업무참고 2025-클라우드대응-②-56

금융분야
상용 클라우드컴퓨팅서비스 보안 관리 참고서

Amazon Web Services

금융보안원
FINANCIAL SECURITY INSTITUTE

Amazon Web Services

CONTENTS

1. 가상자원 관리

1.1. 가상자원 생성 시 최초 계정에 대한 비밀번호 규칙 수립
1.2. 이용자 가상자원 접근 시 로그인 규칙 적용
1.3. 가상자원 루트 계정 접근 시 추가 인증수단 적용
1.4. 가상자원 생성 시 네트워크 설정 적용
1.5. 가상자원 접속 시 보안 방안 수립
1.6. 이용자 가상자원 별 권한 설정
1.7. 이용자 가상자원 내 악성코드 통제방안 수립

1 가상자원 관리

1 기준

식별번호	기준	내용
1.1.	**가상자원 생성 시 최초 계정에 대한 비밀번호 규칙 수립**	**이용자 가상자원 생성 시 최초 계정에 대한 비밀번호 규칙을 수립하여야 한다.**

2 설명

- 이용자 가상자원에 접근하는 계정에 대한 비밀번호 규칙 보안 통제 방안을 수립하여야 한다.

  - 예시

    1) 제3자가 쉽게 유추할 수 없는 비밀번호 작성 규칙 수립

3 우수 사례

- **(가상자원관리시스템)** 해당 사항 없음

- **(AWS CLI)** 해당 사항 없음

4 참고 사항

- AWS의 경우 가상자산에 접근하는 계정의 인증은 PEM 키를 이용합니다. 따라서, 별도의 비밀번호 규칙이 적용되지 않습니다.

[문의처에 관한 정보 - 없음]

1 기준

식별번호	기준	내용
1.2.	**이용자 가상자원 접근 시 로그인 규칙 적용**	**이용자 가상자원 접근 계정에 대한 안전한 로그인 규칙을 수립하여야 한다.**

2 설명

- 이용자는 패스워드 무작위 대입 공격 등에 대응하기 위해 가상자원 접근계정에 대한 안전한 로그인 규칙을 수립하여야 한다.

  - 예시

    1) 로그인 오류에 따른 보안 통제 방안 수립 등

3 우수 사례

- **(가상자원관리시스템)** 해당 사항 없음

- **(AWS CLI)** 해당 사항 없음

4 참고 사항

- AWS의 경우 가상자산에 접근하는 계정의 인증은 PEM 키를 이용합니다. 따라서, 패스워드 무작위 대입 공격은 유효하거나 가능하지 않아 로그인 규칙을 수립이 불필요 합니다.

[문의처에 관한 정보]

1 기준

식별번호	기준	내용
1.3.	**가상자원 루트 계정 접근 시 추가 인증수단 적용**	**이용자 가상자원 루트 계정(root, administrator 등) 접근 시 추가인증 수단을 확보하여야 한다.**

2 설명

- 이용자 가상자원(EC2 인스턴스) 루트 계정 접근 시 추가인증 수단이 확보되어야 한다.(단, 기능이 제공되지 않는 경우 안전한 로그인 수단을 확보하여야 한다.)
  - 예시
    1) 이메일 인증
    2) SMS 인증
    3) 별도 인증도구 활용
    4) SSH PEM Key 등을 통한 안전한 로그인 수단 확보 등
    5) MFA 인증 - 가상자원의 MFA 인증 설정은 소프트웨어 방식의 MFA만 지원한다.

3 우수 사례

1) Multi-Factor Authenticator를 이용한 추가 인증

- EC2 등 가상자원에 대한 MFA 인증은 소프트웨어 방식으로만 지원합니다.
- 본 참고서에서는 Google Authenticator 를 이용한 추가 인증 방식에 대해 설명합니다.

- **(AWS 관리 콘솔)** 해당 사항 없음

- **(AWS CLI)** 해당 사항 없음

- **(가상자원 내 CLI)**
  - 가상자원에 관리자 권한으로 로그인 후 MFA 프로그램을 설치합니다.
  - 참고로 MFA 프로그램 설치 과정 진행을 위해서 사전에 휴대폰에 Google Authenticator 앱이 설치되어 있어야 합니다.

1)-1. 가상자원에 관리자 권한으로 로그인 한 후 아래의 명령을 입력하여 Google Authenticator 프로그램을 설치 합니다.

```
sudo yum install google-authenticator -y
```

1)-2. 프로그램 설치 후 다음의 명령어로 프로그램을 실행합니다.

```
google-authenticator
```

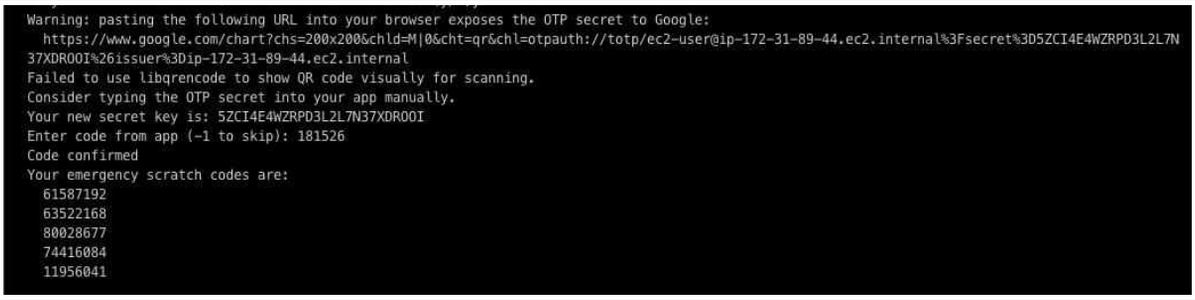
1)-3. MFA 사용 여부를 묻는 질문에 “y” 를 입력합니다.

```
Do you want authentication tokens to be time-based (y/n) y
```

1)-4. 다음 그림의 URL 링크를 복사하여 웹 브라우저에서 접속하면 QR코드가 출력되며, 휴대폰의 Google Authenticator 앱을 실행한 후 해당 QR코드를 스캔하여 MFA를 등록합니다.

| 그림 1-3-1 | Google MFA 등록을 위한 URL 정보

1)-5. 등록이 정상 완료되었다면, 등록된 MFA의 임시코드를 입력합니다. 임시코드를 입력하면 “응급 복구 코드”가 출력되는데 문제가 발생할 경우 사용이 필요한 중요한 코드이므로 복사하여 안전하게 보관해야 합니다.



| 그림 1-3-2 | Google MFA 응급 복구 코드

1)-6. 이후 출력되는 모든 질문에 대해서는 “y” 로 입력합니다.

```
Do you want me to update your "/home/ec2-user/.google_authenticator" file? (y/n)y

Do you want to disallow multiple uses of the same authentication
token? This restricts you to one login about every 30s, but it increases
your chances to notice or even prevent man-in-the-middle attacks (y/n) y

By default, a new token is generated every 30 seconds by the mobile app.
In order to compensate for possible time-skew between the client and the server,
we allow an extra token before and after the current time. This allows for a
time skew of up to 30 seconds between authentication server and client. If you
experience problems with poor time synchronization, you can increase the window
from its default size of 3 permitted codes (one previous code, the current
code, the next code) to 17 permitted codes (the 8 previous codes, the current
code, and the 8 next codes). This will permit for a time skew of up to 4 minutes
between client and server.
Do you want to do so? (y/n) y

If the computer that you are logging into isn't hardened against brute-force
login attempts, you can enable rate-limiting for the authentication module.
By default, this limits attackers to no more than 3 login attempts every 30s.
Do you want to enable rate-limiting? (y/n) y
```

| 그림 1-3-3 | Google MFA 추가 설정 옵션

1)-7. 다음으로 MFA를 사용할 수 있도록 PAM 모듈을 수정해야 합니다. 아래의 명령어를 입력하여 ‘/etc/pam.d/sshd’ 파일 수정 화면으로 진입합니다.

```
sudo vi /etc/pam.d/sshd
```

1)-8. 수정 화면 내용 중 “auth” 항목에 대하여 다음의 내용을 추가하고 저장합니다.

```
auth required pam_google_authenticator.so nullok
```

1)-9. 다음으로 SSH 접속 시 MFA가 사용될 수 있도록 Config를 수정해야 합니다. 이에 ‘/etc/ssh/sshd_config’ 파일 수정 화면으로 진입합니다.

```
sudo vi /etc/ssh/sshd_config
```

1)-10. 그리고 아래의 내용을 추가 반영 합니다.

```
PubkeyAuthentication yes
KbdInteractiveAuthentication yes
UsePAM yes
AuthenticationMethods publickey,keyboard-interactive
```

1)-11. 다음으로 OpenSSH 보안 설정 파일인 '50-redhat.conf' 파일 수정 화면으로 진입합니다.

```
sudo vi  /etc/ssh/sshd_config.d/50-redhat.conf
```

1)-12. 그리고 아래의 내용을 반영합니다.

```
ChallengeResponseAuthentication yes
```

1)-13. 마지막으로 SSH 데몬을 재기동 합니다.

```
sudo systemctl restart sshd
```

1)-14. SSH 데몬 재기동 후 SSH로 접속 시 MFA 인증 요청이 정상적으로 나타나는지를 확인합니다.

| 그림 1-3-4 | SSH로 접속 시 MFA 요청 화면

4 참고 사항

- 본 참고서의 예제로 사용한 Google Authenticator 이외의 다른 MFA 소프트웨어도 사용 가능하며 다른 MFA 소프트웨어 사용 시 해당 소프트웨어의 사용 방법은 각 제조사에서 제공하는 이용가이드를 참고하도록 합니다.

1 기준

식별번호	기준	내용
1.4.	**가상자원 생성 시 네트워크 설정 적용**	**이용자의 가상자원 생성 시 안전한 네트워크 설정을 적용하여야 한다.**

2 설명

○ 외부에서 직접 접속이 불필요한 경우 내부IP 또는 IP대역에서만 접근할 수 있도록 설정하여야 한다.

- 예시

  1) 가상자원 접속 가능한 공인IP(외부)대역 점검 및 제거

  2) 접근가능한 IP 또는 IP 대역 설정

  3) VPC 및 보안 그룹을 통한 내부 네트워크 대역 접근 설정

3 우수 사례

1) 가상자원 접속 가능한 공인IP(외부)대역 점검 및 제거

- 가상자원 생성 시 반드시 보안 그룹을 선택하여 가상자원에 대한 인바운드/아웃바운드 규칙을 적용해야 합니다.
- 보안 그룹의 인바운드 규칙은 가상자원을 기준으로 외부에서 가상자원으로 유입되는 트래픽을 허용하는 규칙으로 이루어짐
- 보안 그룹의 인바운드 규칙은 “프로토콜”, “포트범위”, “소스” 로 구성된다.
- 보안 그룹의 아웃바운드 규칙은 가상자원을 기준으로 가상자원에서 외부로 유출되는 트래픽을 허용하는 규칙으로 이루어짐
- 보안 그룹의 아웃바운드 규칙은 “프로토콜”, “포트범위”, “대상” 으로 구성된다.

○ **(AWS 관리 콘솔)** 외부 IP대역에 대한 점검은 콘솔 홈’ → ‘EC2’ → ‘네트워크 및 보안’ → ‘보안 그룹’ → 특정 ’보안 그룹’ 선택 후 가상자원에 부여된 보안 그룹의 인바운드 규칙에서 확인할 수 있습니다.

- 보안 그룹의 인바운드 규칙 중 프로토콜 “TCP”, 포트범위 “22” 를 이용하여 외부 공인 IP에서 SSH 접근을 허용하는 규칙이 있는지 확인합니다.

| 그림 1-4-1 | 보안 그룹 인바운드 규칙 확인

- 점검 시 불필요한 IP대역이 확인될 경우 해당 화면에서 '인바운드 규칙 편집'을 클릭하여 삭제 등의 조치를 할 수 있습니다.
- 그리고 보안상 SSH 접근(프로토콜 "TCP", 포트범위 "22")에 대해서는 소스 대역을 "0.0.0.0/0"으로 사용하지 않도록 합니다.

| 그림 1-4-2 | 보안 그룹 인바운드 규칙 편집

(AWS CLI)

- API 환경을 통해 보안 그룹 규칙 확인은 아래와 같이 할 수 있습니다.

```
aws ec2 describe-security-groups --group-ids [Security Group ID]
```

Output:

```
{
   "SecurityGroups": [
      {
         "IpPermissionsEgress": [
            {
               "IpProtocol": "-1",
               "IpRanges": [
                  {
                     "CidrIp": "0.0.0.0/0"
                  }
               ],
               "UserIdGroupPairs": [],
               "PrefixListIds": []
            }
         ],
…(생략)…
}
```

| 그림 1-4-3 | CLI 명령 - 보안 그룹 규칙 확인

- 보안 그룹 규칙 점검 후 불필요한 규칙 삭제는 아래와 같이 할 수 있습니다.

```
aws ec2 revoke-security-group-ingress --group-name [보안 그룹 이름] --protocol [프로토콜] \
   --port [포트번호] --cidr [삭제 대상 IP 대역]
```

Output:

```
{
   "Return": true
}
```

| 그림 1-4-4 | CLI 명령 - 인바운드 보안 그룹 규칙 삭제

2) 접근가능한 IP 또는 IP 대역대 설정

- **(AWS 관리 콘솔)** '콘솔 홈' → 'EC2' → '네트워크 및 보안' → '보안 그룹' → 특정 '보안 그룹' 선택→'아웃바운드 규칙' →'인바운드 규칙 편집' 클릭 후 인바운드 규칙에서 가상자원이 접속 가능한 공인 IP(외부)대역 설정할 수 있습니다.
  - 가상자원 관리를 위한 인바운드 규칙 설정 시 포트를 범위로 사용하는 것보다 특정 포트로 한정하여 지정합니다.

| 그림 1-4-5 | 보안 그룹 인바운드 규칙 편집

- **(AWS CLI)** 인바운드 보안 규칙 편집은 아래와 같이 할 수 있습니다.

```
aws ec2 modify-security-group-rules --group-id [보안 그룹 ID] \
   --security-group-rules SecurityGroupRuleId=[인바운드 보안 그룹 규칙 ID], SecurityGroup  \
   Rule='{Description=[주석],IpProtocol=-1,CidrIpv4=[허용 대상 IP 대역]}'
```

Output:

```
{
   "Return": true
}
```

| 그림 1-4-6 | CLI 명령 - 보안 그룹 인바운드 규칙 편집

3) VPC 및 보안 그룹을 통한 내부 네트워크 대역 접근

- **(AWS 관리 콘솔)** '콘솔 홈' → 'EC2' → '네트워크 및 보안' → '보안 그룹' → 특정 '보안 그룹' 선택→'아웃바운드 규칙' →'아웃바운드 규칙 편집' 클릭 후 아웃바운드 규칙에서 가상자원이 접속 가능한 VPC Subnet IP(내부)대역 설정할 수 있습니다.

| 그림 1-4-7 | 보안 그룹 아웃바운드 규칙 편집

- **(AWS CLI)** 아웃바운드 보안 규칙 편집은 아래와 같이 할 수 있습니다.

```
aws ec2 modify-security-group-rules --group-id [보안 그룹 ID] \
   --security-group-rules SecurityGroupRuleId=[아웃바운드 보안 그룹 규칙 ID],SecurityGroup \
   Rule='{Description=test,IpProtocol=-1,CidrIpv4=[허용 대상 IP 대역]}'
```

Output:

```
{
   "Return": true
}
```

| 그림 1-4-8 | CLI 명령 - 보안 그룹 아웃바운드 규칙 편집

4 참고 사항

- 가상자원 생성 시 할당되어야 하는 보안 그룹은 접근 허용 네트워크를 포함하여 규칙을 미리 작성해야 합니다.
- 여러 보안 그룹에서 공통적으로 참조해야 하는 IP 리스트가 있는 경우에는 VPC 에서 지원하는 IP Prefix 기능을 활용할 수 있습니다.

1 기준

식별번호	기준	내용
1.5.	**가상자원 접속 시 보안 방안 수립**	**이용자 가상자원 접속 시 안전한 인증절차를 통해 접속하여야 한다.**

2 설명

- 이용자의 가상자원(인스턴스) 접속 시 안전한 방식을 통해 접근하여야 한다.

  - 예시

    1) SSH를 통한 접속 시 안전한 계정관리 수행(ex. ID/PW 기반이 아닌 Certificate 기반 인증 방식 적용 등)

    2) 클라우드 웹 콘솔에서 직접 실행 시 안전한 인증 방식 적용(해당 인스턴스를 호출할 수 있는 권한을 지닌 이용자인지 검증 등)

3 우수 사례

- **(AWS 관리 콘솔)** 키 페어는 '콘솔 홈' → 'EC2' → '네트워크 및 보안' → '키 페어' 에서 키 페어 생성 버튼을 클릭하여 생성할 수 있습니다.

| 그림 1-5-1 | PEM 키 생성

- **(AWS CLI)** 키 페어는 아래와 같이 생성할 수 있습니다.

```
aws ec2 create-key-pair --key-name [키 페어 이름]
```

Output:

```
{
    "KeyFingerprint": "81:f2:15:31:67:9d:cb:ad:59:eb:08:9a:bc:72:a1:b7:64:9c:75:50",
    "KeyMaterial": "-----BEGIN RSA PRIVATE KEY-----₩nMIIEowIBAAKCAQEAjEPtnW198/kFu3O7ZdVIlxus
    /ZITYalOS67zKe+l8oKmhyNU₩nrbzYu+x9nchMzBGv3GWm5emtAnbDN6gvXQQqdeR9bV3gu9D
    c5OJCOARGrxczg38h₩nlHpWKHUWXkhimjRKjJVOr3A0₩n-----END RSA PRIVATE KEY-----",
    "KeyName": "fcloud",
    "KeyPairId": "key-0d3a9c3eb550e6d2b"
}
```

| 그림 1-5-2 | CLI 명령 - 키 페어 생성

4 참고 사항

- AWS의 경우 가상자산(EC2 등)에 접근하는 계정에 대한 인증을 PEM 키를 이용합니다.

1 기준

식별번호	기준	내용
1.6.	**이용자 가상자원 별 권한 설정**	**이용자 직무 및 권한에 따른 가상자원 별 접근통제 방안(권한 설정 등)을 수립하여야 한다.**

2 설명

- 이용자 직무 및 권한에 따른 가상자원 별 접근통제 방안(권한 설정 등)을 수립하여야 한다.

  - 예시

    1) 가상자원 종류 별 접근통제 방안 수립(ex. IAM을 통한 접근권한 관리)

       - 모든 가상자원에 접근 가능한 Role에 대해서는 최소 인원에 대해서만 부여

3 우수 사례

- **(AWS 관리 콘솔)** 해당 없음

- **(AWS CLI)** 해당 없음

4 참고 사항

- AWS EC2 인스턴스에 PEM 키를 이용하여 접근하는 경우, 각 EC2 인스턴스 별 접근통제 방안은 PEM 키 파일에 접근 권한을 직무별로 부여하도록 해야 합니다.

1 기준

식별번호	기준	내용
1.7.	**이용자 가상자원 내 악성코드 통제방안 수립**	**이용자 가상자원 내 악성코드 통제방안을 수립하여야 한다.**

2 설명

- 이용자 가상자원 내 악성코드 통제방안을 수립하여야 한다.
  - 예시
    1) 이용자가 보유하고 있는 악성코드 통제방안 수립(백신 등)
    2) 클라우드 사업자가 악성코드 통제방안 제공(백신 등)
    3) 백신 등 설치가 불가능한 환경인 경우 그 수준에 준하는 악성코드 통제방안 수립
    4) Amazon GuardDuty 기능 활성화 – Amazon GuardDuty 의 경우 가상자원내의 멀웨어를 자동으로 탐지하는 기능을 제공하므로 GuardDuty 를 활성화하여 가상자원에서 사용되는 멀웨어의 탐지에 활용한다.

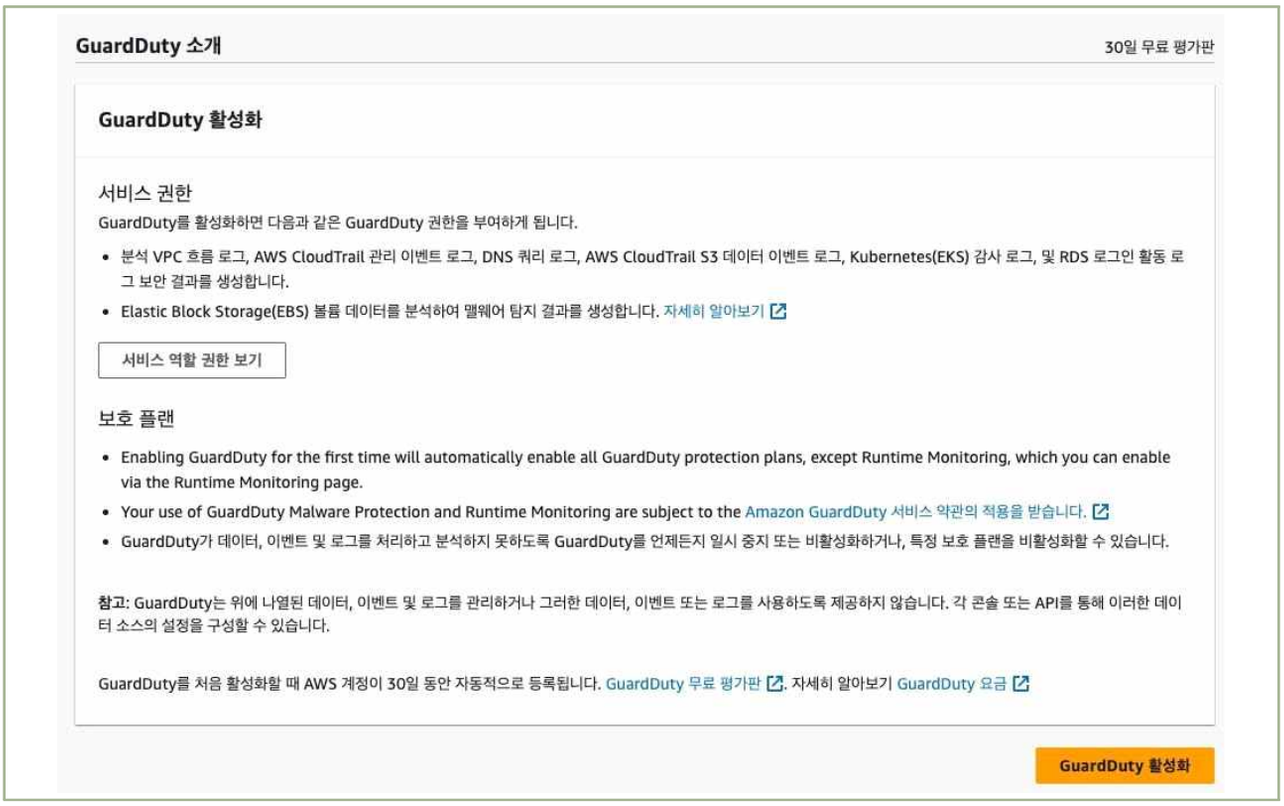
3 우수 사례

1) Amazon GuardDuty를 사용하여 가상자원 내 악성코드 탐지

- Amazon GuardDuty의 경우 가상자원 내의 악성코드를 자동으로 탐지하는 기능인 Malware Protection을 제공하므로 해당 서비스를 활성화 하여 가상자원 내 악성코드를 탐지하여 대응할 수 있습니다.

- **(AWS 관리 콘솔)** Amazon GuardDuty는 '콘솔 홈' → 'Amazon GuardDuty' → 'GuardDuty 활성화' 를 클릭하여 활성화 할 수 있습니다. (30일간 무료 기간 제공)
  - Amazon GuardDuty는 AWS 의 관리형 서비스로 EC2 인스턴스 위협, IAM 위협 등 기본적으로 제공하는 위협탐지 기능이외에 "멀웨어 탐지 기능" 을 추가로 제공합니다.
  - "멀웨어 탐지 기능" 은 GuardDuty에서 옵션으로 제공(추가 비용 발생)하는 기능으로 필요에 따라 활성화/비활성화가 가능합니다.
  - 신규 계정에서 Amazon GuardDuty를 활성화하는 경우 "멀웨어 탐지 기능"은 자동으로 활성화

됩니다.



| 그림 1-7-1 | Amazon GuardDuty 활성화

- **(AWS CLI)** Amazon GuardDuty의 CLI로 활성화는 아래와 같이 할 수 있습니다.

```
aws guardduty create-detector \
--enable

Output:
{
"DetectorId": "b6b992d6d2f48e64bc59180bfexample"
}
```

| 그림 1-7-2 | CLI 명령 - Amazon GuardDuty 활성화

4 참고 사항

- Amazon GuardDuty에서 제공하는 “멀웨어 탐지 기능” 은 Amazon GuardDuty에서 특정 탐지 내역이 발견되는 경우에 한하여 자동으로 대상 가상자원(EC2인스턴스)을 자동으로 스캔하는 방식으로 동작합니다.
- 그리고 가상자원을 보안 위협 탐지 유무와 관계없이 관리자의 요청에 따라 스캔하는 기능도 제공합니다.
- Amazon GuardDuty가 멀웨어를 스캔하는 경우 스캔 대상이 되는 가상자원의 성능에 영향이 없습니다.
- 본 가이드에서 다루지 않는 Amazon GuardDuty에 대한 다양한 기능들은 아래의 서비스 설명 가이드를 참고 바랍니다.

  - https://docs.aws.amazon.com/guardduty/latest/ug/what-is-guardduty.html

2. 네트워크 관리

2.1. 업무 목적에 따른 네트워크 구성
2.2. 내부망 네트워크 보안 통제
2.3. 네트워크 보안 관제 수행
2.4. 공개용 웹 서버 네트워크 분리
2.5. 네트워크 사설 IP 주소 할당 및 관리

2 네트워크 관리

1 기준

식별번호	기준	내용
2.1.	**업무 목적에 따른 네트워크 구성**	**클라우드 환경 내 업무 목적*에 따른 네트워크를 구성하여야 한다.** *** 개발, 운영, 업무 등**

2 설명

- 클라우드 환경 내 업무 목적(개발, 운영, 업무 등)에 따른 네트워크 구성 및 네트워크간 접근 통제 방안을 수립하여야 한다.

  - 예시

    1) VPC 등 네트워크 관련 기능을 통한 네트워크 구성 및 통제

    2) 보안 그룹(Security group) 또는 NACL(Network ACL) 등의 기능을 통한 네트워크 구성 및 통제(인/아웃바운드 통제 등)

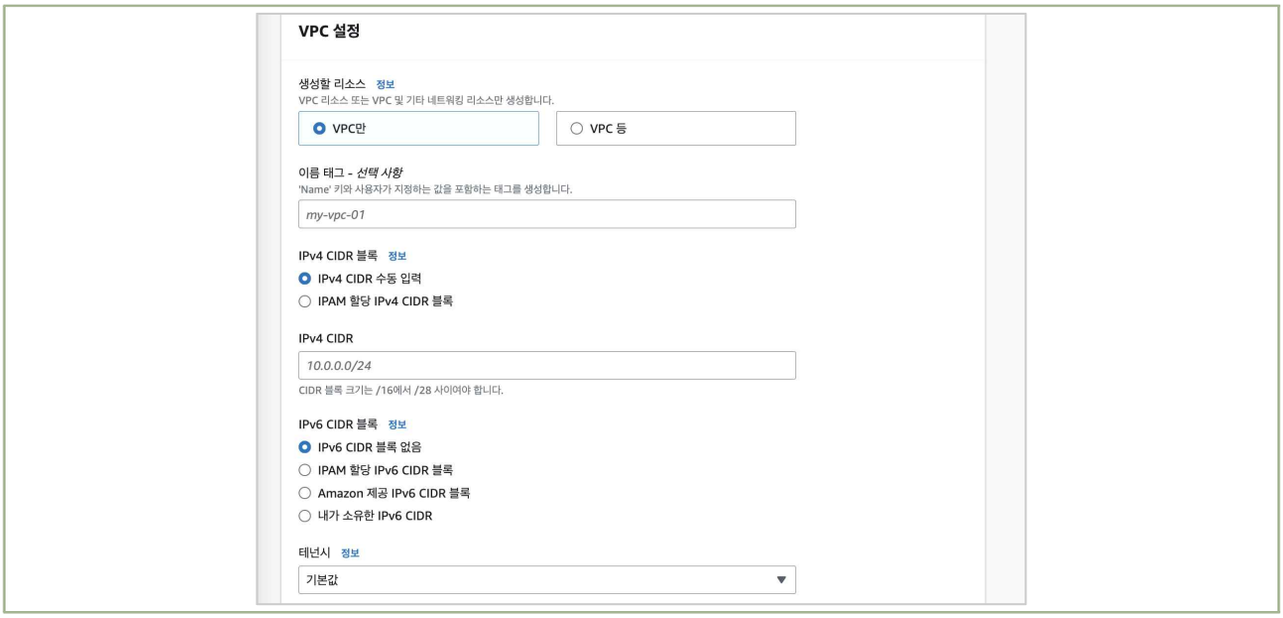
3 우수 사례

1) VPC 등 네트워크 관련 기능을 통한 네트워크 구성 및 통제

(1) VPC 생성

- 계정에 대한 전용 네트워크 공간인 VPC(Virtual Private Cloud)를 클라우드 환경 내 업무 목적(개발, 운영, 운영) 별로 생성할 수 있고 계정마다 리전(Region) 별 최대 5개의 VPC를 생성할 수 있습니다.
- VPC는 동일 계정 내 다른 VPC에 할당된 IP대역과 중복되지 않는 사설IP대역(CIDR 블록)을 부여하여야 하며, VPC 생성 시 서브넷을 함께 생성하거나 물리적으로 분리된 사용자 전용 하드웨어에서 VPC를 생성할 수 있는 테넌시 기능 등이 제공됩니다.

- **(AWS 관리 콘솔)** VPC는 '콘솔 홈' → 'VPC' → 'VPC' 에서 생성 또는 CLI 명령으로 생성할 수 있습니다.



| 그림 2-1-1 | VPC 생성 화면

- **(AWS CLI)** API 환경을 통한 VPC 생성 → '--cidr-block' 변수로 VPC에 부여할 IPv4 대역을 설정하여 생성할 수 있습니다.

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16 \
      --tag-specification ResourceType=vpc,Tags=[{Key=Name,Value=MyVpc}]
```

Output:

```
{
   "Vpc": {
      "CidrBlock": "10.0.0.0/16",
      "DhcpOptionsId": "dopt-5EXAMPLE",
      "State": "pending",
      "VpcId": "vpc-0a60eb65b4EXAMPLE",
      "OwnerId": "123456789012",
      "InstanceTenancy": "default",
      "Ipv6CidrBlockAssociationSet": [],
      "CidrBlockAssociationSet": [
         {
            "AssociationId": "vpc-cidr-assoc-07501b79ecEXAMPLE",
…(생략)…
}
```

| 그림 2-1-2 | CLI 명령 – VPC 생성(IPv4)

- 듀얼 스택 VPC를 생성하려면 Amazon에서 제공하는 IPv6 CIDR 블록을 추가하여 생성할 수 있습니다.

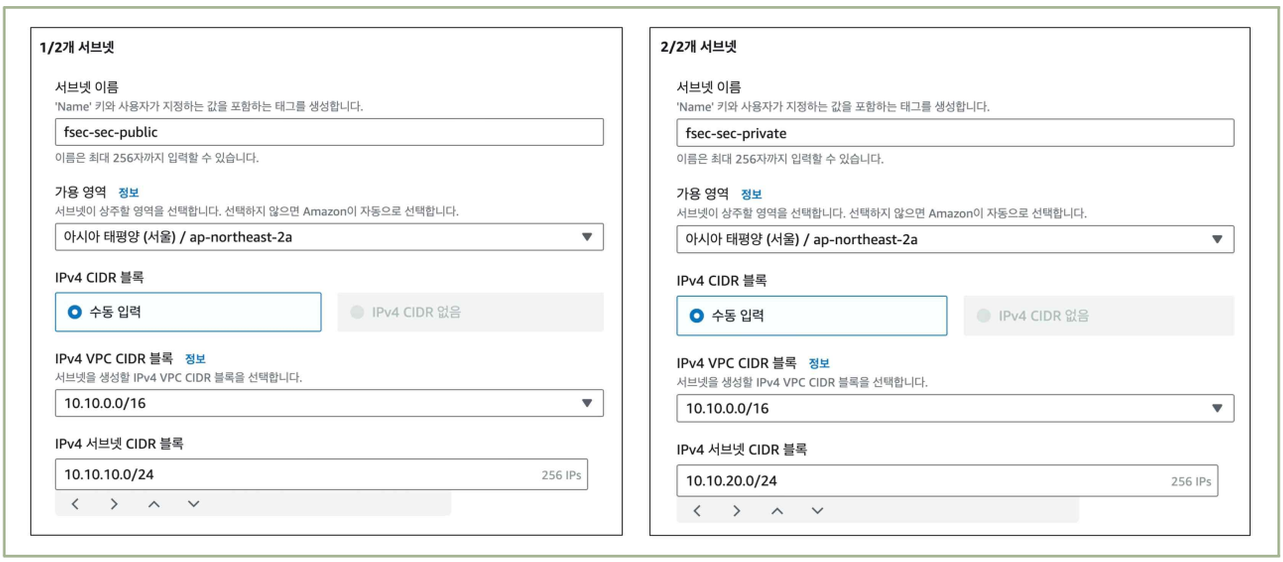
```
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --amazon-provided-ipv6-cidr-block

Output:

{
    "Vpc": {
        "CidrBlock": "10.0.0.0/16",
        "DhcpOptionsId": "dopt-dEXAMPLE",
        "State": "pending",
        "VpcId": "vpc-0fc5e3406bEXAMPLE",
        "OwnerId": "123456789012",
        "InstanceTenancy": "default",
        "Ipv6CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-068432c60bEXAMPLE",
…(생략)…
}
```

| 그림 2-1-3 | CLI 명령 - VPC 생성(듀얼 스택)

(2) 서브넷 생성

- 서브넷은 VPC 내에서 분리된 네트워크 영역이며, 서브넷마다 서로 다른 라우팅 테이블을 적용할 수 있어 네트워크 트래픽을 격리할 수 있습니다.
- 서브넷에는 EC2 인스턴스, RDS 데이터베이스 등 AWS 리소스를 생성할 수 있으며, 서브넷의 IP대역은 VPC 생성 시 지정된 CIDR IP대역 범위 내에서 지정할 수 있습니다.
- 서브넷은 라우팅을 구성하는 방법에 따라, 퍼블릭, 프라이빗, VPN전용, 격리형으로 구분하여 구성이 가능합니다.
- 인터넷 통신이 필요한 상황에서는 퍼블릭 서브넷으로(예시. 웹 서버 등), 인터넷이 격리된 내부 리소스들과 통신이 필요한 상황에서는 프라이빗 서브넷으로(예시. WAS, DB 등) 구성하여 외부 통신과 격리된 형태로 사용할 수 있습니다.

- **(AWS 관리 콘솔)** 서브넷은 '콘솔 홈' → 'VPC' → '서브넷' 에서 생성 또는 CLI 명령으로 생성할 수 있음



| 그림 2-1-4 | 퍼블릭 및 프라이빗 서브넷 생성

- **(AWS CLI)** API 환경을 통한 서브넷 생성 → '--vpc-id' 변수로 서브넷을 생성할 VPC를 지정하고 '--cidr-block' 변수로 생성할 서브넷에 부여할 IPv4 대역을 설정합니다.

```
aws ec2 create-subnet --vpc-id vpc-081ec835f3EXAMPLE --cidr-block 10.0.0.0/24 \
   --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv4-only-subnet}]
```

Output:

```
{
   "Subnet": {
      "AvailabilityZone": "us-west-2a",
      "AvailabilityZoneId": "usw2-az2",
      "AvailableIpAddressCount": 251,
      "CidrBlock": "10.0.0.0/24",
      "DefaultForAz": false,
      "MapPublicIpOnLaunch": false,
      "State": "available",
      "SubnetId": "subnet-0e99b93155EXAMPLE",
…(생략)…
   }
}
```

| 그림 2-1-5 | CLI 명령 - 서브넷 생성(IPv4)

- 듀얼 스택 서브넷 또는 IPv6 전용 서브넷 생성은 다음과 같은 CLI 명령을 사용할 수 있습니다.

```
aws ec2 create-subnet --vpc-id vpc-081ec835f3EXAMPLE --cidr-block 10.0.0.0/24 \
   --ipv6-cidr-block 2600:1f16:cfe:3660::/64 \
   --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv4-ipv6-subnet}]
```

| 그림 2-1-6 | CLI 명령 - 서브넷 생성(듀얼 스택)

```
aws ec2 create-subnet --vpc-id vpc-081ec835f3EXAMPLE --ipv6-native \
   --ipv6-cidr-block 2600:1f16:115:200::/64 \
   --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv6-only-subnet}]
```

| 그림 2-1-7 | CLI 명령 - 서브넷 생성(IPv6 전용)

(3) 라우팅 테이블 생성 및 연결

- 라우팅 테이블은 서브넷 간 네트워크 경로를 제어합니다. 기본 라우팅 테이블은 VPC가 생성될 때 필수로 생성이 되고 기본 라우팅 테이블은 다른 라우팅 테이블과 명시적으로 연결되지 않은 모든 서브넷의 라우팅을 제어하는 역할을 합니다.
- 라우팅 테이블은 서브넷과 명시적으로 연결하여 서브넷의 네트워크 경로를 제어할 수 있어 서브넷 간 통신, 리소스 간 통신을 제어할 수 있습니다.
- VPC 생성 시 서브넷을 같이 생성하지 않고 이후 수동으로 서브넷을 생성할 경우 서브넷에는 VPC 기본 라우팅 테이블이 연결되어서 퍼블릭 서브넷, 프라이빗 서브넷과 같은 구성을 위해서는 서브넷에 연결할 라우팅 테이블을 생성하여야 합니다.

- **(AWS 관리 콘솔)** 라우팅 테이블은 '콘솔 홈' → 'VPC' → '라우팅 테이블' 에서 생성할 수 있습니다.

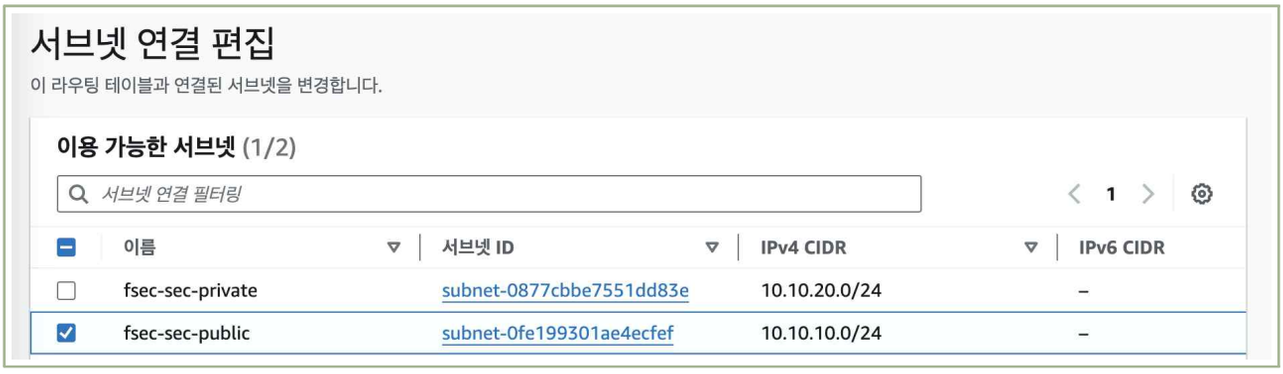
| 그림 2-1-8 | 라우팅 테이블 생성

- **(AWS CLI)** API 환경을 통한 라우팅 테이블 생성 → '--vpc-id' 변수로 서브넷을 생성할 VPC를 지정하고 '--cidr-block' 변수로 생성할 서브넷에 부여할 IPv4 대역을 설정합니다.

```
aws ec2 create-route-table --vpc-id vpc-a01106c2

Output:

{
    "RouteTable": {
        "Associations": [],
        "RouteTableId": "rtb-22574640",
        "VpcId": "vpc-a01106c2",
        "PropagatingVgws": [],
        "Tags": [],
        "Routes": [
            {
                "GatewayId": "local",
                "DestinationCidrBlock": "10.0.0.0/16",
…(생략)…
}
```

| 그림 2-1-9 | CLI 명령 - 라우팅 테이블 생성

- 생성한 라우팅 테이블을 서브넷에 연결하면 연결된 라우팅 테이블 정책에 따른 네트워크 통제가 가능합니다.



| 그림 2-1-10 | 라우팅 테이블과 서브넷 연결

```
$ aws ec2 associate-route-table --route-table-id [RTB-ID] --subnet-id [Subnet-ID]
```

```
$ aws ec2 associate-route-table --route-table-id rtb-01[redacted] --subnet-id subnet-08[redacted]
{
    "AssociationId": "rtbassoc-06[redacted]",
    "AssociationState": {
        "State": "associated"
    }
}
```

| 그림 2-1-11 | CLI 명령 - 라우팅 테이블과 서브넷 연결

(4) 라우팅 테이블 정책 설정 (인터넷 게이트웨이 및 NAT 게이트웨이 설정)

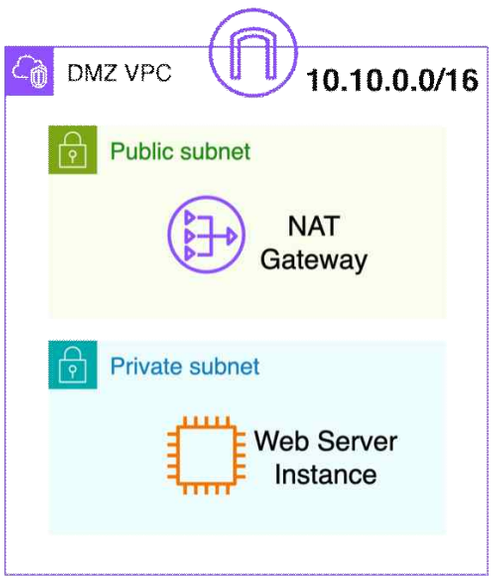
- 라우팅 테이블에 정책이 없는 네트워크 통신은 기본 라우팅 테이블의 정책을 이용하게 되는데, 만약 기본 라우팅 테이블에도 정책이 없을 경우 해당 통신은 실패하게 됨에 따라 보안성과 가용성을 고려한 라우팅 테이블의 정책을 설정해주어야 합니다.
- 퍼블릭 서브넷, 프라이빗 서브넷 구성으로 내외부 통신을 분리하는 경우 퍼블릭 서브넷 라우팅 테이블 정책으로 인터넷 게이트웨이를 추가하여 Ingress(외부→내부 통신) 라우팅으로 설정해주고 프라이빗 서브넷 라우팅 테이블 정책에는 NAT 게이트웨이를 추가하여 Egress 라우팅으로 설정하여 안전한 네트워크 통신을 구성할 수 있습니다.
- 관련 네트워크 보안 통제를 위한 기본적인 라우팅 테이블 정책 예시는 다음과 같습니다.

■ 퍼블릭 서브넷

Destination	Target
[Subnet CIDR]	Local
0.0.0.0/0	[Internet Gateway-ID]

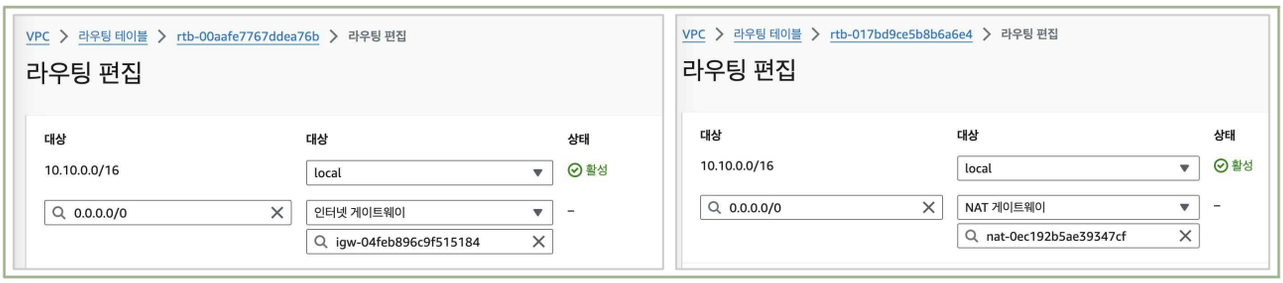
■ 프라이빗 서브넷

Destination	Target
[Subnet CIDR]	Local
0.0.0.0/0	[NAT Gateway-ID]



- **(AWS 관리 콘솔)** 라우팅 테이블 정책은 '콘솔 홈' → 'VPC' → '라우팅 테이블' 에서 대상라우팅 테이블을 선택하고 '라우팅' 탭에서 '라우팅 편집'을 클릭하여 라우팅 정책을 변경할 수 있습니다.

[라우팅 테이블 수정]



- **(AWS CLI)** API 환경을 통한 VPC에 인터넷 게이트웨이 연결 → '--internet-gateway-id' 변수로 인터넷 게이트웨이 ID를 지정하고 '--vpc-id' 변수로 인터넷 게이트웨이를 연결할 VPC를 지정하여 라우팅 테이블을 변경합니다.

```
aws ec2 attach-internet-gateway \
   --internet-gateway-id igw-0d0fb496b3EXAMPLE \
   --vpc-id vpc-0a60eb65b4EXAMPLE
```

Output:

```
none
```

| 그림 2-1-12 | CLI 명령 - 인터넷 게이트웨이 연결

- **(AWS CLI)** API 환경을 통한 내부에서 외부로 인터넷 통신을 위한 NAT 게이트웨이 연결 → '—subnet-id' 변수로 NAT 게이트웨이를 설정할 서브넷을 지정하고 '—allocation-id' 변수로 대상 서브넷에 연결할 NAT 게이트웨이를 지정합니다.

```
aws ec2 create-nat-gateway \
   --subnet-id subnet-0250c25a1fEXAMPLE \
   --allocation-id eipalloc-09ad461b0dEXAMPLE
```

Output:

```
{
   "NatGateway": {
      "CreateTime": "2021-12-01T22:22:38.000Z",
      "NatGatewayAddresses": [
         {
            "AllocationId": "eipalloc-09ad461b0dEXAMPLE"
         }
      ],
      "NatGatewayId": "nat-0c61bf8a12EXAMPLE",
      "State": "pending",
      "SubnetId": "subnet-0250c25a1fEXAMPLE",
      "VpcId": "vpc-0a60eb65b4EXAMPLE",
      "ConnectivityType": "public"
   }
}
```

| 그림 2-1-13 | CLI 명령 - NAT 게이트웨이 생성 및 연결

- **(AWS CLI)** API 환경을 통한 0.0.0.0/0 트래픽의 인터넷 게이트웨이로 통신 라우팅 경로를 생성하는 라우팅 테이블 수정 → '—route-table-id' 변수로 라우팅 정책을 생성할 라우팅 테이블 ID를 지정하고, '—destination—cidr-block' 변수로 목적지 IPv4를 지정하며, 어떤 인터넷 게이트웨이 ID로 트래픽을 전송할 것인지 '—gateway-id' 변수로 지정합니다.

```
aws ec2 create-route --route-table-id rtb-22574640 \
    --destination-cidr-block 0.0.0.0/0 \
    --gateway-id igw-c0a643a9
```

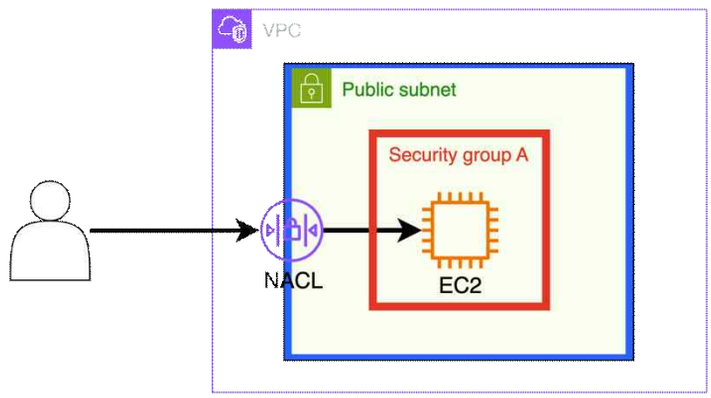
Output:

```
{
    "Return" : true
}
```

| 그림 2-1-14 | CLI 명령 - 라우팅 테이블 수정

2) 보안 그룹(Security group) 또는 NACL(Network ACL) 등의 기능을 통한 네트워크 구성 및 통제(인/아웃바운드 통제 등)

- 네트워크 접근 통제를 위한 보안 기능으로 보안 그룹(Security Group)과 네트워크ACL(Network ACL, 이하 NACL)을 이용할 수 있으며, 이 기능을 이용하여 VPC에 대한 네트워크 접근 통제 환경을 구성할 수 있습니다.



| 그림 2-1-15 | 네트워크 통제 예시 아키텍처

- 보안 그룹은 연결된 리소스(인스턴스) 수준에서 Inbound 및 Outbound 트래픽을 제어하는 접근통제 기능으로 인스턴스와 보안 그룹은 1:1, N:1, 1:N로 구성이 가능합니다.
- NACL은 서브넷 수준에서 Inbound 및 Outbound 트래픽을 제어하는 접근통제 기능으로 서브넷과 NACL은 1:1, N:1 로 구성이 가능합니다.

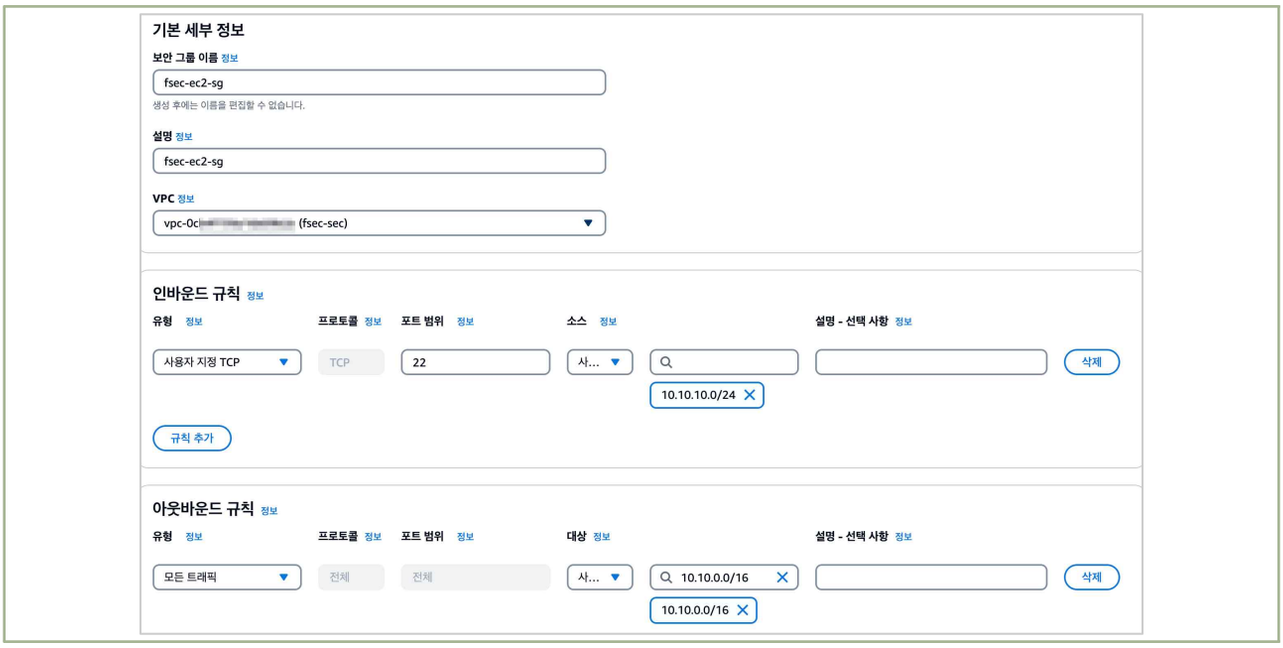
- 보안 그룹과 NACL의 차이점 비교는 아래와 같습니다.

보안 그룹	NACL
Stateful 방식	Stateless 방식
허용(Allow) 규칙만 생성	허용(Allow)와 거부(Deny) 생성
모든 규칙을 적용	규칙 번호 순서로 적용

(1) 보안 그룹 생성

- 보안 그룹은 화이트리스트(White-list)이며, Inbound와 Outbound 규칙을 나누어서 보안 통제를 적용할 수 있습니다.

- **(AWS 관리 콘솔)** 보안 그룹은 '콘솔 홈' → 'EC2' → '보안 그룹' 에서 생성할 수 있습니다.



| 그림 2-1-16 | 보안 그룹 생성

- **(AWS CLI)** API 환경을 통한 보안 그룹 생성 → '—group-name' 변수로 보안 그룹명을 지정하여 보안 그룹을 생성하며, '—description' 변수로 보안 그룹에 대한 설명을 명시할 수 있습니다.

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My security group"
```

Output:

```
{
  "GroupId": "sg-903004f8"
}
```

| 그림 2-1-17 | CLI 명령 - 보안 그룹 생성

- **(AWS CLI)** API 환경을 통한 보안 그룹에 Inbound 규칙 추가 → '--group-id' 변수로 보안 그룹명을 지정하고 '--protocol' 변수로 프토토콜 설정, '--cidr' 및 '--port' 로 통신을 허용할 IP 및 포트를 설정합니다.

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --protocol tcp --port 22 --cidr 203.0.113.0/24
```

Output:

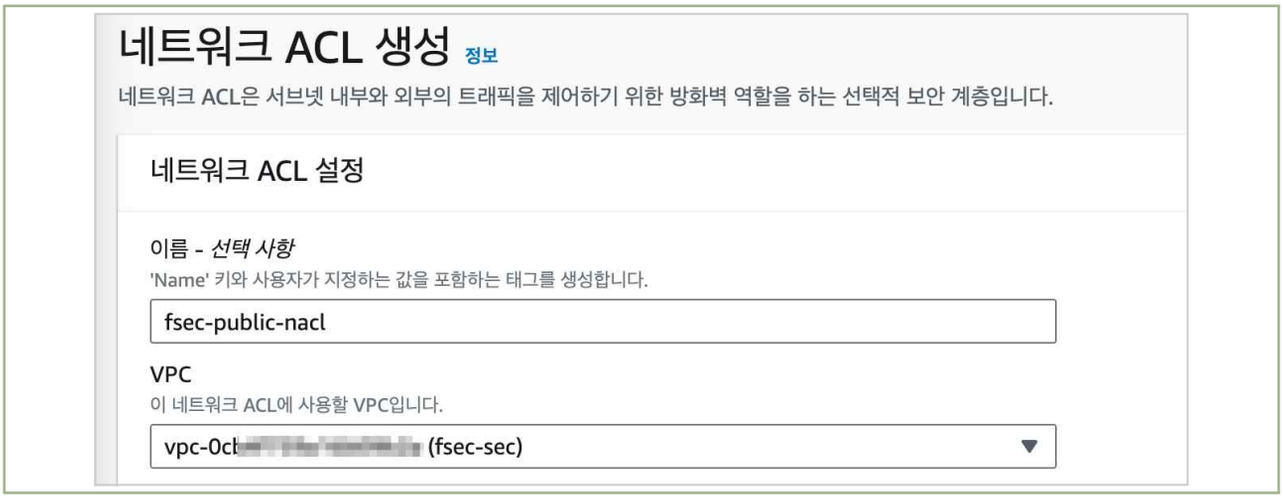
```
{
    "GroupId": "sg-903004f8"
}
```

| 그림 2-1-18 | CLI 명령 - 보안 그룹 규칙 추가

(2) NACL 생성

- NACL의 규칙 생성은 화이트리스트(White-list) 또는 블랙리스트(Black-list)로 가능하며, Inbound와 Outbound 규칙을 생성할 수 있습니다.
- Stateless 방식이므로 허용할 규칙은 Inbound와 Outbound 모두 지정해야 요청 트래픽에 대한 반환 트래픽이 정상적으로 전송됩니다.

- **(AWS 관리 콘솔)** NACL은 '콘솔 홈' → 'VPC' → '네트워크 ACL' 에서 생성할 수 있습니다.



| 그림 2-1-19 | NACL 생성

- **(AWS CLI)** API 환경을 통한 NACL 생성 → '--vpc-id' 변수로 NACL을 생성할 VPC를 지정한 후 NACL을 생성할 수 있습니다.

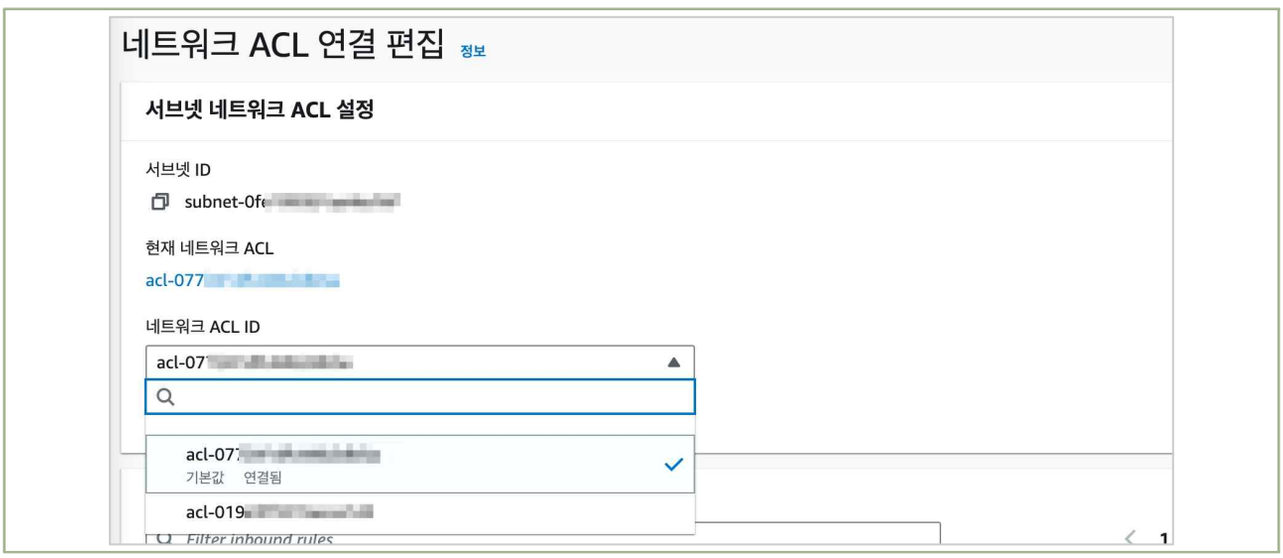
```
aws ec2 create-network-acl --vpc-id vpc-a01106c2

Output:

{
    "NetworkAcl": {
        "Associations": [],
        "NetworkAclId": "acl-5fb85d36",
        "VpcId": "vpc-a01106c2",
        "Tags": [],
        "Entries": [
            {
                "CidrBlock": "0.0.0.0/0",
                "RuleNumber": 32767,
                "Protocol": "-1",
                "Egress": true,
                "RuleAction": "deny"
            },
…(생략)…
}
```

| 그림 2-1-20 | CLI 명령 - NACL 생성

- 서브넷 생성시 기본 NACL이 연결되어 있으며, 신규로 생성한 NACL를 이용할 시 NACL 연결 변경 설정을 해야 합니다.
- AWS CLI로 연결 설정을 변경할 시 NACL 연결ID를 파라미터 값으로 사용해야 함에 따라 NACL 연결ID 확인이 필요합니다.

○ **(AWS 관리 콘솔)** 특정 서브넷에 NACL 연결은 두가지 방법이 있습니다.

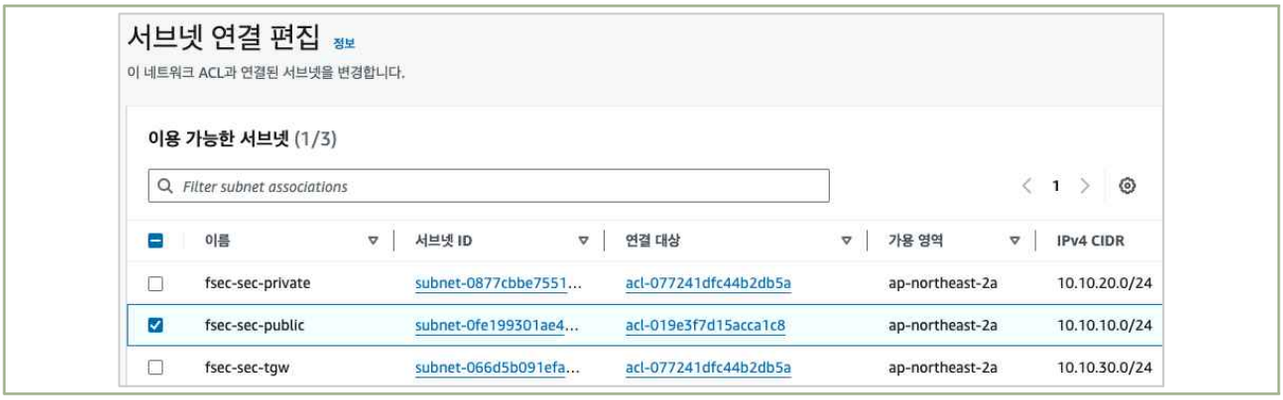
가. '콘솔 홈' → 'VPC' → '네트워크 ACL' 에서 NACL을 선택한 후 '서브넷 연결' → '서브넷 연결 편집' 에서 NACL에 연결할 서브넷을 선택할 수 있습니다.



| 그림 2-1-21 | 서브넷에서 연결할 NACL 선택

나. 콘솔 홈' → 'VPC' → '서브넷' 에서 서브넷을 선택한 후 '네트워크 ACL' → '네트워크 ACL 연결 편집' 에서 서브넷에 연결할 NACL을 선택할 수 있습니다.

[NACL 연결 변경]



| 그림 2-1-22 | NACL에서 연결할 서브넷 선택

- **(AWS CLI)** API 환경을 통한 특정 서브넷에 NACL 연결 → '--association-id' 변수로 NACL 연결 ID를 지정하고, '—network-acl-id' 변수로 NACL ID를 지정하여 NACL을 연결합니다.

```
aws ec2 replace-network-acl-association --association-id aclassoc-e5b95c8c \
   --network-acl-id acl-5fb85d36
```

Output:

```
{
   "NewAssociationId": "aclassoc-3999875b"
}
```

| 그림 2-1-23 | CLI 명령 - 서브넷에 NACL 연결

- 참고로 NACL의 association-id(NACL 연결 ID)를 확인하여야 하며, 해당 값을 확인하기 위해 다음과 같은 AWS CLI 명령어를 사용할 수 있습니다.

```
aws ec2 describe-network-acls \
  -query "NetworkAcls[].Associations[?SubnetId==subnet-0931fc2fa5EXAMPLE"
```

Output:

```
[
   {
      "NetworkAclAssociationId": "aclassoc-0c1679dc41EXAMPLE",
      "NetworkAclId": "acl-0ea1f54ca7EXAMPLE",
      "SubnetId": "subnet-0931fc2fa5EXAMPLE"
   }
]
```

| 그림 2-1-24 | CLI 명령 - NACL 연결 ID 확인

4 참고 사항

1 기준

식별번호	기준	내용
2.2.	내부망 네트워크 보안 통제	클라우드 환경 내 내부망 구성 시 보안 통제 방안을 수립하고 적용하여야 한다.

2 설명

- 클라우드 환경 내 내부망을 구성하는 경우 외부 침입, 비인가 접근 등으로 보호될 수 있도록 보안 통제 방안을 수립하고 적용하여야 한다.

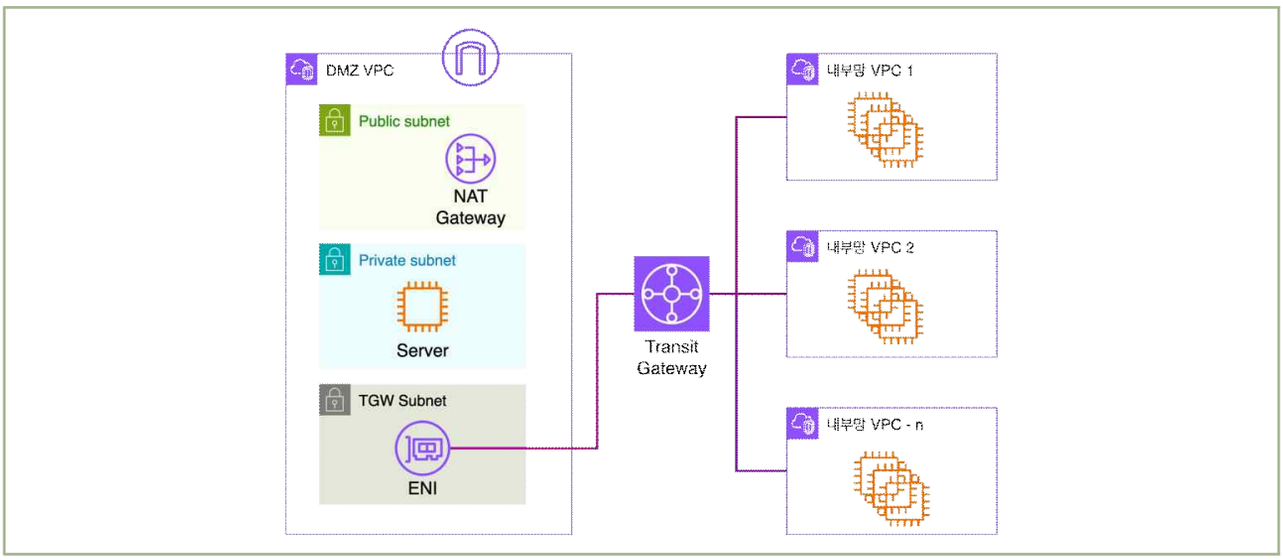
  - 예시

    1) VPC 등 네트워크 관련 기능을 통한 네트워크 접근 통제(인터넷망 등)
    2) 보안 그룹(Security group) 또는 NACL(Network ACL) 등의 기능을 통한 네트워크 구성(인/아웃바운드 통제 등)
    3) 내부망으로 구현한 가상자원(서버, 데이터베이스 등)에 공인IP 미 할당
    4) 방화벽 서비스를 통한 IP 통제

3 우수 사례

1) VPC 등 네트워크 관련 기능을 통한 네트워크 접근 통제(인터넷망 등)

- VPC는 프라이빗한 가상 네트워크로 각 VPC는 서로 논리적으로 분리되어, 다른 사용자 또는 동일 사용자의 타 VPC와는 기본적으로 격리되어 있습니다.
- 인터넷망과 통신하는 VPC(이하 DMZ VPC)와 VPC간 통신 및 On-Premise 정보처리시스템과 통신을 하는 VPC(이하 내부망 VPC)을 분리하고 VPC간 통신은 AWS Transit 게이트웨이로 연결하는 Multi VPC 네트워크 환경을 아래와 같이 구성하여 운영할 수 있습니다.



| 그림 2-2-1 | Transit 게이트웨이를 이용한 네트워크 통제 구성

- 위 그림에서 인터넷 구간과 접점은 DMZ VPC에서 통신하게 되고 내부망 통신은 각각의 내부망 VPC에서 통신을 하게 됩니다.
- 위와 같은 네트워크 접근 통제의 주요 요소는 VPC, 서브넷, Transit 게이트웨이 이며, VPC 및 서브넷 생성은 “2.1-1)” 을 참조하시기 바랍니다.

(1) Transit 게이트웨이 생성

- **(AWS 관리 콘솔)** Transit 게이트웨이는 ‘콘솔 홈’ → ’VPC’ → ’Transit Gateway’ 에서 생성 할 수 있습니다.

| 그림 2-2-2 | Transit 게이트웨이 생성

- 각 옵션에 대한 자세한 설명은 https://docs.aws.amazon.com/ko_kr/vpc/latest/tgw/tgw-transit-gateways.html#create-tgw 을 참조하시기 바랍니다.

- **(AWS CLI)** API 환경을 통한 Transit 게이트웨이 생성 → '--description' 변수로 설명을 명시하고, '—option' 변수로 Amazon 측 ASN 등을 설정하여 생성할 수 있습니다. (참고로 Transit 게이트웨이는 변수 없이 생성도 가능함)

```
aws ec2 create-transit-gateway \
    --description MyTGW \
    --options
AmazonSideAsn=64516,AutoAcceptSharedAttachments=enable,DefaultRouteTableAssociation=enable,
DefaultRouteTablePropagation=enable,VpnEcmpSupport=enable,DnsSupport=enable
```

Output:

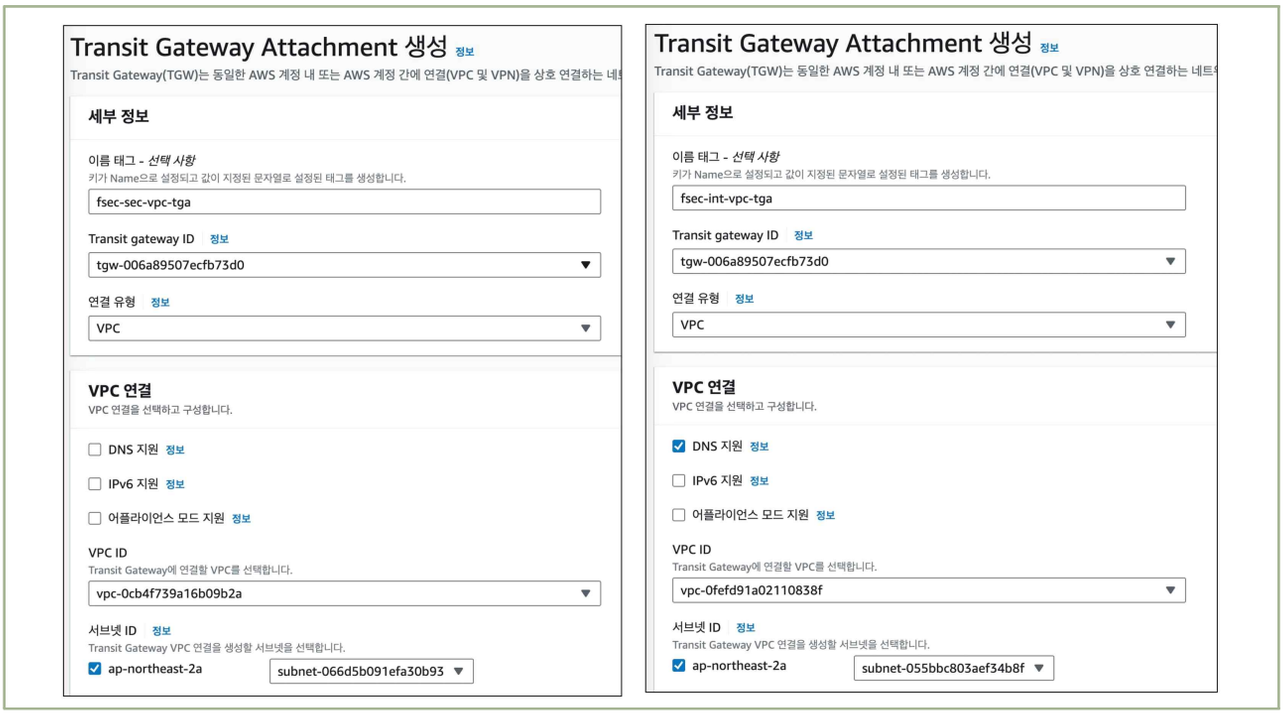
```
{
    "TransitGateway": {
        "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
        "TransitGatewayArn": "arn:aws:ec2:us-east-2:111122223333:transit-gateway/tgw-
0262a0e521EXAMPLE",
        "State": "pending",
        "OwnerId": "111122223333",
        "Description": "MyTGW",
        "CreationTime": "2019-07-10T14:02:12.000Z",

…(생략)…
}
```

| 그림 2-2-3 | CLI 명령 - Transit 게이트웨이 생성

- Transit 게이트웨이를 이용하기 위해서는 VPC 연결이 필요하며, 특정 서브넷을 지정하여 Attachment를 생성하면 해당 서브넷에 Transit 게이트웨이와 통신하는 ENI가 생성됩니다. (생성된 ENI는 '콘솔 홈' → 'EC2' → '네트워크 인터페이스' 에서 확인 가능)

(2) Transit 게이트웨이 Attachment 생성

- **(AWS 관리 콘솔)** Transit 게이트웨이 Attachment와 Transit 게이트웨이 라우팅 테이블은 '콘솔 홈' → 'VPC' → 'Transit Gateway 연결' 에서 생성할 수 있습니다.



| 그림 2-2-4 | Transit 게이트웨이 Attachment 생성

- **(AWS CLI)** API 환경을 통한 Transit 게이트웨이 Attachment 생성 → '—transit-gateway-id' 변수로 Transit 게이트웨이 ID를 지정하고 '—vpc-id' 변수로 Transit 게이트웨이를 설치할 VPC를 지정하며, '--subnet-id' 변수로 Transit 게이트웨이를 설치할 서브넷을 지정하여 Attachment를 생성할 수 있습니다.

[CLI 명령 - Transit 게이트웨이 Attachment 생성]

```
aws ec2 create-transit-gateway-vpc-attachment \
    --transit-gateway-id tgw-0262a0e521EXAMPLE \
    --vpc-id vpc-07e8ffd50f49335df \
    --subnet-id subnet-0752213d59EXAMPLE
```

Output:

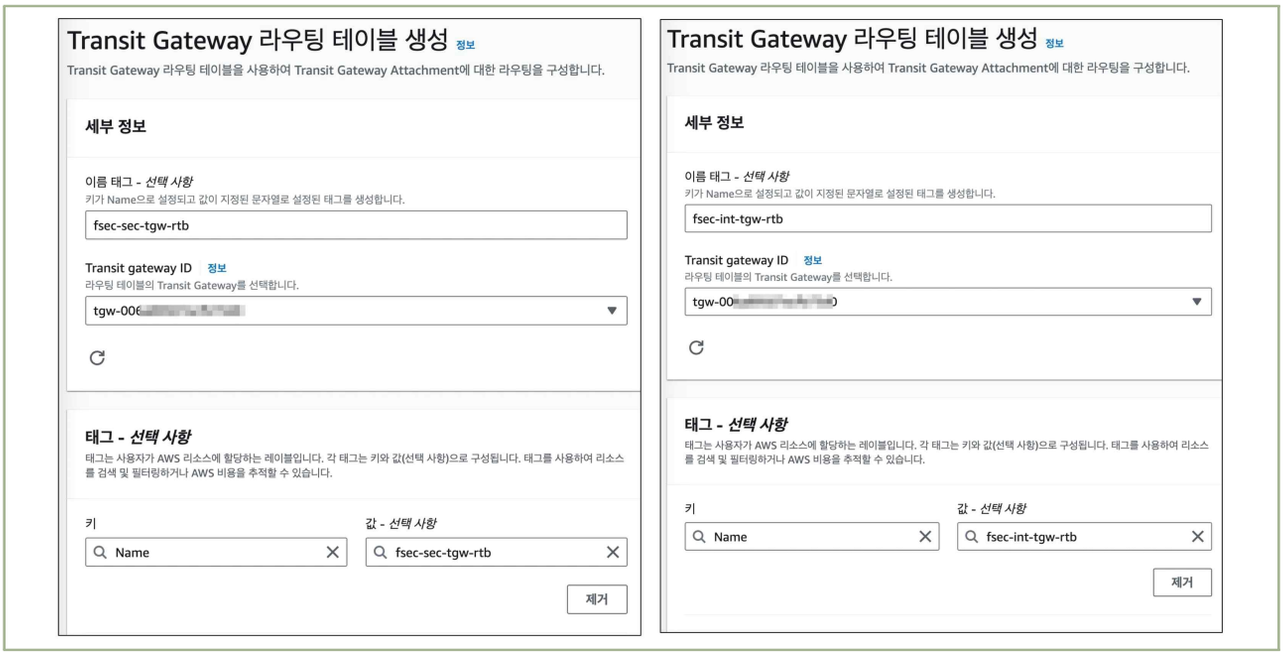
```
{
    "TransitGatewayVpcAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",
        "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
        "VpcId": "vpc-07e8ffd50fEXAMPLE",
        "VpcOwnerId": "111122223333",
        "State": "pending",
        "SubnetIds": [
            "subnet-0752213d59EXAMPLE"
        ],
…(생략)…
}
```

| 그림 2-2-5 | CLI 명령 - Transit 게이트웨이 Attachment 생성

(3) Transit 게이트웨이 라우팅 테이블 생성

- VPC와 Transit 게이트웨이를 연결한 후 VPC간 통신을 할 수 있도록 Transit 게이트웨이 라우팅 테이블을 생성합니다.

- **(AWS 관리 콘솔)** Transit 게이트웨이 라우팅 테이블은 Transit 게이트웨이 라우팅 테이블은 '콘솔 홈' → 'VPC' → 'Transit Gateway 라우팅 테이블' 에서 생성할 수 있습니다.



| 그림 2-2-6 | Transit 게이트웨이 라우팅 테이블 생성

- **(AWS CLI)** API 환경을 통한 Transit 게이트웨이 라우팅 테이블 생성 → '--transit-gateway-id 변수로 Transit 게이트웨이 ID를 지정합니다.

```
aws ec2 create-transit-gateway-route-table \
    --transit-gateway-id tgw-0262a0e521EXAMPLE
```

Output:

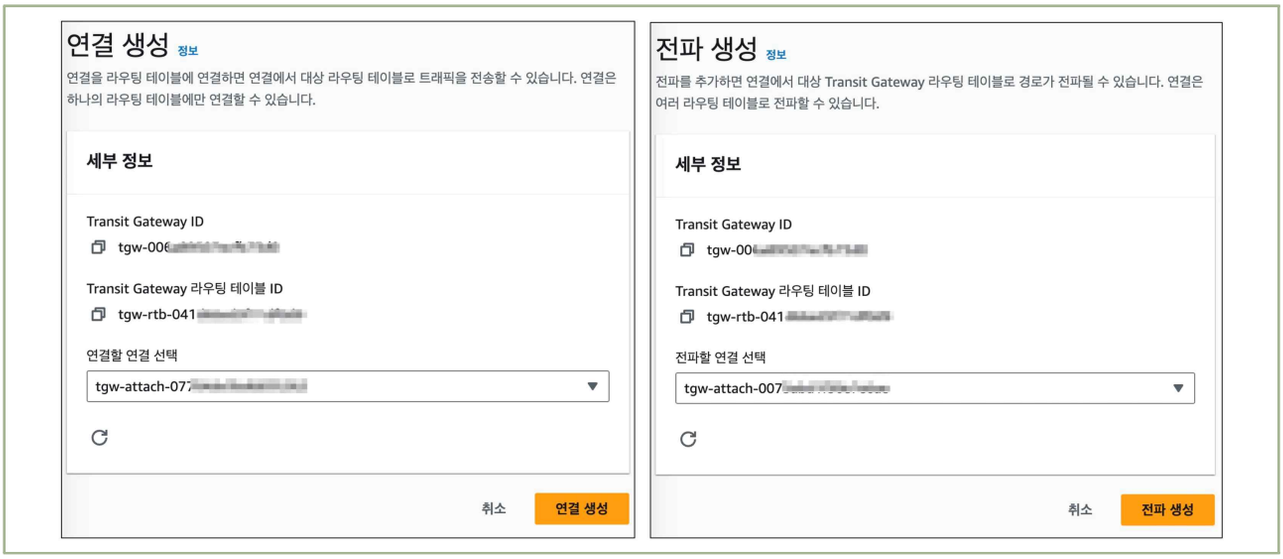
```
{
    "TransitGatewayRouteTable": {
        "TransitGatewayRouteTableId": "tgw-rtb-0960981be7EXAMPLE",
        "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
        "State": "pending",
        "DefaultAssociationRouteTable": false,
        "DefaultPropagationRouteTable": false,
        "CreationTime": "2019-07-10T19:01:46.000Z"
    }
}
```

| 그림 2-2-7 | CLI 명령 - Transit 게이트웨이 라우팅 테이블 생성

(4) Transit 게이트웨이 라우팅 테이블 연결 및 전파

- Transit 게이트웨이 라우팅 테이블과 VPC 간 연결 설정을 Associations(연결)으로 연결하며, 트래픽을 전송할 타겟(Transit 게이트웨이 Attachment)을 Propagations (전파)로 설정하여 라우팅 테이블을 구성합니다.

- **(AWS 관리 콘솔)** Transit 게이트웨이 라우팅 테이블 연결 및 전파는 '콘솔 홈' → 'VPC' → 'Transit Gateway 라우팅 테이블' 에서 대상 라우팅 테이블을 클릭하고 '연결' → '연결 생성' 및 '전파' → '전파 생성' 으로 생성할 수 있습니다.



| 그림 2-2-8 | Transit 게이트웨이 라우팅 테이블 연결(Associations) 및 전파(Propagations)

- **(AWS CLI)** API 환경을 통한 Transit 게이트웨이 라우팅 테이블 연결 및 전파 → '--transit-gateway-id 변수로 Transit 게이트웨이 ID를 지정하여 Transit 게이트웨이에 라우팅 테이블을 연결하고 전파할 수 있습니다.

```
aws ec2 associate-transit-gateway-route-table \
   --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE \
   --transit-gateway-attachment-id tgw-attach-0b5968d3b6EXAMPLE
```

Output:

```
{
   "Association": {
      "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
      "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
      "ResourceId": "vpc-0065acced4EXAMPLE",
      "ResourceType": "vpc",
      "State": "associating"
   }
}
```

| 그림 2-2-9 | CLI 명령 - Transit 게이트웨이 라우팅 테이블 연결(Associations)

```
aws ec2 enable-transit-gateway-route-table-propagation \
   --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \
   --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE
```

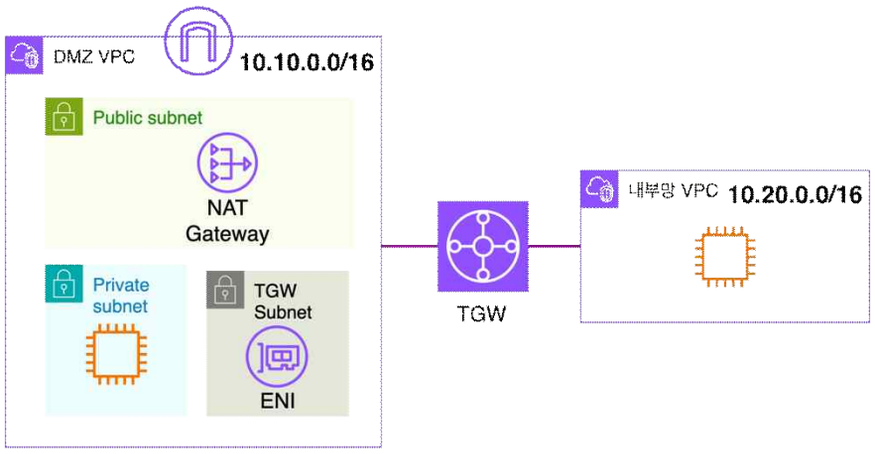
Output:

```
{
   "Propagation": {
      "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
      "ResourceId": "vpc-4d7de228",
      "ResourceType": "vpc",
      "TransitGatewayRouteTableId": "tgw-rtb-0a823edbdeEXAMPLE",
      "State": "disabled"
   }
}
```

| 그림 2-2-10 | CLI 명령 - Transit 게이트웨이 라우팅 테이블 전파(Propagations)

(5) Transit 게이트웨이 구성 확인

- Transit 게이트웨이 구성이 완료되면 아래와 같이 2개의 VPC 간 통신을 할 수 있도록 네트워크 연결 상태가 됩니다.



| 그림 2-2-11 | Transit 게이트웨이를 구성하여 VPC간 통신 연결 예시 아키텍처

- DMZ VPC와 내부망 VPC가 상호 정상적으로 통신하기 위해서는 서브넷의 라우팅 테이블에 목적지 CIDR에 대한 정책 추가가 필요하며, 예시는 아래와 같습니다.
- 서브넷 라우팅 테이블 정책 추가는 “3.1-1)”의 ‘라우팅 테이블 생성 및 연결’ 을 참조 바랍니다.

DMZ VPC 서브넷		내부망 VPC 서브넷	
Destination	Target	Destination	Target
10.20.0.0/16	[TGW-ID]	10.10.0.0/16	[TGW-ID]

- 설정이 완료된 후 DMZ VPC과 내부망 VPC가 정상적으로 통신하는지 확인합니다. (아래 예시는 EC2 인스턴스 to EC2 인스턴스로 연결 확인)

```
sh-5.2$ ip addr show enX0 | grep -oP 'inet \K\S+'
10.10.20.179/24
sh-5.2$ sudo ssh -i "server-key.pem" ec2-user@10.20.142.10
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Thu Mar 28 08:33:19 2024 from 10.10.20.179
[ec2-user@ip-10-20-142-10 ~]$ ip addr show enX0 | grep -oP 'inet \K\S+'
10.20.142.10/20
[ec2-user@ip-10-20-142-10 ~]$
```

| 그림 2-2-12 | EC2인스턴스를 이용하여 VPC 정상 연결 확인

2) 보안 그룹 또는 NACL 등의 기능을 통한 네트워크 구성 (Inbound/ Outbound 통제 등)

- 보안 그룹을 이용한 네트워크 접근 통제는 “2.1-2)” 의 ‘보안 그룹 생성’ 을 참조 바랍니다.
- NACL을 이용한 네트워크 접근 통제는 “2.1-2)” 의 ‘NACL 생성’ 을 참조 바랍니다.

3) 내부망으로 구현한 가상자원(서버, 데이터베이스 등)에 공인IP 미할당

(1) EC2 퍼블릭 IP 통제

- 퍼블릭IP(공인IP)를 가상자원에 할당은 최초 생성시 또는 가상자원 생성 후 EIP(Elastic IP)를 연결하는 방법이 있습니다.
- EC2인스턴스의 퍼블릭 IP 할당 통제는 AWS IAM 에서 Policy를 생성하고 해당 Policy를 IAM Role을 이용하여 할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "*",
            "Condition": { "Bool": { "ec2:AssociatePublicIpAddress": "true" } }
        } ]
}
```

| 그림 2-2-13 | EC2 생성 시 퍼블릭IP 할당을 차단하는 IAM Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ec2:AssociateAddress",
            "Resource": "*"
        } ]
}
```

| 그림 2-2-14 | EC2 생성 후 퍼블릭IP 할당을 차단하는 IAM Policy

- 생성된 IAM Policy를 적용하면 퍼블릭 IP 할당을 통제할 수 있습니다.

[EC2에 퍼블릭 IP 할당 제한]

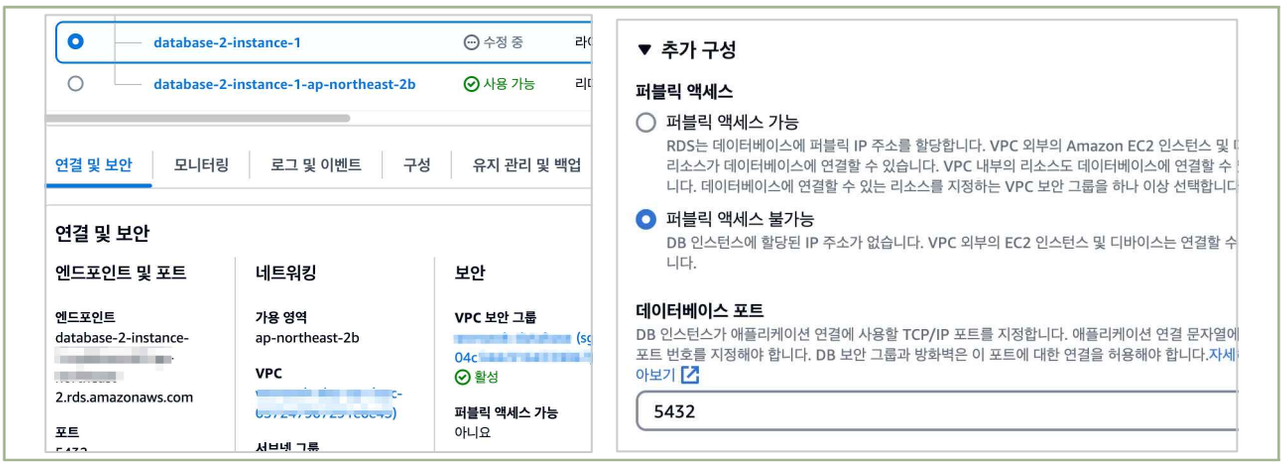
| 그림 2-2-15 | EC2에 퍼블릭 IP 할당시 차단 화면

(2) RDS 퍼블릭 IP 통제

- Amazon RDS는 생성 시 '퍼블릭 액세스' 기능을 활성화하여 RDS 엔드포인트에 퍼블릭 IP를 할당할 수 있으며, 퍼블릭 액세스 옵션을 퍼블릭 IP 할당을 통제할 수 있습니다.

- **(AWS 관리 콘솔)** RDS 데이터베이스 속성은 '콘솔 홈' → 'RDS' → '데이터베이스' 에서 DB인스턴스를 선택하여 수정할 수 있습니다.

[RDS엔드포인트 퍼블릭 IP 할당 제한]



- **(AWS CLI)** API 환경을 통한 RDS 엔드포인트 퍼블릭 IP 제한 → '--db-instance-identifier' 변수로 RDS 인스턴스 명을 지정하고, '--no-publicly-accessible' 변수로 해당 인스턴스에 퍼블릭 IP를 제한할 수 있습니다.

