Pipeline power tool

Software Requirements Specification

Versie 1.0

Inhoudsopgave

[1 Inleiding 3](#_Toc143114443)

[1.1 Doel 3](#_Toc143114444)

[1.2 Weergaveregels 3](#_Toc143114445)

[1.3 Doelgroep 3](#_Toc143114446)

[1.4 Product Scope 3](#_Toc143114447)

[1.5 Referenties 3](#_Toc143114448)

[2 Algemene Beschrijving 4](#_Toc143114449)

[2.1 Productperspectief 4](#_Toc143114450)

[2.2 Productfunctionaliteiten 4](#_Toc143114451)

[2.3 Gebruikersklassen and Karakteristieken 5](#_Toc143114452)

[2.4 Operationele Omgeving 5](#_Toc143114453)

[2.5 Design- en Implementatiebeperkingen 5](#_Toc143114454)

[2.6 Gebruikersdocumentatie 5](#_Toc143114455)

[2.7 Aannames en Afhankelijkheden 6](#_Toc143114456)

[3 Externe Interface-eisen 7](#_Toc143114457)

[3.1 Gebruikersinterface 7](#_Toc143114458)

[3.2 Hardware Interface 8](#_Toc143114459)

[3.3 Software Interface 9](#_Toc143114460)

[3.4 Communicatie Interface 11](#_Toc143114461)

[4 Systeemfunctionaliteiten 12](#_Toc143114462)

[4.1 Functionele Eisen 16](#_Toc143114463)

[4.1.1 Pipeline Power Meter 16](#_Toc143114464)

[4.1.2 Pipeline Plugin 16](#_Toc143114465)

[4.1.3 Pipeline Power API 16](#_Toc143114466)

[4.1.4 Pipeline Carbon Calculator 16](#_Toc143114467)

[Bijlage 1 17](#_Toc143114468)

# Inleiding

## Doel

Het doel van dit document is om het volledige systeem van de open-source Pipeline Power Tool te beschrijven, waaronder het doel van de applicatie, de functionaliteiten die de tool biedt en hoe deze technisch geïmplementeerd en gebruikt kan worden door stakeholders.

## Weergaveregels

Het document is gebaseerd op de 830-1998 – IEEE Recommended Practice for Software Requirements Specifications standaard van het Institute of Electrical and Electronical Engineers (IEEE) en het template van Karl E. Wiegers.

## Doelgroep

Dit document is bedoeld voor de stakeholders gemarkeerd als high interest in de bijlage STAKEHOLDERANALYSE.

**Devops Software Engineers**: Dit zijn directe gebruikers van de Pipeline Power Tool, omdat zij deze implementeren in hun pipelines and dashboards. Daarbij hebben ze ook de mogelijkheid om mee te werken aan de open-source tool.

**Project Managers**: Projectmanagers kunnen analyses maken naar aanleiding van gemaakte dashboards boven op de API en deze doorgeven aan hoger management en Finance departments.

**QA Engineers**: QA Engineers kunnen de tool globaal implementeren in eigen QA-specifieke pipelines en programma’s.

## Product Scope

De Pipeline Power Tool is een multi-component software tool bestaande uit een pipeline add-on, database en REST API. DevOps Software Engineers kunnen de tool gebruiken om het energieverbruik en CPU-gebruik te meten en monitoren van pipeline runs. Vervolgens kunnen hiermee beslissingen gemaakt worden om het energieverbruik en CPU-gebruik te optimaliseren ten goede van Green IT principes.

## Referenties

Afstudeerscriptie: als externe bijlage

* Pipeline Power Tool: <https://github.com/pipelinepowertool>
* IEEE Template: <https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc>

# Algemene Beschrijving

## Productperspectief

Pipelines zijn soms “over-engineered” en worden te pas en te onpas afgetrapt. In dat laatste geval kan de verbruikte energie als verloren beschouwd worden. Daarbij kan het voorkomen dat pipeline hosts een buitensporig krachtige CPU bevatten die maar voor een klein deel benut wordt.

