

Apache Kafka

Course Duration ~65 Hours · 8 Weeks · Live Online · Lab-Heavy · Portfolio of Running Systems

Curriculum reviewed: July 2026 — Kafka moves through KIPs quickly; this syllabus tracks the current release line

[Kafka 4.x](#)
[KRaft](#)
[Producers & Consumers](#)
[Connect & CDC](#)
[Kafka Streams](#)
[Production Ops](#)

In short: Most Kafka courses still teach you to configure ZooKeeper. Kafka removed it in 4.0 — not deprecated, removed — and every current cluster runs in KRaft mode, with no in-place upgrade path from a ZooKeeper-based cluster. This course teaches the Kafka you'll actually be handed: KRaft architecture and its own controller and Raft metric namespaces, the new broker-managed consumer rebalance protocol that ends stop-the-world rebalances (including the consumer-group migration path), and share groups — the emerging queue-style cooperative consumption — taught honestly as an evolving capability. Across 9 modules you stand up and break a multi-broker KRaft cluster, harden producers against message loss, evolve schemas without breaking live consumers, build CDC with Debezium, ship a Kafka Streams application with exactly-once processing, secure and monitor a production cluster, and close with an event-driven capstone defended with a documented failure-mode analysis. Written for engineers: fresh IT graduates heading into data or backend roles, and working engineers closing the ZooKeeper-to-KRaft gap. Aligned to the Confluent developer and administrator certification tracks.

Duration	8 weeks · ~65 hours · live + hands-on	Format	Live online · lab-heavy · portfolio of running systems
Prerequisite	Comfortable programming in Java or Python, basic Linux and networking. No distributed-systems background required.	Curriculum	9 modules, from event-driven thinking to cloud deployment and capstone
Certification	Aligned to the Confluent developer and administrator certification tracks for Apache Kafka	Ideal for	Fresh IT graduates and backend/DevOps/SRE/data engineers moving into event-driven systems

When Kafka Is the Wrong Answer

For low-volume task queues, simple request/response, or workloads a managed queue handles fine, Kafka's operational cost isn't justified — reaching for it reflexively is how platform teams end up on call for infrastructure nobody needed. Knowing that boundary is a senior-engineer signal and a common interview question. This course teaches it explicitly: you will learn to justify *not* using Kafka as confidently as using it.

Course Highlights

- Every module leads with what you build and ship, not what you "learn about"
- KRaft from day one — ZooKeeper covered only as legacy context, never as the main path
- The new broker-managed rebalance protocol, including the consumer-group migration path
- Share groups taught honestly as an evolving queue-style capability
- Break-and-recover labs: kill brokers, force elections, watch replication save you
- Schemas as contracts: evolve a topic without breaking live downstream consumers
- Production operations: security, MirrorMaker 2 DR, KRaft-era monitoring, the incidents that actually page you

Hands-on Projects & Portfolio

- **KRaft Cluster from Scratch** — stand up a multi-broker cluster, break it deliberately, and recover it
- **Order-Processing Pipeline** — idempotent producers, consumer groups and a dead letter topic
- **Schema Evolution Exercise** — change a contract without breaking live downstream consumers
- **CDC Pipeline** — change data capture from a relational database into Kafka and onward to a warehouse
- **Kafka Streams Application** — stateful windowed aggregation with exactly-once processing
- **Secured Cluster** — TLS, SASL authentication, ACLs and per-client quotas
- **Lag-Aware Monitoring Stack** — alerting that catches consumer lag before users notice

- **Capstone** — end-to-end event-driven system with a documented failure-mode analysis

Tools & Stack

Apache Kafka 4.x · KRaft · Kafka Streams · Kafka Connect · Schema Registry · ksqldb · Debezium · Avro · Protobuf · Confluent Cloud · AWS MSK · Strimzi · Kubernetes · Docker · Prometheus · Grafana · Java · Python · Terraform

Certification Alignment

- Developer track: Modules 1–7 cover producers, consumers, schemas, Connect and Streams — the core of the Confluent **developer** certification for Apache Kafka
- Administrator track: Modules 2, 8 and 9 cover cluster architecture, security, operations and DR — the core of the Confluent **administrator** certification
- Presented as preparation and alignment — no official partnership or accreditation implied, and no exam outcome guaranteed

Career Outcomes

Kafka is infrastructure, which makes the skill unusually durable — organisations that adopt it rarely migrate away, and engineers who can operate it under load are scarcer than engineers who can write a producer. Backend, DevOps and data engineering are the natural on-ramps, and the ZooKeeper-to-KRaft transition has left a real gap between what most people were taught and what current clusters run. This course targets:

- Data Engineer · Streaming / Real-Time Data Engineer · Backend Engineer (event-driven systems) · Platform Engineer
- Kafka Administrator · Site Reliability Engineer · Solutions Architect · Integration Engineer

