

1020

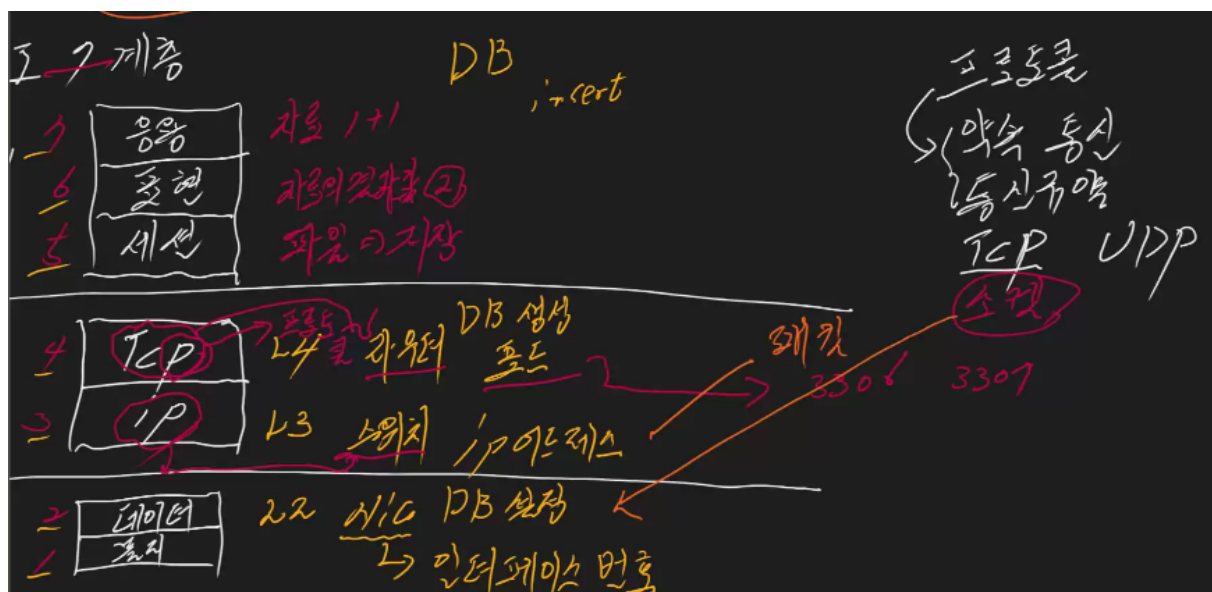
🕒 작성 일시	@2023년 10월 20일 오전 9:34
🏷 태그	

-로드 밸런스 기능 : 과부하가 걸리지 않도록 데이터를 나누어 주는 기능

L4(TCP/UDP), L7(TCP/UDP를 포함한 App의 정보)

	L4 로드밸런서	L7 로드밸런서
네트워크 계층	Layer 4 전송계층(Transport layer)	Layer 7 응용계층(Application layer)
특징	> TCP/UDP 포트 정보를 바탕으로 함	> TCP/UDP 정보는 물론 HTTP의 URI, FTP의 파일명, 쿠키 정보 등을 바탕으로 함
장점	> 데이터 안을 들여다보지 않고 패킷 레벨에서만 로드를 분산하기 때문에 속도가 빠르고 효율이 높음 > 데이터의 내용을 복호화할 필요가 없기에 안전함 > L7 로드밸런서보다 가격이 저렴함	> 상위 계층에서 로드를 분산하기 때문에 훨씬 더 섬세한 라우팅이 가능함 > 캐싱 기능을 제공함 > 비정상적인 트래픽을 사전에 필터링할 수 있어 서비스 안정성이 높음
단점	> 패킷의 내용을 살펴볼 수 없기 때문에 섬세한 라우팅이 불가능함 > 사용자의 IP가 수시로 바뀌는 경우라면 연속적인 서비스를 제공하기 어려움	> 패킷의 내용을 복호화해야 하기에 더 높은 비용을 지불해야 함 > 클라이언트가 로드밸런서와 인증서를 공유해야하기 때문에 공격자가 로드밸런서를 통해서 클라이언트에 데이터에 접근할 보안 상의 위험성이 존재함

