



대규모 시스템 첫걸음

최윤식



발표 대상

01 대규모 시스템에 관심이 있는 주니어 개발자

02 대규모 시스템을 운영하는 개발자

03 취업 준비 중인 개발자



목차

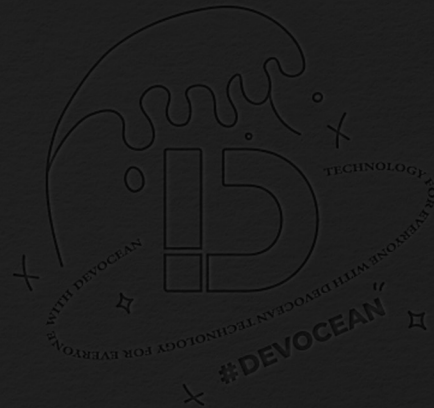
01 Server

02 Database

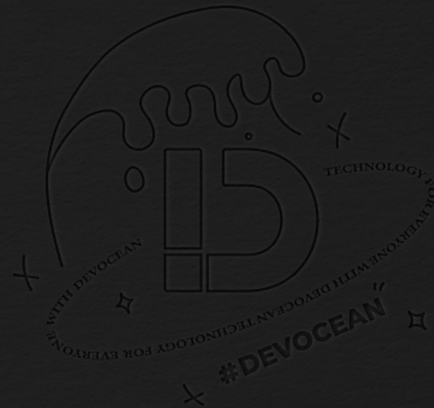
03 Key-value Store

04 Consistent Hashing

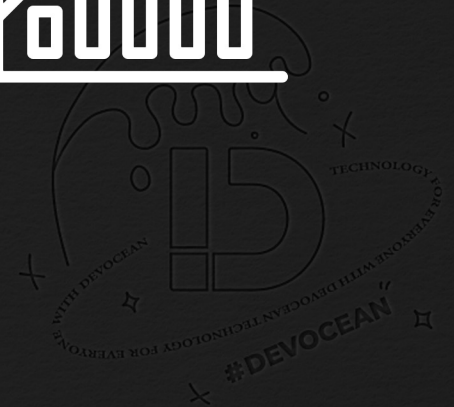
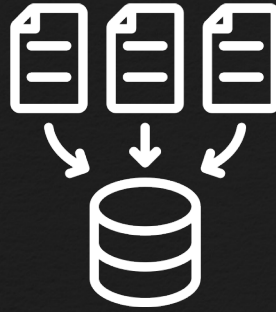
05 GUID



대규모 시스템



00 Intro



성능

고가용성

확장성



목차

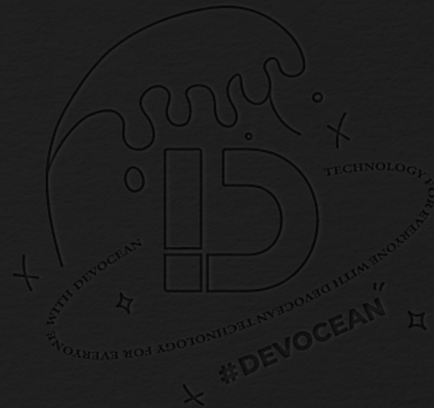
01 Server

02 Database

03 Key-value Store

04 Consistent Hashing

05 GUID



목차

01 Server

02 Database

03 Key-value Store

04 Consistent Hashing

05 GUID



01 Server



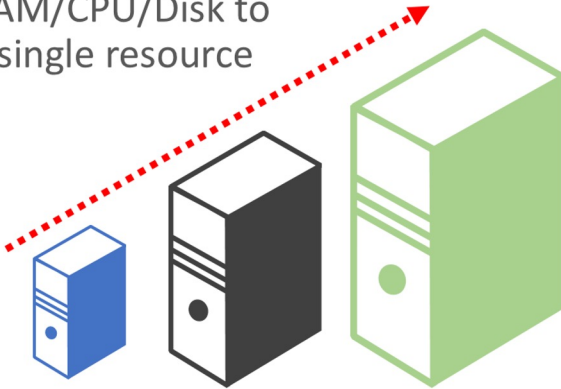
01 Server



01 Server

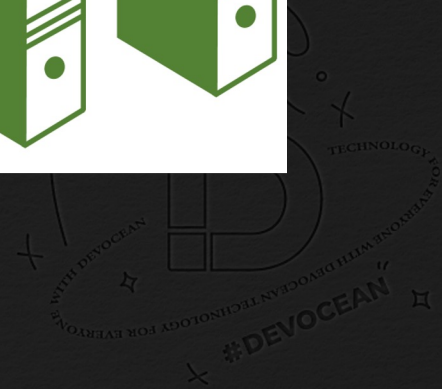
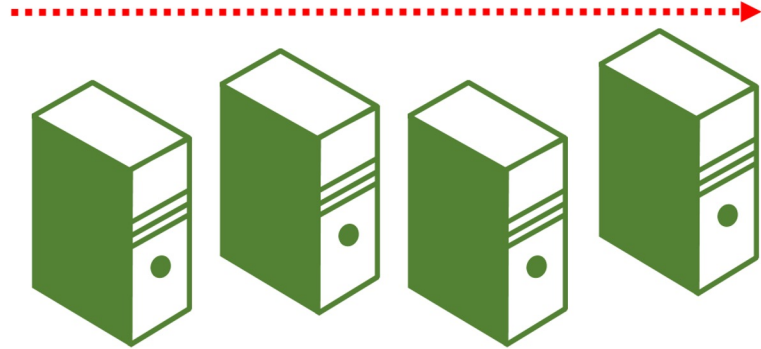
Scale Up (vertical scaling)

Increase capacity by adding RAM/CPU/Disk to a single resource



Scale Out (horizontal scaling)

Increase capacity by adding resources



01 Server



01 Server



01 Server



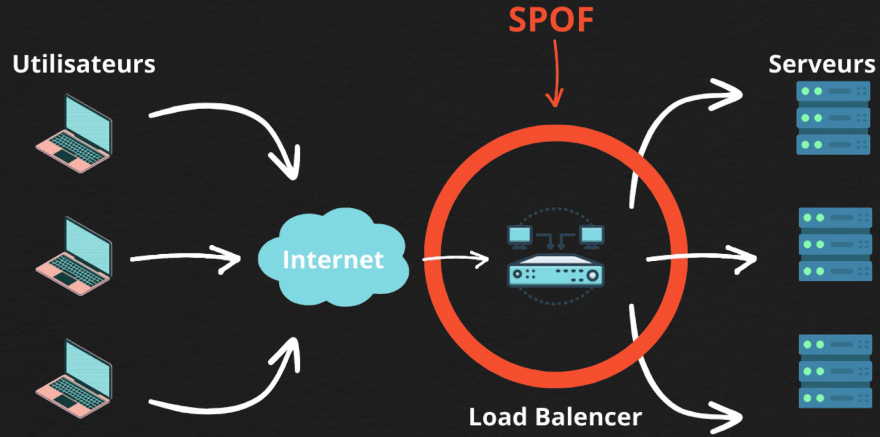
01 Server

SPOF

(Single point of failure)



01 Server

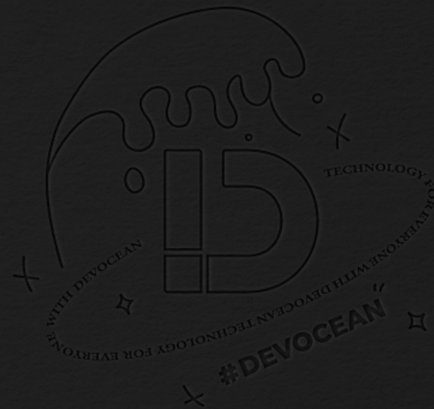


01 Server



01 Server

SPOF !



01 Server



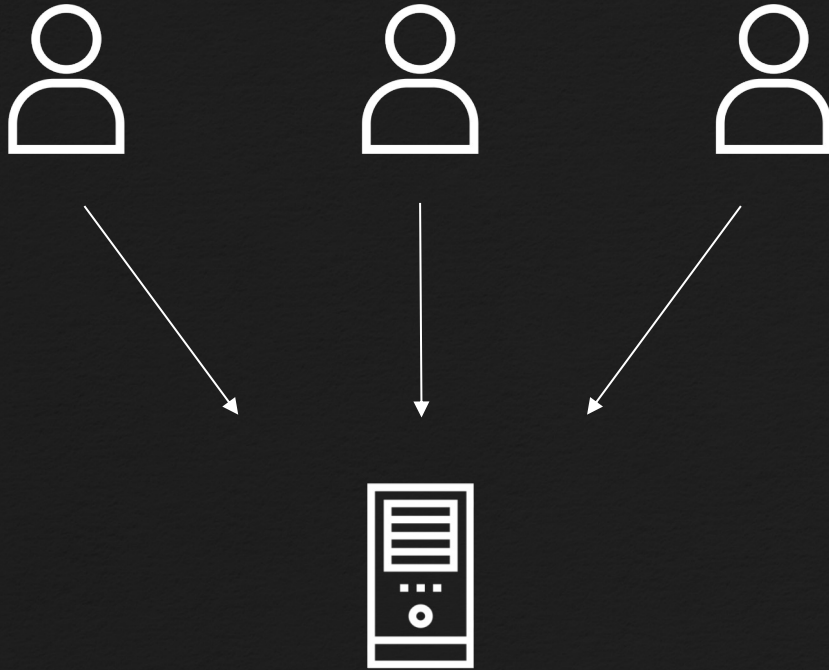
01 Server



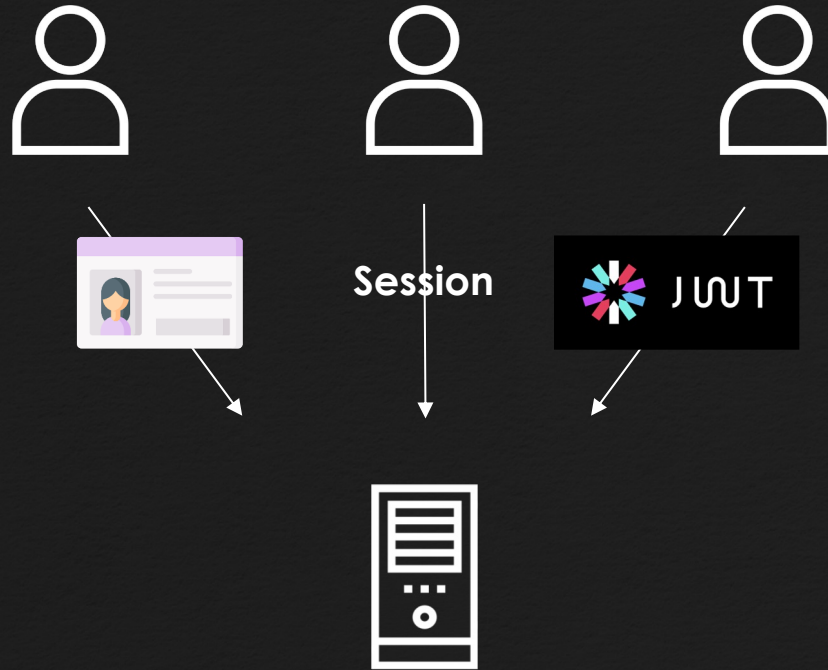
01 Server



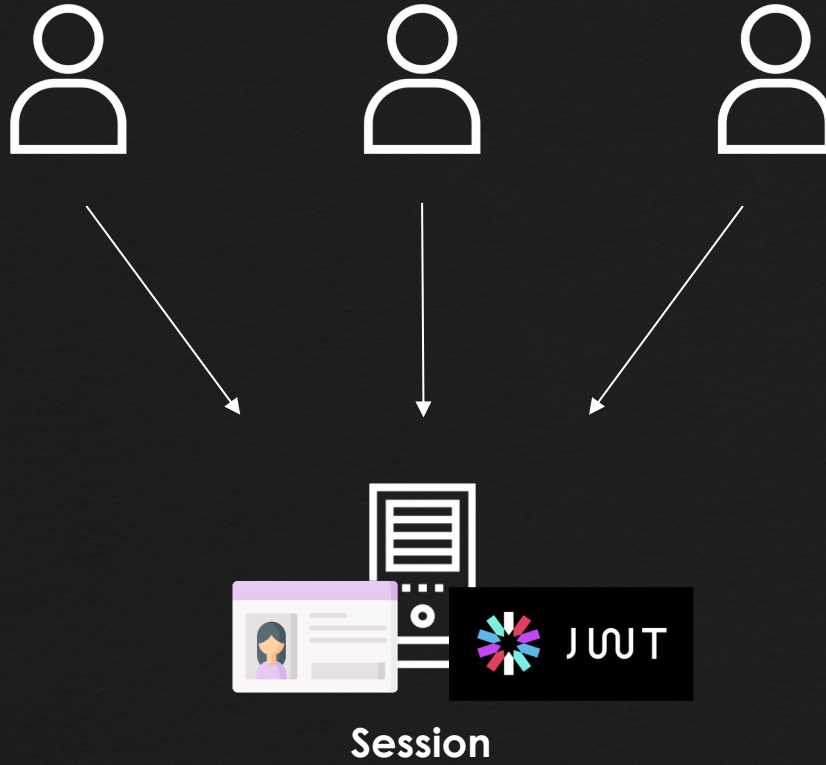
01 Server



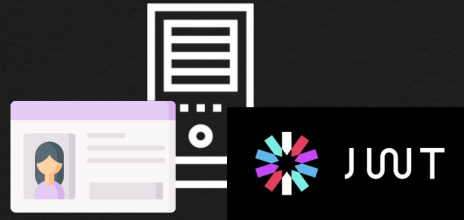
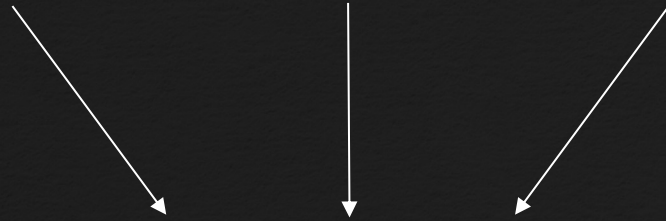
01 Server



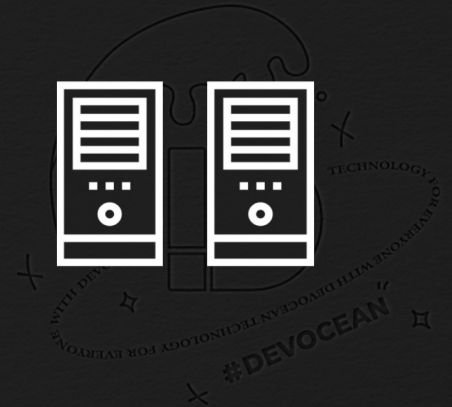
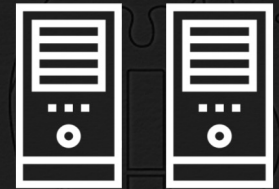
01 Server



