

자동화된 APM을 활용하여 CI/CD를 가속화하고 애플리케이션 성능 향상



목차

01

핵심 요약

02

새로운 표준이 된
변화

03

수동 성능 관리가
실패하는 이유

04

자동화된 APM 실행

05

결론:
CI/CD 민첩성 극대
화

06

IBM 소속 회사인
Instana 소개

01 개요

지속적 통합 및 지속적 제공(Continuous Integration and Continuous Delivery, CI/CD)은 많은 조직에게 달성 목표가 되었습니다. 한편 Docker 및 Kubernetes와 같은 현대적인 기술은 널리 사용되는 도입 애플리케이션 환경이 되었습니다. 이러한 트렌드의 결과로 애플리케이션과 인프라가 점점 더 역동적으로 변하고 있습니다. 더 큰 확장성 요구 사항을 충족하고 빠르게 진화하는 애플리케이션 기능을 수행하기 위해 끊임없이 변하고 있습니다.

그러나 일반적인 조직은 여전히 CI/CD가 애플리케이션 제공 모델이 되기 이전에 설계된 성능 모니터링 기술에 의존하고 있습니다. 이러한 솔루션은 IT 직원이 수동으로 상당한 노력을 기울여야 하므로 조직이 직원 시간을 효과적으로 활용하고, IT 지출을 통제하고, 애플리케이션과 인프라에 대한 의미 있는 가시성을 확보하는 것을 어렵게 만듭니다.

조직이 애플리케이션 문제를 돌파하고 CI/CD 기반 애플리케이션 환경이 제시하는 기회를 충분히 실현할 수 있는 유일한 방법은 자동화된 성능 관리 전략을 채택하는 것입니다. 성능 관리의 각 단계에서 자동화를 도입하면 조직은 비용을 절감하면서 성과를 향상할 수 있습니다.

02 새로운 표준이 된 변화

우리는 급변하고 있는 세상에 살고 있다고 말하면 뻔한 이야기처럼 들릴 수 있습니다. 그러나, IT 조직의 관점에서 보면 오늘날 애플리케이션 환경의 변화 속도는 극도로 빠릅니다.

고객과의 상호작용에서 고객의 요구는 분명히 속도에 치우쳐져 있습니다. 회사가 더 빨리 맞춤화된 비즈니스 애플리케이션을 제공할수록 IT는 더 많은 가치를 제공할 수 있습니다. 이를 통해 실질적으로 IT의 ROI를 높일 수 있습니다.

이와 같은 단순한 현실 때문에 새로운 방법과 기술의 채택이 늘어나고 있습니다. 특히, 제공 속도와 비즈니스 역량 그리고 맞춤형 애플리케이션의 품질을 향상하는 자동화 작업인 CI/CD의 채택이 늘어나고 있습니다.

