



AWS Lambda "Hello, World!"

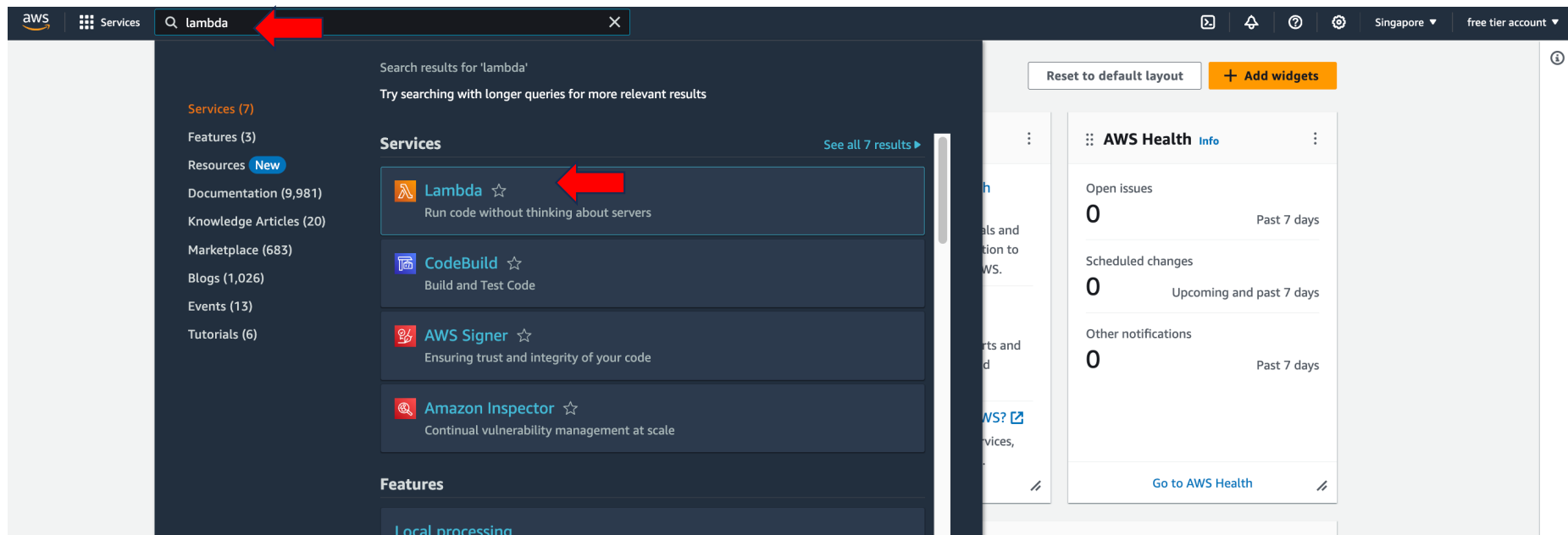


AWS Lambda Service

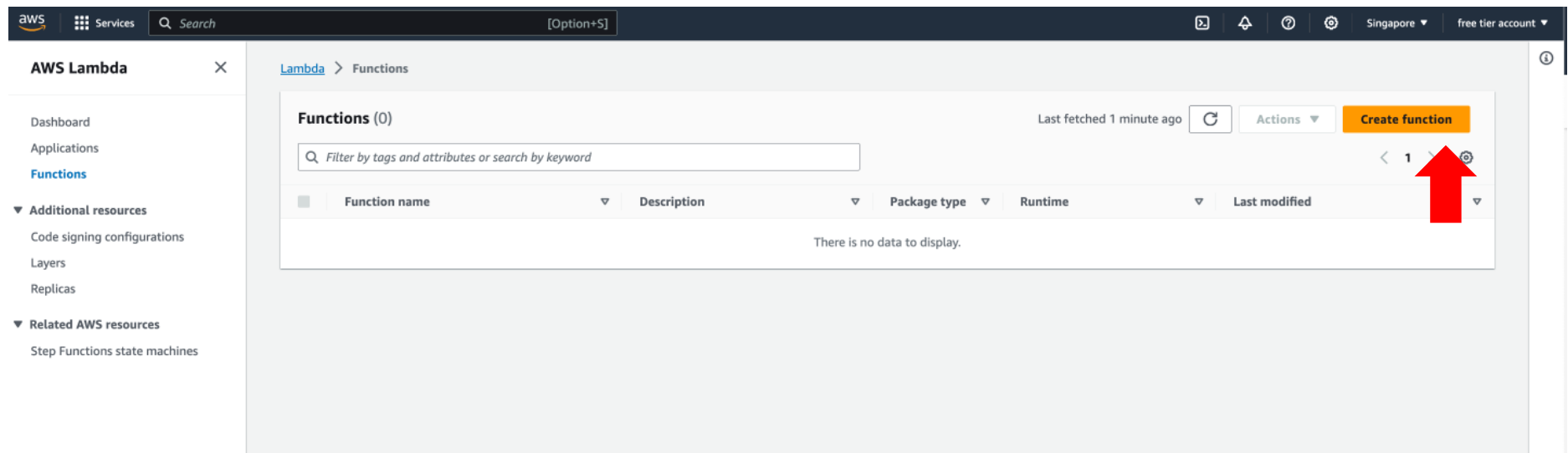
Go to

<https://aws.amazon.com/tutorials/run-serverless-code/>

When you [click here](#), the AWS Management Console will open in a new browser window, so you can keep this step-by-step guide open. In the top navigation bar, search for Lambda and open the AWS Lambda Console.



In the AWS Lambda console, choose **Create function**.




1. Select "use a blueprint"
2. In the Blueprint name box, ensure "Hello world function" with "python 3.10" blueprint is selected.
3. In the Function name box, enter "hello-world-python".
4. For Execution role, select "Create a new role with basic Lambda permissions".

The screenshot shows the AWS Lambda 'Create function' console. At the top, there are three tabs: 'Author from scratch', 'Use a blueprint' (which is selected and highlighted with a red arrow), and 'Container image'. Below the tabs is the 'Basic information' section. It contains several fields: 'Blueprint name' with a dropdown menu showing 'Hello world function' and 'python3.10' (indicated by red arrows), 'Function name' with a text input field containing 'hello-world-python' (indicated by a red arrow), 'Runtime' set to 'python3.10', 'Architecture' set to 'x86_64', and 'Execution role' with three radio button options: 'Create a new role with basic Lambda permissions' (selected, indicated by a red arrow), 'Use an existing role', and 'Create a new role from AWS policy templates'. A blue information box at the bottom states: 'Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.'

Lambda function code

Code is preconfigured by the chosen blueprint. You can configure it after you create the function. [Learn more](#) about deploying Lambda functions.

 This function contains external libraries.

```
1 import json
2
3 print('Loading function')
4
5
6 def lambda_handler(event, context):
7     #print("Received event: " + json.dumps(event, indent=2))
8     print("value1 = " + event['key1'])
9     print("value2 = " + event['key2'])
10    print("value3 = " + event['key3'])
11    return event['key1'] # Echo back the first key value
