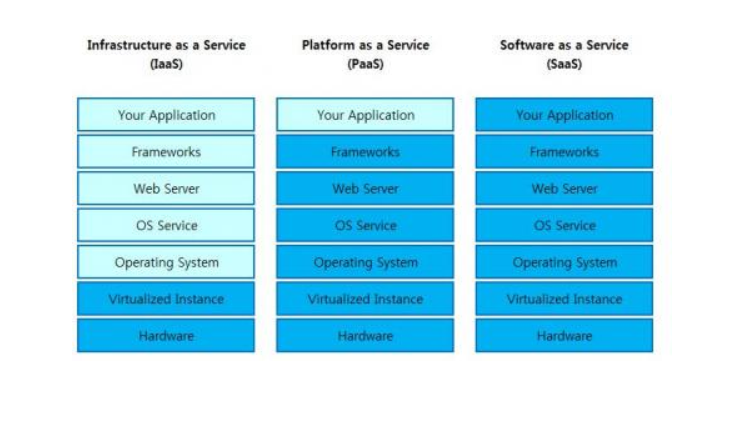
1번. As a Service  
클라우드 서비스의 유형을 나타내며, Infrastructure as a Service(IaaS), Software as a Service (SaaS), Platform as a Service(PasS) 3가지 유형으로 나뉜다.

설명 전에 알아두어야 할 것 : 서버를 운영하기 위해서는 여러가지가 필요한데, 제일 하위단계부터 하드웨어(H/W), 가상인프라 자원(서버,스토리지,네트워크), 운영체제, 운영채제 서비스, 웹서버, 프레임워크, 자신의 어플리케이션이 있다.



1-가. 개념:   
Infrastructure as a Service(IaaS) : 서버, 스토리지, 네트워크를 가상화 환경으로 만들어, 필요에 따라 인프라 자원을 사용할 수 있게 서비스를 제공하는 형태이다.

Software as a Service(SaaS) : “on-demand software”로도 불리며, 소프트웨어 및 관련 데이터는 중앙에 호스팅되고 사용자는 웹 브라우저 등의 클라이언트를 통해 접속하는 형태의 소프트웨어 전달 형태이다.

Platform as a Service(PaaS) : 일반적으로 앱의 개발 및 시작과 관련된 인프라를 만들고 유지보수하는 복잡함 없이 고객이 어플리케이션을 개발, 실행, 관리할 수 있게 하는 플랫폼 제공 형태이다.  
  
1-나. 내가 생각하는 개념 : 클라우드 서비스를 개발자, 사용자가 어떻게 쓸지에 따라서, 다양한 방식을 제공하는 것 같다.   
 먼저, IaaS는 가장 기본적인 서버, 스토리지, 네트워크만을 제공하여, 개발자가 원하는 운영체제, 웹서버, 프레임워크 등을 사용하여 개발할 수 있게 해주는 서비스 같으며,  
 PaaS는 개발자가 자신의 프로그램만 개발하기만 하면 되게끔, 거의 다 만든 서비스,  
 SaaS는 심지어 어플리케이션까지도 사용자가 원하는 것으로 만들어서 제공한 것으로, 완전한 프로그램으로 봐도 될 것 같다.

1-다. 어떻게 쓰일지 :  
 IaaS는 개발자가 자유롭게 OS나 프레임워크를 쓰기 위해, 가장 기초적인 것만을 제공한 것 같다. 따라서 완전히 새로운 프로그램, 혹은 자유로운 형식으로의 개발을 하기 위해서 이용하는 서비스일 것 같다.  
 PaaS는 정해진 플랫폼을 제공하여 개발자가 보다 빠르고 편리하게 개발할 것이고, 이미 정해진 틀에서 자신의 어플리케이션만 적용시키면 되기 때문에, 오류가 줄어들고, 유지보수가 편해질 것이다. 따라서 많이 대중화된 것을 빠르게 만들거나, 응용하여 만드는데 유용할 것 같다.  
 SaaS는 개념에서 유추가 가능할 것 같은데, 클라이언트에게는 프로그램을 제공하고, 나머지 정보들은 중앙에 호스팅 된다고 했으니, 온라인 게임, 브라우저, 등 우리가 이미 접했던 것들 일 것 같다.

