

Mit einem Geleitwort von  CONFLUENT



Anatoly
ZELENIN
Alexander
KROPP

Apache Kafka

Von den Grundlagen
bis zum Produktiveinsatz

HANSER

Zelenin/Kropp
Apache Kafka



Bleiben Sie auf dem Laufenden!

Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter:
www.hanser-fachbuch.de/newsletter



Anatoly Zelenin
Alexander Kropp

Apache Kafka

Von den Grundlagen
bis zum Produktiveinsatz

HANSER

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso übernehmen Autor und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2022 Carl Hanser Verlag München, www.hanser-fachbuch.de

Lektorat: Sylvia Hasselbach

Copy editing: Sandra Gottmann, Wasserburg

Umschlagdesign: Marc Müller-Bremer, München, www.rebranding.de

Umschlagrealisation: Max Kostopoulos

Titelmotiv: © Max Kostopoulos

Satz: Eberl & Kösel Studio GmbH, Altusried-Krugzell

Druck und Bindung: Eberl & Kösel GmbH, Altusried-Krugzell

Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702

Printed in Germany

Print-ISBN: 978-3-446-46187-1

E-Book-ISBN: 978-3-446-47046-0

E-Pub-ISBN: 978-3-446-47265-5

Inhalt

Geleitwort	XI
Über das Buch	XV
1 Einleitung	1
1.1 Einführung in Apache Kafka	2
1.2 Erste Schritte mit Kafka	5
2 Konzepte	11
2.1 Payload	12
2.1.1 Topics	12
2.1.1.1 Anzeigen von Topics	12
2.1.1.2 Erstellen, Anpassen und Löschen von Topics	14
2.1.2 Nachrichten	17
2.1.2.1 Nachrichtentypen	17
2.1.2.2 Datenformate	19
2.1.2.3 Keys und Values	19
2.1.3 Zusammenfassung	21
2.2 Kafka als verteiltes Log	22
2.2.1 Logs	22
2.2.1.1 Was genau ist eigentlich ein Log?	22
2.2.1.2 Grundlegende Eigenschaften eines Logs	23
2.2.1.3 Die Rolle des Logs in Kafka	24
2.2.2 Verteilte Systeme	26
2.2.2.1 Partitionierung	27
2.2.2.2 Consumer Groups	30
2.2.2.3 Replikation	33

2.2.3	Komponenten	36
2.2.3.1	Koordinationscluster	37
2.2.3.2	Broker	38
2.2.3.3	Clients	38
2.2.4	Kafka im Unternehmenseinsatz	38
2.2.5	Zusammenfassung	41
2.3	Zuverlässigkeit	41
2.3.1	Acknowledgements	42
2.3.1.1	Acknowledgement-Strategien in Kafka	43
2.3.1.2	Acknowledgements und ISR	44
2.3.1.3	Nachrichten-Zustellungsgarantien in Kafka	47
2.3.2	Replikation	49
2.3.2.1	Replikation vs. Backup	49
2.3.2.2	Leader-Follower-Prinzip	50
2.3.3	Zusammenfassung	55
2.4	Performance	56
2.4.1	Topics	58
2.4.1.1	Partitionierung und Keys	58
2.4.1.2	Skalierung und Lastverteilung	59
2.4.1.3	Wie viele Partitionen sollten wir haben?	60
2.4.1.4	Verändern der Anzahl an Partitionen	61
2.4.2	Producer Performance	62
2.4.2.1	Producer-Konfiguration	63
2.4.2.2	Producer-Performance-Test	64
2.4.3	Broker Performance	66
2.4.3.1	Die Aufgaben eines Brokers	66
2.4.3.2	Broker-Konfiguration und -Optimierung	67
2.4.4	Consumer Performance	68
2.4.4.1	Consumer-Konfiguration	68
2.4.4.2	Consumer-Performance-Test	69
2.4.5	Zusammenfassung	71
3	Kafka Deep Dive	73
3.1	Verbindung zu Kafka	74
3.2	Nachrichten produzieren und persistieren	75
3.2.1	Producer	76
3.2.1.1	Nachrichten produzieren	76

