



풀 스택 웹 및 모바일 앱 개발 가속화

현대적 앱 개발을 가속화하는 3가지 검증된 팁



웹 및 모바일 애플리케이션 구축 가속화

많은 조직이 고객 만족도를 높이고 직원을 지원할 수 있는 현대적 웹 및 모바일 애플리케이션으로 더욱 빠르게 혁신하기를 원합니다. 이 목표를 달성하려면 개발 팀은 점차 높아지는 기준을 충족하여 사용자 경험과 복잡한 클라우드 기능을 개선하면서 확장 가능하고 안전한 고성능 애플리케이션을 저렴한 비용으로 배포해야 합니다.

현대적 웹 및 모바일 애플리케이션을 구축하려면 팀은 확장 가능한 풀 스택 애플리케이션을 빠르고 성공적으로 제공할 수 있어야 합니다. 초보 개발자와 숙련된 개발자의 기술 격차를 줄이고 내부 프로세스와 통합되는 확장 가능한 옵션을 활용하여 애플리케이션을 빠르게 출시할 수 있는 개발자 도구를 모색하세요.

개발 프로세스를 빠르게 진행하려면 여러 워크스트림에 분산되고 배포된 데이터의 이용을 간소화하는 솔루션을 찾아야 합니다. 이 솔루션을 사용하면 프런트엔드 팀이 데이터 기반 기능을 빠르고 효율적으로 구축할 수 있습니다. 애플리케이션이 현재의 요구 사항에 맞으면서 유연한 조정을 통해 미래의 요구 사항도 충족할 수 있는 아키텍처를 선택하세요. 무엇을 선택하든 애플리케이션 성능과 보안을 최우선으로 고려해야 합니다.

이 가이드에서는 웹 및 모바일 애플리케이션 개발을 가속화하는 3가지 팀을 소개합니다.

- 1** **현대적 도구와 프레임워크** – 유연한 목적별 도구와 프레임워크로 프런트엔드 팀 역량 강화
- 2** **현대적 API** – 통합 GraphQL 및 게시/구독 API로 배포된 데이터 소스 액세스 간소화
- 3** **현대적 컴퓨팅** – 관리형 및 서버리스 서비스로 관리 문제와 복잡성 축소

유연한 목적별 도구와 프레임워크로 프론트엔드 팀 역량 강화

현황

프론트엔드 개발자는 JavaScript, Swift, Kotlin 같은 프론트엔드 프로그래밍 언어, React, React Native, Flutter, Angular 같은 인기 프레임워크, 통합 개발 환경(IDE) 같은 도구를 사용하여 애플리케이션의 사용자 경험을 차별화합니다. 애플리케이션의 프레젠테이션 계층을 코딩하고 API를 통해 데이터에 액세스하고 통합하며 실시간 및 오프라인 기능을 활용하여 웹 애플리케이션 또는 모바일 애플리케이션에 동영상이나 채팅 서비스와 같은 클라우드 기반 기능을 구축합니다.

프론트엔드 개발자는 사용자 인터페이스(UI) 구축의 전문가일 수 있지만 대개 클라우드 전문가가 아닙니다. 웹 또는 모바일 애플리케이션에 클라우드 기능을 구축하고 연결하는 일은 복잡하고 시간이 많이 소모되기 때문에 프론트엔드 개발자들이 사용자 경험을 개선하는 데 집중할 시간이 줄어듭니다.

솔루션

일반적인 사용 사례 구현에 필요한 코드의 양을 줄이는 개발 도구와 프레임워크로 프론트엔드 팀이 생산성을 높일 수 있도록 지원합니다. 인증 및 데이터 액세스와 같은 복잡한 작업을 처리하는 기능을 갖춘 프레임워크를 활용하고 캐싱과 재연결, 데이터 동기화, 충돌 해결을 처리하는 고급 기능을 찾아봅니다. 계속해서 코드 기반 애플리케이션 기능을 사용자 지정하거나 확장할 수 있으면서 개발 과정을 간소화하는 도구를 선택합니다. 클라우드 학습을 용이하게 하는 도구를 선택하면 설정을 빠르게 마치고 워크플로를 간소화하며 보안을 자동화할 수 있으므로 프론트엔드 팀은 고객의 문제를 해결하는 데 집중할 수 있습니다.

