

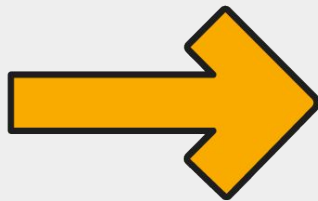


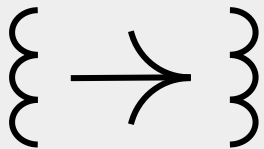
Google Developer Group
Editable University Name

확장 가능한 시스템 설계

with AWS

대규모 시스템 설계 - **GDG** 1기 서버 코어 김기수





1 단일 서버

트래픽을 고려하지 않은
서버설계

2 확장 시작

수직적 규모 확장 **vs** 수평적
규모 확장

로드 밸런서, 데이터베이스
다중화

3 캐시

데이터베이스 부담을 줄이고,
응답 시간을 더 빠르게

4 CDN

물리적 거리에 따른 지연을
줄여 정적 페이지 응답을 더
빠르게

5 메세지큐

서비스간 결합을 느슨하게!
확장성과 안정성을 확보하자

6 데이터베이스 규모 확장

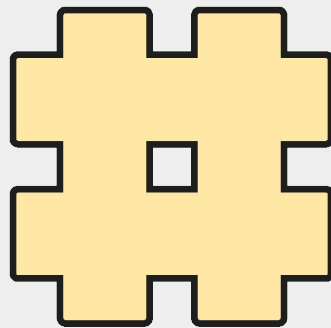
데이터베이스 부하를
줄여보자

Overview

서버는 예상 사용자 수 그리고 용도에 따라 서버 설계가 달라집니다.

예를들어 사내에서만 사용하는 서버에 (역시 용도에 따라 다르지만 일반적인 백오피스를 가정합니다) 최대 **100**만명을 수용하는 **scalable** 한 서버 설계는 오버엔지니어링이 될 것입니다.

한 명의 사용자를 지원하는 시스템에서 시작하여, 최종적으로 몇백만 사용자를 지원하는 시스템을 설계해보겠습니다.

