Step 1

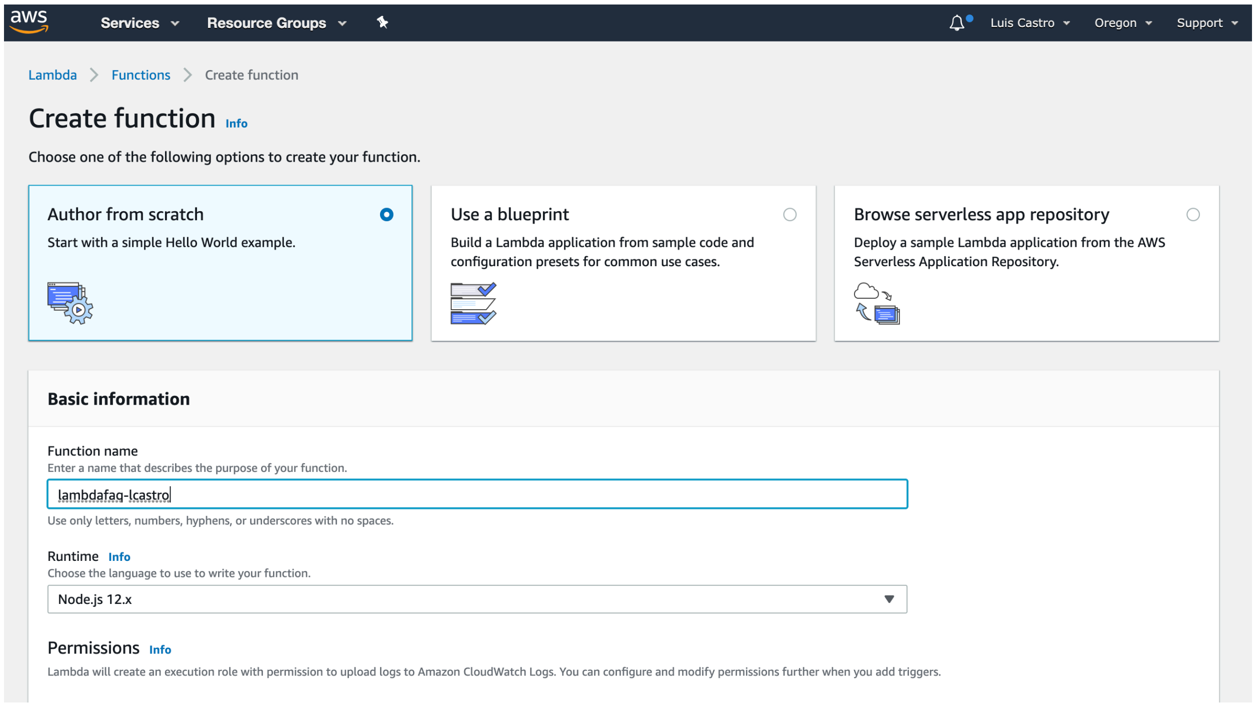
Access the AWS console through the following link:

https://450006219561.signin.aws.amazon.com/console

Step 2

Enter the Lambda service and create a function with the following nomenclature:

- **lambdafaq-<username>**
- Example:
  - lambdafaq-lcastro
- Choose Runtime
  - **Node-.js 10.x**
- Choose or create an execution role
  - Use an existing role
    - lambda_basic_execution
- **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

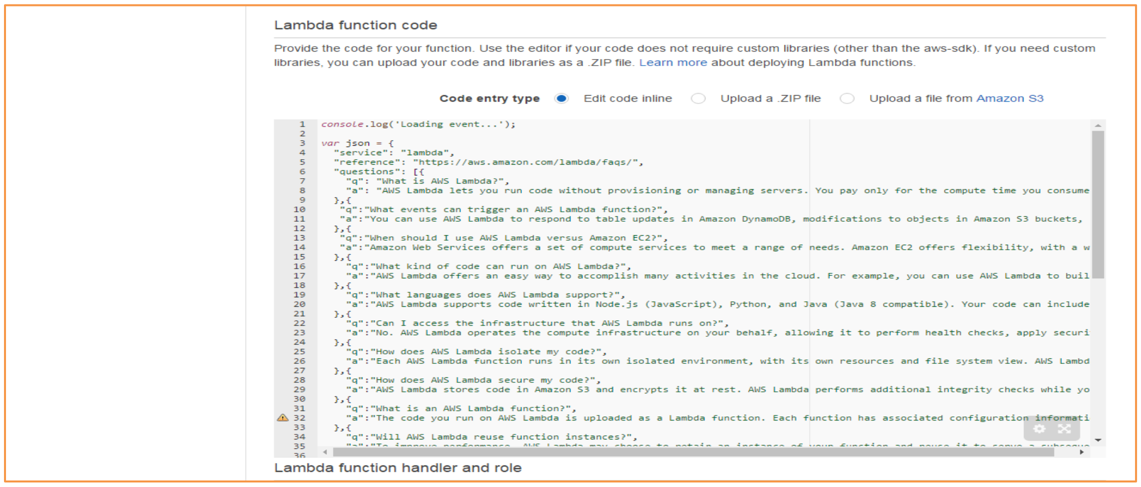
Function name
Enter a name that describes the purpose of your function.
lambdafaq-lcastro
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function.
Node.js 12.x

