node groups. [9]

Om Kubernetes applicaties te starten op een cluster moet Amazon EKS eerst worker nodes hebben. Er kan dan gebruik gemaakt worden van worker nodes via EC2. Hierbij wordt er voor de volledige virtuele machine betaald. AWS Fargate is een andere oplossing waarbij er enkel betaald wordt voor de container resources die gebruikt worden. De worker node is dan in beheer van AWS. Fargate wordt niet gebruikt in deze bachelorproef.

De werking van Amazon EKS is te zien op figuur 1.



Figuur 1: Werking Amazon EKS

1.2 Amazon EC2

Amazon Elastic Compute Cloud (EC2) is een webservice dat virtuele machines aanbiedt in de cloud. Met EC2 kunnen er op een eenvoudige manier extra machines aangemaakt worden. Er zijn verschillende soorten processors, ram combinaties, gpu's beschikbaar. Elke toepassing kan gebruik maken van een specifieke EC2 type machine. Amazon EC2 is ook de enige provider die macOS ondersteunt. [10]

Met EC2 kan je een voorgeconfigureerde template kiezen van AWS of zelf een configuratietemplate aanmaken. Daarnaast kan je de verschillende security- en netwerkpolicies instellen op deze instance/instances.

EC2 gebruikt een pay-as-you-go model en er wordt betaald per uur. De prijs per uur is wel afhankelijk van het type machine dat gekozen wordt. Hierdoor zijn er geen verplichtingen bij het gebruik of verwijderen van een machine.

1.3 Amazon S3

Amazon Simple Storage Service (S3) is een object storage service. S3 is industry-leading op vlak van performance, scalability, availability en durability. Het heeft ook verschillende storage classes zodat de prijs altijd laag blijft. [11]

In deze bachelorproef zal S3 gebruikt worden voor:

- Opslaan van Terraform state file.
- Data van applicaties op de Kubernetes cluster

2 Kubernetes

Kubernetes is een open-source container systeem voor het automatiseren van applicaties, schaalbaarheid en het beheer van toepassingen. Het zorgt er ook voor dat de containers geen downtime hebben. Hierdoor is de applicatie (bijna) altijd beschikbaar. [12]

De eerste release van Kubernetes kwam uit op 7 Juni 2014.

2.1 Concept

Kubernetes zorgt ervoor dat toepassingen eenvoudig uitgerold, onderhouden en schaalbaar zijn. Om dit te bekomen worden enkele concepten gebruikt die hieronder kort uitgelegd worden. [13]

Pod: Een pod is een verzameling van één of meer containers die dezelfde opslag en netwerk gebruiken. Pods worden meestal gecreëerd door deployments of sets.

Service: Een service is een manier om een applicatie die draait in een pod beschikbaar te maken naar buiten. Dit kan via een poort of DNS-name. Er kan ook aan load balancing gedaan worden tussen 2 of meerdere pods.

Namespace: Kubernetes ondersteund meerdere virtuele clusters op dezelfde fysieke cluster. Deze virtuele clusters worden namespaces genoemd.

ReplicaSet: Een replicaset zal op elk moment een bepaald aantal replica pods opzetten. Op deze manier wordt er beschikbaarheid van de applicatie verzekerd.

Deployment: Een deployment is een declaratieve update voor pods en replicasets. Er wordt enkel de gewenste toestand beschreven en Kubernetes maakt de veranderingen aan de pods.

DaemonSet: Een daemonset zorgt ervoor dat alle nodes een pod krijgen met deze applicatie. Als er een node toegevoegd wordt zal er automatisch een pod aangemaakt worden. Als er een node verwijderd wordt zal deze pod ook verwijderd worden.

2.2 Architectuur

Een Kubernetes cluster bestaat uit twee onderdelen. Het eerste onderdeel is de control plane. Dit onderdeel beheert de nodes en pods op de verschillende clusters. Een control plane bestaat uit vijf componenten: [14]

Kube-apiserver: De API server is een component dat de Kubernetes control plane beschikbaar maakt naar buiten. Het is de front-end van de Kubernetes control plane. Dit is het entrypoint van de cluster.

Etcd: Etcd is een consistent en highly-available key value opslag voor alle cluster data. Dit is enkel met informatie over de cluster, pods, services, …. In dit component zit geen applicatiedata van een pod zelf.

Kube-scheduler: De kube-scheduler kijkt uit naar nieuwe pods zonder node en zal deze pods uitrollen op een node. Hierbij zal rekening gehouden worden met de resources die deze pod nodig heeft en de resources die nog beschikbaar zijn op de worker nodes.

Kube-controller-manager: Controller manager is het component dat alle verschillende controllers draait. Er zijn 4 type controllers:

- **Node controller:** Detecteert als een pod down gaat.
- **Job controller:** Wacht op jobs om pods te starten.
- **Endpoint controller:** Zorgt voor het verspreiden van de endpoints in de cluster naar alle nodes.
- **Token controller:** Maak een standaardaccount en API-token aan voor nieuwe namespaces.

Deze verschillende controllers hebben elk een apart proces maar om complexiteit te vermijden zijn deze gecompileerd in 1 proces.

Cloud-controller-manager: De cloud controller manager is verantwoordelijk voor specifieke objecten bij de cloud provider. Bij het aanmaken van een load balancer zal deze component een load balancer aanmaken bij de cloud provider.

Het tweede onderdeel van de Kubernetes cluster is een worker node. Een worker node bestaat uit 3 componenten: [14]

Kubelet: Dit component zorgt ervoor dat containers draaien in een pod. Kubelet zal aan de hand van een set specificaties dan kijken als de containers 'healthy' zijn. Kubelet zal geen containers beheren die niet door Kubernetes zijn aangemaakt.

Kube-proxy: Kube-proxy is een netwerk proxy die op elke node zal draaien in de cluster en implementeert het Kubernetes serviceconcept. Dit component zal ook alle netwerkregels bijhouden op de node/nodes voor inkomende en uitgaande verbindingen.

Container runtime: De container runtime is software waarop de containers draaien. Kubernetes ondersteunt verschillende container runtimes zoals docker, containerd, ….

2.3 Werking

Een Kubernetes cluster wordt aangesproken via 'kubectl'. Kubectl is een command line tool en maakt gebruik van een kubeconfig file om zich te authentiseren bij de juiste Kubernetes Master. Als er een commando via kubectl wordt gestuurd zal dit toekomen bij de Kube API-Server.

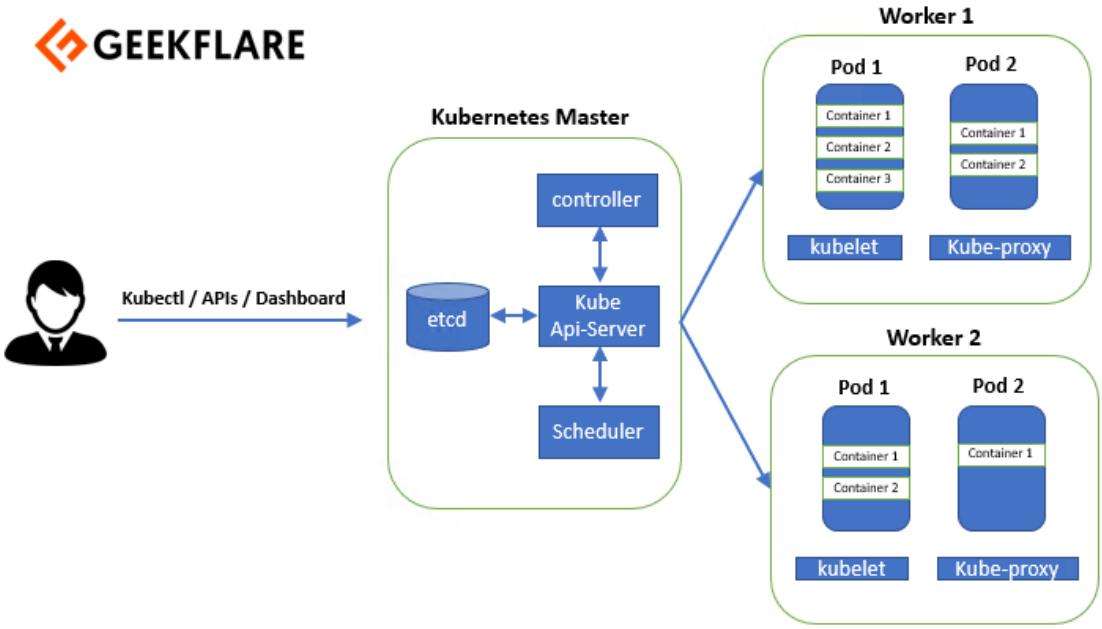
Commando's waarbij er informatie opgevraagd wordt van de cluster worden meestal beantwoord door de controller. Deze controller bezit de informatie over welke pods/containers/services er actief zijn op de cluster.

Als er een applicatie moet gedeployed worden zal er een 'apply' commando uitgevoerd worden. Dit 'apply' commando zal toekomen op de Kube API-Server. De API-server spreekt dan de scheduler aan die de nodige containers en pods zal uitrollen. Ook de cloud controller zal de nodige aanpassingen doen op de cloud indien er een load balancer nodig is. Als de pod uitgerold is op een worker zal kube-proxy ervoor zorgen dat de routering naar de pod werkt voor de volledige cluster.

Nadien zal de controller van de master node continu kijken als de pod niet 'down' gaat. Indien deze down zou gaan zal het bovenstaande proces zich herhalen zodat de pod terug beschikbaar wordt.

Indien de applicatie beschikbaar moet zijn voor de buitenwereld is er een service nodig. Deze service kan ook via een kubectl apply commando toegepast worden op de cluster.

De bovenstaande componenten worden op figuur 2 afgebeeld.



Figuur 2: Kubernetes architectuur [15]

2.4 Helm

Helm is een package manager voor Kubernetes. Met kubectl moeten er verschillende bestanden op de cluster geplaatst worden via een apply commando. Om dit proces eenvoudiger te maken is er Helm.

Helm zorgt ervoor dat verschillende bestanden zoals deployment, daemonset, service, … in 1 pakket zitten. Hierdoor moet er maar 1 commando uitgevoerd worden op de cluster om de volledige applicatie uit te rollen. Updates met helm zijn ook eenvoudig. Wijzigingen aan 1 document zal een volledig nieuwe versie van de applicatie uitbrengen. Helm maakt gebruik van 'helm charts', die makkelijk te delen zijn met een community. Daarnaast heeft Helm ook een rollback functie waardoor er kan teruggekeerd worden naar een oudere versie van de applicatie. [16]

Bepaalde tools in de bachelorproef maken gebruiken van een helm chart.

3 Marktonderzoek tools

De eerste tool van elke categorie is altijd de AWS-oplossing. De andere oplossingen zijn third-party. Bij elke oplossing zal er gekeken worden als bepaalde criteria voldaan zijn. Elke categorie van tools heeft zijn eigen criteria.

Een eerste stap in het marktonderzoek zijn de rapporten van Gartner. Deze rapporten geven een goed beeld wat er momenteel op de markt is en wat de sterke en zwakke punten zijn van elke tool. Met behulp van het Gartner Magic Quadrant is het mogelijk de verschillende categorieën van tools in te delen onder volgende marktpositie. [17]

Leader: Leaders voeren hun visie goed uit en zijn goed voorbereid op de uitdagingen van morgen.

Visionaries: Visionaries begrijpen de markt en hebben een visie voor de uitdagingen van morgen maar voeren deze visie nog niet goed uit.

Niche Players: Niche players focussen goed op 1 specifiek onderdeel maar zijn niet innovatief genoeg en presteren niet beter dan anderen.

Challengers: Challengers voeren hun visie goed uit vandaag maar begrijpen de uitdagingen van morgen niet goed.

Een kleine opmerking bij de rapporten van Gartner is dat de rapporten minder gefocust waren op container tooling waardoor deze tool niet gebruikt kan worden.

Een tweede stap in het marktonderzoek zijn de rapporten van Forrester. Deze rapporten geven net zoals de Gartner rapporten een goed beeld over wat er momenteel op de markt is. Ook hier zijn er 4 groepen die dezelfde marktposities vertegenwoordigen.

Een laatste stap in het marktonderzoek is het bekijken van alternatieve oplossingen die niet vermeld staan in Gartner of Forrester. Door een combinatie van internetbronnen en gebruikersreviews zijn er verschillende nieuwe tools gevonden.

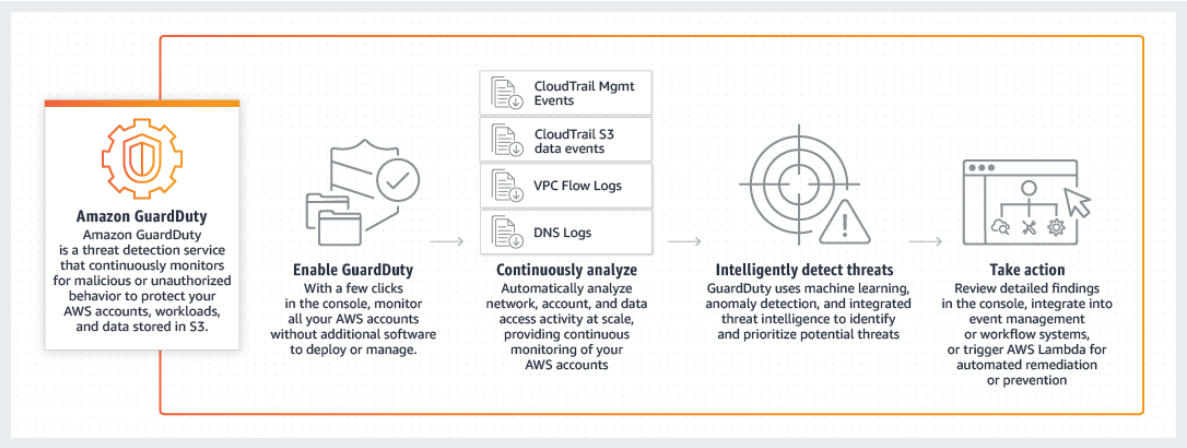
Security Tools

De security tools moeten ervoor zorgen dat de infrastructuur volgens de correcte standaarden wordt uitgerold. Ook moeten de kwetsbaarheden in kaart gebracht worden. Deze tools moeten op een eenvoudige manier de problemen in de infrastructuur kunnen melden aan de verantwoordelijke persoon of team.

- **AWS GuardDuty**

AWS GuardDuty is de security oplossing van AWS. GuardDuty zorgt ervoor dat het volledige AWS-account gemonitord wordt en dat verdachte activiteiten gemeld worden. Met het volledige AWS-account wordt CloudTrial Events, S3, VPC Flow Logs en DNS Logs bedoeld. Deze delen van AWS worden continu geanalyseerd. Het gebruikt machine learning, anomaly detection en hun geïntegreerde threat prevention om mogelijk gevaren te detecteren en te waarschuwen. [18]

Op figuur 3 wordt de werking van GuardDuty aangetoond.



Figuur 3: Werking van AWS GuardDuty [18]

Met GuardDuty kan enkel het AWS-account gemonitord en geanalyseerd worden. Er is geen functionaliteit om per container of per image vulnerabilities te bekijken. Wel zal GuardDuty melden als er een verdachte activiteit heeft plaatsgevonden.

De prijs bij GuardDuty is usage-based. Je betaalt dus enkel wat je verbruikt. De prijzen van GuardDuty zijn verschillend per regio en er is een betere prijs per eenheid naarmate het verbruik stijgt. Hieronder een voorbeeld voor regio Ireland (EU-WEST-1). Er is een trial van 30 dagen beschikbaar.

VPC Flow Log and DNS Log Analysis

First 500 GB / month	$1.10 per GB
Next 2000 GB / month	$0.55 per GB
Next 7500 GB / month	$0.28 per GB
Over 10000 GB / month	$0.17 per GB

- **AlertLogic**

AlertLogic is een bedrijf dat opgericht is in 2002. Het hoofdkantoor situeert zich in Houston, Texas maar heeft andere kantoren in Austin, Cardiff, London, Cali Colombia. AlertLogic heeft een goede reputatie en heeft meer dan 4000 bedrijven die gebruik maken van hun software. Ze zijn ook verkozen tot leider in het Forrester rapport 2020. Tevens hebben ze verschillende awards gewonnen waaronder IT World, Cyber Security Excellence en CRN Channel Chiefs. [19]

AlertLogic MDR werkt op verschillende platformen waaronder publieke cloud, hybrid omgevingen en on-site infrastructuur. AlertLogic heeft ook MDR-oplossingen voor webapplicaties en compliance.

AlertLogic is een SaaS MDR provider met security experts die threats in real-time monitoren. Hun oplossing collecteert verschillende logs van applicaties. Ook hebben ze de feature 'container security'. Met de container security kan AlertLogic cyberattacks in realtime detecteren, logs verzamelen van de containers om inzicht te krijgen in mogelijke threats of compliance problemen. Ze geven proactief meldingen wanneer er een verdachte activiteit gebeurt. [20]

Prijzen zijn niet beschikbaar maar op de website wordt er wel vermeld dat de prijzen ook usage-based zijn. Er is een trial beschikbaar maar enkel als er een contract getekend wordt. Hierdoor kon deze tool niet uitgetest worden.

- **Qualys**

Qualys is opgericht in 1999 als één van de eerste SaaS security bedrijven. Het hoofdkantoor ligt in Foster City, California. Ze hebben verschillende partnerships zoals met Amazon Web Services, Microsoft Azure en Google Cloud Platform. Hun software wordt gebruikt door meer dan 19000 bedrijven over 130 landen. [21]

Qualys heeft een 'Cloud Platform' waarmee je verschillende agents en sensors kan deployen op alle soorten platformen. Er is een mogelijkheid om virtuele machines te beveiligen, containers, webapplicaties, etc. Voordelen van het cloudplatform zijn dat je geen hardware moet kopen of beheren. Het is eenvoudig om uit te breiden. Alles is up-to-date door Qualys en de data wordt beveiligd opgeslagen in databases van Qualys. Er is ook een optie om de data on-premise op te slaan. Het platform analyseert threats en misconfiguraties in realtime en rapporteert deze problemen. Ook kunnen er met de container security verschillende vulnerabilities gezien worden in runtime of in de verschillende images. [22]

De prijs start vanaf $500/ maand voor 1 platform. Voor een volledige infrastructuur wordt er best contact opgenomen met Qualys zodat ze een offerte kunnen opmaken. Er is een trial beschikbaar van 14 dagen.

- **Prisma Cloud**

Palo Alto Networks is een Amerikaans cybersecurity bedrijf met het hoofdkantoor in Santa Clara, Californië. De voorbije jaren heeft Palo Alto verschillende startups overgenomen waaronder Evident.io. Deze startup had zijn focus op infrastructuurbeveiliging en twistlock. Twistlock wordt gebruikt voor container security. Alle functionaliteiten van Palo Alto zitten in 1 product, Prisma Cloud. [23]

Door de verschillende overnames is Prisma Cloud uitgegroeid tot een product met veel verschillende functionaliteiten op vlak van security. Prisma Cloud heeft modules zoals Threat Detection, Data Security, Host Security, Container Security, etc. Bij de optie container security zijn er functionaliteiten zoals

- Vulnerability Management
- Compliance
- Runtime Defense
- Network Visibility
- Incident Response
- CI/CD Security

Prisma Cloud kost $180/ host en er moeten minimum 100 hosts aangekocht worden. Er is een trial beschikbaar voor 30 dagen waarbij alle functionaliteiten uitgetest kunnen worden.

Monitoring Tools

Monitoring Tools moeten ervoor zorgen dat alle mogelijke metrics in kaart gebracht worden. Verder moet er best een mogelijkheid zijn om verschillende containers te monitoren en proactief melden wanneer bepaalde metrics in alarm zouden gaan. Handige features zijn anomaly detection, trends en logs.

