



후원 : Amazon Web Services

# 기업의 애플리케이션 현대화:

아키텍처, 운영, 배포에  
미치는 영향

Larry Carvalho    2020년 4월



# IDC 견해

클라우드를 도입하는 첫 단계에서는 대개 리프트 앤 시프트 방식을 통해 워크로드를 온프레미스 데이터 센터에서 퍼블릭 클라우드로 마이그레이션합니다. 하지만 기업들이 새로운 클라우드 기술을 완전하게 구현한다면 클라우드 마이그레이션을 통해 더 많은 이점을 얻을 수 있습니다.



적절한 계획을 세워 워크로드를 퍼블릭 클라우드로 마이그레이션하면 프로덕션 환경과 유사한 설정으로 환경을 빠르게 복제할 수 있습니다. 또한 새 기능을 신속하게 테스트할 수 있으므로 개발자가 획기적인 기술을 더욱 빠르게 개발할 수 있습니다. 새로운 배포 방법으로 애플리케이션을 도입하는 경우 매우 다양한 이점이 제공되므로, 지속적인 애플리케이션 개선 주기를 진행하고 있는 기업도 많습니다. 이렇게 마이그레이션한 워크로드가 FaaS(서비스로서의 함수), 컨테이너, 마이크로서비스 등의 새로운 클라우드 기능을 활용하는 경우에는 혁신을 더욱 빠르게 추진할 수 있습니다. 이러한 관리형 서비스를 도입하면 애플리케이션 인프라를 패치 및 업데이트하고 확장할 필요가 없으므로 조직의 운영 과정이 크게 개선됩니다.

지난 수십 년간 모놀리식 애플리케이션은 기존의 인프라 및 소프트웨어 아키텍처를 사용하여 구축되어 왔습니다. 따라서 솔루션 포트폴리오에 이러한 애플리케이션이 포함되어 있는 조직은 업무상 필요한 디지털화 추세에 제때 대응할 수 없습니다. 레거시 애플리케이션은 유연성도 낮으므로 팀의

대응 능력도 떨어집니다. 또한 기존 애플리케이션 유지 관리 비용이 IT(정보 기술) 관련 지출에서 높은 비율을 차지하므로, 조직이 기존 솔루션 포트폴리오를 개선하는 데 사용할 수 있는 예산이 제한됩니다. 오늘날에는 클라우드 기술이 급속도로 발전하고 있으므로 엔터프라이즈 애플리케이션 아키텍처도 현대적 기능을 사용할 수 있도록 조정해야 합니다. 그러면 개발자가 새로운 방식과 도구를 활용하여 새 기능을 더욱 빠르게 배포할 수 있으며, 운영 팀은 업무 과정에서 더욱 전략적인 역할을 수행할 수 있습니다.

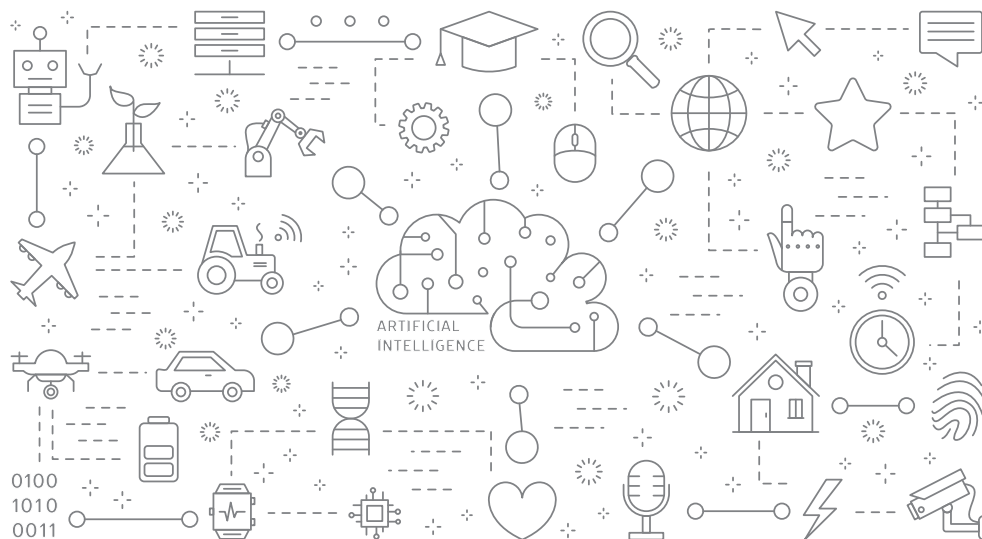
레거시 애플리케이션 포트폴리오를 보유한 여러 대기업이 현재 애플리케이션 현대화 단계를 진행하고 있습니다. 그 이유는 현대적 애플리케이션을 개발함으로써 고객을 위한 새로운 디지털 경험을 생성해 시장 경쟁력을 높이기 위해서입니다. 클라우드 네이티브 아키텍처를 도입하는 조직은 대응 능력을 개선하고 새 애플리케이션 출시 시간을 단축할 수 있습니다. 또한 일상 생활에서 소비자용 애플리케이션을 자주 사용하는 고객들이 기대하는 사용자 경험을 제공할 수 있습니다. 클라우드 공급자가 애플리케이션에 필요한

인프라 관리 작업을 모두 처리하는 서버리스 운영 모델을 도입하면 소프트웨어 릴리스 빈도를 높일 수 있습니다. 개발자와 운영 팀이 새 기능 설계 및 릴리스 작업을 위해 작업 시간을 다시 할당할 필요가 없기 때문입니다. 온프레미스에서 애플리케이션을 실행하는 대다수 기업이 애플리케이션을 현대화하는 첫 단계에서는 일종의 컨테이너가 사용됩니다. 컨테이너는 이식성과 유연성이 매우 우수하지만, 인프라 관리 작업은 여전히 수행해야 합니다. 대부분의 기업이 현대화를 진행하는 이유는 FaaS(서비스로서의 함수)를 도입하기 위해서입니다. FaaS를 도입하면 인프라 관리 작업을 전혀 수행할 필요가 없기 때문입니다. 이 백서에서는 온라인 여행사인 Expedia가 애플리케이션 현대화 과정에서 얻은 교훈을 소개합니다.

