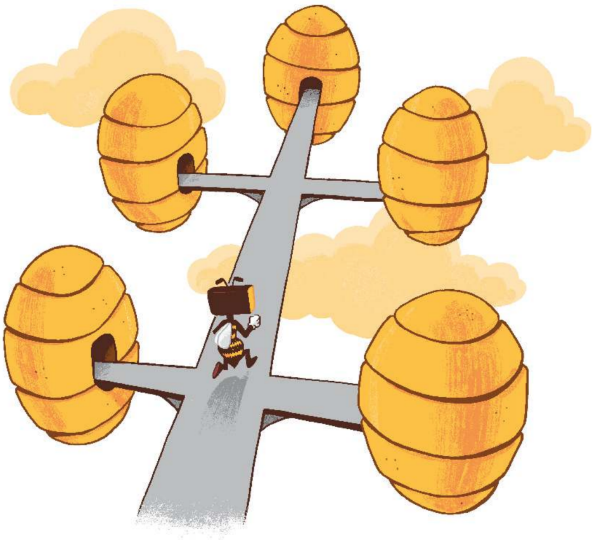
API Gateway Events

In this lesson, you will learn about API Gateway events and how you can view them using the AWS Web Console.

WE'LL COVER THE FOLLOWING

- The API Gateway web console
    - Swagger API definitions
- Resource in API Gateway

This chapter explains how to create HTTP APIs and dynamic web pages using Lambda. You'll also learn about customising API Gateway resources and how to pass parameters to CloudFormation templates.



So far, you've mostly looked into managing Lambda functions, but the sample project you deployed in Chapter 3 configured one more resource: the API Gateway. API Gateway is a service for publishing and managing REST and Websocket APIs.

You're currently using it to just pass through requests from client browsers to a Lambda function, but it can do much more. API Gateway can throttle requests to prevent clients from overloading a back-end service, authenticate and authorise requests, enforce usage plans, and even transform or enrich and authorise requests, enforce usage plans, and even transform or enrich incoming and outgoing payloads. API Gateway does not necessarily have to forward requests to a Lambda function; it could potentially send them to some other web API you host, or to a different AWS service. In the scope of this course, though, you'll always use the Lambda backend.

The API Gateway web console

If a Lambda function has any `Api` events associated with it, SAM will create an API Gateway and wire it up to the function. The `Events` section of the sample template has one such event:

```
Events:
  HelloWorld:
    Type: Api
    Properties:
      Path: /hello
      Method: get
```

Line 17 to Line 22 of code/basic-template/template.yaml

You can easily find the API Gateway resource that SAM configured for you in the AWS Web Console:

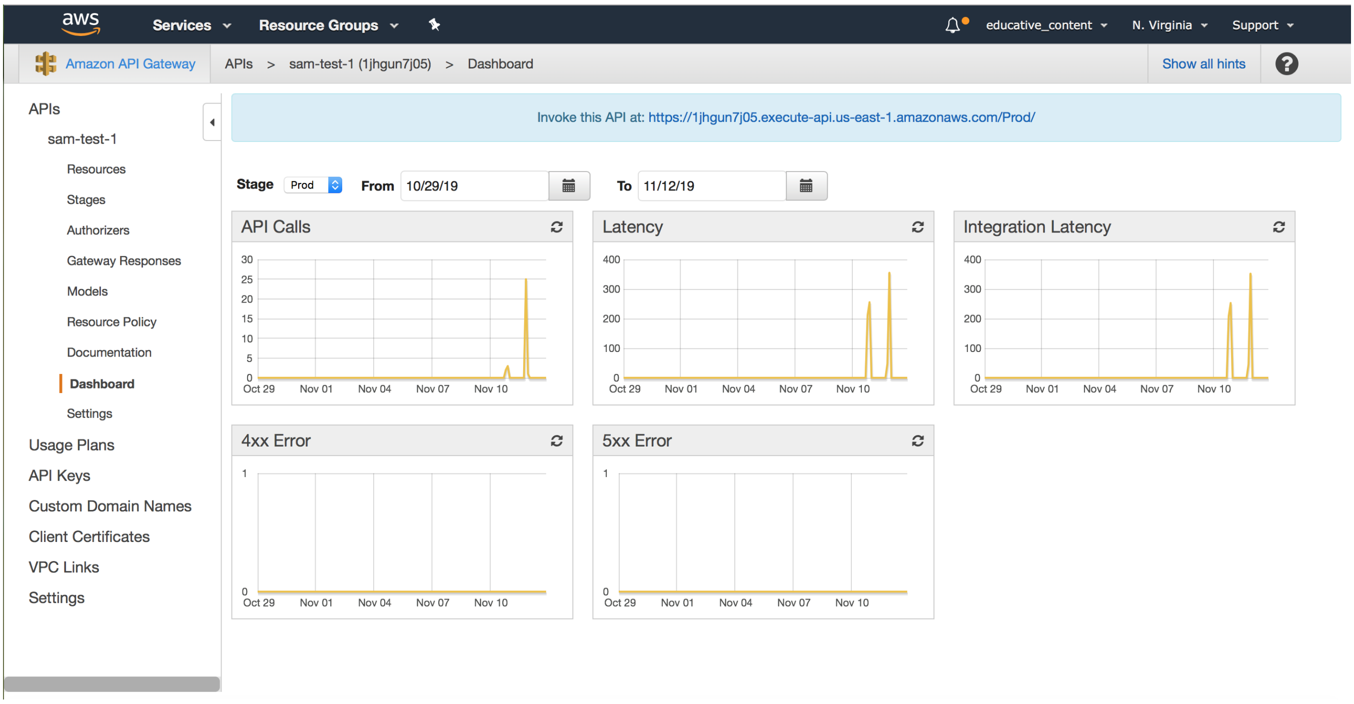
1. Open https://aws.amazon.com and sign in if necessary.
2. Select the CloudFormation service and find the deployed stack (`sam-test-1`).
3. In the *Resources* tab, look for an item of type `AWS::ApiGateway::RestApi` and click the hyperlink in that row.

