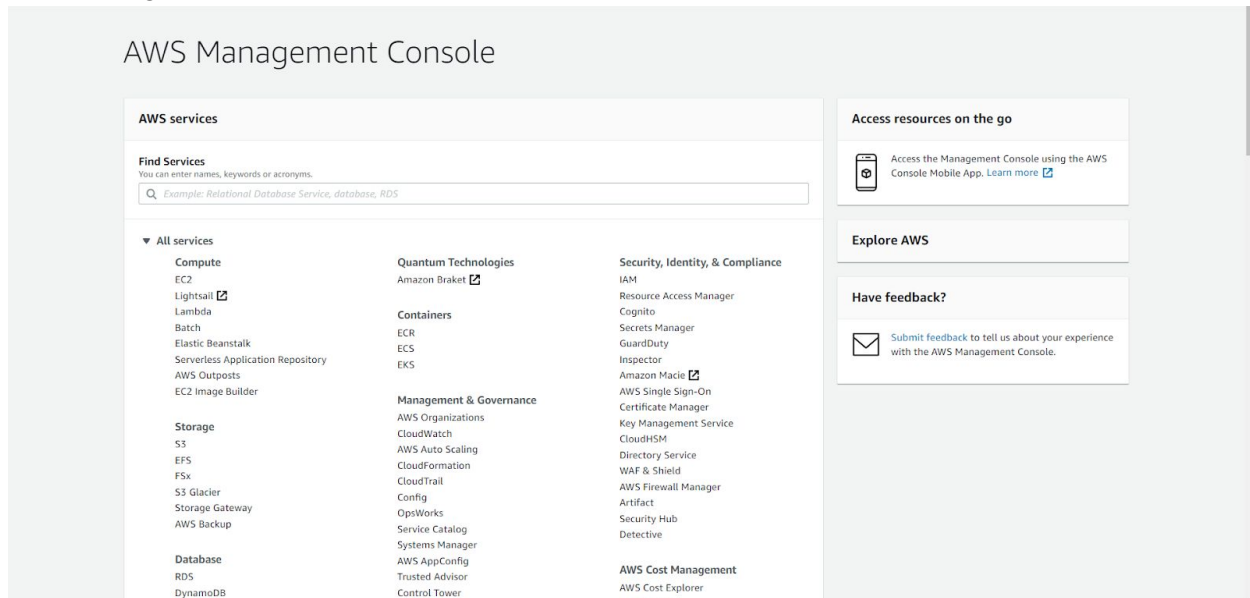


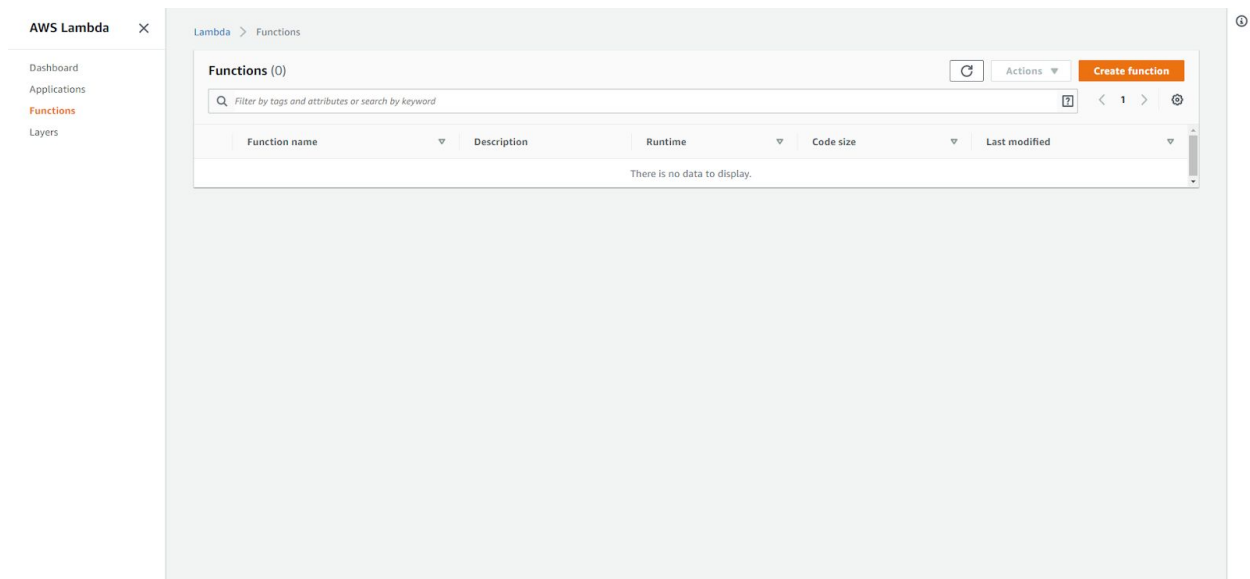
Module 6 Lab 1: Hello World API Gateway Trigger

AWS Management Console



Selling Northern California for this lab.

AWS Lambda



Creating a new Function from scratch.

Create a Function


Lambda > Functions > Create function

Create function [Info](#)

Choose one of the following options to create your function.


Author from scratch

Start with a simple Hello World example.




Use a blueprint

Build a Lambda application from sample code and configuration presets for common use cases.



Browse serverless app repository

Deploy a sample Lambda application from the AWS Serverless Application Repository.



Basic information

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.




Runtime [Info](#)
Choose the language to use to write your function.

Permissions [Info](#)
Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add triggers.
▶ Choose or create an execution role

Cancel Create function

Various ways to start a function.

Basic Information

Basic information


Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function.

