



Java 애플리케이션 현대화 여정을 위한 계획 수립

성공을 위한 실용적인 안내서

Java 애플리케이션 현대화 여정을 위한 계획 수립

02

소개

비즈니스 애플리케이션을 현대화해야 하는 이유

지속적인 변화에 필요한 것은 지속적인 적응

07

1장

애플리케이션 현대화 여정을 위한 계획 수립

요구 사항에 적합한 경로 선택

12

2장

오픈소스 전문가를 통한 현대화

Red Hat과 함께 성공을 위한 기반 배포

17

3장

실제 고객 사례

실제 비즈니스 성과 살펴보기

19

결론

시작하기

현대화 여정을 시작할 준비가 되셨나요?

지속적인 변화에 필요한 것은 지속적인 적응

오늘날 기업은 항상 변화와 불확실성의 문제를 안고 있습니다.

기업의 성공은 끊임없이 적응하고 혁신하며 유용하고 새로운 제품과 서비스를 고객에게 제공하는 역량에 좌우됩니다. 핵심 비즈니스 애플리케이션과 IT 인프라는 이러한 목표를 달성하는 데 중요한 역할을 하며, 주요 경쟁력 차별화 요인이 될 수 있습니다.

동시에 기존 인프라에 대한 시간, 예산, 교육 관련 투자의 가치를 극대화해야 합니다. 하지만 이러한 시스템을 유지관리하다 보면 혁신에 사용될 수 있는 리소스를 사용하게 되기도 합니다. 실제 연구에 따르면 IT 조직은 현재 인프라와 애플리케이션을 운영하는 데 평균적으로 예산의 64%를 지출하며, 단 36%만 혁신과 성장 이니셔티브에 지출합니다.¹

애플리케이션을 현대화하면 리소스를 혁신을 위해 사용할 수 있습니다.

기존 및 레거시 애플리케이션을 지원하고 유지관리하는 데는 일반적으로 상당한 시간, 노력, 비용이 요구됩니다. 이러한 애플리케이션은 특유의 아키텍처로 인해 비즈니스와 고객 요구 사항 변화에 쉽게 또는 즉시 대응하기가 어렵습니다. 게다가 대부분 타사 시스템과 완전히 통합되지 않기 때문에 파트너십을 통해 새로운 서비스를 추가하는 데에도 한계가 있습니다.

기존 애플리케이션을 현대화하면 보유 중인 제품을 더 효율적으로 활용하고, 시간과 예산을 전략적 프로젝트, 기능 개선, 비즈니스 이니셔티브를 지원하는 새로운 애플리케이션을 위해 사용할 수 있습니다. 클라우드 네이티브 및 마이크로서비스 아키텍처와 접근 방식은 효율성, 유연성, 확장성을 제공하여, 애플리케이션 민첩성과 혁신을 강화하고 시간이 지날수록 더 손쉽게 조정할 수 있도록 합니다.

그렇다고 하더라도 이러한 기술을 도입하기란 만만치 않으며, 면밀한 계획과 시간, 리소스가 필요합니다. 따라서 대부분의 조직이 애플리케이션 현대화를 단일 이벤트가 아닌 지속적인 여정으로 받아들이고 있습니다.

이 e-book에서는 애플리케이션 포트폴리오 현대화의 장점을 다루고, 현대화 여정을 계획 및 구현하기 위한 가이드를 제공합니다.

현대적인 하이브리드 클라우드 플랫폼으로 혁신

애플리케이션 현대화와 혁신을 지원하기 위해 하이브리드 클라우드 플랫폼을 도입하는 조직은 다음을 경험할 수 있습니다.

20%

높아진 DevOps 및 개발팀 생산성.²

21%

향상된 IT 인프라 팀 효율성.²

636%

증대된 5년간 ROI(투자수익률).²

\$2160만

연 매출 증대.²

¹ Flexera. "2021년 기술 지출 현황 Flexera 리포트(2021 Flexera State of Tech Spend Report)," 2021년 1월.

² IDC 백서, Red Hat 후원. "Red Hat OpenShift의 비즈니스 가치." 문서 #US47539121, 2021년 3월

애플리케이션 현대화의 장점



민첩성

시장 상황과 고객 요구 사항 변화에 효율적으로 더 손쉽게 대응

- ▶ 신속하고 뛰어난 고품질의 고객 경험을 제공해 경쟁력을 유지하세요.
- ▶ 새로운 매출 기회를 활용하세요.



속도

더 빠르게 혁신적이고 새로운 애플리케이션, 서비스, 기능 제공

- ▶ 변경 리드 타임을 줄이고 애플리케이션과 서비스 배포 빈도는 늘리세요.
- ▶ 개발자 생산성을 개선하고 비즈니스 가치를 제공하는 데 집중하세요.



비용

운영, 개발, 인프라 비용을 비롯하여 기술 부채를 줄입니다.

- ▶ 애플리케이션과 개별 서비스의 규모를 빠르고, 효과적이며, 비용 효율적으로 확장하세요.
- ▶ IT 리소스를 최적화하여 복잡성과 비용을 줄이세요.



보안

IT와 애플리케이션 보안을 강화하고 민감한 데이터 보호

- ▶ 새로운 기술, 통합, 접근 방식을 통해 보안 상태를 평가하고 조정하세요.
- ▶ 자동화를 적용하여 지속적인 보안 정책 준수를 보장하세요.



신뢰성

애플리케이션 복구 능력과 인프라 안정성 향상

- ▶ 애플리케이션과 서비스 장애의 평균 복구 시간(MTTR)을 단축하세요.
- ▶ 애플리케이션 변경 성공률을 높여 안정성을 개선하세요.

성공적인 현대화의 필수 요소

애플리케이션을 현대화하려면 새로운 기술만으로는 부족합니다. 현대화 작업의 장점을 극대화하려면 기술과 플랫폼을 운영 프로세스 및 조직 문화와 결합하는 전체적인 접근 방식이 필요합니다.

문화

직원들이 솔직하게 아이디어를 공유하고 함께 문제를 해결하도록 역량 강화

구성원은 모든 IT 및 비즈니스 이니셔티브의 핵심입니다. 사업부, 네트워크, 보안, 운영, 개발, 인프라 그룹을 비롯한 모든 팀이 현대화 노력에 자유롭게 참여하고 협업해야 합니다.

