

Agenda für ein Seminar “Apache Kafka”

Dr. Rainer Sawitzki, 25.4.2023

Dauer: 2 Tage, 8 Blöcke mit jeweils 90 Minuten. 9:00 - 16:15

Termin: 4./5.5.2023

Ort: Cegos Integrata Frankfurt

Zielgruppe: Entwickler, Architekten

Vorkenntnisse: Grundkenntnisse in (Micro-)Services Architekturen sind von Vorteil, Kenntnisse von Java, Spring, Spring Boot, Grundkenntnisse der Überwachung von Java-Prozessen sind von Vorteil

Übungen: Praktische Übungen sind vorgesehen, Beispielanwendungen werden bereitgestellt

Methode:

- Vortrag, Präsentation, Diskussion, eigenständige Übungen

Sprache: Deutsch

Ressourcen: Beispiele werden über GitHub bereitgestellt, PDF-Handout, digitales Flipchart

Inhalte:

- Einführung Kafka
 - Warum ein neues System? Abgrenzung zu Messaging-Systemen
 - Installation (Standalone, Docker)
 - Werkzeuge
 - Anwendungsgebiete: Tracking, Monitoring, Logging, Statistik & Stream Processing
- Der Kafka-Cluster
 - Horizontale Skalierung im Ring-Cluster
 - Umsetzung mit Kafka und Zookeeper / Kraft
 - Konsequenzen aus dem CAP-Theorem
 - Ausfallsicherheit
- Kafka Messaging
 - Producer und Consumer
 - Topics im Detail
 - Partitionen, Skalierung von Producer und Consumer
 - Schema-Definition, Avro im Detail
 - Garantien zur Nachrichten-Übertragung
 - Von No Guarantee bis Effectively Once
 - Retry
 - Reihenfolge der Nachrichten
- Überwachung
 - Kafka als Java-Prozess
 - JMX-basierte Überwachung
 - Prometheus-Exporter
 - Relevante Metriken
- Best Practices
 - Anzahl Partitionen
 - Größe der Nachrichten
 - Verschlüsselung
 - Administration
 - Anlegen von Topics
 - Log-Files
 - Quotas
- Übersicht weiterer Kafka-Frameworks
 - Kafka-Integration in Spring / Spring Boot
 - Connect
 - Streams

Vom Kunden gesendete Themenliste, 26.4.2023

- Einführung (Abhängig von den Teilnehmern, Normalerweise arbeiten schon alle mit Kafka und kennen notwendige Grundlagen. Eventuell in der Vorstellungsrunde drüber sprechen, wie viel Grundlagen notwendig sind)
 - Batching & Streaming
 - Real-Time Data Feeds
 - Anwendungsgebiete: Tracking, Monitoring, Logging, Statistik & Stream Processing
 - Kafka Grundlagen: Nachrichten, Schemata, Topics, Partitionen, Producer / Consumer, Broker
- Architektur
 - Persistenz
 - Caching im Memory und Pagecache
 - Event Sourcing mit Kafka
 - Schemaverwaltung: Avro, Schema-Evolution, Serialisierung / Deserialisierung
 - Consumer Groups, Exactly-Once-Processing, (Wir Machen At-Least-Once-Processing, vielleicht kann man darüber auch etwas sprechen)
 - Als Consumer vorschreiben, wie ein Schema (Avro) aussieht
- Clustering
 - Clustering Grundlagen: Verfügbarkeit, Ausfallsicherheit, Lastverteilung und Skalierbarkeit
 - Die Zookeeper Registry (Wird/Ist abgekündigt?)
 - Aufbau eines Clusters
 - Loadbalancing
 - Heartbeats
 - Replication & Fault Tolerance
 - Was passiert beim Ausfall eines Knotens im Cluster?
- Stream Processing
 - Was ist ein Stream?
 - Java 8 Grundlagen: Lambdas (abhängig von Teilnehmern)
 - Kafka Streams
 - Beschreiben von Transformationen mit der KStream DSL
 - Kafka-Unterstützung in Spring, Spring Cloud Stream
- Sicherheit/Nachrichtenübermittlung
 - Garantien für Reihenfolge und Ausfallsicherheit
 - Machs nochmal Sam: Retries
 - Verschlüsselte Verbindungen mit SSL/TLS (Konfiguration/Entwicklersicht keine Administration)
- Administration
 - Anlegen und Ändern von Topics (In der dwpbank nutzen wir Kafka Julie, keine Anlage via Admin Client etc. möglich)
 - Log Compaction
 - Skalierung
 - Automatische Bereinigung des Logs
 - Limitieren mit Quotas
- Performance (Entwicklungssicht wäre interessant, was kann man im Code tun?)
 - Performance-Tuning von Kafka
 - Kompression von Nachrichten
 - Anz. Partitionen
 - Größe von Nachrichten

- Wie konfiguriere ich meinen Consumer/Producer korrekt?
 - Optimale Anzahl von Knoten
- Erweiterungen
 - Stream Processing mit Spark (Nutzen wir noch nicht)
 - Import und Export von Daten mit Kafka Connect (MQ, JDBC wird aktuell genutzt)