Setup Lambda Using Zappa

Mobile App Programming

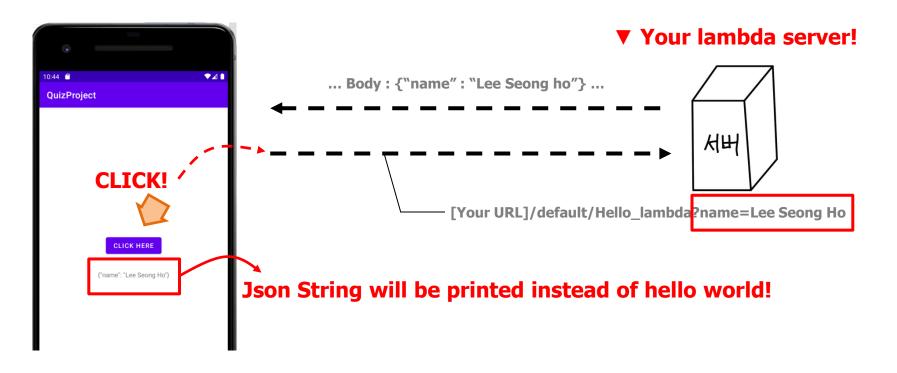
What we learn today?

- · Let's upload Flask web server to Lambda.
 - Implement a simple web server using python Flask Framework.
 - Simply upload web server to AWS Lambda
- Use the previous HTTP example application.

Week 10 Review

Function as a Service

- Lambda? Popular Serverless & FaaS service
 - Provide handler (function) as a service!
 - Can configure with web console



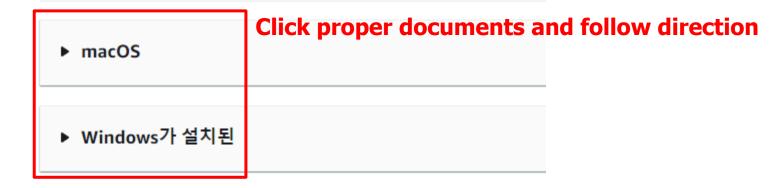
Python 3.x

- For MacOS, https://www.python.org/downloads/macos/
- https://www.python.org/downloads/windows/
 - In my environment, python 3.7.9 is used.
 - Select proper version and download Windows installer

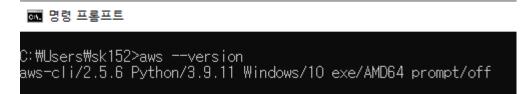


- Install AWS CLI
 - Visit below site and follow direction:

https://docs.aws.amazon.com/ko_kr/cli/latest/userguide/getting-started-install.html



Verify AWS CLI is properly installed:



Type `aws --version` on terminal Path or version can be different.

- Configure AWS CLI with IAM Access keys
 - Open terminal and type followings:



Then you can get API key and Secret Key with .csv



https://velog.io/@gnyiii/Python-Zappa%EB%A5%BC-%EC%9D%B4%EC%9A%A9%ED%95%9C-AWS-Lambda%EC%97%90-flask-Webapp-%EC%84%9C%EB%B2%84%EB%A6%AC%EC%8A%A4-%EA%B5%AC%ED%98%84%ED%95%98%EA%B8%B0

Install pip (python package index)

If it doesn't work...

\$ pip -version

https://bootstrap.pypa.io/get-pip.py --> Download pip installer

\$ python [path]/[to]/[get-pip.py]/get-pip.py

In Mac (or Linux)..

\$ sudo apt update && sudo apt install python3-pip

* https://docs.python.org/ko/3/tutorial/venv.html

Create python virtualenv

And.. Zappa needs activated virtualenv!

In Mac (or Linux)..

\$ sudo pip install virtualenv

\$ virtualenv [가상환경이름]

\$ source [가상환경이름]/bin/activate

In Windows...

