



# Innovatie in IoT

## Verslag

**Bachelor in de Elektronica-ICT**  
**keuzerichting Digital Innovation**

**Hanne Verschueren**

Academiejaar 2022-2023

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

<b>1</b>	<b>FIGUURLIJST .....</b>	<b>4</b>
<b>2</b>	<b>INLEIDING .....</b>	<b>5</b>
<b>3</b>	<b>HET STAGEBEDRIJF .....</b>	<b>6</b>
<b>4</b>	<b>VERDUIDELIJKING TECHNOLOGIEËN/TOOLS.....</b>	<b>7</b>
<b>4.1</b>	<b>Kafka .....</b>	<b>7</b>
<b>4.2</b>	<b>Node-RED.....</b>	<b>8</b>
<b>4.3</b>	<b>Ansible .....</b>	<b>9</b>
<b>4.4</b>	<b>De Elastic Stack.....</b>	<b>10</b>
<b>4.5</b>	<b>Inkscape .....</b>	<b>12</b>
<b>5</b>	<b>PROJECTEN .....</b>	<b>13</b>
<b>5.1</b>	<b>BOA.....</b>	<b>13</b>
5.1.1	Context .....	13
5.1.2	Proces .....	13
5.1.3	Mijn bijdrage .....	15
5.1.4	Conclusie .....	15
<b>5.2</b>	<b>Kafka Magic .....</b>	<b>15</b>
5.2.1	Context .....	15
5.2.2	Proces .....	15
5.2.3	Mijn bijdrage .....	17
5.2.4	Conclusie .....	17
<b>5.3</b>	<b>Smartbox monitoring .....</b>	<b>17</b>
5.3.1	Context .....	17
5.3.2	Proces .....	18
5.3.3	Mijn bijdrage .....	21
5.3.4	Conclusie .....	21
<b>5.4</b>	<b>Stoomketels .....</b>	<b>21</b>
5.4.1	Context .....	21
5.4.2	Proces .....	21
5.4.3	Mijn bijdrage .....	23
5.4.4	Conclusie .....	23
<b>6</b>	<b>CONCLUSIE .....</b>	<b>24</b>
<b>7</b>	<b>DANKWOORD .....</b>	<b>25</b>
<b>8</b>	<b>BRONNENLIJST .....</b>	<b>26</b>

# 1 FIGUURLIJST

Figuur 1 Logo Van Genechten Packaging (Van Genechten Packaging)	6
Figuur 2 Kaart met filialen van VGP (Van Genechten Packaging)	6
Figuur 3 logo Apache Kafka (File:Apache kafka wordtype.svg)	7
Figuur 4 logo Node-RED (Resources)	8
Figuur 5 logo Ansible (File:Ansible logo.svg)	9
Figuur 6 logo van de Elastic Stack (Elastic NV)	10
Figuur 7 logo Inkscape (File:Inkscape Logo.svg)	12
Figuur 8 tv-scherf in fabriek, eindresultaat BOA	13
Figuur 9 dashboard BOA	14
Figuur 10 deel van de Node-RED BOA	14
Figuur 11 Kafka Magic zoeken naar topic "smartboxlogging"	16
Figuur 12 Kafka Magic verfijn zoekopdracht met JavaScript	16
Figuur 13 Kafka Magic voorbeeld van een bericht	17
Figuur 14 stappen die een logfile doorloopt	17
Figuur 15 canvas in Kibana dat duidelijk weergeeft of er smartboxen zonder stroom zijn gevallen	19
Figuur 16 filtering van data op loglevel en service in Kibana discover	19
Figuur 17 dashboard in Kibana	20
Figuur 18 Heartbeat uptime in Kibana	20
Figuur 19 box met de Adam modules die alle signalen binnen neemt	22
Figuur 20 dashboard van de stoomketels	22
Figuur 21 deel van de Node-Red van de stoomketels	23

## 2 INLEIDING

De snelle groei van digitale technologieën heeft geleid tot een steeds grotere vraag naar efficiëntie, automatisering en gegevensanalyse in diverse industrieën. Het vermogen om complexe taken te automatiseren en waardevolle inzichten te verkrijgen uit enorme hoeveelheden gegevens is cruciaal geworden voor het behalen van concurrentievoordeel. In dit tijdperk van technologische vooruitgang heb ik de kans gehad om een stage te volgen waarin ik ben ondergedompeld in enkele veelgebruikte tools: Node-RED, Ansible, Kafka en de Elastic Stack.

Voor mijn stagebedrijf Van Genechten Packaging met 12 filialen verspreid over meerdere landen, is het belang van het bijblijven met de concurrentie en het streven naar operationele efficiëntie duidelijk. Het bedrijf is vastbesloten om gebruik te maken van moderne technologieën, waaronder de eerdergenoemde tools, om betere inzichten te verkrijgen en bepaalde processen te stroomlijnen.

Dit verslag documenteert mijn stage-ervaring en de verworven kennis op het gebied van Node-RED, Ansible, Kafka en de Elastic Stack. Door middel van praktische projecten en uitdagingen heb ik mijn technische vaardigheden verbeterd en een beter begrip ontwikkeld van de toepassingen en mogelijkheden van deze tools. Ik zal beginnen met deze te verduidelijken en vervolgens uitleggen in welke opdrachten ik ze gebruikt heb.

### 3 HET STAGEBEDRIJF



*Figuur 1 Logo Van Genechten Packaging (Van Genechten Packaging)*

Ik heb stagegelopen bij Imas N.V., een dochteronderneming van Van Genechten Packaging.

Van Genechten Packaging is een toonaangevende Europese producent van kartonnen verpakkingen. Het bedrijf is opgericht in 1834 en heeft sindsdien een sterke reputatie opgebouwd in de verpakkingsindustrie.

Met meerdere productielocaties verspreid over Europa, waaronder België, Nederland, Frankrijk, Hongarije en Polen, richt het bedrijf zich op het ontwerpen, produceren en leveren van op maat gemaakte kartonnen verpakkingso oplossingen voor verschillende sectoren, zoals voeding en dranken, cosmetica, farmaceutica, consumentengoederen en industriële producten.

Van Genechten Packaging hecht veel waarde aan duurzaamheid en streeft voortdurend naar het verminderen van de impact van zijn productieprocessen op het milieu. Ze maken gebruik van hernieuwbare en recyclebare materialen en implementeren milieuvriendelijke productietechnieken.

Imas N.V. verzorgt de ICT van Van Genechten Packaging. (Van Genechten Packaging)



*Figuur 2 Kaart met filialen van VGP (Van Genechten Packaging)*

## 4 VERDUIDELIJKING TECHNOLOGIEËN/TOOLS

### 4.1 Kafka



*Figuur 3 logo Apache Kafka (File:Apache kafka wordtype.svg)*

Apache Kafka is een open-source gedistribueerd gegevensstromenplatform dat is ontworpen voor het verwerken van grote hoeveelheden realtime gegevensstromen. Het is ontworpen om te voldoen aan de eisen van moderne applicaties die enorme hoeveelheden gegevens genereren en verwerken, zoals streaminganalyse, realtime monitoring, logverwerking en andere vormen van gegevensintegratie.

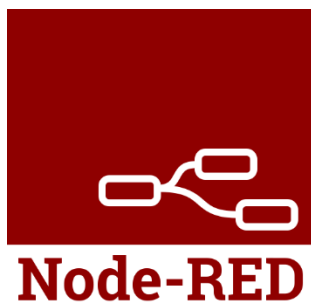
Een Kafka-cluster bestaat uit één of meer servers, ook wel "brokers" genoemd, die verantwoordelijk zijn voor het opslaan en repliceren van de gegevensstromen.

Producers zijn verantwoordelijk voor het genereren en publiceren van records naar specifieke Kafka-topics, terwijl consumers de gegevens van de topics kunnen lezen en verwerken. Kafka biedt een hoog doorvoervermogen en lage latentie, waardoor meerdere producers en consumers gelijktijdig kunnen werken en miljoenen berichten per seconde kunnen worden verwerkt.

