

멀티플라우트

인프라를
위한 아키텍처
가이드

멀티클라우드 인프라를 위한 아키텍트 가이드

02

소개

변화하는 기술 환경

- 비즈니스 목표 달성을 위한 인프라 조정

08

1장

컨테이너가 새로운 패러다임으로 불리는 이유

- 클라우드 네이티브 및 기존 애플리케이션을 위한 쿠버네티스

13

2장

IT 및 개발 가속화를 위한 질문

- 엔터프라이즈 아키텍트의 협업이 필수가 된 시대

15

3장

일관성과 보안 구축

- 멀티 컨테이너 기반 애플리케이션 관리를 위한 요구사항

19

4장

Red Hat과 함께 변화에 대응

- 하이브리드 클라우드는 최종 상태가 아닌 가능성의 세계

변화하는 기술 환경

필요할 때 필요한 곳에
애플리케이션을 배포하고 이를
관리하는 능력은 차별화 요소입니다.

거의 모든 업계가 다음과 같은 결과를 기대하며 디지털 인프라를
현대화하고 있습니다.

- 새로운 시장 활용
- 변화하는 고객 기대에 부응
- 심화되는 경쟁 압력에 대응

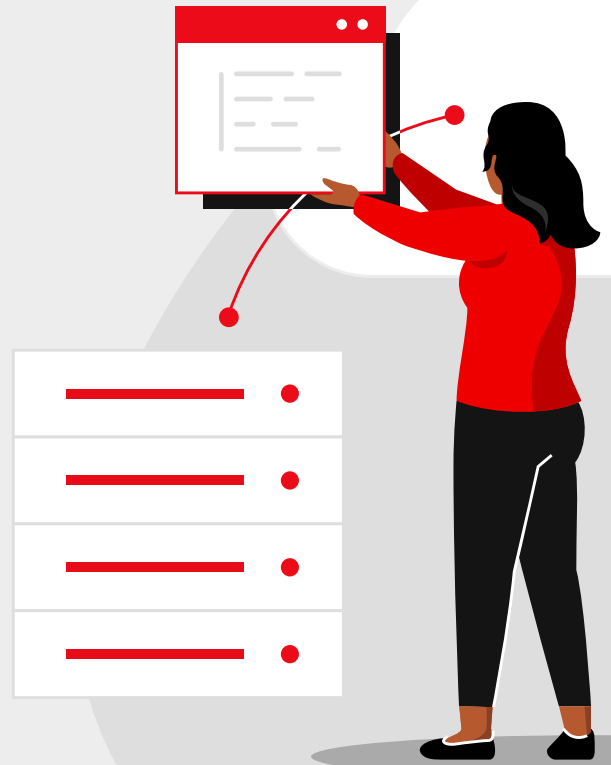
성공을 거두려면 조직은 더욱 빠르고 민첩하고 유연해져야 하며, IT
인프라는 계속해서 중요한 역할을 해야 합니다.

57

의 비즈니스 및 서비스가
2025년까지 디지털 인프라로
제공될 것으로 전망됩니다!

인프라를 현대화하기 위한 이와 같은 움직임은 고객이 있는 곳에 애플리케이션을 도입하려는 노력에서 비롯된 것입니다.

엔터프라이즈 아키텍트로서 여러분은 끊임없이 진화하는 애플리케이션 포트폴리오 전반에서 점차 아키텍처, 기술, 프레임워크가 결합되어 구축된 것을 보았을 것입니다.



IDC에 따르면, 약

50%

의 컨테이너화된 애플리케이션만이 신규 애플리케이션이며,

나머지 50%는 확장성과 이식성을 개선하기 위한 컨테이너 플랫폼으로 리팩토링 또는 리프트 앤 시프트된 기존 애플리케이션입니다.¹

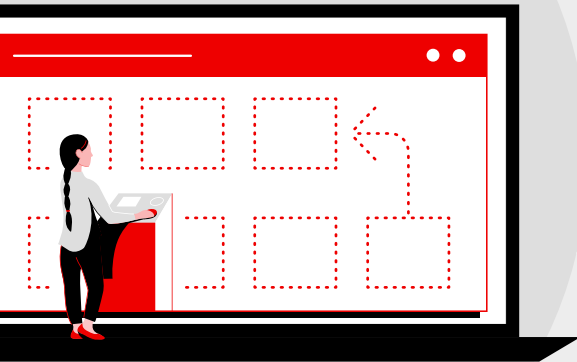
이러한 애플리케이션은 다음과 같은 여러 인프라 전반에서 실행되고 있거나 실행되어야 합니다.

