

# Apache Kafka 구성 및 관리

## 4 Log Retention, Cleanup Policy

## Log 파일 관리 Log Cleanup Policy

### 4.

Log Retention,  
Cleanup Policy

Log는 Consume되어도 지우지 않음

- 많은 Consumer들이 서로 다른 시간에 Log 데이터를 Consume 할 수 있기 때문에
- Broker 혹은 Topic 단위로 Cleanup 정책을 설정함
- `log.cleanup.policy` 파라미터
  - `delete`
  - `compact`
  - `delete,compact`
- 현재 Active Segment의 Log는 cleanup 대상이 아님

## Delete Cleanup Policy

### Log Segment 삭제하는 정책

# 4.

Log Retention,  
Cleanup Policy

### Log Cleaner Thread가 Segment File을 삭제

- `log.cleanup.policy` 파라미터
  - `delete`
- `log.retention.ms` : log 보관 주기 (기본값 : 7 일)
- `log.retention.check.interval.ms` : log segment를 체크하는 주기 (기본값 : 5 분)
- segment 파일에 저장된 가장 최신의 메시지가 `log.retention.ms` 보다 오래된 segment 를 삭제

Topic의 메시지를 모두 삭제하는 방법  
retention.ms 를 활용(운영환경에서는 권장하지 않음)

1. Producer와 Consumer를 모두 shutdown
2. 명령어를 사용하여 해당 Topic의 retention.ms를 0으로 셋팅

```
kafka-config.sh --zookeeper ${zookeeper ip address} --alter --entity-name
topics --entity-name hkim_topic --add-config retention.ms=0
```

3. Cleanup Thread 가 동작할 동안 대기 (기본값 5분 마다 동작)
4. 메시지 삭제 확인 후, 원래 설정으로 원복

```
kafka-config.sh --zookeeper ${zookeeper ip address} --alter --entity-name
topics --entity-name hkim_topic --delete-config retention.ms
```

주의) Log File 자체를 절대로 직접 삭제하면 안 됨

## Compact Cleanup Policy

주어진 Key의 최신 Value만 유지하는 정책

4.

