애자일 통합 아키텍처

컨테이너 기반 및 마이크로서비스 지향 경량 통합 런타임

목차

멀티 클라우드, 분산화 및 마이크로서비스의 요구사항을 신속하게 충족시킬 수 있도록 민첩한 접근 방식을 중심으로 하는 애자일 통합을 통해 디지털 전환을 구현하십시오.

전체 개요

디지털 전환을 추구하는 조직은 통합 기술을 사용하고 구현할 수 있는 새로운 방법을 채택하여 멀티 클라우드, 분산화 및 마이크로서비스라는 목표에 적합한 방식으로 신속하게 이동할 수 있어야 합니다. 애플리케이션 통합 계층은 조직의 생산성 극대화와 큰 관련이 없는 아키텍처 및 개발 모델을 강제하기보다는 조직이 새로운 고객 경험을 구축하는 데 아무런 불편함이 없도록 전환되어야 합니다.

많은 조직이 마이크로서비스 아키텍처와 같은 민첩한 애플리케이션 기술을 채택하고 있으며 이제 이러한 변화의 이점을 인식하기 시작했습니다. 이러한 접근 방식은 기업의 API 전략을 보완하고 가속화합니다. 기업은 또한 이러한 접근 방식을 사용하여 기존의 ESB 인프라를 현대화하고 프라이빗 또는 퍼블릭 클라우드에서 통합 서비스를 보다 효과적으로 관리하고 운영할 수 있는 방법을 모색해야 합니다.

이 백서는 디지털 전환에 필요한 민첩성, 확장성 및 복원성 요구사항을 충족시켜줄 통합 솔루션을 위한 컨테이너 기반의 분산된 마이크로서비스 지향 접근 방식인 **애자일 통합 아키텍처**의 장점을 탐구하는 책을 근거로 합니다.

통합의 변화

IDC는 디지털 전환 이니셔티브에 대한 지출이 향후 5년간 20조 달러의 시장 기회를 창출할 것으로 예측합니다[1]. 이렇게 지출이 급증할 것으로 예상하는 배경은 무엇일까요? 모든 유형의 데이터를 활용하는 애플리케이션 네트워크에서 연결된 경험을 통해 새로운 고객 경험을 구축해야 할 필요성이 끊임없이 커지고 있기 때문입니다.

이는 쉬운 작업이 아닙니다. 프로세스와 정보 소스를 적시에 적절하게 결합하는 것은 매우 어려운 일이며, 특히 SaaS 비즈니스 애플리케이션의 적극적인 도입을 고려할 때는 더욱 그렇습니다. 비즈니스 프로세스에 새로운 데이터 소스를 활용하여 경쟁력 있는 차별화를 창출해야 합니다.

"새로운 고객 경험을 창출하기 위해 조직은 끊임없이 증가하는 일련의 애플리케이션, 프로세스 및 정보 소스를 활용해야 합니다. 이로 인해 통합 역량에 대한 기업의 요구사항과 투자가 크게 확대됩니다."

디지털 전환을 위한 애플리케이션 통합의 가치

새로운 고객 경험 구축을 위한 어젠다를 고려하고 이러한 이니셔티브를 지원하는 서비스 및 API에 대한 데이터 액세스 및 제공 방법에 집중하면 애플리케이션 통합이 제공하는 몇 가지 중요한 이점을 확인할 수 있습니다.

- 불일치 문제의 효과적인 해결: 기업의 멀티 클라우드 환경이 아무리 복잡하게 진화하더라도 시스템과 형식에 관계없이 데이터에 액세스하고 동질성을 구축할 수 있습니다.
- 엔드포인트의 전문지식: 현대적인 통합에는 복잡한 프로토콜 및 데이터 형식과 관련된 스마트 기능이 포함되지만 엔드 시스템 내의 실제 개체, 비즈니스 및 기능에 대한 인텔리전스도 통합됩니다.
- 데이터를 통한 혁신: 애플리케이션 혁신의 상당 부분은 경계를 넘어 데이터를 결합하고 이로부터 의미를 창출하기 위한 것이며 이는 특히 마이크로서비스 아키텍처에서 볼 수 있는 특성에 부합합니다.
- 엔터프라이즈급 아티팩트: 통합 플로우는 오류 복구, 장애 복원, 로그 캡처, 성능 분석 등에 대한 엔터프라이즈급 기능을 포함하는 런타임에서 엄청난 가치가 발생합니다.