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.

Step 3

Open the file called faqlambda.txt, copy it and then paste it as Code Entry Type. Clear current code inside Index.js



Lambda function code

Provide the code for your function. Use the editor if your code does not require custom libraries (other than the aws-sdk). If you need custom libraries, you can upload your code and libraries as a .ZIP file. Learn more about deploying Lambda functions.

Code entry type [✓ Edit code inline] Upload a .ZIP file Upload a file from Amazon S3

```
console.log('Loading event...');

var json = {
  "service": "lambda",
  "reference": "https://aws.amazon.com/lambda/faqs/",
  "questions": [{
    "q": "What is AWS Lambda?",
    "a": "AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume
  },{
   "q":"What events can trigger an AWS Lambda function?",
   "a":"You can use AWS Lambda to respond to table updates in Amazon DynamoDB, modifications to objects in Amazon S3 buckets,
  },{
   "q":"When should I use AWS Lambda versus Amazon EC2?",
   "a":"Amazon Web Services offers a set of compute services to meet a range of needs. Amazon EC2 offers flexibility, with a w
  },{
    "q":"What kind of code can run on AWS Lambda?",
    "a":"AWS Lambda offers an easy way to accomplish many activities in the cloud. For example, you can use AWS Lambda to buil
  },{
    "q":"What languages does AWS Lambda support?",
    "a":"AWS Lambda supports code written in Node.js (JavaScript), Python, and Java (Java 8 compatible). Your code can include
  },{
    "q":"Can I access the infrastructure that AWS Lambda runs on?",
    "a":"No. AWS Lambda operates the compute infrastructure on your behalf, allowing it to perform health checks, apply securi
  },{
    "q":"How does AWS Lambda isolate my code?",
    "a":"Each AWS Lambda function runs in its own isolated environment, with its own resources and file system view. AWS Lambd
  },{
    "q":"How does AWS Lambda secure my code?",
    "a":"AWS Lambda stores code in Amazon S3 and encrypts it at rest. AWS Lambda performs additional integrity checks while yo
  },{
    "q":"What is an AWS Lambda function?",
    "a":"The code you run on AWS Lambda is uploaded as a Lambda function. Each function has associated configuration informati
  },{
    "q":"Will AWS Lambda reuse function instances?",
    "a":"To improve performance, AWS Lambda may choose to retain an instance of your function and reuse it to serve a subseque
```

