S3에 이미지 저장하기

Tags BackEnd

목차

---

1. AWS S3(Simple Storage Service)fks?

- 파일 서버의 역할을 하는 서비스
- 일반적인 파일 서버는 트래픽이 증가함에 따라서 장비를 증설하는 작업을 해야하는데 S3는 이와 같은 것을 대행해줌

  → 트래픽에 따른 시스템적인 문제 걱정X

용어

- 객체 (Object) : S3에 저장된 데이터 하나하나
- 버킷 (Bucket) : 연관된 객체들을 그룹핑한 최상위 디렉토리.

  버킷 단위로 지역 지정 가능

  버킷에 포함된 모든 객체에 대해서 일괄적으로 인증과 접속 제한 가능

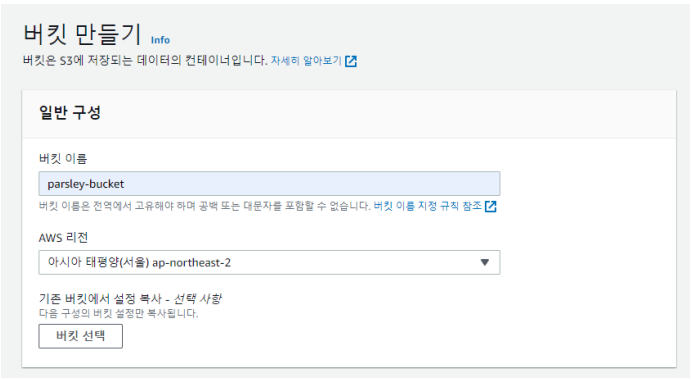
초보자도 이해할 수 있는 S3(Simple Storage Service) | DevelopersIO

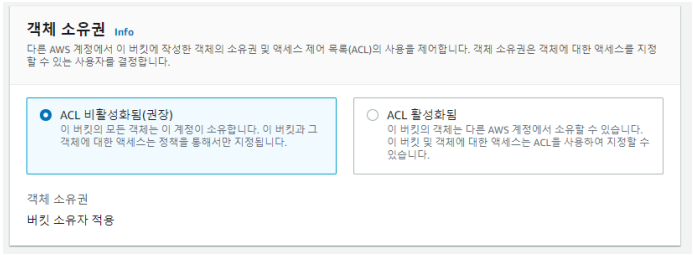
S3(Simple Storage Service)의 개념과 특징에 대해 정리해보았습니다. 안녕하세요 클래스메소드 김재욱(Kim Jaewook)이라고 합니다. 이번에는「초보자도 이해할 수 있는 S3(Simple Storage Service)」라는 주제로 AWS S3 서비스에 대한 개념과 버킷 생성에 대해서 다뤄볼려고 합니다. Simple Storage Service의 약자로 파일 서버의 역할을 하는 서비스다. 일반적인 파일서버는 트래픽이 증가함에 따라서 장비를 증설하

