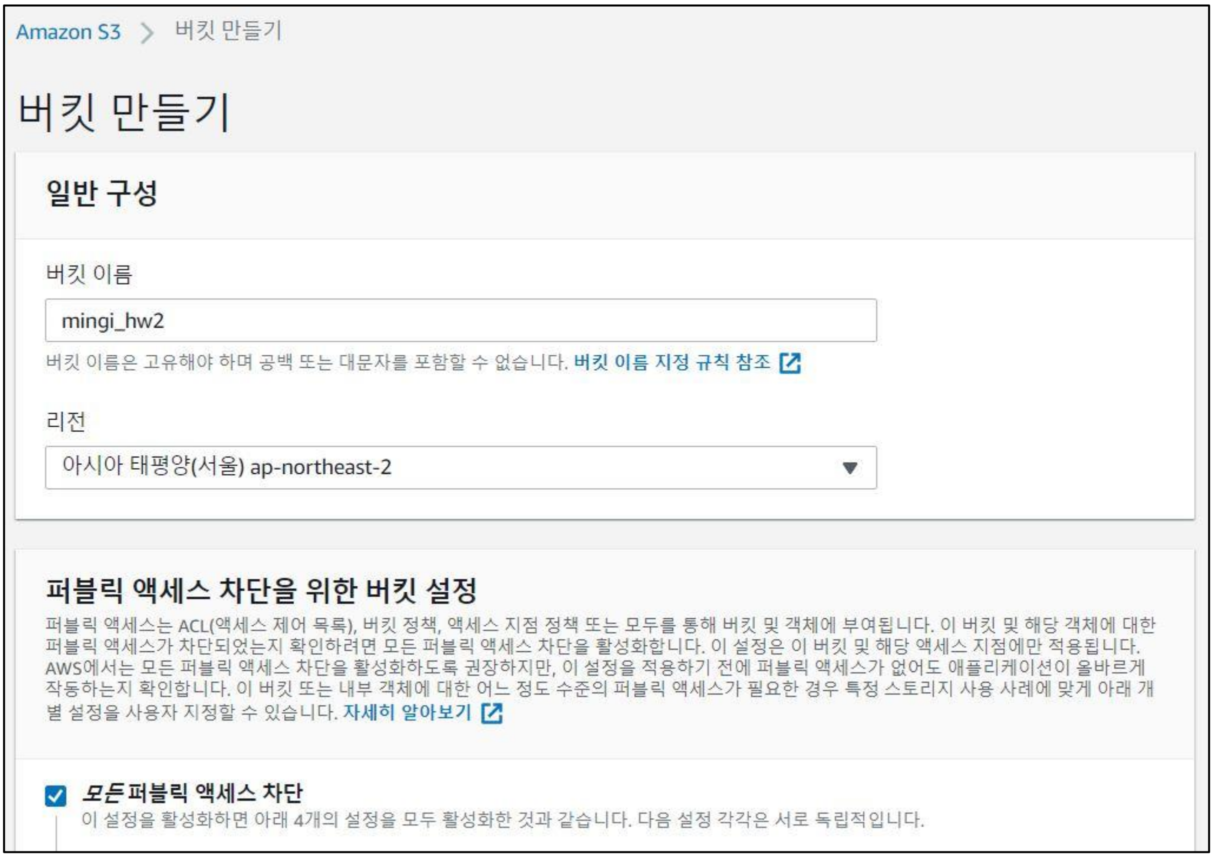
REPORT

과목명	클라우드IOT서비스
담당교수	정 갑 주 교수님
학과	컴퓨터공학부
학년	4학년
학번	201714151
이름	박 민 기
제출일	2020. 05. 11

Contents

1. Task #1

1-1. Create AWS S3 Bucket



버킷 만들기

일반 구성

버킷 이름: mingi_hw2

버킷 이름은 고유해야 하며 공백 또는 대문자를 포함할 수 없습니다. 버킷 이름 지정 규칙 참조

리전: 아시아 태평양(서울) ap-northeast-2

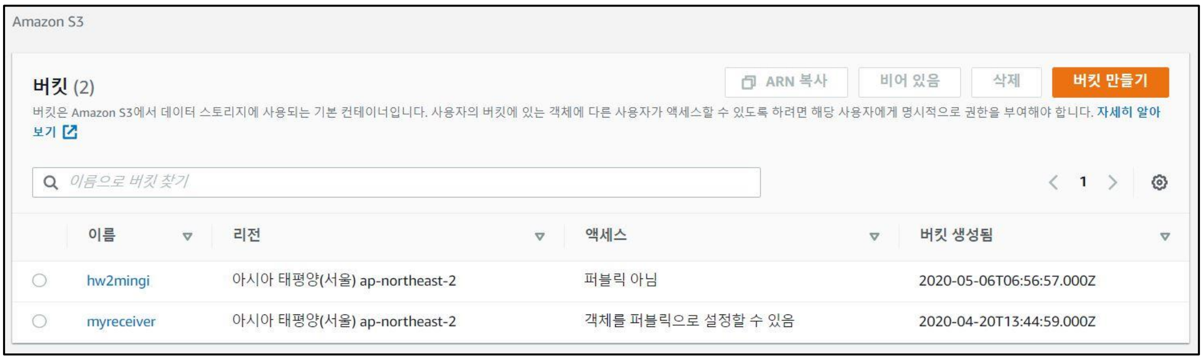
퍼블릭 액세스 차단을 위한 버킷 설정

퍼블릭 액세스는 ACL(액세스 제어 목록), 버킷 정책, 액세스 지점 정책 또는 모두를 통해 버킷 및 객체에 부여됩니다. 이 버킷 및 해당 객체에 대한 퍼블릭 액세스가 차단되었는지 확인하려면 모든 퍼블릭 액세스 차단을 활성화합니다. 이 설정은 이 버킷 및 해당 액세스 지점에만 적용됩니다. AWS에서는 모든 퍼블릭 액세스 차단을 활성화하도록 권장하지만, 이 설정을 적용하기 전에 퍼블릭 액세스가 없어도 애플리케이션이 올바르게 작동하는지 확인합니다. 이 버킷 또는 내부 객체에 대한 어느 정도 수준의 퍼블릭 액세스가 필요한 경우 특정 스토리지 사용 사례에 맞게 아래 개별 설정을 사용자 지정할 수 있습니다. 자세히 알아보기

