**Nomios**

Excelsiorlaan 89.

1930 Zaventem.

**Van Mol Gerrit**

Professionele Bachelor Elektronica-ICT, afstudeerrichting ICT

Academiejaar 2021/2022

**Platform voor automatisatie van firewall configuraties en deployment**

**Stagegegevens**

**Stagiair**

Gerrit Van Mol

**Opleiding**

Elektronica-ICT, afstudeerrichting ICT

**Academiejaar**

2021/2022

**Stageperiode**

31/01 - 30/05/2022

**Stagebegeleider**

Stijn Vanbinnebeeck

**Stageplaats**

Nomios N.V. Belgium

Excelsiorlaan 89.  
1930 Zaventem.

**Mentor(en)**

Sven Sanders

**Automation platform for quick deployment of firewalls**

G. Van Mol

The purpose of this project is automating deployment of firewalls. It will simplify the job of a technician that normally installs and deploys these devices. Automating firewall deployment prevents long and difficult procedures that would otherwise take a long time to get up and running with the device.

The first chapter of this report will cover research about the best possible frameworks available, afterwards a comparison is made between frameworks. Finally, one of these frameworks will selected and used to create a platform. Based on functionality requests, the platform is made to perform the most important tasks to make automated deployment of firewalls possible.

In conclusion this platform will provide a workflow that is quick and ease of use. It will eliminate repetitive tasks and long configuration hours. Eventually the platform will be saving time of technicians and improving efficiency of the company.

Keywords: Platform, automation, framework, time saving

Voorwoord

Voor het behalen van een diploma in het derde jaar en Elektronica-ICT, is het maken van een bachelorproef een vereiste. Deze bundel is het schriftelijke verslag ervan.

Deze bachelorproef is het resultaat van de kennis en vaardigheden die ik heb verzameld zowel op school als op mijn stageplaats. Het was een leerrijke uitdaging, waarbij ik ook heel wat nieuwe ervaringen opdeed. Bij de uitwerking van de bachelorproef kreeg ik tips en ondersteuning van mijn docenten, stagebegeleiders, familie en vrienden. In de eerste plaats zou ik graag de directie van de co-hogeschool Odisee willen bedanken voor de kans die ik kreeg er mijn opleiding te volgen.

Daarnaast wil ik alle docenten bedanken voor hun fantastische begeleiding door heen de jaren van mijn opleiding, en Sanders Sven voor de fantastische begeleiding tijdens mijn stageperiode. In bijzonder wil ik de heer Donné Johan mijn docent applied-programming, data security en tal van andere vakken bedanken, om onder zijn deskundig toezicht theoretische informatie en praktische ervaring te verschaffen.

Ook wil ik graag de mensen bedanken die niet in mij geloofden voor het versterken van mijn motivatie en doorzettingsvermogen.

Tot slot gaat mijn dank naar mijn ouders Youri en Kathleen Van Mol, Ilona Van Mol en mijn grote zus Laure Van Mol voor hun aanmoediging en ondersteuning.

Met het rapport hoop ik de lezer een boeiend en leerrijk verslag te brengen van mijn bachelorproef.

Gerrit Van Mol

Kapelstraat 57 Lochristi, Juni 2022

Inhoud

[Codefragmentlijst 1](#_Toc99551552)

[Figurenlijst 2](#_Toc99551553)

[Tabellenlijst 3](#_Toc99551554)

[Afkortingenlijst 4](#_Toc99551555)

[Begrippenlijst 5](#_Toc99551556)

[Inleiding 6](#_Toc99551557)

[1 Analyse opstelling 7](#_Toc99551558)

[1.1 Platform flowchart 7](#_Toc99551559)

[1.2 Relatie diagram 9](#_Toc99551560)

[1.3 Netwerk diagram 10](#_Toc99551561)

[2 Framework 11](#_Toc99551562)

[2.1 Django geschiedenis 11](#_Toc99551563)

[2.2 Django framework 12](#_Toc99551564)

[2.3 Andere frameworks 13](#_Toc99551565)

[2.3.1. Flask 13](#_Toc99551566)

[2.3.2. Web2py 13](#_Toc99551567)

[2.3.3. CherryPy 14](#_Toc99551568)

[3 Fortimanager 14](#_Toc99551569)

[4 Front-end frameworks 14](#_Toc99551570)

[5 Ansible 15](#_Toc99551571)

[5.1 Ansible toepassingen 15](#_Toc99551572)

[5.1.1 Provisioning 15](#_Toc99551573)

[5.1.2 Configuratie management 16](#_Toc99551574)

[5.1.3 Applicatie deployment 16](#_Toc99551575)

[5.2 Continue levering 17](#_Toc99551576)

[5.3 Ansible architectuur 17](#_Toc99551577)