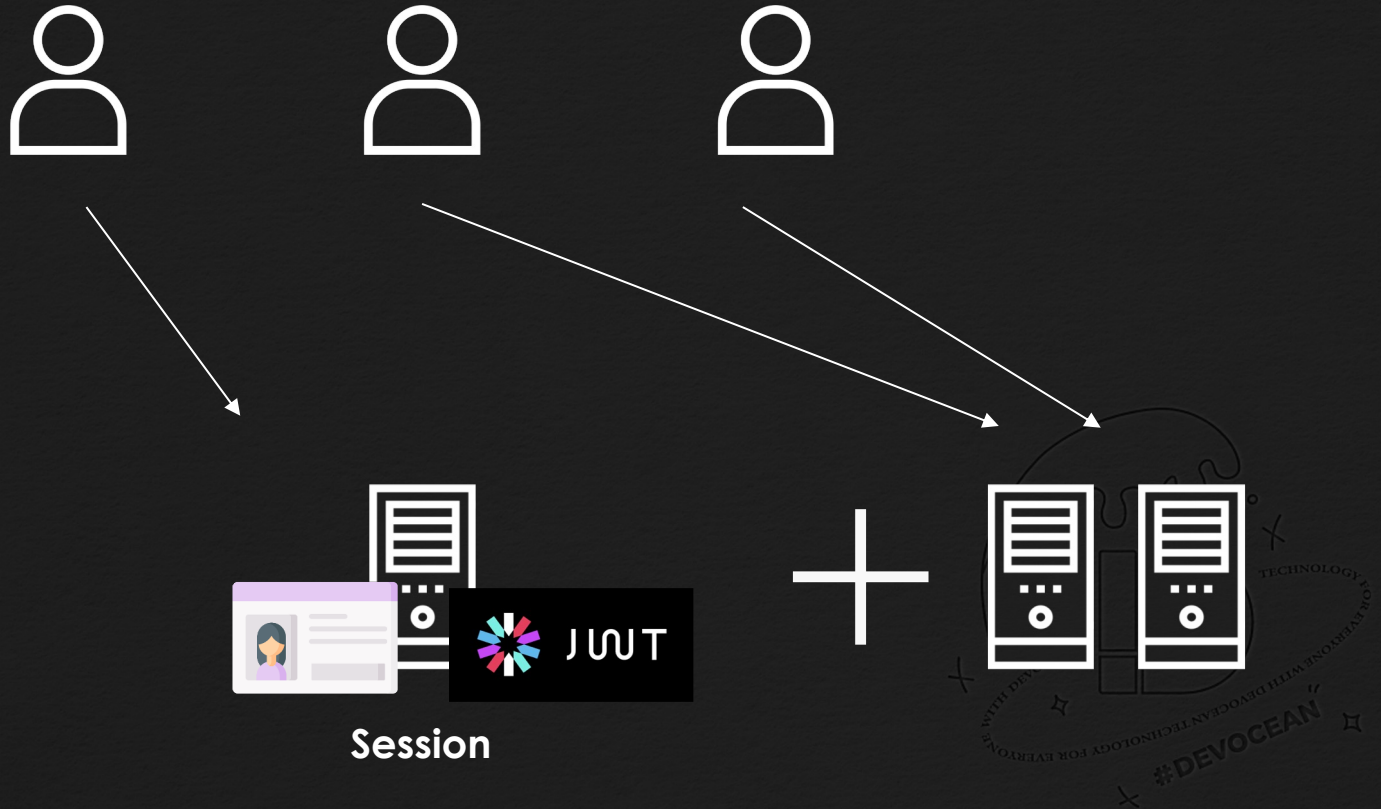
01 Server



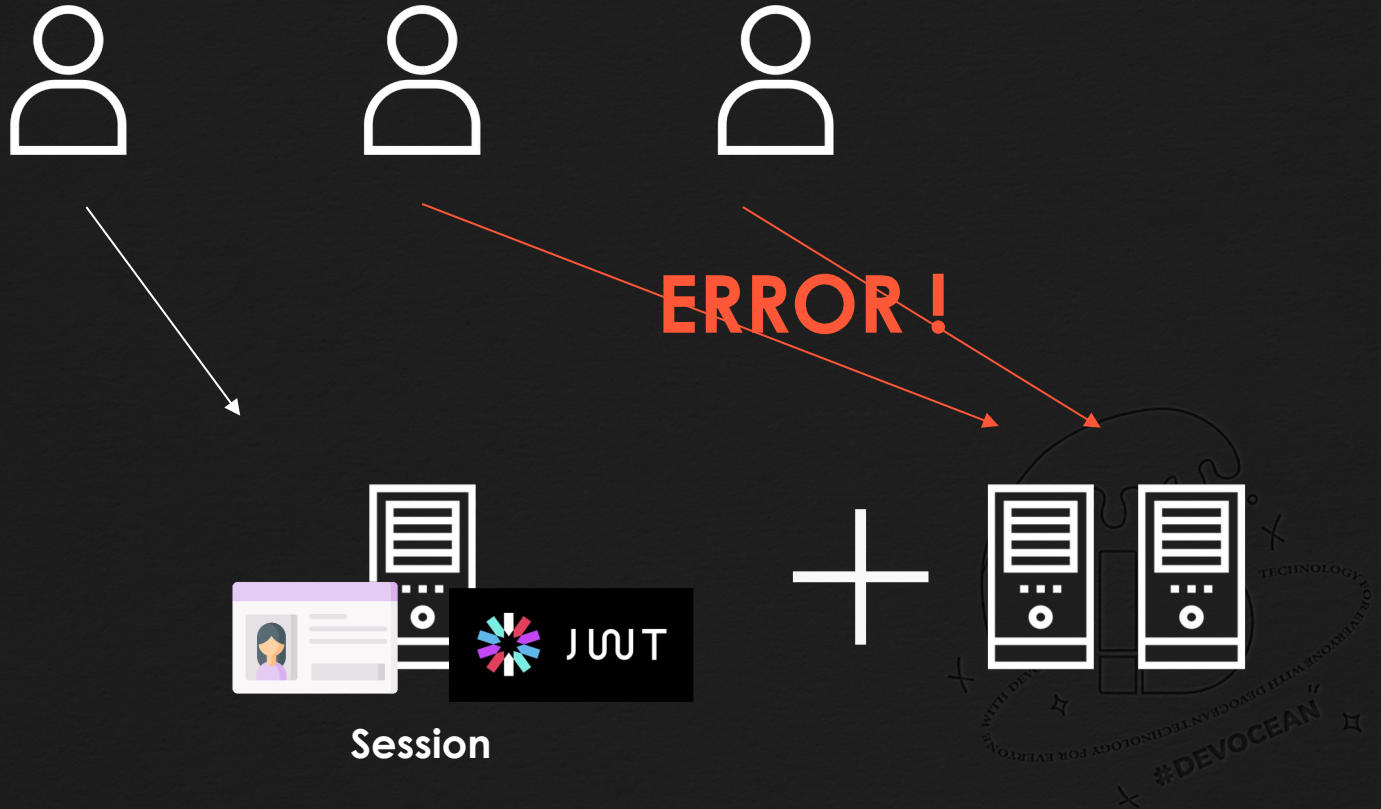
Session



01 Server



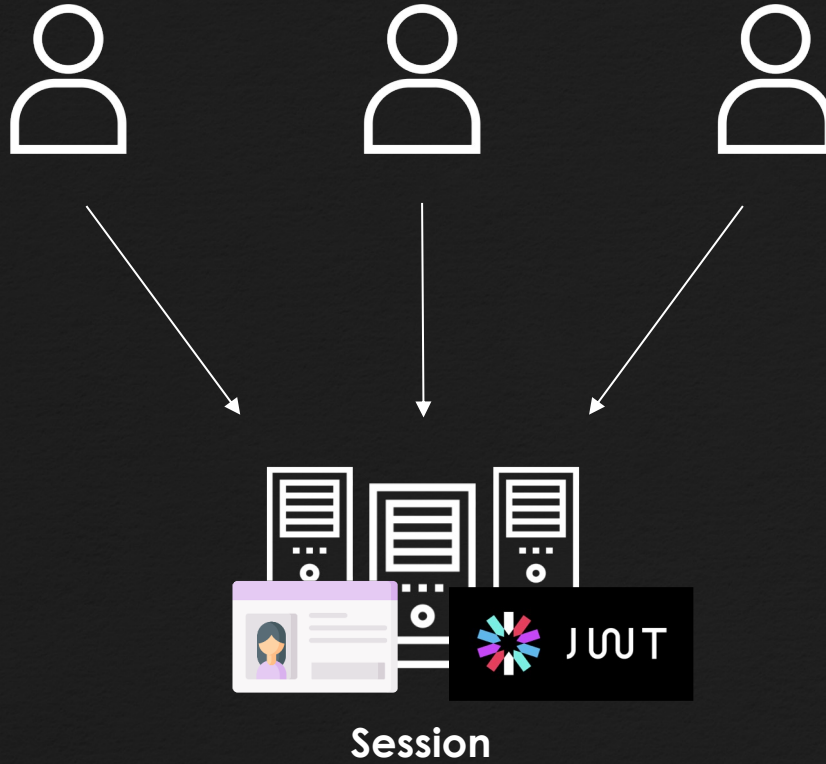
01 Server



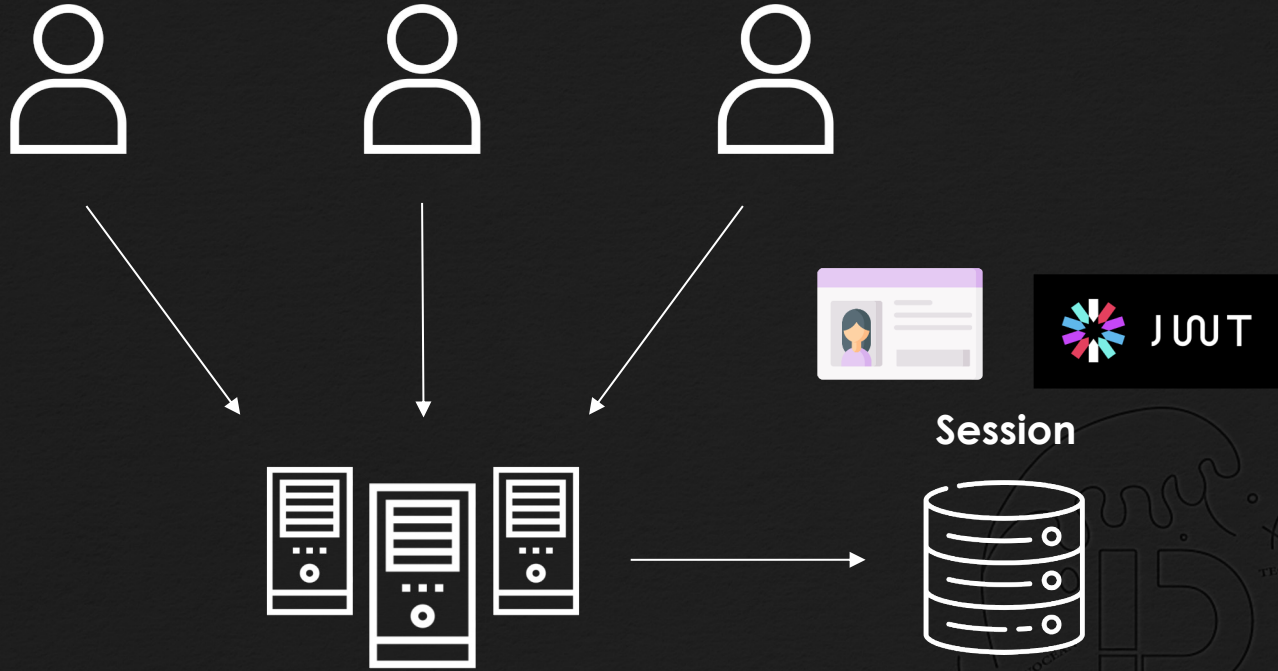
Stateless



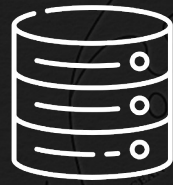
01 Server



01 Server



Session



목차

01 Server

02 Database

03 Key-value Store

04 Consistent Hashing

05 GUID



목차

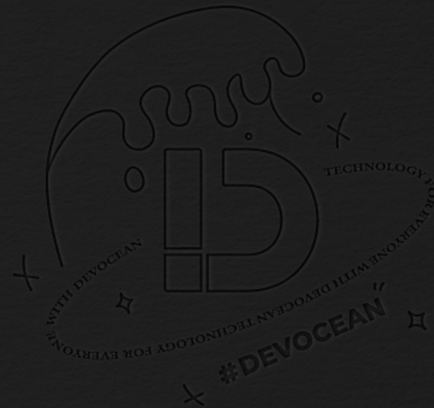
01 Server

02 Database

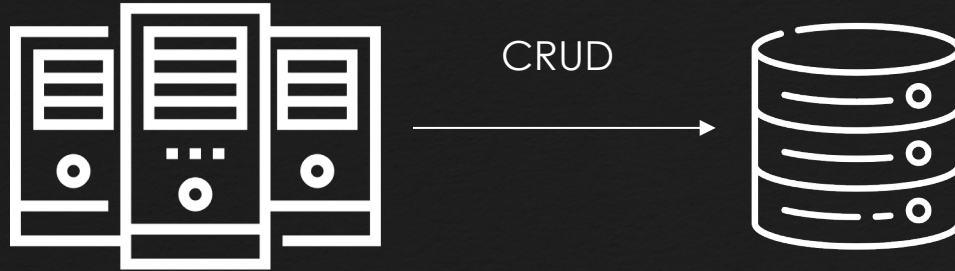
03 Key-value Store

04 Consistent Hashing

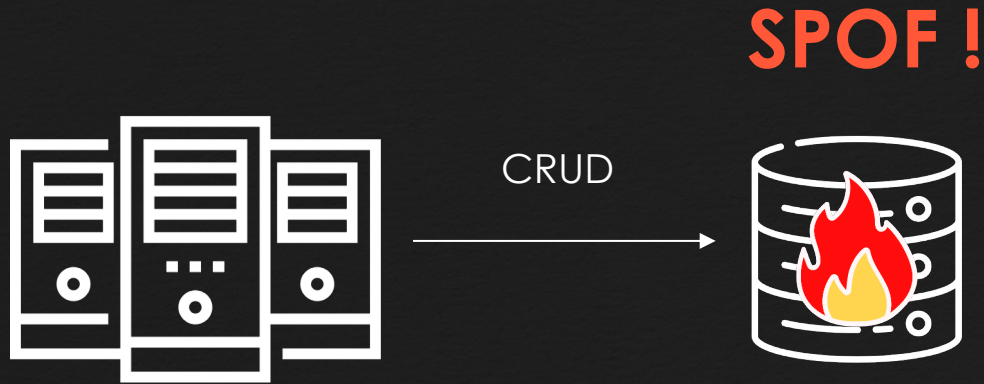
05 GUID



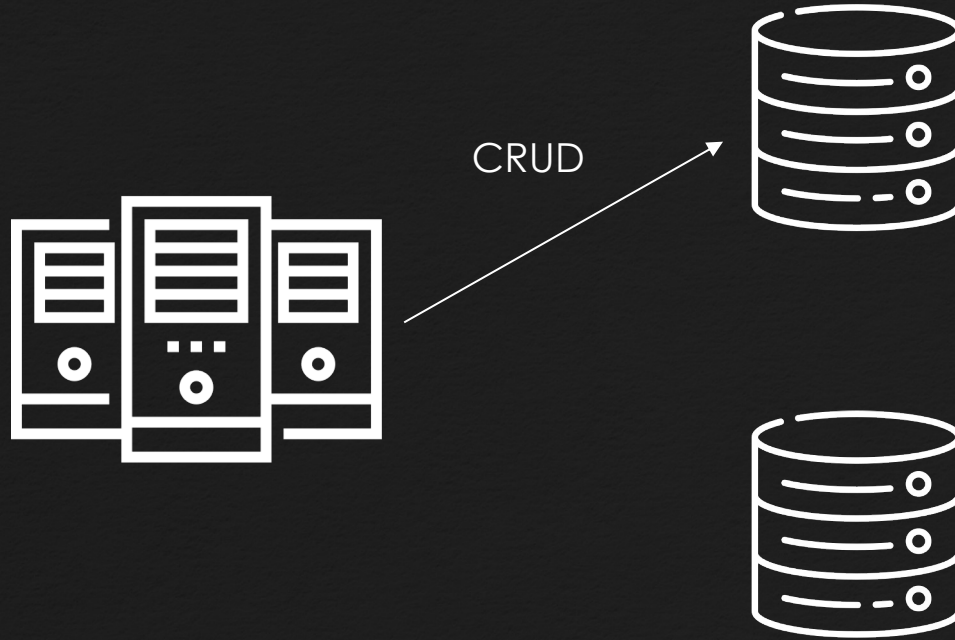
02 Database



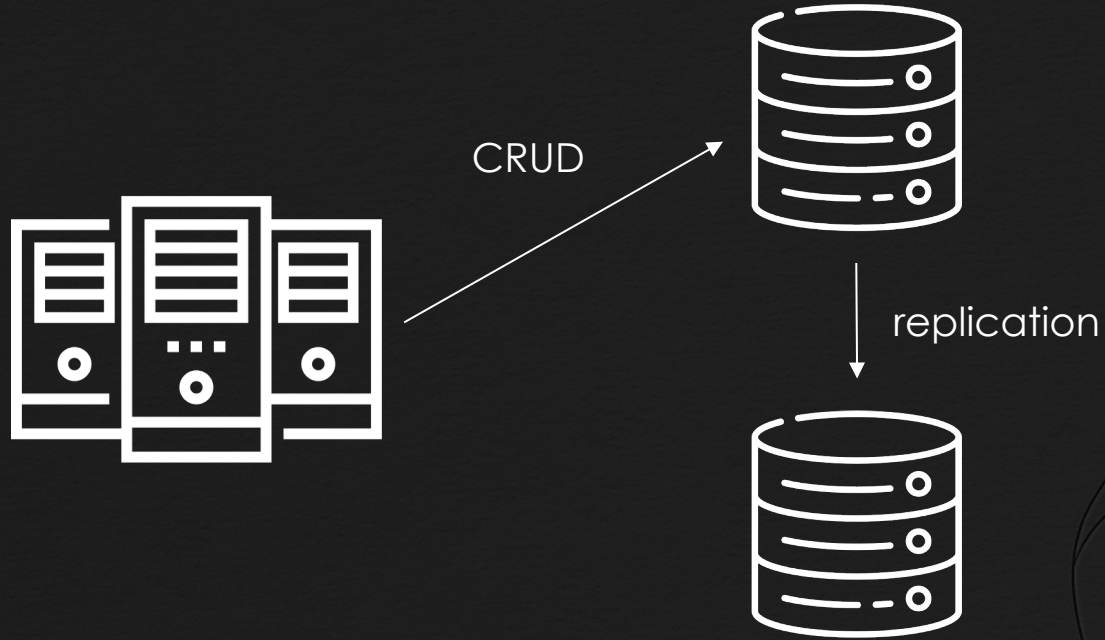
02 Database



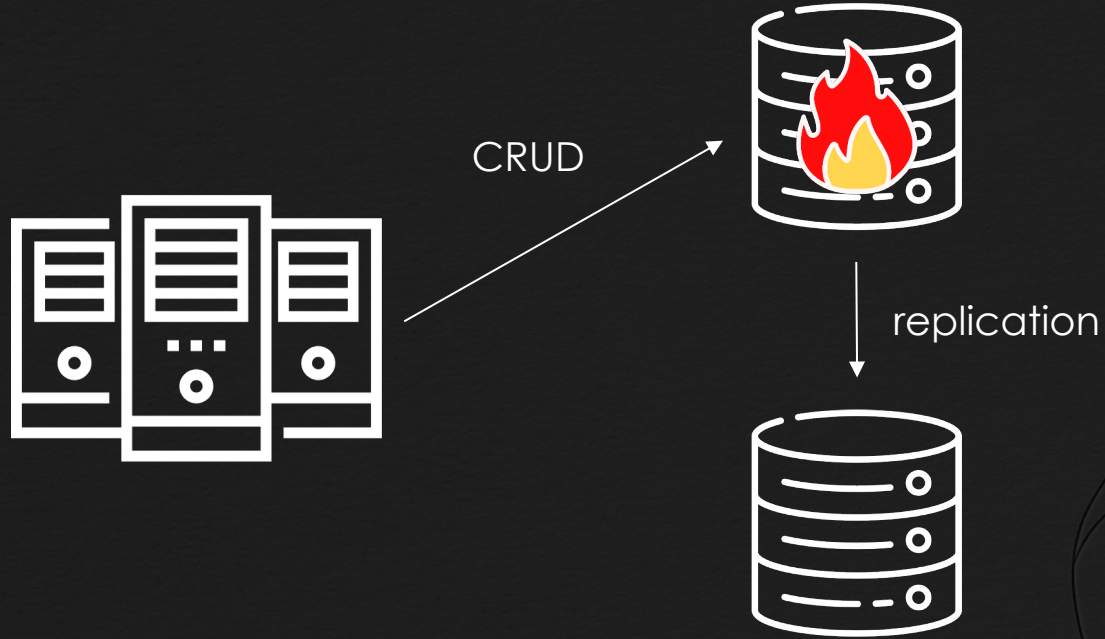
02 Database



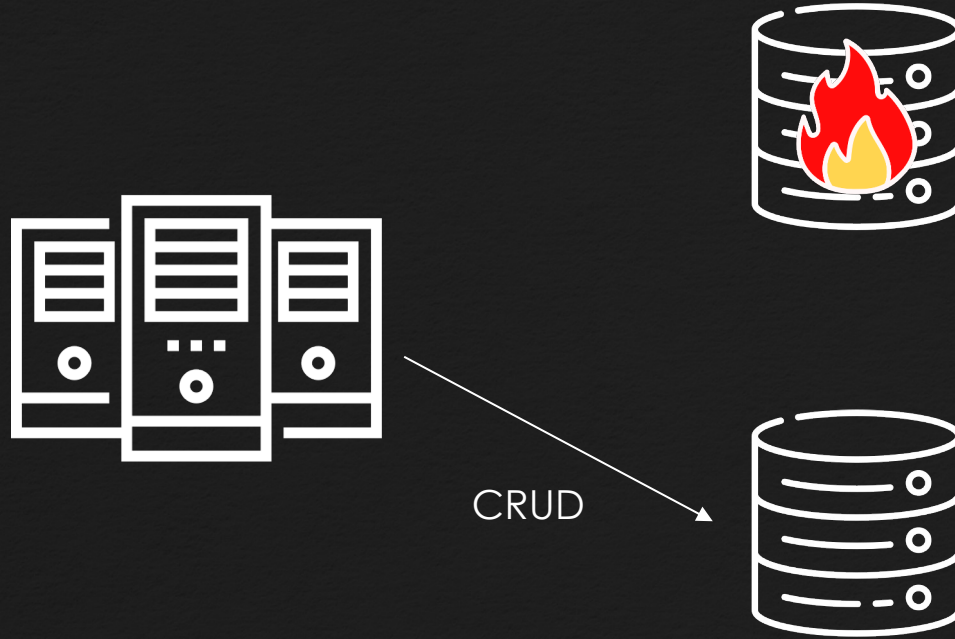
02 Database



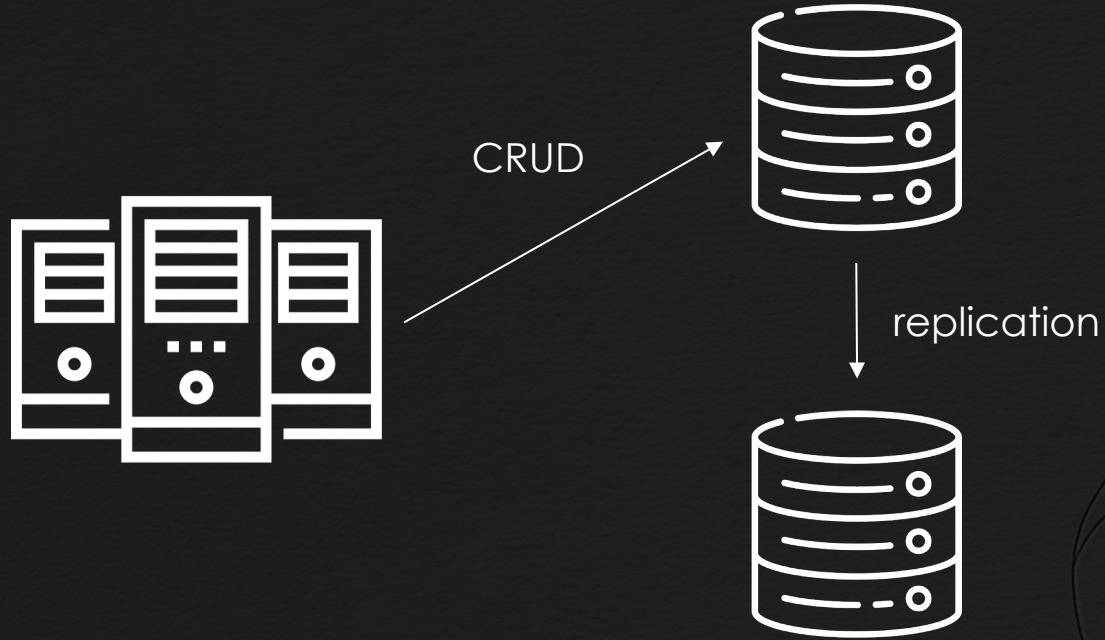
02 Database



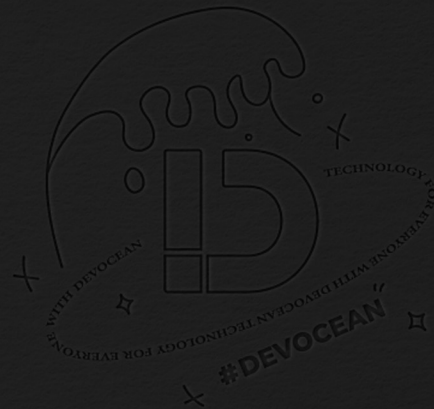
02 Database



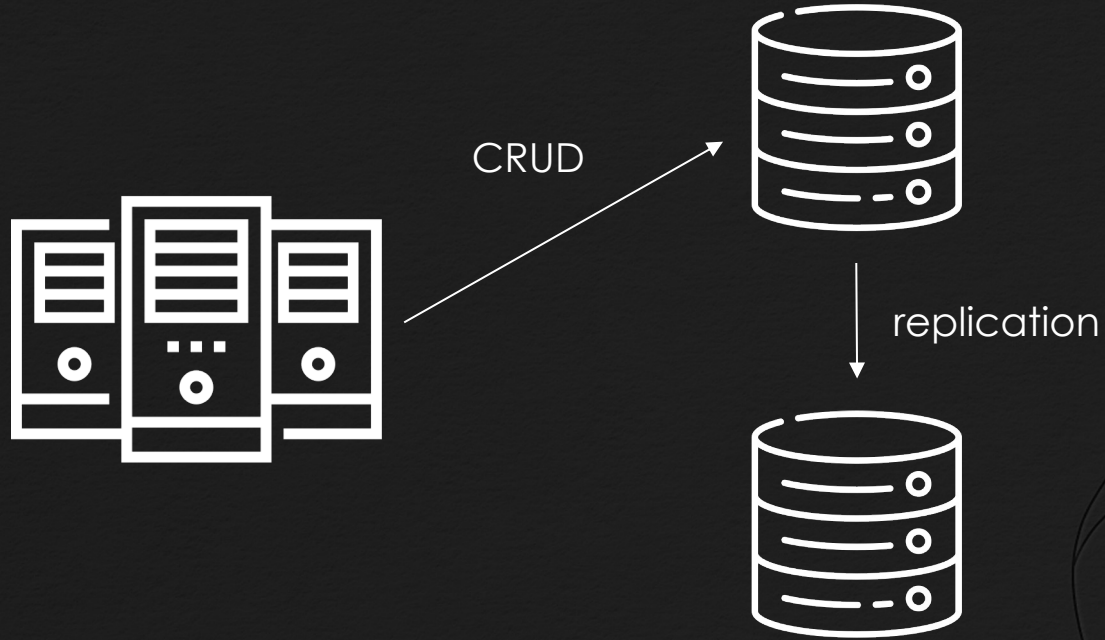
02 Database



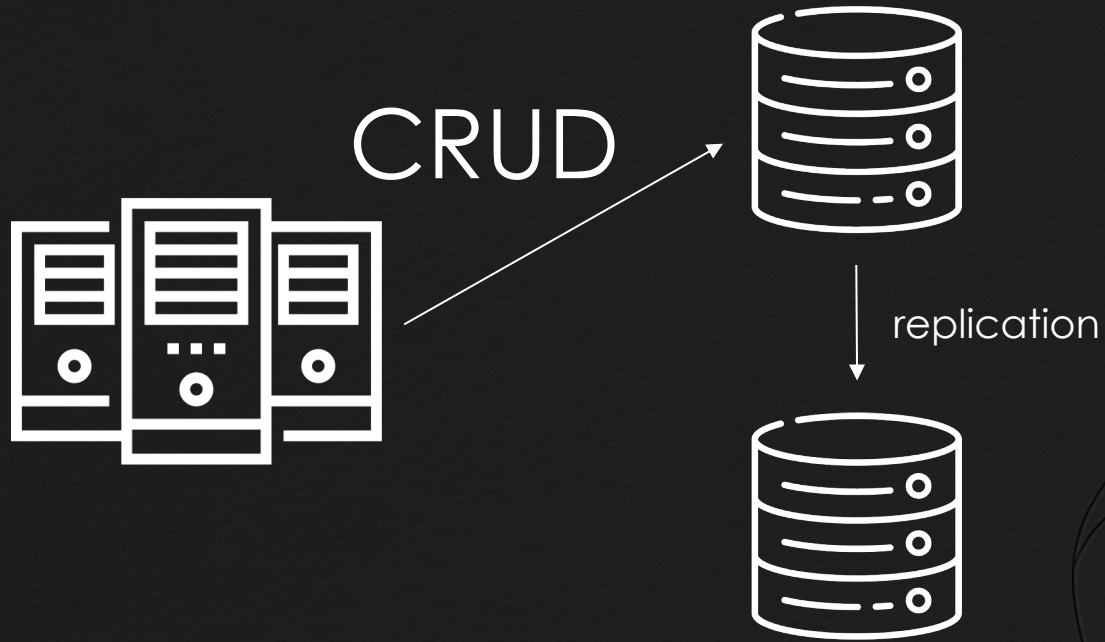
02 Database



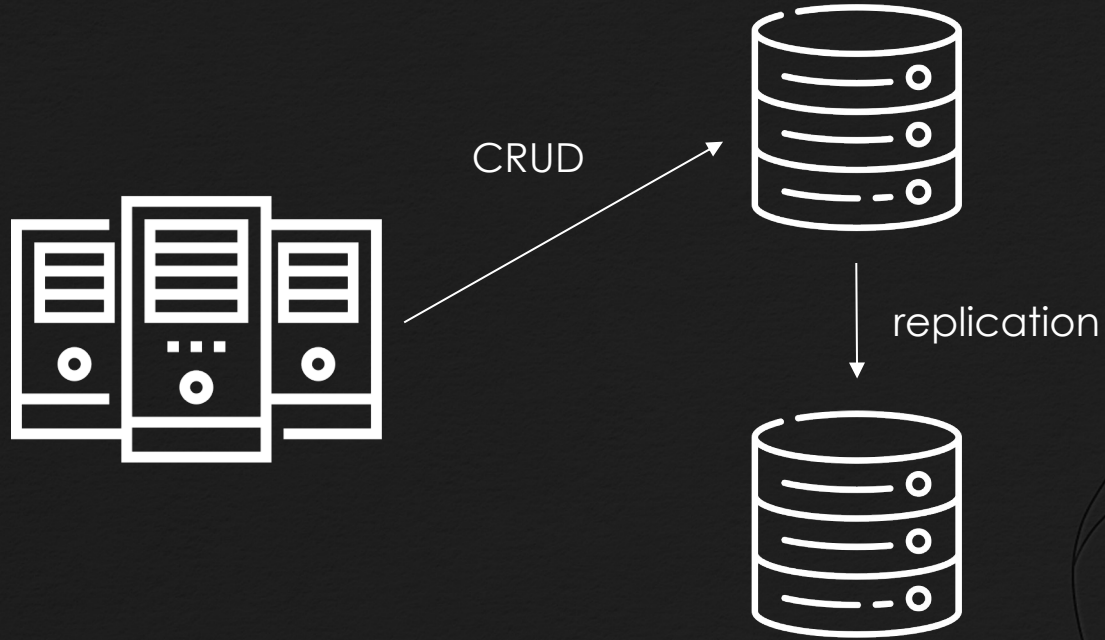
02 Database



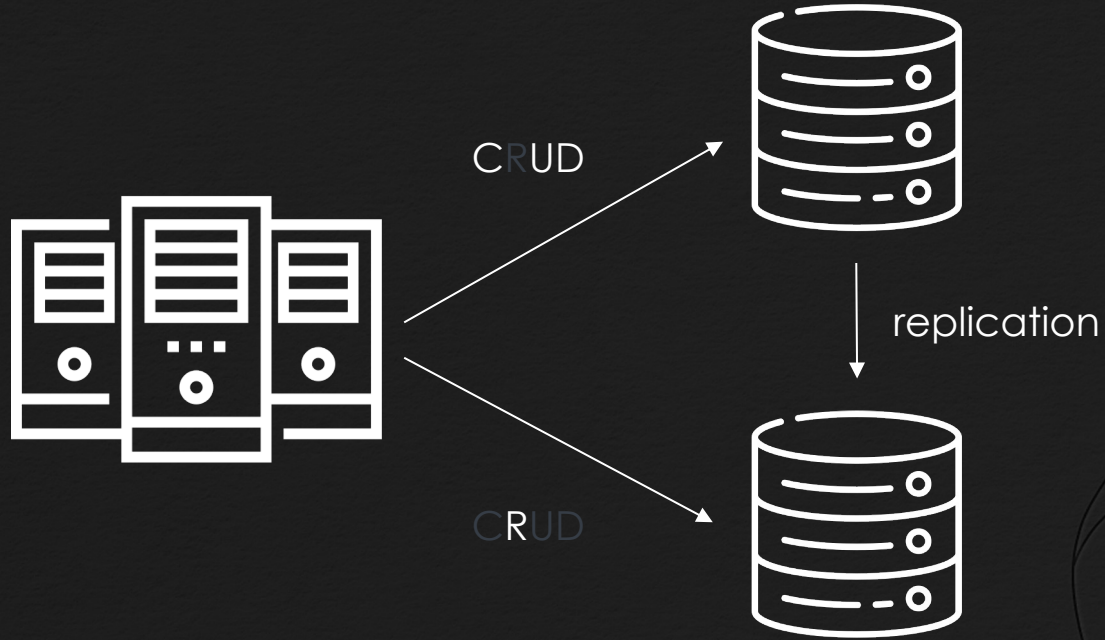
02 Database



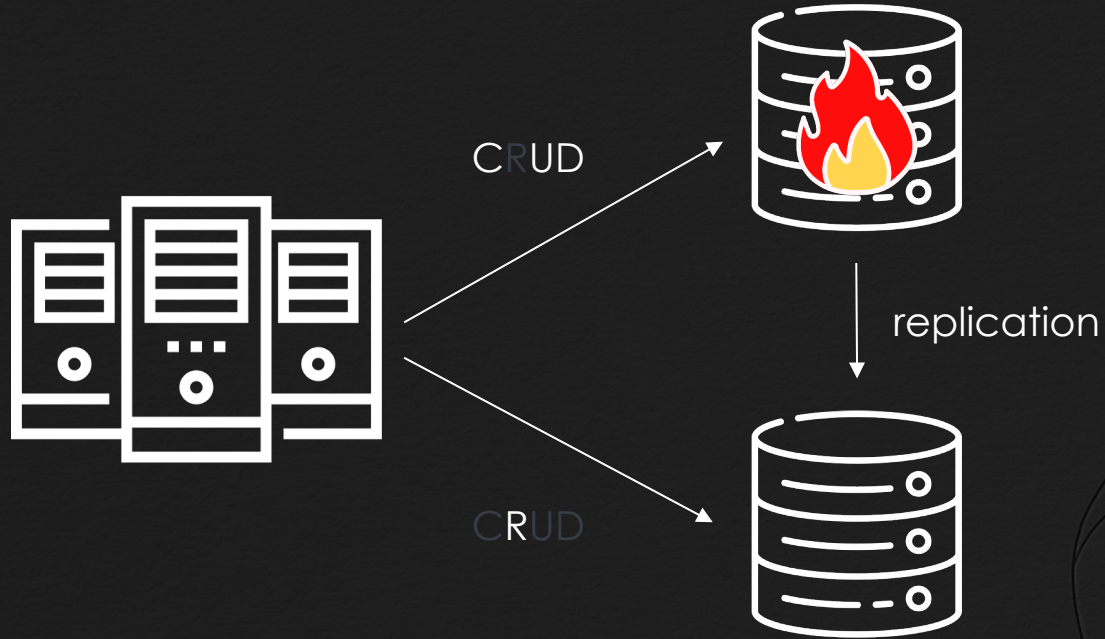
02 Database



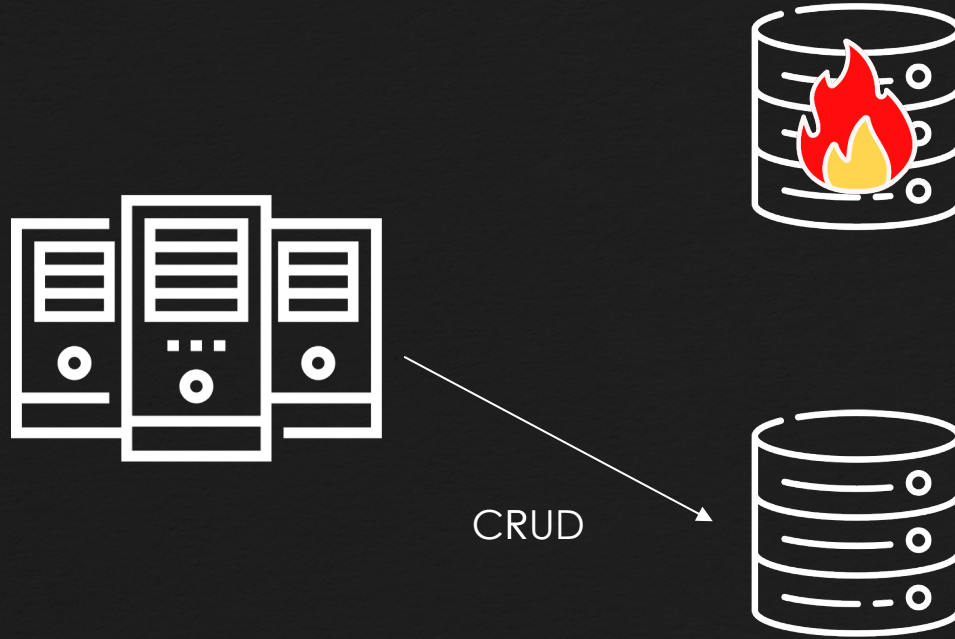
02 Database



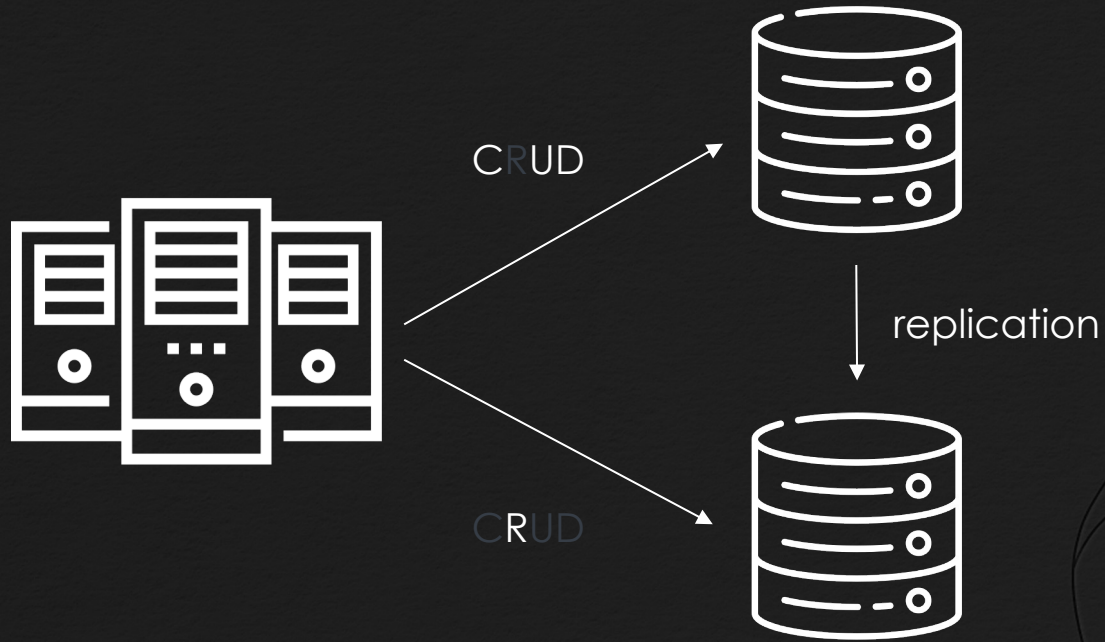
02 Database



02 Database



02 Database

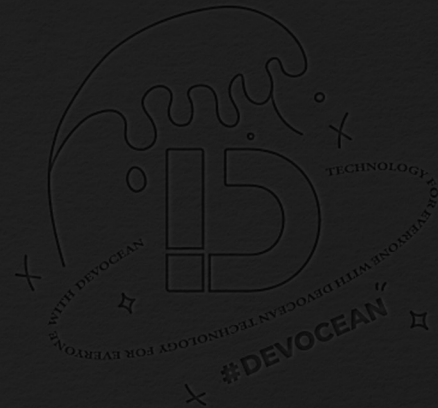


02 Database

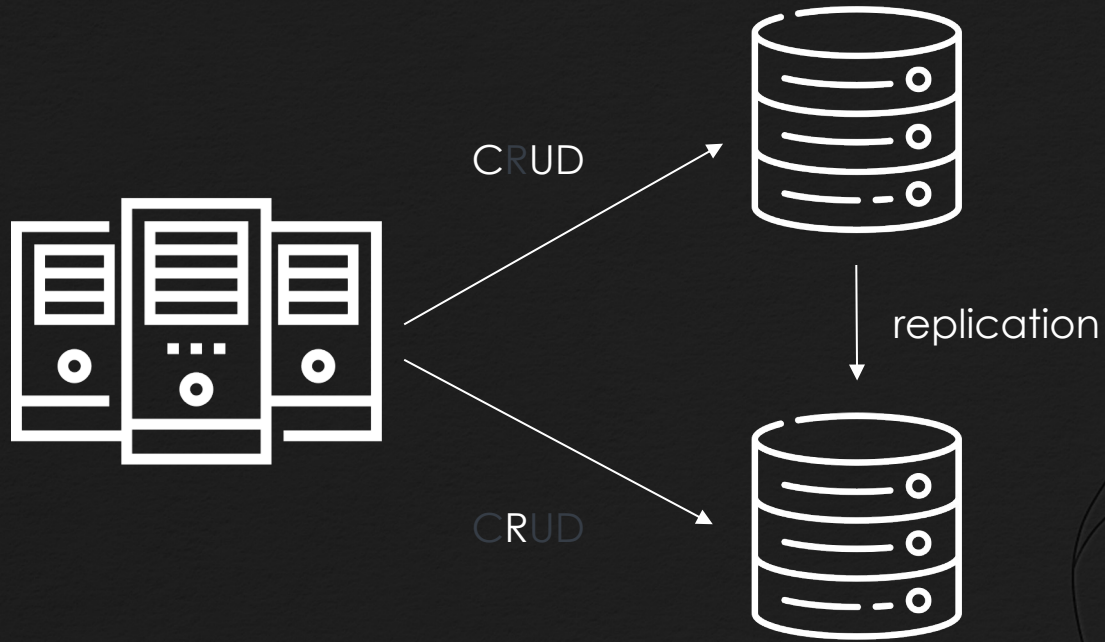


02 Database

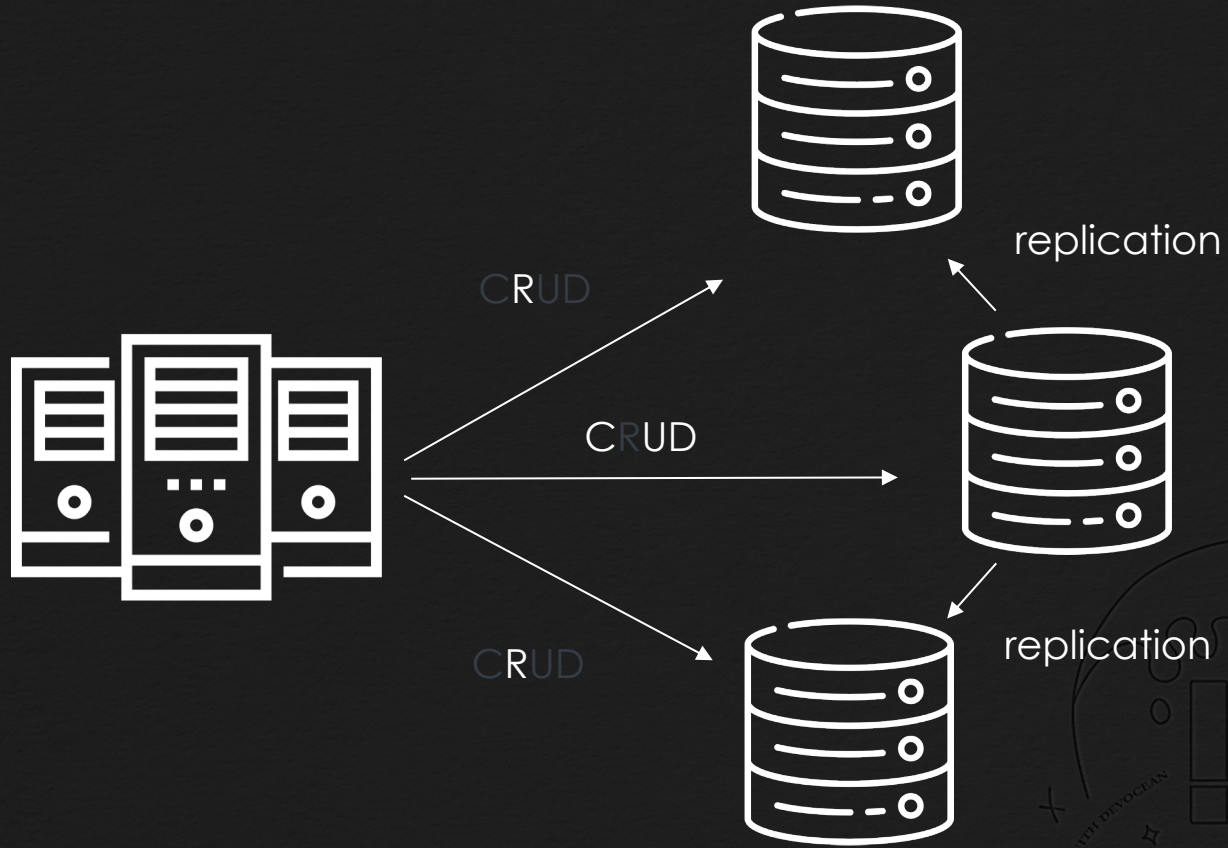
CRUD



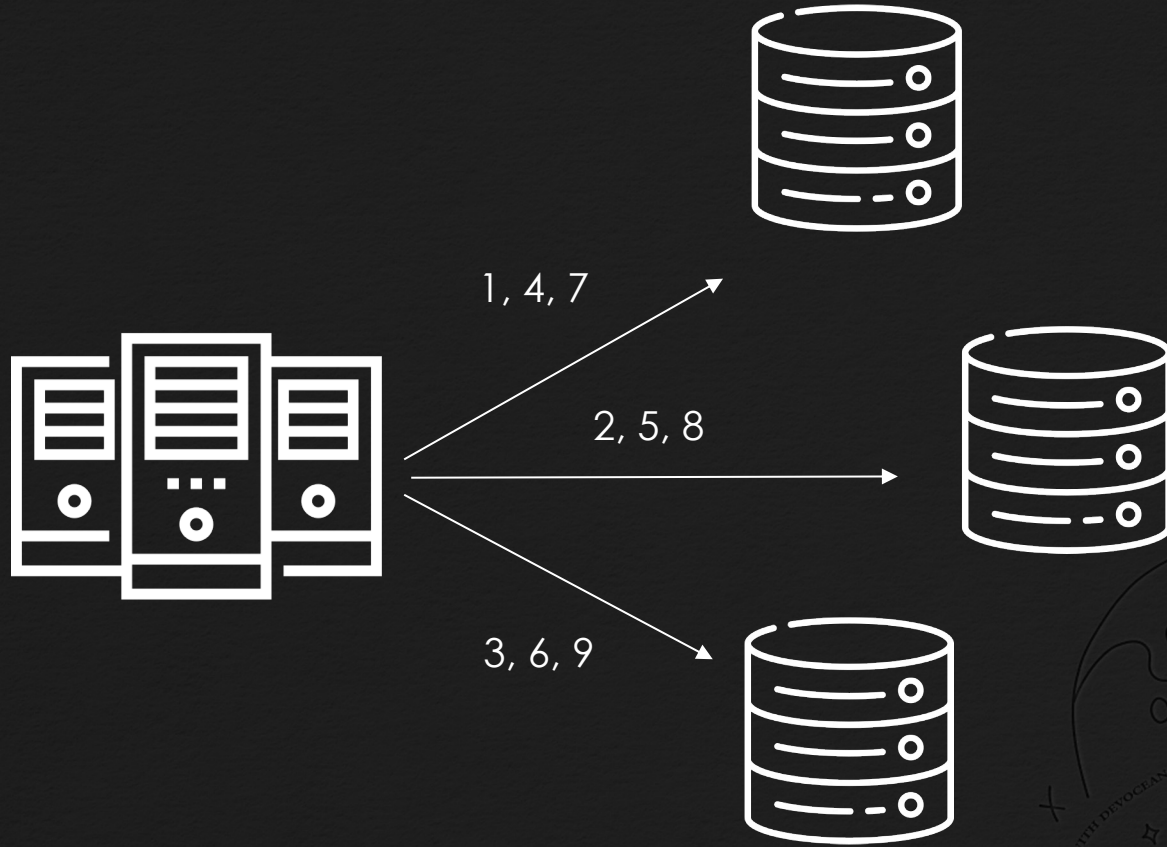
02 Database



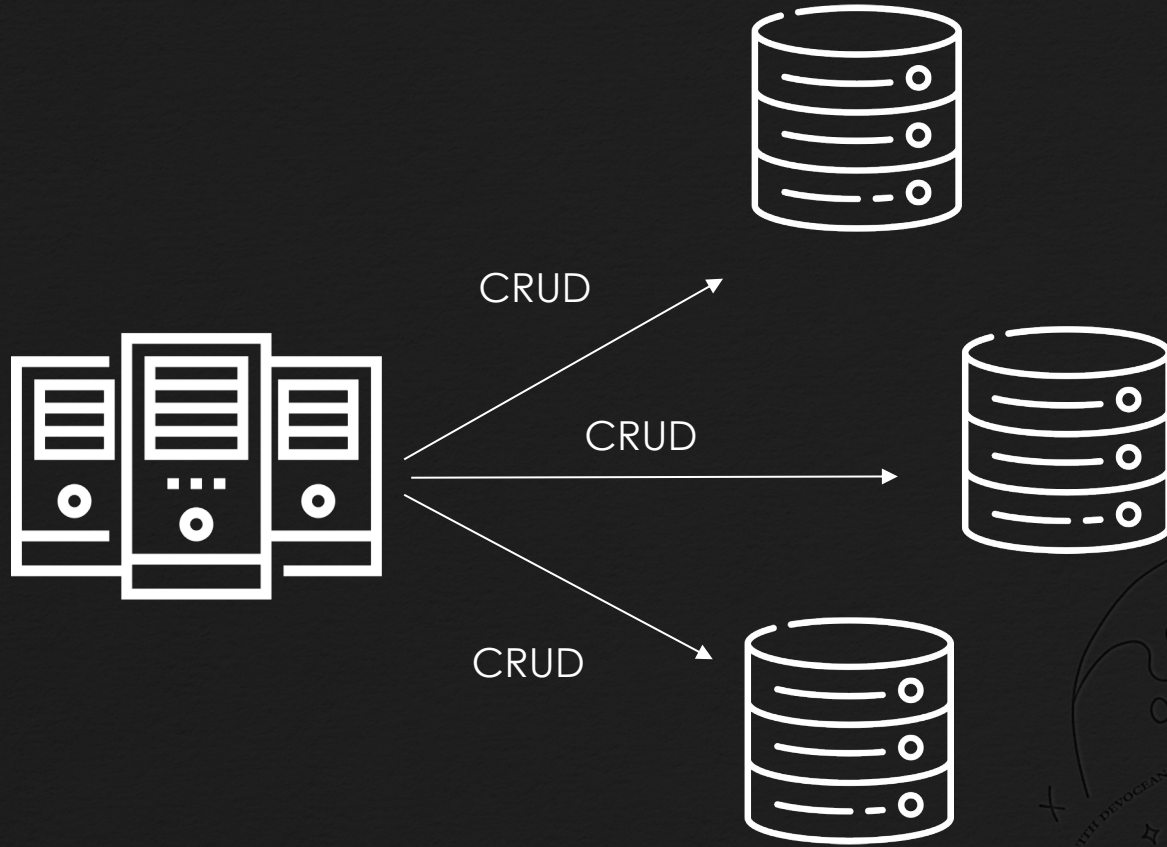
02 Database



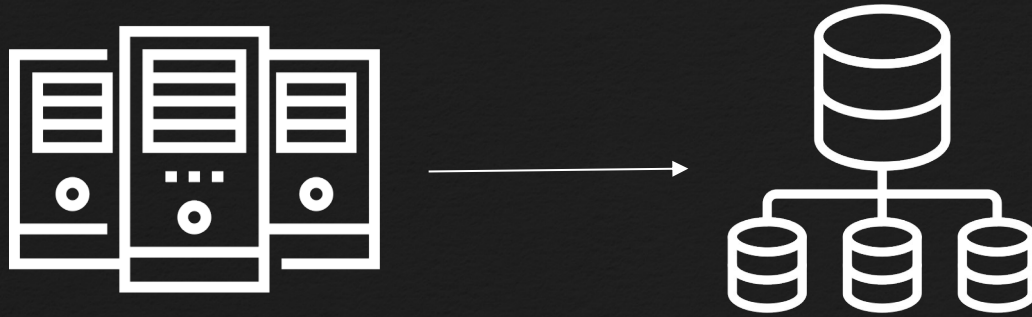
02 Database



02 Database



02 Database



목차

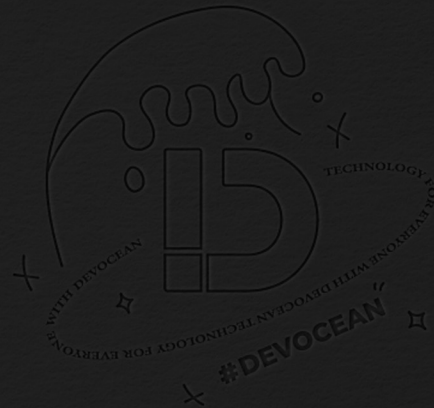
01 Server

02 Database

03 Key-value Store

04 Consistent Hashing

05 GUID



목차

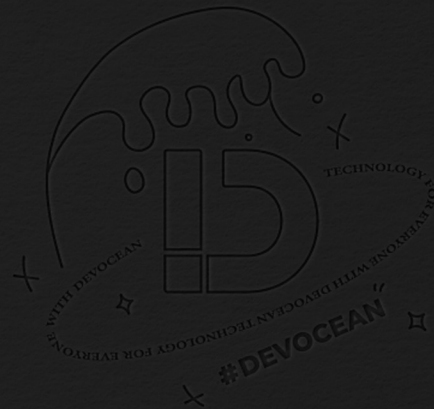
01 Server

02 Database

