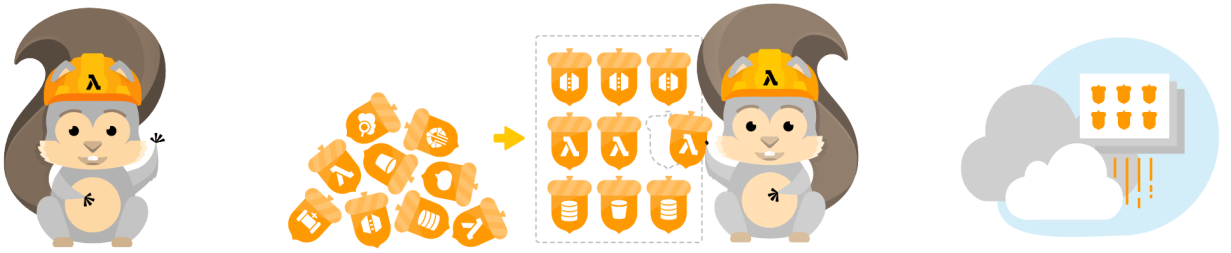
AWS Serverless Application Model: Complete Solution For Serverless Apps

What is Serverless And Why Should You Care?

As a quick aside, you would possibly want to understand why people care about serverless architecture. First, let's define serverless.
The most simple definition of serverless architecture is an application that is composed of **functions triggered** by **events** using a service like AWS Lambda to handle the loading and execution of the application code as well as the provisioning of sufficient computing resources.
The primary benefits of serverless architectures include:

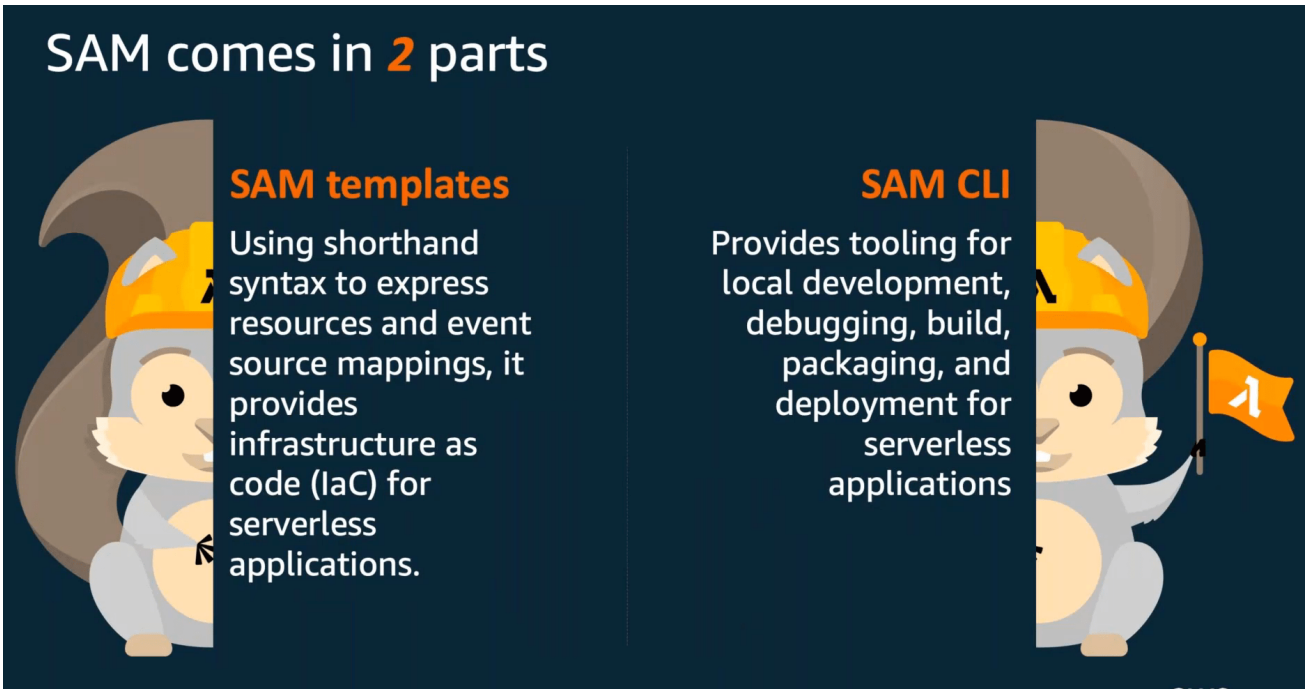
- The ability for developers to focus purely on application functionality and code
- A linear relationship between the efficiency of an application's code and therefore the cost of running it
- A code execution system that only runs when needed
- A significant advantage when running event-driven apps that need only short-duration, single-event transactions

