# Projektbericht Cloud Computing

### Motivation

Die IT Branche ist im ständigen Wandel und regelmäßig setzen sich neue Lösungsansätze durch. In den letzten Jahren sind Microservice Architekturen, Automatisierungspipelines, Containerisierung und Orchestrierung Begriffe die überall auftauchen. Oftmals dauert es jedoch bis die neuen Konzepte in den Unternehmen so umgesetzt werden wie sie gedacht sind. Dies mussten wir auch im Betreuten Praktischen Semester (BPS) schon erfahren.

Monolithische Programme sollen immer öfter durch Microservice Architekturen abgelöst werden. Jedoch gibt es häufig noch Abhängigkeiten zwischen den Services die so nicht bestehen sollten. Auch bei CiCd Pipelines gehen oftmals nicht über die Tests hinaus.

Aber insbesondere bei der Containerisierung und der damit verbundenen Orchestrierung fehlt meistens noch viel Expertise. So wurden zum Beispiel in Projekten während des BPS in Memory Datenbanken verwendet anstatt Docker Container zu verwenden für die Datenbank.

Neben dem nicht Einsatz der Technologien ist aber auch der falsche Einsatz häufig ein Problem. Wie zum Beispiel der Einsatz von Kubernetes weil es im Trend ist obwohl die Skalierung für die Anwendung überhaupt kein Faktor ist, weil die Nachfrage viel zu klein ist. In solchen Fällen werden dann manchmal einzelne Pods in Kubernetes Deployed und der nutzen geht komplett verloren.

Es ist somit auch nicht sonderlich überraschend, dass DevOps Spezialisten aktuell extrem gefragt sind und die Position zu den Bestbezahltesten in der Industrie gehört.

Deshalb ist unsere Motivation und unser Ziel mit diesem Projekt einen breiten Einblick in alle Schritte von Microservices über CiCd Pipeline bis zum Deployment mit Containern und Kubernetes zu bekommen.

### Definition Problemstellung

Für das Projekt soll eine einfache Microservice Architektur umgesetzt werden. Da die Funktionalität der Services nicht im Vordergrund steht in diesem Projekt werden diese sehr einfach gehalten. Die Services werden in Go implementiert. Man kann gut Argumentieren das Go die beste Sprache für Microservices ist.

Für die Datenbanken wird MongoDB verwendet. Diese werden mit MongoDB Atlas deployed.

Jeder Service soll eine vollständige CiCd Pipeline besitzen. Hierfür sollen im ersten Schritt alle Tests erfolgreich durchlaufen und die Code Coverage mit einem Threshhold von 75% überschritten werden.

Im zweiten Schritt wird von jedem Service ein Dockerimage erzeugt und dieses wird auf Dockerhub gepusht.

Mit diesen Images kann dann das Deployment mit Kubernetes erfolgen.

### Lösungsansatz

In dem Projekt wurden 2 einfache Services implementiert. Ein User Service und ein Produkt Service. Diese sollen simple Funktionalitäten bieten die jeder Online Shop braucht.

Der User Service kann User in der Datenbank speichern, abfragen und löschen.

Der Product Service kann Produkte in der Datenbank speichern, abfragen und löschen.

Die Probleme bei der Umsetzung kamen in erster Linie aufgrund der Verwendung der Programmiersprache Go. Da wir noch nicht mit der Programmiersprache vertraut waren mussten wir uns erst in die Sprache und die Tools einarbeiten. Go ist eine Kompilierte, statisch typisierte Sprache die im Auftrag von Google entwickelt wurde. Die Programmiersprache ist recht einfach im Grundsatz und besitzt nur 25 Keywords. In Go gibt es keine Klassen oder Vererbung, lediglich Structs denen Methoden zugeordnet werden können. Durch Interfaces lässt sich mit Composition arbeiten. Sie wurde 2009 veröffentlicht und versucht viele Probleme zu lösen die bei älteren Sprachen die Typisch für Services eingesetzt werden, wie zum Beispiel Java, vorkommen. Während man bei der Service Entwicklung in Java auf Frameworks wie SpringBoot oder Quarkus zurückgreift und dort einen hohen Overhead hat, sind alle notwendigen Funktionalitäten bereits nativ in Go vorhanden. Dennoch haben wir uns dafür entschieden mit der Gin Library zu arbeiten. Dadurch spart man sich etwas Komplexität auf Kosten der Laufzeit.

Ein Problem das viel Zeit in Anspruch genommen hat, war die Methode zur Serialisierung und Deserialisierung von JSON und BSON. Go besitzt Pointer jedoch keine Pointer Arithmetik. Bei der Deklaration von Methoden kann man diese für Pointer oder Values des Objektes definieren. Definiert man eine Methode für einen Objekt Pointer so ist diese aber auch lediglich bei Pointern bekannt und kann von Object Values nicht aufgerufen werden. Wir haben versucht diese Methode über ein Object Value zu nutzen und diesen Fehler lange nicht gefunden.

######################

### Implementierung inklusive Schwierigkeiten

#### Microservices

Da die Microservices sehr einfach gehalten wurden kam es hier zu wenig Schwierigkeiten.

#### Pipeline

### Bewertung

##################

### Fazit

###############

## Anhang

### Architekturschaubild

###############

### Source Code

<https://github.com/JulianTeschner/cc_product_service>

<https://github.com/JulianTeschner/cc_user_service>

https://github.com/JulianTeschner/go\_test\_build\_action

### Beitrag Simon

###############

### Beitrag Julian

###############