

Confluent Developer Skills for Apache Kafka®

3jours / 21h

Objectifs pédagogiques

- Écrire aux producteurs et aux consommateurs pour envoyer et lire des données à Kafka
- Intégrer Kafka à des systèmes externes à l'aide de Kafka Connect
- Écrire des applications de streaming avec Kafka Streams & ksqlDB
- Intégrer une application cliente Kafka avec Confluent Cloud

Modalités d'évaluation : Les objectifs pédagogiques sont évalués à travers la réalisation des parties pratiques (labs dirigés) sous la supervision du formateur délivrant la session de formation.

Public cible

Ce cours est conçu pour les développeurs d'applications, les développeurs ETL (extraction, transformation et chargement) et les data scientists qui ont besoin d'interagir avec les clusters Kafka en tant que source ou destination de données.

Prérequis

Les participants doivent être familiarisés avec le développement d'applications professionnelles en Java (de préférence), C# ou Python.

De plus, une connaissance pratique de l'architecture Apache Kafka est requise pour ce cours, soit par :

- Expérience antérieure, ou
- En suivant Confluent Fundamentals pour Apache Kafka, accessible ici : <https://training.confluent.io/schedule>.

Pour évaluer vos connaissances Kafka pour ce cours, veuillez remplir l'auto-évaluation sur : <https://cnfl.io/fundamentals-quiz>

Programme

Module 1 : Fondamentaux d'Apache Kafka®

- Expliquez la valeur d'une plate-forme de diffusion d'événements distribués
- Expliquez comment l'abstraction de « journal » permet une plate-forme de diffusion d'événements distribuée
- Expliquer les concepts de base de : Brokers, Topics, Partitions et Segments, Records (également appelés messages, événements), politiques de rétention, producteurs, consommateurs et sérialisation, réplication, Kafka Connect

Module 2 : Produire des messages pour Kafka

- Esquissez l'architecture de haut niveau d'un producteur Kafka
- Illustrer le partitionnement basé sur les clés
- Expliquez la différence entre `acks=0`, `acks=1` et `acks=all`
- Configurer `delivery.timeout.ms` pour contrôler le comportement des nouvelles tentatives
- Créer un fichier `producer.properties` personnalisé
- Ajustez le débit et la latence à l'aide du traitement par lots
- Créer un producteur avec Confluent REST Proxy

Module 3 : Consommer des messages de Kafka

- Illustrer comment les groupes de consommateurs et les partitions offrent une évolutivité et une tolérance aux pannes
- Accordez les consommateurs pour éviter les rééquilibrages excessifs
- Expliquer la différence entre les stratégies d'attribution de partition « range » et « round robin »
- Créer un fichier `consumer.properties` personnalisé
- Utiliser l'API Consumer pour gérer les offsets

- Ajuster les requêtes fetch
- Créer un consommateur avec Confluent REST Proxy

Module 4 : Gestion des schémas dans Apache Kafka®

- Décrire les schémas Kafka et leur fonctionnement
- Utilisez le Confluent Schema Registry pour guider l'évolution du schéma
- Écrire et lire des messages à l'aide de Kafka compatible avec les schémas

Module 5 : Traitement de flux avec Kafka Streams

- Comparer KStreams à KTables
- Créer un fichier `streams.properties` personnalisé
- Expliquer ce qu'est le co-partitionnement et pourquoi il est important
- Écrire une application utilisant le Streams DSL (Domain-SpecificLanguage)

Module 6 : Pipelines de données avec Kafka Connect

- Expliquer la motivation de Kafka Connect
- Liste des connecteurs couramment utilisés
- Expliquer les différences entre le mode autonome et le mode distribué
- Configurer et utiliser Kafka Connect

Module 7 : Applications de streaming d'événements avec ksqlDB

- Utiliser ksqlDB pour filtrer et transformer un flux
- Écrire une requête ksqlDB qui joint deux flux ou un flux et une table
- Écrire une requête ksqlDB qui agrège les valeurs par clé et fenêtre de temps
- Rédiger des requêtes Push et Pull et expliquer les différences entre elles
- Créer un connecteur avec ksqlDB

Module 8 : Décisions de conception

- Répertorier les moyens d'éviter les messages volumineux
- Décider quand utiliser ksqlDB vs Kafka Streams vs Kafka Connect SMT
- Expliquer les différences et les compromis entre les garanties de traitement
- Traiter les décisions qui découlent du partitionnement basé sur les clés
- Authentifier une application cliente avec un cluster Kafka sécurisé

Module 9 : Confluent Cloud

- Expliquer ce que signifie « fully-managed » dans le contexte de Confluent Cloud
- Authentifier un client Kafka sur Confluent Cloud
- Effectuer des opérations de base avec la CLI `ccloud`