



De kanarie in de kolenmijn: testen tijdens livegang

Onderzoeksrapport

Afstudeeronderzoek HBO-ICT
Voor de cursus Graduation Software Engineering (CU75050)
aan de HZ University of Applied Sciences

Student Daniel de Cloet

Studentnummer 78998

Datum 19 juni 2022

Inhoudsopgave

1	Inleiding	2
1.1	Aanleiding	2
1.2	Organisatiebeschrijving	2
1.3	Afbakening	3
1.4	Probleem- en doelstelling	3
1.5	Onderzoeksopzet	5
1.6	Leeswijzer	5
2	Context	6
2.1	Bedrijfstermen	6
2.2	Canary Releases	6
2.3	Container Orkestratie	7
2.4	Infrastructure as Code	9
2.5	Onderzoeksmethoden	9
2.6	Samenhang	10
3	Aanpak	11
3.1	Metrics analyse	11
3.2	Testopstelling	12
3.3	Adviesrapport	15
4	Resultaten	16
4.1	Metrics analyse	16
4.2	Testopstelling	19
4.3	Adviesrapport	26
5	Conclusie	27
6	Discussie	29
7	Nawoord	32
8	Literatuurlijst	33
9	Bijlages	35
	Bijlage A Beschrijving onderzochte technologieën	36
	Bijlage B Query ontwerpen prototype	38
	Bijlage C Resultaten testopstelling	44
	Bijlage D Adviesrapport	47

1 Inleiding

1.1 Aanleiding

YourSurprise biedt een webplatform aan waar klanten bestellingen kunnen samenstellen en plaatsen. Er wordt constant doorontwikkeld aan dit platform. Als een nieuwe versie in productie gezet wordt, kan het wel eens voorkomen dat een fout onverwachts optreedt. Dit onderzoek probeert een methode te vinden om de potentiële schade hiervan in te perken.

Troisi (2022) geeft aan dat er altijd een risico is dat problemen zich niet laten zien totdat een applicatie aan klanten blootgesteld wordt ongeacht de hoeveelheid testen en de kwaliteit van het testen voordat er naar productie uitgerold wordt.

Ook voor YourSurprise is dit een belangrijk probleem. Het is namelijk als doel opgenomen in hun meerjarenplanning, de Moonshot. Het is als doel opgenomen voor het team waarin dit onderzoek plaats vindt. Dit doel luidt: "We test new webshop releases with a small group of customers to minimize impact on performance and availability."

1.2 Organisatiebeschrijving

Dit onderzoek vindt plaats bij de organisatie YourSurprise. De organisatie heeft als motto "Spread a little happiness!" (YourSurprise, z.d.-a). Dit is ook de missie van het bedrijf. YourSurprise probeert dit te bereiken door klanten de mogelijkheid te bieden anderen te verrassen met gepersonaliseerde cadeaus.

YourSurprise is een commerciële organisatie actief in de detailhandel sector. Het bedrijf specialiseert zich in het maken van cadeaus. Deze cadeaus zijn door klanten te personaliseren via een webportaal. Het bedrijf biedt aan klanten meer dan 1500 verschillende producten om te laten personaliseren (YourSurprise, z.d.-b). Een aantal van deze producten zijn mokken, glazen, kleding en speelgoed.

Het bedrijf heeft een vestiging in Zierikzee. Daar zijn meer dan driehonderd mensen werkzaam (YourSurprise, z.d.-a). Ze zijn actief in allerlei disciplines zoals onder andere E-commerce, logistiek, cadeau-ontwikkeling en ICT. Het pand heeft een gecombineerde ruimte van 15.000 m² voor kantoor, productie, logistiek en recreatie. De organisatie was begin 2020 op weg naar een jaarlijkse omzet van honderd miljoen euro (PRINTmatters, 2020). Alles waarvan YourSurprise denkt dat het zich uniek maakt, wordt binnen de muren van de organisatie gemaakt. Dit is alles van gespecialiseerde software en services tot hardware en logistiek (YourSurprise, intern document, 4 maart 2022; Van Damme et al., 2022).

Al heeft het bedrijf maar één vestiging, is het toch actief in 21 landen (YourSurprise, z.d.-a).

1.3 Afbakening

Voor YourSurprise is het belangrijkste doel van dit onderzoek dat achterhaald wordt welke metrics relevant zijn om de kwaliteit van een nieuwe applicatieversie te bepalen. Om deze reden is het dan ook niet nodig om het prototype perfect binnen de bestaande omgeving te hebben draaien.

Het belangrijkste resultaat voor de organisatie is advies over welke metrics er relevant zijn voor een mogelijke implementatie van canary releases. Dit advies wordt onderbouwd door bevindingen opgedaan in een testomgeving. Deze testomgeving moet een bestaand project van YourSurprise kunnen simuleren, maar hoeft verder niets gelijk te hebben aan het bestaande platform.

Wel wordt hierbij gebruik gemaakt van de ontwikkelingstechnologieën die YourSurprise toepast in het bestaande platform. Hieronder vallen tools als GitLab, om de code van het prototype in op te slaan, en cloud technologieën, om de testomgeving in op te richten.

Omdat implementatiedetails minder relevant zijn voor de organisatie, zal bij dit onderzoek uitgegaan worden van één deployment tegelijkertijd. Wat er moet gebeuren als een nieuwe deployment gestart wordt terwijl de vorige deployment nog bezig is, zal buiten beschouwing gelaten worden.

1.4 Probleem- en doelstelling

Dit onderzoek heeft als hoofdvraag “Hoe kan YourSurprise er zeker van zijn dat de belangrijkste prestatie-indicatoren van een nieuwe versie niet achteruitgaan bij automatische deployment?”

Omdat er verschil zit tussen de (hoeveelheid) data in de testomgeving en in de productieomgeving, kan het zijn dat een versie wel goed werkt op de testomgeving, maar niet op de productieomgeving (J. Walraven, persoonlijke communicatie, 15 februari 2022). Het doel is dus om na een release snel in te zien of er zich een probleem voordoet. Als er zich een probleem voordoet, moet dit zo min mogelijk klanten beïnvloeden.

Een mogelijke oplossing hiervoor is om de nieuwe versie geleidelijk uit te rollen. Deze vorm van geleidelijk uitrollen van een nieuwe versie staat beter bekend als ‘canary releases’. Het idee hierbij is om stapsgewijs een nieuwe versie uit te rollen naar de klanten en te meten of de nieuwe versie voldoet aan de gestelde kwaliteitseisen. Dit onderzoek is opgezet om uit te zoeken welke metrics een betrouwbaar beeld kunnen geven over de kwaliteit van de nieuwe versie.

De exacte implementatie binnen alle bestaande processen en technologieën van YourSurprise valt hier buiten de scope van het onderzoek. De focus voor dit onderzoek ligt op het achterhalen van welke metrics het meest relevant zijn bij het implementeren van canary releases. Om dit te ondersteunen is het nodig een prototype te bouwen die deze stapsgewijze release uit kan voeren op een simulatie van een YourSurprise project.

Om het beantwoorden van de hoofdvraag mogelijk te maken, zijn in overleg met de organisatie twee deelvragen opgesteld.

1.4.1 Deelvraag 1

Voor YourSurprise ligt het grootste belang van dit onderzoek bij het achterhalen van welke metrics relevant zijn voor het bepalen van de kwaliteit van een canary versie. Dit is dan ook de eerste deelvraag: "Welke metrics zijn beschikbaar en relevant om de kwaliteit van een canary versie te bepalen?"

Dit onderzoek moet achterhalen welke metrics relevant zijn voor het bepalen van de applicatiekwaliteit gedurende het releaseproces. Het gaat hier om zowel bestaande als nieuwe metrics. Ieder van deze metrics moet berekend worden vanuit meetwaarden van een applicatie. Ieder van deze metrics is niets anders dan een getal wat makkelijk te vergelijken is. Dit getal zal iets zeggen over de kwaliteit van de canary versie.

1.4.2 Deelvraag 2

Om de eerste deelvraag te onderbouwen is data nodig uit een testomgeving. Voor het opbouwen van deze testomgeving is het nodig bestaande technieken in kaart te brengen en met een van deze technieken een prototype te bouwen.

Om dit mogelijk te maken, is het nodig te weten welke techniek ingezet kan worden om canary releases te beheren. Hiervoor is de tweede deelvraag opgesteld. Deze luidt: "Welke techniek kan ingezet worden om canary releases te beheren?"

Een geschikte techniek is nodig om deze vorm van deployment toe te passen. YourSurprise stelt geen speciale eisen aan de keuze voor een specifieke techniek. Alles wat relevant lijkt, kan onderzocht worden. Omdat het voor de organisatie enkel van belang is dat de testopstelling de adviezen van relevante metrics ondersteunt, valt het integreren hiervan binnen de bestaande infrastructuur van YourSurprise bij dit onderzoek buiten scope. Daarom is een nieuw Google Cloud Platform project beschikbaar gesteld voor dit onderzoek. Voor de organisatie is het bewijzen van de techniek binnen deze omgeving, zonder alle complexiteit van de bestaande infrastructuur, voldoende.

Om deze deelvraag te beantwoorden, is het nodig om uit te zoeken welke technieken inzetbaar zijn, vervolgens deze tegen elkaar af te wegen en ten slotte een van deze technieken in te zetten in de testomgeving.

1.5 Onderzoeksopzet

Dit onderzoek is opgezet aan de hand van drie deelproducten: de metrics analyse, de testopstelling en het adviesrapport.

Het eerste deelproduct is de analyse van metrics. Deze analyse beschrijft hoe bepaald wordt welke metrics betrouwbaar zijn om de kwaliteit van een applicatieversie te bepalen.

Vervolgens wordt een testopstelling opgezet, dit is het tweede deelproduct. Dit product bevat het grootste deel van de Software-engineering competenties. Het bevat namelijk de analyse van technologieën, het afwegen hiervan en het opstellen van een eerste prototype. Binnen dat prototype worden ook de query's die nodig zijn om de relevante metrics te berekenen, ontworpen en geïmplementeerd.

Ten slotte levert dit onderzoek een adviesrapport op, waarin de bevindingen van dit onderzoek voor de organisatie samengevat worden en de deelvragen beantwoord worden.

1.6 Leeswijzer

Dit onderzoeksrapport loopt langs de context van het onderzoek en de aanpak, resultaten en conclusies van ieder van de deelvragen. Dan volgt een conclusie van de hoofdvraag. Het rapport sluit af met een discussie van de resultaten van het onderzoeksproces.

De context is opgesteld als een theoretisch kader, hierin staan termen beschreven die relevant zijn voor dit onderzoek. Hoofdstuk 2 loopt relevante categorieën langs en beschrijft hierin de betekenis van gebruikte terminologie en sluit af met een korte beschrijving van de samenhang van alle benoemde onderwerpen.

Vervolgens is in Hoofdstuk 3 de aanpak opgesteld als een beschrijving van de activiteiten voor dit onderzoek. Hier wordt per deelproduct beschreven wat de aanpak is.

Hierna volgen de resultaten in Hoofdstuk 4 en conclusies in Hoofdstuk 5 van ieder van de deelvragen. Hoofdstuk 5 sluit af met een conclusie op de hoofdvraag van het onderzoek. Een discussie over de resultaten en conclusies van het onderzoek volgt in Hoofdstuk 6.

Het onderzoeksrapport sluit af met een overzicht van gebruikte literatuur in APA-notatie in Hoofdstuk 8.

2 Context

2.1 Bedrijfstermen

YourSurprise hanteert een aantal specifieke termen voor bedrijfsstructuur en de bedrijfsdoelen. Deze termen hebben een overlappend thema, namelijk de ruimtevaart. Zo zijn er (space)ships, dit zijn de teams die verantwoordelijk zijn voor een specifiek bedrijfsproces (YourSurprise, intern document, 15 februari 2022). Dit onderzoek vindt plaats in ship Engineering Productivity. Dit ship heeft veel verschillende verantwoordelijkheden binnen het bedrijf. Het ship is namelijk verantwoordelijk voor het ondersteunen van ontwikkelaars, zodat zij zich enkel met code bezig hoeven te houden. Daarnaast heeft dit ship de verantwoordelijkheid om alle services in de lucht te houden en zo goed mogelijk te onderhouden. Ten slotte denkt dit ship alle infrastructuur uit en voert deze ook door in zowel ontwikkel- als productieomgeving.

Het driejarenplan staat bekend als de Moonshot. De Moonshot bestaat uit vijf pilaren, ook wel bekend als Missions. Deze Missions bevatten meerdere Goals die concreet aangeven waar het bedrijf wil staan in 2025. Deze Goals worden opgesplitst in meerdere Journeys, dit zijn doelen die in ongeveer een half jaar behaald dienen te worden door een bepaald team. Ieder van deze Journeys wordt opgesplitst in kleinere taken, namelijk Liftoffs (N. Slabbekoorn & J. Walraven, persoonlijke communicatie, 22 februari 2022; YourSurprise, intern document, 22 februari 2022).

2.2 Canary Releases

Canary releases is een manier om gedeeltelijk en stapsgewijs een nieuwe versie naar productie uit te rollen. Het functioneert als een waarschuwingssysteem voor een ontwikkelingsproces. Ook biedt het een manier om vroeg en vaak te testen door te verifiëren of de code klaar is voor productie (Moore, z.d.). Binnen dit onderzoek zal de term 'rollouts-technologie' gebruikt worden om technologieën die canary releases mogelijk maken te beschrijven.

Canary releases komen voort uit bedrijven als Facebook, Google en Netflix, waar het van belang is dat teams zelf volledig verantwoordelijkheid kunnen nemen voor hun product (Bryant, 2018). Hiervoor is het volgens Bryant (2018) nodig om een *cloud-native* ontwikkelingsproces op te zetten waar het inzetten van Canary Releases een belangrijk onderdeel is van het deployment proces.

Rapoport (2014) beschrijft canary releases als een proces waar je begint met een nieuwe versie die je stapsgewijs naar productie uitrolt. Dit doe je met controlepunten tussendoor om go/no-go beslissingen te maken.

Vaak wordt ook aangegeven dat canary releases ingezet worden om Service Level Objectives (SLO's) te behalen. Zo beschrijft Beyer et al. (2018) hoe een organisatie door canary releases toe te passen minder snel hun foutbudget verbruiken.

Aan de hand van opgestelde SLO's heb je als organisatie een bepaald foutbudget. Dit budget geeft aan hoeveel problemen er voor mogen komen voordat er consequenties aan zitten. Slechts een deel van het verkeer bloot stellen aan de nieuwe versie waarvan de kwaliteit onbekend is, biedt een organisatie meer ruimte om te manoeuvreren.

Om wat beter uit te leggen wat dit betekent, volgt een voorbeeld. Dit voorbeeld is overgenomen van Beyer et al. (2018).

Stel er is een organisatie die baat heeft bij het aantal foutmeldingen binnen hun applicatie. Een foutpercentage wordt berekend vanuit het aantal foutmeldingen en het totaal aantal verzoeken naar de applicatie. Over dit foutpercentage hebben ze een SLO opgesteld. Als zij een nieuwe versie live zetten waar het foutpercentage fors omhoog schiet, hebben ze een bepaalde tijd om het te corrigeren voordat ze hun SLO overschrijden. Als ze canary releases toepassen en bijvoorbeeld 5% van het verkeer naar de nieuwe versie toesturen, dan duurt het twintig keer zo lang voordat hun SLO overschreden wordt.

2.3 Container Orkestratie

Container orkestratie is een methode om automatisch een omgeving te voorzien van de juiste services. Er zijn vele platformen die dit aanbieden, maar relevant voor dit onderzoek is Kubernetes.

Een platform zoals Kubernetes bestaat uit drie belangrijke onderdelen. Deze onderdelen zijn het beheren van services, het indelen van services over beschikbare infrastructuur en het beheren van de capaciteit van die infrastructuur (Jawarneh et al., 2019).

