**클라우드 #2주차 과제**

21660019 3A김세진

**1. 플랫폼**

**2. On Demand**

**3. CRM**

**4. VPN**

**5. 오픈스택, 하이퍼바이저, Xen**

**6. 클라우드 기술 동향**

**7. ERP**

**8. API**

**9. DR**

**10. on-premise**

**11. EOS, EOL**

**12. SDN**

**13. Bare Metal**

**1. 플랫폼**

1-(1) 플랫폼의 개념

플랫폼은 전철역이나 철도역에서 화물이나 승객이 열차에 타고 내리기 위해 만든 시설물이다.  
 하지만, 소프트웨어에서는 소프트웨어가 구동 가능한 하드웨어 아키텍처나 소프트웨어 프레임워크(응용 프로그램 프레임워크를 포함하는)의 종류를 설명하는 단어이다. 일반적으로 플랫폼은 컴퓨터의 아키텍처, 운영체제, 프로그램 언어, 그리고 관련 런타임 라이브러리 또는 GUI를 포함한다.

1-(2) 내가 생각하는 플랫폼의 개념

내가 생각하기에, 플랫폼은 일종의 환경같다. 기차를 생각한다면, 기차를 운행하기 위해서는 기찻길을 깔고 기차에 탈 수 있는 시설을 만들어야 한다. 이런 시설을 플랫폼이라 하며, 이 플랫폼을 이용하여 기차, 사람, 화물 등이 이동한다.   
 이러한 관점을 소프트웨어 관점으로 바꿔보자, 초기에는 컴퓨터를 쓸 때, 알지도 못하는 명령어를 입력해가며, 컴퓨터를 썼지만, 운영체제가 깔린 지금은 컴퓨터가 어떻게 돌아가는지 모르는 사람도 컴퓨터를 쉽게 쓸 수 있고, 서로 만나며 교류를 한다.   
 또한, 개발자의 입장에서 프로그램을 짠다고 생각을 해보자. 우리는 기계어로 짤 수도 있겠지만, 너무 힘들다는 것을 알 것이다. 우리가 써본 Eclipse나 VisualStudio는 그러한 과정을 쉽게 할 수 있게 컴퓨터와 사람간의 교류를 이어주는 프로그램이다.   
 이러한 생각으로 나는 플랫폼이라는 것은 누군가와 누군가를 이어주고, 교류를 이어 줄 수 있게 미리 설치되어야 할 일종의 환경이라고 생각한다.

1-(3) 현업에서는 어떻게 쓰일지  
 이 플랫폼이라는 개념은 매우 넓은 의미의 개념인 것 같다. 따라서 예시의 종류가 다양할 것 같은데,  
 첫 번째로, 회사에서 어떤 운영체제의 컴퓨터를 쓸지 정하는 것을 들 수 있다. 회사의 업무나 그 회사의 협력회사들의 운영체제 등에 따라 같은 운영체제를 선택함으로써 호환성을 맞추는 것을 들 수도 있다.  
 두 번째 예시로는, 플랫폼을 개발 툴로 생각한다면, 어떠한 프로그램을 만드느냐에 따라서 해당 개발 툴(플랫폼)을 선택해야 한다. 자바로 프로그램을 짜려면 Eclipse를 쓸 것이고, C,C++,C#같은 경우에는 Visual Studio를 쓸 것이다.  
 이렇게 플랫폼이라는 것은 자신이 무엇을 해야 할지에 따라서 다양하고 많은 플랫폼을 선택할 수 있다.

**2. On Demand**   
2-(1) On Demand의 개념

On-Demand라는 단어는 소비자의 수요에 맞춰 즉각적으로 맞춤형 제품 및 서비스를 제공하는 경제 활동으로, 공급이 아닌 수요가 모든 것을 결정하는 시스템을 의미 한다.  
  
