

| | | |
|------|-------------------|--|
| 제 목 | 05장 AWS S3 연결하기 | |
| 상세내용 | 파일 및 이미지 업로드 (S3) | |

1. S3란? / S3를 왜 사용하는걸까? / 현업에서 S3를 많이 사용할까?

✔ S3란?

▶ 한 줄 요약 : 파일 저장 서비스이다.

우린 사진이나 동영상과 같은 파일들을 구글 드라이브나 iCloud에 저장한다. S3는 이런 종류의 서비스라고 생각하면 된다.

✔ S3를 왜 사용하는 걸까?

백엔드 서버를 구현하다보면 이미지 업로드 기능을 구현할 때가 많다. 그럼 이 이미지 파일을 어디에 저장해야 할까? EC2 내부에 이미지 파일을 저장할 수도 있다. 하지만 서비스를 조금만 운영하다보면 EC2에 쌓이는 파일들이 너무 많아지고 지저분해지게 된다.

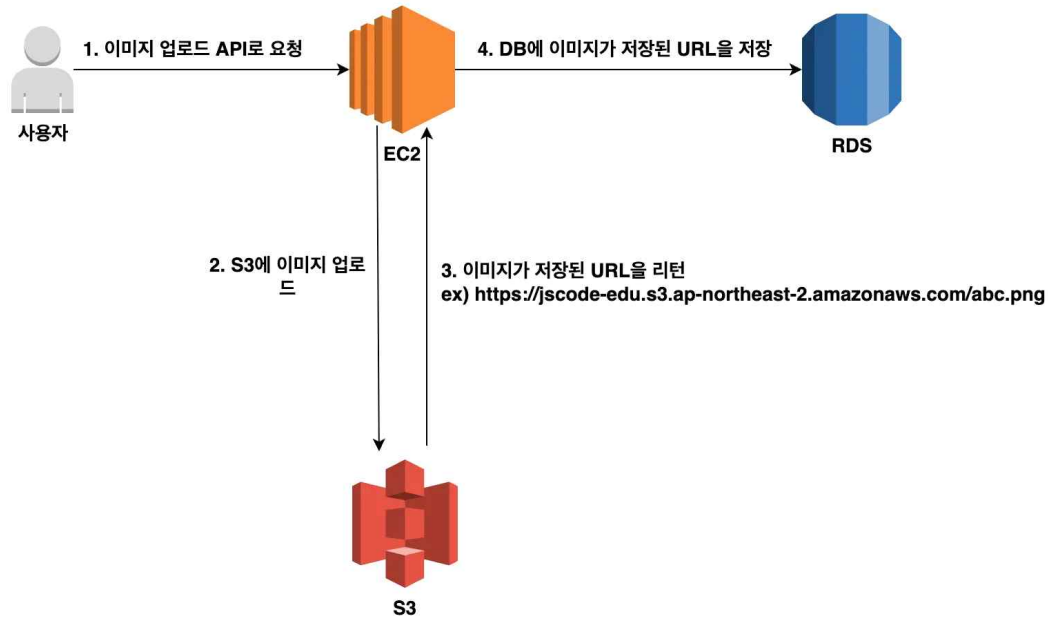
우리가 휴대폰에 저장공간이 있는데도 불구하고 구글 드라이브나 iCloud와 같은 곳에 사진을 옮기는 이유와 비슷하다. S3는 파일 저장에 특화된 서비스이다. 파일 저장 뿐만 아니라 파일을 다운받는 것에 대해서도 최적화되어 있는 서비스이다. 이러한 이유 때문에 S3를 사용한다.

✔ 현업에서 S3를 많이 사용할까?

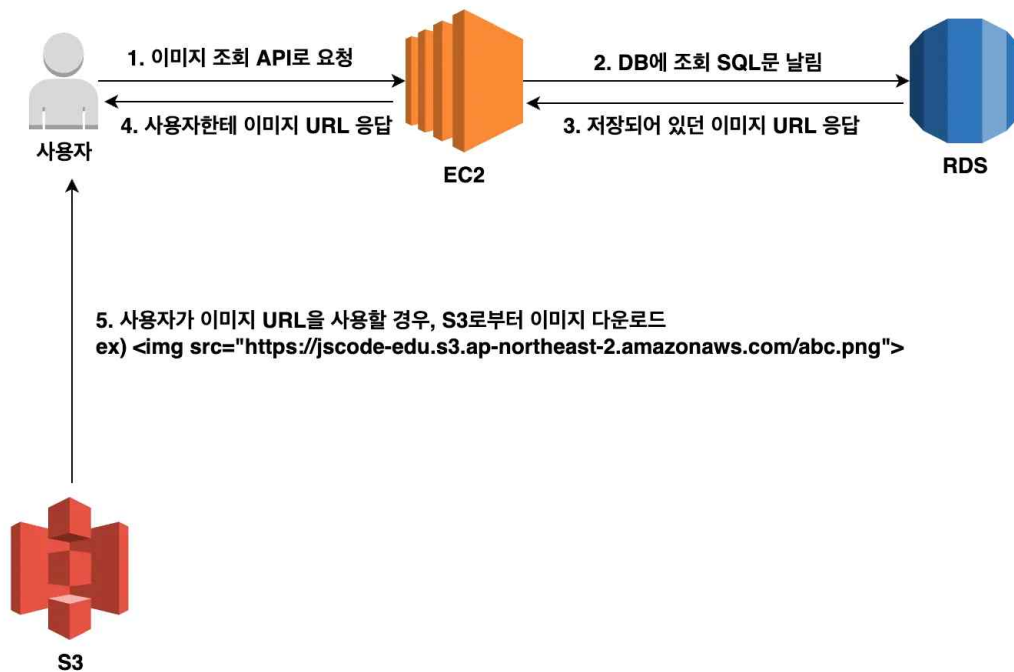
현업에서 파일 업로드 기능을 구현할 때면 대부분은 AWS S3를 활용한다.

