

Datadog (stock symbol: DDOG), 2021. © 2023 CompaniesLogo.com

**Voorbereiden op *scalability*: Datadog monitoring van WeaveWorks Sock Shop**  
**SNB 3**

Mateo Lemmens

Tomas Soors

Victor Beckx

Rasmus Leseberg

Projectomschrijving

Datadog is een monitoringservice voor *cloud-scale* applicaties, die monitoring van servers, databases en andere functies aanbiedt via een *SaaS*-gebaseerd platform. Als integratie van de WeaveWorks Sock Shop kan Datadog relevante data monitoren, die door middel van bepaalde *metrics* en alerts de webshop voorbereidt op vlak van *scalability*. Deze paper onderzoekt features, integraties en alertsystemen van Datadog, met als doel om een overzicht van bruikbare data te scheppen aan de hand van verschillende userscenario’s. De analyse onderzoekt de correlaties tussen de *metrics* van data, gebruikt gesimuleerde userscenario’s om bepaalde microservices onder druk te zetten, en kijkt naar alertsystemen die relevante waarschuwingen weergeven.

De leidende onderzoeksvraag klinkt als volgt: “Welke *metrics* kunnen via Datadog worden verwerkt om een website/Sock Shop voor te bereiden op vlak van *scalability*?”. Meer specifiek, deze onderzoeksvraag zet dus de integraties, *metric* types en alertsystemen van Datadog uiteen, met ondersteuning van gerelateerde deelvragen.

Het grootste deel van de research berust op een technisch onderzoek dat loadtesting scripts zal gebruiken om verschillende userscenario’s te simuleren, met als doel om de microservices van de Sock Shop te targeten en te stresstesten. Vervolgens focust de research zich ook op het correct gebruik en instellen van alertsystemen, zodat de microservices vroegtijdig kunnen communiceren dat die onder druk staan. Door te onderzoeken welke *KPIs* van belang zijn vanuit de alertsystemen is het mogelijk om de Sock Shop voor te bereiden op grotere bezoekersaantallen. Het belang hiervan is dat een groot aantal bezoekers de beschikbaarheid en functionaliteit van de webshop niet mag beperken.

Inhoudsopgave

[Projectomschrijving ii](#_Toc132569609)

[Inhoudsopgave iii](#_Toc132569610)

[Lijst van gebruikte figuren iv](#_Toc132569611)

[Lijst van gebruikte tabellen v](#_Toc132569612)

[Lijst van gebruikte afkortingen vi](#_Toc132569613)

[1 Onderzoeksvraag en hypothese 1](#_Toc132569614)

[1.1 Hoofdvraag en deelvragen 1](#_Toc132569615)

[1.2 Hypothese 1](#_Toc132569616)

[2 Onderzoeksmethode 2](#_Toc132569617)

[3 Literatuurstudie 3](#_Toc132569618)

[3.1 Microservices en containers 3](#_Toc132569619)

[3.2 Wat is microservice *scalability*? 3](#_Toc132569620)

[3.3 Deelvraag 1: *Integrations* 4](#_Toc132569621)

[3.4 Deelvraag 2: Dashboards 5](#_Toc132569622)

[3.5 Deelvraag 3: Monitoring 6](#_Toc132569623)

[4 Uitvoering 8](#_Toc132569624)

[4.1 Hoofdstuk 8](#_Toc132569625)

[4.2 Hoofdstuk 8](#_Toc132569626)

[5 Conclusie 9](#_Toc132569627)

[Bibliografie 10](#_Toc132569628)

[Bijlagen 11](#_Toc132569629)

Lijst van gebruikte figuren

[Figuur 1: Voorbeeld van een loadtest script 2](#_Toc132560467)

Lijst van gebruikte tabellen

Lijst van gebruikte afkortingen

**Afkorting Definitie**

SaaS Software as a Service

KPI Key Performance Indicators

KP Key Performance

API Application Programming Interface

RAM Random Access Memory

CPU Central Processing Unit

OS Operating System

CI/CD Continuous Integration/Continuous Delivery

# Onderzoeksvraag en hypothese

## Hoofdvraag en deelvragen

Deze researchpaper onderzoekt welke *metrics* Datadog verwerkt om een microservice-gebaseerde webshop voor te bereiden op vlak van *scalability*. De onderzoeksvraag is:

