[Kubernetes] 쿠버네티스와 컨테이너, 도커에 대한 기본 개념

원본 스크랩: 쿠버네티스와 컨테이너, 도커에 대한 기본 개념 : 네이버 포스트

요즘 IT 생태계에 관심이 많으신 분들이라면 쿠버네티스라는 단어를 들어 보셨을 텐데요. 쿠버네티스에 대해 개발자들의 관심이 높아지고 있고, 여러 대기업에서 새로운 시스템을 쿠버네티스를 기반으로 전환했다는 소식이 종종 들리고 있습니다. 그렇다면 쿠버네티스는 어떤 것이기에 이렇게 주목받고 있을까요?

쿠버네티스를 접해보지 않은 K는 쿠버네티스에 대해 궁금한 것이 많았습니다. 쿠버네티스를 시작하기 전부터 등장하는 여러 가지 용어와 자료가 이해되지 않아 저에게 도움을 요청했는데요, K의 질문을 재구성하여 쿠버네티스와 관련된 개념을 하나씩 알아보겠습니다!

Q. 쿠버네티스는 컨테이너를 오케스트레이션 하는 도구라는 글을 많이 봤어요. 도커는 컨테이너를 다루는 도구라고 설명하던데, 그렇다면 쿠버네티스는 도커를 다루는 도구인가요?

A. 용어를 먼저 정리해 보면 어떨까요? 컨테이너, 컨테이너 런타임, 도커를 정리한 후 쿠버네티스에 대해 알아보겠습니다.

용어	뜻
컨테이너	앱이 구동되는 환경까지 감싸서 실행할 수 있도록 하는 격리 기술
컨테이너 런타임	컨테이너를 다루는 도구
도커	컨테이너를 다루는 도구 중 가장 유명한 것
쿠버네티스	컨테이너 런타임을 통해 컨테이너를 오케스트레이션 하는 도구
오케스트레이션	여러 서버에 걸친 컨테이너 및 사용하는 환경 설정을 관리하는 행위

https://post.naver.com/viewer/postView.naver?volumeNo=34056187&memberNo=36733075#
먼저 컨테이너부터 살펴보죠.컨테이너란, 우리가 구동하려는 애플리케이션을 실행할 수 있는 환경까지 감싸서, 어디서든 쉽게 실행할 수 있도록 해 주는 기술이에요.

여러분이 PC에 프로그램을 설치할 때를 떠올려보세요. 특정 경로에 맞춰 설치를 해야 하거나, 내 컴퓨터에 필요한 옵션을 일일이 맞춰주느라 설치 과정에서 힘들었던 경험이 있을 텐데요. 컨테이너는 이러한 환경까지 모두 포함하여 독립적으로 프로그램을 실행할 수 있도록 도와주는 기술입니다. 컨테이너 환경을 묶어서 배포한 컨테이너 이미지라는 프로그램을 내려받아 구동하면 실행되기 때문에, 각종 설정 과정이 줄어 들어서 좀 더 편하게 사용할 수 있어요.

컨테이너를 사용할 때 필요한 도구가 컨테이너 런타임입니다. 컨테이너를 쉽게 내려받거나 공유하고 구동할 수 있도록 해주는 도구인데요. 종류도 여러 가지가 있어요. 그중가장 유명한 것이 도커죠. 물론 도커가 사용하는 컨테이너 규격은 표준화되어 있기 때문에 도커가 아닌 다른 컨테이너 런타임들도 도커로 만든 컨테이너를 사용할 수 있습니다.

