애플리케이션 기반 컨테이너 탄력성 자동화

애플리케이션 성능을 보장하면서 시장 출시 속도를 향상하고자 하는 플랫폼 및 DevOps 엔지니어를 위한 자료

목차

경영진 요약

귀사의 경쟁 우위는 아이디어를 비즈니스 트랜잭션으로 재빨리 전환하고 이러한 트랜잭션이 고객을 위해 잘 작동하도록 하는 능력에 달려 있습니다. 기술은 촉진 요인입니다.

컨테이너는 속도, 민첩성, 탄력성, 규모를 제공하여 이러한 애플리케이션이 구축, 배포, 실행되는 방식을 변화시키고 있습니다. 컨테이너는 애플리케이션이 진정으로 어디서나 실행될 수 있는 세상을 안내합니다. 언제 어디서나 하루에 몇 번씩 업데이트와 새로운 기능을 프로덕션 단계로 배포할 수 있고 탄력적인 인프라 공급을 통해 동적으로 변동되는 워크로드 수요를 관리할 수 있습니다. Kubernetes는 조직의 민첩성과 탄력성을 향상할 수 있는 플랫폼이지만, 효율성과 성능을 동시에 보장하기 어렵습니다.

컨테이너화가 제공하는 간편성과 민첩성에도 불구하고, 오케스트레이션 플랫폼은 설명된 방식으로만 서비스를 배포하고 관리하므로 이러한 서비스의 라이프사이클 관리 방법만 제공할 뿐입니다.

컨테이너 플랫폼은 서비스가 SLO를 충족하도록 기본적으로 보장하지 않으며, 동적으로 리소스를 관리할 수 없습니다.
임계값 기반 정책은 지속적 성능을 해결하지 못합니다. 이 접근법은 효과가 있었던 적이 없습니다. 그리고 컨테이너 플랫폼의 변화 속도로 인해 상관관계가 없이 트리거된 자동 스케일링 기능은 결국 문제를 유발할 수 있습니다. 성능을 제공하려면 탄력적인 인프라가 반드시 필요합니다. 그러나 탄력적 인프라에는 원하는 서비스 수준 목표(service-level objective, SLO)를 충족하기 위해 지속적으로 수요, 공급, 제약 사항을 관리하는 자동화된 분석이 필요합니다.

이 백서는 비즈니스 운영을 위한 방법으로 컨테이너 플랫폼을 채택하려 할 때 고려해야 할 주요 개념과 이러한 투자를 자동화를 통해 보호하는 방법을 설명합니다. 이러한 자동화를 통해 비용을 최소화하고 규정을 준수하면서 성능을 보장할 수 있습니다. 또한 서비스 실행을 위해 스스로를 관리하는 Kubernetes 플랫폼을 구현하려면 하향식 분석이 필요한 이유도 제시합니다. 여정 초기에 멀티클라우드 규모를 구축하면 IT 조직은 더 큰 혁신을 이루는 방식과 시기를 근본적으로 변화시킬 운영상의 근육 기억, 즉 무의식적 기억을 얻을 수 있습니다.

속도, 민첩성, 탄력성 및 규모에 대한 약속

Kubernetes는 탄력성을 제공합니다. 그러나 자동으로 애플리케이션 SLO를 충족하거나 보장하지는 않습니다.

컨테이너화 도입 후 성공을 거두려면 개발자에게 필요한 민첩성을 제공하고 계속 변동되는 수요에 맞게 규모를 조정하는 데 필요한 탄력성을 갖추며 필요한 속도로 애플리케이션이 작동하도록 보장할 수 있어야 합니다.

클라우드 네이티브 접근법을 채택하고 애플리케이션을 개별적 서비스 세트로 분해하면 더욱 민첩한 애플리케이션 개발과 배포를 촉진할 수 있습니다. 컨테이너는 서비스를 이동하고 확장할 수 있게 만드는 패키징을 제공합니다. Kubernetes는 디지털 애플리케이션과 서비스 실행을 위한 프레임워크와 제어 지점을 제공합니다. 그러나 비즈니스에 우수한 성능의 엔터프라이즈급 플랫폼을 제공하려면, 이 플랫폼이 제공하는 탄력성을 실현하여 애플리케이션 SLO를 충족하고 보장할 수 있는 기능을 추가해야 합니다.

