|  |
| --- |
|  |





|  |
| --- |
| **MoaaMoaa** |
| **포팅 메뉴얼** |
|  |
|  |
|  |
|  |
|  |

목차

[I. 개요 2](#_Toc127457362)

[1. 프로젝트 개요 2](#_Toc127457363)

[2. 프로젝트 사용 도구 2](#_Toc127457364)

[3. 개발환경 2](#_Toc127457365)

[4. 외부 서비스 2](#_Toc127457366)

[5. Gitgnore 처리한 핵심 키들 2](#_Toc127457367)

[II. 빌드 3](#_Toc127457368)

[1. 환경변수 형태 3](#_Toc127457369)

[2. 빌드하기 4](#_Toc127457370)

[3. 배포하기 4](#_Toc127457371)

[4. 서비스 이용 방법(외부) 9](#_Toc127457372)

[가) Amazon S3 Bucket 9](#_Toc127457373)

# 개요

## 프로젝트 개요

개발자의 프로젝트 및 스터디 팀 빌딩을 도와주는 웹사이트

## 프로젝트 사용 도구

이슈 관리 : JIRA

형상 관리 : Gitlab

커뮤니케이션 : Notion, Kakaotalk, Mattermost

디자인 : Figma

UCC : Filmora

CI/CD : Jenkins, Docker

## 개발환경

VS Code : 1.64.2,

IntelliJ : 22.3.1

JVM : 11.0.17

Node.js : 16.16.0

React: 17버전

SERVER : AWS EC2 Ubuntu 20.04 LTS

DB : MySQL, Redis

## 외부 서비스

Redis cloud : application.yml에 해당 내용 있음

S3 Bucket: application.yml, S3Config.yml에 해당 내용 있음

## Gitgnore 처리한 핵심 키들

React : /node\_modules, /build

Spring : gradlew, gradle/wrapper(develop에서만)



# 빌드

## 환경변수 형태

.env

REACT\_APP\_SERVER\_HOST="http://localhost:8000"

REACT\_APP\_CLIENT\_HOST="http://localhost:3000"

NODE\_PATH = src

.application.yml

배포에선 mysql localhost->mysql, redis localhost->myredis,

redis password->d5a706b2e7a4ed4b

spring:  
 datasource:  
 driver-class-name: com.mysql.cj.jdbc.Driver  
 *# Mysql Settings* url: jdbc:mysql://localhost:3306/moamoa?serverTimezone=UTC&characterEncoding=UTF-8  
 username: moamoa  
 password: '0103'  
 mvc:  
 pathmatch:  
 matching-strategy: *ant\_path\_matcher* jpa:  
 database-platform: org.hibernate.dialect.MySQL5InnoDBDialect  
 open-in-view: false  
 show-sql: true  
 hibernate:  
 ddl-auto: none  
 use-new-id-generator-mappings: false  
 properties:  
 hibernate:  
 *# format\_sql: true* redis:  
 host: localhost  
 port: 6379  
 password: '0103'  
 mail:  
 host: smtp.gmail.com  
 port: 587  
 username: moaamoaaofficial@gmail.com  
 password: imxwfakvkirblhjc  
 properties:  
 mail:  
 smtp:  
 timeout: 5000  
 auth: true  
 starttls:  
 enable: true  
 thymeleaf:  
 prefix: classpath:mail-templates/  
  
logging:  
 level:  
 org:  
 hibernate:  
 SQL: DEBUG  
 type:  
 descriptor:  
 sql:  
 BasicBinder: TRACE  
 com.ssafy.moamoa: debug  
  
security:  
 jwt:  
 secret-key: 6ec2601c-5280-4761-810b-80a5478ec7ed  
 access-token-expire: 1800000  
 *# access-token-expire: 30000* refresh-token-expire: 1209600000  
  
  
cloud:  
 aws:  
 s3:  
 bucket: ssafy.moamoa.image.bucket  
 dir: /images  
 credentials:  
 access-key: AKIAVYLW4VT3B4HD4ENH  
 secret-key: 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  
 region:  
 static: ap-northeast-2  
 auto: false  
 stack:  
 auto: false  
  
 *# multipartFile 용량 늘려주는 설정* servlet:  
 multipart:  
 max-file-size: 30MB  
 max-request-size: 30MB

## 빌드하기

1) Front

npm i

npm run build

2) Back-spring

Gradle 실행

Bootjar 실행

## 배포하기

## Docker 설치

## sudo apt-get install docker

## Docker Container

## Nginx, redis, mysql back\_server docker

## Front\_server mount nginx

Nginx 설정

**nginx.conf**

user nginx;

worker\_processes auto;

error\_log /var/log/nginx/error.log notice;

pid /var/run/nginx.pid;



events {

worker\_connections 1024;

}

http {

include /etc/nginx/mime.types;

default\_type application/octet-stream;

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for"';

access\_log /var/log/nginx/access.log main;

sendfile on;

#tcp\_nopush on;

keepalive\_timeout 65;

#gzip on;

include /etc/nginx/conf.d/\*.conf;

}

**default.conf**

upstream app {

server back\_server:8080;

}

server {

listen 443 ssl;

server\_name moaamoaa.com;



ssl\_certificate /etc/letsencrypt/live/moaamoaa.com/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/moaamoaa.com/privkey.pem;

location / {

root /usr/share/nginx/html;

index index.html index.htm;

}

location /api {

proxy\_pass http://app;

#proxy\_set\_header X-Real-IP $remote\_addr;

#proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

#proxy\_set\_header Host $http\_host;

}

error\_page 404 /404.html;

location = /404.html {

root /usr/share/nginx/html;

index 404.html;

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root /usr/share/nginx/html;

index 500.html;

}

}



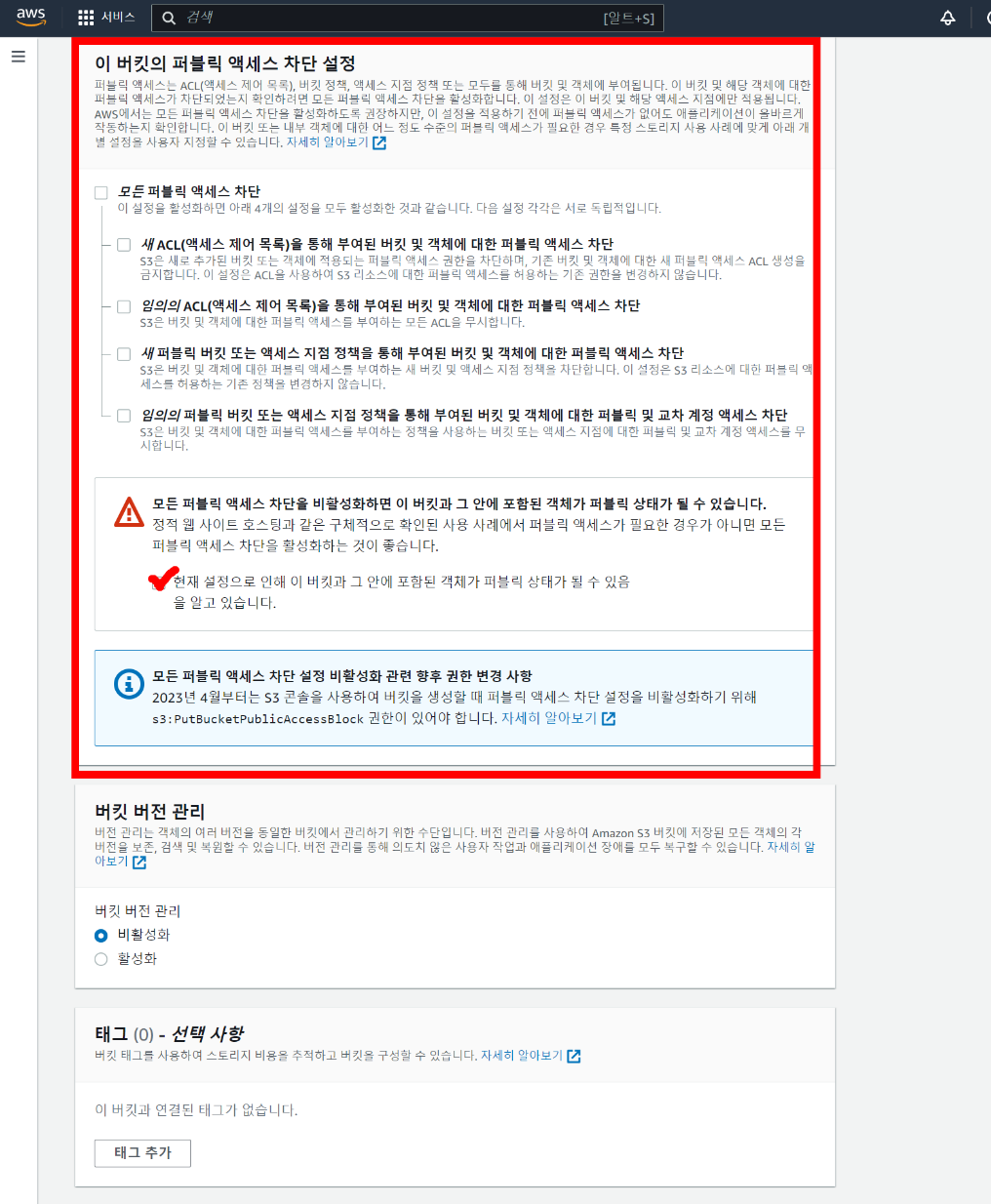
## 서비스 이용 방법(외부)

