Course Objectives

During this instructor-led, hands-on course, you will learn how to use Confluent KSQL to transform, enrich, filter and aggregate streams of real-time data using a SQL-like language. You will also learn how to use the Apache Kafka® Streams library to build streaming applications. Furthermore, you will learn how to test, monitor, secure and scale those streaming applications. You will learn how these applications integrate with the Confluent Streaming platform powered by Apache Kafka®, Kafka Connect, Confluent Schema Registry and Confluent REST Proxy as well as the Confluent Control Center. You will learn the role of streaming in the modern data distribution pipeline, discuss architectural concepts and components of KSQL and Kafka Streams.

Hands-on Training

Throughout the course, hands-on exercises reinforce the topics being discussed. Exercises include:

- Installing KSQL containerized and natively
- Transforming and aggregating data streams with KSQL
- Writing Kafka Streams App using DSL and Processor API
- Testing a Kafka Streams App
- Monitoring a Kafka Streams App
- Securing a Kafka Streams App
- Scaling a Kafka Streams App

Course Duration

This is a three-day training course.

Who Should Attend?

This course is designed for application developers and architects, DevOps engineers and data scientists who need to interact with Kafka clusters as a source of real-time data streams and transform, enrich and join those streams to discover anomalies, analyze behavior or monitor complex systems.

Course Prerequisites

Attendees should be familiar with developing professional apps in Java (preferred), .NET C# or Python. Attendees should also be familiar with the essentials of Apache Kafka.

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

Confluent offers public training in class, online and on demand. Please visit **http://confluent.io/training** for more information.

For inquiries about on-site training, please email **training-admin@confluent.io**

Course Content

Fundamentals

- Application Log
- Log Replication
- Topics, Partitions and Segments
- Kafka Streams
- Stream–Table Dualism
- Stream Processing Jobs

KSQL Use Cases

- Why KSQL
- Sample Use Cases
- KSQL and Licensing

KSQL Overview & Ecosystem

- KSQL and Kafka = easy
- Interactive KSQL Usage
- KSQL Architecture
- KSQL CLI
- KSQL Server Modes
- KSQL & Confluent Control Center

Installing KSQL

- Installing using Containers
- Installing natively

Using KSQL

- Kafka Streams and Tables
- Kafka Message and Data Formats
- Data Manipulation and Aggregation
- User Defined Functions
- Data Enrichment and Joins
- Windowed Aggregations
- Metrics and Observability
- Testing and Monitoring
- Tips, Pitfalls and Limitations

Deploying & Operating KSQL

- Best Practices & Patterns
- KSQL and Security
- Elasticity and Scalability
- Fault Tolerance
- Health Checks

Kafka Streams Architecture

- Motivation and Evolution
- Characteristics
- Freedom of Choice

Kafka Streams Application Anatomy

- Streams App Anatomy
- Streams App Configuration
- Streams App Topology

Kafka Streams DSL

- Stateless and Stateful Operations
- Kafka Streams DSL
- Windowed Operations
- Processor API

Testing Kafka Streams Apps

- Test Categories
- Unit Tests
- Integration Tests with Test Driver and Embedded Kafka

Monitoring Kafka Streams Apps

- Using JMX based Monitoring
- Using Confluent Control Center for Monitoring

Securing Kafka Streams Apps

- Why Security is needed
- Security Overview
- Client-Side Security Features
- Required ACLs
- Encryption in Transit

Sizing & Scaling Kafka Streams Apps

- Elastic Scaling
- How many App Instances?
- Memory Management
- Sizing and Task Placement
- Stateless versus Stateful
- Troubleshooting

Using the KSQL REST API

- Sample Request
- Sample Response