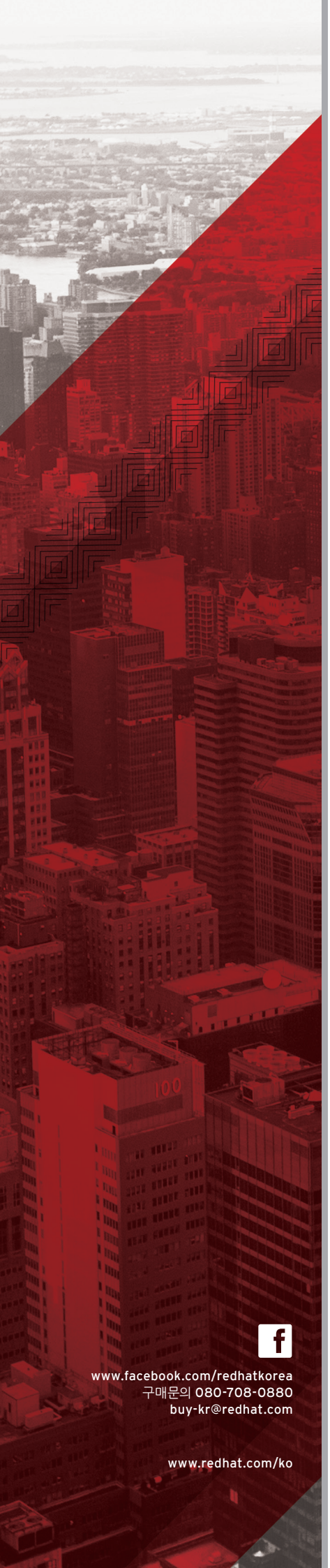

기술 세부 정보

RED HAT 컨테이너 이미지와 호스트 가이드

최적의 컨테이너 이미지와 호스트 선택

목차

www.facebook.com/redhatkorea
구매문의 080-708-0880
buy-kr@redhat.com

www.redhat.com/ko

2017년 8월

100 East Davie Street
Raleigh NC 27601 USA
Phone: +1 919 754 4950
Phone: 888 733 4281
Fax: +1 919 800-3804

소개

Linux® 컨테이너는 개발자의 랩탑에서 프로덕션 클러스터에 이르기까지 다양한 환경 전반에 손쉽게 애플리케이션을 배포하고 실행할 수 있도록 합니다. 실행하는 서버의 업그레이드 및 다운그레이드와 애플리케이션의 업그레이드 및 다운그레이드를 서로 분리(Decouple)함으로써 시스템 관리자와 개발자들은 각자 다른 속도로 작업을 진행할 수 있습니다. 컨테이너는 트랜잭션 방식의 업그레이드과 함께 모든 애플리케이션 디펜던시들을 제공합니다. 따라서 무엇인가 제대로 작동하지 않는다면, 단 몇 초 내에 롤백이 실행됩니다 (오전 2시부터 발생하는 8시간의 변경 실행 시간을 이제 걱정하지 않아도 됩니다).

컨테이너는 운영체제를 컨테이너 호스트와 컨테이너 이미지 등 2개의 별도 요소로 관리함으로써 이들 모든 기능들을 제공합니다. 컨테이너 호스트는 물리서버 혹은 가상머신 상의 리눅스 커널과 같은 운영체제에서 자원 관리 및 보안 기능을 제공합니다. 컨테이너 호스트에서 실행되는 컨테이너 이미지는 운영체제 라이브러리, 애플리케이션 실행환경, 그리고 애플리케이션을 포함합니다. IT 운영팀은 기본적으로 컨테이너 호스트를 관리하며, 부가적으로 컨테이너 이미지 생성, 변경에 대한 컨텐츠도 관리합니다.

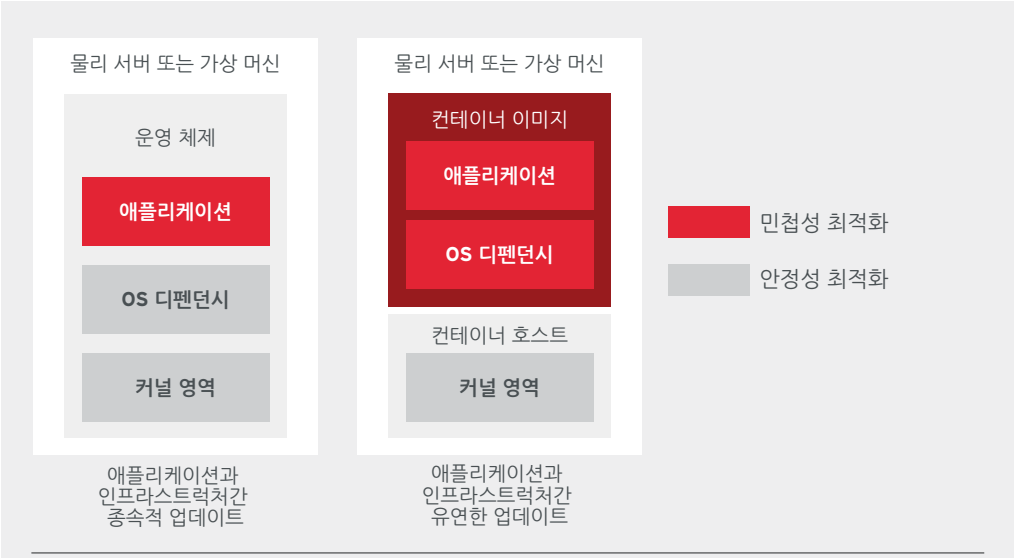

그림 1. 컨테이너 이미지 사용 전/후의 인프라와 애플리케이션

컨테이너는 애플리케이션 이식성을 제공합니다. 이는 팀이 코드를 배포하는 시간과 위치를 보다 효과적으로 제어할 수 있도록 합니다. 그러나 이식성과 호환성은 별개의 개념입니다. 개발 및 운영 팀은 서로 다른 속도로 작업을 수행하고 각각 독립적으로 서버 및 애플리케이션을 업그레이드 및 다운그레이드하며 여러 환경 간에 애플리케이션을 손쉽게 이동시킬 수 있습니다. 그러나 컨테이너 호스트와 컨테이너 이미지 간 성능, 보안 및 호환성 또한 못지않게 중요합니다. 이 문서는 귀사의 애플리케이션을 위해 적합한 컨테이너 이미지와 호스트를 선택하는 방법에 대한 기술적 지침을 제공할 것입니다.