☑ *모든* 퍼블릭 액세스 차단

이 설정을 활성화하면 아래 4개의 설정을 모두 활성화한 것과 같습니다. 다음 설정 각각은 서로 독립적입니다.

1-2. Configure S3 Bucket

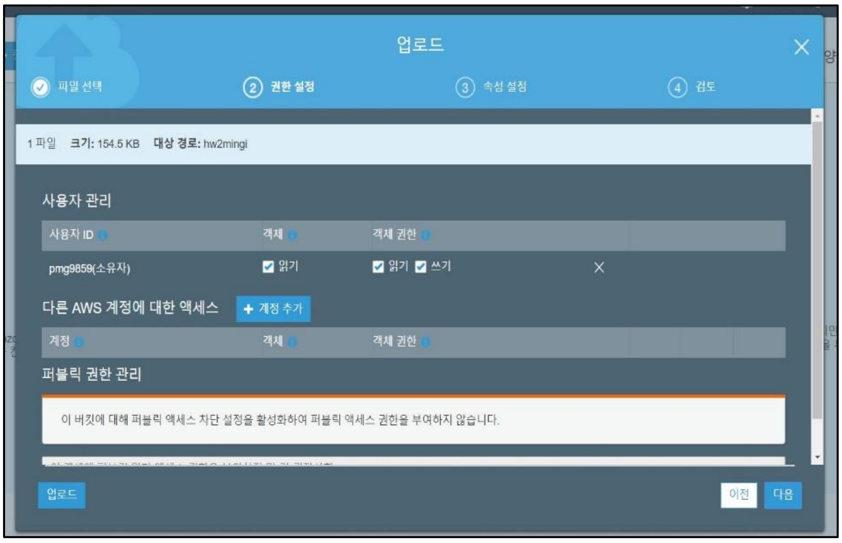


버킷 (2)

버킷은 Amazon S3에서 데이터 스토리지에 사용되는 기본 컨테이너입니다. 사용자의 버킷에 있는 객체에 다른 사용자가 액세스할 수 있도록 하려면 해당 사용자에게 명시적으로 권한을 부여해야 합니다. 자세히 알아보기

	이름	리전	액세스	버킷 생성됨
○	hw2mingi	아시아 태평양(서울) ap-northeast-2	퍼블릭 아님	2020-05-06T06:56:57.000Z
○	myreceiver	아시아 태평양(서울) ap-northeast-2	객체를 퍼블릭으로 설정할 수 있음	2020-04-20T13:44:59.000Z

1-3. Uploading a file from my notebook through AWS Web Console



1 파일 **크기:** 154.5 KB **대상 경로:** hw2mingi

사용자 관리

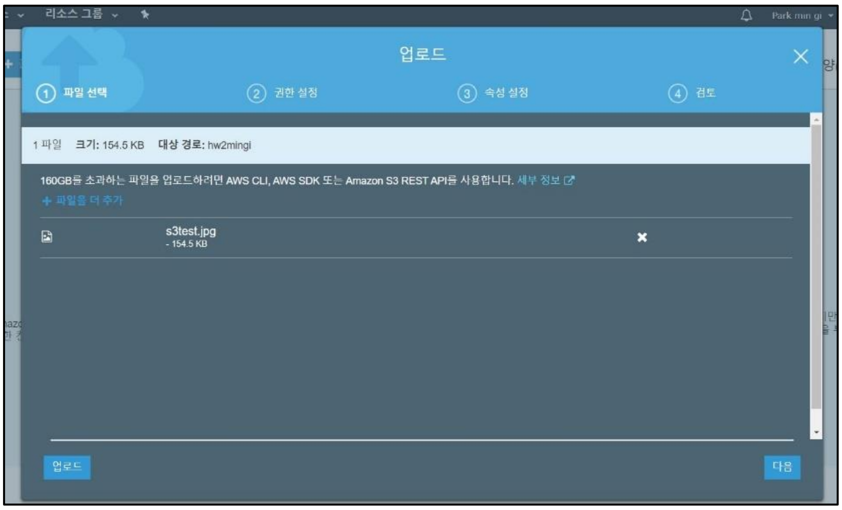
사용자 ID	객체	객체 권한
pmg9859(소유자)	☑ 읽기	☑ 읽기 ☑ 쓰기

다른 AWS 계정에 대한 액세스 + 계정 추가

계정	객체	객체 권한

퍼블릭 권한 관리

이 버킷에 대해 퍼블릭 액세스 차단 설정을 활성화하여 퍼블릭 액세스 권한을 부여하지 않습니다.

