

University of Wolverhampton

School of Mathematics and Computer Science

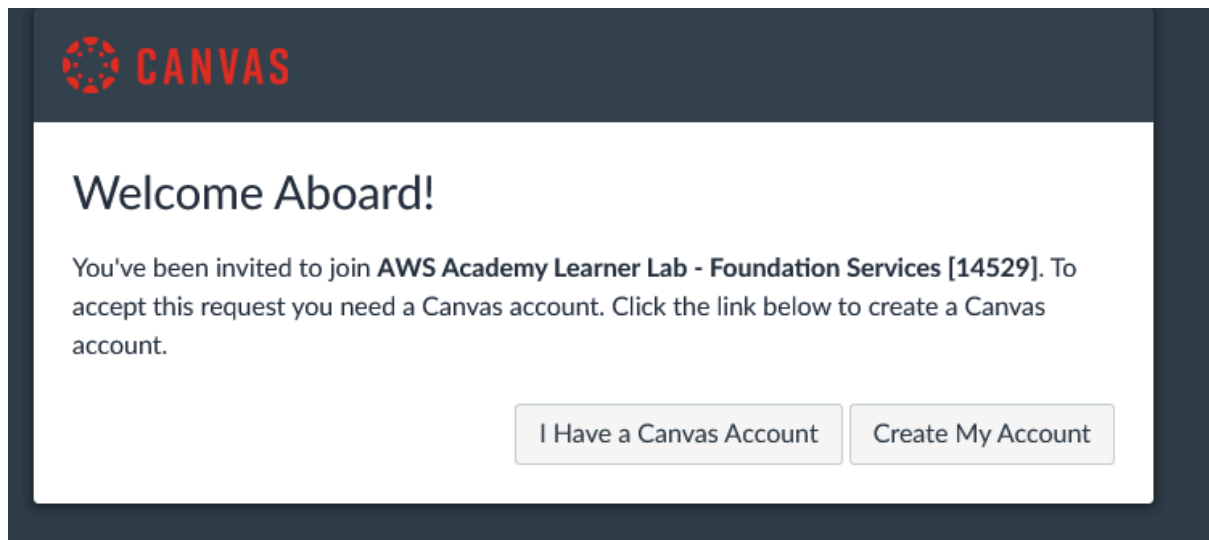
5CS022 Distribute and Cloud Systems Programming Workshop

Overview

In this workshop, you will be introduced to Amazon's AWS cloud service. You will set up your AWS account and try out some of the AWS Cloud Services.

Task - Setting up

Once you have followed your email and click on the AWS Academy link in the invitation email, you will land on the account creation page:



Make sure you click on “Create My Account” to create your account, and eventually you will land on the AWS Academy Canvas Page:

ALLFv1-14529

Account

Dashboard

Courses

Calendar

Inbox

History

Help

Home

Modules

Discussions

View Course Stream

View Course Calendar

View Course Notifications

To Do

Nothing for now

Recent Feedback

Nothing for now

AWS Academy Learner Lab - Foundation Services provides a long-running sandbox environment for ad hoc exploration of AWS services. Within this class, students will have access to a **restricted set of AWS services**. Not all AWS documentation walk-through or sample labs that operate in an AWS Production account will work in the sandbox environment. You will retain access to the AWS resources set up in this environment for the duration of this course. We limit your budget (\$100), so you should exercise caution to prevent charges that will deplete your budget too quickly. If you exceed your budget, you will lose access to your environment and lose all of your work.

Each session lasts for 4 hours by default, although you can extend a session to run longer by pressing the start button to reset your session timer. At the end of each session, any resources you created will persist. However, we automatically shut EC2 instances down. Other resources, such as RDS instances, keep running. Keep in mind that we do not stop some AWS features, so they can still incur charges between sessions. For example, an Elastic Load Balancer or a NAT. You may wish to delete those types of resources and recreate them as needed to test your work during a session. You will have access to this environment for the duration of the class they enrolled you in. When the class ends, your access to the learner lab will also end.

Educator / Teacher Only

If you are an educator using a Learner Lab in your course, see the **Resources** area of the AWS Academy Portal home page for the list of supported services for each Learner Lab class. This sandbox is for educator designed project work, lab exercises, or practice that is created and tested within Learner Lab.

Get Started

Select [Modules](#) to start the course.

Use [Discussions](#) to connect with peers.

Visit [Course Support](#) for help.

Click on Modules and you will see :

ALLFv1-14529 > Modules

Account

Dashboard

Courses

Calendar

Inbox

History

Help

Home

Modules

Discussions

Collapse All

Learner Lab Foundation Services

Learner Lab - Student Guide.pdf

Learner Lab - Foundational Services

End of Course Feedback Survey

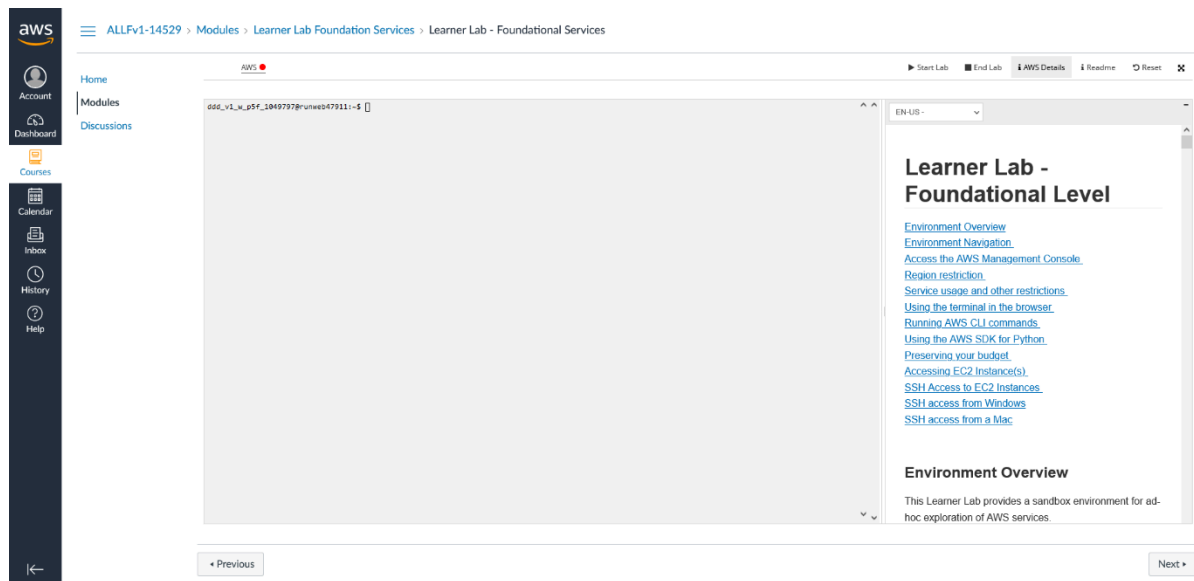
Then click on “Learner Lab - Foundational Services”. The first time you do this, you will be asked to agree to an agreement:

The screenshot shows the Vocareum website interface. On the left is a dark sidebar with navigation links: Home, Modules, Discussions, Courses, Calendar, Inbox, History, and Help. The main content area has a dark header with the Vocareum logo and navigation links: Home, Modules, Discussions, My Classes, Help, and a user profile icon. Below the header, a blue banner reads: "Please read the terms and conditions shown below and click on the 'I agree' button at the bottom of this page to continue." The main content is titled "Terms and Conditions" and includes a welcome message, a paragraph about the functionality of the services, and sections for "1. Agreement to Terms" and "2. Changes to Terms or Services". At the bottom of the page, there are "Previous" and "Next" buttons.

