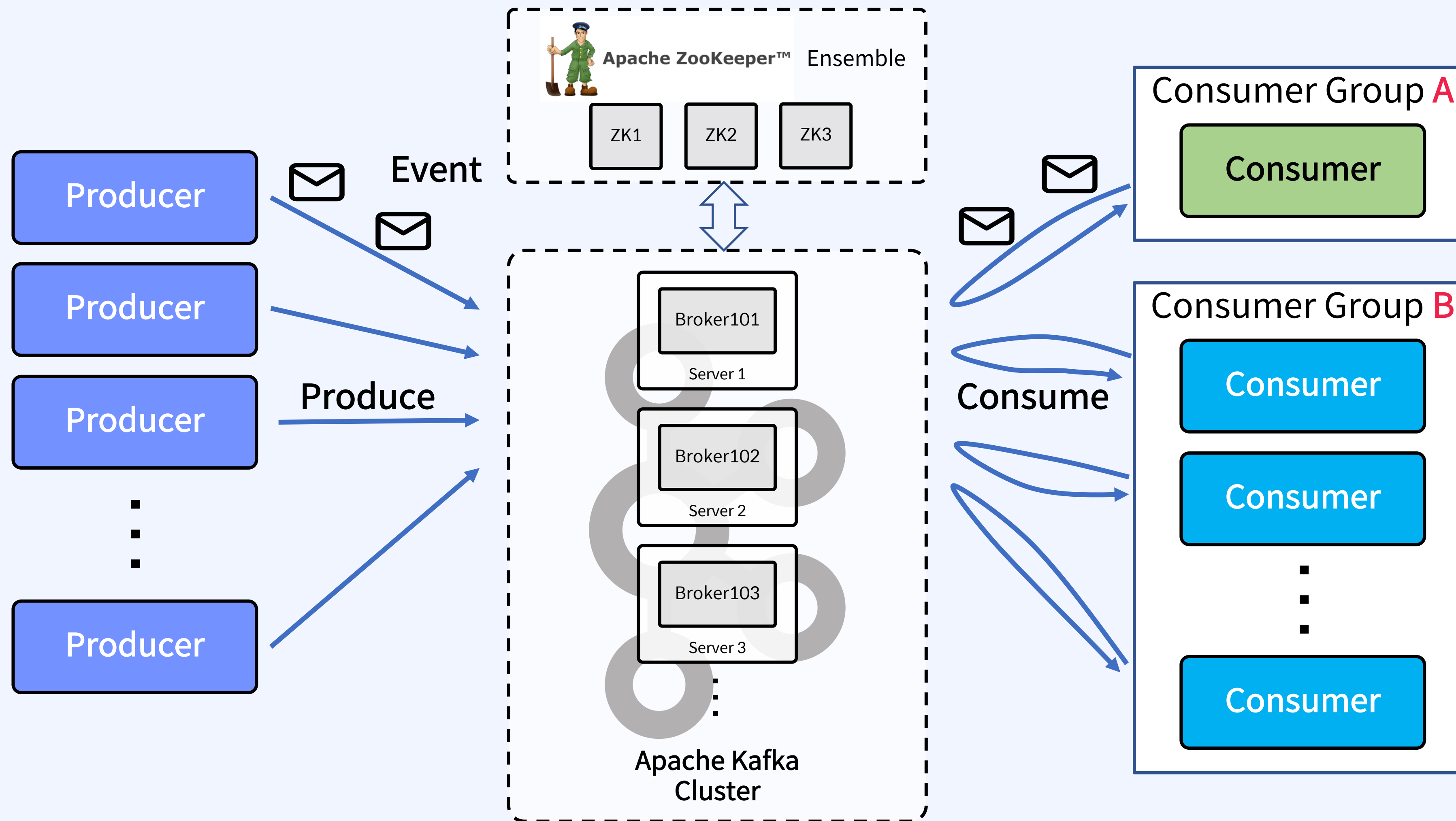


# Apache Kafka 기본 개념 및 이해

3 Broker, Zookeeper

## Apache Kafka 주요 요소 Broker, Zookeeper

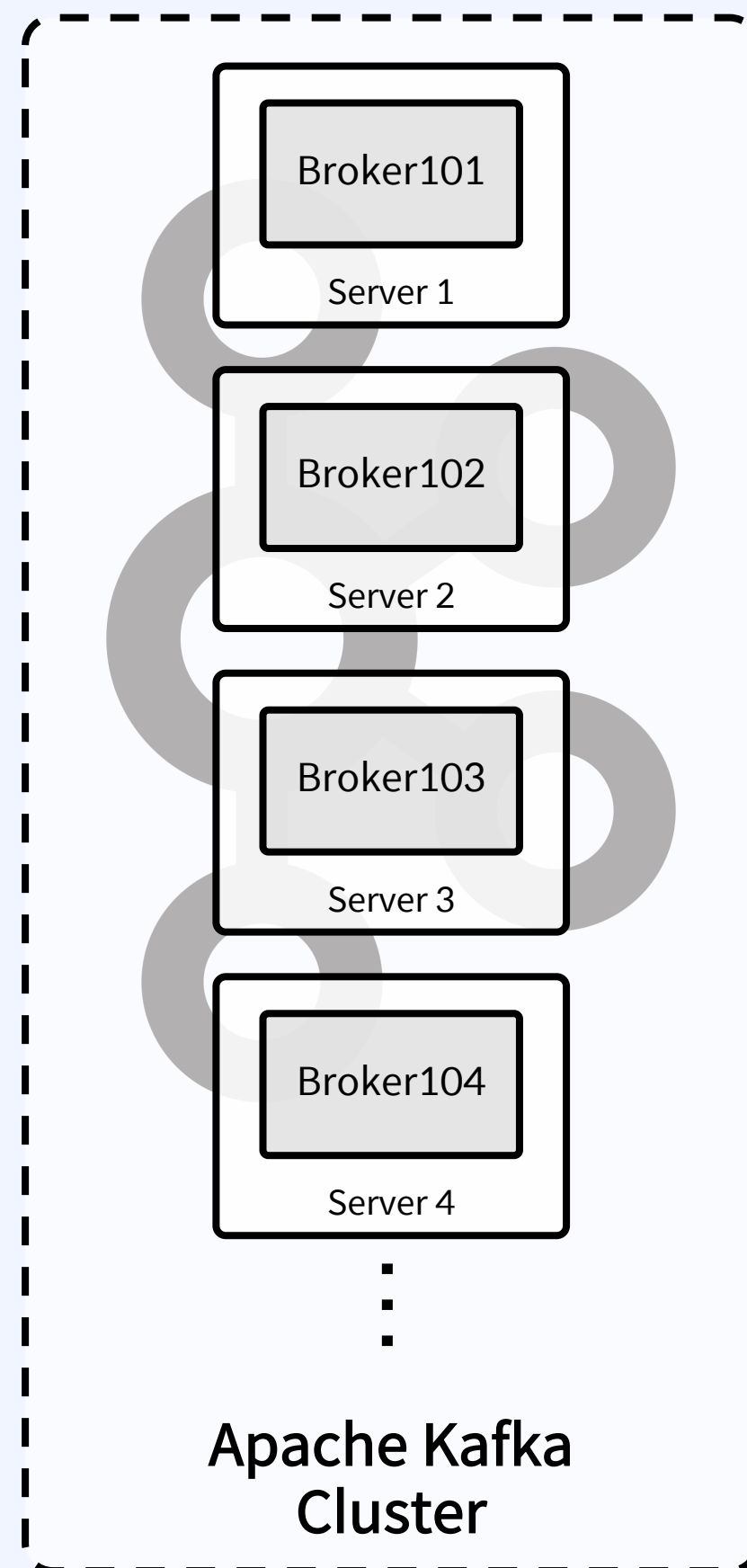
### 3. Broker, Zookeeper



## Kafka Broker

### Topic과 