**클라우드 #3주차 과제**

21660019 3A김세진

**1. 개발단계 SDLC**

**2. 가상화, 클라우드 차이**

**3. CAPEX, OPEX**

**4. HA**

**5. SLA, SLM 클라우드 관점**

**6. 하이퍼바이저 (클라우드 가상머신에서)**

**7. storage NAS,SAN,DAS서버**

**8. 메타데이터(data of data)**

**9. 가상화**

**10. SDN**

**11. IPS, IDS**

**12. 미들웨어 (클라우드에서의)**

**13. DMZ**

**1. 개발단계 SDLC**

1-(1) SDLC의 개념

SDLC는 Software Development Life Cycle, 즉, 소프트웨어 개발 수명주기로써, 시스템 엔지니어링, 정보 시스템, 또는 소프트웨어 공학에서 정보 시스템을 계획, 개발, 시험, 채용하는 과정을 뜻하는 용어이다. 요구사항 분석→설계→개발→테스트→운영 단계로 구성되어있다.

1-(2) 내가 생각하는 SDLC의 개념

새로 만드는 소프트웨어는 일종의 맞춤형 제품이라고 생각할 수 있다. 특정한 수요에 의해 요구되는 사항을 명시적으로 보여주면서 어떤 어떤 기능이 들어가야 한다. 무엇이 꼭 필요하고, 특정 상황에서는 어떻게 작동이 된다는 식의 요구를 분석하고, 그것에 맞게 설계도를 작성하여, 개발하고, 개발 후에는 버그나 보안상 문제점 등이 있는지 테스트한 후, 운영을 하는 방식일 것이다. 이해하기 쉽게 우리가 간단한 움집을 짓는다고 생각해보자, 마구잡이로 짓기 시작한다면 중간에 삐끗하는 순간 모든 것이 무너질 것이고, 시간이 더 걸릴 것이다. 하지만 처음에, 분석을 하고, 설계도를 만들어서 짓기 시작한다면, 처음에는 늦을지라도 후에는 문제가 거의 없이 완벽히 지어질 것이다. 소프트웨어도 마찬가지일 것이다. 처음에는 준비하는 시간이 걸릴테지만, 그만한 완성도를 보여줄 것이다.

1-(3) 현업에서는 어떻게 쓰일지  
 요즘 이슈가 되고 있는 배달어플을 예로 들어보겠다. 처음에 그런 배달어플을 만들어달라고, 소프트웨어를 개발하는 회사에 주문을 했다고 가정해보자, 요구사항 분석으로 우선, 최고 관리자용, 가게주인용, 손님용의 어플이 연결되야 할 것이고, 그것의 정보에 대한 데이터베이스와 연결, 주문, 결제, 위치 등의 많은 요구가 있을 것이다. 이런 요구대로 설계도(UML)를 만들어서, 모든 기능이 어떻게 작동되는지 이해하기 쉽게 만들어서 서로 협업해서 만들고, 만들어진 어플을 알파,베타 서비스 등을 이용해 테스트를 거쳐서 해당 기간동안 문제점을 수정하고, 문제가 없을 시에 완전한 서비스를 실행하는 방식으로 운영할 것 같다.

**2. 가상화, 클라우드 차이점**

2-(1) 가상화, 클라우드의 개념

가상화 : 전산자원의 물리적 특성을 숨기고 사용자에게 추상화된 가상 자원을 제공하는 기술이다.

