Generating Test Events

In this lesson, you will learn how to generate test events using `sam local` and how to work with file streams.

Events coming from S3 will have a different structure to those coming from API Gateway. When you started experimenting with API events in Chapter 6, you used a nice trick to understand the structure: create an empty function that just logs the event to CloudWatch at the start. You could do something similar now, but SAM has another tool that makes this even simpler. You can use `sam local generate-event` to create sample events.

For example, run this command to see what an S3 object upload event looks like:

```
sam local generate-event s3 put
```

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...

● Terminal

You can use this trick to generate test events for many AWS services. The first parameter after `generate-event` is the service name (in this case, `s3`). The second is the type of the event (in this case, you're looking for `put` events). To see the full list of services that SAM can generate events for, run the following command:

```
sam local generate-event --help
```

To see a list of events SAM can generate for a particular service, list the service name before `--help`. For example, to see a list of possible events for `s3`, run the following command:

```
sam local generate-event s3 --help
```

From the sample event, you can see that the event contains an array of records with a single element. The record has an `s3` property that contains the `bucket` and `object` sub-properties. You'll need to pull out the bucket name and the key from the file upload event.

To create a utility function that extracts just the information you need from the event, add a file called `extract-s3-info.js` to the new function code directory (`image-conversion`).

```
module.exports = function extractS3Info(event) {
  const eventRecord = event.Records && event.Records[0],
    bucket = eventRecord.s3.bucket.name,
    key = eventRecord.s3.object.key;
  return {bucket, key};
};
```

code/ch9/image-conversion/extract-s3-info.js

In the next lesson, you'll learn how to work with files!