Scriptie

Automatisch .NET 6 API's deployen op een veilige en consistente manier

Een onderzoek naar de beste manier om .NET 6 API's te deployen en veilige inloggegevens te genereren voor de .NET 6 API's en backend systemen om goed te functioneren.

Naam:	Danny Cao
Studentnummer:	1715437
Bedrijf:	Social Brothers
Begeleider(s):	Hugo Kamps
	Jonne Kampen
School:	Hogeschool Utrecht
1e examinator:	Anna Goessens
2e examinator:	Philippine Waisvisz

Datum:	30/05/2022
Versie:	1.0

Versiebeheer

Versie	Datum	Omschrijving
0.1	24-01-2022	Eerste concept scriptie
0.2	31-01-2022	Feedback verwerkt van docentbegeleider: - Inleiding aangepast - Organisatorische context aangepast - Probleemstelling aangepast
0.3	09-02-2022	Feedback verwerkt van docentbegeleider: - Deelvragen 1 t/m 3 aangepast Deelvraag 4 toegevoegd
0.4	14-02-2022	Deelvraag 2: extra eisen toegevoegd Deelvraag 3: bronnen aangepast
0.5	28-02-2022	Deelvraag 5 toegevoegd Samenvatting toegevoegd Inleiding aangepast Conclusie en aanbeveling toegevoegd Verantwoording resultaten toegevoegd
0.6	04-03-2022	Feedback verwerkt van docentbegeleider: - Onnodige informatie verwijderd - Randvoorwaarden verwijderd - Stakeholders ingekort Feedback verwerkt van bedrijfsbegeleider: - Tabellen gewijzigd voor consistentie Scriptie ingekort naar 50 pagina's Deelvraag 4 aangepast Deelvraag 5 aangepast Extra bronnen toegevoegd Bijlagen toegevoegd met interviews
0.7	8-05-2022	Feedback verwerkt van collega's: - Titel en ondertitel toegevoegd - Spelfouten verbeterd Figuur 25 aangepast
0.8	16-05-2022	Feedback verwerkt van docentbegeleider: - Interview volgorde aangepast - Interview bijlage aangepast - Deelvraag 3: veiligheid toegevoegd
0.9	25-05-2022	Hoofdstuk "proof of concept" toegevoegd
1.0	30-05-2022	Feedback verwerkt van bedrijfsbegeleider: - Proof of Concept aangepast - Conclusie en aanbeveling aangepast

Samenvatting

Er zijn verschillende manieren om software te deployen. Echter, niet alle manieren zijn even veilig en consistent. Ook kost het tijd als het proces niet geautomatiseerd is. Dit heeft vooral te maken met het aantal beschikbare tools om dat te doen en omdat het onduidelijk is wat een goede manier van standaardiseren is. Social Brothers heeft het deployen van de software nog niet gestandaardiseerd en geautomatiseerd en weet niet goed hoe het dat het beste kan doen.

Om die reden heeft Social Brothers de opdracht gegeven te onderzoeken wat de beste manier is om een .NET 6 API automatisch te deployen op een veilige en consistente manier. Ook dient te worden onderzocht hoe tijdens het deploymentproces het beste inloggegevens te genereren zijn voor databases en andere systemen waarvoor inloggegevens vereist zijn. De inloggegevens moeten zo veilig mogelijk worden opgeslagen en toegankelijk zijn voor bevoegde mensen. Dit onderzoek geeft Social Brothers inzicht in het standaardiseren en automatiseren van het deploymentproces, zodat het veilig, consistent en tijdbesparend .NET 6 API's kan deployen.

Om de opdracht te realiseren, zijn eerst het oude en huidige deploymentproces bekeken. Daarna zijn de backend developers geïnterviewd over wat ze in het huidige deploymentproces wel en niet willen behouden. Het blijkt dat de backend developers de verschillende tools die DigitalOcean aanbiedt willen behouden, zoals de monitoring en het beheer van toegang tot de droplet. Daarentegen willen ze niet de onnodige handmatige acties behouden.

Na het interviewen van de backend medewerkers is in literatuur gezocht naar manieren om automatisch wachtwoorden te genereren. Na het vergelijken van de verschillende manieren is Urandom gekozen als de beste oplossing, omdat het op het besturingssysteem van de server staat en niet apart geïnstalleerd hoeft te worden. Ook voldoet het aan de eisen om een veilig wachtwoord te genereren.

Vervolgens is onderzocht wat nodig is om een .NET 6 API automatisch te deployen. Het blijkt dat hiervoor een CI/CD tool en een hosting platform nodig zijn. Verschillende CI/CD tools en hosting platforms zijn met elkaar vergeleken en de backend developers van Social Brothers zijn nogmaals geïnterviewd over hun voorkeur. Het blijkt dat de backend developers een voorkeur hebben voor DigitalOcean als hosting platform. Bij de CI/CD tools waren er geen duidelijke voorkeuren die volledig aansloten op de eisen van Social Brothers. Uiteindelijk is voor Jenkins gekozen, omdat het aan de meeste eisen voldoet.

Vervolgens is een proof of concept gebouwd om een .NET 6 API te deployen. Bij het bouwen van de proof of concept is rekening gehouden met de eisen en wensen van de backend-afdeling van Social Brothers.

Definities

In Tabel 1 hieronder worden afkortingen en woorden gedefinieerd die in deze scriptie gebruikt worden.

Tabel 1. Definities

Afkorting / Woord	**Betekenis**
Software deployment	Het uitrollen van software naar een externe server voor intern of extern gebruik.
Proof of concept	Een softwareoplossing die voldoet aan de specificaties van de opdrachtgever.
VPS	Een Virtual Private Server (VPS) is een fysieke server, die wordt opgedeeld in kleinere virtuele servers. Een VPS biedt de volledige vrijheid: van het beheer van de server, de keuze van het OS, tot aan het installeren van de benodigde software en biedt daarnaast de mogelijkheid om de software up-to-date te houden en voldoende te beveiligen tegen misbruik (TransIP, z.d.).
.NET 6	.NET is een opensource-ontwikkelaarsplatform, gemaakt door Microsoft om verschillende soorten applicaties te bouwen. .NET 6 is de .NET-versie uitgebracht op 08-11-2021.
User story	Een user story is een hulpmiddel bij Agile-softwareontwikkeling dat wordt gebruikt om een beschrijving van een softwarefunctie vast te leggen vanuit het perspectief van een gebruiker. Het gebruikersverhaal beschrijft het type gebruiker, wat deze wil en waarom.

Inhoudsopgave

Inleiding

Deze scriptie is geschreven tijdens de afstudeerstage bij Social Brothers. De afstudeerder, Danny Cao, heeft van november tot en met juni 2022 aan dit document gewerkt.

In Hoofdstuk 1 worden het bedrijf Social Brothers en de stakeholders beschreven. In Hoofdstuk 2 worden de probleemstelling en afbakening geformuleerd. In Hoofdstuk 3 komt de doelstelling aan bod en in Hoofdstuk 4 staan de onderzoeksvragen. Vervolgens is in Hoofdstuk 5 aandacht voor de resultaten van de onderzoeksvragen. Op basis van de onderzoeksresultaten wordt in Hoofdstuk 6 de hoofdvraag beantwoord. In Hoofdstuk 7 staan aanbevelingen en vervolgstappen voor het bedrijf. In Hoofdstuk 8 staat de verantwoording en in Hoofdstuk 9 de evaluatie van de procesgang. De bijlagen bevatten informatie die te gedetailleerd was om op te nemen in de scriptie zelf.

1 Organisatorische context

1.1. Het bedrijf

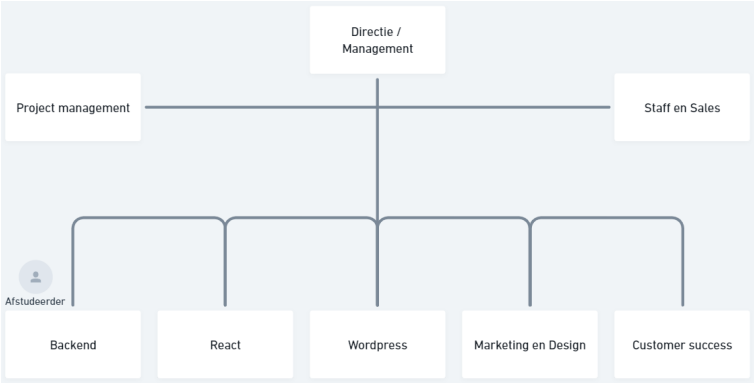
Social Brothers is een digital agency opgericht in 2011 en gevestigd in Utrecht. Het bedrijf is gespecialiseerd in het ontwikkelen van online oplossingen zoals websites, apps en datagedreven marketingcampagnes.

Social Brothers is opgericht door Steven Ammerlaan en Rob Ammerlaan. Het bedrijf heeft een vestiging in Utrecht en een kantoor in Rotterdam voor meetings met klanten die niet naar Utrecht kunnen komen.

De missie van Social Brothers is om klanten verder te helpen binnen digital marketing, design, development en innovatieve oplossingen. De visie van Social Brothers is om de komende jaren verder te groeien om de missie beter te kunnen uitvoeren (Social Brothers, 2020).

Social Brothers is een platte organisatie met een innovatieve sfeer. Het bedrijf staat bijvoorbeeld open voor nieuwe technologievoorstellen.

Social Brothers heeft in totaal 45 medewerkers en is een functionele organisatie. In Tabel 2 zijn de afdelingen van Social Brothers te zien. De opdracht is uitgevoerd in de rol van backend developer (Figuur 1) voor de backend-afdeling.



Figuur 1. Organogram

Tabel 2. Afdelingen

Afdeling	Beschrijving
Staff en Sales	Houdt de administratie bij en verkoopt projecten aan klanten
Project management	Is verantwoordelijk voor de communicatie tussen de klant en de developers en zorgt ervoor dat het project goed verloopt
Backend	Ontwikkelt API's en koppelingen met backend systemen
React	Ontwikkelt web- en mobiele applicaties in ReactJS en React native
Wordpress	Ontwikkelt websites in Wordpress
Marketing	Helpt klanten met marketing campagnes
Design	Ontwerpt interfaces voor applicaties
Customer success	Helpt klanten met vragen en/ problemen

1.2. Stakeholders

In Tabel 3 staan de stakeholders en wordt beschreven wat er van hen verwacht wordt.

Tabel 3. Stakeholders

Stakeholder	**Beschrijving**
Afstudeerder (Danny Cao)	Danny Cao heeft aan de afstudeeropdracht gewerkt. De afstudeerder levert de volgende producten op: scriptie, PoC, zelfreflectie, handleiding en technische documentatie.
Social Brothers (Hugo Kamps, Jonne Kampen, Pim Reijersen)	Social Brothers heeft de opdracht aan de afstudeerder aangeboden. Hugo Kamps is als technische begeleider aangesteld,. Jonne als procesbegeleider en Pim is aangesteld als plaatsvervangend technische begeleider.
Hogeschool Utrecht (Ana Goessens, Philippine Waisvisz)	De Hogeschool zal tijdens en na de uitvoering van dit project de competenties van de stagiair beoordelen. Ana Goessens is door de Hogeschool Utrecht aangesteld als eerste examinator en Philippine Waisvisz als tweede examinator.

2 De opdracht

2.1. Probleemstelling

Social Brothers maakt backendsystemen voor klanten. Om deze software te deployen, moeten steeds dezelfde handmatige acties uitgevoerd worden. Het gaat om de volgende handmatige acties:

1. een VPS aanmaken
2. de VPS configureren
3. het backend systeem op de VPS installeren

De huidige deploymentoplossing heeft de volgende problemen:

- het deployen is foutgevoelig, omdat alles met de hand gebeurt;
- het deployen is inconsistent en moeilijk onderhoudbaar, omdat er geen standaard software is die op een bepaalde manier wordt geïnstalleerd;
- het deployen kost veel tijd en daarom ook geld;
- het deployen is onveilig, omdat er geen standaard software is die wordt geïnstalleerd en daarom ook verouderde software geïnstalleerd kan worden.

Social Brothers wil dat tijdens het deploymentproces automatisch veilige inloggegevens worden gegenereerd en opgeslagen op een centrale plek waar alleen bevoegde mensen bij kunnen.

Social Brothers gebruikt momenteel zelfgemaakte scripts om software te deployen. De scripts automatiseren een aantal stappen van het deploymentproces, maar het bedrijf is daar niet tevreden over, omdat de scripts foutgevoelig is en onnodige handmatige input nodig is.

2.2. Afbakening

2.2.1. Projectgrenzen

Het onderzoek wordt uitgevoerd op de backend-afdeling van Social Brothers. Dat betekent dat alleen rekening wordt gehouden met backend software die op de afdeling wordt gemaakt. Specifiek ligt de focus op .NET 6 API's, omdat de scope van het onderzoek anders te breed wordt.

Aan het eind van het project worden de volgende producten aan het bedrijf geleverd:

1. een proof of concept dat een .NET 6 API automatisch kan deployen, inloggegevens voor backendsystemen kan genereren en veilig is op te slaan.
1. Een handleiding voor de proof of concept;
2. Technische documentatie voor de proof of concept.

De proof of concept dient als ondersteuning van het onderzoek. De afstudeerder is niet verantwoordelijk voor het aanpassen van het huidige deploymentproces. Social Brothers kan de proof of concept gebruiken als voorbeeld van een toekomstige manier van software deployen van .NET 6 API's.

3 Doelstelling

Het doel van dit onderzoek is achterhalen hoe Social Brothers de huidige manier van .NET 6 API deployment kan verbeteren en het proces consistent, veilig, minder foutgevoelig en tijdbesparend kan maken.

De uitkomsten van het onderzoek worden verwerkt in een proof of concept. De proof of concept bewijst dat het onderzoek haalbaar is in de praktijk. Met de proof of concept kan een .NET 6 API automatisch gedeployed worden en worden automatisch inloggegevens gegenereerd voor een database en de server. De gegenereerde inloggegevens worden daarna op een veilige plek opgeslagen.

4 Onderzoeksvragen

Om de doelstelling te bereiken, is de volgende hoofdvraag geformuleerd:

Wat is de beste manier om .NET 6 API's te deployen, zodat automatisch veilige inloggegevens worden gegenereerd voor de backendsystemen die de .NET 6 API's nodig heeft om te functioneren en waarbij deze inloggegevens zo veilig mogelijk worden opgeslagen, zodat het deployment van .NET 6 API's zo consistent, veilig en minder foutgevoeliger wordt?

De hoofdvraag wordt beantwoord met behulp van de volgende deelvragen:

1. Welke manieren van deployment voor .NET 6 API's heeft de backend-afdeling al geprobeerd?
2. Wat is de gewenste situatie om .NET 6 API's te deployen op de backend-afdeling?
3. Hoe zijn zo veilig mogelijk automatisch inloggegevens te genereren en op te slaan voor backendsystemen?
4. Welke manieren zijn er om .NET 6 API's automatisch te deployen?
5. Welke manier om .NET 6 API's automatisch te deployen is het meest geschikt voor Social Brothers?

5 Onderzoeksresultaten

5.1. Deelvraag 1

Welke manieren van deployment voor .NET 6 API's heeft de backend afdeling al geprobeerd?

5.1.1. Methoden

Deze deelvraag is in twee stappen beantwoord. Eerst heeft de afstudeerder in het huidige systeem bekeken hoe het huidige deploymentproces eruitziet. Dit is gedaan op basis van deskresearch. De afstudeerder heeft de backend-medewerkers gevraagd naar handleidingen, voorbeelden en bronnen over hoe zij hun .NET API's deployen.

Daarna zijn de backend-medewerkers geïnterviewd. Dit is gedaan op basis van een semigestructureerd interview, omdat dan doorgevraagd mag worden, waardoor meestal meer gedetailleerde informatie naar voren komt.

Het doel van de interviews was om de gegevens die tijdens de deskresearch verzameld zijn te verifiëren en om te begrijpen wat in het verleden al is geprobeerd. Tijdens de interviews (Bijlage A en Bijlage B) zijn ook vragen gesteld over wat wel en niet goed is aan het deploymentproces en hoe het verbeterd kan worden.

5.1.2. Resultaat

Broncode beheren

Softwareprojecten moeten goed bereikbaar zijn en opgeslagen worden om ze te kunnen deployen. Social Brothers gebruikt Bitbucket om alle backend software op een veilige manier op te slaan.

Bitbucket is een Git-based hosting service voor professionele teams. Het is een centrale plek om Git-repositories te beheren en samen te werken aan broncode (Bitbucket: What Is Bitbucket? | Evaluator Resources, z.d.).

Uit de interviews met backend teamlead, Pim en backend developer, Hugo op 21 december 2021 blijkt dat Social Brothers sinds het .NET API's creëert, al gebruikmaakt van Bitbucket. Dat komt omdat Social Brothers vroeger andere producten gebruikte van Atlassian, de eigenaar van Bitbucket.

Social Brothers heeft een methode om op een overzichtelijke manier software te bouwen. Op Bitbucket is het mogelijk om voor elke user story een branch aan te maken (Figuur 2). Tabel 4 toont de methode die Social Brothers gebruikt.

Figuur 2. Gitflow (git flow, z.d.)

Tabel 4. Branches

Branch	Beschrijving
Master	Dit is de broncode die uiteindelijk op de staging of productie server wordt gedeployed.
Develop	Dit is de broncode die op de test server wordt gedeployed.
Feature	Dit is de branch die wordt aangemaakt voor een nieuwe user story waarbij het doel het toevoegen van een nieuwe functionaliteit is. Voor een project kunnen meerdere feature branches bestaan.
Bugfix	Dit is de branch die wordt aangemaakt voor een nieuwe user story, waarbij het doel het corrigeren of verbeteren van een functionaliteit is. Voor een project kunnen meerdere bugfix branches bestaan.

Deploymentprocessen in het verleden

Microsoft Azure

In september 2021 heeft Social Brothers geëxperimenteerd met Microsoft Azure om .NET API's te deployen. Bij Azure is het mogelijk om snel en simpel met een druk op de knop een project te deployen. In deelvraag 4 wordt hier dieper op ingegaan.

Figuur 3. Microsoft Azure

Social Brothers kwam erachter dat Microsoft Azure voor 99% van de gedeployde projecten niet geschikt is, aangezien het een managed omgeving is. Dat wil zeggen dat Social Brothers verantwoordelijk is voor de veiligheid, het maken van back-ups, en onderhoud van de server waar het gedeployde project op staat. Gebruikmaken van een managed omgeving vindt Social Brothers interessant, maar het vindt de prijs te hoog in verhouding tot wat een dergelijke omgeving biedt.

