레거시 애플리케이션을 컨테이너로 이전

소개

많은 조직에서 초기 퍼블릭 클라우드 프로젝트를 성공적으로 진행하고 있습니다. 하지만 이 프로젝트들은 주로 퍼블릭 클라우드에서 실행하기에 적합한 대상으로 신중하게 선택된 새로운 그린필드 애플리케이션 프로젝트입니다. 이러한 프로젝트가 성공하면서 IT 조직은 클라우드 컴퓨팅이 제공하는 탄력성, 확장성 및 배포 속도에 관심을 가지게 되었습니다. IT 조직은 클라우드 기술을 사용하여 개발자 및 비즈니스 요구 사항에 더욱 신속하게 대응할 수 있습니다.

레거시 애플리케이션은 일반적으로 보안, 규제, 데이터 지역성(data locality) 또는 성능 문제로 인해 퍼블릭 클라우드 배포에 적합하지 않습니다. 또한 레거시 애플리케이션은 클라우드 컴퓨팅 이전에 작성된 경우가 많으므로 기존 인프라에 배포된 상태로 두는 것이 더 간단해 보일 수 있습니다. 그러나 이러한 결정은 현대화를 시도하는 조직에 장애물이 될 수 있습니다. 비용을 줄이면서 대응 능력을 높이려면 레거시 애플리케이션 문제를 해결해야만 성공할 수 있습니다. 이러한 애플리케이션 구동 비용이 IT 비용의 대부분을 차지하는 경우가 많기 때문입니다.

컨테이너는 퍼블릭 클라우드 공급업체가 제공하는 많은 서비스를 가능하게 하는 핵심 기술입니다. 컨테이너 설계는 다양한 자동화의 가능성을 열어주고, 클라우드와 유사한 자동화를 제공하는 플랫폼과 결합해 최적의 애플리케이션 실행 환경을 제공합니다. 레거시 애플리케이션을 컨테이너로 마이그레이션하면 현대화를 가로막는 많은 장애 요소를 제거할 수 있습니다.

레거시 애플리케이션을 컨테이너로 이전하는 이유

확장 및 신속한 대응의 필요성

레거시 애플리케이션은 리소스가 고정되고 제한적인 인프라에 배포되는 경우가 많으며, 리소스 활용도가 낮은 경우도 많습니다. 그러나 수요가 증가하면 스케일 업을 위한 리드 타임이 길어지고 비용이 높아집니다. 퍼블릭 클라우드에서 실행되는 서비스로서의 소프트웨어(SaaS) 애플리케이션이 성공하면서, 대응 능력과 비용에 대한 사용자와 비즈니스의 기대 수준이 달라졌습니다. 내부 애플리케이션이 빠르게 진화할 수 없는 이유를 설명하는 것은 쉽지 않습니다.

많은 레거시 애플리케이션이 과거에 안정적이고 예측 가능한 성장을 이루었지만, 새로운 사용자 중심 수요가 등장하면서 레거시 애플리케이션에 사용할 수 있는 리소스를 신속하게 확장해야 하는 상황이 되었습니다. IT 조직에서 사용자 중심 수요를 예측하기가 어려운 이유는 다음과 같습니다.

- 현재 모바일 및 연결된 애플리케이션에서 기존 애플리케이션에 대한 애플리케이션 프로그래밍 인터페이스(API) 수준의 액세스를 요구하는 것이 일반적입니다.
- 데이터 사이언스 및 머신 러닝의 등장으로 데이터 접근에 대한 수요가 증가하고 있습니다.
- 또한 데이터 및 API를 사용하는 애플리케이션을 비롯해 수요 일부가 IT 조직 외부에서 발생할 수 있습니다.

성장을 예측하고 수요를 제어하기 어렵기 때문에 조직이 신속하게 대응할 수 있도록 기존 애플리케이션을 재배치해야 합니다. 이러한 문제를 해결하기 위해 수요에 따라 실행 중인 컨테이너 수를 늘리거나 줄임으로써 애플리케이션 용량을 조절하는 플랫폼의 컨테이너에서 현대적인 클라우드 스케일 애플리케이션을 실행합니다.

