

# 메시지 송수신 기본

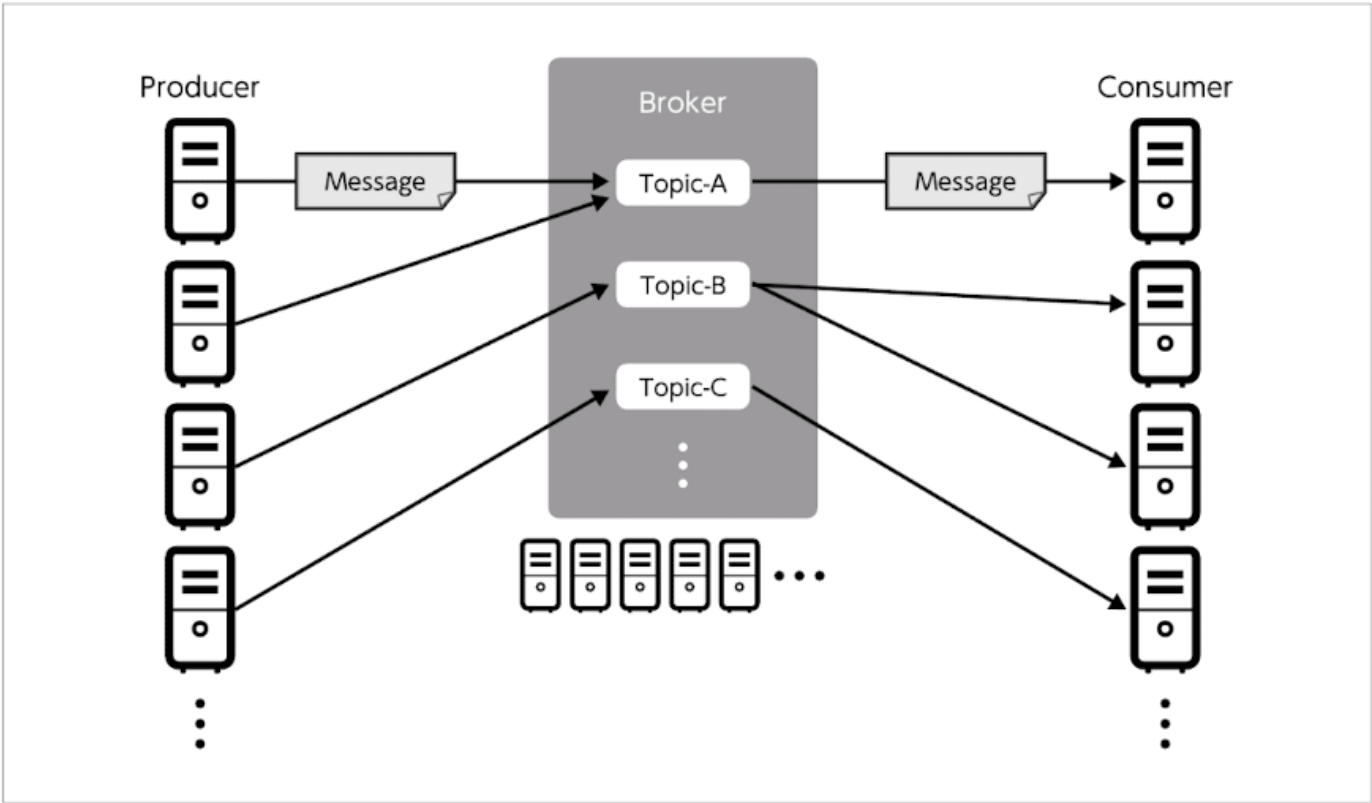


그림 2-1 카프카 개요도

- 카프카의 주요 구성 요소 5가지
  - 브로커
    - 데이터를 수신, 전달하는 서비스
  - 메시지
    - 카프카에서 다루는 데이터의 최소 단위, 카프카가 중계하는 로그의 한 줄 한줄과 센서 데이터 등이 이에 해당
    - 메시지는 key와 Value를 갖게 되며 메시지 전송할 때 파티셔닝에 이용
  - 프로듀서
    - 데이터의 생산자이며 브로커에 메시지를 보내는 애플리케이션
  - 컨슈머
    - 브로커에서 메시지를 취득하는 애플리케이션
  - 토픽
    - 메시지를 종류(토픽)별로 관리하는 스토리지
    - 브로커에 배치되어 관리
    - 프로듀서와 컨슈머는 특정 토픽을 지정하여 메시지를 송수신함으로써 단일 카프카 클러스터에서 여러 종류의 메시지를 중계

## 시스템 구성

# 브로커

- 브로커는 하나의 서버 당 하나의 데몬 프로세스로 동작하여 메시지 수신/전달 요청을 받아들임
- 여러 대의 클러스터로 구성할 수 있으며 브로커를 추가함으로써 수신/전달의 처리량 향상(스케일 아웃)이 가능
- 브로커에서 받은 데이터는 모두 디스크로 내보내기가 이루어져 디스크의 총 용량에 따라 장기간 데이터를 보존할 수 있음