12    #raise Exception('Something went wrong')
13
```

Cancel

Create function



Press the Create Function button

The lambda function *hello-world-python* has been created successfully

The screenshot displays the AWS Lambda console interface. At the top, a green notification bar states: "Successfully created the function **hello-world-python**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

The main heading is "hello-world-python". To the right of the heading are buttons for "Throttle", "Copy ARN", and "Actions".

The "Function overview" section contains a card for the function "hello-world-python" with a "Layers" section showing "(0)". Below this card is a "+ Add trigger" button. To the right of the card is a "+ Add destination" button.

On the right side of the overview, the following details are listed:

- Description: A starter AWS Lambda function.
- Last modified: 44 seconds ago
- Function ARN: `arn:aws:lambda:ap-southeast-1:566069687272:function:hello-world-python`
- Function URL: [Info](#)

Below the overview, there are tabs for "Code", "Test", "Monitor", "Configuration", "Aliases", and "Versions". The "Code" tab is currently selected.

The "Code source" section shows an "Upload from" button. At the bottom, there is a toolbar with "File", "Edit", "Find", "View", "Go", "Tools", and "Window" menus, along with "Test" and "Deploy" buttons.

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

The screenshot shows the AWS Lambda console interface for a function named 'hello-world-python'. The top navigation bar includes 'Lambda', 'Functions', and the function name. Below this, the 'Function overview' section displays the function name, a description ('A starter AWS Lambda function.'), last modified time ('3 minutes ago'), function ARN, and function URL. The 'Code source' section is active, showing the code editor with a Python lambda handler. The 'Test' dropdown menu is open, and a red arrow points to the 'Configure test event' option.

hello-world-python

Throttle Copy ARN Actions

Function overview Info

hello-world-python

Layers (0)

+ Add trigger

+ Add destination

Description
A starter AWS Lambda function.