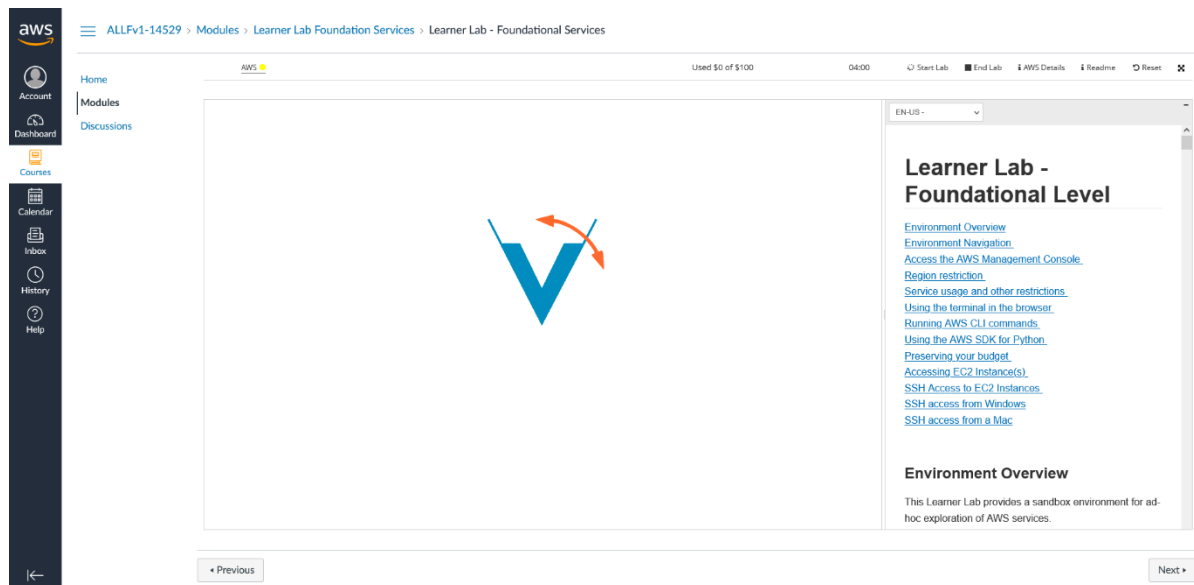
Scroll down and click “Agree”:

This screenshot shows the same Vocareum interface as the previous one, but scrolled down to the "16. General Terms" section. The "Agree" button is now visible at the bottom of the page, below the "Contact Information" section. The "Previous" and "Next" buttons are still present at the very bottom.

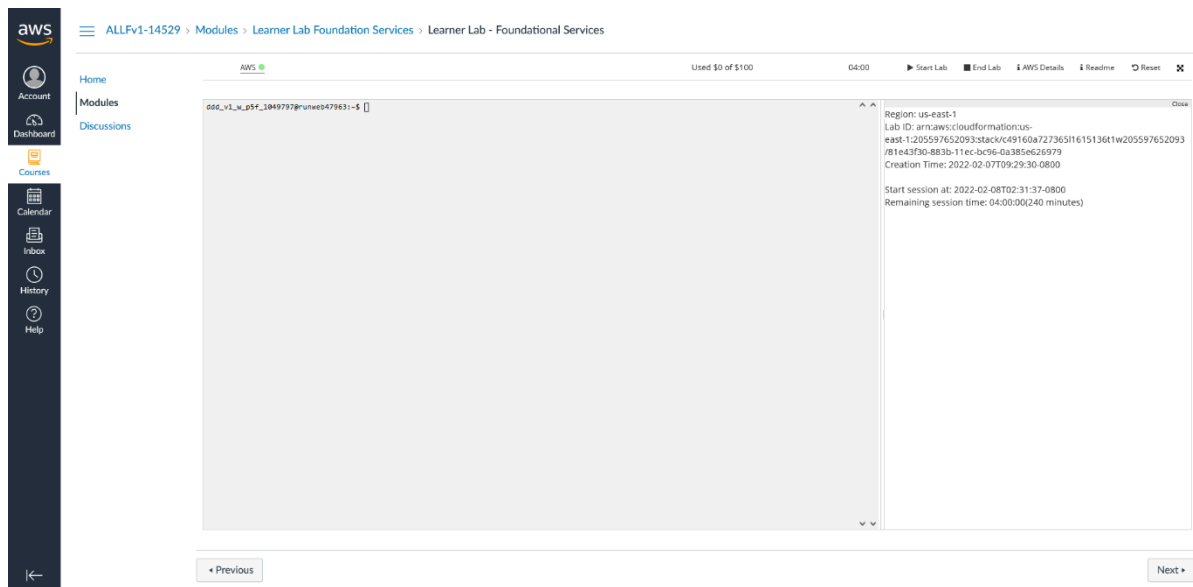
Then you are taken to the normal screen that you will see from now on:



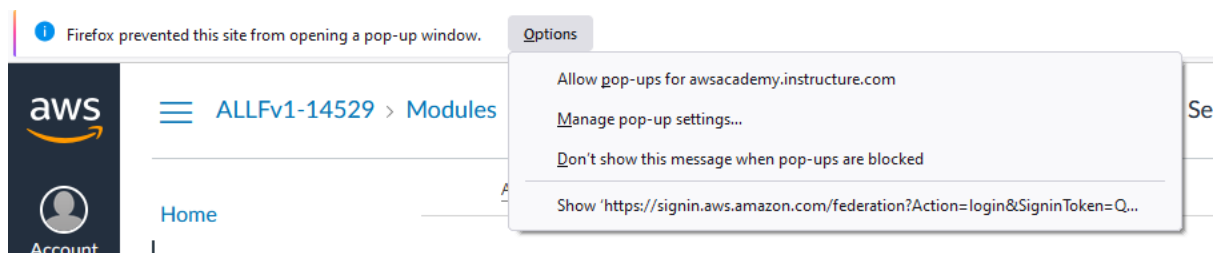
To get started, click on “Start Lab”. Then you will see the AWS lab/software starting up:



This may take a few minutes. When it is ready, you will see this rather uninteresting page:



Click on the AWS link near the top; The one with the green dot. On some web browsers, it might tell you that it stopped a pop-up:




Make sure that you allow the popups.

To allow popups on Google Chrome go on **Settings** click on **Privacy and security** click on **Site settings** (at the bottom) click on **Pop-ups and redirects** click on **Add** button at the bottom and paste the link “https://awsacademy.instructure.com” (or just copy and paste from the webpage).

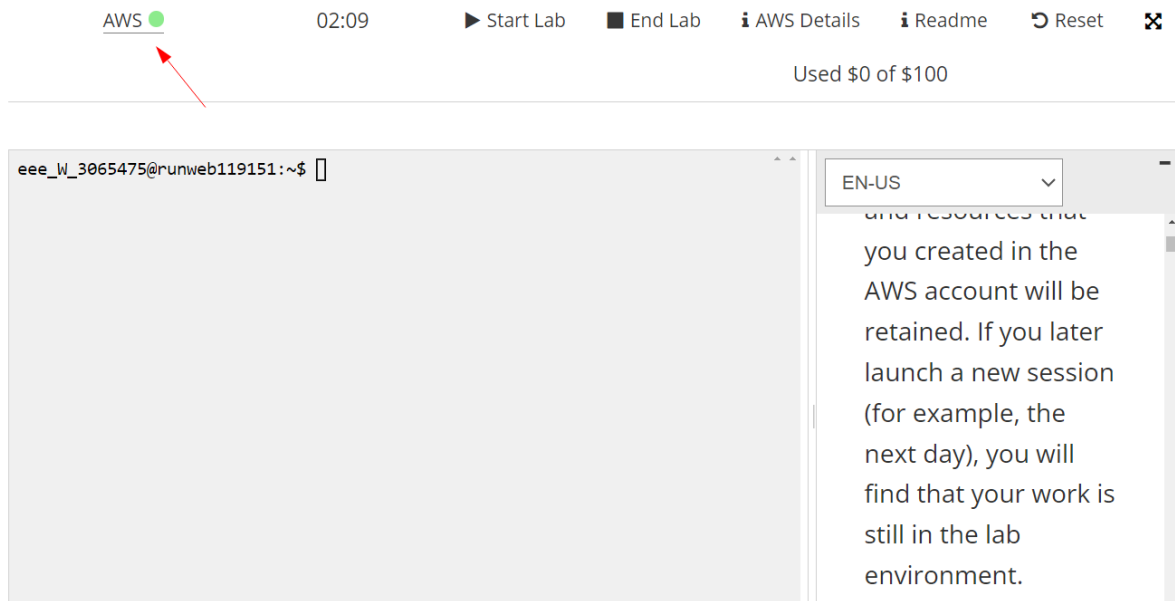
Allowed to send pop-ups and use redirects