리팩터링(Refactoring) 과정에서는 애플리케이션 서버와 데이터 저장소 하나로 구성된 기존 아키텍처를 여러 런타임, 분산 서비스 및 소규모 데이터 저장소로 구축된 현대적 아키텍처로 전환합니다. 현대적 릴리스 파이프라인에는 소스

코드 관리부터 배포까지의 과정을 빠르게 진행할 수 있는 자동화된 개발 수명 주기가 포함됩니다. IDC의 설문조사 결과에 따르면 컨테이너, 컨테이너 오케스트레이션, 서버리스, 마이크로서비스 등의 클라우드 네이티브 기술에 대해 잘 알고 있는 개발자는 전체의 1/5도 되지 않는다고 합니다. 따라서 IT 관리 작업에서는 개발자 대상 교육을 우선적으로 진행하면서 애플리케이션 현대화를 위한 IT 예산을 확보해야 합니다.

최신 애플리케이션을 개발하는 과정에서는 자율 개발 팀이 애플리케이션을 마이크로서비스로 분해하여 컨테이너 또는 서버리스 클라우드 기술에서 실행할 수 있도록 최적화하는 작업을 수행합니다. 이러한 방식을 사용하는 경우 기존 방식처럼 코드 관리부터 배포까지의 과정을 진행할 필요가 없습니다. 또한 혁신을 대규모로 추진할 수 있으며, 보안 위반과 가동 중지 위험도 방지할 수 있습니다. 경쟁 우위를 확보하고 최근의 혁신 기술 동향을 파악하려는 기업은 애플리케이션 현대화를 최우선으로 진행해야 하며 해당 작업에 최대한 집중해야 합니다.





## 이 백서의 내용

이 백서에서는 애플리케이션 현대화를 우선적으로 진행해야 하는 이유와 효율적인 리팩터링 작업을 수행하기 위해 진행해야 하는 권장 단계를 설명합니다. IDC는 Expedia Group과의 면담을 통해 AWS 서비스를 사용하여 애플리케이션 현대화를 진행한 경험, 그리고 현대화 과정에서 얻은 교훈을 파악했습니다. 이 문서에서는 해당 정보를 소개합니다.





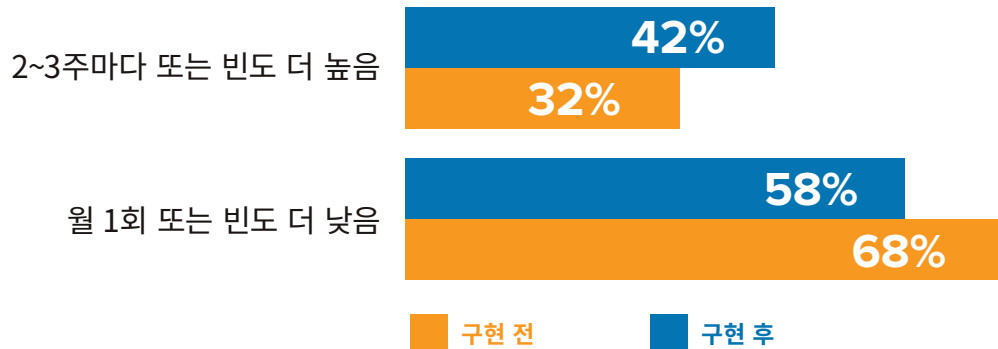
# 상황 개요

## 기업이 애플리케이션을 현대화해야 하는 이유

급속도로 변화하고 있는 시장에서 경쟁력을 유지해야 하는 조직은 현대적 애플리케이션 아키텍처를 구축해야 합니다. 애플리케이션의 새 기능과 관련한 LOB(Line of Business) 요구 사항은 계속해서 변화하고 있습니다. 따라서 개발자는 기존 애플리케이션 개발 도구와 기본 인프라를 사용해 변경 요청을 처리해야 합니다. 새 애플리케이션을 빠르게 개발하여 배포하기 위해 수행해야 하는 단계 중 하나는 DevOps 도입입니다. DevOps는 애플리케이션 작동 과정이 설계 단계부터 시작되는 개발 방법론입니다. DevOps 사용 시에는 개발자와 운영자가 동시에 작업을 진행하여 애플리케이션을 구축 및 릴리스하고 운영합니다. 조직에서는 고객에게 유용한 애플리케이션을 안정적이며 빠르게 배포하기 위해 이 방식을 사용합니다. DevOps 방법론을 사용하면 애플리케이션 배포 빈도를 높일 수 있습니다. 개발자 대상 설문조사인 IDC의 PaaSView and the Developer Survey에서도 이러한 이점을 확인할 수 있습니다. 실제로 DevOps 구현 후 매주 코드 릴리스를 배포할 수 있게 된 조직의 수가 2배로 늘어났습니다(그림 1 참조).