2-(2) 내가 생각하는 On Demand의 개념  
 기존의 기업들이 생산한 제품들은 많이 찍어내고, 그 많은 제품들을 다 팔아야 되었다.  
그때는 적당히 팔아도 이득이었겠지만, 현 시대를 생각해보자, 많은 물건, 많은 트랜드들, 심지어 국가간의 관계조차도 빠르게 변화하고, 어떻게 변화할지 모른다. 그러므로 기업의 입장으로써는 많은 손해가 나오는 것을 막고 싶어했을 것이다. 이러한 관점으로 등장한 것이 이 On Demand라는 서비스 일 것이다. 과하지 않게 소비자들이 원하는 수 많큼, 자원을 소비하여 만들어서 손해를 줄이려는 의도였을 것 같다.   
  
2-(3) 여러가지 On Demand  
VOD : (Video On Demand) 비디오를 원할 때 볼 수 있는 서비스로 소비자는 원하는 시간 원하는 날짜에 결제하여 비디오를 볼 수 있으며, 일시정지, 빠른 재생 등의 모든 서비스를 제공한다.

AOD : (Audio On Demand) 네트워크를 통해 결제를 하여 음향파일을 다운받는 것을 의미하며, 대중은 필요한 음악을 빠르고 손쉽게 접할 수 있다.

FOD : (Fax On Demand) 알고 싶은 정보를 전화기를 이용해 요청하면서 자기의 팩스 번호를 알려 주면 곧 팩스로 정보를 받아 볼 수 있는 서비스이다.

POD : (Publish On Demand) 컴퓨터를 이용하여 고객이 원하는 대로 주문을 받아 책을 제작해 주는 서비스이다. 일반적으로 출판되지 않는 책도 개인에 맞추어 제작할 수 있어서 단종된 제품에 잘 어울린다.  
  
NOD: (News On Demand) 이용자가 컴퓨터 등을 통해 서비스회사의 데이터베이스에 접속하여 원하는 정보를 주문해 받아볼 수 있는 양방향 멀티미디어 서비스다. 독자가 관심 있어 하는 뉴스를 빠르게 접할 수 있어서 정보를 빠르게 얻어야 하는 직종에 유용하다.  
  
2-(4) 현업에서는 어떻게 쓰일지  
 클라우드 컴퓨팅에 관련되어 생각해보면, 사용자의 수가 자주 변화하는 곳에서 유용하게 쓰일 것 같다. 한 관광업체를 예를 들어 보겠다. 이 관광업체는 겨울에만 동남아나 따뜻한 지역으로 관광에 집중 되 있다고 가정하면, 이 관광업체의 자원들(서버자원, 인적자원, 물적자원)등은 여름에는 낭비가 될 것이고, 겨울에는 부족하게 될 것이다. 이러한 사업에서 우리는 수요에 맞게 이러한 자원을 줄이고, 늘리기를 할 줄 안다면 보다 효율적인 사업을 이어갈 수 있을 것이다. 따라서 이렇게 어떠한 특수성이 있는 사업에서 쓰인다면 기존의 체계보다 낭비를 줄임으로써 많은 이익을 얻을 수 있을 것이다.

**3. CRM**  
 3-(1) CRM의 개념

Customer Relationship Management의 약자로 '고객관계관리'라고 한다. 기업이 고객과 관련된 내외부 자료를 분석·통합해 고객 중심 자원을 극대화하고 이를 토대로 고객특성에 맞게 마케팅 활동을 계획·지원·평가하는 과정이다. 이 CRM의 최대 목적은 소비자들을 자신의 고객으로 만들고, 이를 장기간 유지하고자 함이다.  
  
 3-(2) 내가 생각하는 CRM의 개념  
 우리말로 흔히 단골이라는 말이 있다. 자주 들르는 손님이라는 의미이며, 이러한 손님은 가끔 들르거나 처음 들르는 손님보다 가게에 대해 잘 알고 또한 돈을 많이 써온 손님이다.   
마케팅의 관점에서 회사는 고객을 많이 만들어야 이익을 많이 볼 수 있고, 그런 상황에서 기존에 있던 고객이 사라지는 것은 뼈아픈 손해이다. 따라서 기업은 고객에 관한 많은 자료들을 이용하여 고객에게 필요한 것들을 제시하며, 자신들이 고객에게 꼭 필요한 단체라는 것을 어필하는 것이 필수적이다. 이러한 방법을 CRM이라고 하는 것 같다.  
  
