

클라우드 환경의 이해

개발자가 되기 전에 알았으면 좋았을 것들



클라우드 컴퓨팅

- 클라우드 컴퓨팅의 배경

- 하드웨어 성능의 비약적인 성장
 - CPU, 스토리지, 네트워크 장치 등 컴퓨터를 이루는 주요 장치들의 성능이 과거에 비하여 크게 성장을 이루어 네트워크를 통해 실시간으로 자원을 제공하는 것이 가능해졌다.
- 네트워크 기술의 발전
 - 네트워크 기술이 발전함에 따라, 언제 어디서나 원격으로 자원에 접근할 수 있게 되었다.
- 가상화 기술의 발전
 - 가상화 기술이 발전하면서 여러 사용자가 물리적 자원(하드웨어)을 공유하여 사용하고 필요한만큼 자원을 할당받을 수 있게 되었다.

클라우드 컴퓨팅

• 클라우드 컴퓨팅의 발전과정



클라우드 컴퓨팅의 발전 단계

클라우드 컴퓨팅

- 클라우드 컴퓨팅의 유형

- 퍼블릭 클라우드 (Public Cloud)

- CSP(Cloud Service Provider)가 제공하는 일반적인 형태의 클라우드 환경

- 프라이빗 클라우드 (Private Cloud)

- 특정 조직을 위한 전용 클라우드 환경으로, 보안성을 강화할 수 있다.
 - 온프레미스로 구성을 할 수도 있고, CSP를 통해 호스팅을 맡길 수 있다.
 - 클라우드 환경 자체를 관리할 인력이 필요하다.

- 하이브리드 클라우드 (Hybrid Cloud)

- 퍼블릭 클라우드와 프라이빗 클라우드를 결합하여 사용하는 형태로, 각각의 이점을 모두 활용할 수 있다.
 - 구현과 운영의 난이도가 높고, 자칫 잘못하면 각 클라우드의 관리 체계가 나뉘어져 하이브리드로서의 장점을 잃게 될 수 있다.

- 멀티 클라우드 (Multi Cloud)

- 여러 CSP의 서비스를 혼합하여 운영하는 형태이다.
 - 서비스별로 유리한 업체를 선정하여 비용을 최적화할 수 있고, 클라우드 환경 자체에서 발생할 수 있는 위험을 분산 할 수 있다.
 - 다양한 클라우드 환경으로 이루어져있어 관리의 어려움이 있다.

클라우드 컴퓨팅

- 클라우드 컴퓨팅의 모델

- IaaS (Infrastructure as a Service, IaaS)

- 가상화된 컴퓨팅 자원(컴퓨팅, 스토리지, 네트워크 등)을 제공하여 사용자가 인프라를 직접 구성하고 관리할 수 있도록 제공한다.
 - ex) AWS EC2, Azure VM, GCP GCE

- PaaS (Platform as a Service, PaaS)

- 애플리케이션을 개발, 운영하는데 필요한 플랫폼과 환경을 제공한다. 직접 인프라를 구성하거나 관리할 필요가 없어 개발에 집중할 수 있다.
 - ex) AWS Elastic Beanstalk, Azure App Services, GCP App Engine, Vercel, Heroku

