Over 35 Years Of Technology Training

Document Generated: 04/02/2026

Learning Style: Virtual Classroom

Technology:

Difficulty: Beginner

Course Duration: 2 Days

Introduction to Apache Kafka for Developers (TTDS6760)

About This Course:

Apache Kafka is a real-time data pipeline processor. Its high-scalability, fault tolerance, execution speed, and fluid integrations are some of the key hallmarks that make it an integral part of many Enterprise Data architectures.

Geared for experienced Java developers, Introduction to Apache Kafka for Developers is a fast-paced, lab-intensive two day hands-on course that explores the potential of fast data and streaming systems, and how to navigate the complexities of modern streaming architectures. Throughout the course you'll explore the ins and outs of Apache Kafka and learn how it compares to other queue systems like JMS and MQ. You'll learn about Kafka's unique architecture and understand how to effectively produce and consume messages with Kafka & Zookeeper. Through hands-on labs, you'll gain experience in scaling Kafka, navigating multiple data centers, and implementing disaster recovery solutions, while exploring essential Kafka utilities.

You'll also learn the powerful Kafka APIs and become proficient in configuration parameters, Producer and Consumer APIs, as well as advanced features such as message compression and offset management. Gain hands-on with Kafka, including benchmarking Producer send modes, comparing compression schemes, and managing offsets. Experience real-world applications like Clickstream processing to solidify your expertise. Then you'll round off your Kafka journey with an in-depth look at the Kafka Streams API, monitoring, and troubleshooting techniques. You'll learn how to optimize your Kafka deployment with best practices for hardware selection, cluster sizing, and Zookeeper settings.

Course Objectives:

- Implement and configure Apache Kafka effectively, demonstrating a deep understanding of its unique architecture, core concepts, and the differences between Kafka and other queue systems (JMS/MQ).
- Utilize Kafka APIs proficiently, including the Producer and Consumer APIs, and apply advanced techniques such as message compression, offset management, and Producer send modes.
- Design and develop streaming applications using the Kafka Streams API, performing complex operations like transformations, filters, joins, and aggregations, while working with KStream, KTable, and KStore concepts.
- Monitor and troubleshoot Kafka deployments, identifying performance bottlenecks, addressing common issues, and employing best practices for hardware selection, cluster sizing, partition sizing, and Zookeeper settings.
- Apply the skills and knowledge acquired throughout the course to real-world scenarios, showcasing the ability to develop, deploy, and optimize Kafka-based streaming applications for a variety of use cases.

Audience

- This course is geared for experienced Java Developers and architects with Java development background who are new to Kafka. This course is not for non-developers.

Prerequisites:

- Basic Java programming skills; practical Java development background.
- Reasonable experience working with databases
- Basic Linux skills and the ability to work from the Linux command line
- Basic knowledge of Linux editors (such as VI / nano) for editing code.

Course Outline:

Getting Started with Streaming Systems

- Understanding Fast data
- Streaming terminologies
- Understanding at-least-once / at-most-once / exactly-once processing patterns
- Popular streaming architectures
- Lambda architecture
- Streaming platforms overview
- Lab: Hands-on first look at Kafka

Introducing Kafka

- Comparing Kafka with other queue systems (JMS / MQ)
- Kafka Architecture
- Kaka concepts: Messages, Topics, Partitions, Brokers, Producers, commit logs
- Kafka & Zookeeper
- Producing messages
- Consuming messages
- Consumers, Consumer Groups
- Message retention
- Scaling Kafka
- Kafka across multiple data centers and disaster recovery
- Lab: Getting Kafka up and running
- Lab: Using Kafka utilities

Using Kafka APIs

- Configuration parameters
- Producer API - sending messages to Kafka
- Consumer API - consuming messages from Kafka
- Producer send modes
- Message compression
- Commits , Offsets, Seeking

- Managing offsets - auto commit / manual commit
- Lab: Writing Producer / Consumer
- Lab: Benchmarking Producer send modes
- Lab: Comparing compression schemes
- Lab: Managing offsets
- Lab: Clickstream processing

Kafka Streams API

- Introduction to Kafka Streams library
- Features and design
- Streams concepts: KStream / KTable / KStore
- Streaming operations (transformations, filters, joins, aggregations)
- Using Streams API: foreach / filter / map / groupby
- Lab: Kafka Streaming APIs

Monitoring & Troubleshooting Kafka

- Monitoring tools overview
- Monitoring Kafka
- Cluster level and host level monitoring
- Identifying performance bottlenecks
- Troubleshooting common Kafka issues

Bonus Content / Time Permitting

Kafka Best Practices

- Avoiding common mistakes
- Hardware selection
- Cluster sizing
- Partition sizing
- Zookeeper settings
- Compression and batching
- Message sizing
- Monitoring and instrumenting
- Troubleshooting