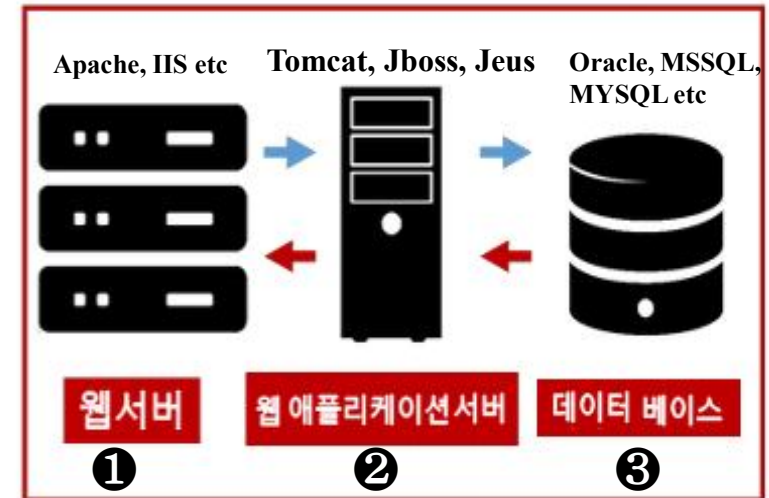


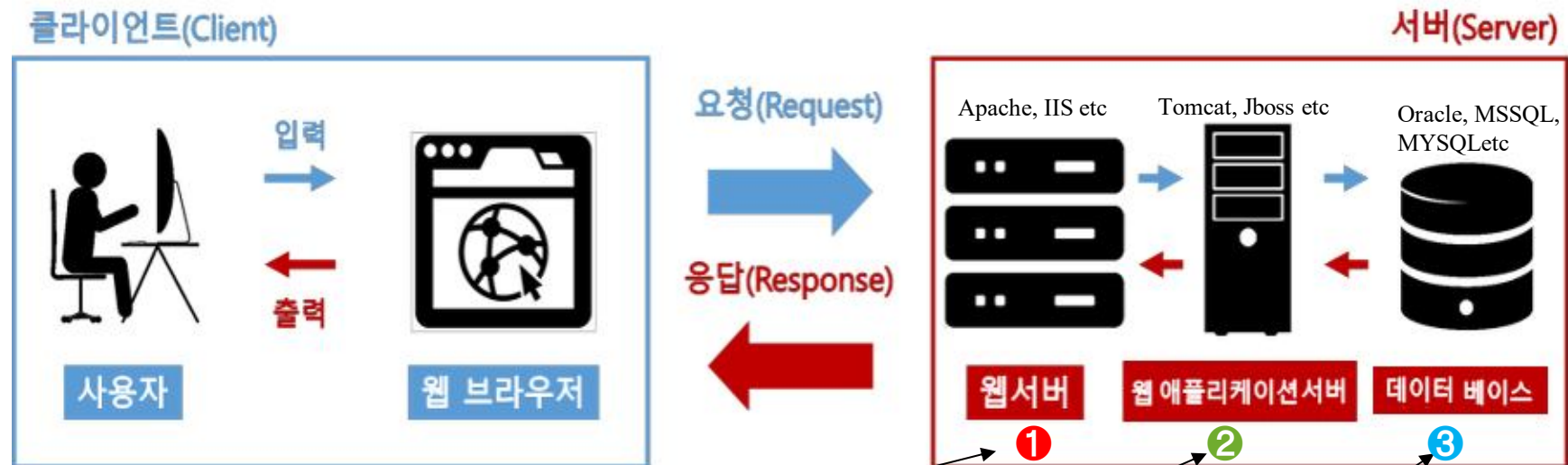
## 클라이언트(Client)