- 데이터센터 전반에 걸쳐 있는 인프라
- 하나 이상의 퍼블릭 클라우드로 확장된 인프라
- 엣지로 확장된 인프라

¹ IDC Technology Spotlight, Red Hat 후원. "디지털 비즈니스 성공에 필수적인 효율적인 멀티클러스터 쿠버네티스 관리(Digital Business Success Depends on Effective Multicloud Kubernetes Management)". IDC, #US47479221, 2021년 2월.

이와 같이 애플리케이션과 인프라가 혼합된 하이브리드 솔루션이 증가하면서 복잡성도 늘어나고 있습니다.

관리하기 어려운 환경과 수동 프로세스로 인해, 운영 팀은 개발자가 필요로 하는 속도에 맞춰 리소스와 서비스를 제공하는 데 어려움을 겪고 있습니다.



✓ 엔터프라이즈 아키텍트는 혁신을 주도하며 인프라 제어, 보안, 안정성을 유지할 수 있도록 지원해드립니다. 성공을 거두려면 유연하고 확장 가능한 모듈식 IT 환경을 반드시 구축해야 합니다.

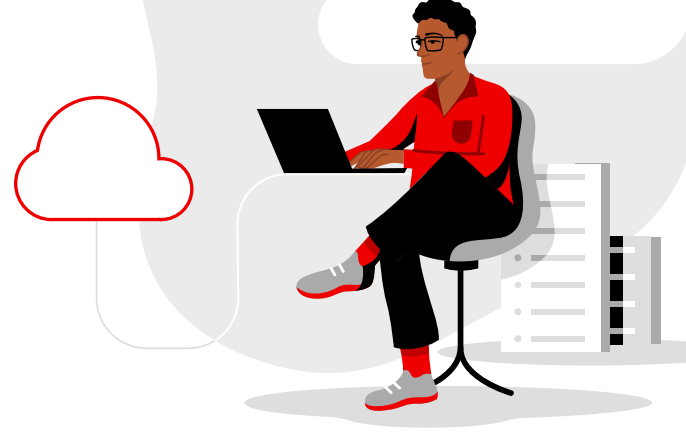
핵심 개념

오늘날 논의 중인 다양한 최신 클라우드 인프라와 관련하여 몇 가지 중요한 아키텍처 고려 사항이 존재하며, 이는 종종 혼용되기도 합니다. 다음 요소들이 모두 함께 민첩성과 이식성을 지원하므로 IT 팀은 개발자 및 비즈니스 팀에 더 빠르게 서비스를 제공할 수 있습니다.

- 1 하이브리드 클라우드
- 2 멀티클라우드
- 3 컨테이너
- 4 쿠버네티스 기술



아키텍트는 핵심 개념에 대한
공통된 이해를 바탕으로 모두가
협력하도록 만드는 전략적
우위를 갖고 있습니다.



1 하이브리드 클라우드

하이브리드 클라우드는 환경에 상관없이 동일한 방법론, 워크플로우, 기술을 사용합니다. 여기에는 퍼블릭 및 프라이빗 클라우드, 온프레미스 데이터센터가 포함되며 옛지까지도 확장할 수 있습니다. 또한 연결되어 있는 단일 환경에 대해 이러한 리소스를 결합하도록 지원합니다.

2 멀티클라우드

멀티클라우드는 여러 프라이빗 또는 퍼블릭 클라우드 공급업체에 호스팅되어 서로 연결된 단일 서비스 세트를 제공하는 접근 방식으로, 클라우드 간에 사전 정의된 상호 의존성이 없습니다.

3 컨테이너

컨테이너를 통해 실행에 필요한 모든 파일을 포함하여 전체 실행(runtime) 환경에서 애플리케이션을 패키징하고 분리할 수 있습니다.

4 쿠버네티스

쿠버네티스는 필요한 만큼 자동으로 규모를 확장하거나 축소하는 등의 컨테이너 운영을 오케스트레이션 및 자동화하여 다수의 수동 프로세스를 줄이는 오픈소스 플랫폼입니다.

비즈니스 목표

달성을 위한 인프라 조정

고객과 디지털 관계를 구축하는 것은 조직이 앞서가는 데 있어서 스스로를 차별화할 수 있는 방법 중 하나입니다.

이러한 접근 방식은 새로운 애플리케이션을 만들거나 기존 애플리케이션을 현대화한다는 것을 의미합니다. 조직이 애플리케이션을 더욱 빠르게 제공하려고 노력할 때 엔터프라이즈 아키텍트는 고객에서 사용자로 사고 방식을 전환하는 데 중요한 역할을 하게 될 것입니다.

예를 들면 다음과 같습니다.