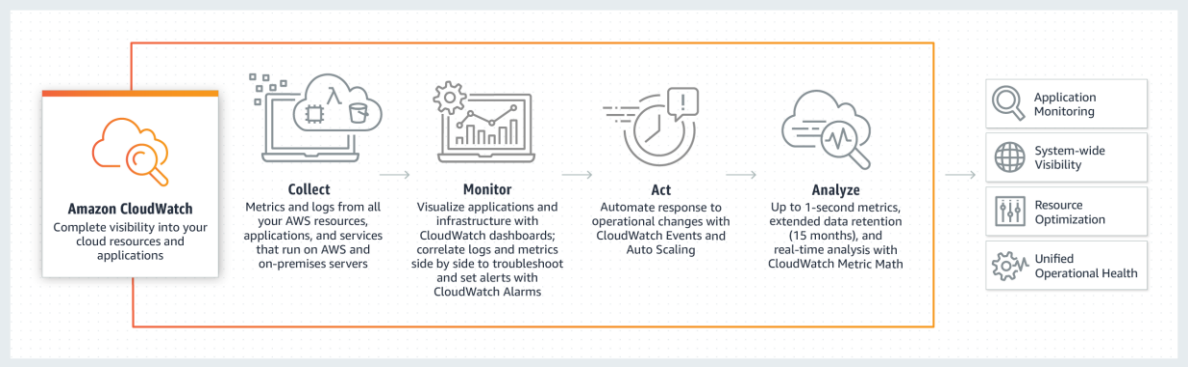
- **AWS Cloudwatch**

Cloudwatch is de AWS-oplossing om aan monitoring te doen. Het heeft de mogelijkheid om verschillende metrics te monitoren. Deze metrics worden weergegeven op verschillende dashboards waarmee vergelijken heel eenvoudig wordt. Ook heeft Cloudwatch de mogelijkheid om logs te verzamelen en te bekijken in combinatie met de metrics.

Met de container monitoring feature wordt er inzicht gegeven in de verschillende metrics van de containers. Voor EKS zullen er specifieke metrics beschikbaar komen na het uitrollen van de AWS Cloudwatch agent. Hiermee kan er specifieker gekeken worden naar de verschillende namespaces en de bijhorende pods.

Omdat Cloudwatch een AWS-tool is, kan er aan auto scaling gedaan worden. Hierdoor kunnen er bij het overschrijden van bepaalde metrics in de Kubernetes cluster extra EC2-instances aangemaakt worden of instances verwijderd worden. [24]

De werking van Cloudwatch is te zien op figuur 4.



Figuur 4: Werking van AWS Cloudwatch. [24]

Cloudwatch werkt ook op een usage-based principe. Dashboards hebben een vaste prijs van $3. Metrics worden betaald per schijf.

Metrics prijzen

First 10,000 metrics	$0.30
Next 240,000 metrics	$0.10
Next 750,000 metrics	$0.05
Over 1,000,000 metrics	$0.02

- **New Relic One**

New Relic One is een bedrijf dat cloudgebaseerde software ontwikkelt om websites en applicaties te monitoren. De hoofdzetel is in San Francisco, Californië. Het platform is ook uitgeroepen als strong performer bij Forrester 2019 en leader volgens Gartner 2020. [25]

New Relic One bestaat uit verschillende onderdelen zoals

- Infrastructure monitoring
- Application Performance Monitoring
- Alerts
- Insights
- Browser
- Synthetics
- Logs
- Mobile

Met de verschillende onderdelen kan er een volledige infrastructuur in kaart gebracht worden. Er is ook een optie om proactief te monitoren waardoor er gewaarschuwd wordt als een metric slecht zou worden.

Een groot voordeel van New Relic One is dat er ook mogelijkheid is om logs te collecteren en te analyseren samen met de verzamelde metrics.

New Relic One heeft een gratis versie maar voor bedrijven zullen de betalende versie moeten aanschaffen. De gratis versie kan je gebruiken tot 100GB/ maand en $0,25 /GB boven de gratis limiet. Voor bedrijven kan er een offerte aangevraagd worden. [26]

- **Datadog**

Datadog is een Software as a Service voor monitoring van cloud applicaties. Het is opgericht in 2011 met de hoofdzetel in New York. Datadog heeft verschillende tevreden klanten waaronder Peloton, Samsung en Nginx. Het platform is ook uitgeroepen tot leader bij Forrester 2019 en Visionair bij Gartner 2020. [27]

Het platform heeft een uitgebreid aantal mogelijkheden. De verschillende mogelijkheden zijn

- Application Performance Monitoring
- Security Monitoring
- Network Monitoring
- Real User Monitoring
- Serverless
- Log Management

Datadog heeft ook een groot pluspunt met zijn log management. Met deze tool kunnen er ook metrics samen met logs bekeken worden. Datadog kan ook proactief monitoren waardoor er een melding gestuurd wordt bij mogelijke problemen.

De prijs van Datadog is $23 per host voor de Enterprise edition van de Infrastructure monitoring. Om logs te verzamelen betaal je $1,70 per 1 miljoen log events/ maand. Afhankelijk van de grootte van de infrastructuur kan dit bedrag oplopen.

Logging Tools

Logging tools moeten ervoor zorgen dat logs van alle mogelijke bronnen verzameld worden en op één plaats beschikbaar zijn. Aanwezigheid van bepaalde features zoals trends, anomaly detection, monitoring zijn zeker een pluspunt.

- **AWS Cloudwatch**

Cloudwatch is de AWS-oplossing voor logging. Deze tool is dezelfde als voor monitoring. Met Cloudwatch kunnen er logs geanalyseerd en gevisualiseerd worden om mogelijke operationele problemen te detecteren.

Cloudwatch heeft zoals hierboven al vermeld container monitoring en kan hierdoor ook logs verzamelen van de verschillende pods/services/applicaties. Deze logs worden allemaal verzameld in de AWS-omgeving. [24]

De werking van AWS Cloudwatch is te zien op figuur 4.

- **ELK Stack (Elastic Stack)**

Elastic Stack is gestart als een open-source project. Het omvat 3 belangrijke tools:

- Elasticsearch: JSON-zoekmachine
- Logstash: Collecteren van Logs.
- Kibana: Visualisatie van Logs

Ondertussen heeft de ELK Stack een betalende optie waardoor een SLA kan gegarandeerd worden.

Elastic Stack heeft verschillende machine learning technieken waarmee problemen in de infrastructuur snel geanalyseerd en opgespoord kunnen worden. De tool heeft ook een grote community en kan snel opgezet worden. [28]

De werking van deze tool is te zien op Figuur 5.

Met logstash zullen de logs gecollecteerd worden. Deze logs zijn beschikbaar voor Elasticsearch en Kibana om aan analyse of visualisatie te doen.

Figuur 5: Werking van Elastic Stack [29]

De prijs is $30/ maand voor Elastic Cloud, de betalende versie van Elastic Stack. Er is een trial aanwezig om de tool uit te proberen.

- **Datadog**

Datadog is een monitoring tool die hierboven is gebruikt. Deze tool zal ook besproken worden als logging tool. Een groot voordeel hierbij is dat er maar 1x betaald moet worden voor een tool met 2 functionaliteiten (monitoring en logging).

Automatisatie Tools

Automatisatie tools kan opgesplitst worden in 2 delen

- Infrastructure as Code
- Azure DevOps

Met Infrastructure as Code kan een volledige infrastructuur opgebouwd worden met enkel code. Hierdoor kan de infrastructuur op een eenvoudige manier opnieuw gebruikt worden. Een ander voordeel is dat niet elke asset in AWS toegevoegd/aangepast moet worden bij een wijziging. De tool zorgt voor alle wijzigingen in de AWS-omgeving.

Belangrijke opmerking bij het gebruik van Infrastructure as a Code. Als de infrastructuur uitgerold is met zo'n tool is het niet de bedoeling om op de Cloud omgeving nog wijzigingen uit te voeren met de hand. Hierdoor kan 'drift' ontstaan. Drift is wanneer de state van de automatisatietool niet meer overeenkomt met de Cloud omgeving.

Infrastructure Management wordt gebruikt om de uitgerolde infrastructuur te beheren. Met dit onderdeel kunnen er verschillende toepassingen automatisch geïnstalleerd en geconfigureerd worden. Op deze manier kan de infrastructuur heel snel operationeel gebracht worden.

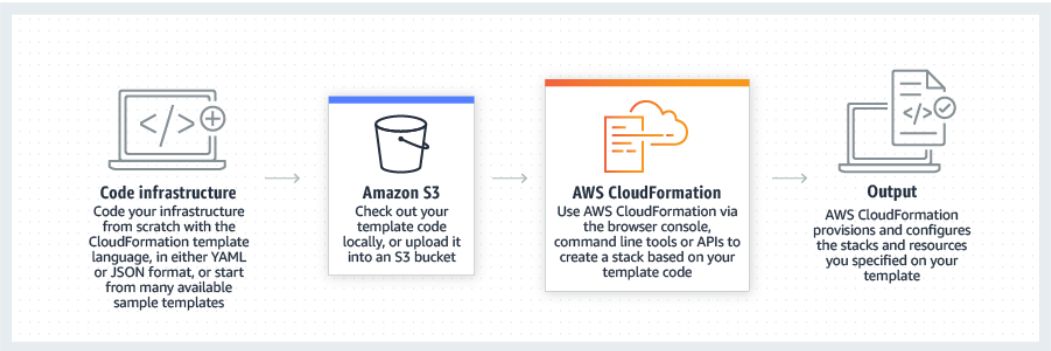
- **Cloudformation**

Cloudformation is de AWS-oplossing om aan Infrastructure as a Code te doen. Cloudformation heeft verschillende templates om direct mee aan de slag te gaan. De templates zijn geschreven in JSON of YAML. Met de AWS Cloud Development Kit is het ook mogelijk om de infrastructuur te schrijven in TypeScript, Python, Java, etc.

Met deze tool kan er in elke AWS-regio een stack uitgerold worden. Ook zal Cloudformation rekening houden met volgtijdelijkheden waardoor alles netjes uitgerold zal worden. De code wordt op een declaratieve manier geschreven. Dat wil zeggen dat de code omschrijft wat er moet uitgerold worden. [30]

De werking van Cloudformation is te zien op figuur 6.

Op de figuur wordt aangegeven dat de template in een S3 bucket wordt geplaatst. Nadien zal Cloudformation deze template inlezen en uitvoeren op het AWS-Platform. Nadien zal de gevraagde configuratie beschikbaar worden.



Figuur 6: Werking van Cloudformation

Cloudformation is gratis voor de eerste 1000 operations per maand. Nadien wordt er $0,0009 betaald per operation.

- **Terraform**

Terraform is een open-source oplossing voor Infrastructure as a Code. Terraform is compatibel met veel cloud platformen zoals AWS, Google Cloud Platform, Microsoft Azure, etc. Het gebruikt ook een eigen taal genaamd HashiCorp Configuration Language. Terraform heeft 3 belangrijke stappen: [31]

- Write: Schrijf de infrastructuur in een declaratieve manier in HCL.
- Plan: Bekijk welke wijzigingen jouw configuratie zal uitvoeren.
- Apply: Maak de aanpassingen op de cloud provider.

De 'state' van de infrastructuur wordt opgeslagen op de harde schijf.

De werking van Terraform is te zien op figuur 7.

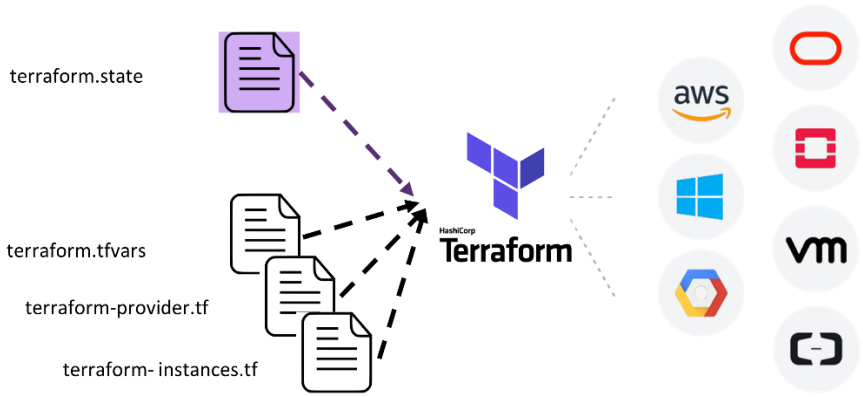
Op de figuur staan er verschillende bestanden. De bestanden hebben volgende betekenis:

- Terraform.state

Dit is de state file van Terraform. Hierin wordt de status van de uitgevoerde elementen opgeslagen. Met deze file weet Terraform wat er op de Cloud omgeving staat en wat nog moet gedaan worden.

- .tf & .tfvars

Deze files zijn de configuratiefiles van Terraform. Ze bevatten variabelen, resources en modules om de nodige infrastructuur uit te rollen op de Cloud omgeving.



Figuur 7: Werking van Terraform

Terraform heeft een grote community met meer dan 450000 commits, meer dan 4000 modules en meer dan 500 providers.

Terraform is volledig gratis om te gebruiken. Er is ook een Enterprise en Business versie waarmee er eenvoudig in team samengewerkt kan worden om de infrastructuur uit te rollen. Voor deze betalende versies wordt best contact opgenomen met HashiCorp. [32]

- **Azure DevOps**

Azure DevOps is een automatisatietool van Microsoft. DevOps is een combinatie van ontwikkeling en bedrijfsactiviteiten. Het is een bundeling van mensen, processen en technologie om doorlopend waarde aan klanten te bieden.

Met Azure DevOps kan er op een eenvoudige manier een geautomatiseerde pipeline gecreëerd worden. Een pipeline is verdeeld in verschillende stages. In deze stages zullen er taken geschreven worden die uitgevoerd moeten worden. Deze taken kunnen scripts, Terraform commando's, …. zijn. [33]

Er wordt gebruik gemaakt van versiebeheer zodat elke versie kan teruggedraaid worden indien er iets faalt.

Azure DevOps is gratis voor 1800 minuten per maand. Nadien moet er extra betaald worden. Er wordt na 5 gebruikers $6/ gebruiker betaald.

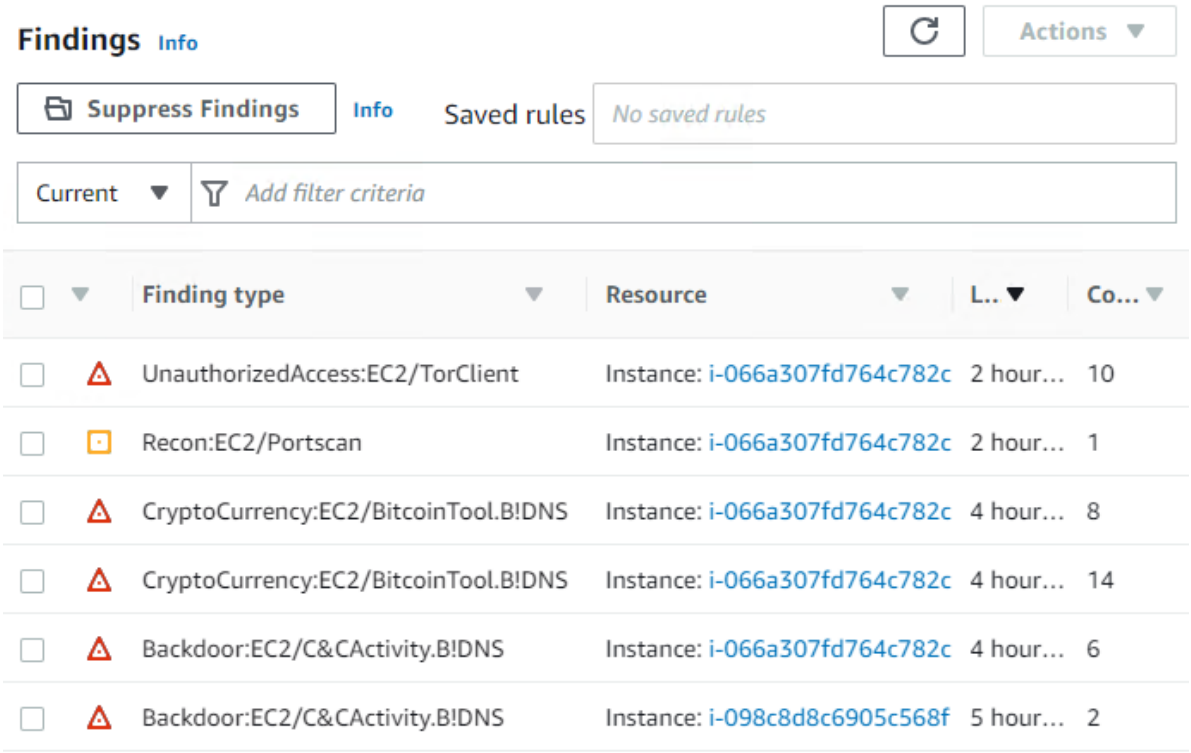
4 Security Tools

4.1 AWS GuardDuty

De configuratie van GuardDuty is heel eenvoudig. Met één enkele klik op een knop wordt GuardDuty geactiveerd. Bij het activeren van GuardDuty wordt er aan AWS-toestemming gegeven om CloudTrial Logs, VPC Flow Logs en DNS query's te analyseren.

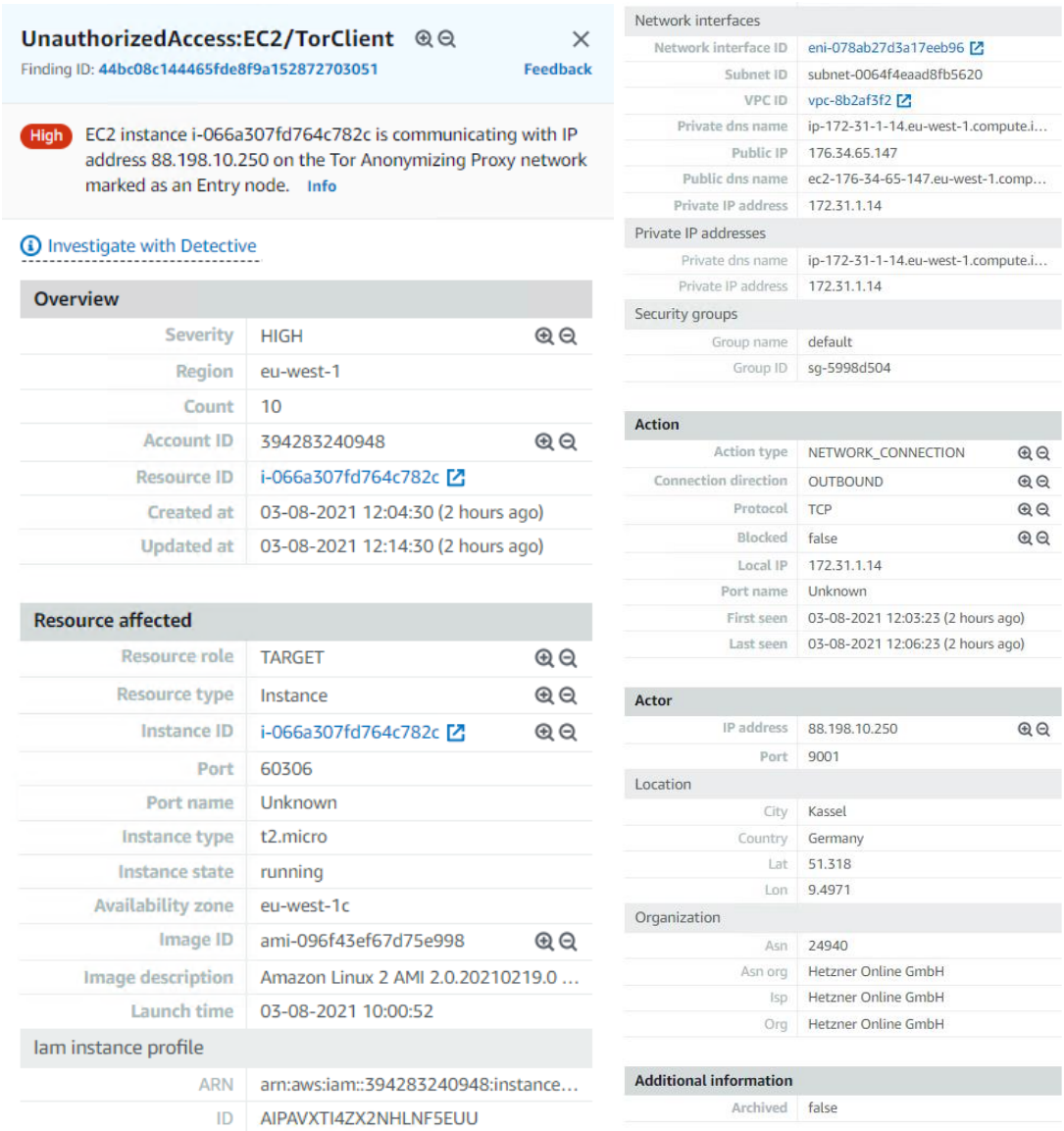
Indien gewenst kunnen de logs opgeslagen worden in een S3-bucket.

