Admissions & Support:
support@mockattempt.com
Phone / WhatsApp: +91 98765 43210
Course Code: KAFK • **Updated:** 01 Nov 2026
https://mockattempt.com/courses/kafka-training

KAFK • DISTRIBUTED EVENT STREAMING & MESSAGING **BEGINNER TO PRINCIPAL APACHE KAFKA ARCHITECT**

100% LIVE INDUSTRY EXPERT COHORT + INTERNSHIP LETTER

Kafka Training

Master Apache Kafka from Zero-Copy OS PageCache, KRaft Consensus & ISR Replication to Idempotent Java Producers, CooperativeSticky Rebalancing, MirrorMaker 2, Debezium Kafka Connect, RocksDB Kafka Streams & JMX Lag Monitoring — with 200 Mock Tests & Voice Gemini AI Interviews at ₹5,999.

Duration: 15–17 Weeks • Live + Hands-On Projects (208 Hrs) • **Modules:** 16 • **Live Topics:** 42 • **Labs & Tasks:** 84 • **Mocks / AI Vivas:** 244 / 244 • **Projects:** 16 Module + 1 Live Client Prod

Fixed Non-Negotiable Tuition Fee: Base Fee **₹5,999** + 18% GST (₹1,080) = **Total Incl. GST: ₹7,079**

Live Batch Tracks: Weekday Live Online Batch (Mon – Fri) (Monday to Friday (5 Live Interactive Classes / Week) • 1.5 Hours / Day (60 Mins Live Architecture & Coding + 30 Mins Hands-On Lab) • 17–19 Weeks Fast-Track) | Weekend Intensive Live Online Batch (Sat – Sun) (Saturday & Sunday (Designed for Working Professionals & College Students) • 3.0 Hours / Day (2 Hrs Deep-Dive Live Coding + 1 Hr Guided Project Build) • 19–21 Weeks Weekend Track)

Core Technologies & Engineering Stack: Apache Kafka 3.8 (KRaft Mode) • Java 21 Kafka Clients API • Idempotent & Transactional Producers • Cooperative Consumer Groups • Kafka Connect & Debezium CDC • Kafka Streams & RocksDB State • MirrorMaker 2 & Prometheus JMX

1. PROGRAM MASTERY OUTCOMES & ELIGIBILITY

End-to-End Engineering Outcomes:

1. Architect fault-tolerant KRaft clusters with tuned partition counts, ISR replication (`acks=all` + `min.insync.replicas=2`), and log compaction
2. Build high-throughput idempotent Java producers and cooperative-rebalance consumer groups with deterministic offset commit guarantees
3. Engineer zero-code CDC pipelines with Kafka Connect, multi-region active-active replication via MirrorMaker 2, and stateful Kafka Streams topologies
4. Clear 200 Timed Mock Exams (10,000 Apache Kafka MCQs) and 200 Spoken Voice + Google Gemini AI Event-Streaming Architecture Interviews
5. KRaft broker architecture, topic partitioning, replication factors, ISR (`min.insync.replicas`), and leader election.
6. Idempotent & transactional producers (`acks=all`), Transactional Outbox pattern, and Avro/JSON Schema Registry contracts.
7. Consumer Group rebalancing, manual offset commits, idempotent consumers, retry topics, and Dead-Letter Queues (DLQ).
8. Stream processing fundamentals (stateful aggregations, windowed joins) and Kafka Connect CDC pipelines.

Eligibility & Prerequisites: Basic familiarity with Java classes/interfaces and Linux command-line basics — distributed log fundamentals start from scratch • Any laptop/PC with JDK 21 and Docker Desktop or local Apache Kafka 3.8+ KRaft binary archive • No prior Kafka industry experience required — Module 1 starts from environment setup, core mental models, and plain-English fundamentals. • Basic computer & internet familiarity on a Windows, macOS, or Linux laptop (8 GB+ RAM recommended for smooth live coding labs). • Ideal for B.Tech / BCA / MCA / B.Sc IT students, non-CS beginners, career switchers, and working developers upgrading to 2026 production standards. • Commitment to attend live Weekday or Weekend classes, complete daily hands-on lab checklists, and practice topic-wise Mock Tests & AI Interviews.

2. DETAILED MODULE & TOPIC-WISE SYLLABUS (16 MODULES • 42 LIVE TOPICS)

MODULE 1 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 1: Distributed Commit Logs, Zero-Copy PageCache & Apache Kafka KRaft Architecture

Understand why append-only immutable commit logs outperform traditional AMQP queues at scale, how sendfile() zero-copy I/O saturates NICs, and how KRaft eliminates ZooKeeper.

