

# Cloud-Computing Projektarbeit

Im Rahmen der Vorlesung "Cloud Computing" lernen Sie die Grundlagen der Cloud im Allgemeinen, die Programmierung von Cloud-nativen Anwendungen sowie weiterführende Konzepte für einen reibungslosen Betrieb. Die Projektarbeit soll das Verständnis der theoretischen Konzepte durch die praktische Anwendung im Kleinen festigen. Erstellen Sie in Gruppenarbeit eine Anwendung, die den folgenden Anforderungen genügt.

## Anforderungen

1. Die Anwendung veröffentlicht ein REST-API mit folgender Struktur:

- **/api/home/troets?limit=10&offset=20&troeter=some\_account\_name**  
Die Schnittstelle gibt alle Troets der Home-Timeline eines Mastodon-Accounts im JSON-Format zurück. Der Query-Parameter **limit** ist optional und schränkt die Anzahl an zurückgegebenen Mastodon-[Status](#)-Objekten ein. Der Query-Parameter **offset** ist optional und gibt an, ab welchem Offset Troets ausgegeben werden sollen. Der Query-Parameter **troeter** ist optional und filtert die Troets auf den Accountnamen. Es sollen nur noch Troets zurückgegeben werden, die von diesem Account gepostet oder geteilt wurden.
- **/api/home/troets/reblogged?troeter=some\_account\_name**  
Die Schnittstelle gibt alle geteilten (reblogged) Troets der Home-Timeline eines Mastodon-Accounts im JSON-Format zurück. Der Query-Parameter **troeter** ist optional und filtert die Troets auf den Accountnamen. Es sollen nur noch Troets zurückgegeben werden, die von diesem Account gepostet oder geteilt wurden.
- **/api/home/troets/direct?troeter=some\_account\_name**  
Die Schnittstelle gibt alle Einträge der Home-Timeline eines Mastodon-Accounts im JSON-Format zurück, die nicht geteilt (reblogged) wurden. Der Query-Parameter **troeter** ist optional und filtert die Troets auf den Accountnamen. Es sollen nur noch Troets zurückgegeben werden, die von diesem Account gepostet oder geteilt wurden.
- **/api/home/troeters**  
Die Schnittstelle gibt alle Accounts im JSON-Format zurück, die auf der Home-Timeline eines Mastodon-Accounts einen Troet gepostet oder geteilt haben.

2. Die Anwendung ist containerisiert und als solche ausführbar.

3. Der Container aus 2. läuft lokal in dem Kind-Cluster aus der [Einführungs-Übung](#).

4. Die Anwendung nutzt die folgenden Kubernetes-Ressourcen, welche über [Tilt](#) in den Cluster deployed werden:

- Deployment - es sollen 2 Replicas laufen
- Probes
  - Startup-Probe
  - Liveness-Probe
  - Readiness-Probe
- Service - alle Anwendungs-Pods sollen über einen Service erreichbar sein

- Ingress - die Anwendung und deren API soll von außerhalb des clusters (nur **lokal!**) unter **/api/cc-service/** erreichbar sein.
- PodDisruptionBudget - es muss immer mindestens ein Pod am laufen bleiben
- ConfigMaps & Secrets - die Konfiguration der Anwendung wird in ConfigMaps abgelegt, geheime Daten in Secrets.

5. Die Anwendung veröffentlicht einen Endpunkt für Metriken (/metrics). Es sollen für alle implementierten APIs die Aufrufzahlen veröffentlicht werden.

6. Die Anwendung bindet für den Zugriff auf Mastodon den [Troetbot](#)-Service an. Der Troetbot bietet einen bequemen Zugriff auf die Mastodon-Home-Timeline über ein REST-API im JSON-Format. Der Troetbot-Service soll im gleichen Kind-Cluster laufen und ist über einen Kubernetes-Service anzusprechen.

Gruppen mit einer Größe von 4 Personen müssen diese API nicht implementieren:  
**/api/home/troets/direct?troeter=some\_account\_name**

## Präsentation

Die Anwendung soll den anderen Teilnehmenden in einer 20-minütigen Präsentation vorgestellt werden. Geht mindestens auf die folgenden Punkte ein:

- Welche Programmiersprache und Frameworks/Libraries wurden verwendet?
- Wie ist der Container aufgebaut?
- Die Kubernetes-Konfigurationen
- Gab es Probleme oder etwaige Besonderheiten? Wenn ja, welche und warum?
- Live-Demo der API

Der vorläufige Präsentationstermin ist der 21.12.2023.

Die Präsentation wird nicht benotet. Das Bestehen ist jedoch Voraussetzung zum Bestehen des Moduls.

## Ausschlüsse

Es wird keine UI erwartet!

Die Anwendung darf in einer beliebigen Programmiersprache implementiert werden.

Abhängig von der Wahl können wir aber ggf. schlecht bis kaum unterstützen!

## Voraussetzungen

Es wird ein Mastodon-Account benötigt, um die Trötbot API nutzen zu können.

## Coaching

Wir unterstützen und beantworten etwaige Fragen in den Übungsstunden.