2. S3를 활용한 아키텍처 구성

✓ 이미지 파일 업로드 과정



✓ 이미지 파일 다운로드 과정



위 예시는 이미지 파일로 예를 들었으나, S3는 이미지 이외에도 다양한 종류의 파일을 저장하는 용도로 많이 사용한다.

[실습] 1. S3 버킷 생성하기

✓ S3에서만 사용하는 용어 정리

S3에서는 **버킷(Bucket)**과 **객체(Object)**라는 용어를 사용한다. 이 용어에 대해 알아보자.

▶ 버킷(Bucket)이란?

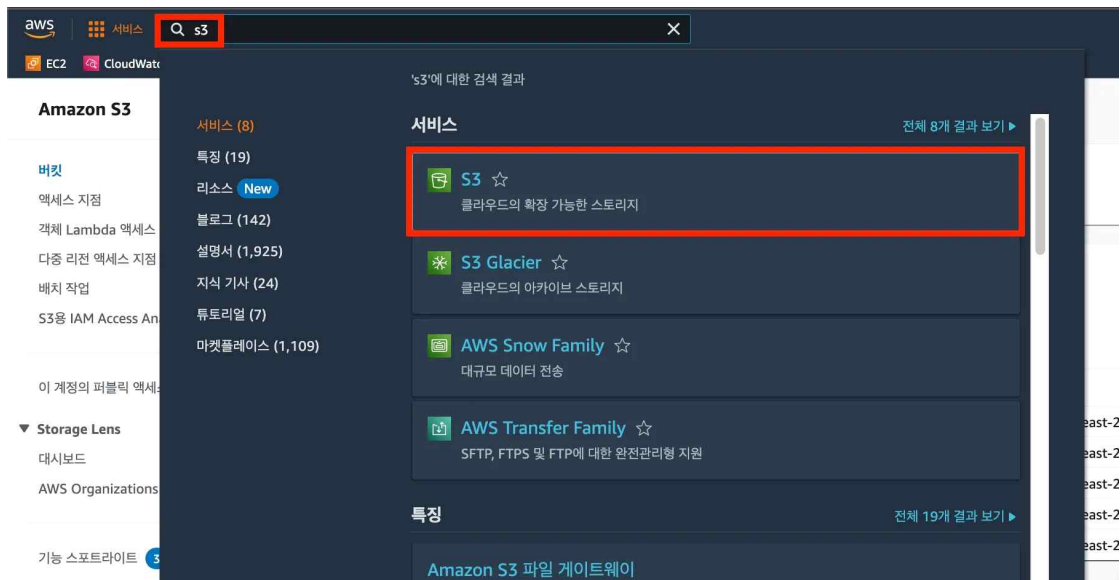
⇒ 깃헙(Github)을 보면 여러 개의 Repository를 만들 수 있다. S3에서도 여러 개의 저장소를 만들 수 있다. 여기서 **하나의 저장소를 버킷(Bucket)**이라고 부른다.

▶ 객체(Object)이란

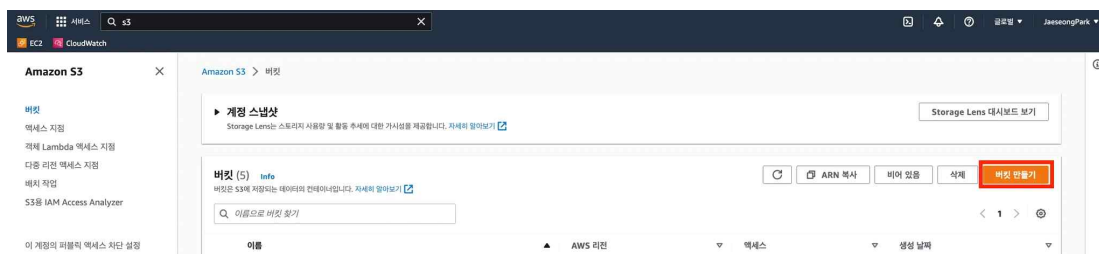
⇒ S3에 업로드한 파일을 보고, S3에서는 파일(File)이라 부르지 않고 객체(Object)라고 부른다. 즉, 객체(Object)란 **S3 버킷에 업로드된 파일**을 의미한다.

✓ 1. S3 버킷 생성하기

1. S3 선택



2. 버킷 만들기



2. 버킷 만들기

버킷 만들기

Info

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

일반 구성

버킷 이름

jscode-uploaded-files

버킷 이름은 전역에서 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

(나머지는 Default 설정으로 그대로 두기)

이 버킷의 퍼블릭 액세스 차단 설정

퍼블릭 액세스는 ACL(엑세스 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.