DigitalOcean

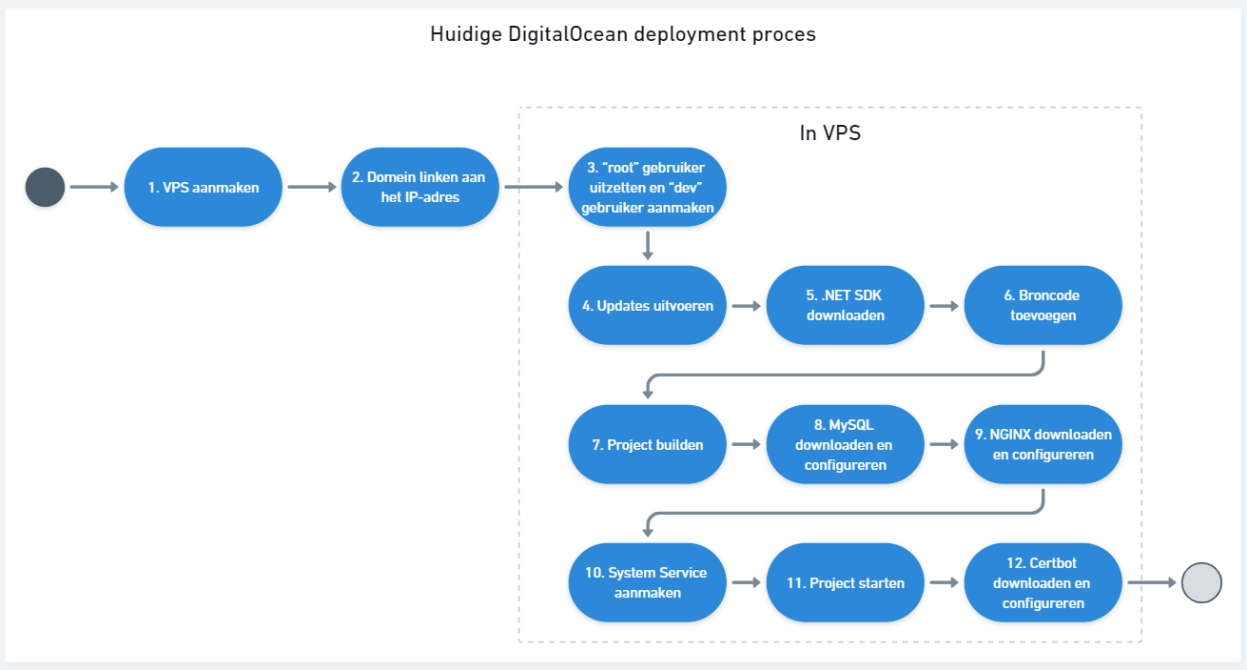
Social Brothers kwam al snel met een andere oplossing, namelijk zelf een VPS aanmaken en met de hand het project op de VPS deployen. Als operating system van de VPS heeft het gekozen voor Ubuntu, omdat dit het meest gebruikte operating system is om software op te hosten.

Als VPS hosting platform is gekozen voor DigitalOcean, omdat hierop goedkoop een server is aan te maken.

Figuur 4. Digitalocean

Social Brothers maakt sinds 2018 gebruik van DigitalOcean om andere soorten software te hosten, namelijk PHP-projecten.

De manier van deployen van de .NET API op een DigitalOcean server wordt beschreven in Figuur 5. Elke stap is een handmatige actie.



Figuur 5. Oude deployment

Het proces bestaat uit twaalf stappen:

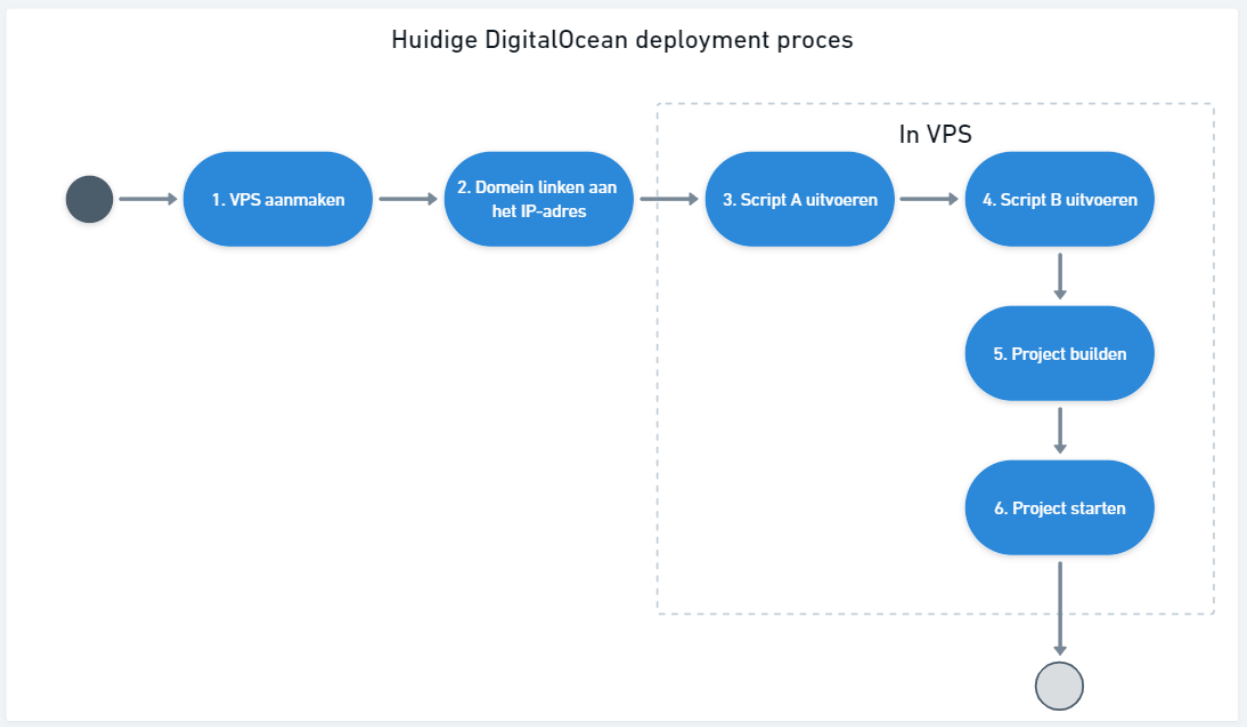
1. Aanmaken van een VPS op DigitalOcean waarop de software gedeployed kan worden
2. De domeinnaam aan het IP-adres van de VPS linken.
3. Voor de veiligheid de rootgebruiker uitzetten en een nieuwe gebruiker genaamd dev aanmaken. Ubuntu maakt bij het aanmaken van een server automatisch een rootgebruiker aan. De gebruiker wordt uitgezet, omdat hij alle rechten heeft om operaties uit te voeren en dat is gevaarlijk om actief te laten.
4. Updates uitgevoerd op de VPS server om de veiligheid van de server te bewaken.
5. Een .NET SDK, oftewel een programma dat een .NET API nodig heeft om te draaien, downloaden.
6. De broncode toevoegen.
7. De broncode bouwen.
8. De MYSQL database downloaden en configureren.
9. De NGINX server installeren en configureren. Dit is nodig om toegang van buiten de server naar het project te leiden.
10. System service aanmaken, zodat de gebouwde versie van de broncode gestart kan worden.
11. Het project starten.
12. Certbot downloaden en configureren, zodat het domein HTTPS-toegang heeft.

De backend-afdeling van Social Brothers heeft vroeger op deze manier de .NET API's gedeployed, maar kwam erachter dat deze manier van software deployen niet helemaal voldoet aan hun eisen. De reden is dat twaalf handmatige stappen uitgevoerd moeten worden om een

project te deployen. Het is moeilijk om iedereen op de afdeling ertoe aan te zetten dezelfde stappen uit te voeren. Dat betekent dat het deploymentproces per persoon kan verschillen en dus niet consistent is.

Huidige deploymentproces

Het huidige deploymentproces van de backend-afdeling lijkt op het oude DigitalOcean deploymentproces beschreven in Figuur 5. Het verschil is dat twee batchscripts zijn gemaakt die een aantal stappen automatiseren (Figuur 6). In Tabel 5 worden de scripts toegelicht. Script A wordt door de rootgebruiker uitgevoerd en Script B door de nieuw aangemaakte gebruiker, genaamd dev. Met de batchscripts wordt het deploymentproces consistenter, omdat de scripts niet per uitvoering afwijken.



Figuur 6. Huidige deployment

Tabel 5. huidige deployment

Scripts	Gebruiker uitvoering	Toelichting
Script A	root	1. Updates uitvoeren 2. Rootgebruiker uitzetten en dev-gebruiker aanmaken
Script B	dev	1. .Net SDK wordt gedownload 2. Broncode wordt toegevoegd 3. MySQL wordt gedownload en geconfigureerd 4. NGINX wordt gedownload en geconfigureerd 5. Certbot wordt gedownload en geconfigureerd

Social Brothers vindt het huidige deploymentproces beter dan het vorige, maar is er nog niet helemaal tevreden over. Onderstaand is te lezen wat de medewerkers van de backend-afdeling over het huidige deploymentproces zeggen (Bijlage A en Bijlage B):

“De scripts werken nooit voor mij, alleen voor Hugo.”

- **Backend team-lead, Pim (21 december 2021)**

“De scripts hebben nog steeds handmatige acties nodig, het uitvoeren daarvan, af en toe op enter drukken en wachtwoorden invoeren, is erg irritant.”

- **Backend developer, Hugo (21 december 2021)**

“Als de script faalt, dan weet ik niet wat er fout ging en moet ik de server verwijderen en een nieuwe aanmaken.”

- **Backend team-lead, Pim (21 december 2021)**

“Het is moeilijk om het gebruik van de scripts te forceren. Hoe weet ik dat de scripts worden gebruikt als ik bijvoorbeeld volgende week op vakantie ben.”

- **Backend developer, Hugo (21 december 2021)**

“De handmatige acties tijdens het uitvoeren van de scripts zijn erg irritant, dat zou ook geautomatiseerd kunnen worden.”

- **Backend team-lead, Pim (21 december 2021)**

5.1.3. Conclusie

Social Brothers heeft in het verleden met drie deploymentprocessen gewerkt.

Microsoft Azure
In september 2021 heeft Social Brothers geëxperimenteerd met Microsoft Azure, maar het bedrijf kwam er al snel achter dat het niet voldeed aan de eisen, omdat het te duur was. Momenteel host Social Brothers alleen nog software op Azure als het een managed omgeving nodig heeft en het binnen het budget past.

Oude deploymentproces DigitalOcean
Social Brothers kwam al snel met een andere oplossing, namelijk DigitalOcean gebruiken om zelf een VPS aan te maken en handmatig te deployen (Figuur 5). De oplossing was goedkoper dan Microsoft Azure, maar het bedrijf was niet tevreden over de vele handmatige acties die het moest uitvoeren voor deployment.

Huidige deploymentproces DigitalOcean
Om het aantal handmatige acties te reduceren, heeft de backend-afdeling een aantal batchscripts gemaakt. De batchscripts zorgden daadwerkelijk voor minder handmatige acties en een hogere consistentie, maar uit de interviews bleek dat de backend-afdeling er nog niet geheel tevreden over was. De voornaamste klachten zijn dat nog steeds onnodige handmatige acties vereist zijn, dat het moeilijk is om het gebruik daarvan te forceren en dat de scripts weleens niet werken.

5.2. Deelvraag 2

Wat is de gewenste situatie om .NET 6 API's te deployen op de backend-afdeling?

5.2.1. Methoden

Deze deelvraag is beantwoord door medewerkers van de backend-afdeling te interviewen. Er is gebruikgemaakt van een semigestructureerd interview, omdat er dan doorgevraagd kan worden. Zo is het makkelijker om de requirements van de gewenste situatie in kaart te brengen. Het doel van de interviews was om te begrijpen wat de gewenste situatie is om een .NET API 6 te deployen en wat daarvoor de requirements zijn.

Na de interviews (Bijlage C en Bijlage D) zijn de resultaten uit deelvraag 1 vergeleken met de resultaten van de interviews om een GAP-analyse te kunnen uitvoeren. Hierdoor was het duidelijk wat er gedaan moest worden om de gewenste situatie te bereiken.

5.2.2. Uitvoering

Requirements

In de interviews voor deelvraag 1 (Bijlage A en Bijlage B) is naar minpunten gevraagd. Het blijkt dat teveel onnodige handmatige acties vereist zijn en dat de uitvoering van de deployment niet consistent is.

Voor deze deelvraag is nogmaals een interview (Bijlage C en Bijlage D) gehouden om de gewenste situatie van .NET 6 API deployment te achterhalen. Daarbij zijn meer gedetailleerde vragen gesteld over de requirements. Op basis van de interviews zijn de volgende requirements opgesteld:

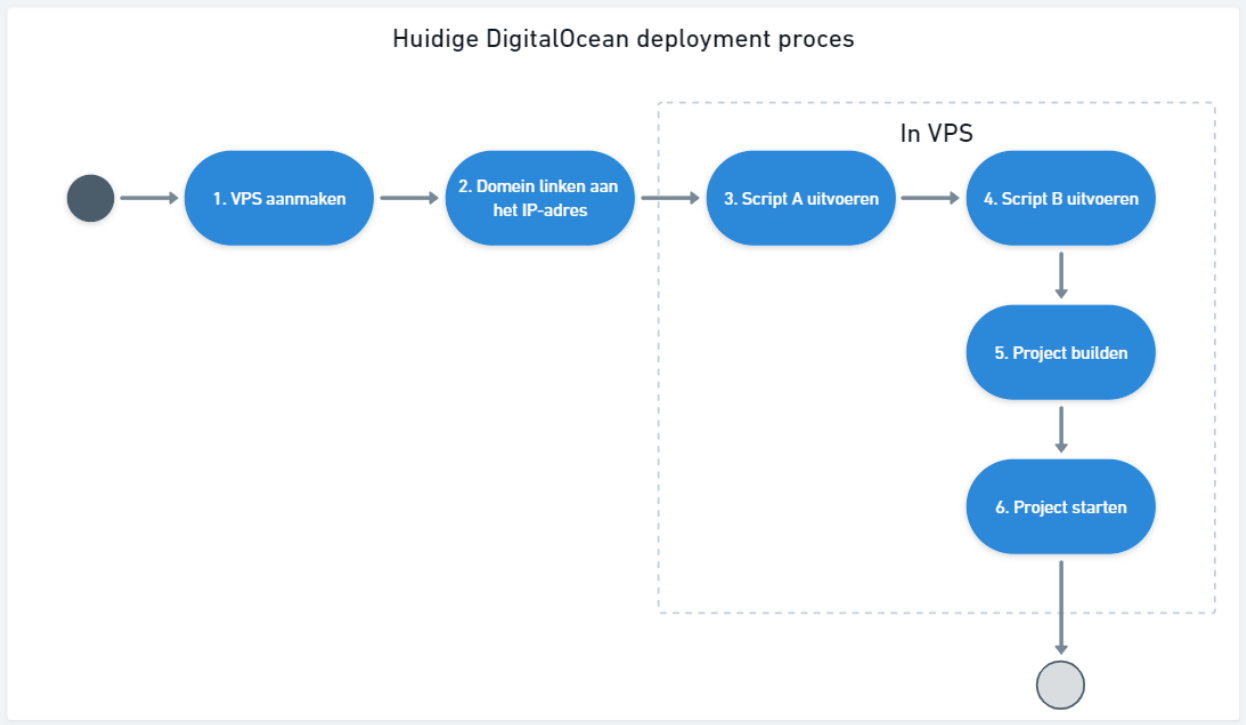
- De onnodige handmatige acties moeten geautomatiseerd worden.
- Het deployment van een .NET 6 API moet voor elke stap gelogd worden.
- Gevoelige gegevens die tijdens het deploymentproces gegenereerd worden, moeten veilig worden opgeslagen.
- Het deployen van een .NET 6 API moet consistent zijn. Voor elke deployment worden dezelfde stappen uitgevoerd.
- Het deploymentproces moet gemakkelijk te herstellen zijn als iets fout gaat.

Tijdens de interviews is ook gevraagd wat goed gevonden wordt in het huidige deploymentproces en wat behouden mag blijven. Daaruit zijn de volgende punten gekomen:

- Een goed overzicht hebben van gedeployde projecten.
- Makkelijk de toegang tot de servers kunnen beheren.
- Toegang kunnen krijgen tot de servers via SSH keys.
- Goede monitoring per deployed project.

Onnodige handmatige acties

Een van de requirements die zijn opgesteld, is dat onnodige handmatige acties geautomatiseerd moeten worden. Figuur 7 toont nogmaals het huidige deploymentproces en Tabel 6 beschrijft welke handmatige acties geautomatiseerd kunnen worden.



Figuur 7. huidige deployment

Tabel 6. Huidige deployment

Stap	Kan automatisch	Beschrijving
1.VPS aanmaken	Ja	Social Brothers gebruikt voor 99% van de projecten hetzelfde type VPS. Als het type VPS verschilt, dan moet het handmatig worden aangepast.
2.Domein linken aan IP	Misschien	Deze stap is misschien te automatiseren. In het huidige deploymentproces kan dat niet, omdat het huidige domeinbeheerder geen API heeft. Als Social Brothers bereid is om over te stappen naar een domeinbeheerder met een API, dan kan dit proces geautomatiseerd worden met een koppeling.
3.Script A uitvoeren	Ja	Deze stap is misschien te automatiseren. In het huidige deploymentproces kan dat niet, omdat het huidige domeinbeheerder geen API heeft. Als Social Brothers bereid is om over te stappen naar een domeinbeheerder met een API, dan kan dit proces

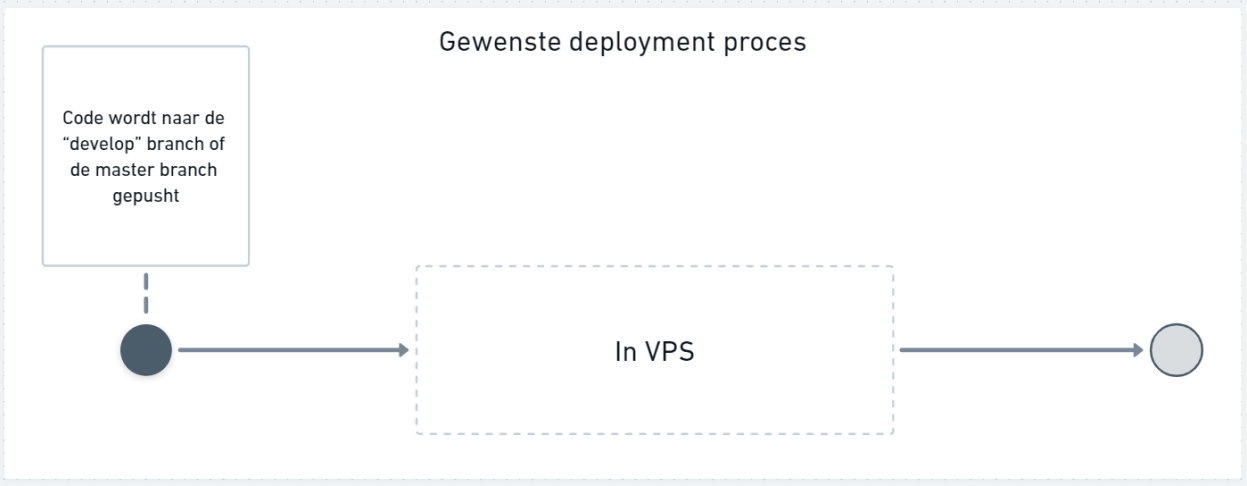
		geautomatiseerd worden met een koppeling.
4.Script B uitvoeren	Ja	Bij het uitvoeren van Script B worden de volgende acties uitgevoerd: 1. .Net SDK wordt gedownload 2. Broncode wordt toegevoegd 3. MySQL wordt gedownload en geconfigureerd 4. NGINX wordt gedownload en geconfigureerd 5. Certbot wordt gedownload en geconfigureerd 6. System service wordt aangemaakt Alle stappen zijn automatiseerbaar. Bij stap 3 moeten inloggegevens gegenereerd en opgeslagen worden. In deelvraag 3 wordt daar verder op ingegaan.
5.Project builden	Ja	De code staat nu na het uitvoeren van stap 4 op de server. Nu is het een kwestie van een commando uitvoeren.
6.Project starten	Ja	Na het uitvoeren van stap 5 staat op de server een gebouwde versie van het project. Het project is te starten door een commando in te voeren.