\$ python -m venv [가상환경이름] \$.\[가상환경이름]\Scripts\activate.bat

Install required modules

\$ pip install flask\$ pip install zappa

https://velog.io/@gnyiii/Python-Zappa%EB%A5%BC-%EC%9D%B4%EC%9A%A9%ED%95%9C-AWS-Lambda%EC%97%90-flask-Webapp-%EC%84%9C%EB%B2%84%EB%A6%AC%EC%8A%A4-%EA%B5%AC%ED%98%84%ED%95%98%EA%B8%B0

Install required modules

```
$ pip install flask$ pip install zappa
```

In Windows, however, error can be occurred while installing kappa.

```
Collecting kappa==0.6.0 from zappa->-r requirements.txt (line 3))
Using carbod https://f.les.pythonhosted.org/packages/ee/fa/lb8328d2199520ef5a257f8a2e9315ed0b0194e35
3a152ca1959490dfbc8/kappa-0.6.0.tar.gz
Complete output from command python setup.py egg_info:
Traceback (most recent call last):
File "cstring>", line 1, in <module>
File "C:\Users\", '::.AppData\Local\Temp\pip-install-itqxz66w\kappa\setup.py", line 54, in <module>
run_setup()
File "C:\Users\", '::.AppData\Local\Temp\pip-install-itqxz66w\kappa\setup.py", line 22, in run_setup
UnicodeDecodeError: 'cp949' codec can't decode byte 0xe2 in position 2339: illegal multibyte seque
nce

Command "python setup.py egg_info" failed with error code 1 in C:\Users\", AppData\Local\Temp\pip-install-itqxz66w\kappa\
```

https://velog.io/@gnyiii/Python-Zappa%EB%A5%BC-%EC%9D%B4%EC%9A%A9%ED%95%9C-AWS-Lambda%EC%97%90-flask-Webapp-%EC%84%9C%EB%B2%84%EB%A6%AC%EC%8A%A4-%EA%B5%AC%ED%98%84%ED%95%98%EA%B8%B0

- Install required modules (Solution for zappa installation error)
 - 1. Download kappa source from: https://github.com/garnaat/kappa
 - **2. Replace setup.py with:** https://gist.github.com/shlee-627/37e3578e38a570b68964268860f90756
 - 3. Reinstall kappa and zappa

```
$ pip install -e .
$ pip install zappa
```

4. If below command is worked, it is properly installed

\$ zappa init

```
What do you want to call this environment (default 'dev'): dev
...
What do you want to call your bucket? (default 'zappa-*****): zappa-app
...
```

https://velog.io/@gnyiii/Python-Zappa%EB%A5%BC-%EC%9D%B4%EC%9A%A9%ED%95%9C-AWS-Lambda%EC%97%90-flask-Webapp-%EC%84%9C%EB%B2%84%EB%A6%AC%EC%8A%A4-%EA%B5%AC%ED%98%84%ED%95%98%EA%B8%B0

Copy simple server code from:

https://gist.github.com/shlee-627/d9f6aa82b99d097b31f2ea891f968652

Basic python flask server structure

PATH and METHOD that belows handler need to deal with.

```
@app.route("/adduser", methods=['POST'])
def update_name():
    content = request.get_json(silent=True)

    username = content["username"]
    age = content["age"]
```

METHOD also can be POST

```
from flask import Flask
from flask import request
from flask import jsonify
```

◀ import request, jsonify from flask

```
@app.route("/getuser", methods=['GET'])
def get name():
   name = request.args.get('name')
                                            Get QueryParameter with key "name"
   exist = False
                                            ret is python dictionary!
    ret = {}
    for usr in users:
        if usr[0] == name:
           ret["username"] = usr[0]
           ret["age"] = usr[1]
                                         OR... return with JSON and status code
           exist = True
                                         return jsonify (ret), 200
           break
   # return isonifv(maze=string decode(
    return jsonify (ret)
```

Return JSON string → Send HTTP response with this JSON string body

```
{"age":30, "username": "test"}
```

```
@app.route("/adduser", methods=['POST'])
def update name():
    content = request.get_json(silent=True)
    username = content["username"]
    age = content["age"]
    exist = False
    for usr in users:
        if usr[0] == username:
            exist = True
    if exist is False:
        users.append ([username, age])
        return jsonify(success=True)
    else:
        return jsonify(success=False)
```

Get JSON request body

<class 'dict'>

Automatically converts JSON string to dict

Also return something here...

→ Client will receive JSON string

{"success":true}

```
if __name__ == "__main__":
    app.run(host='localhost', port=8888)
Client can connect to this server
with localhost:8888
```

Test your web server

Execute your server first

```
$ python ./simple_server.py
```

Send POST request first to add user data

► https://reqbin.com/req/c-dwjszac0/curl-post-json-example

```
Curl Raw (6) EXT $

curl -X POST localhost:8888/adduser  
-H 'Content-Type: application/json'  
-d '{"username":"test", "age":30}'
```

Type below lines in your browser (Send GET request)

http://localhost:8888/getuser?name=test

- Zappa? A tool that allows you to easily deploy Flask-based Web Applications to AWS Lambda
- 1. Activate python virtualenv (refer page 6)
- 2. Create a new folder and copy your web server code inside.