Add

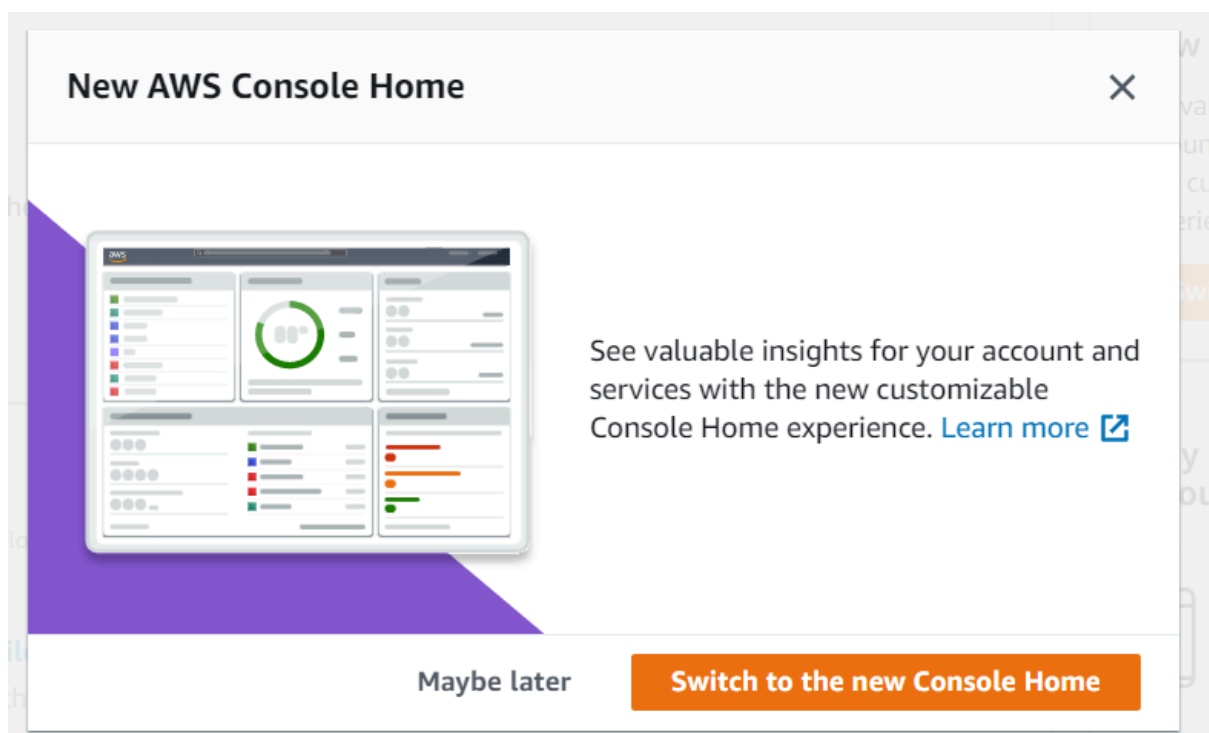
 https://awsacademy.instructure.com



Now try to click on the AWS link again (it should appear green).

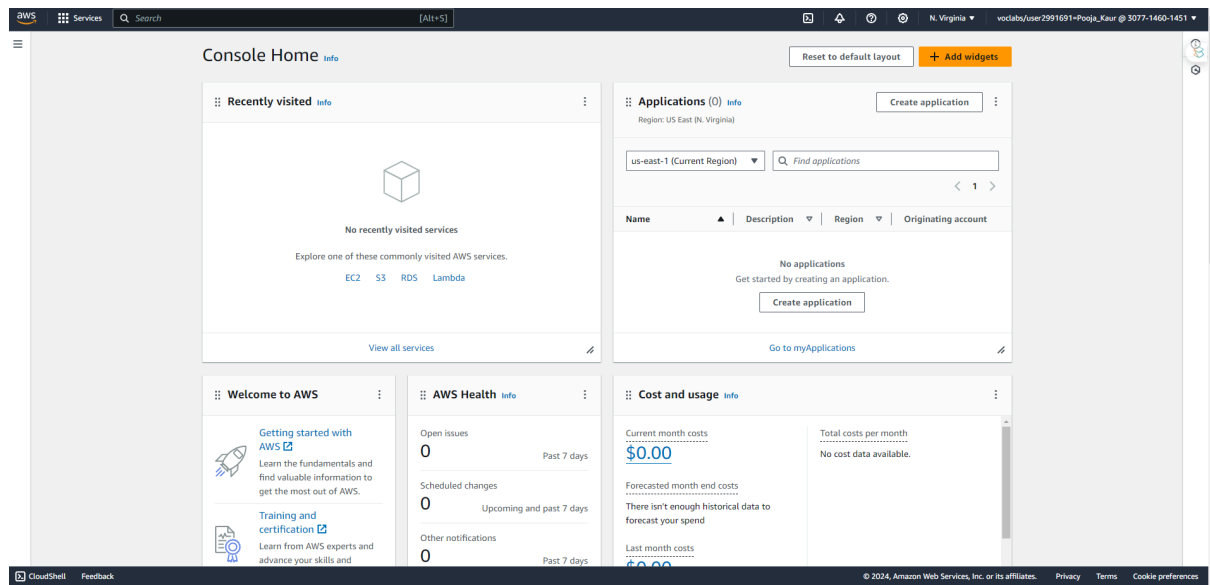


At this point, you may also be offered the chance to try out AWS's new user interface:



I haven't tried it yet, and thus, all screenshots that you will see in this and the following workshops will be based on the current user interface that I am using, and may look different if you try the new "Console Home". So, I suggest you choose "Maybe Later".

Next you will see the AWS Console. This is where you will interact with AWS properly:



Congratulations, you are now set up and ready to use AWS services to learn about Cloud Computing.

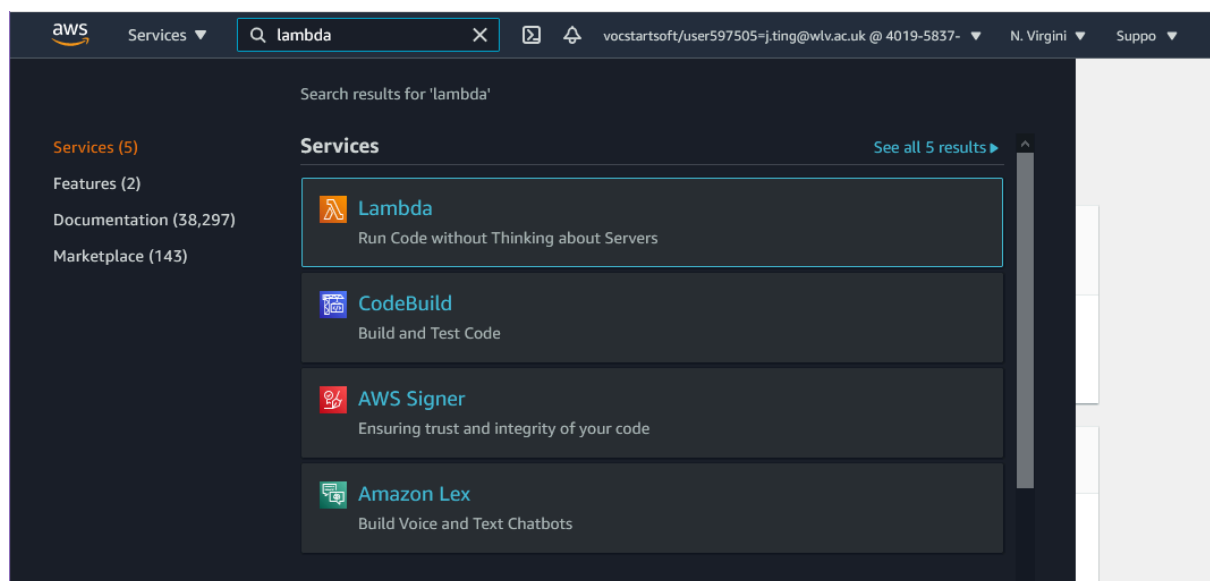
Task - Creating a AWS Lambda "Hello World" Program