 3-(3) 현업에서 어떻게 쓰일지  
 우리는 게임을 하고 종료할 때, 쇼핑을 하고 결제가 끝나거나, 공공 사이트에 들어갔다 나올 때, 종종 설문조사 같은 것을 한다. 해당 제품에 만족했는지, 해당 게임이 재밌었는지, 어떤 점이 불만이었고, 어떤 점에서 다른 사람에게 추천을 해주고 싶었는지 등등에 대해서 말이다. 이것은 기업이 고객의 필요(needs)를 받아들여 그것을 충족할만한 다음 제품을 만드는 데 유용하게 작용할 것이다.

**4. VPN**  
 4-(1) VPN의 개념  
 가상사설망 또는 VPN(Virtual private network)은 공중 네트워크를 통해 한 회사나 몇몇 단체가 내용을 바깥 사람에게 드러내지 않고 통신할 목적으로 쓰이는 사설 통신망이다.  
  
 4-(2) 내가 생각하는 VPN의 개념  
 회사에서 컴퓨터들을 연결하고 그리고 서버에 연결하는 방식에는 통신망을 이용하여 연결한다. 작은 회사라면 이러한 통신망을 유선으로 연결해도 상관이 없을 것 같다. 하지만 회사가 커지면 수많은 선들이 필요할 것이고, 이러한 선들이 많아지는 것은 결코 좋은 일은 아닐 것이다. 또한, 지점을 여러군데 흩어 놓는다면, 상황은 더 심해지는데 물리적으로 연결할 방법에는 돈이 너무 많이 들것이다. 따라서 가상으로 이러한 연결을 시킬 수 있다면 매우 효과적인 방법이었을 것이며, 이러한 방식으로 개발된 것이 VPN인 것 같다.  
 또한 이 VPN의 데이터의 이동에서 암호화가 이루어지기에 보안에 용이하다는 장점도 있다.  
  
4-(3) 현업에서는 어떻게 쓰일지  
 이 VPN의 개념을 보고 생각난 것은 군대에서의 인트라넷이다. 인트라넷은 외부에서 접근할 수 없고 특정 네트워크로 연결된 것이라고 알고 있는데, 이 VPN을 이용하여 하여 보안과 거리의 제한을 줄일 수 있었을 것으로 보인다.   
 그리고 이미 알고 있던 VPN이 있는데 IP를 우회하는 기능도 있는 것으로 안다. 가상통신망이 어떻게 IP를 우회하는지는 이해할 수 없었지만, 나는 IP우회를 통하여 할 수 없던 게임을 실행시킨 적이 있기도 한데, 이런 방식으로도 쓸 수 있다.

**5. 오픈스택, 하이퍼바이저, Xen**  
 5-(1) 오픈스택 : 일반적인 서버에 클라우드 컴퓨팅 서비스를 생성하고, 실행할 수 있게 해주는  
 오픈소스 소프트웨어이다. 또한, 오픈스택은 가상머신을 제공하여 호스트 컴  
 퓨터에서 다수의 운영체제를 동시에 실행 관리 가능하도록 지원하는 플랫폼,  
 즉 오픈소스의 하이퍼바이저를 지원, 관리하는 서비스라고도 할 수 있다.  
 하이퍼바이저 : 하이퍼바이저는 호스트 컴퓨터 1대에서 다수의 운영체제를 실행하기 위  
 한 논리적 플랫폼을 말하먀, 다양한 운영체제가 설치된 가상 서버를 여러  
 대 실행할 수 있게 도와주는 시스템, 프로그램이다.  
  