ONLY server related source should exist inside the folder.

3. Initialize zappa

\$ zappa init

4. Deploy your server

\$ zappa deploy dev

Your environment name

\$ zappa update dev

► After modifying source code...

```
Your Zappa configuration can support multiple production stages, like 'dev', 'staging', and 'production'.
What do you want to call this environment (default 'dev'): dev
AWS Lambda and API Gateway are only available in certain regions. Let's check to make sure you have a profile set up in one that will work.
Okay, using profile default!
Your Zappa deployments will need to be uploaded to a private S3 bucket.
If you don't have a bucket yet, we'll create one for you too.
What do you want to call your bucket? (default 'zappa-y5xcpy03e'):
It looks like this is a Flask application.
What's the modular path to your app's function?
This will likely be something like 'your module.app'.
                                                                         Zappa automatically finds your Flask Web Server
We discovered: simple server.app
Where is your app's function? (default 'simple_server.app'):
You can optionally deploy to all available regions in order to provide fast global service.
If you are using Zappa for the first time, you probably don't want to do this!
Would you like to deploy this application ←[1mglobally←[0m? (default 'n') [y/n/(p)rimary]: n
Okay, here's your zappa_settings.json:
                                          zappa_settings.json will be created
          app function": "simple_server.app",
         runtime": "python3.7",
's3_bucket": "zappa-y5xcpy03e
Does this look ←[32m←[1mokay←[0m? (default 'y') [y/n]: y
 zappa-ex) C:#Users#hsherlcok#Desktop#simple_server>zappa deploy dev
Calling deploy for stage dev..
 Creating simple-server-dev-ZappaLambdaExecutionRole IAM Hole..
 Preating zappa-permissions policy on simple-server-dev-ZappaLambdaExecutionRole IAM Role.
Downloading and installing dependencies...
  - markupsafe==2.1.1: Using locally cached manylinux wheel
Packaging project as zip.
Uploading simple-server-dev-1651229942.zip (6.1MiB)
100%1
                                                                                                                                                                           ■| 6.35M/6.35M [00:00<00:00, 9.43MB/s]
Waiting for lambda function [simple-server-dev] to become active...
Scheduling..
-Scheduled simple-server-dev-zappa-keep-warm-handler.keep_warm_callback with expression rate(4 minutes)!
Uploading simple-server-dev-template-1651229976.json (1.6KiB)
                                                                                                                                                                            | 1.65k/1.65k [00:00<00:00, 36.3kB/s]
Waiting for stack simple-server-dev to create (this can take a bit)
100%
                                                                                                                                                                                 4/4 [00:15<00:00, 3.87s/res]
Deploying API Gateway..
Waiting for lambda fu<mark>rction [simple-server-dev] to be updated...</mark>
Deployment complete!: https://
```