GuardDuty heeft niet veel functionaliteiten. De tool zal de volledige AWS-omgeving controleren op mogelijke aanvallen en alerts geven als er problemen zijn. Op figuur 8 is een lijst te zien met meldingen van GuardDuty.



Figuur 8: GuardDuty: Lijst met meldingen

Als er doorgeklikt wordt op zo'n melding kunnen er meer details opgevraagd worden. Deze details gaan over de resource die slachtoffer is, de netwerkinterface waar het probleem gevonden werd met de netwerkinformatie, de locatie en organisatie van de gevaarlijke connectie. Deze details zijn te zien op Figuur 9.



Figuur 9: GuardDuty: Details van een melding

Het grootste voordeel van GuardDuty is dat er geen third-party tool nodig is om minimum security te hebben. Ook het pay-as-you-go model is handig zodat er enkel betaald wordt voor wat er verbruikt is. Er zijn heel weinig opties om iets te configureren en ondersteuning voor Kubernetes is ook niet aanwezig.

Voordelen

- AWS Native
- Heel eenvoudige configuratie
- Pay-as-you-go model

Nadelen

- Weinig opties
- Geen Kubernetes support

4.2 Qualys

De configuratie van Qualys is relatief eenvoudig. Er moet een sensor gedeployed worden op de Kubernetes cluster via een daemonset. Hiervoor moet de sensor image eerst gepusht worden naar een private container registry.

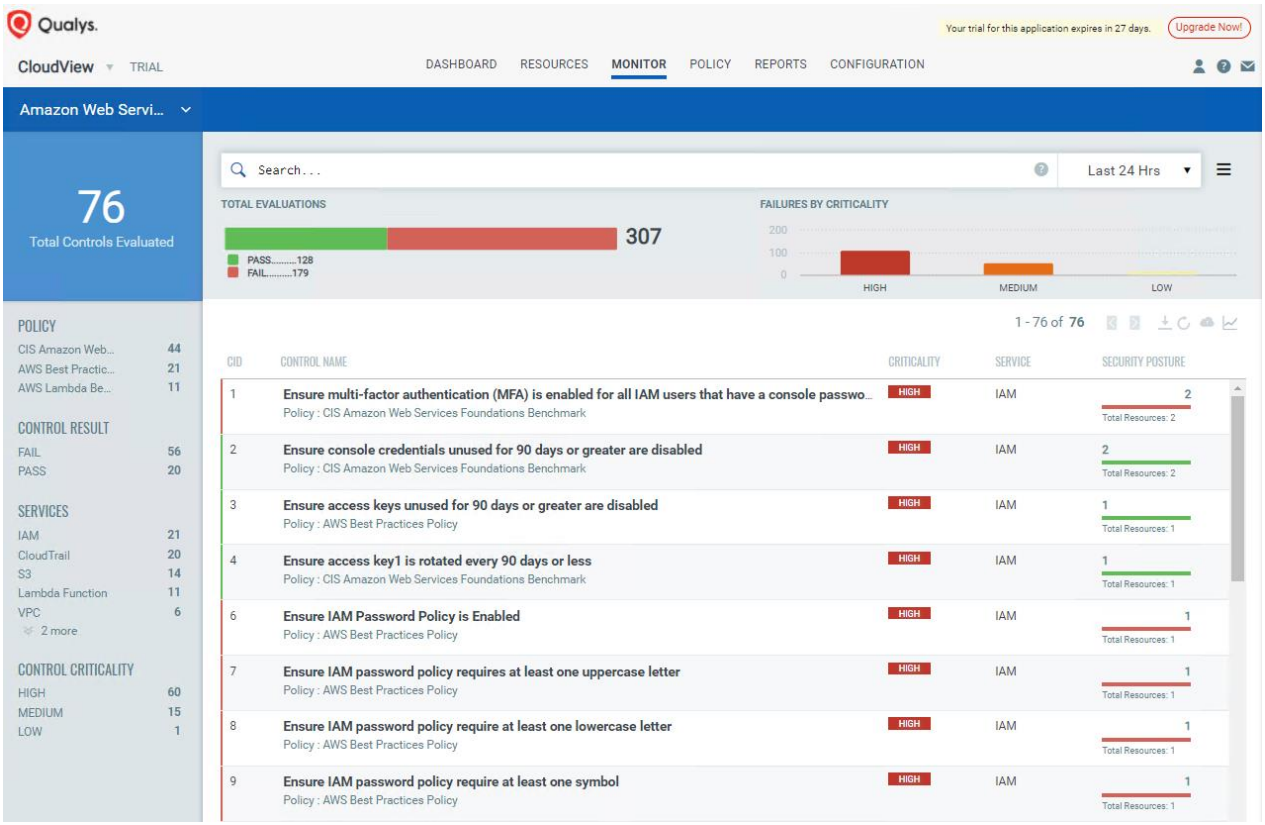
De daemonset moet als laatste nog aangepast worden op volgende waardes

- CustomerID: Het klantennummer bij Qualys
- ActivationID: Activatiecode van Qualys
- Image Location : Locatie van de private container registry waar de sensor image staat.

Als laatste stap moet de daemonset gedeployed worden op de Kubernetes cluster.

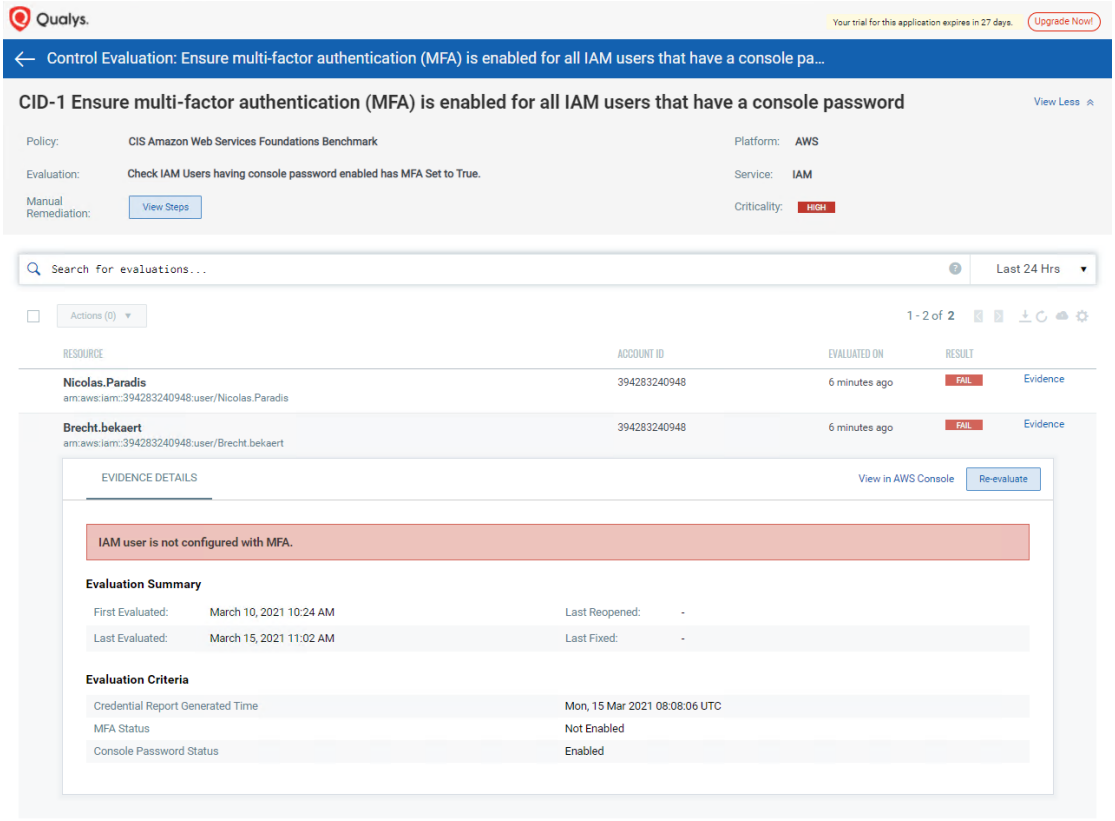
Om de kwetsbaarheden na te gaan van de AWS-omgeving is er een Cloudformation-stack gedeployed op de omgeving. Deze stack geeft Qualys read-only rechten op de omgeving.

Kwetsbaarheden in de AWS-omgeving worden op een duidelijke manier aangetoond. Met behulp van query's en filters kunnen kwetsbaarheden apart bekeken worden. Een lijst van kwetsbaarheden is te zien op figuur 10.



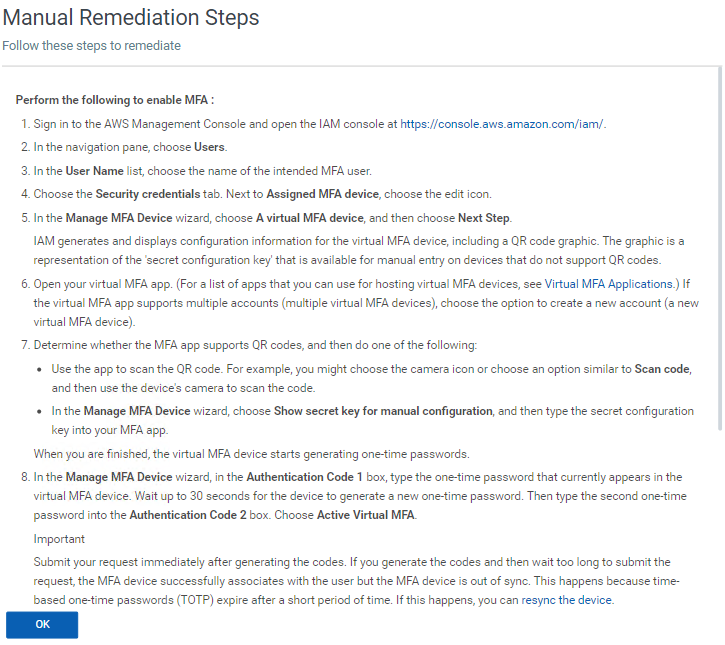
Figuur 10: Qualys: Lijst met kwetsbaarheden in AWS-omgeving

Er kan ook doorgeklikt worden op een kwetsbaarheid om meer details te krijgen. Voorbeeld van een detailweergave is te zien op figuur 11.



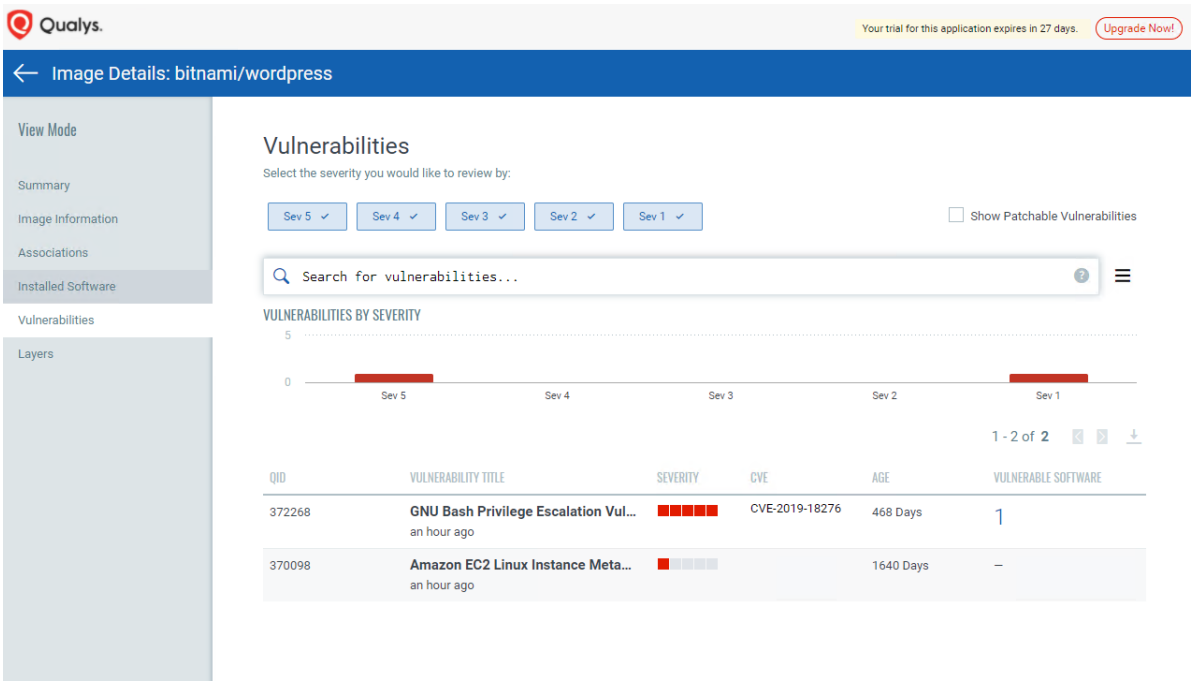
Figuur 11: Qualys: Detailweergave van een kwetsbaarheid

In sommige gevallen is er ook een stappenplan beschikbaar om de kwetsbaarheid te verhelpen. Een voorbeeld van een stappenplan is te zien op figuur 12.

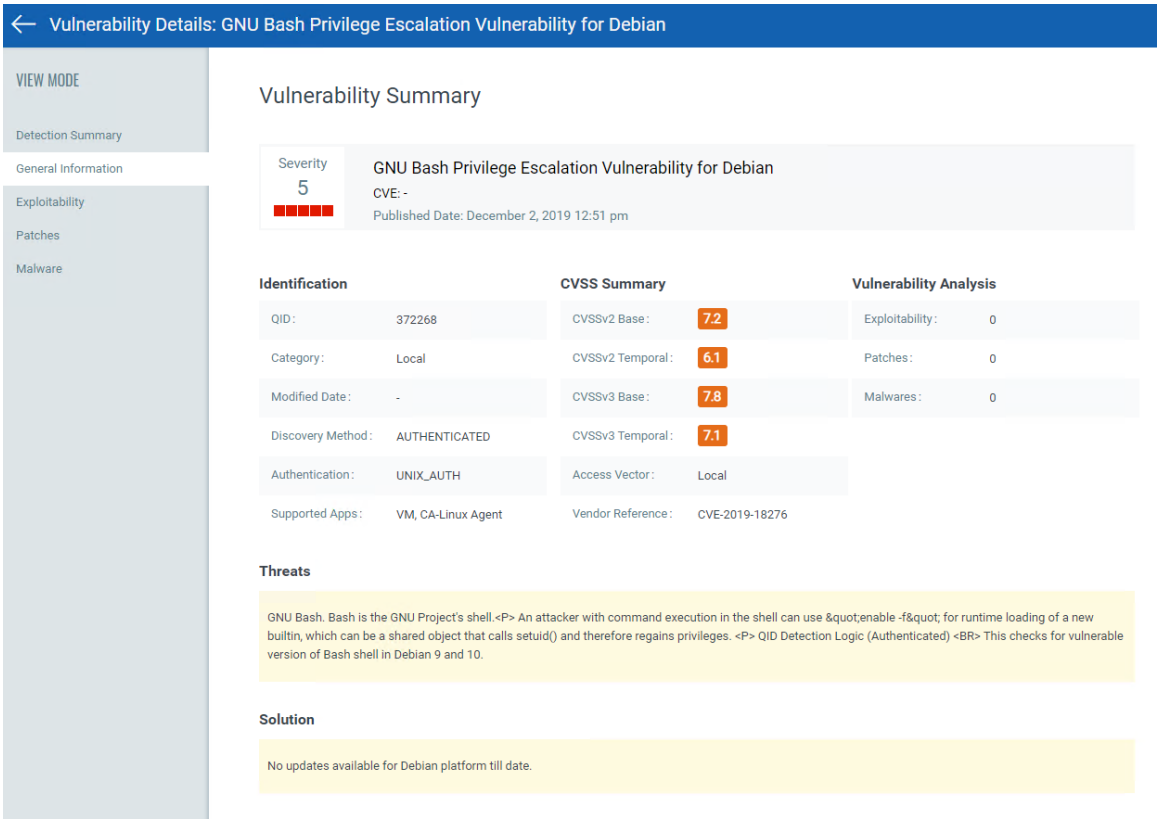


Figuur 12: Qualys: Stappenplan om kwetsbaarheid op te lossen

Naast de beveiliging van de AWS-omgeving is er ook een optie ‘Container Security'. Deze optie zal de kwetsbaarheden weergeven van een bepaalde image. Door op een image te klikken zullen er details weergegeven worden met de kwetsbaarheden en in welke mate ze ernstig zijn. Dit wordt afgebeeld op figuur 13 en figuur 14. Bij sommige kwetsbaarheden is er een stappenplan beschikbaar om dit op te lossen. Kwetsbaarheden kunnen ook weergegeven worden op container niveau. Hierbij wordt gekeken naar de container in runtime.



Figuur 13: Qualys: Overzicht kwetsbaarheden van image



Figuur 14: Qualys: Details van een kwetsbaarheid

Voordelen

- Eenvoudig in gebruik
- Stappenplan bij kwetsbaarheden
- Multi Cloud

Nadelen

- Basis interface
- Duur voor beperkte functionaliteiten

4.3 Prisma Cloud

De configuratie van Prisma Cloud is heel eenvoudig. Prisma Cloud deployt een defender op de cluster aan de hand van een daemonset dat verkregen is via de interface van de tool. In deze interface volstaat het om enkel de clusternaam te wijzigen. Voor het deployen van de daemonset moet er wel een namespace 'twistlock' aangemaakt worden.

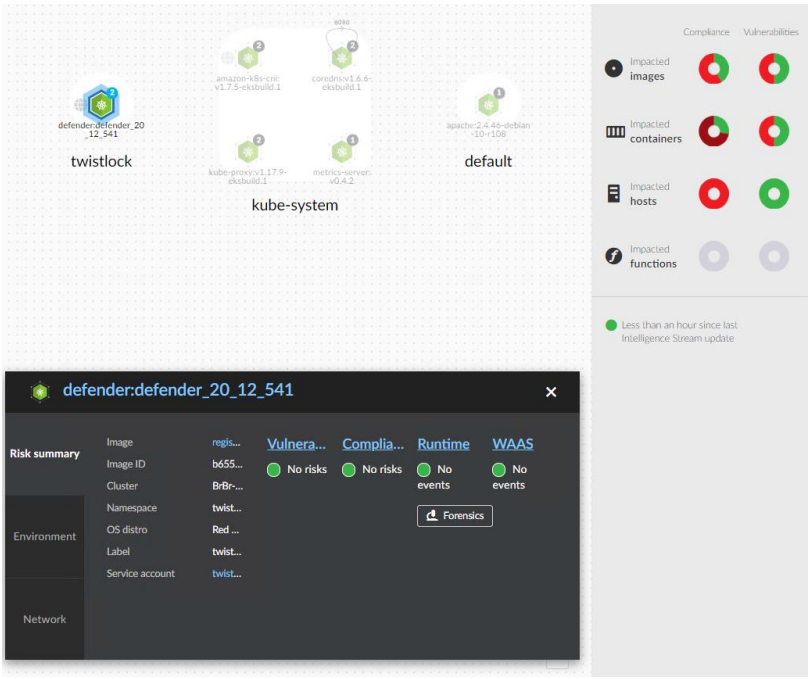
Om de AWS-omgeving te monitoren is er opnieuw een Cloudformation-stack nodig die read-only rechten toekend aan Prisma Cloud.

Prisma Cloud heeft de optie om compliance na te gaan van de resources op de AWS-omgeving. Er zijn verschillende industriestandaards die beschikbaar zijn. Bij het klikken op zo'n standaard zullen er meer details weergeven worden hoe de AWS-omgeving presteert op deze standaard. Er wordt ook onderscheid gemaakt van low, medium en high alerts zodat de fouten opgelost worden aan de hand van de prioriteit van de fouten. Alles wordt weergegeven op een overzichtelijk dashboard. Dit is te zien op figuur 15.