## 그림 1. 애플리케이션 현대화 후 높아진 배포 빈도

**질문:** DevOps 구현 후 조직의 코드 릴리스 배포 빈도는 평균적으로 어느 정도입니까?



출처: IDC의 PaaSView and the Developer Survey, 2019년

현대적 애플리케이션에서는 구성 요소가 서로 독립적이므로 개발자는 중대한 작동 중단 현상 발생에 신경 쓸 필요 없이 새 기능을 안전하게 적용해 볼 수 있습니다. 또한 운영 팀의 역할도 이전과는 전혀 달라집니다. 즉, 소프트웨어 업데이트 설치 등의 수동 작업 대신 작업 지침 결정, 직원 채용 및 현대적 기술 교육, 애플리케이션 전체 상태 확인 관리 같은 더욱 전략적인 작업에 더 많은 시간을 할애할 수 있게 됩니다. 소수의 사용자를 대상으로 기능을 배포하여 정상 작동 여부를 파악할 수 있으며 필요에 따라 변경 사항을 자동으로 롤백할 수 있습니다. 이 프로세스를 진행하면 사용 가능한 여러 솔루션을 적용해 보는 경우 발생하는 비용을 줄일 수 있으므로 기업 현대화 과정의 성공 가능성이 높아집니다.

현대적 애플리케이션 아키텍처를 구축하는 조직이 다양한 이점을 얻을 수 있는 이유는 다음과 같습니다.



- **새 기능 출시 시간 단축:** 모듈식 애플리케이션의 이점 중 하나는 나머지 구성 요소와는 독립적으로 필요한 구성 요소만 변경하고 테스트할 수 있다는 점입니다. 그러므로 코드 변경부터 배포까지의 과정에 소요되는 시간을 줄일 수 있습니다. 기존 애플리케이션의 경우 기능 릴리스에 몇 달이 걸릴 수도 있는 반면, 리팩터링한 애플리케이션에서는 신규 또는 변경된 기능 요청을 수락하여 코드를 수정하고 배포를 완료하여 사용자에게 새 기능을 제공하는 전체 과정을 몇 분 내에 완료할 수 있습니다.



- **인프라 민첩성 개선:** 배포 시 컨테이너와 서버리스 기술을 사용할 수 있으므로 개발자가 클라우드 서비스를 최적화하여 다양한 애플리케이션 수요에 따라 인프라를 즉시 확장할 수 있습니다. 즉, 기존 방식으로 모놀리식 애플리케이션을 지원하는 전체 인프라를 확장하는 대신 특정 사용자 수요를 충족하는 컨테이너 하위 집합만 몇 마이크로초 내에 확장할 수 있습니다.



- **포괄적 보안 정책:** 오늘날에는 애플리케이션이 빠르게 바뀌고 있으며, 그에 따라 코드 수정 시 발생할 가능성이 있는 보안 문제를 적절하게 처리해야 합니다. 현대적 아키텍처를 사용하여 개발 수명 주기에 체크포인트를 추가하면 현대적 아키텍처의 모든 보안 요소를 관리하고 적용할 수 있습니다.



- **지속적인 애플리케이션 변경 과정 수정:** 컨테이너 이식성과 작동 중단 여부를 모니터링하면 여러 리전에 분산되어 있는 애플리케이션을 쉽게 복구할 수 있습니다. 그러면 이러한 애플리케이션에서 고가용성을 표준 요소로 보장할 수 있습니다. 퍼블릭 클라우드 기술 사용 시에 제공되는 중요한 이점 중 하나는 애플리케이션 개발 수명 주기의 모든 단계를 모니터링할 수 있다는 것입니다. 이 모니터링 과정에서 제공되는 폐쇄 피드백 루프를 통해 새 기능을 릴리스하고 사용하는 과정에서 애플리케이션을 지속적으로 개선할 수 있습니다.



- **공통의 목표 달성:** 레거시 인프라에서는 ‘기술적 부채’가 매우 많이 발생하기 때문에 개발 팀은 여러 프로젝트의 백로그를 처리해야 합니다. 이사회와 주주 역시 고위 관리자가 현대적 기술을 더욱 발빠르게 도입하기를 원합니다. 오늘날에는 CEO가 CIO의 직속 상관인 기업도 있고, 소프트웨어를 개발하지 않는 기업이 소프트웨어 업체 출신 CEO를 영입하기도 합니다. 또한 IT 부서에서는 새로 발생하는 수요를 충족하기 위해 설비를 교체하면서 새 기능도 계속 개발해야 하므로 LOB(Line of Business)에서 IT 부서를 신뢰하지 못하는 경우도 많습니다. 현대적 애플리케이션 개발 도구와 DevOps 방법론에 투자하면 LOB(Line of Business) 관리 팀과 IT 팀의 목표와 비전을 모두 실현할 수 있으므로 모든 관련자에게 이점이 제공됩니다.

## 애플리케이션 현대화 진행 단계

장기적으로 '기술적 부채'가 발생해 온 조직은 현대화 작업의 우선 순위를 결정하고 해당 작업에 적합한 직원과 예산을 할당해야 합니다. IDC의 예측 결과에 따르면 2025년까지 전체 기업 중 거의 2/3는 다양한 소프트웨어를 제작하면서 매일 코드 릴리스를 배포하게 될 것이라고 합니다. 또한 새로 제작되는 앱 중 90% 이상은 클라우드 네이티브 앱이고, 애플리케이션 구성 요소 중 80%는 외부에서 공급되며, 개발자 수는 현재보다 1.6배 늘어날 것으로 보입니다. 현대적 클라우드 아키텍처를 기반으로 구축된 애플리케이션에서 장기적으로 제공되는 이점을 적절하게 활용하면 더 저렴한 비용으로 시장 변화 대응 능력을 높일 수 있으며, 이를 통해 현대화 과정을 더욱 빠르게 진행하기 위한 예산을 확보할 수 있습니다.

현대적 애플리케이션 아키텍처로 이전하려는 조직에게 필요한 진행 단계는 다음과 같습니다.



