

## Lab2. Using AWS Lambda

### 목적

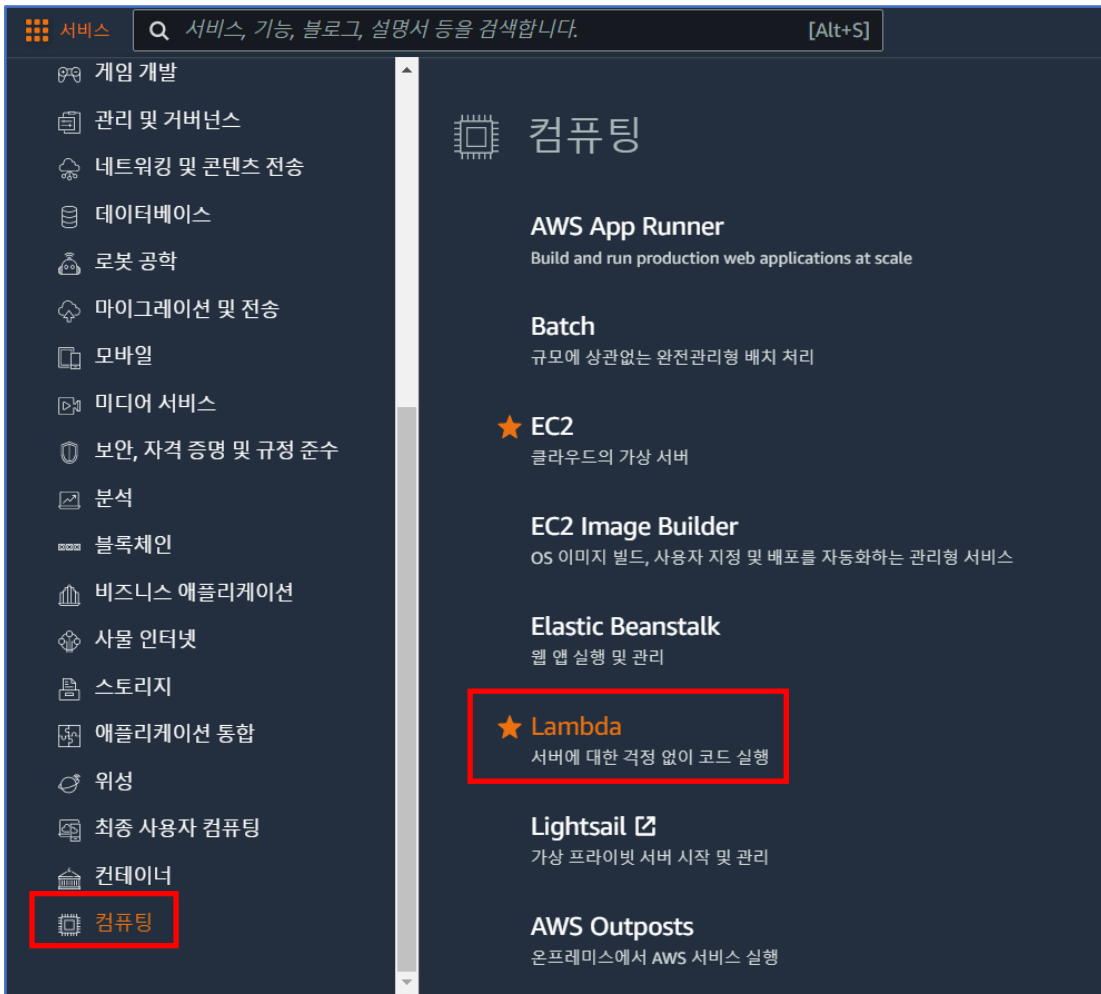
이번 실습에서는 애플리케이션을 배포하기 위해 별도의 EC2 인스턴스의 프로비저닝 없이 개발할 수 있는 서버리스의 대표적인 플랫폼인 Lambda에 대해 다룬다.

### 사전 준비물

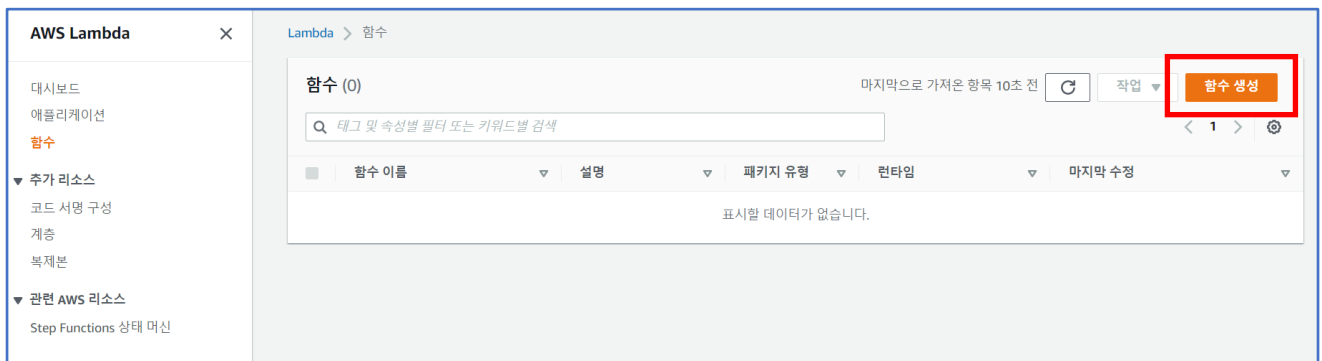
AWS Free-Tier 계정

# 간단한 Lambda 실행하기

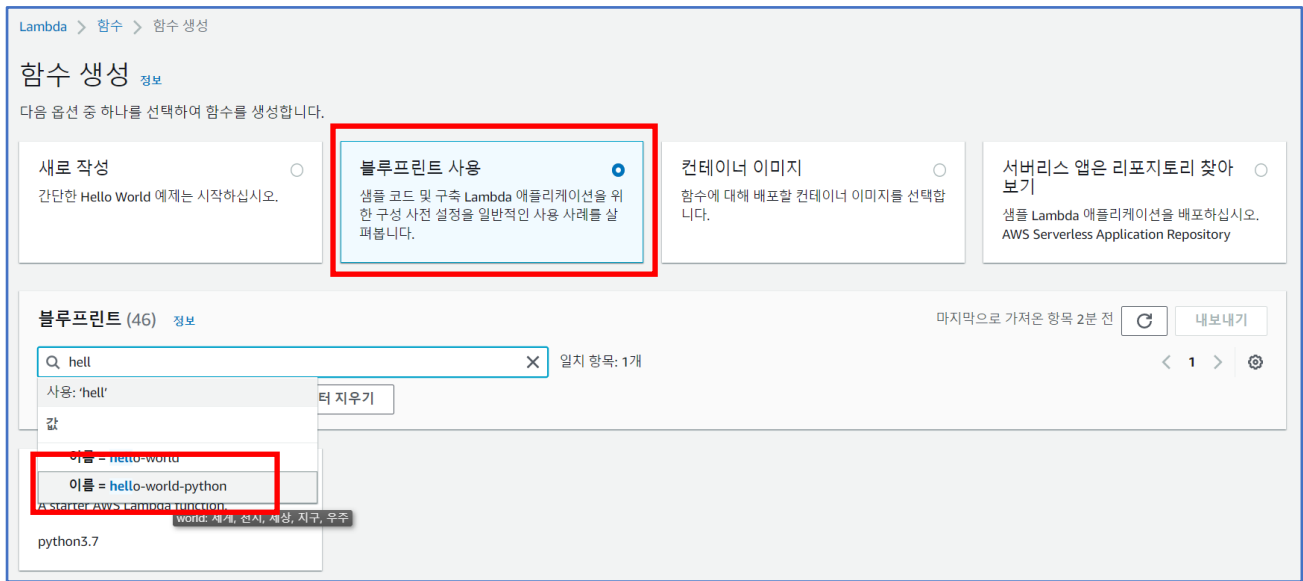
1. [서비스] > [컴퓨팅] > [Lambda] 메뉴를 클릭한다.



