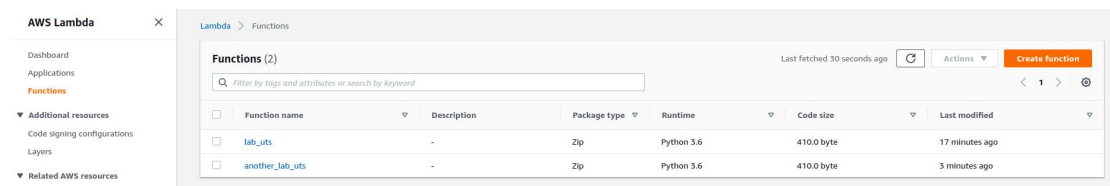


Serverless Code AWS Lambda

Pada berkas ini, akan dipaparkan langkah-langkah pengerjaan beserta screenshotnya dari setiap langkah dalam pembuatan serverless code.

1. Buka aws console, dan masuk ke layanan Lambda
2. Ketika sudah di layanan lambda, click function di sebelah kiri, lalu click create function.



3. Di dalam halaman create function, masukan function name (bebas namanya), bahasa yang digunakan (saya menggunakan python 3.6), arsitektur (default saja, x86_64), lalu gunakan execution role yang sudah ada (robomaker_students), lalu tekan tombol create function.

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

A function with that name already exists.
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

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.

☒ x86_64
☐ arm64

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to /

Change default execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role,

☐ Create a new role with basic Lambda permissions
☒ Use an existing role
☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. Th

Athallah Annafis
1706075022

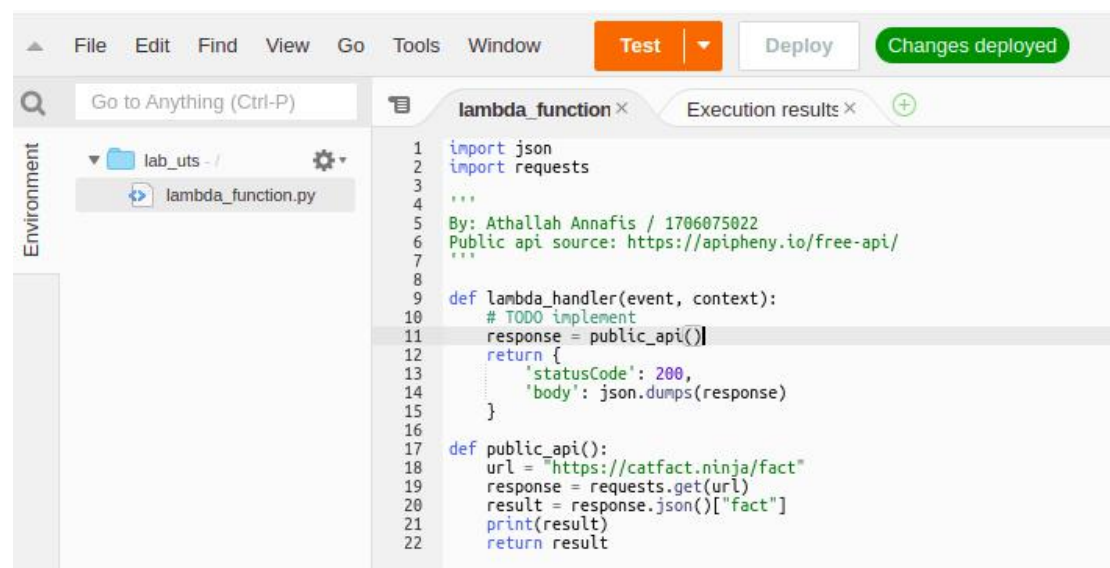
- Setelah itu, masuk ke function yang sudah dibuat, lalu edit scriptnya. Untuk memenuhi spesifikasi tugas, diperlukan library untuk pemanggilan api. Dalam python terdapat library “requests” untuk pemanggilan api. Namun perlu diperhatikan, module “requests” tidak terdaftar secara default pada AWS lambda, oleh karena itu harus diimport dengan cara tersendiri. Berikut script dari function yang saya buat.

```
import json
import requests

'''
By: Athallah Annafis / 1706075022
Public api source: https://apiipheny.io/free-api/
'''

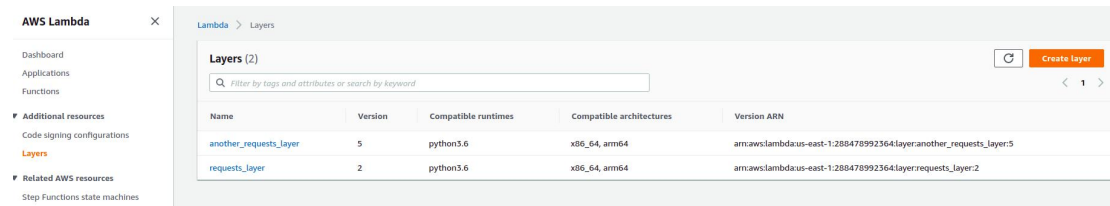
def lambda_handler(event, context):
    # TODO implement
    response = public_api()
    return {
        'statusCode': 200,
        'body': json.dumps(response)
    }

def public_api():
    url = "https://catfact.ninja/fact"
    response = requests.get(url)
    result = response.json()["fact"]
    print(result)
    return result
```



Athallah Annafis
1706075022

5. Agar module python “requests” dapat berjalan, harus dibuat layer untuk dipakai ke functionnya. Caranya kembalike dashboard AWS lambda, lalu click layers, click create.