Logging

De medewerkers van de backend-afdeling vinden logging van belang. Ze hebben dit toegepast in bijna al hun projecten, omdat het inzichtelijker wordt als bijvoorbeeld iets fout gaat. Daarom hebben ze dit als requirement opgesteld voor het gewenste deploymentproces. Ze willen dat voor elke stap die wordt uitgevoerd te zien is wat er precies gebeurt. Door logging toe te passen, wordt het deploymentproces herstelbaar als er iets fout gaat, omdat dan duidelijk is bij welke stap in het deploymentproces iets fout gaat en waarom.

Gewenste situatie

Nu duidelijk is wat er gedaan moet worden om de gewenste situatie te bereiken, wordt in Figuur 8 het gewenste deploymentproces getoond.



Figuur 8. Gewenste deployment proces

Voordat het deploymentproces begint, wordt op Bitbucket code gepusht naar de develop branch of master branch. Als code wordt gepusht naar de develop branch, dan wordt een nieuwe deployment uitgevoerd voor de develop-omgeving. Als code wordt gepusht naar de master branch, dan wordt een nieuwe deployment uitgevoerd voor de staging-omgeving.

De domeinnaam linken aan het IP-adres staat los van het deploymentproces. Het linken van de domeinnaam kan alsnog na het deploymentproces.

Tabel 7 toont de stappen van het gewenste deploymentproces die automatisch worden uitgevoerd. Daarnaast toont de tabel hoe de log er per stap uit komt te zien.

Tabel 7. Stappen gewenste deploymentproces

Stap	Log	Beschrijving
1	VPS created	VPS wordt aangemaakt
2	Dev user created	rootgebruiker wordt uitgezet en dev-gebruiker wordt aangemaakt, waarbij het wachtwoord automatisch gegenereerd en opgeslagen wordt
3	Updated VPS	Updates voor de VPS worden uitgevoerd
4	Download .NET SDK	.NET SDK wordt gedownload en geïnstalleerd
5	Added source code	Broncode wordt aan de server toegevoegd
6	Build successful	Project wordt gebouwd
7	MySQL ready	MySQL wordt gedownload en geconfigureerd, waarbij de inloggegevens worden gegenereerd en opgeslagen
8	NGINX ready	NGINX wordt gedownload en geconfigureerd
9	Started project	Project wordt gestart
10	Certbot ready	CertBot wordt gedownload en geconfigureerd

5.2.3. Conclusie

Tijdens de interviews is gevraagd naar de requirements van de gewenste situatie. Daaruit kwamen de volgende requirements:

- De onnodige handmatige acties moeten geautomatiseerd worden.
- Het deployment van een .NET 6 API moet voor elke stap gelogd worden.
- Gevoelige gegevens die tijdens het deploymentproces gegenereerd worden, moeten veilig worden opgeslagen.
- Het deployen van een .NET 6 API moet consistent zijn. Voor elke deployment worden dezelfde stappen uitgevoerd.
- Het deploymentproces moet gemakkelijk te herstellen zijn als iets fout gaat.

Tijdens de GAP-analyse is geanalyseerd welke stappen uitgevoerd moeten worden om de gewenste situatie te bereiken. Daaruit kwam het resultaat getoond in Figuur 8. Alle stappen zijn te automatiseren behalve het linken van de domeinnaam.

5.3. Deelvraag 3

Hoe zijn zo veilig mogelijk automatisch inloggegevens voor backendsystemen te genereren en op te slaan?

5.3.1. Methoden

Deze deelvraag is beantwoord door deskresearch uit te voeren in de vorm van een literatuuronderzoek. Nu duidelijk is wat de gewenste situatie is en voor welke backendsystemen automatisch inloggegevens gegenereerd moeten worden, kan onderzocht worden hoe dat zo veilig mogelijk te doen is. Voor de backendsystemen is onderzocht of er al bekende manieren zijn om dat zo veilig mogelijk te doen. Daarna zijn oplossingen bekeken die voldeden aan de eisen van Social Brothers. Vervolgens is onderzocht hoe de geautomatiseerde inloggegevens zo veilig mogelijk zijn op te slaan.

5.3.2. Uitvoering

Backendsystemen

Tijdens de interviews voor deelvraag 2 (Bijlage C en Bijlage D) is gevraagd naar backendsystemen waarvoor inloggegevens gegenereerd worden tijdens het deploymentproces. Uit de interviews kwamen twee backendsystemen naar voren: MySQL en VPS.

MySQL is een databasemanagementsysteem. Dit systeem is nodig om gegevens op te slaan voor een softwareproject. De VPS waar het project op gaat draaien, heeft in de huidige flow inloggegevens nodig. Met de inloggegevens kunnen developers toegang krijgen tot de VPS en de benodigde software installeren.

Veiligheid

Het genereren van inloggegevens voor de MySQL database en VPS moet zo veilig mogelijk gebeuren. Wachtwoorden vormen de eerste verdedigingslinie tegen ongeautoriseerde toegang tot de VPS. Hoe sterker het wachtwoord is hoe beter de VPS beschermd is tegen hackers en schadelijke software. Voor het genereren van de inloggegevens moeten daarom voldaan worden aan de volgende eisen:

1. het genereren van wachtwoorden moet niet zichtbaar zijn voor onbevoegde mensen;
2. het genereren van wachtwoorden moet niet van tevoren te bepalen zijn;
3. het gegenereerde wachtwoord moet aan een veilige conventie voldoen.

1. **Zichtbaarheid**

Tijdens een nieuwe deployment wordt een nieuwe VPS aangemaakt. Wachtwoorden moeten gegenereerd worden tijdens de deployment binnen de VPS. Als het wachtwoord binnen de nieuwe VPS wordt gegenereerd, dan kunnen onbevoegde mensen niet bij de server, omdat niemand nog toegang heeft tot de VPS.

2. **Niet van te voren te bepalen**

Het wachtwoord mag van tevoren niet te bepalen zijn. Het wachtwoord mag dus niet buiten het deploymentproces ingevuld worden in een bestand en tijdens de deployment ingevoerd worden. Het wachtwoord moet verder per deployment verschillen.

3. **Wachtwoordconventie**

Volgens de Microsoft-documentatie over wachtwoordconventie voldoet een veilig wachtwoord aan vier eisen:

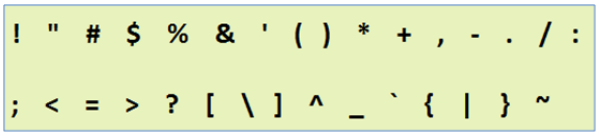
1. het wachtwoord moet minimaal uit 8 karakters bestaan;
2. het wachtwoord heeft hoofdletters en kleine letters;
3. het wachtwoord heeft minimaal een getal;
4. het wachtwoord heeft ook speciale karakters (Figuur 9).

Het wachtwoord dat gegenereerd wordt, moet voldoen aan de conventie dat hoe langer het wachtwoord is, hoe veiliger het is (Windows Security, 2021).

Momenteel gebruikt Social Brothers een password manager om de wachtwoorden te genereren. Social Brothers houdt zich aan de wachtwoordconventie, maar hanteert als minimumlengte zestien karakters. In Tabel 8 staat waar een wachtwoord aan moet voldoen.

Tabel 8. Wachtwoordeisen

1. Bestaat minimaal uit 16 karakters. 2. Bestaat uit minimaal 1 hoofdletter of 1 kleine letter 3. Bestaat uit minimaal 1 nummer 4. Bestaat uit minimaal een speciale karakter (Figuur 10)



Figuur 9. Speciale karakters

Genereren

Het genereren van de wachtwoorden moet gebeuren tijdens het deploymentproces, om zo de veiligheid van het deploymentproces te bewaken. Daarvoor is een programma nodig dat op een veilige manier wachtwoorden kan genereren. Er zijn verschillende programma's die dat kunnen

doen. Hieronder worden de verschillende manieren toegelicht en wordt beschreven welke manier de beste is.

1. **Pwgen**

Pwgen is een wachtwoordgenerator gemaakt en onderhouden door Theodore Ts'o, een Amerikaanse softwareprogrammeur. Pwgen is een populaire manier om wachtwoorden te genereren op Linux-distributies, omdat het eenvoudig te gebruiken is. Het is bijvoorbeeld mogelijk om voor het genereren van een wachtwoord aan te geven waar het wachtwoord aan moet voldoen. Met het commando hieronder is het mogelijk een wachtwoord te genereren dat voldoet aan de conventie:

```
pwgen -c -n -s -y -1 16
```

(Pwgen(1): Make Pronounceable Passwords - Linux Man Page, z.d.)

2. **GPG**

GPG (GNU Privacy Guard) is een programma dat bestanden op Windows, MacOs en Linux kan beveiligen met behulp van de modernste cryptografie. Met GPG is het mogelijk om veel maar dan alleen maar wachtwoorden genereren. Maar het is niet mogelijk om met GPG een wachtwoord te genereren dat voldoet aan de conventie, omdat er geen optie is om aan te geven waar aan het wachtwoord aan moet voldoen. De enigste optie die een gebruiker kan meegeven is hoe lang het wachtwoord moet zijn. (Henry-Stocker, 2018)

3. **Urandom**

In Unix-achtige besturingssystemen is Urandom een speciaal bestand dat dient als nummergenerator. Met Urandom is het mogelijk om een willekeurige tekenreeks te genereren die als wachtwoord kan worden gebruikt.

4. **xkcdpass**

Xkcdpass is een flexibele en scriptbare wachtwoordgenerator die sterke wachtwoordzinnen genereert, geïnspireerd door XKCD 936. Met xkcdpass is het mogelijk om willekeurige bestaande Engelse woorden te genereren. Het doel van xkcdpass is om veilige wachtwoorden te genereren die voor een mens gemakkelijk te onthouden zijn. Het is niet mogelijk om wachtwoorden te genereren die voldoen aan de conventie (Xkcdpass, 2021).

In Tabel 9 is te zien dat twee wachtwoordgenerators voldoen aan de eisen: pwgen en Urandom. Deze twee programma's kunnen allebei willekeurige wachtwoorden genereren die voldoen aan de wachtwoordconventie. Het enige nadeel van pwgen is dat het niet standaard geïnstalleerd is op Linux. Urandom is een standaardprogramma dat beschikbaar is op Linux en niet geïnstalleerd hoeft te worden.

Door Urandom en niet pwgen te gebruiken om wachtwoorden te genereren, is tijd te besparen tijdens het deploymentproces. Daarnaast is Urandom veiliger om te gebruiken, omdat geen software geïnstalleerd hoeft te worden van derde partijen en een standaardprogramma van het operating system gebruikt kan worden.

Met het gegenereerde wachtwoord is het mogelijk om een nieuwe gebruiker aan te maken voor de backendsystemen MySQL en VPS.

Tabel 9. Wachtwoord-generators

	pwgen	***GPG***	***urandom***	***xkcdpass***
Kan wachtwoorden genereren die voldoen aan de conventie	*Ja*	*Nee*	*Ja*	*Nee*
Gebruiksvriendelijk	*Ja*	*Nee*	*Nee*	*Ja*
Populair	*Ja*	*Ja*	*Ja*	*Ja*
Standaard op Linux	*Nee*	*Nee*	*Ja*	*Nee*

Opslaan

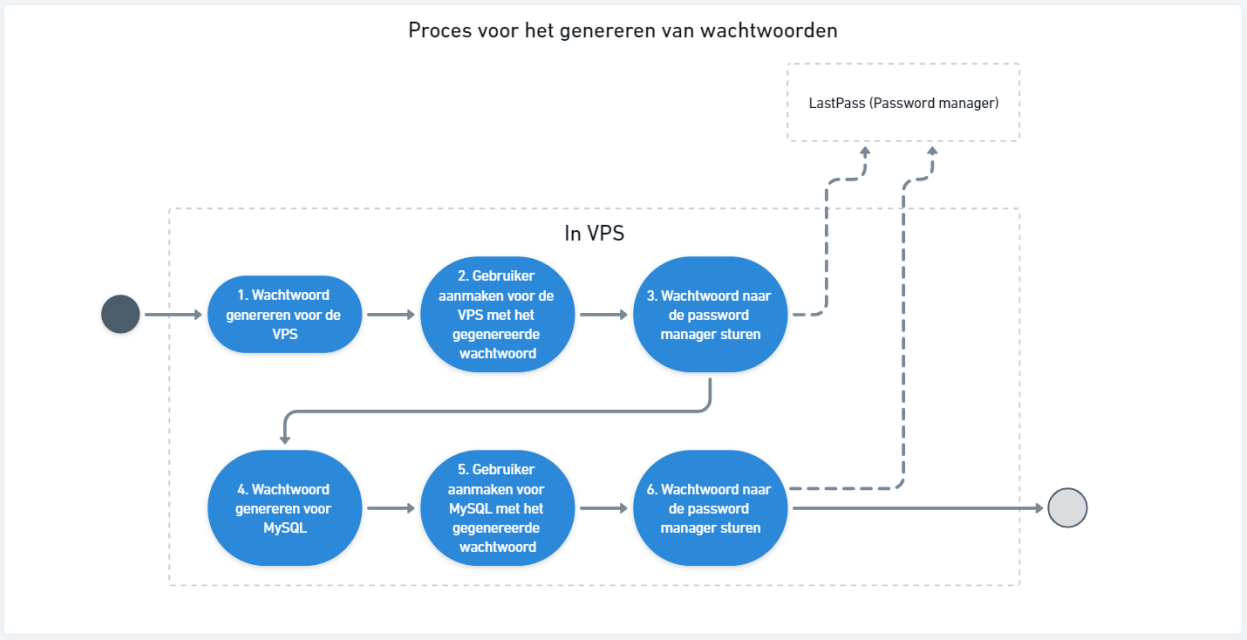
Na het genereren van de wachtwoorden en het aanmaken van de gebruikers voor MySQL en de VPS moeten de inloggegevens ergens opgeslagen worden. Om dat zo veilig mogelijk te doen, moet het opslaan van de inloggegevens voldoen aan de volgende eisen:

- de opgeslagen inloggegevens zijn alleen toegankelijk voor bevoegde mensen;
- de inloggegevens moeten tijdens het versturen naar een veilige plek versleuteld zijn.

Social Brothers gebruikt momenteel LastPass, een password manager om inloggegevens voor verschillende servers en websites op te slaan. Social Brothers gebruikt deze password manager ook in het huidige deploymentproces om wachtwoorden op te slaan. Lastpass heeft een API met het commando “pushsitestousers”, waarbij het mogelijk is om op een veilige manier inloggegevens naar de servers te sturen. De inloggegevens worden dan versleuteld, zodat alleen bevoegde mensen ze kunnen zien.

Deze manier van inloggegevens opslaan voldoet aan de eisen, omdat in Lastpass beheerd kan worden wie toegang heeft tot opgeslagen inloggegevens. Zo is het mogelijk om alleen bevoegde mensen toegang te geven tot de gegenereerde inloggegevens van de deployment.

Door gebruik te maken van de API van Lastpass zijn de inloggegevens die verstuurd worden naar Lastpass versleuteld en worden ze ook versleuteld opgeslagen. Figuur 10 toont hoe het proces eruit komt te zien.



Figuur 10. wachtwoord genereren proces

5.3.3. Conclusie

Het genereren van wachtwoorden moet zo veilig mogelijk gedaan worden omdat wachtwoorden de eerste verdedigingslinie tegen ongeautoriseerde toegang tot de server vormen. Hoe sterker het wachtwoord is hoe beter de server beschermd is tegen hackers en schadelijke software.

Urandom is een programma waarmee de gebruiker veilige wachtwoorden kan genereren voor de backendsystemen MySQL en VPS. Er is gekozen voor Urandom, omdat de wachtwoorden die gegenereerd worden met Urandom voldoen aan de volgende eisen:

- bestaan uit minimaal 16 karakters;
- bestaan uit minimaal 1 hoofdletter of 1 kleine letter;
- bestaat uit minimaal 1 nummer;
- bestaat uit minimaal een speciaal karakter (Figuur 9).

Urandom is een standaardprogramma dat beschikbaar is op Linux en hoeft niet geïnstalleerd te worden. Dit scheelt tijd.

Na het genereren van de inloggegevens en het aanmaken van de gebruikers moeten de inloggegevens veilig opgeslagen worden. Om dat zo veilig mogelijk te doen, moeten ze versleuteld opgeslagen worden. Daarnaast moeten alleen bevoegde mensen toegang kunnen krijgen tot de opgeslagen wachtwoorden.

Social Brothers maakt gebruik van een password manager genaamd Lastpass. Lastpass heeft een API waarmee de gebruiker op een veilige manier versleuteld de wachtwoorden kan versturen en opslaan. Alle medewerkers van Social Brothers hebben een account bij Lastpass

dat gekoppeld is aan hun e-mail. Hiermee kan gemakkelijk geconfigureerd worden wie wel of niet toegang heeft tot de inloggegevens.

5.4. Deelvraag 4

Welke manieren zijn er om .NET 6 API's automatisch te deployen?

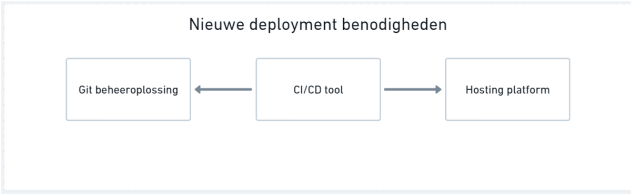
5.4.1. Methoden

Deze deelvraag wordt beantwoord door deskresearch in de vorm van literatuuronderzoek. Voor deze deelvraag worden verschillende manieren geïnventariseerd om een .NET 6 API automatisch te deployen door online naar bestaande oplossingen te zoeken. Daarna worden de verschillende oplossingen met elkaar vergeleken.

5.4.2. Uitvoering

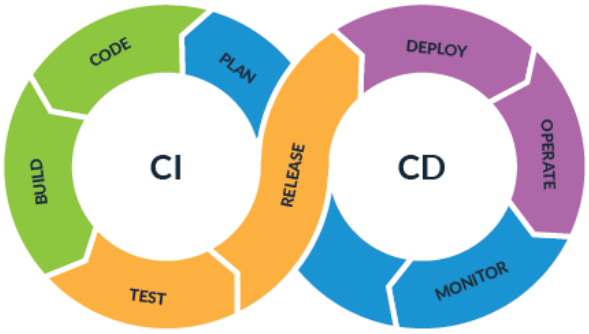
Tools voor het automatisch deployen

Om een deployment uit te voeren, is een stuk software nodig dat automatisch wijzigingen van de broncode bekijkt. Dit wordt ook wel een CI/CD tool genoemd. Met zo'n tool is het mogelijk om een .NET 6 API te deployen wanneer code wordt gewijzigd op een Git-beheeroplossing.



Figuur 11. Nieuwe deployment benodigdheden

CI/CD staat voor Continuous Integration en Continuous Delivery. Continuous Integration betekent dat nieuwe broncode aan oude broncode wordt toegevoegd en getest om zo tot een nieuwe versie van een applicatie te komen.



Figuur 12. CI/CD

Volgens Belmont (2018) is Continuous Delivery de mogelijkheid om allerlei soorten wijzigingen, inclusief nieuwe functies, configuratiewijzigingen, bug fixes en experimenten, veilig en snel op een duurzame manier in productie te nemen of in handen van gebruikers te krijgen.

Het doel is het deployen van software, of het nu gaat om een grootschalig gedistribueerd systeem, een complexe productieomgeving, een ingebed systeem of een app - oftewel, het gaat hier om voorspelbare, routinematige zaken doe op aanvraag worden uitgevoerd.