AWS가 제시하는 이점

AWS Amplify의 도구와 서비스를 통해 프론트엔드 개발자는 공통의 Amazon Web Services(AWS) 지원 백엔드를 사용하는 웹 애플리케이션 및 네이티브 모바일 애플리케이션을 빠르게 구축할 수 있습니다. **AWS Amplify 라이브러리**는 클라우드에서 이미지와 동영상 저장과 같은 일반적인 사용 사례를 애플리케이션에 간단히 통합할 수 있도록 지원하고 널리 사용되는 프론트엔드 프레임워크, 언어, 모바일 플랫폼 등과 긴밀히 통합됩니다. 또한 Amplify command line interface(CLI)와 관리자 UI는 개발자가 필요에 따라 확장 가능한 AWS 백엔드를 빠르게 구축하는데 도움이 됩니다. Amplify는 서비스형 백엔드(BaaS) 제품 및 서비스와 달라서 애플리케이션 사용 사례의 변화에 발맞추어 개발자에게 200여 개의 다양한 AWS 서비스를 제공합니다.

AWS Amplify Studio를 통해 **AWS Management Console** 외부에서도 웹 및 모바일 애플리케이션 전체를 몇 시간 만에 쉽게 구축하여 릴리스할 수 있습니다. Amplify Studio를 사용하면 애플리케이션 백엔드를 신속하게 구축하고 풍부한 UI React 구성 요소를 생성하며 클릭 몇 번으로 UI를 백엔드에 시각적으로 바인딩할 수 있습니다. 개발자는 백엔드 전문 지식 없이도 시각적 인터페이스를 사용하여 데이터 모델, 사용자 인증, 파일 스토리지를 정의할 수 있습니다.

또한 뉴스 피드, 문의 양식, 전자상거래 카드, 기본 요소(버튼, 텍스트 필드, 경고)와 같은 수십 가지의 인기 있는 React 구성 요소 중에서 선택한 후 스타일 가이드에 맞게 사용자 정의하여 프론트엔드 UI 애플리케이션을 빠르게 개발할 수 있습니다. **AWS Amplify Studio Form Builder**를 사용하면 개발자가 React 코드로 자동 생성된 아름다운 React UI 양식을 구축하여(데이터 모델 또는 JSON 객체를 기반으로 생성하거나 처음부터 React 코드로 자동 생성) 시간을 절약할 수 있습니다. Amplify Studio 자동 생성 양식은 코드에서 완벽하게 사용자 지정할 수 있어 양식이 프로젝트를 맞게 조정됩니다.

또한 Amplify Studio는 Figma와 통합되어 디자이너와 개발자 간 워크플로를 간소화합니다. Figma 디자인-코드 접근 방식을 사용하면 개발자는 디자이너가 구축한 Figma 프로토타입을 클린 React 코드로 Amplify Studio에 가져올 수 있으므로 상용구 코드를 작성할 필요가 없습니다. Amplify가 자동으로 기반이 되는 비즈니스 로직과 배포 코드를 생성하므로 개발자는 선호하는 프로그래밍 언어로 사용자 정의 비즈니스 로직 코드를 사용하여 애플리케이션을 확장할 수 있습니다.