Overview

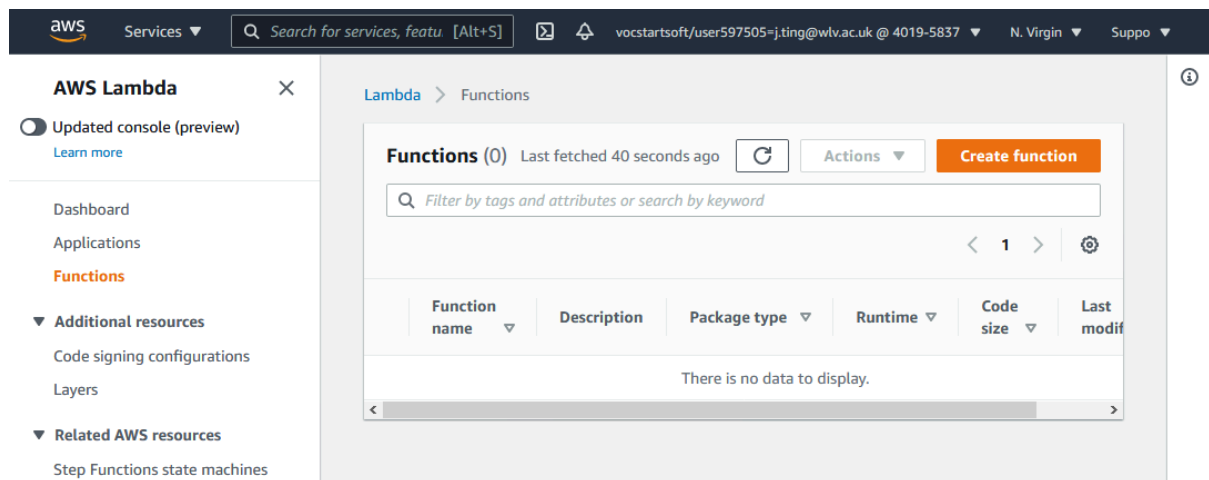
In this section, you will learn the basics of running code on AWS Lambda without setting up or managing servers. You will go through how to create a Hello World Lambda function using the AWS Lambda console. You will learn how to manually invoke the Lambda function using sample event data and review your output metrics.

Step 1: Enter the Lambda Console

In your AWS Console. Search for "Lambda":



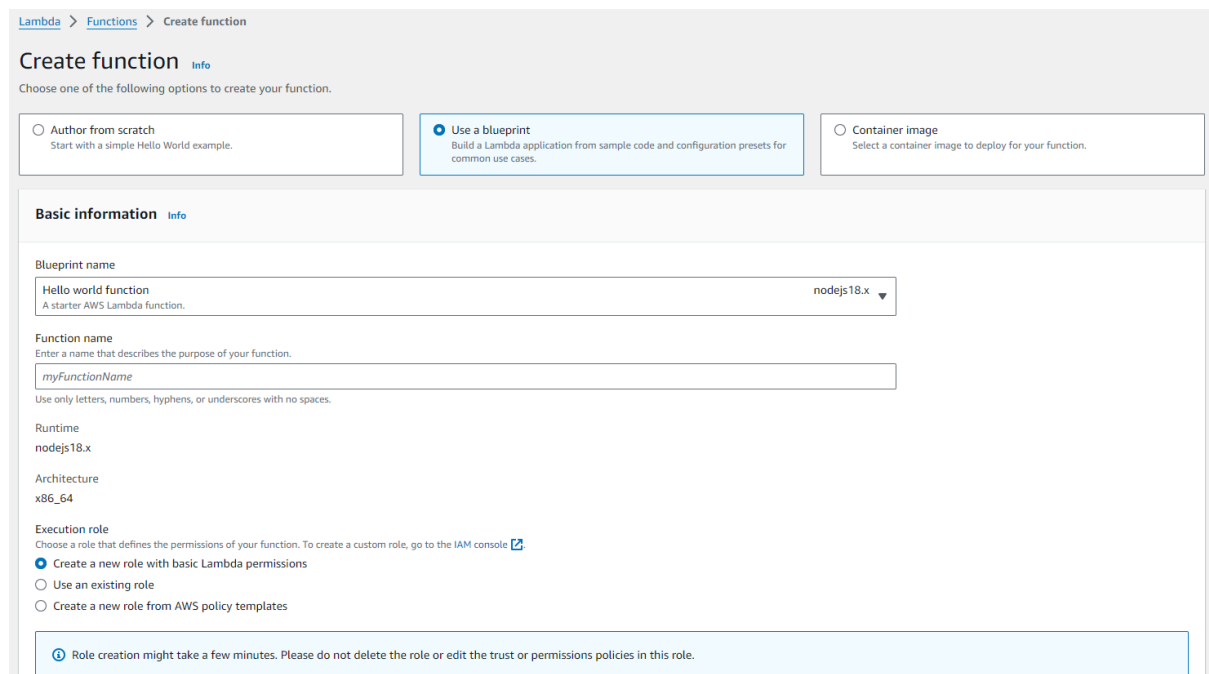
and click it to open the AWS Lambda Console:



Step 2: Lambda Blueprints

Blueprints provide example code to do some minimal processing. Most blueprints process events from specific event sources, such as Amazon S3, DynamoDB, or a custom application.

a. In the AWS Lambda console, select Create Function,



b. Select Blueprints.

c. In the Filter box, type in hello-world-Function and select the Python version:

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.

☐ Container image
Select a container image to deploy for your function.

Basic information [Info](#)

Blueprint name

Hello world function
A starter AWS Lambda function. nodejs18.x ▲

Q Hello X

Get started

Hello world function A starter AWS Lambda function.	nodejs18.x ✓
Hello world function A starter AWS Lambda function.	python3.10

Architecture
x86_64

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

③ Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Step 3: Configure and Create Your Lambda Function

A Lambda function consists of code you provide, associated dependencies, and configuration. The configuration information you provide includes the compute resources you want to allocate (for example, memory), execution timeout, and an IAM role that AWS Lambda can assume to execute your Lambda function on your behalf.

- a. You will now enter Basic Information about your Lambda function:
- Name: You can name your Lambda function here. For this tutorial, enter **hello-world-python**.
 - Role: Select Use an existing role and select **LabRole**

Basic information [Info](#)

Blueprint name

Hello world function
A starter AWS Lambda function.

python3.10 ▼

Function name

Enter a name that describes the purpose of your function.

hello-world-python

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

Runtime

python3.10

Architecture

x86_64

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

Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

LabRole ▼

[View the LabRole role](#) on the IAM console.

