

Aufgabe 2: Kafka

Auch für diese Aufgabe benötigen Sie wieder einen VPN-Zugang.

Der Kafka-Server der DHBW ist erreichbar unter `10.50.15.52:9092`. Kafka ist eine weitere Möglichkeit, ereignisbasierte Datenströme zu speichern. Es bietet ebenfalls ein Subscriber-/Listener-Schema, das hier Producer/Consumer heißt.

Bitte geben Sie ein kurzes Protokoll der Bearbeitung der Aufgaben als PDF-Datei ab (mit Antworten auf die Fragen). Im Protokoll können Sie auch gerne aufgetretene Probleme und deren Lösung beschreiben. Zusätzlich geben Sie bitte Ihre Quelldateien als Zip-Datei ab. Sie können dabei die gesamte Abgabe in einer einzigen Zip-Datei ablegen. Die Abgabe kann in Gruppen von 3-4 Personen im Moodle-Abgabecontainer erfolgen.

(a) Erste Schritte mit Kafka.

Bitte schauen Sie sich folgende Dokumente an:

- Einführung zu Kafka
- Quickstart. Hierfür können Sie gerne schon den DHBW-Server benutzen. Clients werden weiter unten beschrieben. Bitte legen Sie zum Testen Topics mit Präfix `test_` an.
- Kafka-Doku mit API-Dokumentation.
- **Bitte löschen Sie keine Topics auf dem DHBW-Server!**

Sie können als Client beispielsweise `kafkacat/kcat` verwenden, oder alternativ die Kafka-Distribution herunterladen. Diese enthält auch Kommandozeilen-Clients, die unter Windows und Linux lauffähig sind.

Beantworten Sie folgende Fragen:

- Was sind Gemeinsamkeiten und Unterschiede zwischen Kafka und MQTT-Brokern?
- Für welche Szenarien ist Kafka besser geeignet?
- Man findet bei Vergleichen öfters die Aussage, Kafka würde das Modell "Dumb Broker / Smart Consumer" implementieren, während bei MQTT "Smart broker / Dumb Consumer" gilt. Was ist damit gemeint? Was muss ein Kafka-Consumer beachten?
- Was sind Partitionen? Für was kann man sie neben Load-Balancing noch verwenden?

(b) Die allseits beliebten Wetter-Daten mal wieder.

Unter dem Topic `weather` können Sie die Wetterdaten für Mosbach, Mergentheim und Stuttgart abfragen. Schreiben Sie einen Kafka-Client, der die Wetterdaten via Kafka auslesen und auf der Konsole ausgeben kann. Sie können hierzu gerne die Java-basierte Consumer-API verwenden, oder aber jeweilige Client-Implementierungen für andere Programmiersprachen / Frameworks.

- Implementieren und testen Sie ihren Consumer.
- Geben Sie die Quellen mit in Ihrem Archiv ab.

(c) Eigener Producer.

Schreiben Sie einen eigenen Producer. Hierfür wählen Sie am besten ein Topic nach dem folgenden Schema:

```
vlvs_inf19(...) _eindeutiges_kuerzel_oder_gruppenname
```

Die Daten können Sie beliebig wählen, solange diese visualisierbar sind. Beispielsweise können Sie die CPU-Load oder gerne auch kreativere Werte nehmen (in Java via `ManagementFactory.getOperatingSystemMXBean`). Alternativ können Sie dafür auch die Faker-Library nehmen.

- Implementieren und testen Sie ihren Producer.
- Geben Sie die Quellen mit in Ihrem Archiv ab.
- Wozu dienen Key und Value beim Versenden von Kafka-Messages?

(d) Kafka-Pipeline mit Grafana.

Neben der Weiterverarbeitung von Daten in einer Pipeline können diese zusätzlich auch einfach visualisiert werden. Hierfür wird oft Grafana verwendet.