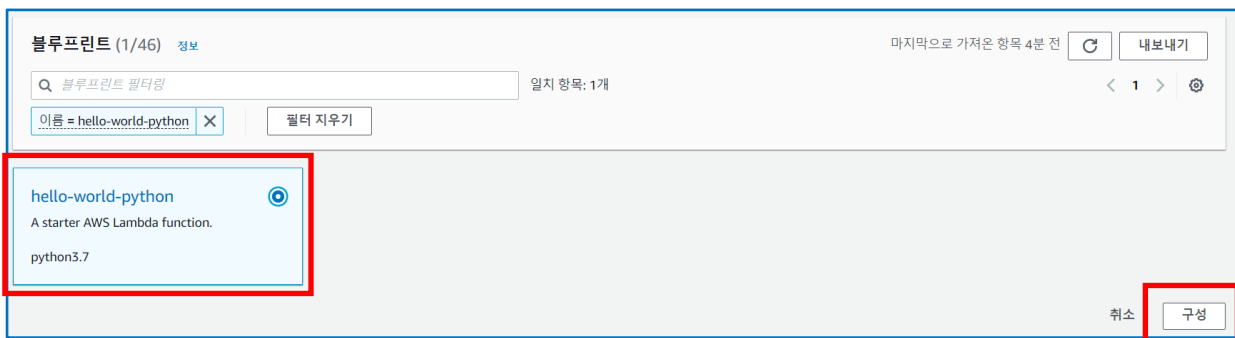
2. [AWS Lambda] 페이지로 들어왔다. 우측상단의 [함수 생성]을 클릭한다.



3. 이번 실습은 직접 코딩하면서 난이도 있는 작업을 하는 것이 아닌, 간단한 **Lambda** 실습을 위해서 **[함수 생성]**에서 **[블루프린트 사용]**을 선택한다. 그리고 그 아래 **[블루프린트]** 섹션에서 이름을 **hello-world-python**을 선택한다.



4. 선택한 **hello-world-python** 카드를 선택하고 **[구성]** 버튼을 클릭한다.



5. **[기본 정보]** 섹션에 **[함수 이름]**을 간단하게 **helloworld0529**로 입력한다. 이 코드는 간단한 실습이기 때문에 권한에 대한 설정 없이 진행하기 위해 **[실행 역할]**은 기본 **Lambda** 권한을 가진 **새 역할 생성**을 선택한다.

Lambda > 함수 > 함수 생성 > 블루프린트 hello-world-python 구성

### 기본 정보 정보

함수 이름

실행 역할

함수에 대한 권한을 정의하는 역할을 선택합니다. 사용자 지정 역할을 생성하려면 [IAM 콘솔](#)로 이동합니다.

☒ 기본 Lambda 권한을 가진 새 역할 생성

☐ 기존 역할 사용

☐ AWS 정책 템플릿에서 새 역할 생성

❗ 역할을 생성하는 데 몇 분 정도 걸릴 수 있습니다. 역할을 삭제하거나 이 역할에서 신뢰 또는 권한 정책을 편집하지 마십시오.

Lambda가 이름이 helloworld0530-role-jg99ztoi이고 Amazon CloudWatch Logs에 로그를 업로드할 수 있는 권한이 포함된 실행 역할을 생성합니다.

6. **[Lambda 함수 코드]** 섹션은 미리 만들어진 코드를 보여준다. **[함수 생성]**을 클릭한다.

### Lambda 함수 코드

코드는 선택한 블루프린트에 의해 사전 구성되어 있습니다. 함수를 생성한 후 코드를 구성할 수 있습니다. Lambda 함수 배포에 대해 [자세히 알아보세요](#).

❗ 이 함수에 외부 라이브러리가 포함됩니다.

런타임  
Python 3.7

```
1 import json
2
3 print('Loading function')
4
5
6 def lambda_handler(event, context):
7     #print("Received event: " + json.dumps(event, indent=2))
8     print("value1 = " + event['key1'])
9     print("value2 = " + event['key2'])
10    print("value3 = " + event['key3'])
11    return event['key1'] # Echo back the first key value
12    #raise Exception('Something went wrong')
13
```

취소

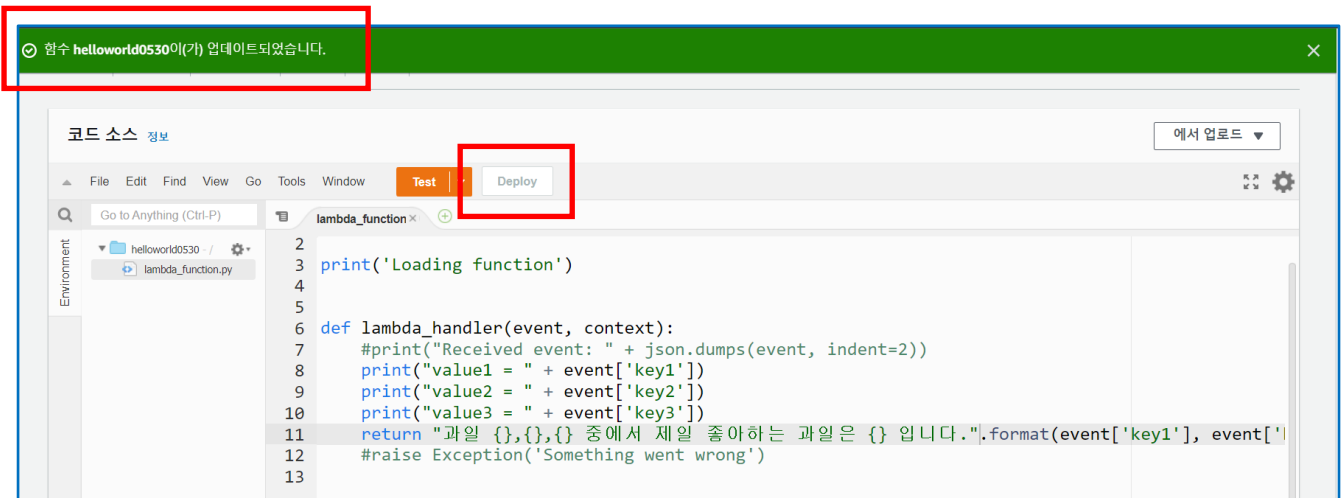
**함수 생성**

7. Lambda 함수 helloworld0530이 잘 생성되었다. 아래 [코드] 탭으로 이동한다.



8. [코드] 탭에서 실행할 코드를 다시 한번 확인한다. 블루프린트에서 확인했던 처음 코드를 다음과 같이 수정했다. 그 후, [Deploy] 버튼을 클릭하여 수정한 코스를 반영한다. 그러면 함수 helloworld0530이(가) 업데이트되었습니다라는 메시지를 초록색바에서 확인할 수 있다.

```
return "과일 {},{},{} 중에서 제일 좋아하는 과일은 {} 입니다.".format(event['key1'], event['key2'], event['key3'], event['key3'])
```



9. 테스트를 위한 소스를 조금 수정하려고 **[테스트]** 탭을 클릭한다. 그리고 다음과 같이 소스를 수정했다.

템플릿 - 선택 사항

hello-world

이벤트 JSON

1 ▾ {

2   "key1": "Orange",

3   "key2": "Apple",

4   "key3": "Melon"

5 }

10. 코드를 수정한 후 페이지를 위로 스크롤업하여 **[이벤트 이름]**에 **helloworldtest**라고 입력한 후 **[저장]** 버튼을 누른다.