```
aws rds modify-db-instance \
   --db-instance-identifier dbName \
   --no-publicly-accessible
```

Output:

```
null
```

| 그림 2-2-16 | CLI - RDS 엔드포인트 퍼블릭 IP 할당 제한

- 퍼블릭 액세스 설정 정보는 CLI명령으로 확인이 가능하며, 출력 결과를 확인하고 퍼블릭 액세스 설정을 할 수 있습니다.

```
aws rds describe-db-instances \
   --query "DBInstances[*].[DBInstanceIdentifier, PubliclyAccessible]"
```

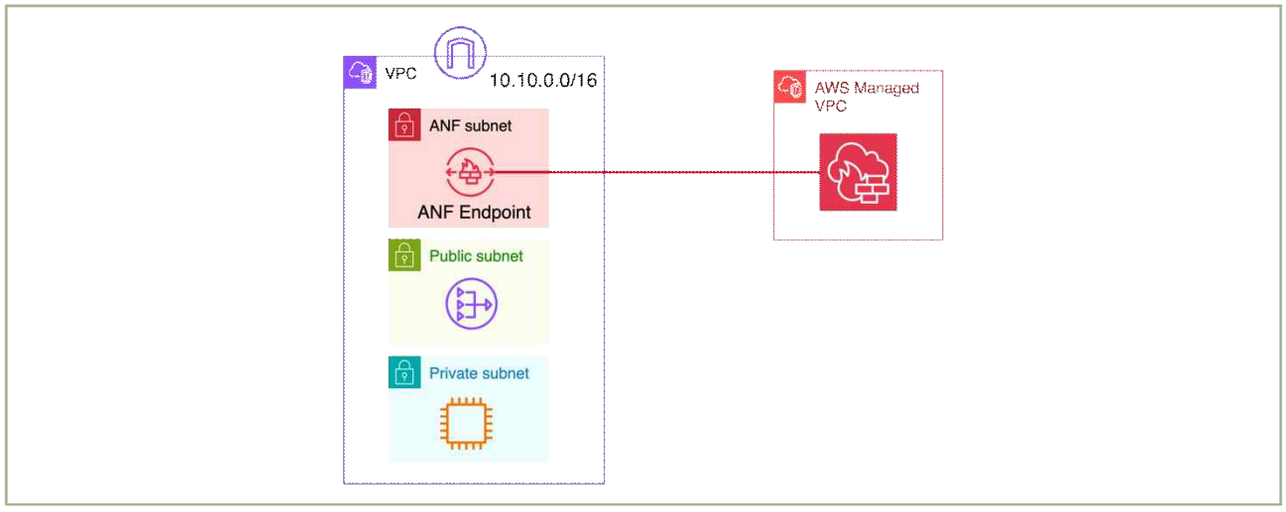
Output:

```
[
   [
      "dbname",
      true
   ]
   ,
   [
      "dbname-ap-northeaste-2a"
      false
   ]
]
```

| 그림 2-2-17 | CLI - RDS 엔드포인트 퍼블릭 IP 할당 상태 확인

4) 방화벽 서비스를 통한 IP통제

- AWS Network Firewall은 VPC에서 생성하여 네트워크를 보호하는VPC 경계에서 트래픽을 필터링할 수 있습니다.
- Stateful Rule Engine과 Stateless Rule Engine 을 제공하며, Stateful Rule Group은 5-Tuple(IP통제), Domain List(도메인 통제), Suricata Compatible IPS Rules (IDS/IPS) 를 제공합니다.
- AWS Network Firewall은 관리형 방화벽 서비스로 GWLB(Gateway Load Balancer) 엔드포인트를 활용하는 구성의 서비스 입니다.
- AWS Network Firewall로 네트워크를 통제할 때 VPC 라우터 정책을 이용하여 GWLB 엔드포인트로 네트워크 트래픽을 전달하여 트래픽 검사를 하는 아키텍처로 구성하는 것이 일반적입니다.



| 그림 2-2-18 | AWS Network Firewall 기본 구조

(1) AWS Network Firewall 생성

- AWS Network Firewall(이하 '방화벽') 생성은 VPC의 서브넷 내에 엔드포인트를 배포하고 방화벽에서 사용할 정책(Policy)를 적용하는 순서로 진행되며, 방화벽 정책에는 Rule Group을 적용하여 네트워크 통제를 수행하게 됩니다.
- CLI로 방화벽 생성 시에는 방화벽 정책을 먼저 생성하고 방화벽 정책의 ARN 정보를 알고 있어야 합니다.

(AWS 관리 콘솔) 방화벽 생성은 '콘솔 홈' → 'VPC' → '방화벽' 에서 생성할 수 있습니다.

| 그림 2-2-19 | 방화벽 엔드포인트를 배포할 서브넷 지정

| 그림 2-2-20 | 방화벽 정책 생성 및 연결

- **(AWS CLI)** API 환경을 통한 방화벽 생성 → '--firewall-name' 변수로 방화벽명을 설정하고, '--firewall-policy-arn' 으로 어떤 방화벽 정책을 설정할 것인지 지정하며, '--vpc-id' 변수 및 '--subnet-mappings' 변수로 방화벽 엔드포인트를 위치할 VPC와 서브넷을 지정하여 방화벽을 생성할 수 있습니다.

```
aws network-firewall create-firewall \
  --firewall-name fsec-fw1 \
  --firewall-policy-arn arn:aws:network-firewall:ap-northeast-2:123456789012:firewall-policy/fsec-fw-
policy \
  --vpc-id vpc-0cb4f739a1EXAMPLE \
  --subnet-mappings SubnetId=subnet-0fe199301aEXAMPLE
```

Output:

```
{
    Firewall": {
        "FirewallName": "fsec-fw1",
        "FirewallArn": "arn:aws:network-firewall:ap-northeast-2:123456789012:firewall-policy/fsec-fw-
policy",
        "VpcId": "vpc-0cb4f739a1EXAMPLE",
        "SubnetMappings": [
            {
                "SubnetId": "subnet-0fe199301aEXAMPLE"
            }
…(생략)…
}
```

| 그림 2-2-21 | 방화벽 생성

- 참고로 방화벽 정책에 대한 ARN 확인은 아래의 AWS CLI 명령으로 확인할 수 있습니다.

```
aws network-firewall describe-firewall-policy \
  --firewall-policy-name fsec-fw-policy \
  --query "FirewallPolicyResponse.[FirewallPolicyName, FirewallPolicyArn]"
```

Output:

```
{
    Firewall": {
        "FirewallName": "fsec-fw1",
        "FirewallArn": "arn:aws:network-firewall:ap-northeast-2:123456789012:firewall-policy/fsec-fw-
policy",
        "VpcId": "vpc-0cb4f739a1EXAMPLE",
        "SubnetMappings": [
            {
                "SubnetId": "subnet-0fe199301aEXAMPLE"
            }
…(생략)…
}
```

| 그림 2-2-22 | 방화벽 정책 ARN 확인

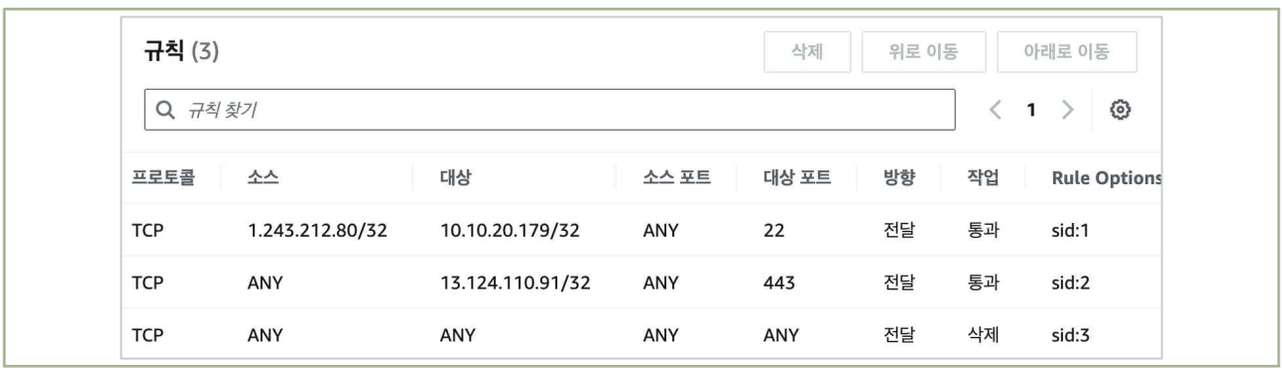
(2) 방화벽 규칙 생성 및 적용

- 방화벽 정책 중 '규칙 평가 순서' 는 Stateful Rule Engine에 적용되는 규칙 적용 방식으로 엄격한 순서(Strict Order)와 작업 순서(Action Order)가 있습니다.

- 엄격한 순서(Strict Order)는 Rule Group에 구성된 규칙 순서대로 평가를 하는 방식으로 일반적인 방화벽 규칙 평가 방식 입니다.
- 반면, 작업 순서(Action Order)는 통과(Pass) 〉 삭제(Drop) 〉 거부(Reject) 〉 알림(Alert) 규칙 순서대로 평가하는 방식을 말합니다.
- 방화벽 규칙은 방화벽 Rule Group 내에 생성하며 Rule Group에 할당한 용량(규칙 수) 한도 내로 생성이 가능합니다.

(AWS 관리 콘솔)

- 방화벽 Rule Group(규칙 그룹)은 '콘솔 홈' → 'VPC' → 'Network Firewall 규칙 그룹' 에서 생성할 수 있습니다.
- 방화벽 Rule Group 생성 시 여러 옵션이 있으나, 본 가이드에서는 다음의 옵션으로 Rule Group을 생성하는 것으로 합니다.
  - 규칙 그룹 유형 : 상태 저장 규칙 그룹 (Stateful Rule Group)
  - 규칙 그룹 형식 : 표준 상태 저장 규칙 (Standard stateful rules)
  - 규칙 평가 순서 : 엄격한 순서 (Strict Order)



프로토콜	소스	대상	소스 포트	대상 포트	방향	작업	Rule Options
TCP	1.243.212.80/32	10.10.20.179/32	ANY	22	전달	통과	sid:1
TCP	ANY	13.124.110.91/32	ANY	443	전달	통과	sid:2
TCP	ANY	ANY	ANY	ANY	전달	삭제	sid:3

| 그림 2-2-23 | 방화벽 Rule Group 생성

- 위 예시의 생성된 방화벽 Rule Group은 규칙 평가 순서에 따라(Strict Order) sid:1, sid:2 는 허용되고 이외의 통신은 sid:3에 의해 삭제(Drop) 됩니다.
- 생성한 방화벽 Rule Group을 방화벽 정책에 추가하여 방화벽에 규칙이 적용되도록 하여야 합니다.
- '콘솔 홈' → 'VPC' → '방화벽 정책' 에서 대상 방화벽 정책을 선택하고 '상태 저장 규칙 그룹' → '작업' → '비관리형 상태 저장 규칙 그룹 추가' 에서 추가할 방화벽 Rule Group을 선택하여 적용합니다.

| 그림 2-2-24 | 방화벽 정책에 방화벽 Rule Group 추가

- 이후 방화벽은 네트워크 트래픽을 위 예시의 생성된 방화벽 Rule Group의 룰에 따라 Pass또는 Drop 하게 됩니다.

4 참고 사항

- 본 가이드에서 다루지 않는 서비스에 대한 다양한 기능들은 아래의 서비스 설명가이드를 참고하시기 바랍니다.
  - Security Group
    https://docs.aws.amazon.com/ko_kr/vpc/latest/userguide/vpc-security-groups.html
  - Network ACL (NACL)
    https://docs.aws.amazon.com/ko_kr/vpc/latest/userguide/vpc-network-acls.html
  - AWS Network Firewall
    https://docs.aws.amazon.com/network-firewall/latest/developerguide/what-is-aws-network-firewall.html

1 기준

식별번호	기준	내용
2.3.	**네트워크 보안 관제 수행**	**클라우드 환경 내 금융회사 가상자원을 보호하기 위한 네트워크 보안 관제를 수행하여야 한다.**

2 설명

- 클라우드 환경 내 가상자원을 보호하기 위해 네트워크 보안 관제를 수행하여야 한다.

  - 예시

    1) 금융회사 보안 관제 서비스와 연동하여 관제 수행(클라우드 내 발생하는 네트워크 트래픽 연동 등을 활용)

    2) 클라우드 서비스 제공자가 제공하는 가상자원 보호를 위한 네트워크 보안관제 및 유사 기능(DDoS, WAF 등) 활용

3 우수 사례

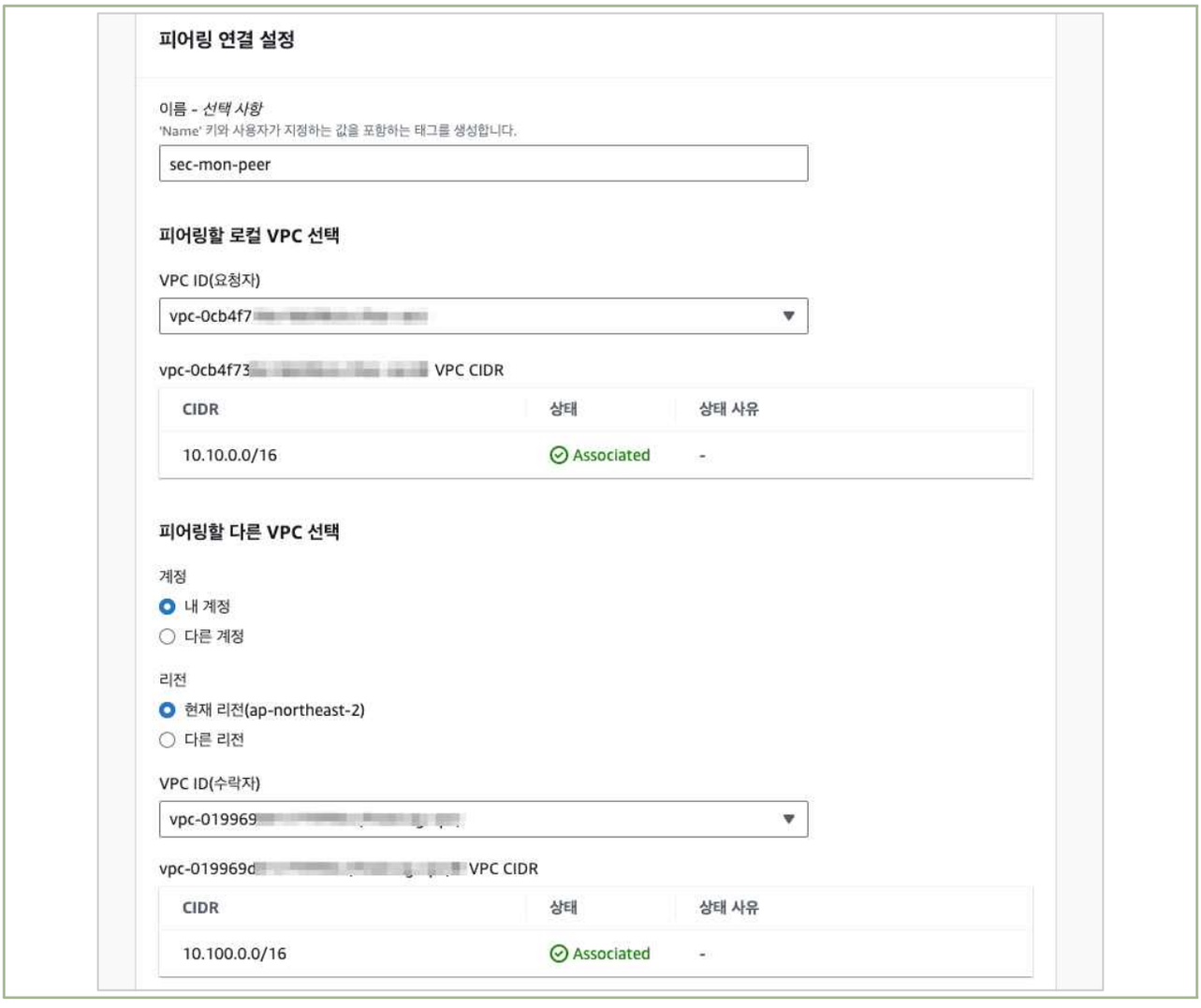
1) 금융회사 보안관제 서비스와 연동하여 관제 수행 (네트워크 트래픽 연동 등을 활용)

- AWS는 금융회사 보안 관제를 위해 네트워크 트래픽을 미러링 할 수 있는 방법으로 VPC Traffic Mirroring(이하 '트래픽 미러링') 기능을 제공합니다.
- 트래픽 미러링은 ENI(Elastic Network Interface)에서 네트워크 트래픽을 복사하는데 사용할 수 있는 기능으로, 트래픽 미러가 가능한 소스 유형은 EC2 인스턴스나 RDS 인스턴스 등이 있습니다.
- 네트워크 보안관제를 위한 아키텍처는 금융회사 보안관제 서비스 시스템의 위치, 어떠한 데이터가 보안관제 대상인지에 따라 따라 구성 방법이 달라질 수 있으며, 본 가이드는 보안관제 서비스 시스템이 AWS에서 구축되어 운영되는 상황을 예시로 설명합니다.
- 동일 VPC 내에서 트래픽 미러링은 별도의 VPC 간 연결 설정 없이 가능하지만, 보안관제 서비스 시스템이 다른 VPC에 위치할 경우 VPC Peering 또는 Transit 게이트웨이 연결이 필요하며, 다른 리전의 VPC에 위치하는 경우 VPC Peering 연결이 필요함
- 본 가이드에서는 보안관제 서비스가 동일 리전 및 다른 VPC에 위치하는 경우를 예시로 합니다.

(1) VPC Peering 연결

- 보안관제 서비스 시스템이 금융회사 계정에 설치되어 있다면 Transit 게이트웨이를 연결하여 VPC간 통신을 구성할 수 있습니다. (본 가이드 3.2-1)-(1) 의 Transit 게이트웨이 생성 부분을 참고 바랍니다.)
- VPC Peering은 논리적으로 분리된 VPC 간 트래픽을 라우팅하기 위한 네트워크 연결로 게이트웨이, VPN 등에 의존하지 않고 VPC 간 통신할 수 있는 VPC 기능 입니다.

- **(AWS 관리 콘솔)** VPC Peering은 게이트웨이는 '콘솔 홈' → 'VPC' → '피어링 연결' 에서 생성할 수 있습니다.



| 그림 2-3-1 | VPC Peering 연결 생성

- VPC Peering 연결을 생성한 후 피어링 할 다른 VPC' 소유주가 Peering을 수락해야 정상적으로 연결이 완료 됩니다.

| 그림 2-3-2 | VPC Peering 연결 요청 수락

- **(AWS CLI)** API 환경을 통한 VPC Peering 연결 생성 → '--vpc-id 변수로 로컬 VPC ID를 지정하고, '--peer-vpc-id' 변수로 연결할 대상 VPC ID를 지정하여 VPC Peering 연결을 생성할 수 있습니다.

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-id vpc-11122233

Output:

{
    "VpcPeeringConnection": {
        "Status": {
            "Message": "Initiating Request to 444455556666",
            "Code": "initiating-request"
        },
        "Tags": [],
        "RequesterVpcInfo": {
            "OwnerId": "444455556666",
            "VpcId": "vpc-1a2b3c4d",
            "CidrBlock": "10.0.0.0/28"
        },
        "VpcPeeringConnectionId": "pcx-111aaa111",
        "ExpirationTime": "2014-04-02T16:13:36.000Z",
        "AccepterVpcInfo": {
            "OwnerId": "444455556666",
            "VpcId": "vpc-11122233"
        }
    }
}
```

| 그림 2-3-3 | CLI - VPC Peering 연결 생성

(2) 라우팅 테이블에 VPC Peering 연결 추가

- 로컬 VPC와 연결 대상 VPC에 트래픽 미러링 소스(Source) 및 대상(Destination)이 될 ENI가 위치하는 서브넷 라우팅 테이블에 통신할 CIDR과 피어링 연결이 각각 필요하며, 예시는 아래와 같습니다.

구 분	로컬 VPC	연결 대상 VPC
VPC CIDR	10.10.0.0/16	10.100.0.0/16
라우팅 추가 내용	(Destination) 10.100.0.0/16 (Target) Peering Connection ID	(Destination) 10.10.0.0/16 (Target) Peering Connection ID

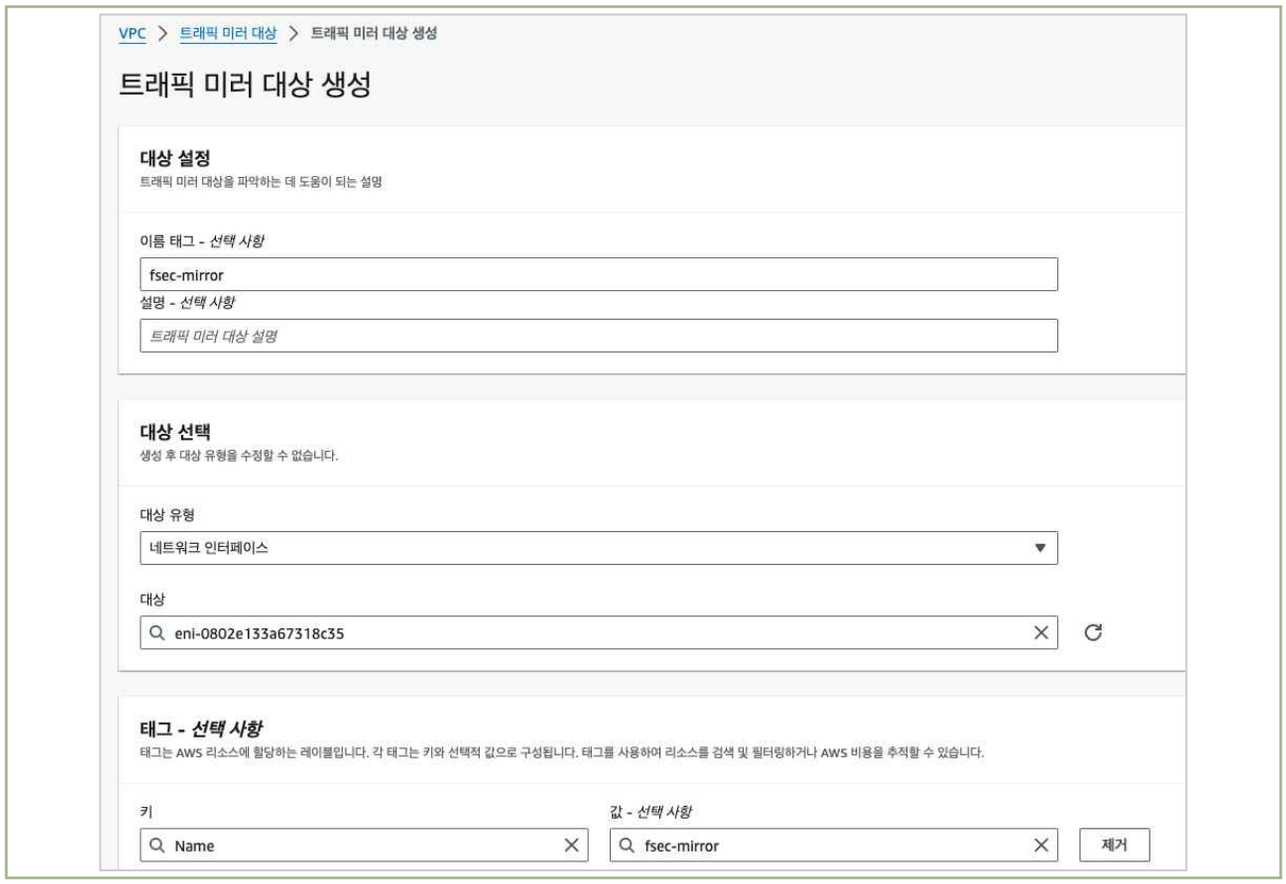
- 라우팅 테이블에 라우팅 추가 방법은 '3.1-1)-(1) 라우팅 테이블 생성 및 연결' 참조 바랍니다.

(3) 트래픽 미러링 설정

- 트래픽 미러링 설정은 '트래픽 미러링 대상' 또는 '트래픽 미러링 필터' 를 먼저 생성하고 이후 '트래픽 미러링 세션' 을 생성합니다.
- 그리고 트래픽 미러링 소스와 대상에 트래픽 미러링에 대한 통신이 원활하도록 보안 그룹 설정을 확인하여야 합니다.

가. 트래픽 미러 대상 생성

- **(AWS 관리 콘솔)** 트래픽 미러 대상은 '콘솔 홈' → 'VPC' → '미러 대상' 에서 생성할 수 있습니다.

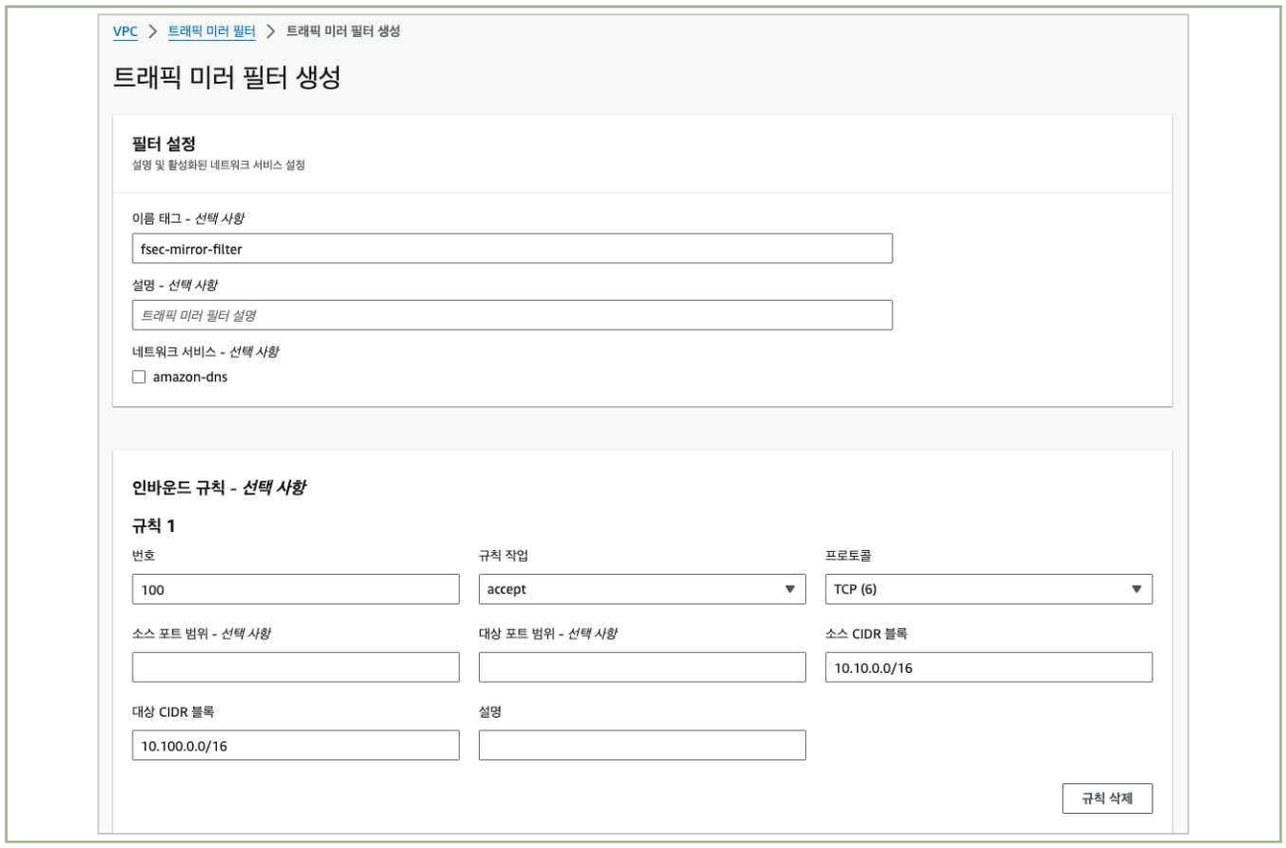


| 그림 2-3-4 | 트래픽 미러링 대상 생성

- 트래픽 미러 대상은 트래픽 미러링을 수신할 대상(Target) 유형(네트워크 인터페이스, NLB, GWLB)을 선택하고, 유형에 대항하는 ENI를 선택하여 설정 합니다.

나. 트래픽 미러 필터 생성

- **(AWS 관리 콘솔)** 트래픽 미러 필터는 '콘솔 홈' → 'VPC' → '미러 필터' 에서 생성할 수 있습니다.

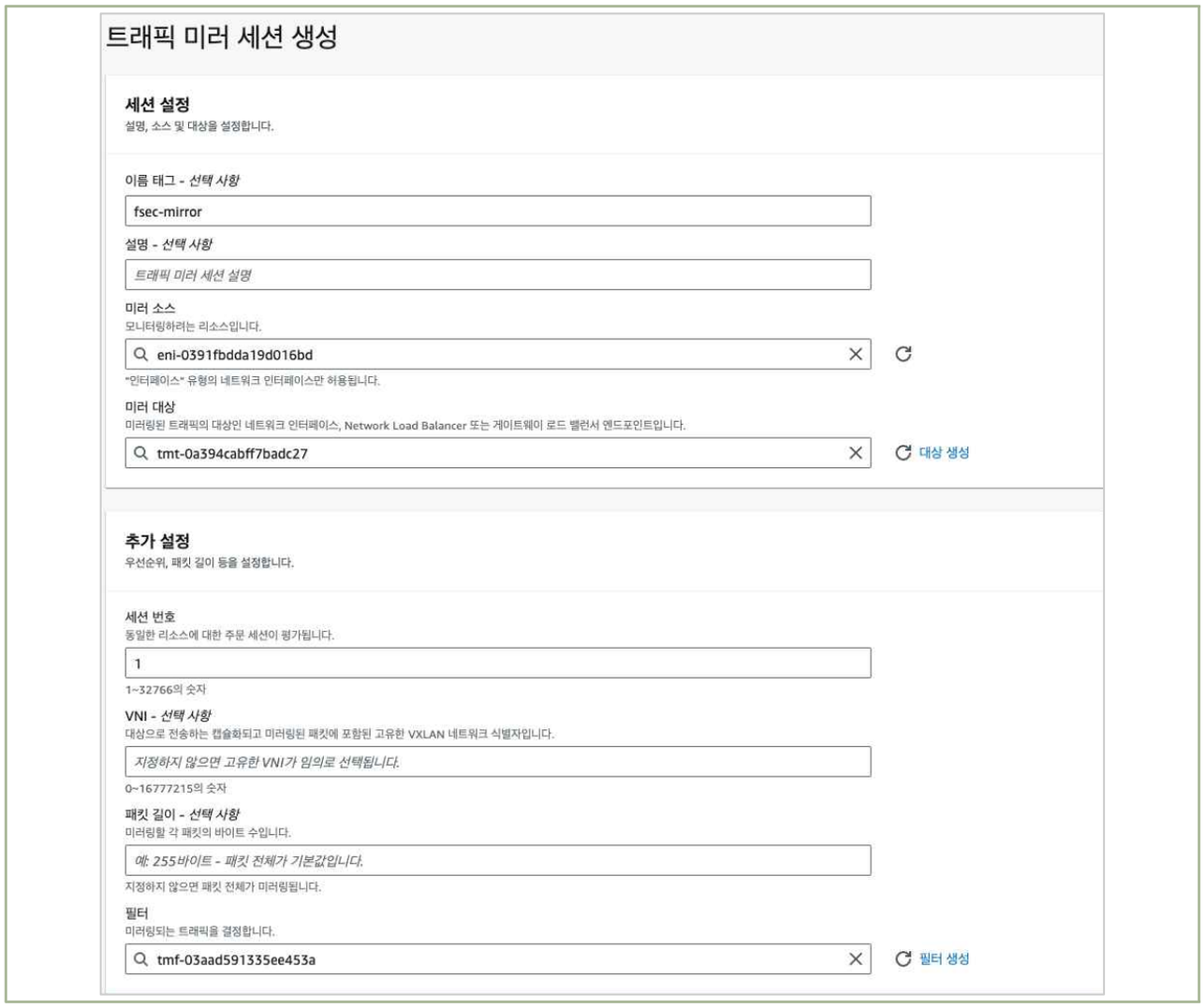


| 그림 2-3-5 | 트래픽 미러링 필터 생성

- 트래픽 미러 필터는 Inbound/Outbound 규칙을 설정하고 이후 트래픽 미러 세션 생성 시 트래픽 필터를 설정하면 네트워크 접근통제가 적용됩니다.

다. 트래픽 미러 세션 생성

■ **(AWS 관리 콘솔)** 트래픽 미러 세션은 '콘솔 홈' → 'VPC' → '미러 세션' 에서 생성 할 수 있습니다.



| 그림 2-3-6 | 트래픽 미러링 필터 생성

■ 트래픽 미러 세션 생성시 다음과 같은 설정을 하여야 합니다. (단, 명시되지 않은 옵션은 선택 사항)

- 미러 소스 : 트래픽 미러링 출발지 ENI
- 미러 대상 : 생성한 트래픽 미러 대상
- 세션 번호 : 우선순위
- 필터 : 생성한 트래픽 미러 필터

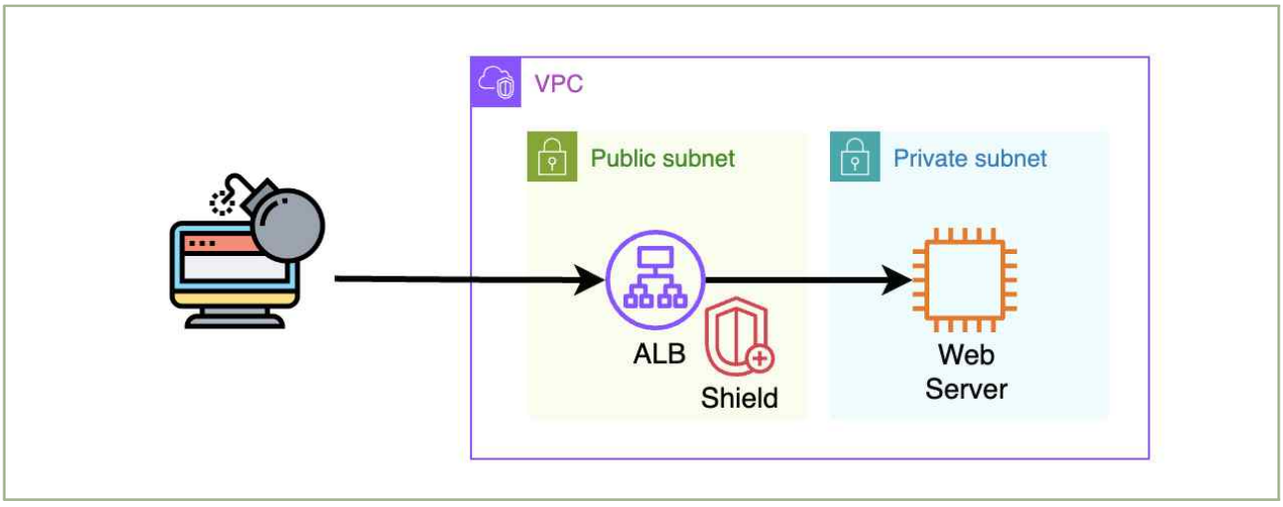
라. 트래픽 미러 소스/대상 ENI에 대한 보안 그룹 설정 (인스턴스 ENI 기준)

- 원활한 트래픽 미러링을 위해서 인스턴스에 적용되어 있는 보안 그룹에 대한 규칙에서 트래픽 허용 여부를 확인합니다.
- 보안 그룹 설정은 '3.1-2)-(1) 보안 그룹 생성' 을 참고 바랍니다.

2) 클라우드 서비스 제공자가 제공하는 가상자원 보호를 위한 네트워크 보안관제 및 유사 기능(DDoS, WAF 등) 활용

(1) AWS Shield (DDoS 공격 대응)

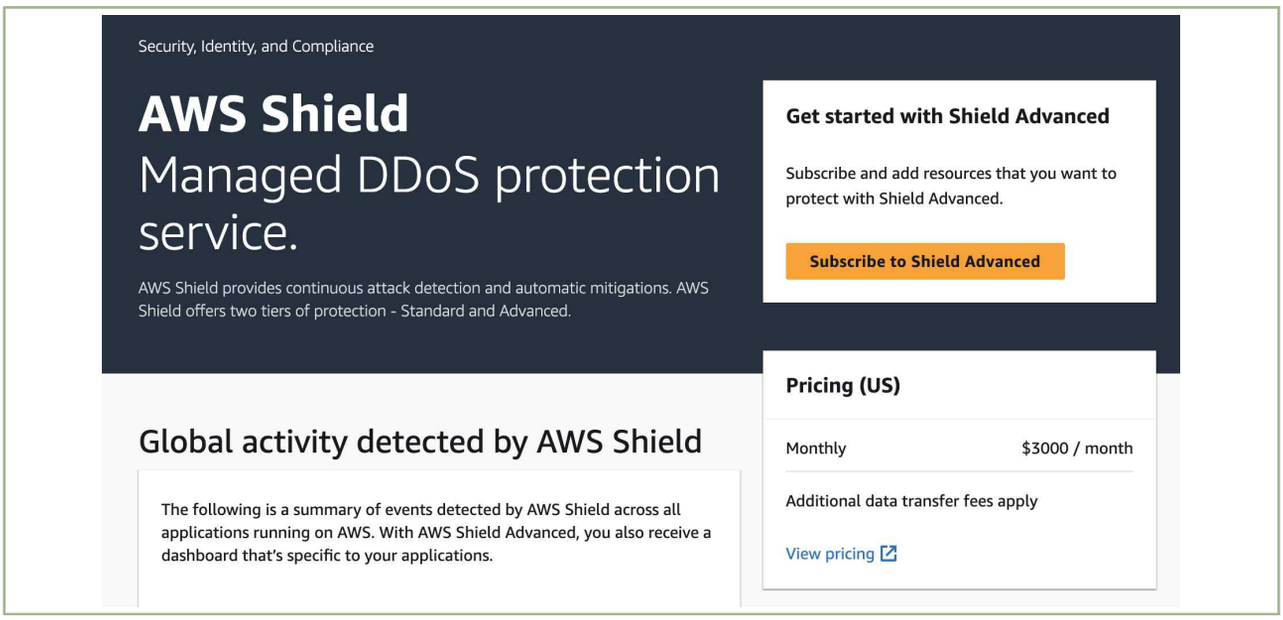
- DDoS(분산 서비스 거부)공격으로부터 금융회사의 가상자원을 보호하기 위해 AWS Shield 서비스를 사용할 수 있습니다.
- 모든 AWS 고객은 추가 비용 없이 Shield Standard의 보호 기능을 이용할 수 있습니다. (별도의 설정 없이 자동 적용됨)
- Shield Advanced는 DDoS공격, Volumetric bots, 취약점 악용 시도와 같은 외부 위협으로부터 애플리케이션을 보호하는 관리형 서비스입니다. (DDoS 공격으로부터 더 높은 수준의 보호를 위해서는 Shield Advanced를 구독하여 이용할 수 있습니다.)
- 본 참고서에서는 Shield Advanced를 사용하여 웹 서비스를 보호하는 구성을 예시로 설명합니다.



| 그림 2-3-7 | Shield Advanced를 사용한 DDoS 공격 대응 구성 예시

(AWS 관리 콘솔)

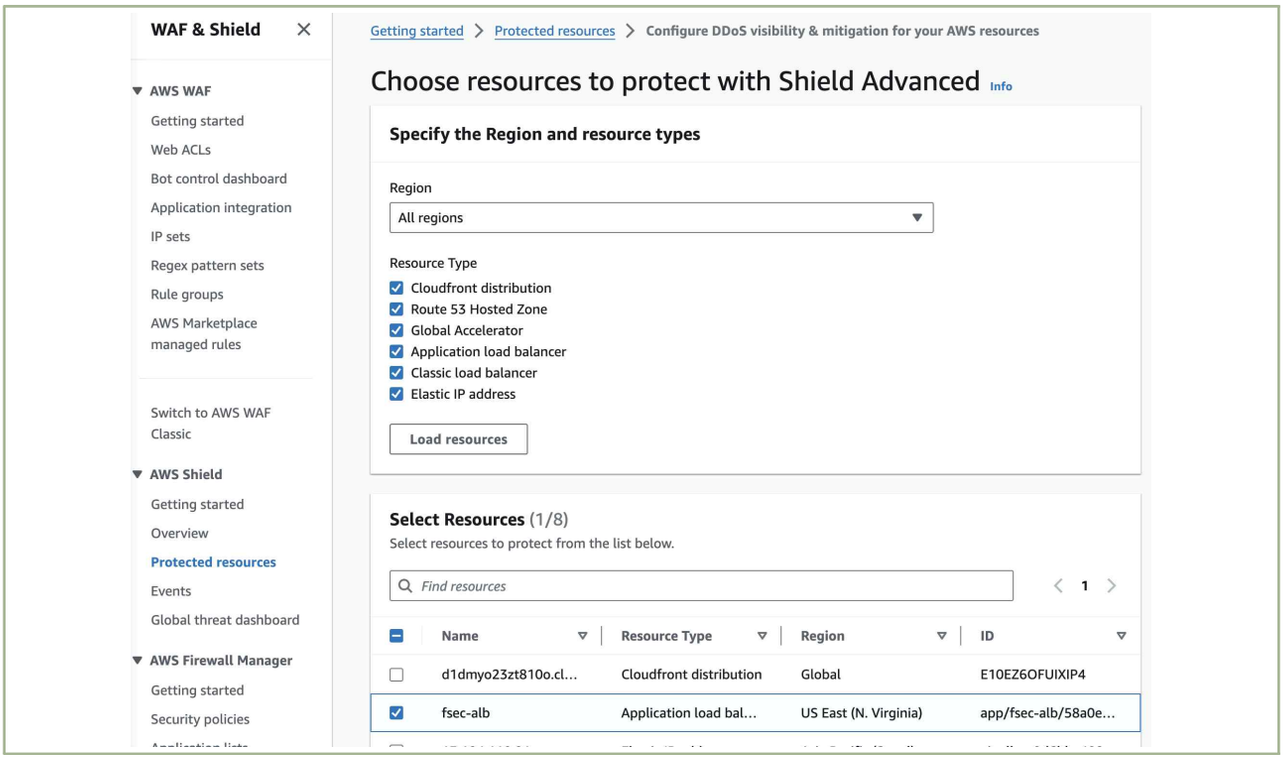
(1)-1. Shield Advanced는 '콘솔 홈' → 'WAF & Shield' → 'AWS Shield' → 'Get started' 에서 구독할 수 있습니다.



| 그림 2-3-8 | AWS Shield Advanced 구독 화면

- Shield Advanced는 사용자가 지정한 리소스에 대해서만 보호하게 됩니다. 따라서 보호할 가상자원 또는 가상자원과 연결된 리소스를 선택하여 보호 대상으로 등록합니다.

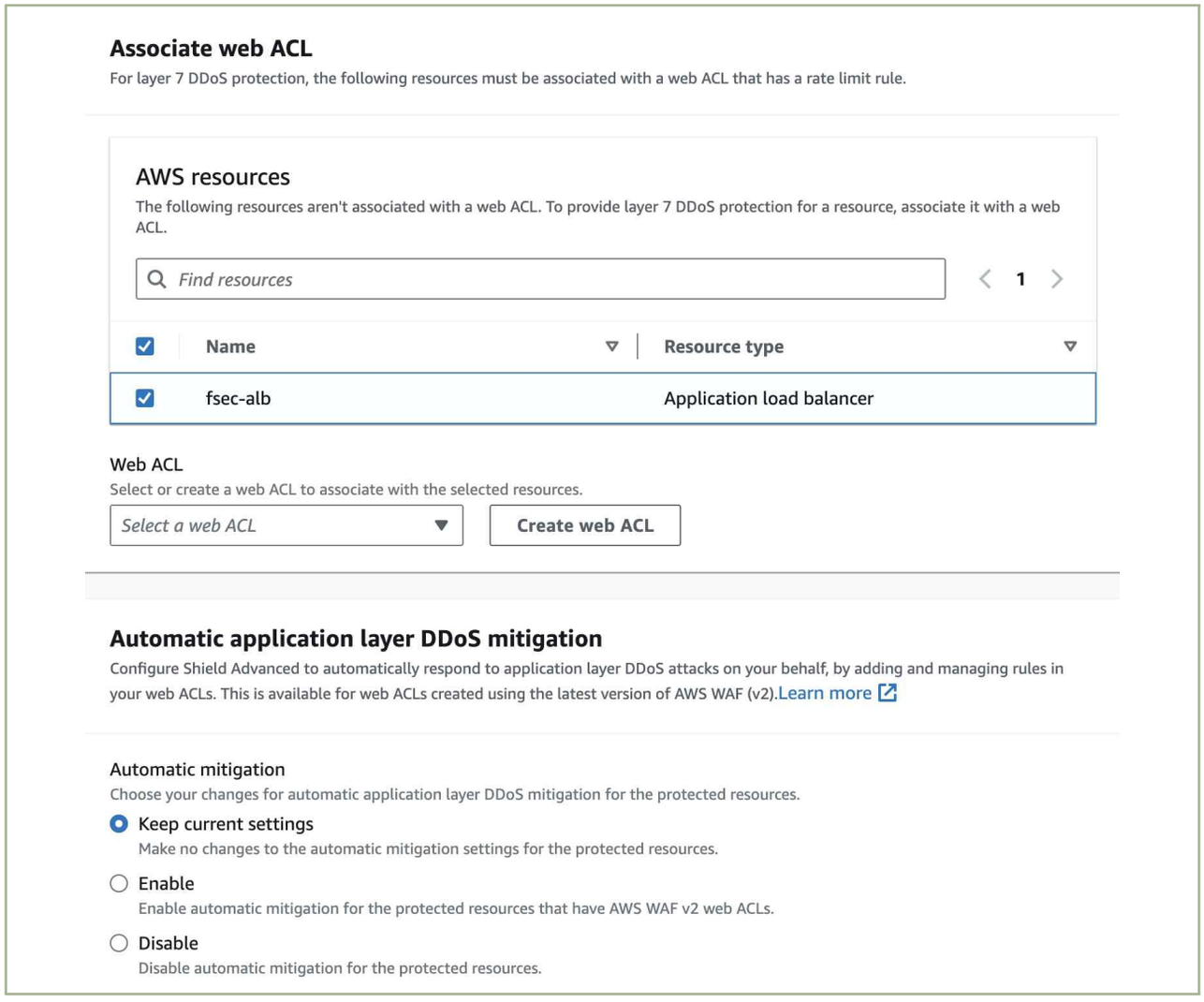
(1)-2. DDoS 공격으로부터 보호할 가상자원은 '콘솔 홈' → 'WAF & Shield' → 'AWS Shield' → 'Protected resources' 에서 등록할 수 있습니다.



| 그림 2-3-9 | Shield Advanced에서 보호할 자원 구성

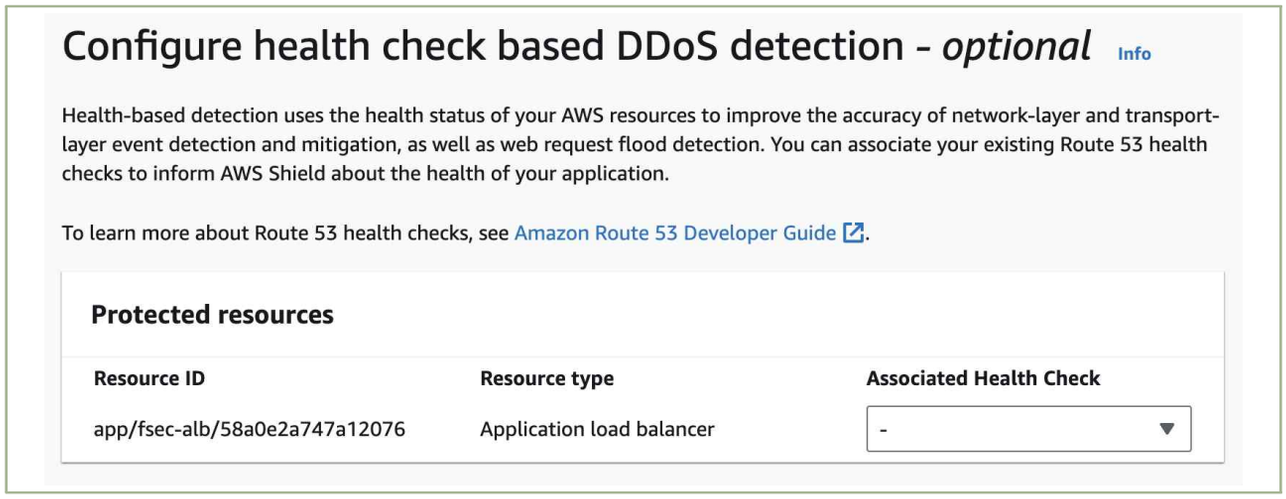
- 보호할 리소스가 구축된 리전(예. Seoul 리전)을 선택하여 리소스를 선택하여 등록할 수 있습니다. 보호 대상으로 선택할 수 있는 리소스 유형은 다음과 같습니다.
  - CloudFront
  - Route53
  - Global Accelerator
  - Application Load Balancer
  - Classic Load Balancer
  - Elastic IP Address

- 만약 보호 대상이 ALB(Application Load Balancer)와 같은 HTTP/HTTPS 서비스를 제공하는 유형이라면 L7 DDoS 완화(mitigation) 옵션을 추가 설정할 수 있으며 DDoS 공격이 발생할 경우 어떻게 대응할지 정할 수 있습니다.

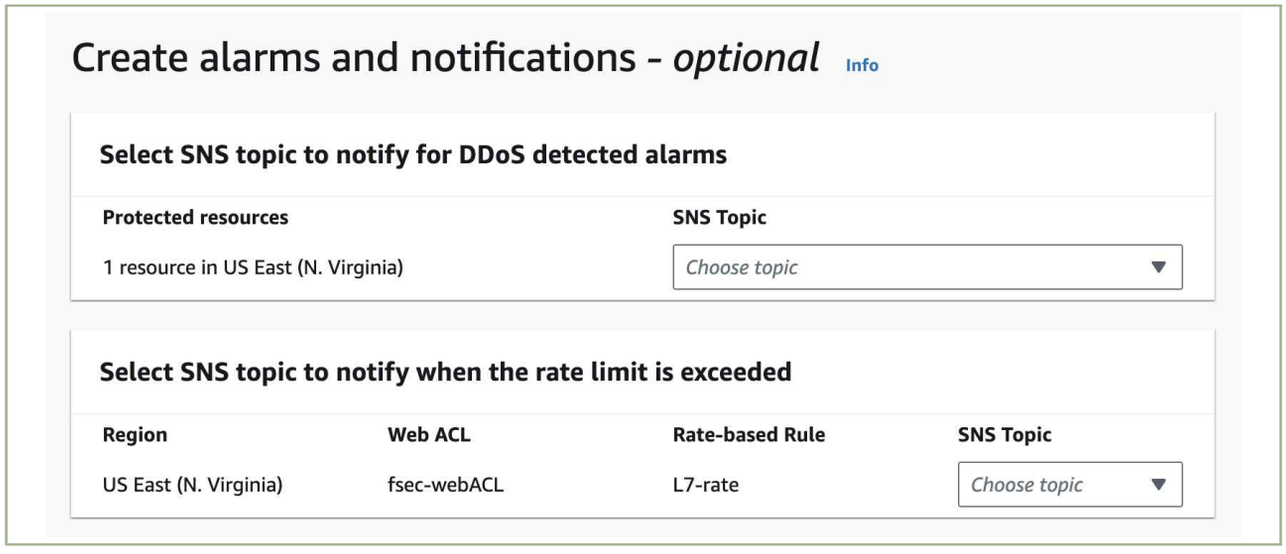


| 그림 2-3-10 | L7 DDoS Mitigation 선택 옵션

- Keep current settings는 DDoS 공격에 대한 자동 완화를 적용하지만 현재 설정을 유지하는 것이고, Enable은 DDoS 공격 발생 시 Count/Block 모드를 선택하여 자동 완화를 적용하는 것이고, Disable은 자동 완화를 적용하지 않는 옵션 입니다.
- Health Check 모니터링 옵션으로 DDoS 공격의 정확성을 향상시킬 수 있으며, AWS SNS Topic을 이용하여 공격 발생 시 Alram을 추가 설정할 수 있습니다.

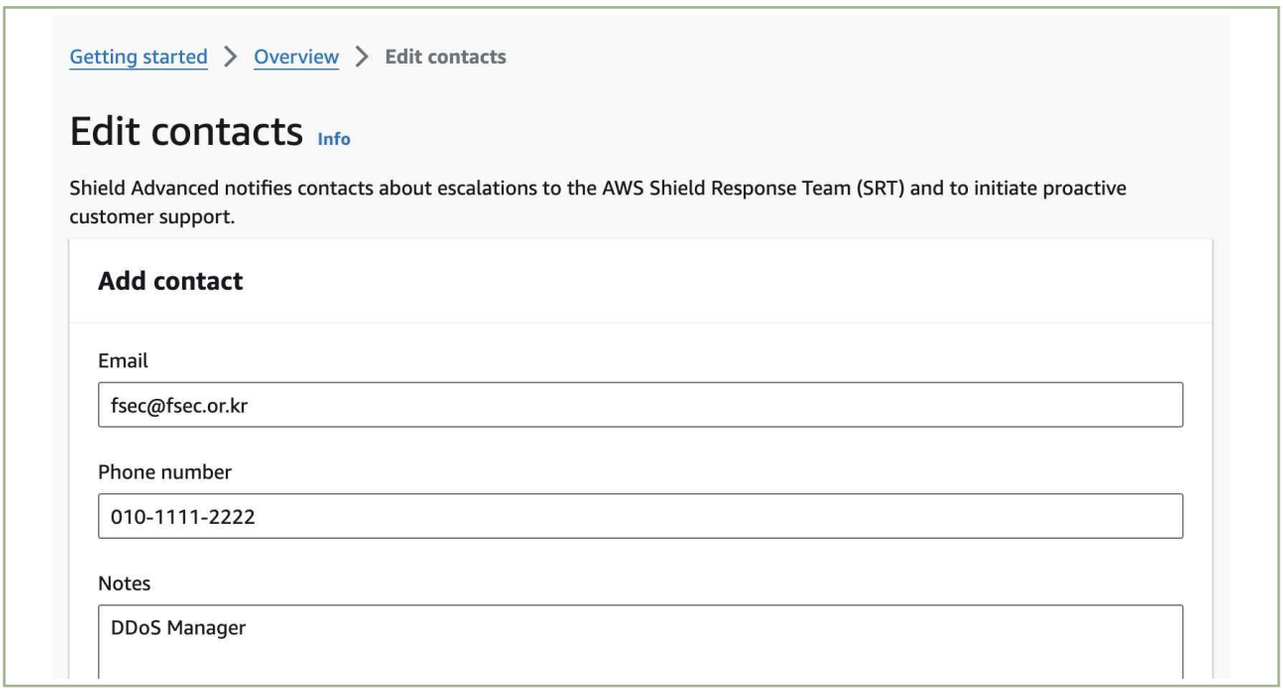


| 그림 2-3-11 | DDoS 공격 모니터링 옵션(Health Check)



| 그림 2-3-12 | DDoS 공격 Alram 옵션

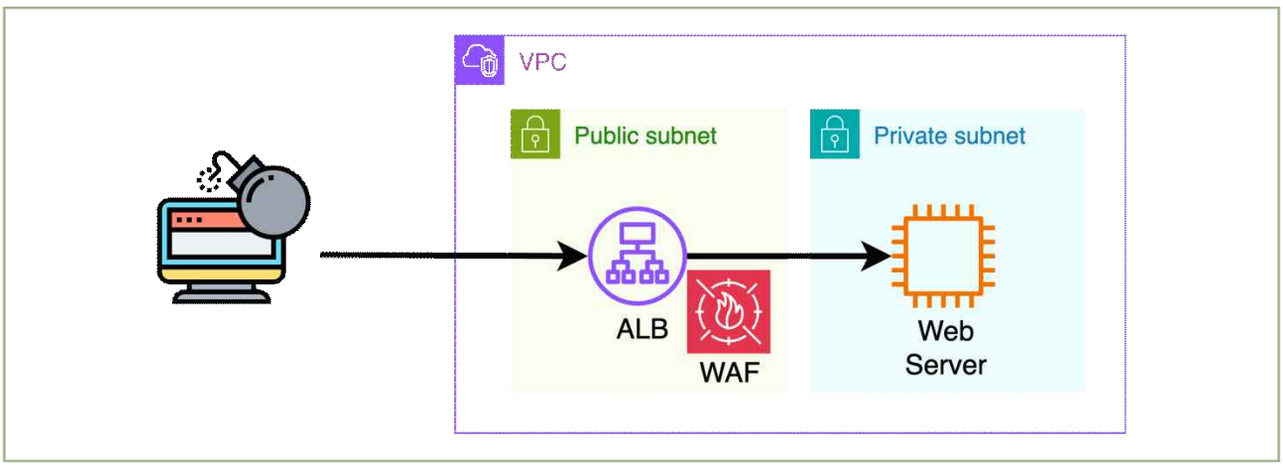
- DDoS 공격에 대한 선제적 대응(Proactive Engagement)를 위해 사용자에게 연락을 제공합니다. 그러기 위해서 최소 1명의 연락처를 등록하여야 하며, DDoS 공격 발생 시 SRT(Shield Response Team)에서 등록된 담당자 연락처로 연락을 하게 됩니다.



| 그림 2-3-13 | Proactive engagement and contacts 등록

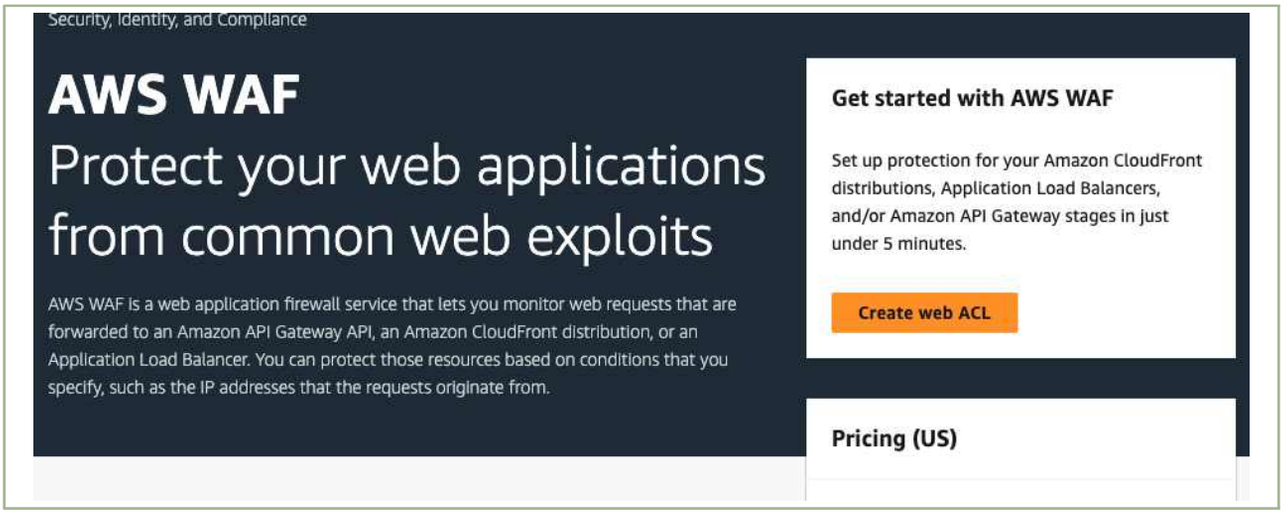
(2) AWS WAF (웹 애플리케이션 방화벽)

- 웹 애플리케이션 리소스로 전달되는 HTTP/HTTPS 요청을 모니터링하여 침해 위협을 대응하기 위해 AWS WAF를 사용할 수 있습니다.
- AWS WAF는 웹 취약점 공격 뿐만 아니라 Bot 공격 및 L7 계층의 DDoS 공격에 대한 대응도 가능하며, 사용자가 직접 하는 사용자 규칙 그룹과 AWS가 제공하는 규칙과 3rd Party 에서 제공하는 규칙을 사용할 수 있는 관리형 규칙 그룹을 제공합니다.
- 본 참고서에서는 AWS WAF 기능 중 Rate-based Rule을 이용하여 L7 DDoS 공격에 대응할 수 있는 구성을 예시로 설명합니다.



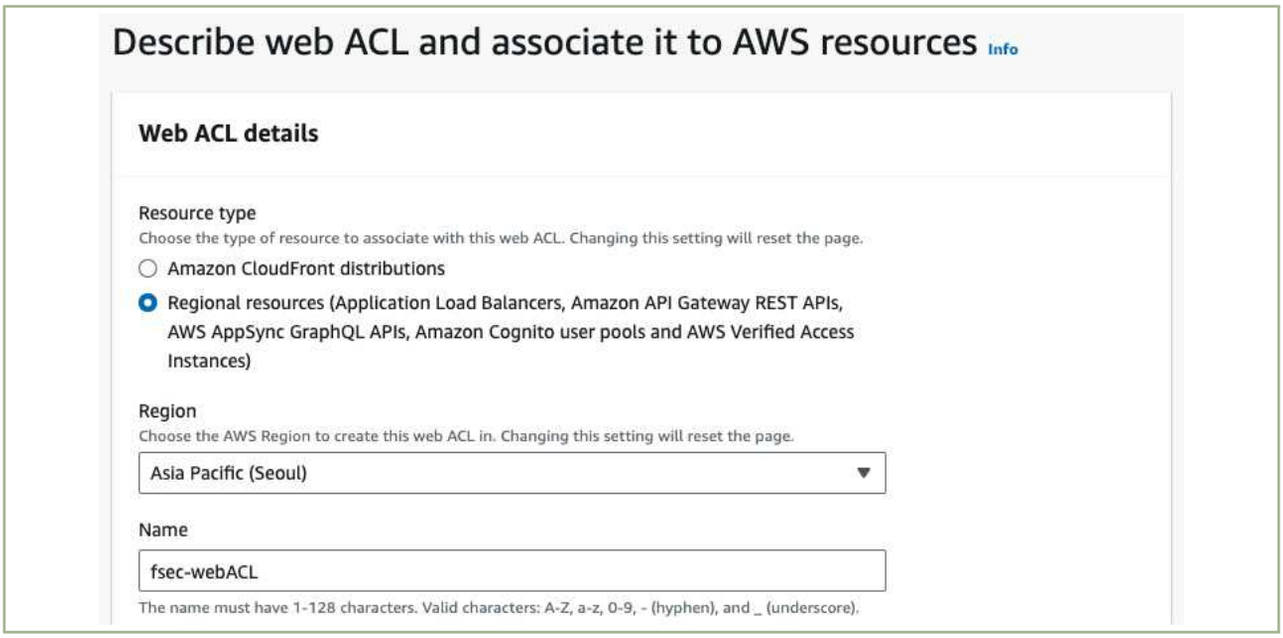
| 그림 2-3-14 | AWS WAF를 사용한 L7 DDoS 공격 대응 구성 예시

- **(AWS 관리 콘솔)** AWS WAF는 '콘솔 홈' → 'WAF & Shield' → 'AWS WAF' → 'Get started' → 'Create Web ACL' 을 선택하여 생성 및 구성할 수 있습니다.



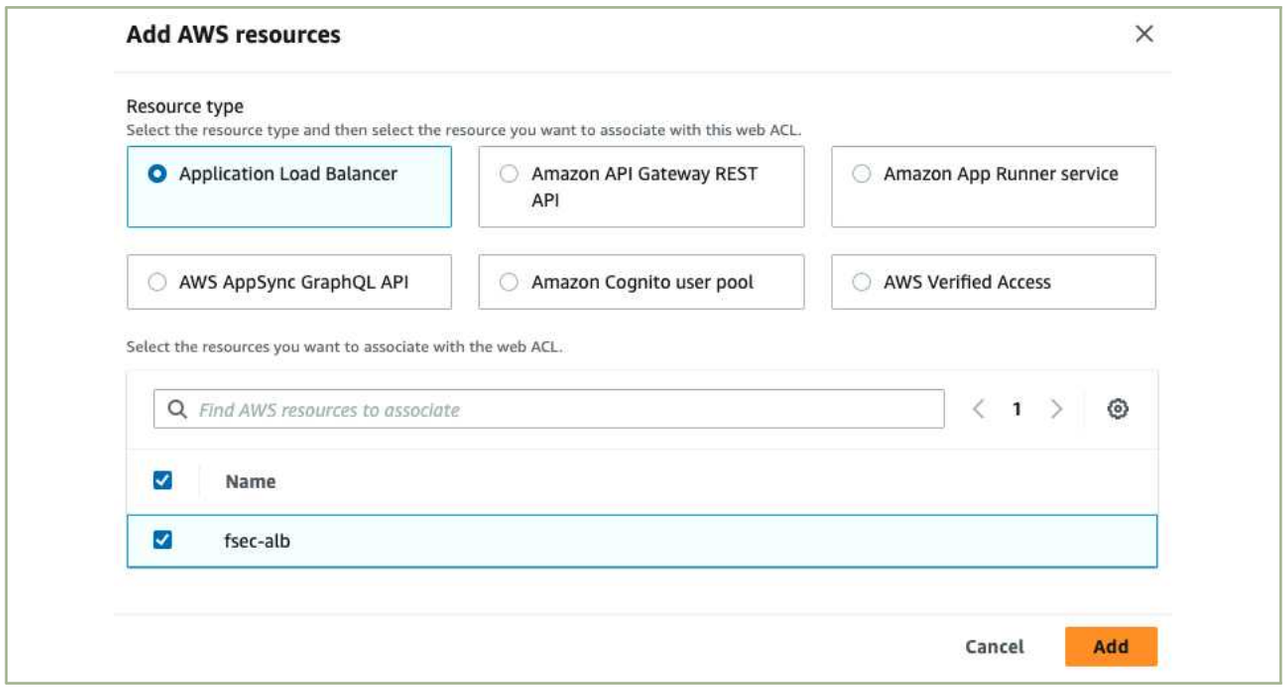
| 그림 2-3-15 | AWS Shield Advanced 구독 화면

- WAF web ACL 생성 시 모니터링하여 보호할 리소스 유형과 리전을 선택하고 사용할 이름을 정합니다.



| 그림 2-3-16 | web ACL 리소스 및 이름 설정

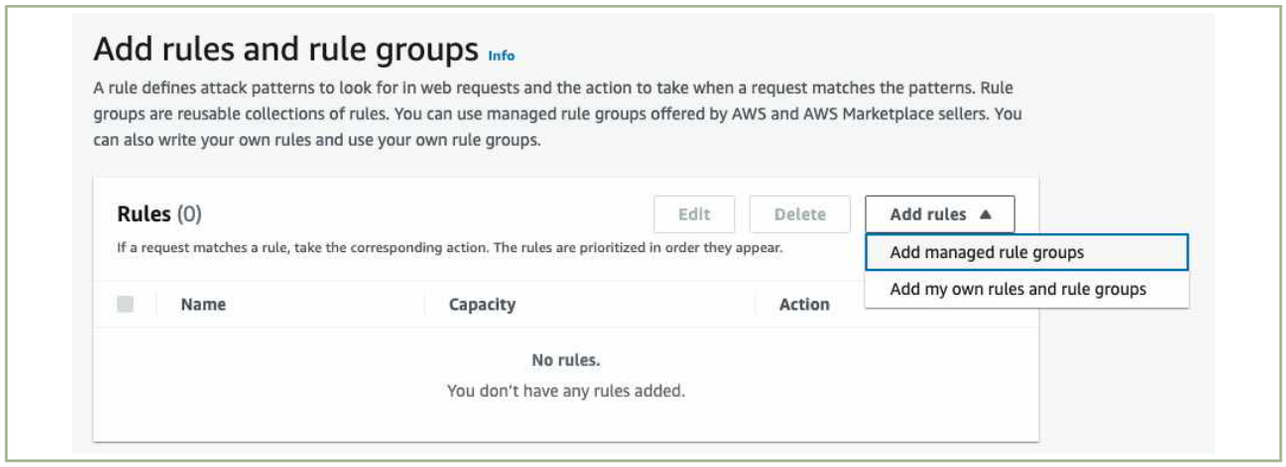
- 그리고 Add AWS resources를 클릭하여 web ACL에서 모니터링할 리소스를 선택합니다. (본 참고서에서는 위의 예시에 따라 ALB를 기준으로 함)



| 그림 2-3-17 | 모니터링 리소스 선택

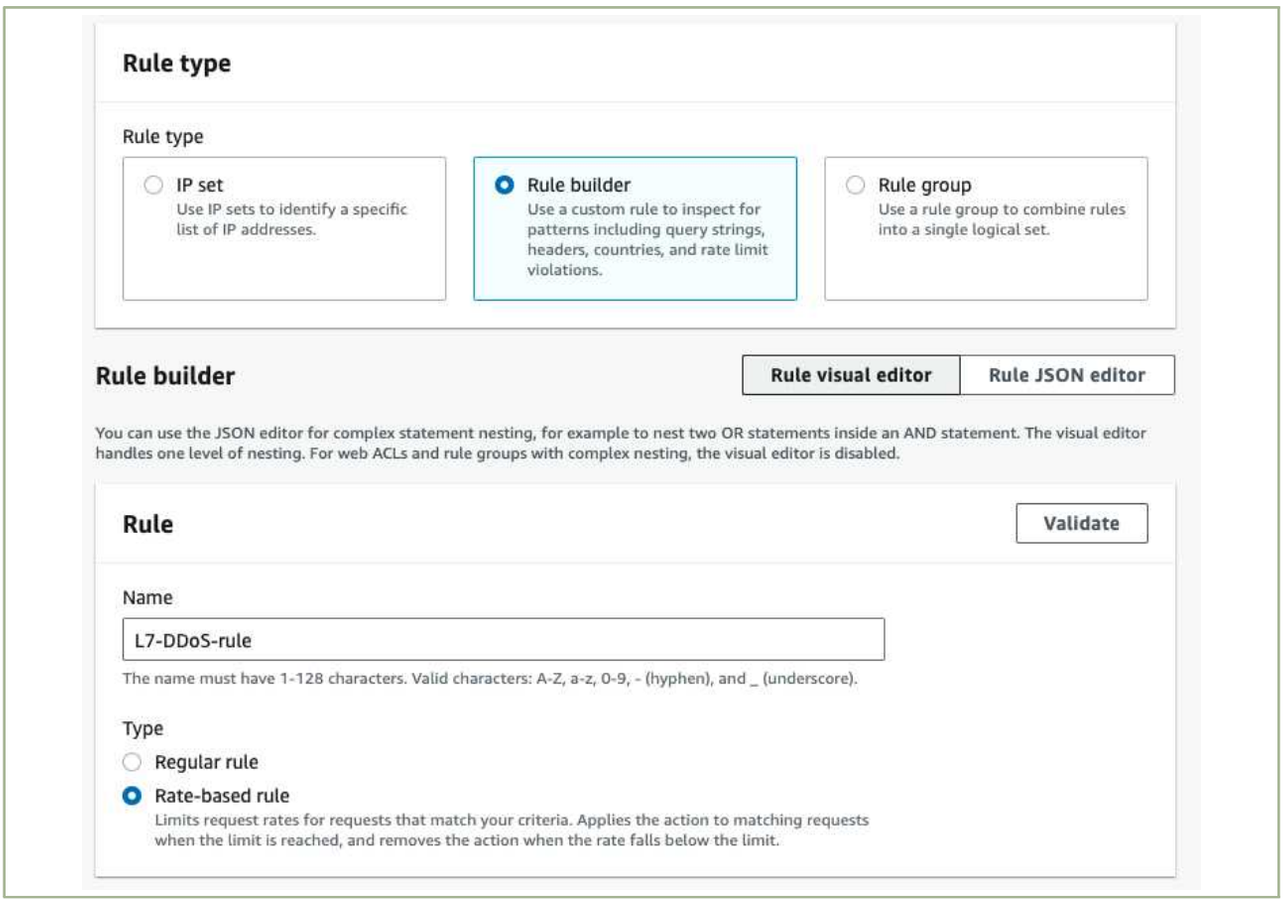
(1)-1. AWS web ACL에 규칙 등록

- 다음으로 생성할 web ACL에 적용할 규칙과 규칙 그룹을 선택합니다. 규칙 그룹은 앞서 소개한 사용자 규칙 그룹(사용자가 직접 생성 및 관리)과 관리형 규칙 그룹을 선택할 수 있습니다.



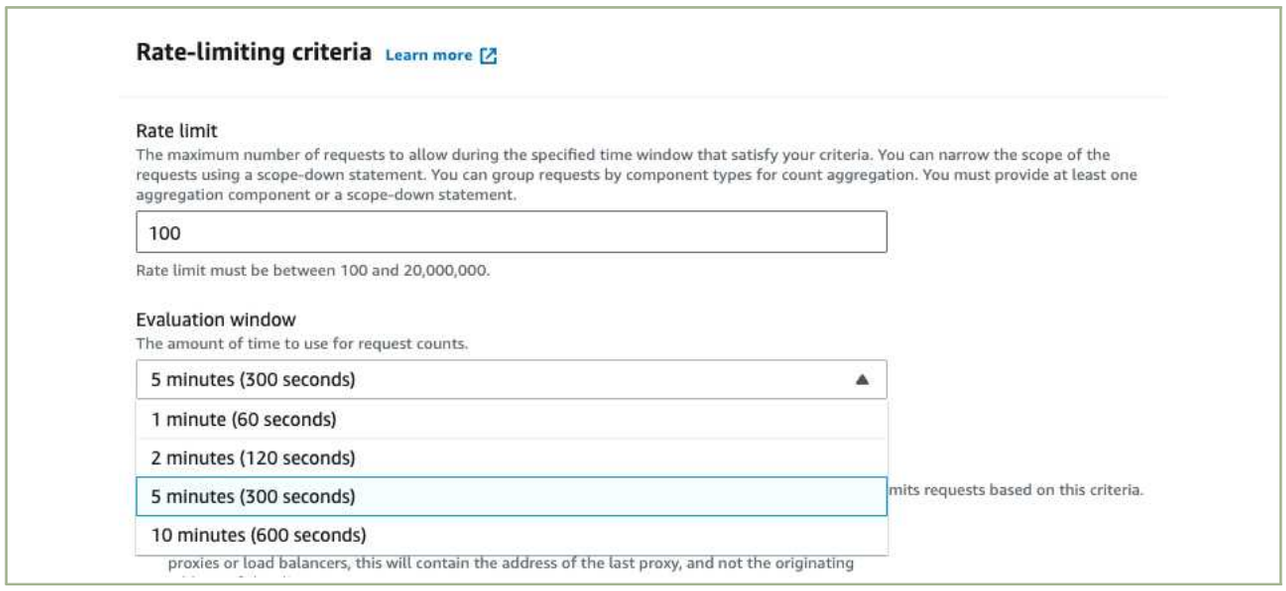
| 그림 2-3-18 | 규칙 그룹 선택 및 적용

- L7 DDoS 공격은 'Rate-based rule' 을 이용하여 대응할 수 있으며, Rule builder 기능을 이용하여 관련 규칙을 생성합니다.
- 규칙은 여러개 를 등록할 수 있고 Priority 순서에 따라서 동작 순위가 결정이 됩니다.



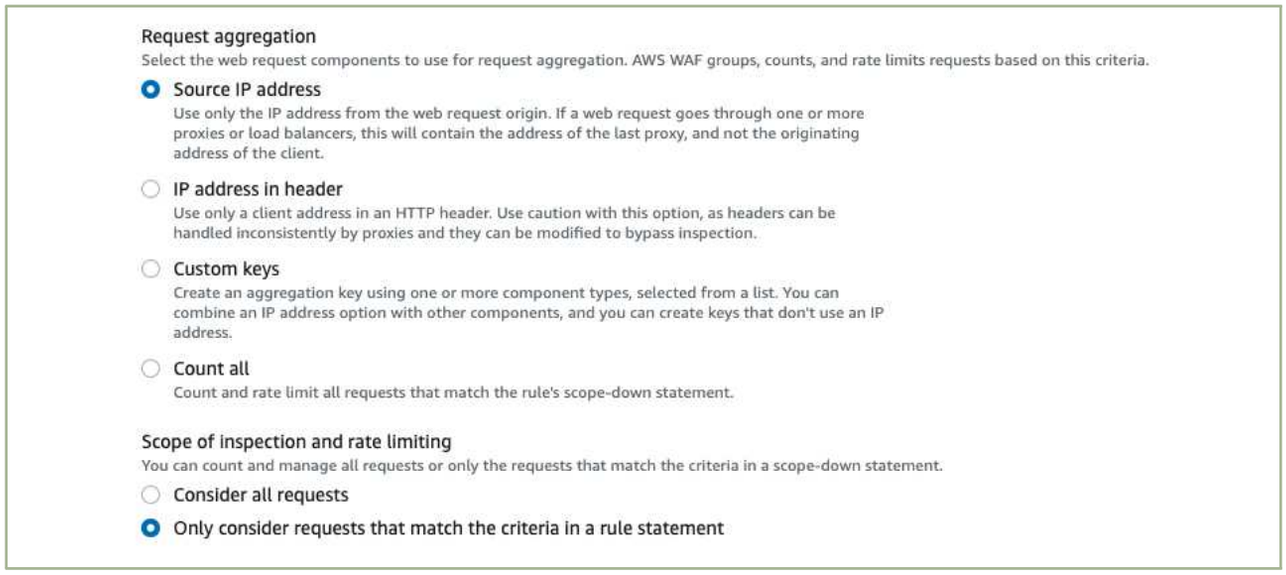
| 그림 2-3-19 | Rule builder를 이용한 규칙 생성

- Rate-based Rule은 정한 시간(Evaluation Window) 내에 Rate Limit으로 지정한 HTTP/HTTPS 요청 수 이상으로 요청이 발생하였을 때 대응을 하는 규칙 입니다.
- 예를 들어 1분에 100건을 설정하였다면 1분 동안 최대99건의 요청은 허용하지만 1분 동안 100건이 넘을 경우 정한 Action(Block/Count/CAPCHA/Challenge)를 수행하게 됩니다.
- 기준은 1,2,5,10분으로 정할 수 있으며, 기본은 5분입니다.



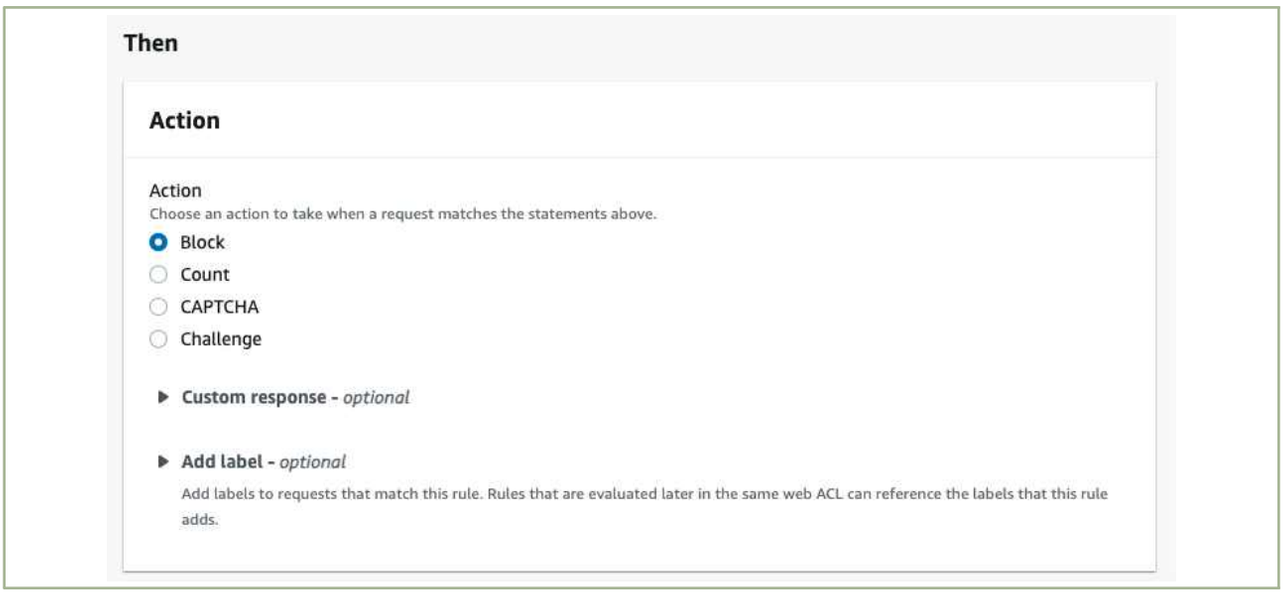
| 그림 2-3-20 | Rate-based Rule 설정

- 다음으로 Rate Limit에 정한 요청 수를 어떤 기준으로 카운트 할 것인지를 설정합니다.
  - Source IP Address : 웹 요청을 한 Source만 사용함
  - IP Address in Header : HTTP/S 헤더의 Client 주소만 사용함
  - Custom Keys : Header, Query string/argument, Cookie, Label, HTTP method, URI path와 같은 컴포넌트 유형을 사용함
  - Count All : Statement의 조건을 사용함(컴포넌트, 국가IP 등 사용)



| 그림 2-3-21 | Request Aggregation 설정

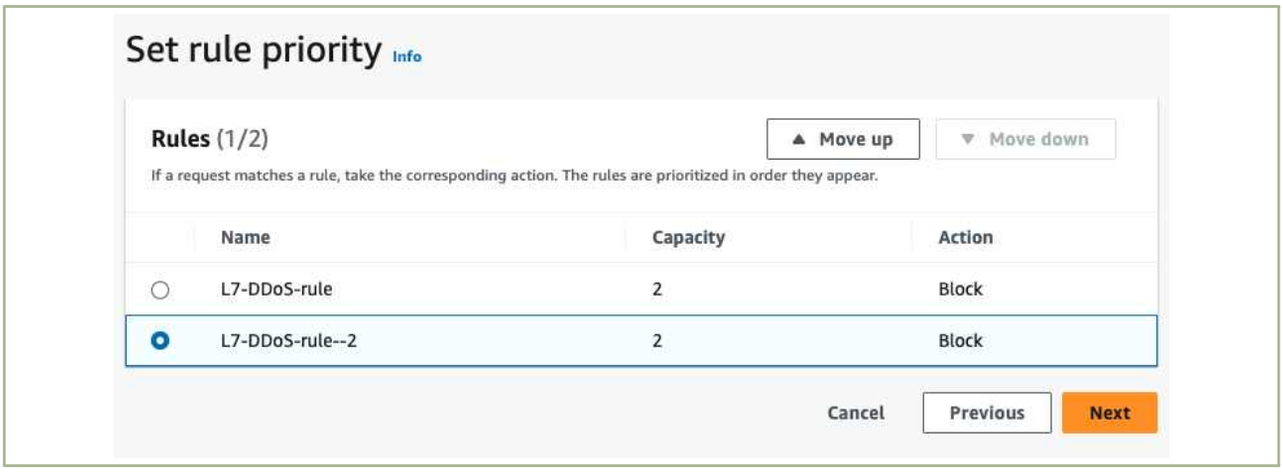
- 마지막으로 Rate-limiting criteria에서 정한 규칙에 매칭될 경우 어떤 대응을 할지 정합니다.



- 미리 생성한 룰 그룹이 없을 경우 Shield Advanced는 사용자가 지정한 리소스에 대해서만 보호하게 됩니다. 따라서 보호할 가상자원 또는 가상자원과 연결된 리소스를 선택하여 보호 대상으로 등록합니다.