컨테이너 이미지

기술

애플리케이션은 실행 환경 및 필수 라이브러리를 필요로 하며, 따라서 운영 환경에 배포될 최종 컨테이너 이미지가 일반적으로 여러 레이어로 이루어져 있다는 것을 이해하는 것이 중요합니다. 첫 번째 이미지는 베이스 이미지로 불리며 시작하는 데 필요한 모든 것을 포함하고 있습니다. 각 팀들은 베이스 이미지를 기반으로 필요한 레이어를 추가합니다. 사용자들은 운영 체제뿐만 아니라 실행 환경, 디버깅 툴 그리고 라이브러리를 포함하고 있는 보다 고도화된 애플리케이션 이미지를 구축할 수 있습니다.

베이스 이미지는 운영 체제 내에 포함되어 있는 것과 동일한 유틸리티와 라이브러리에서 구축됩니다. 올바른 베이스 이미지는 애플리케이션 구축을 위한 안전하고 안정적인 플랫폼을 제공합니다. Red Hat은 Red Hat® Enterprise Linux를 기반으로 베이스 이미지를 제공합니다. 이들 이미지는 일반 운영 체제와 같이 사용될 수 있습니다. 사용자들은 자신의 애플리케이션을 위해 필요에 따라 자유롭게 수정합니다. 예를 들어, 패키지를 설치하며 정상적인 Red Hat Enterprise Linux Server와 마찬가지로 서비스를 시작할 수 있습니다.

Red Hat은 또한 이미 다양한 소프트웨어가 포함된 보다 고도화된 애플리케이션 이미지를 제공합니다. 그 예로, 실행 환경, 데이터베이스, 미들웨어, 스토리지, ISV(Independent Software Vendor) 소프트웨어 등이 있습니다. 이들 이미지에 포함된 소프트웨어는 별도의 수정이 필요없이 실행할 수 있도록 테스트되고 사전 구성되었으며 검증되었습니다. 사전 구축된 컨테이너 이미지는 자체 애플리케이션 이미지를 작성하는 작업을 수행하지 않고도 기존 구성 요소에서 멀티티어(multitier) 애플리케이션을 손쉽게 배포할 수 있도록 합니다.

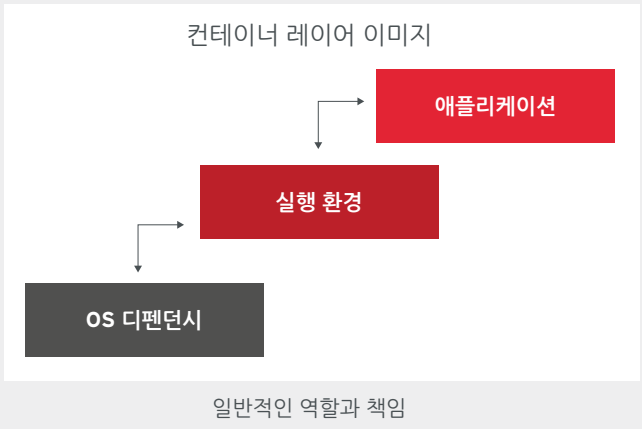

그림 2. 컨테이너 이미지 내 계층별 역할 및 책임

Linux 컨테이너는 종종 이식 가능하다고 설명되지만 이식성이 호환성을 보장하지는 않습니다. 이식성은 종종 Red Hat Enterprise Linux 컨테이너 이미지를 가져와 Linux 배포판으로 구축한 모든 컨테이너 호스트에서 이를 실행할 수 있다는 것으로 이해되고 있지만, 이는 기술적으로 정확한 설명은 아닙니다. Linux 컨테이너 이미지는 하드웨어 아키텍처와 운영 체제에 특정한 라이브러리와 바이너리 등을 포함한 파일의 컬렉션입니다. 컨테이너 이미지가 실행되면 이미지 내부의 바이너리는 정상적인 Linux 운영 체제에서 실행되는 것과 동일하게 실행됩니다. 컨테이너 이미지와 컨테이너 호스트 간에는 반드시 호환성이 확보되어야 합니다.

예를 들어, 32비트 호스트에서 64비트 바이너리를 실행할 수 없으며, x86_64 호스트에서 ARM 컨테이너를 실행할 수 없습니다. 동일한 운영 체제 규칙이 적용됩니다. 컨테이너는 가상화와 다릅니다. 이들은 모든 유형의 변환(Translation) 기능을 제공하지 않으며, 바이너리가 여러 플랫폼상에서 실행되도록 보장하지 않습니다.

그리고 때때로 하나의 x86_64 Linux 배포판 또는 다른 Linux 배포판 용으로 개발된 바이너리를 문제 없이 실행할 수 있으며, 다양한 Linux 배포판 간에 컨테이너 이미지를 시작하고 실행할 수도 있습니다. 사용자들은 컨테이너 이미지와 호스트의 여러 배포판을 함께 혼합해 사용하는 경우, Linux 컨테이너 이미지 내 바이너리로 인해 불일치 문제를 경험할 수 있습니다.

이들 불일치 문제는 디버깅하기 어려운 일시적인 장애에서 컨테이너가 전혀 실행되지 않는 오류에 이르기까지 다양합니다. 더욱 심각한 것은 비호환성이 성능 저하 또는 보안 이슈로 이어질 수 있다는 것입니다. Linux 컨테이너의 이식성은 완전하지 않습니다. 따라서 여러 다른 Linux 배포판을 사용하거나 동일한 Linux 배포판 중 크게 다른 버전을 사용하는 경우, 심각한 문제로 이어질 수 있습니다.

따라서 수평적이고 수직적인 호환성을 모두 보장하기 위해서 특정 클러스터 내 모든 컨테이너와 컨테이너 호스트에서 Linux의 일관된 버전과 배포판을 실행하는 것이 바람직합니다.

