

Red Hat과 AWS 솔루션을 이용한 컨테이너 개발과 배포



Red Hat과 AWS를 통해 컨테이너 사례 가속화

클라우드 기술 발전, 마이크로서비스, 시장 디스럽션이 융합되면서 디지털 트랜스포메이션을 둘러싼 긴박감이 조성되었습니다. 기업은 더 신속하게 전환하고 민첩성과 효율성을 향상하기를 원합니다. 애플리케이션은 이러한 디지털 이니셔티브의 성공에 핵심적인 역할을 하지만 시간이 가장 중요합니다. 이제는 단 몇 주 만에 애플리케이션을 배포해야 하는 것이 일반적입니다.

애플리케이션 컨테이너는 애플리케이션 제공을 가속화하는 데 도움이 됩니다. 핵심은 컨테이너 이미지로, 변경할 수 없는 정적 파일입니다. 이는 실행 가능한 코드가 포함되어 IT 인프라에서 격리된 프로세스로 실행될 수 있습니다. 이미지는 소프트웨어 프로그램이 컨테이너화 플랫폼에서 실행하기 위해 필요한 시스템 라이브러리, 시스템 툴, 기타 플랫폼 설정으로 구성됩니다. 해당 이미지는 호스트 머신의 운영 커널을 공유합니다.

컨테이너는 개발 환경과 배포 환경이 동일하기 때문에 프로세스에 일관성을 부여합니다. 결과적으로 각 단계는 애플리케이션의 변경 불가능한 표준 표현에 대해 검증되므로 릴리스 주기가 크게 단축됩니다.

Linux® 컨테이너는 네임스페이스와 함께 Linux에서 프로세스 격리를 활용하여 격리된 프로세스를 생성하기 때문에 대부분의 디지털 트랜스포메이션 요구 사항에 적합합니다. Red Hat® Enterprise Linux는 엔터프라이즈 환경에서 Linux 컨테이너를 실행하기 위한 표준으로, 컨테이너 구축에 필요한 다양한 옵션을 제공합니다. 개발자는 이러한 옵션을 통해 컨테이너를 쉽게 가동할 수 있지만 배포와 오케스트레이션도 관리해야 합니다.

이 문서에서는 Red Hat 컨테이너 툴과 Amazon Web Services(AWS) 개발자 툴을 활용하여 개발자가 애플리케이션을 컨테이너화하고 제공하는 데 어떤 도움을 얻을 수 있는지 설명합니다.

Linux 애플리케이션 컨테이너의 장점

Linux 애플리케이션 컨테이너를 사용하면 IT가 비즈니스 요구 사항을 더 신속하게 충족할 수 있습니다. 개발자는 애플리케이션을 모듈식 마이크로서비스로 분할한 후 Linux 컨테이너로 배포합니다. 컨테이너화를 수행하면 해당 애플리케이션 구성 요소의 풋프린트가 줄어들어 더욱 빠르게 배포할 수 있습니다.

Linux 컨테이너 애플리케이션은 하드웨어와 운영 체제에서 추상화되기 때문에 아키텍처 전반에서 이식성이 매우 뛰어납니다. 컨테이너를 사용하면 개발자의 노트북에서 프로덕션 클러스터에 이르기까지 다양한 환경에서 애플리케이션을 더 손쉽게 배포하고 실행할 수 있습니다. 또한 애플리케이션이 실행되고 있는 서버에서 애플리케이션 업그레이드와 롤백을 분리합니다. 컨테이너는 운영 간소화, 모듈화, 유연성을 통해 IT 관리자의 부담을 줄여줍니다. 쿠버네티스 또는 Red Hat OpenShift와 같은 오케스트레이션 시스템을 사용하여 컨테이너를 관리하면 모듈화와 유연성의 장점이 더욱 커집니다.

Linux 컨테이너를 사용하면 인프라 비용을 절감할 수 있습니다. 컨테이너는 작고 가벼운 특성으로 인해 동일한 운영 체제를 실행하는 퍼블릭, 프라이빗, 하이브리드, 멀티클라우드 환경은 물론 베어메탈 시스템 전반에서 손쉽게 이동할 수 있습니다. 컨테이너가 기본 운영 체제(OS)와 호환되는 경우 컨테이너는 새 애플리케이션의 개발 속도를 높이고 기존 애플리케이션을 최적화하며 모든 애플리케이션을 연결하는 데 도움이 됩니다.