Who This Course Is For

- Final-year students and fresh IT graduates targeting data or backend engineering roles
- Java, Python, Go and .NET developers moving into event-driven architecture
- Data engineers adding streaming to batch skills
- DevOps, SRE and platform engineers who have inherited a Kafka cluster
- Architects designing microservice and real-time systems
- Engineers on ZooKeeper-era Kafka planning a KRaft migration

Module 1: Event-Driven Thinking

6 Hours

YOU SHIP

- An architecture decision record for a real system: where events beat request/response, where they don't, and where Kafka fits against a traditional broker

TOPICS

- Why request/response breaks down at scale; the log as a primitive; events vs messages vs commands
- Event-driven and event-sourcing patterns
- Where Kafka fits against traditional message brokers; the shape of the problems it actually solves

Assessment: Quiz + Lab

Module 2: Kafka Architecture on KRaft

8 Hours

YOU SHIP

- A multi-broker KRaft cluster stood up from scratch, then broken deliberately — killed brokers, forced elections — and recovered, with the metadata quorum observed throughout

TOPICS

- Brokers, controllers and the Raft-based metadata quorum; topics, partitions, offsets
- Replication and in-sync replicas; leader election; the commit log and retention; log compaction; tiered storage concepts

- A clear-eyed comparison with the ZooKeeper architecture it replaced — work on legacy clusters without being taught to build one

Assessment: Quiz + Lab

Module 3: Producers in Depth

7 Hours

YOU SHIP

- A producer hardened against message loss — keys and partitioning chosen deliberately, acknowledgements, idempotence and retries configured with the trade-offs documented

TOPICS

- The send path; keys and partitioning strategy; acknowledgements and durability trade-offs
- Batching, compression and linger; retries and delivery guarantees; idempotent producers
- The configuration choices that quietly decide whether you lose messages

Assessment: Quiz + Lab

Module 4: Consumers & Consumer Groups

8 Hours

YOU SHIP

- An order-processing consumer group with commit strategies, a dead letter topic, and a live migration from the classic rebalance protocol to the new broker-managed one

TOPICS

- Offsets and commit strategies; partition assignment; rebalancing and the new broker-managed rebalance protocol — including converting consumer groups between classic and new
- Consumer lag as the metric that matters; error handling and dead letter topics
- Share groups for queue-style cooperative consumption — an evolving capability, taught honestly

Assessment: Quiz + Lab

Module 5: Schemas & Data Contracts

6 Hours

YOU SHIP

- A schema evolution exercise: change a live contract in the Schema Registry without breaking downstream consumers, with compatibility modes chosen and justified

TOPICS

- Avro, Protobuf and JSON Schema; the Schema Registry
- Compatibility modes; evolving a schema without breaking downstream consumers
- The topic as a published API contract between teams

Assessment: Quiz + Lab

Module 6: Kafka Connect

7 Hours

YOU SHIP

- A change data capture pipeline: Debezium streaming a relational database into Kafka and onward to a warehouse sink, with connector failures handled and offsets managed

TOPICS

- Source and sink connectors; standalone vs distributed workers; converters and transforms
- Change data capture from databases with Debezium; sinking to object storage, warehouses and search
- Connector failure handling and offset management

Assessment: Quiz + Lab

Module 7: Stream Processing

8 Hours

YOU SHIP

- A Kafka Streams application with stateful windowed aggregation, joins and exactly-once processing — plus a written engine choice: Streams vs Flink vs Spark for a given workload

TOPICS

- Kafka Streams topologies; KStream and KTable duality; stateful operations and state stores
- Windowing; joins; exactly-once semantics; interactive queries; ksqldb
- An honest comparison with Apache Flink and Spark Structured Streaming — choose rather than default

Assessment: Quiz + Lab**Module 8: Operating Kafka in Production**

9 Hours

YOU SHIP

- A production-hardened cluster: TLS, SASL, ACLs and quotas; MirrorMaker 2 disaster recovery; and a KRaft-era monitoring stack that catches consumer lag before users notice

TOPICS

- Cluster sizing and partition planning; replication and rack awareness; security (TLS, SASL, ACLs, quotas)
- Multi-cluster and disaster recovery with MirrorMaker 2; monitoring with the KRaft-era metric namespaces — ZooKeeper dashboards don't carry over; Prometheus and Grafana
- Capacity and cost; upgrades and rolling restarts; the incidents that actually page you

Assessment: Quiz + Lab**Module 9: Cloud, Ecosystem & Capstone**

6 Hours

YOU SHIP

- One end-to-end event-driven system — designed, built, deployed and defended with a documented failure-mode analysis — plus a build-vs-buy recommendation across managed offerings

TOPICS

- Managed Kafka on Confluent Cloud, AWS MSK and Azure Event Hubs' Kafka interface
- Kubernetes deployment with Strimzi; infrastructure as code; build-vs-buy economics
- Capstone design, build and defense

Assessment: Capstone Project + Review