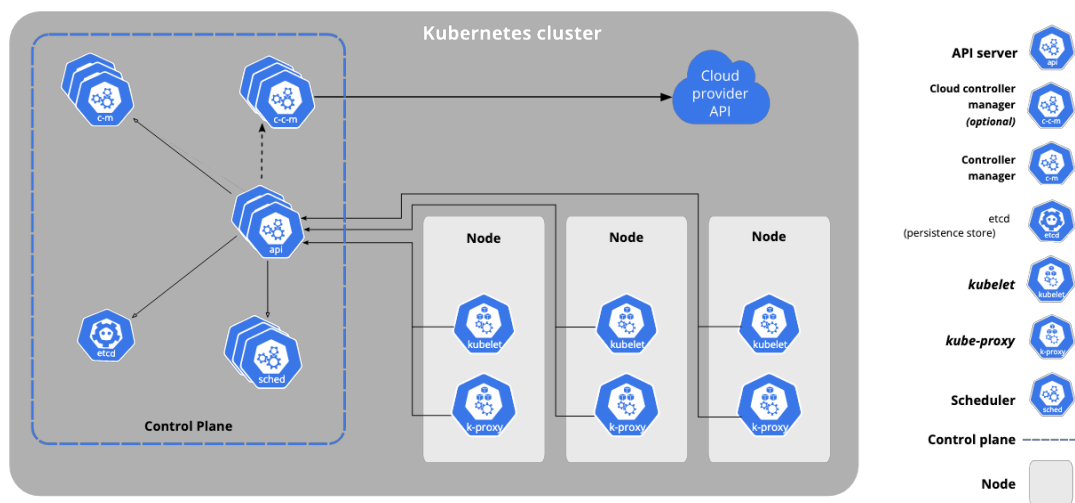
Kubernetes werkt met een master/replica systeem. Hierbij zijn een of meerdere machines binnen een cluster opgezet als control plane. Dit control plane functioneert als gezagvoerder van het cluster. Een gebruiker hoeft enkel een lijst instructies te geven aan een van besturende machines om veranderingen over het hele cluster door te voeren. Een overzicht van deze structuur is opgenomen in Figuur 1.

De volgende vier alinea's zijn grotendeels gebaseerd op informatie verkregen van de Kubernetes documentatie (The Kubernetes Authors, 2021).

Componenten die binnen een Kubernetes cluster draaien worden vaak resources genoemd. De configuratie van de gebruikte resources wordt gedefinieerd in een zogenoemd manifest. Een aantal standaardresources komen specifiek aan bod bij dit onderzoek.

Figuur 1

De componenten van een Kubernetes cluster



Noot. Overgenomen van *Kubernetes Components | Kubernetes*, door The Kubernetes Authors, 2022 (<https://kubernetes.io/docs/concepts/overview/components/>)

De eerste standaardresource is een Pod. Een Pod is het kleinste bouwsteentje van een Kubernetes omgeving. Dit beschrijft eigenlijk één instantie van een applicatie. Hierin staat gedefinieerd welke containers deel uit maken van de applicatie en op welke manieren de applicatie benaderd kan worden.

De volgende standaardresource is een Deployment. Een Deployment beheert onder andere welke Pods er gaan draaien, hoeveel er moeten draaien, of en hoe er automatisch geschaald moet worden en nog vele andere zaken. De Deployment kan ook aangepast worden om een nieuwe versie in productie te zetten. Het beheert dan hoe de oude Pods met de nieuwe Pods vervangen moeten worden gebaseerd op hoe de Deployment geconfigureerd is.

Ten slotte komt een CustomResourceDefinition aan bod. Deze resource biedt de mogelijkheid om nieuwe definities toe te voegen aan het Kubernetes cluster. Deze definities kunnen vervolgens net als een standaardresource gebruikt worden. Veel extensies van Kubernetes, waaronder implementaties van Canary Releases, bieden een aantal CustomResourceDefinitions om hun functionaliteit toe te voegen aan het cluster.

Naast deze standaardresources is een andere term die langs komt in dit onderzoek een zogenoemde ingress controller. Een ingress controller is een technologie die opgebouwd is uit meerdere standaardresources die externe toegang naar services in het cluster mogelijk maakt (The Kubernetes Authors, 2021).

2.4 Infrastructure as Code

Infrastructuur wordt vaak als code gedefinieerd om een aantal volgens Microsoft (2021) belangrijke redenen.

De voornaamste reden is om afwijkingen te voorkomen tussen wat de infrastructuur zou moeten zijn en wat het werkelijk is. Door infrastructuur als code te definiëren is het mogelijk een aanpassing aan de infrastructuur bij de bron aan te passen in plaats van bij het doel waar gebruik van gemaakt wordt.

Daarnaast biedt het ook de mogelijkheid om altijd in dezelfde omgeving te werken, doordat de code aangeeft hoe de omgeving er uit moet zien. Ook wordt aan de hand van deze code, door een automatisch proces, de infrastructuur ingericht zodat altijd dezelfde omgeving opnieuw gebouwd kan worden.

Ten slotte biedt het definiëren van infrastructuur als code de mogelijkheid om een opstelling te toetsen op veelvoorkomende problemen. Er zijn vele tools die een de infrastructuur vanuit de code analyseren om te testen voor veiligheidsproblemen en naleving van standaarden.

Meestal wordt een focus gelegd op infrastructuur in de cloud, maar dit is geen vereiste (Morris, 2015). Bij dit onderzoek ligt de focus wel op infrastructuur in een cloud-omgeving.

Terraform is een methode om infrastructuur via te definiëren die toegepast wordt binnen de organisatie. Dit platform biedt een eigen taal waarin infrastructuur gedefinieerd kan worden. Gebaseerd op deze configuratie kan een uitvoerplan opgesteld, ingezien en uitgevoerd worden (Hashicorp, 2021).

2.5 Onderzoeksmethoden

De elicitering binnen het onderzoek wordt uitgevoerd met een aantal verschillende technieken. Deze technieken komen voornamelijk uit het DOT-framework. Omdat deze technieken als startpunt voor onderzoeksmethodes functioneren, is het de bedoeling dat iedere onderzoeker hier een eigen invulling aan geeft. Om deze reden wordt hier de exacte invulling te benoemd. De beschrijvingen van ieder van deze methodes is gebaseerd op de onderzoeksmethodes zoals Turnhout et al. (2015) ze beschrijven.

Binnen dit onderzoek zijn twee methodes toegepast van Turnhout et al. (2015) waarbij de methodes zonder aanpassing zijn overgenomen. Deze methodes zijn een literatuuronderzoek en interviews.

Een andere methode hier toegepast is de zogenoemde 'Wizard of Oz' onderzoeksmethode. Deze methode is een creatieve weergave om imitaties van een product te visualiseren en tot leven te brengen. Binnen dit onderzoek is voor de initiële elicitatie gebruik gemaakt van deze onderzoeksmethode. Het maken van beslissingen, wat in het eindproduct automatisch zou moeten gebeuren, is hier handmatig overgenomen.

Ten slotte is prototyping toegepast om de analyse van metrics te onderbouwen. Van belang hierbij is om zo snel mogelijk te kunnen achterhalen wat wel en niet werkt en om dwaalsporen binnen het onderzoek te kunnen voorkomen. Zo biedt prototyping een snelle analyse en solide basis voor iteratie.

2.6 Samenhang

Dit onderzoek gaat beschrijven welke metrics het meest relevant en betrouwbaar zijn om canary releases te beoordelen. Om dit te onderbouwen is het nodig om een prototype te bouwen die een release van een YourSurprise project kan simuleren. Dit prototype zal draaien op Kubernetes binnen het Google Cloud Platform ingericht met Infrastructure as Code.

Dit onderzoek wordt uitgevoerd binnen YourSurprise en wordt op de Moonshot gekoppeld aan een deel van de Journey voor het ship Engineering Productivity.

3 Aanpak

Dit hoofdstuk beschrijft de aanpak van de deelproducten van dit onderzoek. Het onderzoek is opgesplitst in drie deelproducten, namelijk de metrics analyse, de testopstelling en een adviesrapport. Ieder van deze deelproducten beantwoordt een gedeelte van de deelvragen.

De metrics analyse beschrijft hoe bij dit onderzoek besloten zal worden hoe betrouwbaar een metric is. Om deze bevindingen te kunnen onderbouwen is een testopstelling nodig. Deze functioneert namelijk als basis om de rest van het onderzoek op uit te kunnen voeren. Het onderzoek sluit af met een adviesrapport waarin alle bevindingen van dit onderzoek opgenomen zijn relevant voor de organisatie.

3.1 Metrics analyse

De metrics analyse is het eerste deelproduct van dit onderzoek. Bij dit deelproduct ligt voor YourSurprise het grootste belang. Dit deelproduct beschrijft het verzamelen van interessante metrics en vervolgens het bepalen van de betrouwbaarheid van deze metrics.

Om deze betrouwbaarheid te bepalen is het eerst nodig om een aantal factoren vast te stellen. De voornaamste twee factoren die de resultaten van dit onderdeel kunnen beïnvloeden zijn het aantal verzoeken dat per seconde opgenomen is in de metrics en de hoeveelheid tijd waarop de metrics gebaseerd zijn. Het aantal verzoeken wat de productieomgeving van YourSurprise buitenom de piekperiodes verwerkt, zal gebruikt worden voor de eerste van deze twee variabelen. Voor de tweede variabele zal een waarde gekozen worden die gebruikt wordt door het huidige monitoring platform tijdens een release.

Vervolgens kan binnen de organisatie achterhaald worden welke metrics gewenst zijn om te onderzoeken. Het doel is om een overzicht te vormen van metrics die gewenst zijn in te zetten bij een implementatie van canary releases. Bij deze analyse worden zoveel mogelijk interne stakeholders betrokken. Dit begint bij ship Engineering Productivity met het in kaart brengen van de redenen waarom dit probleem in de eerste plaats gesignaleerd werd. Vervolgens zullen wensen opgehaald worden bij de teams met een ontwikkelingsexpertise. Hieronder vallen voornamelijk ship Webshop Development en ship Internal Software Systems.

Om bij deze ontwikkelingsteams relevante informatie op te halen, zal het eerst nodig zijn om het onderzoeksdoel bij hen duidelijk te krijgen. Om dit te bereiken zal eerst een visualisatie gemaakt worden van de functionaliteit van een implementatie. Dit zal gedaan worden door een flowchart op te stellen van de stappen die een implementatie zal doorlopen. Daarnaast zal een demo opgezet worden van een implementatie in de stijl van de Wizard of Oz onderzoeksmethode. Dit betekent dat het systeem volledig door de hand gestuurd zal worden om het idee over te brengen zonder een werkende implementatie te hebben.

Al deze wensen worden verzameld en vervolgens vertaald naar een lijstje metrics om te onderzoeken.

Na het ophalen van de te onderzoeken metrics zullen testen uitgevoerd worden om meetwaarden te verzamelen. Deze data moet vervolgens omgezet worden naar de gewenste metrics door ze over een tijdspan te aggregeren. Om te toetsen of een metric afhankelijk is van een kwaliteitsaspect van de applicatie, zullen testen met twee kwalitatief verschillende versies uitgevoerd worden. Met twee versies van gelijke kwaliteit zal getoetst worden wat de hoeveelheid ruis in de metrics is. Deze hoeveelheid ruis zal vervolgens als resultaat gedocumenteerd en gevisualiseerd worden.

3.2 Testopstelling

De implementatie van het prototype volgt een gebruikelijke Software-engineering aanpak. Hier is het namelijk van belang dat eerst een analyse van eisen uitgevoerd wordt. Vervolgens moet bekeken worden welke technologieën beschikbaar zijn. Ten slotte zal een implementatie gebouwd worden die als testopstelling functioneert.

Deze testopstelling moet als basis dienen voor het onderzoek. Het moet een stabiele omgeving bieden waarin de testen van de metrics analyse uit te voeren zijn. Om dit te bereiken zal het uit een testapplicatie bestaan met een aantal andere componenten zoals een database en interne service. Een rollouts technologie zal het uitrollen van deze testapplicatie beheren.

Hoewel in dit onderzoeksrapport de stappen als een lineair proces beschreven staan, zal in werkelijkheid heen en weer gewisseld worden tussen onderzoeksfases. Aan de hand van bevindingen in de implementatiefase zal teruggeschakeld moeten worden naar de analysefase om nieuwe informatie te achterhalen.

3.2.1 Analyse

De analysefase heeft als doel om te achterhalen welke technologie binnen dit prototype het meest geschikt is om canary releases mee te beheren. Dit vormt een deel van het antwoord op de tweede deelvraag. Om dit doel te behalen, zal eerst een lijst mogelijke technologieën opgesteld worden. Vervolgens zal deze lijst tegen elkaar afgewogen worden totdat hier één technologie overblijft om het prototype mee te bouwen.

Eerst is het nodig te achterhalen welke methodes er in de industrie bestaan om binnen de gebruikte technologieën een canary release uit te voeren. Binnen dit onderzoek zal dit gedaan worden door een literatuuronderzoek naar blogs, documentatie en nieuwsberichten over deze technieken in fora die gefocust zijn op Kubernetes en Continuous Integration en Delivery. Ook zal kennis binnen de organisatie gebruikt worden door te spreken met werknemers van ship Engineering Productivity. Het resultaat van dit analysedeel zal een lijst mogelijke technologieën zijn.

Vervolgens zal deze lijst ingekort worden. De technologieën zullen tegen elkaar uitgezet worden en vergeleken worden, gebaseerd op de inzetbaarheid voor dit onderzoek. De technologie heeft enkel als doel om de resultaten van de metrics analyse te ondersteunen.

Deze technologieën zullen afgewogen worden op een aantal factoren. Deze factoren zijn opgenomen in Tabel 1.

De bevindingen van deze analysefase worden gebruikt als onderbouwing van het antwoord op de tweede deelvraag.

3.2.2 Ontwerp

Voor de testopstelling zullen twee ontwerpen gemaakt worden. Het belangrijkste is dat er een ontwerp komt van de query's voor iedere metric. Dit is namelijk de uitwerking van de gewenste metrics naar een implementatie binnen de testopstelling. Daarnaast zal een ontwerp gemaakt worden van de implementatie van de testopstelling, zodat deze testomgeving gereproduceerd kan worden.

Het ontwerp van de query's beschrijft welke meetwaarden nodig zijn en hoe deze gecombineerd en bewerkt moeten worden om tot een score voor de metric te komen. Het resultaat hiervan zijn een aantal ontwerpen die in kaart brengen hoe de metrics geïmplementeerd zullen worden.

Daarnaast is het nodig om de testopstelling in kaart te brengen. Hierbij zullen alle onderdelen die in het Kubernetes cluster terecht komen in een diagram weergegeven worden. Dit diagram zal bestaan uit een aantal onderdelen.

Tabel 1*Overzicht van technologieën*

Eis	Beschrijving
Rollouts tool	Is deze technologie een methode om canary releases te beheren?
Nieuwe metrics	Is het mogelijk om custom query's toe te voegen waarop een canary versie beoordeeld wordt?
Control over stappen	In hoeverre is het mogelijk om het rollouts proces te configureren?
Documentatie	Is er documentatie beschikbaar over het toepassen van deze technologie voor een canary release?
Zekerheid over kwaliteit	Is het mogelijk een slechte versie het proces succesvol te laten doorlopen vanwege een slechte configuratie
Functionaliteit	Hoeveel aanpassingen maakt deze technologie aan bestaande resources in een Kubernetes cluster?

Allereerst zal dit ontwerp de applicatie samen met de rollouts technologie beschrijven. De applicatie zal opgenomen worden in zijn eigen namespace. Binnen de testomgeving beheert de rollouts technologie hoe de testapplicatie draait en waar het binnenkomende verkeer heen gestuurd wordt.

Vervolgens laat dit ontwerp zien welke tools ingezet zijn om verkeer te accepteren en te verwerken. In ieder geval zal gevisualiseerd worden hoe verkeer iedere applicatie bereikt. Ook zal opgenomen worden welke services in het cluster draaien om dit tot stand te laten komen.

Ten slotte zal de monitoring oplossing opgenomen worden in het diagram. De componenten die hiervoor in het cluster draaien zullen gevisualiseerd worden.

3.2.3 Implementatie

De implementatie van de testopstelling zal bestaan uit een testapplicatie en diverse Kubernetes manifestbestanden. De testapplicatie zal een methode bieden om de gewenste prestatie-indicatoren aan te passen. De manifestbestanden configureren alle resources in het Kubernetes cluster die nodig zijn voor de testopstelling.