- **기술 습득 또는 개발:** 현재 대부분의 조직에는 기능, 컨테이너 및 마이크로서비스로 구성된 현대적 엔터프라이즈 아키텍처에 대해 잘 알고 있는 직원이 거의 없습니다. 엔터프라이즈 개발 팀 역시 VM(가상 머신) 중심 기술을 사용하지 않고 새로운 상태 비저장 애플리케이션을 작성하는 방법을 정확하게 파악하고 있지는 못한 상태입니다. 따라서 운영 팀은 문제를 실시간으로 자동 해결하는 복잡한 모니터링 및 로깅 시스템을 설치해야 합니다. 그림 2에 나와 있는 것처럼, 오늘날 새롭게 등장하고 있는 개발 기술을 정확하게 파악하고 있는 직원은 전체의 1/5에 불과합니다. 이 문제를 해결하려면 직원들이 새로운 애플리케이션 배포 방식에 적응할 수 있도록 조직 문화를 바꿔야 합니다. 그러려면 현대적 클라우드 기술이 제공할 수 있는 이점의 모든 측면을 조직 내 인력이 파악해야 합니다.



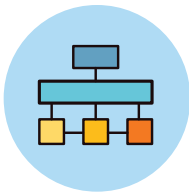
- **포트폴리오 분석:** 엔터프라이즈 애플리케이션 현대화 과정에서는 먼저 기존 포트폴리오를 분석하여 LOB(Line of Business) 요구와 일치하는 솔루션, 즉 조직에 가장 많은 가치를 제공하는 솔루션이 있는지를 확인합니다. 클라우드 네이티브 기술을 활용하기에 더 적합한 구성 요소가 포함된 애플리케이션을 파악하여 마이크로서비스로 분할해야 합니다. 수요가 급증하는 솔루션은 마이크로서비스로 리팩터링하면 효율적입니다. 그러면 확장 및 축소해야 하는 이와 같은 기능을 모놀리식 애플리케이션의 나머지 부분에서 격리할 수 있습니다. 레거시 애플리케이션이 리팩터링된 애플리케이션의 수요를 처리할 수 있도록 하려면 이와 같이 기존 포트폴리오와의 통합을 계획해야 합니다. 리팩터링 작업으로 인한 투자 수익을 조기에 실현할 수 있는 애플리케이션을 우선적으로 리팩터링하면 애플리케이션 현대화를 안정적이면서도 신속하게 진행할 수 있습니다.





- **보안 전략 수립:** 애플리케이션을 빠르게 출시 및 배포하는 것도 중요하지만, 애플리케이션 현대화를 성공적으로 완료하려면 개발 및 배포 수명 주기 전반에서 보안 체크포인트가 포함된 파이프라인을 구축해야 합니다. 규제 및 규정 준수 관련 지침을 따르려면 데이터 주권 및 데이터 프라이버시 관련 요구 사항을 충족해야 합니다. 모든 리팩터링 애플리케이션에 공통의 보안 기능을 적용하면 재사용 가능성을 보장하고 전반적인 작업량을 줄일 수 있습니다.

현대적 애플리케이션 아키텍처에서는 API(애플리케이션 프로그래밍 인터페이스)를 통해 연결된 개별 서비스를 사용합니다. 따라서 데이터 및 애플리케이션 보안을 모두 유지하려면 IAM(Identity and Access Management) 인프라와 API 게이트웨이 둘 다에 적용되는 보안 정책이 반드시 필요합니다. 포괄적인 보안 정책을 수립하면 그린필드 애플리케이션과 API 지원 모놀리식 또는 패키지 애플리케이션 간의 보안 통신이 가능해집니다.



- **모듈식 아키텍처 활용:** 마이크로서비스 사용 시에는 소프트웨어 개발 과정이 기존과는 달라집니다. 즉, 개발자는 API를 통해 연결되는 독립적인 서비스를 생성하게 됩니다. 상호 독립적이며 특수한 서비스인 마이크로서비스를 사용하는 경우 재사용 가능성, 더 효율적인 확장 처리 기능, 개발 시간 단축 등의 여러 가지 이점이 제공됩니다. API를 사용하면 개별 독립 실행형 마이크로서비스로 분해한 애플리케이션 모듈을 독립적으로 호출할 수 있습니다. 그러면 이러한 모듈을 기존 애플리케이션과 통합하여 함께 사용할 수 있습니다. 기업은 애플리케이션 기능을 개선할 수 있는 상황을 평가해야 합니다. 예를 들어 수평 확장 방식을 개선하면서 모듈 분리 시 발생 가능한 데이터 동기화 문제도 고려해야 합니다. Expedia Group의 예를 들어 보면, 호텔이나 공유 숙박 소유자 같은 파트너가 Expedia Group 웹사이트에 부동산 사진과 이미지를 업로드합니다. 이 경우 어떤 애플리케이션에서나 이미지 변환 마이크로서비스를 독립적으로 호출할 수 있습니다. 자체 데이터 저장소가 있는 여러 마이크로서비스로 애플리케이션을 분리하면 이처럼 해당 애플리케이션의 서비스를 지속적으로 제공할 수 있습니다. 애플리케이션을 분리하면 특정 지역 등에서 수요가 더 많은 리소스만 제공하는 방식을 통해 확장성을 높일 수 있습니다.



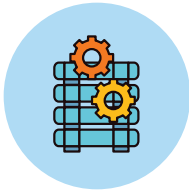
- **더욱 빠른 수명 주기 진행:** 컨테이너, 마이크로서비스 및 함수를 활용하는 개발 프로세스를 진행하면 소스 코드 관리부터 배포까지의 과정을 지속적으로 개선하고 더욱 빠르게 진행할 수 있습니다. DevOps 모델을 도입하면 애플리케이션 기능 요청 대응 능력과 속도를 높일 수 있습니다. 테스트 및 패키징을 포함한 파이프라인 내의 모든 단계는 스크립트를 사용하여 자동화해야 합니다.



