

CS Study 5주차

Network - 3

김신아

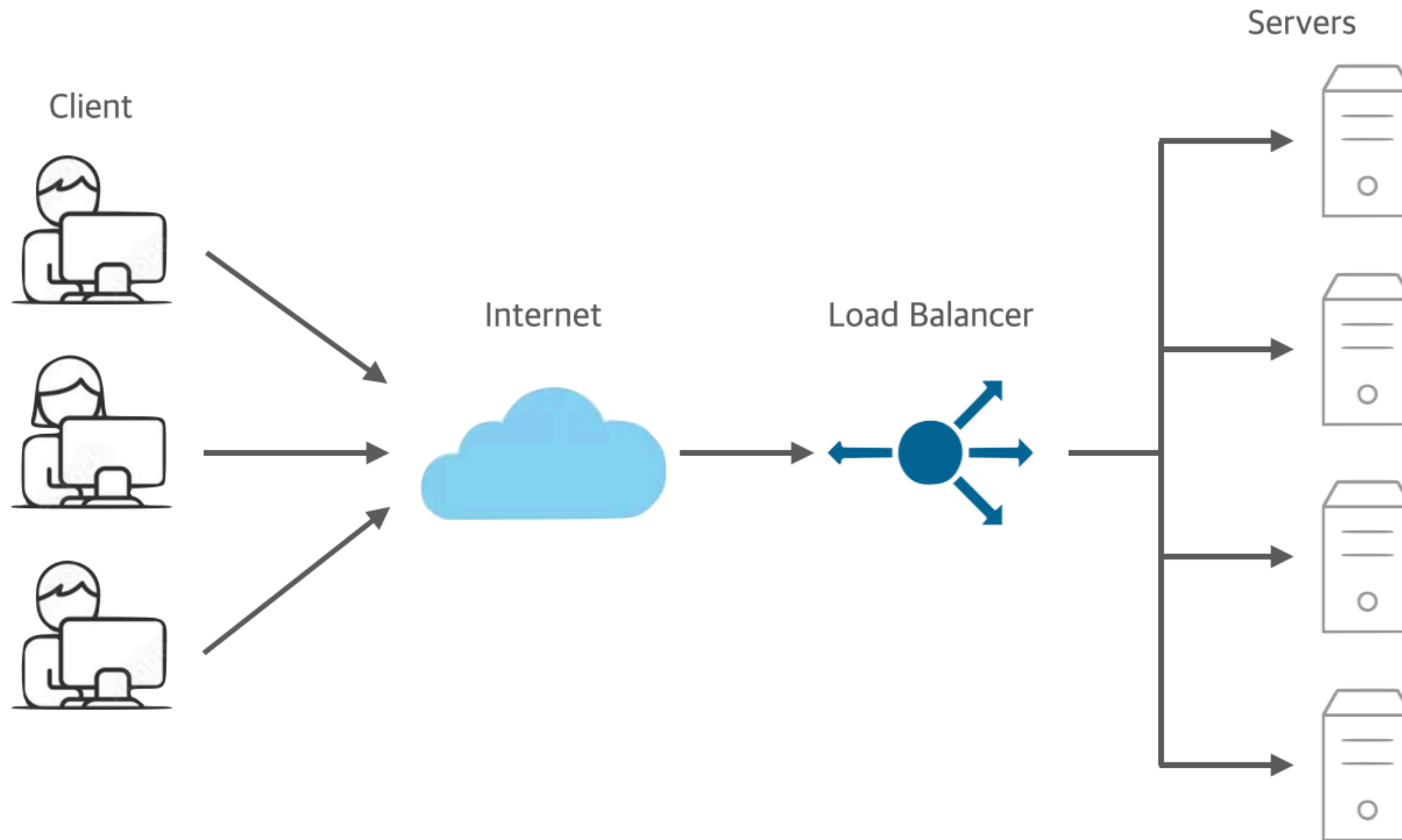
오늘 알아볼 내용은?

로드 밸런싱

주의할점은?

의식의 흐름대로 공부했다는 것..

로드밸런싱(Load Balancing)



- 로드밸런싱(Load Balancing)

: 둘 이상의 CPU or 저장장치와 같은 컴퓨터 자원들에게 작업을 나누는 것

로드밸런싱(Load Balancing)

- 로드밸런싱이 필요한 이유

웹 사이트에 접속하는 인원이 급격히 늘어나고 있다.

모든 트래픽을 감당하기엔 1대의 서버로 부족하다.

대응 방안 두 가지

- **Scale-up** : 하드웨어의 성능 올리기

- **Scale-out** : 여러대의 서버가 나눠서 일하도록 만드는 것

하드웨어 향상 비용이 비싸고, Scale-out을 이용하게 되면 여러대의 서버 덕분에 무중단 서비스를 제공할 수 있다. 이때 여러 서버에게 균등하게 트래픽을 분산시켜주는 역할을 하는 것이 로드밸런서 (Load Balancer)이다.

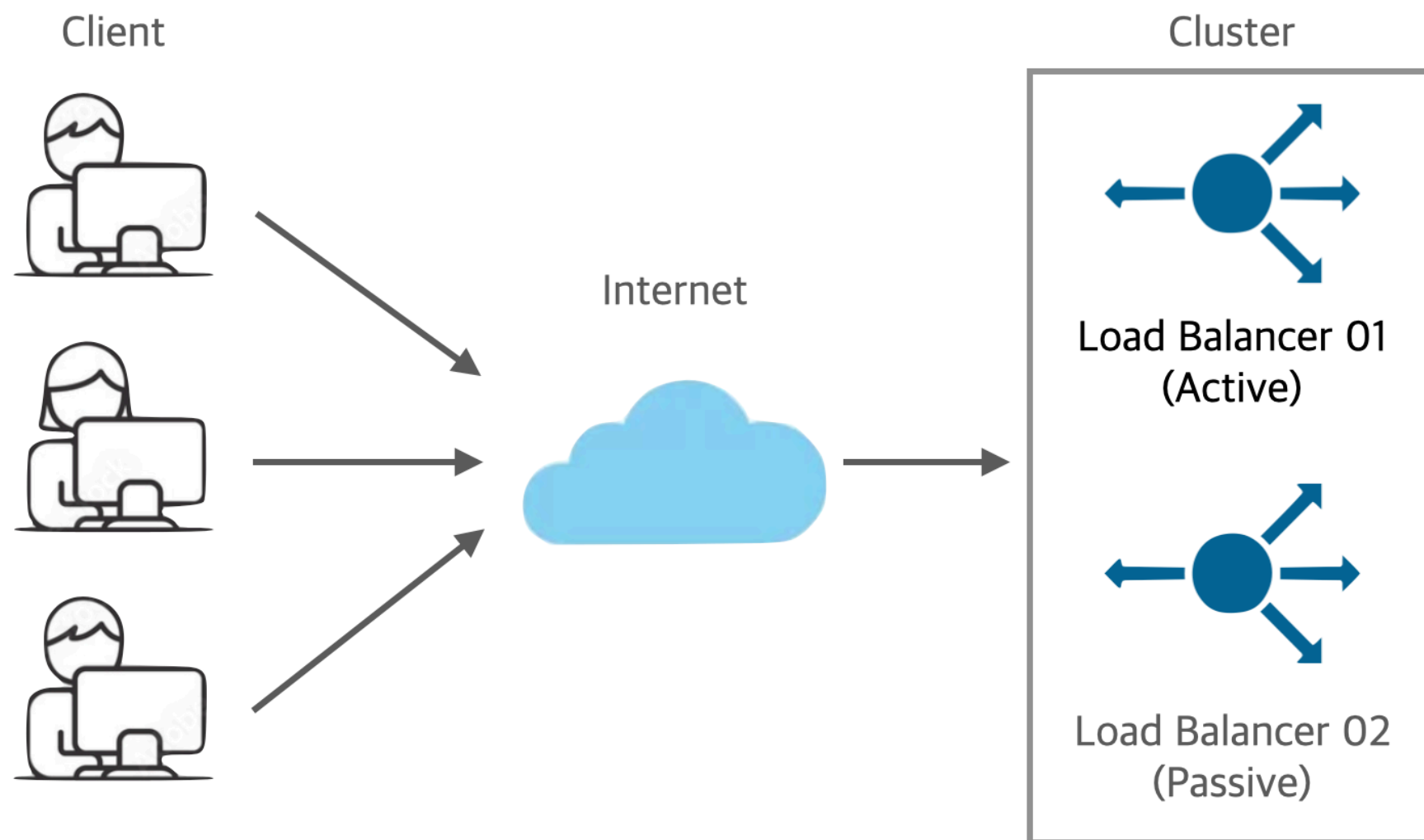
로드밸런싱(Load Balancing)