또한 개발자는 **AWS Amplify Hosting**을 사용하여 React, Vue.js 또는 Next.js 웹 애플리케이션을 배포하고 호스팅할 수 있습니다. 생산성을 극대화하기 위해 Amplify 콘솔에 Git 워크플로를 연결하거나 파일을 업로드하여 기존 도구와 AWS 서비스를 원활하게 통합하는 것도 가능합니다. 그런 다음 구축 설정을 구성하고 Amplify 콘솔에서 직접 **Amazon CloudFront**의 콘텐츠 전송 네트워크(CDN)로 배포하면 됩니다. 이 올인원 솔루션을 사용하면 모든 코드 커밋에서 정적 웹 또는 서버 측 렌더링 애플리케이션, 모바일 앱 랜딩 페이지 또는 프로그레시브 애플리케이션을 지속적으로 배포하여 조직의 성장에 따라 확장되는 빠르고 안전하며 안정적인 애플리케이션을 구축할 수 있습니다.

프론트엔드 개발자는 위 이점을 제공하는 Amplify를 활용하여 그 어느 때보다 빠르게 기능이 풍부한 웹 및 모바일 애플리케이션을 구축할 수 있습니다. Amplify는 일반적인 시나리오의 모범 사례를 인코딩하고 주도적으로 전달함으로써 AWS에서 풀 스택 애플리케이션을 구축하는 데 필요한 전반적인 단계와 코드 행 수를 줄여줍니다. 동시에 비즈니스 요구 사항 변경에 따라 애플리케이션을 사용자 정의하는 유연성을 지원합니다.

고객 사례

Branch Insurance, AWS에서 애플리케이션 개발 가속화

당면 과제

Branch Insurance는 보험 고객의 이름과 주소라는 두 가지 간단한 정보만으로 거의 정확한 보험 가격을 제시하여 보험 고객의 최종 사용자 경험을 크게 간소화하기 시작했습니다. 그러나 이렇게 간소화하기 위해서는 규정을 준수하여 데이터를 신속하게 처리하고 효율적이고 안전하게 저장할 수 있는 강력한 인프라가 필요했습니다. Branch Insurance는 **AWS AppSync**를 백엔드 인프라와 API 서비스의 기반으로 활용하고 이를 Amplify Libraries에 연결하여 권한 부여 흐름을 관리함으로써 애플리케이션 운영 관리에 소요되는 시간을 최소화할 수 있었습니다.

사용한 AWS 도구 및 서비스

- [AWS Amplify](#)
- [AWS AppSync](#)
- [Amazon Cognito](#)
- [Amazon DynamoDB](#)

결과

- 앱 개발 기간을 6개월 단축
- 20명 미만의 개발자로 3년 만에 4개 제품 출시

사례 연구 전체 내용 보기 >



BRANCH

"AWS에서 제품을 구축하는 것은 마치 '쉬운 모드'를 사용하는 것과 같습니다. 관리형 서비스를 사용하여 많은 기능을 간소화할 수 있기 때문이죠. 비즈니스 로직과 인터페이스를 작성하기만 하면 됩니다. 이것이 AWS 사용의 뛰어난 장점입니다."

Joe Emison,
Branch Insurance Co-Founder & CTO

통합 GraphQL 및 게시/구독 API로 배포된 데이터 소스 액세스 간소화

현황

소프트웨어 아키텍처는 끊임없이 진화하고 있습니다. 조직은 전체적인 모놀리식에서 분산 아키텍처로 이동하거나 때로는 두 가지를 조합하여 사용하고 있습니다. 기술 세트, 애플리케이션 또는 프로그래밍 모델을 비롯한 여러 요인에 따라 사용할 아키텍처가 달라집니다. 독립적인 마이크로서비스 또는 구성 요소로 이루어진 분산 아키텍처를 선택하면 팀이 애플리케이션의 여러 부분에서 독립적으로 작업하므로 개발 속도를 높일 수 있습니다. 또한 분산 아키텍처는 내결함성을 높입니다. 장애가 발생한 특정 구성 요소를 격리하여 다른 애플리케이션 구성 요소에 미치는 영향을 최소화할 수 있습니다.

