Using Lambda to publish a python script

Contents

[Getting started 2](#_Toc121918987)

[Accessing the management console 2](#_Toc121918988)

[Create a Lambda function to run a python script 3](#_Toc121918989)

[Add an API Gateway trigger so we can see our results on the web 7](#_Toc121918990)

[View your code on the web 9](#_Toc121918991)

# Getting started

You have been sent an email for 2 courses:

* AWS Academy Cloud Foundations. This provides the learning content, which the DAP lists as "homework" starting next week.
* AWS Academy Learner Lab. This provides the console access previously used in AWS Educate

For this activity, we will be using “AWS Academy Learner Lab”

Log into AWS academy using the link provided in the email sent to your student account. The link is probably: [https://awsacademy.instructure.com/](https://awsacademy.instructure.com/courses/4668)

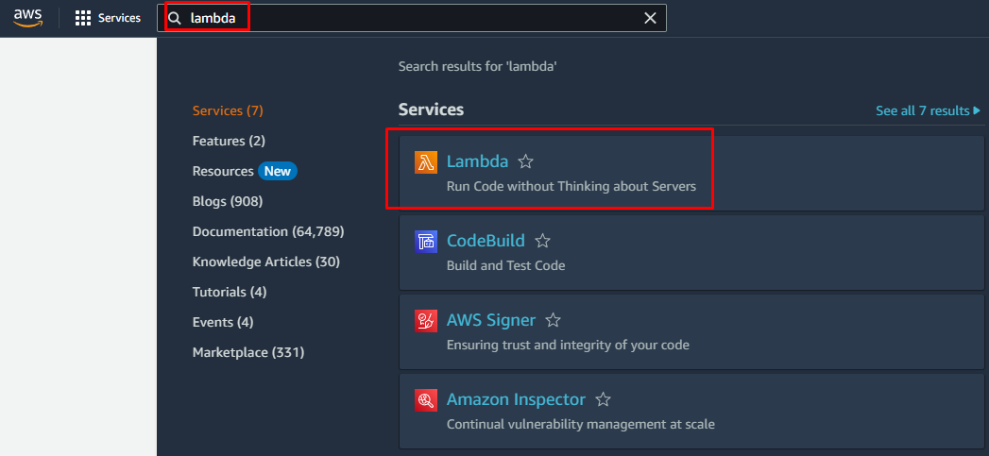
# Accessing the management console

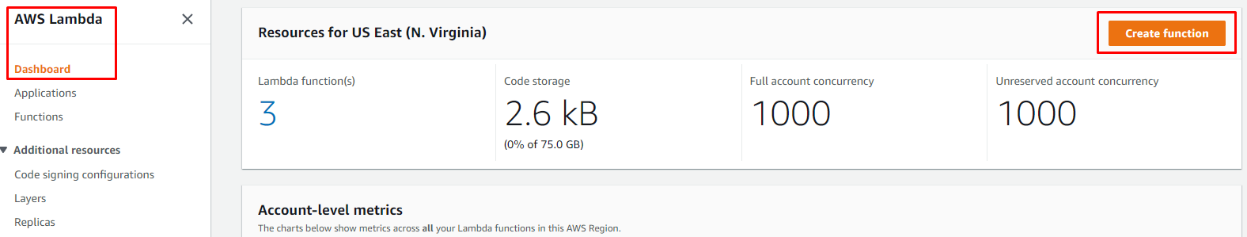
|  |  |
| --- | --- |
| Select the “Modules” hyperlink at the bottom of the page |  |
| Select the “Learner Lab” module |  |
| You should get an agreement. Select “I agree”  If the agreement does not appear, click “Modules” again in the top left. |  |

|  |  |
| --- | --- |
| You are now in the lab environment, ready to start the lab.  Select “Start Lab” |  |
| The lab environment will take a while to power on |  |
| Once powered on, you will get a console similar to the right |  |
| Click the “AWS” link in the top left to access the management console |  |