*“Welke* metrics *kunnen via Datadog worden verwerkt om een website/Sock Shop voor te bereiden op vlak van* scalability?*”*

Om deze onderzoeksvraag te beantwoorden, zijn er verschillende deelvragen opgesteld die zich richten op de aparte features van Datadog:

1. *Integrations*: *“Welke integraties biedt Datadog aan in verband met de Sock Shop voor het verzamelen van* metrics*, en welke features van deze integraties zijn relevant?*
2. Dashboards: *“Welke* metrics *zijn van belang in verband met* scalability, *en hoe kunnen deze verzamelde* metrics *gecorreleerd worden binnen Datadog?”*
3. Monitoring: *“Welke combinatie van monitortypes binnen Datadog kunnen ingesteld worden ter voorbereiding van* scalability*?”*

Deze deelvragen vormen de leidraad van het onderzoek en zullen tijdens het technisch onderzoek ondersteund worden aan de hand van toegepaste userscenario’s.

## Hypothese

De stelling van dit onderzoek is dat het instellen van correcte alerts de WeaveWorks Sock Shop voorbereidt op een *scalability* scenario. De Datadog integraties verzamelen de belangrijkste *KPIs* van specifieke containers*,* en de monitoringtypes bieden de mogelijk aan om vroegtijdig te reageren. Met behulp van een overzichtelijk dashboard is het voor de gebruiker mogelijk om de microservices apart te analyseren en kritische alerts op de juiste plek te tonen.

Met behulp van loadtesting scripts is het mogelijk om de microservices tot hun limiet te belasten, en daardoor de *KPIs* op de juiste hoeveelheid in te stellen. Door herhalende loadtesting scripts uit te voeren is het mogelijk om gemiddelde *KPI*-waarden te bepalen, en deze te gebruiken om de Sock Shop voor te bereiden om te schalen. De loadtesting scripts zijn gebaseerd op realistische userscenario’s die specifieke functies van de microservices targeten, en de scripts zorgen voor geïsoleerde testomgevingen om de microservices onder druk te zetten.

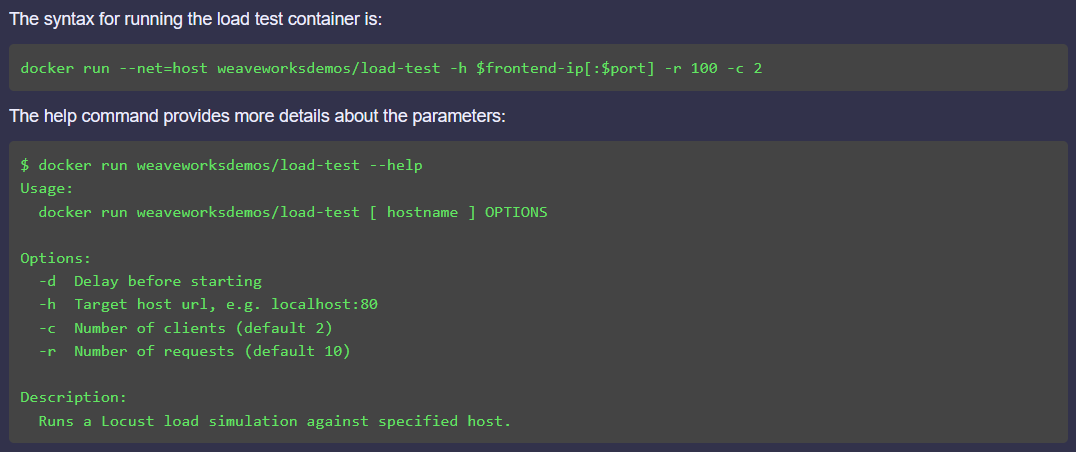
Uiteindelijk is het door de resultaten van het technisch onderzoek mogelijk te bepalen precies wanneer het tijd is om voor de Sock Shop te schalen voor hogere bezoekersaantallen.