3.2.1.2	Serialisierung und Partitionierung	77
3.2.1.3	ACKs und deren Auswirkungen	78
3.2.2	Broker	79
3.2.2.1	Empfang und Persistierung von Nachrichten	80
3.2.2.2	Broker und ACKs	81
3.2.2.3	Optimierung	81
3.2.3	Daten- & Dateistrukturen	82
3.2.3.1	Metadaten, Checkpoints und Topics	82
3.2.3.2	Partitionen	83
3.2.3.3	Log-Dateien und -Indizes	86
3.2.3.4	Segmente	87
3.2.3.5	Gelöschte Topics	88
3.2.4	Replikation	89
3.2.4.1	In-Sync-Replicas	90
3.2.4.2	High Watermark	90
3.2.4.3	Auswirkungen von Verzögerungen bei der Replikation	92
3.2.5	Zusammenfassung	92
3.3	Nachrichten konsumieren	93
3.3.1	Consumer	93
3.3.1.1	Fetch-Request	94
3.3.1.2	Fetch vom nächsten Replica	94
3.3.2	Broker	95
3.3.3	Offsets	96
3.3.3.1	Verwaltung von Offsets	96
3.3.3.2	Praxisbeispiel Offsets	97
3.3.4	Consumer Groups	99
3.3.4.1	Funktionsweise von Consumer Groups	99
3.3.4.2	Kafka Rebalance Protocol	100
3.3.4.3	Verteilung der Partitionen auf die Consumer	101
3.3.4.4	Static Memberships	103
3.3.5	Zusammenfassung	104
3.4	Nachrichten aufräumen	104
3.4.1	Wieso müssen wir Nachrichten aufräumen?	105
3.4.2	Kafkas Aufräum-Methoden	105
3.4.3	Log Retention	106
3.4.3.1	Wann wird ein Log via Retention aufgeräumt?	107
3.4.3.2	Offset Retention	109

3.4.4	Log Compaction	109
3.4.4.1	Wann wird ein Log via Compaction aufgeräumt?	111
3.4.4.2	Wie funktioniert der Log Cleaner?	113
3.4.4.3	Tombstones	114
3.4.5	Zusammenfassung	115
3.5	Cluster-Management	115
3.5.1	Zookeeper-basiertes Cluster-Management	116
3.5.2	KRaft-basiertes Cluster-Management	118
3.5.3	Zusammenfassung	119
3.6	Verarbeitungsgarantien und Transaktionen	119
3.6.1	Idempotenz	120
3.6.2	Transaktionen	121
3.6.2.1	Transaktionen in Datenbanken	122
3.6.2.2	Transaktionen in Kafka	123
3.6.2.3	Transaktionen und Consumer	124
3.6.3	Zusammenfassung	125
4	Kafka im Unternehmenseinsatz	127
4.1	Kafka-Ökosystem	129
4.1.1	Kafka Connect	130
4.1.1.1	Wie funktioniert Kafka Connect?	130
4.1.1.2	Praxisbeispiel Distributed Mode	132
4.1.1.3	Skalierbarkeit und Ausfallsicherheit	134
4.1.2	Kafka Streams	135
4.1.2.1	Stream Processing	135
4.1.2.2	Wie funktioniert Kafka Streams?	136
4.1.2.3	KTables	138
4.1.3	Schema-Management	139
4.1.3.1	Schemas in Apache Kafka	139
4.1.3.2	Schemaanpassung und -kompatibilität	140
4.1.3.3	Schemaformate	141
4.1.3.4	Avro	141
4.1.3.5	Schema Registry	142
4.1.4	Sicherheit	143
4.1.4.1	Verschlüsselung	143
4.1.4.2	Authentifizierung und Autorisierung	144
4.1.4.3	Zookeeper	145