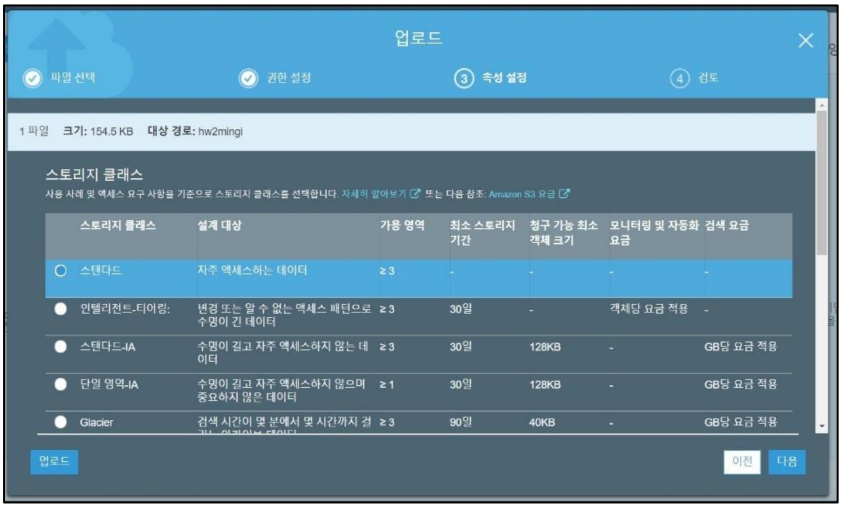


1 파일 **크기:** 154.5 KB **대상 경로:** hw2mingi

160GB를 초과하는 파일을 업로드하려면 AWS CLI, AWS SDK 또는 Amazon S3 REST API를 사용합니다. 세부 정보

+ 파일을 더 추가

s3test.jpg
- 154.5 KB



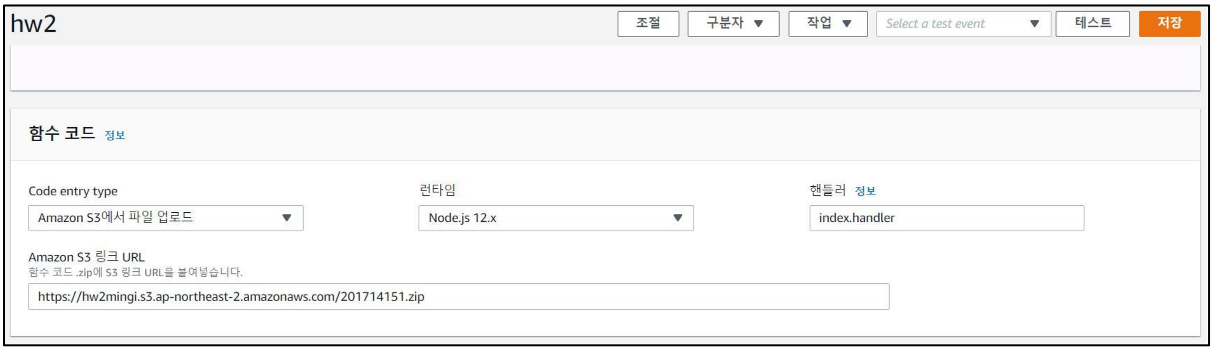
1 파일 **크기:** 154.5 KB **대상 경로:** hw2mingi

스토리지 클래스

사용 사례 및 액세스 요구 사항을 기준으로 스토리지 클래스를 선택합니다. 자세히 알아보기 또는 다음 참조: Amazon S3 요금

	스토리지 클래스	설계 대상	가용 영역	최소 스토리지 기간	청구 가능 최소 객체 크기	모니터링 및 자동화 요금	검색 요금
○	스탠다드	자주 액세스하는 데이터	≥ 3	-	-	-	-
●	인텔리전트-티어링:	변경 또는 알 수 없는 액세스 패턴으로 수명이 긴 데이터	≥ 3	30일	-	객체당 요금 적용	-
●	스탠다드-IA	수명이 길고 자주 액세스하지 않는 데이터	≥ 3	30일	128KB	-	GB당 요금 적용
●	단일 영역-IA	수명이 길고 자주 액세스하지 않으며 중요하지 않은 데이터	≥ 1	30일	128KB	-	GB당 요금 적용
●	Glacier	검색 시간이 몇 분에서 몇 시간까지 걸	≥ 3	90일	40KB	-	GB당 요금 적용

