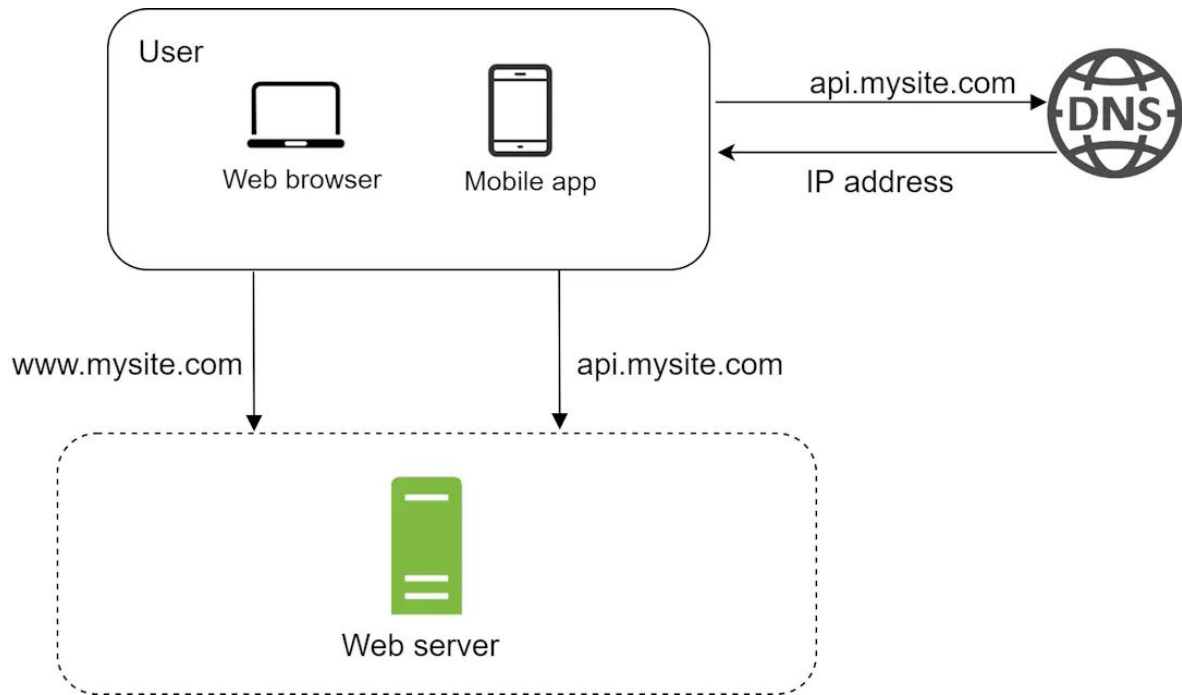




대규모 시스템 설계

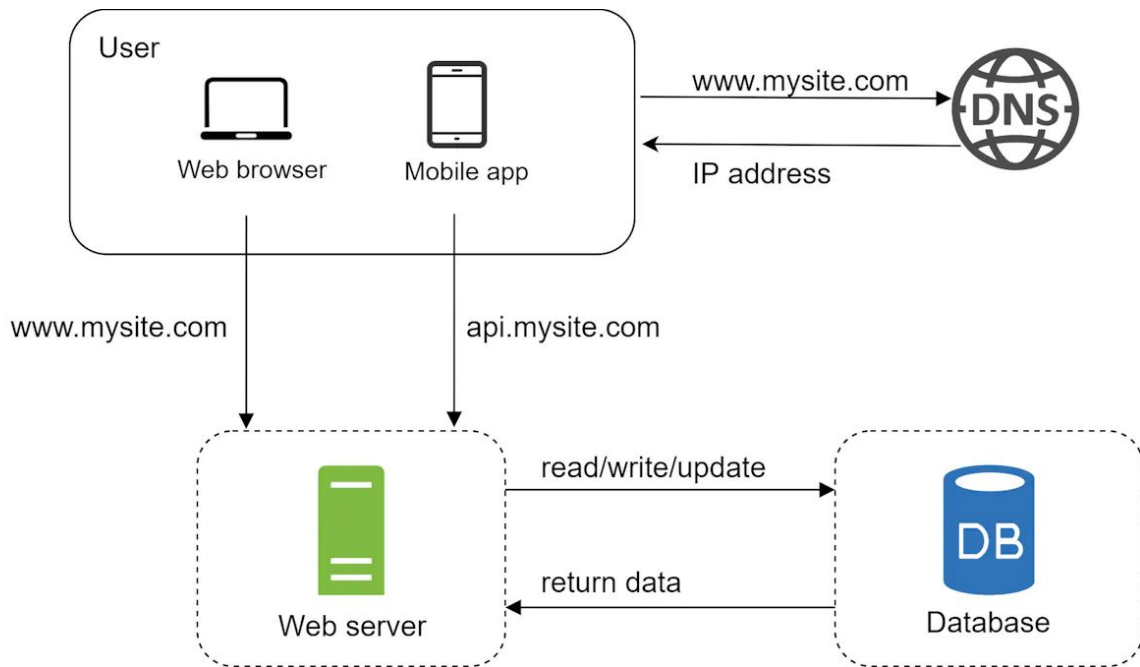
천만 사용자 트래픽 처리하기

단일서버에서 시작



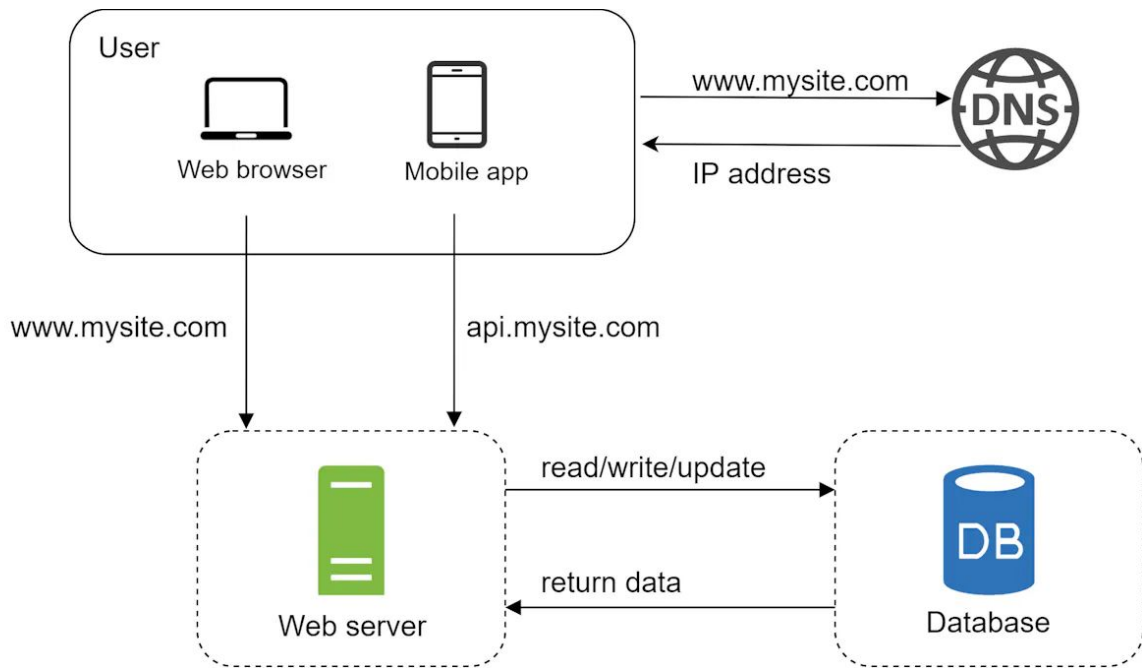
- 가장 간단한 구성 방식
- HTTP사용. 서버는 HTML 혹은 JSON 응답을 반환

데이터베이스 분리하기



- 늘어난 트래픽을 처리하기 위해서 역할을 나누자.
- 웹, 모바일 트래픽 처리 용도서버 데이터베이스 처리서버를 분리

데이터베이스 분리하기



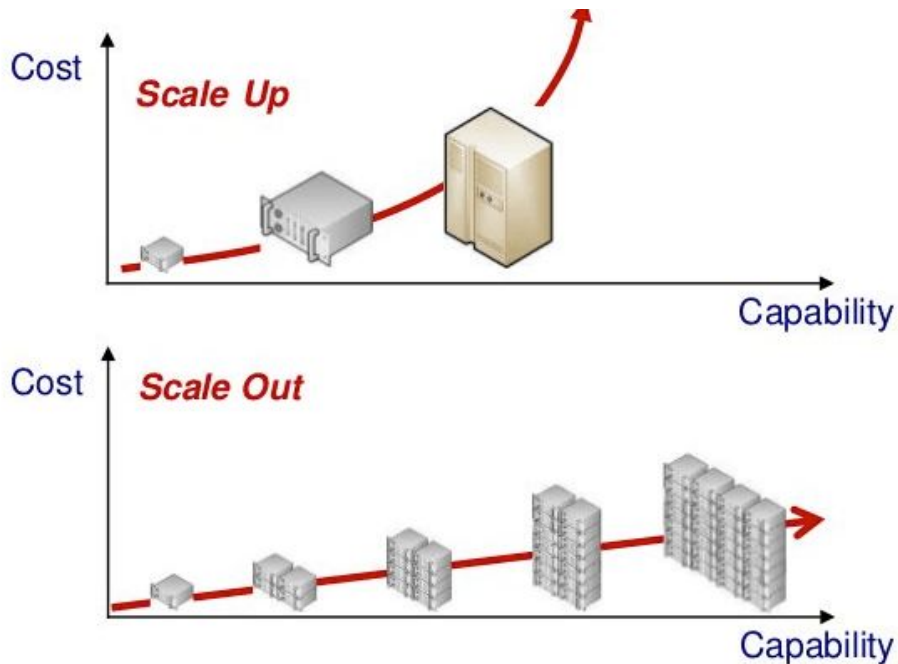
