2️⃣

02. 클라우드 컴퓨팅 소개

: 기존 VS, HS에 비해 적은 비용으로, 동적으로, 필요에 따라 신속하게 시스템 성능을 높였다가 낮출 수 있을까?

“클라우드 컴퓨팅” (POSE)

- **pay-as-you-go**
  - 사용한 만큼만 비용 지불
  - 초기에 막대한 지출이 없음 (장비 구매 & 관리 시설 구축)
- **on-demand resource provisioning**
  - 필요에 따라 자원 할당 받음
- **scalacility (확장성)**
  - 수요에 따라 성능/용량을 늘리거나 줄일 수 있음
- **elasticity (탄력성)**
  - 실시간(신속하게), 자동화된 방식으로 성능/용량을 조절할 수 있음

CC 란?

- 탄생
  - 유틸리티 컴퓨팅에서 유래. 유틸리티 모델로 판매 가능
  - 유틸리티 : 전기, 가스 등 자원을 일정 비용을 내고 사용, 사용한 만큼 비용 지불
  - 2000년대 → **네트워크의 발전, 가상화 기술의 고도화** 등에 힘입어 유틸리티 컴퓨팅의 개념 구현
- 정의
  - 가트너, 포레스트 리서치, 미국국립표준기술연구소 (NIST), 마이크로소프트, 구글, 아마존

- **CC 키워드**
  - **온디맨드 서비스 (on-demand service)**
  - **탄력성(elasticity), 확장성(scalability)**
  - **인터넷 (광대역 네트워크)**
  - **⭐IT 자원의 풀링(Pooling) & 재사용 → (Multitenancy)**
  - **퍼블릭 클라우드 (AWS, Google Cloud, Azure) vs 온-프레미스**
  - **⭐가상화**
  - **데이터 센터 (IT 자원의 집적)**
- **클라우드 컴퓨팅이란**
  - **인터넷을 통해, 가상화된 컴퓨터의 시스템 리소스 (IT 리소스)를 요구하는 즉시 제공 (on-demand delivery)받고 사용하는 기술**
    - IT 리소스 = 컴퓨팅 리소스/자원(컴퓨터 네트워크, 데이터베이스, 서버, 스토리지, 어플리케이션, 서비스 등)
  - 인터넷 기반 컴퓨팅 기술로, **정보/IT자원을 자신의 컴퓨터가 아닌 클라우드(인터넷)에 연결된 다른 컴퓨터로 처리/관리**하는 기술
  - 주문형 접근 가능
  - 사용한 만큼 지불(**pay-as-you-go**), HW 소유하지 않음 → **초기 구축 구매비용 & 지속적인 관리비용 지출 불필요**
  - **최소한의 관리/노력**으로 즉각적 시스템 확장/축소 가능
  - **신속한 서비스 개발 및 배포** 가능
  - 다수의 컴퓨팅 리소스를 가상화 기술로 통합하고, 원격으로 사용자에게 리소스를 제공하는 기술

CC의 특성 (5가지)

1. **온디맨드 셀프 서비스**
   - CC 사용자는 클라우드 **서비스 제공자의 직접적 개입 없이, 언제 어디서든 원하는 만큼**의 컴퓨팅 자원, 네트워크, 스토리지와 같은 **자원을 자동으로 프로비저닝** 할 수 있다
2. **광대역 네트워크 접근**

- 인터넷/네트워크를 통해 다양한 이기중 클라이언트 단말 플랫폼을 통해 클라우드 자원에 접근이 가능함

3. **리소스 풀링**
   - **CC 서비스 제공자의 컴퓨팅 자원은 풀링(Pooling)** 되어서 서로 다른 사용자가 컴퓨팅 자원을 나누어서 사용하는 **멀티테넌트 모델(Multi-tenent model** : 하나의 자원을 다수의 사용자가 상호 간섭/인지 없이 공유하여 사용하는 것)로 **다양한 사용자에게 제공**
   - 풀링된 자원은 사용자의 요구에 맞게 제공/회수되며, 회수된 자원은 다른 사용자에게 재할당 가능
4. **신속한 탄력성 (rapid elasticity)**
   - 클라우드 자원을 신속하게 프로비저닝하여 배포할 수 있고, 사용자의 요구에 따라 자원 규모를 확장/축소 가능
5. **사용량 측정 (measured service)**
   - CC 컴퓨팅 자원 사용량 측정이 가능해야 함
     - 과금 모델 구현 위해 필수 (pay-as-you-go)
   - CC 자원에 대한 모니터링, 제어, 사용량 기록 등이 클라우드 서비스 제공자 & 사용자 모두에게 투명하게 제공