1-4. Check Upload file through AWS Web Console



함수 코드 정보

Code entry type: Amazon S3에서 파일 업로드

런타임: Node.js 12.x

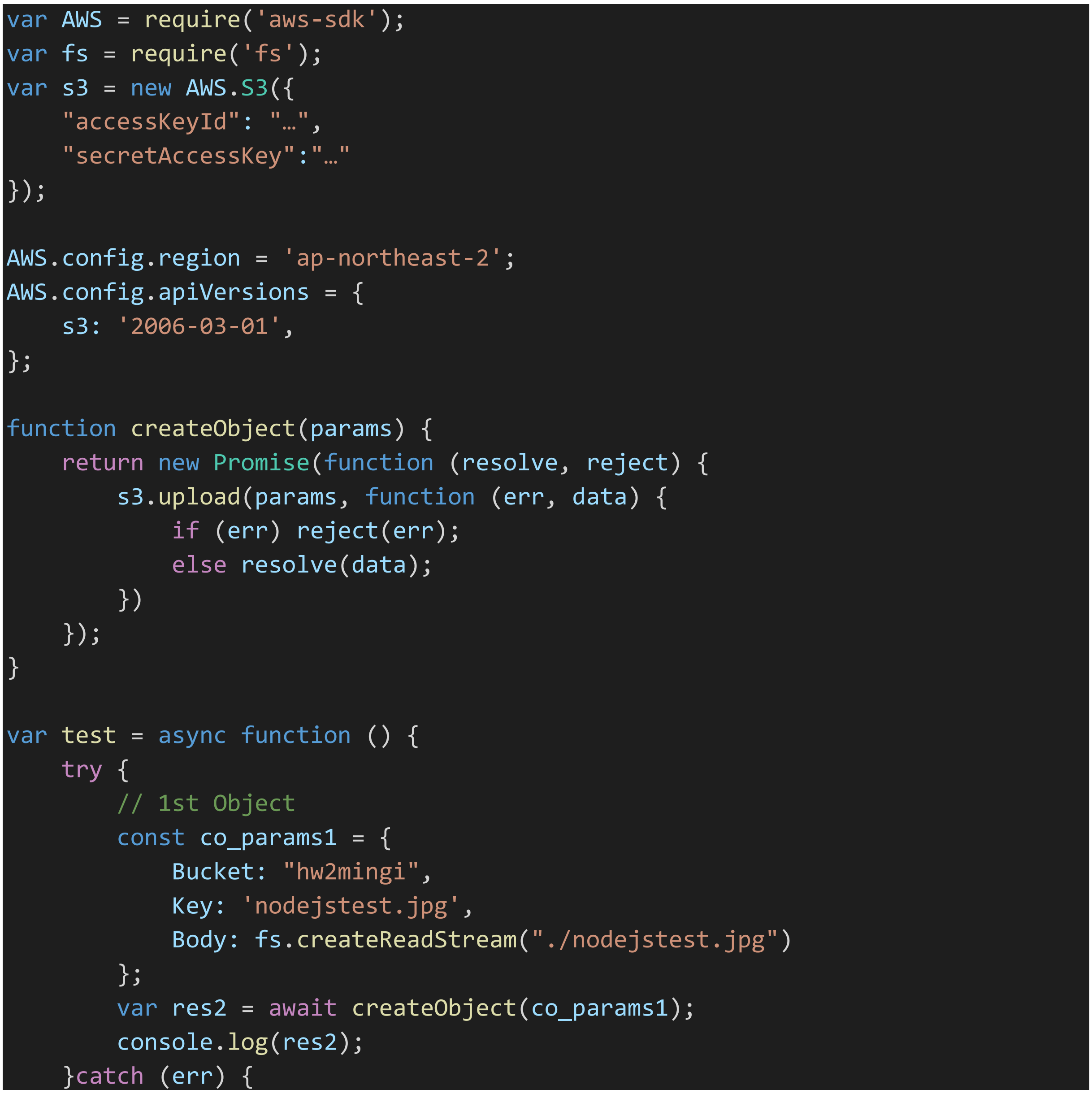
핸들러 정보: index.handler

Amazon S3 링크 URL

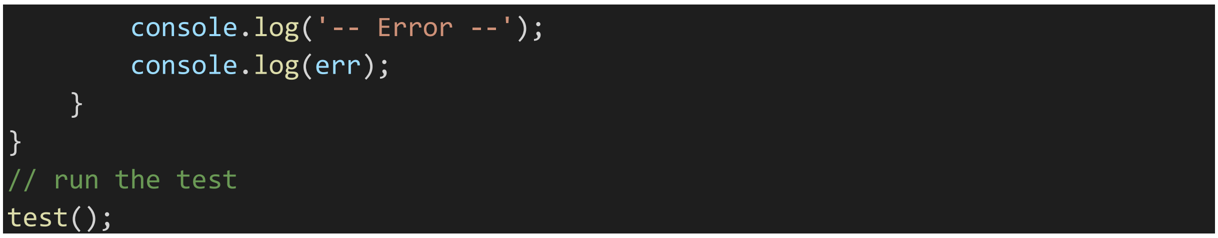
함수 코드 .zip에 S3 링크 URL을 붙여넣습니다.

https://hw2mingi.s3.ap-northeast-2.amazonaws.com/201714151.zip

1-5. Create a Node.JS Application to create a S3 object in the bucket (Code)

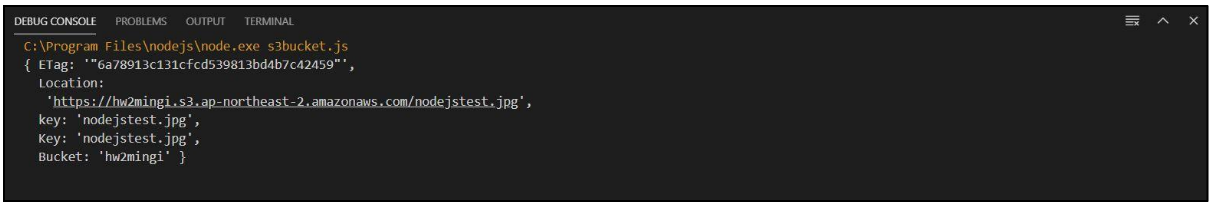


```
var AWS = require('aws-sdk');
var fs = require('fs');
var s3 = new AWS.S3({
    "accessKeyId": "…",
    "secretAccessKey":"…"
});

AWS.config.region = 'ap-northeast-2';
AWS.config.apiVersions = {
    s3: '2006-03-01',
};

function createObject(params) {
    return new Promise(function (resolve, reject) {
        s3.upload(params, function (err, data) {
            if (err) reject(err);
            else resolve(data);
        })
    });
}

var test = async function () {
    try {
        // 1st Object
        const co_params1 = {
            Bucket: "hw2mingi",
            Key: 'nodejstest.jpg',
            Body: fs.createReadStream("./nodejstest.jpg")
        };
        var res2 = await createObject(co_params1);
        console.log(res2);
    }catch (err) {
```



