

Apache Kafka: Producer & Consumer Architecture - Advanced Training

Description

The purpose of this training is to take Producer and Consumer architecture from fundamentals to an advanced level. On the producer side, key/partition selection, batching/linger, acks, idempotent producers, and transactional operations are covered in depth. On the consumer side, consumer groups, rebalancing (cooperative-sticky), offset management, backpressure, and concurrency patterns are addressed. The training also focuses on delivery guarantees (at-most-once, at-least-once, exactly-once), error handling (retry, delayed retry, DLQ), ordering strategies, compression, and performance tuning to build resilient and high-throughput production-grade data pipelines.

Learning Outcomes

- Apply key/partition strategies, batching/linger, acks, idempotent and transactional producer configurations for reliable low-latency writes
- Design scalable consumption patterns using consumer groups, cooperative-sticky rebalancing, offset/commit policies, pause/resume, and concurrency models
- Operate observable, resilient, and high-performance end-to-end Kafka pipelines using delivery guarantees, error handling, ordering, compression, and tuning practices

Target Audience

- Backend developers and microservices teams
- Data engineers and platform teams
- Teams operating high-volume Kafka producer/consumer workloads

Outline

Foundations & Producer Deep Dive

- Kafka recap: topic, partition, key, and ordering relationship
- Producer architecture: buffer, batch, linger.ms, compression.type
- acks, retries, delivery.timeout.ms, and failure scenarios
- Idempotent producer and transactional outbox pattern
- Key/partition strategies and hot-partition mitigation
- Performance practices: throughput vs latency trade-offs
- Lab: Java Producer Hands-on

Consumer Deep Dive & Consumption Patterns

- Consumer groups and partition assignment strategies
- Rebalancing mechanisms and minimization
- Offset management and commit strategies
- Backpressure handling: pause/resume, poll loop, max.poll.*
- Concurrency patterns
- Error handling: retry topics, delayed retry, DLQ
- Lab: Java Consumer Hands-on

Advanced Topics, Performance & Practice

- Delivery guarantees and transactional pipelines
- Ordering and idempotency strategies
- Compression, batch size, fetch tuning impacts
- Observability: lag, end-to-end latency, metrics
- Resilience practices and tuning exercise

Prerequisites

Basic Java programming knowledge is required.