- 늘어난 트래픽을 처리하기 위해서 역할을 나누자.
- 웹, 모바일 트래픽 처리 용 웹서버 그리고 데이터베이스 처리서버를 분리

어떤 Database를 사용해야 할까?

#	RDBMS	NoSQL
특징	조인연산, 트랜잭션 지원	조인연산 지원 X
용도	정형화된 데이터 처리	비정형 데이터, 특수목적 데이터
예시	MySQL, PostgreSQL	CouchDB, Amazon DynamoDB

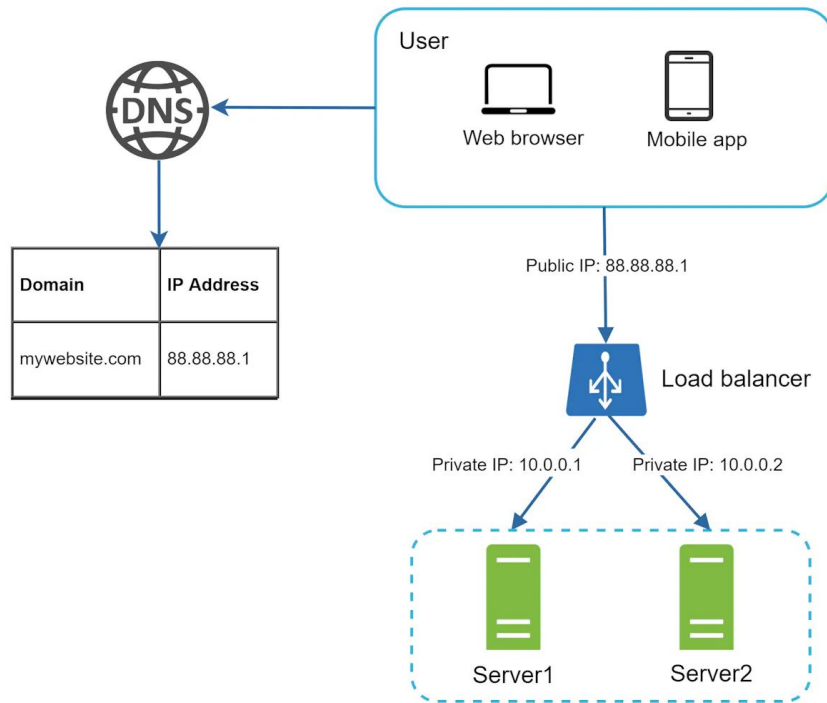
- 대부분의 경우 RDB가 최선. 범용적인 요구사항을 처리할 수 있음
- 대다수 개발자가 이해하며 가장 익숙하게 사용할 수 있음