CICD와 프로덕션 피드백으로 더 빨리 배포
시장 출시 시간을 단축하려면 자동화 기반의 적절한 CICD(continuous integration continuous deployment, 지속적 통합 및 배포) 방법론이 반드시 필요합니다. Google Cloud State of DevOps 2021 보고서[1]에서 응답자들은 CICD를 구현하여 상당한 개선을 이루었다고 응답했습니다.

배포 빈도	주간–월간	시간별–일별
리드 타임 변경	6개월 초과	1시간 미만
장애율 변경	16%–30%	0%–15%

속도가 향상되면 프로덕션 단계에서 끊임없는 변화를 관리하는 방법이 필요하게 됩니다. 또한 서비스 작동 상태와 인프라에 필요한 사항을 예측하는 방법과 관련된 피드백 루프도 필요하게 됩니다. 목표는 SLO를 정의하고, 성능 문제의 리스크를 줄이기 위해 플랫폼이 컨테이너 및 인프라 구성 방법에 대한 피드백을 제공하도록 하는 방법을 확보하는 것입니다.

- 서비스에 리소스를 할당하는 방식을 누가 결정하나요? 어떻게 결정하나요? 설정된 SLO를 기준으로 부하 테스트 및 벤치마킹 등을 수행합니다.
- 성능을 어떻게 측정하나요? 컨테이너와 포드가 올바르게 구성되었는지 확인하기 위한 피드백 루프가 CICD 파이프라인에 존재하나요?
- 새로운 배포를 위한 용량이 항상 충분한지 어떻게 확인하나요?

IBM Turbonomic의 해답 알아보기

옵션	한계점	IBM Turbonomic의 해답
컨테이너 및 포드 활용도 데이터를 수동으로 분석하여 리소스 사양을 결정합니다.	– 데이터 수집 설정 – 분석을 위한 노동력	– 하향식, 애플리케이션 기반 분석을 통해 컨테이너 크기 조정 방법 결정 – CICD에 대한 피드백 – 필요하지 않을 때 요청을 줄일 수 있는 기회
스택의 모든 지점에서 리소스 데이터를 수동으로 분석하여 프로덕션 용량을 결정합니다.	– 여러 소스의 데이터를 수집하기 위한 노동력 – 분석을 위한 노동력	활용도 기반 분석을 통해 전체 스택에 걸쳐 리소스 요구 사항 파악

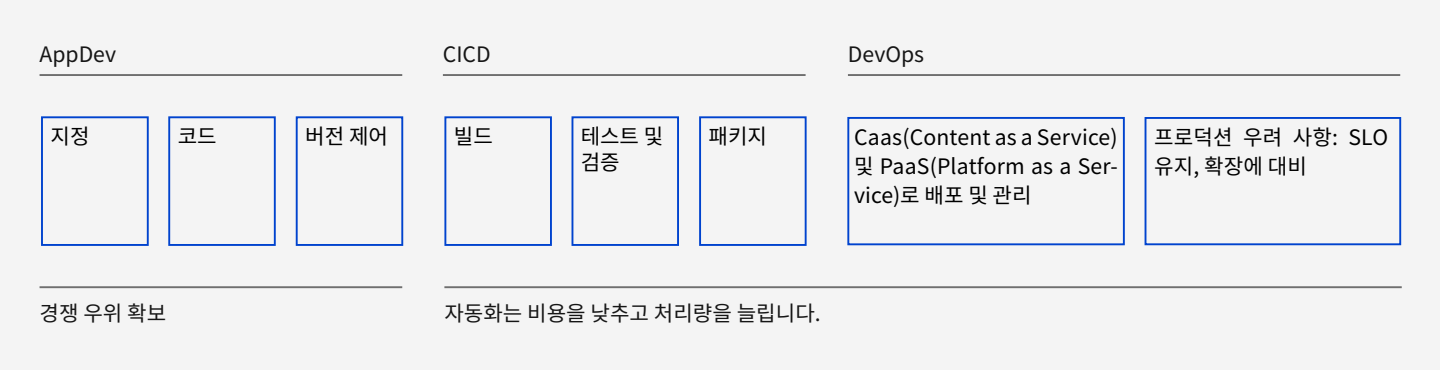


그림 1. 애플리케이션 민첩성을 위한 프로세스.

플랫폼 및 인프라

앱 기반, 전체 스택 관리가 필요한 이유
어떤 컨테이너 플랫폼 또는 기반 인프라(프라이빗 클라우드, 퍼블릭 클라우드, 하이브리드 클라우드, 멀티클라우드 또는 심지어 베어메탈까지)를 선택하든, PaaS(Platform as a Service)가 제시하는 운영상의 문제는 같습니다.

