



INHOUDSTAFEL

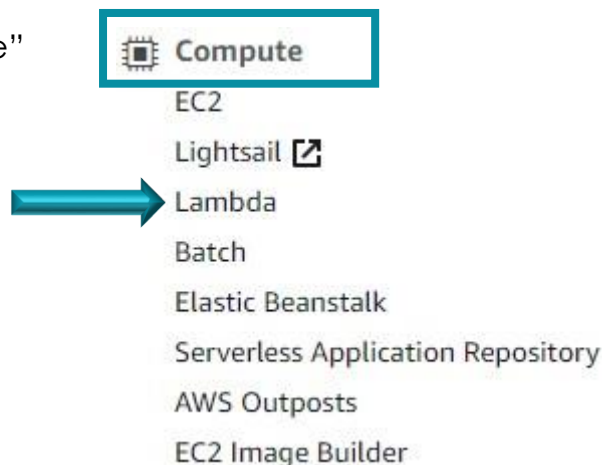
1	Doelstelling.....	3
2	Lambda functie toevoegen aan Aws	3

1 DOELSTELLING

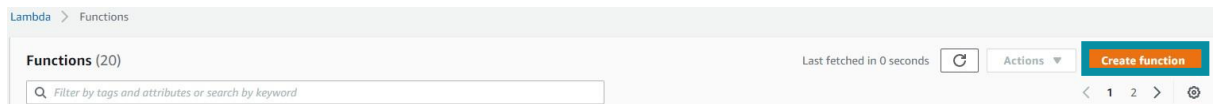
Dit document beschrijft hoe een lambda functie wordt toegevoegd aan het project. Deze functie zorgt ervoor dat als de temperatuur boven een bepaalde waarde komt, de LED aangaat en de LED topic wordt gepublished.

2 LAMBDA FUNCTIE TOEVOEGEN AAN AWS

- Open AWS en login indien nodig
- Navigeer naar "Compute"
- Klik op "Lambda"



- Klik op de knop "Create function"



- Voer de volgende gegevens in :
 - [1] Hoe wens je de functie aan te maken : dit staat standaard op "Author from scratch"
 - [2] Functie naam : Geef de functie een naam
 - [3] Runtime : in welke taal moet de functie geschreven worden. Hier kies je voor "Python 3.8"
- Klik op de knop "Create function" [4]

Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

Author from scratch Info
 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.

Browse serverless app repository
 Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name Info
 Enter a name that describes the purpose of your function.
 r0804148_Cedric_ControlLED
 Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
 Choose the runtime to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
 Python 3.8

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

► Advanced settings

4 **Create function**

- Je functie werd aangemaakt. Je gaat de code die je schreef in python om de LED aan te sturen en die je op Github plaatste kopiëren en in de Lambda-functie plakken
 - Open je Github via volgende url :
<https://github.com/cedric-carels/Cloud-and-Security/tree/main/Assignment%205/lambda>
 - Open het bestand : Control_LED_from_temperature.py
 - Kopieer de code
 - Activeer de AWS website
 - Activeer het tabblad "Code" [1]
 - Klap de folder open waarin je de functie aanmaakte (r0804148_Cédric_ControlLED)
 - Dubbel klik op de functie "lambda_function.py" [2]
 - Plak de gekopieerde code in de lambda-functie [3]

Lambda > Functions > r0804148_Cedric_ControlLED

r0804148_Cedric_ControlLED Throttle Copy ARN Actions

▼ Function overview Info

r0804148_Cedric_ControlLED
 Layers (0)

Description
 -
 Last modified
 24 seconds ago
 Function ARN
 arn:aws:lambda:us-west-2:227558477290:function:r0804148_Cedric_ControlLED

+ Add trigger + Add destination

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

Code source Info Upload from

File Edit Find View Go Tools Window Test Deploy Changes deployed

Go to Anything (Ctrl-P)

Environment

- r0804148_Cedric_C
 - lambda_function.py

Execution results x lambda_function x

```

1 import json
2
3 def lambda_handler(event, context):
4     # TODO Implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')}
8
9
  
