

Cloud Computing
Sommersemester 2021
Projektarbeit

Abgabe: 21.07.2021

Julian Fischer

Personenerkennung und -anonymisierung auf einem Zeitraffer-Bilderstrom

- Kontext
 - Ein Kamerasystem erstellt einen kontinuierlichen Strom an Bildern zwecks Erstellung von zugehörigen Zeitraffervideos
 - Aus DSGVO Gründen ist eine Anonymisierung etwaig aufgenommener Personen notwendig
 - Ein bestehendes open source Projekt ist zu erweitern.
- Gegeben
 - ist ein Container Image zur Ausführung der Bilderkennung
 - eine Menge an Beispiel-Bildern
- Aufgabe
 - Erstellen Sie ein Message-Queue StatefulSet unter Verwendung von RabbitMQ
 - Erstellen Sie eine StatefulSet für eine Datenbank zur Speicherung von Metadaten
 - Beispiele: MySQL, PostgreSQL, etc.
 - **erstellen Sie hierzu ihr eigenes Container Image**
 - Schreiben Sie einen Web-Service (off-topic), der
 - einen Endpunkt zur Verfügung stellt und eingehende Bilder per HTTP Post Request annimmt
 - Eingehende Bilder werden in einem ObjectStore (z.B. S3 Bucket) abgelegt.
 - Speichern Sie die Metadaten, insbesondere die URL des abgelegten Bildes in eine Datenbank
 - Für jedes eingehende Bild legen Sie eine Nachricht in die Message Queue, die das zu bearbeitende Bild per URL referenziert
 - Schreiben Sie einen Worker,
 - der für jede Nachricht in der Warteschlange die Nachricht abrufen,
 - das Bild vom Object-Store lädt
 - Die Objekterkennung / Faceblur (Anonymisierung) darauf anwendet und in einer neuen Bilddatei speichert
 - Speichern Sie das anonymisierte Bild auf S3 in einem Ausgabe-Bucket
 - Löschen das ursprüngliche Bild
 - Verwenden Sie die Acknowledge-Funktion von RabbitMQ, um sicherzustellen dass unvollständig abgearbeitete Jobs in der Queue verbleiben und nur vollständig Abgearbeitete Jobs aus der Queue entfernt werden.
 - Sämtliche Dienste und Abläufe sind in Kubernetes Ressourcen zu beschreiben.
 - Web-Anwendungen und Worker sind zustandslos auszuführen, so dass problemlos mehrere Instanzen parallel betrieben werden können.
 - Instanzieren Sie mindestens zwei Instanzen der Web-Anwendung und zwei Worker, um die Zustandslosigkeit unter Beweis zu stellen.
 - **Nur 3er Gruppen: Installieren und Konfigurieren Sie ein Logging-Framework, um das gesamte System zu überwachen (Fluentd + Open Distro)**

- Abnahme
 - Demonstrieren Sie die Funktionsweise des Systems
 - Eine Bewertung des Quelltextes der Anwendung selbst findet nicht statt
 - Nur die Kontaktfläche mit Kubernetes wird bewertet, so z.B. ob die Anwendung zustandslos ist und z.B. Konfigurations- und sicherheitsrelevante Daten korrekt gehandhabt werden.
 - Erläutern Sie ihre Umsetzung mit besonderem Hinblick auf die Verwendung von Kubernetes
 - Diese Umsetzung ist primärer Gegenstand der Bewertung
 - Demonstrieren Sie die horizontale Skalierung
 - des Web-Services
 - der Worker
 - Demonstrieren Sie das Selbstheilungsverhalten des Systems, in dem Sie Pods der folgenden Komponenten löschen
 - Web-Service
 - Worker
 - StatefulSets
 - Erläutern Sie das zu beobachtende Verhalten.
 - Demonstrieren Sie das Ausbringen einer neuen Anwendungsversion
 - des Web-Services
 - inklusive einer Schema-Migration der Datenbank
 - Erläutern Sie ihr Vorgehen sowie die zu lösenden Herausforderungen
 - Erläutern Sie, ob und wie die Datendienste (RabbitMQ und PostgreSQL) horizontal skaliert werden können.
 - Erläutern Sie die Herausforderungen, die dabei entstehen.
 - Tipp: Studieren sie hierzu die Betriebsanleitung der Datendienste.
 - Die Erläuterung muss keine vollständige Umsetzung skizzieren, sondern die grundsätzliche Herausforderung hervorheben.
- Hinweise
 - Verwenden Sie **minikube** als lokale Kubernetes Lösung
 - Verwenden Sie eine lokale Installation von minio als lokalen Object Store
 - <https://gist.github.com/balamurugana/c59e868a36bb8a549fe863d22d6f0678>
- Optional
 - Hochverfügbare Message Queue (RabbitMQ)
 - Erstellen Sie ein StatefulSet mit 3 Replicas, die sich zu einem hochverfügbaren Verbund zusammenschließen.
 - Stellen Sie die verwendete Message Queue ihrer Anwendung so ein, dass diese Gebrauch vom Message-Queue Cluster macht.