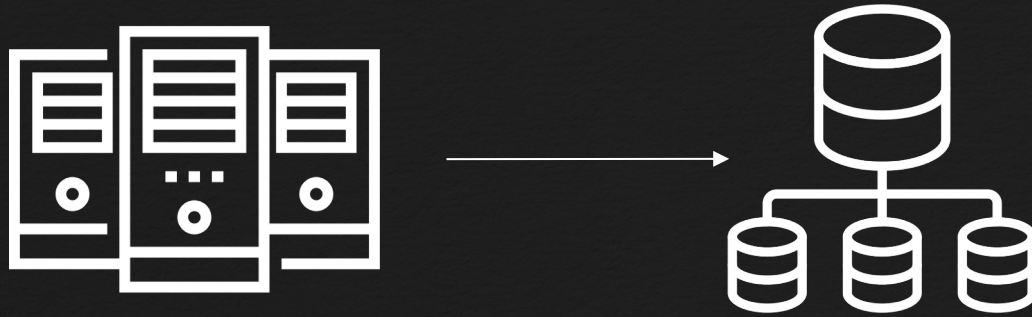
03 Key-value Store

04 Consistent Hashing

05 GUID



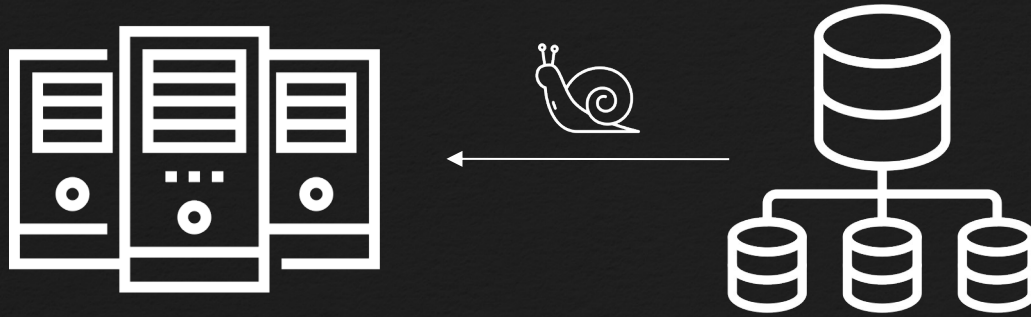
03 Key-value Store



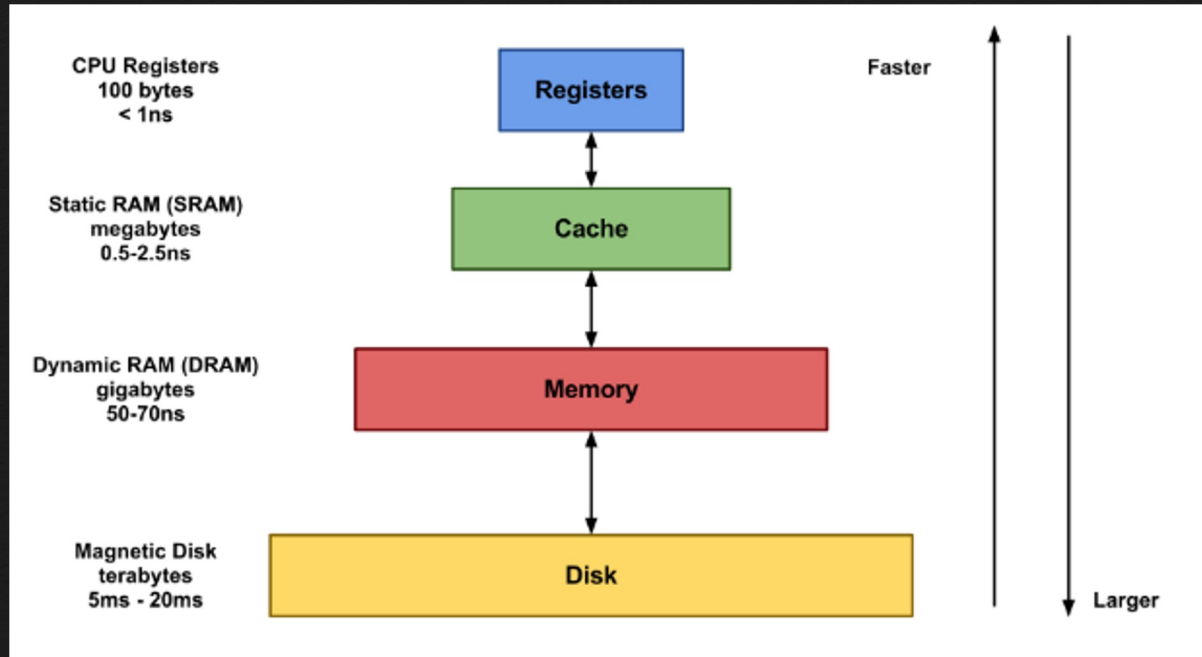
03 Key-value Store



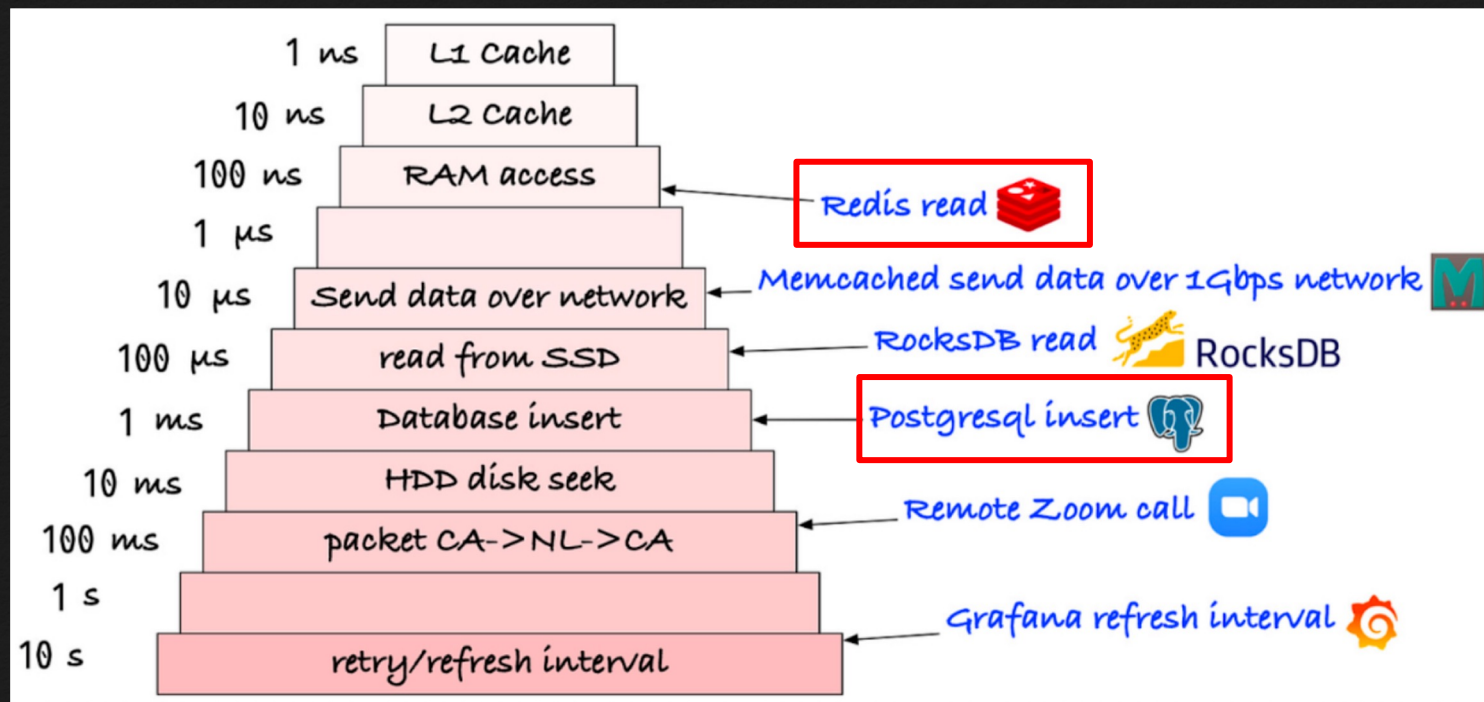
03 Key-value Store



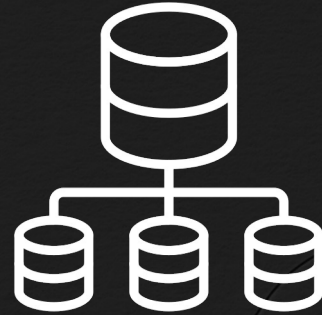
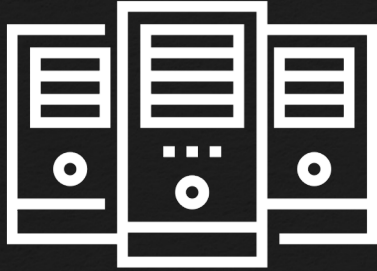
03 Key-value Store



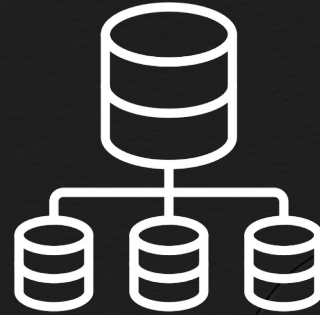
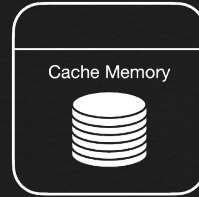
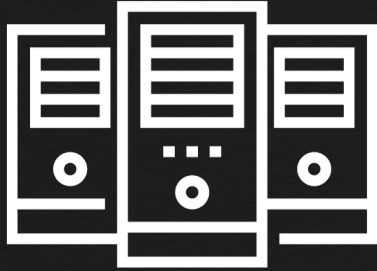
03 Key-value Store



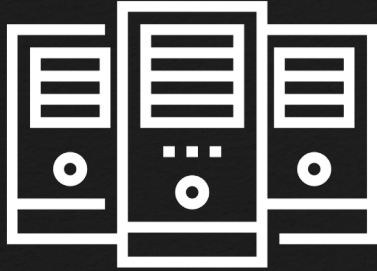
03 Key-value Store



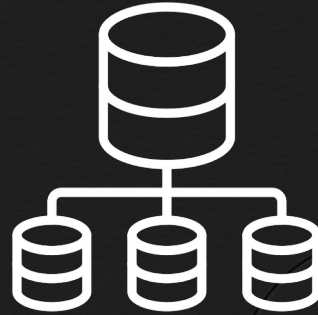
03 Key-value Store



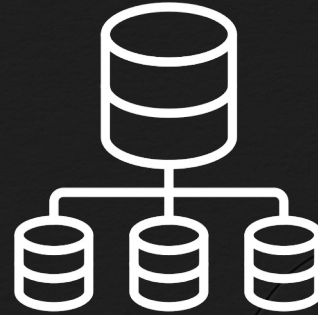
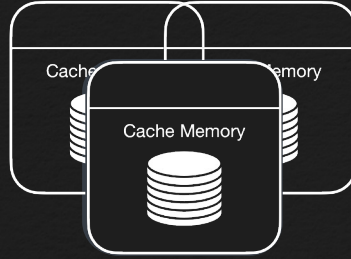
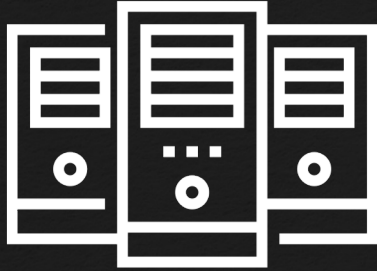
03 Key-value Store



SPOF !

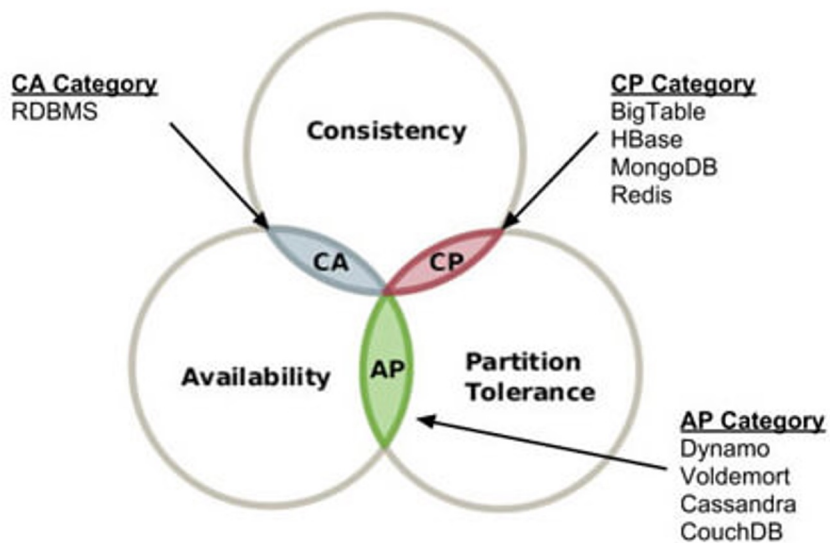


03 Key-value Store

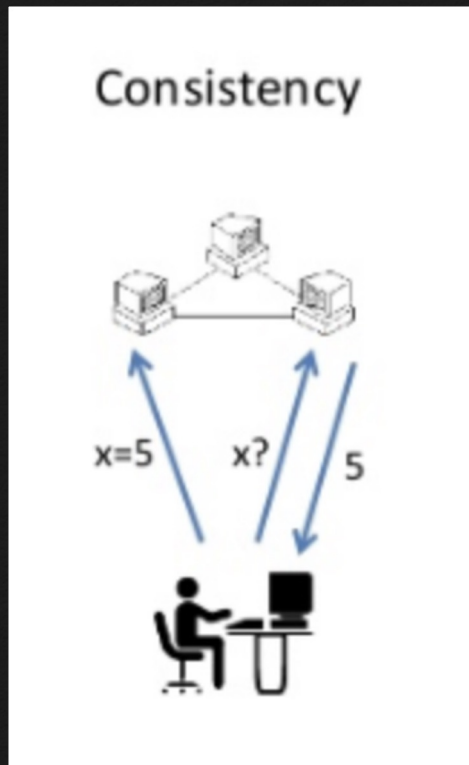


03 Key-value Store

CAP Theorem



03 Key-value Store

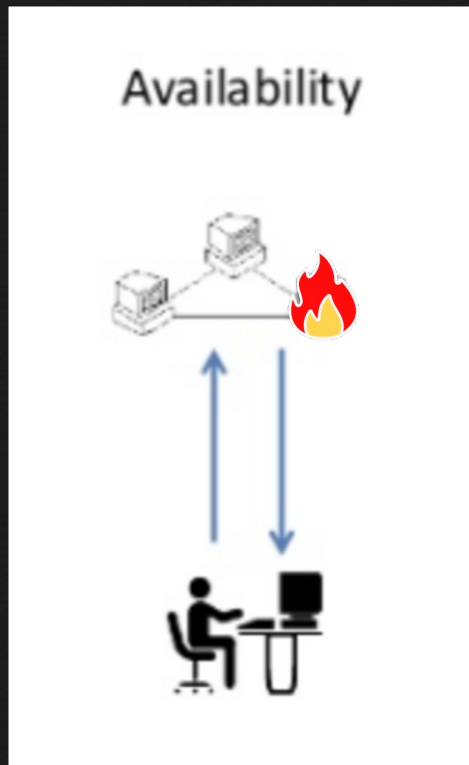


C (Consistency) 데이터 일관성

분산 시스템에 접속하는 모든 클라이언트는 어떤 노드에 접속했느냐에 관계없이 언제나 같은 데이터를 보게 되어야 한다.



03 Key-value Store

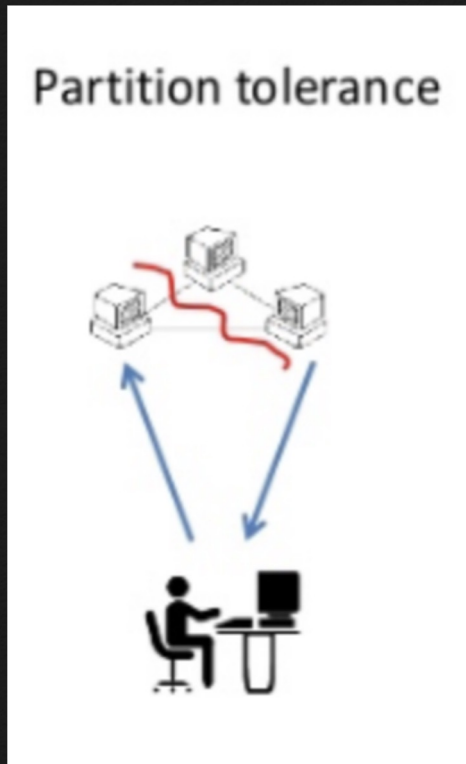


A (Availability) 가용성

분산 시스템에 접속하는 클라이언트는 일부 노드에 장애가 발생하더라도 항상 응답을 받을 수 있어야 한다.



03 Key-value Store

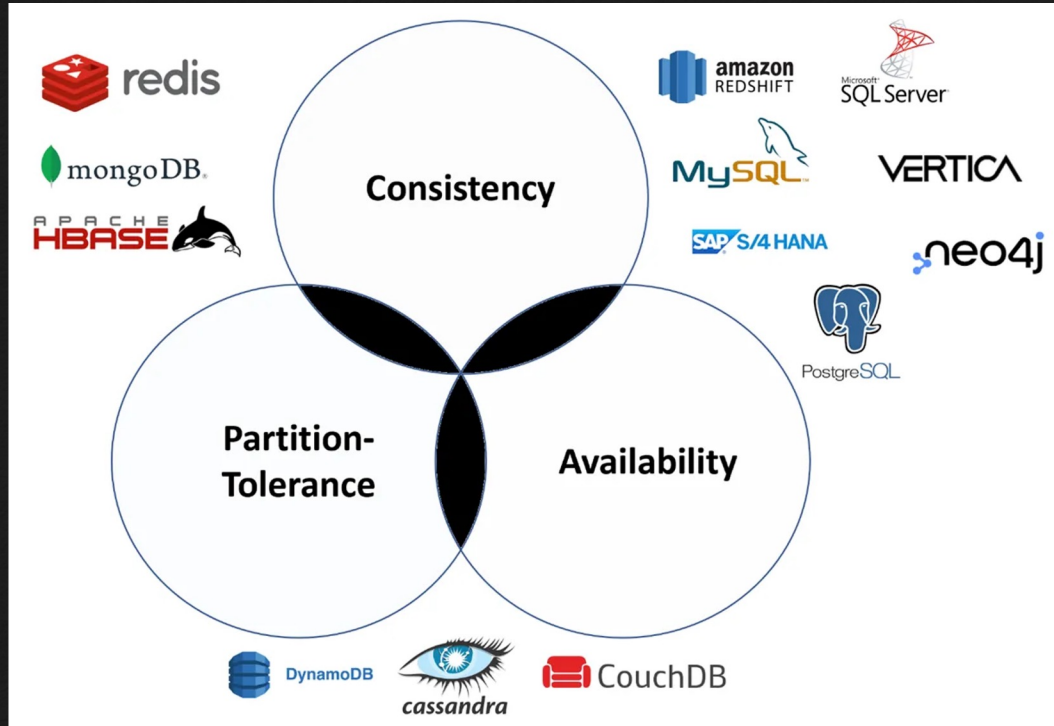


P (Partition tolerance) 파티션 감내

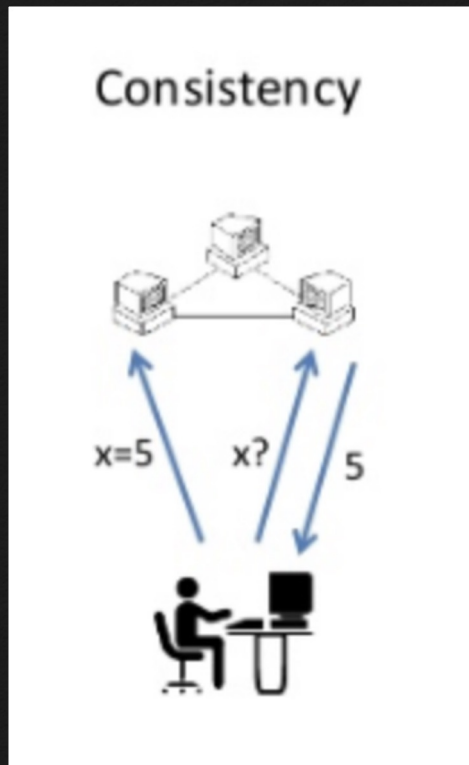
두 노드 사이에 통신 장애가 발생하더라도 시스템은 계속 동작해야 한다.



03 Key-value Store



03 Key-value Store



C (Consistency) 데이터 일관성

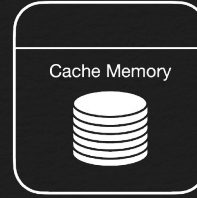
분산 시스템에 접속하는 모든 클라이언트는 어떤 노드에 접속했느냐에 관계없이 언제나 같은 데이터를 보게 되어야 한다.



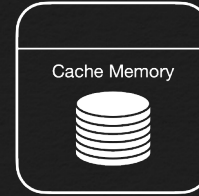
03 Key-value Store - Consistency



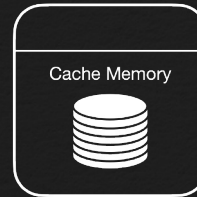
server 1



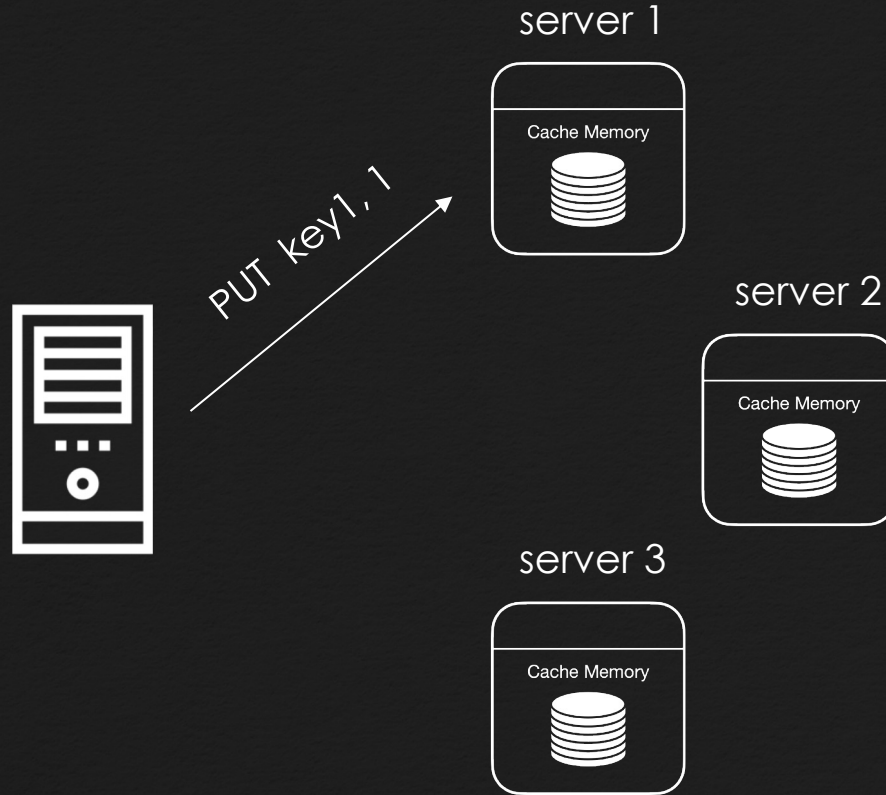
server 2



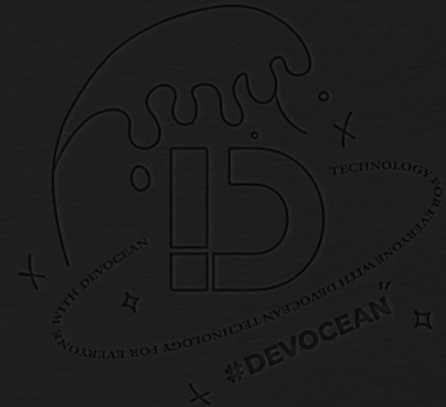
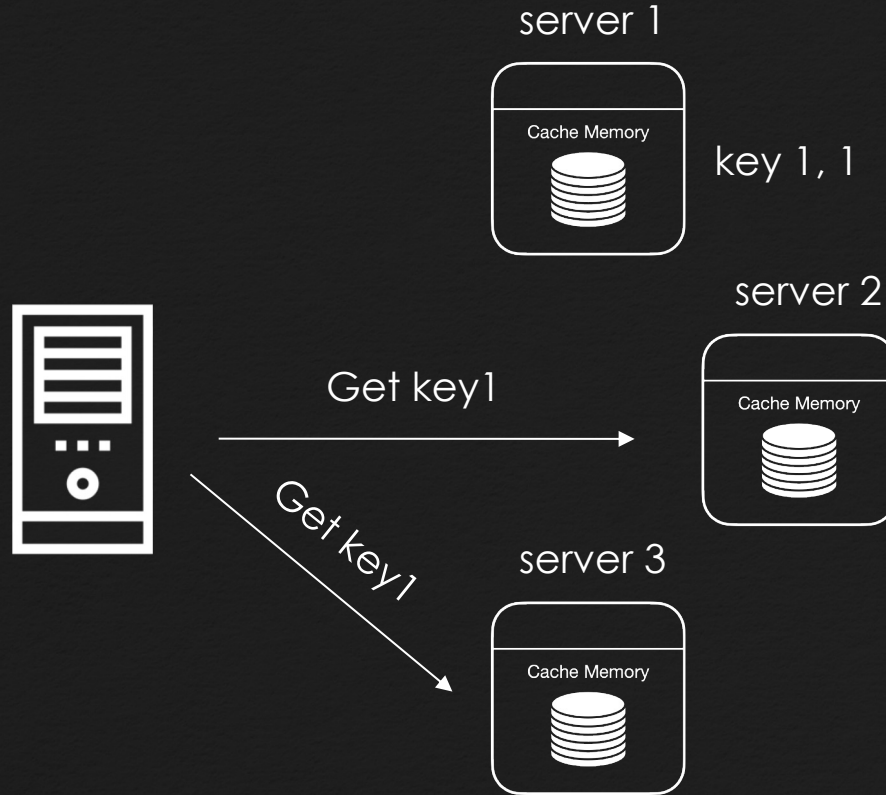
server 3