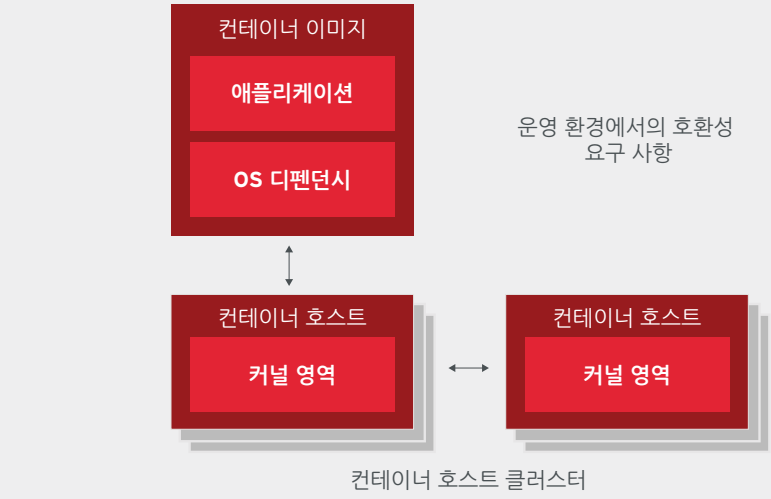

그림 3. 컨테이너 호스트 클러스터 호환성 요구 사항

컨테이너 이미지 내 애플리케이션과 해당 의존 라이브러리는 종종 여러 Linux 배포판에서 실행될 때 노출될 수 있는 커널 기능들과 컨테이너 엔진에 대한 기대치를 가지고 있습니다. 클러스터 내에 Linux 배포판, 버전 또는 구성을 혼합할 때 발생할 수 있는 대표적인 문제들은 다음과 같습니다.

- SELinux(Security-Enhanced Linux)를 지원하지 않는 호스트상에서 SELinux 지원을 기대하며 컴파일된 컨테이너 내 실행 바이너리.[1]
- 컨테이너 엔진과 컨테이너 호스트상의 특정 커널 기능(CAP_PTRACE 등)을 기대하지만, 해당 실행을 거부하도록 구성된 실행 컨테이너 이미지.[2]
- 특정 입력/출력 제어(ioctl) 호출 또는 /proc 및 /sys의 특정 레이아웃을 요구하는 실행 바이너리. 이는 기본 컨테이너 호스트 커널의 버전과 호환되지 않으며, 버전에 따라 달라집니다. 모든 해당 인터페이스는 커널, 사용자 영역 툴 또는 라이브러리 버전을 통해 변경될 수 있습니다.

1 *"Docker, SELinux 및 커널 독립성에 대한 오해 | Fewbytes." 2014년 11월 21일, http://www.fewbytes.com/docker-selinux-and-the-myth-of-kernel-indipendence/. 2017년 3월 22일 액세스됨.*

2 *"Docker 호스트로서의 CentOS로 인해 다른 컨테이너 실행 발생 ...." 2016년 7월7일, http://www.centos.org/forums/viewtopic.php?t=58409. 2017년 3월 22일 액세스됨.*

- 컨테이너 엔진을 지원하는 기본 파일시스템 내 비호환성. 이는 디버깅하기 매우 어려운 문제로 이어질 수 있습니다. 예를 들어, 기본 파일 시스템이 POSIX 또는 확장 속성을 지원하지 않는 경우, 임의의 장애가 애플리케이션 컨테이너에서 발생할 수 있습니다.[3] 클러스터 내에서 Overlay2, 디바이스 매퍼 또는 파일시스템을 혼합하지 마십시오.

- 컨테이너 호스트와 컨테이너 이미지 버전의 불일치. 사용자 영역과 커널 간의 버전 불일치가 클수록 호환되지 않을 가능성이 높아집니다.

  - 최신 바이너리 또는 라이브러리, 최신 버전이 아닌 커널: 컨테이너 호스트에서 실행할 수 없는 최신 커널 기능에 의존하는 실행 바이너리 또는 라이브러리는 문제를 초래할 수 있습니다. 이는 개발 환경에서 최신 커널을 실행하고 프로덕션 환경에서 구형 커널을 사용할 때 발생할 수 있습니다. 이와 같은 불일치가 발생하면 glibc은 최소 커널 버전을 확인할 것입니다. 최소 버전이 충족되지 않는 경우, 프로그램은 종료될 것입니다.[4]

  - 최신 버전이 아닌 바이너리 또는 라이브러리, 최신 커널: 테스트 또는 개발 중 사용된 것보다 최신 커널과 함께 컨테이너 호스트에서 실행 중인 바이너리 또는 라이브러리는 실행의 차이를 초래할 수 있습니다. 이는 특히 glibc이 새로운 런타임 감지 기능을 활성화할 때 분명하게 나타납니다. 늘 테스트한 결과에 기반하여 릴리즈해야 합니다.

- 하드웨어, 커널 및 라이브러리의 차이. 예를 들어, glibc은 특정 버전의 하드웨어에 최적화되어 있습니다. 일부 가속화된 루틴들이 하드웨어 가용성에 따라 선택됩니다. 이는 커널 감지 및 glibc 감지의 조합입니다. 노드 간의 불일치는 애플리케이션에서 실행상의 변화를 야기할 수 있습니다. 예를 들어 루틴은 하드웨어 가속에서 소프트웨어 단독으로 변경되거나 그 반대가 될 수 있습니다. 이는 스케줄링된 노드에 따라 예기치 않게 애플리케이션 속도를 늦추거나 높일 수 있습니다. 애플리케이션이 단일 서비스 또는 API(Application Programming Interface)를 지원하는 여러 컨테이너로 구성되어 있는 경우, 모든 요청이 동일하게 실행되지 않을 수 있습니다.

