


분산서버 처리를 위한 기술들

▶ 로드 밸런싱(Load Balancing)

- 시스템에 대규모로 들어오는 요청(Request)을 연결된 여러 대의 서버로 나누어 부하를 분산하는 방식입니다.
-  밸런싱 장비에 IP가 부여가 되며, 이 로드밸런서에 연결되어 있는 N개의 장치 중 한곳으로 라우팅 합니다.
- 장애가 발생한 서버를 회피하여 연결하므로 이용할 수 없는 서버로 라우팅 되는 것을 방지합니다.

▶ 오토 스케일링(Auto Scaling)

- 서버의 부하를 체크하여 서버를 생성하는 방식입니다(미리 만들어놓은 가상 이미지로 서버를 생성).
- 대부분의 클라우드 서비스 제공 업체들이 오토 스케일링을 지원합니다.

▶ 데이터베이스 샤딩(Database Sharding)

- DB 테이블을 수평 분할(horizontal partitioning)하여 물리적으로 서로 다른 곳에 분산하여 관리합니다.
- 대부분의 DB가 샤딩을 통한 scale-out 을 지원합니다.

▶ 데이터베이스 레플리카(Database Replica)

- 보통 Master 서버에만 쓰기 작업을 하고, 그 Master 서버의 데이터를 복제해서 여러대의 Slave 서버를 만든 후에 Slave 서버에서는 읽기 작업만 수행하도록 하는 방식입니다.
- Replica Set 구조에서 Write를 수행하는 DB는 'Primary', Read를 수행하는 DB는 'Secondary'라고 부릅니다.

▶ 파일 서버(File-Server)

- 파일을 저장하여 관리할 경우 별도의 파일 서버를 사용합니다.
- 데이터 손실을 최대한으로 줄이기 위해 총 3개 이상의 파일서버가 필요합니다.
- AWS의 S3, AZURE의 Blob, GCP의 Google Storage, MongoDB의 GridFS에서 서비스를 제공합니다.

▶ 스케일 업(Scale Up)

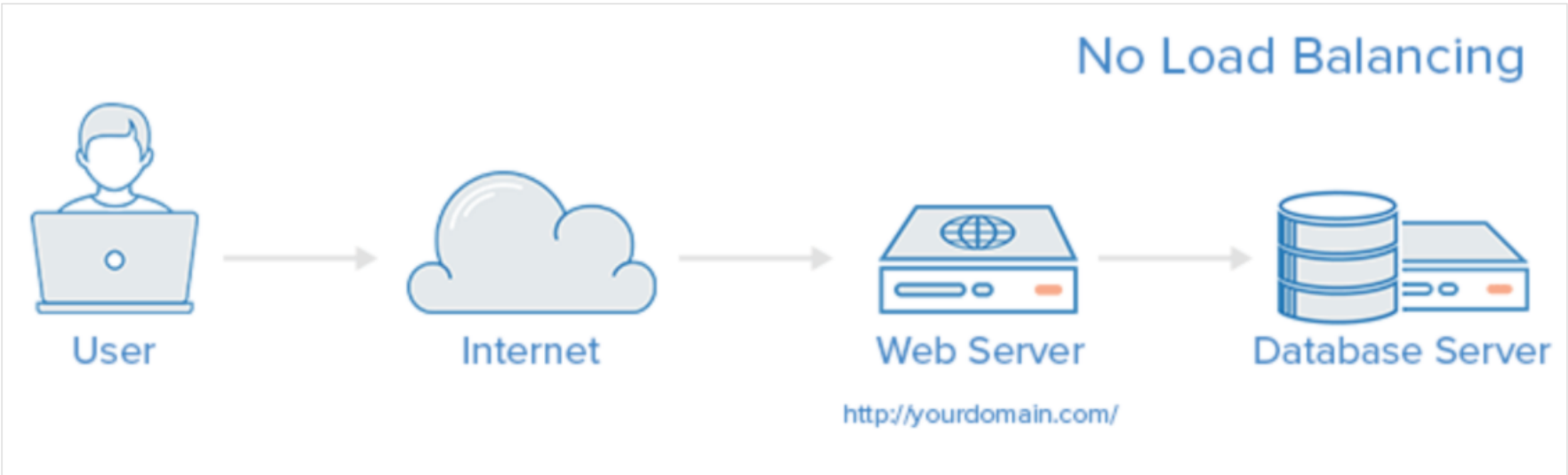
- 서버 장비의 스펙을 업그레이드하여 성능을 향상시킵니다.