What is the Serverless Application Model?



The **AWS Serverless application model (SAM)** is a framework that's used to build serverless applications. SAM establishes patterns for development, keeping all of the code for a serverless application within an equivalent repository structure. AWS SAM uses a mixture of configuration files, pattern models, and command-line tools to ease the process of serverless development. this enables developers to centralize their efforts, reducing code duplication and complexity through a strong build and deployment system.
This framework accommodates 2 components:

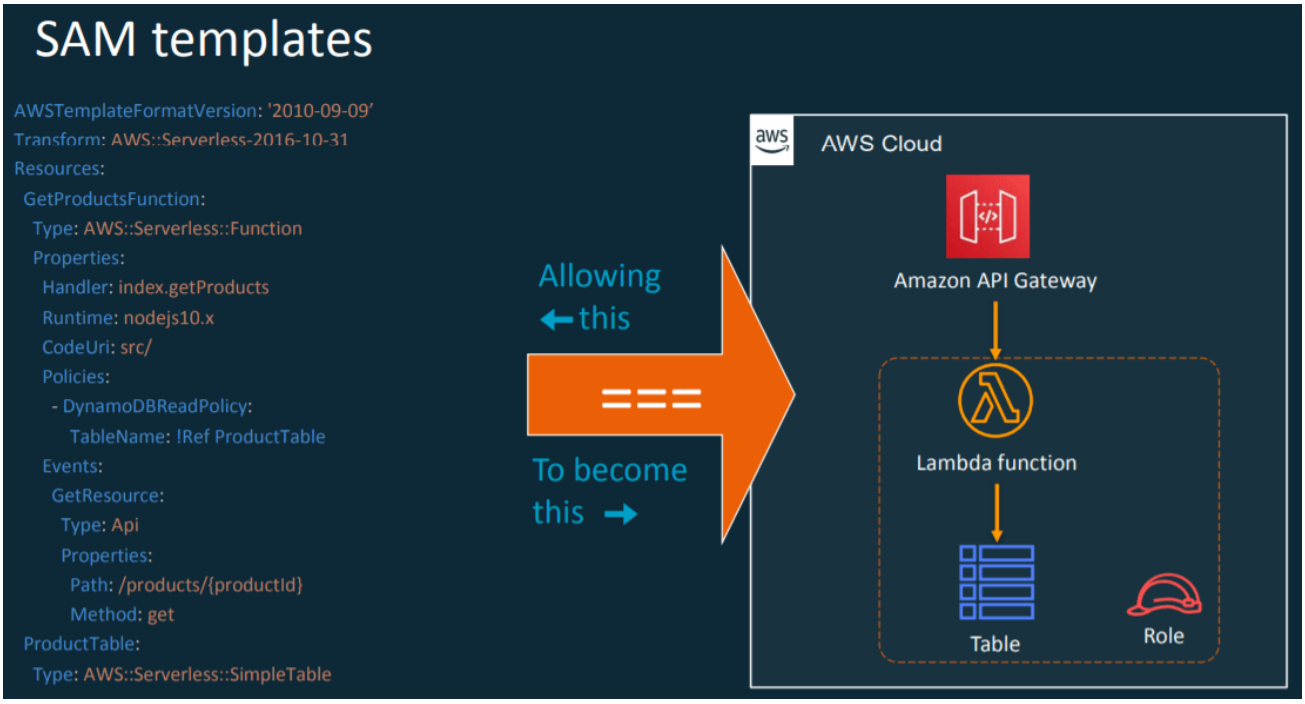
- **SAM template specification**
- **SAM command-line interface**