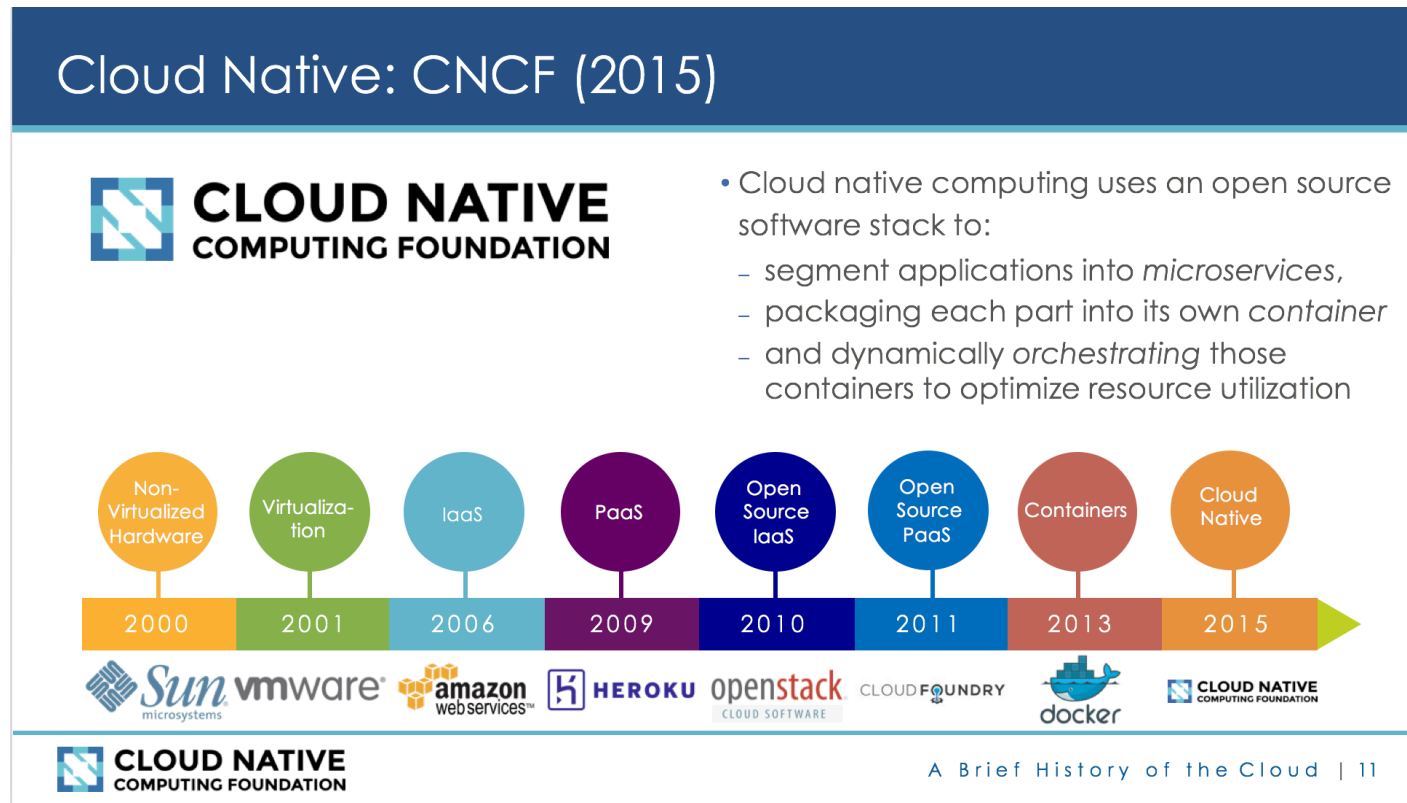
- SaaS (Software as a Service, SaaS)

- 소프트웨어 애플리케이션을 클라우드 환경을 통해 제공하며 별도의 설치, 유지보수 없이 애플리케이션을 사용할 수 있다.
 - ex) 인터넷을 통해 사용할 수 있는 대부분의 서비스

클라우드 네이티브

- 클라우드 네이티브란?

- 클라우드 환경의 이점을 최대한 활용하여 아키텍처를 설계하고 애플리케이션을 개발하여 서비스를 운영하는 방법론



클라우드 네이티브로의 발전 과정

클라우드 네이티브

- 클라우드 네이티브를 구성하는 요소
 - 컨테이너
 - 애플리케이션과 환경을 컨테이너로 만들어 실행 환경 간의 일관성을 보장한다.
 - 컴퓨팅의 최소 자원을 컨테이너를 기준으로 하여 자원을 효율적으로 사용할 수 있다.
 - 애플리케이션 환경을 경량화하여 빠른 배포가 가능하다.
 - 마이크로서비스 아키텍처(MSA, Micro Service Architecture)
 - 애플리케이션을 기능을 중심으로 작은 단위의 서비스로 분할한다.
 - 결합도를 낮추는 효과를 가지며, 장애가 발생하더라도 애플리케이션의 동작이 중지하지는 않는다.
 - 서비스에 적합한 환경(기술스택)을 기능에 맞게 선택하여 적용할 수 있다.
 - 지속적 통합 및 배포(CI/CD, Continuous Integration/Continuous Deployment)
 - 지속적 통합(CI, Continuous Integration)
 - 코드의 변경 사항을 감지하여 자동화된 파이프라인을 통해 빌드 및 테스트를 진행합니다
 - 지속적 배포(CD, Continuous Deployment)
 - 코드의 변경사항을 애플리케이션 운영 환경에 반영합니다.