- **파이프라인 자동화 및 모니터링을 통해 피드백 제공:** 구축, 테스트 및 배포 단계를 포함한 전체 파이프라인을 자동화할 때는 개발자가 장애 관련 피드백을 즉시 확인할 수 있도록 해당 프로세스를 모니터링해야 합니다. 개발자에게 피드백을 제공하면 향후 자동화할 수 있는 단계와 개선 가능성이 있는 부분을 파악할 수 있습니다.



- **적절한 데이터베이스를 통해 효율적으로 데이터 활용:** 분산 애플리케이션의 확장, 보안 및 가용성을 처리하기에 적합한 데이터베이스 유형을 선택할 때는 여러 가지 옵션 중에서 선택이 가능합니다. 전자 상거래, 게임, 지역 및 사물 인터넷(IoT) 등의 다양한 사용 사례를 지원하기 위해 관계형, 그래프, 키-값, 시계열 데이터베이스 등의 옵션을 사용할 수 있습니다. 오픈 소스 데이터베이스 인스턴스를 선택하면 비용을 줄일 수 있으며, 인 메모리 캐싱 서비스를 선택하면 모바일 및 웹 애플리케이션에서 대량의 요청을 처리할 수 있습니다.



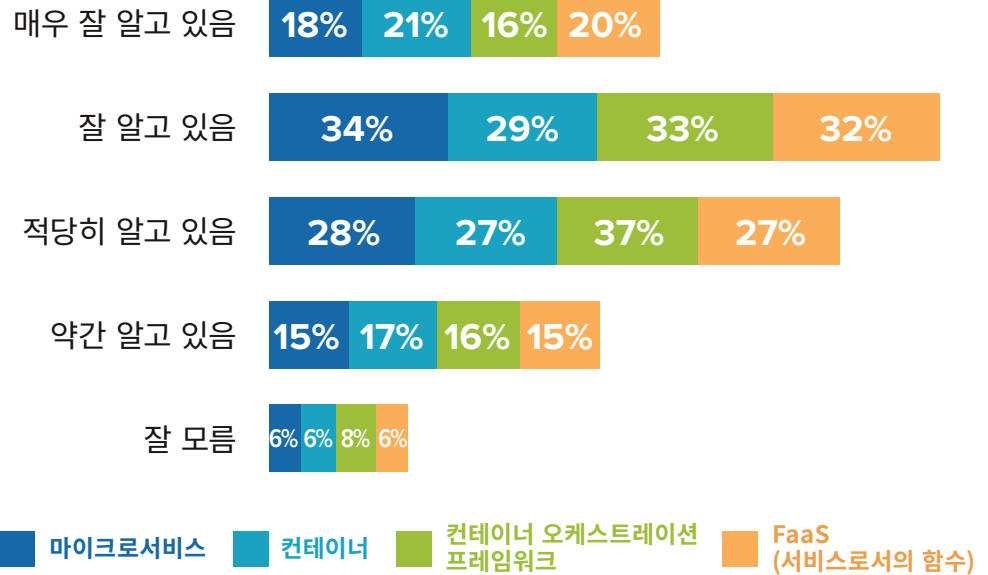
- **서버리스 모델 활용:** 서버리스 기술을 활용하면 인프라 관리 작업을 줄이는 동시에 고가용성을 기본적으로 제공할 수 있습니다. 따라서 개발자는 원하는 언어로 비즈니스 로직을 작성하는 작업에 집중할 수 있습니다. 결제 금액은 작업 단위를 기준으로 측정되며 가격은 실행 시간을 기준으로 책정됩니다. 이러한 작업 단위와 실행 시간은 자동으로 확장 및 축소할 수 있으므로 총 소유 비용도 낮아집니다. 서버리스 운영 모델을 도입하면 개발자는 핵심 역량을 중점적으로 개발하고, 부가가치를 제공하지 않는 작업은 클라우드 서비스 공급자가 처리하도록 할 수 있습니다. 서버리스 모델 평가 시에는 콜드 시작 관련 문제를 고려해야 합니다. “콜드” 함수를 초기화하는 데 시간이 많이 걸리면 빠르게 확장해야 하는 애플리케이션의 경우 문제가 될 수 있습니다. 대다수 공급업체에서는 이 문제의 해결 방법으로 사전 워밍업 함수를 제공하고 있습니다. 서버리스 기술의 발전과 함께 콜드 시작 문제가 해결되고 있으며, 새로운 개발 아키텍처에 대한 인지도도 높아짐에 따라 애플리케이션 개발 시 서버리스 아키텍처를 최우선으로 선택하는 기업도 늘어나고 있습니다.



- **현대화 시작 전략으로 컨테이너화 고려:** 클라우드 서비스 사용 시에는 공급자가 인프라의 여러 수준에서 대다수 관리 작업을 처리해 주므로, 사용자가 복잡한 워크로드 관리 작업을 수행할 필요가 없습니다. 컨테이너 기반 애플리케이션의 경우 클라우드 서비스 공급자가 클러스터를 관리하며, 이벤트 중심 애플리케이션의 경우에는 공급자가 전체 백엔드를 관리합니다. IoT 및 스트림 처리 애플리케이션은 이벤트 기반 컴퓨팅에 적합합니다. 그에 따라 모든 신규 애플리케이션의 기본 옵션으로 서버리스를 선택하여 경쟁력을 높이는 기업이 갈수록 늘어나고 있습니다.