통합 환경은 엔터프라이즈 및 시장 컴퓨팅 수요에 맞춰 변화하고 있지만, 어떻게 SOA 및 ESB에서 현대적인 컨테이너 기반의 애자일 통합 아키텍처로 전환되었을까요?

지금까지의 여정: SOA, ESB 및 API

애자일 통합의 미래를 기대하려면 먼저 기존에 어떤 일이 있었는지를 이해해야 합니다. 서비스 지향 아키텍처(SOA) 패턴은 밀레니엄의 시작과 함께 등장했으며, 초반에는 구축된 표준 SOA가 광범위하게 채택되면서 모든 시스템이 동기 노출 패턴을 통해 다른 시스템을 발견하고 상호작용할 수 있는 밝은 미래를 예고했습니다.

시간이 조금 지난 뒤에는 이전의 허브 앤 스포크(hub-and-spoke) 패턴을 기반으로 한 백엔드 시스템에 연결성을 제공하는 기술인 엔터프라이즈 서비스 버스(ESB)로의 이동이 있었습니다. 많은 기업이 ESB 패턴을 성공적으로 구현했지만, 이 개념은 클라우드 네이티브 공간에서 많은 사랑을 받고 있지는 못합니다. 무겁고 민첩성이 부족한 것으로 나타났기 때문입니다. 우리는 어떻게 한쪽 끝에서 다른 쪽 끝으로 이동하게 되었을까요?

그 이유는 다음과 같은 몇 가지 상호관련 요인 때문입니다.

- SOA는, 특히 엔터프라이즈급 프로그램에 비용을 부담하는 입장에서는 ESB의 구현보다 복잡했습니다.
- ESB 패턴은 프로덕션 서버 클러스터에 수십 또는 수백 개의 통합을 설치하여 전체 기업을 위한 단일 인프라를 형성했습니다. ESB 패턴이 무거운 중앙집중화를 요구하지는 않지만, 결과물은 거의 항상 이에 의한 문제에 시달렸습니다.

[1]IDC MaturityScape Benchmark: Digital Transformation Worldwide, 2017, Shawn Fitzgerald. Golluscio.

- 중앙집중식 ESB 패턴의 경우 프로젝트가 바뀌면 인터페이스를 재사용할 수 없었기 때문에 기업이 기대했던 상당한 비용절감 효과를 제공하지 못하는 경우가 많았습니다.
- ESB와 같은 전사적 이니셔티브는 자금 조달에 어려움을 겪었으며, 종종 자금 조달 비용을 충당하기에 충분할 정도로 재사용할 수 있는 서비스에만 적용되었습니다.

ESB 패턴은 비즈니스 이니셔티브의 맥락에서 구체적으로 적용되지 않았기 때문에 전사적 이니셔티브에 대한 지속적인 자금 지원을 보장하는 데 어려움이 있었습니다.

그 결과, 이 SOA 전문가 팀이 창출한 서비스는 애초의 의도였던 조력자가 아닌 프로젝트의 병목 지점이 되었습니다. 이로 인해 중앙집중식 ESB 패턴은 일반적으로 나쁜 평판을 얻게 되었습니다.

ESB 패턴에 적용되는 서비스 지향 아키텍처는 다른 시스템의 데이터를 보다 빠르게 통합하여 새로운 애플리케이션을 생성할 수 있는, 즉 재사용 가능하고 동시에 사용할 수 있는 서비스 및 API를 만들기 위한 전사적 이니셔티브입니다.

