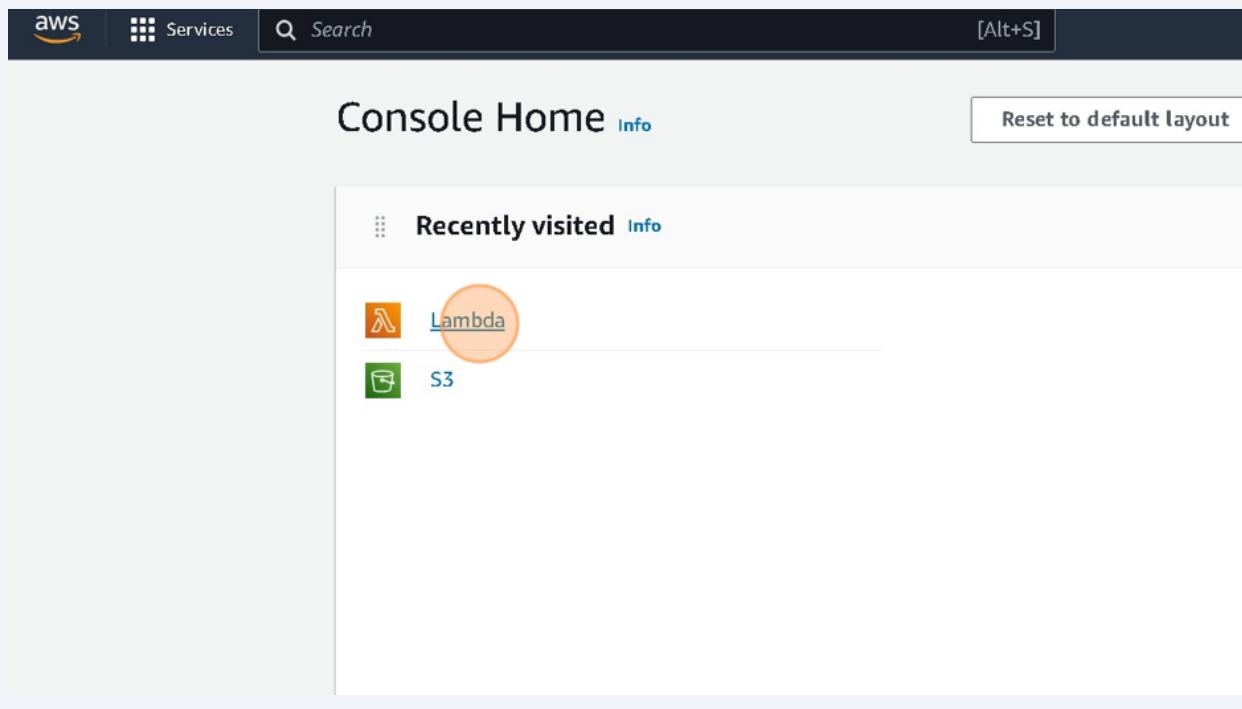


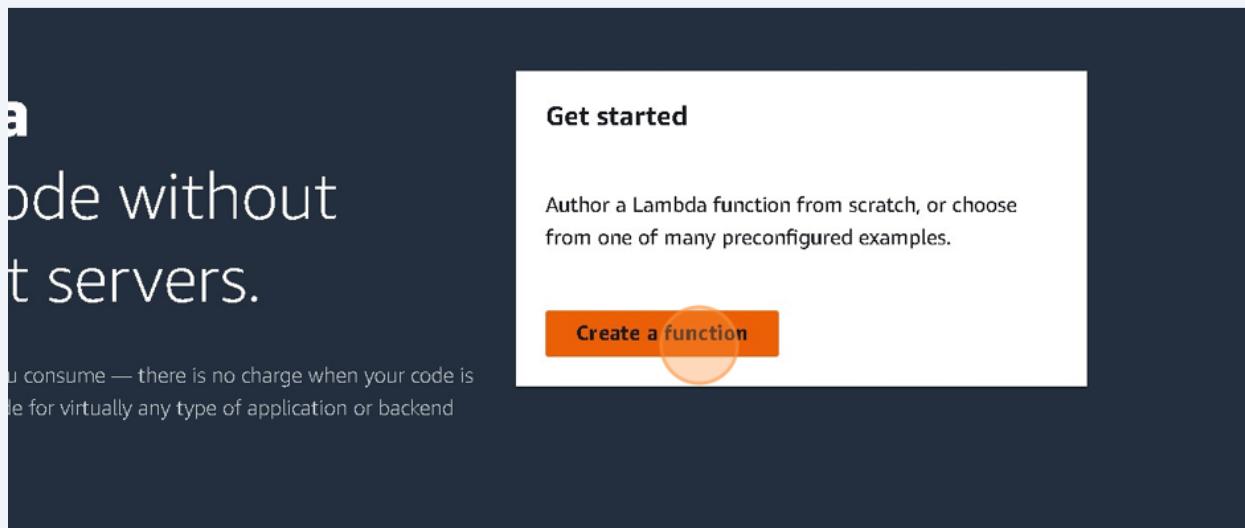
Guide to Build and Test Lambda Function with S3 Trigger

Scribe 

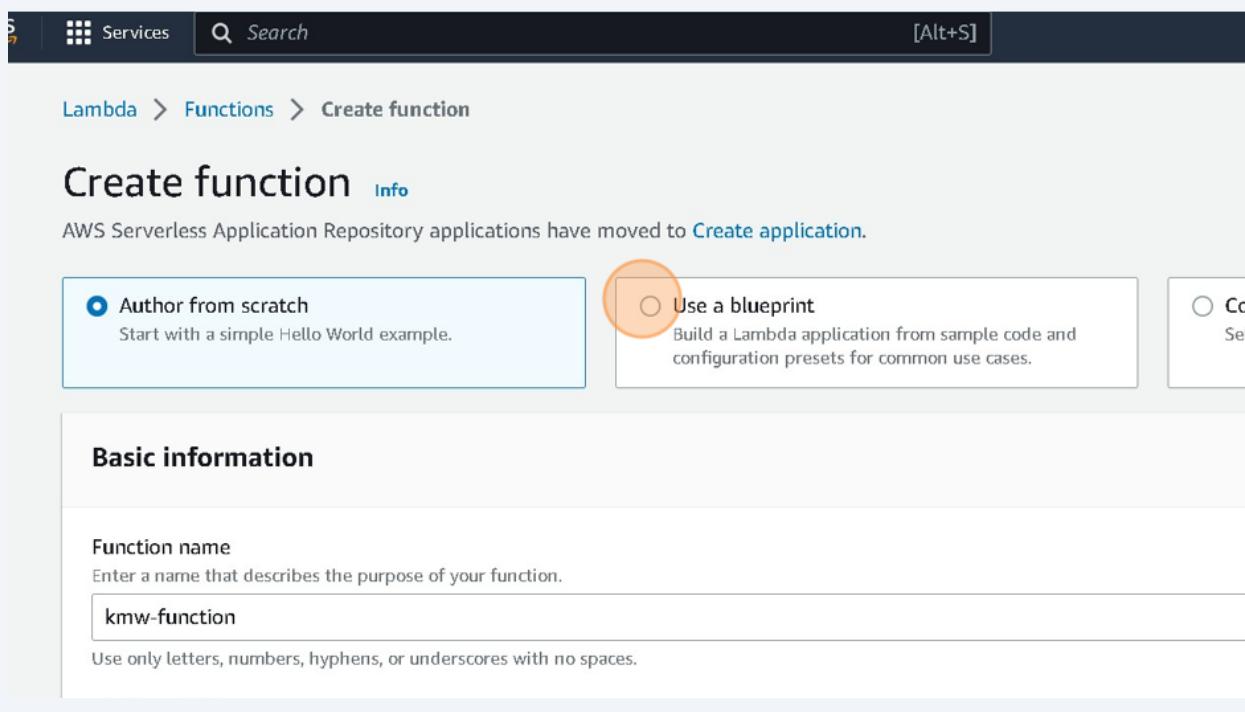
- 1 Click "Lambda"



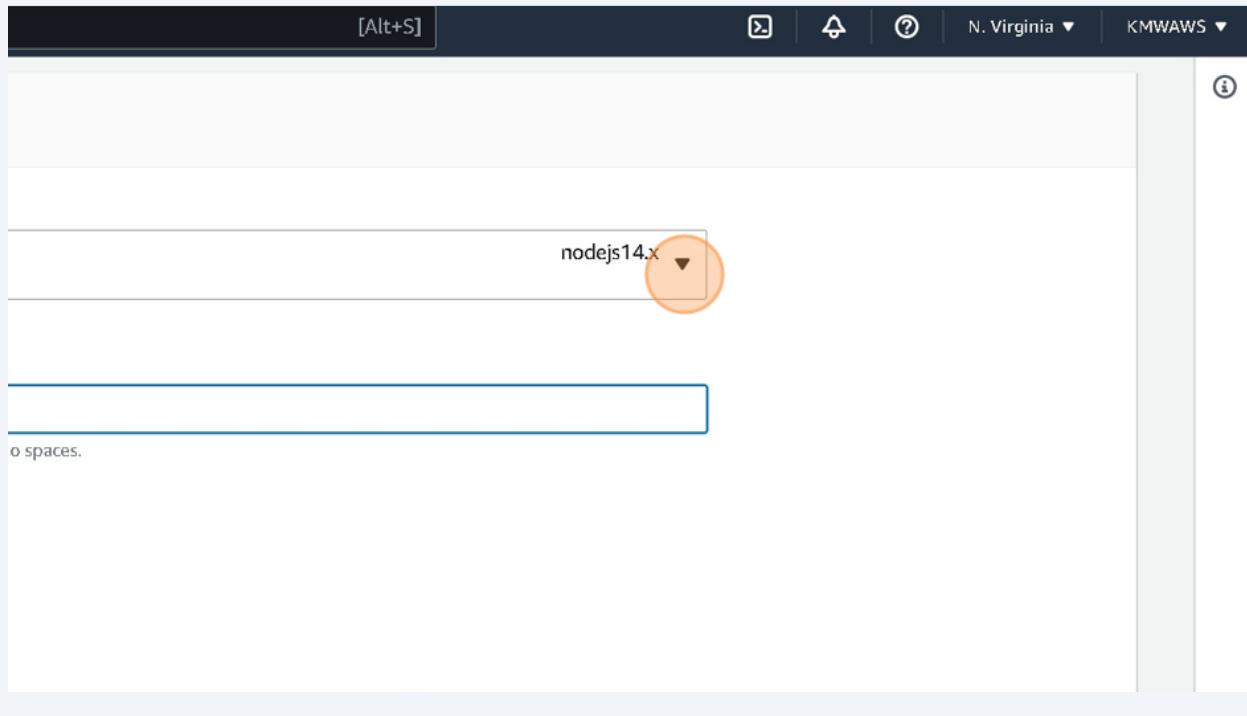
- 2 Click "Create a function"



