

Getting started with FaaS : An Introduction to AWS Lambda

Madhura Chaganty

Engineering Manager @Paythru



Key Topics

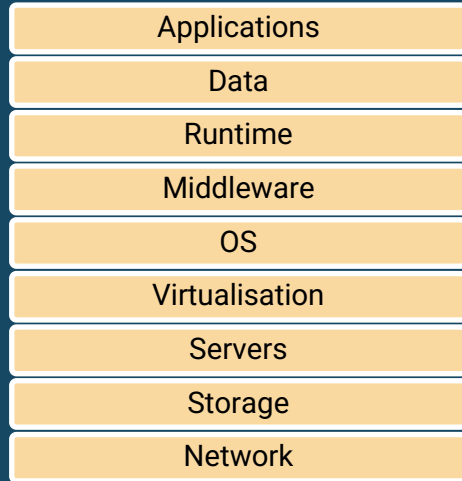


- Serverless Computing
- Function as a Service
- AWS Lambda overview
- Invocation models and execution environment
- Managing Lambda functions
- Best practices

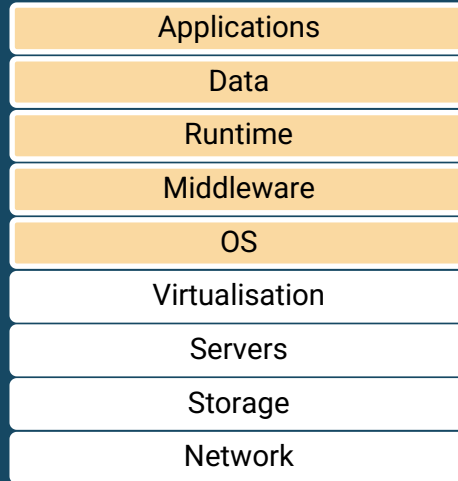


Understanding Serverless Computing

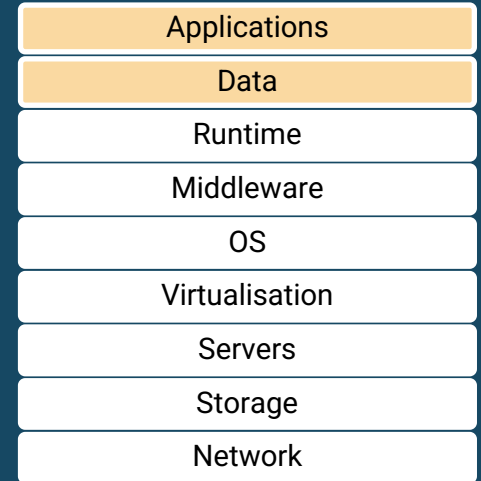
On-Premises



Cloud Computing



Serverless Computing



○ You manage

○ Cloud provider manages

Four Technical Criteria For Serverless






- ✓ **No infrastructure provisioning**
- ✓ **Automatic scaling**
- ✓ **Pay for value (actual compute time)**
- ✓ **Highly available and secure**








Serverless Computing Is Not Just FaaS



Compute & Access

-  Computing engine for Containers
-  API Gateway
-  **Function As A Service**
-  Application Integration Service
-  Serverless Event Bus




Storage

-  Key-values Database
-  NoSQL Database
-  Document Storage

User & Identity Management

-  Authentication
-  IAM

Messaging & Streaming

-  Queue service
-  Notification service
-  Data streams



Function As A Service



Key points

Function you write

Event driven execution (e.g. file uploaded to s3)