4.1.5	Desaster-Management	145
4.1.5.1	Was kann schon schiefgehen?	146
4.1.5.2	Backups in Kafka	147
4.1.5.3	Stretched Cluster	147
4.1.5.4	MirrorMaker	149
4.1.6	Zusammenfassung	152
4.2	Vergleich mit anderen Technologien	152
4.2.1	Klassische Messaging-Systeme	153
4.2.2	REST	155
4.2.3	Relationale Datenbanken	155
4.2.4	Kafka als Streaming-Plattform	157
4.2.5	Zusammenfassung	158
4.3	Kafka-Referenzarchitektur	159
4.3.1	Deployment-Modelle und Hardware-Anforderungen	159
4.3.1.1	Kafka auf eigener Hardware	160
4.3.1.2	Kafka in virtualisierten Umgebungen	160
4.3.1.3	Kafka in der Public Cloud selbst betreiben	160
4.3.1.4	Kafka in Kubernetes	160
4.3.1.5	Kafka as a Service	161
4.3.2	Broker	161
4.3.3	Koordinationscluster	162
4.3.4	Monitoring und Logging	162
4.3.5	Zusätzliche Werkzeuge	163
4.3.6	Zusammenfassung	164
5	Anhang: Kafka-Testumgebung aufsetzen	165
5.1	Betriebssysteme	166
5.2	Kafka herunterladen	166
5.3	Zookeeper starten	166
5.4	Kafka starten	168
5.5	Einzelne Broker stoppen	170
5.6	Umgebung aufräumen	170
	Register	171

Geleitwort

Technologie transformiert unseren Alltag und unsere Gesellschaft wie nie zuvor. Traditionelle, erfolgreiche Konzerne müssen sich neu erfinden und digitale Prozesse und Innovationen anbieten, um langfristig konkurrenzfähig zu bleiben. Ein wichtiger und stetig wachsender Teilbereich davon ist die Verarbeitung von Daten in Echtzeit: Ansonsten hat der Kunde bereits den Laden verlassen, bevor Empfehlungen oder Coupons via Push-Notification geschickt wurden. Ansonsten ist der Betrug passiert, bevor der Algorithmus warnen kann. Ansonsten ist die Maschine defekt, bevor die Anomalien der Sensoren erkannt wurden.

Diese Transformation geschieht über alle Industrien hinweg: Aus Läden erwachsen Webshops, Banken werden in Software-Services verwandelt, Taxibestellungen laufen über Software und auch Autos bestehen zu einem Großteil mittlerweile daraus. Wir können also zusammenfassend sagen: Unternehmen verwandeln sich in Daten-Infrastrukturen. Die Möglichkeit, diese Daten in Echtzeit verarbeiten zu können, ist essenziell für den zukünftigen Unternehmenserfolg. Eine moderne Infrastruktur für die Datenverarbeitung in Echtzeit ist Pflicht für den zukünftigen Unternehmenserfolg.

Paradigmenwechsel hin zu „Data in Motion“ mit Apache Kafka

Die Cloud ist die Zukunft für Datenzentren: flexibel, elastisch, skalierbar, immer auf dem neuesten Stand der Technik. Für die Datenverarbeitung findet mit Daten-Streaming gerade ein ähnlicher Paradigmenwechsel statt: Echtzeit, agil, skalierbar, Entkopplung verschiedener Anwendungen und Technologien.

Beim Daten-Streaming werden Daten kontinuierlich in Bewegung verarbeitet („Data in Motion“), anstatt erst alles in einer Datenbank, einem Data Warehouse oder einem Data Lake zu speichern, wo die Daten dann irgendwann (oftmals zu spät) in einem Batch-Prozess („Data at Rest“) verarbeitet werden.