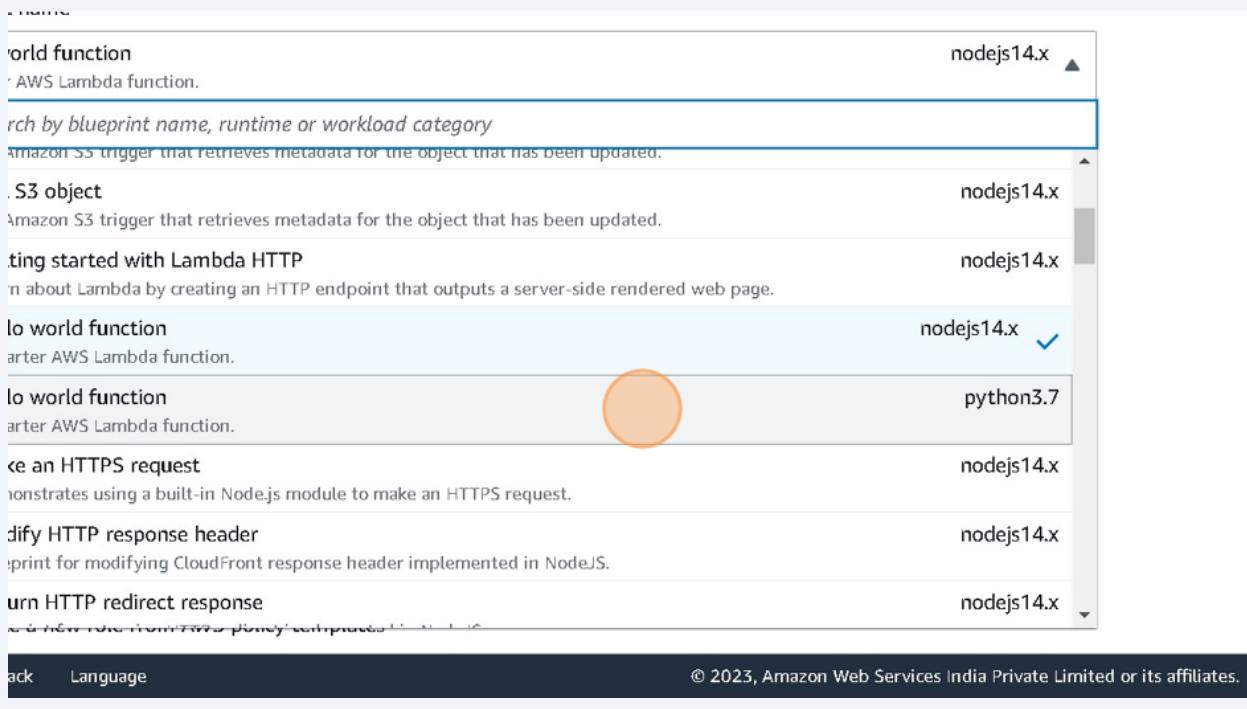
- 3 Click: "Use a blueprint".



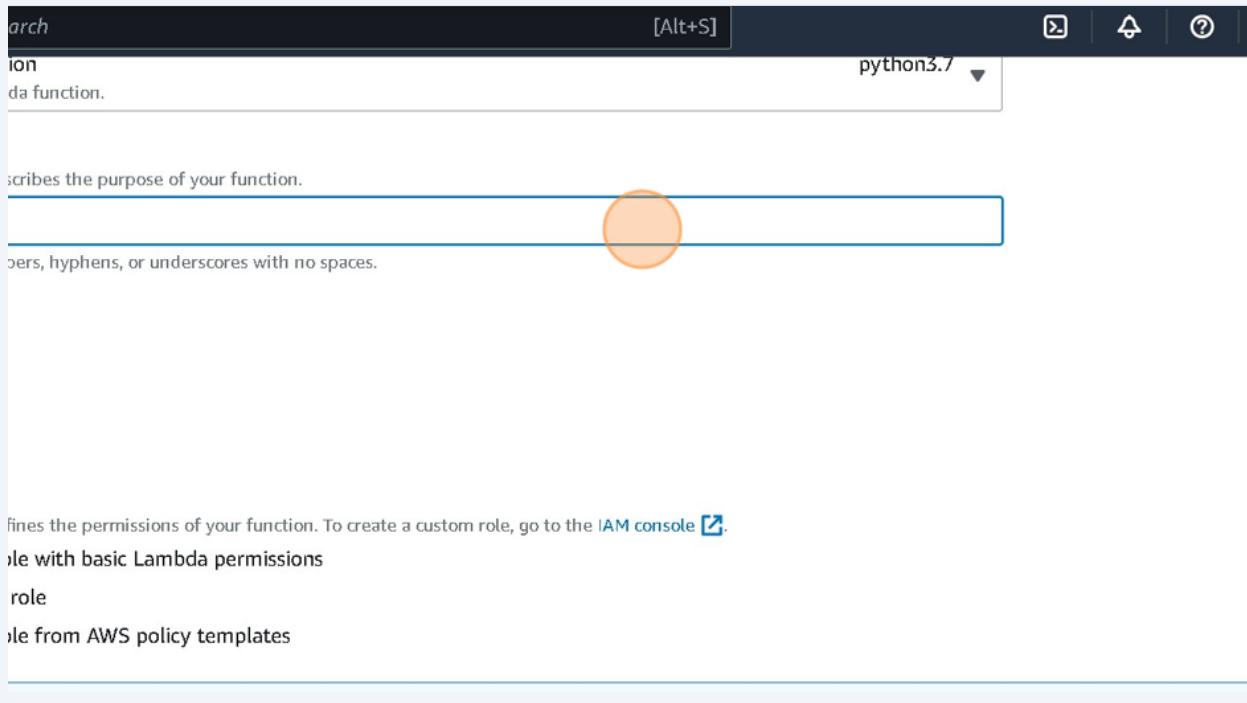
4 Click: "nodejs14.x"



5 Click "python3.7"

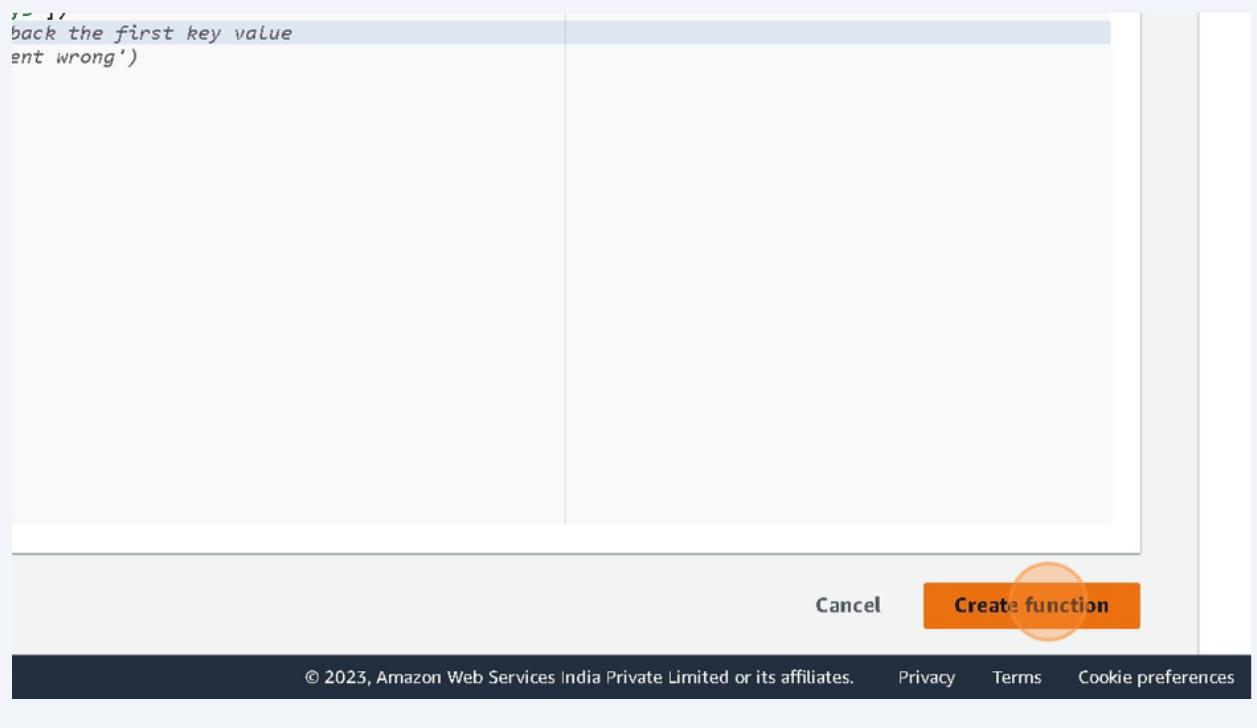


- 6 Click the "Function name" field.

