Setting Up the Prerequisites

This lesson contains instructions to help you set up the prerequisites to using AWS Lambda.

WE'LL COVER THE FOLLOWING

This chapter explains how to install and configure AWS SAM (Serverless Application Model) command-line tools and prerequisites.

The AWS SAM

The AWS Serverless Application Model (SAM) is a set of products that simplify developing, testing, and deploying applications using AWS Lambda. One part of SAM runs on developer machines and build servers, helping to prepare for deployment. Another aspect of SAM runs in AWS data centres during the deployment process. Unfortunately, as is the case with much AWS software deployment process. Unfortunately, as is the case with much AWS software, these products have overlapping names. Nobody outside Amazon really cares too much about the distinction, so informally they all get called SAM. Although it is possible to use different SAM products separately, in most cases they work together, so a clean separation between them isn't especially important. In this chapter, you'll learn how to set up the first part of SAM, which runs on developer machines.

> This product was previously called 'SAM-local', and today AWS documentation calls it the SAM Command Line Interface (SAM CLI).

The main command-line tool which you'll be installing and executing is just called `sam`. In the next chapter, you will take a deeper look at the part of SAM that runs remotely in AWS.

SAM development tools help with building and packaging projects for deployment to AWS, debugging, simulating a Lambda environment, retrieving logs, and generating sample events for testing. You can set up command-line tools so you can use them with any editor, integrated development environment, or build system. Various AWS toolkits for popular editors and development environments also include these tools, making it possible to test, simulate, and debug Lambda functions directly from a visual interface. The AWS Cloud9 cloud-based integrated development environment also supports SAM tools.

> **Note that you don't need to follow the provided setup instructions for this course. The coding environment for you to practice all the commands in the widgets throughout this course has already been set up. You can follow the setup instructions if you want to run code on your local machine. However, you are encouraged to use the widgets to get hands-on practice so that you save time and effort. So, if you want to use the environment on this platform, you can skip two lessons and move directly to the Configuring Access Credentials lesson.**

Installing the required tools

AWS SAM CLI depends on a set of underlying tools which will be explained below, but whose full setup instructions are out of the scope of this course. That kind of information is easy to find online once you know what to look for, and it is likely to be updated more frequently directly on the individual tool websites. If you experience any problems with installing the tools below or want to use an alternative way of managing the packages, check out the *AWS SAM Developer Guide* online.

Python

The SAM command-line tools are actually a set of Python scripts so you will need the Python runtime installed on your machine. You can use Python 2 version 2.7 or later, or Python 3 version 3.6 or later. Most Linux and macOS machines already have some version of Python installed. If you are unsure about this, check whether Python is installed and which version you have by running the following command:

```
python --version
```

If this command prints an error or you are using an outdated version, you will need to upgrade Python. In case you need to upgrade or install Python, get the one-click installer from https://www.python.org.

PIP

In addition to the Python runtime, you will also need to have the `pip` package management tool. Most Python installations already have one. To check whether it is installed on your machine, run the following command:

```
pip --version
```

If this command prints an error, `pip` is missing on your system, so install it using the instructions from https://pip.pypa.io.

AWS

The final prerequisite is the basic AWS command-line tools package. Most developers that access AWS in any way usually have those tools already installed. To check whether your system already has those tools, run the installed. To check whether your system already has these tools, run the following command:

```
aws --version
```

If the command prints an error, you can download the tools by running the following command:

```
pip install awscli
```

Alternatively, get a ZIP or a binary installer for your operating system from the AWS CLI documentation site.

> **Installing on macOS**
>
> In order for Python to work on macOS, you will also need to set up basic development command-line tools. You can install them by running the following command:
>
> ```
> xcode-select --install
> ```
>
> If you installed Python using `brew`, make sure it's version 2.7 or later. Older versions will not work well with SAM.

Docker

SAM tools use Docker, a container management system, to simulate the Lambda execution environment for local testing and debugging. You don't need to install the full Docker service; the free (community) Docker Desktop tools are enough for all development tasks, including working with all the examples in this course. If you do not have Docker Desktop tools already installed, get the one-click installer for your operating system directly at the Docker website. You will not need Docker to actually run the code in production.

In the next lesson, you will learn how to install NodeJs and AWS SAM CLI.