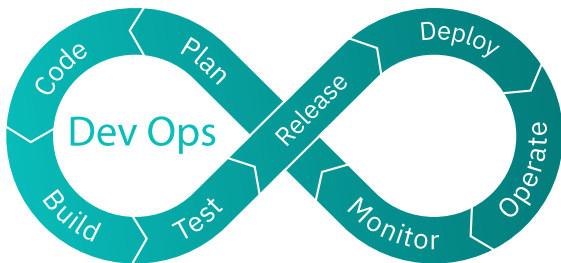


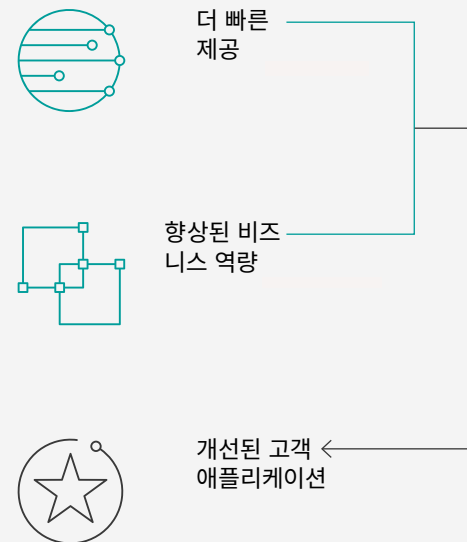
그림 1. 지속적 통합 및 제공 파이프라인

CI/CD 주기의 마지막 단계는 모니터링입니다. 그러므로 비즈니스 애플리케이션 모니터링이 더욱 자동화되고 원활해질수록 CI/CD 주기를 완료하고 애플리케이션 향상 주기를 다시 시작하기가 더 쉬워집니다.

역동적 애플리케이션 환경에서 변하지 않는 것은 환경이 끊임없이 변한다는 사실뿐입니다. 애플리케이션의 구조가 각 계층에서 지속적으로 변화됩니다. 새로운 호스트가 언제나 온라인 상태가 되었다가 사라지고 컨테이너로 인해 프로비저닝 작업이 훨씬 더 역동적으로 변화됩니다. 개발자는 운영 팀과 확인하지 않고 끊임없이 완전히 새로운 API와 서비스를 구축하고 프로비저닝합니다. 언제든 개선 작업이 수행되고 문제가 해결될 수 있으므로 애플리케이션 코드 조차도 변화될 수 있습니다. 팀이 CI/CD에 더 가까워질수록 변화의 빈도는 늘어납니다.

그 결과, 모니터링 대상인 환경과 보조를 맞추기 위해 성능 관리 구성, 모니터링 대시보드, 의존성 맵핑 및 알림 규칙도 자동으로 변화될 수 있어야 합니다. 그렇지 못하면 IT 팀은 환경을 관리하는 데 필요한 정확한 가시성을 확보하지 못하게 되어 조직이 사용자에게 영향을 줄 수 있는 큰 장애의 위험에 처하게 됩니다.

CI/CD 자동화로 인한 이점:



복잡한 의존성

이와 같은 끊임없는 변화는 다양한 구성요소 간의 의존성에 영향을 줍니다. 어떤 서비스든 데이터와 함께 고유한 수직적 소프트웨어 스택 또는 다른 서비스의 프로세싱에 의존합니다.

의존성을 항상 이해하는 것이 왜 중요합니까? 문제 해결 때문입니다! 복잡한 환경에서 문제의 근본 원인을 파악하려면 의존성 분석 작업을 수행해야 합니다. 느린 요청 또는 오류로 발생한 요청의 원인은 무엇일까요? 요청은 여러 프레임워크와 인프라에 걸쳐 여러 서비스를 기반으로 수행되므로 이 질문에 대답하려면 반드시 각 요청의 구조적 의존성을 이해해야 합니다. 그러나 마지막 단락에서 설명했듯이 의존성은 끊임없이 변합니다!

이러한 의존성을 수동으로 해석하는 일은 현실적으로 가능하지 않습니다. 특히, 코드 배포 또는 인프라 확장 때문에 의존성이 빠르게 변화되는 경우에 이는 실현 가능한 일이 아닙니다. 인간 운영자가 어느 순간에 수동으로 의존성 맵핑에 성공한 경우라도 금방 이러한 맵핑은 유효하지 않게 됩니다. 또한, 수동 의존성 해석은 리소스를 매우 많이 소모하며 최고의 엔지니어를 투입해야 하는 일입니다.

의존성을 항상 이해하는 것이 왜 중요합니까?

문제 해결 때문입니다!