### Amazon S3 Bucket

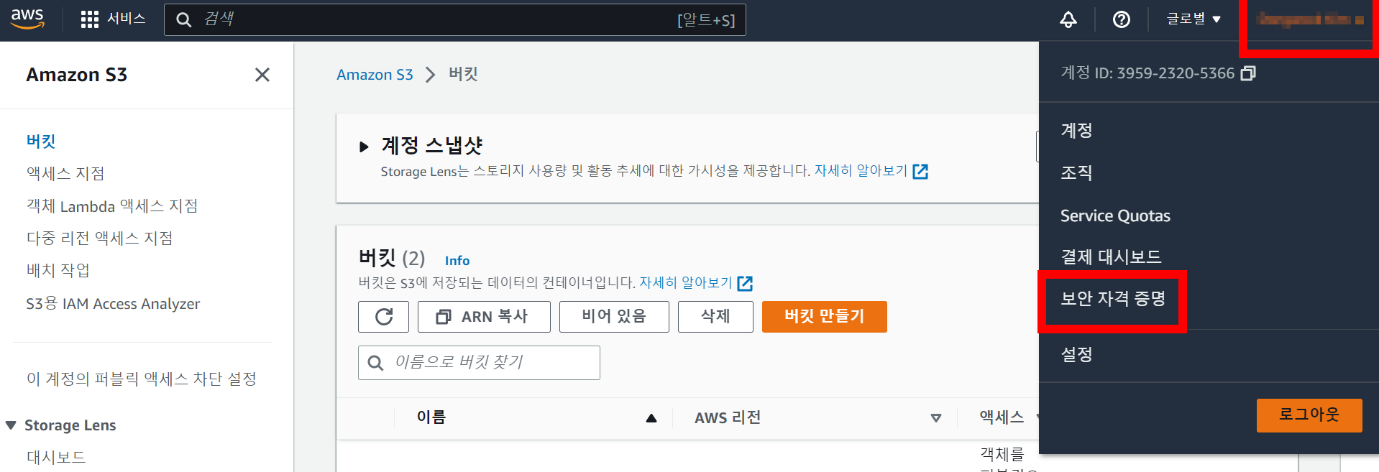
<https://s3.console.aws.amazon.com/s3/get-started?region=ap-northeast-2&region=ap-northeast-2> 접속