Wat Kafka onderscheidt, is zijn mogelijkheid om gegevensstromen te bufferen en te bewaren gedurende een bepaalde periode. Dit stelt consumers in staat om gegevens te lezen en te verwerken op hun eigen tempo, zonder invloed uit te oefenen op de snelheid van de producers. Dit alles maakt Kafka zeer veerkrachtig en zorgt ervoor dat gegevens niet verloren gaan, zelfs als er tijdelijke onderbrekingen of storingen optreden in het systeem.

In het kort biedt Apache Kafka een robuust, schaalbaar en gedistribueerd platform voor het verwerken van realtime gegevensstromen. Het heeft zich bewezen als een waardevol instrument voor organisaties die streven naar realtime gegevensintegratie, streaminganalyse en het bouwen van gegevens gestuurde applicaties. (Kafka)

## 4.2 Node-RED



*Figuur 4 logo Node-RED (Resources)*

Node-RED is een open-source visuele programmeeromgeving waarmee gebruikers op een intuïtieve manier de automatisering van taken en het verbinden van apparaten en diensten kunnen ontwerpen en implementeren.

Node-RED maakt gebruik van een browser gebaseerde interface die het mogelijk maakt om logische "flows" te bouwen door middel van het slepen en neerzetten van vooraf gedefinieerde functionele blokken, die "nodes" worden genoemd. Deze nodes vertegenwoordigen verschillende functionaliteiten, zoals gegevensinvoer, -verwerking en -uitvoer, en kunnen met elkaar worden verbonden om de gewenste workflows te creëren.

De kracht van Node-RED ligt in de mogelijkheid om gegevens te verzamelen, te transformeren en door te sturen tussen verschillende apparaten, diensten en systemen. Het ondersteunt een breed scala aan protocollen en integraties, waaronder HTTP, MQTT, TCP/IP, databases, cloudplatforms en IoT-apparaten. Hierdoor kunnen gebruikers snel en eenvoudig geautomatiseerde workflows ontwerpen die informatie uit verschillende bronnen kunnen verzamelen, verwerken en weergeven.

Node-RED biedt ook een uitgebreide bibliotheek van nodes, waaronder zowel de basisfunctionaliteit als specifieke integraties met populaire platforms en diensten. Bovendien kunnen gebruikers hun eigen aangepaste nodes maken en delen met de gemeenschap, waardoor de functionaliteit van Node-RED verder kan worden uitgebreid.

De visuele interface van Node-RED maakt het toegankelijk voor zowel ervaren ontwikkelaars als niet-technische gebruikers. Het stelt gebruikers in staat om complexe workflows te ontwerpen zonder uitgebreide programmeerkennis, terwijl het nog steeds de flexibiliteit biedt om aangepaste logica toe te voegen met behulp van JavaScript.

Node-RED kan worden toegepast in verschillende domeinen, zoals IoT (Internet of Things), home automation, industriële automatisering, dataverwerking en integratie, en meer. Het biedt een intuïtieve en flexibele manier om geautomatiseerde taken en processen te creëren en te beheren, waardoor het een populaire keuze is geworden voor zowel hobbyisten als professionele ontwikkelaars.

Kortom, Node-RED is een krachtig visueel programmeerplatform waarmee gebruikers geautomatiseerde workflows kunnen ontwerpen en implementeren door middel van het verbinden van vooraf gedefinieerde functionele blokken. Met zijn eenvoudige interface, uitgebreide bibliotheek van nodes en brede

ondersteuning voor integraties, stelt Node-RED gebruikers in staat om snel en efficiënt gegevens te verzamelen, te verwerken en te beheren in diverse toepassingsgebieden. (Node-RED)

### 4.3 Ansible



ANSIBLE

*Figuur 5 logo Ansible (File:Ansible logo.svg)*

Ansible is een open-source IT-automatiseringstool die is ontworpen voor het vereenvoudigen van de implementatie, configuratie en beheer van computersystemen. Het biedt een eenvoudige, op taal gebaseerde syntax en een agentloze architectuur.

In plaats van te vertrouwen op complexe scripts of aangepaste code, maakt Ansible gebruik van zogenaamde "playbooks" om automatiseringstaken uit te voeren. Een playbook is een bestand dat een reeks instructies bevat die beschrijven hoe een systeem moet worden geconfigureerd en beheerd. Deze instructies worden geschreven in YAML, wat een eenvoudig te begrijpen en te lezen syntax biedt.

Met Ansible kunnen systeembeheerders en ontwikkelaars taken automatiseren, zoals het installeren en configureren van software, het uitrollen van updates, het beheren van gebruikersaccounts en het bewaken van de systeemstatus. Playbooks kunnen ook worden gebruikt voor het coördineren van taken over meerdere servers, waardoor schaalbaarheid en consistentie worden bereikt.

Een van de belangrijkste kenmerken van Ansible is de agentloze architectuur. Dit betekent dat er geen permanente software of agent op de doelsystemen hoeft te worden geïnstalleerd. In plaats daarvan maakt Ansible gebruik van SSH (Secure Shell) en andere beheerprotocollen om op afstand met de doelsystemen te communiceren en de vereiste taken uit te voeren. Dit maakt het eenvoudig om met diverse besturingssystemen en netwerkinfrastructuur te werken zonder extra complexiteit.

Ansible heeft een grote en actieve gemeenschap die bijdraagt aan de ontwikkeling en ondersteuning van de tool. Hierdoor is er een uitgebreide bibliotheek van vooraf gedefinieerde modules beschikbaar die kunnen worden gebruikt om veelvoorkomende taken te automatiseren en tijd te besparen.

Kort samengevat is Ansible een krachtige en flexibele IT-automatiseringstool die systeembeheerders en ontwikkelaars in staat stelt om taken te automatiseren, configuraties te beheren en de efficiëntie van computeromgevingen te vergroten. Met zijn eenvoudige syntax, agentloze architectuur en modulaire structuur biedt Ansible een intuïtieve en gestroomlijnde manier om complexe IT-taken te automatiseren en consistentie te verzekeren. (ansible)



## 4.4 De Elastic Stack



*Figuur 6 logo van de Elastic Stack (Elastic NV)*

De Elastic Stack, voorheen bekend als de ELK Stack, is een set van open-source tools en technologieën die worden gebruikt voor het verzamelen, opslaan, doorzoeken, analyseren en visualiseren van grote hoeveelheden gegevens. Het bestaat uit drie hoofdcomponenten: Elasticsearch, Logstash en Kibana, die samen een end-to-end oplossing bieden voor gegevensverwerking en -analyse.

- **Logstash:** Logstash is een gegevensverzamelingsengine die wordt gebruikt voor het verzamelen, filteren, transformeren en verrijken van gegevens uit verschillende bronnen. Het kan logs, gebeurtenissen en andere gestructureerde en ongestructureerde gegevens verwerken en doorsturen naar Elasticsearch of andere bestemmingen. Logstash biedt een breed scala aan ingebouwde plugins om gegevens uit verschillende bronnen te verzamelen en te verwerken, en kan ook worden uitgebreid met aangepaste plugins om aan specifieke behoeften te voldoen.
- **Elasticsearch:** Elasticsearch is een gedistribueerd zoek- en analyse machine die is ontworpen voor snelle en schaalbare gegevensopslag en -doorzoeking. Het maakt gebruik van een JSON-gebaseerd querytaal en biedt geavanceerde zoekmogelijkheden, zoals full-text zoeken, geavanceerde filteropties en aggregaties. Elasticsearch kan enorme hoeveelheden gegevens indexeren en razendsnel doorzoeken, waardoor het geschikt is voor realtime toepassingen en gegevensanalyse.
- **Kibana:** Kibana is een krachtige tool voor gegevensvisualisatie en -analyse. Het biedt een web gebaseerde interface waarmee gebruikers gegevens uit Elasticsearch kunnen ontdekken, verkennen en visualiseren. Met Kibana kunnen gebruikers dashboards maken, interactieve grafieken en diagrammen genereren, gegevens filteren en aggregaties uitvoeren om inzichten uit de gegevens te halen. Kibana heeft een gebruiksvriendelijke interface en biedt geavanceerde mogelijkheden voor het verkennen en presenteren van gegevens op een visueel aantrekkelijke en begrijpbare manier.