# Onderzoeksmethode

De onderzoeksmethode voor deze researchpaper is gebaseerd op een technisch onderzoek wat gebruik maakt van loadtesting scripts. Het Datadog platform zal de verzamelde data van de tests tonen in een dashboard, om zo bepaalde *KPIs* als alerts vast te leggen.

Voor de containers of microservices waarvoor integraties bestaan worden deze ook gebruikt. Alle containers krijgen bijvoorbeeld de Docker integratie, de databases krijgen de MongoDB en MySQL integratie, de ‘front-end’ microservice krijgt de NodeJs integratie en de ‘edge-router’ microservice krijgt de Traefik integratie.

Gerichteloadtesting scripts zullen zoals vermeld de microservices van de webapplicatie onder druk zetten. Deze scripts zijn gebaseerd op realistische userscenario’s, die gericht bezoekersverkeer simuleren en specifieke functies van de webshop targeten. Scripts zorgen ervoor dat deze testen gemakkelijk en consistent zijn in uitvoering. Een voorbeeld van een loadtesting container wat de ‘front-end’ microservice belast is te zien in *Figuur 1*.



Figuur : Voorbeeld van een loadtest script [1]

Het voorbeeld in *Figuur 1* maakt gebruik van een WeaveWorks container als loadtest om de ‘front-end’ microservice te belasten, maar om bepaalde functies van de webshop te isoleren zijn zelfgeschreven scripts waarschijnlijk eerder van toepassing. Het doel zal zijn om de kritische functies van de webshop te isoleren, zoals bestellingen plaatsen, inloggen, de cataloog van producten bekijken, zodat de limieten van de kritische infrastructuur vastgelegd kunnen worden.

Het volgende hoofdstuk zal de integraties en monitoringtypes van Datadog uiteenzetten, met als introductie een dieper inzicht tot microservices, containers en schaalbarheid.

# Literatuurstudie

De hoofdvraag is gefocust op *scalability* binnen een webapplicatie context, waar de Sock Shop webapplicatie uit containers bestaat die microservices draaien. *Scalability* heeft daardoor een specifieke context en definitie.

## Microservices en containers

Een microserviceframework bestaand uit microservices die binnen containers draaien, creëert een systeem dat de knelpunten van een centrale database vermijdt. Het maakt ook *CI/CD*-pipelines voor applicaties mogelijk en moderniseert de *technology stack* [2].

Containers zijn een lichtgewicht, efficiënte oplossing voor applicaties om tussen omgevingen te communiceren en onafhankelijk te werken. Alles wat nodig is (behalve het gedeelde besturingssysteem op de server) om de applicatie uit te voeren, is verpakt in het containerobject: de code, runtime, systeemafhankelijke tools, library’s en andere *dependencies*. Containers zijn een standaardmanier om microservices te implementeren, omdat de microservice als containerimage verpakt is, elke service bestaat binnen een aparte container, en schalen gebeurt op basis van gewijzigde containerinstanties.

Microservices daarentegen zijn een populaire optie voor het moderniseren van applicaties. Enerzijds zijn ze kostenbesparend, waardoor ze ideaal zijn voor kleinere bedrijven. Anderzijds kunnen microservices ook cloudnative zijn, zoals de Sock Shop in dit geval, waardoor ruimte wordt bespaard die normaal nodig is voor *on*-*premise* systemen. Wat nog crucialer is, microservices omzeilen de problemen van monolithische applicaties. De problemen van monolithische applicaties zijn typisch:

* Ze zijn moeilijk en langzaam te onderhouden en te testen.
* Het repareren van een functie met bugs of onderhoud betekent downtime voor de hele applicatie.
* Moeite met het beheersen van verschillende programmeertalen.

Toch is schaalbaarheid het belangrijkste voordeel van microservices in vergelijking met monolithische applicaties; de capaciteit van microservices kan worden geconfigureerd om dynamisch te schalen afhankelijk van navraag [3].