Deze implementatie begint bij een testapplicatie. Deze testapplicatie moet neergezet kunnen worden door de verschillende rollouts technologieën. De testapplicatie zal een simpele webapplicatie zijn die het mogelijk maakt om de gewenste prestatie-indicatoren aan te passen. Hiermee kan deze testapplicatie tussen twee versies een verbetering of verslechtering van de metrics simuleren. Deze testapplicatie wordt vervolgens uitgerold door de rollouts technologie.

Rondom de testapplicatie en de rollouts technologie zijn een aantal andere componenten nodig. Deze componenten zijn verantwoordelijk voor het accepteren en juist afhandelen van verkeer en het verzamelen van metrics over de applicaties in het cluster. Ook zullen de interne componenten nodig voor de testapplicatie opgezet worden. Deze implementatie zal gelijk zijn aan de architectuur die uitgedacht is bij de ontwerpfase.

Met deze volledige opstelling worden verzoeken geaccepteerd, verwerkt en beantwoord. Over de afhandeling van deze verzoeken wordt data verzameld en opgeslagen. De volgende stap is om deze data te aggregeren zodat er een getal uit gehaald kan worden. Met dit getal wordt vervolgens een beslissing gemaakt of de canary versie voor een bepaalde metric anders scoort dan de stabiele versie. Door de query's uit de ontwerpfase te implementeren is het mogelijk om vanuit data uit de testopstelling te onderbouwen hoe betrouwbaar iedere metric is als indicator van applicatiekwaliteit.

3.3 Adviesrapport

Het laatste deelproduct van dit onderzoek is een adviesrapport. Dit rapport zal naast de resultaten van de vorige twee deelproducten een advies bevatten. Gebaseerd op de bevindingen van dit onderzoek zal een advies opgenomen worden over welke gewenste metrics wel en niet meegenomen kunnen worden bij een implementatie. Daarnaast zal ook de gebruikte testopstelling beschreven worden.

De focus voor dit rapport ligt op het benoemen van de gewenste metrics. Aan de hand van de resultaten van de metrics analyse zal beschreven worden hoe iedere metric ingezet kan worden om de kwaliteit van een nieuwe versie te bepalen.

Daarnaast zal beschreven worden hoe de resultaten verzameld zijn, door een overzicht te geven van de architectuur van de testopstelling.

4 Resultaten

Dit hoofdstuk beschrijft de resultaten van dit onderzoek. Deze resultaten zijn opgesplitst per deelproduct.

4.1 Metrics analyse

De metrics analyse is het deelproduct waar YourSurprise het meeste belang bij heeft. Dit deelproduct beschrijft het proces van het verzamelen van meetwaarden, het omzetten hiervan naar metrics en het bepalen van de betrouwbaarheid van deze metrics.

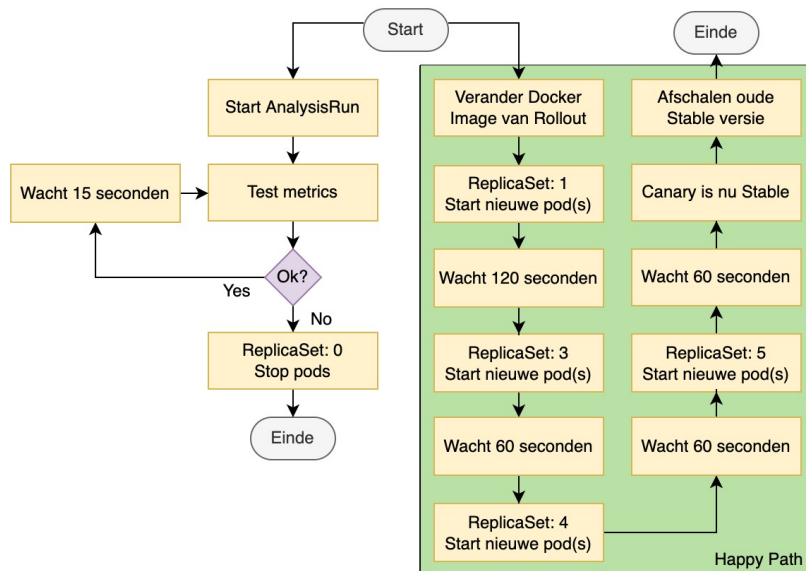
Hiervoor was het eerst nodig om een aantal randvoorwaarden op te stellen. Dit gaat over het aantal verzoeken waar de metrics over meten. Dit is in twee stukken gesplitst. In Instana, het monitoringplatform wat YourSurprise voor de webshop inzet, is de data bekeken van alle verzoeken die binnengekomen zijn over een tijdspan buiten een piekperiode. Deze tijdspan startte net na Valentijnsdag en eindigde exact een maand later. Over deze tijd kwamen er gemiddeld ongeveer vijftig verzoeken per seconde binnen. Dit getal is overgenomen in de metrics analyse. Het andere getal dat relevant is, beschrijft over welke tijdspan de metrics iets zeggen. Deze tijdspan is vastgezet op 1 minuut voor dit onderzoek. Dit getal is overgenomen van het dashboard wat in Instana opgezet is om releases van de webshop van YourSurprise te volgen.

Het idee achter canary releases was nog vrij onbekend bij de stakeholders. Voordat opgehaald kon worden welke metrics interessant waren voor dit onderzoek, moest bij hen duidelijk gemaakt worden wat het doel van dit onderzoek was. Met behulp van de 'Wizard of Oz' onderzoeksmethode is het idee gedeeld bij de IT-demo. Bij deze demo sluiten bijna alle stakeholders van dit onderzoek aan, waardoor het een perfect moment is om het concept achter het onderzoek te verduidelijken. Bij deze demo is laten zien hoe een mogelijke implementatie zou werken door een diagram van de stappen die het zou doorlopen en een demo van een handgestuurd prototype te getoonde diagram is te zien in Figuur 2.

Na het delen van het idee achter dit onderzoek konden de wensen voor de te onderzoeken metrics opgehaald worden. Dit is begonnen bij ship Engineering Productivity, omdat zij met de onderzoeksvraag zijn gekomen. Nadat hier duidelijk was wat de wensen waren, is verder gegaan met het achterhalen van de wensen bij de teams met een ontwikkelingsexpertise. Dit waren ship Webshop Development en ship Internal Software Systems. Ook is gesproken met de E-commerce afdeling om te achterhalen welke baat zij bij dit onderzoek hebben.

Figuur 2

Flowchart canary release



De metrics die boven gekomen zijn vanuit gesprekken met ship Engineering Productivity zijn als volgt:

- De tijd tussen verzoek en antwoord.
- Het aantal verzoeken naar een database.
- De hoeveelheid CPU-tijd nodig per HTTP-verzoek.
- De hoeveelheid werkgeheugen nodig voor de applicatie.
- Het percentage van niet-succesvolle HTTP-statuscodes.

Vanuit ship Webshop Development en ship Internal Software Systems zijn ook een aantal extra metrics naar boven gekomen. Deze metrics zijn meer gefocust op applicatieniveau.

- De hoeveelheid gefaalde server-side renders.
- De totale hoeveelheid aanvragen (in verhouding tot de traffic-split van de canary).
- Het aantal bytes wat verstuurd wordt.
- De belasting van de database.
- Het aantal aanvragen naar interne services (vanuit back-end).
- Het aantal cache hits en misses.
- Het aantal FPM-threads in gebruik.
- Het aantal JavaScript foutmeldingen per browser.
- Het aantal aanvragen naar externe services (vanuit de browser).
- Het aantal database foutmeldingen.

Ook is de E-commerce afdeling gevraagd, maar deze afdeling ziet weinig toegevoegde waarde voor hun afdeling met betrekking tot dit onderzoek. Zij zien voornamelijk meerwaarde in het tegelijkertijd testen van meerdere versies in de vorm van A/B/x-testen. Zij zien geen meerwaarde bij het garanderen van kwaliteitsaspecten bij het uitrollen van een nieuwe versie.

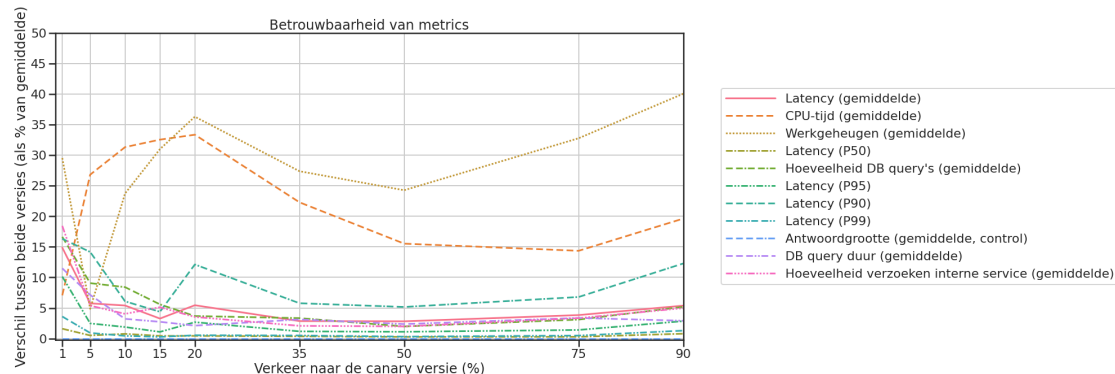
Vervolgens zijn testen uitgevoerd op kwalitatief verschillende applicatieversies. Door een rollout te testen met applicatieversies met verschillende kwaliteitsaspecten is het mogelijk om te toetsen of een metric gebruikt kan worden wanneer er een daadwerkelijk kwaliteitsverschil is. Deze testen hebben een enorme hoeveelheid variabelen en zijn hierom enkel gebruikt om te toetsen of een metric juist geïmplementeerd is. Door te testen met twee versies waarvan de kwaliteitsaspecten gelijk zijn, is in kaart gebracht wat de hoeveelheid ruis en onzekerheid in iedere metric is.

Alle meetwaarden benodigd om de metrics te berekenen, zijn tijdens deze testen opgeslagen. Deze data is vervolgens geaggregeerd. Over een tijdspan van 5 minuten is het gemiddelde verschil tussen de metrics van beide versies gekeken. Vervolgens is dit verschil omgezet naar een verhouding vergeleken met het gemiddelde van de metrics.

Van deze resultaten is een visualisatie gemaakt. Deze visualisatie is opgenomen in Figuur 3. De details over het meten van de data, het verwerken van de gemeten data naar de geaggregeerde getallen en het opstellen van de visualisatie staan genoteerd in Bijlage D.

Figuur 3

Afwijking metrics per canary split percentage (0%-50%)



4.2 Testopstelling

Het tweede deelproduct van dit onderzoek is een testopstelling. Deze opstelling zorgt voor een consistente omgeving om de metrics analyse in uit te voeren.

Dit begint met een analyse naar manieren om canary releases te beheren. Daarna volgt een ontwerp van de query's die geïmplementeerd moeten worden en een overzicht van de architectuur van de testopstelling. Dit deelproduct sluit af met de implementatie van de ontworpen architectuur.

4.2.1 Analyse

De analysefase heeft als doel om te achterhalen welke technologie binnen dit prototype het meest geschikt is om canary releases mee te beheren. Dit begint met het opstellen van een lijst mogelijk inzetbare technologieën vanuit bestaande kennis in de organisatie en een literatuuronderzoek. Vervolgens moet een keuze gemaakt worden binnen deze opties naar een technologie om de testopstelling mee te bouwen.

Vanuit het ship Engineering Productivity kwam een technologie van Netflix naar voren, namelijk Spinnaker in combinatie met Kayenta. Ook zijn twee opties bekeken die aangeraden worden door de organisaties achter technologieën die YourSurprise nu al inzet in de digitale omgeving. In dit kader is binnen Google Cloud Platform een technologie bekeken, namelijk GCP Operations. Ook is gekeken naar de canary release mogelijkheden binnen GitLab.

Om zo veel mogelijk opties te bekijken, is verder gezocht op fora over Kubernetes en Continuous Integration en Delivery. Hierbij werden geen resultaten gevonden totdat expliciet gezocht werd op 'Automated Canary Analysis'. Hieruit zijn ook een aantal opties gerold. Deze opties zijn Argo Rollouts van Argoproj, Flagger van Flux en Iter8 voornamelijk gesponsord door IBM.

Nadat dit rijtje mogelijke technologieën opgesteld was, is achterhaald wat ieder van deze technologieën aanbiedt. Een overzicht van de resultaten hiervan is opgenomen in Tabel 2. De volledige beschrijving van de technologieën is opgenomen in Bijlage A.

Tabel 2

Overzicht van technologieën

Technologie	Type	Beschrijving
Spinnaker	CI/CD	Het is een Continuous Delivery platform. Het kan canary releases aansturen gebaseerd op metrics.
GCP Operations	CI/CD	Het is een Google Cloud hosted versie van Spinnaker.
GitLab	CI/CD	Het is een Continuous Delivery platform. Het kan canary releases blindelings aansturen. Het kan dashboards bieden voor handmatige interventie.
Argo Rollouts	Kubernetes canary controller	Het is een controller met bijbehorende CRD's. Het beheert een rollout vanuit het cluster.
Flagger	Kubernetes canary controller	Het is een controller met bijbehorende CRD's. Het beheert een rollout vanuit het cluster.
Iter8	Load generator	De recente versie is een load-generator. Het valideert SLO's in een aparte omgeving.

Voor dit onderzoek heeft het geen meerwaarde om een volledig Continuous Delivery platform op te zetten. YourSurprise maakt hiervoor al gebruik van GitLab dat ook bij dit onderzoek gebruikt kan worden. Iter8 is ook afgefallen omdat dit enkel een generator van verkeer bleek te zijn. Het afwegen van technologieën is voortgezet met de overgebleven opties: Argo Rollouts en Flagger.

Deze twee opties voldoen aan het eerste criterium. Dit criterium eist namelijk dat de technologie de mogelijkheid biedt om canary releases te beheren. De twee opties zijn tegen elkaar uitgezet door middel van een proof of concepten en hierop vervolgens beoordeeld. De resultaten hiervan zijn opgenomen in Tabel 3.

Tabel 3

Afweging van overgebleven technologieën

	Argo Rollouts	Flagger
Nieuwe metrics	Custom query's mogelijk via AnalysisTemplate	Custom query's mogelijk via MetricTemplate
Controle over stappen	Tijd tussen stappen, gewenste hoeveelheid verkeer naar iedere versie en gewenste hoeveelheid instanties van de applicatie volledig te beheren gedurende het gehele proces	Tijd tussen stappen, gewenste hoeveelheid verkeer naar iedere versie en gewenste hoeveelheid instanties van de applicatie globaal te beheren. Waardes zijn constant voor het gehele proces
Documentatie	Documentatie gericht op canary analyse vergeleken met de stabiele versie	Documentatie gericht op SLO-check
Zekerheid over kwaliteit	AnalysisRun en Rollout staan los; het is mogelijk een versie als stabiel te verklaren voordat een analyse afgerond is	De rollout wacht op in ieder geval één succesvolle analyse
Functionaliteit	Voegt een nieuw CRD toe die een Deployment vervangt	Voegt een nieuw CRD toe die een Service vervangt

De deelconclusie van deze analysefase beschrijft de gekozen technologie. Bij dit onderzoek is voor de implementatie gebruik gemaakt van Argo Rollouts.

Deze keuze is gebaseerd op het feit dat Argo Rollouts de mogelijkheid biedt exact te controleren hoe de rollout plaats zal vinden. Flagger biedt meer zekerheid over de kwaliteit door enkel de rollout door te zetten als de canary in ieder geval de eerste ronde checks slaagt. Al is deze zekerheid een van de belangrijkste factoren voor een productie-omgeving, voor het prototype n dit onderzoek is dat veel minder relevant dan het hebben van controle over het proces. Ook sluit de documentatie van Argo Rollouts beter aan bij canary analyse, de focus van dit onderzoek, terwijl Flagger meer gericht is op het controleren van SLO's op een nieuwe versie.

4.2.2 Ontwerp

Voor de testopstelling zijn twee ontwerpen gemaakt. Het eerste resultaat is een ontwerp van de query's. Hierin staat een overzicht van de manier waarop de data die verzameld wordt in de testopstelling omgezet wordt naar een getal dat wat zegt over de gewenste kwaliteitsaspecten. Het tweede resultaat is een overzicht van de architectuur van de testopstelling. Dit is een overzicht van de componenten die ingezet zijn in het Kubernetes cluster en de manier waarop die gestructureerd zijn.