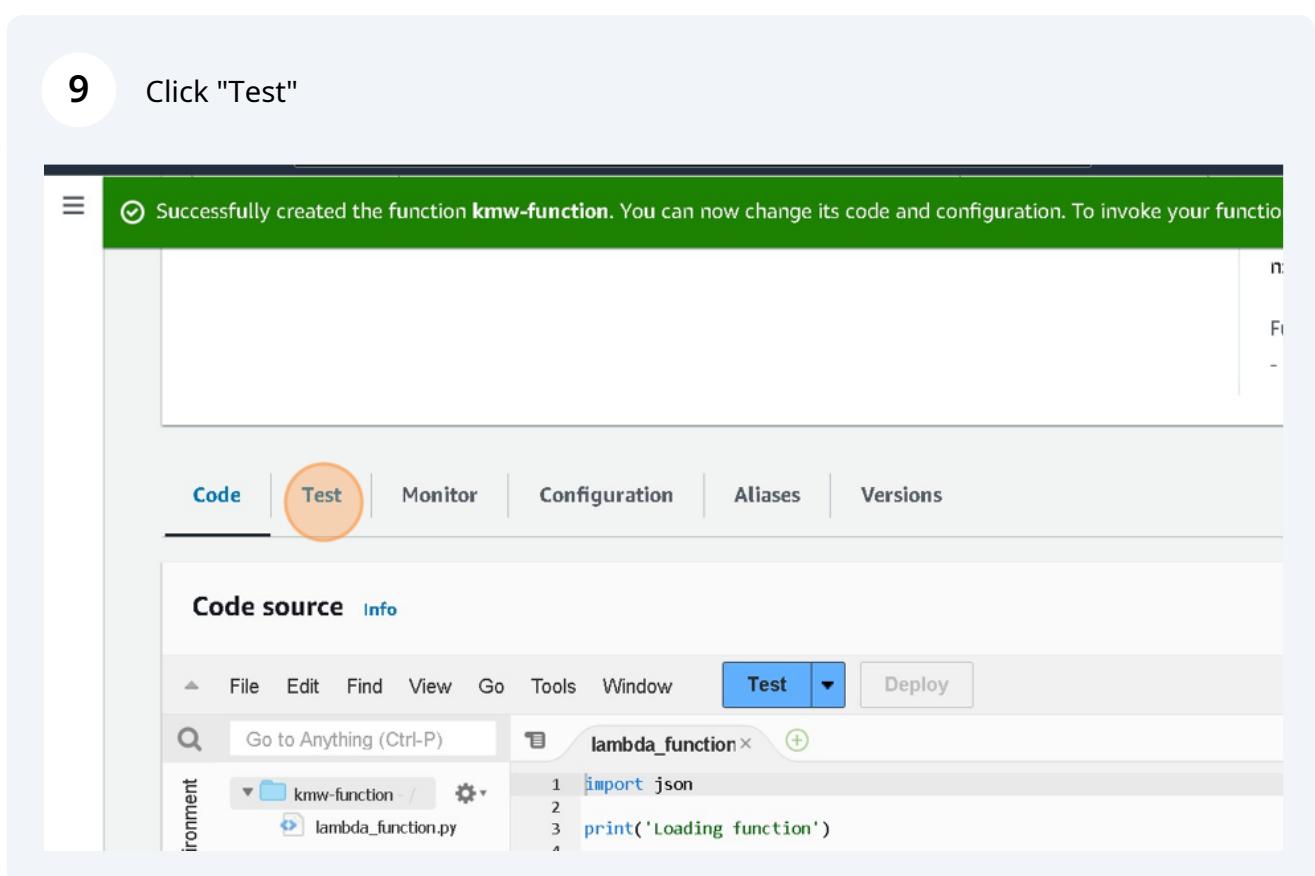


- 7 Type "kmw-function"

8 Click "Create function"



9 Click "Test"



10 Replace: "value1 with Hello World".

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

```
1  {
2   "key1": "value1",
3   "key2": "value2",
4   "key3": "value3"
5 }
```

11 Click the "Event name" field.

Successfully deleted.

Configure an event, configure the JSON event, then choose Test.

Event name ● Edit saved event

Event names can contain lowercase letters, numbers, dots, hyphens and underscores.

Event names must be unique across all AWS Lambda functions and aliases in your account. You can't use the same event name for multiple functions or aliases. If you try to use an event name that's already in use, you'll receive an error message. To avoid conflicts, we recommend using a descriptive name for your event, such as "HelloWorldEvent" or "MyFunctionEvent".

Event names must be unique across all AWS Lambda functions and aliases in your account. You can't use the same event name for multiple functions or aliases. If you try to use an event name that's already in use, you'll receive an error message. To avoid conflicts, we recommend using a descriptive name for your event, such as "HelloWorldEvent" or "MyFunctionEvent".

12 Type "kmw-event"

13 Click "Save"

The screenshot shows the AWS Lambda function configuration interface. At the top, there is a green bar with the text "5, in lambda_handler". Below it, a log entry is displayed: "-f982261de7d6 Duration: 17.00 ms Billed Duration: 18 ms Memory Size: 128 MB Max Memory Used: 156f-f982261de7d6". The main area contains configuration sections for "Handler", "Runtime", "Memory Size", and "Timeout". At the bottom right, there are two buttons: "Save" (highlighted with a red circle) and "Test". A note below the buttons says "Configure the JSON event, then choose Test." There is also a link "Edit saved event".

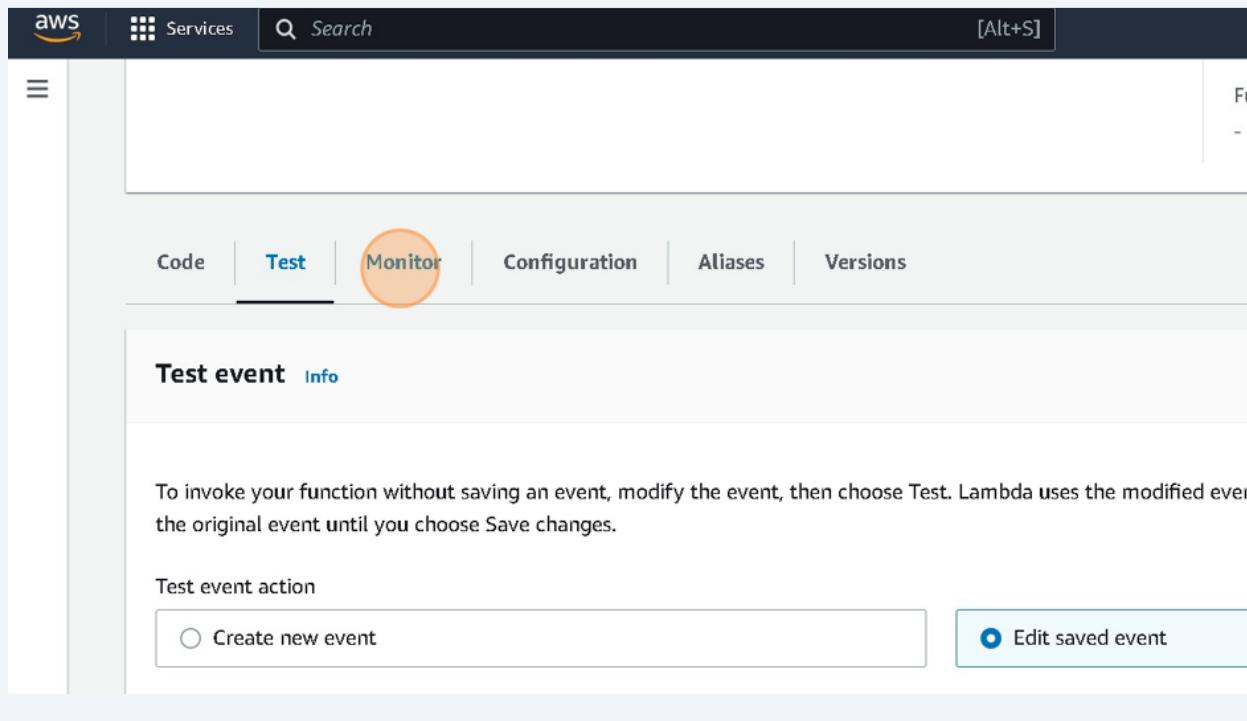
14 Click "Test"

The screenshot shows the AWS Lambda function configuration interface. At the top, there's a green header bar with the text "9, in lambda_handler". Below it, a log entry is displayed: "-f982261de7d6 156f-f982261de7d6 Duration: 17.00 ms Billed Duration: 18 ms Memory Size: 128 MB Max Memory Used:". On the right side of the interface, there are three buttons: "Delete", "Save", and "Test", with "Test" being highlighted with a red circle. Below these buttons, a note says "Modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event." At the bottom left, there's a "Edit saved event" button.