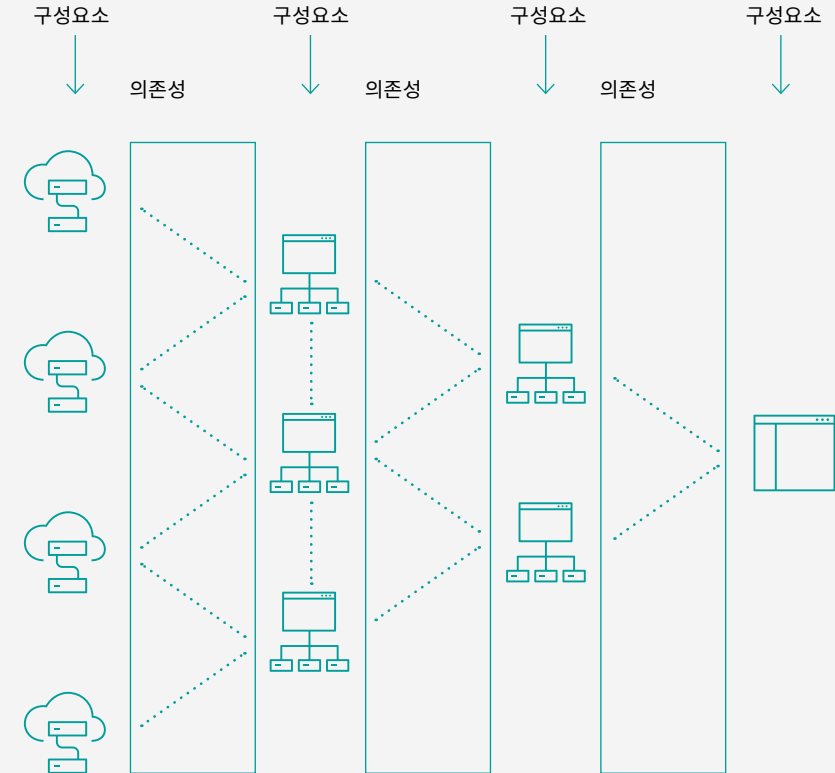


그림 2. 하드웨어, 소프트웨어 또는 코드 등 어떤 서비스든 데이터와 함께 고유한 수직적 소프트웨어 스택 또는 다른 서비스의 프로세싱에 의존합니다. 변화의 도미노 효과로 환경 전체에서 문제가 악화될 수 있습니다.

역동적 애플리케이션의 등장

애플리케이션 개발의 경우 속도 향상의 필요성 때문에 새로운 서비스를 빠르게 구성하고 제공하도록 지원하는 기술이 광범위하게 채택되었습니다.

특히, 지속적 통합 및 제공 파이프라인은 속도와 품질을 향상하는 주된 프로세스로 인식되고 있습니다. 또한, 이러한 프로세스를 강화하기 위해 컨테이너, 마이크로서비스, 서버리스 컴퓨팅, Kubernetes 오케스트레이션과 같은 새로운 아키텍처가 조사되고 채택되고 있습니다.

점점 더 많은 워크로드가 역동적 기술로 이동하면서(Docker Hub가 2014년에 출시된 이후 이 허브에서 한 달만에 80억 회의 풀이 발생했고 지금까지 총 1,300억 풀이 발생했음)¹ 애플리케이션 환경 안에서 변화의 속도와 범위와 관련하여 직시해야 할 새로운 현실이 생겨났습니다.

IT 산업은 CI/CD 용어와 방법론을 중심으로 통합되고 있습니다. 한편, 퍼블릭 클라우드 플랫폼은 이제 CI/CD 제공 파이프라인으로 쉽게 통합되는 Kubernetes 관리형 컨테이너 프로세싱을 서비스 방식으로 제공합니다. 분명한 것은 애플리케이션 구성과 제공의 세계에서 중대한 변화가 이루어지고 있다는 것입니다.