AWS SAM template specification

SAM Template specification is used to define serverless applications. It provides an easy syntax to explain the functions, APIs, permissions, configurations, and events that structure a serverless application. AWS SAM template file is a single, deployable, versioned entity for your serverless application.

- Can mix in other non-SAM CloudFormation resources within the
- same template i.e. Amazon S3, Amazon Kinesis, AWS Step Functions
- Supports use of Parameters, Mappings, Outputs, etc
- Supports Intrinsic Functions ie: Ref, Sub, Join, Select, Split
- Can use ImportValue (exceptions for RestApiId, Policies, StageName attributes)
- YAML or JSON



Go through this AWS Blog to get a clear understanding of **AWS Organizations**

AWS SAM Command Line Interface (CLI)

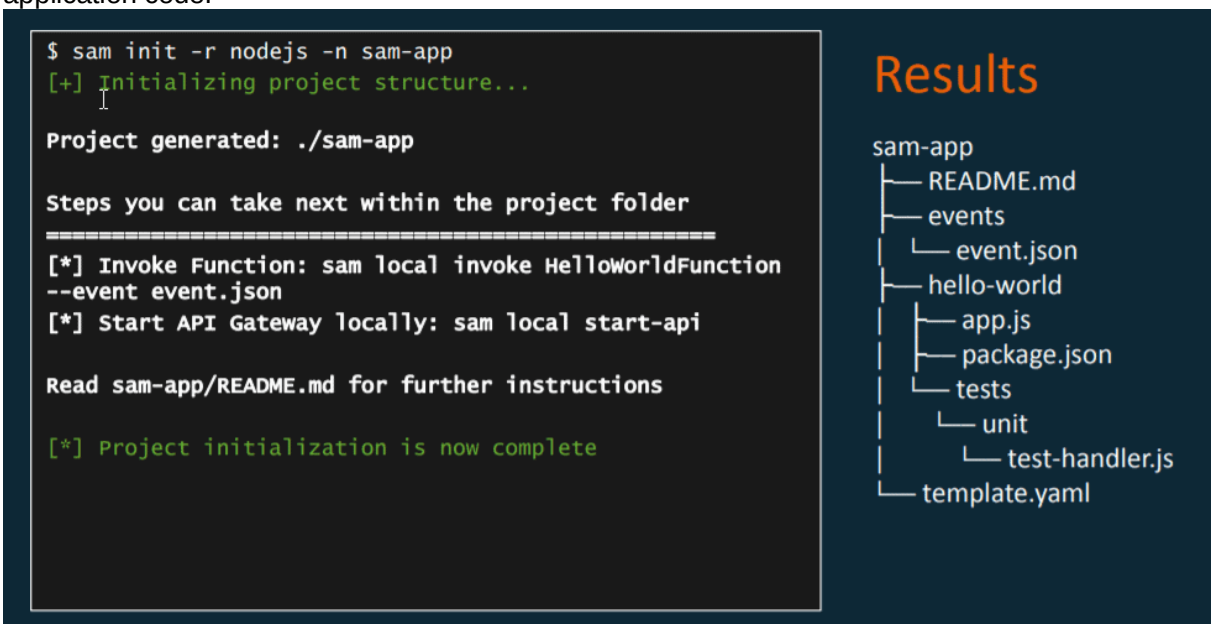
The SAM CLI is used to create serverless applications that are defined by AWS SAM templates.