2번 오픈스택   
 2-가 개념 :

IaaS를 구축할수 있는 오픈소스 소프트웨어이다. 즉 일반적인 서버에서 클라우드 컴퓨팅 서비스를 생성하고, 실행할 수 있게 해주는 소프트웨어이다.

2-나 내가 생각하는 개념 :  
 우리가 어떤 프로그램을 쓰기 위해서는 해당 프로그램의 설치자(마법사)를 다운받아서 마법사의 도움을 받아 프로그램을 깔듯이 컴퓨터에 클라우드 서비스를 이용할 수 있게 오픈스택이라는 소프트웨어를 이용하여 기본적인 것을 구축하고, 설치하는 프로그램인 것 같다.

2-다 어떻게 쓰일지:   
위의 글과 비슷하게 오픈스택이라는 소프트웨어를 이용하여 클라우드 컴퓨팅을 실현시켜, 어떤 서버컴퓨터, 클라이언트 컴퓨터에 클라우드를 연결시킬 수 있을 것 같다.

3번 플랫폼   
3-가. 개념:  
 플랫폼은 응용 소프트웨어를 실행하기 위해 쓰이는 하드웨어와 소프트웨어의 결합이다. 플랫폼은 하나의 운영체제 또는 컴퓨터 아키텍처라고 단순히 말할 수 있으며 그 두 가지를 통칭해서 말할 수도 있다.  
3-나. 내가 생각하는 개념:  
 사전의 개념과는 달리 기차역을 생각해보면, 우리는 다른 역에 가기 위해서는 이미 정해진 노선을 따라가야 한다. 즉 이미 정해진 곳(역,플랫폼)으로 가서 올바른 방향으로 몸을 태운채 이동하기만 하면 되듯이, 이미 정해진 방식대로 짜기만 되는 플랫폼이 제공되고, 그것을 이용하여 개발이 손쉽고 편해지는 역할을 한다고 본다.(1번의 PaaS에서의 플랫폼)  
3-다. 어디에 쓰일지:  
 이미 우리가 써왔던 것들이 플랫폼일지 모른다. 자바를 배우면서 썼던 IDE인 이클립스, c,c++을 배우기 위해서 썻던 VisualStudio 같은 프로그램들이 플랫폼과 비슷한 느낌일 것 같다. 이미 짜여진 방식대로 쓰기만하면 개발이 완료되게끔 하는 프로그램과 같을 것 같다.

4번 클라우드 유형별 장단점  
4-가. 개념:  
 프라이빗 클라우드 : 제한된 네트워크 상에서 특정 기업이나 특정 사용자만을 대상으로 하는 클라우드 서비스의 자원과 데이터는 기업 내부에 저장된다. 또한 기업이 자원의 제어권을 갖고 있으며, 보안성이 매우 뛰어나다. 개별 고객의 상황에 맞게 클라우드 기능을 커스터마이징 할 수 있다는 장점이 있다. 대신에 초기 구축비용이 들고, 규모를 확장하기가 어렵다.  
 퍼블릭 클라우드 : 특정 기업이나 사용자를 위한 서비스가 아닌 인터넷에 접속 가능한 모든 사용자를 위한 클라우드 서비스 모델이다. 데이터나 기능, 서버 같은 자원은 각 서비스에서 사용자 별로 권한 관리가 되거나 격리되어, 서비스 사용자 간에는 전혀 간섭이 없다는 장점이 있다. 대신에, 정보의 보안성에 대한 문제가 항상 제기된다.  
 하이브리드 클라우드 : 퍼블릭 클라우드와 프라이빗 클라우드를 병행해 사용하는 방식이 여겨져 왔으나, 최근에는 개념이 모호해진 경향이 있어 가상 서버와 물리 서버를 결합한 형태를 말하기도 한다. 이럴 경우 퍼블릭 클라우드의 유연성, 경제성, 신속성과 물리 서버의 보안성, 안정성 등을 함께 취할 수 있는 장점이 있다.