가상 머신(VM)과 비교해 컨테이너는 다음 용도에 가장 적합합니다.

- ▶ 클라우드 네이티브 애플리케이션 빌드
- ▶ 마이크로서비스 패키징
- ▶ DevOps 또는 지속적 통합/지속적 제공(CI/CD) 사례 도입
- ▶ 동일한 운영 체제를 공유하는 다양한 IT 플랫폼 전반에서 확장 가능한 IT 프로젝트 이동

반면 VM은 단일 컨테이너보다 더 많은 작업을 실행할 수 있으므로 일반적으로 모놀리식 워크로드에 사용됩니다. 하지만 이처럼 확장된 기능으로 인해 OS, 애플리케이션, 라이브러리에 의존하게 되어 VM의 이식성이 저하됩니다.

컨테이너와 비교해 VM은 다음 용도에 가장 적합합니다.

- ▶ 기존, 레거시, 모놀리식 워크로드 수용
- ▶ 위험한 개발 사이클 분리
- ▶ 인프라 리소스(예: 네트워크, 서버, 데이터) 프로비저닝
- ▶ 다른 OS 내에서 또 다른 OS 실행(예: Linux에서 Unix 실행)

환경에서 실행되는 VM과 운영 체제의 수를 줄이면 라이선스 요구 사항이 크게 줄어듭니다. Linux 컨테이너를 사용하면 이러한 비용을 더욱 절감할 수 있습니다. Red Hat과 AWS의 툴과 플랫폼은 컨테이너화된 클라우드 네이티브 애플리케이션을 구축하고 DevOps와 CI/CD 사례를 채택하여 민첩성을 높이고 출시 시간을 앞당기는 데 도움이 될 수 있습니다.

Red Hat Enterprise Linux에서 애플리케이션 컨테이너 개발

Red Hat 솔루션을 사용하면 Red Hat Enterprise Linux에서 애플리케이션 컨테이너를 구축할 수 있습니다. 경량의 개방형 표준 기반 컨테이너 툴킷이 완벽하게 지원되며 Red Hat Enterprise Linux 8에 포함되어 있습니다. 또한 이러한 툴은 AWS 소프트웨어 개발 키트(SDK) 및 클라우드 개발 키트(CDK)와 연결되어 엔터프라이즈 애플리케이션을 구축하고 테스트할 수 있습니다.

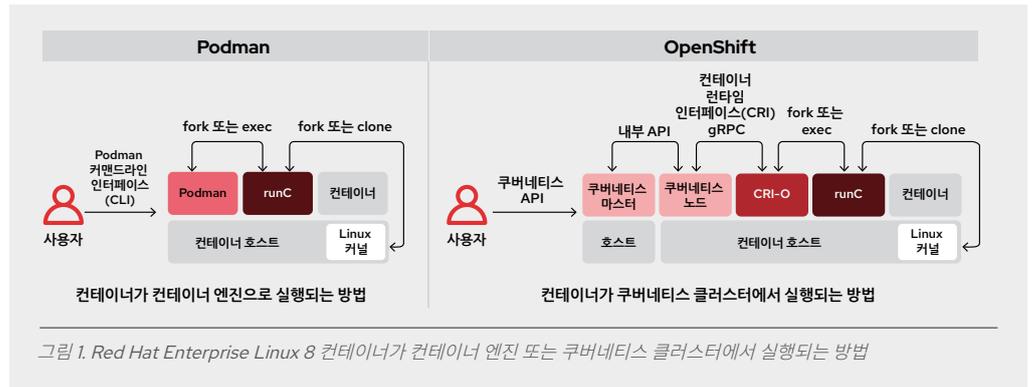
컨테이너를 개발한 후에는 배포해야 합니다. 오케스트레이션에 쿠버네티스를 사용하면 진행을 방해할 수 있는 복잡성, 확장성, 서비스 검색 등과 관련된 문제가 발생할 수 있습니다. Red Hat OpenShift는 쿠버네티스와 컨테이너 배포, 관리와 관련된 잠재적인 문제를 추상화하여 코드 작성에 집중할 수 있도록 설계되었습니다. Red Hat Enterprise Linux의 경우 작고 민첩한 툴 세트를 통해 컨테이너를 쉽게 개발할 수 있습니다.