Lambda function handler and role

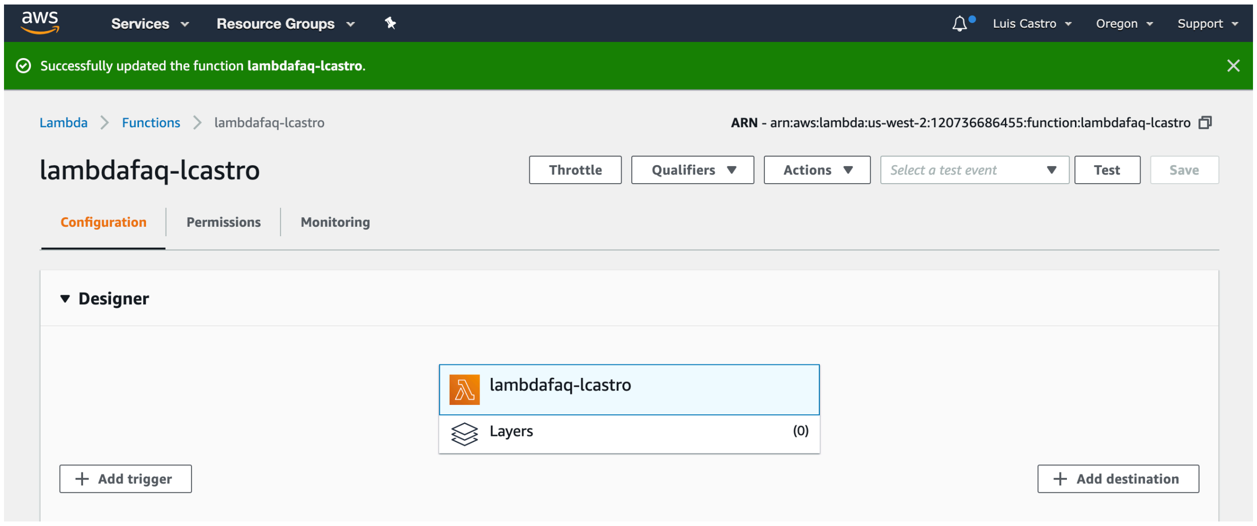
Step 4

Choose the following options

Handler

- Index.handler

Choose the other default values and click Save



Successfully updated the function **lambdafaq-lcastro**.

Lambda > Functions > lambdafaq-lcastro

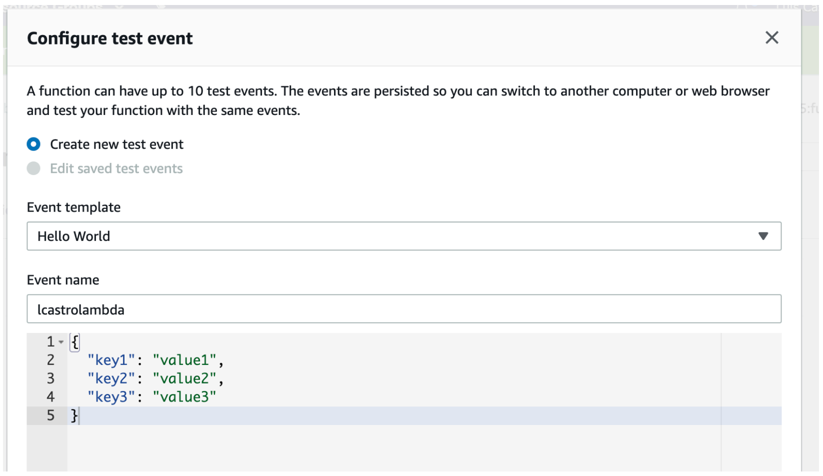
ARN - arn:aws:lambda:us-west-2:120736686455:function:lambdafaq-lcastro

lambdafaq-lcastro

Step 5

Inside the Lambda Test function menu

- **Create a New Test Event**
- Event Template
    - **Hello World**
- Event Name
    - **Nombre de usuario + Lambda**
        - **Lcastrolambda**
- Event Name
    - **Nombre de usuario + Lambda**
- **Eliminate keys 1, 2 y 3**
- Verify that it is as follows and click Save and Test:
    - **{ }**
- Click on **Create**



Configure test event

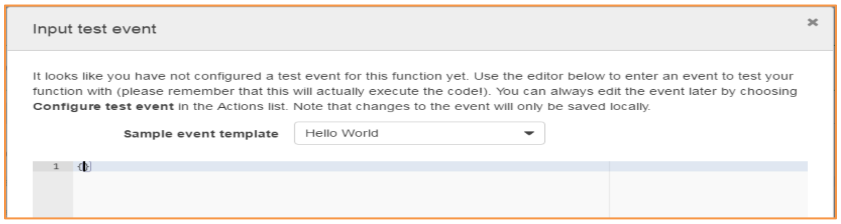
A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

- Create new test event
- Edit saved test events

Event template: Hello World

Event name: lcastrolambda

```
{
  "key1": "value1",
  "key2": "value2",
  "key3": "value3"
}
```



Input test event

It looks like you have not configured a test event for this function yet. Use the editor below to enter an event to test your function with (please remember that this will actually execute the code!). You can always edit the event later by choosing **Configure test event** in the Actions list. Note that changes to the event will only be saved locally.