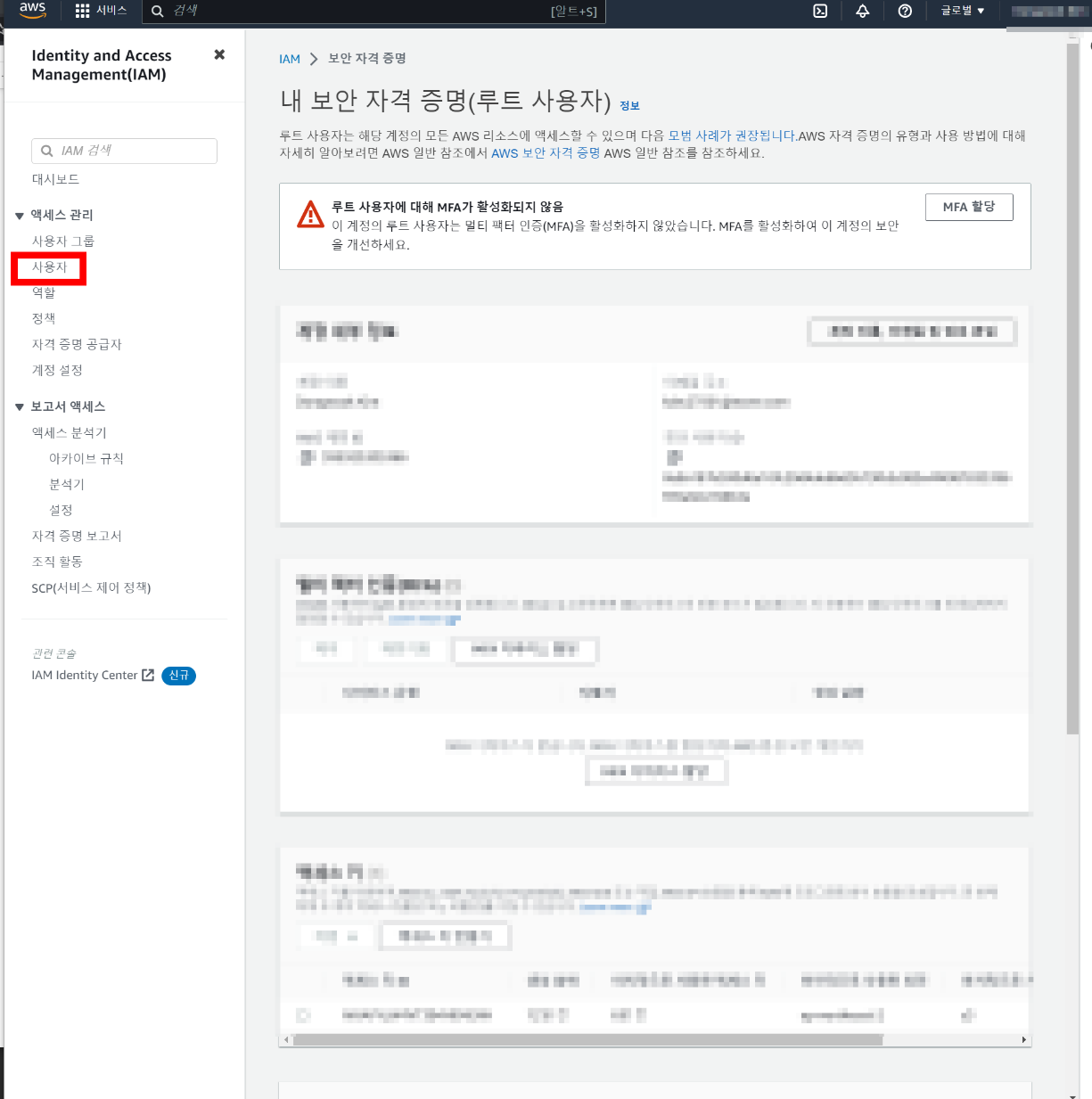


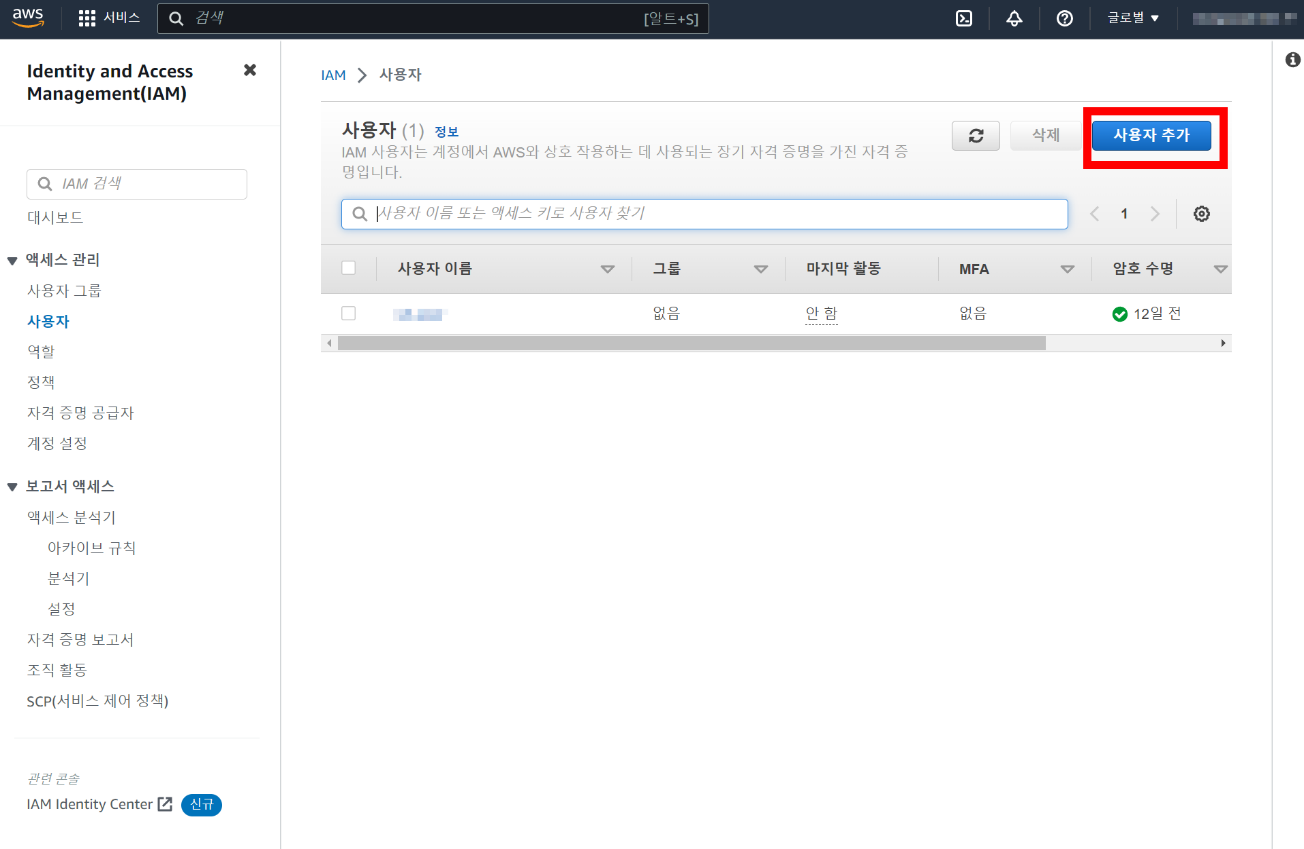


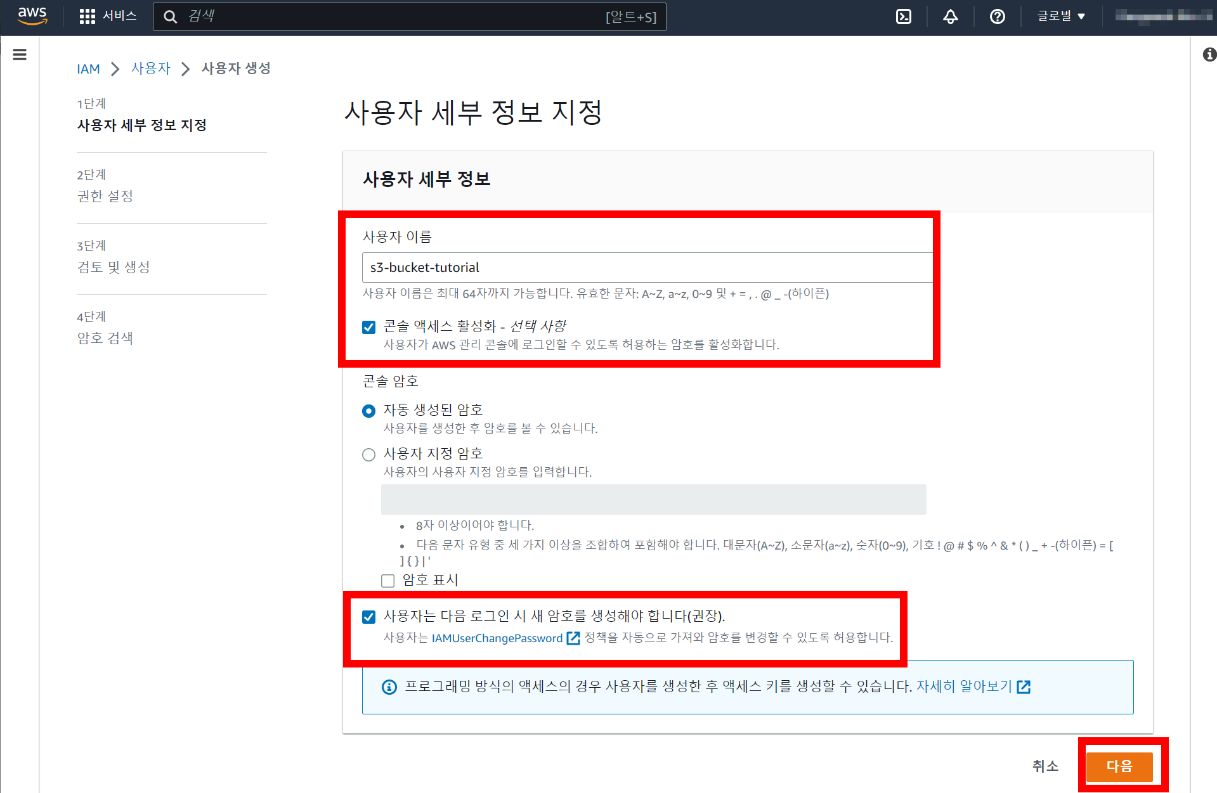