Red Hat Enterprise Linux 8과 컨테이너 개발

Red Hat Enterprise Linux는 더 적은 리포지토리와 더 다양해진 개발자 툴로 컨테이너 개발을 간소화합니다. Red Hat은 이러한 툴 외에도 자체 이미지를 위한 기반으로서 기본 이미지를 제공합니다. 이러한 기본 이미지 중 일부는 Node.js, PHP, Java™, Python을 사용하여 구축된 비즈니스 애플리케이션에서 로깅, 데이터 수집, 인증과 같은 인프라 기능에 이르기까지 다양한 활용 사례를 대상으로 합니다.

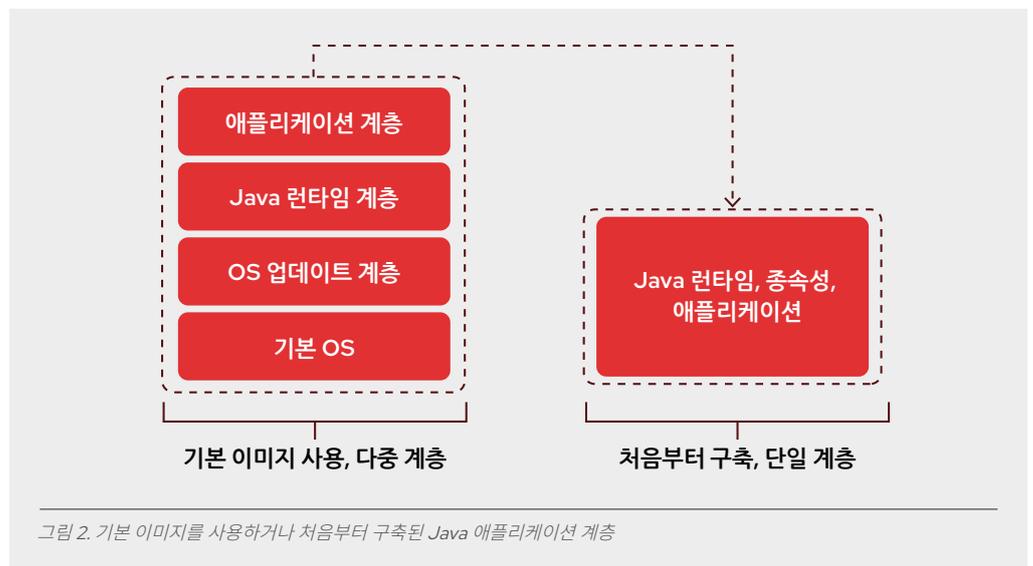
Red Hat Enterprise Linux 8에는 커맨드라인 툴(예: Buildah, Podman, runC, Skopeo), Universal Base Image(UBI), Red Hat Quay의 리포지토리, 추가 리포지토리를 위한 새로운 기능이 도입되었습니다. 이러한 기능을 통해 복잡한 컨테이너를 한층 더 쉽게 개발할 수 있습니다.

다음 다이어그램은 Red Hat Enterprise Linux 8 컨테이너의 아키텍처를 보여줍니다.



Buildah

Buildah를 사용하면 데몬이나 Docker를 설치하지 않고도 컨테이너를 처음부터 구축할 수 있습니다. 처음부터 구축하는 데 유용한 활용 사례는 Java 애플리케이션의 개발 이미지와 스테이징 또는 프로덕션 이미지를 비교하여 고려하는 것입니다. 개발 중에는 Java 애플리케이션의 컨테이너 이미지에 Java 컴파일러와 Maven 및 기타 툴이 필요할 수 있습니다. 그러나 프로덕션 환경에서는 Java 런타임과 패키지만 필요할 수 있습니다.



Buildah는 데몬이 없기 때문에 호스트에 특별한 인프라를 설치하지 않고도 컨테이너에서 더 쉽게 실행할 수 있습니다.

