

실습_02주차(2)_Virtual Machine

가상화

하나의 컴퓨터에서 여러 개의 가상적인 컴퓨터를 만들어 사용하는 기술

애플리케이션 프로그램은 특정 ISA(명령어)+OS에 맞춰서 설계됨. 예를 들어, 매킨토시 앱은 PowerPC+ MacOS 위에만 돌아가고, 리눅스 앱은 x86 + Linux 위에서만 돌아감. 가상머신은 이러한 애플리케이션을 돌리는 데 있어서 생기는 한계점을 없애기 위해 등장

- **데스크톱 가상화** : 비즈니스에서 다음과 같은 상황이 발생

- Windows 10과 고객 관계 관리 소프트웨어가 설치된 데스크톱 컴퓨터를 필요로 하는 고객 서비스 팀
- 영업 애플리케이션을 위한 Windows Vista를 필요로 하는 마케팅 팀

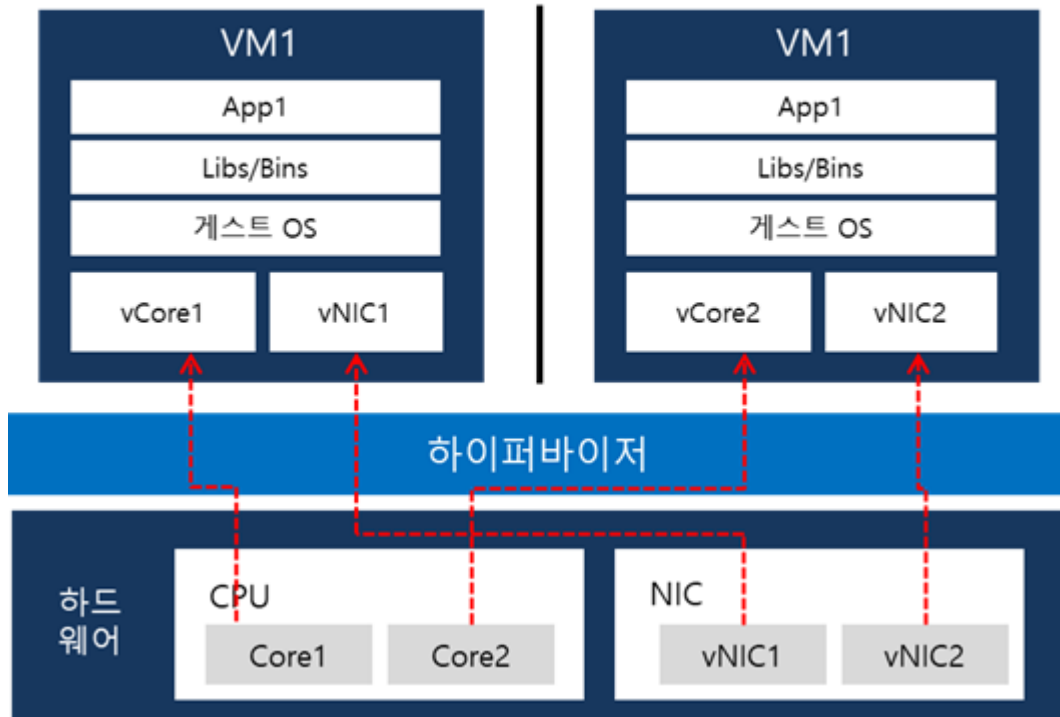
→ 각 os가 구비된 데스크톱 하드웨어를 사면 장비 가격 + 운용비 발생, 원격으로 접근할 수 있는 가상머신으로 os 가상화를 하면 이러한 문제 해결!

- **서버 가상화** : 여러 대의 서버를 물리적으로 운영하는 것보다 하나의 서버에서 여러 대의 서버를 운영 가능

→ 각 머신의 물리적 자원을 최대한 활용하고 투자 비용을 줄이기 위한 방법으로 사용

Hypervisor 가상화(Virtual Machine)