Permissions [Info](#)
Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add triggers.
▼ Choose or create an execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).
☒ Create a new role with basic Lambda permissions
☐ Use an existing role
☐ Create a new role from AWS policy templates

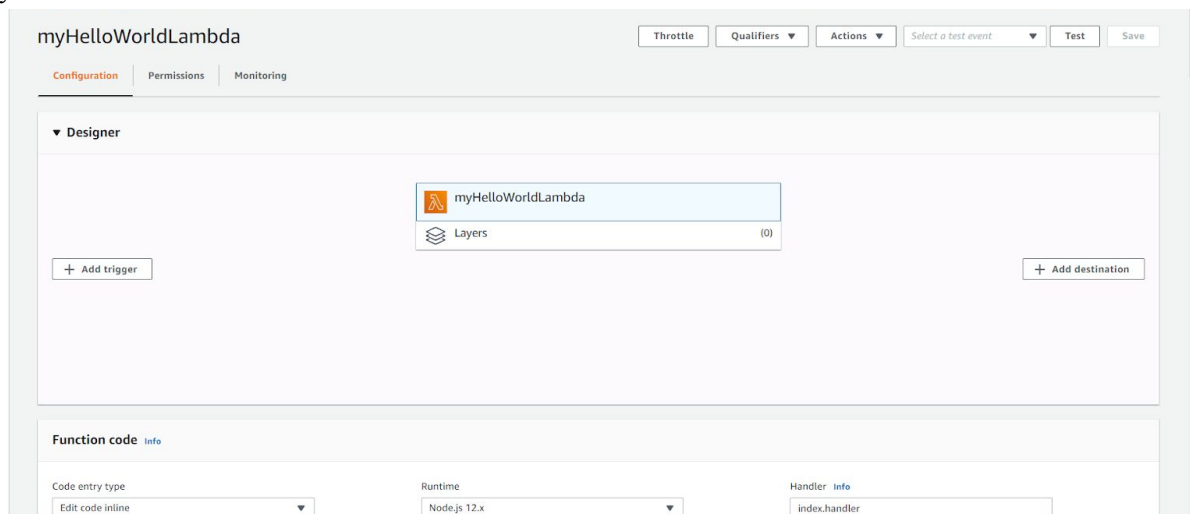
 Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Lambda will create an execution role named myHelloWorldLambda-role-gg8njjc, with permission to upload logs to Amazon CloudWatch Logs.

Cancel Create function

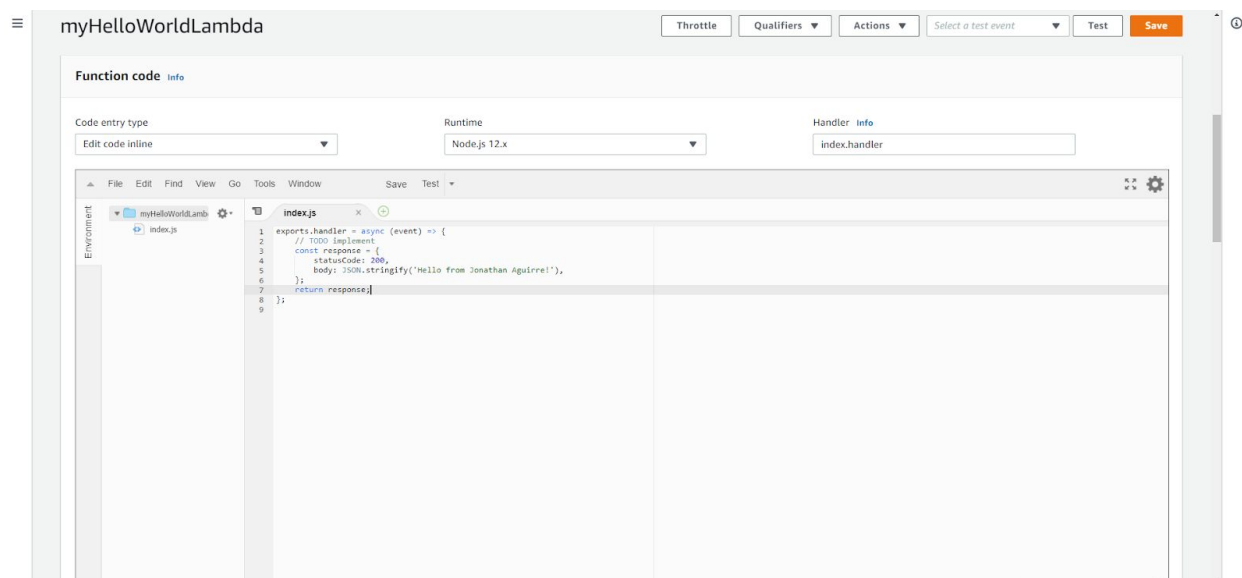
The basic information for this lambda such as name, runtime and role permissions.

myHelloWorldLambda



myHelloWorldLambda created.

Function Code



Entry change where a string will return “Hello from Jonathan Aguirre!”.

myHelloWorldLambda

The screenshot shows the configuration page for the Lambda function 'myHelloWorldLambda'. At the top, there are tabs for 'Throttle', 'Qualifiers', 'Actions', 'Select a test event', 'Test', and 'Save'. Below these, the function's code is shown as '7/21 JavaScript Spaces: 4'. The main configuration area is divided into several sections: 'Environment variables (0)' with an 'Edit' button; 'Tags (0)' with a 'Manage tags' button; 'Execution role' with a dropdown menu and a link to the IAM console; and 'Basic settings' with an 'Edit' button. The 'Basic settings' section shows 'Description' as '-', 'Memory (MB)' as '128', and 'Timeout' as '0 min 3 sec'.

Environment variables and tags can be set here.

myHelloWorldLambda

This screenshot shows the 'Execution role' and 'Basic settings' sections of the Lambda function configuration. The 'Execution role' section has a dropdown menu set to 'Use an existing role' and a link to the IAM console. Below this, the 'Existing role' section shows a dropdown menu with the selected role 'service-role/myHelloWorldLambda-role-ggff8njc' and a link to view the role in the IAM console. The 'Basic settings' section shows 'Description' as '-', 'Memory (MB)' as '128', and 'Timeout' as '0 min 3 sec'. Below these sections, there are sections for 'VPC' (set to 'None') and 'AWS X-Ray' (with a checkbox for 'Active tracing' and a link to 'View traces in X-Ray').

