
1-Month Expert Architecture Training Plan

Chapter 2: Microservices and Bulk Data Ingestion (W1D2)

Prepared by: Sami Mbarki
Solution Architect, Java Expert
ISYGO Consulting Services 19 September 2025

ISYGO Consulting Services

Inspire Success, Your Goals & Opportunities

Document ID: ede2b784-641c-41b2-8f8d-d9307e59fe85

Version: 1.0

Confidential: For Internal Training Use Only

Expert Training in Microservices and Polyglot Persistence

Sami Mbarki
Solution Architect, Java Expert
ISYGO Consulting Services

19 September 2025

Contents

1	Microservices Architecture	2
2	Kafka Connect for Bulk Ingestion	5
3	Lab: Microservice with Kafka Connect	6
4	Exercises	6

1 Microservices Architecture

Microservices decompose applications into independent, loosely coupled services, ideal for the capstone platforms multitenant document processing.

Microservices

Microservices are small, autonomous services focused on specific business capabilities, communicating via APIs or events.

Key Characteristics:

- **Single Responsibility:** Each service handles one function (e.g., upload, analysis).
- **Polyglot Persistence:** Uses tailored databases (e.g., PostgreSQL for metadata, S3 for documents).
- **Independent Deployment:** Deploys without affecting other services.
- **Role in EDA:** Consumes/produces events for decoupling and scalability.

Inputs/Outputs:

- **Inputs:** Events (e.g., DocumentUploaded), API requests (e.g., HTTP POST).
- **Outputs:** Processed events (e.g., AnalyzedDocument), API responses.

Process Flow:

1. Upload service receives document via REST API (e.g., POST /upload).
2. Publishes DocumentUploaded event to Kafka topic tenantA-documents.
3. Analysis service consumes event, processes with LLaMA 3, writes results to Delta Lake.

Benefits:

- **Scalability:** Scale services independently (e.g., 10 upload pods, 20 analysis pods).
- **Resilience:** Isolated failures (e.g., analysis failure doesn't affect uploads).
- **Agility:** Faster development cycles (e.g., 2-week sprints per service).

Example: Upload service stores metadata in PostgreSQL, publishes to Kafka for LLaMA 3 analysis.

Analogy: Microservices are chefs specializing in one dish, working in parallel.

Capstone Connection: Handles tenant-specific uploads, analysis, and archival processes.

Edge Cases:

- **Service Failure:** Use circuit breakers to prevent cascading failures.
- **Interservice Dependency:** Minimize with event-driven communication.

Bounded Context

A bounded context defines a services domain scope, ensuring clear boundaries.

Key Characteristics:

- **Domain Isolation:** Limits scope to specific business capability.
- **Consistency:** Owns data model and event schemas.
- **Role in EDA:** Prevents overlap in event schemas and responsibilities.

Inputs/Outputs:

- **Inputs:** Domain events (e.g., DocumentUploaded).
- **Outputs:** Consistent data model and events.

Process Flow:

1. Define context (e.g., upload service for document ingestion).
2. Design events and schemas (e.g., DocumentUploadEvent).
3. Implement service logic with clear boundaries.

Benefits:

- **Clarity:** Reduces ambiguity in service responsibilities.
- **Decoupling:** Avoids distributed monoliths by isolating domains.

Example: Upload service owns upload events, analysis owns LLaMA 3 results.

Analogy: A bounded context is a department with clear roles and responsibilities.

Capstone Connection: Defines tenant-specific service boundaries for isolation.

API Gateway

An API Gateway routes client requests to microservices, providing a unified entry point.

Key Characteristics:

- **Routing:** Directs requests to appropriate services (e.g., /upload to upload service).
- **Load Balancing:** Distributes traffic across service instances.
- **Security:** Authenticates requests (e.g., OAuth2).
- **Role in EDA:** Fronts HTTP-based services, integrates with Kafka for events.

Inputs/Outputs:

- **Inputs:** Client HTTP requests (e.g., POST /upload).
- **Outputs:** Service responses (e.g., 200 OK, JSON payload).

Process Flow:

1. Client sends request to gateway (e.g., Kong).
2. Gateway authenticates (e.g., JWT), routes to service.
3. Service processes request, responds via gateway.

Benefits:

- **Simplification:** Single entry point for clients.
- **Security:** Centralizes authentication and rate limiting.

Example: Kong routes tenantA uploads to upload service.

Analogy: An API Gateway is a receptionist directing visitors to departments.

Capstone Connection: Routes tenant-specific requests to appropriate services.

Circuit Breaker

A circuit breaker prevents cascading failures by stopping calls to failing services.

Key Characteristics:

- **Fault Tolerance:** Stops calls when failure threshold reached.
- **States:** Closed (normal), open (block calls), half-open (retry).
- **Role in EDA:** Protects event consumers from downstream failures.

Inputs/Outputs:

- **Inputs:** Service calls (e.g., API requests).
- **Outputs:** Success/failure signals or fallback responses.

Process Flow:

1. Monitor service calls for failures (e.g., 500 errors).
2. Open circuit if threshold exceeded (e.g., 5 failures in 10s).
3. Retry after timeout (e.g., 30s), move to half-open.

Benefits:

- **Resilience:** Prevents system-wide failures.
- **Recovery:** Auto-retries improve availability.

Example: Stops calls to failing LLaMA 3 analysis service, returns cached results.