Om de query's te kunnen ontwerpen, zijn eerst de beschikbare meetpunten in kaart gebracht. De query's zijn zo opgesteld dat ze iets zeggen over de afgelopen tijd. Op het moment dat de query uitgevoerd wordt haalt deze de meetpunten op van de afgelopen minuut om de berekening op uit te voeren. Vervolgens rolt hier een getal uit als indicatie voor dat kwaliteitsaspect van de canary versie van de applicatie.

De volgende query's zijn opgenomen in deze ontwerpfase:

- De gemiddelde tijdsduur per HTTP-verzoek van de laatste minuut
- Het foutpercentage van HTTP-verzoeken van de laatste minuut
- De gemiddelde grootte van het antwoord per HTTP-verzoek van de laatste minuut
- De gemiddelde hoeveelheid aanvragen naar de interne service per HTTP-verzoek van de laatste minuut
- De gemiddelde hoeveelheid database query's per HTTP-verzoek van de laatste minuut
- De gemiddelde tijdsduur per database query van de laatste minuut
- Het foutpercentage van database query's van de laatste minuut

De gerealiseerde ontwerpen van deze query's zijn te zien in Bijlage B.

Nadat bekend was welke technologieën ingezet zouden worden binnen de testopstelling, was het van belang om de beoogde opstelling alvast uit te denken. Niet alleen vanwege de reproduceerbaarheid van de opstelling, maar ook om eventuele conflicten in de opstelling alvast af te vangen.

Het ontwerp bestaat uit een visualisatie van alle Kubernetes resources die in de testopstelling ingezet worden. Ieder stukje functionaliteit staat in zijn eigen namespace.

Het verkeer komt binnen op het cluster bij de ingress gateway van Istio. Istio zoekt tussen alle gateways naar de juiste bestemming gebaseerd op een aantal regels. De enige regel die in het prototype gebruikt is, is het domein waarop verkeer binnen komt.

Als verkeer bestemd is voor de testapplicatie, dan gaat het naar de 'program' namespace. De gateway in deze namespace weet af van de service die vervolgens weer af weet van alle instanties van de applicatie. De service is verantwoordelijk voor het verdelen van het verkeer over alle beschikbare instanties gebaseerd op de gewenste hoeveelheid verkeer naar iedere versie gedurende een canary release proces.

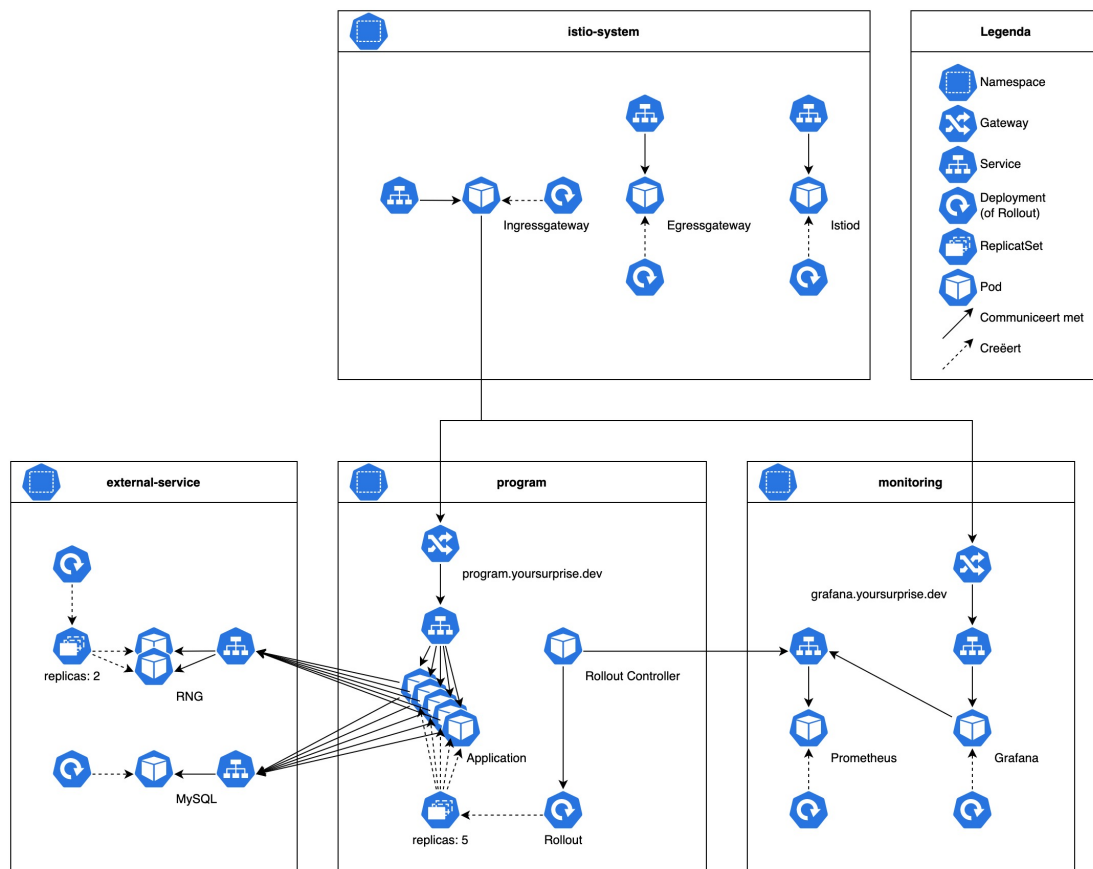
De andere bestemming die opengezet is op de testopstelling is Grafana. Binnen Grafana is het mogelijk om query's op te stellen en de resultaten daarvan te visualiseren. De data wordt opgehaald uit een gekoppelde Prometheus instantie.

Interne communicatie is ook gevisualiseerd in het diagram. Hieronder valt bijvoorbeeld de rollout controller die gebaseerd op data die in Prometheus opgeslagen is beslissingen moet nemen over de canary release. Ook valt hier het verkeer onder wat de applicatie naar de database of interne service (RNG) stuurt.

Interne communicatie met Prometheus is om duidelijkheid binnen het diagram van Figuur 4 te behouden niet opgenomen, sinds Prometheus alle pods uitleest om meetwaarden op te halen.

Figuur 4

Diagram testopstelling



4.2.3 Implementatie

De testopstelling bestaat uit een aantal losse onderdelen. Eerst bestaat het uit een applicatie die neergezet kan worden door de verschillende rollouts technologieën. Daarnaast biedt het een methode om binnen een Kubernetes cluster verkeer te accepteren en te verwerken. Ten slotte verzamelt deze opstelling meetwaarden. De meetwaarden worden door query's geaggregeerd naar metrics. Deze metrics zijn ook in te zien.

Tijdens het onderzoek naar technologieën om een canary release te beheren zijn ook alle bijbehorende voorbeeld testapplicaties van de in paragraaf 4.2.1 genoemde technologieën gebruikt. Ook Argo Rollouts biedt een kleine applicatie aan om hun canary release technologie te demonstreren. Deze applicatie bevat al een hoop functionaliteit die nodig is voor dit onderzoek. Deze demoapplicatie is als basis gebruikt voor de testapplicatie van deze testopstelling.

De testapplicatie van Argoproj moet gestart worden met een instelling voor een kleur als indicatie van versie en biedt dan een webpagina aan. De browser van een bezoeker van deze webpagina begint dan verzoeken naar de achterliggende API te sturen. Deze API antwoordt met de vooraf ingestelde kleur, maar kijkt naar de instellingen die vanuit de browser ingesteld zijn om eventueel een 500 HTTP-statuscode terug te geven of het antwoord te vertragen. Verzoeken worden samen met hun status en versie gevisualiseerd op de webpagina.

Naast deze testapplicatie was het nodig om een aantal extra componenten op te zetten. Deze componenten zijn verantwoordelijk voor het accepteren en juist afhandelen van verkeer en het verzamelen van meetwaarden over de applicaties in het cluster. Ook bestaat dit uit een database en een interne service om hier meetwaarden over op te halen.

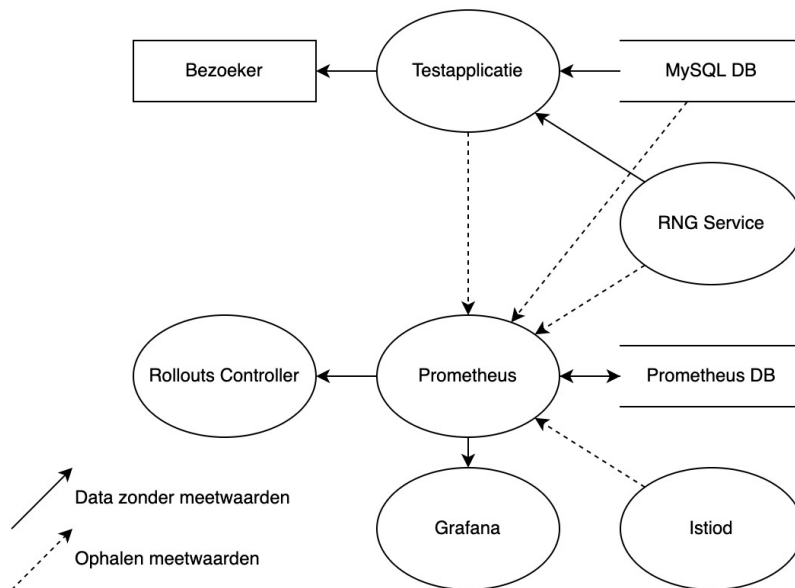
Voor de eerste twee van deze extra componenten is gekeken naar compatibiliteit met de rollouts technologie. Voor het accepteren en afhandelen van verkeer is gekozen voor Istio. Hier is gebruik gemaakt van hun 'demo'-profiel. De Kiali extensie is hieraan toegevoegd om de testopstelling te kunnen visualiseren. Prometheus is ingezet om alle meetwaarden te verzamelen en een mogelijkheid te bieden deze te aggregeren naar metrics. Hier is Grafana aan toegevoegd voor het handmatig uitvoeren en visualiseren van query's.

Om meetwaarden te kunnen verzamelen over interne services, zijn zowel een database als interne applicatie neergezet bij de testopstelling. De database bevat een tabel waarop 'select' query's uitgevoerd worden. De interne applicatie genereert voor ieder verzoek een random getal en geeft dit terug als antwoord.

Een aantal screenshots van deze testopstelling zijn opgenomen in Bijlage C. Een dataflow diagram van deze opstelling is opgenomen in Figuur 5. Al het verkeer loopt in de werkelijkheid via Istio als een soort proxy. Deze stroom is niet opgenomen in deze figuur, omdat dan niet meer te zien is wat de werkelijke stroom van informatie is.

Figuur 5

Dataflow testopstelling



4.3 Adviesrapport

De onderzoeksoepzet en de onderzoeksresultaten zijn verzameld voor de organisatie en besproken met ship Engineering Productivity. Omdat zij de originele onderzoeksvraag gesteld hebben, is samen met hen de inhoud van het adviesrapport bepaald.

Het adviesrapport beschrijft twee verschillende categorieën metrics. Er is een tweedeling gemaakt tussen metrics die niet of nauwelijks afhankelijk zijn van ruis en metrics die wel sterk door ruis of externe factoren beïnvloed worden.

Vanuit deze tweedeling in metrics is vervolgens een advies uitgebracht over de hoofdvraag van het onderzoek ("Hoe kan YourSurprise er zeker van zijn dat de belangrijkste prestatie-indicatoren van een nieuwe versie niet achteruitgaan bij automatische deployment?"). Advies aan YourSurprise is dan ook dat dynamisch grenzen gebruikt kunnen worden voor de stabiele metrics, maar slechts de meest extreme kwaliteitsafwijkingen afgevangen kunnen worden met de metrics die door veel ruis beïnvloed worden.

Het opgestelde adviesrapport is opgenomen in dit onderzoeksrapport in Bijlage D.

5 Conclusie

De hoofdvraag van dit onderzoek is opgesteld als volgt: “Hoe kan YourSurprise er zeker van zijn dat de belangrijkste prestatie-indicatoren van een nieuwe versie niet achteruitgaan bij automatische deployment?”. De hoofdvraag is opgedeeld in twee deelvragen die eerst worden beantwoord.

De eerste deelvraag luidt: “Welke metrics zijn beschikbaar en relevant om de kwaliteit van een canary versie te bepalen?”. De te onderzoeken metrics zijn opgehaald door middel van meerdere elicitatiemethodes. Deze metrics zijn vervolgens in de testopstelling geïmplementeerd en hierin getoetst op betrouwbaarheid om de kwaliteit van een nieuwe versie te bepalen. De bevindingen zijn verwerkt naar een adviesrapport dat vervolgens gedeeld is met de organisatie.

In dit adviesrapport is advies uitgebracht vanuit de onderzoeksresultaten. Er worden twee categorieën metrics geïdentificeerd. Vervolgens wordt geadviseerd hoe ieder van deze categorieën in te zetten is bij een implementatie van canary releases. Ook staan overige wensen genoteerd die buiten de scope van dit onderzoek vielen, maar wel bij de elicitatiefase gevonden zijn.

De tweede deelvraag luidt: “Welke techniek kan ingezet worden om canary releases te beheren?”. Er is onderzoek gedaan naar verschillende technologieën die de mogelijkheid bieden om canary releases te beheren. Vele opties zijn onderzocht, maar bijna alle opties zijn bij nader onderzoek weggevallen doordat de technologie niet doet wat hunt marketing belooft. Daardoor zijn slechts Flagger en Argo Rollouts vergeleken op een aantal factoren die relevant zijn voor het opzetten van een testopstelling voor dit onderzoek. Er waren geen specifieke eisen gesteld vanuit YourSurprise. Een implementatie is, vanwege betere mogelijkheden tot controle van het proces, opgezet met Argo Rollouts.

Een antwoord op de hoofdvraag kan nu geformuleerd worden vanuit de deelconclusies.

Er is gekeken naar de mogelijkheden om bij automatische deployment, met een klein deel van de gebruikers, de kwaliteit van een nieuwe release te valideren. Hiervoor zijn gewenste prestatie-indicatoren in de vorm van metrics in kaart gebracht. Vervolgens zijn deze in een testomgeving getest. YourSurprise heeft in haar huidige situatie een schaalgrootte waarop vele gewenste metrics sterk beïnvloed worden door ruis of externe factoren.

De gevonden stabiele metrics kunnen door YourSurprise ingezet worden om met zekerheid de kwaliteit van een nieuwe applicatieversie te bepalen. De metrics die sterk door ruis of externe factoren beïnvloed worden, kunnen enkel ingezet worden om enorme kwaliteitsveranderingen te detecteren.

Door canary releases in te zetten voor nieuwe releases van de webshop kan YourSurprise voor een aantal stabiele metrics verzekeren dat deze metrics niet achteruit gaan. Voor overige kwaliteitsindicatoren kan vanwege de schaalgrootte van YourSurprise slechts op een buitengewoon afwijkende waarde getoetst worden.

6 Discussie

6.1 Metrics

De factor die de resultaten en de conclusies van dit onderzoek veruit het meeste beïnvloedt, is de waarde van metrics waarop getest wordt. Bijvoorbeeld, een latency die 10 milliseconden varieert rond een waarde van 35 milliseconden (29% schommeling) ervaart relatief meer ruis dan diezelfde metric rond een waarde van 170 milliseconden (6% schommeling).

Voor dit onderzoek zijn de waardes van deze metrics zo dicht mogelijk overgenomen vanuit de getallen die gerapporteerd worden door Instana, het monitoring platform dat YourSurprise in de productieomgeving inzet. De gebruikte waarden voor metrics die in de testopstelling in te stellen zijn, zijn opgenomen in Tabel 4.

Tabel 4

Overzicht van instellingen metrics

Variabele	Gebruikte waarde	Notities
Aantal verzoeken	50 verzoeken per seconde	De webshop ontvangt gemiddeld 45,5 verzoeken per seconde.
Duur van verzoeken	Gemiddeld: 50 ms 90e percentiel: 120 ms	Dit zijn de latencies van de webshop zoals geaggregeerd door Instana.
Database verzoeken per HTTP-request	1-2 per HTTP-verzoek	De webshop stuurt gemiddeld ongeveer 25 database verzoeken per HTTP-verzoek. De testopstelling kan deze hoeveelheid data niet aan, dus is deze hoeveelheid verlaagd naar een hoeveelheid die het wel aan kan.
Foutpercentage HTTP-verzoeken	0,01% van HTTP-verzoeken	Dit is de hoeveelheid 500 status-codes die de klanten te zien krijgen.