프로세스

협업과 커뮤니티 문화를 강화하는 프로세스로 혁신 촉진

프로세스는 조직 내에서 처음부터 끝까지 프로젝트를 진행시킵니다. 애플리케이션 생성, 배포, 관리, 조정을 위한 명확하고 합의된 프로세스와 모범 사례는 현대화 이니셔티브에 무엇보다 중요합니다.

기술

규모, 속도, 유연성을 제공하는 오픈 하이브리드 클라우드 기반 구축

조직의 기본 IT 플랫폼은 애플리케이션을 빌드, 실행, 및 관리하는 기능을 제공합니다. 여러 세대의 애플리케이션을 지원하는 통합 하이브리드 클라우드 플랫폼은 원하는 속도로 현대화하는 데 필요한 유연성과 확장성을 제공할 것입니다.



// 현대화의 핵심 개념과 기술

애플리케이션 현대화는 여러 가지 주요 기술과 접근 방식을 통합합니다. 모든 현대화 여정에 필수인 것도 있고 특정 시나리오에만 필요한 것도 있습니다.



클라우드 네이티브 접근 방식

클라우드 네이티브 기술은 조직이 퍼블릭, 프라이빗, 하이브리드 클라우드와 같은 현대적이고 역동적인 클라우드 환경에서 확장 가능한 애플리케이션을 개발하고 실행하도록 지원합니다. 조직 전반에 클라우드 네이티브 접근 방식을 적용하여 변화하는 비즈니스와 시장 상황에 따라 애플리케이션과 서비스를 더욱 신속하게 개발, 배포, 개선, 확장할 수 있습니다.



마이크로서비스

마이크로서비스 아키텍처는 애플리케이션을 여러 기능으로 구분합니다. 이러한 기능적인 마이크로서비스는 독립적이지만 기존의 모놀리식 애플리케이션과 동일한 태스크를 수행합니다. 마이크로서비스는 일반적으로 경량이며, 독립적으로 확장이 가능하고, 여러 애플리케이션 전반에서 공유가 가능합니다. 마이크로서비스 아키텍처는 클라우드 네이티브 접근 방식과 잘 맞으며 애플리케이션 현대화 이니셔티브 과정에서 도입되는 경우도 종종 있습니다.



DevSecOps

DevSecOps 또는 **DevOps**는 비즈니스 가치와 민첩성을 높이기 위해 고품질의 애플리케이션 제공 속도를 높이는 문화, 자동화, 플랫폼 설계에 대한 하나의 접근 방식을 의미합니다. 개발 및 배포 프로세스 속도를 높이기 위해 팀, 신속한 운영, 표준화된 환경 전반의 협업에 초점을 맞추고 있습니다. DevSecOps를 통해 보안, 개발, 운영 팀의 인력이 협력하여 가시성, 피드백, 모범 사례, 인사이트를 공유합니다.



이벤트 기반 아키텍처

이벤트 기반 아키텍처는 요청보다는 이벤트의 캡처, 커뮤니케이션, 처리, 지속성을 중심으로 구성된 애플리케이션 설계 모델입니다. 이벤트 기반 애플리케이션은 최소한의 결합을 요구하기 때문에 현대적인 분산형 환경과 아키텍처에서도 잘 작동합니다. 개발자는 반응형 프로그래밍 언어로 이벤트 기반 애플리케이션을 생성하여 더 많은 유연성과 옵션을 갖출 수 있습니다.



하이브리드 클라우드 애플리케이션 플랫폼

하이브리드 클라우드 플랫폼은 데이터센터, 엣지, 클라우드 환경에서 툴, 애플리케이션 및 모델을 개발, 배포, 관리할 수 있는 통합 소프트웨어 기반을 제공합니다. 주요 기능은 다음과 같습니다.

- ▶ 기존의 가상화된 애플리케이션, 클라우드 네이티브 애플리케이션, 컨테이너화된 애플리케이션 지원
- ▶ 통합 DevOps 및 운영 관리 기능
- ▶ IT 제어 셸프 서비스 리소스 제공
- ▶ 애플리케이션 개발 툴
- ▶ 여러 유형의 클라우드 환경 전반의 애플리케이션과 워크로드 이식성
- ▶ 타사 기술 벤더와 오픈소스 커뮤니티와의 통합

대부분의 애플리케이션 현대화 여정에서 포괄적이고 조정 가능한 하이브리드 클라우드 플랫폼은 매우 중요하며 원하는 속도로 더 효율적으로 혁신하도록 지원할 수 있습니다.



현대적인 개발 프레임워크

개발 프레임워크는 개발자가 **Java™**와 같은 프로그래밍 언어를 사용하여 애플리케이션을 생성할 수 있도록 확장 가능한 플러그형 패턴의 코드 흐름을 제공합니다. 개발자는 이러한 반복 가능한 프레임워크를 프로그래밍 애플리케이션의 기반으로 사용하여 개발 시간을 줄이고 일관성을 높일 수 있습니다. 예를 들어, **Quarkus** 및 **Spring**과 같은 현대적인 쿠버네티스 네이티브 Java 개발 프레임워크를 활용하면 **컨테이너**, **마이크로서비스**, **서버리스**와 같은 새로운 인프라 기술을 통합할 수 있습니다.



서비스 메쉬

서비스 메쉬는 마이크로서비스 간의 커뮤니케이션을 제공하며 복구 능력, 보안, 관측성, 라우팅 제어, 인사이트와 같은 기능을 추가합니다. 서비스 메쉬는 애플리케이션의 다양한 부분들이 서로 데이터를 공유하는 방식을 제어합니다. 투명하고, 확장 가능한 전용 인프라 계층으로 애플리케이션 외부에 위치한 서비스 메쉬는 개발자가 기본적인 표준 서비스를 재구축하거나 연결하기 보다는 유용한 애플리케이션과 기능을 생성하는 데 집중하도록 합니다.



서버리스 모델

서버리스란 개발자가 기본 인프라를 관리할 필요 없이 애플리케이션을 빌드하고 실행할 수 있게 하는 클라우드 네이티브 개발 모델입니다. 클라우드 제공업체나 IT 운영 팀이 서버에 대한 프로비저닝, 유지 관리, 스케일링 등의 일상적인 작업을 처리하므로 개발자는 코딩에 집중할 수 있습니다. 서버리스 애플리케이션은 한번 배포되면 요구 사항이나 이벤트 트리거에 따라 자동으로 스케일 업 또는 "제로(0)" 수준까지 스케일 다운하여 사용하지 않는 중요한 인프라 리소스를 확보해줍니다.