## 그림 2. 현대적 애플리케이션 아키텍처에 대한 친숙도

**질문:** FaaS(서비스로서의 함수), 컨테이너 오케스트레이션 프레임워크, 컨테이너, 마이크로서비스 등에 대해 어느 정도나 알고 계십니까?



nn = 2,500 출처: IDC의 PaaSView and the Developer Survey, 2019년





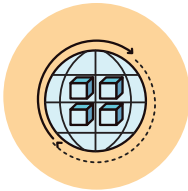
## AWS고객 프로파일 :

온라인 여행사인 Expedia Group은 세계적으로도 유명한 온라인 여행 브랜드가 포함된 광범위한 브랜드 포트폴리오를 보유하고 있습니다. 후발 업체들이 데이터를 수집하여 고객에게 획기적인 서비스를 신속하게 제공하고 있는 상황에서, Expedia Group의 목표는 여행 업계의 고객 경험을 지속적으로 개선하는 것입니다.

가격의 영향을 매우 많이 받는 숙박 업계는 고객의 선호도를 반영하기 위해 지속적으로 변화하고 있습니다. 이에 따라 여행 플랫폼에서는 더욱 간편한 고객 경험을 제공하기 위해 고객이 휴가 계획을 세울 때 광범위한 옵션을 제공하고 있습니다. 하지만 모놀리식 애플리케이션을 활용하는 여행 플랫폼을 제공하는 기업은 고객에게 독특한 경험을 제공하기가 어렵습니다. 즉, 계속해서 늘어나는 여행 서비스 공급업체의 데이터를 연결할 수 있는 경험을 인터페이스 하나로 제공할 수는 없습니다. 고객의 여행 플랫폼 사용 방식은 시간, 지역, 계절에 따라 크게 달라집니다. 따라서 수요를 기준으로 확장 가능한 기술을 개발하는 동시에 여러 인터페이스를 통해 연결하는 사용자에게 고가용성을 제공할 수 있어야 합니다.

Expedia Group의 포트폴리오에 포함된 다양한 여행 브랜드는 여행을 최대한 즐기려는 고객에게 독특한 경험을 제공하고자 합니다. Expedia Group의 전체 브랜드는 여행 정보 조사, 계획 및 예약과 관련한 모든 서비스를 제공합니다. 가장 쾌적한 항공편 좌석 선택, 다른 이용자들의 호텔 리뷰 확인, 목적지 도착 후의 일정 계획 등 다양한 서비스를 이용할 수 있습니다. 관광 여행과 출장 관련 서비스가 모두 포함되어 있는 Expedia Group 포트폴리오에서는 실속형, 럭셔리 등의 다양한 취향 및 예산 옵션을 제공합니다. 수많은 여행 서비스 공급업체(숙박 시설 1백만 개 이상, 항공사 500개 이상, 유람선 업체 수십 개)에 연결되는 여행 플랫폼을 업데이트하거나 개선하려면 기능을 상호 독립적으로 신속하게 테스트하여 릴리스할 수 있는 현대적 아키텍처가 필요합니다.

IDC는 Expedia Group의 자회사인 Vrbo 소속 최고 엔지니어 Kuldeep Chowhan과의 면담을 진행하여 Expedia가 모놀리식 앱을 리팩터링해 현대적 아키텍처로 전환한 과정에서 활용한 모범 사례를 확인했습니다.



- **공통 구성 요소 재사용:** Expedia Group은 여러 국가에서 사용되는 다수의 여행 예약 웹사이트를 운영하고 있습니다. 이처럼 대규모로 웹사이트를 운영하기 위해 Expedia Group은 공통 애플리케이션 구성 요소를 여러 애플리케이션에서 재사용할 수 있도록 패키지로 만들어 70개가 넘는 국가에서 사용되는 200개 이상의 여행 예약 웹사이트에 제공하고 있습니다. Expedia Group은 이 방식을 통해 4,000개 이상의 마이크로서비스를 운영하고 있습니다. 이처럼 공통 구성 요소를 활용하는 경우 조직 전체에서 효율성, 단순성 또는 확장성을 높일 수 있기 때문입니다. 또한 Expedia Group은 AWS Lambda를 활용해 이러한 공통 구성 요소 중 몇 가지의 데이터 준비를 지원하고 있습니다. 기술을 통한 가치 창출 요인이 확인되면 Expedia Group의 엔지니어링 기술을 신속하게 활용하여 확장 가능한 공통 패턴을 결정합니다. 그러면 모든 실무 팀과 기술 팀이 해당 패턴을 최대한 활용할 수 있습니다. Expedia Group은 분산형 시스템 아키텍처를 사용하므로 여러 팀이 반복적인 작업을 통해 조직적으로 기술 수준을 높일 수 있습니다. 그러므로 지속적 사업 확장을 중단하지 않고 Expedia Group의 기술을 계속해서 개선할 수 있습니다. 즉, Expedia Group은 대응 능력과 엔지니어링 효율성을 중요시하는 기업 문화를 확립하여 사업 확장과 기술 개선 간의 균형을 적절하게 유지하고 있습니다. 하지만 여기에 만족하지 않고 앞으로도 엔지니어링 효율성을 최우선으로 고려하는 기업 문화를 통해 업무 방식을 지속적으로 개선할 계획입니다.