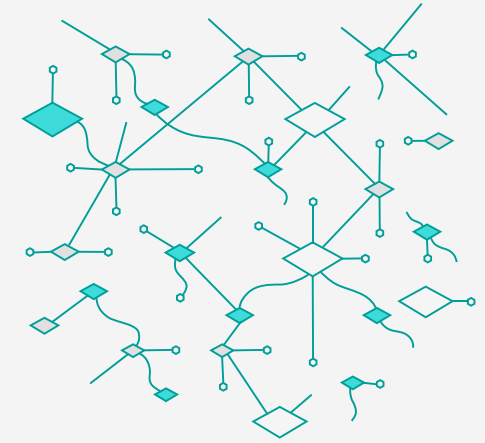


그림 1. 변화는 다양한 구성요소 간의 실제 의존성에 영향을 주고 몇 초만에 시스템을 중단시킬 수 있습니다.

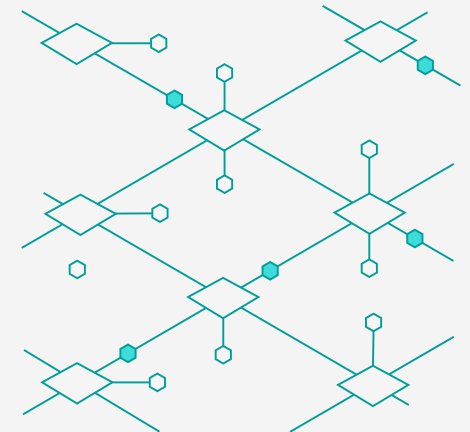


그림 2. Instana는 자동으로 관찰 가능성 지표를 수집하고 각 요청을 추적하고 각 프로세스의 프로파일을 작성합니다. 그리고 환경의 복잡성을 뚫고 어떻게 모든 것이 맥락에 따라 조화롭게 연결되는지 맥락에 따라 보여줍니다.

03

수동 성능 관리가 실패하는 이유

수동 성능 모니터링의 문제점

기존의 애플리케이션 성능 모니터링(application performance monitoring, APM)은 역동적 애플리케이션에 맞게 설계되지 않았기 때문에 개발 팀이 수동 코딩 작업을 통해 자체 모니터링 기능을 설정하도록 지원하기 위해 새로운 오픈소스 기술이 등장했습니다. 성능 지표를 제공하건, 애플리케이션 경로를 추적하건, 코드의 다른 세부 정보를 노출하건, 이러한 오픈소스 모니터링 툴은 프로덕션 성능 모니터링의 관점에서 고유한 문제점을 안고 있습니다.

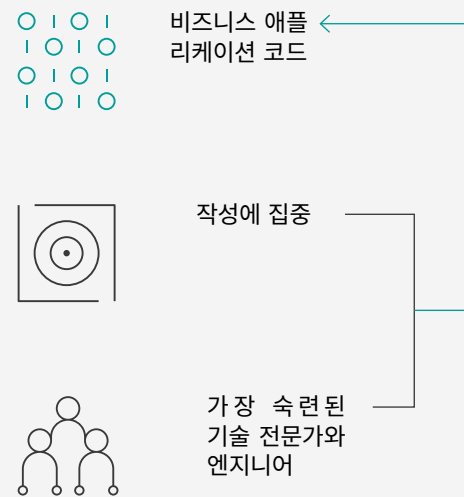
애플리케이션 성능을 수동으로 모니터링하려면 엔지니어는 다음과 같은 작업을 완료해야 합니다.

- 데이터 수집자 작성
- 분산된 요청을 추적하기 위해 수동 코드 추적 수행
- 데이터 저장소 구성
- 일반적으로 리버스 엔지니어링을 통해 의존성을 파악하고 지정
- 상관관계를 파악하기 위해 데이터 선택
- 상관관계 시각화를 위한 대시보드 구축
- 알림 규칙 및 임계값 구성

가장 큰 문제는 이러한 작업을 수행하는 사람이 일반적으로 회사에서 가장 숙련도가 높고 높은 급여를 받는 기술 전문가와 엔지니어라는 점입니다.

간단히 말해, 오픈소스는 성능 관리를 간소화하지만 너무 많은 수동 작업이 요구되므로 배포 속도 저하, 비용 상승, 추가 노동력에 대한 요구라는 문제가 발생할 수 있습니다. 귀중한 개발자가 비즈니스 애플리케이션 코드 대신 모니터링 코드를 작성할 경우 발생하는 기회 비용을 고려하면 문제는 더 심화됩니다.