- SAM CLI provides commands that verify and make sure that AWS SAM template files are written consistent with the specification.
- SAM CLI provides Lambda-like execution environments locally in order that you'll invoke Lambda functions locally, and step-through debugs them.
- It provides the commands to package and deploy serverless applications to the AWS cloud, and so on.

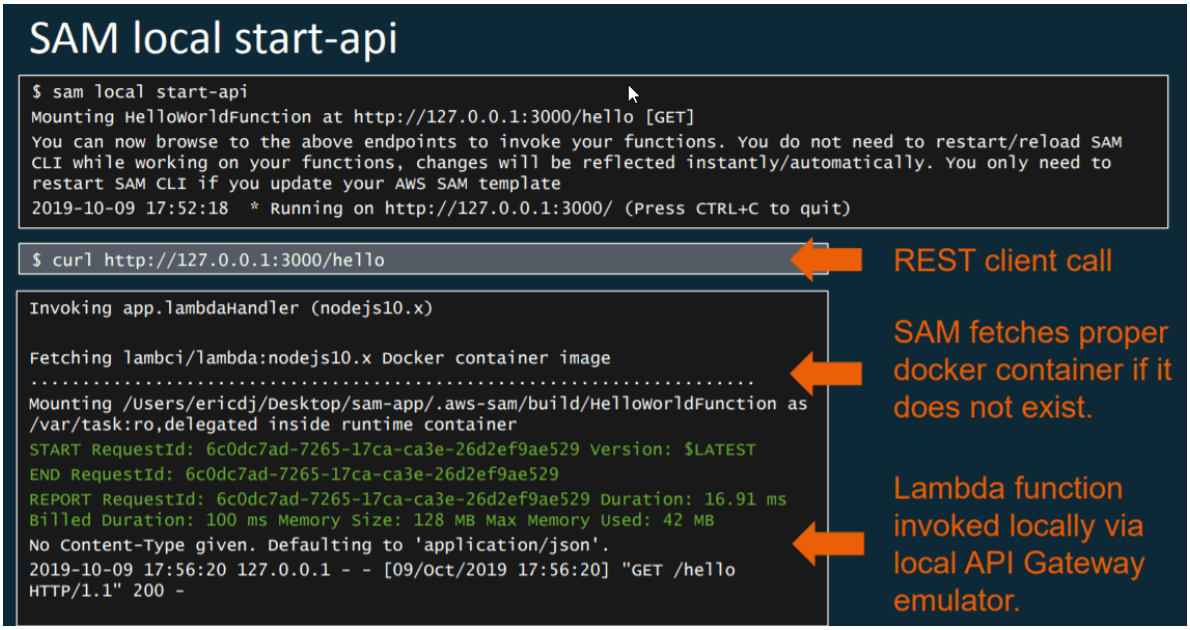
Following are the foremost used commands within the SAM CLI:

- **sam init** – For the first-time user, this command without any parameters creates a Hello World application. you'll select the language (e.g., NodeJS or Python) during which you would like to get the code. The command generates a preconfigured AWS SAM template and example

application code.