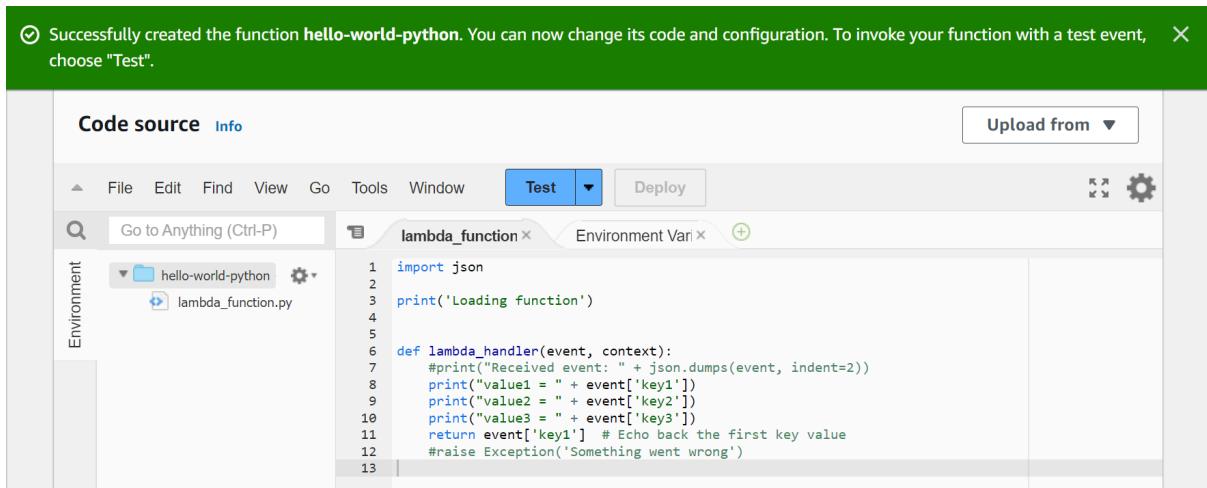


b. Go to the bottom of the page and click **Create Function**:

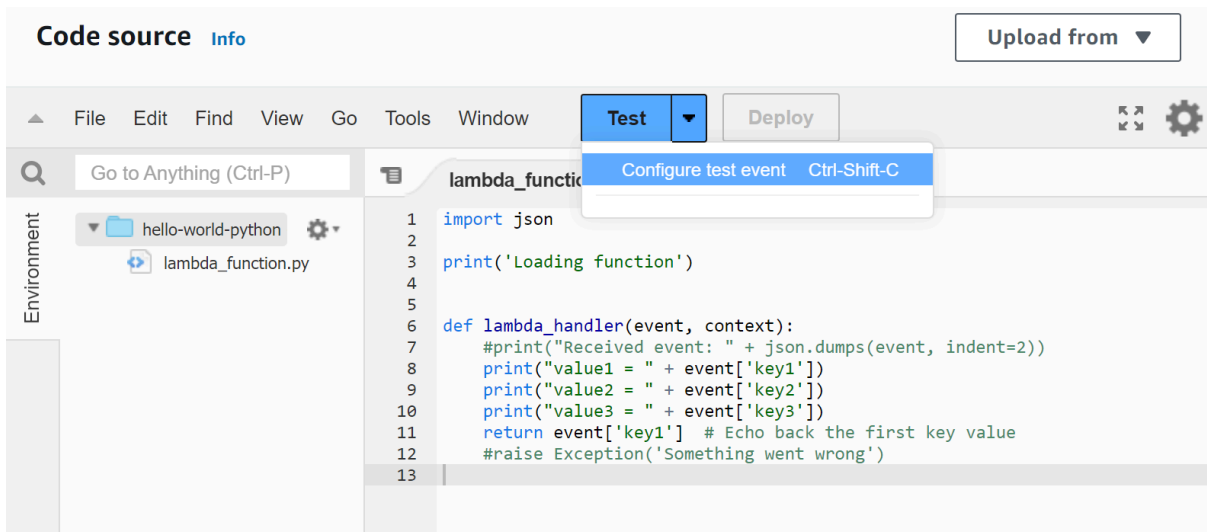
(Note: In AWS Lambda you specify a **handler** (a method/function in your code) where AWS Lambda can begin executing your code. AWS Lambda provides event data as input to this handler, which processes the event.)

Step 4: Invoke Lambda Function and Verify Results

Scroll down the page and double-click the "lambda_function.py" file and the console will show the hello-world-python Lambda function - you can now test the function, verify results, and review the logs:



a. Select **"Configure Test Event"** from the drop-down menu called "Test":



b. The editor pops up to enter an event to test your function.

- Choose Hello World from the Sample event template list from the Input test event page.
- Type in the event name HelloWorldEvent.
- You can change the values in the sample JSON, but don't change the event structure. For this tutorial, replace the value of key1 with "hello, world!".

Configure test event



A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

☒ Create new event

☐ Edit saved event

Event name

HelloWorldEvent

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

☒ Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

☐ Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

hello-world

Event JSON

Format JSON

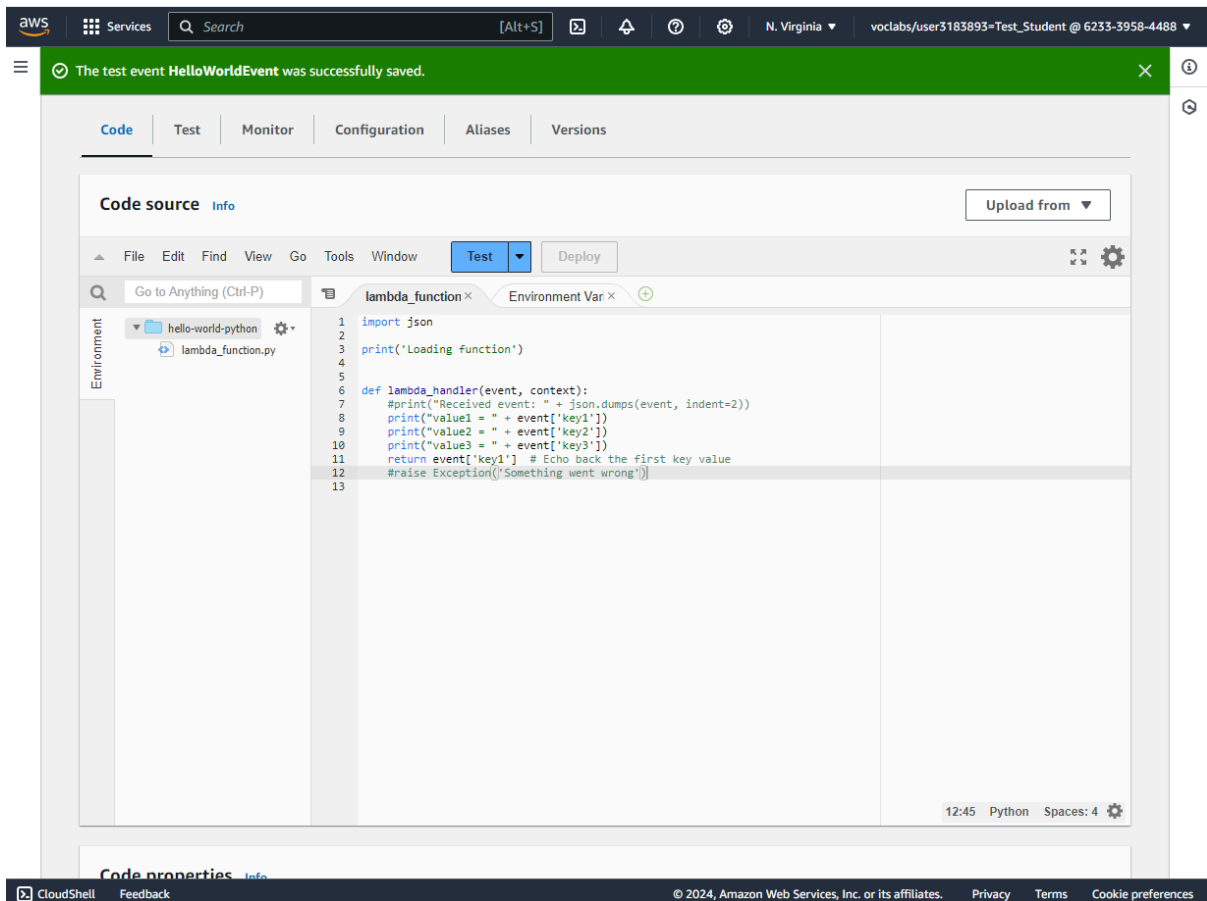
```
1 {  
2   "key1": "Hello, World!",  
3   "key2": "value2",  
4   "key3": "value3"  
5 }
```