https://dev.classmethod.jp/articles/for-beginner-s3-explanation/

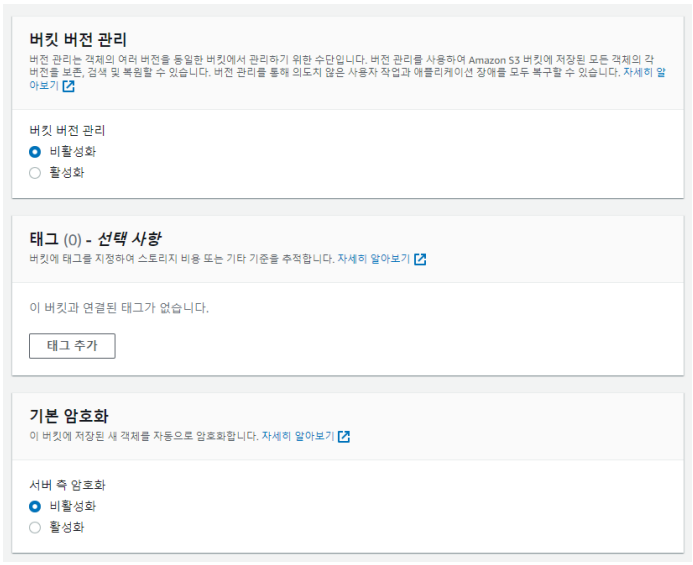
2. Bucket 생성

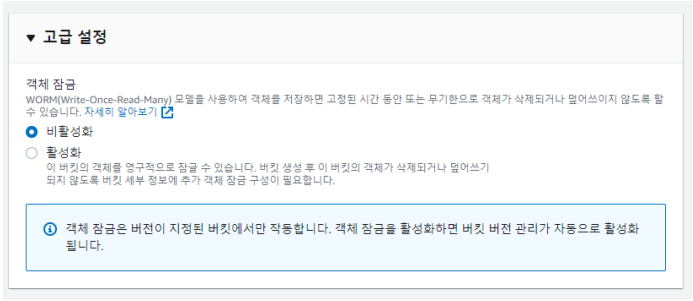
▼ 버킷 만들기

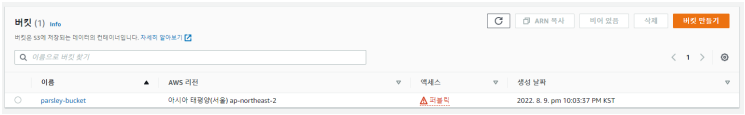



객체 소유권 Info
다른 AWS 계정에서 이 버킷에 작성한 객체의 소유권 및 액세스 제어 목록(ACL)의 사용을 제어합니다. 객체 소유권은 객체에 대한 액세스를 지정할 수 있는 사용자를 결정합니다.
ACL 비활성화됨(권장)
이 버킷의 모든 객체는 이 계정이 소유합니다. 이 버킷과 그 객체에 대한 액세스는 정책을 통해서만 지정됩니다.
ACL 활성화됨
이 버킷의 객체는 다른 AWS 계정에서 소유할 수 있습니다. 이 버킷 및 객체에 대한 액세스는 ACL을 사용하여 지정할 수 있습니다.
객체 소유권
버킷 소유자 적용

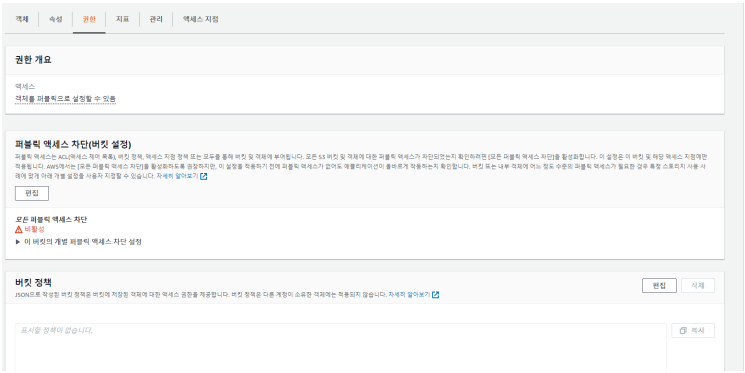

이 버킷의 퍼블릭 액세스 차단 설정
퍼블릭 액세스는 ACL(액세스 제어 목록), 버킷 정책, 액세스 지점 정책 또는 모두를 통해 버킷 및 객체에 부여됩니다. 이 버킷 및 해당 객체에 대한 퍼블릭 액세스가 차단되었는지 확인하려면 모든 퍼블릭 액세스 차단을 활성화합니다. 이 설정은 이 버킷 및 해당 액세스 지점에만 적용됩니다. AWS에서는 모든 퍼블릭 액세스 차단을 활성화하도록 권장하지만, 이 설정을 적용하기 전에 퍼블릭 액세스가 없어도 애플리케이션이 올바르게 작동하는지 확인합니다. 이 버킷 또는 내부 객체에 대한 어느 정도 수준의 퍼블릭 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.
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.


버킷 버전 관리
버전 관리는 객체의 여러 버전을 동일한 버킷에서 관리하기 위한 수단입니다. 버전 관리를 사용하여 Amazon S3 버킷에 저장된 모든 객체의 각 버전을 보존, 검색 및 복원할 수 있습니다. 버전 관리를 통해 의도치 않은 사용자 작업과 애플리케이션 장애를 모두 복구할 수 있습니다. 자세히 알아보기
버킷 버전 관리
비활성화
활성화
태그 (0) - 선택 사항
버킷에 태그를 지정하여 스토리지 비용 또는 기타 기준을 추적합니다. 자세히 알아보기
이 버킷과 연결된 태그가 없습니다.
태그 추가
기본 암호화
이 버킷에 저장된 새 객체를 자동으로 암호화합니다. 자세히 알아보기
서버 측 암호화
비활성화
활성화