CI/CD 파이프라인

지속적 통합/지속적 배포(CI/CD) 파이프라인은 DevOps 또는 사이트 신뢰성 엔지니어(SRE) 방식을 통해 소프트웨어 제공을 개선합니다. 이러한 파이프라인은 자동화를 적용하여 컴파일링 코드, 유닛 테스트, 코드 분석과 같은 애플리케이션 라이프사이클 관리 프로세스를 최적화합니다. 클라우드 네이티브와 컨테이너화된 환경으로 전환한 많은 조직이 **GitOps** 워크플로우를 통해 CI/CD 파이프라인을 확장하여 하이브리드 클라우드 인프라 전반에서 애플리케이션 제공을 자동화합니다.



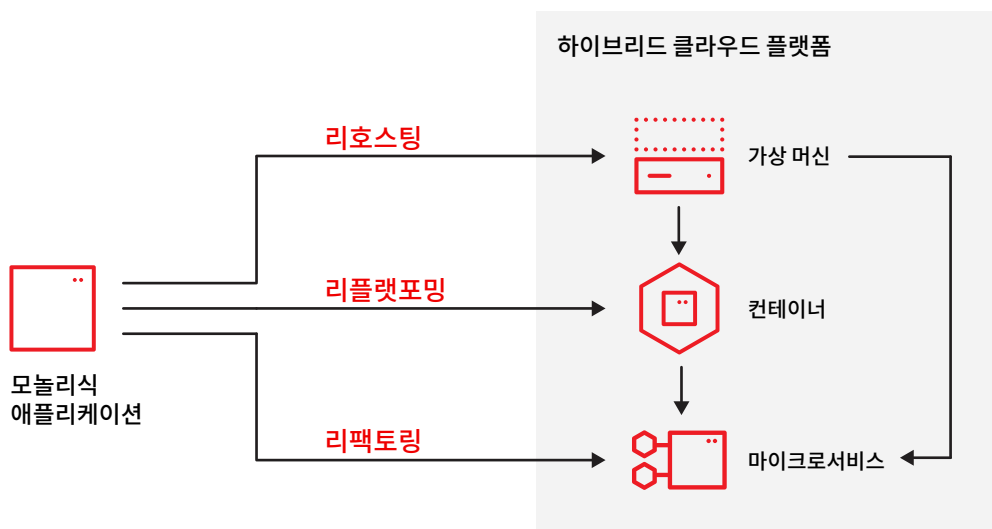
애플리케이션 현대화 여정을 위한 계획 수립

애플리케이션 요구 사항에 적합한 경로 선택

애플리케이션 현대화에는 3가지 주요 경로가 있습니다. 고유의 기술, 프로세스, 문화, 시간을 요하는 각 경로는 다양한 장점을 제공함으로써 디지털 환경에서 더 효과적으로 조직을 운영하도록 지원합니다.

모든 애플리케이션을 똑같이 현대화해야 하는 것은 아닙니다. 각 애플리케이션의 특성과 조직의 현재 및 향후 요구 사항에 최적화된 방법을 선택할 수 있습니다. 또한 현재 애플리케이션에 최소한의 변화를 주는 방법을 선택하여 요구 사항 변화에 따라 추가로 애플리케이션을 현대화할 수도 있습니다.

- 1** **경로 1: 리호스팅(rehosting)**
하이브리드 클라우드 플랫폼의 가상 머신에서 애플리케이션을 배포합니다.
- 2** **경로 2: 리플랫폼(replatforming)**
쿠버네티스 기반 클라우드 플랫폼에서 실행되는 컨테이너에 애플리케이션을 배포합니다.
- 3** **경로 3: 리팩토링(refactoring)**
애플리케이션을 클라우드 네이티브 마이크로서비스를 사용하여 다시 빌드하고, 새로운 기술을 통합하며, 하이브리드 클라우드 플랫폼에 배포합니다.



경로 1: 리호스팅(rehosting)

하이브리드 클라우드 플랫폼에서 실행되는 가상 머신에 기존 애플리케이션을 현재 상태로 배포합니다.

리호스팅 현대화 경로에는 기존 애플리케이션 서버에서 실행되는 애플리케이션을 하이브리드 클라우드 플랫폼에서 실행되는 가상 머신으로 리프트 앤 시프트하는 작업이 포함됩니다. 모놀리식 애플리케이션은 애플리케이션 서버에서 변함없이 유지되며 기존의 모든 통합 및 종속성을 그대로 유지합니다. 기존 플랫폼을 기반으로 외부 데이터와 통합을 계속 유지할 수 있습니다.

리호스팅은 일반적으로 짧은 시간 내에 완료되므로 마이그레이션 비용이 적게 들지만 다른 현대화 경로보다는 장점이 적습니다. 그렇더라도 리호스팅은 가상화된 애플리케이션, 컨테이너화된 애플리케이션, 클라우드 네이티브 애플리케이션을 일관된 단일 플랫폼으로 통합하여 향후 클라우드 네이티브 운영에 대비할 수 있도록 합니다.

리호스팅된 모놀리식 Java 애플리케이션의 예시는 아래와 같습니다.



리호스팅을 계획하는 기업은

20%

의 사용자 정의 레거시 애플리케이션을 현대화 이니셔티브의 일환으로 리호스팅하고자 합니다.³



팁:

기존 애플리케이션 서버 중에는 가상 머신으로 전환하기 전에 현대적인 실행(runtime) 환경에 애플리케이션을 재배포해야 하므로 가상 머신에서 작동하지 않는 경우가 생길 수 있습니다. 실행(runtime) 환경을 변경해야 하는 경우에는 애플리케이션의 플랫폼을 재구성하여 컨테이너에 배포(경로 2)함으로써 현대화 작업을 최적화하는 방안을 고려해 보세요.

3 Konveyor. "2022년 애플리케이션 현대화 현황 리포트(State of application modernization report 2022)," 2022년.

경로 2: 리플랫폼(Replatforming)

하이브리드 클라우드 플랫폼의 컨테이너에서 실행되는 현대적인 실행(runtime) 환경에 애플리케이션을 재배포합니다.