- 환경을 관리하는 컨테이너 실행. 이는 Kubernetes 마스터, Red Hat OpenShift Container Platform 마스터, docker 데몬 또는 Linux 커널과 직접 상호 작용해야 하는 분산 시스템 관리 작업에서 일반적입니다. 환경 관리는 SPC(super-privileged container)의 일반적인 활용 사례로서, 오케스트레이션 레이어에서 커널까지 모든 API 간의 완벽한 호환성을 요구할 수 있습니다. 또한, 이들 작업은 클러스터에 서비스를 복원하는 데 매우 중요할 수 있으므로 어떤 개별 애플리케이션보다도 더 큰 위험을 수반합니다.

컨테이너의 완벽한 이식성은 컨테이너 호스트 또는 컨테이너화된 프로세스가 실패할 때 매우 중요합니다. 실패한 컨테이너는 다른 호스트에서 신속하게 재시작되어야 합니다. Linux의 여러 다른 버전과 배포판을 기반으로 하는 컨테이너 호스트와 컨테이너 이미지를 혼합하는 경우, 호환성 문제로 인해 컨테이너 재시작 시 실패할 위험성이 높아지게 됩니다. 운영 환경이 중단된 상태에서 비호환성 문제를 발견하기가 가장 어렵습니다.

운영 환경 수준의 호환성 표준을 만족시키려면 다음의 3가지 구체적인 요구 사항이 충족되어야 합니다.

1. 클러스터 내 모든 컨테이너 이미지는 동일한 베이스 이미지를 기반으로 작성되어야 합니다. 컨테이너 이미지 내 소프트웨어 스택 및 라이브러리 (사용자 영역) 버전은 컨테이너 호스트 커널과 호환되어야 합니다.

---

3 "Bug 1268347 - posix_fallocate emulation on NFS ... - Red Hat Bugzilla." 2015년 10월 2일, https://bugzilla.redhat.com/show_bug.cgi?id=1268347. 2017년 6월 22일 액세스됨.

4 "바이너리용 glibc의 최소 버전 받기 - Alan's Web Ramblings." 2015년 12월 12일, http://agardner.me/golang/cgo/c/dependencies/glibc/kernel/linux/2015/12/12/c-dependencies.html. 2017년 3월 22일 액세스됨.

2. 클러스터 내 모든 컨테이너 호스트는 호환 하드웨어, 커널 및 라이브러리가 있어야 하며, 이들 모두가 일치되는 것이 바람직합니다. Red Hat Enterprise Linux, Red Hat Enterprise Linux Atomic Host 및 모든 마이너 버전들은 기본값으로 API/ABI 호환성 지침과 이러한 수준의 호환성을 제공합니다.[5]

3. 모든 컨테이너 호스트는 동일하게 구성되어야 하며 특히 네트워크 및 스토리지가 동일해야 합니다. 공유 스토리지(NFS, Gluster, iSCSI 등), 로컬 스토리지(overlay2 또는 devicemapper) 타임 서버, 컨테이너 런타임(docker), 커널 매개변수, SELinux, seccomp 등에 대한 모든 구성과 액세스는 동일해야 합니다. Red Hat Enterprise Linux Atomic Host의 단순한 구성은 이와 같은 우려 중 상당 부분을 없앨 수 있습니다.

이들 모든 요구 사항들이 충족될 때에만 최고 수준의 이식성이 보장됩니다. Red Hat은 오직 Red Hat이 제공한 컨테이너 이미지 및 컨테이너 호스트만 운영 환경 수준의 호환성 요구 사항을 유지하도록 함께 테스트하고 있습니다. 이는 Red Hat의 API/ABI (Application Binary Interface) 보증을 통해 단순화되며, 클라우드 환경에서 그 중요성이 더욱 높아지고 있습니다.[5]

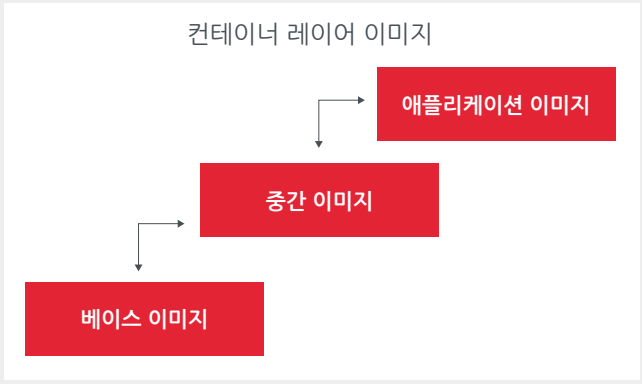

그림 4. Red Hat은 애플리케이션 컨테이너의 모든 계층을 위해 이용할 수 있는 인증된 컨테이너 이미지를 제공합니다.

주요 특징 및 기능

Red Hat은 애플리케이션 개발 및 배포를 위한 다양한 이미지를 제공합니다.

Red Hat이 개발한 *베이스 이미지(base image)*는 애플리케이션을 위한 안전하고 안정적인 토대입니다.

- **Red Hat Enterprise Linux 표준 이미지.** 모든 애플리케이션을 위해 사용될 수 있는 모든 기능을 갖춘 베이스 이미지. yum, systemd, 그리고 Red Hat Enterprise Linux에 포함된 모든 패키지에 대한 액세스 권한이 포함되어 있으며, 정상적인 Red Hat Enterprise Linux의 주요 릴리스 또는 업데이트 릴리스 라이프사이클을 따릅니다. Red Hat은 Red Hat Enterprise Linux 6 및 7 기반 버전에 대한 철저한 테스트를 실시했습니다.
- **Red Hat Enterprise Linux atomic 이미지.** 빠르게 변화하는 개발 환경을 목표로 하는 작고 간결한 베이스 이미지. 이미지 크기를 줄이기 위해 microdnf를 사용하며 systemd를 포함하고 있지 않습니다. 항상 Red Hat Enterprise Linux 7의 최신 버전을 기반으로 하며 최신 마이너 릴리스만 지원합니다.

5 "Red Hat Enterprise Linux 7: 애플리케이션 호환성 가이드 - Red Hat Customer Portal" 2017년 6월 12일, https://access.redhat.com/articles/rhel-abi-compatibility. 2017년 6월 22일 액세스됨.

Red Hat이 설계 및 개발한 *중간 이미지(intermediate image)*는 미리 구성된, 애플리케이션 구축을 위한 출발점을 제공하기 때문에, 인프라가 아니라 애플리케이션에 집중할 수 있습니다.

