

솔루션 개요

API 설계자를 위한 마이크로서비스 가이드

설계자를 위한 마이크로서비스 가이드

CA 테크놀로지스의 도구로 마이크로서비스 및
API를 더욱 빠르게 개발

CA 테크놀로지스는 기업에 확장 가능하고 안전한
마이크로서비스 솔루션을 위한 검증된 플랫폼을
제공합니다.

개요

당면 과제

애플리케이션 경제에 직면한 수많은 기업에서는 변화하는 요구 사항에 빠르게 적응하기 위해 애플리케이션을 재구축해야 합니다. 하지만, 덩치가 큰 애플리케이션을 출시(및 지원)하는 기존의 방법으로는 충분하지 않습니다. 오늘날의 엔터프라이즈 설계자와 애플리케이션 담당 VP는 다음과 같은 과제를 안고 있습니다.

- 최신 애플리케이션을 배포 및 출시하는 데 걸리는 시간을 수 개월 또는 수 년에서 며칠 또는 몇 주로 줄이는 방법은 무엇인가? 그리고 앱 업데이트 시 다운타임을 최소화하는 방법은 무엇인가?
- 최신 애플리케이션을 구축하려면 언어적 기반이 서로 다른 여러 개발 팀을 어떻게 활용해야 하는가?
- 요구 사항의 변경에 따라 애플리케이션을 확장하면서 이로 인해 발생하는 인프라 비용을 최소화하는 방법은 무엇인가?

기회

이러한 당면 과제가 등장한 원인은 최신 애플리케이션을 구축하는 데 민첩성과 확장성이 더욱 중요해졌기 때문입니다. 하지만 기존의 애플리케이션 개발 방식으로는 이러한 환경을 지원할 수 없습니다.

CA 테크놀로지스는 전체 수명 주기 API 관리를 마이크로서비스로까지 확장했습니다. 마이크로서비스는 최고의 솔루션을 함께 사용하여 최신 아키텍처를 위한 플랫폼과 민첩성 및 확장성을 갖춘 안전한 환경을 제공해 주는 통합 솔루션입니다. CA는 기업에서 베스트 프랙티스와 업계 최고의 기술을 사용하여 최신 아키텍처를 더욱 빠르고 실용적으로 구축하도록 지원합니다.

이점

CA 테크놀로지스는 마이크로서비스 아키텍처를 위한 설계, 개발, 배포, 모니터링 및 관리 플랫폼을 제공합니다.

- **CA Live API Creator** - 마이크로서비스 설계 및 배포를 위해 사용이 간편한 솔루션을 제공하여 최신 애플리케이션의 시장 출시 기간을 앞당겨 줍니다.
- **CA Microgateway** 및 **CA API Gateway** - 마이크로서비스 아키텍처에 필요한 계층화된 보안 및 관리 인프라를 제공하여 엔터프라이즈 IP를 안전하게 보호합니다.
- **OAuth Toolkit** - CA API Gateway의 구성 요소로, 아이덴티티 및 액세스 관리 솔루션의 인증 및 통합을 통해 앱 사용자의 고객 환경을 개선하고 엔터프라이즈 보안을 포괄적으로 향상시킵니다.
- **CA Application Performance Management** - 컨테이너화된 시스템을 모니터링하여 애플리케이션 가용성과 뛰어난 고객 환경을 능동적으로 확보합니다.

섹션 1:

마이크로서비스 및 디지털 혁신

이 문서에서는 기업에서 애플리케이션 경제의 요구 사항을 충족하는 데 필요한 민첩성을 마이크로서비스를 통해 제공하는 방법을 설명합니다. 디지털 혁신은 쉽게 이루어지지 않습니다. 마찬가지로, 최신 애플리케이션 아키텍처를 구축하기 위한 준비 및 구현 과정도 만만치 않습니다. 이제 다음 질문에 대한 답을 알려 드리겠습니다.

- 마이크로서비스란?
- 마이크로서비스가 중요한 이유는?
- 마이크로서비스가 정말로 필요한가?
- 마이크로서비스 사용 방법은?
- 마이크로서비스와 API의 관계
- 마이크로서비스의 보안 방법은?

마이크로서비스란?