고객 유치율은 **단일 연도 예약(SYB)**으로 측정할 수 있습니다. 단일 연도 예약은 일반적으로 매출 관련 지표이지만, 디지털 관계를 구축하는 것은 개별 트랜잭션 이상의 의미를 지닙니다.

많은 조직이 **서비스로서의 소프트웨어(SaaS)** 관점으로 전환하며 고객 요구사항 변화와 새로운 기회에 대응하기 위해 **일일 활성 사용자(DAU)**에 초점을 맞추고 있습니다. 엔터프라이즈 아키텍트는 그에 따라 대응하기 위해 기존 및 신규 인프라를 적절하게 결합할 수 있는 방법을 파악해야 합니다.

IDC에 따르면

97%

의 기업이 클라우드 네이티브 애플리케이션을 지원하기 위해 온프레미스 리소스 및 한 개 이상의 퍼블릭 클라우드

플랫폼에 **연결된 하이브리드 클라우드 및 멀티클라우드 인프라**를 활용할 것으로 예측합니다.¹



¹ IDC Technology Spotlight, Red Hat 후원. "디지털 비즈니스 성공에 필수적인 효율적인 멀티클러스터 쿠버네티스 관리(Digital Business Success Depends on Effective Multicloud Kubernetes Management)". IDC, #US47479221, 2021년 2월.

필연적이고 지속적인 변화

우리는 새로운 기회에 적응하고 대응하기 위해 조직에 유연성이 필요하다는 이야기를 항상 듣지만, 최신 기술을 무작정 도입한다고 해서 장기적인 성공이 보장되는 것은 아닙니다.

예를 들어, 고객 소프트웨어 애플리케이션은 경쟁 우위 가능성을 갖춘 핵심 영역으로 급부상하고 있지만, 비용 증가와 더불어 IT 팀에 복잡성과 부담을 줄 수도 있습니다.



이 e-book에서는 조직에 적합한 전략을 구축하기 위한 계획을 세울 수 있도록 하이브리드 클라우드 접근 방식의 핵심 구성 요소, 이점 및 장단점을 살펴봅니다.

컨테이너가 새로운 패러다임으로 불리는 이유



현재와 미래를 위한 컨테이너의 이해

컨테이너는 실행에 필요한 모든 파일을 포함하여 전체 실행(runtime) 환경에서 애플리케이션을 패키징하고 분리할 수 있는 Linux® 하위 시스템을 사용합니다. 따라서 전체 기능을 유지하면서 컨테이너화된 애플리케이션을 환경 간에 쉽게 이동할 수 있게 해줍니다.

컨테이너를 활용해 애플리케이션을 배포하면 다음이 가능해집니다.

- 배포 속도 향상
- 워크로드 이식성 증가
- 인프라, IT 팀, 프로세스의 부담 완화

컨테이너는 애플리케이션이 실행 환경에서 추상화되도록 함으로써 프라이빗 데이터센터, 퍼블릭 클라우드 또는 엣지 기기 등 모든 환경 전반에서 간편하고 일관되게 애플리케이션을 배포하도록 지원합니다.

올인원 아키텍처 솔루션이라 생각될 수도 있으나 조직의 장기적인 계획에 컨테이너를 통합하려면 몇 가지 사항을 더 고려해야 합니다.

널리 보급된 컨테이너

컨테이너 기술은 십년 넘게 오래 사용되어 왔지만, 최근에 와서야 조직이 규모에 적합한 민첩성과 회복 탄력성을 갖추기 위한 주요 방법으로 점차 인정받고 있습니다.

그 이유는 무엇일까요?



1 주요 차별화 요소가 된 규모

가상 머신(VM)에 비해 컨테이너는 필요한 최소한의 소프트웨어 패키지만 포함해 훨씬 더 작아질 수 있으며 몇 분 단위가 아닌 수백 밀리초 내에 시작할 수 있습니다.

따라서 컨테이너는 지속적인 배포보다는 애플리케이션이 필요한 특정 시점에 이를 배포하고 실행할 수 있어, 컨테이너를 사용하는 팀의 민첩성이 크게 높아집니다.

필요에 따라 서비스를 제공할 수 있다는 점도 팀이 기본 하드웨어를 사용하여 컨테이너 기반 솔루션이 도달할 수 있는 전반적인 규모를 향상시키는 데 도움이 됩니다.

