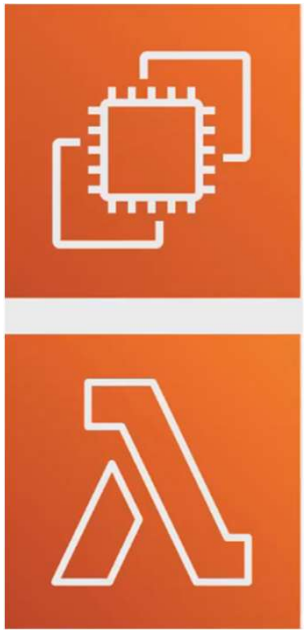
AWS Gyakorlat
Lambda

Felde Imre

AWS Lambda



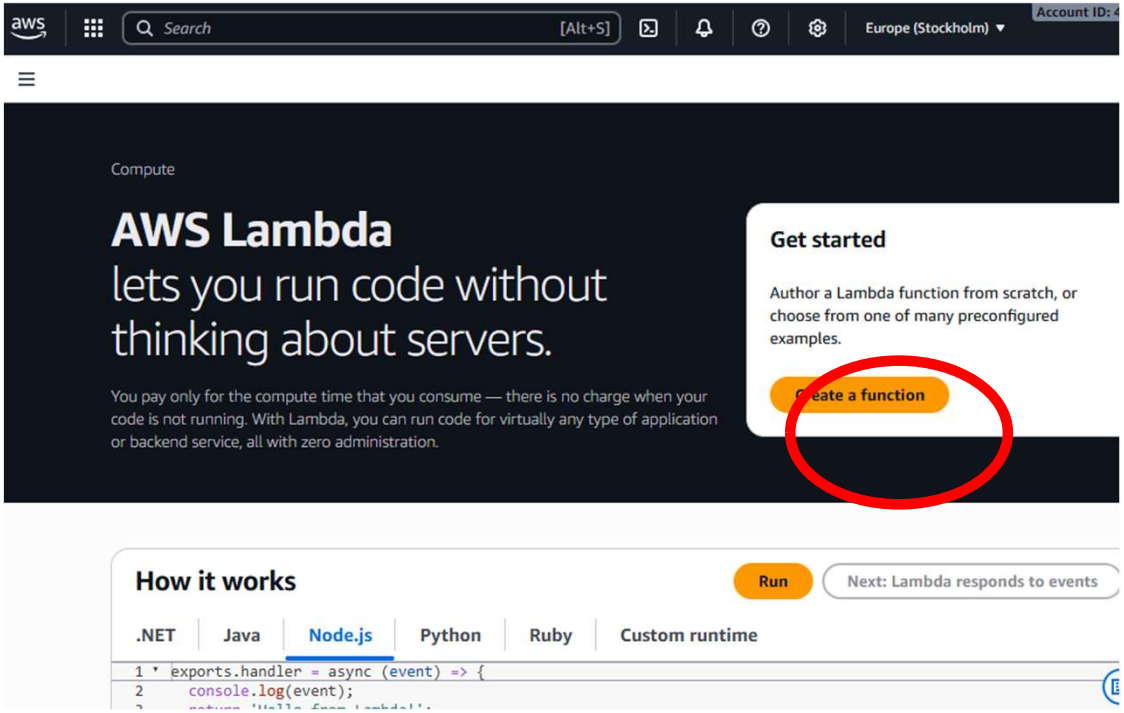
- Virtuális szerverek a felhőben
- RAM és CPU korlátozza
- Folyamatosan futnak
- A skálázás azt jelenti, hogy szervereket lehet hozzáadni vagy eltávolítani