   
 이 하이퍼바이저는 크게 2가지 종류로 나뉜다.   
 첫째는, Native(Bare Metal) 하이퍼바이저 방식으로, 하드웨어 바로 위에 단에 하이퍼바이저가  
 올라가며, 하이버파이저 위에 게스트 운영체제가 올라간다.   
 장점으로는 호스트OS가 없어 리소스를 할당하지 않아도 되며, 호스트형 가상화에 비해 오버헤  
 드가 적고, 물리 컴퓨터 리소스 관리가 유연하다.  
 단점으로는 자체적 관리 기능을 갖고 있지 않아 별도의 관리 콘솔 혹은 관리 컴퓨터가 필요하  
 다는 단점이 있다.  
 두번째는, Hosted 하이퍼바이저 방식으로, 하드웨어 위에 운영체제가 있으며, 그 위에 하이퍼  
 바이저가 설치되고, 그 다음에 게스트운영체제가 올라간다.   
 장점으로는 게스트 OS의 종류에 제약이 적고, 손쉽게 도입이 가능하다는 장점이 있으며,  
 단점으로는 물리 컴퓨터의 하드웨어를 에뮬레이트하는 방식으로 오버헤드가 크다는 단점이  
 있다.  
 이때, 첫번째 방식의 하이퍼바이저 중 하나가 바로 Xen(XenServer)이라는 것이 있다.

5-(2) 기술사례

**6. 클라우드 기술 동향**  
대부분의 기업이 기술정보(IT)에 대한 자본 투자를 축소하고, 서비스 제공 속도의 가속을

목표로 삼아 일부 프로세스를 클라우드로 이전해 왔다. 그러나 클라우드는 전사적 디지털

변환을 위한 플랫폼으로서의 기능을 포함해 더 많은 역할을 담당할 수 있다.  
  
6-(1). 인공지능 부분  
 딜로이트의 설문조사에 따르면 오늘날 전체 기업 중 49%가 클라우드 기반의 서비스를 이용해 인공지능(AI)을 도입하고 있다고 한다.  
 클라우드 제공 업체들은 대규모의 선행 투자 없이 AI 시스템을 개발하고, 시험하며, 개선하고, 운용하기 쉽게 해 주는 도구와 서비스를 점점 더 많이 제공 함으로써 AI 기술에 대한 수요를 부채질하고 있다. 이런 도구와 서비스에는 머신러닝에 최적화된 하드웨어, 음성 인식 및 문장 분석을 자동화하는 API(Application Programming Interface), 생산 성 향상 자동화 머신러닝 모델링 도구, AI 개발 워크플로 플랫폼이 포함된다.  
 기술시장 리서치 기업인 트랙티카(Tractica)는 2025년까지 AI가 퍼블릭 클라우드 서비스 전체 매출의 최대 50%를 차지 할 거라고 전망한다.  
  
6-(2). 자동화된 데이터 분류  
 IDC의 최근 설문조사는 기업들이 애플리케이션과 데이터를 퍼블릭 클라우드에서 사내 설비 혹은 프라이빗 클라우드로 다시 옮기는 주요 이유가 보안에 대한 우려임을 보여 준다. 클라우드에 있는 데이터에 대해 상세한 개별 통제 권한을 확보하는 방법이 있고, 자동화된 데이터 분류는 이 작업을 더 용이하게 만들어 준다.  
  
6-(3). 산업 클라우드  
 산업 클라우드(특정 산업에 맞춤화된 클라우드 기반의 앱)는 점점 더 인기가 커지고 있다. IDC는 기업들이 이미 산업 특화된 솔루션과 전문 지식을 클라우드 서비스 공급자들에게 요구하고 있다는 사실을 발견했다. 전반적으로, IDC는 산업 클라우드 시장이 2021년까지 두 자릿수의 성장률로 빠르게 확대될거라 예상한다.   
 5개 산업군(헬스케어, 금융, 제조, 공공, 유통 및 도매)의 산업 클라우드 시장에 대한 지출 규모가 2018년 375억 달러에서 2019 년에는 약 450억 달러로 증가할 것으로 예상된다.  
  