Objectives: ✓ Contrast Kafka immutable append-only partition logs against destructive read-and-delete message brokers • ✓ Trace OS PageCache sequential disk I/O and Linux `sendfile()` zero-copy kernel transfers • ✓ Explain KRaft Raft-based controller quorums, `__cluster_metadata` partition, and broker heartbeats

1.1 Apache Kafka Fundamentals — Immutable Commit Log, Sequential Disk I/O & Zero-Copy `sendfile()` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Apache Kafka Fundamentals — Immutable Commit Log, Sequential Disk I/O & Zero-Copy `sendfile()` in real Kafka systems — specifically: Core abstract...

1.2 Apache Kafka Architecture — Brokers, KRaft Controller Quorum, Partitions & `__cluster_metadata` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Apache Kafka Architecture — Brokers, KRaft Controller Quorum, Partitions & `__cluster_metadata` in real Kafka systems — specifically: Inside a mod...

Module 1 Hands-On Capstone Project (100 Marks): Event-Driven Order Backbone Throughput & Retention Sizing Blueprint — Design a 6-broker KRaft cluster blueprint capable of ingesting 250,000 order events/sec (1 KB payload) with 7-day retention, 3x replication, and zero ZooKeeper dependencies.

MODULE 2 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 2: Enterprise Event-Streaming Patterns & Production KRaft Cluster Bootstrap

Map real-world event-sourcing, CDC, log aggregation, and CQRS workloads onto Kafka, then format storage UUIDs and launch production KRaft brokers.

Objectives: ✓ Select optimal Kafka patterns for transactional outbox, CQRS read models, clickstream telemetry, and SIEM log pipelines • ✓ Format KRaft storage directories using `kafka-storage.sh random-uuid` and `format` without ZooKeeper • ✓ Tune OS kernel parameters (`vm.swappiness=1`, `vm.max_map_count`, XFS/ext4 `noatime`, open file descriptors)

2.1 Apache Kafka Use Cases — Transactional Outbox, Event Sourcing, CQRS & Real-Time Telemetry [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Apache Kafka Use Cases — Transactional Outbox, Event Sourcing, CQRS & Real-Time Telemetry in real Kafka systems — specifically: Architectural patt...

2.2 Installing & Bootstrapping Apache Kafka in KRaft Mode (`kafka-storage.sh` & Listener Security) [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Installing & Bootstrapping Apache Kafka in KRaft Mode (`kafka-storage.sh` & Listener Security) in real Kafka systems — specifically: Step-by-step...

Module 2 Hands-On Capstone Project (100 Marks): 3-Node Local KRaft Cluster Bootstrap & OS Kernel Readiness Verifier — Automate formatting and validating a 3-node combined/isolated KRaft cluster configuration with deterministic cluster UUIDs, advertised listeners, and JVM G1GC tuning.

MODULE 3 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 3: Kafka Topics, Segment Lifecycle, Log Compaction & Administrative Provisioning

Master partition parallelism math, active vs closed log segments, delete vs compact cleanup policies, and declarative topic creation via CLI and AdminClient.

Objectives: ✓ Size partition counts accurately based on target consumer throughput and broker open-file limits • ✓ Contrast `cleanup.policy=delete` (`retention.ms` / `retention.bytes`) with `cleanup.policy=compact` tombstones (`null` value) • ✓ Disable `auto.create.topics.enable` and provision governed topics via `NewTopic` and `kafka-topics.sh`

3.1 Kafka Topics Deep Dive — Partitions, Log Segments, Retention Policies & Key Compaction [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Kafka Topics Deep Dive — Partitions, Log Segments, Retention Policies & Key Compaction in real Kafka systems — specifically: How topics split into...

3.2 Creating & Managing Kafka Topics (`kafka-topics.sh` & Idempotent Java `AdminClient`) [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Creating & Managing Kafka Topics (`kafka-topics.sh` & Idempotent Java `AdminClient`) in real Kafka systems — specifically: Creating, describing, a...

Module 3 Hands-On Capstone Project (100 Marks): Automated Topic Governance & Schema-Safe Provisioning Engine — Build a Java `AdminClient` provisioning tool that creates time-retained event topics and compacted entity-snapshot topics while rejecting unsafe replication factors or partition reductions.

MODULE 4 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 4: ISR Replication, High Watermark Durability & Producer Internals

Guarantee zero data loss with In-Sync Replicas (ISR), Leader Epochs, and High Watermarks, and trace the Java Producer RecordAccumulator, batching, compression, and acks pipeline.