마이크로서비스를 사용하면 애플리케이션의 규모 조정이 쉽고 개발이 빨라져서 혁신을 달성하고 시장 출시를 가속화합니다. 그러나 마이크로서비스를 채택하면 프론트엔드 팀에서 분산된 마이크로서비스와 데이터 소스에 연결해야 할 수 있습니다. 그러면 사용자용 애플리케이션에 필요한 데이터에 액세스하기가 더 어려워집니다. 백엔드 팀은 프론트엔드 팀이 애플리케이션에서 데이터를 쉽게 사용할 수 있도록 데이터 액세스를 간소화하여 웹 및 모바일 개발 가속화에서 중요한 역할을 합니다.

솔루션

GraphQL과 같은 기술은 여러 개의 백엔드 데이터 소스에서 데이터 액세스와 통합의 속도와 효율을 높입니다. GraphQL은 API용 쿼리 언어이자 기존 데이터로 해당 쿼리를 수행하기 위한 런타임입니다. GraphQL은 API에 있는 데이터를 완전히 이해할 수 있는 설명을 제공하기 때문에 프론트엔드 개발자는 여러 백엔드 데이터베이스와 마이크로서비스에서 각 API 호출에 필요한 데이터를 정확히 쿼리할 수 있습니다. 백엔드 개발 팀은 GraphQL을 사용하여 기존 데이터 소스를 하나의 엔드포인트로 추상화하고 프론트엔드 팀이 더 쉽게 데이터에 액세스할 수 있게 합니다. 기존 API 세트에 대해 여러 엔드포인트를 하나의 GraphQL API로 추상화하면 API를 마이그레이션하거나 처음부터 애플리케이션을 다시 작성할 필요가 없다는 장점이 있습니다.

GraphQL을 사용하여 통합 API 계층을 만들면 REST API 전략을 보완하여 배포된 서비스 아키텍처와의 상호 작용을 단순화할 수 있습니다. 또한 구독을 포함하도록 API를 확장하면 GraphQL 사용자가 모든 이벤트 소스의 업데이트된 실시간 데이터를 구독하는 웹, 모바일, 사물 인터넷(IoT) 클라이언트에 게시할 수 있습니다.

AWS가 제시하는 이점

AWS AppSync는 개발자가 안전하고 성능이 우수한 서버리스 GraphQL 및 Pub/Sub API를 통해 앱을 데이터와 이벤트에 연결할 수 있도록 하여 최신 웹 애플리케이션과 모바일 애플리케이션의 개발을 간소화합니다. AWS AppSync는 **Amazon DynamoDB, Amazon EventBridge, AWS Lambda**와 같은 데이터 소스에 안전하게 연결하는 등의 어려운 작업을 처리합니다. AWS AppSync는 완전관리형 GraphQL API와 Pub/Sub API의 설정과 관리에 더해 내장형 보안, 모니터링, 로깅, 추적을 통한 Auto Scaling과 고가용성, 지연 시간 단축을 위한 캐싱(선택 사항)을 제공합니다. 지금은 사용자가 지정된 Virtual Private Cloud(VPC) 엔드포인트에서만 API에 액세스하도록 제한 설정할 수 있는 **프라이빗 API**도 제공합니다.

AWS AppSync GraphQL API는 여러 데이터베이스, 마이크로서비스, API에서 데이터를 안전하게 쿼리하거나 업데이트하는 단일 엔드포인트를 제공하여 애플리케이션 개발을 간소화합니다. GraphQL을 사용하면 단일 네트워크 요청으로 하나 이상의 데이터 소스에서 적절한 데이터를 애플리케이션에 제공하고 여러 데이터 소스를 단일 GraphQL API로 결합할 수 있습니다.

AWS AppSync Pub/Sub API는 서버리스 WebSocket 연결을 통해 구독한 API 클라이언트에 업데이트된 데이터를 자동으로 게시하여 매력적인 실시간 경험을 만드는 작업을 간소화합니다. GraphQL API를 확장하여 구독을 포함하면 모든 이벤트 소스의 업데이트된 실시간 데이터를 구독하는 웹, 모바일 및 IoT 클라이언트에 게시할 수 있습니다. GraphQL 없이 단순한 게시/구독 API로 시작할 수도 있습니다. 어느 쪽이든 AWS AppSync는 최대 수백만 개의 연결된 클라이언트에 대해 서버리스 WebSocket 연결 생성, 메시지 브로드캐스트, 팬아웃을 수행합니다.