De Elastic Stack kan worden toegepast in verschillende toepassingsgebieden, zoals logbeheer en analyse, beveiligingsanalyse, infrastructuurbewaking, bedrijfsanalyse en meer. Het stelt organisaties in staat om grote hoeveelheden gegevens te verzamelen, te doorzoeken, te analyseren en te visualiseren op een schaalbare en efficiënte manier. Door de combinatie van Elasticsearch, Logstash en Kibana kunnen gebruikers waardevolle inzichten verkrijgen, patronen ontdekken en problemen oplossen op basis van hun gegevens.

Kort samengevat biedt de Elastic Stack een complete oplossing voor gegevensverwerking en -analyse, van het verzamelen en opslaan van gegevens met Elasticsearch en Logstash tot het visualiseren en analyseren van gegevens met Kibana.

Naast Elasticsearch, Logstash en Kibana bevat de Elastic Stack ook een reeks tools genaamd "Beats". Beats zijn lichtgewichtige data-shippers die zijn ontworpen om verschillende soorten gegevens te verzamelen en naar Elasticsearch of Logstash te sturen voor verdere verwerking en analyse.

Beats bieden specifieke functionaliteiten voor het verzamelen van verschillende soorten gegevens. Hieronder leg ik kort de twee beats die ik gebruikt heb uit.

- **Filebeat:** Filebeat wordt gebruikt voor het verzamelen en doorsturen van logbestanden van verschillende bronnen, zoals systeemlogs, applicatielogs of auditlogs. Het kan logs rechtstreeks van bestanden lezen en verzenden naar een output zoals Elasticsearch, Logstash of Kafka.
- **Heartbeat:** Heartbeat is een onderdeel van de Elastic Stack en wordt gebruikt voor het monitoren van de beschikbaarheid en prestaties van services en servers. Het kan periodieke ping verzoeken uitvoeren naar verschillende bronnen, zoals HTTP-eindpunten, databaseservers en netwerkapparaten, om de reactietijden en statussen te controleren. Heartbeat kan waarschuwingen genereren wanneer er onverwachte vertragingen of uitval optreden, waardoor het mogelijk is om proactief problemen op te sporen en de uptime van services te bewaken.

Met de combinatie van Elasticsearch, Logstash, Kibana en Beats biedt de Elastic Stack een complete end-to-end oplossing voor gegevensverzameling, verwerking, analyse en visualisatie. Het stelt gebruikers in staat om waardevolle inzichten uit diverse gegevensbronnen te halen en complexe gegevens workflows te automatiseren en te beheren. (Elastic Stack)

## 4.5 Inkscape



*Figuur 7 logo Inkscape (File:Inkscape Logo.svg)*

Inkscape is een gratis en open-source grafisch ontwerpprogramma dat wordt gebruikt voor het maken en bewerken van vectorafbeeldingen. Het biedt een breed scala aan tools en functies die ontwerpers in staat stellen om complexe illustraties, logo's, diagrammen en andere grafische elementen te maken.

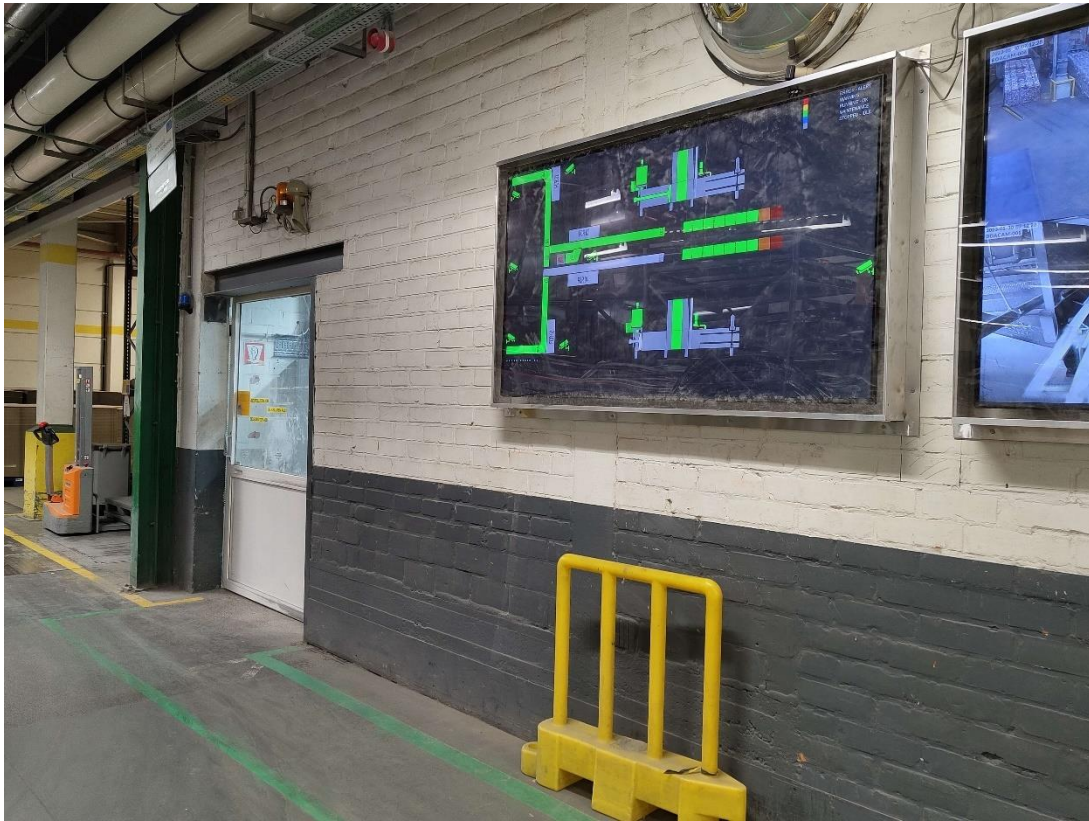
Met Inkscape kunnen gebruikers vormen tekenen, tekst toevoegen, kleuren en verlopen toepassen, objecten groeperen en transformeren, paden bewerken en nog veel meer. Het ondersteunt ook het importeren en exporteren van verschillende bestandsformaten, waaronder SVG (Scalable Vector Graphics), PNG, JPEG en PDF.

Een van de krachtige functies van Inkscape is de mogelijkheid om met vectorafbeeldingen te werken. In tegenstelling tot rasterafbeeldingen, die zijn opgebouwd uit pixels en kunnen vervagen bij vergroting, behouden vectorafbeeldingen hun helderheid en scherpte, ongeacht de grootte. Dit maakt Inkscape zeer geschikt voor grafisch ontwerp, logo-ontwerp, webgraphics en andere toepassingen waarbij schaalbaarheid en precisie belangrijk zijn.

Kortom, Inkscape is een veelzijdig grafisch ontwerpprogramma waarmee gebruikers professionele vectorafbeeldingen kunnen maken en bewerken met behulp van een breed scala aan tools en functies. (Inkscape)

## 5 PROJECTEN

### 5.1 BOA



*Figuur 8 tv-scherm in fabriek, eindresultaat BOA*

#### 5.1.1 Context

Binnen de fabriek bevindt zich een machine genaamd de BOA, die kartonresten samenvoegt tot grote balen die gerecycled kunnen worden. Wanneer de machine blokkeert, moet deze worden stilgelegd. Om de BOA efficiënter te monitoren en sneller in te kunnen grijpen, wordt deze gemonitord met sensoren.