▶ 스케일 아웃(Scale Out)

- 서버 장비의 수를 늘려 성능을 향상시킵니다.

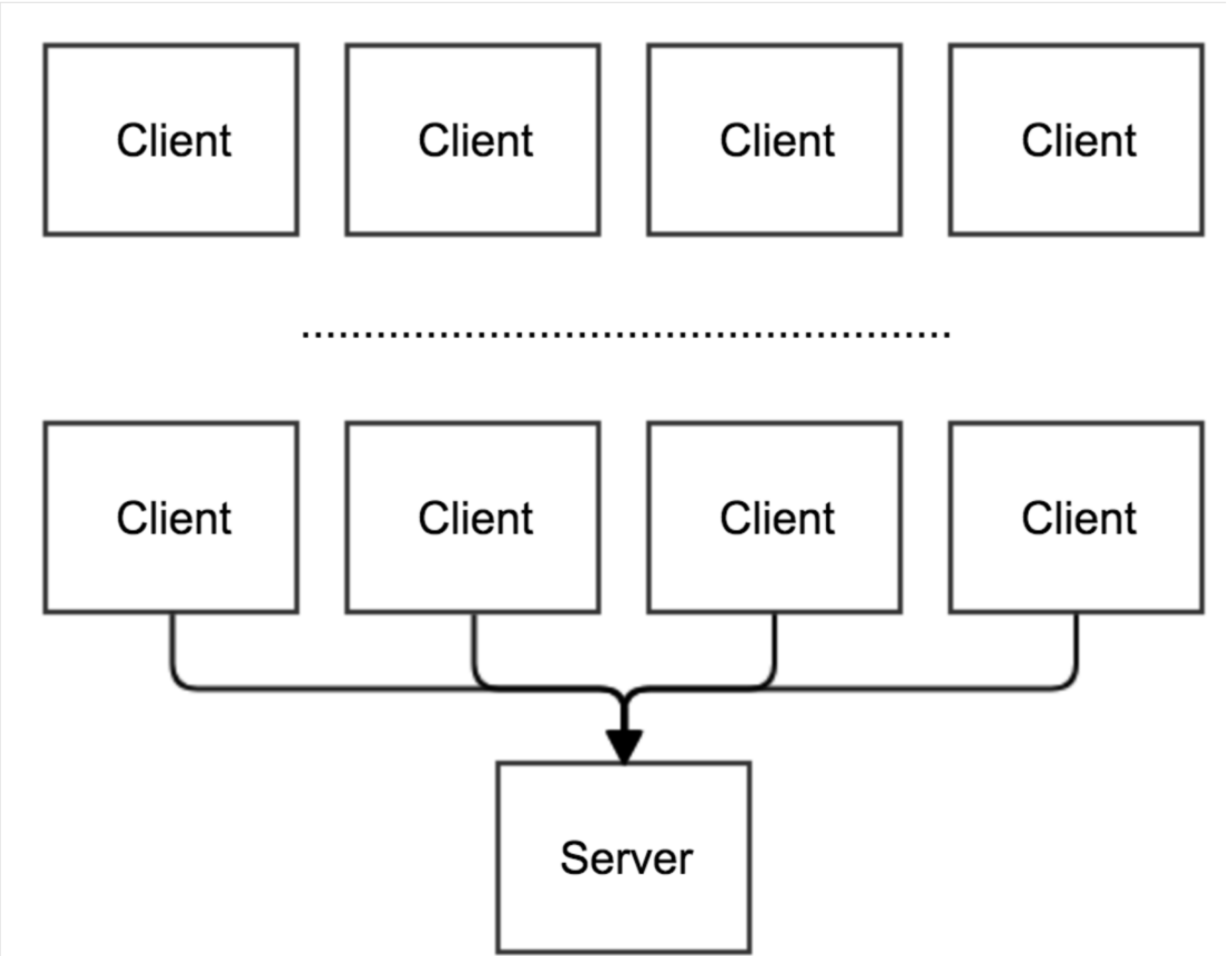
왜 Load Balancer가 필요한가요?