웹서버 수평확장 vs 수직확장



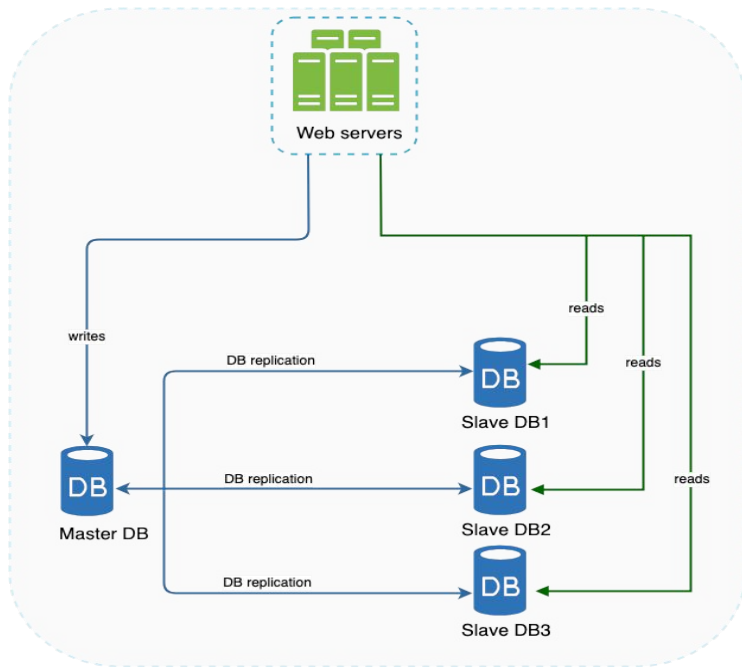
- 수직확장(Scale Up)은 서버의 CPU, Memory를 늘림. 단순하지만 자원을 무한대로 증설할 수 없음
- 수평확장(Scale Out)은 서버를 추가하여 대응. 대규모 트래픽에 맞추어 비용 효율적 대응 가능

로드밸런서(load balancer)