- 로드밸런서(Load Balancer)가 서버를 선택하는 방식
 - 라운드 로빈(Round Robin) : CPU 스케줄링의 라운드 로빈 방식 활용
 - Least Connections : 연결 개수가 가장 적은 서버 선택(트래픽으로 인해 세션이 길어지는 경우 권장)
 - Source : 사용자 IP를 해싱하여 분배 (특정 사용자가 항상 같은 서버로 연결되는 것 보장)

로드밸런싱(Load Balancing)

- 로드밸런서(Load Balancer)의 장애 대비

: 서버를 분배하는 로드 밸런서에 문제가 생길 수 있기 때문에 로드 밸런서를 이중화하여 대비



IP 해시 방식

- IP 해시 방식

: 클라이언트의 IP 주소를 해시 처리하고, 특정 서버로 요청을 매핑하여 트래픽을 분배하는 방식이다. 사용자의 IP 주소를 해시 처리하기 때문에 사용자가 항상 동일한 서버로 연결되는 것을 보장한다.

* Hashing, 임의의 길이를 지닌 데이터를 고정된 길이의 데이터로 매핑하는 것, 또는 그러한 함수

IP 해시 방식

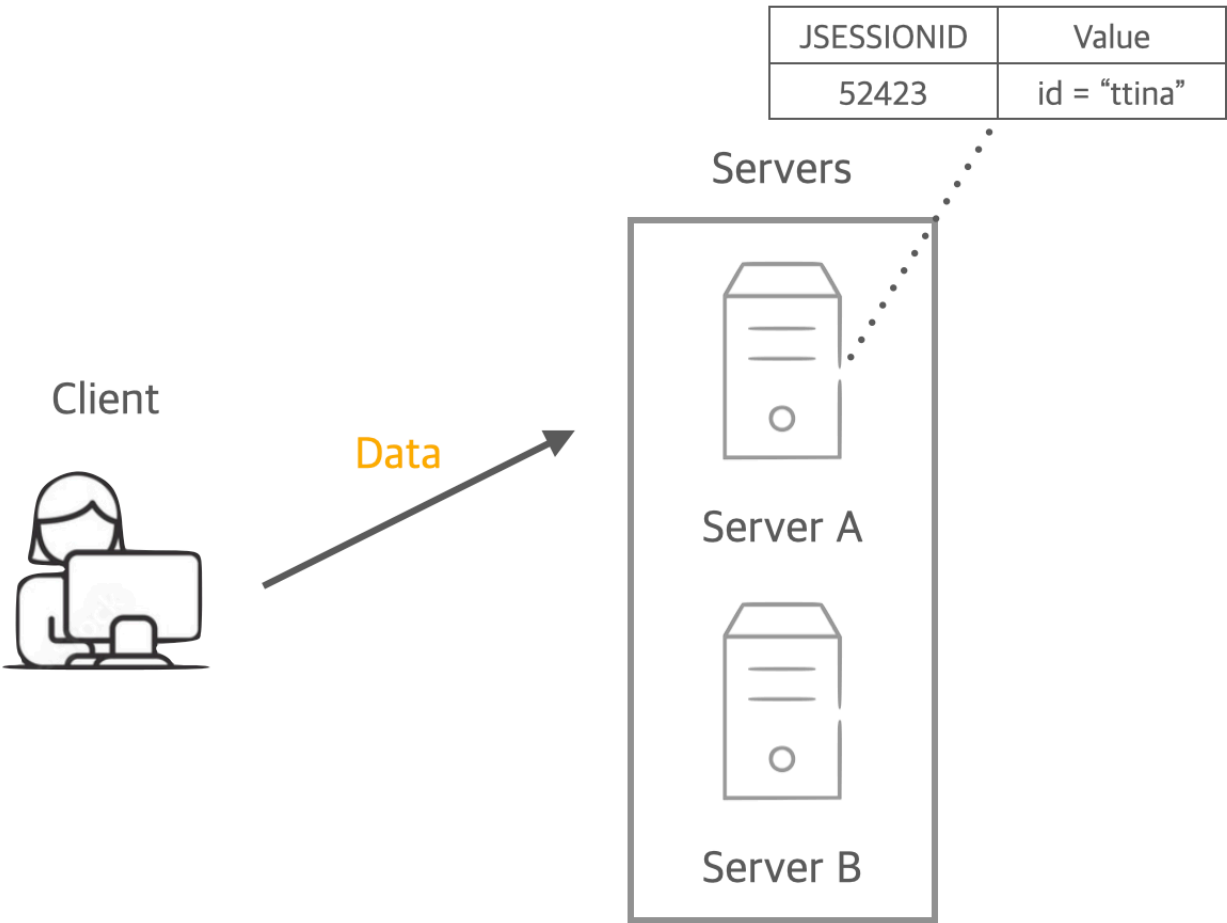
- IP 해시 방식을 사용하는 이유

: IP Hash Table은 세션 클러스터링이 불가능한 경우 세션 일관성(Session Persistence)을 위해서 IP를 기반으로 어느 인스턴스로 보내야 할지 처음부터 확정하는 방법이다.

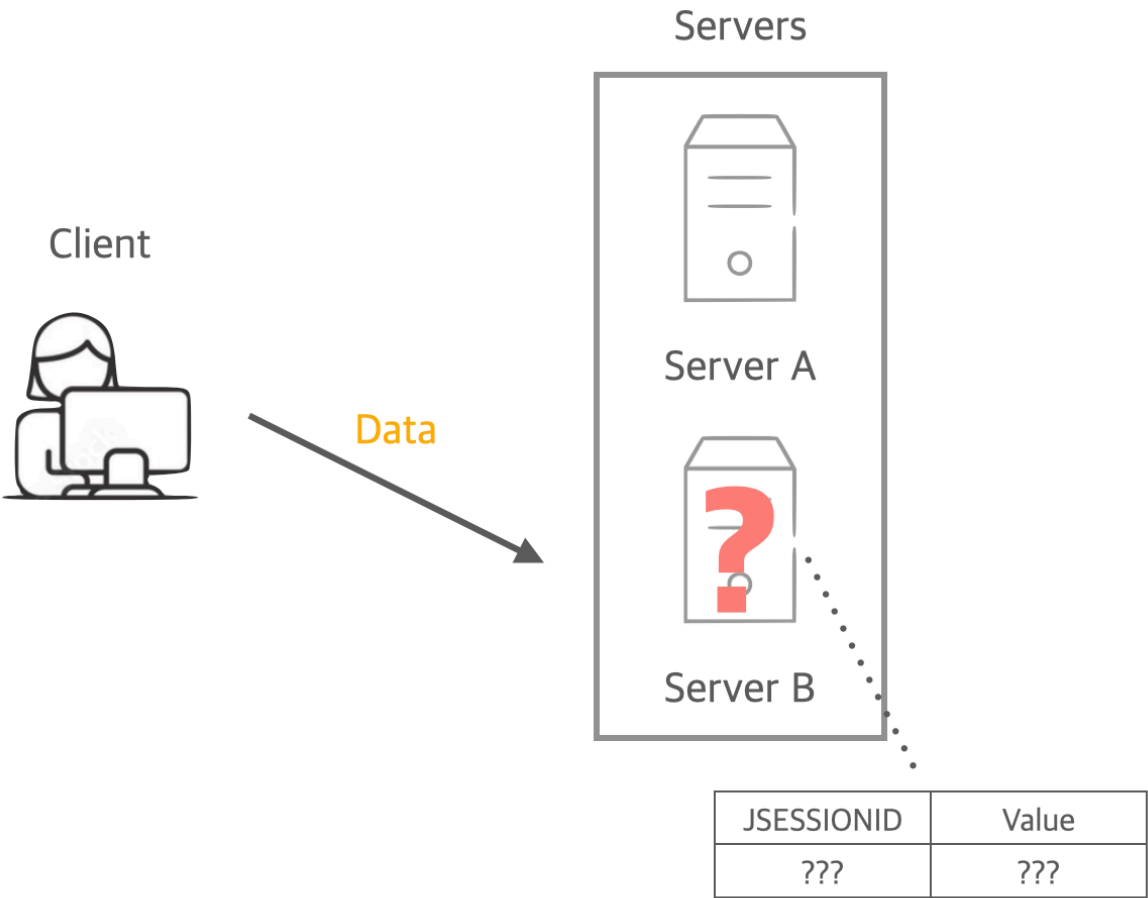
sticky-session 을 구현하기 위한 방법 중 하나이다.

