

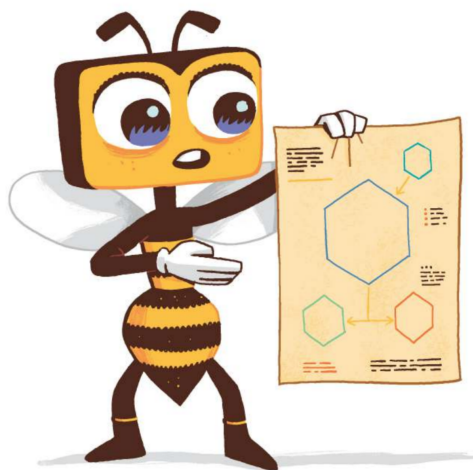
Initialising the Application

In this lesson, you will learn how to initialize the web service project using AWS SAM.

WE'LL COVER THE FOLLOWING ^

- The SAM command-line tool
 - The `sam init` command
 - The `--runtime` argument
 - The `--name` argument
 - The `--app-template` argument

This chapter introduces the basic workflow of serverless deployments. You'll learn how to manage infrastructure with the AWS Serverless Application Model (SAM) and CloudFormation, and how to use API Gateway to send HTTP requests to Lambda functions.



The SAM command-line tool

SAM command-line tools can generate sample projects and events so that you can get started easily.

The `sam init` command #

Let's dive right in and create a sample web service backed by a Lambda function. You will use the `code` directory to execute the commands. To create a simple SAM project in a new subdirectory, run the following command:

```
sam init --runtime nodejs12.x --name app --app-template hello-world
```



You will be asked for the following permission:

```
Allow SAM CLI to download AWS-provided quick start templates from Github [Y/n]:
```

Enter `y` as a response. You should get a quick confirmation about the initialisation of a new project:

```
Cloning app templates from https://github.com/aws-labs/aws-sam-cli-app-templates.git
-----
Generating application:
-----
Name: app
Runtime: nodejs12.x
Dependency Manager: npm
Application Template: hello-world
Output Directory: .
```

If this command printed an error on your local machine, there is some issue with the setup. You should look at the chapter where we have the information on how to set up SAM command-line tools and other prerequisites.

You can try the above commands in the terminal provided below:

Environment Variables		^
Key:	Value:	
LANG	C.UTF-8	
LC_ALL	C.UTF-8	
AWS_ACCESS_KEY_ID	Not Specified...	
AWS_SECRET_ACCE...	Not Specified...	
BUCKET_NAME	Not Specified...	
AWS_REGION	Not Specified...	

The `--runtime` argument

The `--runtime` argument tells SAM which programming language you'll use to write the Lambda function, or more precisely which execution environment it is intended to run in. The `--runtime` argument is important for generating the sample project. You will learn how to add more functions to the same project later, and you can even mix functions that are executed in different languages.

To speed up scaling and operations, Lambda has pre-packaged execution environments, called *runtimes*, for many popular languages. The `nodejs12.x` runtime tells Lambda to run your function under Node.js 12, an execution engine for JavaScript. SAM can also create example projects in other languages. For example, use `java8` for a Java function. Check out the [AWS Lambda runtimes](#) page for a complete list.

The `--name` argument

The `--name` argument tells SAM how to call the application, or more precisely the name for the subdirectory where the application files are stored. In the previous example case, `app` was used, so SAM will create a new subdirectory called `app` and copy the sample files there.

Here are the key ones:

- `hello-world` is a directory containing the JavaScript (Node) source code for your Lambda function.
- `hello-world/package.json` is the standard Node.js package manifest, describing dependencies required by the function.
- `hello-world/app.js` is the actual service request handler.
- `template.yaml` describes the operational infrastructure required by the function.

You can check out these files in the terminal in which you executed your commands by changing the directory using `cd` and then using the `ls` and `cat` commands.

The `--app-template` argument

The `--app-template` argument tells SAM which template to apply when initialising the application. You can see the list of standard templates in the `aws-sam-cli-app-templates` project on GitHub. SAM can also use your templates, which might be useful for teams that often create similar Lambda functions. For example, you could create a template for message queue handling, and then quickly apply it when creating a new payment processor that connects to your payments queue. Companies can also use templates to standardise project directory layouts. To specify your own template location, use the `--location` parameter and point to a GIT repository or a local directory.

Now let's proceed to the next lesson where you will learn about the *CloudFormation Template* and the *CloudFormation Stack*.