



# Accelerate full-stack web and mobile application development

3 tips proven to help teams build  
modern apps faster



# Build web and mobile applications faster

Organizations want to innovate faster with modern web and mobile applications that delight customers and support employees. To do so, development teams must meet the ever-rising bar for advancing user experiences and complex cloud features—while delivering scalable, secure, and high-performance applications at a low cost.

When building modern web and mobile applications, ensure that your teams are set up to deliver scalable, full-stack applications quickly and successfully. Look for developer tools that close the gap between new and experienced developers—helping your teams launch applications quickly while leveraging extensible options that integrate with internal processes.

To accelerate your development processes, search for solutions that streamline access to dispersed, distributed data across workstreams. This will enable frontend teams to build data-driven features quickly and efficiently. Make architectural choices to support the ability of your application to fit today and flex to meet the needs of tomorrow. Whatever you decide, ensuring application performance and security remains your number one job.

In this guide, we share three tips to speed the development of web and mobile applications:

- 1 Modern tools and frameworks** – Empower your frontend teams with flexible, purpose-built tools and frameworks
- 2 Modern APIs** – Simplify access to distributed data sources with unified GraphQL and Pub/Sub APIs
- 3 Modern compute** – Reduce management issues and complexities with managed and serverless services



## TIP #1

# Empower your frontend teams with flexible, purpose-built tools and frameworks

## The situation

Frontend developers create a differentiating user experience for their applications using frontend programming languages like JavaScript, Swift, or Kotlin; popular frameworks like React, React Native, Flutter, or Angular; and tools such as integrated development environments (IDEs). They code the presentation layer of applications, access and integrate data through APIs, and build cloud-powered features, such as video or chat, with real-time and offline capabilities for their web or mobile applications.

While frontend developers may be experts in building user interfaces (UIs), they are typically not cloud experts. Building and connecting cloud functionality into a web or mobile application can be complex and time consuming, reducing the time frontend teams can use to focus on developing the user experience.

## The solution

Help your frontend teams be more productive with development tools and frameworks that reduce the amount of code needed to accomplish common use cases. Leverage frameworks with deep capabilities for the most complicated tasks—such as authentication and data access—and look for advanced features to handle caching, reconnection, data synchronization, and conflict resolution. Select tools that simplify development without sacrificing the flexibility to customize or extend code-based application features. Choosing tools that reduce the cloud learning curve results in quicker setups, faster workflows, and automated security—giving frontend teams the freedom to focus on the problems they’re solving for your customers.

# How AWS can help

**AWS Amplify** offers tools and services to help frontend developers quickly build web and native mobile applications that share a common Amazon Web Services (AWS)–powered backend. **AWS Amplify Libraries** simplify the integration of common use cases, like storing images and video in the cloud, into your application and integrate deeply with popular frontend frameworks, languages, and mobile platforms. Additionally, the Amplify command line interface (CLI) and admin UI can help developers quickly create an extensible AWS backend as needed. Unlike backend-as-a-service (BaaS) offerings, Amplify allows developers access to the breadth of over 200 AWS services as their application use cases evolve.

**AWS Amplify Studio** lets you easily build and ship complete web and mobile applications in hours outside of the **AWS Management Console**. With Amplify Studio, you can quickly build an application backend, create rich UI React components, and visually bind a UI to the backend in clicks. Developers can use the visual interface to define a data model, user authentication, and file storage without backend expertise.

Developers can also accelerate their frontend UI application development by selecting from dozens of popular React components, such as news feeds, contact forms, ecommerce cards, and primitives (buttons, text fields, alerts), and customize them to fit their style guide. With **AWS Amplify Studio Form Builder**, developers can save time by building beautiful React UI forms automatically generated as React code—either from their data model, any JSON object, or from scratch. Amplify Studio auto-generated forms are fully customizable in code, so your form adapts to your project—not the other way around.

Amplify Studio also integrates with Figma, streamlining the designer-developer workflow. Using a Figma design-to-code approach, developers can import designer-built Figma prototypes into Amplify Studio as clean React code, saving them the need to write boilerplate code. Amplify automatically generates the underlying business logic and deployment code so that developers can extend applications with custom business logic code in their preferred programming language.

Developers can also use **AWS Amplify Hosting** to deploy and host any React, Vue.js, or Next.js web applications. Seamlessly integrate existing tools and AWS services for maximum productivity by connecting Git workflows or uploading files to the Amplify console. Then, simply configure build settings and deploy directly from the Amplify console to the **Amazon CloudFront** content delivery network (CDN). This all-in-one solution allows you to continuously deploy a static web or server-side-rendered application, a mobile app landing page, or a progressive application on every code commit for fast, secure, and reliable applications that scale with your organization.