Analogy: A circuit breaker is a fuse preventing electrical overload.

Capstone Connection: Protects tenant services from cascading failures.

Saga

A saga coordinates distributed transactions across microservices using events.

Key Characteristics:

- **Event-Driven:** Uses events to sequence steps.
- **Types:** Choreography (event-based), orchestration (centralized coordinator).
- **Compensating Transactions:** Undoes steps on failure.
- **Role in EDA:** Ensures eventual consistency across services.

Inputs/Outputs:

- **Inputs:** Command events (e.g., UploadStarted).
- **Outputs:** Completed or compensated transactions.

Process Flow:

1. Upload service starts saga, publishes UploadStarted event.
2. Analysis service consumes, processes with LLaMA 3, publishes AnalysisCompleted.
3. Failure triggers compensating event (e.g., UploadCancelled).

Benefits:

- **Consistency:** Ensures eventual consistency without locks.
- **Scalability:** Handles distributed workflows efficiently.

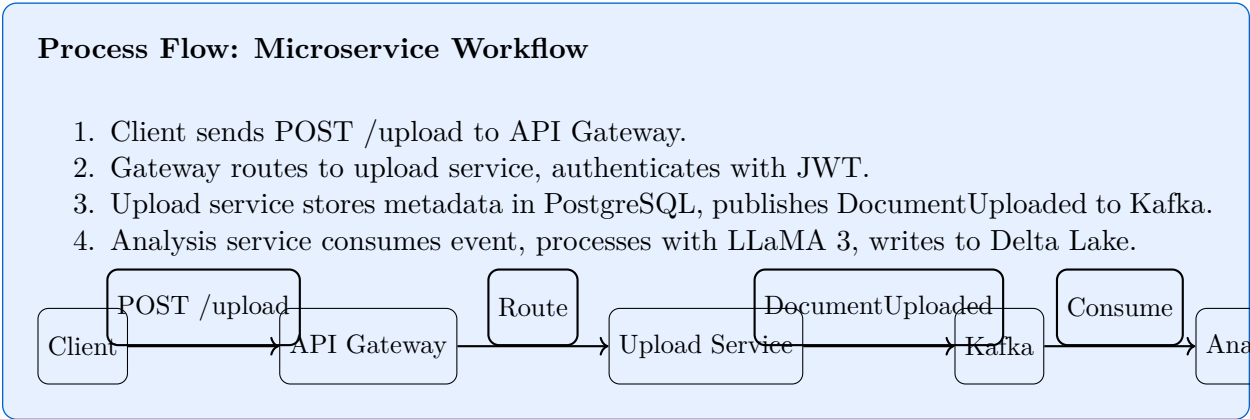
Example: Coordinates upload to archival for tenant documents.

Analogy: A saga is a relay race, passing tasks between runners.

Capstone Connection: Manages tenant workflows (e.g., upload to LLaMA 3 analysis).

Feature	Microservices	Monolith	Serverless
Structure	Independent services	Single codebase	Function-based
Database	Per-service (polyglot)	Shared	Managed
Scaling	Horizontal	Vertical	Auto-scaled
Deployment	Per-service	Big bang	Event-triggered
Complexity	High (distributed)	Moderate	Low
Cost	Infrastructure	Fixed	Pay-per-use

Table 1: Architecture Comparison for Document Processing



2 Kafka Connect for Bulk Ingestion

Kafka Connect integrates external systems with Kafka for bulk data ingestion.

Kafka Connect

Kafka Connect is a framework for streaming data between Kafka and external systems (e.g., S3, databases).

Key Characteristics:

- **Source/Sink:** Source connectors ingest data, sink connectors export data.
- **Scalability:** Distributed workers handle parallel tasks.
- **Fault Tolerance:** Retries and Dead Letter Queues (DLQs) handle failures.
- **Role in EDA:** Bridges external data sources to Kafka events.

Inputs/Outputs:

- **Inputs:** External data (e.g., S3 files, database rows).
- **Outputs:** Kafka events or external writes (e.g., to Delta Lake).

Process Flow:

1. Source connector polls external system (e.g., S3 bucket).
2. Converts data to Kafka records (e.g., Avro format).
3. Writes records to topic (e.g., bulk-uploads).

Benefits:

- **Scalability:** Handles 100K files/day with distributed workers.
- **Reusability:** Pre-built connectors reduce development time.

Example: S3 Source Connector ingests tenant PDFs from S3 to Kafka.

Analogy: Kafka Connect is a conveyor belt moving goods between warehouses.

Capstone Connection: Ingests bulk tenant PDFs for LLaMA 3 processing.

Technical Note: Kafka Connect Optimization

Tune `tasks.max` based on partition count (e.g., 10 tasks for 10 partitions). Use DLQs for failed records. Monitor connector lag with Prometheus.^a

^aDLQs prevent data loss during ingestion failures.

3 Lab: Microservice with Kafka Connect

This lab builds a microservice and configures Kafka Connect for bulk ingestion.

Steps:

1. Develop Spring Boot upload service with REST API.
2. Configure S3 Source Connector to ingest PDFs.
3. Test end-to-end flow with Testcontainers.

4 Exercises

1. Design microservices for upload, analysis, and archival processes.

2. Configure JDBC sink connector to write analysis results to PostgreSQL.
3. Implement a choreography-based saga for upload-to-archival workflow.