[6 Praktische uitwerking 17](#_Toc99551578)

[7 Hosting 18](#_Toc99551579)

[Conclusie 19](#_Toc99551580)

[Nawoord 20](#_Toc99551581)

[Literatuurlijst 21](#_Toc99551582)

[Bijlagenoverzicht 22](#_Toc99551583)

[Bijlage 1: Platform flowchart 23](#_Toc99551584)

[Bijlage 2: Relatie diagram 24](#_Toc99551585)

# Codefragmentlijst

**Geen gegevens voor lijst met afbeeldingen gevonden.**

# Figurenlijst

[Figuur 1: Platform logica overzicht 8](https://hubkaho-my.sharepoint.com/personal/gerrit_vanmol_student_odisee_be/Documents/Documenten/Odisee_2021-2022/BachelorProef/Papers/V1_AutomatisatiePlatform.docx#_Toc99128677)

[Figuur 2: Connectie relatie diagram 9](https://hubkaho-my.sharepoint.com/personal/gerrit_vanmol_student_odisee_be/Documents/Documenten/Odisee_2021-2022/BachelorProef/Papers/V1_AutomatisatiePlatform.docx#_Toc99128678)

[Figuur 3: Netwerk/hosting diagram 10](https://hubkaho-my.sharepoint.com/personal/gerrit_vanmol_student_odisee_be/Documents/Documenten/Odisee_2021-2022/BachelorProef/Papers/V1_AutomatisatiePlatform.docx#_Toc99128679)

[Figuur 4: Django/Python logo [9] 11](https://hubkaho-my.sharepoint.com/personal/gerrit_vanmol_student_odisee_be/Documents/Documenten/Odisee_2021-2022/BachelorProef/Papers/V1_AutomatisatiePlatform.docx#_Toc99128680)

[Figuur 5: Django tijdlijn [8] 12](https://hubkaho-my.sharepoint.com/personal/gerrit_vanmol_student_odisee_be/Documents/Documenten/Odisee_2021-2022/BachelorProef/Papers/V1_AutomatisatiePlatform.docx#_Toc99128681)

[Figuur 6: Flask logo [3] 13](https://hubkaho-my.sharepoint.com/personal/gerrit_vanmol_student_odisee_be/Documents/Documenten/Odisee_2021-2022/BachelorProef/Papers/V1_AutomatisatiePlatform.docx#_Toc99128682)

[Figuur 7: CherryPy logo [10] 14](https://hubkaho-my.sharepoint.com/personal/gerrit_vanmol_student_odisee_be/Documents/Documenten/Odisee_2021-2022/BachelorProef/Papers/V1_AutomatisatiePlatform.docx#_Toc99128683)

[Figuur 8: Ansible stage levenscyclus [6] 15](https://hubkaho-my.sharepoint.com/personal/gerrit_vanmol_student_odisee_be/Documents/Documenten/Odisee_2021-2022/BachelorProef/Papers/V1_AutomatisatiePlatform.docx#_Toc99128684)

[Figuur 9: Zero downtime applicatie-update [11] 16](https://hubkaho-my.sharepoint.com/personal/gerrit_vanmol_student_odisee_be/Documents/Documenten/Odisee_2021-2022/BachelorProef/Papers/V1_AutomatisatiePlatform.docx#_Toc99128685)

# Tabellenlijst

[Tabel 1: Django versie tijdlijn [2] 12](#_Toc100751657)

# Afkortingenlijst

ZTP Zero Touch Provisioning

DB Database

MySQL My Structured Query Language

HTML Hyper Text Markup Language

UI User Interface

MVC Model View Controller

IDE Integrated Development Environment

SSL Secure Sockets Layer

API Application Programming Interface

JSON JavaScript Object Notation

ORM Object Relational Mapper

CRUD Create, Read, Update, Delete

CSRF Cross-site Request Forgery

XSS Cross-site Scripting

IDE Integrated Development Environment

DVCS Distribution Version Control System

CI Continuous Integration

CD Continuous Development

CLI Command Line Interface

# Begrippenlijst

Variabele Element van bepaald type dat een tijdelijke waarde bijhoudt

Redundant Herhaling/dubbele gegevens

Property Eigenschap van een object

Provisioning Het proces voor opstellen van IT-infrastructuur

Batch processing Runnen van “jobs” dat geen gebruikers interactie nodig heeft

Staging Een replica van de productie omgeving om testen uit te voeren

# Inleiding

Dit project wordt een platform dat de opzet van firewalls automatiseert. Momenteel werkt de field engineer op een tijd consumerende manier waarbij alle configuraties van elk toestel handmatig gebeuren. Het platform geeft de technieker(s) de mogelijkheid snel en efficiënt meerdere toestellen te plaatsen binnen een bepaald tijdsframe. Aan de hand van zero touch provisioning zullen de techniekers weinig of geen voorkennis moeten hebben om een firewall toestel te plaatsen en operationeel te krijgen. Als een field engineer on-site is en geen basis configuratie aanwezig is op het toestel kan deze worden gedownload van het platform naar een USB-stick. Eenmaal de configuratie op de stick staat, kan deze worden overgezet naar de firewall. Met de basis configuratie aanwezig kan het platform verdere connectie maken met de firewall, om updates uit te sturen met verdere configuratie.