☐ 새 ACL(엑세스 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(엑세스 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 구체적인 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.

(나머지는 Default 설정으로 그대로 두기)

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

취소

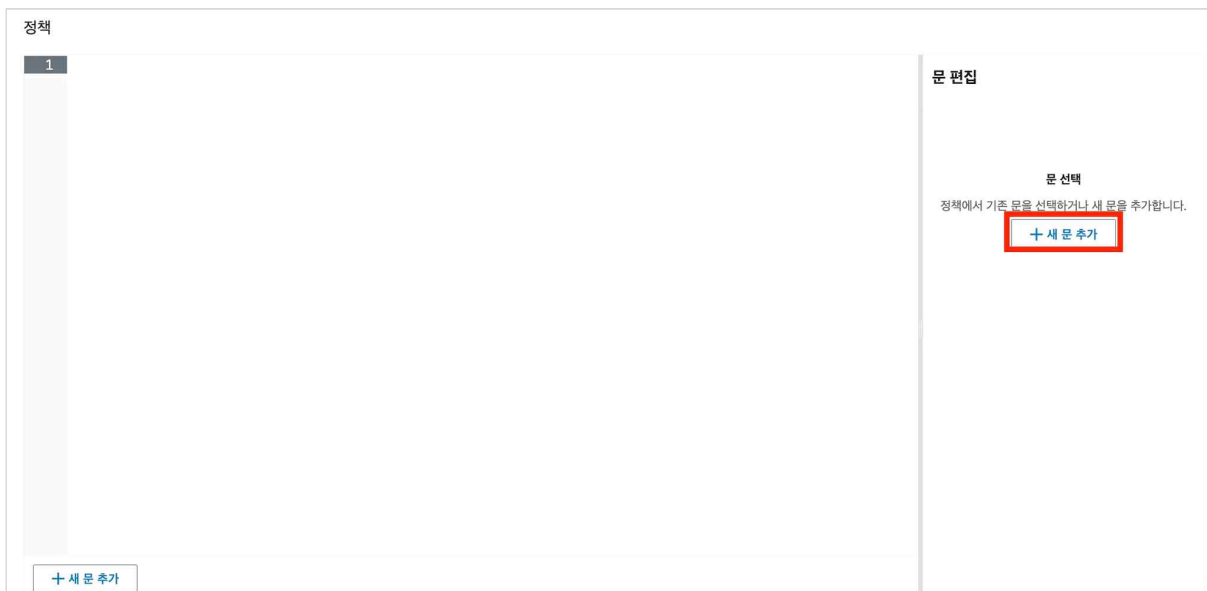
버킷 만들기

✔ 2. 버킷에 정책 추가하기

▶ 정책(Policy)이란 ?

⇒ 권한(Permission)을 정의하는 JSON 문서를 의미한다. AWS는 기본적으로 대부분의 권한이 주어져있지 않다. AWS의 특정 소스에 접근하려면 권한을 허용해 주어야 한다. 권한을 허용할 때 작성해야 하는 게 정책(Policy)이다.

▶ 특정 서비스에서 상품 이미지를 모든 사용자한테 보여주고 싶다고 가정해보자. 그러면 버킷에서 상품 이미지를 다운로드해서 사용할 수 있어야 한다. 버킷에서 이미지 파일을 조회할 수 있게 정책을 추가해보겠다.



정책

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "Statement1",
6       "Principal": {},
7       "Effect": "Allow",
8       "Action": [],
9       "Resource": []
10    }
11  ]
12 }
```

문 편집 Statement1 제거

1. 작업 추가

서비스 선택

Q S3 X

사용 가능

S3

S3 Object Lambda

S3 Outposts

2. 리소스 추가

3. 조건 추가 (선택 사항)

+ 새 문 추가

정책

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "Statement1",
6       "Principal": {},
7       "Effect": "Allow",
8       "Action": [
9         "s3:GetObject"
10      ],
11       "Resource": []
12    }
13  ]
14 }
```

