



# S3에 이미지 저장하기

Tags

BackEnd

## 목차

1. AWS S3(Simple Storage Service)란?  
용어
2. Bucket 생성
3. IAM 등록
4. SpringBoot 연동

## 1. AWS S3(Simple Storage Service)란?

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



**Amazon S3**  
Simple Storage Service

## 2. Bucket 생성

### ▼ 버킷 만들기

### 버킷 만들기

Info

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

#### 일반 구성

버킷 이름

parsley-bucket

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

AWS 리전

아시아 태평양(서울) ap-northeast-2

기본 버킷에서 설정 복사 - 선택 사항

다음 구성의 버킷 설정만 복사됩니다.

버킷 선택

## 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.

버킷 소유자 적용

### 이 버킷의 퍼블릭 액세스 차단 설정

파블릭 액세스는 ACL에서 모든 공개 버킷 지정 또는 모든 항목에 대해 버킷 및 객체에 부여하는 권한을 나타내며, 퍼블릭 액세스가 확인되면 모든 퍼블릭 액세스 작업을 실행할 수 있습니다. 이 설정은 이 버킷 및 해당 객체로 대한 AWS에서는 모든 퍼블릭 액세스 작업을 실행하도록 강제하지만, 이 설정을 적용하기 전에는 퍼블릭 액세스가 없어도 애플리케이션이 올바르게 작동하며 실행됩니다. 이 버킷 또는 내비 게이션에 대한 어느 정도 수준의 퍼블릭 액세스가 필요한 경우 특정 스토리지 사용 사례에 맞게 아래 개괄 설정을 사용하여 지칭할 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 자단합니다. 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

사전 관리는 객체의 여러 버전을 동일한 버킷에서 관리하기 위한 수단입니다. 사전 관리를 사용하여 Amazon S3 버킷에 저장된 모든 객체의 각 버전을 보존, 검색 및 복원할 수 있습니다. 사전 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 [\[2\]](#)

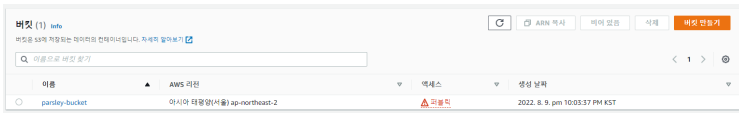
### ● 비활성화

○ **활성화**

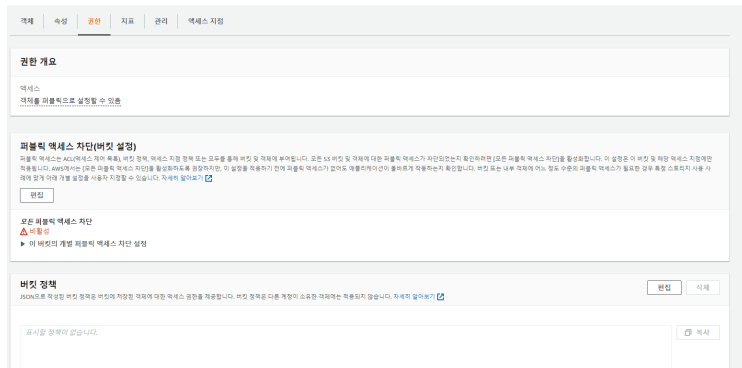
이 버킷의

이 버킷의 객체를 영구적으로 잠글 수 있습니다. 버킷 생성 후 이 버킷의 객체가 삭제되거나 덮어쓰기 되지 않도록 버킷 세부 정보에 추가 객체 잠금 구성이 필요합니다.

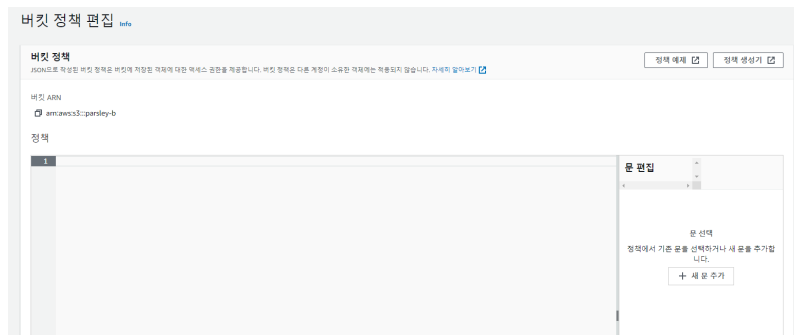
① 객체 잠금은 버전이 지정된 버킷에서만 작동합니다. 객체 잠금을 활성화하면 버킷 버전 관리가 자동으로 활성화됩니다.



## ▼ 버킷 오류 해결



버킷 정책 > 편집 클릭



정책 생성기 클릭



## AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see [key concepts in Using AWS Identity and Access Management](#). Here are sample policies.

### Step 1: Select Policy Type

A policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy S3 Bucket Policy

### Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a [description of elements](#) that you can use in statements.

Effect ☒ Allow ☐ Deny

Principal

Use a comma to separate multiple values.

AWS Service Amazon S3 ☐ All Services ("\*")

Use multiple statements to add permissions for more than one service.

Actions 1 Action(s) Selected ☐ All Actions ("\*")

Amazon Resource Name (ARN) arn:aws:s3:::parley-buck

ARNs should follow the following format: arn:aws:s3:::[bucket-name]/[key-name].  
Use a comma to separate multiple values.