Deze tool is ontwikkeld om Green IT en Green DevOps te stimuleren, door developers, productmanagers en andere directe stakeholders inzicht te geven in het energieverbruik en CPU-gebruik van de pipelines, waarmee zij acties kunnen ondernemen.

## Productfunctionaliteiten

**Pipeline Power Meter**

Bereken CPU-gebruik

Bereken CPU-energieverbruik

Bereken CPU PID CPU-gebruik

Bereken CPU PID energieverbruik

Opslaan in CSV

**Pipeline Plugins**

Starten van Power Meter

JSON opslaan als pipeline artifact

Aggregeren van metingen uit CSV

Versturen van data naar database

**Pipeline Power API**

Ophalen van data uit database

Verwerken van data

Afhandelen van verzoeken vanuit de Pipeline Carbon Calculator

**Pipeline Carbon Calculator**

Invullen van pipeline gegevens op website

Weergeven van verbruiksdata gekoppeld aan voorbeelden voor perspectief

**Dashboard**

Weergeven van verbruiksdata, afgesteld op de individuele business requirements

## Gebruikersklassen and Karakteristieken

**DevOps Software Engineers**

Implementeren van Pipeline Power Tool in de pipelines

Meewerken aan het open-source project

Maken van dashboards

Analyseren van energieverbruik

**Project Managers**

Maken van dashboards

Analyseren van energieverbruik

## Operationele Omgeving

Self-hosted dockerized pipeline Linux agents in versie 1.0. Later kan dit uitgebreid worden naar Cloud-hosted dockerized Linux pipelines zodra meer bekend is over het energieverbruik van cloud-machines.

## Design- en Implementatiebeperkingen

* De eerste versie van de Pipeline Power Tool maakt gebruik van Intel RAPL, waardoor de tool alleen op self-hosted Linux agents met een Intel CPU werkt.
* De tool is vendor-independent, maar voor het gemak van de developers zal de eerste versie van de Power Pipeline Tool een Jenkins plugin bevatten.
* Alle tools zijn self-hosted. Pipeline Power Tool biedt geen hosting aan, met uitzondering van de C executable, pipeline plugin en de broncode.

## Gebruikersdocumentatie

* Externe bijlage Scriptie: Onderzoek en verantwoording van de Tool.
* Bijlage 1: API-documentatie voor code-generation.
* GitHub: https://github.com/pipelinepowertool.
* ELK Documentatie: https://www.elastic.co/guide/index.html

## Aannames en Afhankelijkheden

* De Pipeline Power Meter wordt gehost in een AWS S3 bucket. Als deze bucket offline is zal de tool niet gebruikt kunnen worden.
* De Pipeline Power Meter heeft toegang tot de RAPL-data, waar root access voor nodig is. Dit kan verkregen worden door -r--r--r-- toestemming te geven op het RAPL energy\_uj bestand.
* De database en het dashboard maken gebruik van de ELK stack (ElasticSearch, Logstash & Kibana) of MongoDB, door het gebruik van hoge ongestructureerde data volumes die niet geüpdatet worden en geaggregeerde/gefilterde queries. De tool kan probleemloos geïntegreerd worden met bestaande ELK-stack implementaties.
* De Pipeline Power API en de pipeline agent hebben toegang nodig tot de ElasticSearch database.
* Voor een correcte afspiegeling van de werkelijkheid is het nodig dat de agents in Docker containers of hun eigen virtuele omgeving draaien. Bij een server per agent opzet zullen er veel processen meegenomen worden die niet direct verbonden zijn met de taken van de pipeline.
* De Pipeline Power Meter is een C executable. De pipeline agent moet dit ondersteunen.
* De programmeertaal van de REST API en pipeline plugin is Java 17.
* De Calculator is geschreven in VueJS.

# Externe Interface-eisen

## Gebruikersinterface

De user interface van het Kibana dashboard is dynamisch en zal veranderen per bedrijf per use-case. De Pipeline Carbon Calculator, te zien in Figuur 1, is echter een website die gelijk is voor alle gebruikers.