클라우드 : 인터넷을 매개로 하여 서비스 형태로 제공되며, 가상화된 자원에 기반하여 규모 변경이 동적으로 이루어지는 컴퓨팅 방식이다  
  
2-(2) 내가 생각하는 가상화, 클라우드의 개념과 차이점  
 가상화는 하나의 서버에 하이퍼바이저라는 소프트웨어를 돌리고 그 위에 여러 os를 돌리는 것을 의미한다. 이러한 방식의 이유는 자원을 효율적으로 쓰기 위함인데, 본래의 서버는 하나의 목적에 한서버를 사용했고, 트래픽 과부하를 방지하기 위해서 필요보다 2배 이상의 용량을 설치했다. 이로 인해 평소에는 쓰지 않는 리소스에 인해 낭비가 발생했고, 이러한 점을 고치기위해 나온 것이 바로 가상화이다. 가상화는 하나의 서버를 여러 조각으로 나눠서 여러 개의 서버가 있는 것처럼 사용할 수 있었고, 이런 방법을 통해 리소스를 늘리거나 줄이는 것의 시간을 줄일 수 있었고, 낭비없이 효율적으로 서버를 활용하게 되었다.  
 클라우드라는 것은 가상화라는 기술의 발전에서부터 나온 것인데, 가상화가 하나의 서버를 여러 서버로 사용할 수 있게 바뀜으로써, 그 여러 서버 중 하나를 사용하고자 할 때 사용할 수 있는 것을 의미한다. 내가 만약 서버를 설치한다고 가정한다면, 나는 물리적인 서버에, 네트워크를 위한 장비, 시설, 기타 관련된 모든 것들을 설치하고, 시험하고, 수많은 절차를 걸쳐야한다. 하지만 클라우드 컴퓨팅을 한다면, 기존의 어딘가에 있는 서버에서 일부분을 할당받아 운용하면 되는 것이다.   
 따라서 가상화와 클라우드의 차이점은 차이점이라고 보기에는 애매하지만 가상화라는 기술의 발달로 클라우드 서비스라는 것이 가능해졌고, 가상화는 하나의 서버를 여러 서버로 만들 수 있으며, 클라우드는 그 나뉘어진 여러 서버 중 하나를 내가 할당 받을 수 있는 서비스인 것이다.  
  
2-(3) 현업에서 어떻게 사용할지  
 현업에서 가상화는 기존의 서버를 더 효율적으로 사용하기 위해서 나왔기 때문에, 서버의 비용을 줄이기에 썼을 것 같다. 예를 들면, 기존의 사용량이 적지만 꼭 필요한 10대의 서버를 하나의 서버만으로 충족시키면서 기존 9개의 서버를 줄임으로써 발생하는 막대한 이득이 있을 것이다. 클라우드에 관해서는 위에서도 잠깐 말이 나왔지만, 자신의 독자적인 서버를 운영하려고 엄청난 돈을 쓰지 않아도 된다. 자기가 원하는 만큼 늘리고 줄이는 것도 할 수 있으며, 이러한 방법은 새롭게 사업을 시작하는 데 발생하는 금액을 최소화할 수 있고, 트래픽을 잘못 계산하여 서버가 터지는 상황에 대비를 할 수 있다.

**3. CAPEX , OPEX**  
 3-(1) CAPEX, OPEX의 개념

CAPEX란 Capital Expenditure의 약자이다. 미래의 이윤을 창출하기 위해 지출된 비용이다.

OPEX란 Operating Expenditure의 약자이다. CAPEX를 통해 취득한 자산을 유지 보수하거나 운영하는데 들어가는 비용을 말한다.  
  
 3-(2) 내가 생각하는 CAPEX, OPEX의 개념  
 어려운 말 같지만 매우 쉽게 말하자면, CAPEX는 초기투자비용이라고 생각하면 될 거 같고, OPEX는 운영유지비라고 생각하면 될 것 같다. 아파트를 예시로 들자면, 초기투자비용은 아파트 건설에 들어가는 자재, 인력, 시간 등 모든 자원의 비용이고, 운영 유지비는 아파트 리모델링, 관리자, 보안, 시설 보수등에 들어가는 자재, 인력등 모든 자원의 비용이다.   
  
 3-(3) 현업에서 어떻게 쓰일지  
 과연 클라우드 컴퓨팅에 이러한 CAPEX, OPEX가 어디서 쓰일까를 생각해보았는데, 기존의 서버와의 관계를 나타내기에 이 용어가 가장 효과적일 것 같았다. 기존의 서버는 초기 투자비용이 굉장히 많이 들어가는 반면에, 이후 운영비용은 관리만 잘한다면 많이 줄기 때문이다. 하지만 클라우드 컴퓨팅은 초기비용이 매우 감소하지만, 매달 사용료를 지불하는 것에 운영비가 나가기 때문이다. 따라서 이러한 CAPEX와 OPEX사이의 비용을 계산해서 어떤 식의 운영을 해 나가냐를 알아내는 것은 사업의 시작에 가장 중요한 요소중의 하나일 것이다.