  
6-(4). IT 운영과 소프트웨어 개발 모델  
 데브옵스(DevOps)와 같은 새로운 IT 운영 모델과 클라우드-네이티브 개발과 같은 새로운 앱 개발 패러다임의 도입 증가는 조직이 클라우드로부터 더 많은 가치를 끌어 내도록 돕고 있다. 데브옵스는 소프트웨어개발(Dev)과 IT운영(Ops) 팀간의 협업을 촉진한다. 이는 또한 코드의 시험 및 배치와 같은 프로세스를 자동화해 클라우드 도입만으로 달성 가능한 수준 이상으로 소프트웨어 제공의 효율성을 향상시킨다  
 IDC는 데브옵스 도구 시장이 5년간 15%의 연 평균 성장률로 성장해 2022년에는 80억 달러 규모 에 도달한다고 예측한다.  
 일부 기업에서는, IT 운영 모델이 더 심화된 발전을 이뤄, 클라우드 서비스 공급자가 많은 시스템 관리 작업을 자동화하고 인프라 및 보안 관리를 대신하는 ‘노옵스(NoOps)’와 서버리스 환경으로 진화 중이다. 이는 상당한 비용 절감과 기타혜택을 창출할 수 있다.**7. ERP**

7-(1) ERP의 개념  
 (Enterprise Resource Planning)기업 자원 관리라고 하며, 기업 전체의 자원을 효과적이며, 통합적으로 관리하고 경영의 효율화를 기하는 수단이다. 즉, 정보의 통합을 위해 기업의 모든 자원을 최적의 상태로 관리하자는 개념으로 업무 통합 관리라고 볼 수 있다. 좁은 의미에서는 통합적인 자료를 구축하여 회사의 자금, 회계 구매, 생산, 판매 등 모든 업무의 흐름을 효율적으로 자동 조절 해 주는 것이다.

7-(2) 내가 생각하는 ERP의 개념   
나는 공장에서 아르바이트를 해본 경험이 있다. 조금밖에 있지 않아서 그 공장의 전체 메커니즘을 이해하지는 못했지만, 한 개의 라인을 익숙하게 익혔다. 그 공장에서 내가 깨닳은 것은 부품 하나 하나가 이어지는 공정에서 하나라도 삐끗하면 모든 것이 날라간다는 것이다. 일개 작은 공장의 라인에서도 무엇인가 잘못 돌아가면 손해가 이만 저만이 아니었던 것이다.   
 회사는 훨씬 크고 복잡한 구조로 이어져있다. 그런 회사가 어느 부분에서도 끊임없이 유기적으로 맞추어 돌아간다면 회사는 최고의 성공률을 기록할 것이다. 그러기 위해서는 모든 것을 잘 파악하고 있어야 되고, 모든 업무가 유기적으로 흘러가야 한다. 그러기 위해서 정보를 얻고 그 정보를 분석해서 모든 자원을 낭비하지 않고 관리하는 방식을 ERP라고 하는 것 같다.

7-(3) 현업에서 어떻게 쓰일지

아주 작은 가게를 예로 들어보겠다. 예전에 골목식당이라는 프로그램을 본적이 있다. 그곳에 나온 백종원씨는 가게가 잘 안되는 참여자에게 어떻게 하면 효율적이고 빠르게 음식을 내놓고, 어떻게 빨리 서빙하며, 최고의 매출을 달성할 수 있는지 생각해보라고 따끔하게 지적했었다.  
 이후에 그 참여자는 간단한 재료는 미리 손질을 하여 준비하고, 테이블의 위치를 바꿔 최소한의 동선으로 주문을 받을 수 있게 하였으며, 셀프바는 누구나 알기 쉽게 바꿨다.   
 효과는 매우 좋았다. 기본 20분 이상이 걸리던 음식들이 10분정도로 줄었으며, 주문이 꼬이는 일도 없어졌다. 손님들의 만족도는 오르고, 이익은 극대화된 것이다.  
 이러한 방식으로 기존의 자원을 어떻게 효율적으로 썼느냐에 따라서 엄청난 효과를 볼 수도 있다는 것이다.   
 회사를 대입해 본다면, 자주 접촉하는 부서끼리 가까이 붙이거나, 회계를 더 효율적이고 빠르게 처리, 버려지는 자원에 대한 창의적 생각, 소모되는 비용의 절감에 신경씀으로써 많은 이익을 벌어들일 수 있을 것이다.