(내용 수정)



ip 어드레스 : 패킷

DB 설정 : 소켓

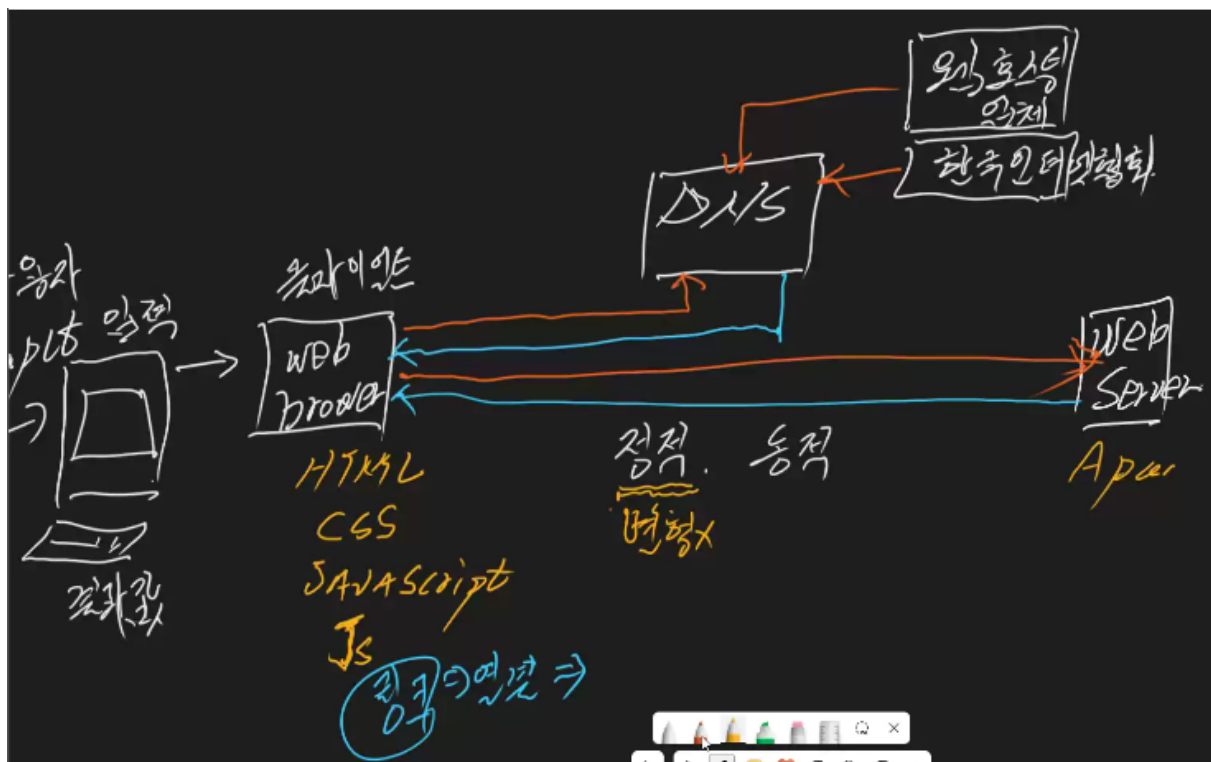
인터페이스 번호 : 비트

웹 서버의 구조

DNS : 질문에 해당하는 ip주소를 알려주는 서버

웹호스팅 업체, 한국 인터넷 협회: 자체 DNS 서버를 두고 DNS 서버에 정보를 제공한다.