No infrastructure provisioning

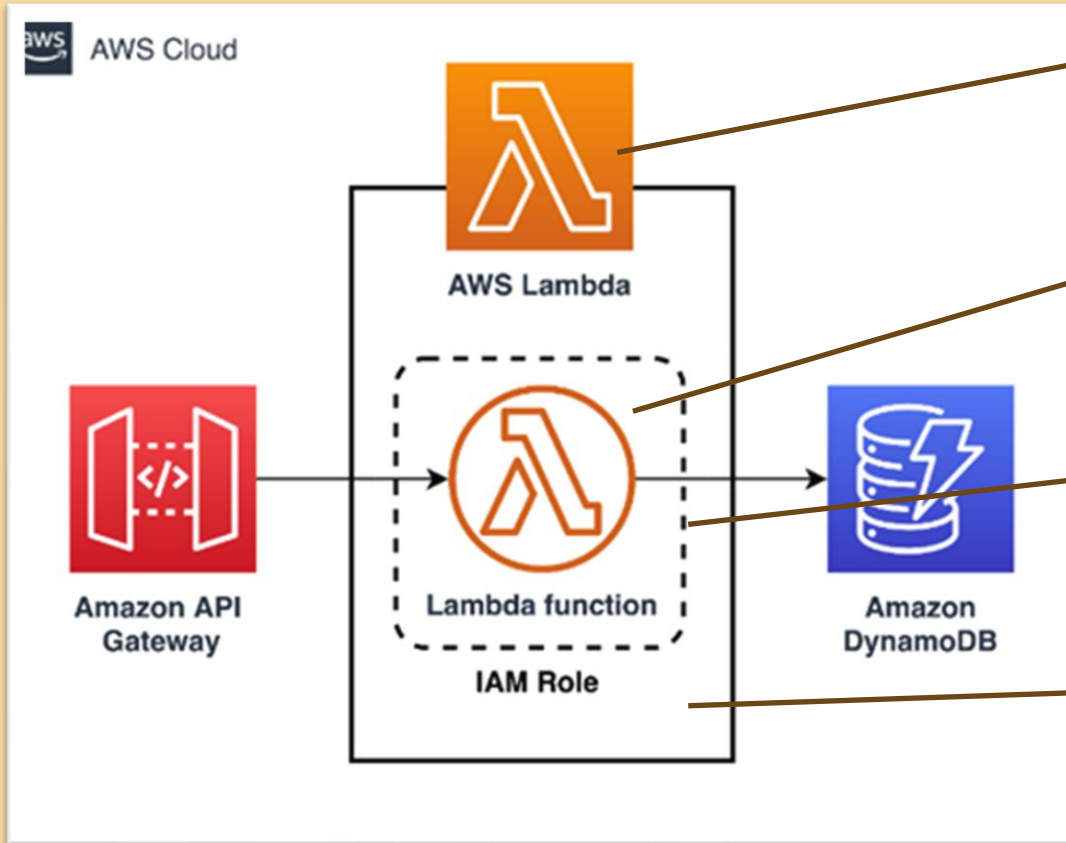
Automatic scaling

Pay per execution

High availability and security



AWS Lambda



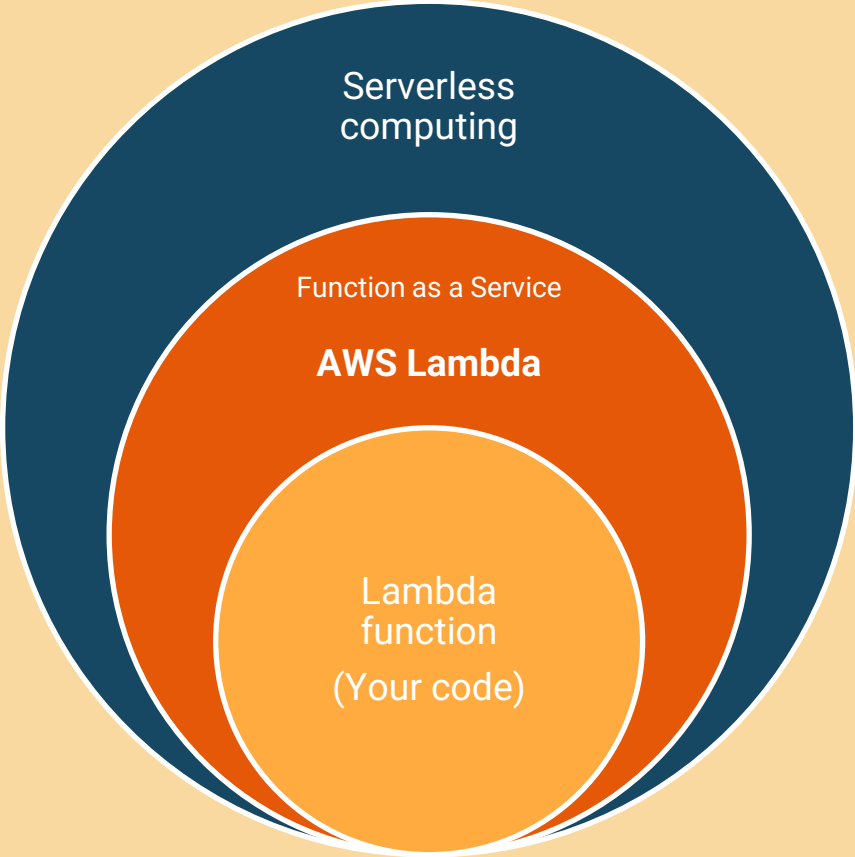
Serverless computing service managed by AWS

You write code in any of the supported languages (Python, Java, Go, node.js ...)