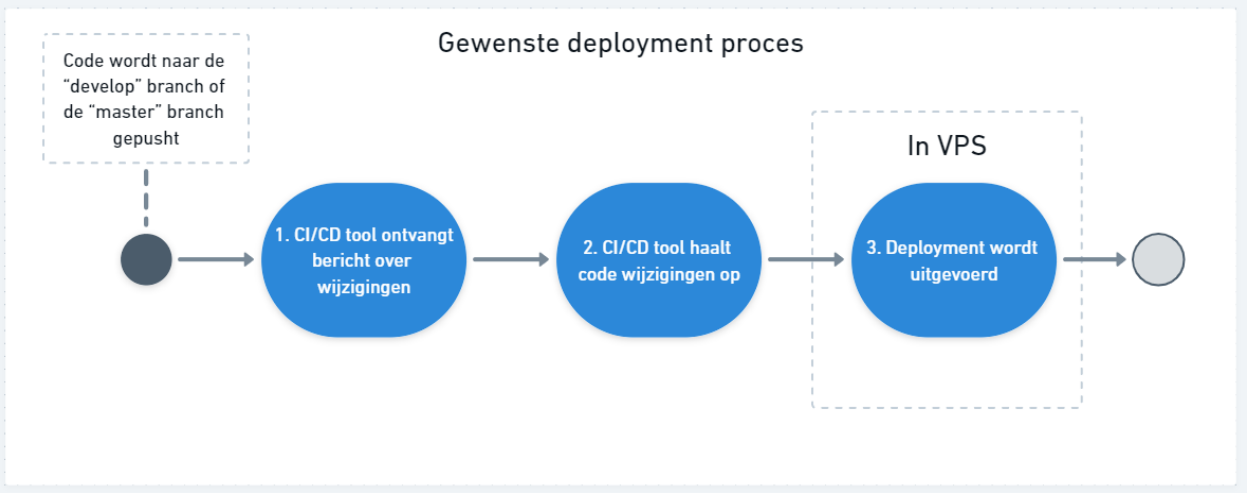
Het implementeren van Continuous Delivery helpt de volgende pluspunten te bereiken:

Tabel 10. CI/CD

Pluspunt	Beschrijving
Sneller op de markt zijn	Door CI/CD toe te passen kunnen developers snel nieuwe functionaliteiten/ wijzigingen op de markt zetten.
Hogere kwaliteit	Er kunnen automatische testen uitgevoerd worden tijdens de deployment en de deployment stoppen als bepaalde testen niet slagen. Zo wordt er voor gezorgd dat alleen code met hoge kwaliteit gedeployed kan worden.
Lagere kosten	Elk succesvol softwareproduct of -service zal in de loop van zijn levensduur aanzienlijk evolueren. Door te investeren in build-, test-, implementatie- en omgeving automatisering, worden de kosten voor het maken en leveren van incrementele wijzigingen aan software verlaagd.
Betere producten	Door Continues delivery toe te passen kan er snel nieuwe functionaliteiten live gezet worden. Dit betekent dat we snel feedback kunnen krijgen van gebruikers gedurende de hele levenscyclus van de levering op basis van werkende software.

> Elk succesvol softwareproduct of -service zal in de loop van zijn levensduur aanzienlijk evolueren. Door te investeren in build-, test-, implementatie- en omgeving automatisering, verlagen we de kosten van het maken en leveren van incrementele wijzigingen aan software aanzienlijk door veel van de vaste kosten die gepaard gaan met het deploymentproces te elimineren. (What Is Continuous Delivery? - Continuous Delivery, z.d.)

Een .NET 6 API moet ergens naartoe gedeployed worden, zodat deze online beschikbaar is. Daarvoor is een hosting platform nodig. Om een .NET 6 API te deployen, zijn een CI/CD tool en hosting platform nodig (Figuur 13). Figuur 14 toont het deploymentproces.



Figuur 14. Gewenste deployment proces

Er wordt niet gekeken naar Git-beheeroplossingen, omdat Social Brothers al tevreden is met de huidige Git-beheeroplossingen. De oplossingen die momenteel op de markt zijn, verschillen niet veel qua werking. Ze maken namelijk allemaal gebruik van Git en hebben bijna allemaal dezelfde features (Van Gumster Feed, 2018). Het enige wat relevant is aan een Git-beheeroplossing in het deploymentproces, is of het CI/CD ondersteunt. Bitbucket doet dit wel.

Hieronder wordt beschreven welke CI/CD tools en hosting platforms beschikbaar zijn.

CI/CD Tools

De CI/CD tools en criteria's zijn gekozen gebaseerd op het artikel van (Nguyen, 2021) over de beste CI/CD tools.

1. **Jenkins**

Jenkins is een open source-automatiseringsserver. Het helpt bij het automatiseren van de onderdelen van softwareontwikkeling met betrekking tot bouwen, testen en deployen, waardoor Continuous Integration en Continuous Delivery worden vergemakkelijkt (Jenkins, z.d.),

Jenkins is momenteel de populairste automatiseringsserver. Dat komt mede dankzij de vele plugins die Jenkins heeft met andere systemen. Jenkins heeft momenteel meer dan 1800 plugins. Met de plugins kan Jenkins geïntegreerd worden met andere systemen. Zo kan bijvoorbeeld per deployment een bericht gestuurd worden naar een Slack-kanaal met de Slack-plugin.

Figuur 15. Jenkins

Een andere optie is dat e-mails verstuurd worden naar bepaalde mensen als een deployment fout gaat. Dat maakt het systeem configureerbaar en uitbreidbaar. Ook is het systeemplatform onafhankelijk. Dat betekent dat Jenkins geïnstalleerd kan worden op de grootste besturingssystemen MacOS, Windows en Linux. Jenkins is open source en volledig gratis te gebruiken. Het nadeel is wel dat Jenkins zelf gehost moet worden.

2. **CircleCI**

CircleCI is een Continuous Integration en Continuous Delivery platform dat een ontwikkelteam helpt met het deployen van software. CircleCI is een service die wordt gehost in de cloud of de gebruiker kan ervoor kiezen om zelf te hosten. Door gebruik te maken van de cloud hoeft de initiële opzet niet lang te duren en is de gebruiker niet zelf verantwoordelijk voor het onderhoud en het beschikbaar houden van de service (Belmont, 2018).

In tegenstelling tot Jenkins is CircleCI niet open source. Voor de cloud-versie en eigen hosted-versie moet betaald worden.

Figuur 16. CircleCI

Het voordeel van het gebruiken van CircleCI is dat het eenvoudig en snel is om het systeem op te stellen in vergelijking tot Jenkins. Het nadeel is dat het niet beschikt over plugins en dus niet net zo configureerbaar en uitbreidbaar is als Jenkins. Ook moet betaald worden voor het gebruiksgemak van CircleCI.

3. Bitbucket Pipelines

Bitbucket Pipelines is een geïntegreerde CI/CD service in Bitbucket. Het geeft de mogelijkheid om code in Bitbucket automatisch te bouwen, testen en deployen. Het voordeel van het gebruiken van Bitbucket Pipelines is dat de source code en deployment in dezelfde omgeving beschikbaar zijn. Bitbucket Pipelines is gemakkelijk te integreren met Slack, zodat de status van een deployment overzichtelijker wordt (Get Started with Bitbucket Pipelines | Bitbucket Cloud, 2020).

Figuur 17. Bitbucket pipelines

Bitbucket pipelines is net als CircleCI een cloud hosted service. Bitbucket pipelines biedt niet de mogelijkheid om hun software op een eigen server te hosten zoals CircleCI.

Social Brothers maakt al gebruik van Bitbucket, waardoor deze oplossing geschikt is. De kosten van Bitbucket pipelines zitten in de kosten van Bitbucket inbegrepen. Er hoeft dus niet apart voor betaald worden als de gebruiker al een Bitbucket-abonnement heeft.

4. Azure Pipelines

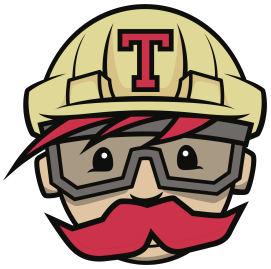
Azure Pipelines is de CI/CD tool van Microsoft Azure. Het is net als Bitbucket Pipelines een cloudoplossing. Het is niet mogelijk om Azure Pipelines zelf te hosten. Wat Azure Pipelines speciaal maakt, is dat het geïntegreerd is in het Microsoft Azure platform. Als de code in Azure wordt beheerd, is het dus niet nodig om naar een andere omgeving te navigeren om de status van de deployment te bekijken. Ook is het eenvoudig om te deployen naar de servers van Azure (What Is Azure Pipelines? - Azure Pipelines, 2021).

Figuur 18. Azure Pipelines

Azure pipelines is gratis te gebruiken voor open source-projecten, maar voor closed source-projecten zijn de kosten $ 40 of $ 15, gebaseerd op de eigenschappen (Use Azure Pipelines - Azure Pipelines, 2021).

5. **Travis CI**

Travis CI is net als CircleCI een cloudoplossing. Het is een van de eerste CI/CD tools op de markt. In het begin was het open source, maar in november 2020 is het gewijzigd naar een closed source-oplossing (Jeff Geerling, z.d.).



Figuur 19. travisCI

Het voordeel van het gebruiken van Travis Ci is dat het eenvoudig te integreren is met hosting platforms als DigitalOcean en Microsoft Azure. Travis CI was vroeger voornamelijk populair voor open source-projecten, omdat het out-of-the-box integreerbaar was met Github en gratis te gebruiken was.

Hosting platforms

Bij de hosting platforms is naar drie soorten platforms gekeken: IaaS (DigitalOcean), all-in-one platform (Azure) en PaaS (Heroku). De soorten hosting platforms en criteria's zijn gebaseerd op de interviews uitgevoerd tijdens deelvraag 2. Hieronder worden de hosting platforms beschreven.

1. **DigitalOcean**

DigitalOcean is een cloud hosting provider die cloud computing-services biedt aan klanten. In januari 2018 behaalde het de titel van het op twee na grootste cloud hosting-bedrijf ter wereld op het gebied van webgerichte computers (DigitalOcean, z.d.).

De backend-afdeling van Social Brothers gebruikt momenteel DigitalOcean om .NET API's en PHP API's te hosten. De belangrijkste redenen daarvoor worden hieronder besproken.

Figuur 20. DigitalOcean

Gebruiksvriendelijkheid
De meeste populaire cloud service providers maken dingen te ingewikkeld door geavanceerde functies te bieden die de gebruikersinterface in gevaar brengen en deze vol te proppen met extra functies.

De gebruikersinterface is esthetisch, functioneel en zonder alle toeters en bellen die dingen voor nieuwe gebruikers compliceren. Het aantal links, knoppen en one-click features is optimaal om toegang tot de beschikbare functionaliteit te garanderen.

Documentatie
De beste manier om meer te weten te komen over een bron is via de officiële documentatie. De documentatie van DigitalOcean is uitgebreid en concreet. Het bevat alles, van tutorials en installatiehandleidingen tot how-to-handleidingen en walkthroughs.

Van het opzetten van een eenvoudige LAMP-stack tot het implementeren van een complex Kubernetes-cluster, de documentatie omvat elk aspect dat een softwareontwikkelaar zou kunnen hinderen.

DigitalOcean API
DigitalOcean beschikt ook over een API. Met de DigitalOcean API kunnen droplets binnen DigitalOcean op een eenvoudige, programmatische manier beheerd worden met behulp van conventionele HTTP-verzoeken.

Alle functionaliteiten binnen het DigitalOcean-configuratiescherm zijn ook beschikbaar via de API, zodat complexe acties gescript kunnen worden (DigitalOcean API Docs | DigitalOcean Documentation, z.d.).

Betaalbare prijzen
De prijsstelling onderscheidt het van andere cloud computing-bedrijven die vergelijkbare hostingdiensten aanbieden. Het basisplan is vastgesteld op $ 5 per maand en de keuze uit betalingsopties per uur en per maand maakt het betaalbaar voor kleine startups en individuele softwareontwikkelaars om het platform te adopteren (Pricing Calculator, z.d.).

Schaalbaarheid en prestatie
Met DigitalOcean is het eenvoudig om een bestaande VPS te schalen. Met een druk op een knop is het mogelijk om de CPU, RAM of opslagruimte te schalen (How to Resize Droplets | DigitalOcean Documentation, z.d.).

Het is een van de eerste aanbieders van op SSD gebaseerde virtuele machines en het maakte gebruik van IPv6 voordat andere serviceproviders het zelfs maar overwogen. De druppels die DigitalOcean biedt, hebben een snelle opstarttijd van ongeveer 55 seconden (Resources - Cloud Performance Report, z.d.).

Locatie datacenters
Momenteel heeft DigitalOcean dertien datacenters op belangrijke locaties, waarmee het een aanzienlijk deel van de wereldwijde gebruikers kan bedienen. Tabel 11 toont de specifieke locaties.

Tabel 11 DigitalOcean datacenters (Regional Availability Matrix | DigitalOcean Documentation, z.d.)

Locatie	Datacenter(s)
New York City, United States	nyc1, nyc2, nyc3
Amsterdam, the Netherlands	ams2, ams3
San Francisco, United States	sfo1, sfo2, sfo3
Singapore	sgp1
London, United Kingdom	lon1
Frankfurt, Germany	fra1
Toronto, Canada	tor1
Bangalore, India	blr1

Actieve digitale community
Een van de beste dingen die dit cloudplatform biedt, is een actieve digitale community die helpt vragen te beantwoorden en feedback te geven, zodat iedereen ervan kan profiteren.

De DigitalOcean-infrastructuur heeft altijd nieuwe geavanceerde technologieën geïntegreerd, zelfs als ze nog niet volledig zijn uitgerijpt. De community-experts zijn er om te helpen.

2. **Microsoft Azure**

In de kern is Azure een openbaar cloud computing platform, met oplossingen zoals Infrastructure as a Service (IaaS), Platform as a Service (PaaS) en Software as a Service (SaaS), die kunnen worden gebruikt voor services zoals analyse, virtueel computergebruik, opslag, netwerken en meer.

Figuur 21. Azure

Momenteel gebruikt Social Brothers Microsoft Azure voor het hosten van webapplicaties waarbij een hoge beschikbaarheid vereist is. Het bedrijf vindt Azure prettig, omdat verschillende systemen zoals de database en servers door Azure beheerd worden. De hoge kosten zijn de reden dat het niet gebruikt wordt om de rest van de webapplicaties te hosten.

Azure wordt bekeken omdat Social Brothers tevreden is over de verschillende aspecten, op de prijs na. Mogelijk is het momenteel wel relevant voor Social Brothers om volledig gebruik te gaan maken van Azure, doordat de integratie met het .NET framework goed is vergeleken met de andere hosting platforms.

Integratie met .NET
Azure en het .NET framework zijn allebei ontwikkeld door Microsoft. Azure is altijd het eerste platform dat de hosting van het nieuwste .NET framework ondersteunt (Get Started with Azure and .NET, 2021). Bij de officiële documentatie van Microsoft over .NET staan handleidingen en tutorials over hoe een .NET applicatie op Azure gehost kan worden (Azure App Service Support for .Net 6 Now Generally Available | Azure, z.d.).

Documentatie
Azure heeft net als DigitalOcean een uitgebreide documentatie van producten die Azure tot zijn beschikking heeft (Azure Documentation, z.d.). De documentatie van Azure bevat net als DigitalOcean alles, van tutorials en installatiehandleidingen tot how-to-handleidingen en walkthroughs.

Deployment
Azure heeft een eigen CI/CD en Git-beheeroplossing: Azure Pipelines en Azure Repos. Samen met de integratie met het .NET framework maakt dit het deployen van een .NET 6 API gemakkelijker dan bij andere hosting platforms (Directory of Azure Cloud Services, z.d.).

Hoge beschikbaarheid
In tegenstelling tot andere cloud platforms biedt de Microsoft Azure-cloud een hoge beschikbaarheid en redundantie in datacenters op wereldwijde schaal. Hierdoor kan Azure een

Service Level Agreement of SLA aanbieden van 99,95% (ongeveer 4,38 uur downtime per jaar), iets wat de meeste bedrijven niet kunnen bereiken (Service Level Agreements Summary, z.d.).

Schaalbaarheid
Azure biedt net als DigitalOcean een eenvoudige manier om een VPS te schalen. Dat kan namelijk met een klik op een knop. Ook is het is mogelijk om dat automatisch te laten doen door Azure als reactie op internetverkeer of een gedefinieerd schema (Azure Virtual Machine Scale Sets Overview - Azure Virtual Machine Scale Sets, 2021).

Prijs
Het gebruik van een VPS in Azure is prijziger dan bij Digitalocean. Voor een soortgelijke VPS betaalt een klant bij Azure bijna 50% meer dan bij DigitalOcean (Pricing - Linux Virtual Machines, z.d.).

3. Heroku

Heroku is een cloud platform waarmee bedrijven apps kunnen bouwen, leveren, monitoren en schalen. Heroku is de snelste manier om van idee naar URL te gaan en infrastructuurproblemen te omzeilen (About Heroku | Heroku, z.d.).

Bij Heroku hoeft een bedrijf zich geen zorgen te maken over het infrastructuurprobleem van het hosten van een app. Het bedrijf kan zich focussen op het developen van de app, terwijl Heroku zorgt voor het schalen en deployen van de app.

Bij Heroku wordt code naar containers genaamd “dynos” gedeployed om te hosten. Dynos zijn gevirtualiseerde Linux-containers die zijn ontworpen om code uit te voeren op basis van een door de gebruiker opgegeven opdracht (Heroku Dynos | Heroku, z.d.).

Figuur 22. Heroku

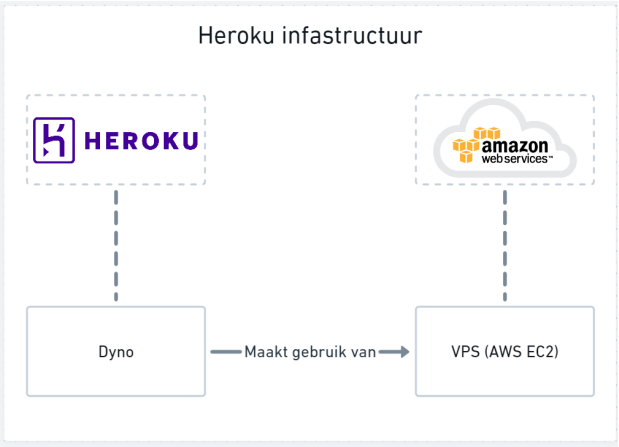
Deployment
Op het platform van Heroku zit een ingebouwde CI/CD tool. Het is mogelijk om bij het aanmaken van een dyno deze meteen aan een Git repository te linken. Bij het linken zijn de opties beschikbaar om de code naar een dyno te deployen bij een wijziging in een specifieke Git branch.

Ondersteuning .NET 6
Heroku ondersteunt momenteel .NET 6 niet officieel. Het is dus officieel niet mogelijk om .NET 6 API’s te deployen met Heroku. Wel is het mogelijk om via Docker de .NET 6 API’s te deployen.

Architectuur

Heroku is een managed omgeving, net als Microsoft Azure. Heroku maakt het proces om apps te deployen een stuk gemakkelijker vergeleken met DigitalOcean. Dat komt omdat Heroku meer tools biedt om het deploymentproces te vereenvoudigen. Het deployen naar een dyno kost minder tijd doordat de gebruiker na het linken van de Git repository niets meer hoeft te doen.

Het nadeel is dat de developer niet de vrijheid heeft om binnen de dyno toegang te krijgen. Bij DigitalOcean is het wel mogelijk om bij een VPS in te loggen via SSH of via een gebruikersnaam en wachtwoord.



Figuur 23. Heroku infrastructuur

Heroku maakt op de achtergrond gebruik van een VPS. Een dyno wordt gehost op een AWS EC2-instantie. Dat is een type VPS van Amazon Web Services, het hosting platform van Amazon (Figuur 23).