By enabling the above benefits, Amplify is helping frontend developers build feature-rich web and mobile applications faster than ever. By encoding best practices for common scenarios and delivering them in an opinionated way, Amplify reduces the overall steps and lines of code required to build full-stack applications on AWS—while providing dev teams with the flexibility to customize applications as business requirements evolve.

## CUSTOMER STORY

# Branch Insurance accelerates application development on AWS

## Challenge

Branch Insurance set out to radically simplify the end-user experience for insurance customers by offering bindable prices based on just two simple pieces of information: the customer's name and the customer's address. However, offering this simplicity required powerful infrastructure to process data quickly and store it efficiently and securely in compliance with regulations. By leveraging [AWS AppSync](#) as the foundation for its backend infrastructure and API service and connecting it to Amplify Libraries to manage the authorization flow, Branch Insurance was able to minimize time spent managing operations for its applications.

## Tools and services used

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

## Results

- Six-month acceleration in app development velocity
- Four products launched in just three years with a team of fewer than 20 developers

[Read the full case study ›](#)



BRANCH

**“Building a product on AWS is like doing it on ‘easy mode’ because there’s so much that’s simplified by using managed services. We just write business logic and interfaces. That’s the great benefit of using AWS.”**

Joe Emison,  
Co-Founder & CTO, Branch Insurance



# Simplify access to distributed data sources with unified GraphQL and Pub/Sub APIs

## The situation

Software architectures are continuously evolving. Organizations are moving away from holistic monoliths toward distributed architectures or sometimes using a combination of both. Which architecture to use depends on many factors, including skill sets and application or programming models. When organizations choose distributed architectures comprising independent microservices or components, they can increase development velocity because teams can work on different parts of the application independently. Distributed architectures also improve fault tolerance. Failures of certain components can be isolated to minimize impact to other application components.

With microservices, applications can be easier to scale and faster to develop, enabling innovation and accelerating time to market. However, adopting microservices may require frontend teams to connect to distributed microservices and data sources, making it more difficult to access the data they need for their user-facing applications. By streamlining data access in a way that makes it easier for frontend teams to consume in the application, backend teams play an important role in accelerating web and mobile development.

## The solution

Technologies such as GraphQL can increase the speed and efficiency of data access and integration across multiple backend data sources. GraphQL is a query language for APIs and a runtime for fulfilling those queries with your existing data. GraphQL provides a complete and understandable description of the data in your API, giving frontend developers the ability to query exactly the data that's needed on each API call from multiple backend databases and microservices. Backend development teams can use GraphQL to abstract existing data sources into a single endpoint and provide easier data access to frontend teams. For a set of existing APIs, abstracting multiple endpoints into a single GraphQL API delivers GraphQL advantages—without the work of migrating APIs or rewriting the application from scratch.

Creating a unified API layer with GraphQL complements your REST API strategy to help frontend developers simplify their interactions with distributed service architectures. GraphQL users can also publish real-time data updates from any event source to subscribed web, mobile, and Internet of Things (IoT) clients by extending the API to include subscriptions.