AWS AppSync는 다른 AWS 서비스와의 통합도 제공하므로 API를 보호하고 모니터링하고 관련 문제를 해결하는 작업을 쉽게 할 수 있습니다. 다음과 같은 서비스와 통합할 수 있습니다.

1. 실시간으로 감사 로그를 모니터링하는 **AWS CloudTrail**
2. 지표 및 로그를 관리하는 **Amazon CloudWatch**
3. 분산형 애플리케이션의 동작을 추적하는 **AWS X-Ray**
4. 일반적인 웹 익스플로잇으로부터 API를 보호하는 **AWS WAF**



고객 사례

ADP, MyADP 및 ADP 모바일 애플리케이션에서 진화하는 단일 글로벌 경험 제공

당면 과제

Automatic Data Processing(ADP)는 플래그십 데스크톱 및 모바일 솔루션인 MyADP와 ADP Mobile을 현대화하여 1,700만 명의 사용자에게 원활한 사용자 경험(UX)을 제공하고자 했습니다. 이를 위해 ADP는 애플리케이션을 통합, 표준화, 현대화하기 위해 노력했습니다.

ADP는 오류를 일으키지 않고 서로 다른 레코드 시스템의 경험을 단일화하기 위해 혁신할 필요가 있었습니다. Devi Ramachandran(ADP의 DevOps Senior Director)은 "급여 명세서와 복지 혜택 가입서가 동일한 속도로 열려야 하는데, 이 두 콘텐츠 소스는 서로 다른 인프라 세트에 있습니다."라며 "그것이 처음부터 우리가 해결해야 할 과제였으며 시스템을 AWS로 마이그레이션하면서 간소화되었습니다."라고 말했습니다. ADP는 서로 다른 인프라에서 제공하는 애플리케이션 API 액세스를 간소화하기 위해 AWS AppSync를 사용하여 다양한 백엔드와 소스의 데이터를 단일 엔드포인트로 통합했습니다. 그 결과, 백엔드에서 데이터 집계를 간소화하고 사용자 경험을 개선할 수 있었습니다.

사용한 AWS 도구 및 서비스

- [AWS AppSync](#)
- [Amazon ECS](#)
- [AWS Fargate](#)
- [AWS Lambda](#)

결과

- 앱 스토어 평점 4.5 이상 유지
- 글로벌 UX를 위한 이동성 달성
- 다중 리전 아키텍처를 통한 복원력 향상
- 지연 시간 기반 라우팅으로 지연 시간 감소
- 제한과 오류를 없애기 위해 트래픽 버스트에 맞게 규모 조정

사례 연구 전체 내용 보기 >



"우리가 AWS를 사용하는 이유는 인프라 관리 팀이 아닌 제품 개발 팀이 되고 싶어서입니다."

Devi Ramachandran,
ADP DevOps Senior Director

관리형 서버리스 서비스로 관리 문제와 복잡성 축소

현황

Deloitte에 따르면 IT 리소스의 80%가 비즈니스 운영에 사용되고 혁신에 쓰이는 리소스는 20%라고 합니다. 이 비율을 개선하여 혁신의 비중을 높이는 가장 빠른 방법은 인프라 관리와 같이 비즈니스에 중요하지 않은 활동에 쓰이는 시간을 줄이는 것입니다. 서버, 스토리지, 네트워킹, 운영 체제 관리는 시간과 비용 소모가 많고 구매 주기 때문에 운영이 복잡합니다. 팀은 인프라 관리에 점점 더 많은 시간을 쓰고 있습니다. 따라서 인프라 관리 시간을 줄이는 것은 최적화하고 혁신에 재투자하는 중요한 기회가 될 것입니다.

