

1 Lab. Lambda, API Gateway
2
3 [실습목표]
4 1. Lambda를 생성하고 테스트 할 수 있다.
5 2. Amazon API Gateway를 통해 Lambda를 실행 할 수 있다.
6 3. 브라우저에서 API Gateway를 호출할 수 있다.
7
8 -Refer to 'Lambda 실습 구성도1.png'
9
10 [실습순서]
11 1. Lambda 생성
12 2. API Gateway 생성 - Lambda 연결
13 3. API Gateway 배포
14
15
16
17 1. Lambda 생성
18 -Lambda 기능 : 파라미터로 인사(greet) 및 대상(name)을 받아 리턴하고 전달될 파라미터가 없을 경우 'Hello World'를 리턴
19 1)Lambda > 함수 > 함수 생성
20 2)[새로 작성] 선택
21 3)함수이름 : {계정이름}-demo
22 4)런타임 : python 3.9 선택
23 5)[함수 생성] 클릭
24
25 6)[코드] 소스 에디터에서 다음 코드 편집
26 import json
27
28 print ('Lamba Start!')
29
30 def lambda_handler(event, context):
31 print('Received Event : ' + json.dumps(event, indent=2))
32 greet = 'Hello'
33 name = 'World'
34
35 if 'greet' in event:
36 greet = event['greet']
37
38 if 'name' in event:
39 name = event['name']
40
41 greetings = greet + ' ' + name + '!'
42 print(greetings)
43 return greetings
44
45 7)[Deploy] 버튼 클릭
46 8)테스트 수행1 (빈 값 테스트)
47 -[테스트] 탭을 클릭
48 -[이벤트 작업 테스트] > 새 이벤트 생성
49 -[이벤트 이름] : {계정이름}-event
50 -[이벤트 JSON]에서 Json에 값을 지우고 [저장] > [테스트] 버튼 클릭
51 -테스트 실행 후 상단에 나타나는 [실행결과: 성공 (로그)의 [세부 정보]를 클릭하여 테스트 결과 확인
52 --전달된 파라미터가 없으므로 'Hello World' 출력
53
54 9)테스트 수행2 (파라미터 전달 테스트)
55 -[이벤트 JSON]에 다음의 테스트 코드 입력
56 {
57 "greet" : "Hi",
58 "name" : "Henry"
59 }
60 -[저장] > [테스트] 버튼 클릭
61 -테스트 실행 후 상단에 나타나는 [실행결과: 성공 (로그)의 [세부 정보]를 클릭하여 테스트 결과 확인
62 "Hi Henry!"
63
64
65
66 2. API Gateway 생성
67 1)[서비스] > [네트워킹 및 콘텐츠 전송] > [API Gateway] > API
68 2)[API 생성] 버튼 클릭
69 3)[API 유형 선택] > [RESP API] > [구축] 버튼 클릭
70 4)[프로토콜 선택] > [REST]
71 5)[새 API 생성] > [새 API]
72 6)[설정] > [API 이름] : {계정이름}-greeting-api
73 7)[API 생성] 버튼 클릭
74 8)[작업] > [리소스 생성] 메뉴 클릭
75 9)[새 하위 리소스] 에서
76 -[리소스 이름] : {계정이름}-greeting
77 -[리소스 경로] : {계정이름}-greeting
78 -[리소스 생성] 버튼 클릭
79 10)[리소스] 영역에 위에서 만든 리소스 선택
80 -[작업] > [메소드 생성] 메뉴 클릭
81 -[메소드]- 'GET' 선택 후 체크 버튼 클릭
82 -우측 [GET 설정] 영역의 [Lambda 함수]에 위 1번에서 생성한 Lambda의
ARN(예:arn:aws:lambda:ap-northeast-2:503237308475:function:dgb-00-demo)을 복사하여 붙임
83 -[저장] 버튼 클릭

```

84 -[Lambda 함수 권한 추가] 팝업차에서 [확인] 클릭
85 11)위에서 만든 GET 메소드 선택
86 -[통합 요청] 클릭
87 -[매핑 템플릿]을 클릭하여 확장
88 -[요청 본문 패스스루] > [정의된 템플릿이 없는 경우] 선택
89 -[Content-Type] > [매핑 템플릿 추가] 클릭 후 'application/json' 입력
90 -체크 버튼 클릭
91 -[템플릿 생성] > [메서드 요청 패스스루] 선택
92 -[템플릿]에 아래 코드 입력
93     *전달된 파라미터를 json으로 변경해주는 템플릿
94     #set($name = $input.params('name'))
95     #set($greet = $input.params('greet'))
96
97     { "code" : "1"
98
99     #if($name != "")
100     , "name" : "$name"
101     #end
102
103     #if($greet != "")
104     , "greet" : "$greet"
105     #end
106
107     }
108
109 -[저장] 버튼 클릭
110