## Wat is microservice *scalability*?

Containerservers schalen (*scalen)* wanneer hun systeemresources (CPU, RAM, netwerkbandbreedte, enz.) worden verhoogd om te voldoen aan de navraag die het systeem vereist. Containers kunnen hun resources reguleren op twee manieren:

1. Verticaal schalen: de applicatiecapaciteit vergroten door de capaciteit van de servers te vergroten, virtueel of fysiek. Voorkeur ligt hier bij *stateful* apps, omdat deze vereisen dat de klantinformatie tussen sessies moet worden bewaard om te werken.
2. Horizontaal schalen: het aantal serverinstanties vergroten om de vraag te beheren. Voorkeur voor *stateless* applicaties omdat deze geen klantgegevens opslaan.

*Stateless* en *stateful* protocollen verschillen van elkaar. Een *stateless* systeem stuurt een *request* naar de server en stuurt het antwoord (of de status) van de server terug zonder enige informatie op te slaan. Aan de andere kant verwachten *stateful* systemen een antwoord, volgen deze informatie op, en verzenden het verzoek opnieuw als er geen antwoord wordt ontvangen [4].

In het geval van horizontaal schalen is er ook het idee van *downscaling*, namelijk het verminderen van serverinstanties wanneer de navraag daalt. Op deze manier kan schalen ook de kosteneffectiviteit van microservices verbeteren [3].

Vervolgens zullen bepaalde integraties van Datadog besproken worden om meer inzicht te bieden in de opbouw van het technische onderzoek.

## Deelvraag 1: *Integrations*

*Integrations*: *“Welke integraties biedt Datadog aan in verband met de Sock Shop voor het verzamelen van* metrics*, en welke features van deze integraties zijn relevant?”*

Om gegevens correct te kunnen monitoren en het type van *metrics* te bepalen, is het belangrijk om eerst te begrijpen hoe Datadog gegevens verzamelt. Daarnaast is het ook belangrijk om te begrijpen uit welke onderdelen de Sock Shop is opgebouwd om zo de juiste integratie features te bepalen.

### Wat zijn integraties?

Integraties zijn de fundamentele bouwstenen voor de verzameling van data binnen Datadog. Integraties bestaan uit verschillende tools en features die de mogelijkheid bieden om een bepaalde service te meten en gegevens hierover te verzamelen. Binnen Datadog bestaan er meer dan 600 verschillenden integraties, beschikbaar voor diverse services zoals: Docker, Apache, en MySQL onder andere.

Datadog biedt deze integraties aan op drie verschillende manieren. De meest voorkomende manier is een integratie die via de Datadog agent draait. Een ontwikkelaar heeft ook de mogelijkheid om binnen het Datadog platform een integratie te ontwikkelen op maat van hun applicatie.

De verzamelde gegevens noemen we ook wel logsof *metrics*. Naast data verzamelen is het ook de verantwoordelijkheid van de integraties, samen met de Datadog agent, om de data door te sturen naar een centrale plaats. Deze centrale plaats is het online Datadog platform [5].

### De Datadog agent

De Datadog agentis een open sourceapplicatie die de basis vormt om gegevens te loggen over een bepaalt systeem. De software is online beschikbaar via het Datadog platform en de broncode is gepubliceerd op GitHub [6].

De agentwordt geconfigureerd op ieder individueel systeem waar een verzameling van informatie op moet gebeuren, en waarvan de informatie beschikbaar gesteld moet zijn op het online Datadog platform. De agentkan op diverse besturingssystemen draaien (Windows, Linux, macOS), en is met andere woorden de brug tussen enerzijds het systeem, en anderzijds de online omgeving van Datadog. Daarnaast vormt deze ook de basis voor de installatie en configuratie van de Datadog integraties. Zonder de agentkunnen bepaalde integraties niet werken. Ten slotte gaat de agentvia de Datadog *API* de verzamelde informatie die van de integraties komen doorsturen naar een centrale plaats, namelijk het online Datadog platform [7] [8].

De kracht van de Datadog agent is dat er binnen een omgeving meerdere agentstegelijk op diverse systemen kunnen draaien. Ze bieden een ondersteuning aan de integraties, en ze sturen allemaal informatie door naar een centrale plaats.

### Relevante integraties voor de Sock Shop

Via de Datadog agent is het mogelijk om systeemrelevante gegevens te meten, bijvoorbeeld RAM en CPU gebruik. Deze zijn ook in het online platform opgenomen en kunnen eventueel relevant zijn om een *scalability* scenario voor te bereiden.