Objectives: ✓ Explain Leader vs Follower fetch replication, High Watermark (`HW`), Leader Epoch, and `unclean.leader.election.enable=false` • ✓ Combine `replication.factor=3`, `min.insync.replicas=2`, and `acks=all` for zero committed message loss • ✓ Tune Producer `RecordAccumulator` memory (`buffer.memory`), `batch.size`, `linger.ms`, and `compression.type=lz4/zstd`

4.1 Kafka Topic Replication — Leaders, Followers, ISR, High Watermark & Leader Epoch [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Kafka Topic Replication — Leaders, Followers, ISR, High Watermark & Leader Epoch in real Kafka systems — specifically: How follower brokers pull-r...

4.2 Apache Kafka Producer Architecture — Serializer, Partitioner, `RecordAccumulator` & Idempotence [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Apache Kafka Producer Architecture — Serializer, Partitioner, `RecordAccumulator` & Idempotence in real Kafka systems — specifically: Tracing `Kaf...

Module 4 Hands-On Capstone Project (100 Marks): Zero-Data-Loss Financial Ledger Producer & Durability Simulator — Configure and benchmark a financial ledger producer pipeline that guarantees durability across single-broker crashes while maximizing batch compression efficiency.

MODULE 5 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 5: CLI Ingestion, Delivery Semantics & Java Producer Engineering

Compare fire-and-forget, synchronous, and asynchronous send modes, test ingestion via `kafka-console-producer.sh`, and write clean, leak-free Java 21 producer clients.

Objectives: ✓ Ingest keyed and header-tagged records from CLI using `kafka-console-producer.sh --property parse.key=true` • ✓ Contrast fire-and-forget, blocking `.get()`, and non-blocking async send throughput profiles • ✓ Implement graceful producer shutdown (`flush()` / `close(Duration)`) preventing buffered batch loss on pod termination

5.1 Sending Data to Kafka Topics — CLI Key Parsing, Sync vs Async Delivery & `delivery.timeout.ms` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Sending Data to Kafka Topics — CLI Key Parsing, Sync vs Async Delivery & `delivery.timeout.ms` in real Kafka systems — specifically: Using `kafka-...

5.2 Creating a Kafka Producer in Java — Try-With-Resources, `StringSerializer` & Transactional Sends [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Creating a Kafka Producer in Java — Try-With-Resources, `StringSerializer` & Transactional Sends in real Kafka systems — specifically: Writing idi...

Module 5 Hands-On Capstone Project (100 Marks): High-Throughput Async Telemetry Producer with Graceful SIGTERM Drain — Build a Java 21 telemetry producer service that ingests 50,000 keyed JSON/Avro events asynchronously, enforces bounded memory backpressure (`max.block.ms`), and flushes all in-flight batches...

MODULE 6 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 6: Async Producer Callbacks, Consumer Groups & Cooperative Rebalancing

Capture `RecordMetadata` offsets and non-retriable exceptions in non-blocking callbacks, and master Consumer Group partition assignment and `CooperativeStickyAssignor` zero-stop-the-world rebalancing.

Objectives: ✓ Handle `Callback.onCompletion(RecordMetadata, Exception)` without blocking the Kafka `Sender` I/O thread • ✓ Size Consumer Group instances relative to partition counts (`consumers <= partitions`) to prevent idle instances • ✓ Switch from eager stop-the-world rebalancing (`RangeAssignor`) to incremental `CooperativeStickyAssignor` and static membership (`group.instance.id`)

6.1 Kafka Producer Callbacks — `RecordMetadata`, Retriable vs Non-Retriable Errors & I/O Thread Safety [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Kafka Producer Callbacks — `RecordMetadata`, Retriable vs Non-Retriable Errors & I/O Thread Safety in real Kafka systems — specifically: Inspectin...

6.2 Kafka Consumers & Consumer Groups — Partition Ownership, Heartbeats & `CooperativeStickyAssignor` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Kafka Consumers & Consumer Groups — Partition Ownership, Heartbeats & `CooperativeStickyAssignor` in real Kafka systems — specifically: How `group...

Module 6 Hands-On Capstone Project (100 Marks): Dead-Letter Routing Producer Callback & Zero-Pause Cooperative Consumer Fleet — Build an async producer callback handler that routes failed records to a DLQ without blocking `Sender` I/O, paired with a static-member `CooperativeStickyAssignor` consumer fleet.

MODULE 7 OF 16 • 11 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 7: CLI Stream Debugging, Consumer Lag Forensics & Offset Reset Operations

Inspect live headers, keys, partitions, and uncommitted/committed records with `kafka-console-consumer.sh`, and diagnose lag or safely rewind offsets using `kafka-consumer-groups.sh`.