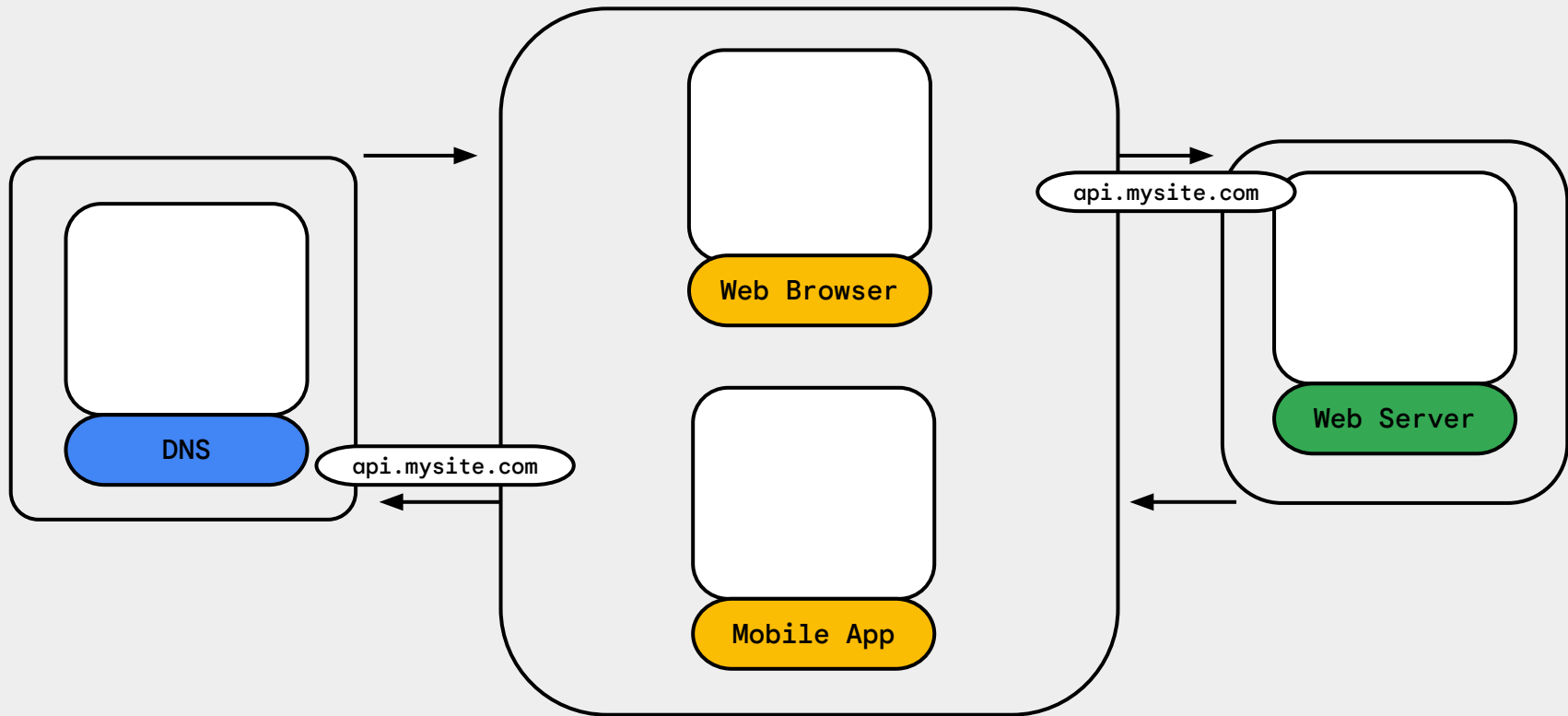


1-100

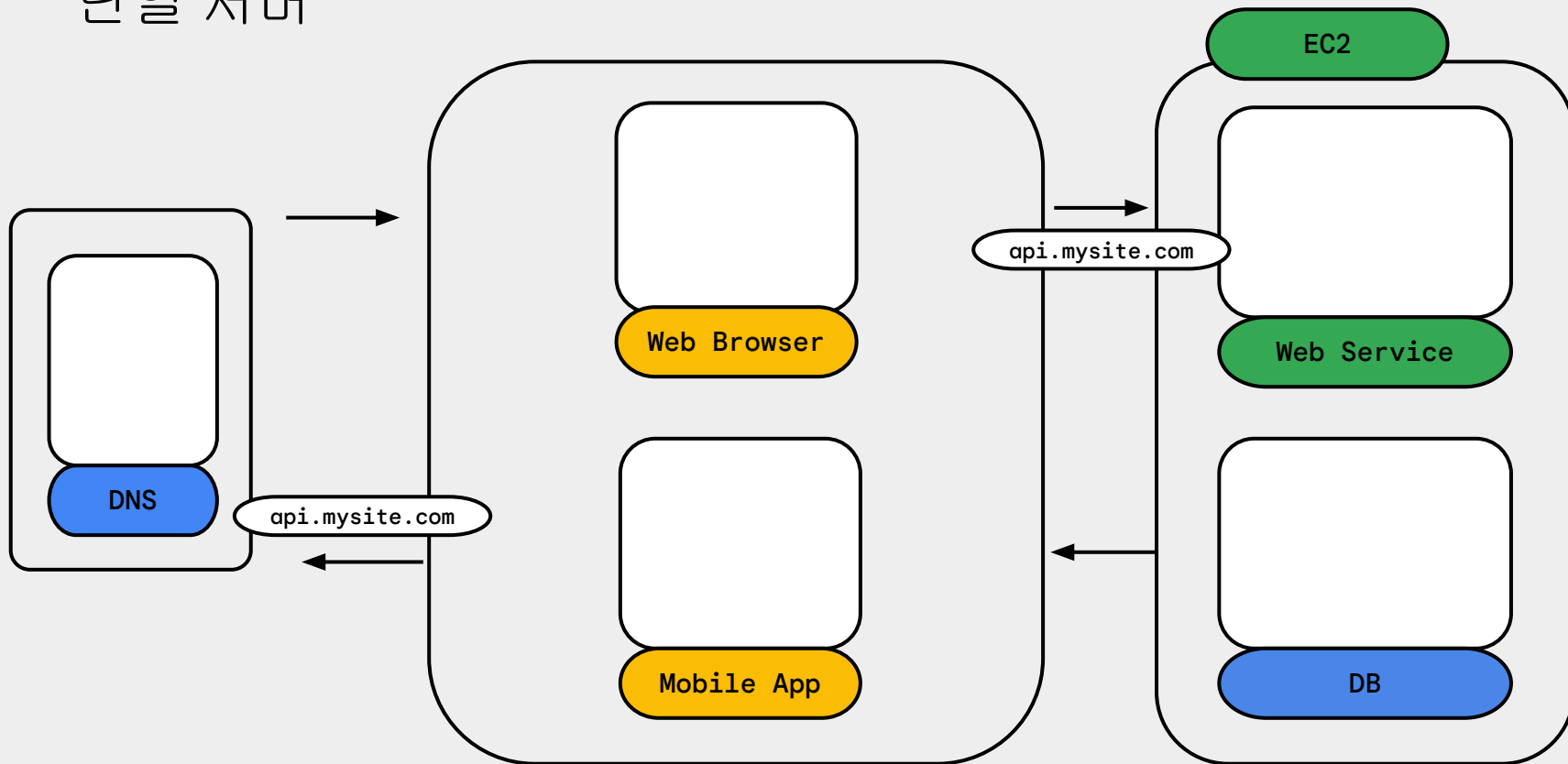
단일서버

- 천 리 길도 한 걸음 부터,,, 단 한대의 서버부터

단일 서버



단일 서버



100-1k

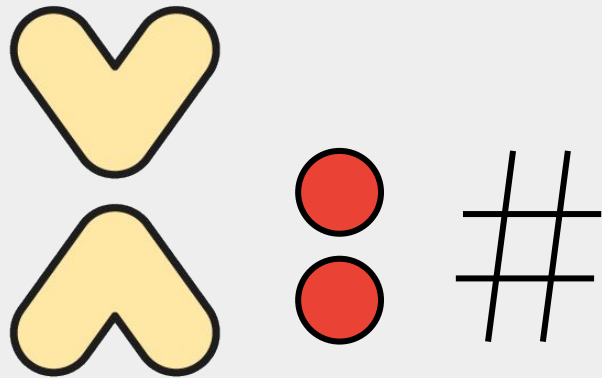
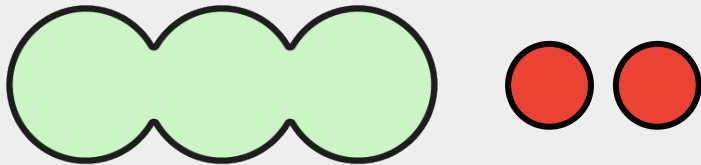
확장의 시작
- 수평적 확장

어떤 확장 방식?

수직적 규모 확장 - **scale up**

서버 성능 업그레이드!

고사양 자원 투입!

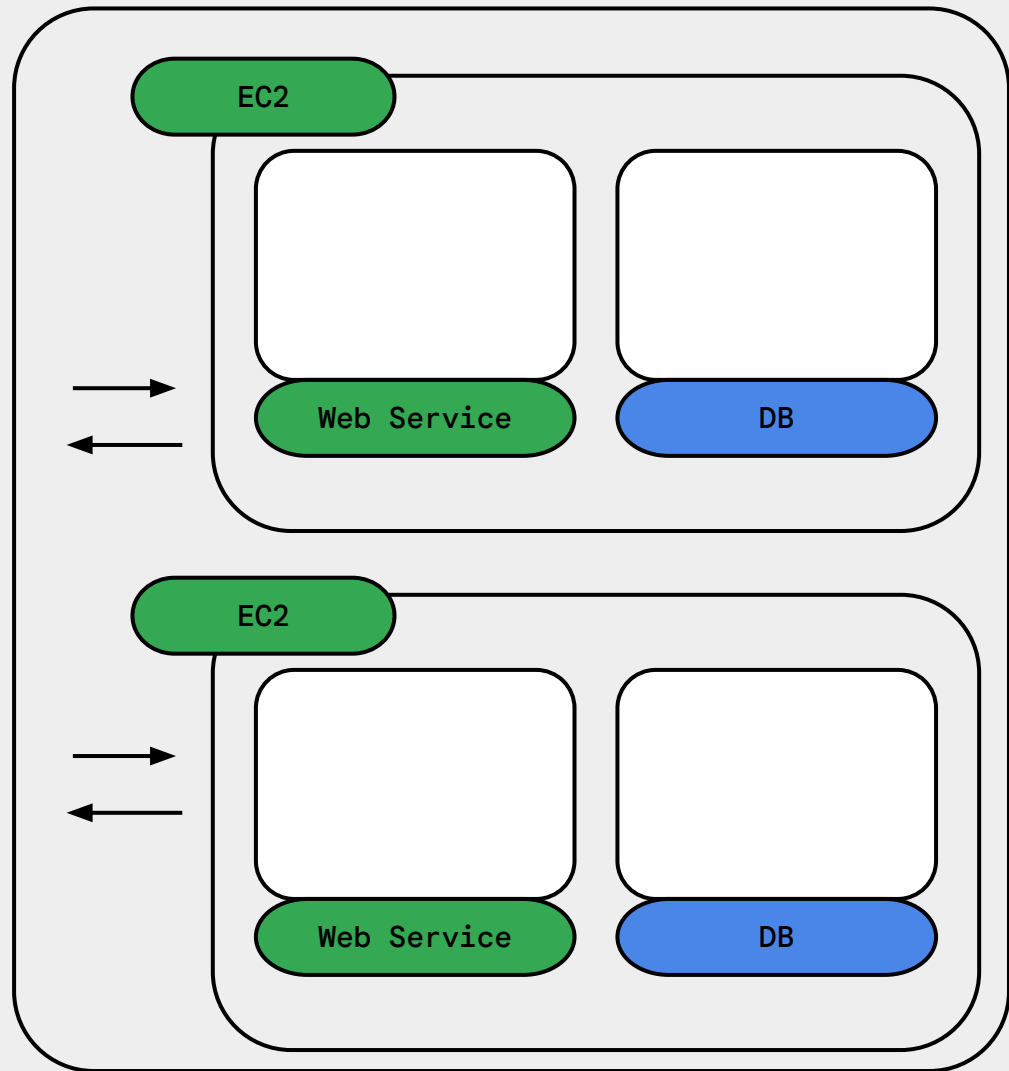
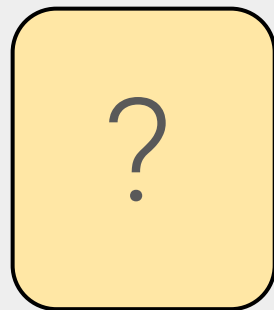