디지털 경제에서 모놀리식 애플리케이션은 더 이상 효율적이지 않습니다. 기업 전체의 개발 팀은 개발 시간, 배포 및 확장성 때문에 대규모 애플리케이션을 처리하는 데 어려움을 겪고 있습니다. 데브옵스 프레임워크와 애자일 방법론이 광범위하게 채택됨에 따라 개발 팀은 복잡한 애플리케이션 사일로를 단순한 코드 블록으로 나누기가 까다롭다는 것을 느끼게 되었고 여기에서 마이크로서비스가 탄생했습니다.

설계자가 고안한 설계 패턴은 복잡한 애플리케이션을 간단하고 세분화해 주고 재사용 및 상호 운용이 가능하며 서로 독립적으로 수정 및 배포가 가능한 프로세스였습니다. 이렇게 세분화된 프로세스를 마이크로서비스라고 합니다.

마이크로서비스 구성 요소

[API Academy](#)를 통해 마이크로서비스 아키텍처에서 몇 가지 공통된 특징을 파악했으며, 업계에서는 이러한 특징을 표준으로 채택했습니다. 이를 바탕으로 마이크로서비스는 다음과 같은 중요한 특징을 보입니다.

- 작은 크기
- 메시징 사용
- 컨텍스트로 구분
- 자율적으로 개발
- 독립적 배포 가능
- 탈중앙화
- 언어에 구애받지 않음
- 자동화된 프로세스로 구축 및 릴리스

마이크로서비스 및 마이크로서비스 아키텍처는 서로 혼용되기도 하지만, 정확히 같지는 않습니다. 마이크로서비스 아키텍처는 고도로 자동화된 엔지니어링 스타일의 변경 가능한 소프트웨어 시스템으로, 맞춤 기능의 마이크로서비스로 구성됩니다.

보통 이러한 구성 요소는 [Docker 컨테이너](#)에 배포되어 Docker 컨테이너를 여러 개 사용하고 있어도 중앙 집중식 런타임 환경을 제공합니다. Docker 컨테이너를 사용하면 관리 부담이 줄어드는 장점이 있습니다.

마이크로서비스가 중요한 이유?

디지털 경제에서 생존하고자 하는 모든 디지털 기업은 속도와 확장을 원합니다. 회사의 빠른 시장 진입이 중요하다면, 증가하는 고객의 요구 사항을 지원하기 위해 적절하게 확장할 수 있는 능력도 중요합니다. 하지만 핵심은 확장에서의 속도와 안전성입니다. 안전을 해치지 않고서 속도와 확장을 달성할 때에만 진정한 성공이라고 할 수 있습니다.

애자일 및 데브옵스 모델은 소프트웨어 자산에 대한 중앙화를 벗어나 분산된 소유권을 지원하며 변경 결과를 빠르게 반영하며 빠른 배포를 촉진합니다. 하지만 복잡한 모놀리식 애플리케이션을 독립적 단위로 현명하게 나누려면 설계 전략을 수립해야 하며 그것이 바로 마이크로서비스입니다. 거대한 애플리케이션을 마이크로서비스로 분할하면 개발 팀은 업데이트와 독립적 배포를 민첩하게 진행할 수 있습니다. 이를 통해 크고 복잡한 빌드에 대한 의존성이 사라지고 수요를 충족할 확장성을 갖추기 위해 지나치게 정교한 아키텍처를 마련할 필요도 없어집니다.

기업에서 마이크로서비스로 혜택을 얻을 수 있을까요?

기업마다 요구 사항이 다르며, 마이크로서비스가 솔루션을 제공하는 데 있어 바람직한 패턴일 때도 있고, 모놀리식 애플리케이션이 더 적합할 때도 있습니다.

모놀리식 애플리케이션의 장점은 다음과 같습니다.

- 일반적으로 초기 구축 용이
- 일반적으로 초기 테스트 용이
- 통합 개발 환경(IDE)을 사용하는 경우 더 적합할 수 있음

하지만 다음과 같은 단점도 있습니다.