문 편집 Statement1 제거

1. 작업 추가

모든 서비스 > S3

Q GetObject X

액세스 수준 - 읽기 또는 쓰기

☒ GetObject ⓘ

☐ GetObjectAcl ⓘ

☐ GetObjectAttributes ⓘ

☐ GetObjectLegalHold ⓘ

☐ GetObjectRetention ⓘ

☐ GetObjectTagging ⓘ

☐ GetObjectTorrent ⓘ

☐ GetObjectVersion ⓘ

☐ GetObjectVersionAcl ⓘ

☐ GetObjectVersionAttributes ⓘ

2. 리소스 추가 추가

3. 조건 추가 (선택 사항) 추가

+ 새 문 추가

정책

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "Statement1",
6       "Principal": {},
7       "Effect": "Allow",
8       "Action": [
9         "s3:GetObject"
10      ],
11       "Resource": []
12    }
13  ]
14 }
```

문 편집 Statement1 제거

1. 작업 추가

모든 서비스 > S3

Q GetObject X

액세스 수준 - 읽기 또는 쓰기

☒ GetObject ⓘ

☐ GetObjectAcl ⓘ

☐ GetObjectAttributes ⓘ

☐ GetObjectLegalHold ⓘ

☐ GetObjectRetention ⓘ

☐ GetObjectTagging ⓘ

☐ GetObjectTorrent ⓘ

☐ GetObjectVersion ⓘ

☐ GetObjectVersionAcl ⓘ

☐ GetObjectVersionAttributes ⓘ

2. 리소스 추가 추가

3. 조건 추가 (선택 사항) 추가

+ 새 문 추가

리소스 추가

선택한 서비스에 대해 추가할 리소스 유형과 ARN을 지정합니다.

서비스

S3

리소스 유형

object

리소스 ARN

arn:aws:s3:::jscode-uploaded-files/*

취소

리소스 추가

▶ 리소스 ARN : **arn:aws:s3:::{내가 만든 버킷명}/***의 형식으로 입력하기

▶ ARN이란?

⇒ Amazon Resource Number의 약자이다. AWS에 존재하는 리소스를 표현하는 문법이다.

정책

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "Statement1",
6       "Principal": "*",
7       "Effect": "Allow",
8       "Action": [
9         "s3:GetObject"
10      ],
11      "Resource": [
12        "arn:aws:s3:::jscode-uploaded-files/*"
13      ]
14    }
15  ]
16 }
```

+ 새 문 추가

문 편집

Statement1

제거

1. 작업 추가

서비스 선택

Q S3

포함됨

S3

사용 가능

S3 Object Lambda

S3 Outposts

2. 리소스 추가

추가

3. 조건 추가 (선택 사항)

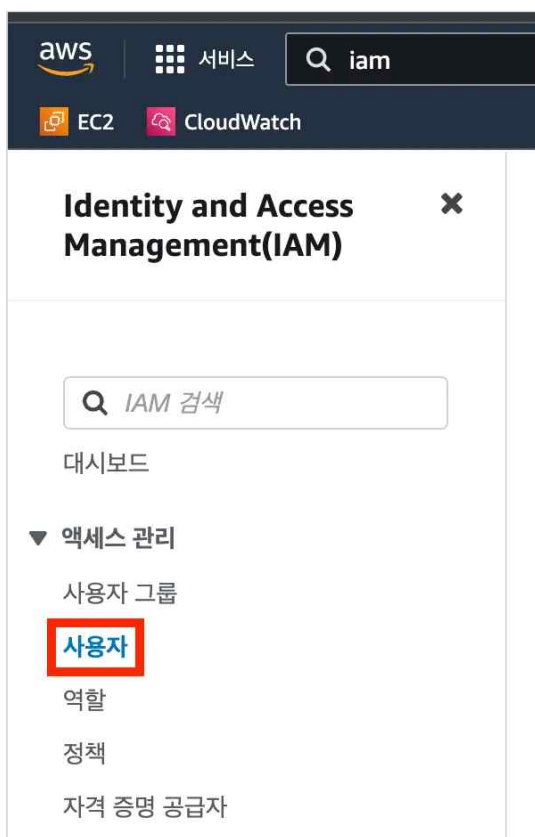
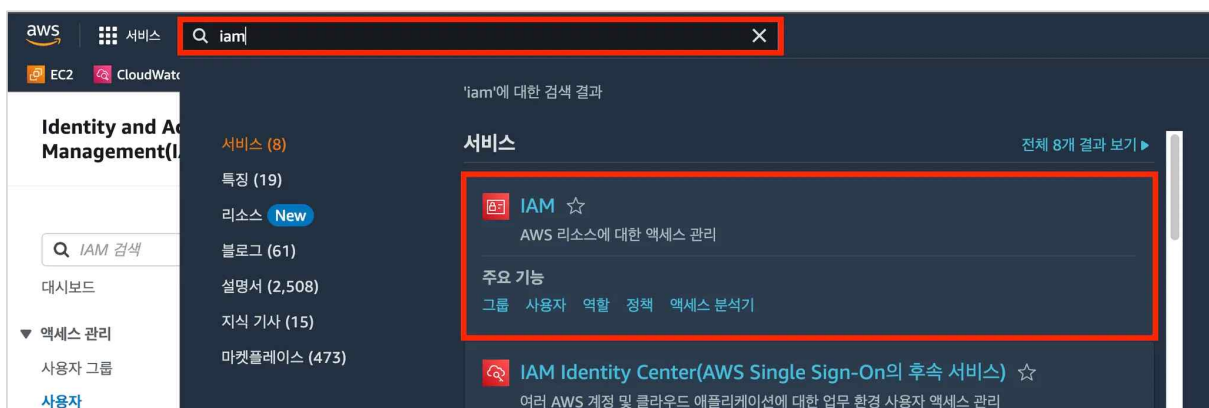
추가