(1)-2. 규칙의 Priority 설정

- 다음으로 web ACL에 등록한 규칙에 대한 우선 순위를 설정 합니다.
- 우선 순위가 높은 규칙이 Action이 Count일 경우 다음 우선 순위 규칙을 검사하긴 하지만, Action이 Block 또는 Pass일 경우 다음 우선 순위 규칙을 평가하지 않으므로 우선 순위가 web ACL 동작에 영향을 미치게 됨에 따라 우선 순위 설정이 중요합니다.

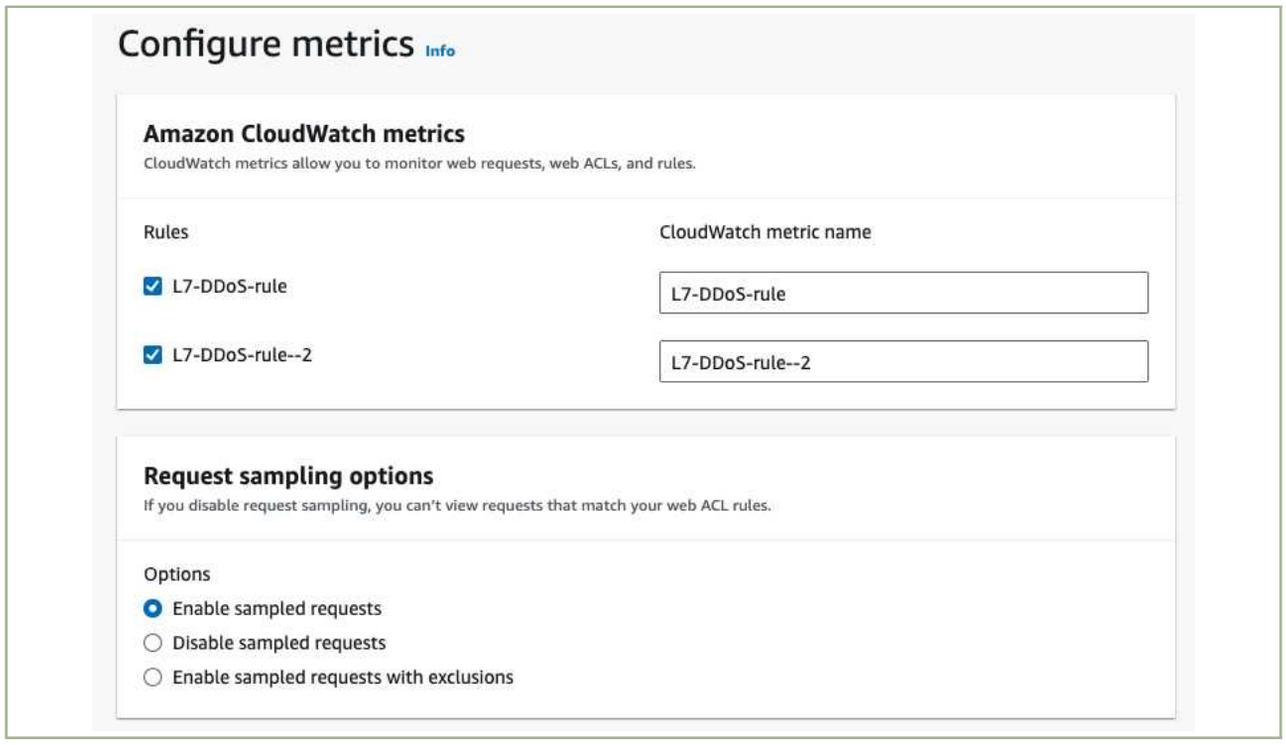


| 그림 2-3-22 | 규칙 Priority 설정

(1)-3. Configure metrics 설정

- CloudWatch Metric을 이용하여 모니터링 할 규칙을 선택할 수 있고, 검사된 웹 요청과

일치하는 규칙의 그래프(Source IP, URI, 일치한 규칙, Action, 시간 등) 정보를 활성화 할 것인지 선택할 수 있습니다.



| 그림 2-3-23 | 규칙에 대한 모니터링 설정

- 생성할 web ACL에 대한 검토 후 web ACL 생성이 완료되면, ALB로 유입되는 웹 요청을 AWS WAF가 검사하여 L7 DDoS 공격에 대한 대응을 할 수 있습니다.

4 참고 사항

- 본 가이드에서 다루지 않는 서비스에 대한 다양한 기능들은 아래의 서비스 설명가이드를 참고 바랍니다.
  - Traffic Mirroring
    https://docs.aws.amazon.com/vpc/latest/mirroring/what-is-traffic-mirroring.html
  - VPC Peering
    https://docs.aws.amazon.com/ko_kr/vpc/latest/peering/what-is-vpc-peering.html
  - AWS WAF & Shield
    https://docs.aws.amazon.com/waf/latest/developerguide/waf-chapter.html

[문의처에 관한 정보]

1 기준

식별번호	기준	내용
2.4.	공개용 웹 서버 네트워크 분리	클라우드 환경을 통한 공개용 웹 서버 구현 시 내부통신망과 분리하여 내부통신망과 외부통신망 사이 별도의 독립된 통신망("이하 DMZ")을 구현하고 안전하게 보호하여야 한다.

2 설명

- 클라우드 환경을 통한 공개용 웹 서버의 경우 내부통신망과 분리하여 내부통신망과 외부통신망 사이 별도의 독립된 통신망에 구현하고 접근통제를 수행하여야 한다.

  - 예시

    1) VPC 등 네트워크 분리 기능을 통한 DMZ 망 구축 후 공개용 웹 서버 구현

    2) 공개용 웹 서버 직접 접근 시 통제(ACL 등)에 의한 중요단말기 등에서 접근하도록 관리

3 우수 사례

1) VPC 등 네트워크 분리 기능을 통한 DMZ 망 구축 후 공개용 웹 서버 구현

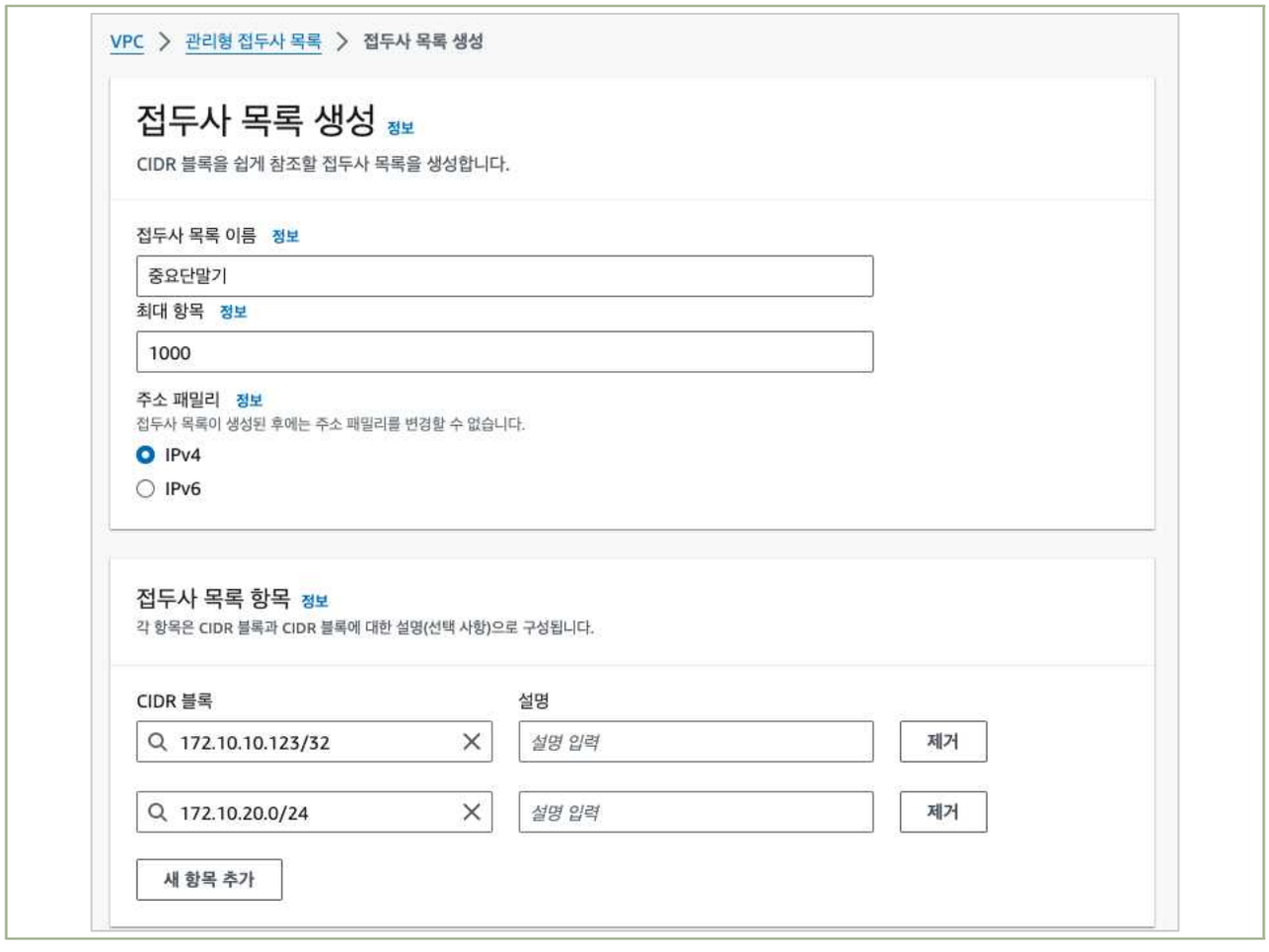
- 공개용 웹 서버는 DMZ망 VPC에 구성하고 내부망 VPC와 연결은 Transit 게이트웨이 또는 VPC Peering을 이용하여 구성할 수 있으며, 이 때 DMZ망 VPC와 내부망 VPC 사이에는 방화벽 등 네트워크 접근 통제 방안을 적용할 수 있습니다.
- 멀티 VPC 를 이용하여 DMZ망과 내부망을 분리하는 구성은 '3.2-1) VPC 등 네트워크 관련 기능을 통한 네트워크 접근 통제(인터넷망 등)' 내용을 참고하여 할 수 있습니다.

2) 공개용 웹 서버 직접 접근 시 통제(ACL 등)에 의한 중요단말기 등에서 접근하도록 관리

- EC2 인스턴스를 이용하여 공개용 웹 서버를 구축할 경우 EC2 인스턴스에 설정된 보안 그룹과 VPC 및 서브넷의 NACL에서 접근이 필요한 대상에 대해서만 허용 규칙을 설정하여 적용할 수 있습니다.
- 접근이 필요한 중요단말기 등에 대한 리스트를 Prefix Lists로 관리할 수 있으며, 생성한 Prefix Lists를 공개용 웹 서버 인스턴스의 보안 그룹에 허용으로 적용하면, 리스트 내의 IP에 대해서 공개용 웹 서버에 접근이 가능합니다.

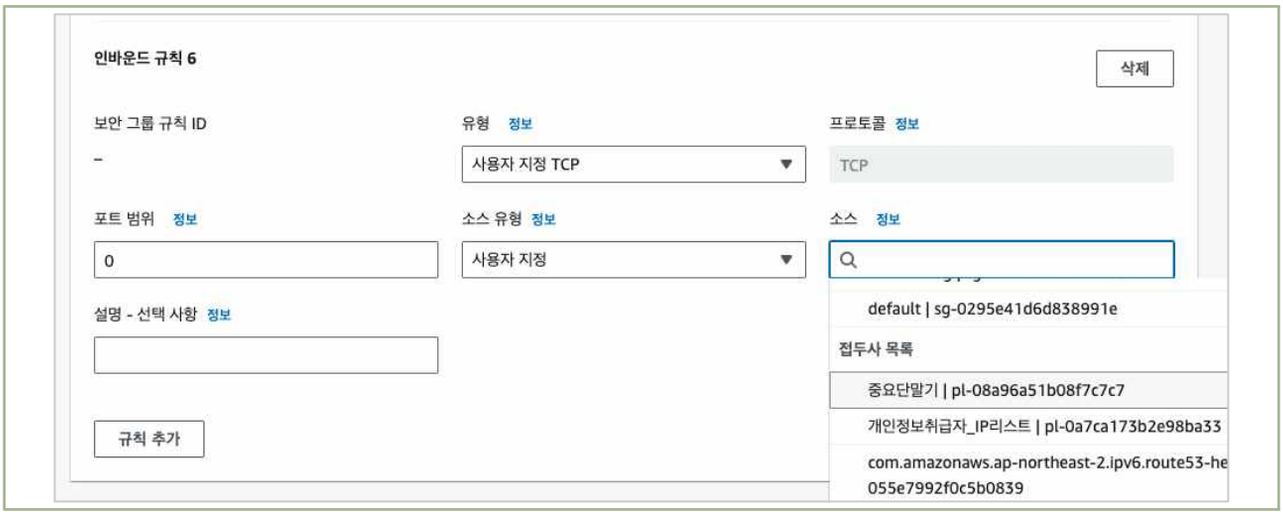
- 보안 그룹 및 NACL을 이용한 네트워크 구성 및 통제는 '3.1-2) 보안 그룹(Security group) 또는 NACL(Network ACL) 등의 기능을 통한 네트워크 구성 및 통제(인/아웃바운드 통제 등)'을 참고하여 할 수 있습니다.

- **(AWS 관리 콘솔)** Prefix Lists는 '콘솔 홈' → 'VPC' → '관리형 접두사 목록' 에서 생성 할 수 있습니다.



| 그림 2-4-1 | Prefix Lists 생성

- 등록된 Prefix List를 보안 그룹에 등록하고 접근통제 대상이 변경이 필요할 경우 Prefix List의 항목을 변경하면 해당 Prefix List를 소스(Source)로 등록한 모든 보안 그룹은 자동으로 변경된 항목이 적용됩니다.



| 그림 2-4-2 | 보안 그룹에 Prefix Lists 적용

4 참고 사항

- 본 가이드에서 다루지 않는 서비스에 대한 다양한 기능들은 아래의 서비스 설명가이드를 참고
  - Prefix Lists
    https://docs.aws.amazon.com/vpc/latest/userguide/managed-prefix-lists.html

[문의처에 관한 정보]

1 기준

식별번호	기준	내용
2.5.	**네트워크 사설 IP주소 할당 및 관리**	**클라우드 환경을 통한 내부망 네트워크 구현 시 사설 IP부여 등으로 보안을 강화하고, 내부IP 유출을 금지하여야 한다.**

2 설명

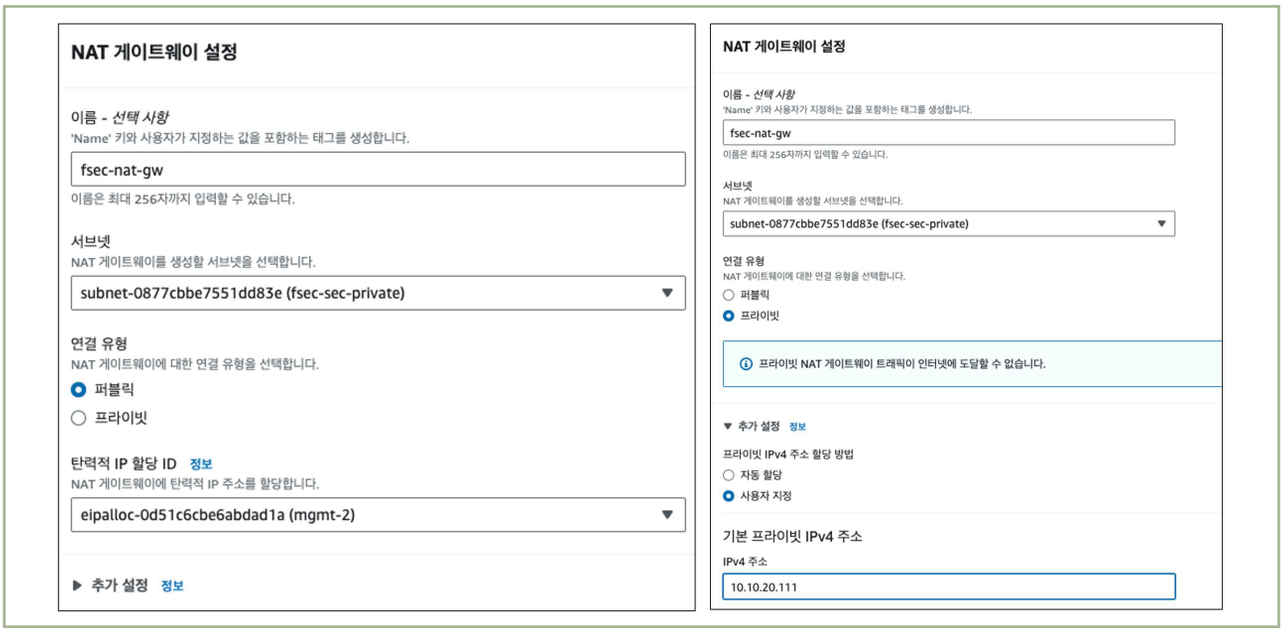
- 클라우드 환경 내 내부망 네트워크 구현 시 사설IP를 부여하고 주기적으로 현황을 검토하여야 한다.
  - 예시
    1) 인터넷 게이트웨이, NAT 게이트웨이 등 관련 기능을 통해 사설IP부여 및 IP 관리 수행
    2) 사설 IP 할당 현황에 대한 주기적 검토 수행

3 우수 사례

1) 인터넷 게이트웨이, NAT 게이트웨이 등 관련 기능을 통해 사설IP부여 및 IP 관리 수행

- AWS VPC에 구성되는 리소스는 기본적으로 VPC에 설정된 CIDR 범위 내의 프라이빗 IP(사설 IP)가 부여되며, EIP(Elastic IP)와 같은 퍼블릭IP(공인IP)가 필요한 경우는 리소스 생성 시 또는 생성 후 별도 구성으로 부여할 수 있습니다.
- 내외부 통신 시 프라이빗 IP로 변환(NAT)은 NAT 게이트웨이를 이용하여 가능합니다.

- **(AWS 관리 콘솔)** NAT 게이트웨이는 '콘솔 홈' → 'VPC' → 'NAT 게이트웨이' 에서 생성 할 수 있습니다.



| 그림 2-5-1 | NAT 게이트웨이 생성 (퍼블릭/프라이빗)

- NAT 게이트웨이는 퍼블릭/프라이빗 두가지 연결 유형이 있습니다.
  - 퍼블릭 : 할당한 EIP(퍼블릭IP)로 변환하여 통신
  - 프라이빗 : 선택한 서브넷 CIDR 범위 내 지정한 프라이빗 IP로 변환하여 통신
- 퍼블릭 NAT게이트웨이는 인터넷 통신이 필요한 리소스가 있을 경우 주로 사용할 수 있으며, 프라이빗 NAT게이트웨이는 On-Premise와 통신시 또는 타 VPC와 통신을 할 경우 IP관리를 위해 사용할 수 있습니다.

2) 사설 IP 할당 현황에 대한 주기적 검토 수행

- 사용하는 리소스에 IP 현황 확인은 '네트워크 인터페이스' 기능에서 확인할 수 있으며, 퍼블릭IP 및 프라이빗 IP 모두 확인이 가능합니다.
- 확인된 프라이빗 IP 할당 현황에 대한 주기적 검토는 고객의 내부 정책에 따라 수행할 수 있습니다.

4 참고 사항

- 본 가이드에서 다루지 않는 서비스에 대한 다양한 기능들은 아래의 서비스 설명가이드를 참고 바랍니다.
  - NAT 게이트웨이
    https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html

[문의처에 관한 정보]

3. 계정 및 권한 관리

3.1. 클라우드 계정 권한 관리
3.2. 이용자별 인증 수단 부여
3.3. 인사 변경 사항 발생 시 계정 관리
3.4. 클라우드 가상자원 관리 시스템 관리자 권한 추가인증 적용
3.5. 클라우드 가상자원 관리 시스템 로그인 규칙 수립
3.6. 계정 비밀번호 규칙 수립
3.7. 공개용 웹 서버 접근 계정 제한

3 계정 및 권한 관리

1 기준

식별번호	기준	내용
3.1.	**클라우드 계정 권한 관리**	**클라우드 서비스 이용 시 업무 및 권한에 따라 계정을 관리하여야 한다.**

2 설명

- 클라우드를 이용하는 임직원의 업무 및 권한에 따라 계정을 관리하여야 한다.

  - 예시

    1) 자격 증명 등의 기능을 이용하여 계정 권한 관리

    2) 사전에 정의된 행위만이 가능하도록 역할을 생성

- 콘솔 최상위 관리자(ex. 최초 가입계정 등)은 서비스 운영에 활용하지 않아야 한다.

  - 예시

    1) 부득이 일부 서비스에 대해 관리자 권한이 필요한 경우, 신규로 계정을 생성하여 필요한 권한을 부여한 후 활용

    2) 예외적으로 반드시 최초 콘솔 가입계정을 이용하여야 하는 특정 서비스의 경우에는, MFA등 추가 인증 방식을 구현하고 접속 IP를 제한 하는 등 강화된 보안환경 구성

3 우수 사례

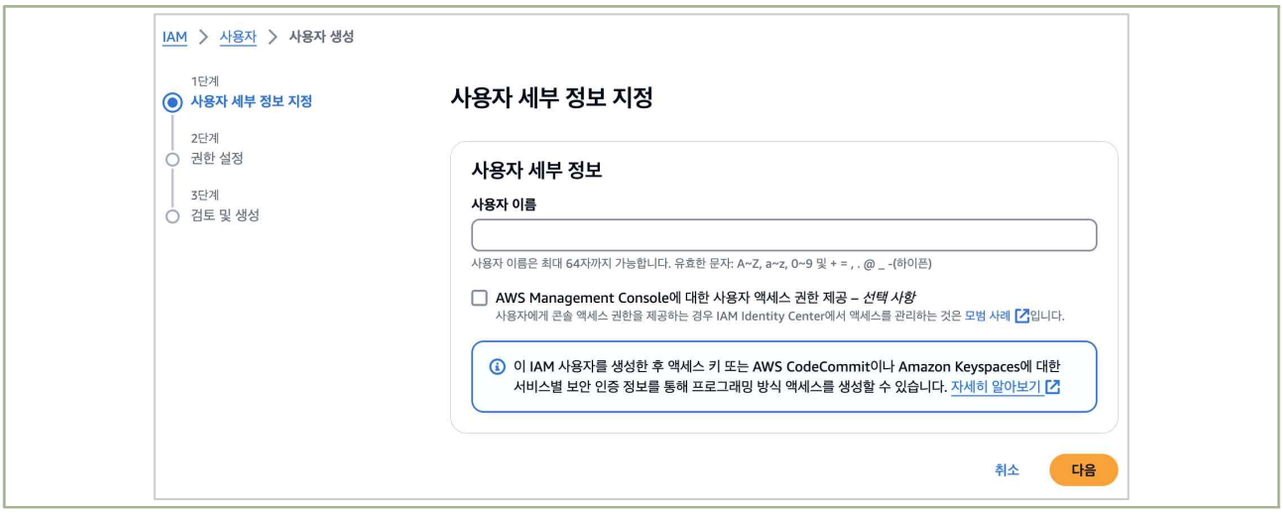
1) 자격 증명 등의 기능을 이용하여 계정 권한 관리

(1) 사용자 생성

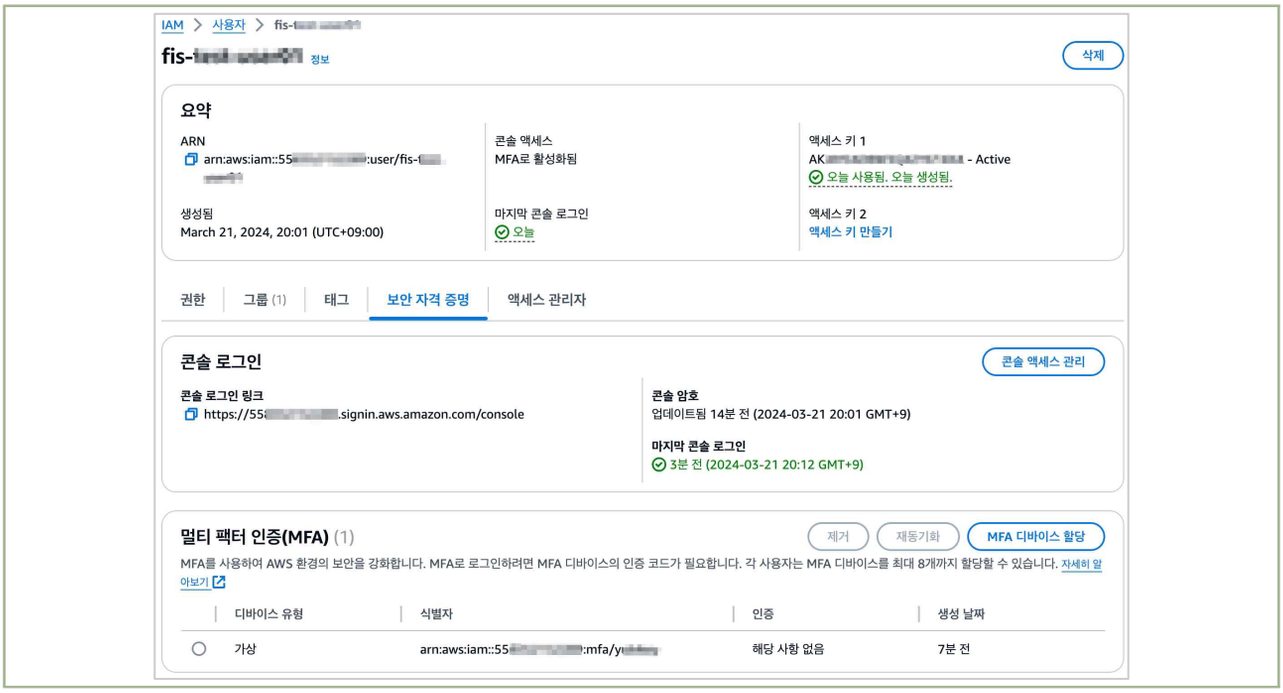
- IAM 사용자는 개별 계정이 아니라 해당 계정 내의 사용자를 말하며, 각 사용자는 고유의 AWS Management Console 액세스 암호 또는 프로그래밍 방식의 요청을 할 수 있도록 하는 개별

액세스 키를 생성하고 독립적으로 사용할 수 있습니다.

- **(AWS 관리 콘솔)** 사용자 생성은 '콘솔 홈'→'서비스 명 IAM 입력'→ '사용자'→ '사용자 생성' 또는 CLI 명령으로 생성할 수 있습니다.



| 그림 3-1-1 | 사용자 생성 화면



| 그림 3-1-2 | 사용자 생성 후 세부 정보 화면

- **(AWS CLI)** CLI를 통해서 사용자를 생성하기 위해서는 'iam create-user' 명령어를 사용하며, 아래와 같이 사용자명을 그 인자로 입력하여 생성할 수 있습니다.

```
aws iam create-user \
    --user-name Bob
```

Output:

```
{
    "User": {
        "UserName": "Bob",
        "Path": "/",
        "CreateDate": "2023-06-08T03:20:41.270Z",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:user/Bob"
    }
}
```

| 그림 3-1-3 | CLI 명령 - 사용자 생성

- 생성된 사용자가 필요한 사용자 그룹에 속하도록 'iam add-user-to-group'을 명령을 사용하여 지정할 수 있습니다.

```
aws iam add-user-to-group \
    --user-name Bob \
    --group-name Admins
```

| 그림 3-1-4 | CLI 명령 - 사용자 그룹 지정

- 사용자 그룹 지정 CLI 명령어는 별도의 결과 출력이 없으므로 결과를 확인하기 위해서는 'iam get-group' 명령어로 확인할 수 있습니다.

```
aws iam get-group \
    --group-name Admins
```

Output:

```
{
    "Group": {
        "Path": "/",
        "CreateDate": "2015-06-16T19:41:48Z",
        "GroupId": "AIDGPMS9RO4H3FEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:group/Admins",
        "GroupName": "Admins"
    },
    "Users": []
}
```

| 그림 3-1-5 | CLI 명령 - 사용자 그룹 지정 결과 확인

- 사용자에게 콘솔 로그인 권한이 필요할 경우 'iam create-login-profile' 명령어를 사용하여 콘솔 로그인 초기 비밀번호를 할당할 수 있으며, 최초 로그인 시 패스워드가 변경되도록 '--password-reset-required' 옵션을 사용할 수 있습니다.

```
aws iam create-login-profile \
    --user-name Bob --password BobLogin8P@ssword --password-reset-required

Output:

{
    "LoginProfile" : {
        "UserName": "Bob",
        "CreateDate": "2024-03-21T11:24:36+00:00",
        "PasswordResetRequired": true
    }
}
```

| 그림 3-1-6 | CLI 명령 - 콘솔 비밀번호 설정 및 최초 로그인 시 변경 설정

- 그리고 'iam create-access-key' 명령어를 사용하여 웹 콘솔이 아니라 AWS CLI 나 다른 프로그램에서 사용할 수 있는 Access key/Secret Access key를 생성할 수 있습니다.

```
aws iam create-access-key \
    --user-name Bob

Output:

{
    "AccessKey": {
        "UserName": "Bob",
        "Status": "Active",
        "CreateDate": "2015-03-09T18:39:23.411Z",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"
    }
}
```

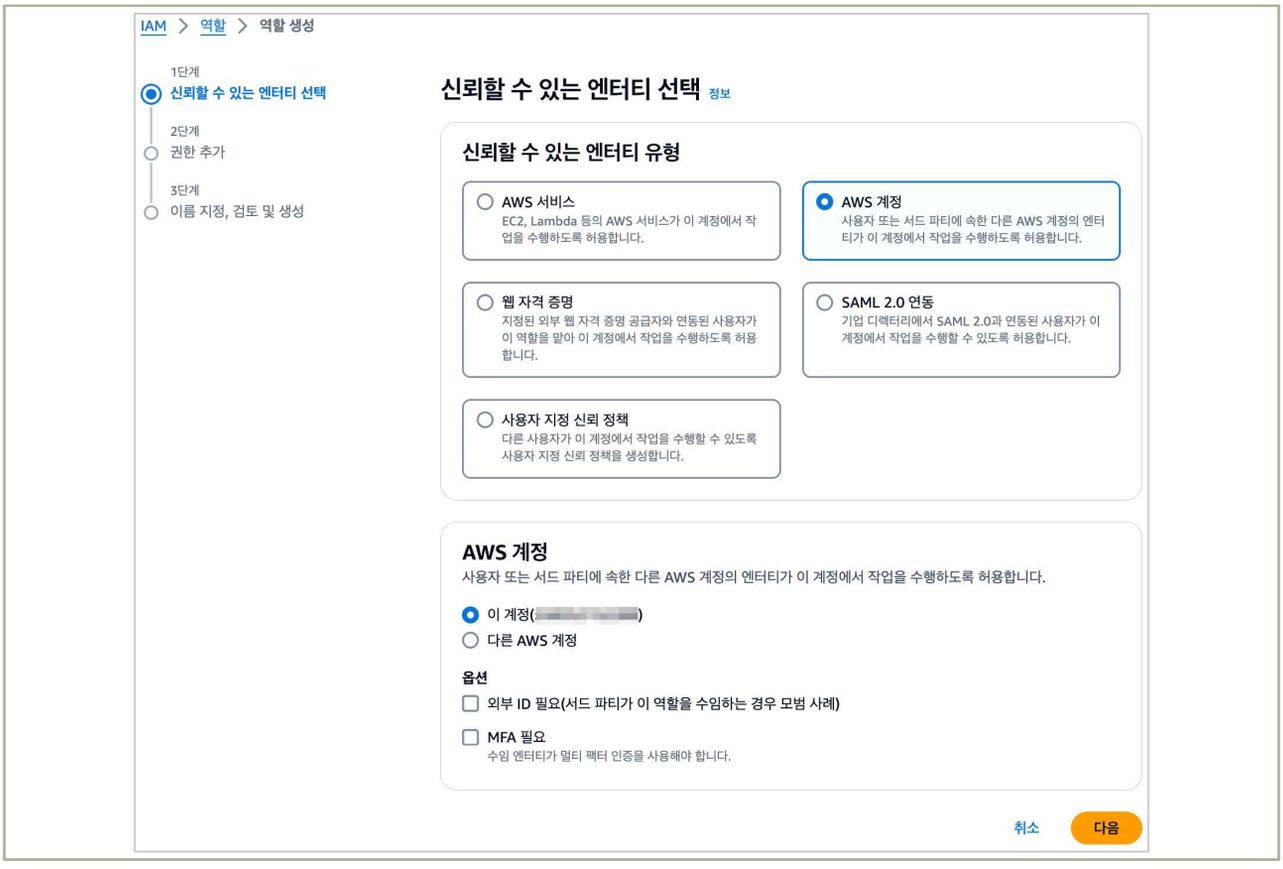
| 그림 3-1-7 | CLI 명령 - Access Key, Secret Key 생성

2) 사전에 정의된 행위만 가능하도록 역할 생성

(1) 역할 생성

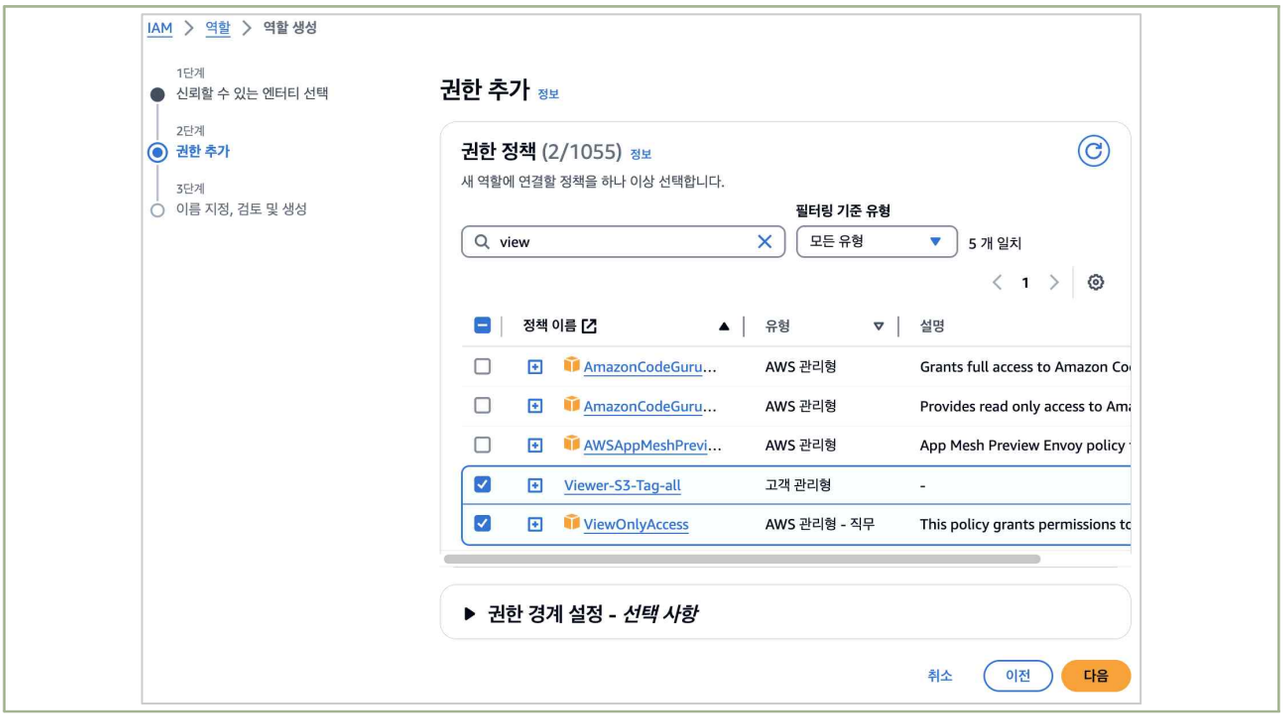
- IAM 사용자의 자격 증명은 별도로 파기하지 않는 이상 계속 유지가 되는 장기 자격 증명 입니다.
- 따라서 실제 업무를 수행할 때는 사전에 정의된 행위만을 수행하는 별도의 역할을 생성하여 권한 통제를 하여야 하고, 해당 역할에서 사용하는 임시 자격 증명을 사용하도록 하여 보다 보안 통제를 높일 수 있습니다.

- **(AWS 관리 콘솔)** 역할 생성은 '콘솔 홈'→'서비스 명 IAM 입력'→ '역할'→ '역할 생성' 또는 CLI 명령으로 할 수 있습니다.



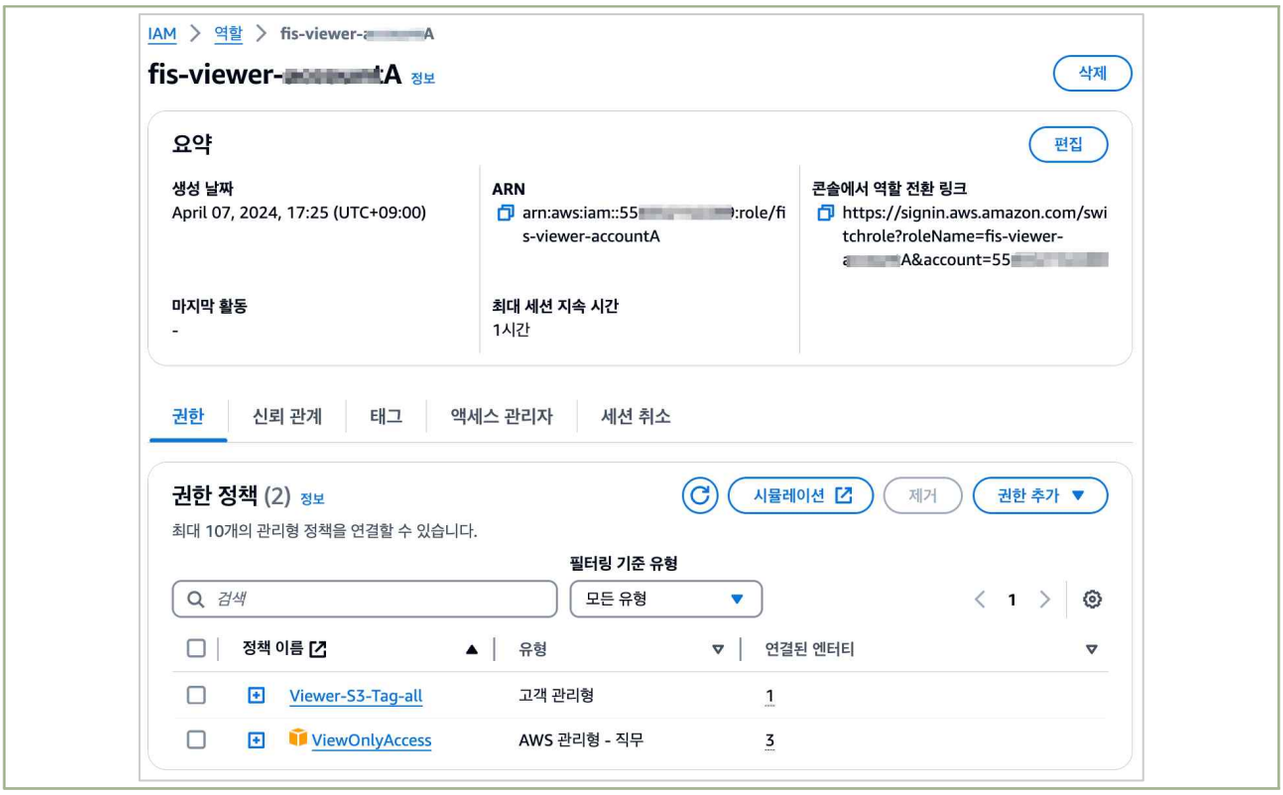
| 그림 3-1-8 | 역할 생성 화면

- 역할은 해당 역할을 사용하도록 신뢰할 수 있는 엔터티를 제약하여야 하며, 제약의 대상은 'AWS 서비스', 'AWS 계정(이 계정 또는 다른 AWS 계정)', '연동된 자격증명(웹 자격증명 또는 SAML 2.0연동)', '사용자 지정'을 통해서 한정할 수 있습니다.
- 역할을 사용할 수 있는 신뢰 대상을 한정 한 뒤에는, 실제 역할에 부여될 권한을 '사전에 정의된 행위'에 한정하여 지정합니다.



| 그림 3-1-9 | 역할 생성 – 권한 추가

– 역할이 생성 된 후에는 해당 역할의 세부 정보 화면을 통해서 역할 전환 링크를 확인하고, 해당 링크를 통해서 '사전에 정의된 행위'를 수행하도록 합니다.



| 그림 3-1-10 | 역할 생성 후 세부 정보

- **(AWS CLI)** CLI를 통해서 역할을 생성하기 위해서는 'iam create-role' 명령어를 사용할 수 있습니다.

```
aws iam create-role \
    --role-name Test-Role \
    --assume-role-policy-document file://Test-Role-Trust-Policy.json \
    --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Va
lue": "Seattle"}'
```

Output:

```
{
    "Role": {
        "Path": "/",
        "RoleName": "Test-Role",
        "RoleId": "AKIAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:role/Test-Role",
        "CreateDate": "2023-05-25T23:29:41+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "Statement1",
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "arn:aws:iam::123456789012:root"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        },
        "Tags": [
            {
                "Key": "Department",
                "Value": "Accounting"
            },
            {
                "Key": "Location",
                "Value": "Seattle"
            }
        ]
    }
}
```

| 그림 3-1-11 | CLI 명령 - 역할 생성

- 역할에 사전에 정의된 행위를 수행하도록 권한을 부여하기 위해서는 'iam attach-role-policy' 명령어를 사용할 수 있고 아래와 같이 역할명과 사전에 정의된 행위를 수행할 수 있도록 만들어진 정책의 AWS 리소스 이름(ARN)을 그 인자로 입력하여 생성합니다.

```
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \
    --role-name ReadOnlyRole
```

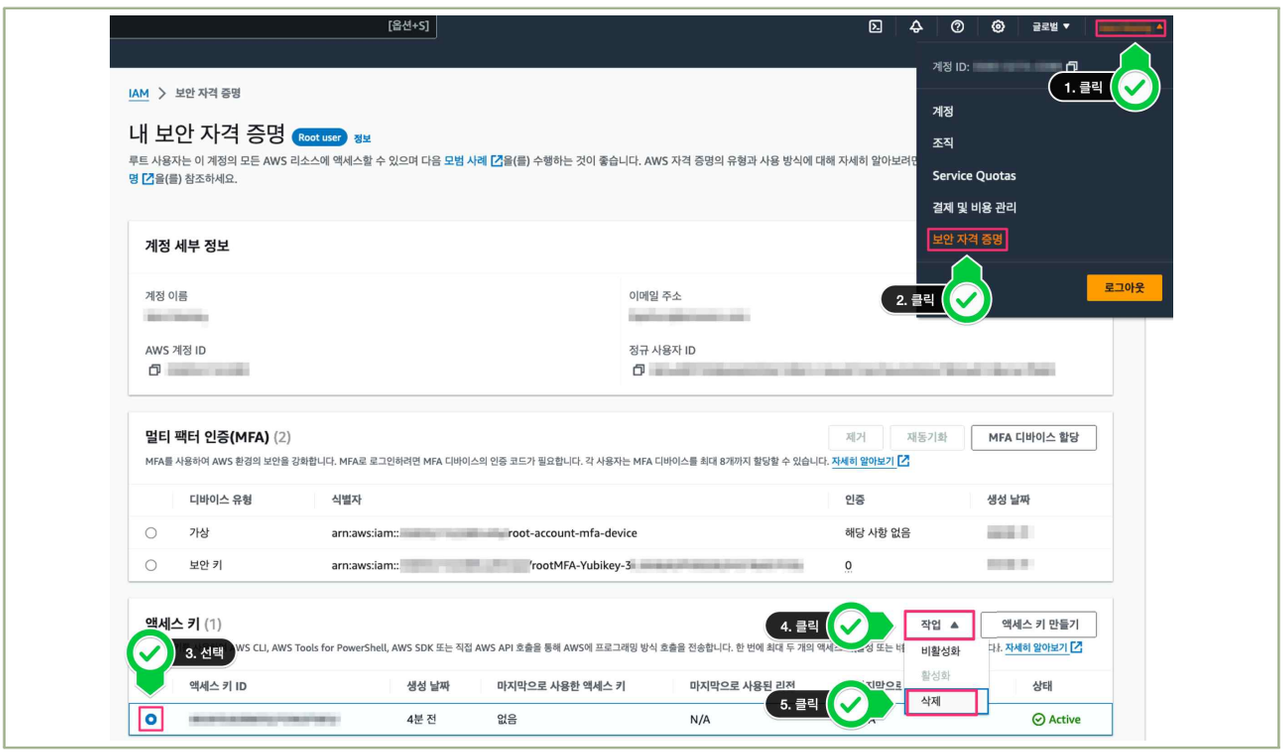
| 그림 3-1-12 | CLI 명령 - 역할에 정책 부여

3) 콘솔 최상위 관리자 계정 미사용

(1) 최상위 관리자 액세스 키 제거

- 콘솔 최상위 관리자(최초 가입계정 root)가 프로그램 방식의 API 호출을 할 수 없도록 액세스 키를 제거하도록 합니다.

- **(AWS 관리 콘솔)** 콘솔 최상위 관리자에 대한 권한 제거는 '콘솔 홈'→ root user 이름 클릭→ '보안 자격 증명'→ '액세스 키' 선택→ '작업'→ '삭제' 를 수행하여 할 수 있습니다.



| 그림 3-1-13 | 최상위 관리자(root) 액세스 키 제거

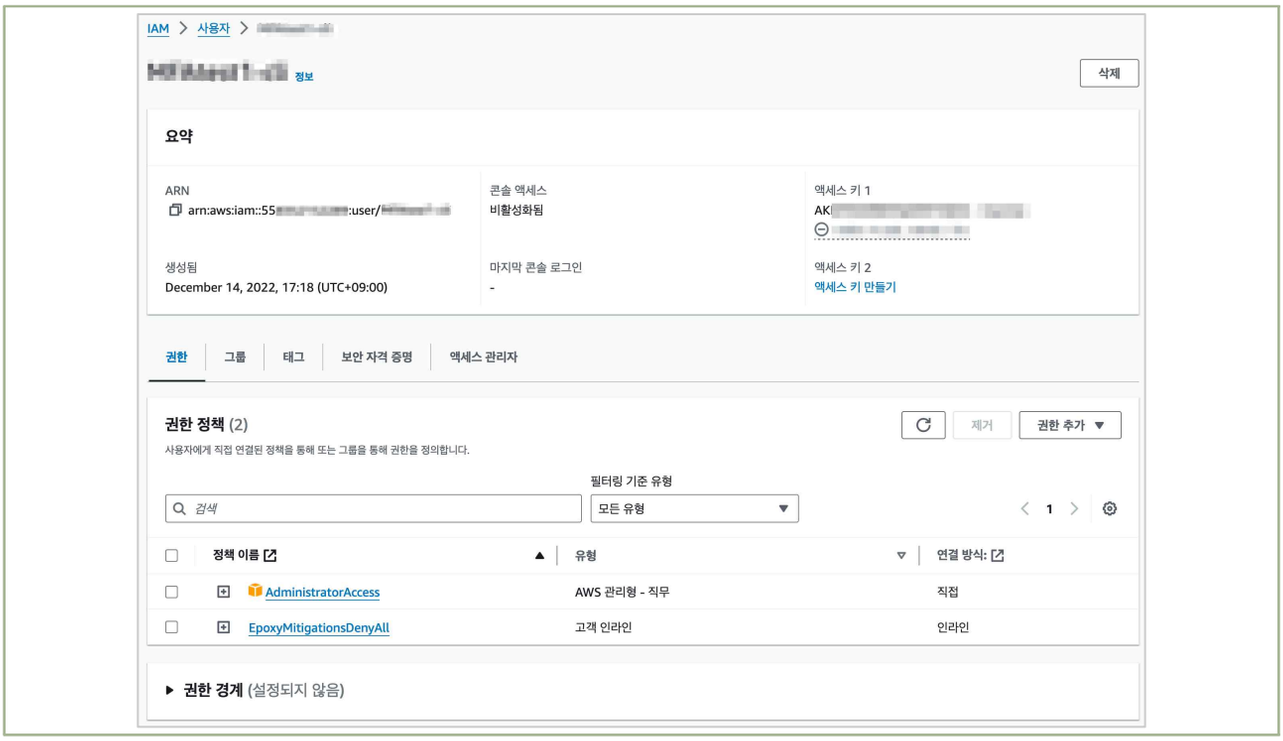
- **(AWS CLI)** 'aws iam delete-access-key' 명령을 통해서 root 사용자의 access key id를 지정하여 삭제 수행할 수 있습니다.

```
aws iam delete-access-key \
   --access-key-id AKIAIOSFODNN7EXAMPLE
```

| 그림 3-1-14 | CLI 명령 - 최상위 관리자(root) 액세스 키 제거

(2) 최상위 관리자 계정의 예외적 사용

- 예외적으로 최상위 관리자 계정의 사용이 필요한 경우에도 최상위 관리자 계정에서 운영을 하는 것이 아니라 신규 계정을 생성하고 필요한 권한을 부여하고 생성한 신규 계정을 활용하도록 합니다.
- 본 참고서에서는 관리자 권한이 필요한 경우를 예시로 합니다.



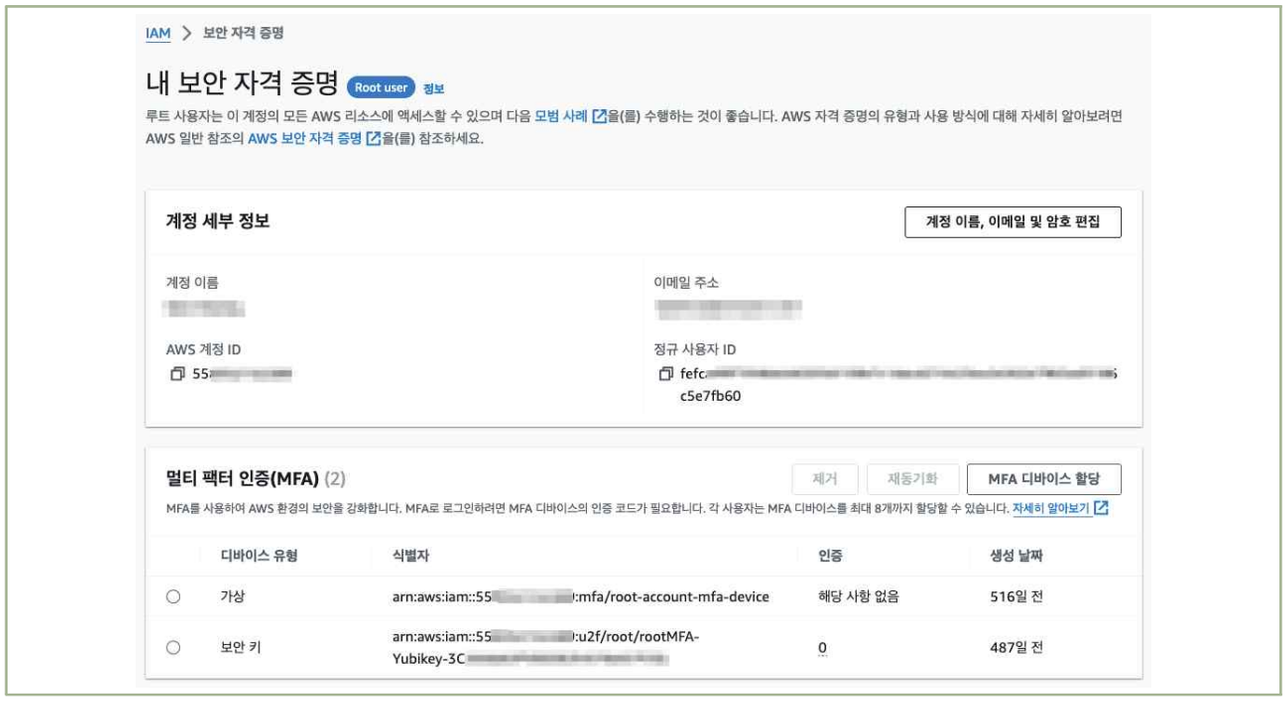
| 그림 3-1-15 | 신규 생성한 계정에 필요한 사용자 권한(ex. Admin) 부여

- 'aws iam attach-user-policy' 명령을 통해서 신규 계정에 'AdminstratorAccess' 정책을 할당합니다.

```
aws iam attach-user-policy \
    --policy-arn arn:aws:iam::aws:policy/AdministratorAccess \
    --user-name Alice
```

| 그림 3-1-16 | CLI 명령 - 신규 계정에 관리자 권한 부여

- 만약 불가피하게 최상위 관리자 계정을 이용해야 하는 특정 서비스가 있는 경우 반드시 Multi-Factor Authenticator 등 추가 인증 방식을 적용하고, 접속 IP에 대한 통제 등 강화된 보안 환경을 구성하여 이용하여야 합니다.



| 그림 3-1-17 | 최상위 관리자 계정에 MFA 등록

- 그리고 최상위 관리자 계정에 대한 접속은 SCP(서비스 제어 정책)를 활용하여 지정된IP를 제한하여 통제하도록 합니다.
- 'AWS Organizations'→ 대상 AWS 계정 선택→ '정책' 탭→ '연결'을 통해서 해당 계정 root 사용자에게 적용할 서비스 제어 정책을 선택하여 적용함. 이때 서비스 정책의 예시는 아래와 같이 허용할 IP 주소를 명시적으로 기입할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RestrictService4Root",
      "Effect": "Deny",
      "NotAction": [
        "iam:*",
        "sso:*",
        "account:*",
        "servicequotas:*",
        "budgets:*",
        "aws-portal:*",
        "cur:*",
        "billing:*"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringLike": {
          "aws:PrincipalArn": [
            "arn:aws:iam::*:root"
          ]
        },
        "NotIpAddress": {
          "aws:SourceIp": [
            "123.123.123.123/32",
            "234.234.234.234/32"
          ]
        }
      }
    },
    {
      "Sid": "AllowAll",
      "Effect": "Allow",
      "Action": [
        "*"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

| 그림 3-1-18 | IP통제를 위한 SCP

4 참고 사항

- 이용자가 개별적으로 이용중인 별도의 사용자 계정관리, 접근 관리 솔루션 적용이 가능하며 AWS Marketplace에서 관련 솔루션을 확인할 수 있습니다.

[문의처에 관한 정보]

1 기준

식별번호	기준	내용
3.2.	**이용자별 인증 수단 부여**	**클라우드 서비스를 이용하는 임직원(이용자)별 인증 수단을 할당하여야 한다.**

2 설명

- 클라우드 서비스를 이용하는 임직원(이용자)별 인증 수단을 부여하여야 하며, 필요 시 추가인증을 적용할 수 있어야 한다.(외부직원 포함)

  - 예시

    1) IAM(Identity and Access Management) 기능 등을 이용하여 이용자별 인증수단 적용
    2) 업무 중요도에 따른 MFA 추가 인증(OTP, 바이오 인증 등) 고려

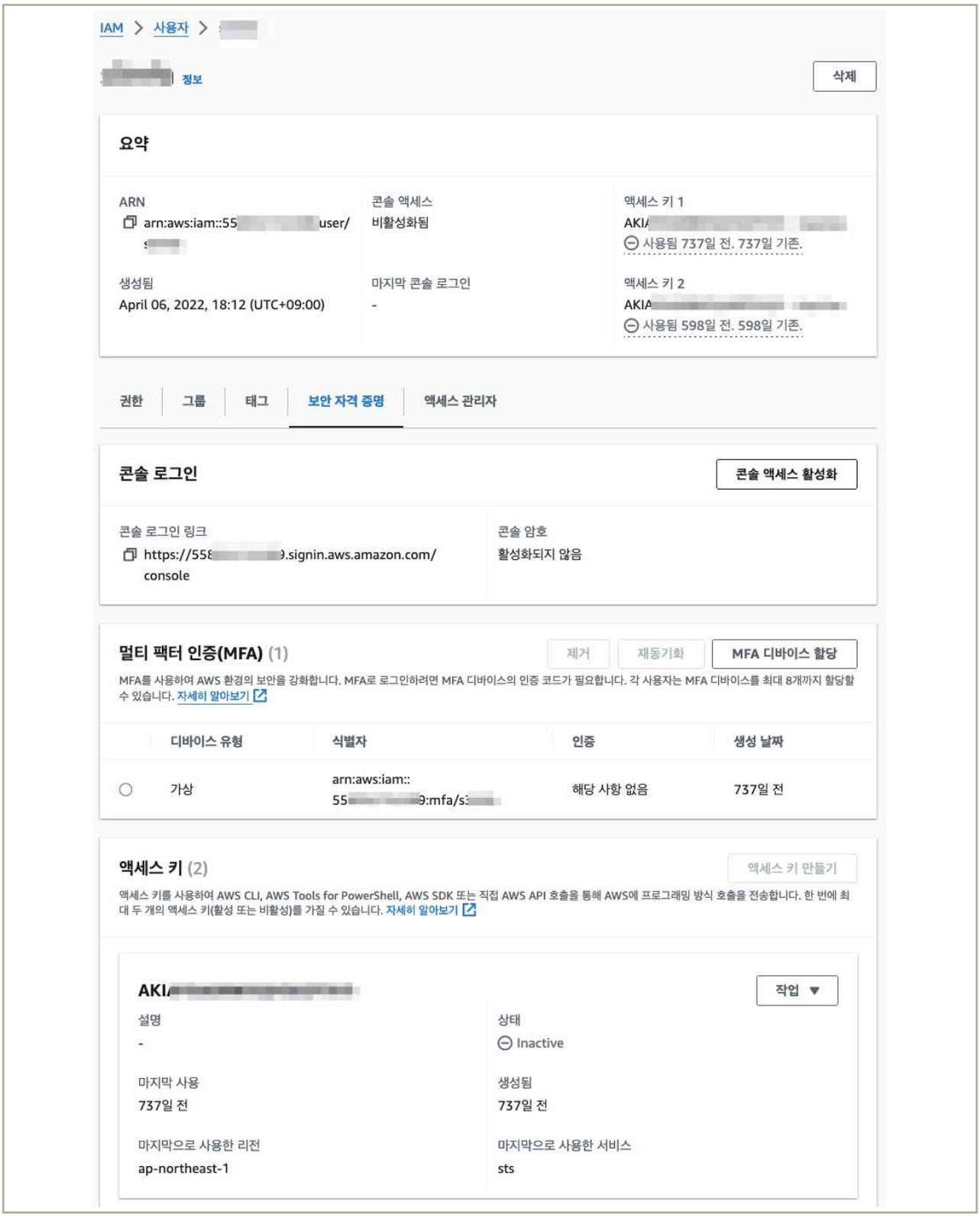
3 우수 사례

1) IAM 기능을 이용하여 이용자별 인증수단 적용

(1) 이용자별 인증수단 적용

- 각 이용자는 고유의 AWS Management Console 액세스 암호 또는 프로그래밍 방식의 요청을 할 수 있도록 하는 개별 액세스 키를 생성하고 독립적으로 사용할 수 있습니다.

- **(AWS 관리 콘솔)** 이용자별 인증수단 적용은 '콘솔 홈'→ '서비스 명 IAM 입력'→ '사용자'→ 해당 사용자명 클릭→ '보안 자격 증명'탭에서 확인하고 변경/추가할 수 있습니다.

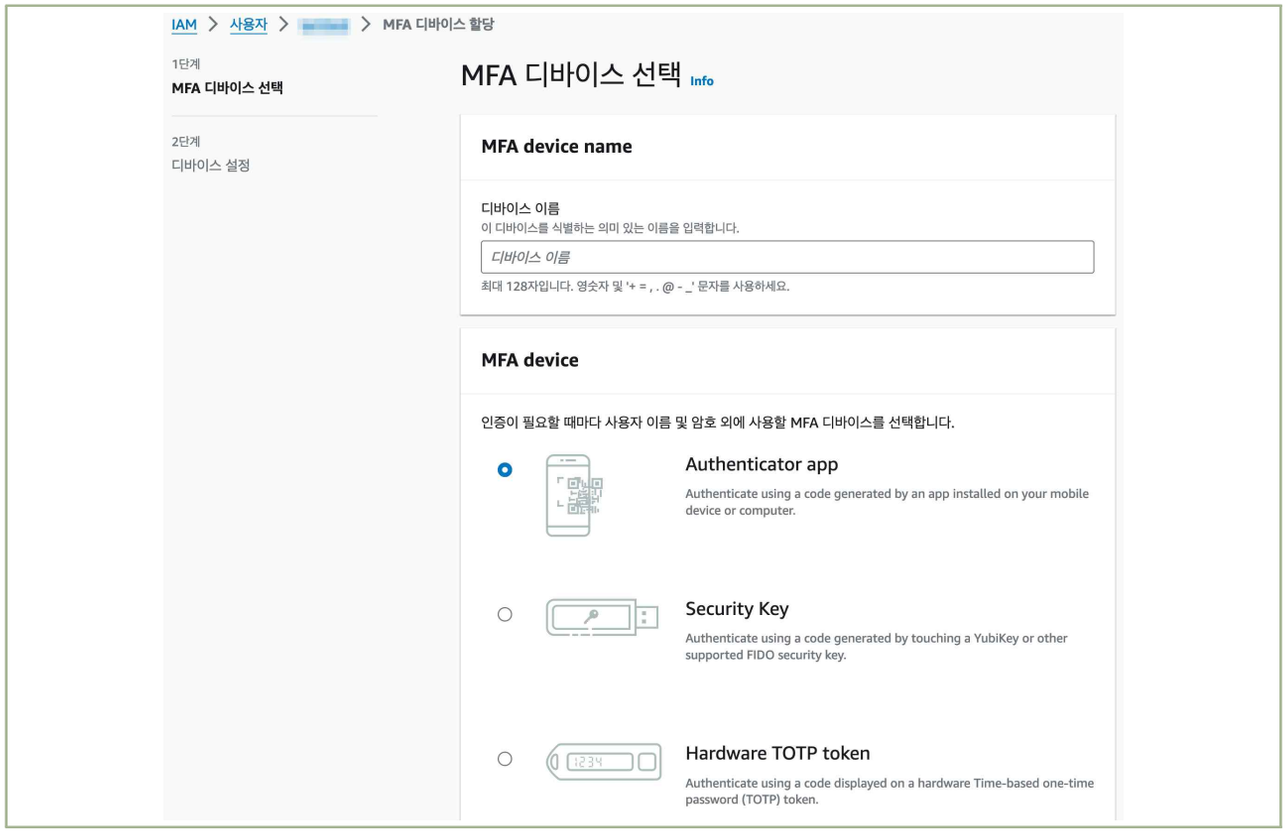


| 그림 3-2-1 | 이용자별 인증수단 적용

- **(AWS CLI)** 이용자별 콘솔 로그인 프로파일, 프로그램 접근을 위한 Access Key 생성은 “3.1 클라우드 계정관리, 우수사례 1)”을 참조하여 수행할 수 있습니다.

2) 업무 중요도에 따른 MFA 추가 인증 고려

- **(AWS 관리 콘솔)** MFA 추가 인증을 위해서 '콘솔 홈'→ '서비스 명 IAM 입력'→ '사용자'→ 해당 사용자명 클릭→ '보안 자격 증명' → '멀티 팩터 인증(MFA)' → 'MFA 디바이스 할당'에서 등록된 MFA 장치를 확인하고 변경/추가할 수 있습니다.



| 그림 3-2-2 | MFA 추가 인증 등록

- **(AWS CLI)** 'aws iam create-virtual-mfa-device' 명령을 통해 가상의 MFA 장치를 생성하고, 'aws iam enable-mfa-device' 명령을 통해 이용자에게 해당 MFA 장치를 할당하도록 수행할 수 있습니다.

```
aws iam create-virtual-mfa-device \
    --virtual-mfa-device-name BobsMFADevice \
    --outfile C:/QRCode.png \
    --bootstrap-method QRCodePNG
```

Output:

```
{
    "VirtualMFADevice": {
        "SerialNumber": "arn:aws:iam::210987654321:mfa/BobsMFADevice"
}
```

| 그림 3-2-3 | CLI 명령 - 추가 인증 장치 생성

```
aws iam enable-mfa-device \
    --user-name Bob \
    --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \
    --authentication-code1 123456 \
    --authentication-code2 789012
```

| 그림 3-2-4 | CLI 명령 - 등록한 MFA를 사용자에게 연동

4 참고 사항

- 별도의 SSO(Single Sign-On)를 사용하는 경우에는 IAM Identity Center를 활용하여 SSO 사용자의 인증수단을 통해서 접속할 수 있도록 구현 가능함.

  https://docs.aws.amazon.com/ko_kr/singlesignon/latest/userguide/what-is.html

1 기준

식별번호	기준	내용
3.3.	**인사 변경 사항 발생 시 계정 관리**	**이용자의 인사 변경(휴직, 전출, 퇴직 등) 발생 시 지체 없이 이용자 계정 삭제, 중지 등의 조치를 수행하여야 한다.**

2 설명

- 클라우드를 이용하는 임직원의 인사 변경 사항 발생 시 지체 없이 이용자 계정 삭제, 중지 등의 조치를 수행하여야 한다.

  - 예시

    1) 인사 변경이 발생한 이용자의 계정 삭제 또는 중지

    2) 인사 변경이 발생한 이용자가 공용 계정 이용 시 계정 비밀번호 변경

3 우수 사례

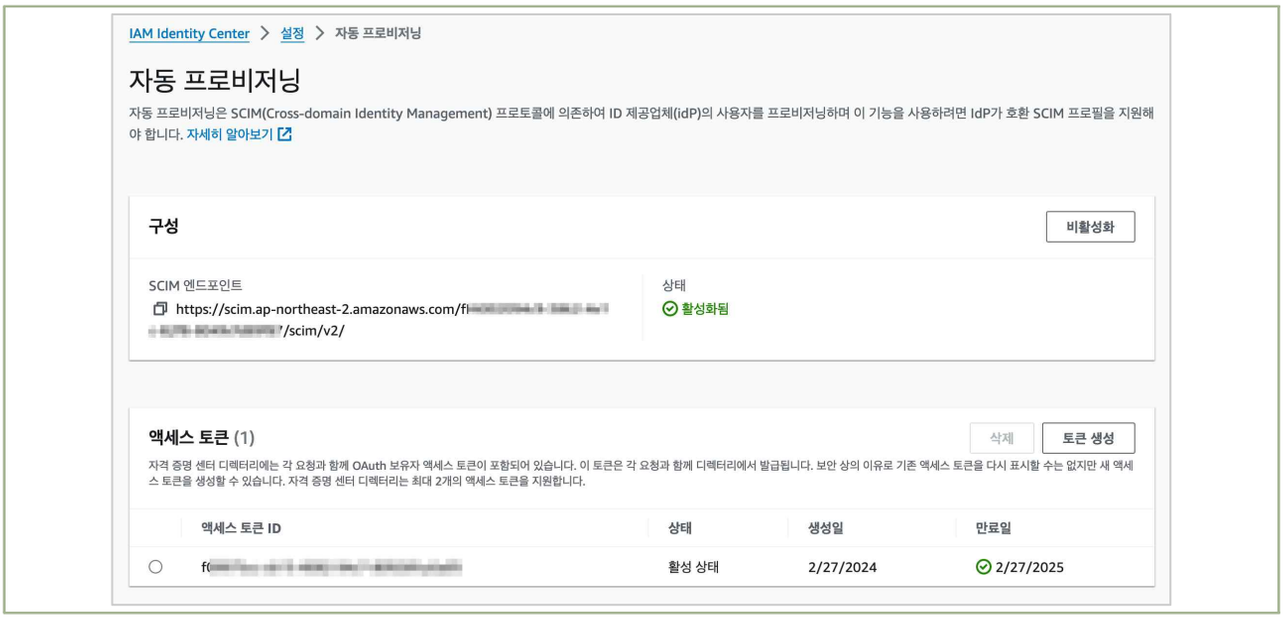
1) 인사 변경 발생 시 이용자 계정 삭제 및 중지

- 인사 변경이 발생한 경우, 해당 내용이 자동으로 반영될 수 있도록 IAM Identity center를 통해서 회사의 인사 변경이 반영되는 idP(외부 자격 증명 공급자)를 연동하도록 구성합니다.

- **(AWS 관리 콘솔)** 회사의 인사 변경이 반영되는 idP의 SAML 메타데이터를 사용하여, IAM Identity Center의 자격증명 소스로 등록하기 위해서 '콘솔 홈'→ '서비스 명 IAM Identity Center 입력'→ '설정'→ '자격 증명 소스 변경'탭을 통해서 설정을 진행할 수 있습니다.

| 그림 3-3-1 | 자격 증명 소스 선택

- IAM Identity Center에 회사의 idP를 자격증명 소스로 등록한 뒤에는, 인사 정보 변경이 반영되도록 자동 프로비저닝을 구성함. 이를 위해서 'IAM Identity Center 콘솔 홈'→ '설정'→ '자동 프로비저닝' 활성화를 수행합니다.



| 그림 3-3-2 | 자동 프로비저닝 활성화

(AWS CLI)

- AWS IAM Identity Center는 CLI를 이용한 연동설정을 따로 지원하지 않음으로 콘솔을 통해서 구성하도록 해야 합니다.
- 회사의 인사 시스템에서 특정 이용자에 대한 계정 삭제를 수행하도록 아래의 AWS API를 활용하여 직접 API Call을 호출하도록 할 수 있으나 권장 되지 않으며 IAM Identity Center를 통해서 통합 관리하는 것이 모범사례로 권장됩니다.

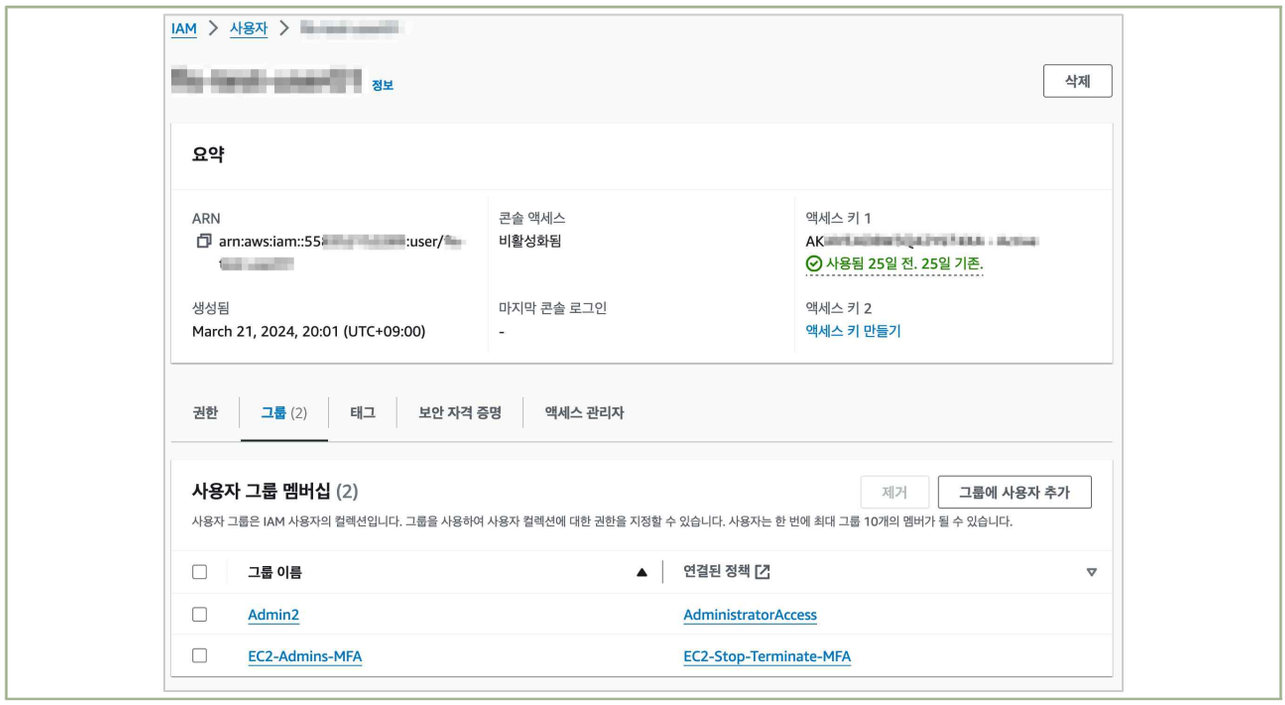
```
aws iam delete-user \
    --user-name Bob
```

| 그림 3-3-3 | CLI 명령 - IAM 사용자 삭제

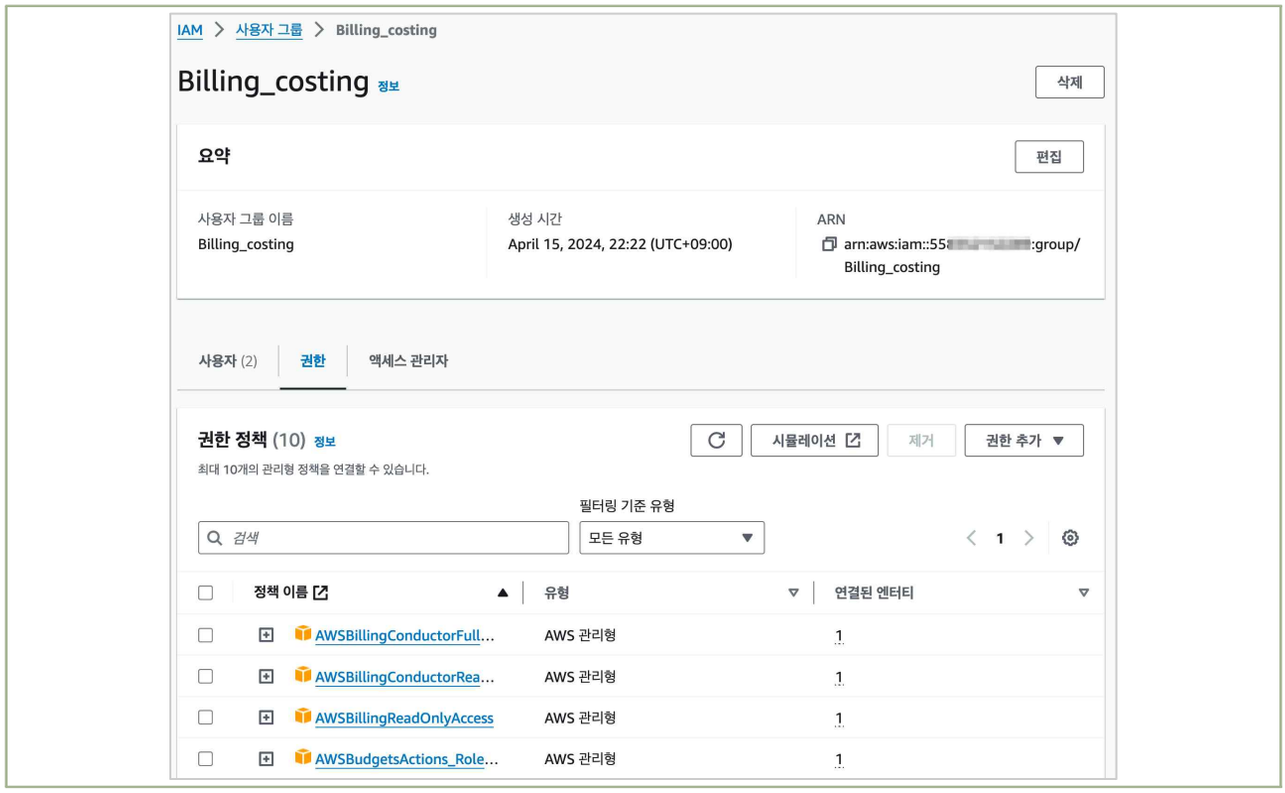
2) 인사 변경이 발생한 이용자가 공용계정 이용 시 계정 비밀번호 변경

- 각 이용자는 구별된 개별 계정을 사용하도록 구성하며, 공용 계정은 생성하지 않도록 합니다. 공통의 업무를 위하여 공통의 권한이 필요한 경우, Group을 활용하여 해당 그룹에 속한 이용자가 해당 그룹에 속한 권한을 통해 업무를 수행 하도록 합니다.

- **(AWS 관리 콘솔)** ‘콘솔 홈’→ ’서비스 명 IAM 입력’→ ‘이용자 계정(사용자명) 클릭’→ ‘그룹’ 탭을 통해서 각 이용자 별로 그룹을 설정하거나, ‘콘솔 홈’→ ’서비스 명 IAM 입력’→ ‘사용자 그룹’ 선택→ ‘그룹 이름’을 클릭하여 이용자 및 권한을 조정할 수 있습니다.



| 그림 3-3-4 | IAM 사용자 관리



| 그림 3-3-5 | IAM 사용자 그룹 관리

- **(AWS CLI)** 이용자를 특정 그룹에 포함시키는 방법은 “3.1 클라우드 계정관리, 우수사례 1)”을 참조 바랍니다.

4 참고 사항

- 여러 다양한 idP와 연계하는 방법은 아래의 링크를 통해서 확인 할 수 있습니다.
  - Microsoft Active Directory
    https://docs.aws.amazon.com/ko_kr/singlesignon/latest/userguide/gs-ad.html
  - Okta
    https://docs.aws.amazon.com/ko_kr/singlesignon/latest/userguide/gs-okta.html
  - Google Workspace
    https://docs.aws.amazon.com/ko_kr/singlesignon/latest/userguide/gs-gwp.html
  - CyberArk
    https://docs.aws.amazon.com/ko_kr/singlesignon/latest/userguide/cyberark-idp.html
  - OneLogin
    https://docs.aws.amazon.com/ko_kr/singlesignon/latest/userguide/onelogin-idp.html
  - Ping identity
    https://docs.aws.amazon.com/ko_kr/singlesignon/latest/userguide/pingidentity.html

1 기준

식별번호	기준	내용
3.4.	**클라우드 가상자원 관리 시스템 관리자 권한 추가인증 적용**	**클라우드 서비스 관리자 권한으로 로그인 시 추가인증 수단을 적용하여야 한다.**

2 설명

○ 클라우드 환경(콘솔 등)에 관리자 권한으로 로그인 시 추가인증 수단을 적용하여야 한다.

- 예시

1) 이메일 인증
2) SMS 인증
3) 별도 인증도구(OTP, 바이오 인증 등) 활용 등

3 우수 사례

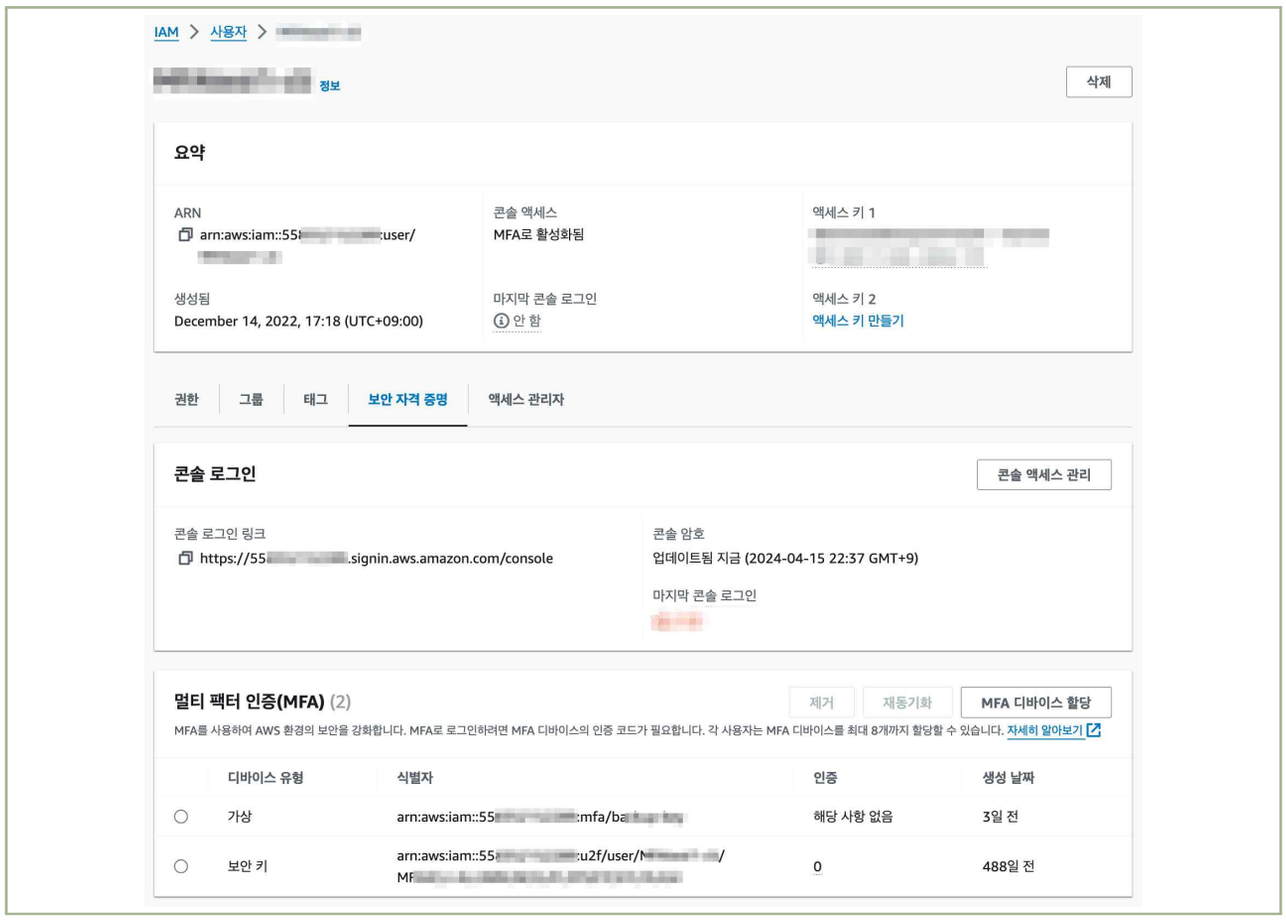
1) 클라우드 환경에 관리자 권한 로그인 시 MFA 적용

- AWS 환경에 관리자 권한으로 로그인 시에는

(1) 등록된 개별 사용자 계정(등록된 이메일 등)
(2) 콘솔 접속을 위한 강력한 암호 정책이 적용된 password
(3) MFA 장치를 통한 추가 인증

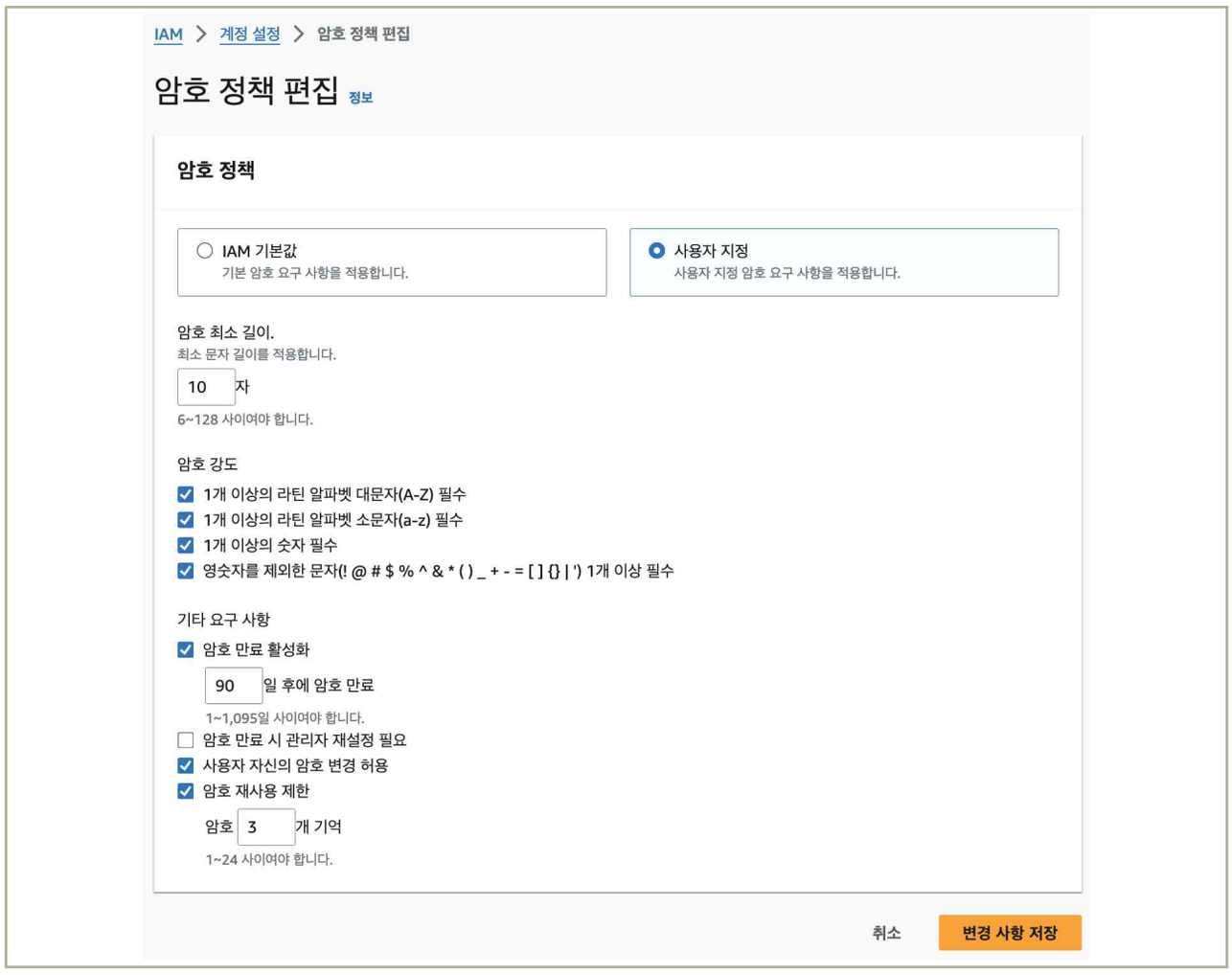
절차 후 이용하도록 합니다.