- 코드베이스가 커지면 관리가 어려움
- 커진 코드베이스 때문에 시간이 갈수록 반복 처리 능력이 저하됨
- 확장이 어려움(애플리케이션의 구성 요소 하나만 확장 문제가 있어도 전체 애플리케이션의 인프라를 확장해야 함)
- 코드베이스를 기반으로 하기 때문에 혁신이 어려움
- 학습 곡선이 가파르고 코드베이스를 기반으로 하기 때문에 새로운 프로그래머를 투입하기 어려움

이러한 점 때문에 조직은 적절한 민첩성을 확보하고 확장하는 데 많은 어려움을 겪을 수 있습니다. 하지만 기존 모놀리식 환경을 이미 사용하고 있으며 애자일 방식이 필요하지 않다면 마이크로서비스 기반 솔루션이 중요하지 않을 수도 있습니다.

마이크로서비스를 구현하고 싶은데 어떤 장점이 있으며, 무엇을 희생해야 할까요?

마이크로서비스 아키텍처를 구축하면 애플리케이션 경제에서 여러 장점이 있습니다.

마이크로서비스 기반 애플리케이션의 장점은 다음과 같습니다.

- 큰 규모의 애플리케이션에 더 나은 아키텍처: 전체 애플리케이션에 영향을 주지 않고 구성 요소를 구축/전환할 수 있습니다. 또한 문제가 생겨도 해당 구성 요소만 영향을 받습니다.
- 더욱 민첩한 솔루션: 필요한 경우 애플리케이션의 세그먼트를 더 쉽게 전환할 수 있습니다.
- 학습이 용이함: 각 구성 요소가 작고 격리되어 있으므로, 구성 요소의 작업과 작업 방식, 애플리케이션에서 상호작용하는 방식을 확인하기 더 간편합니다.
- 확장이 용이함: 전체 애플리케이션이 아니라 더 높은 확장성이 필요한 구성 요소만 변경되므로 기업은 상당한 비용을 절감할 수 있습니다.

하지만 다음과 같은 단점도 있습니다.

- 모놀리식 애플리케이션보다 파트 이동이 많아 모니터링이 매우 중요함
- 커진 코드베이스에 대한 최신 백엔드 인프라 요구 사항이 늘어남
- 전체 애플리케이션 테스트가 어려움(단, 개별 구성 요소 테스트는 훨씬 쉬워짐)
- 엔드포인트 대신 해결해야 하는 보안 요구 사항이 발생함

디지털 혁신을 추구하고 민첩성을 높이려는 기업이라면 마이크로서비스의 장점이 단점보다 대체로 더 많습니다.

어떻게 하면 기업에서 마이크로서비스를 실행할 수 있을까요?

애자일 엔터프라이즈의 성숙도를 평가합니다. 조직이 애자일 방식을 사용하고 데브옵스를 채택했거나 혹은 채택할 계획이라면 마이크로서비스를 수용할 준비가 된 것입니다.

작은 개발자 그룹을 여러 개 만듭니다. 작은 개발자 그룹이 작은 서비스/API 집합을 효율적으로 소유하고 작업할 수 있도록 합니다. 이를 통해 느슨한 결합 및 독립적 배포가 가능해집니다.

도메인 중심의 설계를 채택합니다. 대규모 애플리케이션을 비즈니스 기능 또는 직무에 기반한 간단한 서비스로 분할합니다. 서비스를 세분화할수록 더 효율적입니다.

마이크로서비스와 API는 어떤 관련이 있을까요?

마이크로서비스 구성 요소는 시스템의 다른 구성 요소와 통신할 수 있을 때만 가치가 있습니다. 즉, 각각의 인터페이스 또는 API가 있어야 합니다. 높은 수준의 코드 분리, 독립성 및 모듈화를 달성해야 하는 것처럼 API 즉, 구성 요소 인터페이스 역시 느슨하게 결합되도록 해야 합니다. 그렇지 않으면 두 마이크로서비스를 서로 독립적으로 배포할 수 없습니다. 이러한 독립적 배포는 속도와 안전의 균형을 위한 기본 목표입니다.

마이크로서비스 앞에 있는 API 계층은 애플리케이션에서 세분화된 마이크로서비스를 구분하므로 클라이언트 측 애플리케이션(예: 모바일)에 대한 지원을 활용할 수 있습니다. API 계층은 마이크로서비스 오케스트레이션과 보안 적용에 적합합니다.

