



# Confluent Certified Developer for Apache Kafka

**CODICE**  
NOVCCA

**DURATA**  
5 Giorni

**PREZZO**  
2.500,00€ (iva escl.)

**LINGUA**  
Italiano - Inglese

**MODALITÀ**  
Virtual Classroom  
Corso in aula

## SCHEDULAZIONE

- A Richiesta

## OBIETTIVI

- ? Approfondire le funzionalità e l'utilizzo di Apache Kafka.
- ? Esplorare le modalità di integrazione con Java e architetture a microservizi.
- ? Comprendere le best practice per il monitoring e l'operatività di sistemi Kafka.
- ? Il corso prepara alla certificazione, mediante esercitazioni e simulazione d'esame l'ultimo giorno (nel caso i discenti scelgano di sostenere l'esame)
- ? Tutti i capitoli sono dotati di esercizi interattivi guidati in aula, volti ad offrire una conoscenza pratica e non solo teorica degli argomenti.

## CONTENUTI

### 1. Introduzione a Kafka

- ? Concetti fondamentali: Topic, Broker, Partizioni, Replication, Consumer Group, Cluster, Cluster Multipli. Ordine dei Messaggi. Paradigma di base rispetto e confronto rispetto ad altri Message Oriented Middleware
- ? Componenti di Kafka e della Distribuzione Confluent: Cluster, Broker, Connect, kSQLDB, Schema Registry, Rest Proxy
- ? Zookeeper vs. KRaft
- ? Delivery Semantic: At least once, At most once, Exactly Once
- ? Resilienza e gestione degli errori.
- ? In Sync Replicas (ISR).
- ? Log Compaction.

### 2. Architetture Event Driven

- ? Utilizzo di Kafka per migliorare la robustezza e le prestazioni dell'architettura.
- ? Introduzione a pattern fondamentali:
- ? Saga Pattern: coordinamento di transazioni distribuite.
- ? Sistemi di Compensazione: gestione di errori e rollback in scenari complessi.
- ? Fan-Out ed Elaborazioni Parallele: distribuzione di eventi a più consumatori per elaborazioni simultanee.



? CQRS/ES (Command Query Responsibility Segregation/Event Sourcing): separazione tra comandi e query con gestione eventi.

### **3. Installazione di Kafka**

? Configurazione e utilizzo tramite CLI.

? Installazione e gestione con Docker.

? Installazione e configurazione con Confluent

? Presentazione di varie soluzioni tramite container e composizioni Docker

### **4. Kafka Producer**

? Presentazione della Producer API

? Invio di messaggi e utilizzo di serializzatori. Esempi di codice.

? Produzione con Chiave, effetti e casi d'uso. Impatti con il ripartizionamento.

? Configurazioni At Most Once

### **5. Kafka Consumers**

? Sottoscrizione ai topic.

? Gestione di commit e offset. Esempi di codice.

? Configurazioni At Least Once e Exactly Once. Commenti su vantaggi e svantaggi.

? Utilizzo di deserializzatori.

? Avro, Schema Registry. Esempi di codice.

? Transazioni in Kafka. Esempi di utilizzo e caveat.

### **6. Meccanismi interni di Kafka**

? Elaborazione delle richieste.

? Storage e configurazioni per alta affidabilità.

? Kafka Connect. Sink e Source, Esempi pratici. Uso di Connect Hub in Confluent. Esempi di configurazione di Sink e Source.

### **7. Integrazione Kafka con Java**

? Stream API. Concetti di Ktable e Kstream.

? Integrazione di Kafka in applicazioni basate su Java.

? Utilizzo di KSQLDB in confluent. Scrittura di SQL Streaming, mappatura sui concetti di Stream.

? Utilizzo di Flink. Tumbling Window, State Snapshot. Concetti avanzati di Streaming Analytics e architetture di streaming (Non materia d'esame)

### **8. Kafka in Architetture a Microservizi**

? Utilizzo con Spring Boot, Spring Cloud e Spring Streams.

? Esempio pratico di implementazione di un service bus Kafka in un'architettura a microservizi.

### **9. Monitoring e Operations di Kafka**

? Introduzione alla distribuzione Confluent e ai suoi servizi principali.

? Confluent Control Center: monitoraggio delle prestazioni e gestione semplificata di cluster Kafka.

? Metriche di monitoring principali e parametri di configurazione in Confluent relativi (solo per la parte oggetto d'esame)

? Best Practices per ottimizzazione delle performance a livello di tuning (solo per la parte oggetto d'esame)

? Servizi integrati per:

? Monitoraggio di topic, producer e consumer.

? Visualizzazione dei flussi di dati in tempo reale.

? Configurazione e gestione semplificata dei cluster.

? Quote e ACL



? Cenni alla sicurezza (connessioni TLS / Kerberos)

#### **10. Aggiornare Kafka con KRaft**

? Introduzione a KRaft (Kafka Raft) e i cambiamenti rispetto a ZooKeeper.

? Vantaggi principali:

? Semplificazione dell'architettura eliminando la dipendenza da ZooKeeper.

? Miglioramenti nella scalabilità e nella gestione dei metadati.

? Maggiore resilienza grazie al protocollo Raft.

? Procedura per la migrazione a KRaft.

? Best practice e novità introdotte nelle ultime versioni di Kafka.

*Prezzi e corsi potrebbero subire variazioni; si consiglia di verificare sul sito [www.novanext.it/training](http://www.novanext.it/training).*