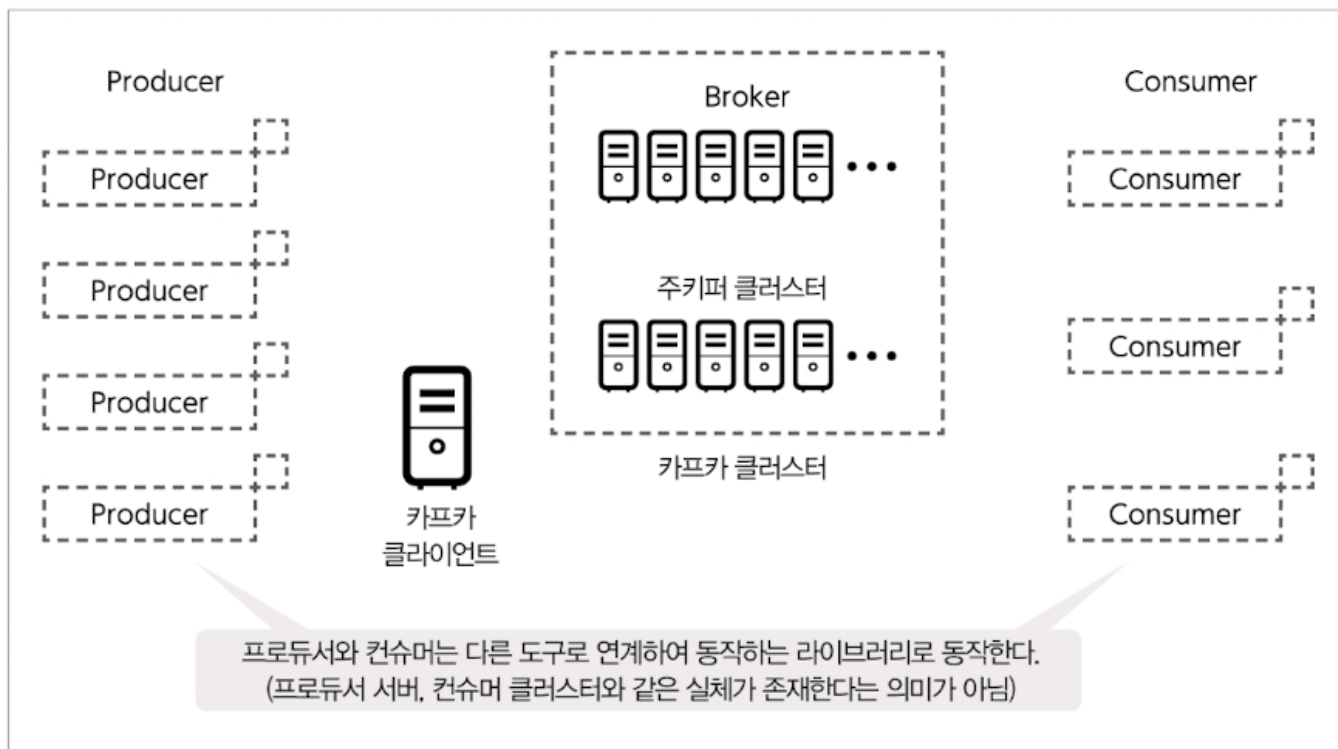


그림 2-2 카프카 시스템 구성

## 프로듀서 API / 컨슈머 API

- 프로듀서/컨슈머를 구현하는 기능은 브로커로 데이터를 보내고 브로커에서 데이터를 받기 위한 라이브러리로 제공
- 프로듀서는 프로듀스 API를 이용하여 브로커에 데이터를 송신하기 위해 구현된 애플리케이션
- 실제 사례로는 각종 로그 전송 및 미들웨어와 연동하여 동작하기 때문에 프로듀서 API를 내포한 도구, 미들웨어를 통해 이용하는 형태 등 다양하다
- 사용할 수 있는 서드 파티의 플러그인
  - Apache Log4j
  - Apache Flume
  - Fluentd
  - Logstash

## 컨슈머

- 컨슈머 API를 이용해 브로커에서 메시지를 취득하도록 구현된 애플리케이션
- 브로커는 메시지를 영속화하기 위해 브로커에 도달하는 즉시 컨슈머에서 취득해야 한다는 제약이 없기 때문에, 메시지를 디스크에 보관하며 보관되어 있는 동안 언제든지 메시지 취득이 가능

- 일정 기간 데이터를 축적한 스토리지에서의 데이터 추출 및 실시간 처리를 위한 애플리케이션의 데이터 입력등으로 이용
- 카프카 연계를 위한 컨슈머 기능을 갖춘 기존 제품들 리스트
  - Apache Spark
  - Apache Samza
  - Apache Flink
  - Apache Flume
  - Fluentd
  - Logstash