테스트 이벤트

저장 테스트

이벤트를 저장하지 않고 함수를 호출하려면 JSON 이벤트를 구성한 다음 테스트를 선택합니다.

이벤트 작업 테스트

● 새 이벤트 생성 ● 저장된 이벤트 편집

이벤트 이름

helloworldtest

문자, 숫자, 점, 하이픈 및 밑줄을 사용하여 최대 25자로 구성합니다.

이벤트 공유 설정

● 프라이빗

이 이벤트는 Lambda 콘솔 및 이벤트 생성자만 사용할 수 있습니다. 총 10개를 구성할 수 있습니다. 자세히 알아보기

○ 공유 가능

이 이벤트는 공유 가능한 이벤트에 액세스하고 이를 사용할 수 있는 권한이 있는 동일한 계정 내 IAM 사용자가 사용할 수 있습니다. 자세히 알아보기

템플릿 - 선택 사항

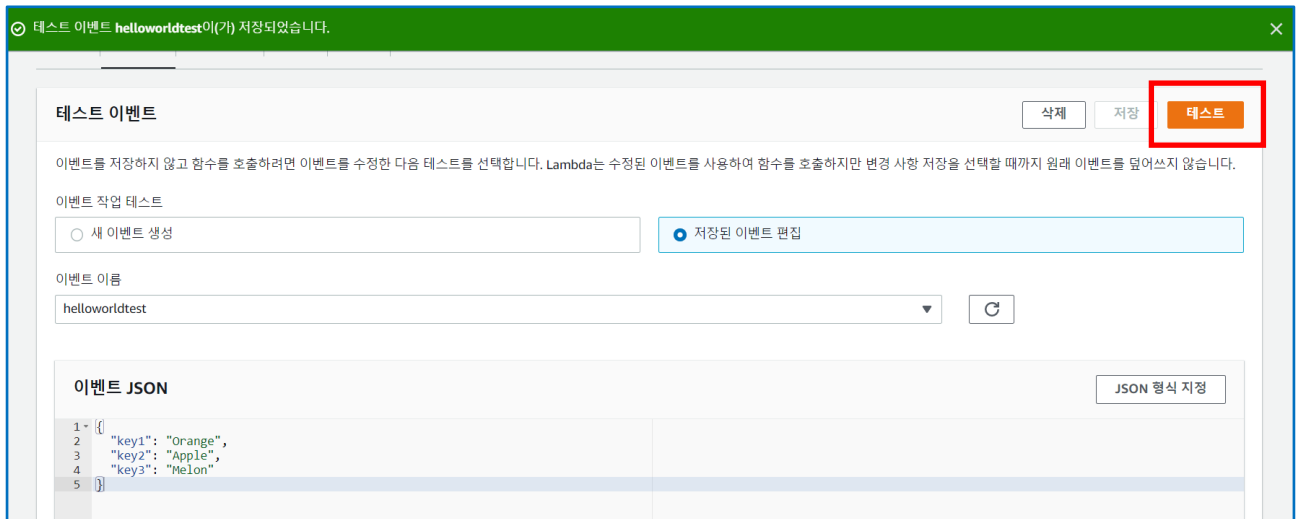
hello-world

이벤트 JSON

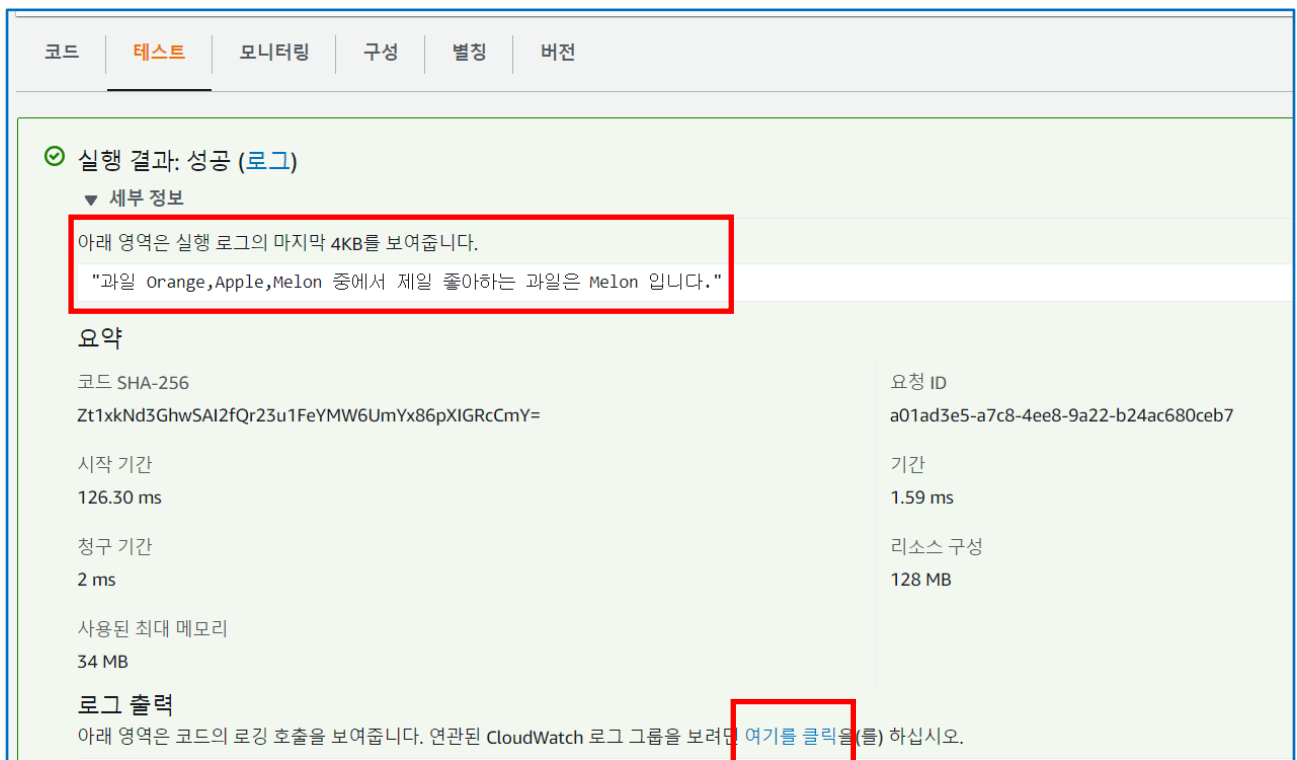
JSON 형식 지정

1 ▾ {

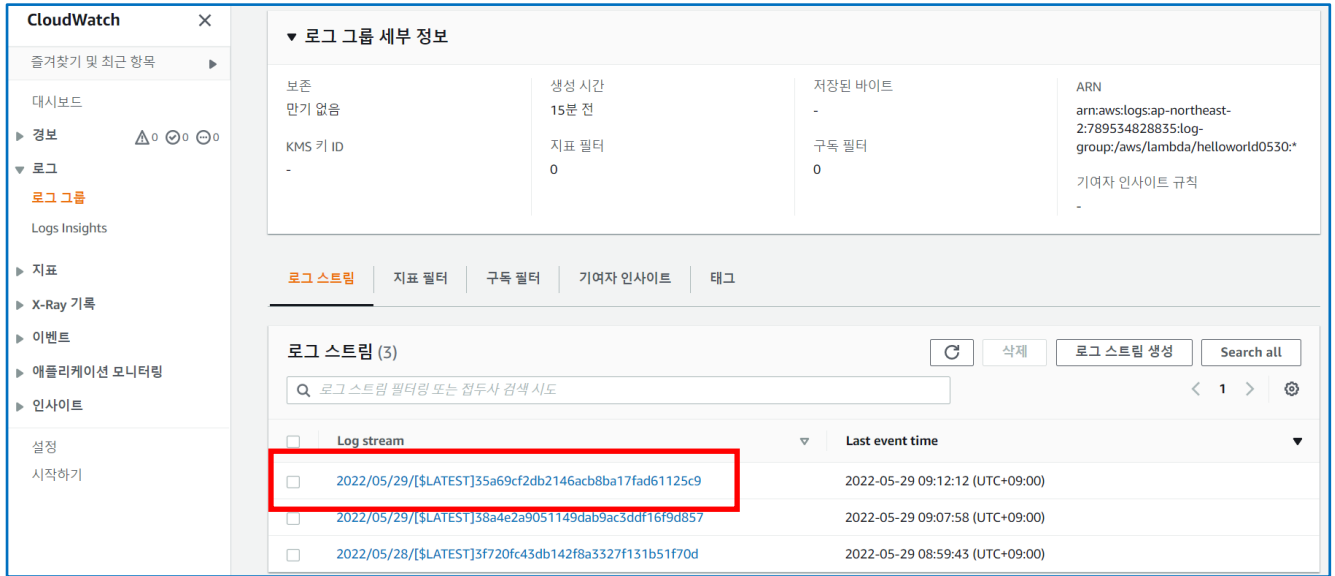
11. 그리고 [테스트] 버튼을 클릭한다.



12. 테스트가 성공적으로 진행되었고 실행 결과에 보면 성공이라고 되어 있으며, [세부 정보]를 클릭하면 간단한 정보를 확인할 수 있다. [로그 출력] 부분에 보면 연관된 CloudWatch 로그 그룹을 보려면 여기를 클릭을(를) 하십시오.에서 링크를 클릭해본다.



13. Lambda는 로그에 대해 자동으로 작업을 진행하여 **CloudWatch** 페이지로 이동하는 것을 볼 수 있다. [로그 스트림] 섹션에서 가장 마지막 로그를 클릭해본다.



CloudWatch

▼ 로그 그룹 세부 정보

보존 만기 없음	생성 시간 15분 전	저장된 바이트 -	ARN arn:aws:logs:ap-northeast-2:789534828835:log-group:/aws/lambda/helloworld0530:*
KMS 키 ID -	지표 필터 0	구독 필터 0	기여자 인사이트 규칙 -

로그 스트림 | 지표 필터 | 구독 필터 | 기여자 인사이트 | 태그

로그 스트림 (3)

로그 스트림 필터링 또는 접두사 검색 시도

Log stream	Last event time
2022/05/29/[\$LATEST]35a69cf2db2146acb8ba17fad61125c9	2022-05-29 09:12:12 (UTC+09:00)