De Sock Shop is hoofdzakelijk opgebouwd uit drie elementen. Er zijn twee databases: MongoDB en MySQL, de rest van de services draaien allemaal in verschillende Docker containers. Aangezien dat MongoDB en MySQL op verschillende servers draaien gaat er op de agent van die servers enkel de integratie voor dat typen database geconfigureerd worden. Alle andere servers, waar de verschillende services van de Sock Shop onderverdeeld zijn over Docker containers, krijgen de Datadog Docker integratie geïnstalleerd.   
  
Bepaalde services van de Sock Shop zouden ook met behulp van meer specifieke integraties gemonitord kunnen worden, bijvoorbeeld zou de NodeJs integratie voor de ‘front-end’ service relevant kunnen zijn, of de Traefik integratie voor de ‘edge-router’ service.

## Deelvraag 2: Dashboards

Dashboards: *“Welke* metrics *zijn van belang in verband met* scalability, *en hoe kunnen deze verzamelde* metrics *gecorreleerd worden binnen Datadog?”*  
  
Dashboards zijn handig om *KP* *metrics* te visualiseren en te volgen. Er zijn een aantal features die het opzetten van een dashboard mogelijk maken, zoals Widgets, *Querying, Functions, Template Variables,* en *API* [9]*.* In dit onderzoek komen voornamelijk Widgetsaan board, wat bouwstenen zijn om een visueel dashboard op te bouwen. Dashboards hebben een raster lay-out, wat gemakkelijk met drag-and-drop te navigeren is, en bieden ook aanvullende opties om bepaalde datacollectie te debuggen. Realtime vervolg van de draaiende microservices en hun alerts is het uiteindelijke doel van dit onderzoek, om ervoor te zorgen dat de Sock Shop op tijd geschaald kan worden in het geval van toenemend bezoekersverkeer [9].

### Welke metric types zijn er?

Elke *metric* die bij Datadog wordt ingediend, moet een type hebben. Het type van een *metric* beïnvloedt hoe de *metric*-waarden worden weergegeven wanneer ze worden opgevraagd, en wat de grafische mogelijkheden binnen Datadog zijn. Het type *metric* wordt weergegeven in het detailzijpaneel voor de gegeven *metric* op de ‘Metrics Summary’ pagina [10].

De volgende *metric* types zijn door Datadog geaccepteerd:

* COUNT
* RATE
* GAUGE
* DISTRIBUTION

Welke type van *metric* toepasselijk zal zijn hangt van de microservice af, en wordt dieper onderzocht tijdens het technisch onderzoek. Het volgende stuk zal de monitoringtypes van Datadog uiteenzetten.

## Deelvraag 3: Monitoring

Monitoring: *“Welke combinatie van monitortypes binnen Datadog kunnen ingesteld worden ter voorbereiding van* scalability*?”*

### Wat zijn monitors binnen Datadog?

Zoals eerder vermeld in dit onderzoek is het verzamelen van gegevens via de integraties van Datadog, en het correleren van deze gegevens met de dashboards van Datadog van groot belang voor monitoring en *scalability*. Het is echter niet voldoende om alleen gegevens op een centrale locatie te verzamelen. Het is van belang om te weten wanneer er verdachte of kritieke veranderingen optreden in deze gegevens. Hiervoor biedt Datadog de functie 'Monitors' aan.

Deze monitors controleren actief de verzamelde gegevens, de mogelijke integraties die van toepassing zijn op de systemen, netwerkeindpunten en meer [11]. Gebruikers kunnen monitors aanmaken om specifieke waarschuwingen te creëren. Dit kan door gebruik te maken van beschikbare monitortypes die Datadog aanbiedt en gepersonaliseerde *KPIs* die voor de gebruiker van belang zijn. Alle monitors die zijn aangemaakt worden weergegeven onder 'Manage Monitors' en kunnen daar worden gemaakt, verwijderd, gekopieerd, gedempt of aangepast.