○ **(AWS 관리 콘솔)** '콘솔 홈'→ '서비스 명 IAM 입력'→ '사용자'→ 해당 사용자명 클릭→ '보안 자격 증명' → '멀티 팩터 인증(MFA)' → 'MFA 디바이스 할당'에서 등록된 MFA 장치를 확인하고 변경/추가할 수 있습니다.



| 그림 3-4-1 | 관리자 권한을 가진 사용자

- ‘콘솔 홈’→ ’서비스 명 IAM 입력’→ ‘계정 설정’→ ‘암호 정책 편집’ 을 통해서 ‘암호의 최소 길이’, ‘1개 이상의 알파벳 대문자’, ‘1개 이상의 알파벳 소문자’, ‘1개 이상의 숫자’, ‘1개 이상의 특수문자 (! @ # $ % ^ & * ( ) _ + - = [ ] {} | ')’, ‘특정 기간 이후 암호 만료 지정’, ‘기존에 사용한 암호의 재사용 제한’ 과 같은 강력한 암호 정책을 적용하도록 합니다.



| 그림 3-4-2 | 암호 정책 수정

(AWS CLI)

- AWS CLI를 이용하여 관리자 권한을 가진 이용자에게 MFA를 등록하는 방법은 “3.2 이용자별 인증 수단 부여, 우수사례 2)”를 참조 바랍니다.
- 계정의 암호 정책 ‘aws iam update-account-password-policy’ 명령을 통해서 할 수 있습니다.

```
  update-account-password-policy
[--minimum-password-length <value>]
[--require-symbols | --no-require-symbols]
[--require-numbers | --no-require-numbers]
[--require-uppercase-characters | --no-require-uppercase-characters]
[--require-lowercase-characters | --no-require-lowercase-characters]
[--allow-users-to-change-password | --no-allow-users-to-change-password]
[--max-password-age <value>]
[--password-reuse-prevention <value>]
[--hard-expiry | --no-hard-expiry]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
[--debug]
[--endpoint-url <value>]
[--no-verify-ssl]
[--no-paginate]
[--output <value>]
[--query <value>]
[--profile <value>]
[--region <value>]
[--version <value>]
[--color <value>]
[--no-sign-request]
[--ca-bundle <value>]
[--cli-read-timeout <value>]
[--cli-connect-timeout <value>]
```

| 그림 3-4-3 | IAM 패스워드 정책 설정

4 참고 사항

1 기준

식별번호	기준	내용
3.5.	**클라우드 가상자원 관리 시스템 로그인 규칙 수립**	**이용자 가상자원 관리 시스템 접근 계정에 대한 안전한 로그인 규칙을 수립하여야 한다.**

2 설명

- 이용자는 패스워드 무작위 대입 공격 등에 대응하기 위해 가상자원 관리 시스템 계정에 대한 안전한 로그인 규칙을 수립하여야 한다.

  - 예시

    1) 로그인 오류에 따른 보안 통제 방안 수립 등

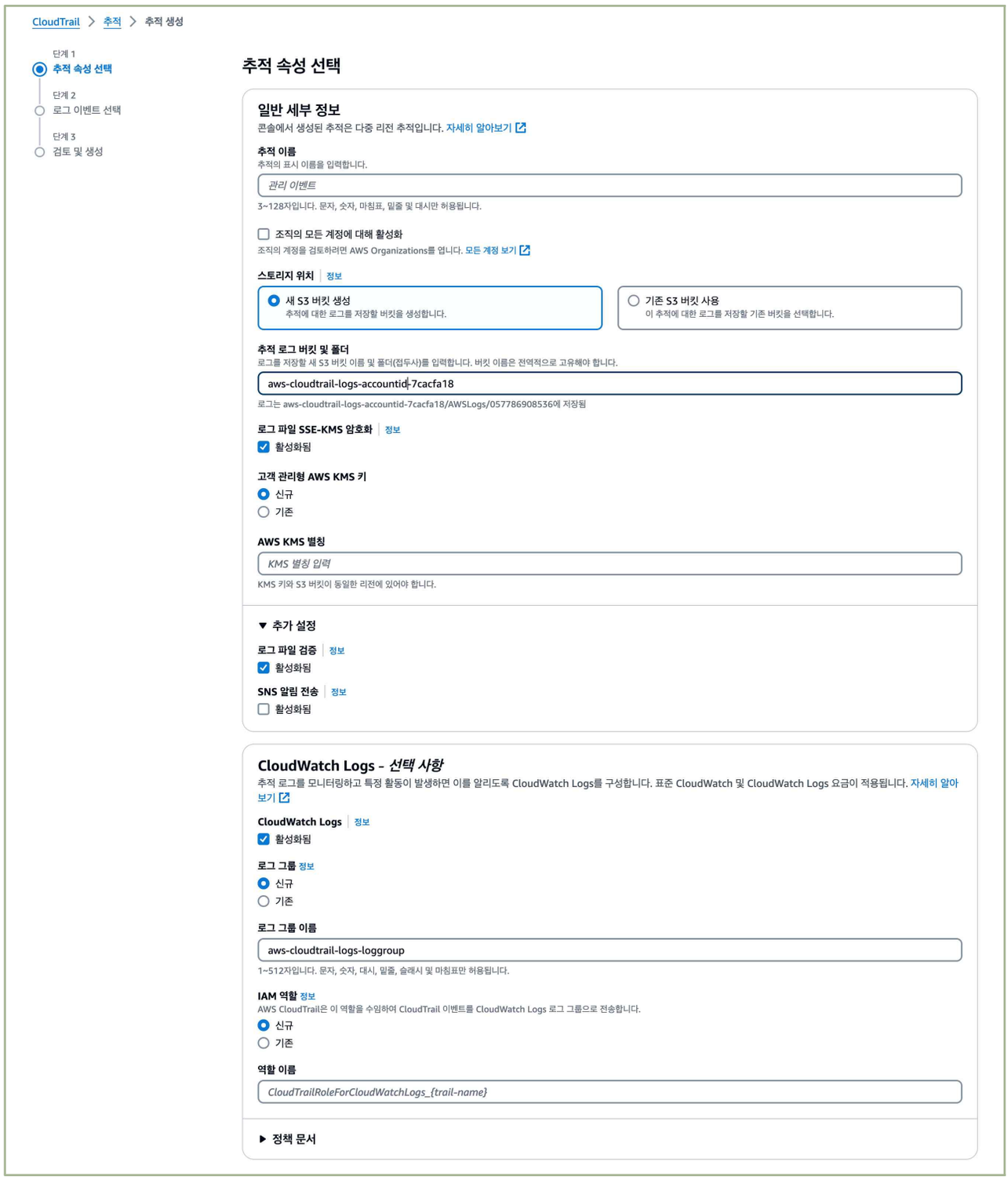
3 우수 사례

1) 로그인 오류에 따른 보안 통제 방안

(1) CloudTrail 생성

- AWS에서 로그인 오류에 따른 별도 조치를 관리할 수 있는 기능은 제공하지 않지만 CloudTrail 이벤트 기록을 통해 로그인 오류 확인은 가능하며 이를 위해 CloudTrail 추적 생성이 필요 합니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → 'CloudTrail' → '추적' → '추적 생성' 으로 CloudTrail을 생성하여 AWS API 호출 이벤트를 로깅할 수 있습니다.

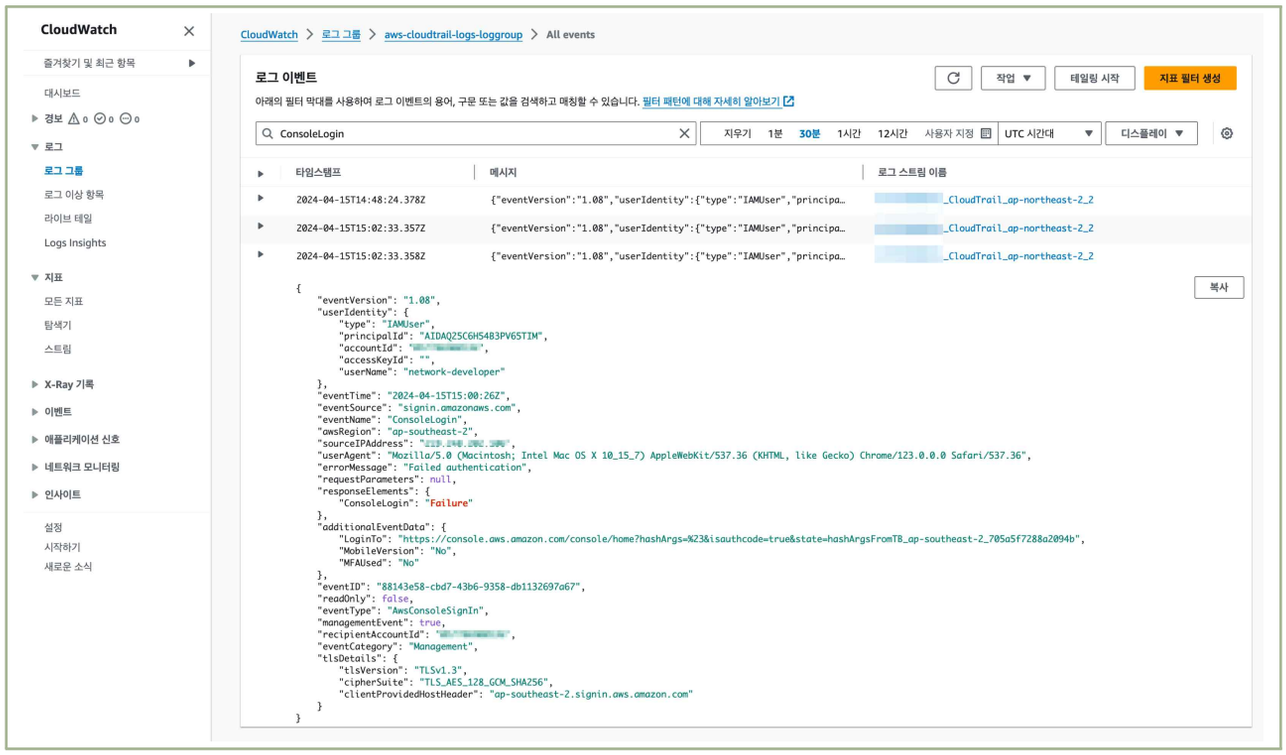


| 그림 3-5-1 | CloudTrail 추적 생성

- 기본적으로 CloudTrail 추적은 S3버킷을 지정하여 저장하며 추가적으로 CloudWatch Logs저장이 가능합니다.
- CloudWatch Logs에 저장할 경우CloudWatch로그 그룹에서 이벤트 확인이 가능하며 지표 필터 및 알람 생성을 통해 로그인 오류 시 알람 설정을 할 수 있습니다.

(2) CloudTrail 이벤트 확인

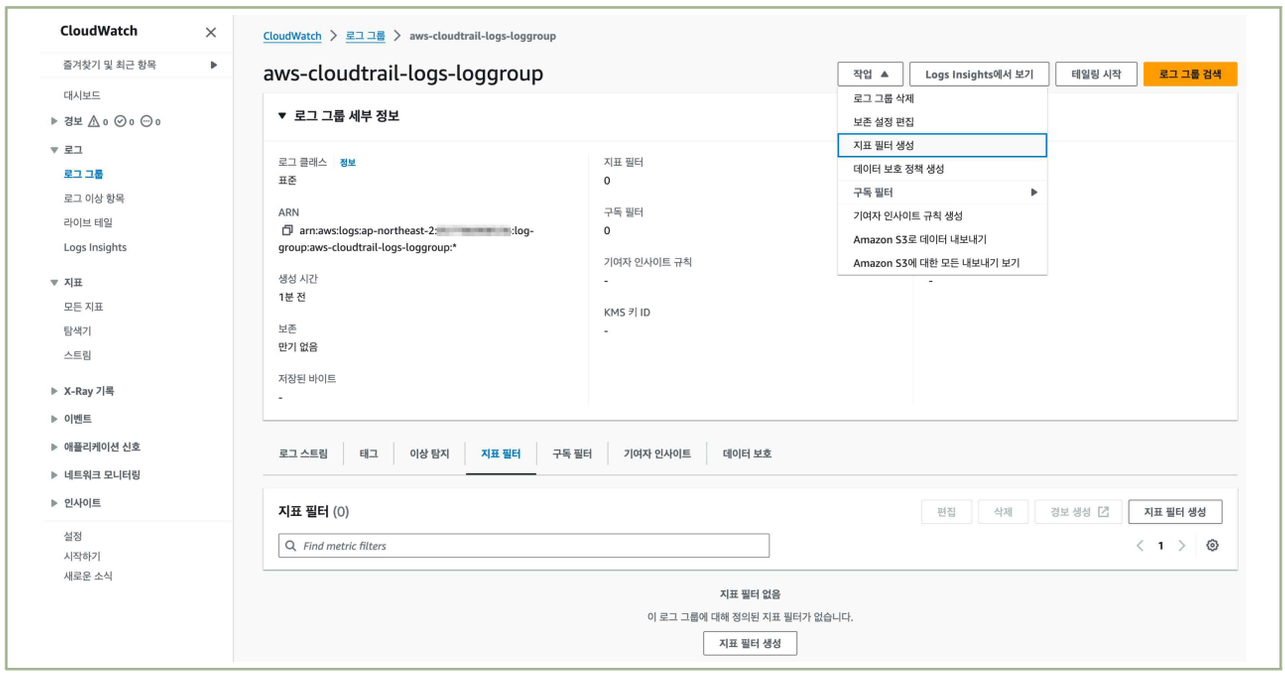
- **(AWS 관리 콘솔)** '콘솔 홈' → 'CloudWatch' → '로그 그룹' → '모든 로그 스트림 검색'에서 필터 조건을 'ConsoleLogin'으로 검색하여 로그 이벤트를 확인할 수 있습니다.



| 그림 3-5-2 | CloudWatch 로그 그룹에서 이벤트 확인

(3) CloudWatch 지표 생성

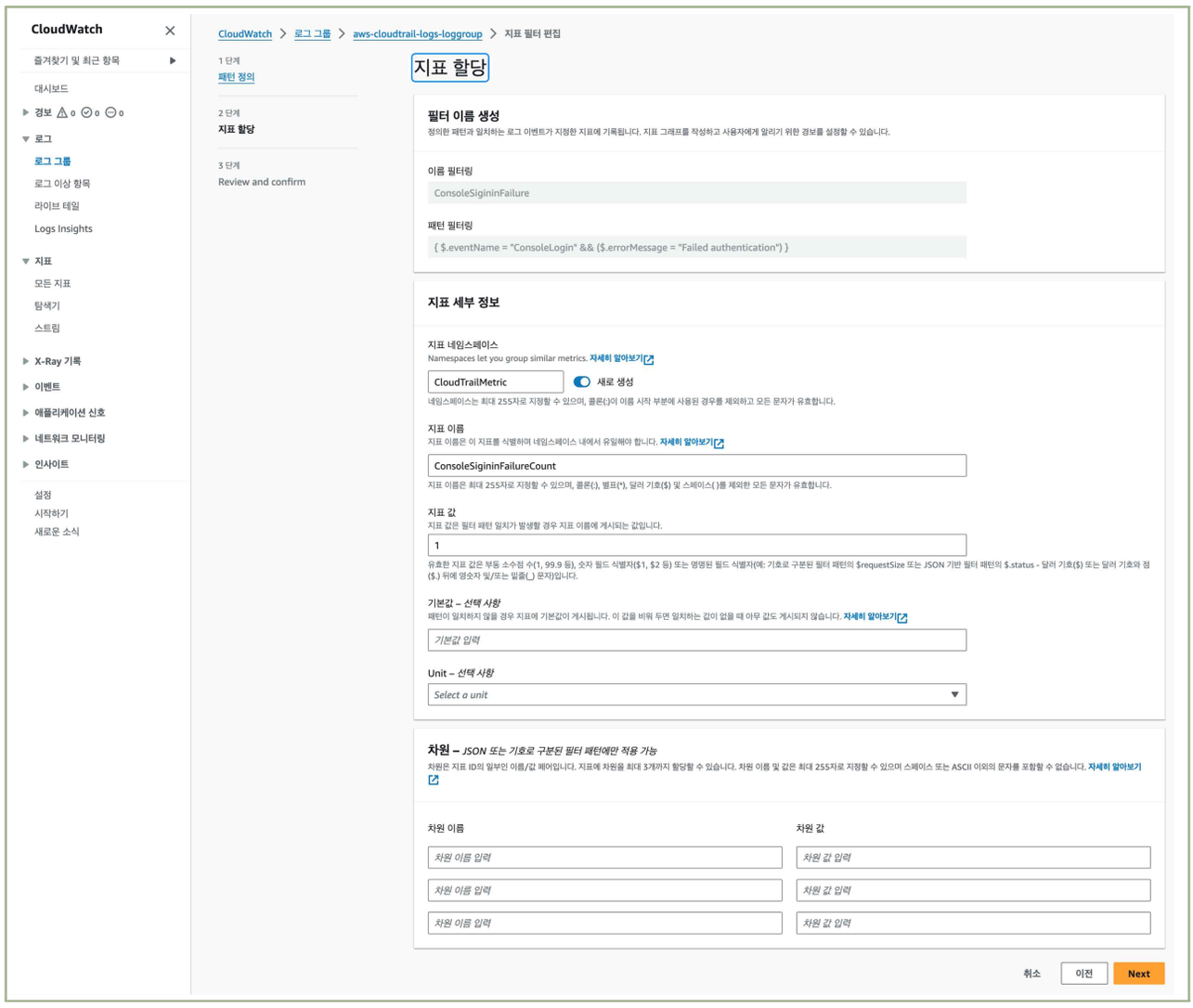
- **(AWS 관리 콘솔)** '콘솔 홈' → 'CloudWatch' → '로그 그룹' → '작업'에서 '지표 필터 생성' 을 클릭하여 지표 필터를 생성할 수 있습니다.



| 그림 3-5-3 | CloudWatch 로그 그룹에서 지표 필터 생성

- CloudWatch 로그 그룹에서 지표 필터를 생성하고 해당 지표 필터에 경보 생성하여 전달이 필요한 내용을 Amazon SNS 서비스를 활용하여 알람을 받을 수 있습니다.
- 로그인에 실패한 IAM 사용자의 경우 다음의 패턴 필터링 입력하여 지표 필터를 할 수 있습니다.

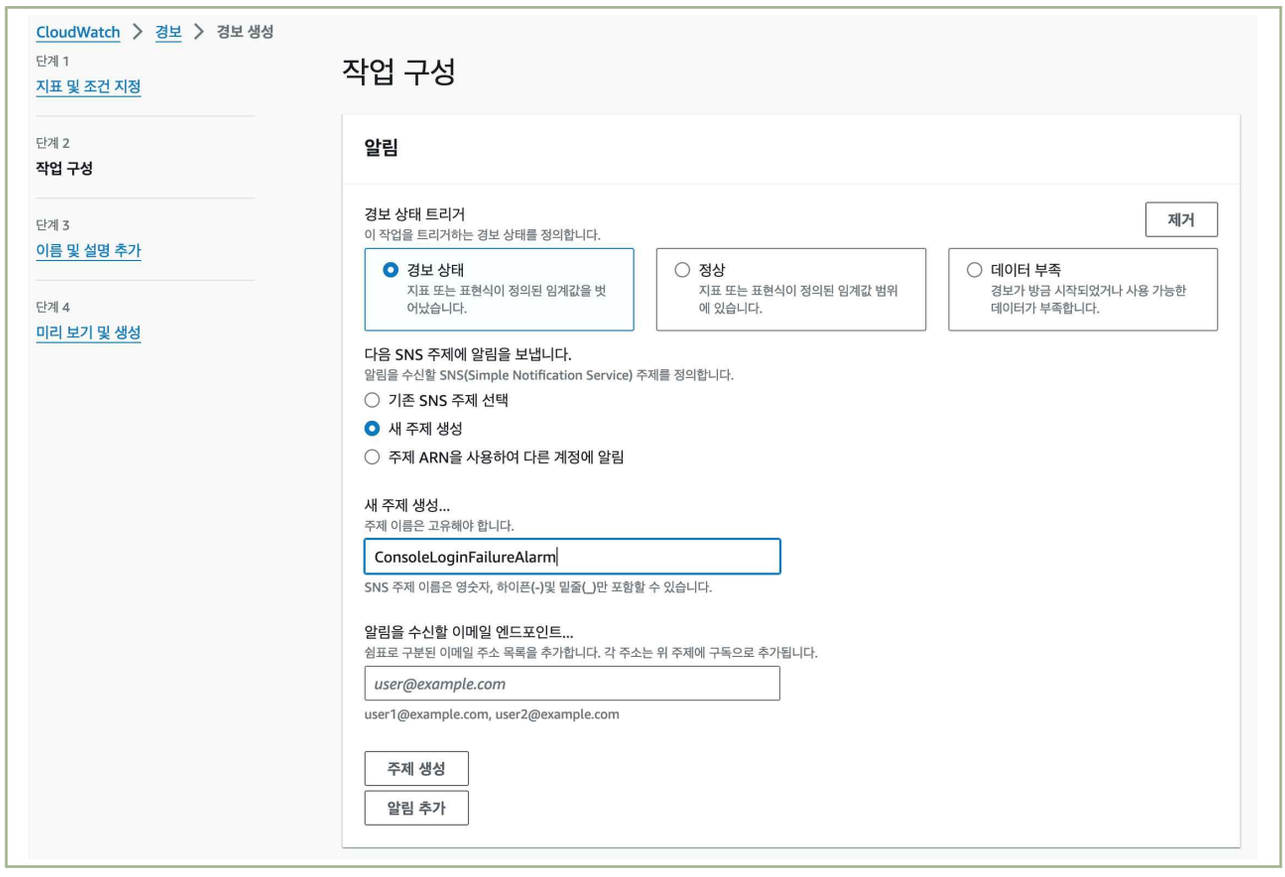
```
{ ($.eventName = “ConsoleLogin”) && ($.errorMessage = "Failed authentication") }
```



| 그림 3-5-4 | CloudWatch 로그 그룹에서 지표 필터 생성

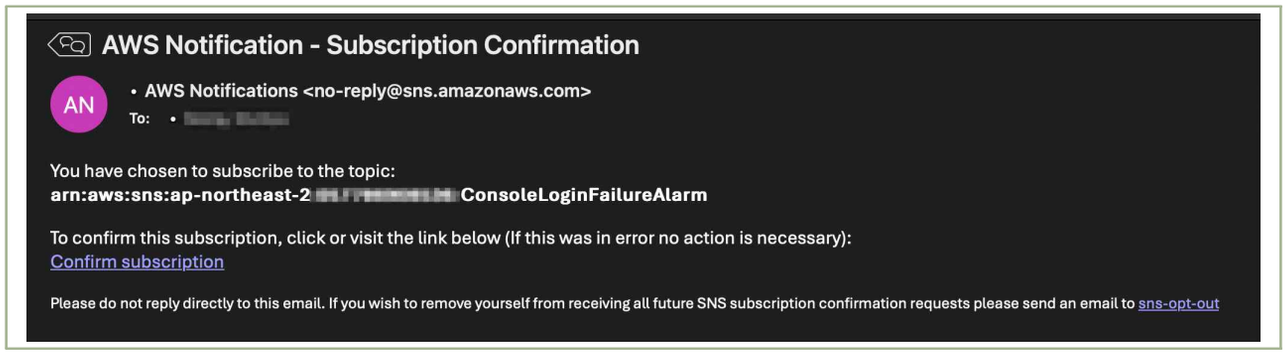
(4) CloudWatch 경보 생성

- **(AWS 관리 콘솔)** '콘솔 홈' → 'CloudWatch' → '로그 그룹' → '지표 필터'에서 '경보 생성' 버튼을 클릭하여 경보 생성할 수 있습니다.



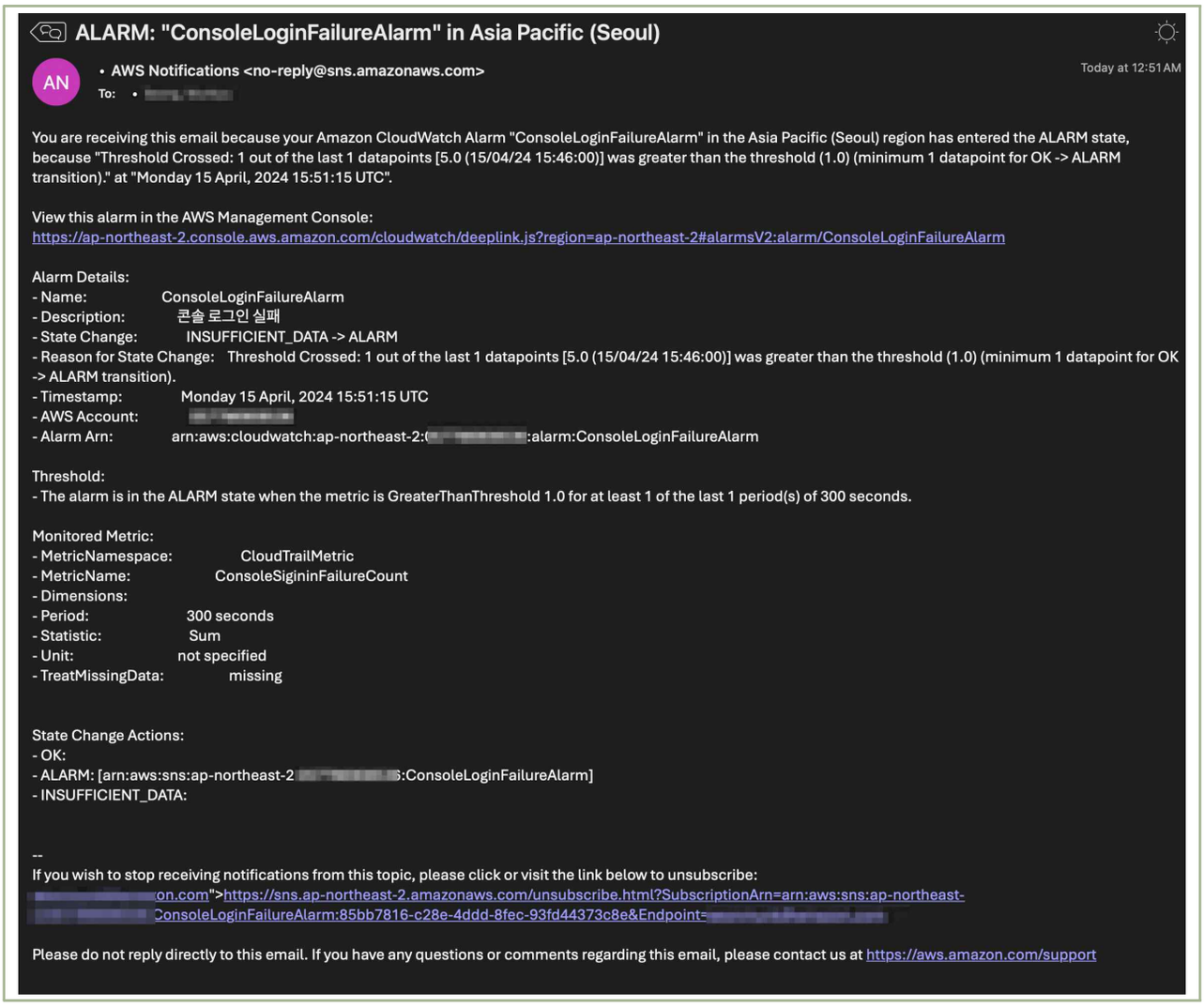
| 그림 3-5-5 | CloudWatch 경보 생성

- SNS 주제의 경우 미리 만들지 않고 '새 주제 생성'으로 즉시 생성할 수 있습니다.



| 그림 3-5-6 | Amazon SNS 알림 구독 승인 요청 메일

- SNS 알림 구독을 승인하면 이후 해당 메일로 설정된 경보라 송신되어 이메일로 알람을 수신할 수 있습니다.



ALARM: "ConsoleLoginFailureAlarm" in Asia Pacific (Seoul)

AN • AWS Notifications <no-reply@sns.amazonaws.com> Today at 12:51 AM
To: • [redacted]

You are receiving this email because your Amazon CloudWatch Alarm "ConsoleLoginFailureAlarm" in the Asia Pacific (Seoul) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [5.0 (15/04/24 15:46:00)] was greater than the threshold (1.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Monday 15 April, 2024 15:51:15 UTC".

View this alarm in the AWS Management Console:
https://ap-northeast-2.console.aws.amazon.com/cloudwatch/deeplink.js?region=ap-northeast-2#alarmsV2:alarm/ConsoleLoginFailureAlarm

Alarm Details:
- Name: ConsoleLoginFailureAlarm
- Description: 콘솔 로그인 실패
- State Change: INSUFFICIENT_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [5.0 (15/04/24 15:46:00)] was greater than the threshold (1.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Monday 15 April, 2024 15:51:15 UTC
- AWS Account: [redacted]
- Alarm Arn: arn:aws:cloudwatch:ap-northeast-2:[redacted]:alarm:ConsoleLoginFailureAlarm

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanThreshold 1.0 for at least 1 of the last 1 period(s) of 300 seconds.

Monitored Metric:
- MetricNamespace: CloudTrailMetric
- MetricName: ConsoleSigninFailureCount
- Dimensions:
- Period: 300 seconds
- Statistic: Sum
- Unit: not specified
- TreatMissingData: missing

State Change Actions:
- OK:
- ALARM: [arn:aws:sns:ap-northeast-2[redacted]:ConsoleLoginFailureAlarm]
- INSUFFICIENT_DATA:

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
[redacted]on.com">https://sns.ap-northeast-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:ap-northeast-[redacted]ConsoleLoginFailureAlarm:85bb7816-c28e-4ddd-8fec-93fd44373c8e&Endpoint=[redacted]

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

| 그림 3-5-7 | 로그인 오류에 대한 경보 이메일 수신

- **(AWS CLI)** 'API 환경을 통한 CloudTrail 추적 생성' → 'create-trail' 명령어를 통해 계정에서 발생하는 API 이벤트 기록을 로깅하여 로그인 오류 이벤트 확인 가능합니다.

```
aws cloudtrail create-trail --name Trail1 --s3-bucket-name my-bucket --is-multi-region-trail
```

Output:

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "Trail1",
    "TrailARN": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail1",
    "LogFileValidationEnabled": false,
    "IsMultiRegionTrail": true,
    "S3BucketName": "my-bucket"
}
```

| 그림 3-5-8 | CLI 명령 - CloudTrail 추적 생성

- CloudWatch 지표 필터 및 경보 생성에 대한 CLI는 제공하지 않습니다.

4 참고 사항

- AWS Management Console 로그인 이벤트는 아래 문서를 참고 바랍니다.
  - https://docs.aws.amazon.com/ko_kr/awscloudtrail/latest/userguide/cloudtrail-event-reference-aws-console-sign-in-events.html

1 기준

식별번호	기준	내용
3.6.	**계정 비밀번호 규칙 수립**	**클라우드 가상자원 관리 시스템 로그인 계정 생성 시 비밀번호 규칙을 수립하여 적용하여야 한다.**

2 설명

- 클라우드 가상자원 관리시스템 접근 가능한 계정 생성 시 안전한 비밀번호 규칙을 수립하여 적용하여야 한다.

  - 예시

    1) 제3자가 쉽게 유추할 수 없는 비밀번호 작성 규칙 수립

3 우수 사례

1) 제3자가 쉽게 유추할 수 없는 비밀번호 작성 규칙 수립

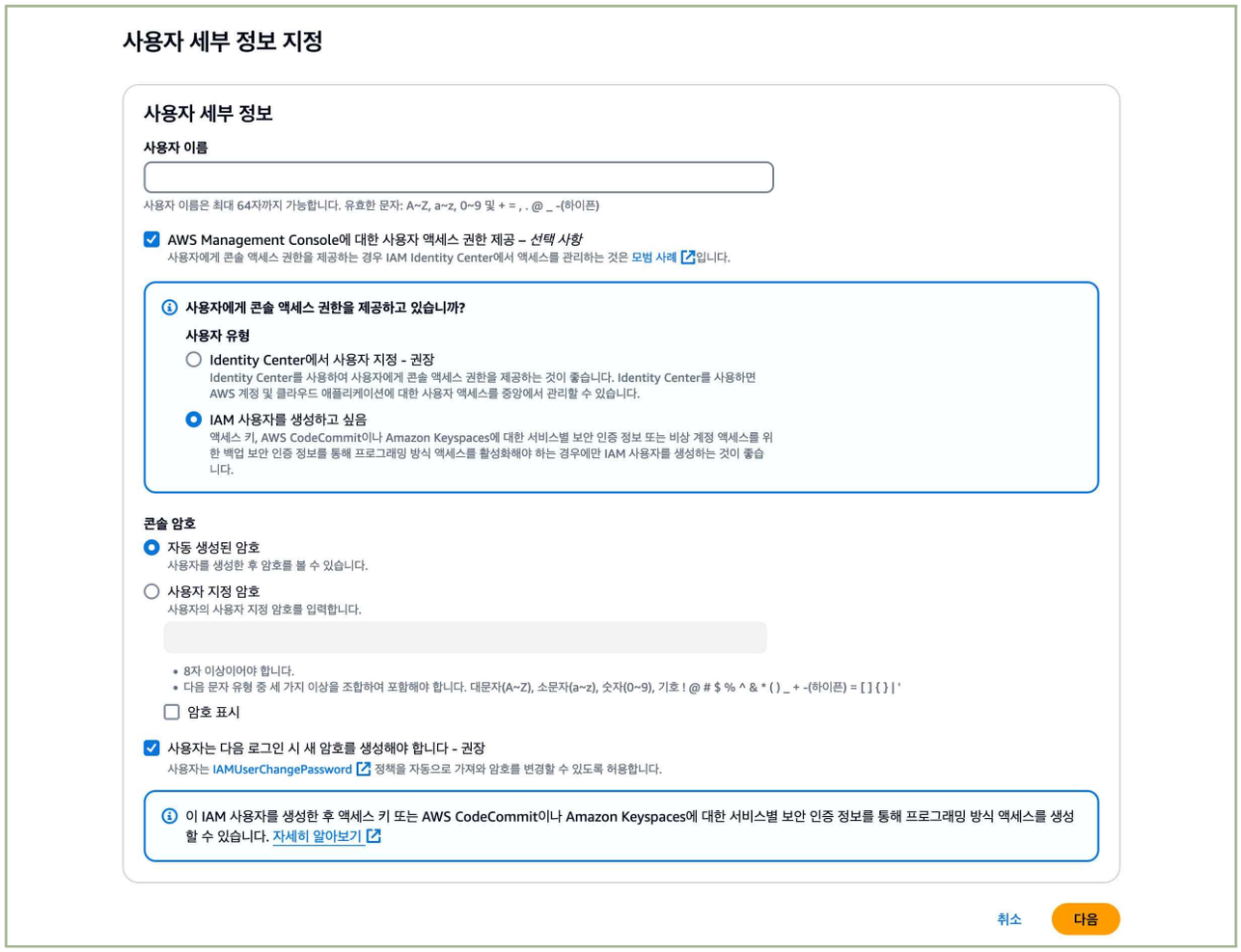
- **(AWS 관리 콘솔)** ‘콘솔 홈’ → ‘IAM’ → ‘계정 설정’ → ‘암호 정책’에서 관리자 및 사용자 계정 생성 시 이용자에게 제공하는 암호 정책 확인이 가능합니다.

| 그림 3-6-1 | 계정 설정에서 암호 정책 확인

- 비밀번호 정책의 변경이 필요한 경우 암호 정책을 ‘사용자 지정’으로 선택 후 편집할 수 있습니다.

| 그림 3-6-2 | 사용자 암호 정책 편집

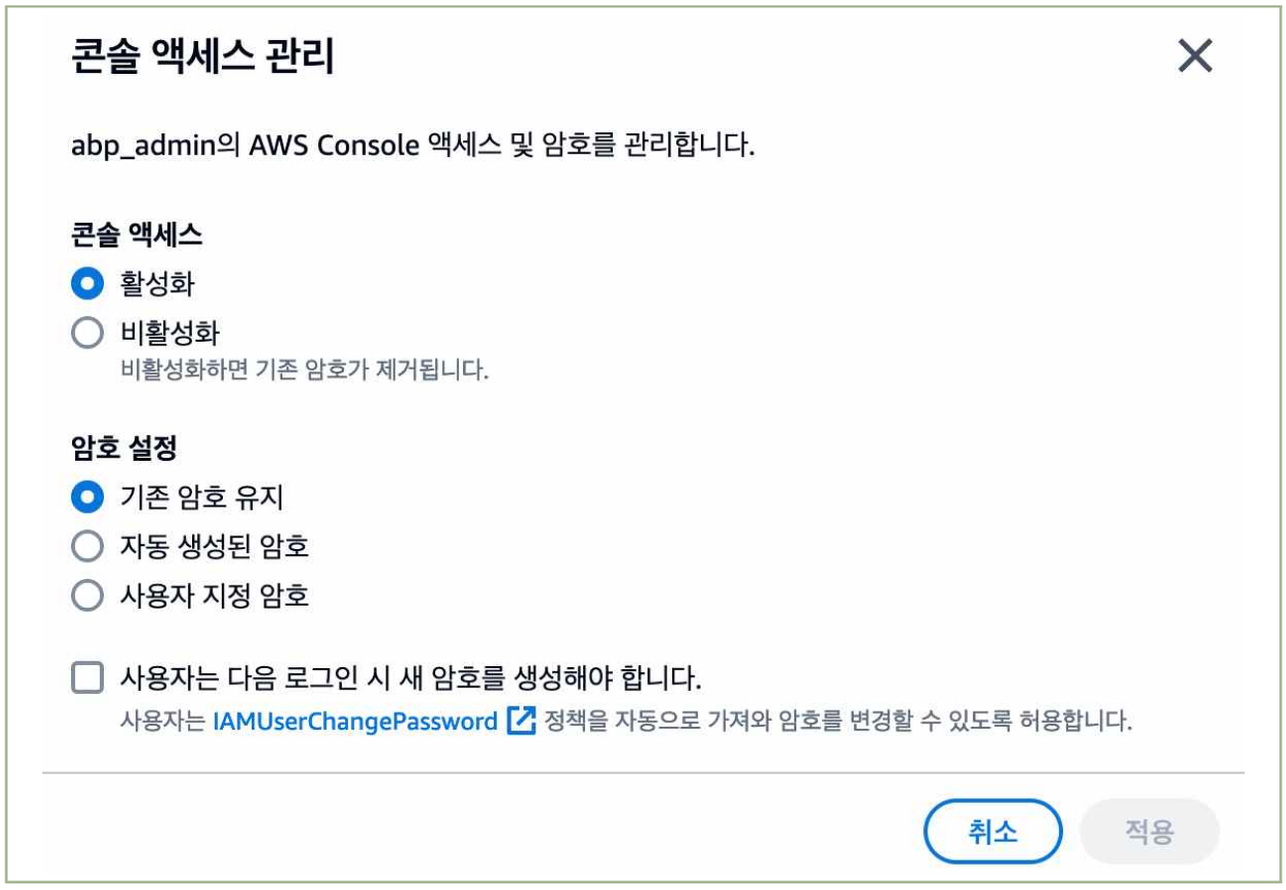
- 비밀번호의 최소 길이 및 암호 강도, 기타 요구 사항에 대한 설정을 할 수 있습니다.
- 그리고 관리자 및 사용자 계정 생성 시 이용자에게 접속 권한, 비밀번호 생성 및 로그인 시 새로운 암호 설정을 할 수 있습니다.



| 그림 3-6-3 | 사용자 계정 생성 시 AWS 관리 콘솔 접속 권한 설정

- 사용자 세부 정보 지정 시 'AWS Management Console에 대한 사용자 엑세스 권한 제공 - 선택사항' 을 체크하여 할 수 있습니다.
- 콘솔 암호를 '사용자 지정 암호' 선택 시 이용자에게 지정된 암호 제공이 가능합니다.
- '사용자는 다음 로그인 시 새 암호를 생성해야 합니다' 옵션을 체크하면 사용자는 이후 첫 로그인 시 새로운 암호를 생성하도록 강제화 할 수 있습니다.

| 그림 3-6-4 | 사용자의 보안 자격 증명



| 그림 3-6-5 | 사용자의 콘솔 액세스 관리

- **(AWS CLI)** API 환경을 통한 계정 비밀번호 규칙 확인' -〉 'get-account-password-policy' 변수를 통해 계정에서 제공하는 비밀번호 규칙을 확인할 수 있습니다.

```
aws iam get-account-password-ploicy
```

Output:

```
{
    "PasswordPolicy": {
        "AllowUsersToChangePassword": false,
        "RequireLowercaseCharacters": false,
        "RequireUppercaseCharacters": false,
        "MinimumPasswordLength": 8,
        "RequireNumbers": true,
        "RequireSymbols": true
    }
}
```

| 그림 3-6-6 | CLI 명령 - 계정에서 제공되는 암호 정책 확인

```
aws iam update-account-password-policy
        --minimum-password-length 8
        --require-numbers
```

| 그림 3-6-7 | CLI 명령 - 계정에서 제공하는 암호 정책 변경

4 참고 사항

- 비밀번호 규칙 설정이 'IAM 기본값' 일 경우 AWS CLI로 get-account-password-policy 명령어를 실행하면 에러가 발생하며, 비밀번호 규칙 설정이 '사용자 지정' 인 경우 정상적으로 결과를 회신하게 됩니다.

1 기준

식별번호	기준	내용
3.7.	**공개용 웹 서버 접근 계정 제한**	**클라우드를 통해 공개용 웹 서버를 운영하는 경우 접근 계정을 적절하게 제한하여야 한다.**

2 설명

- 클라우드 환경을 통해 공개용 웹 서버를 운영하는 경우 접근 계정을 적절하게 제한하여야 한다.
  - 예시
    1) 계정 관리 기능을 통해 공개용 웹 서버만 접근 가능한 계정을 개인별 부여하여 관리
    2) 공개용 웹 서버에 접근 가능한 계정으로 로그인 시 추가인증 수단 적용 등

3 우수 사례

1) 계정 관리 기능으로 공개용 웹 서버만 접근 가능한 계정 관리

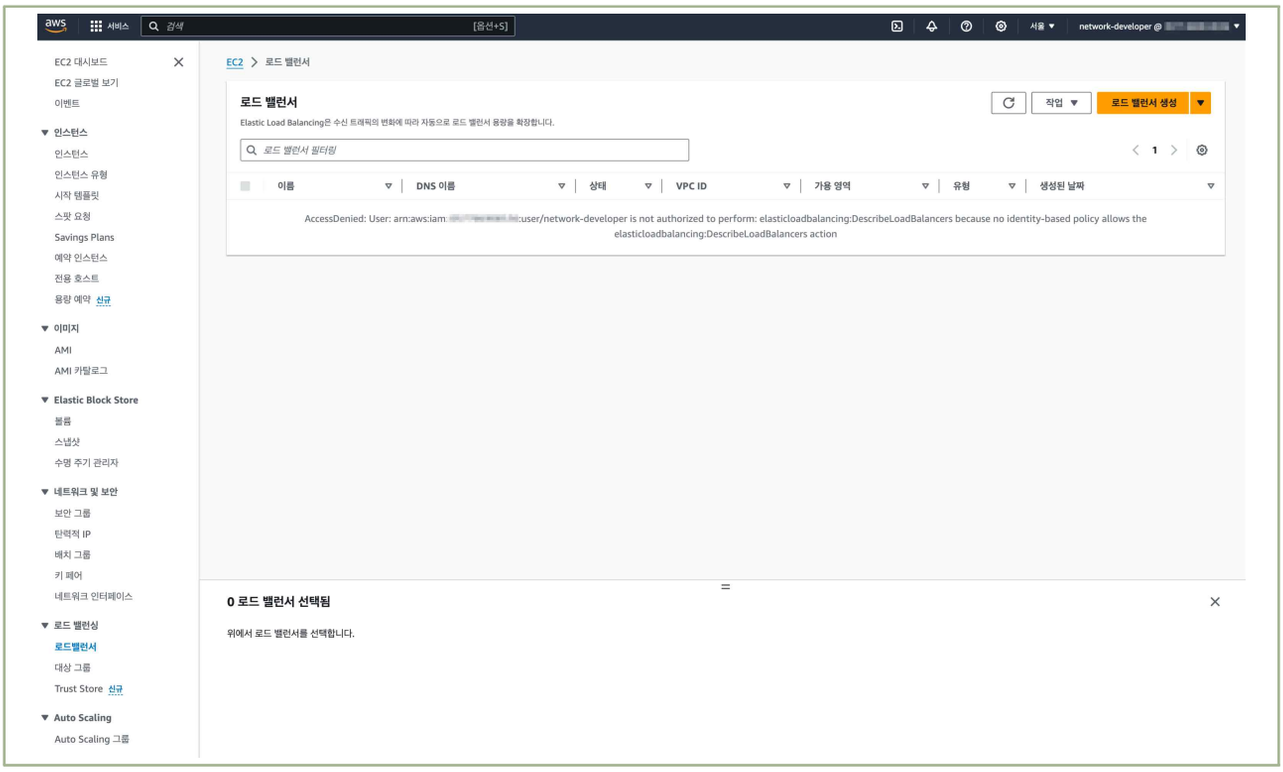
(1) ELB 권한 통제

- AWS에서 공개용 웹 서버는 Elastic Load Balancing(ELB) 혹은 Amazon EC2에 웹 서버를 직접 설치하여 관리 가능하며 각 서비스에 접근 가능한 권한이 필요 합니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → 'IAM' → '사용자' → '권한 추가' 에서 AWS 관리형 정책을 연결하여 권한 부여할 수 있습니다.

| 그림 3-7-1 | ELB 접근 권한 부여

- ELB에 대한 전체 접근 권한이 필요한 경우에는AWS 관리형 정책인 ElasticLoadBalancingFullAccess 정책을 IAM유저 권한에 연결할 수 있습니다.

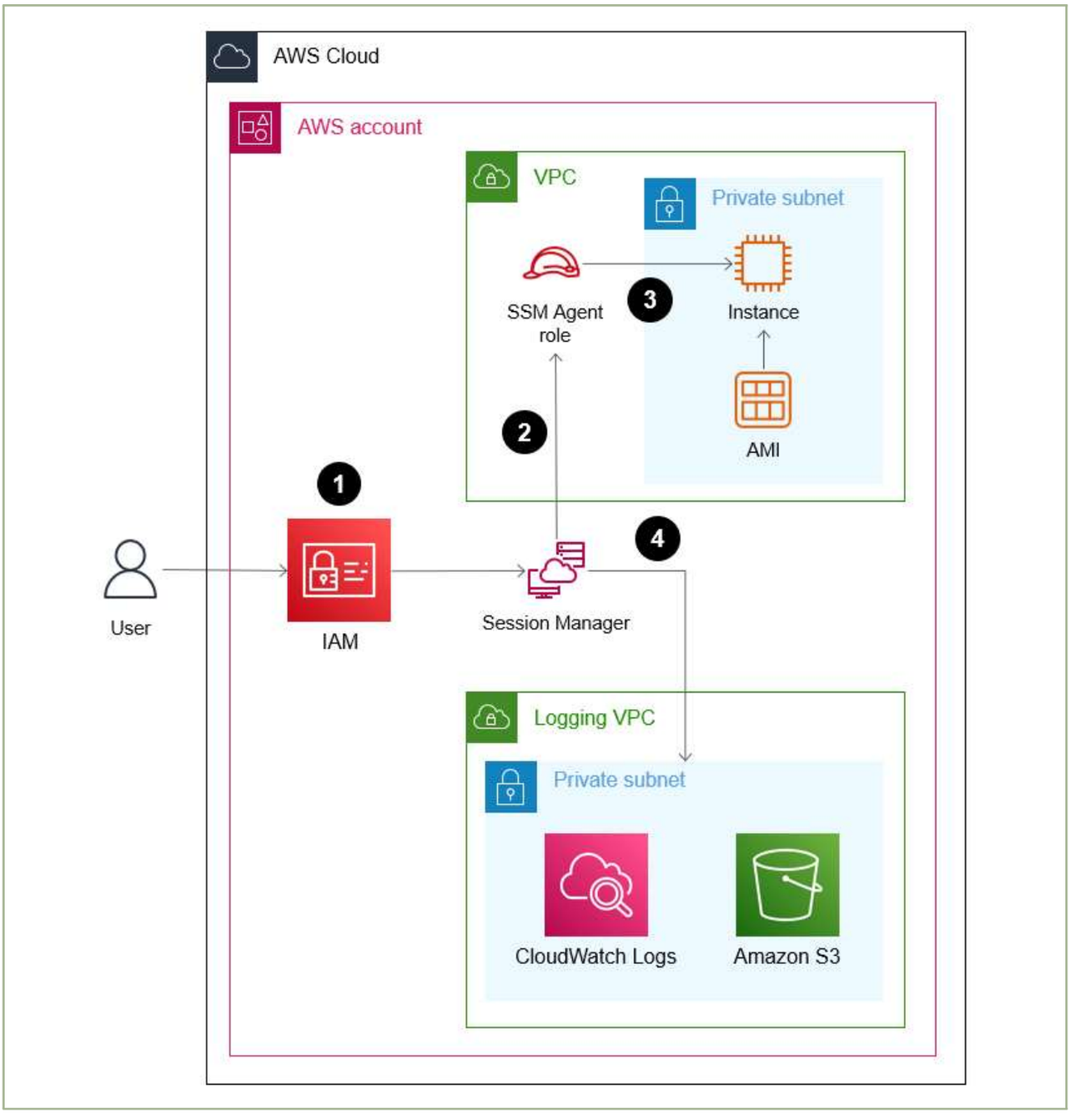


| 그림 3-7-2 | ELB 접근 권한이 없는 경우

- 위 화면은 권한이 없기 때문에 API 호출 시 AccessDenied 에러가 발생합니다. 이렇게 필요한 계정에만 권한을 부여하여 운영할 수 있습니다.

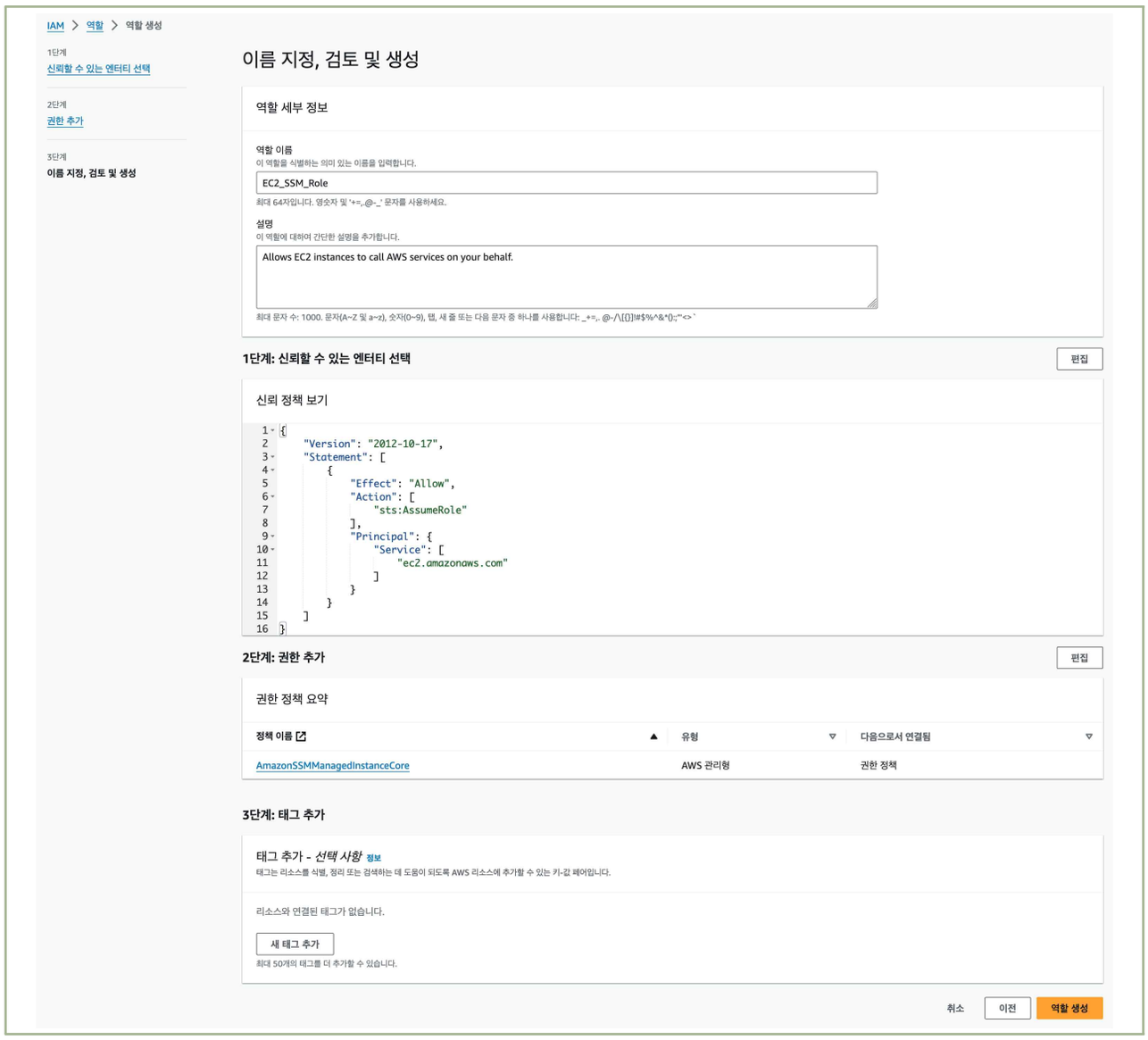
(2) 세션 관리자를 활용한 통제

- 공개용 웹 서버 EC2 인스턴스에 연결하는 여러가지 방법 중 세션 관리자를 사용하여 연결하는 방법이 있습니다.
- 사용자는 IAM을 통해 자격 증명 및 보안 인증 정보를 인증하고 세션 관리자를 통해 SSH 세션을 시작할 수 있습니다. 이 때 EC2인스턴스에 설치된 AWS System Manager SSM 에이전트는 세션 관리자와 연결이 됩니다.



| 그림 3-7-3 | 세션 관리자를 이용한 EC2 연결 아키텍처

- **(AWS 관리 콘솔)** '콘솔 홈' → 'IAM' → '역할' → '역할 생성'에서 AWS 서비스는 EC2를 선택하고 권한 정책은AmazonSSMManagedInstanceCore를 선택하여 역할을 생성합니다.



| 그림 3-7-4 | 세션 관리자 연결을 위한 IAM 역할 생성

- 그리고 '홈' → 'EC2' → '인스턴스' → '작업' →'보안'에서 IAM 역할 수정을 클릭하고 IAM 역할에 세션 관리자 접속을 위해 만든 'EC2_SSM_Role'을 설정하여 통제를 할 수 있습니다.

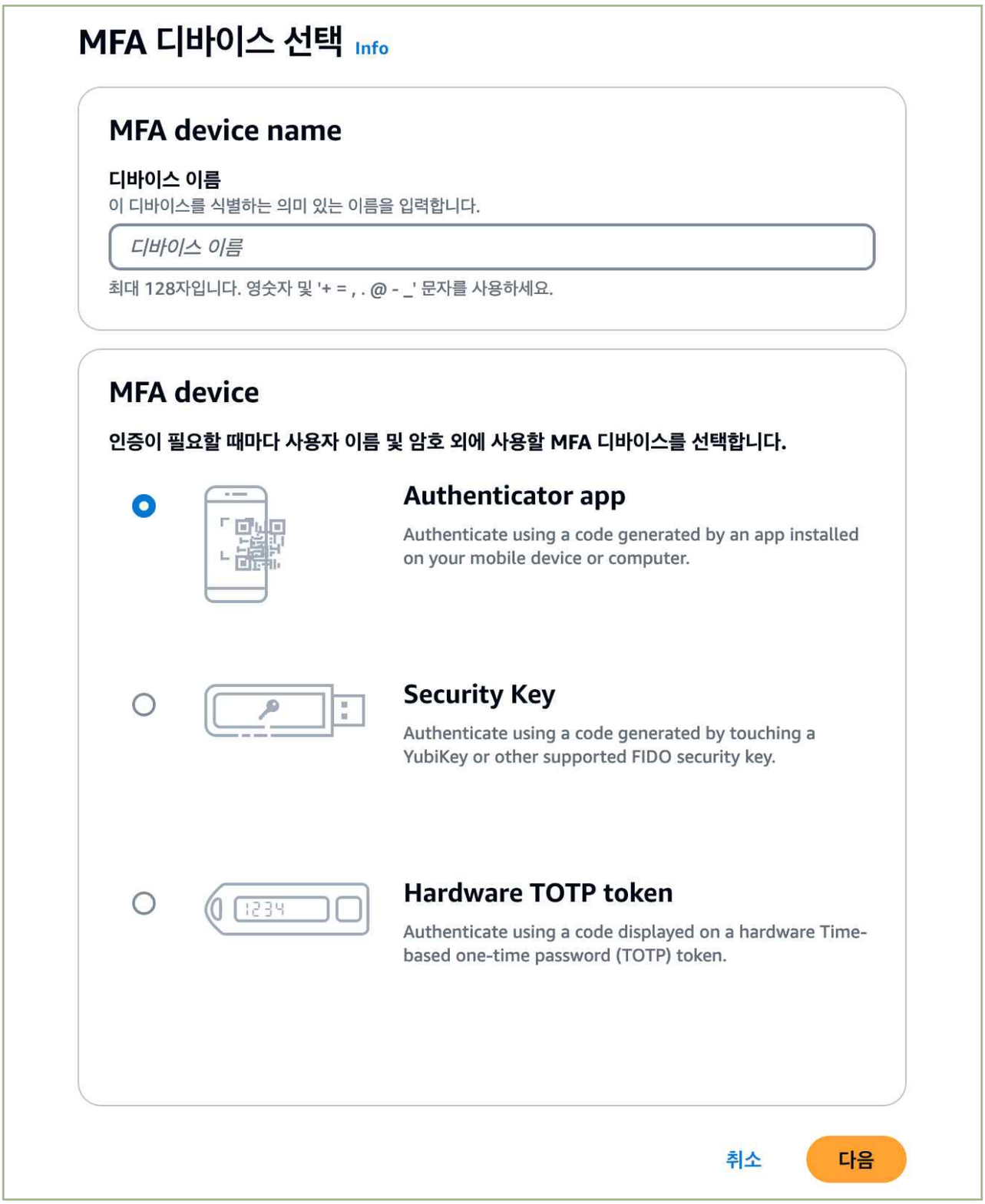
| 그림 3-7-5 | 세션 관리자 연결을 위한 EC2 인스턴스에서 IAM 역할 설정

| 그림 3-7-6 | 세션 관리자를 통해 EC2 인스턴스 연결

2) 공개용 웹 서버에 접근 가능한 계정 로그인 시 추가인증 적용

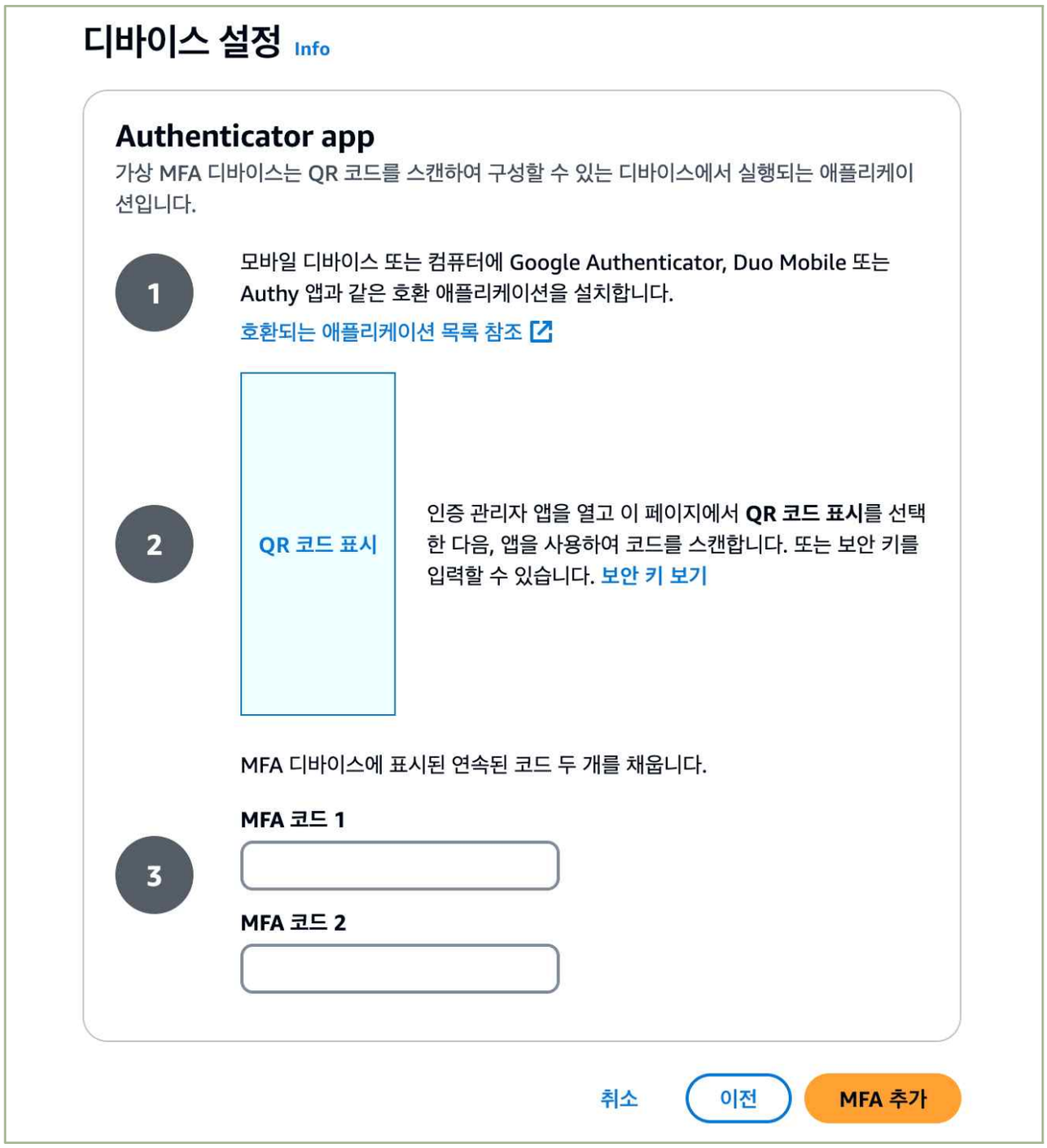
- **(AWS 관리 콘솔)** '콘솔 홈' → 'IAM' → '사용자' → '사용자 이름'의 멀티 팩터 인증(MFA)에서 MFA 디바이스 할당이 가능합니다.

| 그림 3-7-7 | MFA 할당



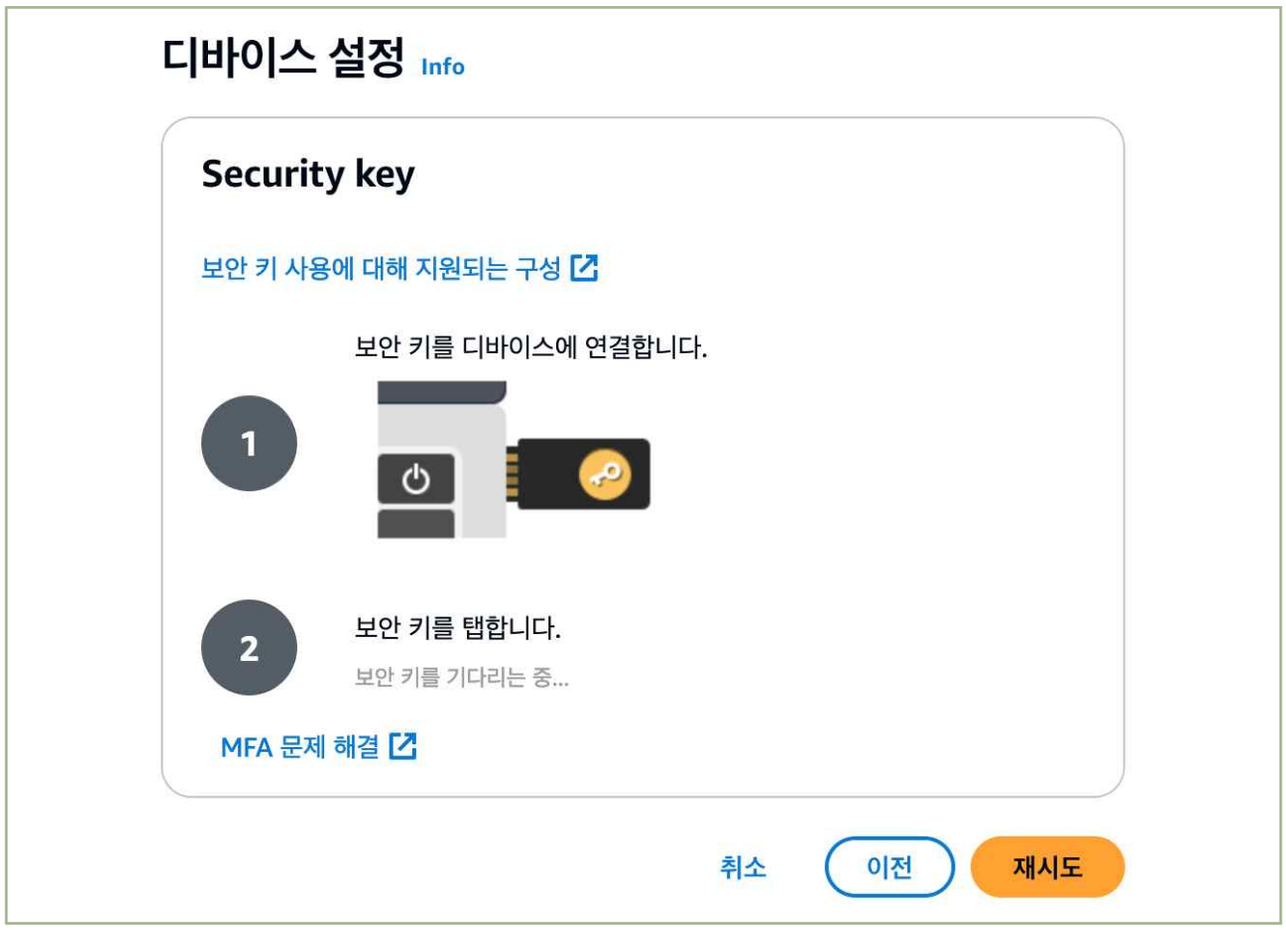
| 그림 3-7-8 | MFA 디바이스 설정 화면

- MFA 디바이스 이름을 입력하고 MFA 디바이스 타입을 선택합니다. 각 디바이스 타입에 따라서 입력방식은 상이합니다.
  - Authenticator app인 경우 연속된MFA 코드 2개를 입력하게 됩니다.

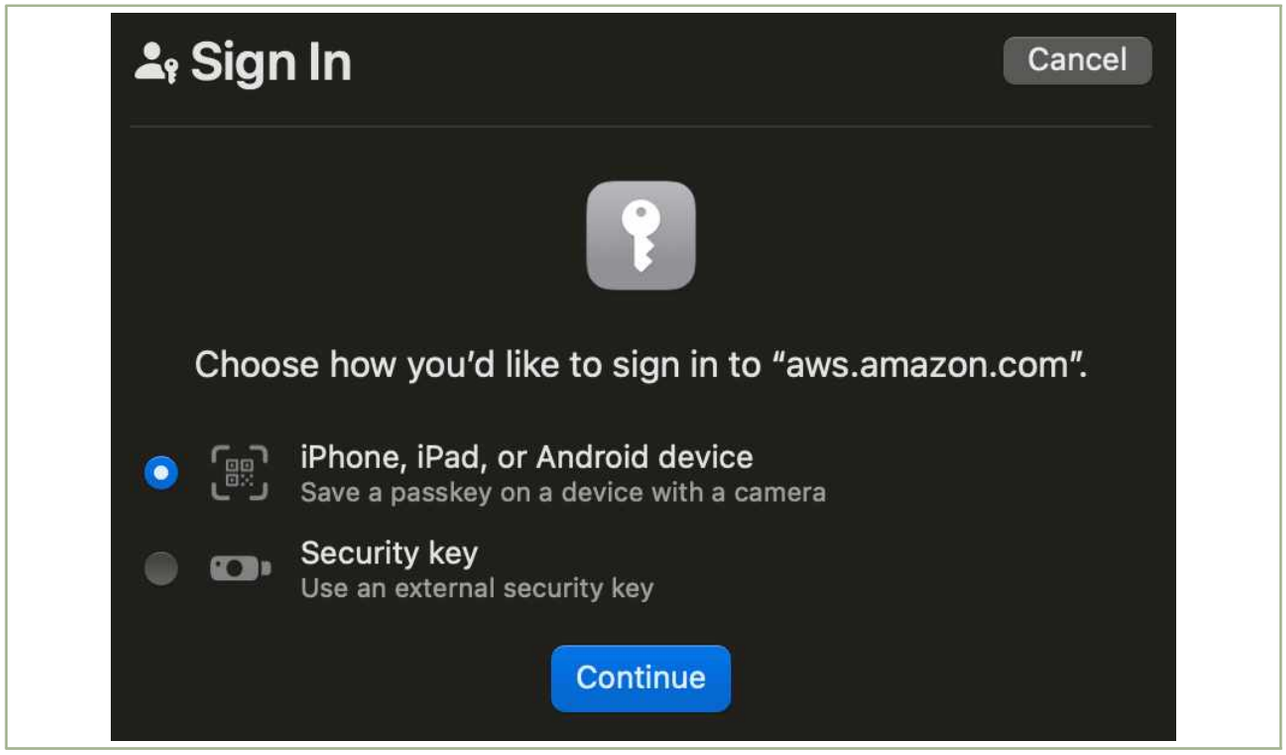


| 그림 3-7-9 | MFA 디바이스 설정 - 코드 입력 화면

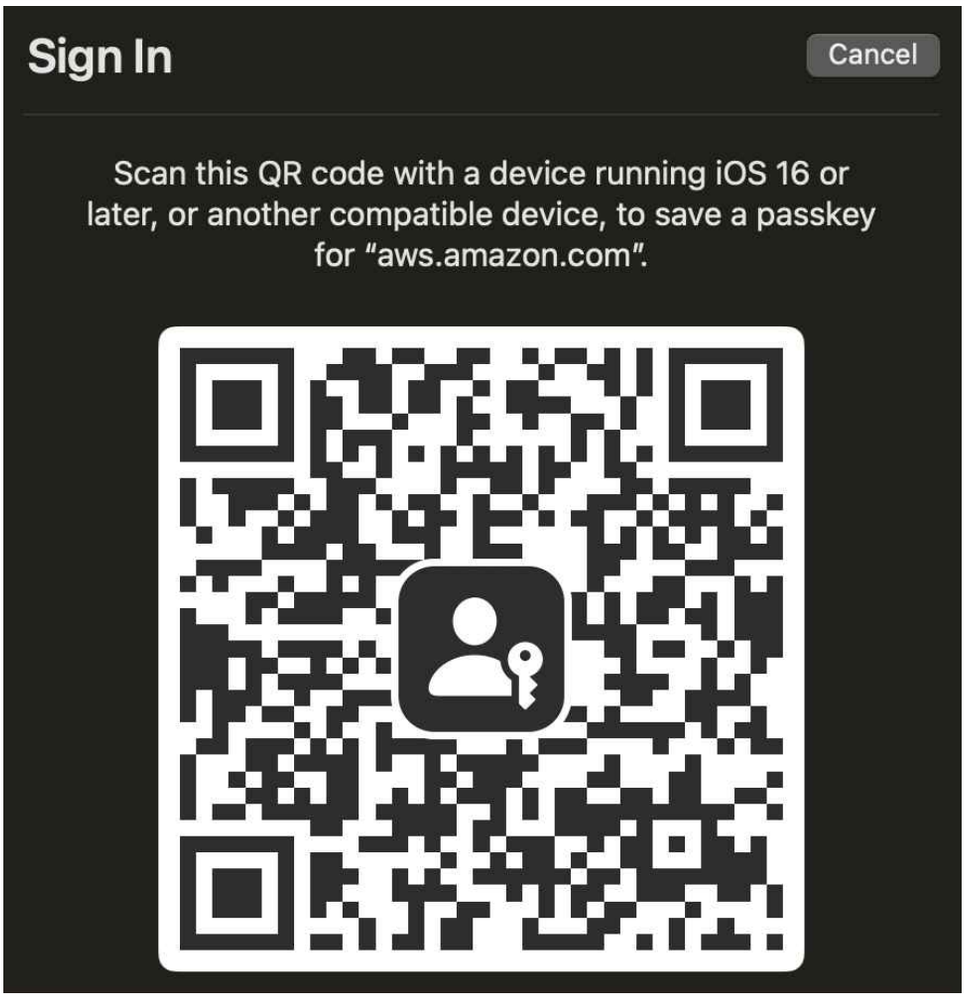
■ Security Key인 경우 iPhone, iPad, or Android device 혹은 Security key 선택 합니다.



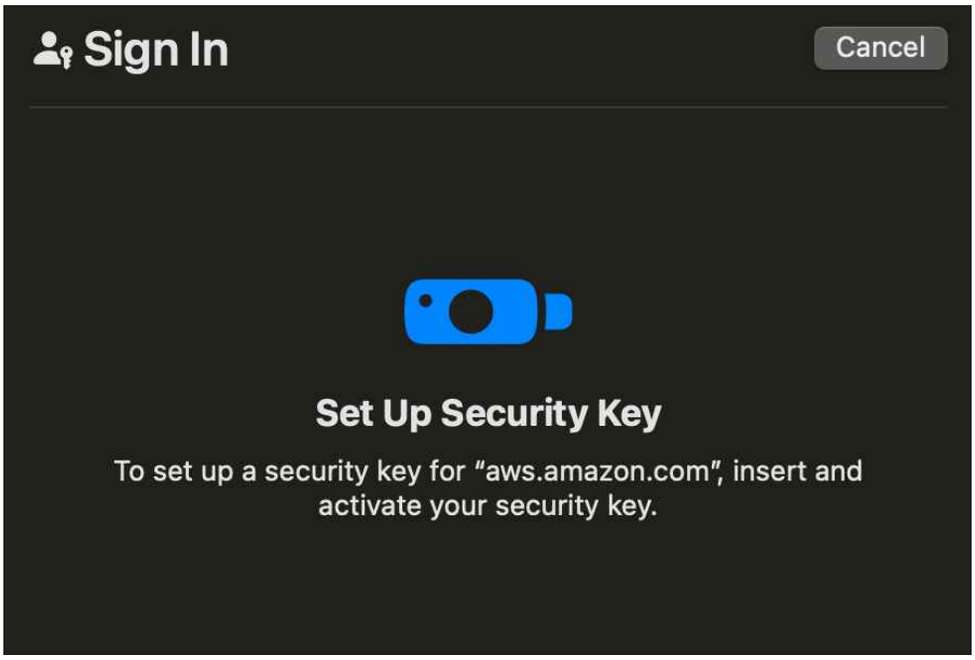
| 그림 3-7-10 | MFA 디바이스 설정 - Security Key 설정 화면



| 그림 3-7-11 | MFA 디바이스 설정 - Sign In 화면



| 그림 3-7-12 | MFA 디바이스 설정 - Sign In : iPhone, iPad, or Android device 화면



| 그림 3-7-13 | MFA 디바이스 설정 - Sign In : Secirity Key 화면

■ Hardware OTP token 경우 디바이스 일련 번호 및 MFA 6자리 입력 합니다.

| 그림 3-7-14 | MFA 디바이스 설정 – Hardware MFA device 화면

○ **(AWS CLI)** API 환경을 통한 MFA 디바이스 활성화' → 'enable-mfa-device' 명령어로 IAM 유저의 MFA 디바이스를 활성화 할 수 있습니다.

```
aws iam enable-mfa-device \
      --user-name Bob \
      --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \
      --authentication-code1 123456 \
      --authentication-code2 789012
```

| 그림 3-7-15 | CLI 명령 – MFA 디바이스 활성화

- 그리고 'API 환경을 통한 가상 MFA 디바이스 생성' → 'create-virtual-mfa-device' 명령어로 가상 MFA 디바이스를 생성할 수 있습니다.

```
aws iam create-virtual-mfa-device \
      --virtual-mfa-device-name BobsMFADevice \
      --outfile C:/QRCode.png \
      --bootstrap-method QRCodePNG
```

Output:

```
{
    "VirtualMFADevice": {
        "SerialNumber": "arn:aws:iam::210987654321:mfa/BobsMFADevice"
    }
}
```

| 그림 3-7-16 | CLI 명령 – 가상MFA 디바이스 생성

4 참고 사항

- 멀티 팩터 인증(MFA)의 추가 가이드는 아래 AWS 공식 문서를 참조
  - https://docs.aws.amazon.com/ko_kr/IAM/latest/UserGuide/id_credentials_mfa.html

4. 암호키 관리

4.1. 암호화 적용 가능 여부 확인
4.2. 암호키 관리 방안 수립
4.3. 암호키 서비스 관리자 권한 통제
4.4. 암호키 호출 권한 관리
4.5. 안전한 암호화 알고리즘 적용

4 암호키 관리

1 기준

식별번호	기준	내용
4.1.	암호화 적용 가능 여부 확인	관련 법령(전자금융거래법, 신용정보법 등)에 따른 암호화 대상이 저장 및 처리되는 가상자원(서버, 스토리지 등)에 대한 암호화 기능 적용 여부를 확인하여야 한다.

2 설명

- 관련 법령(전자금융거래법, 개인정보보호법, 신용정보법 등)에 따라 암호화가 필요한 대상이 저장 및 처리되는 가상자원에 대해서는 암호화 적용을 고려하여야 한다.

  - 예시

    1) 클라우드의 키 관리 서비스를 통해 CSP 사업자의 관리형 Key로 암호화
    2) 클라우드의 키 관리 서비스를 통해 이용자 관리형 Key로 암호화
    3) 이용자가 직접 관리하는 Key로 암호화 등

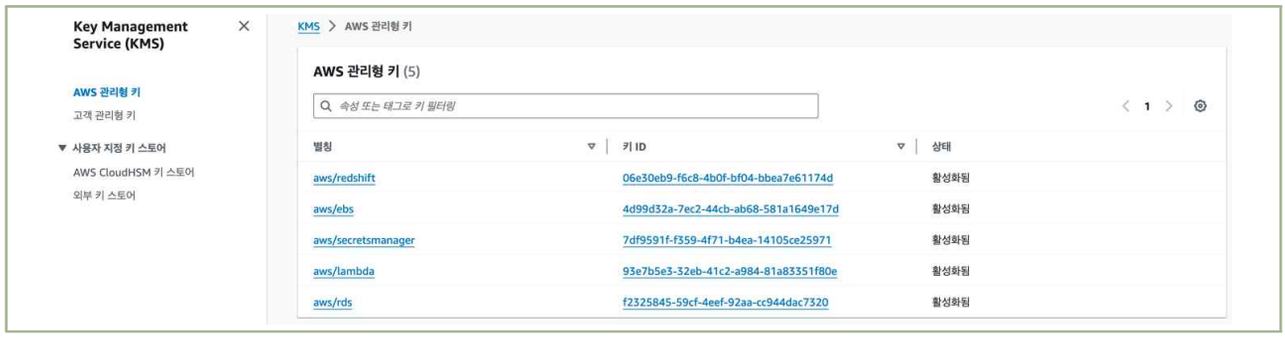
3 우수 사례

1) 클라우드의 키 관리 서비스를 통해 CSP 사업자의 관리형 Key로 암호화

(1) AWS 관리형 키 목록 확인

- 본 참고서는 AWS의 데이터 암호화 대상 서비스들 중 EBS를 예시로 다룹니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → 왼쪽 상단의 검색에 'KMS' 입력 → 'Key Management Service' 클릭 → 'AWS 관리형 키' 클릭 → 'aws/ebs'키의 상태가 '활성화됨' 인지 확인합니다.



| 그림 4-1-1 | AWS 관리형 키 목록

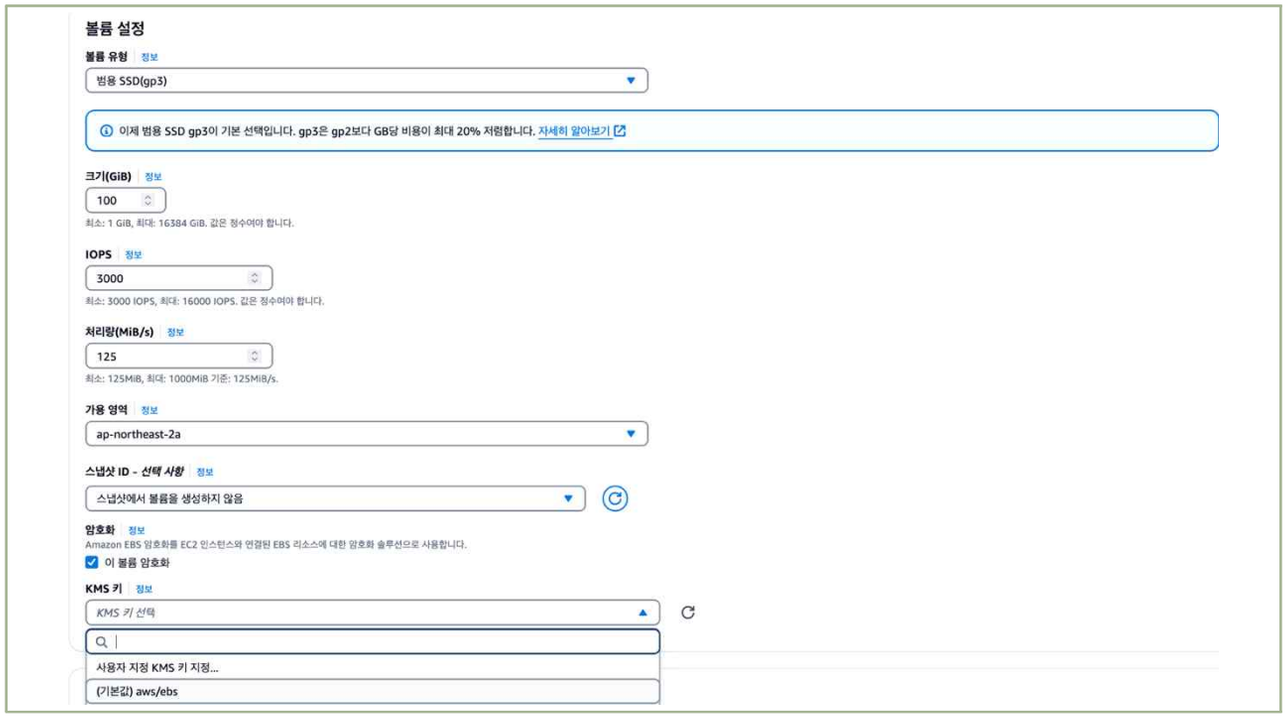
- **(AWS CLI)** API 환경을 통한 리전 내의 모든 KMS 키의 별칭이름 및 ARN 가져오기 → 'AWS 관리형 키'의 별칭인 'alias/aws/ebs' 항목을 확인합니다.

```
$ aws kms list-aliases

{
    "Aliases": [
        {
            "AliasName": "alias/access-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1516435200.399,
            "LastUpdatedDate": 1516435200.399
        },
        {
            "AliasName": "alias/financeKey",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/financeKey",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1604958290.014,
            "LastUpdatedDate": 1604958290.014
        },
        {
            "AliasName": "alias/ECC-P521-Sign",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ECC-P521-Sign",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1693622000.704,
            "LastUpdatedDate": 1693622000.704
        },
        {
            "AliasName": "alias/ImportedKey",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ImportedKey",
            "TargetKeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
            "CreationDate": 1493622000.704,
            "LastUpdatedDate": 1521097200.235
        },
        {
            "AliasName": "alias/aws/dynamodb",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/dynamodb",
            "TargetKeyId": "0987ab65-43cd-21ef-09ab-87654321cdef",
            "CreationDate": 1521097200.454,
            "LastUpdatedDate": 1521097200.454
        },
        {
            "AliasName": "alias/aws/ebs",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/ebs",
            "TargetKeyId": "abcd1234-09fe-ef90-09fe-ab0987654321",
            "CreationDate": 1466518990.200,
            "LastUpdatedDate": 1466518990.200
        },
        {
            "AliasName": "alias/aws/redshift",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/redshift"
        },
    ]
}
```

| 그림 4-1-2 | CLI 명령 - AWS 관리형 키 탐색

(2) AWS 관리형 키로 EBS 볼륨 암호화

- **(AWS 관리 콘솔)** '콘솔 홈' → '왼쪽 상단의 검색에 EBS 입력' → 특징 카테고리의 '볼륨' 클릭 → 오른쪽 상단 '볼륨 생성' 클릭 → 암호화 필드의 '이 볼륨 암호화' 클릭 → KMS 키 리스트에서 3-1-1에서 확인한 '(기본값)aws/ebs' 키 선택 → 하단의 '볼륨 생성'을 클릭하여 암호화된 볼륨을 생성합니다.



| 그림 4-1-3 | 볼륨 암호화를 위한 KMS 키 선택

- **(AWS CLI)** API 환경을 통한 AWS 관리형 키로 EBS 볼륨 암호화 → 아래 CLI를 실행합니다. (실행 결과로 서울 리전의 모든 EBS 볼륨 암호화 기능을 디폴트로 활성화시킴).

