

1장. 시스템과 인프라 기초 지식

✓ Docker를 도입하기 앞서 알아둬야 할 시스템 기반의 개요와 시스템 기반을 다루기 위해 필요한 인프라 기술 학습

→ 개발한 app을 release하여 최종 사용자가 이용할 수 있도록 하려면 시스템 기반을 구축하고 그 위에 application의 실행 환경을 마련해야 함.

· 시스템 기반? application을 가동시키기 위해 필요한 Hardware, OS/middleware 등과 같은 infra.

· 미들웨어 (middleware)? 공통 서비스 및 기능을 app에 제공하는 software.

= 양쪽을 연결하여 data를 주고 받을 수 있도록 중간 (middle)에서 매개 역할을 하는 software.

· Docker? application 실행 환경을 작성·관리하기 위한 platform

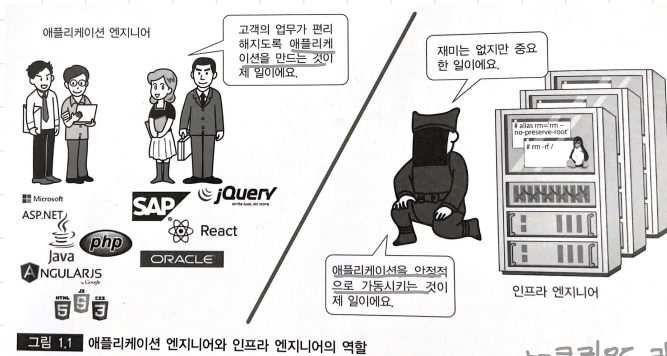
1.1. 시스템 기반의 기초 지식

* 폭포형 모델을 사용한 시스템 개발

[일반적인 역할 분담]

[application 실행 환경의 구축] → ("네트워크나 하드웨어에 능통한") 인프라 엔지니어 담당.

[application 개발] → ("업무 지식 및 프로그래밍", "테스트 방법과 같은 기술을 잘 아는") application 엔지니어 담당



[장점] 기업의 비즈니스 정보를 보안성 높게 관리.

[단점] 시스템 구축하는데 있어서 시간·비용

→ 클라우드 컴퓨팅 기술이 나오기 이전의 기업 인프라 구축의 일반적인 방식.

[현재] 클라우드 등장으로 흐름 변화

· 자사에서 데이터센터/기계를 보유

→ 온프레미스 환경에서 가동시켰던 서버들을, 클라우드 상의 가상 인스턴스로 옮김.

→ on-premise? 소프트웨어 등 솔루션을 클라우드 같은 원격 환경이 아닌 "직접적으로 보유한 전산실 서버에 직접 실행해 운영하는 방식"

· 데이터베이스나 네트워크와 같은 클라우드 서비스를 이용

→ 실행 환경의 구축 범위가 극도로 감소.

→ 짧은 사이클로 리스를 최소화하는 스타일로 변화

온프레미스 (on-premise) ← 반대 → 오프프레미스 (off-premise)

↓
클라우드 방식

· 그래서 기업에서 <보안성 높은 데이터 → 온프레미스 환경
" 낮은 " → 클라우드 환경

→ Cloud? 네트워크 전반에서

확장 가능한 리소스를 추상화, 풀링·공유하는 IT환경

→ application이 실행되는 환경.

[서버/스토리지 등의 자원을 미리 확보하고 이를 사용자 요청에 따라 제공한다는 개념.]

(1대의 물리 host 상에서 움직이는 시스템과는 달리)

· 클라우드를 구성하는 대부분의 기술: 분산환경에서 가동시키는 것이 기본.

· 분산환경 → 인프라 엔지니어가 수동으로 operation(운영)을 하지 않음.

→ 자동화된 툴을 사용하여 orchestration(오케스트레이션) 함.

(여러 대의 서버에 각 컨테이너를 적절히 배치하고 관리하는 것.)

① Pod(포드)? 여러 컨테이너를 묶는 것.

· 인프라엔지니어 → 인프라 기술 + (application 엔지니어 업무) 코드 작성 스킬 요구.

· application 엔지니어 → (인프라 엔지니어 업무) 제품 환경에 대한 배포·테스트 스킬 요구,
↳ OS(커널)나 네트워크 등과 같은 인프라 기술의 기초지식 필요.

✓ 여기서는 application 엔지니어가 Docker를 사용하기 위해 필요한 인프라 기술의 기초에 대해 설명