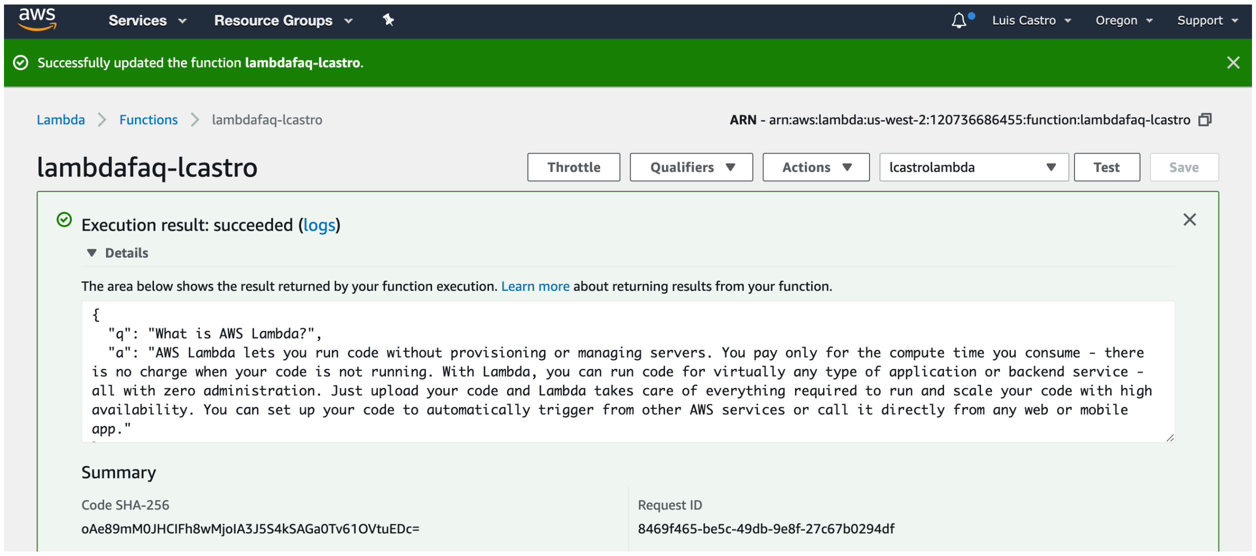
Sample event template: Hello World

```
{}
```

Step 6

Choose the previously created test (Ex: lcastrolambda) and click on Test

Execution result: Succeeded



Successfully updated the function **lambdafaq-lcastro**.

ARN - arn:aws:lambda:us-west-2:120736686455:function:lambdafaq-lcastro

lambdafaq-lcastro

Execution result: succeeded (logs)

The area below shows the result returned by your function execution. Learn more about returning results from your function.

```
{
  "q": "What is AWS Lambda?",
  "a": "AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume - there
is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service -
all with zero administration. Just upload your code and Lambda takes care of everything required to run and scale your code with high
availability. You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile
app."
```

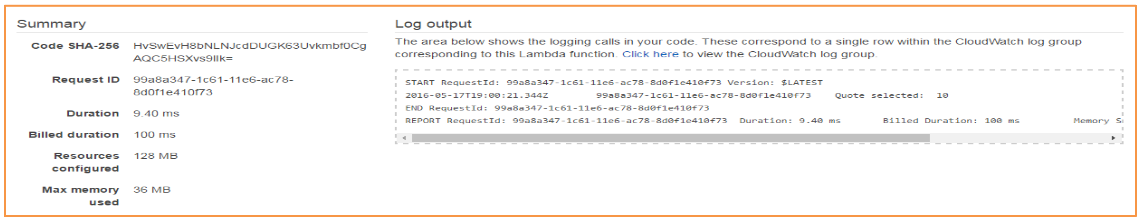
Summary

Code SHA-256	Request ID
oAe89mM0JHCIFh8wMjoIA3J5S4kSAGa0Tv61OVtuEDc=	8469f465-be5c-49db-9e8f-27c67b0294df

Step 7

Verify in the Execution Summary the following parameters

- **Duration**
  - 9,40 ms
- **Billed Duration**
  - 100 ms
- **Max Mem Used**
  - 36 MB



Summary

Code SHA-256	HvSwEvH8bNLNJcdDUGK63Uvkmbf0Cg AQC5HSXvs9lIk=
Request ID	99a8a347-1c61-11e6-ac78-8d0f1e410f73
Duration	9.40 ms
Billed duration	100 ms
Resources configured	128 MB
Max memory used	36 MB