반면에 마이크로서비스 아키텍처는 개별 애플리케이션을 보다 민첩하고, 확장 가능하며, 탄력적으로 만들 수 있는 방법에 대한 옵션입니다.

애자일 통합 아키텍처의 사례

마이크로서비스 개념이 애플리케이션 영역에서 크게 인기를 끄는 이유는 무엇일까요? 마이크로서비스는 애플리케이션 구조화에 대한 대안적인 접근 방식이기 때문입니다. 애플리케이션은 동일한 서버에서 실행되는 대규모 코드의 사일로가 아니라 완전히 독립적으로 실행되는 작은 구성요소의 집합으로 설계되었습니다.

마이크로서비스 아키텍처는 다음과 같은 세 가지 중요한 이점을 제공합니다.

1. 뛰어난 **민첩성**: 마이크로서비스는 완전히 분리하여 이해할 수 있으며 독립적으로 변경할 수 있을 만큼 작습니다.

2. 탄력적인 **확장성**: 리소스 사용을 비즈니스 모델에 완벽하게 연결할 수 있습니다.

3. 개별적 **복원성**: 적절한 디커플링을 통해 하나의 마이크로서비스를 변경하더라도 런타임 시 다른 서비스에 영향을 미치지 않습니다.

이러한 이점을 고려하여 일반적으로 중앙집중화된 사일로 방식으로 구현되는 통합을 마이크로서비스 아키텍처를 기반으로 하는 새로운 관점으로 재구성한다면 어떻게 될까요? 이것이 이른바 **"애자일 통합 아키텍처"**입니다.

애자일 통합 아키텍처는 "통합 솔루션을 위한 컨테이너 기반의 분산된 마이크로서비스 지향 아키텍처"로 정의됩니다.

애자일 통합 아키텍처에는 서로 관련되어 있긴 하지만 별개의 세 가지 측면이 있습니다.

측면 1: 세분화된 통합 배포.
사일로 방식의 ESB 통합을 별개의 런타임으로 분리함으로써 얻을 수 있는 이점은 무엇입니까?

측면 2: 분산된 통합 소유권.
보다 세분화된 접근 방식을 잘 활용하기 위해 조직 구조를 어떻게 조정해야 할까요?

측면 3: 클라우드 네이티브 통합 인프라.
통합에 대한 완전한 클라우드 네이티브 접근 방식을 통해 얻을 수 있는 이점은 무엇입니까?

측면 1: 세분화된 통합 배포.

고가용성(HA) 단일 통합 서버 쌍에 모든 통합이 배포되는 통합 허브 또는 ESB 패턴의 중앙집중식 배포는 프로젝트에 병목현상을 초래하는 것으로 나타났습니다. 공유 서버에 배포하는 경우에는 기존의 중요한 인터페이스가 불안정해질 위험이 있습니다. 개별 프로젝트는 새 기능에 액세스하기 위해 통합 미들웨어의 버전 업그레이드를 선택할 수 없습니다.

전사적인 ESB 구성요소를 더 작고 관리하기 쉬우며 특정 작업용으로 분해할 수 있습니다. 경우에 따라 표시되는 각 인터페이스에 대해 하나의 런타임까지 상세히 확인할 수도 있습니다. 이러한 "세분화된 통합 배포" 패턴은 특화된 적절한 크기의 컨테이너를 제공하여 민첩성, 확장성 및 복원성을 향상시키며 과거의 중앙집중식 ESB 패턴과는 매우 다른 모습을 보여줍니다. 그림 1은 중앙집중식 ESB와 세분화된 통합 배포가 어떻게 다른지 간략하게 보여줍니다.