Last modified
3 minutes ago

Function ARN
arn:aws:lambda:ap-southeast-1:566069687272:function:hello-world-python

Function URL Info

Code Test Monitor Configuration Aliases Versions

Code source Info Upload from

File Edit Find View Go Tools Window Test Deploy

Go to Anything (% P)

Environment

hello-world-python

lambda_function.py

```
1 import json
2
3 print('Loading function')
4
5
6 def lambda_handler(event, context):
7     #print("Received event: " + json.dumps(event, indent=2))
8     print("value1 = " + event['key1'])
9     print("value2 = " + event['key2'])
10    print("value3 = " + event['key3'])
11    return event['key1'] # Echo back the first key value
12    #raise Exception('Something went wrong')
13
```

Configure test event

The editor pops up so you can enter an event to test your function.

- Select **Create new event**.
- Type in an event name like **HelloWorldEvent**.
- Retain default setting of **Private** for Event sharing settings.
- Choose **hello-world** from the template list.
- You can change the values in the sample JSON, but don't change the event structure. For this tutorial, replace *value1* with **hello, world!**.
- Click **Save** button

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

aws

Services

Search

[Option+S]

Singapore

free

hello-world-python

ThrottleCopy ARNActions

Function overviewInfo

hello-world-python

Layers(0)

+ Add trigger

+ Add destination

Description

A starter AWS Lambda function.

Last modified

10 minutes ago

Function ARN

arn:aws:lambda:ap-southeast-1:566069687272:function:hello-world-python

Function URL

Info

CodeTestMonitorConfigurationAliasesVersions

Code sourceInfo

Upload from

FileEditFindViewGoToolsWindowTestDeploy

Go to Anything (P)