사용자 → 입력 → 결과값 → WebBrowser(클라이언트) → DNS(Domain Name System)-웹 호스팅 업체, 한국 인터넷 협회 - DNS → WebBrowser → WebServer → WebBrowser (HTML, CSS (정적 자료), 동적자료)



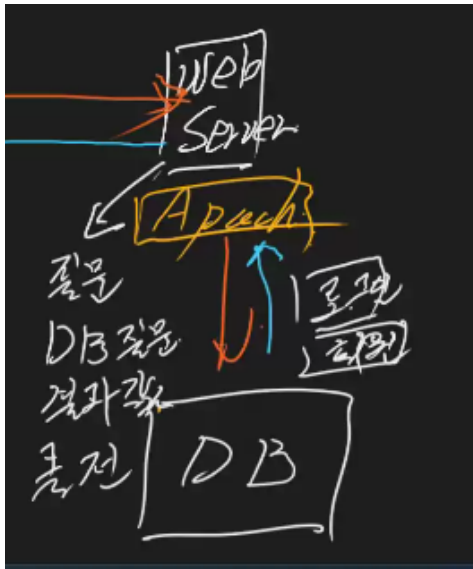
Js(JAVAScript)

정적자료 순서 : HTML → CSS → JS

HTML 만으로도 정적 파일을 받을 수 있다.

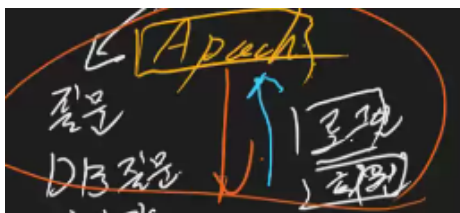
WebServer : 질문을 받고 DB에 다시 질문하고 결과에 대한 답을 얻어 결과값을 확인 후 클라이언트에 게 전송

최초 WebServer의 정의 : 웹 클라이언트가 질문에 대답



-DB에 접속하면서 로그인(회원)기능이 생긴다.