IP 해시 방식

First



Second



Sticky Session

- **Sticky Session**

- 첫 Request에 대한 응답을 준 서버에 껌딱지처럼 붙어있는 것이다.
- 특정 세션의 요청을 처음 처리한 서버로만 보내는 것이다.

- **특정 서버로 요청 처리 고정시키는 방법**

- Cookie
- 클라이언트의 IP tracking

- **단점**

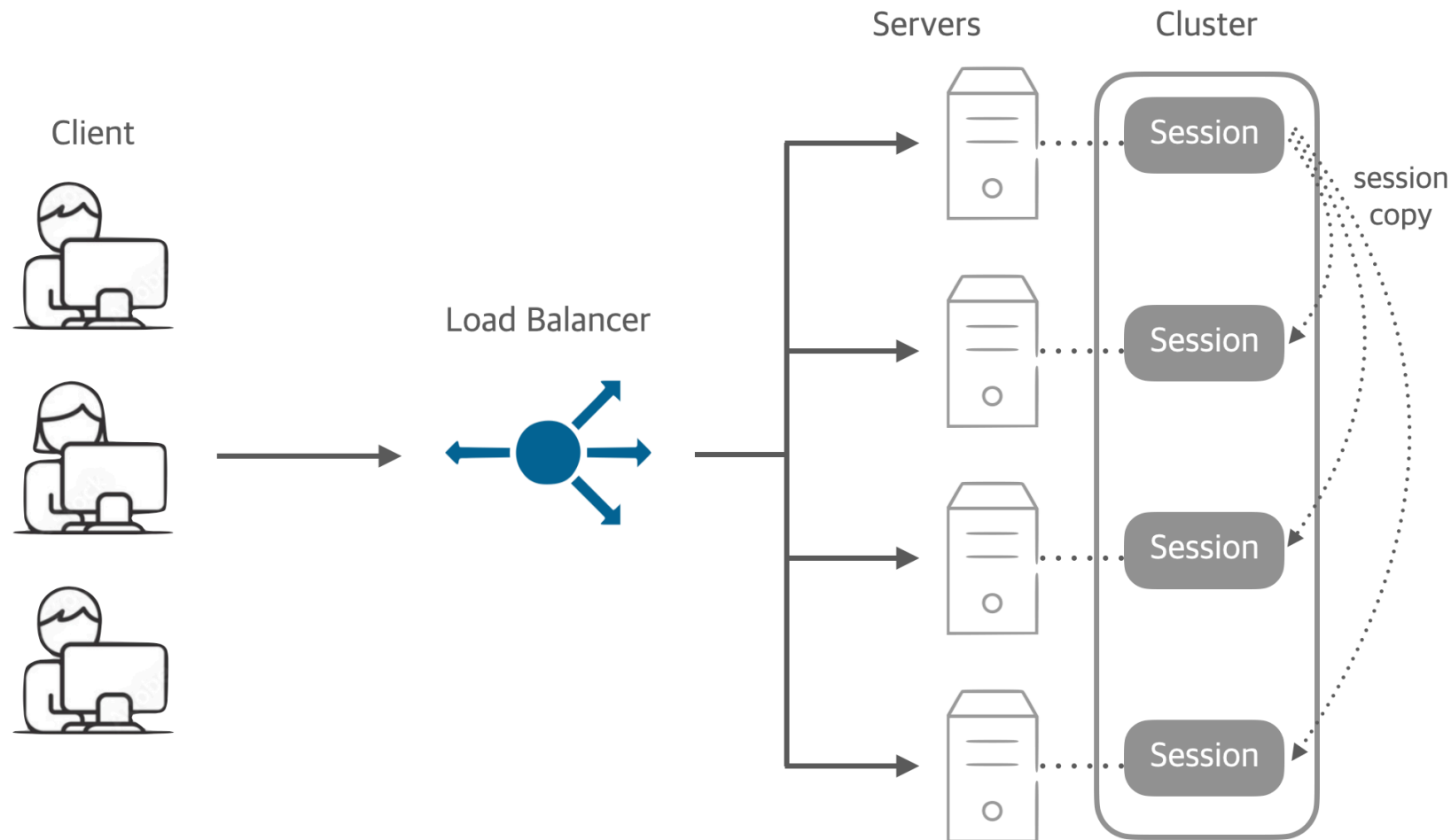
- 로드밸런싱이 잘 동작하지 않을 수 있다.
- 특정 서버만 과부하가 올 수 있다.
- 특정 서버 Fail시 해당 서버에 붙어 있는 세션들이 소실될 수 있다.

=> 이러한 단점들을 고려한 세션 관리 기법 중 Session Clustering 방식이 있다.

Session Clustering

• Session Clustering

: 여러 WAS의 세션을 동일한 세션으로 관리하는 것



• 단점

- Scale-out 관점에서 새로운 서버가 하나 뜰 때마다 기존에 존재하던 WAS에 새로운 서버의 IP/Port를 입력해서 클러스터링 해주어야 한다.
 - 새로운 서버를 띄우면 기존 서버에 수정이 발생하고, 휴먼 에러가 발생할 가능성도 충분하다.
- => 이러한 단점들을 고려하여 session server를 따로 두고 관리하는 방식이 있다.

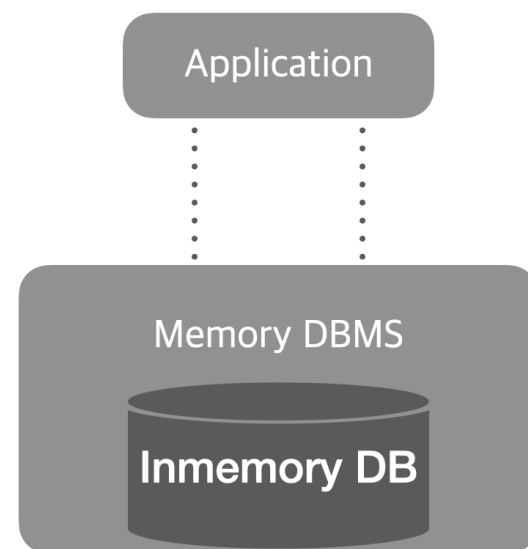
Session Storage 분리

- Session Storage 분리

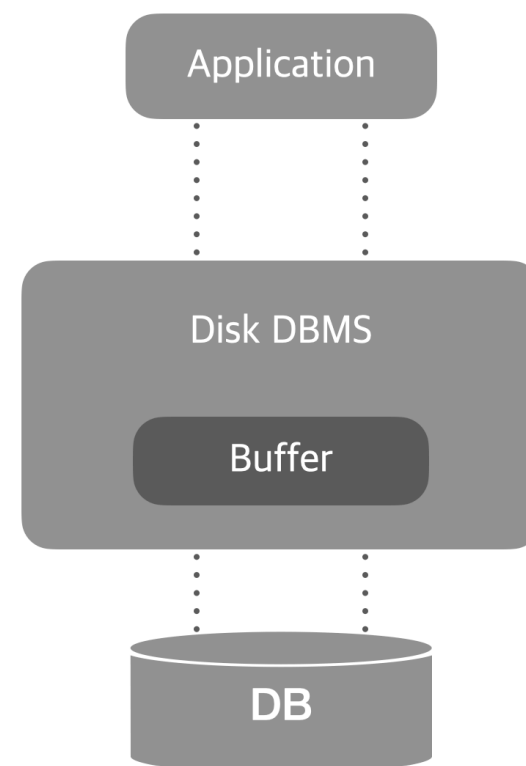
: 기존 서버가 갖고있는 세션 저장소를 이용하는 것이 아니라, 별도의 세션 저장소를 사용하여 세션 스토리지를 분리하는 것이다.