Partition을 유지 및 관리

Kafka Broker는 Partition에 대한 Read 및 Write를 관리하는 소프트웨어

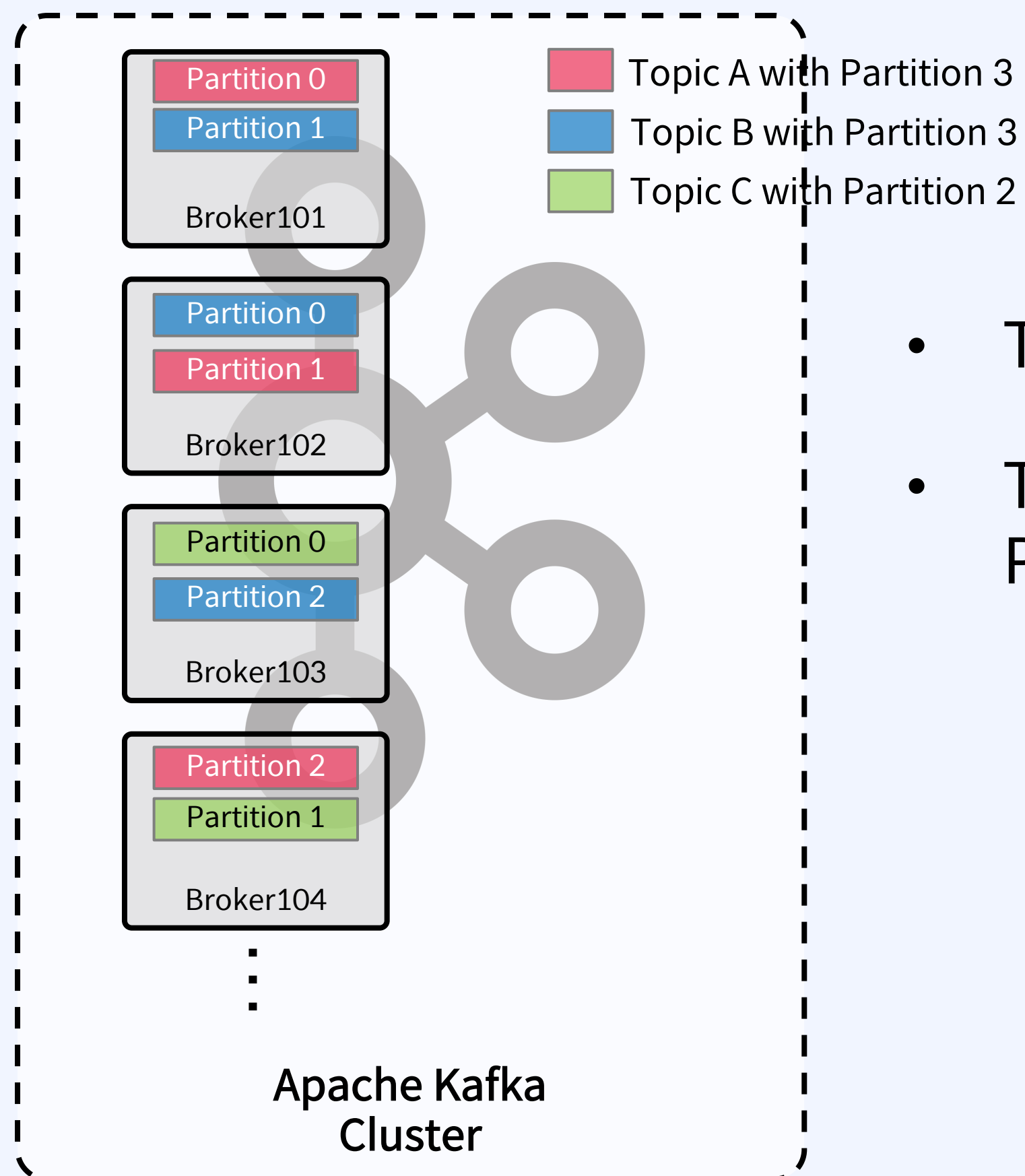


- **Kafka Server**라고 부르기도 함
- Topic 내의 Partition 들을 분산, 유지 및 관리
- 각각의 Broker들은 ID로 식별됨 (**단, ID는 숫자**)
- Topic의 일부 Partition들을 포함  
→ Topic 데이터의 일부분(Partition)을 갖을 뿐 데이터 전체를 갖고 있지 않음
- **Kafka Cluster**: 여러 개의 Broker들로 구성됨
- Client는 특정 Broker에 연결하면 전체 클러스터에 연결됨
- **최소 3대 이상**의 Broker를 하나의 Cluster로 구성해야 함  
→ **4대 이상을 권장함**