Het foutpercentage van HTTP-verzoeken is vanwege de lage hoeveelheid in de productieomgeving van de YourSurprise webshop achterwege gelaten. Als het percentage van HTTP-verzoeken wat fout gaat binnen de testopstelling ingesteld zou zijn aan dat de productieomgeving van de webshop ervaart, zou er gemiddeld vier minuten tussen iedere foutmelding zitten. In een uur tijd, zou dit betekenen dat slechts 15 foutmeldingen gerapporteerd worden. Het is eigenlijk gegarandeerd dat de kwaliteit van de nieuwe applicatieversie volledig bepaald zal worden door de toevallige verdeling van fouten over beide versies. Om deze reden is deze metric niet verder meegenomen bij het uitvoeren van de testen.

6.2 Stakeholders E-commerce

Tijdens het onderzoek is ook met de E-commerce afdeling besproken of zij baat zouden hebben bij dit onderzoek. Het eerste wat bij het gesprek gezegd werd, was: "Eindelijk wordt er iets gedaan om te zorgen dat [IT] niets sloopt". Bij doorvragen kwam naar voren dat ze baat hebben bij een aantal metrics. Hieronder vallen zaken als de click-through rate (hoeveel mensen doorklikken als deel van de totale hoeveelheid mensen die de content zien) en de Net Promoter Score (de metric die YourSurprise toepast om klanttevredenheid te bepalen).

Vanuit de literatuur wordt aangegeven dat metrics gebaseerd op dit soort zwakke signalen juist niet te gebruiken zijn (Rapoport, 2014). Ook wordt aangegeven dat metrics die niet direct te linken zijn aan de te testen wijzigingen of afhankelijk zijn van externe factoren niet meegenomen moeten worden (Beyer et al., 2018).

Ook zijn deze metrics bijna niet te integreren in een testopstelling die niet beschikbaar is voor klanten. Daarbij geeft de literatuur aan dat ze niet bruikbaar zijn voor canary analyse. Daarom zijn deze metrics niet gebruikt bij dit onderzoek en niet meegenomen in de testopstelling.

6.3 Public cloud

Er is binnen het onderzoek tweemaal een Kubernetes cluster ingericht voor de testopstelling. Met slechts twee clusters is er geen enkele garantie dat het geen outliers in Google Cloud zijn. De kwaliteit van de nodes van het cluster beïnvloedt de resultaten van het onderzoek.

De vraag is of het testen met een outlier een probleem is voor dit onderzoek. Het kernpunt van de metrics analyse is om metrics te vinden die niet afhankelijk zijn van de toevallige kwaliteit van de cloud omgeving. Daarom is het belangrijk dat niet een (te) stabiele omgeving gebruikt wordt. Door slechts twee clusters te gebruiken voor het testen, is het waarschijnlijk niet gelukt om de meest instabiele situatie te gebruiken bij het testen. Het zou dus zo kunnen zijn dat metrics die dit onderzoek stabiel verklaard heeft, toch niet stabiel zijn in bepaalde situaties van de public cloud.

6.4 Toegevoegde waarde

Binnen ship Engineering Productivity, waar dit onderzoek uitgevoerd is, worden veel nieuwe technieken en methodes uitgetoetst. Het doel van deze experimenten is om de kwaliteit van de productieomgeving te verhogen, de ontwikkelomgeving te verbeteren en de kosten te verminderen.

Dit onderzoek is een van die experimenten om de kwaliteit van de productieomgeving te verbeteren. Het inzetten van canary releases was een kans die nog niet eerder onderzocht was. Nu is bekend wat deze techniek voor meerwaarde kan bieden bij YourSurprise en wat de aandachtspunten zijn als in de toekomst een implementatie gepland wordt.

6.5 Vervolgonderzoek

Het adviesrapport beschrijft welke metrics op welke manier inzetbaar zijn om de kwaliteit van een nieuwe applicatieversie te bepalen. Als na dit onderzoek de wens bij YourSurprise ligt om canary releases op deze manier in te zetten in de productieomgeving van de webshop, dan zijn er genoeg mogelijkheden voor vervolgonderzoek.

De belangrijkste stap in het vervolgonderzoek is dan om een implementatieplan op te stellen. Dit plan zal in ieder geval bestaan uit een onderzoek naar productiewaardige technologieën, een migratieplan om van de huidige deployment strategie over te stappen naar canary releases en een implementatie van de in dit onderzoek geteste metrics binnen de productieomgeving.

7 Nawoord

Dit onderzoek ben ik begonnen als experiment om zelf te kunnen proeven hoe het is om in een DevOps-team te werken. Zodoende heb ik samen met mijn bedrijfsbegeleider een onderzoeksdoel uit kunnen denken wat niet alleen past bij mijn afstudeerrichting en relevant was voor de organisatie, maar ook een doel wat interessant en uitdagend was voor mij. Nu ik aan het einde van dit onderzoek sta, mag ik zeggen dat dit een geslaagd experiment is. Over de afgelopen maanden heb ik enorm veel plezier gehad in de werkzaamheden van mijn opdracht, maar ook de kleine werkzaamheden daarbuiten die ik geproefd heb.

Met het onderzoek nu afgerond achter me, wil ik de tijd nemen om mijn bedrijfsbegeleider, Jan Walraven, te bedanken voor zijn interesse, voor het meedenken en voor het ondersteunen. Daarnaast wil ik hem natuurlijk ook bedanken voor het zijn van een toffe en inspirerende collega.

Ook wil ik binnen YourSurprise Arne Timmerman bedanken voor het bieden van een kans om in een tak van de IT te stappen waar ik nog geen professionele ervaring in had.

Uiteraard wil ik ook mijn docentbegeleiders, Elio Tolhoek en Frans Blauw, hartelijk bedanken voor alle ondersteuning in het doorlopen van deze fase van mijn opleiding.

Ten slotte wil ik graag iedereen bij YourSurprise heer hartelijk bedanken, met name ook de IT-afdeling, voor al het enthousiasme en betrokkenheid bij mijn onderzoek.

Daniel de Cloet

19 juni 2022

PS Uiteraard ook Leopoldo Castillo bedankt voor alle heerlijke lunches!

8 Literatuurlijst

- Beyer, B., Murphy, N. R., Rensin, D. K. & Kawahara, K. (2018). Canarying Releases [E-book]. In S. Thorne (Red.), *The Site Reliability Workbook* (p. 22). O'Reilly. Verkregen 4 mei 2022, van <https://sre.google/workbook/table-of-contents/>
- Bryant, D. (2018, september 20). *A Comprehensive Guide to Canary Releases*. Verkregen 9 juni 2022, van <https://blog.getambassador.io/cloud-native-patterns-canary-release-1cb8f82d371a>
- Hashicorp. (2021, december 15). *What is Terraform*. Verkregen 19 januari 2022, van <https://www.terraform.io/intro>
- Jawarneh, I. M. A., Bellavista, P., Bosi, F., Foschini, L., Martuscelli, G., Montanari, R. & Palopoli, A. (2019). Container Orchestration Engines: A Thorough Functional and Performance Comparison. *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. <https://doi.org/10.1109/icc.2019.8762053>
- Microsoft. (2021, juni 29). *What is Infrastructure as Code? - Azure DevOps*. Verkregen 4 mei 2022, van <https://docs.microsoft.com/en-us/devops/deliver/what-is-infrastructure-as-code>
- Moore, M. (z.d.). *Automate Canary Testing for Continuous Quality*. Verkregen 5 april 2022, van <https://www.blazemeter.com/shiftright/automate-canary-testing-continuous-quality>
- Morris, K. (2015, maart 2). *What is Infrastructure as Code?* Verkregen 19 februari 2022, van <https://infrastructure-as-code.com/book/2015/03/02/what-is-infrastructure-as-code.html>
- PRINTmatters. (2020, februari 12). *Your Surprise: met gepersonaliseerde cadeaus op weg naar honderd miljoen euro omzet*. Verkregen 29 oktober 2021, van <https://printmatters.nl/nieuws/2020/02/12/yoursurprise-op-weg-naar-honderd-miljoen-euro-omzet>
- Rapoport, R. (2014, augustus 28). *Canary Analyze All The Things: How We Learned to Keep Calm and Release Often* [Conferentiepresentatie] [QCon 2014, New York City, Verenigde Staten van Amerika]. <https://www.infoq.com/presentations/canary-analysis-deployment-pattern/>
- The Kubernetes Authors. (2021, juni 16). *Glossary | Kubernetes*. Verkregen 4 mei 2022, van <https://kubernetes.io/docs/reference/glossary/?core-object=true&extension=true>
- The Kubernetes Authors. (2022, april 30). *Kubernetes Components | Kubernetes*. Verkregen 31 mei 2022, van <https://kubernetes.io/docs/concepts/overview/components/>
- Troisi, H. (2022, februari 11). *Automate Canary Analysis on Kubernetes with Argo*. Verkregen 21 maart 2022, van <https://blog.px.dev/argo-rollouts/>

- Turnhout, K., Van Jacobs, M., Kamp, I., Mulholland, C., Neuman, A., Rouwhorst, S. & Van Vlies, L. (2015). *CMD Methods Pack: Find a combination of research methods that suit your needs*. Verkregen 9 mei 2022, van <https://cmdmethods.nl/>
- Van Damme, B., Van Bale, J. & Meliefste, K. (2022, maart 16). *How we improved our internal logistics by developing a Put To Light cabinet* [Conferentiepresentatie] [YourSurprise Meetup, Zierikzee, Nederland].
- YourSurprise. (z.d.-a). *De missie, voordelen en bedrijfscultuur van YourSurprise*. Verkregen 29 oktober 2021, van <https://nl.indeed.com/cmp/Yoursurprise/about>
- YourSurprise. (z.d.-b). *Home [LinkedIn page]*. Verkregen 29 oktober 2021, van <https://nl.linkedin.com/company/yoursurprise-com>

9 Bijlages

Bijlage A Beschrijving onderzochte technologieën

In deze bijlage staan de beschrijvingen uitgeschreven van de onderzochte technologieën bij de analysefase van de testopstelling. Deze technologieën zijn alle mogelijkheden die bekeken zijn om canary releases mee te beheren.

Spinnaker

Spinnaker is een Continuous Delivery platform die canary releases kan beheren door zogenoemde retriever en judge te gebruiken. De retriever is hier meestal Kayenta en is verantwoordelijk voor het ophalen van meetwaarden en het opslaan in een tijdserie-database. De judge is verantwoordelijk voor het omzetten van deze waarden naar metrics en vervolgens het beoordelen hiervan. Dit bestaat uit het valideren van data, het opschonen van data, het aggregeren van data, het vergelijken van metrics tussen de verschillende versies en ten slotte het berekenen van een score voor de canary versie.

GCP Operations

GCP Operations is een technologie die volledig in Google Cloud Platform kan draaien. Deze tool biedt logging en monitoring. Voor het beheer van canary releases valt deze tool terug op een versie van Spinnaker die door Google beheerd wordt.

GitLab

GitLab biedt een manier om op basis van labels in een Kubernetes cluster de huidige staat van een release te visualiseren. Het gebruikt een selector om te zien of het label 'track' de waarde 'canary' heeft en kan op die manier visualiseren hoe de canary release er voor staat. Het beheer van het canary release proces en de dynamische traffic split zijn verouderde functionaliteiten.

Argo Rollouts

Argo Rollouts is niets anders dan een hoop Custom Resource Definitions (CRD's) die een aantal resource definiëren en een controller die het proces beheert. Er is een plug-in voor 'kubect!' beschikbaar om testen makkelijker te maken. Om Argo Rollouts in te stellen biedt het een 'Rollout' resource aan die de standaard Kubernetes 'Deployment' uitbreidt. In deze resource zijn de verschillende release strategieën in te stellen.

Flagger

Flagger is net zoals Argo Rollouts een set CRD's met een bijbehorende controller. Flagger bevat een 'Canary' resource die in te stellen is, de controller gebruikt deze resource om aangewezen 'Deployment' resources te beheren.

Iter8

Iter8 leek op het eerste gezicht een tool die de mogelijkheid biedt om SLO's te valideren door een nieuwe versie bloot te stellen aan een gedeelte van de gebruikers. Ook leek het integratie met Istio te hebben om de verdeling van het binnenkomen van verkeer te beheren. Dit leek zo te zijn omdat deze applicatie dit tot versie 0.7 doet. Deze versie is niet meer ondersteund sinds november 2021. Versie 0.8 en daarna is niets anders dan een verkeersgenerator. Deze tool creëert nu artificieel verkeer om SLO's te valideren. Iter8 leek bij het onderzoek naar mogelijke tools het juiste soort tool te zijn, maar blijkt bij nadere inspectie niet meer te doen wat voor dit onderzoek relevant is.

Bijlage B Query ontwerpen prototype

In deze bijlage staan de figuren van het ontwerp van de query's die gebruikt zijn bij het prototype. Deze query's combineren de meetwaarden uit de tijdseries database zodat een getal er uit rolt wat iets zegt over een kwaliteitsaspect van de applicatie. Als de meetwaarden op deze manier geaggregeerd is, dan is het een metric.

In deze bijlage zijn de volgende figuren opgenomen:

- De gemiddelde tijdsduur per HTTP-verzoek van de laatste minuut in Figuur 6
- Het foutpercentage van HTTP-verzoeken van de laatste minuut in Figuur 7
- De gemiddelde grootte van het antwoord per HTTP-verzoek van de laatste minuut in Figuur 8
- De gemiddelde hoeveelheid aanvragen naar de interne service per HTTP-verzoek van de laatste minuut in Figuur 9
- De gemiddelde hoeveelheid database query's per HTTP-verzoek van de laatste minuut in Figuur 10
- De gemiddelde tijdsduur per database query van de laatste minuut in Figuur 11
- Het foutpercentage van database query's van de laatste minuut in Figuur 12

Wat in deze figuren nog ontbreekt zijn de filters per service. Het gaat hier om het veld *destination_service* wat als filter toegevoegd moet worden. Dit filter kan dan op de waarde voor de stabiele of canary versie van de service gezet worden, zodat enkel de meetwaarden van een specifieke versie opgehaald worden.

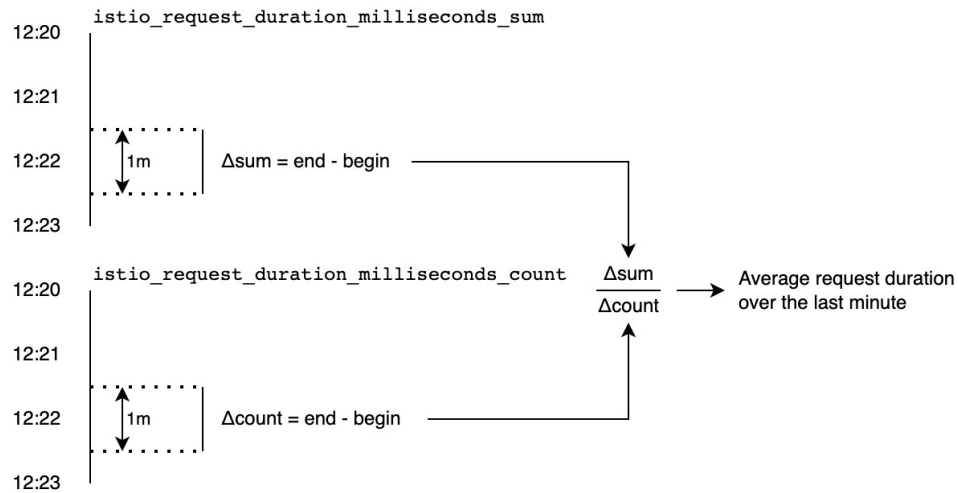
Ook ontbreekt het omzetten van de metrics voor een specifieke versie naar een score voor de canary. Voor al deze metrics is gebruik gemaakt van de volgende formule:

$$score = canary/stable$$

Dit zorgt ervoor dat de score die uit het geheel rolt een vergelijking is ten opzichte van de stabiele versie. Waar absolute getallen niet relevant zijn, is deze methode toegepast. Dit betekent dat ieder van de volgende query's tweemaal toegepast is, een keer op de canary versie en een keer op de stabiele versie. Vervolgens zijn de resultaten van deze query's op elkaar gedeeld om een eindscore te kunnen berekenen.