Execution role includes a newly made service role made for this function.

myHelloWorldLambda

myHelloWorldLambda

Throttle Qualifiers Actions Select a test event Test Save

Concurrency

Unreserved account concurrency **1000**

☒ Use unreserved account concurrency
☐ Reserve concurrency

Provisioned concurrency

To enable your function to scale without fluctuations in latency, use provisioned concurrency. Provisioned concurrency runs continually and has separate pricing for concurrency and execution duration. [Learn more](#)

Provisioned concurrency configurations (0)

Find configuration

Qualifier	Type	Provisioned concurrency	Status	Details
No configurations				

Add configuration

Asynchronous invocation

Maximum age of event: [Info](#)
6 h 0 min 0 sec

Dead-letter queue service: [Info](#)
None

Retry attempts: 2

Edit

The rest of the settings. All have been left at default so far.

Basic settings

Lambda > Functions > myHelloWorldLambda > Edit basic settings

Edit basic settings

Basic settings

Description - optional

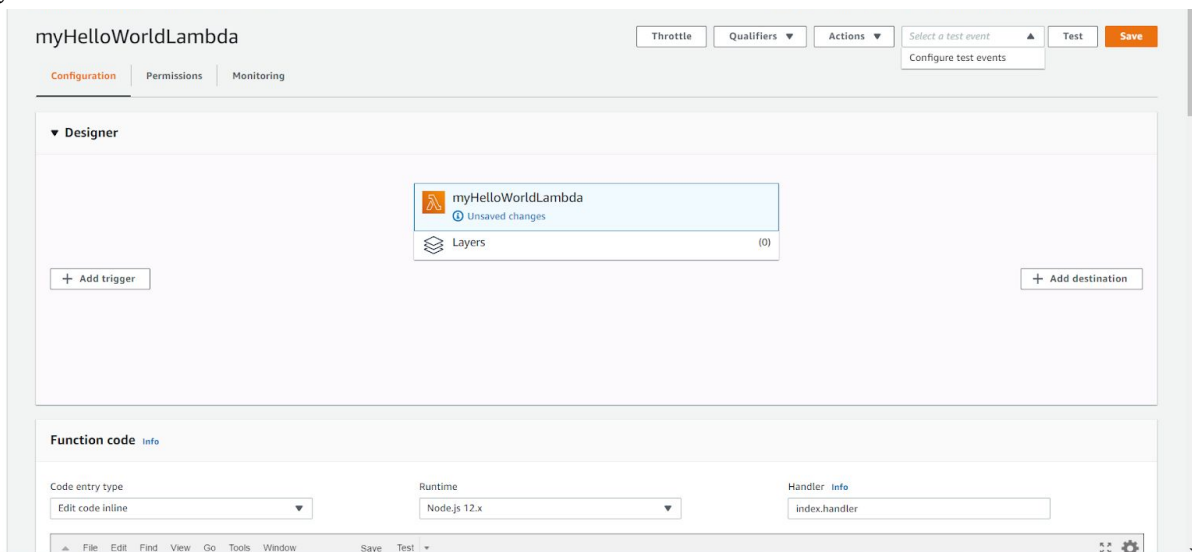
Memory (MB) [Info](#)
Your function is allocated CPU proportional to the memory configured.
128 MB

Timeout [Info](#)
0 min 3 sec

Cancel Save

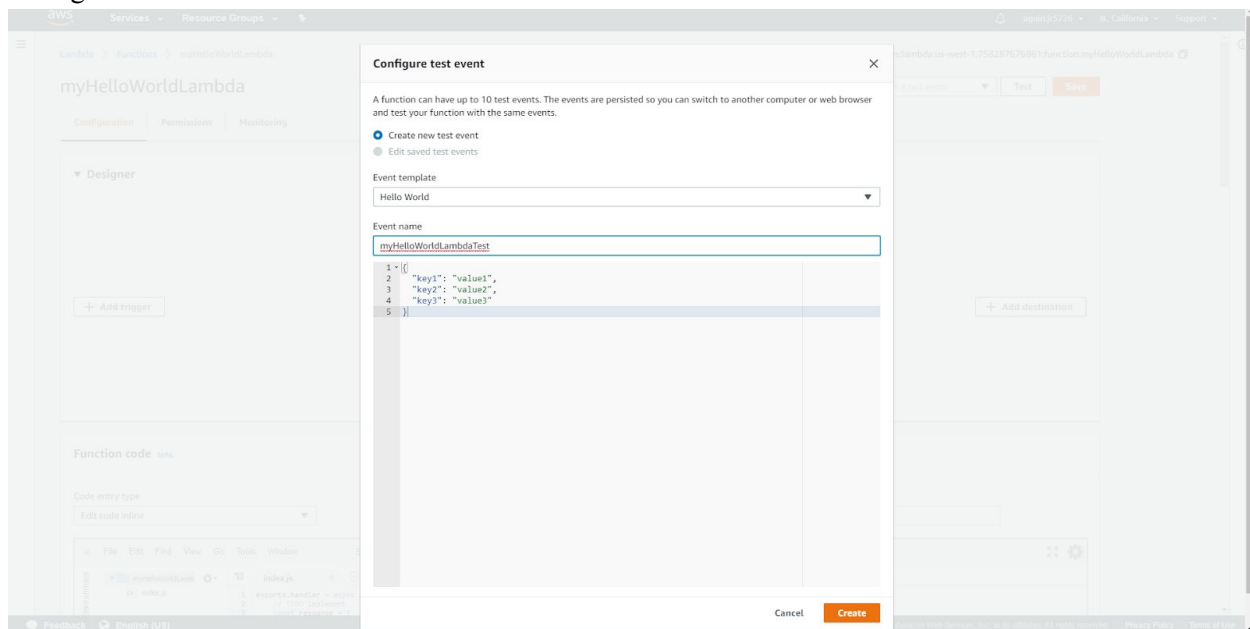
Increasing memory will also increase CPU usage.