Das Open Source Framework Apache Kafka hat sich dabei als De-facto-Standard für „Data in Motion“ etabliert. Vor über zehn Jahren wurde Kafka bei LinkedIn entwickelt, um sehr große Mengen von Logdaten („Big Data“) zu verarbeiten. Heute existieren zahlreiche verschiedene Anwendungsfälle für Kafka. Darunter fallen neben den analytischen Szenarien auch viele transaktionale Deployments wie Zahlungsplattformen, Trading-Apps, Retail-Anwendungen, Location-based Mobility Services, Supply Chain Management oder das operative Management von 5G Telco-Infrastrukturen.

Kafka ist so beliebt, weil es mehrere relevante Charakteristika bietet, die bisher auf verschiedene Frameworks oder Produkte verteilt waren: Messaging, Storage und Caching sowie Datenintegration (via Kafka Connect) und Datenverarbeitung (via Kafka Streams). Die Kombination aus hoher Skalierbarkeit, Zero Downtime und Zero Data Loss ermöglicht den Aufbau transaktionaler und analytischer Lösungen mit einer einzigen hochverfügbaren Infrastruktur. Dies differenziert Kafka von existierenden Messaging- (MQ), Integrations- (ETL/ESB) und Datenverarbeitungstools.

Kafka-Ökosystem: APIs, Tools, Automatisierung und SaaS

Die große weltweite Kafka Community erweitert das Kafka-Ökosystem um zusätzliche Skripte, Frameworks und kommerzielle Produkte. Die Schema Registry für Data Governance, ksqldb als SQL-Abstraktionsebene für Stream Processing, kafkacat als Command Line Utility, cloud-native Kubernetes-Operatoren oder APIs für Programmiersprachen jenseits von der JVM wie beispielsweise C++, Go oder JavaScript ermöglichen den Aufbau von Kafka als zentrales Echtzeit-Nervensystem für das Unternehmen. In der Cloud besteht sogar die Möglichkeit, den Betrieb komplett auszulagern, da ein Serverless SaaS wie Confluent Cloud dem Nutzer unter kritischen SLAs ermöglicht, sich auf seine Kafka-Anwendungen zu fokussieren.

Moderne Microservice- und Data-Mesh-Architekturen sind mittlerweile der Standard, um agile Entwicklung sowie einen automatisierten Betrieb von Anwendungen zu ermöglichen. Kafka ermöglicht – im Gegensatz zu REST und Webservice-Architekturen – eine richtige Entkopplung verschiedener Anwendungen mittels Domain-driven Design (DDD). Auch in Zukunft wird nicht jede Anwendung Echtzeit sein. Nicht nur Legacy-Systeme, sondern auch Reporting für Business Intelligence und Modeltraining für Machine Learning bleiben batch-basiert. Kafka ermöglicht Backpressure Handling und unterstützt dadurch sowohl Echtzeit- als auch Batch- und Request-Response-Schnittstellen mit einer einzigen Datendrehscheibe.

Hybride Architekturen mit mehreren Kafka-Clustern sind dabei nicht die Ausnahme, sondern die Regel. Dabei gibt es diverse Anwendungsfälle wie beispielsweise Disaster Recovery, Aggregation, Migration oder globale Projekte. Der Betrieb erfolgt in unterschiedlichen Infrastrukturen, beispielsweise im Data Center, in (Multi-)Cloud oder auch für Edge Computing in einer Fabrik, Maschine oder Drohne. Bidirektionale Replikation in Echtzeit erlaubt analytische als auch transaktionale Kommunikation mit all den Vorteilen des Kafka-Protokolls.

Beispiele aus der Industrie

Abschließend noch ein paar Beispiele für Kafka-Deployments aus verschiedenen Branchen für verschiedene Anwendungsfälle, um die Vielfältigkeit von Kafka zu veranschaulichen:

Die Bank NORD/LB baut ihre konzernweite IT-Transformation auf Kafka auf, um Szenarien wie Betrugserkennung, Kundenbindung und Handelsplätze zu optimieren

Die Versicherung Generali verwendet Confluent, um Hunderte von Legacy-Datenbanken via Change Data Capture (CDC) in der neuen Kubernetes-basierten Anwendungslandschaft zu integrieren.

