

# 7장. 레디스 데이터 백업 방법

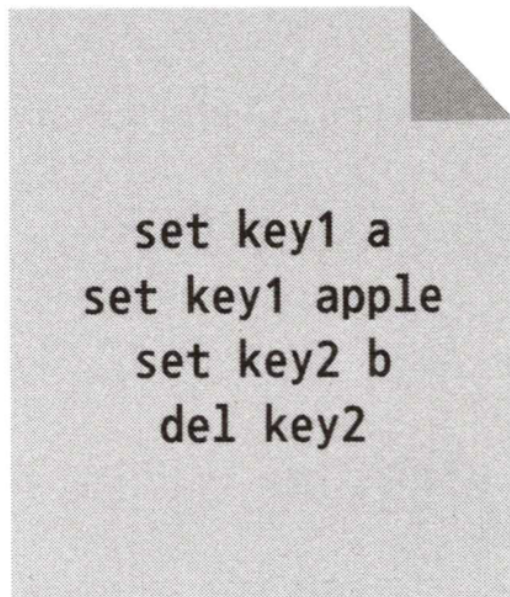
## TL;DR

- RDB와 AOF 두 가지 백업 방식을 지원한다.
- 레디스 복구 시점은 재시작될 때뿐이며, AOF or RDB 존재하면 파일 로드한다.

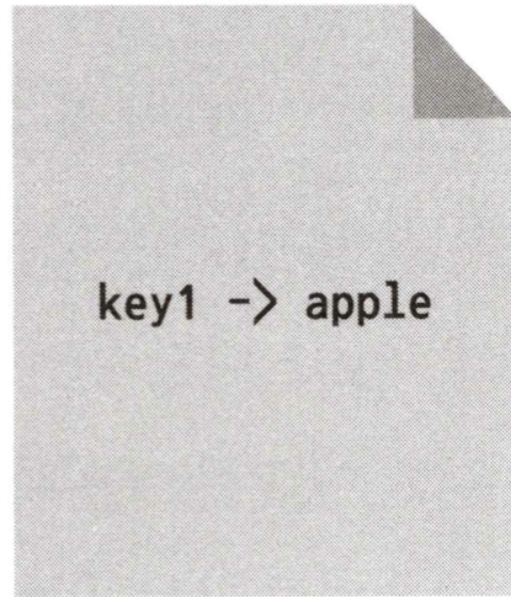
## 데이터 영구 저장

- 레디스 모든 데이터는 메모리에서 관리된다.
  - 장애로 인스턴스가 재시작되면 데이터가 손실될 수 있다.
  - 가용성을 위한 복제 구조이지만 백업과 복제는 목적부터 다르다는 것을 알아야한다.(복제는 가용성, 백업은 장애 상황에서 데이터 복구)
- 의도하지 않게 데이터를 삭제하는 커맨드가 실행되는 경우는 데이터를 안전하게 유지할 수 없음

