Course Objectives

During this hands-on course, you will learn to:

• Identify common patterns and use cases for real-time stream processing
• Understand the high level architecture of Apache Kafka® Streams
• Write real-time applications with the Kafka Streams API to filter, transform, enrich, aggregate, and join data streams
• Describe how ksqlDB combines the elastic, fault-tolerant, high-performance stream processing capabilities of Kafka Streams with the simplicity of a SQL-like syntax
• Author ksqlDB queries that showcase its balance of power and simplicity
• Test, secure, deploy, and monitor Kafka Streams applications and ksqlDB queries

Hands-on Training

Throughout the course, you will interact with hands-on lab exercises to reinforce stream processing concepts.

Exercises include:

• Anatomy of a Kafka Streams Application
• Joining Two Streams
• Using the Kafka Streams Processor API
• Testing a Kafka Streams Application
• Using ksqlDB
• Using the ksqlDB REST API
• Scaling a Kafka Streams Application
• Securing a Kafka Streams Application
• Getting Metrics from a Kafka Streams Application
• Using JConsole to monitor a Kafka Streams Application
• Monitoring a Kafka Streams Application in Confluent Control Center

Prerequisites

Attendees should be familiar with developing professional apps in Java (preferred), .NET, C#, Python, or another major programming language.

Additionally, students require a strong knowledge of Kafka architecture as well as knowledge of Kafka client application development, either through:

• Prior experience, or
• By taking the recommended prerequisites: Confluent Fundamentals for Apache Kafka® and Confluent Developer Skills for Building Apache Kafka®

Participants are required to provide a laptop computer with unobstructed internet access to fully participate in the class.

To sign-up for one of our courses, visit us here.

Who Should Attend?

This course is designed for application developers, architects, DevOps engineers, and data scientists who need to interact with Kafka clusters to create real-time applications to filter, transform, enrich, aggregate, and join data streams to discover anomalies, analyze behavior, or monitor complex systems.
## Content

<table>
<thead>
<tr>
<th>MODULE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
</table>
| Motivation and Concepts for Streams | • Motivation and Use Cases for Real-Time Streaming  
• High Level Comparison of Kafka Streams and ksqlDB  
• Stream Processing Concepts |
| Kafka Streams Architecture | • Kafka Streams' Place in the Kafka Ecosystem  
• High Level Architecture Design  
• Kafka Streams Data Types |
| Writing Kafka Streams Applications | • Anatomy of a Kafka Streams Application  
• Kafka Streams DSL — Stateless Operations  
• Kafka Streams DSL — Aggregations  
• Kafka Streams DSL — Windowed aggregations  
• Kafka Streams DSL — Joins  
• Kafka Streams DSL — Summary  
• Processor API  
• Optimizations |
| Testing Kafka Streams Applications | • Get streams of data into and out of Kafka with Kafka Connect and REST Proxy  
• Maintain data formats and ensure compatibility with Schema Registry and Avro  
• Build real-time streaming applications with Confluent ksqlDB & Kafka Streams |
| The Confluent Platform | • Unit Tests  
• Integration Tests  
• Stress Tests  
• End-to-end Tests |
| Introduction to ksqlDB | • Sample Use Cases  
• End-to-end Examples  
• Interacting with ksqlDB |
| Using ksqlDB | • Data Manipulation  
• Aggregations  
• Testing |
| Deployment | • Parallellism  
• Elasticity  
• Fault tolerance  
• Capacity planning  
• Troubleshooting  
• ksqlDB-specific considerations |
| Security | • Security Overview  
• Access Control  
• Examples  
• ksqlDB-specific considerations |
| Monitoring | • JMX  
• Confluent Control Center  
• ksqlDB-specific Considerations |

Confluent offers instructor-led courses in both traditional and virtual classroom formats, as well as in a self-paced format available through the Confluent Self-Paced Subscription. Visit confluent.io/training for more information.

**Disclaimer:** Subscriptions purchased with Training Credits will have an end date matching the Training Credits order.