- Disk Based Database

: 데이터를 디스크에 저장, 관리하는 데이터베이스이다. 우리가 흔히 알고 있는 Oracle, MySQL, MS-SQL 등이 여기에 해당한다. 디스크에 저장된 데이터는 지속성 및 안정성이 보장되고, 메모리에 비해 가격이 저렴하여 쉽게 확장이 가능하기 때문에 대용량 데이터를 처리하는 데 용이하다.



Inmemory DBMS

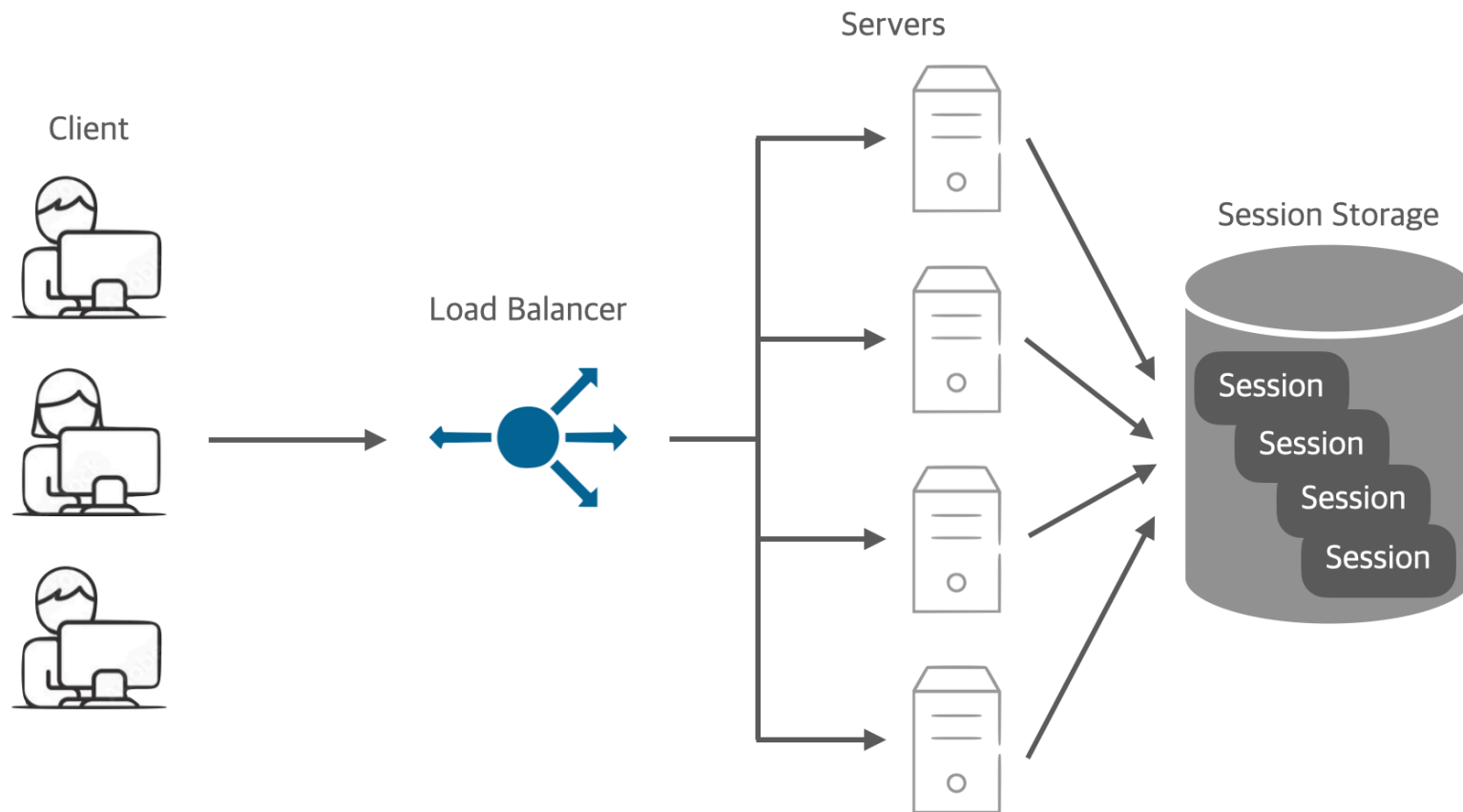


Disk DBMS

Inmemory DB

• Inmemory DB 방식

: 데이터 스토리지의 메인 메모리에 설치되어 운영되는 방식의 데이터베이스 시스템이다. 메모리 접근이 디스크 접근보다 빠르며 데이터를 조회할 때 검색 시간이 줄어든다. 대표적으로 Redis와 Memcached가 있다.

