Introduction to Kafka with Confluent

Apache Kafka is an open-source stream-processing and messaging platform created by LinkedIn and donated to the Apache Software Foundation. The Kafka supports a standard, high-throughput; low-latency platform for handling real-time data feeds and supports clients across multiple different programming platforms.

Confluent is wrapper around Kafka with features as enterprise-ready Event Streaming for Publish and subscribe to stream records, rest proxy with Kafka, control center, ksql mapping etc.

Objectives

To introduce the Apache Kafka and Confluent platform to developers and testing team.

Prerequisite Knowledge

The participants must have sound Java/ MS.Net programming skills with application messaging knowledge and having knowledge on SQL commands, REST API is required.
The testing team must be conversant with java and Eclipse environment in addition to above prerequisites**.**

Target population

Maximum 20 participants with each one working on separate System having the preconfigured set up as specified below.

Training Methodology

The theoretical topics are discussed interactively and technical details are demonstrated with practical examples.
The participants work on the hands on case studies which strengthen the concepts learned**.**
Each topic is supplemented with practical demonstrations and hands on exercises for the participants.

Class room setup

The Intel core **c**ompatible CPU with Windows 10/Ubuntu Linux 18.04 64 bit system having minimum 8GB RAM and 500GB HDD. Other required software JDK1.8 64 bit, Google Chrome/Firefox mozilla latest browser, Adobe Acrobat Reader to be installed. For Window 10 systems the **Windows Subsystem for Linux** (WSL)must be installed.
The zip distributions for Kafka and other tools will be shared.

Live internet connection with reasonable speed and download permissions is required to download the dependencies and tools.

The participants must have admin rights on their systems.

Course Duration: Three Days

Instructor: Prakash Badhe.

Course outline
This course plan is based on the discussions with the team at client.

Day1

Introduction to Apache Kafka

- Application messaging in Enterprise
- JMS, RabbitMQ, ActiveMQ overview
- Required efficiency and throughput of Messaging platform
- Kafka features, architecture and eco system
- Terminologies
  - Broker
  - Cluster
  - Publish-Subscribe model of communication
  - Producer, consumer and groups
  - Topics, partitions and replications
  - Index and offsets
  - Streams
- Kafka in enterprise
- Kafka configuration and Zookeeper
- Starting Zookeeper and Kafka server
- Sample messaging with kafka
- Kafka logs
- Data storage for topics with partitions
- Offsets and indexes for consuming the data from topics
- Parallelism for consumers in Kafka
- Delivery Semantics for consumers in same and different groups

Kafka API and Usage

- Kafka clients with java and .Net
- Producer and Consumer API
- Connector API for producer and consumers
- Kafka Connect for import/export
- Streams API to work with input streams
- Kafka Broker Discovery
- Kafka Guarantees for delivery
- Topic replication in Kafka cluster
- Failover protection in Kafka cluster of brokers

Kafka streams

- Kafka Streams Architecture
- Processor topology for the stream processing
- Source and Sink Processors
- The stream partitions **and** stream tasks
- Receiving data feed via streaming on Kafka
- Kafka throughput, low latency and efficiency

Kafka Use Cases

- Failover and guaranteed communication across applications/users
- Publish data to any number of systems or applications
- Match supplier and clients in B2B applications for large volume of data
- Real time analytics of usage criteria
- Predictive behavior based on input data
- Execute number of real time services in backend for social media applications
- Manage real time backend operations in online shopping portals
- Kafka in analytics and AI applications

Day2 and Day3

Kafka Advanced with Confluent

- Confluent Introduction and features
- Confluent CLI
- Confluent Platform Quick Start
- Kafka extended API
- Expose REST API for Kafka with Confluent rest proxy
- Produce and consume messages with REST API
- With REST view the state of the cluster
- Manage Kafka with REST API
- Load balancing with multiple instances
- Control Center

Kafka with Schemas for Records

- Message serialization and de-serialization for Kafka records
- Standard and custom data formats for kafka records
- Apache Avro for custom data schemas
- The writer schema and reader schema and versioned schemas
- Use AVRO schema in producer and in Kafka streams.
- SchemaStore implementation
- Reading and Writing with Kafka Topic

- Using versioned schemas
- Schema Registry usage

KSQL: Streaming SQL engine for Kafka

- Stream processing with SQL statements.
- The KSQL architecture
- KSQL Server and CLI
- KSQL components
    - KSQL engine
    - REST interface
    - KSQL CLI
    - KSQL UI
- Confluent Hub client to install add ons
- Install and start Kafka Connect Datagen source connector
- Create KSQL Tables and Streams with options
- Ktable config options
    - Encoding
    - Key
    - Fields
- KSQL Language
- DDL and DML Statements
- Stream, table, STRUCT types
- Units and formats
- Time and Windows
- RUN SCRIPT command
- Configure mock data
- Identify data triggers
- Write and run queries with the KSQL tab in Control Center
- Monitor the logs on consumers
- Hands on queries with dummy data

Testing Strategies with Kafka

- Use cases for testing
- Testing with APIs and consoles
- Testing for published data acknowledgements
- Testing for consumers and groups
- Test with zerocode and Junit
- Explore testing with kafka-streams-test-utils with TopologyTestDriver
- Testing for data transformation and validation
- Testing with KSQL
- Testing with streams

- Testing the streaming joins
- Integration testing with EmbeddedKafkaCluster
- Best practices for testing

Performance Monitoring Overview

- Analyze the data throughput and efficiency
- Tracking metrics from
    - Brokers
    - Consumer
    - Producers
    - Zookeeper,
- Monitoring platforms: Overview on LinkedIn's Burrow and Datadog

Best Practices with Kafka

- Benchmark the performance
- Number of broker nodes in cluster
- Load balancing strategies
- Manage topics and partitions
- Consuming with streams
- Manage the offsets and indexes
- Control the data size
- Data formats and schemas
- Cumulative Backups

****