**4. HA**  
 4-(1) HA의 개념  
 HA는 high availabilty의 줄임말이며, 고 가용성이라는 의미를 가진다. 정보기술에서 HA란 바람직한 정도로 긴 시간동안 지속적으로 운영이 가능한 시스템이나 컴포넌트를 가르킨다.  
  
 4-(2) 내가 생각하는 HA의 개념  
 서버에서 이러한 고가용성을 보장하기 위해서는 장애가 생겼을 때, 지속적으로 운영이 가능하게 하려면, 서버를 2대를 묶어야 한다. 쓰던 서버가 장애가 발생하면 다른 서버가 그 업무를 수행하는 방식으로 이 HA를 보장할 수 있다.   
  
4-(3) 현업에서는 어떻게 쓰일지  
클라우드 컴퓨팅을 지원해주는 서버도 당연하게 물리적인 서버이다. 따라서 이러한 서버가 망가지면, 그곳에서 지원하던 모든 서버가 터진다는 것을 의미할 것이다. 따라서 이러한 상황을 막기위해 HA를 보장해야 할 것이다. 한 서버가 고장이 나면 다른 서버로 그 서버를 대신하는 방식으로 말이다.

**5. SLA, SLM (클라우드 관점)**

5-(1) SLA, SLM의 개념

SLA란 어떤 서비스가 제공될 것인지를 측정이 가능한 지표로 명시한 문서이며, 서비스 공급자와 사용자간에 서비스에 대한 측정지표와 목표 등에 대한 협약서이다.  
 SLM이란 SLA가 잘 적용되고 있는지 확인, 검토, 모니터링 하는 일이며, SLA에서 정의한 바에 대해 실제 서비스 수준을 측정/평가하고 개선 방안을 마련하는 과정이다.  
   
5-(2) 내가 생각하는 SLA, SLM의 개념  
 SLA는 소프트웨어를 사고 파는 사람들 간의 계약서 같다. 주로 큰 거래에서 예상치 못한 실수나 불만족할 것에 대비해 계약을 하는데, 소프트웨어도 마찬가지일 것이다. 원하는 만큼의 성능이 보장되지 못하거나 갑작스런 오류에 손해가 나는 것을 방지하기 위해 미리 약속을 해놓는 것이다.   
 SLM은 이러한 계약이 잘 지켜지는지 꾸준히 확인하는 과정이다. 이러한 과정에는 실제로는 잘 되고 있는지, 개선할 방안이 무엇인지에 대한 방법들을 찾는 것이다.  
  
5-(3) 현업에서는 어떻게 쓰일지

클라우드에서 이 SLA, SLM이 어떻게 쓰일지를 생각해 보았다. 먼저 클라우드라는 것은 다른 서버의 자원을 가상화해서 내 것처럼 쓸 수 있는데, 이 과정에서 나는 특정 규모의 자원을 특정 시간 등 여러가지 조건을 붙여서 사용 할 수 있다. 이때 우리는 원하는 만큼 성능을 부여받지 못할 수도 있고, 또한 본래의 서버의 문제가 발생함에 따라 우리의 서버도 문제가 발생할 수도 있기에, 이럴 때에 대비하여 계약서를 쓸 것이고, 이것이 SLA, 즉 특정 서비스의 만족도에 대한 계약이 될 것이다. 그리고 이러한 계약을 잘 지켜지고 있는지 확인하는 과정이 SLM이고, 이 과정은 서비스가 문제없이 잘 돌아가고 있는지 등의 여러 지표를 가지고 모니터링 되서 확인이 될 것이다.

**6. 하이퍼바이저 (클라우드 가상머신에서)**  
  
6-(1). 하이퍼바이저의 개념  
 하이퍼바이저는 호스트 컴퓨터 1대에서 다수의 운영체제를 실행하기 위한 논리적 플랫폼을 말하며, 다양한 운영체제가 설치된 가상 서버를 여러 대 실행할 수 있게 도와주는 시스템, 프로그램이다.  
  