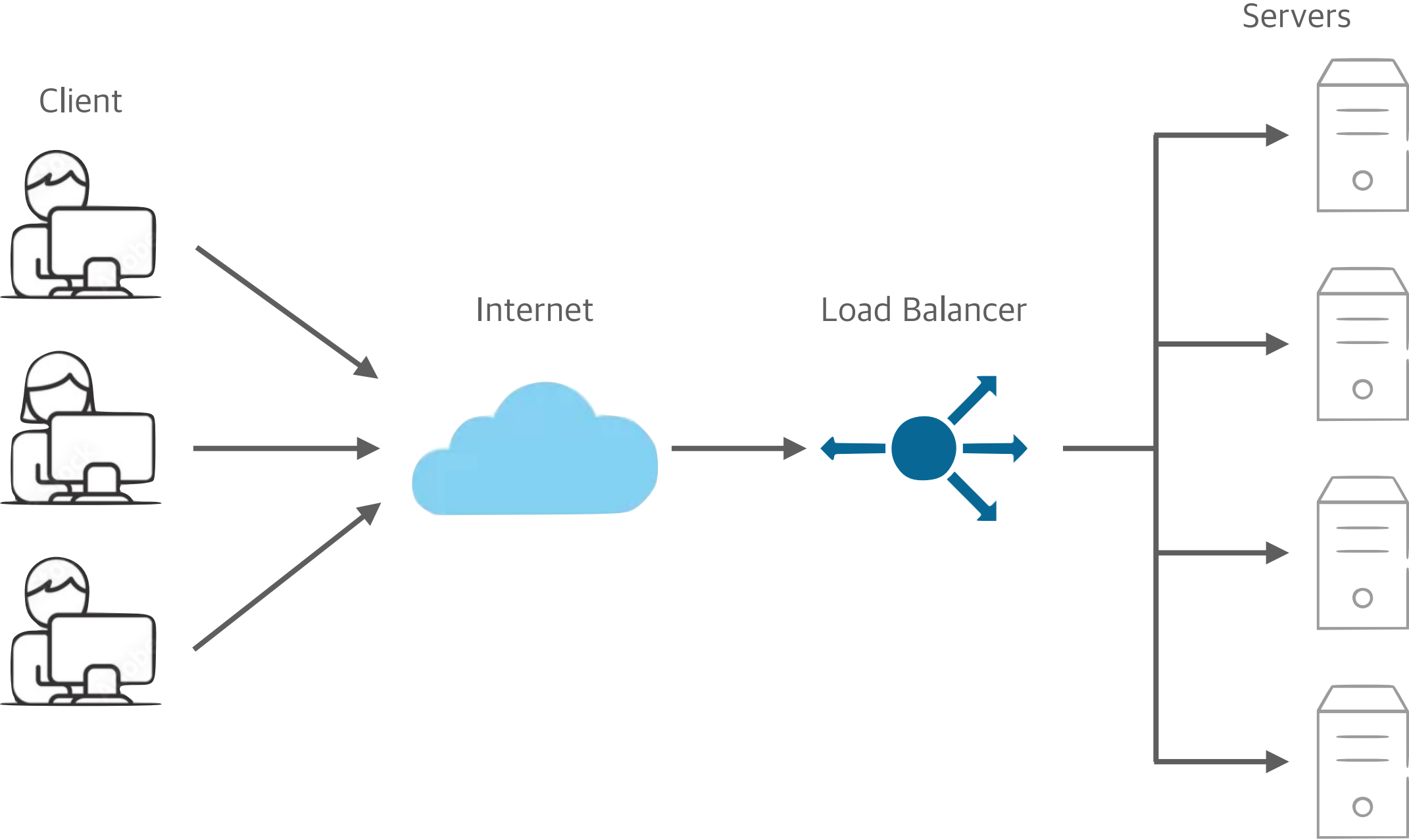


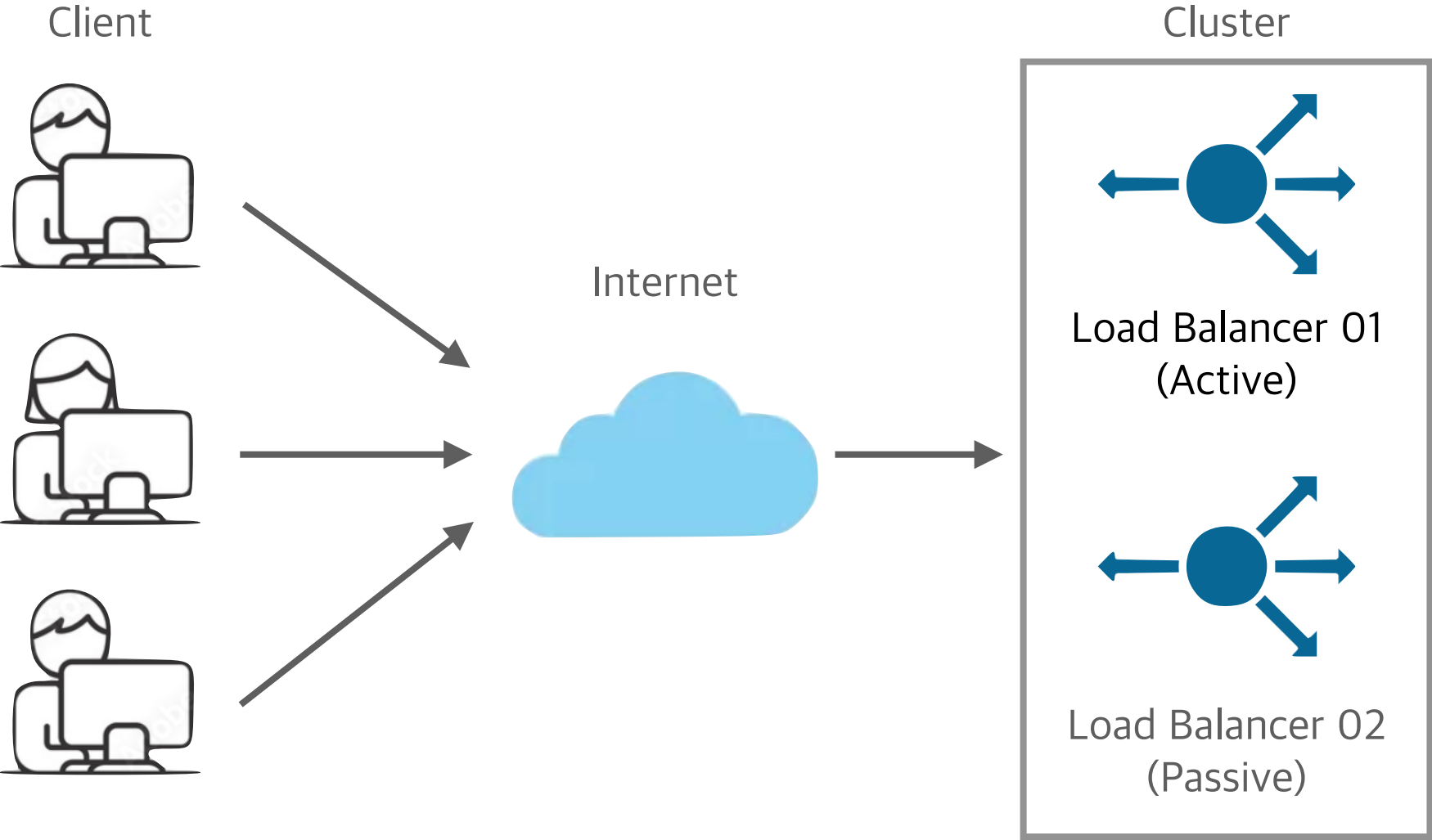
• 단점

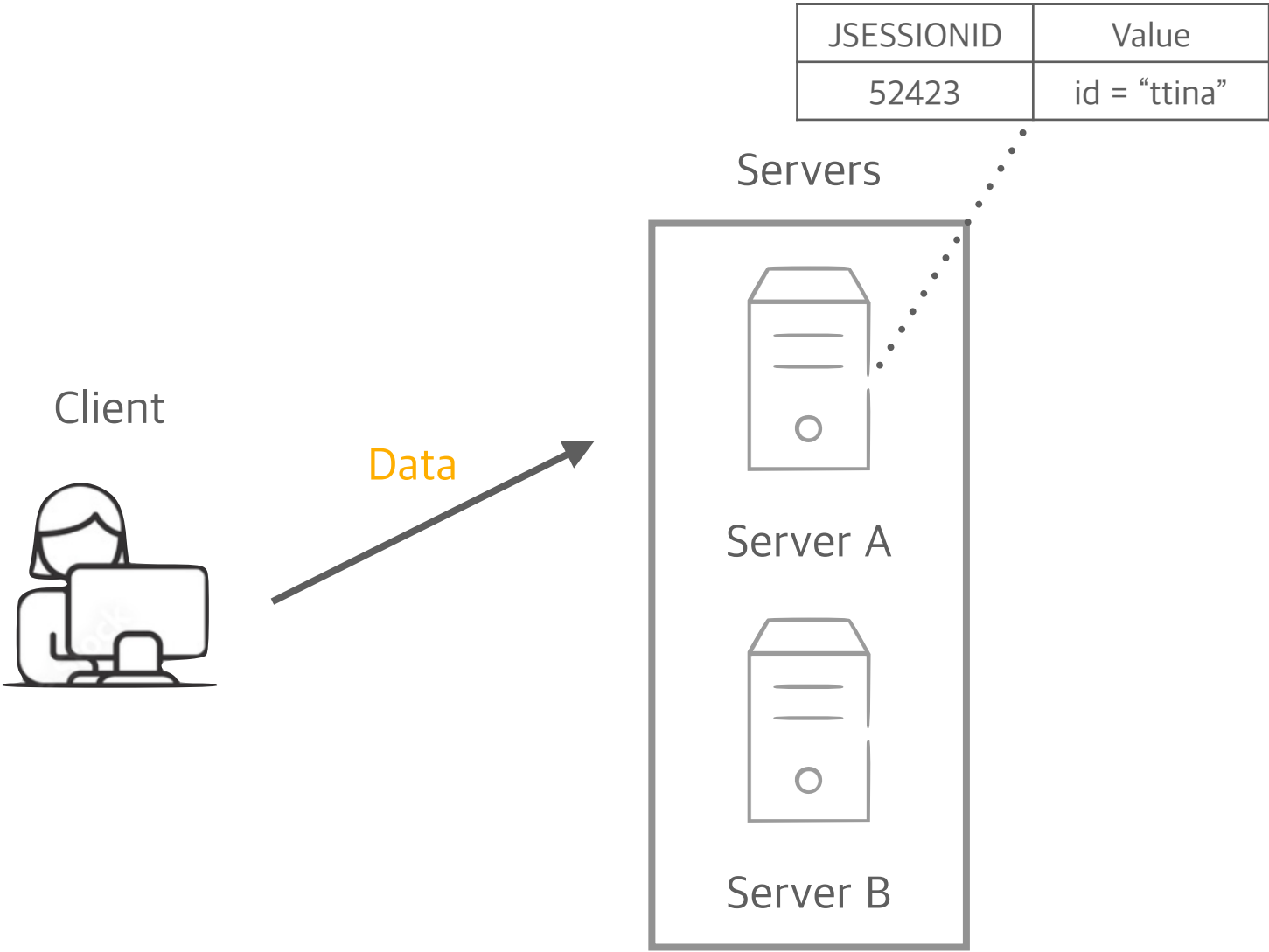
- 메모리가 휘발성이다. DB서버 전원이 꺼지면 자료들이 삭제된다.