Die Deutsche Bahn nutzt das Kafka-Ökosystem, um in dessen Reiseinformationssystem Ereignisse aus diversen Schnittstellen zu integrieren und in Echtzeit zu korrelieren.

Der Zulieferer Bosch setzt auf Confluent Cloud für die asynchrone Kommunikation zwischen Anwendungen und Microservices in dessen IoT-Projekten für Echtzeit-Streaming und persistente Speicherung von Ereignissen.

Dieses Buch gibt einen sehr guten Überblick über die verschiedenen Komponenten von Kafka. Dabei werden die Grundkonzepte, APIs und Architekturen beschrieben, sodass jeder sein nächstes Kafka-Projekt erfolgreich gestalten kann. Viel Spaß und Erfolg dabei.

Kai Wähner

Field CTO bei Confluent

Über das Buch

Wir haben lange nach einem Einstiegswerk für Apache Kafka¹ auf Deutsch gesucht. Als uns dann der Hanser Verlag gefragt hat, ob wir es schreiben wollen, war der Beschluss schnell gefasst. Nach einigen Startschwierigkeiten und mehr Zeit als geplant, halten Sie nun unser Buch in der Hand. Danke für Ihr Vertrauen.

Wir wissen aus eigener Erfahrung, dass es einfach ist, sich mit Kafka selbst zu überfordern. Gute Informationen zu Kafka auf Deutsch zu finden ist nicht immer leicht. In diesem Buch möchten wir Sie dabei unterstützen, einen guten Einstieg in Kafka zu finden, und Ihnen alles auf den Weg mitgeben, was Sie für Ihre Kafka-Mission benötigen.

Dieses Buch basiert auf den Schulungen, die wir für unsere Kunden geben. Uns ist es wichtig, nicht nur die fachlichen Grundlagen zu vermitteln, sondern auch unsere Erfahrungen zu teilen, wie Unternehmen Kafka erfolgreich bei sich einsetzen und was Sie dafür zusätzlich benötigen.

Während wir dieses Buch schreiben, befindet sich die Apache-Kafka-Welt im Umbruch. Neben zahlreichen Änderungen in Kafka 3.0, wird Zookeeper durch eine eigene Clusterkoordinationslösung ersetzt. Bis Redaktionsschluss wurde Kafka 3.0 noch nicht veröffentlicht, aber wir erwarten die Veröffentlichung in Kürze. Wir haben mit großer Sorgfalt darauf geachtet, dass alle relevanten zu erwartenden Änderungen im Buch vermerkt sind. Wir werden auf der Seite <https://streamcommit.de/buch/updates/> mögliche Fehler und unerwartete Änderungen vermerken.

Anatoly Zelenin und Alexander Kropp

August 2021

■ Wer sollte das Buch lesen?

Wir haben uns beim Schreiben dieses Buches an unseren Schulungsteilnehmern orientiert und möchten insbesondere diesen Zielgruppen das Bestmögliche auf ihrem Weg mitgeben. Wir glauben, dass dieses Buch allen ITlern, die sich mit Kafka beschäftigen möchten, einen guten Startpunkt gibt.

¹⁾ Apache Kafka <https://kafka.apache.org/> ist eine eingetragene Marke der Apache Software Foundation <https://apache.org/>.