회사에서 가장 숙련되고 가장 높은 급여를 받는 기술 전문가와 엔지니어가 모니터링 코드가 아니라 비즈니스 애플리케이션 코드를 작성하도록 지원하는 것을 목표로 삼아야 합니다.



자체 시스템을 구축하기 위한 진정한 노력

최근까지 모니터링 시스템을 수동으로 설정해도 문제가 발생하지 않았습니다. 애플리케이션 코드와 애플리케이션 인프라(미들웨어, 앱 서버 등)가 자주 변화되지 않았기 때문입니다. IT 팀은 박스를 프로비저닝하고 IP 주소를 설정하고 소프트웨어를 로드하고 모니터링을 설정한 후 수년간 이러한 시스템을 그대로 두면 되었습니다.

또한, 기존 기술을 기반으로 구축된 애플리케이션의 규모가 급격하게 변동할 가능성도 없었습니다. 어느 순간에든 실행되는 애플리케이션 인스턴스와 호스트 서버의 수는 일반적으로 변하지 않았습니다. 따라서, 정적인 환경에서 기술 스택을 수동으로 구성해도 조직은 문제 없이 성능을 모니터링하고 관리할 수 있었습니다.

그러나 워크로드가 컨테이너와 같은 기술, 그리고 CI/CD와 같은 방법론을 기반으로 하는 역동적 환경으로 이동하는 경우, 수동 성능 관리 전략과 자체 구축 솔루션은 실패하게 됩니다. 그 이유는 여러 가지입니다.



규칙 악화

환경이 끊임없이 변화되면, 모니터링 툴이 애플리케이션과 서비스 상태가 정상인지 결정하는 데 사용하는 규칙도 지속적으로 변경되어야 합니다. 규칙이 변경되지 않으면(이런 경우는 규칙 업데이트를 위해 수동 개입에 의존할 경우 규칙이 변경되지 않을 가능성이 큼), 규칙은 곧 악화됩니다. 예를 들어, 새로운 서비스를 배포하거나 오케스트레이터가 워크로드를 이동할 경우 규칙이 업데이트될 때까지 상태 점검 규칙은 환경의 의존성을 정확하게 해석하지 못합니다.

규칙이 수동으로 업데이트되지 않으면 모니터링 알림과 인사이트는 더 이상 유효하지 않은 구성을 기반으로 제공될 것입니다. 이러한 악화로 인해 가시성이 저해되고 인프라 또는 서비스 장애의 위험이 증가합니다.

속도를 저하시키는 수동 모니터링

추적 및 모니터링 코드를 작성하는 것과 같은 작업에는 시간이 너무 많이 소요됩니다. 수동으로 모니터링 정보를 해석하고 애플리케이션 코드 또는 환경의 아키텍처가 변할 때마다 성능 관리 규칙을 수동으로 업데이트하는 작업도 시간이 너무 많이 소요됩니다.

간단히 말해, 자동화 없이 인간은 급변하는 환경을 지원할 수 없습니다. 성능을 수동으로 관리하려고 하면 애플리케이션 릴리스 주기가 상당히 느려질 것입니다. 이것이 비즈니스에 의미하는 바는 IT 직원의 값비싼 전문 지식을 부적절하게 사용하게 된다는 점입니다.

급변하는 사용자의 요구 사항과 보조를 맞추고자 하는 조직은 반드시 소프트웨어 제공 속도를 높일 수 있어야 합니다. 비즈니스의 요구 사항은 이러한 수요를 분명히 높일 것이므로 앞으로 애플리케이션은 분명히 더욱 역동적으로 변할 것입니다.



자동화된 APM 실행

수동 모니터링이 이렇게 수고스러운데 왜 모든 조직이 아직 모니터링을 자동화하지 않았는지가 궁금해 집니다.