마이크로서비스 보안 방법

거의 모든 마이크로서비스 구현에서 발견되는 일반적인 패턴은 마이크로서비스가 제공하는 API 엔드포인트에 API 게이트웨이로 보안을 적용하는 것입니다. 최신 API 게이트웨이는 마이크로서비스에 필요한 추가적인 핵심 기능인 변환 및 오케스트레이션을 제공합니다. 마지막으로, 대부분의 성숙한 구현 환경에 있는 API 게이트웨이는 서비스 검색 도구와 결합되어 마이크로서비스 클라이언트의 요청을 라우팅합니다. 마이크로서비스 아키텍처는 자유도가 매우 높습니다. 복잡한 엔터프라이즈 애플리케이션을 위한 아키텍처가 구현된 성숙한 마이크로서비스 조직에서는 수백 개의 마이크로서비스가 배포된 경우가 일반적입니다. 이 경우 고려해야 하는 매우 중요한 요소가 바로 보안입니다. 거의 모든 마이크로서비스 구현을 확인해 보면, 여러 마이크로서비스가 제공하는 API 엔드포인트를 적절한 API를 사용하여 보호하는 것을 알 수 있습니다. 마이크로서비스가 제공하는 API는 서로를 호출하거나, 프론트엔드, (공개된 측) API에서 호출하거나, 모바일 애플리케이션, 웹 애플리케이션 및 파트너 시스템과 같은 API 클라이언트에서 직접 호출할 수 있습니다. 널리 권장되는 접근법은 OAuth 공급자와 결합된 적절한 API 게이트웨이를 사용하여, 마이크로서비스 지원 시스템의 공개 측 API 엔드포인트 호출에 보안을 적용하는 것입니다. API 게이트웨이는 모든 마이크로서비스 아키텍처의 핵심 구성 요소이며 서비스 구현 및 모든 소비 클라이언트 간의 공통 브리지입니다.

API 게이트웨이는 다음을 제공합니다.

- 인증, 권한 부여 및 위협 차단을 위한 중앙 집중식 보안 적용.
- 여러 프로토콜에서 보호된 리소스에 대한 라우팅 및 중재.
- 비즈니스 수준 속도 제한 및 할당량 적용을 위한 서비스 수준 관리.
- 서비스 호출 감소를 위한 서비스 오케스트레이션.
- 모놀리식 백엔드에서 각 애플리케이션별 인터페이스를 노출하기 위한 서비스 외관.

게이트웨이가 마이크로서비스 대신 보안을 적용합니다. 하지만 보통은 OAuth 공급자가 보안 세션을 관리하고 API 게이트웨이와 나란히 작업합니다.

섹션 2:

마이크로서비스에 적합한 CA 테크놀로지스 도구

앞서 언급한 대로, 마이크로서비스에는 이동하는 파트가 많으며, 모든 파트가 원활하게 작동해야 합니다. CA 테크놀로지스는 이러한 상황을 전체적으로 살펴보고, 전체 수명 주기 API 관리를 플랫폼으로 삼아 완벽한 솔루션을 제공합니다.

CA Live API Creator는 CA의 자동화된 하위 코드 대체 개발 솔루션입니다. 이 솔루션은 도메인 중심 마이크로서비스 및 REST/JSON API를 애플리케이션 백엔드로 생성하고 노출하여 레거시는 물론 최신 데이터 소스 및 앱의 기존 데이터와 기능에 대한 액세스를 제공합니다. 또한 개발자가 포인트 앤 클릭 방법을 사용해 다양한 데이터 소스 전체에서 데이터를 결합하는 새로운 REST 엔드포인트를 생성할 수 있도록 지원합니다. API 소유자는 선언적 비즈니스 규칙, JavaScript 이벤트 처리, 역할 기반 보안 및 대화형 테스트로 API를 확장할 수 있습니다. 또한 이미 API 관리 기능을 도입한 회사는 CA Live API Creator를 통해 API 수명 주기 범위를 확장함으로써 단순히 기존 게이트웨이 및 포털 서비스를

관리하고 시행하는 단계에서 벗어나 데이터 계층에 가까운 API까지 생성할 수 있습니다. CA Live API Creator를 사용하면 내부 애플리케이션, 모바일 개발 프로젝트, DaaS(Data-as-a-Service) 노출, 사물인터넷(IoT) 활성화 및 파트너 통합을 위한 애플리케이션 백엔드를 신속하게 만들 수 있습니다.