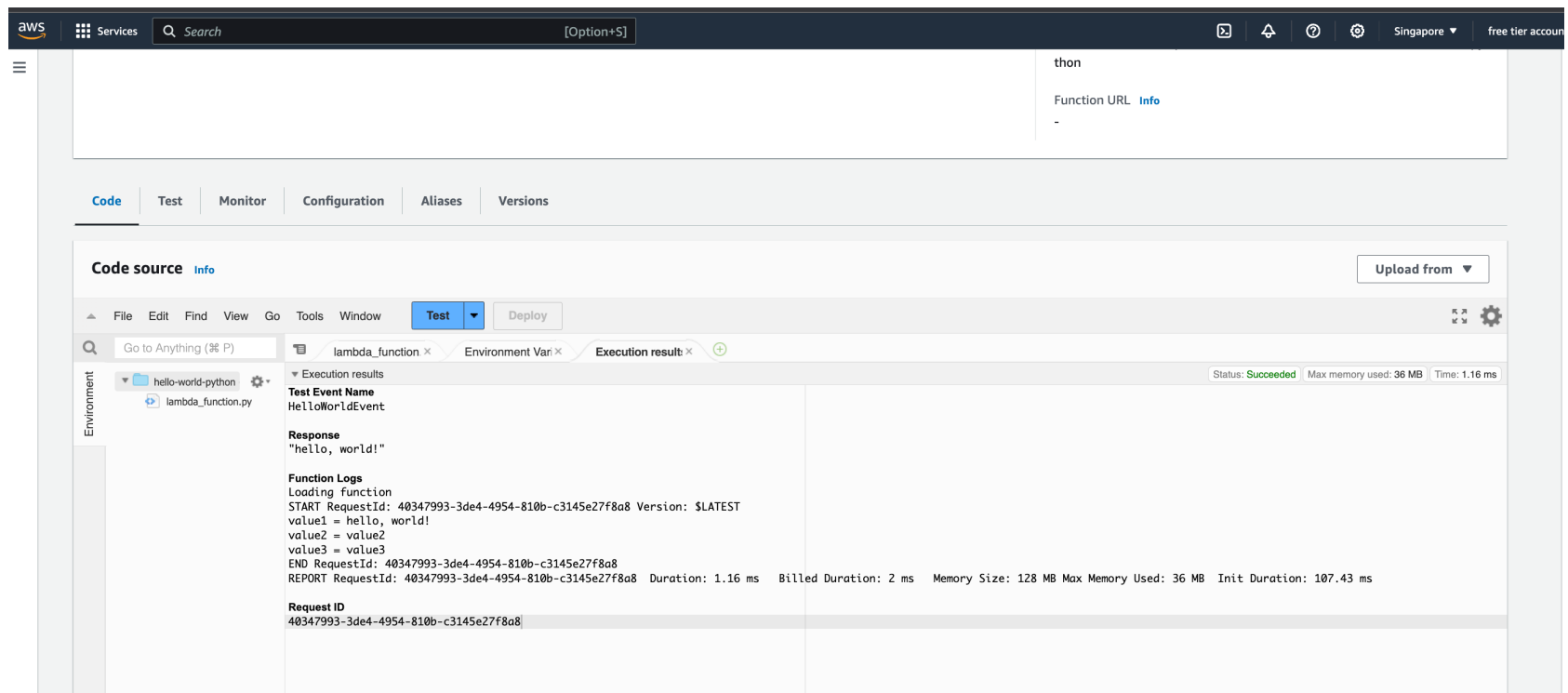
Environment

hello-world-python

lambda_function.py

```
1 import json
2
3 print('Loading function')
4
5
6 def lambda_handler(event, context):
7     #print("Received event: " + json.dumps(event, indent=2))
8     print("value1 = " + event['key1'])
```

Upon successful execution, view the results in the console:
The Execution results tab verifies that the execution succeeded



Extra mile: monitors Lambda metrics

- AWS Lambda automatically monitors Lambda functions and reports metrics through Amazon CloudWatch.
 - This page is optional and is intended for participants who wish to delve deeper into the topic, as we didn't cover CloudWatch in this workshop.
 - For more information about CloudWatch, see the [Amazon CloudWatch Developer Guide](#).

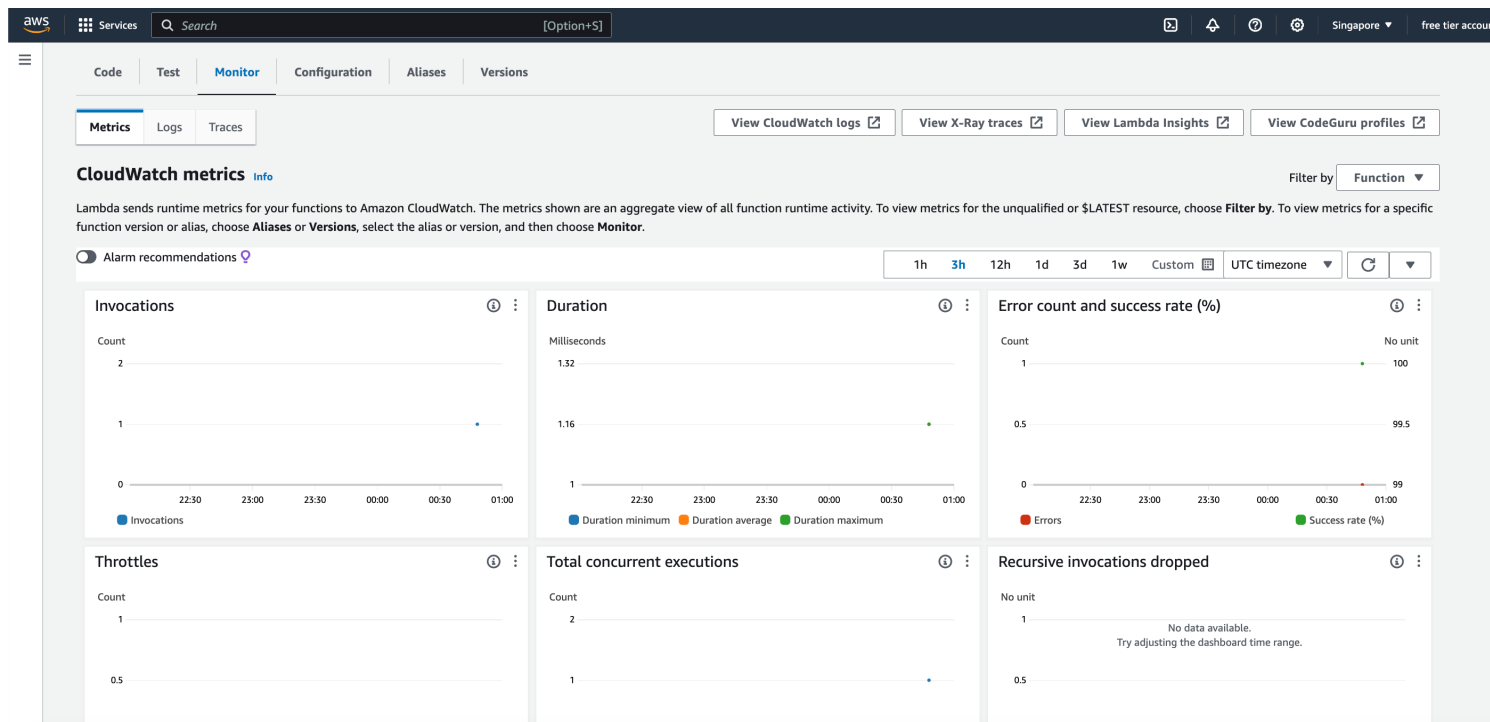
Invoke the Lambda function a few more times by repeatedly choosing the **Test** button. This will generate the metrics that can be viewed in the next step.

