

Программа курса «DEVKI: Apache Kafka для инженеров данных»

О курсе: 5-дневный практический интенсив предназначен для разработчиков и инженеров, которые хотят не просто изучить теорию, а получить реальные навыки программирования для распределенной обработки потоков событий (Event Stream Processing). Вы научитесь писать код для работы с распределенным журналом Kafka, использовать его в парадигмах Publish/Subscribe и Message Queue, а также освоите лучшие практики обработки потоковых данных. Наше Apache Kafka обучение построено на реальных кейсах, что позволит вам сразу после курса применять знания в рабочих проектах.

Аудитория:

- Developer Java, Python, Scala
- DevOps инженер
- Дата Инженер
- Архитекторы

Уровень подготовки:

- Начальный опыт работы в Unix
- Начальный опыт программирования на Java
- Опыт работы с Hadoop Distributed File System (желательно)

Продолжительность курса: 24 академических часа, 5 дней по 4, 8 ак. часов дистанционно

Содержание программы

1. Введение в Apache Kafka: ключевые концепции и архитектура

Введение в Apache Kafka:

- Архитектура и история возникновения Kafka.
- Основные компоненты: брокеры, Zookeeper, KRAFT mode.
- Основные API Apache Kafka и принципы их взаимодействия и killer features.

Topic, partition и offset в Apache Kafka:

- Управление топиками.
- Отслеживание последнего прочитанного сообщения.
- Семантика доставки сообщений (at least once, at most once, exactly once)

Комплексная практика:

- **Работа из командной строки:** Управление топиками и обмен сообщениями через консольные утилиты Kafka.

2. Взаимодействие с Kafka: Producer и Consumer API

Producer API:

- Настройка и использование Producer API для публикации сообщений.

Consumer API:

- Принципы работы Consumer API
- Consumer Group для параллельной обработки.
- Сериализация данных.

Комплексная практика:

- **Разработка Producer/Consumer:** Создание базового приложения для отправки и чтения данных на **Java** и **Python**.

3. Поточковая обработка: Kafka Streams DSL

Потоковая обработка:

- Архитектура Kafka Streams.

- Основные абстракции Kafka Streams
- Изучение декларативного DSL для описания трансформаций.
- Обзор stateless и stateful операций (фильтрация, агрегация, оконные функции).

Комплексная практика:

- **Конвейер обработки на Kafka Streams DSL:** Построение потокового приложения для трансформации данных в реальном времени.
- **Сериализация Avro:** Пример типизированной сериализации данных с использованием **Schema Registry** и **Avro**.
- **ksqlDB:** Интерактивные запросы к потокам данных в Kafka с помощью **ksqlDB**.

4. Продвинутая обработка: Processor API

Processor API:

- Изучение низкоуровневого **Processor API** для создания кастомных топологий.
- Работа с **State Store** и использование **Punctuator** для задач по расписанию.

Комплексная практика:

- **Кастомная логика на Processor API:** Реализация сложной логики обработки с помощью низкоуровневого API.

5. Интеграция с экосистемой: Kafka Connect

Kafka Connect:

- Архитектура Kafka Connect.
- Обзор доступных коннекторов и их режимов работы (standalone и distributed).

Комплексная практика:

- **Интеграция через Kafka Connect:** Настройка готовых коннекторов для интеграции Kafka с файловой системой.