- Red Hat 소프트웨어 컬렉션은 node.js, perl, php, python, ruby, rails 등을 포함한 실행 환경용 이미지를 제공합니다.
- .Net Core 컨테이너 이미지는 업계 유일하게 상업적으로 지원되는 Linux상의 .Net Core 배포판에 신속하게 액세스할 수 있도록 합니다.

Red Hat이 설계 및 개발한 *애플리케이션 이미지(Application image)*는 데이터베이스, 분산 캐시, ISV 이미지 등과 같이 사전 구축된 구성 요소에 액세스할 수 있도록 함으로써 애플리케이션을 신속하게 시작할 수 있도록 지원합니다.

- Red Hat 소프트웨어 컬렉션 이미지는 Apache HTTPD, MariaDB, MySQL, Nginx, Passenger, PostgreSQL, Redis 등에 신속하고 손쉽게 액세스할 수 있도록 합니다.
- 또한, Red Hat은 NoSQL 데이터베이스 및 메시징 플랫폼에서 로드 밸런서와 보안 스캐닝 툴에 이르는 파트너 ISV들의 컨테이너 이미지도 제공합니다.

모든 Red Hat 컨테이너 이미지는 Red Hat Enterprise Linux에서 구축됩니다. 사용자에게 최고의 경험을 제공하기 위해 Red Hat Enterprise Linux 베이스 이미지(표준 및 atomic)는 이들 표준을 준수합니다.

아키텍처 및 빌드 프로세스

- 안정성과 호환성을 보장하기 위해 Red Hat Enterprise Linux와 동일한 엔지니어링, 품질 보증 및 테스트를 거칩니다.
- glibc, libseccomp, libselinux 등과 같은 코어, 커널 기반 라이브러리와 함께 제공됩니다. 이들 라이브러리는 Red Hat Enterprise Linux 커널과의 긴밀한 통합 주기를 따르며 컨테이너 이미지 및 컨테이너 호스트의 라이프사이클 전반에 걸쳐 호환성을 보장합니다.
- System Tap, GDB, ABRT, sosreport, tcpdump, kdump 등의 툴을 긴밀히 통합합니다.[6] 이와 같은 통합을 통해 운영 팀은 프로덕션 클러스터를 디버깅하기 위해 이용하는 모든 툴에 액세스할 수 있으며 Red Hat은 호스트 운영 체제의 커널과 컨테이너 간에 발생할 수 있는 문제를 재현하고 복구하도록 지원할 수 있습니다.

보안

- 2016년 1,346개 취약점을 트랙킹하고 해결한 제품 보안 팀이 보호합니다.[7] 트랙킹 데이터(Errata 및 OVAL)는 모든 Red Hat 이미지 내 RPM을 위해 생성됩니다.
- 프로덕션 컨테이너 클러스터에서 보안 규정이 준수되도록 보장하는 동일한 보안 강화와 에러터(errata) 프로세스를 따릅니다.

성능 및 테스트

- Red Hat OpenShift Online[8] 및 Red Hat OpenShift Dedicated[9] 환경에서 큰 규모의 프로덕션 단계에서 사용됩니다. 이에 따라 이들 이미지는 실전을 거쳐 더욱 강화되었으며 대규모 멀티테넌트 Kubernetes 환경에서 큰 규모로 테스트되었습니다.
- 성능, 안정성 및 보안을 보장하는 GCC 및 glibc를 이용해 바이너리를 개발했습니다. Red Hat은 이들 모두의 개발 및 유지 보수에 크게 기여하고 있습니다.

---

6 *"[atomic-devel] 충돌 처리 방법 - 프로젝트 아토믹(Project Atomic)." https://lists.projectatomic.io/projectatomic-archives/atomic-devel/2015-February/msg00026.html. 2017년 3월 23일 액세스됨.*

7 *"2016년 Red Hat 제품 보안 위험 보고서 - Red Hat 고객 포털." 2017년 3월 7일, https://access.redhat.com/blogs/766093/posts/2957221. 2017년 7월 25일 액세스됨.*

8 *"OpenShift Online (차세대) 개발자 프리뷰" https://www.openshift.com/devpreview/. 2017년 3월 23일 액세스됨.*

9 *"Red Hat OpenShift Dedicated에 대한 자세한 내용" https://www.openshift.com/dedicated/. 2017년 3월 23일 액세스됨.*

- GCC에 대한 액세스를 포함하고 있습니다. 332개의 최적화 패스(optimization pass)와 33만 번의 테스트를 통해 Red Hat은 지속적으로 코드에 기여하며 테스트 범위를 확장하고 있습니다. 테스트에서는 구문(Syntax)을 검증하고 최적화 패스를 확인하며 디버거를 통해 실행하고 진행할 수 있는 컴파일된 코드를 검사합니다.[10]
- 매우 안정적이고 충분히 테스트된 C 라이브러리인 glibc에 대응하여 연결되어 있습니다.
- 네트워크 유틸리티 및 서버가 컨테이너 내에서 예상대로 실행되도록 보장하는 glibc DNS 리졸버(resolver)를 사용합니다.
- 다음과 같은 컨테이너 이미지와 호스트 커널 간에 호환성을 보장하도록 Red Hat Enterprise Linux 커널과 관련하여 구축하고 테스트했습니다.
  - syscalls
  - IOCTL
  - /proc 및 /sys (슈퍼 사용자 권한을 보유한(super-privileged) 컨테이너)
  - glibc 및 기본 호스트 커널(C 코드)

지원 및 인증

- 기본 Red Hat Enterprise Linux 컨테이너 호스트 서브스크립션(스탠다드, 프리미엄 등)과 동일한 수준의 지원을 받습니다.[11]
- 컨테이너용 서브스크립션은 매우 투명하고 편리하게 관리됩니다. 등록된 컨테이너 호스트에서 실행 중인 컨테이너로 권한이 자동으로 부여되며, 이는 Red Hat 호스트상에서 Red Hat Enterprise Linux 컨테이너를 편리하고 손쉽게 구축해 실행할 수 있도록 합니다. 서브스크립션 인증을 통해 사용자들은 Red Hat Satellite 또는 Red Hat 고객 포털에서 직접 Red Hat Enterprise Linux 컨테이너 내부의 패키지를 손쉽게 설치하고 업데이트할 수 있습니다. Red Hat 서브스크립션 관리를 사용하지 않거나 Red Hat이 아닌 호스트에서 실행되는 Red Hat 컨테이너로 서브스크립션 키를 수동으로 전달하는 방식은 지원되지 않습니다.