수평적 규모 확장 - **scale out**

서버를 여러개!

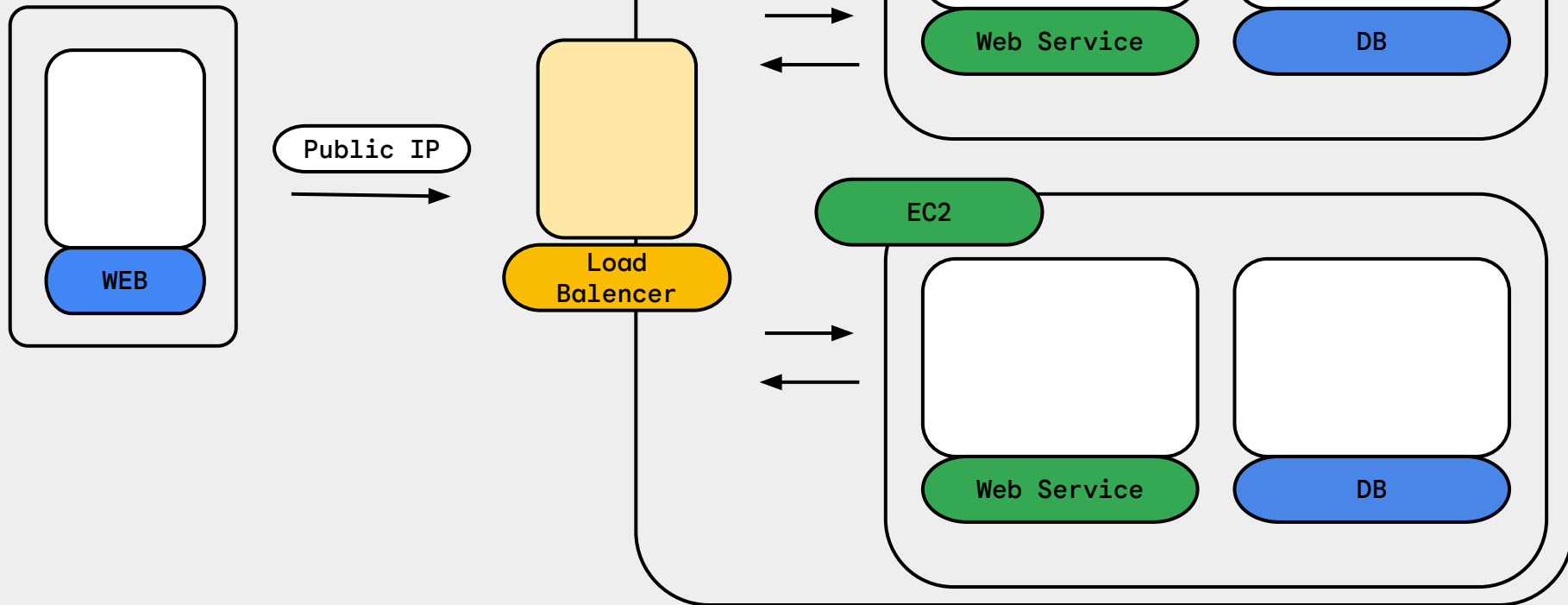
서버 추가투입!