Rapporten kunnen ook aangemaakt worden om een overzicht te krijgen van de compliance fouten. Deze rapporten kunnen dagelijks of wekelijks automatisch gegenereerd worden zodat wijzigingen snel gedetecteerd worden. Deze rapporten kunnen ook per mail verstuurd worden naar een persoon.

Figuur 15: Prisma Cloud: Compliance dashboard

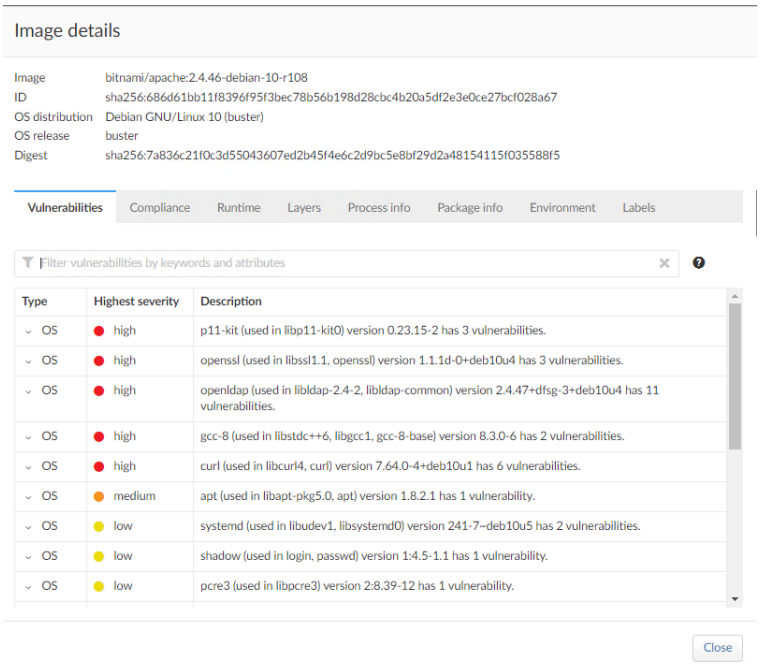
Container Security wordt verzorgd door Twistlock. Hierdoor worden kwetsbaarheden in containers en images makkelijk zichtbaar en duidelijk weergegeven. Met Prisma Cloud is er een visuele weergave van de containers en namespaces op de cluster. Dit is te zien op figuur 16. Bij het klikken op een container zullen er meer details beschikbaar zijn over de kwetsbaarheden.



Figuur 16: Prisma Cloud: Visuele weergaven van namespaces en containers

Naast de containers worden ook de images en de worker nodes beveiligd. Bij de images kan er gekeken worden naar de verschillende kwetsbaarheden. Deze worden op een overzichtelijke manier opgelijst. Bij het klikken op een image zullen er meer details beschikbaar zijn. Dit is te zien op figuur 17. Naast de security van containers wordt de netwerktraffic gemonitord. Elke verdachte handeling zal standaard een alert geven.

Met Prisma Cloud is het ook mogelijk om bij bepaalde kwetsbaarheden andere zaken te blokkeren. Standaard staat alles ingesteld op alert-only maar dit kan zeker aangepast worden.



Figuur 17: Prisma Cloud: Details van een image

Voordelen

- Eenvoudige configuratie
- Visueel sterke oplossing
- Veel functionaliteiten
- Multi Cloud

Nadelen

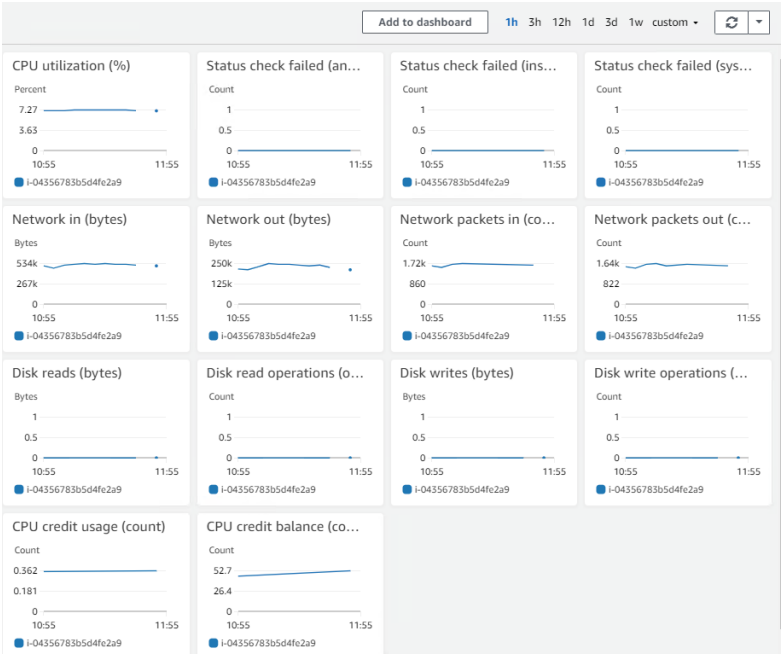
- Prijs
- Documentatie soms verwarrend

5 Monitoring Tools

5.1 AWS Cloudwatch

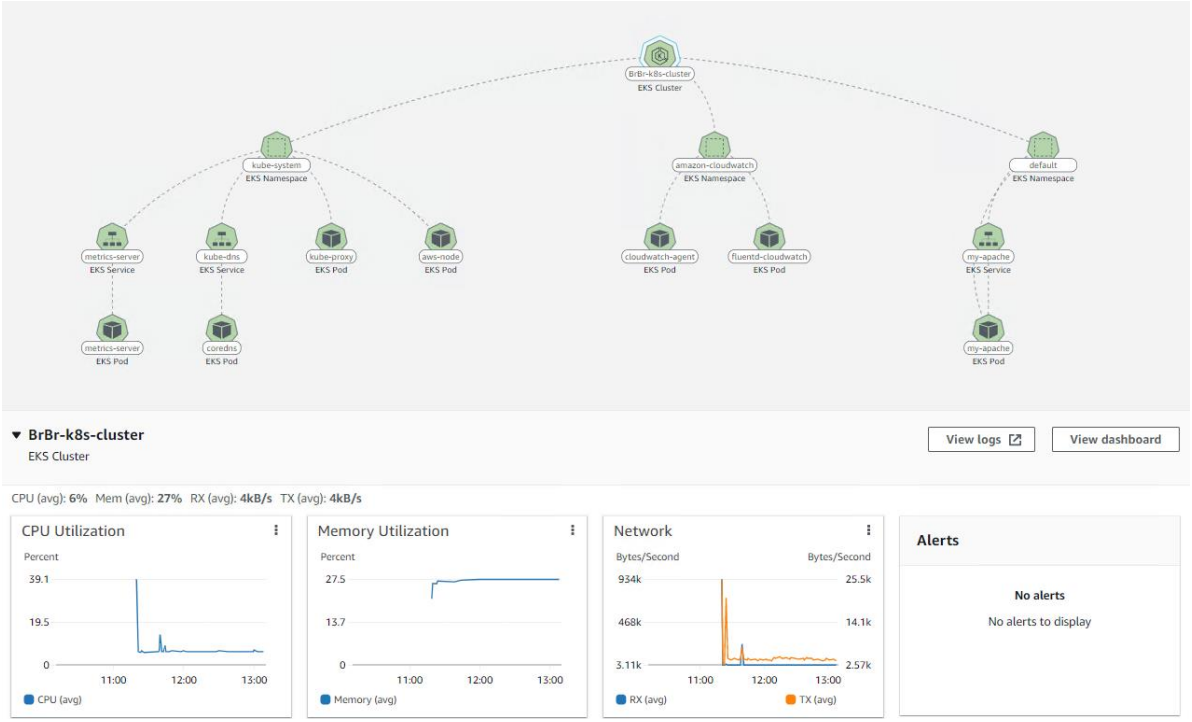
AWS Cloudwatch is ook relatief eenvoudig in configuratie. Er moet ook zoals andere tools een daemonset gedeployed worden op de Kubernetes cluster. Voor monitoring van de Kubernetes cluster zal er geen probleem zijn na het deployen van de daemonset. Om de worker nodes te monitoren moeten de juiste IAM-rollen toegekend zijn aan de instances.

Cloudwatch heeft verschillende dashboards beschikbaar. Er is een dashboard voor EC2-instances. Op dit dashboard zijn verschillende metrics te zien zoals CPU usage, disk usage, network usage en CPU-credit usage. Een voorbeeld hiervan is te zien op figuur 18. In het voorbeeld staat de filter ingesteld op 1 uur. Het is ook mogelijk een andere tijdsindeling te selecteren om over een langer interval de metrics te analyseren.



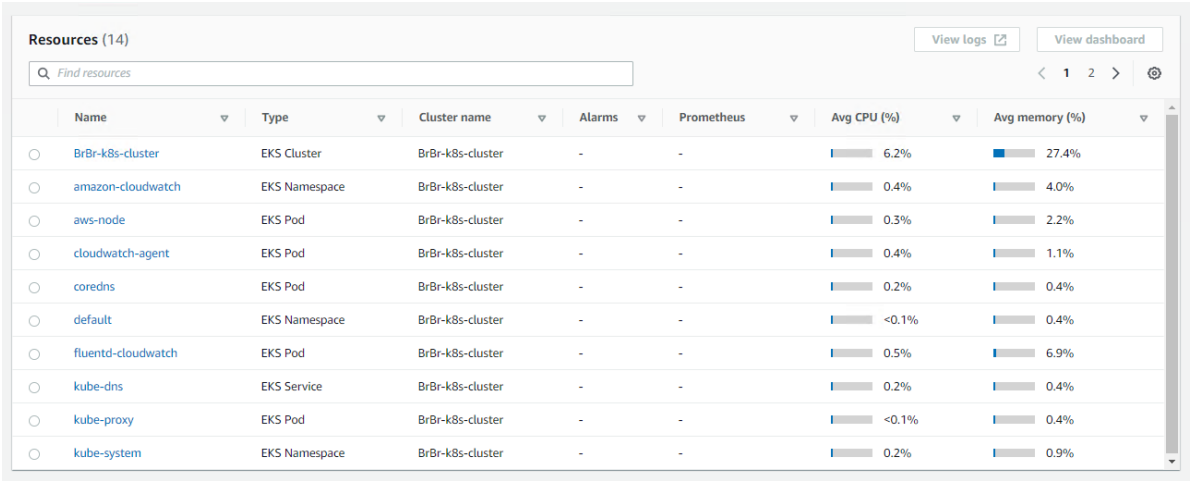
Figuur 18: AWS Cloudwatch: Dashboard van EC2-instance

Om de Kubernetes cluster te monitoren hebben we container monitoring nodig. Hiervoor is de daemonset op de cluster gedeployed. Door deze container monitoring is er een visuele weergave van de verschillende namespaces, pods, services in de cluster. Ook zijn er al enkele metrics beschikbaar onderaan de map. Een voorbeeld is te zien op figuur 19.



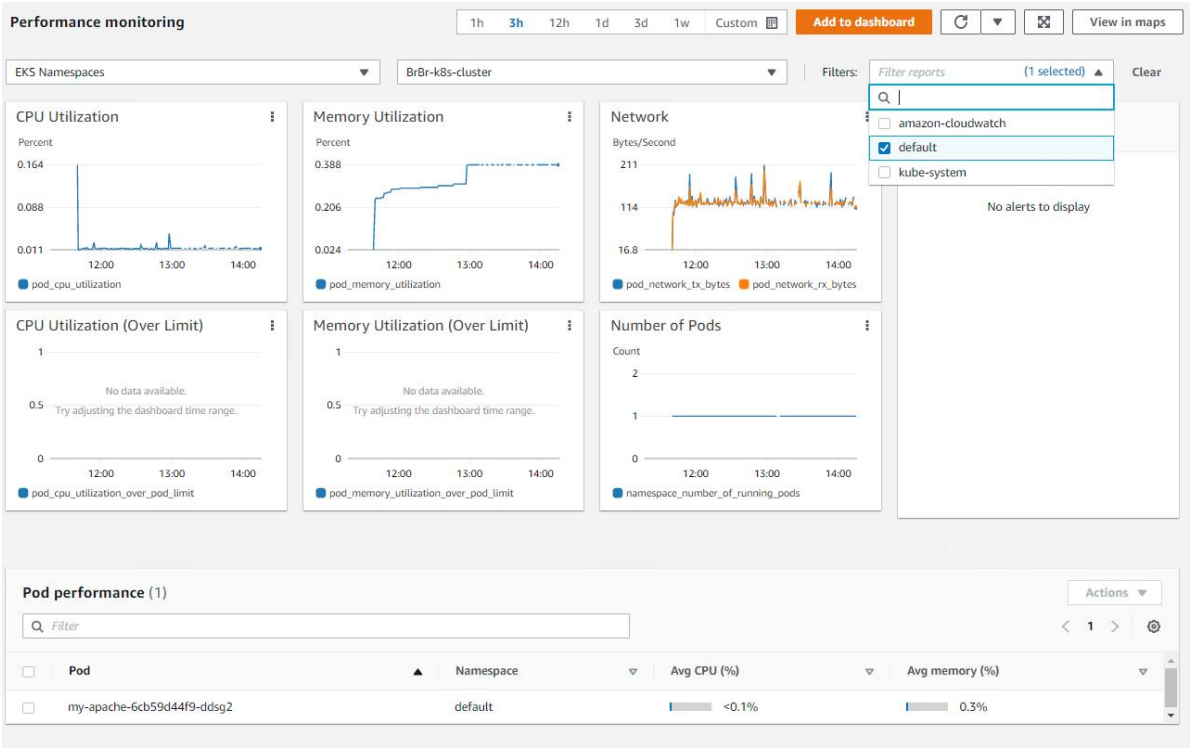
Figuur 19: AWS Cloudwatch: Visuele weergave van de cluster

Deze visuele weergave heeft ook een lijst waarbij de average CPU usage en average memory usage vermeld staan. Ook de ingestelde alarmen zijn zichtbaar in deze lijst. Deze lijst is te zien op figuur 20. Bij het klikken op een onderdeel in de lijst wordt er een gedetailleerd dashboard getoond. In figuur 21 zijn de details van de namespace 'default' beschikbaar. Deze details zijn CPU usage, memory usage, network usage en welke pods er aanwezig zijn in deze namespace.



Name	Type	Cluster name	Alarms	Prometheus	Avg CPU (%)	Avg memory (%)
BrBr-k8s-cluster	EKS Cluster	BrBr-k8s-cluster	-	-	6.2%	27.4%
amazon-cloudwatch	EKS Namespace	BrBr-k8s-cluster	-	-	0.4%	4.0%
aws-node	EKS Pod	BrBr-k8s-cluster	-	-	0.3%	2.2%
cloudwatch-agent	EKS Pod	BrBr-k8s-cluster	-	-	0.4%	1.1%
coredns	EKS Pod	BrBr-k8s-cluster	-	-	0.2%	0.4%
default	EKS Namespace	BrBr-k8s-cluster	-	-	<0.1%	0.4%
fluentd-cloudwatch	EKS Pod	BrBr-k8s-cluster	-	-	0.5%	6.9%
kube-dns	EKS Service	BrBr-k8s-cluster	-	-	0.2%	0.4%
kube-proxy	EKS Pod	BrBr-k8s-cluster	-	-	<0.1%	0.4%
kube-system	EKS Namespace	BrBr-k8s-cluster	-	-	0.2%	0.9%

Figuur 20: AWS Cloudwatch: Lijst van resources van de cluster



Figuur 21: AWS Cloudwatch: Details van namespace 'default'

Om ervoor te zorgen dat er proactief gemonitord kan worden kunnen er alarmen ingesteld worden. Deze alarmen kunnen op AWS-resources ingesteld worden of op resources van de Kubernetes cluster. Het instellen van zo'n alarm zal via een bepaalde metric gebeuren. Na het selecteren van de metric kan er gekozen worden voor 'static' of 'anomaly detection'.

Bij static zal de metric niet boven of onder een geselecteerde waarde mogen komen. Indien deze metric toch buiten de grenzen gaat zal het alarm afgaan. Bij anomaly detection zal Cloudwatch de metric de eerste uren bestuderen en een normaal genereren. Er wordt ook een verschil met normaal ingesteld. Nadien zal bij het overschrijden van de metric een alarm afgaan.

Als een alarm afgaat kunnen er verschillende acties ondernomen worden afhankelijk van de geselecteerde metric. Per alarm kan er een contactpersoon ingesteld worden om per mail op de hoogte gebracht te worden. Daarnaast kunnen bij sommige metrics extra instances aangemaakt worden of bestaande instances herstart of gestopt worden.

Voordelen

- AWS Native
- Pay-as-you-go model
- Eenvoudig in gebruik

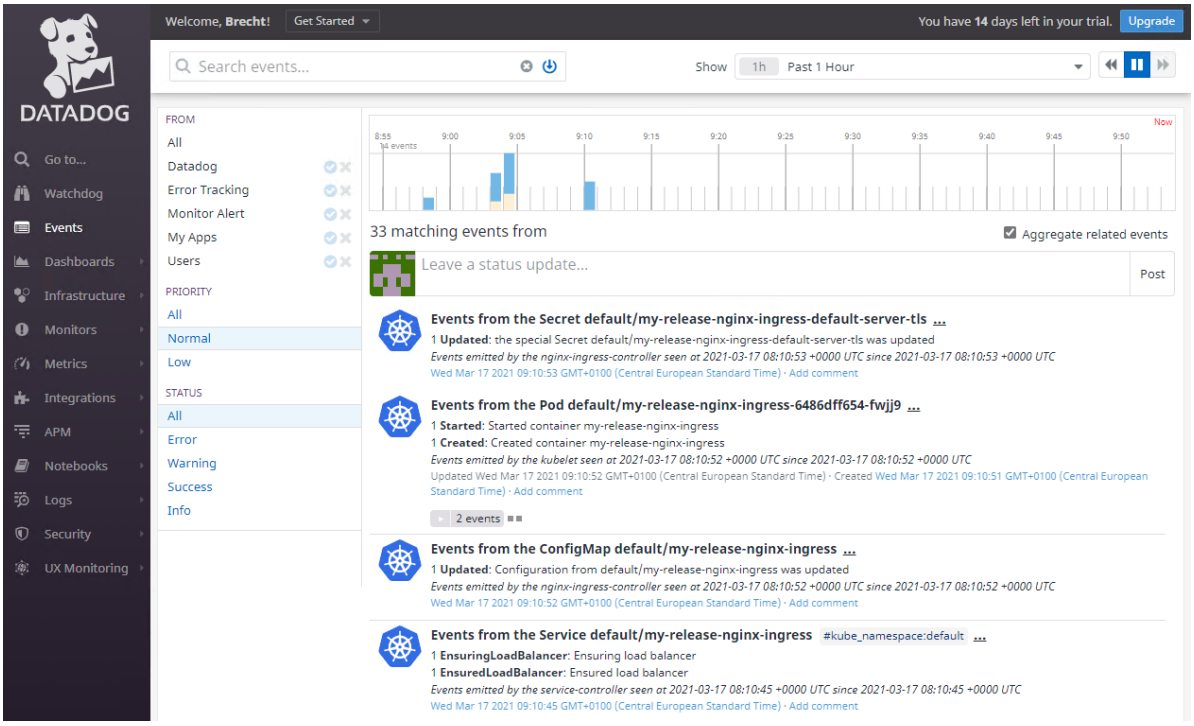
Nadelen

- IAM-rollen voor bepaalde zaken te monitoren
- Visueel minder sterk
- Geen multi cloud

5.2 Datadog

De configuratie van Datadog is ook relatief eenvoudig. Datadog wordt op de Kubernetes cluster geïnstalleerd via helm charts. Bij deze helm charts zal er een lokaal 'datadog-values.yaml' bestand toegevoegd worden met parameters voor Datadog. Het 'datadog-values.yaml' bestand is te downloaden van Datadog zelf. In het bestand is de cluster name, process collection en network monitor gewijzigd.