레거시 애플리케이션을 컨테이너에서 실행할 때의 장점

- 이식성: 인프라에서 애플리케이션을 분리하고 컨테이너를 지원하는 모든 플랫폼에서 애플리케이션을 실행합니다.
- 확장성: 수요에 대응하기 위해 필요에 따라 확장 또는 축소가 가능하며, 리소스 활용도가 높습니다.
- 유연성: 필요하지 않을 때 리소스를 묶어둘 필요 없이, 손쉽게 컨테이너를 배포하여 필요 시 테스트 환경을 조성할 수 있습니다.
- 다양한 언어 및 기술 활용: 다양한 언어, 데이터베이스, 프레임워크, 툴링을 선택할 수 있도록 지원하여 코드 작성 시기에 관계없이 레거시 기술과 보다 현대적인 기술을 함께 구동할 수 있습니다.

변화를 위한 레거시 시스템 포지셔닝

레거시 시스템과 새로운 그린필드 개발 기회는 연관되는 경우가 많습니다. 새로운 애플리케이션과 서비스는 보통 레거시 애플리케이션의 데이터를 필요로 하며 레거시 시스템에서 트랜잭션을 실행하여 서비스를 수행할 수 있습니다. 현대화에 대한 일반적인 접근 방식은 새로운 기술에 구현된 새로운 인터페이스와 서비스를 레거시 시스템보다 우선하는 것입니다.

퍼블릭 클라우드에서 새로운 개발을 연결해 레거시 애플리케이션을 내부에서 실행하면 복잡성과 보안 과제가 더욱 가중됩니다. 특히 네트워크 관련 문제는 진단 및 추적이 더욱 어렵습니다. 현대적인 툴을 사용할 수 없는 이전 인프라에서 레거시 애플리케이션을 실행하는 경우 이 문제는 더욱 까다로워집니다.

레거시 시스템에 의존하는 새로운 애플리케이션은 테스트가 필요합니다. 현대적인 개발 방법론은 품질과 신뢰성 향상을 위해 자동화된 테스트에 의존하는 경향이 있으므로 레거시 애플리케이션은 테스트 환경에서 더 많은 리소스를 필요로 할 수 있습니다. 또한 개발 팀은 새로운 코드를 개발하고 테스트하기 위해 추가로 격리 상태에 있을 수 있는 레거시 애플리케이션 테스트 환경에 액세스해야 할 수 있습니다.

컨테이너에 레거시 애플리케이션을 배포하면 변화를 가로막는 장애 요소를 없애고 진화를 위한 유연성을 제공할 수 있습니다. 프로세스는 기존 인프라에서 애플리케이션을 분리한 다음 동일한 플랫폼을 사용하여 레거시 애플리케이션 및 새로운 그린필드 개발을 호스팅하는 것으로 시작됩니다. 두 가지 모두 동일한 컨테이너 또는 클라우드 플랫폼에서 공존할 수 있으며 동일한 툴로 관리할 수 있으며, 기존 인프라의 제약 없이 자동화 및 현대적인 관리 툴을 레거시 애플리케이션과 함께 사용하면 운영 효율성이 향상됩니다.

레거시 애플리케이션을 컨테이너로 이전하기 위한 고려 사항

퍼시스턴트 스토리지

클라우드 네이티브가 아닌 애플리케이션에는 데이터, 로그 그리고 경우에 따라 구성에 대한 퍼시스턴트 스토리지가 필요합니다. 그러나 컨테이너는 단기간 동안 존재하도록 설계되었습니다. 다른 조치가 이루어지지 않으면 컨테이너를 재시작할 때 컨테이너 내부에 기록된 모든 내용이 손실됩니다. 컨테이너가 퍼시스턴트 스토리지에 액세스할 수 있도록 조정하면 레거시 애플리케이션을 사용할 수 있습니다. 일반적으로 컨테이너는 여러 머신으로 구성된 클러스터에서 실행되므로 컨테이너가 실행될 수 있는 클러스터의 모든 머신에서 퍼시스턴트 데이터 스토리지를 사용할 수 있어야 합니다. 사용 가능한 스토리지 유형은 컨테이너 플랫폼과 해당 플랫폼이 실행되는 인프라에 따라 크게 달라집니다.