Locatie

Momenteel heeft Heroku datacenters op acht locaties, waarmee het een aanzienlijk deel van de wereldwijde gebruikers kan bedienen. Tabel 12 toont de locaties.

Tabel 12. Locaties Heroku

ID	Locatie	Runtime
EU	Europe	Common Runtime
US	United States	Common Runtime
Dublin	Dublin, Ireland	Private Spaces
Frankfurt	Frankfurt, Germany	Private Spaces
Oregon	Oregon, United States	Private Spaces
Sydney	Sydney, Australia	Private Spaces
Tokyo	Tokyo, Japan	Private Spaces
Virginia	Virginia, United States	Private Spaces

Prijs

Afgezien van het gratis abonnement, begint het betaalde abonnement bij $ 7 per dyno per maand. Dit is meer geschikt voor persoonlijke projecten. Voor professioneel gebruik biedt Heroku abonnementen van $ 25 tot $ 500 per dyno per maand (Pricing | Heroku, z.d.).

Momenteel betaalt Social Brothers op DigitalOcean per VPS $ 10 per maand om een .NET API te hosten. Dat is inclusief de databasehosting. Op Heroku zou een .NET API $ 25 per maand kosten, exclusief de database.

Om de database te hosten, is een add-on nodig. De prijs verschilt per database. De goedkoopste is de postgres database add-on van Heroku zelf met een gratis abonnement voor kleine projecten. Het goedkoopste abonnement kost $ 9 per maand en is eveneens bedoeld voor kleinere projecten. Voor professioneel gebruik biedt Heroku abonnementen van $ 50 tot $ 16000 per maand per database add-on (Elements Marketplace: Heroku Postgres, z.d.).

Schaalbaarheid en prestatie
Vergeleken met DigitalOcean is het net zo gemakkelijk om dyno's op te schalen. Er is slechts een druk op de knop nodig. Ook is het mogelijk om dat automatisch door Heroku te laten doen. Heroku heeft meer opties om te schalen. Zo kan gekozen worden tussen horizontaal schalen en verticaal schalen. Bij horizontaal schalen worden meer dyno's toegevoegd om het internetverkeer te spreiden tussen meerdere dynos en daarmee internetverkeer aan te kunnen. Bij verticaal schalen krijgt de dyno meer CPU en RAM (Heroku Dynos - Scaling | Heroku, z.d.).

Voor de prijs die betaald wordt, is de performance van DigitalOcean beter dan die van Heroku. Bij Heroku betaalt een ontwikkelaar meer dan dubbel de prijs voor wat de ontwikkelaar bij DigitalOcean krijgt.

Uitbreidbaarheid
Heroku beschikt over tools en services voor het ontwikkelen, uitbreiden en bedienen van apps. Deze worden door een externe provider of Heroku voor de ontwikkelaar onderhouden. Er bestaan add-ons, zodat ontwikkelaars zich kunnen concentreren op hun eigen applicatielogica en niet hoeven te steunen op de complexiteit van het op volle productiecapaciteit laten draaien van ondersteunende services (Add-Ons Overview | Heroku Dev Center, z.d.).

Documentatie
Heroku heeft net als DigitalOcean een uitgebreide documentatie. Alles is gescheiden in hoofdstukken, paragrafen, tutorials. Van handleidingen per programmeertaal tot het oplossen van foutmeldingen. De documentatie is zowel voor beginners als ervaren gebruikers relevant (Heroku Dev Center, z.d.).

Domeinen
Heroku ondersteunt net als DigitalOcean het linken van eigen domeinnamen. Bij Heroku gaat dat wel een stuk gemakkelijker. Er zijn namelijk meerdere mogelijkheden om dat te doen, bijvoorbeeld met de Heroku CLI en in het Heroku dashboard.

Wat Heroku onderscheidt van DigitalOcean is dat Heroku bij het aanmaken van een dyno automatisch gekoppeld is aan een Heroku-domein. Dit maakt het prettig om mee te werken, omdat de gebruiker zelf niet een testdomein hoeft te gebruiken of een IP-adres hoeft te onthouden (Custom Domain Names for Apps | Heroku Dev Center, z.d.).

Heroku API
De platform-API van Heroku stelt ontwikkelaars in staat om Heroku te automatiseren, uit te breiden en te combineren met andere services. Het is mogelijk om de platform-API te gebruiken om programmatisch apps te maken, add-ons te voorzien en andere taken uit te voeren die voorheen alleen konden worden uitgevoerd met de Heroku CLI of het dashboard.

5.4.3. Conclusie

Tabel 13 toont de CI/CD tools en Tabel 14 de hosting platforms. De CI/CD tools en hosting platforms zijn met elkaar vergeleken. De getallen zijn gebaseerd op het uitgevoerde onderzoek. De onderwerpen die worden gebruikt om te vergelijken, zijn gebaseerd op de belangrijkste verschillen tussen CI/CD tools en hosting platforms.

Tabel 13. Eigenschappen CI/CD tools

	Jenkins	**Bitbucket pipelines**	**CircleCI**	**Azure Pipelines**	**Travis CI**	
Open Source	Ja	Nee	Nee	Nee	Nee	
Gebruiksgemak*	2/5	4/5	4/5	3/5	2/5	
Setup**	2/5	4/5	3/5	4/5	2/5	
ingebouwde functies***	3/5	2/5	4/5	2/5	3/5	
Integratie****	5/5	2/5	4/5	2/5	3/5	
Hosting	Zelf	Cloud	Zelf en Cloud	Cloud	Zelf en Cloud	* **1 = Moeilijk, 5= Makkelijk** ** **1= Moeilijk, 5 = Makkelijk** *** **1 = Weinig, 5 = Veel** **** **1 = Slecht, 5 = Goed**
Prijs	Gratis	$	$$$	$	$$	

Tabel 14 Eigenschappen hosting

	DigitalOcean	**Microsoft Azure**	**Heroku**	
Prijs	$	$$	$$$	
Gebruiksgemak	5/5	4/5	5/5	
Integratie	3/5	5/5	5/5	
Deployment .NET 6 API	2/5	5/5	3/5	* **1= Moeilijk, 5 = Makkelijk** ** **1 = Slecht, 5 = Goed** *** **1 = Moeilijk, 5 = Makkelijk** **** **1 = Slecht, 5 = Goed**
Overzicht	5/5	3/5	5/5	

5.5. Deelvraag 5

Welke manier om .NET 6 API's automatisch te deployen is het meest geschikt voor Social Brothers?

5.5.1. Methoden

Deze deelvraag is beantwoord door deskresearch te doen in de vorm van vergelijkend onderzoek en door een semigestructureerd interview te houden. De oplossingen van deelvraag 4 zijn met elkaar vergeleken. Daarna zijn de medewerkers van de backend-afdeling geïnterviewd over de deploymentoplossingen. Door de interviews (Bijlage E en Bijlage F) werd duidelijk welke oplossing het best bij Social Brothers past.

Uiteindelijk is de architectuur beschreven om een .NET 6 API automatisch te deployen.

5.5.2. Uitvoering

Bij deelvraag 4 zijn oplossingen bekeken om .NET 6 API's automatisch te deployen. Om de oplossing mogelijk te maken, zijn een CI/CD tool en hosting platform nodig. In Tabel 13 en 14 staan de resultaten.

Wensen hosting platform en CI/CD tool

Op 11 februari 2022 zijn Backend team lead (Pim) en Backend developer (Hugo) geïnterviewd via Slack (Bijlage E en Bijlage F) over wat ze van belang vinden voor de onderzochte deploymentoplossingen. In Tabel 15 en 16 staat wat de backend-afdeling van Social Brothers van belang vindt voor CI/CD tools en een hosting platform.

Tabel 15. Wensen CI/CD tool

Hosting	
Belangrijk	**Beschrijving**
Firewall	Dit vindt Social Brothers belangrijk, omdat het dan mogelijk wordt om de toegang tot de server te beheren. Met een firewall kan ingesteld worden welke IP-adressen wel en niet toegang hebben tot de server.
SSH keys	Dit is van belang zodat mensen met een bepaalde SSH key toegang kunnen krijgen tot de server zonder daarvoor te hoeven inloggen met een gebruikersnaam en wachtwoord.
Prijs	De prijs van een server moet niet te hoog zijn. Momenteel is Social Brothers tevreden met de kosten van DigitalOcean, namelijk $ 10 voor een droplet. Dat is inclusief de database. Als de prijs hoger is, moeten volgen Social Brothers meer functionaliteiten aangeboden worden.
Overzicht	Social Brothers host veel projecten voor klanten. Het vinden van een project binnen het hosting platform moet overzichtelijk zijn. Bij DigitalOcean is het mogelijk om mapjes aan te maken om zo projecten te scheiden.
Monitoring	Dit is belangrijk om te zien of een applicatie goed draait. Het huidige hosting platform van Social Brothers biedt monitoring aan om CPU, RAM, schijfgebruik en netwerkgebruik

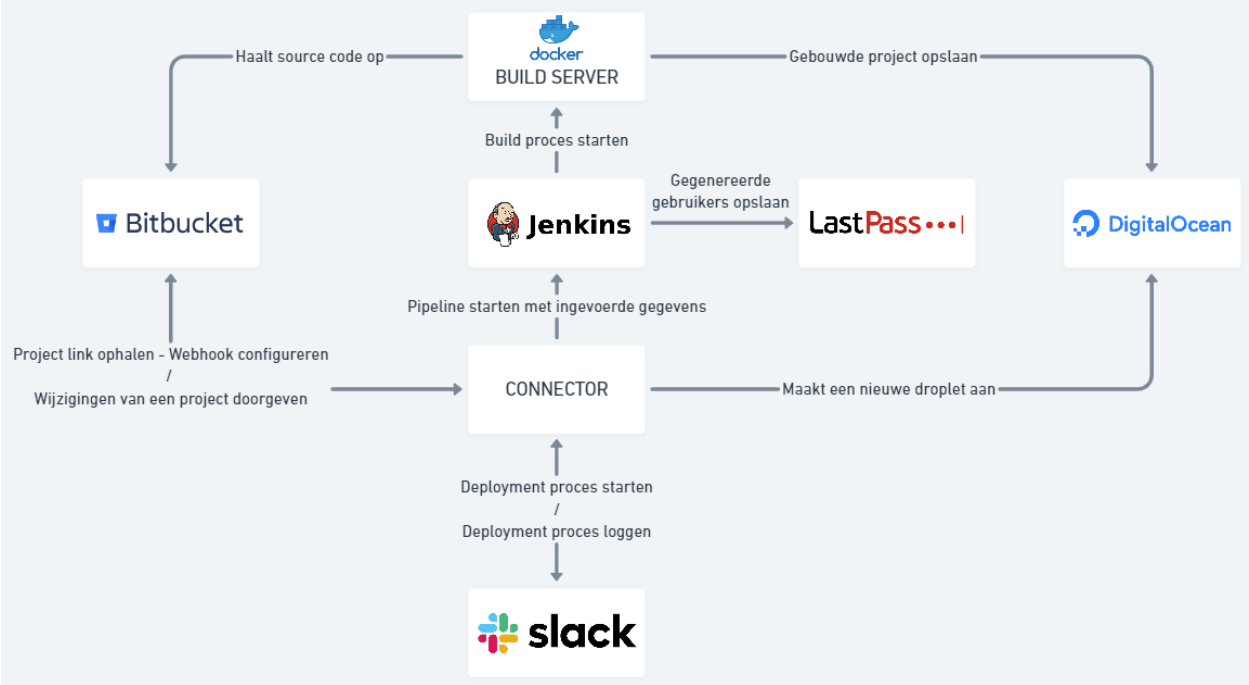
	van een droplet te zien.
Docker	Social Brothers vindt het gebruik van Docker interessant. Het zou een pre zijn als het hosting platform het gebruik van Docker ondersteunt.

Tabel 16. Wensen Hosting

CI/CD tools	
Belangrijk	**Beschrijving**
Uitbreidbaar	Social Brothers wil dat de CI/CD tool niet vastzit aan het hosting platform of de Git-beheeroplossing. De CI/CD tool moet uitbreidbaar zijn en ingezet kunnen worden voor andere hosting platforms. Ook is het belangrijk om een integratie te hebben met Slack, zodat de statussen van een deployment zichtbaar zijn.
Prijs	Social Brothers is bereid om voor het gebruik van een CI/CD tool te betalen. De betaalde oplossing moet dan wel meer functionaliteiten en gebruiksgemak bieden dan de gratis oplossingen.
Cloud	Social Brothers heeft een voorkeur voor een hosted CI/CD cloud in plaat van self-hosted CI/CD cloud. Dit wil Social Brothers, omdat het dan niet nodig is om zelf de CI/CD tool op te zetten en te beheren.
Gebruiksgemak	De initiële opzet van een CI/CD tool mag moeilijk zijn. Daarna wil Social Brothers dat het gebruik van de CI/CD tool gemakkelijk is.
Logging	Tijdens het deployen van een .NET 6 API moet inzichtelijk zijn welke stappen worden uitgevoerd en welke stappen goed en niet goed zijn uitgevoerd.

Gekozen oplossing

Er zijn geen CI/CD tools die volledig aansluiten op de wensen van Social Brothers. Social Brothers heeft wel een duidelijk voorkeur voor een hosting platform. In Figuur 24 staat een overzicht van de gekozen deploymentoplossing.



Figuur 24. deployment oplossing

Jenkins

Jenkins is de CI/CD tool die het beste bij Social Brothers past. Het bedrijf wil namelijk een CI/CD-oplossing hebben die uitbreidbaar is en goed een goede integratie heeft met meerdere systemen. Daarnaast is Jenkins betaalbaar. Jenkins is een open source CI/CD tool en dus gratis. Er hoeft alleen voor de hosting betaald te worden. Jenkins is moeilijker op te zetten dan de andere onderzochte CI/CD tools, maar is na het opzetten gemakkelijk te gebruiken.

Jenkins heeft een goede integratie met Slack. Jenkins heeft daar een plugin voor. Het is relatief eenvoudig om een Slack-kanaal aan te maken en te integreren met Jenkins om het deploymentproces te loggen.

In dit deploymentproces is Jenkins het brein van de architectuur. Jenkins heeft namelijk twee belangrijke verantwoordelijkheden. Ten eerste is Jenkins verantwoordelijk voor het configureren van de VPS die door de Connector is aangemaakt. Tijdens de configuratie worden de updates uitgevoerd, wordt de dev-gebruiker aangemaakt, wordt de root uitgezet en worden gegenereerde wachtwoorden naar lastpass gestuurd. Ten tweede is Jenkins verantwoordelijk voor het starten van het bouw- en deployproces.

DigitalOcean
Social Brother is tevreden over het gebruik van DigitalOcean. Alle belangrijke punten die Backend team lead (Pim) en Backend developer (Hugo) hebben besproken en die beschreven zijn in Tabel 15 zijn beschikbaar in DigitalOcean. Tijdens de interviews (Bijlage E en Bijlage F) waren ze positief over DigitalOcean. Wel waren er puntjes die beter konden. Zo vonden ze de monitoring niet gedetailleerd genoeg.

Op DigitalOcean is het mogelijk om alles te hosten. Het is bijvoorbeeld niet een platform voor specifieke applicaties, zoals Heroku. DigitalOcean ondersteunt bijvoorbeeld wel het gebruik van Dockerapplicaties.

In dit deploymentproces heeft DigitalOcean twee verantwoordelijkheden. Ten eerste is DigitalOcean verantwoordelijk voor het hosten van het project dat naar de droplet gedeployed wordt. Ten tweede is het verantwoordelijk voor het opslaan van het gebouwde project.

Bitbucket
Social Brothers gebruikt Bitbucket om de source code op te slaan en te beheren. Het is tevreden over het gebruik van Bitbucket en niet van plan over te stappen naar een andere source code-beheerder.

In dit deploymentproces is Bitbucket verantwoordelijk voor het opslaan van de source code en het aangeven van wijzigingen op branches. Deze wijzigingen kunnen opgevangen worden door Jenkins.

Lastpass
Zoals beschreven in deelvraag 2, maakt Social Brothers al gebruik van Lastpass om inloggegevens op te slaan. Het bedrijf is hier tevreden over en niet van plan om ervan af te stappen. Lastpass beschikt over een API, waarbij het mogelijk is om via een HTTP(S) call wachtwoorden naar de password manager te sturen en op te slaan. De password manager past dus goed bij de deploymentoplossing. LastPass is in dit deploymentproces verantwoordelijk voor het opslaan van inloggegevens die DigitalOcean via de LastPass API verstuurt.

Slack
Social Brothers gebruikt Slack als communicatieapplicatie. Daarnaast wordt het gebruikt om foutmeldingen en processen van projecten te loggen. Zoals beschreven in deelvraag 2 vindt Social Brothers dit loggen van groot belang.

Slack heeft in dit deploymentproces twee verantwoordelijkheden. Ten eerste is het verantwoordelijk voor het starten van het deploymentproces. De gebruikers vullen de gegevens in die via de Connector naar Jenkins wordt gestuurd. Ten tweede is Slack verantwoordelijk voor het tonen van foutmeldingen tijdens het gehele deploymentproces.

Connector
De Connector is nodig om de ingevoerde gegevens die via Slack binnenkomen juist door te sturen naar Jenkins. De ingevoerde gegevens moeten namelijk gevalideerd worden voordat ze doorgestuurd worden. De Connector is verantwoordelijk voor de volgende punten:

1. het valideren van de ingevoerde gegevens via Slack;
2. het aanmaken van een nieuwe droplet;
3. het bijhouden van gedeployde projecten;
4. het doorsturen van foutmeldingen naar Slack.

Build Server
De build server is nodig om de source code in een Docker image te bouwen. De Docker image wordt dan opgeslagen in de DigitalOcean container registry.

5.5.3. Conclusie

Hieronder toont Tabel 17 de deployment tools die het beste bij Social Brothers passen.

Tabel 17. Gekozen oplossingen

Type	Keuze	Beschrijving
Hosting	DigitalOcean	Social Brothers gebruikt momenteel DigitalOcean als hosting platform en is daar tevreden over. Vooral de prijs-kwaliteitsverhouding is goed. Na een vergelijking met andere hosting providers is uiteindelijk voor deze hosting provider gekozen, omdat deze voldoet aan alle eisen van Social Brothers.
CI/CD tool	Jenkins	Jenkins is de meest geschikte CI/CD tool voor Social Brothers, omdat deze voldoet aan de belangrijkste eisen die het bedrijf gesteld heeft, namelijk dat het een goedkope en uitbreidbare CI/CD oplossing is die geïntegreerd kan worden met andere systemen door het gebruik van plugins. Het enige nadeel is dat Social Brothers het zelf moet onderhouden.

6 Proof of concept

6.1.1. Methoden

De proof of concept dat door de afstudeerder gemaakt is kan een .NET 6 API op een automatische wijze deployen. Na de onderzoeksfase is aan de proof of concept gewerkt volgens de Scrum methode. Dit werd gedaan omdat de proof of concept werd beschouwd als een intern backend project. De backend afdeling binnen Social Brothers werkt alleen maar volgens de scrum methode. Om de kwaliteit van de code te bewaken was het ook de bedoeling om voor elke “feature” van de proof of concept de code te laten controleren door een collega. Dat moest gedaan worden voordat het toegevoegd kon worden aan de “master” branch. Het doel van de proof of concept is om te bewijzen dat het mogelijk is met de gemaakte keuzes tijdens het onderzoek een .NET 6 API te deployen op een automatische wijze.