2 인프라 전반에서 이식 가능한 컨테이너

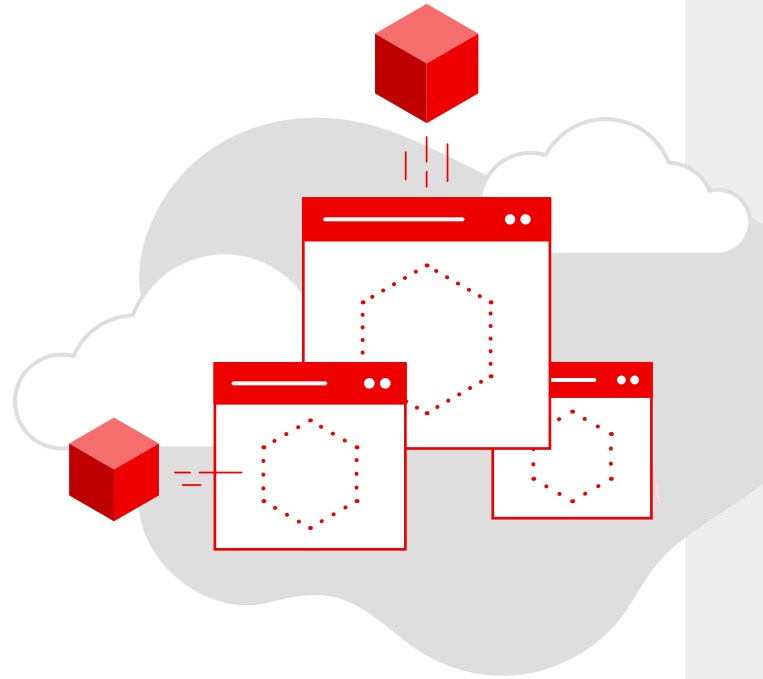
VM 기반의 아키텍처에서 아키텍트는 여러 클라우드 전체에 배포해 리스크를 차단할 수 있을지는 몰라도, 상호 운용이 불가능한 여러 VM 형식으로 인한 문제점에 부딪히게 될 수 있습니다.

물론 대안이 있기는 하지만 팀이 해결해야 할 복잡성을 추가할 뿐입니다.

컨테이너는 "한번 구축하면 어디에나 배포 가능"한 설계로 운영 팀이 여러 클라우드 공급업체에서 애플리케이션을 실행할 수 있도록 합니다.

컨테이너 오케스트레이션 엔진(쿠버네티스)이 있기만 하면 됩니다. 하지만 컨테이너화를 도입하기 전에 고려해야 할 몇 가지 장단점이 있습니다.

주요 장점과 고려 사항은 다음과 같습니다.



장점

☐ 배포 시간 단축

컨테이너는 호스트 환경에서 애플리케이션을 격리하여 VM보다 배포 시간을 단축하고, 짧은 시작 시간으로 더 큰 다양성을 제공합니다.

☐ 이식성 향상

컨테이너는 애플리케이션을 격리시키기 때문에 다양한 플랫폼, 다양한 클라우드 벤더 간 이식성이 매우 뛰어납니다. 실행하기 위한 Linux 운영 체제만 있으면 됩니다.

☐ 일관된 운영

DevOps 팀은 배포 위치와 관계없이 동일한 방식으로 애플리케이션을 실행할 수 있습니다.

확장성 향상

필요에 따라 여러 컨테이너와 동시에 실행하여 애플리케이션의 확장성을 높입니다. 전체 애플리케이션에 영향을 주거나 서버를 더 추가할 필요없이 추가 컨테이너를 수초 내에 배포할 수 있습니다.



더 나은 애플리케이션 개발

개발, 테스트, 프로덕션 주기를 가속화하고자 하는 DevOps 팀의 노력과 민첩성을 지원합니다.

고려 사항

보안

컨테이너는 일반적으로 계층이라고 불리는 여러 수준을 포함하여 또다른 컨테이너를 기반으로 하는 것으로 정의됩니다. 보안을 위해 컨테이너의 기반이 되는 모든 계층은 반드시 감사를 거쳐 컴플라이언스를 보장해야 합니다. 기존 스택에 비해 여기에는 더 많은 작업이 필요합니다.

도입

도입 시간을 단축하고 컨테이너의 모든 장점을 얻기 위해 조직은 도입 시점부터 컨테이너를 운영하기 위한 기술을 갖추어야 합니다. 이는 직원 교육이나 새로운 전문가 고용이 필요하다는 뜻입니다.

장기적인 유지 관리

귀사가 아직 퍼블릭 클라우드를 도입할 준비가 되지 않았더라도, 온프레미스 인프라는 컨테이너화된 애플리케이션과 같은 방식으로 규모를 확장하거나 축소할 수 있어야 하며 이는 유지 관리에 영향을 줄 수 있습니다.