2022/05/29/[\$LATEST]38a4e2a9051149dab9ac3ddf16f9d857	2022-05-29 09:07:58 (UTC+09:00)
2022/05/28/[\$LATEST]3f720fc43db142f8a3327f131b51f70d	2022-05-29 08:59:43 (UTC+09:00)

14. 해당 로그에 기록된 내용을 자세히 확인할 수 있다.



CloudWatch > Log groups > /aws/lambda/helloworld0530 > 2022/05/29/[\$LATEST]35a69cf2db2146acb8ba17fad61125c9

로그 이벤트

아래의 필터 막대를 사용하여 로그 이벤트의 용어, 구분 또는 값을 검색하고 매칭할 수 있습니다. 필터 패턴에 대해 자세히 알아보기

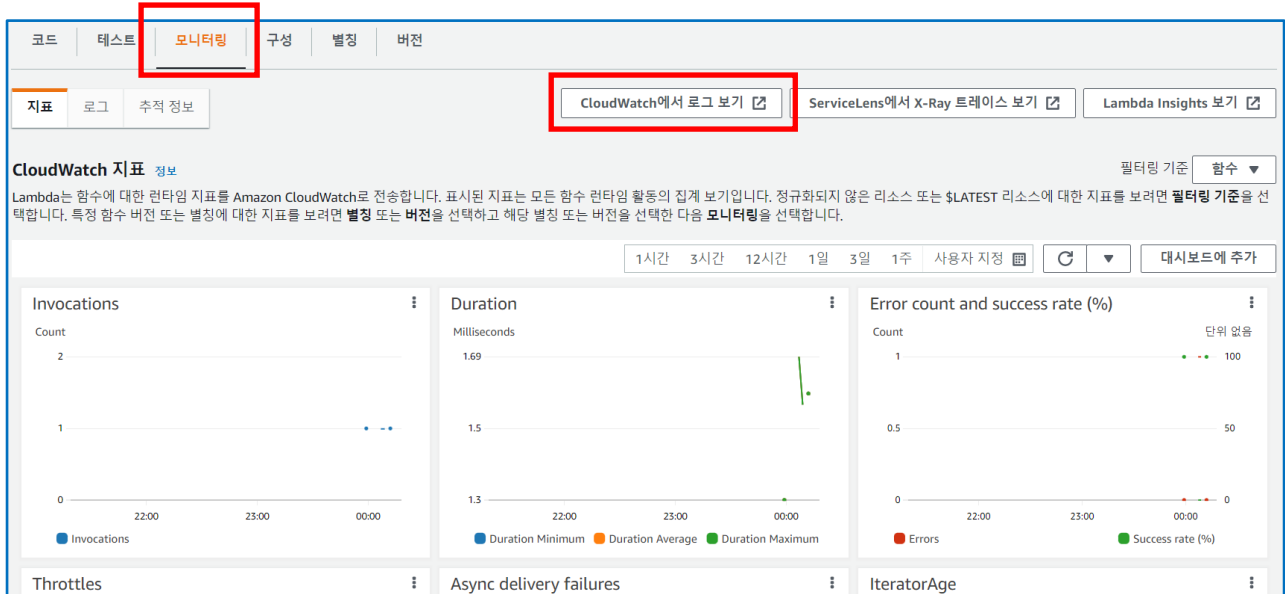
텍스트로 보기 | 작업 | Create Metric Filter

이벤트 필터링

타임스탬프	메시지
	현재 이전 이벤트가 없습니다. <a href="#">재시도</a>
2022-05-29T09:12:12.156+09:00	START RequestId: a01ad3e5-a7c8-4ee8-9a22-b24ac680ceb7 Version: \$LATEST
2022-05-29T09:12:12.156+09:00	Loading function
2022-05-29T09:12:12.158+09:00	value1 = Orange
2022-05-29T09:12:12.158+09:00	value2 = Apple
2022-05-29T09:12:12.158+09:00	value3 = Melon
2022-05-29T09:12:12.174+09:00	END RequestId: a01ad3e5-a7c8-4ee8-9a22-b24ac680ceb7
2022-05-29T09:12:12.174+09:00	REPORT RequestId: a01ad3e5-a7c8-4ee8-9a22-b24ac680ceb7 Duration: 1.59 ms Billed Duration: 2 ms Memory Size: 128 MB Max Mem...
	현재 최신 이벤트가 없습니다. <a href="#">자동 재시도를 일시 중지했습니다.</a>



15. 또한 [모니터링] 탭을 보면 보다 자세한 모니터링 정보를 확인할 수 있고, 여기에서도 [CloudWatch에서 로그 보기]를 클릭하면 위와 동일하게 CloudWatch 페이지로 이동하게 된다.