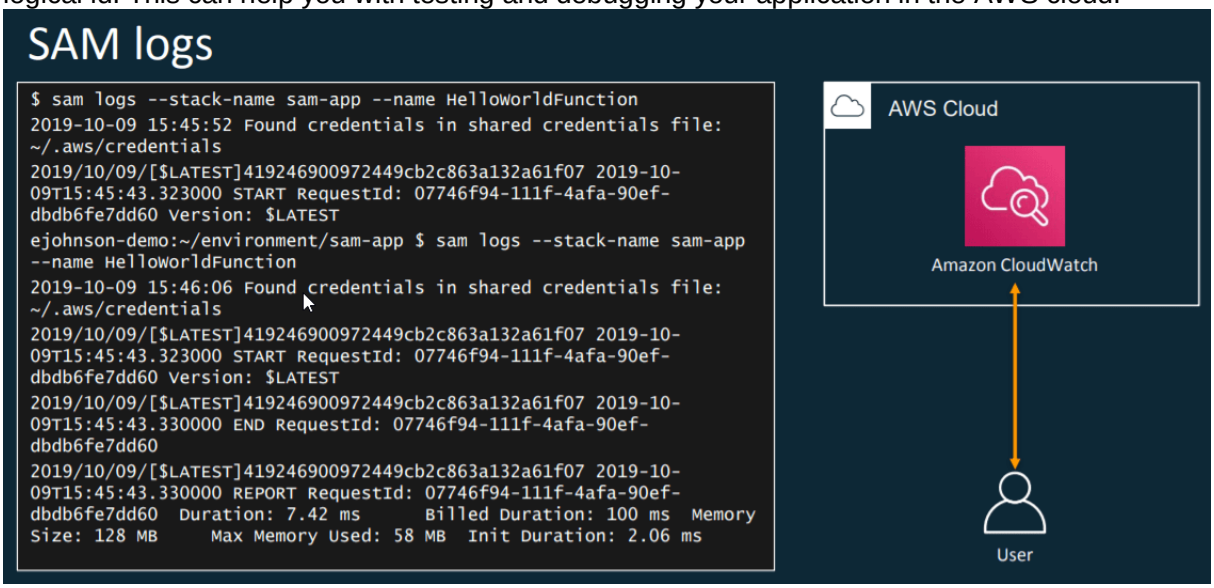


- **sam local invoke and sam local start-api** – These commands create Lambda-like execution environments in local to enable the local testing and debugging, before deploying it to the cloud.

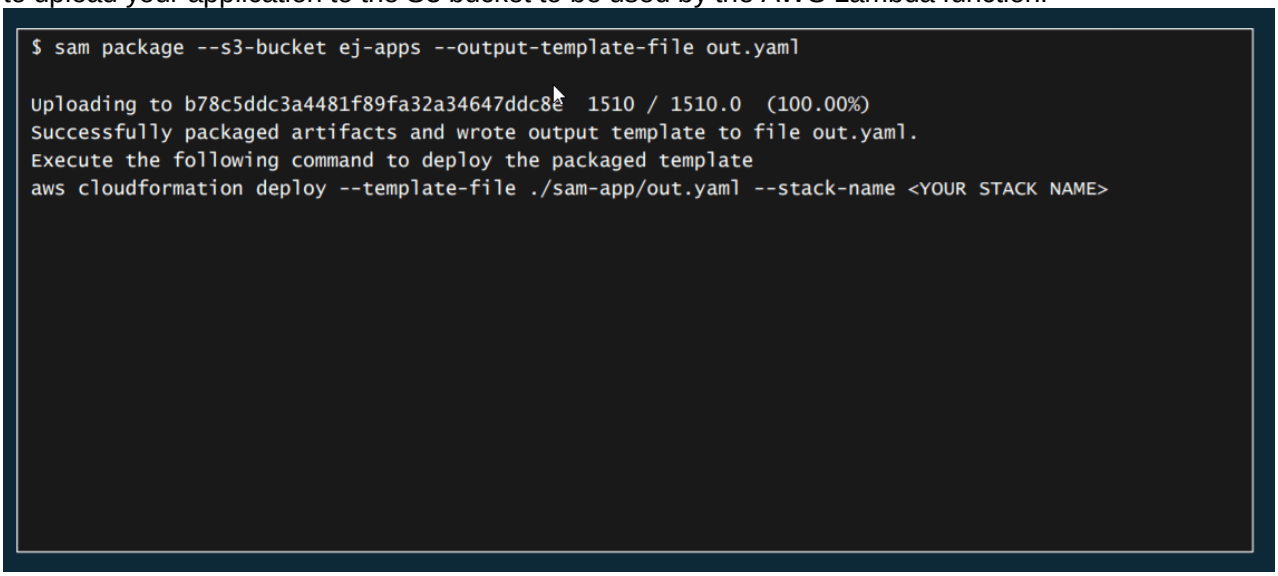


- **sam logs** – As functions are part of an AWS CloudFormation stack, you can use this command to fetch logs generated by your AWS Lambda function based on the function's

logical id. This can help you with testing and debugging your application in the AWS cloud.

