

컨테이너 기술을 꼭 도입해야 하는 이유

IT 기획 / 운영 / 개발 측면에서 컨테이너 기술의 필요성



OPENNARU UNCONTACT WORKSHOP CONTENTS

04 IT 기획자 측면에서
컨테이너 기술의 필요성

05 개발자 측면에서
컨테이너 기술의 필요성

06 운영자 측면에서
컨테이너 기술의 필요성

07 컨테이너 기술과
가상화 기술 비교

IT 기획자 측면에서 컨테이너 기술의 필요성



디지털 트랜스포메이션이란

디지털 기술의 급격한 발전으로 인한 혁신적인 발전 방식을 뜻합니다. 즉, 비즈니스 동향, 전략, 사례, 솔루션, 서비스, 플랫폼의 혁신을 포함합니다. 기업은 언제나 기술 변화에 맞춰 진화해야 했습니다.

이제는 디지털 서비스와 소프트웨어를 신속히 제공할 수 있는 기업의 역량이 비즈니스의 성공과 실패를 좌우하게 되었습니다.

특히 코로나19 사태로 인한 비대면 접촉 비즈니스가 일상화 되면서 이러한 흐름은 한층 가속화 중에 있습니다. 기업들은 새로운 비즈니스모델, 제품 및 서비스를 위해 클라우드, 모빌리티, 빅데이터, AI 등의 디지털 기술들을 활용할 수 있는 역량을 높이기 위한 노력과 경쟁을 하고 있습니다.

기획자 측면의 장점

특징	설명
디지털 트랜스포메이션 기반 구축	디지털 트랜스포메이션이 실현하려면 IT 서비스 개발과 개선 '속도'가 비즈니스 우위를 확보하기 위한 필수적인 조건
비용 절감	높은 시스템 자원 사용율에 따른 비용 절감
	Guest OS 에 대한 라이선스 제거
	Guest OS 유지보수비용 및 관리비용 제거
하이브리드 클라우드 실현	Private 클라우드 구축하여 Public 클라우드로 즉시 이전 가능
	Private Cloud 와 Public Cloud 를 단일한 기술과 운영방법으로 관리
	컨테이너를 기반으로 하는 오픈소스이기 때문에 사용자들이 특정업체에 종속되지 않고 클라우드의 환경들을 이전
DevOps 기반	컨테이너는 DevOps 빌드/테스트/배포 파이프라인을 간소화

개발자 측면에서 컨테이너 기술의 필요성



개발자 입장에서 볼때,

개발 PC에서 검증된 코드를 컨테이너를 구성하면 더이상 소프트웨어를 구성할 필요가 없습니다. 개발에 대한 민첩성과 효율성이 향상되고, 개발팀의 생산성과 만족도도 향상됩니다.

컨테이너는 가상머신과 마찬가지로 애플리케이션을 관련 라이브러리 및 종속항목과함께 패키지로 묶어 소프트웨어 서비스를 위한 환경을 분리할 수 있도록 해줍니다.

가상화 기술과 컨테이너 기술이 유사점은 여기까지입니다. 컨테이너를 사용하면 개발자와 IT 운영팀이 훨씬 작은 단위로 업무를 수행할 수 있어 이점이 훨씬 많습니다.

개발자 측면의 장점

특징	설명
효율적인 개발 환경 구축	개발 환경 구축 기간 단축 / OS가상화로 격리된 테스트 환경 구축
	기존가상화대비작은시스템리소스로개발환경구축
배포 편의성	이미지를 통한 빌드, 배포 자동화
	개발자 환경/테스트 환경/스테이징 환경/운영 환경에 대한 일관성 보장으로 장애 요인 제거와 장애 원인 파악 시간 단축
민첩한 개발	컨테이너를 통한 짧은 주기로 요구사항 정의와 릴리즈를 반복하는 AGILE DEVELOPMENT 지원
서비스 무정지 환경 제공	서비스 정지 없이 시스템 운영이 가능하여 배포시간과 횟수에 대한 제약이 없음
마이크로 서비스 아키텍처	마이크로 서비스는 컨테이너로 구성하고 배포, 운영하는 것이 매우 유리
DevOps 기반	컨테이너는 DevOps 빌드/테스트/배포 파이프라인을 간소화

운영자 측면에서 컨테이너 기술의 필요성



운영자 입장에서 볼 때,

인프라 레벨에서 애플리케이션의 종속성에 대한 고민을 할 필요가 없습니다.