## Kafka Broker ID 와 Partition ID 의 관계 아무런 관계도 없음

### 3. Broker, Zookeeper

Broker ID와 Partition ID 간에는 아무런 관계가 없음  
➔ 어느 순서에나 있을 수 있음

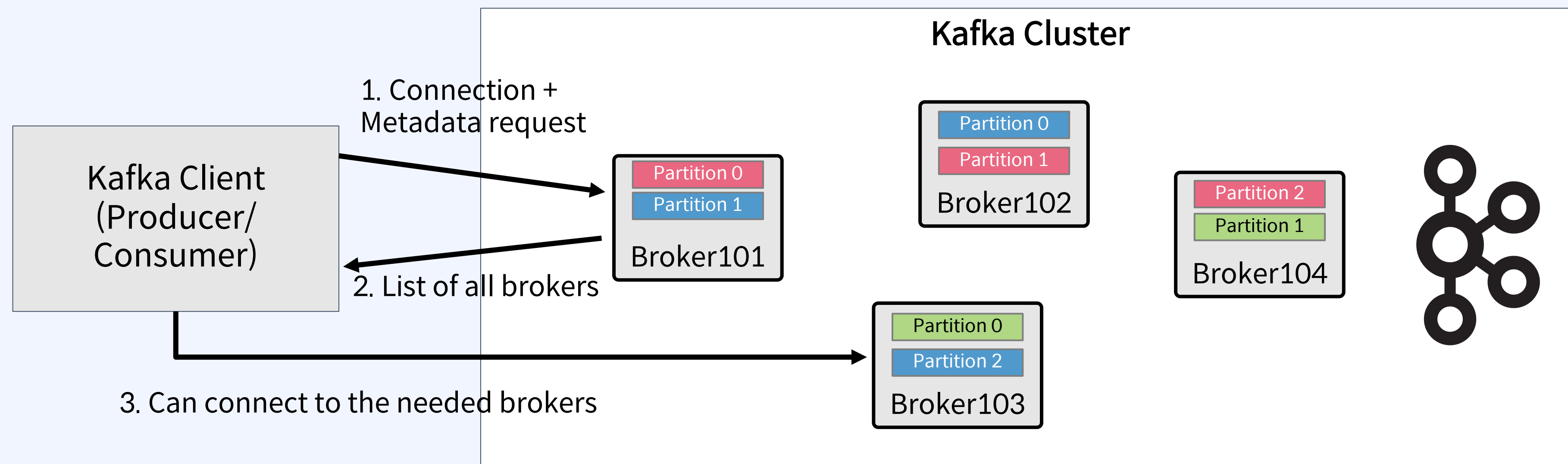


- Topic을 구성하는 Partition들은 여러 Broker 상에 분산됨
- Topic 생성시 Kafka가 자동으로 Topic을 구성하는 전체 Partition들을 모든 Broker에게 할당해주고 분배해 줌

## Bootstrap Servers Broker Servers를 의미

모든 Kafka Broker는 Bootstrap(부트스트랩) 서버라고 부름

- 하나의 Broker에만 연결하면 Cluster 전체에 연결됨  
→ 하지만, 특정 Broker 장애를 대비하여, 전체 Broker List(IP, port)를 파라미터로 입력 권장
- 각각의 Broker는 모든 Broker, Topic, Partition에 대해 알고 있음(Metadata)