수평적 확장 서버



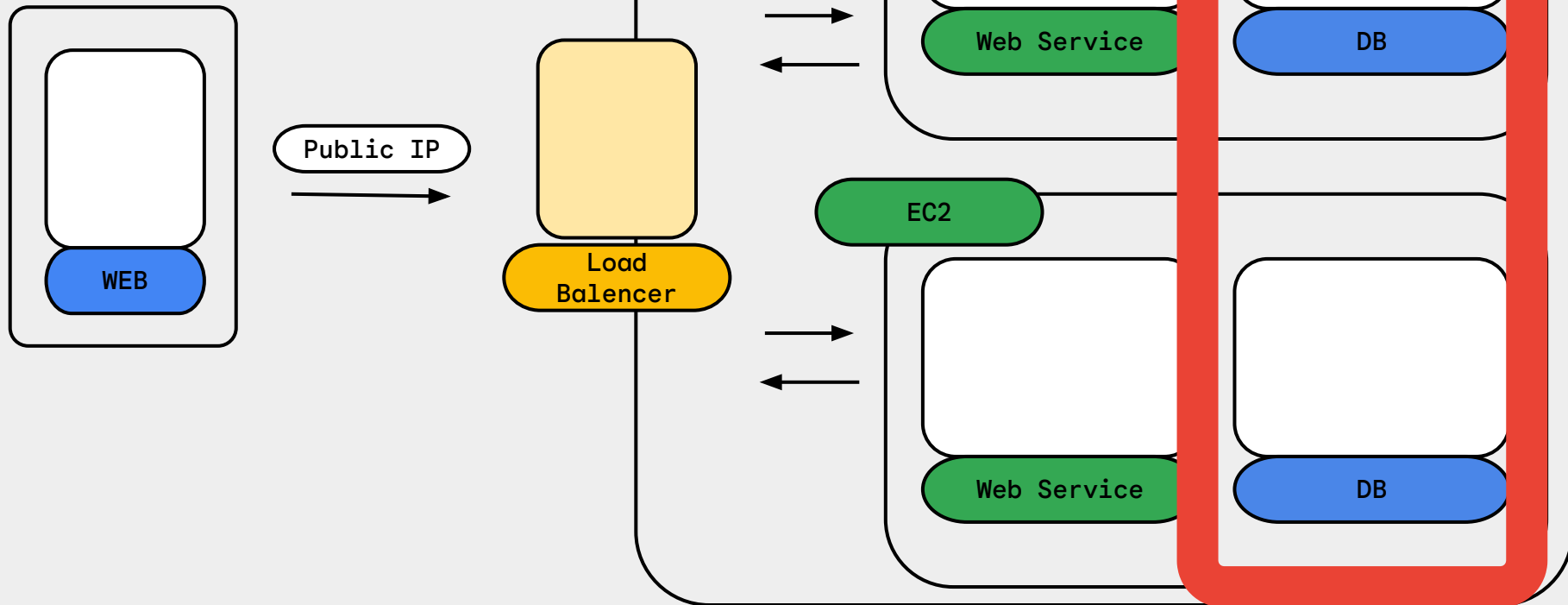
수평적 확장 서버

- Load Balancer



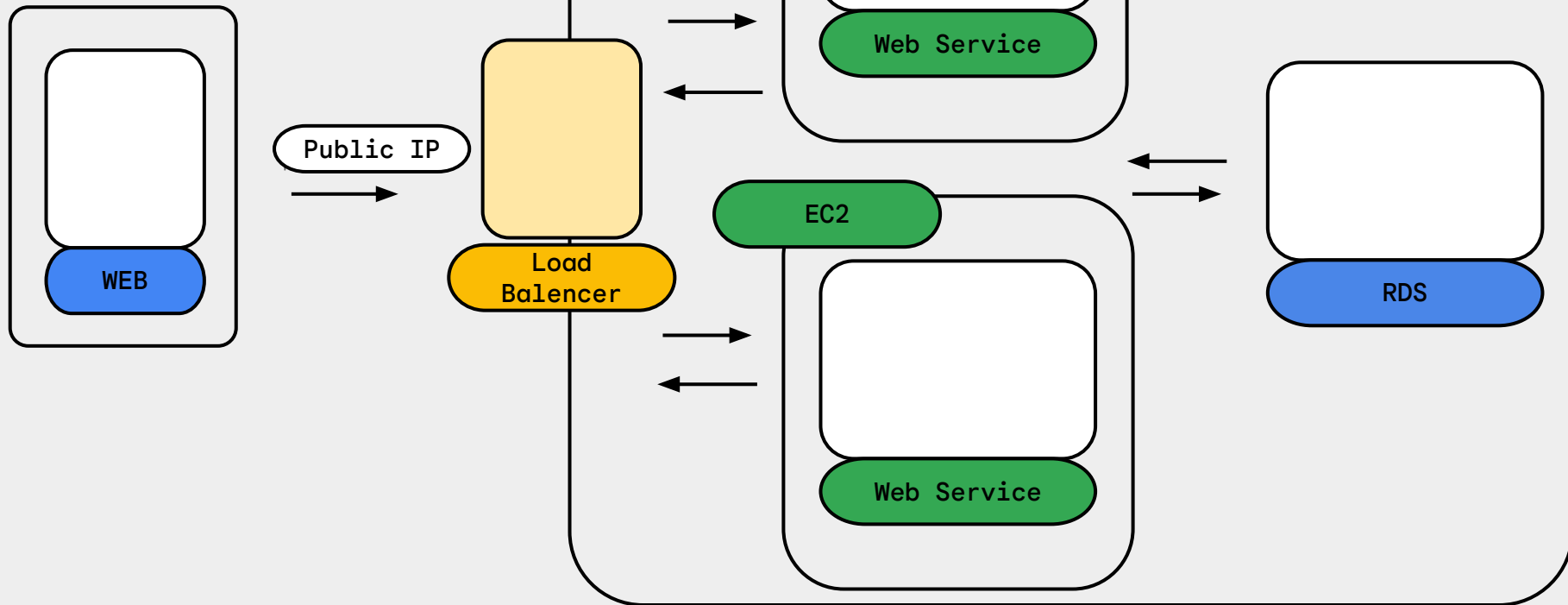
수평적 확장 서버

- Database