Deploy it to AWS Lambda

AWS Lambda executes your function on a trigger

So far..



Where's AWS Lambda?



The screenshot shows the AWS console search interface. At the top left, the 'aws' logo and 'Services' menu are visible. A search bar contains the text 'lambda'. Below the search bar, the results are categorized into 'Services (5)', 'Features (2)', 'Blogs (853)', and 'Documentation (63,586)'. The 'Services' section is expanded, showing the 'Lambda' service with its icon and the tagline 'Run Code without Thinking about Servers'. Red and blue boxes highlight the search bar and the Lambda service result, respectively.

aws Services

Q lambda

Search results for 'lambda'


Services (5)

Features (2)

Blogs (853)

Documentation (63,586)

Services

 **Lambda** ☆

Run Code without Thinking about Servers

Where's AWS Lambda?



Lambda ×

Resources for Europe (London) Create function

Lambda function(s)	Code storage	Full account concurrency	Unreserved account concurrency
0	0 byte (0% of 75 GB)	1000	1000

Account-level metrics
The charts below show metrics across all your Lambda functions in this AWS Region.

1h **3h** 12h 1d 3d 1w Custom 📅 UTC timezone ▼ 🔄 ▼ ⋮

Error count and success rate 📘 ⋮ Throttles 📘 ⋮ Invocations 📘 ⋮



Ways to create new Lambda function

AWS console

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 using a configuration preset.

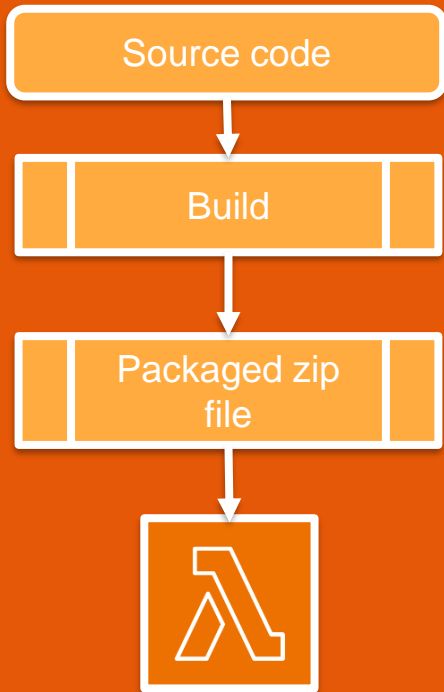
Basic information

Function name
Enter a name that describes the purpose of your function.

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

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Java.

Upload deployment package



CLI command and s3 bucket

```
aws lambda create-function --function-name EncryptPDF \
--zip-file fileb://lambda_function.zip --handler lambda_function.lambda_handler \
--runtime python3.12 --timeout 15 --memory-size 256 \
--role arn:aws:iam::123456789012:role/LambdaS3Role \
--region us-west-2 \
--logging-config LogFormat=JSON
```