```
aws ec2 enable-ebs-encryption-by-default
```

Output:

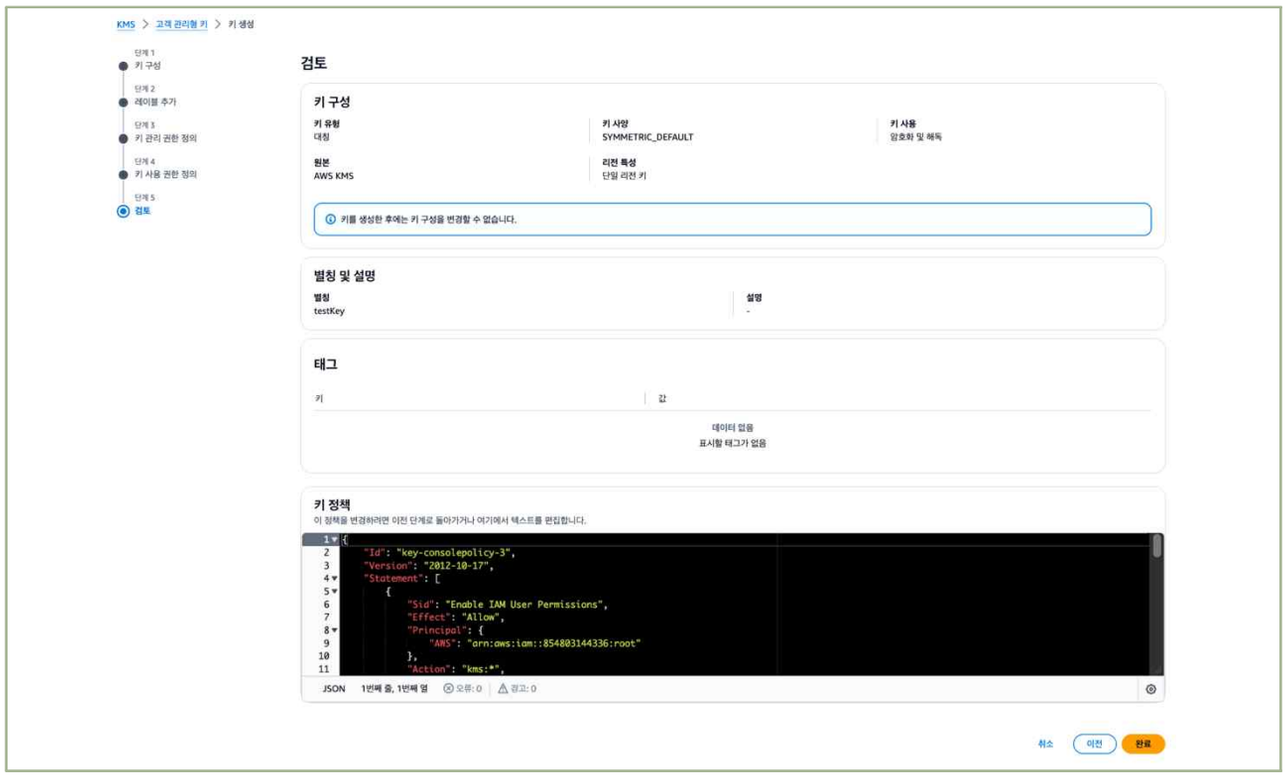
```
{
    "EbsEncryptionByDefault": true
}
```

| 그림 4-1-4 | 리전 별 EBS 볼륨에 대한 Default 암호화 키 설정

2) 클라우드의 키 관리 서비스를 통해 이용자 관리형 Key로 암호화

(1) 이용자 관리형 Key 생성

- **(AWS 관리 콘솔)** ‘콘솔 홈’ → 왼쪽 상단의 검색에 ‘KMS’ 입력 → ‘Key Management Service’ 클릭 → ‘고객 관리형 키’ 클릭 → 오른쪽 상단의 ‘키 생성’ 클릭 → ‘키 구성 정보 설정’에서 ‘대칭’ / ‘암호화 및 해독’ 선택 후 ‘다음’ 클릭 → 별칭 입력 후 ‘다음’ 클릭 → ‘키 관리자’ 지정 후 ‘다음’ 클릭 → ‘키 사용자’ 지정 후 ‘다음’ 클릭 → ‘검토’ 단계에서 설정한 정보 확인 후, 하단 ‘완료’ 클릭합니다.



| 그림 4-1-5 | 이용자 관리형 Key 생성

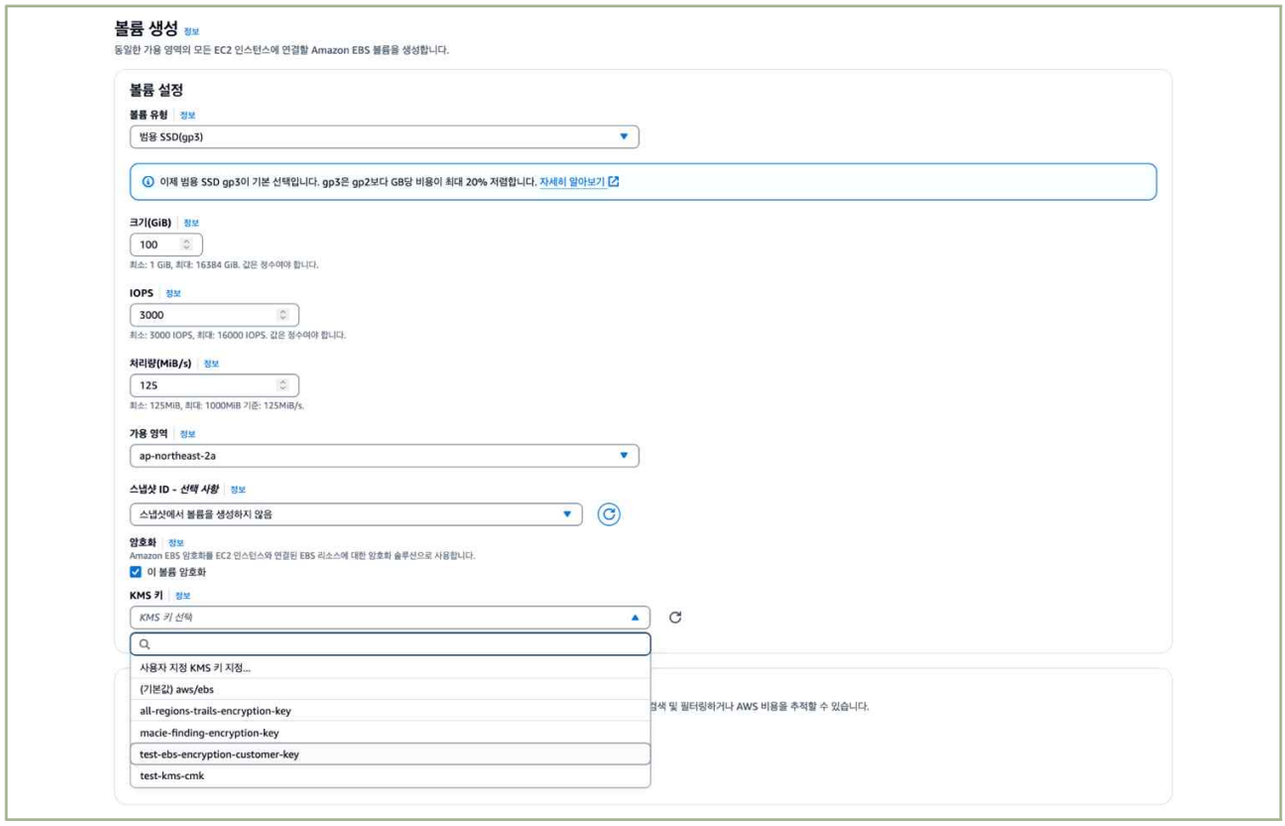
- **(AWS CLI)** API 환경을 통한 이용자 관리형 키로 EBS 볼륨 암호화 → 아래 CLI를 실행하고 생성된 암호화 키의 정보(KeyId)를 확인합니다.

```
$ aws kms create-key
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1502910355.475,
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "MultiRegion": false
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
    }
}
```

| 그림 4-1-6 | CLI 명령 - 이용자 관리형 Key 생성

(2) 이용자 관리형 Key로 EBS 볼륨 암호화

- **(AWS 관리 콘솔)** '콘솔 홈' → 왼쪽 상단의 '검색'에 'EBS' 입력 → 특징 카테고리의 '볼륨' 클릭 → 오른쪽 상단 '볼륨 생성' 클릭 → 암호화 필드의 '이 볼륨 암호화' 클릭 → KMS 키 리스트에서 (1)단계에서 생성한 이용자 관리형 키 선택 → 하단의 '볼륨 생성'을 클릭하여 암호화된 볼륨을 생성합니다.



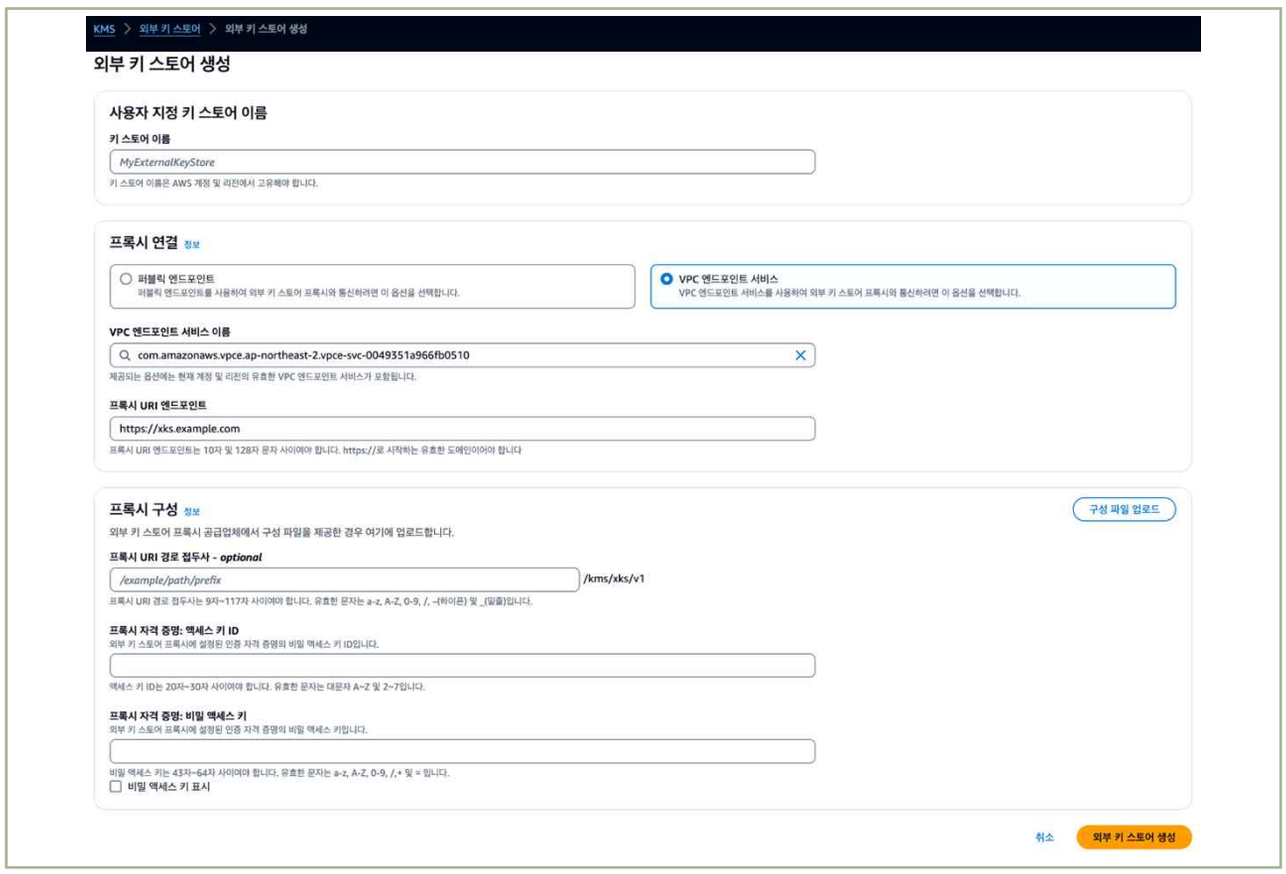
| 그림 4-1-7 | 볼륨 암호화를 위한 KMS 키 선택

- **(AWS CLI)** '1)-(2) AWS 관리형 키로 EBS 볼륨 암호화'의 AWS CLI 내용 참조 바랍니다.

3) 이용자가 직접 관리하는 Key로 암호화

(1) 외부 키 스토어 생성

- **(AWS 관리 콘솔)** '콘솔 홈' → 왼쪽 상단의 검색에 'KMS' 입력 → 'Key Management Service' 클릭 → '사용자 지정 키 스토어' 클릭 → 왼쪽 메뉴바에서 '외부 키 스토어' 클릭 → 고유한 이름으로 이용자가 직접 관리하는 외부 키 스토어 이름 설정 → 프록시 연결 항목에서 'VPC 엔드포인트 서비스' 선택(VPC 내부 통신) → VPC 엔드포인트 서비스 이름에 기 생성된 외부 키 스토어를 위한 VPC 엔드포인트 서비스 이름이 자동으로 표시됨 → 프록시 구성 항목에서 외부 키 스토어 프록시의 인증 자격증명을 위한 엑세스 키와 비밀 액세스 키를 입력하고 '외부 키 스토어 생성'을 클릭합니다.



| 그림 4-1-8 | 외부 키 스토어 생성

- **(AWS CLI)** API 환경을 통한 외부 키 스토어 생성 → 붉은색으로 표시된 각 항목을 설정된 값으로 변경하여 실행합니다.(단, '*--xks-proxy-uri-path*'값은 '*/kms/xks/v1*'을 그대로 입력) → 리턴 결과인 '*CustomKeyStoreId*'값을 기록해 둡니다.

```
aws kms create-custom-key-store \
    --custom-key-store-name ExamplePublicEndpointXKS \
    --custom-key-store-type EXTERNAL_KEY_STORE \
    --xks-proxy-connectivity PUBLIC_ENDPOINT \
    --xks-proxy-uri-endpoint "https://myproxy.xks.example.com" \
    --xks-proxy-uri-path "/example-prefix/kms/xks/v1" \
    --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE, RawSe
cretAccessKey=DXjSUawnel2fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

Output:

```
{
    "CustomKeyStoreId": cks-2234567890abcdef0
}
```

| 그림 4-1-9 | 외부 키 스토어 생성

(2) 외부 키 스토어에서 KMS 키 생성

- **(AWS 관리 콘솔)** VPC는 '콘솔 홈' → 왼쪽 상단의 검색에 'KMS' 입력 → 'Key Management Service' 클릭 → '사용자 지정 키 스토어' 클릭 → 왼쪽 메뉴바에서 '외부 키 스토어' 클릭 → '(1) 외부 키 스토어 생성'에서 생성한 외부 키 스토어를 선택 → 오른쪽 상단에서 '이 키 스토어에서 KMS 키 생성'을 선택 → 외부 키 스토어에 있는 KMS용 암호화 키 ID를 입력하고 '다음' 클릭(새로 생성되는 KMS키를 외부 키 스토어의 암호화 키와 연결시키는 과정임) → 생성할 키의 별칭을 입력하고 '다음'을 클릭 → 키 관리자 지정 후 '다음' 클릭 → 키 사용자 지정 후 '다음' 클릭 → 설정 정보 확인 후 하단 '완료'를 클릭합니다.

- **(AWS CLI)** API 환경을 통한 외부 키 스토어에서 KMS키 생성 → '*--custom-key-store-id*' 는 '(1) 외부 키 스토어 생성' 단계에서 생성시 확인한 '*CustomKeyStoreId*'을 입력하고, '*--xks-key-id*'는 외부 키 스토어에서 생성한 KMS용 암호화 키 ID를 지정합니다.

```
aws kms create-key \
    --origin EXTERNAL_KEY_STORE \
    --custom-key-store-id cks-9876543210fedcba9 \
    --xks-key-id bb8562717f809024
```

Output:


```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-123
4567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2022-12-02T07:48:55-07:00",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "CustomKeyStoreId": "cks-9876543210fedcba9",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "EXTERNAL_KEY_STORE",
        "XksKeyConfiguration": {
            "Id": "bb8562717f809024"
        }
    }
}
```


| 그림 4-1-10 | 외부 키 스토어에서 KMS 키 생성

(3) 외부 키 스토어에서 생성한 KMS 키로 EBS 볼륨 암호화

- '(2) 이용자 관리형 키로 EBS 볼륨 암호화' 의 내용을 참조 바랍니다.

4 참고 사항

1) 외부 키 스토어 구성을 지원하는 공급업체와 스펙은 아래에서 확인 가능합니다.

- AWS KMS FAQ(https://aws.amazon.com/ko/kms/faqs/) 의 '외부 키 스토어' 항목 참조

2) 외부 키 스토어가 지원해야 되는 KMS 키 스펙은 다음에서 확인 가능합니다.

- https://aws.amazon.com/ko/kms/faqs/

3) EBS암호화 외 본 가이드에서 다루지 않는 다른 주요 암호화 기능에 대한 가이드는 아래 목록을 참조합니다.

- Amazon RDS
  https://docs.aws.amazon.com/ko_kr/AmazonRDS/latest/UserGuide/Overview.Encryption.html
- Amazon S3
  https://docs.aws.amazon.com/ko_kr/AmazonS3/latest/userguide/serv-side-encryption.html
- Amazon Redshift
  https://docs.aws.amazon.com/ko_kr/redshift/latest/mgmt/working-with-db-encryption.html
- Amazon DynamoDB
  https://docs.aws.amazon.com/ko_kr/amazondynamodb/latest/developerguide/EncryptionAtRest.html
- Amazon EMR
  https://docs.aws.amazon.com/ko_kr/emr/latest/ManagementGuide/emr-data-encryption-options.html

1 기준

식별번호	기준	내용
4.2.	암호키 관리 방안 수립	암호화 기능 이용 시 암호키 관리방안을 수립하여야 한다.

2 설명

○ 암호화 기능 이용 시 암호키 관리 방안을 수립하여야 한다.

- 예시

1) KMS(Key Management Service)를 통한 암호화 키 방안 수립(생성, 변경, 폐기 등)
2) 클라우드 서비스 제공자가 직접 제공하는 암호화 키 이용 시 적절한 관리방안 수립
3) 키 사용기간 수립 및 암호키 유출 등에 대응할 수 있도록 키 삭제 및 재적용 관련 기능 수립
4) 생성된 암호화 키를 안전하게 보관할 수 있는 방안 수립 등

3 우수 사례

1) KMS를 통한 암호화 키 관리 방안 수립

- AWS KMS를 이용한 암호화 키 생성 방안에 대한 내용은 [섹션 4.1]의 '암호키 생성' 참조 바랍니다.
- AWS KMS를 이용한 암호화 키 변경(교체) 방안에 대한 내용은 아래 '3)-(1) 자동 키 교체 활성화'를 참조 바랍니다.
- AWS KMS를 이용한 암호화 키 삭제 방안에 대한 내용은 아래 '3)-(2) 키 삭제'를 참조 바랍니다.

2) 클라우드 서비스 제공자가 직접 제공하는 암호화 키 이용 시 적절한 관리방안 수립

- AWS가 직접 제공하는 AWS 관리형 키는 AWS KMS를 사용하여 안전하게 보관되며, AWS KMS와 통합된 AWS서비스의 데이터 암호화를 위해 고객 대신 AWS가 생성 및 관리를 수행합니다.
- 암호화 키 또는 키 정책을 만들거나 유지할 필요가 없으며 AWS 관리형 키에 대한 비용도 발생하지 않습니다.
- 이용자는 AWS 계정에서 AWS 관리형 키와 키 정책을 확인하고 AWS CloudTrail 로그에서

사용을 감사할 권한이 있지만, AWS 관리형 키에 대해서는 속성을 변경하거나, 교체 및 키 정책을 변경과 삭제와 같은 행위는 할 수 없습니다.

- 이용자의 개인적인 다른 용도의 암호화 작업에서 직접 AWS 관리형 키를 사용할 수 없습니다. 따라서 일반적으로는, 리소스를 보호하는 암호화 키를 제어할 필요가 없을 경우에만, AWS 관리형 키를 선택하는 것이 좋습니다.

3) 키 사용기간 수립 및 키 삭제 및 재적용

(1) 자동 암호화 키 교체 활성화

- **(AWS 관리 콘솔)** '콘솔 홈' → 왼쪽 상단의 검색에 'KMS' 입력'→ 'Key Management Service' 클릭 → '고객 관리형 키' 클릭 → 목록에서 관리할 키 별칭을 클릭 → '키 교체' 탭 클릭 → '매년 이 KMS키를 자동으로 교체합니다' 를 선택 하고 '저장'을 클릭합니다.

| 그림 4-2-1 | 암호화 키 자동 교체 활성화

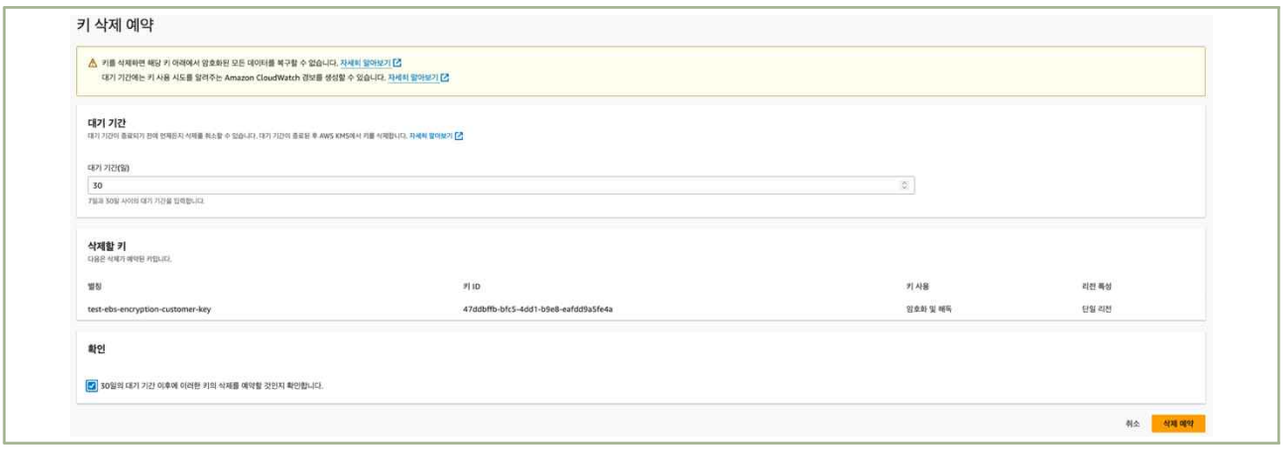
- **(AWS CLI)** API 환경을 통한 자동 키 교체 활성화 → 다음 CLI 명령을 실행하여 자동 키 교체 기능을 활성화하고 결과를 확인합니다.

```
$ aws kms enable-key-rotation --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

$ aws kms get-key-rotation-status --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyRotationEnabled": true
}
```

| 그림 4-2-2 | 암호화 키 자동 교체 활성화

(2) 암호화 키 삭제

- **(AWS 관리 콘솔)** '콘솔 홈' → 왼쪽 상단의 검색에 'KMS' 입력 → 'Key Management Service' 클릭 → '고객 관리형 키' 클릭 → 목록에서 관리할 키 선택 → 오른쪽 상단의 '키 작업' 클릭 → 드롭다운 리스트의 '키 삭제 예약' 선택 → 대기기간을 7 ~ 30일 사이로 설정 → '30일간의 대기 기간 이후에 이러한 키의 삭제를 예약할 것인지 확인합니다.' 선택 후 '삭제 예약'을 클릭합니다.



| 그림 4-2-3 | 암호화 키 삭제 예약

- **(AWS CLI)** API 환경을 통한 KMS키 삭제 예약 → 다음 CLI 명령에서 삭제할 키 ID를 입력하고 키 삭제 예약 대기 기간을 주고 실행합니다.

```
$ aws kms schedule-key-deletion --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --pending-window-in-days 10
```

| 그림 4-2-4 | CLI 명령 - 암호화 키 삭제 예약

(3) 암호화 키 활성화 / 비활성화

- **(AWS 관리 콘솔)** '콘솔 홈' → 왼쪽 상단의 검색에 'KMS' 입력 → 'Key Management Service' 클릭 → '고객 관리형 키' 클릭 → 목록에서 관리할 키 선택 → 오른쪽 상단의 '키 작업' 클릭 → 드롭다운 리스트의 '비활성화' 선택 → '이 키를 비활성화할 것인지 확인합니다.'를 선택하고 '키 비활성화'를 클릭합니다.

| 그림 4-2-5 | 암호화 키 비활성화

- 동일한 과정으로 다시 암호화 키를 활성화 할 수 있습니다.

- **(AWS CLI)** API 환경을 통한 KMS키 활성화/비활성화 → 다음 CLI 명령에서 삭제할 키 ID를 입력하고 키 활성화/비활성화를 실행합니다.

```
$ aws kms disable-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

| 그림 4-2-6 | 암호화 키 비활성화

```
$ aws kms enable-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

| 그림 4-2-7 | 암호화 키 활성화

4) 생성된 암호화 키를 안전하게 보관할 수 있는 방안

- 이용자의 암호화 정책에 따라 암호화 키를 보관할 수 있으며, AWS는 암호화 키를 안전하게 보관하기 위해 AWS KMS(Key Management Service)와 AWS CloudHSM 서비스를 제공합니다.
- AWS KMS와 AWS CloudHSM에 대한 보안 내용과 가이드는 참고 사항에서 확인 바랍니다.

4 참고 사항

- AWS KMS 보안 내용 및 가이드
  - https://docs.aws.amazon.com/ko_kr/kms/latest/developerguide/data-protection.html

- AWS CloudHSM 보안 내용 및 가이드
  - https://docs.aws.amazon.com/ko_kr/cloudhsm/latest/userguide/security.html

1 기준

식별번호	기준	내용
4.3.	**암호키 서비스 관리자 권한 통제**	**클라우드 암호키 서비스 이용 시 관리자 권한은 최소인원에게 부여하고 모니터링하여야 한다.**

2 설명

- 클라우드 환경 내 암호키 관리 서비스(ex. KMS) 이용 시 암호키 서비스 관리자 권한을 적절하게 통제하여야 한다.

  - 예시

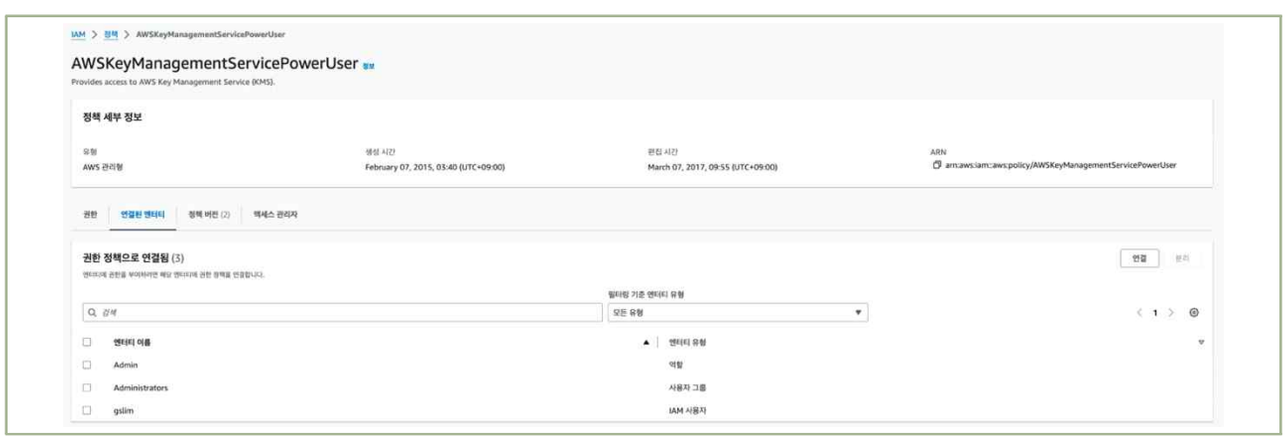
    1) 암호키 관리 서비스 관리자 권한은 최소인원에게 부여하고 부여현황에 대해 상시모니터링 수행

    2) 사용자가 생성하는 각 키에 대해서는 관리자를 별도 지정할 수 있어야 하며, 각 조건에 따라 최소한의 권한 부여 등

3 우수 사례

1) 암호화 키 관리 서비스 관리자 부여 현황 모니터링

- **(AWS 관리 콘솔)** '콘솔 홈' → 왼쪽 상단의 검색에 'IAM' 입력 → 주요 기능의 '정책' 클릭 → KMS의 주요권한들을 가지고 있는 정책인 '*AWSKeyManagementServicePowerUser*'를 '검색'에 주고 찾아서 클릭 → '연결된 엔터티' 탭 클릭 → '권한 정책으로 연결됨' 섹션에 해당 권한 정책을 소유한 관리 주체들을 엔터티 이름 컬럼에서 확인합니다.



| 그림 4-3-1 | 암호화 키 관리 서비스에 대한 관리자 부여 현황 모니터링

- **(AWS CLI)** API 환경을 통한 암호키 관리 서비스 관리자 부여 현황 모니터링 → 다음 CLI에서 AWS 어카운트 ID(빨간색)를 이용자의 것으로 변경하고, '*/TestPolicy*' 부분을 '*AWSKeyManagementServicePowerUser*'로 변경하여 실행합니다.

```
aws iam list-entities-for-policy \
    --policy-arn arn:aws:iam::123456789012:policy/TestPolicy
```

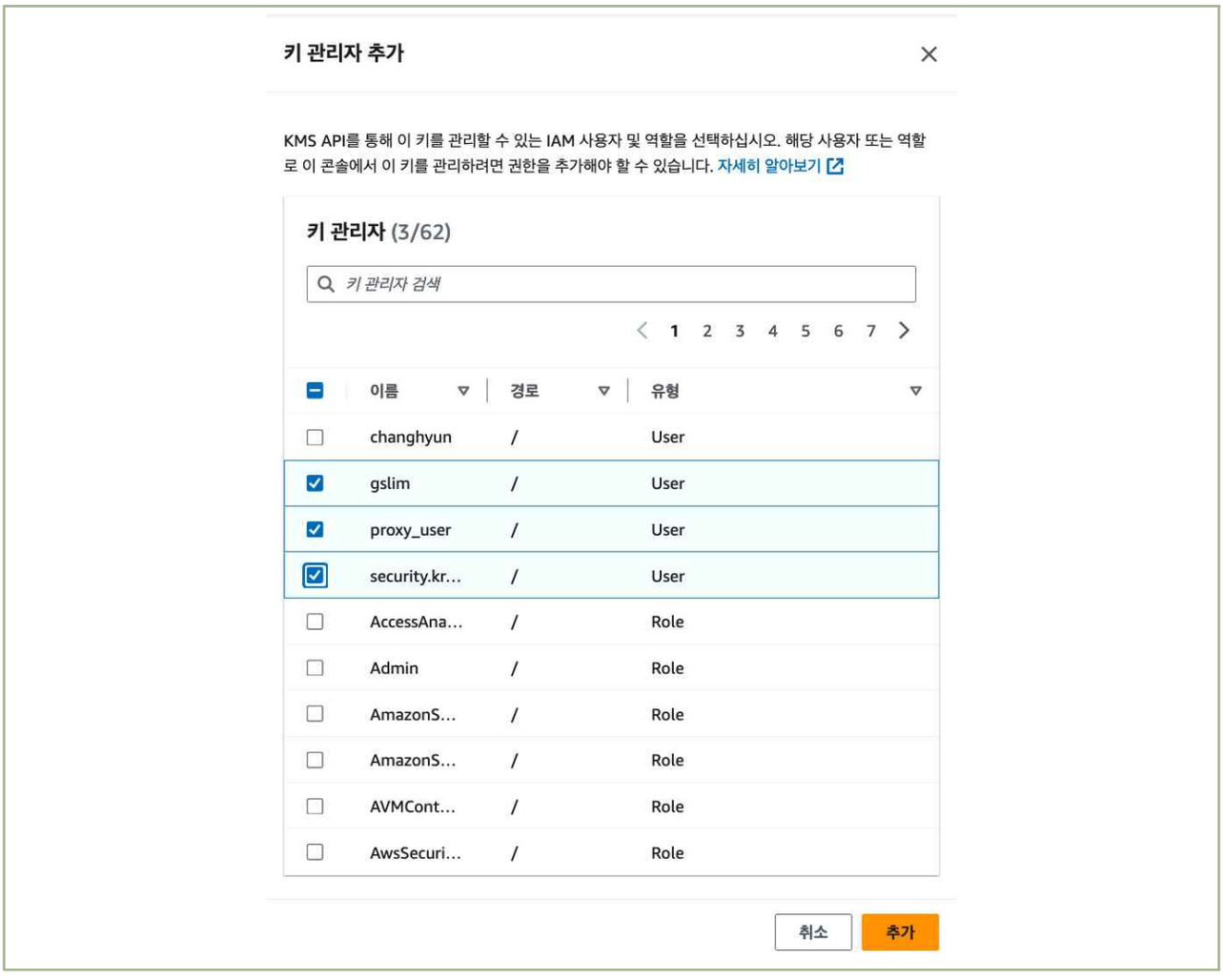
Output:

```
{
    "PolicyGroups": [
        {
            "GroupName": "Admins",
            "GroupId": "AGPACKCEVSQ6C2EXAMPLE"
        }
    ],
    "PolicyUsers": [
        {
            "UserName": "Alice",
            "UserId": "AIDACKCEVSQ6C2EXAMPLE"
        }
    ],
    "PolicyRoles": [
        {
            "RoleName": "DevRole",
            "RoleId": "AROADBQP57FF2AEXAMPLE"
        }
    ],
    "IsTruncated": false
}
```

| 그림 4-3-2 | 암호키 관리 서비스 관리자 부여 현황 모니터링

2) 암호화 키 별 관리자 별도 지정

- **(AWS 관리 콘솔)** '콘솔 홈' → 왼쪽 상단의 검색에 'KMS' 입력 → 'Key Management Service' 클릭 → '고객 관리형 키' 클릭 → '고객 관리형 키' 목록에서 관리자를 별도 지정할 암호화 키를 클릭 → '키 관리자' 섹션에서 '추가' 클릭 → 목록에 있는 관리 주체 후보들 중에서 선택(복수 선택 가능)하고 하단의 '추가' 클릭하여 해당 키 관리자로 등록합니다.



| 그림 4-3-3 | 암호화 키 별 관리자 별도 지정

- **(AWS CLI)** API 환경을 통해, 'AWSKeyManagementServicePowerUser' 정책을 줄 수 있는 대상은 IAM user, IAM group, IAM role 3가지 유형으로, 각각 'attach-user-policy', 'attach group-policy', ' attach-role-policy' 라는 다음 3개의 CLI 명령어를 통해 수행할 수 있습니다.

```
aws iam attach-user-policy \
    --policy-arn arn:aws:iam::aws:policy/AdministratorAccess \
    --user-name Alice
```

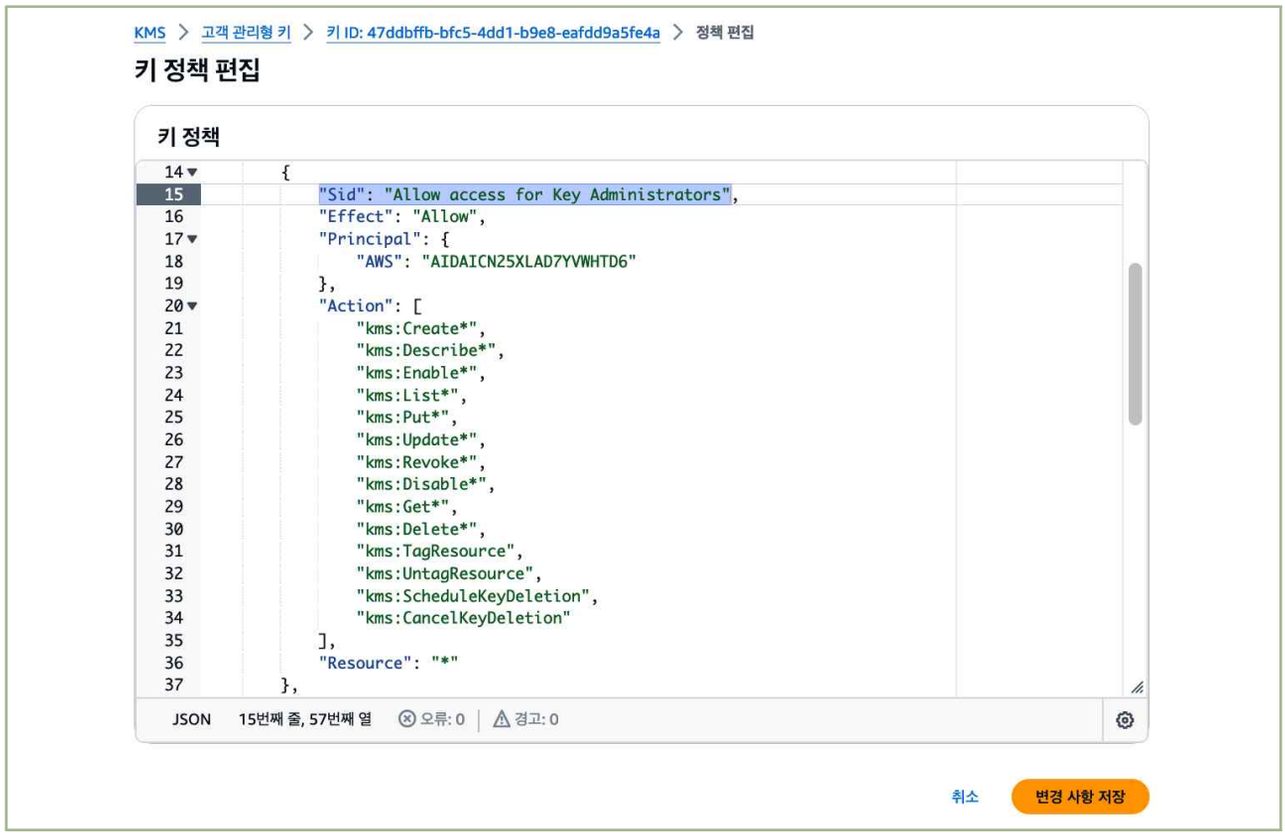
```
aws iam attach-group-policy \
    --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \
    --group-name Finance
```

```
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \
    --role-name ReadOnlyRole
```

| 그림 4-3-4 | CLI명령 - 암호화 키 별 관리자 별도 지정

3) 암호화 키 관리자 최소 권한 부여

- **(AWS 관리 콘솔)** ‘콘솔 홈’ → 왼쪽 상단의 검색에 ‘KMS’ 입력 → ‘Key Management Service’ 클릭 → ‘고객 관리형 키’ 클릭 → 목록에 있는 키들 중에서 권한 관리할 키를 클릭 → ‘키 정책’ 탭에서 ‘편집’ 클릭 → 키 정책 편집 에디터에서 스크롤하여 ‘"*Sid": "Allow access for Key Administrators*"’를 탐색 → ‘Action’ 항목에 있는 기능들 중 필요 없는 기능을 제거하고 오른쪽 아래 ‘변경 사항 저장’을 클릭하여 반영합니다.(각각의 Action에 대한 설명은 링크를 참조)



| 그림 4-3-5 | 암호화 키 관리자에 대한 최소 권한 부여

- **(AWS CLI)** API 환경을 통한 암호키 관리자 최소권한 부여 → (링크) 를 참조하여 ‘key_policy.json’파일을 생성 → 생성된 정책 파일에서 "*Sid": "Allow access for Key Administrators"*를 탐색 → ‘Action’ 부분은 (링크)를 참조하여 필요 없는 Action들을 제거하고 수정 → 다음 CLI에서 타겟 키 ID를 변경하고, ‘key_policy.json’ 파일의 경로를 주어 실행합니다.

Contents of `key_policy.json` :

```
{
    "Version" : "2012-10-17",
    "Id" : "key-default-1",
    "Statement" : [
        {
            "Sid" : "Enable IAM User Permissions",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:root"
            },
            "Action" : "kms:*",
            "Resource" : "*"
        },
        {
            "Sid" : "Allow Use of Key",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:user/test-user"
            },
            "Action" : [
                "kms:DescribeKey",
                "kms:ListKeys"
            ],
            "Resource" : "*"
        }
    ]
}
```

| 그림 4-3-6 | key_policy.json 파일

```
aws kms get-key-policy \
    --policy-name default \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --output text
```

Output:

```
{
    "Version" : "2012-10-17",
    "Id" : "key-default-1",
    "Statement" : [
        {
            "Sid" : "Enable IAM User Permissions",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:root"
            },
            "Action" : "kms:*",
            "Resource" : "*"
            },
            {
            "Sid" : "Allow Use of Key",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:user/test-user"
            },
            "Action" : [ "kms:Describe", "kms:List" ],
            "Resource" : "*"
        }
    ]
}
```

| 그림 4-3-7 | 암호키 관리자 최소 권한 부여

4 참고 사항

1 기준

식별번호	기준	내용
4.4.	**암호키 호출 권한 관리**	**클라우드 암호키 호출 권한을 관리하여야 한다.**

2 설명

- 클라우드 암호키 호출에 관한 사항(암호화, 복호화, 암호키 변경, 삭제 등)은 이용자의 권한 및 업무에 따라 적절하게 부여하고 관리하여야 한다.

  - 예시

    1) 암호키 관리 서비스(KMS)를 통해 암호키 호출 시 목적에 따라 권한 부여

    2) 암호키 호출 권한 현황에 대한 모니터링 및 주기적 검토 수행

3 우수 사례

1) 암호화 키 관리 서비스를 통해 암호키 호출 시 목적에 따라 권한 부여

- **(AWS 관리 콘솔)** ‘콘솔 홈’ → 왼쪽 상단의 검색에 ‘KMS’ 입력 → ‘Key Management Service’ 클릭 → ‘고객 관리형 키’ 클릭 → 목록에 있는 키들 중에서 권한 관리할 키를 클릭 → ‘키 정책’ 탭에서 ‘편집’ 클릭 → 키 정책 편집 에디터에서 스크롤하여 ‘*"Sid": "Allow use of the Key"*를 탐색 → ‘Action’ 항목에 있는 기능들 중 필요 없는 기능을 제거하고 오른쪽 아래 ‘변경 사항 저장’을 클릭하여 반영합니다.(각각의 Action에 대한 설명은 링크 를 참조)

| 그림 4-4-1 | 암호화 키 사용자 최소 권한 부여

- **(AWS CLI)** API 환경을 통한 암호키 사용자 최소권한 부여' → 링크 를 참조하여 'key_policy.json'파일을 생성 → 생성된 정책 파일에서 *"Sid": "Allow use of the Key"*를 탐색 → 'Action'부분을 링크를 참조하여 필요 없는 Action들을 제거하고 수정 → 다음 CLI에서 타겟 키 ID를 변경하고, 'key_policy.json' 파일의 경로를 주어 실행합니다.

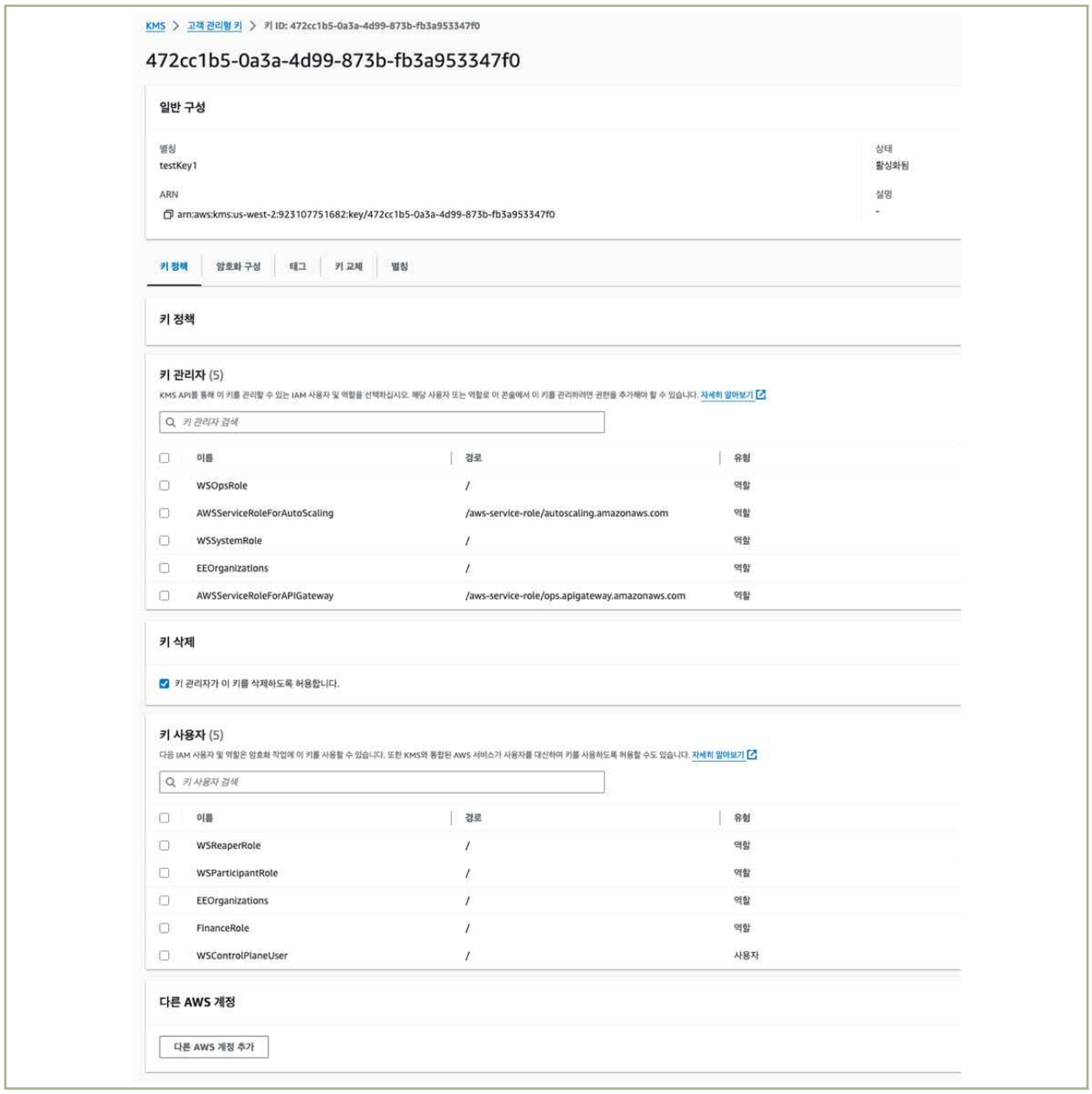
| 그림 4-4-2 | key_policy.json 파일

```
aws kms put-key-policy \
    --policy-name default \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --policy file://key_policy.json
```

| 그림 4-4-3 | 암호키 사용자 최소 권한 부여

2) 암호화 키 호출 권한 현황에 대한 모니터링 및 주기적 검토

- **(AWS 관리 콘솔)** '콘솔 홈' → 왼쪽 상단의 검색에 'KMS' 입력 → 'Key Management Service' 클릭 → '고객 관리형 키' 클릭 → 목록에 있는 키들 중에서 사용 권한자를 조회할 키를 클릭 → '키 정책' 탭을 스크롤해서 '키 사용자' 섹션에서 해당 암호화 키를 사용할 권한을 가진 주체들을 식별합니다.



| 그림 4-4-4 | 암호키 호출 권한 현황에 대한 모니터링

- **(AWS CLI)** API 환경을 통한 암호키 호출 권한 현황에 대한 모니터링 → 조회할 Key-ID 값을 주고 다음 CLI를 실행 → 리턴된 JSON 구문에서 *"Sid": "Allow use of the Key"*를 탐색 → '*Principal*'부분에 해당 키의 사용권한을 소유한 주체들을 식별합니다.

```
aws kms get-key-policy \
    --policy-name default \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --output text
```

Output:

```
{
    "Version" : "2012-10-17",
    "Id" : "key-default-1",
    "Statement" : [
        {
            "Sid" : "Enable IAM User Permissions",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:root"
            },
            "Action" : "kms:*",
            "Resource" : "*"
            },
            {
            "Sid" : "Allow Use of Key",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:user/test-user"
            },
            "Action" : [ "kms:Describe", "kms:List" ],
            "Resource" : "*"
        }
    ]
}
```

| 그림 4-4-5 | 암호키 호출 권한 현황에 대한 모니터링

4 참고 사항

1 기준

식별번호	기준	내용
4.5.	안전한 암호화 알고리즘 적용	암호화 기능 이용 시 안전한 암호화 알고리즘을 적용하여야 한다.

2 설명

- 암호화 기능 이용 시 안전한 암호화 알고리즘을 적용하여야 한다.(또는 확인하여야 한다.)
  - 예시
    1) 이용자가 관리하는 암호키로 암호화 기능 적용 시 안전한 암호화 알고리즘 적용(금융부문 암호기술 활용 가이드 등 참고)
    2) 클라우드 KMS 서비스를 통해 암호화 시 안전한 암호화 알고리즘을 제공하는지 확인

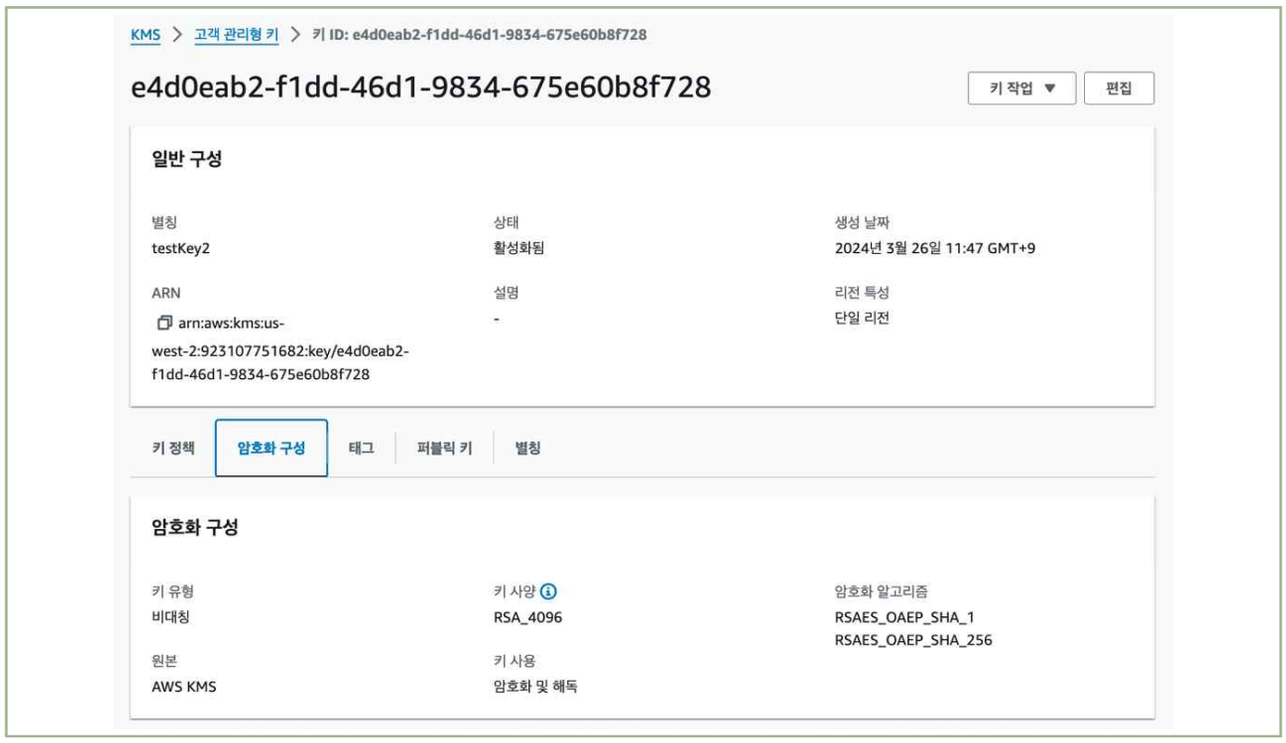
3 우수 사례

1) 이용자가 관리하는 암호화 키로 암호화 기능 적용 시 안전한 암호화 알고리즘 적용

- 금융부문 암호기술 활용 가이드 등 참고하여 이용자가 안전한 알고리즘을 선택하고 적용할 수 있습니다.

2) AWS KMS 서비스를 통해 암호화 시 안전한 암호화 알고리즘을 제공하는지 확인

- **(AWS 관리 콘솔)** ‘콘솔 홈’ → 왼쪽 상단의 검색에 ‘KMS’ 입력 → ‘Key Management Service’ 클릭 → ‘고객 관리형 키’ 클릭 → 목록에 있는 키들 중에서 알고리즘을 확인할 키를 클릭 → ‘암호화 구성’ 탭 클릭하여 해당 키의 상세 정보에 있는 ‘암호화 알고리즘’을 확인합니다.



| 그림 4-5-1 | 안전한 암호화 알고리즘을 제공하는지 확인

- **(AWS CLI)** API 환경을 통한 안전한 암호화 알고리즘을 제공하는지 확인 → 다음 CLI를 조회할 Key-ID를 주고 실행 → 리턴 결과 중 "*EncryptionAlgorithms*"에서 해당 키의 암호화 알고리즘을 확인합니다.(아래 샘플의 '*SYMMETRIC_DEFAULT*' 는 AWS KMS가 대칭키 암호화에 사용하는 AES 256 알고리즘을 말하며, 링크 참조)

```
aws kms describe-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Output:


```
{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "Arn": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": 1.499288695918E9,
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "Description": "",
    "Enabled": true,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "AWS_KMS",
    "EncryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
    ]
  }
}
```


| 그림 4-5-2 | 안전한 암호화 알고리즘을 제공하는지 확인

3) AWS KMS가 제공하는 암호화 알고리즘 내역

- AWS KMS가 제공하는 각각의 암호화 키 유형별 자세한 설명은 다음 링크에서 확인 할 수 있습니다.
  - 대칭키(SYMMETRIC_DEFAULT) 사양 - 링크
  - RSA 키 사양 - 링크
  - 타원 곡선 키 사양 - 링크
  - SM2키 사양(중국 리전 전용) - 링크

4 참고 사항

- N/A

5. 로깅 및 모니터링 관리

5.1. 가상자원 이용(생성, 삭제, 변경 등)에 관한 행위추적성 확보
5.2. 가상자원 이용 행위추적성 증적 모니터링
5.3. 이용자 가상자원 모니터링 기능 확보
5.4. API 사용(호출대상, 호출자, 호출일시 등)에 관한 행위추적성 확보
5.5. 네트워크 관련 서비스(VPC, 보안 그룹, ACL 등)에 관한 행위추적성 확보
5.6. 계정 변동사항에 대한 행위추적성 확보
5.7. 계정 변경사항에 관한 모니터링 수행

5 로깅 및 모니터링 관리

1 기준

식별번호	기준	내용
5.1.	**가상자원 이용(생성, 삭제, 변경 등)에 관한 행위추적성 확보**	**이용자의 가상자원(서버, 데이터베이스, 스토리지 등) 이용 관련 행위에 대한 추적성(로그 등)을 확보하여야 한다.**

2 설명

- 이용자의 가상자원 이용 관련 일련의 행위에 대한 추적성을 확보할 수 있는 방안이 마련되어야 한다.

  - 예시

    1) 가상자원 변경 사항에 관한 행위(생성, 변경, 삭제 등)
    2) 가상자원에 접속한 일시, 접속자 및 접근을 확인할 수 있는 접근기록
    3) 가상자원을 사용한 일시, 사용자 및 가상자원의 형태(서버, 데이터베이스, 스토리지 등)를 확인할 수 있는 접근기록
    4) 가상자원내 전산자료의 처리 내용을 확인할 수 있는 사용자 로그인, 액세스 로그 등 접근기록

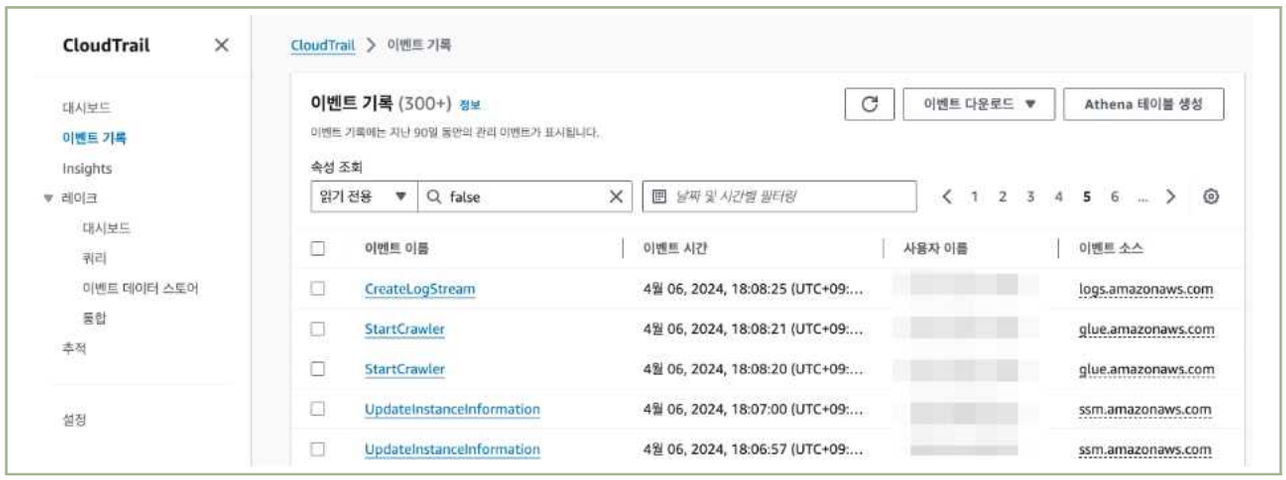
3 우수 사례

1) 가상자원 변경 사항에 관한 행위(생성, 변경, 삭제 등)

(1) CloudTrail 이벤트 기록 확인

- AWS CloudTrail은 AWS 계정 내에서의 모든 행위(가상자원 생성, 변경, 삭제, 접속 등)를 별도 비용 없이 저장하며, 90일 내의 로그는 별도 설정 없이 확인할 수 있습니다.

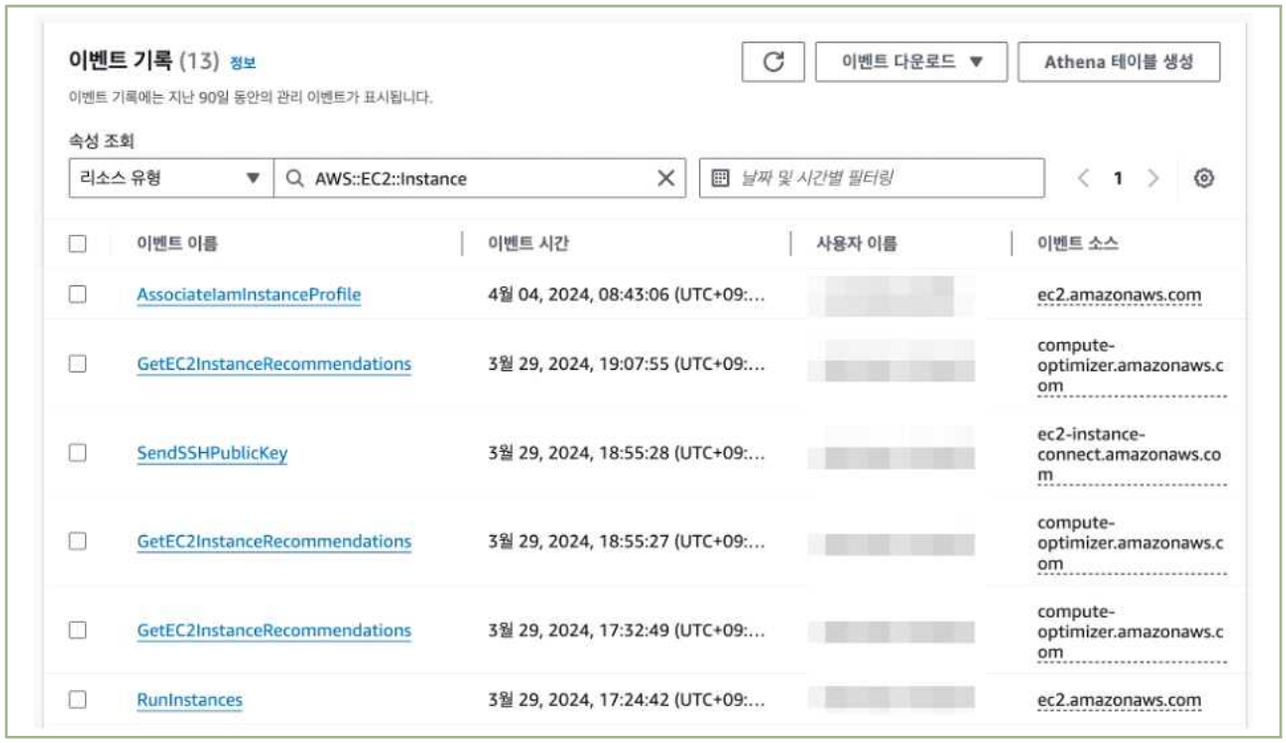
- **(AWS 관리 콘솔)** ‘콘솔 홈’ → ‘CloudTrail’ → ‘이벤트 기록’에서, 가상자원 변경 사항에 관한 행위를 확인할 수 있습니다.



| 그림 5-1-1 | AWS CloudTrail 이벤트 기록

(2) Amazon EC2 관련 행위 확인

- **(AWS 관리 콘솔)** '콘솔 홈' → 'CloudTrail' → '이벤트 기록' → '리소스 유형' 에서 'AWS::EC2::Instance' 를 입력하여 조회하면, Amazon EC2와 관련된 모든 행위를 확인할 수 있습니다.



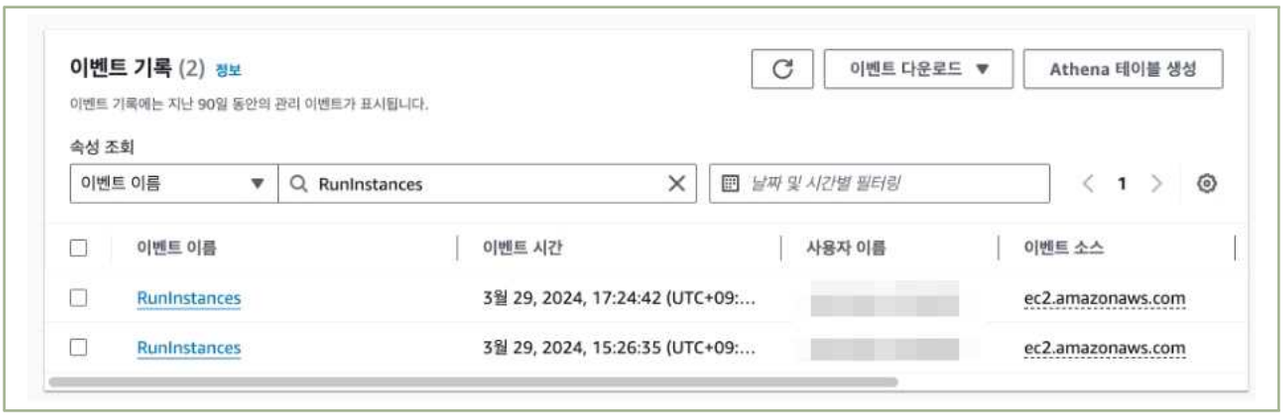
| 그림 5-1-2 | Amazon EC2 관련 행위 확인

- 조회를 할 수 있는 추가 리소스 유형 예시는 다음과 같습니다.

리소스 유형	설명
AWS::RDS::DBInstance	Amazon DB 인스턴스와 관련된 이벤트. RDS DB 인스턴일수도 있고, Aurora 클러스터의 DB 인스턴일 수도 있음
AWS::S3::Bucket	Amazon S3(Simple Storage Service) 버킷과 관련된 이벤트
AWS::EC2::VPC	Amazon VPC(Virtual Private Cloud)와 관련된 이벤트

(3) Amazon EC2 생성 행위 확인

- **(AWS 관리 콘솔)** '콘솔 홈' → 'CloudTrail' → '이벤트 이름' 에서 'RunInstances' 를 입력하면, Amazon EC2 생성 이벤트를 확인할 수 있습니다.



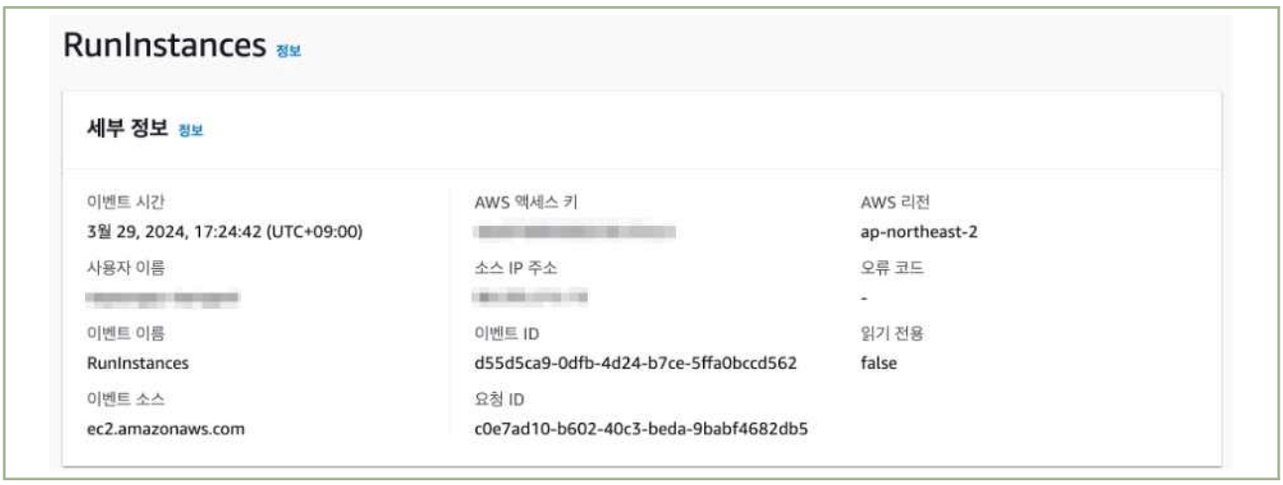
| 그림 5-1-3 | Amazon EC2 생성 확인

- 확인할 수 있는 이벤트 이름에 대한 예시는 다음과 같습니다.

리소스 유형	설명
CreateDBInstance	새로운 DB 인스턴스 생성 이벤트
CreateBucket	새로운 S3 버킷 생성 이벤트
DescribeVpcs	하나 이상의 VPC 정보 확인 이벤트

(4) CloudTrail 이벤트 상세 정보

- **(AWS 관리 콘솔)** '콘솔 홈' → 'CloudTrail' → '이벤트 이름' 을 클릭하면 이벤트 시간, 사용자 이름, 소스 IP 주소 등 더욱 상세한 정보와 JSON형태의 이벤트 레코드를 확인할 수 있습니다.



| 그림 5-1-4 | 이벤트 세부 정보

| 그림 5-1-5 | 이벤트 레코드 정보

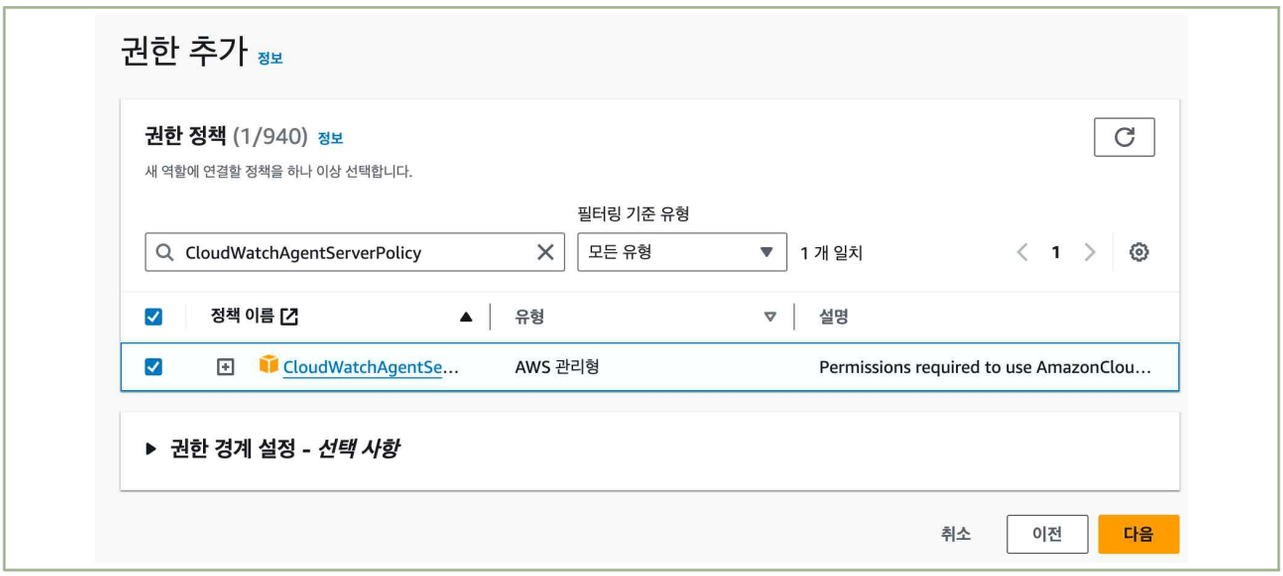
2) 가상자원에 접속한 일시, 접속자 및 접근을 확인할 수 있는 접근기록

- 가상자원(EC2)에 대한 접속 기록은 SSH를 이용한 접근과 EC2 Instance Connect를 이용한 접근으로 분류 할 수 있습니다.

(1) SSH를 이용한 접속 기록

- CloudWatch Agent를 이용하여 /var/log/wtmp 로그를 수집할 수 있습니다.
- EC2에서 CloudWatch Agent를 수행하기 위한 IAM 역할을 생성하고 권한을 추가합니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → 'IAM' → '역할' → '역할 생성' → 'AWS 서비스' → 서비스 또는 사용 사례에서 'EC2' 선택 → '다음' → 목록에서 'CloudWatchAgentServerPolicy' 정책을 선택 → '다음' → 원하는 역할 이름을 정하고 (예, CloudWatchAgentServerRole) → '역할 생성' 으로 생성 및 권한을 추가할 수 있습니다.



| 그림 5-1-6 | IAM 역할 생성 - 권한 추가

- 다음 EC2에 IAM 역할을 연결합니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → 'EC2' → '인스턴스' → 연결할 EC2 선택 → '작업' → '보안' → 'IAM 역할 수정' → IAM 역할에서 방금 만들어진 역할 선택으로 IAM 역할을 업데이트 할 수 있습니다.

| 그림 5-1-7 | EC2의 IAM 역할 업데이트

- 아래 명령어를 이용하여 EC2 인스턴스에 CloudWatch Agent를 설치합니다.

- **(가상자원 내 CLI)**

- CloudWatch Agent를 다운로드 합니다.

```
wget https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm
```

```
[ec2-user@ip-10-0-1-4 ~]$ wget https://amazoncloudwatch-agent.s3.amazonaws.com/a
mazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm
--2024-04-06 14:00:27--  https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_
linux/amd64/latest/amazon-cloudwatch-agent.rpm
Resolving amazoncloudwatch-agent.s3.amazonaws.com (amazoncloudwatch-agent.s3.ama
zonaws.com)... 3.5.17.212, 3.5.28.40, 52.216.60.193, ...
Connecting to amazoncloudwatch-agent.s3.amazonaws.com (amazoncloudwatch-agent.s3
.amazonaws.com)¦3.5.17.212¦:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 100451634 (96M) [application/octet-stream]
Saving to: ‘amazon-cloudwatch-agent.rpm’

amazon-cloudwatch-a 100%[===================>]  95.80M  7.96MB/s    in 22s

2024-04-06 14:00:50 (4.31 MB/s) - ‘amazon-cloudwatch-agent.rpm’ saved [100451634
/100451634]
```

| 그림 5-1-8 | CloudWatch Agent 다운로드

- CloudWatch Agent를 설치 합니다.

```
sudo rpm -U ./amazon-cloudwatch-agent.rpm
```

- CloudWatch Agent 설정 파일을 생성합니다. 설정파일은 Wizard를 통해서 간단하게 설정할 수 있으며, EC2 지표 이외에도 로그 파일을 설정하여 CloudWatch로 전송하는 것도 가능합니다.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard
```

```
Are you satisfied with the above config? Note: it can be manually customized after the wizard co
mpletes to add additional items.
1. yes
2. no
default choice: [1]:
1
Do you have any existing CloudWatch Log Agent (http://docs.aws.amazon.com/AmazonCloudWatch/lates
t/logs/AgentReference.html) configuration file to import for migration?
1. yes
2. no
default choice: [2]:
2
Do you want to monitor any log files?
1. yes
2. no
default choice: [1]:
1
Log file path:
/var/log/wtmp
Log group name:
default choice: [wtmp]

Log group class:
1. STANDARD
2. INFREQUENT_ACCESS
default choice: [1]:

Log stream name:
default choice: [{instance_id}]
```

| 그림 5-1-9 | CloudWatch Agent 설정 파일 생성

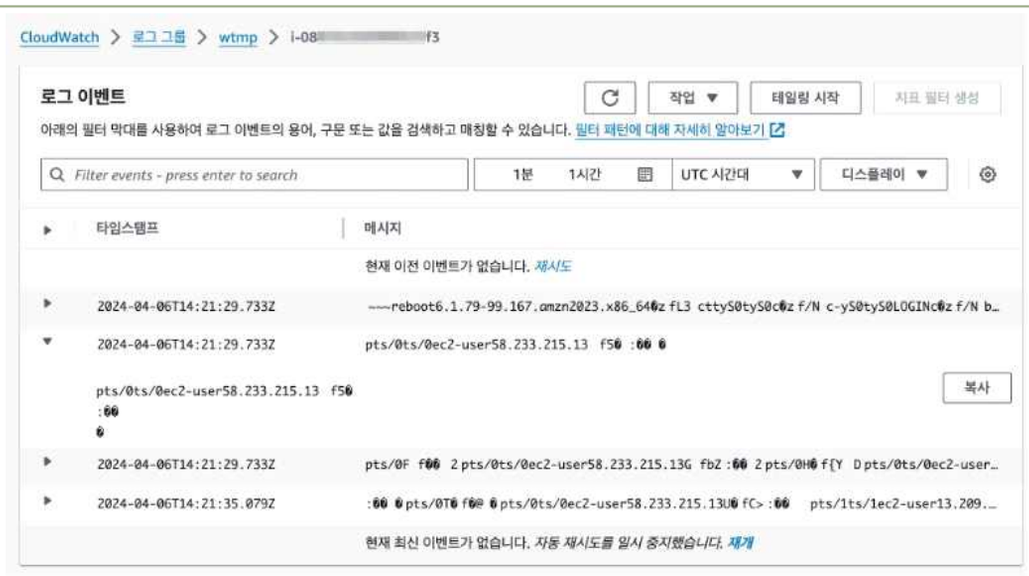
- CloudWatch Agent를 실행합니다.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:[configuration-file-path] -s
```

```
[ec2-user@ip-10-0-1-4 bin]$ sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json -s
****** processing amazon-cloudwatch-agent ******
I! Trying to detect region from ec2 D! [EC2] Found active network interface I! imds retry client will retry 1 timesSuccessfully fetched the config and saved in /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/file_config.json.tmp
Start configuration validation...
2024/04/06 14:21:27 Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/file_config.json.tmp ...
2024/04/06 14:21:27 I! Valid Json input schema.
2024/04/06 14:21:27 D! ec2tagger processor required because append_dimensions is set
2024/04/06 14:21:27 D! pipeline hostDeltaMetrics has no receivers
2024/04/06 14:21:27 Configuration validation first phase succeeded
I! Detecting run_as_user...
I! Trying to detect region from ec2
D! [EC2] Found active network interface
I! imds retry client will retry 1 times
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent -schematest -config /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml
Configuration validation second phase succeeded
Configuration validation succeeded
amazon-cloudwatch-agent has already been stopped
Created symlink /etc/systemd/system/multi-user.target.wants/amazon-cloudwatch-agent.service → /etc/systemd/system/amazon-cloudwatch-agent.service.
[ec2-user@ip-10-0-1-4 bin]$
```

| 그림 5-1-10 | CloudWatch Agent 실행 화면

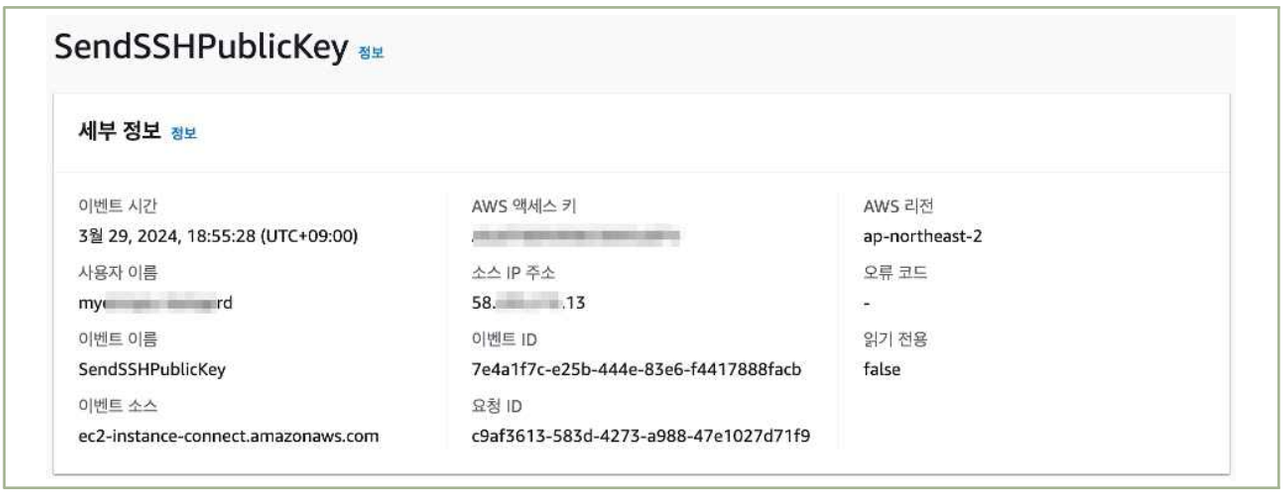
- CloudWatch 에서 SSH를 이용한 EC2 접속 기록을 확인할 수 있습니다.



| 그림 5-1-11 | SSH를 이용한 EC2 접속 기록 확인

(2) EC2 Instance Connect를 이용한 접속 기록

- **(AWS 관리 콘솔)** '이벤트 소스' → 'ec2-instance-connect.amazonaws.com' 입력하면, 'SendSSHPublicKey' 이벤트를 통해, 접속자 및 접속IP 등의 세부 정보를 확인할 수 있습니다.



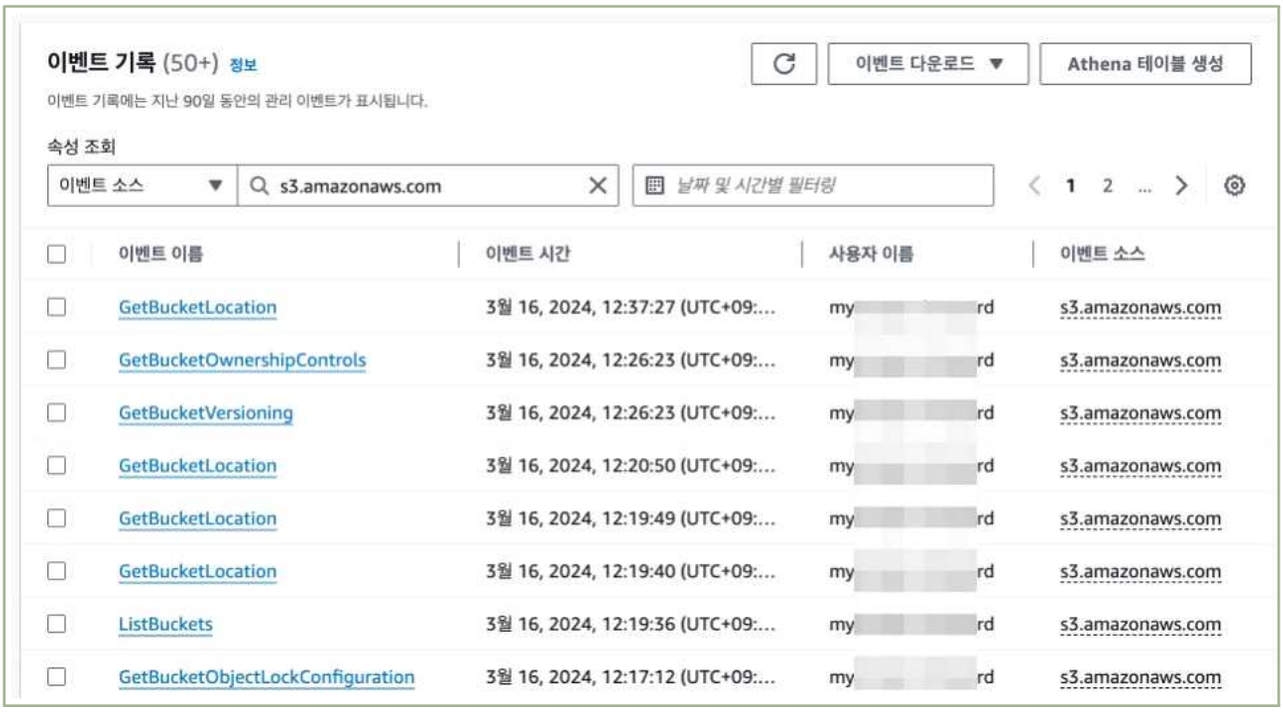
| 그림 5-1-12 | EC2 Instance Connect를 이용한 접속 기록

3) 가상자원을 사용한 일시, 사용자 및 가상자원의 형태(서버, 데이터베이스, 스토리지 등)를 확인할 수 있는 접근기록

- 서버, DB, 스토리지 등에 대한 사용 행위는 CloudTrail에서 확인할 수 있습니다.

(1) S3 사용에 관한 행위

- **(AWS 관리 콘솔)** '콘솔 홈' → 'CloudTrail' → '이벤트 기록' 에서 '이벤트 소스' 에 's3.amazonaws.com' 을 입력하면 S3 사용에 관한 행위를 확인할 수 있습니다.



| 그림 5-1-13 | S3 사용에 관한 행위 기록

- S3 뿐만 아니라 데이터베이스인 RDS나 컴퓨팅인 EC2 등에 대한 이벤트 소스 확인도 가능합니다.

이벤트 소스	설명
s3.amazonaws.com	S3 서비스
rds.amazonaws.com	RDS (Relational Database Service) 서비스
ec2.amazonaws.com	EC2 (Elastic Compute Cloud) 서비스

- **(AWS CLI)** 'aws cloudtrail lookup-events'명령을 통해 이용하여 CloudTrail 이벤트를 조회할 수 있습니다.

