IDG
STRATEGIC
MARKETING SERVICES

IDG Summary

컨테이너에 대한 6가지 오해

컨테이너(Container) 기술의 성장세가 눈부시다. 불과 3년 만에 거의 모든 운영체제 업체와 클라우드 서비스 업체의 지원을 받고 있다. 그러나 단기간에 급성장하면서 컨테이너에 대한 맹신 혹은 근거 없는 우려가 동시에 나타나고 있다. 운영체제와 보안, 가상화, 이동성 등의 측면에서 컨테이너 기술에 대해 잘못 알려진 내용을 자세히 살펴보고, 기업이 더 합리적으로 컨테이너 기술을 도입하는 방안을 제시한다.

컨테이너에 대한 6가지 오해

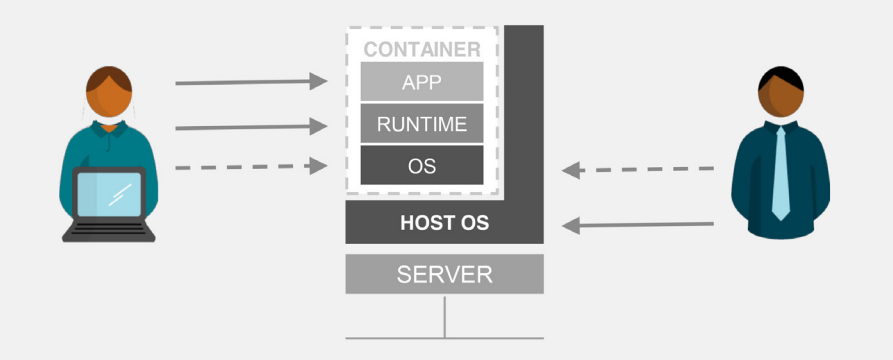
김호중 | 한국레드햇 부장

컨테이너(Container) 기술의 성장세가 눈부시다. 대표적인 오픈소스 컨테이너 기술인 '도커(Docker)'는 발표된 지 불과 3년 만에 여러 운영체제 개발업체와 클라우드 서비스 업체의 지원을 받고 있고, 관련 커뮤니티와 툴 업체의 움직임도 활발하다. 컨테이너는 운영체제 가상화 기반에서 애플리케이션과 이를 실행하는 데 필요한 요소를 묶은 일종의 패키징 기술이다. 여러 개 실행시키거나 다른 운영환경으로 옮겨서 실행할 수도 있어 간편하게 애플리케이션을 운영, 확장할 수 있다.

컨테이너 기술이 급부상한 또 다른 이유는 개발과 운영의 틈새를 메우는 이른바 '데브옵스(DevOps)'에 도움이 되기 때문이다. 그동안 개발자가 만든 시스템을 운영 환경으로 넘겨 배포하는 과정에서 문제가 생겨 시스템 가동 시기가 늦어지는 일이 많았다. 이때 컨테이너를 이용하면 패키징한 애플리케이션을 운영 환경에 그대로 이식해 실행할 수 있다. 개발자는 개발에 더 집중하고, 관리자는 손쉽게 배포, 관리할 수 있다. IT 아키텍트는 테스트 기간과 배포 시 오류를 줄이면서도 필요에 따라 유연하게 인프라를 확장할 수 있다.

물론 기대만 있는 것은 아니다. 시장조사업체 포레스터 리서치가 2015년 IT와 개발 관련 의사결정권자 171명을 대상으로 조사한 결과를 보면 응답자의 절반 이상인 53%가 컨테이너와 호스트 관련 보안에 대해 우려를 표시했다. 많은 기업이 제이보스(JBoss), 아파치, 마이SQL 등 오픈소스 툴을 컨테이너 이미지로 만든 것을 외부에서 가져다 사용하는데, 이 이미지 안에 백도어가 있는지 알 수 없고 검증하기도 힘들기 때문이다.

그림 1 | 리눅스 컨테이너 개념도



컨테이너에 대한 6가지 오해

이처럼 기대와 우려가 공존하는 가운데, 문제는 컨테이너에 대한 오해가 기업 전반에 퍼져 있다는 점이다. 이제 본격적으로 성장하는 기술인만큼 시행착오를 줄이고 장점을 극대화하기 위해 컨테이너 기술에 대한 정확한 이해가 절실하다. 지금부터 컨테이너에 대한 오해 6가지를 자세히 살펴보자.