Objectives: ✓ Format keys, timestamps, headers, and isolation levels (`--isolation-level read_committed`) in `kafka-console-consumer.sh` • ✓ Audit partition `CURRENT-OFFSET`, `LOG-END-OFFSET`, and `LAG` across active consumer groups • ✓ Execute dry-run (`--dry-run`) before committing timestamp or offset resets (`--reset-offsets --to-datetime`)

7.1 Kafka Console Consumer — Key/Header Formatting, `--from-beginning` & `read_committed` Isolation [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Kafka Console Consumer — Key/Header Formatting, `--from-beginning` & `read_committed` Isolation in real Kafka systems — specifically: Inspecting l...

7.2 Kafka Consumer Group CLI — `LAG` Diagnostics, Hot-Key Skew & Safe `--reset-offsets --dry-run` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Kafka Consumer Group CLI — `LAG` Diagnostics, Hot-Key Skew & Safe `--reset-offsets --dry-run` in real Kafka systems — specifically: Using `kafka-c...

Module 7 Hands-On Capstone Project (100 Marks): Incident Incident-Response Consumer Lag Auditor & Dry-Run Offset Rewind Runbook — Create an operational CLI + Java AdminClient diagnostic utility that detects skewed consumer-group partition lag (hot keys) and validates safe `--dry-run` offset rewinds after a bad downstre...

MODULE 8 OF 16 • 11 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 8: Java Consumer Poll Loops, Manual Offset Commits & Multi-Cluster MirrorMaker 2

Write thread-safe Java consumer poll loops with manual at-least-once offset commits and `ConsumerRebalanceListener`, and architect multi-region disaster recovery with MirrorMaker 2.

Objectives: ✓ Eliminate data loss from `enable.auto.commit=true` by committing offsets manually (`commitSync` / `commitAsync`) AFTER downstream persistence • ✓ Handle `WakeupException` for clean shutdown and commit in-progress offsets inside `onPartitionsRevoked` • ✓ Replicate topics and translate consumer offsets across geo-distributed Kafka clusters using MirrorMaker 2 (`MirrorSourceConnector` + `MirrorCheckpointConnector`)

8.1 Creating a Kafka Consumer in Java — Poll Loop, `enable.auto.commit=false` & `WakeupException` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Creating a Kafka Consumer in Java — Poll Loop, `enable.auto.commit=false` & `WakeupException` in real Kafka systems — specifically: Why `KafkaCons...

8.2 Multi-Cluster Kafka Architectures — MirrorMaker 2, Active-Active Replication & Stretch Clusters [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Multi-Cluster Kafka Architectures — MirrorMaker 2, Active-Active Replication & Stretch Clusters in real Kafka systems — specifically: Designing ge...

Module 8 Hands-On Capstone Project (100 Marks): At-Least-Once Idempotent Java Consumer & Active-Passive DR Failover Blueprint — Implement a production Java `KafkaConsumer` poll loop with `enable.auto.commit=false`, rebalance-safe offset commits, `wakeup()` shutdown, and a MirrorMaker 2 cross-region DR config.

MODULE 9 OF 16 • 11 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 9: Kafka Connect CDC Pipelines & Stateful Kafka Streams Topologies

Stream database binlogs/WALs code-free with Kafka Connect + Debezium Single Message Transforms, and build embedded stateful stream-processing microservices using the Kafka Streams DSL.

Objectives: ✓ Deploy Distributed Kafka Connect workers (`connect-offsets`, `connect-configs`, `connect-status`) with Debezium CDC and Dead Letter Queues • ✓ Transform, filter, join, and aggregate real-time events using the Kafka Streams `StreamsBuilder` DSL • ✓ Configure `processing.guarantee=exactly_once_v2` (EOS) for atomic read-process-write stream topologies

9.1 Kafka Connect & Debezium CDC — Distributed Workers, Internal Compacted Topics, SMTs & DLQ [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Kafka Connect & Debezium CDC — Distributed Workers, Internal Compacted Topics, SMTs & DLQ in real Kafka systems — specifically: Declarative Source...

9.2 Kafka Stream Processing — Embedded `KafkaStreams` Topologies, `exactly_once_v2` & `TopologyTestDriver` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Kafka Stream Processing — Embedded `KafkaStreams` Topologies, `exactly_once_v2` & `TopologyTestDriver` in real Kafka systems — specifically: Why K...