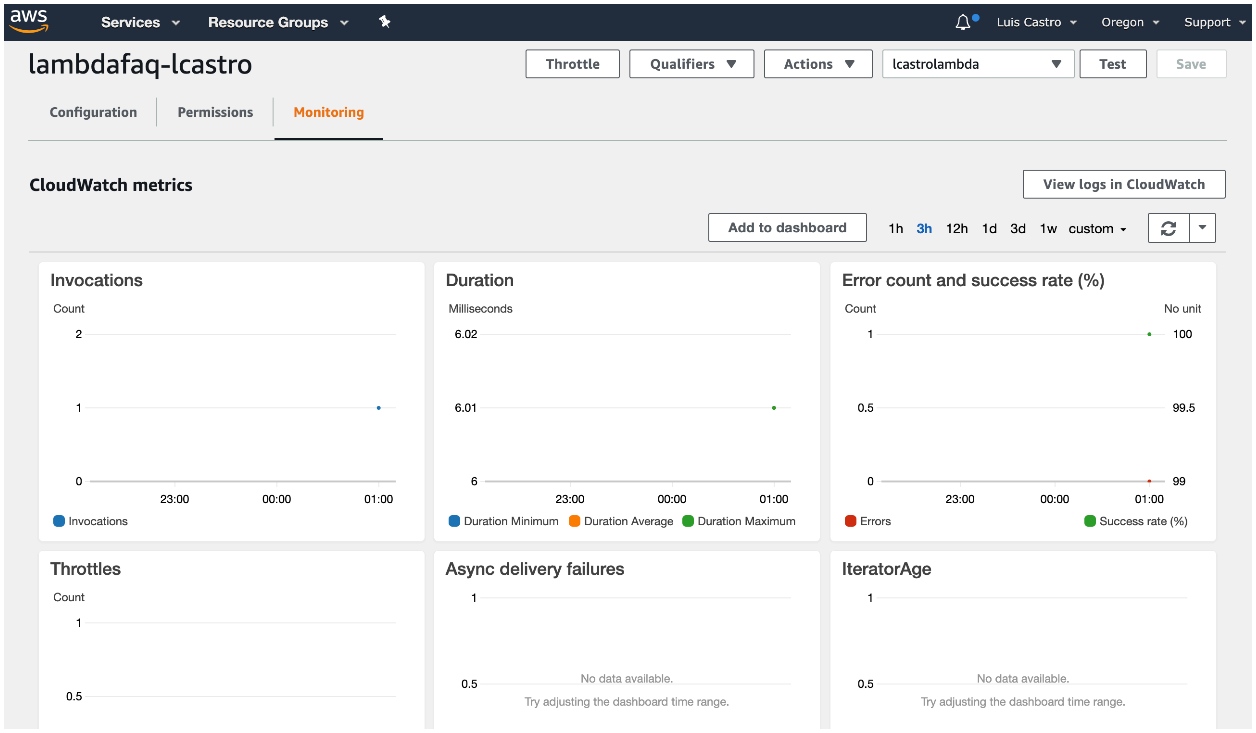
Log output

The area below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. Click here to view the CloudWatch log group.

```
START RequestId: 99a8a347-1c61-11e6-ac78-8d0f1e410f73 Version: $LATEST
2016-05-17T19:00:21.344Z        99a8a347-1c61-11e6-ac78-8d0f1e410f73    Quote selected:  10
END RequestId: 99a8a347-1c61-11e6-ac78-8d0f1e410f73
REPORT RequestId: 99a8a347-1c61-11e6-ac78-8d0f1e410f73  Duration: 9.40 ms       Billed Duration: 100 ms        Memory S
```

Step 8

Click on “Monitoring” and check the CloudWatch alarms



Step 9

Enter the Designer of the created function and click on Add Trigger

Choose the following values:

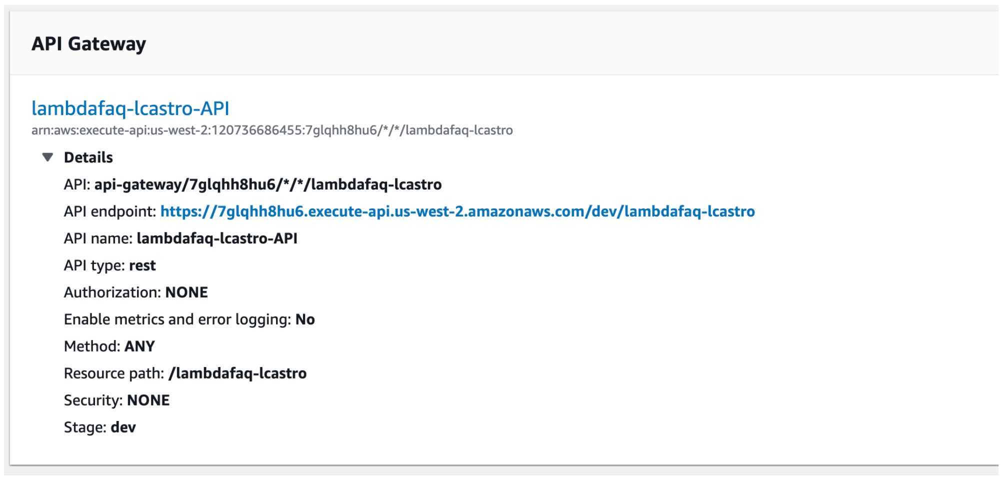
1. **API Gateway**
2. **Create an API**
3. **API Type: REST API**
4. **Security: Open**
5. **Additional Settings**