Buildah는 Open Container Initiative(OCI)를 준수하는 Linux 컨테이너 이미지를 생성하거나 수정하는 데 필요한 기본 요구 사항만 제공하기 때문에 기존 빌드 파이프라인에 쉽게 통합할 수 있습니다. 예를 들어, Buildah는 최종 이미지에 필요하지 않은 경우 패키지 관리자(DNF/YUM)를 포함하지 않은 컨테이너를 어셈블할 수 있습니다. Buildah를 사용하면 이러한 컨테이너를 간단하고 일관된 보안으로 구축할 수 있을 뿐만 아니라 오버헤드를 줄이고(따라서 이미지 크기도 줄일 수 있음) 클라우드 네이티브 애플리케이션에 필요한 사용자 정의를 확장할 수도 있습니다.

Buildah의 가장 강력한 용도는 Dockerfile을 작성하는 것과 유사하게 이미지 생성을 위해 Bash 스크립트를 작성하는 것입니다.

Podman

Podman을 사용하면 별도의 데몬 없이 컨테이너를 관리할 수 있고 Docker 커맨드라인 인터페이스(CLI)와도 호환됩니다. Podman에서는 runC 컨테이너 런타임 프로세스를 통해 이미지 레지스트리, 컨테이너, 이미지 스토리지, Linux 커널과의 직접적인 상호 작용이 가능합니다.

루트 기능을 제공하는 데몬이나 루트 권한 없이도 Podman을 실행할 수 있으므로 Podman에서 이미지를 쓸 수 있는 별도의 위치가 필요합니다. Podman은 사용자의 홈 디렉터리에 있는 리포지토리를 사용하므로 /var/lib/containers를 쓰기 가능하도록 설정하거나 보안 문제를 일으킬 수 있는 기타 사례를 적용하지 않아도 됩니다. 이렇게 하면 모든 사용자가 별도의 컨테이너와 이미지 세트를 보유하게 되어 서로 간섭하지 않고 동일한 호스트에서 Podman을 동시에 사용할 수 있습니다. 사용자가 작업을 마치면 공통 레지스트리로 변경 사항을 푸시하여 다른 사람과 이미지를 공유할 수 있습니다.

runC

Buildah와 Podman은 OCI 런타임인 runC를 사용하여 컨테이너를 시작합니다. 따라서 runC로 이미지를 구축하고 실행하거나 Docker 형식의 이미지를 실행할 수 있습니다. 이 Go 언어 기반의 툴은 런타임 사양을 읽고 Linux 커널을 구성하여, 컨테이너 프로세스를 생성하고 시작합니다. runC의 하위 수준 코드는 Docker와 공유할 수 있지만 runC는 Docker 플랫폼 구성 요소에 의존하지 않습니다. Linux 네임스페이스, 실시간 마이그레이션을 지원하고 이식 가능한 성능 프로필이 있습니다. 또한 SELinux, 제어 그룹(cgroups), 보안 컴퓨팅 모드(seccomp) 등과 같은 Linux 보안 기능을 완벽하게 지원합니다.

Skopeo

Skopeo는 OCI 이미지를 저장할 수 있는 모든 위치의 이미지를 검사하고 해당 위치에 이미지를 전송합니다. Skopeo가 출시되기 전에는 이미지를 검사하려면 일부 메타데이터만 검사하려는 경우에도 전체 이미지를 가져와야 했습니다. Skopeo는 레이어를 포함한 모든 이미지 속성을 표시하므로 이미지를 호스트로 가져올 필요가 없습니다. 검사 결과 컨테이너 이미지를 한 위치에서 다른 위치로 복사해야 하는 경우 Skopeo를 사용하여 복사할 수 있습니다.

Skopeo는 리포지토리에서 이미지를 삭제하고, 에어 갭(air-gapped) 또는 네트워크 연결이 끊긴 환경에서 배포할 수 있도록 외부 이미지 리포지토리를 내부 레지스트리에 동기화합니다. 리포지토리에 필요한 경우, Skopeo는 인증을 위해 적합한 자격 증명과 인증서를 통과시킬 수 있습니다.

Red Hat Universal Base Image

전통적으로 Linux 컨테이너를 각 대상 플랫폼에 맞게 구축되어야 합니다. [Red Hat Universal Base Image\(UBI\)](#)는 이러한 문제를 해결합니다. UBI는 개발자가 애플리케이션을 구축하고 제공할 수 있는 엔터프라이즈급 기본 컨테이너 이미지입니다. Red Hat Enterprise Linux 8에서는 모든 Red Hat 기본 이미지를 UBI로 사용할 수 있습니다. UBI를 사용하면 Red Hat 서브스크립션 없이도 Red Hat Enterprise Linux를 기반으로 컨테이너 이미지를 구축하고 재배포할 수 있으며 사용자도 서브스크립션이 필요하지 않습니다.