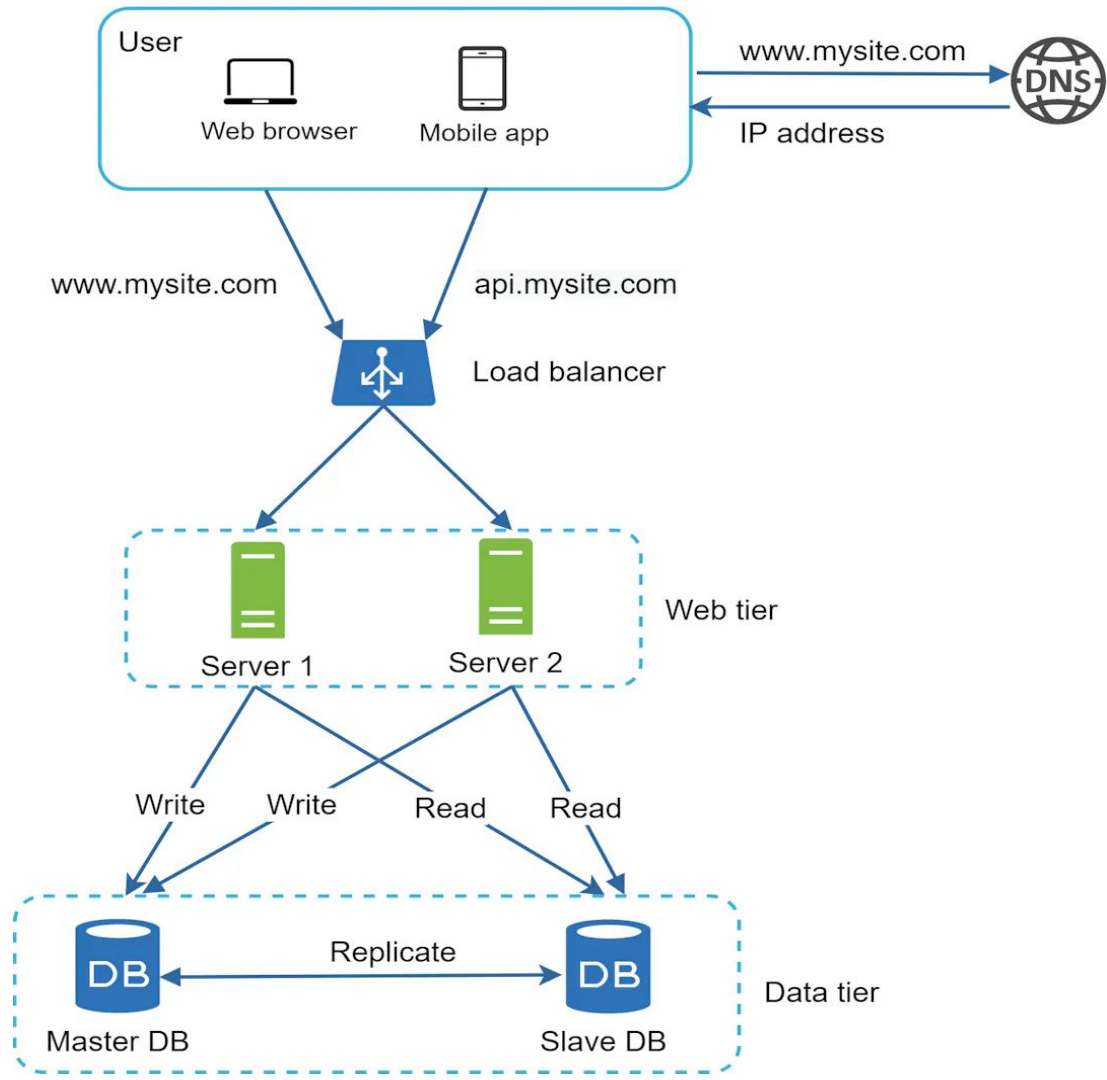


- 웹 서버들에게 트래픽 부하를 고르게 분산하는 역할
- 자동적으로 트래픽을 분산하며, 만약 서버1이 다운되면, 모든 트래픽은 서버2로 전송

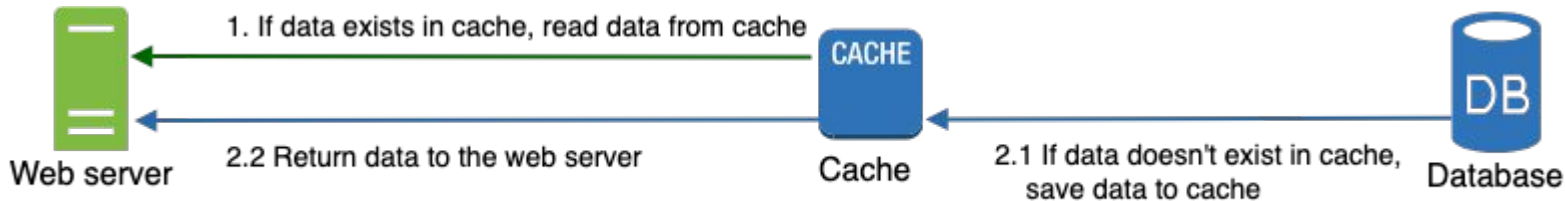
데이터베이스 다중화



- DB의 주(master)-부(slave) 관계를 설정 데이터 원본은 주 서버에, 사본은 부 서버에 저장
 - 주 서버 : 쓰기 연산, insert, delete, update
 - 부 서버 : 주 서버의 사본을 전달받고, 읽기 연산만 지원

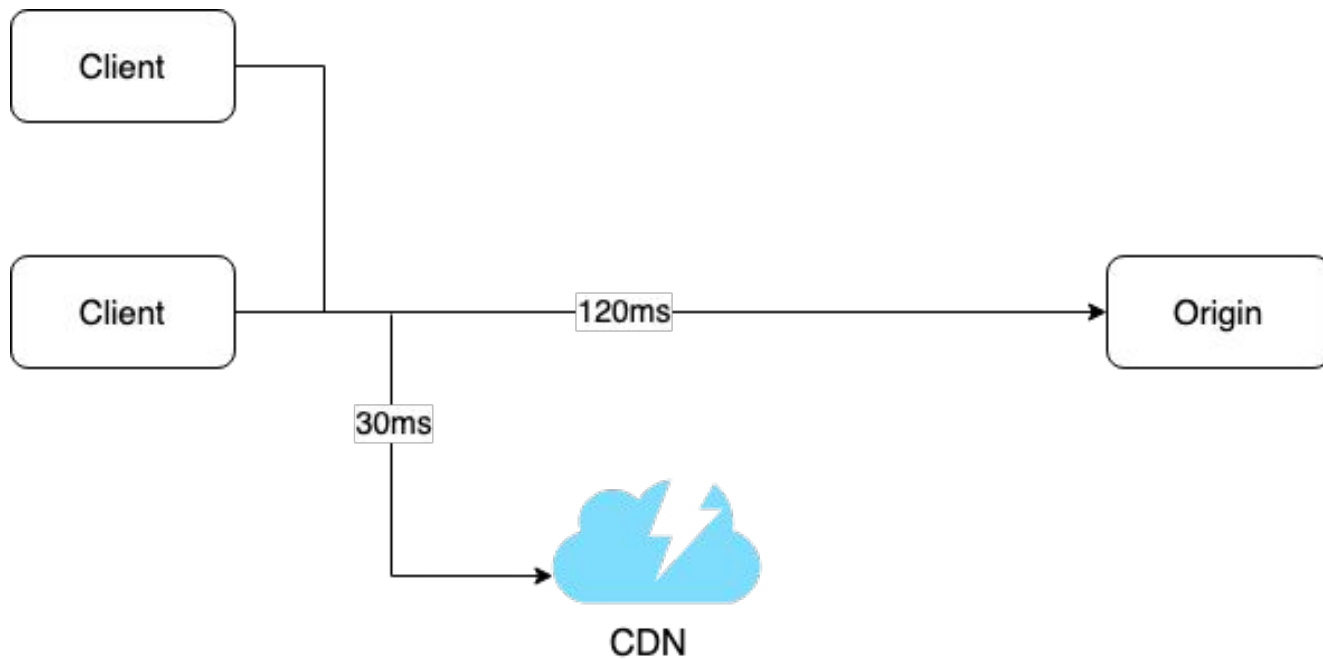


캐시(Cache)