# How AWS can help

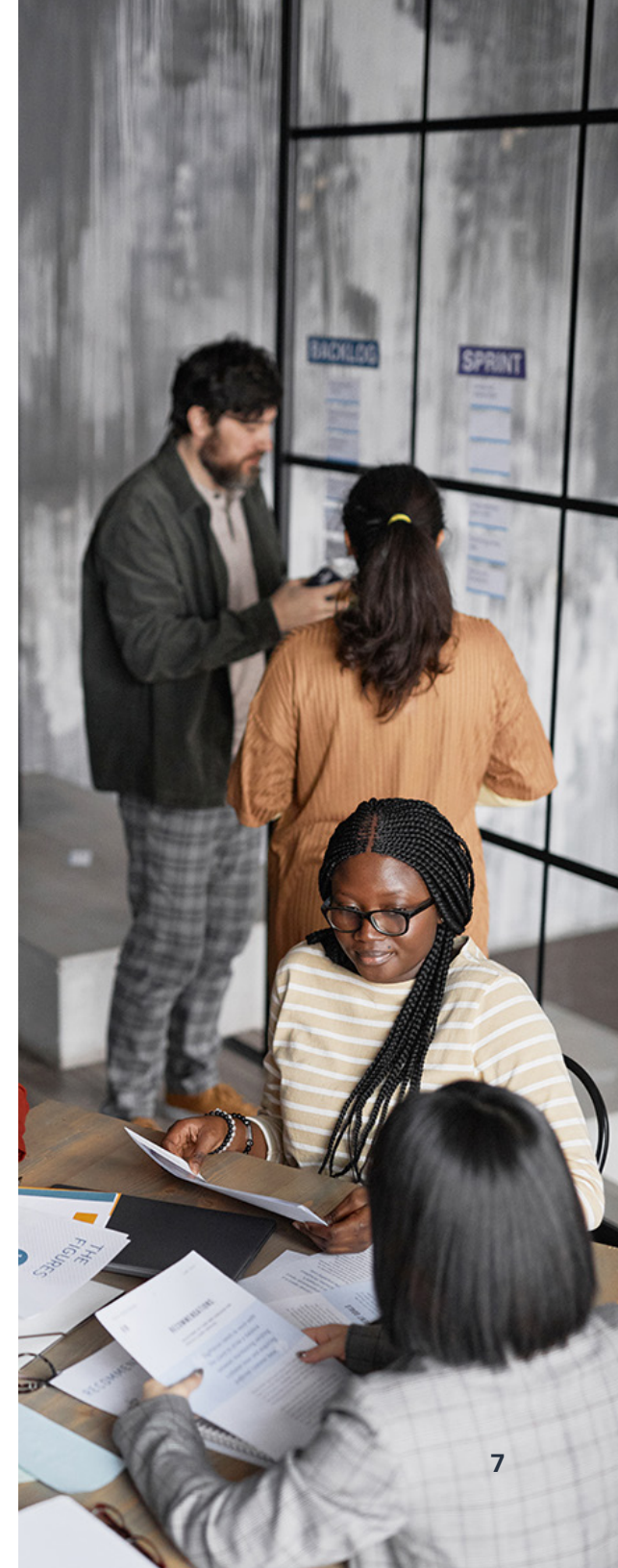
AWS AppSync simplifies the development of modern web and mobile applications by allowing developers to connect their apps to data and events with secure, serverless, and performant GraphQL and Pub/Sub APIs. AWS AppSync handles the heavy lifting involved in securely connecting to data sources like [Amazon DynamoDB](#), [Amazon EventBridge](#), and [AWS Lambda](#). AWS AppSync offers fully managed GraphQL API and Pub/Sub API setup, administration, auto scaling and high availability with built-in security, monitoring, logging and tracing, and optional caching for low latency. AWS AppSync now also offers [Private APIs](#) that allow users to set restrictions such that their APIs can only be accessed from specified virtual private cloud (VPC) endpoints.

AWS AppSync GraphQL APIs simplify application development by providing a single endpoint to securely query or update data from multiple databases, microservices, and APIs. Use GraphQL to power your applications with the right data from one or more data sources with a single network request and combine multiple data sources into a single GraphQL API.

AWS AppSync Pub/Sub APIs simplify the creation of engaging real-time experiences by automatically publishing data updates to subscribed API clients via serverless WebSocket connections. Publish real-time data updates from any event source to subscribed web, mobile, and IoT clients by extending your GraphQL API to include subscriptions. Or start with a simple Pub/Sub API without GraphQL. Either way, AWS AppSync handles the creation of serverless WebSocket connections, message broadcasts, and fanout to up to millions of connected clients.

AWS AppSync offers integrations with other AWS services, making it easy to secure, monitor, and troubleshoot APIs. These services include:

1. [AWS CloudTrail](#) for monitoring audit logs in real time
2. [Amazon CloudWatch](#) for managing metrics and logs
3. [AWS X-Ray](#) for tracing the behavior of distributed applications
4. [AWS WAF](#) to protect APIs from common web exploits



## CUSTOMER STORY

# Evolving ADP's single global experience in MyADP and ADP mobile applications

## Challenge

Automatic Data Processing (ADP) wanted to modernize its flagship desktop and mobile solutions, MyADP and ADP Mobile, to create a seamless user experience (UX) for 17 million users. To achieve this, ADP committed to consolidating, standardizing, and modernizing its application.

ADP had to innovate to create a single experience for disparate systems of record without introducing error. "The speed at which pay statements open up should be the same speed at which benefits enrollment open, but these are two different sources of content on two different sets of infrastructure," says Devi Ramachandran, senior director of DevOps at ADP. "That's been our challenge from the beginning, and migrating our systems to AWS made everything simpler." To simplify application API access provided by those different infrastructures, ADP used AWS AppSync to bring together data from the various backends and sources into a single endpoint. As a result, ADP was able to streamline data aggregation on the backend and create a better experience for users.

## Tools and services used

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

## Results

- Maintained 4.5+ app store rating
- Achieved portability for global UX
- Improved resiliency through multi-region architecture
- Reduced latency with latency-based routing
- Scaled for bursts of traffic to eliminate throttling and errors

[Read the full case study ›](#)



**"We're using AWS because we want to be a product development team and not an infrastructure management team."**

Devi Ramachandran,  
Senior Director, DevOps, ADP



# Reduce management issues and complexities with managed and serverless services

## The situation

Deloitte reports that 80 percent of IT resources are focused on running the business, while 20 percent are focused on innovation. The fastest way to improve that ratio and bias it toward innovation is to reduce time spent on activities that are not core to your business—such as managing infrastructure. Managing servers, storage, networking, and operating systems is time consuming, expensive, and operationally complex due to procurement cycles. Managing infrastructure consumes more and more of your teams' time (and represents a critical opportunity to optimize and reinvest in innovation).