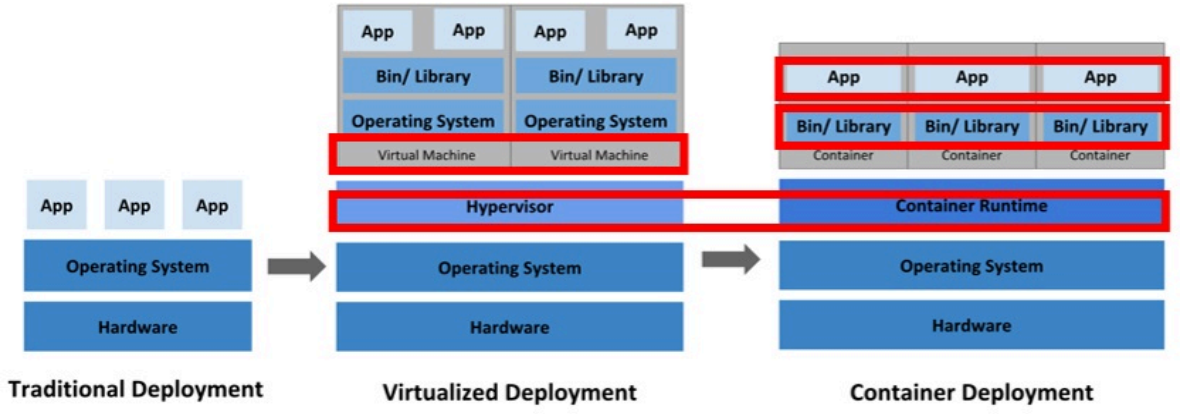
쿠버네티스는 컨테이너 런타임을 통해 컨테이너를 다루는 도구를 말해요. 쿠버네티스가 해주는 일은 여러 서버(노드)에 컨테이너를 분산해서 배치하거나, 문제가 생긴 컨테이너를 교체하거나, 컨테이너가 사용할 비밀번호나 환경 설정을 관리하고 주입해 주는 일 등입니다. 이것을 컨테이너 오케스트레이션이라고 하죠.

Q. 쿠버네티스가 컨테이너를 다루는 도구라면 도커를 다루는 도구는 아니란 말이네요?

A. 네 앞에서 언급했듯이, 쿠버네티스의 역할은 컨테이너를 분산 배치, 상태 관리 및 컨테이너의 구동 환경까지 관리해 주는 도구이고, 도커는 컨테이너를 다루는 도구(컨테이너 런타임) 중 하나입니다. 쿠버네티스는 컨테이너를 다루기 위해 도커 이외에도 다양한 컨테이너 런타임 소프트웨어를 사용할 수 있습니다.

Q. 쿠버네티스를 검색하면 가상머신과 컨테이너 비교에 대한 그림이 매번 나오는데 이해가 잘 안돼요. 쉽게 설명해 주실 수 있을까요?

A. 음.. 어떤 그림인지 알 것 같아요. 아래 '애플리케이션 배포 환경의 변화'를 말하는 거죠?



https://post.naver.com/viewer/postView.naver?volumeNo=34056187&memberNo=36733075#

애플리케이션 배포 환경의 변화(Traditional Development)
(출처 : https://kubernetes.io/ko)

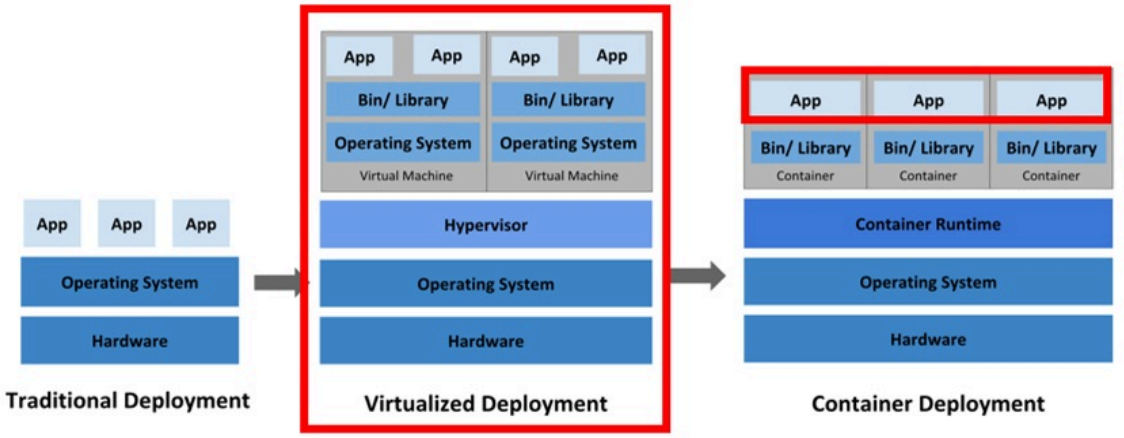
아마 이 그림을 이해하는 데 있어서 가장 어려웠던 것은 첫 번째로는하이퍼바이저(Hypervisor)와 컨테이너 런타임(Container Runtime)과 같은 용어가 익숙하지 않았기 때

문일 거예요. 두 번째는VM, Bin/Library, App이 어떤 의미인지 와닿지 않아서였을 텐데요. 좀 더 예시를 들어서 설명해 보겠습니다.