A screenshot of a calculator

Description automatically generated

Figuur 1: Ontwerp Pipeline Carbon Calculator

* **Dropdown CI/CD tool**: Zoals vermeld zal de eerste versie van de tool enkel ondersteuning hebben voor Jenkins Pipelines. Dit kan later echter uitgebreid worden naar onder andere Azure DevOps. Bij het klikken op een CI/CD tool zullen de input field aangepast worden naar de syntax van de desbetreffende tool.
* **Job Input Field**: Naam van de pipeline (niet verplicht).
* **Branch Input Field**: Naam van de branch (niet verplicht) en alleen nodig als de pipeline gekoppeld is aan een GIT branch.
* **Build Input Field**: Nummer van de pipeline build (niet verplicht)
* **URL Input Field**: De volledige URL van de pipeline kan hier ingevuld worden, waarna de Job, Branch, en Build Input Fields automatisch gevuld worden met de juiste waardes. Dit werkt ook andersom (niet verplicht).
* **Calculate Button**: Stuurt de ingevulde waardes van het input form naar de API.
* **n Watts**: Aantal watts verbruikt in de geselecteerde pipeline(s).
* **n kWh**: Aantal kWh verbruikt in de geselecteerde pipeline(s).
* **n pipeline runs**: Het aantal pipeline runs van de geselecteerde pipeline(s).
* **n pipeline time (hr:m)**: De totale duur van de geselecteerde pipeline(s).
* **n g CO2 produced**: Het aantal gram of kg van CO2 uitstoot.
* **Equivalent**: Een dynamische vergelijking van CO2 verbruik om perspectief te geven aan de gebruiker.

## Hardware Interface

Figuur 2: Pipeline Agent Host Interface

De minimale hardware eisen van de tool is een Intel CPU van versie Sandy Bridge of hoger in verband met het gebruik van de Intel RAPL interface. Daarbij zal het mogelijk zijn om vanuit een controller de tool te laten draaien op de “execution environments” van de agents (Figuur 2), zodat alleen het energieverbruik van een specifieke pipeline run wordt gemeten.

Verder moet de hardware in staat zijn om een REST API te hosten, en heeft dus minimaal 1GB RAM nodig.

De hardware eisen van de ELK stack zijn te lezen op de officiële website en kunnen veranderen per gebruiksdoeleinde.

## Software Interface

Figuur 3: Software Componenten

In Figuur 3 zijn alle softwarecomponenten van de Pipeline Power Tool afgebeeld.   
De groen gemarkeerde componenten komen met de tool en kunnen geïntegreerd worden in de, eventueel bestaande, commerciële componenten die met rood zijn gemarkeerd. Ieder component dient wel op een eigen server gehost te worden, met uitzondering van de Pipeline Power Reader en de Jenkins Plugin (in versie 1.0), die via een openbare S3 bucket te verkrijgen zijn. De Power Reader wordt via een standaard **curl** commando opgehaald in de Pipeline Plugin. In de Plugin worden de verbruiksmetingen intern opgeslagen in een CSV. Versie 1.0 komt ook met een ElasticSearch integratie, maar wordt later uitgebreid met een MongoDB integratie.

De CSV bestaat uit records met de headers te zien in Tabel 1:

|  |  |
| --- | --- |
| Header | Beschrijving |
| pid | Het PID van een process dat draait op een agent. |
| cpu\_util | Het CPU gebruik (percentage) |
| cpu\_power | Het CPU verbruik (watts) |

Tabel 1: Metingen CSV Headers

Deze records worden per seconde aangevuld door de Pipeline Power Meter.