In de fabriek hangt een groot scherm met daarop een tekening die de machine weergeeft. Verschillende delen van de tekening veranderen van kleur afhankelijk van de situatie, of deze goed of slecht is. Het scherm was al ontwikkeld, maar bleek niet volledig accuraat en de code was moeilijk te onderhouden. Het doel was daarom om het scherm opnieuw te ontwerpen op een meer onderhoudsvriendelijke manier.

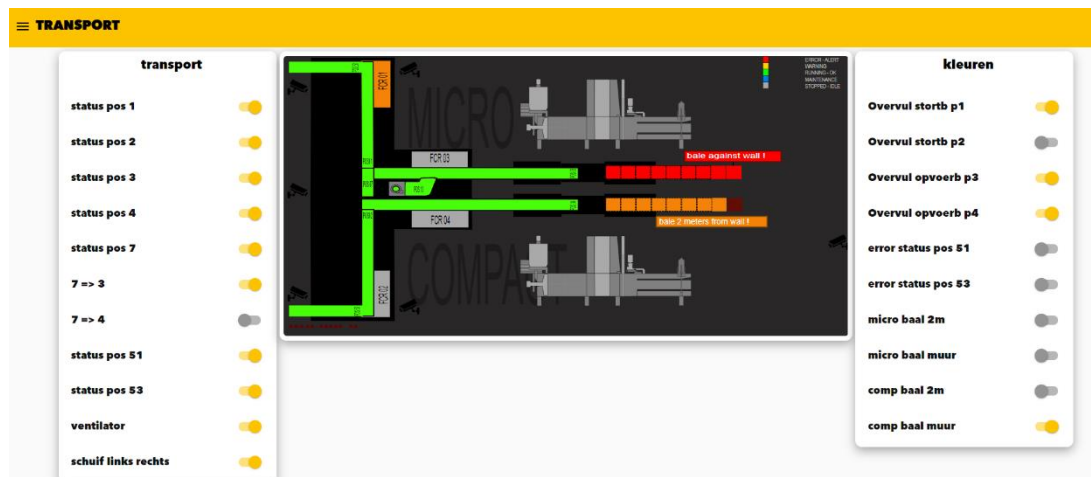
#### 5.1.2 Proces

Het project begon met Node-RED lessen van de oorspronkelijke maker van het scherm. Deze lessen werden bijgewoond door de IoT-Administrator, een andere stagiair en mezelf.

Nadat de lessen waren afgerond, zijn we begonnen met het verbeteren van het scherm. De van kleur veranderende tekening was gemaakt met Inkscape. De andere stagiair en ik zijn begonnen met het opruimen van de Inkscape-

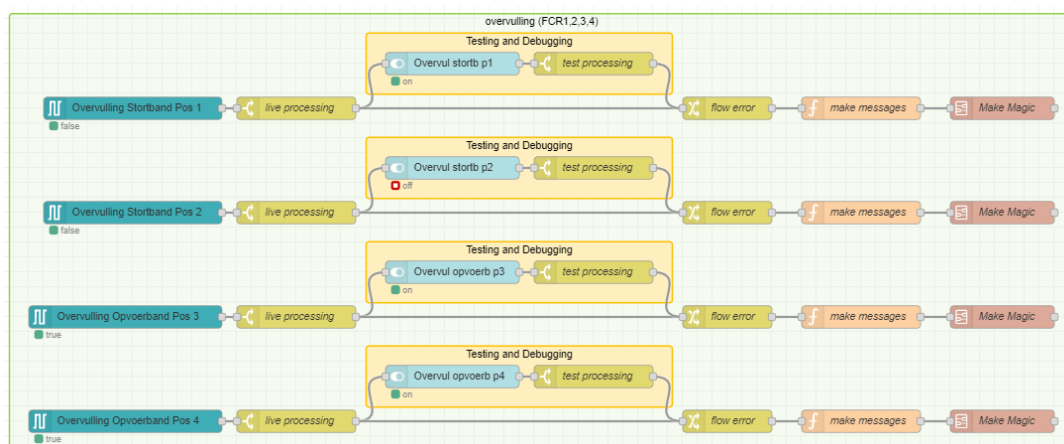
tekening. De oorspronkelijke maker van het scherm had al een basisconcept bedacht voor de nieuwe versie van Node-RED. Nadat de andere stagiair en ik de Node-RED-flow hadden aangevuld, heb ik het project zelfstandig voortgezet.

In de nieuwe versie heb ik ook een dashboard gecreëerd waarmee elk signaal gesimuleerd kan worden. Hierdoor kunnen we gemakkelijk aan anderen laten zien hoe de tekening reageert op bepaalde signalen. Dit dashboard is ook erg handig voor testdoeleinden. Bijvoorbeeld, als we willen dat iets groen wordt bij 'true' en rood bij 'false', en het dashboard toont iets anders, dan weten we dat er een programmeerfout is opgetreden.



Figuur 9 dashboard BOA

Ik heb gestreefd naar een gestandaardiseerde aanpak voor bijna alle signalen, zodat het toevoegen van een nieuw signaal zo eenvoudig mogelijk is. Men kan simpelweg een bestaand voorbeeld kopiëren en alleen de kleuren en CSS-ID's van de SVG aanpassen. Er waren echter enkele situaties waarin bepaalde signalen afhankelijk waren van meerdere factoren of waarin sommige signalen prioriteit hadden. In deze gevallen kon de gestandaardiseerde aanpak niet altijd worden toegepast. Ik heb alle uitzonderingen en bijzonderheden gedocumenteerd op een intern bedrijfs-Wikipedia-platform.



Figuur 10 deel van de Node-RED BOA

De signalen die we gebruiken, worden geleverd door een PLC. In de loop der jaren is er echter wat kennis verloren gegaan over de betekenis van bepaalde signalen en zijn er dubbele signalen ontstaan waarvan niemand wist welke nog relevant waren. Om dit probleem op te lossen, hadden we een testavond georganiseerd. Ik had in Node-RED enkele schermen gemaakt waarop alle signalen zichtbaar waren en van kleur veranderden afhankelijk van hun status. Hierdoor konden we sensoren testen en zien welk signaal erop reageerde. Nadat de juiste signalen in Node-RED werden gebruikt, heb ik een document opgesteld waarin de tekening en de kleurcodes worden uitgelegd aan de medewerkers in de fabriek. Het aangepaste scherm is vervolgens op de tv in de fabriek geplaatst.

### 5.1.3 Mijn bijdrage

- Het opschonen van een deel van de Inkscape-tekening en het aanpassen van de ID's van de SVG-bestanden (voor CSS).
- Het creëren van een dashboard waarmee alle signalen kunnen worden gesimuleerd.
- Het ontwikkelen van een speciaal testdashboard voor de PLC-signalen.
- Het ontwerpen en implementeren van een groot deel van de Node-RED-flow voor het scherm in de fabriek.
- Het documenteren van het gehele projectproces en de oplossingen die zijn toegepast.

### 5.1.4 Conclusie

Ik ben bijzonder dankbaar voor de Node-RED-lessen die we in het begin hebben gehad. Het heeft het eenvoudig gemaakt om PLC-signalen binnen te halen, en omdat je ook je eigen code kunt schrijven, zijn er eindeloze mogelijkheden. Ik heb de opgedane kennis zelfs toegepast in mijn eigen projecten thuis. Ik heb veel geleerd van dit project en ben verheugd dat de oplossing nu daadwerkelijk in gebruik is genomen in de fabriek.