- API Name: dejar el valor default

- Deployment Stage: default

6. **Add**

Click on the Generated Endpoint API



API Gateway

lambdafaq-lcastro-API
arn:aws:execute-api:us-west-2:120736686455:7glqhh8hu6/*/*/lambdafaq-lcastro

Details

API: **api-gateway/7glqhh8hu6/*/*/lambdafaq-lcastro**
API endpoint: **https://7glqhh8hu6.execute-api.us-west-2.amazonaws.com/dev/lambdafaq-lcastro**
API name: **lambdafaq-lcastro-API**
API type: **rest**
Authorization: **NONE**
Enable metrics and error logging: **No**
Method: **ANY**
Resource path: **/lambdafaq-lcastro**
Security: **NONE**
Stage: **dev**

Step 10

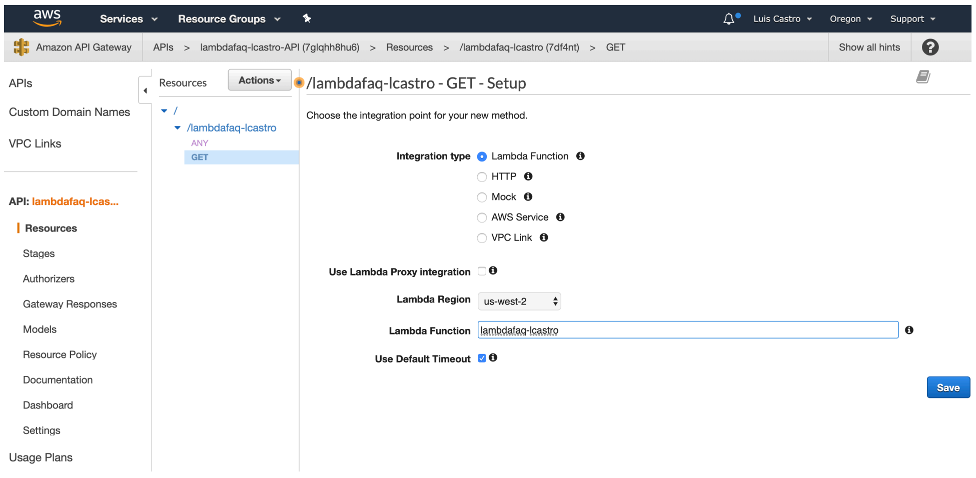
Go to API Gateway service and click on Created API

In Resources choose the lambda function created and click Actions> Create Method

Choose GET and Click on the Check Mark

Choose the following values:

1. Integration Type: **Lambda Function**
2. Use Lambda Proxy Integration: **leave uncheck**
3. Lambda Region: **The region where the Lambda was created**
4. Lambda Function: **Write the name of the Lambda function**
5. Save



/lambdafaq-lcastro - GET - Setup

Choose the integration point for your new method.