오해 1. 컨테이너는 새로운 기술이다?

많은 기업이 컨테이너를 완전히 새로운 기술로 생각하고 소극적으로 접근한다. 그러나 사실은 그렇지 않다. 도커가 처음 세상에 공개된 자리는 2013년 파이콘(Pycon) 서밋 행사였다. 그러나 LXC(LinuX Container), 네임스페이스, 제어그룹(Control Groups), SE리눅스(SecurityEnchanced Linux) 등 도커의 주요 요소 기술은 이미 20년 가까이 된 것들이다. 운영체제 가상화 기술은 이미 2000년 프리BSD의 제일스(Jails)에서도 사용됐다. 리눅스 보안 모듈인 SE리눅스가 나온 것은 2003년, 제어그룹과 네임스페이스 기술이 나온 것은 각각 2007, 2008년이다.

도커가 기존 기술을 잘 조합해 파괴적인 컨셉으로 내놓은 것은 충분히 평가할만하다. 그러나 도커는 신기술이 아니다. 오래된 리눅스 가상화 기술이 없었다면 세상에 나올 수 없었다. 이는 곧 리눅스를 접해본 기업이라면 손쉽게 도커의 세계에 발을 들여놓을 수 있음을 의미한다. 가상화에 안주하거나 혹은 도커를 가상화와 전혀 다른 기술로 보고 외면하기보다는 기업 환경에 따라 잘 활용할 수 있는 방안을 고민해야 한다.

오해 2. 컨테이너는 기업용으로 준비가 안됐다?

아직 낯선 기술처럼 보여 기업용으로 쓰기에는 한계가 있다고 생각할 수 있다. 그러나 실제로는 그렇지 않다. 컨테이너 기술은 집적도와 배포 속도, 성능 면에서 충분히 성숙했다. 컨테이너는 VM(Virtual Machine: 가상 머신)보다 10배 정도 집적도가 높다. 기존에는 서버 1대에 VM을 10대 설치했다면 컨테이너는 100개 이상 운영할 수 있다. 이는 컨테이너 구조에 하이퍼 바이저와 VM 운영체제 계층이 없어 물리적인 자원을 덜 사용하기 때문이다. 일반적으로 가상화 환경에서는 이들이 전체 자원의 10~20% 정도를 사용하는 것으로 알려졌다.

배포 속도는 경쟁자가 없을 정도다. 물리 서버에 운영환경을 구축하려면 보통 27시간 정도 소요된다. VM에서는 템플릿 기능을 이용한다고 해도 12분 정도 걸린다. 반면 컨테이너 인스턴스를 새로 만드는 데는 10초면 충분하다. 가상화 기술이 급속히 확산한 배경에는 장비의 활용효율을 높인 것 외에 새로운 환경을 빠르게 구성한다는 점이 주효했다. 컨테이너는 이러한 시간을 더 획기적으로 단축한다. 특히 VM은 환경을 구성한 후 데이터를 올리고 다른 시스템과 연동하는 등의 수작업이 필요하지만, 컨테이너를 쓰면 이런 작업까지 컨테이너 이미지에 넣어 생략할 수 있다.

컨테이너 기술은 성능 면에서 많은 장점이 있다. 레드햇의 테스트 결과를 보

그림 2 | 물리서버-가상머신-컨테이너 간 설정 시간 비교

면 계산식 연산의 경우 베어메탈과 같은 성능을 나타냈고 분석작업 중심의 애널리틱스 앱 실행 성능은 베어메탈과 비교해 불과 6% 떨어지는 것으로 나타났다. 가상화 기술이 10~20% 정도 성능을 잡아먹는 것을 고려하면 놀라운 결과임을 알 수 있다.

무엇보다 컨테이너를 지원하는 퍼블릭 클라우드 서비스가 이미 상용화됐다는 점은 이 기술이 기업용으로 손색이 없다는 증거다. 구글은 구글문서나 구글드라이브 등의 자체 서비스에 컨테이너를 사용 중이며, 아마존 AWS나 MS 애저(Azure)도 2015년부터 컨테이너 서비스를 시작했다.

오해 3. 운영체제는 중요하지 않다?