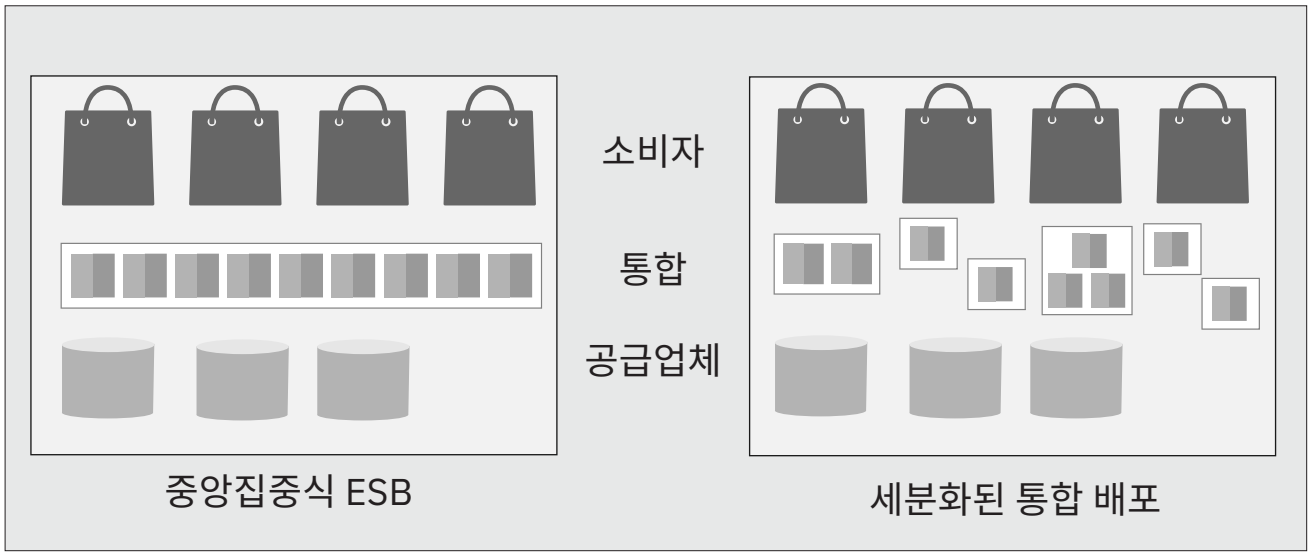

그림 1: 중앙집중식 ESB 및 세분화된 통합 배포의 간략 비교

세분화된 통합 배포는 마이크로서비스 아키텍처의 이점을 활용합니다. 세분화된 통합 배포의 관점에서 마이크로서비스가 제공하는 이점에 대해 다시 살펴보겠습니다.

- **민첩성:** 병목현상이 발생할 수 있는 중앙집중식 그룹 또는 인프라에 연연하지 않고 여러 팀이 독립적으로 통합 작업을 수행할 수 있습니다. 개별 통합 플로우는 다른 플로우와 독립적으로 변경, 재구성 및 배포될 수 있으므로 변경사항을 보다 안전하게 적용하고 생산 속도를 극대화할 수 있습니다.
- **확장성:** 개별 플로우를 자체적으로 확장할 수 있으므로 클라우드 인프라의 효율적이고 탄력적인 확장을 활용할 수 있습니다.
- **복원성:** 별도의 컨테이너에 배포된 분리된 통합 플로우는 메모리, 연결 또는 CPU와 같은 공유 리소스를 도용하여 서로 영향을 미칠 수 없습니다.

민첩성, 확장성 및 복원성을 고려할 때 분산된 통합 없이는 이러한 세분화된 통합의 이점을 얻을 수 없다는 점을 유념해야 합니다.

지금 다운로드할 수 있는 애자일 인프라 아키텍처에서 세분화된 통합에 대해 자세히 알아보세요.

측면 2: 분산된 통합 소유권.

서비스 지향 아키텍처가 직면한 중요한 과제는 중앙 통합 팀과 인프라가 서비스 계층을 생성하도록 강제하는 경향이 있다는 것입니다.