그 이유는 세 가지입니다.

- 최근까지 완전 자동 모니터링 기술이 존재하지 않았습니다.
- 소프트웨어 엔지니어는 문제가 발생하면 자체 해결책을 코딩을 통해 마련하는 경향이 있습니다. 이 때문에 코드 기반 모니터링 접근법을 취하게 됩니다. 이러한 접근법은 단기간 동안 효과가 있지만 지속 가능하거나 확장 가능하지 않습니다.
- IT 팀은 기존 모니터링 툴에 이전에 투자했으므로 새로운 고속 환경에서도 이러한 툴이 효과를 발휘하기를 바라며 이를 활용하려고 합니다. 그러나, 안타깝지만, 기존 모니터링 툴은 새로운 환경에서는 효과적으로 작동하지 않습니다.

이러한 이유를 기억하면서 완전 자동화된 APM 솔루션에 포함되어야 할 기능을 살펴보겠습니다.

- 전체 인프라 및 애플리케이션 스택의 자동 발견 및 모니터링. 이 기능은 현재 사용되지 않으며 릴리스 프로세스를 느리게 하는 CI/CD 프로세스의 핵심 요소입니다.
- 변화가 발생할 경우 자동 업데이트되는 복잡한 의존성의 실시간 맵핑. 이 기능은 성능 문제를 신속하게 해결하기 위한 근본 원인 분석에 반드시 필요합니다.
- 성능 문제를 신속하게 자동으로 파악. 긍정 오류를 최소화하고 애플리케이션 제공 지연을 방지하는 유일한 방법은 자동 규칙 구성 및 모니터링을 기반으로 성능 문제를 신속하게 파악하는 것입니다.

- 애플리케이션이 정상적으로 행동하도록 역동적 기준선을 정립한 다음 이러한 기준선에 따라 이상을 발견하기 위해 머신 러닝과 AI를 사용하는 규칙 구성 및 알림. 이 기능을 활용하면 인간이 지루하게 구성, 모니터링, 데이터 해석을 수행할 필요가 없으며 긍정 오류 알림을 최소화할 수 있습니다.

- 자동 모니터링 설정—에이전트 배포, 코드 인스트루먼트이션, 인프라 발견 등. 인간 관리자가 데이터를 수집하기 위해 기울이는 노력을 최소화하면서 수집된 데이터를 극대화합니다.

기존의 성능 관리 툴과 현대적인 오픈소스 모니터링 툴 모두 이러한 자동화 기능이 없습니다. 구체적인 작업은 다르지만, 모니터링을 최적화하려면 이 두 가지 툴 모두 너무 많은 수동 구성, 프로그래밍, 설정, 관리 작업을 요구합니다.

귀사의 CI/CD는 최대한으로 민첩성을 구현하고 있습니까?

역동적 애플리케이션을 성공적으로 관리하려면 자동화된 모니터링을 활용하는 것 이상의 노력이 필요합니다.

IT 팀은 다음과 같은 질문을 통해 어느 정도로 모든 성능 관리 작업을 성공적으로 자동화했는지 지속적으로 평가해야 합니다.

- 새로운 릴리스를 내놓은 후 애플리케이션 성능에 대한 충분한 가시성을 확보할 때까지 시간이 얼마나 걸립니까?
- 새로운 애플리케이션 또는 서비스 배포가 이루어질 경우 모니터링 규칙을 업데이트하는 데 시간이 얼마나 걸립니까?

- 개발자가 추적 코드를 작성하는 데 얼마만큼의 시간과 노력을 기울입니까?

- 매달 또는 매분기마다 놓치고 있는 성능 인시던트 또는 가용성 인시던트는 몇 개입니까?