Integration type: [✓ Lambda Function] HTTP Mock AWS Service VPC Link

Use Lambda Proxy integration: ☐

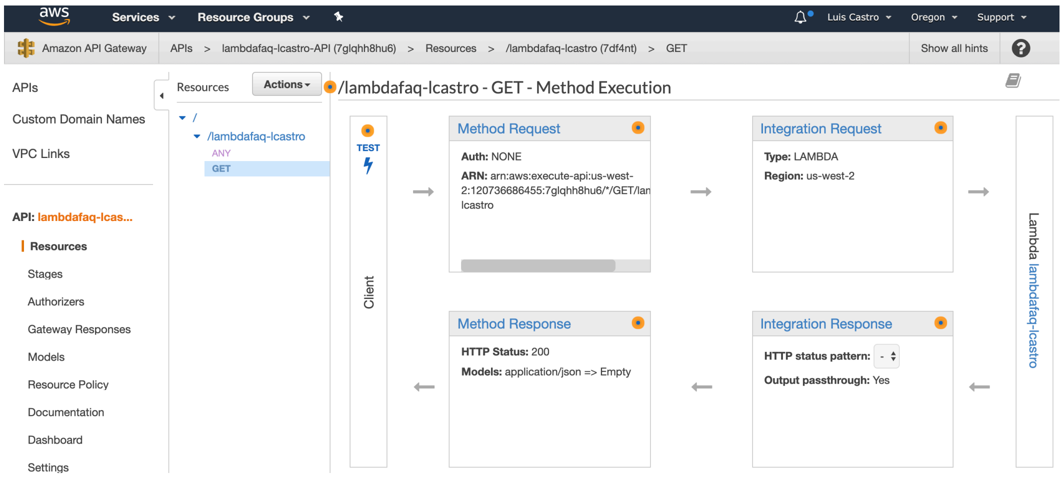
Lambda Region: us-west-2

Lambda Function: lambdafaq-lcastro

Use Default Timeout: ☑

Step 11

Test that the API Gateway responds on the Lambda function, by clicking Test



Auth: NONE
ARN: arn:aws:execute-api:us-west-2:120736686455:7glqhh8hu6/*/GET/lambdafaq-lcastro

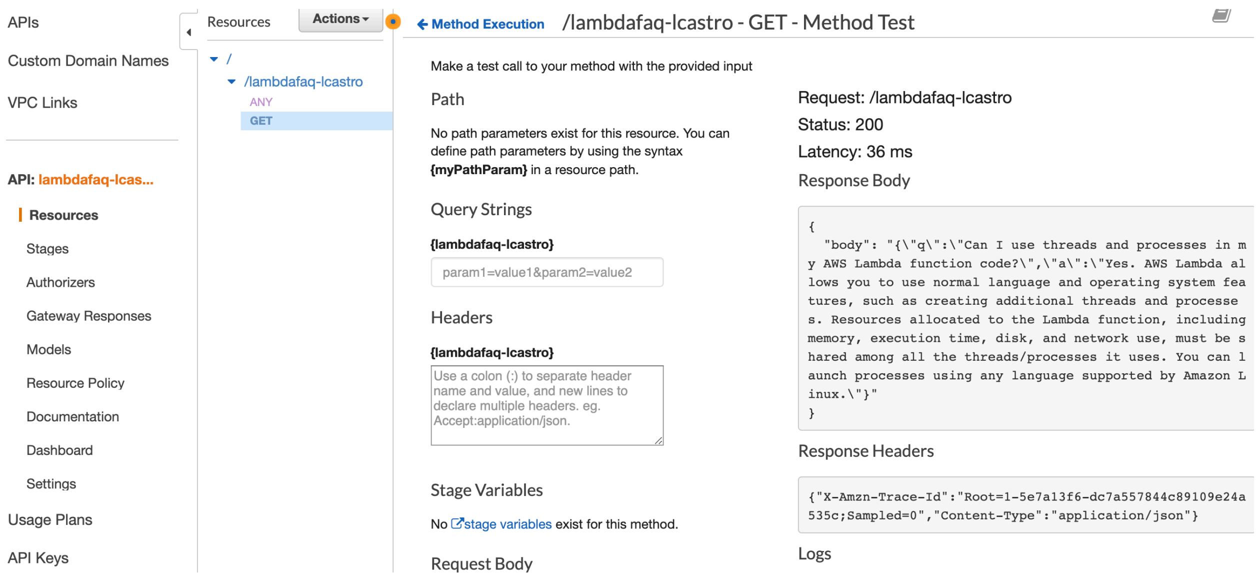
Type: LAMBDA
Region: us-west-2

HTTP Status: 200
Models: application/json => Empty

HTTP status pattern: -
Output passthrough: Yes

Step 12

Validate that the Test returns the expected values of the Lambda function



Make a test call to your method with the provided input

Path

No path parameters exist for this resource. You can define path parameters by using the syntax **{myPathParam}** in a resource path.

Query Strings

{lambdafaq-lcastro}

param1=value1&param2=value2

Headers

{lambdafaq-lcastro}

Use a colon (:) to separate header name and value, and new lines to declare multiple headers. eg. Accept:application/json.

Stage Variables

No stage variables exist for this method.

