



서버

웹서버와 WAS

https://blog.naver.com/gi_balja/223028077537

1. 웹서버

클라이언트(브라우저)의 HTTP 요청에 따라 HTML, CSS, JS, 이미지 파일과 같은 **정적 파일**을 응답하여 제공하는 소프트웨어

- HTTP 프로토콜을 사용하여 클라이언트와 통신하며 WAS로 요청을 전달하는 역할
- Nginx, Apache

2. WAS(Web Application Server)

클라이언트 요청에 대한 **동적인 처리**를 담당

- 웹 서버와 달리 애플리케이션 로직을 실행(사용자의 요청에 따라 변경되는 데이터) 회원가입, 로그인 등
- 서버 측에서 프로그램 로직을 실행하여 데이터베이스와 상호작용하거나 비즈니스 로직을 처리
- 데이터베이스 연동, 세션, 트랜잭션 관리, 보안, 로깅 가능
- Java, Python, PHP 등과 같은 서버 측 언어로 작성된 코드 실행
- Tomcat, JBoss, node.js

웹서버를 사용하는 이유

⇒ 현대의 웹 애플리케이션에서는 웹서버와 WAS를 함께 사용하여 성능과 확장성을 극대화

웹서버는 정적인 콘텐츠를 처리하고 요청을 WAS로 전달

WAS는 동적 요청을 처리하고 결과를 웹서버를 통해 사용자에게 반응

1. 정적인 파일을 클라이언트에게 전달하는 역할을 웹서버에 위임하여 WAS 작업 부담 감소
2. 웹 페이지에 대한 접근을 제어 (SSL/TLS 프로토콜을 사용하여 데이터 암호화, 액세스 제어, 웹 방화벽)
3. 비동기 처리 방식을 사용하여 높은 성능 제공(Nginx, Apache의 웹서버는 이벤트 기반, 멀티 프로세싱, 스레드 풀 등의 기술을 사용하여 수천 대의 클라이언트 요청을 동시에 처리 가능)

웹서비스 응답 순서



클라이언트 → 웹서버 → WAS → DB

Nginx 사용 이유

1. 높은 성능과 메모리 사용

비동기 I/O 처리 방식을 사용하여 높은 성능을 제공

⇒ 대규모 웹 사이트에서도 빠른 응답 시간을 보장할 수 있음.

⇒ Nginx는 적은 메모리 사용량으로 높은 성능 제공하여 서버 운용 비용 절감 가능

2. 리버스 프록시 사용이 가능

Proxy : 인터넷 접속 시 보안상의 문제로 직접 통신을 주고받을 수 없을 때 대리로 통신을 수행하는 기능

크게 **포워드 프록시** 와 **리버스 프록시** 로 나뉜다.

- **포워드 프록시**
 - 클라이언트와 인터넷 사이
 - 클라이언트가 정보 요청 시 포워드 프록시가 대신 받아 서버에게 전달, 이 후 서버의 응답 또한 포워드 프록시가 대신 받아 전달
 - 클라이언트의 IP 주소가 웹서버에 노출되지 않기 때문에 클라이언트의 위치나 신원 식별 어려움
 - 미디어 스트리밍 지원하여 미디어 파일을 캐시하고 클라이언트에게 빠르게 제공
- **리버스 프록시**
 - 인터넷과 백엔드 사이에 있는 서버 영역
 - **로드 밸런싱** : 클라이언트의 요청을 각각의 서버로 전달하여 요청에 대항하는 웹 서버로 길을 분배하는 것
 - 캐싱 서버로 이용 가능
 - 클라이언트가 이미지를 요청했을 때 처음에는 서버에서 가져오지만 이후 동일한 요청이 있는 경우 캐시 서버에서 가져와 클라이언트에게 전달

3. SSL(Secure Sockets Layer) 지원

- 웹 사이트와 사용자 간 통신을 암호화하고 보안을 유지하는 프로토콜
- HTTPS(HTTP Secure)
- Nginx는 HTTPS 인증서 제공

4. 데이터 압축

클라이언트가 보내는 요청이 Text인 경우 gzip을 사용하여 해당 데이터 압축

5. 비동기 처리

이벤트 루프 방식을 사용하여 높은 성능을 제공

이를 통해 동시에 여러 요청이 들어왔을 때 많은 트래픽을 동시에 처리할 수 있음.