[KAFKA 기본 개념]

Kafka 같은 스트리밍 기술로 데이터가 생성되는 즉시 클러스터에 가져올 수 있음.

Kafka 서버의 클러스터를 구성해서 웹 서버나 센서 등의 발행자로부터 오는 모든 메시지를 저장하고 동시에 필요한 대상에게 발행

Kafka connector : DB를 위한 플러그인 모듈

- Connector는 생산자의 데이터를 kafka에 끊임없이 발행하고 싶을 때 사용하거나 DB를 kafka에 연결해 새로운 데이터를 수신하고 그 DB 테이블의 새로운 행으로 가져온다.
- 웹 서버로부터 읽은 데이터를 처리해 분산하고 여러 그룹에 복사하는 것이 가능

< kafka 명령어 > (sandbox hortonworks 서버 환경)

1. Kafka 바이너리 파일과 구성 파일이 어디에 설치되어있는지 확인

   ▶ **cd /usr/hdp/current/kafka-broke**

2. 새로운 stream 생성하기

▶ **./kafka-topics.sh --create --zookeeper sandbox.hortonworks.com:2181 replication-factor 1 --partitions 1 --topic fred**

- Kafka는 zookeeper에 의존해 어떤 스트림이 존재하는지 추적
- --zookeeper + 서버 이름
- :2181 은 zookeeper를 실행하는 포트
- --replication-factor 1 –partitions 1 은 한 개의 서버를 의미
- -- topic fred : 스트림 토픽 이름 지정 ex) fred
- 서버 이름을 localhost로 지정해도 됨

3. Topic이 잘 만들어졌는지 확인하기

▶ **./kafka-topics.sh --list --zookeeper sandbox-hdp.hortonworks.com:2181**

Zookeeper가 있어야지 요청을 수행할 수 있고, 명령어를 실행하면 지금까지 만들어 놓은 토픽목록 확인 가능

4. 데이터 발행하기

▶ **./kafka-console-producer.sh --broker-list sandbox-hdp.hortonworks.com:6667 --topic fred**

- Kafka 실제 서버 지정을 config/server.properties 파일이 listener=PLAINTEXT 부분과 형식이 같아야지 에러가 안나고 정상적으로 처리가 가능
- ./kafka-console-producer.sh : kafka와 소통하며 데이터를 발행하는 샘플 생산자 앱
- Localhost:6667 은 kafka의 실제 서버를 지정

5. 2개의 세션을 열어서 하나의 세션은 kafka를 발행하는데 전념하고 다른 하나의 세션은 kafka 콘솔 생산자 스크립트를 사용해 해당 스트림 스레드에 발행

▶ **cd /usr/hdp/current/kafka-broker/bin**

▶ **./kafka-console-consumer.sh sandbox-hdp.hortonworks.com:6667 --zookeeper sandbox-hdp.hortonworks.com:2181 --topic fred --from-beginning**

- Zookeeper를 실행하는 포트와 데이터를 발행한 포트가 동일해야함
- --from-beginning : 토픽의 처음부터 가져오겠다고 지정 (지정하지 않으면 새로운 메시지만 가져오게 됨)
- 첫번째 세션인 생산자 창에서 fred 토픽으로 메시지를 전송한다면 두번째 세션인 소비자 창에서도 실시간으로 당연히 메시지 확인이 가능하다ㅏ.