6-(2) 내가 생각하는 하이퍼바이저의 개념(클라우드 가상머신에서)  
클라우드라는 것은 결국에 다른 서버의 자원을 내 것처럼 자원을 할당받아 쓰는 것을 의미하는데 이 원래 서버가 여러 서버처럼 사용 가능하게 하는 소프트웨어가 하이퍼바이저이다. 즉, 이 하이퍼바이저라는 것은 클라우드 컴퓨팅을 실현시키는데 가장 기본적인 기술을 가진 소프트웨어라고 볼 수 있다. 단순히 하나의 서버가 여러 개의 서버의 역할을 수행한다는 말보다 더 큰 의미를 갖는데, 어떤 곳에는 많은 부분을 할당할 수 있고, 어떤 곳은 적은 부분을 할당함으로 써 유동적인  
서버 관리가 되고, 또한 효율적인 관리를 할 수 있는 소프트웨어인 것이다.  
  
6-(3) 현업에서 어떻게 쓰일지  
 위의 내 생각에서 볼 수 있듯이 현업에서 실제 서버에 하이퍼바이저를 적용하고, 개인이 신청한 OS를 그 위에 올려서 서버를 쓸수 있게 만들것이다. 개인은 이러한 서버의 일부분(자원)을 자신이 설정하여 자신의 서버가 있는 것처럼 느낄 수 있고, 이러한 가상 서버로 자신의 어플리케이션을 만들거나 사업을 진행할 수 있다. 더 탄력적이게  
  
6-(4) 물리서버, 논리서버 자원통합 등의 개념에서 생각   
 물리서버라는 것은 우리가 아는 서버이다. 일반적인 많은 클라이언트(가정집 컴퓨터 등)에게 데이터를 주고 받는 좋은 컴퓨터라고 생각하면 된다. 논리 서버는 이 물리 서버안의 네트워크상에서 동작하는 서버라고 생각하면 되는데 WEB서버, DB서버를 예시로 들 수 있다.  
 먼저 한 물리서버를 여러 서버로 나누는 것을 위에서 하이퍼바이저를 이용한다고 설명했다. 이렇게 물리서버를 나눠서 나뉜 서버도 논리서버라고 볼 수 있다. 따라서 이 경우에 하이퍼바이저는 하나의 물리서버를 많은 논리서버로 나눠서 작동시키는데 필요한 소프트웨어라고 볼 수 있다. **7. Storage NAS,SAN,DAS**

7-(1) Storage의 개념  
 storage는 컴퓨터에서 자료를 일시적으로, 또는 영구히 보존하는 장치를 말한다. 대체로 비휘발성의 기억 장치를 의미한다.   
 + Network Storage의 개념  
 네트워크가 발전함에 따라 하드디스크를 떼어다 복사하는 것보다 네트워크를 통해 전달이 더빨라졌기 때문에 네트워크를 통해 데이터를 저장, 복사 등 디스크 작업을 할 수 있는 저장 장치이다.  
   
NAS(Network Attached Storage) : 네트워크 결합 스토리지로, LAN을 통해 Storage와 Server를 접속하는 방식이다.  
DAS(Direct Attached Storage) : 서버와 외장형 저장장치를 전용 케이블로 직접 접속하는 방법이다  
SAN(Storage Area Network) : 광 저장장치영역네트워크라고 부르며, 서버가 fiver channel switch를 통해 storage를 연결하는 방법이다.

7-(2) 내가 생각하는 Storage의 개념   
제일 먼저 있던건 DAS다 DAS는 값비싼 외장 저장장치를 서버에 연결한 형태로, 하드웨어적인 연결을 함으로써 서버의 용량을 늘렸다. 하지만 네트워크 기술의 발달로 이러한 방법으로 서버의 용량을 늘리거나 저장하는 방식은 돈이 더 많이 들며, 다른 서버와의 데이터교환이 힘들었기에 새로히 NAS와 SAN이라는 개념이 등장했다. 둘 다 DAS와 차이점은 너트워크를 통해 저장 장치를 연결했다는 점이고, NAS는 근거리 통신망인 LAN을 통해 같은 네트워크를 가진 곳에서 NAS에 저장된 파일을 공유할 수 있으며, SAN의 경우에는 정화히는 이해하기 힘들었지만 여러서버에 걸친 저장장치를 네트워크를 통해 연결한 것을 말하는 것 같다.