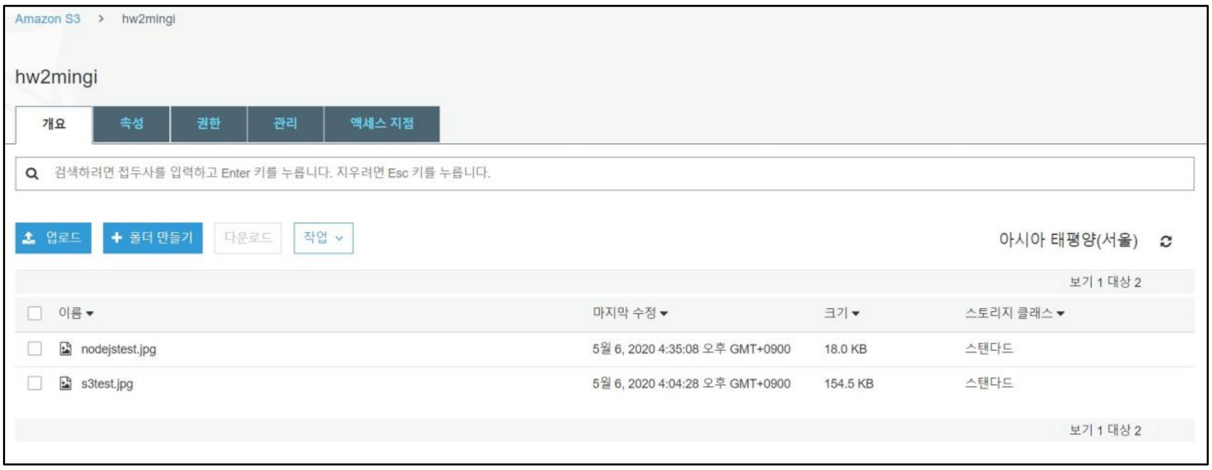
```
        console.log('-- Error --');
        console.log(err);
    }
}
// run the test
test();
```

1-6. Compile



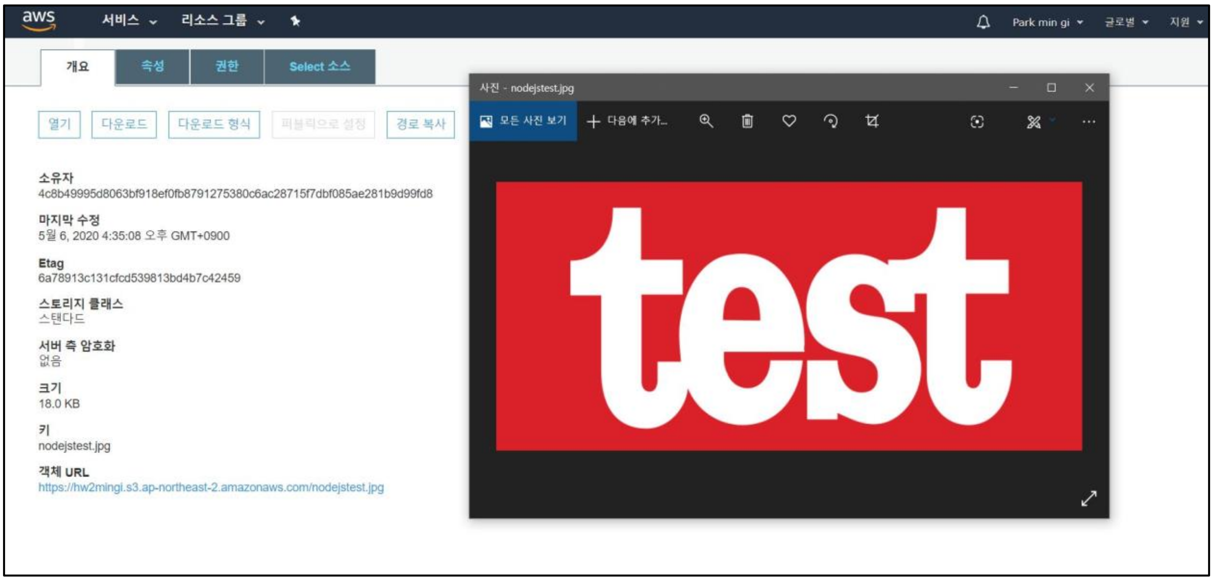
```
C:\Program Files\nodejs\node.exe s3bucket.js
{ ETag: '"6a78913c131cfcd539813bd4b7c42459"',
  Location:
   'https://hw2mingi.s3.ap-northeast-2.amazonaws.com/nodejstest.jpg',
  key: 'nodejstest.jpg',
  Key: 'nodejstest.jpg',
  Bucket: 'hw2mingi' }
```

1-7. Configure S3 Object through AWS Web Console



이름	마지막 수정	크기	스토리지 클래스
□ nodejstest.jpg	5월 6, 2020 4:35:08 오후 GMT+0900	18.0 KB	스탠다드
□ s3test.jpg	5월 6, 2020 4:04:28 오후 GMT+0900	154.5 KB	스탠다드

1-8. Configure Image



소유자
4c8b49995d8063bf918ef0fb8791275380c6ac28715f7dbf085ae281b9d99fd8

마지막 수정
5월 6, 2020 4:35:08 오후 GMT+0900

Etag
6a78913c131cfcd539813bd4b7c42459

스토리지 클래스
스탠다드

서버 측 암호화
없음

크기
18.0 KB

키
nodejstest.jpg

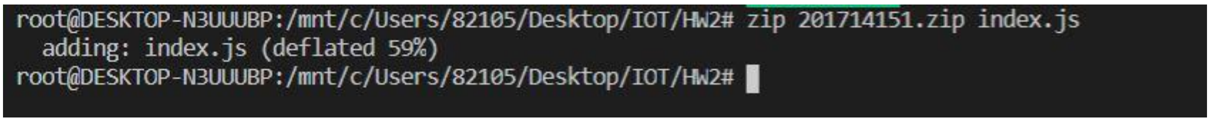
객체 URL
https://hw2mingi.s3.ap-northeast-2.amazonaws.com/nodejstest.jpg