Alternatively, select the *API Gateway* service in the AWS Web Console and find the API with the same name as your stack (if you followed the steps in Chapter 3, this would be `sam-test-1`).

> ### Swagger API definitions
>
> API Gateway can also support rich API model definitions through Swagger, a popular API configuration framework. This might be useful if you are deploying a service API for strongly typed languages or mobile clients because you can even generate the client SDK code based on a Swagger template. In this course, though, you'll always use the Proxy

> integration since JavaScript has no types anyway.

The API Gateway Web Console shows several sections under the API name, such as *Resources*, *Stages*, and *Dashboard*. The Dashboard screen (see figure below) shows a quick summary of recent requests, along with basic latency metrics. It might be interesting to see basic operational statistics during troubleshooting.



Invoke this API at: https://1jhgun7j05.execute-api.us-east-1.amazonaws.com/Prod/

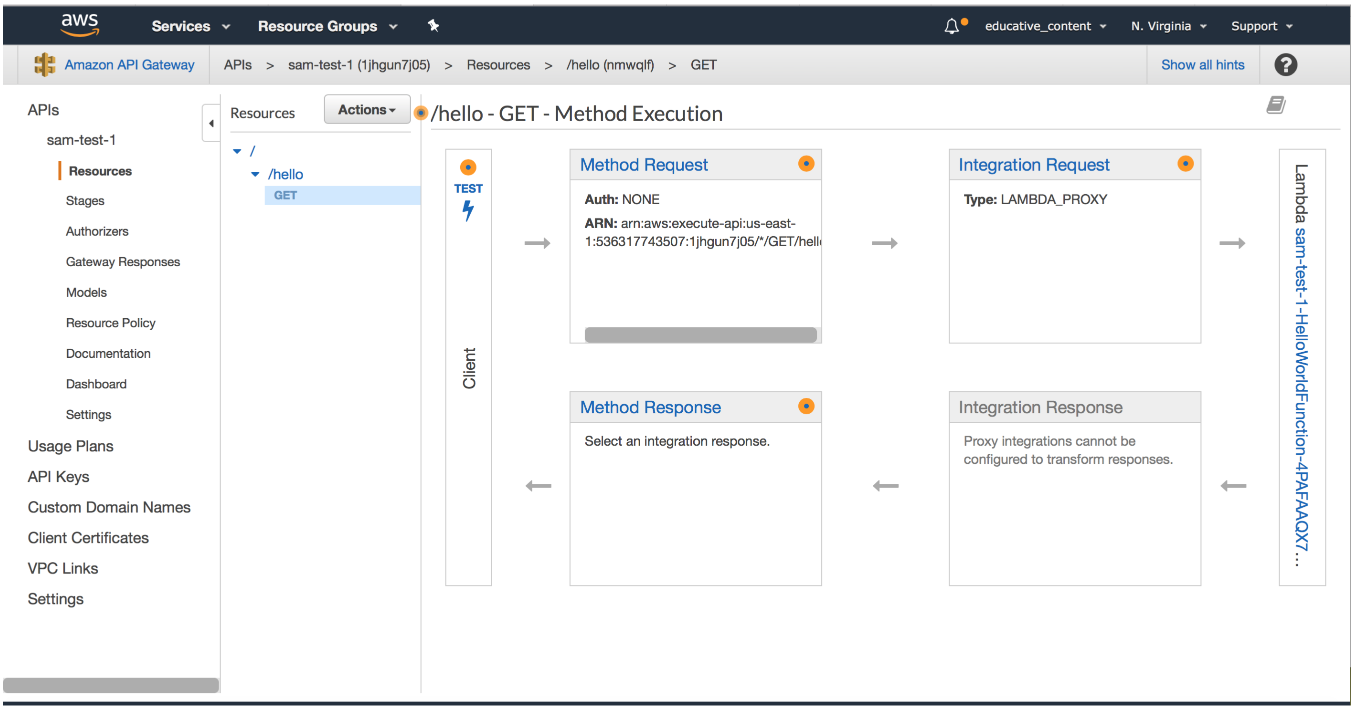
Stage Prod From 10/29/19 To 11/12/19

API Gateway Web Console dashboard shows basic operational statistics.

Resource in API Gateway

In the API Gateway terminology, a *resource* is an endpoint configured to handle an HTTP request on a specific path, and with a specific HTTP method. SAM automatically added a single resource based on the `Events` property of the Lambda function and configured to handle `GET` requests to the `/hello` path. Select the *Resources* section in the left menu under your API to see that resource (see figure below).

To keep things simple, SAM did not set up any authentication and did not configure any conversions or transformations for the responses. It just set up API Gateway to pass all the data to Lambda and returned back whatever Lambda sent it. This type of integration is called *Lambda Proxy*, which you will be able to see in the *Integration Request* box on the screen of your Web Console (top-right box in the figure below). Note that Lambda proxy responses do not just contain the raw HTML, but also a JSON object with HTTP response information, explained more in the next section.



/hello - GET - Method Execution

Auth: NONE

ARN: arn:aws:execute-api:us-east-1:536317743507:1jhgun7j05/*/GET/hell

Type: LAMBDA_PROXY

Select an integration response.

Proxy integrations cannot be configured to transform responses.

API Gateway Web Console shows the details of a resource, including the authentication setup and the back end it talks to. It also enables us to send test requests easily.

Now that you have been introduced to the API gateway events, in the next lesson, you will learn how to customise responses for API Gateway.