Client가 한 두명인 경우에는 어떨까요?



- Server는 여유롭게 사용자가 원하는 결과를 응답 해줄 수 있습니다.

하지만 Client가 한 두명이 아닌 수천만명이라면 어떨까요?



- Server는 모든 사람들의 응답을 해주려고 노력하지만 결국엔 지치게 되어 동작을 멈추게 됩니다.

문제를 해결하기 위해서는 어떻게 해야할까요?

- **Scale-up**: Server가 더 빠르게 동작하기 위해 하드웨어 성능을 올리는 방법.
- **Scale-out**: (하나의 Server 보다는)여러 대의 Server가 나눠서 일을 하는 방법.

Scale-out의 장점은 무엇이 있을까요?

- (하드웨어 향상하는 비용보다)서버 한대 추가 비용이 더 적습니다.
- 여러 대의 Server 덕분에 무중단 서비스를 제공할 수 있습니다.

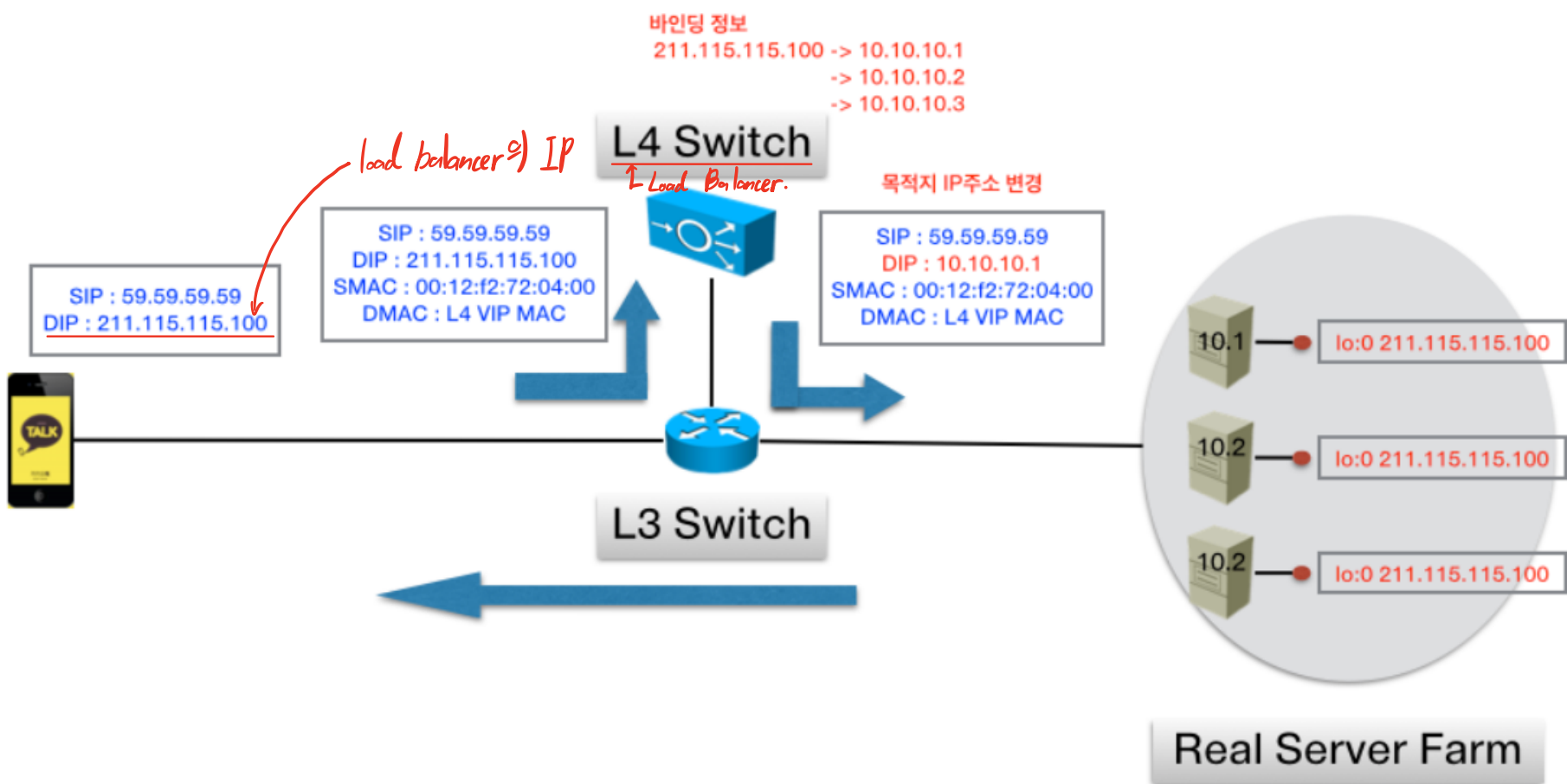
여러 대의 Server에게 균등하게 Traffic을 분산시켜주는 역할을 하는 것이 Load Balancer 입니다.