컨테이너 기술은 하드웨어 위에 호스트 운영체제를 설치하고 그 위에 애플리케이션과 라이브러리를 패키지로 묶은 컨테이너를 올리는 구조로 돼 있다. 이 때문에 많은 실무자가 운영체제는 중요하지 않다고 생각한다. 그러나 실제로는 반대다. 오히려 컨테이너를 잘 관리하려면 이를 지원하는 운영체제의 안정성이 필수적이다.

예를 들어 아파치, 제이보스 등을 설치한 웹서비스 환경에서 SSL 라이브러리 설치해 https 인증을 하는 환경을 가정해 보자. 기존의 베어메탈 환경이라면 여기서 SSL 라이브러리에 문제가 생겼을 때 당장 이와 연결된 모든 애플리케이션에서 문제가 발생한다. 단일 라이브러리를 공유하는 구조이기 때문이다. 반면 컨테이너는 내부 라이브러리를 컨테이너별로 사용하므로, 한 컨테이너가 장애를 일으켜도 다른 서비스는 영향을 받지 않는다. 운영체제가 이러한 컨테이너의 특성을 잘 알고 지원해야 안정적으로 컨테이너를 관리할 수 있다.

라이브러리 문제는 호스트 운영체제 측면에서도 중요하다. 군소업체의 장비나 주문 제작한 장비를 사용하다 보면 드라이버를 지원하지 않아 장애가 발생하곤 한다. 특히 컨테이너는 구동에 필요한 최소한의 패키지로 구성되므로, 다양한 하드웨어를 지원하는 호스트 운영체제를 선택하는 것이 필수적이다.

오해 4. 컨테이너는 아무 곳으로나 이동할 수 있다?

컨테이너를 지원하는 환경이라면 자유롭게 옮겨서 사용할 수 있을 것으로 생각하는 사람이 많다. 하지만 이것은 반만 맞고 반은 틀린 것이다. 기본적인 전제조건은 도커를 지원하는 리눅스 커널 이상을 사용해야한다는 것이다. 그렇다면 우분투, 센트OS 등의 하위버전에서 패키징한 컨테이너 이미지를 최신 상위버전에 올리면 정상 작동할까? 그렇지 않을 수 있다. 커널 모듈과 시스템 라이브러리, 배포판별 설정에 따라 문제가 발생할 수 있다.

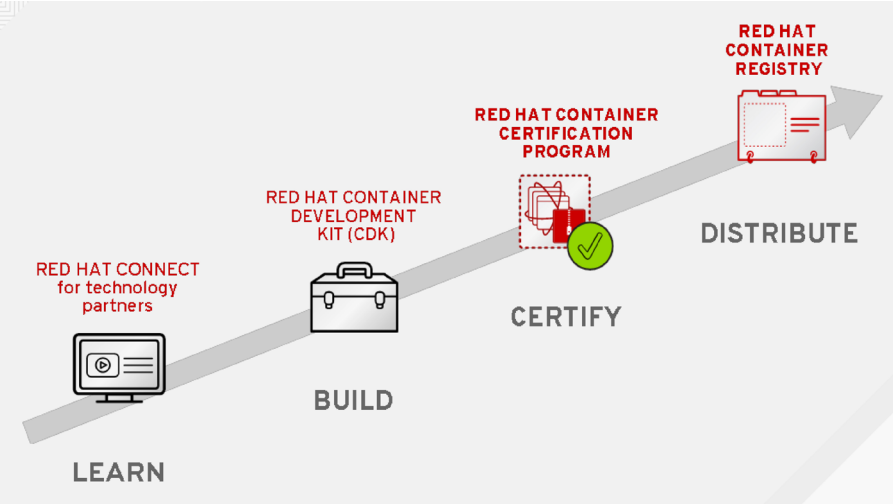
더 근본적으로는 이러한 호환성에 대한 검증과 레퍼런스가 아직 충분하지 않다. 처음엔 잘 실행되다가도 운영체제를 업데이트하는 과정에서 문제가 나타날 수도 있다. 그래서, 컨테이너용 리눅스는 이미지 업데이트 방식을 지원한다. 현재 컨테이너 기술 관련해 표준화된 방법론과 포맷 등을 정하기 위해 OCI(Open Container Initiative)재단이 만들어졌다. 지난 4월 13일 나온 도커 1.11버전에서는 OCI에서 정한 도커 컨테이너 포맷과 런타임을 runC 형태로 적용했다. 다만, 개발자는 패키징된 컨테이너가 실제 운영환경에서 잘 동작하는지 검증이 필요하다.