---

10 "Testing… Testing… GCC - RHD 블로그 - Red Hat Developers." 2017년 2월 13일, https://developers.redhat.com/blog/2017/02/13/testing-testing-gcc/. 2017년 3월 23일 액세스됨.

11 "Red Hat 컨테이너 지원 정책 - Red Hat 고객 포털" 2017년 2월 23일, https://access.redhat.com/articles/2726611. 2017년 3월 23일 액세스됨.

컨테이너 호스트

기술

애플리케이션은 필수 실행 환경과 라이브러리를 제공하는 컨테이너 이미지에서 구축됩니다. 애플리케이션 컨테이너 이미지는 컨테이너 호스트에서 실행하면 애플리케이션 컨테이너로 전환됩니다. 실행 컨테이너는 실제로 SELinux, cgroups, namespace 등과 같은 커널 기술들을 이용하는 격리된 프로세스입니다. 이러한 커널 기술들은 보안 계층과 성능 격리 기능을 제공하지만, 가상화와 동일한 수준의 분리를 제공하지는 않습니다. 이는 호스트 커널의 버전과 품질이 컨테이너화된 프로세스의 보안과 성능을 결정한다는 것을 의미합니다. 또한, 하드웨어의 물리적 아키텍처는 어떤 컨테이너가 호스트(x86, ARM, POWER 등)상에서 실행될 수 있는지를 결정합니다.

컨테이너 호스트의 가치는 컨테이너를 실행하는 데 필요한 모든 구성 요소와 함께 구축, 테스트, 검증되었다는 데 있습니다. 컨테이너 호스트는 애플리케이션 컨테이너의 보안, 성능 및 지원성에 기여하는 주요 구성 요소입니다. 컨테이너 호스트의 지원 정책 및 라이프사이클은 시간 경과에 따라 환경을 유지하고 업그레이드를 수행하는 데 필요한 시간과 비용에 지대한 영향을 미칩니다.

컨테이너 호스트는 테스트되고 검증된 Linux 커널, 호환 컨테이너 엔진, 그리고 운영 환경에서 실행, 문제 해결(Troubleshoot), 진단 및 이슈 해결 등을 수행하는 데 필요한 모든 라이브러리와 툴을 포함하고 있습니다. 즉, 엔드투엔드(End-to-end) 지원이 이루어집니다. 각 환경을 위한 최적의 컨테이너 호스트를 선택할 때 반드시 이들 모든 요소들을 고려해야 합니다.

주요 특징 및 기능

Red Hat은 Red Hat Enterprise Linux Server와 Red Hat Enterprise Linux Atomic Host 등 2개의 컨테이너 호스트를 제공합니다. 이들 두 제품 모두 Red Hat Enterprise Linux를 기반으로 컨테이너 이미지를 구축하고 테스트하며 그 실행을 검증합니다. 두 컨테이너 호스트 모두 스토리지, 네트워크 하드웨어, 소프트웨어, 하이퍼바이저 및 클라우드 제공업체들로 구성된 대규모 에코시스템을 통해 테스트 및 검증되었습니다. 외견상 이들 두 제품은 매우 비슷해 보이지만 관리되는 방법은 분명하게 다릅니다.

Red Hat Enterprise Linux Server는 모든 애플리케이션들을 위한 범용 운영 체제로 설계되었으며 일부 고객들은 운영 환경에서 수천 개의 인스턴스를 실행하고 있습니다. Red Hat Enterprise Linux Server는 해당 워크로드를 지원하는 데 필요한 각 인스턴스를 구성할 수 있는 유연성을 제공합니다. 이와 같은 유연성은 그에 상응하는 노력을 요합니다. 매우 다양한 패키지 조합이 가능하기 때문에 운영을 위한 관리방법이 다르며 보다 전통적인 툴로 처리됩니다.

Red Hat Enterprise Linux Atomic Host는 여러 다양한 배포, 운영 및 관리 패러다임을 위해 설계되어 대규모 환경에서도 쉽게 관리할 수 있습니다. 모든 Red Hat Enterprise Linux Atomic Hosts는 배포 시에 구성되고 변경 불가능한 호스트와 동일하게 관리되도록 설계되었습니다. 이는 자동화 기능을 통해 쉽게 구성되도록 설계되었으며 프라이빗 클라우드, 퍼블릭 클라우드 또는 개발자 랩탑의 가상 머신 등에 이르기까지 매우 일회적인 환경에 유용합니다.

많은 환경에서 운영 조직들은 필요한 워크로드를 위해 이상적인 패키지 세트로 이루어진 표준 운영 환경(일명 코어 빌드 또는 골든 이미지)를 생성합니다. Red Hat Enterprise Linux Atomic Host는 컨테이너 호스트를 위해 최적화된 독자적인 패키지 세트를 제공함으로써 코어 빌드의 개념을 확장한 것입니다. 이에 따라 Red Hat Enterprise Linux Atomic Host는 클라우드 환경 내에 구축하기 위한 경량화된 옵션으로 자리매김하고 있습니다.

