

Programming of S3 using Lambda

1. S3 Bucket List
2. Object List
3. Object Download
4. Object Upload

1. S3 Bucket List

1) Lambda Function

① 함수코드

Function Name : ex-lambda-s3-1

```
import json
import boto3
import botocore

def lambda_handler(event, context):
    s3 = boto3.resource("s3")
    bucket_list = []
    for bucket in s3.buckets.all():
        bucket_list.append(bucket.name)

    return {
        'statusCode': 200,
        'body': json.dumps(bucket_list)
    }
```

② 함수 정책 추가 : 구성 ➔ 권한 ➔ 실행 역할 ➔ 역할 이름 클릭 ➔ 권한 정책 ➔ 권한 추가 ➔ 정책연결
➔ AmazonS3FullAccess ➔ 정책 연결

2) 테스트

- Test Event : te-ex-lambda-s3-1

- Event JSON : 무관

3) 테스트 결과

Bucket이 없을 경우

```
{
  "statusCode": 200,
  "body": "[]"
}
```

Bucket이 있을 경우

```
{
  "statusCode": 200,
  "body": "[\"ex-lambda-s3-1\", \"ex-lambda-s3-2\"]"
}
```

2. Object list

1) Lambda Function

① 함수코드

Function Name : ex-lambda-s3-2

```
import json
import boto3
import botocore
import os

def lambda_handler(event, context):
    bucket_name = event ["bucketName"]
    s3 = boto3.resource("s3")
    bucket = s3.Bucket(bucket_name)
    obj_list = []
    print("< 파일 목록 >")
    # 디렉토리 지정할 경우 for obj in bucket.objects.filter(Prefix = ""):
    for obj in bucket.objects.all():
        obj_list.append(obj.key)

    return {
        "statusCode": 200,
        "body": json.dumps(obj_list)
    }
```

② 함수 정책 추가 : 구성 → 권한 → 실행 역할 → 역할 이름 클릭 → 권한 정책 → 권한 추가 → 정책연결
→ AmazonS3FullAccess → 정책 연결

2) 테스트

Test Event Name : te-ex-lambda-s3-2

Event JSON

```
{
  "bucketName": "bucket-name"
}
```

3) 테스트 결과

Response

```
{
  "statusCode": 200,
  "body": "[\"images/\", \"images/test-img.jpg\", \"test-img.jpg\"]"
}
```

3. Object Download

1) Lambda Function

① 함수코드

Function Name : ex-lambda-s3-3

```
import boto3
import base64
from urllib import parse

def lambda_handler(event, context):
    bucket = event["queryStringParameters"]["bucketName"]
    object = event["queryStringParameters"]["objectName"]
    # 버킷에 있는 파일 읽기
    s3=boto3.resource("s3")
    obj=s3.Object(bucket,object)
    myBody=obj.get()['Body'].read()
    object = parse.quote(object) → URL 문자열을 한글파일명 적용을 위한 인코딩
    return {
        "isBase64Encoded": True, → 내용이 이진데이터를 담고 있는지, 기본값은 False
        "statusCode": 200,
        "headers": {
            "Cache-Control": "no-cache",
            "Content-Type": "application/octet-stream", → 다운로드 받기 위한 헤더 설정
            "Content-Disposition": "inline; filename="+object
        },
        "body": base64.b64encode(myBody) → 이진 데이터를 텍스트 형태로 변환
    }
```

② 함수 정책 추가 : 구성 → 권한 → 실행 역할 → 역할 이름 클릭 → 권한 정책 → 권한 추가 → 정책연결
→ AmazonS3FullAccess → 정책 연결

2) API Gateway 설정 : ex-lambda-s3-3-API

① 통합유형 설정 변경

메서드 → 통합요청 → “Lambda 프록시 통합 사용”에 체크



② API Gateway 설정 변경

“application/octet-stream”와 “*/*”를 이진 미디어 형식에 추가

API : *api-gate-name* → 설정

이진 미디어 형식

이진 유형으로 처리할 미디어 유형을 지정하여 API의 이진 지원을 구성할 수 있습니다. API Gateway는 **Content-Type** 및 **Accept** HTTP 헤더를 살펴보고 본문의 처리 방식을 결정합니다.

[+ 이진 미디어 형식 추가](#)

4) 메서드 배포

리소스 → 메서드 → “작업”버튼 → 배포

API 배포

API를 배포할 스테이지를 선택합니다. 예를 들어, API 테스트 버전이라면 베타라는 이름의 단계에 배포할 수 있습니다.

배포 스테이지

default

배포 설명

취소

배포

4) 테스트

스테이지 → default → 메서드(Any/Get ...) → “url 호출” → 파라미터 추가

```
https://urlPath.../default/ex-lambda-s3-3?bucketName=ex-lambda-s3-1&objectName=울산과학대학교.pptx
```

4. Object Upload

1) Lambda Function

① 함수코드

Function Name : ex-lambda-s3-4

```
import json
import boto3
import datetime

def lambda_handler(event, context):
    bucket = "ex-lambda-s3-1"
    file_name = str(datetime.datetime.now())[:7]
    file = dict()
    file['customerID'] = 'jinyes'
    file['age'] = '25'
    file['product'] = 'aws_solution'
    result = upload_file_s3(bucket, file_name + '.json', file)

    if result:
        return {
            'statusCode': 200,
            'body': json.dumps("upload success")
        }
    else:
        return {
            'statusCode': 400,
            'body': json.dumps("upload fail")
        }

def upload_file_s3(bucket, file_name, file):
    encode_file = bytes(json.dumps(file).encode('UTF-8'))
    s3 = boto3.client('s3')
    try:
        s3.put_object(Bucket=bucket, Key=file_name, Body=encode_file)
        return True
    except:
        return False
```

② 함수 정책 추가 : 구성 ➔ 권한 ➔ 실행 역할 ➔ 역할 이름 클릭 ➔ 권한 정책 ➔ 권한 추가 ➔ 정책연결
➔ AmazoneS3FullAccess ➔ 정책 연결

2) 테스트

S3 서비스 버킷을 통해 확인

ex-lambda-s3-1 정보

객체 | 속성 | 권한 | 지표 | 관리 | 액세스 지정

객체 (4)

객체는 Amazon S3에 저장되어 있는 기본 엔티티입니다. Amazon S3 [인벤토리](#)를 사용하여 버킷에 있는 모든 객체의 목록을 얻을 수 있습니다. 다른 사용자가 객체에 액세스할 수 있게 하려면 명시적으로 권한을 부여해야 합니다. 자세히 알아보기

🔍 접두사로 객체 찾기

< 1 > ⚙️

<input type="checkbox"/>	이름	▲	유형 ▼	마지막 수정 ▼	크기 ▼	스토리지 클래스 ▼
<input type="checkbox"/>	198x238.jpg		jpg	2022. 12. 15. am 5:23:43 AM KST	36.5KB	Standard
<input type="checkbox"/>	2022-12-15 03:26:35.json		json	2022. 12. 15. pm 12:26:38 PM KST	60.0B	Standard
<input type="checkbox"/>	울산과학기술대학교.pptx		pptx	2022. 12. 15. am 6:49:21 AM KST	544.0KB	Standard
<input type="checkbox"/>	test-img.jpg		jpg	2022. 12. 13. pm 5:46:53 PM KST	228.6KB	Standard