**8. API**  
 8-(1) API의 개념  
 응용 프로그램 프로그래밍 인터페이스)는 응용 프로그램에서 사용할 수 있도록, 운영체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스를 뜻한다.  
  
 8-(2) 내가 생각하는 API의 개념  
 개념적으로는 어려운 설명이다. 결국엔 인터페이스라는 것인데, 인터페이스라는 것은 어떤 것과 어떤 것 사이의 연결고리를 의미한다. UI는 유저인터페이스로 유저와 기계사이의 연결을 쉽게 만들어주는 매개체가 된다. 이와 같은 의미로 API는 어플리케이션 사이의 매개체를 의미한다.   
쉽게 예시를 들자면, 우리가 위치 기반의 어플리케이션을 만든다고 했을 때, 우리는 GPS와 같은 기능까지 처음부터 짜기 시작한다면 엄청난 시간과 노력이 들것이다. 하지만 이미 있는 기능인 GPS를 어떠한 곳에서 가져와서 버튼하나로 컨트롤 할 수 있다면 이보다 쉬운 것은 없을 것이다.  
이처럼 어플래케이션간의 매개체를 API라고 한다.  
  
 8-(3) 현업에서 어떻게 쓰일지

이번에 코로나 사태에 관련된 뉴스 중에서 소프트웨어의 활약상이 몇몇 나왔다. 중학생들끼리 위치기반, 정부 데이터를 가지고 코로나 위험지역을 지도에 표시해서 알려주는 어플을 만든 것이다. 이렇게 빨리 그런 어플을 개발할 수 있던 것은 기존에 있던 지도의 API와 정부의 공공데이터API등을 이용하여 기존에 있던 기능을 자신의 어플리케이션에 빠르게 적용할 수 있었기 때문일 것이다.

**9. DR 인프라측면에서 Degest Recovery 재해복구**  
 9-(1) DR의 개념  
 IT 관점에서 말하는 재해복구(Disaster Recovery, DR)란, 각종 재해 및 위험요소에 의해 정보시스템이 중단됐을 때 이를 정상으로 회복시키는 것을 의미한다. 인프라 및 서비스를 최종적으로 복구하는 것까지를 넓은 의미의 DR로 이해할 수도 있겠지만, 실제로 기업 IT서비스를 운영하는 입장에서는 이미 일어난 재해를 뒤로 하고 빠르게 서비스를 재개하는 것이 최우선 과제다. 따라서 IT 운영 측면에서의 DR은 서비스를 최대한 빠른 시간 내에 다른 시스템으로 대체해 운영할 수 있도록 평소에 대비하는 것을 말한다.  
  
 9-(2) 내가 생각하는 DR의 개념  
 재해라고 부풀려 말하지만, 쉽게 말해서 정전으로 서버가 다운되거나, 어떠한 불가피한 상황으로 데이터가 날라갈 수 있는 상황도 있을 것이다. 검색을 잠시 해봤는데 흔히 9.11테러를 지적한다. 그 당시에 350여 기업의 전산시스템도 건물붕괴와 함께 파괴되었다. 그 당시 데이터복구를 할 수 있던 150여개의 기업을 제외한 나머지 기업은 도산했다고 전해진다. 이렇게 기업은 생각치도 못한 방법으로 기업의 전산시스템, 데이터, 기밀 등을 날려버릴 수 있고, 그에 대한 결과는 기업의 몰락으로 이어 질 수도 있기에 보험을 들어놓듯 DR, 즉 복구를 할 수 있게 대비를 해놓아야 한다.  
  