Er wordt onderzocht met welke technologieën en protocollen het platform best werkt om communicatie met de externe apparaten mogelijk te maken. Hoe kunnen configuraties het efficiëntst worden beheerd. Of het mogelijk is meerdere connecties tegelijk tot stand te brengen tussen het platform en toestellen terplekke. De klant vraagt om gebruik te maken van het Django framework, Fortimanager en/of Ansible. Andere frameworks en automatisatie engines worden daarnaast ook nog vergeleken.

Bij deze opdracht heeft het bedrijf een aantal eisen. Deze omvatten dan vooral een volledig functionele login voor administrator en engineers. Alsook had men graag de geautomatiseerde basis setup van een firewall waarbij connectiviteit zoals een ping naar Fortigatemanager functioneert zoals het hoort.

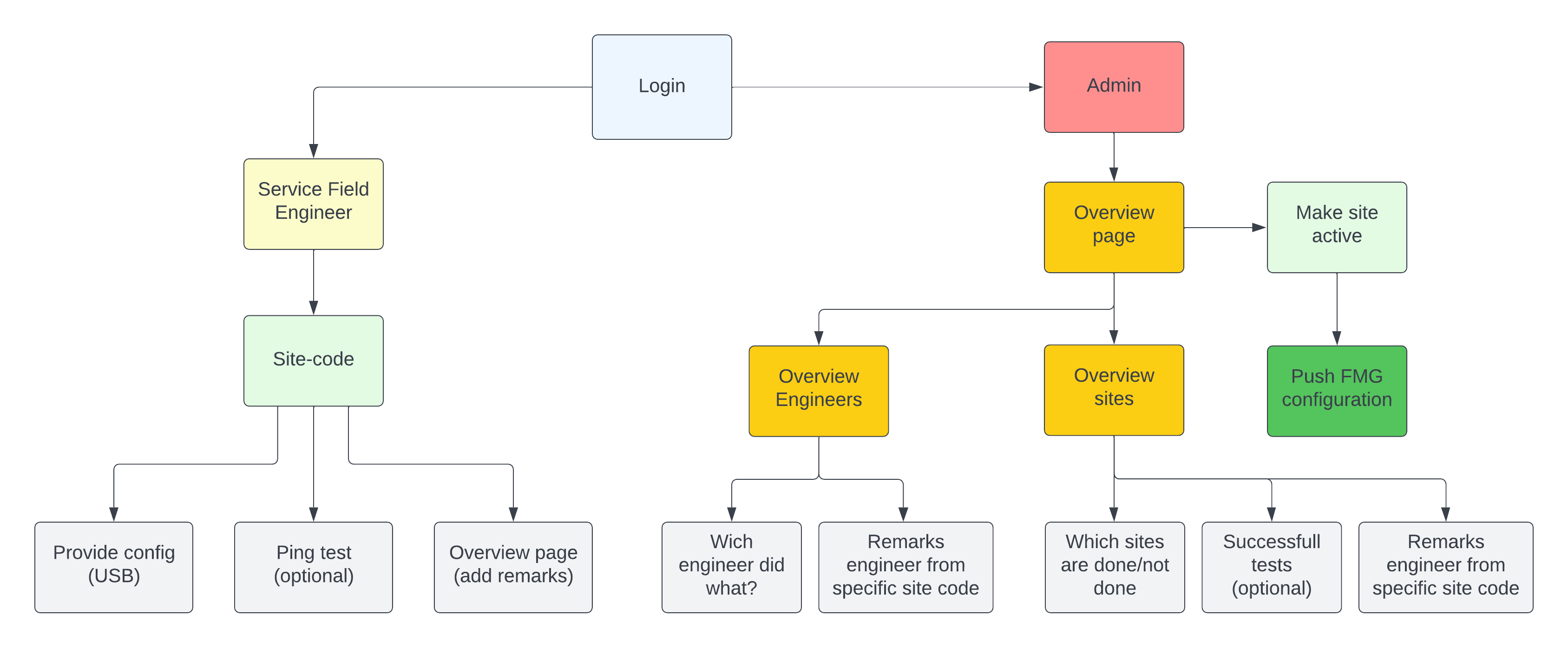
De klant heeft daarnaast geen vooropgestelde beperkingen of technische randvoorwaarden waaraan moet worden voldaan.