Invoking the Lambda function

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
```

Entry point

Incoming Request

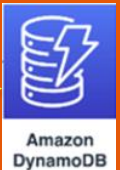


Triggering event

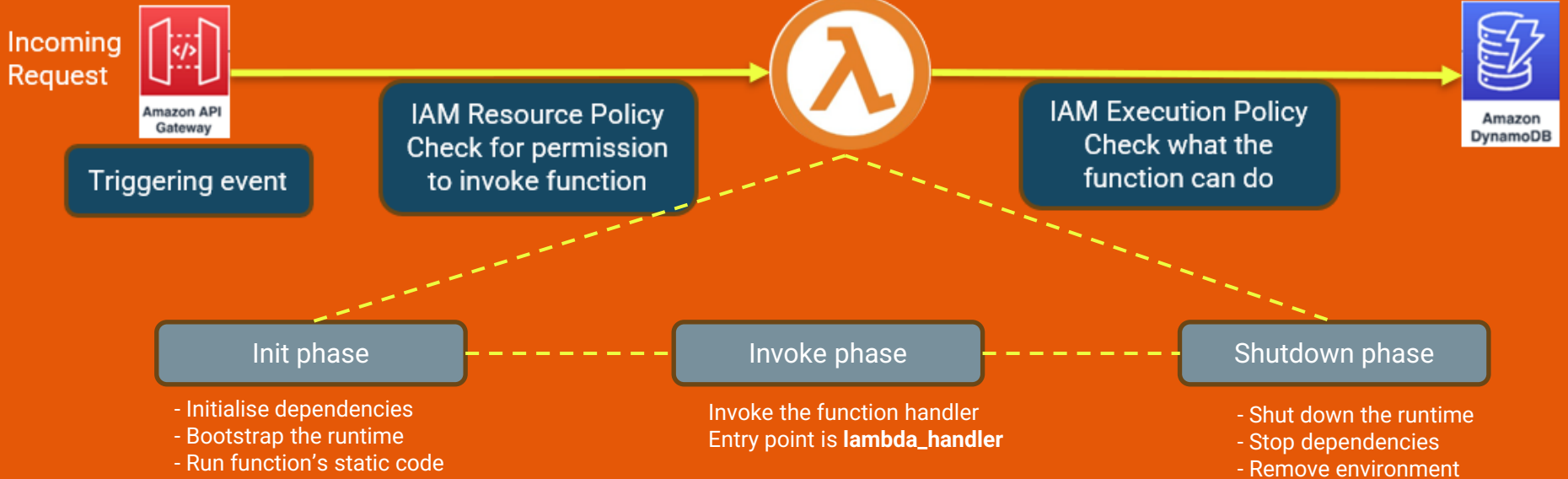
IAM Resource Policy
Check for permission to
invoke function



IAM Execution Policy
Check what the function
can do



Execution lifecycle



Cold start – New execution environment is required to run the lambda function

Warm start – Lambda service retains the environment instead of destroying it

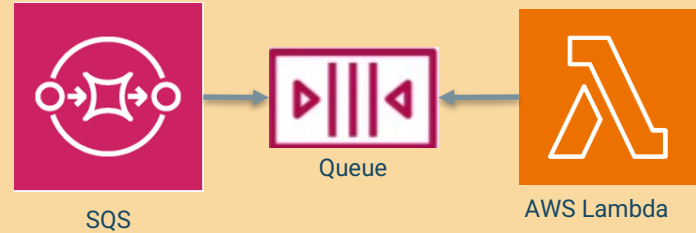


Understanding invocations models

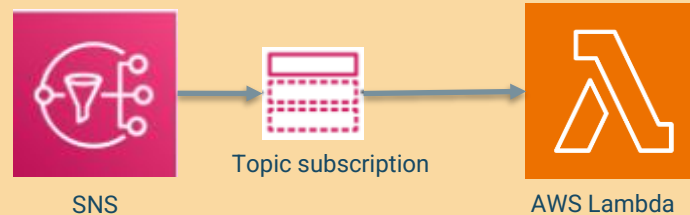
Synchronous invocation



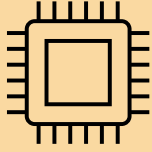
Polling invocation



Asynchronous invocation



Configuring Lambda function



Memory

Max memory = 10GB



Timeout

Max Timeout = 15 mins



Concurrency

Max Concurrency = 1000



Charges proportional to memory and function duration (GB-secs)
Billing based on runtime in 1ms increments

Basic settings [Info](#)

Description - *optional*

Memory [Info](#)

Your function is allocated CPU proportional to the memory configured.

 MB

Set memory to between 128 MB and 10240 MB

Ephemeral storage [Info](#)

You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#)

 MB

Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

SnapStart [Info](#)

Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#)

Supported runtimes: Java 11, Java 17, Java 21.

Timeout

 min sec

Pricing

- Allocated memory 512 MB