오해 5. 컨테이너는 기본적으로 보안이 잘 돼 있다?

컨테이너는 보안이 잘 돼 있다고 생각하는 사람이 많지만 실제로 그렇지 않다. 반얀옵스(BanyanOps)의 조사 결과를 보면 도커 허브에서 제공하는 공식 컨테이너 이미지 중 30% 이상에서 최상위 보안 취약점이 발견됐다. 많은 기업이 컨테이너 이미지를 도커 허브에서 손쉽게 갖다 쓰고 있는데, 이런 편리함이 보안 구멍을 될 수 있음이 명확해졌다.

보안이 취약한 것은 비단 도커 허브만이 아니다. 컨테이너 패키징 내에 들어가는 모듈 자체가 안전하지 않을 가능성도 있다. 예를 들어 인터넷에서 제이보스, 톰캣 등을 가져다 패키징하는 과정에서 애플리케이션, 라이브러리, 런타임 빌드 환경이 안전한지, 백도어가 없는지 등을 확인해야 하지만 많은 기업이 비

그림 3 | 레드햇의 컨테이너 보안



용과 인력 등의 이유로 이 과정을 생략한다.

호스트 운영체제의 보안은 다른 의미에서 위험에 노출돼 있다. 레드햇을 비롯한 리눅스 운영체제 대부분이 SE리눅스를 지원한다. 활성화하면 바로 사용할 수 있다. 그러나 상당수 기업이 SE리눅스를 쓰지 않는다. 프로세스와 프로토콜, 포트 번호, 사용자, 파일 등을 제어하는 정책을 만들어야 하기 때문이다. 최근엔 관리툴의 사용자 인터페이스(UI)에서도 이런 정책을 설정할 수 있게 지원하지만 이조차도 번거롭게 생각하는 관리자가 많다. 그래서, 컨테이너에서는 SE 리눅스를 기본 활성화하여 사용할 수 있도록 한다.

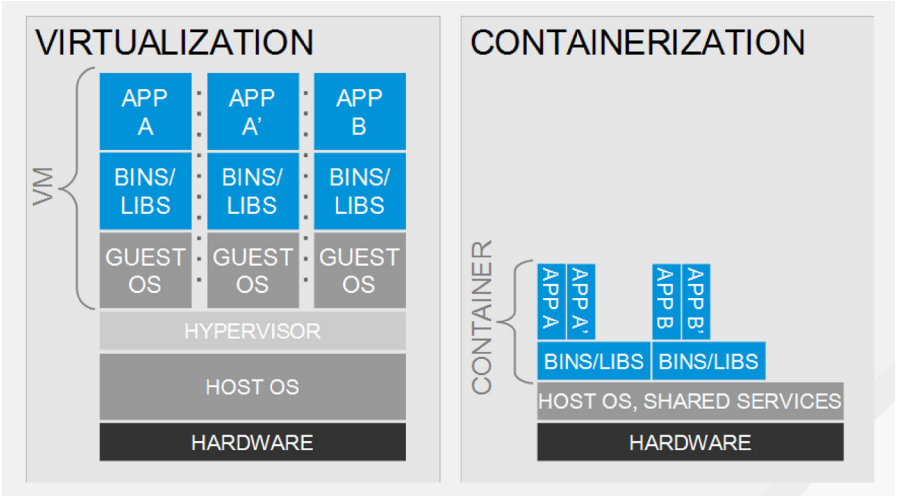
오해 6. 컨테이너는 가상화와 같다?

컨테이너는 가상화와 구조가 비슷하므로 같은 것으로 생각하거나 혹은 완전히 대체할 수 있다고 생각하는 실무자가 많다. 그러나 컨테이너는 몇 가지 점에서 가상화 대비 장점 혹은 단점을 갖고 있다. 가상화와 비교했을 때 컨테이너의 가장 큰 장점은 실시간적인 민첩성과 대응이 필요한 곳에 적용할 수 있는 확장성이다. 보통 가상화는 자원이 부족하면 해당 VM에 자원을 더 할당하는 '스케일 업(Scale Up)' 방식으로 확장한다. 반면 컨테이너는 같은 역할을 하는 컨테이너를 하나 더 만들어 실행하는 '스케일 아웃(Scale Out)' 방식이다.