사업적 동인 (클라우드 활용 이유)

- **인프라 규모 산정의 용이성**

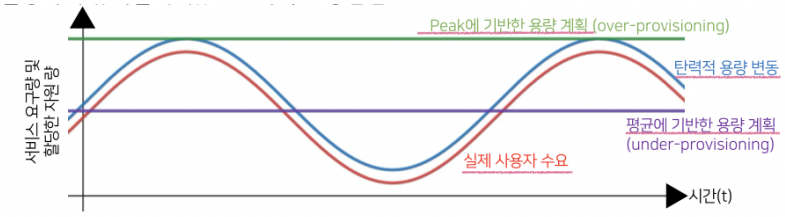
  서비스를 **비용 효율적으로 운영**하며 **서비스 품질을 일정하게 유지하는 것은 어려움**

  - **(용량 > 수요)** IT 자원을 **Peak Demand**에 맞추어 구축하면, Peak Hour 이외의 시간에는 **자원이 낭비 (over-provisioning)**
  - **(용량 < 수요)** IT 자원을 **평균 수요**에 맞추어 구축하면, Peak Hour때 서비스 **품질 저하가 발생 (under-provisioning)**

  인프라 규모 산정을 위해서 용량 계획을 수행하는데, **CC를 활용하면 효과적인 용량 계획을 수립 가능**

- **용량 계획 (capacity palanning)**
  - IT 자원, 서비스의 미래 수요를 예측하고 이를 충족시키는 용량을 확보

        - 용량 : IT 자원이 주어진 시간 내 달성할 수 있는 일의 최대치
    - **기업의 용량 계획 : 용량과 수요의 차이 최소화 전략 필요**
        1. **리드 전략 (lead strategy)** : 수요를 **미리 예상**, IT 자원을 추가해 용량을 늘림
        2. **지연 전략 (lag strategy)** : 용량이 **최대치**에 다다르면 IT **자원을 추가**해 용량을 늘림
        3. **일치 전략 (match strategy)** : 수요가 증가하는 만큼 **조금씩 IT 자원을 추가**해 용량 늘림
- **비용 절감**
    - 서비스 용량 → 최대 사용 요구량에 맞추어 설계. IT 인프라 구축에 많은 비용 설계
    - 인프라 유지/관리에 많은 비용 소요
    - **클라우드 컴퓨팅 사용**
        - **가상화 기반 자원 풀링 → 적은 자원, 다수의 사용자 서비스 가능**
        - **동적으로 자원 확장/축소 가능, 사용한 만큼만 비용 지불**
        - **물리적 IT 자원관리 필요 X, 운영 인력 및 유지보수 비용 절감**
- **탄력성 & 조직의 민첩성**
    - 수요 변화에 따라 **신속히 시스템 용량 변동** → 서비스 품질 유지하며 IT 서비스 사용 비용 최소화



- **가용성 (Availability)**
    - 가용성, 서비스 중단 → **비즈니스 수익성에 영향**
    - **동적 확장** 중요
    - 가용성 → **신뢰성 (Reliability)과도 직결**
    - **인프라 활용 가능한 시간 기준**으로 측정

        - 시스템 유지보수, 장애로 인한 **shutdown 시간 (= downtime)을 고려**, 인프라가 **활용 가능한 시간의 비율**을 측정해 가용성 지표로 활용
    - 클라우드 활용 시
        - 시스템 장애 최소화
        - 장애 발생 시, 이중화 or 서비스 이관 (migration) 기법 이용하여 서비스 중단시간 최소화
        - 서비스 장애 일정 수준 이상 발생 시, 클라우드 제공자가 적절한 보상해줌 (SLA : 서비스 수준 협약)
- **투명한 사용량 측정**
    - 사용한 만큼 비용 지불이 가능
    - 자원 사용량 정확하게 모니터링 가능
- **복원성**
    - **QoS & SLA (서비스 품질) 보장을 위해 복원성 (resiliency)를 보장할 수 있는 다양한 기법** 구현
- 클라우드 적용 사례
    - 넷플릭스 2015.03 AWS
    - 포켓몬 고 구글 클라우드 플랫폼