6.1.2. Uitvoering

Tijdens de uitvoering van deelvraag 2 is er gevraagd naar de belangrijkste requirements van het nieuwe deploymentproces. De requirements voor de proof of concept zijn als volgt:

1. Geen onnodige handmatige invoer of acties.
2. Elke stap moet gelogd worden.
3. Gegenereerde wachtwoorden moeten veilig opgeslagen worden.
4. Het deploymentproces moet consistent zijn.
5. Het deploymentproces moet herstelbaar zijn.

Geen onnodige handmatige invoer of acties

In de vorige deploymentproces beschreven in deelvraag 1 wordt in totaal 12 stappen beschreven die uitgevoerd moeten worden om een .NET API te deployen. Deze stappen bestaan uit meerdere handmatige acties die uitgevoerd moeten worden door het hele deploymentproces. Veel van deze stappen zijn zoals beschreven in deelvraag 1 overbodig.

In de gerealiseerde proof of concept wordt het aantal stappen teruggedrongen naar maar 1 handmatige actie het invoeren van de deployment gegevens. Deze stap hoeft maar een keer uitgevoerd te worden aan het begin van de deployment via Slack. De gegevens die via slack ingevoerd moet worden zijn als volgt:

1. Droplet naam
2. Bitbucket link
3. Servertype (production, development of staging)
4. Naam van de database
5. Naam van het project
6. Domeinnaam
7. Branch van het project
8. Eventuele tags

Elke stap moet gelogd worden

Logging is erg belangrijk omdat het dan mogelijk is om te zien of iets goed gegaan is of niet. Dit gebruikt Social Brothers voor bijna al hun projecten. Vooral met een automatische deployment systeem moet er achterhaald kunnen worden of alle stappen van een nieuwe deployment goed zijn uitgevoerd. Voor de proof of concept worden alle uitgevoerde stappen gelogd. Dit wordt gedaan in hetzelfde Slack kanaal waar de deployment is uitgevoerd. Door gebruik te maken van de soepele integratie van Slack en Jenkins is dit gemakkelijk te doen.

Gegenereerde wachtwoorden moeten veilig opgeslagen worden

In het deployment proces beschreven in deelvraag 1 moeten alle wachtwoorden handmatig ingevoerd worden bij het aanmaken van een gebruiker. De gegevens van de aangemaakte gebruiker moet dan handmatig in LastPass ingevoerd worden.

In de gerealiseerde proof of concept worden alle wachtwoorden automatisch gegenereerd door gebruik te maken van Urandom, een programma dat wachtwoorden kan genereren. De wachtwoorden worden automatisch ingevoerd bij het aanmaken van een gebruiker en opgeslagen in LastPass via de LastPass API.

Het deploymentproces moet consistent zijn

Het deploymentproces dat uitgevoerd wordt door de gerealiseerde proof of concept is consistenter dan het huidige deploymentproces. Dit komt omdat alle stappen automatisch worden uitgevoerd op dezelfde manier. Alle gegevens die aan het begin van het deploymentproces worden ingevoerd worden van te voren gecontroleerd op validiteit.

Het deploymentproces moet herstelbaar zijn

Het deploymentproces is herstelbaarder gemaakt door gebruik te maken van Docker en door het deploymentproces in Jenkins te splitsen in drie processen.

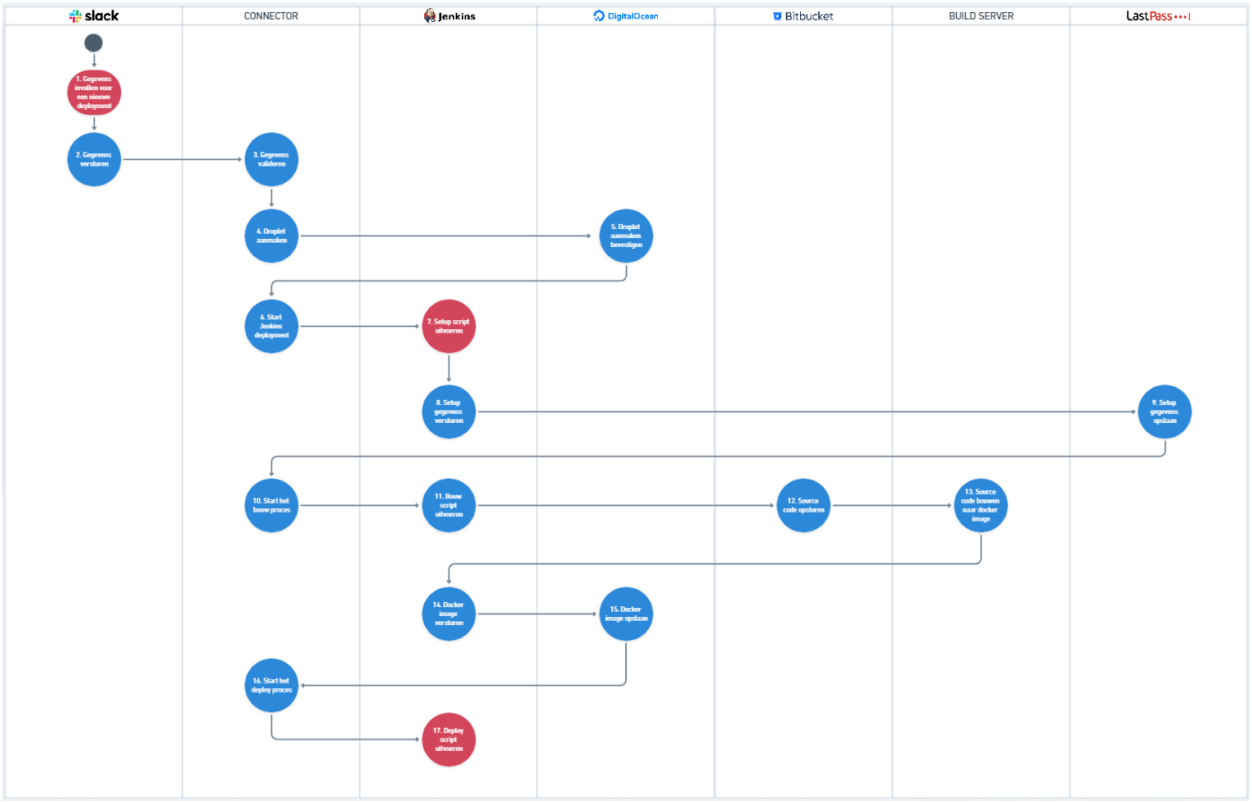
Door gebruik te maken van Docker is het gemakkelijk te zien of een bepaald systeem goed draait. Docker heeft interne tools waarbij het mogelijk is om logs op te vragen binnen een systeem. Ook zijn de systemen makkelijk te verwijderen en opnieuw te installeren als iets niet goed draait.

Het deploymentproces is in Jenkins gesplitst in drie processen. Create, build en deploy. De eerste stap bestaat uit het aanmaken van de server en het aanmaken van de gebruikers. De tweede stap bestaat uit het bouwen van het project. De derde stap bestaat uit het ophalen van het gebouwde project om hem vervolgens in de aangemaakte server op te starten.

Het opsplitsen van het deploymentproces maakt het herstelbaarder omdat ze alle drie los uitvoerbaar zijn. Als bijvoorbeeld het build proces faalt dan is het mogelijk om hem los uit te voeren in plaats van het nogmaals uitvoeren van het hele proces.

Architectuur

Hieronder staat in Figuur 25 per stap het nieuwe deploymentproces beschreven. Elke stap wordt gelogd in Slack. In Tabel 18 worden de stappen 1, 7 en 17 extra toegelicht.



Figuur 25. Nieuwe .NET 6 deploymentproces

Tabel 18. Toelichting stappen

Stap	Beschrijving	Toelichting
1	Gegevens invullen voor een nieuwe deployment	De gegevens die ingevuld moeten worden zijn als volgt: 1. Droplet naam 2. Bitbucket link 3. Servertype (production, development of staging) 4. Naam van de database 5. Naam van het project 6. Domeinnaam 7. Branch van het project 8. Eventuele tags

7	Setup script uitvoeren	De setup script bestaat uit de volgende stappen: 1. MySQL en NGINX docker compose bestanden ophalen 2. MySQL database starten 3. Nieuwe dev en remote gebruikers aanmaken 4. Nieuwe database schema aanmaken 5. Nieuwe dev server gebruiker aanmaken
17	Deploy script uitvoeren	De deploy script bestaat uit de volgende stappen: 1. Docker compose bestand ophalen voor de API 2. Docker image ophalen 3. Domeinnaam linken aan het IP adres 4. Appsettings configureren voor de API 5. Migraties uitvoeren 6. API starten 7. NGINX server starten 8. Slack loggen dat het deployment proces geslaagd is.

7 Conclusie

Wat is de beste manier om .NET 6 API's te deployen, zodat automatisch veilige inloggegevens worden gegenereerd voor de backendsystemen die de .NET 6 API's nodig heeft om te functioneren en waarbij deze inloggegevens zo veilig mogelijk worden opgeslagen, zodat de deployment van .NET 6 API's consistent, veilig en minder foutgevoelig wordt?

Het huidige deploymentproces voldoet niet langer aan de verwachtingen van Social Brothers. Er zijn teveel handmatige acties nodig om een .NET 6 API te deployen.

Social Brothers heeft in het verleden geëxperimenteerd met verschillende deploymentprocessen. Een van de systemen waarmee het geëxperimenteerd heeft is Azure. Dit systeem is eenvoudig om mee te werken. Het is namelijk mogelijk om met een druk op de knop .NET projecten live te zetten. Echter, voor het gebruiksgemak van Azure moet een hoge prijs betaald worden, een prijs die voor Social Brothers te hoog is.

Het huidige deploymentproces van Social Brothers biedt veel flexibiliteit. DigitalOcean maakt het mogelijk om van alles te hosten, maar de flexibiliteit gaat ten koste van het gebruiksgemak. Het biedt namelijk geen tools om op een eenvoudige manier .NET 6 API's te deployen, zoals Azure wel doet. De backend-afdeling heeft voor het deploymentproces verschillende scripts gemaakt om .NET projecten gemakkelijker te deployen, maar is er niet tevreden over. De scripts zijn namelijk niet consistent en vereisen nog steeds veel handmatige invoer.

Allereerst zijn verbeterpunten bekeken in de scripts die Social Brothers gebruikt om .NET API's te deployen. Bij het gebruik van de scripts is het nodig om handmatig inloggegevens in te voeren om gebruikers aan te maken. De inloggegevens moeten handmatig in de password manager LastPass ingevoerd worden om op te slaan. Dit proces kan geautomatiseerd worden door de scripts aan te passen en het programma Urandom te gebruiken om veilige wachtwoorden te genereren. Daarna is het mogelijk om de LastPass API te gebruiken om de aangemaakte gebruikers op te slaan.

Het huidige proces is bekeken en er is onderzocht wat nodig is om het volledige proces te automatiseren. Om het proces te automatiseren, moet geïnvesteerd worden in een CI/CD tool. Dit is een programma dat softwareontwikkelaars helpt softwaredeployment te automatiseren. Met de CI/CD tool is het mogelijk om op een automatische manier scripts uit te voeren. Zo is het mogelijk om via HTTP de DigitalOcean API aan te roepen om een droplet aan te maken en daarna de nieuwe scripts uit te voeren.

Verder zijn hosting platforms bekeken die het deploymentproces kunnen vereenvoudigen, maar er zijn geen hosting platforms die beter bij Social Brothers passen dan het huidige DigitalOcean.

Momenteel zijn er veel verschillende CI/CD tools op de markt. Door medewerkers van de backend-afdeling meerdere keren te interviewen (21 december 2021 en 11 februari 2022) is gebleken dat er geen CI/CD tool is die aan alle eisen van Social Brothers voldoet. Uiteindelijk is Jenkins gekozen, omdat deze aan de meeste eisen voldoet.

Door de gewenste situatie en de eisen en wensen van Social Brothers te bekijken, is uiteindelijk gekozen voor de beschreven architectuur in Figuur 24 en voor het deploymentproces beschreven in Figuur 25.

De beste manier om .NET 6 API's te deployen, is door Bitbucket te gebruiken als Git-based hosting service, Jenkins als CI/CD tool en DigitalOcean als hosting platform. Slack wordt gebruikt om het deploymentproces te starten. De scripts worden door Jenkins uitgevoerd tijdens het aanmaken van een droplet in DigitalOcean. De gegenereerde gebruikers worden daarna verzameld in de DigitalOcean droplet en via de API naar Lastpass gestuurd voor opslag.

Door dit deploymentproces te gebruiken wordt het deployen van een .NET 6 API consistent, veilig en minder foutgevoelig.

8 Aanbevelingen

Op basis van het antwoord op de hoofdvraag zijn aanbevelingen gedaan. Deze aanbevelingen maken Social Brothers duidelijk welke stappen het moet zetten na het realiseren van het nieuwe deploymentproces.

Tijdens de realisatie van het nieuwe deploymentproces is het handig om het bestaande deploymentproces te blijven gebruiken. Social Brothers heeft namelijk geen andere optie.

Na het realiseren van het nieuwe deploymentproces is het aan te bevelen om het eerst te testen op interne projecten. Het kan namelijk gebeuren dat het door onverwachte omstandigheden een project niet kan deployen.

In het deploymentproces zijn er een aantal stappen die een aantal seconden wachten. Dit veroorzaakt soms problemen als een proces langer duurt dan normaal. Om dit systeem robuuster en productieklaar te maken moeten deze stappen worden vervangen. Deze stappen kunnen worden vervangen door te controleren of de vorige stap goed is uitgevoerd.

Verder moeten de medewerkers van Social Brothers ingelicht worden over het nieuwe deploymentproces. Ze moeten weten welke commando's ze in Slack moeten invoeren om een nieuwe deployment uit te voeren en weten hoe ze kunnen zien of een deployment goed is uitgevoerd.

Vervolgonderzoek dient te worden verricht om te onderzoeken hoe de appsettings van de .NET API 6 het beste toegevoegd kan worden aan het project tijdens het deploymentproces. Ook moet onderzocht worden wat de beste manier is om migraties uit te voeren zonder daarvoor source code op de server te hoeven downloaden.

9 Verantwoording resultaten

Hieronder wordt beschreven of tijdens het uitvoeren van de opdracht voldaan is aan de kwaliteitscriteria. Deze kwaliteitscriteria zijn beschreven in het plan van aanpak.

Modellen
Vooraf is in het plan van aanpak beschreven dat alle modellen moeten voldoen aan de UML-standaard. Dit is gedaan door gebruik te maken van Lucidchart en Whimsical.

De Unified Modeling Language, afgekort UML, is een modelmatige taal om objectgeoriënteerde analyses en ontwerpen voor een informatiesysteem te maken. UML is in de jaren negentig ontworpen door Grady Booch, James Rumbaugh en Ivar Jacobson en is sinds 1997 de standaard (Unified Modeling Language, z.d.).

Documenten
De opgestelde documenten voldoen aan de standaard beschreven in Van Steehouder & Jansen, 2016. De documenten zijn (bijna) elke week gecontroleerd door de docentbegeleider Philippine

en bedrijfsbegeleider Hugo. Ook het feedbackproces is verlopen zoals beschreven in het plan van aanpak.

Source code
Ook aan de kwaliteitscriteria voor de source code-opslag is voldaan. De source code van het deploymentproces wordt opgeslagen op Bitbucket, de Git-based hosting service die wordt gebruikt binnen Social Brothers.

10 Evaluatie procesgang

Er zijn geen onverwachte risico's opgetreden. Het grootste risico tijdens het onderzoek was het implementeren van de proof of concept. De eerste architectuurbeschrijving was namelijk niet realiseerbaar, omdat er onvoldoende kennis was over Jenkins. De architectuur is aangepast door er een connector tussen te bouwen. Op die manier was toch de proof of concept te realiseren volgens de eisen en wensen van Social Brothers.

De planning is tijdens de afstudeerperiode aangepast. Dit komt omdat sprake was van miscommunicatie tijdens de planningsfase. De afstudeerperiode begon op 15 november 2021 en er werd vanuit gegaan dat de afstudeerperiode 20 weken zou duren, maar tijdens de oplevering van de tweede proof of concept bleek dit niet te kloppen. De afstudeerder kreeg na de oplevering namelijk te horen dat het tweede concept pas op 16 mei ingeleverd hoefde te worden en de definitieve scriptie op 30 mei.

Het onderzoek is anders uitgevoerd dan beschreven in het plan van aanpak. Bij deelvraag 5 moest namelijk nog een extra interview uitgevoerd worden om de beste CI/CD tool en hosting server te achterhalen.

De communicatie met de Hogeschool Utrecht en het afstudeerbedrijf verliep goed. Om de twee weken hadden de afstudeerder en de docentbegeleider een gesprek om de voortgang van de scriptie te bespreken. De bedrijfsbegeleider had elke week een moment ingepland met de afstudeerder om de scriptie en de proof of concept te bespreken.

Het afstudeerbedrijf is tevreden over de gerealiseerde proof of concept en producten. Ook is de aanbeveling voor het gebruik van de proof of concept goedgekeurd. Het afstudeerbedrijf gaat de proof of concept eerst gebruiken voor kleine, interne projecten. Tijdens deze fase wordt bekeken of de proof of concept goed werkt. Als dit het geval is, wordt het in de rest van de projecten ingezet.

Bibliografie

Bitbucket: What is Bitbucket? | Evaluator Resources. (z.d.). Atlassian Documentation. Geraadpleegd op 30 december 2021, van https://confluence.atlassian.com/confeval/development-tools-evaluator-resources/bitbucket/bitbucket-what-is-bitbucket

Password must meet complexity requirements (Windows 10) - Windows security. (2021, 29 oktober). Microsoft Docs. Geraadpleegd op 10 januari 2022, van https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/password-must-meet-complexity-requirements

xkcdpass. (2021, 5 september). Xkcdpass. Geraadpleegd op 10 januari 2022, van https://pypi.org/project/xkcdpass/

Gitflow git flow. (z.d.). [Illustratie]. https://leanpub.com/git-flow/read. https://leanpub.com/git-flow/read

Henry-Stocker, S. (2018, 31 juli). Using gpg encryption to protect files on Linux. Network World. Geraadpleegd op 12 januari 2022, van https://www.networkworld.com/article/3293052/encypting-your-files-with-gpg.html

pwgen(1): make pronounceable passwords - Linux man page. (z.d.). Pwgen. Geraadpleegd op 12 januari 2022, van https://linux.die.net/man/1/pwgen

TransIP. (z.d.). Wat is een VPS? TransIP. Geraadpleegd op 8 december, 2021, van https://www.transip.nl/knowledgebase/artikel/118-wat-is-een-vps/

What is UML | Unified Modeling Language. (z.d.). UML. Geraadpleegd op 8 december 2021, van https://www.uml.org/what-is-uml.htm

Social Brothers. (2020, 8 juni). Digital bureau, Websites, Apps en Online Marketing. Geraadpleegd op 30 januari 2022, van https://socialbrothers.nl/

Jenkins. (z.d.). Jenkins. Geraadpleegd op 6 februari 2022, van https://www.jenkins.io/doc/

Belmont, J. (2018). Hands-On Continuous Integration and Delivery: Build and release quality software at scale with Jenkins, Travis CI, and CircleCI. Packt Publishing.