## 서버(Server)



①	FrontServer (Web Server)	<ul style="list-style-type: none"> <li>• HTTP request를 받아 HTTP response를 보내는 서버</li> <li>• Apache, IIS, Nginx 등의 미들웨어가 동작</li> </ul>
②	Application Server	<ul style="list-style-type: none"> <li>• 업무처리 서버 (결제, 주문 등을 수행하는 프로그램 실행환경)</li> <li>• Apache Tomcat, WebSphere Application의 미들웨어로 구현</li> <li>• PHP나 Perl 실행환경이 필요 시 mod_perl 또는 mod_php 등 설치</li> </ul>
③	DB Server	<ul style="list-style-type: none"> <li>• 애플리케이션 실행 시 생성되는 데이터 관리 기능을 가진 미들웨어</li> <li>• MySQL, PostgreSQL, Oracle Database</li> </ul>



<http://www.google.co.kr:80/sport/guest.php?who=anony&type=human>

URL= 프로토콜://[FQDN][Port번호]      요청 URI

URI = 프로토콜://주소(또는 IP):포트번호/경로 또는 파일?파라미터=값

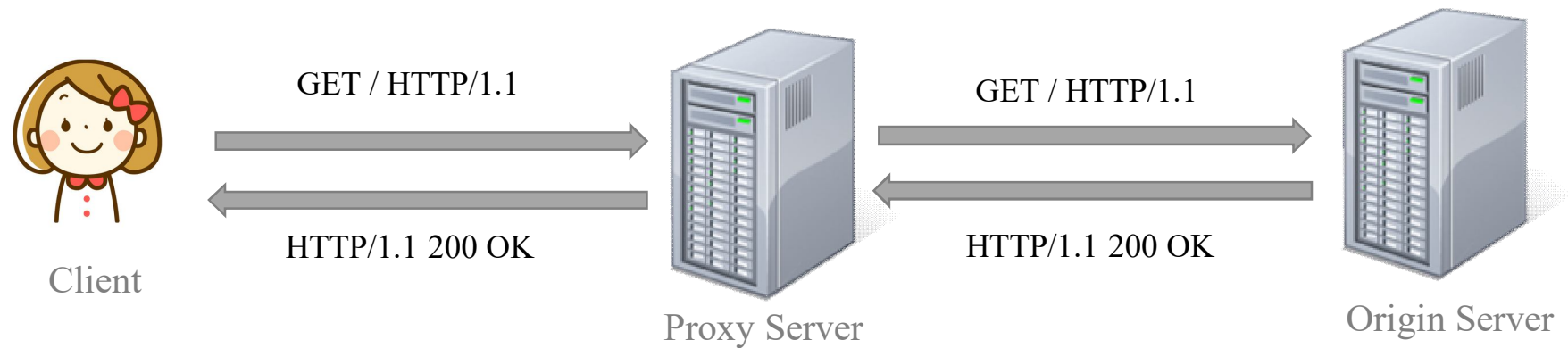
URL(Uniform Resource Locator)  
URN(Uniform Resource Name)