이는 프로젝트가 항상 중앙 통합 팀에 종속적이기 때문에 프로젝트가 진행되는 속도에 지속적인 마찰을 일으켰습니다. 중앙 팀은 통합 기술은 잘 알고 있지만 통합하는 애플리케이션을 이해하지 못하는 경우가 많았기 때문에 요구사항 변환이 느리고 오류가 발생하기 쉬웠습니다.

많은 조직이 애플리케이션 팀에서 자체적으로 서비스를 생성해 소유하는 것을 선호했지만 당시의 기술과 인프라로는 이를 지원할 수 없었습니다.

세분화된 통합 배포로 이동하면서 통합의 생성 및 유지보수에 대한 소유권을 분산시킬 수 있게 되었습니다. 새로운 기능의 구현을 간소화함으로써 비즈니스 애플리케이션 팀에서 통합 작업을 수행하는 것은 그다지 비합리적인 일이 아닙니다.

세분화된 통합 배포에 대해 더 자세히 알고 싶으신가요? 지금 다운로드할 수 있는 애자일 인프라 아키텍처 문서에서 자세한 내용을 알아보세요.

측면 3: 클라우드 네이티브 통합 인프라

최근 몇 년 동안 통합 런타임에는 큰 변화가 있었습니다. 이러한 경량 런타임을 진정한 클라우드 네이티브 방식으로 사용할 수 있게 된 것입니다. 이를 통해 기존 전용 메커니즘의 클러스터 관리, 확장, 가용성 및 실행 중인 클라우드 플랫폼에 대해 많은 부담을 덜 수 있게 되었습니다.

이는 단지 컨테이너 기반 환경에서 이를 실행하는 것보다 훨씬 더 많은 것을 의미합니다. 이는 쿠버네티스(Kubernetes) 및 기타 수많은 클라우드 표준 프레임워크와 같은 오케스트레이션 기능을 최대한 활용하여 기본적이고 광범위한 기능이 아니라 구체적이고 세부적으로 기능할 수 있어야 한다는 것을 의미합니다(즉, 숲이 아닌 나무에 접근).

"숲보다 나무 접근 방식"을 채택하면 DevOps 팀이 환경 및 솔루션과 상호작용하는 전반적인 방식에 영향을 미치며, 더 많은 솔루션이 경량 아키텍처로 이동함에 따라 효율성을 높일 수 있습니다.

애자일 통합 아키텍처를 수용하기 위해 현대의 통합 런타임은 어떻게 변화했을까요?

분명히 애자일 통합 아키텍처에는 매우 다른 방식으로 구현되는 통합 토폴로지가 필요합니다. 그러한 측면의 중심에는 컨테이너 기반 환경에서 실행될 수 있고 클라우드 네이티브 배포 기법에 적합한 최신 통합 런타임이 있습니다. 현대의 통합 런타임은 과거의 그것과는 매우 다릅니다. 이러한 차이점 중 몇 가지를 살펴보겠습니다.

- **빠른 경량 런타임:** 도커(Docker)와 같은 컨테이너에서 실행되며 몇 초만에 시작하거나 중지할 수 있을 정도로 충분히 가볍고 쿠버네티스(Kubernetes)와 같은 오케스트레이션 프레임워크를 통해 쉽게 관리할 수 있습니다.
- **무종속성:** 데이터베이스나 메시지 큐가 더 이상 필요하지 않지만 필요 시에는 쉽게 연결할 수 있습니다.
- **파일 시스템 기반 설치:** 파일 시스템에 바이너리를 배치하고 이를 시작하기만 하면 간단하게 설치할 수 있기 때문에 도커(Docker) 이미지의 계층화된 파일 시스템에 적합합니다.
- **DevOps 도구 지원:** 런타임은 지속적인 통합 및 배포가 가능해야 합니다. 스크립트 및 속성 파일 기반 설치, 빌드, 배포, 구성을 통해 "코드로서의 인프라" 구현을 지원해야 합니다. DevOps 파이프라인에 신속하게 포함될 수 있도록 표준 빌드 및 배포 도구용 템플릿 스크립트를 제공해야 합니다.
- **API 우선: 기본 통신 프로토콜은 RESTful API여야 합니다.** RESTful API로 표시하는 통합은 기본적인 것이어야 하며 Open API 사양과 같은 일반적인 규칙을 기반으로 해야 합니다. 다운스트림 RESTful API 호출 역시 정의 파일을 통한 검색을 비롯한 간단한 것이어야 합니다.
- **디지털 연결성:** 통합 런타임이 항상 제공해온 풍부한 엔터프라이즈 연결성 뿐만 아니라 최신 리소스에 대한 연결성도 보장되어야 합니다. NoSQL 데이터베이스(MongoDB, Cloudant 등) 및 Kafka와 같은 메시징 서비스가 그 예입니다. 또한 Salesforce와 같은 SaaS 애플리케이션을 위해 다양한 애플리케이션 인텔리전트 커넥터에 액세스할 수 있어야 합니다.