클라우드 네이티브

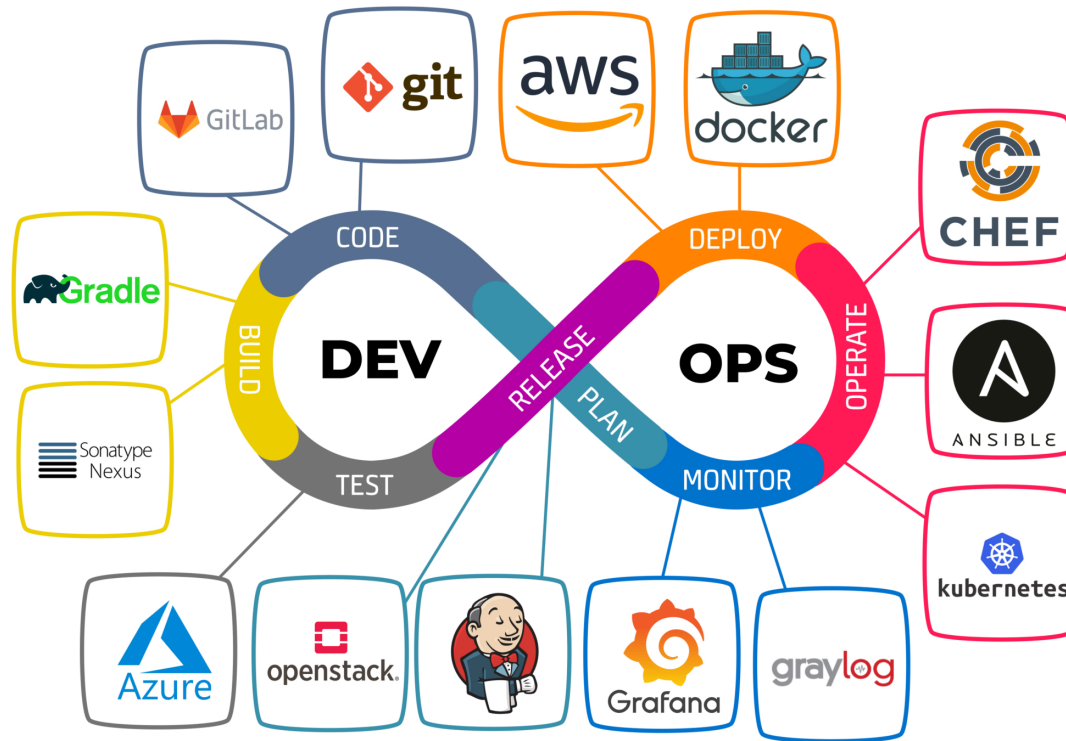
- 클라우드 네이티브를 구성하는 요소

- DevOps(Development + Operations)

- 문자 그대로 개발 조직과 운영 조직의 결합
 - 분리되어 있는 두 개의 조직 사이에서 발생하던 사일로 현상을 최소화하고, 소프트웨어 개발 주기를 단축시키는 방법론
 - 이를 통해 고객에게 지속적으로 가치를 제공할 수 있다.
 - 고객의 피드백을 받아 빠르게 제품을 개선해나가는 애자일 방법론과 비슷한 흐름을 갖고 있다.

클라우드 네이티브

- 클라우드 네이티브를 구성하는 요소



DevOps ++

- **SRE(Site Reliability Engineering, 사이트 신뢰성 엔지니어링)**
 - 안정적인 서비스 운영을 지원하기 위한 방법론으로 구글에서 개발되었다.
 - SRE에서는 시스템의 신뢰성과 안정성을 강조한다.
 - 주요 업무 및 목적
 - 모니터링
 - 안정적인 서비스 운영을 위해 애플리케이션의 성능 및 오류 등을 모니터링하여 개발 조직과의 협업을 통해 개선한다.
 - 자동화
 - 반복적으로 수행되는 작업을 자동화하여 효율성을 높이고, 시스템 운영에 대한 부담을 줄인다.

DevOps ++

- 플랫폼 엔지니어링(Platform Engineering)

- 제품 개발 조직이 개발에 집중할 수 있도록 플랫폼을 구축하고 관리한다.
 - 여기서 플랫폼은 인프라 구축, 배포 자동화, 서비스 모니터링 등을 개발자들이 쉽게 이용할 수 있도록 추상화시킨 것을 의미한다. IDP(Internal Developer Platform)이라고도 부름.
- *DevOps*의 최종진화라고 부르기도 한다.
- 주요 업무 및 목적
 - 제품 개발 조직의 업무 효율성 향상
 - 개발 조직에서 인프라와 다양한 도구들을 쉽게 사용할 수 있도록 추상화하여 지원하여 개발 속도를 향상 시킨다.

DevOps ++