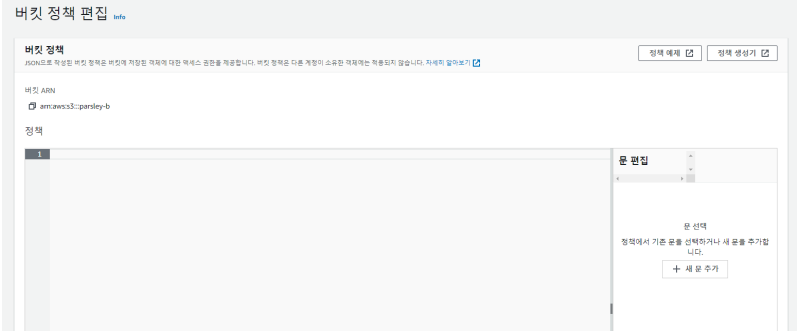

고급 설정
객체 잠금
WORM(Write-Once-Read-Many) 모델을 사용하여 객체를 저장하면 고정된 시간 동안 또는 무기한으로 객체가 삭제되거나 덮어쓰이지 않도록 할 수 있습니다. 자세히 알아보기
비활성화
활성화
이 버킷의 객체를 영구적으로 잠글 수 있습니다. 버킷 생성 후 이 버킷의 객체가 삭제되거나 덮어쓰기되지 않도록 버킷 세부 정보에 추가 객체 잠금 구성이 필요합니다.
객체 잠금은 버전이 지정된 버킷에서만 작동합니다. 객체 잠금을 활성화하면 버킷 버전 관리가 자동으로 활성화됩니다.



