

# 사전 학습 키워드 정리

---

## VM(Virtual Machine)

VM은 컴퓨팅 환경을 소프트웨어로 구현한 것을 의미합니다. VM상에 운영체제나 응용프로그램을 설치 및 실행 할 수 있습니다. 가상머신에는 시스템 가상머신과 프로세스 가상 머신이 있습니다. 먼저 시스템 가상머신은 완전한 시스템 플랫폼, 즉 완전한 각각의 운영체제의 실행을 지원합니다. 그리고 프로세스 가상머신은 하나의 단일프로그램을 실행합니다, 즉 단일 프로세스만을 지원합니다. 이 둘 가상머신의 중요한 특징은 해당 가상머신안에서 실행되는 소프트웨어는 해당 가상머신이 제공하는 환경과 자원에 제한을 받는다는 것입니다. 시스템 가상머신은 단일의 물리적 자원을 다중의 논리적 자원으로 다중화합니다. 하나의 물리적 자원 안에서 각각의 고립된 여러 운영체제를 동시에 사용 할 수 있는것입니다. 이때 여러 운영체제들을 일컬어 게스트 운영체제라고합니다. 프로세스 가상머신의 목적은 플랫폼 의존성을 제거하는 것입니다. 대표적으로 JVM(Java Virtual Machine)을 예로 들 수 있습니다. 이는 가상머신 상에서 응용프로그램을 실행하게 함으로써 어떤 플랫폼에서든지 같은 실행방식으로 프로그램이 실행되게끔 하고, 같은 응용프로그램을 해당 가상머신을 통해 어떤 플랫폼에서든지 실행 가능하게 합니다.

## HTTP(HyperText Transfer Protocol)

팀 버너스리와 그가 속한 팀이 HTML과 함께 고안해낸 HTML 전송 프로토콜입니다. TCP와 UDP를 사용하며 포트번호는 80입니다. 계속해서 발전해나가고 있어 현재 HTTP/2버전이 등장한 상태이며 제일 많이 사용되어지는 버전은 HTTP 1.1입니다. 해당 프로토콜은 서버/클라이언트 모델을 따르는데 클라이언트가 서버에 요청(request)을 하면 서버가 이에 대한 응답(response)을 전달합니다. 이러한 HTTP의 모델은 서버가 클라이언트의 요청에 따른 응답만 전송해 주면 되기 때문에 불특정 다수를 대상으로 하는 서비스에 적합하며 계속적으로 서버와 클라이언트가 연결되어 있는 상태가 아니기 때문에 서버가 감당할 수 있는 최대 클라이언트 연결 수 보다 많은 수의 클라이언트의 요청을 처리 할 수있습니다. 또한 HTTP는 무상태(Stateless)라는 특징을 가지는데 이는 서버와 클라이언트가 지속적인 연결을 유지하고 있지 않기때문에 생גיע 된 특징입니다. 클라이언트가 서버에 어떠한 정보와 함께 요청하기 전까지는 클라이언트의 상태가 어떤지 서버가 전혀 알수 없으며 해당 특징때문에 클라이언트의 정보를 유지하기 위한 기술인 Cookie와 같은 기술이 등장하게되었습니다.

## Protocol

프로토콜은 통신상의 규약을 의미하는데, 원거리 통신 상에서 수신과 송신의 양측에서 어떠한 데이터를 주고받는 양식과 규칙의 체계를 의미합니다. 원거리 통신 시 수신과 송신 측은 같은 프로토콜을 사용해서 통신해야 합니다. 수신측에서 전달한 데이터의 형식과 송신측에서 받기로 되어있던 데이터의 형식이 다르다면 송신측에서 수신측이 보낸 데이터가 의미하는바가 무엇인지 전혀 알수 없기 때문입니다.

## bash(Bourne-again shell)

사용, 수정, 배포 등의 제한이 없는 프리웨어(freeware)이며 GNU 프로젝트를 위해 브라이언 폭스가 작성한 유닉스 셸입니다. 1989년 발표되었으며 GNU 운영체제와 리눅스, 맥 os x 그리고 다윈 등 운영체제의 기본 shell로 탑재되어 있습니다. 여기서 shell이라 함은 명령어 처리기를 의미합니다. 컴퓨터의 사용자가 컴퓨터에게 어떠한 동작을 수행하게끔 명령어를 입력하는데 사용되어집니다.

## Proxy

프록시 서버는 클라이언트가 자신을 경유해 다른네트워크 서비스에 간접적으로 접속할 수 있게 해주는 컴퓨터 혹은 응용프로그램을 의미합니다. 이때 해당 프록시 서버가 수행하는 기능을 가리켜 '프록시'라고 하며 프록시 기능을 수행하는 '것'을 프록시 서버라고 칭합니다. 기존의 프록시를 포워드 프록시 라고 한다면 이와는 반대로, 리버스 프록시는 클라이언트를 대

신해서 한개 이상의 서버그룹에 클라이언트의 요청을 전달합니다. 그리고 해당 요청에 대한 응답을 리버스 프록시 서버를 통해서 클라이언트에게 전달하게 되는데 이를 통해서 클라이언트는 내부망을 인지하지 못하게 되는 효과를 거두게 됩니다.