7-(3) 현업에서 어떻게 쓰일지

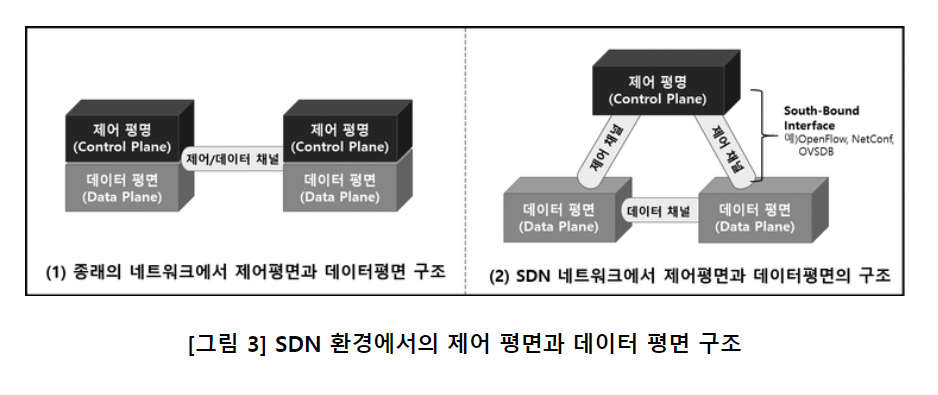
우선 DAS의 경우 예전에 저장공간 물리적으로 늘리거나 데이터의 백업을 위해서 복사하여 물리적으로 저장하기 위해 쓰였을 것 같다.  
 그리고 새로 나온 NAS는 종종 보았는데 우리 학교에서도 NAS라는 것을 보기도 했고, 직접 클릭까지 해보아서 약간 유추를 할 수 있었다. NAS는 같은 LAN, 네트워크로 연결되 있을 때, 서버가 가진 데이터를 그 네트워크를 통해 공유할 수 있었던 것 같다. 우리는 이것을 통하여 시험 때 필요한 데이터(사진, 텍스트파일)등을 그 저장 장치에서 복사해 올 수 있었다. 이러한 방식으로 특정 네트워크 안의 사람이라면, 이 스토리지를 통하여 데이터를 공유하는 방식에 많이 쓰일 것 같다.   
 SAN은 이해하기가 조금 난해한 말들이 많아서 이해가 맞는지 모르겠다. 위에서 말했던 것처럼 NAS, SAN은 DAS와 달리 네트워크를 통해 데이터를 옮기는 것이 다르다고 했다. 그럼 NAS와 SAN의 차이점은 무엇일까. 나는 그것을 DAS의 기능을 나눠가진 것으로 해석했다. DAS는 저장공간을 물리적으로 늘리거나, 데이터를 백업하기 위해서 복사하는 역할을 했다고 생각한다. 이중 데이터를 백업하고 복사하는 역할을 NAS가 맡고 SAN은 저장공간을 늘리기 위해 여러 storage를 네트워크로 연결한 것이라고 생각한다.

**8. 메타데이터**  
 8-(1) 메타데이터의 개념  
 데이터를 설명하는 데이터이다. 어떤 목적을 가지고 만들어진 데이터이다.  
  
 8-(2) 내가 생각하는 메타데이터의 개념  
 개발언어를 몇 번 접하다 보면, 메타데이터라는 말을 듣게 된다. 쉽게는 데이터를 설명하는 데이터라는데, 나는 이 말을 예시를 듣고 나서야 이해가 되었다. 이 메타데이터라는 것은 2가지로 나뉘는데 첫번째는 사람을 위한 메타데이터, 그리고 프로그램을 위한 메타데이터였다. 사람을 위한 메타 데이터는 가장 쉽게 주석이라고 했는데, 주석은 // 같은 표시로 실제 프로그래밍에 영향을 주지 않고 사람이 보기에 이해하기 쉽게 표시해 논 것이다. 이와 같이 메타데이터라는 것은 어떤 목적을 가지고 데이터를 설명하기 위한 데이터라는 것을 이해할 수 있다.  
  