4-나. 내가 CEO라면 어떤 유형의 클라우드를 선택할지   
 정확한 이해를 하기에는 아직 모르는게 많지만, 프라이빗 클라우드는 보안이 매우 뛰어나지만,무거운 느낌이 든다. 따라서 이러한 클라우드를 쓰기 적합한 곳은 은행이나, 돈, 국가기밀에 관련된 회사, 나라에서 쓰일 것 같고, 퍼블릭 클라우드는 자주 바꿀 수 있고, 혁신이 이루어지는 회사에서 자주 쓰일 것 같다. 따라는 내가 CEO라면 최신의 트렌드를 따라가며, 항상 혁신적으로 앞서나가는 회사를 만들기에 적합한 퍼블릭 클라우드를 사용할 것 같다.

5번 SLA란  
5-가. SLA의 개념  
SLA(Service Level Agreement)는 서비스 수준 계약이며, IT서비스를 제공하는 업체와 사용하는 업체간의 협약을 의미한다. 협약에는 서비스에 대해서 측정지표와 목표등에 대한 협약이 들어가있다.

5-나 내가 생각하는 개념  
한 소프트웨어를 만든 업체와 그 소프트웨어를 사용하고자 하는 업체가 계약을 한다고 하면, 사용하는 업체는 해당 제품이 잘 작동하고 충분히 오래 쓸 수 있는지를 걱정할 것이고, 파는 업체는 자신이 만든 제품의 장점들을 보여주면서 보증을 해야 할 것이다. 이러한 관점에서 상호간의 신뢰의 의미로서, ‘작동이 잘 안 된다’, ‘문제점이 많다’ 싶으면, 가격을 할인해주는 등의 협약을 맺는 것 같다.

5-다. 현업에서 어떻게 쓰이는지.. :  
 한 회사에서 어떤 소프트웨어를 사용하고자 할 때, 먼저 계약을 하고 쓸 것 같다.   
은행을 예로 들면, 은행의 서비스가 은행의 잘못이 없이 다운이 됬다면, 해당 서비스를 제공하는 체가 잘못한 것이므로, 일부 돈을 보상해주는 식으로 사용될 거 같다.

6번 하이퍼바이저란  
  
6-가. 하이퍼바이저의 개념:  
 하이퍼바이저는 호스트 컴퓨터에서 다수의 운영체제를 동시에 실행하기 위한 논리적 플랫폼을 말한다. 가상화 머신 모니터 혹은 가상화 머신 매니저라고 부른다.  
  
6-나. 내가 생각하는 개념:  
 비슷한 이름의 개념을 자바에서 들어본 것 같아서 설명하자면 자바에 JVM이라는 단어가 있다. 자바 가상 기계라는 것인데, 자바가 어떠한 운영체제에서 실행시켜도 JVM이 알아서 실행시켜주는 것을 의미한다. 이것과 비슷하게 하이퍼바이저라는 것은 운영체제를 잠시 바꿔줄 수 있는 역역할 하는 것 같다.   
  
6-다. 현업에서 어떻게 쓰일지:  
 내가 알기로 프로그램, 어플, 심지어 기본 명령어 조차도 운영체제에 따라 다르다는 것으로 알고 있다. 만약에 내가 window를 쓰고 있을 때, 리눅스의 프로그램을 깔고 실행하려고 하면, 안된다는 것을 의미한다. 이럴 때 하이퍼바이저를 써서 일시적으로 그 프로그램을 쓸 수 있게 하는 방식으로 쓰일 것 같다.

7번 마이그래이션이란?  
  
7-가. 마이그래이션의 개념 :  
 마이그레이션은 시스템 이관, 시스템 마이그레이션을 의미하며, 명령어 집합이나 프로그램의 이관을 동반한다. 어떠한 플랫폼에서 다른 플랫폼으로의 이관을 의미한다.  
  
