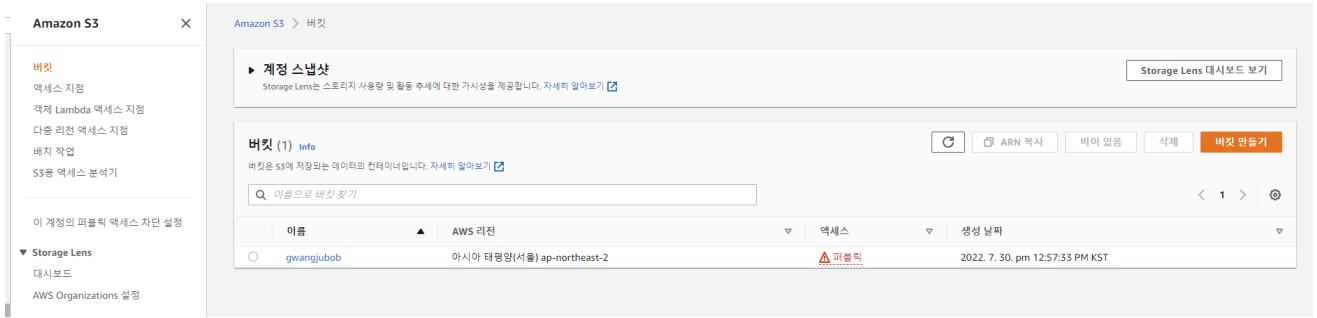


AWS S3, REACT 이미지 업로드

 Created By	
 Stakeholders	
 Status	
 Type	
 Created	@2022년 8월 17일 오후 2:27
 Last Edited Time	@2022년 8월 17일 오후 5:04
 Last Edited By	

AWS S3 생성


- AWS 가입 절차 진행 (처음 가입시 1년 Free-Tier)
 - Free-Tier 월 별 용량 (이후 과금)
 - get 요청 2만건
 - put, copy, post 요청 2천건
 - 저장용량 5GB
- AWS S3 서비스 접속 후 버킷 만들기



- 권한 및 정책 설정

권한 개요

액세스


 퍼블릭

퍼블릭 액세스 차단(버킷 설정)

퍼블릭 액세스는 ACL(액세스 제어 목록), 버킷 정책, 액세스 지정 정책 또는 모두를 통해 버킷 및 객체에 부여됩니다. 모든 S3 버킷 및 객체에 대한 퍼블릭 액세스가 차단되었는지 확인하려면 세스 차단]을 활성화하도록 권장하지만, 이 설정을 적용하기 전에 퍼블릭 액세스가 없어도 애플리케이션이 올바르게 작동하는지 확인합니다. 버킷 또는 내부 객체에 어느 정도 수준의 퍼블릭

편집

모든 퍼블릭 액세스 차단

 비활성

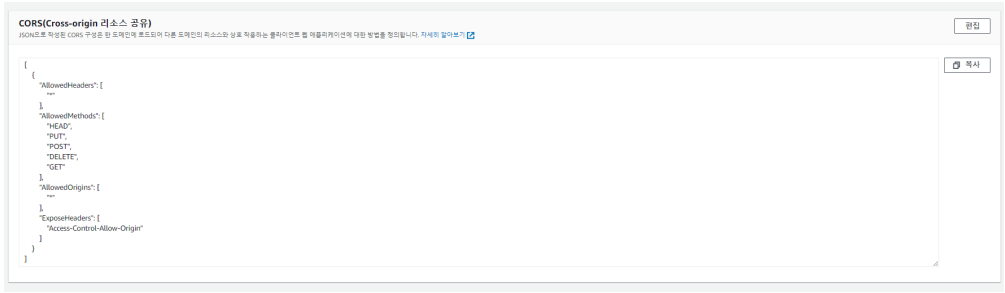
▶ 이 버킷의 개별 퍼블릭 액세스 차단 설정

버킷 정책

JSON으로 작성된 버킷 정책은 버킷에 저장된 객체에 대한 액세스 권한을 제공합니다. 버킷 정책은 다른 계정이 소유한 객체에는 적용되지 않습니다. 자세히 알아보기

```
{
  "Version": "2012-10-17",
  "Id": "Policy1659153874021",
  "Statement": [
    {
      "Sid": "Stmt1659153865950",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::gwangjubob/*"
    }
  ]
}
```

- CORS 정책 설정



- 버킷 생성 후 업로드를 눌러서 파일이 잘 올라가면 성공



REACT에서 S3 사용하기

- AWS 내 계정 > 보안 자격 증명 > 액세스 키 발급 (엑세스 키 2개 발급! 저장해두기!)



- React에서 aws-sdk 설치 (AWS - Software Development Kit)

```
npm install aws-sdk
```

- .env에 발급받은 액세스키 설정

- AWS에 업로드하는 함수 추가 (AWS.S3.ManageUpload)

```
import AWS from "aws-sdk";
import { v1 } from "uuid";

// region과 발급 받은 키로 설정
AWS.config.update({
  region: "ap-northeast-2",
  accessKeyId: process.env.REACT_APP_AWS_ACCESS_KEY,
  secretAccessKey: process.env.REACT_APP_AWS_SECRET_KEY
});

// 이미지 파일을 인자로 받아 업로드 하는 함수
export const uploadFile = (file: File) => {
  const upload = new AWS.S3.ManagedUpload({
    params: {
      Bucket: "gwangjubob", // 올릴 버킷 이름
      Key: `${v1()}.png`, // key는 파일 이름, 고유값을 넣기 위해 uuid 사용
      Body: file // 이미지 파일
    }
  });
};

const promise = upload.promise();

return promise.then(res => {
  return res.Location; // 업로드 후 이미지 주소를 반환 (url 형식)
});
};
```

```
// url을 인자로 받아 이미지 파일 삭제하는 함수
export const deleteFile = (url: string) => {
  const fileUrl = url.split("amazonaws.com/").pop();
  const del = new AWS.S3();
  del.deleteObject(
    {
      Bucket: "gwangjubob",
      Key: fileUrl as string
    }
  );
  // function (err, data) {
  //   if (err) console.log(err, err.stack);
  //   else console.log(data);
  // }
};

export default {};
```