- 알림 폭풍(짧은 기간 동안 급격하게 많이 알림이 발생하는 경우)에 어떻게 대처하고 있습니까? 알림으로 인한 피로를 느끼지 않으면서 각 알림에 효과적으로 대응할 수 있습니까? 알림을 통해 신속하게 근본 원인을 추적하여 다수의 알림이 동일한 기저 문제로 인해 발생한다는 것을 알아낼 수 있습니까?

- 모니터링 및 성능 관리 프로세스가 다른 애플리케이션 제공 파이프라인만큼 자동화되었습니까? 그렇지 않다면 어떻게 더 자동화할 수 있을까요?

IT 팀의 규모에 상관없이 효과적인 성능 관리 전략에는 자동화가 반드시 필요합니다. 이를 통해 새로운 비즈니스 서비스를 고속으로 제공할 수 있습니다.

마이크로서비스, 클라우드 컴퓨팅, 컨테이너의 시대에 탄생한 자동화된 성능 관리 솔루션인 IBM Observability by Instana는 CI/CD의 이점을 진정으로 실현하도록 지원합니다.

역동적인 CI/CD 기반 애플리케이션 환경의 성능을 관리하도록 특별히 설계된 IBM Observability by Instana는 AI 및 자동화를 활용하여 수동으로 노력을 기울이지 않아도 포괄적이고 활용 가능한 인사이트를 제공합니다.

06

IBM 소속 회사인 Instana 소개

IBM 소속 회사인 Instana는 복잡하고 현대적인 클라우드 네이티브 애플리케이션을 운영하는 기업에게 **Enterprise Observability Platform**을 통해 **자동화된 애플리케이션 성능 모니터링** 기능을 제공합니다. 이러한 애플리케이션이 모바일 디바이스 또는 IBM Z® 메인프레임 컴퓨터를 포함하여 온프레미스 또는 퍼블릭 또는 프라이빗 클라우드 중 어디에 있건 상관없습니다.

AI를 활용하여 하이브리드 애플리케이션 내부의 심층적인 맥락 의존성을 찾아내는 Instana와 함께 현대적인 하이브리드 애플리케이션을 제어해 보십시오. 또한, Instana는 폐쇄 루프 DevOps 자동화를 지원하는 개발 파이프라인에 대한 가시성을 제공합니다.

이러한 기능은 고객이 애플리케이션 성능을 최적화하고 혁신을 지원하고 위험을 완화하는 데 필요한 활용 가능한 피드백을 제공하므로 DevOps는 효율성을 향상하고 소프트웨어 제공 파이프라인에 가치를 추가하는 동시에 서비스 및 비즈니스 수준 목표를 충족할 수 있습니다.

[자세히 보기 →](#)



© Copyright 2021 Instana, an IBM Company

IBM Corporation
(07326) 서울특별시 영등포구 국제금융로 10 서울국제금융센터(3 IFC)
Tel. 02-3781-5114

Produced in the United States of America
2021년 4월

IBM 및 IBM 로고는 전세계 여러 국가에 등록된 International Business Machines Corporation.의 상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다. 현재 IBM 상표 목록은 [ibm.com/trademark](https://www.ibm.com/trademark)에 있습니다.

Instana는 IBM 소속 회사인 Instana, Inc.의 상표입니다.

이 문서는 최초 발행일을 기준으로 하며, 통지 없이 언제든지 변경될 수 있습니다. IBM이 영업하는 모든 국가에서 모든 오퍼링이 제공되는 것은 아닙니다.

이 문서의 정보는 상품성, 특정 목적에의 적합성에 대한 보증 및 타인의 권리 침해에 대한 보증이나 조건을 포함하여(단, 이에 한하지 않음) 명시적이든 묵시적이든 일체의 보증 없이 “현상태대로” 제공됩니다. IBM 제품은 제공 조건으로 체결된 계약의 이용 약관에 따라 보증됩니다.

1 Docker knits together Hub stats, says Pulls over 8 billion(Docker는 허브 통계를 취합하여 80억 회 이상의 풀이 발생했다고 말함), DevClass, 2020년 2월 5일.