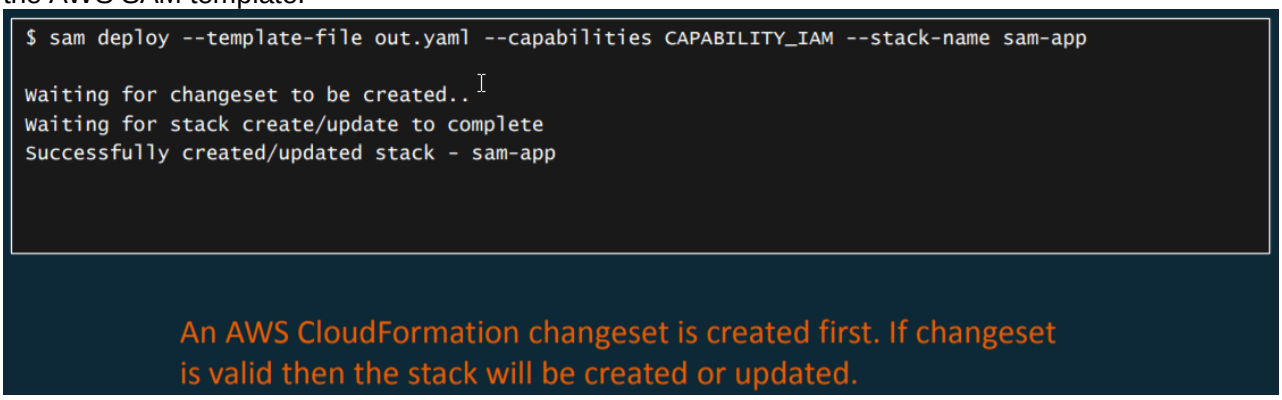


- **sam package** – This command is used to bundle your application code and dependencies into a zip or jar file also known as “deployment package”. The deployment package is required to upload your application to the S3 bucket to be used by the AWS Lambda function.



- **sam deploy** – This command is used to deploy a serverless application to the AWS cloud. It creates the AWS resources and sets permissions and other configurations that are defined in

the AWS SAM template.



Benefits of AWS SAM

AWS SAM not only creates Lambda resources but many other AWS resources through the template configuration. it's several advantages over the essential CloudFormation templates:

- **Single-Deployment Configuration**
  AWS SAM brings together all the related components and resources and operates on one CloudFormation stack. it's deployed as a single versioned entity that shares configuration (such as memory and timeouts) between resources and deploys all related resources together.
- **Extension of AWS CloudFormation**
  AWS SAM is an extension of AWS CloudFormation. It can use a full suite of resources, intrinsic functions, and other template features available in AWS CloudFormation. In AWS SAM templates the Resources section can contain both AWS CloudFormation resources and AWS SAM resources. It brings more brevity and fewer configuration compared to CloudFormation for creating serverless applications.
- **Local Debugging and Testing**
  The SAM CLI provides a Lambda-like execution environment locally. The serverless application defined by SAM templates is deployed to the present environment and may be tested and debugged locally. this permits the simulation of an AWS Lambda environment locally and ensures the code will run on the cloud without issues. It reduces the value of testing and debugging code within the cloud. AWS has provided the AWS Toolkit for various IDEs, like Visual Studio, IntelliJ, JetBrains, and others. This accelerates the feedback loop by making it possible to run and troubleshoot issues within the local that you simply might face within the cloud.
- **Deep Integration with Development Tools**
  AWS SAM is often used with several other AWS tools for building serverless applications. to get new applications, you'll attend the AWS Serverless Application Repository. The AWS Cloud9 IDE is often used for coding, testing, and debugging AWS SAM–based serverless applications. CodeBuild, CodeDeploy, and CodePipeline are used for continuous integration and deployment.