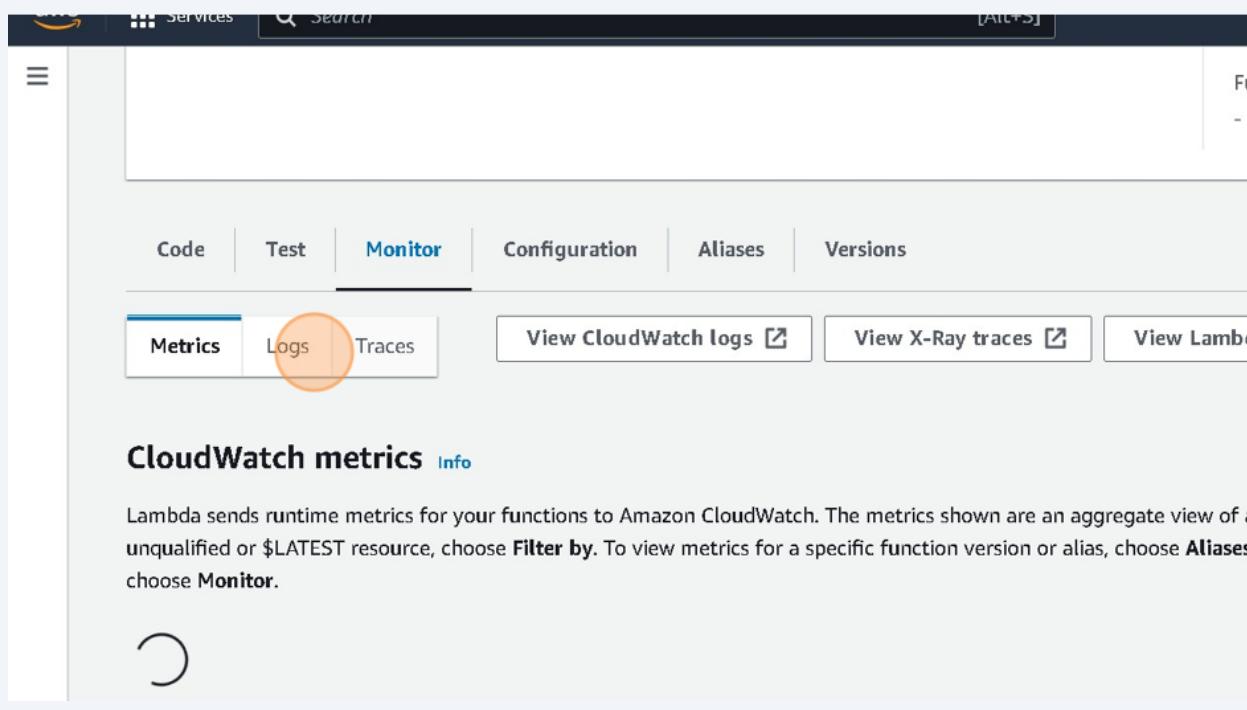
15 Successful

The screenshot shows the AWS Lambda function test results page. At the top, there's a dark header bar with the AWS logo, "Services", a search bar, and a "[Alt+S]" key shortcut. Below it, a green success message says "The test event kmw-event was successfully saved." The navigation tabs at the top are "Code", "Test" (which is highlighted with a red circle), "Monitor", "Configuration", "Aliases", and "Versions". The main content area shows a green box with a checkmark icon and the text "Executing function: succeeded (logs)". It also includes a "Details" section and a log entry: "The area below shows the last 4 KB of the execution log." followed by the text "Hello World". A "Summary" section provides execution details: "Code SHA-256: 1iEOulzd8iYhA31AxKFXGKzh072nzUYD28Sw4WSncA8=" and "Request ID: 14e0f9cb-1b2a-4d13-a083-". Below this, "Duration: 3.32 ms" and "Billed duration: 4 ms" are listed.

16 Click "Monitor"



17 Click "Logs"



18 Click "View CloudWatch logs"

The screenshot shows the AWS Lambda function configuration interface. At the top, there are tabs for Code, Test, Monitor (which is underlined in blue), Configuration, Aliases, and Versions. Below these, there are three buttons: Metrics, Logs (which is highlighted with an orange circle), and Traces. To the right of these buttons are four links: View CloudWatch logs, View X-Ray traces, View Lambda Insights, and another View CloudWatch logs link. The 'View CloudWatch logs' link is also circled in orange.

CloudWatch Logs Info

Lambda logs all requests handled by your function and automatically stores logs generated by your code through Amazon CloudWatch Metrics. You can instrument it with custom logging statements. The following tables list the most recent and most expensive function invocations. If you're using a specific function version or alias, visit the **Monitor** section at that level.

1h 3h 12h 1d

19 Click "2023/06/14/[\$LATEST]f46302197f21489fa51fc47f3d5bbfec"

The screenshot shows the AWS CloudWatch Log Streams page. At the top, there are tabs for Log streams (which is underlined in orange), Metric filters, Subscription filters, Contributor Insights, Tags, and Data protection. Below these tabs, there is a search bar with the placeholder "Filter log streams or try prefix search" and two checkboxes: "Exact match" and "Show expired". The main area displays a table of log streams. The first row contains a checkbox, the log stream name "Log stream", and a dropdown arrow. The second row contains a checkbox, the log stream name "2023/06/14/[\$LATEST]f46302197f21489fa51fc47f3d5bbfec", and the timestamp "2023-06-14 11:30:24 (UTC+05:30)". The third row contains a checkbox, the log stream name "2023/06/14/[\$LATEST]a5a0baa4da044312aa672f86d0312344", and the timestamp "2023-06-14 11:24:33 (UTC+05:30)". The bottom of the page has a Language dropdown and a copyright notice: "© 2023, Amazon Web Services India Private Limited or its affiliates."

20 You can now see the Logs related to your "Lambda Functions".

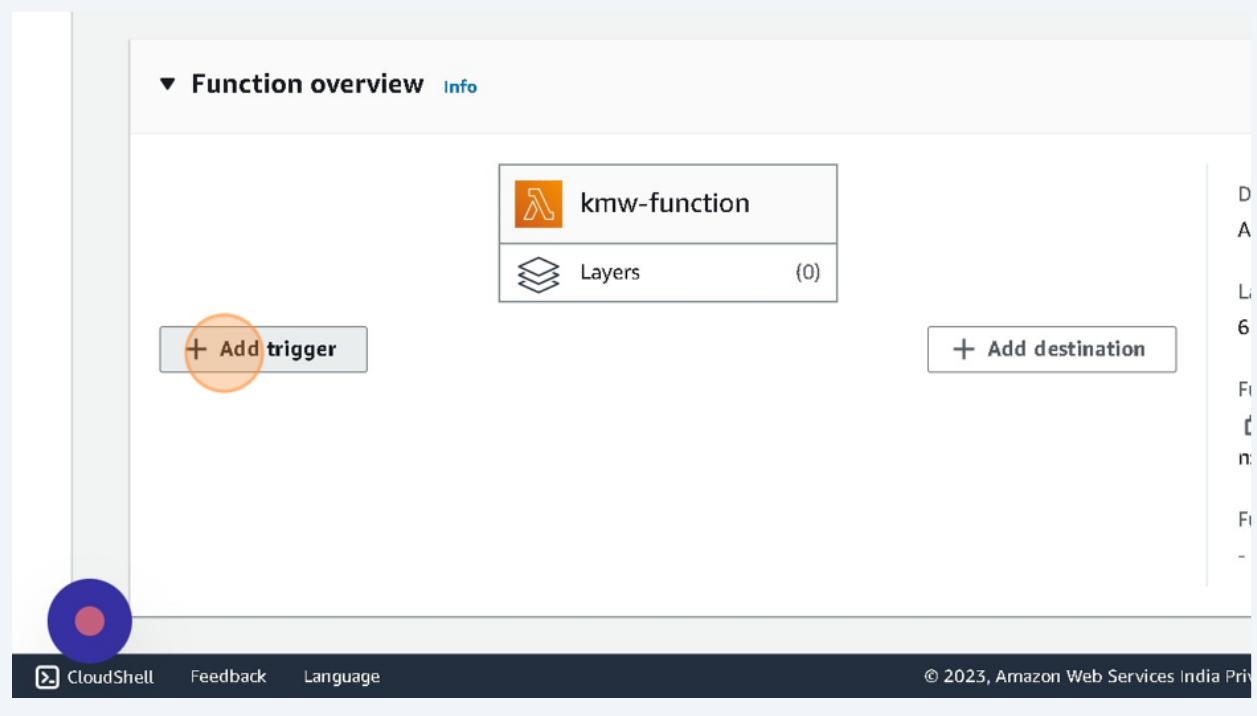
The screenshot shows the AWS Lambda logs interface. At the top, there's a header with tabs for 'Logs' and 'Metrics'. Below the header, the log entries are displayed in a table format:

Time	Message
14T11:30:24.355+05:30	INIT_START Runtime Version: python:3.7.v27 Runtime Version ARN: arn:aws:lambda:us-eas...
14T11:30:24.468+05:30	Loading function
14T11:30:24.470+05:30	START RequestId: 785a20c4-cc71-4b0a-abfc-bbb6cd66d74e Version: \$LATEST
14T11:30:24.471+05:30	value1 = Hello World
14T11:30:24.471+05:30	value2 = value2
14T11:30:24.471+05:30	value3 = value3
14T11:30:24.473+05:30	END RequestId: 785a20c4-cc71-4b0a-abfc-bbb6cd66d74e
14T11:30:24.473+05:30	REPORT RequestId: 785a20c4-cc71-4b0a-abfc-bbb6cd66d74e Duration: 1.75 ms Billed Durat...

At the bottom of the log table, a message says "No newer events at this moment. Auto retry paused. [Resume](#)". To the right of this message is a yellow "Back to top" button with an upward arrow icon.

At the very bottom of the page, there's a footer bar with links: "© 2023, Amazon Web Services India Private Limited or its affiliates.", "Privacy", "Terms", and "Cookie preference".

21 1. Now Again: Go to the Top
2. Click "Add trigger"



22 Click "Select a source"



Lambda > Add trigger

Add trigger

Trigger configuration Info

Select a source



Cancel



23 Click "S3"

Trigger configuration Info

Select a source



aws messaging multi-protocol



aws cluster



aws storage



aws messaging notifications pub-sub push



aws queue



aws networking vpc vpc lattice



CloudShell

Feedback

Language

© 2023, Amazon Web Services India

24 Click the "Bucket" field.

Trigger configuration [Info](#)

 S3
aws storage

Bucket
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.

Bucket must be in region us-east-1

Event types
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.



All object create events 

Prefix - optional
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

25 Click "kmwbucket"

Trigger configuration [Info](#)

 S3
aws storage

Bucket
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.

kmwbucket  

Event types
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.



All object create events 

Prefix - optional
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

26 Click this checkbox.

Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

e.g. images/

Suffix - optional

Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.

e.g. jpg

Recursive invocation

If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#)

I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.



CloudShell

Feedback

Language

Cancel



© 2023, Amazon Web Services India Private Limited

27 Click "Add"

Objects with keys that start with matching characters.

Objects with keys that end with matching characters.