컨테이너 오케스트레이션

대부분의 애플리케이션은 동시에 실행되고 서로 연결되어야 하는 컨테이너로 구성됩니다. 예를 들어, 3 계층 구조(3-tier) 애플리케이션을 구성하는 요소는 별도의 컨테이너에서 실행됩니다. 웹 또는 애플리케이션 컨테이너는 수요 증가에 따라 클러스터에서 더 많은 머신으로 동적 스케일 아웃이 가능하다는 장점이 있습니다. 컨테이너 스케줄링 및 관리 프로세스를 컨테이너 오케스트레이션이라고 하며, 이는 컨테이너 플랫폼의 핵심 작업입니다.

네트워킹

애플리케이션에는 배포 방식의 핵심인 특정 네트워킹 요구 사항이 있는 경우가 많습니다. 컨테이너 환경에서 가상 네트워크를 다시 생성해야 할 수도 있고, 경우에 따라서 물리적 네트워킹 하드웨어를 컨테이너 환경에서 가상화해야 할 수도 있습니다. 스토리지와 마찬가지로 애플리케이션의 가상 네트워크는 컨테이너가 실행되는 각 호스트에서 사용할 수 있어야 합니다. 컨테이너 플랫폼은 서로 다른 컨테이너에서 실행되는 애플리케이션의 구성 요소를 연결하는 가상 네트워크 환경을 관리하고, 컨테이너 플랫폼에서 실행되는 다른 애플리케이션으로부터 해당 구성 요소를 격리합니다.

Red Hat

쿠버네티스에 대하여

표준 컨테이너 플랫폼으로 사실상 자리잡은 쿠버네티스는 스케일에 따라 대량의 컨테이너를 구동하는 Google의 경험을 기반으로 컨테이너의 배포와 관리를 자동화하는 오픈소스 플랫폼입니다. 쿠버네티스는 자동으로 컨테이너를 재시작하고, 각기 다른 호스트에서 컨테이너를 다시 스케줄링하며, 자동 스케일링과 같은 활용 사례에 대해 컨테이너를 복제하여 애플리케이션을 원하는 최종 상태로 유지관리하는 자동화된 자가 복구(Self-healing) 메커니즘을 사용합니다. 쿠버네티스는 널리 사용되는 Docker 컨테이너 형식을 포함하여 기본적으로 Linux 컨테이너와 연동됩니다.

Red Hat은 Google 다음으로 쿠버네티스 프로젝트에 가장 크게 기여하는 기업으로 손꼽힙니다.[1] 2015년부터 쿠버네티스 프로젝트는 Cloud Native Computing Foundation에서 관리되고 있습니다. 쿠버네티스는 개방성 덕분에 업계에서 널리 채택되고 신속한 혁신을 촉진했으며, 쿠버네티스에 기반하는 오픈소스 프로젝트를 생성하게 되었습니다.

애플리케이션을 위한 컨테이너 구축

개발자는 애플리케이션 및 필수 종속성을 컨테이너 이미지에 구축하기 위한 툴이 필요합니다. 코드 변경 및 완료된 릴리스에 대해 이 프로세스를 반복해야 합니다. 롤아웃 중에 운영자 또는 개발자는 현재 실행 중인 컨테이너 이미지 대신 새 이미지를 배포할 수 있는 기능도 필요합니다. 이러한 태스크를 수행하기 위한 하위 수준의 툴이 있지만 컨테이너 플랫폼을 활용하면 이 프로세스가 훨씬 간편해집니다.

애플리케이션을 구동하기 위해 컨테이너를 구축하려면 애플리케이션을 실행할 수 있는 언어, 런타임, 프레임워크, 애플리케이션 서버가 필요한 경우가 많습니다. 이러한 리소스는 기본 컨테이너 이미지를 기반으로 하는 빌드 프로세스 중에 가져올 수 있습니다. 기본 이미지에 대한 소스는 여러 가지가 있지만, 신뢰할 수 있는 알려진 소스에서 이를 획득하는 것은 쉽지 않습니다. 기본 이미지는 안전하고 최신 상태이며 알려진 취약점이 없어야 합니다. 취약점이 발견되면 기본 이미지를 업데이트해야 합니다. 또한 사용자는 컨테이너가 오래된 이미지를 기반으로 하는지를 확인할 수 있는 방법이 필요합니다.