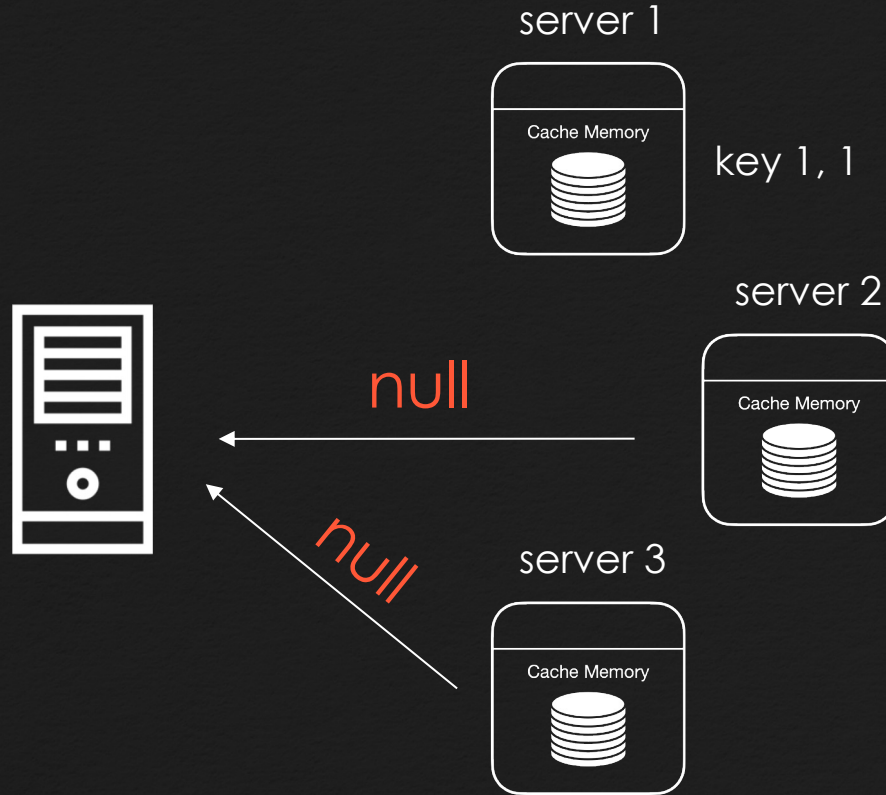
03 Key-value Store - Consistency



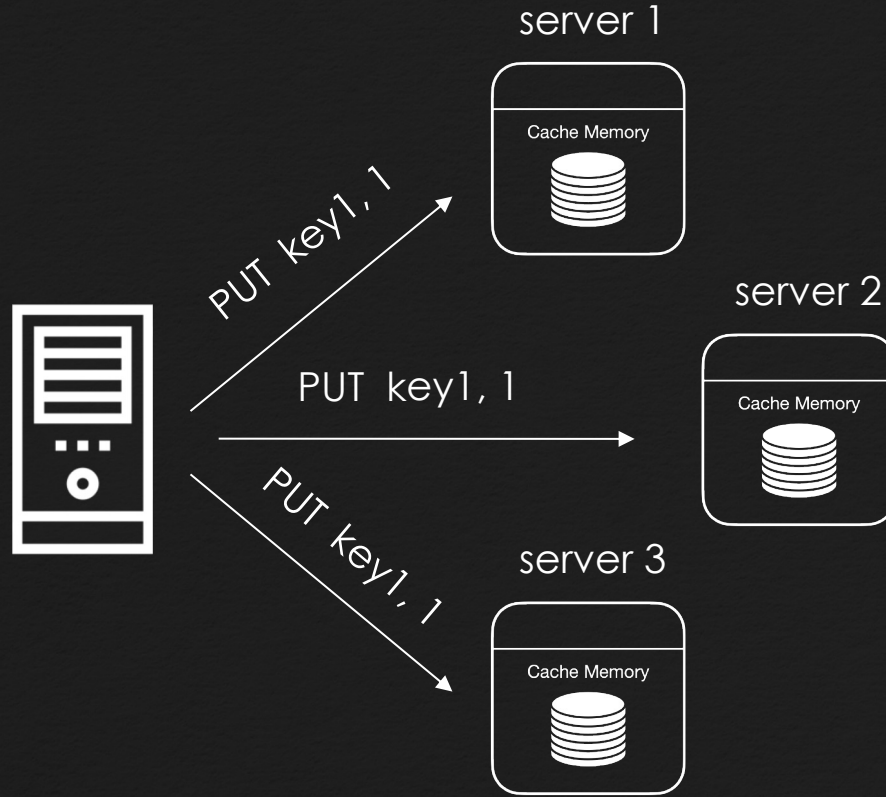
03 Key-value Store - Consistency



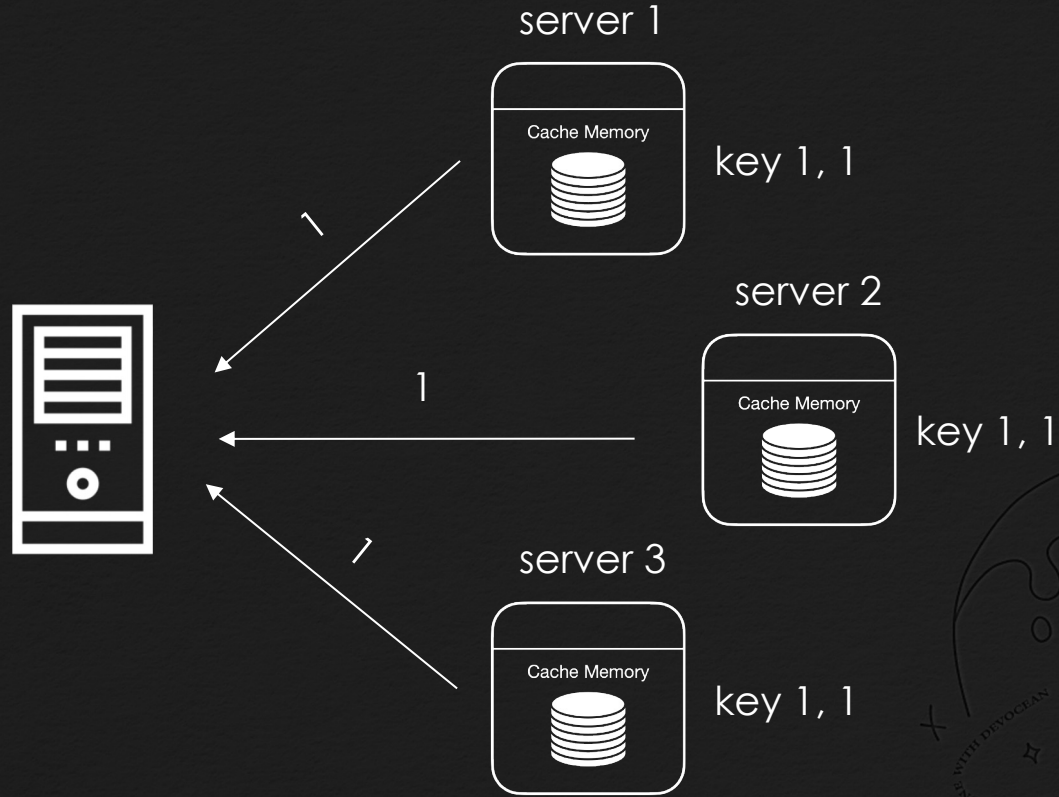
03 Key-value Store - Consistency



03 Key-value Store - Consistency



03 Key-value Store - Consistency



03 Key-value Store - Consistency

Quorum Consensus



03 Key-value Store - Consistency

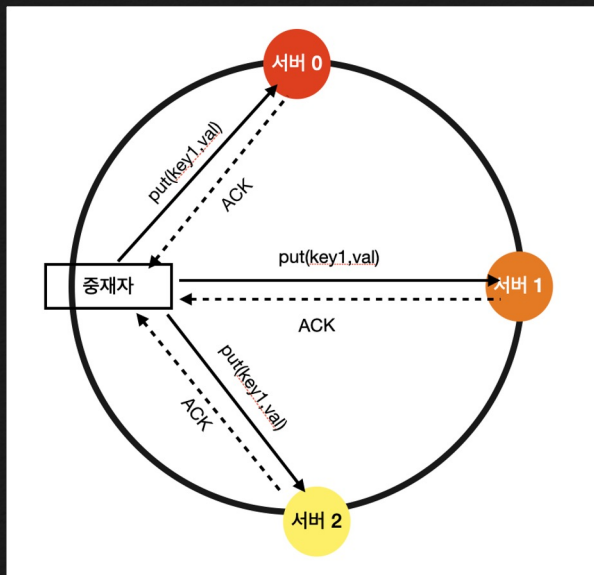


Quorum :

Quorum는 여러 사람의 합의로 운영되는 의사기관에서 의결을 하는데 필요한 최소한의 참석자 수이다. 위키백과



03 Key-value Store - Consistency



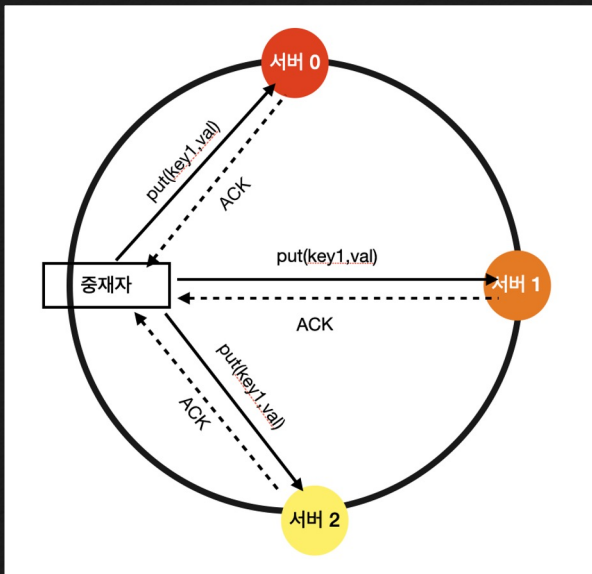
N = 사본의 개수

W (Write) = 쓰기 연산에 대한 Quorum

R (Read) = 읽기 연산에 대한 Quorum



03 Key-value Store - Consistency



$R = 1, W = N$: 빠른 읽기 연산에 최적화된 시스템

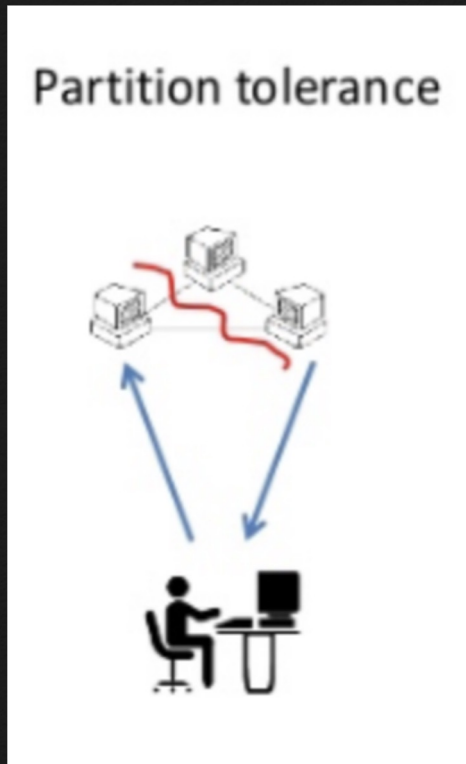
$W = 1, R = N$: 빠른 쓰기 연산에 최적화된 시스템

$W + R > N$: 강한 일관성이 보장됨

$W + R \leq N$: 강한 일관성이 보장되지 않음

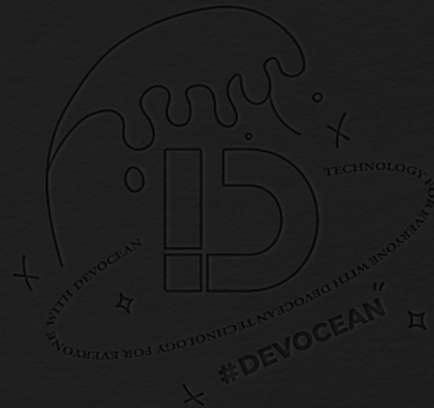


03 Key-value Store



P (Partition tolerance) 파티션 감내

두 노드 사이에 통신 장애가 발생하더라도 시스템은 계속 동작해야 한다.



03 Key-value Store - Partition tolerance

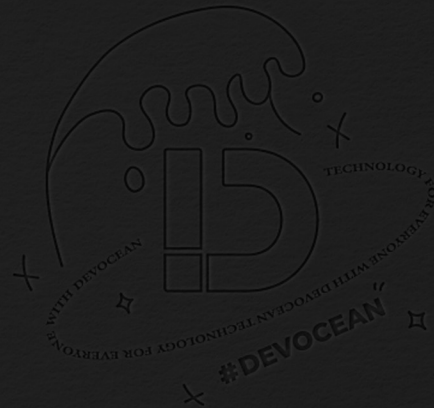


연결이 끊김

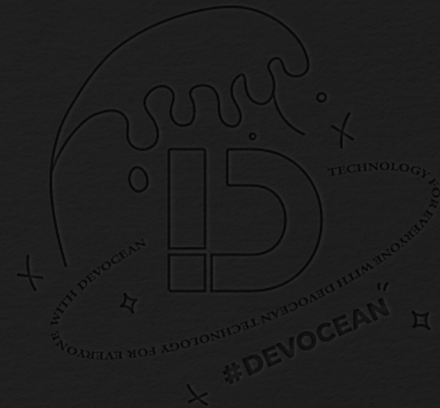
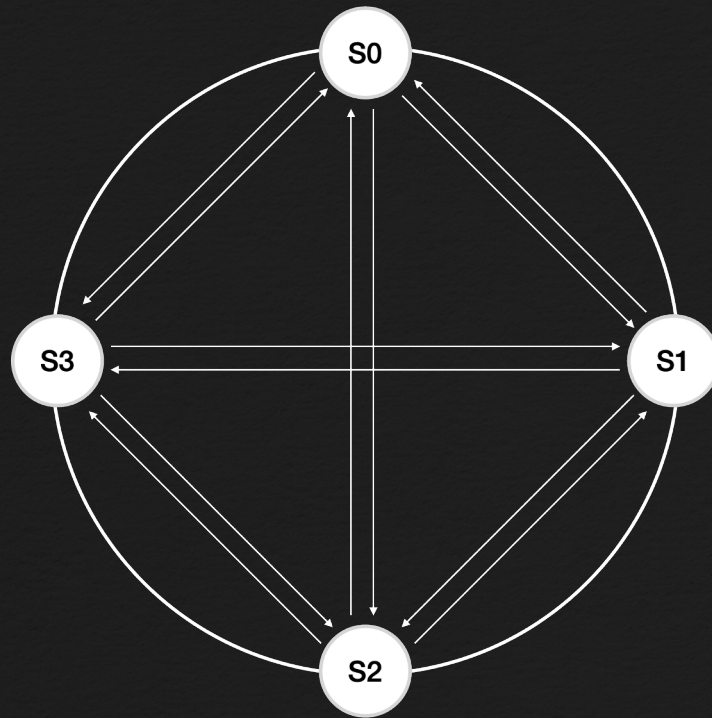
네트워크 변경이 감지되었습니다.

ERR_NETWORK_CHANGED

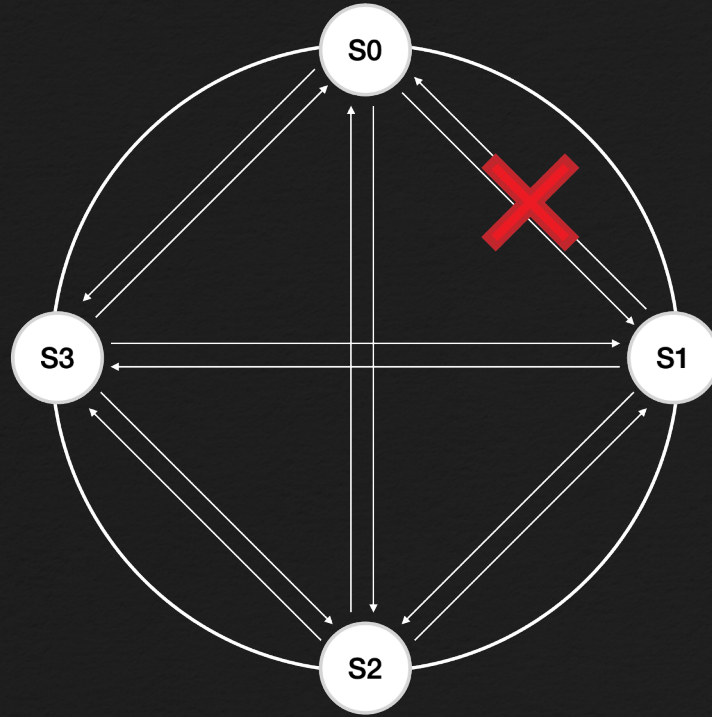
새로고침



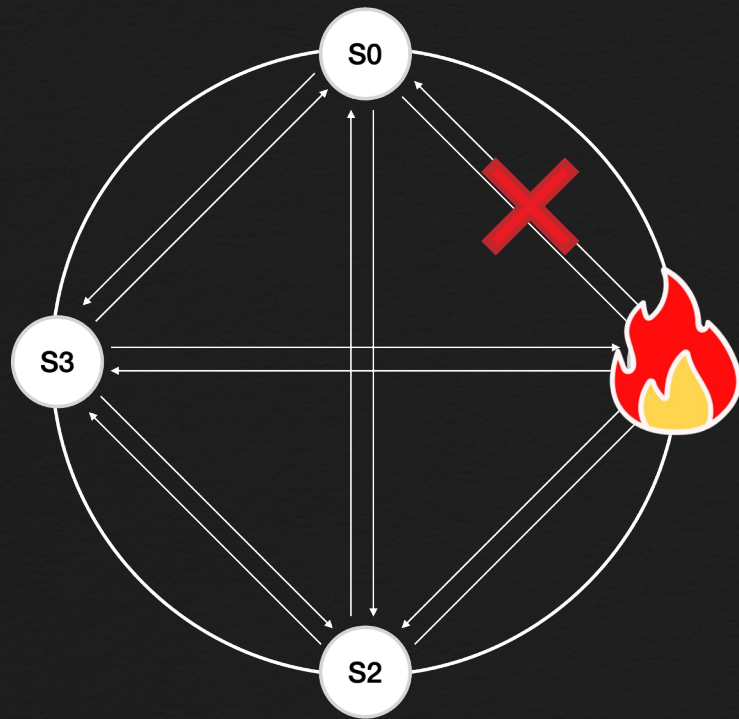
03 Key-value Store - Partition tolerance



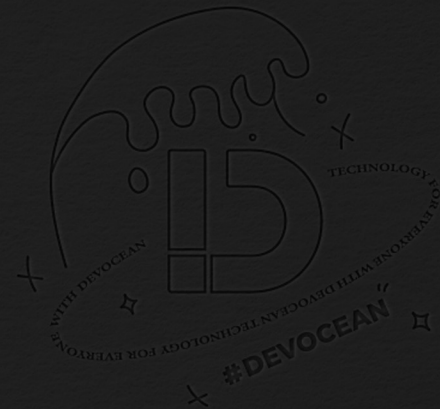
03 Key-value Store - Partition tolerance



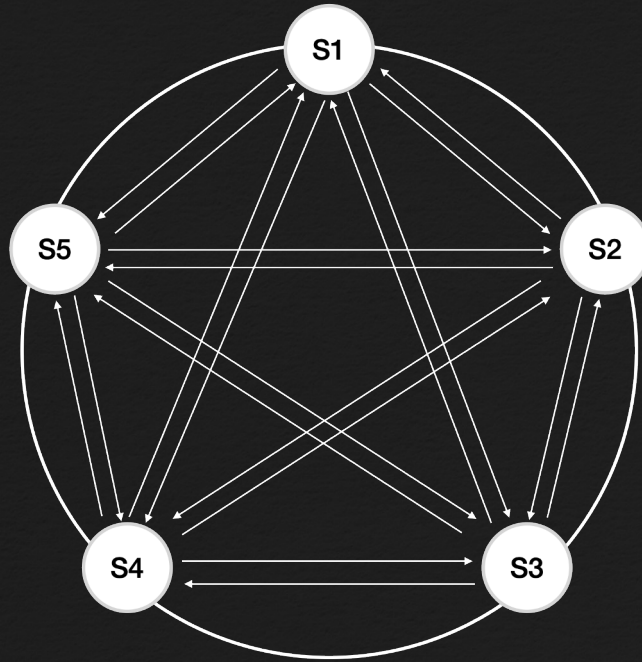
03 Key-value Store - Partition tolerance



S1 장애!



03 Key-value Store - Partition tolerance

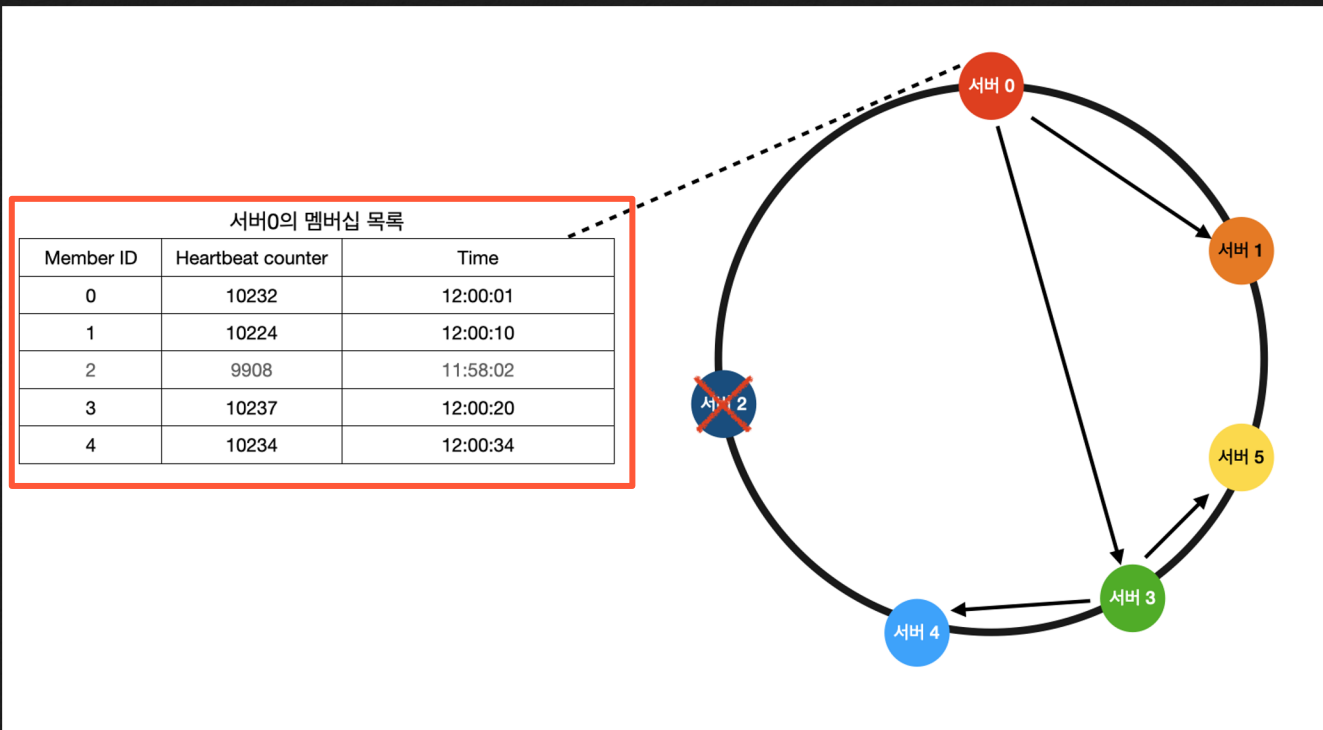


03 Key-value Store - Partition tolerance

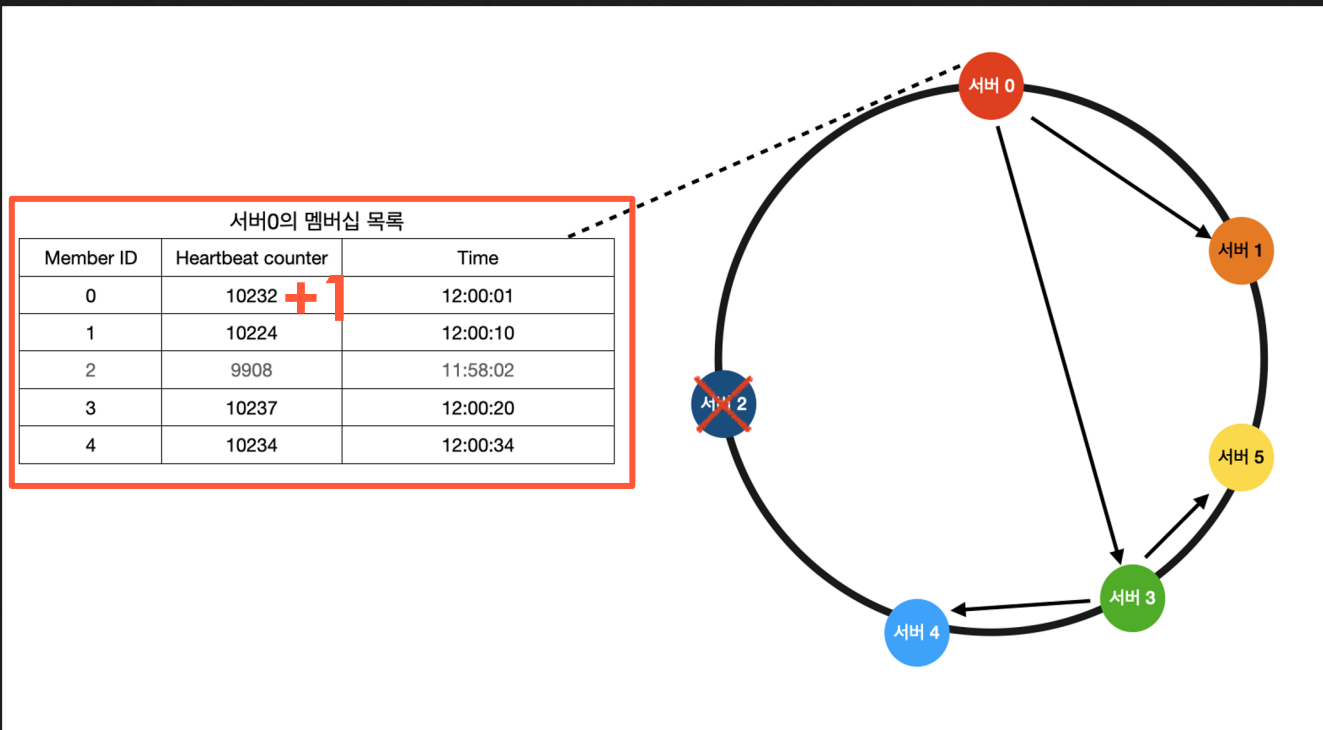
Gossip protocol



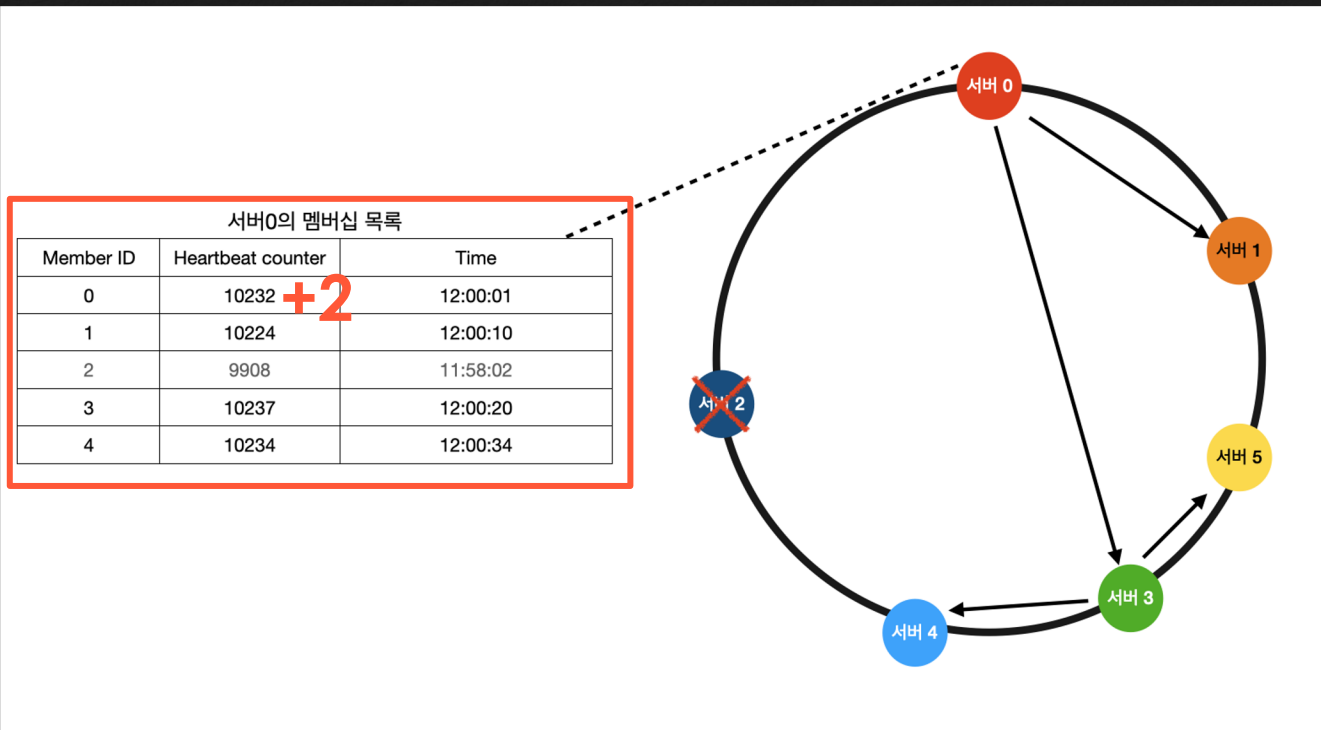
03 Key-value Store - Partition tolerance



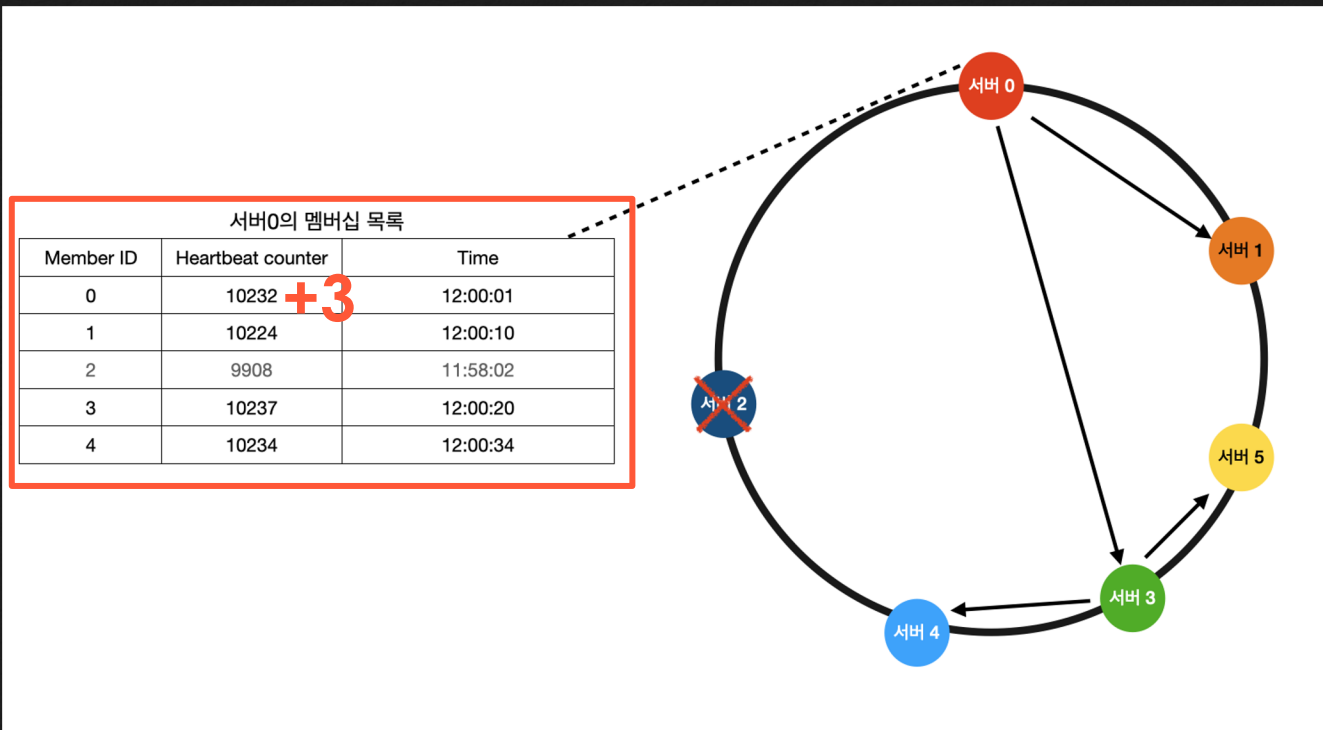
03 Key-value Store - Partition tolerance