```

```

111 12)좌측의 메서드 이름 'GET' 클릭
112 -[테스트] 클릭하여 [GET - 메서드 테스트] 화면으로 이동
113 -쿼리 문자열에 값을 입력 - greet=값&name=값 형태로 입력
114 -[쿼리 문자열] : greet=Hola&name=Henry
115 -[테스트] 버튼 클릭
116 -우측의 테스트 결과 확인 > [쿼리 문자열]에 입력된 파라미터대로 응답이 오는지 확인
117 --[응답 본문] : "Hola Henry!"
118 -상단의 [메서드 실행]을 클릭하여 이전화면 이동
119

```

```

120 13)[응답 본문]을 JSON형태로 받을 수 있도록 수정하기 위해 [통합 응답] 클릭
121 -[GET - 통합 응답] 페이지에서 목록의 화살표를 클릭하여 확장
122 -[매핑 템플릿] 클릭하여 확장 > [매핑 템플릿 추가] 클릭
123 -[Content-Type] : application/json 입력
124 -체크 버튼 클릭
125 -[템플릿 생성] > [메서드 요청 패스스루] 선택
126 -템플릿 입력 :
127     {
128     "greeting" : "$input.path('$')"
129     }
130 -[저장] 버튼 클릭
131

```

```

132 14)상단의 [메서드 실행]을 클릭하여 이전 화면으로 이동
133 -[테스트] 클릭하여 테스트 화면으로 이동
134 -[쿼리 문자열] 입력 -> greet=값&name=값 형태로 입력
135 --예:greet=Hola&name=Henry
136 -[테스트] 버튼을 클릭
137 -우측의 테스트 결과 확인 > [쿼리 문자열]에 입력된 파라미터대로 [응답 본문]에 json 형태의 응답이 오는지 확인
138
139
140

```

3. API Gateway 배포

```

142 1)[API Gateway] > [API] > 위 2번에서 생성한 API 클릭
143 2)2번에서 생성한 리소스 선택
144 3)[작업] > [API 배포] 메뉴 선택
145 4)[API 배포] 팝업에서
146 -[배포 스테이지] : [새 스테이지] 선택
147 -[스테이지 이름] : {계정이름}-stage
148 -[배포] 버튼 클릭
149 -생성된 url([URL 호출 값]과 위 2번에서 생성한 [리소스] 이름 및 파라미터를 브라우저에서 호출하여 API Gateway에 접속 한 후 정상적으로 메시지가
리턴되는지 확인한다.
150     ex)
151     --api 배포시 생성되는 url : https://fdghf778th.execute-api.ap-northeast-2.amazonaws.com/dgb-00-stage
152     --배포 리소스 명 : dgb-00-greeting
153     --전달 파라미터 : ?greet=Hi&name=Henry
154     --Browser URL :
155     https://fdghf778th.execute-api.ap-northeast-2.amazonaws.com/dgb-00-stage/dgb-00-greeting?greet=Hi&name=Henry
156     --결과
157     {
158     "greeting" : "Hi Henry!"
159     }

```