격리



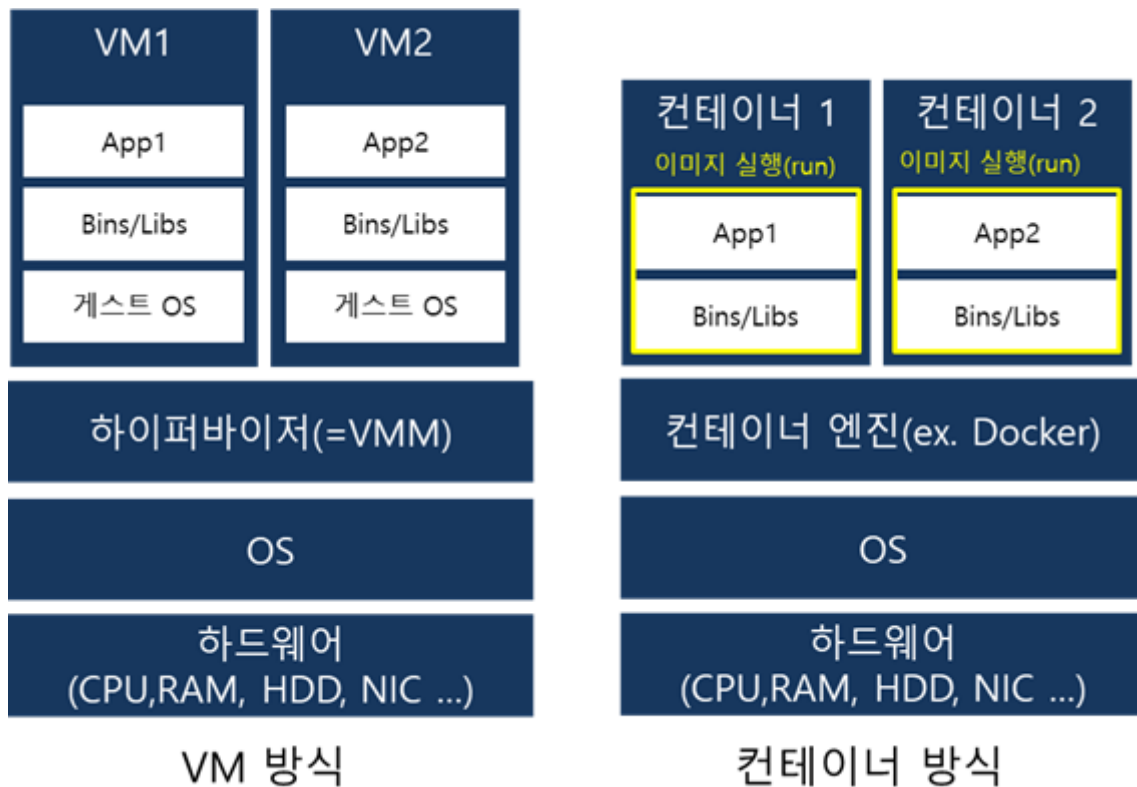
- Hypervisor : 호스트 시스템에서 다수의 게스트 OS를 구동할 수 있게 하는 소프트웨어
 - 하드웨어를 가상화하면서 하드웨어와 각각의 VM을 모니터링하는 중간 관리자 역할
- 물리적인 서버에서 하나 혹은 그 이상의 독립적인 운영체제가 돌아가는 구조
 - 즉, **물리적 서버의 OS 위에 여러 다른 독립적인 OS가 가상적으로 돌아가는 구조**
- 장점
 - 물리적 서버의 리소스를 더 효율적으로 사용가능
 - 각 VM은 논리적으로 **분리**되어 있어 한 VM에 오류가 발생해도 다른 VM으로 확산되지 않음
 - 다양한 테스트 환경에 적합(ex : 악성 코드를 분석할 때 감염을 방지하기 위해 사용됨)
 - 관리가 용이

- 단점
 - 게스트 OS 구동 자체에 들어가는 리소스가 많이 소모됨
 - 애플리케이션을 실행시킬 때에도 먼저 VM을 띄우고 자원을 할당한 다음 게스트 OS를 부팅하여 애플리케이션을 실행시켜야 함

ex) Oracle **VirtualBox**, VMWare, Parallels Desktop(Mac)

Container 가상화

- 호스트 운영체제 위에 여러 격리된 컨테이너를 만들어서 동작하는 가상화 기술
- 하나의 os 환경을 여러 개의 분리된 공간(컨테이너)로 만들고, 각 공간마다 다른 os 환경을 제공하는 방식
- 애플리케이션 실행에 필요한 라이브러리나 기타 구성 파일들을 이미지 형태로 배포할 수 있음
 - ex) Ubuntu 이미지 : Ubuntu를 실행하는데 필요한 모든 파일을 갖고 있음 , 이미지 다운 후 컨테이너를 생성 및 실행하면 내 작업 공간과는 분리된 Ubuntu 환경 세팅 가능



- 장점
 - 하이퍼바이저와 게스트 OS가 필요하지 않아 가벼움
 - 복제와 배포에 용이
 - VM에 비해 애플리케이션을 실행, 배포하는 과정이 가볍기 때문에 더 많은 애플리케이션을 구동할 수 있음

ex) Docker

실습

Virtual box를 통해 가상환경에서 Ubuntu를 설치, 각종 task를 수행

Virtual box 는 오라클에서 만든 가상머신 솔루션으로 무료로 제공되는 오픈소스

Ubuntu는 리눅스 배포판으로 다음과 같은 특징을 가지고 있음

- 다중사용자, 다중작업을 지원하기 때문에 **서버를 운영**하기에 적합
- **오픈소스**이기 때문에 커뮤니티가 방대
- 기본적으로 무료