오랜 시간 테스트를 거친 Red Hat Enterprise Linux 기본 이미지를 기반으로 하는 UBI는 컨테이너에서 개발된 클라우드 네이티브 및 웹 애플리케이션 활용 사례의 기반이 되도록 설계되었습니다. 또한 커뮤니티 프로젝트 또는 자체 지원을 선호하는 사용자를 위해 CentOS 기반 컨테이너 이미지를 생성하는 추가 작업이 제거되었습니다.

[UBI를 사용하여 컨테이너화된 애플리케이션](#)을 구축하고, 선택한 레지스트리 서버로 이를 푸시한 후 프로덕션으로 전송하여 손쉽게 공유할 수 있습니다. 재배포가 자유롭기 때문에 다른 벤더의 플랫폼에도 배포할 수 있습니다. 이 기능은 Ubuntu 또는 다른 Linux 플랫폼을 사용하는 경우 유용하며, 다른 플랫폼을 사용하더라도 Red Hat Enterprise Linux에 익숙해지는 데 도움이 됩니다.

Red Hat Quay의 리포지토리

컨테이너 이미지를 저장, 구축, 배포하는 프라이빗 컨테이너 레지스트리인 [Red Hat Quay](#)에서 이미지 리포지토리를 생성할 수도 있습니다. Red Hat Quay는 이미지의 보안 취약점을 분석하여 보안 위험을 완화하는 데 도움이 되는 잠재적 문제를 식별합니다. Red Hat Quay에서는 두 가지 방법으로 리포지토리를 생성합니다. 하나는 푸시(Docker 또는 Podman에서)를 사용하는 것이고 다른 하나는 Red Hat Quay 사용자 인터페이스(UI)를 사용하는 것입니다. Red Hat Quay에서는 퍼블릭 리포지토리에 액세스할 수도 있습니다.

추가 리포지토리

[추가 리포지토리](#)에는 오픈소스 Red Hat Enterprise Linux 리포지토리에 포함되지 않은 독점 라이선스 패키지가 포함되어 있습니다. 이러한 패키지는 Red Hat Enterprise Linux 8에서 지원되지 않지만 애플리케이션 컨테이너화의 일부로 다른 소프트웨어에 액세스할 수 있는 유연성을 제공합니다.

Amazon Elastic Compute Cloud(EC2)에서 실행되는 Red Hat Enterprise Linux

Amazon Elastic Compute Cloud(Amazon EC2)에서 실행되는 Red Hat Enterprise Linux는 Amazon EC2 컴퓨팅과 Red Hat Enterprise Linux가 결합된 것입니다. Red Hat은 Amazon EC2용 기본 Red Hat Enterprise Linux 이미지를 유지 관리합니다. 업데이트는 Red Hat에서 제공하는 즉시 수신됩니다. 컴퓨팅 환경과 보안이 안정적으로 유지되며 Red Hat Enterprise Linux 인증 애플리케이션에서 지원 가능성을 유지 관리합니다. Red Hat Enterprise Linux를 실행하는 Amazon EC2는 컨테이너화된 애플리케이션을 포함하여 광범위한 애플리케이션을 제공하기 위해 가상 개발 환경과 신뢰할 수 있는 플랫폼을 제공합니다. 분산형 아키텍처를 통해 애플리케이션을 더 유연하고 민첩하게 배포할 수 있습니다.

Red Hat Enterprise Linux on EC2에서는 핵심 컨테이너 언어로 간주되는 Go와 함께 C++, Java, Rust를 비롯하여 공통 언어를 위한 AWS SDK에 액세스할 수 있습니다. 해당 언어 모두 애플리케이션을 컨테이너화하는 데 사용할 수 있습니다. Red Hat Enterprise Linux에 포함된 컨테이너 빌딩 블록과 AWS SDK를 결합할 수도 있습니다. 또한 Red Hat Universal Base Image(UBI)와 애플리케이션 스트림을 활용하여 원하는 위치에서 컨테이너화된 애플리케이션을 구축, 공유, 협업할 수 있습니다.