## AOF, RDB



**AOF**  
Append Only File

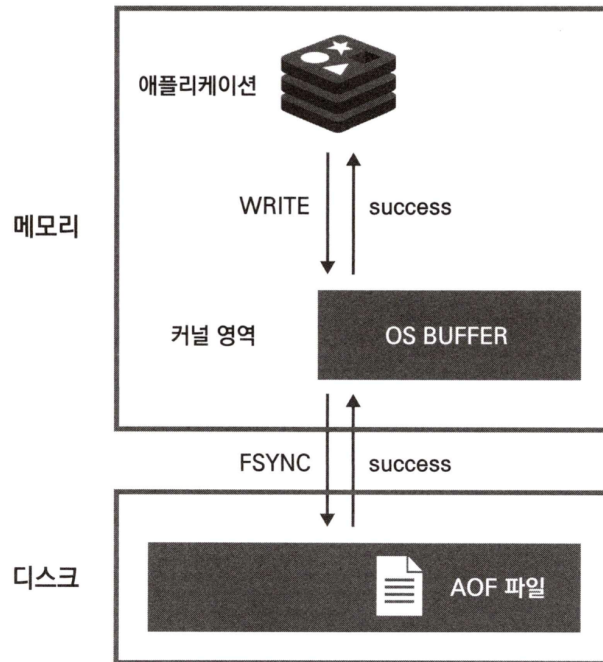


**RDB**  
스냅샷

## Append Only File

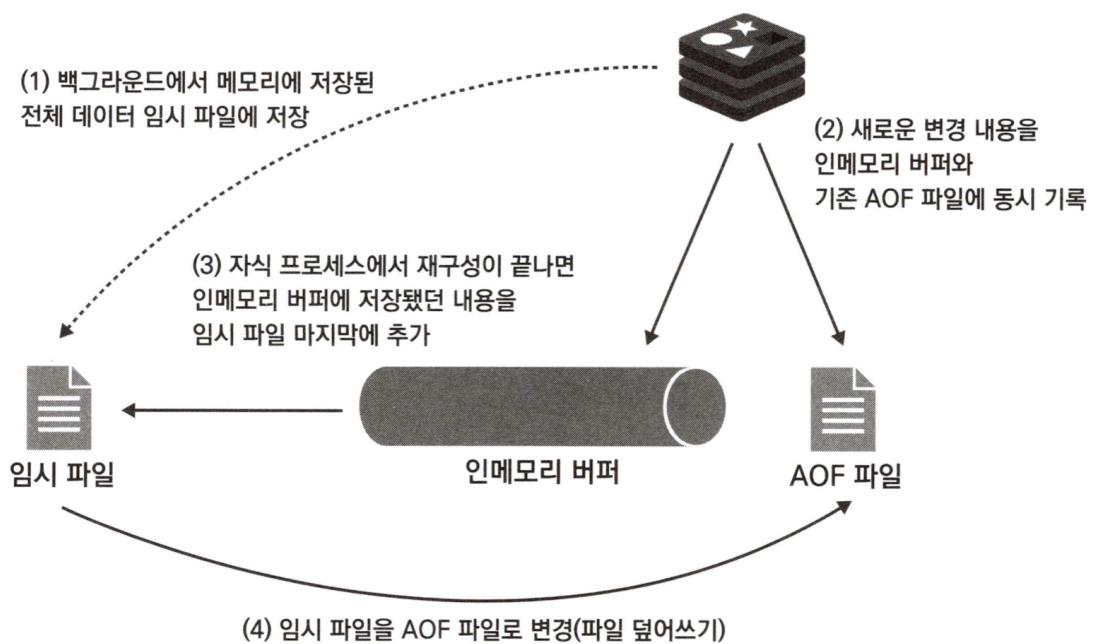
- 처리한 모든 쓰기 작업을 차례대로 기록
- 복원 시에는 파일을 다시 읽어가며 데이터 세트 재구성
- RESP 형태로 저장
- 파일 크기가 크고 주기적으로 압축 필요하지만 원하는 시점으로 복구 가능
- 커맨드를 삭제하고 재시작하면 데이터를 바로 복구할 수 있다.
- 메모리상의 데이터가 변경되는 커맨드만 기록한다.