Cancel

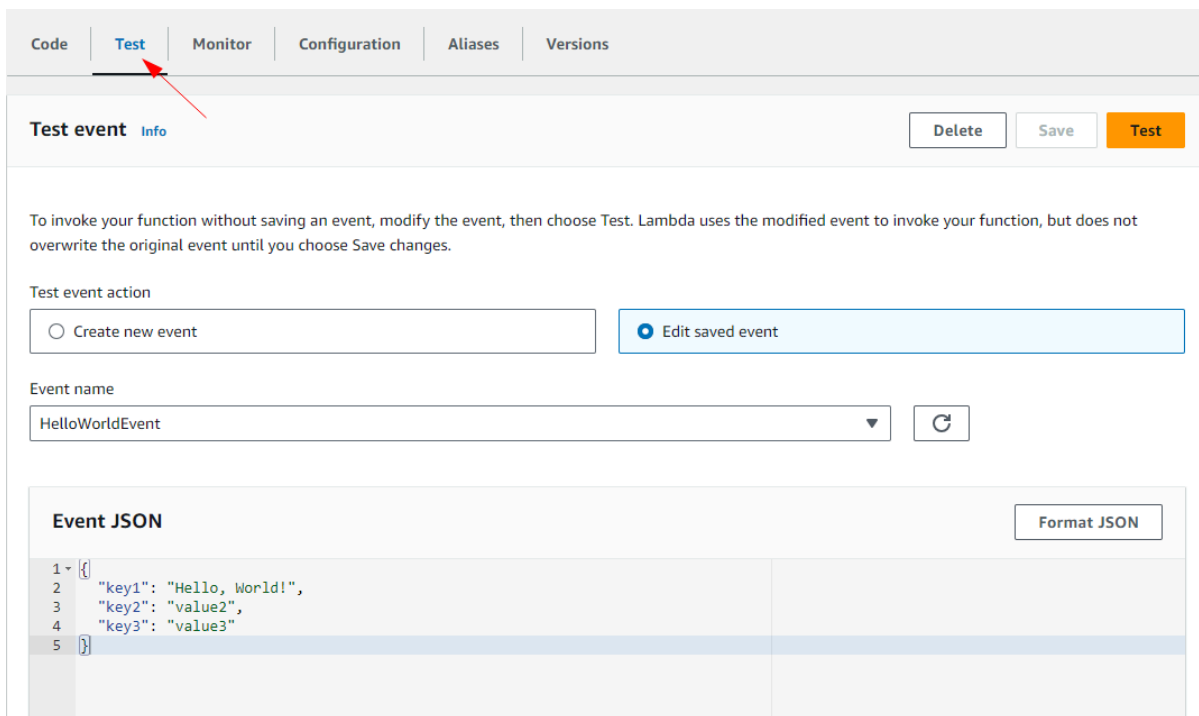
Invoke

Save

c. Select Save.



d. Click Test next to the Code option.




e. Upon successful execution, view the results in the console:

- The Execution results section verifies that the execution succeeded, now click detail in the green box (Execution function).

The screenshot shows the AWS Lambda console interface. At the top, there are tabs: Code, Test, Monitor, Configuration, Aliases, and Versions. The 'Test' tab is selected. Below the tabs, a green banner indicates 'Executing function: succeeded (logs [link])' with a 'Details' link. Below this, the 'Test event' section is visible. It includes buttons for 'Delete', 'Save', and 'Test' (which is circled in red). A text block explains that the 'Test' action uses a modified event without saving it. There are two radio buttons for 'Test event action': 'Create new event' and 'Edit saved event' (which is selected). Below this is a dropdown for 'Event name' with the value 'HelloWorldEvent' and a refresh button. At the bottom, there is a section for 'Event JSON' with a 'Format JSON' button. The JSON content is as follows:

```
1 {  
2   "key1": "Hello, World!",  
3   "key2": "value2",  
4   "key3": "value3"  
5 }
```

- The Summary section shows the key information reported in the Log output.
- The Log output section will show the logs generated by the Lambda function execution.



Executing function: succeeded ([logs](#))

▼ Details

The area below shows the last 4 KB of the execution log.

"Hello, World!"

Summary

Code SHA-256

PSUW0zelrnes9uLESxQPasVN3sqbP3U1T0Bf4tPcues=

Request ID

05b2b3f3-f88b-4e81-b494-1f6b1fd7b032

Init duration

78.99 ms

Billed duration

5 ms

Max memory used

31 MB

Execution time

1 minute ago (March 28, 2024 at 07:06 PM GMT+5:45)

Function version

\$LATEST

Duration

4.06 ms

Resources configured

128 MB

Log output

The section below shows the logging calls in your code. [Click here](#) to view the corresponding CloudWatch log group.

Loading function

START RequestId: 05b2b3f3-f88b-4e81-b494-1f6b1fd7b032 Version: \$LATEST

value1 = Hello, World!

value2 = value2

value3 = value3

END RequestId: 05b2b3f3-f88b-4e81-b494-1f6b1fd7b032

REPORT RequestId: 05b2b3f3-f88b-4e81-b494-1f6b1fd7b032 Duration: 4.06 ms Billed Duration: 5 ms Memory Size: 128 MB Max Memory Used: 31 MB Init Duration: 78.99 ms

Code source [Info](#)

Upload from ▼

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

lambda_function x Environment Var x Execution result: x

Environment

hello-world-python

lambda_function.py

▼ Execution results

Status: Succeeded Max memory used: 31 MB Time: 8.49 ms

Test Event Name

HelloWorldEvent

Response

"Hello, World!"

Function Logs

START RequestId: d05ae3b6-4eff-4b82-a84e-f9e5b2a8598e Version: \$LATEST

value1 = Hello, World!

value2 = value2

value3 = value3

END RequestId: d05ae3b6-4eff-4b82-a84e-f9e5b2a8598e

REPORT RequestId: d05ae3b6-4eff-4b82-a84e-f9e5b2a8598e Duration: 8.49 ms Billed Duration: 9 ms Memory Size: 128 MB Max Memory Used: 31 MB