```

3

- Opgelet, in de code moet de `led_topic = "<YOUR_LED_TOPIC>"` aanpassen naar de benaming van de `led_topic` die je meegaf in taak 1 in Arduino IDE. Je vindt deze benaming onder Github :
 - Open je Github via volgende url :
<https://github.com/cedric-carels/Cloud-and-Security/tree/main/Assignment%201/ESP32-MQTT>
 Naam van bestand : ESP32-MQTT.ino

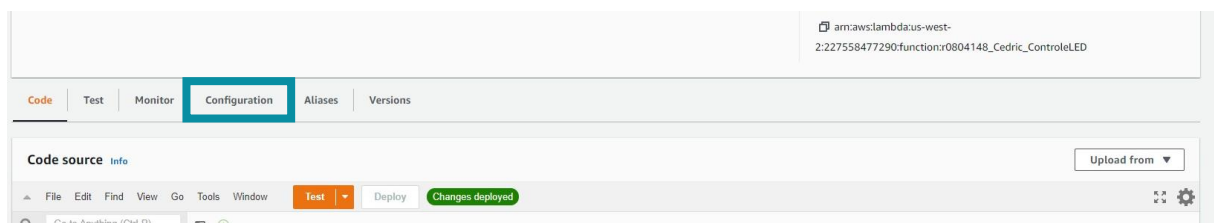
```
#define MQTT_SERIAL_RECEIVER_CH "r0804148/led"
```

```
1 import json
2 import boto3
3
4 led_topic = "<YOUR_LED_TOPIC>" → "r0804148/led"
5 temp = 22.00
6
7 client = boto3.client('iot-data', region_name='eu-west-1')
```

- Klik op de knop “Deploy” om de code te bevestigen

```
1 import json
2 import boto3
3
4 led_topic = "r0804148/led"
5 temp = 22.00
6
7 client = boto3.client('iot-data', region_name='eu-west-1')
8
9 def lambda_handler(event, context):
10     if float(event["temperature"]) > temp:
11         client.publish(topic=led_topic, qos=1, payload="1")
12     else:
13         client.publish(topic=led_topic, qos=1, payload="0")
14     return {
15         'statusCode': 200,
16         'body': json.dumps('control LED')
17     }
```

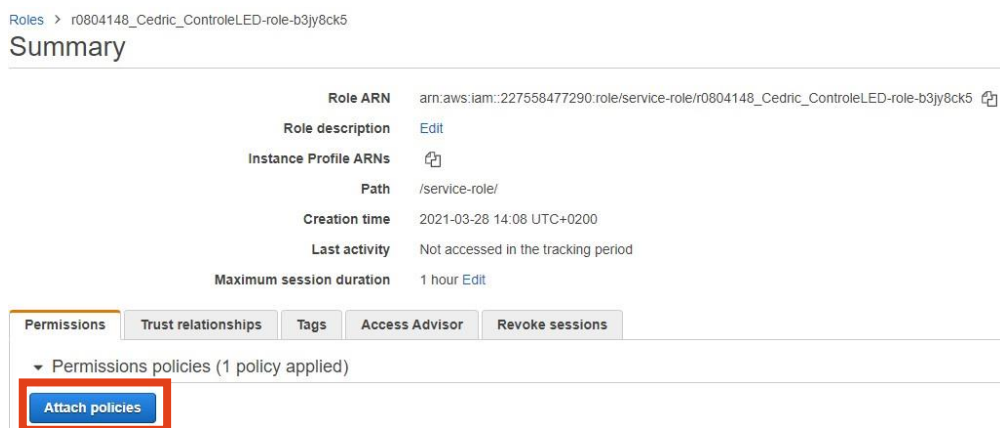
- Activeer het tabblad “Configuration”



- Kies in het linkervenster de “Permissions”
- Klik op de link onder “Role name”