## 5.2 Kafka Magic

### 5.2.1 Context

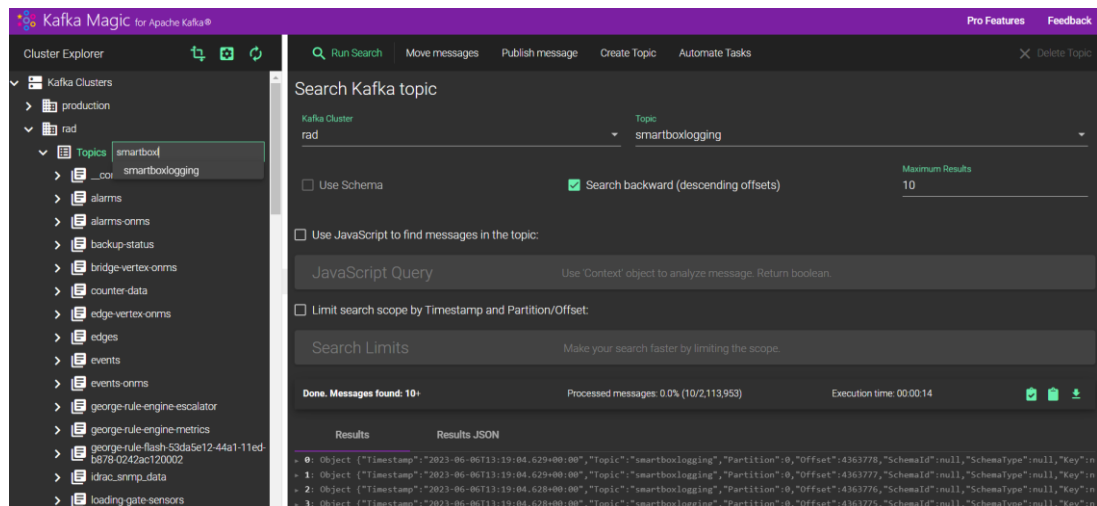
Het bedrijf streeft ernaar om verschillende processen te standaardiseren, waarvan één aspect het gebruik van Kafka omvat. Vanwege het brede gebruik ervan binnen het bedrijf, was er behoefte aan een applicatie om Kafka te beheren. Er was al een keuze gemaakt voor een specifieke applicatie, en mijn opdracht was om deze op te zetten.

### 5.2.2 Proces

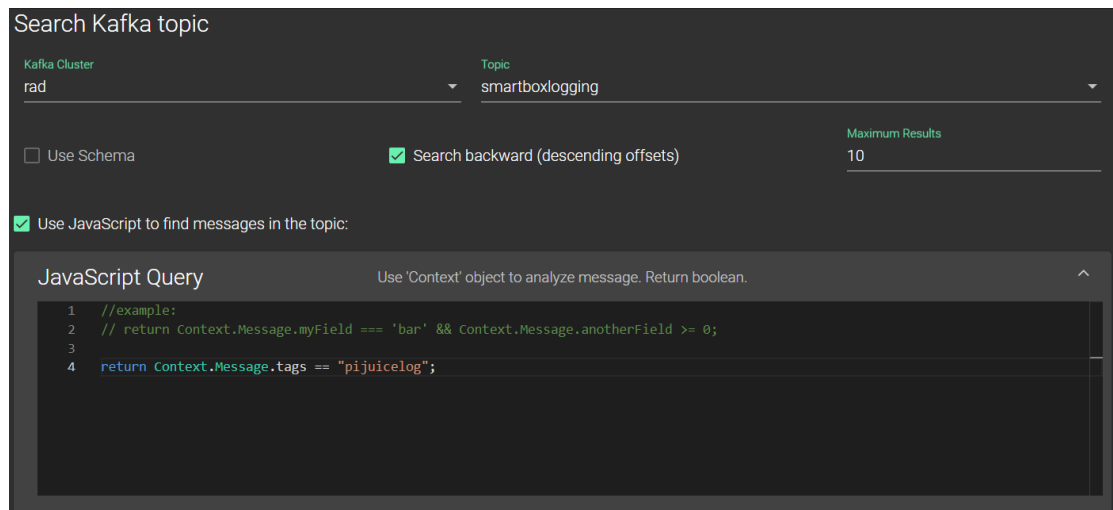
Het bedrijf had gekozen voor Kafka Magic. Ik begon met het installeren van deze applicatie op mijn laptop om te begrijpen hoe alles geconfigureerd moest worden. Het gebruik van de tool verliep vrij soepel. Echter, de documentatie voor Kafka Magic was niet erg uitgebreid, waardoor het lastig was om de juiste configuratiebestanden te maken zonder handmatige aanpassingen. Nadat ik de stappen had ontdekt om Kafka Magic te installeren en de juiste configuratiebestanden had gemaakt, heb ik een Ansible playbook gemaakt. Het bedrijf maakt voor elke installatie gebruik van Ansible playbooks, zodat de

installatie eenvoudig opnieuw kan worden uitgevoerd of op meerdere servers kan worden toegepast.

Daarna kwam een andere stagiair mij helpen. Zijn stage richtte zich voornamelijk op het opzetten van servers. Het Ansible playbook dat ik had geschreven, is dan nog aangepast en aangevuld. Bijvoorbeeld, van Kafka Magic is een service gemaakt, waardoor het eenvoudiger is om deze te starten, stoppen en herstarten. Het Ansible playbook werkte niet op een server omdat Kafka Magic alleen verzoeken van zichzelf accepteerde. Daarom heeft de andere stagiair een proxy gemaakt.



Figuur 11 Kafka Magic zoeken naar topic "smartboxlogging"



Figuur 12 Kafka Magic verfijn zoekopdracht met JavaScript

```

0:
  Timestamp: "2023-06-06T13:19:34.648+00:00"
  Topic: "smartboxlogging"
  Partition: 0
  Offset: 4363839
  SchemaId: null
  SchemaType: null
  Key: null
  Headers: null
  Message:
    @timestamp: "2023-06-06T13:19:34.648Z"
    @metadata: Object {"beat": "filebeat", "type": "_doc", "version": "8.7.0", "_id": "Km_ckIg8KMDHtoFY4Kn4"}
    event: Object {"timezone": "+02:00"}
    log: Object {"offset": 14200477, "file": {"path": "/tmp/log/pijuice.log"}}
    message: "2023-06-06 15:19:27 ; log_level: FINE ; charge_percentage: 99 ; charge_error: NO_ERROR ; firmware_version: 1.5 ; firmware_variant: 0 ; firmware_error: NO_ERROR ;"
    tags: Array[1] ["pijuicelog"]
    input: Object {"type": "filestream"}
    ecs: Object {"version": "8.0.0"}
    host: Object {"containerized": false, "name": "smartboxtest030004.rad.priv.vangenechten.com", "ip": ["10.1.60.229", "fe80::6ad4:bacf:ee16:68f"], "mac": [
    agent: Object {"version": "8.7.0", "ephemeral_id": "e02287ce-6ace-4405-8993-162c4142f8b8", "id": "201c5de9-e797-44e3-a80d-9cc57d7d94c3", "name": "smartb

```

Figuur 13 Kafka Magic voorbeeld van een bericht

### 5.2.3 Mijn bijdrage

Ik heb onderzocht hoe Kafka Magic correct geïnstalleerd moest worden, inclusief de juiste configuratie. Ik heb ook meegeholpen aan de Ansible playbook maar dit was vooral de andere stagiair.

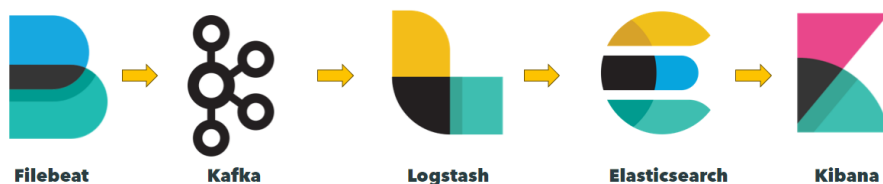
### 5.2.4 Conclusie

Dit project leek eerst niet heel uitdagend op mijn laptop, maar bij de installatie op een server kwamen we steeds weer problemen tegen. Het was geweldig om met een andere stagiair samen te werken die meer kennis had op dat gebied. We konden samen op zoek gaan naar oplossingen, en als ik bijvoorbeeld een mogelijke oplossing vond zoals de proxy, kon hij deze implementeren.