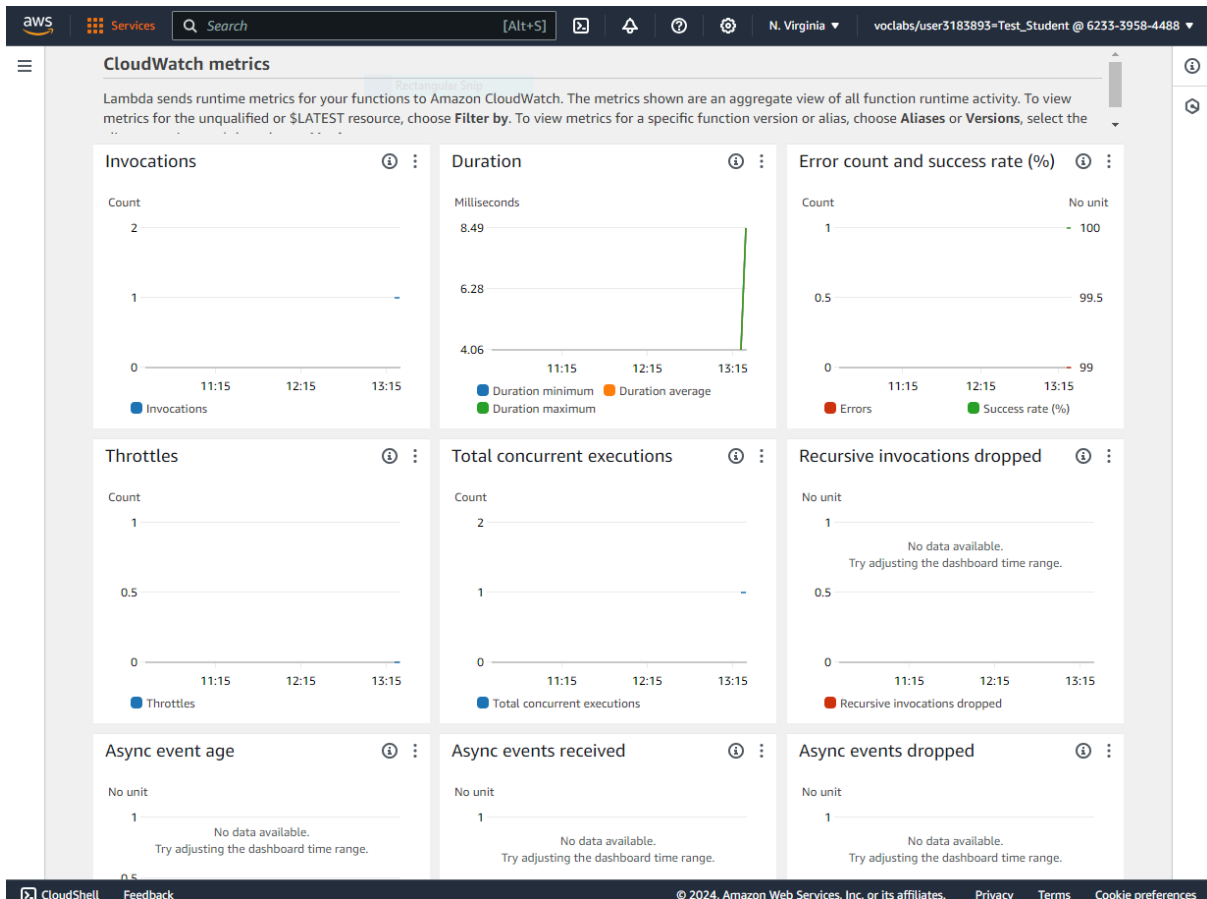
Request ID

d05ae3b6-4eff-4b82-a84e-f9e5b2a8598e

Step 5: Monitor Your Metrics

AWS Lambda automatically monitors Lambda functions and reports metrics through Amazon CloudWatch. To help you monitor your code as it executes, Lambda automatically tracks the number of requests, the latency per request, and the number of requests resulting in an error and publishes the associated metrics.

- Invoke the Lambda function a few more times by repeatedly clicking the Test button. This will generate the metrics that can be viewed in the next step.
- Select the Monitor tab to view the results.



- c. Scroll down to view the metrics for your Lambda function. Lambda metrics are reported through Amazon CloudWatch. You can leverage these metrics to set custom alarms.

The Monitoring tab will show several CloudWatch metrics.

Step 6: Updating the Test Event and the Lambda Function

- a. To change the test event, go to the Test tab and click on the "HelloWorldEvent" menu:

Test event
Info
Delete
Save
Test

To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save changes.

Test event action

☐ Create new event
☒ Edit saved event

Event name

HelloWorldEvent
Private saved events
HelloWorldEvent
Format JSON

```

1 {
2   "key1": "Hello, World!",
3   "key2": "value2",
4   "key3": "value3"
5 }
```

b. Change the event values to the following (obviously change my name for yours!):

Test event
Info
Delete
Save
Test

To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save changes.

Test event action

☐ Create new event
☒ Edit saved event

Event name

HelloWorldEvent
Format JSON

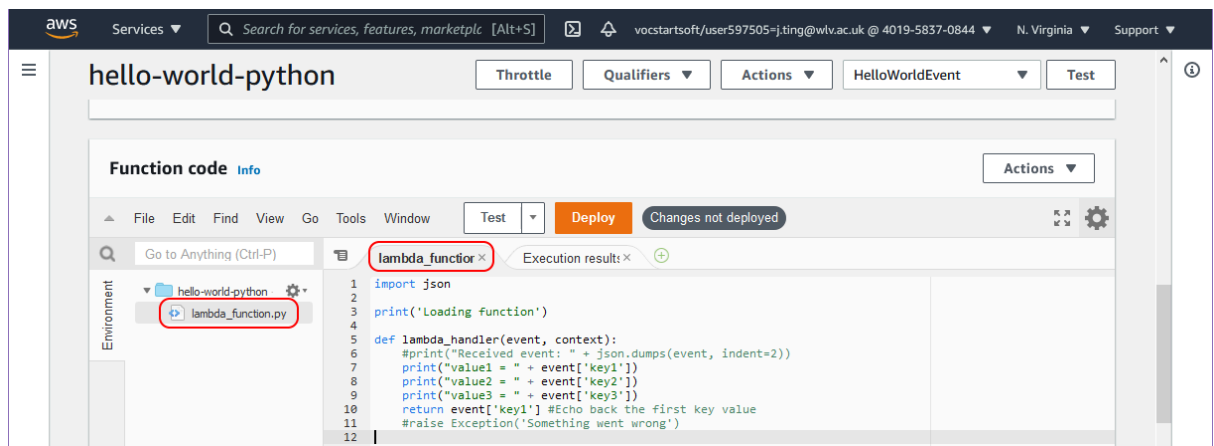
```

1 {
2   "key1": "Hello, ",
3   "key2": "Pratima Gautam!"
4 }
```

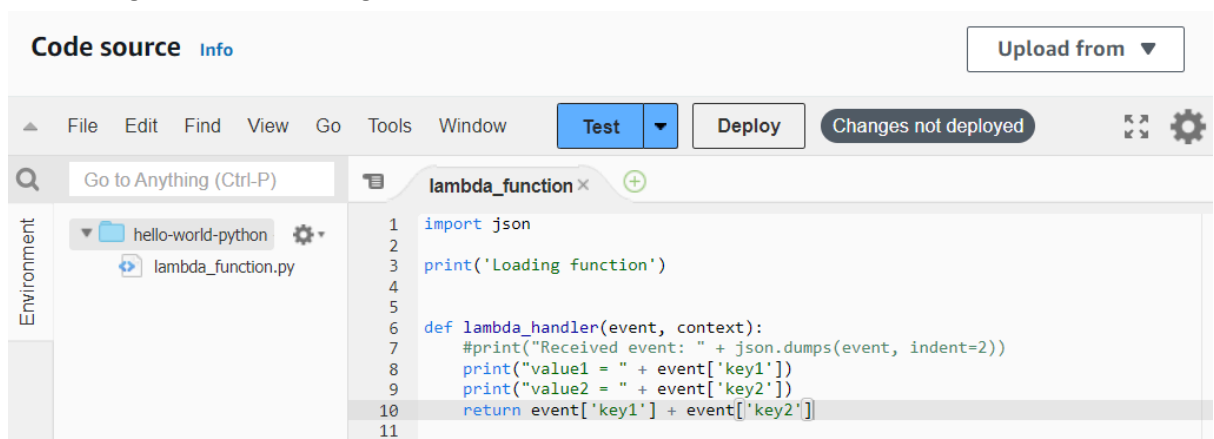
Note that we have also deleted "key3".

c. Click **Save**.