- **지속적 전달:** 지속적 전달은 표준 DevOps 파이프라인 도구에 맞는 명령줄 인터페이스 및 템플릿 스크립트를 통해 가능합니다. 이에 따라 인터페이스 구현에 필요한 지식을 줄이고 전달 속도를 높일 수 있습니다.
- **향상된 도구 지원:** 통합을 위한 향상된 도구 지원이란 통합에 대한 지식이 없는 개인도 대부분의 인터페이스를 구성만으로 구축할 수 있음을 의미합니다. 일반적인 통합 패턴에 대한 템플릿을 추가하면 통합 모범 사례가 도구에 반영되어 작업을 더욱 간소화할 수 있습니다. 심층 통합 전문가가 자주 필요하지 않으며, 분산된 통합을 설명하는 다음 섹션에서 볼 수 있듯이 일부 통합은 애플리케이션 팀이 수행할 수 있습니다.

최신 통합 런타임은 애자일 통합 아키텍처의 세 가지 측면, 즉 세분화된 배포, 분산된 소유권 및 진정한 클라우드 네이티브 인프라에 매우 적합합니다.

클라우드 네이티브 인프라에 대해 더 자세히 알고 싶으신가요? 지금 애자일 통합 아키텍처 문서를 다운로드하세요.

통합 플랫폼을 위한 애자일 통합 아키텍처

이 백서에서는 애자일 통합 아키텍처에 구현되는 애플리케이션 통합 기능에 대해 중점적으로 살펴보고 있습니다. 그러나 많은 엔터프라이즈 솔루션은 몇 가지 중요한 통합 기능을 적용해야만 해결될 수 있습니다. 통합 플랫폼(또는 일부 애널리스트는 "하이브리드 통합 플랫폼"으로 언급)은 이러한 기능을 결합하여 조직이 보다 효율적이고 일관된 방식으로 비즈니스 솔루션을 구축할 수 있도록 합니다.

많은 업계 전문가들이 이 통합 플랫폼의 가치에 동의합니다. Gartner는 다음과 같이 설명합니다.

하이브리드 통합 플랫폼(HIP)은 온프레미스 및 클라우드 기반의 통합 프레임워크이며 각기 다른 기술을 가진 인력(통합 전문가 및 비전문가)이 다양한 통합 사용 사례를 지원할 수 있도록 하는 거버넌스 기능입니다. 통합을 담당하는 애플리케이션 책임자는 HIP 기능 프레임워크를 활용하여 통합 전략 및 인프라를 현대화하고 디지털 비즈니스에서 새로운 사용 사례를 해결할 수 있어야 합니다[2].