퍼블릭 클라우드 과제

IT 조직이 퍼블릭 클라우드를 도입할 때 직면하는 과제 중 하나는 퍼블릭 클라우드에서 제공하는 인프라, 관리 및 자동화 소프트웨어가 IT 조직 자체 데이터센터에서 사용하는 것과 다르다는 점입니다. 많은 퍼블릭 클라우드 툴 및 서비스를 온프레미스에서 실행할 수 없으므로 내부에서 실행되는 애플리케이션과 함께 사용할 수 없습니다.

지리적 가용성, 다양성, 비용과 같은 이유로 둘 이상의 퍼블릭 클라우드를 사용하는 조직이 많습니다. 그러나 퍼블릭 클라우드 공급업체별로 제공하는 벤더별 인터페이스, 툴, 서비스가 다릅니다.

컨테이너와 클라우드는 자동화를 통해 운영 효율성을 개선할 수 있는 엄청난 잠재력이 있습니다. 또한 컨테이너는 DevOps 사례와 문화를 구현하기에 이상적인 환경입니다. 그러나 호스팅된 애플리케이션이 있는 모든 위치에서 서로 다른 플랫폼을 사용하는 클라우드 전략은 운영자와 개발자가 너무 많은 작업을 추적하고 파악해야 하는 부담을 줄 수 있습니다.

Red Hat의 접근 방식: 어디서나 실현 가능한 클라우드 경험

Red Hat OpenShift®는 자동화된 풀스택 오퍼레이션으로 하이브리드 클라우드 및 멀티클라우드 배포를 관리하는 엔터프라이즈급 쿠버네티스 컨테이너 플랫폼으로, 퍼블릭 클라우드를 간소화하고 자동화합니다. 여기에는 엔터프라이즈급 Linux® 운영 체제, 컨테이너 런타임, 네트워킹, 모니터링, 레지스트리, 인증 및 권한 부여 솔루션이 포함됩니다.

온프레미스 데이터센터 또는 프라이빗 클라우드에 관계없이 원하는 인프라에 Red Hat OpenShift Container Platform을 배포할 수 있고, 인프라 관리의 부담을 덜 수 있도록 대부분의 퍼블릭 클라우드 공급업체는 Red Hat OpenShift를 관리형 서비스로 제공합니다.

일관된 하이브리드 클라우드 기반으로 운영 간소화

Red Hat OpenShift는 클라우드 플랫폼에서 새로운 개발이 이루어질 경우 레거시 애플리케이션을 온프레미스에서 유지해야 할 때 발생하는 문제를 해결하는 데 도움이 됩니다. 기본 클라우드 또는 컨테이너 플랫폼의 세부 정보를 추상화하여 공통 애플리케이션 플랫폼을 생성함으로써 하이브리드 및 멀티클라우드 배포로 손쉽게 전환할 수 있습니다.

기존 애플리케이션과 신규 애플리케이션을 위한 Red Hat OpenShift의 공통 운영 인터페이스는 애플리케이션을 내부 또는 외부에서 실행하든 관계없이 운영을 간소화합니다. 애플리케이션이 어디에서 실행되든 동일한 툴, 콘솔, 절차가 사용되며, 운영자는 익숙해지는 데 걸리는 시간을 줄여 생산성을 더욱 빠르게 높일 수 있습니다. 환경별로 다른 작동 방식을 기억할 필요가 없어, 문제를 더 빨리 진단하고 해결할 수 있습니다.

