**Apache kafka**

**Abstract**

Apache kafka is a distributed streaming platform.Kafka is run as a cluster on one or more services that can span on multiple data centers.Kafka cluster stores streams of records in categories called topics.Each record consist of a key,timestamp and value.Kafka is an open source system.It handles millions of messages.

**History:**

**Apache Kafka** is an open source stream processing software platform developed by LinkedIn and donated to the Apache software Foundation, written in Scala and Java. The project aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds.

**Kafka has 4 core APIs:**

1. The **Producer API** allows an application to publish a stream of records to one or more Kafka topics.

A topic is a category or feed name to which records are published. Topics in Kafka are always multi-subscriber; that is, a topic can have zero, one, or many consumers that subscribe to the data written to it.

Producers publish data to the topics of their choice. The producer is responsible for choosing which record to assign to which partition within the topic. This can be done in a round-robin fashion simply to balance load or it can be done according to some semantic partition function (say based on some key in the record). More on the use of partitioning in a second.

1. The **Consumer API** allows an application to subscribe to one or more topics and process the stream of records produced to them.

Consumers label themselves with a *consumer group* name, and each record published to a topic is delivered to one consumer instance within each subscribing consumer group. Consumer instances can be in separate processes or on separate machines.

If all the consumer instances have the same consumer group, then the records will effectively be load balanced over the consumer instances.

If all the consumer instances have different consumer groups, then each record will be broadcast to all the consumer processes.

1. The **Streams API** allows an application to act as a *stream processor*, consuming an input stream from one or more topics and producing an output stream to one or more output topics, effectively transforming the input streams to output streams.
2. The **Connector API** allows building and running reusable producers or consumers that connect Kafka topics to existing applications or data systems. For example, a connector to a relational database might capture every change to a table.

**Requirements:**

* at least 8 GB RAM
* at least 500 GB Storage
* Ubuntu 14.04 or later, RHEL 6, RHEL 7, or equivalent
* Access to Kafka (specifically, the ability to consume messages and to communicate with Zookeeper)
* Access to Kafka Connect instances (if you want to configure Kafka Connect)
* Ability to connect to the server from the user’s web browser.
* Kafka Brokers running Kafka Version 0.10.1.0 or later
* Kafka Java Producers and Consumers running 0.10.1.0 or later
* Google Chrome

Aishwarya s. shirse

(36071)