Module 9 Hands-On Capstone Project (100 Marks): Debezium Outbox CDC Connector + Real-Time Fraud Aggregation Kafka Streams Service — Design a Distributed Kafka Connect Debezium PostgreSQL Outbox connector config with DLQ error tolerance, feeding a Java `KafkaStreams` topology that flags accounts exceeding 5 high-value tra...

MODULE 10 OF 16 • 11 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 10: KStream/KTable Duality, Event-Time Windowing & Production JMX Observability

Master stream-table duality (`KStream`, `KTable`, `GlobalKTable`), event-time windows with grace periods, RocksDB standby replicas, and zero-blindspot JMX/Prometheus broker & consumer monitoring.

Objectives: ✓ Distinguish append-only `KStream` events from changelog-backed `KTable` and broadcast `GlobalKTable` lookups • ✓ Handle out-of-order late events accurately using Event-Time `TimestampExtractor`, Tumbling/Hopping/Session windows, and `.grace(Duration)` • ✓ Alert on critical JMX metrics (`UnderReplicatedPartitions`, `OfflinePartitionsCount`, `ActiveControllerCount`, request queue saturation, and consumer lag)

10.1 Stream Processing Concepts — `KStream` vs `KTable` Duality, Event-Time Windows & RocksDB State [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Stream Processing Concepts — `KStream` vs `KTable` Duality, Event-Time Windows & RocksDB State in real Kafka systems — specifically: Stream-Table...

10.2 Kafka Monitoring & SRE Observability — JMX Exporter, Critical Broker MBeans & Lag Alerting [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Kafka Monitoring & SRE Observability — JMX Exporter, Critical Broker MBeans & Lag Alerting in real Kafka systems — specifically: Exposing Kafka JM...

Module 10 Hands-On Capstone Project (100 Marks): Windowed Revenue Aggregator (`RocksDB` + `StandbyReplicas`) & SRE JMX Alerting Suite — Build an event-time tumbling-window revenue aggregator backed by RocksDB changelog state and `num.standby.replicas=1`, plus a Prometheus/JMX health evaluator alerting on cluster degradation.

MODULE 11 OF 16 • 16 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

KRaft Internals, Idempotent/Transactional Producers & Partition Strategy

Production-grade Kafka Training engineering module covering KRaft Internals, Idempotent/Transactional Producers & Partition Strategy with hands-on labs and capstone build.

Objectives: ✓ KRaft Consensus Metadata Quorum (Zero ZooKeeper) & ISR Replication — Purpose: Understand leader election, high watermarks, and why acks=all + min.insync.replicas=2 prevents data loss. • ✓ Idempotent Producers (PID + Sequence Numbers) & EOS Transactions — Purpose: Eliminate duplicate messages during network retries and atomic consume-transform-produce flows. • ✓ Partition Key Sizing, Hot-Key Skew Mitigation & Log Compaction — Purpose: Balance throughput across brokers while preserving strict per-customer event order. • ✓ Batching (linger.ms, batch.size) & LZ4/Zstd Compression Tuning — Purpose: Multiply Kafka throughput 5x while slashing cross-AZ network bandwidth bills.

11.1 KRaft Consensus Metadata Quorum (Zero ZooKeeper) & ISR Replication [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply KRaft Consensus Metadata Quorum (Zero ZooKeeper) & ISR Replication in real Kafka systems — specifically: Understand leader election, high watermar...

11.2 Idempotent Producers (PID + Sequence Numbers) & EOS Transactions [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Idempotent Producers (PID + Sequence Numbers) & EOS Transactions in real Kafka systems — specifically: Eliminate duplicate messages during network...

11.3 Partition Key Sizing, Hot-Key Skew Mitigation & Log Compaction [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Partition Key Sizing, Hot-Key Skew Mitigation & Log Compaction in real Kafka systems — specifically: Balance throughput across brokers while prese...

11.4 Batching (linger.ms, batch.size) & LZ4/Zstd Compression Tuning [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Batching (linger.ms, batch.size) & LZ4/Zstd Compression Tuning in real Kafka systems — specifically: Multiply Kafka throughput 5x while slashing c...

Module 11 Hands-On Capstone Project (100 Marks): Zero-Data-Loss Financial Event Backbone on Apache Kafka — Design, implement, test, and document 'Zero-Data-Loss Financial Event Backbone on Apache Kafka' applying every production concept from Module 11 (KRaft Internals, Idempotent/Transactional Pr...

MODULE 12 OF 16 • 16 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Consumer Groups, Cooperative Rebalancing, DLQ Retries & Schema Registry

Production-grade Kafka Training engineering module covering Consumer Groups, Cooperative Rebalancing, DLQ Retries & Schema Registry with hands-on labs and capstone build.