Om de records vervolgens naar de ElasticSearch database te sturen wordt gebruik gemaakt van een generieke Pipeline Plugin Library, zodat deze ook op een later moment voor andere CI/CD DevOps tools gebruikt kan worden. Deze Plugin Library aggregeert de verzamelde data en stuurt deze op naar de ElasticSearch cluster. Hierbij dient de mapping van Tabel 2 aangehouden te worden.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Beschrijving | Voorbeeld waarde | Doel | Type |
| cpu\_util\_pipeline | Het gemiddelde CPU-gebruik van alle pipeline specifieke taken. | 0.13 | Analyse van het CPU-verbruik van een pipeline agent. | float |
| cpu\_power\_pipeline | De verbruikte energie van alle pipeline specifieke taken. | 1 | Analyse van het energieverbruik van een pipeline agent. | float |
| metadata.timestamp\_start | Een timestamp van de start van de pipeline in milliseconden sinds de epoch | 1688209281 | Berekeningen, waaronder kWh, en analyse | date |
| metadata.timestamp\_end | Een timestamp van het einde van de pipeline in milliseconden sinds de epoch | 1688211231 | Berekeningen, waaronder kWh, en analyse | date |
| metadata.tool | De gebruikte CICD-tool | jenkins | Analyse via filteren op CI/CD-tooling. | text |
| metadata.jenkins.job | De naam van de Jenkins pipeline | pipeline1 | Filteren en aggregeren van verbruiksdata op job niveau. | text |
| metadata.jenkins.branch | De naam van de Jenkins pipeline branch (mits deze vanuit GIT komt) | master | Filteren en aggregeren van verbruiksdata op job + branch niveau. | text |
| metadata.jenkins.build | Het nummer van de Jenkins pipeline branch build | 1 | Filteren en aggregeren van verbruiksdata op job + branch + build niveau. | integer |

Tabel 2: CSV - Database Mapping

De developer krijgt in de pipeline niet de CSV te zien. Na de pipeline run wordt er een JSON gearchiveerd met de data die verzonden is naar de ElasticSearch cluster.

De Pipeline Power API is een Dockerized RESTful API die, vooralsnog, enkel tussen de Calculator en de ElasticSearch database staat en verantwoordelijk is voor het verwerken van HTTP-verzoeken en data. Om connectie te maken met de eigen ElasticSearch database zullen de database-credentials als environment variables meegegeven moeten worden via het meegeleverde docker-compose bestand.

De endpoints die worden ondersteund staan in Tabel 3 en specificatie van de gehele API is te vinden in Bijlage 1.

|  |  |  |  |
| --- | --- | --- | --- |
| Sinds Versie | HTTP Methode | Path | Beschrijving |
| 1.0 | GET | calculator/jenkins | Verbruiksdata van alle pipelines |
| 1.0 | GET | calculator/jenkins&job={job | Verbruiksdata van een specifieke pipeline |
| 1.0 | GET | calculator/jenkins?job={job}&branch=${branch} | Verbruiksdata van een specifieke branch in een specifieke pipeline |
| 1.0 | GET | calculator/jenkins?job={job}&build=${build} | Verbruiksdata van een specifieke build in een specifieke pipeline |
| 1.0 | GET | calculator/jenkins?job={job}&branch=${branch}&build=${build} | Verbruiksdata van een specifieke build in een specifieke branch in een specifieke pipeline |

Tabel 3: REST API Endpoints

## Communicatie Interface

Alle communicatie tussen componenten gaat over het HTTP-protocol.

# Systeemfunctionaliteiten

In dit hoofdstuk worden eerst de user stories beschreven die nodig zijn voor het MVP (SRS-versie 1.0). Vervolgens worden de functionele eisen per component met groene markering in Figuur 3 vastgesteld.

|  |  |
| --- | --- |
| User Story #1 | |
| Beschrijving:  Als gebruiker wil ik het CO2 verbruik van mijn pipelines inzien in een JSON, zodat ik mijn verbruik kan bekijken. | |
| Scenario: | Verbruik inzien. |
| Given | De gebruiker navigeert naar de pipeline. |
| When | De gebruiker configureert de meettool. |
| And | Start de pipeline. |
| Then | Wordt een JSON gegenereerd als pipeline artifact. |
| And | Ziet de gebruiker het verbruik. |

|  |  |
| --- | --- |
| User Story #2 | |
| Beschrijving:  Als gebruiker wil ik het CO2 verbruik van mijn pipelines inzien in een standaard overzicht met CO2 equivalenties, zodat ik mijn verbruik kan vergelijken en relativeren. | |
| Scenario: | Verbruik relativeren |
| Given | De gebruiker navigeert naar de Pipeline Carbon Calculator. |
| When | De gebruiker selecteert een CI/CD tool. |
| And | Vult geen of meerdere pipeline gerelateerde velden in. |
| And | Klikt op de Calculate knop. |
| Then | Stuurt de website een verzoek naar de Pipeline Power API. |
| And | Verkrijgt de website de benodigde data. |
| And | Ziet de gebruiker het verbruik en de CO2 equivalenties. |