리플랫폼에는 애플리케이션을 분리(lift) 및 수정하고 하이브리드 클라우드 플랫폼의 컨테이너에서 실행되는 현대적인 실행(runtime) 환경으로 이동(shift)하는 과정이 수반됩니다. 일부 애플리케이션은 다른 애플리케이션에 비해 리플랫폼을 위해 더 많이 변경해야 하는 경우도 있습니다. 예를 들어, 기본적인 Java 애플리케이션은 거의 변경 없이 OpenJDK와 같은 컨테이너화된 Java 런타임의 장점을 누릴 수 있습니다. 하지만 엔터프라이즈 애플리케이션은 컨테이너에 배포하기 전에 현대적인 실행(runtime) 환경(예: Red Hat® JBoss® Enterprise Application Platform, IBM WebSphere Liberty 또는 Red Hat JBoss Web Server)으로 마이그레이션해야 합니다.

이 경로는 일반적으로 리호스팅보다 시간이 더 오래 걸리지만 장점이 더 많습니다. 단일 하이브리드 클라우드 플랫폼에서 애플리케이션을 통합하면 운영이 간소화되고 셀프 서비스 기능을 제공할 수 있습니다. 리플랫폼된 애플리케이션은 하이브리드 클라우드 플랫폼의 모든 기본 기능도 이용할 수 있습니다.

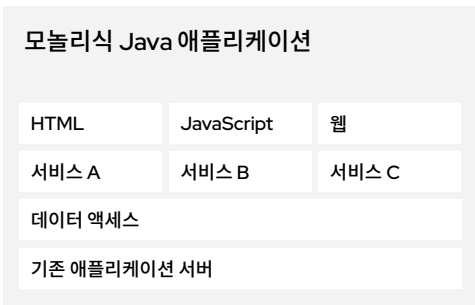
리플랫폼된 모놀리식 Java 애플리케이션의 예시는 아래와 같습니다.



리플랫폼을 계획하는 기업은

18%

의 사용자 정의 레거시 애플리케이션을 현대화 이니셔티브의 일환으로 리플랫폼하고자 합니다.⁴



4 Konveyor. "2022년 애플리케이션 현대화 현황 리포트(State of application modernization report 2022)," 2022년.

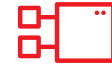
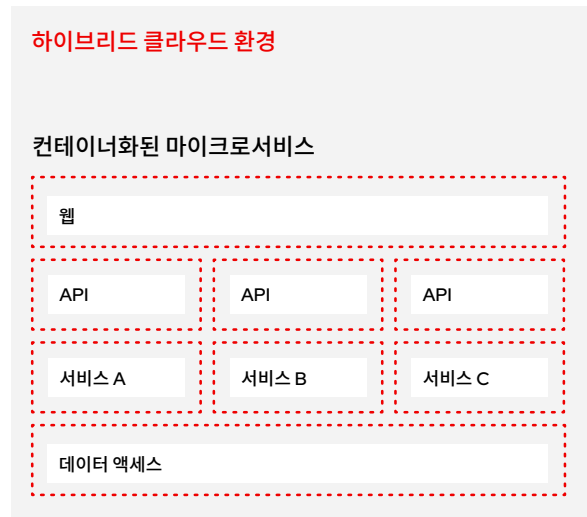
경로 3: 리팩토링(refactoring)

애플리케이션을 하이브리드 클라우드 플랫폼의 서비스 메쉬 내에 배포된 마이크로서비스로 재구축합니다.

리팩토링에는 애플리케이션 서비스를 하이브리드 클라우드 플랫폼의 서비스 메쉬 내에 배포된 마이크로서비스로 다시 개발하는 과정이 포함됩니다. 시간이 지나면 서비스를 재구축할 수 있으므로 기존 애플리케이션 아키텍처의 기능을 점차 새로운 애플리케이션 아키텍처로 이동할 수 있습니다. 이 재개발 프로세스 중에 기본 기술을 업그레이드하고 인공지능 및 머신 러닝(AI/ML), 분석, 자동 스케일링(Automatic scaling), 서버리스 기능, 이벤트 기반 아키텍처와 같은 새로운 클라우드 네이티브 기능에 포함할 수도 있습니다.

리팩토링은 시간이 가장 많이 걸리는 만큼 장점도 가장 많습니다. 리팩토링은 리호스팅 및 리플랫폼의 장점을 모두 제공하면서도 혁신적인 신기술을 이용해 비즈니스 민첩성과 가치를 높이는 데 도움이 됩니다.

리팩토링된 모놀리식 Java 애플리케이션의 예시는 아래와 같습니다.



리팩토링을 계획하는 기업은

17%

의 사용자 정의 레거시 애플리케이션을 현대화 이니셔티브의 일환으로 리팩토링하고자 합니다.⁵

⁵ Konveyor. "2022년 애플리케이션 현대화 현황 리포트(State of application modernization report 2022)," 2022년.

여정 계획하기

애플리케이션 포트폴리오 전체를 평가하여 각 애플리케이션에 가장 적합한 경로를 결정하세요.

1

확인

각 애플리케이션을 현재 플랫폼에서 새로운 하이브리드 클라우드 플랫폼으로 마이그레이션하는 과정과 관련한 기술 및 운영 리스크를 파악합니다. 두 플랫폼의 차이점을 비롯하여 이러한 특성이 애플리케이션에 어떤 영향을 미칠지 정확히 이해해야 합니다. 애플리케이션을 새로운 플랫폼에서 실행하는 데 필요한 변경 사항을 기록하세요.

2

제품 평가

가능한 각 현대화 경로와 관련된 비용, 장점, 우려 사항을 평가합니다.

비용:

- ▶ 개발 시간 및 노력
- ▶ 지속적인 운영 비용
- ▶ 플랫폼 라이선스 및 인프라 비용
- ▶ 비즈니스 잠재력 손실

장점:

- ▶ 운영 효율성
- ▶ 혁신 잠재력
- ▶ 개발 생산성 향상
- ▶ 새로운 비즈니스 및 매출 기회

고려해야 할 기타 요인:

- ▶ 애플리케이션 기대 수명
- ▶ 새로운 기능 요구 사항
- ▶ 기술 역량 및 코드 재사용
- ▶ 플랫폼 라이선스 갱신 기간

3

선택

애플리케이션과 관련한 조직의 목표, 예산, 제한된 일정에 가장 적합한 현대화 경로를 선택합니다.

4

검증

각 마이그레이션 과정 이후, 애플리케이션이 필요한 비즈니스 로직을 올바르게 제공하는지 검증합니다.

