카프카 찍먹

2023-03-14 권정인

카프카란 무엇인가?

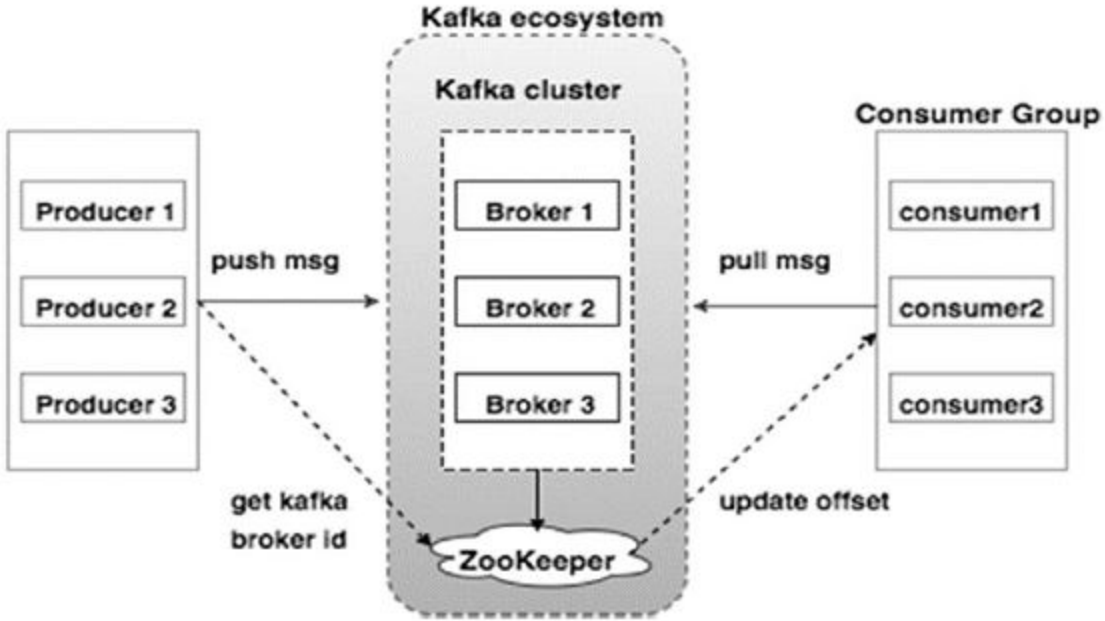
메시지 브로커 vs 이벤트 브로커의 차이

- 메시지 브로커
  - 메시지를 받아서 적절히 처리하고 나면 즉시 또는 짧은 시간 내에 삭제되는 구조
  - ex) Redis Queue, RabbitMQ, AWS SQS
- 이벤트 브로커
  - 필요한 시간(Retention)동안 이벤트 보존 가능
  - 이벤트 브로커는 메시지 브로커의 역할을 할 수 있음
  - ex) Kafka, Kinesis, AWS EventBridge

카프카

카프카의 아키텍처는 스트리밍 데이터를 처리하고 저장하기 위한 **내결함성**, **확장성**, **분산형** 플랫폼을 제공하도록 설계되어있다.

카프카 아키텍처


Kafka ecosystem
Kafka cluster
Consumer Group
Producer 1
Producer 2
Producer 3
push msg
Broker 1
Broker 2
Broker 3
pull msg
consumer1
consumer2
consumer3
get kafka
broker id
ZooKeeper
update offset

토픽

- 카프카는 메시지를 토픽으로 구성하며, 토픽은 본질적으로 카프카 브로커에 저장되는 레코드(Record) 스트림이다. 토픽은 프로듀서가 레코드를 게시하는 카테고리 또는 피드 이름으로 생각할 수 있다.
- 관심사가 같은 메시지를 모아준다는 점에서 슬랙 채널에 비유할 수 있다.

프로듀서

- 프로듀서는 카프카 토픽에 데이터를 쓰는 애플리케이션이다. 프로듀서는 센서나 로그 파일과 같이 데이터를 생성하는 모든 애플리케이션이 될 수 있다.
- 프로듀서는 하나 또는 여러 토픽에 쓸 수 있다.

브로커

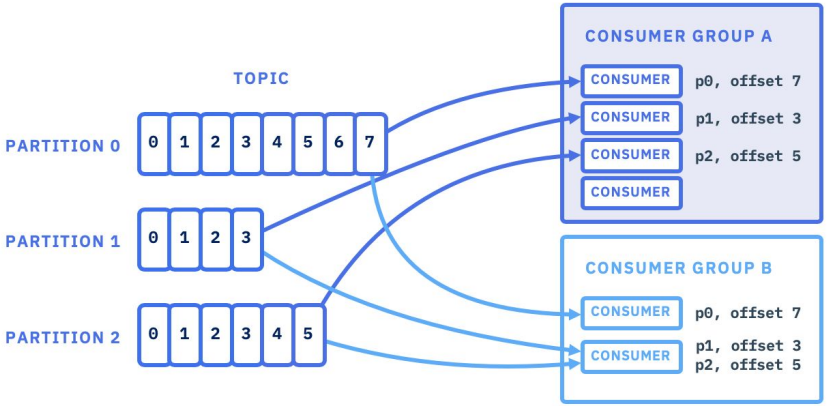
- 브로커는 데이터의 저장과 복제를 관리하는 Kafka 서버다.
- 브로커는 프로듀서로부터 메시지를 수신하여 디스크에 저장하고 컨슈머에게 제공하는 역할을 담당한다.
- Kafka 클러스터의 각 브로커는 고유 ID로 식별되며, 각 브로커는 데이터의 하위 집합을 저장한다.