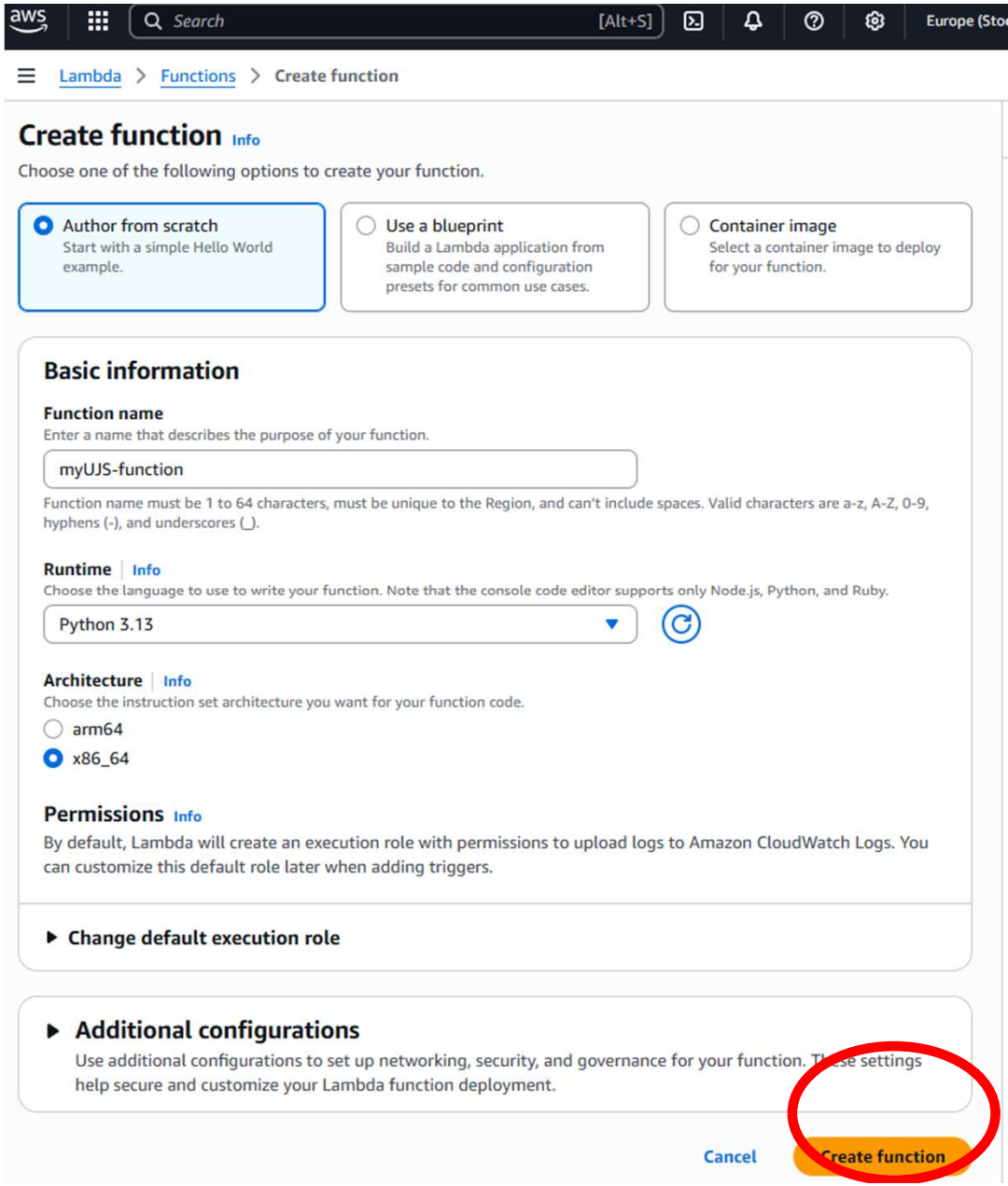
- Virtuális függvények – nincs szükség szerverek kezelésére
- Időkorlátosak – rövid futásidejűek
- Igény szerint futnak
  A skálázás automatikus!