마이크로서비스 아키텍처는 애플리케이션에 적합할까요?

많은 조직이 일부 또는 전체 애플리케이션에 마이크로서비스 아키텍처를 배포할 계획을 가지고 있습니다. 각 애플리케이션을 평가하여 애플리케이션을 재설계하는 것이 합리적인지, 그리고 조직에 가치를 가져다 줄지 확인하세요. 예를 들어, 사용자 수가 적거나 확장 필요성이 제한적인 상황에서 애플리케이션을 재설계하면 비용이 장점보다 더 클 수도 있습니다. 최종적인 목표는 각 애플리케이션에 최상의 ROI(투자수익률)를 가져다 줄 아키텍처와 환경을 선택하는 것입니다.

오픈소스 전문가를 통한 현대화

오픈소스 소프트웨어와 클라우드 네이티브 기술, 쿠버네티스 전문 기업인 Red Hat은 애플리케이션을 현대화하고 애자일 클라우드 네이티브 IT 환경을 구축하여 변화하는 비즈니스 요구 사항을 지원합니다. 또한 애플리케이션 현대화 여정을 위한 완전하고 통합된 오픈 하이브리드 클라우드 기반을 제공합니다. Red Hat 솔루션은 다음과 같은 장점을 제공합니다.

- ▶ 현재 위치에서 시작하여 조직이 원하는 속도로 적시에 필요한 만큼 많이 또는 적게 애플리케이션을 현대화합니다.
- ▶ 기존 및 현대화된 애플리케이션을 지원하는 플랫폼과 제품으로 운영 효율성, 회복 탄력성, 적응성을 높입니다.
- ▶ 선호하는 런타임, 언어, 개발 툴을 계속 사용하면서, 동일 프로세스를 통해 모든 애플리케이션을 관리합니다.
- ▶ 표준화된 워크플로우, 지속적인 통합, 여러 환경에 대한 지원으로 일관되고 간소화된 개발자 경험을 제공합니다.

현대화 여정의 어느 단계에 있던 Red Hat은 조직이 현대화 작업의 장점을 극대화하도록 지원합니다.



애플리케이션 현대화 여정 간소화

Red Hat Consulting 전문가는 고객과 협력을 통해 애플리케이션 현대화 솔루션을 평가하고 구현하여, 비용을 절감하고 효율성을 개선하며 개발 속도를 높일 수 있도록 합니다. Red Hat 전문가가 고객이 팀 및 조직 차원에서 조직 전반에 애플리케이션을 빠르게 구축하고 발전시키는 데 필요한 사례와 툴을 개발하고 문화를 조성하도록 지원합니다.

클라우드 네이티브 성공에 필요한 기술 역량 강화

Red Hat은 조직이 기술 격차를 줄이고 비즈니스 과제를 해결할 수 있도록 포괄적인 교육 과정 커리큘럼을 제공합니다. 여러 티어에서 이용할 수 있는 Red Hat 교육 서브스크립션은 자기 주도식 및 강의식 온라인 교육 과정, 클라우드 기반 랩, 인증 테스트를 비롯하여 Red Hat 교육 과정에 무제한 액세스할 수 있는 권한을 제공합니다.

현대화를 위한 일관된 기반 배포

Red Hat 플랫폼은 애플리케이션 현대화에 필요한 유연성, 상호운용성, 일관성을 제공합니다.



Red Hat OpenShift®는 보안 중심의 하이브리드 클라우드 플랫폼으로, 다양한 환경에서 안심하고 애플리케이션을 개발, 배포, 관리할 수 있도록 합니다. 가상화, 클라우드 네이티브, 컨테이너화된 워크로드를 지원하여 원하는 속도로 워크로드를 현대화하고 이동시킬 수 있도록 하는 한편 모든 애플리케이션의 라이프사이클 관리를 통합하고 간소화합니다. 이 엔터프라이즈급 플랫폼에는 개발 프레임워크, CI/CD 파이프라인, 서버리스 개발 모델, 서비스 메시와 같이 기존의 모놀리식 애플리케이션을 현대화하기 위한 핵심 구성요소가 포함되어 있습니다.

Red Hat OpenShift는 개발자 생산성을 높이고 혁신을 촉진할 수 있도록 최적화되었습니다. 통합된 자동화는 효율적이고 재사용 가능한 파이프라인을 구축하고 애플리케이션 개발, 테스트, 배포 속도를 높입니다. 셀프 서비스 기능은 개발자가 IT 팀의 수동 개입 없이도 사전 승인된 리소스에 더욱 빠르게 액세스할 수 있게 해줍니다. 기본 제공 툴, 프레임워크, 런타임은 개발자가 선호하는 언어와 워크플로우를 사용하여 빠르게 작업을 시작하고 지속할 수 있도록 합니다.

Red Hat의 대규모 **인증 파트너 에코시스템**을 통해 타사 제품과 서비스를 활용하여 조직 내 환경을 사용자 정의할 수 있습니다. 다수의 벤더가 인증 Red Hat OpenShift **오퍼레이터**를 제공하므로 타사 제품의 설치와 관리를 자동화하고 간소화할 수 있습니다.

또한 추가 보안 및 관리 구성 요소로 Red Hat OpenShift 배포의 가치를 확장할 수 있습니다. **Red Hat Advanced Cluster Security for Kubernetes**는 Red Hat OpenShift 환경의 위협 벡터와 중요 취약점에 대한 인사이트를 제공합니다. DevOps 및 보안 툴과 통합되어 위협을 완화하고, 애플리케이션에 대한 운영 위험을 최소화하는 보안 정책을 실행합니다. **Red Hat Advanced Cluster Management for Kubernetes**는 Red Hat OpenShift 클러스터에 대한 향상된 가시성과 제어 기능을 제공합니다. 멀티클러스터 관리를 통합하고, 정책 기반의 거버넌스를 제공하며, 애플리케이션 라이프사이클 관리를 확장합니다.