## The solution

One of the easiest ways to “operate less” is by offloading the operational burden to a cloud provider by adopting managed or serverless services for your infrastructure. Reducing time spent provisioning, managing, and patching servers enables your developers to spend more time innovating—writing and iterating code for the products that delight your customers. With serverless computing, there's no infrastructure to provision or manage. Your infrastructure scales automatically by unit of consumption—rather than server unit. With serverless, your organization pays only for the compute it consumes; availability and fault tolerance are built in. Serverless computing removes the burden of server operations, so you can focus on building the application rather than managing and scaling the infrastructure to support the application.

# How AWS can help

A great place to start is **AWS Lambda**. This serverless compute service runs code as a highly available, scalable, secure, fault-tolerant application while completely abstracting the underlying compute environment. This allows you to focus primarily on application development, speeding time to market, and lowering the total cost of ownership (TCO). Lambda functions can be triggered by events, which means that it sits idle until an event is triggered, and then the system powers into action. In fact, you can automatically trigger Lambda functions from over 200 AWS services, HTTP calls, and software-as-a-service (SaaS) applications—without writing a single line of integration code.

Once the event is triggered, resources are automatically spun up and down to meet the need of the task—at millisecond speed. There is no need to provision or configure infrastructure, and you pay only for what you use. This means faster time to production at the lowest possible TCO.

Lambda offers maximum agility with minimal operations, but in some cases, you may want to use a container orchestrator to run containers. This gives you the flexibility to run serverless or fine-tune your infrastructure to add value to your business. Your teams can complement **AWS Lambda** with **Amazon Elastic Container Service (Amazon ECS)** on **AWS Fargate**, a serverless container service.



## CUSTOMER STORY

# ProSiebenSat.1 Media SE delivers interactive TV experiences using AWS serverless solutions

## Challenge

Seven.One Entertainment Group, a wholly owned subsidiary of innovative digital group ProSiebenSat.1 Media SE, had an opportunity to build a highly engaging, interactive television solution to better delight and engage its customers. The company decided to migrate its on-premises viewer engagement solution to AWS using a serverless architecture to handle the peaks of event-based traffic.

Using managed services like AWS AppSync, Seven.One Entertainment Group can develop GraphQL APIs by offloading the heavy lifting of securely connecting to DynamoDB data at scale. Their serverless architecture, powered by the event-driven compute of Lambda, also scales to support viewer interaction during events. It can deliver results in near real time to millions of fans using AWS AppSync real-time subscriptions.

As a result of using AWS, the company reduced the monthly costs of its solution by 60 percent and empowered employees to focus on differentiating tasks that create exciting new features rather than spend time self-managing infrastructure.

## Tools and services used

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

## Results

- Reduced monthly costs of its solution by 60 percent
- Achieved 13 ms latency during peak hours
- Decreased the time for distributing video content
- Made an API available to end-user traffic through increased scalability

[Read the full case study ›](#)



**“Using AWS, we can focus on what we do best, which is writing code. We don’t have to worry about patching up operating systems or scaling.”**

**Fabian Desoye,**  
Head of Metadata Systems, Seven.One  
Entertainment Group

# Getting started

Now that you've explored our three tips for accelerating the development of modern web and mobile applications, you're probably anxious to get started. We advise teams to begin with one or more of these tips to complement their current approach to building modern applications.

Empower your frontend teams with Amplify and reduce the overall steps and lines of code required to build full-stack applications to help developers launch faster. You can also simplify access to distributed data sources with AWS AppSync. By abstracting existing data sources into a unified GraphQL API endpoint, you will provide easier data access to frontend teams to build data-driven features. You can even differentiate your application with real-time data experiences using a simple Pub/Sub API. And AWS helps you operate less infrastructure using services like Lambda.

Whichever path you choose, AWS will help you meet the ever-rising bar for user experience and cloud services—allowing your teams to deliver a scalable, high-performance application at low cost. Discover how your organization can quickly and easily build and launch feature-rich web and mobile applications. Or, jump in and get started today with one or more of our tools or services.

[Start your application modernization journey ›](#)

[Connect with an AWS expert ›](#)



## Explore more resources for building web and mobile applications

[AWS Amplify ›](#)

[AWS AppSync ›](#)

[Build and deploy a full-stack Next.js web app with AWS Amplify ›](#)

[Build modern APIs with GraphQL ›](#)

[Build real-time application experiences with serverless WebSockets, GraphQL, and AWS AppSync ›](#)

[Gain practical tips in the AWS AppSync GraphQL blog ›](#)

[Get the latest feature releases and how-to guides on the frontend web and mobile blog ›](#)