[실습] 2. S3에 파일 업로드 할 수 있도록 IAM에서 액세스 키 발급받기

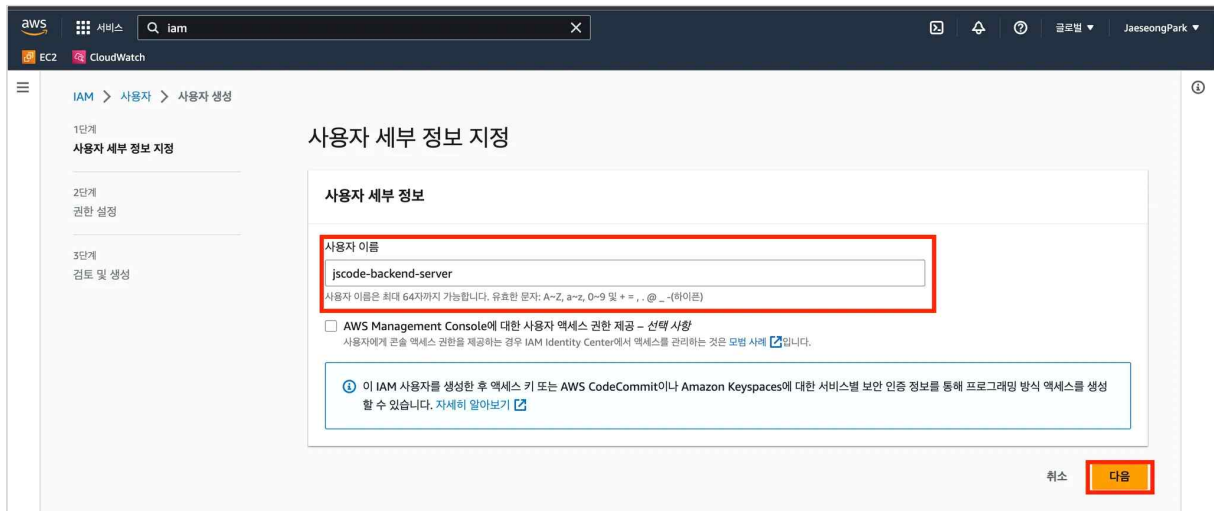
✓ 목표

기본적으로 AWS의 리소스에 아무나 접근을 못하게 막아놨기 때문에 S3에 접근해서 파일을 업로드할 수가 없다. 하지만 백엔드 서버가 S3에 접근해서 파일을 업로드할 수 있어야 한다. S3에 접근할 수 있는 권한을 받기 위해 IAM이라는 곳에서 권한을 부여받아야 한다.

✓ 1. IAM에서 '사용자' 메뉴로 들어가기



✓ 2. 사용자 추가하기



권한 설정

기존 그룹에 사용자를 추가하거나 새 그룹을 생성합니다. 직무별로 사용자의 권한을 관리하려면 그룹을 사용하는 것이 좋습니다. [자세히 알아보기](#)

권한 옵션

○ 그룹에 사용자 추가

기존 그룹에 사용자를 추가하거나 새 그룹을 생성합니다. 그룹을 사용하여 직무별로 사용자 권한을 관리하는 것이 좋습니다.

○ 권한 복사

기존 사용자의 모든 그룹 멤버십, 연결된 관리형 정책 및 인라인 정책을 복사합니다.

● 직접 정책 연결

관리형 정책을 사용자에게 직접 연결합니다. 사용자에게 연결하는 대신, 정책을 그룹에 연결한 후 사용자를 적절한 그룹에 추가하는 것이 좋습니다.

권한 정책 (1/1087)

새 사용자에게 연결할 정책을 하나 이상 선택합니다.

Search: S3Full 1 개 일치

S3Full

필터 지우기

| <input checked="" type="checkbox"/> | 정책 이름 | 유형 | 연결된 엔터티 |
|-------------------------------------|--------------------|---------|---------|
| <input checked="" type="checkbox"/> | AmazonS3FullAccess | AWS 관리형 | 1 |

▶ 권한 경계 - 선택 사항

권한 경계를 설정하여 이 사용자에게 대한 최대 권한을 제어합니다. 권한 관리를 다른 사용자에게 위임하는 데 사용되는 이 고급 기능을 사용합니다. [자세히 알아보기](#)

취소

이전

다음

검토 및 생성

선택 사항을 검토합니다. 사용자를 생성한 후 자동 생성된 암호를 보고 다운로드할 수 있습니다(활성화된 경우).

사용자 세부 정보

사용자 이름
jscode-backend-server

콘솔 암호 유형
None

암호 재설정 필요
아니요

권한 요약

< 1 >

이름 [v]

유형 [v]

다음과 같이 사용 [v]

AmazonS3FullAccess

AWS 관리형

권한 정책

태그 - 선택 사항

태그는 리소스를 식별, 구성 또는 검색하는 데 도움이 되도록 AWS 리소스에 추가할 수 있는 키 값 페어입니다. 이 사용자와 연결할 태그를 선택합니다.

리소스와 연결된 태그가 없습니다.

새 태그 추가

최대 50개의 태그를 더 추가할 수 있습니다.

취소

이전

사용자 생성

✓ 3. 액세스 키 만들기

Identity and Access Management(IAM)

사용자가 성공적으로 생성됨
AWS Management Console에 로그인하기 위한 사용자의 암호와 이메일 주소를 보고 다운로드할 수 있습니다.

사용자 보기

IAM > 사용자

사용자 (3) 정보

IAM 사용자는 계정에서 AWS와 상호 작용하는 데 사용되는 장기 자격 증명을 가진 자격 증명입니다.