- 현재의 수요와 확장된 수요를 충족하기에 충분한 용량이 있는지 어떻게 결정하는가?
- 더 많은 애플리케이션 노드를 시작해야 할 때는 어떻게 결정하는가?
- 일시 중지가 필요한 때를 어떻게 결정하는가?
- 수요 급증에 어떻게 대처해야 하는가?
- 버스팅을 위해 퍼블릭 클라우드 리소스를 어떻게 활용해야 하는가?
- 스택 전반에서 고가용성(high availability, HA)과 복원력을 어떻게 보장하는가?
- 비즈니스 제약 사항을 어떻게 실행하는가?

컨테이너 플랫폼이 제공하는 탄력성을 활용하면 애플리케이션의 최고 수요의 합이 아니라 애플리케이션의 평균 수요의 합에 맞춰 프로비저닝할 수 있습니다. 이러한 능력을 활용하고 변동하는 수요에 맞게 지속적으로 확장되고 축소되는 플랫폼을 제공하려면 애플리케이션에 필요한 컴퓨팅, 스토리지, 네트워크가 필요한 시기에 확보되도록 리소스 활용 결정을 지속적으로 내리는 소프트웨어가 필요합니다.

IBM Turbonomic의 해답 알아보기

옵션	한계점	IBM Turbonomic의 해답
연계 서비스 그룹(ASG), 가용성 세트 등 자동 스케일링 그룹을 제공하는 서비스 제공자 이용	– 임계값 기반 정책 – 특정 노드를 확장할 수 없음: 모든 노드가 같은 제약 사항, 같은 레이블 등이어야 함	– 하향식 애플리케이션 기반 SLO – 애플리케이션 수요를 충족하기 위해 지속적으로 인프라 리소스 조정 – 적절한 컨테이너, 포드, 노드를 수직 또는 수평으로 지속적으로 확장 및 축소 – 지속적으로 포드를 적절한 노드에 배치
스택의 모든 지점에서 리소스 데이터를 분석하여 프로덕션 용량 결정 .	– 여러 소스의 데이터를 수집하기 위한 노동력 – 분석을 위한 노동력	– 활용도 기반 분석을 통해 전체 스택에 걸쳐 리소스 요구 사항 파악 – 적절한 컨테이너, 포드, 노드를 수직 또는 수평으로 지속적으로 확장 및 축소 – 지속적으로 조치를 트리거하여 병목현상 방지

규모에 맞게 SLO 충족을 위해 운영
컨테이너 플랫폼의 목적은 비즈니스를 위해 원하는 서비스 수준으로 애플리케이션을 실행하는 것입니다. 애플리케이션 수가 늘어나도 지속적으로 성능을 보장해야 합니다. 일반적으로 우리는 고객이 첫 애플리케이션 1~3개에 대해 12개월이 넘는 시간을 소요하는 것을 확인했습니다. 그 다음 추가되는 애플리케이션의 경우 학습된 기술과 베스트 프랙티스를 활용하므로 추가로 6~12개월이 걸릴 수 있습니다. LOB(line of business)가 무엇이 가능한지 알게 되면, 관리해야 할 개별 서비스의 수와 규모는 인간의 관리 능력 범위를 벗어납니다. 스테이트리스(stateless) 서비스를 구축한 경우에도 컨테이너의 임시적 속성을 활용하면 최종 고객 경험에 영향을 주는 성능의 저하를 얼마나 허용할 수 있을까요? 수요뿐만 아니라 빨라지는 변화의 속도까지 관리하기 위해 무엇을 할 수 있을까요? 해답은 자동화에 있습니다. 이러한 자동화는 SLO 보장에 필요한 서비스 인스턴스의 수, 워크로드 크기 및 배치의 구성, 인프라에서 규정을 준수하는 리소스를 제공하는 것과 관련된 장단점 분석을 기반으로 한 조치들을 통해 구현됩니다.

