

Steps to Deploy SAM Applications

In this lesson, you will learn about the steps required to deploy SAM applications.

WE'LL COVER THE FOLLOWING ^

- Steps for deployment

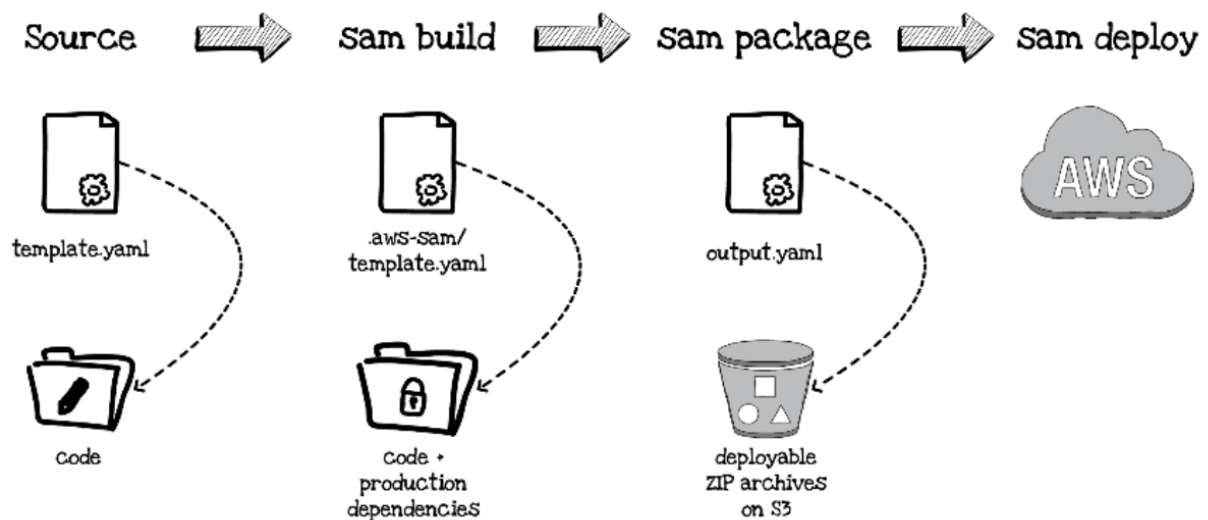
CloudFormation is amazingly powerful. It can deploy and configure almost any resource type available in the Amazon cloud, safely upgrading and downgrading entire networks of services. However, in order to achieve such flexibility, CloudFormation also needs to be very generic. It does not know about the structure of JavaScript, Python, or Java projects, and it does not understand how to interact with programming language packages or dependency managers. In order to create a Lambda function, CloudFormation expects a fully self-contained ZIP archive. That archive needs to contain all the required files for the Lambda function, including the source or compiled code, third-party dependencies, and native binary packages. It also needs to be somewhere on Amazon S3 where CloudFormation can read it.

The complicated deployment process is the reason why there are so many packaging tools for Lambda functions, including the AWS SAM command-line tools. SAM understands the typical project layouts for various programming languages and knows how to interact with platform package managers.

Steps for deployment

In the example project, the `CodeUri` parameter for a Lambda function points to a local directory with source code files. This is quite usual for teamwork and storing in a version control system, and you will use this approach in almost all projects. But you cannot send that template directly to AWS. In general, turning a SAM application on your disk into resources running in AWS requires three steps:

1. **Build**: create a clean copy of all Lambda functions, remove test and development resources, and download third-party dependencies.
2. **Package**: bundle each function into a self-contained ZIP archive, upload it to S3, and produce a copy of the source application template that points to remote resources instead of local directories.
3. **Deploy**: upload the packaged template to CloudFormation, and execute the changes to create a running infrastructure.



Deploying with SAM: 'sam build' creates a local self-contained copy, the 'sam package' function packages to S3, and 'sam deploy' creates a stack using CloudFormation.

In the next lesson, you'll build the application. See you there!