# S3를 이벤트 트리거로 지정하여 Lambda Function Test 하기

1. 다음과 같이 새롭게 코딩하였다. 아직 변경사항을 Deploy하지 않았기 때문에 **[Deploy]** 버튼을 클릭한다.

```
import json

import urllib.parse

import boto3

s3 = boto3.resource('s3')

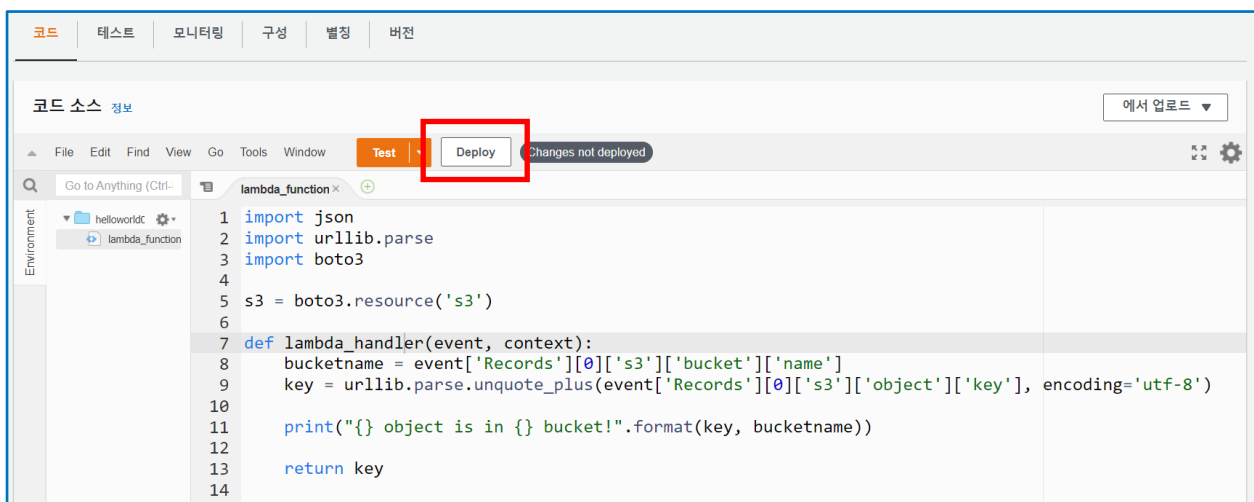
def lambda_handler(event, context):

    bucketname = event['Records'][0]['s3']['bucket']['name']

    key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')

    print("{} object is in {} bucket!".format(key, bucketname))

    return key
```



2. 새 이벤트를 위해 [새 이벤트 생성]을 클릭한다.

### 테스트 이벤트

이벤트를 저장하지 않고 함수를 호출하려면 JSON 이벤트를 구성한 다음 테스트를 선택합니다.

이벤트 작업 테스트

☒ 새 이벤트 생성

☐ 저장된 이벤트 편집

이벤트 이름

MyEventName

문자, 숫자, 점, 하이픈 및 밑줄을 사용하여 최대 25자로 구성합니다.

이벤트 공유 설정

☒ 프라이빗

이 이벤트는 Lambda 콘솔 및 이벤트 생성자만 사용할 수 있습니다. 총 10개를 구성할 수 있습니다. [자세히 알아보기](#)

☐ 공유 가능

이 이벤트는 공유 가능한 이벤트에 액세스하고 이를 사용할 수 있는 권한이 있는 동일한 계정 내 IAM 사용자가 사용할 수 있습니다. [자세히 알아보기](#)

3. [템플릿]에서 S3 Put을 선택한다.

### 템플릿 - 선택 사항

helloworldtest

Q s3|

**AWS**

Rekognition **S3** Request

**S3** Delete

**S3** Put

4. 템플릿을 변경하면 그에 맞는 새로운 템플릿 코드가 확인된다.

템플릿 - 선택 사항

s3-put

이벤트 JSON JSON 형식 지정

```
1 {
2   "Records": [
3     {
4       "eventVersion": "2.0",
5       "eventSource": "aws:s3",
6       "awsRegion": "us-east-1",
7       "eventTime": "1970-01-01T00:00:00.000Z",
8       "eventName": "ObjectCreated:Put",
9       "userIdentity": {
10        "principalId": "EXAMPLE"
11      },
12      "requestParameters": {
13        "sourceIPAddress": "127.0.0.1"
14      },
15      "responseElements": {
16        "x-amz-request-id": "EXAMPLE123456789",
17        "x-amz-id-2": "EXAMPLE123/5678abcdefghijklmbdaisawesomemnopqrstuvwxyzABCDEFGH"
18      },
19      "s3": {
20        "s3SchemaVersion": "1.0",
21        "configurationId": "testconfigrule",
22        "bucket": {
23          "name": "example-bucket",
24          "ownerIdentity": {
25            "principalId": "EXAMPLE"
26          },
27          "arn": "arn:aws:s3:::example-bucket"
28        },
29        "object": {
30          "key": "test%2Fkey",
31          "size": 1024
```

5. 이벤트의 이름은 **putEvent**라고 이름을 넣는다.

테스트 이벤트

이벤트를 저장하지 않고 함수를 호출하려면 JSON 이벤트를 구성한 다음 테스트를 선택합니다.

이벤트 작업 테스트

☒ 새 이벤트 생성

이벤트 이름

putEvent

문자, 숫자, 점, 하이픈 및 밑줄을 사용하여 최대 25자로 구성합니다.

6. 아래와 같이 테스트를 위한 코드를 일부 수정한다. 여기서 지정된 **bucket**의 이름은 이벤트를 실행하기 전 생성되어 있어야 한다. Bucket의 이름과 간단한 이미지 이름을 즉 오브젝트 이름을 입력한다.

### 이벤트 JSON

```
3 {
4   "eventVersion": "2.0",
5   "eventSource": "aws:s3",
6   "awsRegion": "us-east-1",
7   "eventTime": "1970-01-01T00:00:00.000Z",
8   "eventName": "ObjectCreated:Put",
9   "userIdentity": {
10    "principalId": "EXAMPLE"
11  },
12  "requestParameters": {
13    "sourceIPAddress": "127.0.0.1"
14  },
15  "responseElements": {
16    "x-amz-request-id": "EXAMPLE123456789",
17    "x-amz-id-2": "EXAMPLE123/5678abcdefghijklambdaisawesome/mnopqrstuvwxyzABCDEFGH"
18  },
19  "s3": {
20    "s3SchemaVersion": "1.0",
21    "configurationId": "testConfigRule",
22    "bucket": {
23      "name": "henry0529-bucket",
24      "ownerIdentity": {
25        "principalId": "EXAMPLE"
26      },
27      "arn": "arn:aws:s3:::henry0529-bucket"
28    },
29    "object": {
30      "key": "Textbook.jpg",
31      "size": 1024,
32      "eTag": "0123456789abcdef0123456789abcdef",
33      "sequencer": "0A1B2C3D4E5F678901"
34    }
35  }
36 }
```

7. **Amazon S3**로 이동하여 새로운 **bucket**을 다음과 같이 생성한다. [버킷 이름]은 위 **Lambda** 코드와 같아야 한다.

Amazon S3 > 버킷 > 버킷 만들기

## 버킷 만들기 정보

버킷은 S3에 저장되는 데이터의 컨테이너입니다. [자세히 알아보기](#)

### 일반 구성

버킷 이름

henry0529-bucket

버킷 이름은 고유해야 하며 공백 또는 대문자를 포함할 수 없습니다. [버킷 이름 지정 규칙 참조](#)

AWS 리전

아시아 태평양(서울) ap-northeast-2

기존 버킷에서 설정 복사 - 선택 사항  
다음 구성의 버킷 설정만 복사됩니다.

버킷 선택

8. [모든 퍼블릭 액세스 차단]을 모두 해제한다.

