Category:

Serverless

AWS Serverless
Application Model
(SAM)

Created by:

Ashish Prajapati

What?
- The AWS Serverless Application Model (AWS SAM) is an open-source framework that you can use to build serverless applications on AWS. AWS SAM consists of two components - AWS SAM template specification (to define your serverless application) and AWS SAM CLI (a command line tool that operates on AWS SAM templates to build serverless applications).

Why?
- AWS SAM makes it easy to organize related components and resources, and operate on a single stack.
- You can use AWS SAM to define and deploy your infrastructure as config.
- It provides shorthand syntax to express functions, APIs, databases, and event source mappings.

When?
- You need a shorthand syntax to describe your serverless application (Lambda functions, API endpoints, DynamoDB tables, and other resources) using a simple YAML template with just a few lines per resource. You need a Lambda-like local execution environment to help you catch issues by providing parity with the actual Lambda execution environment.

Where?
- AWS SAM is available in all regions where AWS Lambda is available.
- You can install AWS SAM CLI on any supported Linux, Mac, or Windows platform using pip. It is included with AWS Cloud9. You can also use SAM CLI to locally debug Lambda functions written in Node.js, Java, Python, and Go.

Who?
- You construct an AWS SAM template to declare and configure the components of your application.
- AWS SAM templates are an extension of AWS CloudFormation templates, so any resource that you can declare in an AWS CloudFormation template can also be declared in an AWS SAM template.

How?
- Using AWS SAM CLI you initialize and configure applications, debug locally using IDEs like Visual Studio Code or JetBrains WebStorm, and deploy to the AWS Cloud. During deployment, SAM transforms and expands the shorthand SAM syntax into an AWS CloudFormation template and then, CloudFormation provisions your resources.

How much?
- There is no additional charge to use AWS SAM. You pay for the AWS resources created using SAM in the same manner as if you created them manually.