## WAS(Web Application Server)

웹 애플리케이션 서버는 동적 서버 콘텐츠를 수행하는 것으로 부터 웹서버와 구별된다. 웹서버는 HTML과 같은 정적문서에 대해 HTTP규약에 따라 클라이언트와 통신을 하는것이 주 역할이라면 웹 애플리케이션 서버의 주 역할은 웹서버가 수행하지 못하는 동적 기능을 수행하여 데이터를 처리합니다. 예를 들면 보안, 트래픽관리, 사용자관리, DB관리, 트랜잭션 관리 등이 있습니다.

## SSH(Secure Shell)

네트워크 상의 다른 컴퓨터에 로그인하거나 원격시스템에서 명령을 실행하고 다른 시스템으로 파일을 복사 할 수 있도록 하는 그러한 응용프로그램 또는 프로토콜을 의미합니다. SSH는 안전한 원격 통신을 위해 사용되어진다. 기존에 Telnet을 사용했었는데 해당 통신에서는 데이터를 암호화하는 과정이 없기때문에 네트워크 상의 누군가 통신을 가로채 통신내용을 엿볼 가능성이 있다는 문제가 존재했습니다. 이러한 보안상의 문제 때문에 SSH가 설계되었고 주로 사용되어지고 있습니다. 기본적으로 22번 포트를 사용하며 리눅스, 유닉스 시스템에서 주로 사용됩니다.

## Cache

캐시는 어떠한 두 객체 사이에 위치해 데이터의 전송속도 혹은 데이터에 대한 접근속도를 극복하기 위해 사용되어집니다. 자주 사용되어지는 데이터를 좀더 빠르게 접근 할 수 있는 임시 장소에 복사해 둔다면 해당 데이터가 필요할때마다 해당 데이터를 가져오는 시간이 좀더 빨라질 것입니다. 이처럼 캐시는 데이터에 대한 접근 속도를 개선시켜 시스템의 효율성을 위해서 사용되어집니다.

## load balancer

대용량의 서비스를 운영하려면 각각의 서버로 이루어진 서버그룹에 대한 대용량트래픽의 부하 분산이 필수적인데 이때 해당 서버그룹의 각각의 서버에 부하를 분산시켜주는 기능을 하는것이 바로 이 로드밸런서 입니다. 네트워크 상단에 L4스위치 가상서버가 서버로 들어오는 패킷을 실제 서버로 균일하게 부하를 분산 시키며 작동합니다. 이는 고가의 서버로 확장하지 않고 저렴한 다수의 서버를 사용하여 서비스를 운영하는게 가능하게 함으로써 경제적으로 비용을 절감 할 수 있으며 또한 서버 확장이 용이하다는 장점을 가집니다. 여기서 L4스위치라 함은 OSI 7계층 모델에서의 4계층인 전송계층에서 작동하는 스위치라는것을 의미합니다.

## Instance

인스턴스라 함은 정해진 기능을 수행하는 일련의 '복제품'들이라고 할 수 있습니다. 일반적으로 한대의 서버에서 어떠한 기능을 수행하는데에 있어서 하나의 프로세스가 그 기능을 도맡아 수행할 수 있지만 여러개의 프로세스가 그 기능을 함께 수행 할 수도 있습니다. 이때 여러개의 프로세스들은 각각이 인스턴스 이며 모두 같은 기능을 수행합니다. 이때 하나의 서버에서 여러개의 프로세스들이 같은 기능을 수행하기 때문에 Port 충돌이 일어날수 있습니다. 그렇기 때문에 각각의 Port를 다르게 설정해 주어야 합니다.

## Docker

도커란 컨테이너 기반의 가상화 플랫폼입니다. 다양한 프로그램들과 실행환경을 컨테이너로 추상화하고 동일한 인터페이스를 제공하여 프로그램의 배포 및 관리를 용이하게 만들어줍니다. 어떤 프로그램이든지 컨테이너로 추상화 할 수 있고 어디서든 실행 할 수 있습니다. 여러개의 컨테이너들은 동일한 커널을 공유하지만 각 컨테이너는 오직 사전에 정의된 양의 리소스만 점유할 수 있습니다. 이 도커를 통해 컨테이너를 만들고 관리하면 다수의 응용프로그램 등이 자율적으로 하나의 물리 머신이나 여러개의 가상 머신 위에서 구동될수 있게 되므로 고도의 분산시스템을 생성하는 일이 단순해집니다.

## Container

도커에서 이용하는 컨테이너란 격리된 공간에서 프로세스가 동작하는 기술입니다. 가상화 기술의 일종이지만 기존의 가상화와는 다른 방식을 취합니다. 이 프로세스 격리방식을 이용한 컨테이너 기술을 통해 CPU나 메모리는 딱 해당 컨테이너에서 작동하는 프로세스가 필요한 만큼만 사용하게되고 이는 성능의 최적화를 이끌어냅니다. 하나의 서버에 여러개의 컨테이너를 실행하여 분산시스템을 구축할 수 있으며 각각의 컨테이너들은 서로 영향을 미치지 않고 독립적으로 실행됩니다.