임계값은 문제를 해결해 주지 않습니다.
컨테이너 플랫폼은 사용 가능한 서비스를 최소로 갖추도록 보장할 것입니다. 서비스 하나에 장애가 발생하면 이를 다시 시작하려고 할 것입니다. 그러나 우수한 고객 경험을 보장하려고 한다면, 성능 저하와 장애가 발생하기 전에 시스템이 대응할 수 있어야 합니다. 수요를 충족하기 위해 수평적 자동 스케일링을 기본적으로 설정할 수 있지만, 어떤 메트릭이 필요한 리소스를 가장 잘 알려주는지 결정하고 임계값 그리고 상한값과 하한값을 구성하여, 이것이 프로덕션의 수요에 맞게 기능할지 테스트하고 추정한 다음, 배포된 모든 서비스에 대해 이를 반복해야 합니다. 하나의 애플리케이션에 100개가 넘는 서비스가 있다고 생각해 보세요. 이러한 각 정책은 서로 아무런 상관관계가 없습니다.

서비스의 포드를 더 추가하면 다른 영역에 정체가 발생하지 않는다고 어떻게 보장할 수 있을까요? 잘못 구성된 포드를 복제하고 있나요? 먼저 수직 스케일링이 필요한가요? 노드 정체를 어떻게 관리하고 인접 노이즈 현상에 어떻게 대처하며 이러한 수요를 충족하는 데 사용될 수 있는 사용되지 않는 할당 리소스를 어떻게 찾을까요?

게다가, 컨테이너, 포드, 수평적 포드 오토스케일러(HPA) 또는 클러스터 자동 스케일링 정책을 구성하는 일은 일회성 작업이 아닙니다. 최선의 추측은 지속적으로 모니터링하고 재정의해야 합니다. 여러분의 팀이 이러한 임계값을 수동으로 설정 및 재설정하지 않아도 된다면 절약된 시간으로 무엇을 할 수 있을까요?

이러한 구성을 올바르게 설정하는 것은 중요하며 디지털 혁신 전략의 성공적 실행에 직접적 영향을 미칩니다. 몇 번의 잘못된 배포만으로도 구축 중인 플랫폼과 시스템의 채택 속도를 상당히 느리게 할 수 있습니다. 그리고 이러한 제어 지점을 구성하는 데 너무 많은 시간과 수동 노동력이 사용되면 조직이 플랫폼 우선 조직이 되는 데 상당한 지장을 줄 수 있습니다. 귀사는 이러한 지연을 허용할 여유가 있나요? 분석 엔진을 사용하여 모든 리소스 간의 균형을 관리할 수 있고, 컨테이너의 수직적 스케일링 한계와 요청, 필요한 포드의 수, 그리고 포드를 재배포하고 클러스터 리소스를 관리하기 위한 배치 결정을 정의할 수 있는 제어 시스템이 필요합니다.