Red Hat Enterprise Linux Atomic Host는 rpm-ostree로 불리는 특수한 기술을 이용해 업데이트됩니다. 이에 따라 Red Hat Enterprise Linux Atomic Host에 대한 업데이트 방법이 Red Hat Enterprise Linux Server에 대한 업데이트와는 매우 달라지게 되며 전체 호스트는 선언적인 단일 atomic 트랜잭션 내에서 업데이트됩니다. atomic 모델은 업데이트해야 하는 특정 패키지를 계산하는 것이 아니라 모든 규모로

일관성을 제공하고 필요에 따라 롤백을 실행할 수 있도록 합니다. 변경 불가능한 ostree 계층들과 트랜잭션 방식의 업데이트를 통해 Red Hat Enterprise Linux Atomic Host는 모든 개별 호스트에 대한 제어가 제한된 대규모 분산 시스템 환경에서 관리될 수 있습니다. 이는 개발자의 랩탑에서 하나의 인스턴스가 실행되는지, 프로덕션 환경에서 수천 여 대 호스트를 보유하고 있는 지와 관계 없이 배포를 단순화합니다.[12]

신뢰할 수 있는 스토리지 구성 요소는 퍼시스턴트 데이터에 대한 액세스를 필요로 하는 애플리케이션 컨테이너에 특히 중요합니다. 통합, 테스트 그리고 컨테이너 오케스트레이션(Kubernetes)에서 컨테이너 호스트 커널(Linux)에 이르기까지 전체 소프트웨어 스택을 지원할 수 있는 역량은 데이터 무결성과 보안을 보장하는 데 필수적입니다.

Red Hat은 자체 스토리지 스택의 각 계층에 많은 핵심 개발자들을 투입하고 있습니다. 또한, 업스트림 Kubernetes, Gluster 및 Linux 커널 커뮤니티(그리고 기타 다수)에 직접 기여하고 있는 것은 물론, 컨테이너 호스트의 설계, 구축, 그리고 유지 관리 단계에서 회사내의 전문가들을 활용하고 있습니다.

Red Hat은 매우 신중하게 스토리지 구성 요소들을 검증하고 있습니다. 각 계층은 튜닝 및 보안을 유지하며 신중하게 컨테이너 플랫폼(Kubernetes Persistent Volumes) 내 데이터 구조에서 컨테이너 호스트 내 지원 파일 시스템(XFS, Overlay2 등) 및 블록 스토리지 구성 요소(devicemapper, iscsi, 파이버(fiber) 채널)에 이르기까지 고객 데이터의 무결성을 보호합니다.

아키텍처 및 빌드 프로세스

사용자에게 우수한 경험과 완전한 지원을 제공하기 위해,[13] Red Hat Enterprise Linux Server 및 Red Hat Enterprise Linux Atomic Host는

- 동일한 RPM 컨텐츠에서 개발되었으며 모니터링, 관리 에이전트 및 지원 툴로 이루어진 대규모 에코시스템을 형성하고 있습니다.
- glibc, libseccomp, libselinux 등과 같은 코어, 커널 기반 라이브러리와 함께 제공됩니다. 이러한 라이브러리는 Red Hat Enterprise Linux 컨테이너 이미지와의 긴밀한 통합 주기를 따르며 컨테이너 이미지 및 컨테이너 호스트의 독립적인 라이프사이클 전반에 걸쳐 호환성을 보장합니다.
- System Tap, GDB, ABRT, sosreport, tcpdump, kdump 등과 같은 툴과 긴밀하게 통합됩니다.[14] 이는 운영 팀이 프로덕션 클러스터를 디버깅하는 데 사용하는 모든 툴에 액세스할 수 있도록 보장합니다. 이를 통해 Red Hat은 컨테이너 실행 시 발생할 수 있는 문제를 재현하고 복구할 수 있도록 지원할 수 있습니다. 여기에는 호스트 운영 체제의 커널과 컨테이너 간의 호환성 문제가 포함됩니다.

보안

- SELinux, cgroups 및 namespaces를 이용해 컨테이너화된 프로세스 간에 커널 기반 격리를 실행합니다.
- 2016년 1,346개 취약점을 트랙킹하고 해결한 제품 보안 팀이 보호합니다.[15] 트랙킹 데이터(Errata 및 OVAL)는 Red Hat Enterprise Linux와 Red Hat Enterprise Lunix Atomic Host 내 RPM을 위해 생성됩니다.
- 프로덕션 컨테이너 클러스터에서 보안 규정 준수를 쉽게 검증할 수 있도록 보장하는 보안 강화와 에러터(errata) 프로세스를 따릅니다.

---

12 "RHEL Server 및 RHEL Atomic Host 간의 차이점" 2017년 5월 31일, https://access.redhat.com/articles/2772861.

13 "Red Hat 컨테이너 지원 정책 - Red Hat 고객 포털" 2017년6월8일, https://access.redhat.com/articles/2726611. 2017년 7월 25일 액세스됨.

14 "Atomic Host상에서 kexec/kdump을 구성하는 방법 - 고객 포털"2016년 12월5일, https://access.redhat.com/solutions/2792901. 2017년 3월 20일 액세스됨.

15 "2016년 Red Hat 제품 보안 위험 보고서 - Red Hat 고객 포털." 2017년 3월 7일, https://access.redhat.com/blogs/766093/posts/2957221. 2017년 7월 25일 액세스됨.

성능 및 테스트

- 사용자 공간과 커널 간에 동일한 엔지니어링, 품질 보증 및 테스트를 따릅니다.
- 성능, 안정성 및 보안을 보장하기 위해 GCC 및 glibc의 호환 버전을 이용해 구축된 바이너리들이 포함되어 있습니다. Red Hat은 개발과 유지 보수 모두에 크게 기여하고 있습니다.
- Red Hat OpenShift Online[16] 및 Red Hat OpenShift Dedicated[17] 환경에서 큰 규모의 프로덕션 단계에 사용됩니다. 이에 따라 이들 컨테이너 호스트는 더욱 강화되었으며 대규모 멀티테넌트 Kubernetes 환경에서 모든 규모로 테스트되었습니다.