운영의 미래

클라우드 네이티브 기능을 충분히 활용하고 각기 다른 퍼블릭 클라우드에서 컨테이너를 실행하려면 각각의 클라우드를 관리하고 유지할 별도의 사이트 신뢰성 엔지니어링(SRE) 팀이 필요할 수 있으며 이는 많은 비용을 유발합니다.

클라우드 네이티브 및 기존 애플리케이션을 위한 쿠버네티스

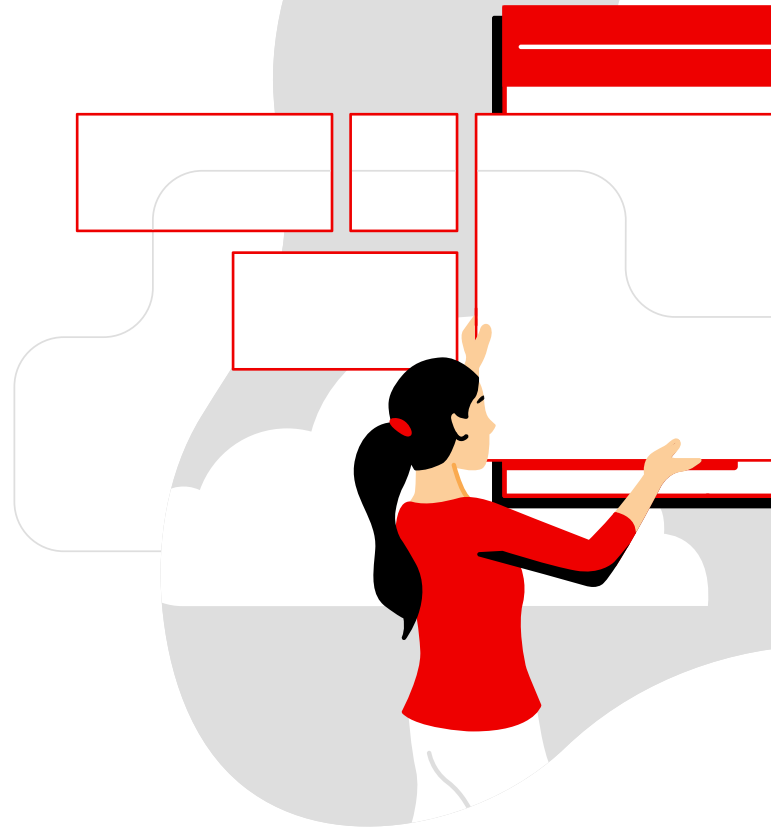
대부분의 엔터프라이즈 아키텍트가 해결해야 할 과제는 애플리케이션 제공 속도를 높이는 데 그치지 않습니다. 레거시 IT를 새로운 솔루션과 통합하여 조직의 핵심적인 기존 애플리케이션을 유지하고 모든 프로세스가 원활하게 작동하게 해야 합니다.

쿠버네티스 정의

쿠버네티스는 컨테이너화된 애플리케이션을 배포, 관리, 스케일링하는 데 필요한 대부분의 수동 프로세스를 자동화하는 오픈소스 컨테이너 오케스트레이션 플랫폼입니다.

사실상 컨테이너를 관리하고 배포하는 컨트롤 플레인으로 여겨지는 쿠버네티스는 컨테이너화된 기존 클라우드 네이티브 앱을 규모에 맞게 제공하고 관리하도록 지원하기도 합니다.

컨테이너가 새로운 복잡성을 야기하는 것처럼 보일 수 있지만 쿠버네티스는 자동화를 적용하여 운영을 간소화합니다.



IT 및 개발

가속화를 위한 질문

디지털 인프라를 최대한
활용하려면 엔터프라이즈
아키텍트는 질문하고 모든 부서가
협력할 수 있도록 해야 합니다.

엔터프라이즈 아키텍트가 향후 요구사항을 고려하여 기존 인프라를 극대화할 전략을 개발하며 적절한 질문을 던지는 것은 복잡한 인프라 문제를 관리하고 해결하는 데 필요한 정보를 제공해줄 수 있습니다.



엔터프라이즈 아키텍트의 협업이 필수가 된 시대

새로운 디지털 인프라를 위한 기반을 마련하기 전에
애플리케이션 포트폴리오의 현재 상황을 평가하기 위한 핵심
질문을 통해 이와 관련한 목표를 이해하는 것이 중요합니다.

다음 질문으로 시작하는 것이 좋습니다.