# Proxy Server

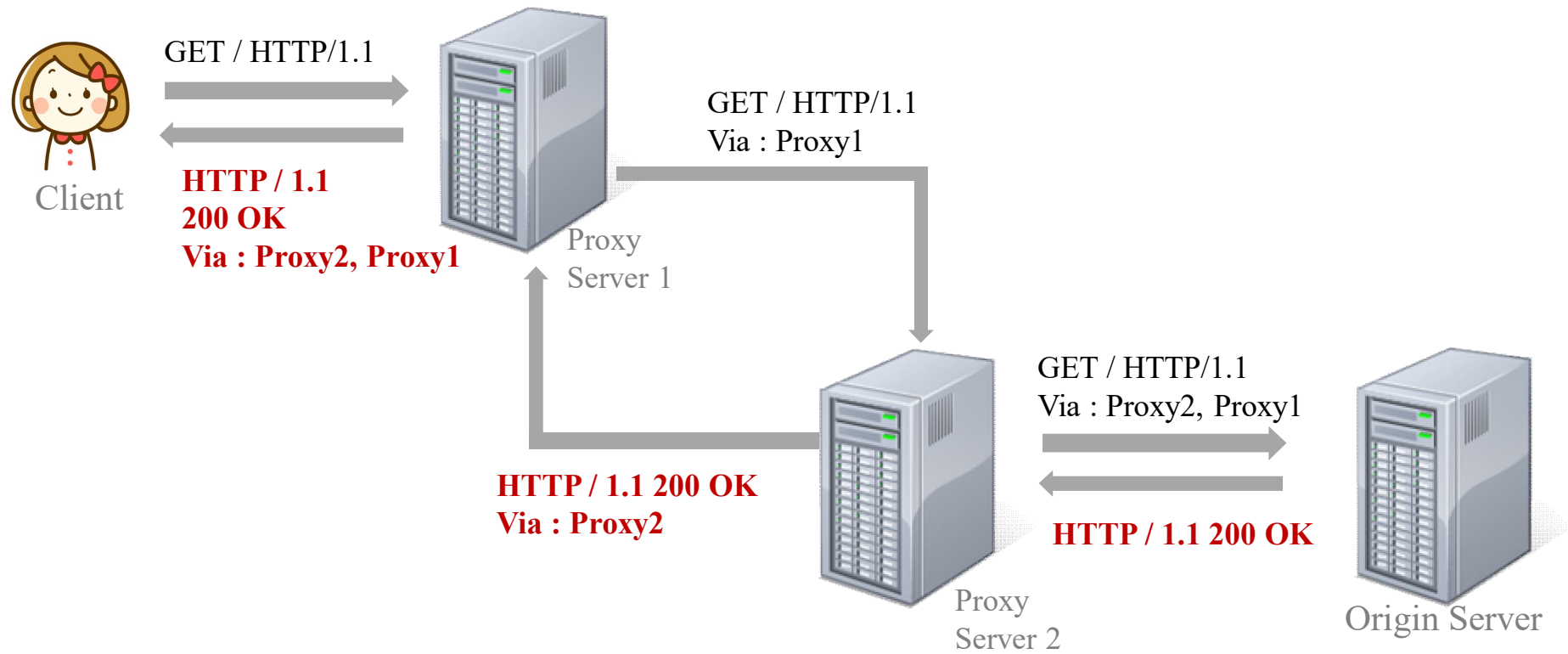
## 1) 정의

■ 서버와 클라이언트간의 요청과 응답을 중계하는 역할(프로그램)을 수행하는 서버(프로그램)