스토리지

- 지원 가능한 환경을 보장하기 위해 특화된 스토리지 테스트 및 최적화 기능을 제공합니다.
  - Red Hat OpenShift Container Platform에서 PV(퍼시스턴트 볼륨)으로 테스트되었습니다.
  - devicemapper 및 Overlay2 그래프 드라이버(OverlayFS는 XFS의 사용 필수)를 지원합니다.
  - dockerpool을 관리하고 귀사의 환경을 위한 그래프 드라이버를 선택하는 데 편리한 툴을 제공합니다.
  - XFS 오류 처리 제어 강화. 씬 풀(thin pool)이 완전히 채워지면, 컨테이너가 정상적으로 정지될 수 있도록 보장합니다.
  - Red Hat Enterprise Linux Atomic Host는 최적의 성능을 위해 lvm 씬 풀에 의해 지원되는 devicemapper로 기본 설정되어 있습니다.
  - Red Hat Enterprise Linux는 최신 버전의 OverlayFS를 포함하고 있으며 이전의 몇몇 OverlayFS 결함들이 다음과 같은 배포판에 포함되지 않도록 보장합니다.
    - Overlay whiteouts이 가시화되지 않도록 방지
    - unix 도메인 소켓 생성 이슈에 대한 하드링크(hardlink) 수정
    - OverlayFS에 대한 SELinux 지원 예정

지원 및 인증

- 기본 Red Hat Enterprise Linux 서브스크립션 (standard, premium 등)과 동일한 수준의 지원을 받습니다.
- 베어 메탈 서버, 하이퍼바이저, 프라이빗 클라우드 플랫폼(OpenStack®), 그리고 퍼블릭 클라우드 제공업체 등으로 이루어진 동일한 에코시스템과의 호환성 및 안정성을 보장하는 동일한 지원 매트릭스를 따릅니다.
- 컨테이너용 서브스크립션은 매우 투명하고 편리하게 관리됩니다. 등록된 컨테이너 호스트에서 실행 중인 컨테이너로 권한이 자동으로 부여됩니다. 이는 Red Hat 호스트상에서 Red Hat Enterprise Linux 컨테이너를 편리하고 손쉽게 구축해 실행할 수 있도록 합니다. 서브스크립션을 통해 사용자들은 Red Hat Satellite 또는 Red Hat 고객 포털에서 직접 Red Hat Enterprise Linux 컨테이너 내부의 패키지를 손쉽게 설치하고 업데이트할 수 있습니다. Red Hat 서브스크립션 관리를 사용하지 않거나 Red Hat이 아닌 호스트에서 실행되는 Red Hat 컨테이너로 서브스크립션 키를 수동으로 전달하는 방식은 지원되지 않습니다.

---

16 "OpenShift Online 개발자 프리뷰" https://www.openshift.com/devpreview/index.html.

17 OpenShift Dedicated: https://www.openshift.com/dedicated/index.html

결론

컨테이너는 가상화와 유사한 형태의 격리와 구획화(compartmentalization)를 실행하지만 기본 운영 체제에 대한 의존성과 지원 프로필은 전통적인 사용자 애플리케이션의 형태와도 유사합니다. 이에 따라, Red Hat은 기본 컨테이너 호스트를 위한 모든 일상적인 지원과 인증 요구 사항을 토대로 애플리케이션으로서 Red Hat 컨테이너 이미지를 지원하고 있습니다. 이는 Red Hat 고객들을 보호하고 Red Hat 제품에 대해 기대하는 것과 동일한 수준의 신뢰성, 책임성 및 지원을 누릴 수 있도록 보장합니다.

Red Hat 컨테이너 이미지

현재 Red Hat Enterprise Linux 서브스크립션을 보유하고 있는 고객들은 인증되고 지원되는 Linux 컨테이너 이미지에 액세스할 수 있습니다. 모든 공식 Red Hat 이미지는 다음과 같은 장점을 포함하고 있는 Red Hat Container Catalog를 통해 제공됩니다.

- 공식 컨테이너 이미지의 신뢰할 수 있는 출처
- 보안된 액세스
- 아키텍트가 어떤 리포지토리, 이미지 및 태그를 사용할 것인지를 신속하게 결정할 수 있도록 돕는 단순하면서도 강력한 보안 툴
- 쉽고 신속하게 소프트웨어를 설정 및 실행할 수 있는 기능
- 업데이트된 컨테이너 관련 자료에 대한 액세스

RED HAT 컨테이너 호스트

현재 Red Hat Enterprise Linux 서브스크립션을 보유하고 있는 고객들은 인증되고 지원되는 컨테이너 호스트에 액세스할 수 있습니다. 큰 규모에서 운영을 단순화하는 것을 선호하거나 개발을 위한 단순한 컨테이너 호스트를 필요로 하는 고객들은 Red Hat Enterprise Linux Atomic Host를 통해 큰 이점을 얻을 수 있으며, 이를 Red Hat Enterprise Linux 서브스크립션의 일부로서도 이용할 수 있습니다. Red Hat Enterprise Linux 와 Red Hat Enterprise Linux Container Host는 모든 인증된 물리서버, 가상머신, 클라우드 플랫폼(OpenStack) 또는 클라우드 제공업체 호스트 등에서 실행될 수 있으며, 보다 우수한 컨테이너를 실행하고 지원받기 원하는 Red Hat 고객들에게 많은 배포 옵션을 제공합니다.

액세스 방법

Red Hat Container Catalog를 방문해 Red Hat이 제공하는 모든 이미지에 대해 자세히 알아 보세요:
http://red.ht/2qrkW8K

Red Hat 포털을 방문해 Red Hat이 제공하는 컨테이너 호스트에 대해 자세히 알아 보세요:
http://red.ht/2q3gO92

Red Hat Developers 사이트에 방문하여 무료 개발자 서브스크립션을 이용해 시작해 보세요:
http://red.ht/2qkiM7H

RED HAT 소개

Red Hat은 세계적인 오픈소스 솔루션 공급업체로서 커뮤니티 기반의 접근 방식을 통해 신뢰도 높은 고성능 클라우드, Linux, 미들웨어, 스토리지, 가상화 기술을 제공합니다. 또한, 전세계 고객에게 높은 수준의 지원과 교육 및 컨설팅 서비스를 제공하여 권위있는 어워드를 다수 수상한 바 있습니다. Red Hat은 기업, 파트너, 오픈소스 커뮤니티로 구성된 글로벌 네트워크의 허브 역할을 하며 고객들이 IT의 미래를 준비하고 개발할 수 있도록 리소스를 공개하여 혁신적인 기술 발전에 기여하고 있습니다.

www.facebook.com/redhatkorea
구매문의 080-708-0880
buy-kr@redhat.com

www.redhat.com/ko
#8326_0817