스케일 아웃 방식이 중요한 것은 장애 대응이나 서비스 확장 시 더 유연하게 대응할 수 있기 때문이다. 컨테이너는 장애가 발생했을 때 기존 컨테이너를 살리는 대신 새로 하나를 만들어 실행하는 것이 더 빠를 만큼 가볍고 유연하다. 특히 스케일 아웃 방식의 인프라 확장에 최적화된 클라우드가 점점 기업 인프라의 주류로 자리 잡고 있는 것도 컨테이너의 장래를 밝게 보는 이유다.

컨테이너에도 가상화 대비 부족한 기능이 있다. 특히 네트워크와 스토리지 관련된 일부 기능은 추가 기술개발이 필요하다는 지적이 많다. 그러나 클라우드와 결합한 웹이나 소셜 환경에서 컨테이너는 매력적인 성능과 확장성을 제공한다. 구글을 비롯해 이베이와 그루폰, 우버, 바이두 등 대규모 온라인 서비스를

그림 4 | 가상화와 컨테이너 비교



제공하는 업체가 도커를 널리 사용하는 것도 이 때문이다.

6가지 오해에 대한 레드햇의 대답

레드햇은 2013년 도커가 발표된 직후부터 도커를 공식 지원해 왔다. 2015년 구글 쿠버네티스(Kubernetes)를 통합한 컨테이너 운영체제 '아토믹 호스트(Atomic Host)'를 내놓았고, 커맨드 라인 대신 UI 기반에서 운영할 수 있는 PaaS(Platform as a Service) 솔루션 '오픈시프트(OpenShift)'도 발표했다.

레드햇이 이처럼 발 빠르게 도커에 대응할 수 있었던 것은 도커와 같은 컨테이너 기술이 기본적으로 리눅스 가상화를 기반으로 하기 때문이다. 레드햇에 컨테이너 기술은 새롭지 않다. 오히려 많은 기업이 이를 손쉽게 받아들일 수 있도록 지원할 준비가 돼 있다. 실제로 레드햇은 오픈시프트의 2.X 아키텍처를 3.X 버전으로 올리면서 그 구조를 도커 기반의 엔진으로 완전히 바꿨다.

또한, 레드햇은 3,700개 이상의 하드웨어를 인증해 안정적으로 컨테이너를 지원한다. 호환 테스트를 거치지 않은 하드웨어를 기업 인프라에 사용하면 드라이버나 라이브러리 측면에서 문제가 발생할 수 있다. 레드햇은 방대한 인증 프로그램을 통해 기업을 위한 현실적인 대안을 제시한다.

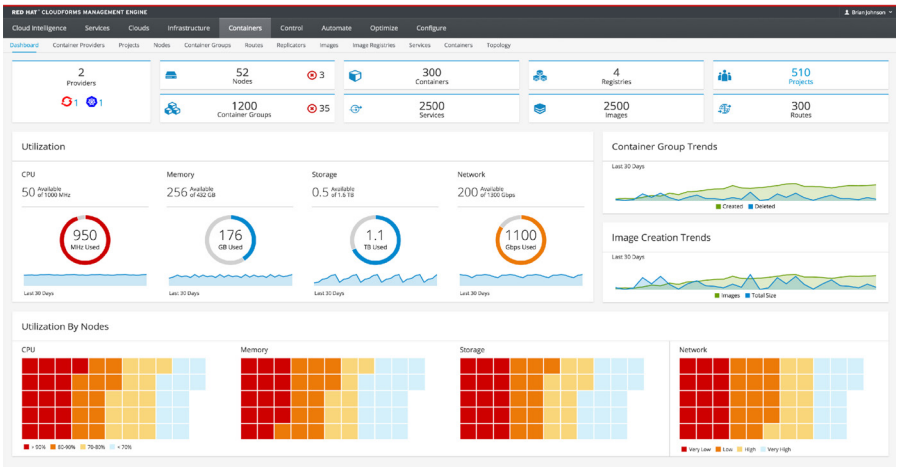
아직 충분히 검증되지 않은 컨테이너 호환성 관련해서, RHEL(Red Hat Enterprise Linux)과 아토믹 호스트는 훌륭한 대안이다. 이를 사용하면 컨테이너를 이동시켜 운영하는 환경에서 장애를 최소화하면서 관리할 수 있다. 특히 레드햇은 온프레미스(On-Premise) 환경뿐만 아니라 마이크로소프트, 구글, 아마존 등의 퍼블릭 클라우드까지 지원한다. 이들 환경에서 RHEL로 컨테이너를 구성하면 서로 연동해 확장할 수 있다.