```
aws cloudtrail lookup-events –max-items 〈정수값〉
```

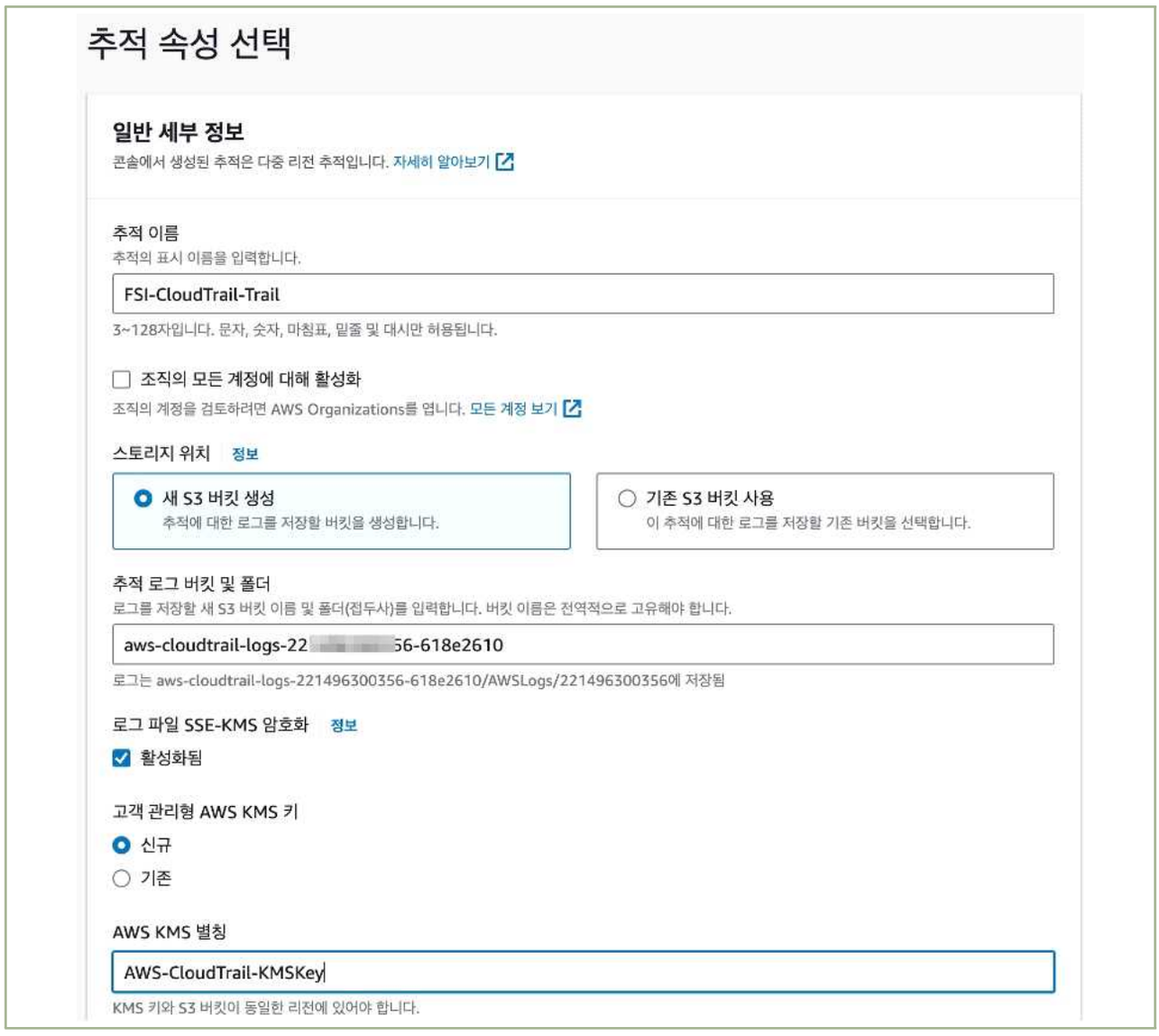
- 또한 아래와 같은 옵션을 추가하여 원하는 이벤트를 조회하는데 도움을 받을 수 있습니다.

옵션	설명
--max-items 〈정수값〉	반환할 이벤트 수를 지정할 수 있으며, 1에서 최대 50까지 정의 가능
--start-time 〈타임스탬프〉	지난 90일간의 이벤트에 대해 시간 범위를 지정할 수 있으며, 모든 타임스탬프는 UTC 로 표시
--lookup-attributes 〈정수값〉	속성 키/값 페어를 정의할 수 있으며, 값 이름은 대/소문자 구분

4 참고 사항

- AWS CloudTrail에 저장되는 로그는 기본 90일까지만 저장되고, 90일 이상 된 로그는 자동으로 삭제됨. 90일 이상 된 로그는 'AWS CloudTrail 추적' 기능을 통하여 Amazon S3에 영구 저장이 가능합니다.
- CloudTrail의 추적 기능을 이용하여 아래와 같은 정보를 참고하여 로그 추적 내용을 생성하여 저장할 수 있습니다.

입력 정보	설명
추적 이름	추적의 표시 이름을 입력
스토리지 위치	CloudTrail 로그를 저장할 S3를 새로 생성할지, 기존 것을 사용할지 설정. 로그는 버킷명/AWSLogs/계정ID 에 저장됨.
AWS KMS 별칭	저장된 로그를 암호화할 KMS 별칭 입력



| 그림 5-1-14 | CloudTrail 추적 생성

- 추적은 1개까지 무료로 생성 및 전송 할 수 있으며, 2개부터는 사용량에 대한 별도의 비용이 부과됩니다.

1 기준

식별번호	기준	내용
5.2.	가상자원 이용 행위추적성 증적 모니터링	**이용자의 가상자원(서버, 데이터베이스, 스토리지 등) 이용 관련 행위에 대한 추적성(로그 등)을 확보하여야 한다.** **가상자원 이용에 관한 행위추적성 증적에 대해 모니터링 및 주기적 검토를 수행하여야 한다.**

2 설명

○ 클라우드 가상자원 이용에 관한 행위추적성 증적에 대해 모니터링 및 주기적 검토를 수행하여야 한다.

- 예시

1) 클라우드 가상자원 이용에 관한 행위추적성 증적(ex. 감사로그 등)에 대한 상시 모니터링 수행
2) 금융회사 내부규정 등 관련 규정을 통해 수립된 검토 기간에 맞추어 클라우드 가상자원 이용에 관한 행위추적성 증적에 대한 주기적 검토 수행

3 우수 사례

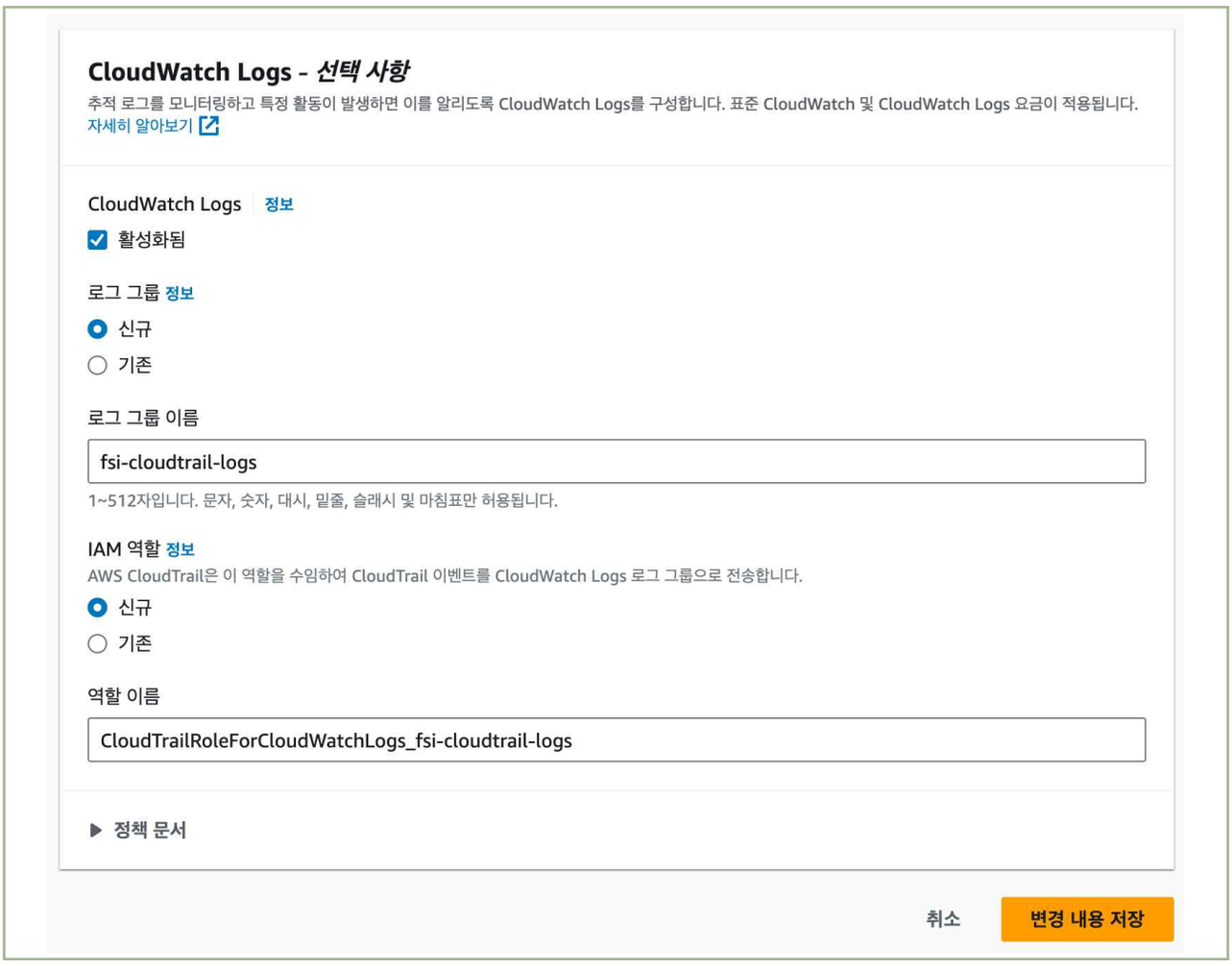
1) 클라우드 가상자원 이용에 관한 행위추적성 증적(ex. 감사로그 등)에 대한 상시 모니터링 수행

- Amazon CloudWatch 경보 생성을 통하여, 가상자원 이용 행위에 대한 추적성(로그 등)을 확인할 수 있는 CloudTrail 로그에 대한 모니터링을 할 수 있습니다.

(1) CloudTrail 추적 생성

○ **(AWS 관리 콘솔)**

- 신규로 CloudTrail 추적을 생성해야 할 경우 '콘솔 홈' → 'CloudTrail' → '추적' → '추적 생성'을 통하여 CloudTrail 로그를 S3에 저장할 수 있습니다. 이때 CloudWatch Logs에도 CloudTrail 로그를 보내도록 설정이 가능합니다.
- 기존 CloudTrail 추적이 있는 경우 '콘솔 홈' → 'CloudTrail' → '추적' → 원하는 추적 선택 → CloudWatch Logs에서 '편집' → CloudWatch Logs의 '활성화됨'을 체크하여 설정이 가능합니다.



| 그림 5-2-1 | CloudWatch Logs 설정

- CloudWatch Logs 로그 그룹에 이벤트를 전송하도록 추적을 구성하면, 일반적으로 API 호출 후 평균 5분 이내에 로그 그룹에 이벤트가 전달됩니다.

(AWS CLI)

- 기존 로그 그룹이 없는 경우 CloudWatch Logs create-log-group 명령을 사용하여 CloudWatch Logs 로그 그룹을 로그 이벤트의 전달 엔드포인트로 생성할 수 있습니다.

```
aws logs create-log-group --log-group-name [name]
```

- CloudTrail이 CloudWatch Logs 로그 그룹에 이벤트를 전송할 수 있게 하는 역할 생성하기 위해, 아래와 같이 정책 문서가 포함된 JSON 파일 생성합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

| 그림 5-2-2 | assume_role_policy_document.json

- 다음 명령을 실행하여, CloudTrail에 대해 AssumeRole 권한이 있는 역할 생성합니다.

```
aws iam create-role --role-name [role_name] --assume-role-policy-document file://<path to assume_role_policy_document>.json
```

- 지정된 로그 그룹에 CloudWatch Logs 로그 스트림을 생성하고, 이 로그 스트림에 CloudTrail 이벤트를 전달하는 데 필요한 권한을 CloudTrail에 부여하기 위해 아래 내용의 파일 생성합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {

      "Sid": "AWSCloudTrailCreateLogStream2014110",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream"
      ],
      "Resource": [

"arn:aws:logs:[region:accountID]:log-group:[log_group_name]:log-stream:[accountID]_CloudTrail_[region]*"
      ]

    },
    {
      "Sid": "AWSCloudTrailPutLogEvents20141101",
```

```
        "Effect": "Allow",
        "Action": [
          "logs:PutLogEvents"
        ],
        "Resource": [

"arn:aws:logs:[region:accountID]:log-group:[log_group_name]:log-stream:[accountID]_CloudTrail_
[region]*"
        ]
      }
   ]
}
```

| 그림 5-2-3 | role-policy-document.json

- 그 다음 정책을 역할에 적용합니다.

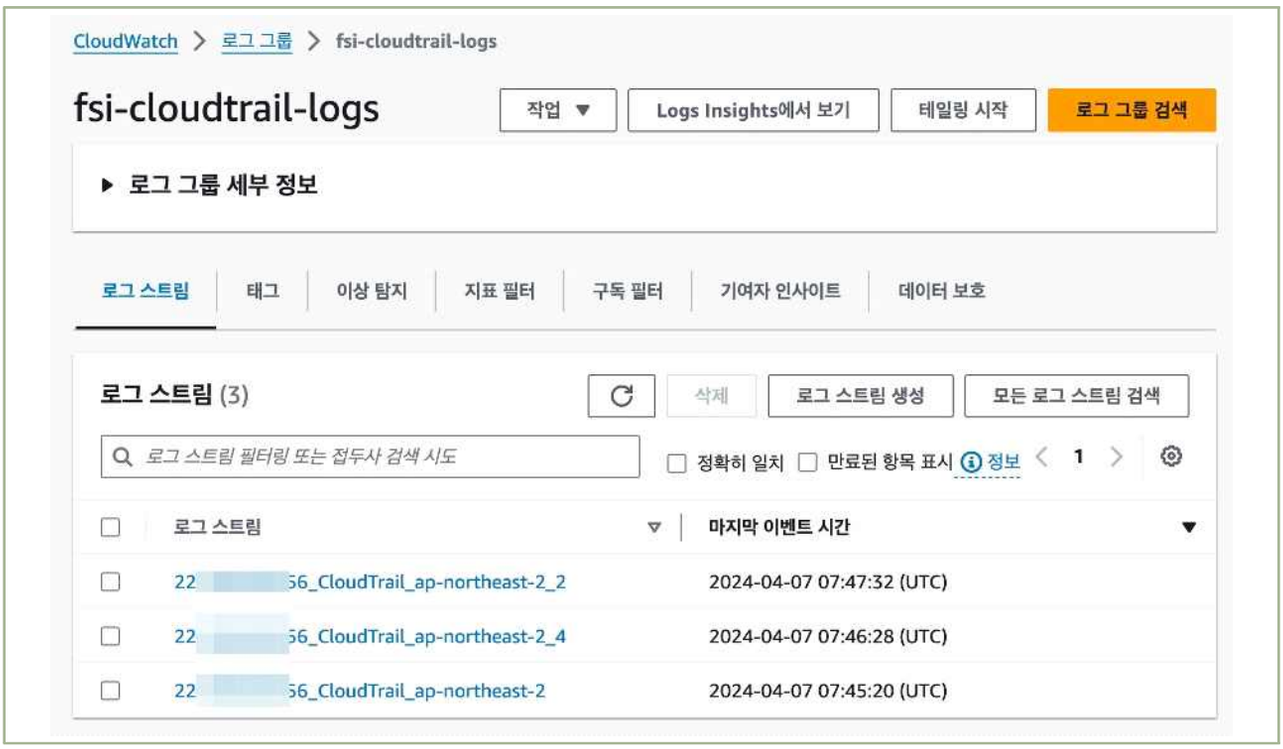
```
aws iam put-role-policy --role-name [role_name] --policy-name cloudtrail-policy
--policy-document file://<path to role-policy-document>.json
```

- 마지막으로 위에서 생성한 로그 그룹을 기존의 CloudTrail에 업데이트 합니다.

```
aws cloudtrail update-trail --name [trail_name] --cloud-watch-logs-log-group-arn
[log_group_arn] --cloud-watch-logs-role-arn [role_arn]
```

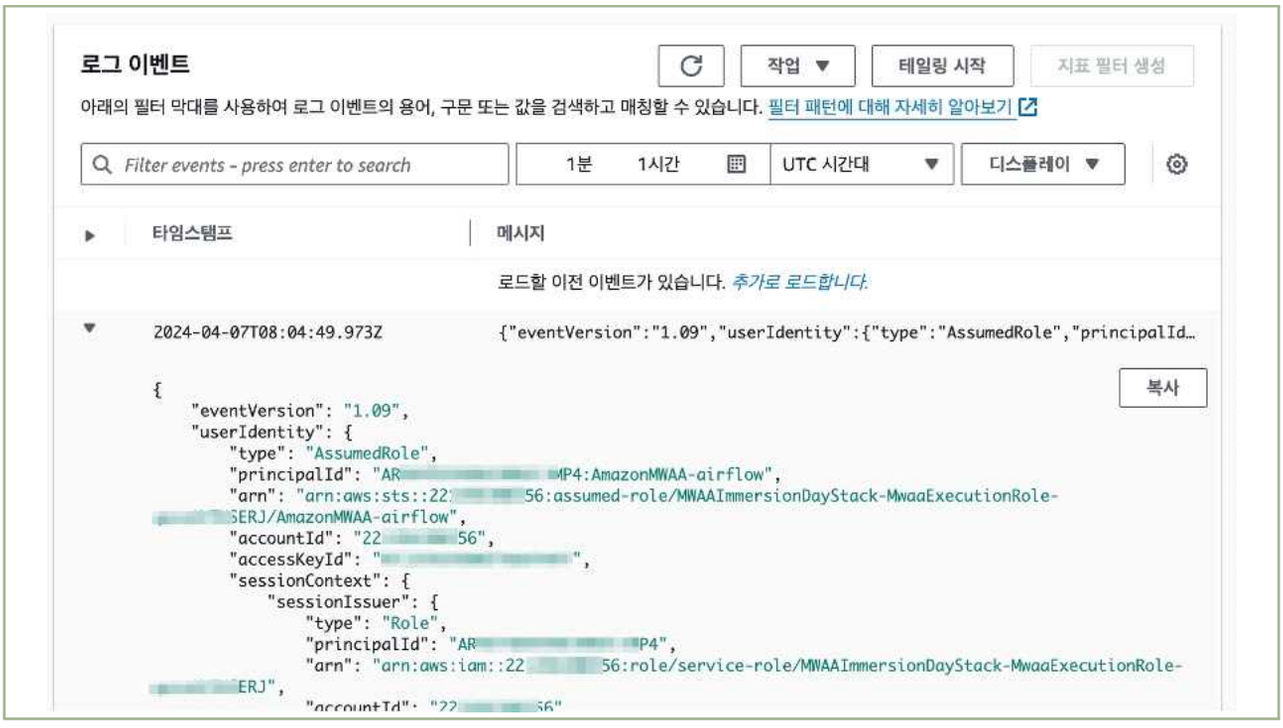
(2) CloudWatch 에서 이벤트 확인

- **(AWS 관리 콘솔)** ‘콘솔 홈’ → ‘CloudWatch’ → ‘로그’ → ‘로그 그룹’ → 추적에 대해 지정한 로그 그룹 선택 합니다.



| 그림 5-2-4 | CloudWatch 로그 그룹에서 이벤트 확인

- 원하는 로그 스트림을 선택한 후 '펼침' 버튼을 선택하면 이벤트에 대한 세부 내용을 확인할 수 있습니다.



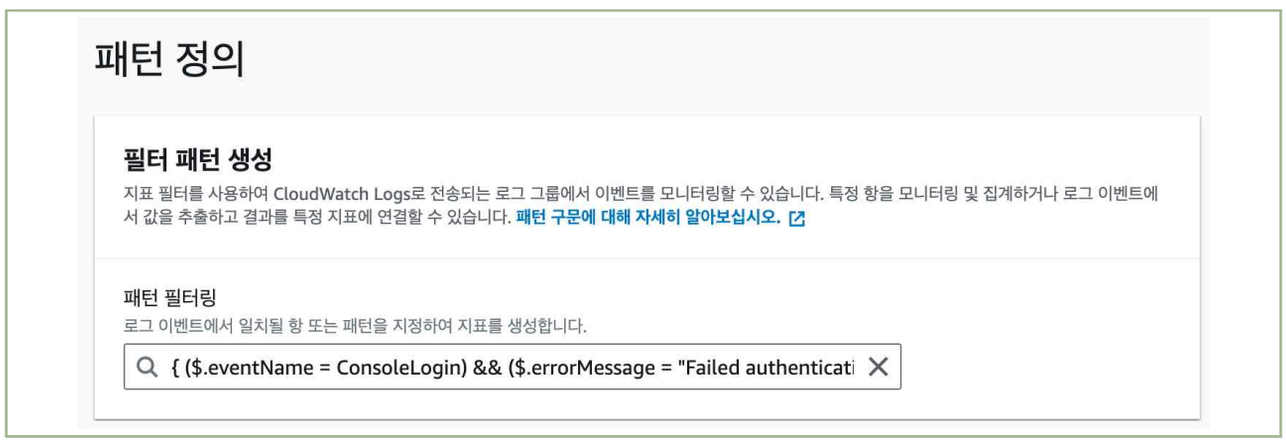
| 그림 5-2-5 | 이벤트 세부 내용 확인

(3) CloudWatch 경보 생성

- CloudWatch를 이용하여 필요한 상황에 대한 경보를 생성할 수 있습니다.
- 본 참고서에서는 '5분 동안 3회 이상 AWS 관리 콘솔에 대한 로그인을 실패하였을 때' 에 대한 경보 생성을 예시로 합니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → 'CloudWatch' → '로그' → '로그 그룹' → 추적에 대해 생성한 로그 그룹 선택 → '지표 필터' → '지표 필터 생성' 을 클릭하고 '필터 패턴 생성' 의 패턴 필터링을 아래와 같이 입력합니다.

```
{ ($.eventName = ConsoleLogin) && ($.errorMessage = "Failed authentication") }
```

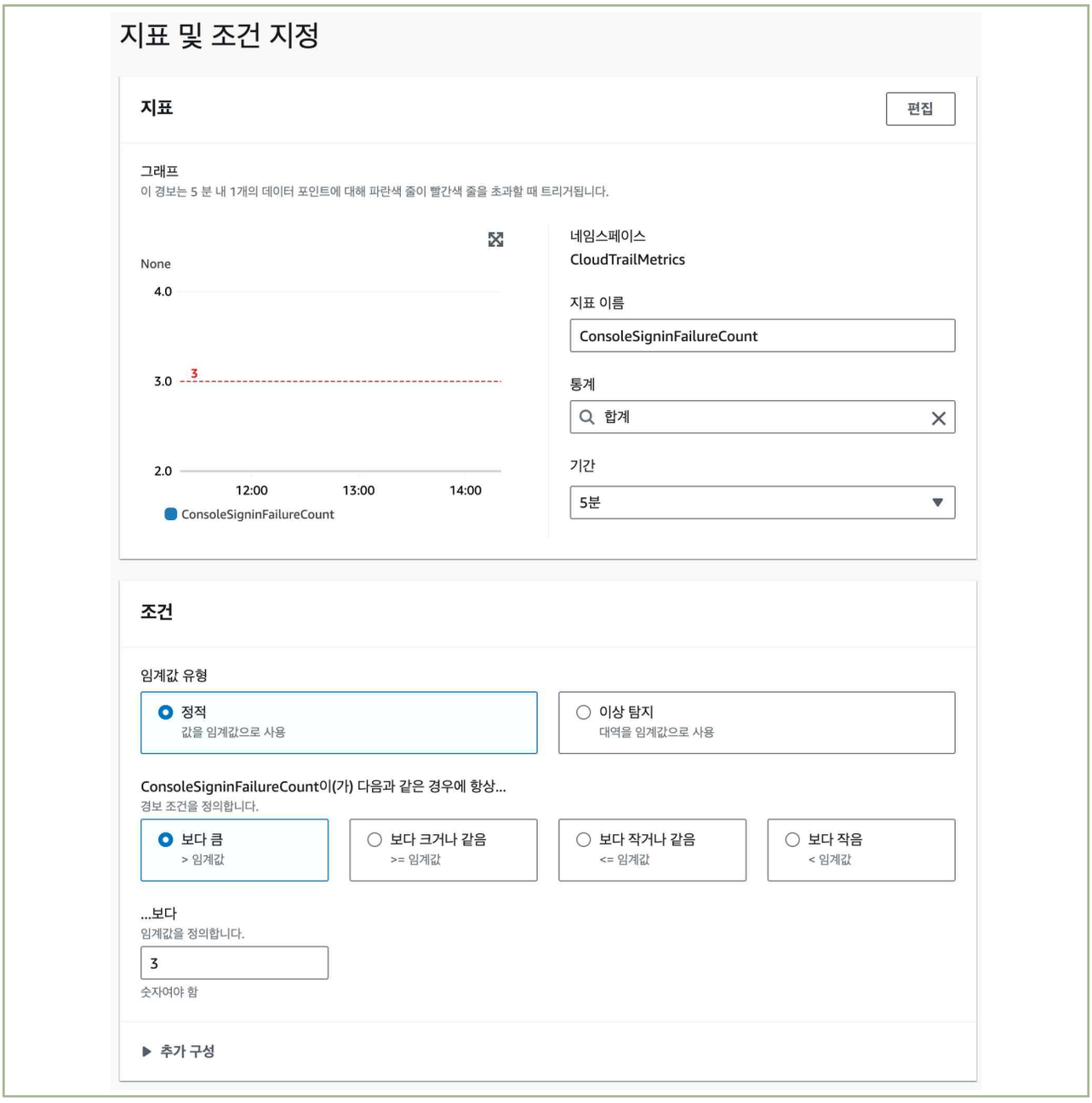


| 그림 5-2-6 | 패턴 정의

- 이름 필터링에 ConsoleSignInFailures 입력
- 지표 네임스페이스에 CloudTrailMetrics 입력
- 지표 이름에 ConsoleSigninFailureCount 입력
- 지표 값에 1입력 → '다음' → '지표 필터 생성'

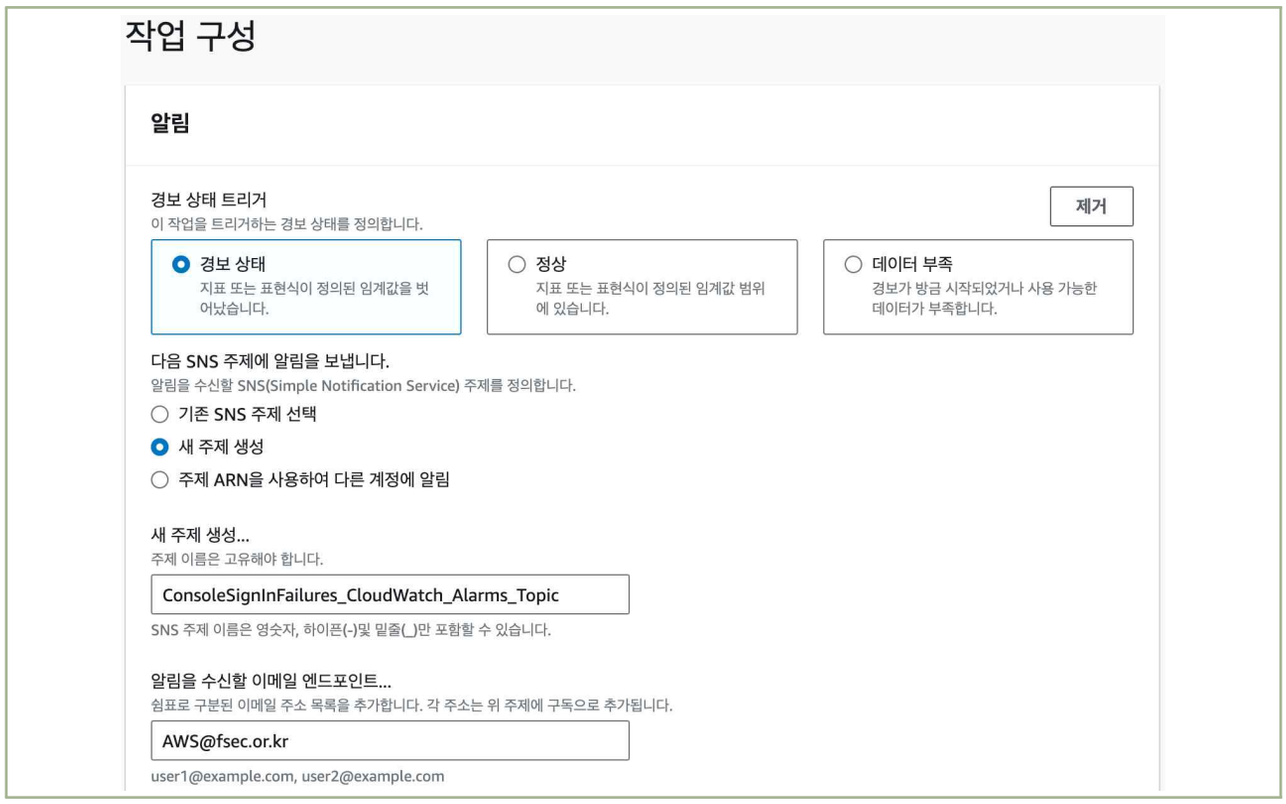
| 그림 5-2-7 | 지표 할당

- 지표 필터를 생성하면 CloudTrail 추적 로그 그룹에 대한 CloudWatch Logs 로그 그룹 세부 정보 페이지가 열리며, 아래 절차에 따라 경보 생성이 가능합니다.
  - ■ 지표 필터 탭에서 위에서 생성한 지표 필터를 선택 → '경보 생성'
  - ■ 지표 및 조건 지정을 원하는 수치로 조정하여 입력. 이 예시에서는 로그인 실패가 3회 이상 발생시 경보가 발생하도록 임계값을 3으로 정하고 나머지는 기본값으로 설정



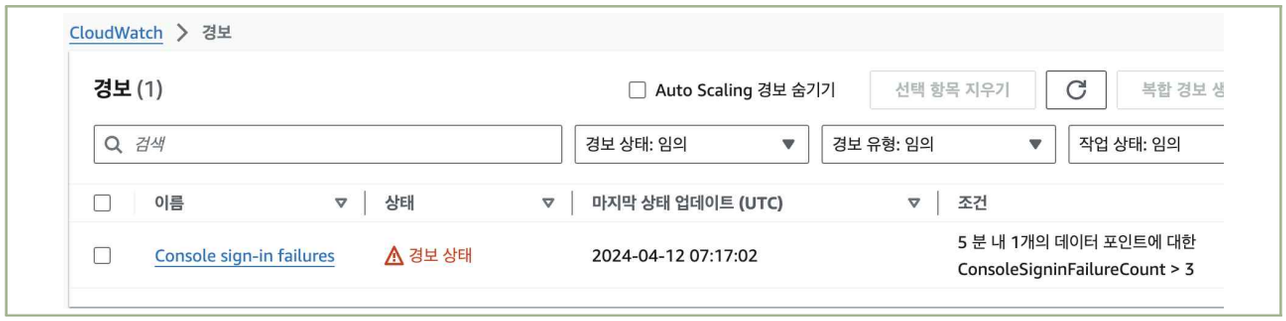
| 그림 5-2-8 | 지표 및 조건 지정

- 다음으로 Amazon SNS 서비스를 이용하여 SNS 서비스 Topic으로 알람을 보냅니다. (Amazon SNS 서비스는 알람을 발송할 수 있는 서비스)
- 알림 화면에서 '새 주제 생성' 선택하고 Topic(주제) 명을 설정합니다. 그 다음 수신할 이메일 주소를 입력하여 알람을 구성할 수 있습니다.

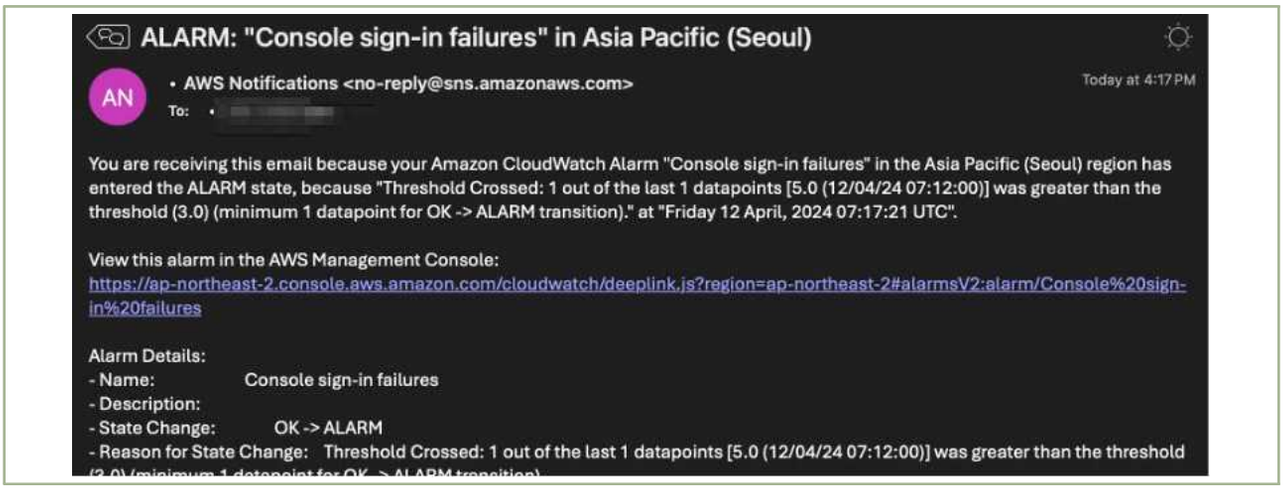


| 그림 5-2-9 | Amazon SNS로 알람 설정

- 그리고 경보 이름도 지정하여야 하며, Console sign-in failures와 같이 원하는 이름을 입력하여 경보를 생성합니다.
- 이후 변경사항에 대해 알람을 수신할 수 있으며 보안 그룹 변경과 같은 알람도 확인할 수 있습니다.



| 그림 5-2-10 | 경보 등록 확인



| 그림 5-2-11 | 등록한 이메일로 알람 수신

4 참고 사항

- CloudTrail이 로그 파일을 전송한 후 해당 파일이 수정, 삭제 또는 변경되지 않았는지 확인하기 위해 CloudTrail 로그 파일 무결성 검증을 사용할 수 있습니다. 관련 내용은 다음의 링크를 참고 바랍니다.
  - https://docs.aws.amazon.com/ko_kr/awscloudtrail/latest/userguide/cloudtrail-log-file-validation-intro.html

1 기준

식별번호	기준	내용
5.3.	**이용자 가상자원 모니터링 기능 확보**	**이용자 가상자원 운용에 관한 모니터링 기능을 확보하여야 한다.**

2 설명

○ 이용자 가상자원 가용성 확보 및 장애 대응을 위한 모니터링 기능을 확보하여야 한다.

- 예시

1) 가상자원 상태 모니터링(사용량, 트래픽 용량 등)
2) 가상자원 장애 모니터링(장애 발생 시 담당자 공지 등)
3) 가상자원 장애 발생 시 장애상황기록부 작성 등
4) 가상자원 네트워크 정책 변경(삭제 등) 모니터링

3 우수 사례

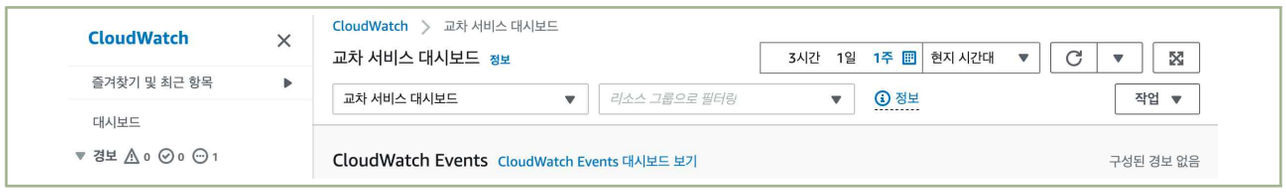
1) 가상자원 상태 모니터링(사용량, 트래픽 용량 등)

- CloudWatch는 AWS 리소스 전반의 데이터를 수집하여 전체 시스템의 성능을 파악할 수 있도록 하고, 사용자가 경보를 설정하고, 변화에 자동으로 대응하고, 운영 상태에 대한 통합된 뷰를 볼 수 있도록 해주는 서비스 입니다.
- CloudWatch의 교차 서비스를 이용하여 사용중인 모든 AWS 서비스의 대시보드와 상호 작용을 할 수 있습니다.

(1) 사용중인 모든 AWS 대시보드 확인

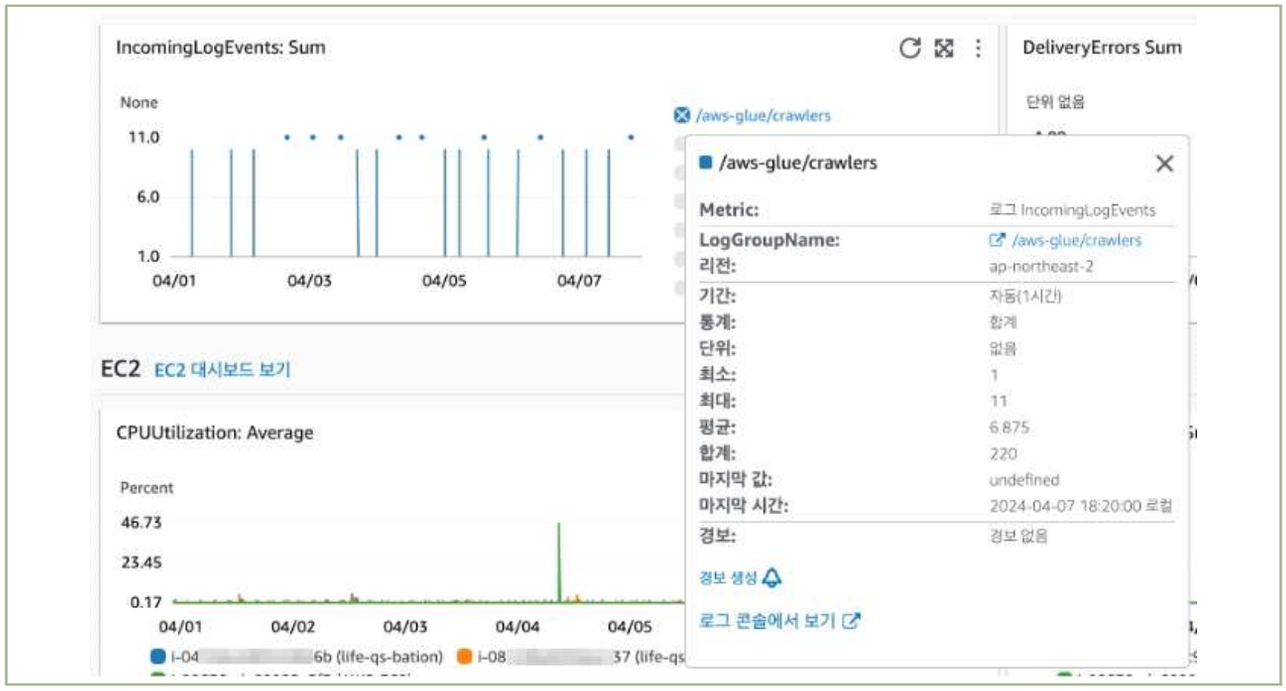
○ **(AWS 관리 콘솔)**

- '콘솔 홈' → 'CloudWatch' → Overview 화면에서 '개요' 드롭다운 → '교차 서비스 대시보드' 를 선택하여 이용할 수 있습니다.



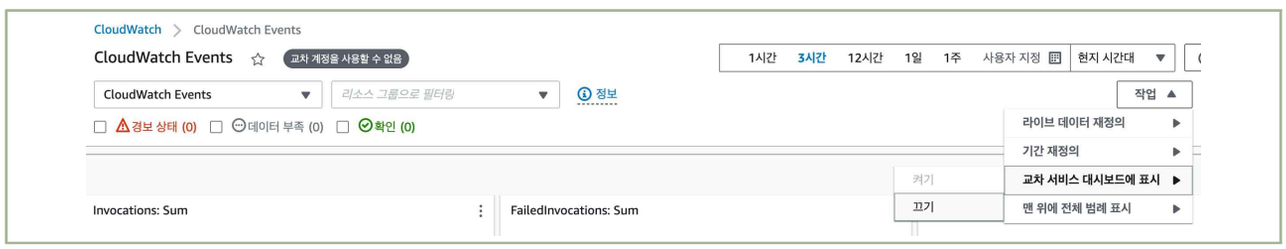
| 그림 5-3-1 | CloudWatch 교차 서비스 대시보드

- 드릴다운 하고 싶은 지표나 리소스를 클릭하여 집중적으로 탐색할 수 있습니다.



| 그림 5-3-2 | 리소스 매트릭 탐색

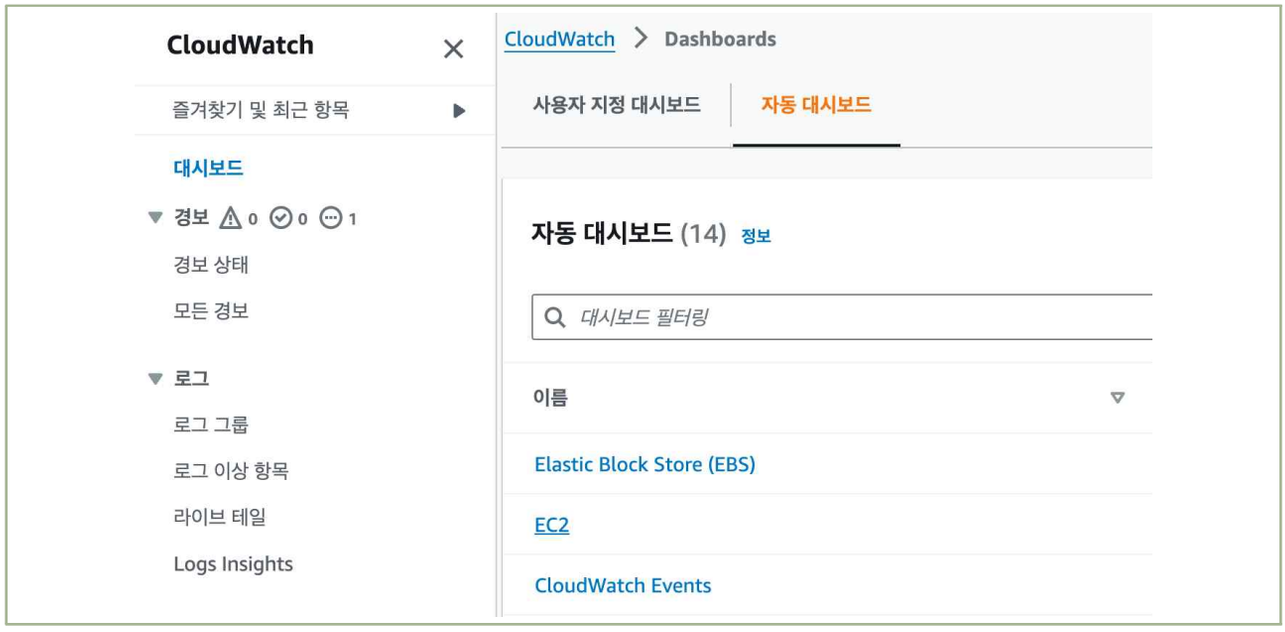
- 서비스의 지표를 교차 서비스 대시보드에서 제거하려면 '홈' → 'CloudWatch' → Overview에서 제거하려는 서비스를 선택할 수 있고 이때 해당 서비스에 대한 지표만 표시하도록 보기가 변경됩니다.



| 그림 5-3-3 | 교차 서비스 대시보드 Disable

(2) 단일 서비스 대시보드 확인

- **(AWS 관리 콘솔)** CloudWatch를 이용하여 단일 AWS 서비스를 집중적으로 확인할 수 있으며, '콘솔 홈' → 'CloudWatch' → '대시보드' → '자동 대시보드' → 확인하고자 하는 서비스를 선택합니다.



| 그림 5-3-4 | CloudWatch 자동 대시보드

- 임의의 그래프에서 지표를 보다 자세히 살펴보려면 그래프 위의 '작업 아이콘' → '지표에서 보기'를 선택합니다.

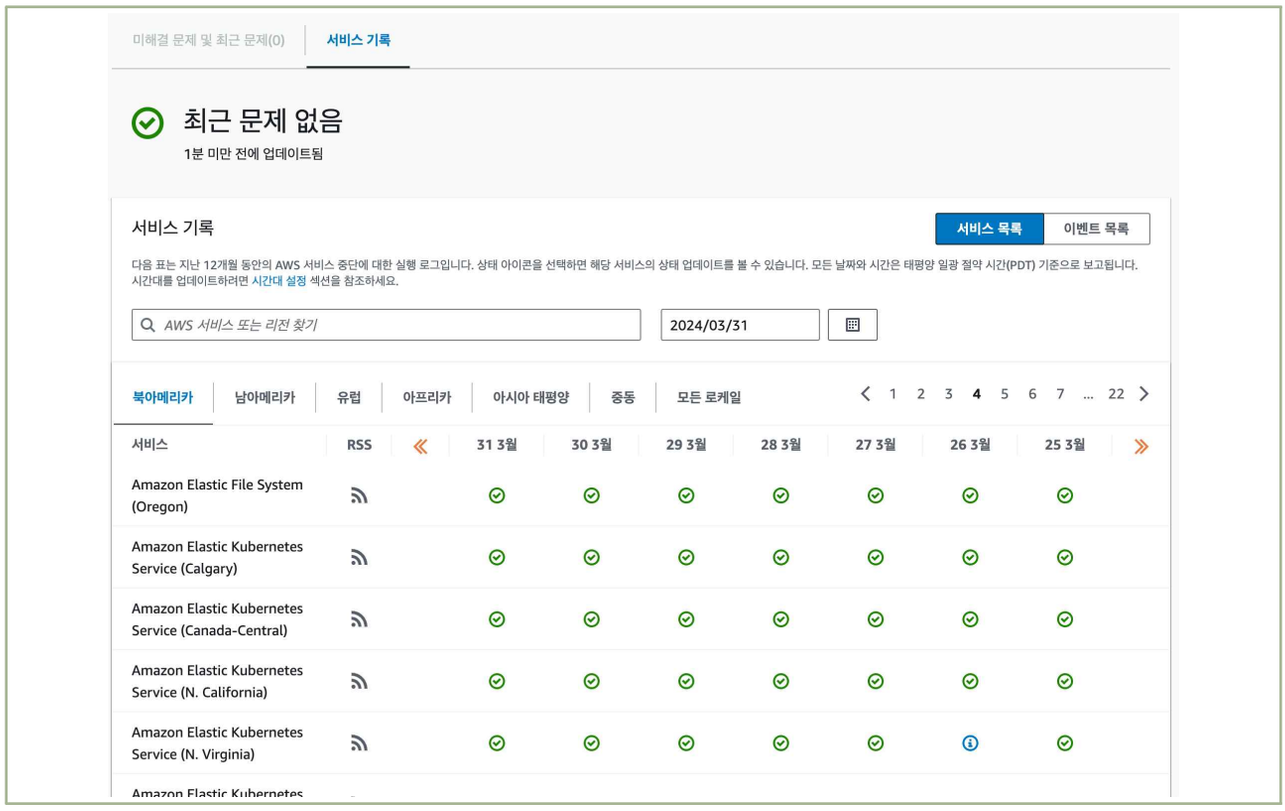
| 그림 5-3-5 | 상세 지표 확인

2) 가상자원 장애 모니터링(장애 발생 시 담당자 공지 등)

- AWS 환경에서 장애는 대규모 AWS 서비스에 대한 장애(리전, 서비스)와 개별 가상자원에 대한 장애로 분류할 수 있습니다.

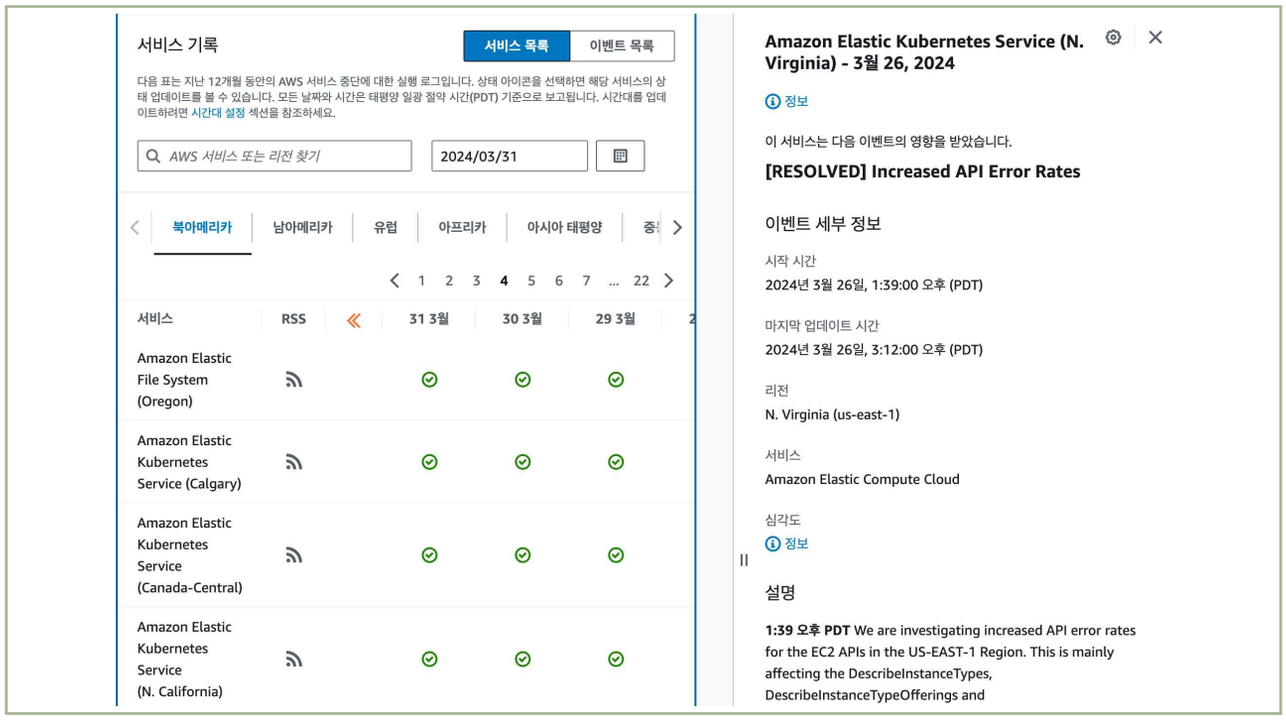
(1) 대규모 AWS 서비스 장애

- AWS Health Dashboard 서비스는 해당 AWS 리전 전반의 서비스에 대해 보고된 서비스 이벤트 표기합니다. 해당 내용은 AWS에 로그인하지 않아도 사용할 수 있습니다.
- https://health.aws.amazon.com/health/status 에 접속하여 '서비스 상태' → '미해결 문제 및 최근 문제' 에서 다음과 같은 내용을 확인할 수 있습니다.
  - 이벤트 이름 및 영향을 받는 리전. 예: 운영 문제 – Amazon Elastic Compute Cloud (버지니아 북부)
  - 서비스 이름
  - 이벤트의 심각도 (예: 정보 또는 성능 저하)
  - 이벤트의 최근 업데이트 타임라인
  - AWS 서비스 해당 목록도 이 이벤트의 영향을 받음
- 또한 '서비스 상태' → '서비스 기록'에서 지난 12개월 동안의 모든 AWS 서비스 중단 여부를 확인할 수 있습니다.



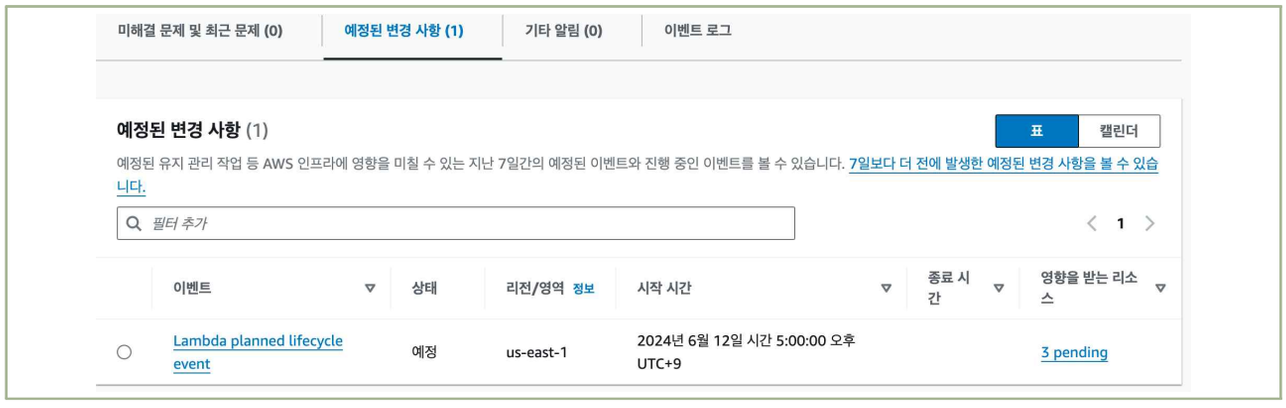
| 그림 5-3-6 | AWS Health 대시보드

- RSS아이콘을 선택하면 지정한 AWS 리전에서 특정 서비스에 대한 알림을 RSS 피드 구독 형태로 수신할 수 있고 아이콘을 선택하면 이벤트에 대한 자세한 내용 확인할 수 있습니다.



| 그림 5-3-7 | 서비스 기록 확인

- AWS 계정이 있는 경우 계정 상태 보기의 '계정 상태 열기'를 선택하여 로그인하면 계정과 관련된 이벤트 확인이 가능합니다.



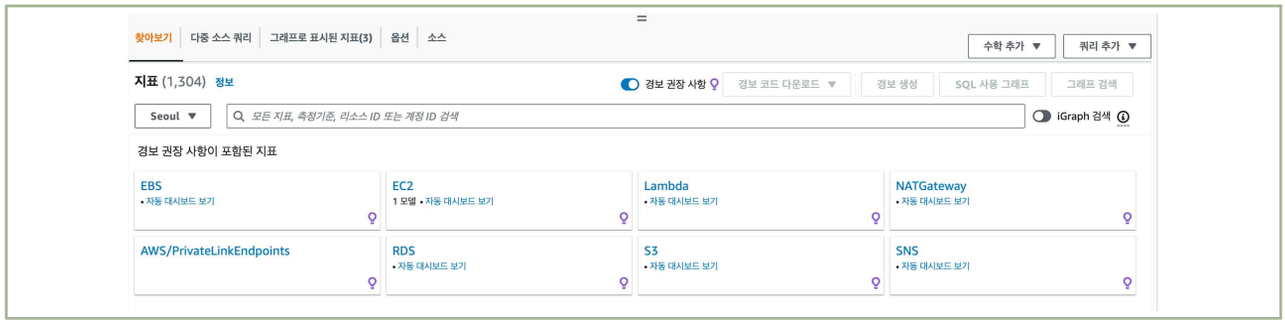
| 그림 5-3-8 | 계정 관련 이벤트 확인

(2) 개별 AWS 서비스 장애

- AWS CloudWatch 경보를 사용하여 개별 AWS 서비스 장애에 대해 알람을 수신할 수 있습니다.

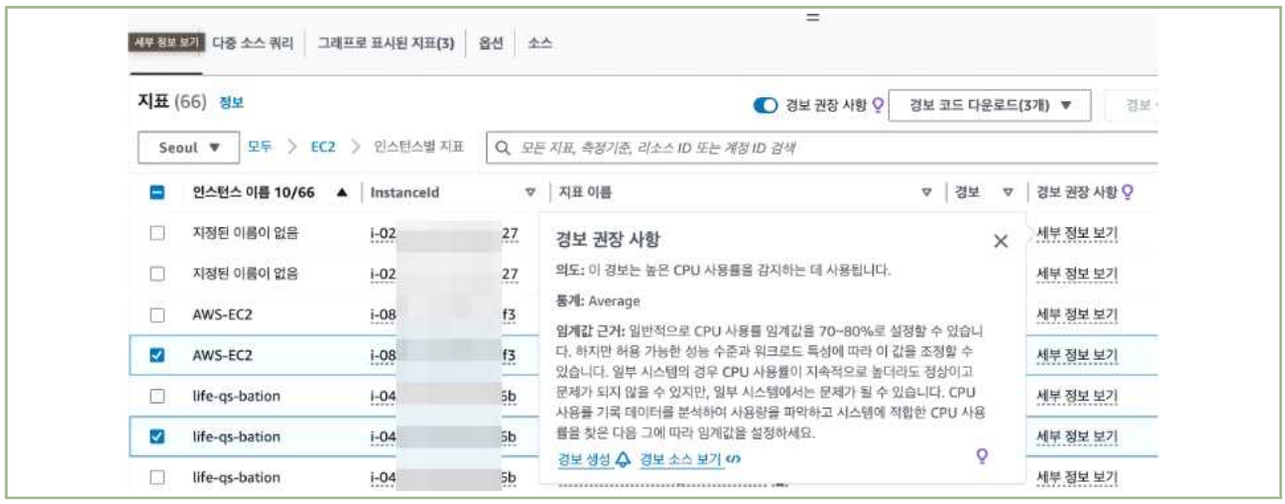
- 별도 기준에 따른 알람도 가능하며, AWS에서 권장하는 경보도 제공합니다. AWS 권장 사항을 따르면 AWS 인프라에 대한 중요한 모니터링을 놓치지 않을 수 있습니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → 'CloudWatch' → '지표' → '모든 지표' → '찾아보기' → '경보 권장 사항' 선택하여 확인할 수 있습니다.



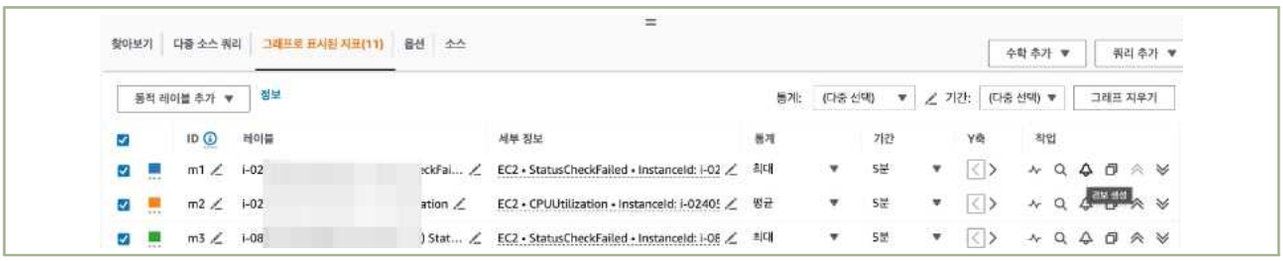
| 그림 5-3-9 | CloudWatch 지표 화면

- 서비스의 네임스페이스 선택 → '세부 정보 보기' 를 통하여 지표에 대한 경보 의도와 권장 임계치 값을 확인할 수 있습니다.



| 그림 5-3-10 | 경보 권장사항 임계치 확인

- 지표 중 하나에 대한 경보를 생성하기 위해서는 '그래프로 표시된 지표' → '경보 생성' 으로 생성할 수 있습니다.

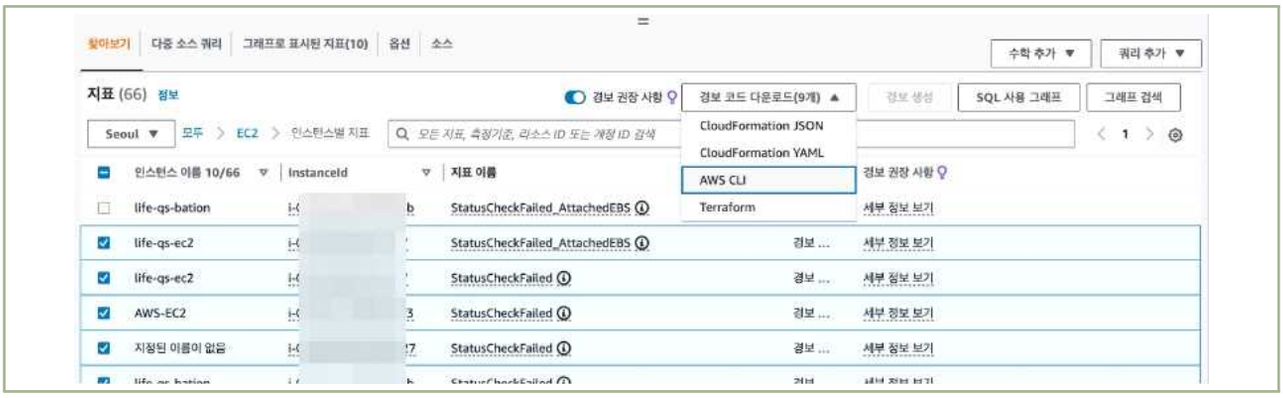


| 그림 5-3-11 | 경보 생성

- 원하는 값으로 변경 한 뒤 Amazon SNS 서비스의 Topic을 생성하여 알림을 설정하면 경보를 생성할 수 있습니다.

| 그림 5-3-12 | Amazon SNS 서비스 이용 알림 생성

- AWS 서비스의 모든 권장 경보에 대한 코드형 인프라 경보 정의를 다운로드하여 일괄 적용도 가능합니다. '홈' → 'CloudWatch' → '지표' → '모든 지표' → '찾아보기' → '경보 권장 사항' → 적용을 원하는 지표 선택 → '경보 코드 다운로드(n개)' → 원하는 방식으로 다운로드 할 수 있습니다.



| 그림 5-3-13 | 경보 코드 다운로드 화면

```
aws cloudwatch put-metric-alarm \
--alarm-name 'AWS/EC2 CPUUtilization InstanceId=i-02[redacted]27' \
--alarm-description '이 경보는 EC2 인스턴스의 CPU 사용률을 모니터링하는 데 도움이 됩니다.
--no-actions-enabled \
--metric-name 'CPUUtilization' \
--namespace 'AWS/EC2' \
--statistic 'Average' \
--dimensions '[{"Name":"InstanceId","Value":"i-02[redacted]27"}]' \
--period 300 \
--evaluation-periods 3 \
--datapoints-to-alarm 3 \
--threshold 80 \
--comparison-operator 'GreaterThanThreshold' \
--treat-missing-data 'missing'

aws cloudwatch put-metric-alarm \
--alarm-name 'AWS/EC2 CPUUtilization InstanceId=i-04[redacted]6b' \
--alarm-description '이 경보는 EC2 인스턴스의 CPU 사용률을 모니터링하는 데 도움이 됩니다.
--no-actions-enabled \
--metric-name 'CPUUtilization' \
--namespace 'AWS/EC2' \
--statistic 'Average' \
--dimensions '[{"Name":"InstanceId","Value":"i-04[redacted]6b"}]' \
--period 300 \
--evaluation-periods 3 \
--datapoints-to-alarm 3 \
--threshold 80 \
--comparison-operator 'GreaterThanThreshold' \
```

| 그림 5-3-14 | AWS CLI 경보 코드 예시

4) 가상자원 네트워크 정책 변경(삭제 등) 모니터링

(1) CloudWatch 경보 생성

- 클라우드 인프라 환경 구성을 변경하였을 때 경보를 발생시켜 알람을 받을 수 있으며, 보안 그룹과 같은 네트워크 변경에 대한 모니터링 및 알람도 가능합니다.

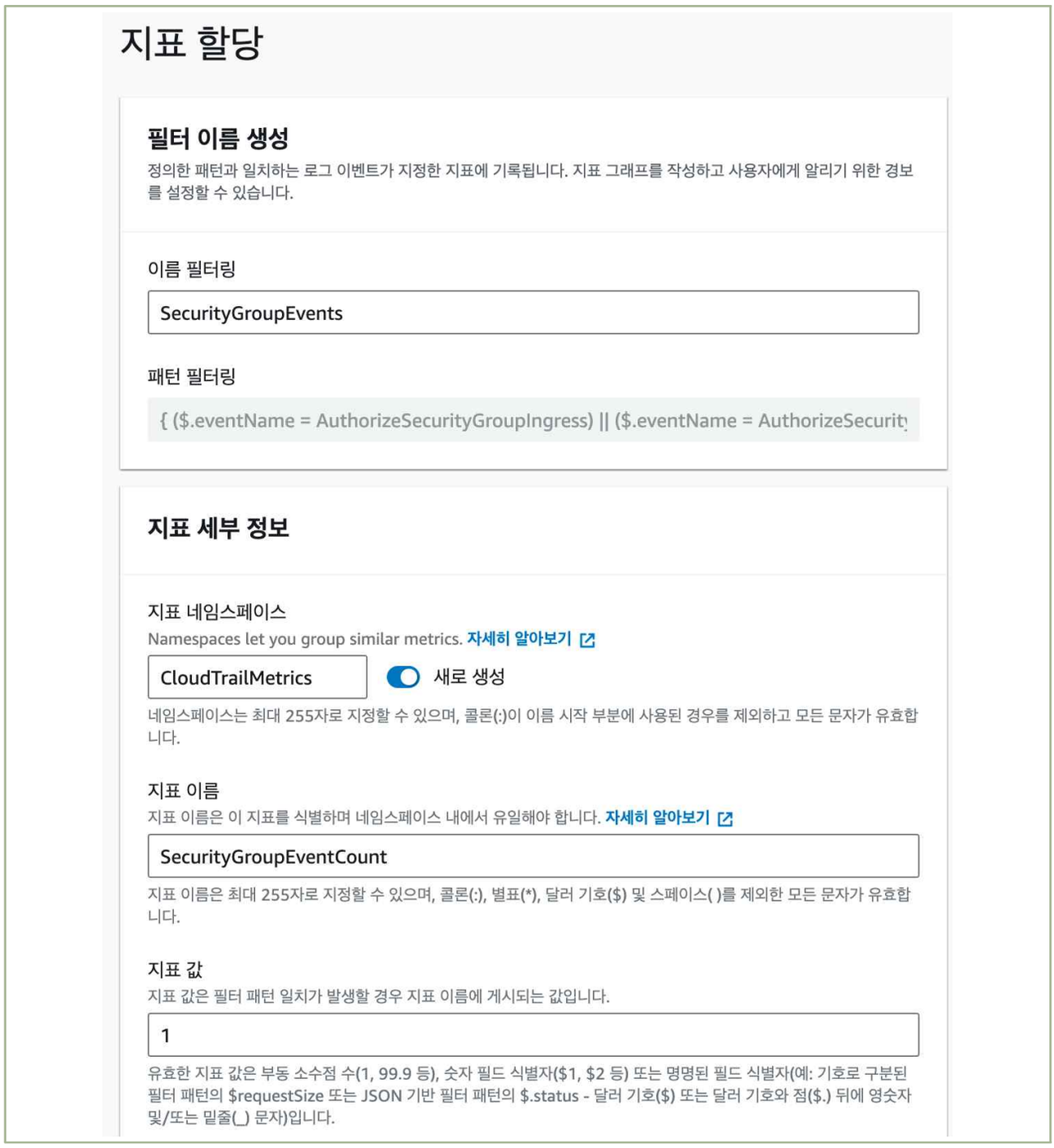
- **(AWS 관리 콘솔)** '콘솔 홈' → 'CloudWatch' → '로그' → '로그 그룹' → 추적에 대해 생성한 로그 그룹 선택 → '지표 필터' → '지표 필터 생성' 을 선택하고, 필터 패턴 생성의 패턴 필터링에 아래와 같이 입력하여 경보에 사용할 필터 패턴을 생성합니다.

```
{ ($.eventName = AuthorizeSecurityGroupIngress) || ($.eventName =
AuthorizeSecurityGroupEgress) || ($.eventName = RevokeSecurityGroupIngress) ||
($.eventName = RevokeSecurityGroupEgress) || ($.eventName = CreateSecurityGroup) ||
($.eventName = DeleteSecurityGroup) }
```

| 그림 5-3-15 | 패턴 필터링 예

| 그림 5-3-16 | 필터 패턴 정의

- 이름 필터링에 SecurityGroupEvents 입력
- 지표 네임스페이스에 CloudTrailMetrics 입력
- 지표 이름에 SecurityGroupEventCount 입력
- 지표 값에 1입력 → '다음' → '지표 필터 생성'

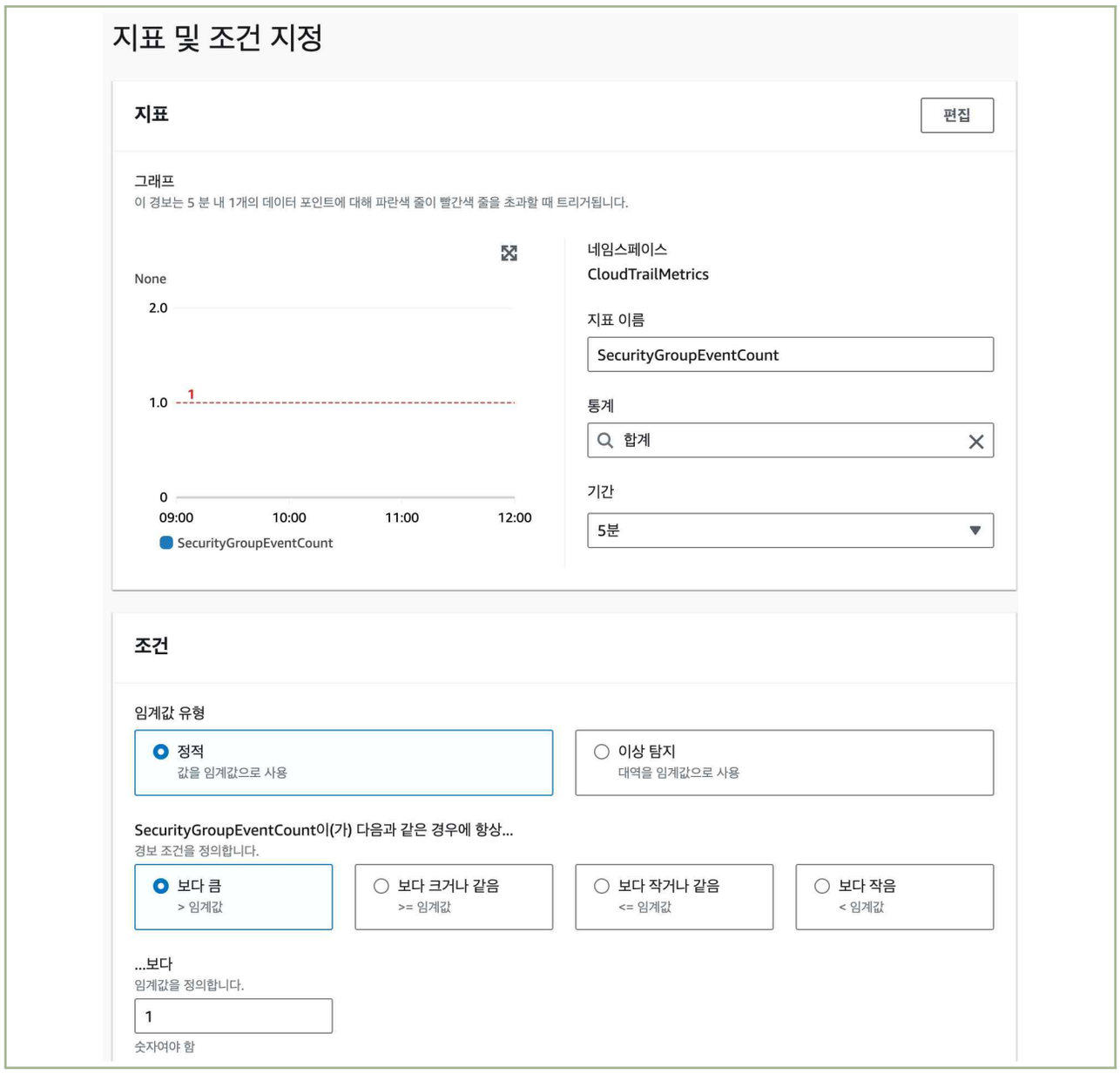


| 그림 5-3-17 | 지표 할당

- 지표 필터를 생성하면 CloudTrail 추적 로그 그룹에 대한 CloudWatch Logs 로그 그룹 세부 정보 페이지가 열리는데, 다음과 같은 절차에 따라 경보 생성이 가능합니다.

  ■ 지표 필터 탭에서 위에서 생성한 지표 필터를 선택 → '경보 생성'

  ■ 지표 및 조건 지정을 원하는 수치로 조정하여 입력. 이 예시에서는 보안 그룹을 1회라도 수정하면 경보가 발생하도록 임계값을 1로 정하고 나머지는 기본값으로 설정



| 그림 5-3-18 | 지표 및 조건 지정

- 다음으로 Amazon SNS 서비스를 이용하여 알림을 보내는 설정을 합니다. 해당 부분은 위의 2) 및 3) 에서 설명한 Amazon SNS 내용을 참조 바랍니다.

4 참고 사항

- 설정을 권장하는 경보는 다음 링크의 내용을 참조 바랍니다.
  - https://docs.aws.amazon.com/ko_kr/AmazonCloudWatch/latest/monitoring/Best_Practice_Recommended_Alarms_AWS_Services.html

1 기준

식별번호	기준	내용
5.4.	API 사용(호출대상, 호출자, 호출일시 등)에 관한 행위추적성 확보	API 사용 이력에 대한 행위추적성(로그 등)을 확보하여야 한다.

2 설명

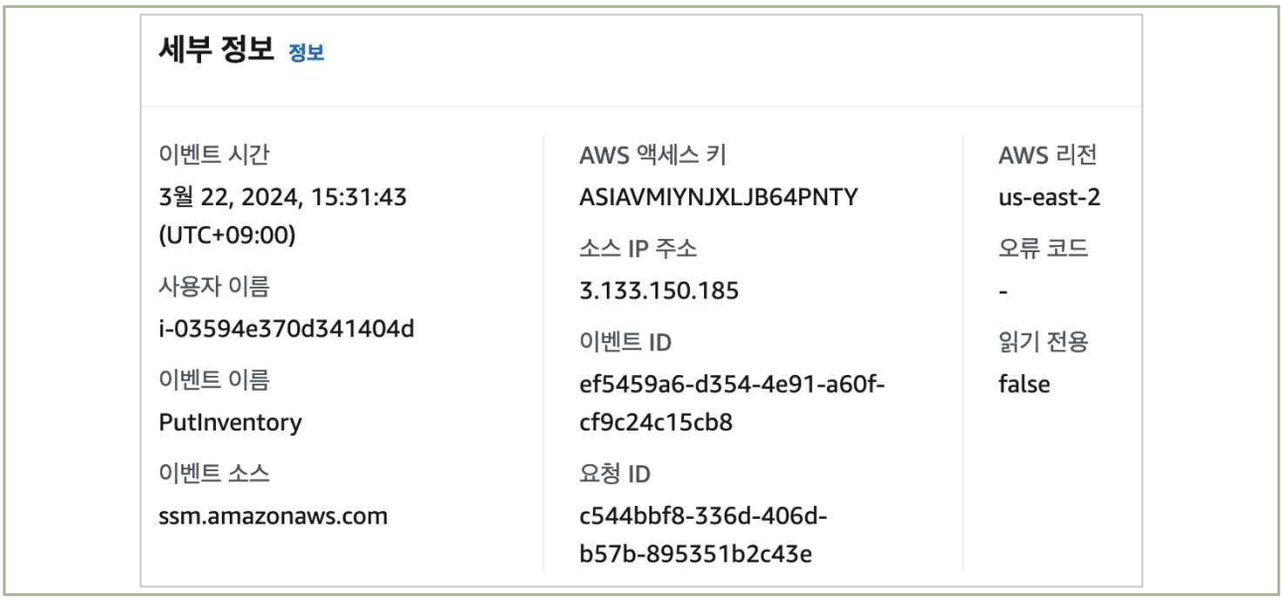
- API 사용 이력에 대한 행위추적성을 확보하여야 한다.

  – 예시

  1) API 호출에 관한 정보(호출대상, 호출자, 호출일시 등)

3 우수 사례

- **(AWS 관리 콘솔)** 사용자 활동 및 API 사용에 대한 추적은 '콘솔 홈' → 'CloudTrail' → '이벤트 기록' –〉 확인하고자 하는 API 이벤트 이름을 선택하여 세부 정보를 확인할 수 있습니다.



| 그림 5-4-1 | API 이벤트에 대한 세부 정보 확인

– 세부 정보에서는 다양한 정보 확인이 가능하며, 호출대상은 '이벤트 이름', 호출자는 '이벤트 소스' 그리고 호출일시는 '이벤트 시간' 으로 판단이 가능합니다.

API 사용 이력	관련된 필드	샘플 예	설명
호출대상	- 이벤트 이름 - 사용자 이름	PutInventory i-03594e370d341404d	호출의 대상이 된 이벤트 이름과 실제 대상인 사용자 이름 확인 가능 (예제 에서는 EC2 인스턴스 ID 가 해당)
호출자	- 이벤트 소스 - 소스 IP 주소	ssm.amazonaws.com 3.133.150.185	호출한 소스 IP 주소와 해당 서비스를 호출한 이벤트 소스 확인 가능
호출일시	- 이벤트 시간	3월 22, 2024, 15:31:43 (UTC+09:00)	이벤트가 호출된 시간을 기록

- 이벤트 레코드는 이벤트의 전체정보를 JSON 오브젝트로 제공합니다. 해당 레코드에는 요청한 시간과 위치 뿐만 아니라 API 요청한 작업을 파악하는 데 도움이 되는 필드 정보를 포함하고 있습니다.

| 그림 5-4-2 | API 이벤트 레코드 JSON 오브젝트 보기

- **(AWS CLI)** API 환경으로 이벤트 필드 조회는 'aws cloudtrail lookup-events' 명령을 사용하여 지난 90일 동안의 CloudTrail 관리 이벤트에서 현재 이벤트를 조회할 수 있습니다. 본 참고서는 EventName 을 ConsoleLogin 으로 필터링 하여 조회하는 것을 설명합니다.

```
aws cloudtrail lookup-events --lookup-attributes \
AttributeKey=EventName,AttributeValue=ConsoleLogin
```

Output:


```
{
  "Events": [
    {
        "EventId": "654ccbc0-ba0d-486a-9076-dbf7274677a7",
        "Username": "my-session-name",
        "EventTime": "2021-11-18T09:41:02-08:00",
        "CloudTrailEvent":
"{₩"eventVersion₩":₩"1.02₩",₩"userIdentity₩":{₩"type₩":₩"AssumedRole₩",₩"principalId₩":₩"A
ROAJIKPFTA72SWU4L7T4:my-session-name₩",₩"arn₩":₩"arn:aws:sts::123456789012:assumed-
role/my-role/my-session-
name₩",₩"accountId₩":₩"123456789012₩",₩"sessionContext₩":{₩"attributes₩":{₩"mfaAuthentic
ated₩":₩"false₩",₩"creationDate₩":₩"2016-01-
26T21:42:12Z₩"},₩"sessionIssuer₩":{₩"type₩":₩"Role₩",₩"principalId₩":₩"AROAJIKPFTA72SWU4
L7T4₩",₩"arn₩":₩"arn:aws:iam::123456789012:role/
…(생략)…
        "EventName": "ConsoleLogin",
        "Resources": []
    }
  ]
}
```


| 그림 5-4-3 | CLI 명령 - EventNane 필드 조회

- Lookup-events 명령에는 다음과 같은 옵션을 사용할 수 있습니다.
  - --max-items 〈정수값〉
    - 반환할 이벤트 수를 지정할 수 있으며, 1에서 최대 50까지 정의 가능합니다.
      예) aws cloudtrail lookup-events -max-items 10
  - --start-time 〈타임스탬프〉
    - 지난 90일간의 이벤트에 대해 시간 범위를 지정할 수 있으며, 모든 타임스탬프는 UTC 로 표시됩니다.
  - --lookup-attributes 〈정수값〉
    - 속성 키/값 페어를 정의할 수 있으며, 값 이름은 대/소문자를 구분합니다.

- 관리 이벤트 조회 시 Attributes 에서 키 값으로 사용할 수 있는 속성은 다음과 같습니다.
  - AWS 액세스 키
  - 이벤트 ID

- 이벤트 이름
- 이벤트 소스
- 읽기 전용
- 리소스 이름
- 리소스 유형
- 사용자 이름

- 출력되는 필드에 대한 설명은 아래와 같습니다.

필드	설명
Event	조회 속성과 지정된 시간 범위를 기반으로 한 조회 이벤트 목록
CloudTrailEvent	이벤트가 반환되었음을 나타내는 객체를 포함한 JSON 문자열
EventId	반환된 이벤트 GUID를 포함한 문자열
EventName	반환된 이벤트 이름을 포함한 문자열
EventSource	요청이 이루어진 AWS 서비스
EventTime	이벤트 날짜 및 시간(UNIX 시간 형식)
Resources	반환된 이벤트가 참조하는 리소스 목록
ResourceName	이벤트가 참조하는 리소스 이름을 포함하는 문자열
ResourceType	이벤트가 참조하는 리소스 유형을 포함하는 문자열
Username	반환된 이벤트에 대한 계정 사용자 이름을 포함하는 문자열
NextToken	이전 lookup-events 명령에서 결과의 다음 페이지를 가져오는 문자열

4 참고 사항

- 이용자가 별도로 SIEM 솔루션을 사용하는 경우, SIEM 솔루션으로 사용자 활동 및 API 사용 추적 관련 로그를 저장할 수 있으며, 저장 방법은 사용하는 SIEM 솔루션 제조사로 문의 바랍니다.
- 사용자 활동 및 API 사용 추적 로그를 사용하기 위해서는 CloudTrail이 구성되어 있어야 합니다.
- AWS 명령줄 인터페이스 사용에 대한 일반적인 정보는 AWS 에서 제공하는 'AWS Command Line Interface 사용 설명서' 를 참고 바랍니다.
- CloudTrail AWS CLI 명령은 대소문자를 구분합니다.
- CloudTrail의 기본 로그 보관 기간은 90 일 이므로, 장기 보관이 필요한 경우, CloudTrail Lake, CloudWatch 및 S3 에 보관을 검토할 수 있습니다.
- S3, Lambda 에서 사용되는 API 호출은 '데이터 이벤트' 로그를 남기도록 구성하여야 합니다.
- 이벤트 조회 요청 속도는 계정당, 리전당 초당 2회로 제한되며, 이 한도를 초과하면 제한 오류가 발생합니다.

1 기준

식별번호	기준	내용
5.5.	**네트워크 관련 서비스(VPC, 보안 그룹, ACL 등)에 관한 행위추적성 확보**	**이용자의 클라우드 네트워크 서비스 이용 시 발생하는 사항에 대한 행위추적성(로그 등)을 확보하여야 한다.**

2 설명

- 클라우드 환경에서 네트워크 서비스(VPC, NAT 등) 사용 시 발생하는 사항에 대한 행위추적성(로그 등)을 확보하여야 한다.

  - 행위 감사로그

    1) 네트워크 서비스 이용에 관한 사항(VPC, NAT 규칙 생성 및 변경 등) 등

3 우수 사례

(1) VPC 네트워크 행위 추적성

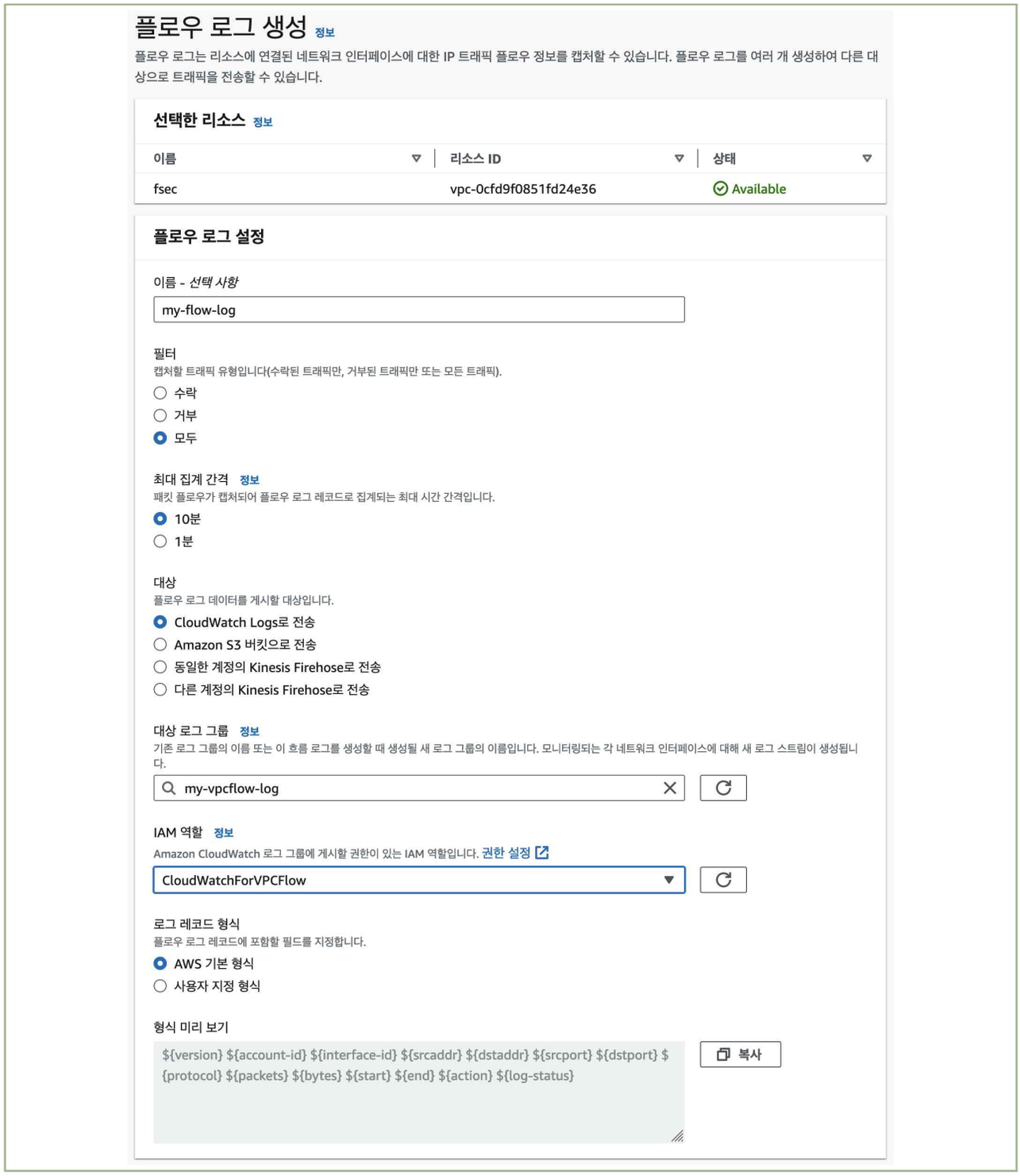
- VPC 안에서 네트워크 흐름에 대한 세부 정보를 로그로 기록할 수 있으며 VPC, 서브넷 그리고 네트워크 인터페이스 단위로 플로우 로그 생성이 가능합니다. 로그 기록이 필요한 곳에 해당 단위별로 플로우 로그 생성을 설정합니다. 다음은 VPC 플로우 로그에서 사용되는 각 필드에 대한 설명을 포함한 도표입니다.

필드	설명
버전	플로우 로그 버전 정보 (예) 2,3,4,5
어카운트 ID	AWS 어카운트 ID
인터페이스 ID	ENI 인터페이스 ID
출발지 주소	출발지 주소
목적지 주소	목적지 주소
출발지 포트	출발지 포트 (예) 36490
목적지 포트	목적지 포트 (예) 443
프로토콜 번호	IANA 에 정의된 프로토콜 번호 (예) TCP,6
패킷 개수	전송된 패킷 개수
전송량 (bytes)	트래픽 전송량으로 바이트(bytes) 표현
첫번째 패킷 시간	타임스탬프로 시간 기록 (예) 1560385064
마지막 패킷 시간	타임스탬프로 시간 기록 (예) 1560385064
플로우 액션	NACL, 보안 그룹의 허용 차단 유무
로그 상태	OK, NODATA 와 같은 상태 정보

| 그림 5-5-1 | VPC Flow Logs 구성

(AWS 관리 콘솔)

- VPC Flow Logs는 VPC 전체를 대상으로 로그를 기록할 수 있으며 '콘솔 홈' → VPC → VPC → '플로우 로그 생성' 을 통하여 로그 필터, 전송 대상 등을 정의할 수 있습니다. 필터는 로그 기록을 허용된 트래픽 또는 거부된 트래픽으로 한정하여 기록하거나 모두 다 기록할 수 있습니다.