Objectives: ✓ CooperativeStickyAssignor vs Stop-the-World Eager Rebalances — Purpose: Scale consumer pods up and down without freezing event processing across all partitions. • ✓ Manual Offset Commits, At-Least-Once Idempotency & Poison-Pill Isolation — Purpose: Ensure offsets commit only after durable DB writes and route bad payloads to DLQs. • ✓ Non-Blocking Retry Topics (retry-5s -> retry-1m -> DLT) — Purpose: Retry transient downstream API timeouts without blocking healthy partition traffic. • ✓ Confluent/Apicurio Schema Registry (Avro/Protobuf) Compatibility Modes — Purpose: Enforce BACKWARD/FULL schema rules in CI so producer deploys never crash consumers.

12.1 CooperativeStickyAssignor vs Stop-the-World Eager Rebalances [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply CooperativeStickyAssignor vs Stop-the-World Eager Rebalances in real Kafka systems — specifically: Scale consumer pods up and down without freezin...

12.2 Manual Offset Commits, At-Least-Once Idempotency & Poison-Pill Isolation [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Manual Offset Commits, At-Least-Once Idempotency & Poison-Pill Isolation in real Kafka systems — specifically: Ensure offsets commit only after du...

12.3 Non-Blocking Retry Topics (retry-5s -> retry-1m -> DLT) [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Non-Blocking Retry Topics (retry-5s -> retry-1m -> DLT) in real Kafka systems — specifically: Retry transient downstream API timeouts without bloc...

12.4 Confluent/Apicurio Schema Registry (Avro/Protobuf) Compatibility Modes [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Confluent/Apicurio Schema Registry (Avro/Protobuf) Compatibility Modes in real Kafka systems — specifically: Enforce BACKWARD/FULL schema rules in...

Module 12 Hands-On Capstone Project (100 Marks): Resilient Multi-Stage Retry & Schema-Governed Consumer Fleet — Design, implement, test, and document 'Resilient Multi-Stage Retry & Schema-Governed Consumer Fleet' applying every production concept from Module 12 (Consumer Groups, Cooperative Rebalancin...

MODULE 13 OF 16 • 16 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Kafka Connect, Debezium Change Data Capture (CDC) & Outbox Streaming

Production-grade Kafka Training engineering module covering Kafka Connect, Debezium Change Data Capture (CDC) & Outbox Streaming with hands-on labs and capstone build.

Objectives: ✓ Kafka Connect Distributed Workers, Source/Sink Connectors & SMTs — Purpose: Stream data between databases, S3, and Elasticsearch with config instead of custom glue code. • ✓ Log-Based Debezium CDC (WAL/Binlog) vs Polling Anti-Patterns — Purpose: Capture every row insert/update/delete in milliseconds with near-zero primary DB load. • ✓ Transactional Outbox Event Router Pattern — Purpose: Publish clean domain events atomically alongside microservice SQL commits. • ✓ Connector Secret Providers, Dead-Letter Sink Routing & Offset Resets — Purpose: Operate Connect clusters securely and recover broken connectors cleanly.

13.1 Kafka Connect Distributed Workers, Source/Sink Connectors & SMTs [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Kafka Connect Distributed Workers, Source/Sink Connectors & SMTs in real Kafka systems — specifically: Stream data between databases, S3, and Elas...

13.2 Log-Based Debezium CDC (WAL/Binlog) vs Polling Anti-Patterns [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Log-Based Debezium CDC (WAL/Binlog) vs Polling Anti-Patterns in real Kafka systems — specifically: Capture every row insert/update/delete in milli...

13.3 Transactional Outbox Event Router Pattern [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Transactional Outbox Event Router Pattern in real Kafka systems — specifically: Publish clean domain events atomically alongside microservice SQL...

13.4 Connector Secret Providers, Dead-Letter Sink Routing & Offset Resets [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Connector Secret Providers, Dead-Letter Sink Routing & Offset Resets in real Kafka systems — specifically: Operate Connect clusters securely and r...

Module 13 Hands-On Capstone Project (100 Marks): Real-Time PostgreSQL-to-Kafka CDC & Search Sync Pipeline — Design, implement, test, and document 'Real-Time PostgreSQL-to-Kafka CDC & Search Sync Pipeline' applying every production concept from Module 13 (Kafka Connect, Debezium Change Data Capture...