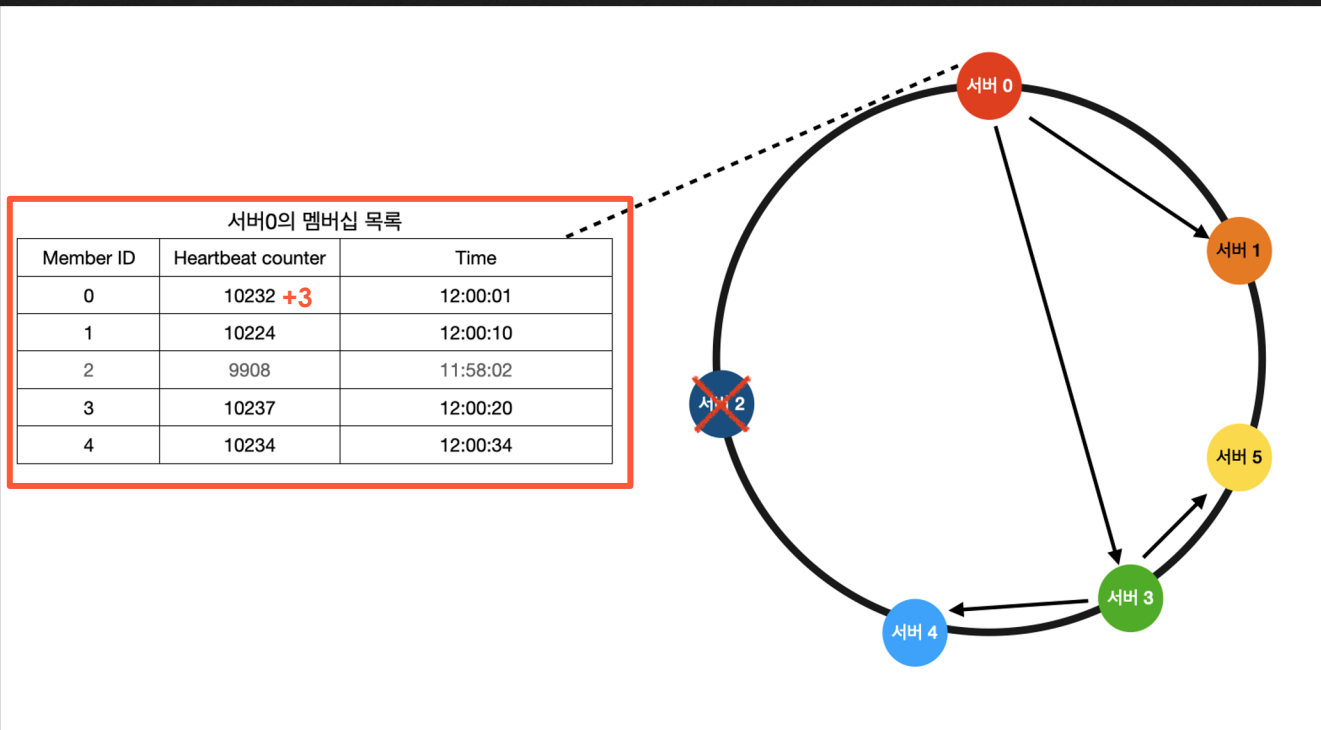
03 Key-value Store - Partition tolerance



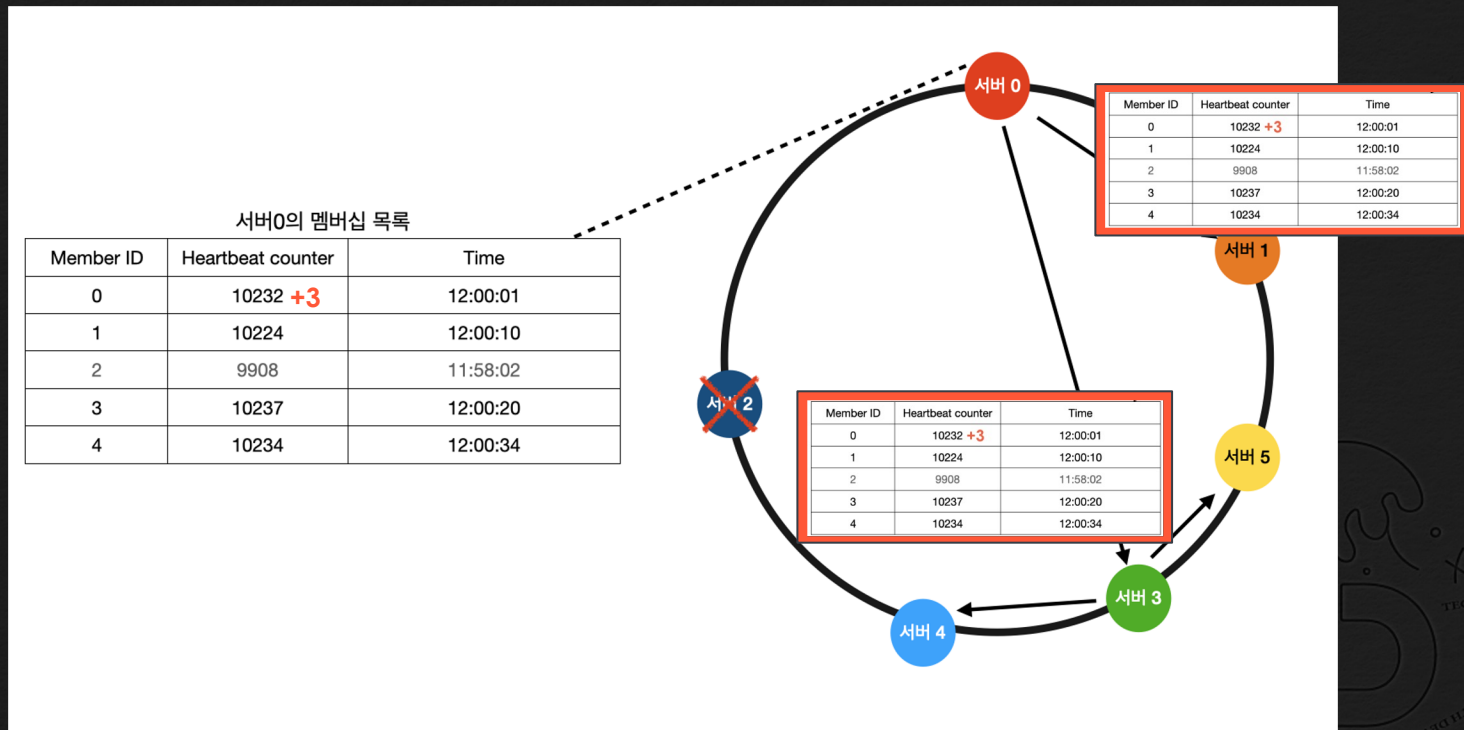
03 Key-value Store - Partition tolerance



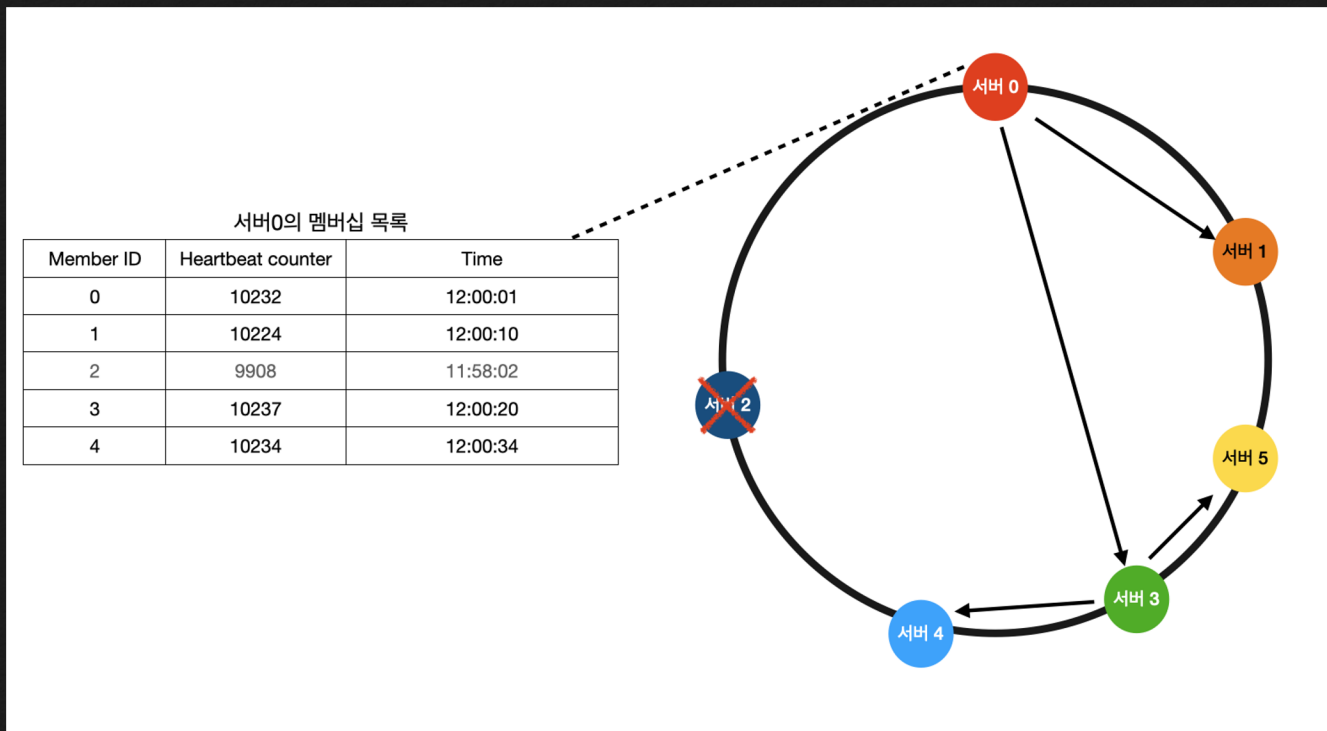
03 Key-value Store - Partition tolerance



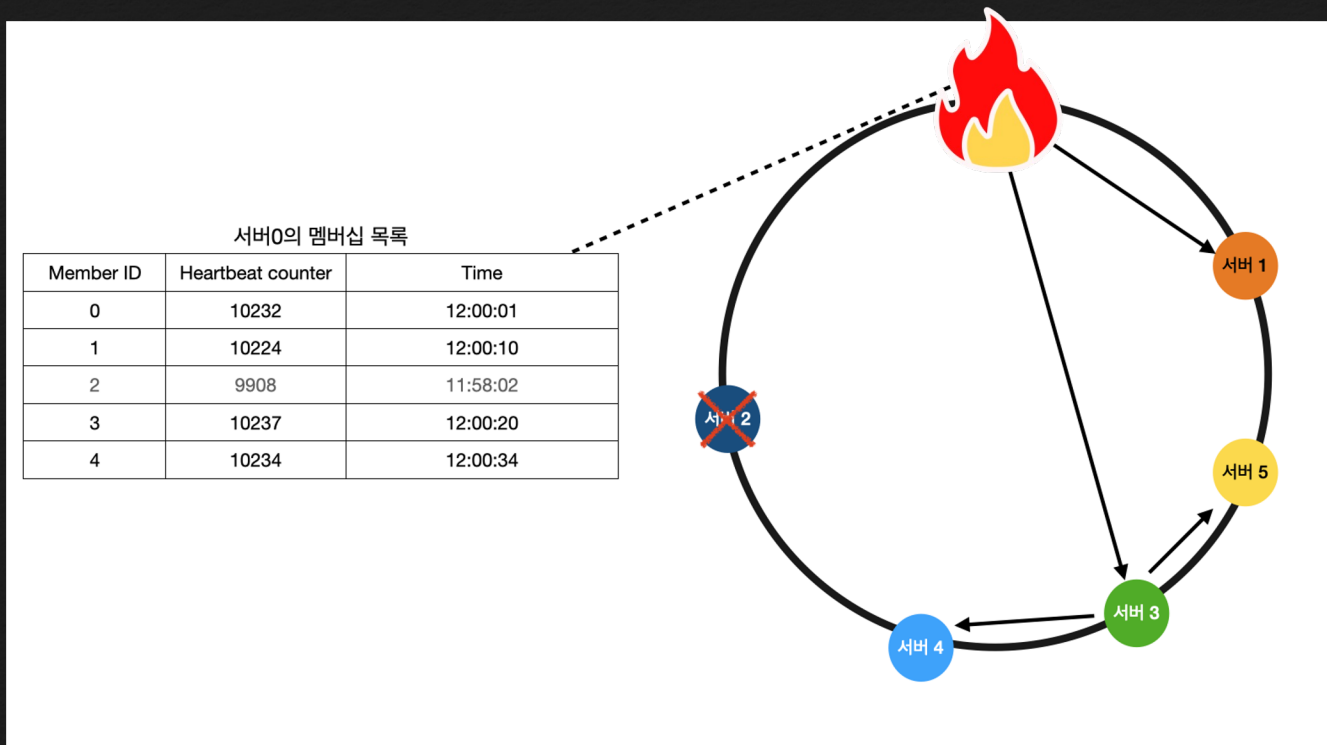
03 Key-value Store - Partition tolerance



03 Key-value Store - Partition tolerance



03 Key-value Store - Partition tolerance



Quorum Consensus

$R = 1, W = N$: 빠른 읽기 연산에 최적화된 시스템

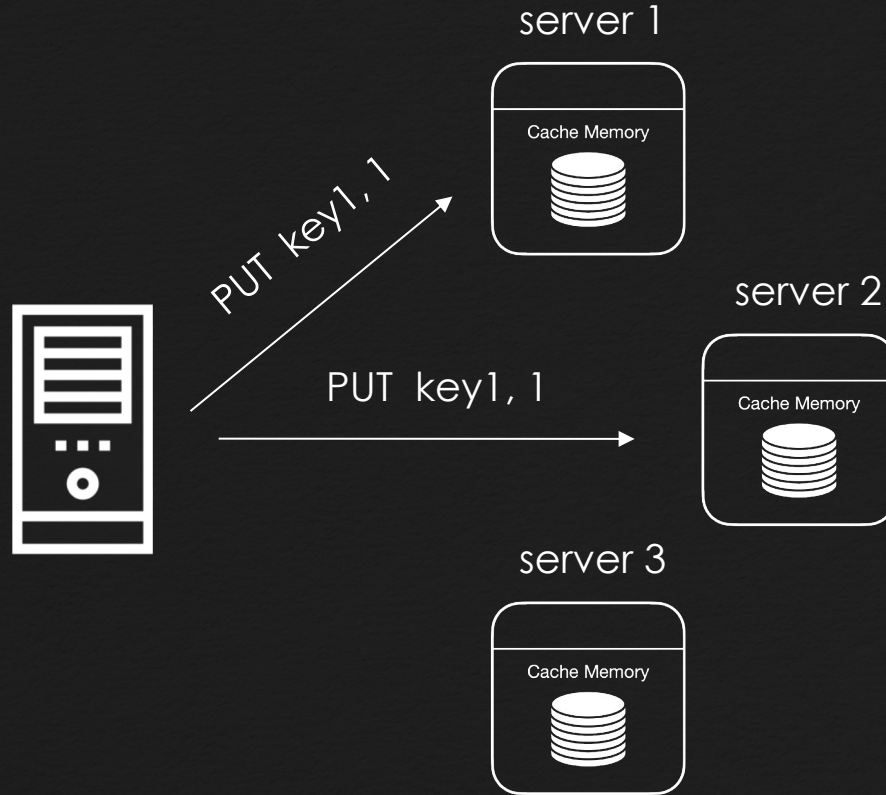
$W = 1, R = N$: 빠른 쓰기 연산에 최적화된 시스템

$W + R > N$: 강한 일관성이 보장됨

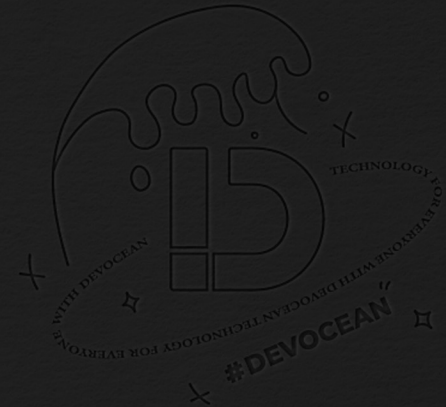
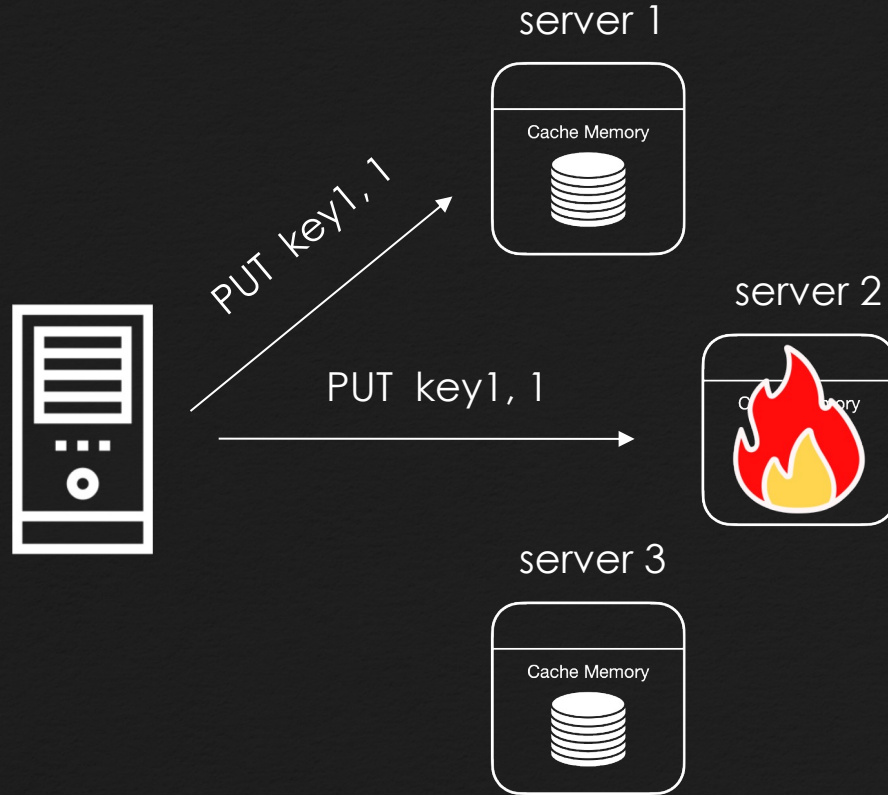
$W + R \leq N$: 강한 일관성이 보장되지 않음



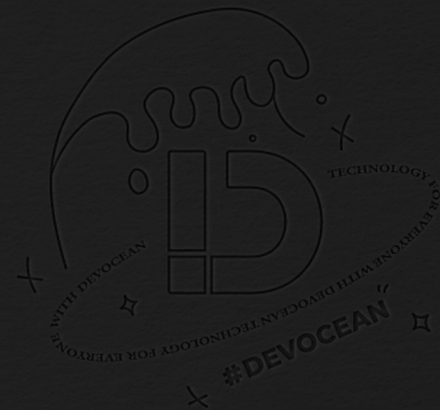
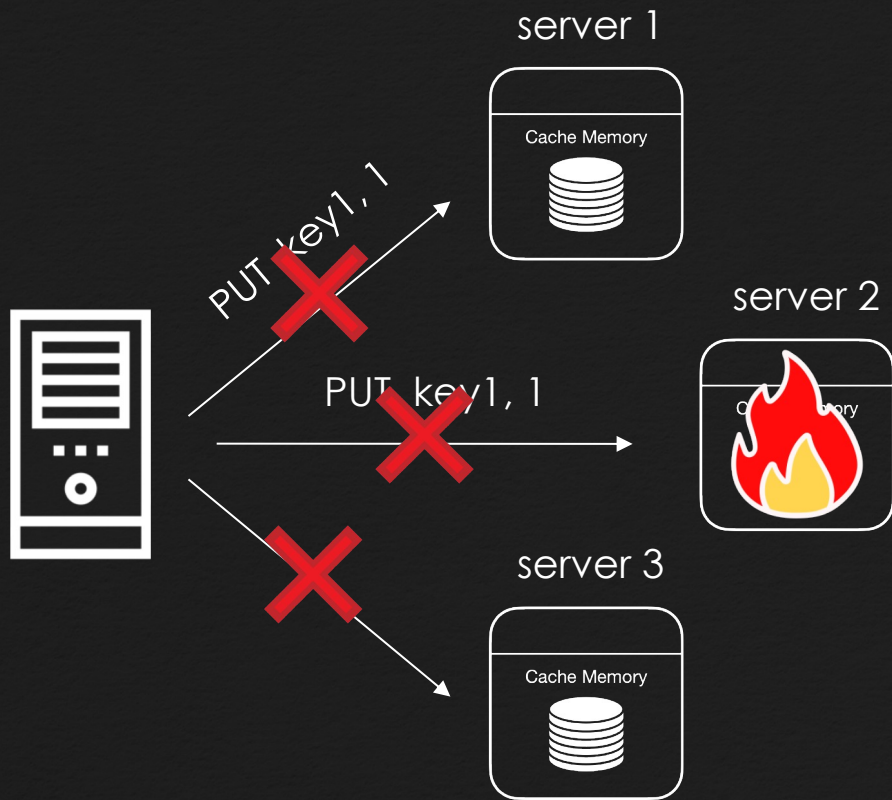
03 Key-value Store - Partition tolerance



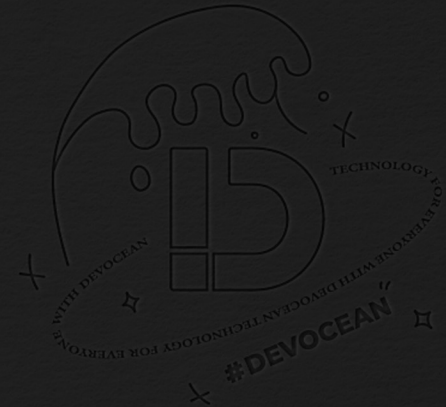
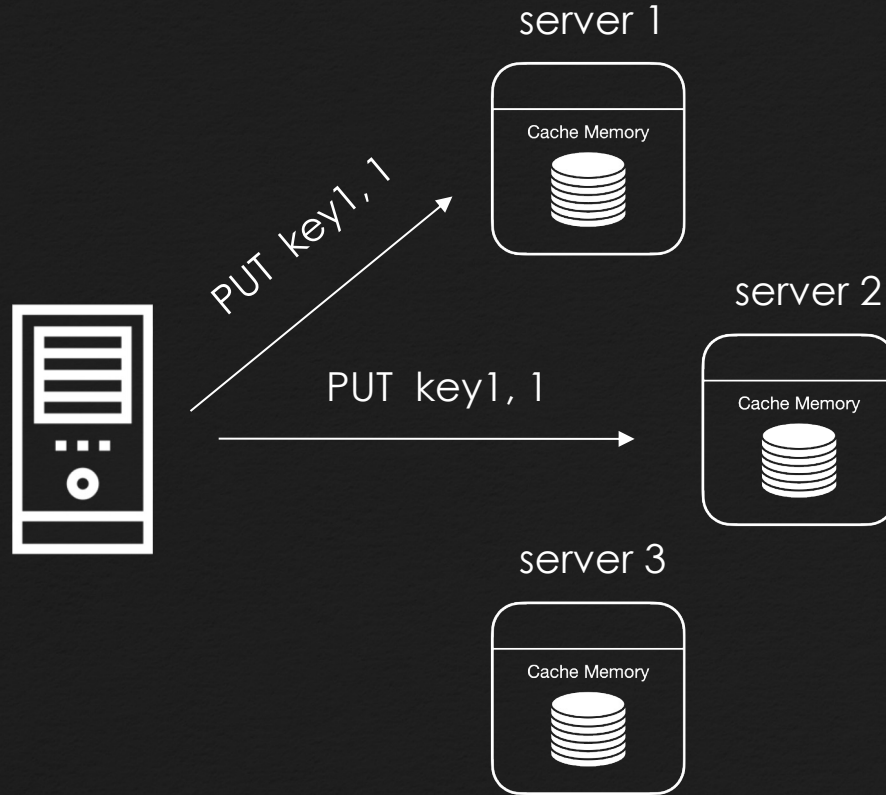
03 Key-value Store - Partition tolerance



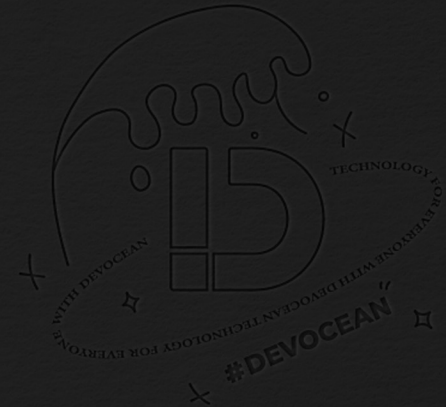
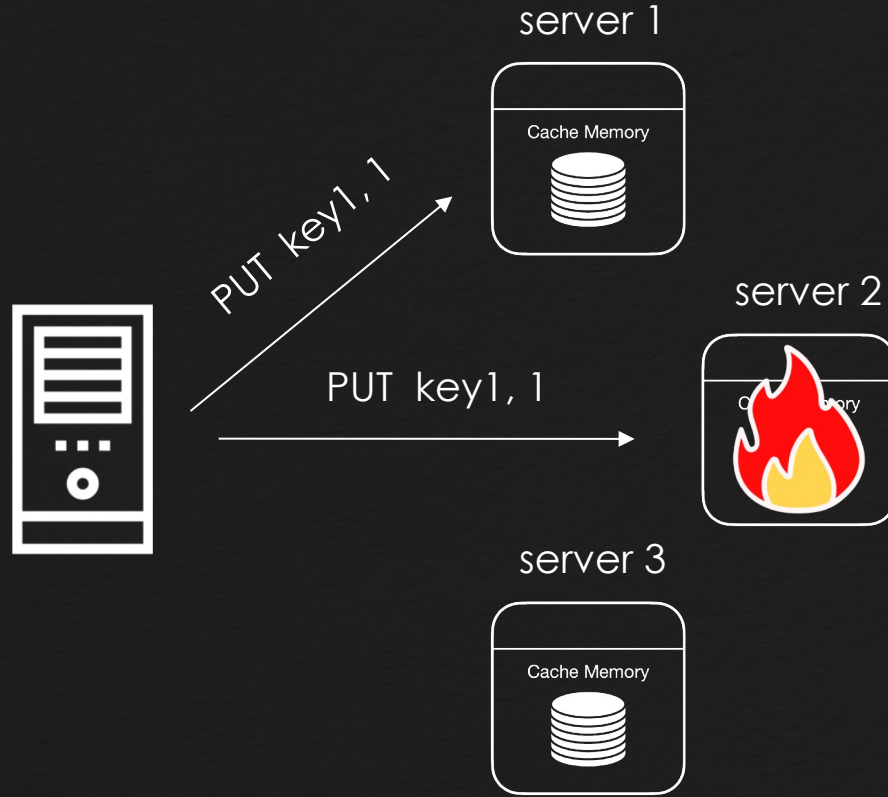
03 Key-value Store - Partition tolerance



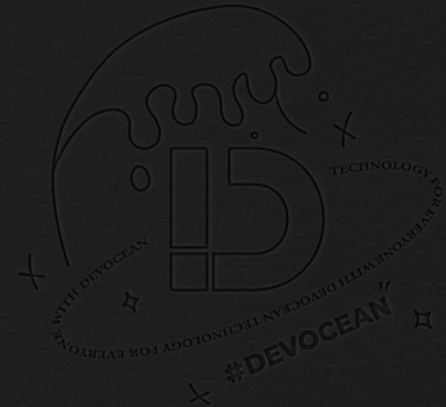
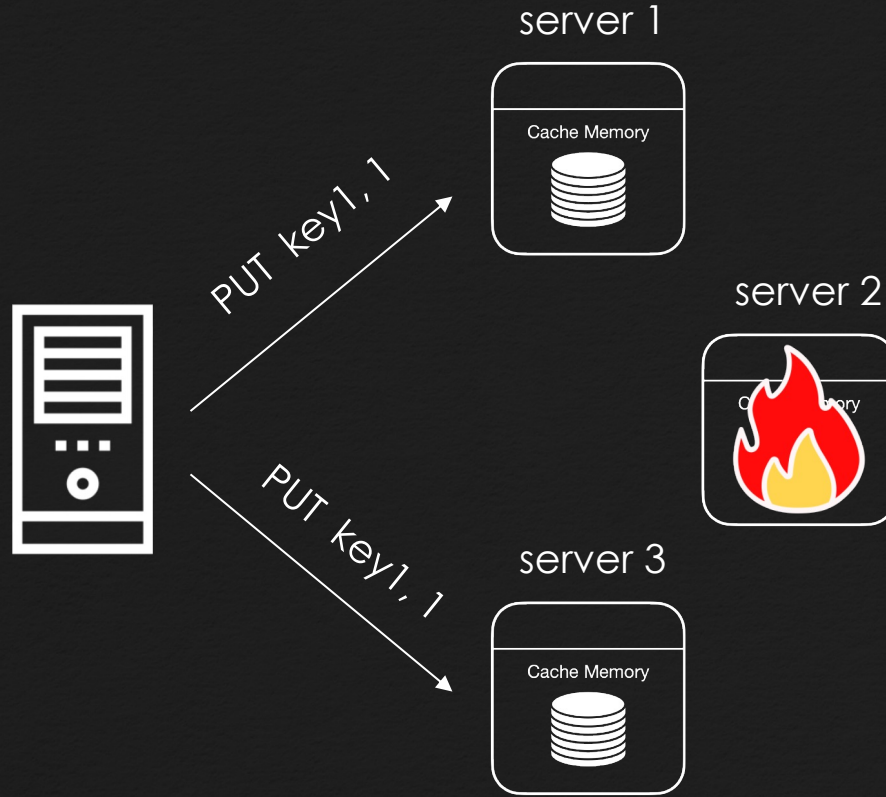
03 Key-value Store - Partition tolerance



03 Key-value Store - Partition tolerance



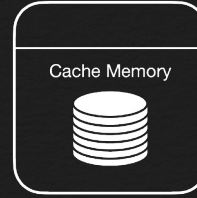
03 Key-value Store - Partition tolerance