  
 9-(3) 현업에서 어떻게 쓰일지(예전에 배운 기억으로)

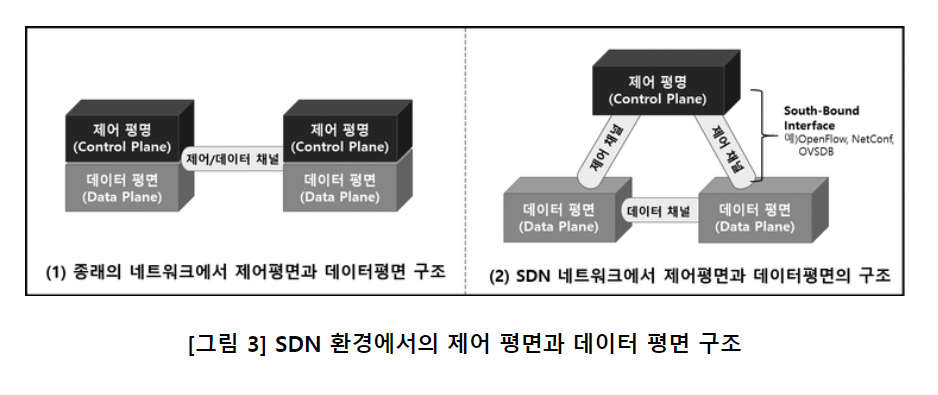
가장 쉬운 방법으로는 복사본을 백업 시켜놓는 방식이 있다. 간단하지만, 역시 복사본도 날라갈 수 있고, 영구히 보관하기 위해서는 엄청난 양의 데이터를 계속 쌓아놓아야 한다는 물리적 한계가 존재할 것이다.  
 다른 방법으로는 같은 전산시스템을 2개를 만드는 것이다. 하나가 작동이 불가능해도 다른 하나가 제 역할을 할 수 있게, 이러한 방식은 위의 방법보다 안전하겠지만 데이터가 중복되어 저장되는 일은 꽤나 안 좋은 것이라고 알고 있다.  
 알고 있는 마지막 방법은 클라우드에 백업을 시켜놓는 것이다. 이 방법은 데이터가 분산저장 되어 꽤나 D(Degest)로부터 안전하다고 들었다.

**10. on-premise** 10-(1) on-premise의 개념  
 클라우드와 반대 되는 개념이며, 클라우드가 새로운 방식이라면 on-premise는 기존의 것, 즉 기존의 물리형태의 서버를 의미한다.   
  
 10-(2) 내가 생각하는 on-premise의 개념  
 물리적 형태의 서버라 함은 흔히 떠올리는 커다란 전산실의 컴퓨터이다. 클라우드와의 차이점을 들자면, 서버를 사고, 구축하는데 시간과 돈이 꽤 많이 소모되며, 관리업무가 많고, 자원의 낭비가 있을 수 있다. 이러한 물리적인 형태의 서버가 이제는 가상화를 통하여 클라우드에서 처리가 가능해졌다.   
  
 10-(3) 현업에서 어떻게 쓰일지

기존의 사용되고 있는 서버들은 계속 사용되겠지만, 새로 시작하는 기업, 회사에서는 클라우드 컴퓨팅을 이용할 것 같다. 이유는 초기에 서버가 얼마나 이용될지 알기가 힘들며, 사용한 만큼 돈만 지불하면 되고, 초기 구축기간이 매우 빠르기에 사업에 용이하기 때문이다.   
 on-premise에서 현재의 클라우드 컴퓨팅으로 바뀌는 격변기에는 많은 사람들이 새로운 기술에 대한 장점을 알았어도 그것에 대한 불안감이 컷기에 잘 바꾸지 않았겠지만, 이제는 많은 서버들이 사라지고, 클라우드 형태로 넘어가기에 점점 더 많은 기업들이 on-premise에서 클라우드로 갈갈 탈 것 같다.