Add Conditions (Optional)

Add Statement

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

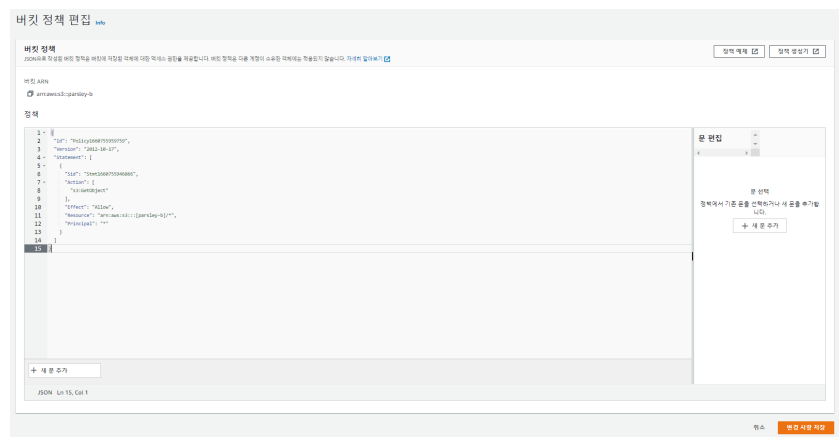
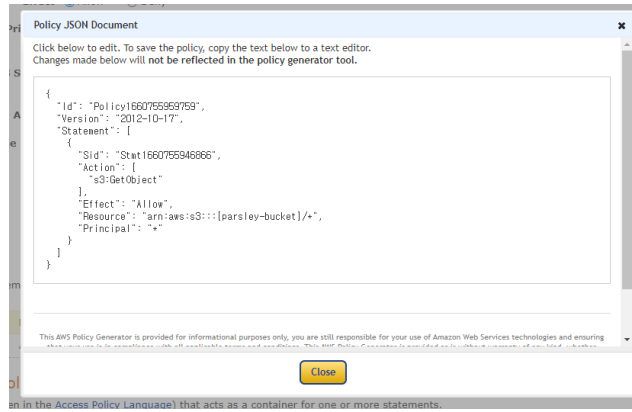
You added the following statements. Click the button below to Generate a policy.

Principal(s)	Effect	Action	Resource	Conditions
* *	Allow	s3:GetObject	arn:aws:s3:::[parley-bucket]/*	None

### Step 3: Generate Policy

A policy is a document (written in the [Access Policy Language](#)) that acts as a container for one or more statements.

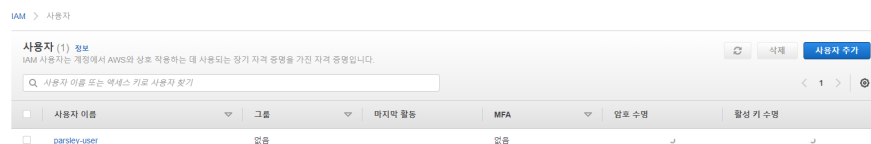
Generate Policy Start Over



생성한 버킷 정책 붙여넣기 후 저장

### 3. IAM 등록

#### ▼ 사용자 추가



### 사용자 추가

1 2 3 4 5

#### 사용자 세부 정보 설정

동일한 액세스 유형 및 권한을 사용하여 한 번에 여러 사용자를 추가할 수 있습니다. [자세히 알아보기](#)

사용자 이름\*

[다른 사용자 추가](#)

#### AWS 액세스 유형 선택

이러한 사용자가 주로 AWS에 액세스하는 방법을 선택합니다. 프로그래밍 방식의 액세스만 선택하면 사용자가 위임된 역할을 사용하여 콘솔에 액세스하는 것을 방지할 수 없습니다. 액세스 키와 자동 생성된 암호가 마지막 단계에서 제공됩니다. [자세히 알아보기](#)

- AWS 자격 증명 유형 선택\*** ☒ **액세스 키 - 프로그래밍 방식 액세스**  
AWS API, CLI, SDK 및 기타 개발 도구에 대해 액세스 키 ID 및 비밀 액세스 키 을(를) 활성화합니다.
- ☐ **암호 - AWS 관리 콘솔 액세스**  
사용자가 AWS Management Console에 로그인할 수 있도록 허용하는 비밀번호 을(를) 활성화합니다.

## 사용자 추가

1 2 3 4 5

### 권한 설정

그룹에 사용자 추가

기존 사용자에서 권한 복사

기존 정책 직접 연결

정책 생성

정책 필터

AdministratorAccess

4 결과 표시

	정책 이름	유형	사용 용도
<input checked="" type="checkbox"/>	AdministratorAccess	직무 기반	Permissions policy (1)
<input type="checkbox"/>	AdministratorAccess-Amplify	AWS 관리형	없음
<input type="checkbox"/>	AdministratorAccess-AWSElasticBeanstalk	AWS 관리형	없음
<input type="checkbox"/>	AWSAuditManagerAdministratorAccess	AWS 관리형	없음

### 권한 경계 설정

AdministratorAccess : 관리자와 동일한 권한을 가짐

## 사용자 추가

1 2 3 4 5

**성공**

아래에 표시된 사용자를 생성했습니다. 사용자 보안 자격 증명을 보고 다운로드할 수 있습니다. AWS Management Console 로그인을 위한 사용자 지침을 이메일로 보낼 수도 있습니다. 지금이 이 자격 증명을 다운로드할 수 있는 마지막 기회입니다. 하지만 언제든지 새 자격 증명을 생성할 수 있습니다.

AWS Management Console 액세스 권한이 있는 사용자가 <https://818568719421.signin.aws.amazon.com/console>에 로그인할 수 있습니다.

.csv 다운로드

	사용자	액세스 키 ID	비밀 액세스 키
<input checked="" type="checkbox"/>	parsley-admin	[REDACTED]	***** 표시

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'

```