# Analyse opstelling

In volgende subhoofdstukken worden relatie diagrammen en logica processen beschreven. Deze processen zijn vastgelegd en vooropgesteld door het externe bedrijf waar deze opdracht voor uitwerkt wordt. Met de opgestelde schema’s, flowcharts en diagrammen worden processen visueel verduidelijkt.

## 1.1 Platform flowchart

Omdat er binnen het bedrijf een bepaalde werk structuur moet worden aangenomen voor het uitrollen en registreren van firewalls is deze workflow ook vastgelegd en gevisualiseerd in een flowchart. In onderstaande flowchart (Figuur 1: Platform logica overzicht) is de huidige logica zichtbaar, met daarbij de minimumpagina’s die gebruikers kunnen zien op het platform afhankelijk van welke soort rol een gebruiker heeft.



Figuur : Platform logica overzicht

## Relatie diagram

Het relatie diagram (Figuur 2: Connectie relatie diagram) visualiseert hoe de verschillende technologieën zullen samenwerken en geconnecteerd worden. Onder deze technologieën verstaat men dan het backend, front-end en netwerk gedeelte. Er wordt een beter beeld gegeven van de cruciale componenten en velden die in het gehele project zullen moeten bijgehouden worden. Op die manier kan door de field engineer sneller maar ook beter een correcte en werkende omgeving opgebouwd worden.

Figuur : Connectie relatie diagram

## Netwerk diagram

Nadat de vermelde delen van uit de relatie diagram (Figuur 2: Connectie relatie diagram) actief en operationeel zijn zal er een productie omgeving worden gecreëerd. Productieversies worden aan de hand van vooropgestelde netwerkinfrastructuur publiek gemaakt. Op die manier kunnen werknemers en andere partners gebruik maken van het gerealiseerde platform/service. De netwerk diagram (Figuur 3: Netwerk/hosting diagram) zal ook opnieuw een beter beeld geven van welke componenten nodig zijn, hoe ze communiceren intern en hoe ze beschikbaar worden voor de buitenwereld.

(Niet definitief wordt later toegevoegd).

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Figuur : Netwerk/hosting diagram

# Framework

Om het platform uit te werken is er een framework nodig dat kan worden gebruik als fundamenteel center punt om verder op te bouwen. Aan de hand van dit framework zal het platform worden uitgewerkt. Er werd voor dit project extern gekozen voor het Django framework waarbij een potentiële overweging voor eventuele andere frameworks worst uitgesloten.

Hoe dan ook zal er in dit hoofdstuk worden terug gekeken op wat de geschiedenis is van Django, waar het vandaan komt en hoe het is ontstaan. Alsook zullen meerdere voor-/nadelen worden overlopen en vergelijken met andere populaire frameworks. Op die manier wordt er een beter zicht gecreëerd over de verschillende frameworks en voor welke toepassingen ze best ingezet kunnen worden.

## 2.1 Django geschiedenis

Origineel is het Django framework ontstaan en ontworpen door “Lauwrence journal world” in 2003 [1]. In 2004 werd het een open source project en kreeg het de naam Django. De naam wordt uitgesproken als “Jango” en is vernoemd achter de jazzgitarist Django Reinhardt. Een web team dat nieuwswebsites beheerden maakten gebruik van veel herhaaldelijke code en ontwerpen. Deze herhaaldelijke code werd uiteindelijk uitgewerkt als een algemeen web development framework. Het framework werd steeds meer gekend en verbeterd waarbij de eerste officiële versie (v0.90) werd uitgerold in november 2005. Ondertussen zitten we aan versie (v3.2.12) sinds december 2021. Bij elke versie worden mogelijke problemen weggewerkt en features toegevoegd zoals nieuwe templates, database typen, etc. Django is intussen een veelzijdig framework dat de mogelijkheid biedt meerdere soorten websites te creëren.



Figuur : Django tijdlijn [10]

Volgende tabel (Tabel 1: Django versie tijdlijn [2]) geeft een gedetailleerdere weergave van de voorbije Django versies.