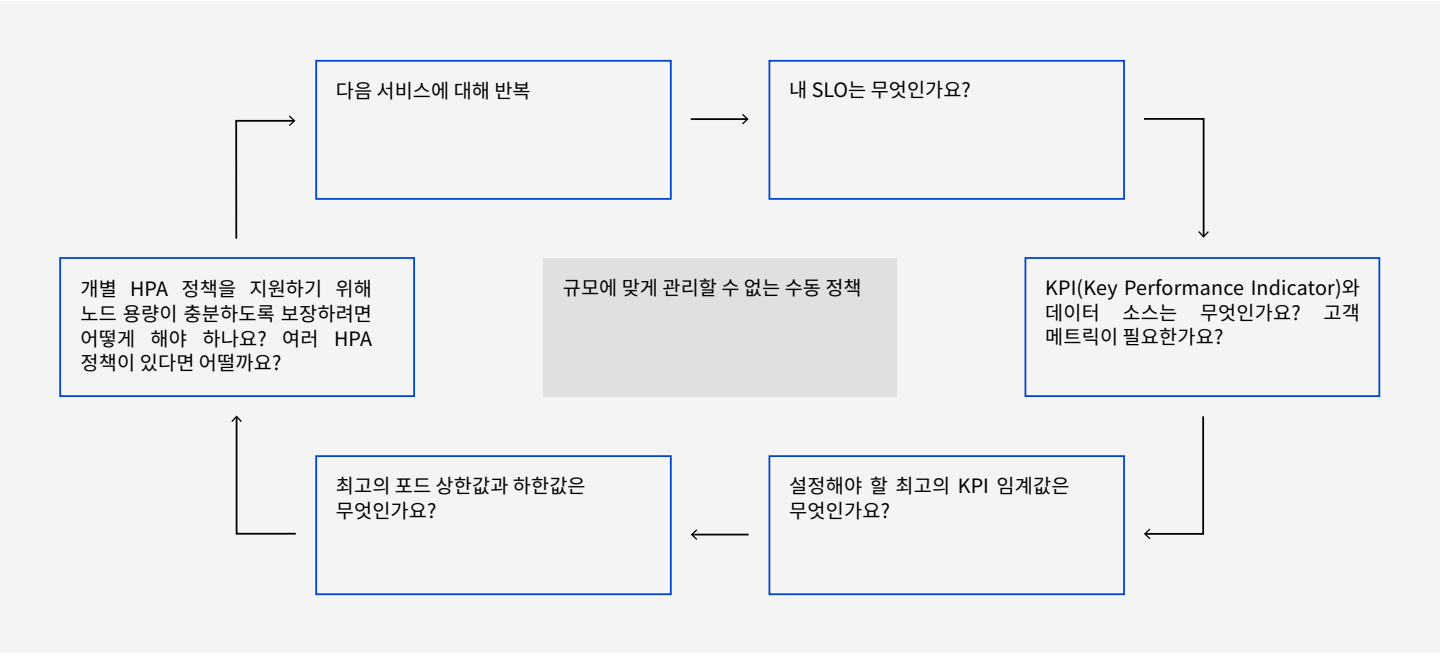


그림 2. 규모에 맞게 관리할 수 없는 수동 정책

IBM Turbonomic의 해답 알아보기

옵션	한계점	IBM Turbonomic의 해답
포드의 확장 및 축소를 트리거해야 하는 경우의 HPA 임계값 기반 정책	– 서비스당 구성 – 서비스를 위한 모든 포드의 평균 기반 – KPI 및 임계값, 포드 상한값 및 하한값을 수동으로 정의	– 하향식 애플리케이션 기반 SLO – SLO 충족을 위해 응답 시간 데이터를 사용하여 서비스의 수평적 확장을 촉진 – 적절한 컨테이너, 포드, 노드를 수직 또는 수평으로 지속적으로 확장 및 축소 – 지속적으로 포드를 적절한 노드에 배치 – 애플리케이션 수요를 충족하기 위해 지속적으로 인프라 리소스 조정
수직적으로 컨테이너를 확장하기 위한 수직적 포드 오토스케일러(VPA) 임계값 기반 정책	– 각 서비스에 대해 정의해야 함 – 베타 프로젝트: 자체적인 리스크를 감수하고 사용 – 조치를 취하기 위해 노드 용량에 액세스하지 않음	
더 나은 노드에 재배포되도록 포드의 장애를 허용	장애를 일으킬 준비가 된 포드에서의 트랜잭션에 대한 열악한 사용자 경험	
Prometheus 관측성 솔루션이 데이터 수집 및 통합	– 데이터 분석을 제공하지 않음 – 조치를 제공하지 않음	

앱 기반 접근법

애플리케이션 SLO가 인프라를 결정해야 함
미션 크리티컬 애플리케이션의 컨테이너화는 수많은 이점을 가져다주는 투자입니다. 그러나 속도, 탄력성, 이동성이라는 이점을 완전히 누리려면 연중무휴 24시간 적시에 적절한 리소스 결정을 내릴 수 있는 소프트웨어가 필요합니다. 이러한 소프트웨어가 없으면 복잡성으로 인해 속도가 저하될 것입니다.

IBM® Turbonomic® Application Resource Management는 애플리케이션의 실행 위치가 어디든 미션 크리티컬 애플리케이션을 Kubernetes 플랫폼 및 기반 인프라와 연결합니다. 이 소프트웨어는 실시간 애플리케이션 수요에 따라 논리적 레이어부터 물리적 레이어까지 스택의 모든 레이어에서 제약 사항과 상호종속성을 고려하여 적시에 적절한 조치를 결정함으로써 항상 애플리케이션이 작동하는 데 필요한 사항이 제공되도록 지원합니다. 실시간으로, 예약을 통해 또는 DevOps 파이프라인의 일부로 실행할 수 있습니다.

지능형 사이즈 조정: 컨테이너의 사이즈를 어떻게 조정해야 할까요?
- 배포를 통해 자동화합니다. 파이프라인의 일부로 실행하고 지속적으로 사이즈를 재조정합니다(예: YAML, Jenkins 등).
- 실시간으로 자동화합니다. Kubernete를 통해 동적으로 실행합니다.