Zappa will return lambda URL

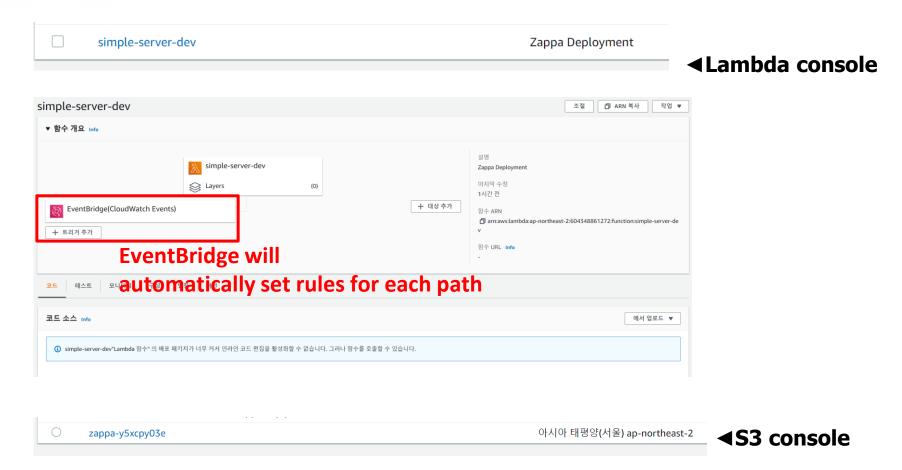
► Send POST request example (w/ curl)

```
2 amazonaws com/dev/adduser
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 17
Connection: keep-alive
Date: Fri, 29 Apr 2022 11:50:29 GMT
x-amzn-Requestld: 580da7d3-d162-47d7-8e44-a861d3a26fdd
x-amzn-Remapped-Content-Length: 17
x-amz-apigw-id: RV2Y4GeXoEOFY8g=
X-Amzn-Trace-Id: Root=1-626bd105-1c57cdcb2c30a4b052dc6ea4:Sampled=0
X-Cache: Miss from cloudfront
Via: 1.1 62a062f12ca71f7fcf29dd4f665321e0.cloudfront.net (CloudFront)
X-Amz-Cf-Pop: ICN55-C1
X-Amz-Cf-Id: z9LBRh-ary1zcRgtpJojd4LpyXXEn-6T9SZHk5TEMsPswk710oKnSg==
{"success":true}
```

Response from server

▶ Send GET request example (w/ browser)

In AWS Lambda...



- This week we will make a SIMPLE Monte-Carlo Simulation API
- Your server can handle following requests with following path and parameter:
- GET request
 - /monte-carlo/pi
 - "n" : [number of iterations]
 - When server receives a request, your server runs monte-carlo pi-estimation with [number of iterations] times. The result of monte-carlo estimation should be stored in server.
 - After [number of iterations] iterations, your server sends a response with following status code 200 and JSON body:
 - {"pi": [estimated pi value], "elapse time": [elapsed time to execute monte-carlo]}
- GET request
 - /get/pi
 - When server receives a request, your server returns stored pi value with JSON body.
 - {"pi": [stored pi value]}

- POST request
 - /update/pi
 - Your client will send POST request with following JSON string:
 - {"pi" : [the value of pi]}
 - When request is received, your server stored received data and returns success=True.

jsonify<mark>(</mark>success=True)

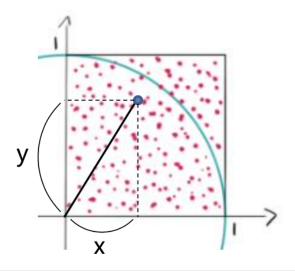
HINTs

- Elapse Time: https://www.programiz.com/python-programming/examples/elapsed-time
- Random number generation :
 https://stackoverflow.com/questions/33359740/random-number-between-0-and-1-in-python
- sqrt in python :

import math math.sqrt (value)

In week 6...

- REMIND: What's Pi-estimation using Monte-Carlo Simulation
 - Estimate Pi using Monte Carlo Simulation
 - Monte Carlo simulations are used to model the probability of different outcomes in a process that cannot easily be predicted due to the intervention of random variables.
 - How to estimate pi?



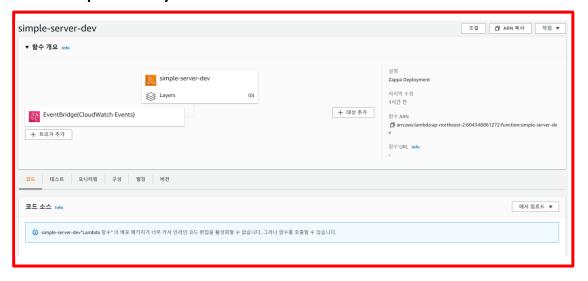
- 1) Get two random float values range $0\sim1$ (x, y)
- 2) If (x,y) is inside of circle, $\sqrt{x^2 + y^2} \le 1$
- 3) If (x,y) is outside of circle, $\sqrt{x^2 + y^2} > 1$
- 4) Run (1)-(3) many times
- 5) Calculate (number of dots inside) / (total dots)

GET request (monte-carlo/pi) Example



POST request (update/pi) Example

- Copy your lambda URL in notepad and store as studentid_url.txt
- Change your [flask_server_file_name].py into studentid.py
- Captures your lambda console's "Function Description" (함수 개요):



We <u>WILL NOT</u> care your lambda handler name.

Submit both text file, <student_id>.py and captured images in ICAMPUS with zip file named <student_id>.zip