myHelloWorldLambda



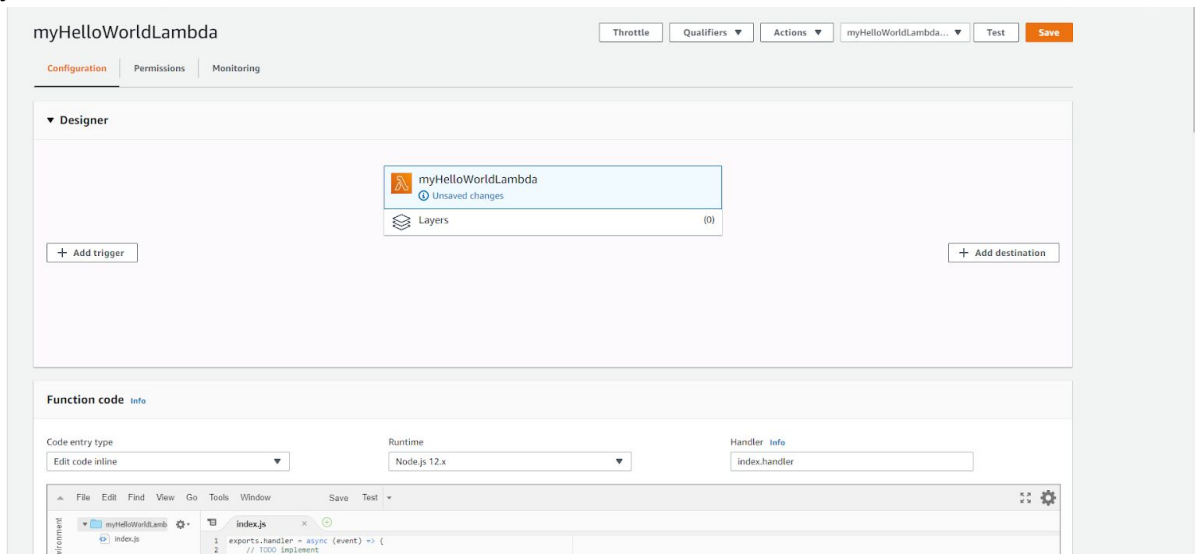
Scrolling back to the top and choosing “Select a test event”.

Configure test event



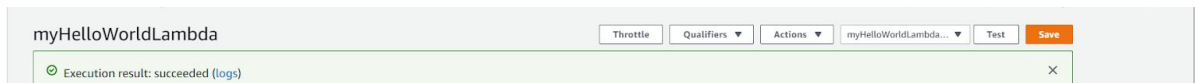
Creating a test event, nothing will be sent.

myHelloWorldLambda



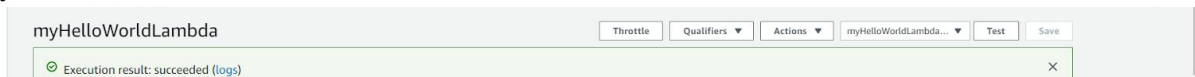
Selecting “Test” to test the event we just created.

Execution result: succeeded



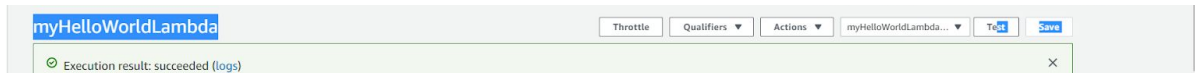
Note that the result returned does not show greetings with Jonathan Aguirre. This is because I didn't save the function, top right hand corner.

myHelloWorldLambda



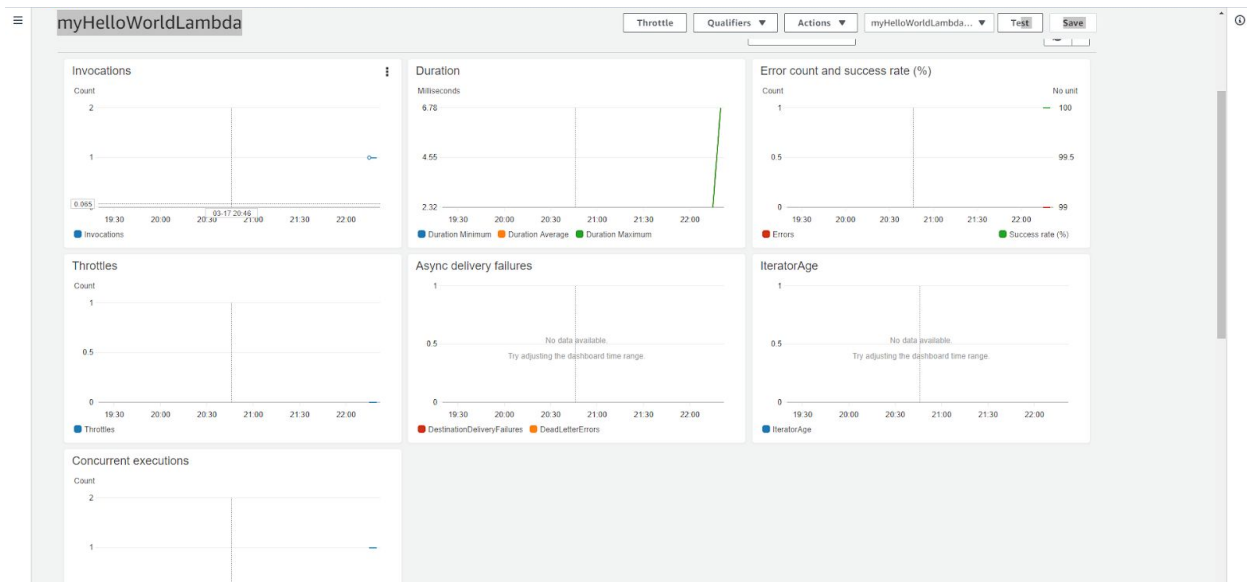
Successfully saved but the test did not rerun.