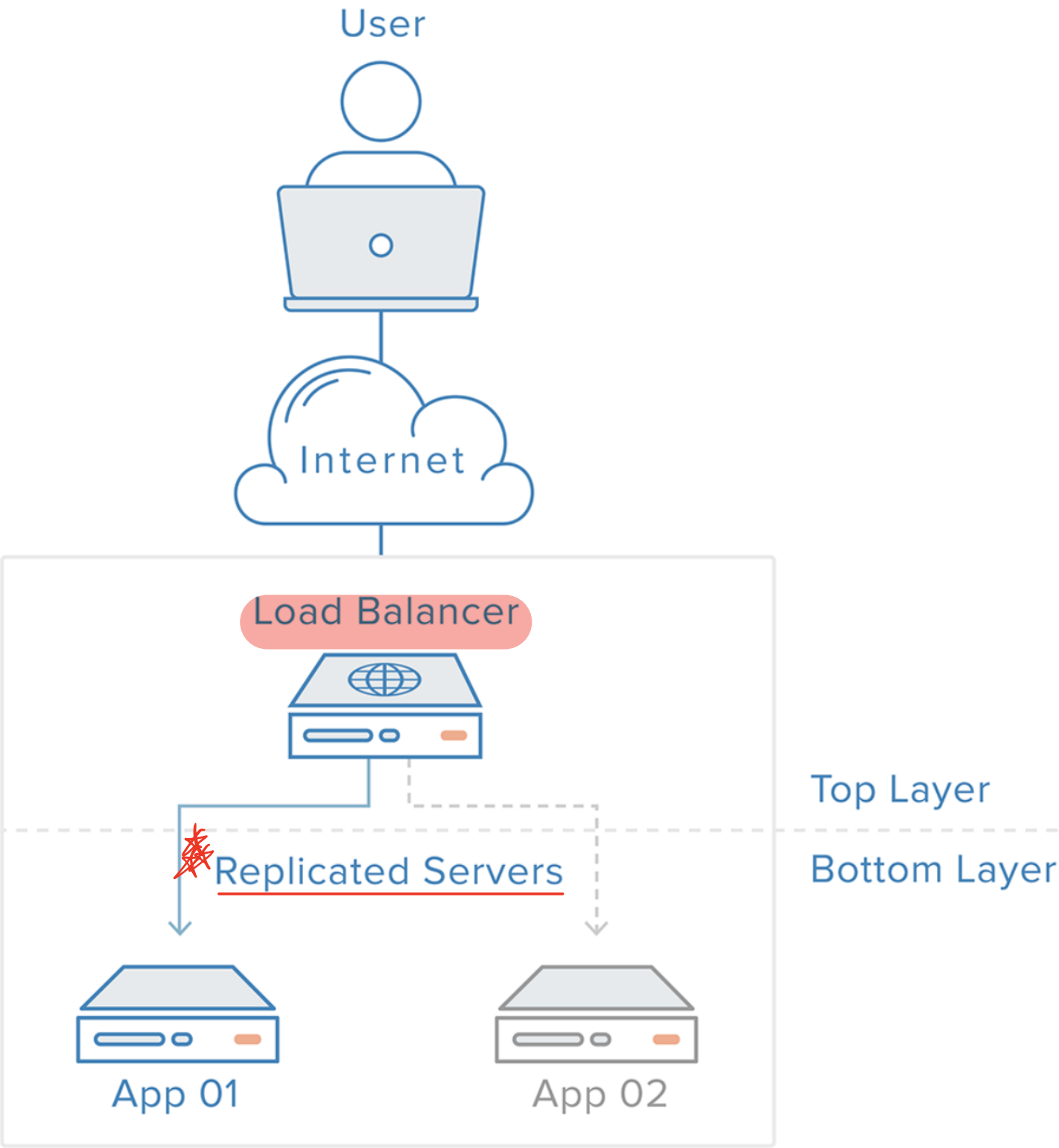
Load Balancing이란?