AWS Lambda előnyök

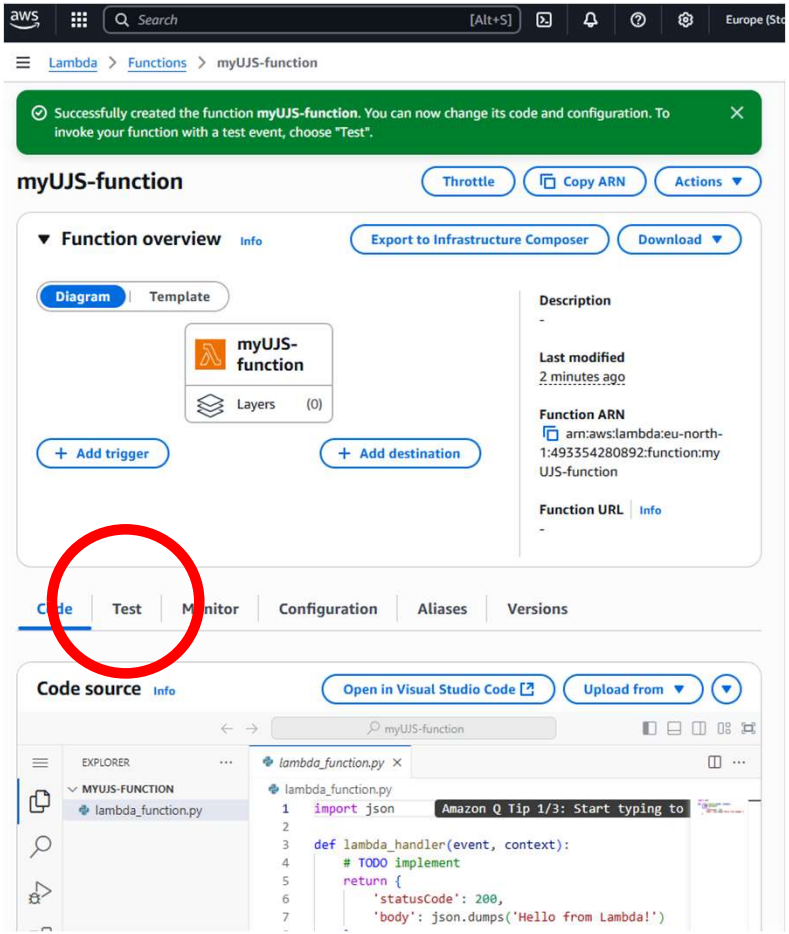
- **Egyszerű árképzés**
  - Fizetés kérésenként és számítási idő alapján
  - Ingyenes szint: 1 000 000 AWS Lambda kérés és 400 000 GB számítási idő
- Teljesen integrált az egész AWS Stack-kel
- Sok programozási nyelvvel integrálható
- Könnyen monitorozható az AWS CloudWatch segítségével
- Egyszerűen növelhetők az erőforrások függvényenként (akár 1,5 GB RAM-ig!)
- A RAM növelése javítja a CPU és a hálózat teljesítményét is!