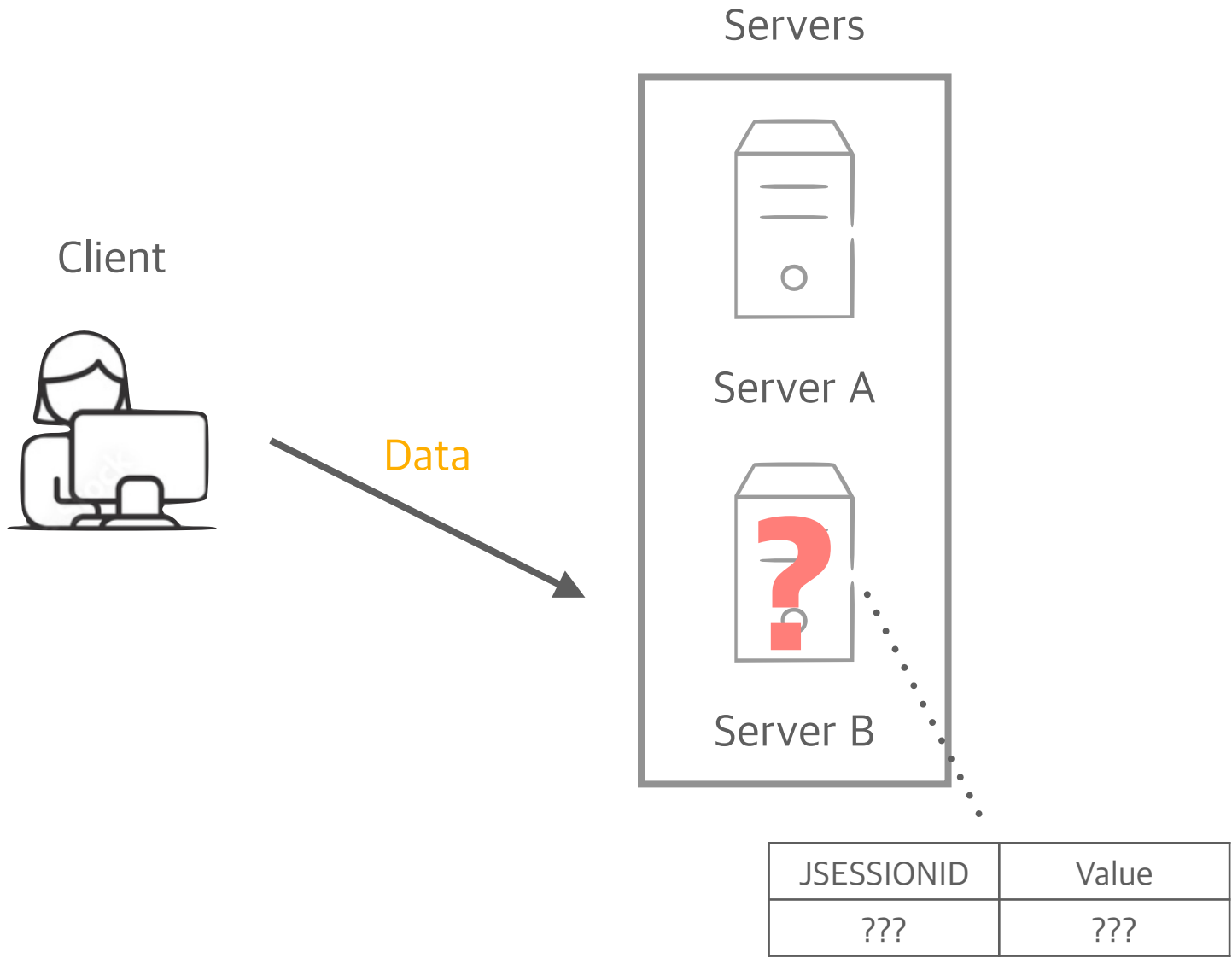
로그인 세션 같은, 서버가 꺼져도 상관없는 임시 데이터를 주로 사용한다.