## 주키퍼

- 카프카의 브로커에 있어 분산 처리를 위한 관리 도구로 아파치 주키퍼가 필요
- 주키퍼는 하둡 등 병렬 분산 처리용 프로그램으로 설정 관리, 이름 관리, 동기화를 위한 잠금 관리를 위한 구조로 자주 사용
- 카프카에 있어서는 분산 메시징의 메타데이터를 관리하기 위한 구성 요소로 기능
- 주키퍼 클러스터의 구조상 3, 5처럼 홀수로 구성하는 것이 일반적

## 카프카 클라이언트

- 토픽 작성 등 카프카의 동작 및 운영 상에 필요한 조작을 실행하는 서버
- 메시지의 송수신을 처리하는 서버가 아님

## 카프카 클러스터

- 카프카는 여러 대의 브로커 서버, 주피커 서버로 이루어진 클러스터링의 메시지 중계 기능과 메시지 송수신을 위한 라이브러리 그룹으로 구성
- 주키퍼에 의해 구성된 클러스터 시스템을 카프카 클러스터라고 정의

## 카프카 구축

표 3-1 3대의 서버로 카프카 클러스터를 구축할 때의 동작 환경

호스트명	역할	설명
kafka-broker01	브로커	서버 3대에서 카프카 클러스터를 구축한다.
kafka-broker02	브로커	
kafka-broker03	브로커	
producer-client	프로듀서	카프카에 메시지를 송신한다.
consumer-client	컨슈머	카프카에서 메시지를 수신한다.
kafka-client	카프카 클라이언트	카프카 클러스터의 상태 관리 및 운영을 위한 각종 조작을 처리한다.

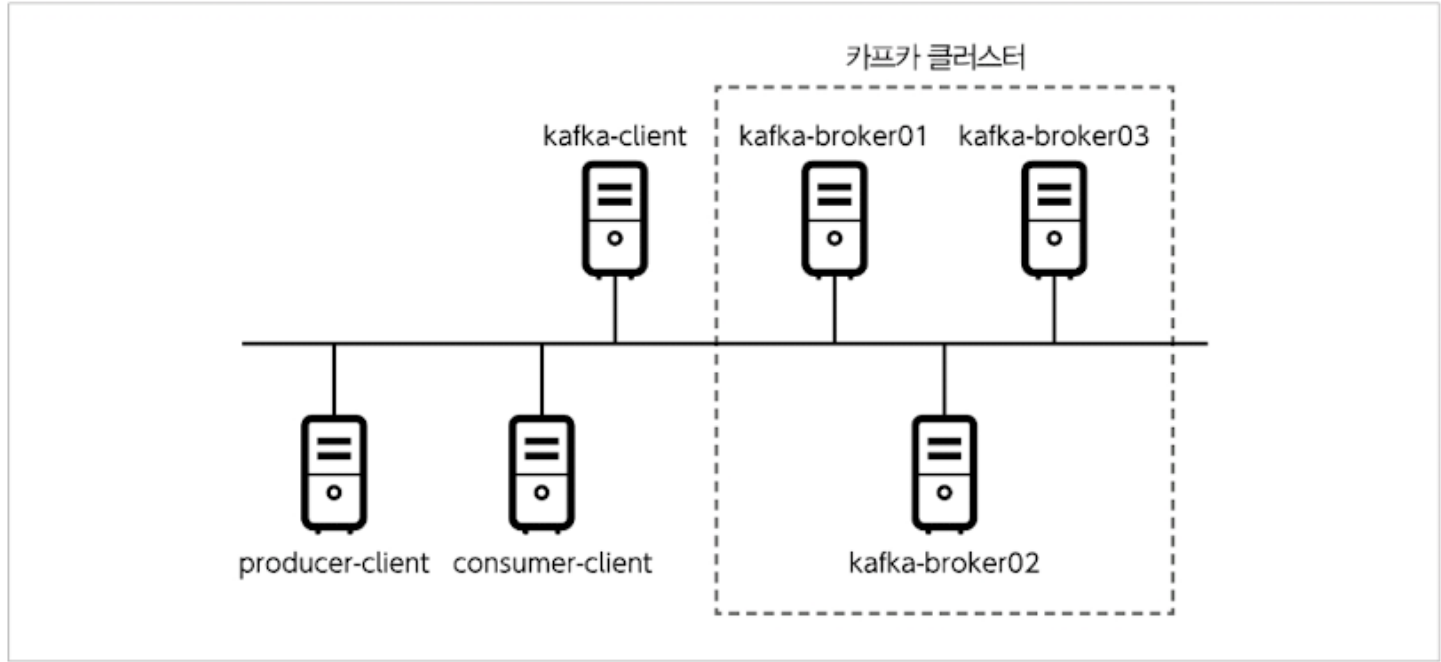


그림 3-1 3대의 서버에서 카프카 클러스터 구축하기