Log Retention,  
Cleanup Policy

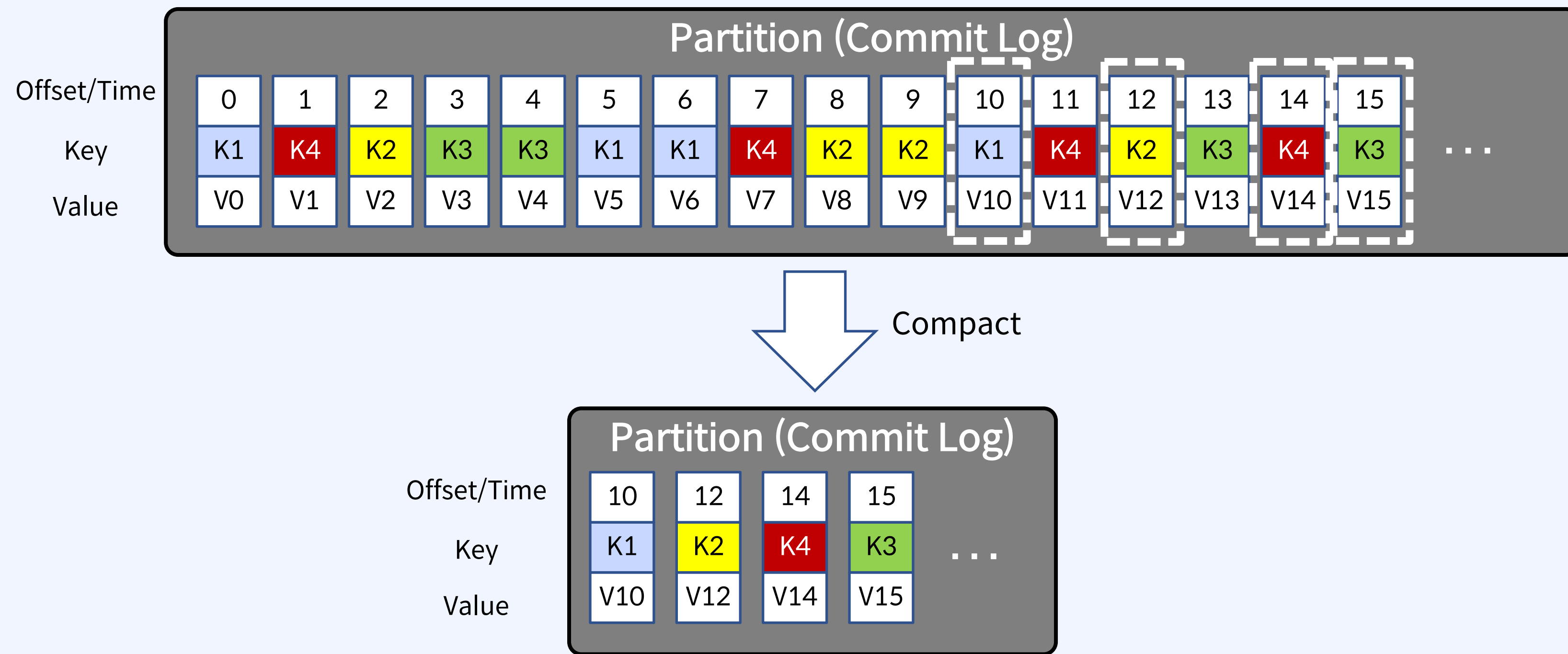
### 각 Key의 최신 Value만을 유지

- Key가 있는 메시지를 사용하면 Custom Partitioner를 사용하지 않는 한, 특정 Key를 가지는 모든 메시지는 동일한 Partition으로 send 됨
- Compact(압축) 정책은 Partition별로 특정 Key의 최신 Value만 유지하며 압축
- 동일한 Key를 가진 메시지가 다른 Partition에 있는 경우, 동일한 Key를 가진 여러 메시지가 여전히 있을 수 있음  
→ 중복 제거용 기술이 아님

