

## AP서버, ApApplication Server

먼저 Application Server란, 말그대로 서버 그 자체

네트워크가 연결되어있기만 하다면, 그 네트워크를 통해 서버와 Endpoint 간의 통신을 할 수 있는 Server 이다.

즉, HTTP 뿐 아니라 TCP, UDP 등 다양한 프로토콜을 전달받아 클라이언트에 다양한 서비스를 제공한다

## 웹서버, Web Server



Web Server 는 HTTP 프로토콜을 주로 처리하는 서버 (따라서 Web Server는 Application Server에 포함)

HTTP Request를 받아 HTTP Response를 주며, Request를 처리하기 위해 Static HTML, Image 또는 JSON 등을 이용

JSP, 서블릿, ASP 등이 이용되어 요청에 대한 단순 응답을 반환하는 간단한 구조

인터넷 발달 초기, 단순히 인터넷을 통해 문서 조회만 가능하던 시절의 HTTP 서버들은 주로 정적인 동작만 하는 Web Server 였다고 할 수 있다.

대표적으로 Apache

Apache는 정적인 처리에 특화된 웹서버이고, 세트로 있는 Tomcat은 Servlet Container로 정적인 데이터와 동적인 데이터 처리가 가능하지만, 주로 동적인 데이터의 처리를 하는 Web Application Server이다.

## WAS, Web application Server

WAS는 HTTP 를 이용하는 Application Server

인터넷의 발달하면서 WAS 가 태어남

인터넷의 편한 접근성과 HTTP 라는 단순하면서도 효율적인 프로토콜의 발달로, 인터넷을 통해 단순히 문서 조회 이상의 많은 것들을 요구

기존에 TCP / UDP 등의 프로토콜들이 처리하던 전자상거래, 파일 공유 등의 기능 들을 HTTP 로 수행하려다 보니 나타난 새로운 형태의 서버

추가로 말하자면 정적인 HTTP 데이터 처리에 특화된 Web server에 동적인 데이터를 이용하게끔 하는 Container를 엮으면 WAS가 되며, WAS는 HTTP 를 이용하는 Application Server로 볼 수 있다.

이러한 미세한 차이 때문에 일반적으로 위에 언급한 **Static Server** 의 경우(WAS라고 말하기 보다는) Web Server 혹은 스테틱 서버 라고 표현을 많이 하고, HTTP 프로토콜을 사용하지 않는 TCP 서버 등은 (WAS가 아닌) App Server 혹은 Application Server 로 표현한다.

출처: <https://jins-dev.tistory.com/entry/Application-Server-와-Web-Server-에-대하여>

## DB서버

데이터베이스 저장만을 담당하는 서버

✱ 일반적으로 WAS 서버와 Database서버를 분리하는 걸 권장

### 단일 서버와 분리서버의 물리적 차이

@ 단일 서버 구성

웹 클라이언트 => 연결 <= 웹서버 WAS + 데이터베이스 서버

@ WAS와 데이터베이스 서버를 분리 구성

웹 클라이언트 => 연결 <= 웹서버 WAS => 연결 <= 데이터베이스 서버

## WAS와 데이터베이스 서버 분리는 언제 필요한가?

- DB서버를 분리하는 가장 큰 이유는 WAS(Web Application Server)에서 처리할 부분이 DB서버에 비해 상대적으로 많은 경우
  - 예를들어 쇼핑몰 등에서 사진 및 영상이 많을 수 있으므로(상대적으로 데이터베이스의 사양보다는) WAS 쪽 사양을 높이는 방법이 필요 → 하지만 동일 서버로 구현된 경우 WAS의 성능만 높이기 어려우므로 당연히 데이터베이스 서버를 함께 올릴 수 밖에 없음 → 가격과 성능을 고려할 경우 비효율적인 측면이 증가
- 메모리에 대한 의존도가 높은 DB
  - (WAS와 달리) Database는 다른 하드웨어보다 메모리 영향이 큼. 즉 성능을 높이기 위한 기술적 방법이 각기 다르게 적용되어야 하는데 하나의 단일 서버 구현은 한계

## WAS와 DB서버 분리의 장점

### 1. 확장성

성능을 높이기 위한 확장성입니다. 앞서 말한 것처럼 단일 머신의 스펙을 높이기 위해 병렬 구조의 Scale out 방식의 서버 증설이 필요한 경우 DB 서버 분리는 반드시 필요한 과정 중 하나입니다. 이렇게 해야 자원을 WAS 서버와 공유하지 않고 독점적으로 사용하므로 더 높은 성능을 기대할 수 있습니다.

### 2. 보안성

더 높은 보안을 위해서 나누는 것이 좋습니다. 만약 WAS 서버에 해킹이 발생한 경우 예를들어 봅니다. 해킹에 의해 루트 권한 모두가 넘어갈 수도 있고 이 경우 데이터에 심각한 문제가 발생할 수 있습니다. 하지만 별도로 데이터베이스 서버로 분리하여 제한적인 권한을 부여한 경우 심각한 문제로부터 상당 부분 보호할 수 있게됩니다.

### 3. 기타 백업 등등

형상 관리 툴을 사용하는 경우 당연히 소스 등은 히스토리와 백업에 안전할 수 있습니다. 하지만 데이터베이스의 경우 별도의 백업하고 주기적으로 백업을 수행할 경우 그 크기도 작지 않으므로 분리했을 때 백업 방법의 선택의 폭이 넓다 하겠습니다.

최근에는 대용량 트래픽 부하를 막기 위해 로드 밸런싱(Load Balancing)의 방법이 필요합니다. 중간에 로드밸런서를 두고 여러개의 웹서버를 분산하여 사용하려면 WAS와 데이터베이스 서버를 각각 다르게 운영해야 합니다. 그래야 WAS와 Database의 Scale Out을 각각 다르게 설정 사용할 수 있습니다.