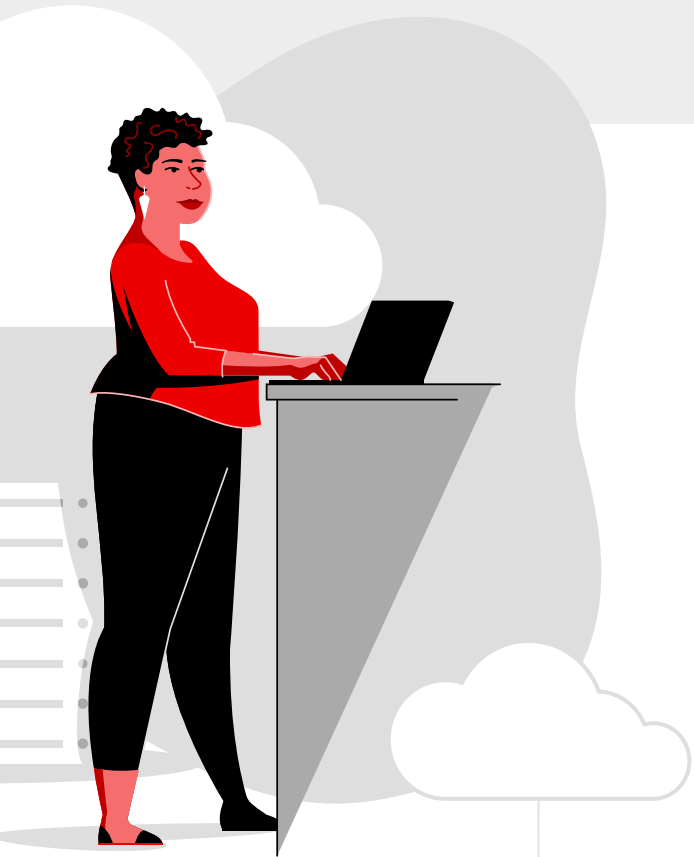
- 애플리케이션을 통해 해결하려는 비즈니스 문제는 무엇인가?
- 최종 사용자는 누구인가?
- 관련 데이터는 어디에 저장되어 있는가?
- 애플리케이션이 액세스하려는 데이터는 특별 보안 또는 법적 요건이 필요한가?
- 누가 애플리케이션에 액세스해야 하는가?
- 애플리케이션은 다수의 위치에서 실행되는가 아니면 데이터센터, 클라우드 또는 엣지 등 각기 다른 위치에서 실행되는가?
- 애플리케이션이 실행하려는 작업과 관련한 보안 모델은 무엇인가?



특히 이러한 질문에 답하려면 인프라로 대화를 시작할 수는 없습니다. 그보다는 IT 팀 및 여러 사업부 전반의 고위 관리자들을 한자리에 모아 애플리케이션을 통해 실현하려는 비즈니스 목표를 이해해야 합니다.

애플리케이션의 작동 방식과 목표를 파악했으면

최적의 인프라 설정 또는 조정 방식에 대해 합리적인 의사결정을 내릴 수 있습니다.



일관성과 보안 구축

증가하는 애플리케이션 포트폴리오 관리

컨테이너는 퍼블릭 클라우드 공급업체와 관련하여 다양한 선택 옵션을 제시합니다. 컨테이너가 전체 실행 환경을 통해 애플리케이션을 패키징하고 격리하면 사용자는 이렇게 컨테이너화된 애플리케이션을 여러 클라우드 간에 이동하면서도 모든 기능을 그대로 유지하고 일관성과 보안을 강화할 수 있습니다.

따라서 독점 솔루션의 제약으로 인한 워크로드 지원 여부가 아니라 업타임, 스토리지 공간, 비용 등의 범용 표준을 기준으로 퍼블릭 클라우드 공급업체를 자유롭게 선택할 수 있습니다.

마이크로서비스

마이크로서비스란 소프트웨어 작성을 위한 하나의 아키텍처 접근 방식으로, 애플리케이션을 상호 독립적인 최소 구성 요소로 분할합니다. 컨테이너와 함께 클라우드로 애플리케이션을 옮길 수 있게 해줍니다.

운영의 미래

컨테이너를 곧바로 도입하기 전에 고려해야 할 몇 가지 장단점이 있습니다.

컨테이너를 도입하거나 확장하기 전에 향후 운영 팀의 역할에 관한 질문을 반드시 해봐야 합니다.

모든 퍼블릭 클라우드에서 동일한 경험을 제공할 수 있다고 가정할 때 인프라 전략이 잘못된 방향으로 갈 수 있습니다.

흔히 쿠버네티스를 AWS, Google Cloud Platform, Microsoft Azure에서 실행할 경우 모두 쿠버네티스이기 때문에 한 팀이 전체를 관리해도 괜찮다고 오해합니다. 하지만 항상 그렇지 않습니다.

퍼블릭 클라우드 간에 미묘한 차이가 있을 수 있습니다.

