Serverless Compute

In this lecture, we will cover AWS Lambda, a **serverless** computation service provided by AWS that is applied across various verticals today.

[**AWS Lambda**](#_1dnlrfs661x9) **1**

[**Provisioning a Simple Lambda Service**](#_86d7lqstm56p) **2**

# 

# 

# 

# 

# 

# 

# 

# AWS Lambda

From the **AWS Lambda homepage:**

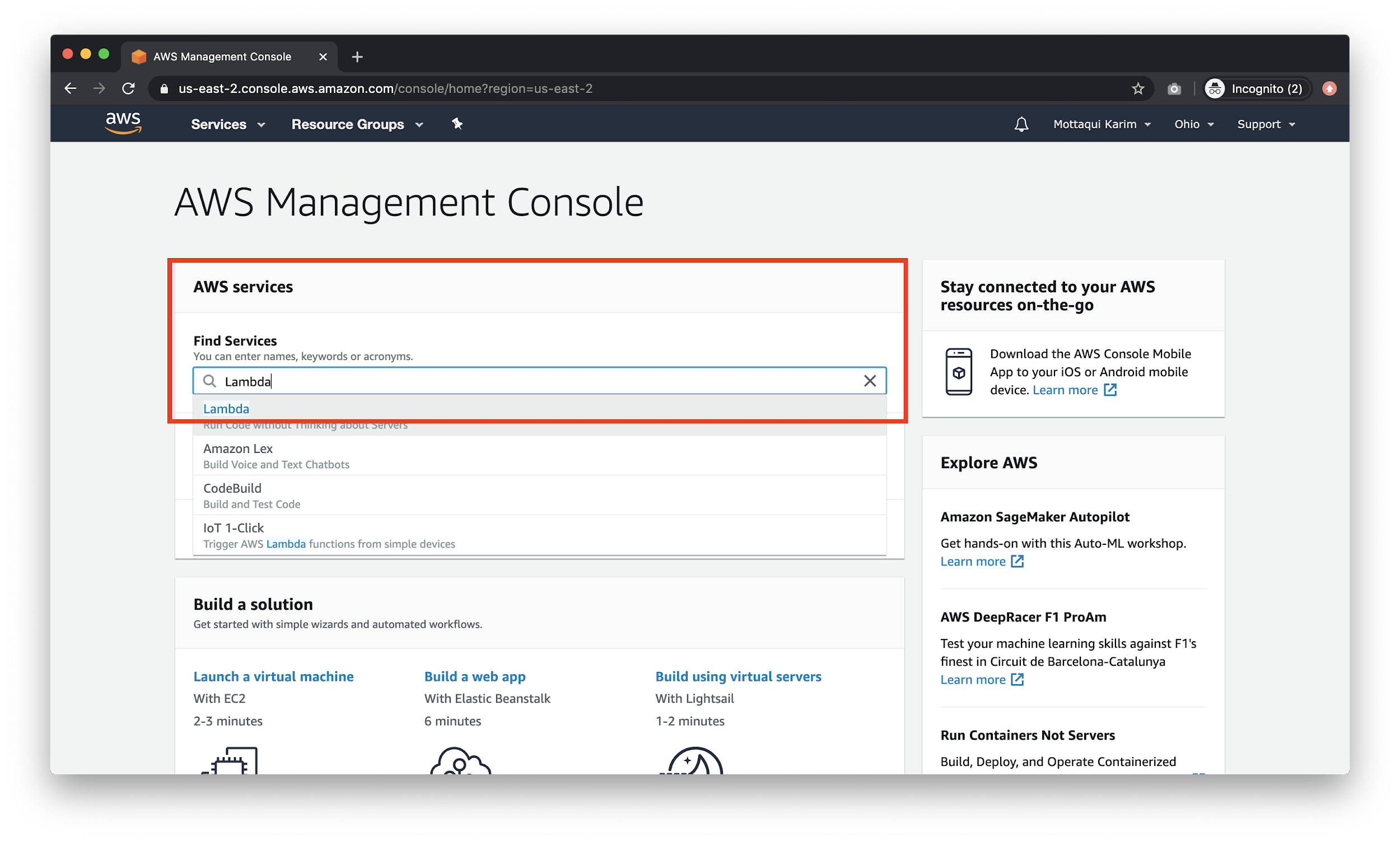
|  |  |
| --- | --- |
|  | *AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume.*  *With Lambda, you can run code for virtually any type of application or backend service - all with zero administration. Just upload your code and Lambda 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.* |

Lambda - and more importantly - serverless compute paradigms are quite powerful in that they allow us to manage **streaming data** (more on that in coming lectures).