기술적 혁신 (CC에 영향을 준 기술)

- **클러스터링 : 클라우드 컴퓨팅은 대규모의 클러스터링된 IT 자원을 효율적으로 사용하는 기술을 기반으로 함.**
    - 여러 자원을 묶어 하나의 거대한 자원으로 활용되는 기술
    - 클러스터 : 서로 연결되어 단일 시스템으로 작동하는 독립적인 IT 자원의 그룹 → **전용 고속 통신 링크를 통해 동기화 필요**
    - **이중화 (HA, high availability)와 장애 조치 (failover)** 기능을 클러스터에 내장 가능 → **가용성, 신뢰성 증가, 시스템 장애율 낮아짐**
        - 풀링 방식으로 자원 사용가능

- **그리드 컴퓨팅**
  - 원거리 통신망으로 연결된 **서로 다른 기종의 컴퓨터들**을 **광대역 네트워크**를 이용하여 **하나의 단일 시스템으로 구성하여 컴퓨팅 자원을 공유하는 시스템**
  - 가상의 대용량 고성능 컴퓨터 구성, 고도의 연산 작업 혹은 대용량 처리 수행 목적으로 사용
- **가상화**
  - **IT 자원의 가상 인스턴스를 만드는 기술**
  - 제한된 물리적 자원을 많은 수의 **가상 자원으로 가상화** → 독**립적인 자원 제공**

기본 개념, 용어 정리

- **클라우드**
  - 확장 가능하고 측정 가능한 IT 자원을 원격으로 제공하기 위한 IT 환경
- **IT 자원**
  - 물리적 IT 자원, 가상의 IT 자원 통칭
  - 가상 서버, SW 프로그램과 같은 sw 기반 자원 + 물리 서버, 네트워크 장비와 같은 hw 기반의 자원 포함
- **On-premise** (온 프레미스 : 직접 설치/운영) ↔ 클라우드
  - IT 자원을 클라우드와 같은 원격 환경이 아닌, 자체적으로 보유한 전산실 서버에 직접 설치해 운영/제공하는 방식
- **클라우드 서비스 제공자 vs 클라우드 서비스 소비자**
  - **제공자** : **CSP (Cloud Service Provider)**
    - 클라우드 기반 IT자원을 제공, 서비스하는 기업
  - **소비자**
    - 클라우드 서비스 사용하는 사람/기업
  - **MSP (Managed Service Provider)**
    - 클라우드 **관리** 서비스를 제공하는 기업

        - MSP는 CSP와 고객을 연결하는 역할.
            - 마이그레이션, 컨설팅, 운영 관리
- (성능) 확장, Scaling
    - IT 자원을 늘이거나 줄이는 것
    - HS : 수평적 확장
        - 같은/유사한 형태의 자원을 추가 할당하거나 제거하는 것
    - VS : 수직적 확장
        - IT 자원을 더 높거나 낮은 사양의 다른 자원으로 교체하는 것

수평적 확장 (HS)	수직적 확장 (VS)
저비용	고비용
확장 중에도 지속적인 서비스 가능	확장(=교체) 중에는 서비스 중단
자원 복제, 자동 확장 가능	불가능
확장하는 경우, 추가적인 HW 필요	기존 자원을 교체하는 방식
확장에 제약 없음	최대 용량, 기술 한계 등에 제약을 받음

CC 장점

- 비용 절감 (초기비용, pay-as-you-go)
- 확장성 증대 (수요 변화에 따라 신속한 반응)
- 가용성, 신뢰성 증대 (HA, failover 기능)