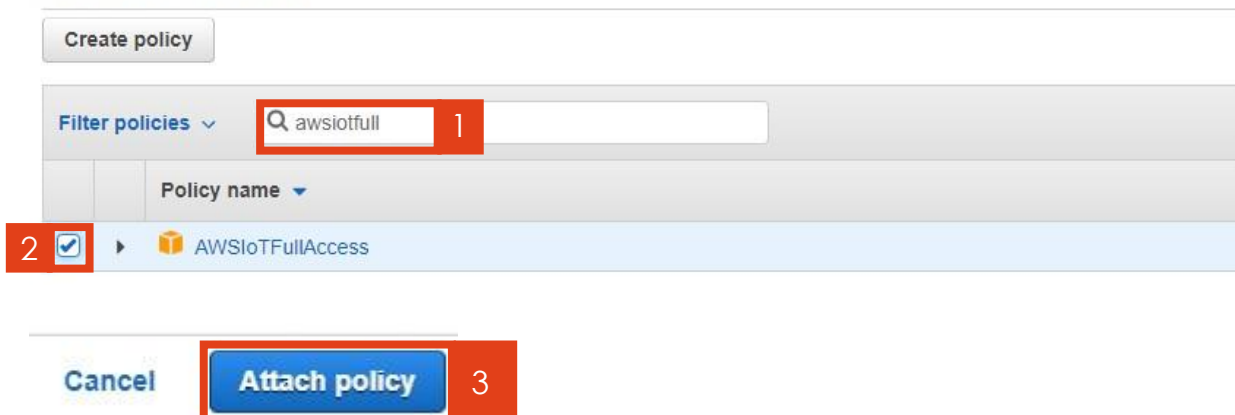
- De link wordt geopend. Klik op de knop “Attach policies”



- Klik in het zoekveld naast “Filter policies” en zoek “awsiotfull” [1]
- Vink het vakje voor “AWSIoTFullAccess aan [2]
- Klik op de knop “Attach policy” [3]

Add permissions to r0804148_Cedric_ControlLED-role-b3jy8ck5

Attach Permissions



- De nieuwe policy is aangemaakt

Policy name	Policy type	
AWSIoTFullAccess	AWS managed policy	✕
AWSLambdaBasicExecutionRole-b8943a70-2aba-4c50-b416-b2cc52da3eb0	Managed policy	✕

- Klik links boven aan in het scherm op “AWS” om terug te gaan naar de home-pagina
- We gaan een nieuwe regel aanmaken :
 - Ga naar “AWS IoT” door hierop te klikken [1]
 - Ga naar “Act” door hierop te klikken [1]
 - Klik op “Rules” [1]
 - Klik op “Add action” [2]

The screenshot shows the AWS IoT console interface. On the left, the navigation menu is visible with 'AWS IoT' and 'Act' highlighted. The main panel displays the configuration for a rule named 'r0804148_Cedric_rule'. The rule is currently 'ENABLED'. The configuration includes a description, a rule query statement 'SELECT * FROM 'iot/topic'', and two actions: 'Store a message in an Amazon S3 bucket' and 'Send a message to a downstream HTTPS endpoint'. The 'Add action' button is highlighted with a red box and the number 2.