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Notes** |
| 0.90 | 16 Nov 2005 |  |
| 0.91 | 11 Jan 2006 | "new-admin" |
| 0.95 | 29 Jul 2006 | "magic removal" |
| 0.96 | 23 Mar 2007 | "newforms", testing tools |
| 1.0 | 3 Sep 2008 | API stability, decoupled admin, unicode |
| 1.1 | 29 Jul 2009 | Aggregates, transaction based tests |
| 1.2 | 17 May 2010 | Multiple db connections, CSRF, model validation |
| 1.3 | 23 Mar 2011 | Class based views, staticfiles |
| 1.5 | 26 Feb 2013 | Time zones, in browser testing, app templates. |
| 1.6 | 6 Nov 2013 | Python 3 Support, configurable user model |
| 1.7 | 2 Sep 2014 | db transaction management, connection pooling. |
| 1.8 LTS | 2 Sep 2014 | Migrations, application loading and configuration. |
| 1.8 LTS | 1 Apr 2015 | Native support template(s) engines. |
| 1.9 | 1 Dec 2015 | Automatic password validation. New styling for admin UI. |
| 1.10 | 1 Aug 2016 | Full text search for PostgreSQL. New-style middleware. |
| 1.11 LTS | 4 Apr 2017 | Last support Python 2.7. |
| 2.0 | 2 Dec 2017 | First Python 3-only release, Simplified URL routing syntax, Mobile friendly admin. |
| 2.1 | 1 Aug 2018 | Model "view" permission. |
| 2.2 LTS | 1 Apr 2019 | Security release. Supported until at least April 2022 |
| 3.0 | 2 Dec 2019 | ASGI-support |
| 3.1 | 4 Aug 2020 | Asynchronous views and middleware |
| 3.2 LTS | 6 Apr 2021 | Extended Support until at least April 2024 |
| 4.0 | 7 Dec 2021 | Extended Support until at least April 2023 |

Tabel : Django versie tijdlijn [2]

## 2.2 Django framework

Django is een gratis open source framework, gemaakt om te focussen op de development van onderhoudbare websites. Sinds 2005 geeft het de mogelijkheid om snel maar ook veilige en schaalbare websites te ontwikkelen in de Python programmeertaal. Het framework heeft een zeer actieve gemeenschap waar elke ontwikkelaar steeds terecht kan voor vragen. Daarnaast is de documentatie ook zeer uitgebreid en goed onderhouden. Het framework heeft ook een non-profit waarbij meer dan 10 000 packages worden ondersteund dat voor een rijke set aan features zorgt. Alle packages/features worden steeds frequent onderhouden en ondersteund. Met een rijke set aan features en packages, bijvoorbeeld ingebouwde API’s, CMS, UA, CAPTCHA-ondersteuning en form validatie.

Graphical user interface, application

Description automatically generatedHet makkelijke aan Django is dat alle achterliggende delen al inbegrepen zijn, zo is het niet nodig om een backend te maken aan de hand van API’s, JavaScript, etc. Met Django kan er worden gefocust op de applicatie zelf en is minder werk vereist dan dat de ontwikkelaar van scratch zou moeten beginnen [3] [4].

Figuur : Django/Python logo [11]

Zoals eerder vermeld werd door de opdrachtgever opgelegd om gebruik te maken van het Django framework. Hierdoor is de eigen keuze voor gebruik van een alternatief framework onmogelijk. Hoe dan ook worden mogelijke alternatieve frameworks niet uitgesloten en in een later hoofdstuk in de kijker gezet met hun voor- en nadelen. Hoe dan ook is het gekozen framework voor de uitwerking van dit project beslist door meerdere achterliggende redenen. Het framework laat toe om een volledig datamodel op te bouwen in Python zonder dat men MySQL nodig heeft. Het JSON-formaat en andere vaak voorkomende data typen worden actief ondersteund door Django MySQL. Gebruik makende van een ORM kan Django traditionele databank structuren en tabellen converteren in Python klassen. Op deze manier kan men sneller en gemakkelijker werken in één en dezelfde Python omgeving.

Django heeft de structuur van MVC-MVT waarbij men krachtige componenten van het met het framework kan maken [5]. Aan de hand van de modellen in MVC kunnen de gecreëerde webapplicaties worden geconnecteerd met Django en kan de data worden beheerd. Database velden worden vaak omgezet in klasse attributen. Deze attributen zijn ook gedefinieerd in Python, waardoor ze gemakkelijk worden beheerd met de ingebouwde Django database. Het web framework biedt een snelle mogelijkheid om te integreren met een applicatie database. Aan de hand van CRUD kan men meerdere operaties uitvoeren op duurzame gegevens zoals het maken, lezen, bijwerken en verwijderen van gegevens. Naast CRUD zijn andere features zoals user management, software administratie features en anderen steeds integreer baar via packages [3].

## Andere frameworks

Naast Django zijn er tal van andere frameworks die men zou kunnen kiezen. Flask is bijvoorbeeld ook één van de meest populaire frameworks voor web development naast Django. Het gekozen framework heeft tal van voordelen maar ook een aantal nadelen waarom men het juist niet zou kiezen. Zo is bijvoorbeeld de volledige projectstructuur niet te vermeiden wanneer men een enkele webpagina wil maken. Onderstaand worden een drietal populaire frameworks vergeleken met Django [5].