myHelloWorldLambda



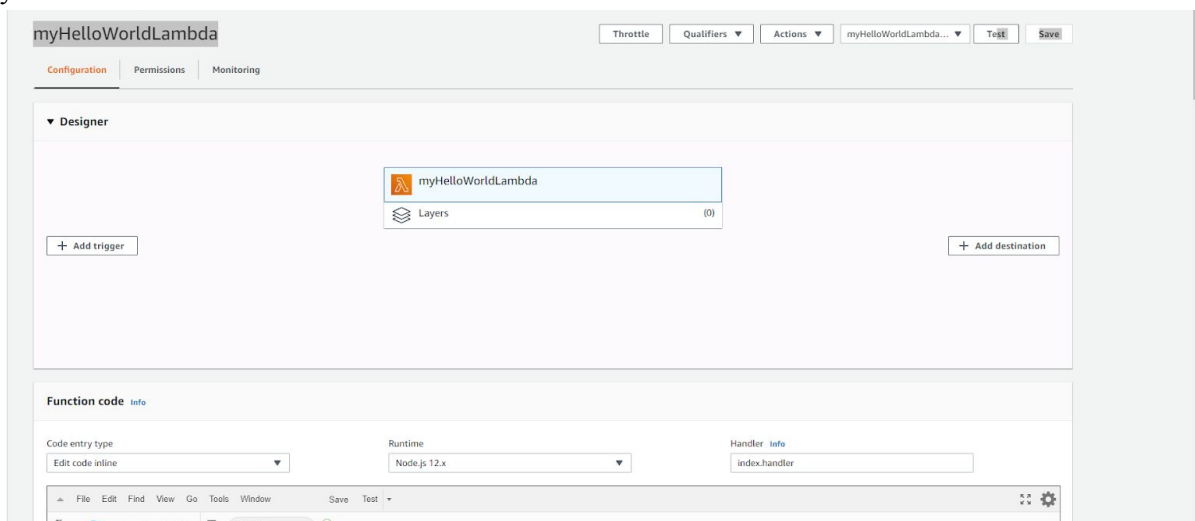
The results now match.

CloudWatch metrics information



The result of testing myHelloWorldLambda from CloudWatch.

myHelloWorldLambda



Back at the Configuration page, selecting “Add trigger”.

Add trigger

Trigger configuration

API Gateway
api application-services aws serverless

We'll set up an API Gateway endpoint with a **proxy integration** type (learn more about the input and output format). Any method (GET, POST, etc.) will trigger your integration. To set up more advanced method mappings or subpath routes, visit the [Amazon API Gateway console](#).

API
Pick an existing API, or create a new one.
[Create a new API](#)

Choose a template

☒ **HTTP API**
Ideal for building REST APIs that proxy to Lambda functions.

☐ **REST API**
Ideal for building REST APIs where you want full control over the request and response with API management capabilities.

What are we creating in this template?
This template enables you to create REST APIs that are fully managed, cost efficient, and high performance. Get started quickly with a single endpoint. Then add additional capabilities such as routes, rate limiting, and custom domains.

What you will get

- Ease of use: easy to operate and configure.
- Low latency: ideal for building APIs for latency-sensitive workloads.
- Cost effective: ideal for workloads that are likely to grow large.

To learn more about various features supported go to [API Gateway Documentation](#).

Security
Configure the security mechanism for your API endpoint. [Learn more](#).

Additional settings

Lambda will add the necessary permissions for Amazon API Gateway to invoke your Lambda function from this trigger. [Learn more about the Lambda permissions model](#).

The trigger for this function will consist of an API Gateway. Gives the function an HTTP URL for function execution. Did not create the Gateway here.

Amazon API Gateway

Amazon API Gateway
create, maintain, and secure APIs at any scale

Amazon API Gateway helps developers to create and manage APIs to back-end systems running on Amazon EC2, AWS Lambda, or any publicly addressable web service. With Amazon API Gateway, you can generate custom client SDKs for your APIs to connect your back-end systems to mobile, web, and server applications or services.

Choose an API type

HTTP API
Build low-latency and cost-effective REST APIs with built-in features such as OAuth and OAuth2, and native CORS support.
Works with the following: Lambda, HTTP backends
[Import](#) [Build](#)

WebSocket API
Build a WebSocket API using persistent connections for real-time use cases such as chat applications or dashboards.
Works with the following: Lambda, HTTP, AWS Services
[Build](#)

REST API
Develop a REST API where you gain complete control over the request and response along with API management capabilities.
Works with the following: Lambda, HTTP, AWS Services
[Import](#) [Build](#)

REST API (private)
Create a REST API that is only accessible from within a VPC.
Works with the following: Lambda, HTTP, AWS Services
[Import](#) [Build](#)

Creation of the Amazon API Gateway via the type “REST API”.

Amazon API Gateway

The screenshot shows the 'Create your first API' modal dialog in the Amazon API Gateway console. The dialog is titled 'Create your first API' and contains a welcome message: 'Welcome to Amazon API Gateway. To create your first API, we have pre-populated the import form with a Pet Store API defined using Swagger 2.0. To get started, close this modal and select Import in the Create API form.' There is an 'OK' button at the bottom right of the modal. The background shows the 'Create' page of the Amazon API Gateway console, which includes sections for 'Choose the protocol', 'Create new API', 'Example API', and 'Settings'.

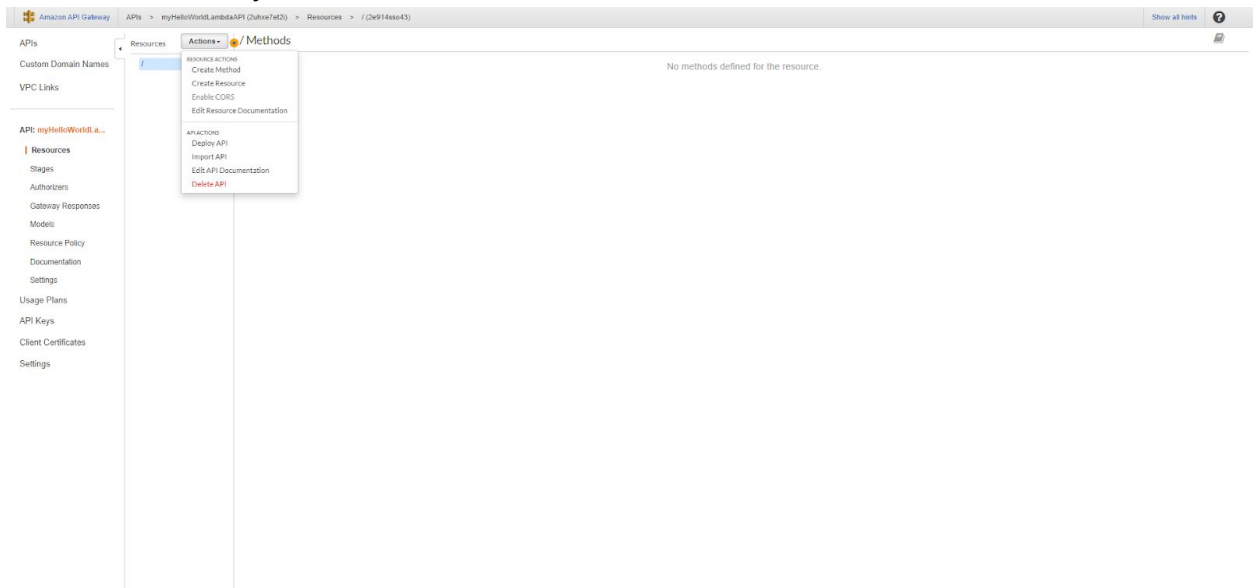
Welcome screen to Amazon API Gateway for the first time.