6. Pada halaman create layer, diperlukan file .zip untuk import modulnya. File ini bisa dibuat pada lokal komputer dengan cara menjalankan beberapa command berikut (ubuntu). Selengkapnya di [sini](#):

- Jalankan “python3 -m venv env” pada suatu folder kosong.
- Lalu jalankan “source env/bin/activate” untuk memasuki venv nya.
- Setelah itu, jalankan “mkdir python” dan “cd python”
- Lalu, jalankan “pip3 install requests -r .” untuk mendapatkan module requests.
- Setelah selesai, lakukan “cd ..” dan compress folder python dengan cara “zip -r requests.zip python” untuk mendapatkan file .zip yang diperlukan.
- Lalu, unggah “requests.zip” ke dalam layer yang ingin kita buat, nama layer bebas, deskripsi opsional. Untuk arsitektur centang keduanya saja untuk menjaga-jaga. Untuk runtime **sesuaikan dengan versi python komputer saat membuat requests.zip** (jalankan python3 --version) dan lambda function yang telah dibuat, yaitu python 3.6. Lalu tekan create.

Name

Description - optional

☒ Upload a .zip file
☐ Upload a file from Amazon S3

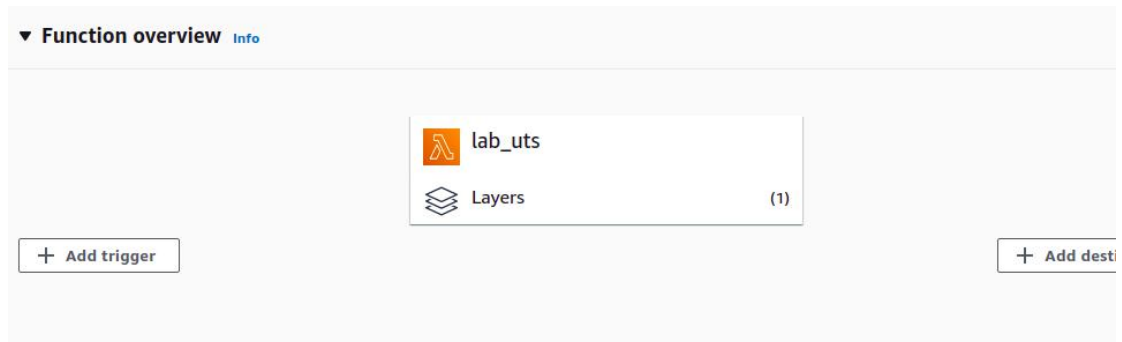
requests.zip (794.9 kB)
For files larger than 10 MB, consider uploading using /

Compatible architectures - optional [Info](#)
Choose the compatible instruction set architectures fo

☒ x86_64
☒ arm64

Compatible runtimes - optional [Info](#)
Choose up to 15 runtimes.

- g) Perlu diperhatikan, isi dari “request.zip” harus berstruktur “python/[requests_modules]”, saya sudah coba selain struktur seperti ini, module “requests” tidak bisa digunakan.
7. Kembali ke lambda function yang telah dibuat, pakai layer yang sudah dibuat dengan cara klik “layers” pada bagian ini.



Lalu klik add layer

Layers <small>Info</small>						Edit	Add a layer
Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN		
1	requests_layer	2	python3.6	x86_64, arm64	arn:aws:lambda:us-east-1:288478992564:layer:requests_layer:2		

8. Pada halaman add layer, pilih **custom layers** dan pilih layer yang telah dibuat dari langkah sebelumnya, dengan versi yang tersedia. Lalu klik create layer.

Choose a layer

Layer source Info

Choose from layers with a compatible runtime and instruction set architecture or specify the Amazon Resource Name (ARN) of a layer version. You can also [create a new layer](#).

☐ **AWS layers**
Choose a layer from a list of layers provided by AWS.

☒ **Custom layers**
Choose a layer from a list of layers created by your AWS account or organization.

☐ **Specify an ARN**
Specify a layer by providing the ARN.

Custom layers

Layers created by your AWS account or organization that are compatible with your function's runtime.

requests_layer ▼

Version

2 ▼

9. Selamat, module requests bisa digunakan! Berikut output dari program yang saya buat. Untuk testing, karena script saya tidak memerlukan parameter apapun, maka diisi kosong saja.

Athallah Annafis
1706075022

Test Event Name uts_event	
Response { "statusCode": 200, "body": "\nIn 1987, cats overtook dogs as the number one pet in America (about 50 million cats resided in 24 million homes in 1986). About 37% of American homes today have at least one cat.\n"	
Function Logs START RequestId: 36fda679-4530-45a6-95ed-e61acde82866 Version: \$LATEST In 1987, cats overtook dogs as the number one pet in America (about 50 million cats resided in 24 million homes in 1986). About 37% of American homes today have at least one cat. END RequestId: 36fda679-4530-45a6-95ed-e61acde82866 REPORT RequestId: 36fda679-4530-45a6-95ed-e61acde82866 Duration: 271.03 ms Billed Duration: 272 ms Memory Size: 128 MB Max Memory Used: 41 MB Init Duration: 116.62 ms	
Request ID 36fda679-4530-45a6-95ed-e61acde82866	

Demikian laporan terkait lab uts tentang serverless yang memanggil api publik. Sekian dan terima kasih. Api yang saya pakai adalah mengembalikan string mengenai fakta kucing secara random, dengan struktur seperti di bawah ini.

```
{  
  'fact': 'There are approximately 100 breeds of cat.',  
  'length': 42  
}
```

Untuk output, saya hanya mengambil “fact” nya saja.

Sekian dan terima kasih.