 8-(3) 현업에서 어떻게 쓰일지

현업에서는 위의 사례와 같이 협업을 할 때 어떠한 코드가 무슨 코드를 나타내는지, 이 변수가 무엇인지를 만들어서 붙여 놓는 방식으로 사람을 위한 메타데이터로 쓰일 수 있고, 프로그램을 위한 메타데이터의 예시로는 어노테이션이라는 기능을 들 수 있는데, 이 데이터의 경우에는 프로그램에게 이 어노테이션이 붙었을 때 어떤 역할을 추가적으로 해야된다는 것을 말한다. 이러한 작업은 프로그램이 돌아갈 때, 프로그램이 오류가 적게 유기적으로 돌아가는데 도움이 될 것이다.

**9. 가상화**  
 9-(1) 가상화의 종류에 따른 개념  
서버 가상화 : 하나의 물리 서버는 여러 개의 가상 머신으로 전환될 수 있으며, 각각의 가상 서버는 마치 독립적인 물리 서버처럼 동작한다.  
데스크톱 가상화 : 물리적으로 존재하진 않지만 실제 작동하는 컴퓨터 안에서 작동하는 또 하나의 컴퓨터를 만들 수 있는 기술이다.  
네트워크 가상화 : 기존에 하드웨어로 제공되던 네트워크 리소스를 소프트웨어로 추상화하는 것을 말한다.   
  
 9-(2)내가 생각하는 가상화의 개념  
 서버 가상화는 하나의 물리적인 서버를 하이퍼바이저를 사용하여 여러 서버처럼 활용할 수 있게 바꾸는 기술을 말하며, 우리가 이전에 가상화라는 말에 쓸 때 주로 이 용어를 썼다.   
 데스크톱 가상화를 설명하기 전에 데스크톱은 흔히 우리의 데스크톱 컴퓨터를 의미하는데 이것의 가상화라는 것은 이런 데스크톱의 본체가 존재하지 않다는 것이다. 즉, 본체없이, 마우스 키보드, 모니터만 있어도 내 컴퓨터를 사용하듯 다른 컴퓨터 본체의 일부를 사용하여 내 컴퓨터인 것 처럼 사용할 수 있다는 것이다.  
 네트워크의 가상화는 SDN의 개념과 비슷해 보이는 것 같은데, 하드웨어로 존재하던 네트워크의 장비를 소프트웨어적으로 처리함으로써 보이지 않게 되어서 가상화라고 말하는 것 같다.   
  
 9-(3) 현업에서 어떻게 쓰일지

서버 가상화의 경우에 클라우드 서비스를 제공할 때 하나의 서버가 여러 서버로 나눈 것 거처럼 작동하는데 사용할 것이고, 데스크톱 가상화는 조금 신기했는데, 우리가 이사를 할 때, 컴퓨터를 옮길 필요가 없어 질 것 같다. 네트워크 가상화의 경우에는 물리적인 네트워크를 일일이 연결하는 방법에서는 속도가 줄기 때문에 이런 것을 소프트웨어적으로 연결하면서 큰 네트워크 형성에 도움을 주었을 것 같다.

**10. SDN** 10-(1) SDN의 개념  
 소프트웨어 정의 네트워킹(Software defined networking, SDN)은 개방형 API(오픈플로우)를 통해 네트워크의 트래픽 전달 동작을 소프트웨어 기반 컨트롤러에서 제어/관리하는 접근방식이다. 트래픽 경로를 지정하는 컨트롤 플레인인과 트래픽 전송을 수행하는 데이터 플레인이 분리되어 있다. 따라서 네트워크의 세부 구성정보에 얽매이지 않고 요구사항에 따라 네트워크를 관리할 수 있다.  
  