## AOF 안전성

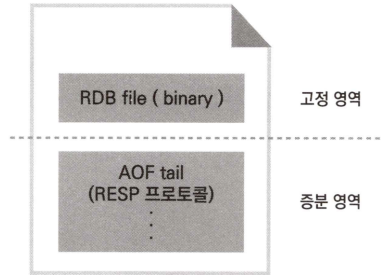


## 버전 7 이전과 이후 AOF 재구성

7 이전



이전



- 하나의 파일로 관리됨

7 이후

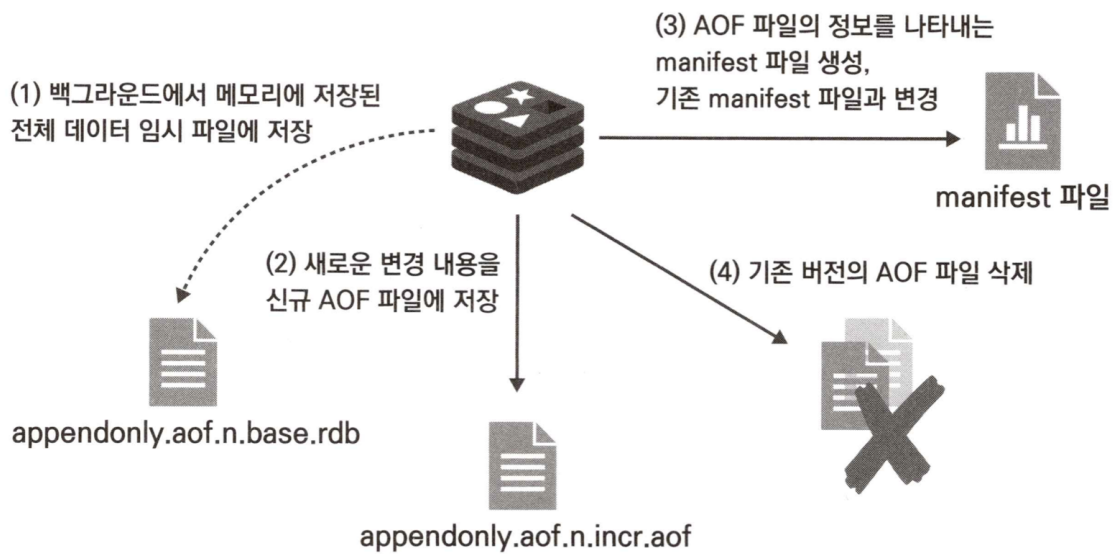
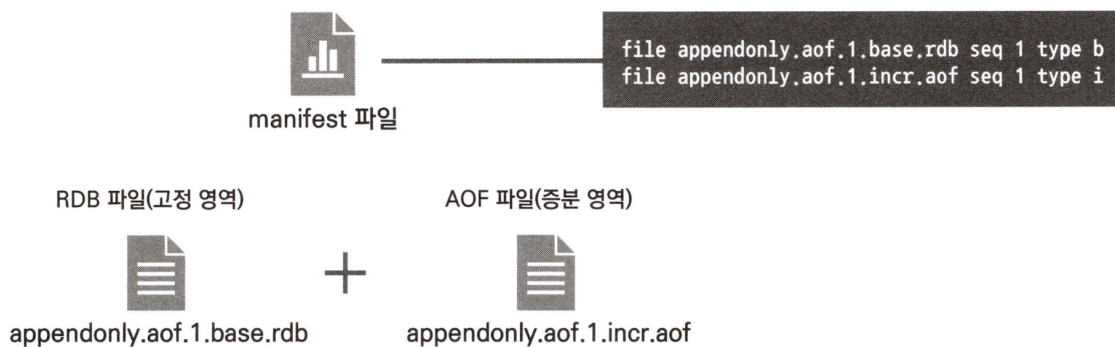


그림 7-6 버전 7 이후에서 AOF 재구성이 실행되는 과정

이후



- 여러개의 파일로 관리됨

- 옵션
- AppendFsync no: Write 시스템 콜 호출해 커널 영역에 쌓이는데 확인해 쓰기 성능이 가장 빠르다
- AppendFsync always: AOF 항상 Write, FSYNC 시스템콜 함께 호출한다. 기다리기 때문에 성능 가장 느리다
- AppendFsync everysec: 1초에 한 번씩 FSync 콜 호출, 쓰기 성능은 no 옵션 사용했을 때와 거의 비슷

## Redis DataBase

- 일정 시점에 메모리에 저장된 데이터 전체를 저장(snapshot)
- 우리가 해석할 수 없는 바이너리 형태로 저장
- 복원이 빠르지만 특정 시점으로 복구 불가능
- 지정 시간 단위로 파일이 저장되어 데이터 손실 가능성이 있음

### 3가지 방법

- save <기간(초)> <기간 내 변경된 키의 개수> -> save 900 1 (900초 동안 1개 이상의 키가 변경된 경우)
- dbfilename
- dir