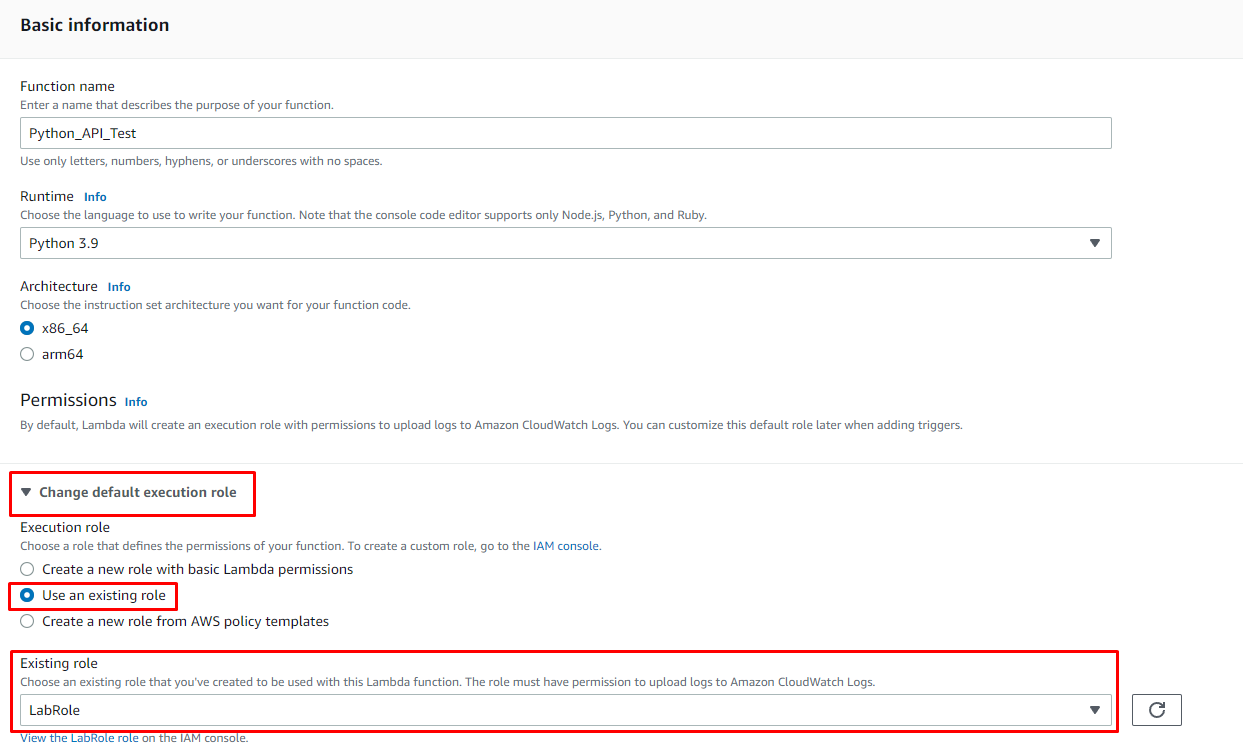
# Create a Lambda function to run a python script

1. In the search bar, type in Lambda and select “Lambda” to enter the Lambda Dashboard



1. Press the orange “Create Function” button
2. You will now be at the Create Function screen. Select Author from scratch and enter the following Basic Information

* Function Name: Python\_API\_Test
* Runtime: Python 3.9
* Architecture: x86\_64
* Permissions:
  + Press the arrow next to Change default execution role
  + Select Use an existing role
  + Press the drop down arrow under Existing role and choose LabRole