If you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#)

for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

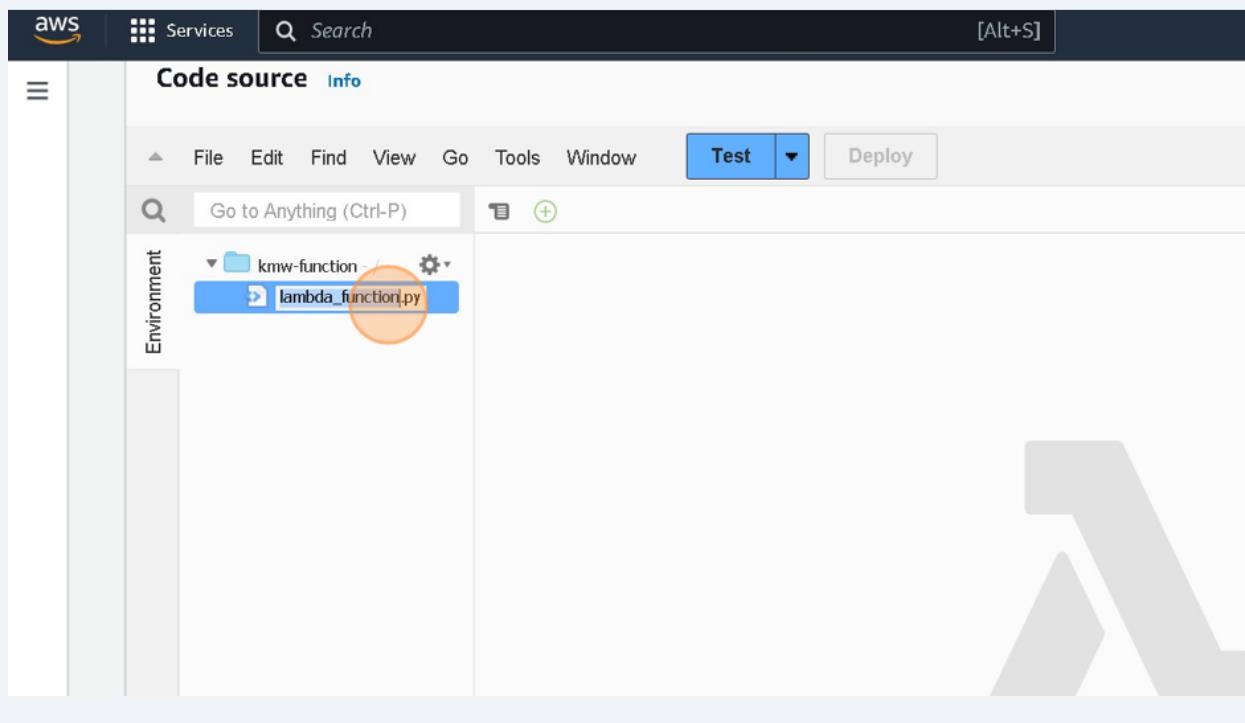
AWS S3 to invoke your Lambda function from this trigger. [Learn more](#)

Cancel

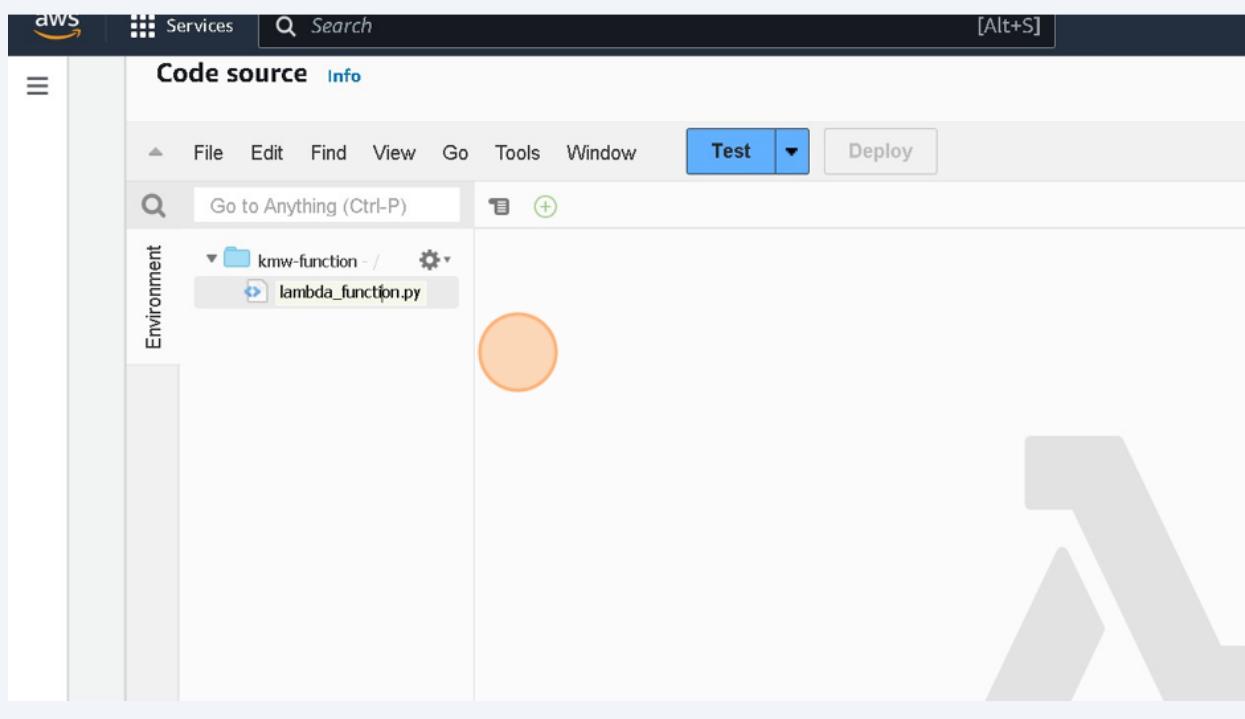
Add

© 2023, Amazon Web Services India Private Limited or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

28 Click here.



29 Click here.



30 Type " trigger DOWN ENTER"

31 Click "Using an Amazon S3 trigger to invoke a Lambda function"

A screenshot of a Google search results page. The search query is "aws lambda s3 trigger tutorial with python". The results show approximately 21,10,000 results in 0.45 seconds. The top result is a link to Amazon.com titled "Using an Amazon S3 trigger to invoke a Lambda function". Below the link, it says "In this tutorial, you use the console to create a Lambda function and configure a trigger for an Amazon Simple Storage Service (Amazon S3) bucket." There are also links for "Prerequisites", "Create an Amazon S3 bucket", and "Create the Amazon S3 trigger". Below the search results, there is a "Videos" section with a thumbnail for a YouTube video titled "AWS S3 File Upload + Lambda Trigger (Tutorial In Python ...)" uploaded by "YouTube · Be A Better Dev".

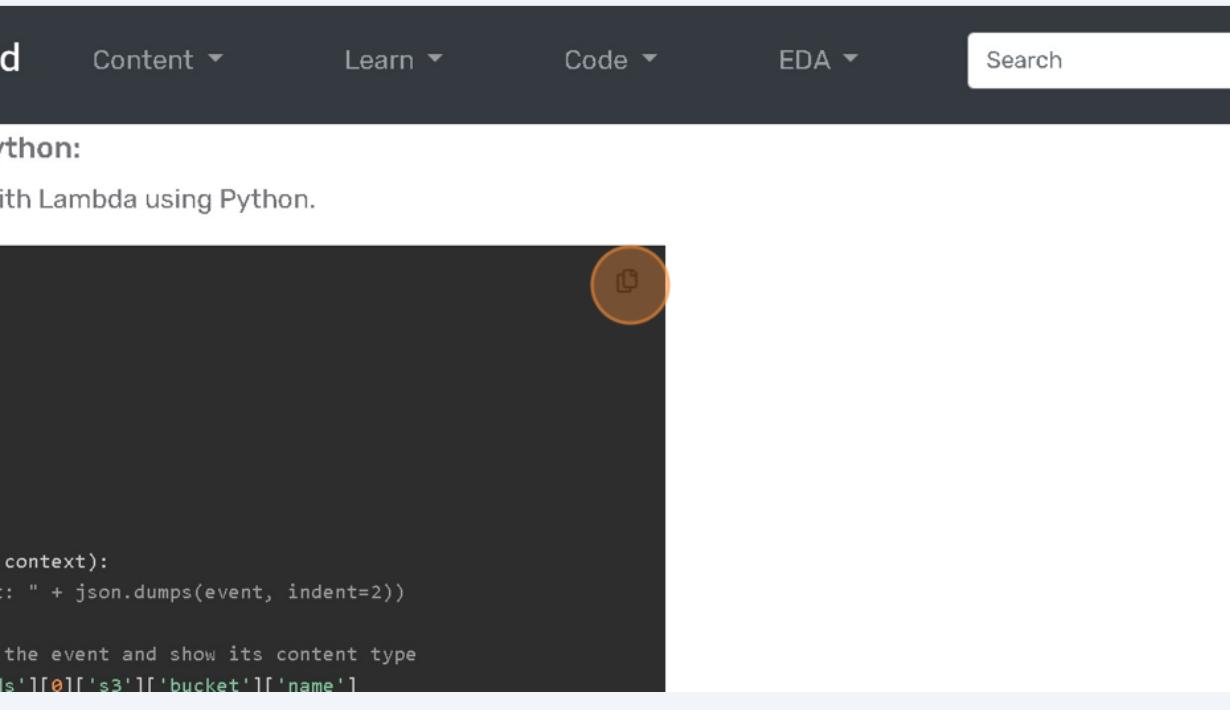
32 Click "Python"

The screenshot shows the AWS Lambda console interface. On the left, there's a sidebar with various navigation links. In the main area, there's a large text block about retrieving object metadata from S3. Below it, a tab navigation bar has several tabs: Node.js, Python (which is highlighted with an orange circle), Java, .NET, Go, and Ruby. Under the Python tab, there's a section titled "Example index.mjs" containing the code `console.log('Loading function');`. There are also icons for copy and like.