---

1 *Stackalytics. "Kubernetes," 2019년 12월 6일 액세스. https://www.stackalytics.com/cncf?module=kubernetes.*

공통 애플리케이션 플랫폼은 애플리케이션의 이식성과 배포 유연성을 높여줍니다. 컨테이너에는 여러 컨테이너를 오케스트레이션하여 완전한 애플리케이션을 제공하는 데 필요한 배포 세부 정보가 모두 포함되어 있지는 않습니다. 쿠버네티스는 여러 YAML 파일을 사용해 배포 및 구성 세부 정보를 저장합니다. Red Hat OpenShift가 쿠버네티스에 가치를 더하는 영역 중 하나는 운영자와 개발자가 YAML 파일을 직접 편집할 필요가 없도록 그래픽 사용자 인터페이스(GUI) 및 배포 템플릿을 제공하는 점입니다.

배포 템플릿은 Red Hat OpenShift에서 애플리케이션을 배포하고 OpenShift 클러스터 간에 애플리케이션을 이동하는 프로세스를 간소화합니다. 이 템플릿은 애플리케이션 코드에 포함하거나 별도로 보관할 수 있습니다. 또한 애플리케이션을 Red Hat OpenShift 서비스 카탈로그에 추가해, 애플리케이션 및 소프트웨어 구성 요소를 포인트 앤 클릭(point-and-click) 방식으로 배포할 수 있습니다.

여러 Red Hat OpenShift 클러스터를 관리하기 위해 Red Hat OpenShift 4는 통합 하이브리드 클라우드 콘솔을 도입했습니다. 이 기능은 온프레미스 또는 멀티클라우드에서 실행할 수 있는 클러스터 전반에서 중앙 집중식 관리 및 시각화 툴을 제공합니다.

컨테이너에서 애플리케이션 개발

애플리케이션을 컨테이너로 마이그레이션하기 전에 애플리케이션 코드를 컨테이너 이미지에 빌드해야 합니다. Red Hat OpenShift는 개발자에게 리소스가 프로비저닝될 때까지 기다리지 않고 컨테이너를 빌드 및 실행할 수 있는 셀프 서비스 플랫폼을 제공합니다. 이 영역은 Red Hat OpenShift가 쿠버네티스에 가치를 더하는 주요 영역 중 하나입니다.

Red Hat OpenShift를 통해 개발자들은 지속적 통합/지속적 제공(CI/CD)을 위한 자동화된 빌드를 설정할 수 있습니다. 새 코드가 소스 코드 버전 제어 시스템에 체크인될 때마다 빌드가 자동으로 시작되며, 빌드가 성공적으로 완료되면 자동으로 배포되어 이전 버전을 대체합니다. 이 기능은 자동화된 테스트 및 지속적인 개선을 지원합니다. Red Hat OpenShift는 정교하고 자동화된 빌드 파이프라인을 생성하기 위한 다양한 기능을 제공합니다. 개발자는 복잡하게 처음부터 빌드 환경을 구축하지 않고도 Jenkins와 같은 익숙한 툴을 사용할 수 있습니다.

IT 운영팀은 제어를 유지하고 개발자는 클러스터에 대한 관리 액세스 권한 없이 작업할 수 있습니다. Red Hat OpenShift는 보안을 통해 여러 테넌트를 지원합니다. 빌드를 실행하든 로그인하여 실행 코드를 디버깅하든 관계없이 개발자가 수행하는 모든 태스크는 Red Hat OpenShift를 기반으로 컨테이너 내부에서 실행됩니다. 개발 태스크는 컨테이너에서 실행되기 때문에 다른 컨테이너 및 클러스터 자체와 격리됩니다.

개발자를 위한 툴

Red Hat은 개발자가 컨테이너에서 구동할 애플리케이션을 구축하도록 지원하는 다양한 툴을 제공합니다.

- Red Hat CodeReady Studio는 컨테이너와 멀티플 프로그래밍 모델을 위한 광범위한 툴링 세트를 갖춘 기존 데스크톱 통합 개발 환경(IDE)입니다.
- Red Hat CodeReady Workspaces는 Red Hat OpenShift에서 실행되는 쿠버네티스 네이티브 개발자 워크스페이스 서버로서, 개발자가 로컬 머신에 소프트웨어를 설치하거나 코드를 복사하지 않아도 되는 브라우저 기반 IDE를 제공합니다.
- Red Hat CodeReady Containers는 사전 구성된 최소 Red Hat OpenShift 환경으로, 개발자들이 노트북에서 실행할 수 있는 완전한 독립형 개발 환경을 제공합니다.
- Red Hat Container Catalog는 개발자가 기본 이미지로 사용할 수 있도록 신뢰할 수 있는 소스의 검증된 컨테이너 이미지 라이브러리를 제공합니다.