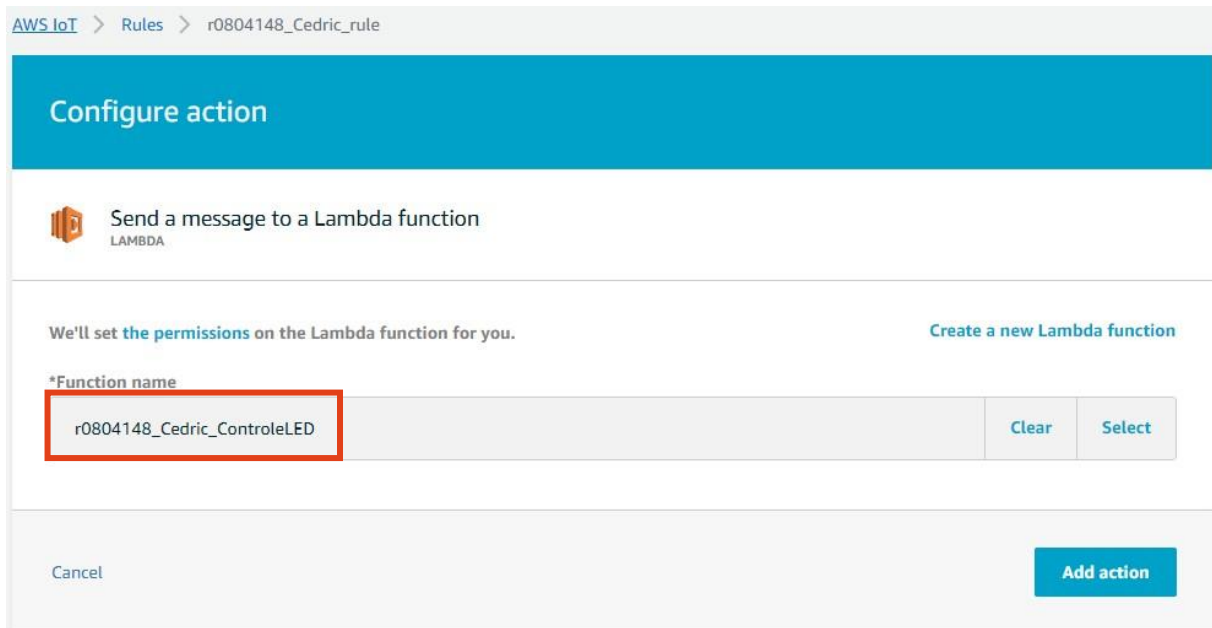
- Kies “Send a message to a Lambda function” uit de keuzelijst door het bolletje aan te klikken

The screenshot shows a single action card titled 'Send a message to a Lambda function'. It includes a blue circle icon with a white dot inside, indicating it is the selected action. The text reads 'Send a message to a Lambda function' and 'LAMBDA'.

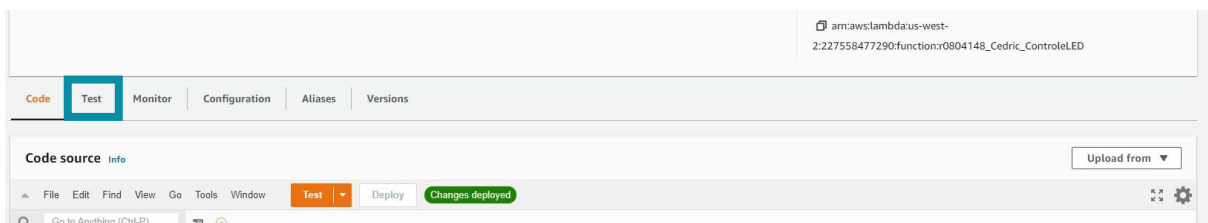
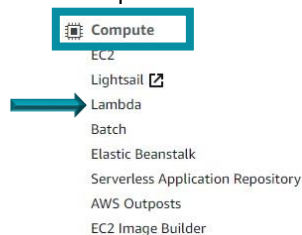
- Klik op de knop “Configure action”



- Vul de functienaam in die je gaf aan de functie



- We gaan de functie nu testen. Dit doe je als volgt :
 - Klik links boven aan in het scherm op “AWS” om terug te gaan naar de home-pagina
 - Navigeer naar “Compute”
 - Klik op “Lambda”
- Kies het tabblad “Test”



- Vul de gevraagde gegevens in :
 - [1] Event : New event = staat standaard ingesteld
 - [2] Template : staat standaard ingevuld
 - [3] Name : kies een naam die je wenst te geven aan je test
 - [4] Code : pas de code aan (zie code onder punt 4)

Test event

Invoke your function with a test event

- ☒ New event
- ☐ Saved event

1

Template

hello-world

2

Name

Mytest

3

```
1 {  
2   "device_id": "value1",  
3   "temperature": "25",  
4   "humidity": "60"  
5 }
```

4

- Klik op de knop "Save changes"

Save changes

- Klik op de knop "Invoke"

Invoke