|  |  |
| --- | --- |
| User Story #3 | |
| Beschrijving:  Als gebruiker wil ik het CO2 verbruik van mijn pipelines inzien in een eigen dashboard, zodat ik mijn verbruik kan analyseren. | |
| Scenario: | Verbruik analyseren |
| Given | De gebruiker navigeert naar de eigen analyse tool (Kibana, PowerBI, Grafana) |
| When | De gebruiker selecteert de ElasticSearch database als datasource. |
| And | Vult de connection string in. |
| And | Klikt op connect. |
| Then | Krijgt de gebruiker toegang tot de data. |
| And | Kan de gebruiker eigen dashboards maken voor analyse. |

|  |  |
| --- | --- |
| User Story #4 | |
| Beschrijving:  Als softwareontwikkelaar wil ik de tool eenvoudig aan een Jenkins pipeline koppelen. | |
| Scenario: | Koppelen aan Jenkins pipeline |
| Given | De developer maakt of bewerkt een pipeline. |
| When | De developer kiest een plugin. |
| And | Selecteert Pipeline Power Tool. |
| And | Koppelt de Tool aan de ElasticSearch omgeving en ingest pipeline. |
| And | Start de pipeline. |
| Then | Begint de Pipeline Power Meter direct met meten. |
| And | Stuurt de metingen uiteindelijk naar de ElasticSearch omgeving. |

|  |  |
| --- | --- |
| User Story #5 | |
| Beschrijving:  Als softwareontwikkelaar wil ik de tool eenvoudig aan een CI/CD tool koppelen. | |
| Scenario: | Koppelen aan andere pipeline (TBD). |
| Given | De developer bouwt een plugin. |
| When | De developer integreert de Pipeline Plugin Library in het plugin project. |
| Then | Kan de developer bestaande code en logica benutten zonder eerst de tool te analyseren. |

|  |  |
| --- | --- |
| User Story #6 | |
| Beschrijving:  Als gebruiker wil ik het CO2 verbruik van mijn pipelines, gekoppeld aan de koolstofintensiteit van het land waarin de agent gehost is, inzien in het standaard overzicht met CO2 equivalenties, zodat ik mijn verbruik kan vergelijken, relativeren en beslissingen kan nemen met betrekking tot landkeuze van agent hosting. | |
| Scenario: | Verbruik relativeren met land specifieke koolstofintensiteit. |
| Given | De gebruiker navigeert naar de Pipeline Carbon Calculator. |
| When | De gebruiker selecteert een CI/CD tool. |
| And | Vult geen of meerdere pipeline gerelateerde velden in. |
| And | Vult de ISO-code van een land in. |
| And | Klikt op de Calculate knop. |
| Then | Stuurt de website een verzoek naar de Pipeline Power API. |
| And | Verkrijgt de website de benodigde data. |
| And | Ziet de gebruiker het verbruik en de CO2 equivalenties gekoppeld aan de koolstofintensiteit van een land. |

|  |  |
| --- | --- |
| User Story #7 | |
| Beschrijving:  Als gebruiker wil ik het CO2 verbruik van mijn pipelines, gekoppeld de kosten van energieverbruik, inzien in het standaard overzicht met CO2 equivalenties, zodat ik mijn verbruik kan vergelijken, relativeren en financiële beslissingen kan nemen met betrekking tot uitgaven aan energie. | |
| Scenario: | Verbruik relativeren met energiekosten. |
| Given | De gebruiker navigeert naar de Pipeline Carbon Calculator. |
| When | De gebruiker selecteert een CI/CD tool. |
| And | Vult geen of meerdere pipeline gerelateerde velden in. |
| And | Klikt op de Calculate knop. |
| Then | Stuurt de website een verzoek naar de Pipeline Power API. |
| And | Verkrijgt de website de benodigde data. |
| And | Ziet de gebruiker het verbruik en de CO2 equivalenties. |
| When | De gebruiker vult de kostprijs per kWh in het overzicht |
| Then | Ziet de gebruiker de totale financiële impact van het energieverbruik van de pipelines. |

## Functionele Eisen

### Pipeline Power Meter

* REQ-1: Het programma meet de CPU-gebruik (utilization) en het CPU-verbruik.
* REQ-2: Het programma slaat de metingen op in een CSV-bestand.