수평적 확장 서버

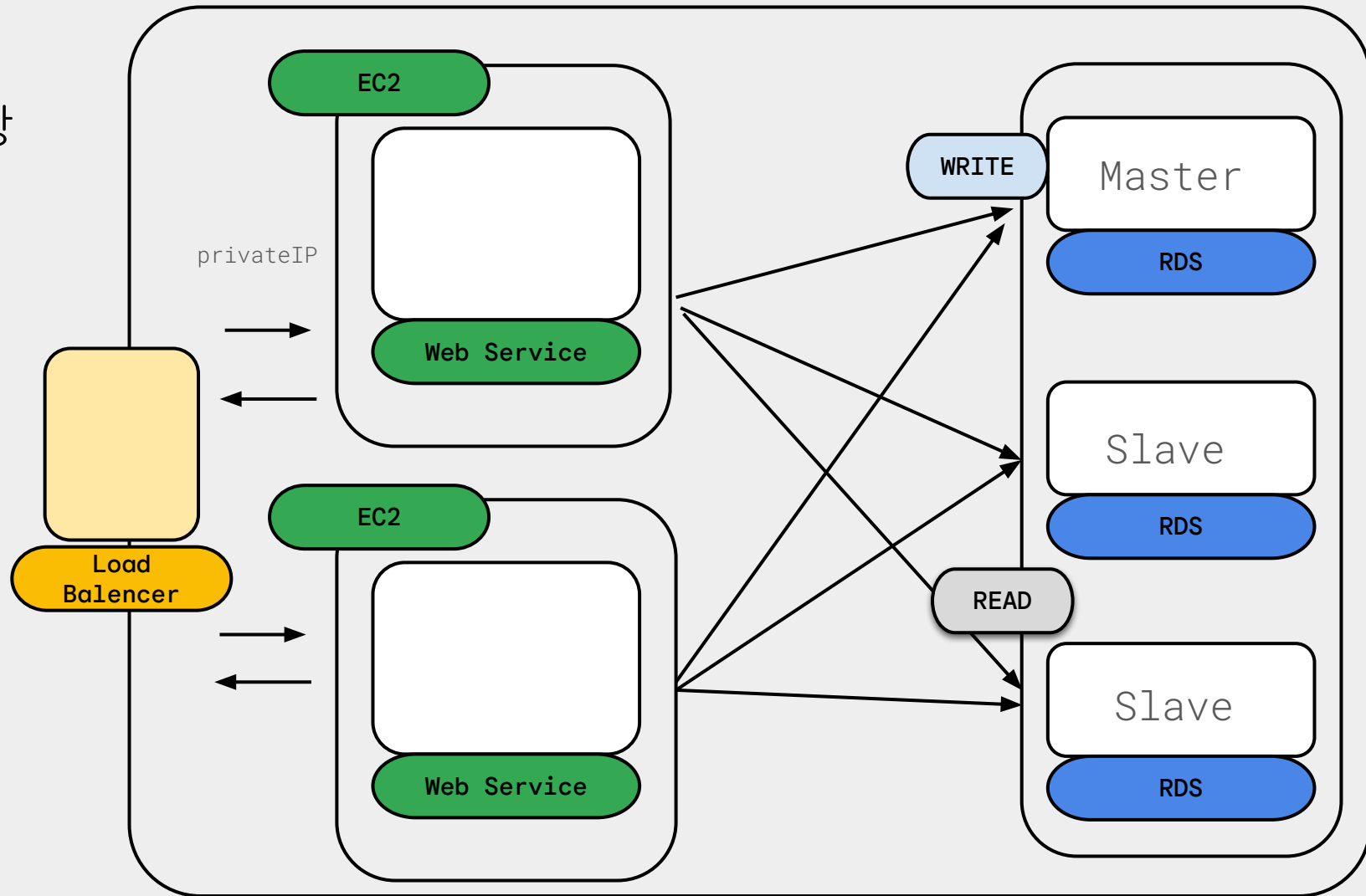
- Database



1k-10k

DB : SOS π π

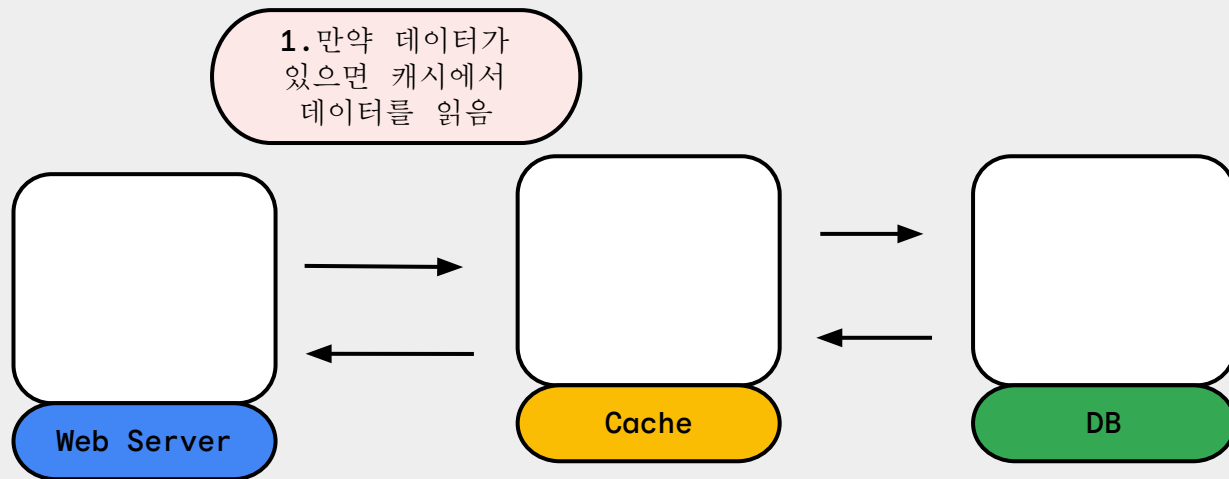
DB 확장



