Introduction to Kafka

What is Kafka and services it provides?

Kafka is used for building real-time data pipelines and streaming apps. It is horizontally scalable, fault-tolerant, wicked fast, and runs in production in thousands of companies.

* Kafka is run as a cluster on one or more servers that can span multiple datacenters.
* The Kafka cluster stores streams of *records* in categories called *topics*.
* Each record consists of a key, a value, and a timestamp.

Uses of Kafka:

1. As a messaging System - Pub-Sub model
2. As a storage system - Kafka connect for pulling and pushing from source and sink data system
3. Kafka for stream processing - Kafka stream for processing, transforming on stream from one or more topic.

API’s provided by Kafka for above functionality in respective order:

1. Kafka-client for producer-consumer.
2. Many users of Connect won't need to use this API directly, though, they can use pre-built connectors without needing to write any code. Additional information on using Connect is available [here](https://kafka.apache.org/documentation.html#connect).
3. Kafka-stream for Stream processing.

Concepts and Terminologies of Kafka system:

1. Zookeeper – Orchestration, management of Kafka cluster.
2. Topic – Feed name where records are published.
3. Partitions – Each topic is further distributed into partitions for parallelism.
4. Replication factor – Number of nodes in cluster where topic data must be distributed.
5. Offsets – Pointer/Position of consumer in the log/partition.
6. Producers
7. Consumers
8. Consumer groups – Consumers are assigned to consumer groups,

# Consumer in consumer groups should be <= partitions of a topic.

1. Retention policy – Period (mostly in days) for which the record must be available in topic.
2. Leader and follower – Every partition has one broker node in cluster as leader where record is written and fetched from. While follower nodes simply replicate the partitions on their respective broker.

Installing Kafka Cluster on docker

To install Kafka Cluster on Docker container run below commands in CMD or PowerShell

1. docker run -d --name zookeeper -p 2181:2181 zookeeper

This will fetch and run official docker image of zookeeper from docker hub in docker container and port bind it to 2181 between target and host port

1. docker run -d --name kafka\_0 -e KAFKA\_ZOOKEEPER\_CONNECT=<Machine IP>:2181 -e KAFKA\_ADVERTISED\_LISTENERS=PLAINTEXT://<Machine IP>:9092 -e KAFKA\_BROKER\_ID=0 -p 9092:9092 confluentinc/cp-kafka:5.1.2

KAFKA\_ZOOKEEPER\_CONNECT: Tells Kafka broker where the zookeeper instance is running.

KAFKA\_ADVERTISED\_LISTENERS: Tells outer clients where to communicate with Kafka.

BROKER\_ID: Id with which current broker is registered with zookeeper.

This will fetch and run Kafka broker image provided by “*confluent platform”* with port binding on 9092.

**Note:** Run above command with different broker id and Kafka port binding for running multiple instances of Kafka.

1. Working with Topics. Go inside any of the Kafka container and create, list, describe or delete topics with below commands.

docker exec -it <container> /bin/bash

kafka-topics --list --zookeeper <Machine IP>:2181

kafka-topics --create --topic <topic name> --replication-factor 3 --partitions 1 --zookeeper <Machine IP>:2181

kafka-topics --describe --topic <topic name> --zookeeper <Machine IP>:2181

kafka-topics --delete --topic <topic name> --zookeeper <Machine IP>:2181

1. Create producer and consumer in different console and play with topics.

kafka-console-producer --broker-list <Machine IP>:9092 --topic <topic name>

kafka-console-consumer --bootstrap-server <Machine IP>:9092 --topic <topic name>

Kafka readings for the interested

* Official Kafka Website explains concepts really well

https://kafka.apache.org/

* Setting up multi node cluster on docker understanding

<https://www.agiratech.com/kafka-zookeeper-multi-node-cluster-setup/>

<https://blogs.perficient.com/2017/05/25/how-to-install-kafka-cluster-in-the-docker-containers/>

* Env configuration for confluent docker image

<https://docs.confluent.io/current/installation/docker/docs/config-reference.html>

* Kafka Listener understanding

<https://rmoff.net/2018/08/02/kafka-listeners-explained/>

* Understanding consumer groups

<https://dzone.com/articles/dont-use-apache-kafka-consumer-groups-the-wrong-wa>