- 트래픽 부하, 캐시, 보안 등 여러가지 목적으로 사용



- Multi-Layer Web Proxy



# 기능

- 보안

- 실제 서버 또는 클라이언트의 IP를 숨김
- 프록시 서버의 IP만 공개함으로써 해킹 대비 가능

- 성능

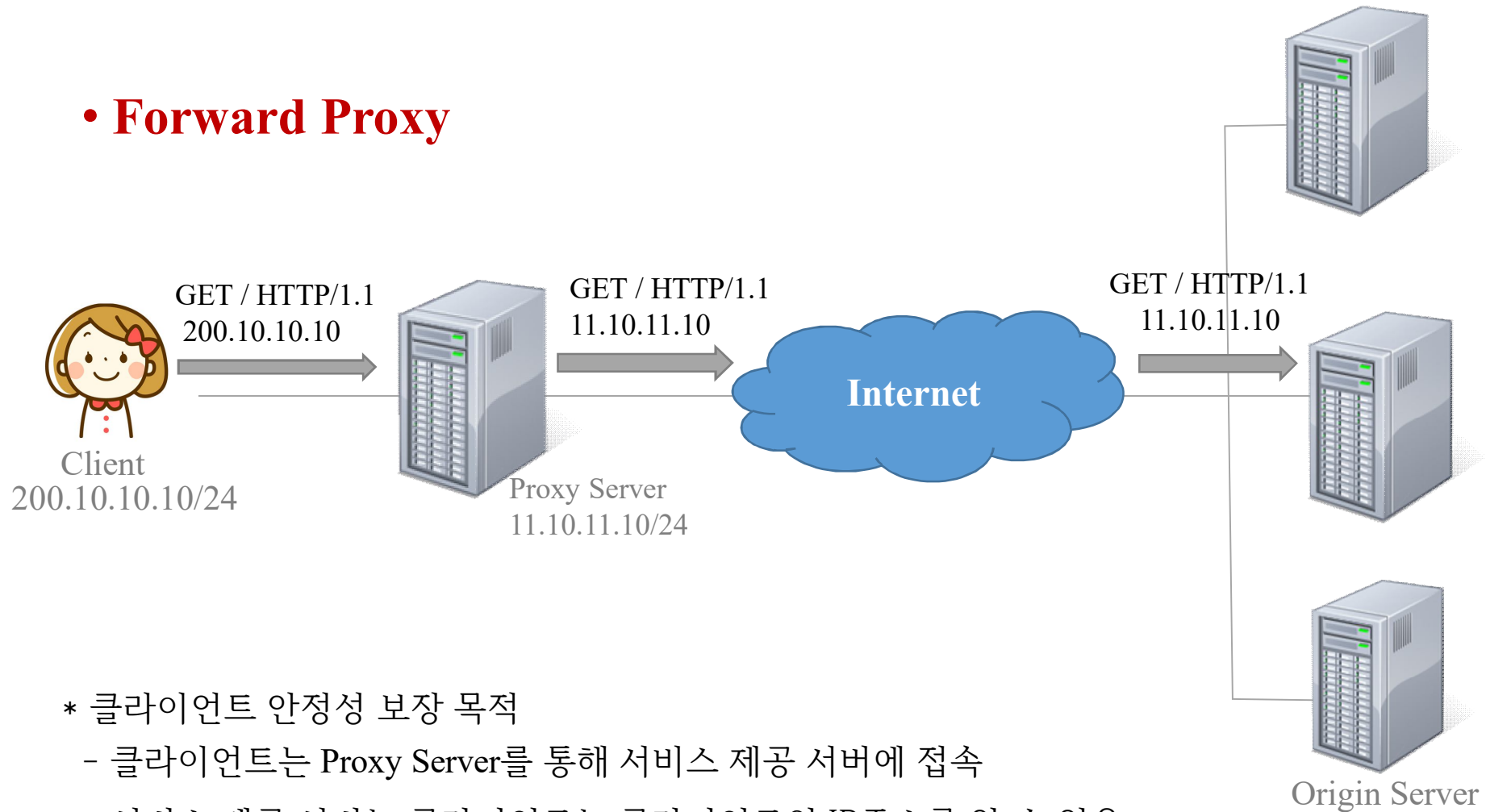
- 캐싱 기능과 트래픽 분산으로 성능 향상을 가져올 수 있음
- 캐싱 기능 : 자주 사용되는 동일한 요청을 캐싱하여 재활용하는 방식
- 실제 서버를 호출하지 않고 프록시 서버가 대신 응답을 주어 서버의 자원 사용을 줄임