## 5.3 Smartbox monitoring

### 5.3.1 Context

Het bedrijf heeft ongeveer 170 smartboxen, verdeeld over 12 filialen. Deze smartboxen bevatten een Raspberry Pi en zijn geplaatst bij machines in de fabriek. Er was behoefte aan een monitoringssysteem voor deze boxen, zodat eventuele problemen sneller konden worden gedetecteerd. Het bedrijf wilde enkele logbestanden verzamelen in een Elasticsearch-database en deze visualiseren voor verduidelijking.



Figuur 14 stappen die een logfile doorloopt



### 5.3.2 Proces

Toen het project werd toegelicht, heb ik me eerst enkele dagen verdiept in de Elastic Stack. Ik heb de documentatie doorgenomen en enkele Udemy-cursussen gevolgd om een beter idee te krijgen van de mogelijkheden van de Elastic Stack. Vervolgens heb ik nogmaals gesproken met de IoT-beheerder om de details van zijn verwachtingen te bespreken.

#### 5.3.2.1 Filebeat

Ik stelde voor om Filebeat te installeren op elke smartbox. In Filebeat heb ik de logbestanden gedefinieerd en ingesteld dat deze via Kafka worden verzonden. Binnen het bedrijf is het gebruikelijk om alles via Kafka te versturen. Filebeat houdt bij welke logregels al zijn verzonden en stuurt alleen de nieuwe regels door.

Het zou veel werk zijn om handmatig in te loggen op elke smartbox, Filebeat te downloaden en in te stellen. Daarom heb ik Ansible geleerd, dat binnen het bedrijf al wordt gebruikt voor het opzetten van alle servers. Ik heb een Ansible playbook geschreven voor de installatie en configuratie van Filebeat. Met één commando kan het Ansible playbook worden uitgevoerd op alle smartboxen.

Tijdens de vergadering met de IoT-Administrator werd duidelijk dat het handig zou zijn om te kunnen filteren op bijvoorbeeld filiaal of smartbox-naam, zodat er achteraf gezocht kon worden naar bijvoorbeeld smartboxen uit Turnhout. Deze informatie stond niet in de logbestanden, dus moest ik een manier vinden om dit toe te voegen. Uiteindelijk heb ik ervoor gezorgd dat deze informatie werd toegevoegd op basis van het MAC-adres van de smartbox waar de logbestanden vandaan komen. In Filebeat heb ik ingesteld dat bij elke logregel die wordt doorgestuurd, ook het MAC-adres wordt meegezonden. Er was al een Excel-bestand waarin het MAC-adres, filiaal en smartbox-naam van elke smartbox stonden vermeld. Hiervoor heb ik een Python-script geschreven dat JSON-bestanden genereerde die konden worden gebruikt in Logstash.

#### 5.3.2.2 Logstash

Logstash is het onderdeel van de Elastic Stack waarmee je berichten kunt verwerken en opschonen voordat je ze in de Elasticsearch-database plaatst. De input was Kafka, de output was Elasticsearch en daartussen waren veel filters toegepast.

In Logstash werden de JSON-bestanden gebruikt die ik had gemaakt. In deze bestanden werd gezocht naar het MAC-adres. De bijbehorende filiaalnaam, filiaalnummer en smartbox-naam werden vervolgens aan het bericht toegevoegd.

Doordat het MAC-adres werd meegestuurd in Filebeat, werden ook veel onnodige gegevens meegezonden. Deze werden in Logstash verwijderd. In Filebeat had ik ook een tag toegevoegd voor elk logbestand. Hierdoor kon ik de berichten onderscheiden en verschillende filters toepassen, afhankelijk van het logbestand waar ze vandaan kwamen.

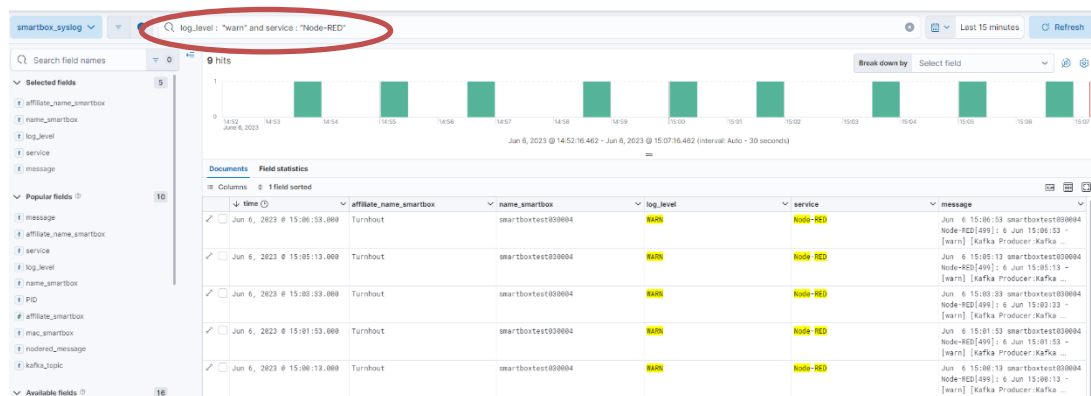
Voor elk type logbestand had ik aparte grok-patronen gemaakt. Deze patronen haalden belangrijke informatie uit de logregels, zoals een timestamp, logniveau of batterijpercentage. Door deze informatie apart te extraheren, kan er

gemakkelijk worden gezocht naar logregels met bijvoorbeeld logniveau WARN of ERROR. Elk logbestand had ook een andere notatie voor de datum. In de Logstash-filters werd dit omgezet naar een formaat dat Elasticsearch herkent als tijdstempel.

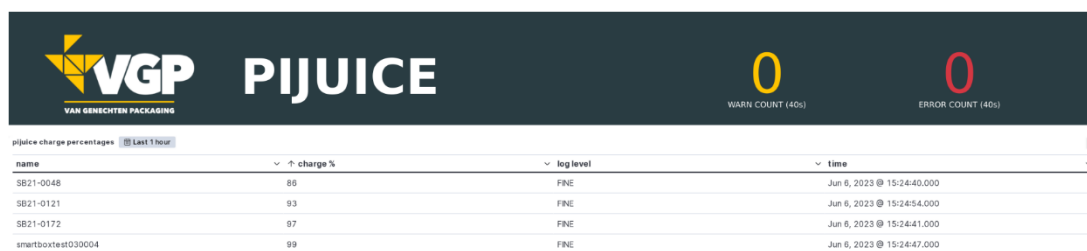
### 5.3.2.3 Elasticsearch en Kibana

Na Logstash worden de berichten opgeslagen in de Elasticsearch-database. Kibana is de visualisatietool van de Elastic Stack. In Kibana kunnen ook instellingen worden gemaakt voor Elasticsearch, zoals de indexen waarin de berichten worden opgeslagen. Hiervoor heb ik de mapping gedefinieerd. De mapping geeft aan welk gegevenstype elk veld van een bericht heeft. Elasticsearch doet dit automatisch, maar niet altijd correct.