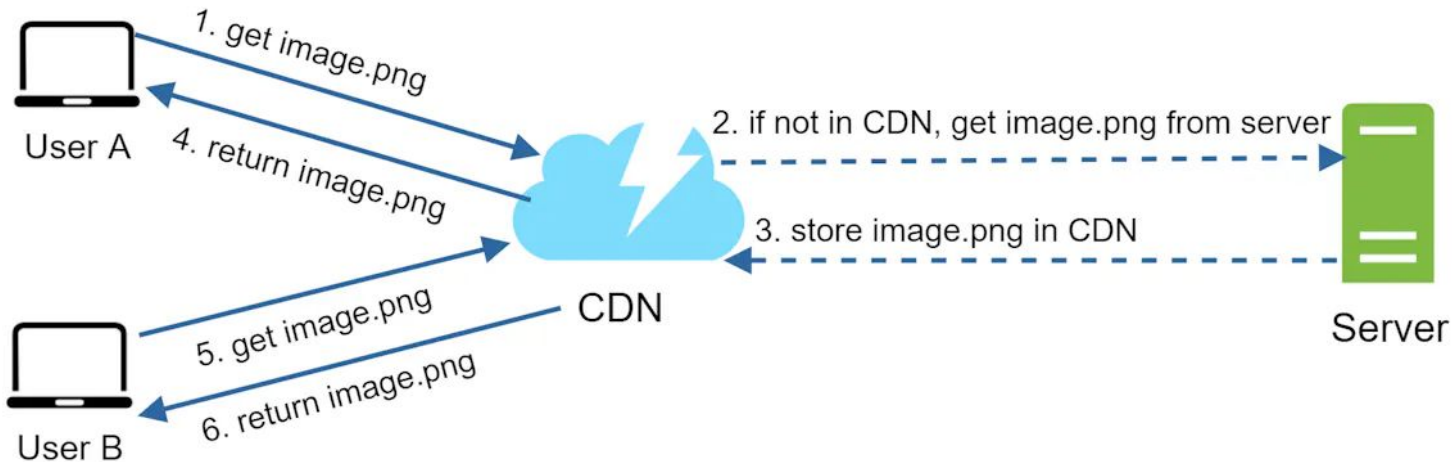
- 실행이 오래걸리는 연산 또는 자주 참조되는 데이터를 메모리 적재
- 데이터베이스 부하를 줄일 수 있음
- 캐시 계층은 규모를 독립적으로 확장 가능

콘텐츠 전송 네트워크(CDN)



- 정적 콘텐츠를 전송하는 데 쓰임. **지리적**으로 분산된 서버의 네트워크.

콘텐츠 전송 네트워크(CDN)



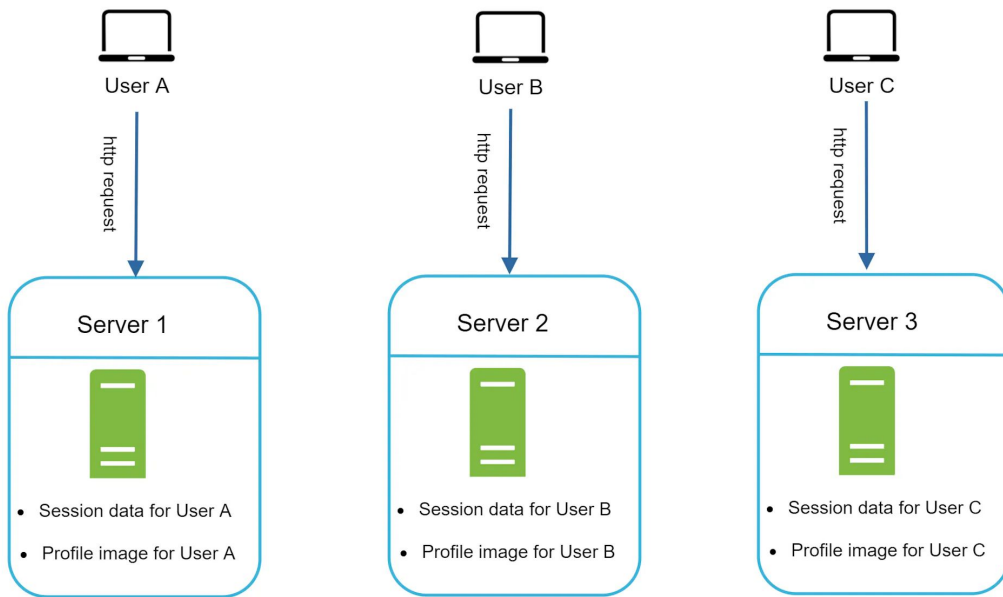
사용자 A 가 이미지 URL으로 image.png에 접근한다. (URL 도메인은 CDN 서비스 사업자가 제공)

CDN 서버의 캐시에 해당 이미지가 없는 경우, 서버는 원본 서버에 요청하여 파일을 가져온다.

원본 서버가 파일을 CDN 서버에 반환. (응답 HTTP header에 파일이 언제까지 캐시될 수 있는지 TTL(Time-To-Live)값 존재)

