Kafka & Flink Training (10 Half-Days x 4 Hours)

Pre-requisite Skills

- Strong Java programming experience
- Familiarity with IBM App Connect Enterprise (ACE)
- Experience with Docker, Kubernetes, OpenShift
- Comfortable using CLI (Linux terminal)
- Understanding of distributed systems and integration concepts
- Familiar with reading/writing structured files (CSV, JSON, Avro)

Infrastructure & Setup

Cloud Platform: Google Cloud Platform (GCP)

- Java 17
- Docker & Docker Compose
- Confluent Kafka Platform
- Apache Flink
- Git, Maven

Networking:

- Open ports: 9092 (Kafka), 2181 (Zookeeper), 8081 (Schema Registry), 8083 (Kafka Connect), 8088 (ksqlDB), 8080 (Flink UI)

⏱ Day 1 (Half-Day): Apache Kafka Fundamentals

1. Introduction to Apache Kafka
    - Overview of Kafka use cases in modern integration systems
    - Kafka as an event streaming platform
    - Kafka ecosystem components
2. Kafka Architecture
    - Brokers, Topics, Partitions, Producers, Consumers
    - Offset tracking and replication
    - Consumer Groups, Rebalancing
    - Fault tolerance and high availability
3. Kafka Setup
    - Launch Kafka & Zookeeper via Docker Compose
    - Validate setup with CLI tools
    - Create topic, produce and consume messages
4. Kafka UI
    - Launch Kafka UI
    - View cluster status, topics, offsets, consumer groups

⏱ Day 2 (Half-Day): Java Clients + Schema Registry

1. Kafka Producer API in Java
    - Producer properties: acks, retries, batch size
    - Create and send messages using custom Java app
2. Kafka Consumer API in Java
    - Polling loop, offset commits, group.id, auto.offset.reset
    - Subscribe and consume messages from topic
3. Introduction to Schema Registry
    - Why use a Schema Registry?
    - Role in enforcing data contracts
    - Format support: Avro
4. Avro Serialization in Java
    - Generate Avro schemas
    - Produce and consume Avro data

⏱ Day 3 (Half-Day): Kafka Connect

1. Kafka Connect Concepts
    - Source & Sink Connectors
    - Standalone vs Distributed mode
2. FileSource → Kafka → FileSink
    - Install and configure connectors using Docker
    - Set up FileSource and FileSink with config files
    - Observe data ingestion and output

⏱ **Day 4 (Half-Day): ksqlDB**

1. Introduction to ksqlDB
   - SQL for Streams
   - Difference between STREAM and TABLE
2. Running ksqlDB
   - Launch ksqlDB server using Docker
   - Register Kafka topic as STREAM
   - Write simple queries (SELECT, FILTER, WHERE)

⏱ **Day 5 (Half-Day): Kafka Advanced + Schema Evolution**

1. Kafka Consumer Group Internals
   - Lag tracking, partition assignment strategies
   - Offset commits and rebalancing
2. Schema Evolution
   - Compatibility modes
   - Forward/backward/full compatibility
   - Schema Registry settings and versioning
3. Kafka Reliability Patterns
   - Idempotent producers
   - Exactly-once semantics (conceptual)

   Debugging & Tuning Kafka Clients
   - Logs and metrics
   - Message retries, timeouts
   - Tuning producer/consumer throughput

⏱ Day 6 (Half-Day): Kafka on OpenShift + Monitoring

1. Kafka on OpenShift
   - Deploy Kafka
   - Create Kafka topic
   - Access Kafka from within OpenShift pod
2. Producing/Consuming from Pod
   - Use CLI tools from pod
   - Validate end-to-end pipeline with Java producer/consumer
3. Kafka Monitoring Tools
   - Monitor consumer lag, partitions, message throughput
   - Use Kafka UI and Conduktor dashboards

⏱ Day 7 (Half-Day): Introduction to Flink

1. What is Apache Flink?
   - Batch vs Stream processing
   - Flink vs Kafka Streams vs Spark Streaming
2. Flink Architecture
   - JobManager, TaskManager, Checkpoints
   - Parallelism and scaling
   - Operator chains and fault tolerance
3. Flink Setup on GCP
   - Launch Flink using Docker
   - Submit a simple Java job
   - Monitor job in Flink Web UI

⏱ Day 8 (Half-Day): Java Stream Processing in Flink

1. Flink DataStream API in Java
   - Stream transformations: map, filter, keyBy, flatMap
2. Java Flink Job
   - Read from file or socket
   - Apply transformation and print results
3. Windowing & Time Semantics
   - Tumbling and Sliding windows
   - Processing time vs Event time
   - Basic window aggregation
   - Checkpointing basics
4. Debugging and Tuning Flink Jobs
   - Task parallelism, thread utilization
   - Backpressure handling
   - Operator chain inspection

⏱ Day 9 (Half-Day): Flink with Kafka

1. Kafka Connector in Flink
   - FlinkKafkaConsumer / FlinkKafkaProducer setup
   - Kafka connector dependencies in Maven
2. Kafka → Flink → Kafka Java Job
   - Read data from Kafka topic
   - Transform and write to new Kafka topic
   - Validate output using Kafka UI

⏱ Day 10 (Half-Day): Best Practices + Capstone

1. Kafka Best Practices
    - Topic design, partitioning, compression
    - Throughput tuning and lag reduction
    - Error handling and retries
2. Flink Best Practices
    - State size management
    - Operator chaining and resource tuning
    - Time handling strategies
3. Capstone Lab
    - Ingest file → Kafka (Avro)
    - Process with Flink (transform + window)
    - Write to Kafka → Query via ksqlDB
    - Validate pipeline via Kafka UI