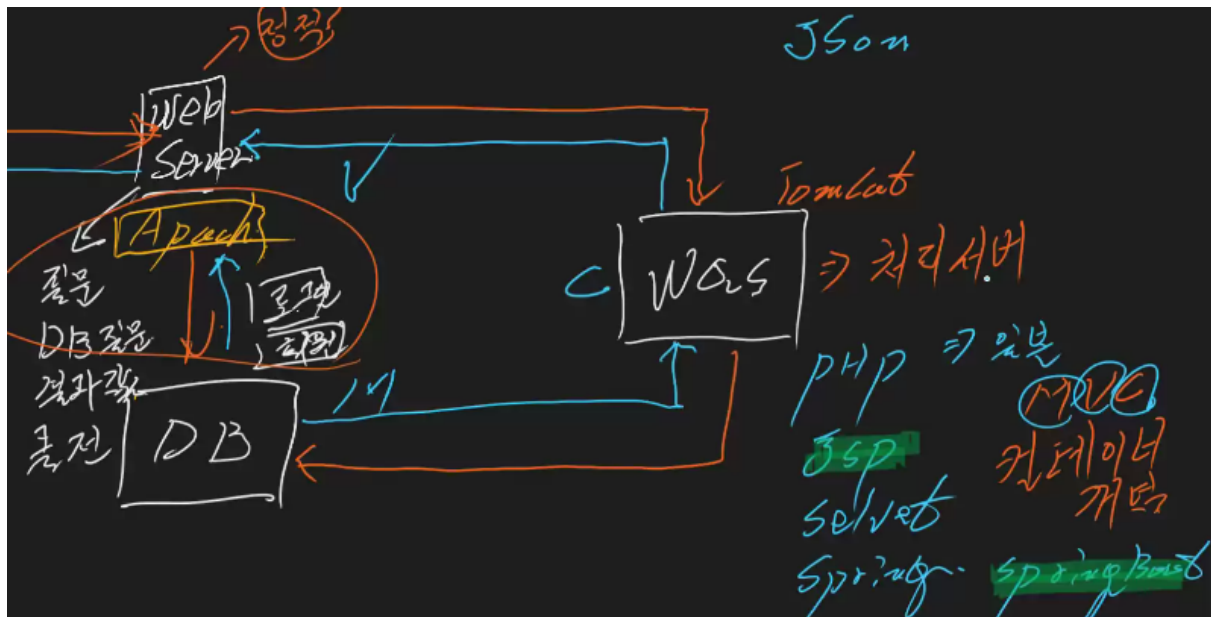
- WAS ⇒ TomCat : DB 의 기능을 대신함



정적 파일 웹서버 소지는 동일

이벤트(동작)가 발생 하기 시작

WAS⇒ WebServer의 질문값을 처리하는 서버



WebServer → Was → DB → Was → WebServer

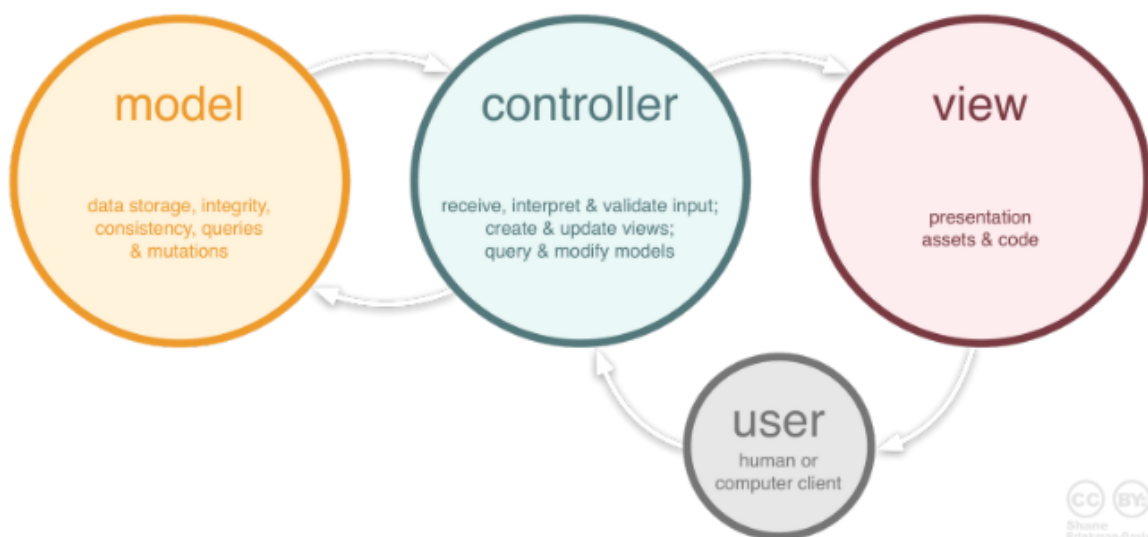
자료형태 : JSon

WAS의 구동하여 사용하는 tool : PHP(일본) → JSP → servlet → spring → spring Boot

JSP 최초로 MVC 모델 제공

스프링 - JPA 등장 - 전자정보 프레임 워크 규격 제작

MVC(model view controller) 모델 → 프레임 워크 : 미리 규격화된 처리 수단

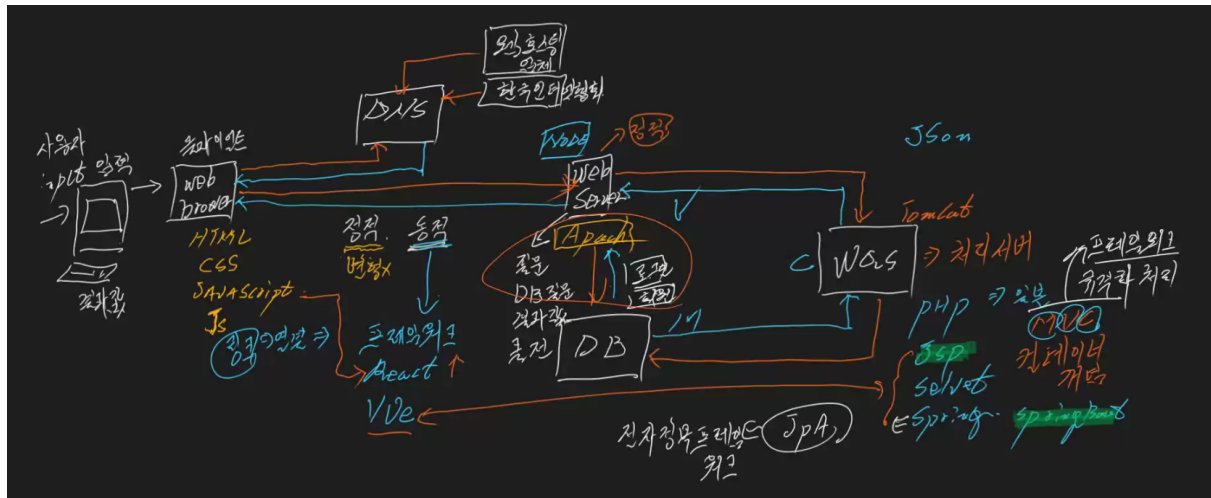


동적 파일을 받기 위해 프레임워크 (React, Vue): 내부적으로 실행되는 동적파일의 일

react : JS 기반의 기능을 향상

VUE : JS 기반으로 만들어 졌으면 JSP,SELVET, Soringboot에 호환성이 좋음

WAS로 인하여 정보 처리가 빨라지고 웹 위에서 구동 되는 프로그램 구현이 더욱 편해졌다.



차후 학습 내용

Web client : HTML, CSS, JS, TS(MS에서 개발) / IDE : VScode(MS에서 개발/ 가장 많이 사용)

React.JS, Vue.JS

Web server: Apache, node.JS(MS에서 개발), Next

Was : Jsp, selvet, JPA OMR, MyBaytis, loom book, greadel, Mvern, spring

DB : MariaDB, MySQL, Orcel, heidi, workbanch

JAVA/클라우드 AWS(웹 호스팅/ OS 리눅스) ⇒ 2차 프로젝트 끝나고 수업

└ 1년 무료 (도메인 구입 → 내것 500 ~1500원) 업데이트 자동 배포 (CICD/가능하면)

git hub 는 DB의 구조를 따라가기 때문에 영문만 가능

git hub 는 스냅샷 형식으로 저장

git clone 주소 : 복제

설정이 안된 상태에서 다른 사람에 git hub에 파일을 올릴 수 없다.