Amazon API Gateway Create

The screenshot shows the 'Create' page in the Amazon API Gateway console. The page is titled 'Create' and includes sections for 'Choose the protocol', 'Create new API', 'Settings', and 'Create API'. The 'Create new API' section is active, showing a form with the following fields: 'API name*' (pre-filled with 'myHelloWorldLambdaAPI'), 'Description' (empty), and 'Endpoint Type' (set to 'Regional'). There is a 'Create API' button at the bottom right of the page.

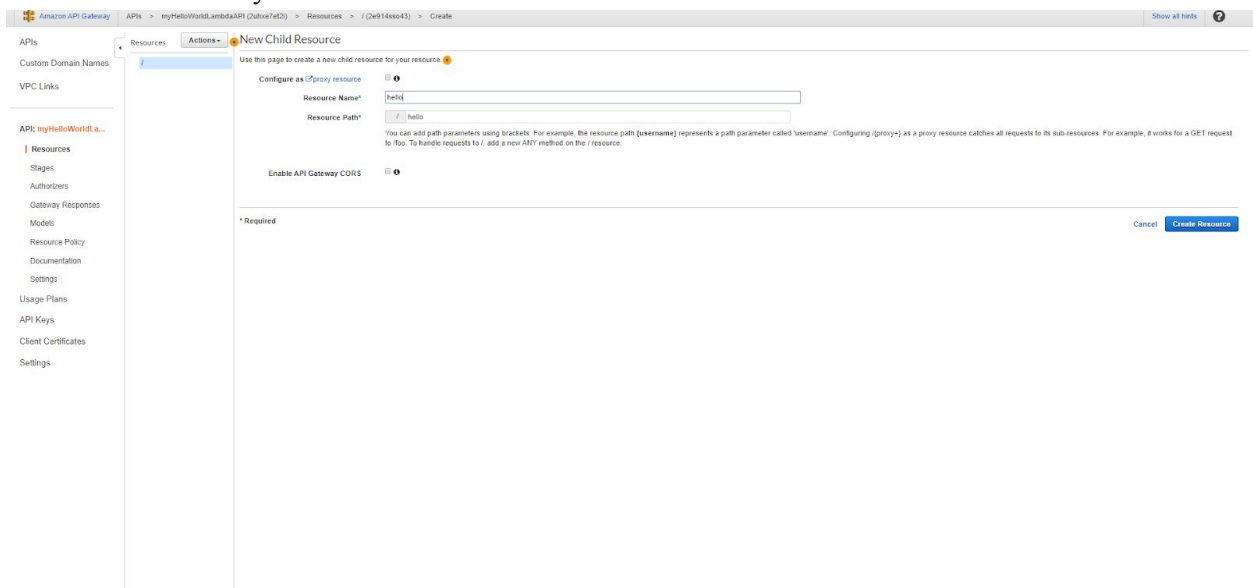
Choose New API and input an API name. Everything else left as default.

Amazon API Gateway



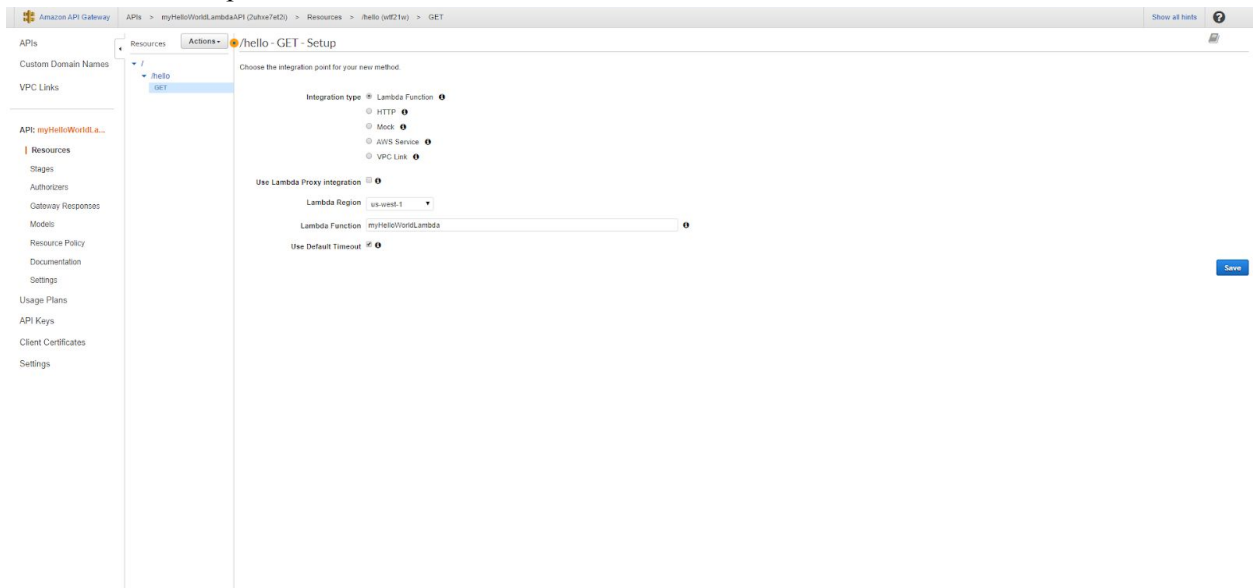
Creating a new resource.