- Schauen Sie sich das Tutorial zu Grafana an.
- Benutzen Sie das DHBW-Grafana unter <http://10.50.15.52:3000>. Login bitte bei mir via Mail erfragen, oder (besser) via privatem Chat in BBB. Bitte nur eine Person pro Gruppe (Sie können die Accountdaten selbstverständlich dann in der Gruppe teilen).
- Oder installieren Sie sich lokal eine Grafana-Instanz, Pro Gruppe dürfte nur jeweils eine nötig sein. Die einfachste Möglichkeit ist die Installation via Docker.
- Ziel ist, die Wetter-Daten via Grafana zu visualisieren für die 3 Standorte.
- Wie bekommen wir die Daten nun vom Broker ins Grafana?

Prinzipiell können Sie gerne auch einen anderen Weg gehen, als den in der Aufgabe vorgeschlagenen. Eine Alternative wäre beispielsweise ein eigener PostgreSQL- oder MySQL-Server, oder auch Prometheus.

Noch einfacher geht es allerdings mit Graphite, einer Datenbank für Zeitseriendaten.

- Machen Sie sich mit Graphite vertraut.
- Schauen Sie sich insbesondere an, wie Daten in Graphite abgespeichert werden.
- Wir müssen umdenken! Einfach mal JSON reinfüttern wird nicht gehen.
- Überlegen Sie sich eine sinnvolle Namenshierarchie.
- Für erste Schritte können Sie gerne die auf dem Kafka-Server installierte Graphite-Instanz nehmen.

Diese ist unter <http://10.50.15.52/> zu erreichen. Falls Sie jedoch eine eigene Grafana-Instanz haben, empfiehlt sich auch hier eine eigene Installation, da Sie ansonsten die Daten mehrfach von der DHBW herunter- und wieder hochladen müssen (Kafka → eigener Client → Graphite → Grafana). Wenn Sie dies nicht stört, können Sie den Server gerne benutzen.

Bitte benutzen Sie dann dafür ein geeignetes Namensschema, beispielsweise eine Kombination aus Kursname und Gruppennummer als Präfix, um Kollisionen zu vermeiden.

Aufgabe: Schreiben Sie einen Kafka-Consumer, der die Daten an Ihre Graphite-Instanz übergibt.

Für Java können Sie beispielsweise das `metrics-graphite`-Paket von Dropwizard zum Speichern von Daten in Graphite benutzen.

Um Bandbreite zu sparen können Sie Ihren Consumer auf dem Epyc-Server laufen lassen. Dieser ist via SSH unter `10.50.15.120` zu erreichen. Derzeit sind dort Python, Java und Mono installiert – bei Sonderwünschen bitte bei mir melden.

Benutzer: `vl-vs`, Passwort: `oRnYDnjbqB`

Hinweis: ein Consumer muss nicht immer wieder am Anfang des Topics mit Lesen beginnen. Sie können ein Offset abspeichern (`commit`), damit die Arbeit das nächste Mal dort weitergehen kann. Hierzu ist es sinnvoll, einer Consumer-Gruppe beizutreten. Wählen Sie möglichst eine eindeutige Gruppennummer für Ihren Client.

Sie können sich überlegen, ob Sie die Daten alle direkt an Graphite weitergeben, oder sie vorher sinnvoll aggregieren. Da die Wetterdaten eigentlich nur alle 10 Minuten aktualisiert werden, kann hier eine Filterung sinnvoll sein.

Sobald die Daten sinnvoll in Graphite stehen, kann in Grafana eine Data Source für den Zugriff konfiguriert werden. Nun können Sie die Temperaturdaten einfach visualisieren! Graphite bietet zwar selbst eine Visualisierungsmöglichkeit – diese ist aber eher sehr einfach gehalten.

- Geben Sie den Code Ihres Kafka-Consumers ab.
- Geben Sie ebenfalls einen Screenshot Ihres Grafana-Dashboards ab (**mit Gruppen-Id, bzw. Matrikelnummern im Titel des Graphen**).