**11. EOS, EOL**  
 11-(1) EOS, EOL의 개념  
 EOS : (End of Sale) 제조에서 제품에 대한 생산 및 판매 중단을 중단하기 시작하는 날짜입니다. 제 3자 공급업체 등에서 다른 방법을 경유하여 구입이 가능할지라도 실질적으로 공식 제조사에서 제품을 만드는 것은 끝임을 표시하는 날짜다.   
 EOL : (End of Life) 제조에서 제품에 대한 수명이 끝이 났다고 판단하여 이 날짜 이후로 제품에 대한 마케팅, 유지, 판매 등에 대한 지원을 일절 중단함을 의미한다.  
  
 11-(2) 내가 생각하는 EOS, EOL의 개념  
 흔히 휴대폰을 생각해보면, 우리는 새로운 버전의 폰이 나오고, 기존의 것은 생산이 적어지다가 결국에 단종이 된다. 이런 경우를 EOS라고 생각한다. 그래도 이 제품이 생산이 되지 않지만, 이미 생산된 제품도 있고, 수리도 받을 수 있다. 하지만 몇 년이 더 지나보면, 더 이상 이 제품은 보이지도 않고 수리점에서 조차 부품이 없어 수리를 할 수가 없다. 이런 경우를 EOL이라고 생각한다.  
  
 11-(3) 현업에서 어떻게 쓰일지

최근에 도쿄올림픽에 관한 뉴스를 많이 보았다. 그 뉴스에는 여러가지 문제점들에 관해서 흘러나왔다. 그 중에 마이크로소프트웨어 사의 윈도우 7에 관한 이야기가 나왔다.   
 지금의 버전은 윈도우 10이지만, 일본에는 예전의 7이 아직도 대량이 남아있다는 것이다. 하지만 마이크로소프트웨어는 이 윈도7의 서비스 종료를 선언했다. 즉 EOL을 선언한 것이다. 이로 인해, 7버전은 보안에 취약해졌고, 헤커들은 이러한 컴퓨터에 쉽게 침투를 할 수 있다는 뉴스였다.   
이렇듯 모든 제품, 심지어 소프트웨어에서 조차도 제품의 유통기한이 있다는 것을 말한다.

**12. SDN**  
 12-(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.  
  
 12-(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.   
  
 12-(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.

**13. Bare Metal**  
 13-(1) Bare Metal의 개념  
 베어메탈은 어떤 소프트웨어도 설치되어 있지 않은 상태의 하드웨어를 의미한다.

13-(2) 내가 생각하는 Bare Metal의 개념

말 그대로 컴퓨터가 있을 때 운영체제를 포함한 그 어떤 소프트웨어가 깔려있지 않은 순수한 형태의 하드웨어인 것 같으며, 최근에 운영체제없이 하이퍼바이저를 설치하는 native타입의 하이퍼바이저에 쓰임에 딱 맞는 것 같다.   
  
 13-(3) 현업에서 어떻게 쓰일지

하이퍼바이저라는 것이 나오기 전의 베어메탈은 자기가 원하는 방식으로 컴퓨터를 쓰기 위해 적절했을 것 같다. 예를 들면, 어떤 운영체제를 쓸지, 어떤 기본 프로그렘을 설치할지, 필요한 것은 직접 설치하고, 불필요한 것은 아예 설치도 하지 않는 방식으로 자신만의 컴퓨터를 만들 때 사용됬을 것이다.   
 하이퍼바이저가 나오면서는 가장 먼저 이 베어메탈 바로 위에 하이퍼바이저가 올라가며, 그 위로 운영체제가 올라가게 되었을 것이다. 이러한 방식은 한 개의 컴퓨터가 여러 개의 처리를 할 수 있게 시켜줬을 것이다.   
 한 컴퓨터에서 여러 분야의 업무에 대한 처리를 동시에 하는데 유용할 것이다.