☐ 모든 퍼블릭 액세스 차단  
이 설정을 활성화하면 아래 4개의 설정을 모두 활성화한 것과 같습니다. 다음 설정 각각은 서로 독립적입니다.

☐ 새 ACL(액세스 제어 목록)을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단  
S3은 새로 추가된 버킷 또는 객체에 적용되는 퍼블릭 액세스 권한을 차단하며, 기존 버킷 및 객체에 대한 새 퍼블릭 액세스 ACL 생성을 금지합니다. 이 설정은 ACL을 사용하여 S3 리소스에 대한 퍼블릭 액세스를 허용하는 기존 권한을 변경하지 않습니다.

☐ 임의의 ACL(액세스 제어 목록)을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단  
S3은 버킷 및 객체에 대한 퍼블릭 액세스를 부여하는 모든 ACL을 무시합니다.

☐ 새 퍼블릭 버킷 또는 액세스 지점 정책을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단  
S3은 버킷 및 객체에 대한 퍼블릭 액세스를 부여하는 새 버킷 및 액세스 지점 정책을 차단합니다. 이 설정은 S3 리소스에 대한 퍼블릭 액세스를 허용하는 기존 정책을 변경하지 않습니다.

☐ 임의의 퍼블릭 버킷 또는 액세스 지점 정책을 통해 부여된 버킷 및 객체에 대한 퍼블릭 및 교차 계정 액세스 차단  
S3은 버킷 및 객체에 대한 퍼블릭 액세스를 부여하는 정책을 사용하는 버킷 또는 액세스 지점에 대한 퍼블릭 및 교차 계정 액세스를 무시합니다.

 모든 퍼블릭 액세스 차단을 비활성화하면 이 버킷과 그 안에 포함된 객체가 퍼블릭 상태가 될 수 있습니다. 정적 웹 사이트 호스팅과 같은 구체적으로 확인된 사용 사례에서 퍼블릭 액세스가 필요한 경우가 아니면 모든 퍼블릭 액세스 차단을 활성화하는 것이 좋습니다.

☒ 현재 설정으로 인해 이 버킷과 그 안에 포함된 객체가 퍼블릭 상태가 될 수 있음을 알고 있습니다.

9. [태그] 설정도 하고 나머지는 기본값을 사용하기로 하고 [버킷 만들기]를 클릭한다.

태그 (1) - 선택 사항  
버킷에 태그를 지정하여 스토리지 비용 또는 기타 기준을 추적합니다. 자세히 알아보기

키

값 - 선택 사항

Name

henry0529-bucket

제거

태그 추가

기본 암호화

이 버킷에 저장된 새 객체를 자동으로 암호화합니다. 자세히 알아보기

서버 측 암호화

☒ 비활성화

☐ 활성화

고급 설정

버킷을 생성한 후 파일과 폴더를 해당 버킷에 업로드할 수 있고, 추가 버킷 설정도 구성할 수 있습니다.

취소

버킷 만들기


10. 해당 버킷의 정책을 다음과 같이 편집한다. **Bucket** 정책 편집은 7주차 Lab2. Create Amazon S3 buckets and Manage.pdf를 참조한다.

## 버킷 정책 편집 정보

### 버킷 정책

JSON으로 작성된 버킷 정책은 버킷에 저장된 객체에 대한 액세스 권한을 제공합니다. 버킷 정책은 다른 계정이

버킷 ARN

 `arn:aws:s3:::henry0529-bucket`

### 정책

```
1 {
2   "Id": "Policy1653788131336",
3   "Version": "2012-10-17",
4   "Statement": [
5     {
6       "Sid": "Stmt1653788113649",
7       "Action": [
8         "s3:GetObject"
9       ],
10      "Effect": "Allow",
11      "Resource": "arn:aws:s3:::henry0529-bucket/*",
12      "Principal": "*"
13    }
14  ]
15 }
```

11. 정책 편집을 저장하여 **Bucket**의 퍼블릭 액세스가 가능하도록 설정했다.

✔ 버킷 정책을 편집했습니다.

Amazon S3 > 버킷 > henry0529-bucket


## henry0529-bucket 정보

퍼블릭 액세스 가능

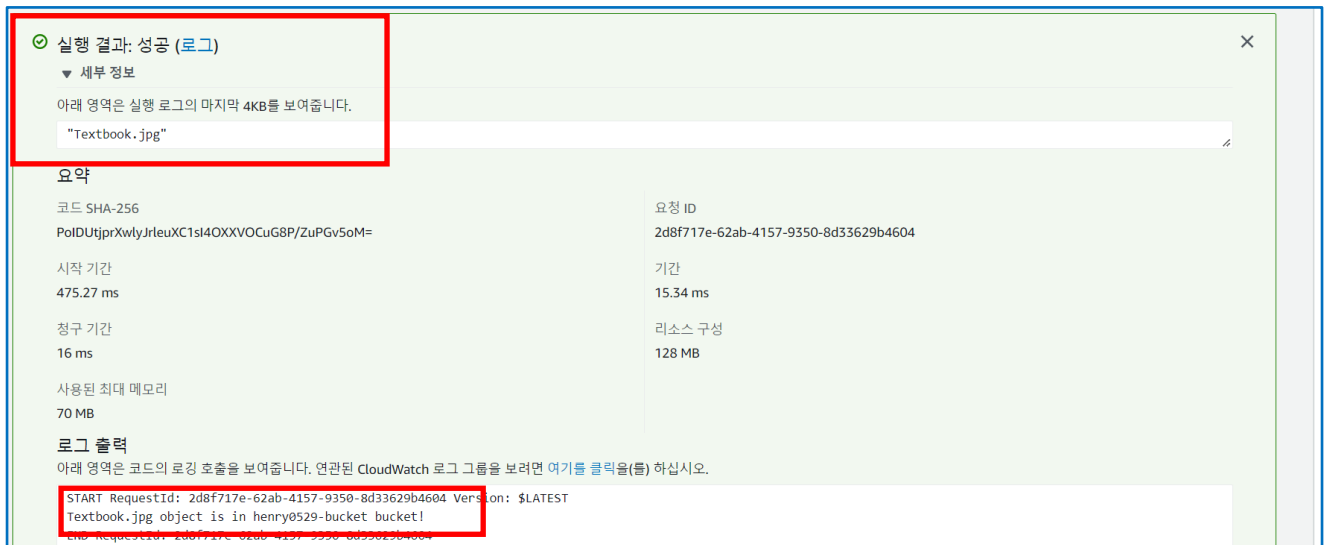
객체 | 속성 | **권한** | 지표 | 관리 | 액세스 지점

### 권한 개요

액세스

 퍼블릭

12. 다시 **Lambda Function**의 **[테스트]** 탭에서 **[저장]** 버튼을 클릭하여 저장한 후 **[테스트]** 버튼을 클릭하여 테스트한다. 다음과 같은 성공 실행 결과를 얻었다. **[세부 정보]**를 클릭한다. 아래 **[로그 출력]**에 보면 **Textbook.jpg object is in henry0529-bucket bucket!**이라는 문장을 확인할 수 있다.



실행 결과: 성공 (로그)

▼ 세부 정보

아래 영역은 실행 로그의 마지막 4KB를 보여줍니다.