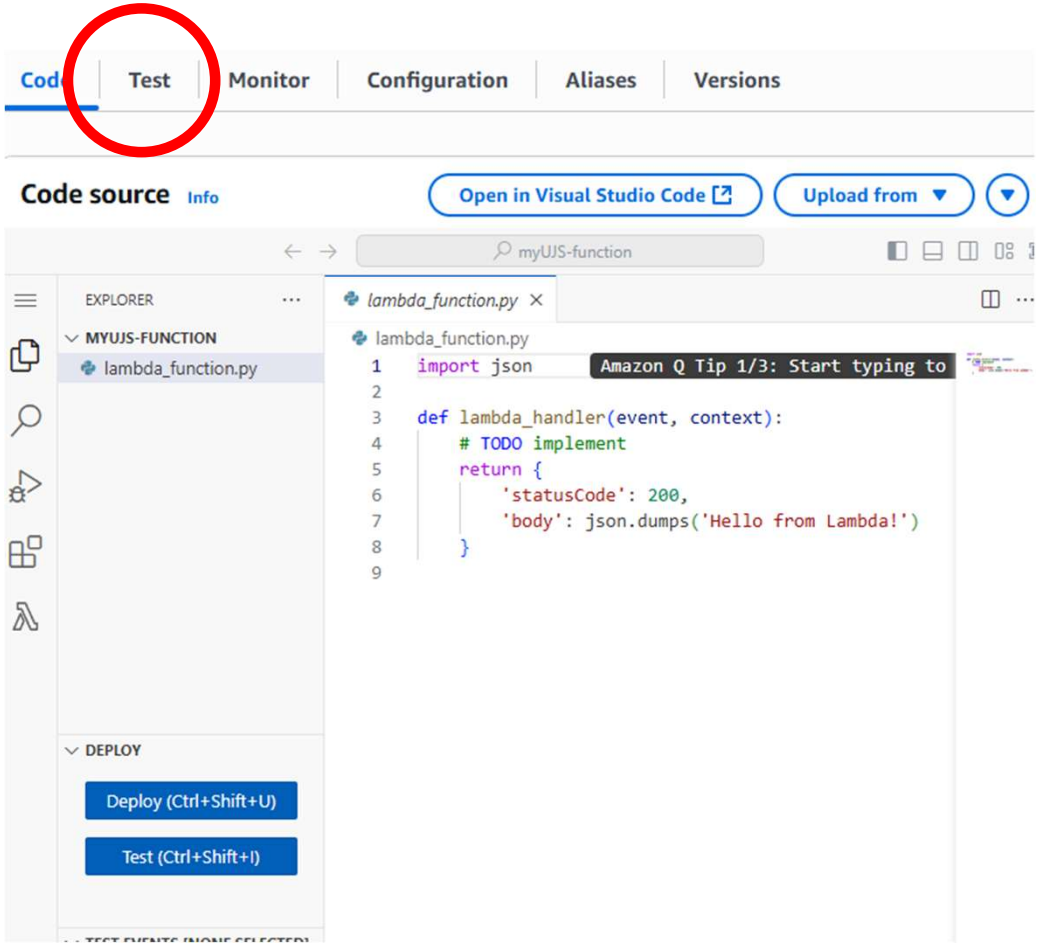
AWS Lambda


aws
Search
[Alt+S]
Europe (Stockholm)
Compute
AWS Lambda
lets you run code without thinking about servers.
You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.
Get started
Author a Lambda function from scratch, or choose from one of many preconfigured examples.
Create a function
How it works
Run
Next: Lambda responds to events
.NET
Java
Node.js
Python
Ruby
Custom runtime
exports.handler = async (event) => {
console.log(event);