Gartner가 언급한 핵심 사항 중 하나는 통합 플랫폼을 통해 조직의 다양한 사용자가 자신의 요구에 가장 적합한 사용자 경험에서 작업을 할 수 있다는 것입니다. 즉, 비즈니스 사용자는 직관적인 문제 해결을 도와주는 보다 단순한 경험을 통해 생산성을 높일 수 있으며 IT 전문가는 보다 복잡한 엔터프라이즈 시나리오를 처리할 수 있도록 전문적인 수준의 제어 권한을 갖습니다. 그러면 이러한 사용자는 전체적으로 거버넌스를 유지하면서 공유된 자산을 재사용하여 함께 작업할 수 있습니다.

디지털 전환의 새로운 사용 사례를 충족시키는 것은 다양한 사용자 커뮤니티를 지원하는 것만큼 중요합니다. 이 백서에서는 이러한 새롭게 등장하는 사용 사례에 대해서도 살펴보겠지만, 먼저 통합 플랫폼이 갖춰야 하는 주요 기능에 대해 자세히 알아볼 필요가 있습니다.

IBM Cloud 통합 플랫폼

IBM Cloud 통합은 일련의 핵심 통합 기능을 빠르고 간단하면서도 신뢰할 수 있는 일관성을 제공하는 플랫폼으로 통합합니다. 강력한 통합 및 API를 몇 분 만에 쉽게 구축할 수 있고, 뛰어난 성능과 확장성을 제공하며, 엔터프라이즈급 보안과 함께 탁월한 엔드 투 엔드 기능을 제공합니다.

IBM Cloud 통합 플랫폼에는 6가지의 핵심 통합 전문기술이 각각 동급 최고의 기능과 연결되어 있습니다. 핵심 통합 기술은 다음과 같습니다.

API 관리:
조직 내외부의 특정 개발자 커뮤니티를 위해 재사용 가능한 API로 비즈니스 서비스를 표시하고 관리합니다. 조직은 고유한 데이터 및 서비스 자산을 효과적으로 공유하여 새로운 애플리케이션과 새로운 비즈니스 기회를 창출할 수 있는 API 전략을 도입합니다.

보안 게이트웨이:
API, API가 전송하는 데이터 및 이를 실행하는 시스템을 보호하는 DMZ 지원 에지 기능을 통해 연결성과 통합을 기업 외부로까지 확장합니다.

애플리케이션 통합:
온프레미스 또는 클라우드에 있는 애플리케이션 및 데이터 소스를 연결하여 언제 어디서나 데이터를 사용할 수 있도록 비즈니스 정보를 조정 및 교환합니다.

[2]Hype Cycle for Application Infrastructure and Integration, 2017, Elizabeth Golluscio.

메시징:
시스템 또는 네트워크 문제 발생 시 메시지 손실, 중복 또는 복잡한 복구 없이 신뢰할 수 있는 메시지 전달을 제공하여 언제 어디서나 실시간 정보를 사용할 수 있도록 보장합니다.

데이터 통합:
분석을 위해 데이터 웨어하우스 또는 데이터 레이크 내에서 비즈니스에 관한 일관된 뷰를 생성할 수 있도록 데이터를 액세스하고 정리하고 준비합니다.

고속 전송:
온프레미스와 클라우드 간에 또는 클라우드와 클라우드 간에 대량의 데이터를 향상된 보안 수준으로 빠르고 예측 가능하게 전송합니다. 데이터가 매우 큰 경우 조직이 클라우드 플랫폼을 빠르게 채택할 수 있도록 지원합니다.

이 티저 백서를 통해 통합 플랫폼의 일부로서 필요한 다양한 핵심 기능, 이러한 기능이 함께 작동하기 위한 요구사항, 그리고 플랫폼의 민첩성, 확장성 및 복원성을 향상시키기 위해 애자일 통합 아키텍처를 채택하는 방법에 대한 폭넓은 관점을 얻으셨기를 바랍니다.