1. Scroll to the bottom of the screen and press the orange Create function button

A picture containing graphical user interface

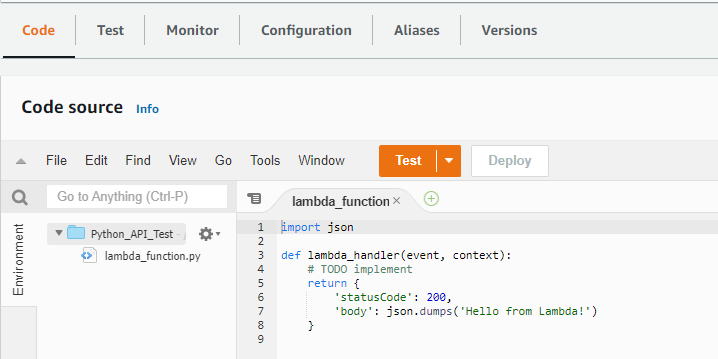
Description automatically generated

1. You should now be taken into your new function called Python\_API\_Test Overview screen

Graphical user interface, text, application

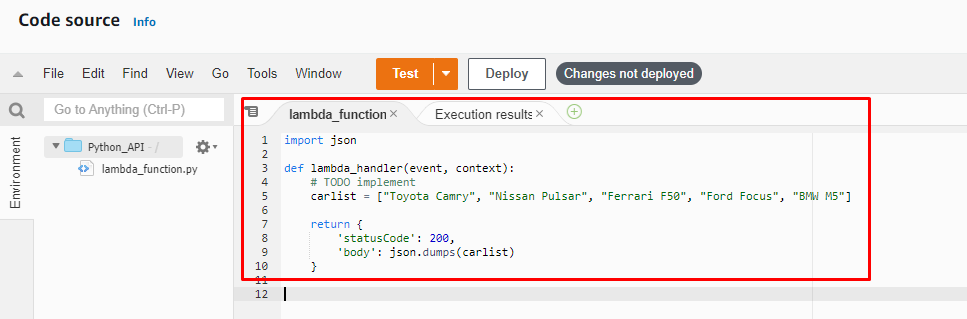
Description automatically generated

1. Ensure the code tab is selected (this will be highlighted orange) and you should have a sample python script called lambda\_function.py available

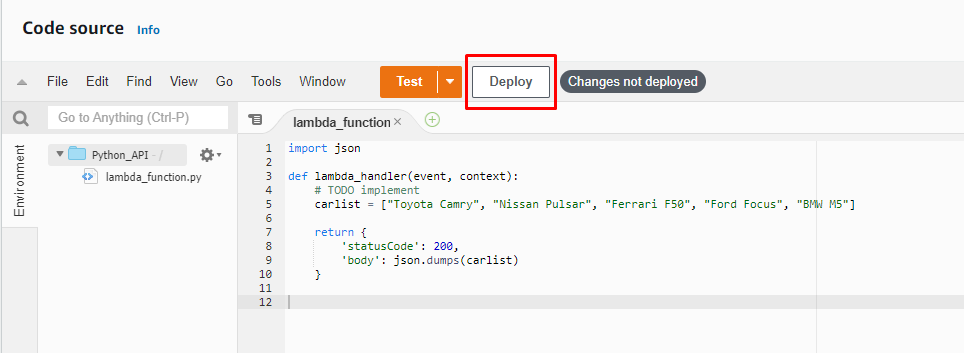


1. Delete the code within the lambda\_function and replace it with the following code (copy and paste from import through to the final })

|  |
| --- |
| import json  def lambda\_handler(event, context):  # TODO implement  carlist = ["Toyota Camry", "Nissan Pulsar", "Ferrari F50", "Ford Focus", "BMW M5"]    return {  'statusCode': 200,  'body': json.dumps(carlist)  } |



1. We will now test our code from within Lambda
   1. Press Ctrl + S on your keyboard from within the lambda\_function code to save your changes
   2. Press the Deploy button



* 1. Press the orange Test button (a new window will open to Configure test event). Use the following settings for your test event
     + Create new event
     + Event name: PythonTest
     + Event sharing settings: Private
     + Template: hello-world
     + Event JSON:

Graphical user interface, text, application, Word

Description automatically generated

* 1. Press the orange Save button

A picture containing graphical user interface

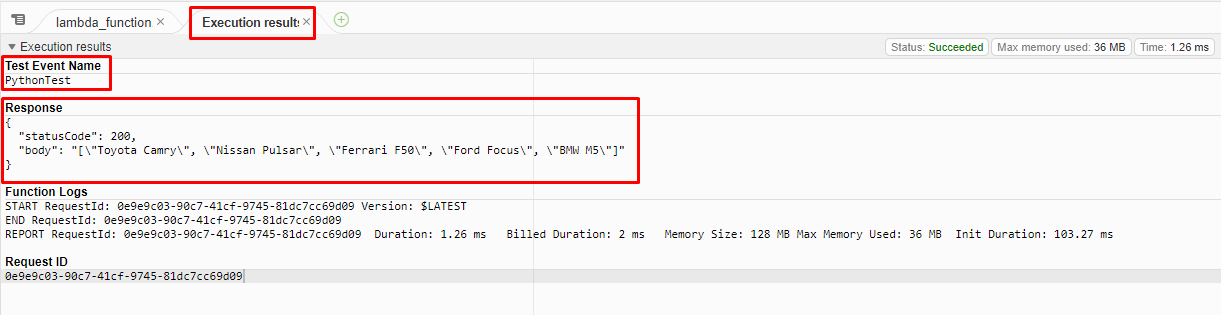
Description automatically generated

* 1. Press the orange Test button again

Graphical user interface, diagram

Description automatically generated

You should now have a new tab called Execution results. If your code is ok, you should see the below output



# Add an API Gateway trigger so we can see our results on the web

1. Within the Function overview, press the + Add trigger button

Graphical user interface, application

Description automatically generated

1. The Add trigger window will open. In the select a source drop-down under Trigger configuration, choose API Gateway

Graphical user interface, application, Teams

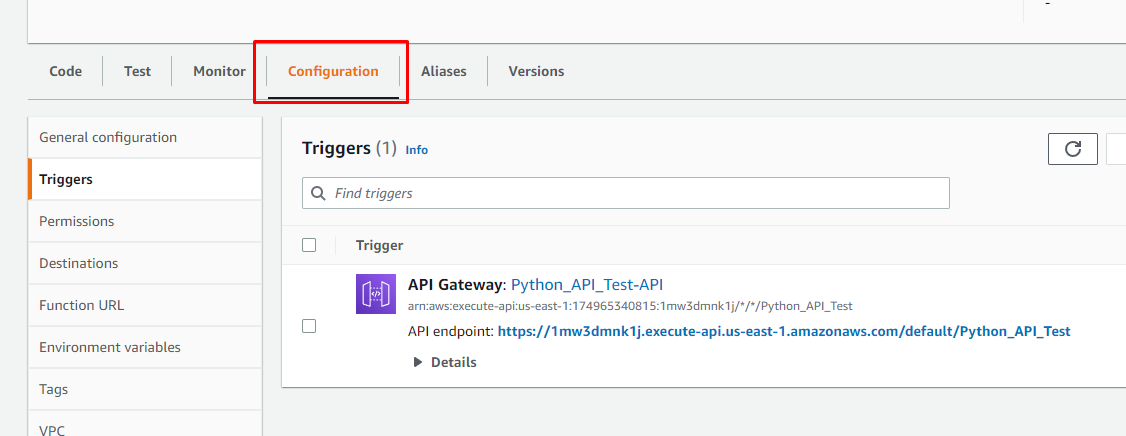
Description automatically generated

1. Select Create a new API under Intent and use the following settings
   * API type: HTTP API
   * Security: Open
2. Press the orange Add button

Graphical user interface, text, application, email

Description automatically generated

You should now be returned to the Function overview page but this time, you will be in the tab Configuration. Check that you have an API Gateway with an API endpoint



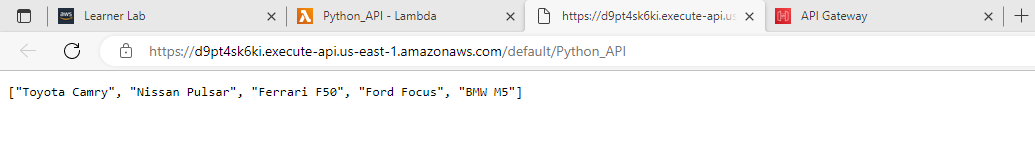
# View your code on the web

1. Press the bold blue API endpoint hyperlink under your API Gateway. A new browser tab should open

Graphical user interface, text, application

Description automatically generated

If everything is set up and working properly, you should see the cars listed from our script



Well done. You have now completed using the AWS Service called Lambda to modify a basic python script and created an API Gateway (another AWS Service) to display this script on the world wide web.