### 2.3.1. Flask

Afbeelding met tekst

Automatisch gegenereerde beschrijvingFlask is een micro-framework dat vaak gebruikt wordt voor kleiner projecten. Het biedt de mogelijkheid snel een schaalbare website te creëren. Dit framework komt vaker voor bij startups, aangezien men kan starten met het ontwikkelen van een basis applicatie. Naar mate het aantal gebruikers stijgt kan de applicatie mee schallen om verkeer te compenseren. Het framework bouwt over bestaande API’s, maakt ook gebruik van microservices en micro-front-end(s) om laadtijden van een pagina of website te onderdrukken.

Figuur : Flask logo [5]

Meerdere standaardvariabelen zijn vlot toegankelijk en bewerkbaar gemaakt vanuit de basisopstelling. Flask heeft net als Django ook algemene beveiligingsmaatregelen getroffen zoals session-based management, paswoord hashing, http-authenticatie en CSRF om te beschermen tegen mogelijke zwakten in de website. Het is een relatief snel framework en heeft een simpel web interface voor front-end waarbij weinig configuratie vereist is [5].

### 2.3.2. Web2py

Web2py is ook een web gebaseerd framework dat de ontwikkelaar toe laat om meerdere applicaties te gaan uitrollen en onderhouden van uit de browser. Het framework is cross-platform wat wil zeggen dat het compatibel is op meerdere toestellen. Daarnaast maakt Web2py geen gebruik van extra packages of dependencies maar enkel de standaard Python bibliotheek. Bij installatie en configuratie zijn er geen vooropgestelde voorwaarden om het framework operationeel te krijgen. Net als Django heeft Flask ook een aantal ingebouwde beveiligingsmitigaties zoals, XSS, SQL-injectie, etc.

Web2py biedt een volledige eigen IDE waarbij men websites gemakkelijk kan aanpassen ook na dat ze online actief zijn [5].

### 2.3.3. CherryPy

CherryPy is een object-oriented web development framework. Het bevat een implementatie van de Ruby programmeertaal en integreert het Ruby on Rail’s routing systeem in Python. Hierdoor kan men URL’s mappen aan controllers. CherryPy is ook gemaakt voor korte ontwikkelingstijden en hoge productiviteit om met weinig code mooie resultaten te bereiken. Vergeleken met Django is CherryPy een zeer ligt Python framework en één van de oudste. In tegenstelling tot Django is CherryPy eerder geschikt om one-page statische websites te maken en de mogelijkheid geeft om meerdere HTTP-request(s) tegelijk te verwerken. Ook kunnen meerdere HTTP-servers tegelijk naast elkaar draaien voor een redundante opstelling.

Figuur : CherryPy logo [12]

# Fortimanager

(Moet nog worden uitgewerkt)

Wat is fortimanager, waarvoor dient het, zijn er alternatieven, waarom is het gekozen?

# Front-end frameworks

(Moet nog worden uitgewerkt)

Moet/kan er nog een extra front-end framework op Django? Welke zijn integreer baar?

# Ansible

Ansible is een open source automatisatie engine die repetitieve taken zoals configureren van servers of uitrol van applicaties kan gaan automatiseren. Aan de hand van CI/CD kunnen snel vernieuwde versies van een applicatie online gebracht worden zonder enige downtime. Ansible brengt meer consistentie, betrouwbaarheid en schaalbaarheid naar een IT-omgeving. Met de YAML-taal is het gemakkelijker om scripts/configuraties te lezen en schrijven als standaard Engels.

Gebruikmakende van een enkele controle machine kunnen meerdere machines worden aangestuurd. Alle machines die onder de controle machine vallen worden aangesproken via OpenSSH. Met SSH kan de controle machine updates en configuraties uitvoeren op andere machines. Het is mogelijk om naast SSH-authenticatie gebruik te maken van Kerberos, LDAP of andere authenticatie managementsystemen [6].

Ansible kan dus worden gebruikt door meerdere applicaties zoals provisioning van virtuele machines, configuratie management, applicatie uitrol, beveiliging automatisatie en meer. Om de Django applicatie vlot uit te rollen en online te krijgen zal de Ansible engine een belangrijke rol spelen. Mogelijks in samenwerking met andere services of achterliggende Linux machines.

Figuur : Ansible stage levenscyclus [7]

## 5.1 Ansible toepassingen

Onderstaande hoofdstukken zullen dieper in gaan op mogelijke manieren om Ansible te gebruiken.

### 5.1.1 Provisioning

Bare metal-servers hebben af en toe nood aan provisioning. Met Ansible kan men gebruik veel zaken integreren zoals datacenterbeheer tools om servers aan te spreken en een provisioning te starten. Bij gebruik van verschillende hypervisor(s), virtuele opslag/netwerken kan Ansible een diverse selectie van modules en omgevingen ondersteunen. Door gebruik van sommige modules is het mogelijk om gemakkelijker te gaan schalen.