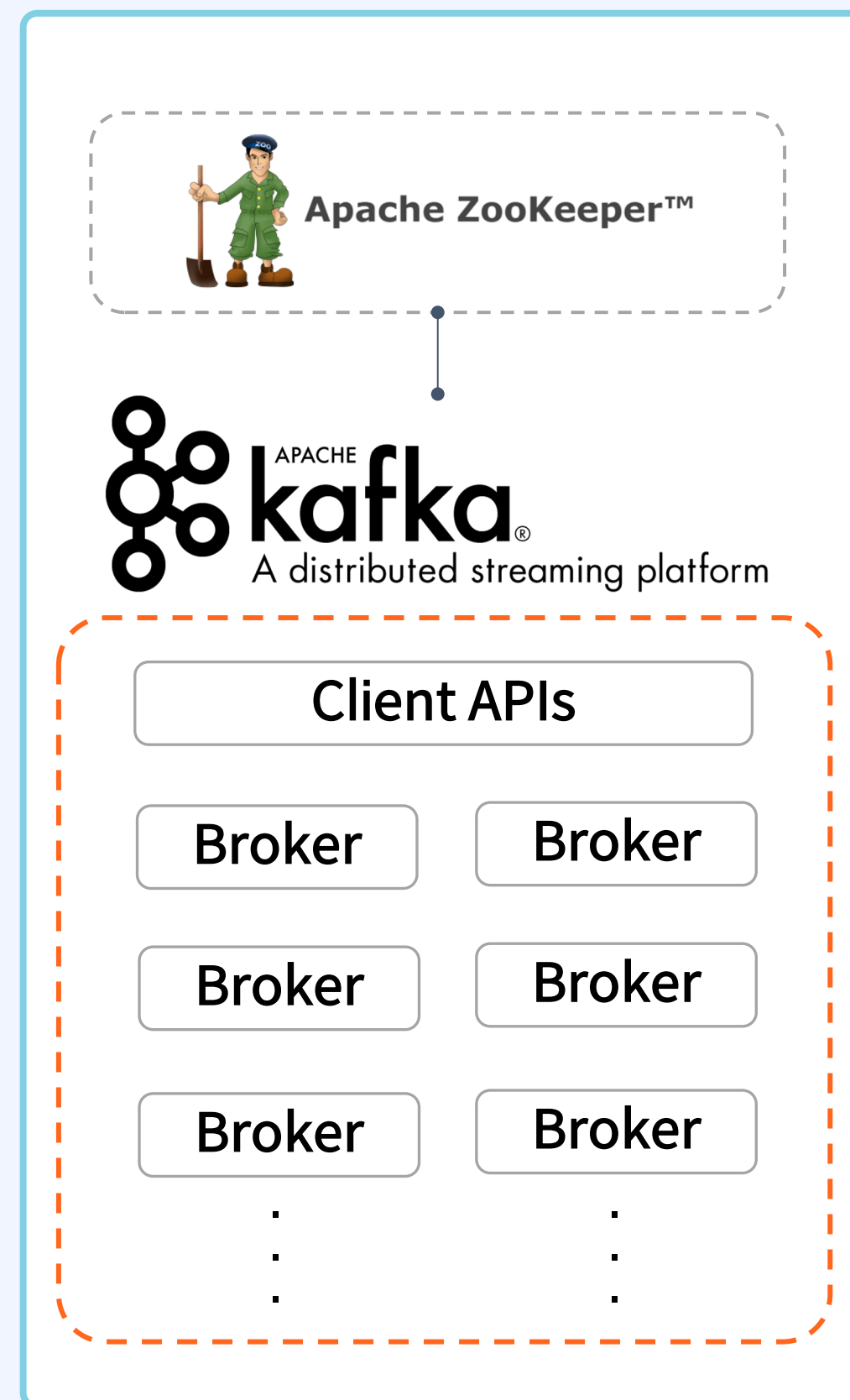


## Zookeeper Broker를 관리

3.

Broker, Zookeeper

Zookeeper는 Broker를 관리 (Broker 들의 목록/설정을 관리)하는 소프트웨어



- Zookeeper는 변경사항에 대해 Kafka에 알림  
→ Topic 생성/제거, Broker 추가/제거 등
- Zookeeper 없이는 Kafka가 작동할 수 없음  
→ KIP<sup>1)</sup>-500 을 통해서 Zookeeper 제거가 진행중  
→ 2022년에 Zookeeper를 제거한 정식 버전 출시 예정중
- Zookeeper는 홀수의 서버로 작동하게 설계되어 있음  
(최소 3, 권장 5)
- Zookeeper에는 Leader(writes)가 있고 나머지 서버는 Follower(reads)