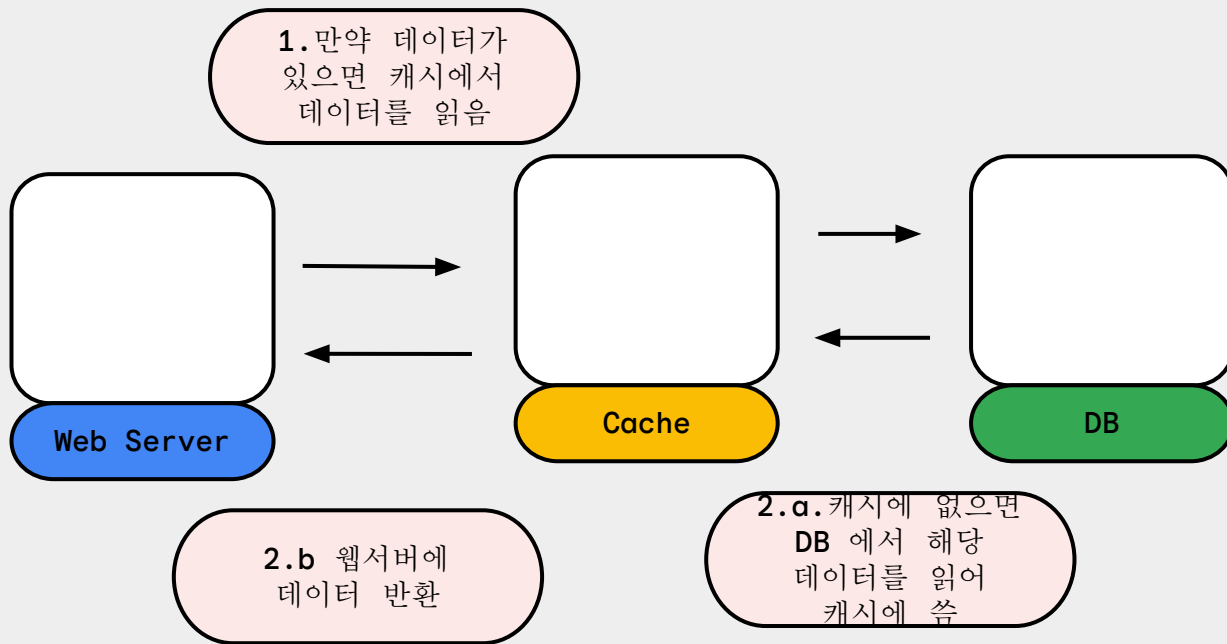
1k-10k

고도화 : Cache

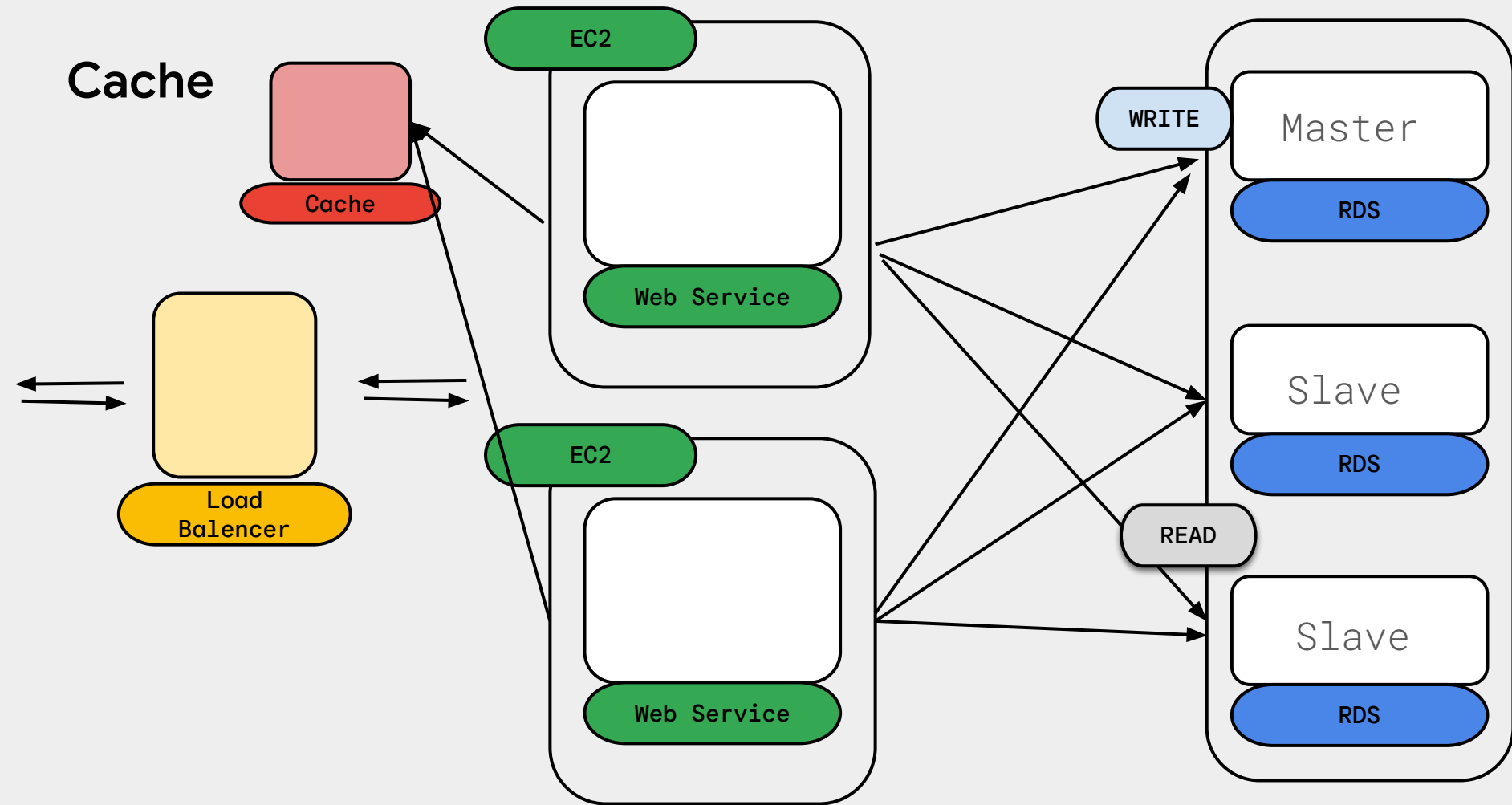
Cache 흐름



Cache 흐름



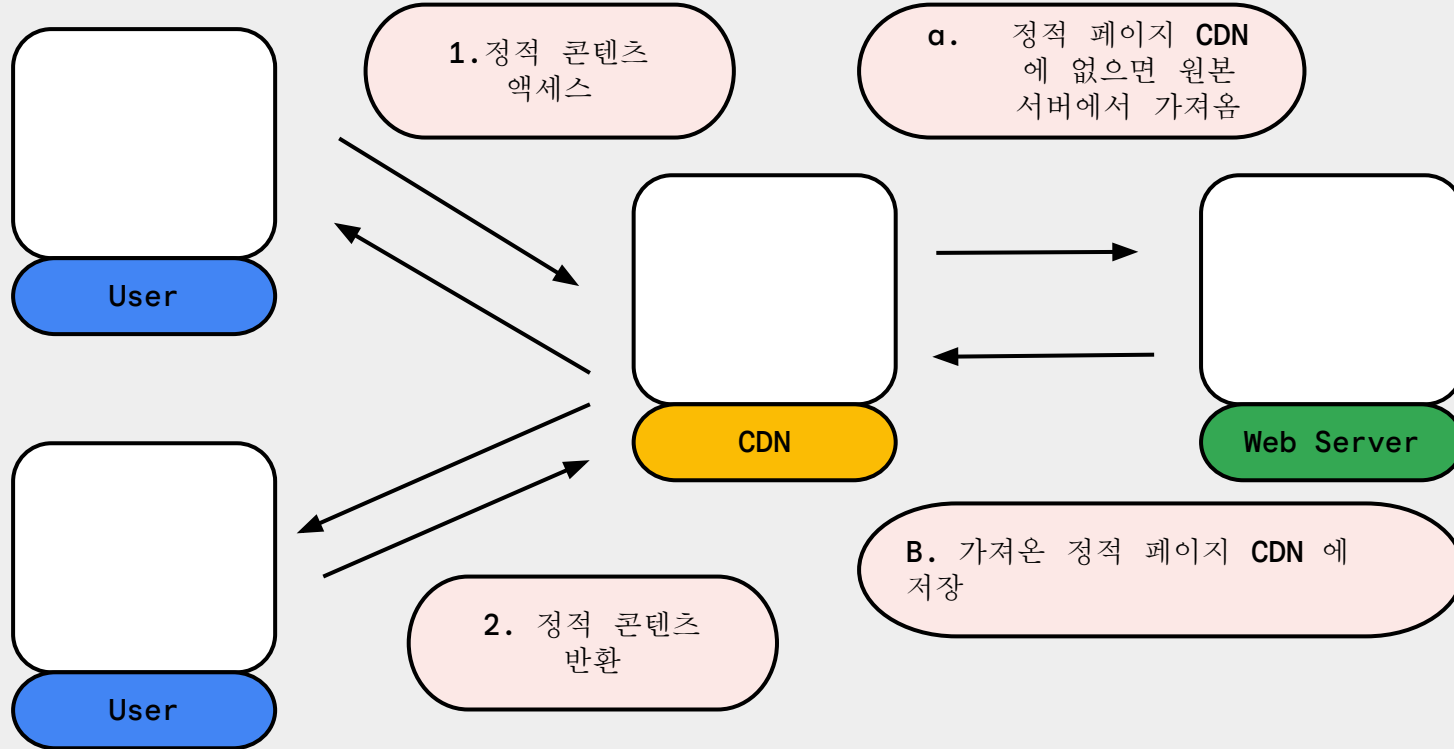
Cache