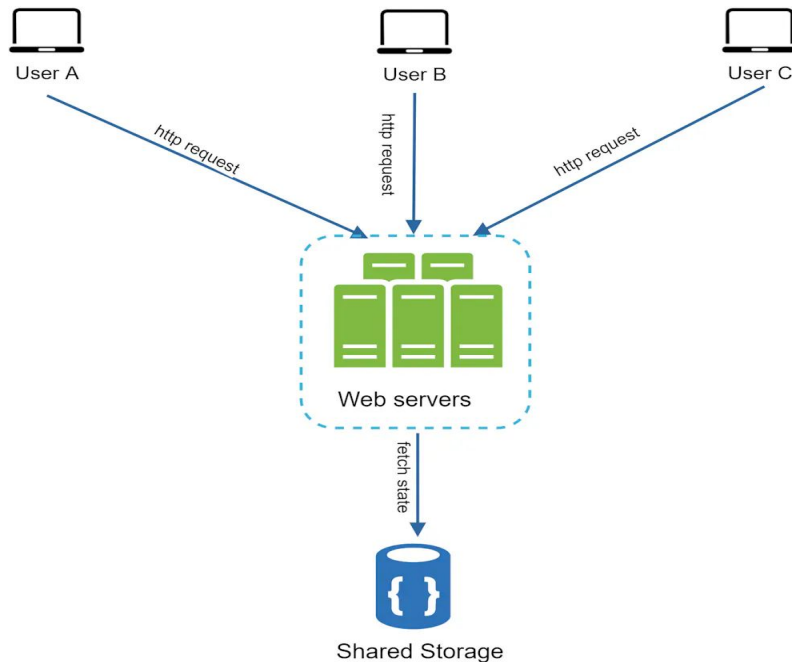
CDN 서버는 파일을 캐시하고 사용자 A에게 반환. 이미지는 TTL에 명시된 시간까지 캐시

무상태(stateless) 웹 계층

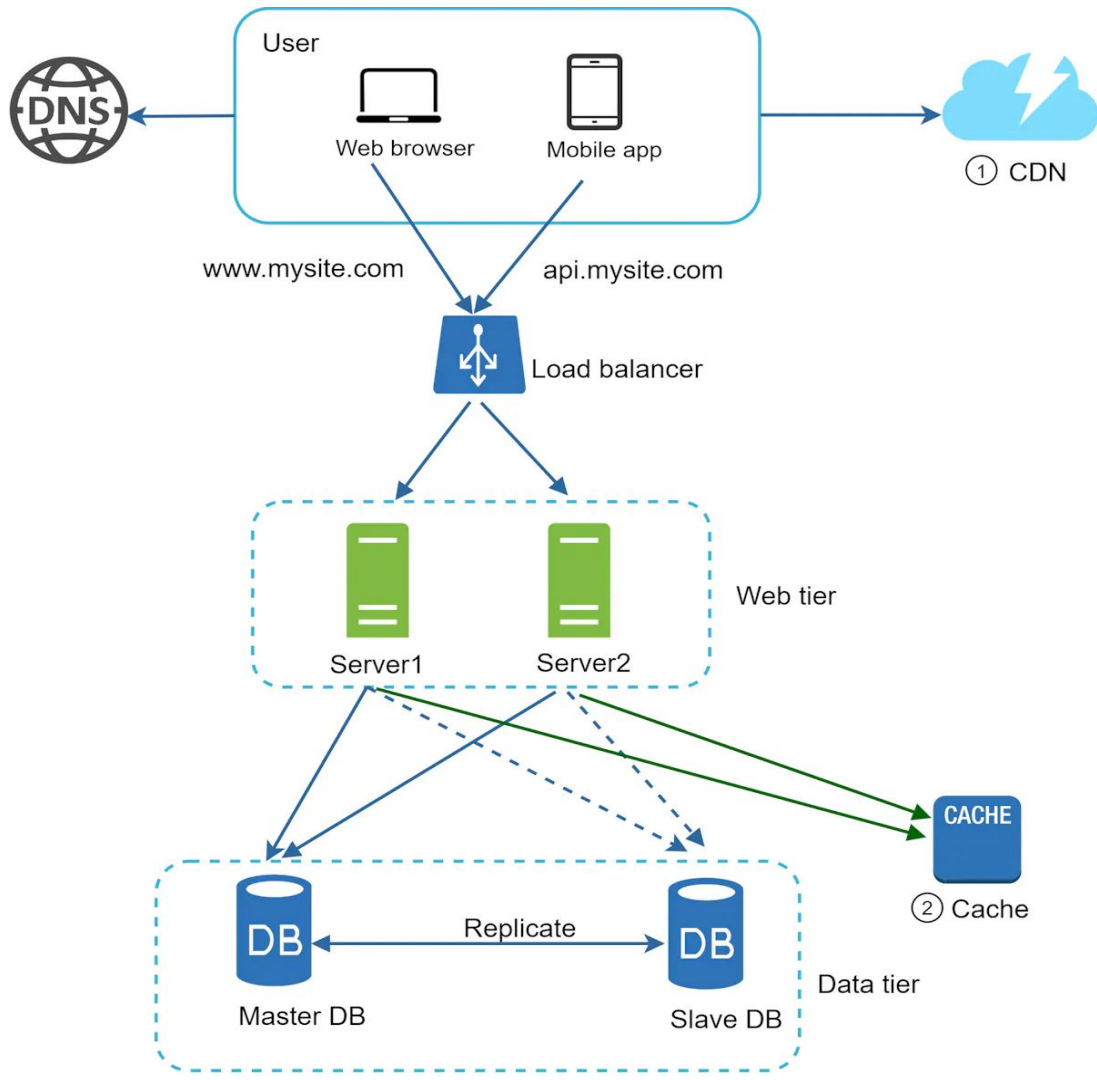


- 수평확장을 위해서 상태 정보(사용자 세션 데이터 등등)를 웹 계층에서 제거 필요
- 상태 정보를 RDBMS, NoSQL 같은 지속성 저장소에 보관