연속적 배치: 언제 포드를 이동해야 할까요? 어느 노드로 이동해야 할까요?
- Kubernetes를 통해 실시간으로 그리고 동적으로 실행합니다. 운영 중단 없는 스테이트리스 서비스에 대해서만 실행합니다.

동적 스케일링: 언제 클러스터를 스케일아웃 또는 스케일백해야 할까요? 얼마나 해야 할까요?
- 코드형 인프라 또는 Kubernetes Cluster API를 통해 실시간으로 그리고 동적으로 클러스터 스케일링을 실행합니다.

SLO 기반 스케일링: 애플리케이션 응답 시간 SLO를 충족하려면 언제 포드를 스케일아웃 또는 스케일백해야 할까요? 얼마나 해야 할까요?

SLO 기반 스케일링의 전제 조건:
- 애플리케이션이 수평적 스테이트리스 마이크로서비스를 위해 설계되었습니다.
- Kubernetes가 제공하지 않는 SLO 데이터의 정의와 소스가 있습니다.

이러한 지능적 자동화가 귀하, 귀하의 팀 그리고 귀하의 비즈니스에 의미하는 바는 무엇일까요? Kubernetes를 온프레미스, 클라우드, 베어메탈 서버 또는 어떤 조합을 기반으로 실행하든 IBM Turbonomic이 제공하는 공유한 이점은 다음과 같습니다.

앱을 위한 크루즈 컨트롤: 응답 시간 SLO를 설정하면 AI 기반 소프트웨어가 앱이 어디에서 실행되든 플랫폼과 기반 인프라에서 SLO 충족에 필요한 리소스가 제공되도록 지원합니다.

수동 노동 최소화: 개발자, DevOps 및 사이트 신뢰성 엔지니어(SRE)가 임계값, 제약 조건 또는 자동 스케일링 정책을 설정할 필요가 없습니다. 소프트웨어가 실제로 자동화할 수 있는 조치를 제공하여 사용자에게 적합한 리소스 결정을 내립니다.

용량에 과다 지출하지 않음: 리소스 관련 결정을 내리기 위해 개발자에게 의존하지 않아도 됩니다. 개발자들은 종종 안전성을 위해 과다프로비저닝하곤 합니다. IBM 소프트웨어는 애플리케이션 수요를 바탕으로 필요한 리소스 서비스를 정확하게 결정합니다.

자신 있게 DevOps 가속화: 안전하게 배포 빈도와 규모를 늘립니다. IBM의 분석 기능은 DevOps 워크플로우와 통합되어 새로 배포된 서비스 및 기존 서비스가 항상 작동하도록 지원합니다.

더 쉽게 확장에 대비: IBM의 소프트웨어로 새로운 서비스의 온보딩을 시뮬레이션할 수 있습니다. 새로운 확장을 지원하는 데 정확히 몇 개의 노드가 더 필요한지 결정할 수 있습니다.

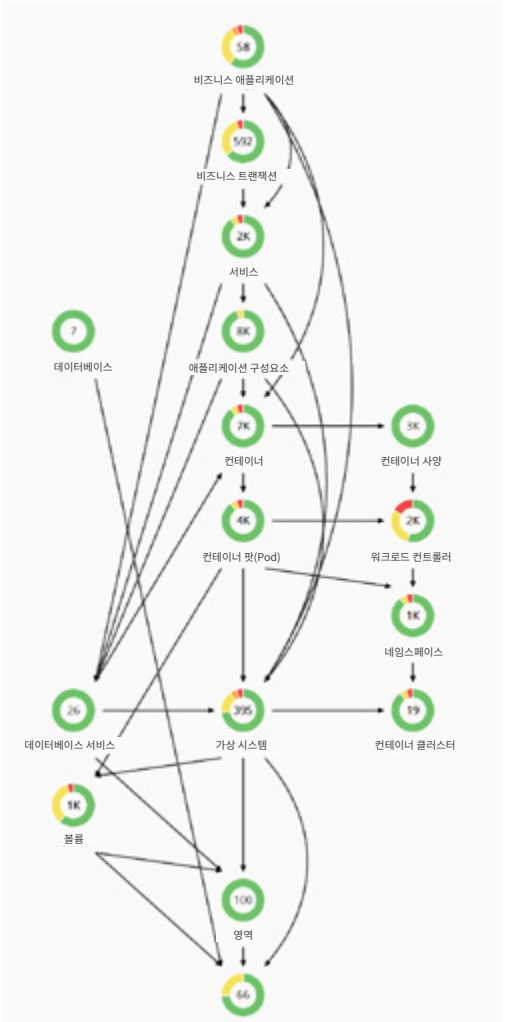
고객 사례 소개

팬데믹 동안 디지털 혁신 가속화

Kubernetes 플랫폼과 기반 인프라 내에서 IBM Turbonomic의 동적 리소스 활용 기능은 응답 시간을 낮게 유지했습니다.