버킷 생성 후 오른쪽 상단에 있는 메뉴에서 [보안 자격 증명] 클릭



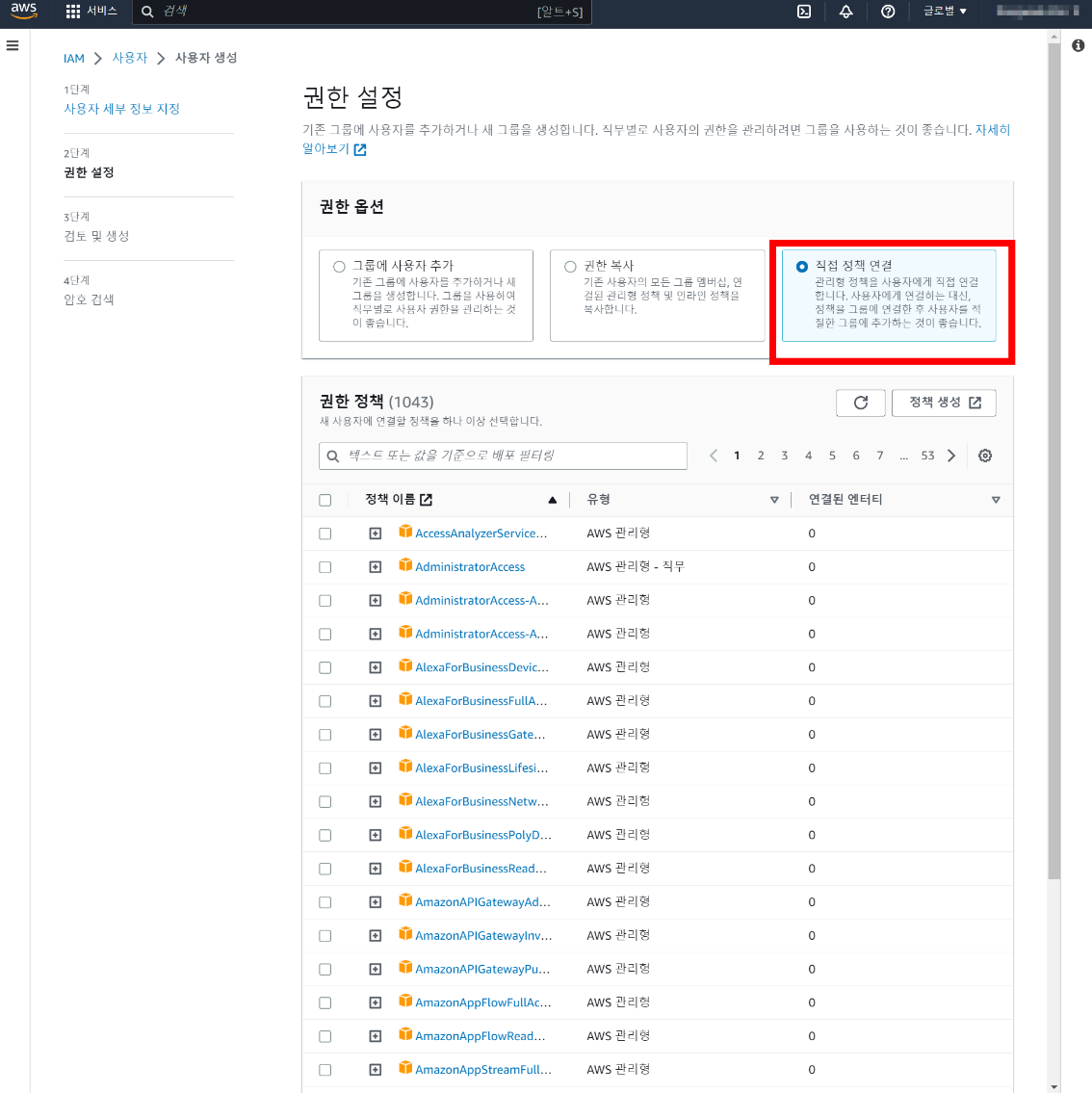


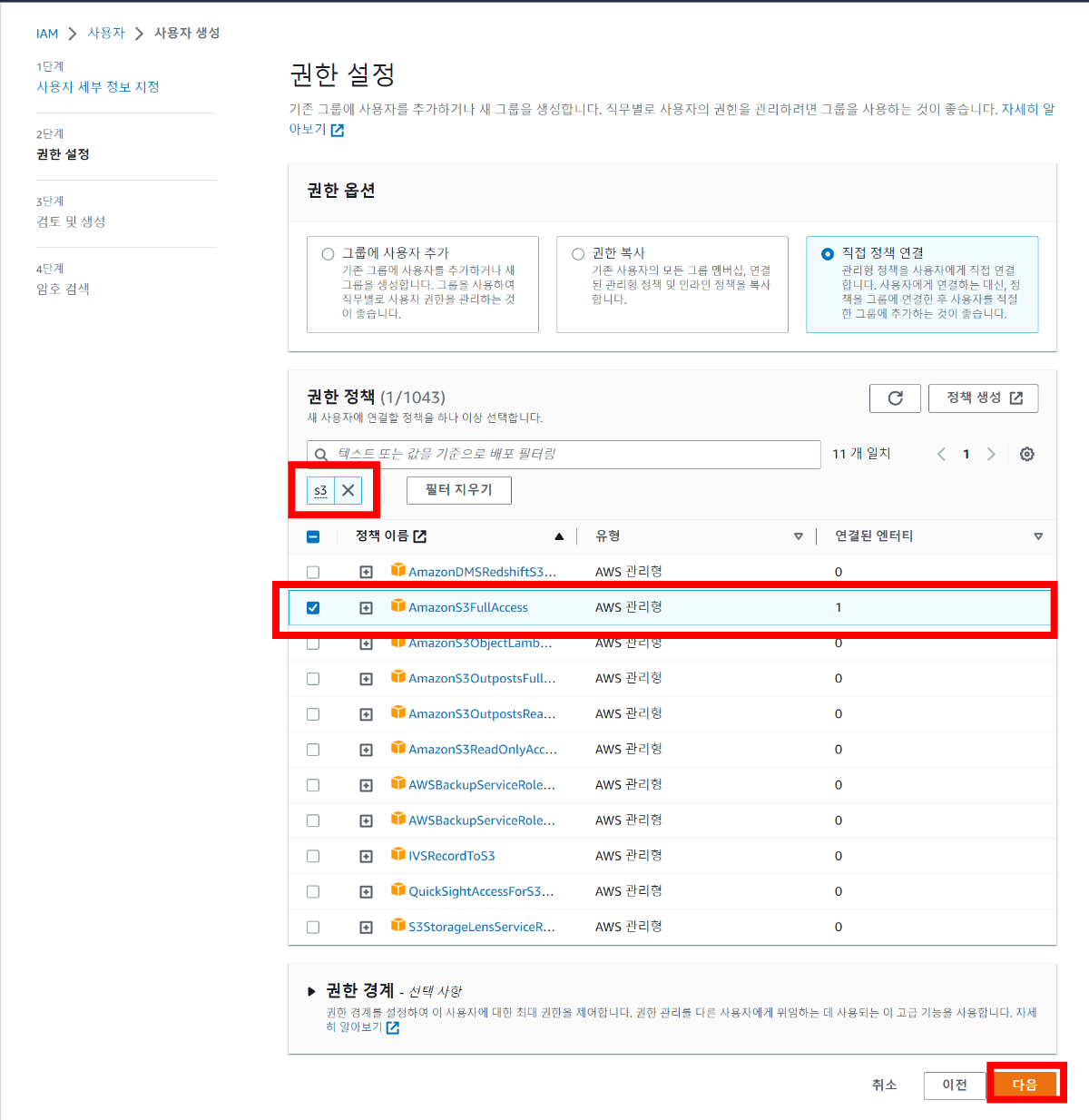


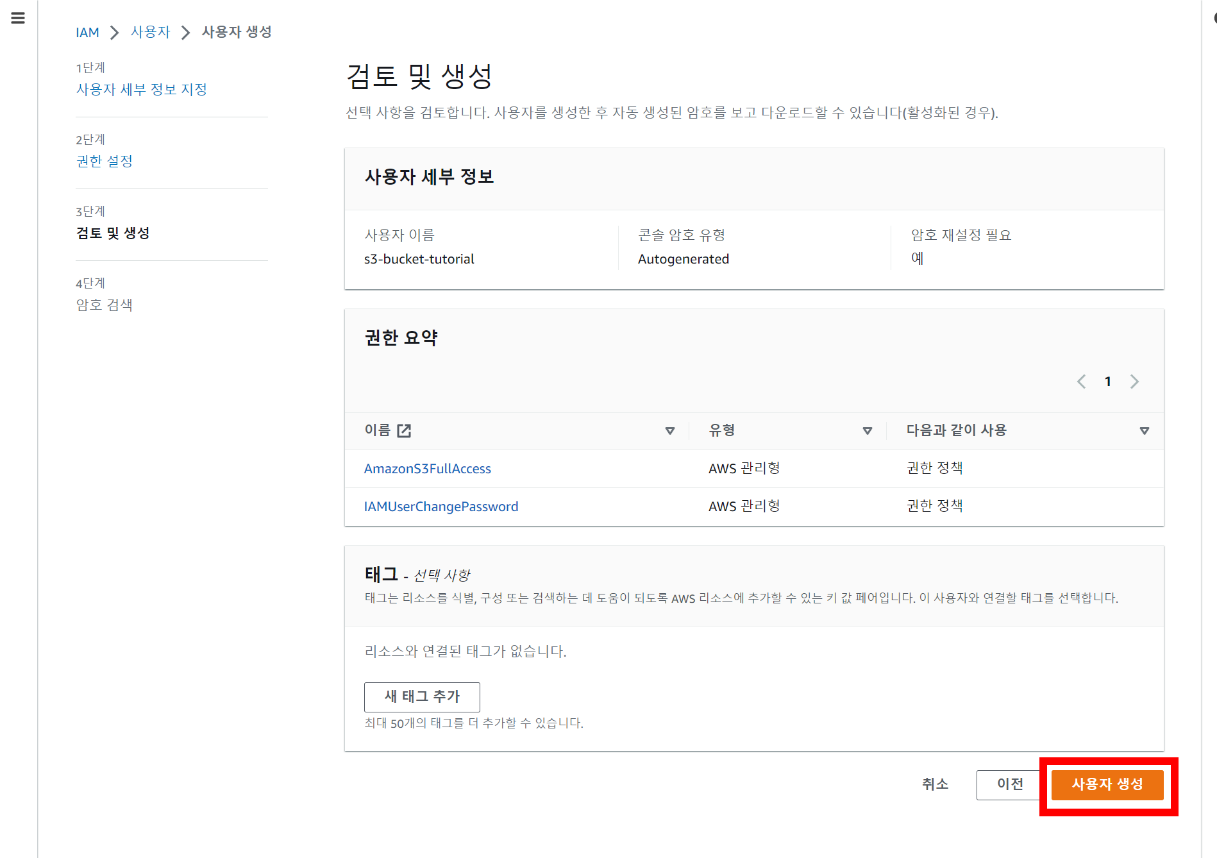