Request Body

Request: /lambdafaq-lcastro
Status: 200
Latency: 36 ms
Response Body

```
{
  "body": "{\"q\":\"Can I use threads and processes in my AWS Lambda function code?\",\"a\":\"Yes. AWS Lambda allows you to use normal language and operating system features, such as creating additional threads and processes. Resources allocated to the Lambda function, including memory, execution time, disk, and network use, must be shared among all the threads/processes it uses. You can launch processes using any language supported by Amazon Linux.\"}"
}
```

Response Headers

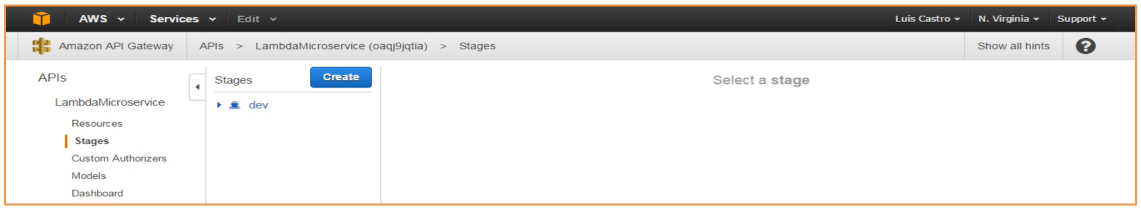
```
{"X-Amzn-Trace-Id":"Root=1-5e7a13f6-dc7a557844c89109e24a535c;Sampled=0","Content-Type":"application/json"}
```

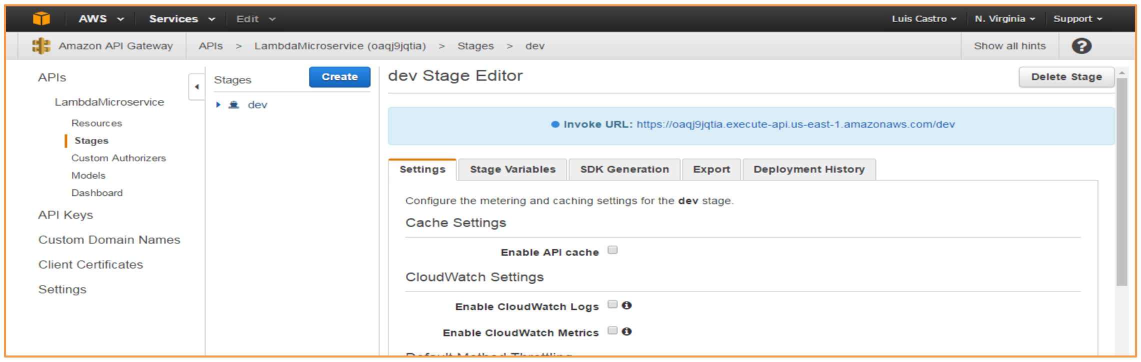
Logs

Step 13

Then in the left menu mark Stages and verify the Invoke URL in a new browser

- It should be similar to the following
  - https://oaqj9jqtia.execute-api.us-east-1.amazonaws.com/dev
  - Note:
    - If you have errors when opening the URL, add the name of the created function (Ex: lambdafaq-lcastro) at the end as follows:
    - https://oaqj9jqtia.execute-api.us-east-1.amazonaws.com/dev/lambdafaq-lcastro





Invoke URL: https://oaqj9jqtia.execute-api.us-east-1.amazonaws.com/dev

Configure the metering and caching settings for the **dev** stage.

Cache Settings

Enable API cache ☐

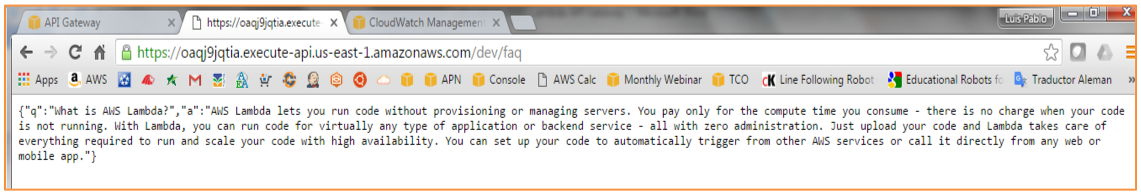
CloudWatch Settings

Enable CloudWatch Logs ☐

Enable CloudWatch Metrics ☐

Step 14

Verify that the URL works properly showing FAQ information, do it several times and validate that each time it shows different information



```
{"q":"What is AWS Lambda?","a":"AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume - there is no charge when your code
is not running. With Lambda, you can run code for virtually any type of application or backend service - all with zero administration. Just upload your code and Lambda takes care of
everything required to run and scale your code with high availability. You can set up your code to automatically trigger from other AWS services or call it directly from any web or
mobile app."}
```