<Backend>

* 백엔드의 구성  
  총 3가지의 분야로 구성이 됨
* 1. 서버 관리
* 2. 데이터베이스 구축
* 3. 백엔드 개발

<server>

* 일종의 컴퓨터
* 데이터 저장, 통신을 목적으로 하는 컴퓨터
* 서버를 중심으로 여러 사용자가 데이터 저장, 통신하는 중앙 매개체 역할을 함  
  (현재 상황에서는 내가 개발한 건 나만 볼 수 있음)\_
* 서버가 있어야지 만 내가 개발한 프로그램을 통신을 통해서 남들이 보게 할 수 있음(배포)

<server 확보 방안>

1. 서버 구입  
   - 직접 구입하는 것  
   - 초기에 큰 비용이 들어 감 + 서버의 환경 관리를 직접 해야 함  
   - 한 번 큰 비용이 들어가면, 추가적인 비용은 발생하지 않음
2. 서버 호스팅  
   - 서버를 임대하는 것  
   - 초기 비용이 없으나, 대신 사용한 만큼 매월 비용을 지불해야 함  
   - 서버의 환경 관리나 문제 해결 등을 대신해 줌  
   - 언젠가는 서버 구입비용보다 높아질 것

<server 호스팅 업체>

* Naver Cloud Platform, Cafe24 / AWS, Asure 등

<AWS>

* Amazon Web Services
* 서버 호스팅 외에도 다양한 클라우드 서비스를 제공하고 있음
* 외장 하드, SMS, 푸시 알림 등등(유료)

<클라우드>

클라우드 가상의 공간을 의미하는 기술

* 이 가상의 공간을 통신 등으로 이용할 수 있게 해놓은 기술이 클라우드

<Server – Client 구조>

* 클라이언트가 필요한 내용을 서버에서 받아오고자 하고 서버는 해당 내용을 클리이언트에게 줌
* 그 과정에서 그래픽화를 위해서 웹브라우저가 필요함
* 장점 – 잠수함 패치가 가능하다. 개발자가 프로젝트를 관리하기가 편하다.
* 단점 – 매번 통신이 필요하다
* 무거운 프로그램은 직접 통신으로 구현 해야함

<운영 체제>

OS / 컴퓨터의 하드웨어를 사용자가 소프트웨어적으로 이용할 수 있게 하는 프로그램

* Linux
* 운영체제의 한 종류이며
* 서버 컴퓨터 전용 운영 체제
* 오픈 소스여서, 리눅스 깊은 곳 까지 자유롭게 접근할 수 있음
* Linux 다루는 것도 매우 잘해야함 <마치 window 나 MacOS 다루 듯이>

AWS 서버 만들기

* 인스턴스 만들고
* Ppk (키페어) 받고
* 인스턴스 시작
* 초기 한 개 서버까지는 무료
* 보안 프로토콜 수정하기(default 수정 x)
* 8080 포트 추가하고 저장

Window 환경에서 AWS 서버 사용하기

* Putty 접속하기
* 로그인 하기 : (ubuntu)로 접속

Linux

Cli 기반 (command line int)

<명령어>

~ : home 폴더 (가장 처음 접속되는 폴더)

Cd, Ls, touch,

기본색상(화이트)로 나오는 것은 일반 파일

파란색상은 폴더, 빨간색상은 압축파일, 초록색상은 실행파일

Rm, rm -rf(force)

Tab(자동완성), 방향키 위아래(log)

Sudo apt update (업데이트)

Sudo apt upgrade (업데이트 적용)

<웹서버>

* Server Vs Web Server
* 둘 다 컴퓨터
* 둘 다 데이터 저장, 통신을 목적으로 하는 것은 동일
* 통신 방법이 다름
* 1. Sever 는 TCP 통신을 함  
  TCP는 양방향 통신
* 2. Web Sever 는 HTTP 통신을 지원 함  
  HTTP는 단방향 통신 (한 쪽은 요청만 한쪽은 응답만 가능)  
  그래서 통신이 간단함  
  Real time 통신이 되지 않음(채팅, 알림 등등)  
    
  Web이 HTTP통신을 써야 하니깐, 우리는 웹 서버가 필요한 상황  
  - web server 구축이 중요

Web Server 구축

* HTTP 도 통신이기에 깡으로 구축하는 건 매우 어려움
* 이를 자동으로 해주는 소프트웨어를 사용할 것
* WAS(Web Application server)
* 직접 HTTP를 구현했을 때랑 비교해서, 당연히 동작 속도가 느림
* 그리고 내 입맛대로 커스텀이 어려움

WAS의 종류

* 백엔드 언어를 뭘 쓰냐에 따라서 WAS가 정해져 있음
* JSP(java 기반 언어) -> java를 백엔드에 쓸 때는 Tomcat 이라는 WAS를 사용함
* 서버 -> 웹서버, 웹서버 구축에 was 사용, 백엔드는 jsp -> tomcat 사용

<웹 서버 구축>

1. Java 설치
   1. sudo(관리자 권한으로/ 꼭 필요한 경우에만 사용하기) apt(리눅스에서 뭔가 설치) install openjdk-11-jdk
   2. ubuntu 계정은 관리자가 되어있으므로 그냥 가능함(다른 계정이 있으면 불가능)
   3. “java –version”으로 설치 및 버전 확인
   4. 환경변수 설정하기  
      - readlink -f /usr/bin/java로 경로 찾아서 JAVA\_HOME 경로 설정  
      (vi 에디터로 profile 설정 해주기 (처음 profile을 작성해서 변경할 때는 sudo vi profile로 접속하기) 재부팅 명령어 source profile)
2. Tomcat 설치
   1. 톰캣 접속
   2. Tar.gz 링크 주소 복사
   3. 홈디렉토리(계정아래)에 설치
   4. “wget 링크” 웹으로부터
   5. “tar -zxvf 파일 이름” 압축해제
   6. 웹서버 설치 완료

Startup.sh – 웹서버 실행

Shutdown.sh – 웹서버 종료

“./실행파일” 파일 실행

HTTP 통신이 가능한 상태에 도달.(브라우저를 통한 접속이 가능)

톰캣 설명

Bin(binary) : 실행파일 모음

Conf(configure): 톰캣 설정 파일

Lib(library) : 톰캣에 추가적으로 설치할 때 넣는 디렉토리

Webapps: 작성한 코드가 저장되는 디렉토리(기본적인 프로젝트 파일 5개가 나옴)

:8080 (톰캣의 포트) 초기 설정으로는 AWS에서 막아 두었으므로 설정이 필요함

<주소>

Ip + 포트 = 사무실 위치 + 층 수

과제

1. 서울에 인스턴스 만들기
2. 메인 페이지 만들기   
   버튼 4개 만들기  
   1. 1주차 과제  
   2. 2주차 과제  
   3. 3주차 과제  
   4. 4주차 과제
3. 서버에 배포하기(서버 열기)
4. 파일 질라(무료) 사용해서 톰캣에 자료 올리기 (공부 필요)