The screenshot shows 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 regional settings for 'Singapore'. The function name 'hello-world-python' is displayed at the top right, along with buttons for 'Throttle', 'Copy ARN', and 'Actions'. Below this, the 'Function overview' section shows the function icon, name, and a 'Layers' section with '(0)' layers. There are buttons for '+ Add trigger' and '+ Add destination'. To the right, a 'Description' box contains the text 'A starter AWS Lambda function.', 'Last modified 10 minutes ago', 'Function ARN arn:aws:lambda:ap-southeast-1:566069687272:function:hello-world-python', and 'Function URL Info'. Below the overview, a tabbed interface shows 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The 'Code source' section is active, displaying a code editor with a file explorer on the left showing 'hello-world-python' and 'lambda_function.py'. The code editor has a menu bar (File, Edit, Find, View, Go, Tools, Window) and a toolbar with 'Test' and 'Deploy' buttons. A red arrow points to the 'Test' button. The code in the editor is as follows:

```
1 import json
2
3 print('Loading function')
4
5
6 def lambda_handler(event, context):
7     #print("Received event: " + json.dumps(event, indent=2))
8     print("value1 = " + event['key1'])
```

Select the **Monitor** tab to view the results.

- The Monitoring tab will show seven CloudWatch metrics: *Invocations*, *Duration*, *Error count and success rate (%)*, *Throttles*, *Async delivery failures*, *IteratorAge*, and *Concurrent executions*.



Clean Up - Select the **Actions** button and select **Delete function**

The screenshot shows 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 the region 'Singapore'. The breadcrumb trail is 'Lambda > Functions > hello-world-python'. The function name 'hello-world-python' is displayed at the top, along with buttons for 'Throttle', 'Copy ARN', and 'Actions'. The 'Actions' dropdown menu is open, showing options: 'Publish new version', 'Create alias', 'Export function', and 'Delete function'. A red arrow points to the 'Delete function' option. Below the function name, there are tabs for 'Code', 'Test', 'Monitor' (selected), 'Configuration', 'Aliases', and 'Versions'. Under the 'Monitor' tab, there are sub-tabs for 'Metrics', 'Logs', and 'Traces'. The 'Metrics' sub-tab is active, showing 'CloudWatch metrics'. The metrics section includes a filter dropdown set to 'Function' and a time range selector set to '3h'. Three charts are displayed: 'Invocations' (Count vs Time), 'Duration' (Milliseconds vs Time), and 'Error count and success rate (%)' (Count vs Time). The 'Invocations' chart shows a count of 2. The 'Duration' chart shows a minimum of 1.32, average of 1.16, and maximum of 1.16 milliseconds. The 'Error count and success rate (%)' chart shows 1 error and a success rate of 100%.

aws Services Search [Option+S] Singapore free tier account

hello-world-python

Throttle Copy ARN Actions

- Publish new version
- Create alias
- Export function
- Delete function

Function overview Info

Code Test Monitor Configuration Aliases Versions

Metrics Logs Traces

View CloudWatch logs View X-Ray traces View Lambda Insights View CodeGuru profiles

CloudWatch metrics Info

Filter by Function

Lambda sends runtime metrics for your functions to Amazon CloudWatch. The metrics shown are an aggregate view of all function runtime activity. To view metrics for the unqualified or \$LATEST resource, choose **Filter by**. To view metrics for a specific function version or alias, choose **Aliases** or **Versions**, select the alias or version, and then choose **Monitor**.

Alarm recommendations

1h 3h 12h 1d 3d 1w Custom UTC timezone

Invocations

Count

2

1

0

22:30 23:00 23:30 00:00 00:30 01:00

Invocations

Duration

Milliseconds

1.32

1.16

1

22:30 23:00 23:30 00:00 00:30 01:00

Duration minimum Duration average Duration maximum

Error count and success rate (%)

Count

1

0.5

0

22:30 23:00 23:30 00:00 00:30 01:00

Errors Success rate (%)