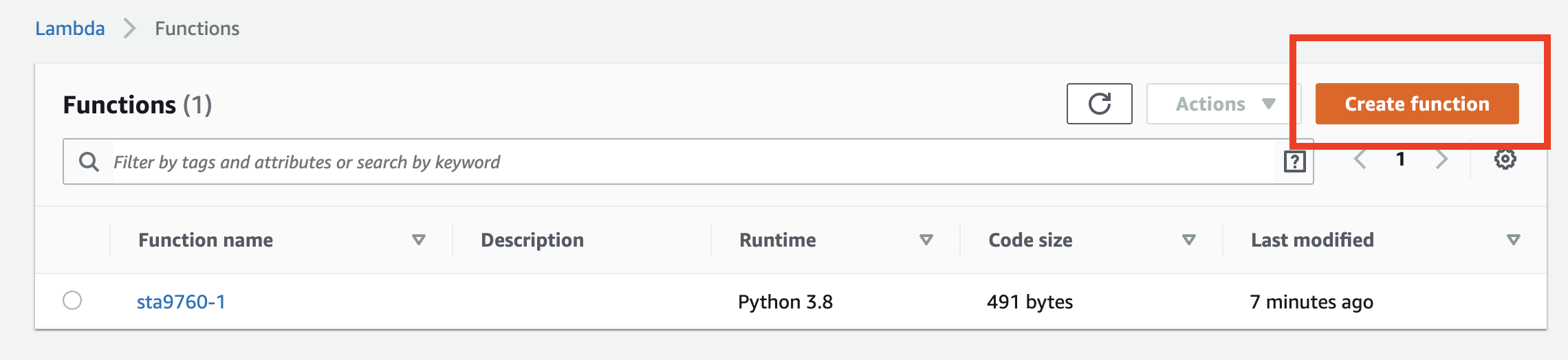
For now, we will discuss how we can get started with AWS Lambda functions and leverage them for our own benefit.

# Provisioning a Simple Lambda Service

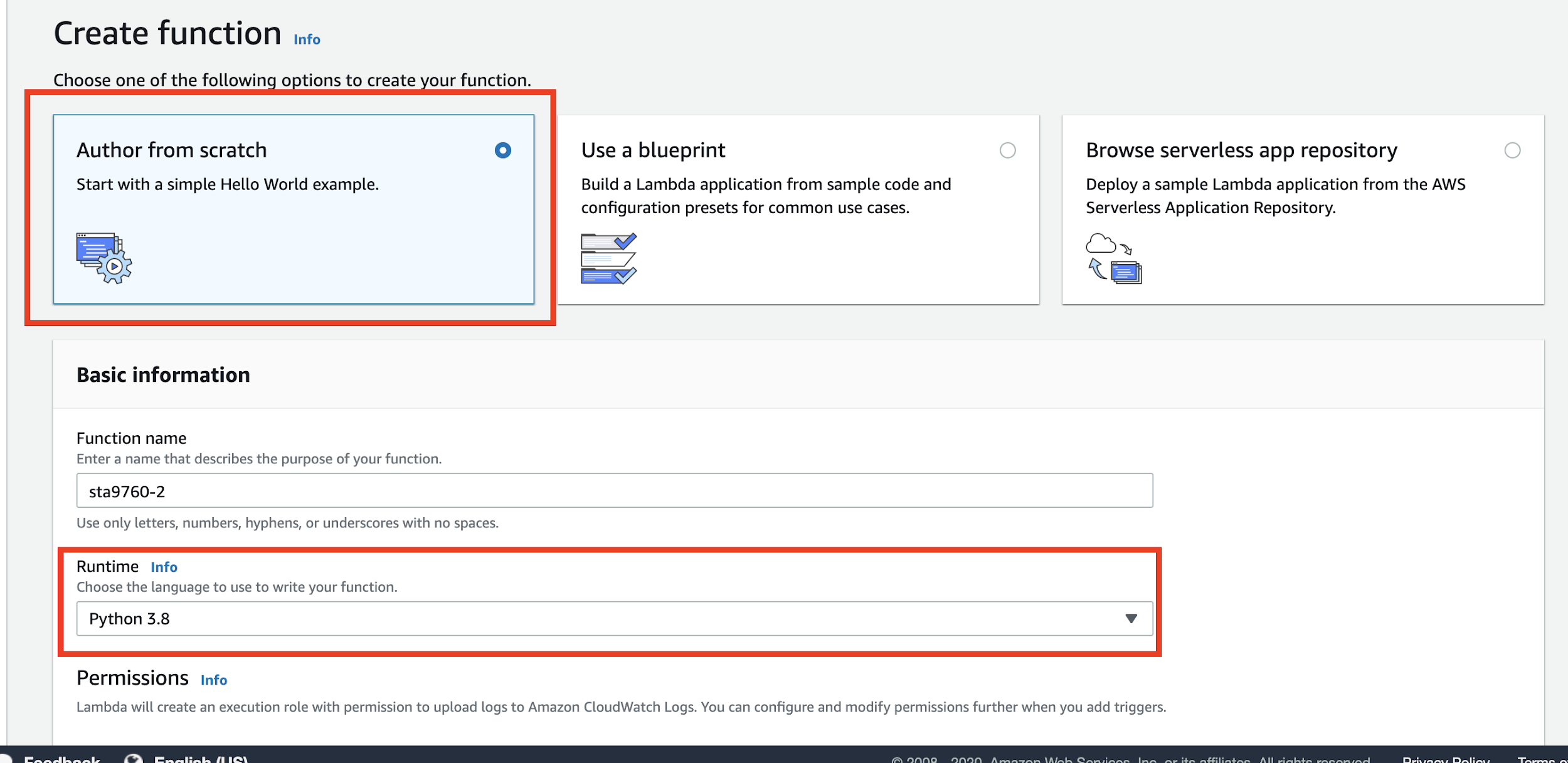
Let’s begin by creating a simple Lambda service for ourselves. Begin by going to AWS Console homepage and typing in “Lambda”.