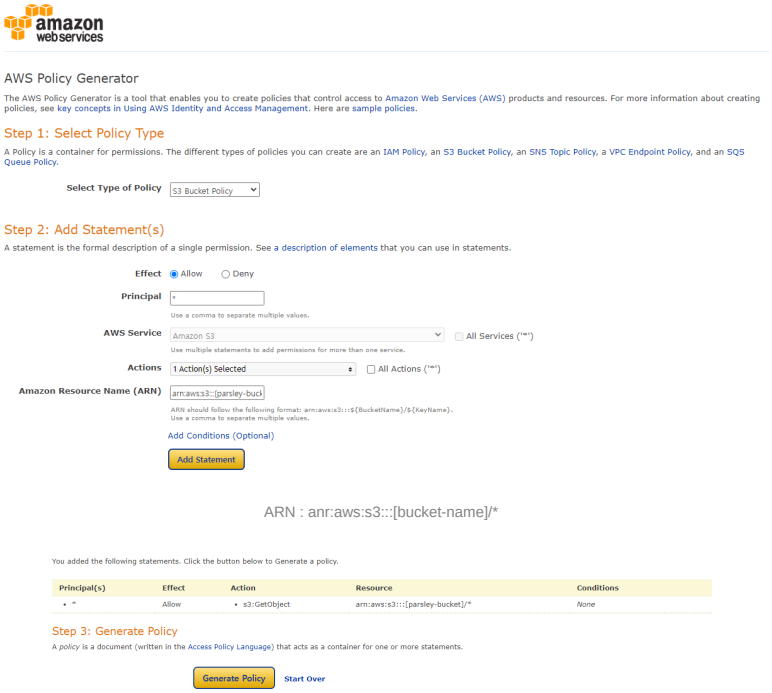
▼ 버킷 오류 해결



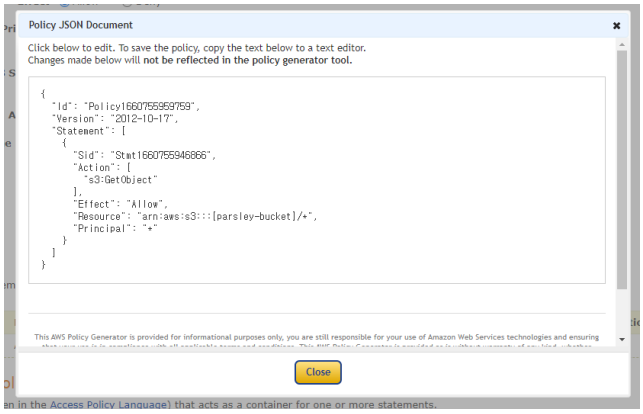
버킷 정책 > 편집 클릭

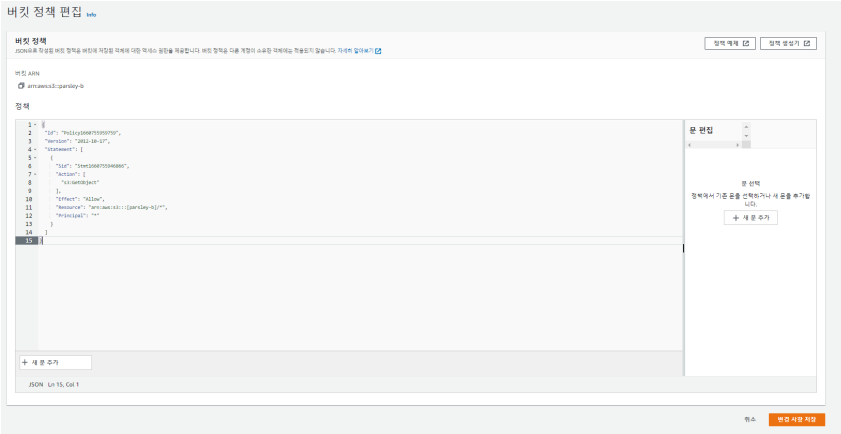


정책 생성기 클릭



ARN : anr:aws:s3:::[bucket-name]/*

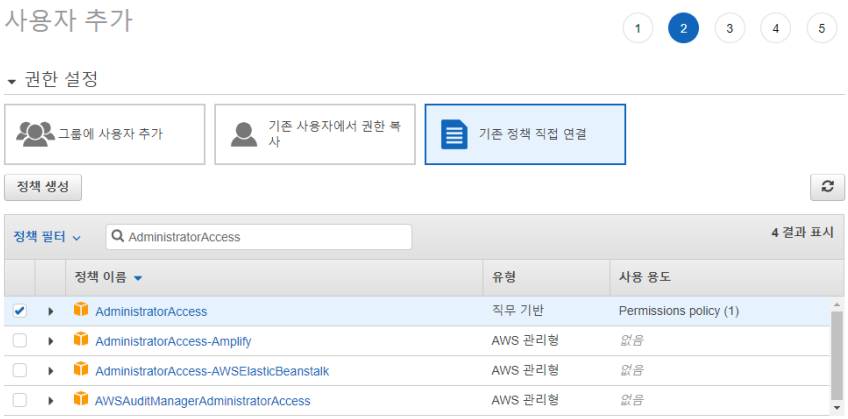




생성한 버킷 정책 붙여넣기 후 저장

3. IAM 등록

▼ 사용자 추가



AdministratorAccess : 관리자와 동일한 권한을 가짐

키 ID와 비밀 액세스 키 저장 필수

4. SpringBoot 연동

- application.properties

```
# Amazon S3 Account Credentials
cloud.aws.credentials.access-key=[access-key-id]
cloud.aws.credentials.secret-key=[secret-access-key]

# Amazon S3 bucket Info
cloud.aws.s3.bucket=[bucket-name]
cloud.aws.region.static=[region-name]
cloud.aws.stack.auto=false

# AWS S3 Bucket URL
cloud.aws.s3.bucket.url=https://s3.ap-northeast-2.amazonaws.com/[bucket-name]

# file size
spring.servlet.multipart.max-file-size=20MB
spring.servlet.multipart.max-request-size=20MB

#logging.level.com.amazonaws.util.EC2MetadataUtils: error
```

- Application.java

```
@SpringBootApplication
public class GroupCallApplication {

    static{
        System.setProperty("com.amazonaws.sdk.disableEc2Metadata", "true");
    }

}
```

- AwsS3Config.java

```
package com.ssafy.config;

import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class AwsS3Config {

    @Value("${cloud.aws.credentials.access-key}")
    private String accessKey;
    @Value("${cloud.aws.credentials.secret-key}")
    private String secretKey;
    @Value("${cloud.aws.region.static}")
    private String region;

    @Bean
    public AmazonS3Client amazonS3Client(){
        BasicAWSCredentials awsCredentials = new BasicAWSCredentials(accessKey, secretKey);
        return (AmazonS3Client) AmazonS3ClientBuilder.standard().withRegion(region).withCredentials(new AWSStaticCredentialsProvider(awsCredentials)).build();
    }
}
```

- build.gradle

```
implementation 'org.springframework.cloud:spring-cloud-starter-aws:2.2.6.RELEASE'
```