03 Key-value Store - Partition tolerance

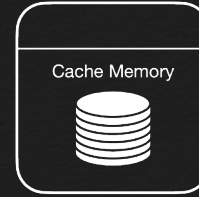


server 1



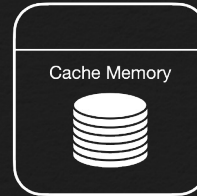
key 1, 1

server 2

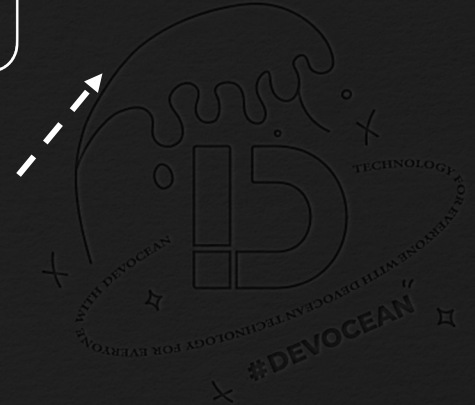


key 1, 1

server 3



key 1, 1



목차

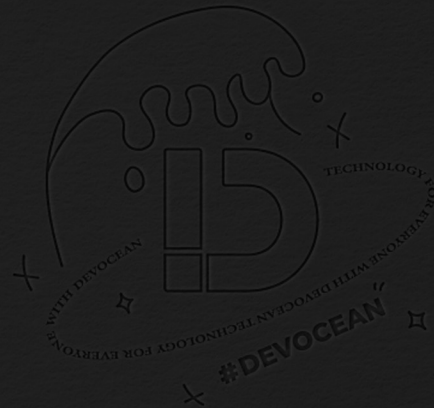
01 Server

02 Database

03 Key-value Store

04 Consistent Hashing

05 GUID



목차

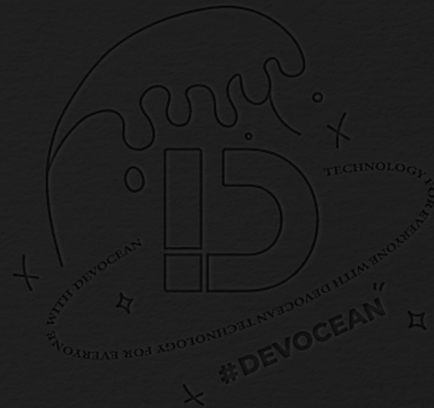
01 Server

02 Database

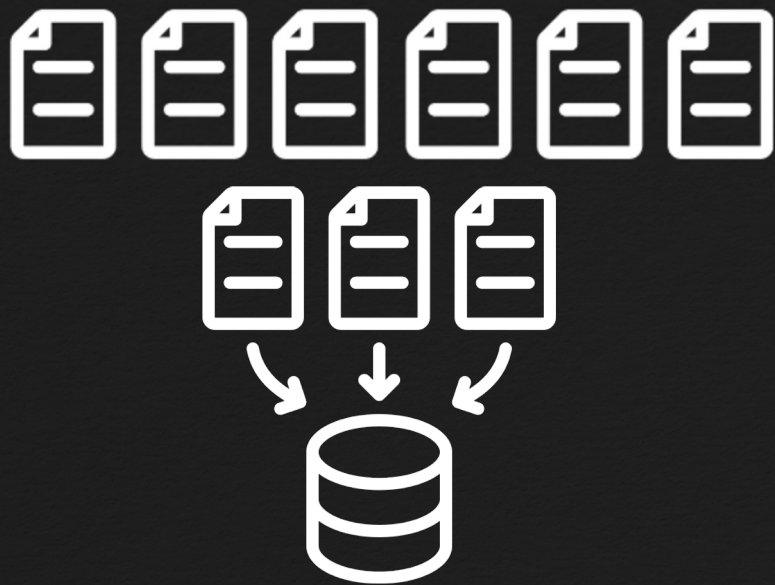
03 Key-value Store

04 Consistent Hashing

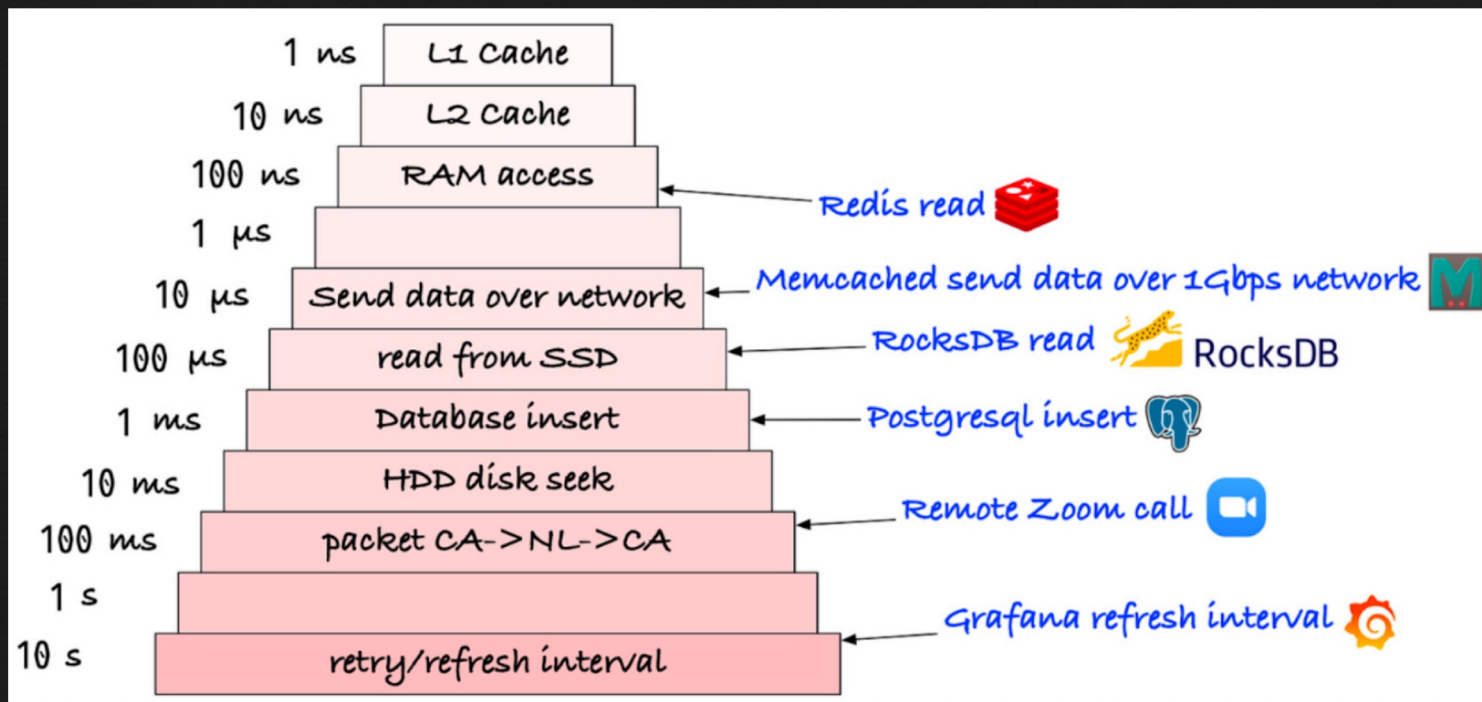
05 GUID



04 Consistent Hashing



04 Consistent Hashing



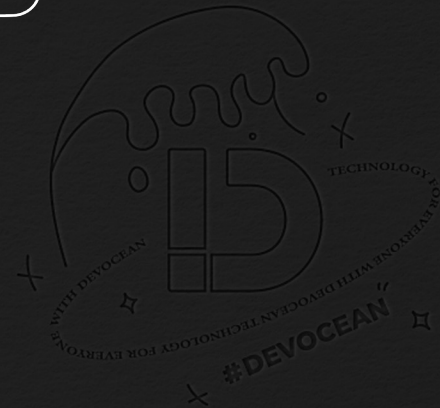
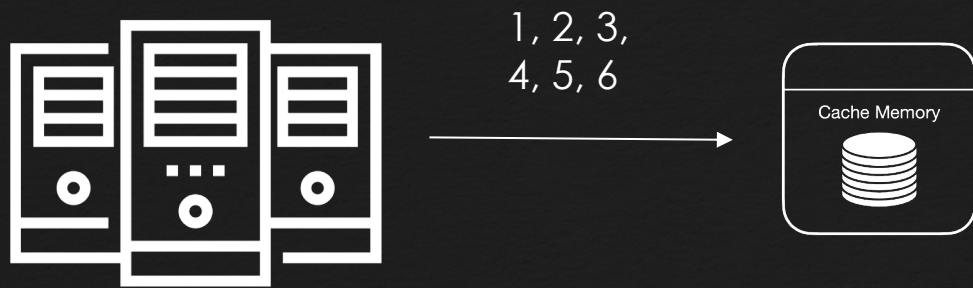
04 Consistent Hashing

메모리 최적화 캐시 노드 - 현재 세대

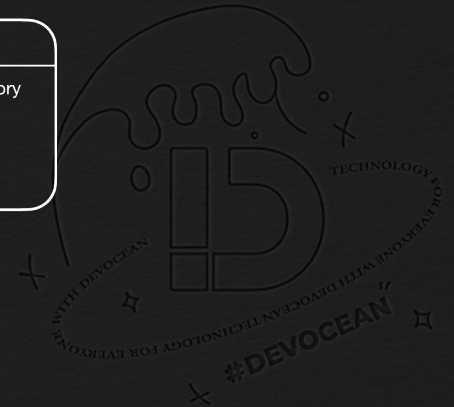
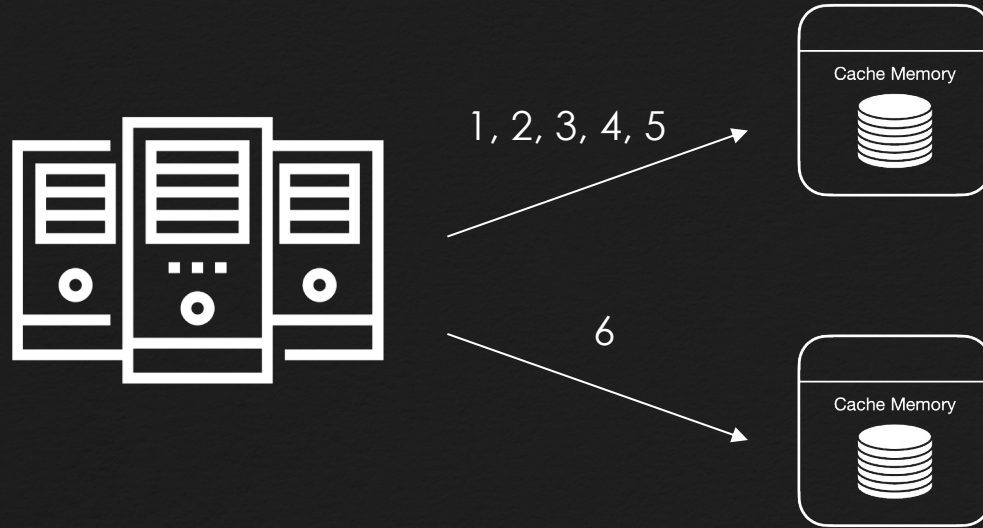
캐시 노드 유형	vCPU	메모리	네트워크 성능	시간당 요금
cache.r7g.large	2	13.07 GiB	최대 12.5기가비트	USD 0.262
cache.r7g.xlarge	4	26.32 GiB	최대 12.5기가비트	USD 0.522
⋮				
cache.r5.12xlarge	48	317.77 GiB	10기가비트	USD 6.207
cache.r5.24xlarge	96	635.61 GiB	25기가비트	USD 12.414



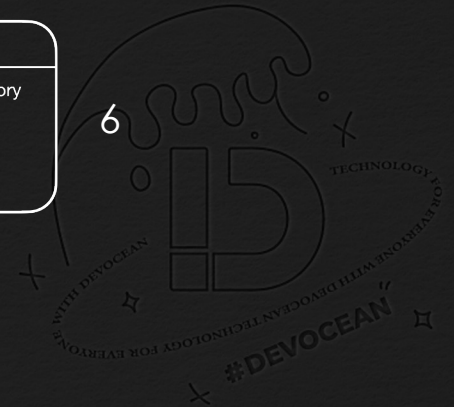
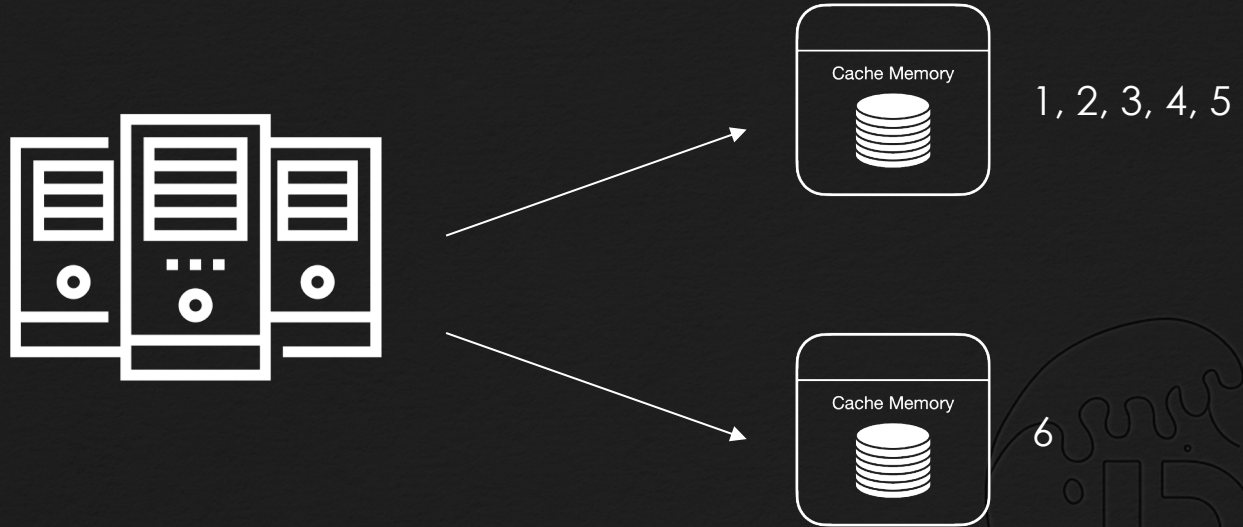
04 Consistent Hashing



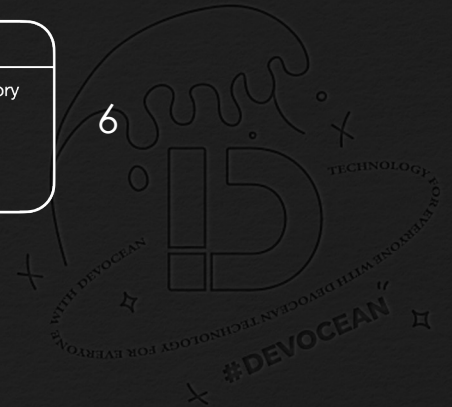
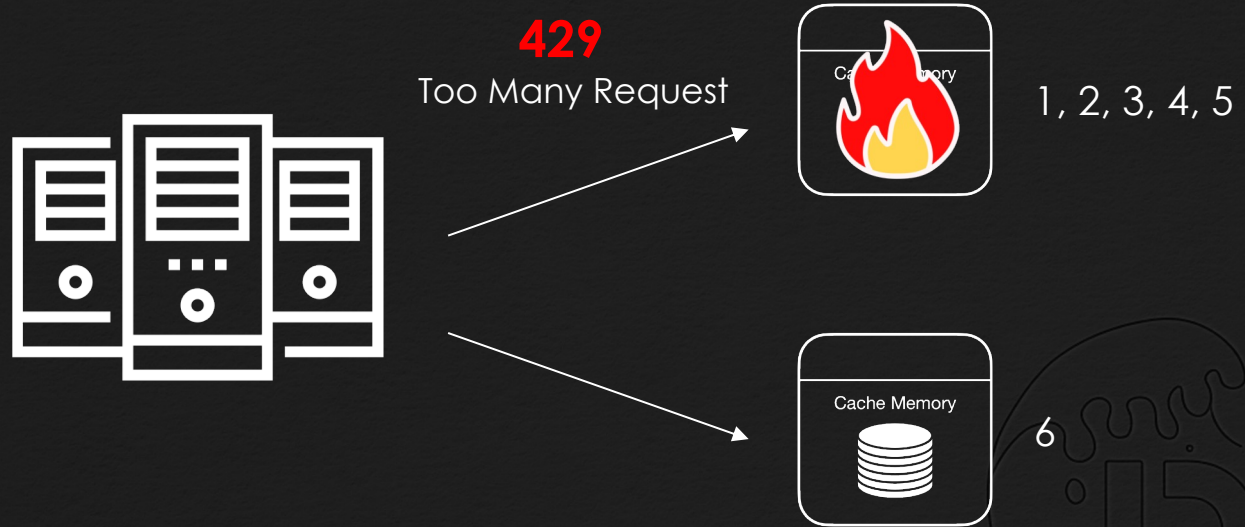
04 Consistent Hashing



04 Consistent Hashing



04 Consistent Hashing



04 Consistent Hashing

$$\text{server Index} = \text{hash} \% 4$$

서버 인덱스

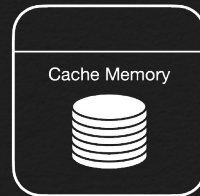
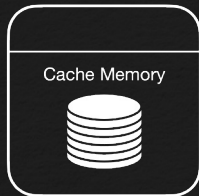
0

1

2

3

서버



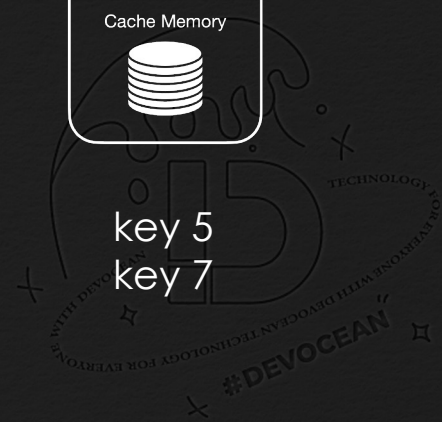
키

key 1
key 3

key 0
key 4

key 2
key 6

key 5
key 7

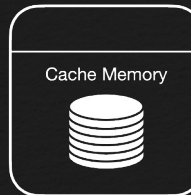


04 Consistent Hashing

서버 인덱스

0

서버



키

key 1
key 3

key 0
key 4

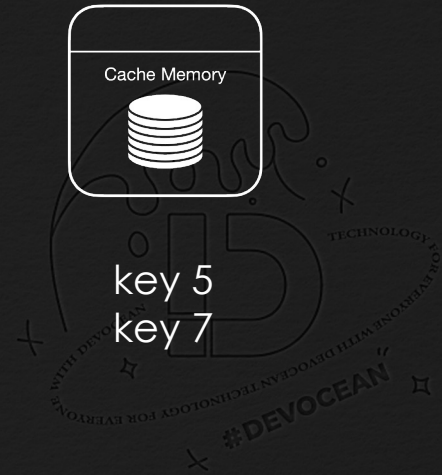
key 2
key 6

키	해시	해시 % 4 (서버 인덱스)
key0	18358617	1
key1	26143584	0
key2	18131146	2
key3	35863496	0
key4	34085809	1
key5	27581703	3
key6	38164978	2
key7	22530351	3

3



key 5
key 7



04 Consistent Hashing

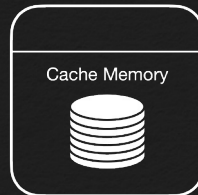
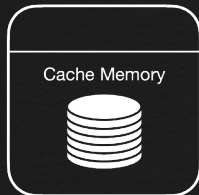
$$\text{server Index} = \text{hash} \% 3$$

서버 인덱스 0

2

3

서버



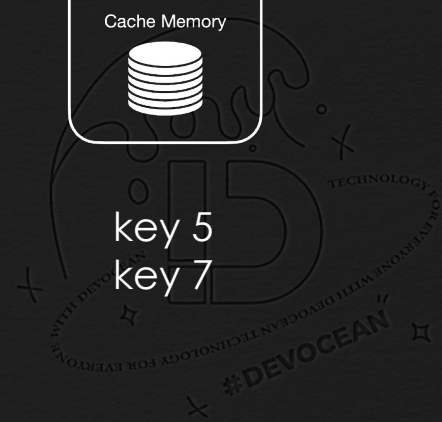
키

key 1
key 3

key 0
key 4

key 2
key 6

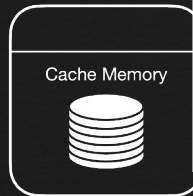
key 5
key 7



04 Consistent Hashing

서버 인덱스 0

서버



키

key 1
key 3

key 0
key 4

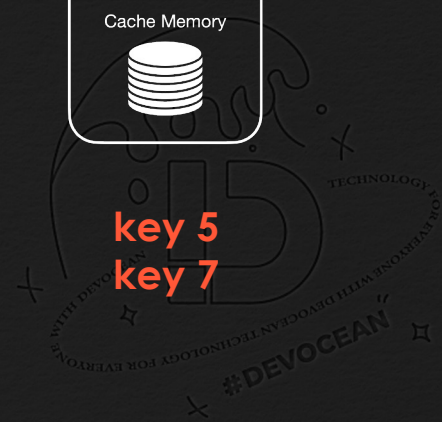
key 2
key 6

키	해시	해시 % 4 (서버 인덱스)
key0	18358617	0
key1	26143584	0
key2	18131146	1
key3	35863496	2
key4	34085809	1
key5	27581703	0
key6	38164978	1
key7	22530351	0

