**AWS SAM**

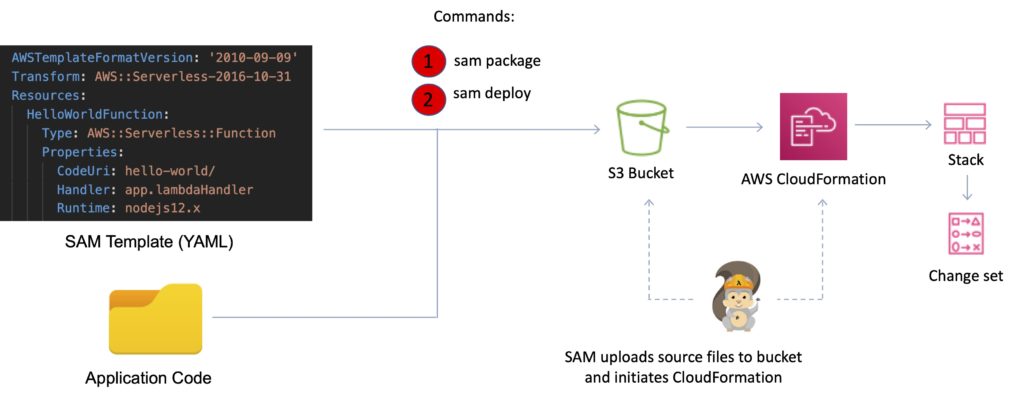
The AWS Serverless Application Model (AWS SAM) is an open-source framework for building serverless applications.

It provides shorthand syntax to express functions, APIs, databases, and event source mappings.

With just a few lines per resource, you can define the application you want and model it using YAML.

During deployment, SAM transforms and expands the SAM syntax into AWS CloudFormation syntax, enabling you to build serverless applications faster.

AWS SAM supports all AWS CloudFormation template items such as Outputs, Mappings, Parameters, etc.



To get started with building AWS SAM-based applications, use the AWS SAM CLI. SAM CLI provides a Lambda-like execution environment that lets you locally build, test, and debug applications defined by SAM templates.

You can also use the SAM CLI to deploy your applications to AWS.

Key benefits of SAM

Single deployment configuration:

* You can use SAM to organize related components, share configuration such as memory and timeouts between resources, and deploy all related resources together as a single, versioned entity.

Integration with development tools

SAM integrates with a suite of AWS serverless tools.

You can find new applications in the [**AWS Serverless Application Repository**](https://serverlessrepo.aws.amazon.com/applications)

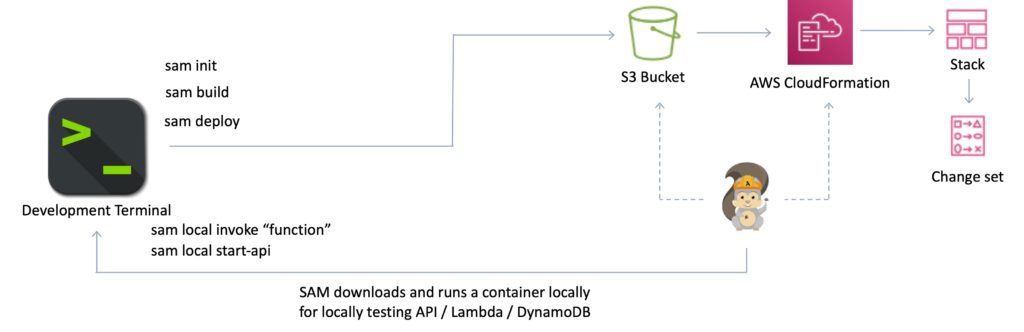
The [**AWS Cloud9**](https://aws.amazon.com/cloud9/) IDE can be used to author, test, and debug SAM-based serverless applications,

[**AWS CodeBuild**](https://aws.amazon.com/codebuild/), [**AWS CodeDeploy**](https://aws.amazon.com/codedeploy/), and [**AWS CodePipeline**](https://aws.amazon.com/codepipeline/) can be used to build a deployment pipeline.

Local testing and debugging

You can use SAM CLI to step-through and debug your code.

SAM CLI provides a Lambda-like execution environment locally and helps you catch issues upfront.



Built on AWS CloudFormation

AWS SAM is an extension of AWS CloudFormation, so you get the reliable deployment capabilities of CloudFormation.

You can also define resources using CloudFormation in your SAM template and use the full suite of resources, intrinsic functions, and other template features that are available in AWS CloudFormation.

Built-in best practices

You can deploy your infrastructure as config to leverage best practices such as code reviews.

You can enable gradual deployments through AWS CodeDeploy and tracing using AWS X-Ray with just a few lines of SAM config.

Deployment Details

A SAM template file is a YAML configuration that represents the architecture of a serverless application.

You use the template to declare all of the AWS resources that comprise your serverless application in one place.

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.

All configuration code is in YAML.

The “Transform” header indicates it’s a SAM template: Transform: ‘AWS::Serverless-2016-10-31’

There are several resource types:

* AWS::Serverless::Function (AWS Lambda)
* AWS::Serverless::Api (API Gateway)
* AWS::Serverless::SimpleTable (DynamoDB)
* AWS::Serverless::Application (AWS Serverless Application Repository)
* AWS::Serverless::HttpApi (API Gateway HTTP API)
* AWS::Serverless::LayerVersion (Lambda layers)

Only two commands are required to package and deploy to AWS.

Use either:

sam package

sam deploy

Or use:

aws cloudformation package

aws cloudformation deploy

SAM Policy Templates are a list of templates to apply permissions to your Lambda functions.

Examples:

* S3ReadPolicy.
* SQSPollerPolicy.
* DynamoDBCrudPolicy.

Serverless Application Repository

The AWS Serverless Application Repository is a managed repository for serverless applications.

It enables teams, organizations, and individual developers to store and share reusable applications, and easily assemble and deploy serverless architectures in powerful new ways.

Using the Serverless Application Repository, you don’t need to clone, build, package, or publish source code to AWS before deploying it.

Instead, you can use pre-built applications from the Serverless Application Repository in your serverless architectures, helping you and your teams reduce duplicated work, ensure organizational best practices, and get to market faster.

Integration with AWS Identity and Access Management (IAM) provides resource-level control of each application, enabling you to publicly share applications with everyone or privately share them with specific AWS accounts.

You can browse all applications in the serverless application repository here:

<https://serverlessrepo.aws.amazon.com/applications>