보시다시피 Red Hat Enterprise Linux 8 컨테이너 개발은 개별 컨테이너와 컨테이너 이미지의 개발, 구축, 관리를 위한 실행 가능한 옵션입니다. 그러나 **Red Hat OpenShift®**를 사용하면 컨테이너 개발과 배포의 확장성과 효율성이 한층 더 향상됩니다.

Red Hat OpenShift on AWS: 컨테이너 배포와 개발 결합

컨테이너는 클라우드 네이티브 애플리케이션을 구축하거나, 마이크로서비스를 패키징하거나, DevOps 또는 CI/CD 사례를 사용하거나, 동일한 운영 체제를 공유하는 다양한 IT 포트폴리오에서 확장 가능한 IT 프로젝트를 이동하는 경우 이상적입니다. 운영 체제, 애플리케이션, 라이브러리에 연결되지 않아 이식 가능하고 경량화되어 있지만, 문제점도 있습니다. 위험한 개발 주기 분리

또한 컨테이너 런타임만 사용하는 경우 복잡성은 장애물이 될 수 있습니다. 운영을 확장하려면 컨테이너를 추가해야 하므로 별도의 프로세스 체인이 빠르게 생성됩니다. 이로 인해 서버에 컨테이너가 너무 많이 생성되어 성능이 저하될 수 있습니다. 또한 컨테이너(볼륨)에서 분리된 스토리지를 추적하지 못할 수 있습니다. 따라서 컨테이너 배포에 오케스트레이션 계층이 중요합니다. 쿠버네티스를 오케스트레이터로 선택해도 고유한 네이티브 쿠버네티스를 배포하는 작업은 복잡하고 문제가 발생할 수 있습니다. 쿠버네티스와 함께 Red Hat Quay를 사용하여 전체 애플리케이션을 시작하고 실행하는 데 필요한 이미지와 구성 세부 정보를 관리할 수도 있습니다. 그러나 Red Hat OpenShift는 쿠버네티스의 복잡성을 거의 모두 제거하므로 쿠버네티스에서 최대한 빨리 시작하고 실행할 수 있습니다.

Red Hat OpenShift에서 사용 가능한 개발 툴:

- ▶ **CodeReady Workspaces**를 사용하면 원격 개발 팀이 버튼 하나만 클릭하여 환경을 프로비저닝하고 공유할 수 있어 더 빠르게 시작하고 상호 작용 대기 시간을 단축할 수 있습니다.
- ▶ **Quarkus 프레임워크**를 사용하면 쿠버네티스 네이티브 Java 애플리케이션을 구축할 수 있습니다.
- ▶ **Buildpack**과 **Kaniko**에 대한 미리 보기 지원은 Buildah를 통해 S2I 및 Dockerfile 빌드와 함께 사용할 수 있습니다.
- ▶ **Helm**은 차트와 릴리스 작업을 간소화합니다.

Red Hat OpenShift에서 컨테이너 배포

Red Hat OpenShift는 컨테이너 플랫폼을 더 간편하게 배포하고 관리할 수 있게 설계되었습니다. 통합 플랫폼 모니터링, 자동화된 유지 관리 운영, 업그레이드 기능을 갖춘, 관리자 친화적인 엔터프라이즈급 쿠버네티스 컨테이너 플랫폼입니다.

Red Hat OpenShift는 일단 작성하면 어디서나 실행할 수 있다는 뜻의 "WORA(Write Once, Run Anywhere)" 전략을 제공하여 온프레미스 또는 퍼블릭, 프라이빗, 하이브리드 클라우드 환경에서 가장 의미 있는 워크로드를 실행할 수 있도록 합니다. AWS에서 Red Hat OpenShift는 AWS 콘솔에 나열된 또 다른 서비스로 통합되고 패키징되므로 OpenShift 클러스터를 더 신속하고 간편하게 생성하고 일관되게 관리할 수 있습니다. 또한 온디맨드 청구, 단일 청구서, AWS 지원 문의 옵션이 제공됩니다. AWS에 **Red Hat OpenShift 클러스터를 배포**할 때는 기본 설정 또는 사용자 정의 AWS 설정을 사용합니다. 직접 프로비저닝한 AWS 인프라에 클러스터를 배포할 수도 있습니다. 제공된 AWS CloudFormation 템플릿을 필요에 맞게 수정할 수 있습니다.