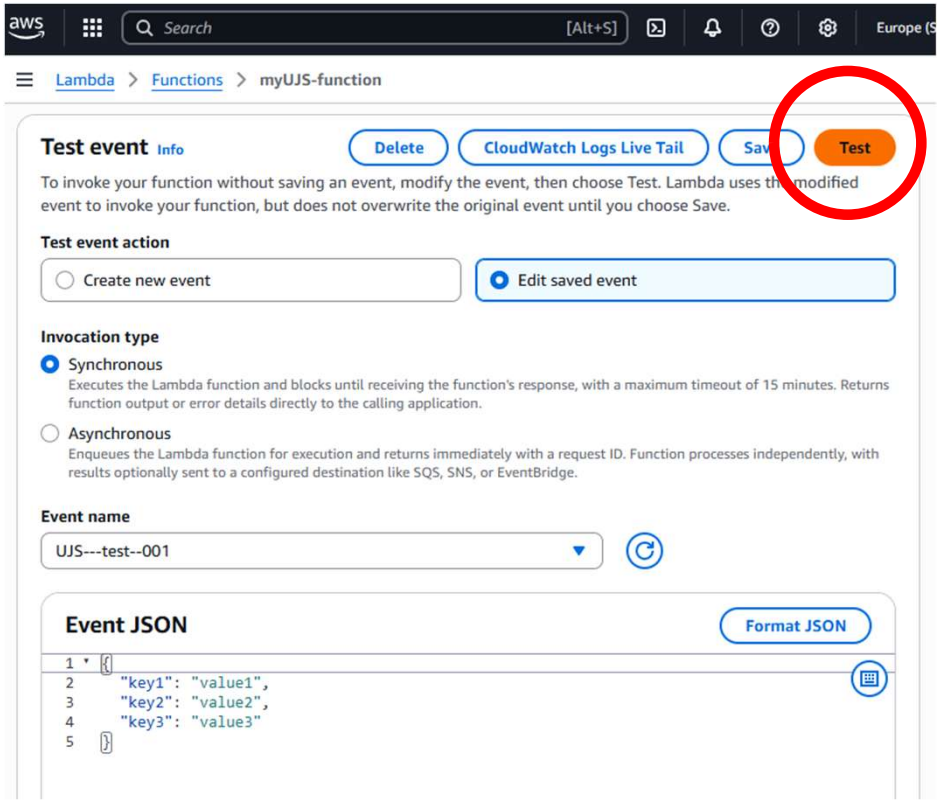

aws
Search
[Alt+S]
Lambda
Functions
Create function
Create function
Info
Choose one of the following options to create your function.
Author from scratch
Start with a simple Hello World example.
Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.
Container image
Select a container image to deploy for your function.
Basic information
Function name
Enter a name that describes the purpose of your function.
myUJS-function
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).
Runtime
Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.13
Architecture
Info
Choose the instruction set architecture you want for your function code.
arm64
x86_64
Permissions
Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.
Change default execution role
Additional configurations
Use additional configurations to set up networking, security, and governance for your function.
help secure and customize your Lambda function deployment.
Cancel
Create function