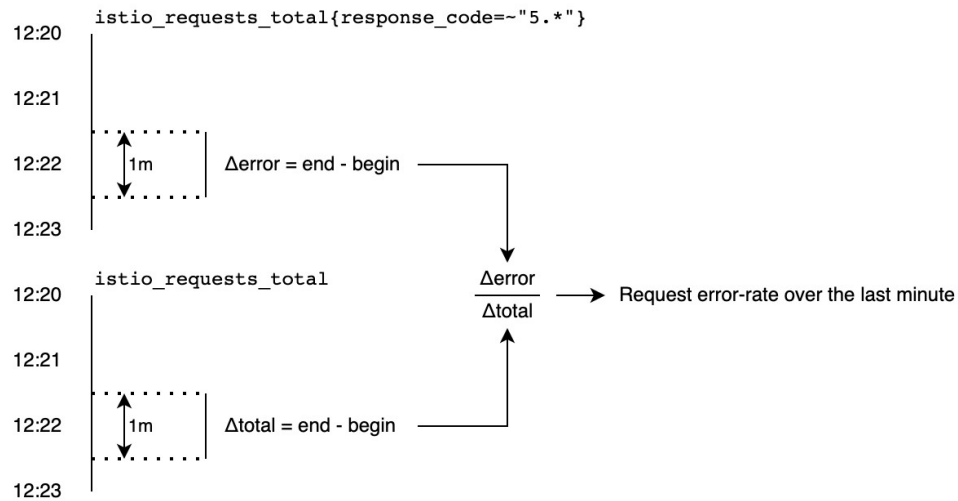
Figuur 6

Query ontwerp gemiddelde tijdsduur requests



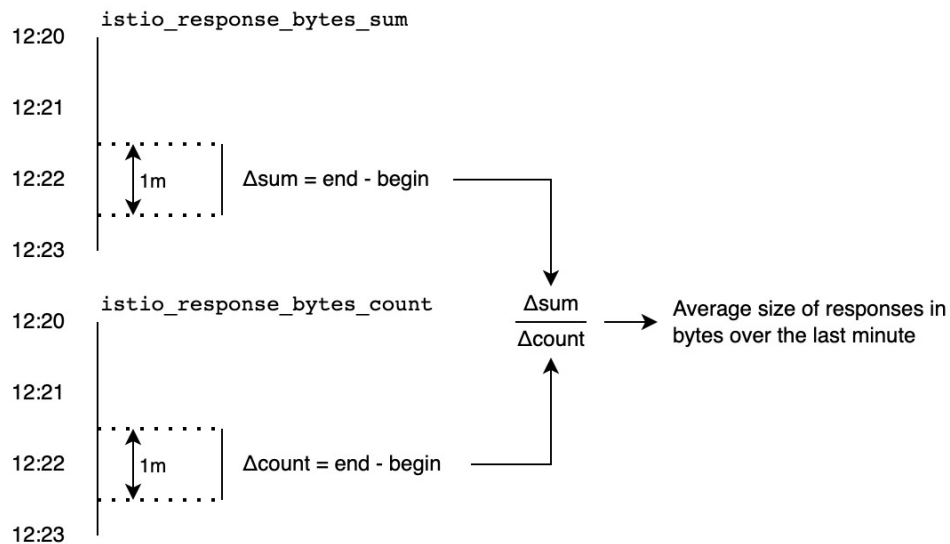
Figuur 7

Query ontwerp foutpercentage requests



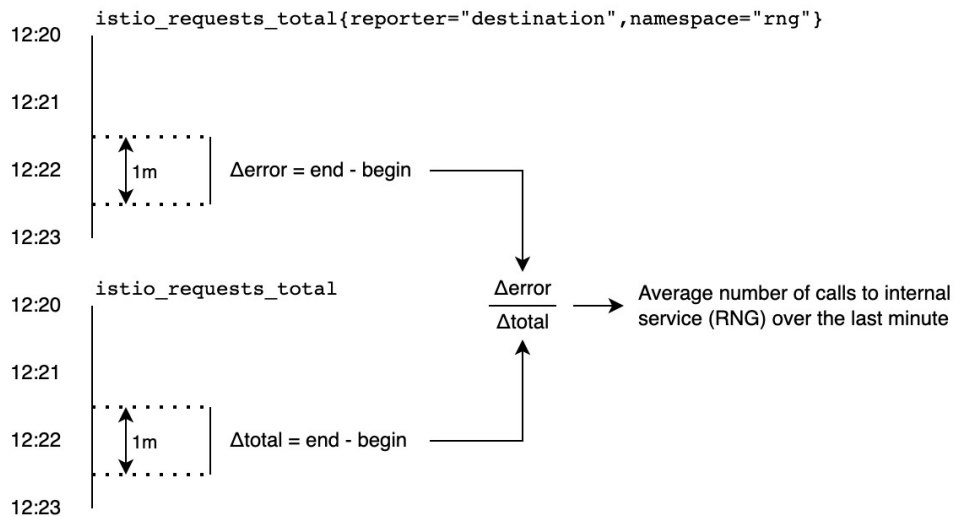
Figuur 8

Query ontwerp gemiddelde response grootte



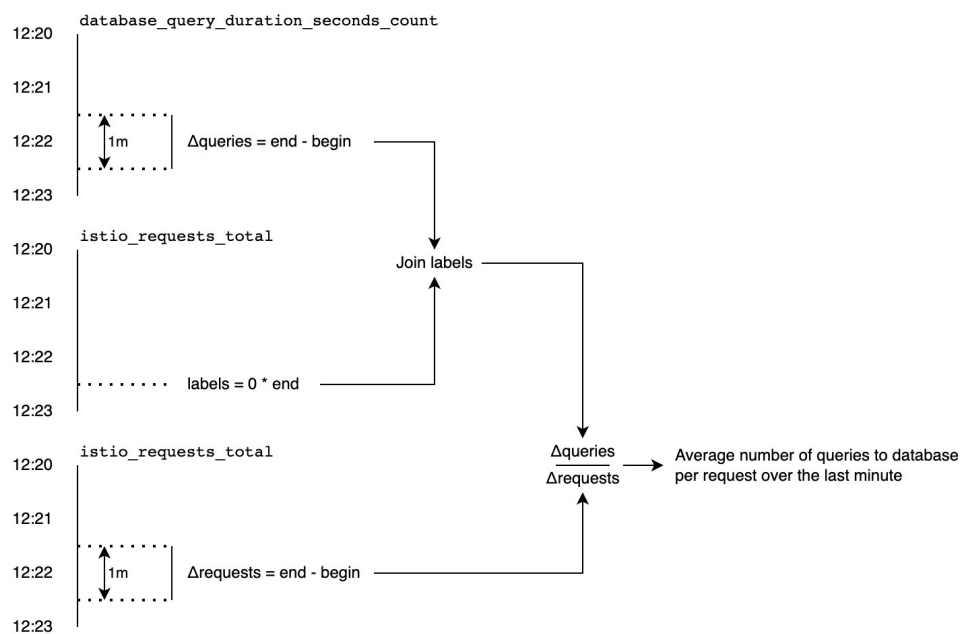
Figuur 9

Query ontwerp gemiddelde hoeveelheid aanvragen naar interne service



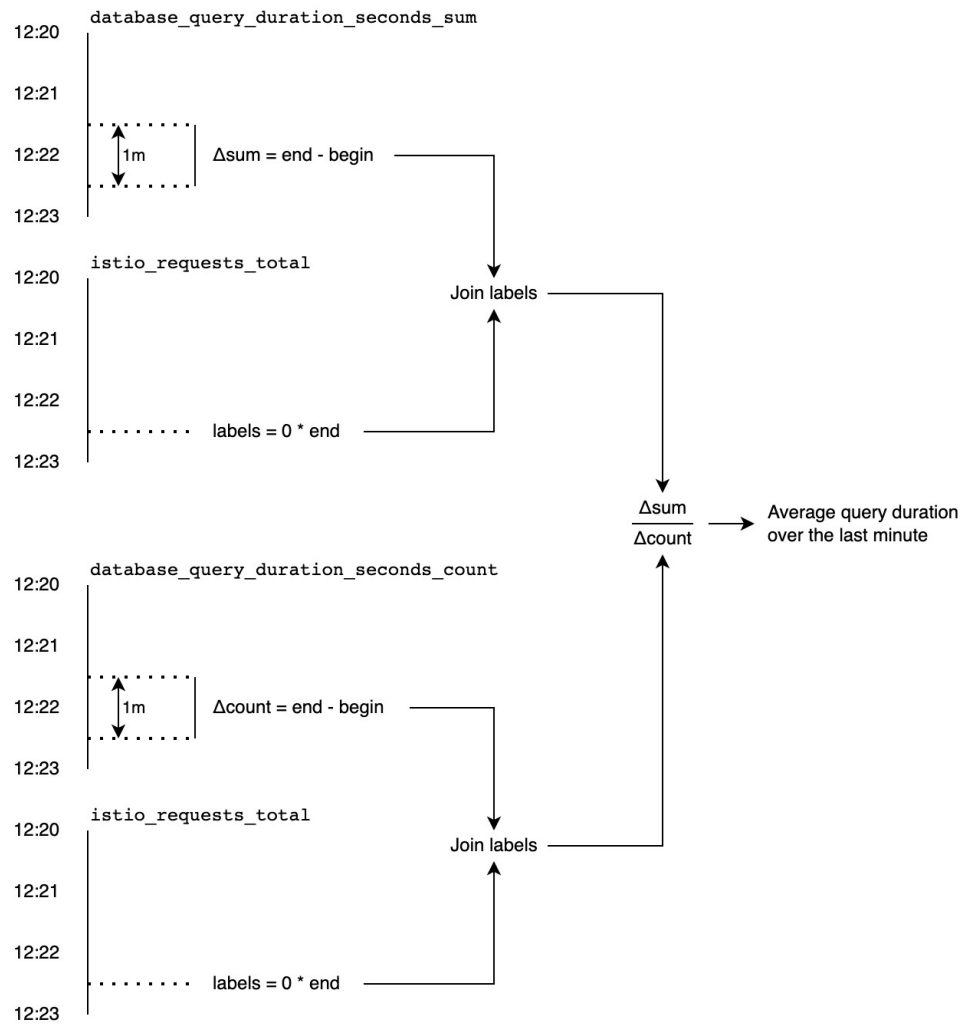
Figuur 10

Query ontwerp gemiddelde hoeveelheid query's op database



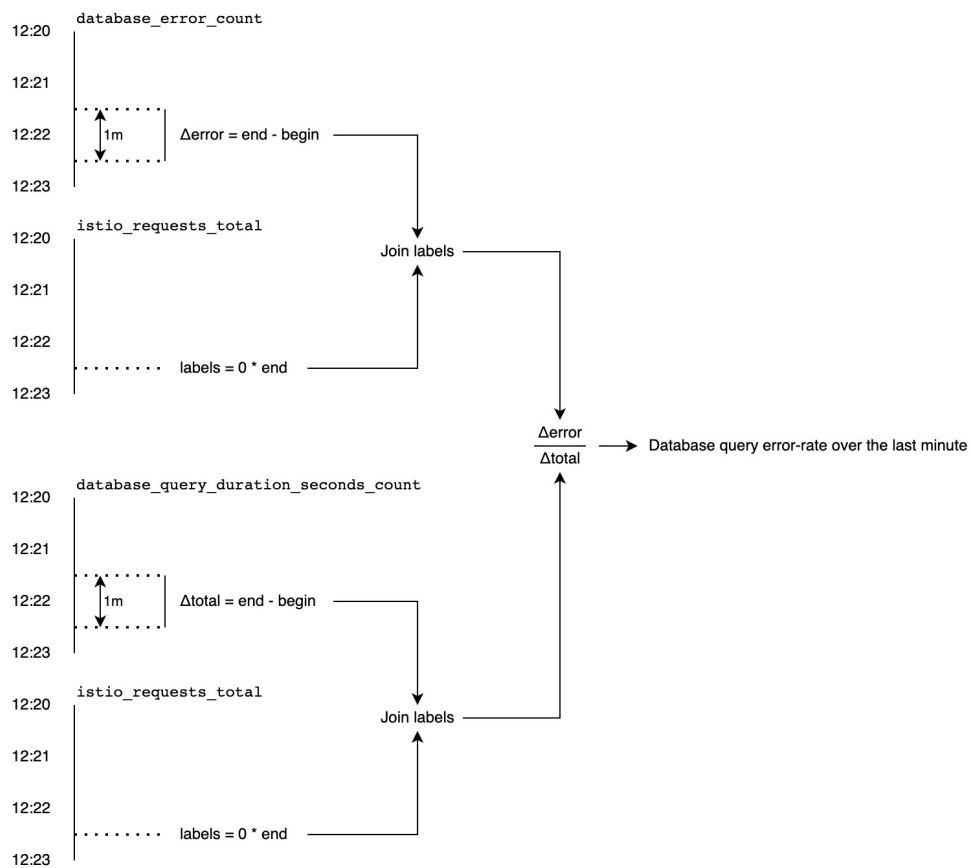
Figuur 11

Query ontwerp gemiddelde tijdsduur van database query's



Figuur 12

Query ontwerp foutpercentage van database query's



Bijlage C Resultaten testopstelling

Deze bijlage bevat alle screenshots van de testopstelling. Deze bijlage is in vier onderdelen gesplitst. Het begint bij een screenshot van de testapplicatie. Vervolgens laat het zien hoe metrics gevisualiseerd zijn binnen Grafana. Daarna komt het overzicht van het verkeer in de applicatie door Kiali. Ten slotte bevat deze bijlage een Visualisatie van het verschil tussen twee metrics, zoals deze in het adviesrapport opgenomen is.

Testapplicatie

De testapplicatie is gebaseerd op de demoapplicatie die Argoproj aanbiedt om hun rollouts mee te testen. Deze applicatie bestaat uit drie belangrijke onderdelen.

Bovenaan staan de sliders, dit zijn de instellingen. Hiermee kunnen per versie de geïmplementeerde kwaliteitsindicatoren aangepast worden.

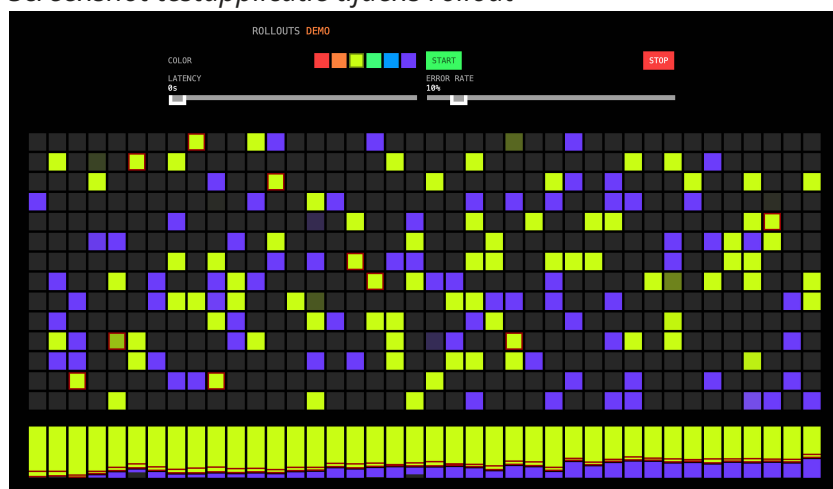
In het midden staat een grid met daarin een visualisatie van alle verzoeken die beantwoord zijn. Deze vlakjes zijn gekleurd gebaseerd op de versie (kleur) die het verzoek verwerkt heeft. Ook staat in de rand aangegeven door een rode kleur of het verzoek een 500 statuscode ontving.

Onderaan staat een tijdlijn van de verdeling verzoeken die door iedere versie beantwoord zijn. Ook staat hierin met een rood lijntje per versie aangegeven welk deel van de verzoeken met een 500 statuscode beantwoord zijn.

Een screenshot van de testapplicatie zoals deze er uit ziet in de browser van een bezoeker is opgenomen in Figuur 13.