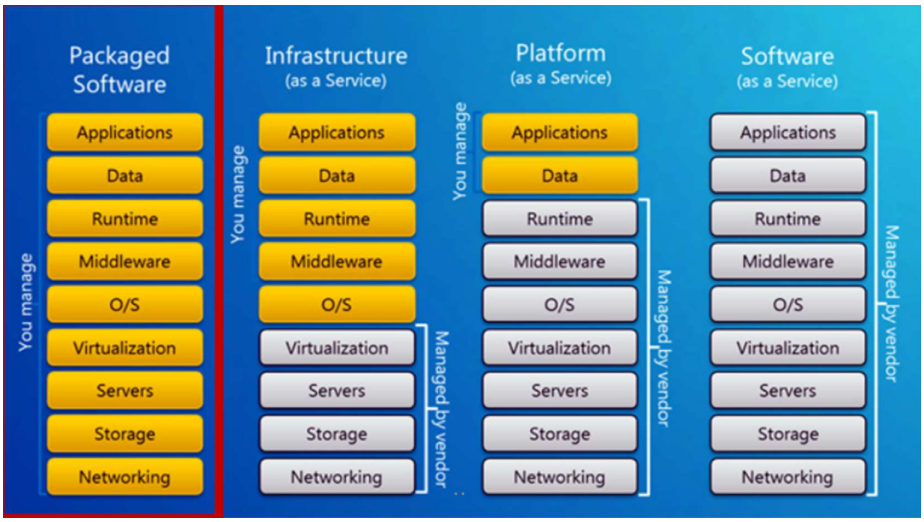
CC 단점

- **보안 취약성 증가**
    - On-premise는 외부 접속 차단은 가능
    - 반면, CC는
        - 제공자가 데이터 접근 간으

        - 다른 사용자와 동일한 IT 인프라 공유 → 피해 시 다른 사용자에게 확산
        - 네트워크를 통해 중요 데이터 송수신 → 네트워크 위협에 대비해야 함
- 클라우드 제공자 사이 제한된 이식성(Portability), 상호 운용성
    - 다른 클라우드로 이동 어려움
    - 클라우드에 의존하지 않는 범용 App 개발 어려움
- 법적 이슈
    - 데이터 센터의 물리적 위치에 따른 법적 이슈
- 운영, 관리 제어 권한 축소
    - 사용자가 낮은 수준의 IT 제어권 할당
    - 자원 제어, 최적화에 한계
- 네트워크 의존성 증가
    - 네트워크 연결 반드시 필요
    - 인터넷 장애, 혼잡 → 클라우드 컴퓨팅 서비스 품질 저하
- 사건/사고 발생 시 피해 규모 큼
    - 대응 방안
        - **제공자**
            - 인프라 설비 확충, 고도화, 장애 대응 시스템(이중화) 구축
            - 네트워크 서비스 제공자 다변화
        - **사용자**
            - 실수 방지 위한 전문인력 채용, 직원 교육 강화
            - 멀티 클라우드 활용을 통한 클라우드 장애 대비

CC 모델

- (참고) 클라우드를 사용하지 않는 전통적인 서비스 모델 (기업이 직접 데이터센터를 보유/관리/운영, on premise)
  - 물리적인 기기, 장비, 하드웨어(CPU, RAM, Storage, Network device 등)를 모두 직접 구매하고 설치/관리 해야함
  - OS 및 필요 SW를 직접 설치, 네트워크 환경을 직접 구성, 서버 운영 및 관리를 직접 수행
  - 서비스 운영을 위한 모든 과정을 사용자가 직접 수행해야 하므로, 시간/비용이 많이 소요됨
  - 단, 사용자가 모든 것을 제어하기 때문에 최적의 customizing을 할 수 있음



- **IaaS (Infrastructure as a Service)**
  - 물리적인 IT 자원을 가상 자원의 형태로 제공해 주는 서비스
  - 서비스 사용자는 클라우드에서 기본적인 HW 인프라(CPU/MEM/HDD, 네트워크 등)만 제공받음
  - 물리적인 IT 인프라를 구축하지 않고도 IT 자원을 사용할 수 있는 장점
  - 단, 물리 인프라 이외에는 사용자가 직접 설치/관리함(OS 설치, SW 제품 개발 및 설치, 서비스 운영 등)
    - Ex) Amazon EC2, MS Azure, Openstack
- **PaaS (Platform as a Service)**
  - IaaS 서비스 위에, '플랫폼'을 만들어서 서비스를 제공하는 것으로, 미들웨어/라이브러리 등이 설치된 개발 환경을 '플랫폼'으로 생각할 수 있다
  - HW 인프라 뿐 아니라 개발 플랫폼 (OS, 미들웨어, 프로그래밍 언어, 라이브러리, SW개발도구 등)을 클라우드로부터 제공받음
  - 서비스 이용자는 개발환경이 구축된 환경을 제공 받으므로 모두 동일한 개발환경을 손쉽게 확보할 수 있고, 즉시 SW개발에 집중할 수 있고, SW를 개발 후 설치하기만 하면 서비스를 운영할 수 있음
  - 단, HW 인프라에 대한 제어가 어렵고, 제공되는 플랫폼과 개발 제품간의 호환이 되지 않을 경우 문제 발생