Red Hat OpenShift에서 컨테이너 개발

Red Hat OpenShift에는 S2I(Source-to-Image)가 포함되어 있으므로 애플리케이션 코드에서 컨테이너로 바로 이동할 수 있어 컨테이너 개발 복잡성이 크게 감소합니다. 애플리케이션 소스 코드를 입력으로 사용하는 이미지를 작성하고 어셈블된 애플리케이션을 출력으로 실행하는 새 이미지를 생성하기만 하면 됩니다. Dockerfile과 S2I의 두 가지 방법을 사용하여 OpenShift에서 빌드할 수 있습니다.

사용자 제공 Dockerfile은 OpenShift 빌더에 포함된 Buildah를 호출하여 이미지를 생성합니다. Buildah도 결과 이미지에 태그를 지정하고 푸시합니다. 또는 S2I를 사용하여 Buildah 기반 Dockerfile 빌드와 동일한 흐름을 따라 Dockerfile을 생성할 수 있습니다. 결과 빌더 이미지에는 해당 실행 가능 이미지(빌드 아티팩트라고도 함)를 생성하는 데 필요한 특정 인텔리전스가 포함되어 있습니다.

S2I를 통해 알 수 있듯이 Red Hat OpenShift는 배포용 플랫폼인 동시에 컨테이너 개발용 플랫폼입니다. 개발자는 기존의 특정 AWS 인프라에 코딩하는 대신 쿠버네티스 플랫폼에서 구축할 수 있습니다. 또한 Red Hat OpenShift에서는 컨테이너가 포함된 애플리케이션 개발에 다양한 방식으로 접근할 수 있으므로 상황에 맞는 방식을 사용하면 됩니다. 쿠버네티스에 익숙하지 않고 코딩만 하려는 개발자는 물론 최대한의 유연성을 추구하는 전문 쿠버네티스 개발자의 요구 사항도 충족합니다.

Red Hat OpenShift는 개발자가 이미 사용하고 있는 언어, 데이터베이스, 툴을 지원합니다. 또한 AWS에서 제공하는 것과 같은 애플리케이션 개발 서비스에 손쉽게 액세스할 수 있습니다.

Red Hat OpenShift의 컨테이너를 위한 AWS 개발자 툴

AWS 툴을 사용하면 Red Hat OpenShift 컨테이너를 훨씬 더 쉽게 개발할 수 있습니다. 이러한 툴은 Red Hat Enterprise Linux를 포함하여 Red Hat OpenShift 이외의 플랫폼에서도 사용할 수 있습니다.

AWS CodeArtifact는 모든 규모의 조직이 소프트웨어 개발 프로세스에 사용되는 소프트웨어 패키지를 더욱 안전하게 저장, 게시, 공유할 수 있는 전체 관리형 아티팩트 리포지토리 서비스입니다.

AWS Cloud9은 브라우저 기반 셸에서 코드를 작성, 실행, 디버깅하는 데 필요한 통합 개발 환경(IDE)으로, 추가 소프트웨어를 설치하거나 Git 푸시를 수행하거나 명령을 입력할 수 있습니다.

AWS 클라우드 개발 키트는 프로그래밍 언어의 친숙함과 표현력을 활용하여 애플리케이션을 모델링하므로 새로 학습해야 할 내용이 거의 없어 AWS에 신속하게 온보딩할 수 있습니다.

Amazon CodeGuru는 성능 문제를 해결하는 방법과 비효율적인 코드 실행으로 예상되는 비용을 시각화하고 이에 대한 권장 사항을 제공하여 개발자가 문제 해결 우선 순위를 지정할 수 있도록 지원합니다. Red Hat OpenShift의 기존 소프트웨어 개발 워크플로우에 통합되어 코드 품질을 개선하고 애플리케이션에서 비용이 가장 많이 드는 코드 라인을 식별할 수 있는 지능적인 제안을 제공합니다.

지금 컨테이너 구축, 배포, 관리를 시작하세요

애플리케이션 컨테이너를 구축하는 일은 어려운 일이 아닙니다. Red Hat 컨테이너 툴과 AWS 개발자 툴이 도움이 될 수 있습니다. 컨테이너 구현의 마지막 단계인 컨테이너 배포와 오케스트레이션은 프로세스에서 가장 어려운 부분일 수 있습니다. Red Hat OpenShift는 대부분의 쿠버네티스 복잡성을 제거하여 구축 방식에 관계없이 컨테이너를 대규모로 배포하고 관리할 수 있도록 지원합니다.