Figuur 13

Screenshot testapplicatie tijdens rollout



Prometheus en Grafana

De meetwaarden over alle componenten in de testopstelling zijn opgenomen en opgeslagen door Prometheus. Vervolgens zijn in Grafana een aantal query's opgezet die deze meetwaarden omzetten naar een metric die iets zegt over een kwaliteitsaspect van de applicatie.

De resultaten van deze query's zijn in een dashboard gezet, waardoor in een oogopslag te zien is wat per versie de staat van ieder van deze metrics is. Een screenshot hiervan is opgenomen in Figuur 14.

Figuur 14

Overzicht van metrics in Grafana

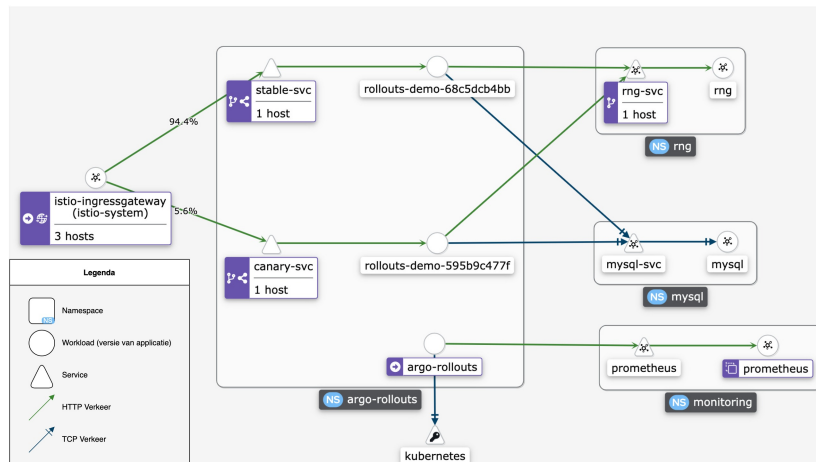


Overzicht van Kiali

Binnen Istio, de ingress controller die al het verkeer accepteert en naar de juiste applicatie doorstuurt, is een extensie aangezet voor Kiali. Kiali is een applicatie die al het verkeer wat door Istio gaat, kan visualiseren. Een visualisatie van het verkeer tijdens een rollout is opgenomen in Figuur 15.

Figuur 15

Visualisatie van testopstelling door Kiali

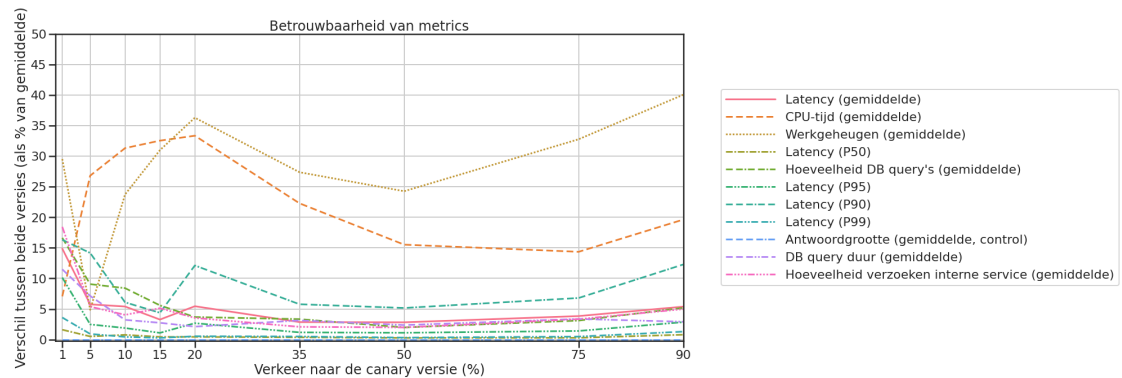


Output van metrics analyse

Aan het eind van de metrics analyse is een visualisatie gemaakt van alle data. Deze data geeft per geteste verdeling van het verkeer de hoeveelheid ruis aan in de metrics. Dit is opgenomen als de afwijking tussen de metrics van beide versies als percentage van de gemiddelde waarde van de metrics. Dit overzicht is opgenomen in Figuur 16.

Figuur 16

Inzicht betrouwbaarheid van de metrics



Inhoudsopgave

1	Inleiding	48
1.1	Over het project	48
1.2	Leeswijzer	48
2	Probleem- en doelstelling	50
2.1	Probleemstelling	50
2.2	Doelstelling	50
2.3	Afbakening	50
3	Onderzoeksopzet	51
3.1	Elicitatiemethodes	51
3.2	Metrics analyse	51
4	Resultaten	53
4.1	Gewenste metrics	53
4.2	Prototype	54
4.3	Metrics analyse	55
5	Aanbevelingen	58
5.1	Onderzoeksvraag	58
5.2	Metrics	59
5.3	Wensen die niet onderzocht zijn	60
5.4	Overzicht	62
6	Literatuurlijst	63

1 Inleiding

1.1 Over het project

YourSurprise biedt een webplatform aan waar klanten bestellingen kunnen samenstellen en plaatsen. Er wordt constant doorontwikkeld aan dit platform. Als er een nieuwe versie in productie gezet wordt, kan het wel eens voorkomen dat een fout onverwachts optreedt. Dit onderzoek probeert te achterhalen welke metrics gebruikt kunnen worden om de kwaliteit van een nieuwe versie te beoordelen.

Troisi (2022) geeft aan dat er altijd een risico is dat problemen zich niet laten zien totdat een applicatie aan klanten blootgesteld wordt ongeacht de hoeveelheid testen en kwaliteit van het testen voordat er naar productie uitgerold wordt.

Dit is ook een belangrijk probleem voor YourSurprise, waar dit als een doel opgenomen is in de Moonshot. Deze Journey voor ship Engineering Productivity is opgenomen als “We test new webshop releases with a small group of customers to minimize impact on performance and availability.”

Het onderzoek is gestart met doel om zekerder te zijn over de betrouwbaarheid van een nieuwe versie van applicaties. Later bleek dat YourSurprise meer baat heeft bij het onderzoek naar metrics dan een implementatie. Dit adviesrapport beschrijft de bevindingen over dat onderdeel van het onderzoek.

1.2 Leeswijzer

Dit adviesrapport beschrijft de probleem- en doelstelling van het uitgevoerde onderzoek. Hier wordt ook de afbakening van het onderzoek benoemd. Vervolgens wordt de onderzoeksofzet beschreven. Dan volgt een overzicht van de resultaten die uit het onderzoek gerold zijn. Ten slotte worden alle aanbevelingen genoteerd die vanuit de resultaten gevormd zijn.

De probleem- en doelstelling is opgenomen als de initiële onderzoeksvraag met de opgestelde hoofd- en deelvragen. Daarnaast beschrijft de afbakening de afspraken die (gedurende het onderzoek) gemaakt zijn om het onderzoek in te perken.

De onderzoeksofzet beschrijft de elicitiemethodes die ingezet zijn bij het onderzoek. Ook beschrijft het de methode die gebruikt is om metrics te analyseren.

De resultaten gaan in op de resultaten van de elicatie. Hierin staat beschreven welke metrics naar voren zijn gekomen als interessant om te onderzoeken. Ook staat hier een overzicht van het prototype wat opgezet is om de bevindingen te ondersteunen. Ten slotte staat hier een overzicht van alle metrics die bekeken zijn bij het onderzoek. Dit functioneert als onderbouwing van de betrouwbaarheid van de metrics bij het toepassen binnen het prototype.

Het adviesrapport sluit af met de aanbevelingen voor de organisatie opgesplitst in drie delen. Eerst de directe adviezen, dit functioneert als een lijstje doelen om een implementatie van canary releases mogelijk te maken. Daarnaast een lijstje algemene adviezen die meegenomen moeten worden bij verder onderzoek. Dit zijn zaken die naar voren gekomen zijn bij het literatuuronderzoek of andere elicatiemethodes die niet een directe betrekking hebben op de analyse van mogelijke metrics.

2 Probleem- en doelstelling

2.1 Probleemstelling

YourSurprise heeft een groot digitaal platform waar meerdere keren per dag een deployment plaats vindt. Er worden vele testen uitgevoerd op nieuwe code om te zorgen dat de meeste problemen voorkomen worden. Dit begint met een automatische teststraat. Echter, sommige problemen zijn niet te testen tijdens dit proces. Nadat het de automatische teststraat uitkomt, wordt nieuwe functionaliteit handmatig getest. Ook dit is (natuurlijk) niet volledig dekkend. Hierdoor komt het wel eens voor dat een versie in productie gezet wordt die een probleem heeft dat pas op dat moment naar voren komt.

Dit onderzoek probeert zekerheid te bieden tijdens deze livegang door een extra teststap te onderzoeken. Dit testproces zal bestaan uit het uitrollen van de nieuwe versie naar een fractie van de bezoekers, vervolgens het bijhouden van cruciale metrics van zowel de nieuwe als de oude versie, ten slotte het automatisch bepalen van de kwaliteit van de nieuwe versie en gepaste actie ondernemen.

2.2 Doelstelling

Om dit probleem aan te pakken heeft YourSurprise gevraagd onderzoek uit te voeren naar zogenoemde Canary Deployments.

Voor dit onderzoeksdoel is de volgende hoofdvraag opgesteld: “Hoe kan YourSurprise er zeker van zijn dat de belangrijkste prestatie-indicatoren van een nieuwe versie niet achteruitgaan bij automatische deployment?”

2.3 Afbakening

Voor YourSurprise is het belangrijkste doel van dit onderzoek om te achterhalen welke metrics betrouwbaar zijn om een release op te baseren. Om dit te achterhalen is het nodig om data te verzamelen vanuit een prototype. Dit prototype hoeft niet inzetbaar te zijn in de bestaande infrastructuur van YourSurprise.

Wel lijkt de testomgeving van dit onderzoek op de infrastructuur van YourSurprise. Om de bevindingen zo relevant mogelijk te maken, is het prototype opgezet in een apart Kubernetes cluster.

Omdat implementatiedetails minder relevant zijn voor de organisatie, zal bij dit onderzoek uitgegaan worden van één deployment tegelijkertijd. Wat er moet gebeuren als een deployment gestart wordt terwijl de vorige nog bezig is, zal buiten beschouwing gelaten worden.

3 Onderzoeksopzet

3.1 Elicitatiemethodes

Om impressies, wensen en requirements op te halen, zijn elicitiemethodes uit het DOT-framework toegepast. Het DOT-framework is een vakgebied onafhankelijke set aan onderzoeksmethoden. Bij dit onderzoek zijn voornamelijk twee methodes ingezet die hierna kort beschreven worden.

De voornaamste manier van informatieverzameling die toegepast is bij dit onderzoek is om te spreken met ontwikkelaars om zo veel mogelijk verschillende potentiële metrics in kaart te brengen.

Daarnaast is gebruik gemaakt van de zogenoemde 'Wizard of Oz' onderzoeksmethode. Deze methode is eigenlijk je publiek voor de gek houden door te doen alsof je een werkend prototype hebt. Dit bereik je door handmatig het prototype aan te sturen. Bij dit onderzoek is dit ingezet door bij de demo handmatig de beslissingen te maken die het systeem (later) zelf zou moeten maken. Op deze manier is het gelukt om snel en visueel het doel van dit onderzoek te delen zonder een volledig werkend prototype te hebben.

Nadat de eerste eisen met de 'Wizard of Oz' onderzoeksmethode duidelijk werden, is een daadwerkelijk prototype gebouwd die de mogelijkheid biedt om een canary proces te doorlopen. Dit is vervolgens uitgevoerd om de metrics analyse te ondersteunen.

3.2 Metrics analyse

Naast het ophalen van interessante metrics om verder te analyseren is het nodig om een aanpak te hebben om deze metrics daadwerkelijk te analyseren. Dit proces is in drie delen gesplitst. Het begint bij het opzetten van de testopstelling. Vervolgens zijn testen uitgevoerd met deze testopstelling. Ten slotte is de data uit deze testen vergeleken om tot een conclusie te komen.

De testopstelling bestaat uit een aantal componenten in een Kubernetes cluster. Het belangrijkste is de testapplicatie die beheerd wordt door de rollouts technologie 'Argo Rollouts'. Hieromheen draaien Istio, om verkeer te accepteren, Prometheus, om meetwaarden over de applicaties op te halen, en een database en een interne applicatie waar de testapplicatie gebruik van maakt.

Vervolgens zijn metingen uitgevoerd met dit platform. Het rollouts proces van nieuwe versie tot een beslissing over de kwaliteit is doorlopen.

Eerst tussen twee versies met verschillende kwaliteitsaspecten. Het doel is om hieruit te halen of een afwijking in kwaliteit gedetecteerd kan worden. Dit is gedaan door te zien of de rollouts technologie tot de verwachte beslissing komt. Dit functioneert enkel als validatie van de opgestelde query's. Zo kan gezien worden of de resultaten van de query's werkelijk overeen komen met de gewenste metric.

Vervolgens is een test uitgevoerd met twee identieke versies waarbij de meetwaarden opgenomen zijn. Over de resultaten van deze tweede test is vervolgens gekeken naar de hoeveelheid ruis tussen de twee versies per stap van het rollouts proces.

4 Resultaten

4.1 Gewenste metrics

Om de metrics op te halen die gewenst zijn vanuit YourSurprise is in gesprek gegaan met een aantal stakeholders. Enerzijds binnen het team van het onderzoek, ship Engineering Productivity. Anderzijds een aantal developers van verschillende ontwikkelingsteams.

Vanuit deze onderzoeksstap zijn de volgende metrics naar voren gekomen. Deze metrics zijn voor zover mogelijk verder onderzocht tijdens dit onderzoek.

- De tijd tussen verzoek en antwoord.
- Het aantal verzoeken naar een database.
- De hoeveelheid CPU-tijd nodig per HTTP-verzoek.
- De hoeveelheid werkgeheugen nodig voor applicatie.
- De verhouding van niet succesvolle HTTP-statuscodes.
- De hoeveelheid gefaalde server-side renders.
- De totale hoeveelheid aanvragen (in verhouding tot de traffic split van de canary).
- Het aantal bytes wat verstuurd wordt.
- De belasting van de database.
- Het aantal aanvragen naar interne services (vanuit backend).
- Het aantal cache hits en misses.
- Het aantal FPM-threads in gebruik.
- Het aantal JavaScript foutmeldingen per browser.
- Het aantal aanvragen naar externe services (vanuit de browser).
- Het aantal database foutmeldingen.

4.2 Prototype

Het prototype bestaat uit een aantal componenten in een Kubernetes cluster. Het eerste component is een testapplicatie die het mogelijk maakt kwaliteitsaspecten in te stellen. Verder staan er een aantal applicaties omheen om het mogelijk te maken verkeer te accepteren en meetwaarden op te slaan. Vervolgens worden deze omgezet naar metrics en dan gevisualiseerd.

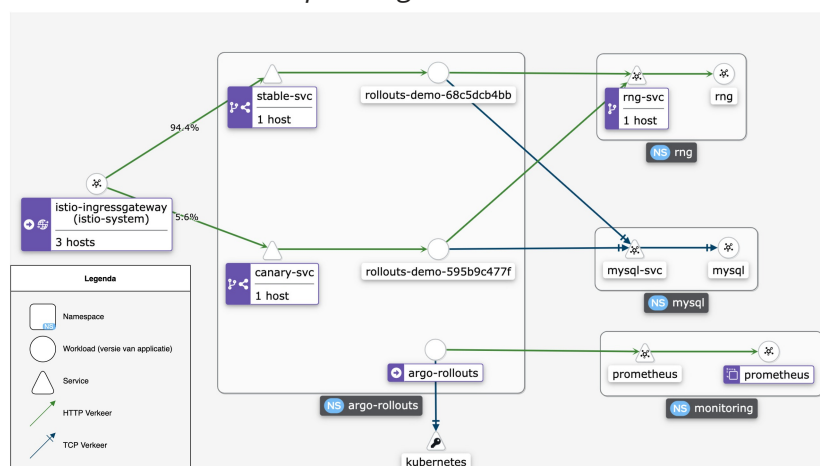
De testapplicatie is gebaseerd op de rollouts demo-applicatie die Argoproj aanbiedt. Deze applicatie maakt het al mogelijk om foutpercentage en latency in te stellen. Tijdens het onderzoek is deze applicatie verder uitgebreid om onder andere een interne service en een database toe te voegen en nieuwe kwaliteitsaspecten instelbaar te maken.