사용자 이름 또는 액세스 키로 사용자 찾기

| <input type="checkbox"/> | 사용자 이름 | 그룹 | 마지막 활동 | MFA | 암호 수명 | 활성 키 수명 |
|--------------------------|-----------------------|----|--------|-----|--------|---------|
| <input type="checkbox"/> | admin | 없음 | 21일 전 | 가상 | 119일 전 | 119일 전 |
| <input type="checkbox"/> | jscode-backend-server | 없음 | 언 함 | 없음 | 없음 | - |
| <input type="checkbox"/> | s3 | 없음 | 22일 전 | 가상 | 없음 | 89일 전 |

MCA 디비전스가 어스인리 MCA 디비전으로 합쳐졌으며 AISC 하계 회의 이야 개설했다

IAM > 사용자 > jscode-backend-server > 액세스 키 만들기

1단계
액세스 키 모범 사례 및 대안

2단계 - 선택 사항
설명 태그 설정

3단계
액세스 키 검색

액세스 키 모범 사례 및 대안

보안 개선을 위해 액세스 키와 같은 장기 자격 증명을 사용하지 마세요. 다음과 같은 사용 사례와 대안을 고려하세요.

☐ **Command Line Interface(CLI)**
AWS CLI를 사용하여 AWS 계정에 액세스할 수 있도록 이 액세스 키를 사용할 것입니다.

☐ **로컬 코드**
로컬 개발 환경의 애플리케이션 코드를 사용하여 AWS 계정에 액세스할 수 있도록 이 액세스 키를 사용할 것입니다.

☐ **AWS 컴퓨팅 서비스에서 실행되는 애플리케이션**
Amazon EC2, Amazon ECS 또는 AWS Lambda와 같은 AWS 컴퓨팅 서비스에서 실행되는 애플리케이션 코드를 사용하여 AWS 계정에 액세스할 수 있도록 이 액세스 키를 사용할 것입니다.

☐ **서드 파티 서비스**
AWS 리소스를 모니터링 또는 관리하는 서드 파티 애플리케이션 또는 서비스에 액세스할 수 있도록 이 액세스 키를 사용할 것입니다.

☒ **AWS 외부에서 실행되는 애플리케이션**
애플리케이션을 온프레미스 호스트에서 실행하거나 로컬 AWS 클라이언트 또는 서드 파티 AWS 플러그 인을 사용할 수 있도록 이 액세스 키를 사용할 것입니다.

☐ **기타**
귀하의 사용 사례가 여기에 나열되어 있지 않습니다.

이 사용 사례에서는 액세스 키를 사용해도 되지만 모범 사례를 따릅니다.

- 액세스 키를 일반 텍스트, 코드 리포지토리 또는 코드로 저장해서는 안 됩니다.
- 더 이상 필요 없는 경우 액세스 키를 비활성화하거나 삭제합니다.
- 최소 권한을 활성화합니다.
- 액세스 키를 정기적으로 교체합니다.

액세스 키 관리에 대한 자세한 내용은 [AWS 액세스 키 관리 모범 사례](#)를 참조하세요.

취소 다음

IAM > 사용자 > jscode-backend-server > 액세스 키 만들기

1단계
액세스 키 모범 사례 및 대안

2단계 - 선택 사항
설명 태그 설정

3단계
액세스 키 검색

설명 태그 설정 - 선택 사항

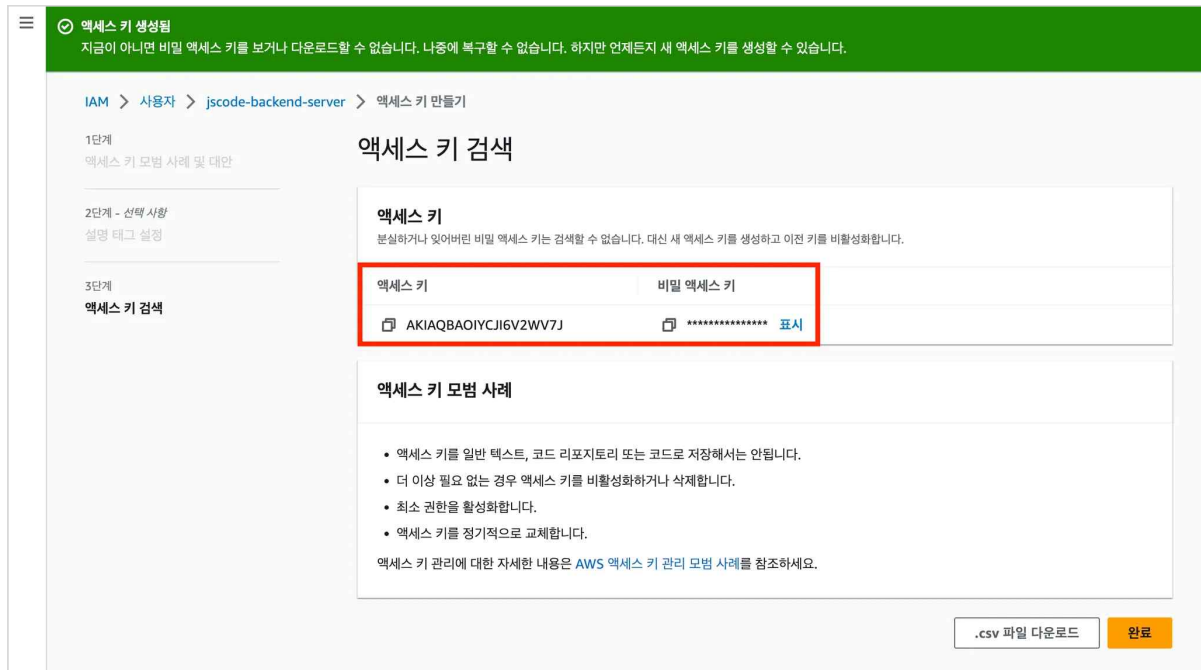
이 액세스 키에 대한 설명은 이 사용자에게 태그로 연결되고, 액세스 키와 함께 표시됩니다.