부하 분산, 백업, 로깅 등은 각 퍼블릭 클라우드에 대한 고유한 아키텍처를 필요로 하며, 쿠버네티스 버전과 기능도 클라우드 공급업체에 따라 매우 다를 수 있습니다.

Red Hat® OpenShift®와 같은 추상화 계층이 없는 경우, 별도의 SRE 팀이 각 환경을 관리해야 하며 이는 비용을 기하급수적으로 높이게 됩니다.



멀티 컨테이너 기반 애플리케이션 관리를 위한 요구사항

- 엔터프라이즈 레디(enterprise-ready) 인프라
- 마스터와 노드에 충분한 머신
- 메시징 서버

Red Hat OpenShift

자동화된 풀스택 운영으로 하이브리드 클라우드, 멀티클라우드 및 엣지 배포를 관리하는 보안 중심의 컨테이너 플랫폼입니다. Red Hat OpenShift에는 다음과 같은 장점이 있습니다.



1 일관성

개발자와 운영 팀이 애플리케이션을 공통된 방식으로 패키징, 배포, 관리할 수 있도록 인프라 전반에서 공통된 추상화 계층을 제공합니다.

2 속도

개발자 툴, 다양한 코딩 언어, 데이터 및 스토리지 서비스, 애플리케이션 제공 자동화와 DevOps 프로세스 지원을 위한 전체 지속적 통합/지속적 제공(CI/CD) 서비스에 대한 셀프 서비스 액세스를 통해 더욱 신속하게 애플리케이션을 빌드, 테스트, 배포, 실행합니다.



3 보안

그 결과, 운영 체제에서 애플리케이션까지, 그리고 소프트웨어 라이프사이클과 공급망 전반에서 지속적인 풀스택 보안을 제공합니다. 게다가 빌트인 인증과 권한 부여, 암호 관리, 감사, 로깅 및 통합 컨테이너 레지스트리를 사용하여 리소스와 사용자 권한을 세부적으로 제어할 수 있습니다.

4 관리

시스템 관리자와 운영 팀은 애플리케이션, 가상 머신, 컨테이너를 단일 컨트롤 플레인에서 관리할 수 있습니다. 일반적인 관리 플랫폼의 경우, 관리자와 개발자가 여러 팀의 클러스터, 서비스, 역할을 중앙 관리 콘솔에서 관리합니다.

5 확장

고도화된 관리 및 자동화 기능을 통해 애플리케이션을 빠르고 효율적으로 확장하세요. Red Hat OpenShift는 컨테이너를 대규모로 관리할 수 있도록 컨트롤 플레인을 제공하고 혁신적인 컨테이너 라이프사이클 관리 기능을 제공합니다.

Red Hat과 함께 변화에 대응

하이브리드 클라우드는 최종 상태가 아니라 변화에 대응할 수 있는 역량을 의미합니다.

오늘날 더욱 빠르게 솔루션을 제공하고 미래를 위한 디지털 인프라를 설정하려면 엔터프라이즈 아키텍트는 새로운 솔루션과 기존 기술을 통합하여 비용을 관리해야 합니다.

하지만 전면적인 클라우드 마이그레이션이 그리 간단하지 않다는 것이 대다수 조직의 현실입니다.

이런 측면에서, IT의 미래는 바로 하이브리드 클라우드입니다.



각 워크로드에 적합한 퍼블릭 또는 프라이빗 클라우드를 결정하는 전략을 세우는 과정은 복잡하며, 지금의 계획이 나중에도 항상 적합한 것은 아닙니다.

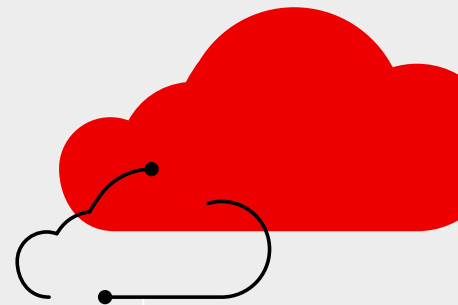
특히 앞으로 변화가 불가피한 경우 새로운 클라우드, 툴, 통합을 아우르는 장기적인 역량을 유지하려면 하나의 클라우드나 하나의 벤더에만 머물러 있어서는 안 됩니다. 만능 솔루션은 존재하지 않기 때문입니다.

핵심은 바로 클라우드 리소스를 사용하는 최적의 전략과 이러한 전략에 변화가 필요할 때 적응할 수 있는 유연성을 결합할 수 있도록 조직에 알맞은 접근 방식을 구축하는 것입니다.