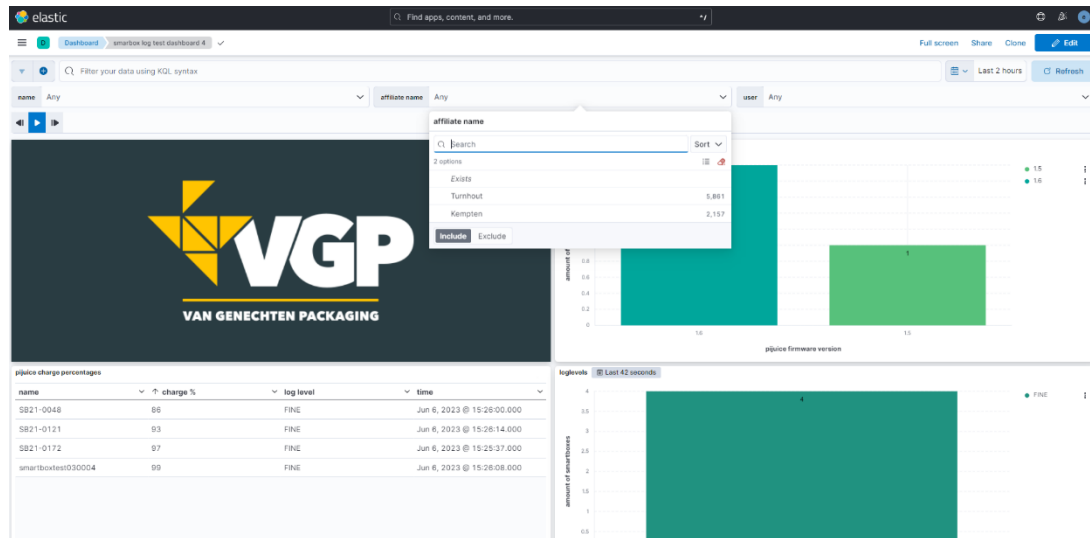
In Kibana kunnen bijvoorbeeld eenvoudig berichten worden gezocht van een specifieke smartbox met logniveau WARN, dankzij de filtering die in Logstash is toegepast. Daarnaast heb ik ook een dashboard gemaakt. In dit dashboard staan grafieken die laten zien hoe het gaat met de smartboxen. Als de grafieken aangeven dat er een probleem is, kan er gemakkelijk worden achterhaald bij welke smartbox dit probleem hoort. Er zijn ook enkele canvassen. Een canvas in Kibana is een soort dashboard dat periodiek wordt vernieuwd. Zo heb ik bijvoorbeeld een canvas gemaakt waarop direct te zien is of er smartboxen zijn zonder stroom.



Figuur 16 filtering van data op loglevel en service in Kibana discover



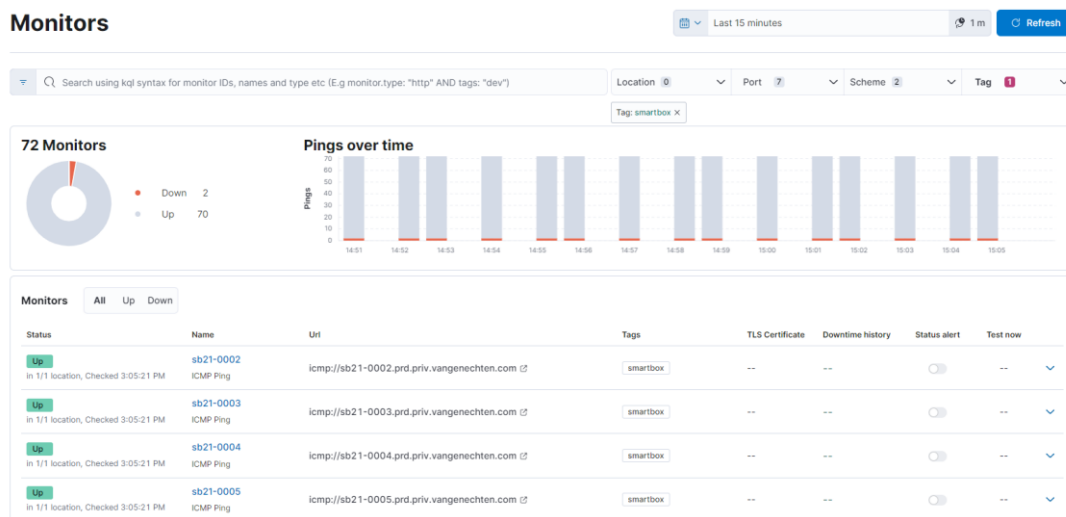
Figuur 15 canvas in Kibana dat duidelijk weergeeft of er smartboxen zonder stroom zijn gevallen



Figuur 17 dashboard in Kibana

### 5.3.2.4 Heartbeat

In een vergadering werd aangegeven dat er behoefte was aan een manier om te controleren of de smartboxen nog bereikbaar waren en of bepaalde services nog actief waren. Daarom stelde ik Heartbeat voor, en dat idee werd positief ontvangen. Met Heartbeat kunnen controles worden gedefinieerd met ICMP, TCP of HTTP. De Excel die ik eerder gebruikte om JSON-bestanden voor Logstash te genereren, heb ik ook weer gebruikt voor Heartbeat. Met een ander Python-script genereer ik een YAML-bestand dat Heartbeat gebruikt om alle smartboxen te controleren. In Kibana is het dan duidelijk zichtbaar of er boxen niet bereikbaar zijn.



Figuur 18 Heartbeat uptime in Kibana

Uiteindelijk heb ik alles ontwikkeld en is het gereed voor productie. Aan het einde van mijn stage waren de productieservers echter nog niet klaar, waardoor het nog niet in gebruik kon worden genomen.

### 5.3.3 Mijn bijdrage

Ik heb hier zelfstandig aan heb gewerkt. Ik kon altijd vragen stellen aan mijn stagebegeleiders wanneer ik vastliep.

### 5.3.4 Conclusie

Ik vond dit een zeer interessant project. Ik had nog geen ervaring met de Elastic Stack en het was een uitdaging om dit te ontdekken. Naarmate ik meer begrip kreeg van een onderdeel van de stack, verliep het begrijpen van de andere onderdelen soepeler omdat de documentatie van alle componenten op een vergelijkbare manier was opgesteld. Hierdoor kon ik steeds meer informatie opzoeken in de documentatie in plaats van afhankelijk te zijn van video's of forums.

## 5.4 Stoomketels

### 5.4.1 Context

In de fabriek zijn er meerdere stoomketels die bij een eventuele explosie dodelijk kunnen zijn voor mensen in de nabije omgeving. Daarom wilden ze de stoomketels kunnen monitoren. Buiten de ruimte waar de stoomketels zich bevinden, wilden ze een grote tv plaatsen. Op deze tv moest een tekening komen waarbij verschillende delen van kleur veranderen afhankelijk van de status van de stoomketels.

### 5.4.2 Proces

Voor dit project kregen we een groot aantal signalen, waaronder digitale, analoge en rs485-signalen, evenals enkele digitale signalen waarvan de pulsen geteld moesten worden. Het bedrijf wilde werken met Adam-modules van Advantech. Er waren verschillende modules aangeschaft die ik één voor één heb getest met behulp van de IoT-Administrator. We hebben onderzocht hoe ze werken en hoe we de signalen kunnen verkrijgen in Node-RED, de tool die het scherm voor de tv creëert.

Iemand uit de fabriek had al een handgetekende schets gemaakt van de stoomketels. Ik heb deze schets vervolgens nagemaakt in Inkscape.

Nadat ik wist hoe alle modules werkten en een tekening had gemaakt, hebben we een vergadering gepland met de mensen uit de fabriek. Hierin kon ik al mijn vragen over de signalen stellen. Zij hebben vervolgens een Excel-bestand aangevuld met alle signalen en hun betekenissen, bijvoorbeeld 'true' voor alarm, 'false' voor oké, of 'true' voor aan, 'false' voor uit, ...

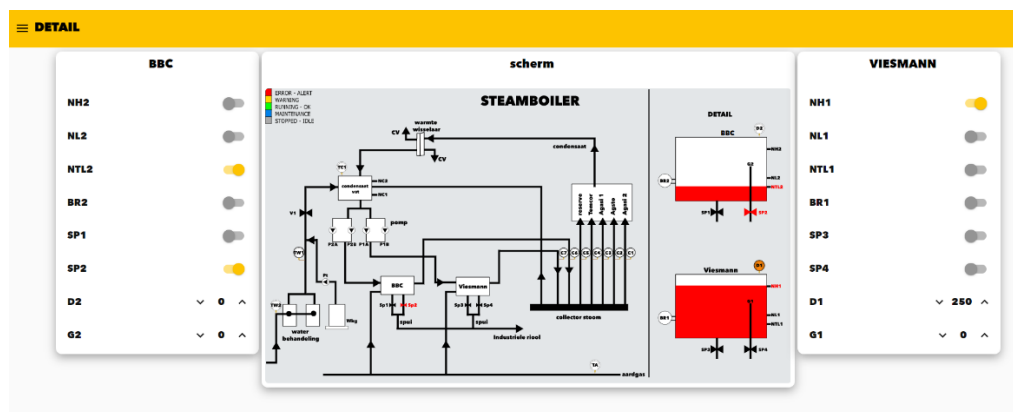
Ik heb verschillende voorstellen gedaan met combinaties van Adam-modules om de signalen uit te lezen. Uiteindelijk hebben we gekozen voor drie modules waarop al Node-RED draaide. Op twee van deze modules werden alle signalen gelezen en naar Kafka gestuurd. Op de derde module werden ook de signalen van de andere modules via Kafka ontvangen en in één groot bericht samengevoegd. Op die manier hadden we in Kafka één bericht dat alle signalen

bevatte, zodat er achteraf niet geweten hoeft te zijn welk signaal op welke module zat.