- 트래픽 분산

- 일부 프록시 서버는 로드 밸런싱도 제공하여 여러 대의 분산된 서버가 있다면 서버의 트래픽을 분산시킴

# 기능 - 보안적 측면

## • Forward Proxy

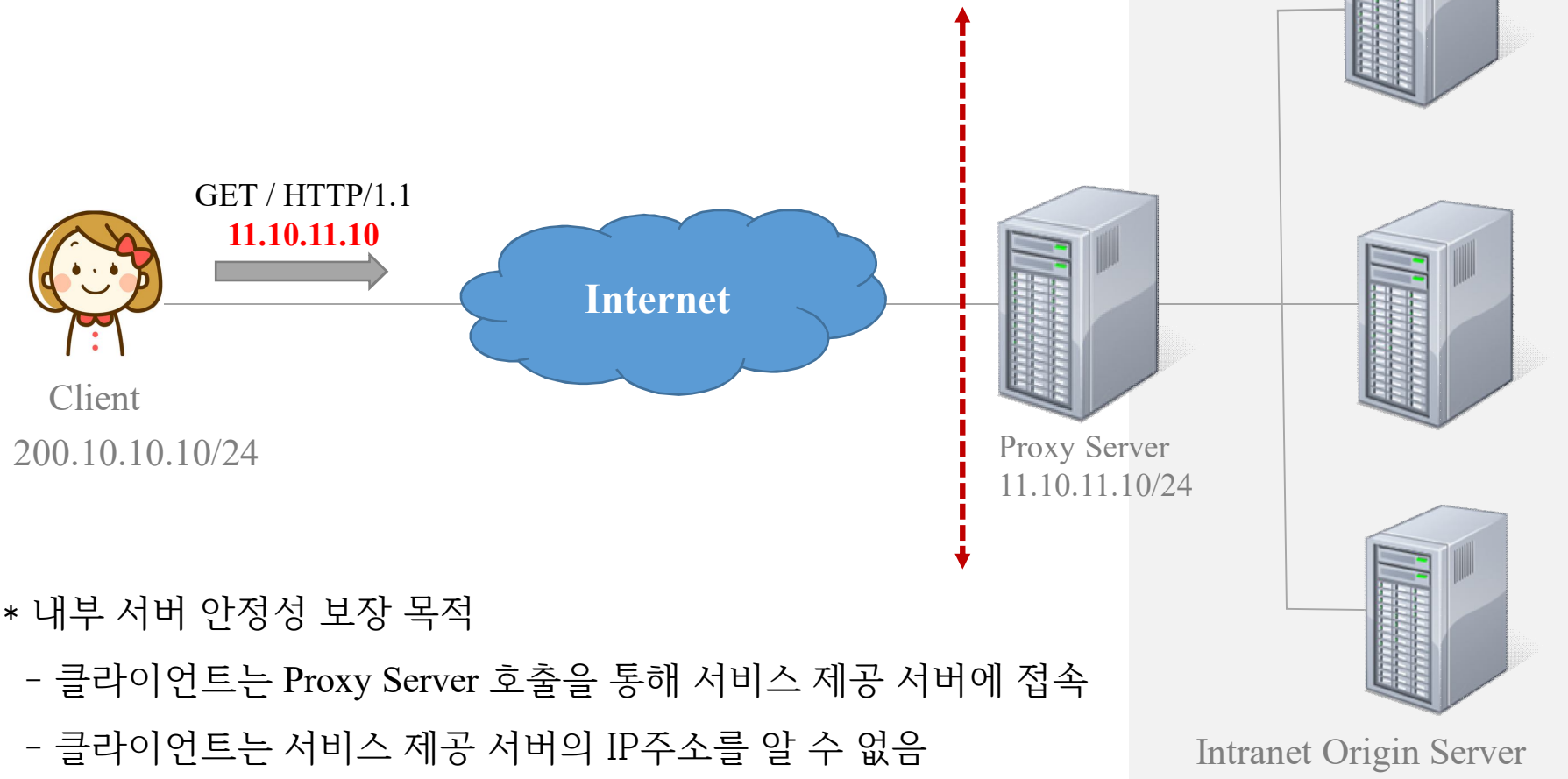


\* 클라이언트 안정성 보장 목적

- 클라이언트는 Proxy Server를 통해 서비스 제공 서버에 접속