- Red Hat OpenShift Application Runtimes는 여러 언어 및 프로그래밍 스타일을 지원하는 Red Hat OpenShift 통합 런타임 컬렉션으로, 클라우드 네이티브 개발을 간소화합니다.
- Red Hat Application Migration Toolkit은 개발자가 레거시 애플리케이션의 코드를 평가하여 현재 애플리케이션 서버 및 미들웨어와 같은 최신 플랫폼에서 실행하는 데 필요한 변경 사항을 확인할 수 있도록 도와주는 툴 모음입니다.

레거시 애플리케이션을 컨테이너로 이전

애플리케이션 컨테이너가 빌드되면 애플리케이션 배포를 위한 다음 단계는 스토리지 및 네트워킹을 구성하는 것입니다. 영구적인 스토리지의 필요성을 충족하기 위해 Red Hat OpenShift에서 정의된 애플리케이션은 실행 시 자동으로 애플리케이션의 컨테이너에 연결되는 퍼시스턴트 스토리지 볼륨을 사용하도록 구성할 수 있습니다. 개발자는 운영 팀이 프로비저닝한 스토리지 풀을 이용해 컨테이너 기반 애플리케이션을 위한 탄력적인 스토리지를 관리할 수 있습니다. Red Hat OpenShift Container Storage를 사용하면 소프트웨어 정의 퍼시스턴트 스토리지를 만들 수 있으며, Red Hat OpenShift 클러스터에서 실행되는 애플리케이션에 대한 블록, 파일 또는 오브젝트 액세스 방법을 제공합니다.

컨테이너에서 실행되는 애플리케이션을 위한 가상 프라이빗 네트워킹, 라우팅, 로드 밸런싱은 쿠버네티스 및 Red Hat OpenShift에서 제공하는 플랫폼의 일부로 내장되어 있습니다. 네트워킹은 애플리케이션 배포 구성의 일부로 선언적 방식으로 지정됩니다. 애플리케이션별 네트워크 구성은 소스 코드와 함께 저장되어 코드형 인프라(IaC)가 될 수 있고, 애플리케이션별 인프라 구성을 각 애플리케이션에 연결하면 애플리케이션 배포를 이동, 추가 또는 변경할 때 안정성이 향상됩니다.

소프트웨어 정의 라우팅 및 로드 밸런싱은 애플리케이션이 자동으로 확장 또는 축소되도록 하는 데 중요한 역할을 합니다. 또한 Red Hat OpenShift에서 실행되는 애플리케이션은 롤링 배포를 활용하여 위험을 줄일 수 있습니다. Red Hat OpenShift의 기본 제공 서비스 라우팅을 사용하면 롤링 배포 전략을 통해 일부 사용자 집단을 대상으로 새 코드를 테스트할 수 있습니다. 문제가 발생하면 Red Hat OpenShift에서 컨테이너를 사용하여 이전 버전으로 더욱 손쉽게 롤백할 수 있습니다.

마지막으로, Red Hat OpenShift Service Mesh는 분산형 애플리케이션의 복원력과 성능을 향상시킵니다. OpenShift Service Mesh는 서비스 간 커뮤니케이션의 로직을 전용 인프라 레이어로 추상화하여 더욱 효율적인 커뮤니케이션을 지원하고 분산형 애플리케이션의 복구 능력을 향상시킵니다. OpenShift Service Mesh는 보안 중심의 엔터프라이즈 플랫폼을 기반으로 Istio 서비스 메쉬, Jaeger(추적 기능), Kiali(시각화 기능)를 통합합니다.