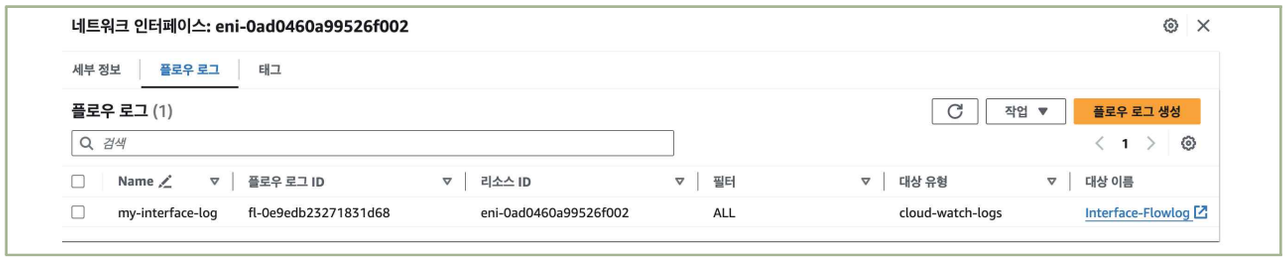


| 그림 5-5-2 | VPC Flow Logs 생성

- 서브넷으로 한정하여 플로우 로그 생성도 가능합니다.

| 그림 5-5-3 | 서브넷 Flow Logs 생성

- 네트워크 인터페이스 단위별 설정은 'EC2' -〉 '네트워크 및 보안' -〉 '네트워크 인터페이스' -〉 '플로우 로그' 에서 설정할 수 있습니다.



| 그림 5-5-4 | 네트워크 인터페이스 Flow Logs 생성

(2) VPC Flow Log를 CloudWatch로 전송

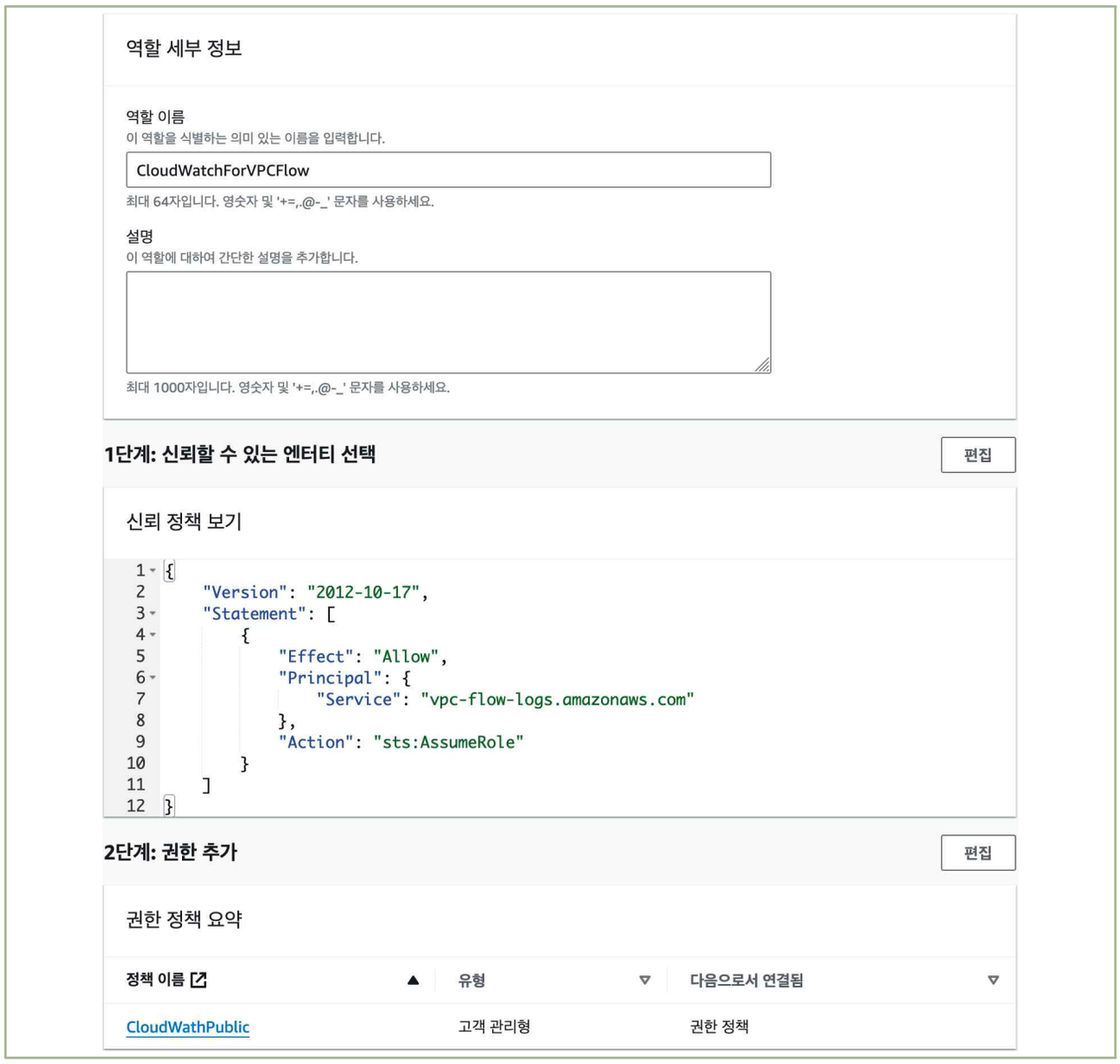
- VPC Flow Log 전송 대상은 3곳으로 설정이 가능하며, CloudWatch로 전송하는 경우 IAM 정책과 역할의 생성 그리고 CloudWatch의 로그 그룹 생성이 필요합니다.

◦ (AWS 관리 콘솔)

- IAM 정책 생성은 '콘솔 홈' → 'IAM' → '정책' → '정책 생성'에서 할 수 있으며, 아래와 같이 CloudWatch에 필요한 Action 정책을 생성합니다.

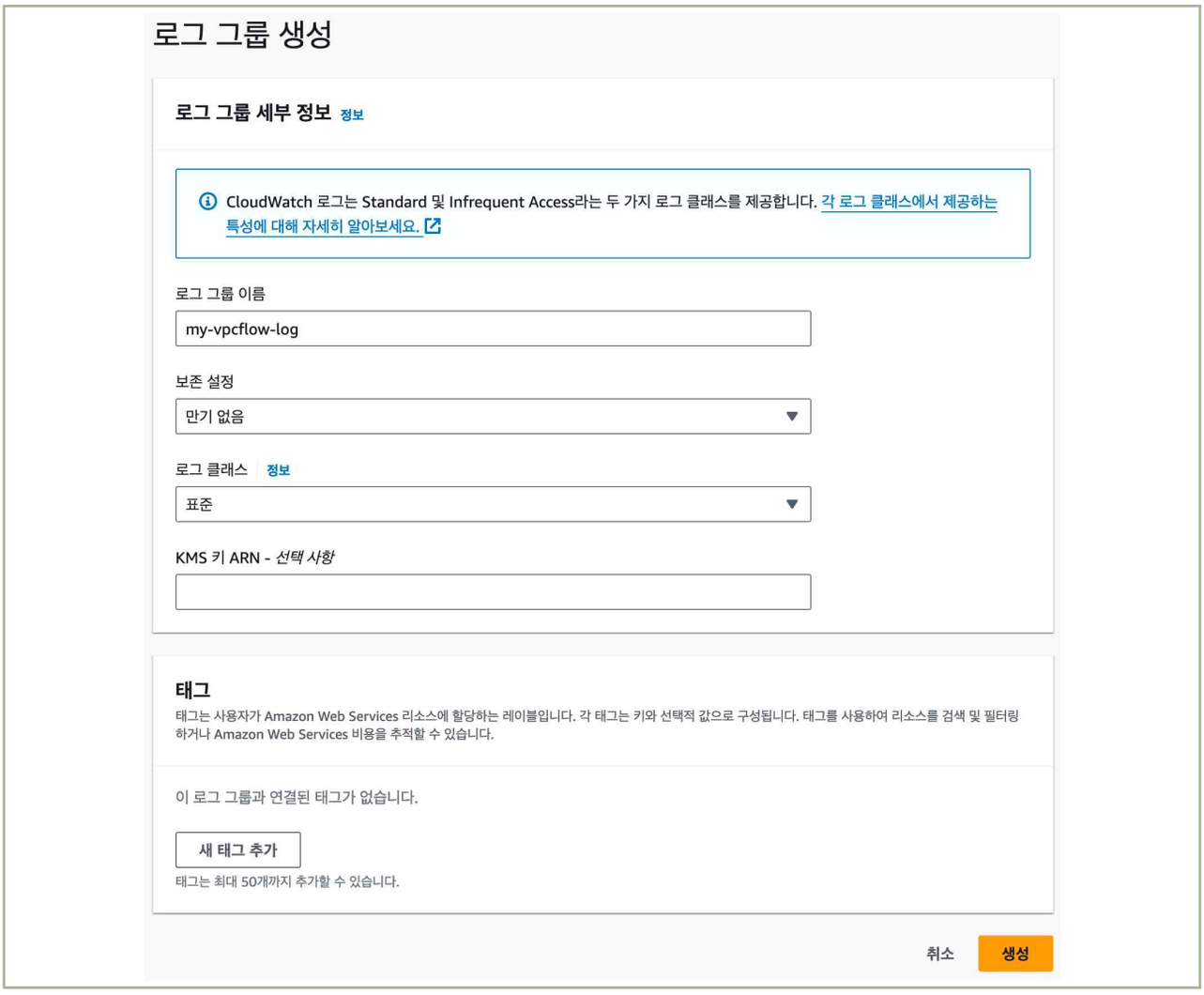
| 그림 5-5-5 | CloudWatch에 필요한 IAM 정책 생성

- IAM 역할 생성은 '콘솔 홈' → 'IAM' → '역할' → '역할 생성'에서 할 수 있으며, 아래와 같이 vpc-flow-logs.amazonaws.com 에 대한 신뢰 정책을 추가하고, 앞서 만들었던 정책 권한을 추가합니다.



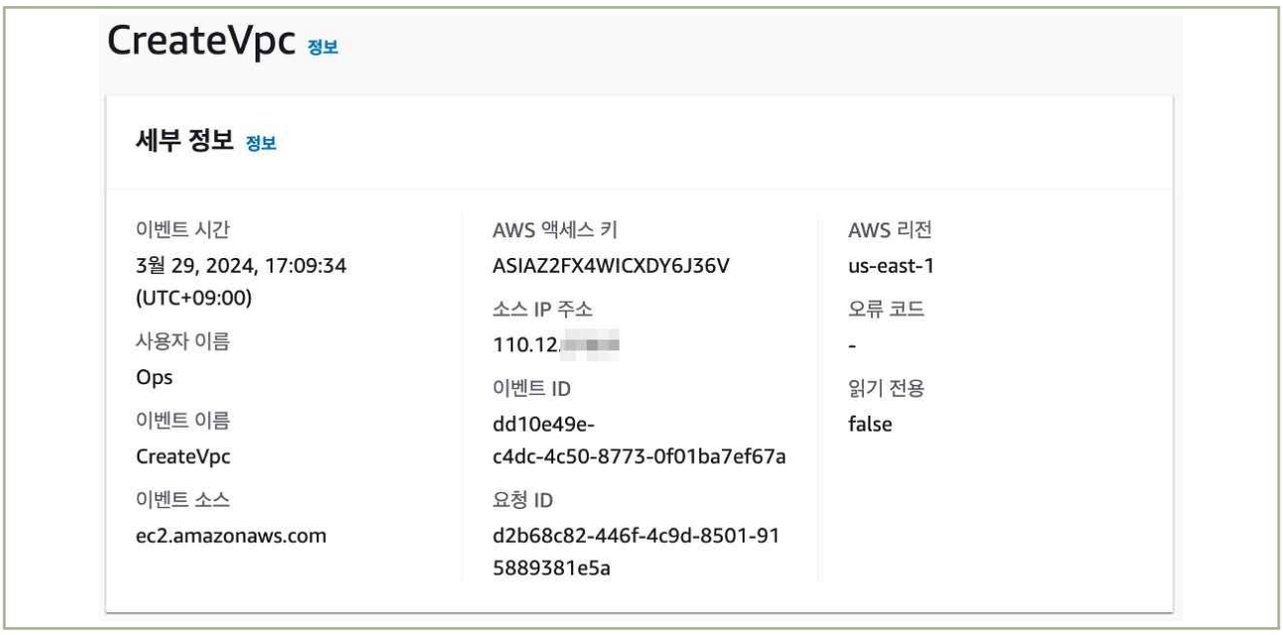
| 그림 5-5-6 | CloudWatch를 위한 IAM 역할 생성

- CloudWatch 로그 그룹 생성은 '콘솔 홈' → 'CloudWatch' → '로그 그룹' → '로그 그룹 생성' 을 통해 CloudWatch에 기록할 VPC Flow Log의 그룹을 생성할 수 있습니다.



| 그림 5-5-7 | CloudWatch 로그 그룹 생성

- VPC 행위 감사는 '콘솔 홈' → 'CloudTrail' → '이벤트 기록' → '관련 로그 검색' 및 해당 Action에 해당하는 이벤트 이름으로 검색하여 수행할 수 있습니다.
- 만약 VPC 를 신규로 생성하는 경우 'CreateVpc' 이벤트가 기록되며, 이벤트 상세 정보도 확인할 수 있습니다.



| 그림 5-5-8 | CloudTrail 의 CreateVpc 이벤트

- 기본적으로 이벤트의 발생 시간, 사용자 이름, 이벤트 이름 및 소스 등을 확인할 수 있습니다. 이벤트 레코드에는 '세부 정보' 에서 포함하지 않는 더욱 많은 정보를 제공하고 있으며, 이벤트 레코드의 주요 필드 설명은 다음 도표를 참고 바랍니다.

필드	설명
eventTime	요청이 완료된 날짜와 시간 (UTC 시간 표시)
userIdentity	요청한 사용자 정보
eventSource	요청이 이뤄진 서비스 (예) ec2.amazonaws.com
eventName	해당 서비스를 위한 API 작업명
eventRegion	요청이 이뤄진 리전 정보 (예) ap-northeast-2
sourceIPAddress	요청이 발생한 IP 주소
UserAgent	요청이 이뤄진 에이전트명 (예) console.amazonaws.com, aws-cli/1.3.23 Python/2.7.6 Linux/2.6.18-164.el5
requestParameters	파라미터가 있는 경우 요청과 함께 전송됨
responseElements	변경이 이뤄지는 작업의 응답 항목(작업생성, 업데이트,삭제등)
eventType	이벤트 레코드 유형 (예) AwsApiCall –호출된 API, AwsConsoleAction – API 호출이 아닌 콘솔에서 수행된 작업
readOnly	작업이 읽기 전용 작업인지 식별 (예) true 또는 false
resources	이벤트에서 엑세스되는 리소스 목록 (예) 리소스 ARN

| 그림 5-5-9 | 이벤트 레코드 주요 필드

- VPC 를 변경하는 경우 VPC 변경에 해당하는 이벤트가 기록됩니다. 예를 들어, VPC 의 CIDR 을 추가하는 경우에는 'AssociateVpcCidrBlock' 이벤트가 기록됩니다. 그리고 이벤트 레코드에서 세부적인 변경 내역을 확인할 수 있습니다.

```
        "eventTime": "2024-03-31T09:44:21Z",
        "eventSource": "ec2.amazonaws.com",
        "eventName": "AssociateVpcCidrBlock",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "110.12.[redacted]",
        "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:123.0) Gecko/
    20100101 Firefox/123.0",
        "requestParameters": {
            "AssociateVpcCidrBlockRequest": {
                "VpcId": "vpc-0cfd9f0851fd24e36",
                "CidrBlock": "10.11.0.0/16"
            }
```

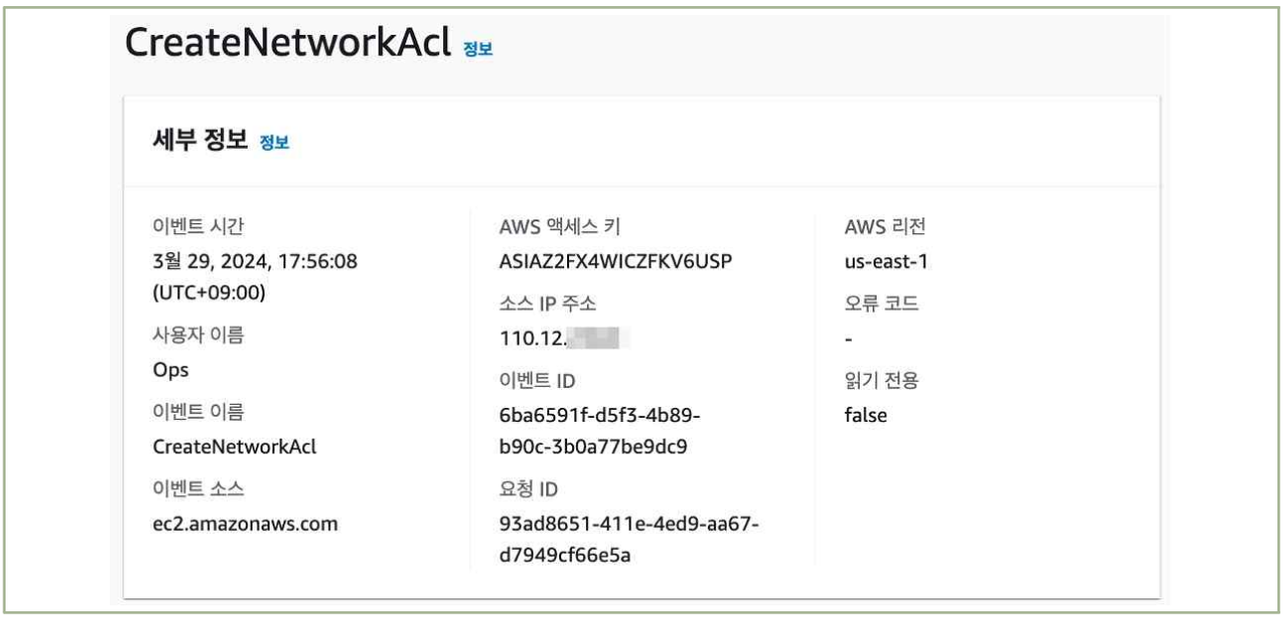

| 그림 5-5-10 | VPC 규칙에서 CIDR 블럭을 추가한 경우

- 만약 VPC 를 삭제하는 경우 'DeleteVpc' 이벤트가 기록됩니다.

| 그림 5-5-11 | CloudTrail의 DeleteVpc 이벤트

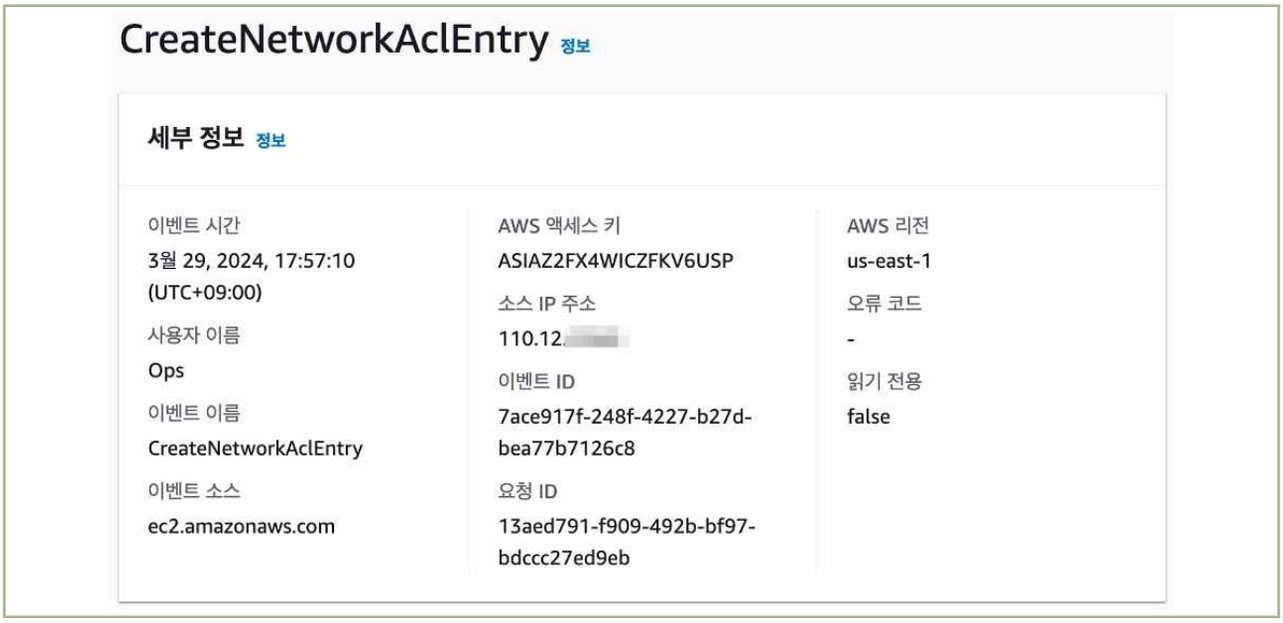
(3) 네트워크 ACL 규칙 행위 감사

- **(AWS 관리 콘솔)** '콘솔 홈' → 'CloudTrail' → '이벤트 기록' → '관련 로그 검색' 및 해당 Action에 해당하는 이벤트 이름으로 검색 하여 이벤트를 확인할 수 있습니다.
  - 네트워크 ACL 을 신규 생성하는 경우 CreateNetworkAcl 이벤트가 기록되므로 해당 행위를 감사할 수 있습니다.



| 그림 5-5-12 | CloudTrail 의 CreateNetworkAcl 이벤트

- 네트워크ACL을 변경하는 경우 요청한 내용에 따라 해당하는 이벤트가 기록되며, 예를 들어 네트워크 ACL에 규칙을 새롭게 추가하는 경우 'CreateNetworkAclEntry' 이벤트가 기록됩니다.



| 그림 5-5-13 | CloudTrail 의 CreateNetworkAclEntry 이벤트

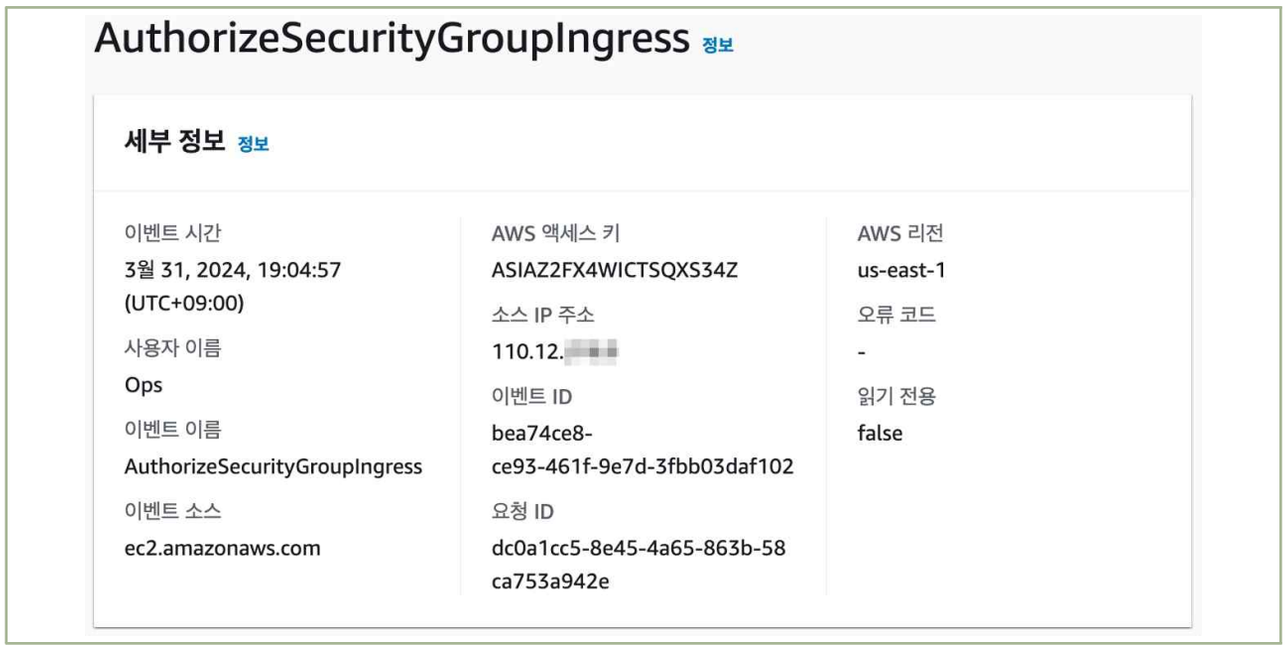
- 네트워크 ACL을 삭제하는 경우 DeleteNetworkAcl 이벤트가 기록됩니다.

| 그림 5-5-14 | CloudTrail 의 DeleteNetworkAcl 이벤트

- 보안 그룹에 대한 이벤트도 동일한 방법으로 검색할 수 있으며, 보안 그룹을 생성하는 경우 'CreateSecurityGroup' 이벤트가 기록됩니다.

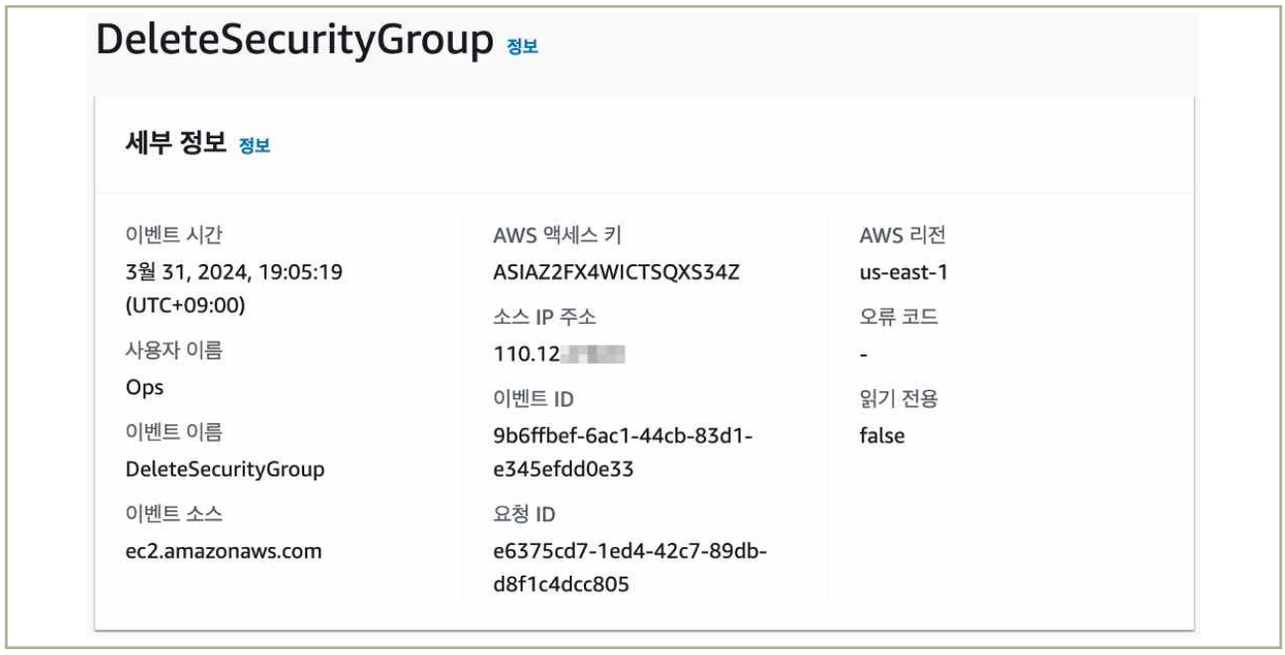
| 그림 5-5-15 | CloudTrail 의 CreateSecurityGroup 이벤트

- 보안 그룹을 변경하는 경우에도 관련 이벤트가 기록됩니다. 예를 들어, 인바운드 규칙을 추가할 경우 'AuthorizeSecurityGroupIngress' 이벤트가 기록됩니다. 이벤트 레코드에는 실제 인바운드 규칙과 관련된 세부 정보를 포함하고 있습니다.



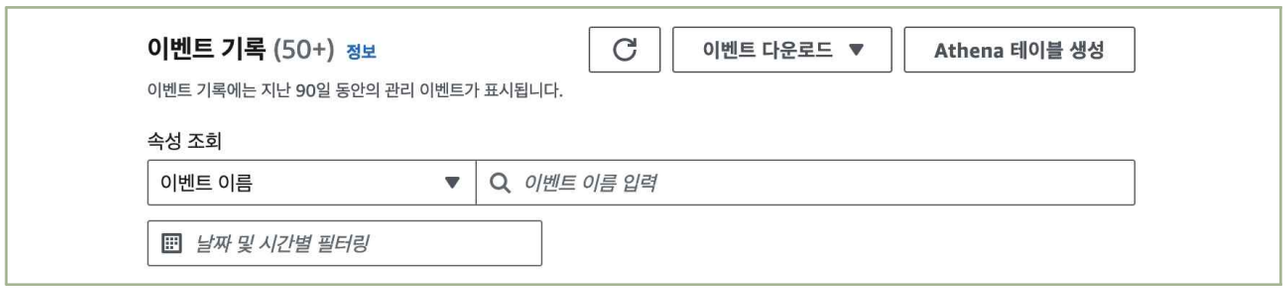
| 그림 5-5-16 | CloudTrail 의 AuthorizeSecurityGroupIngress 이벤트

- 보안 그룹을 삭제하는 경우 'DeleteSecurityGroup' 이벤트가 기록됩니다.



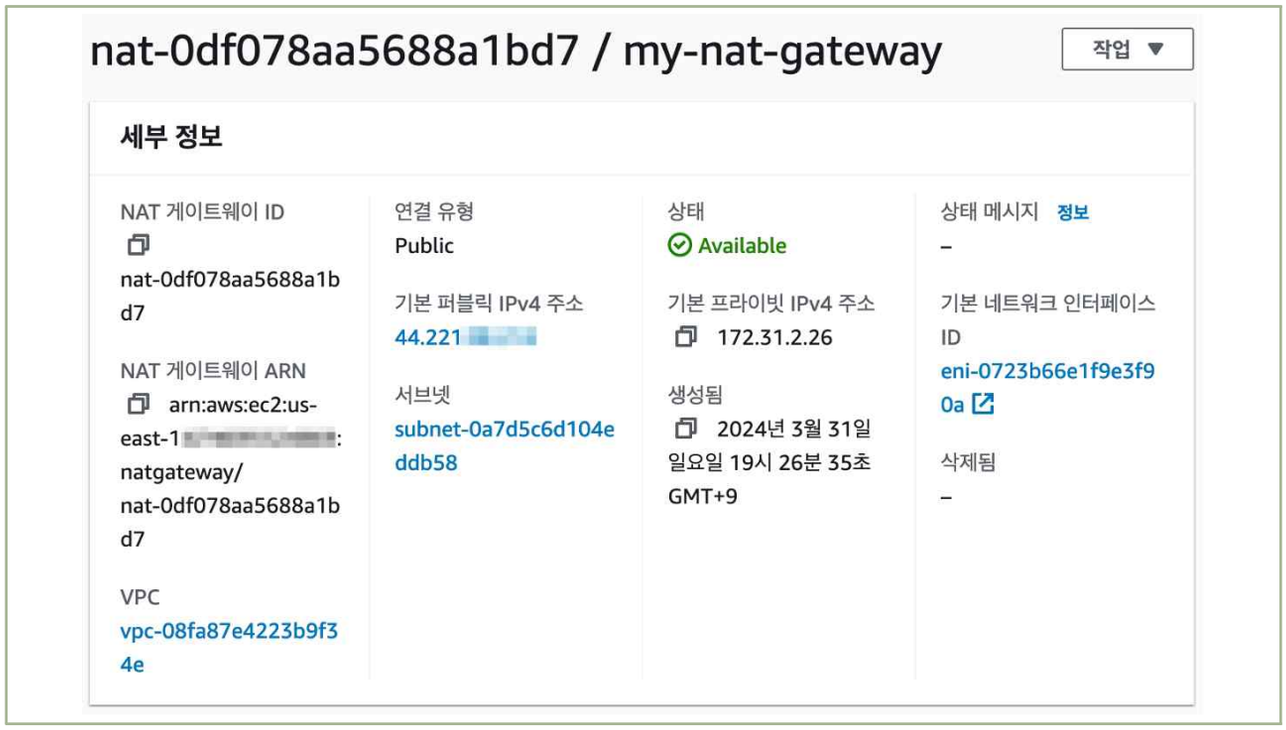
| 그림 5-5-17 | CloudTrail 의 DeleteSecurityGroup 이벤트

- 이벤트는 다양한 조건으로 검색이 가능하며, 이벤트 ID, AWS 엑세스 키, 리소스 이름, 이벤트 이름 등으로 필터링 하여 검색할 수 있습니다.
- 그리고 검색된 데이터는 이벤트를 CSV 또는 JSON 으로 다운로드도 할 수 있습니다.



| 그림 5-5-18 | CloudTrail 이벤트 기록 조회

- '콘솔 홈' → 'VPC' → 'NAT 게이트웨이' 에서 NAT 게이트웨이를 생성할 수 있으며, 생성된 NAT 게이트웨이의 세부 정보를 확인할 수 있습니다. 그리고 앞서 생성한 VPC Flow Log에서 NAT 게이트웨이의 주소도 확인할 수 있습니다. 다만, NAT를 통과하게 되면 Source IP는 Original IP가 아닌 NAT의 IP로 변경되어 기록이 됩니다.



| 그림 5-5-19 | NAT 게이트웨이 세부 정보 확인

[TIP] VPC 플로우 로그 포맷을 확장할 경우 패킷 레벨의 출발지 IP, 목적지 IP 주소를 기록할 수 있습니다. NAT 를 통과 하기 이전의 원본 소스의 IP를 알 수 있고 VPC 플로우의 관련된 필드명은 'pkt-srcaddr' , 'pkt-dstaddr' 입니다.

- NAT 게이트웨이를 생성하거나 삭제한 기록은 CloudTrail 에서 관련된 이벤트 명으로 확인이 가능하며, NAT 를 생성한 경우에는 'CreateNatGateway' 이벤트로 만약 NAT를 삭제한 경우에는 'DeleteNatGateway' 이벤트로 기록이 됩니다.

4 참고 사항

- VPC 플로우 로그는 네트워크 트래픽 경로 외부에서 수집되므로 네트워크 처리량이나 지연시간에 영향을 주지 않습니다.
- 그리고 언급된 VPC Flow Logs의 로그 정보는 Version2를 기반으로 설명합니다.

1 기준

식별번호	기준	내용
5.6.	**계정 변동사항에 대한 행위추적성 확보**	**클라우드 계정 변동사항에 대한 행위추적성(로그 등)을 확보하여야 한다.**

2 설명

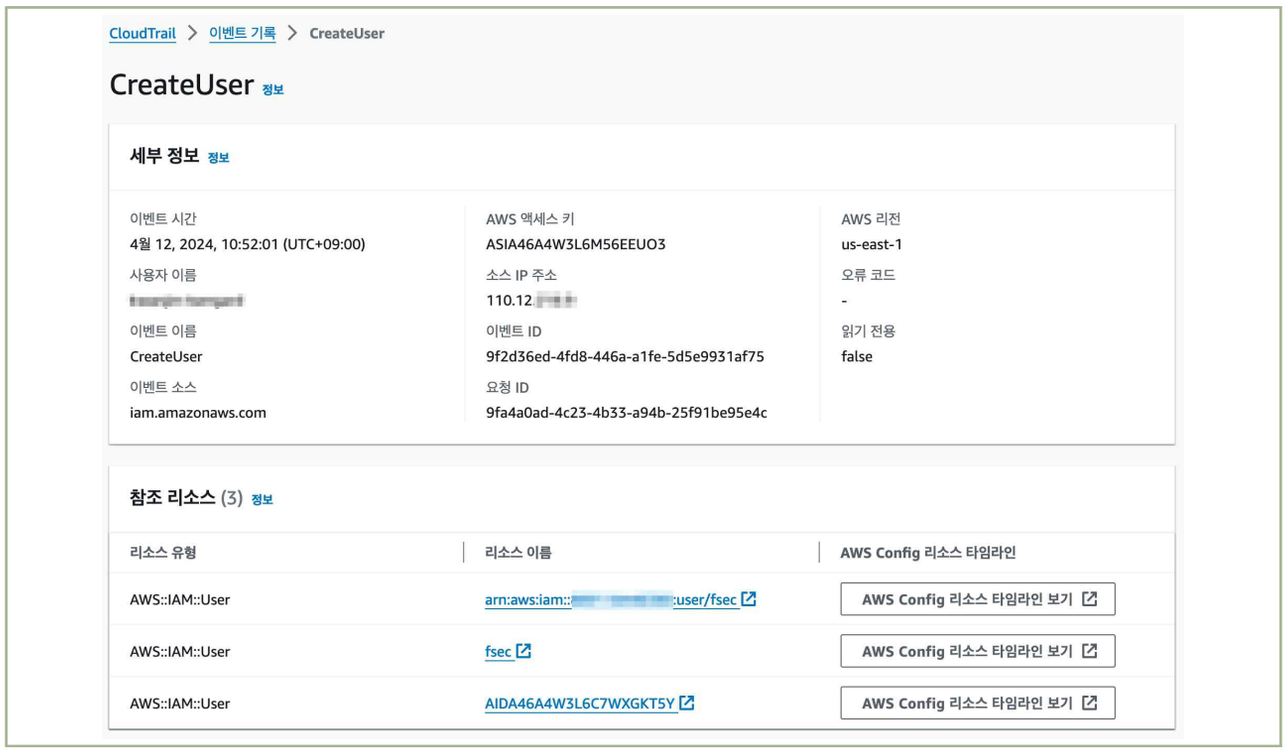
- 클라우드 계정 변동사항에 대한 행위추적성(로그 등)을 확보하여야 한다.
  - 행위 감사로그
    1) 클라우드 가상자원 관리시스템 접속 계정 생성, 변경, 삭제에 관한 사항
    2) 클라우드 가상자원(서버, 데이터베이스 등) 접속 계정 생성, 변경, 삭제에 관한 사항

3 우수 사례

(1) 가상자원 관리시스템 접속 계정 생성 이벤트 기록

- 계정을 생성하게 되면 CreateUser 이벤트를 호출하게 되고, 해당 이벤트에서 생성한 접속 계정에 대한 정보를 확인할 수 있습니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → 'CloudTrail' → '이벤트 기록' 에서 생성된 계정과 관련된 기록을 검색하여 확인할 수 있습니다.

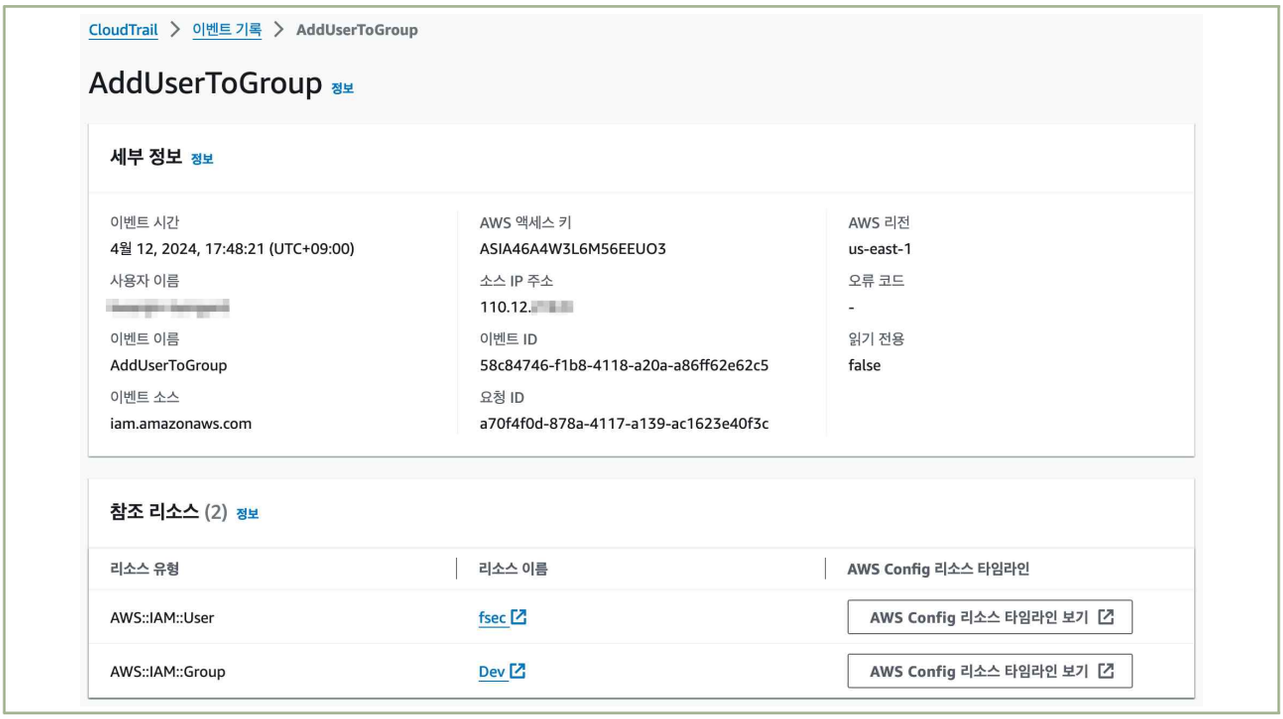


| 그림 5-6-1 | 가상자원 관리시스템 접속 계정 생성 이벤트

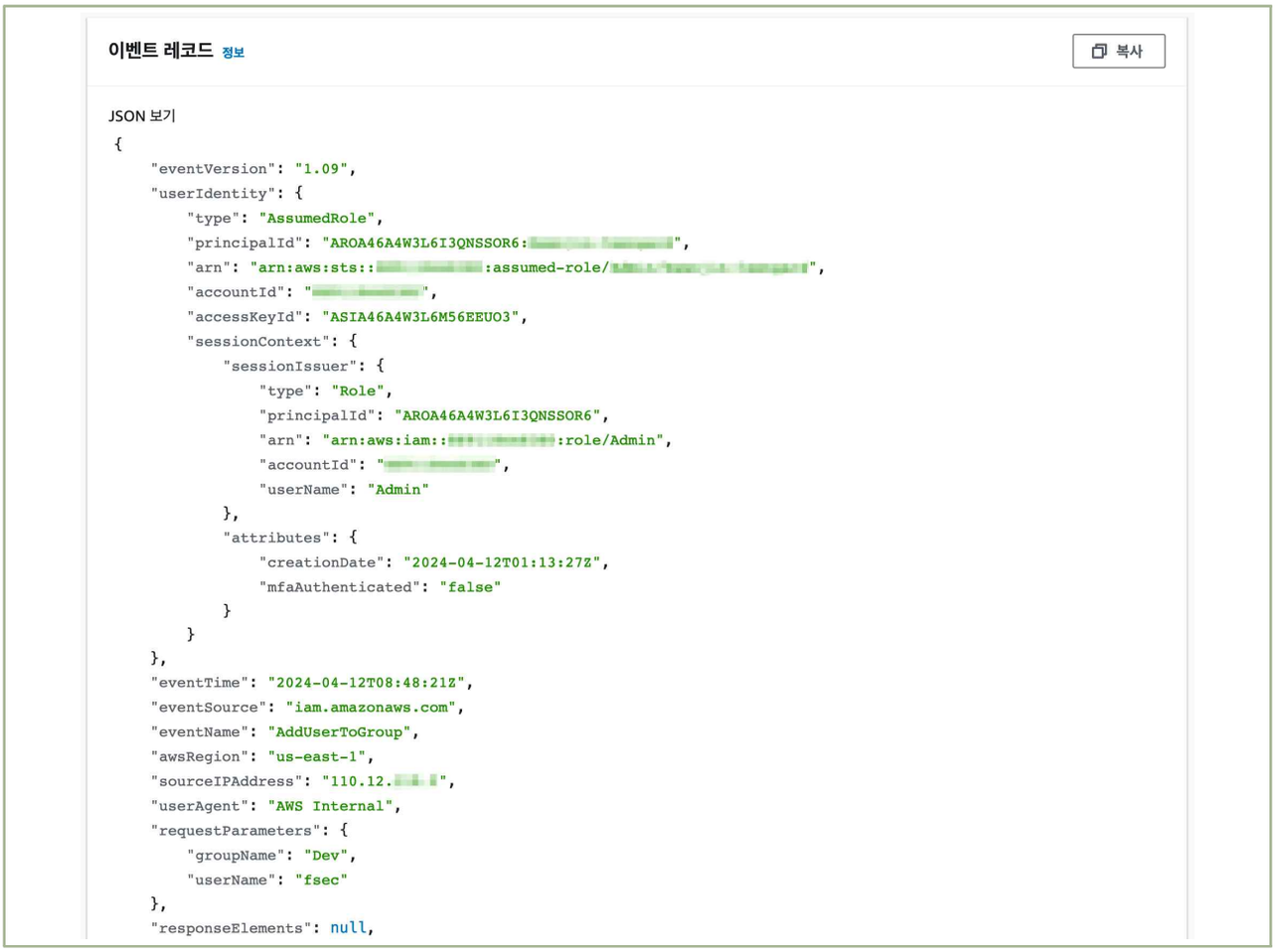
(2) 가상자원 관리시스템 접속 계정 변경 이벤트 기록

- 계정 변경과 관련해서는 변경 내용에 따라 해당하는 이벤트가 발생하게 됩니다. 만약 사용자를 그룹에 추가할 경우 AddUserToGroup 이벤트를 호출하게 되며, 해당 이벤트에서 변경한 계정에 대한 정보를 확인할 수 있습니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → 'CloudTrail' → '이벤트 기록' 에서 변경된 계정과 관련된 기록을 검색하여 확인할 수 있습니다.



| 그림 5-6-2 | 가상자원 관리시스템 접속 계정 변경 이벤트

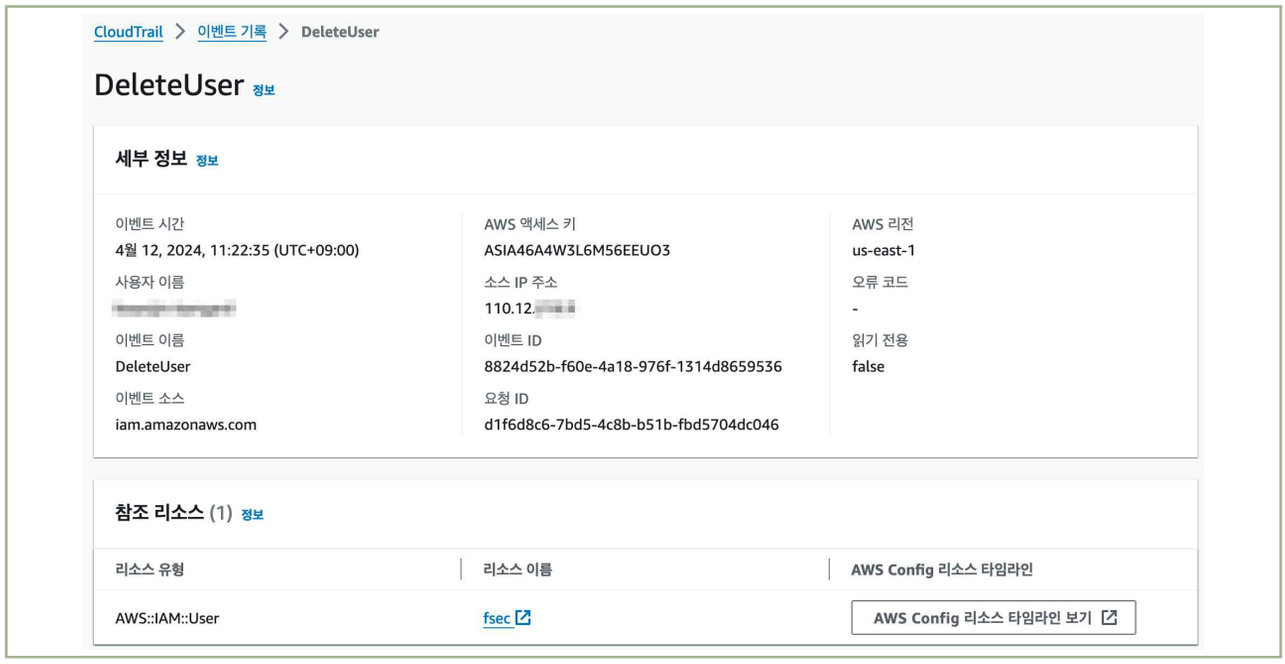


| 그림 5-6-3 | 접속 계정 변경 세부 이벤트 레코드

(3) 가상자원 관리시스템 접속 계정 삭제 이벤트 기록

- 계정을 삭제하게 되면 DeleteUser 이벤트를 호출하게 되고, 해당 이벤트 이름으로 삭제된 계정 정보를 확인할 수 있습니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → 'CloudTrail' → '이벤트 기록' 에서 삭제된 계정과 관련된 기록을 검색하여 확인할 수 있습니다.



| 그림 5-6-4 | 가상자원 관리시스템 접속 계정 삭제

(4) 계정 관련 행위 추적성 API

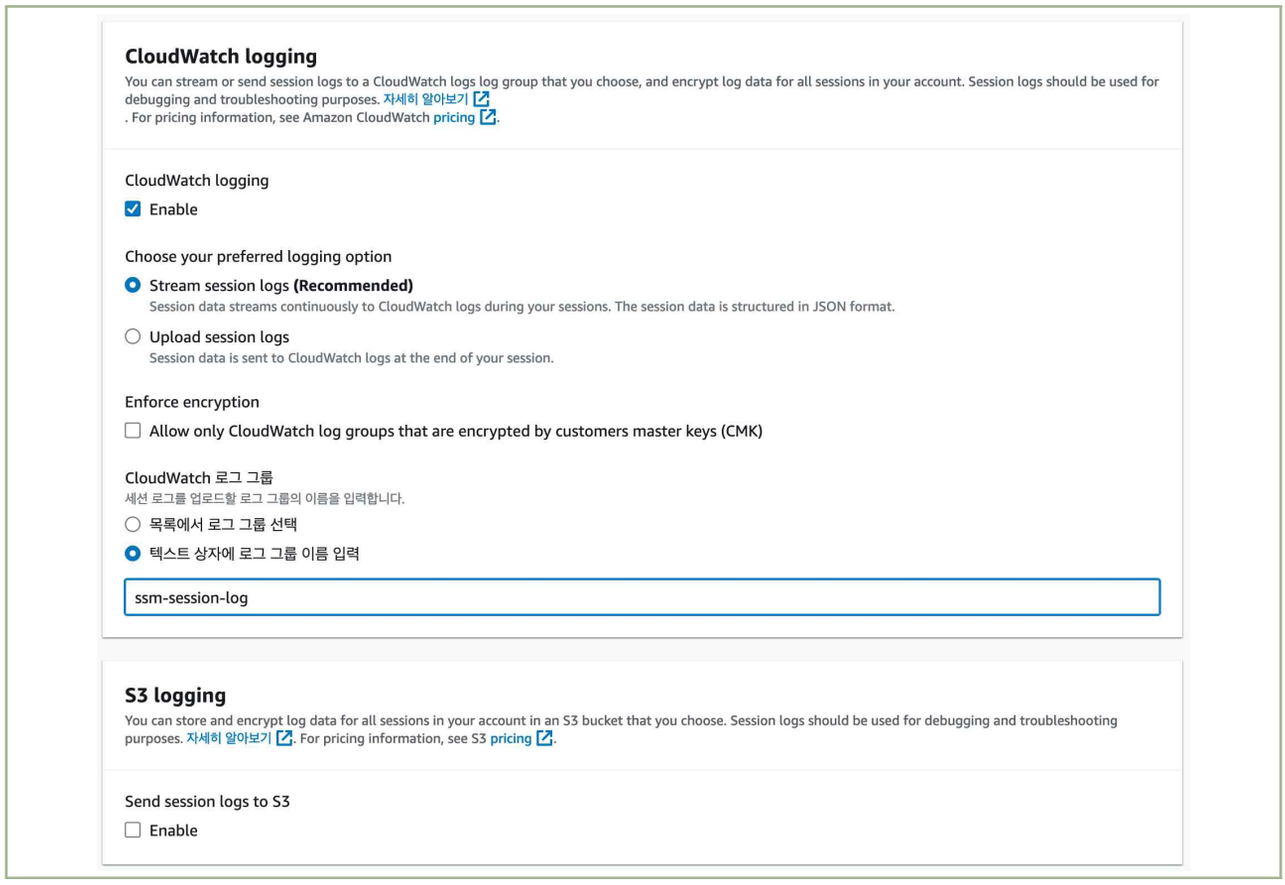
- 계정의 행위 추적과 관련하여 도움을 줄 수 있는 이벤트 이름은 아래의 도표를 참고하시기 바랍니다.

| 표 5-6-1 | 계정변동 사항 관련 행위 이벤트 이름

이벤트 이름	이벤트 설명
CreateUser	사용자 생성
CreateLoginProfile	특정 IAM 사용자에 대한 패스워드 생성
CreateAccessKey	사용자에게 프로그래밍 방식 액세스 권한 부여
AddUserToGroup	사용자를 하나 이상의 그룹에 추가
AttachUserPolicy	관리 정책을 지정된 사용자에게 연결
UpdateUser	사용자 이름 또는 경로를 변경
DeleteUser	사용자 삭제
DeleteLoginProfile	사용자의 암호 삭제
DeleteAccessKey	사용자의 액세스키 삭제
DeleteUserPolicy	사용자의 인라인 정책 삭제
DetachUserPolicy	사용자에 연결된 관리형 정책을 분리
DeleteSSHPublicKey	사용자에 연결된 SSH 퍼블릭 키 삭제

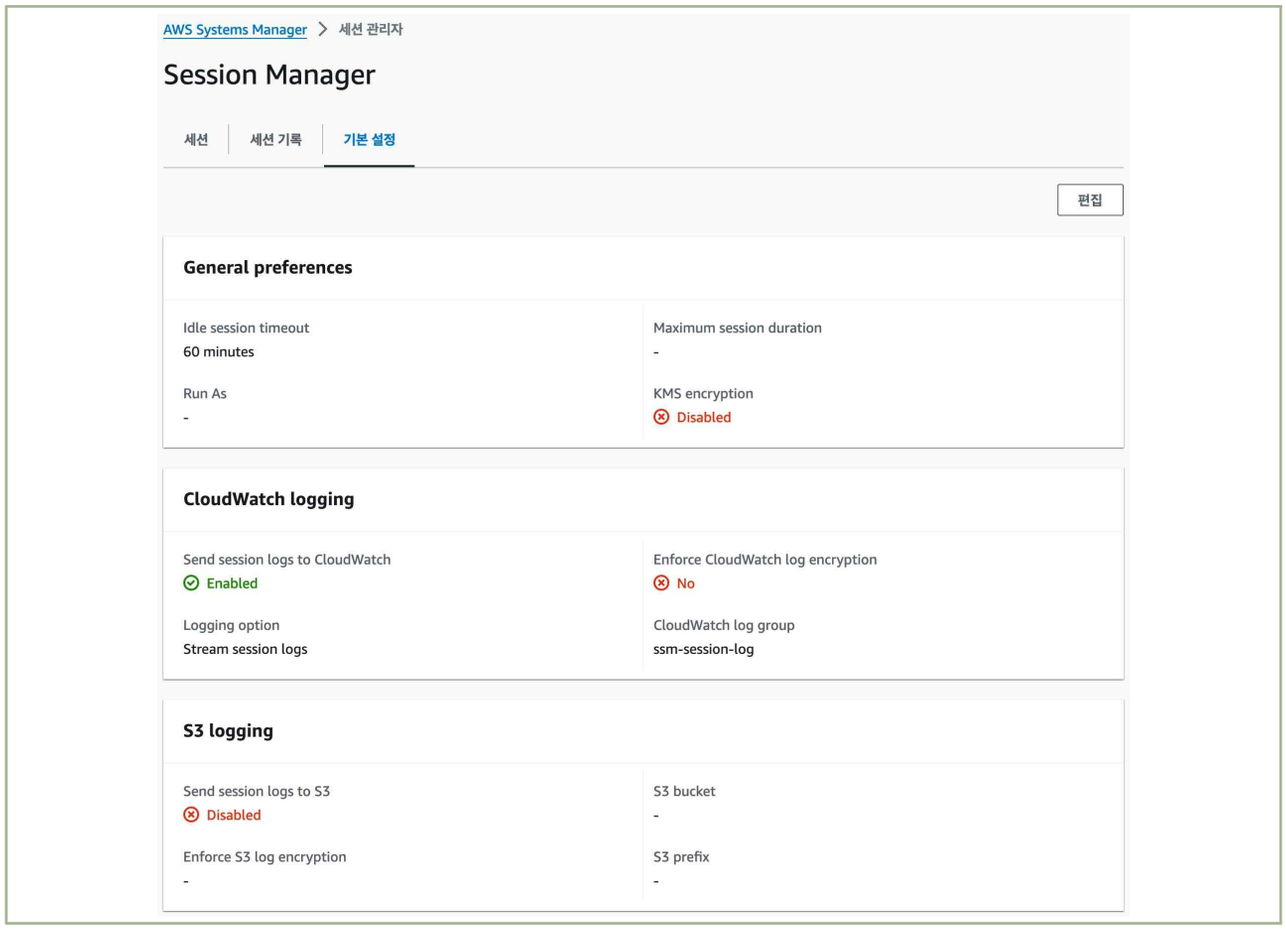
(5) 클라우드 가상자원(서버) 접속 계정 기록

- 클라우드 내에서 생성한 가상자원의 내부에서 수행하는 계정 생성, 수정, 삭제는 클라우드 제공자의 책임영역에 해당되지 않아 관련 모니터링이 필요할 경우 EC2 인스턴스는 AWS Systems Manager(SSM)를 이용한 세션 로그 전송 또는 '7.1' 에서 기술한 CloudWatch를 이용한 시스템 내의 특정 경로 로그파일 전송을 고려할 수 있습니다.
- SSM에서 세션 로그 기능은 세션 로그를 CloudWatch 또는 S3에 기록할 수 있으며, 세션 로그에는 계정의 변경사항 뿐만 아니라 세션에서 발생하는 모든 로그가 기록됩니다. 다만 SSM을 통해 접속하는 경우에 한합니다.
- 아래 예제는 CloudWatch 로깅 기능을 'Enable' 하였고, 로깅 옵션으로 'Stream session logs' 가 선택된 예시 입니다.
- 이후 CloudWatch 에 기록할 로그 그룹의 이름을 입력하고 저장하면 기록이 되며, 만약 로그 그룹이 없을 경우 아래의 '로그 그룹 생성' 부분을 참고하시기 바랍니다.



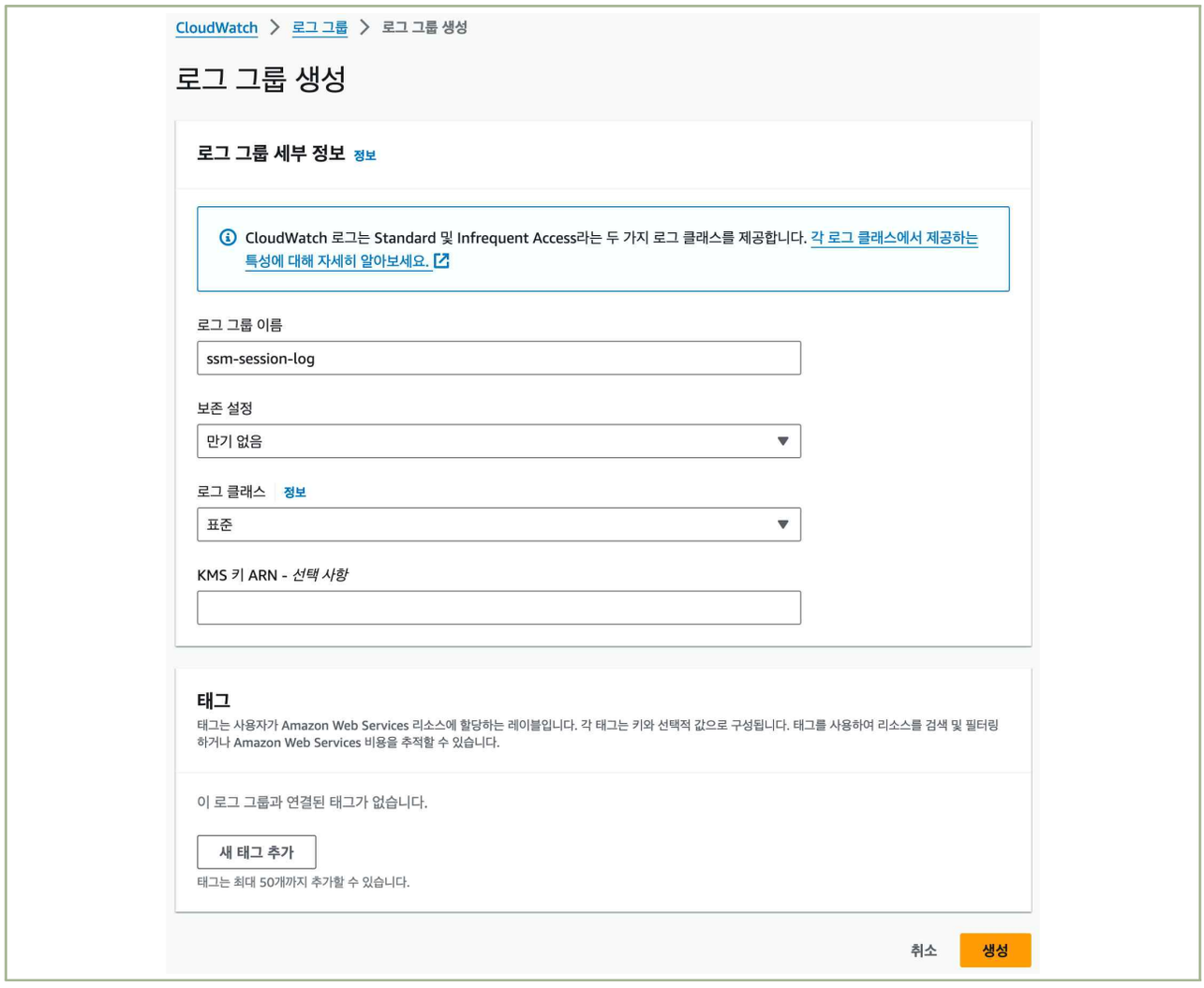
| 그림 5-6-5 | 세션 로그 기록 설정

- 설정이 완료된 후 세션 관리자의 기본 설정을 보면 CloudWatch의 세션 로그 기록이 설정되어 있음을 확인할 수 있습니다.



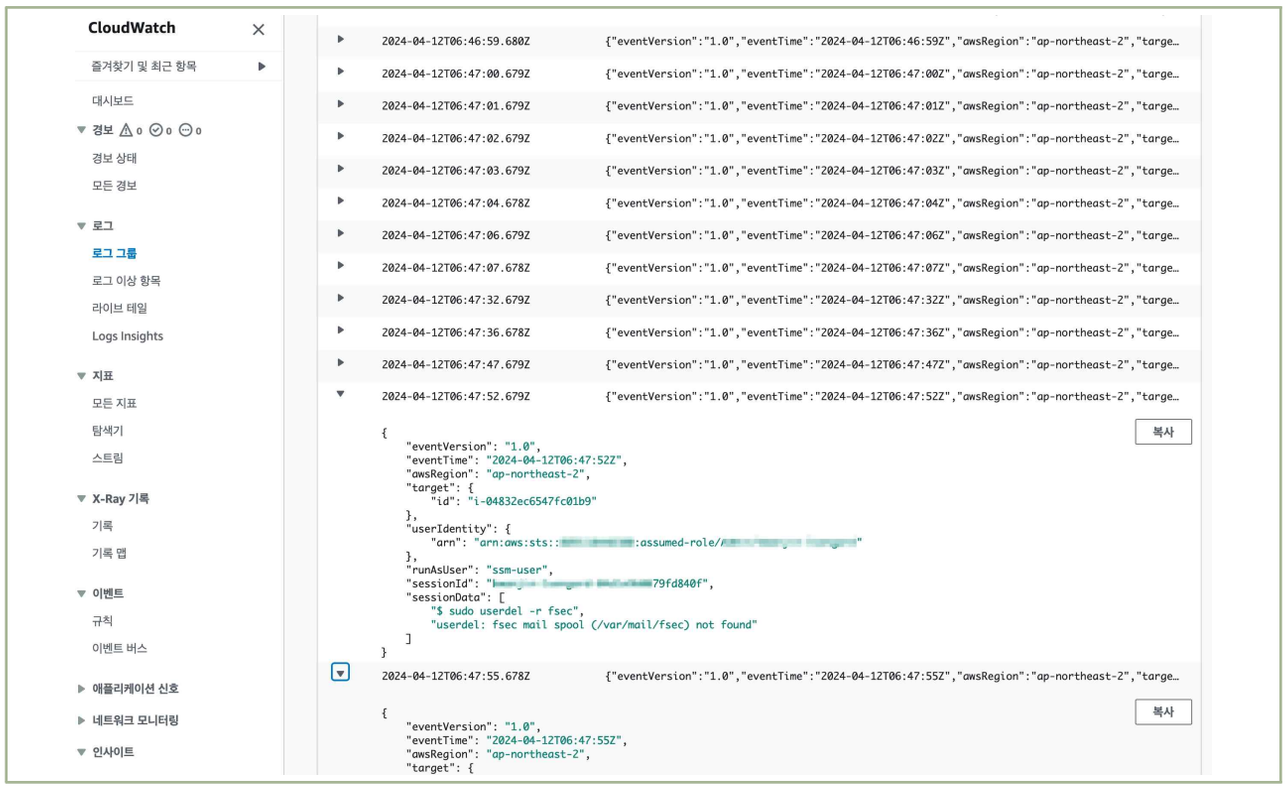
| 그림 5-6-6 | 세션 로그 기록 설정 확인

- 세션 로그 기록 설정을 완료한 이후, 세션매니저를 통해 접속한 기록의 경우 로그가 기록 됩니다.
- CloudWatch에 로그를 전송하기 위해서는 CloudWatch에 '로그 그룹' 을 생성해야 하며, 얼마나 오래동안 보관할지와 로그 클래스 등을 설정할 수 할 수 있습니다. 데이터를 암호화하여 저장하는 것이 필요하면 KMS 키를 선택하여 암호화된 형태의 로그 그룹을 만들 수 있습니다.



| 그림 5-6-7 | 로그 그룹 생성

- 세션 매니저를 통해 시스템 접근 시 접근한 시스템에 대한 세션 로그가 기록되며, 화면의 예와 같이 사용자를 삭제하기 위해 수행된 'sudo userdel -r fsec' 의 명령어 와 같은 사용 기록도 함께 기록되므로 명령어 이력을 확인할 수 있습니다.

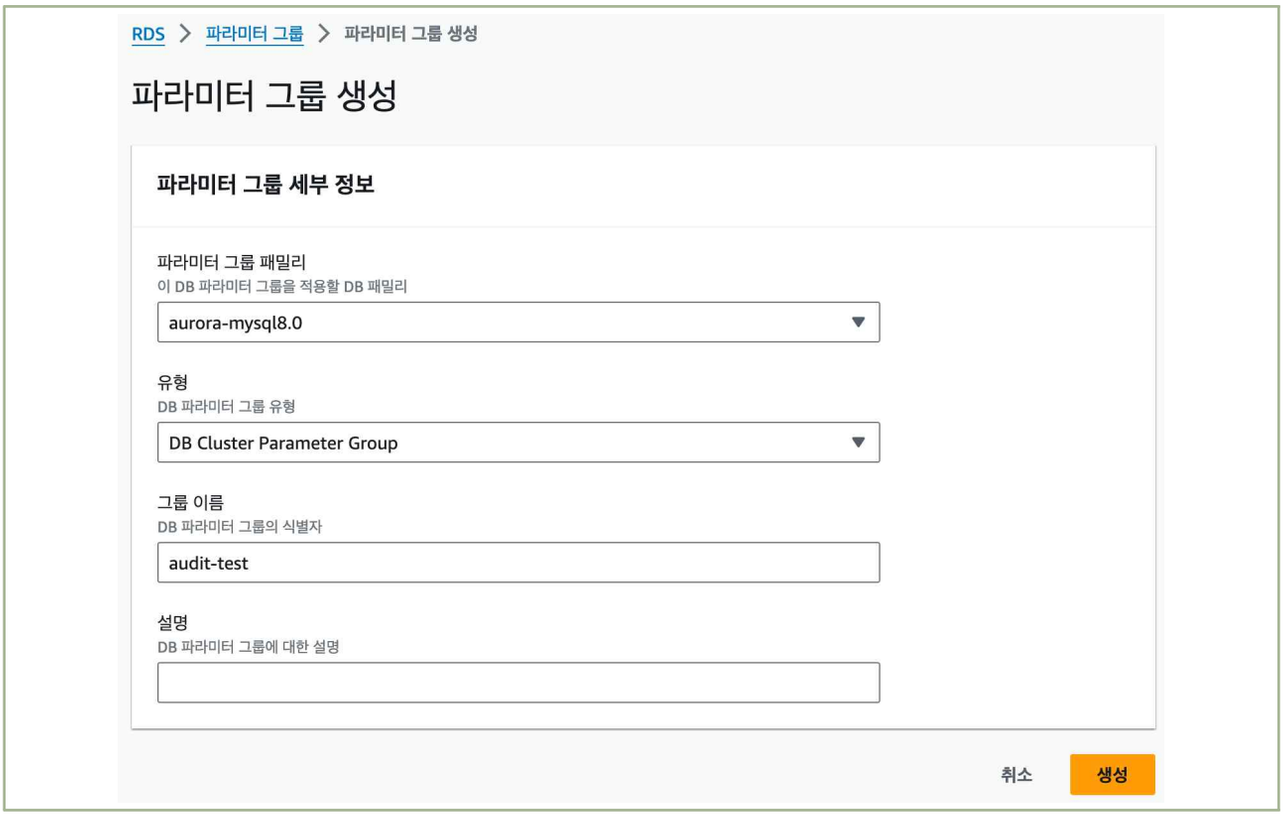


| 그림 5-6-8 | 세션 로그 기록 확인

(6) 데이터베이스 계정 관리 기록

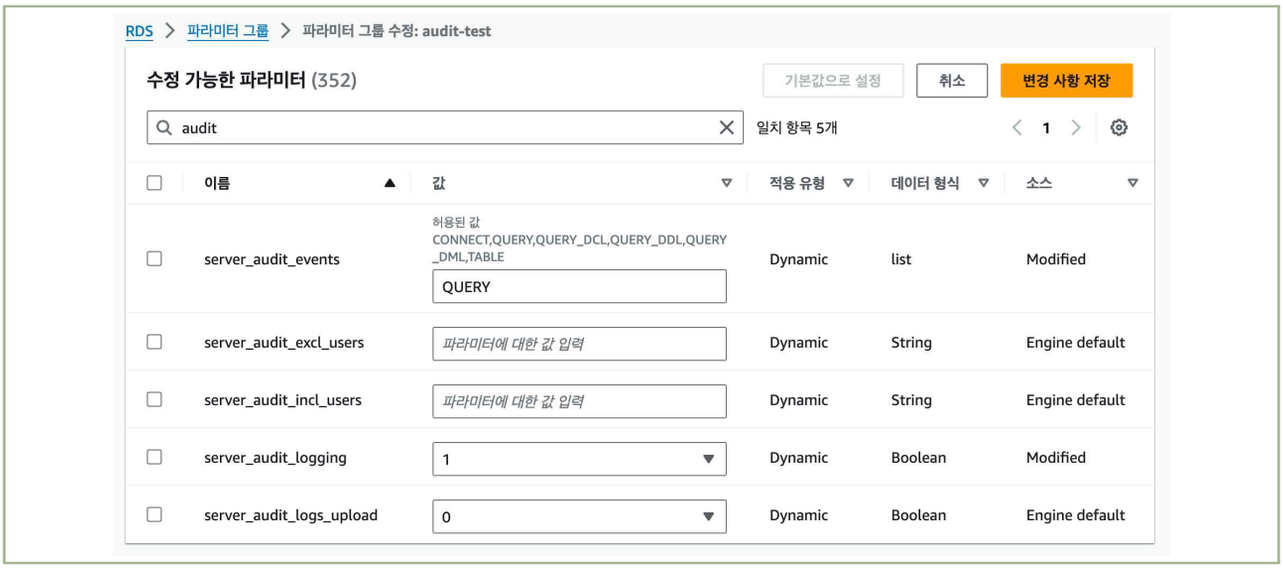
- 사용자가 사용하는 데이터베이스의 종류에 따라 계정 정보 기록이 다르므로, 그에 맞는 방법 확인이 필요합니다. 본 참고서에서는 많이 사용하는 데이터베이스 중에 하나인 Amazon RDS 의 Aurora Mysql 을 대상으로 합니다. 데이터베이스의 감사 로그를 이용하면 쿼리 정보 등을 기록할 수 있으므로 계정 생성, 삭제 등의 쿼리 정보를 확인할 수 있습니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → ''RDS' -〉 '파라미터 그룹' -〉 '파라미터 그룹 생성' 에서 파라미터 그룹을 신규로 생성하고 사용하는 그룹 패밀리와 유형 그리고 그룹 이름을 작성합니다.



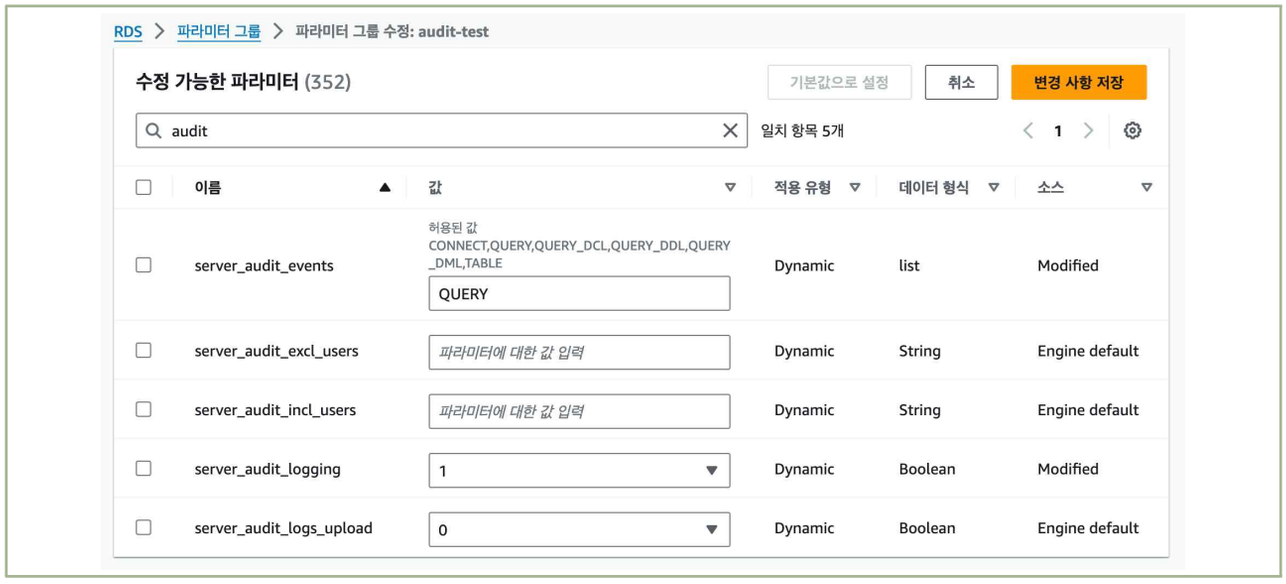
| 그림 5-6-9 | 데이터베이스 파라미터 그룹 생성

- 생성한 파라미터 그룹에서 '편집' 을 통해 'audit' 으로 검색해 보면 수정 가능한 파라미터들이 있으며, 아래의 값을 변경합니다.
  - server_audit_logging : 값을 1로 변경
  - server_audit_events : 감사 로그에서 기록할 데이터를 선택 (본 참고서에서는 'QUERY' 를 사용함)



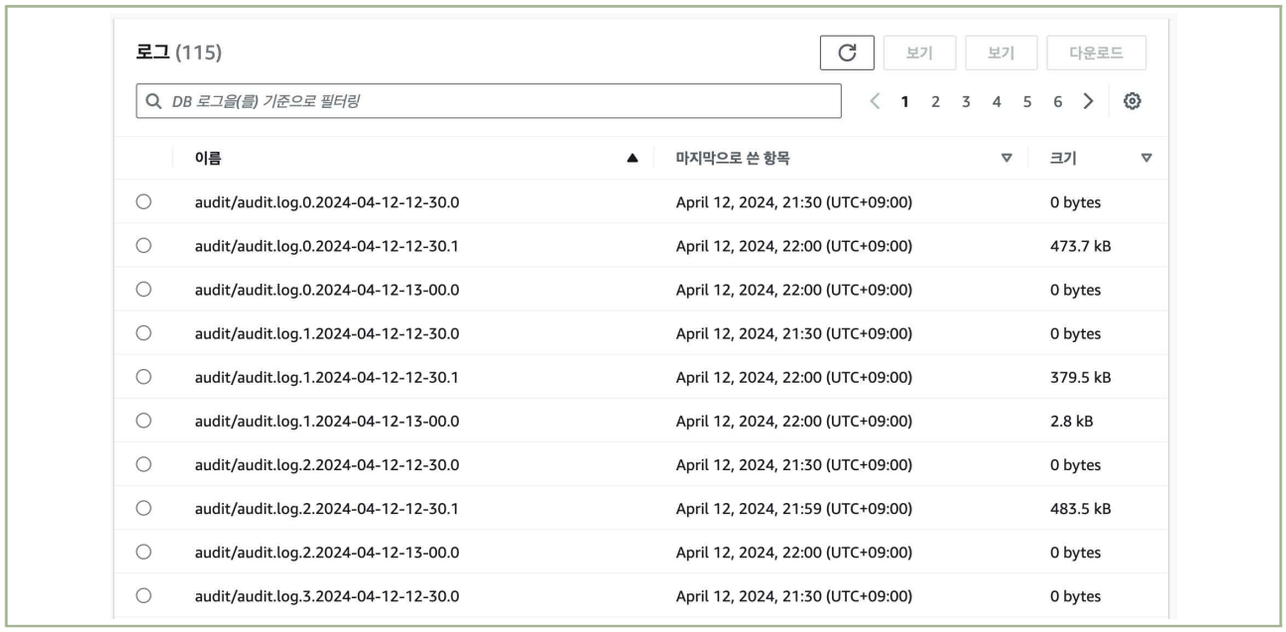
| 그림 5-6-10 | 파라미터 그룹에서 감사 로그 설정

- '콘솔 홈' → 'Amazon RDS' → '데이터베이스' 에서 생성한 DB 클러스트를 선택하고 '수정' 을 눌러서 설정 정보를 변경합니다. 데이터베이스 옵션에 'DB 클러스트 파라미터 그룹' 을 앞서 만들어준 파라미터 그룹으로 변경해 주고 저장합니다.
- 주의할 점은 기본 파라미터 그룹에서 사용자 지정 파라미터 그룹으로 변경한 후, 변경 사항을 반영하기 위해서는 반드시 재부팅이 필요합니다.



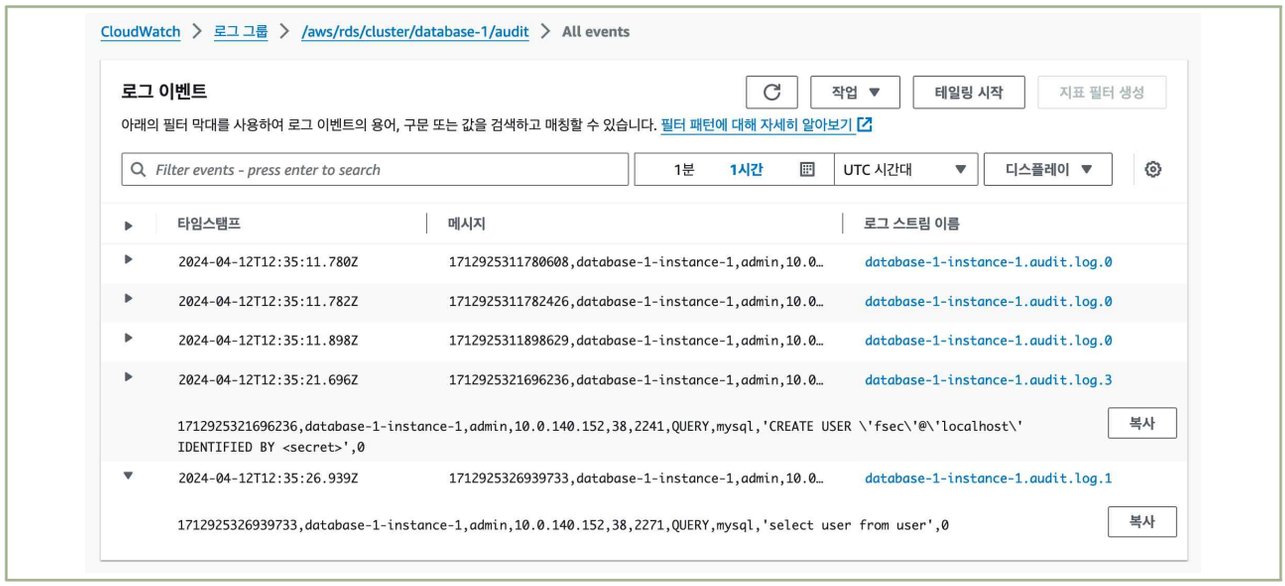
| 그림 5-6-11 | 데이터베이스 옵션 파라미터 그룹 변경 설정

- 데이터베이스의 인스턴스를 클릭하면 '로그 및 이벤트' 항목에서 아래와 같이 감사 로그를 확인할 수 있습니다. 해당 로그를 선택하고 '보기' 를 누르면 실제 로그를 확인할 수 있습니다.



| 그림 5-6-12 | 감사 로그 확인

- 이외 CloudWatch 의 로그 그룹에서 생성한 RDS 의 데이터베이스의 Audit 그룹에서 감사 로그 이벤트를 좀더 쉽게 확인 및 검색해 볼 수 있으며, 아래 화면과 같이 쿼리를 이용해 사용자 생성과 사용자 확인하는 쿼리의 기록을 확인할 수 있습니다.



| 그림 5-6-13 | CloudWatch 를 통한 감사 로그 확인

4 참고 사항

- 클라우드 가상 자원 내에서 발생하는 계정 변경 기록은 사용하려는 목적에 따라 Third-Party 제품을 고려해 볼 수 있습니다.
- Amazon RDS 의 데이터베이스 활동에 대한 감사 로그 구성은 아래 링크를 참고 바랍니다.
  - https://aws.amazon.com/ko/blogs/korea/configuring-an-audit-log-to-capture-database-activities-for-amazon-rds-for-mysql-and-amazon-aurora-with-mysql-compatibility/

1 기준

식별번호	기준	내용
5.7.	**계정 변경사항에 관한 모니터링 수행**	**클라우드 서비스 이용 계정 변경사항(생성, 삭제 등)에 관한 로깅 및 모니터링을 수행하여야 한다.**

2 설명

- 클라우드 서비스 이용 계정 변경사항에 관한 모니터링을 수행하여야 한다.

  - 예시

    1) 계정 변경사항에 관한 상시 모니터링 수행
    2) 전자금융감독규정 및 금융회사 내부규정 등에 수립된 주기에 맞추어 주기적 검토 수행
    3) 관리자 계정에 대해서는 이중확인 수행 등

3 우수 사례

1) 계정 변경사항 모니터링

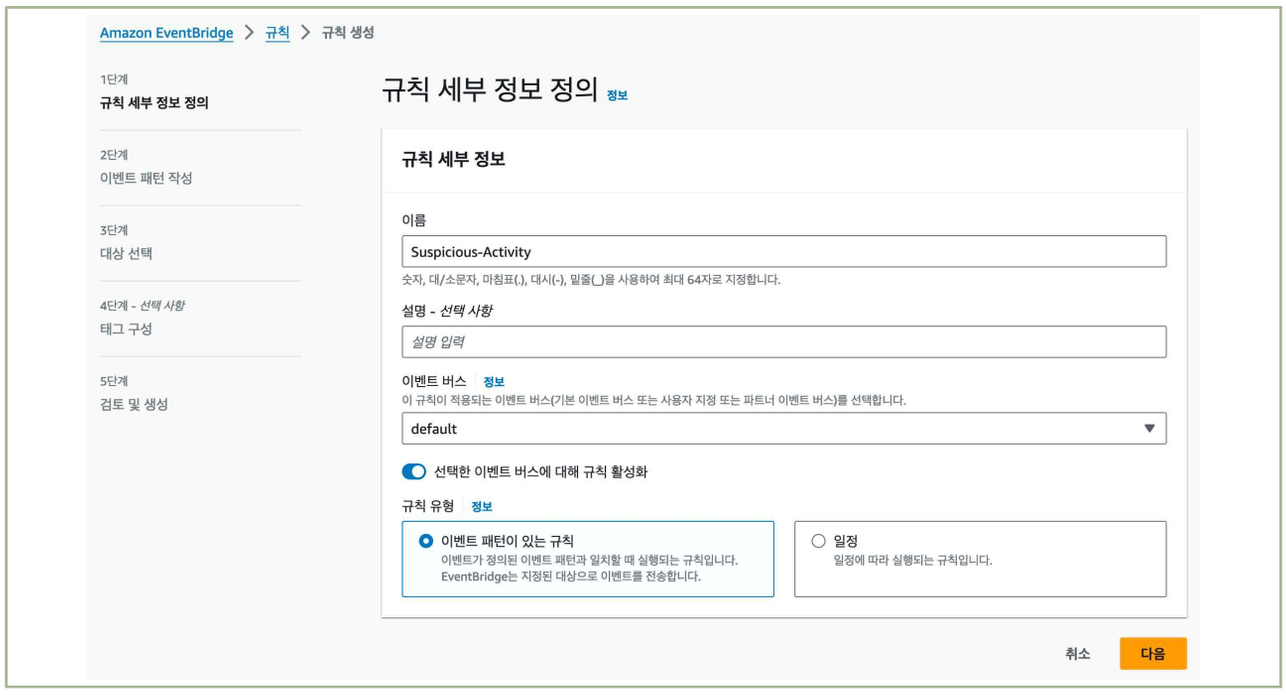
- 계정과 관련된 변경사항 발생시 상시 모니터링 구성을 위하여 Amazon EventBridge 를 이용하여 구성할 수 있습니다. '5.6 계정 변동사항에 대한 행위추적성 확보' 에 언급된 계정 변경과 관련된 이벤트가 호출되었을 경우 관련 내용을 통보 받을 수 있습니다.

(1) 계정 변경사항 모니터링 범위 검토

- 계정변경을 모니터링 하기 위해 해당 범위와 어떤 이벤트 이름을 사용할 것인지 검토가 필요합니다.
- 사용자 계정의 생성, 변경, 삭제와 관련된 이벤트 이름은 CreateUser, UpdateUser, DeleteUser 이며 필요에 따라 모니터링에 포함할 이벤트 이름을 확장합니다.