맨 왼쪽 Traditional Deployment(전통적 배포)는 가상화 이전, 그러니까 오래전부터 쓰이던 방식입니다. 물리적인 컴퓨터 한 대에 하나의 OS를 깔고 여러 가지 프로그램을 설치하는 방식이죠. 우리는 PC 한 대에 윈도우를 하나 설치하고, 여러 가지 게임이나 워드프로세서 등을 깔아서 사용하잖아요. 이와 비슷한 방식이라고 생각하면 됩니다. 가장 오래되고 단순한 방식이며 단일 목적 시스템이라면 이것으로도 별 무리가 없겠죠.

그렇다면, 이 방식이 가진 문제점에 대해 한 번 생각해 볼까요? 인터넷 뱅킹을 위해 보안 프로그램을 많이 설치했더니 게임이나 웹 브라우저 성능이 떨어지는 걸 경험, 누구나 한 번쯤 해보셨을 텐데요. 한 대의 컴퓨터에서 모든 것을 처리하려고 하면 어떤 프로그램 동작이 다른 프로그램의 동작을 간섭하거나, 특정 프로그램이 성능을 독점할 경우 또 다른 프로그램의 성능이 떨어지는 단점이 있습니다. 성능이 떨어지는 정도에 그치지 않고 프로그램의 중단까지도 유발할 수도 있죠. 그렇다면 이 문제를 해결하기 위해서는 어떻게 해야 할까요? “인터넷 뱅킹 전용 컴퓨터를 구입해 두 대를 쓰면 되지 않을까? 그럼 게임도 쾌적하게 하고 인터넷 뱅킹은 필요할 때만 할 수 있을 테니!” 이렇게 생각해 볼 수도 있겠지만, 그러기엔 비용이 너무 많이 들겠죠!

그렇다면 어떤 식으로 해결할 수 있을지 살펴보겠습니다.



https://post.naver.com/viewer/postView.naver?volumeNo=34056187&memberNo=36733075#

애플리케이션 배포 환경의 변화(Traditional Development)

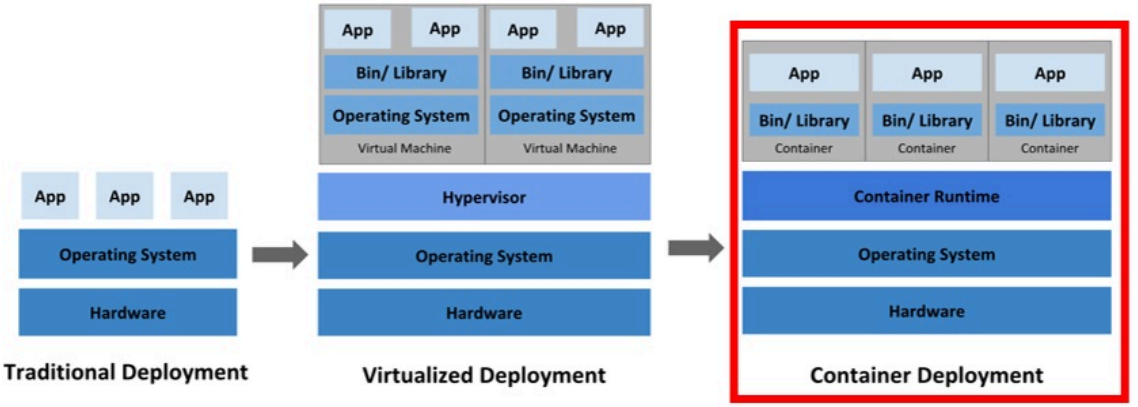
(출처 : https://kubernetes.io/ko)