이와 같은 적응력을 유지하기 위해서는 원하는 모든 환경에서 일관된 유연한 플랫폼이 필요하며, 다양한 애플리케이션 출시를 지원할 수 있는 통합, 데이터, 분석 등의 서비스를 갖춰야 합니다.

하이브리드 클라우드 접근 방식은 팀이 새로운 애플리케이션을 개발하고 스케일링하면서 기존 시스템의 이점을 그대로 누릴 수 있도록 합니다.

그러나 분리된 IT 환경을 관리하다 보면 기술 및 프로세스에 예기치 못한 어려움이 발생할 수 있습니다. 신뢰할 수 있는 파트너와 함께 하이브리드 클라우드 전략을 수립하여 변화하는 비즈니스 요구사항을 지속적으로 지원하기 위한 모범 사례도 배울 수 있습니다.



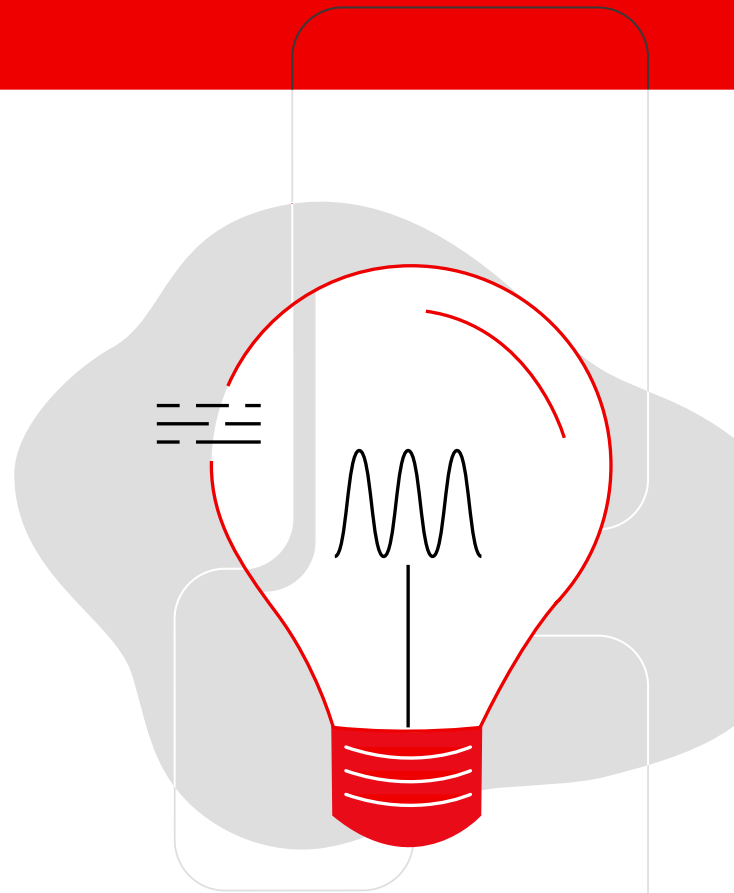


하이브리드 클라우드 접근 방식을 통해 IT 조직 전반에서 다음과 같은 작업을 수행할 수 있습니다.

- 모든 클라우드 또는 팟프린트에서 워크로드를 실행할 수 있는 하이브리드 클라우드 인프라 구축
- 애플리케이션을 더 빠르고 쉽게 제공하기 위한 클라우드 네이티브 개발 도입
- 스케일에 따라 프로세스를 향상하기 위한 자동화
- 하이브리드 클라우드 접근 방식을 최대한 활용하기 위한 팀 프로세스 통합

이러한 주요 측면을 결합하면 모든 인프라 상에서 다양한 워크로드를 실행할 수 있는 일관된 플랫폼을 제공하고, 관리 및 자동화 기능을 통합하며, 비용이 많이 드는 리팩토링이나 재교육 없이도 퍼블릭 클라우드 공급업체를 변경하거나 추가할 수 있습니다. 그리고 어떤 독점 소프트웨어를 사용하더라도 조직의 유연한 오픈 표준에 연결이 가능합니다.

결국 적합한 하이브리드 클라우드 인프라 전략은 요구사항과 목표를 기반으로 조정 및 적응할 수 있는 역량을 조직에 제공해야 합니다.



멀티클라우드

를 인프라를 인프라를 인프라를 인프라를
아키텍트 위한 아키텍트 위한 아키텍트
가이드 가이드 가이드 가이드 가이드

자세한 정보가 필요하신가요?

IT의 미래를 설계하고 구축하도록 Red Hat이 어떤 지원을 제공하는지 살펴보세요.

조직에 적합한 솔루션은 무엇인지 평가해보세요.
[Red Hat 제품 체험판 센터를 방문하세요.](#)



F29609/0821

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.