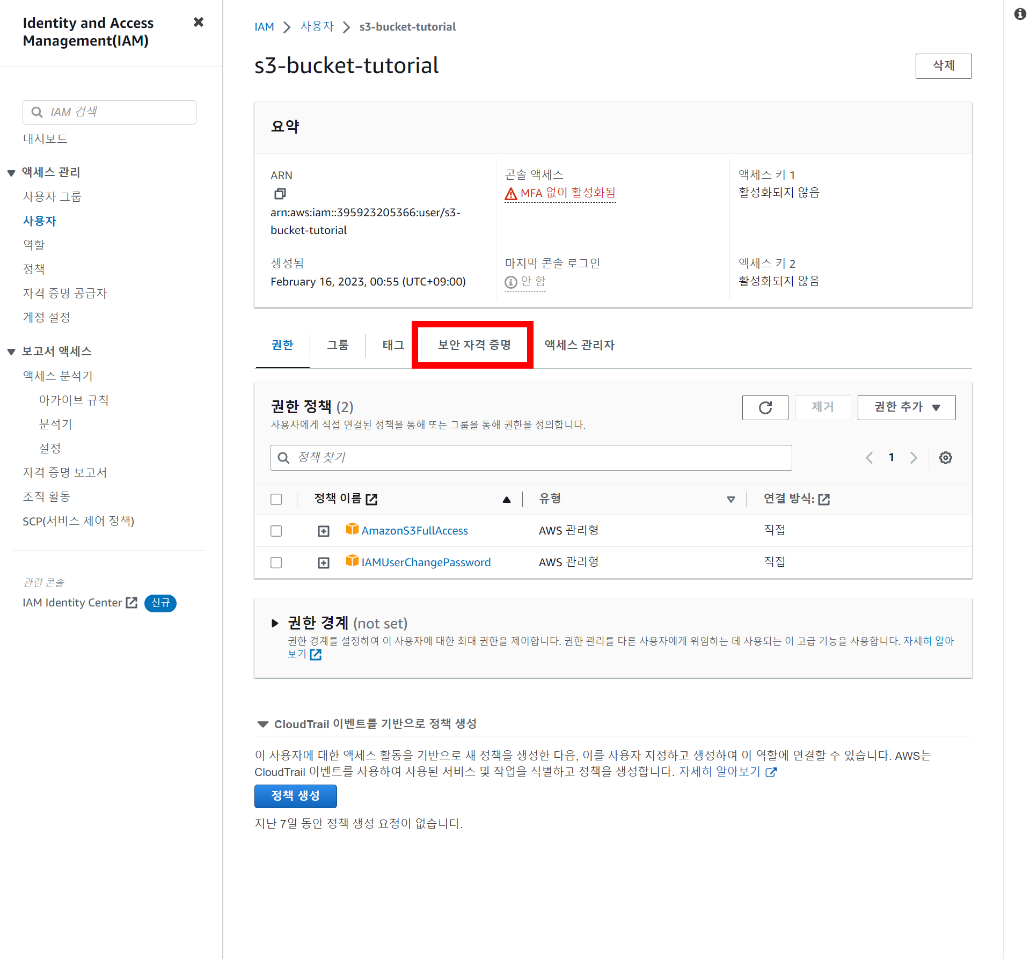








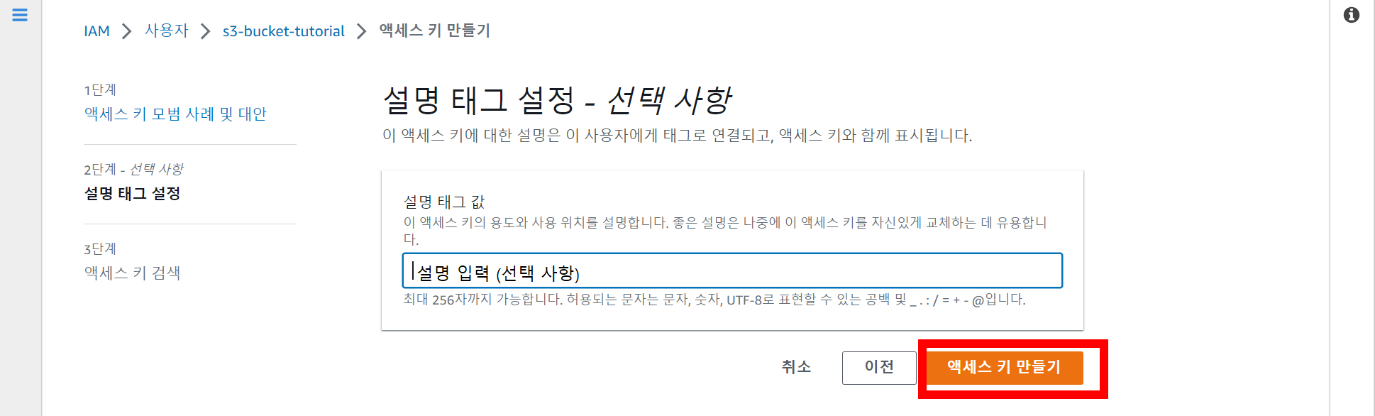
사용자 생성 후 해당 사용자 정보 접근 후 IAM 설정