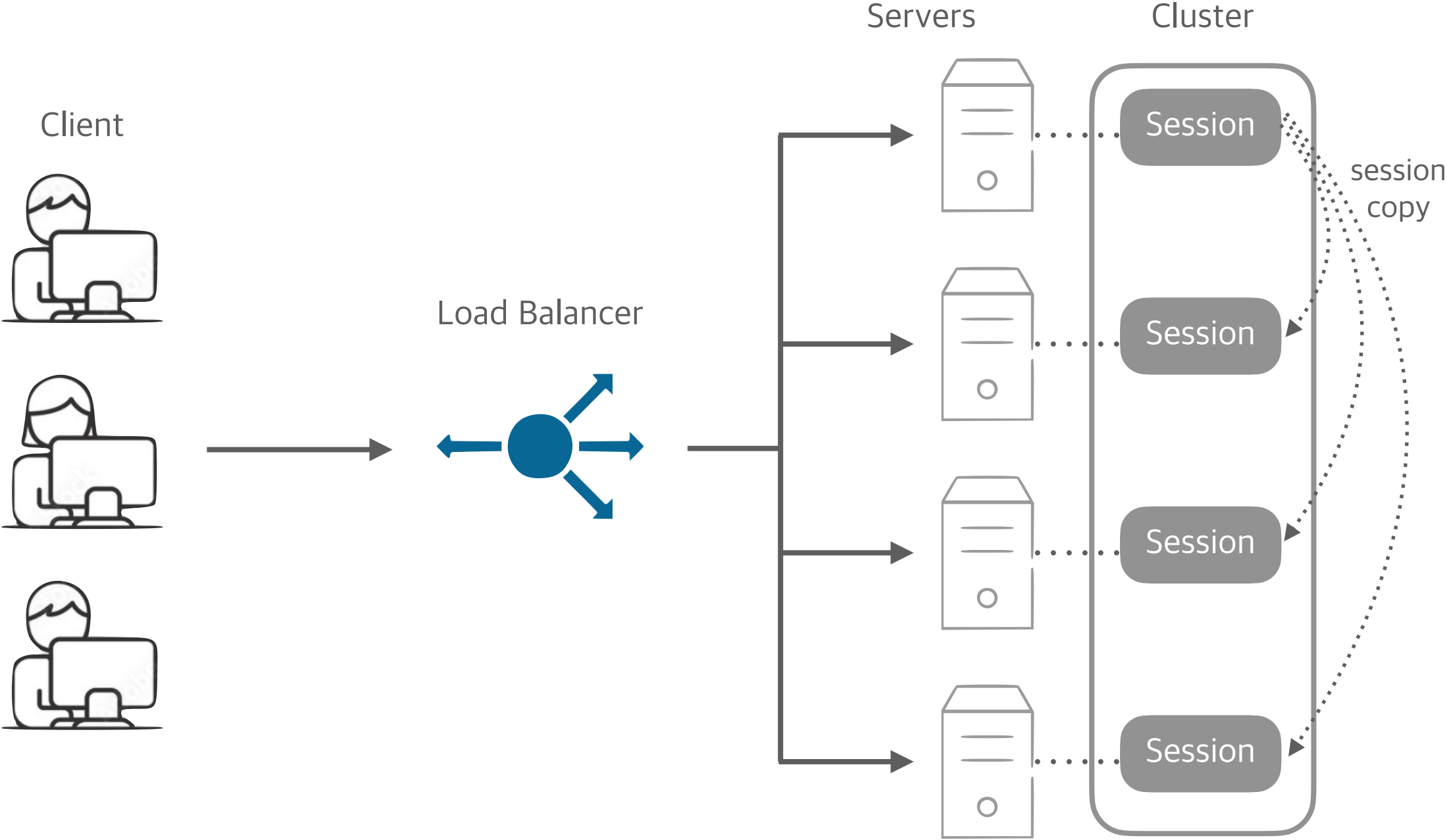
Thank You

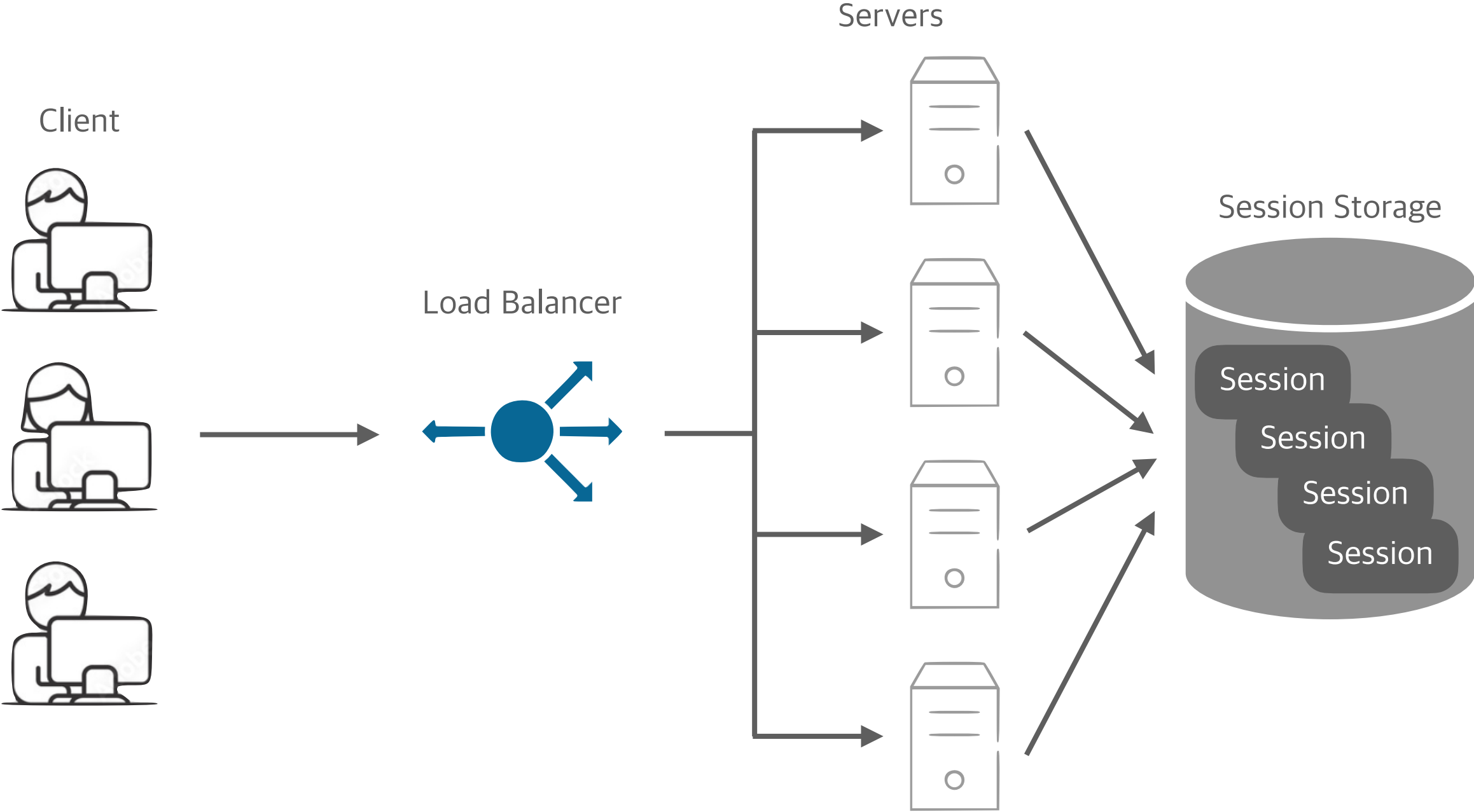


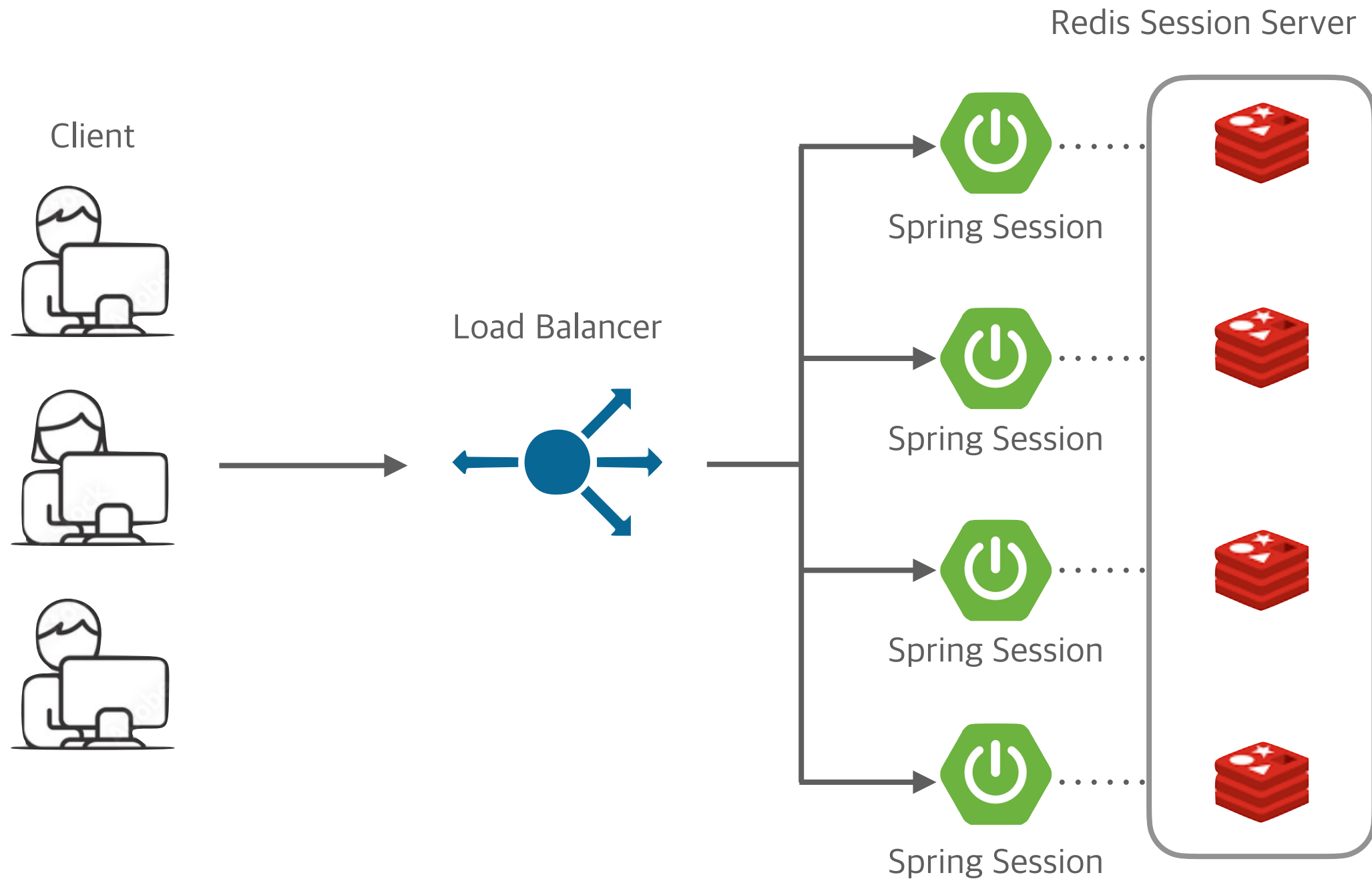


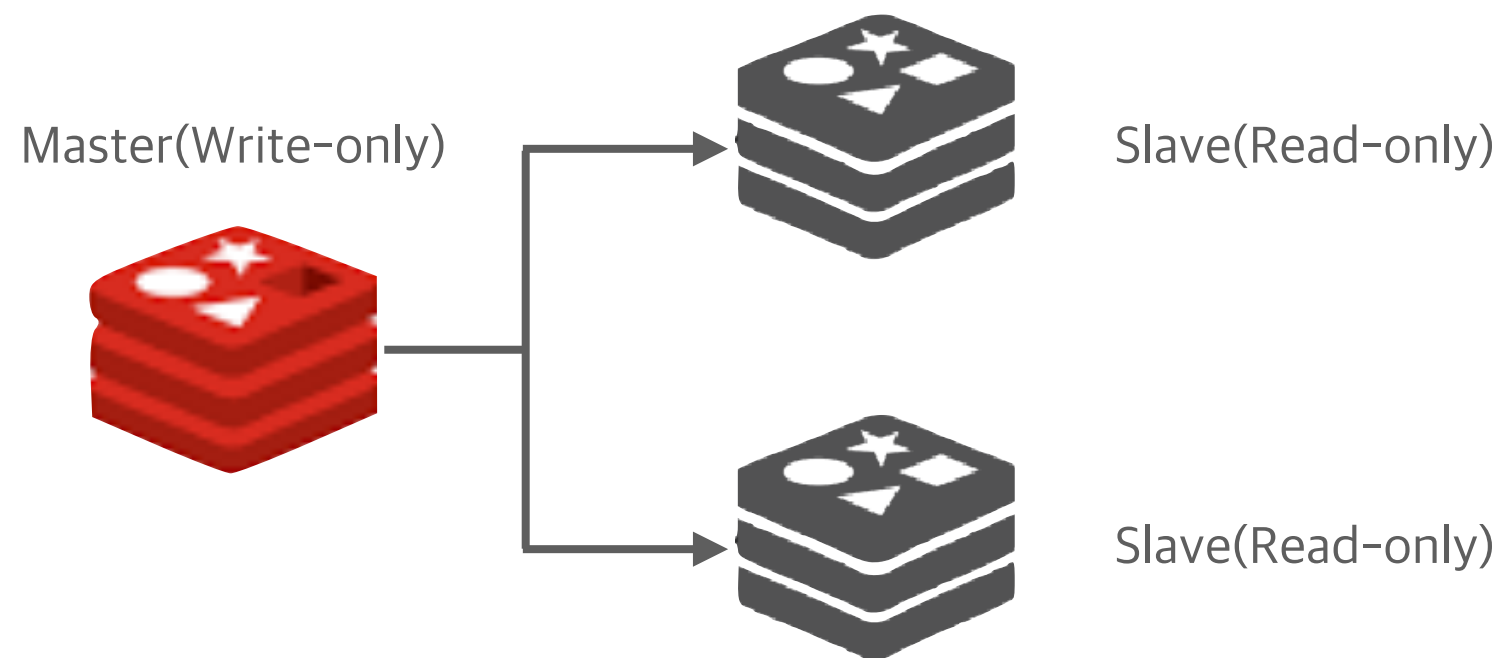


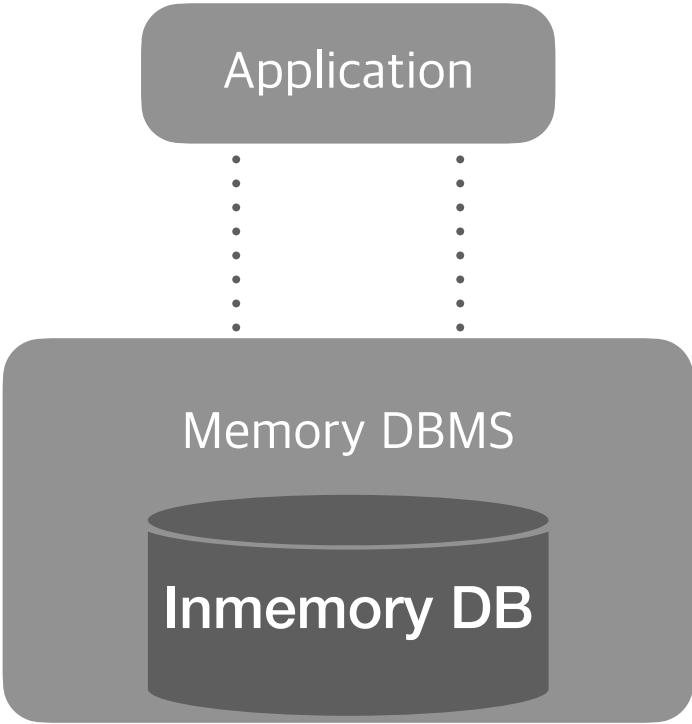