Automatisch ingestelde notificaties voor aangemaakte monitors kunnen de gebruiker of het team op de hoogte stellen wanneer er een verdachte of kritieke verandering optreedt in de verzamelde gegevens. Dit kan via e-mail, Slack, of andere integraties van Datadog. Deze notificaties worden alleen verstuurd wanneer een ingestelde *KPI* wordt overschreden. De notificatie kan diverse informatie bevatten, zoals details over de overschreden *KPI* en eventuele screenshots. De mobiele Datadog applicatie biedt ook de mogelijkheid om deze monitors op te volgen [11].

### Welke monitor types biedt Datadog aan?

Monitors kunnen ingesteld worden met een bepaald monitortype, waarvan het type monitor afhankelijk is van de waarschuwingssoort die de gebruiker of het team wil ontvangen.

Het hostmonitortype gaat bijvoorbeeld controleren of de geconnecteerde host via Datadog agent data aan het versturen is naar het Datadog platform. Hierdoor kan een alert onmiddellijke notificaties versturen naar de gebruiker of het team, als één of meerdere hosts geen data meer verstuurt naar de Datadog server [12]. Dit wil dan zeggen dat één of meerdere hosts uitgevallen zijn door overbelaste services. Op tijd reageren op monitors zorgt voor een snelle responstijd en helpt om de Sock Shop voor te kunnen bereiden op vlak van *scalability*.

Het volgende hoofdstuk toont de uitvoering van het onderzoek.

# Uitvoering

## Hoofdstuk

### Hoofdstuk

## Hoofdstuk

# Conclusie

Bibliografie

|  |  |
| --- | --- |
| [1] | „Load test,” Sock Shop, 2017. [Online]. Available: https://microservices-demo.github.io/docs/load-test.html. [Geopend 16 April 2023]. |
| [2] | „Microservices And Containers,” AVI Networks, [Online]. Available: https://avinetworks.com/what-are-microservices-and-containers/. [Geopend 16 April 2023]. |
| [3] | „Microservices Scalability as a Business Issue,” Optimus Information, 29 June 2022. [Online]. Available: https://www.optimusinfo.com/microservices-scalability-as-a-business-issue/. [Geopend 15 April 2023]. |
| [4] | C. BasuMallick, „Stateful vs. Stateless: Understanding the Key Differences,” spiceworks, 20 September 2022. [Online]. Available: https://www.spiceworks.com/tech/cloud/articles/stateful-vs-stateless/. [Geopend 16 April 2023]. |
| [5] | „Introduction to Integrations,” Datadog, [Online]. Available: https://docs.datadoghq.com/getting\_started/integrations/. [Geopend 14 April 2023]. |
| [6] | Sagar, „Getting Started with the Datadog Agent on Ubuntu Linux,” Talearning, 31 March 2022. [Online]. Available: https://adamtheautomator.com/datadog-agent/. [Geopend 14 April 2023]. |
| [7] | „Getting Started with the Agent,” Datadog, [Online]. Available: https://docs.datadoghq.com/getting\_started/agent/. [Geopend 14 April 2023]. |
| [8] | „Datadog Agent,” Cribl, [Online]. Available: https://docs.cribl.io/shared/sources-datadog-agent/#:~:text=Datadog%20Agent%20is%20open%2Dsource,or%20containers%20on%20the%20host. [Geopend 14 April 2023]. |
| [9] | „Dashboards,” Datadog, [Online]. Available: https://docs.datadoghq.com/dashboards/. [Geopend 16 April 2023]. |
| [10] | „Metri Types,” Datadog, [Online]. Available: https://docs.datadoghq.com/metrics/types/?tab=count. [Geopend 16 April 2023]. |
| [11] | „Monitor Types,” Datadog, [Online]. Available: https://docs.datadoghq.com/monitors/types/. [Geopend 15 April 2023]. |
| [12] | „Alerting,” Datadog, [Online]. Available: https://docs.datadoghq.com/monitors/. [Geopend 15 April 2023]. |

Bijlagen

1. Omschrijving Bijlage A
2. Omschrijving Bijlage B
3. Omschrijving Bijlage C
4. Omschrijving Bijlage A
5. Omschrijving Bijlage B
6. Omschrijving Bijlage