무상태(stateless) 웹 계층

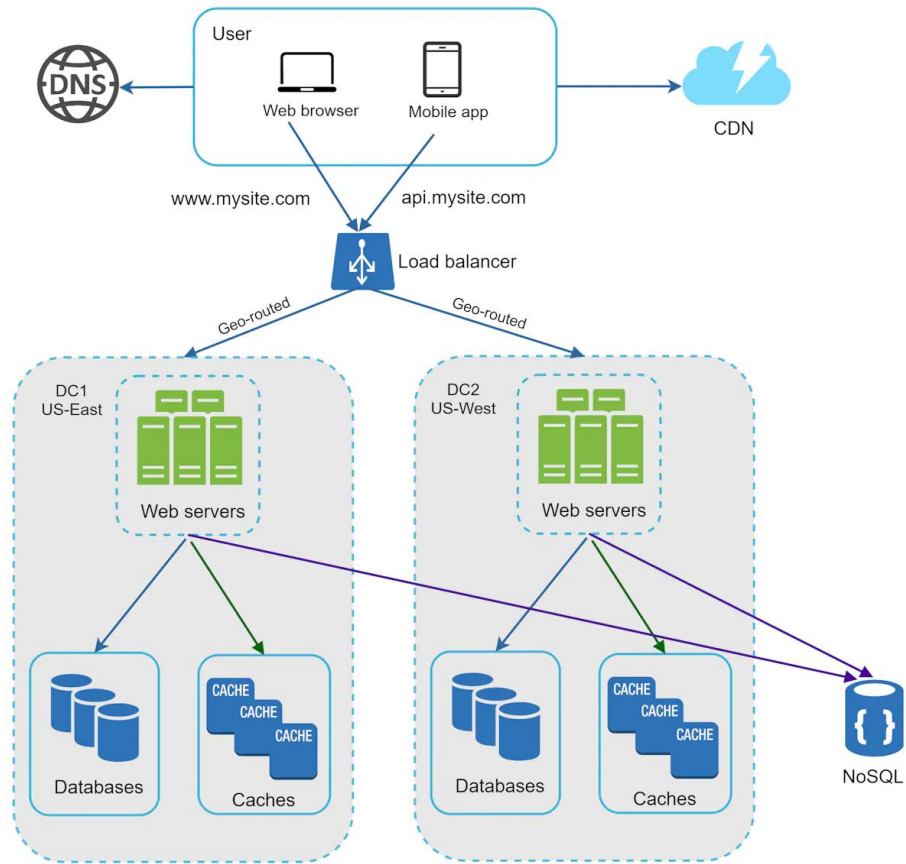


- 수평확장을 위해서 상태 정보(사용자 세션 데이터 등등)를 웹 계층에서 제거 필요
- 상태 정보를 RDBMS, NoSQL 같은 지속성 저장소에 보관

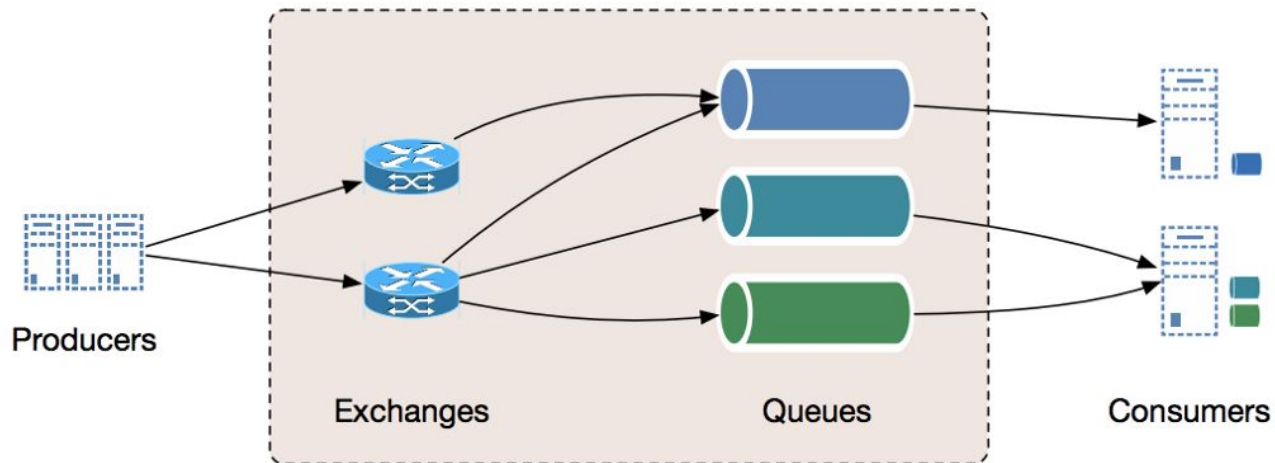


데이터 센터

- 다른 지역의 사용자도 쾌적하게 접근
- geoDNS는 사용자의 위치를 인지
 - 가장 가까운 데이터센터로 안내
- 장애상황에서 안정성 확보 가능



메시지 큐(message queue)



- 시스템의 컴포넌트를 분리하여 각각 독립적으로 확장할 수 있어야 함
- 메시지의 무손실(durability)을 보장하는, 비동기 통신을 지원하는 컴포넌트
- 메시지의 버퍼 역할을 하며, 비동기적으로 전송