Red Hat OpenShift를 통해 얻을 수 있는 비즈니스 가치

Red Hat OpenShift 환경을 사용하는 조직:

20%

높아진 DevOps 및 개발팀 생산성⁶

29%

빨라진 애플리케이션 개발 라이프사이클⁶

21%

향상된 IT 인프라 팀 효율성⁶

⁶ IDC 백서, Red Hat 후원. "Red Hat OpenShift의 비즈니스 가치." 문서 #US47539121, 2021년 3월



Quarkus는 경량화된 마이크로서비스와 서버리스 애플리케이션을 구축하기 위한 쿠버네티스 네이티브 Java 프레임워크입니다. **Red Hat build of Quarkus**는 Red Hat OpenShift와 통합되어 있으며, 낮은 메모리 사용, 빠른 스타트업 시간, 널리 사용되는 Java 표준 지원, 향상된 개발자 경험을 제공합니다.



오픈소스 **Knative** 프로젝트를 기반으로 하는 **Red Hat OpenShift Serverless**는 기본 인프라에서 애플리케이션 개발을 추상화합니다. 트리거되는 즉시 시작되어 요구 사항에 따라 신속히 스케일링되며 사용한 후에는 "제로(0)"로 스케일링되는 이벤트 기반 애플리케이션을 빌드, 배포, 실행하도록 합니다.



Red Hat OpenShift Pipelines는 Tekton 오픈소스 프로젝트를 기반으로 하는 클라우드 네이티브 CI/CD 솔루션입니다. Red Hat 개발자 툴과의 긴밀한 통합으로 반복 가능한 간소화된 단계를 통해 독립적으로 스케일링이 가능한 파이프라인을 빠르게 생성할 수 있습니다.



Red Hat OpenShift Service Mesh는 마이크로서비스 기반 애플리케이션을 일관된 방식으로 연결, 관리, 관측할 수 있게 해줍니다. 또한 마이크로서비스 간 트래픽을 통합, 관리, 추적, 모니터링, 분석하기 위한 일련의 오픈소스 프로젝트로 구성되어 있습니다.



Red Hat Application Foundations

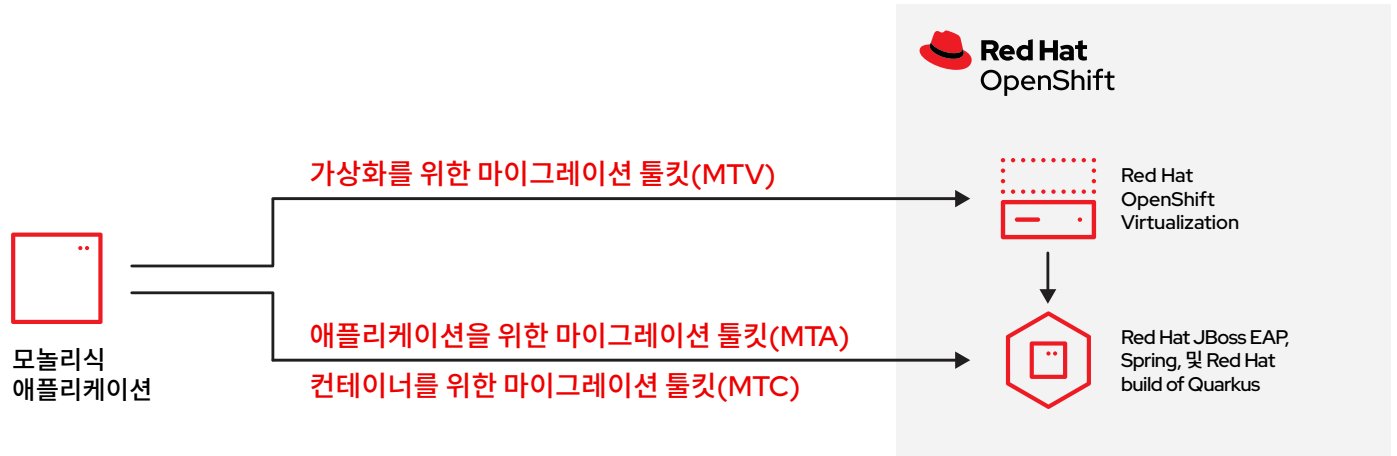
Red Hat Application Foundations는 소프트웨어를 개발하고 현대화하기 위한 일련의 포괄적인 구성 요소를 제공합니다. 이러한 기술은 하이브리드 클라우드 환경 전반의 보안을 고려하여 규모에 따라 애플리케이션을 빌드, 배포, 운영할 수 있도록 설계되었습니다.

애플리케이션을 현대화하기 위한 핵심 구성요소는 다음과 같습니다.

- ▶ Red Hat build of Quarkus
- ▶ Red Hat Single Sign-On
- ▶ Red Hat이 지원하는 OpenJDK
- ▶ Red Hat JBoss Web Server
- ▶ Red Hat JBoss Enterprise Application Platform
- ▶ Red Hat Integration
- ▶ 애플리케이션을 위한 마이크레이션 툴킷

안정적인 마이그레이션

Red Hat은 여러 **마이그레이션 툴**을 제공하여 현대화 여정을 계획하고 애플리케이션을 Red Hat OpenShift으로 이동시키도록 지원합니다.



가상화를 위한 마이그레이션 툴킷

규모에 맞게 가상 머신을 Red Hat OpenShift로 마이그레이션

가상화를 위한 마이그레이션 툴킷(MTV)은 VMware vSphere 또는 Red Hat Virtualization에서 **Red Hat OpenShift Virtualization**으로 가상 머신을 마이그레이션하도록 지원합니다. Red Hat OpenShift에 포함된 Red Hat OpenShift Virtualization을 사용하면 컨테이너 및 클라우드 네이티브 워크로드와 함께 가상 머신을 관리할 수 있습니다. MTV는 규모에 맞는 마이그레이션을 위한 간소화된 프로세스를 제공합니다. 또한 온사이트 분석 기능이 포함되어 있어 마이그레이션을 시작하기 전 잠재적인 문제점을 발견하고 해결할 수 있습니다.

애플리케이션을 Red Hat OpenShift로 리호스팅할 때 MTV를 사용하여 가상 머신을 마이그레이션할 수 있습니다.

애플리케이션을 위한 마이그레이션 툴킷

애플리케이션을 클라우드 지원 플랫폼으로 현대화 및 마이그레이션

애플리케이션을 위한 마이그레이션 툴킷(MTA)은 다년간의 경험을 바탕으로 완성된 일련의 툴로, 대규모 애플리케이션 현대화와 마이그레이션 프로젝트를 지원합니다. 코드 분석과 마이그레이션 속도를 높이고, 추정 작업량을 제공하며, 애플리케이션을 클라우드와 컨테이너 환경으로 이동시키도록 지원합니다.