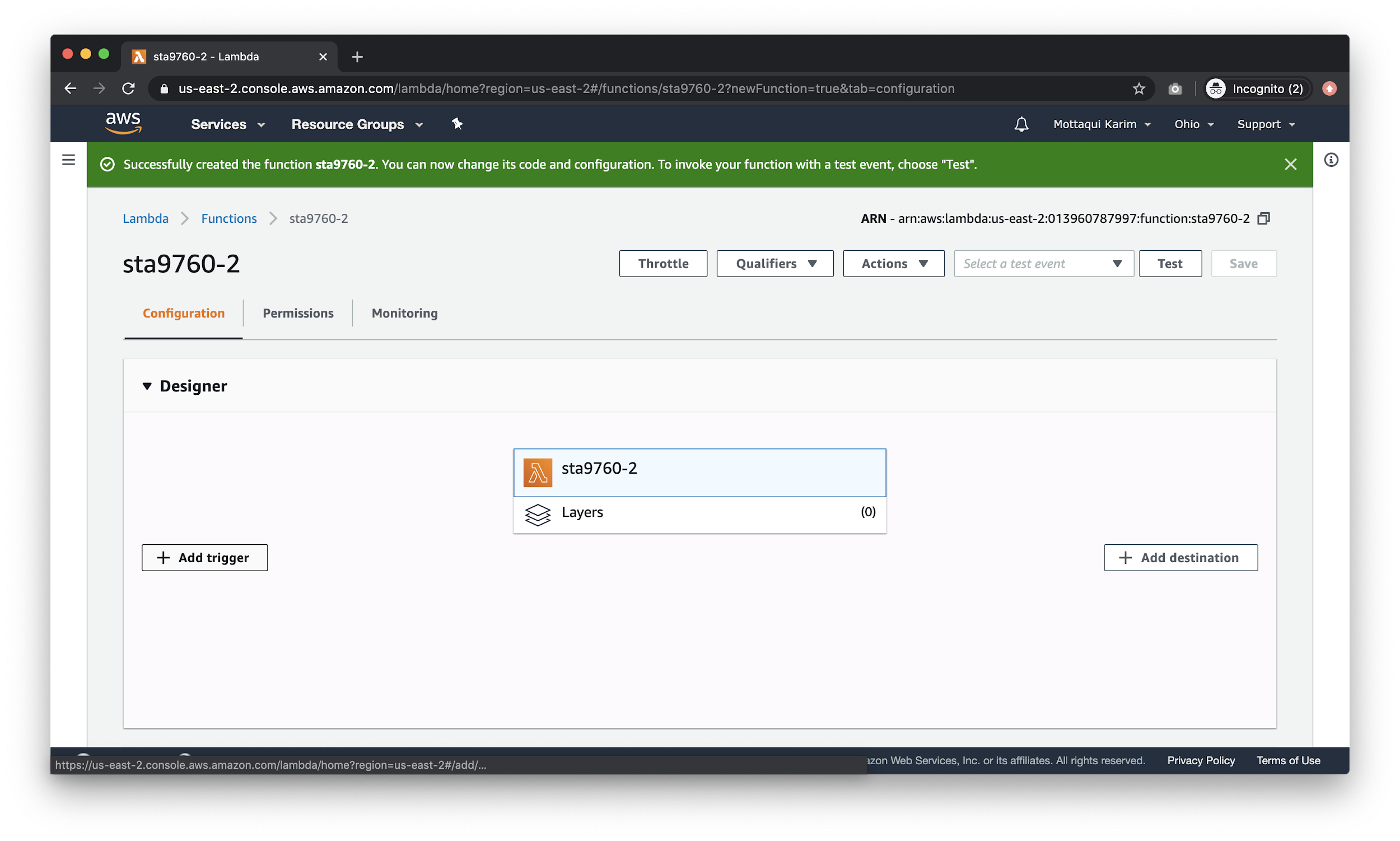
Then, click on the **Create Function** button to go ahead and start the process for making a new lambda function.