마이크로서비스의 관점에서 이 솔루션은 다음 몇 가지 중요한 과제를 해결합니다.

- **모듈성:** CA Live API Creator를 사용하여 대형 애플리케이션을 리소스라고 불리는 독립적 단위로 분해함으로써 데이터 통합, 비즈니스 로직 및 강력한 API 상호 작용 계층과 같이 앱 개발에 필요한 모든 것을 제공합니다. 리소스는 기본 스키마와는 별개인 메시지 기반 RESTful API입니다.
- **제공 속도:** 리소스 정의는 여러 데이터베이스의 여러 개체를 통합하는 포인트 앤 클릭입니다.
- **제로 배포:** 마이크로서비스는 배포와 관련된 지원을 완전히 제거합니다. 정의된 리소스는 컴파일 및 배포 없이 저장을 클릭하는 즉시 실행 가능합니다.
- **자동화된 배포:** 마이크로서비스를 내보내고(예: 개발 단계에서) 스크립트를 이용해 이를 가져올 수 있습니다(예: 프로덕션 단계로).
- **응집성:** 종속성을 자동화해서 처리하므로 마이크로서비스 하나를 배포한다고 해도 다른 마이크로서비스에는 아무런 영향을 미치지 않습니다.
- **작업 분리:** CA Live API Creator는 마이크로서비스 생성과 비즈니스 로직을 분리합니다. 이때 비즈니스 로직은 기본 도메인에서 정의됩니다.

CA Microgateway는 컨테이너화된 경량의 게이트웨이이며, 고도로 분산된 환경에서 확장할 수 있도록 설계되었습니다. 서비스 검색, 라우팅, 속도 제한 및 최종 단계 보안 기능을 제공하여 일반적인 마이크로서비스 패턴을 지원하며, 개발자가 디자인 타임에 제공된 정책 템플릿을 사용하여 손쉽게 배포 및 구성할 수 있습니다. 또한 스크립트화된 프로덕션 배포를 위한 업계 표준 DevOps 도구와 통합되며, 새 템플릿을 생성하고 사전 구성된 신규 컨테이너에 포함하여 사용자 지정/새 사용 사례를 지원하도록 확장할 수 있습니다.

CA Microgateway를 사용하는 개발자는 마이크로서비스 환경 내에서 새로운 패턴이 발생할 때 이러한 패턴을 수용할 수 있으며, 크고 작은 마이크로서비스 아키텍처에 필요한 트래픽 관리와 조정 기능을 이용할 수 있습니다. 마지막으로, CA Microgateway는 규제 대상 산업에 필요한 보안 및 내결함성을 제공합니다.

CA API Gateway(Essentials, Enterprise 및 CA Mobile API Gateway 포함)는 정책 관리와 런타임 정책 적용을 결합하고 비즈니스 및 최종 사용자(위치에 관계 없이) 간의 정책 적용 지점을 제공함으로써 엔터프라이즈급 마이크로서비스를 위한 업계 선도적인 게이트웨이 기능을 제공합니다. 기존 IAM(아이덴티티 및 액세스 관리) 솔루션과 통합되어 플러그앤플레이 솔루션을 만들 수 있는 CA API Gateway를 통해 기업은 내부 개발자는 물론 서드파티 개발자에게 기업의 데이터 및 애플리케이션을 선택적으로 공개할 수 있습니다. CA API Gateway는 확장이 용이하고 고가용성이 필요한 장애 조치 환경에 배포할 수 있기 때문에 마이크로서비스에 이상적인 Docker를 포함한 다양한 폼 팩터로 배포할 수 있습니다. 또한 이 솔루션에는 레거시부터 REST 및 JSON에 이르는 다양한 프로토콜 간의 완벽한 변환과 레거시부터 모바일, 클라우드, 소셜에 이르는 브리지를 제공하는 프로토콜 브리징도 포함됩니다.