- Ex) Heroku, Google App Engine, IBM Bluemiz, OpenShift, SalesForce

- **SaaS (Sofrware as a Service)**

  - (인프라와 플랫폼을 포함하여) SW 서비스를 클라우드에서 제공받는 형태
  - 서비스를 직접 개발/설치하지 않고, 클라우드에서 제공하는 서비스를 사용
  - SW/서비스 개발 및 설치가 필요 없고, 서비스 관리도 필요 없음
  - 단, customize가 어려움

  - Ex) G메일, 구글 클라우드, 네이버 클라우드. MS 오피스365, 드롭박스 등

- CC 모델에 따른 보안 강화 방안

구분		IaaS	PaaS	SaaS	FaaS	CaaS	BaaS
기능 설명		서버, 스토리지, 네트워크 등과 같은 IT 인프라를 제공	애플리케이션 개발 및 배포에 필요한 플랫폼을 제공	웹 브라우저 또는 모바일 앱을 통해 접근할 수 있는 소프트웨어 애플리케이션을 제공	서버리스 아키텍처에서 실행되는 코드 조각(함수)을 제공	컨테이너 제작 및 배포에 필요한 관리를 제공	모바일 애플리케이션 개발에 필요한 백엔드 인프라를 제공
구현 기술		가상화 및 스토리지	데이터베이스 관리	웹 애플리케이션	서버리스 컴퓨팅 프레임워크	컨테이너 오케스트레이션	클라우드 데이터베이스
보안 관점	위협 요소	• 가상머신 이미지 보안 취약점 • 네트워크 구성 및 액세스 제어	• 애플리케이션 코드 보안 취약점 • 인증 및 접근제어	• 데이터 유출 • 취약한 계정 및 암호 노출	• 코드 인젝션 • 인증 및 접근제어	• 컨테이너 이미지 취약점	• 데이터 노출 및 유출 • 인증 및 접근제어
	대응 방안	• 가상 머신 및 네트워크 보안 • 서버 하드닝, 스토리지 보안	• 애플리케이션 보안 • 런타임 환경 보안 • 데이터베이스 보안	• 웹 애플리케이션 보안 • 인증 및 권한 관리 • 소프트웨어 업데이트 보안	• 함수 코드 보안 • 실행 환경 보안 • 이벤트 및 메시지 보안	• 클라이언트 애플리케이션 보안 • API 및 데이터 저장소 보안	• 컨테이너 이미지 보안 • 호스트 및 네트워크 보안 • 서비스 및 애플리케이션 보안
		• 데이터 암호화 / 보안 인증 및 접근 제어 / 감사 추적 / HSM(Hardware Security Module)					

- 클라우드 배포 모델 (Deplyment model)

  클라우드 환경의 형태, 소유권과 규모, 접근 방법 등에 의해 4가지로 구분됨.

  1. **Public Cloud**
     - 제 3자인 클라우드 서비스 제공자가 소유하고 공개적 접근 가능한 클라우드 서비스
     - 클라우드 인프라가 일반 사용자에게 오픈되는 형태
  2. **Private Cloud**

- 하나의 조직만을 위해, 특정 기관의 사용자에게만 오픈되어 운영되는 클라우드

3. **Community Cloud**
    - 특정 커뮤니티에 속한 소비자에게만 접근 허용 → 다수의 기관에 속한 사용자에게 서비스
    - 1,2의 중간 형태
4. **Hybrid Cloud**
    - 위 3가지 모델 혼용하여 사용하는 클라우드 서비스