리플랫폼링 또는 리팩토링을 수행하는 경우 MTA를 사용하여 애플리케이션을 평가하고 분석할 수 있습니다. MTA는 애플리케이션 플랫폼 업그레이드와 **Red Hat JBoss Enterprise Application Platform**, Spring, 또는 Red Hat build of **Quarkus**와 같은 클라우드 네이티브 실행(runtime) 환경으로의 마이그레이션을 지원합니다.

컨테이너를 위한 마이그레이션 툴킷

Red Hat OpenShift 클러스터 간 애플리케이션 마이그레이션

컨테이너를 위한 마이그레이션 툴킷(MTC)은 다양한 쿠버네티스 플랫폼의 스테이트풀 및 스테이트리스 애플리케이션을 Red Hat OpenShift 4 클러스터로 이동시킬 수 있도록 지원합니다. MTC는 다음의 쿠버네티스 리소스, 퍼시스턴트 볼륨 데이터, 내부 컨테이너 이미지를 마이그레이션합니다.

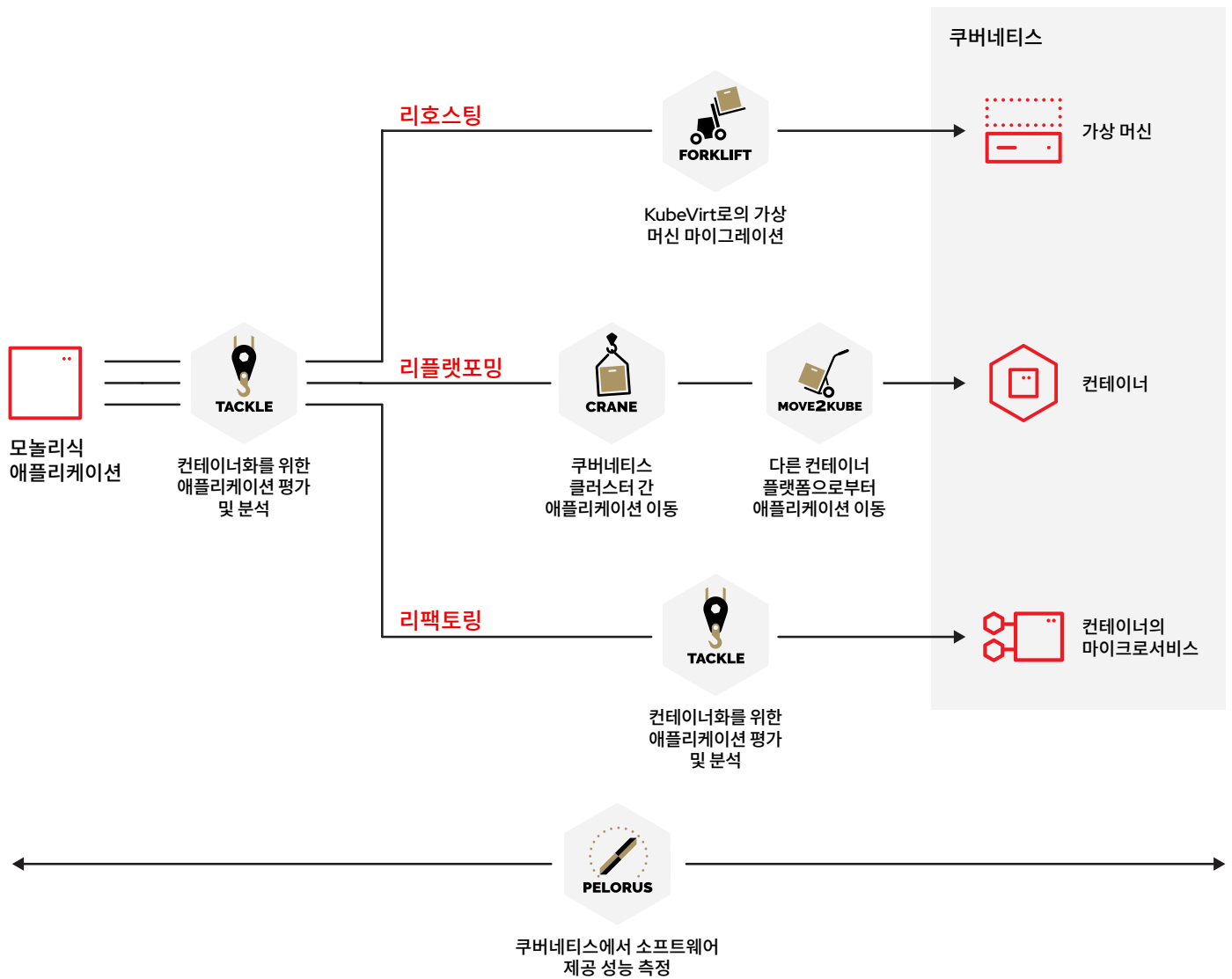
- ▶ Red Hat OpenShift 3 클러스터
- ▶ 비 Red Hat 쿠버네티스 플랫폼
- ▶ 기타 Red Hat OpenShift 4 클러스터

리플랫폼링 또는 리팩토링을 수행하는 경우 MTC를 사용하여 클러스터 간에 컨테이너화된 애플리케이션을 이동시킬 수 있습니다.

Konveyor 커뮤니티에 참여하세요

Red Hat과 IBM은 조직이 쿠버네티스 기반의 애플리케이션 플랫폼을 더욱 쉽게 이동시킬 수 있도록 Konveyor 커뮤니티를 구축했습니다. **Konveyor**는 애플리케이션을 쿠버네티스와 클라우드 네이티브 기술로 리호스팅, 리플랫폼, 리팩토링하기 위한 오픈소스 툴을 사용하여 애플리케이션을 현대화할 수 있도록 지원하는 커뮤니티입니다.

MTV, MTC, 및 MTA와 같은 Red Hat과 IBM의 마이그레이션 기술을 기반으로 하는 Konveyor는 쿠버네티스로 간편하고 손쉽게 마이그레이션할 수 있도록 다수의 커뮤니티 개발 프로젝트를 제공하고 있습니다. 또한 Konveyor 커뮤니티는 커뮤니티 회원들이 다양한 정보를 논의하고 검증할 수 있도록 정기적인 모임을 개최하고 있습니다.