마이크로서비스 아키텍처 배포의 관점에서 CA Microgateway 및 CA API Gateway는 다음 몇 가지 중요한 과제를 해결합니다.

- **보안:** 앞서 언급했듯이 이 솔루션은 정책 적용의 중심부 역할을 할 수 있습니다. API 게이트웨이를 이용하여 모든 API/마이크로서비스 액세스에 항상 보안을 적용하는 것이 더 효과적이며, 대개의 경우 서비스 간 호출에 API 게이트웨이를 도입할 때의 오버헤드가 무시할 수준임을 감안하면 충분한 가치가 있습니다.
- **변환 및 오케스트레이션:** CA API Gateway를 사용하면 백엔드 마이크로서비스를 조율하고 세세한 부분은 개발자 친화적인 인터페이스를 적용하여 불편을 해소할 수 있는 API 인터페이스를 구성할 수 있습니다.
- **라우팅:** CA API Gateway는 클라이언트 앱에서 마이크로서비스로 라우팅할 때 발생하는 복잡성을 없애 줍니다. 마이크로서비스와 연결된 외부 URI가 요청되면, 솔루션이 서비스 검색 시스템의 HTTP 또는 DNS 인터페이스와 연결하여 API 클라이언트를 올바른 서비스로 라우팅합니다.

CA Microgateway는 배포에 최적화되어 있으며, 동일한 PaaS 및 컨테이너 관리 환경 내에서 CA Microgateway가 관리하는 마이크로서비스에 따라 확장됩니다. CA Microgateway는 종종 CA API Gateway와 결합되어 애플리케이션 계층 또는 네트워크의 가장자리에 배포되며, OAuth 서버 및 중앙 집중식 로깅/감사 시스템 등의 기존 인프라와 통합됩니다. CA API Gateway는 마이크로서비스 환경에서 CA Microgateway와 이상적으로 결합되지만, 기존의 모놀리식 및 API 중심(마이크로서비스 이외의) 아키텍처에도 적용됩니다.

CA의 **OAuth Toolkit**은 업계 최고의 CA API Gateway와 함께 실행됩니다. OAuth Toolkit은 OAuth 공급자와 토큰 관리 시스템을 통해 웹, 모바일 및 기타 애플리케이션의 마이크로서비스 액세스를 제어합니다. OAuth Toolkit을 사용하면 기존 아이덴티티 인프라를 확장하여 이러한 OAuth 공급자 기능을 제공할 수 있으며, 뛰어난 확장성을 지원합니다. 이 솔루션에 포함된 기능은 다음과 같습니다.

- 2단계 및 3단계 OAuth 흐름 모두에서 액세스 토큰을 발급하기 위한 OAuth 권한 부여 서버
- API 액세스 제어 및 정책 시행을 위한 OAuth 리소스 서버
- OAuth 클라이언트 및 사용자 구현을 위한 사용자 지정 가능한 템플릿
- 널리 사용되는 모든 IAM 및 SSO(Single Sign-On) 솔루션과의 통합
- OAuth와 다른 액세스 제어 표준 간의 연결 기능
- 토큰 유형(예: JWT)을 선택하는 기능
- 맞춤형 사용자 환경에서 사용자 지정 핸드셰이크를 구현하는 기능

CA의 OAuth Toolkit을 사용하면 마이크로서비스에 배포된 인증 메커니즘을 구축하여 안전한 솔루션을 완성할 수 있습니다.

CA APM(Application Performance Management)은 동적 마이크로서비스와 컨테이너의 단기적 특성을 관리하는 독자적인 아키텍처를 제공합니다. CA APM은 모놀리식 시스템에 가장 적합한 기존의 정적 토폴로지 매핑과 인스트루멘테이션이 마이크로서비스와 관련성이 부족하다는 점을 인지하고 있으며, 혁신적인 미래형 접근 방식을 통해 컨테이너화된 시스템을 관리합니다. 이 접근 방식은 현대 시스템의 복잡성, 특히 마이크로서비스의 상호 종속성과 커뮤니케이션 흐름에 대한 빠르고 단순한 구성과 가시성에 바탕을 두고 있습니다.