 10-(2) 내가 생각하는 SDN의 개념  
 기존의 네트워크 장비 내부에는 패킷을 받아들이는 컨트롤 플레인과 패킷을 전송하는 데이터 플레인이 동일선상에서 정해진 채널로 서로 데이터를 주고 받았지만, SDN은 컨트롤 플레인을 데이터 플레인과 분리하여 그 위에 두는 것이다. 이러한 방식은 일종의 중앙 제어식으로 컨트롤 플레인이 많은 데이터 플레인을 효율적으로 관리 할 수 있는 것 같다.   
  
 10-(3) 현업에서 어떻게 쓰일지

구글과 페이스북 같은 기업들을 보면 수많은 컴퓨터에서 각자의 데이터를 쏟아내며 트래픽을 높일 것이다. 이런 처리를 다루는 수십만 대의 서버를 연결시키기에 물리적으로 한 개에서 한 개로 이어지는 네트워크 구조는 매우 느린 처리 방식일 것이다. 따라서 등장한 게 SDN인 것 같으며, 이러한 SDN은 많은 서버를 이어주는데 그 속도나 제어면에서 매우 효율적으로 사용될 것 같다.

**11. IPS, IDS**

11-(1) IPS, IDS의 개념  
IPS : (Intrusion Prevention System) 직역하면 침입 예방 시스템이며, 비인가 프로그램이나 해커의 공격에 대해서 미리 rule파일에 정의가 되있다면 차단을 시키거나 통과시켜주거나 하는 식의 설정이 된 시스템이다.  
   
IDS : (Intrusion Detection System) 직역하면 침입 탐지 시스템이며, 비인가 프로그램이나 해커의 공격이 이뤄지면 해당 트래픽을 복사하여 로그를 만들고 해당 로그파일을 가지고 관리자가 대응을 할 수 있게 하는 시스템이다.   
  
 11-(2) 내가 생각하는 IPS, IDS의개념  
일단 IPS, IDS는 모두 비인가 프로그램이나 해커의 공격을 막기 위해서 만들어진 보안관련 소프트웨어인 것 같다. 나는 이것을 메이플이라는 게임으로 이해를 해 보았다. 이 RPG게임의 특성상 매크로라는 프로그램은 꽤나 매력있게 다가왔기 때문에 매크로에 대해서 공부를 조금 한적이 있는데, 그 때에 그 게임의 보안 프로그램에서 들은 것이 있다. 클라이언트의 메모리를 탐색하여 매크로 프로그램과 관련된 프로그램이 실행 중이라면, 서버와 해당 클라이언트와의 연결을 끊어 버리는 식이었다. 이 방식은 IPS라고 볼 수 있는데, 미리 짜여진 rule파일에 이러한 작업들이 미리 되있기 때문에, 매크로 프로그램이 실행 중이라면 끊어지는 것이다. 해결 방법은 매크로 프로그램을 허용가능한 프로그램으로 속이면 간단히 통과가 되는 식이었다. 간단히 매크로 프로그램을 internet explorer가 실행중인 것처럼 처리하는 방식이었다. 이러한 경우에 아무런 제지없이 프로그램이 사용되어졌는데, 이러한 방식의 IPS의 취약점 때문에 아마 더 강한 개념이 IDS인 것 같다. 실시간으로 불법 프로그램이나 해커의 공격의 트래픽을 복사하여 로그를 만들고 사람이 직접 차단하거나 대응할 수 있게끔 말이다.

11-(3) 현업에서 어떻게 쓰일지  
 IPS는 위의 예시에서 보듯이 어떤 서비스의 보안 프로그램과 같은 곳에 많이 사용될 것 같다. 수 많은 돈이 오가기 때문에 보안이 철저해야 하며, 또한 수많은 트레픽이 몰려들기 때문에 관리자가 일일이 처리하기가 곤란하기 때문에 일차적으로 부정프로그램이나 해커를 막는데 도움이 될 것이다.   
 IDS는 왠지 IPS에서 처리하지 못한 심오한 단계의 보안을 처리하는데 사용할 것 같다. 예시를 들어보자면, 범국가적인 해킹에서 처리를 살펴보면 이해가 쉬울 것이다. 우리는 종종 어떤 국가 기관이 뚫렸다던가 어떤 기업에서 데이터가 유출 됬다는 소식을 가끔 듣는다. 그런 기관이나 기업이 과연 보안에 투자를 하지 않고 당한 것일까? 내가 생각하기에는 아니다. 그래도 뚫리는 것이 보안이기 때문에 이중적으로 막고 방어하는 장치가 IDS인 것 같다.   
 따라서 내가 생각하기에 IPS라는 시스템 위에 IDS시스템까지 놓아서 보안을 철저히 하는 식으로 사용할 것 같다.