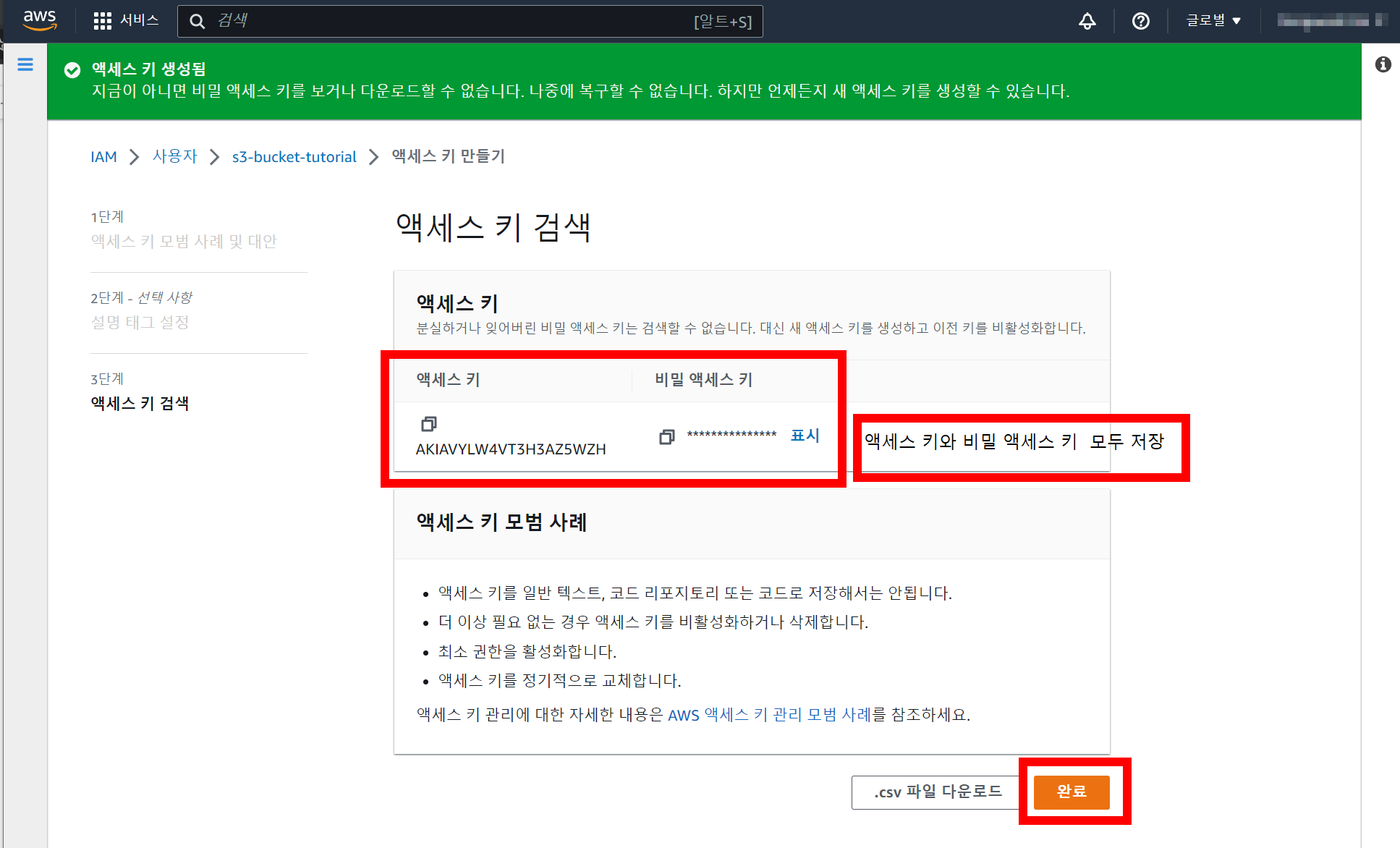




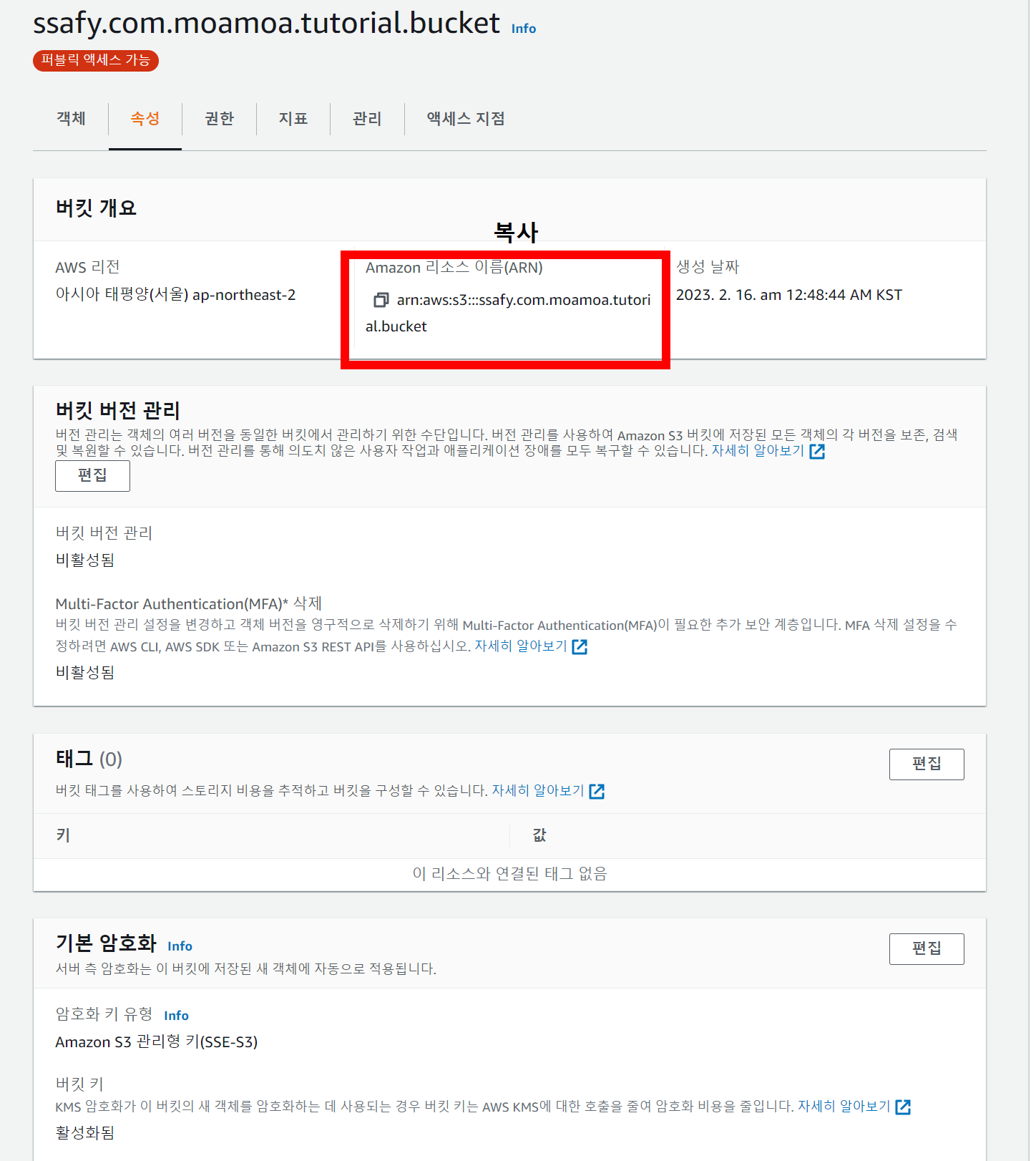


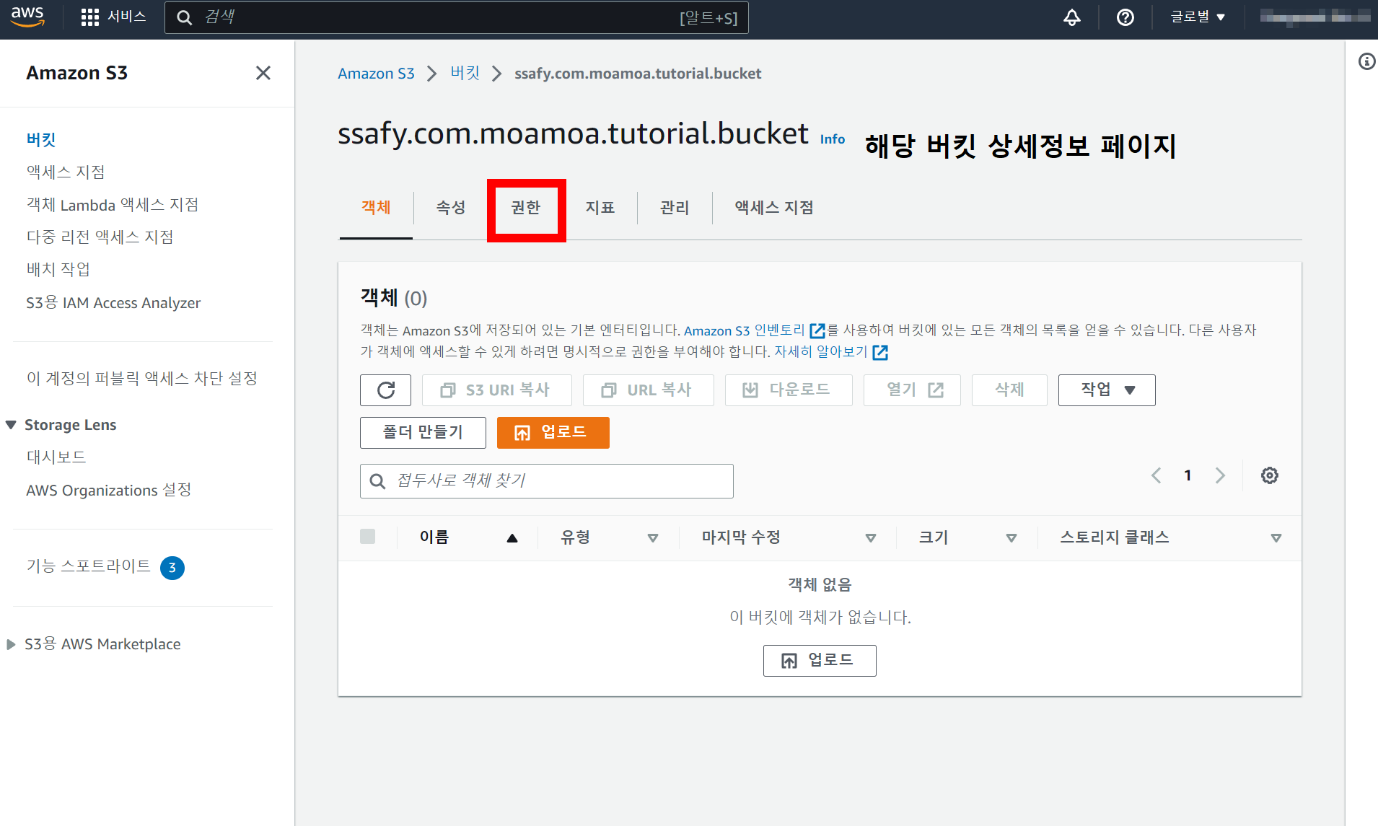


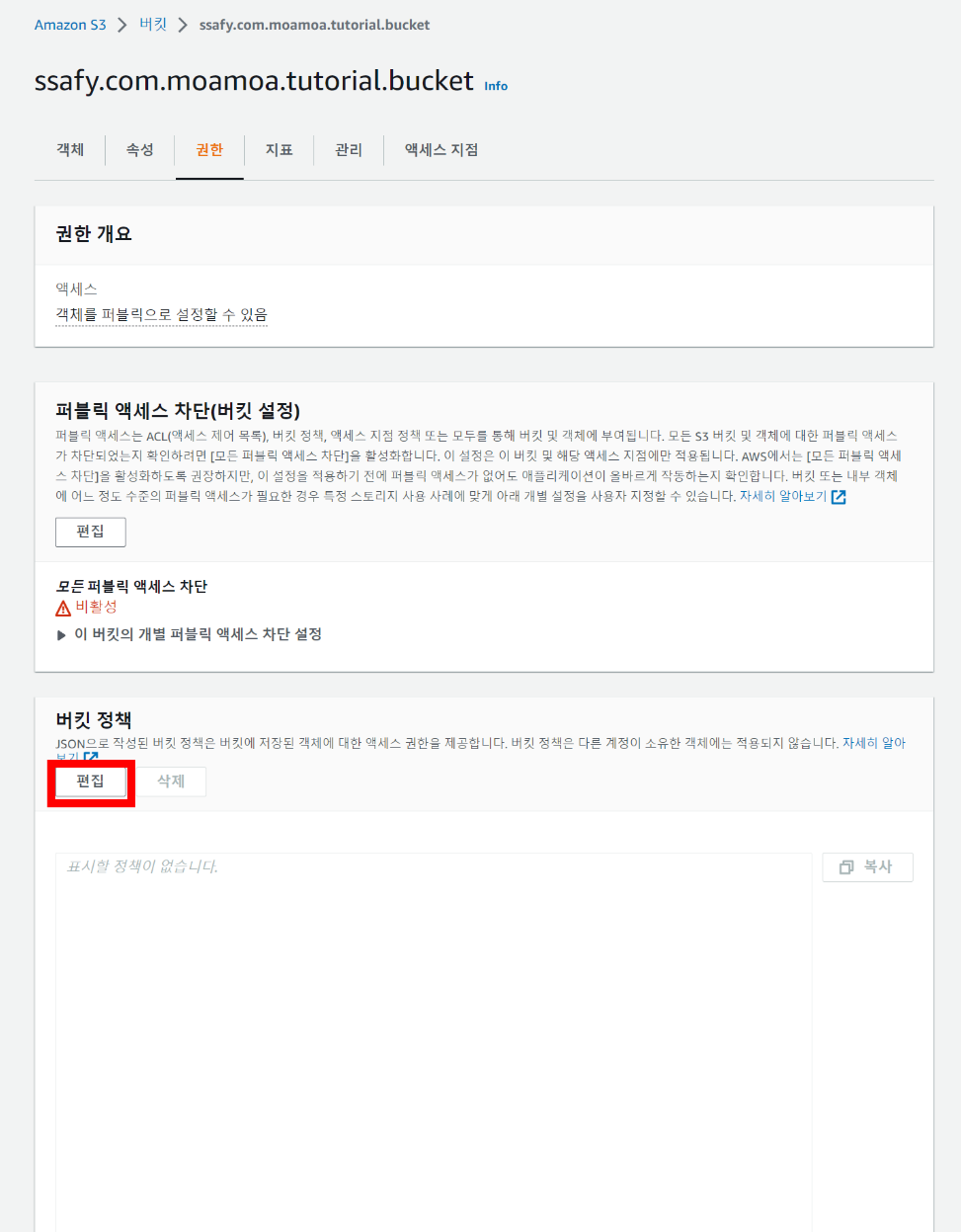