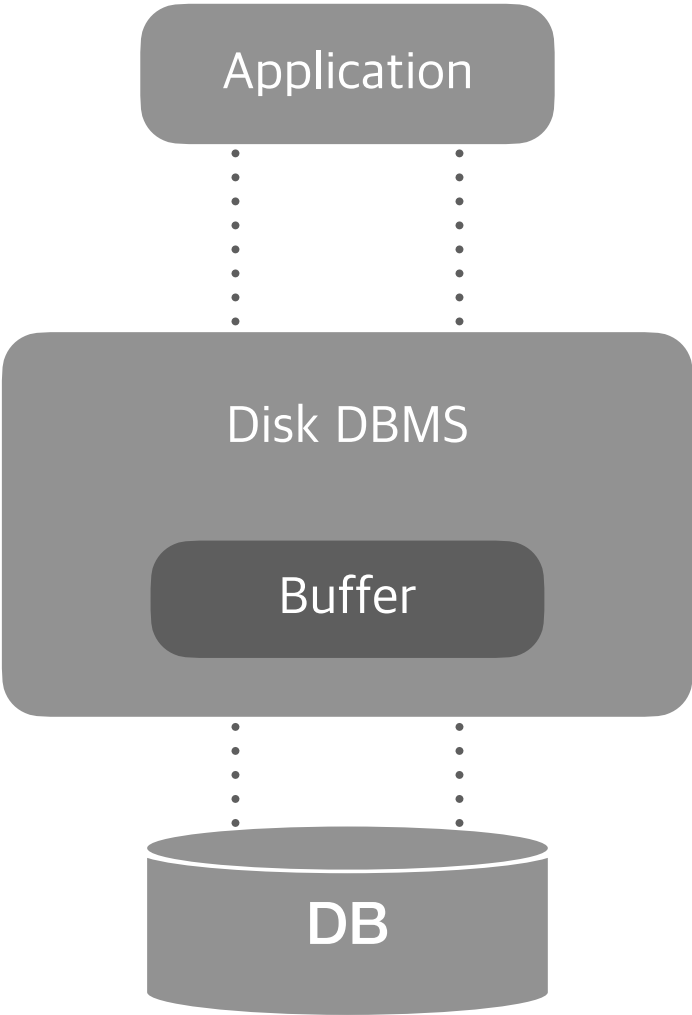








Inmemory DBMS



Disk DBMS

별첨

<https://gyoogle.dev/blog/computer-science/network/Load%20Balancing.html>

<https://smjeon.dev/web/sticky-session/>

<https://chagokx2.tistory.com/94>

<https://junshock5.tistory.com/84>