- **신속한 배포:** 현대적 아키텍처를 사용하도록 애플리케이션을 리팩터링하는 경우 제공되는 이점 중 하나는 기능 변경 사항을 프로덕션 환경에 배포하는 데 걸리는 시간을 단축하면서 변경된 기능의 작동 신뢰도는 높일 수 있다는 점입니다. 이러한 이점을 활용하려면 애플리케이션 수명 주기의 지속적 통합/연속 배포 파이프라인을 자동화해야 합니다. Expedia Group은 A/B 테스트 프레임워크를 활용해 일부 고객에게 기능을 롤아웃하여 테스트를 진행합니다. 이 방식을 적용하면 개발자가 새 기능을 적용해 본 후 전 세계 릴리스 전에 고객 요구를 충족하는 기능만 롤아웃할 수 있습니다. 또한 Expedia Group은 이 방식을 통해 변경 사항을 몇 분 내에 배포할 수 있으므로 비즈니스 사용자의 기능 요청 백로그를 줄일 수 있습니다. 모놀리식 애플리케이션 사용 시에는 변경 사항을 배포하려면 몇 주, 심지어는 몇 달이 걸리기도 했습니다. Expedia Group은 현대적 릴리스 파이프라인에서 기능을 설정 및 해제한 다음 약 1시간 이내에 전 세계 애플리케이션에 해당 기능을 적용하는 기능을 활용함으로써 새로운 기능을 빠르게 릴리스하고 있습니다.



배포 과정에서 발생하는 위험을 줄이려면 새 기능이 포함된 페이지로는 전체 트래픽 중 5%만 라우팅하고, 새 코드가 사용자의 페이지 사용 과정에서 문제를 발생시키지 않고 정상적으로 작동하는지 확인해야 합니다. 새 코드가 정상 작동함을 확인한 후에는 5단계에 걸쳐 라우팅하는 트래픽의 양을 5%에서 100%까지 늘려야 합니다. 새 기능의 신뢰도가 높아지면 전체 사용자 중 5%를 대상으로만 테스트 트래픽을 전송하는 방식에서 전체 트래픽을 전송하는 방식으로 신속하게 전환할 수 있습니다. 즉, 코드의 완성도가 높아지면 배포 속도도 급격하게 빨라집니다.



- **DevOps:** 애플리케이션과 관련된 Expedia Group의 핵심 엔지니어링 원칙 중 하나는 “모든 애플리케이션은 구축 담당자의 소유”라는 것입니다. 기존에는 특정 배포 기능을 일부 개발자만 사용할 수 있었지만 이제는 애플리케이션 팀 전체에서 배포를 처리합니다. 엔지니어링 팀에서 모든 도구를 구축하므로 개발자는 로깅 지표, 분산 추적 등의 모든 모니터링 솔루션을 사용하여 애플리케이션을 구축 및 배포할 수 있습니다. DevOps 사례를 활용하는 경우 개발 팀에 애플리케이션 관리용 도구를 제공할 수 있으므로 소스 코드 작성부터 배포까지의 수명 주기에서 팀 간에 마찰이 발생하지 않습니다. Expedia Group이 컨테이너, 마이크로서비스 및 서버리스 플랫폼을 사용하도록 현재 애플리케이션을 리팩터링하기 위해 수행한 단계를 전반적으로 관찰한 결과, 리팩터링 이후 출시 시간이 크게 단축되었음을 확인할 수 있었습니다. 구체적으로 설명하자면, 모놀리식 애플리케이션 사용 시 Expedia Group이 새 애플리케이션과 기능을 릴리스하려면 몇 달이 걸렸지만 현대적 애플리케이션 플랫폼에서는 하루에도 몇 번씩 릴리스를 진행할 수 있습니다. Expedia Group은 DevOps 사례 및 애플리케이션 아키텍처 현대화에 투자한 결과 고객에게 이전보다 훨씬 많은 기능을 제공할 수 있게 되었습니다. 예를 들어 Vrbo의 경우 지난 3년 동안 릴리스 속도가 12배나 높아졌습니다. 이처럼 새 기능을 쉽게 적용해 볼 수 있게 된 Expedia Group은 고객에게 새 기능을 배포하는 과정의 위험을 줄일 수 있었습니다. 또한 재사용 가능성이 높아져 조직 내의 기능 개발 속도도 빨라졌습니다.



# Amazon Web Services를 사용한 애플리케이션 현대화

애플리케이션 설계를 시장 경쟁력을 높이기 위한 핵심 구성 요소로 간주하는 기업이 갈수록 늘어나고 있습니다. 그에 따라, 업무상의 문제를 해결하기 위해 애플리케이션 현대화를 결정하는 기업도 많습니다. 기업은 유용한 제품을 구축하여 더욱 빠르게 제공함으로써 신규 고객을 확보하고자 합니다. 애플리케이션 현대화를 진행하면 이 목표를 달성할 수 있습니다.

AWS 포트폴리오에서는 조직이 현대적 애플리케이션을 구축하고 레거시 애플리케이션을 현대화하는 데 필요한 서비스를 제공합니다. 해당 분야에서 심층 전문 지식을 보유한 AWS는 현대적 애플리케이션을 구축 및 운영하는 데 필요한 모든 서비스를 제공하며, 여기에는 서버리스 기술, 컨테이너, 특별히 구축된 광범위한 데이터베이스, 개발자 도구, 기계 학습 등이 포함됩니다. 이러한 이유로 인해 빠르게 성장하는 스타트업, 대기업 및 주요 정부 기관을 비롯한 수백만의 고객이 AWS를 통해 인프라를 지원하고 대응 능력을 높이는 동시에 비용을 줄이고 있습니다.

대다수 AWS 오퍼링은 서버리스 도구를 통해 관리되므로 조직 내에서 개발자와 운영자의 역할이 달라집니다. 예를 들어 AWS Lambda를 사용하는 기업의 경우 비즈니스 로직 코드만 작성하면 되고 프로비저닝, 확장 및 패치 적용은 모두 AWS에서 처리합니다. 관리형 서비스의 사용량이 늘어나는 동시에 릴리스 파이프라인 등의 기능 자동화 비율도 증가함에 따라 운영 팀은 기업에서 더욱 전략적인 역할을 수행할 수 있으며 개발자는 배포를 고려하여 애플리케이션을 설계할 수 있습니다. 이처럼 역할이 변화하면 특정 제품을 기준으로 통합된 소규모 자율 교차 기능 팀을 만들 수 있습니다. 현대적 애플리케이션의 이점을 효율적으로 활용하려면 관련 기술 자체를 개발하는 동시에 현대적 팀을 구축하고 조직 문화도 확립해야 합니다.