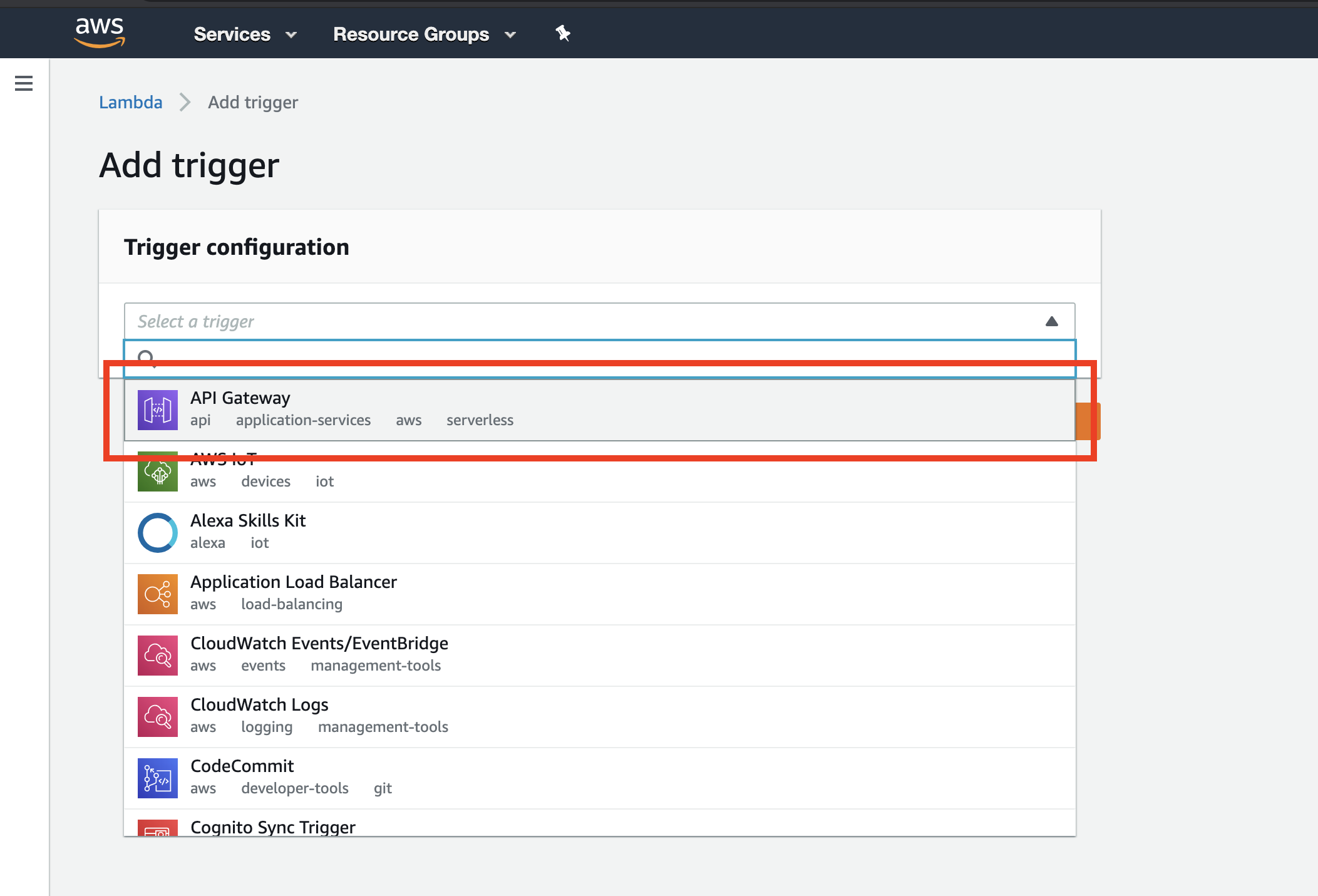
Make sure you have the “blank” item picked and while filling out the fields, ensure you have **Python 3.8** chosen.



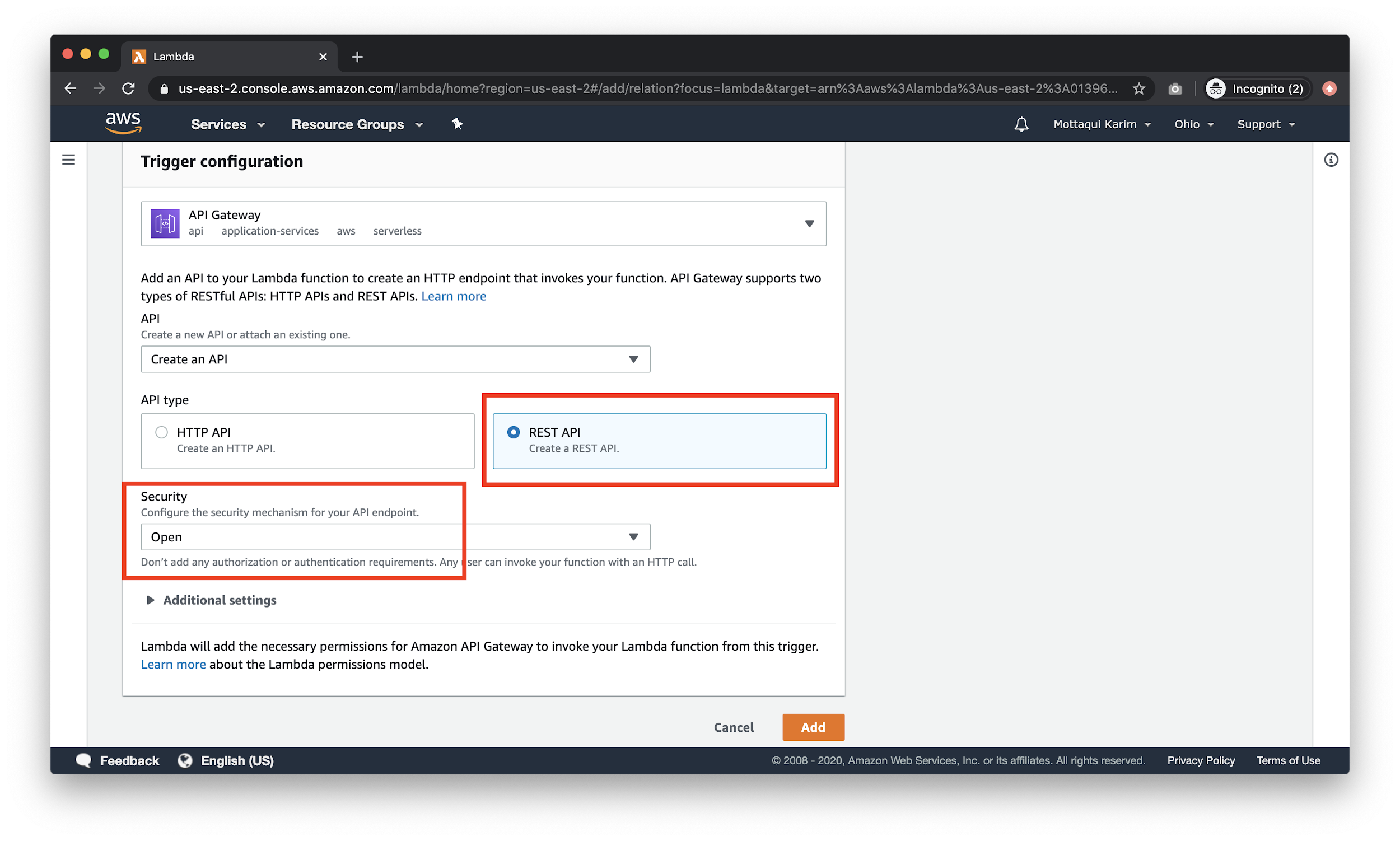
Once you hit the **Create Function** button on the bottom left, expect to be redirected here:



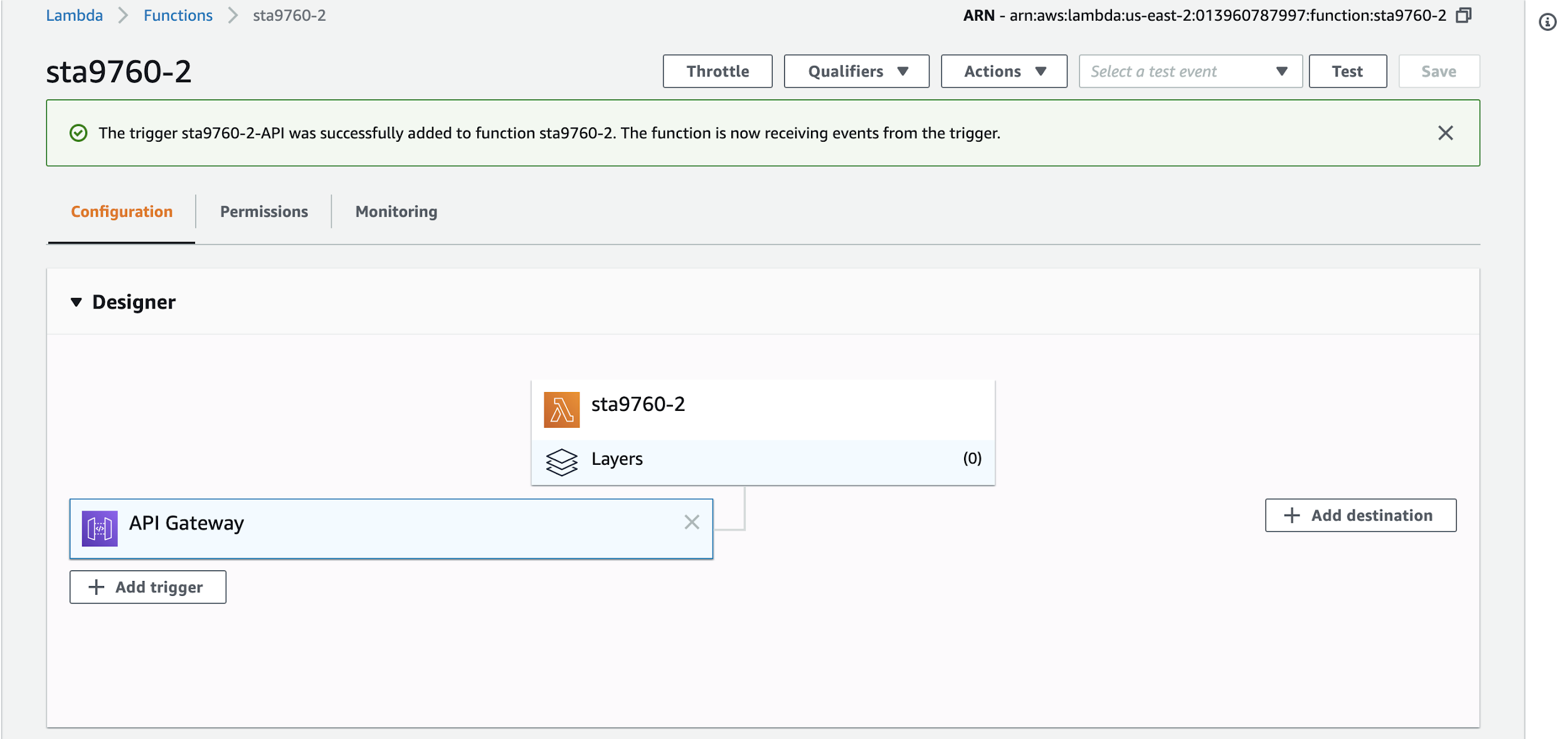
Click on the **Add trigger** button on the left. Then, make sure you choose **API Gateway** as your trigger.



This will give you the following. Make sure you choose **Open** for the **Security** option.