2. Task #2

2-1. index.js(Store input argument into the S3 object)

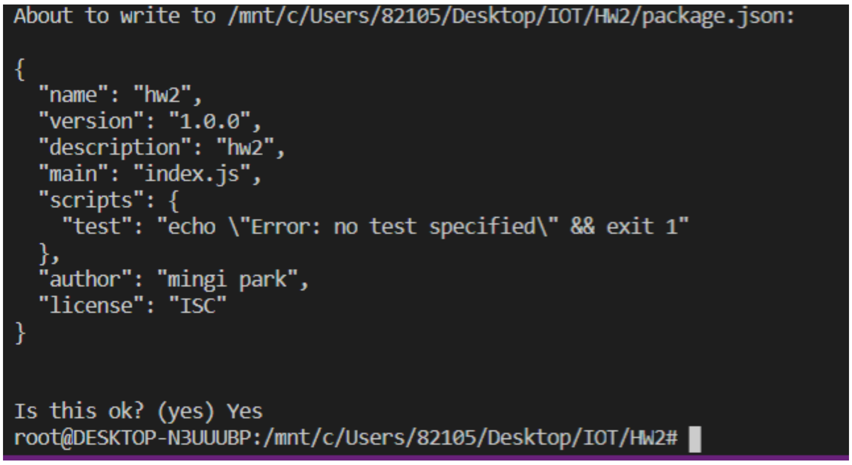
```
var keys = require('./credentials.js');
var fs = require('fs');
var AWS = require('aws-sdk');
AWS.config.region = 'ap-northeast-2';

var s3 = new AWS.S3({
    "apiVersion": '2006-03-01',
    "accessKeyId": keys.aws_access_key_id ,
    "secretAccessKey": keys.aws_secret_access_key
});

function createObject(params){
    return new Promise(function (resolve, reject) {
        s3.upload(params, function (err, data) {
            if (err) reject(err);
            else resolve(data);
        });
    });
}
exports.handler = async () => {
    var params = {
        Bucket : 'hw2mingi',
        Key : 'invoke_result.jpg',
        Body : fs.createReadStream("./test_image.jpg")
    }

    const response = {
        statusCode : 200,
        body: ''
    };

    try{
        var res = await createObject(params);
        response.body = 'Success!';
        return response;
    }catch(err){
        response.body = err; return response;
    }
};
```

2-2. zip index.js



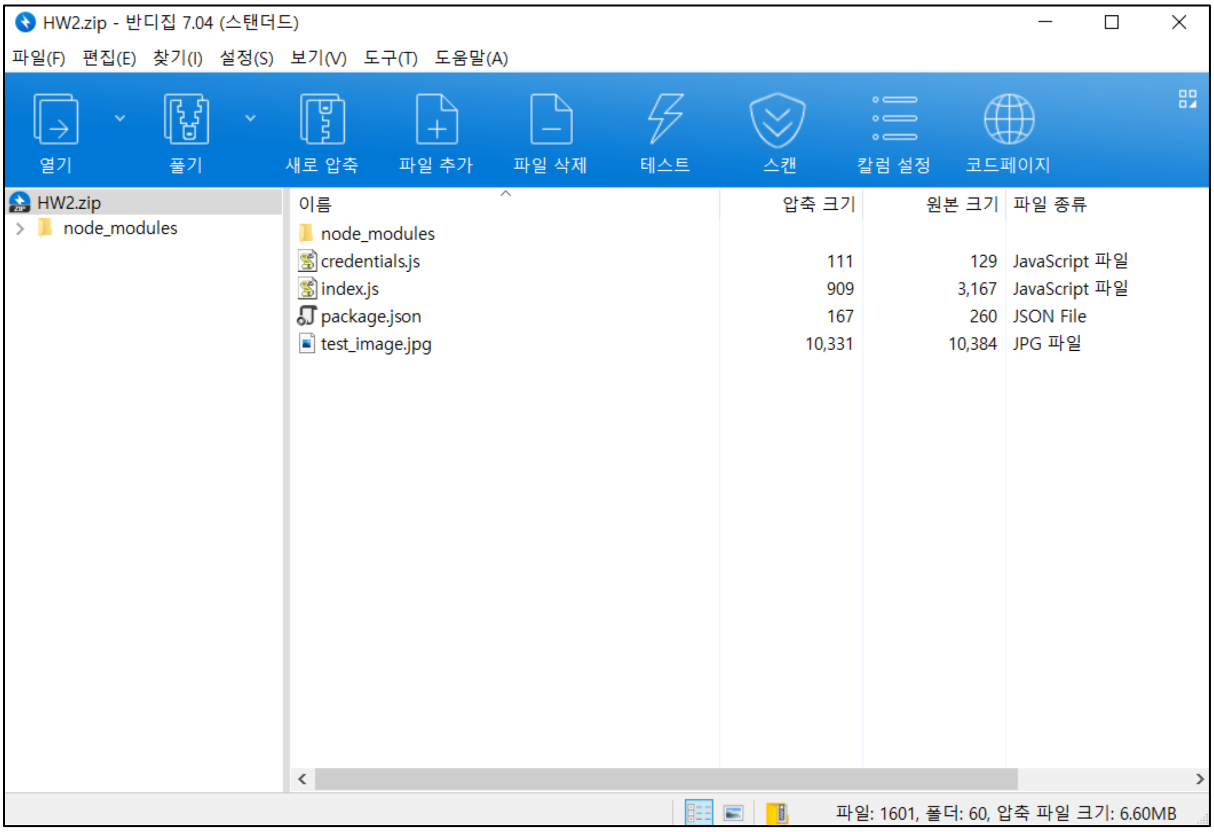
```
root@DESKTOP-N3UUUBP:/mnt/c/Users/82105/Desktop/IOT/HW2# zip 201714151.zip index.js
  adding: index.js (deflated 59%)
root@DESKTOP-N3UUUBP:/mnt/c/Users/82105/Desktop/IOT/HW2# 
```