# 향후 전망

디지털 트랜스포메이션은 최근의 기업 이니셔티브에서 최우선으로 진행해야 하는 작업이 되었습니다. 최신 기술을 효율적으로 활용하여 ‘기술적 부채’를 발생시키지 않는 신흥 스타트업이 기존과는 전혀 다른 사업 방식을 활용하는 환경에서는 디지털 트랜스포메이션을 반드시 진행해야 합니다. 디지털 트랜스포메이션 과정에서 클라우드 공급자가 제공하는 다양한 클라우드 네이티브 서비스를 활용하는 기업은 인프라 효율성을 높이는 동시에 애플리케이션을 더욱 빠르게 개발할 수 있습니다. 오늘날에는 인력 노후화로 인해 레거시 애플리케이션 지원 비용이 갈수록 증가하고 있습니다. 따라서 기업은 클라우드 네이티브 아키텍처를 활용하여 애플리케이션을 현대화하는 이니셔티브를 즉시 진행해야 합니다.

현대적 애플리케이션 개발 및 배포 플랫폼에서는 모놀리식 애플리케이션을 여러 구성 요소(마이크로서비스)로 리팩터링할 수 있습니다. 서버리스 운영 모델을 도입하면 이러한 리소스를 관리하는 복잡한 작업을 줄일 수 있습니다. 그러므로 코드 변경부터 프로덕션까지의 과정을 진행하는 기업은 이러한 리소스를 안전하게 관리할 수 있는 개발자 도구 세트를 클라우드 공급자가 제공하는지를 확인해야 합니다.





# 과제/기회

오늘날의 기업은 기존 기술을 통해 사업상의 요구를 충족하기가 어렵다는 점을 잘 알고 있지만, 기존의 조직 문화에서는 IT 부서가 일상적인 운영 과정에 새로운 클라우드 기술 패러다임을 활용하기가 매우 어렵습니다. DevOps를 도입하면 애플리케이션 개발 및 배포 수명 주기를 자동화하여 더욱 빠르게 진행할 수 있습니다. 하지만 개발자가 제출한 코드를 수동으로 검토하는 방식에 익숙한 직원들은 소프트웨어 릴리스 프로세스 자동화를 선호하지 않습니다. AWS에는 폭넓은 내부 경험을 토대로 하여 구축된 다양한 서비스가 포함되어 있으며, 이러한 서비스를 활용하면 자동화를 통해 소프트웨어 구축 및 배포 방식을 획기적으로 개선할 수 있습니다. 하지만 기업이 클라우드 기술 도입 과정을 시작하여 계속 신속하게 진행할 수 있도록 적절한 기술도 제공해야 합니다. 이러한 고객은 전문적인 클라우드 기술을 보유한 AWS 파트너가 참여하는 대규모 에코시스템을 통해 적절한 지원을 받을 수 있습니다.



## 결론

신규 사업 기회를 활용하려면 현대적 기술 플랫폼을 적절하게 구축해야 합니다. 현대적 기술 플랫폼을 구축하려면 조직 문화를 변경하는 동시에 현대적 개발 도구 적용을 위한 신기술도 습득해야 합니다. 클라우드 도입 과정을 진행하는 기업은 기존 애플리케이션 리팩터링 단계를 반드시 진행해야 합니다. IDC의 예측 결과에 따르면 2023년까지 ICT(정보 통신 기술) 지출액 중 50% 이상은 디지털 트랜스포메이션과 혁신 분야에 직접 투입될 예정이라고 합니다. 이 백서 작성을 위해 면담을 진행했던 Expedia Group과 같이 디지털 트랜스포메이션 목표를 달성하려는 기업은 현대적 소프트웨어 배포 방법과 도구를 도입하여 업무상의 요구를 충족하는 애플리케이션 기능을 신속하게 개발하여 배포해야 합니다.

## IDC 소개

IDC(International Data Corporation)는 정보 기술, 통신 및 소비자 기술 시장을 위한 마켓 인텔리전스, 자문 서비스 및 이벤트를 제공하는 세계적 선도 기업입니다. IDC는 IT 전문가, 기업 경영인, 투자 업계가 기술 구매 및 사업 전략에 있어 사실에 근거한 결정을 내리도록 돕습니다. 1,100명 이상의 IDC 애널리스트들이 전 세계 110여 개국에서 전 세계는 물론 특정 국가와 지역을 대상으로 기술 및 업계의 기회와 동향에 대한 관한 전문 지식을 제공합니다. 50년 동안 IDC는 고객이 주요 사업 목표를 달성하는 데 도움을 주는 전략적 통찰을 제공해 왔습니다. IDC는 세계 최고의 기술 미디어, 연구조사 및 이벤트 회사인 IDG의 자회사입니다.



## IDC Research, Inc.

5 Speen Street  
Framingham, MA 01701  
USA  
508.872.8200  
Twitter: @IDC  
idc-community.com  
www.idc.com

## 저작권 고지

IDC 정보 및 데이터의 외부 출판 - IDC 정보를 광고, 보도 자료 또는 홍보 자료에 사용하려면 사전에 IDC의 담당 부사장 또는 해당 국가 관리자로부터 서면 승인을 받아야 합니다. 그러한 요청을 할 경우 제안서 초안을 첨부해야 합니다. IDC는 어떠한 이유로든 외부 사용 승인을 거부할 권리가 있습니다.

저작권 2020 IDC. 서면 허가 없이 복제하는 행위는 전적으로 금지됩니다.