솔루션

'운영 부담을 축소'하는 가장 쉬운 방법은 인프라에 관리형 서비스 또는 서버리스 서비스를 도입하여 운영 부담을 클라우드 제공업체에 이전하는 것입니다. 프로비저닝, 관리, 서버 패치에 할애하는 시간을 줄이면 개발자는 고객을 만족시키는 제품의 코드를 작성하고 반복하는 혁신 작업에 더 많은 시간을 쏟을 수 있습니다. 서버리스 컴퓨팅에는 프로비저닝하거나 관리할 인프라가 없습니다. 인프라는 서버 단위가 아닌 소비 단위에 따라 자동으로 규모가 조정됩니다. 서버리스에서는 사용하는 컴퓨팅에 대해서만 비용을 지불하면 되고 가용성과 내결함성이 기본으로 제공됩니다. 서버리스 컴퓨팅은 서버 운영의 부담을 없애주기 때문에 애플리케이션을 지원하는 인프라 관리 및 규모 조정과 같은 작업이 아닌 애플리케이션 구축에 좀 더 집중할 수 있게 됩니다.

AWS가 제시하는 이점

AWS Lambda는 첫 시작에 좋습니다. 이 서버리스 컴퓨팅 서비스는 기본 컴퓨팅 환경을 완전히 추상화하면서 코드를 가용성, 확장성, 안전성, 내결함성이 뛰어난 애플리케이션으로 실행합니다. 그러므로 애플리케이션 개발에 주로 집중할 수 있어 출시가 단축되고 총 소유 비용(TCO)이 절감됩니다. Lambda 함수는 이벤트로 트리거할 수 있습니다. 즉, 이벤트가 트리거될 때까지 유휴 상태로 있다가 시스템 전원이 작동합니다. 실제로 200개가 넘는 AWS 서비스, HTTP 호출 및 서비스형 소프트웨어(SaaS) 애플리케이션에서 통합 코드를 한 줄 작성하지 않고도 Lambda 함수를 자동으로 트리거할 수 있습니다.

이벤트가 트리거되면 리소스는 밀리초의 속도로 작업의 필요에 맞게 크기가 자동으로 확장 또는 축소됩니다. 인프라를 프로비저닝하거나 구성할 필요가 없으며 사용한 만큼만 비용을 지불하면 됩니다. 즉, TCO는 최소한으로 유지하면서 프로덕션 시간을 단축할 수 있습니다.

Lambda에서는 최소한의 운영으로 최대의 민첩성을 얻을 수 있지만 경우에 따라 컨테이너 오케스트레이터를 사용하여 컨테이너를 실행할 수도 있습니다. 이 경우, 서버리스로 실행하거나 인프라를 미세 조정하여 비즈니스에 가치를 더할 수 있는 유연성이 생깁니다. 서버리스 컨테이너 서비스인 **AWS Fargate**의 **Amazon Elastic Container Service(Amazon ECS)**를 사용하여 **AWS Lambda**를 보완할 수 있습니다.



고객 사례

ProSiebenSat.1 Media SE, AWS 서버리스 솔루션을 사용하여 대화형 TV 경험 실현

당면 과제

혁신적인 디지털 그룹인 ProSiebenSat.1 Media SE의 전액 출자 자회사인 Seven.One Entertainment Group은 고객 만족도를 높이고 참여를 유도할 수 있는 매력적인 대화형 TV 솔루션을 구축하는 기회를 얻었습니다. 회사는 이벤트 기반 트래픽 피크를 처리하기 위해 서버리스 아키텍처를 사용하여 온프레미스 시청자 참여 솔루션을 AWS로 마이그레이션하기로 결정했습니다.