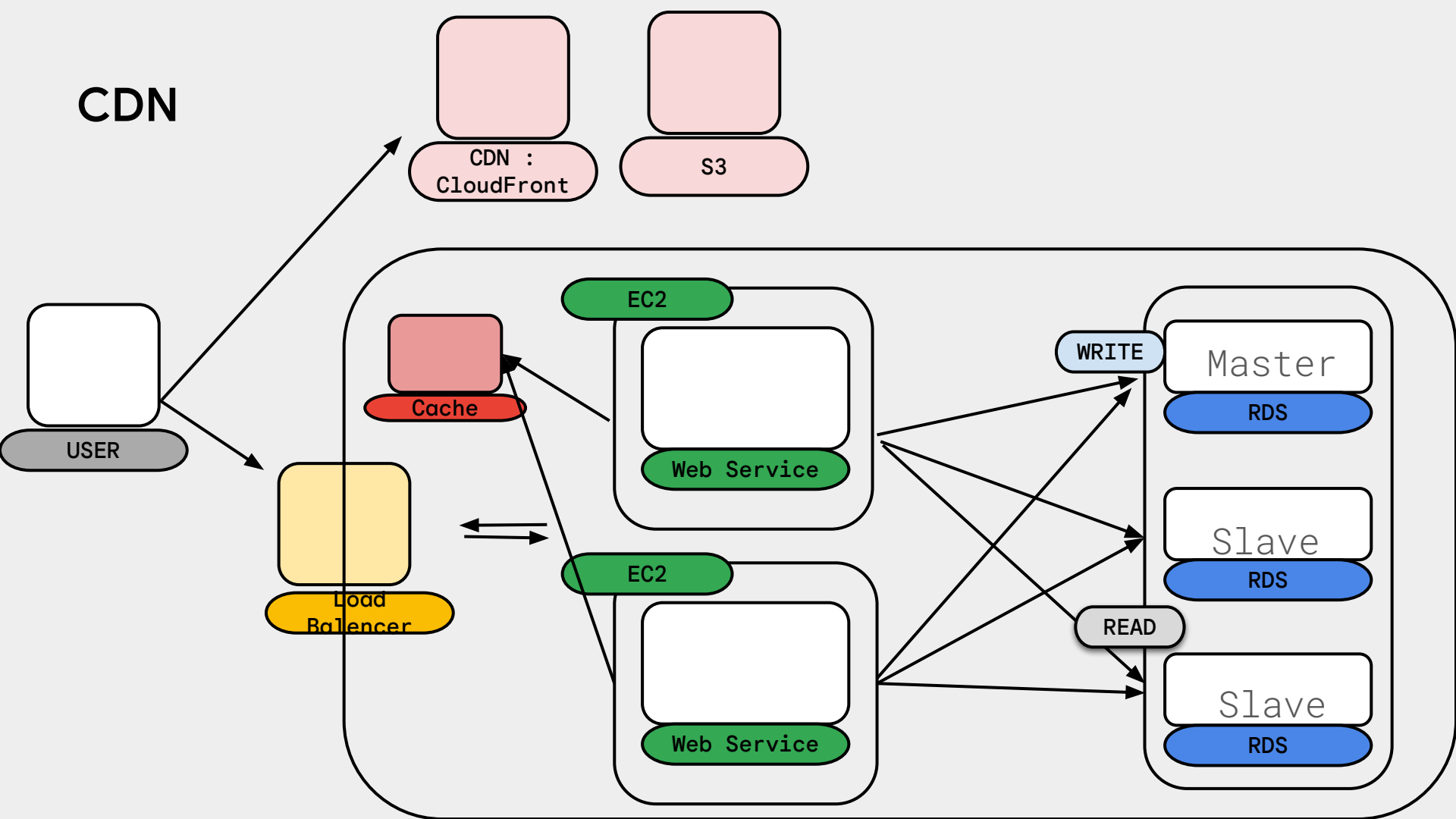


10k-100k

콘텐츠는? CDN

CDN

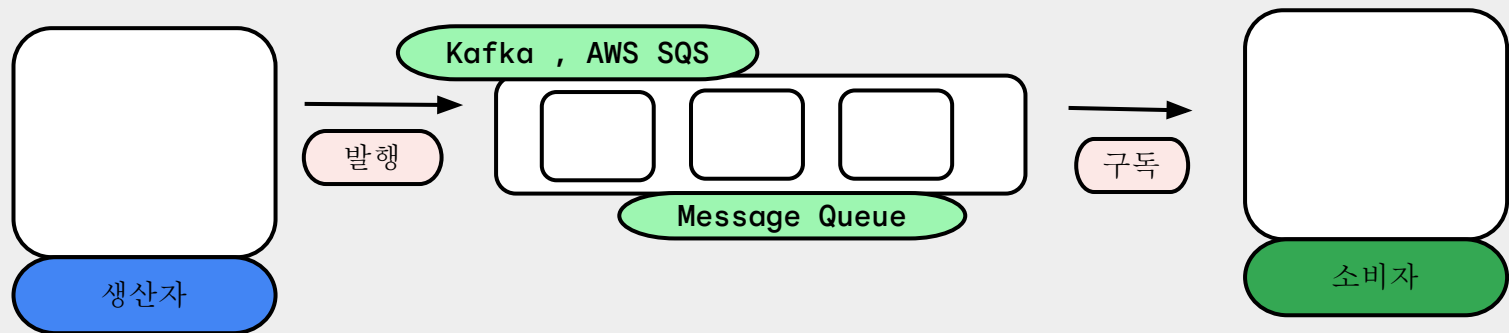


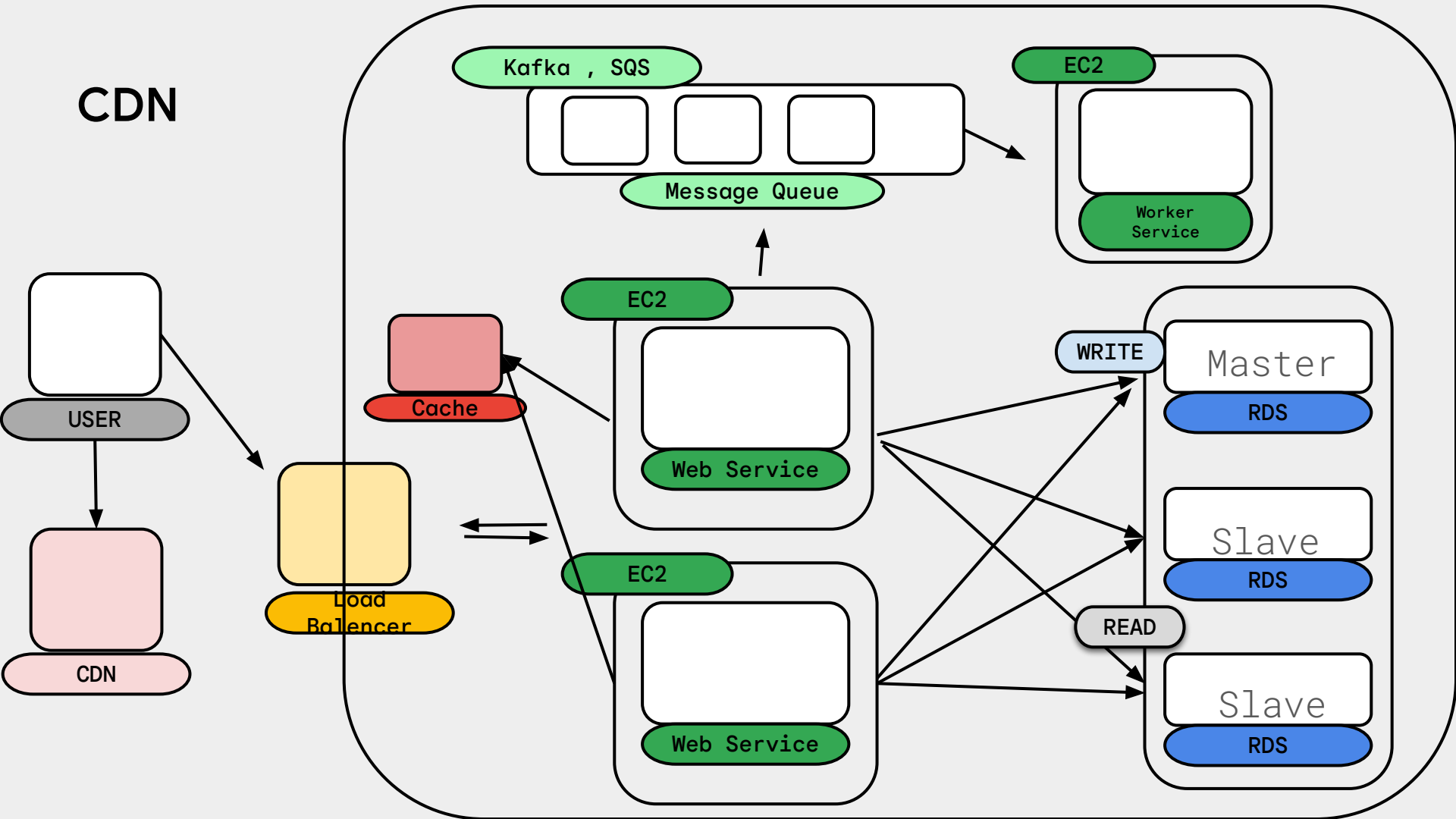