인프라는 더욱 클라우드에 가까운 기술들로 관리할 수 있습니다.

온-프레미스나 클라우드를 가리지 않고 동일한 관리 방법으로 빌드/배포/운영할 수 있습니다.

가상 머신은 하드웨어 스택을 가상화 합니다. 컨테이너는 이와달리 OS를 가상화하여 여러개의 컨테이너를 OS 커널에서 직접 실행 합니다. 컨테이너는 기존의 가상화 기술보다 훨씬 가볍게 동작하며, OS 커널을 공유하고, 시작 시간이나 종료 시간이 빠르고, 메모리를 훨씬 적게 차지합니다.

운영자 측면의 장점

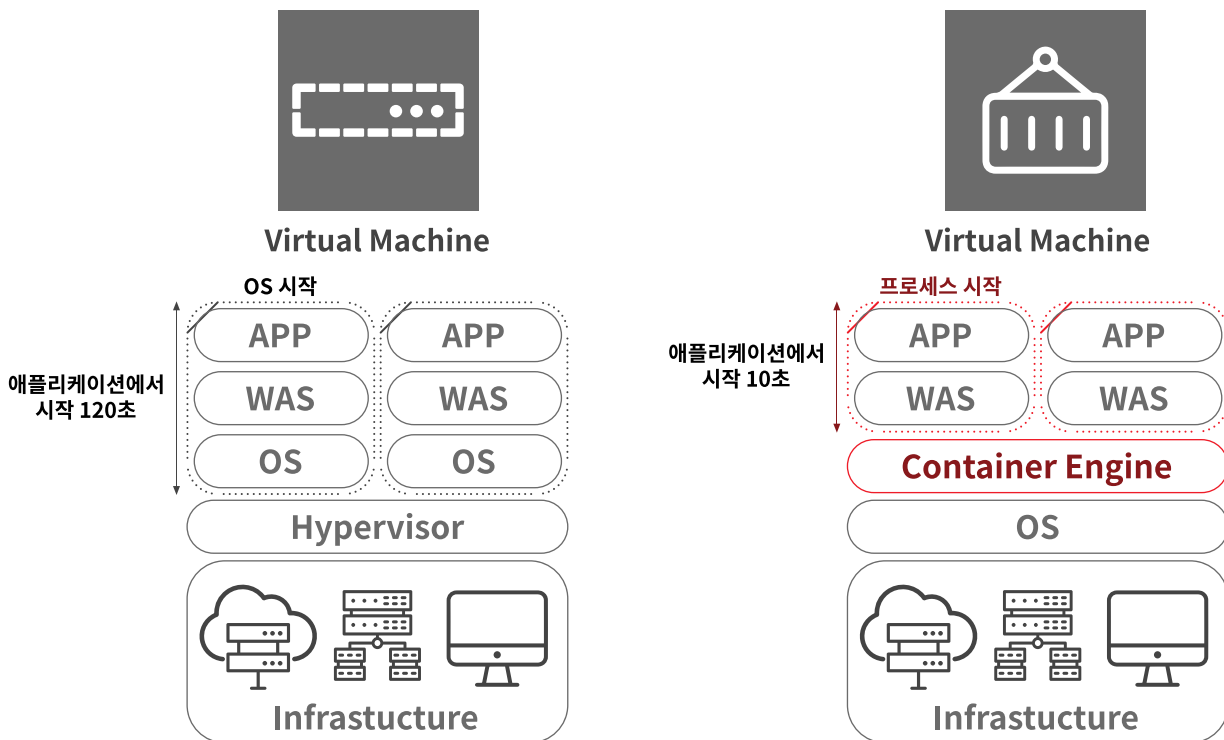
특징	설명
낮은 오버헤드와 빠른 시작	최소한의 CPU와 메모리만 사용하여 비용 절감과 부하가 작아 고성능 제공
	Guest OS가 없기 때문에 OS 부팅없이 애플리케이션을 실행하여 빠른 시작(호스트OS 에서프로세스로실행)
높은 이동성 (Portability)	Public Cloud (AWS, Azure, Google)와기업내에서Linux 운영체제라 면어디서나운영및이식이용이함
구축 기간 단축	컨테이너 환경은 개발, 스테이징, 운영 환경을 단순한 복사로 구축하여 작업 시간을 단축하고 일관성을 제공하여 환경에 의한 문제 원인 제거
장애 대응	배포, 시스템 유지 보수, 장애 발생시 무정지 작업이 가능
	컨테이너 이미지 단위로 배포하고 운영하기 때문에 장애시 전환 시간을 단축
클라우드 네이티브 운영 환경 실현	이미지 형태의 배포로 환경 차이에 의한 장애원인 제거
	스케줄링(Scheduling), 컨트롤링(Controlling), 자가 복구(Self Healing), 오토 스케일링(Auto Scaling), 롤링 업데이트(Rolling Update)