Red Hat OpenShift의 가치는 단순한 배포 솔루션이 아니라는 것입니다. 개발자는 플랫폼에서 컨테이너를 개발한 다음 배포하기 위해 다양한 선택을 할 수 있습니다. Red Hat OpenShift는 AWS에서 자체 관리형 옵션과 전체 관리형 옵션으로 사용할 수 있습니다. Red Hat OpenShift를 사용하면 AWS에서 자체 OpenShift 구현을 배포하고 관리할 수 있습니다. Red Hat OpenShift Service on AWS와 Red Hat OpenShift Dedicated에서 전체 Red Hat Cloud Services를 제공하므로 애플리케이션 구축과 출시에 집중할 수 있습니다.

Red Hat OpenShift의 추가 개발 툴:

- ▶ Odo를 사용하면 개발자가 CLI로 코드를 반복할 수 있습니다. 쿠버네티스, Red Hat OpenShift를 지원하며, 표준 정의를 통한 툴에 대한 개방형 모델과 Quarkus를 사용한 신속한 반복 Java 개발을 지원합니다.
- ▶ 개발자는 Knative 서빙과 이벤트에 대한 서버리스 지원을 통해 Strimzi(OpenShift에서 Apache Kafka 실행용)와 Service Mesh를 포함하는 서버리스 및 이벤트 기반 애플리케이션을 구축할 수 있습니다.
- ▶ Red Hat OpenShift 파이프라인의 Tekton, GitHub 작업용 OpenShift 플러그인, Jenkins, GitLab 러너 지원은 지속적인 통합 기능을 제공합니다.

Red Hat과 AWS는 컨테이너를 개발하거나 배포하는 방법에 관계없이 프로세스를 간소화하고 가속화하는 솔루션을 제공하므로 비즈니스에 필요한 애플리케이션을 더 빠르게 생성하고 배포할 수 있습니다.

Red Hat과 AWS의 파트너십이 Red Hat Enterprise Linux로 현대적인 인프라를 구축하는 데 어떤 도움이 되는지 알아보거나, 다음에 대해 자세히 알아보세요.

- ▶ [Red Hat Enterprise Linux](#) [도큐멘테이션](#)
- ▶ [Podman](#)
- ▶ [AWS의 개발자 툴](#)

한국레드햇 홈페이지 <https://www.redhat.com/ko>



Red Hat 소개

Red Hat은 세계적인 오픈소스 소프트웨어 솔루션 공급업체로서 커뮤니티 기반의 접근 방식을 통해 신뢰도 높은 고성능 Linux, 하이브리드 클라우드, 컨테이너, 쿠버네티스 기술을 제공합니다. 또한 Red Hat은 고객이 클라우드 네이티브 애플리케이션을 개발하고, 신규 및 기존 IT 애플리케이션을 통합하고, 복잡한 환경을 자동화하고 관리할 수 있도록 지원합니다. **Fortune 선정 500대 기업의 신뢰를 받는 어드바이저인 Red Hat**은 전 세계 고객에게 **권위 있는 어워드를 수상**한 지원, 교육 및 컨설팅 서비스를 제공하여 모든 산업 분야에서 오픈 혁신의 이점을 실현할 수 있도록 최선을 다하고 있습니다. Red Hat은 기업, 파트너, 커뮤니티로 구성된 글로벌 네트워크의 허브 역할을 하며 고객들이 성장하고, 확장하고, 디지털 미래에 대비할 수 있도록 지원합니다.

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

Copyright © 2021 Red Hat, Inc. Red Hat, Red Hat 로고 및 OpenShift는 미국과 그 외 국가의 Red Hat, Inc. 또는 계열사의 상표이거나 등록 상표입니다. Linux®는 미국 및 기타 국가에서 Linus Torvalds의 등록 상표입니다. 기타 모든 상표는 해당 소유자의 재산입니다. Java 및 모든 Java 기반 상표와 로고는 미국 및 기타 국가에서 Oracle America, Inc.의 상표 또는 등록 상표입니다. *공간이 없는 경우 “Java는 Oracle America, Inc.의 상표입니다.”라고 대체할 수 있습니다.