MODULE 14 OF 16 • 16 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Stateful Kafka Streams (RocksDB, Windows) & Production Cluster Ops

Production-grade Kafka Training engineering module covering Stateful Kafka Streams (RocksDB, Windows) & Production Cluster Ops with hands-on labs and capstone build.

Objectives: ✓ KStream vs KTable Duality, GlobalKTable & Stream-Table Joins — Purpose: Enrich high-velocity click/payment events with customer tier metadata in real time. • ✓ Tumbling, Hopping & Session Windows + Watermarks for Late Events — Purpose: Detect 5-failed-card-attempts-in-2-minutes fraud bursts accurately even on delayed mobile events. • ✓ RocksDB State Stores, Changelog Topics & Standby Replicas — Purpose: Fail over stateful stream processors in seconds instead of replaying hours of state. • ✓ Consumer Lag Alerting, Cruise Control Rebalancing, mTLS/SASL & MirrorMaker 2 — Purpose: Monitor consumer lag SLAs, enforce ACL security, and replicate cross-region DR clusters.

14.1 KStream vs KTable Duality, GlobalKTable & Stream-Table Joins [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply KStream vs KTable Duality, GlobalKTable & Stream-Table Joins in real Kafka systems — specifically: Enrich high-velocity click/payment events with...

14.2 Tumbling, Hopping & Session Windows + Watermarks for Late Events [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Tumbling, Hopping & Session Windows + Watermarks for Late Events in real Kafka systems — specifically: Detect 5-failed-card-attempts-in-2-minutes...

14.3 RocksDB State Stores, Changelog Topics & Standby Replicas [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply RocksDB State Stores, Changelog Topics & Standby Replicas in real Kafka systems — specifically: Fail over stateful stream processors in seconds in...

14.4 Consumer Lag Alerting, Cruise Control Rebalancing, mTLS/SASL & MirrorMaker 2 [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Consumer Lag Alerting, Cruise Control Rebalancing, mTLS/SASL & MirrorMaker 2 in real Kafka systems — specifically: Monitor consumer lag SLAs, enfo...

Module 14 Hands-On Capstone Project (100 Marks): Real-Time Fraud Velocity & Sliding-Window Aggregator — Design, implement, test, and document 'Real-Time Fraud Velocity & Sliding-Window Aggregator' applying every production concept from Module 14 (Stateful Kafka Streams (RocksDB, Windows) & Pro...

MODULE 15 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Schema Registry Compatibility, Transactional Outbox & Dead-Letter Topologies

Self-explanatory list-and-bullet production mastery module covering Schema Registry Compatibility, Transactional Outbox & Dead-Letter Topologies in Kafka Training, paired with hands-on labs, capstone repository, Topic Mock Tests, and Spoken Gemini AI Viva.

Objectives: ✓ Schema Registry Evolution Rules (Backward, Forward & Full Compatibility) — Purpose: Evolve event payloads across teams without breaking downstream Kafka consumers in production. • ✓ Non-Blocking Retry Topics, Dead-Letter Topics (DLT) & Idempotent Consumer Deduplication — Purpose: Isolate malformed or transiently failing records without stalling partition consumption. • ✓ Change Data Capture (CDC) with Debezium & Transactional Outbox Architecture — Purpose: Stream committed database changes into Kafka reliably with zero dual-write race conditions. • ✓ Build, verify, and document 'Avro/Protobuf Schema-Governed Order Pipeline with Retry Tiers & DLT Replay' with clean Git commits and edge-case protection. • ✓ Clear 100% of Topic Timed Mock Tests and Spoken Voice + Google Gemini AI Technical Interviews.

15.1 Schema Registry Evolution Rules (Backward, Forward & Full Compatibility) [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Schema Registry Evolution Rules (Backward, Forward & Full Compatibility) in real Kafka systems — specifically: Evolve event payloads across teams...

15.2 Non-Blocking Retry Topics, Dead-Letter Topics (DLT) & Idempotent Consumer Deduplication [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Non-Blocking Retry Topics, Dead-Letter Topics (DLT) & Idempotent Consumer Deduplication in real Kafka systems — specifically: Isolate malformed or...

15.3 Change Data Capture (CDC) with Debezium & Transactional Outbox Architecture [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Change Data Capture (CDC) with Debezium & Transactional Outbox Architecture in real Kafka systems — specifically: Stream committed database change...

Module 15 Hands-On Capstone Project (100 Marks): Avro/Protobuf Schema-Governed Order Pipeline with Retry Tiers & DLT Replay — Architect, implement, test, and document 'Avro/Protobuf Schema-Governed Order Pipeline with Retry Tiers & DLT Replay' applying every bullet-wise engineering concept from Module 15 (Schema Re...