"Textbook.jpg"

요약

코드 SHA-256	PoIDUtjprXwlyJrleuXC1sl4OXXVOCuG8P/ZuPGv5oM=	요청 ID	2d8f717e-62ab-4157-9350-8d33629b4604
시작 시간	475.27 ms	기간	15.34 ms
청구 시간	16 ms	리소스 구성	128 MB
사용된 최대 메모리	70 MB		

로그 출력

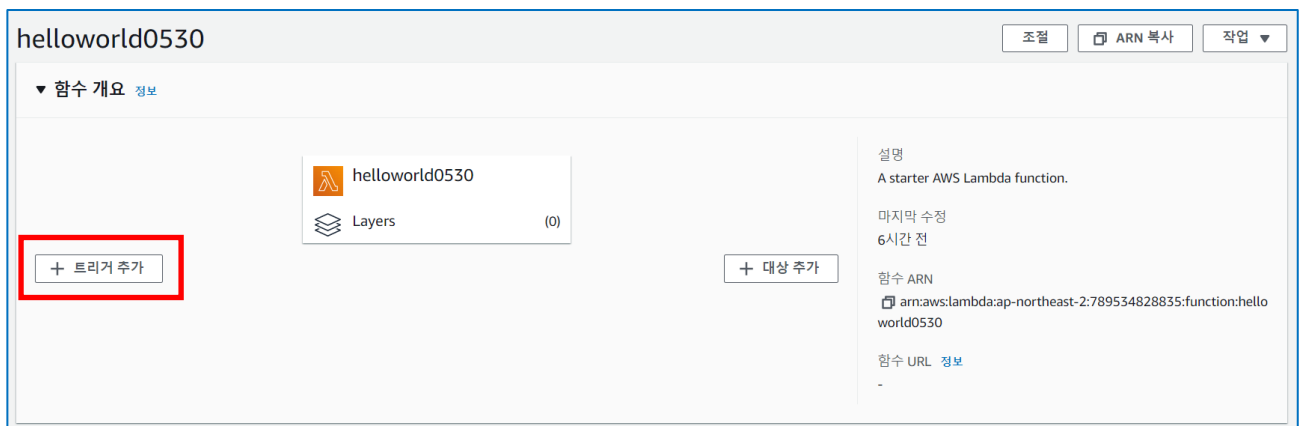
아래 영역은 코드의 로그 호출을 보여줍니다. 연관된 CloudWatch 로그 그룹을 보려면 [여기를 클릭](#)을(를) 하십시오.

START RequestId: 2d8f717e-62ab-4157-9350-8d33629b4604 Version: \$LATEST

Textbook.jpg object is in henry0529-bucket bucket!

END RequestId: 2d8f717e-62ab-4157-9350-8d33629b4604


13. 코드 테스트를 마쳤으니, 이제 실제로 S3에 객체를 올렸을 때 제대로 동작하는지 확인해 보자. **[+ 트리거 추가]**를 클릭한다.




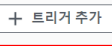
helloworld0530

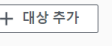
조절 ARN 복사 작업 ▼

▼ 함수 개요 정보

 helloworld0530

 Layers (0)

 + 트리거 추가

 + 대상 추가

설명

A starter AWS Lambda function.

마지막 수정

6시간 전

함수 ARN

arn:aws:lambda:ap-northeast-2:789534828835:function:helloworld0530

함수 URL [정보](#)

-




14. **[추가 트리거]** 페이지로 이동했다. **[트리거 구성]** 섹션에서 목록 중 **S3**를 선택하면 그 아래 화면이 변경된다. **[버킷]** 목록에서 위에서 생성한 bucket 이름을 선택하고 **[이벤트 유형]**은 **모든 객체 생성 이벤트** 그대로 사용하기로 한다.

Lambda > 추가 트리거

## 추가 트리거

### 트리거 구성

 **S3**  
aws storage

**버킷**  
이벤트 소스의 역할을 하는 S3 버킷을 선택하십시오. 버킷은 함수와 같은 리전에 있어야 합니다.

henry0529-bucket

**이벤트 유형**  
Lambda 함수를 트리거하려는 이벤트를 선택합니다. 필요에 따라 이벤트의 접두사 또는 접미사를 설정할 수 있습니다. 하지만 각 버킷에서 개별 이벤트는 접두사나 접미사가 겹쳐서 객체 키가 동일해질 수 있는 구성을 여러 개 가질 수 없습니다.

모든 객체 생성 이벤트

**접두사 - 선택 사항**  
필요할 경우, 일치하는 문자로 시작하는 키를 사용하여 객체에 대해 알림을 제한하려는 단일 접두사를 입력합니다.


예: images/

**접미사 - 선택 사항**  
필요할 경우, 일치하는 문자로 끝나는 키를 사용하여 객체에 대해 알림을 제한하려는 단일 접미사를 입력합니다.

예: .jpg

15. 나머지 값은 기본값 그대로 사용하기로 하고, **[재귀 호출]** 체크 박스에 체크하고 **[추가]**를 클릭한다.

Lambda는 Amazon S3이(가) 이 트리거에서 Lambda 함수를 호출하는 데 필요한 권한을 추가합니다. Lambda 권한 모델에 대해 [자세히 알아보기](#).

 **재귀 호출**  
함수가 S3 버킷에 객체를 쓰는 경우 입력 및 출력에 다른 S3 버킷을 사용하고 있는지 확인합니다. 동일한 버킷에 쓰면 재귀 호출이 생성될 위험이 증가하며 이는 Lambda 사용량 증가 및 비용 증가를 발생시킬 수 있습니다. [자세히 알아보기](#)

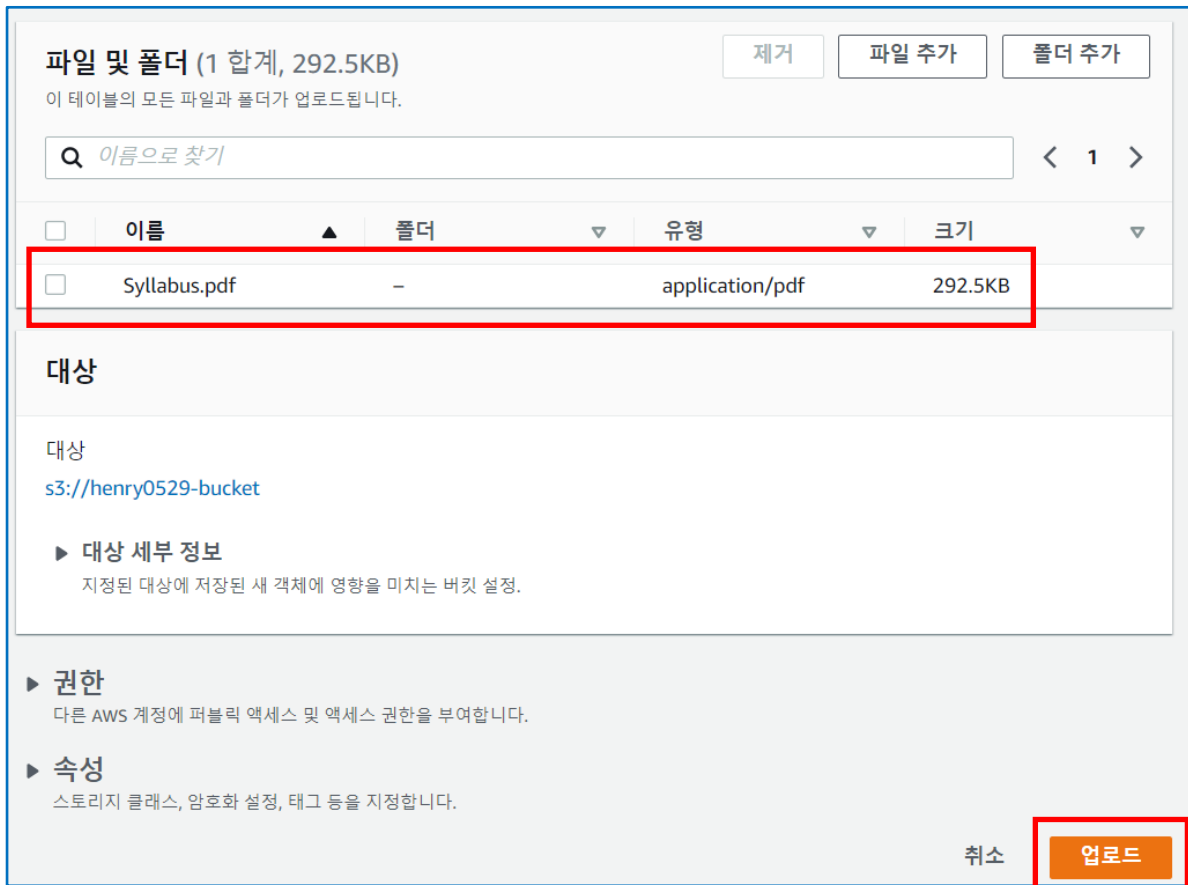
☒ 입력과 출력 모두에 동일한 S3 버킷을 사용하는 것은 권장되지 않으며, 이 구성으로 인해 재귀 호출, Lambda 사용량 증가 및 비용 증가가 발생할 수 있음을 알고 있습니다.