로그, 메트릭 그리고 자동화

로그

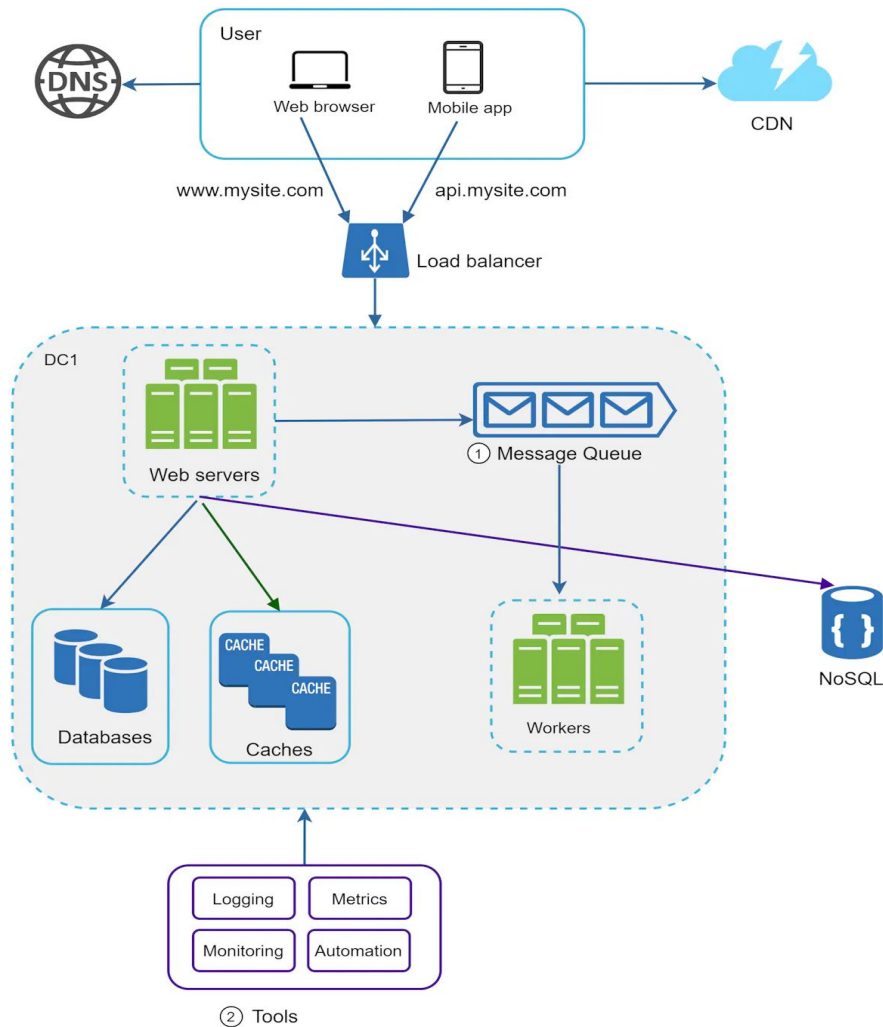
- 분산시스템의 오류와 문제를 파악

메트릭

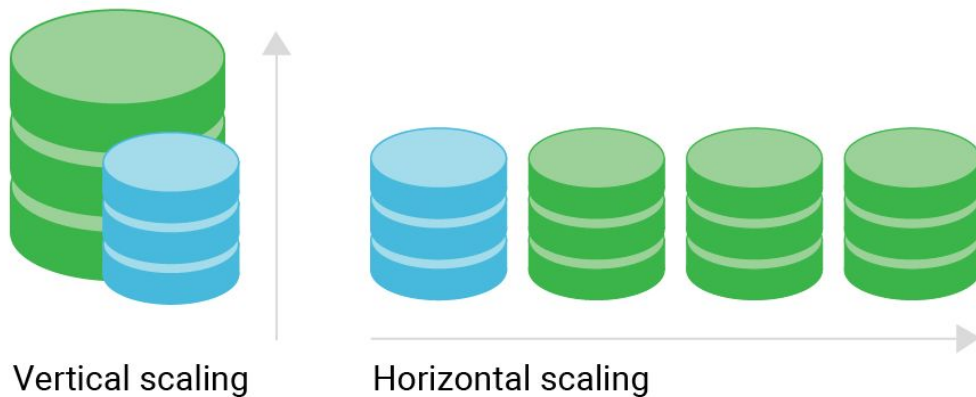
- 시스템의 상태파악
- 비즈니스 정보 획득

자동화

- 시스템 복잡성 대응 개발생산성 향상



데이터베이스 규모 확장



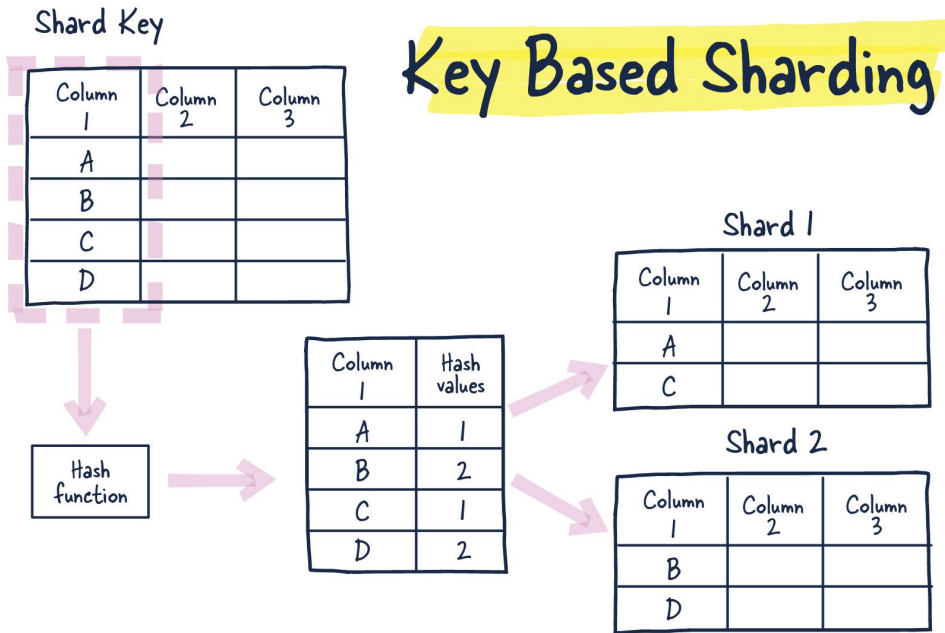
수직적 확장(Scale up)

- 고성능 자원(CPU, RAM, 디스크) 을 증설하는 방법

수평적 확장(sharding)

- 더 많은 서버를 추가해 성능 향상
- 샤딩 : 대규모 데이터베이스를 샤드(shard) 단위로 분할하는 기술
- 샤드는 같은 스키마를 사용하지만, 보관되는 데이터는 중복 X

데이터베이스 규모 확장



샤딩을 진행하는 경우 고민거리..

- 특정 샤드에 요청이 집중 (유명인사 문제)
- Join이 어려워짐