- No. of invocations 20,000 times/month
- Execution duration 1 sec



<https://calculator.aws/#/createCalculator/Lambda>



Free Tier

The Lambda free tier includes 1M free requests per month and 400,000 GB-seconds

▼ Show calculations

Unit conversions

Amount of memory allocated: $512 \text{ MB} \times 0.0009765625 \text{ GB in a MB} = 0.5 \text{ GB}$

Amount of ephemeral storage allocated: $512 \text{ MB} \times 0.0009765625 \text{ GB in a MB} = 0.5 \text{ GB}$

Pricing calculations

$20,000 \text{ requests} \times 1 \text{ ms} \times 0.001 \text{ ms to sec conversion factor} = 20.00 \text{ total compute (seconds)}$

$0.50 \text{ GB} \times 20.00 \text{ seconds} = 10.00 \text{ total compute (GB-s)}$

$10.00 \text{ GB-s} - 400000 \text{ free tier GB-s} = -399,990.00 \text{ GB-s}$

$\text{Max} (-399990.00 \text{ GB-s}, 0) = 0.00 \text{ total billable GB-s}$

Tiered price for: 0.00 GB-s

Total tier cost = 0.00 USD (monthly compute charges)

$20,000 \text{ requests} - 1000000 \text{ free tier requests} = -980,000 \text{ monthly billable requests}$

$\text{Max} (-980000 \text{ monthly billable requests}, 0) = 0.00 \text{ total monthly billable requests}$

$0.50 \text{ GB} - 0.5 \text{ GB (no additional charge)} = 0.00 \text{ GB billable ephemeral storage per function}$

Lambda costs - With Free Tier (monthly): 0.00 USD

Monitoring and logging



Code | Test | **Monitor** | Configuration | Aliases | Versions

Monitor [Info](#)

[View CloudWatch logs](#)

[View X-Ray traces](#)

[View Lambda Insights](#)

[View CodeGuru profiles](#)