Amazon API Gateway



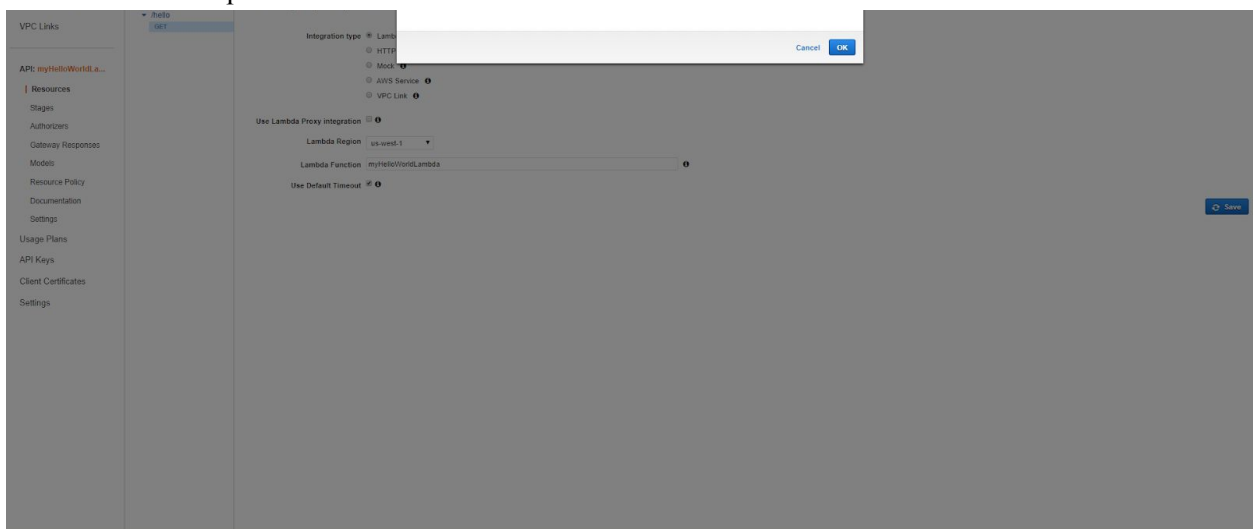
Created “hello” resource.

/hello - GET - Setup



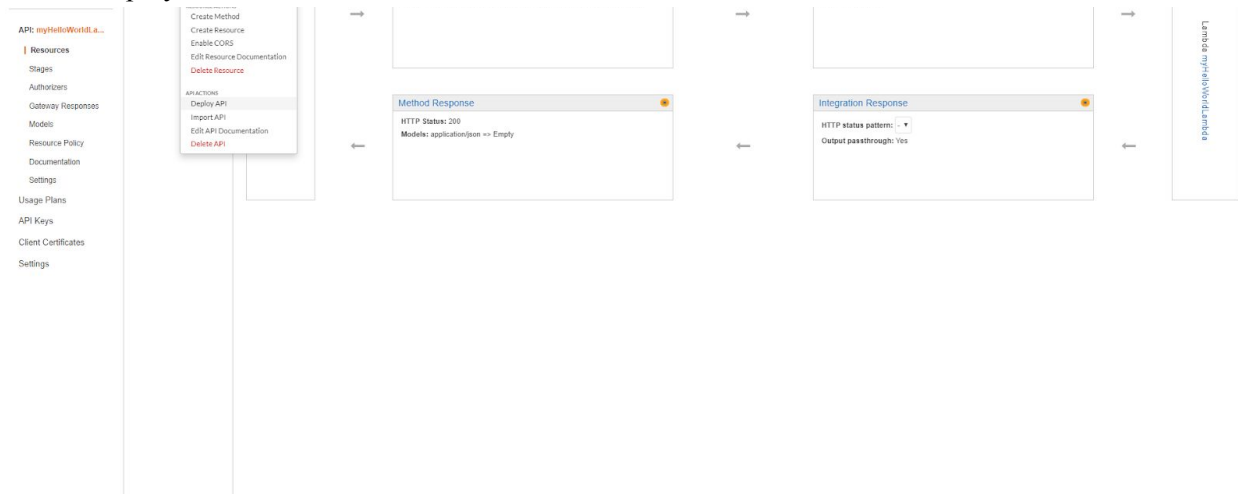
A GET method is set up with the Lambda Function created previously.

/hello - GET - Setup



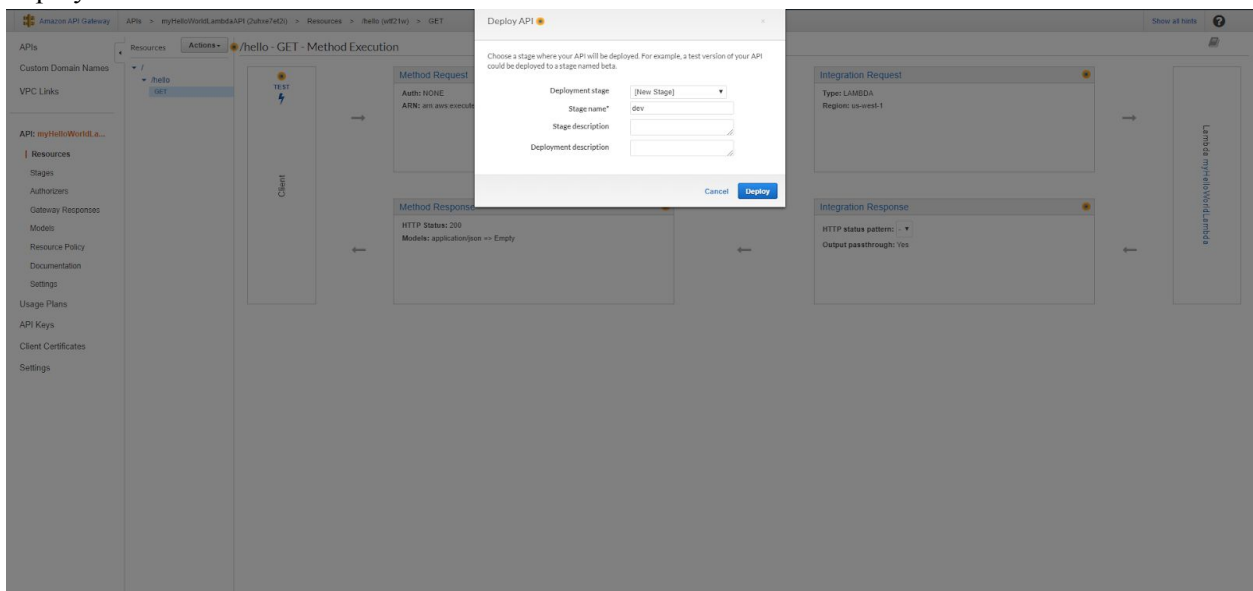
Confirmation of the GET method creation.