레드햇의 컨테이너 보안 전략은 체계적이다. '레드햇 커넥트(Red Hat Connect)'를 통해 기술 파트너를 육성하고, '컨테이너 개발 키트(Container Development Kit: CDK)'는 개발 라이브러리와 툴의 보안을 검증해 배포한다. 또한, 컨테이너 인증 프로그램을 통해 컨테이너의 보안을 보증하고, 레드햇 컨테이너 레지스트리(Red Hat Container Registry)를 이용하면 보안이 확인된 컨테이너를 안심하고 내려받아 사용할 수 있다.

레드햇의 컨테이너 관련 제품 중 독보적 경쟁력을 가진 것이 관리 툴이다. 관리에 대한 실무자의 우려는 상당하다. 포레스터 리서치 조사결과를 보면 응답자의 41%는 개발툴, 프로세스 등과의 통합을 우려했고, 35%는 앱과 인프라스트럭처 관리의 어려움을 호소했다. 이 중 관리 업무는 특히 기업에 민감하다. 가상화 환경에서는 가상머신(VM)에 에이전트를 심거나 API(Application Programming Interface) 요청을 하는 형태로 모니터링할 수 있다. 그러나 컨테이너 관리는 아직 새로운 영역이어서 컨테이너 모니터링을 위해 더 많은 개발이 필요하다.

현재 컨테이너 관리툴 대부분이 커맨드 라인 환경에서 설치, 배포하지만, 레드햇의 '클라우드폼즈(CloudForms)'는 직관적이다. 하이브리드 클라우드를 지원하는 모니터링 툴로, 인프라부터 도커 컨테이너까지 관리, 모니터링할 수 있

그림 5 | 레드햇 클라우드폼즈



다. 특히 자원이 모자라면 퍼블릭 클라우드로 자동 확장해 인프라를 운영하도록 정책을 만들 수도 있다. 이런 방식의 컨테이너 관리 제품은 현재 클라우드폼즈가 거의 유일하다.

이제 우리도 컨테이너에 눈을 돌릴 때

컨테이너 기술은 빠르게 확산, 발전하고 있다. 특히 올해는 스토리지와 네트워크 관련 기능이 보강되는 것은 물론 관리에 대한 이슈가 본격적으로 제기되고 있다. 아직은 몇몇 요소를 더 보완해야 하지만 장기적으로는 가상화 이상으로 성장할 가능성도 있다.

컨테이너 기술에 관심이 있다면 현재 인프라에서 도입할 수 있는 여건이 되는지 확인하는 것으로 시작하는 것이 좋다. 컨테이너는 결국 애플리케이션을 잘 운영하는 기술이므로, 대상 애플리케이션의 구조가 컨테이너에 적합한지 봐야 한다. 가상화 경험이 있고 마이크로 서비스 아키텍처 형태를 검토하고 있다면 컨테이너가 최고의 대안이 될 것이다. 운영환경에 서둘러 적용할 필요는 없다. 충분히 검증한 후 도입하는 것이 안전하다. 운영환경을 검증할 때는 컨테이너 저장소(레지스트리)와 영구(persistent) 볼륨 등 컨테이너 기반의 배포나 할당이 가능한 아키텍처에 대한 검토가 필수적이다. 문화적인 접근도 중요하다. 컨테이너를 효과적으로 개발, 운영하려면 결국 개발 환경과 운영 환경을 일정 수준에서 표준화해야 한다. 이른바 '데브옵스' 트렌드와 같은 맥락이다. 때에 따라 기업의 IT 조직 문화 전반에 대한 재편이 필요할 수도 있다.

최근 들어 국내 기업도 업종과 규모를 가리지 않고 컨테이너에 관한 관심이 점차 커지고 있다. 특히 소셜 커머스나 포탈 업체 등은 이미 내부적으로 조직을 데브옵스로 재편하고 활발하게 컨테이너 기술을 검토하고 있다. 이들 기업의 서비스는 컨테이너의 장점을 살리기에 매우 유리하다는 것도 적극적인 행보에 나서는 이유다. 이들 선도업체를 중심으로 1~2년 이내에 의미있는 국내 도입 사례가 나온다면 컨테이너 기술이 기업 전반으로 확산하는 전환기가 찾아올 것으로 기대된다.