- 서비스 제공 서버는 클라이언트는 클라이언트의 IP주소를 알 수 없음

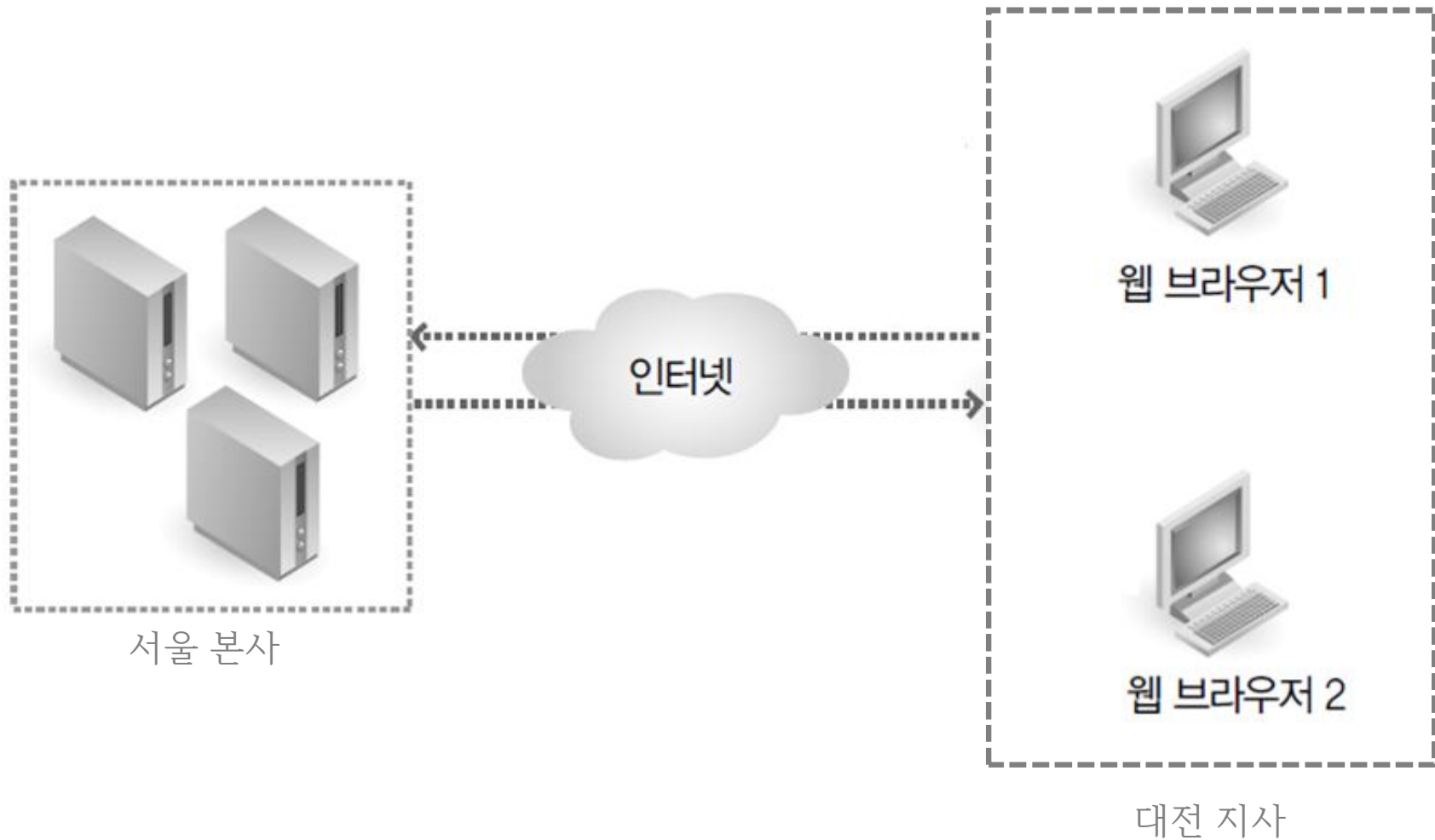
# 기능 - 보안적 측면

## • Reverse Proxy



# 기능 - 성능적 측면

- HTTP Proxy가 없는 경우 작업 지연 발생





# 기능 - 성능적 측면

- HTTP Proxy가 있는 경우 작업 지연이 낮음