2-3. create package.json



```
About to write to /mnt/c/Users/82105/Desktop/IOT/HW2/package.json:

{
  "name": "hw2",
  "version": "1.0.0",
  "description": "hw2",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "mingi park",
  "license": "ISC"
}


Is this ok? (yes) Yes
root@DESKTOP-N3UUUBP:/mnt/c/Users/82105/Desktop/IOT/HW2# 
```

2-4. Zip the File (module + credentials + package.json + create object(test_image.jpg) + index.js)

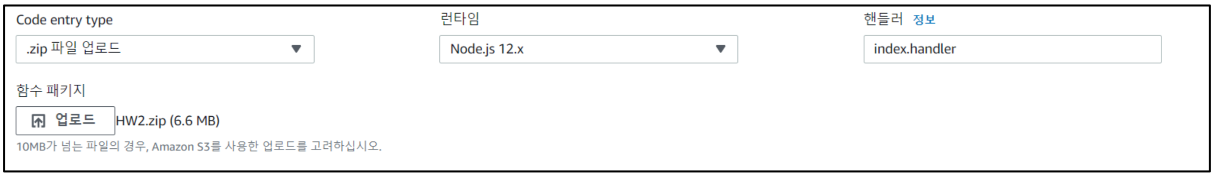


HW2.zip - 반디집 7.04 (스탠더드)
파일(F) 편집(E) 찾기(I) 설정(S) 보기(V) 도구(T) 도움말(A)
열기
풀기
새로 압축
파일 추가
파일 삭제
테스트
스캔
칼럼 설정
코드페이지
HW2.zip
node_modules

이름	압축 크기	원본 크기	파일 종류
node_modules			
credentials.js	111	129	JavaScript 파일
index.js	909	3,167	JavaScript 파일
package.json	167	260	JSON File
test_image.jpg	10,331	10,384	JPG 파일

파일: 1601, 폴더: 60, 압축 파일 크기: 6.60MB

2-5. Zip file Upload to Lambda



Code entry type: .zip 파일 업로드

런타임: Node.js 12.x

핸들러 정보: index.handler

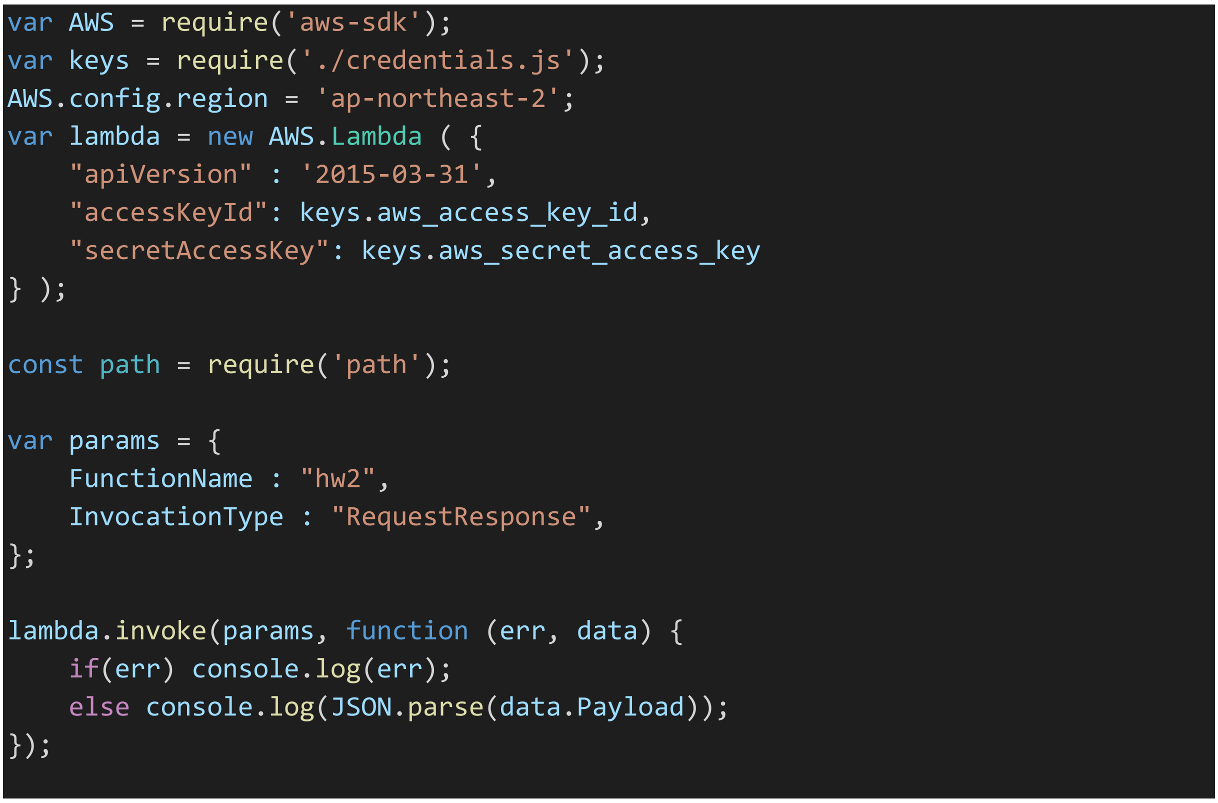
함수 패키지

업로드 HW2.zip (6.6 MB)

10MB가 넘는 파일의 경우, Amazon S3를 사용한 업로드를 고려하십시오.