실제 고객 사례



수처리 및 수자원 관리 기술 부문에서 업계를 선도하고 있는 DAB Pumps는 비즈니스 확장을 지원하고 프로세스와 서비스 제공 속도를 높이며 고객 경험을 개선하기 위해 오픈 하이브리드 클라우드 기술을 통해 IT 인프라를 현대화하고자 했습니다. 이같은 작업은 통합 아키텍처를 업데이트하여 기업의 여러 엔터프라이즈 애플리케이션 간의 커뮤니케이션을 개선하는 것이 핵심이었습니다.

DAB Pumps는 Red Hat OpenShift를 사용하여 모놀리식 환경에서 마이크로서비스 환경으로 전환하고 있습니다. 먼저 기존 미들웨어를 Red Hat OpenShift로 마이그레이션하고, 해당 플랫폼에서 클라우드 네이티브 애플리케이션을 개발하기 시작했습니다. 또한 Red Hat Integration 기술을 사용하여 제품 라이프사이클 관리(PLM), 고객 관계 관리(CRM), 비즈니스 인텔리전스(BI)와 같은 중요 시스템에 대한 데이터 액세스를 통합하고 가속화했습니다.

DAB Pumps는 다양한 Red Hat 기술을 사용하여 이기종 시스템을 통합하고, 애플리케이션 롤아웃 속도를 60% 높이고, 클라우드 환경으로의 이동을 준비할 수 있었습니다.



인프라 전반에
중앙화된 관리



신규 애플리케이션
출시 시간 60% 단축



기업 전반에 실시간
데이터 가용성 설정



"Red Hat과의 협업을 통해 명확한 표준을 바탕으로 더욱 안전하게 새로운 서비스와 업데이트된 제품의 시장 출시 기간을 이전보다 2배 이상 단축할 수 있었습니다."

Stefano Dalla Grana
DAB Pumps IT 책임자



Asiakastiето Group은 북유럽 지역에서 혁신적인 디지털 비즈니스와 소비자 정보 서비스를 제공하는 선도적인 업체입니다. 오픈 बैं킹으로의 전환을 지원하고, 새로운 데이터 프라이버시 및 보안 요구 사항을 준수하고, 북유럽의 높은 부채율을 해결하기 위해 이 회사는 데이터 기반의 신용 평가 솔루션을 구축하기로 결정했습니다. Asiakastiето는 개인의 상환 능력을 더 정확히 평가하여 개인 부채와 채무 불이행을 줄이기 위해 Red Hat OpenShift, Red Hat Integration, Quarkus를 사용해 자체 계정 인사이트 애플리케이션을 개발했습니다.



덴마크의 여러 대규모 은행에 IT 서비스를 제공하는 **Bankdata**는 고품질 IT 솔루션을 빌드, 구현, 실행하고 있습니다. **Miracle**의 도움으로 Bankdata는 Red Hat OpenShift를 도입하고 CI/CD 플랫폼을 구축했습니다. 리소스 효율성과 롤백 속도를 더욱 극대화하기 위해 Miracle과 Bankdata 팀은 현재 사용 중인 Spring Boot Java 프레임워크와 Quarkus의 성능과 효율성을 비교하는 실험을 실시했습니다. 그 결과, 테스트 애플리케이션의 Quarkus 네이티브 버전은 더 빠른 부팅 시간, 처리되는 호출당 적은 메모리 사용량, 더 적은 CPU 사용량을 제공하는 것으로 나타났습니다.



Lufthansa Technik은 항공사가 유지 보수 체계화 및 일정 수립 개선에 데이터를 사용함으로써 비행 지연과 취소를 미연에 방지하도록 지원하는 AVIATAR라는 디지털 플랫폼을 운영하고 있습니다. 빠른 성장과 고객의 요구 사항 증가에 대처하기 위해 이 회사는 Red Hat OpenShift on Microsoft Azure에 기반을 둔 마이크로서비스 아키텍처로 이전하기로 결정했습니다. AVIATAR 팀은 클라우드 리소스 사용량을 줄이기 위해 Quarkus를 배포했습니다. 현재 AVIATAR 팀은 Quarkus를 사용하여 새로운 마이크로서비스를 개발하고 있으며, 성능을 향상하고 리소스 사용을 최적화하기 위해 기존 Spring Boot 마이크로서비스를 마이그레이션하는 과정에 있습니다.



"오픈소스는 혁신을 위한 더욱 빠른 속도와 범위를 제공해주고 있으며, Red Hat은 더욱 강화된 보안 기능과 고품질의 지원을 제공하고 있습니다."⁸

Reetta Sinelampi
Asiakastiето 개발 책임자

부팅 시간

1초 미만

Quarkus 네이티브 애플리케이션 버전

57%

더 적은 로드 시 호출당 메모리 사용량(Quarkus 네이티브 애플리케이션 버전)



"Quarkus를 사용하여 서비스 가용성과 응답 시간을 희생하지 않고도 3배 더 높은 고밀도 배포를 실행할 수 있었습니다."

Thorsten Pohl
Lufthansa Technik 디지털 제품부 AVIATAR
제품 소유자 오토메이션 & 플랫폼 아키텍트

⁸ Red Hat 보도 자료. "Asiakastiето, Red Hat으로 오픈 बैं킹 혁신 실현(Asiakastiето unlocks open banking innovation with Red Hat)." 2020년 5월 27일.

시작하기

애플리케이션 포트폴리오를 현대화하면 디지털 환경에서 더욱 빠르게 혁신하고 효과적으로 경쟁할 수 있습니다. Red Hat은 성공적인 여정을 위한 제품, 전문성, 서비스를 제공하고 있습니다. 원하는 속도로 현대화하고, 모든 애플리케이션을 위한 완전하고 통합된 기반으로 효율성, 회복 탄력성, 유연성을 높이세요.

▶ Red Hat OpenShift 무료 체험

▶ Red Hat OpenShift 자세히 알아보기

현대화 준비 상태를 평가해 보세요

조직이 애플리케이션 현대화를 위해 얼마나 잘 준비되어 있는지, 여정을 가속화하기 위해 할 수 있는 것은 무엇인지, 무료로 제공되는 온라인 자가 평가를 통해 확인해 보세요.



Red Hat Consulting과 함께 현대화 여정에서 앞서 나아가세요.

Red Hat 전문가가 현대화에 필요한 사례, 툴, 문화를 개발할 수 있도록 도와드립니다.

▶ Red Hat Consulting에 대해 자세히 알아보기