7-나. 내가 생각하는 개념 :   
 간단하게 생각하면, 어떠한 목적이나, 내 컴퓨터가 고장나서, 컴퓨터에서 다른 컴퓨터로 파일을 옮겨야 한다고 가정해보자, 우리는 usb를 사용하거나 어떤 블로그, 이메일 등으로 파일을 옮길 수 있을 것이다. 아니면 메인보드를 때서 바꿔가는 식으로도 쓸 수 있고, 이러한 모든 행위가 마마이그레이 인 것 같다.  
  
7-다. 현업에서 어떻게 쓰일지 :  
 내가 생각하는 게 맞는지 검색을 해봤는데, 클라우드 컴퓨팅에 관한 것이 나왔다. 내가 생각하는 하드웨어적인 측면이 아니라 내 컴퓨터에서 클라우드 컴퓨팅 플랫폼으로 프로그램등을 이관 시킬 수 있다고 한다. 이러한 방식은 보다 안전하고, 빠르게 이관할 수 있을 것으로 보인다.

8 컴포넌트란?   
  
8-가. 컴포넌트의 개념 :   
 컴포넌트란 소프트웨어의 재사용을 위해서 나온 기술이다. 일종의 부품이라고 생각할 수도 있다.  
  
8-나. 컴포넌트와 모듈의 차이:  
 흔히 하드웨어는 각각의 독립된 기능을 가진 모듈로 이루어져 있다. 이 모듈에 이상이 있거나 업그레이드를 하려면 모듈을 교환해주기만 하면 된다.  
 하지만 소프트웨어에서는 여러 소프트웨어가 교환되게 만들어지지 않는다. 즉 호환이 이뤄지게 개발하지 않는다는 말이다. 이에 따라 재사용을 가능하게 하기 위해 나온 기술이 컴포넌트 기술이다.   
  
8-다. 내가 생각하는 개념, 차이 :  
 처음에는 컴포넌트가 부품 같은 것이라고 생각하고 있었다. 하지만 모듈과의 차이가 있다는 것으로 생각이 바꼈다. 자바를 예로 들자면, 클래스라는 것이 있고, 인터페이스라는 것이 있다.  
하나의 실행에는 여러 클래스로 이뤄져있다. 이와 같은 클래스는 하나의 부품이라고 한다면, 우리가 다른 기능을 수행하게 바꾸기 위해서는 아예 다시 클래스를 짜야 한다. 이때 우리는 처음에 인터페이스로 규격화를 시켜놓았다면 이 코드를 재사용할 수 있다. 이런 식의 방식으로 부품을 만드는게 컴포넌트인 것 같다.

9번 가상화란?  
  
9-가. 가상화의 개념 :  
 가상화는 하이퍼바이저를 사용하여 하나의 물리적 머신에 다수의 가상 머신을 만드는 것이다. 이러한 가상 머신은 하나의 머신의 컴퓨팅 리소스에만 의존하여 물리적 머신과 같이 작동하기 때문에, 가상화를 사용하면 IT조직은 단일서버(호스트)에서 여러 개의 운영 체제를 실행할 수 있습니다.  
  
9-나. 내가 생각하는 가상화 :  
 내가 이해한게 맞을지 모르지만, 이 가상화라는 것은 일종의 컴퓨터를 효율적으로 사용하는 방식인 것 같다. 예를 들면, 회사에 서버가 3개가 있고 서로 다른 일을 한다고 가정하면, 어떤 서버는 일을 많이 할수도 있고, 어떤 서버는 별로 할 일이 없을 수 있다. 이러한 상황일 때, 가상화를 사용하여 서버에 가상 머신을 만들어 다른 일을 할 수있게 만드는 것 같다.   
  
9-다. 현업에서 어떻게 쓰일지 :  
 아직까지 개념이 부족해서 잘 모르겠지만, 결국에 가상화라는 것은 리소스를 효율적으로 분배하고, 많은 일이 필요한 곳에는 많은 자원을, 적은 일이 필요한 곳에는 적은 자원을 배분하는 것 같다. 따라서 현업에서는 사용량의 변화가 급격한 곳에서 낭비를 하지 않고 자원을 사용할 때 쓰일 것 같다.