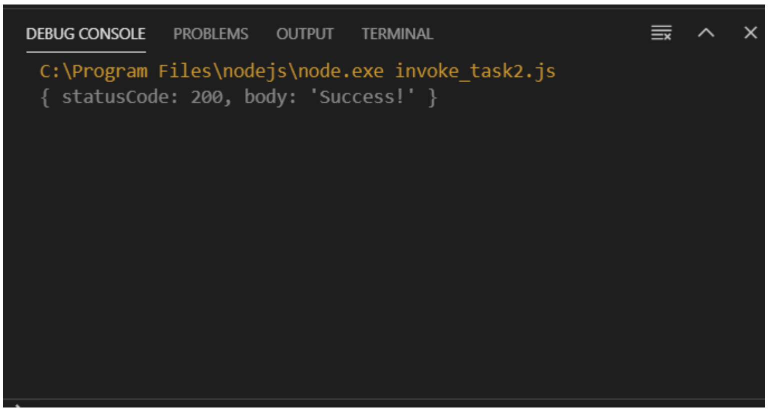
3. Task #3

3-1.invoke_task2.js



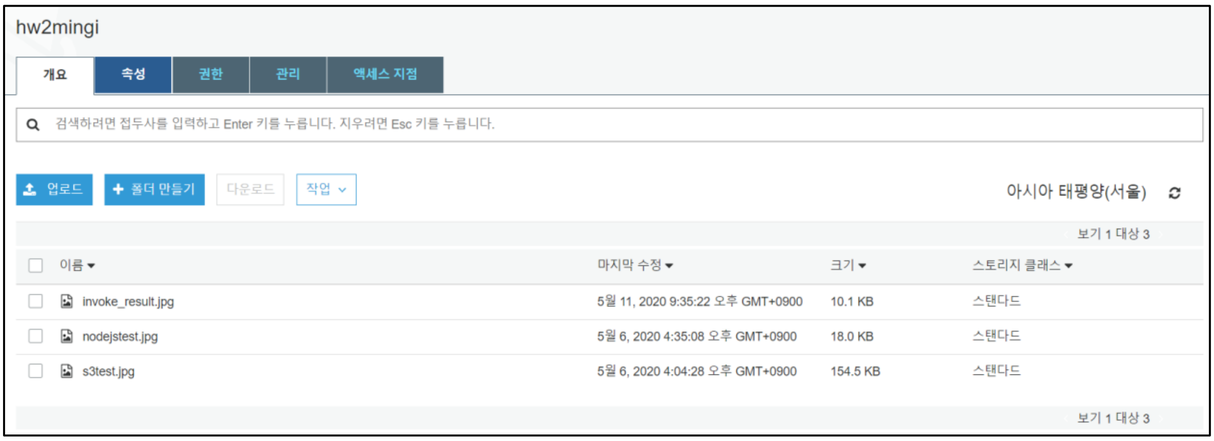
```
var AWS = require('aws-sdk');
var keys = require('./credentials.js');
AWS.config.region = 'ap-northeast-2';
var lambda = new AWS.Lambda ( {
    "apiVersion" : '2015-03-31',
    "accessKeyId": keys.aws_access_key_id,
    "secretAccessKey": keys.aws_secret_access_key
} );

const path = require('path');

var params = {
    FunctionName : "hw2",
    InvocationType : "RequestResponse",
};

lambda.invoke(params, function (err, data) {
    if(err) console.log(err);
    else console.log(JSON.parse(data.Payload));
});
```

3-2. Confiugre in the Debug Console



```
C:\Program Files\nodejs\node.exe invoke_task2.js
{ statusCode: 200, body: 'Success!' }
```

3-3. Configure at the Web Concole(S3 Bucket)



□ 이름	마지막 수정	크기	스토리지 클래스
□ invoke_result.jpg	5월 11, 2020 9:35:22 오후 GMT+0900	10.1 KB	스탠다드
□ nodejstest.jpg	5월 6, 2020 4:35:08 오후 GMT+0900	18.0 KB	스탠다드
□ s3test.jpg	5월 6, 2020 4:04:28 오후 GMT+0900	154.5 KB	스탠다드

=> I can found the result(invoke_result.jpg) at the S3 Bucket(hw2mingi) !

4. Conclusion

처음에는 Local 상에 있는 객체(파일)을 AWS 상에 옮긴다는 것이 잘 이해가 되지 않아 어려움을 겪었다. 하지만 각각의 개별 시스템을 이해하고 업로드를 원하는 객체(파일)을 .zip에 함께 압축하여 올린 뒤 Lambda 함수를 invoke 시키면 된다는 것을 깨달았다. 전체적인 시스템 구조를 자각할 수 있는 좋은 기회가 되었던 것 같다. 또한 Lambda 함수에서 함수식만이 아닌 파일 업로드까지도 수행할 수 있다는 것을 알게 되었다.