Get started with Bitbucket Pipelines | Bitbucket Cloud. (2020, 9 december). Atlassian Support. Geraadpleegd op 6 februari 2022, van https://support.atlassian.com/bitbucket-cloud/docs/get-started-with-bitbucket-pipelines/
What is Azure Pipelines? - Azure Pipelines. (2021, 18 oktober). Microsoft Docs. Geraadpleegd op 6 februari 2022, van https://docs.microsoft.com/en-us/azure/devops/pipelines/get-started/what-is-azure-pipelines?view=azure-devops

Use Azure Pipelines - Azure Pipelines. (2021, 17 november). Microsoft Docs. Geraadpleegd op 6 februari 2022, van https://docs.microsoft.com/en-us/azure/devops/pipelines/get-started/pipelines-get-started?view=azure-devops

Travis CI's new pricing plan threw a wrench in my open source works | Jeff Geerling. (z.d.). Travis CI. Geraadpleegd op 6 februari 2022, van https://www.jeffgeerling.com/blog/2020/travis-cis-new-pricing-plan-threw-wrench-my-open-source-works#:%7E:text=Nov%202%2C%202020%3A%20Travis%20CI,put%20into%20read%2Donly%20mode.

DigitalOcean. (z.d.). Second Largest Web Hosting Provider. Geraadpleegd op 6 februari 2022, van https://www.digitalocean.com/press/releases/digitalocean-now-worlds-second-largest-web-hosting-provider

DigitalOcean API Docs | DigitalOcean Documentation. (z.d.). DigitalOcean API. Geraadpleegd op 6 februari 2022, van https://docs.digitalocean.com/reference/api/api-reference/

Pricing Calculator. (z.d.). Pricing. Geraadpleegd op 6 februari 2022, van https://www.digitalocean.com/pricing/calculator

How to Resize Droplets | DigitalOcean Documentation. (z.d.). droplet resize. Geraadpleegd op 6 februari 2022, van https://docs.digitalocean.com/products/droplets/how-to/resize/#:%7E:text=You%20can%20resize%20Droplets%20from,or%20using%20the%20DigitalOcean%20API.&text=Next%

Resources - Cloud Performance Report. (z.d.). Performance DigitalOcean. Geraadpleegd op 6 februari 2022, van https://www.digitalocean.com/resources/cloud-performance-report

Regional Availability Matrix | DigitalOcean Documentation. (z.d.). Region DigitalOcean datacenters. Geraadpleegd op 6 februari 2022, van https://docs.digitalocean.com/products/platform/availability-matrix/

Get started with Azure and .NET. (2021, 15 september). Microsoft Docs. Geraadpleegd op 6 februari 2022, van https://docs.microsoft.com/en-us/dotnet/azure/intro

Azure App Service support for .Net 6 now generally available | Azur. . . (z.d.). Microsoft Azure. Geraadpleegd op 6 februari 2022, van https://azure.microsoft.com/en-us/updates/azure-app-service-support-for-net-6-now-generally-available/

Azure documentation. (z.d.). Microsoft Docs. Geraadpleegd op 6 februari 2022, van https://docs.microsoft.com/en-us/azure/?product=popular

Directory of Azure Cloud Services. (z.d.). Microsoft Azure. Geraadpleegd op 6 februari 2022, van https://azure.microsoft.com/en-us/services/

Service Level Agreements Summary. (z.d.). Microsoft Azure. Geraadpleegd op 6 februari 2022, van https://azure.microsoft.com/en-us/support/legal/sla/summary/

Azure virtual machine scale sets overview - Azure Virtual Machine Scale Sets. (2021, 2 november). Microsoft Docs. Geraadpleegd op 6 februari 2022, van https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/overview

Pricing - Linux Virtual Machines. (z.d.). Microsoft Azure. Geraadpleegd op 6 februari 2022, van https://azure.microsoft.com/en-us/pricing/details/virtual-machines/linux/

About Heroku | Heroku. (z.d.). About Heroku. Geraadpleegd op 6 februari 2022, van https://www.heroku.com/about

Heroku Dynos | Heroku. (z.d.). Dynos. Geraadpleegd op 6 februari 2022, van https://www.heroku.com/dynos

Pricing | Heroku. (z.d.). Heroku Pricing. Geraadpleegd op 6 februari 2022, van https://www.heroku.com/pricing

Elements Marketplace: Heroku Postgres. (z.d.). Heroku add-ons. Geraadpleegd op 6 februari 2022, van https://elements.heroku.com/addons/heroku-postgresql

Heroku Dynos - Scaling | Heroku. (z.d.). Heroku Scaling. Geraadpleegd op 6 februari 2022, van https://www.heroku.com/dynos/scaling

Add-ons Overview | Heroku Dev Center. (z.d.). Heroku Add-Ons Overview. Geraadpleegd op 6 februari 2022, van https://devcenter.heroku.com/articles/add-ons#:%7E:text=Heroku%20add%2Dons%20are%20components,party%20provider%20or%20by%20Heroku.

Heroku Dev Center. (z.d.). Heroku Devcenter. Geraadpleegd op 6 februari 2022, van https://devcenter.heroku.com/

Custom Domain Names for Apps | Heroku Dev Center. (z.d.). Heroku Domains. Geraadpleegd op 6 februari 2022, van https://devcenter.heroku.com/articles/custom-domains

Van Gumster Feed, J. (2018, 30 augustus). 6 places to host your git repository. Opensource.Com. Geraadpleegd op 4 maart 2022, van https://opensource.com/article/18/8/github-alternatives

Nguyen, O. (2021, 22 december). Best 14 CI/CD Tools You Must Know | Updated for 2022. Katalon Solution. Geraadpleegd op 4 maart 2022, van https://www.katalon.com/resources-center/blog/ci-cd-tools/

Bijlagen

Bijlage A: Interview Backend developer (Hugo) D1

Danny: Hugo, kan jij beschrijven, hoe het huidige deployment proces van een .NET API er uit ziet?

Hugo: Oké. Op het moment dat wij en .NET API klaar hebben, ik moet wel zeggen dat .NET 6, gebruiken wij nog niet dus eigenlijk dus we werken momenteel met .NET 5. Wij gaan natuurlijk binnenkort migreren naar .NET 5.

Het zal voor de grootste deel hetzelfde werken, maar ik zal gewoon vertellen hoe het proces zelf nu gaat. Op het moment dat wij een API klaar hebben en die moet op een server terechtkomen of we bouwen er nog aan. Wat wij doen, wij navigeren naar DigitalOcean, onze VPS host/beheer centrum en daar maken wij een VPS aan. Dat is een VPS van 10 dollar, met als operating system een linux met Ubuntu, die wordt gevestigd in Amsterdam.

Wat wij doen, de server wordt voor ons aangemaakt en dan krijgen wij een IP-adres en daarmee kunnen wij via SSH inloggen we hebben nu twee Batch scripts staan die ons helpt met server configuratie eentje is voor een initiële setup van de server. Wat normaal de eerste stap is dat wij als met “root” inloggen en dat is een script die een “dev” user aanmaakt, dus dat je niet meer op “root” zit, configureert de firewall, die installeert een matchfix agent van DigitalOcean zodat we betere monitoring krijgen. En ja, volgens mij is het altijd een beetje

Daarna heb je dus de dev user. Daarmee log je in, en dan voer je de script uit specifiek voor .NET maar we hebben ook een lamp, script voor Apache. Maar .NET, doen wij met NGINX en .NET zelf natuurlijk. Dus dat script, wat die doet, is MySQL installeren, .NET installeren en NGINX installeren, de firewall daarvoor configureren Certbot installeren, en de system service configureren voor het .NET proces.

Dus hoe dat werkt, is zodra MySQL is geïnstalleerd is en de .NET repository hebben gecloned dan moeten wij eerst het .NET proces, het DLL runnen als het ware in Linux. Dat doen we nu momenteel door een daemon service dus daar configureren wij een service voor, die wordt opgestart bij de service samen en dat is gewoon een proces die .NET gebruikt om de DLL te draaien. En nou ja, de DLL die staat geconfigureerd op port 5000 en via NGINX met reverse proxy leiden wij de gebruiker door naar port 5000, via de reverse proxy kunnen wij ook, ssl zo configureren op een bepaald domein en dat is het eigenlijk. Je maakt ook een MYSQL aan via het script.

Danny: Waarom hebben jullie gekozen voor voor zo'n deployment proces?

Hugo: Ja, eigenlijk is dat iets wat we al een aantal jaren gebruiken. Digitalocean is een systeem waarin je heel makkelijk zo'n VPS aan kan slingeren en voor ons is dat ook gewoon heel interessant.

Wij hebben redelijk veel klanten met een redelijk kleine user base, dus we hoeven ook niet zoveel managed hosting features te gebruiken en dat is natuurlijk vaak veel duurder vergeleken met Azure is het peperduur voor dingen die wij hebben.

Dus als wij voor elke klant die een kleine API heeft, een Azure omgeving zouden moeten draaien, dan zou dat veel te veel geld kosten en die klanten willen dat toch niet betalen. Dus, Digitalocean is een goedkopere oplossing. Vrijere oplossing, flexibeler natuurlijk ook redelijk wat verschillende projectjes dus dan is het eigenlijk ook wel relaxed om zelf daar de macht over te hebben, hoe je het configureert en installeert.

Wij doen dit al een hele tijd en dat bevalt goed en het werkt goed. We hebben natuurlijk die Batch scripts kort geleden opgezet om het nog wat meer te automatiseren, want eerst werd het nog veel met de hand gedaan. Vroeger volgde we een tutorial op Google en dan nou ja, iedereen gebruikt weer een andere tutorial. Dus het was net iets anders.

Dus nu voor consistentie hebben ook scripts. Er zijn twee mensen binnen het bedrijf die echt heel veel met SSH op de servers gaan, en dat zijn Pim en ik en de andere mensen die dat doen, die worden daar langzaam maar in als geïntroduceerd.

Zonder de scripts zullen ze niet direct hele gekke dingen gaan doen. Ehm ja, dus eigenlijk voelt het veilig.

Danny: Vind je de scripts consistent?

Hugo: Momenteel wel, ja, redelijk consistent. Maar eigenlijk ook niet heel consistent, want het wordt met scripts uitgevoerd, waar nog handmatig acties bij komen kijken, zoals het zelf wachtwoorden moeten genereren. Dus elke keer een SSH user wordt aangemaakt of een linux user op de server dan moet je daar een wachtwoord voor genereren, die opslaan. Als er een MYSQL user wordt aangemaakt, moet je een wachtwoord genereren en die opstaan, en dat drie keer, dus dat zijn redelijk veel handmatig acties. Dat maakt het redelijk onhandig. Als je één keer iets verkeerds invoert per ongeluk of je drukt op een verkeerde toets, dan gaat waarschijnlijk iets verkeerd in de script, waardoor het in één keer verkeerd gaat en je, ja eigenlijk pech hebt. Dus dan zou je net zo goed opnieuw kunnen beginnen, de vps verwijderen en weer opnieuw aanmaken niet ideaal, dus het heeft zeker mankementen.

Danny: Je had het net over de Batch scripts, die je kan uitvoeren op een op de server. Gebruikt iedereen dat, of doet iedereen dat op een andere manier?

Hugo: Dat is een kwestie van opvoeden. Die scriptie zijn er nu natuurlijk en we proberen ze eigenlijk altijd als er vps wordt aangemaakt, te gebruiken. De scripts bestaan al een paar

maanden, twee maanden ofzo en ze zijn tot nu toe eigenlijk elke keer gebruikt. Alleen, ja, we hebben nu ook al een tijdje niet meer VPS hoeven aan te maken en ik ben wel benieuwd van stel, ik ben volgende week op vakantie en Max of Pim maakt een vps aan, wordt dan niet vergeten om dat te gebruiken. daar ben ik benieuwd naar, maar ik denk dat het zeker ook prima kan gaan. Het is alleen ja, je kan het niet forceren en dat is het gevaar. Dus het zou geweldig zijn als het vrijwel geen manuele handeling bij komt kijken, om maar zo te zeggen dat je niet hoeft te vertrouwen op het geheugen van een persoon.

Danny: Jullie huidige manier van deployment hebben jullie dat vroeger anders gedaan? Of is dit altijd al zo geweest?

Hugo: We gebruiken Digitalocean al een hele tijd en die scripts gebruiken we sinds kort. Dus vroeger was het inderdaad, zoals ik zei, tutorials opzoeken en en de software installeren en ook heel vaak trial en error, omdat sommige mensen niet zo ervaren zijn. Maar verder is het een beetje hopen dat het allemaal lukt.

Azure is daarnaast ook een tijdje gebruikt voor .NET deployment. Heel prettig, omdat het gewoon een beetje plug and play is of tenminste je kan met een paar klikken een server hebben draaien op Azure en je hoeft daar bijna niks voor te doen. Heel relaxed alleen, het kost wel geld en ik vind Azure niet heel overzichtelijk. Vooral niet voor onze manier van werken met heel verschillende klanten, heel veel verschillende projecten. In het verleden, is onderzoek gedaan naar het gebruik van Heroku, maar dat is nooit gebruikt. Digitalocean is eigenlijk wel de oudste die we gebruiken en ik denk sinds 2018 al.

Hugo: Daarvoor gebruikten we nu je het vraagt Antagonist, dus dan had je gewoon één server waar PHP op stond, en daar stonden dan verschillende projecten op. Dus dat was heel slecht, maar dat was heel erg. In het begin stadia van Social Brothers hadden we ook nog onze Wordpress hosting op Antagonist en dat soort dingen.

Danny: Hoe verschilt het deployment proces in vergelijking met bijvoorbeeld de Frontend afdeling of is dat hetzelfde?

Hugo: Goeie vraag, Frontend nou ja, als we het hebben over de maatwerkafdeling, want we hebben natuurlijk de maatwerkafdeling, en Wordpress afdeling bij Social Brothers. We hebben het nu over maatwerk front end die bouwt in React en React native voor mobiele apps dus we hebben natuurlijk verschillende front end die moeten worden gedeployed.

De Front-end afdeling heeft de CI/CD aardig op orde, die gebruiken voor de Front-end web apps Vercel. Vercel is een product van, NextJS een hosting systeem waar je een project aan kan maken daarin link, jij een bitbucket repository.

Vanaf dat moment gaat vercel voor elke push, naar wat voor branche dan ook (kun je zelf wel configureren) een deployment maken. Dus dan kun je eigenlijk: Elk stukje code die jij, waar dan ook pushed, testen na een paar minuten nadat het project is gebuiled op Vercel.

Vercel regelt SSL. Vercel regelt domein. Je kan de DNS makkelijk naar vercel verwijzen. Je hebt daarvoor een hele makkelijke stappenplan. Het is een heel chill systeem om mee te werken.

Hugo: Voor React Native gebruiken ze nu CircleCI, daar wordt ook op code push of merges ook allemaal processen in gang gezet. Waardoor in de app center in Microsoft een android app beschikbaar wordt gesteld voor gebruikers.

Danny: Zijn er nog dingen die je kwijt wil over het huidige, deployment proces dat wellicht interessant is voor mij om te weten?

Hugo: Voor ons huidige, deployment proces, die scriptie werken en de handmatige acties zijn het vervelendst en dat zijn de keren dat je op enter moet druk en de keren dat je toch nog even met de hand iets moet aanpassen in één van de bestanden op de server. De keren dat je een wachtwoord moet genereren. Die zijn momenteel gewoon wel het meest foutgevoelig. Natuurlijk zorgen de scripts voor een hoop consistentie maar het is nog steeds niet dichtgetimmerd dat is vervelend, dus daar wil ik heel graag van af.

Bijlage B: Interview Backend team lead (Pim) D1

Danny: kan jij beschrijven hoe het huidige deployment proces eruit ziet voor een .NET API?

Pim: We hebben een check out van het project op de servers staan. Dan doen we alle dingen die nodig zijn, zoals database installeren. Negen van de tien keer wordt dat op een digitalocean server gedaan. En een van de tien keer wordt dat op een TransIP server gedaan als een ISO certificering nodig is.

Pim: Vroeger gebruikte we Azure en daarna hadden we gekeken of het mogelijk was om het überhaupt DigitalOcean te gebruiken en dat werkt prima en dat ging goed en dat kost vijf euro in plaats van 150. Dus dan valt Azure snel weg. Pas als we van die enterprise applicaties gaan hosten, dan gaan we weer terug naar Azure.

Danny: Het huidige manier van software deployment. Voert iedereen dezelfde stappen uit?

Pim: We waren gewend om het handmatig te doen, want dat werkt prima. Toen hebben we de handmatige acties geprobeerd te automatiseren door scripts te gebruiken, nou beginnen er steeds meer mensen bij te komen, dus dan volstaat het eigenlijk niet meer.

Danny: Hoe ervaar jij de scripts die gemaakt zijn?

Pim: Ik heb ze twee keer gebruikt. Bij de tweede keer ging het niet goed en dan ben ik gelijk de draad kwijt, want dat is een script. Als ik hem handmatig doe en het gaat niet goed, dan weet ik ook wat ik gedaan heb en dan fix ik het weer. Met het script was het chaos, heb ik de hele server weggegooid en gevraagd of Hugo hem aan wilde maken. Dus het werkt niet optimaal.

Danny: Wat vind je minder goed aan het deployment proces? Dus de scripts? En wat nog meer?

Pim: Ja de scripts werken dus niet heel goed.

Pim: Wat ik wel goed vind, is dat de servers nu steeds meer hetzelfde zijn ingesteld. De structuur van de mappen zijn hetzelfde dat was vroeger al anders, dus dat is wel goed. En wat niet zo goed is, is dat het niet werkt, niet stabiel op het moment, dus het is niet stabiel en het is moeilijker uitbreidbaar denk ik.

Danny: Wat vind je nog meer positief aan het huidige deployment proces behalve de folderstructuur?

Pim: Dus dat dat de servers gewoon hetzelfde ingericht worden, het is allemaal met LetsEncrypt. Je weet waar de projecten staan op de server, vroeger was dat even zoeken. Ze

hebben allemaal een dev user waarmee je inlogt. Dus dat is wel fijn. Dat scheelt wel dat je niet elke keer weer opnieuw moet gaan zoeken naar de informatie die nodig hebt.

Danny: Wat vind je van de handmatige acties die uitgevoerd moeten worden om de dev user en inloggegevens te genereren?

Pim: Ja, veel onnodige acties, omdat het dezelfde acties zijn die uitgevoerd moeten worden. Dat is heel vervelend. Het zou fantastisch zijn als dat gedeelte gewoon automatisch gaat.

Danny: Hebben jullie vroeger eigenlijk andere manier gehad om software te deployen?

Pim: Sinds ik hier zit (oktober 2018), is het wel Digitalocean.

Danny: Weet jij ook toevallig hoe andere afdeling hun software deployen?

Pim: Front end gebruikt vercel, ben er niet heel bekend mee. De wordpress afdeling gebruikt DeployHQ. Ze hebben in het verleden gebruik gemaakt van docker. Dat vond ik ook wel interessant, maar ook weinig ervaring mee.

Danny: Zijn er nog dingen over het huidige deployment proces dat wellicht interessant is voor mij om te weten?

Pim: Houd rekening met Letsencrypt. Houd rekening met de broncode die liever niet op de server staat. Houd rekening met dat het misschien interessant is als de ontwikkel, staging, development en productie dezelfde inrichting heeft. Het is niet dat je van die dingen hebt, dat het werkt op staging en op productie gaat het kapot. Er zijn een hoop dingen die verbeterd kunnen worden.

Bijlage C: Interview Backend developer (Hugo) D2

Danny: Dan ga ik je nu vragen stellen over de gewenste situatie. Dus wat willen jullie eigenlijk? Wat is jullie ideale deployment proces?

Hugo: Dit steel ik van de front-end afdeling. Sluit goed aan bij je vorige vraag.

Hugo: Dat je gewoon naar Vercel navigeert een nieuw project aanmaakt daar je repository aan linkt en dat verder alles geregeld is, dat je gewoon een domein hebt met ssl. Dat elke branche wijziging te testen is op een bepaalde url.