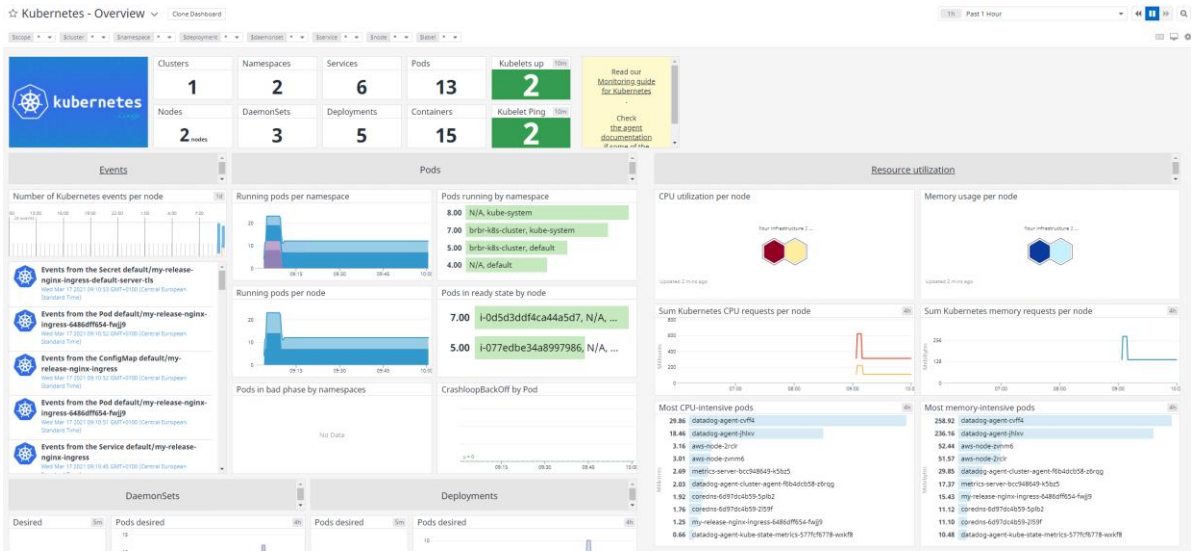
Datadog heeft een heel interessante events pagina waar alle acties die genomen worden op de Kubernetes cluster vermeld worden. Op deze events pagina is te zien wanneer bepaalde pods op de cluster komen of verwijderd worden, service-informatie, secret informatie, …. De events kunnen ook gefilterd worden op prioriteit, status en in tijd. Ook is het mogelijk zelf een status update toe te voegen aan de events pagina. Een voorbeeld van de events pagina is te zien op figuur 22.



Figuur 22: Datadog: Events pagina

Datadog heeft ook de mogelijkheid om een dashboard te gebruiken. In Datadog is het mogelijk om voor Kubernetes standaard dashboards te creëren. Op deze dashboards staan al verschillende metrics zoals aantal namespaces, aantal pods, aantal containers, meest CPU intensieve pod, …. Een voorbeeld van het standaard dashboard is te zien op figuur 23.

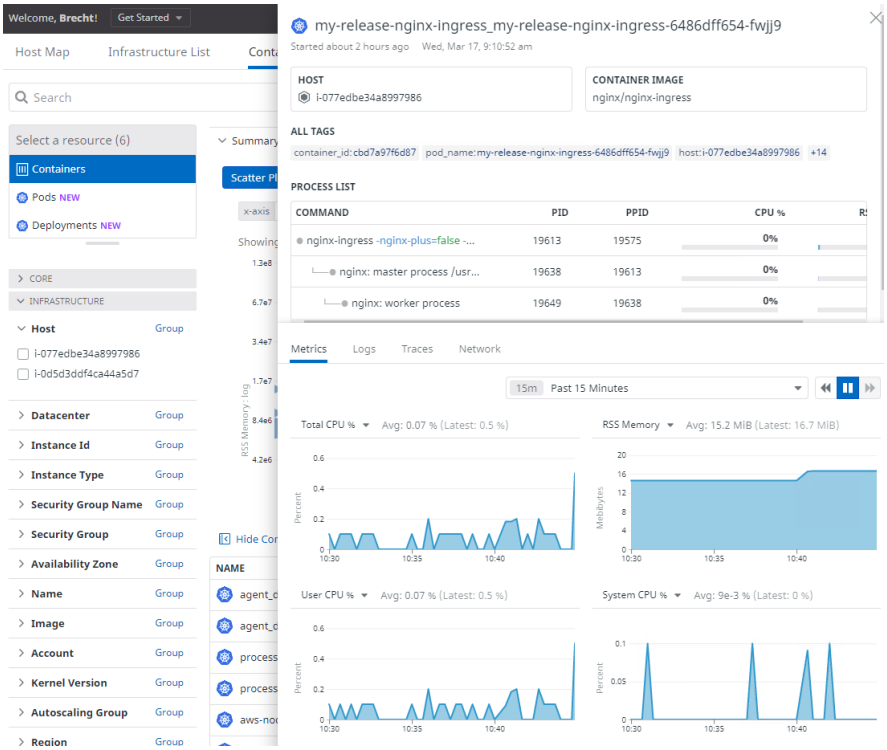
Er zijn ook dashboards aanwezig op de instances op AWS te monitoren. Deze dashboards zijn minder interessant omdat deze geen zicht geeft over de resources die verbruikt worden in de Kubernetes cluster.



Figuur 23: Datadog: Dashboard Kubernetes: Overview

Met Datadog is er ook een mogelijkheid om de systeeminformatie van de instances op te vragen en de metrics zoals CPU en memory, maar ook welke containers en processen er draaien op de instance.

Verder in Datadog kan er een lijst gegenereerd worden met containers, pods en deployments. Deze lijst kan gefilterd worden op host, namespace, region, …. In deze lijst kan er geklikt worden op een specifieke container/pod en kunnen de metrics weergegeven worden van die specifieke container/pod. Een voorbeeld hiervan is te zien op figuur 24.

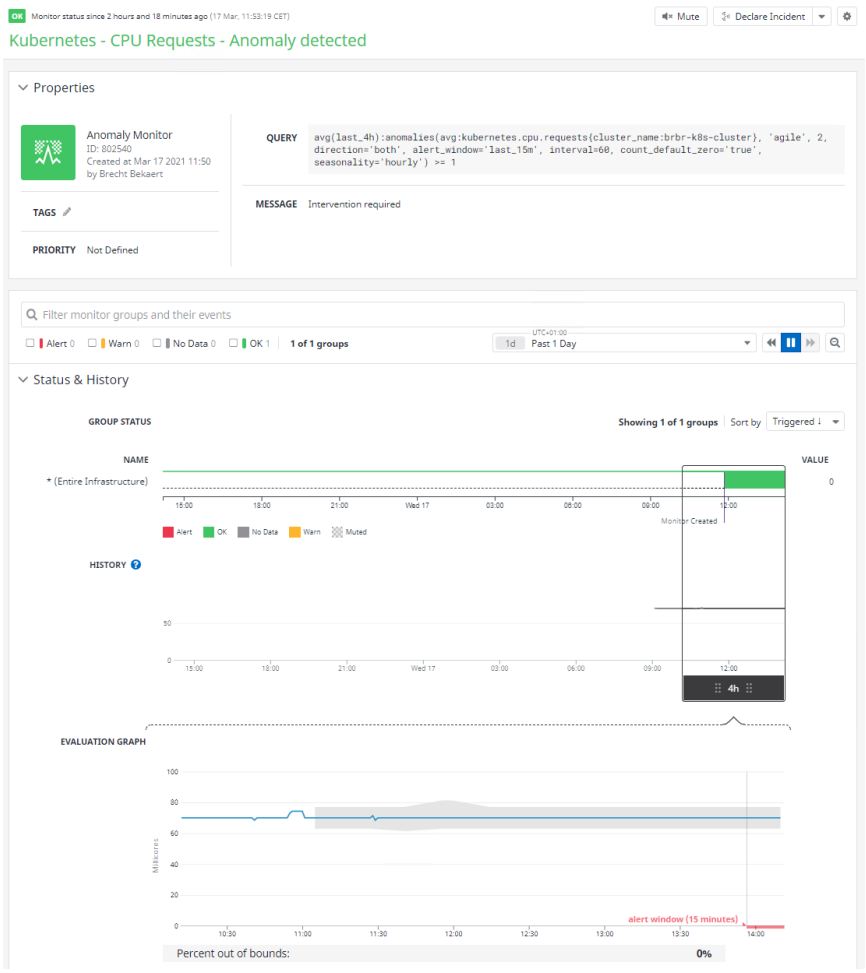


Figuur 24: Datadog: Metrics van een container

Datadog heeft ook alarmen. Met alarmen kan een persoon op de hoogte worden gebracht van een incident dat plaatsgevonden heeft. Een incident is een alarm dat is af is gegaan omdat een bepaalde metric een bepaalde waarde overschreden heeft.

Met Datadog kan er op verschillende types gaan monitoren zoals host, metrics, anomaly, APM, network, events, ….

Als je op een aangemaakte monitor klikt, wordt er een grafiek aangemaakt van de metric om te zien hoe de metric zich veranderd heeft. Een voorbeeld is te vinden op figuur 25.



Figuur 25: Datadog: Details van een monitor

Voordelen

- Eenvoudig in gebruik
- Multi cloud
- Visueel sterke oplossing

Nadelen

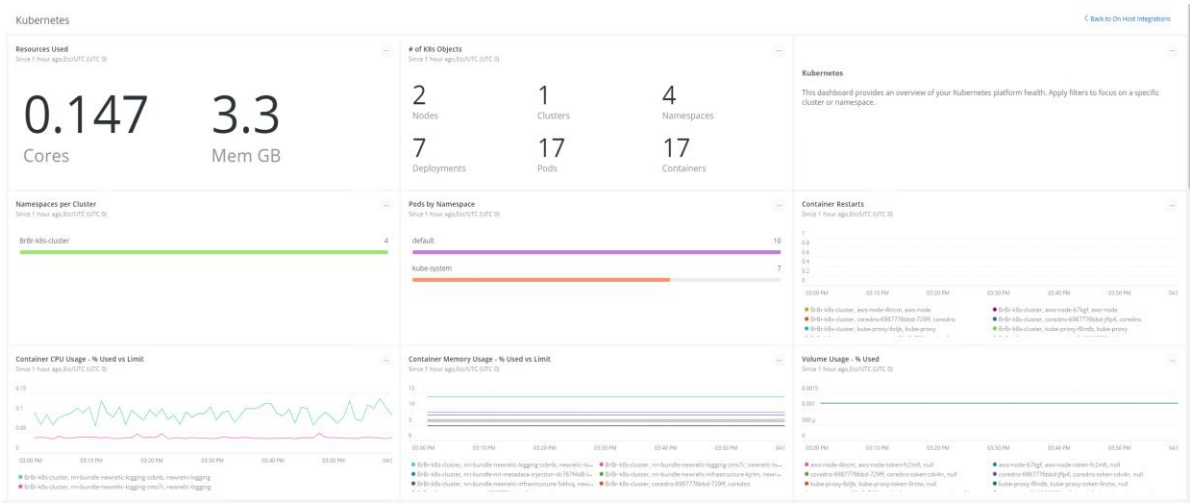
- Configuratie van Datadog values.
- Soms traag in het laden

5.3 New Relic

De configuratie van New Relic is ook eenvoudig. Er moet opnieuw een daemonset op de cluster geplaatst worden. De daemonset is te downloaden via de New Relic website.

New Relic heeft ook een events pagina zoals Datadog maar deze pagina heeft minder mogelijkheden. Op deze events pagina zal enkel informatie over de cluster komen en dus geen informatie over de wijzigingen in de tool zelf.

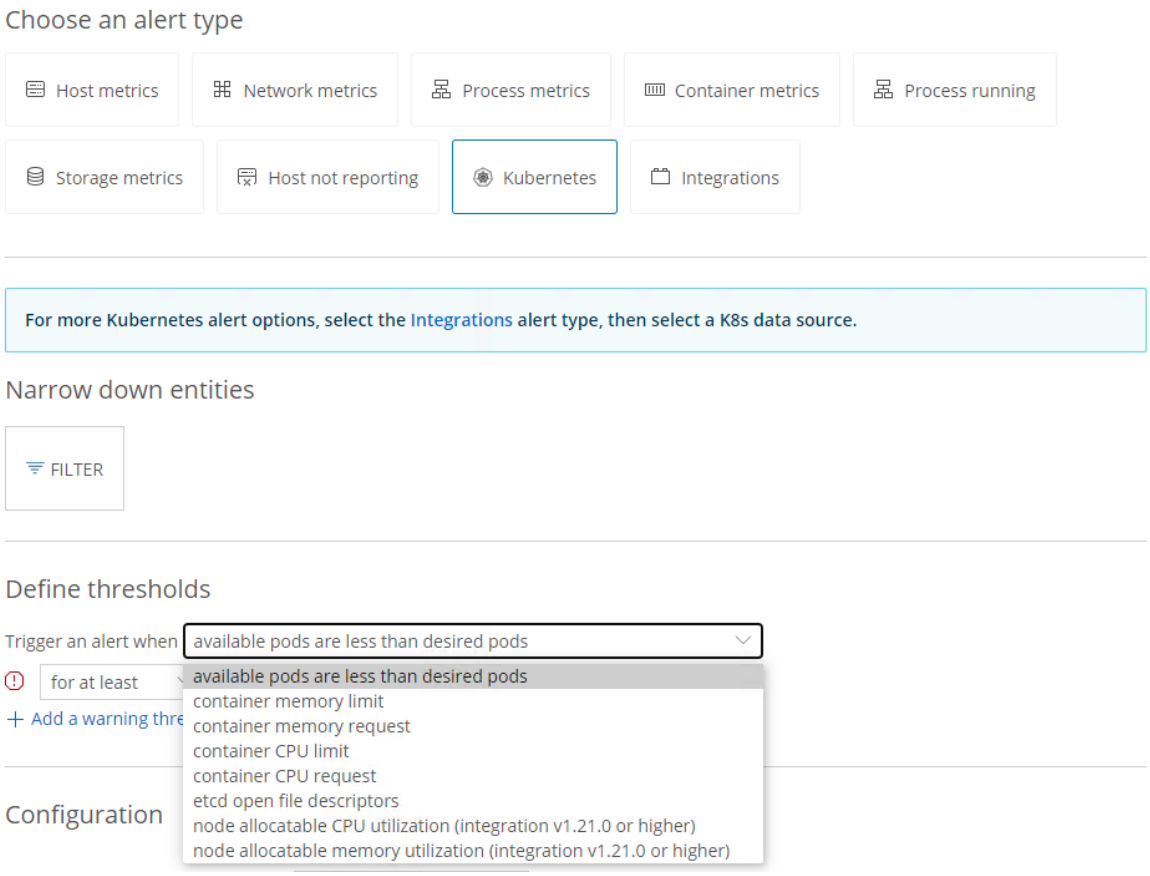
Daarnaast heeft New Relic ook een standaard Kubernetes dashboard met enkele belangrijke metrics. Metrics zoals resources used, Kubernetes objects, pods per namespace, …. Een voorbeeld van dit dashboard is te zien op figuur 26. Naast het Kubernetes dashboard zijn er ook dashboards beschikbaar van de worker nodes. Deze hebben dan CPU usage, Memory usage, network usage, disk usage als belangrijke metrics. Indien gewenst kan er ook een eigen dashboard aangemaakt worden met de metrics naar keuze.



Figuur 26: New Relic: Kubernetes dashboard

Naast de dashboards van Kubernetes is er ook een mogelijkheid om de verschillende containers op de cluster te bekijken. Op die lijst kunnen er ook filters toegepast worden. Er kan vervolgens op een container geklikt worden om meer details te zien van deze container. De details zijn redelijk beperkt. De belangrijkste metrics die aanwezig zijn is CPU usage, memory usage, storage used, restart count, is ready, ….

Verder kunnen er ook alarmen ingesteld worden zodat er proactief kan gemonitord worden. Er kunnen alarmen ingesteld worden die afgaan vanaf een bepaalde metric een bepaalde threshold heeft bereikt, maar er kan ook ingesteld worden dat het alarm pas zal afgaan vanaf de metric binnen 1 week de threshold zou bereiken. Hierdoor kunnen we proactief monitoren. Een voorbeeld van het aanmaken van een alarm is te zien in figuur 27.



Figuur 27: New Relic: Aanmaken van een nieuw alarm

Voordelen

- Eenvoudige confituratie
- Multi cloud

Nadelen

- Prijs kan oplopen
- Volgens Gartner minder sterk mee met de markt

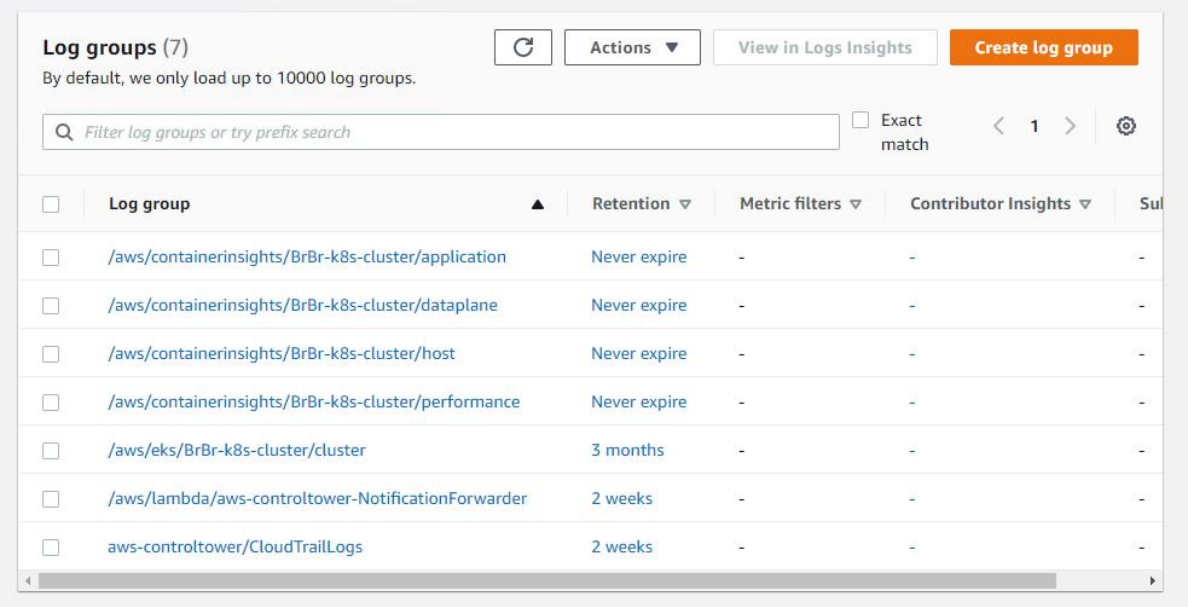
6 Logging Tools

6.1 AWS Cloudwatch

De configuratie van AWS Cloudwatch is ongeveer hetzelfde als voor monitoring. Er moet enkel een extra daemonset op de cluster geplaatst worden. Deze daemonset is te downloaden via AWS en bevat de clusternaam en regio van de cluster.

Om logging te activeren op de master node moet dit via Terraform aangeduid worden met de parameter “enabled_cluster_log_types”.

