1. What is kafka streams?

Kafka Streams is a data processing and transformation library that is built within the Kafka ecosystem. It is not an external library created by a third party but rather an integral part of Kafka itself. With Kafka Streams, you can create various types of applications such as data transformation, data enrichment, threat detection, monitoring, and alerting. Kafka Streams is implemented as a standard Java application and can be launched like any other Java application. Unlike other stream processing frameworks like

Spark, Flink, or Nifi, Kafka Streams does not require the creation of a separate cluster. It leverages the scalability, elasticity, and fault-tolerance features provided by Kafka, making it highly reliable. One of the standout features of Kafka Streams is its support for exactly-once processing semantics.

This means that it guarantees that each record is processed exactly once, which is a significant advantage in the streaming world. Kafka Streams processes records individually in real-time, without any batching, making it a true streaming solution. Another benefit of Kafka Streams is its ability to handle applications of any size.

Whether you are working on a small project or a large-scale enterprise application, you can write the same code and achieve consistent scalability and performance. Overall, Kafka Streams is a powerful and versatile library that allows you to build real-time data processing applications within the Kafka ecosystem, providing seamless integration and access to Kafka's robust features.

1. First Kafka Streams Application: WordCount

rem download kafka at https:\\www.apache.org\dyn\closer.cgi?path=\kafka\0.11.0.0\kafka\_2.11-0.11.0.0.tgz

rem extract kafka in a folder

rem WINDOWS ONLY

rem open a shell - zookeeper is at localhost:2181

bin\windows\zookeeper-server-start.bat config\zookeeper.properties

rem open another shell - kafka is at localhost:9092

bin\windows\kafka-server-start.bat config\server.properties

rem create input topic

bin\windows\kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic streams-plaintext-input

rem create output topic

bin\windows\kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic streams-wordcount-output

rem start a kafka producer

bin\windows\kafka-console-producer.bat --broker-list localhost:9092 --topic streams-plaintext-input

rem enter

kafka streams udemy

kafka data processing

kafka streams course

rem exit

rem verify the data has been written

bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic streams-plaintext-input --from-beginning

rem start a consumer on the output topic

bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 ^

--topic streams-wordcount-output ^

--from-beginning ^

--formatter kafka.tools.DefaultMessageFormatter ^

--property print.key=true ^

--property print.value=true ^

--property key.deserializer=org.apache.kafka.common.serialization.StringDeserializer ^

--property value.deserializer=org.apache.kafka.common.serialization.LongDeserializer

rem start the streams application

bin\windows\kafka-run-class.bat org.apache.kafka.streams.examples.wordcount.WordCountDemo

rem verify the data has been written to the output topic!

Softs:

Apache Maven, so just have Apache Maven download Apache Maven is a way for us to manage our dependencies for programs and this is really needed to make sure that we package and build and bringing the dependencies we need for our application.

IntelliJ