Vercel heeft ook redelijk goeie versioning, dus je kan makkelijk de deployments terugzetten en zo komt er eigenlijk niks handmatigs meer bij kijken en dat is gewoon heel prettig. Dus het enige workflow die jezelf moet creëren is je gitflow in in je repository. Dus als een feature branch wordt aangemaakt kun je ze testen op Vercel.

Op Vercel kan je een pull request maken op develop. Als die wordt goedgekeurd dan komt die op de develop variant van Vercel te staan. Daar kun je weer allemaal DNS aan linken. Je dev domein, je staging domein. je productie domein. Dat voorbeeld zou super chill zijn ook voor ons.

Danny: En Vercel. Is dat ook een open-source of moet je daarvoor betalen?

Hugo: Het is betaald, maar je betaalt voor gebruikers accounts, niet voor deployments. Dus je betaalt per gebruiker en niet voor hoe veel projecten je hebt staan. Dus dat is redelijk uniek. Op DigitalOcean betaal je natuurlijk per server die je hebt draaien. Ehm ja, dat is op zich prijstechnisch best wel prima, omdat het gewoon niet duur is. Maar als je kijkt naar Vercel dat je alleen betaald voor gebruikers, daar zal natuurlijk het grootste gedeelte, ook hun hosting door gecompenseerd worden. En maar ja, dat is ook wel interessant. Maar Vercel is natuurlijk alleen maar voor React, dus dat heeft nul toegevoegde waarde voor onze use case alleen de werking ervan vind ik heel interessant.

Danny: En als ik bijvoorbeeld aan kom met een soort van Vercel voor back-end software zou dat dan ook goed zijn?

Hugo: Ja, zeker, ja, kijk uiteindelijk, ik ben wel gewend aan de vrijheid die je hebt op een server om om bepaalde dingen te kunnen installeren. Hè, dus als ik kijk naar Vercel daar heb je weinig invloed op hoe en wat, en wat goed is aan bijvoorbeeld DigitalOcean, is dan weer van stel, we willen een keer voor een project Redis gebruiken als in memory database, dan kunnen we gewoon naar DigitalOcean gaan, dat installeren en ons code daaraan linken en dat is natuurlijk heel chill hoe je dat soort dingen en dan nog wel waarborgt geen idee.

Danny: Ja, inderdaad, dus als ik het goed begrijp, met nieuwe deployment proces is het doel om zo min mogelijk handmatige acties uit te voeren?

Hugo: Ja, want meeste wat wij doen is gewoon klaar, gewoon eigenlijk altijd hetzelfde. Er zijn heel specifieke uitzonderingen waarin wij dus zo'n Redis installatie nodig hebben. Maar ja, als het niet nodig is, dan voorkomen we dat ook.

Danny: Oké, en zijn er nog meer doelen die jullie willen bereiken met het nieuwe, deployment proces, behalve acties automatiseren.

Hugo: Ja, ik zou er ook goede logging voor willen. Dat we wel goed kunnen inzien wat er gebeurd. En dat enige SSH toegang nog steeds wel prettig zou zijn of in ieder geval dat je bij de bestanden via de repository kan. Dus bijvoorbeeld, stel, je hebt inderdaad een project. Dat je daar nog wel inzicht hebt in de logs van zo'n .NET proces.

Hugo: En dat ook een bepaalde stappen van deployment ook worden gelogt naar een bijvoorbeeld een slack channel. We hebben ook een Vercel channel waar je alle deployment die worden geïnitieerd en geslaagd zijn of gefaald zijn, ook worden geplaatst.

Danny: Zijn er nog dingen die je kwijt wilt over het nieuwe deployment proces?

Hugo: Ja, misschien een heel klein puntje. Sommige klanten van ons hebben redelijk gevoelige gegevens die ze binnen onze systeem gaan opslaan waar we mee gaan koppelen, denk aan medische bedrijven. Die zijn vereisen een ISO certificering dat betekent dus dat we voor die bedrijven niet hosten op DigitalOcean, want dat is een Amerikaans bedrijf. Voor ISO vereiste applicaties hosten wij bij TransIP, een Nederlands bedrijf die ook die certificering heeft, waar wij dus ook geoorloofd bij kunnen hosten. In sommige gevallen zou het dus nodig zijn dat wij ons product gaan hosten op een ISO gecertificeerde omgeving, dat is een sitenote. Dat is geen vereiste, want DigitalOcean gebruiken we ook als main hosting platform.

Bijlage D: Interview Backend team lead (Pim) D2

Danny: Hoe ziet jouw ideale deployment proces eruit?

Pim: Code wordt geschreven in een branch, daar wordt een pull request voor aangemaakt naar develop, die wordt goedgekeurd of afgekeurd. Maar laten we zeggen dat het wordt goedgekeurd. Dan worden unit testen uitgevoerd. Dan wordt die code automatisch gedeployed naar de develop server vanuit de develop branch als de unit testen goed zijn gegaan. Daar worden op de develop server eventueel migraties uitgevoerd die in de code zaten. De code wordt gebuild en daarna op de server gezet.

Pim: Dan herhaalt het riedeltje zich, want develop wordt uiteindelijk naar master gemerged. Vanaf master gaat dat automatisch naar staging. Dan moeten we een manier vinden om dat automatisch van een branch naar productie te gaan deployen. Maar dat vind ik een hele spannende stap.

Pim: Dus voor nu zou mijn droom zijn dat het tot aan staging automatisch geregeld is en dat we productie, dat dat gefaciliteerd wordt, maar dat hoeft niet per se volautomatisch te gaan.

Danny: Waarom zou je dat eng vinden als het ook naar productie automatisch zou gaan?

Pim: Omdat ik eerst wil zien dat het goed gaat met staging, want staging mag ontploffen, productie niet. Dus ik wil eerst zien dat het eigenlijk altijd goed gaat met staging deployment en dan naar productie als dat goed gaat.

Pim: Als productie kapot gaat, dan moet je eerst op de server de backup terugzetten. En dan ga je waarschijnlijk data verliezen dus als het misgaat op productie vind ik dat spannend.

Danny: Wat is jullie belangrijkste doel die jullie willen bereiken met het nieuwe deployment proces?

Pim: Tijd besparen, ik vind het opzetten van servers irritant. Het zoeken naar de informatie die je nodig hebt is irritant. Zelf unit test gaan draaien voor het hele project kost tijd. Dat is mijn belangrijkste doel. Ervoor zorgen dat het niet zoveel tijd kost.

Danny: Welke back-end systemen gebruiken jullie allemaal waarvoor inloggegevens gegenereerd moeten worden?

Pim: MYSQL en login voor de server.

Danny: Zijn er nog system die jullie in de toekomst willen gebruiken?

Pim: Docker. Ik heb een paar keer naar docker gekeken maar het lukt me nooit om daar een beetje goede implementatie van te maken, maar ik vind het concept dat je ontwikkel omgeving hetzelfde is als alle andere. Dat vind ik wel heel interessant.

Pim: Wat ik belangrijk vind is dat het deployment proces veilig is, tijd bespaard. En dat de omgevingen waar de applicaties in draaien consistent en veilig zijn.

Pim: Moet de nieuwe oplossing gratis zijn of mag het ook een betaalde oplossing zijn?

Pim: Het hoeft niet gratis, maar als het niet gratis is, dan moet het wel echt een all-in-one pakket zijn waarbij we gewoon alles kunnen.

Danny: Wat vind jij zelf van Azure? Dat is volgens mij ook goede all-in-one pakket

Pim: Het is wel duur, want je betaalt niet enkel voor de deployment maar ook de hosting van de servers zijn best prijzig omdat het een managed omgeving is.

Pim: Wij hebben voor een klant een proof of concept draaien die een userbase van honderdduizend actieve gebruikers per dag verwacht. Als het eenmaal draait, dan gaan we naar Azure toe, want dan komt dat hele load-balancing erbij en er komt een bepaalde uptime bij. Dus dan is dat heel chill om een managed omgeving te hebben. Als wij dat allemaal moeten onderhouden, beheren, configureren dan gaat het fout gaat dus dan is Azure wel interessant, maar ik denk dat de meeste projecten die we doen niet geschikt zijn.

Pim: En verder terug te komen op je originele vraag, of ik nog opmerkingen had. Ik denk dat het heel belangrijk is dat we het ook meenemen in de organisatie, en dan vooral de afdeling. Dus de mensen van de afdeling moeten op de hoogte zijn van het proces, niet zo diep als dat wanneer ze het handmatig zouden doen. Maar het is belangrijk dat als iemand weet dat je op de knop approve of merge drukt, dat het dan ook gedeployed wordt en ook op een server gezet wordt en dat die een melding kan krijgen, als testen falen. Dus dat is voor mij wel heel belangrijk: dat mensen een beetje weten hoe het proces er uit ziet.

Bijlage E: Interview Backend developer (Hugo) D5

Danny: Wat vind je goed aan DigitalOcean?

Hugo: De kosten, Firewalls, Monitoring, load-balancing. DigitalOcean heeft een gebruiksvriendelijke manier om projecten te kunnen categoriseren. Het biedt heel veel features aan voor de prijs. Het is ook makkelijk om een droplets aan te maken.

Danny: Zijn er ook dingen die jij slecht vindt?

Hugo: Logging zou uitgebreider kunnen. En de monitoring van een droplet zou ook uitgebreider kunnen.

Danny: En ze: zouden jullie bereid zijn om voor een andere platform te kiezen als dat bijvoorbeeld het deployment proces makkelijker maakt.

Hugo: Ja. Als het meer biedt dan wat DigitalOcean biedt.

Danny: Wat vind jij dan belangrijk in een hosting platform?

Hugo: De vrijheid om van alles te hosten. Wij hosten bij DigitalOcean niet alleen maar .NET projecten maar ook PHP. Je wilt niet verschillende hosting platforms hebben voor verschillende projecten.

Danny: Wat vind jij belangrijk in een CI/CD tool?

Hugo: Makkelijk te beheren wanneer het is opgezet. Dat het ingezet kan worden voor meerdere hosting platforms. Dat het goedkoop is.

Danny: Hoe belangrijk vindt je de beschikbare locaties van een hosting platform?

Hugo: Ja wel belangrijk. Het project moet gehost worden dichtbij het doelgroep. Momenteel zijn er veel projecten die in Nederland draaien. Dus locatie in Nederland is belangrijk. Het zou kunnen dat we in de toekomst een project internationale klanten krijgen. Dan is het kunnen hosten in het buitenland ook belangrijk.

Danny: Vind jij open source belangrijk in een CI/CD tool?

Hugo: Nee dat is niet zo belangrijk.

Danny: Hoe belangrijk vind je de kosten van een CI/CD tool?

Hugo: Als het een betaalde oplossing is dan moet het wel meer features hebben dan een onbetaalde oplossing.

Danny: Vind jij gebruiksgemak belangrijk? Hoe hoog zo jij dat beoordelen?

Hugo: Erg belangrijk. Het is erg belangrijk dat het makkelijk is om te gebruiken. Als iets stuk gaat dan wil je niet veel tijd besteden om het te fixen.

Danny: Hoe hoog zou jij uitbreidbaarheid van de CI/CD tools beoordelen? Dus bijvoorbeeld integratie met Slack of andere systemen.

Hugo: Ja dat is wel belangrijk. Wij gebruiken natuurlijk veel gebruik van Slack het is wel belangrijk dat er logging is zodat wij weten wat er gebeurd.

Danny: Vind jij het belangrijk dat het self hosted of cloud hosted is?

Hugo: Mijn voorkeur gaat naar cloud hosted, omdat je dan zelf niet dingen hoeft te hosten. Is dan ook minder overhead.

Danny: Architectuur laten zien. [Mening over architectuur vragen] Bitbucket - Bitbucket Pipelines - DigitalOcean - LastPass.

Hugo: Misschien slim om Slack te gebruiken om de initiële setup te doen voor. Zeg maar het aanmaken van de server en de scripts uitvoeren.

Bijlage F: Interview Backend team lead (Pim) D5

Danny: Welke tools je daarvoor nodig? En ja, en tot de conclusie komen dat je daarvoor een CI/CD tool nodig hebt en een hosting platform. Een hosting platform hebben jullie al. Ik heb nog steeds naar verschillende hosting platforms gekeken. Om te kijken of het beter werkt met het deployen van software. Dus wat vinden jullie goed aan DigitalOcean?

Pim: De kosten vind ik goed.

Pim: Want we hadden eerst de .NET applicaties op Azure draaien, en toen zijn we in gesprek geweest met Azure over hoe je het goedkope kan en anders kan, en die konden eigenlijk niet concurreren met DigitalOcean. Dat zei die vent toen ook letterlijk van "ja, met die prijs ga ik niet concurreren". Dus toen zijn we, maar naar DigitalOcean gegaan. Dus moesten we alle Microsoft SQL's omzetten naar MySQL's en toen kosten het gemiddeld tien euro per maand opeens.

Danny: Dus alleen de prijs, ook andere dingen die jullie goed vinden van DigitalOcean?

Pim: Ze hebben wat features. daarbij zitten zoals een API en een load balancer en Firewall.

Pim: En het managen van SSH keys zodat gelijk de juiste mensen toegang hebben als je een nieuwe droplet aanmaakt. Daar kan je verschillende linux distributies kiezen, verschillende datacenters dus het is eigenlijk een open omgeving. Dan Azure, maar ze hebben wel een aantal managed features die alleen bij je grote dingen zoals Azure ziet. Dus is dat is wel chill.

Danny: Wat kan zo'n firewall feature?

Pim: Bij de firewall kun je eigenlijk zeggen welke IP adressen er binnenkomen en uitgaan en welke je toestaat. Dat kun je dan op server niveau en ook op account niveau dus dan is het voor alle servers instellen.

Pim: Een goed voorbeeld, De SSH login, dus die gaan over een bepaalde poort en die poort accepteert alleen maar verbindingen van specifieke IP's. Bij een aantal servers dus die Firewall die zorgt daarvoor.

Danny: Zijn er nog meer dingen?

Pim: Ja, als ik in DigitalOcean ga kijken, ga ik wel meer dingen vinden voor je. Ik weet niet of we zo ver willen gaan, ben er namelijk tevreden over.

Danny: Dat is dan prima. Zijn er ook dingen die jij slecht vindt?

Pim: Slecht weet ik niet. Dingen die beter kunnen is in ieder geval wat je dus nu hebt in zo'n een unmanaged omgeving is dat je zelf verantwoordelijkheid hebt om je server ook up to date

houden en dat soort dingetjes een security update uitvoeren. Dat soort geneuzel daar dan vind ik wel jammer. Dat is wel vervelend, omdat het allemaal beter moeten houden en voor de rest niet echt. Ja, het zijn ook andere features die ik mis zoals uitgebreider logging.

Danny: Hoe kan je dan nog uitgebreider?

Pim: Dat de logging buiten de server kan bereiken.

Pim: Dat is iets wat je ook wel in die managers omgevingen veel ziet dat is wel heel handig. Een die twee dingetjes denk ik.

Danny: Is de logging dan niet zelf aan te maken?

Pim: Ja, maar goed, het is natuurlijk de diensten. We hadden het over de diensten die de hosting partijen aanbiedt want wat we nu hebben qua logging is natuurlijk Sentry, dat doet wel iets, maar diepe errors op de server zelf, die komen niet in het Sentry naar voren, want dan is het al gecrasht voordat Sentry is opgestart. Dus die mis ik en kunnen we zelf doen. Maar ik denk dat het wel pittig is, pittiger.

Danny: En ze: zouden jullie bereid zijn om voor een andere platform te kiezen als dat bijvoorbeeld het deployment proces makkelijker maakt.

Pim: Ja. Als er duidelijke voordelen heeft, dan waarom niet?

Danny: Wat vind jij dan belangrijk in een hosting platform?

Pim: Dat het een acceptabele prijs heeft. Dat ik zelf genoeg controle houd om te tweaken voor servers die niet helemaal lekker lopen.

Danny: Weet jij wat een CI/CD is?

Danny: Heb je daar enige ervaring mee?

Pim: Heb een klein beetje ervaring met jenkins.

Danny: En heb jij ook ervaring met Bitbucket Pipelines?

Pim: Ja, Hugo, meer, ik heb ze wel een keer gezien, maar we nooit echt veel mee gedaan zelf.

Danny: Wat vind jij belangrijk in een CI/CD tool?

Pim: Dat we er geen omkijken naar hebben, nadat het is opgezet. Het opzetten mag best wel tijd kosten, maar als het eenmaal draait, moet het gaan draaien. Belangrijk is dat het unit testen

uitvoert dat zou chill zijn, dus dat die verifieert dat wat online gaat, ook goed is, dat die de in het geval van .NET de broncode niet op de server komt te staan.

Pim: Je wilt hem van tevoren builden en dan pas naar de server toe. Misschien dat we snel terug kunnen dus als die gedeployed heeft en er is iets helemaal fout gegaan dat we dat snel kunnen reverten.

Danny: Wat bedoel je met reverten? Hoe zie je dat voor je?

Pim: Terug naar de vorige versie het hoeft niet per se een feature van die tool te zijn, maar het kan natuurlijk ook via Bitbucket dat je vorige commit als master neerzet of zo. Maar dan moet wel rekening mee gehouden worden. Want wat doen we als het misgaat want als we uiteindelijk allemaal via de tools gaan, deployen en het veel minder handmatig doen, en dan wordt handmatig opeens een skill die waarschijnlijk niet iedereen gaat hebben, maar iedereen moet wel iets kunnen doen als het misgaat.

Danny: Vind jij open source belangrijk in een CI/CD tool?

Pim: Dat maakt mij niet uit.

Danny: En de kosten dan?

Pim: Vind ik moet ook iets over zeggen. Het mag iets kosten. Als dat gewoon de functionaliteit waard is, als het ons tijd bespaart, dan is het natuurlijk al heel snel het geld waard.

Danny: Vind jij gebruiksgemak belangrijk? Hoe hoog zo jij dat beoordelen?

Pim: In het uiteindelijke gebruik moet het makkelijk zijn. Het opzetten van de server per project kan wat complexer zijn, want ik verwacht, als we dat dan één keer goed hebben gedaan, dat dat gewoon herhalen wordt van diezelfde stappen.

Pim: Of het liefst, dat dan geautomatiseerd, maar het uiteindelijk gebruik daarvan moet. Natuurlijk, ja, het liefst onzichtbaar zijn, weet je wel. Ik zou het gebruiksgemak zelf erg hoog beoordelen.

Danny: Hoe hoog zou jij uitbreidbaarheid van de CI/CD tools beoordelen? Dus bijvoorbeeld integratie met Slack of andere systemen.

Pim: Ja, ik denk er blijkbaar, het is wel een hele goeie, want we hebben natuurlijk nu DigitalOcean. Dan heb je voor een andere klant, weer een andere partij omdat zij een ISO certificering willen. Dan gaan straks een kwestie van tijd voordat de projectjes teruggaan naar Azure of andere platforms met een ISO certificering. Als de tool platform onafhankelijk is dat zou dat geweldig zijn.

Danny: Vind jij het belangrijk dat het self hosted of cloud hosted is?

Pim: Maakt mij niet uit.

Danny: Want Cloud hosted moet je meestal wel voor betalen. Self hosted is meestal een gedoe met zelf beheren.

Pim: Ja, maar ik mij op zich niet zo heel veel uit.

Danny: Architectuur laten zien. [Mening over architectuur vragen] Bitbucket - Jenkins - DigitalOcean - LastPass.

Pim: Dat ziet er goed uit. Dus er wordt nu twee processen uitgevoerd? Slack voor de initiële setup en daarna gaat dat automatisch?

Danny: Ja

Pim: Oke klinkt goed.