- 하나의 인터넷 서비스가 발생하는 트래픽이 많을 때 여러 대의 서버가 분산처리하여 서버의 로드울 증가, 부하량, 속도저하 등을 고려하여 적절히 분산처리하여 해결해주는 서비스입니다.

주요 기능은 어떻게 있을까요?

- **NAT(Network Address Translation)**
 - 사설 IP 주소를 공인 IP 주소로 바꾸는 데 사용하는 통신망의 주소 변조기입니다.
- Tunneling
 - 인터넷상에서 눈에 보이지 않는 통로를 만들어 통신할 수 있게 하는 개념
 - 데이터를 캡슐화해서 연결된 상호 간에만 캡슐화된 패킷을 구별해 캡슐화를 해제할 수 있습니다.
- DSR(Dynamic Source Routing protocol)
 - 로드 밸런서 사용 시 서버에서 클라이언트로 되돌아가는 경우 목적지 주소를 스위치의 IP 주소가 아닌 클라이언트의 IP 주소로 전달해서 네트워크 스위치를 거치지 않고 바로 클라이언트를 찾아가는 개념입니다.





종류는 어떤 것이 있을까요?

L2

- Mac주소를 바탕으로 Load Balancing합니다.

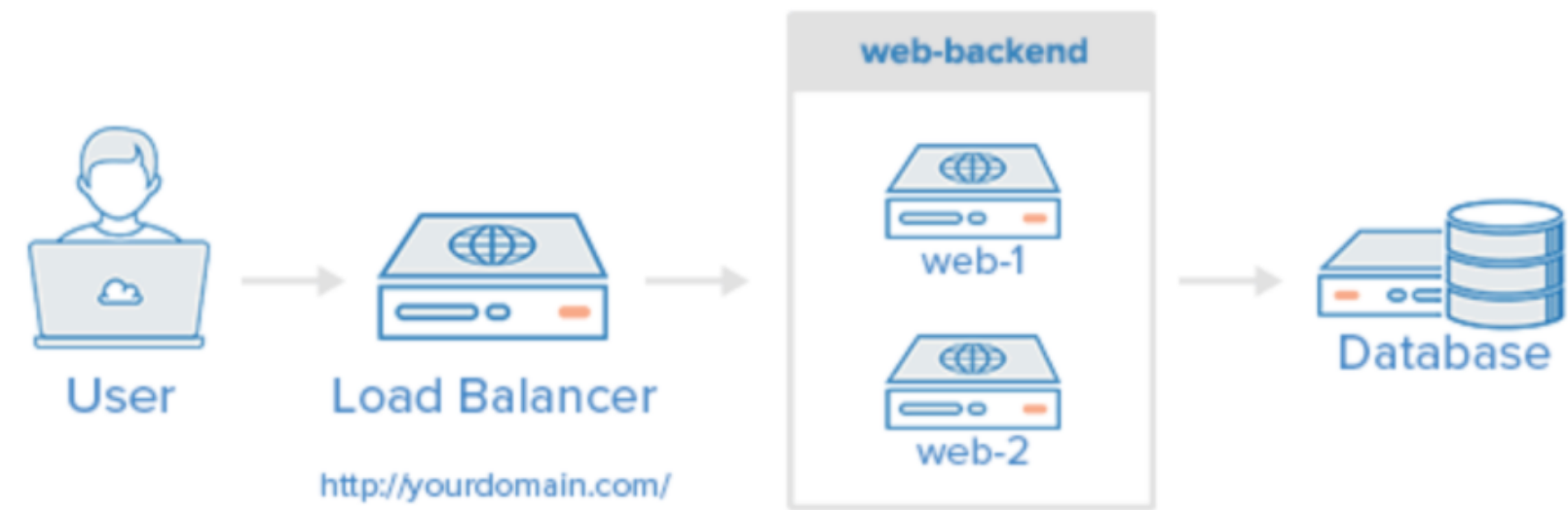
L3

- IP주소를 바탕으로 Load Balancing합니다.