취소 **추가**

16. 이제 **S3**에 오브젝트를 업로드하면 **helloworld0530**이라는 람다함수가 작동할 것이다.



17. 파일 한 개를 업로드한다. 실습에서는 **Syllabus.pdf** 파일을 업로드하였다.



18. 파일 업로드를 성공적으로 마쳤으면 **Lambda** 페이지로 이동한다.

이 페이지에서 나가면 아래의 정보를 더 이상 확인할 수 없습니다.

### 요약

대상 s3://henry0529-bucket	성공 1개 파일, 292.5KB (100.00%)	실패 0개 파일, 0B (0%)
-----------------------------	--------------------------------	----------------------

### 파일 및 폴더

파일 및 폴더 (1 합계, 292.5KB)

이름으로 찾기

이름	종류	유형	크기	상태	오류
Syllabus.pdf	-	application/pdf	292.5KB	성공	-

19. [모니터링] 탭의 [로그] 탭을 클릭한다.

코드 테스트 **모니터링** 구성 별칭 버전

지표 로그 추적 정보 CloudWatch에서 로그 보기 ServiceLens에서 X-Ray 트레이스 보기 Lambda Insights 보기

CloudWatch 지표 정보 필터링 기준 합수

Lambda는 함수에 대한 런타임 지표를 Amazon CloudWatch로 전송합니다. 표시된 지표는 모든 함수 런타임 활동의 집계 보기입니다. 정규화되지 않은 리소스 또는 \$LATEST 리소스에 대한 지표를 보려면 **필터링 기준**을 선택합니다. 특정 함수 버전 또는 별칭에 대한 지표를 보려면 **별칭** 또는 **버전**을 선택하고 해당 별칭 또는 버전을 선택한 다음 **모니터링**을 선택합니다.

1시간 3시간 12시간 1일 3일 1주 사용자 지정 대시보드에 추가

#### Invocations

Count

#### Duration

Milliseconds

#### Error count and success rate (%)

Count

20. 목록 중 가장 마지막 **Timestamp**를 클릭해 보자.

지표 **로그** 추적 정보 CloudWatch에서 로그 보기 ServiceLens에서 X-Ray 트레이스 보기 Lambda Insights 보기

### CloudWatch Logs 통찰력 정보

Lambda는 함수에서 처리하는 모든 요청을 로깅하고 코드로 생성된 로그를 Amazon CloudWatch Logs를 통해 자동으로 저장합니다. 코드를 검증하려면 사용자 지정 로깅 명령문을 사용하여 코드를 확인합니다. 다음 테이블에는 모든 함수 활동에서 가장 최근에 가장 비용이 많이 드는 함수 호출이 나열되어 있습니다. 특정 함수 버전 또는 별칭에 대한 로그를 보려면 해당 수준의 **[모니터링]** 섹션을 참조하세요.

1시간 3시간 12시간 1일 3일 1주 사용자 지정 대시보드에 추가

#### Recent invocations


#	Timestamp	RequestID	LogStream	DurationInMs	BilledDurationInMs	MemorySet
1	2022-05-29T05:32:54.787Z	61764a41-2a72-42b4-a2bd-33d5cd6cae4a	2022/05/29/[\$LATEST]e9311d8113724b9fb64e25657cf6880	6.77	7	128
2	2022-05-29T05:12:14.171Z	2d8f717e-62ab-4157-9358-8d33629b4604	2022/05/29/[\$LATEST]12498c62f5b7a431b80015bc46b97d8e9	15.34	16	128
3	2022-05-29T04:41:55.996Z	ab22b81a-a093-44b8-ae58-2cf9e9a9e975	2022/05/29/[\$LATEST]b58e66122d54499f9892416359a2d8a3	1.32	2	128
4	2022-05-29T04:38:47.503Z	8493121d-3652-400b-8945-ae9c7483198e	2022/05/29/[\$LATEST]b58e66122d54499f9892416359a2d8a3	1.37	2	128
5	2022-05-29T04:35:57.478Z	d87b1dce-cb20-4741-9877-9935ac38ea93	2022/05/29/[\$LATEST]b58e66122d54499f9892416359a2d8a3	1.44	2	128
6	2022-05-29T04:35:53.421Z	80968298-634d-4568-99ec-20ccc0f9f89e	2022/05/29/[\$LATEST]b58e66122d54499f9892416359a2d8a3	1.31	2	128
7	2022-05-29T04:34:39.681Z	f025ade6-cd02-40f4-8cb6-ebd7648b31d3	2022/05/29/[\$LATEST]b58e66122d54499f9892416359a2d8a3	1.68	2	128
8	2022-05-29T04:33:24.639Z	5a44e41c-23f4-49e3-9e13-2893a7bf8e82	2022/05/29/[\$LATEST]2cb68c2aae97463d93e09f4ba6f75898	1.41	2	128
9	2022-05-29T04:32:34.557Z	43996d1c-b33e-471c-94b0-c7a5a4b603e2	2022/05/29/[\$LATEST]2cb68c2aae97463d93e09f4ba6f75898	8.3	9	128



21. CloudWatch 로그 이벤트의 메시지에 위에서 업로드한 파일의 이벤트가 잘 처리됐음을 확인할 수 있다.

CloudWatch > Log groups > /aws/lambda/helloworld0530 > 2022/05/29/[\$LATEST]e9311d8113724b9fbf64e25657cf6880

### 로그 이벤트

아래의 필터 막대를 사용하여 로그 이벤트의 용어, 구문 또는 값을 검색하고 매칭할 수 있습니다. [필터 패턴에 대해 자세히 알아보기](#)

☐ 텍스트로 보기  **작업** ▼

1m 30m 1h 12h **Custom (3h)**  

▶ 타임스탬프	메시지
	현재 이전 이벤트가 없습니다. <a href="#">재시도</a>
▶ 2022-05-29T14:32:54.780+09:00	START RequestId: 61764a41-2a72-42b4-a2bd-33d5cd6cae4a Version: \$LATEST
▶ 2022-05-29T14:32:54.780+09:00	Syllabus.pdf object is in henry0529-bucket bucket!
▶ 2022-05-29T14:32:54.787+09:00	END RequestId: 61764a41-2a72-42b4-a2bd-33d5cd6cae4a
▶ 2022-05-29T14:32:54.787+09:00	REPORT RequestId: 61764a41-2a72-42b4-a2bd-33d5cd6cae4a Duration: 6.77 ms Billed Duration: 7 ms Memory Size: 128 MB Max Mem...
	현재 최신 이벤트가 없습니다. 자동 재시도를 일시 <a href="#">중지</a> 했습니다.