33 Click "Amazon S3 trigger to invoke a Lambda function"

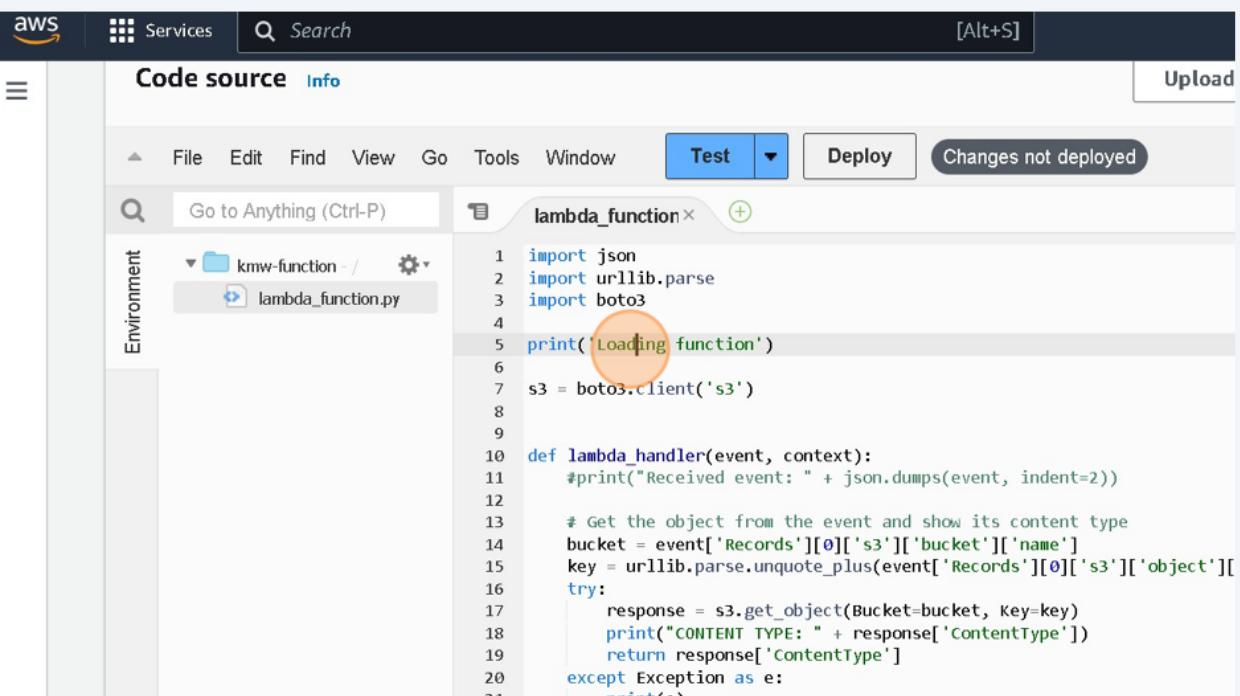
The screenshot shows the AWS Lambda Developer Guide page. At the top, there's a navigation bar with "AWS Lambda > Developer Guide". Below the navigation bar, there's a tab bar with Node.js, Python (highlighted with an orange circle), Java, .NET, Go, and Ruby. A text box contains the message: "For a Python example, see [Amazon S3 trigger to invoke a Lambda function](#) on the Serverless Land website." To the right, there's a sidebar titled "On this page" with a list of links related to Lambda development.

34 Click here.



```
python:  
ith Lambda using Python.  
  
context):  
t: " + json.dumps(event, indent=2))  
  
the event and show its content type  
s'![0]['s3']['bucket']['name']
```

35 Double-click here.



aws Services Search [Alt+S] Upload

File Edit Find View Go Tools Window Test Deploy Changes not deployed

Go to Anything (Ctrl-P) lambda_function

Environment kmw-function / lambda_function.py

```
1 import json
2 import urllib.parse
3 import boto3
4
5 print('Loading function')
6
7 s3 = boto3.client('s3')
8
9
10 def lambda_handler(event, context):
11     #print("Received event: " + json.dumps(event, indent=2))
12
13     # Get the object from the event and show its content type
14     bucket = event['Records'][0]['s3']['bucket']['name']
15     key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'])
16     try:
17         response = s3.get_object(Bucket=bucket, Key=key)
18         print("CONTENT TYPE: " + response['ContentType'])
19         return response['ContentType']
20     except Exception as e:
21         print(e)
```

36 Click here.

The screenshot shows the AWS Lambda function code editor for a function named 'lambda_function'. The code is written in Python and handles an S3 trigger. A specific line of code, which is highlighted with a red oval, is: `key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'])`. This line extracts the key of the S3 object from the event payload. The rest of the code prints the event and attempts to get the object from the S3 bucket.

```
1 import json
2 import urllib.parse
3 import boto3
4
5 print('Lambda Triggered')
6
7 s3 = boto3.client('s3')
8
9
10 def lambda_handler(event, context):
11     #print("Received event: " + json.dumps(event, indent=2))
12
13     # Get the object from the event and show its content type
14     bucket = event['Records'][0]['s3']['bucket']['name']
15     key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')
16
17     try:
18         response = s3.get_object(Bucket=bucket, Key=key)
19         print("CONTENT TYPE: " + response['ContentType'])
20         return response['ContentType']
21     except Exception as e:
22         print(e)
23         print('Error getting object {} from bucket {}. Make sure they exist and your bucket has the correct CORS configuration.'.format(key, bucket))
24         raise e
```

Use the function see triggers, layer destinations to you can see the types of resources visualization:

Triggers are AWS resources that initiate a function.

Destinations are resources that receive of an invocation or failure. You can use Lambda to send records when you invoke asynchronous your function process records from a stream contents of the i

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms

37 Click "Test"

The screenshot shows the AWS Lambda function configuration page for the same 'lambda_function'. At the top, there are tabs for 'Code', 'Test' (which is highlighted with a red oval), 'Monitor', 'Configuration', 'Aliases', and 'Versions'. Below the tabs, there's a 'Code source' section with an 'Info' link and an 'Upload from' button. The main area shows the function code in a code editor. At the bottom of the code editor, there's a toolbar with 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', a 'Test' button (highlighted with a red oval), a 'Deploy' button, and a status message 'Changes not deployed'.

Code source [Info](#) [Upload from](#)

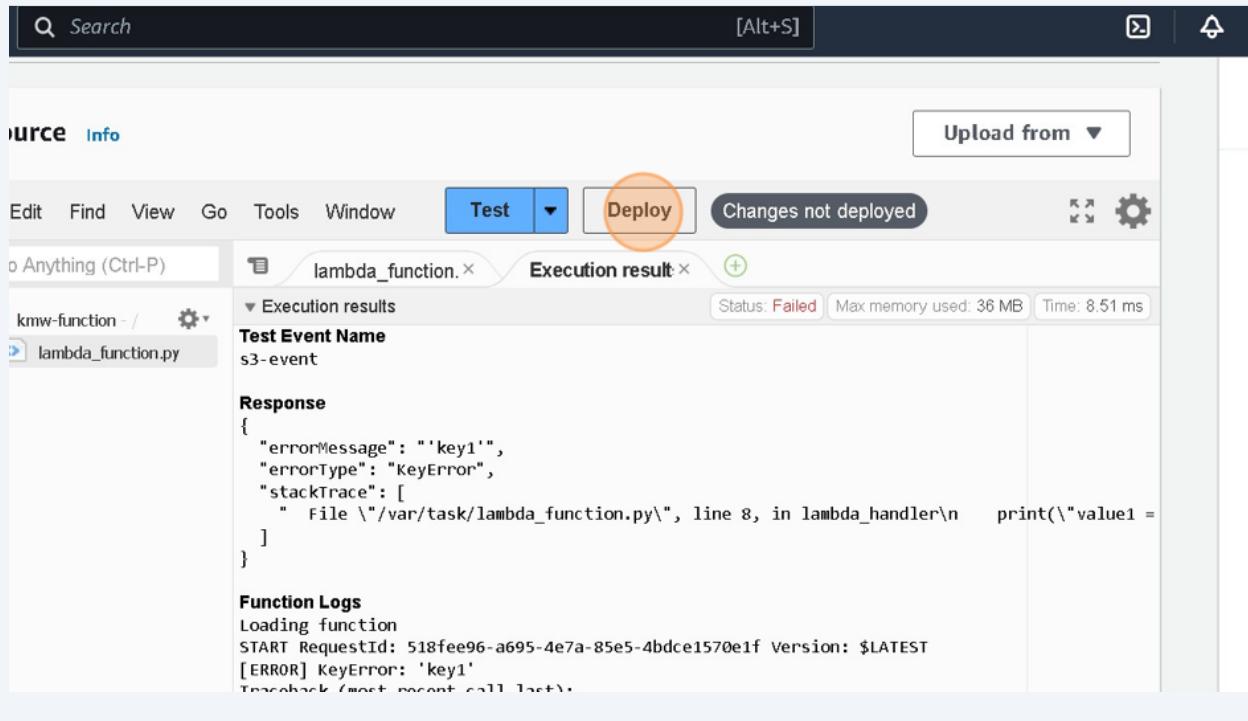
File Edit Find View Go Tools Window **Test** Deploy Changes not deployed

Go to Anything (Ctrl-P)