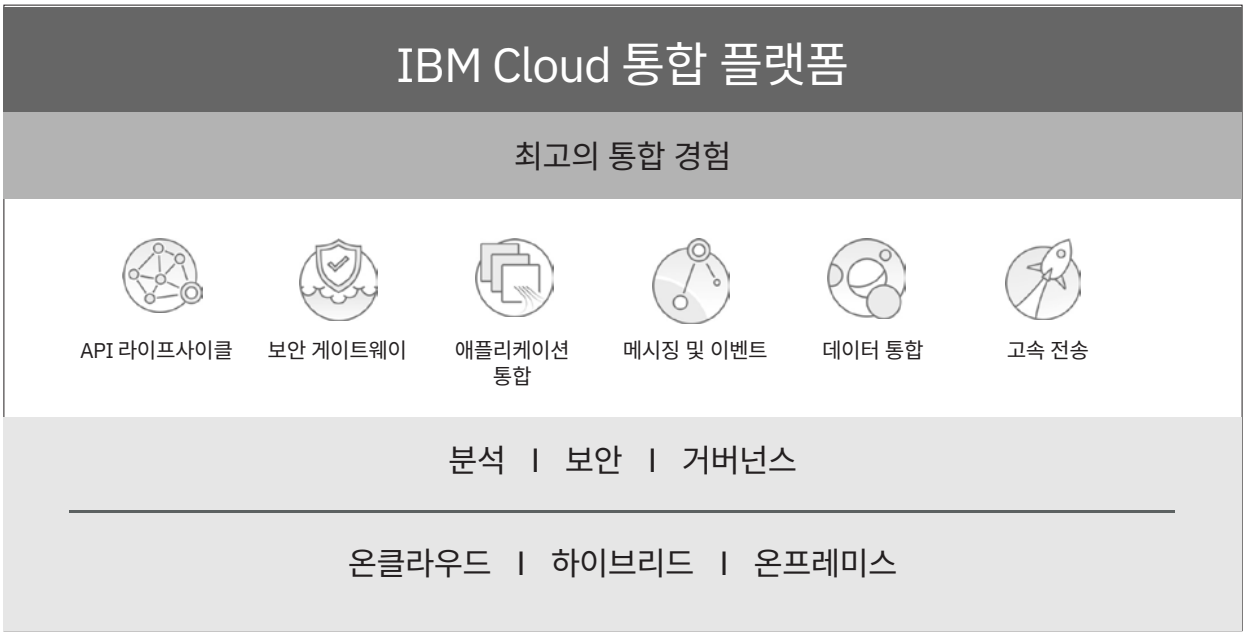

그림 2: IBM Cloud Integration 플랫폼

모든 내용이 종합되어 있는 전자책을 다운로드하여 애자일 통합 아키텍처에 대해 자세히 알아보세요.

IBM Corporation
Global Technology Services
Route 100
Somers, NY 10589

미국에서 제작
2018년 8월

본 문서의 내용(관련 세금을 제외한 통화 또는 가격 참조 포함)은 최초 발행일 현재 상태이며 IBM에 의해 언제든지 변경될 수 있습니다. IBM이 사업을 운영하는 국가라도 일부 제품은 공급되지 않을 수 있습니다.

여기서 소개된 성능 데이터와 고객 사례는 오로지 예시를 목적으로 작성된 것입니다. 실제 성능 결과는 특정 구성 및 운영 환경에 따라 달라질 수 있습니다.

IBM 제품 및 프로그램과 함께 사용되는 기타 제품 또는 프로그램을 평가 및 검증하는 것은 사용자의 책임입니다.

이 문서의 정보는 상품성에 대한 보증, 특정 목적의 적합성 여부 및 저작권을 침해하지 않는다는 보증 및 조건을 포함해 명시적 또는 묵시적 보증 없이 "있는 그대로" 제공됩니다. IBM 제품은 제공된 약정에 명시된 조항 및 조건에 따라 보증됩니다.

실제 사용할 수 있는 스토리지 용량은 비압축 및 압축된 데이터 모두에 대해 보고된 내용일 수 있으며 보고된 수치보다 더 낮을 수 있습니다.

재활용하십시오

82018782KRKO-00