마이크로서비스 아키텍처에서 CA APM은 다각적인 모니터링 솔루션 역할을 합니다. 기본 서비스로 제공되는 에이전트 미사용 모니터링 기능은 컨테이너 및 종속성의 검색을 자동화하여 CPU 포화도, 오류 비율 및 지연 시간과 같은 주요 상태 지표를 즉각적으로 가시화합니다. 이 솔루션 자체만으로 강력한 서비스이며, 컨테이너 속성 및 데이터 모델을 자동으로 캡처하는 기능으로 한층 더 강화되었습니다. 이를 통해 마이크로서비스 성능을 여러 관점에서 살펴볼 수 있습니다. 이 접근 방식은 성능이 자동 집계되는 서비스 보기로 엔지니어가 복잡한 토폴로지를 빠르고 쉽게 정리할 수 있기 때문에 마이크로서비스 아키텍처에 잘 어울립니다.

많은 경우에 컨테이너 모니터링은 애플리케이션 중심의 성능 지표로 한층 강화되어야 합니다. CA APM은 컨테이너에서 애플리케이션 인스트루멘테이션을 통해 이를 지원합니다. 그러면 마이크로서비스 아키텍처를 지원하는 상황에서 고급 애플리케이션 성능 서비스에 액세스할 수 있습니다. 예를 들어, 정적 기술은 성능 기준값을 관리하고 경보 노이즈를 줄이는 동시에, 트랜잭션 추적 및 선별 지원은 자세한 증거를 수집하고 해결 워크플로를 구축할 수 있습니다.

에이전트를 실행하지 않는 컨테이너 중심의 모니터링 및 심층적인 애플리케이션 인스트루멘테이션은 자체적으로도 훌륭한 서비스지만, CA APM은 이들이 제공하는 정보를 결합하여 보다 뛰어난 통찰력을 제공합니다. CA APM은 애플리케이션 성능을 컨테이너 상태에 자동으로 연결하여 데브옵스 팀에 정확한 문제의 근본 원인에 관한 지표와 상세한 정보를 제공하므로, 컨테이너 애플리케이션 구성이 최고의 성능을 발휘합니다.

섹션 3:

다음 단계

오늘날의 데브옵스 및 애자일 선호 엔터프라이즈는 빠른 변경과 신속한 배포를 위해 노력하고 있습니다. 이러한 기업에 있어 마이크로서비스 아키텍처는 유용하겠지만 완벽한 해결책은 아닙니다. 조직은 더 독립적이며 민첩한 소규모 개발 팀을 만들 수 있지만, 그에 따라 IT는 변화하는 수요에 더 적극적으로 대처해야 합니다.

IT는 개발자들이 생성하는 마이크로서비스를 API 전략에 맞게 조정해야 합니다. 이러한 마이크로서비스의 보안은 무엇보다 중요합니다. 이러한 상황에서 API 게이트웨이 활용은 IT에 큰 이점이 될 것입니다. 또한 속도와 확장을 모색 중이라면 안전도 그에 못지 않게 중요하며, 강력한 관리 구성 요소는 필수라는 점을 간과해서는 안 됩니다.

마이크로서비스에 대해 자세히 알아보려면 다음 e-Book을 다운로드하십시오.

[Microservice Architecture: Aligning Principles, Practices, and Culture](#)

마이크로서비스 및 CA API Management에 대해 자세히 알아보려면 ca.com/microservices를 방문하십시오.



ca.com/kr을 통해 CA 테크놀로지스를 만나 보세요.



CA 테크놀로지스(NASDAQ: CA)는 회사가 변화를 통해 애플리케이션 경제의 기회를 잡을 수 있도록 하는 소프트웨어를 만듭니다. 소프트웨어는 모든 업종, 모든 기업의 핵심입니다. 계획부터 개발, 관리 및 보안에 이르기까지 CA는 전 세계 기업들과 함께 모바일, 프라이빗 및 퍼블릭 클라우드, 분산 및 메인프레임 환경에서 생활, 거래, 소통의 방식을 바꾸고 있습니다. 자세한 내용은 ca.com/kr을 참조하십시오.