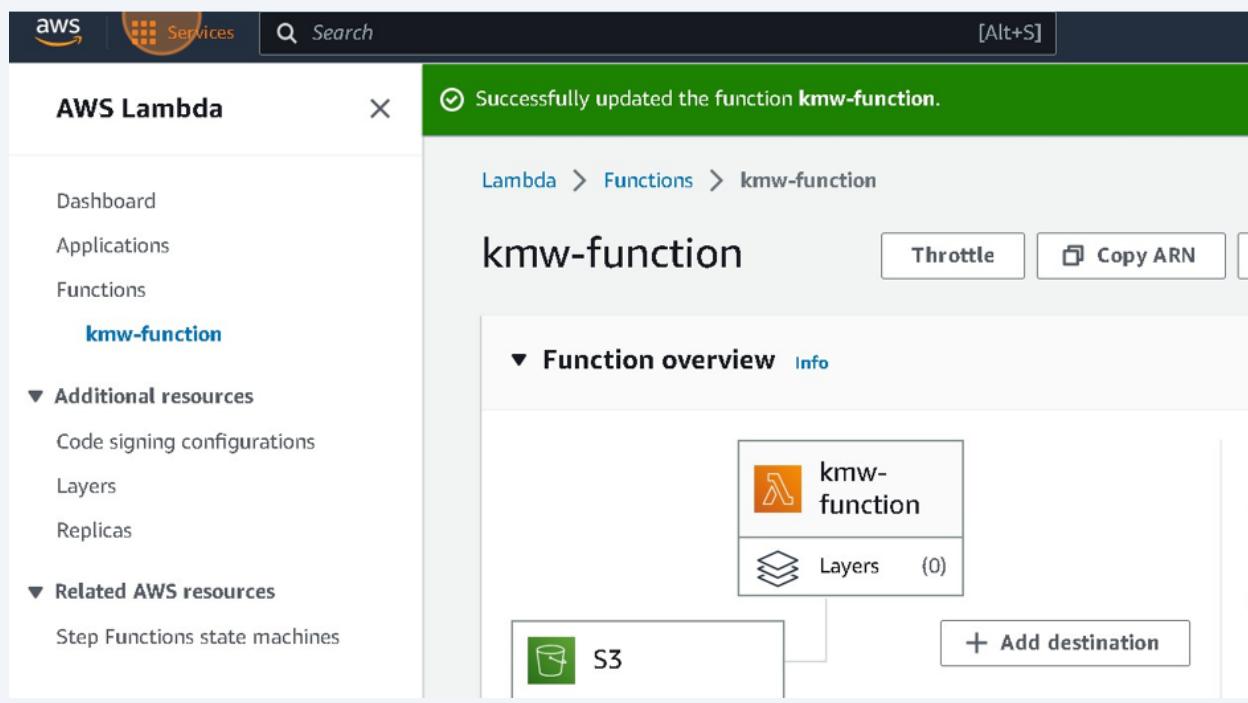
lambda_function

```
1 import json
2 import urllib.parse
3 import boto3
4
5 print('Lambda Triggered')
6
7 s3 = boto3.client('s3')
8
9
10 def lambda_handler(event, context):
11     #print("Received event: " + json.dumps(event, indent=2))
```

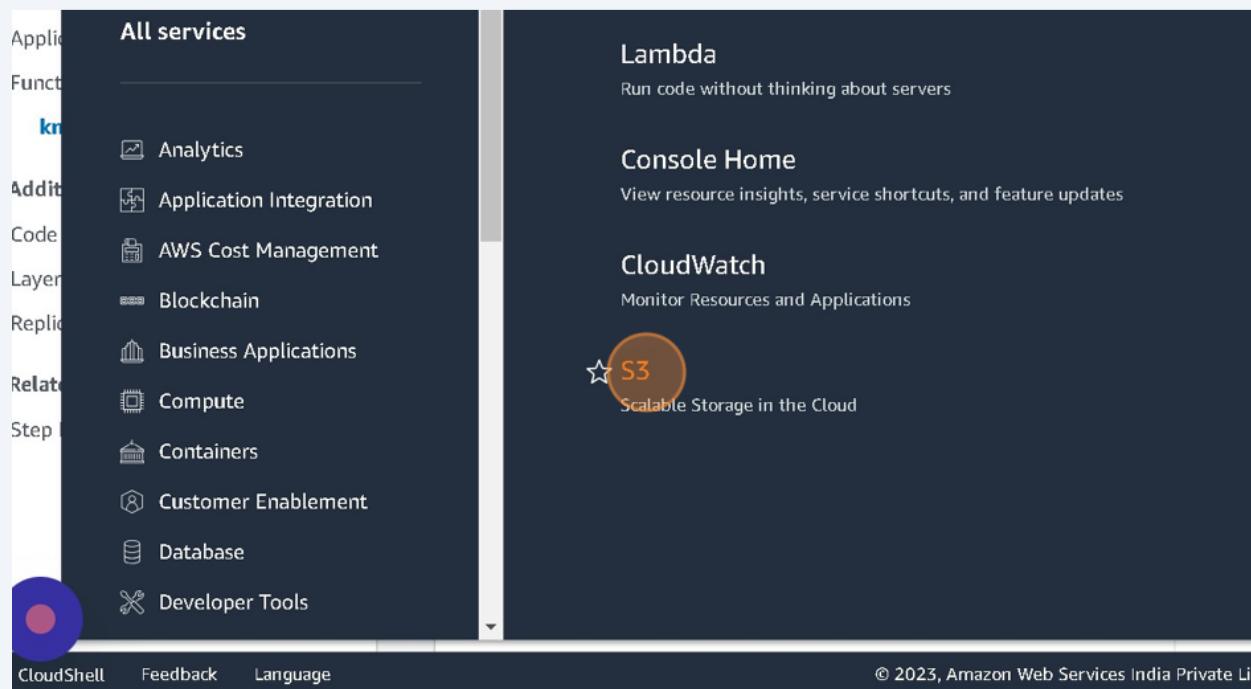
38 Click "Deploy"



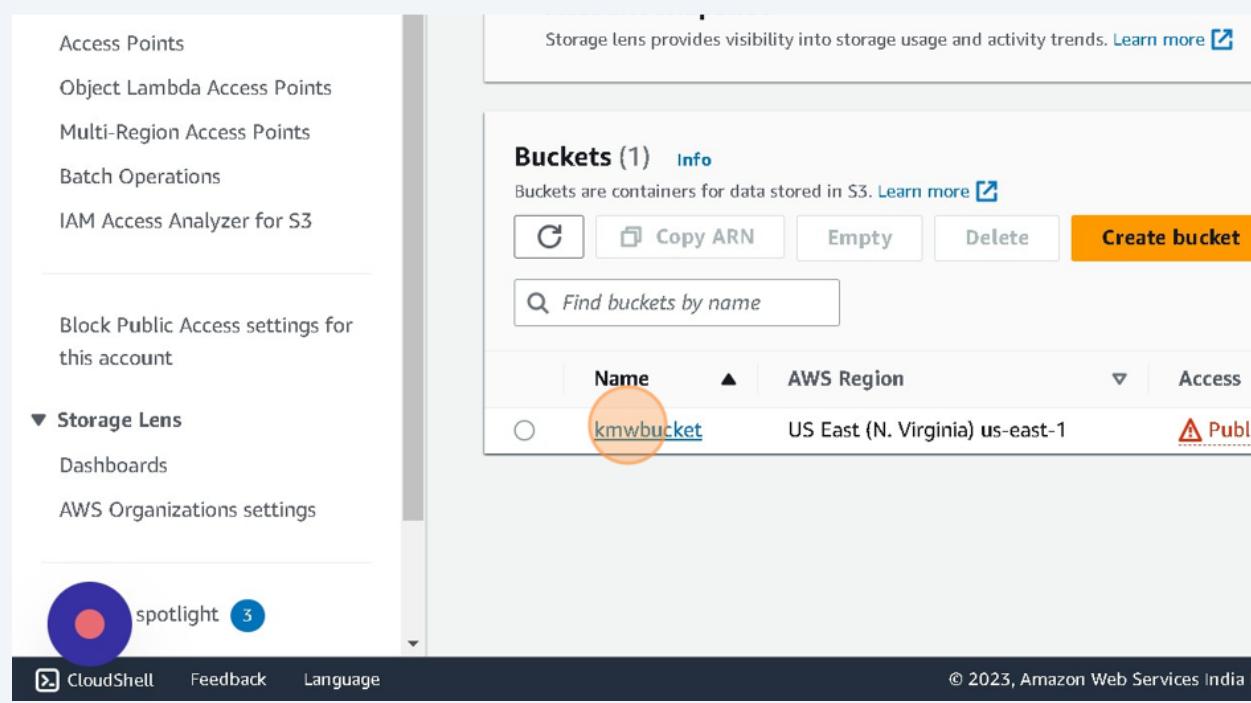
39 Click "Services"



40 Click "S3"



41 Click "kmwbucket"



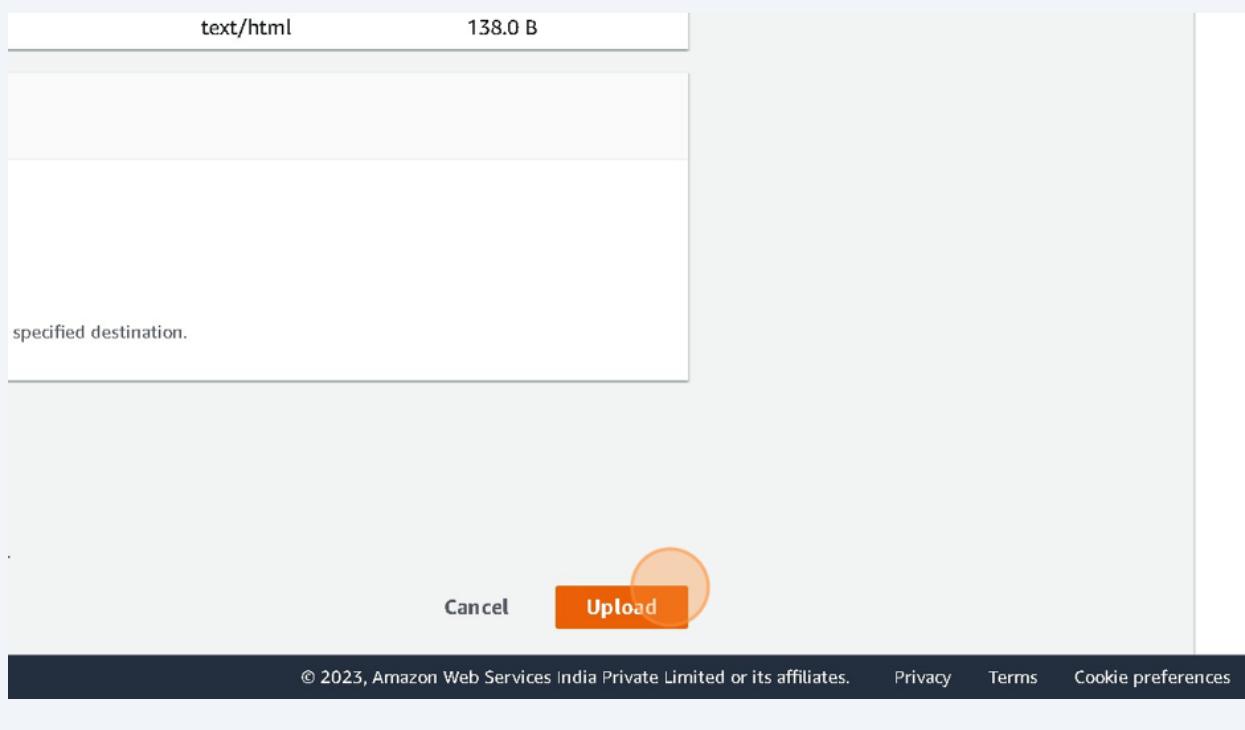
42 Click "Upload"