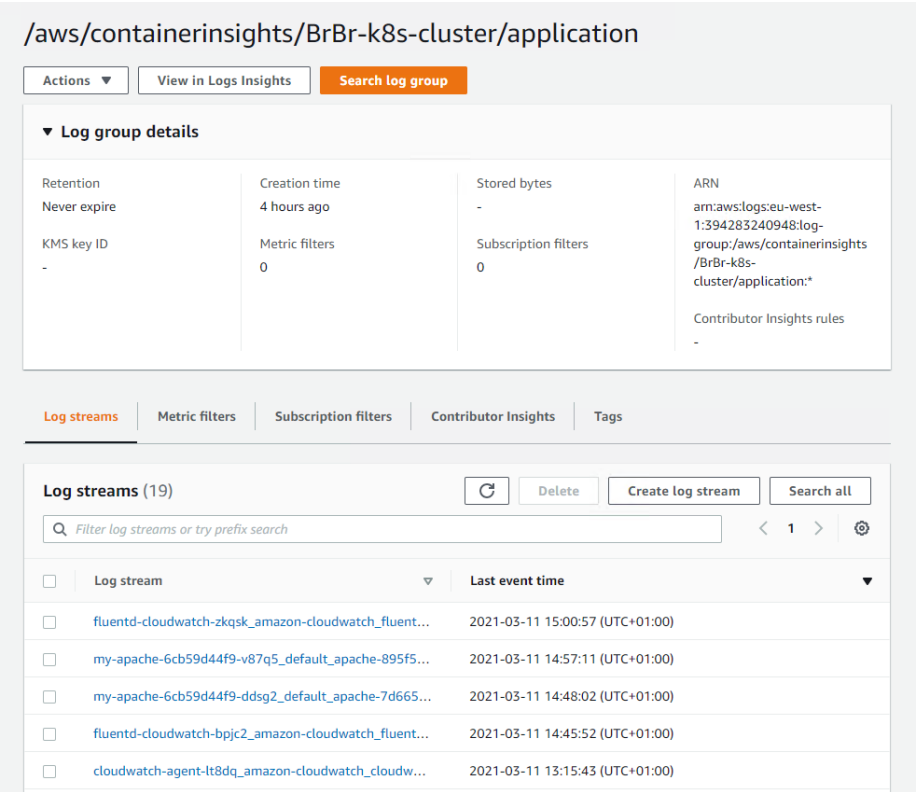
Door de daemonset op de cluster te deployen worden er verschillende log groups aangemaakt. Deze log groups bevatten logs van de verschillende onderdelen van de Kubernetes cluster. Een voorbeeld van de verschillende log groups is te zien op figuur 28.



Figuur 28: AWS Cloudwatch: Voorbeeld van log groups

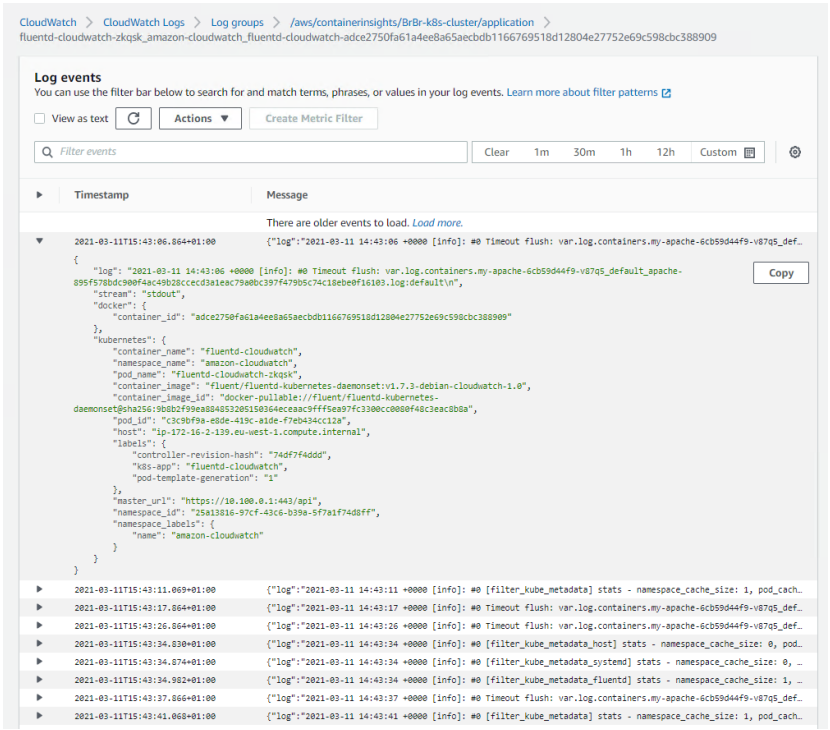
Om de logs van de verschillende containers te bekijken kan de log group “/aws/containerinsights/BrBr-k8s-cluster/application” bekeken worden.

In deze categorie zitten logs van de verschillende containers op de cluster. Bij het doorklikken op deze log group wordt er een lijst gegenereerd met de verschillende containers. Een voorbeeld van zo’n lijst is te zien op figuur 29.



Figuur 29: AWS Cloudwatch: Lijst van containers

Bij het klikken op een specifieke container worden de logs weergegeven van die container. Dit is te zien op figuur 30.



Figuur 30: AWS Cloudwatch: Logs van specifieke container

Als logging voor de master node geactiveerd is, komt er een log group '/aws/eks/BrBr-k8s-cluster/cluster'. In deze log group zijn logs te vinden over de verschillende taken van de master node. Er is een log group die de API-server van de master node zal loggen. Er is een 2e log group die de authenticatie met de cluster bijhoudt van iedereen. Er is een 3e log group die de uitgerolde pods en containers zal bijhouden en er is een 4e log group die logs bijhoudt over de AWS-resources zoals load balancer toewijzingen.

Voordelen

- AWS Native
- Pay-as-you-go model
- Eenvoudig in gebruik

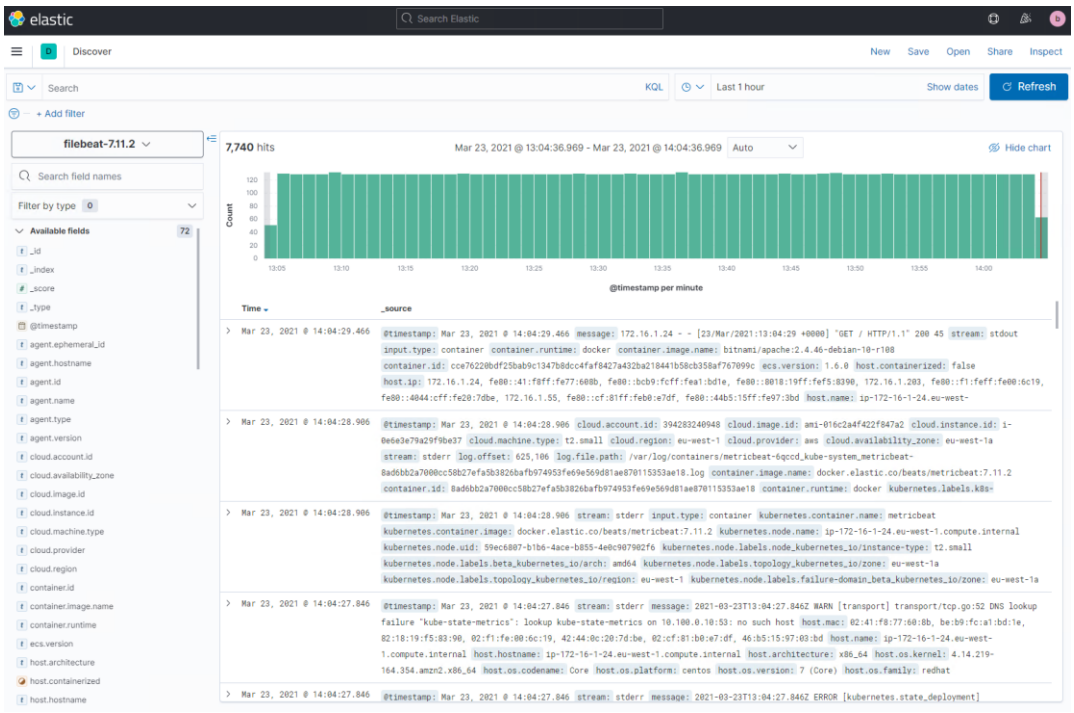
Nadelen

- Visueel minder sterk
- Geen multi cloud

6.2 ELK Stack

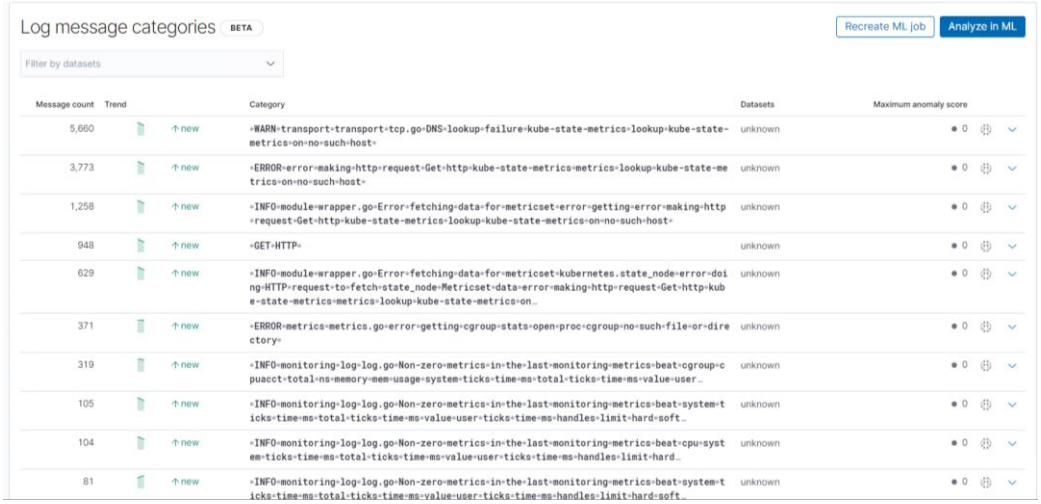
De configuratie van ELK stack is eenvoudig. Er moet een daemonset gedeployed worden van de FileBeat tool. Deze tool is onderdeel van de ELK stack.

ELK stack heeft een functionaliteit om de logs in realtime te kunnen weergeven. Hierbij kan al een filter geplaatst worden zodat enkel de logs te zien zijn van die specifieke filter. Later kan er een lijst opgevraagd worden met alle logs. In deze lijst kan er opnieuw gefilterd worden op verschillende values. Een voorbeeld hiervan is te zien op figuur 31.



Figuur 31: ELK Stack: Lijst met logs

Daarnaast heeft de ELK stack ook de mogelijkheid om patronen te herkennen in de verschillende logs. Hierdoor kunnen de logs opgedeeld worden in verschillende categorieën en worden deze overzichtelijk opgeslagen. Een voorbeeld van patronen is te zien op figuur 32.



Figuur 32: ELK Stack: Patronen in logs

Voordelen

- Goedkoop
- Machine learning
- Integratie met andere toepassingen mogelijk

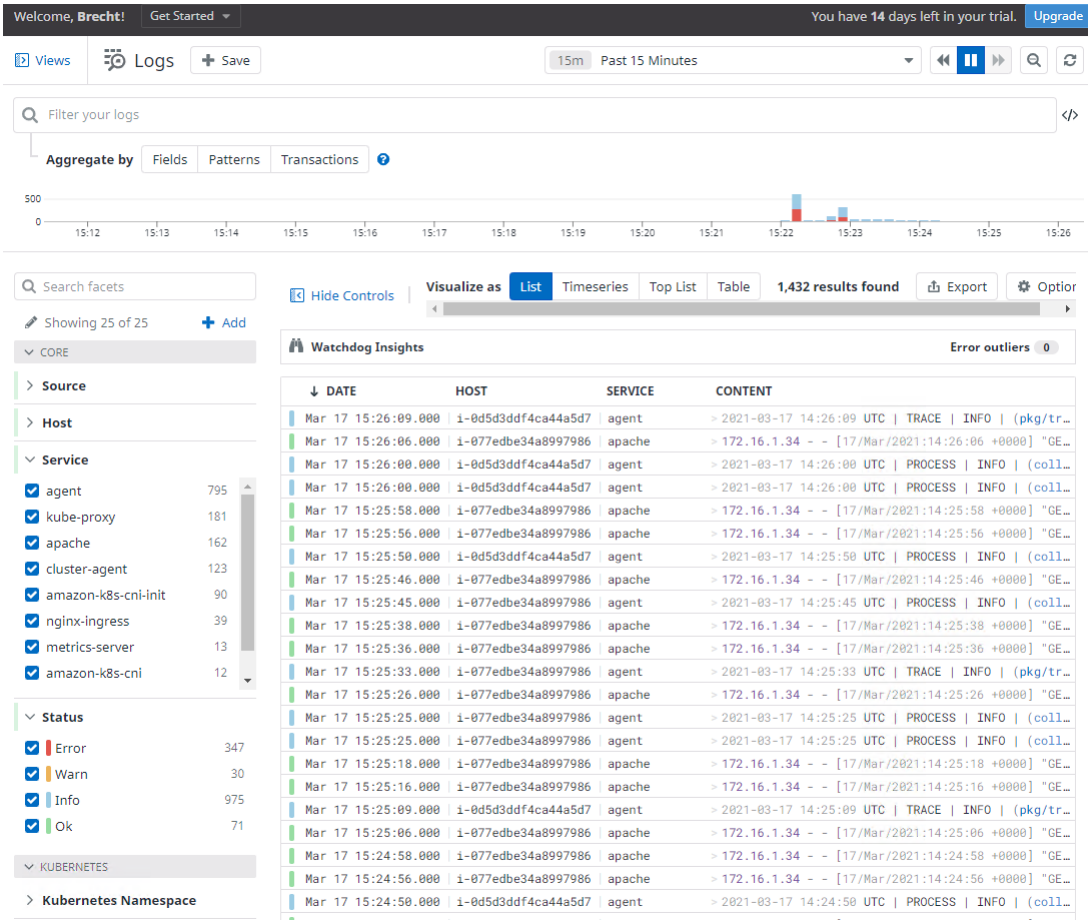
Nadelen

- Virtuele machine nodig

6.3 Datadog

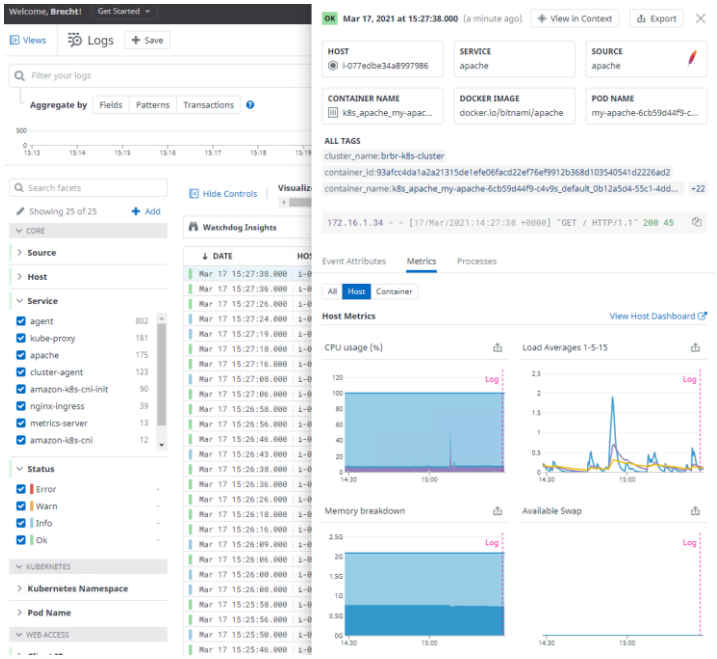
De configuratie van Datadog is op dezelfde manier als bij de monitoring tool. Er moet wel een aanpassing gebeuren in de 'datadog-values.yaml' file. In deze file met de parameter 'logging' op true geplaatst worden. Dit zorgt ervoor dat de logs van de verschillende containers opgeslagen zullen worden in Datadog.

Datadog zal een lijst aanbieden met daarin alle logs van alle containers. Deze lijst kan gefilterd worden op host, service, namespace, pod name, …. De lijst en mogelijke filters is te zien op figuur 33.



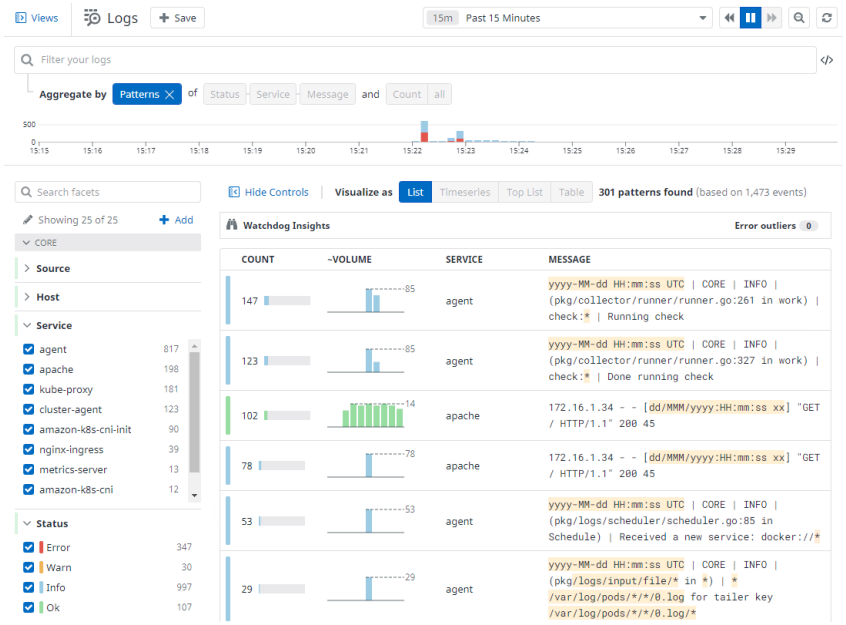
Figuur 33: Datadog: Lijst van logs

Als er vervolgens op een log geklikt wordt en de monitoring functionaliteiten van Datadog zijn ook ingeschakeld, is het mogelijk om de metrics bij die log te bekijken. Er zijn zowel host metrics en container metrics beschikbaar. Ook is het mogelijk om de processen die op dat moment aan het draaien waren te raadplegen. Een voorbeeld van de details is te zien op figuur 34.



Figuur 34: Datadog: Details van een log

Er kunnen bij Datadog ook patronen gezocht worden in de logs. Deze zorgen ervoor dat repetitieve logs snel terug te vinden zijn en geeft een inzicht in de logs op de cluster. Een voorbeeld van de patronen is te zien op figuur 35.



Figuur 35: Datadog: Patronen in logs

Voordelen

- Eenvoudig in gebruik
- Multi cloud
- Visueel sterke oplossing
- Metric bij de logs (Indien Datadog als monitoring tool gekozen wordt)

Nadelen

- Configuratie van Datadog values.

7 Automatisatie Tools

7.1 Cloudformation

Om de Kubernetes cluster op te zetten met Cloudformation is er gebruik gemaakt van de tool Eksctl.

Eksctl is een tool van Weaveworks dat gebruik maakt van Cloudformation. Met het onderstaande commando wordt een eenvoudige Kubernetes cluster opgezet a.d.h.v. de yaml file die als argument gebruikt wordt. Dit yaml bestand is te zien op codefragment 1.

```
Kubectl create -f BrBr-k8s-cluster-cf.yaml
```

```yaml
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: BrBr-k8s-cluster
  region: eu-west-1

nodeGroups:
  - name: ng-1
    instanceType: t2.small
    desiredCapacity: 2
    volumeSize: 20
    ssh:
      allow: false

```

Codefragment 1: Cloudformation: Eksctl yaml file

In codefragment 1 worden enkele metadata meegegeven voor de cluster. Deze metadata is de clusternaam en region. Ook wordt er 1 node group geconfigureerd met t2.small instances. Er is ook een optie om met ssh te connecteren naar de t2.small instances, maar deze optie is uitgeschakeld voor de veiligheid van de cluster.

7.2 Terraform

Met Terraform kan er Infrastructure as code gemaakt worden. Er kan gebruik gemaakt worden van 1 groot bestand. Dit wordt echter snel onoverzichtelijk. Terraform ondersteunt het gebruik van meerdere files. Hierdoor kan er een opsplitsing gemaakt worden tussen provider, modules, resources, variabelen, … Terraform heeft 2 manieren op het script op te bouwen.

- Resources

Met resources kunnen er verschillende zaken van de AWS-infrastructuur apart ingesteld worden. Elke resource beschrijft een infrastructuur object. Uiteraard zijn er meerdere resources nodig om 1 geheel te vormen van de infrastructuur. Het is ook mogelijk om dependencies in te stellen op bepaalde resources zodat deze enkel uitgevoerd worden als deze dependency nageleefd is. Er is echter een betere manier om een Terraform script samen te stellen.

