



외부 서비스 정보

Amazon S3

- 어디서나 원하는 양의 데이터를 저장하고 검색할 수 있도록 구축된 객체 스토리지
- 내구성, 가용성, 성능, 보안 및 거의 무제한의 확장성을 아주 저렴한 요금으로 제공하는 단순한 스토리지 서비스

S3의 장점

- 모든 종류의 데이터를 원하는 형식으로 저장
- 저장할 수 있는 데이터의 전체 볼륨과 객체 수에는 제한이 없음
- Amazon S3는 간단한 key 기반의 객체 스토리지이며, 데이터를 저장 및 검색하는데 사용할 수 있는 고유한 객체 키를 할당.
- Amazon S3는 간편한 표준 기반 REST 웹 서비스 인터페이스를 제공

AWS 리전

- Amazon S3 버킷을 생성할 때 AWS 리전을 지정
- S3 Standard, S3 Standard-IA 및 S3 Glacier 스토리지 클래스의 경우, 객체는 AWS 리전 내에 각기 지리적으로 분리된 최소 3개의 가용 영역에 걸쳐 여러 디바이스에 자동 저장

1. 준비 작업

- AWS 계정의 Access key 와 Secret key
 - IAM 계정을 생성하여 S3에 대한 role 부여 후, 진행하길 권장
- S3 버킷 생성
 - 권한을 수정하지 않고 파일을 업로드하면, 퍼블릭 액세스가 차단된 상태이기 때문에 아래의 403 에러가 발생
 - 퍼블릭 액세스 차단 해제를 하고, 버킷 정책을 설정하는 방법을 사용
 - 외부에서 파일을 접근할 수 있도록 퍼블릭 액세스 차단을 비활성화
 - 버킷 정책 작성

```
{
  "Version": "2012-10-17",
  "Id": "Policy1577077078140",
  "Statement": [
    {
      "Sid": "Stmt1577076944244",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::{버킷명}/*"
    }
  ]
}
```

2. 환경설정

- build.gradle

```
dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
    implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
    implementation 'org.springframework.boot:spring-boot-starter-web'
    runtimeOnly 'mysql:mysql-connector-java'
    compileOnly 'org.projectlombok:lombok'
    annotationProcessor 'org.projectlombok:lombok'

    // AWS S3
    compile group: 'org.springframework.cloud', name: 'spring-cloud-aws', version: '2.2.1.RELEASE', ext: 'pom'

    testImplementation('org.springframework.boot:spring-boot-starter-test') {
        exclude group: 'org.junit.vintage', module: 'junit-vintage-engine'
    }
}
```

3. AWS 설정

- aws 설정은 application.properties 파일에 작성

```
# aws bucket
cloud.aws.credentials.accessKey={YOUR_USER_ACCESSKEY}
cloud.aws.credentials.secretKey={YOUR_USER_SECRETKEY}
cloud.aws.s3.bucket=s3-bucket-ven
cloud.aws.region.static=ap-northeast-2
cloud.aws.stack.auto=false
```

- region 설정 : 서울은 ap-northeast-2
- 과금의 문제가 있으니 꼭!!!!!!!!!!!!!!!!!!!!!! .gitignore 처리하기!!!!!!!!!!!!!!!!!!!!!!

4. 관련 구현 코드는 git 참고

CloudFront

- CloudFront는 정적, 동적 콘텐츠를 빠르게 응답하기 위한 캐시 기능을 제공하는 CDN 서비스
- 캐싱을 지원하기 때문에 S3에 저장된 콘텐츠를 직접 접근하지 않아도 됨.
- S3와 함께 사용하는 것을 권장(더 빠른 응답과 낮은 가격)
- 배포 생성 후 활성화가 되면 생성한 버킷과 연동한 후 CloudFront 도메인으로 S3 객체 접근
 - 도메인.net/파일명 로 파일을 다운로드하고 접근할 수 있음