Naast bare metal servers en hypervisor(s) is het ook mogelijk om netwerkinfrastructuur en cloudoplossingen te leveren. De netwerk automatiseringsmogelijkheden zorgen voor validatie en continue werking van fysieke netwerkapparaten.

Wanneer Ansible een rol speelt voor een (publieke of private) Cloud omgeving is het mogelijk om niet alleen de clouddiensten te leveren, maar ook de onderliggende infrastructuur en applicaties binnen de Cloud te onderhouden [7].

### 5.1.2 Configuratie management

Ansible is een state-driven resourcemodel, wat inhoudt dat het de toestand van de systemen en services beschrijft. De huidige status van de doelmachine is niet relevant, aangezien Ansible de status van de machine naar wens zal transformeren. Dit leidt tot minder potentiële fouten in vergelijking met script gebaseerde oplossingen, omdat deze vaak onomkeerbare acties veroorzaken. Ansible heeft ook de mogelijkheid om de configuratie te valideren op mogelijke syntaxisfouten [8].

### 5.1.3 Applicatie deployment

Zoals eerder al werd vermeld is het mogelijk om te beschrijven wat de gewenste staat is van een machine. De gewenste staat van een machine is gedefinieerd in een Ansible “Playbook”. Playbooks worden uitgevoerd tegen een groep/enkele host waarbij de gewenste staat voor elke host hetzelfde is. Op die manier kan men configuratie fouten vermijden en is de configuratie met zekerheid op elke machine hetzelfde. Op die manier maakt Ansible herhalende opdrachten zoals jaarlijks onderhoud en andere configuraties betrouwbaarder.

Diagram

Description automatically generatedPlaybooks worden verzonden via SSH naar de gedefinieerde doel host met de OS-compatibele modules, toestel specifieke modules zijn vergelijkbaar met opdrachten in de CLI voor het besturingssysteem van de doelmachine. Zodra de machine de module heeft ontvangen, worden ze ter plaatse uitgevoerd, eenmaal de taak is voltooid (staat geüpdatet) wordt de module verwijderd van de doel host [8].

Figuur : Zero downtime applicatie-update [13]

Zoals de lezer kan zien in bovenstaande afbeelding (Figuur 9) is het mogelijk om applicaties en updates uit te rollen, zonder dat de gebruiker enige downtime opmerkt. De gebruiker merkt geen downtime omdat elke server of servergroep apart worden geüpdatet door behulp van de load balancer. Wanneer de ontwikkelaar gebruik maakt van DVCS is het mogelijk om downloadbare artifacts te maken van actieve applicaties. Hoe dan ook, het is ook mogelijk om gebruik te maken van REST API’s waarbij een aantal variabelen worden aangepast op een andere service. Als er eenmaal variabele aanpassingen zijn of de uitrol is van een nieuwe update kunnen mail services worden opgesteld of een pushnotificatie sturen. Op die manier is de ontwikkelaar steeds up-to-date en weet hij/zij of de aanpassing al is doorgevoerd of niet.

## 5.2 Continue levering

Met de stage en testmethode kunnen Ansible-"plays" worden gevalideerd en getest. In het inventory bestand is het mogelijk om omgevingen in te delen in verschillende groepen van machines. Dit maakt het mogelijk om test en productiemachines te definiëren. Het is echter aangeraden om productie en testmachines te definiëren in aparte invenotory bestanden, op die manier worden enige verrassingen voorkomen wanneer een configuratie wordt uitgerold. Wanneer een "play" klaar is, kan het worden getest op de staging machines. Als de test slaagt, kan ervoor gekozen worden om ook automatisch de “play” op productiemachines uit te voeren waarbij het eindproduct beschikbaar wordt voor gebruikers.

In een later deel van dit hoofdstuk zullen “plays” in meer detail worden besproken. Op dit moment, weet dat een “play” bestaat uit een set aan geschreven instructies in de YAML-taal om een ​​gewenste machineconfiguratie in te tellen. Deze machines kunnen bestaan uit monitoringsystemen, netwerk toestellen, load balancers, webservers en andere soorten toestellen. Een veelvoorkomende manier om Ansible te gebruiken is met een CI-systeem waarbij succesvolle builds van een play worden gepubliceerd [9].

Taken die plaatsvinden bij het testen:

* De CI zorgt ervoor dat Ansible een playbook uitvoert om een ​​staging omgeving te implementeren met de nieuwe apparaat configuratie of toepassing
* Wanneer de test slaagt, kan er een melding worden verzonden om de uitrol te bevestigen naar productieomgeving
* Na publicatie kan de configuratie of applicatieversie worden opgeslagen in een artefact op de CI-server voor latere raadpleging of andere doeleinden.

