# AWS SERVERLESS APPLICATIONS: THE BEGINNER'S GUIDE

Serverless applications are changing the way companies do business by enabling them to deploy much faster and more frequently. Amazon's AWS Serverless Applications Model (AWS SAM) has been a game changer in this space, making it easy for developers to create, access and deploy applications, thanks to simplified templates and code samples.

## What You'll Learn in This Guide

This guide serves as an excellent primer for developers who are keen to learn how to program in AWS SAM. After a brief introduction of the Model, we will review each of the following key resources:

- AWS Lambda
- Amazon API Gateway
- Amazon DynamoDB
- AWS Repository

But first, let's refresh ourselves on these key terms which will be referenced throughout the following text:

- **Function-as-a-Service (Faas)**: Think of FaaS as a ready-to-implement framework that can be easily tailored to the needs of an enterprise business. In the realm of pre-packaged services, Function-as-a-Service is also sometimes known as Framework-as-a-Service or FaaS, falls in between Software-as-a-Service and Platform-as-a-Service. You can learn more about FaaS here.
- **Compute service**: An on-demand FaaS that specializes in serverless computing.

- **Serverless application**: An application, programmed in the cloud, that requires no server maintenance.

## Who uses AWS SAM?

AWS SAM is an important resource for any developer who is primed in serverless computing, or who wants to learn more about serverless architecture. In fact, the resources available within the Model make it easy for an entry-level programmer to get their feet wet with low-cost, efficient serverless computing services provided by Amazon.

## The AWS Serverless Application Model

The AWS Serverless Application Model uses CloudFormation technology to provide a seamless interface for the tools you need to build, access and run serverless applications.

Overall, the goal of AWS SAM is to make the creation, deployment and execution of serverless applications as simple as possible. Previous iterations lacked the capacity to define all resources of the application. However, today, this can be done using AWS SAM templates with just a few choice code snippets.

Now let's take a look at the serverless resources that make up the AWS SAM:

## AWS Lambda

The AWS Lambda Function-as-a-Service (FaaS) platform exists so that developers can run code without any server administration. Using the Lambda compute service, developers can simply upload code to deploy and Amazon handles the administration, charging only for runtime accumulated.

Supported languages include: Node.js, Java, C#, Go and Python.

One important distinction about Lambda is the ability to create both a single, simple, scalable function with a runtime trigger defined in Lambda, or something as robust as your own application backend. Of note, functions created on Lamba's architecture are scalable for optimum performance.

The model is also an open source platform where code and other serverless applications can easily be shared between developers. In addition, Lambda's console makes it easy to create applications based on Lambda's model with just a few clicks.

## Amazon API Gateway

Using the Amazon API Gateway service, developers can create, deploy, secure and monitor the frontend of their serverless application.

Like AWS Lambda, which is designed to create the backend of an application, Amazon API Gateway takes the complexity out of writing and deploying code that executes the front-door entry point to your application. The Gateway APIs work with AWS Lambda backends, as well as Amazon EC2 and other web application services.

Demystifying this process means having streamlined functions for each of the following:

- Running many versions of the same API for testing purposes

- Securing the frontend of your application with authorizations
- Monitoring API use
- Scalability

Again, all this is accomplished on behalf of the developer without server administration.

In addition, Amazon keeps the service low-cost and affordable for developers by only charging for calls made to APIs and data transfers out of them.

## Amazon DynamoDB

Similar to the above-mentioned applications, Amazon DynamoDB is a fully-managed, fully scalable service. It comes in the form of a NoSQL database that is 100% integrated in the cloud.

It offers developers convenience and ease of use through some of the following key benefits:

- Optimized for scalability
- Low-maintenance
- No server administration required
- Easy to use

Using Amazon DynamoDB, developers can make resilient, low-latency databases that are optimized for scaling to meet the needs of any business. Its features make it ideal for mobile, web, gaming, ad tech, IoT and many more applications.

## AWS Repository

The most recent addition to the Model, AWS Repository is a searchable ecosystem that allows developers to find serverless applications and their components for deployment. Previously only available in public preview, as of February 2018 the Repository is now part of the Lambda console.

What's great about the AWS Repository is its mission to further simplify serverless application development. Here are the basic steps:

1. *Search and Discover*: A developer can search the repository for code snippets, functions, serverless applications and their components.
2. *Configure*: They can then set environment variables, parameter values and more before deploying. For example, they can go the plug-and-play route by adding them to a larger application framework, or take them apart and tinker to further customize them. In addition, if need be, pull requests can also be submitted to authors.
3. *Deploy*: Once an application is deployed, it can be managed from the AWS Management Console. A developer can follow prompts to name, describe and upload their serverless applications and components to the ecosystem where they can be shared internally and with other developers across the ecosystem. This feature makes AWS SAM a truly open source environment.

# What are the Benefits of Programming in AWS SAM?

You can build serverless applications for almost any type of backend service without having to worry about scalability and mull over managing servers. Let's take a closer look at the many benefits

that building serverless applications in AWS SAM has to offer:

### Low Cost and Efficient

AWS SAM is low-cost and efficient for developers because of its pay-as-you-go structure. The features of the platform only charge developers for usage, meaning you never pay for more of a service than you use.

### Simplified Processes

The overarching goal of AWS SAM is ease-of-use. By design, many of the updates made over the years since its original release are focused on simplifying application development so that programmers have more freedom to create in the open source ecosystem.

### Quick, Scalable Deployment

AWS SAM makes deployment quick and simple by allowing developers to upload code to AWS and letting Amazon handle the rest. They also provide robust testing environments so developers don't miss a beat. All of this occurs on a platform that is easy to scale, allowing apps to grow and change to meet business objectives.

### Convenient and Accessible

Undoubtedly, AWS SAM offers a convenient solution for developing in the cloud. But it's serverless nature also means that it is a highly accessible platform. The wide reach of the internet makes it easy to execute code on-demand from anywhere.

### Decreased Time to Market

Overall, choosing a serverless application platform saves time and money that would otherwise be spent managing and operating servers or runtimes, whether on-premises or in the cloud. Because developers can create apps in a fraction of the time (think hours not weeks or months), they are able to focus more of their attention on accelerating innovation in today's competitive digital economy.

## AWS SAM for Serverless Applications

It's clear that AWS SAM is a highly efficient, highly scalable, low-cost and convenient solution for developers who are interested in programming in the cloud.

Through several iterations of AWS SAM functions, like CloudFormation and early versions of Lambda, Amazon has come closer and closer to perfecting a serverless solution for developers. And the recent addition of the AWS Repository has only helped to sweeten the deal.

But for those who haven't yet made the switch, the allure around the AWS Repository update may not be enough to silence concerns that arise from:

- A general lack of control over the ecosystem that developers are coding in
- Vendor lock-in that may occur when you sign up for any FaaS
- Session timeouts that require developers to rewrite code, making it more complex instead of simplifying the process

The first two points are common drawbacks of any outsourcing strategy while the last may necessitate the implementation of a few new workarounds. Regardless, the many benefits of programming in AWS SAM cannot be overlooked.

If you are looking to rapidly deploy applications that will help the grow your business and deliver better products and services, BMC can help.

BMC also offers hands-on application development, cloud support and automation assistance to customers with our Application Deployment and DevOps solutions.