Seven.One Entertainment Group은 AWS AppSync와 같은 관리형 서비스를 사용하여 대규모 DynamoDB 데이터에 안전하게 연결하는 업무 부담을 이전함으로써 GraphQL API를 개발할 수 있습니다. Lambda의 이벤트 기반 컴퓨팅으로 구동되는 서버리스 아키텍처는 이벤트 중에 규모가 조정되어 시청자의 상호 작용을 지원합니다. AWS AppSync 실시간 구독을 사용하여 수백만 명의 팬에게 거의 실시간으로 결과를 알릴 수 있습니다.

AWS를 사용한 결과, 이 회사는 솔루션의 월별 비용을 60% 절감하고 직원들이 인프라 자체 관리에 시간을 보내는 대신 흥미로운 새 기능을 만드는 차별화 작업에 집중할 수 있도록 했습니다.

사용한 AWS 도구 및 서비스

- [AWS AppSync](#)
- [Amazon DynamoDB](#)
- [Amazon CloudFront](#)
- [AWS Lambda](#)

결과

- 솔루션 월별 비용 60% 절감
- 피크 시간에 13밀리초의 지연 시간 달성
- 비디오 콘텐츠 배포 시간 단축
- 확장성이 향상되어 최종 사용자 트래픽에 API 사용 가능

[사례 연구 전체 내용 보기 >](#)



"AWS를 사용하면 우리가 가장 잘할 수 있는 코드 작성에 집중할 수 있습니다. 운영 체제 패치나 규모 조정에 대해 걱정할 필요가 없습니다."

Fabian Desoye,

Seven.One Entertainment Group Head of Metadata Systems

시작하기

현대적인 웹 애플리케이션과 모바일 애플리케이션 개발을 가속화하는 세 가지 팁을 살펴보았습니다. 지금이라도 당장 시작하고 싶으실 것입니다. AWS에서는 먼저 이 팁 중 한두 가지를 선택하여 현대적 애플리케이션을 개발하는 기존 접근법을 보완하는 것을 권장합니다.

AWS Amplify는 프론트엔드 팀의 역량을 강화하고 풀 스택 애플리케이션 구축에 필요한 전반적인 단계와 코드 행 수를 줄여주기 때문에 개발자가 더 빠르게 앱을 출시할 수 있습니다. 또한 AWS AppSync를 사용해 분산 데이터 소스에 대한 액세스를 간소화하는 것이 가능합니다. 기존 데이터 소스를 하나의 통합된 GraphQL API 엔드포인트로 추상화하면 프론트엔드 팀이 데이터에 쉽게 액세스하여 데이터 기반 기능을 구축할 수 있습니다. 나아가 단순한 게시/구독 API를 사용해 실시간 데이터 경험으로 애플리케이션을 차별화할 수도 있습니다. 또, AWS Lambda와 같은 서비스를 통해 운영 인프라를 줄이는 효과도 있습니다.

어떤 방법을 선택하든 AWS는 사용자 경험과 클라우드 서비스의 높은 기준을 충족하도록 지원하고 확장 가능하며 성능이 높은 애플리케이션을 저렴하게 배포할 수 있도록 도와줍니다. 기능이 풍부한 웹 애플리케이션과 모바일 애플리케이션을 빠르고 쉽게 구축하고 출시하는 방법을 알아보거나 지금 AWS 도구 또는 서비스를 시작해 보세요.

[애플리케이션 현대화 여정 시작 >](#)

[AWS 전문가에게 문의하기 >](#)



웹 애플리케이션 및 모바일 앱 구축을 위한 추가 리소스 살펴보기

[AWS Amplify >](#)

[AWS AppSync >](#)

[AWS Amplify를 사용하여 풀 스택 Next.js 웹 앱 구축 및 배포 >](#)

[GraphQL를 사용하여 현대적 API 구축 >](#)

[서버리스 WebSocket, GraphQL, AWS AppSync를 사용하여 실시간 애플리케이션 경험 구축 >](#)

[AWS AppSync GraphQL 블로그에서 실용적인 팁 확인 >](#)

[프론트엔드 웹 및 모바일 블로그에서 최신 기능 릴리스 및 방법 가이드 보기 >](#)