3

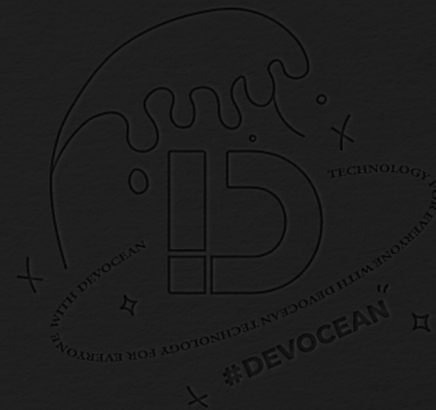


key 5
key 7

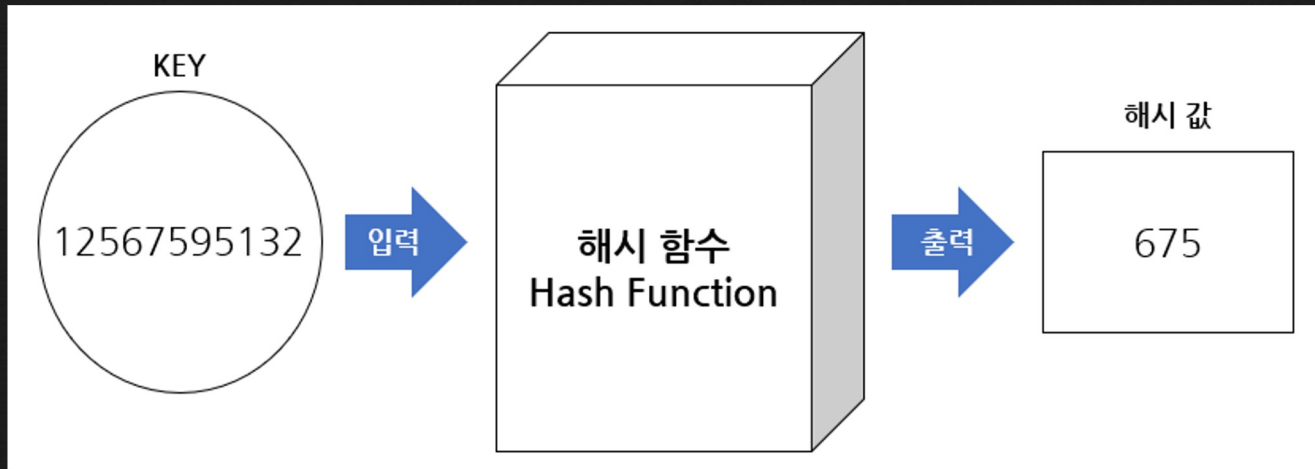


04 Consistent Hashing

Consistent Hashing



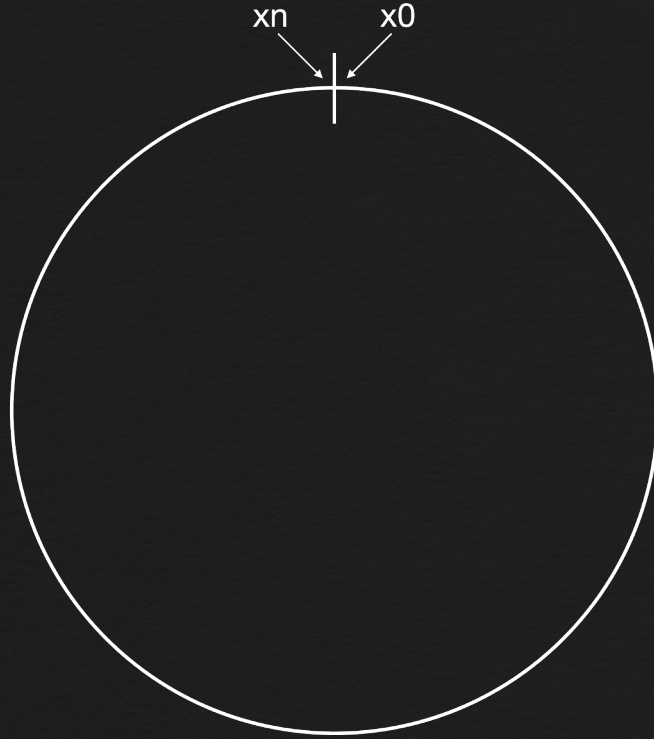
04 Consistent Hashing



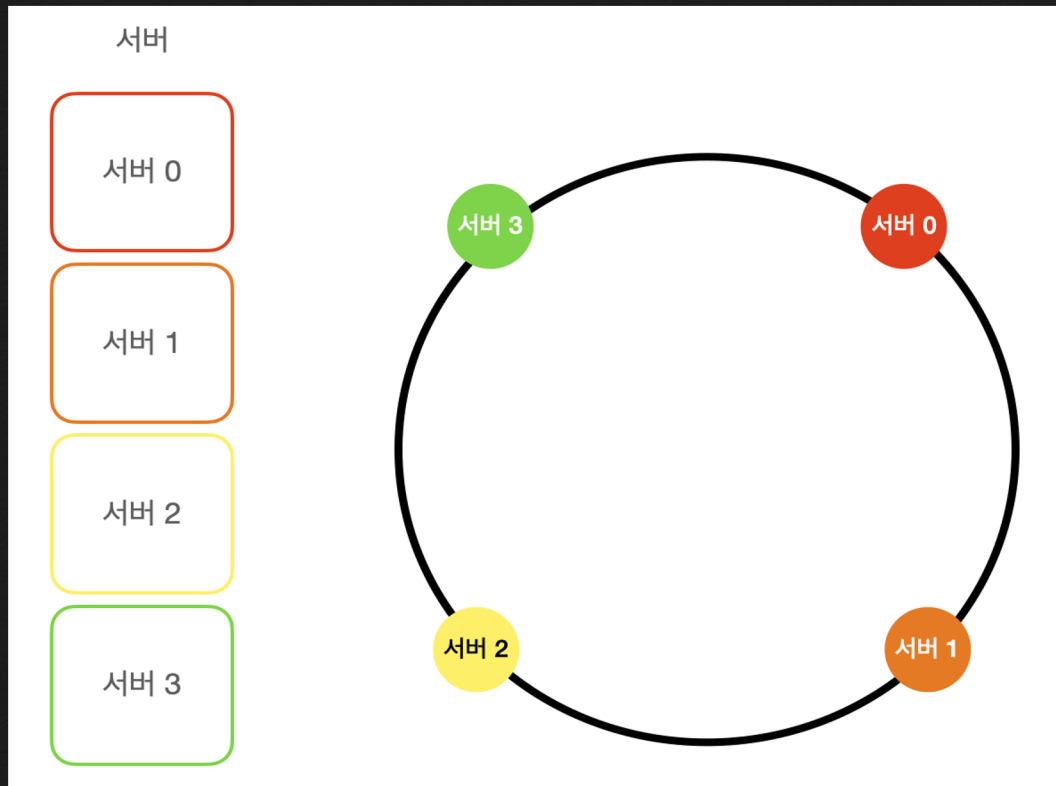
04 Consistent Hashing



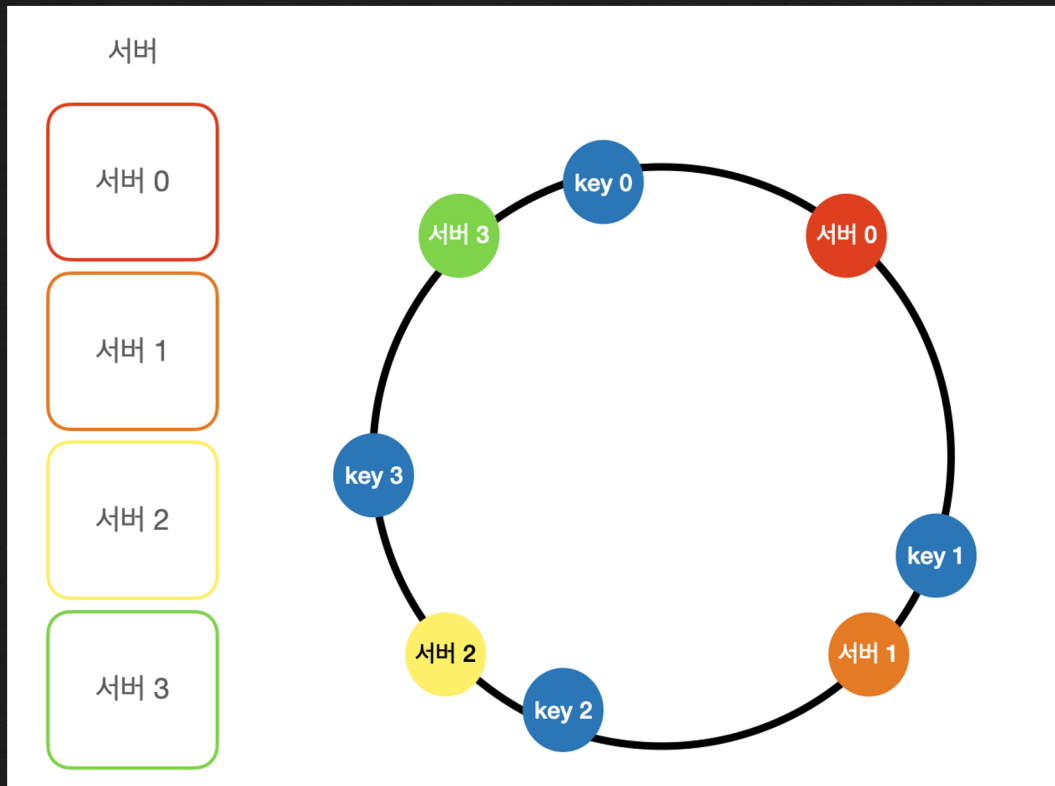
04 Consistent Hashing



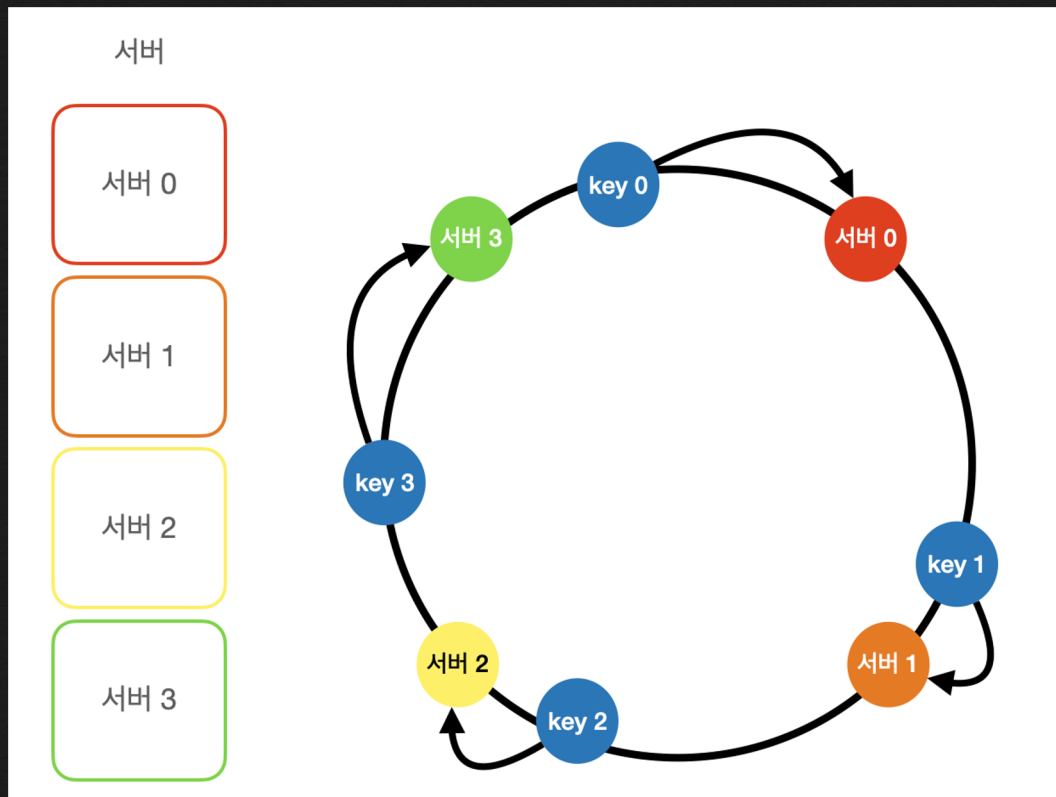
04 Consistent Hashing



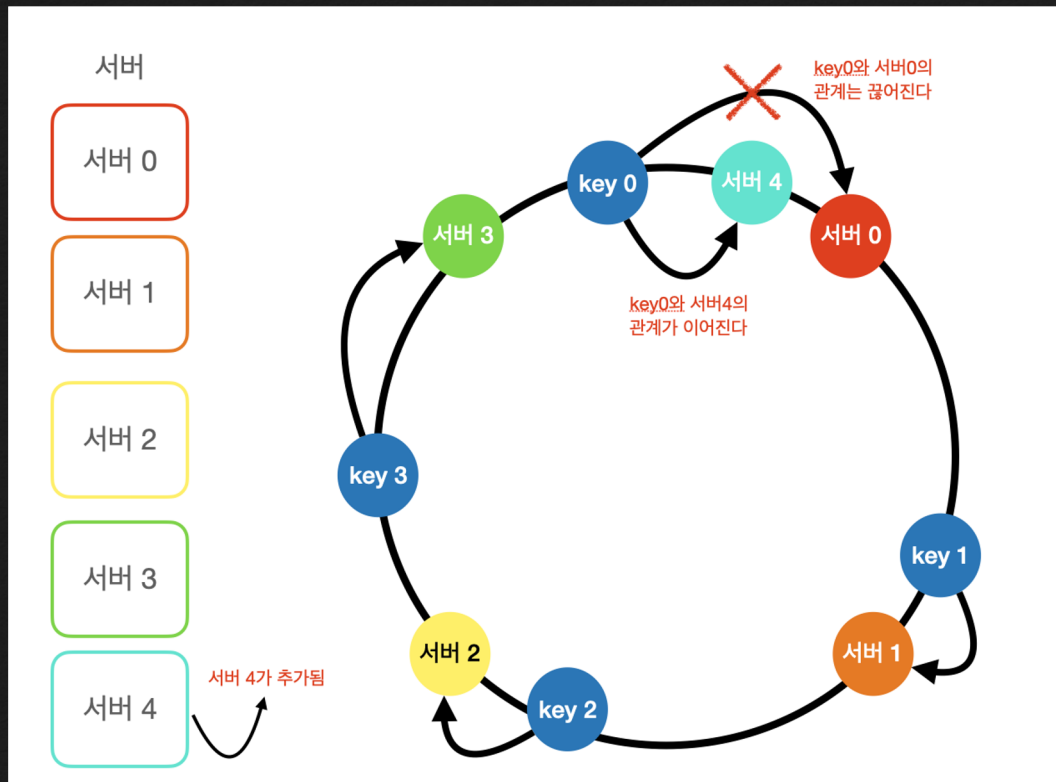
04 Consistent Hashing



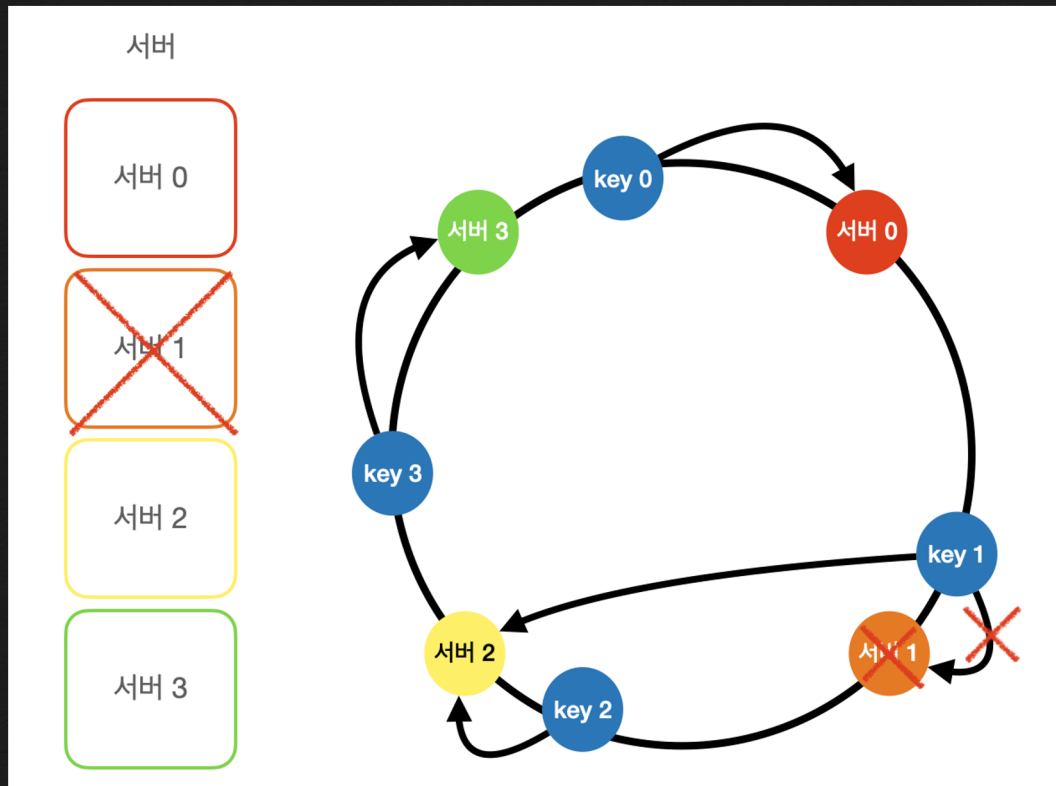
04 Consistent Hashing



04 Consistent Hashing



04 Consistent Hashing



목차

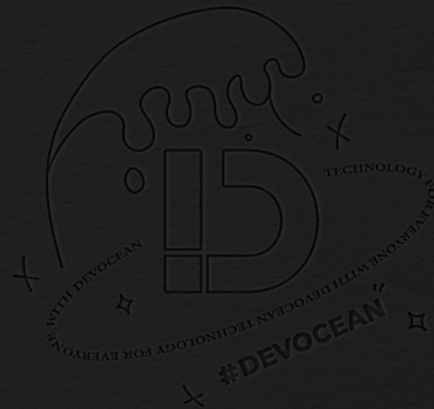
01 Server

02 Database

03 Key-value Store

04 Consistent Hashing

05 GUID



목차

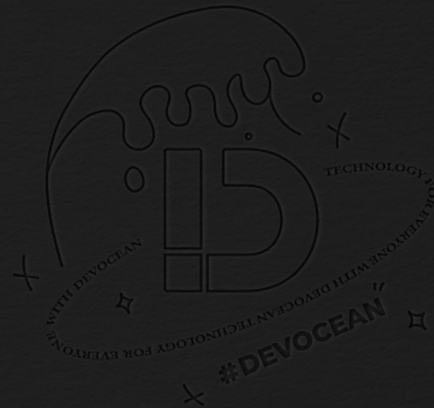
01 Server

02 Database

03 Key-value Store

04 Consistent Hashing

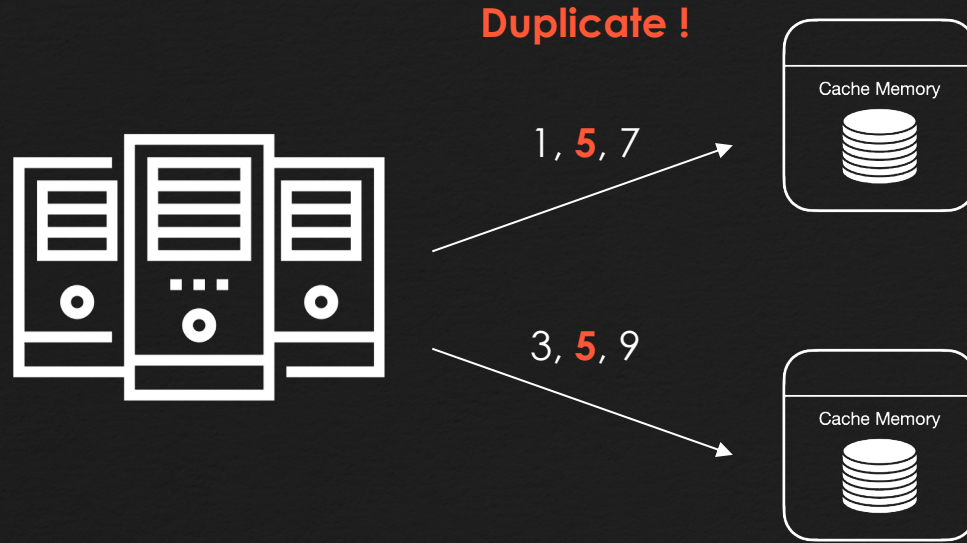
05 GUID



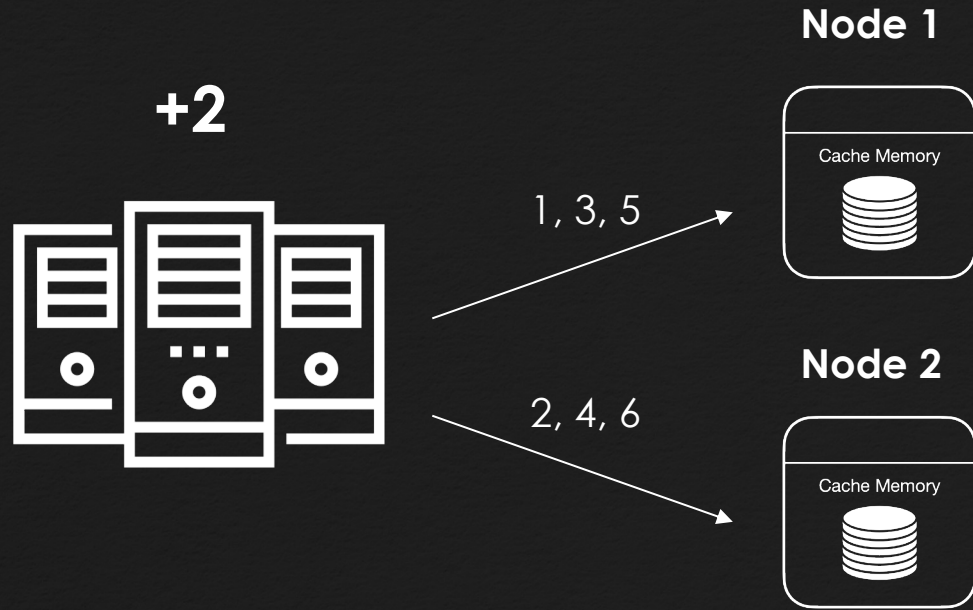
05 GUID



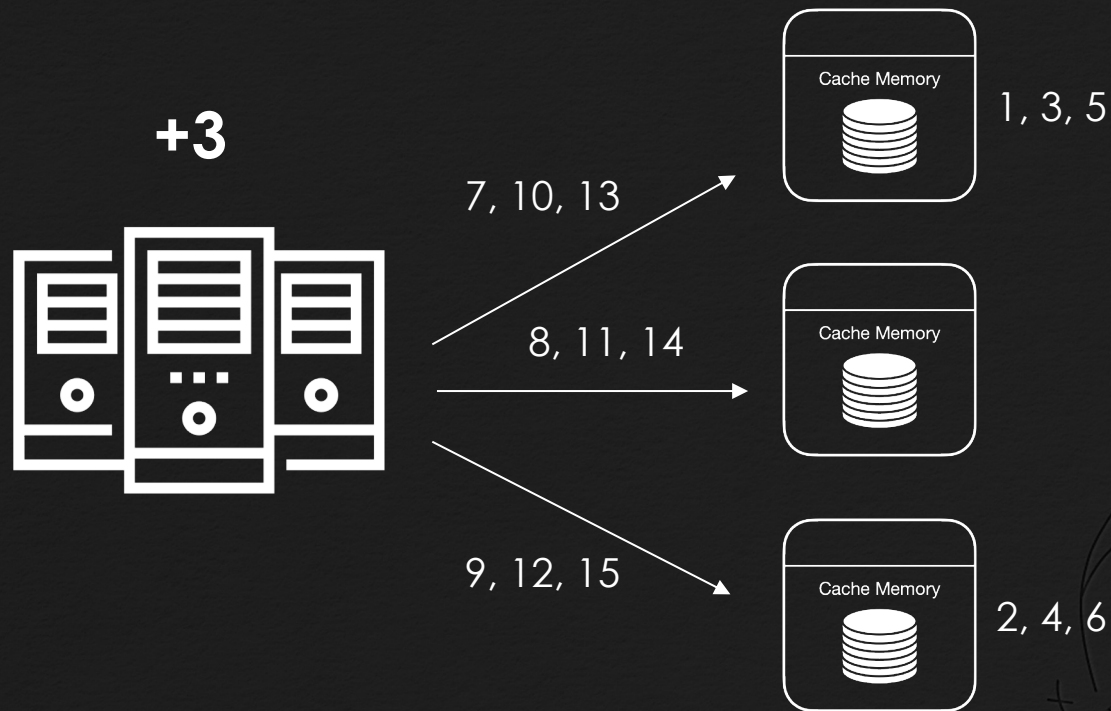
05 GUID



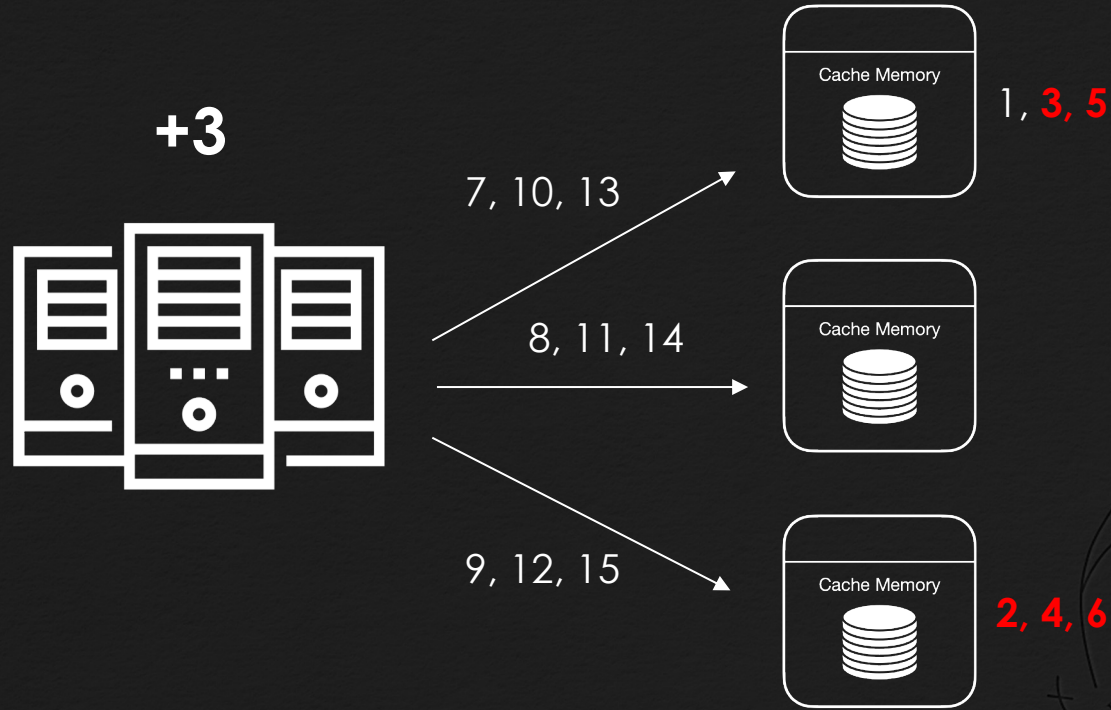
05 GUID



05 GUID



05 GUID



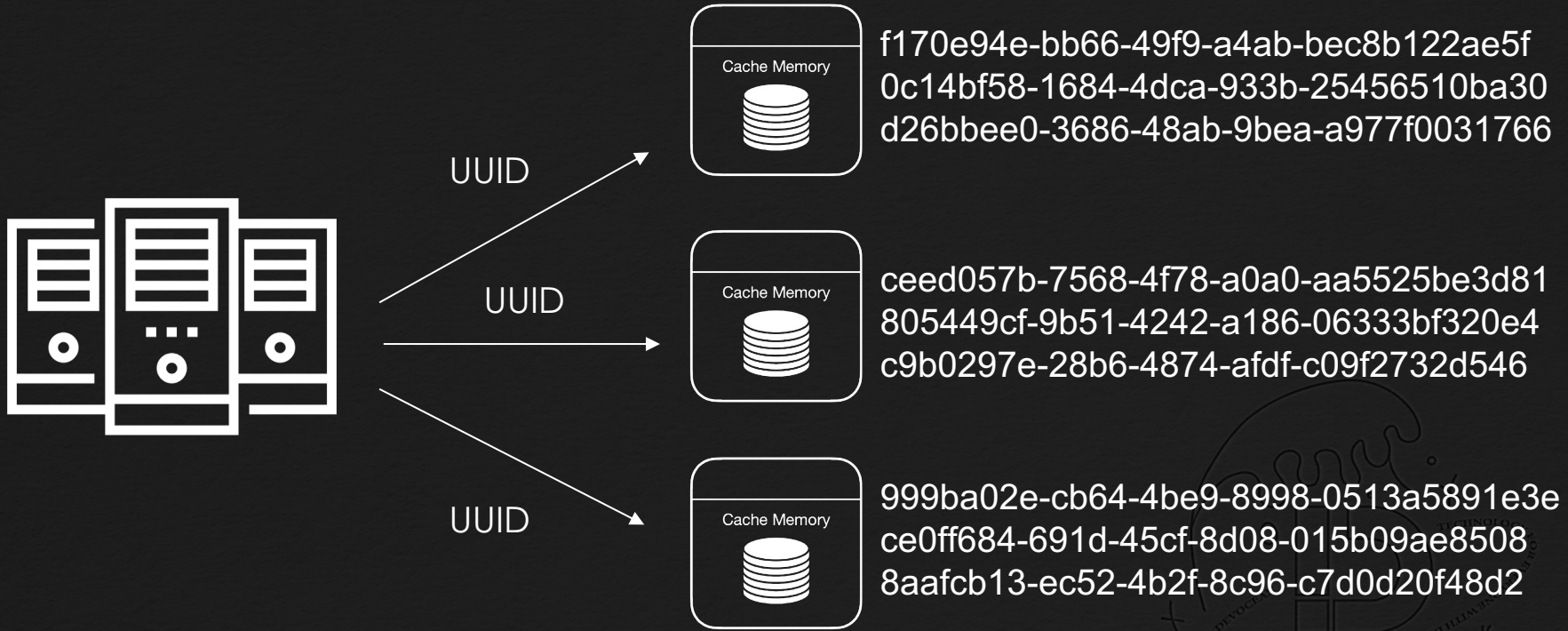
05 GUID

UUID v4

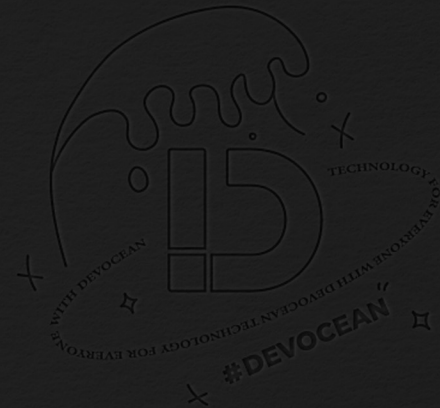
4243f599-229a-43e6-9392-02a529f6e98b



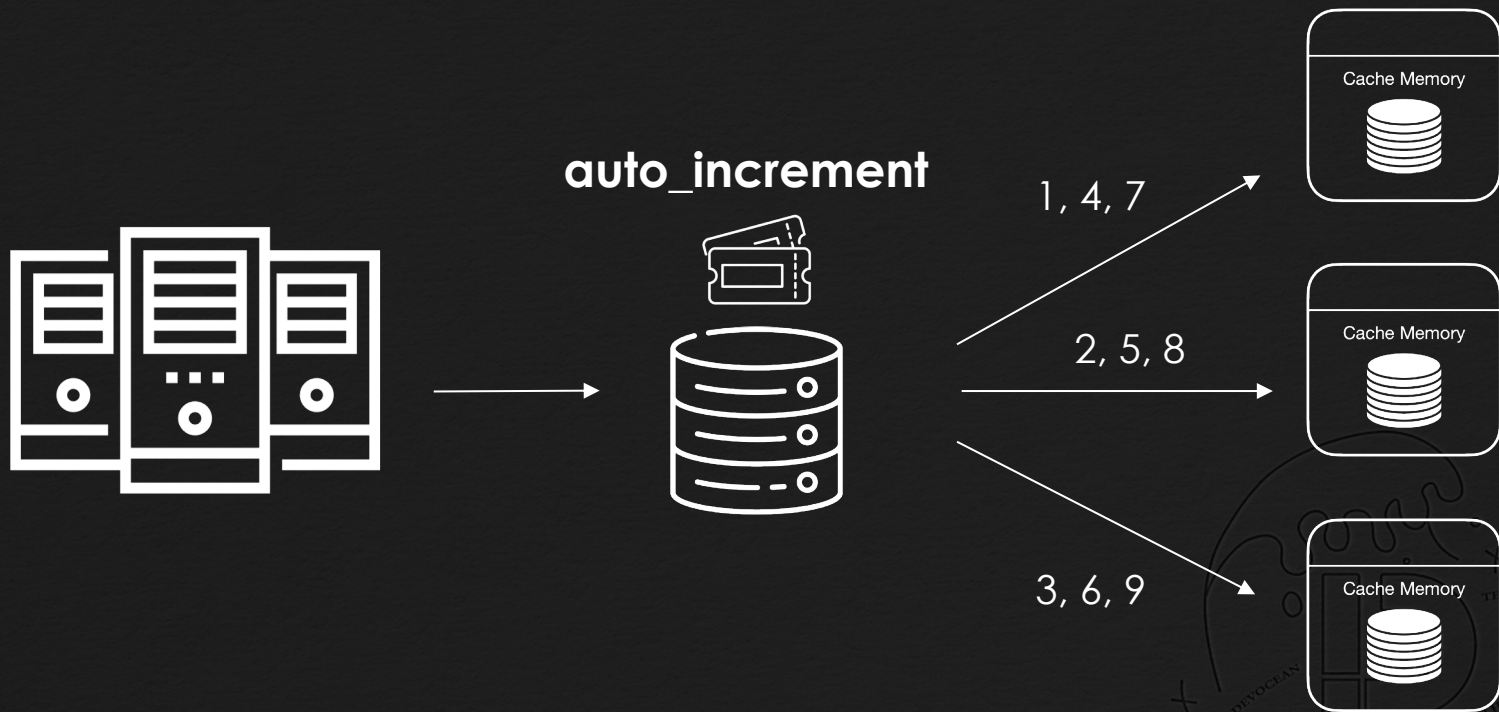
05 GUID



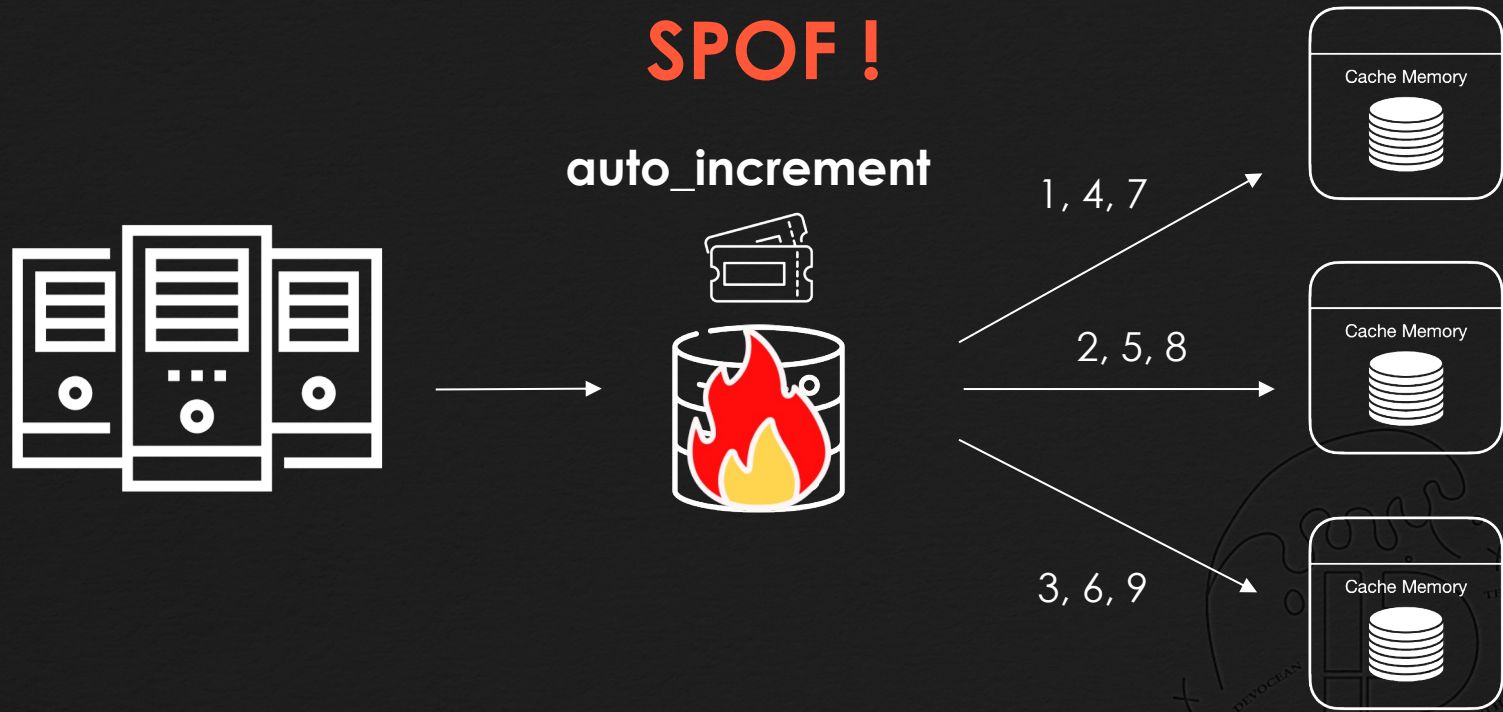
Ticket Server



05 GUID



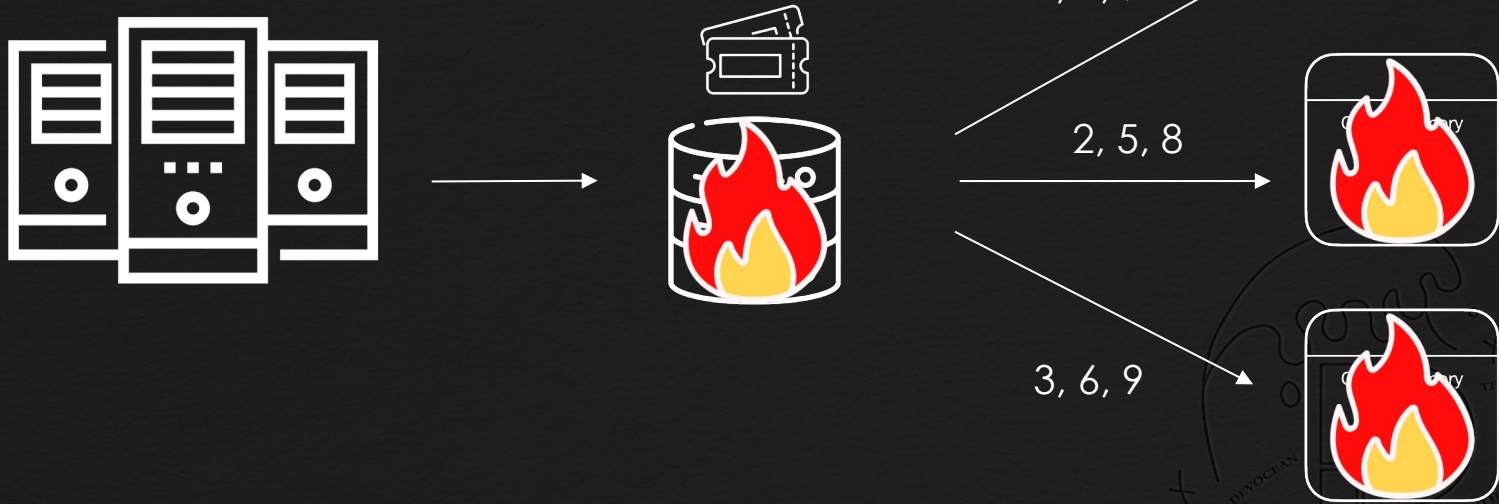
05 GUID



05 GUID

SPOF !

auto_increment



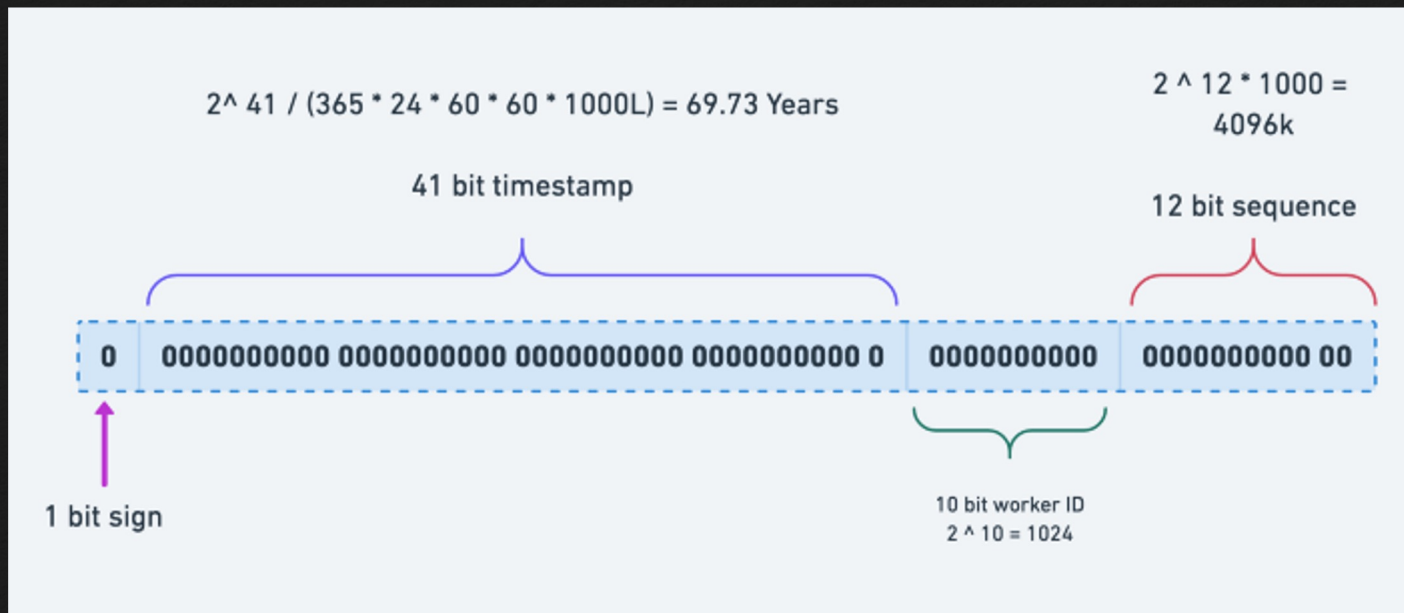
05 GUID

GUID

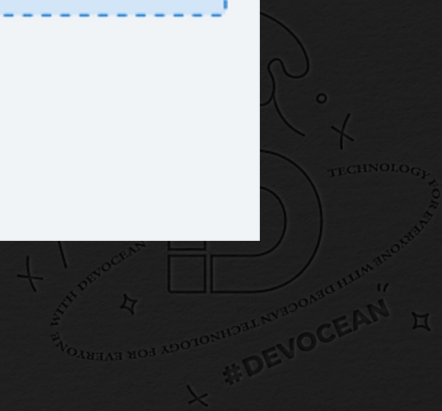
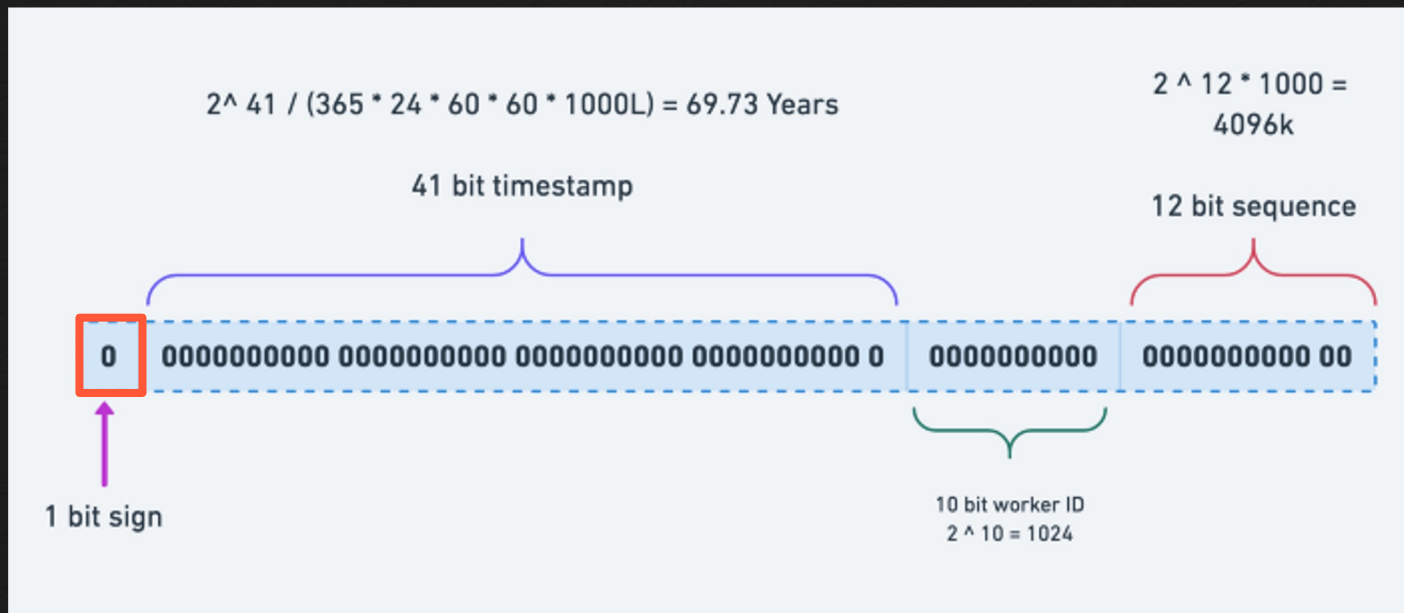
(Globally Unique Identifier)



