



UNIVERSITY OF APPLIED SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Bachelor Thesis

Vergleich eines Usecases mit Serverless Technologie gegenüber Spring Boot Technologie am Beispiel von Instant Payments

Eingereicht am:

25. August 2021

Eingereicht von:

Silas Hoffmann

Traberweg 52

22159 Hamburg

Tel.: (040) 643 94 73

E-mail: inf103088@stud.fh-wedel.de

Referent:

Prof. Dr. Dennis Säring

Fachhochschule Wedel

Feldstraße 143

22880 Wedel

Phone: (041 03) 80 48-43

E-mail: dennis.saering@fh-wedel.de

Betreut von:

Kai Roßdeutscher

DPS Engineering GmbH

Eiffestraße 78

20537 Hamburg

Phone: (040) 25 15 41-44

E-mail: kai.rossdeutscher@dps.de

Inhaltsverzeichnis

Abbildungsverzeichnis	III
List of Listings	IV
1 Einleitung	1
1.1 Überblick	1
1.2 Motivation	1
1.3 Zielsetzung	1
2 Zielsetzung	2
2.1 Problemstellung	2
2.2 Lösungsweg	2
3 Ist-Analyse	3
3.1 JBoss (Microprofile)	3
3.2 Probleme	3
4 Vorgehensmodell	4
4.1 Herleitung Analyseparametern	4
4.1.1 Kriterienkatalog	4
4.2 Anforderungen an Prototypen	4
4.2.1 Festlegung fiktiver Workflow	4
4.3 Anfoderungen an Containerisierungsplattform	4
4.4 Anforderungen an Lasttest	4
5 Problemlösung	5
5.1 Festlegung konkreter Implementierung	5
5.1.1 Containerisierung	5
5.1.1.1 Docker Swarm	5
5.1.1.2 Container Lifecycle	5
5.1.2 Business Logik	5
5.1.2.1 Node.js	5
5.1.2.2 Spring Boot	5
5.1.3 Durchführung Lasttest	6
5.1.4 Timeline	6
5.1.5 Testbedingungen	6
5.1.5.1 Server Specs	6
6 Auswertung der Ergebnisse	7
6.1 Begründung Startupzeit	7
7 Zusammenfassung	8
8 Ausblick	9
9 Literaturverzeichnis	10
10 Eidesstattliche Erklärung	11

Abbildungsverzeichnis

List of Listings

1

Einleitung

1.1 Überblick

- Was gibt es bisher, nur kurz anreissen
- wird im Grundlagen Kapitel tiefergehender behandelt...

1.2 Motivation

- Problemstellung kurz beschreiben
- DPS möchte untersuchen ...

1.3 Zielsetzung

- Welche Erkenntnisse sollen gewonnen werden?
- Kriterienkatalog nennen

2

Zielsetzung

2.1 Problemstellung

- Vergleich der Startup-Zeit einer containerbasierten Cloudanwendung bei steigender Last. Es soll auf Unterschiede zwischen Spring und Node eingegangen werden.
- insbesondere auf Durchsatz der Instant Payments eingehen

2.2 Lösungsweg

- Prinzip / Ablauf erklären
- Grundlegende Struktur des Prototypen erklären

[?]

3

Ist-Analyse

3.1 JBoss (Microprofile)

- aktuelle Architektur beschreiben
- Hinweis darauf geben, dass Prototyp in der Thesis vereinfacht mit Spring dargestellt wird

3.2 Probleme

- Probleme mit aktuellem System (Stichwort Deployment, Wartbarkeit)
- Prof. hat extra darauf hingewiesen, dass es nicht nur um die Vorteile der Cloud gehen soll

4

Vorgehensmodell

4.1 Herleitung Analyseparametern

- Problem reduzieren
- Messkriterien festlegen
- Quellen finden

4.1.1 Kriterienkatalog

- Vorallem Skalierbarkeit und Performance
- Transaktionen - Durchsatz wichtig
- Deployment egal, da bereits im Container deployed wird
- Generell sagen, warum einige Aspekte egal sind
- Wartbarkeit nicht so wichtig
- Ressourcennutzung
- Störungsfälle (Chaos Monkey)
- Non Functionals - Kai hat PDF geschickt

4.2 Anforderungen an Prototypen

4.2.1 Festlegung fiktiver Workflow

4.3 Anforderungen an Containerisierungsplattform

4.4 Anforderungen an Lasttest

5

Problemlösung

5.1 Festlegung konkreter Implementierung

5.1.1 Containerisierung

5.1.1.1 Docker Swarm

5.1.1.2 Container Lifecycle

- Auf verschiedene Schichten eingehen
- Auf Ergebnisse beziehen

5.1.2 Business Logik

5.1.2.1 Node.js

5.1.2.2 Spring Boot

- Prototypen im Detail erläutern

```
cat .env
```

```
# qbn: queue bound (level) n  
# cbn: container bound (level) n
```

```
QB0=15  
QB1=30  
QB2=100  
CB0=1  
CB1=5  
CB2=10  
CB3=30
```

5 Problemlösung

Tabelle 5.1: Server Specs

Prozessor	Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz
Kerne	6 Prozessoren á 16 Kerne
RAM	16 GB
Storage	150 GB

$\underline{QL3}$ $QB2 < MC$	UP $abs(CB0 - CB3)$	UP $abs(CB1 - CB3)$	UP $abs(CB2 - CB3)$	OK –
$\underline{QL2}$ $QB1 < MC \leq QB2$	UP $abs(CB0 - CB2)$	UP $abs(CB1 - CB2)$	OK –	DOWN $abs(CB2 - CB3)$
$\underline{QL1}$ $QB0 < MC \leq QB1$	UP $abs(CB0 - CB1)$	OK –	DOWN $abs(CB1 - CB2)$	DOWN $abs(CB1 - CB3)$
$\underline{QL1}$ $QB0 < MC \leq QB1$	UP $abs(CB0 - CB1)$	OK –	DOWN $abs(CB1 - CB2)$	DOWN $abs(CB1 - CB3)$
$\underline{QL0}$ $QB0 == MC$	OK –	DOWN $abs(CB0 - CB1)$	DOWN $abs(CB0 - CB2)$	DOWN $abs(CB0 - CB3)$
	$\underline{CL0}$ $CB0 == MC$	$\underline{CL1}$ $CB0 < MC \leq CB1$	$\underline{CL1}$ $CB1 < MC \leq CB2$	$\underline{CL1}$ $CB2 < MC \leq CB3$

5.1.3 Durchführung Lasttest

5.1.4 Timeline

5.1.5 Testbedingungen

- Kommt in den Anhang
- hat Prof. zwar als eigenes Kapitel erwahnt, bin mir aber nicht sicher ob das wirklich noetig ist
- auf welcher Hardware werden Tests durchgefuehrt?
- chaos monkey / Stoerfaelle erlaeutern

5.1.5.1 Server Specs

6

Auswertung der Ergebnisse

6.1 Begründung Startupzeit

- Warum Node.js schneller ist

7

Zusammenfassung

8

Ausblick

9

Literaturverzeichnis

10

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Ort, Datum

Silas Hoffmann