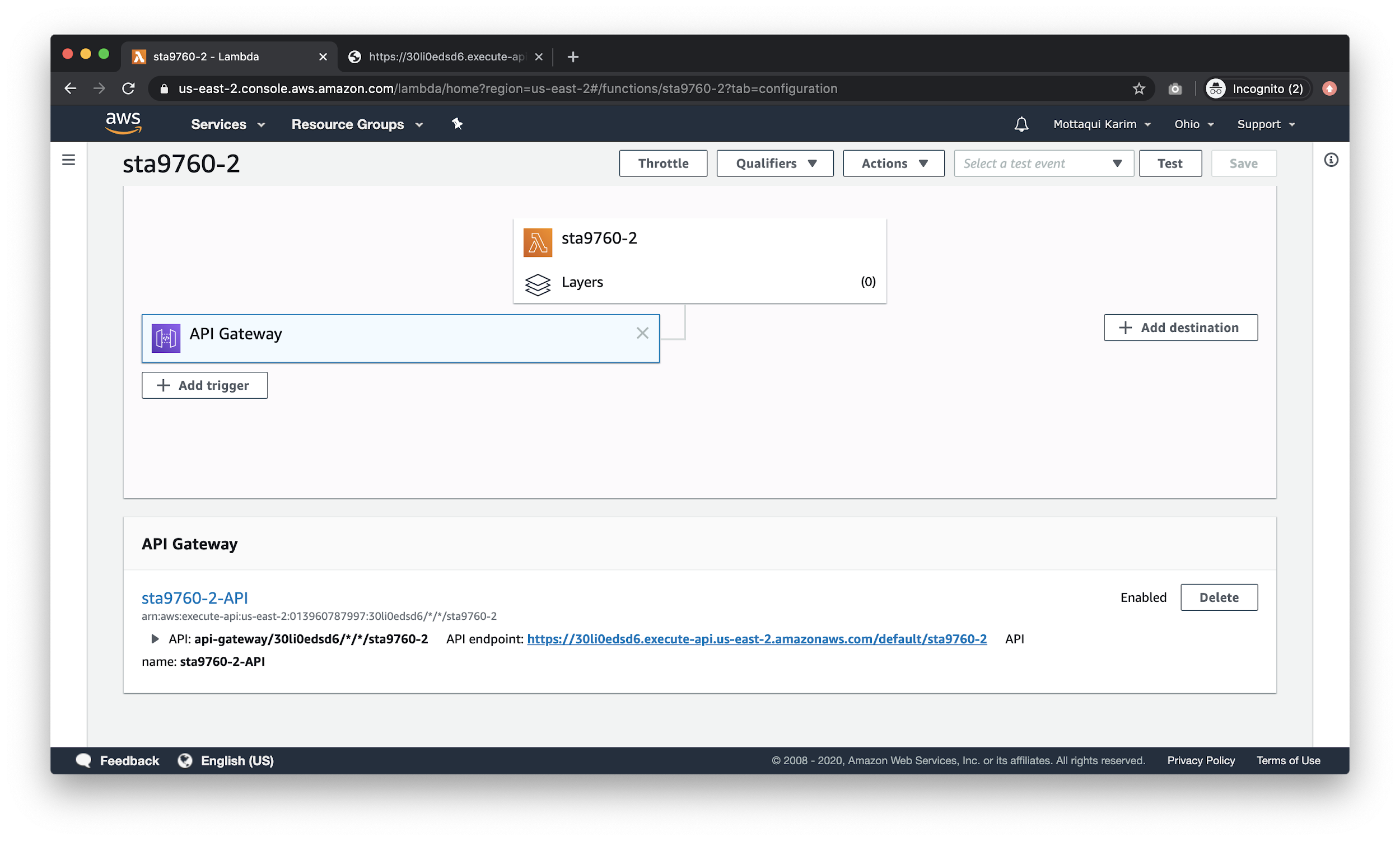


Once you hit **Add**, you should see the following:

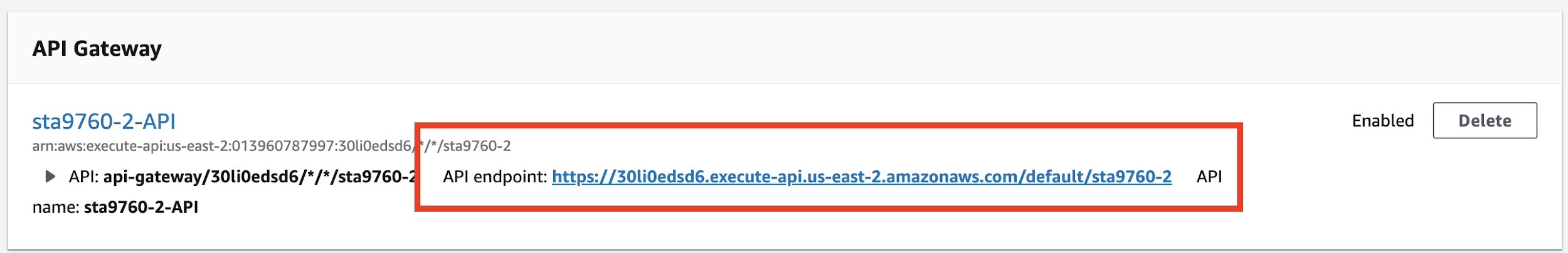


**Awesome**. You now have a Lambda function configured with an API Gateway trigger. **What does this mean**?

Basically, you function will be “run” every time an URL is called, either in the browser or via Python’s requests module or any other way an API endpoint is invoked. Scroll down, you should see something like:



Note:

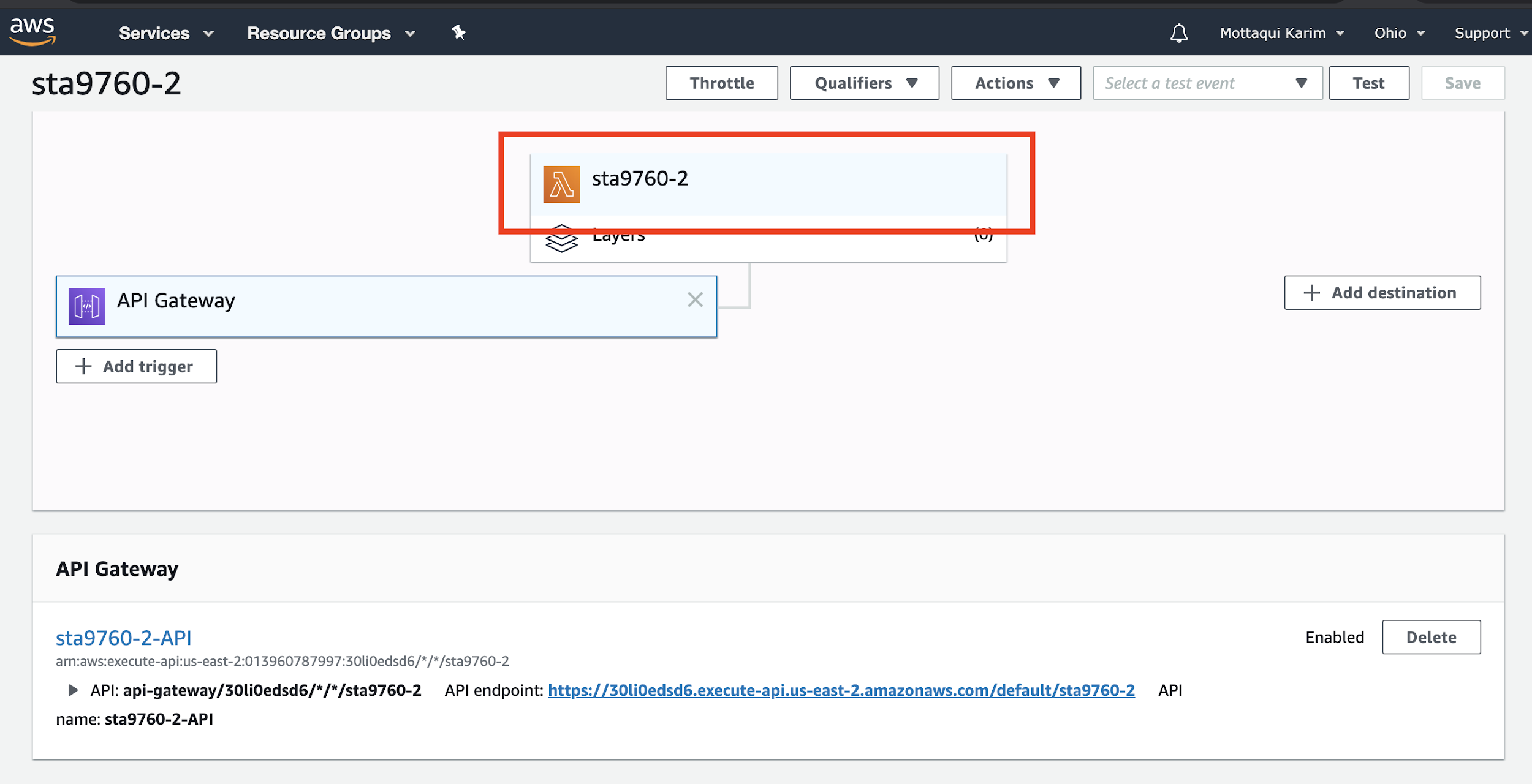


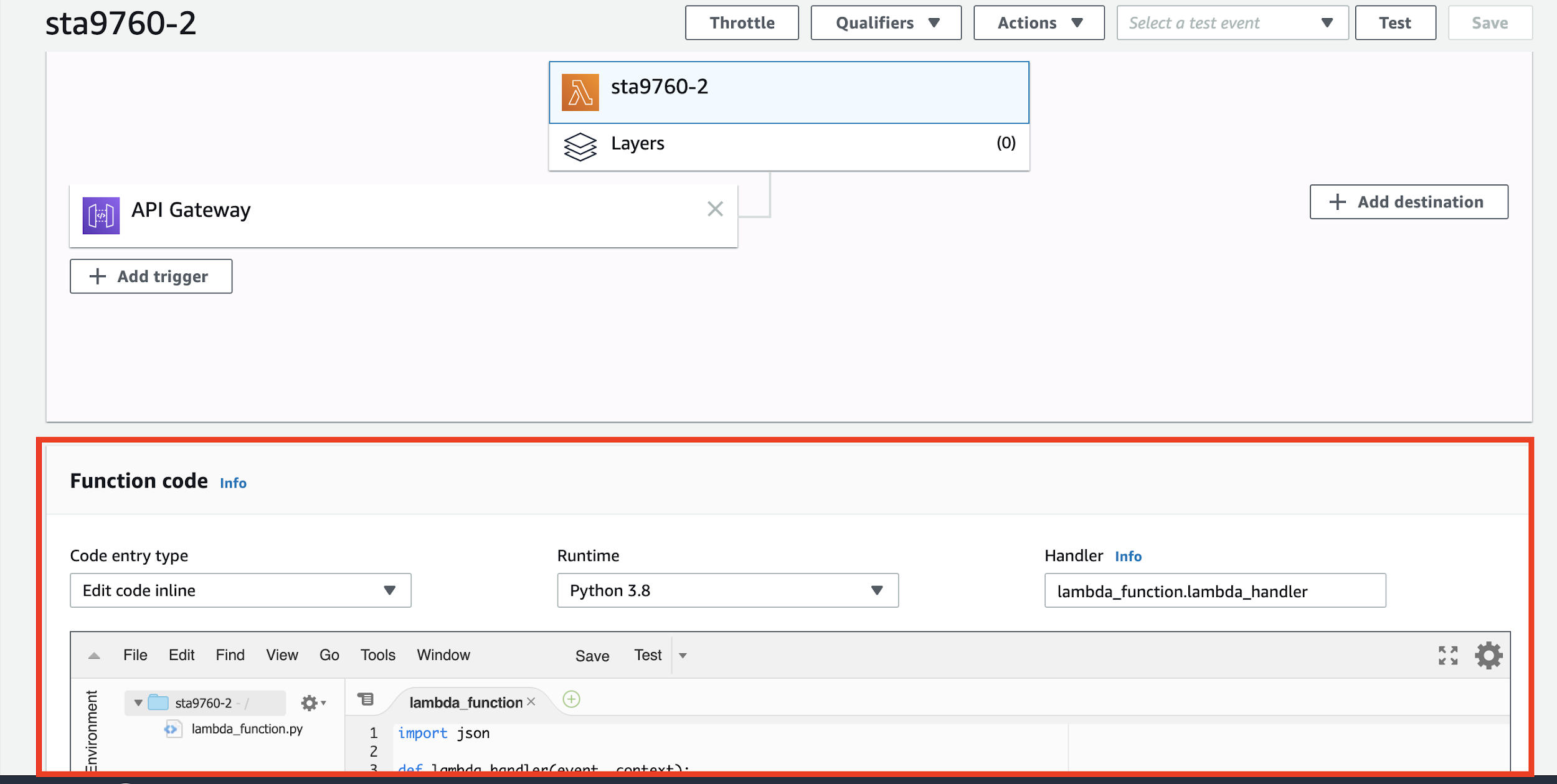
That **URL** is your API endpoint. Click it, you should see something like:



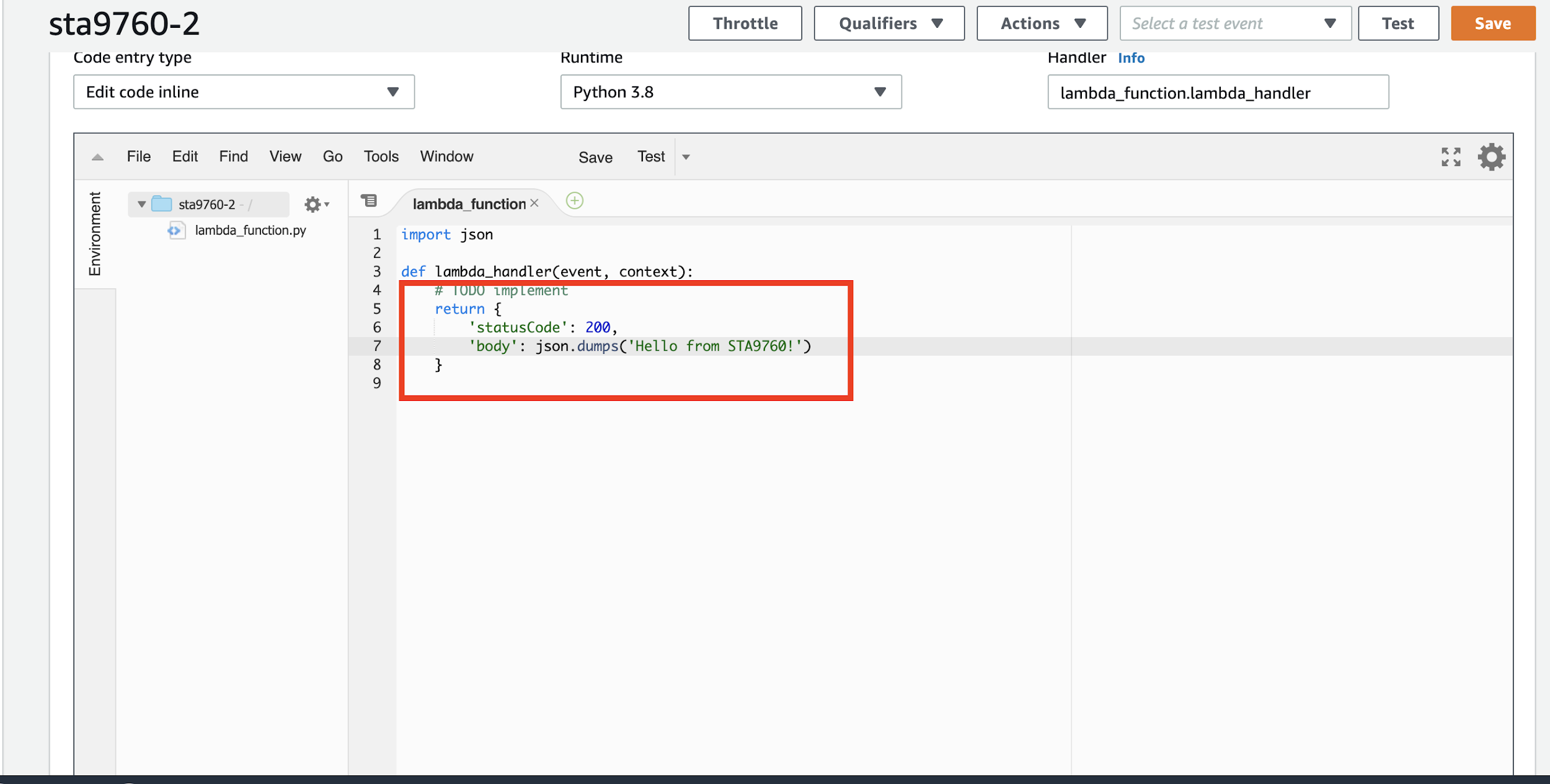
This is your working Lambda function!

Now, let’s update this function. Select the redbox and then scroll down.





You can write and update code for your lambda right there in the browser!



Change some code, hit **Save** and then refresh your link in browser.