**12. 미들웨어**

12-(1) 미들웨어의 개념  
 미들웨어란 시스템 소프트웨어와 응용 소프트웨어 사이 또는 두 가지 다른 종류의 응용 프로그램 사이에서 조정 및 중개 역할을 하는 프로그램을 의미한다. 시스템에 분산돼 있는 응용프로그램과 데이터, 웹 서버간 연결을 최적화 시켜주는 역할을 하는 것이다. 또한 클라이언트(사용자) 프로그램과 서버 프로그램 사이에 존재하면서 클라이언트와 서버간에 연결을 유지/관리하면서, 클라이언트의 작업 처리요구를 서버에 전달하는 일을 한다.

12-(2) 내가 생각하는 미들웨어의 개념  
 매우 간단히 이해한 바를 말하자면, 어떤 프로그램과 어떤 프로그램 사이의 가운데서 운영이나 연결에 중개역할을 수행함으로써 효율적인 관리를 해주는 소프트웨어를 미들웨어라고 하는 것 같다. 예시로는 DBMS, WAS등의 소프트웨어를 예로 들었는데, 이러한 소프트웨어는 각각 DB와 응용소프트웨어, 웹서버와 클라이언트사이의 중계를 하는데, 이 둘은 그 중간단계에서 각각의 소프트웨어가 효율적으로 쓰일 수 있게 가운데서 인터페이스 역할을 한다. 트레픽 과부화나 연결에 관한 문제들을 다루는 등의 역할을 하면서 말이다.   
  
 12-(3) 현업에서 어떻게 쓰일지

실제 클라우드 컴퓨팅에서 생각을 해보자. 실제 서버와 자신의 프로그램 사이의 연결을 적절히 문제없이 만들기 위해서 쓰일 것 같다.

**13. DMZ**

13-(1) DMZ의 개념

DMZ는 인프라 네트워크의 구성중에서 외부 인터넷망과 내부 인프라넷망의 사이에 위치하는 중간지대를 지칭한다. 즉, 인프라네트워크의 보안영역의 일부이다.

13-(2) 내가 생각하는 DMZ의 개념

회사를 생각해보면, 내부에 LAN으로 연결된 네트워크가 있고, 외부와의 네트워크는 인터넷을 통해서 연결을 한다. 이때 방화벽이라는 것을 통해서 불법적인 침입을 막는다. 하지만 내부의 데이터를 외부에 제공하는 것이 회사에 필요하다고 했을 때, 모든 방화벽을 해제하는 것은 위험이 꽤나 클 것 이다. 따라서 내부 방화벽과 외부 방화벽 사이에 DMZ라는 중간 지대의 서버를 설정하고, 외부로 들어내야 할 데이터를 DMZ에 저장한다면 외부에서 이 데이터로 접근할 수는 있지만, 내부로는 접속이 힘들기 때문에 보안의 문제에 적절한 방법이 된 것 같다.  
  
 13-(3) 현업에서 어떻게 쓰일지

위의 설명의 예시에서 설명한듯이 내부 네트워크와 외부 네트워크의 완충지대이기 때문에 서로의 데이터가 왔다 갔다 해야 하는 서비스를 가진 곳에 크게 쓰일 것 같다. 예를 들면, 배달 앱을 생각했을 때, 당연히 고객정보나 사업자의 정보는 내부 네트워크에 두어서 데이터가 드러나지 않게 하고, 상품명, 가격, 배달비 등의 데이터는 DMZ의 층에 두어서 외부와 주고 받을 수 있게 설설정하는 쓰일 것 같다.