설명 태그 값

이 액세스 키의 용도와 사용 위치를 설명합니다. 좋은 설명은 나중에 이 액세스 키를 자신있게 교체하는 데 유용합니다.

최대 256자까지 가능합니다. 허용되는 문자는 문자, 숫자, UTF-8로 표현할 수 있는 공백 및 _ . : / = + - @입니다.

취소 이전 **액세스 키 만들기**



[실습] 3. S3를 활용해 Express 서버에 이미지 업로드 기능 구현하기

✓ 1. EC2 인스턴스에서 Express 서버 clone하기

```
$ git clone https://github.com/JSCODE-EDU/s3-express-sample.git
$ cd s3-express-sample
$ npm i
```

✓ 2. .env 파일 수정하기

내가 발급받은 액세스 키와 버킷명에 맞게 .env 파일을 수정하면 된다.

.env

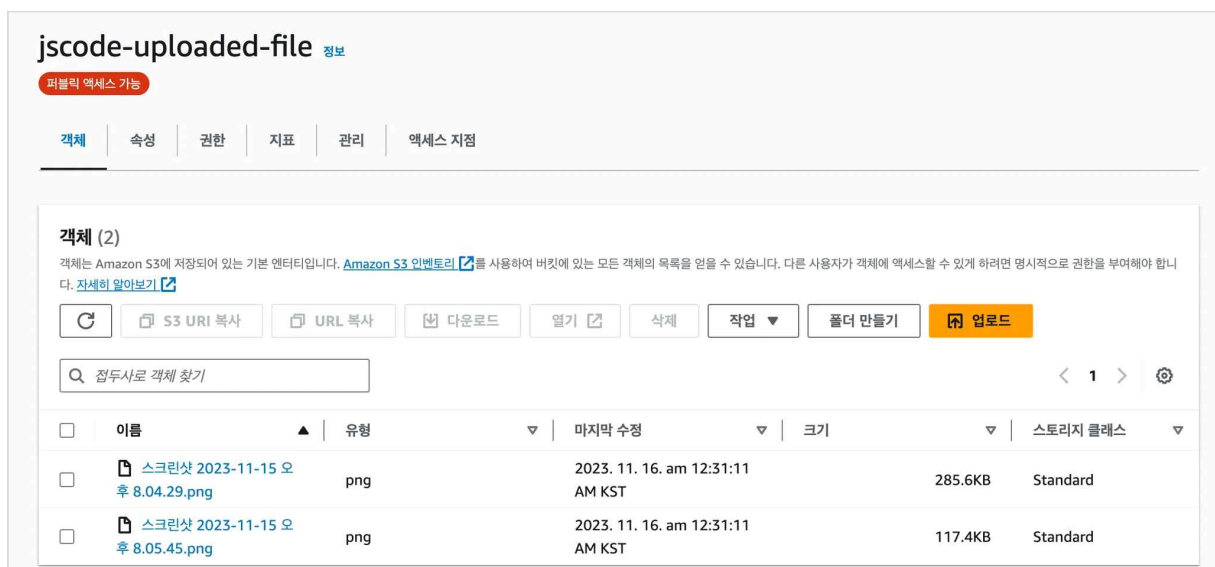
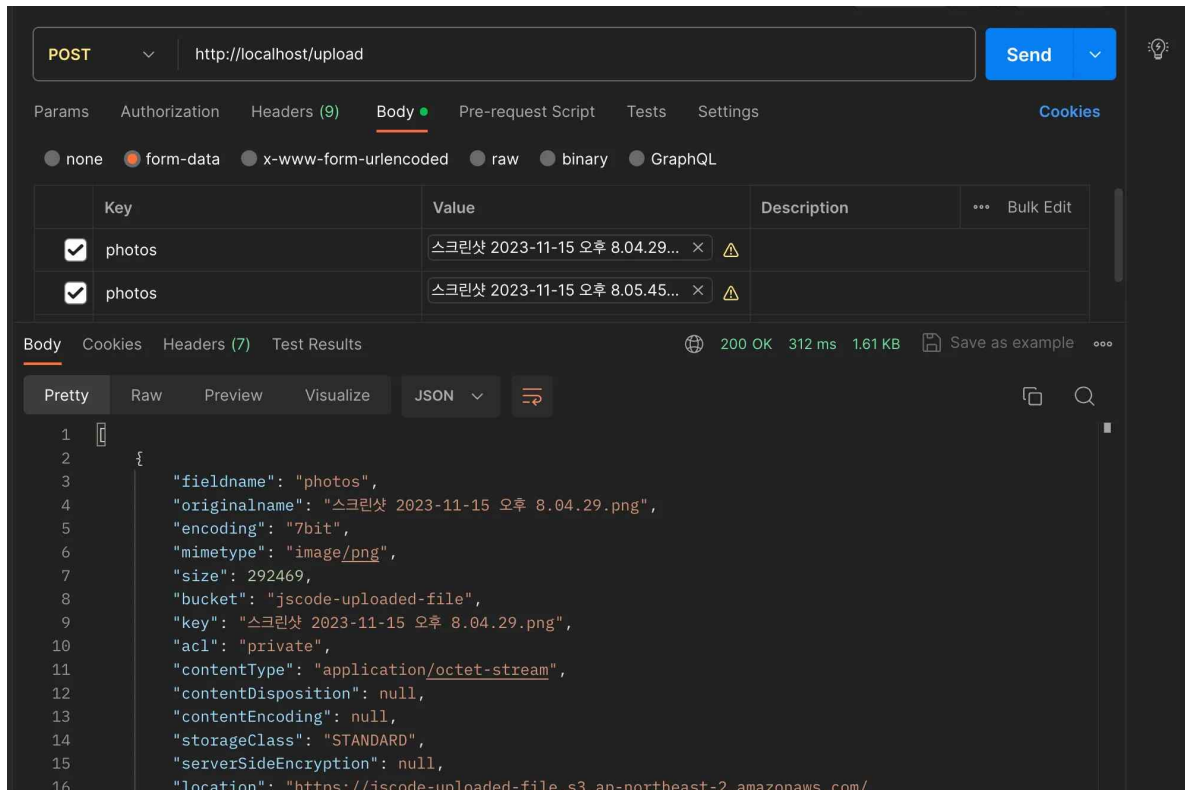
```
AWS_ACCESS_KEY=____
AWS_SECRET_ACCESS_KEY=____
AWS_S3_BUCKET=____
```