1) KIP : Kafka Improvement Proposal

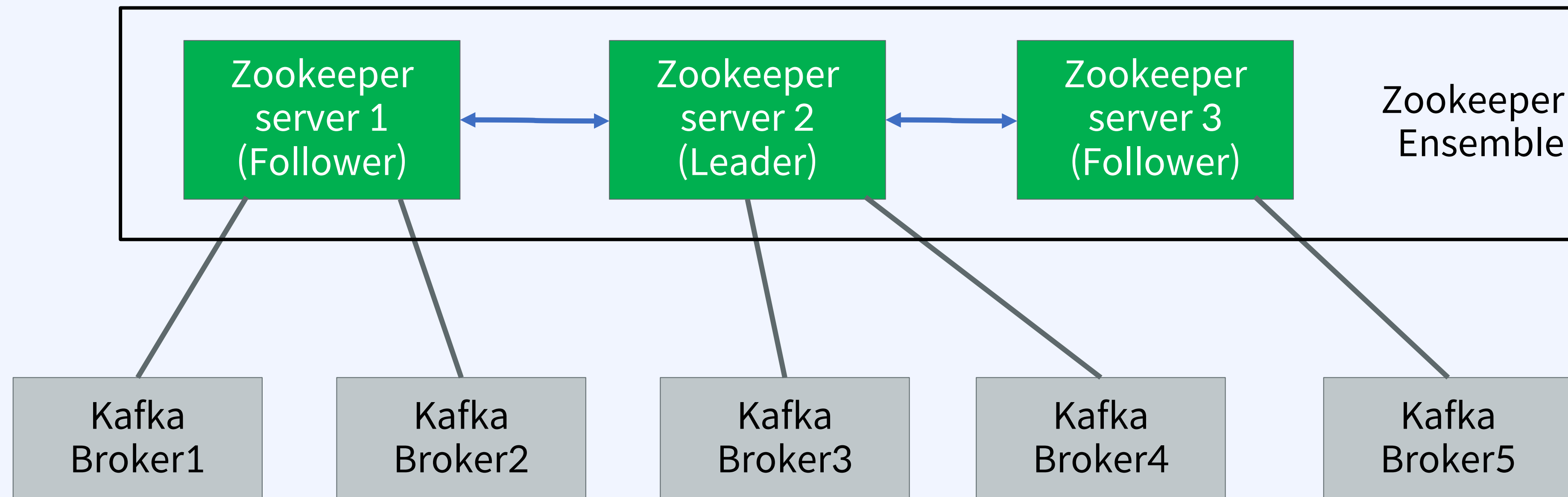
## Zookeeper 아키텍처

### Leader/Follower 기반 Master/Slave 아키텍처

Zookeeper는 분산형 Configuration 정보 유지, 분산 동기화 서비스를 제공하고 대용량 분산 시스템을 위한 네이밍 레지스트리를 제공하는 소프트웨어

분산 작업을 제어하기 위한 Tree 형태의 데이터 저장소

➔ Zookeeper를 사용하여 멀티 Kafka Broker들 간의 정보(변경 사항 포함) 공유, 동기화 등을 수행





## Zookeeper Failover Quorum 알고리즘 기반

3.

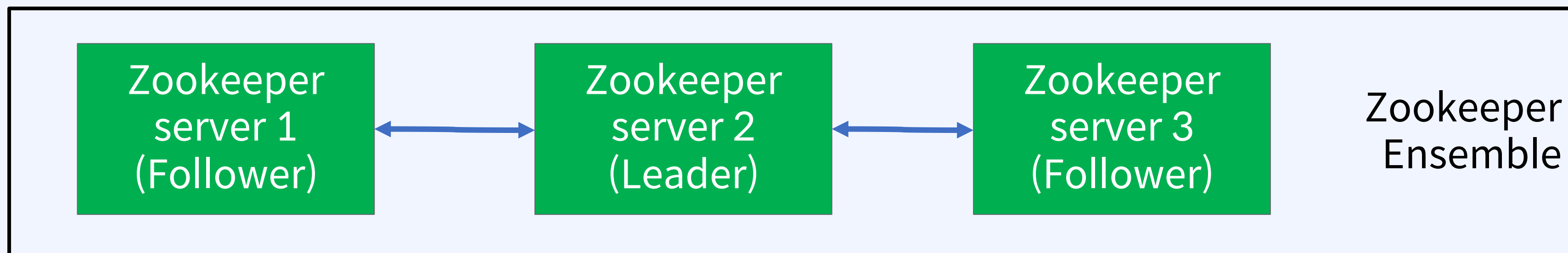
Broker, Zookeeper

Ensemble은 Zookeeper 서버의 클러스터

Quorum(쿼럼)은 “정족수”이며, 합의체가 의사를 진행시키거나 의결을 하는데 필요한 최소한의 인원수를 뜻함

분산 코디네이션 환경에서 예상치 못한 장애가 발생해도 분산 시스템의 일관성을 유지시키기 위해서 사용

Ensemble이 3대로 구성되었다면 Quorum은 2, 즉 Zookeeper 1대가 장애가 발생하더라도 정상 동작  
Ensemble이 5대로 구성되었다면 Quorum은 3, 즉 Zookeeper 2대가 장애가 발생하더라도 정상 동작





## Summary

Zookeeper와 Broker는 서로 다르다

3.

Broker, Zookeeper

- Broker는 Partition에 대한 Read 및 Write를 관리하는 소프트웨어
- Broker는 Topic 내의 Partition 들을 분산, 유지 및 관리
- 최소 3대 이상의 Broker를 하나의 Cluster로 구성해야 함  
➔ 4대 이상을 권장함
- Zookeeper는 Broker를 관리 (Broker 들의 목록/설정을 관리)하는 소프트웨어
- Zookeeper는 홀수의 서버로 작동하게 설계되어 있음 (최소 3, 권장 5)