이 문제를 해결하기 위해 등장한 해법이 위 그림의Virtualized Deployment(가상화 배포)에요. 가상머신(Virtual Machine)을 기반으로 배포를 하는 방법이죠. 중간에 위치한하이퍼바이저는 하나의 시스템 상에서 가상 컴퓨터를 여러 개 구동할 수 있도록 해 주는 중간 계층

을 의미한다는 정도로만 알고 계시면 될 거예요. 그리고App은 실행하고자 하는 프로그램, Bin/Library는 프로그램이 실행하는데 필요한 환경과 관련된 파일이라고 생각하면 됩니다.

가상머신은 말 그대로 가상 컴퓨터입니다.컴퓨터이기 때문에 우리가 컴퓨터를 조립할 때 CPU, 메모리 및 SSD를 장착하듯, 가상머신에도 CPU, 메모리, 저장 장치 등을 개별적으로 할당할 수 있죠. 앞에서 말한 게임 프로그램과 인터넷 뱅킹 보안 프로그램이 간섭을 일으켜 문제가 된다면 두 프로그램을 각각의 가상머신에서 실행시켜보세요. 하나의 컴퓨터 안에서 두 개의 가상화된 컴퓨터가 동작하기 때문에 서로 간섭을 일으키지 않게 됩니다. 가상머신 성능을 조절해 게임 컴퓨터에는 좀 더 많은 CPU와 메모리를, 인터넷 뱅킹용 컴퓨터에는 적은 CPU와 메모리를 할당하여 사용할 수 있겠죠? 또한 서버와 같이 다중화와 분산 처리가 중요한 시스템이라면 시스템 자원 상황에 따라 가상머신 개수를 늘리고 줄이는 것도 좀 더 유연하게 처리할 수 있습니다.

다만 이 방식이 전통적 배포 방식보다는 분명 효율적이지만, 가상머신은 완전한 컴퓨터이고 가상머신에 일일이 운영체제를 설치해야 하기 때문에 컨테이너 중심의 배포(Container Deployment)보다는 무거운 편이에요.



https://post.naver.com/viewer/postView.naver?volumeNo=34056187&memberNo=36733075#

애플리케이션 배포 환경의 변화(Traditional Development) * 출처 : https://kubernetes.io/ko

마지막으로 Container Deployment(컨테이너 중심의 배포)에 대해 살펴보겠습니다. 하이퍼바이저라는 부분이 Container Runtime으로 대체되었고, Virtual Machine이라고 된 부분은 Container로 대체가 되었죠?컨테이너는 가상머신과 달리 프로그램 구동을 위해서 OS를 매번 설치할 필요가 없어요.그림에서 보는 것과 같이 OS는 하나만 사용합니다.

컨테이너 기반 배포는 전통적 배포 위에 Container Runtime이 올라가 있는 것 같은데 물리적인 컴퓨터 상에서만 유효한 것인지 궁금하실 텐데요. 꼭 그렇지는 않습니다. 컨테이너 런타임 위에 OS와 하드웨어가 층으로 쌓여 있는 그림을 보고 전통적인 배포 위에서 컨테이

너 런타임을 올렸다고 오해를 하곤 하는데, 컨테이너는 OS 하단이 어떻게 동작하는지 직접 관심을 두지 않아요. 그래서 가상머신 위에 올라간 OS에서 컨테이너 기반 배포를 하는 것이 가능합니다.

앞에서 게임과 인터넷 뱅킹 프로그램이 한 컴퓨터에 설치된다면, 서로 간섭을 일으켜 성능 저하나 오류를 발생시킬 수 있다고 말씀드렸죠. 지금부터 이 문제를 컨테이너 중심의 배포 방식에서는 어떻게 해결하는지 알아보기 위해 게임과 인터넷 뱅킹 소프트웨어가 각각 컨테이너로 배포된다고 가정해 보겠습니다.(물론 이럴 일은 없겠지만, 동작 방식의 예시라고 생각해 주세요)