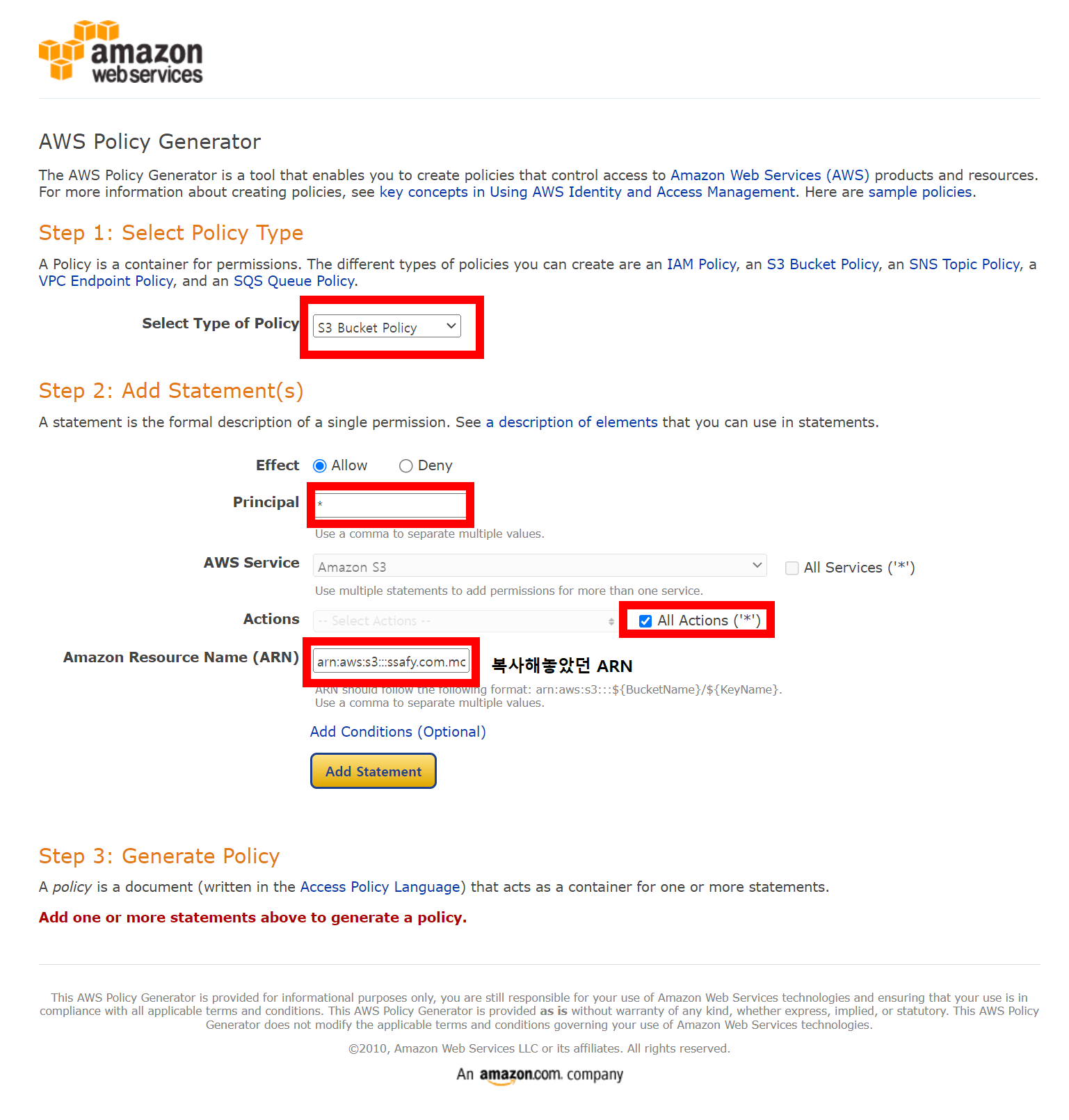
버킷 생성 후 해당 버킷 정책 설정

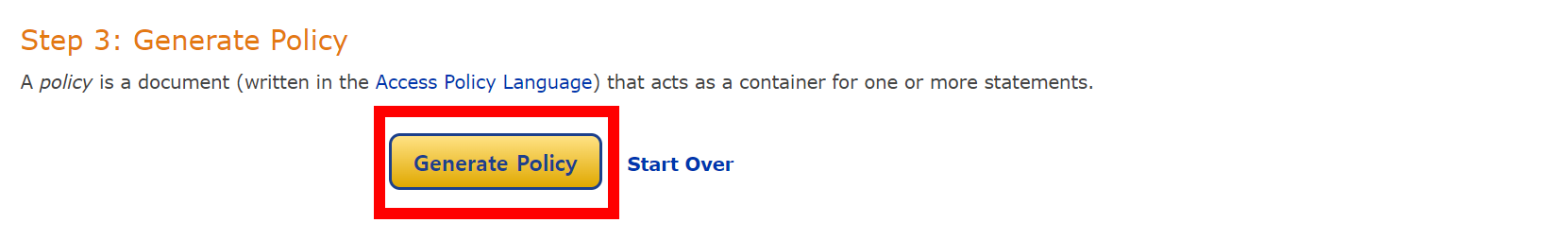