IT-ArchitektInnen unterstützen wir dabei zu verstehen, wie sich Kafka in die Unternehmenslandschaft eingliedern kann und welche Konzepte sich bewährt haben. AdministratorInnen verstehen nicht nur die Kafka-Grundlagen, sondern lernen, wie sich Kafka am besten betreiben lässt und auch wie sie ihre Teams dabei unterstützen können, das Beste aus Kafka zu machen. Für EntwicklerInnen geben wir hier einen guten Überblick über die Konzepte und einen sehr kurzen Einstieg in die Programmierung mit Kafka. Dieses Buch ist aber kein Entwickler-Handbuch, da dies ein eigenes Buch wäre. Wir glauben auch, dass technische Teamleiter in diesem Buch viele interessante Konzepte lernen werden, um ihr Team bestmöglich zu unterstützen und Kafkas Wesen besser zu verstehen. Auch wenn Sie sich zu keiner dieser Gruppen zugehörig fühlen, wünschen wir Ihnen viel Freude bei der Lektüre dieses Werkes.

■ Konventionen im Buch

Kursiv stellen wir neu eingeführte Begriffe dar.

Sämtliche Code-Beispiele sind in der Unix-Shell-Syntax geschrieben.

Code-Beispiele im Fließtext stellen wir in Monospace dar.

Ausführliche Code-Beispiele werden folgendermaßen formatiert:

```
# Ein Kommentar
$ ein befehl zum eintippen
Ausgabe des Befehls
> Manche Befehle benötigen zusätzliche Eingaben.
```

Zeilen, die mit einer Raute (#) anfangen, sind unsere Kommentare und können ignoriert werden. Zeilen, die mit einem Dollar-Symbol (\$) beginnen, sind jene, die wir in unsere Kommandozeile eintippen können (ohne das Dollar-Symbol). Alles, was einzutippen ist, wird **fett** formatiert.

■ Nutzung von Code-Beispielen und Abbildungen

Dieses Buch enthält zahlreiche Code-Beispiele. Diese sollen das Verständnis erleichtern und die Konzepte praxisnah vermitteln. Code-Beispiele aus diesem Buch dürfen beliebig verwendet werden. Wir freuen uns über eine Quellenangabe, diese ist aber nicht Pflicht. Eine solche Quellenangabe kann wie folgt aussehen:

Apache Kafka: Von den Grundlagen bis zum Produktiveinsatz; von Anatoly Zelenin und Alexander Kropp; © 2022 Carl Hanser Verlag München

Um unsere Abbildungen in anderen Werken zu verwenden, bitten wir Sie, sich mit uns in Verbindung zu setzen:

buch@streamcommit.de

■ Danksagung

Unser Dank geht an unsere Familien, Freunde und Bekannte, die dieses Werk erst ermöglicht haben. Danke für eure Unterstützung in all den Jahren. Danke dafür, dass Ihr zu uns gehalten und uns wo immer möglich unterstützt habt!

Wir möchten der Apache Kafka Community, allen beteiligten Menschen und Unternehmen für die Erschaffung und die Pflege dieser Software danken.

Wir möchten Thomas Trepper dafür danken, dass er dieses Buchprojekt erst möglich gemacht hat und uns mit dem Hanser Verlag in Verbindung gebracht hat. Vom Hanser Verlag möchten wir insbesondere Sylvia Hasselbach für die Unterstützung und die fantastische Zusammenarbeit danken. Auch für die Geduld, die sie manchmal mit uns aufbringen musste.

Wir danken unseren Schulungsteilnehmern und Kunden für die Unterstützung und die Erfahrungen, die wir gemeinsam mit ihnen machen durften. Ohne ihre Unterstützung wäre das Buch weder finanziell noch inhaltlich zu stemmen gewesen.

Unser Dank geht auch an all die wunderbaren Menschen, die uns beim Erstellen des Buches direkt unterstützt haben. An Kai Wähner für das Vorwort. An Thomas Natzschka für die textuelle Unterstützung. Natürlich auch an all diejenigen, die uns mit ihrem Feedback zum Buch unterstützt haben und es Korrektur gelesen haben: Inga Blundell, Walter Forkel, Daniela Griesinger, Tobias Heller, Vincent Latzko, Andrej Olunczek, Elin Rixmann, Thomas Trepper und David Weber.