애플리케이션 환경 개선

레거시 애플리케이션이 Red Hat OpenShift의 컨테이너에서 실행되면 개선할 수 있는 기회가 생깁니다. CI/CD, 빌드 및 배포 자동화, 자동화된 테스트, 롤링 배포를 사용하여 신뢰성이 향상된 새로운 코드를 더 자주 릴리스할 수 있습니다. 코드를 더욱 자주 릴리스할 수 있다는 것은 조직이 비즈니스 요구 변화에 더욱 효과적으로 대응할 수 있음을 의미합니다.

현대화에 대한 일반적인 접근 방식은 새로운 기술에 구현된 새로운 인터페이스와 서비스를 레거시 시스템보다 우선하는 것입니다. 모든 것을 컨테이너에서 실행할 수 있고 각 컨테이너 내에서 어떤 언어나 기술을 사용하는지 중요하지 않은 경우, 이러한 접근 방식이 훨씬 간편합니다. Red Hat OpenShift의 가상 네트워킹 기능과 서비스 메쉬를 사용하면 애플리케이션 구성 요소를 보다 쉽게 안정적으로 연결할 수 있습니다.

Red Hat OpenShift를 사용해 레거시 애플리케이션과 함께 최신 미들웨어를 더욱 손쉽게 배포할 수도 있습니다. Red Hat은 컨테이너의 OpenShift 클러스터에서 바로 실행할 수 있는 통합 및 메시징 시스템, 비즈니스 프로세스 관리, 의사 결정 관리 소프트웨어를 제공합니다. 이러한 기능을 사용해 애플리케이션을 연결하여 애자일 인테그레이션을 구현할 수 있습니다.

결론

하이브리드 클라우드 및 멀티클라우드에 대한 Red Hat의 접근 방식은 온프레미스 또는 퍼블릭 클라우드에서 실행할 때 모두 기존 애플리케이션과 신규 애플리케이션을 지원하는 공통 애플리케이션 플랫폼을 제공합니다. 그에 따라 애플리케이션 이식성을 확보한 조직은 최적의 환경에서 유연하게 워크로드를 실행할 수 있게 됩니다. 여러 기본 클라우드 및 컨테이너 플랫폼의 세부 정보가 추상화되므로 애플리케이션 실행 위치에 관계없이 운영자와 개발자의 생산성이 향상됩니다.

레거시 애플리케이션을 컨테이너화하고 Red Hat OpenShift에서 기존의 애플리케이션과 신규 애플리케이션을 실행하면 많은 장점이 있습니다. 쿠버네티스 및 OpenShift로 컨테이너 기반 아키텍처를 오케스트레이션하면 애플리케이션의 신뢰성과 확장성을 개선하면서 개발자의 오버헤드를 줄일 수 있습니다. Red Hat OpenShift의 풀스택 자동화, 개발자 셀프 서비스, CI/CD 기능은 또한 지속적인 개선 프로세스를 위한 기반을 제공합니다.

https://www.redhat.com/ko/solutions/hybrid-cloud-infrastructure#scale에서 컨테이너 및 스케일에 따른 컨테이너 실행에 대해 자세히 알아보세요.

한국레드햇 홈페이지 https://www.redhat.com/ko

RED HAT 정보

Red Hat은 세계적인 엔터프라이즈 오픈소스 솔루션 공급업체로서 커뮤니티 기반 접근 방식을 통해 신뢰도 높은 고성능 Linux, 하이브리드 클라우드, 컨테이너, 쿠버네티스 기술을 제공합니다. 또한 고객으로 하여금 신규 및 기존 IT 애플리케이션을 통합하고, 클라우드 네이티브 애플리케이션을 개발하며, 업계를 선도하는 Red Hat의 운영 체제를 기반으로 표준화하는 동시에 복잡한 환경의 자동화, 보안 및 관리를 실현할 수 있도록 지원합니다. Red Hat은 전 세계 고객에게 높은 수준의 지원과 교육 및 컨설팅 서비스를 제공하여 권위 있는 어워드를 다수 수상한 바 있으며, Fortune 선정 500대 기업의 신뢰를 받는 어드바이저로 인정받고 있습니다. 또한 기업, 파트너, 오픈소스 커뮤니티의 전략적인 파트너로서 고객들이 디지털 미래에 대비할 수 있도록 지원하고 있습니다.

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

www.redhat.com/ko
#F18866_0120