컨테이너 기술과 가상화 기술 비교

가상 머신은,

Guest OS 에 BIOS, 가상CPU, 가상메모리, 가상디스크, 가상NIC 등을 제공하여 마치 물리 서버에서 실행되는 것처럼 보입니다. 이 때문에 게스트OS 입장에서는 OS 시작과 정지 등의 운영에 있어서 물리서버와 아무런 차이점 없이 운용이 가능합니다.

컨테이너는 이와달리 OS를 가상화하여 여러개의 컨테이너를 OS 커널에서 직접 실행합니다. 컨테이너는 기존의 가상화 기술보다 훨씬 가볍게 동작하며, OS 커널을 공유하고, 시작 시간이 나 종료 시간이 빠르고, 메모리를 훨씬 적게 차지합니다.



구분	하이퍼 바이저형 가상화	컨테이너형 가상화
시작 시간	몇 분	몇 초
이미지 크기	수GB ~ 수백GB OS를 포함하여 애플리케이션과 필요한 런타임 소프트웨어	~ 수백MB 애플리케이션과 런타임 소프트웨어만
Guest OS	Windows/Linux 등 다양한 선택 가능	호스트OS 와 동일한 OS
이식성	대부분 가상 이미지에 대한 변환이 필요함	컨테이너 이미지 그대로 사용 가능
데이터 관리	VM 내부 또는 연결된 스토리지에 저장	컨테이너 내부에 있는 데이터는 종료시 소멸되며, 필요에 따라 스토리지를 이용하여 저장
Guest OS와의 관계	Guest OS는 하드웨어(가상)로 인식	Host OS를 커널수준으로 분리하여 OS를 가상화 형태로 사용하여 필요에 따라 호스트와 리소스 공유 가능



신청 바로가기

 **opennaru** 비대면 클라우드 네이티브 워크샵 문의/신청

오픈나루 박준영 차장

010-9913-9780

junyoung.park@opennaru.com

 **opennaru**

Copyright © 2021 Opennaru, Inc. All Rights Reserved.

무단 전재 재배포 금지

본 PDF 문서는 오픈나루(주)의 자산으로, 저작권법의 보호를 받습니다.

오픈나루(주)의 허락없이 PDF문서를 온라인 사이트 등에 무단 게재, 전재 하거나 유포할 수 없습니다.

 **opennaru** 비대면 클라우드 네이티브 워크샵 문의/신청

오픈나루 박준영 차장

010-9913-9780

junyoung.park@opennaru.com