## Ansible architectuur

# Praktische uitwerking

(Moet nog worden uitgewerkt)

# Hosting

Hoe kan men dit platform meerdere malen gaan opstellen op andere locaties en de configuratie automatiseren? Is dat mogelijk met Ansible en wat voor servers of services zijn nodig om een Django platform te lanceren? Hoe zal een service engineer verbinding kunnen maken met het platform (via VPN of andere manier)?

# Handleiding

Hier zullen delen van het platform worden overlopen, waarbij handelingen en stappen worden verduidelijk mocht dat nodig zijn om bepaalde acties te bereiken.

# Conclusie

(Moet nog worden uitgewerkt)

● Welke doelstellingen zijn bereikt?

● Als er bepaalde doelstellingen niet bereikt zijn, wat zijn de redenen hiervoor? Hoe kon dit voorkomen worden?

● Gaan jouw bereikte resultaten in de toekomst toegepast worden? Heb je daar stappen voor ondernomen?

● Hoe kunnen je resultaten in de toekomst nog verbeterd worden?

● Is er nog vervolgonderzoek nodig?

● Misschien heb je ervaren dat bepaalde werksituaties minder efficiënt aangepakt worden. Heb jij tips om deze op te lossen?

# Nawoord

(Moet nog worden uitgewerkt)

● Hoe verliep de samenwerking met de collega’s?

● Hoe verliep het rapporteren naar collega’s en verantwoordelijken?

● In welke mate kon je een probleem zelfstandig aanpakken?

● Wat heb je bijgeleerd over bedrijfseigen afspraken i.v.m. veiligheid, milieu, kwaliteit, …?

● Op welke manier heb je het eigen werk geëvalueerd en in welke mate deed je dat systematisch

# Literatuurlijst

|  |  |
| --- | --- |
| [1] | JavaPoint, „Django history - javapoint,” [Online]. Available: https://www.javatpoint.com/django-tutorial. [Geopend 22 03 2022]. |
| [2] | Wikipedia contributors, „Django (web framework),” 05 04 2022. [Online]. Available: https://en.wikipedia.org/wiki/Django\_(web\_framework). [Geopend 13 04 2022]. |
| [3] | I. Education, „What is Django - IBM,” 29 03 2021. [Online]. Available: https://www.ibm.com/cloud/learn/django-explained. [Geopend 21 03 2022]. |
| [4] | „Django introduction,” MDN contributors, 18 02 2022. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction. [Geopend 20 03 2022]. |
| [5] | J. Patel, „Django Alternatives: Top 5 Python Frameworks To Use in 2022,” 02 03 2022. [Online]. Available: https://www.monocubed.com/blog/django-alternatives/. [Geopend 22 03 2022]. |
| [6] | R.H Ansible, „About Ansible - Ansible documentation,” [Online]. Available: https://docs.ansible.com/ansible/latest/index.html. [Geopend 03 11 2021]. |
| [7] | R.H Ansible, „Red Hat Ansible use case - Provisioning,” [Online]. Available: https://www.ansible.com/use-cases/provisioning. [Geopend 03 11 2021]. |
| [8] | R.H. Ansible, „Ansible for Configuration management,” [Online]. Available: https://www.ansible.com/use-cases/configuration-management. [Geopend 03 11 2021]. |
| [9] | R.H. Ansible, „Ansible For Continuous Delivery,” [Online]. Available: https://www.ansible.com/use-cases/continuous-delivery. [Geopend 03 11 2021]. |
| [10] | A. Technologies, „Django timeline – Skillsets, Budget and Benefits,” 05 01 2021. [Online]. Available: https://www.agiratech.com/django-web-development-skillsets-budget-and-benefits. [Geopend 21 03 2022]. |
| [11] | A. Martin, „Django logo,” 03 08 2021. [Online]. Available: https://njkhanh.com/flask-django-or-pyramid-choose-the-right-python-framework-for-your-project-p5f33363134. [Geopend 24 03 2022]. |
| [12] | R. U. J. Atienza, „CherryPy - A Minimalist Python Web Framework | Unixmen,” 11 02 2022. [Online]. Available: https://www.unixmen.com/cherrypy-minimalist-python-web-framework/. [Geopend 23 03 2022]. |
| [13] | R.H. Ansible, „Ansible for application deployment,” [Online]. Available: https://www.ansible.com/use-cases/application-deployment. [Geopend 03 11 2021]. |

# Bijlagenoverzicht

Bijlage 1: Platform flowchart

Bijlage 2: Relatie diagram

Bijlage 3: Netwerk/hosting diagram

## Teams Description automatically generated with low confidenceBijlage 1: Platform flowchart

## Full relation diagramBijlage 2: Relatie diagram