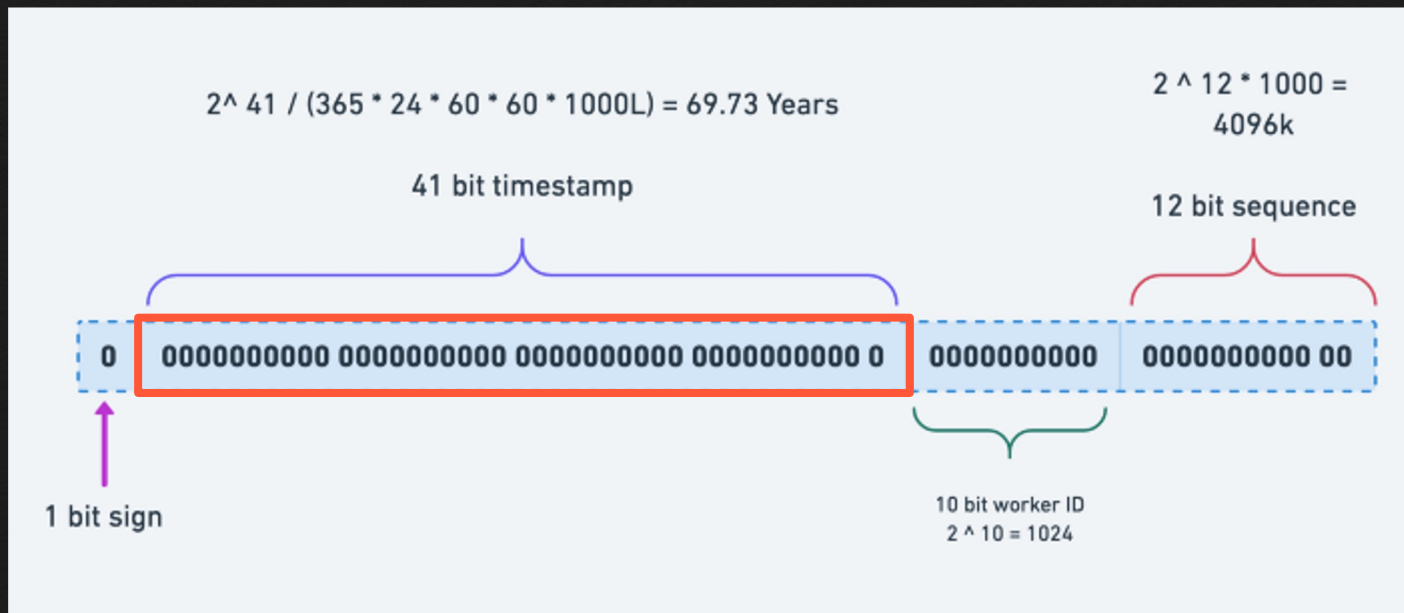
05 GUID



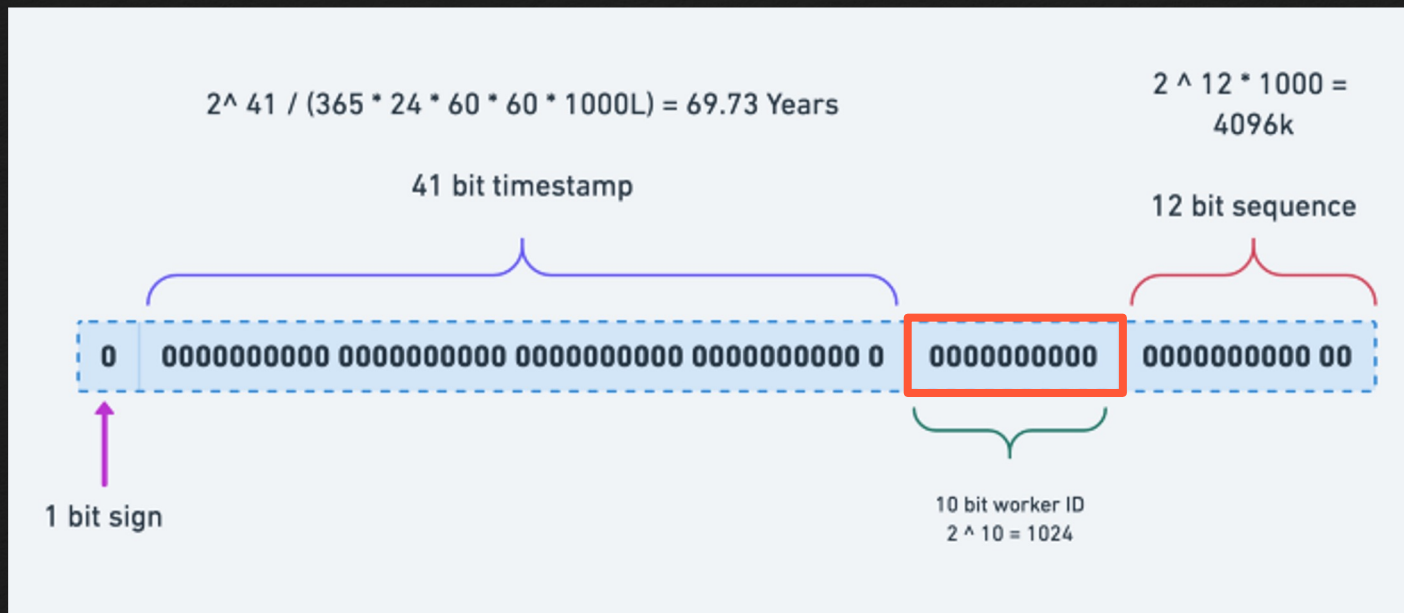
05 GUID



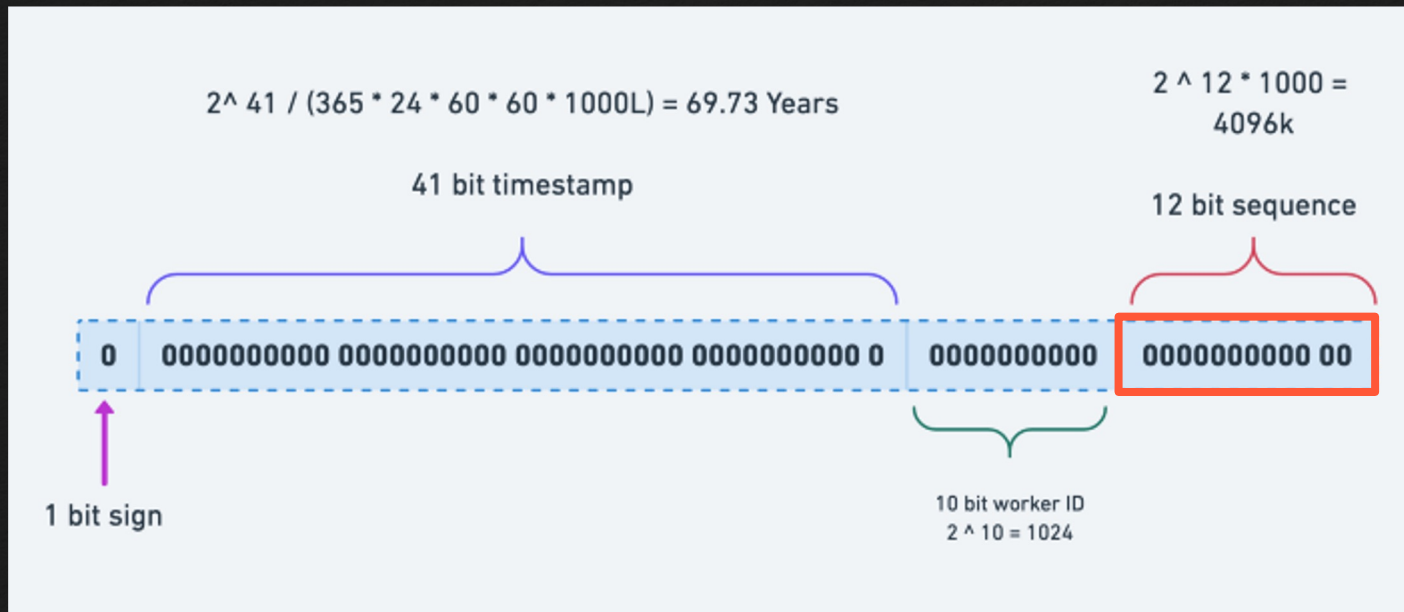
05 GUID



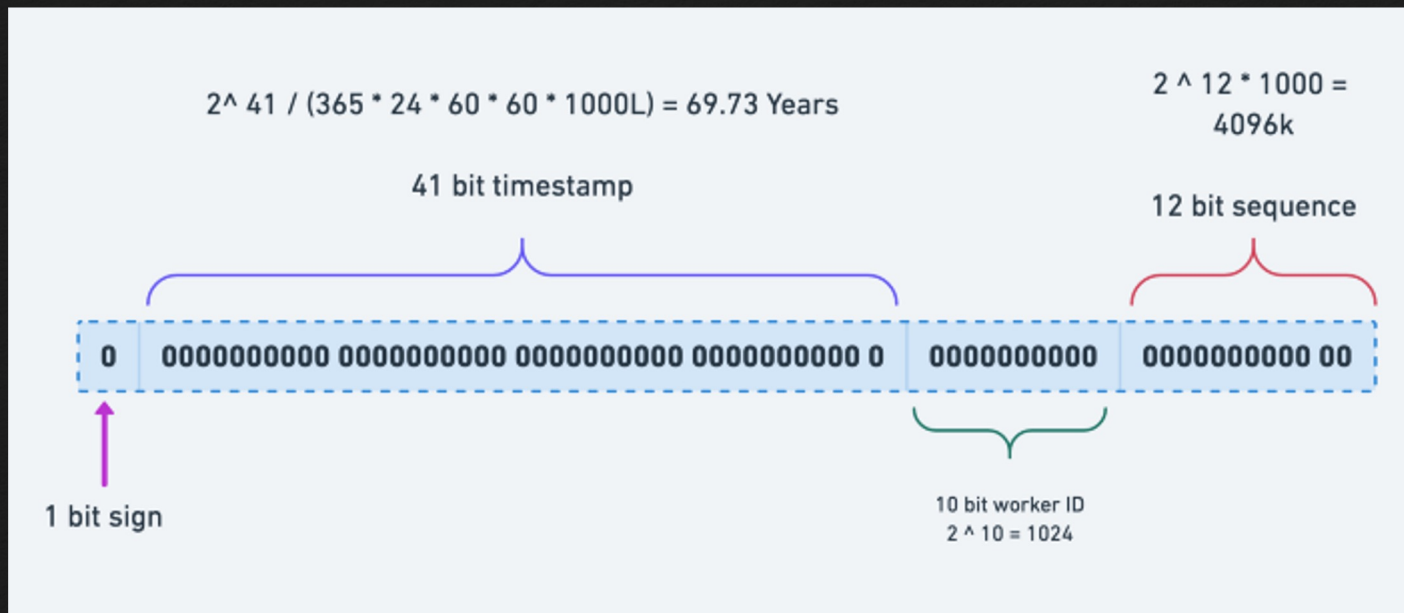
05 GUID



05 GUID



05 GUID



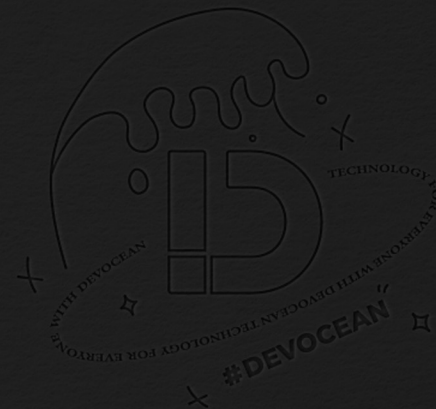
06 Next Step

GUID

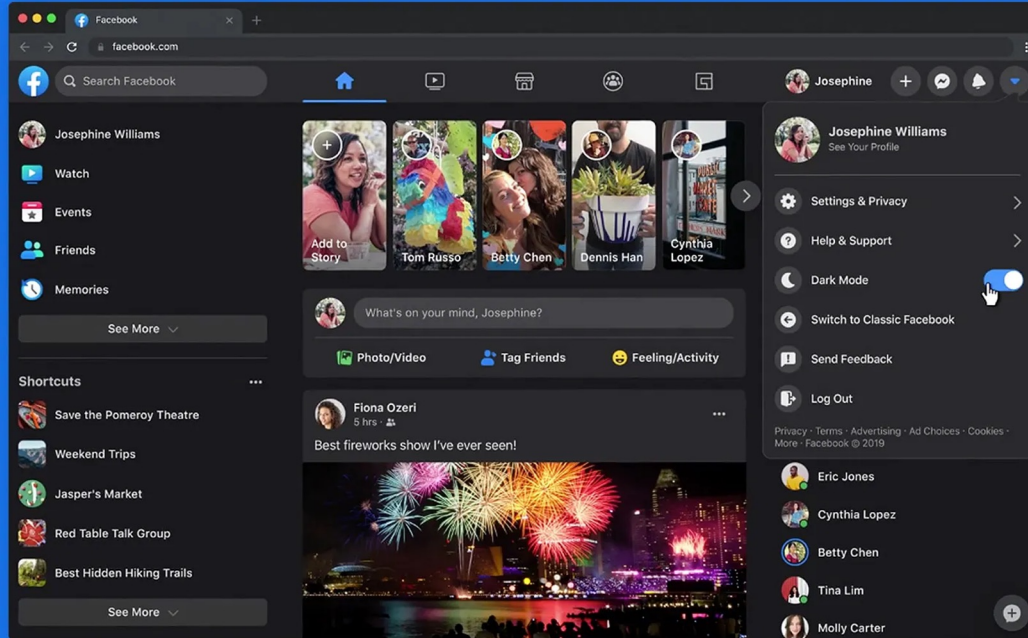
(Globally Unique Identifier)



06 Next Step



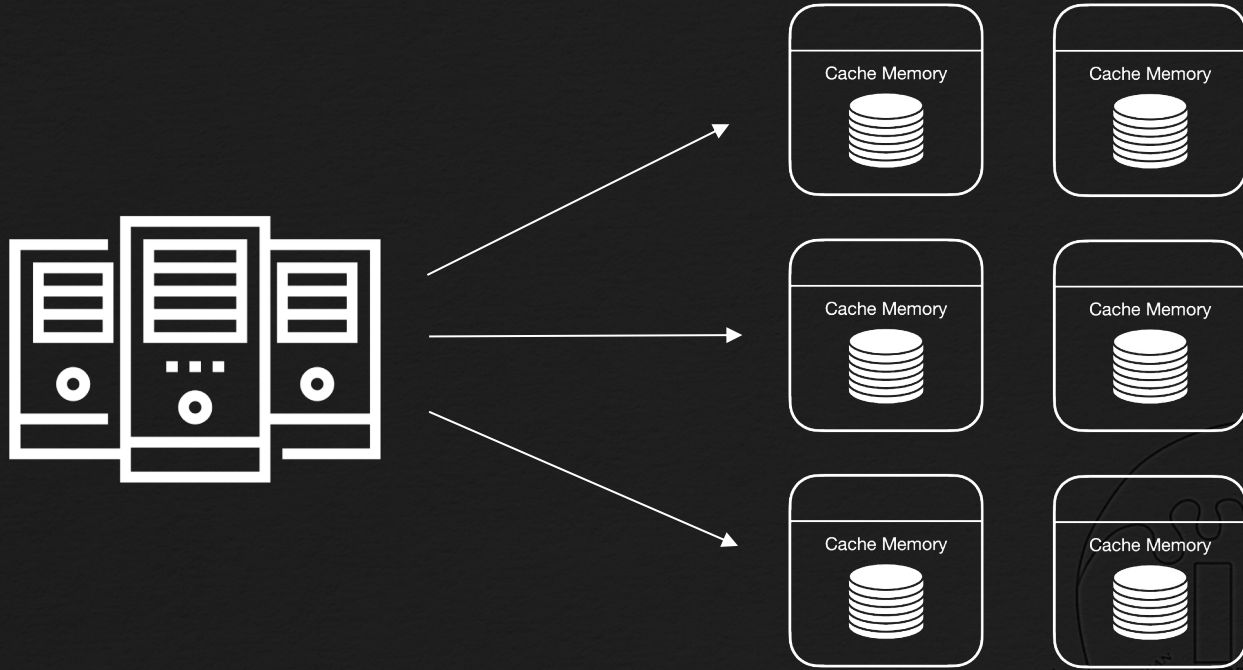
06 Next Step



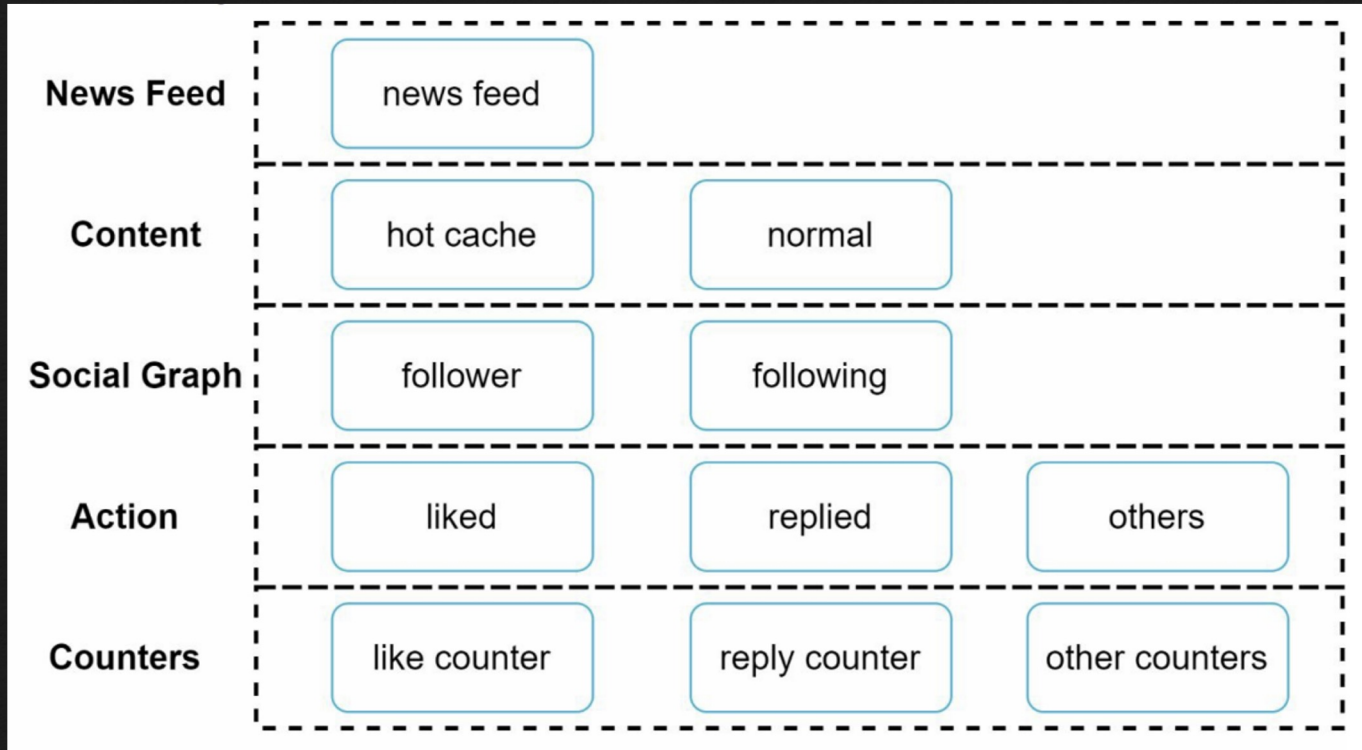
06 Next Step



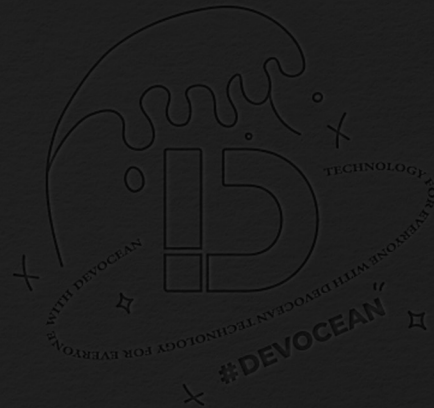
06 Next Step



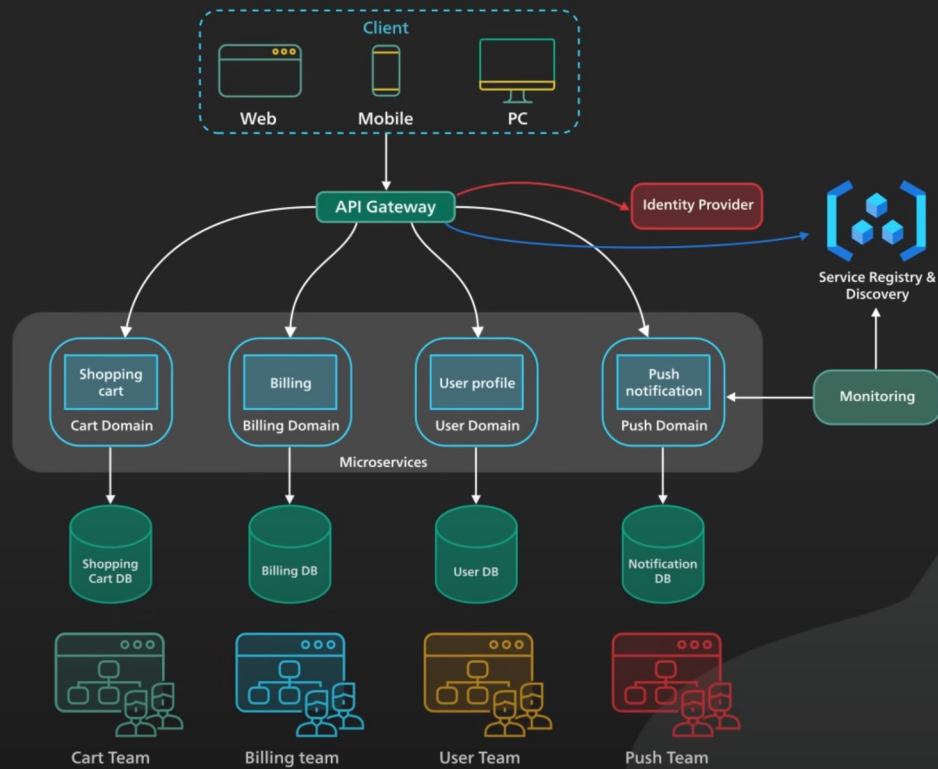
06 Next Step



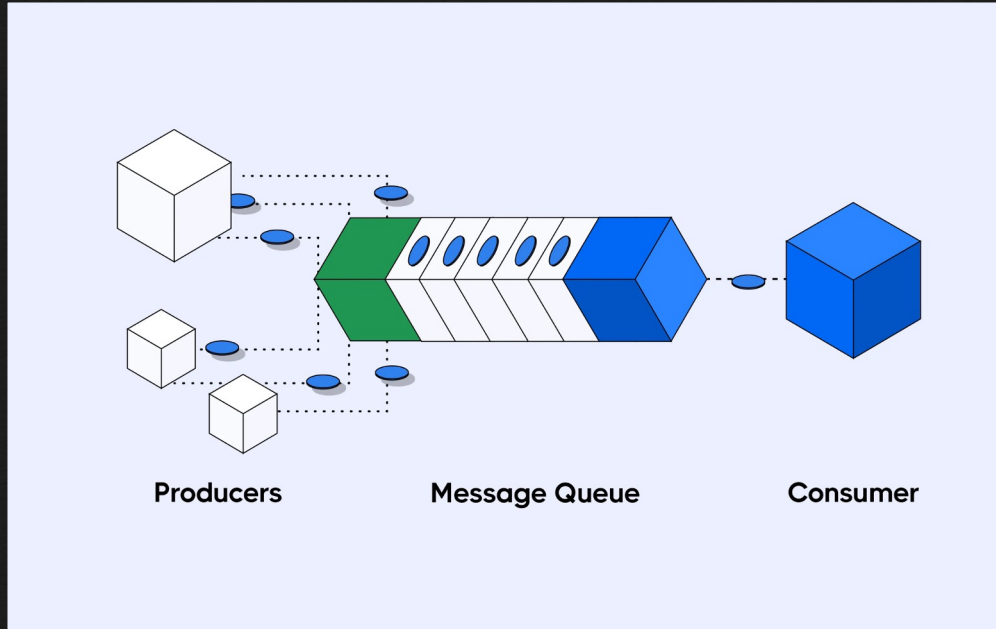
대규모 시스템



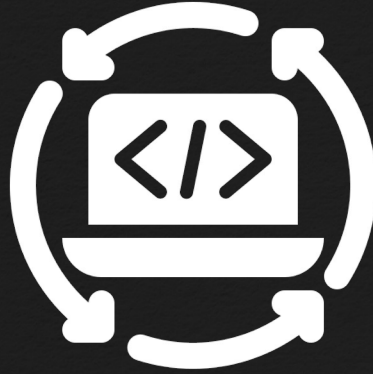
06 Next Step



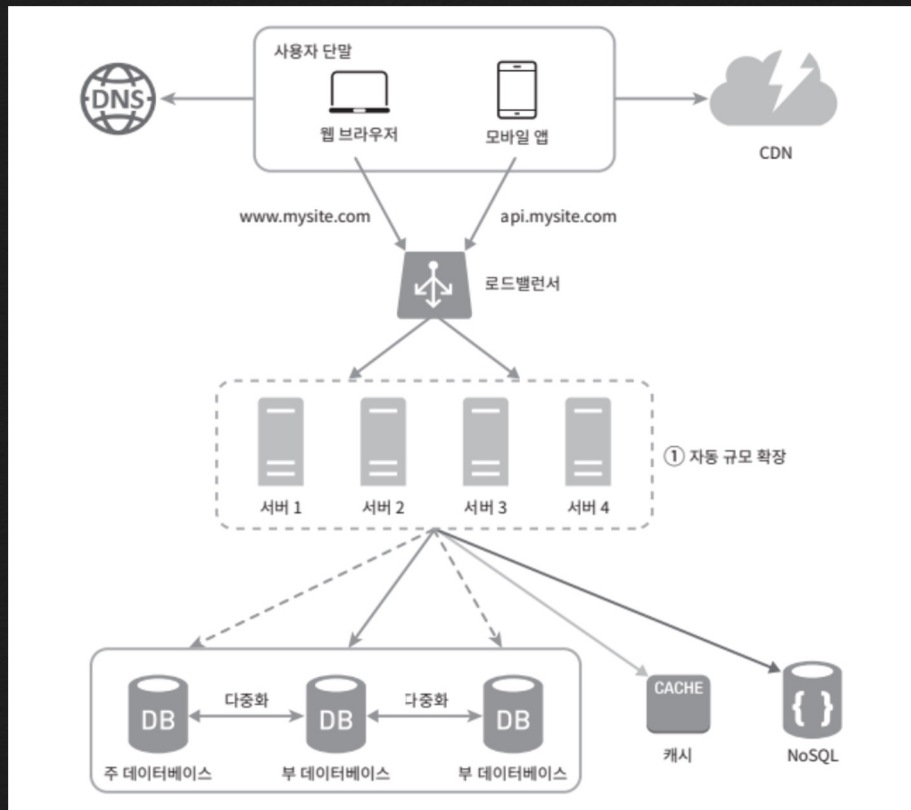
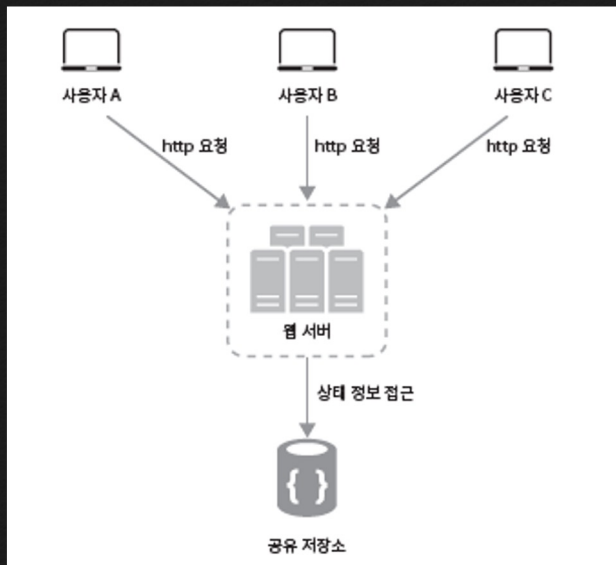
06 Next Step



06 Next Step



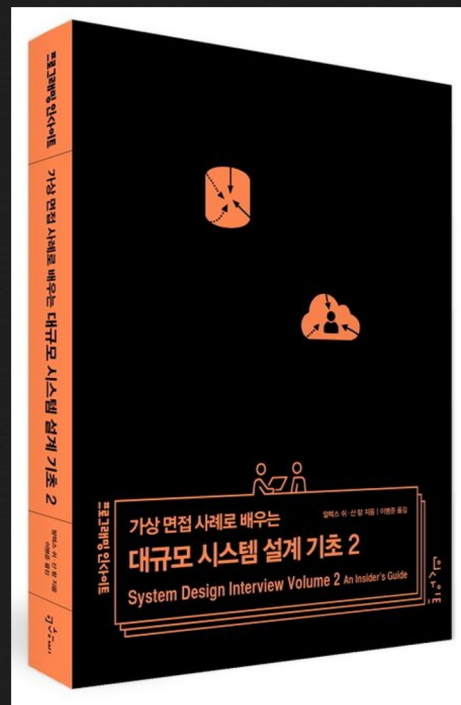
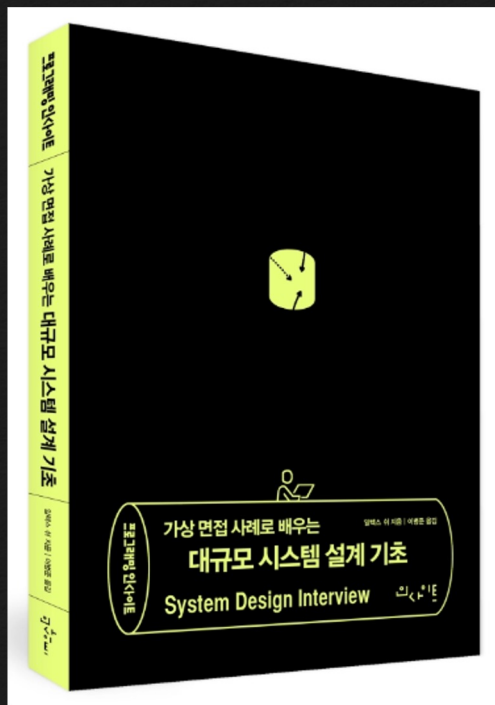
06 Next Step



06 Next Step



06 Next Step



06 Next Step

choiys0311@gmail.com