주의) 개인의 .env 파일은 반드시 .gitignore에 추가해서 Github에 올라가지 않게 해야 합니다.

(위 Github Repository는 편의상 .env를 Github에 올린 것 뿐입니다.)

✓ 3. S3에 파일 업로드가 잘 되는 지 확인하기

▶ 포스트맨 사용



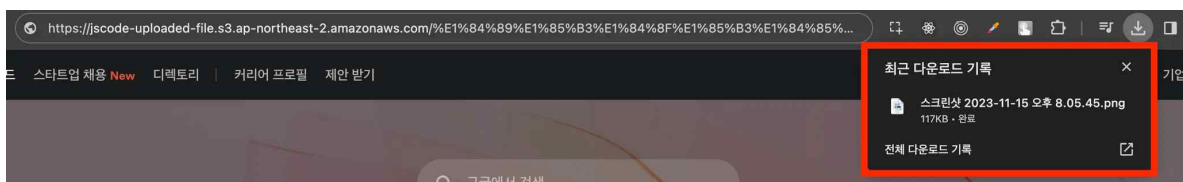
✓ 4. S3에 저장된 URL로 이미지 잘 조회되는 지 확인

방법 1) HTML에서 확인하기

index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
  <h1>이미지 테스트</h1>
  
</body>
</html>
```

방법 2) 주소창에 주소 쳐보기



[제거] 비용 나가지 않게 S3 깔끔하게 종료하기

✔ 비용 나가지 않게 S3 깔끔하게 종료하기

1. 버킷 안에 있는 객체 전부 삭제하기

jscode-uploaded-file 정보

퍼블릭 액세스 가능

객체 | 속성 | 권한 | 지표 | 관리 | 액세스 지점

객체 (2)

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](#)

🔄 S3 URI 복사 URL 복사 다운로드 열기 **삭제** 작업 폴더 만들기 업로드

🔍 접두사로 객체 찾기

| <input checked="" type="checkbox"/> | 이름 | 유형 | 마지막 수정 | 크기 | 스토리지 클래스 |
|-------------------------------------|--------------------------------|-----|----------------------------------|---------|----------|
| <input checked="" type="checkbox"/> | 스크린샷 2023-11-15 오후 8.04.29.png | png | 2023. 11. 16. am 12:31:11 AM KST | 285.6KB | Standard |
| <input checked="" type="checkbox"/> | 스크린샷 2023-11-15 오후 8.05.45.png | png | 2023. 11. 16. am 12:31:11 AM KST | 117.4KB | Standard |

2. 버킷 삭제하기

버킷 (1) 정보

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

🔄 ARN 복사 비어 있음 **삭제** 버킷 만들기

🔍 이름으로 버킷 찾기

| 이름 | AWS 리전 | 액세스 | 생성 날짜 |
|----------------------|----------------------------|-----|---------------------------------|
| jscode-uploaded-file | 아시아 태평양(서울) ap-northeast-2 | 퍼블릭 | 2023. 11. 15. pm 9:58:44 PM KST |