- Modules

Een betere manier om een Terraform script op te stellen is a.d.h.v. modules. Een module is een container met meerdere resources samen. Hierdoor zijn de verschillende resources veel overzichtelijker en blijft het Terraform script proper.

```
module "vpc" {
  source = "terraform-aws-modules/vpc/aws"

  name = "my-vpc"
  cidr = "10.0.0.0/16"

  azs             = ["eu-west-1a", "eu-west-1b", "eu-west-1c"]
  private_subnets = ["10.0.1.0/24", "10.0.2.0/24", "10.0.3.0/24"]
  public_subnets  = ["10.0.101.0/24", "10.0.102.0/24", "10.0.103.0/24"]

  enable_nat_gateway = true
  enable_vpn_gateway = true

  tags = {
    Terraform = "true"
    Environment = "dev"
  }
}
```

Figuur 36: Terraform: Voorbeeld module

In figuur 36 is een Terraform module te zien van een VPC. Deze module heeft verschillende resources ingesteld staan zoals CIDR, private subnet, public subnet, …. Op bovenstaande manier kunnen verschillende resources eenvoudig beheerd worden.

8 Toolselectie

Bij de toolselectie zal elke tool kort besproken worden met de voor- en nadelen en waarom een bepaalde tool gekozen is. De gekozen tools staan vetgedrukt in onderstaande tekst.

Security Tools

GuardDuty voldoet voor deze bachelorproef niet aan de nodige functionaliteiten. Het is wel een meerwaarde om het eventueel in combinatie te gebruiken met een andere tool indien er andere services gebruikt worden van AWS.

Qualys zal iets lastiger zijn om de te updaten omdat de image op een eigen image registry moet staan. De AWS-omgeving beveiligen is zeker een pluspunt. Daarnaast is er wel een overzichtelijk maar verouderd dashboard voor zowel de AWS-omgeving als de container security. Multi Cloud is ook een pluspunt bij Qualys. Voor de prijs van Qualys is dit geen goede optie.

Prisma Cloud is voor Security Tools de winnaar. Het heeft een goede documentatie en is eenvoudig te configureren. Het heeft veel functionaliteiten met visueel sterke dashboards en overzichten. Het is ook Multi Cloud. De prijs van Prisma Cloud is hoog maar je krijgt veel functionaliteiten en een goede support van Prisma Cloud zelf. Prisma Cloud zal verder gebruikt worden voor de automatisatie van de Cloud omgeving.

Monitoring Tools

Cloudwatch heeft interessante functionaliteiten, maar heeft wel enkele beperkingen. Er moet voor elke resource een IAM-rol toegekend worden aan deze resource om gemonitord te kunnen worden. Cloudwatch is ook niet multi cloud omdat het AWS-native is.

New Relic heeft ook interessante functionaliteiten maar volgens Gartner loopt New Relic achter op de markt. Hierdoor wordt New Relic niet verder gebruikt.

Datadog heeft een uitstekende tool waarbij er veel metrics beschikbaar zijn. Tevens is het een visueel sterke tool en het is heel eenvoudig in gebruik. Datadog kan ook gebruikt worden voor de andere cloud providers zoals Azure of Google Cloud Platform. Datadog zal verder gebruikt worden voor de automatisatie van de Cloud omgeving.

Logging Tools

Cloudwatch heeft niet veel functionaliteiten om de logs op een praktische manier weer te geven. Wel heeft Cloudwatch de mogelijkheid om de master node van de Kubernetes cluster te loggen. Indien gewenst kan Cloudwatch gebruikt worden in combinatie met een andere tool.

De ELK stack heeft niet voldoende functionaliteiten en is niet praktisch in gebruik voor deze bachelorproef.

Datadog heeft in vergelijking met Cloudwatch en ELK Stack meer functionaliteiten en de tool is overzichtelijker opgebouwd. Er kan meer gefilterd worden en het heeft betere proactieve monitors. Tevens wordt Datadog ook gebruikt als monitoring tool en kunnen er metrics opgevraagd worden bij een log. Datadog zal verder gebruikt worden voor de automatisatie van de Cloud omgeving.

Automatisatie Tools

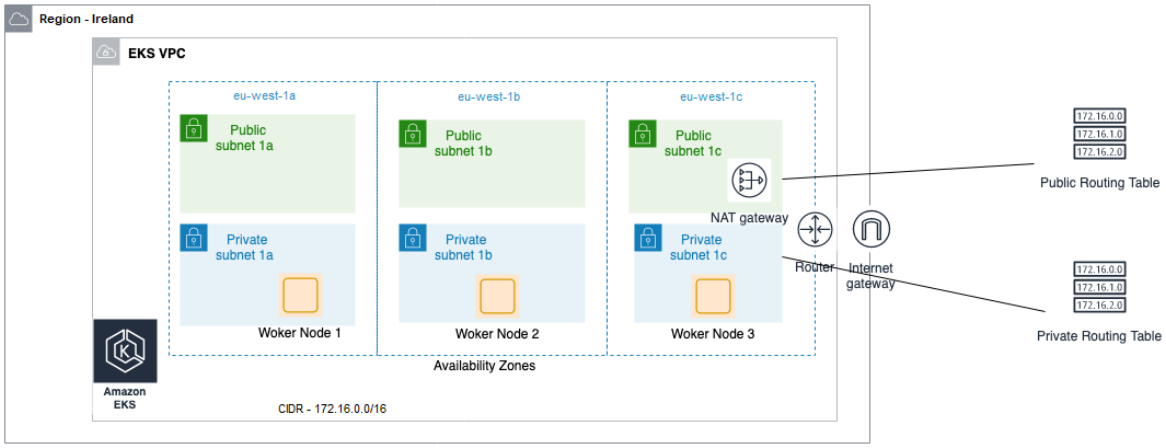
Cloudformation is AWS-native en wordt gebruikt door verschillende andere tools. EKS heeft veel dependencies om dit script op een eenvoudige manier te schrijven. Een ander nadeel is dat de geschreven code enkel op AWS kan gebruikt worden.

Terraform is multi cloud en heeft een eenvoudige structuur. Er kunnen verschillende onderdelen van de cluster in verschillende bestanden opgeslagen worden zodat de code overzichtelijk blijft. Terraform heeft ook een zeer goede documentatie waardoor het eenvoudig is om extra modules toe te voegen.

9 Automatisatie van de cloud omgeving

De beginsituatie van de opdracht is een lege AWS-omgeving. Het is de bedoeling dat er een geautomatiseerde uitrol gedaan wordt van de Kubernetes cluster en de bijhorende operationele tooling om deze cluster te onderhouden. Voor de security tool is gekozen voor Prisma Cloud en voor de monitoring en logging tool is gekozen voor Datadog.

9.1 Schema cloud omgeving



Figuur 37: Schema cloud omgeving

Het bovenstaande schema bevat de belangrijkste elementen van de cloud omgeving. Er wordt gebruik gemaakt van de regio 'Ireland' of 'eu-west-1'. Verder wordt er op deze region een VPC gemaakt genaamd 'EKS'. In deze VPC zal de Kubernetes master node en de worker nodes terecht komen. De master node wordt beheerd door AWS en zal redundant over de verschillende availability zones beschikbaar zijn. De worker nodes zullen elk op een verschillende availability zone geplaatst worden. Deze worker nodes worden ook in het private subnet van elke zone geplaatst. Tevens is er ook een Internet & NAT gateway aanwezig voor de publieke en private communicatie van de Kubernetes cluster.

9.2 Uitrol van de Kubernetes cluster

Om bovenstaand schema uit te rollen is er gekozen voor Terraform. Er is een Terraform script geschreven dat opgedeeld is in 4 bestanden genaamd backend.tf, var.tf, provider.tf en modules.tf. Deze 4 bestanden rollen een Kubernetes cluster uit op de AWS region 'eu-west-1'.

- Backend.tf

```
terraform {
    backend "s3" {
        bucket         = "brbr-azure"
        encrypt        = false
        key            = "tf/terraform.tfstate"
        region         = "eu-west-1"
    }
}
```

Codefragment 2: Terraform: backend.tf

Met het backend.tf bestand zal de Terraform state file van de infrastructuur bewaard worden in een S3 bucket 'brbr-azure' op de region 'eu-west-1'. De state file is de status van de infrastructuur op AWS op dit moment. Het houdt bij welke verschillende resources er al bestaan op AWS. Hierdoor kan er makkelijk iets toegevoegd worden bij een Terraform script zonder de volledige infrastructuur opnieuw te moeten opzetten.

- Var.tf

```
data "aws_eks_cluster" "cluster" {
  name = module.eks.cluster_id
}

data "aws_eks_cluster_auth" "cluster" {
  name = module.eks.cluster_id
}

data "aws_availability_zones" "available" {
  state = "available"
}

locals {
  cluster_name = "BrBr-k8s-cluster"
}
```

Codefragment 3: Terraform: var.tf

Met de var.tf worden de variabelen van het Terraform script bewaard. De eerste 2 variabelen worden aan het cluster id gekoppeld, de 3e variabele wordt aan de beschikbare zones gekoppeld van eu-west-1 en de 4e variabele is de clusternaam.

- Provider.tf

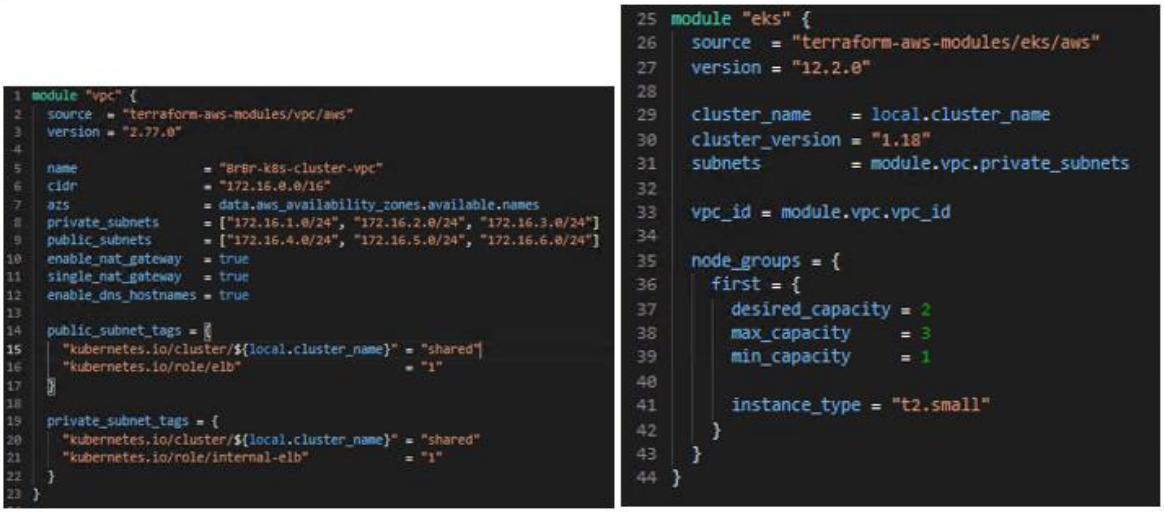
```
provider "aws" {
  region = "eu-west-1"
}

provider "kubernetes" {
  host                   = data.aws_eks_cluster.cluster.endpoint
  cluster_ca_certificate = base64decode(data.aws_eks_cluster.cluster.certificate_authority.0.data)
  token                  = data.aws_eks_cluster_auth.cluster.token
}
```

Codefragment 4: Terraform: provider.tf

Er zijn in het Terraform script 2 providers gedefinieerd. Een eerste provider is de 'AWS-provider. Hierbij moet enkel de regio aangegeven worden. In dit voorbeeld wordt eu-west-1 gebruikt. De tweede provider is 'Kubernetes'. In de Kubernetes provider zijn enkele parameters beschikbaar voor de authenticatie met de cluster.

- Modules.tf



```
module "vpc" {
  source  = "terraform-aws-modules/vpc/aws"
  version = "2.77.0"

  name                 = "BrBr-k8s-cluster-vpc"
  cidr                 = "172.16.0.0/16"
  azs                  = data.aws_availability_zones.available.names
  private_subnets      = ["172.16.1.0/24", "172.16.2.0/24", "172.16.3.0/24"]
  public_subnets       = ["172.16.4.0/24", "172.16.5.0/24", "172.16.6.0/24"]
  enable_nat_gateway   = true
  single_nat_gateway   = true
  enable_dns_hostnames = true

  public_subnet_tags = {
    "kubernetes.io/cluster/${local.cluster_name}" = "shared"
    "kubernetes.io/role/elb"                      = "1"
  }

  private_subnet_tags = {
    "kubernetes.io/cluster/${local.cluster_name}" = "shared"
    "kubernetes.io/role/internal-elb"             = "1"
  }
}
```

```
module "eks" {
  source  = "terraform-aws-modules/eks/aws"
  version = "12.2.0"

  cluster_name    = local.cluster_name
  cluster_version = "1.18"
  subnets         = module.vpc.private_subnets

  vpc_id = module.vpc.vpc_id

  node_groups = {
    first = {
      desired_capacity = 2
      max_capacity     = 3
      min_capacity     = 1

      instance_type = "t2.small"
    }
  }
}
```

Codefragment 5: Terraform: modules.tf

In het modules.tf bestand staan 2 modules geschreven. De eerste module in het bestand is VPC. In de VPC-module is de naam van de module ingevuld. Daarnaast is er ook een CIDR ingesteld. Deze waarde geeft aan uit welke range de IP-adressen komen. Ook de availability zones die actief zijn op dit moment worden als waarde meegegeven. Ook de private en publieke subnetten worden specifiek gedefinieerd. Daarnaast zijn er nog 2 tags voor de private en publieke subnetten. Deze tags zorgen ervoor dat het publieke of private subnet de juiste cluster en load balancer toekend. Dit is nodig omdat AWS anders een verkeerde routering zou doen.

De tweede module is EKS. De module EKS zorgt voor de Kubernetes master node van de infrastructuur. Deze module heeft een clusternaam, versie, VPC en private subnetten van het VPC nodig. Daarnaast wordt er in dit Terraform script gewerkt met een node group in deze module. De node group zal ervoor zorgen dat er minimum 1 worker node en maximum 3 worker nodes beschikbaar zijn. De node group zal er ook voor zorgen dat het 2 worker online kan houden. Deze worker nodes zijn type ‘t2.small’. Deze instances hebben 1 vCPU, 2GB memory en 20GB EBS-opslag.

Het Terraform script wordt op Azure DevOps geplaatst in de repository ‘AWS Infra’. Deze repository wordt later gebruikt om de infrastructuur automatisch te laten uitrollen.

9.3 Integratie van de tools

Prisma Cloud

Prisma Cloud wordt geïntegreerd met een daemonset van de Prisma Cloud defender. De daemonset wordt afgehaald van de Prisma Cloud website en wordt op Azure DevOps geplaatst in de repository 'AWS Kubernetes'. Deze repository wordt later gebruikt om de daemonset geautomatiseerd toe te passen op de Kubernetes cluster. Het toepassen van de daemonset gebeurt pas na het aanmaken van een namespace 'Twistlock'. Met onderstaande commando's wordt Prisma Cloud op de cluster geïntegreerd. Dit wordt later gebruikt in de pipeline.

```
Kubectl create namespace twistlock
kubectl apply -f "/home/vsts/work/r1/a/_AWS Kubernetes -
Main/prismacloud_daemonset.yaml"
```

Datadog

Datadog wordt geïntegreerd met Helm charts. Om deze Helm charts te bekomen moeten er 2 Helm repositories toegevoegd worden aan Helm.

```
helm repo add datadog https://helm.datadoghq.com
helm repo add stable https://charts.helm.sh/stable
Helm repo update
```

Tevens is er ook een datadog-values.yaml bestand die als parameter moet meegeven worden bij de Helm charts. Dit bestand is gedownload van de Datadog website en heeft enkele wijzigingen:

- **Clustername** : BrBr-k8s-cluster
- **Logging** : False -> True
- **Process collection** : False -> True
- **DNS Stats** : False -> True
- **Network Monitor** : False - > True

Opnieuw zal de Datadog agent uitgerold worden in een aparte namespace. Deze namespace noemt 'Datadog'. Het bestand zal ook opgeslagen worden in de repository 'AWS Kubernetes' op Azure DevOps. Dit zal later gebruikt worden om te automatiseren. In het commando zal het datadog-values.yaml bestand meegegeven worden, de locatie van Datadog (EU of US), de API-key van Datadog en de namespace waarin de agent mag gedeployt worden. Onderstaand commando wordt gebruikt om Datadog te integreren.

```
Helm install datadog-agent -f "/home/vsts/work/r1/a/_AWS Kubernetes - Main/datadog-
values.yaml" --set datadog.site='datadoghq.eu' --set datadog.apiKey=XXX --namespace
datadog datadog/datadog
```

9.4 Azure DevOps

Repositories

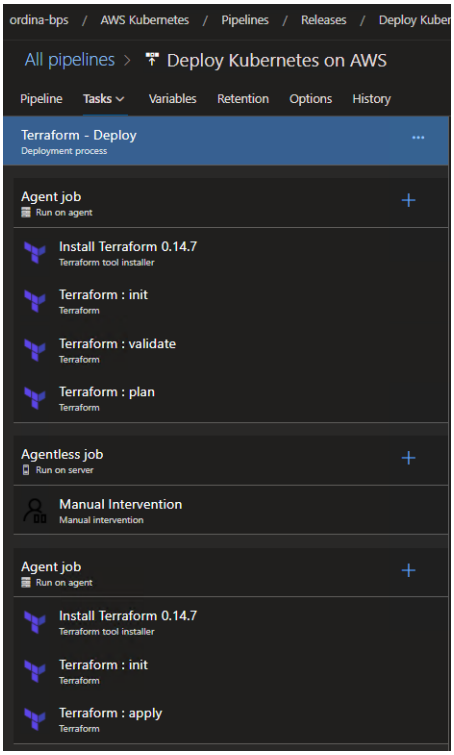
Zoals in bovenstaande hoofdstukken al aangehaald is zijn er 2 repositories aanwezig om de automatisatie te realiseren. Het voordeel van deze repositories is bij wijzigingen aan één van de twee repositories zal de automatisatiepipeline getriggerd worden die de wijzigingen automatisch toepast op de cluster of op de configuratie van de tools.

Pipelines

De automatisatie van de cloud omgeving en de integratie van de tools is opgesplitst in 2 pipelines. Dit is noodzakelijk omdat de cloud omgeving eerst moet opgezet zijn vooraleer de tools geïntegreerd kunnen worden.

Pipeline 1: Deploy Kubernetes on AWS

De eerste pipeline is het uitrollen van Kubernetes op de AWS-omgeving. Hiervoor wordt de repository ‘AWS Infra’ gebruikt in combinatie met de tool ‘Terraform’.



Figuur 38: Azure DevOps: Deploy Kubernetes on AWS - Pipeline

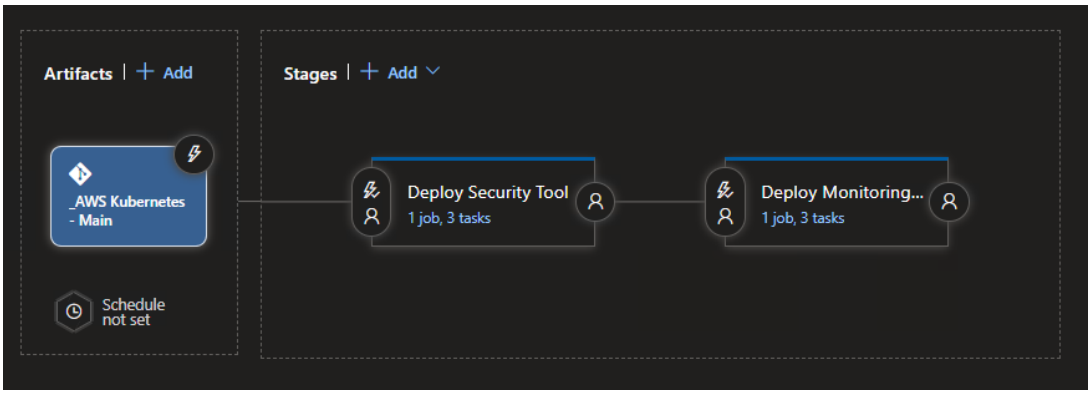
De stage van de pipeline is opgedeeld in 3 stappen. Een eerste stap is het installeren van Terraform. Na het installeren wordt Terraform geïnitieerd met de AWS Infra repository die gelinkt is aan de pipeline. Hierbij worden de AWS-credentials meegegeven. Nadien worden de bestanden in de AWS Infra repository gevalideerd. Als de bestanden gevalideerd zijn wordt er een Terraform plan uitgevoerd op deze repository met AWS-credentials. Dit Terraform plan zal alle wijzigingen weergeven die op de AWS-omgeving uitgevoerd gaan worden.