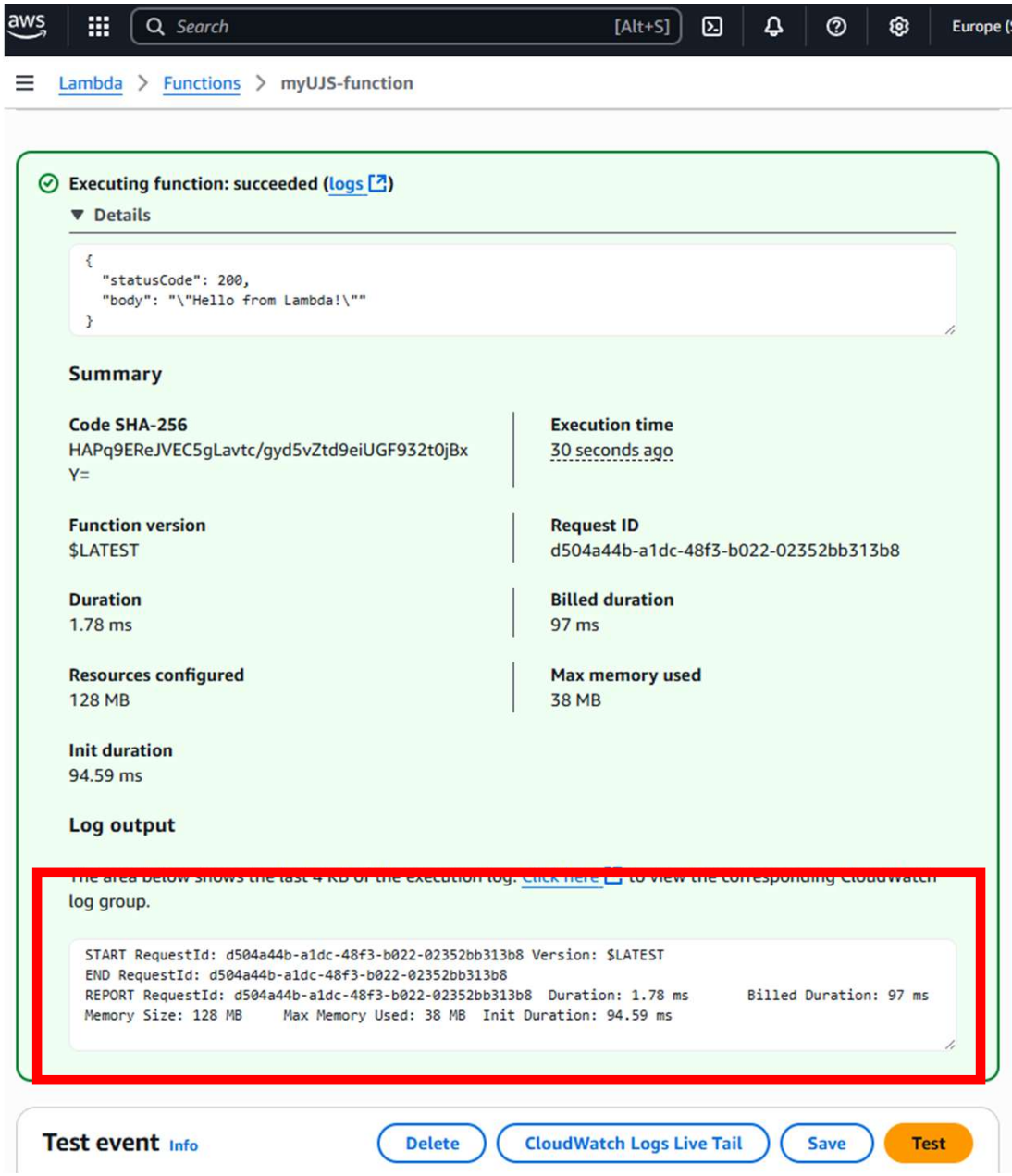
Lamda test


Search
[Alt+S]
Europe (Sto
Lambda > Functions > myUJS-function
Successfully created the function myUJS-function. You can now change its code and configuration. To invoke your function with a test event, choose "Test".
myUJS-function
Throttle
Copy ARN
Actions
Function overview Info
Export to Infrastructure Composer
Download
Diagram
Template
myUJS-function
Layers (0)
+ Add trigger
+ Add destination
Description
-
Last modified
2 minutes ago
Function ARN
arn:aws:lambda:eu-north-1:493354280892:function:myUJS-function
Function URL Info
-
Code
Test
Monitor
Configuration
Aliases
Versions
Code source Info
Open in Visual Studio Code
Upload from
myUJS-function
EXPLORER
MYUJS-FUNCTION
lambda_function.py
lambda_function.py
import json
Amazon Q Tip 1/3: Start typing to
def lambda_handler(event, context):
# TODO implement
return {
'statusCode': 200,
'body': json.dumps('Hello from Lambda!')


Code
Test
Monitor
Configuration
Aliases
Versions
Code source Info
Open in Visual Studio Code
Upload from
myUJS-function
EXPLORER
MYUJS-FUNCTION
lambda_function.py
lambda_function.py
import json
Amazon Q Tip 1/3: Start typing to
def lambda_handler(event, context):
# TODO implement
return {
'statusCode': 200,
'body': json.dumps('Hello from Lambda!')
}
DEPLOY
Deploy (Ctrl+Shift+U)
Test (Ctrl+Shift+I)

Lambda Test


aws
Search
[Alt+S]
Europe (S
Lambda > Functions > myUJS-function
Test event Info
Delete
CloudWatch Logs Live Tail
Sav
Test
To invoke your function without saving an event, modify the event, then choose Test. Lambda uses th modified event to invoke your function, but does not overwrite the original event until you choose Save.
Test event action
Create new event
Edit saved event
Invocation type
Synchronous
Executes the Lambda function and blocks until receiving the function's response, with a maximum timeout of 15 minutes. Returns function output or error details directly to the calling application.
Asynchronous
Enqueues the Lambda function for execution and returns immediately with a request ID. Function processes independently, with results optionally sent to a configured destination like SQS, SNS, or EventBridge.
Event name
UJS---test--001
Event JSON
Format JSON
1 {
2 "key1": "value1",
3 "key2": "value2",
4 "key3": "value3"
5 }


aws
Search
[Alt+S]
Europe (
Lambda > Functions > myUJS-function
Executing function: succeeded (logs)
Details
{
"statusCode": 200,
"body": "\"Hello from Lambda!\""
}
Summary
Code SHA-256
HAPq9EReJVEC5gLavtc/gyd5vZtd9eiUGF932t0jBxY=
Execution time
30 seconds ago
Function version
$LATEST
Request ID
d504a44b-a1dc-48f3-b022-02352bb313b8
Duration
1.78 ms
Billed duration
97 ms
Resources configured
128 MB
Max memory used
38 MB
Init duration
94.59 ms
Log output
log group.
START RequestId: d504a44b-a1dc-48f3-b022-02352bb313b8 Version: $LATEST
END RequestId: d504a44b-a1dc-48f3-b022-02352bb313b8
REPORT RequestId: d504a44b-a1dc-48f3-b022-02352bb313b8 Duration: 1.78 ms Billed Duration: 97 ms
Memory Size: 128 MB Max Memory Used: 38 MB Init Duration: 94.59 ms
Test event Info
Delete
CloudWatch Logs Live Tail
Save
Test