100k-1M

서비스간 결합을 느슨하게
-메세지 큐

Message Queue 흐름

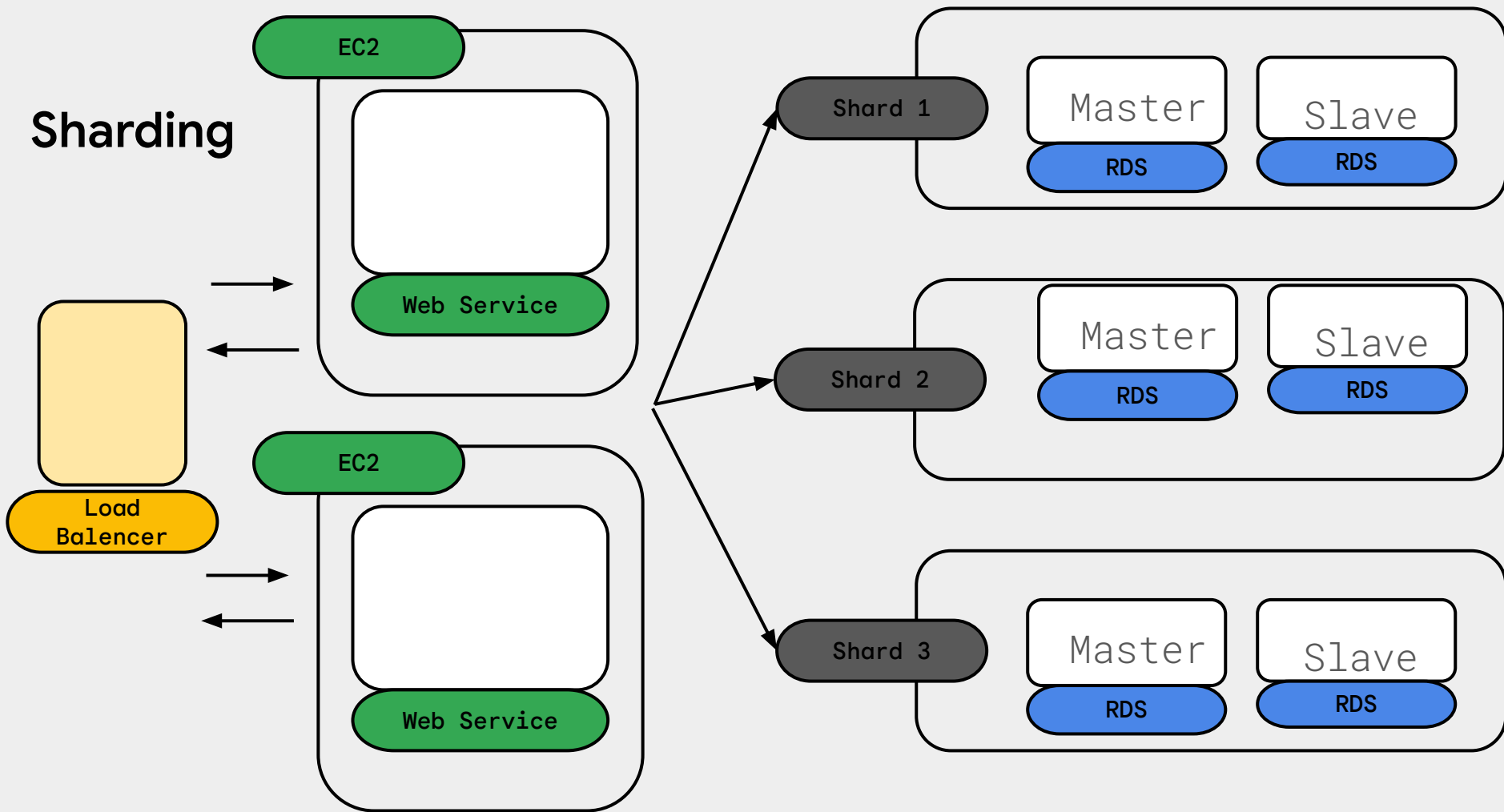




1M-50M

아무리 나뉘어도 **DB**가 느리다
Sharding

Sharding



Q&A