Om verkeer te accepteren en hier meetwaarden over beschikbaar te stellen is Istio toegepast. Istio zorgt voor een service mesh, waardoor het mogelijk is al het verkeer in het cluster meetbaar te maken. Met Kiali, een plugin voor Istio, is later tijdens een rollout deze opstelling inzichtelijk gemaakt. Dit figuur is opgenomen in Figuur 17.

Het laatste onderdeel van het prototype is de monitoring en visualisatie van metrics. Dit is in het prototype bereikt door de combinatie van Prometheus en Grafana toe te passen. Prometheus haalt alle ingestelde meetwaarden op en slaat deze op in een database. Vervolgens kan met behulp van Grafana een visualisatie van deze data gemaakt worden. De metrics zijn ingesteld en vervolgens op een dashboard visueel neergezet.

Figuur 17

Kiali overzicht van testopstelling



4.3 Metrics analyse

De metrics analyse beschrijft het proces om te achterhalen welke metrics tijdens het rollout proces stabiel genoeg zijn om een rollout op te baseren. Alle metrics die ruis vertonen zijn opgenomen bij deze test. Ook is een metric meegenomen die geen ruis vertoont als controle van deze analysemethode. Deze metric zou in elk geval een verschil van nul moeten geven.

Om deze analyse uit te voeren zijn alle query's opgesteld die nodig zijn om de juiste data op te halen. Hiervoor is het eerst nodig dat de meetwaarden zelf op te halen zijn, vervolgens moeten deze omgezet worden naar een metric. Dan kan een berekening uitgevoerd worden met deze metrics om te achterhalen hoeveel ruis ze bevatten.

Binnen deze metrics analyse is voor ieder getest percentage de rollout gepauzeerd voor iets meer dan zes minuten. Net voordat de volgende stap uitgevoerd wordt, is een meting gedaan. Bij deze meting is naar de afgelopen vijf minuten gekeken naar de metrics en het verschil tussen beide versies berekend. Vervolgens is dit omgezet naar een percentage afwijking van het gemiddelde tussen beide versies.

De twee versies zijn identiek opgesteld, waardoor de schuld van alle verschillen tussen deze metrics bij een factor ligt die niets zegt over de kwaliteit van de applicatie. Dit kan zijn dat de metric grotendeels afhankelijk is van een factor zoals hoeveelheid verkeer of ruis die niet te beïnvloeden is.

De metingen zijn zoals hiervoor beschreven uitgevoerd. Vervolgens zijn alle data vanuit deze test opgenomen in een overzicht. Dit overzicht is te zien in Tabel 5.

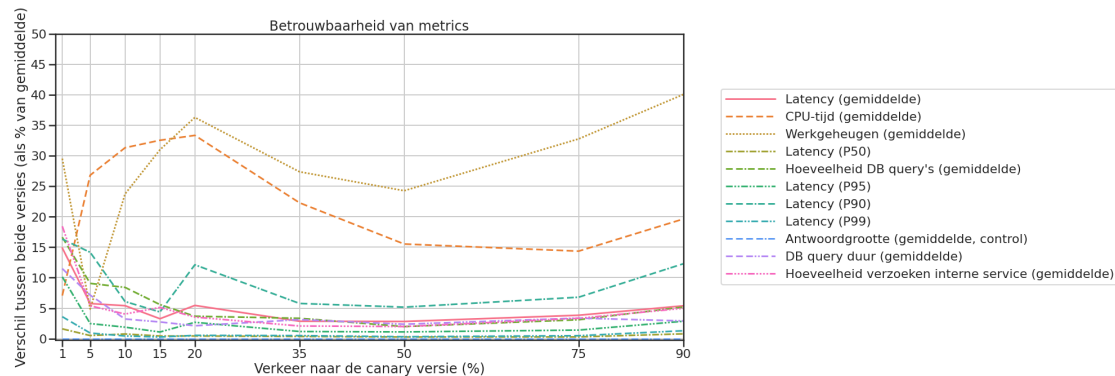
Afwijking metrics per canary split percentage

[illegible]

Al deze data is vervolgens verwerkt naar een overzicht. Dit staat opgenomen in Figuur 18. Dit figuur bevat de data die in Tabel 5 staat. Een aantal metrics hebben een enorme afwijking. Om deze reden is een tweede figuur toegevoegd, namelijk Figuur 19, die enkel afwijkingen van nul tot vijf procent laat zien.

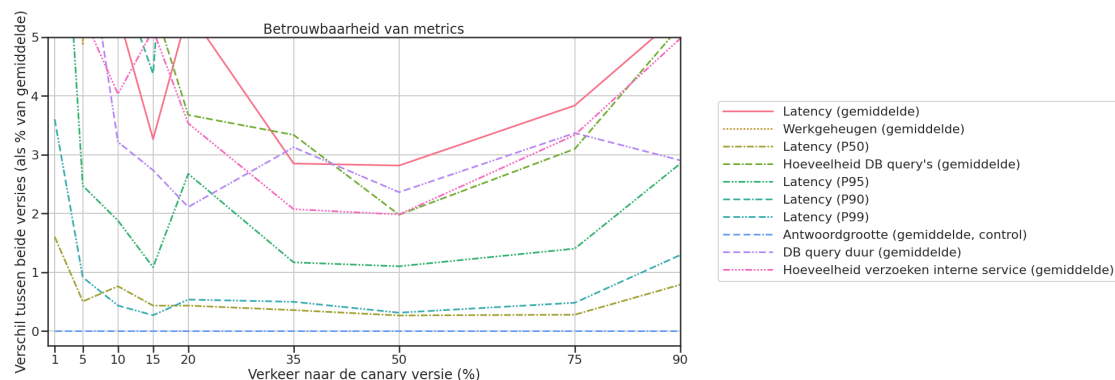
Figuur 18

Afwijking metrics per canary split percentage (0%-50%)



Figuur 19

Afwijking metrics per canary split percentage (0%-5%)



5 Aanbevelingen

Dit hoofdstuk bevat de adviezen die uit dit onderzoek voortkomen. Dit begint bij een antwoord op de hoofdvraag van het onderzoek. Vervolgens beschrijft het de verschillende categorieën metrics die geïdentificeerd zijn. Ook staan de wensen beschreven die bij de elicitatie van dit onderzoek boven kwamen, maar buiten de scope van het onderzoek vallen. Ten slotte staan alle metrics in een figuur op een rijtje als overzicht.

5.1 Onderzoeksvraag

Tijdens het onderzoeken naar de mogelijkheden om canary deployments in te zetten voor de webshop van YourSurprise is de volgende hoofdvraag onderzocht: "Hoe kan YourSurprise er zeker van zijn dat de belangrijkste prestatie-indicatoren van een nieuwe versie niet achteruitgaan bij automatische deployment?".

Bij het onderzoek zijn twee categorieën metrics geïdentificeerd die een beeld vormen over de gewenste prestatie-indicatoren. YourSurprise kan beide soorten metrics gebruiken, maar ze vereisen net een andere aanpak.

Stabiele metrics zijn metrics die bij de huidige productieomgeving van de webshop door weinig ruis beïnvloed worden. YourSurprise kan gebruik maken van dynamische grenzen op deze categorie metrics. Dit betekent dat een nieuwe applicatieversie op deze kwaliteitsaspecten getest kan worden met een maximale verandering vergeleken met de vorige versie. Een concreet voorbeeld van deze vorm van toetsen is: de nieuwe versie mag maximaal 10% langer doen om een verzoek te beantwoorden dan de vorige versie.

Naast de stabiele metrics zijn er ook een groot aantal metrics die sterk beïnvloed worden door ruis. Deze kwaliteitsaspecten kunnen niet gebruikt worden op de gebruikelijke manier waarop een canary beoordeeld wordt. YourSurprise kan deze metrics slechts gebruiken met vrij extreme waardes. Een concreet voorbeeld van deze vorm van toetsen is: de nieuwe versie mag maximaal een foutpercentage van HTTP-verzoeken van 50% hebben.

Door canary releases in te zetten voor nieuwe deployments van de webshop kan YourSurprise voor een paar stabiele metrics verzekeren dat deze niet achteruit gaan. Voor overige kwaliteitsindicatoren kan slechts op een enorm afwijkende waarde getoetst worden.

5.2 Metrics

Dit onderzoek maakt onderscheid tussen twee verschillende type metrics. Deze zijn gecategoriseerd aan de hand van de huidige hoeveelheid verkeer naar de webshop. Deze types beschrijven de stabiele metrics en de metrics die veel ruis vertonen.

5.2.1 Stabiele metrics

Een aantal metrics komen naar voren als stabiele metrics. Deze metrics worden bij ieder HTTP-verzoek een of meerdere keren gemeten, waardoor de stabiliteit voldoende is om de kwaliteit van een nieuwe versie te bepalen.

De geteste metrics die in deze categorie vallen zijn de latency (gemiddelde en percentielen tot het 90e percentiel), het aantal verzoeken naar een database of interne service en de duur van die verzoeken. Op deze metrics zou YourSurprise op dit moment een canary release kunnen baseren.

5.2.2 Metrics met veel ruis

Naast de stabiele metrics zijn er meer die veel ruis vertonen. Dit zijn metrics die externe factoren hebben of op een fractie van de verzoeken gemeten wordt.

Geteste metrics in deze categorie zijn de CPU-tijd per verzoek, het geheugengebruik per verzoek en het percentage van HTTP- en databaseverzoeken die fout gaan.

Een concreet voorbeeld is het percentage HTTP-verzoeken wat met een fout beantwoord wordt. In andere woorden, hoeveel verzoeken een 500 status-codes opleveren. Dit ligt in de productieomgeving van de webshop rond de 0.01% (een in tienduizend). Een vertienvoudiging van het aantal fouten in de webshop is op de korte termijn niet te onderscheiden van ruis.

Wat wel met deze metrics gedaan kan worden is extremen afvangen. Dit betekent dat een absoluut getal als maximum kan worden gebruikt. Een voorbeeld van deze manier van toetsen is: het deel van de verzoeken wat een 500 status-code oplevert, mag nooit meer zijn dan 50% voor de nieuwe versie.

Op deze manier zijn de grenzen niet dynamisch. Ze zijn vooraf bepaald als extreme grens die nooit overschreden zal worden bij een kwalitatief goede applicatieversie. De grenzen van deze categorie metrics moeten zo opgesteld zijn dat het een serieus teken is om onmiddellijk de canary release af te kappen als ze overschreden worden.

5.3 Wensen die niet onderzocht zijn

Niet alle wensen die bij de elicitering naar voren gekomen zijn, waren mogelijk om binnen dit onderzoek uit te voeren.

Een aantal metrics die gewenst zijn om te proberen bij canary releases waren vanwege de keuzes in de testopstelling niet mogelijk te simuleren. Ook waren een aantal gewenste metrics niet te gebruiken bij het beoordelen van de kwaliteit van een nieuwe versie.

Naast wensen voor metrics kwamen ook een aantal implementatiewensen boven. Deze zijn hier genoteerd omdat er tijdens het onderzoek verder niets mee gedaan is.

5.3.1 Metrics die niet bekeken konden worden

Een aantal gewenste metrics zijn niet mogelijk om te testen binnen de testopstelling. Dit komt doordat de metric enorm buiten scope valt of doordat een vereiste technologie niet gebruikt wordt binnen deze opstelling. Hierna volgen de metrics die niet bekeken konden worden met daarbij de onderbouwing voor het weglaten van deze metrics.

Er zijn een aantal metrics die een technologie vereisen die niet in de testopstelling gebruikt is. Zo vereist de metric over het aantal PHP-FPM threads in gebruik een testopstelling die gebruik maakt van PHP. De testapplicatie, gebaseerd op de demo-applicatie van Argoproj, is geschreven in Go. Het toetsen op het aantal cache hits en misses vereist caching. Caching ligt enorm buiten de scope van de testopstelling. De gewenste metric beschrijft namelijk specifiek caching in Fastly. Ten slotte vereist de metric over het aantal gefaalde server-side renders een implementatie van deze techniek binnen de testopstelling.

Ook zijn een aantal metrics niet meegenomen omdat ze een front-end nodig hebben. De front-end van de testopstelling is een applicatie die data over de canary release zichtbaar maakt. Om deze reden zijn metrics die een front-end vereisen niet meegenomen in het onderzoek. Het gaat hier om het aantal foutmeldingen die in de front-end JavaScript code veroorzaakt worden en het aantal verzoeken wat de browser van een bezoeker naar externe services stuurt. Een specifiek voorbeeld wat genoemd werd als externe service is Algolia. Het testen van de metric over de hoeveelheid verzoeken die de browser per applicatieversie verstuurt vereist een implementatie van zowel een front-end om de verzoeken te versturen als het sessie-bewust splitsen van verkeer tussen de versies.

Verder staat de metric over het aantal bytes wat verstuurd wordt los. Deze metric vereist namelijk een volledig functionerende applicatie. Deze metric is instelbaar gemaakt in de testopstelling, maar enkel als vast nummer. Er wordt niet geadviseerd over deze metric omdat dit als controlewaarde functioneerde voor het valideren van de metrics analyse. Deze waarde was namelijk per definitie tussen alle versies identiek.

Ook zijn een aantal metrics niet getest omdat ze onmogelijk zijn om te testen. Het gaat hier om metrics die over de belasting van een gedeelde resource gaan. Bijvoorbeeld hoeveel CPU-tijd nodig is om alle database query's te verwerken. Deze gedeelde database wordt voor meer dan alleen deze applicatie gebruikt, waardoor het niet mogelijk is om de impact van de canary applicatie hierop te bepalen.

5.3.2 Implementatiewensen

Tijdens het onderzoek zijn een aantal wensen naar voren gekomen die niet gebruikt zijn voor het resultaat, omdat dit implementatiewensen waren. Deze wensen zijn op een rijtje gezet om onderzoek naar een implementatie alvast wat houvast te geven.

Gert de Pagter kwam met een voorbeeld waar canary releases een negatieve impact op de innovatiemogelijkheden zou hebben. De nieuwe versie van de page-builder stuurt veel minder verzoeken naar de database. Bij het livezetten van deze versie zou dit invloed hebben op een kwaliteitsindicator die over database query's gaat. Als vooraf bekend is dat een kwaliteitsindicator enorm zal aanpassen, maar dat deze aanpassing gewenst is, moet het mogelijk zijn om het canary proces te omzeilen. In dit voorbeeld gaat het om een verandering die positief is op de metrics, maar het zou ook andersom om welke reden dan ook gewenst kunnen zijn.

Lesley van der Woude gaf bij een demo van de voortgang van het onderzoek aan dat hij graag een specifieke versie zou willen testen. Dit zou voor hem enkel relevant zijn als canary releases bij een deployment naar de acceptatieomgeving toegepast worden. Het onderzoeken van het inzetten van canary releases in de acceptatieomgeving valt buiten scope van dit onderzoek. Ook moet hier overwogen worden of dit iets is wat gewenst is. Er moet dan namelijk genoeg verkeer op de acceptatieomgeving zijn om een beslissing over de kwaliteit van de applicatieversie te maken.

5.4 Overzicht

Alle metrics zijn gegroepeerd per categorie opgenomen in Figuur 20.

Figuur 20

Overzicht metrics

Stabiel	
Aantal verzoeken naar database	Aantal verzoeken naar interne service
Duur van verzoeken naar database	Duur van verzoeken naar interne service
Latency	

Veel ruis	
CPU-tijd per verzoek	Percentage HTTP-verzoeken die fout gaan
Geheugengebruik per verzoek	Percentage database-verzoeken die fout gaan

Buiten Scope	
PHP-FPM threads	Front-end foutmeldingen
Cache hits/misses	Aantal front-end verzoeken naar externe service
Server-side rendering	Aantal verzoeken naar applicatie vanuit front-end
Aantal bytes verstuurd	

Niet mogelijk	
Belasting van gedeelde componenten	

6 Literatuurlijst

Troisi, H. (2022, februari 11). *Automate Canary Analysis on Kubernetes with Argo*.
Verkregen 21 maart 2022, van <https://blog.px.dev/argo-rollouts/>