Nadat de Terraform plan aangemaakt is zal de stage tijdelijk gepauzeerd worden. Dit wordt gerealiseerd met de 'Manual intervention' blok. Bij dit blok wordt er gevraagd aan een gebruiker om een Terraform plan na te kijken en goed te keuren zodat de stage kan verdergaan. Indien het plan niet goed is zal de pipeline falen.

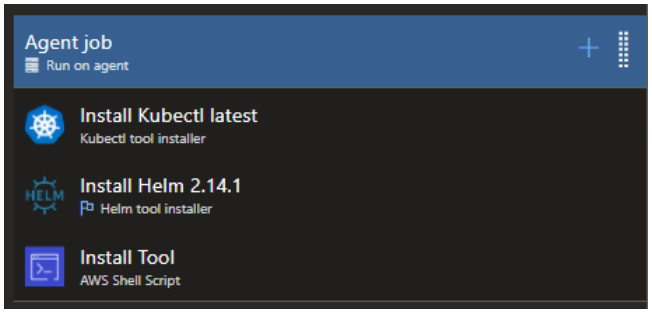
Als er goedkeuring gegeven is zal de stage de derde stap uitvoeren. Hierbij wordt opnieuw Terraform geïnstalleerd. Nadien wordt Terraform opnieuw geïnitieerd met de AWS Infra repository en de AWS-credentials. De laatste stap van de stage is een Terraform apply waarbij de bestanden op de AWS Infra repository toegepast zullen worden op de AWS-omgeving.

Pipeline 2: Deploy tools on Kubernetes

Als de eerste pipeline geslaagd is kan de tweede pipeline gestart worden. Deze pipeline gebruikt de AWS Kubernetes repository met de Prisma Cloud daemonset en het datadog-values.yaml bestand.



Figuur 39: Azure DevOps: Deploy tools on Kubernetes - Stages



Figuur 40: Azure DevOps: Deploy tools on Kubernetes – Pipeline

In figuur 39 zijn er twee stages te zien. Voor elke tool is een verschillende stage aangemaakt. Deze twee stages zijn gelijk opgebouwd zoals te zien is op figuur 40. De stage is opgebouwd met 3 stappen. Een eerste stap is het installeren van Kubectl. Een tweede stap is het installeren van Helm. De derde en laatste stap is het integreren van de tool met een AWS Shell Script. Aan het AWS Shell Script zijn AWS-credentials verbonden zodat de integratie met dezelfde account gebeurt zoals het uitrollen van de omgeving.

- AWS Shell Script: **Install IAM-Authenticator**

Om ervoor te zorgen dat het AWS Shell Script kan communiceren met de Kubernetes cluster moet de AWS-IAM-Authenticator geïnstalleerd worden op de agent. Deze Authenticator is nodig om in de volgende stappen de 'kubeconfig' op te halen van de AWS-omgeving en om de communicatie met de Kubernetes cluster op te zetten. Het integreren van de AWS-IAM-Authenticator wordt met onderstaand script gedaan.

```
# Install IAM-Authenticator
curl -o aws-iam-authenticator https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/aws-iam-authenticator
chmod +x ./aws-iam-authenticator
mkdir -p $HOME/bin && cp ./aws-iam-authenticator $HOME/bin/aws-iam-authenticator && export PATH=$PATH:$HOME/bin
echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
```

- AWS Shell Script: **Initialize KUBECONFIG**

Nadat de AWS-IAM-Authenticator geïnstalleerd is op de agent kan de kubeconfig afgehaald worden van AWS. Deze kubeconfig bevat de authenticatie keys om te communiceren met de cluster. Om de kubeconfig af te halen is de regio en de naam van de Kubernetes cluster nodig. Nadat onderstaand commando uitgevoerd is zal het AWS Shell Script ingesteld zijn op deze Kubernetes cluster.

```
# Initialize KUBECONFIG
aws eks --region eu-west-1 update-kubeconfig --name BrBr-k8s-cluster
```

- AWS Shell Script: **Command K8S - Install Prisma cloud**

Deze stap is de laatste voor de integratie van Prisma Cloud (Stage 1). Met onderstaande stap wordt er een 'twistlock' namespace aangemaakt en wordt de daemonset geïmplementeerd. Deze daemonset staat op de AWS Kubernetes repository. Na enkele seconden zal de tool actief zijn op de Kubernetes cluster.

```
kubectl create namespace twistlock
kubectl apply -f "/home/vsts/work/r1/a/_AWS Kubernetes - Main/prismacloud_daemonset.yaml"
```

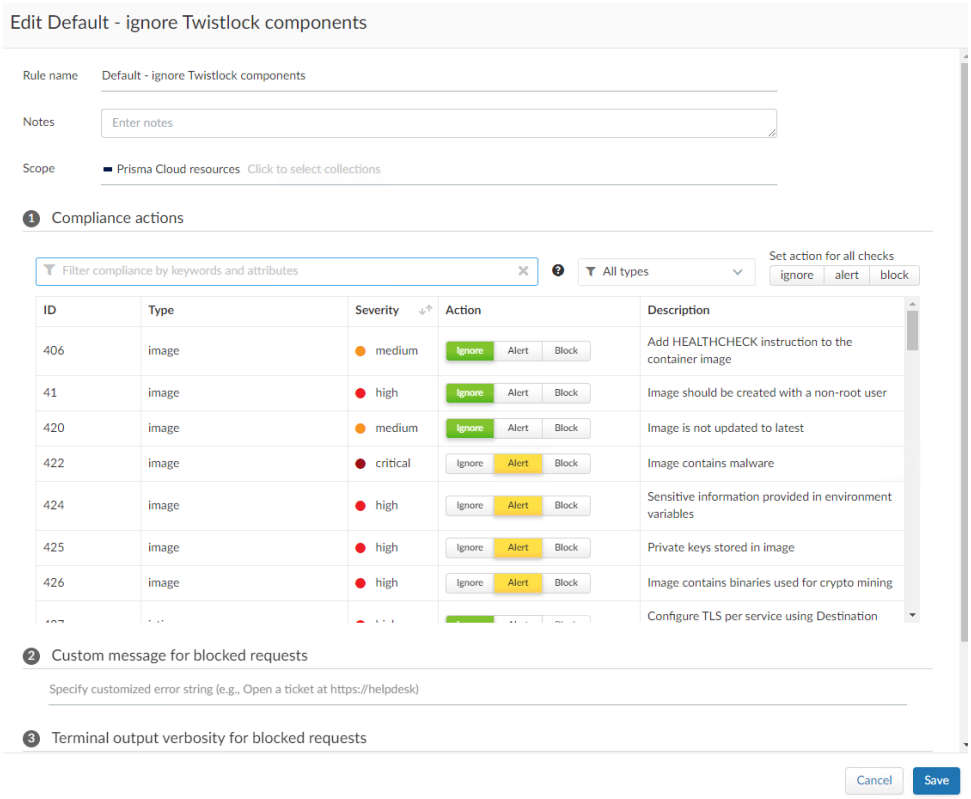
- AWS Shell Script: **Command K8S - Install Datadog**

Deze stap is de laatste stap voor de integratie van Datadog (Stage 2). Bij deze stap zal er eerst een namespace aangemaakt worden genaamd datadog. Nadien zullen 2 Helm repositories toegevoegd worden en zal Helm geüpdatet worden zodat deze herkend kunnen worden. Na het updaten van Helm kan de datadog-agent geïntegreerd worden op de Kubernetes cluster. Datadog zal na enkele seconden beschikbaar worden op de cluster.

```
kubectl create namespace datadog
helm repo add datadog https://helm.datadoghq.com
helm repo add stable https://charts.helm.sh/stable
helm repo update
helm install datadog-agent -f "/home/vsts/work/r1/a/_AWS Kubernetes - Main/datadog-values.yaml" --set datadog.site='datadoghq.eu' --set datadog.apiKey=XXX --namespace datadog datadog/datadog
```

9.5 Prisma Cloud

Deze tool is operationeel zonder extra configuratie. Het gedrag van de tool Prisma Cloud is standaard alert-only. Hierdoor zal er niets geblokkeerd worden.

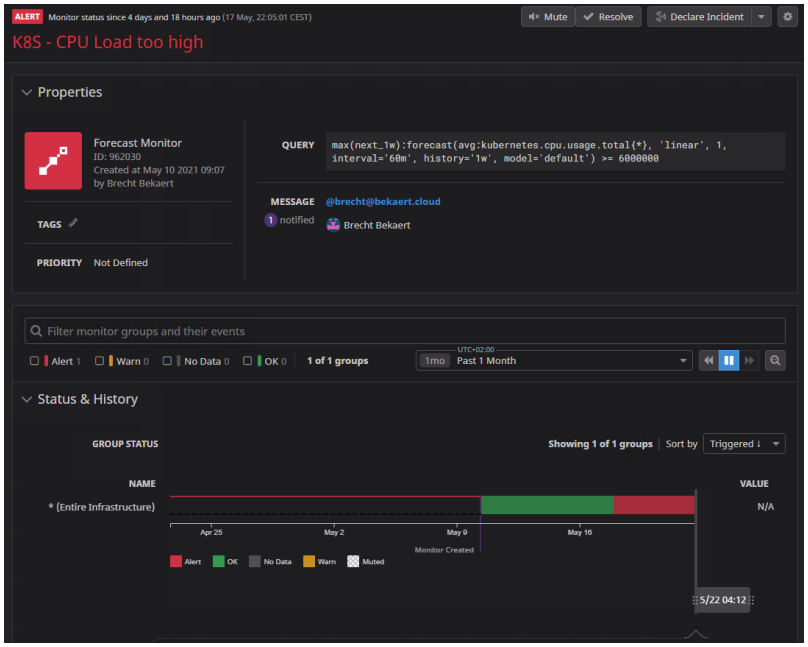


Figuur 41: Prisma Cloud: Lijst met vulnerabilities en gedrag van de tool

Er zijn uiteraard wel opties om bepaalde vulnerabilities of compliance regels te bewerken zodat dit wel geblokkeerd wordt. Verder in de tool is het ook mogelijk om bepaalde images uit te sluiten voor acties bij vulnerabilities. Een voorbeeld van de actielijst is te zien op figuur 41.

9.6 Datadog

Als de tool Datadog geïntegreerd is door Azure DevOps zal de datadog agent op de Kubernetes cluster metrics doorsturen naar Datadog. Om ervoor te zorgen dat er alerts uitgestuurd worden bij mogelijke problemen moet dit ingesteld worden in de monitor tab. Na het instellen van de monitor zal er een alert verstuurd worden als een bepaalde metric overschreden wordt. In figuur 42 is er een monitor ‘CPU-load too high’ die in alarm staat.



Figuur 42: Datadog: Monitor in alarm

Besluit

Voor de security tools is gebleken dat AWS GuardDuty niet geschikt is als security tool. Na verder onderzoek naar third-party tools is Prisma Cloud gekozen als security tool. Prisma Cloud heeft een eenvoudige configuratie. Daarnaast heeft Prisma Cloud verschillende functionaliteiten dat het beveiligen van een Kubernetes cluster heel makkelijk maakt. De container security is top op de markt door de overname van Twistlock enkele jaren geleden. De prijs is subscription-based.

Voor de monitoring tool is gebleken dat AWS Cloudwatch een eenvoudige configuratie heeft maar de functionaliteiten zijn redelijk beperkt. Na verder onderzoek is Datadog gekozen als monitoring tool. Datadog heeft een eenvoudige configuratie. Datadog heeft ook verschillende functionaliteiten waarbij de beste functionaliteit de proactieve monitors zijn. De prijs is subscription-based.

Verder is bij de logging tool gebleken dat AWS Cloudwatch beperkte functionaliteiten heeft. Wel heeft het een functionaliteit om de master node van de Kubernetes cluster te loggen. Na verder onderzoek is Datadog gekozen als logging tool. Datadog heeft een heel overzichtelijke interface en werkt in combinatie met metrics (Omdat Datadog is gekozen is voor monitoring). De prijs is ook subscription-based.

Voor de automatisatie is gekozen voor Terraform. Cloudwatch is in vergelijking met Terraform te uitgebreid voor deze use-case. Terraform is ook multi cloud waardoor de infrastructuur heel eenvoudig kan verplaatst worden van cloud provider.

Verder is Azure DevOps gebruikt voor het automatiseren van de Kubernetes cluster en het integreren van de tools. Azure DevOps is gebruikt op vraag van Ordina.

Bibliografie

[1] „Over Ordina,” [Online]. Available: https://www.ordina.be/over/. [Geopend 14 2 2021].

[2] „Ahead Of Change,” [Online]. Available: https://www.ordina.be/over/ahead-of-change/. [Geopend 15 02 2021].

[3] „Kernwaarden Ordina,” [Online]. Available: https://www.ordina.be/kernwaarden/. [Geopend 15 02 2021].

[4] „Digital Workplace,” Ordina, [Online]. Available: https://www.ordina.be/diensten/digital-workplace/. [Geopend 15 02 2021].

[5] „Robotic Process Automation,” Ordina, [Online]. Available: https://www.ordina.be/diensten/robotic-process-automation/. [Geopend 15 02 2021].

[6] „SAP Services,” Ordina, [Online]. Available: https://www.ordina.be/diensten/sap-services/. [Geopend 15 02 2021].

[7] „Hybrid cloud & IT operations,” Ordina, [Online]. Available: https://www.ordina.be/diensten/infrastructuur/. [Geopend 15 02 2021].

[8] „Wat is amazon web services,” [Online]. Available: https://www.dataweb.nl/wat-is-amazon-web-services/. [Geopend 03 07 2021].

[9] „Amazon EKS,” [Online]. Available: https://aws.amazon.com/ec2/?ec2-whats-new.sort-by=item.additionalFields.postDateTime&ec2-whats-new.sort-order=desc. [Geopend 01 03 2021].

[10] „Amazon EC2,” [Online]. Available: https://aws.amazon.com/ec2/?ec2-whats-new.sort-by=item.additionalFields.postDateTime&ec2-whats-new.sort-order=desc. [Geopend 01 03 2024].

[11] „Amazon S3,” [Online]. Available: https://aws.amazon.com/s3/. [Geopend 01 03 2021].

[12] „Kubernetes,” [Online]. Available: https://kubernetes.io/. [Geopend 16 03 2021].

[13] „Concepts | Kubernetes,” [Online]. Available: https://kubernetes.io/docs/concepts/. [Geopend 01 03 2021].

[14] „Kubernetes Components,” [Online]. Available: https://kubernetes.io/docs/concepts/overview/components/. [Geopend 1 03 2021].

[15] „Understanding Kubernetes Architecture,” [Online]. Available: https://geekflare.com/kubernetes-architecture/. [Geopend 1 03 2021].

[16] „Helm,” [Online]. Available: https://helm.sh. [Geopend 02 03 2021].

[17] „Gartner Magic Quadrants," [Online]. Available: https://www.gartner.com/en/research/methodologies/magic-quadrants-research. [Geopend 10 02 2021].

[18] „AWS GuardDuty," [Online]. Available: https://aws.amazon.com/guardduty/. [Geopend 11 02 2021].

[19] „AlertLogic - About Us," [Online]. Available: https://www.alertlogic.com/company/about-us/. [Geopend 09 02 2021].

[20] „AlertLogic - Container Security," [Online]. Available: https://www.alertlogic.com/why-alert-logic/comprehensive-coverage/containers/. [Geopend 09 02 2021].

[21] „Qualys - About Us," [Online]. Available: https://www.qualys.com/company/. [Geopend 11 02 2021].

[22] „Qualys Cloud Platform," [Online]. Available: https://www.qualys.com/cloud-platform/. [Geopend 12 02 2021].

[23] „Palo Alto Networks - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Palo_Alto_Networks. [Geopend 12 02 2021].

[24] „Amazon Cloudwatch - Application Monitoring," [Online]. Available: https://aws.amazon.com/cloudwatch/. [Geopend 13 02 2021].

[25] „New Relic - About," [Online]. Available: https://newrelic.com/about. [Geopend 13 02 2021].

[26] „New Relic One," [Online]. Available: https://newrelic.com/platform. [Geopend 13 02 2021].

[27] „Cloud Monitoring as a Service - Datadog," [Online]. Available: https://www.datadoghq.com/. [Geopend 15 02 2021].

[28] „Elastic Enterprise Search," [Online]. Available: https://www.elastic.co/enterprise-search. [Geopend 21 02 2021].

[29] „Elastic Stack - A brief Introduction," [Online]. Available: https://hackernoon.com/elastic-stack-a-brief-introduction-794bc7ff7d4f. [Geopend 21 02 2021].

[30] „Cloudformation Features," [Online]. Available: https://aws.amazon.com/cloudformation/features/. [Geopend 17 02 2021].

[31] „Terraform by HashiCorp," [Online]. Available: https://www.terraform.io/. [Geopend 17 02 2021].

[32] „HashiCorp Pricings," [Online]. Available: https://www.hashicorp.com/products/terraform/pricing. [Geopend 17 02 2021].

[33] „Azure DevOps," [Online]. Available: https://azure.microsoft.com/en-us/overview/. [Geopend 01 04 2021].

[34] „Stackify Retrace,” [Online]. Available: https://stackify.com/. [Geopend 20 02 2021].

Bijlagen

Bijlage 1A: Actieplan

Bijlage 1B: Actieplan

Odisee
DE CO-HOGESCHOOL

Bachelorproef

Studiegebied Industriële Wetenschappen en Technologie

Brecht Bekaert
Professionele Bachelor Elektronica - ICT
Afstudeerrichting ICT
Academiejaar 2020/2021

Implementatie van operationale tooling voor een kubernetes cluster in AWS

Bijlagen

Ordina Belgium
Blarenberglaan 3 bus B
2800 Mechelen
België

Bijlage 1A: Actieplan

odisee	**Odisee** **Studiegebied IWT** **Opleiding Electronica-ICT** 2019-2020	**Technologiecampus Gent** Gebr. Desmetstraat 1 9000 GENT Tel.: (09) 265 86 10 Fax: (09) 225 62 69

ACTIEPLAN (vereisten op pg. 2)

Student(e): Brecht Bekaert — Groep: ICT

Stageplaats: Ordina Belgium

Stageleid(st)er (interne promotor): Serge Fabre

Stagementor (externe promotor): Geert Clissen

Stap	Inhoud	Streef-datum	Werkelijke datum	Opvolging
1	Marktonderzoek naar tools voor security en vergelijken van de tools - Functionaliteiten: Advies over status van systeem	10/02		
2	Marktonderzoek naar tools voor monitoring en vergelijken van de tools - Functionaliteiten: Proactief monitoring	16/02		
3	Marktonderzoek naar tools voor automation en vergelijken van de tools - Functionaliteiten - Integratie met andere platformen	18/02		
4	Marktonderzoek naar tools voor logging/auditing en vergelijken van de tools - Functionaliteiten - Wie is aangemeld, wie heeft welke container opgestart?	23/02		

Bijlage 1B: Actieplan

5	Samenvatting marktonderzoeken security, monitoring, logging en automation	25/02			
6	Onderzoek naar Kubernetes en EKS - Wat is Kubernetes? - Hoe opzetten? - Functionaliteiten	02/03			
7	Testen en experimenteren met EKS in testomgeving.	04/03			
8	Testen en experimenteren met security tools. - Functionaliteiten - Welke tool werkt het best?	9/03			
9	Testen en experimenteren met monitoring tools. - Functionaliteiten - Welke tool werkt het best?	23/03			
10	Testen en experimenteren met logging/auditing tools. - Functionaliteiten - Welke tool werkt het best?	01/04			
	Paasvakantie 02/04 -> 11/04				
11	Testen en experimenteren met automation tools. - Functionaliteiten - Welke tool werkt het best?	22/04			
12	Automatisatiescript om omgeving op te zetten.	05/05			