### Pipeline Plugin

* REQ-3: De plugin is beschikbaar in Jenkins.
* REQ-4: De plugin laat de developer de credentials van de ElasticSearch omgeving invullen.
* REQ-5: De plugin maakt connectie met de ElasticSearch omgeving.
* REQ-6: De plugin aggregeert de metingen uit de CSV.
* REQ-7: De plugin mapped de geaggregeerde metingen naar de verwachte waardes in Tabel 2.
* REQ-8: De plugin archiveert de geaggregeerde metingen in een JSON.
* REQ-9: De plugin stuurt de gemappte metingen naar de database omgeving van de developer.

### Pipeline Power API

* REQ-10: De API accepteert een verzoek met filters
* REQ-11: De API haalt data op uit de ElasticSearch database met de filters
* REQ-12: De API mapped data naar de gewenste format.
* REQ-13: De API stuurt de gemappte data terug.

### Pipeline Carbon Calculator

* REQ-14: De webpagina laat de gebruiker een CI/CD-tool selecteren.
* REQ-15: De webpagina laat filters zien met syntax gelinkt aan de CI/CD-tool.
* REQ-16: De webpagina stuurt een verzoek op naar de Pipeline Power API.
* REQ-17: De webpagina laat verbruiksdata zien.

Bijlage 1

openapi: 3.0.3  
info:  
 title: Swagger Pipeline Power API - OpenAPI 3.0  
 description: |-  
 This is the specification of the supported endpoints in the Pipeline Power API  
 contact:  
 email: support@madebysven.com  
 license:  
 name: Apache 2.0  
 url: http://www.apache.org/licenses/LICENSE-2.0.html  
 version: "1.0"  
paths:  
 /calculator/jenkins:  
 get:  
 summary: Finds power stats of all pipelines  
 parameters:  
 - $ref: "#/components/parameters/job"  
 - $ref: "#/components/parameters/branch"  
 - $ref: "#/components/parameters/build"  
 - $ref: "#/components/parameters/country"  
 responses:  
 '200':  
 description: Successful operation  
 content:  
 application/json:  
 schema:  
 $ref: '#/components/schemas/PowerUsageResponse'  
 '403':  
 description: Access denied  
 '500':  
 description: Problem in server  
  
components:  
 parameters:  
 job:  
 in: query  
 name: job  
 schema:  
 type: string  
 description: The name of the pipeline  
 example: pipeline1  
 branch:  
 in: query  
 name: branch  
 schema:  
 type: string  
 description: The branch of the pipeline job  
 example: master  
 build:  
 in: query  
 name: build  
 schema:  
 type: integer  
 description: The build of the pipeline job  
 example: 1  
 country:  
 in: query  
 name: country  
 schema:  
 type: string  
 description: The country code where the pipeline is hosted  
 example: NL  
 schemas:  
 PowerUsageResponse:  
 type: object  
 properties:  
 kwhTotal:  
 description: Power Usage in Kilowatt-hour  
 type: string  
 co2Total:  
 description: Total CO2 Produced in grams  
 type: string  
 kwhPerRun:  
 description: Power Usage per pipeline run in Kilowatt-hour  
 type: string  
 co2PerRun:  
 description: CO2 Produced per run in grams  
 type: string  
 kwhPerHour:  
 description: Power Usage per hour in Kilowatt-hour  
 type: string  
 co2PerHour:  
 description: CO2 Produced per hour in grams  
 type: string  
 cpuUtilization:  
 description: Average CPU Utilization  
 type: string  
 pipelineRuns:  
 description: Amount of pipeline runs  
 type: string  
 runtime:  
 description: Runtime of selected pipeline(s) in seconds  
 type: string