The screenshot shows the AWS S3 console. The top navigation bar has tabs for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' tab is selected. A red circle highlights the 'Upload' button, which is orange with white text. Other buttons visible include 'Create folder', 'Copy S3 URI', 'Copy URL', 'Download', 'Open', and 'Delete'. Below the buttons is a search bar with the placeholder 'Find objects by prefix'. A table lists one object: 'index.html' (Type: html, Last modified: June 14, 2023, 11:08:24 (UTC+05:30), Size: 180). The left sidebar contains several collapsed sections. At the bottom, there are links for 'Feedback' and 'Language'.

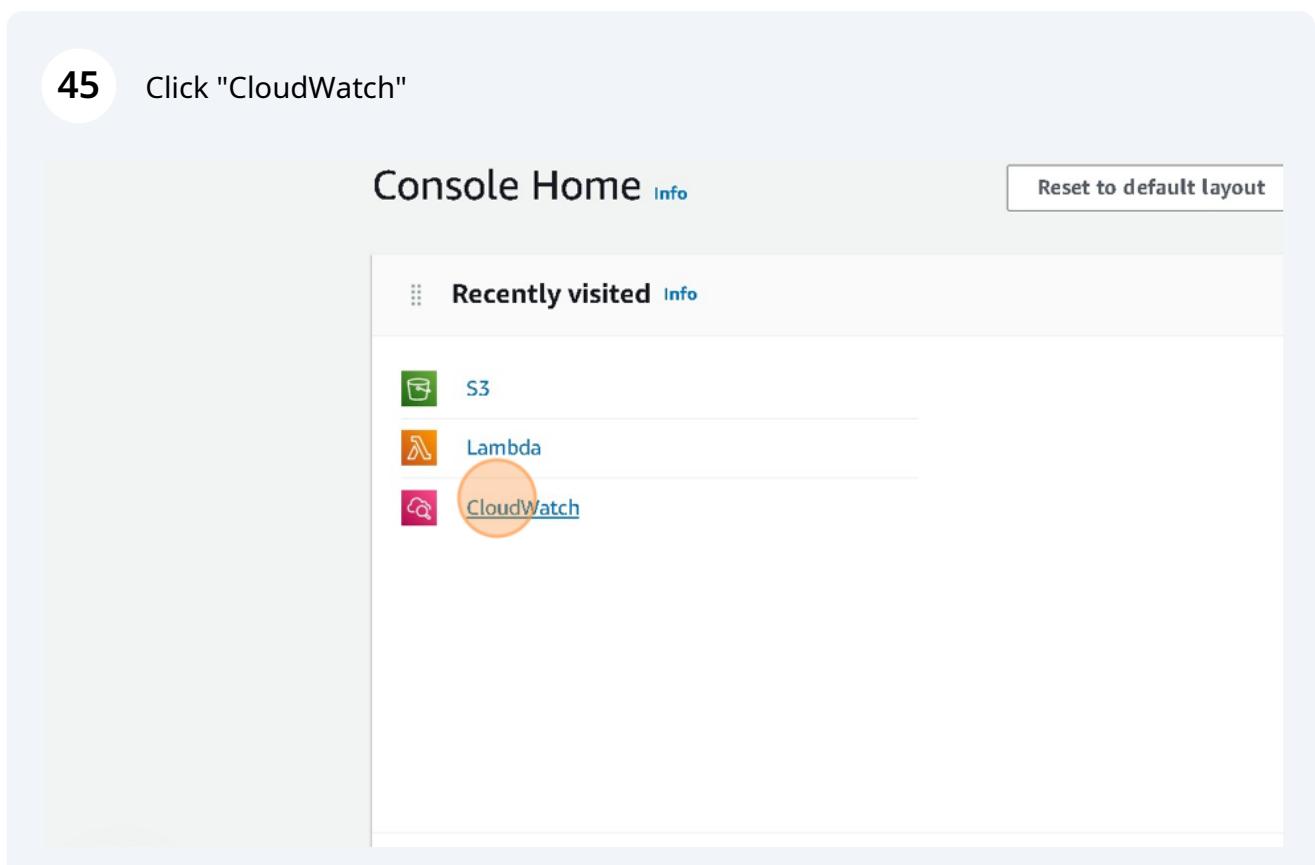
43 Click "Add files"

The screenshot shows the 'Add files' dialog for AWS S3. It displays a message: 'Drop files and folders you want to upload here, or choose Add files or Add folder.' A red circle highlights the 'Add files' button, which is orange with white text. Below the message, it says '1 file will be uploaded.' A preview area shows a single file entry. At the bottom, there is a table header with columns: 'Folder', 'Type', and 'Size'.

44 Click "Upload"



45 Click "CloudWatch"



46 Click here.

The screenshot shows the AWS CloudWatch interface. On the left, a sidebar menu lists various CloudWatch services: Alarms, Logs (which is highlighted with an orange circle), Metrics, X-Ray traces, Events, Application monitoring, and Insights. Below these are Favorites and recent dashboards. At the top right, there is a search bar, a 'CloudWatch' title, and time filters set to '3h'. The main content area is titled 'Get started with CloudWatch' and includes sections for creating alarms and dashboards, along with a 'Create a default dashboard' button.

47 Click "Log groups"

The screenshot shows the AWS CloudWatch Logs Insights interface. The sidebar on the left shows the same navigation as the previous screenshot, with 'Logs' expanded and 'Log groups' highlighted with an orange circle. The main panel is titled 'Logs Insights' and contains a query editor with the following code:

```
1 fields @timestamp, @message, @LogStream, @Log
2 | sort @timestamp desc
3 | limit 20
```

Below the code, there are buttons for 'Run query', 'Cancel', 'Save', and 'History'. A note at the bottom states: 'Queries are allowed to run for up to 60 minutes.'

48 Double-click "/aws/lambda/kmw-function"

Favorites and recents

Dashboards

▶ Alarms ⚠ 0 ✓ 0 ○ 0

▼ Logs

Log groups

Logs Insights

Live tail [New](#)

▶ Metrics

▶ X-Ray traces

▶ Events

▶ Application monitoring

▶ Insights

Log groups (1)

By default, we only load up to 10000 log groups.

C Actions ▾ View in Logs Insights Start tailing C

Filter log groups or try prefix search

<input type="checkbox"/>	Log group	Data pro...
<input type="checkbox"/>	/aws/lambda/kmw-function	-

49 Click "2023/06/14/[\$LATEST]21b7cc2d480642a89944af9ebe2f13fa"

0

Log streams Metric filters Subscription filters Contributor Insights Tags Data

Log streams (5)

C Delete Create log stream

Filter log streams or try prefix search Exact match Show e

<input type="checkbox"/>	Log stream	Last event time
<input type="checkbox"/>	2023/06/14/[\$LATEST]21b7cc2d480642a89944af9ebe2f13fa	2023-06-14 16:55:09 (UTC+05:30)
<input type="checkbox"/>	2023/06/14/[\$LATEST]c087e23c52ae409699ff008a4b2cb847	2023-06-14 16:52:58 (UTC+05:30)
<input type="checkbox"/>	2023/06/14/[\$LATEST]d554040510f84e8ab14768b9f79c7468	2023-06-14 12:26:15 (UTC+05:30)
<input type="checkbox"/>	2023/06/14/[\$LATEST]f46302197f21489fa51fc47f3d5bbfec	2023-06-14 11:30:24 (UTC+05:30)
<input type="checkbox"/>	2023/06/14/[\$LATEST]a5a0baa4da044312aa672f86d0312344	2023-06-14 11:24:33 (UTC+05:30)

ack Language © 2023, Amazon Web Services India Private Limited or its affiliates

50 Logs Generated

C Actions ▾ Start tailing Create metric filter

Filter events Clear 1m 30m 1h 12h Custom Display ▾

Timestamp	Message
	No older events at this moment. Retry
2023-06-14T16:55:08.215+05:30	INIT_START Runtime Version: python:3.7.v27 Runtime Version ARN: arn:aws:lambda:...
2023-06-14T16:55:08.547+05:30	Lambda Triggered
2023-06-14T16:55:08.738+05:30	START RequestId: 27146c5f-d7c3-40cf-8334-39d82bdb97b4 Version: \$LATEST
2023-06-14T16:55:09.027+05:30	CONTENT TYPE: text/html
2023-06-14T16:55:09.043+05:30	END RequestId: 27146c5f-d7c3-40cf-8334-39d82bdb97b4
2023-06-14T16:55:09.043+05:30	REPORT RequestId: 27146c5f-d7c3-40cf-8334-39d82bdb97b4 Duration: 304.14 ms Billed...
	No newer events at this moment. Auto retry paused. Resume

© 2023, Amazon Web Services India Private Limited or its affiliates. [Privacy](#) [Terms](#)