## Log Compaction

시스템 오류 후 상태를 복원하는 데 유용

Key가 있는 메시지에 대해서만 작동

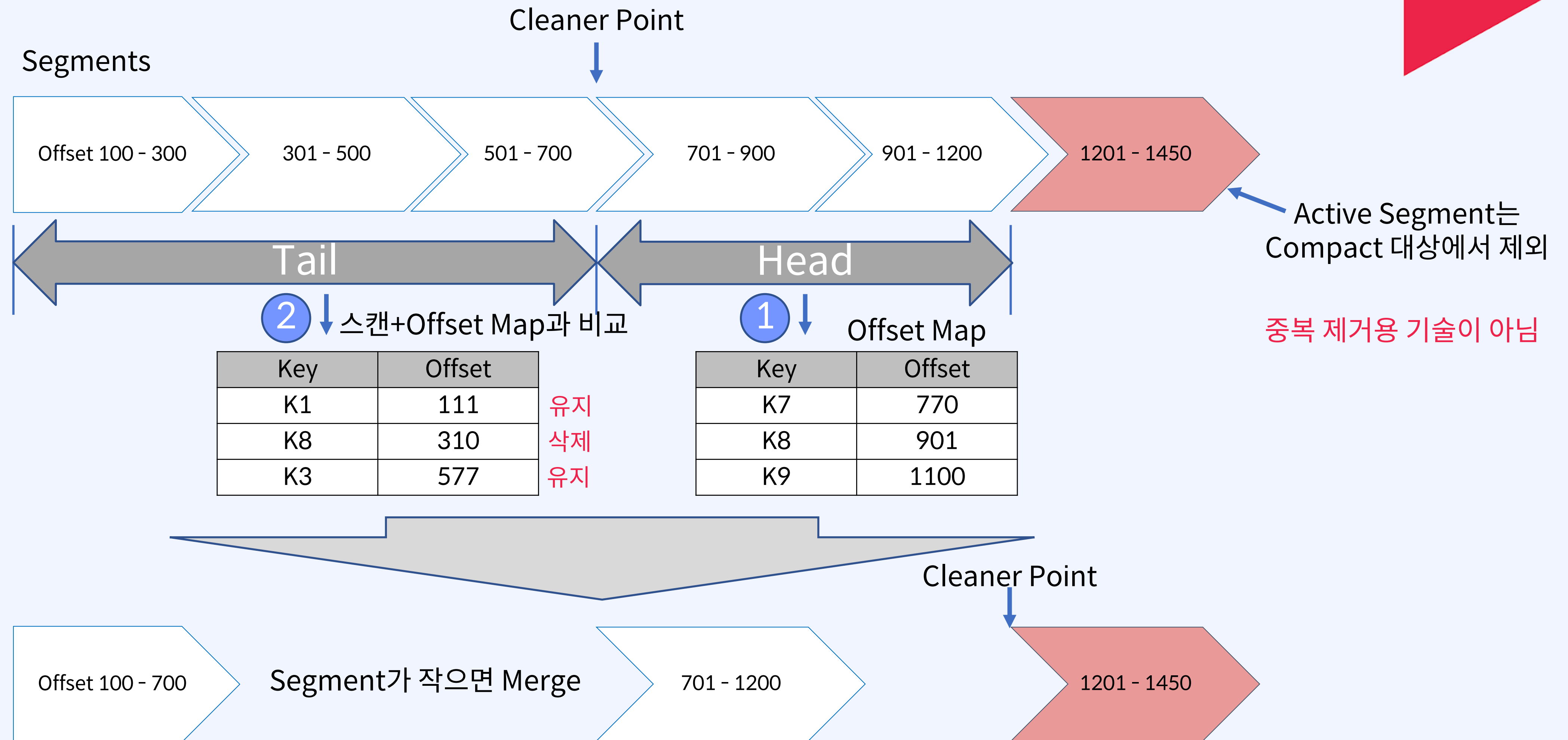


압축이 없으면 Consumer는 항상 전체 로그를 읽고 결국 각 Key에 대한 가장 최신 상태에 도달할 수 있지만, 로그 압축을 사용하면 오래된 데이터를 읽지 않기 때문에 Consumer가 최종 상태에 더 빨리 도달할 수 있음(예, \_\_consumer\_offsets Topic)

## Log Compaction 동작 원리

4.

Log Retention,  
Cleanup Policy



## Log Compaction 설정

### Compaction 성능 튜닝 관련 옵션

# 4.

Log Retention,  
Cleanup Policy

- **log.cleaner.min.cleanable.ratio** (기본값 : 0.5)  
Head 영역 데이터가 Tail 영역보다 크면(기본값 50%), Cleaner 시작
- **log.cleaner.io.max.bytes.per.second** (기본값 : 무제한)  
Log Cleaner의 Read/Write 의 처리량을 제한하여 시스템 리소스 보호 가능
- 동일한 Key를 갖는 메시지가 매우 많은 경우, 더 빠른 정리를 위해서 아래의 파라미터를 증가시켜야 함
  - **log.cleaner.threads** (기본값 : 1)
  - **log.cleaner.dedupe.buffer.size** (기본값 : 134,217,728)



## Tombstone Message

Log Compaction시 특정 Key 데이터 삭제

### Tombstone Message



- Compaction 사용시에 Key로 **K**를 가지는 메시지를 지우려면, **동일한 Key(K)에 null value**를 가지는 메시지를 Topic으로 보내면 됨
- Consumer는 해당 메시지가 지워지기 전에(기본 1 day), 해당 메시지를 consume할 수 있음
- 메시지를 지우기 전 보관 기간(기본 1 day)은 **log.cleaner.delete.retention.ms** 로 조정

## Summary

### Log Cleanup Policy

4.

Log Retention,  
Cleanup Policy

- Log Cleanup Policy : Delete, Compact, Delete&Compact
- Tombstone message