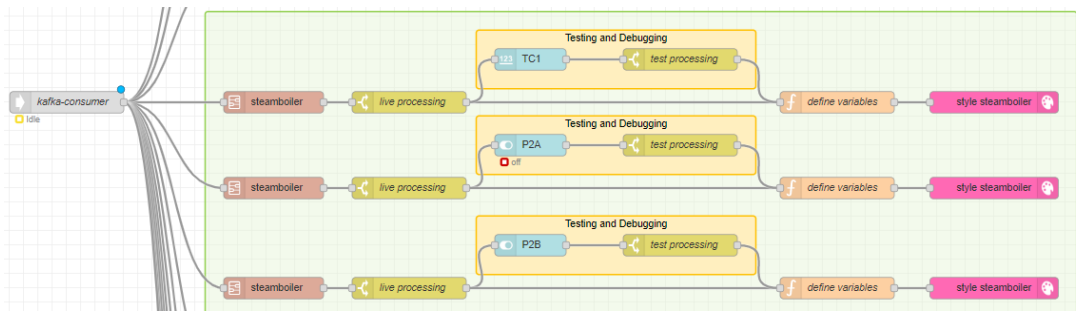


Figuur 19 box met de Adam modules die alle signalen binnen neemt

Om de tekening van kleur te laten veranderen, had ik een andere Node-RED-flow gemaakt op een smartbox. Deze flow abonneert op het grote bericht met alle signalen. Ik heb een gestandaardiseerde aanpak ontwikkeld om alle signalen te gebruiken. Als er later een nieuw signaal wordt toegevoegd, kan een bestaand signaal worden gekopieerd en hoeven slechts enkele elementen te worden aangepast, zoals de signaalnaam, de CSS-ID's van de elementen die van kleur moeten veranderen en de kleuren zelf. Op deze manier is het project gemakkelijk te onderhouden.



Figuur 20 dashboard van de stoomketels



Figuur 21 deel van de Node-Red van de stoomketels

### 5.4.3 Mijn bijdrage

- De nieuwe Adam-modules getest en geëvalueerd.
- Een ontwerp gemaakt voor een box om alle signalen te verzamelen.
- De bedrading van de box verzorgd.
- Node-RED geïmplementeerd op de Adam-modules.
- Node-RED geconfigureerd op de smartbox.
- Het project gedocumenteerd.

### 5.4.4 Conclusie

De behuizing is volledig bedraad en de Node-RED-implementatie is grotendeels voltooid. Mijn stage was echter afgerond voordat de box kon worden opgehangen en de signalen konden worden aangesloten. Ik ben zeer tevreden met dit project. Ik heb het gevoel dat ik veel invloed had op hoe het werd uitgevoerd en dat er werd geluisterd naar mijn suggesties.

## **6 CONCLUSIE**

Deze stageperiode was een waardevolle en verrijkende ervaring, waarin ik de gelegenheid kreeg om me onder te dompelen in verschillende nieuwe technologieën. De begeleiding die ik gedurende mijn stage heb ontvangen, is zeer waardevol geweest bij het ontwikkelen van mijn vaardigheden en het verkennen van de mogelijkheden van deze technologieën.

Tijdens de stage heb ik de kans gehad om hands-on ervaring op te doen met het implementeren en toepassen van deze technologieën in verschillende contexten. De mogelijkheid om te werken aan diverse projecten heeft mijn leerproces verrijkt en mij in staat gesteld om mijn theoretische kennis om te zetten in praktische vaardigheden.

In conclusie kan ik stellen dat deze stage-ervaring een belangrijke rol heeft gespeeld in mijn professionele ontwikkeling. Dankzij de ondersteuning en begeleiding van mijn mentors heb ik niet alleen nieuwe technologieën leren kennen, maar ook geleerd hoe ik ze effectief kan toepassen bij echte scenario's. Deze stage heeft mijn vertrouwen vergroot en me voorbereid op toekomstige uitdagingen in het werkveld. Ik ben dankbaar voor de kans die ik heb gekregen en ik kijk enthousiast uit naar de verdere toepassing en ontwikkeling van mijn nieuwe vaardigheden.

## **7 DANKWOORD**

Graag wil ik van deze gelegenheid gebruik maken om mijn oprechte dank uit te spreken aan Van Genechten Packaging / Imas N.V. voor de mooie stagekansen die mij is geboden. Mijn waardering gaat uit naar Luc Pluys, Brent Vliegen, Bryan Antonis, Wout Bertels en George Proost voor hun begeleiding tijdens mijn stageperiode. Hun expertise en ondersteuning hebben sterk bijgedragen aan mijn professionele ontwikkeling.

Daarnaast wil ik ook nog Paul Boonen, mijn begeleider van Thomas More, bedanken.

Ik ben ook dankbaar voor het hele team van Van Genechten Packaging / Imas N.V. voor hun warme ontvangst en samenwerking gedurende mijn stage. Jullie professionele omgeving en collegialiteit hebben mijn leerervaring nog waardevoller gemaakt.



## 8 BRONNENLIJST

*ansible.* (sd). Opgeroepen op 4 juni 2023, van ansible:  
<https://www.ansible.com/>

*ChatGPT.* (sd). Opgeroepen op 4 juni 2023, van <https://chat.openai.com/>

*Elastic NV.* (sd). Opgeroepen op 6 juni 2023, van Wikipedia:  
[https://en.wikipedia.org/wiki/Elastic\\_NV](https://en.wikipedia.org/wiki/Elastic_NV)

*Elastic Stack.* (sd). Opgeroepen op 4 juni 2023, van elastic.co: [elastic.co/elastic-stack/](https://elastic.co/elastic-stack/)

*File:Ansible logo.svg.* (sd). Opgeroepen op 6 juni 2023, van wikimedia commons: [https://commons.wikimedia.org/wiki/File:Ansible\\_logo.svg](https://commons.wikimedia.org/wiki/File:Ansible_logo.svg)

*File:Apache kafka wordtype.svg.* (sd). Opgeroepen op 6 juni 2023, van wikimedia commons:  
[https://commons.wikimedia.org/wiki/File:Apache\\_kafka\\_wordtype.svg](https://commons.wikimedia.org/wiki/File:Apache_kafka_wordtype.svg)

*File:Inkscape Logo.svg.* (sd). Opgeroepen op 6 juni 2023, van Wikimedia commons: [https://commons.wikimedia.org/wiki/File:Inkscape\\_Logo.svg](https://commons.wikimedia.org/wiki/File:Inkscape_Logo.svg)

*Inkscape.* (sd). Opgeroepen op 4 juni 2023, van [inkscape.org](https://inkscape.org/):  
<https://inkscape.org/>

*Kafka.* (sd). Opgeroepen op 4 juni 2023, van <https://kafka.apache.org/>

*Node-RED.* (sd). Opgeroepen op 4 juni 2023, van [nodered.org](https://nodered.org/):  
<https://nodered.org/>

*Resources.* (sd). Opgeroepen op 6 juni 2023, van [nodered.org](https://nodered.org/about/resources/):  
<https://nodered.org/about/resources/>

*Van Genechten Packaging.* (sd). Opgeroepen op 14 mei 2023, van <https://vangenechten.com>