컨슈머

- 컨슈머는 카프카 토픽에서 데이터를 읽는 애플리케이션이다.
- 컨슈머는 하나 이상의 토픽을 구독하고 토픽에 기록된 순서대로 데이터를 읽는다.
- 컨슈머는 컨슈머 그룹의 일부가 될 수 있으며, 이를 통해 여러 인스턴스에 걸쳐 메시지 읽기 부하를 분산할 수 있다.

컨슈머 그룹

- 컨슈머 그룹은 일련의 카프카 토픽을 소비하기 위해 함께 작업하는 컨슈머들의 집합
- 그룹 내의 각 컨슈머는 토픽의 고유한 파티션에서 읽는다. Kafka는 각 파티션이 그룹의 한 구성원만 사용하도록 하여 워크로드가 컨슈머 간에 균형을 이루도록 한다.



파티션

- 카프카 토픽은 카프카에서 병렬 처리의 단위인 파티션으로 나뉜다. 각 파티션은 프로듀서에 의해 지속적으로 추가되는 정렬된 불변의 레코드 시퀀스이다. 각 파티션은 내결함성을 위해 여러 브로커에 걸쳐 복제된다.
- 메시지를 저장하는 물리적인 파일
- 한 파티션 내에서만 순서 보장
- 프로듀서는 라운드로빈 또는 키로 파티션 선택
  - 키가 있는 경우에는 키의 hash 값으로 파티션 선택
  - 같은 키를 갖는 메시지는 같은 파티션에 저장 → 같은 키는 순서 유지

복제

- Kafka는 고가용성과 내결함성을 보장하기 위해 데이터의 기본 복제를 제공합니다. 각 파티션은 여러 브로커에 걸쳐 복제되므로 한 브로커가 다운되면 다른 브로커가 이를 대신할 수 있습니다. 복제 계수에 따라 클러스터에 유지되는 각 파티션의 복사본 수가 결정됩니다.
- 리더(leader) - 팔로워(follower)

로컬에서 실행해보기

docker-compose.yml (zookeeper, kafka 생성)

docker-compose up

토픽 만들기

```
docker-compose exec kafka kafka-topics --create --topic
euljiro-kafka-jungin --bootstrap-server kafka:9092 --replication-factor 1
--partitions 1
```

토픽 확인

docker-compose exec kafka kafka-topics --describe --topic euljiro-kafka-jungin --bootstrap-server kafka:9092

메시지 발행해보기

https://github.com/yyna/kafka-euljiro

컨슈머

docker-compose exec kafka bash

kafka-console-consumer --topic euljiro-kafka-jungin --bootstrap-server kafka:9092

기타

Streams

카프카 스트림이라고도 하는 카프카 스트리밍은 개발자가 카프카 플랫폼 위에서 실시간 스트림 처리 애플리케이션을 구축할 수 있도록 해주는 라이브러리이다. 실시간으로 데이터를 처리하기 위한 가볍고 사용하기 쉬우며 확장성이 뛰어난 프레임워크를 제공한다.

use cases)

1. 실시간 분석
2. 사기 탐지
3. IoT 데이터 처리
4. 소셜 미디어 정서 분석
5. 클릭스트림 처리

Connect

데이터베이스, 메시징 시스템, 데이터 웨어하우스 등 외부 시스템과 Kafka를 통합할 수 있는 프레임워크를 제공하는 오픈 소스 도구이다. 이 도구를 사용하면 사용자 정의 코드를 작성하거나 복잡한 통합을 유지 관리할 필요 없이 Kafka와 다른 시스템 간에 데이터를 쉽고 안정적으로 스트리밍할 수 있다.

Monitoring

1. Kafka Manager: Kafka Manager는 아파치 카프카 클러스터를 관리하고 모니터링하기 위한 웹 기반 인터페이스를 제공하는 오픈 소스 도구이다. 이 도구는 원래 Yahoo! 에서 개발했으며 현재는 Contributor 커뮤니티에서 유지 관리하고 있다.
2. Prometheus: 카프카 클러스터를 모니터링하는 데 자주 사용되는 오픈 소스 모니터링 및 경보 시스템이다. Prometheus는 주기적으로 Kafka와 Kafka 에코시스템의 다른 구성 요소에서 메트릭을 스크랩하고 수집된 메트릭을 시계열 데이터베이스에 저장하는 방식으로 작동한다. 그런 다음 수집된 메트릭은 Grafana와 같은 도구를 사용하여 쿼리하고 시각화할 수 있다.
3. Datadog