- d. Next scroll down to your Lambda function code :



- e. Then change it to the following:

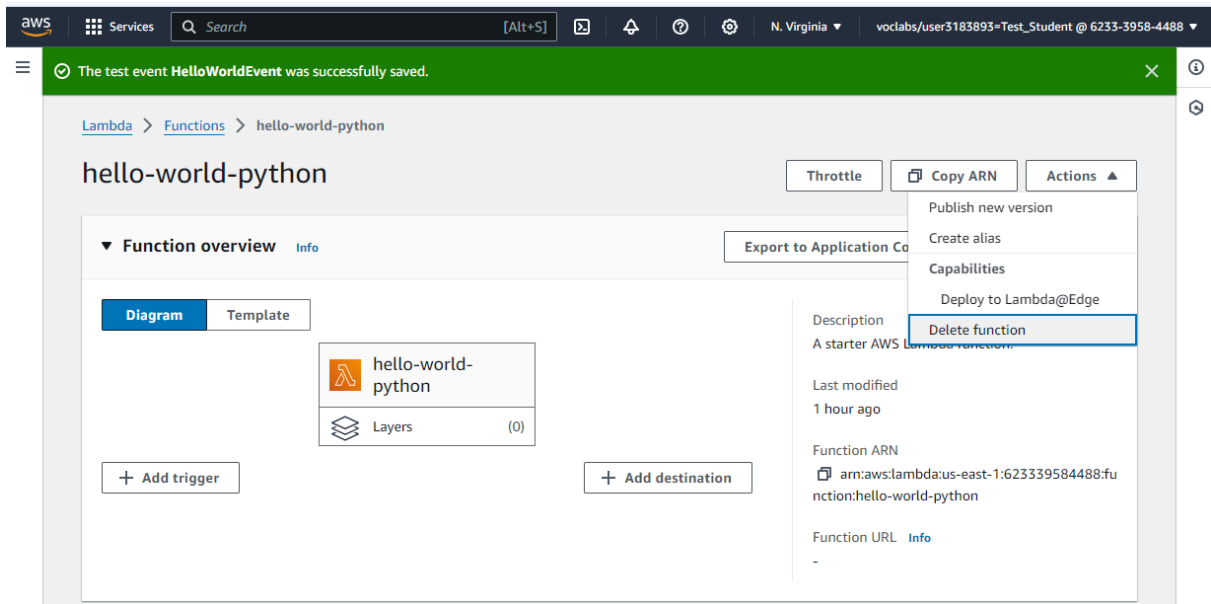


and click the **"Deploy"** button. If successful, it would say "Changes deployed"

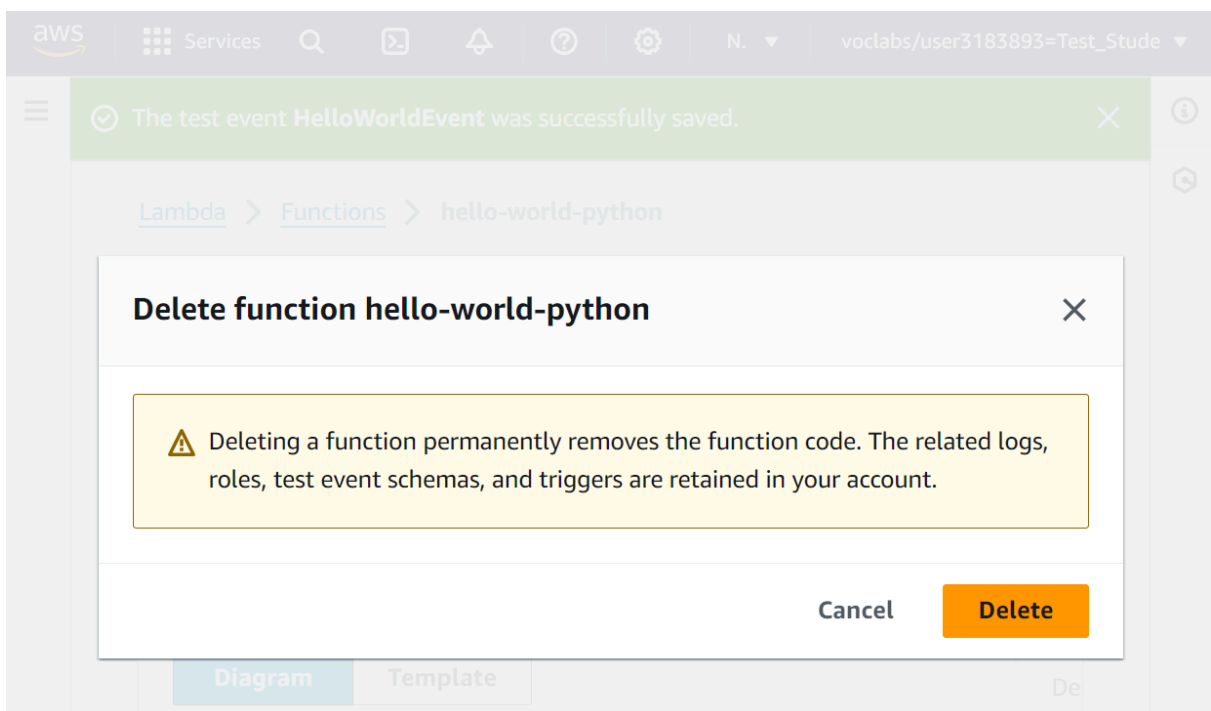
- f. Now go back up the page and click "Test". It should now change the test output to something like this with the "Hello, Pratima Gautam!" as the function result:
- g. Now try it a few more times with different test events and different outputs.

Step 7: Delete the Lambda Function

When you don't need your Lambda function any more, you can delete it from the AWS Lambda console.



- a. Select the Actions button and click Delete Function.



- b. You will be asked to confirm your termination - select Delete.

Please note that the delete step is optional. I am just showing you how it is done. For your coursework assessment later on, please don't delete the functions, or I won't be able to test them when I am marking.

Workshop Task

Now, using the steps you learnt in this workshop, create a new Lambda function to add 3 numbers together and return a message that says "The sum of 3 and 4 and 5 is 12". The numbers are the 3 test event values, and thus can be changed for each test run:

The screenshot displays the AWS Lambda console interface for a function named 'hello-world-python'. The top navigation bar includes the AWS logo, 'Services', a search bar, and user information. The main content area shows the function's details, including its ARN and a 'Test' button. A modal window titled 'Execution result: succeeded (logs)' is open, displaying the function's output and summary.

hello-world-python [Throttle] [Qualifiers] [Actions] [HelloWorldEvent] [Test]

ARN - `arn:aws:lambda:us-east-1:401958370844:function:hello-world-python`

Execution result: succeeded (logs)

▼ Details

The area below shows the result returned by your function execution. [Learn more](#) about returning results from your function.

"The sum of 3 and 4 and 5 is 12"

Summary

Code SHA-256	Request ID
<code>rHDxm1VUDRx/wZytxL2XL8QlIFIZkN+I78USpKkETPc=</code>	<code>9e1e9336-85dd-43c7-865b-e7e0e056043a</code>
Init duration	Duration
110.66 ms	1.56 ms
Billed duration	Resources configured
2 ms	128 MB
Max memory used	
48 MB	

Log output

The section below shows the logging calls in your code. [Click here](#) to view the corresponding CloudWatch log group.

```
START RequestId: 9e1e9336-85dd-43c7-865b-e7e0e056043a Version: $LATEST
value1 = 3
value2 = 4
value3 = 5
END RequestId: 9e1e9336-85dd-43c7-865b-e7e0e056043a
REPORT RequestId: 9e1e9336-85dd-43c7-865b-e7e0e056043a Duration: 1.56 ms Billed Duration: 2 ms Memory
Size: 128 MB Max Memory Used: 48 MB Init Duration: 110.66 ms
```

This is the end of this workshop. Now go back to the AWS Academy website and click on "End lab" to stop AWS from consuming any further credits:



AWS

Used \$0 of \$100

04:00

Start Lab

End Lab

AWS Details

Readme

Reset

oss_v1_x_p5f_1049797@unee0479831-\$ []

Region: us-east-1
Lab ID: arn:aws:cloudformation:us-east-1:205597652093:stack/c49160a727365116151361w205597652093/81e43f30-883b-11ec-bc96-0a385e26979
Creation Time: 2022-02-07T09:29:30-0800
Start session at: 2022-02-08T02:31:37-0800
Remaining session time: 04:00:00(240 minutes)

Previous

Next