이때 게임과 인터넷 뱅킹은 하나의 OS 상에서 구동됩니다. 이것만 보면 전통적 배포에서 하나의 OS 위에 프로그램을 여러 개 구동시킨 것과 별 차이가 없어 보여요. 그렇지만 컨테이너 중심의 배포는 여기서 중요한 기술적인 차이점이 하나 있습니다. 게임과 인터넷 뱅킹이 각각 실행되면서 '이 컴퓨터에서 나만 구동되고 있다'라고 판단할 수 있도록, 실제로 두 프로그램 간에 간섭을 일으킬 수 없는 장벽을 쳐요. 이러한 장벽을 치는 것과 동시에 OS는 게임과 인터넷 뱅킹이 사용할 수 있는 CPU, 메모리 등의 자원 또한 독립적으로 사용할 수 있도록 할당하고 관리합니다. 이러한 과정을 통해 게임과 인터넷 프로그램은 스스로를 '서로 다른 컴퓨터에서 깔려 있다'고 생각하게 됩니다. 물론 OS의 관점에서 보자면 둘 다 OS 상에서 구동되는 프로그램이지만 말이죠. 이와 같은 컨테이너 동작 방식을 OS 커널을 공유하는 가상화라고 표현하기도 해요.

이때 주의할 점이 하나 있습니다. 컨테이너는 OS를 공유하는 방식이기 때문에, 어떤 프로그램의 문제가 다른 프로그램을 간섭할 수는 없습니다. 그러나 내 프로그램의 문제가 OS에 문제를 일으킬 경우에는 OS에서 구동 중인 전체 컨테이너의 문제가 될 가능성이 있죠. 우리는 이 점에 신경을 써야 합니다.

◎ 쿠버네티스를 만나는 여러 가지 방법
◎ 쿠버네티스를 이루고 있는 여러 가지 구성 요소
◎ 쿠버네티스 클러스터 운영자를 위한 모니터링
◎ 클라우드 오케스트레이션을 위한 쿠버네티스와 오픈시프트의 차이◎ 쿠버네티스 보안, 어떻게 해야 할까?
◎ 쿠브플로우(Kubeflow): 쿠버네티스 기반의 AI 플랫폼

지금 말씀드린 기술에 대해 자세하게 설명하지는 않을게요. 다만, 리눅스를 기준으로 내가 실행한 프로그램이 독립된 환경에서 실행되는 것처럼 격리시켜주고, CPU, 메모리 및 저장장치와 같은 자원도 실행한 프로그램이 독립적으로 쓸 수 있도록 해주는 namespace 및 cgroup이라는 기술이 있다는 것만 알아 두시면 됩니다! 설명이 좀 길었죠? 마지막으로 다시 정리해 볼게요.

	전통적인 배포	가상머신 기반 배포	컨테이너 기반 배포
컴퓨터	물리적인 컴퓨터 1대	물리적인 컴퓨터 1대에 다수의 가상머신 존재	컴퓨터 형태에 영향 받지 않음
OS	물리적인 컴퓨터 1대에 OS 1개가 설치됨	물리적인 컴퓨터 OS 1대 + 다수의 가상머신에 각각 OS설치	컴퓨터 형태와 관련 없이 설치된 OS 1개
리소스	컴퓨터 1대의 자원을 여러 프로그램이 나눠 씀	하이퍼바이저를 통해 가상머신 별 개별적 자원 할당	OS에서 프로그램별로 자원을 할당하고 관리함
격리 수준	격리되지 않아 프로그램 간 간섭 발생	각 가상머신이 완전하게 격리됨	프로그램 실행 환경은 격리되지만 OS 환경은 공유됨
문제 전이 가능성	특정 프로그램의 문제가 시스템 전체의 중단을 가져올 수 있음	특정 가상머신의 문제가 다른 가상머신의 문제로 전이될 가능 성이 낮음	특정 프로그램의 문제가 다른 프로그램에 간섭을 일으키지는 않지만, 특정 프로그램의 문제 가 OS 문제를 유발할 경우 시스템 중단 가능성 있음

https://post.naver.com/viewer/postView.naver?volumeNo=34056187&memberNo=36733075#

 # Reference[1] 컨테이너 인프라 환경 구축을 위한 쿠버네티스/도커 (조훈, 심근우, 문성주 지음)