클라우드 왜 사용할까?

- 비용 절감 : 필요한 만큼만 리소스 할당 & 그만큼만 비용 지출
- 확장성 : 트래픽 변화에 따라 자원 확대/축소 가능.
- 가용성 : 여러 데이터센터 사용 -> 안정적 서비스 제공
- 접근성 : 인터넷만 있으면 언제 어디서든 접근 가능

클라우드 컴퓨팅, 클라우드 네이티브

클라우드(컴퓨팅) :

온디멘드 방식으로 인프라 + 소프트웨어를 인터넷 통해 사용자에게 제공.

클라우드 네이티브 :

클라우드의 이점을 최대로 활용할 수 있는 기술, 아키텍쳐, 방법론
Ex) 도커(컨테이너), 쿠버네티스, CI/CD, MSA 등등.

클라우드 서비스 모델 (IaaS, PaaS, SaaS)

나뉘는 기준 : 어느 범위까지 클라우드에서 제공하는가.

클라우드 서비스 모델 (IaaS, PaaS, SaaS)

- IaaS (Infrastructure as a Service) :

가장 기본적 인프라만 제공. (네트워크, 스토리지, 서버, 가상화 등)
관련 서비스 : aws EC2, Azure VM

- PaaS(Platform as a Service) :

IaaS 제공 포함 + 운영체제, 미들웨어, 런타임 제공.
관련 서비스 : aws Elastic Beanstalk, GCP App Engine

*미들웨어 : 서로 다른 매체가 통신할 수 있도록 돕는 소프트웨어. 운영체제-애플리케이션 사이에 존재.
Ex) Web Server, WAS

클라우드 서비스 모델 (IaaS, PaaS, SaaS)

- SaaS (Software as a Service) :

인프라, 운영체제 뿐만 아닌 애플리케이션 형태의 완전한 소프트웨어 제공.

특징 : 유지/관리에 가장 효율적이나, 직접 제어할 수 있는 부분이 적음.

관련 서비스 : aws Marketplace, Google Workspace

스케일 업, 스케일 아웃

클라이언트의 요청을 안정적으로 처리하기 위함.

- 스케일 업 : 하나의 컴퓨터의 성능을 향상
- 스케일 아웃 : 컴퓨터 개수 자체를 늘려 부하 분산.

클라우드 환경 -> 스케일 아웃 선호

Why

1.탄력성 : 필요에 맞게 리소스 확장/축소

2.비용 효율성 : 사용량 맞춰 리소스 조절 -> 불필요한 비용 절감

3.장애 허용 : 한 서버 죽어도 다른 서버 이용.

고가용성

- High Availability :

서버, 네트워크, 스토리지, 애플리케이션을 오랜 기간동안 안정적으로 운영할 수 있는 성질.

- 달성 방법 :

1.이중화 : 하나의 데이터를 여러개로 복사하여 저장

2.다중 리전, 다중 AZ : 여러 지역에 배포, 하나의 리전 내에서 여러 AZ에 서버 분산 배치

3.로드밸런싱, 오토스케일링

로드밸린성, 오토스케일링

- 로드밸런싱 :

여러 서버에 트래픽을 분산.
균등하게 분배해 성능, 가용성 확보.
스케일 아웃 아키텍처의 핵심 요소.

작동 과정,원리 :
1.요청 -> 로드밸런싱 도착
2.분산 알고리즘 거쳐 로드밸런싱 -> 여러 서버로 분배
3.서버 응답 -> 클라이언트로 전달

로드밸런싱, 오토스케일링

- 오토스케일링 :

리소스 사용량에 맞춰 VM 자동 확대, 축소

갑작스레 트래픽 증가하면 인스턴스 생성, 줄어들면 제거.
-> 유연한 대처로 비용 절감.

오토스케일링 사용 기준 :

- CPU 사용률
- Cloud 로드밸런싱 사용량

서버리스

사용자는 IT 인프라를 신경쓰지 않고, 오직 함수단위의 로직만 생성하면 된다.

-> 이러한 특성으로 인해 FaaS(Function as a Service),
또는 백엔드 서비스를 추상화했다는 의미의 BaaS(Backend as a Service) 라고 불림.

- 단점 :
1.콜드스타트 : 요청 없으면 실행 안하다가, 요청 오면 실행환경 뛰움 -> 첫 요청시 지연 발생.
2.장시간 실행 불가 : 오래 걸리는 작업 불가능. (동영상 인코딩, 대용량 데이터 분석 등)
3.인프라 제어 어려움 : 인프라 직접 커스텀 불가능.