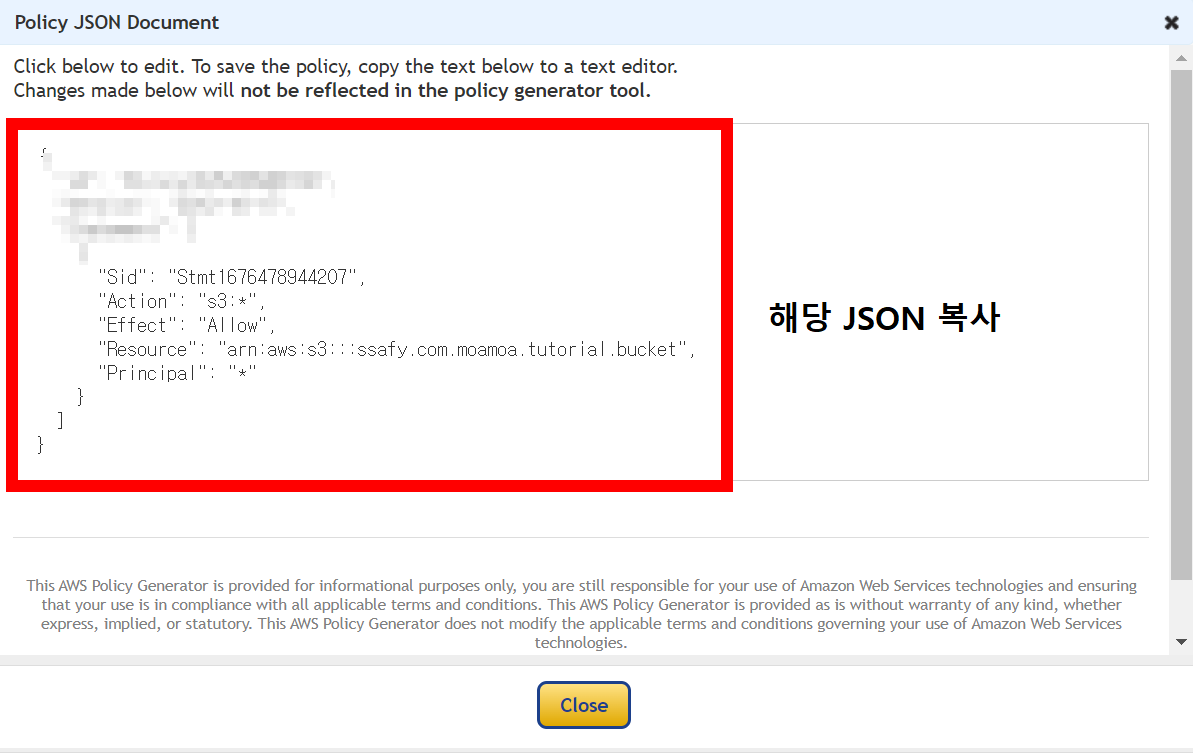














dependencies **{** implementation 'io.awspring.cloud:spring-cloud-starter-aws:2.4.4'  
**}**

application.yml