L4

- Transport Layer(IP와 Port) Level에서 Load Balancing을 합니다.
- TCP, UDP

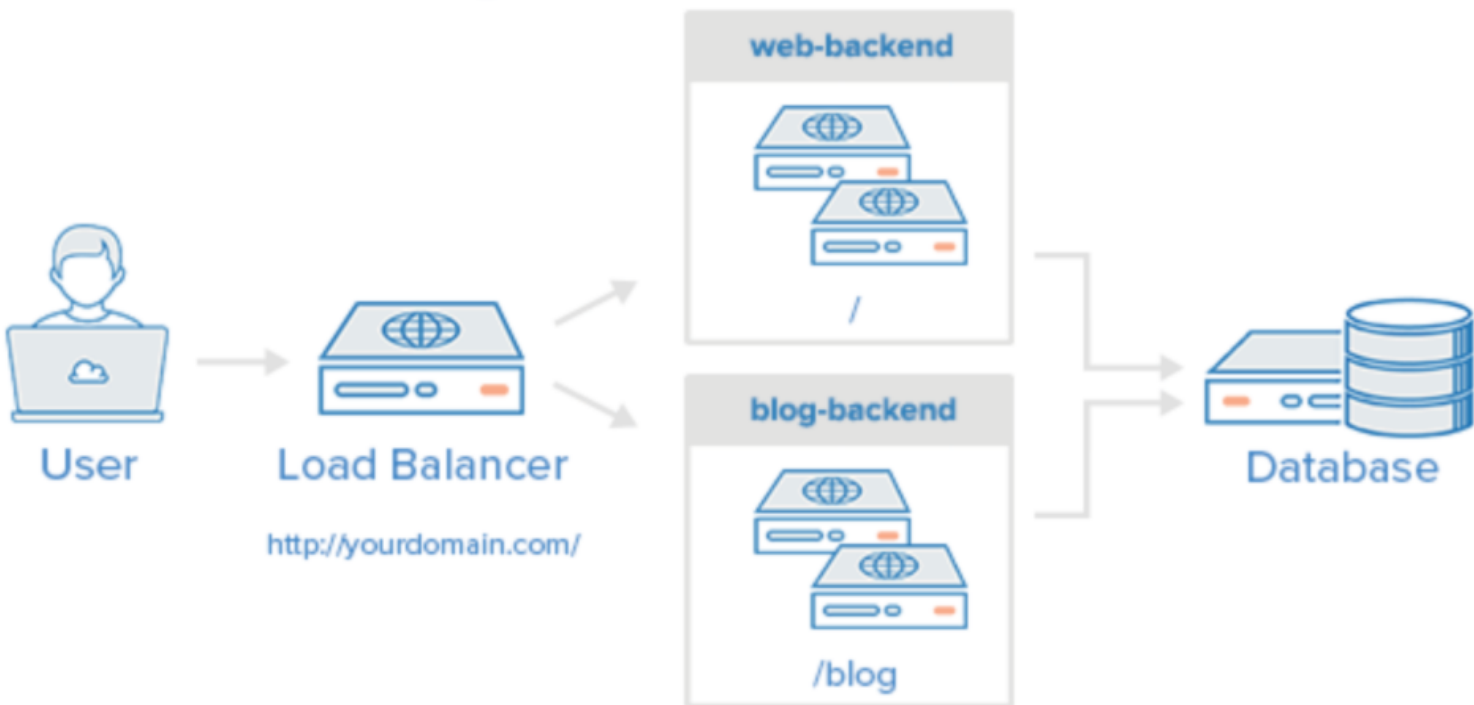
Layer 4 Load Balancing



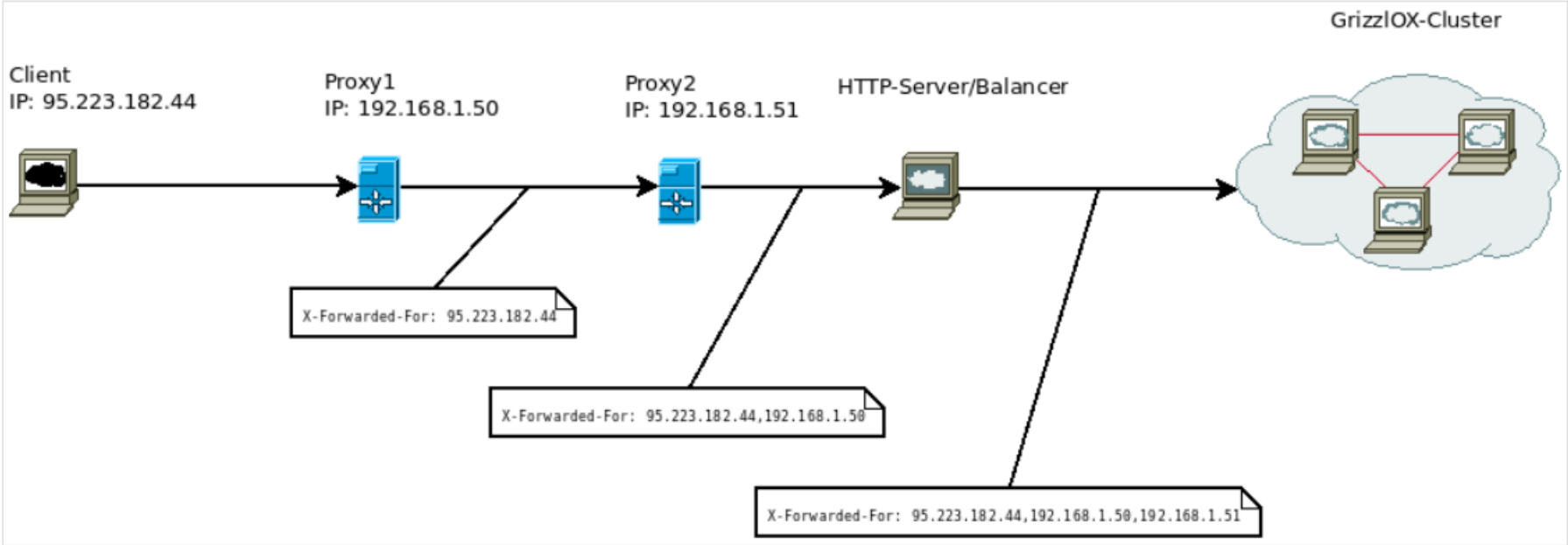
L7

- Application Layer(사용자의 Request) Level에서 Load Balancing을 합니다.
- HTTP, HTTPS, FTP

Layer 7 Load Balancing



HTTP



- X-Forwarded-For
 - HTTP 또는 HTTPS 로드 밸런서를 사용할 때 클라이언트의 IP 주소를 식별하는 데 도움을 줍니다.
- X-Forwarded-Proto
 - 클라이언트가 로드 밸런서 연결에 사용한 프로토콜(HTTP 또는 HTTPS)을 식별하는 데 도움을 줍니다.
- X-Forwarded-Port
 - 클라이언트가 로드 밸런서 연결에 사용한 포트를 식별하는 데 도움을 줍니다.

Load Balancer는 어떤 기준으로 Server를 선택할까요?

- Round Robin
 - 단순히 Round Robin으로 분산하는 방식입니다.
- Least Connections
 - 연결 개수가 가장 적은 서버를 선택하는 방식입니다.
 - 트래픽으로 인해 세션이 길어지는 경우 권장하는 방식입니다.
- Source
 - 사용자의 IP를 Hashing하여 분배하는 방식입니다.
 - 사용자는 항상 같은 서버로 연결되는 것을 보장합니다.