/hello - Deploy API



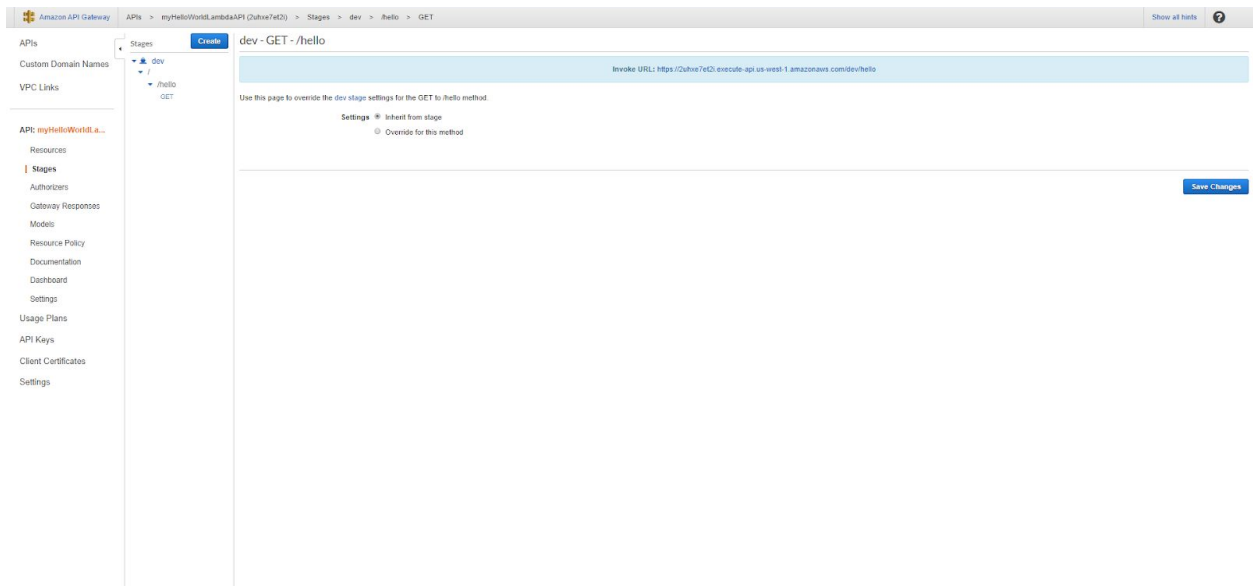
Deploy API under applications.

Deploy API



Stage name selected.

dev - GET -/hello



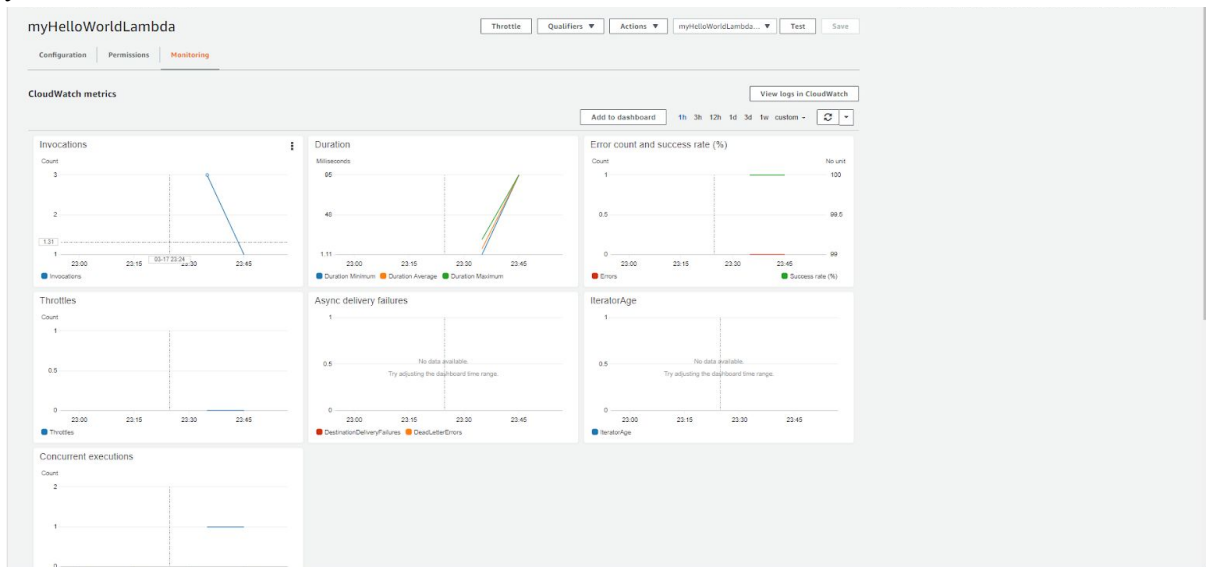
Invoke URL noted above in blue.

Webpage



Visiting the URL resource executes the lambda function created.

myHelloWorldLambda CloudWatch metrics



API Gateway has been attached and it's activity is seen via CloudWatch.