MODULE 16 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

KRaft Cluster Operations, Consumer Lag SRE & Event-Driven System Design

Self-explanatory list-and-bullet production mastery module covering KRaft Cluster Operations, Consumer Lag SRE & Event-Driven System Design in Kafka Training, paired with hands-on labs, capstone repository, Topic Mock Tests, and Spoken Gemini AI Viva.

Objectives: ✓ CooperativeStickyAssignor, Static Group Membership & Preventing Rebalance Storms — Purpose: Keep high-throughput consumer groups stable during rolling pod deployments and GC pauses. • ✓ Monitoring Under-Replicated Partitions, ISR Shrinks, Disk Retention & Compaction — Purpose: Detect broker bottlenecks early and tune min.insync.replicas + acks=all for zero data loss. • ✓ Multi-Region MirrorMaker 2 & Event-Driven Architecture Interviews — Purpose: Design ordering guarantees, partition keys, and replayable event-sourced systems in interviews. • ✓ Build, verify, and document 'Consumer-Lag Autoscaling Monitor, Rebalance Tuning & Disaster-Recovery Mirroring' with clean Git commits and edge-case protection. • ✓ Clear 100% of Topic Timed Mock Tests and Spoken Voice + Google Gemini AI Technical Interviews.

16.1 CooperativeStickyAssignor, Static Group Membership & Preventing Rebalance Storms [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply CooperativeStickyAssignor, Static Group Membership & Preventing Rebalance Storms in real Kafka systems — specifically: Keep high-throughput consum...

16.2 Monitoring Under-Replicated Partitions, ISR Shrinks, Disk Retention & Compaction [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Monitoring Under-Replicated Partitions, ISR Shrinks, Disk Retention & Compaction in real Kafka systems — specifically: Detect broker bottlenecks e...

16.3 Multi-Region MirrorMaker 2 & Event-Driven Architecture Interviews [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Multi-Region MirrorMaker 2 & Event-Driven Architecture Interviews in real Kafka systems — specifically: Design ordering guarantees, partition keys...

Module 16 Hands-On Capstone Project (100 Marks): Consumer-Lag Autoscaling Monitor, Rebalance Tuning & Disaster-Recovery Mirroring — Architect, implement, test, and document 'Consumer-Lag Autoscaling Monitor, Rebalance Tuning & Disaster-Recovery Mirroring' applying every bullet-wise engineering concept from Module 16 (KRa...

3. LIVE CLIENT PRODUCTION PROJECT, GLOBAL CERTIFICATIONS & CAREER SCOPE

End-of-Course Live Client Production Project + Official Internship Confirmation Letter

Project: StreamPulse Event-Driven Order Pipeline, Dead-Letter Recovery & Outbox Relay • **Confirmed Role:** Apache Kafka & Event-Driven Systems Intern

Domain & Production Infra: High-Volume Retail & FinTech Event Streaming Client (Apache Kafka KRaft Cluster + Schema Registry (Avro/JSON) + Kafka Connect + Consumer Groups)

- Designed partitioned Kafka topics, replication factors, and key-based ordering strategies for high-throughput events.
- Built idempotent producers, transactional outbox relays, and manual-commit consumer groups with zero message loss.
- Configured Schema Registry compatibility rules, exponential retry topics, and Dead-Letter Queue (DLQ) replay tooling.
- Monitored consumer lag, broker throughput, and partition rebalancing on a live containerized Kafka cluster.

Aligned Global Vendor Certifications: **CCDAK** (Confluent Certified Developer for Apache Kafka (CCDAK)) • **DVA-C02** (AWS Certified Developer – Associate)

Target Job Roles: Kafka & Event-Driven Backend Engineer, Real-Time Data Streaming Engineer, Distributed Systems Software Engineer, Platform / Messaging Infrastructure Engineer

Realistic Annual CTC Scope: **Associate Streaming / Backend Engineer (0–1 Yr):** ₹6.5 LPA – ₹12.0 LPA **Distributed Microservices / Data Engineer (1–3 Yrs):** ₹13.0 LPA – ₹25.0 LPA **Principal Event-Driven Systems Architect (3–6+ Yrs):** ₹26.0 LPA – ₹48.0+ LPA

MockAttempt Academy Pvt. Ltd. • Email: support@mockattempt.com • Phone/WhatsApp: +91 98765 43210 • Web: https://mockattempt.com/courses/kafka-training
Try 2 Free Topic-Wise 50-MCQ Mock Tests & Spoken Gemini AI Interviews daily on every topic above before enrolling.