이 고객은 600만 명 이상의 고객을 보유한 남미 최대 규모의 보험회사 중 하나입니다. 이 회사는 기존 환경과 차세대 환경의 리소스 활용 방식을 관리할 때 업계 표준 접근법을 취했기 때문에 디지털 혁신과 팬데믹에 대한 대응이 느려지고 있었습니다.

IBM Turbonomic 자동화 기능이 휴가철 수요 급증 동안 응답 시간을 낮게 유지했습니다.
이 고객에게는 해당 지역에서 운항하는 최대 규모의 저비용 항공사 중 한 곳과 통합되는 비즈니스 앱이 있습니다. 여행자 보험은 이 앱을 통해 예약되었습니다. 그림 3에서 볼 수 있는 고점은 여러 날 계속된 부활절 휴가와 관련이 있습니다. 앱에서 수요가 증가해도 Kubernetes 플랫폼과 기반 인프라 내에서 IBM Turbonomic의 동적 리소스 활용 기능으로 응답 시간을 낮게 유지할 수 있었습니다.



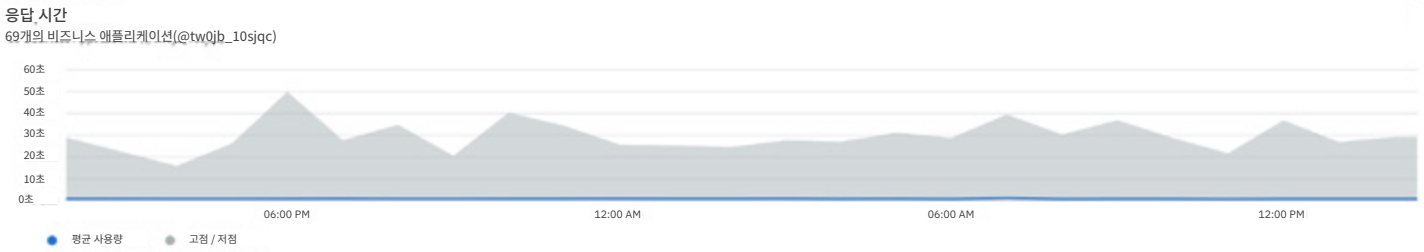


그림 3. 개별 비즈니스 애플리케이션과 응답 시간에 대한 전체 스택 뷰, 수요가 고점에 도달한 동안에도 자동화를 통해 응답 시간을 낮게 유지함

57개의 미션 크리티컬 애플리케이션
- 예: 자동차의 GPS: 차량 도난 신고, 새로운 정책에 대한 견적 등
- ~3,000개의 포드 (~7,000개의 컨테이너로 구성됨)
- Dynatrace에 연결됨

자동화
- 컨테이너 사이즈 재조정(스테이징)
- 연속 배치(모두)

티켓 ~70% 감소

IBM Company인 Turbonomic 소개

IBM® Turbonomic® Application Resource Management는 하이브리드 및 멀티클라우드 환경에서 애플리케이션에 동적으로 리소스를 제공하여 애플리케이션 성능과 거버넌스를 보장하기 위해 고객이 사용하는 애플리케이션 리소스 관리(ARM) 소프트웨어를 제공합니다. IBM Turbonomic 네트워크 성능 관리(NPM)는 기업, 통신사 및 매니지드 서비스 제공자로 구성된 멀티벤더 네트워크에서 규모에 맞게 지속적 네트워크 성능을 보장하는 데 도움을 주는 현대적인 모니터링 및 분석 솔루션입니다.

IBM Turbonomic 지능형 자동화에 대한 자세한 정보는 ibm.com/cloud/turbonomic 또는 IBM 담당자를 통해 확인할 수 있습니다.

(07326) 서울특별시 영등포구 국제금융로 10
서울국제금융센터(3IFC)
TEL: (02)3781-5114

Produced in the United States of America
2022년 3월

[1] State of DevOps 2021, Google Cloud, 2021