Filter metrics by **Function** ▼

Alarm recommendations [?](#)

1h

3h

12h

1d

3d

1w

Custom

UTC timezone ▼

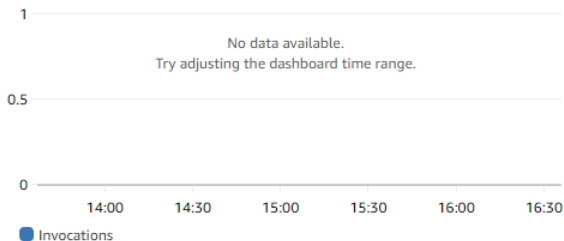


CloudWatch metrics

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**.

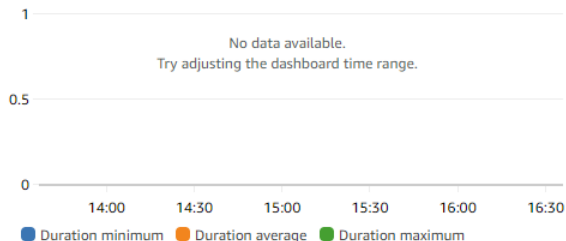
Invocations

No unit



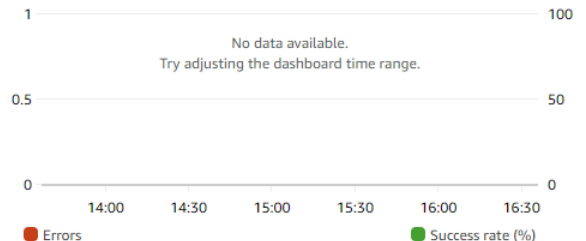
Duration

No unit



Error count and success rate (%)

No unit



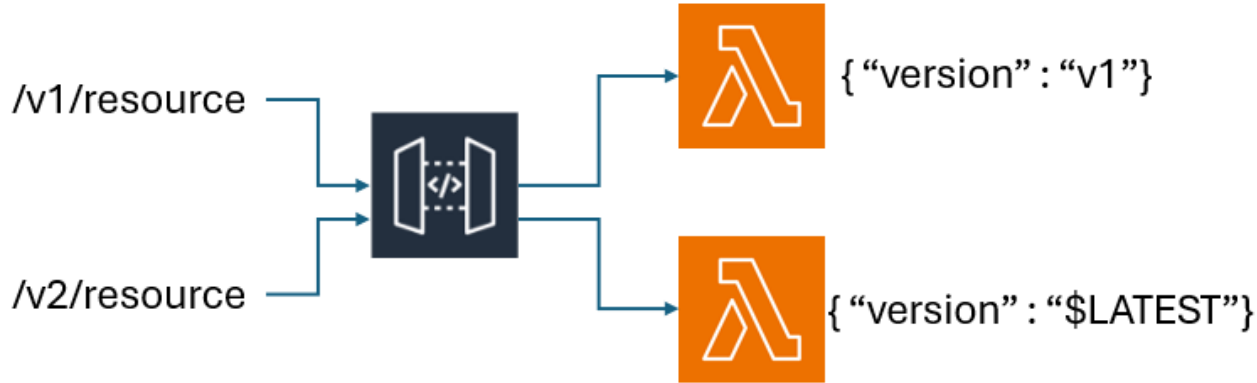
Environment variables

- Adjust behaviour without updating code
- Stored as key-value pair
- Validate values in your function
- Supports encryption for security



ENVIRONMENT	DEVELOPMENT	Remove
databaseHost	lambdadb	Remove

Versioning



- Manage deployments, publish beta version without affecting live
- Unpublished version is `$(LATEST)` (new code)
- Version numbers are never reused
- Ability to define aliases (e.g. version 1 (alias prod), version 2 (alias test))

Best practices



- **Separate business logic**
- **Write modular functions**
- **Keep the functions stateless**
- **Minimise the size of deployment package**
- **Including logging statements**
- **Use environment variables**
- **Avoid recursion**
- **Return response codes (e.g. 200 ok)**

Resources



AWS Lambda docs

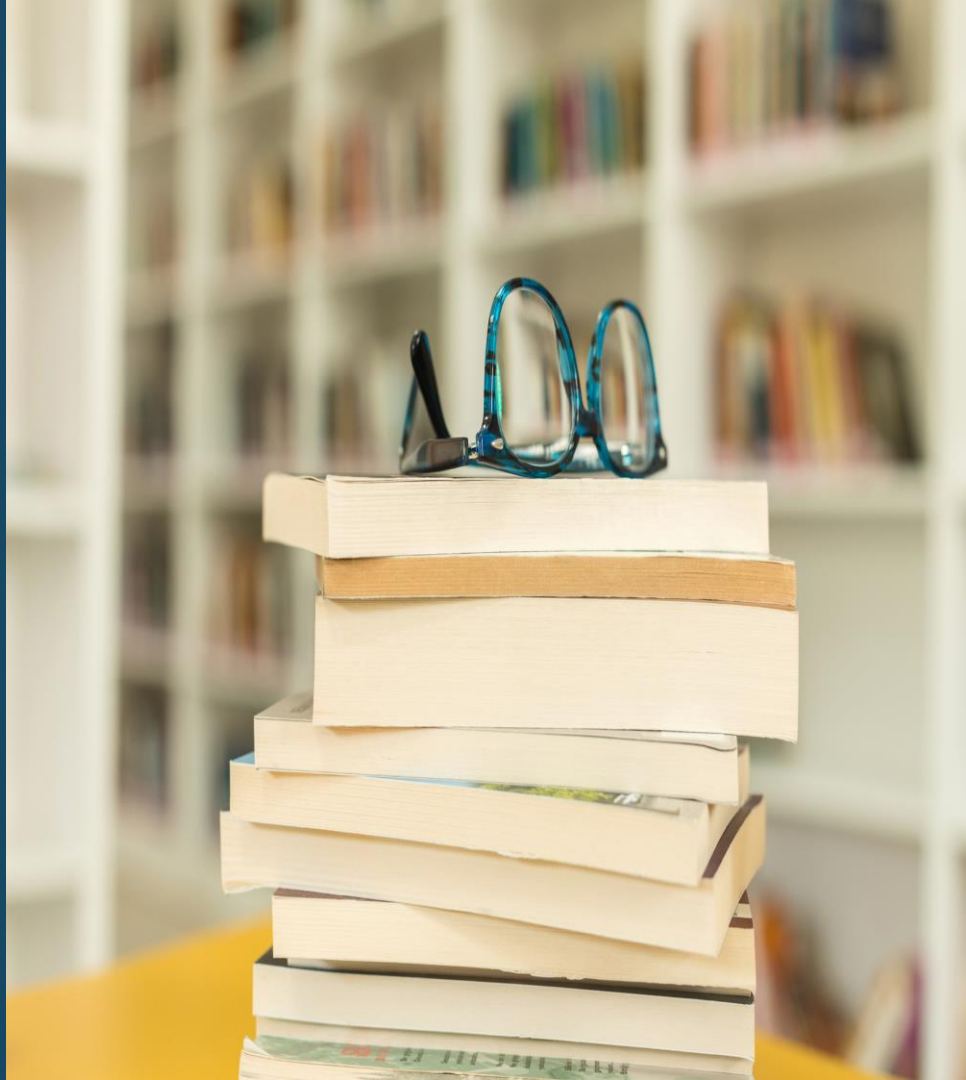


Run Serverless 'Hello world'
with AWS Lambda

Books

AWS Lambda in Action: Event-driven serverless applications
By Danilo Poccia

Serverless Architectures on AWS: With examples using AWS Lambda
By Peter Sbarski



Thank you!



Any questions?