```
c:\Wgit-work\exm\exm02>git push -u origin main
remote: Permission to CEB79/exm02.git denied to papawon.
fatal: unable to access 'https://github.com/CEB79/exm02.git/': The requested URL returned error: 403

c:\Wgit-work\exm\exm02>git remote add origin https://github.com/CEB79/exm02.git/
error: remote origin already exists.

c:\Wgit-work\exm\exm02>git push -u origin main
remote: Permission to CEB79/exm02.git denied to papawon.
fatal: unable to access 'https://github.com/CEB79/exm02.git/': The requested URL returned error: 403
```

git hub 에서 다운로드 할 때 레파지토리 명이 같을 경우 cloen 명령어가 불가능하다.

1. 사용자 인증
2. 다운받은걸 내용 수정 후 업로드가 가능 한가
3. 브랜치 전략

깃허브 패스워드 발급후 clone 시에 주소에 같이 넣어 사용한다.

패스워드 : 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

주소: <https://github.com/je9898/hello02.git>

git clone

<https://github.com/je9898/hello02.git>

2번

git branch -M main

git clone

<https://github.com/je9898/hello02.git>

git pull origin main

git push -u origin main

콜라보레이트

브랜치

깃허브 정보 업로드 : `git push origin` (브랜치명)

브랜치 보기 : `git branch`

브랜치 생성 : `git branch` (브랜치 이름)

브랜치 선택 : `git checkout ceb`