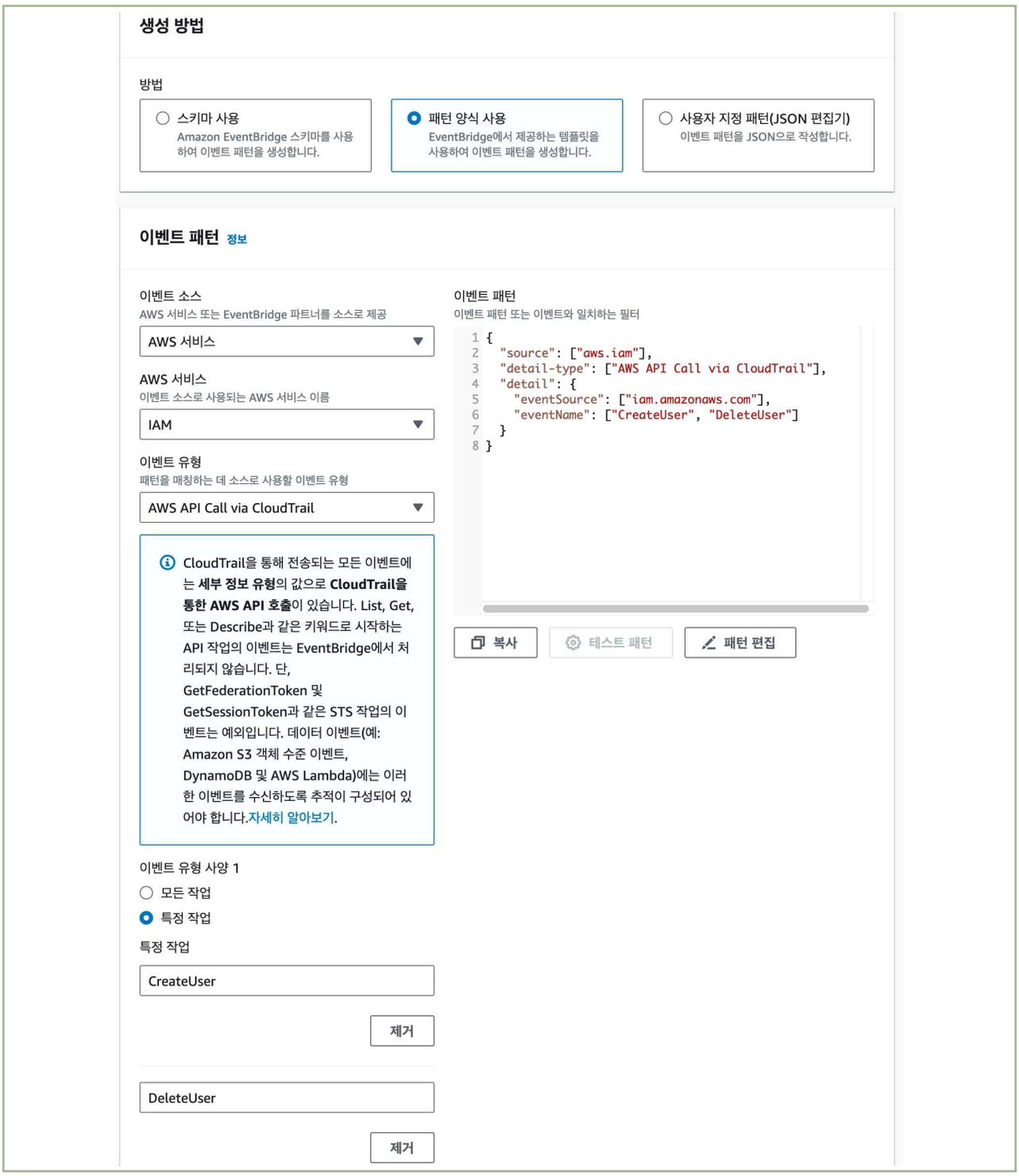
(2) 계정 변경사항 모니터링 구성

- **(AWS 관리 콘솔)** 이벤트를 모니터링 하기 위해 Amazon EventBridge 를 사용하며, '콘솔 홈' → 'Amazon EventBridge' → '버스' → '규칙' → '규칙 생성' 으로 모니터링을 위한 규칙을 생성합니다. 규칙 세부 정보에서 사용할 이름을 넣어주고, 규칙 유형은 '이벤트 패턴이 있는 규칙'을 선택합니다.



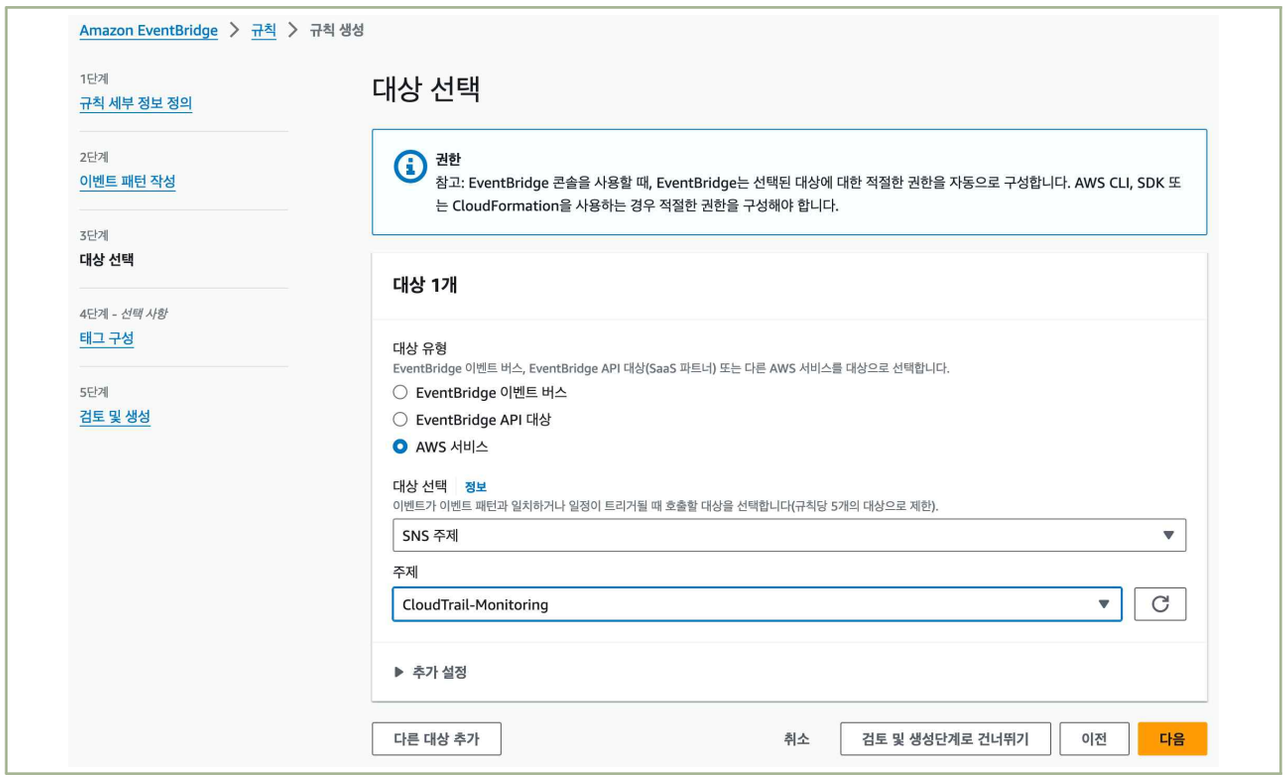
| 그림 5-7-1 | Amazon EventBridge 규칙 생성

- 모니터링 대상이 되는 이벤트 이름을 탐지하기 위한 패턴을 만들어 준다. '생성 방법' 은 '패턴 양식 사용' 을 선택하고 이벤트 패턴에서 다음과 같이 선택합니다.
  - 이벤트 소스 : AWS 서비스
  - AWS 서비스 : IAM
  - 이벤트 유형 : AWS API Call via CloudTrail
  - 이벤트 유형 사양 : 특정 작업을 선택 후 CreateUser, DeleteUser 입력



| 그림 5-7-2 | 이벤트 패턴 생성 방법

- 이벤트 패턴을 작성 후, 해당 이벤트 패턴에 매칭되는 경우 대상을 선택합니다. 여러가지 방법을 이용할 수 있지만, 여기서는 AWS 서비스인 Amazon SNS 서비스를 이용하여 탐지된 이벤트를 전송합니다. Amazon SNS에 대한 설정은 아래 '(3) 계정 변경사항 알림받기' 를 참고 바랍니다.



| 그림 5-7-3 | 이벤트 규칙 매칭 후 수행할 대상 선택

- 본 참고서에서는 Amazon SNS 서비스의 이메일을 이용한 기능을 이용하여 알람을 전송합니다. 기본 설정은 이메일을 통해서 CloudTrail 의 JSON 이벤트 레코드가 전송되므로, 이해하기 쉽게 '추가 설정' 에서 변환을 구성합니다.
- '추가 설정'에서 '입력 변환기'를 선택한 후 '입력 경로' 와 '템플릿'을 화면과 같이 변경하며, 이벤트 레코드에서 시간, 이벤트 이름, 사용자 이름, IP 주소를 추출해 내고 '템플릿' 에서 보기 쉽게 아래와 같이 이메일 전달 내용을 작성합니다.

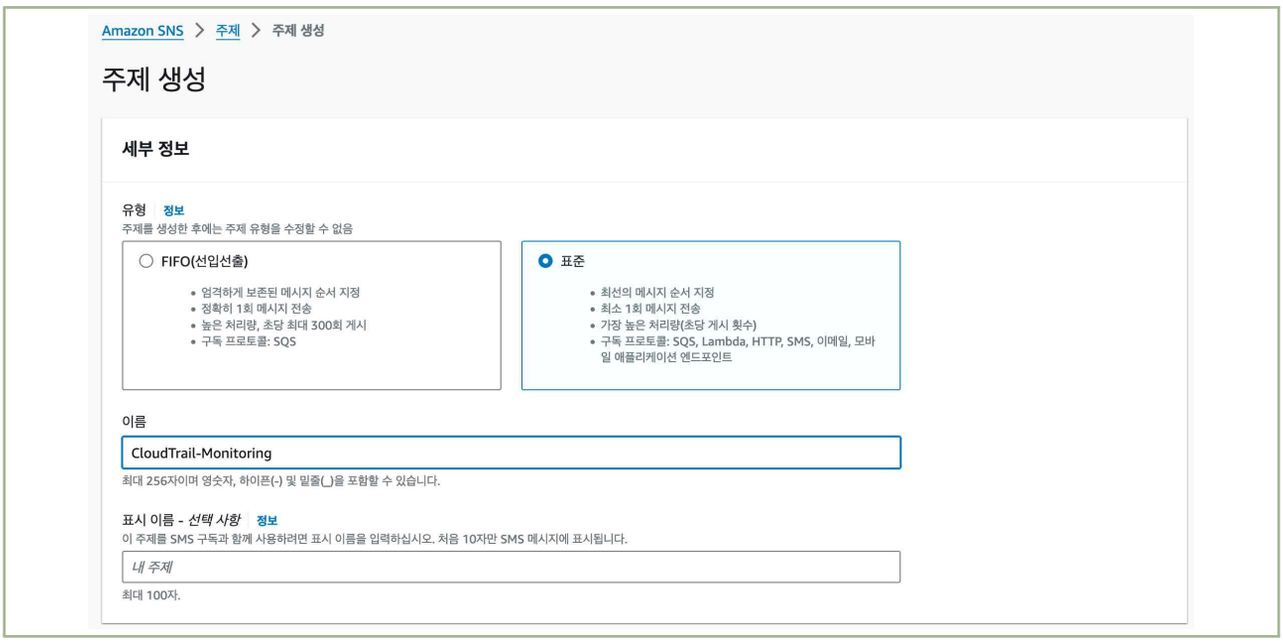
"CloudTrail 에서 계정관련(생성/삭제) 이벤트가 확인되었습니다. 〈time〉 에 〈eventname〉 의 이벤트 호출이 발생하였고, 요청한 사용자 이름은 〈username〉 이며 요청지 IP 는 〈ip〉 입니다. "

- 위의 가이드 멘트는 예시이므로 상황에 맞게 구성해서 사용합니다.

| 그림 5-7-4 | 대상 입력 변환기 구성

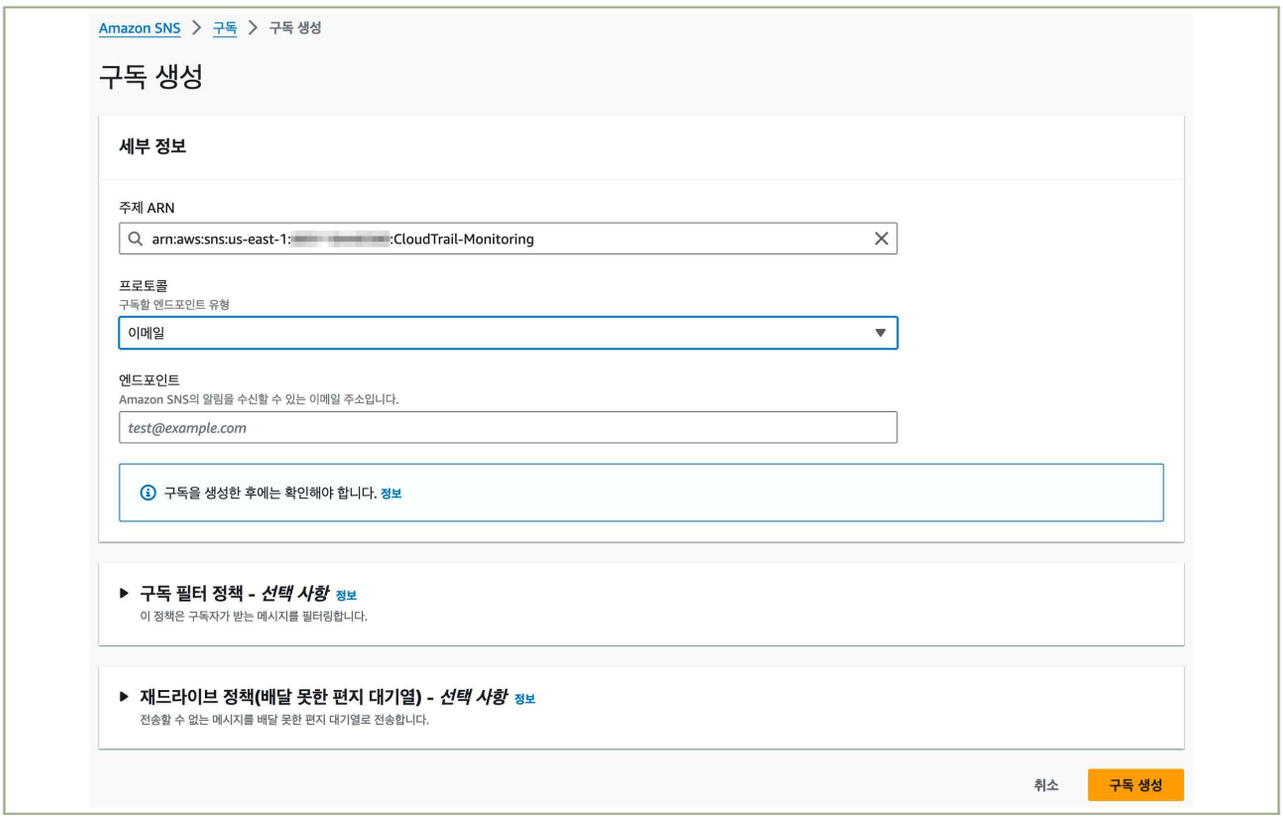
(3) 계정 변경사항 알림 받기

- **(AWS 관리 콘솔)** 알림을 받기 위한 방법은 Amazon SNS 서비스를 이용하며, 이메일 구성을 설정합니다. '콘솔 홈' → 'Amazon SNS' → '주제' → '주제 생성' 을 선택하고 유형은 '표준' 을 선택하며 '이름' 을 넣고 생성합니다.



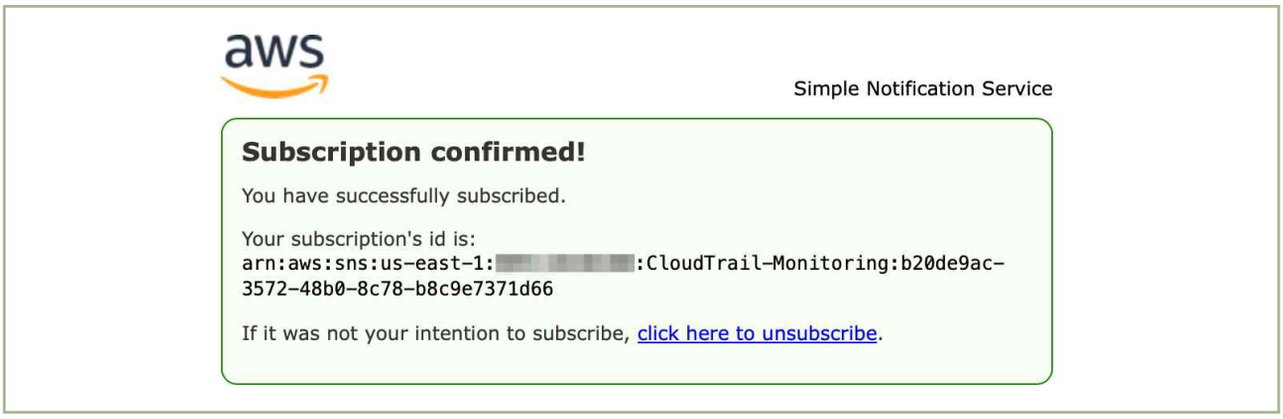
| 그림 5-7-5 | 알림을 전달받기 위한 주제 생성

- 주제를 생성한 후, 해당 주제에 대한 '구독' 을 생성해야 합니다. 주제 ARN 은 앞서 만들었던 '주제' 를 넣어주면 되고, 전송할 프로토콜을 '이메일' 로 선택합니다. 마지막으로 이메일을 전달받을 사용자의 이메일 주소를 '엔드포인트' 에 넣어주고 '구독 생성'을 선택합니다.



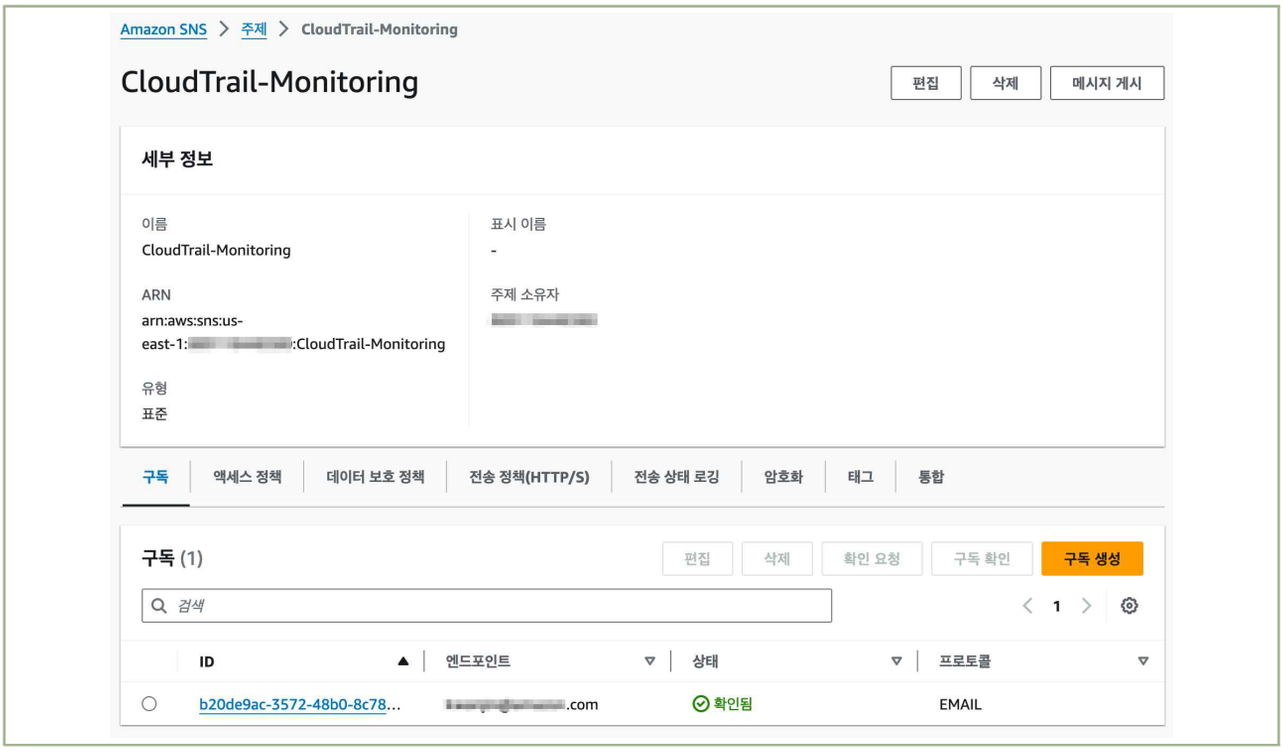
| 그림 5-7-6 | 이메일 알림 구독 생성

- 구독을 생성한 후에는 '엔드포인트' 에 기록해 두었던 이메일 주소로 '구독' 확인을 위한 메일이 발송 됩니다. 전달 받은 메일에서 확인을 하면 다음과 같이 최종적으로 구독이 되었음을 알려 줍니다.



| 그림 5-7-7 | 이메일 알림 구독 성공 화면

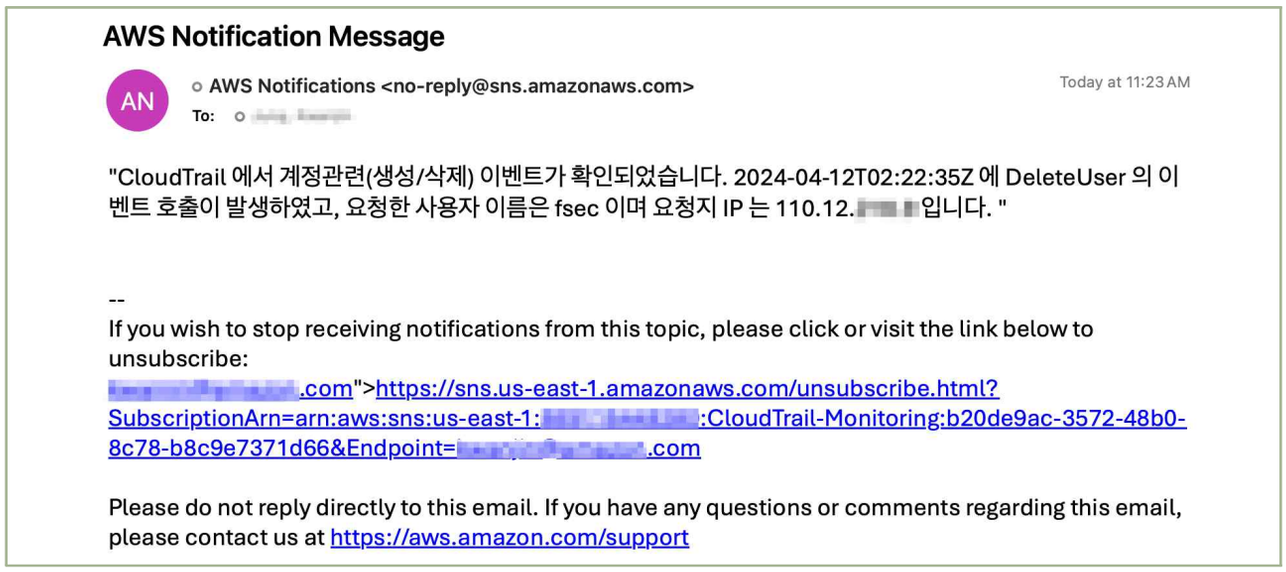
- 최종 구독이 완료되면 구독 상태 화면에서 이메일 주소의 상태가 '확인됨' 으로 변경되어 있음을 확인할 수 있습니다.



| 그림 5-7-8 | 최종 알림 구독 상태 화면

(4) 계정 변경사항 발생 시 알림 통보

- 계정 변경과 관련된 이벤트가 발생했을 때, 설정된 이메일 주소로 관련된 내용이 이메일로 전송되며, 이메일에 기록되는 내용은 사용자가 원하는 문구를 설정하여 사용할 수 있습니다.



| 그림 5-7-9 | 계정 변경사항 이메일 통보

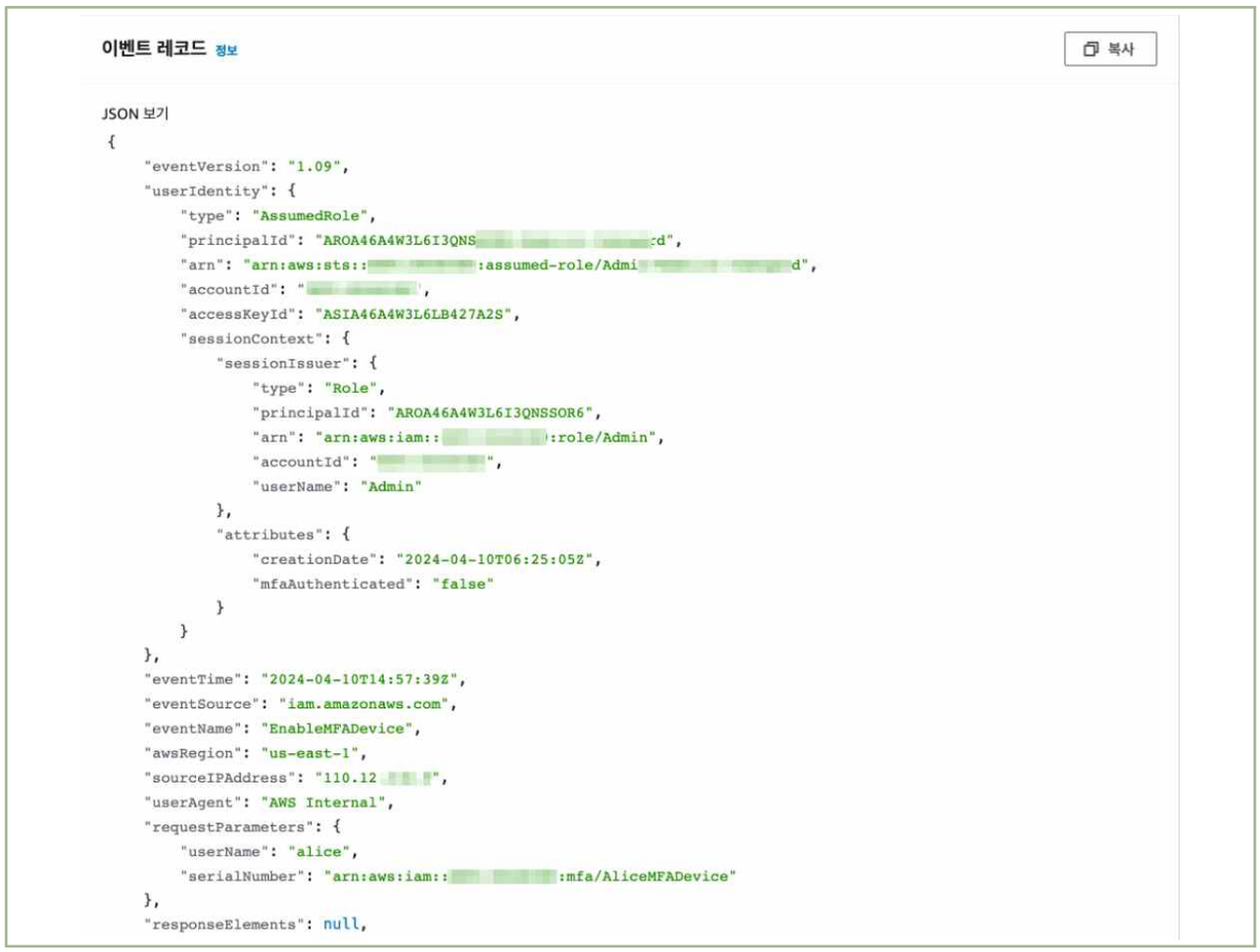
2) 관리자 계정 MFA 수행 이벤트 확인

- 이중확인(MFA) 을 사용하는 경우 'EnableMFADevice' 이벤트가 기록되고 해당 이벤트로 관련 내용을 확인할 수 있습니다.

(1) MFA 적용 이벤트 확인

- MFA를 EnableMFADevice 이벤트가 발생하게되고 일반적으로 MFA 를 통한 인증일 경우 이벤트 레코드 세부정보인 sessionContext 안에 'mfaAuthenticated' 값을 통해 해당 사용자가 MFA 가 적용된 인가인지 확인할 수 있습니다. MFA 를 사용한 경우 해당 값은 'true' 가 됩니다.

| 그림 5-7-10 | CloudTrail에서 MFA 이벤트 확인



| 그림 5-7-11 | 이벤트 상세 레코드

(2) MFA 설정

- MFA 설정은 '3.7 공개용 웹 서버 접근 계정 제한' 의 내용을 참조 바랍니다.

4 참고 사항

6. API 관리

6.1. API 호출 시 인증 수단 적용
6.2. API 호출 시 무결성 검증
6.3. API 호출 시 인증키 보호대책 수립
6.4. API 세션 및 서명 유효기간 적용
6.5. API 호출 구간 암호화 적용

6 API 관리

1 기준

식별번호	기준	내용
6.1.	API 호출 시 인증 수단 적용	클라우드 가상자원 관리를 위한 API 호출 시, 안전한 인증수단을 적용하여 보안성을 강화하여야 한다.

2 설명

- API 호출 시 이용자를 인증할 수 있는 수단을 적용하여야 한다.

  - 예시

    1) API 호출이 가능한 IP 지정

IAM 정책을 통해 Source IP 를 제한하는 설정을 Management Console 과 CLI 를 기반으로 설명

    2) IAM 기능과 연동하여 API를 호출할 수 있는 권한 제어 등

IAM 정책을 통해 권한을 제한하는 설정을 Management Console 과 CLI 를 기반으로 설명

3 우수 사례

- **(AWS 관리 콘솔)** '홈' → 'Identity and Access Management(IAM)' → '정책' → '정책생성'을 클릭
- '권한 지정' 메뉴의 정책 편집기에서 'JSON' 을 선택한 후 아래의 내용을 붙여넣기한 후 다음을 클릭한다.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "*",
```

```
            "Resource": "*",
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "203.0.113.0/24"
                    ]
                }
            }
        }
}
```

[Source IP 제한 정책 생성 화면]

- 다음 화면에서 '정책 이름'을 입력한 후 '정책 생성' 버튼을 클릭한다.
- 생성된 정책은 IAM 사용자 혹은 역할에 부여하여 적용한다.
- **(AWS CLI)** 메모장이나 문서편집기 등을 이용하여 아래의 정책을 별도의 파일로 저장한다. 이 예시에서는 policy.json 이라는 이름으로 저장한다.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "*",
        "Resource": "*",
        "Condition": {
            "NotIpAddress": {
                "aws:SourceIp": [
                    "192.0.2.0/24",
                    "203.0.113.0/24"
                ]
            }
        }
    }
```

- 아래의 명령어를 실행하여 새로운 정책을 생성한다.

```
aws iam create-policy \
    --policy-name DenyAccessExceptIP \
    --policy-document file://policy.json
```

[AWS CLI 를 이용한 IAM 정책 생성]

- 아래의 명령어를 실행하여 생성된 정책을 IAM 사용자나 역할에 부여한다.

  - IAM 사용자에 정책 부여

```
aws iam attach-user-policy \
    --user-name <USER_NAME> \
    --policy-arn <POLICY_ARN>
```

주의. 정상적으로 정책이 할당되면 별도의 내용이 출력되지 않고 프롬프트로 이동한다.

[AWS IAM 사용자에 정책 할당]

  - IAM 역할에 정책 부여

```
aws iam attach-role-policy \
    --role-name <ROLE_NAME> \
    --policy-arn <POLICY_ARN>
```

[AWS IAM 역할에 정책 할당]

4 참고 사항

- AWS IAM 정책 레퍼런스

  - https://docs.aws.amazon.com/cli/latest/reference/iam/attach-user-policy.html

  - https://docs.aws.amazon.com/cli/latest/reference/iam/attach-role-policy.html

[문의처에 관한 정보]

1 기준

식별번호	기준	내용
6.2.	API 호출 시 무결성 검증	**클라우드 가상자원 관리를 위한 API 호출 시, 무결성을 보장하여야 한다.**

2 설명

- API 호출 시 호출 메시지의 무결성을 보장하기 위한 방안을 확보하여야 한다.

  - 예시

    1) 로그인 오류에 따른 보안 통제 방안 수립 등

3 우수 사례

- AWS 의 클라우드 가상 자원 관리를 위해 사용되는 API 는 “AWS Signature Version 4(줄여서 SigV4)” 라고 하는 Signature 를 반드시 AWS API 호출에 인증 정보와 함께 전송하여야 한다. AWS 는 SigV4 를 이용하여 사용자가 전송한 AWS API 의 무결성을 검증하게 되며, AWS SDK 를 이용하거나 AWS 의 관리 콘솔 혹은 AWS CLI 를 이용하는 경우 사용자는 SigV4 와 관련하여 별도의 설정을 하지 않아도 된다.

4 참고 사항

- AWS API 의 Signature 구조 설명 링크

  - https://docs.aws.amazon.com/ko_kr/IAM/latest/UserGuide/reference_sigv.html

[문의처에 관한 정보]

1 기준

식별번호	기준	내용
6.3.	**API 호출 시 인증키 보호대책 수립**	**API 호출 시 인증 키를 안전하게 보관하고 관리할 수 있는 방안을 마련해야 한다.**

2 설명

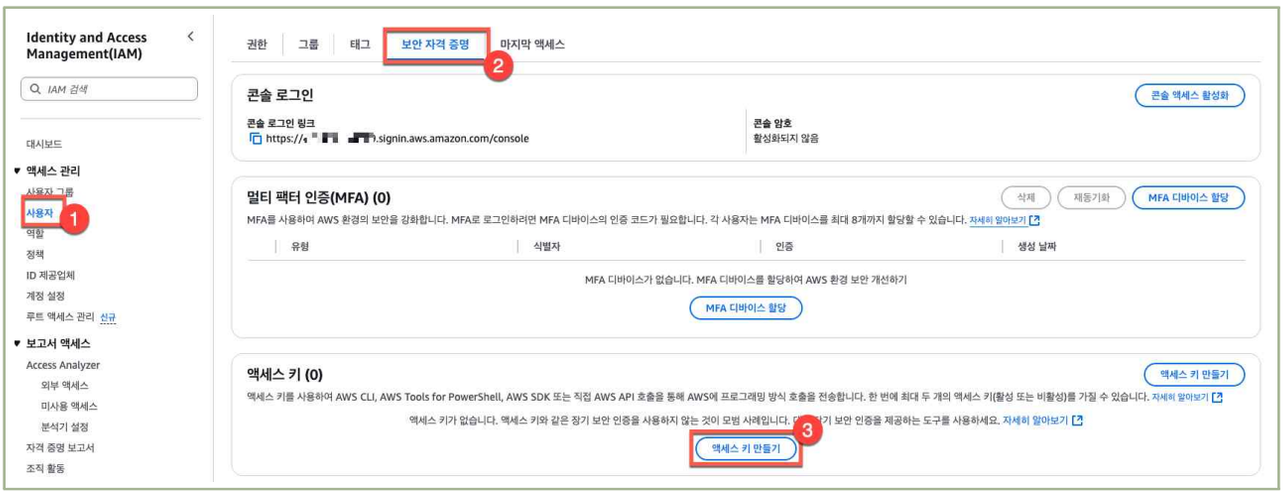
- API 호출 시 인용되는 유니크 값(ex. 보안키 등)은 안전하게 보관 및 관리하여야 한다.

  - 예시

    1) API 보안키 생성 시 이용자에게 1회만 노출 등

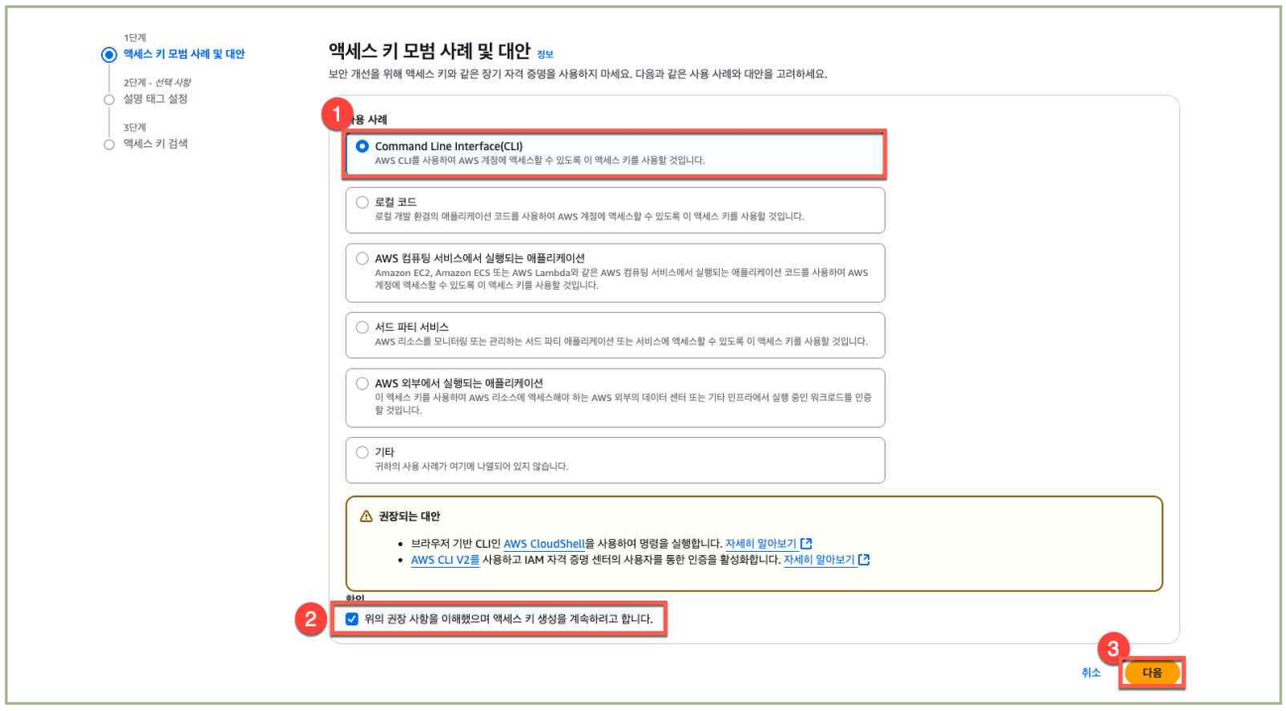
3 우수 사례

- AWS API 호출 시 사용하는 인증키(Access Key, Secret Key)는 AWS IAM 사용자를 사용하는 경우에 한하여 고정적인 인증키가 발급된다. 따라서, 반드시 필요한 경우가 아니라면 가급적 AWS IAM 사용자에서 제공하는 고정적인 인증키를 사용하지 않고 AWS IAM 역할에서 제공하는 임시 인증키(Access Key, Secret Key, Token)을 사용하기를 권고한다.
- **(AWS 관리 콘솔)** AWS IAM 사용자의 고정적인 인증키 발급을 위해 '홈' → 'Identity and Access Management(IAM)' → '사용자' → 고정키 발급을 원하는 사용자를 선택한 후 '보안 자격증명' 클릭 → '액세스 키 만들기' 클릭



| 그림 |

- 액세스 키 모범 사례 및 대안에서 원하는 사용 사례 선택(이 예시에서는 'Command Line Interface(CLI)'선택) → 화면에 출력된 '권장되는 대안' 을 확인한 후 확인에 있는 체크박스 체크 → '다음' 버튼을 클릭

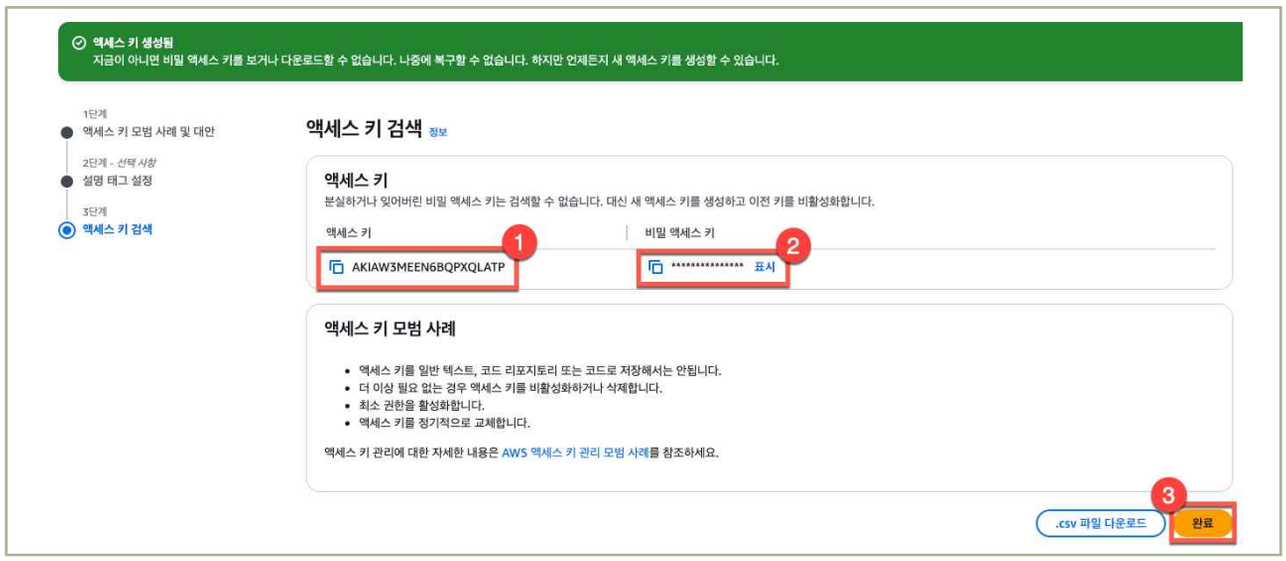


| 그림 |

- 태그 설정 값을 설정하고 태그를 설정하지 않는 경우 '액세스 키 만들기' 버튼 클릭

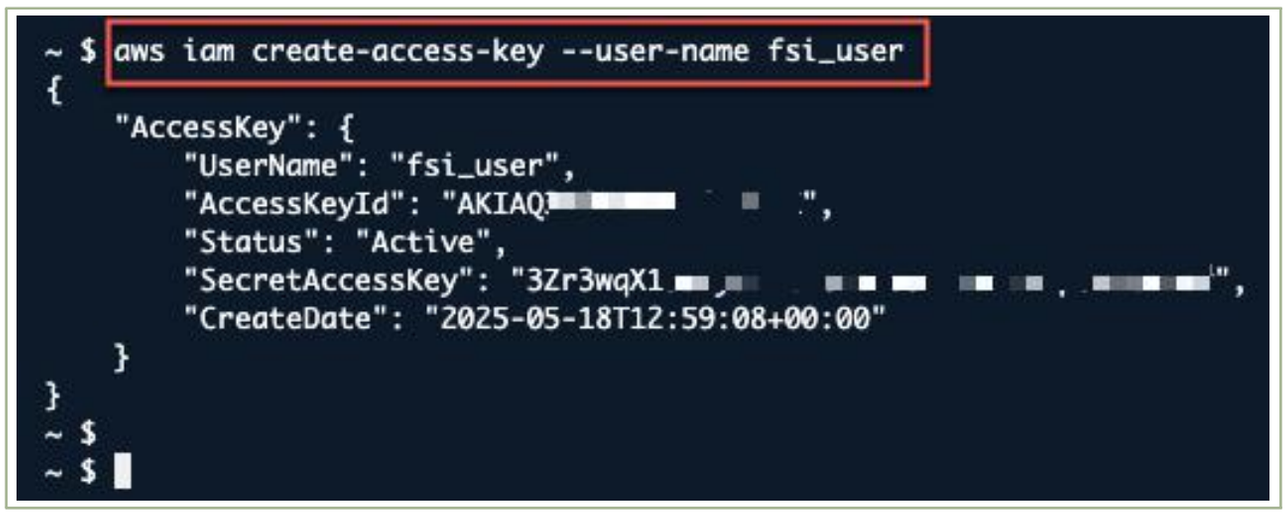
| 그림 |

- 화면에 나타난 '액세스 키' 와 '비밀 액세스 키' 를 복사하여 안전한 곳에 저장한 후 '완료'버튼 클릭



| 그림 |

- **(AWS CLI)** 'aws iam create-access-key --user-name 사용자이름' 명령을 실행하여 액세스키, 보안 액세스키를 생성한다.



| 그림 |

4 참고 사항

[문의처에 관한 정보]

1 기준

식별번호	기준	내용
6.4.	**API 세션 및 서명 유효기간 적용**	**클라우드 가상자원 관리를 위해 API 기능 이용 시 세션 및 서명값에 대한 유효기간을 설정하여야 한다.**

2 설명

- API 기능 이용 시 보안성 향상을 위해 API 세션 및 서명 값에 대한 유효기간을 설정하여야 한다.

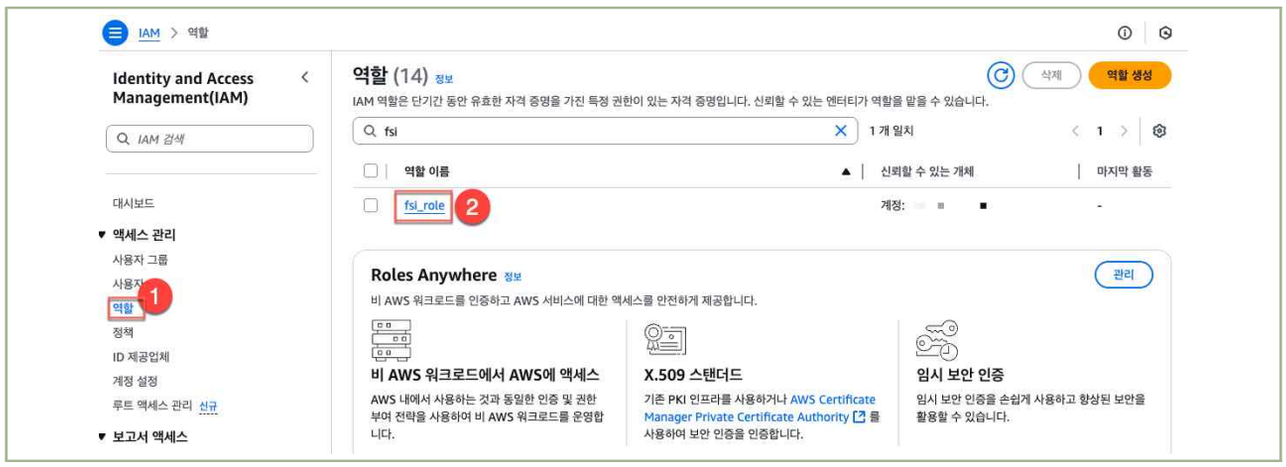
  – 예시

  1) 서명 유효기간 확인

  2) API 호출 세션의 유효기간 설정 기능 적용

3 우수 사례

- **(AWS 관리 콘솔)** AWS IAM 사용자의 고정적인 인증키 발급을 위해 '홈' → 'Identity and Access Management(IAM)' → '역할' 에서 원하는 역할을 선택

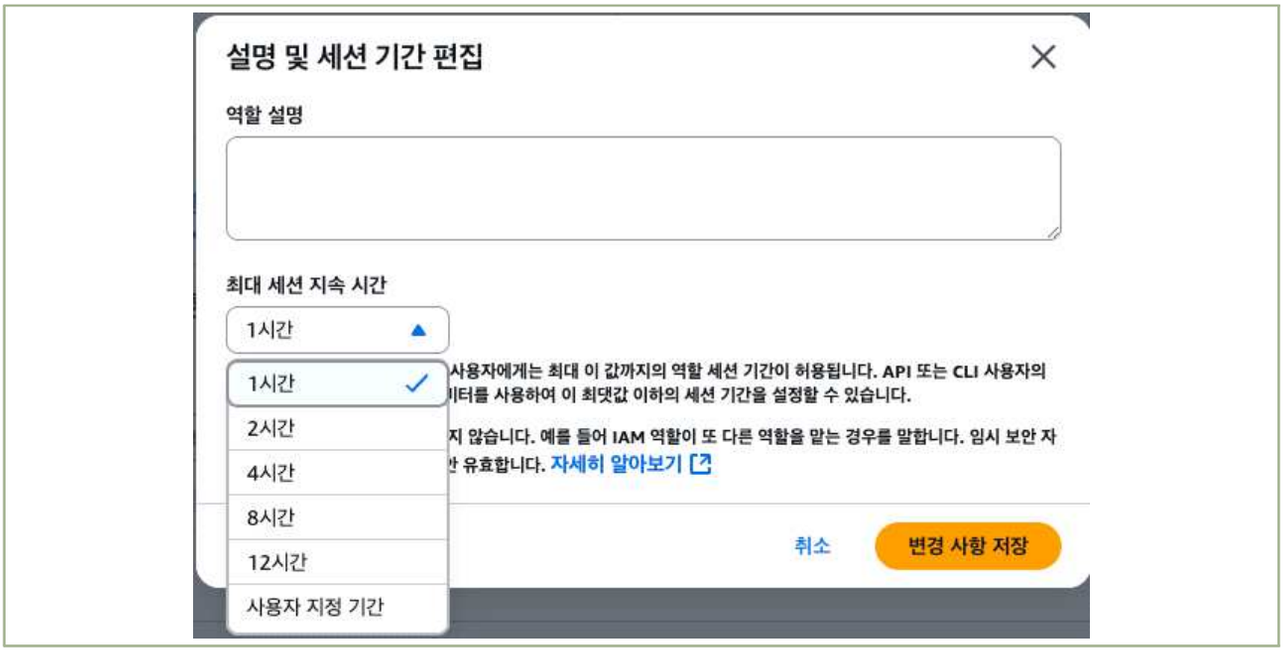


| 그림 |

– 최대 세션 지속 시간을 수정하기 위해 화면 우측 상단의 '편집' 버튼을 클릭

| 그림 |

- 설정하고자 하는 '최대 세션 지속 시간' 을 선택한 후 '변경 사항 저장' 버튼을 클릭하여 설정 저장



| 그림 |

- **(AWS CLI)** 'aws iam update-role –role-name 역할이름 --max-session-duration 최대 세션 지속 시간(초단위)' 명령을 실행하여 지정한 IAM 역할의 최대 세션 지속 시간을 설정한다.

  참고. 실행 명령이 정상적으로 적용되는 경우 별다른 아웃풋이 나타나지 않음

- **(AWS CLI)** 'aws iam get-role –role-name 역할이름' 명령을 사용하여 실행하여 설정한 최대 세션 지속 시간이 정상적으로 반영되었는지 확인한다.

| 그림 |

4 참고 사항

- 역할 설정 업데이트 관련 참고 문서
  - https://docs.aws.amazon.com/ko_kr/IAM/latest/UserGuide/id_roles_update-role-settings.html#

[문의처에 관한 정보]

1 기준

식별번호	기준	내용
6.5.	API 호출 구간 암호화 적용	클라우드 가상자원 관리를 위한 API 호출 시 암호화된 통신구간을 적용하여야 한다.

2 설명

- API를 통한 클라우드 가상자원 관리 수행 시 네트워크 트래픽 보호를 위한 암호화된 통신구간을 적용하여야 한다.

  - 예시

    1) SSL 적용 등

3 우수 사례

1) AWS 서비스의 API 호출

- AWS 서비스는 통신을 위해 TLS(Transport Layer Security)를 사용하는 HTTPS 엔드포인트를 제공하여 AWS API와 통신할 때 전송 중 암호화를 기본적으로 제공
- AWS API 호출을 사용하는 경우 최소 TLS 1.2 이상 사용
- TLS 1.3 을 사용할 것을 권장
- 제공되는 서비스의 통신 암호화를 확인하는 경우 다음과 같이 curl 명령어를 통해 확인하여 TLS 버전 및 암호화 스위트 확인 가능

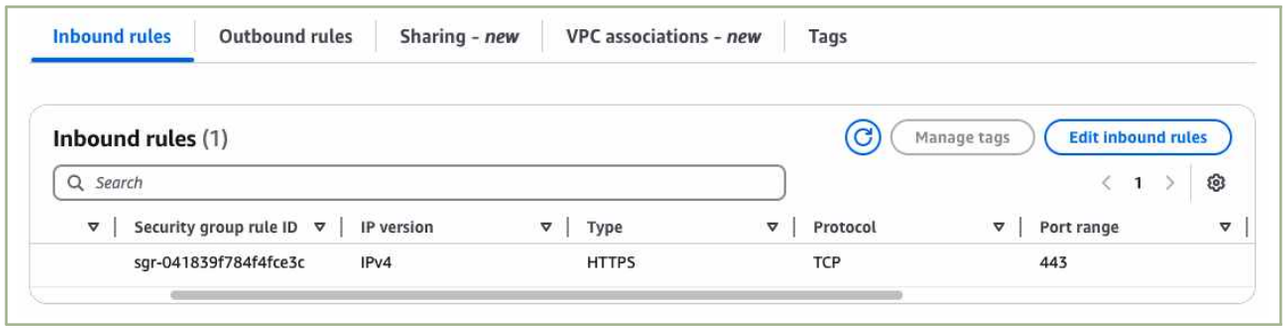
[AWS 서비스 TLS 지원 확인]

```
# curl -v https://s3.ap-northeast-2.amazonaws.com
```

- AWS SDK 의 boto3 는 기본적으로 최신 TLS 를 지원
- API 호출 시 기본적으로 서명 버전 4(Signature Version 4)사용하여 요청 암호화
- 필요 시 AWS Private Link를 활용하여 VPC 내부에서 AWS 서비스로의 Private 통신 구현 및 인터넷 경유 없이 AWS 네트워크 내에서 암호화된 통신이 가능

- **(보안 그룹을 활용해 비암호화 접근 통제)** 보안 그룹을 사용하여 가상 프라이빗 클라우드(VPC)에서 안전하지 않은 HTTP 프로토콜을 차단하고 HTTPS 만 허용

[보안 그룹을 활용해 HTTPS 포트 접근 허용]



4 참고 사항

- **(IAM 권한을 통해 암호화만 허용)** IAM 정책 작성 시 “aws:SecureTransport” 값을 True로 적용하여 암호화된 트래픽만으로 접근 통제

[IAM 정책의 Condition 예]

```
"Condition": {
        "Bool": {
          "aws:SecureTransport": "true"
        }
      }
```

- AWS 서비스 엔드포인트 확인
  https://docs.aws.amazon.com/general/latest/gr/aws-service-information.html
- AWS Well-Architected Framework 의 SEC09-BP02 Enforce encryption in transit 내용 참고
  (https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/sec_protect_data_transit_encrypt.html)

7. 스토리지 관리

7.1. 스토리지 접근 관리
7.2. 스토리지 권한 관리
7.3. 스토리지 업로드 파일 제한

7 스토리지 관리

1 기준

식별번호	기준	내용
7.1.	**스토리지 접근 관리**	**스토리지 목적에 따라 외부 공개 차단 등 적절한 접근통제를 수행하여야 한다.**

2 설명

- 스토리지 목적에 따라 외부 공개 차단 등 적절한 접근통제를 수행 하여야 한다.

  - 예시

    1) 외부 공개가 불필요한 경우, 스토리지 퍼블릭 엑세스 차단

    2) 스토리지에 접근 가능한 계정(IAM) 적용

    3) URL로 접근 시 접근 가능한 시간, 별도 IP 지정 등

3 우수 사례

1) 외부 공개가 불필요한 경우, 스토리지 퍼블릭 엑세스 차단

- Amazon S3에서 외부의 접근이 필요하지 않은 경우 퍼블릭 엑세스 차단을 설정하여 스토리지의 데이터를 외부로부터 안전하게 지킬 수 있다.
- **(AWS 관리 콘솔)** '홈' → 'Amazon S3' → '버용버킷' → '해당 버킷' → '권한' 에서, 퍼블릭 액세스 차단 설정 확인 가능

[Amazon AWS 관리 콘솔]

- **(기본 설정)** 기본적으로 새 버킷은 퍼블릭 액세스를 허용하지 않는다. 해당 기능을 사용하면 계정 관리자와 버킷 소유자가 리소스 생성 방식에 관계없이 적용되는 Amazon S3 리소스에 대한 퍼블릭 액세스를 제한하는 중앙 집중식 제어를 쉽게 설정 가능하다.

2) 스토리지에 접근 가능한 계정(IAM 적용)

- Amazon S3에 접근하기 위해 IAM 정책을 통해 권한을 부여해 주어야 한다.

(1) **(S3 접근 권한 부여)** S3에 접근하기 위해 사용자에게 필요한 S3 권한을 부여해 주어야 한다. 접근할 S3 버킷과 허용할 권한을 할당하며, 주요 권한은 아래 도표에서 참고할 수 있다.

[사용자 S3 권한 할당 정책 예제]

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::버킷명/*",
                "arn:aws:s3:::버킷명"
            ]
        }
    ]
}
```

[주요 S3 권한]

S3 권한	설명
s3:GetObject	객체에 대한 읽기 권한
s3:PutObject	객체에 대한 업로드 권한
s3:DeleteObject	객체에 대한 삭제 권한
s3:ListBucket	S3 버킷 리스트 권한
s3:CreateBucket	S3 버킷 생성 권한
s3:DeleteBucket	S3 버킷 삭제 권한

- **(사전 구성된 S3 IAM 정책)** AWS 에서 제공하는 관리형 정책을 사용하여 권한을 부여할 수도 있다. 다만, 세부적으로 조절하지 못하므로 세부적인 권한 설정이 필요한 경우 할당할 권한과 접근할 리소스(S3) 에 대해서 명확히 지정하는 것이 권장된다.

  - AmazonS3ReadOnlyAccess : 읽기 전용 접근

  - AmazonS3FullAccess : 모든 권한 접근

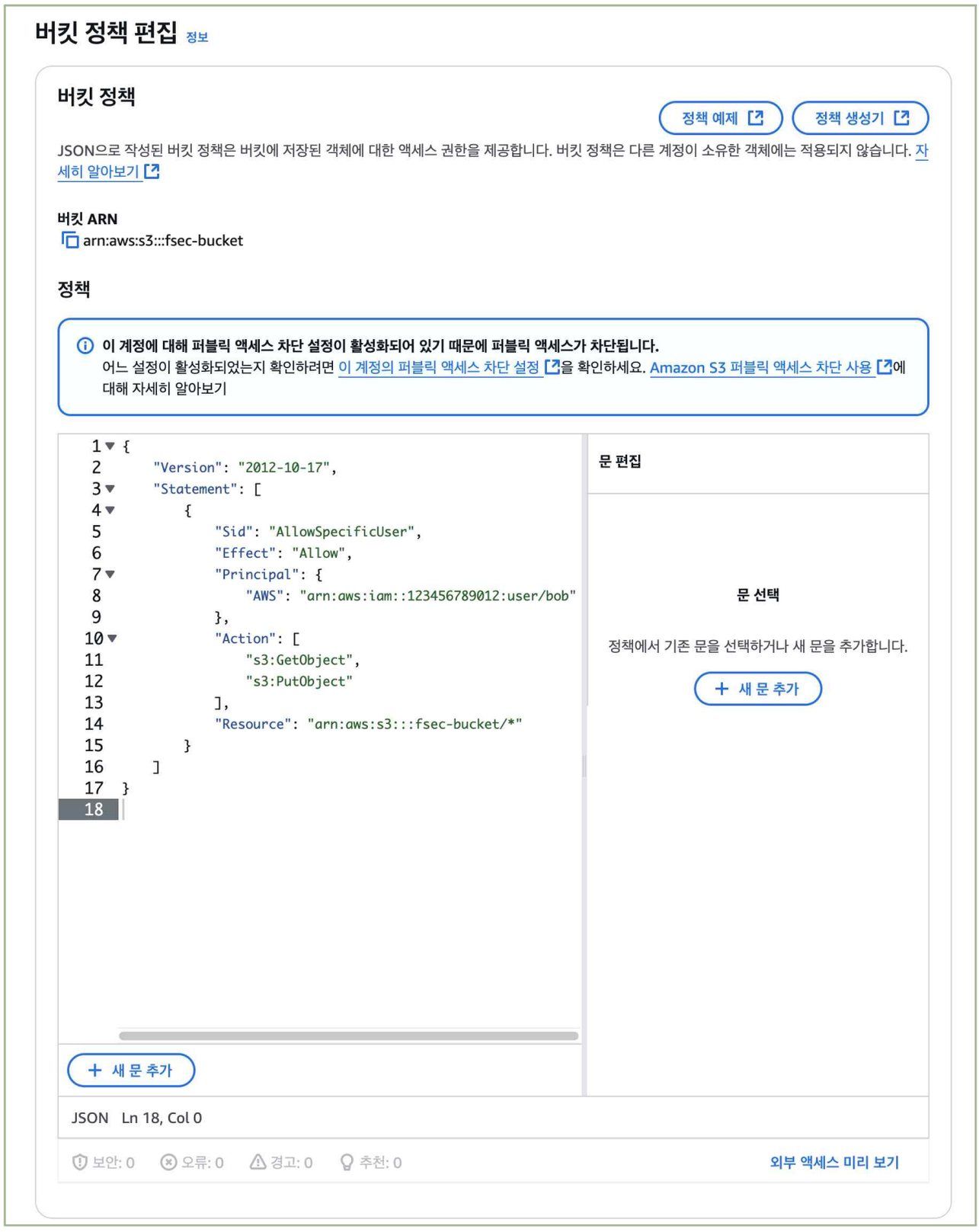
(2) **(AWS 관리 콘솔)** 사용자에게 접근 권한을 주는 것과 다르게 버킷 자체에도 정책을 설정하여 버킷별 접근 권한을 설정할 수 있다.

- **(AWS 관리 콘솔)** '홈' → 'Amazon S3' → '범용 버킷' → '해당 버킷' → '버킷 정책 편집' 에서 접근할 계정에 대해서 정책을 추가해 주면 된다.

[S3 버킷 정책 예]

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSpecificUser",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/사용자계정"
            },
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::버킷명/*"
        }
    ]
}
```

[S3 버킷 정책 편집 화면]


버킷 정책 편집 정보
버킷 정책
정책 예제
정책 생성기
JSON으로 작성된 버킷 정책은 버킷에 저장된 객체에 대한 액세스 권한을 제공합니다. 버킷 정책은 다른 계정이 소유한 객체에는 적용되지 않습니다. 자세히 알아보기
버킷 ARN
arn:aws:s3:::fsec-bucket
정책
이 계정에 대해 퍼블릭 액세스 차단 설정이 활성화되어 있기 때문에 퍼블릭 액세스가 차단됩니다.
어느 설정이 활성화되었는지 확인하려면 이 계정의 퍼블릭 액세스 차단 설정을 확인하세요. Amazon S3 퍼블릭 액세스 차단 사용에 대해 자세히 알아보기
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "AllowSpecificUser",
"Effect": "Allow",
"Principal": {
"AWS": "arn:aws:iam::123456789012:user/bob"
},
"Action": [
"s3:GetObject",
"s3:PutObject"
],
"Resource": "arn:aws:s3:::fsec-bucket/*"
}
]
}
문 편집
문 선택
정책에서 기존 문을 선택하거나 새 문을 추가합니다.
새 문 추가
새 문 추가
JSON Ln 18, Col 0
보안: 0
오류: 0
경고: 0
추천: 0
외부 액세스 미리 보기

- **(IAM 정책 평가)** AWS에서는 명시적 거부(Explicit Deny)가 항상 허용(Allow)보다 우선하며, 리소스 기반 정책(예: S3 버킷 정책)과 자격 증명 기반 정책(IAM 정책)이 모두 적용될 경우, 둘 중 하나라도 작업을 허용하면 해당 작업이 허용된다. 하지만 리소스 정책에서 'Deny'를 설정하면, IAM 정책에서 'Allow'를 설정하더라도 접근이 거부된다.

3) URL로 접근 시 접근 가능한 시간, 별도 IP 지정 등

- 외부 접근을 통해 액세스 하는 경우 시간 및 IP 지정으로 접근 제한을 세부적으로 제어할 수 있다.

(1) **(버킷 IP 제한)** 접근할 버킷이 퍼블릭으로 설정되어 있을 경우, 다음과 같이 버킷 정책을 통해 접근할 IP 를 제어할 수 있다.

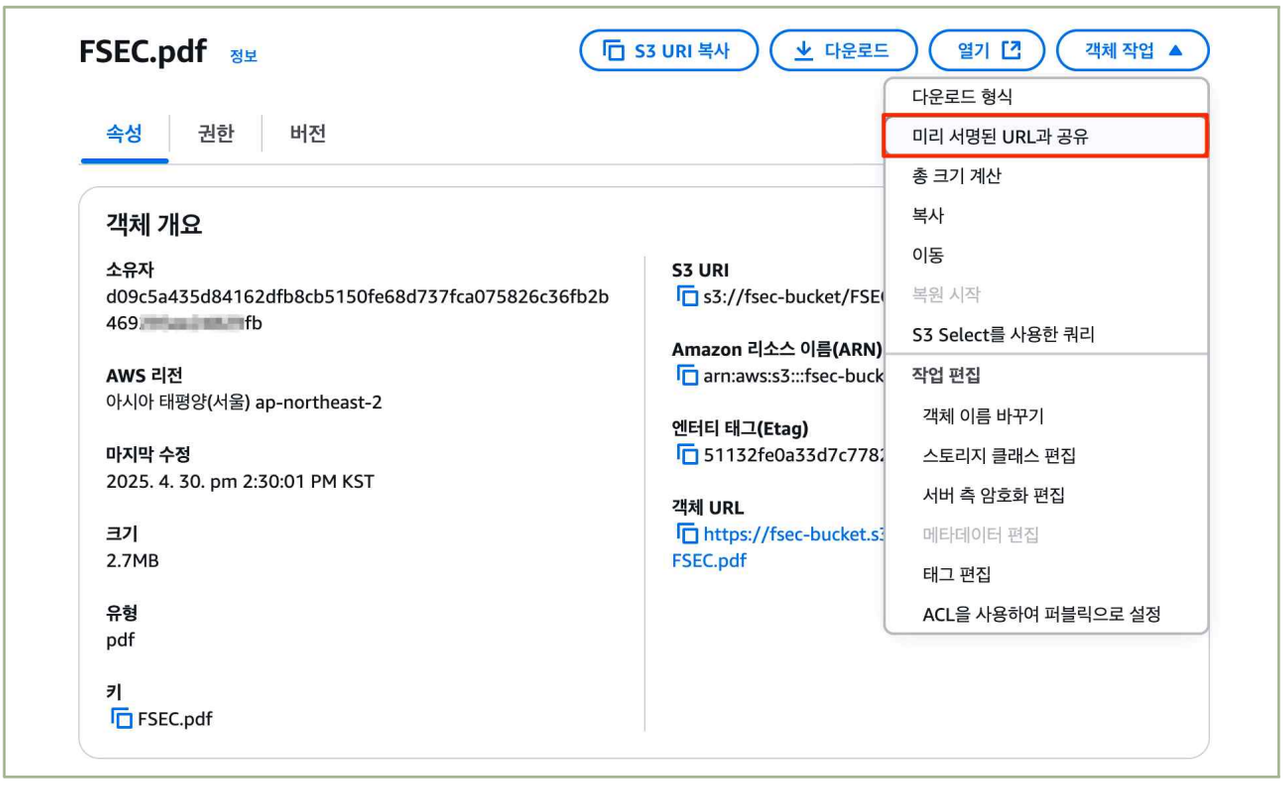
[S3 버킷 정책 예]

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadWithIPRestriction",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::버킷명/*",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": [
                        "x.x.x.x/24"
                    ]
                }
            }
        }
    ]
}
```

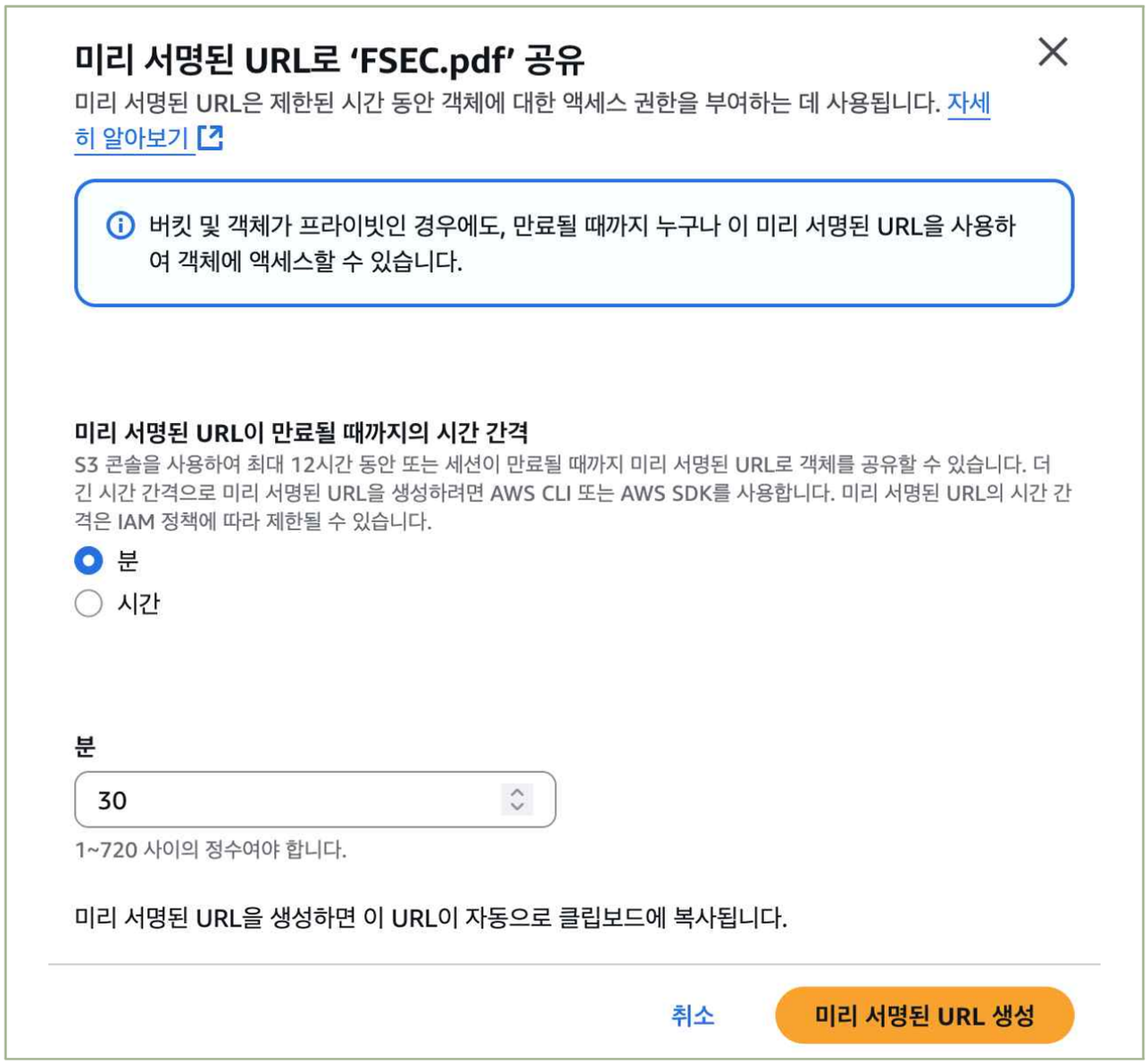
(2) **(Presigned URL)** 미리 서명된 URL을 사용하여 제한된 시간 동안 객체에 액세스 권한을 부여한다.

- **(AWS 관리 콘솔)** '홈' → 'Amazon S3' → '범용 버킷' → '해당 버킷' → '해당 오브젝트 선택' → '객체 작업' → '미리 서명된 URL과 공유' 를 통해 Presigned URL 을 생성한다.

[오브젝트에서 '객체작업'을 통해 Presigned URL 생성]



[만료되는 Presigned URL 생성]



- **(AWS CLI)** AWS CLI 를 이용하여 S3 URI 에 대해 손쉽게 Presigned URL 을 생성할 수 있다.

[CLI 기반으로 Presigned URL 생성]

```
% aws s3 presign s3://fsec-bucket/FSEC.pdf --expires-in 3600
https://fsec-bucket.s3.amazonaws.com/FSEC.pdf?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA46A4[삭제]PVU%2F20250529%2Fap-northeast-2%2Fs3%2Faws4_request&X-Amz-Date=20250529T103709Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=7d09563f6793700630cd58ada341926249e8373e10826d616fe6e0fdfb84d549
```

4 참고 사항

- AWS Organizations 을 사용하는 경우 S3 각 버킷마다 정책을 설정하기 보다 리소스 전체에 정책을 반영할 수 있다. RCP(Resource Control Policies) 을 활용하면 조직의 계정에 있는 리소스에 대한 기본 권한을 정의할 수 있다.

1 기준

식별번호	기준	내용
7.2.	**스토리지 권한 관리**	**스토리지 목적에 따라 읽기, 쓰기 등 권한을 세분화하여 적용하고 관리하여야 한다.**

2 설명

- 스토리지 목적에 따라 읽기, 쓰기 등 권한을 세분화하여 적용하고 관리하여야 한다.

  - 예시

    1) 스토리지 객체 권한(읽기, 쓰기 등)을 목적에 따라 적용

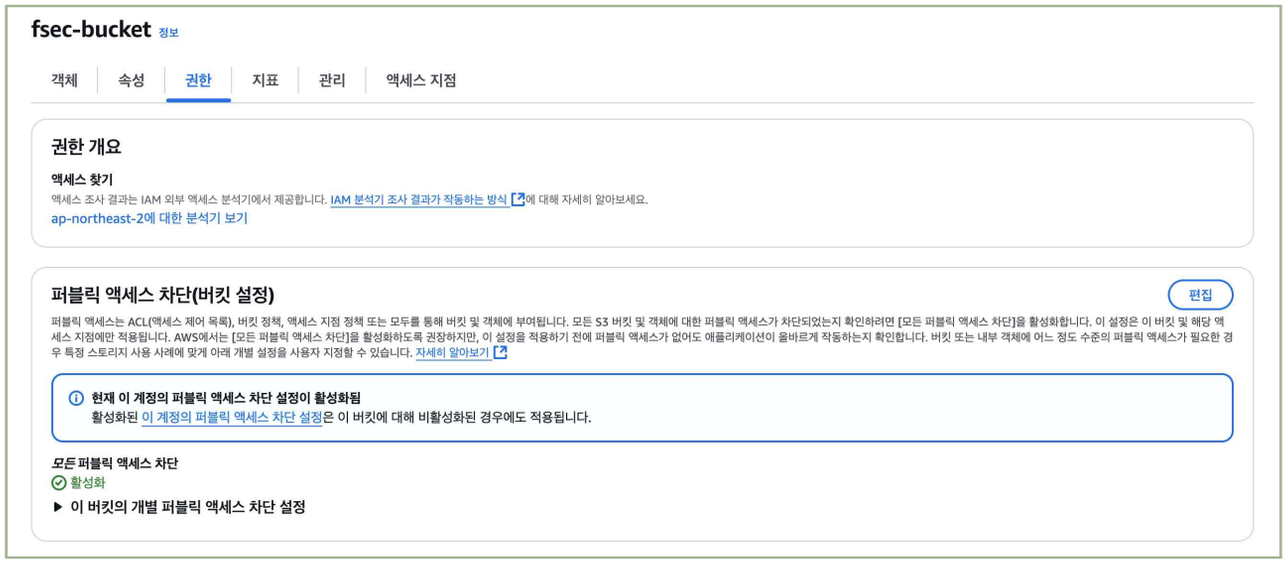
    2) 스토리지 권한 부여 현황에 대해 주기적인 검토 수행 등

3 우수 사례

1) 스토리지 객체 권한(읽기, 쓰기 등)을 목적에 따라 적용

- Amazon S3 에서 스토리지 객체의 권한을 읽기와 쓰기 형태로 제한하려면 IAM 정책, 버킷 정책 또는 ACL(Access Control List)을 사용하여 목적에 맞는 권한을 설정할 수 있다.
- **(AWS 관리 콘솔)** ‘홈’ → ‘Amazon S3’ → ‘General purpose buckets’ -〉 ‘해당 버킷’ -〉 ‘Permissions’ 에서, 권한과 관련한 설정 확인 가능

[Amazon S3 버킷 퍼미션]



(1) (IAM 정책을 사용한 권한 제한) IAM 정책은 특정 사용자, 그룹 또는 역할에 대해 S3 객체의 읽기와 쓰기 권한을 세부적으로 정의할 수 있음

[특정 버킷 내 객체를 읽기만 허용하는 정책]

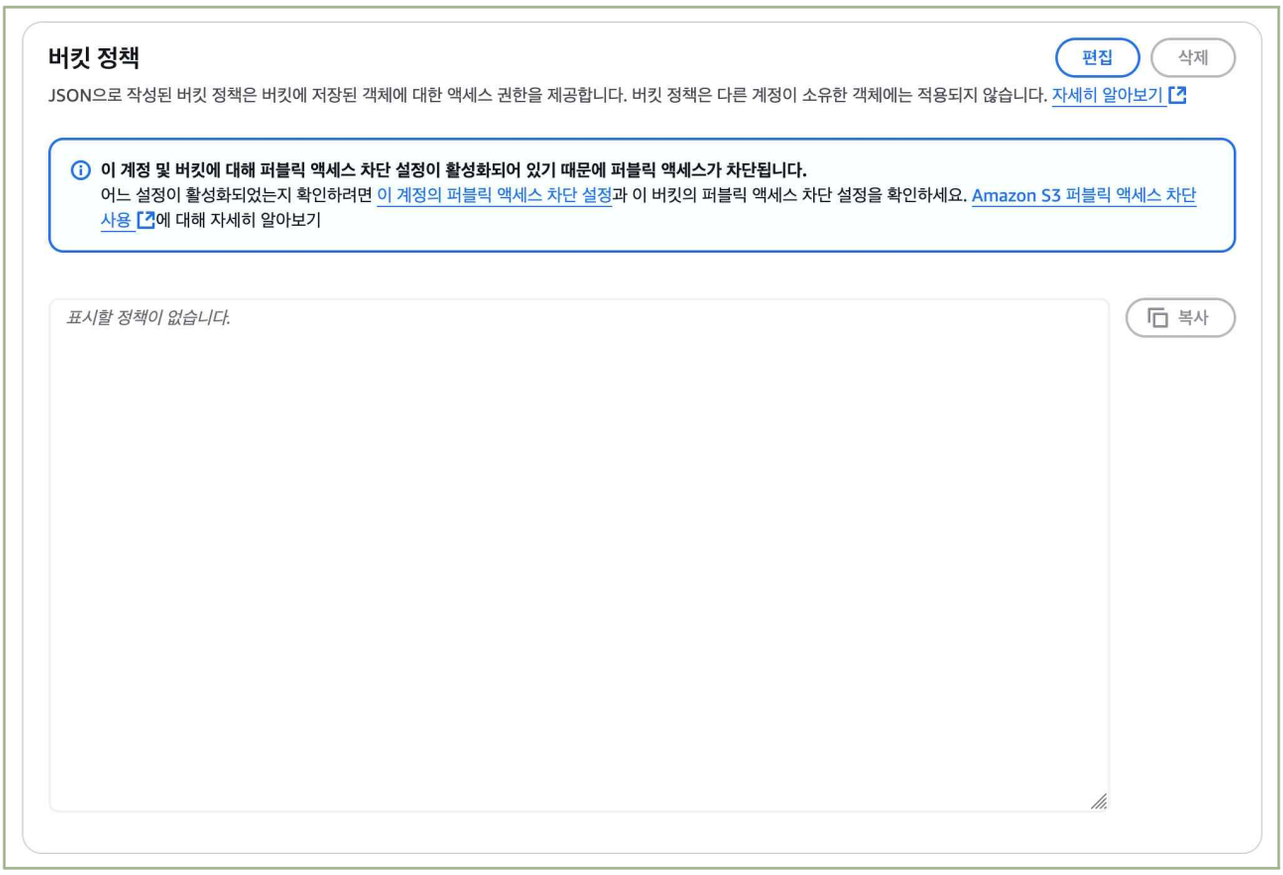
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:GetObject"],
      "Resource": ["arn:aws:s3:::example-bucket/*"]
    }
  ]
}
```

[특정 버킷 내 객체를 업로드하거나 수정만 허용하는 정책]

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:PutObject"],
      "Resource": ["arn:aws:s3:::example-bucket/*"]
    }
  ]
}
```

(2) (버킷 정책을 사용한 권한 제한) ‘Amazon S3’ →’버킷’ →’권한’ → ‘버킷 정책’에서 리소스 정책 생성

[Amazon S3 버킷 정책 설정]



- **(읽기/쓰기 버킷 정책)** 버킷 정책은 특정 사용자, 계정 또는 IP 주소에 대해 버킷 및 객체의 읽기/쓰기 권한을 설정

[특정 IP 주소에서만 읽기 접근 허용]

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": ["s3:GetObject"],
      "Resource": ["arn:aws:s3:::example-bucket/*"],
      "Condition": {
        "IpAddress": {"aws:SourceIp": "x.x.x.x/24"}
      }
    }
  ]
}
```

[특정 AWS 계정에서만 쓰기 접근 허용]

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {"AWS": ["arn:aws:iam::123456789012:root"]},
      "Action": ["s3:PutObject"],
      "Resource": ["arn:aws:s3:::example-bucket/*"]
    }
  ]
}
```

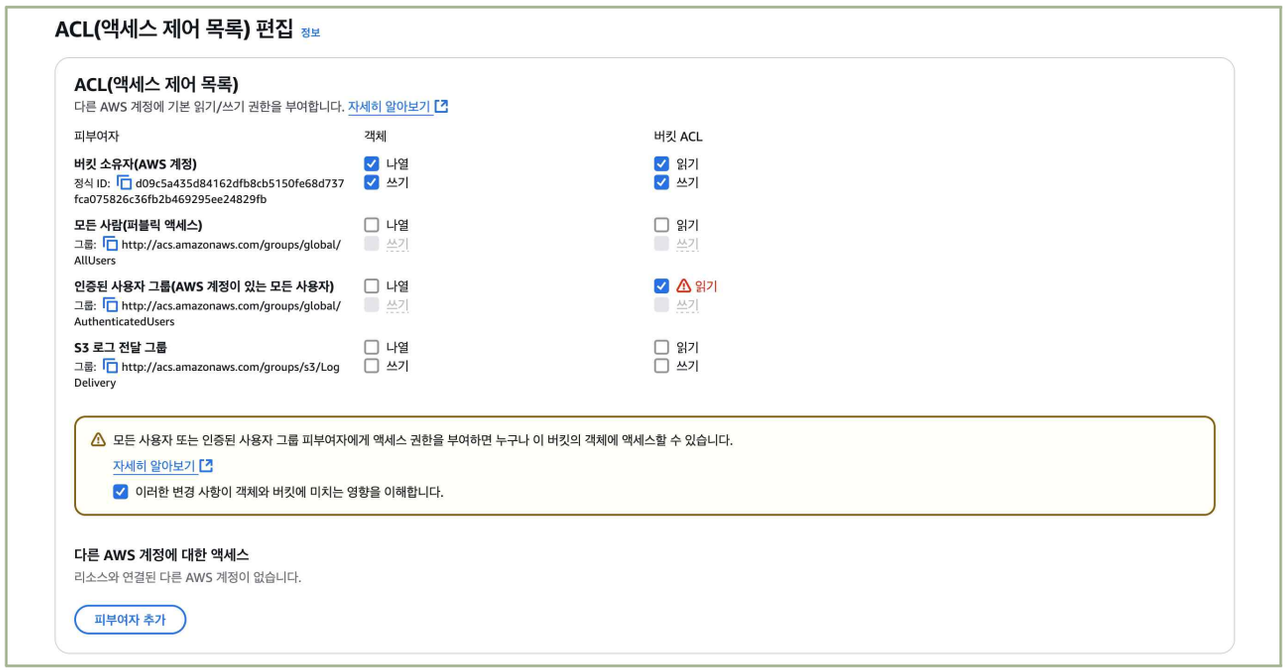
(3) (버킷 정책을 사용한 권한 제한) ‘Amazon S3’ →’버킷’ →’권한’ → ‘ACL(액세스 제어 목록)’에서 권한을 제어한다. ACL 은 객체 단위로 읽기 및 쓰기 권한을 부여하거나 제한할 수 있다. 하지만 더 이상 권장되지 않는 방법이며, IAM 이나 버킷 정책을 사용하는 것을 권장한다.

- **(기본 설정)** 객체의 소유권이 버킷 소유자로 설정되어 있으며, 모든 ACL 은 Disabled 되어 있다. ACL 편집을 사용하기 위해서는 ‘객체 소유권 편집’ 에서 ‘ACL 활성화됨’ 을 선택해 주어야 한다.

[객체 소유권 편집]

- **(ACL 편집)** ‘편집’ 버튼을 클릭하여 필요한 권한을 부여한다.

[ACL(액세스 제어 목록) 편집]



(4) (조건 기반 접근 제어) 권한을 더욱 세분화하기 위해 조건 키를 사용할 수 있다.

- **(조건 키)** 아래와 같은 조건키를 사용하여 세부적으로 통제할 수 있다.

조건 키	설명
aws:SourceIP	요청이 발생한 IP 주소 기준으로 접근 제어
aws:PrincipalTag	IAM 사용자나 역할에 지정된 태그 기반 접근 제어
aws:RequestedRegion	특정 AWS 리전으로 접근 제한
aws:MultiFactorAuthPresent	MFA 인증 여부로 접근 제어
aws:ResourceTag	리소스에 지정된 태그 기반 접근 제어

- 다양한 조건 키는 IAM 문서를 참고

[HTTPS 접근 차단]

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": ["s3:*"],
      "Resource": ["arn:aws:s3:::example-bucket/*"],
      "Condition": {"Bool": {"aws:SecureTransport": false}}
    }
  ]
}
```

[특정 경로(Prefix) 에서만 접근 허용]

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:GetObject", "s3:PutObject"],
      "Resource": ["arn:aws:s3:::example-bucket/fsec-folder/*"]
    }
  ]
}
```

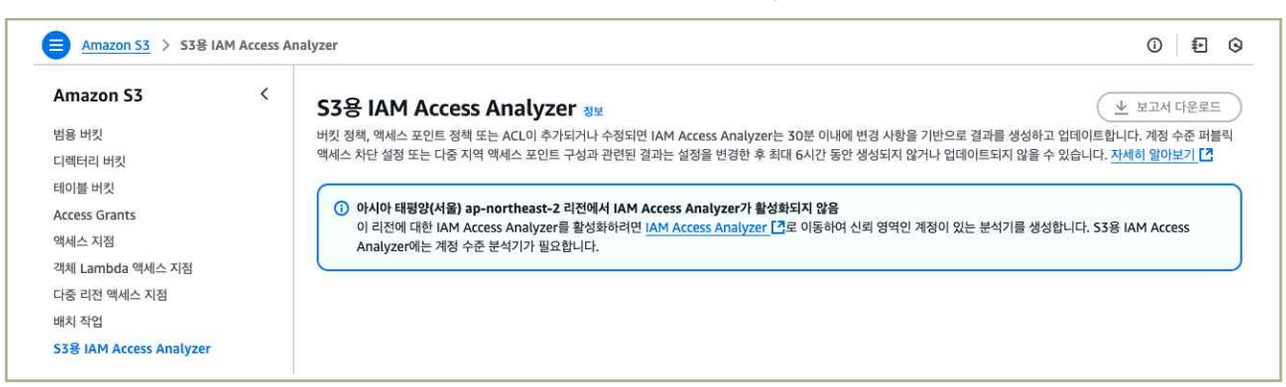
2) 스토리지 권한 부여 현황에 대해 주기적인 검토 수행 등

- Amazon S3 에서 IAM Access Analyzer 및 AWS Config 등을 활용하여 주기적으로 권한 부여 현황에 대해 검토할 수 있다.

(1) (IAM Access Analyzer 를 이용한 검토) IAM Access Analyzer for S3 는 조직 외부의 계정을 포함하여 인터넷사의 모든 사용자 또는 기타 AWS 계정에 대한 액세스를 허용하도록 구성된 S3 버킷을 검토할 수 있다.

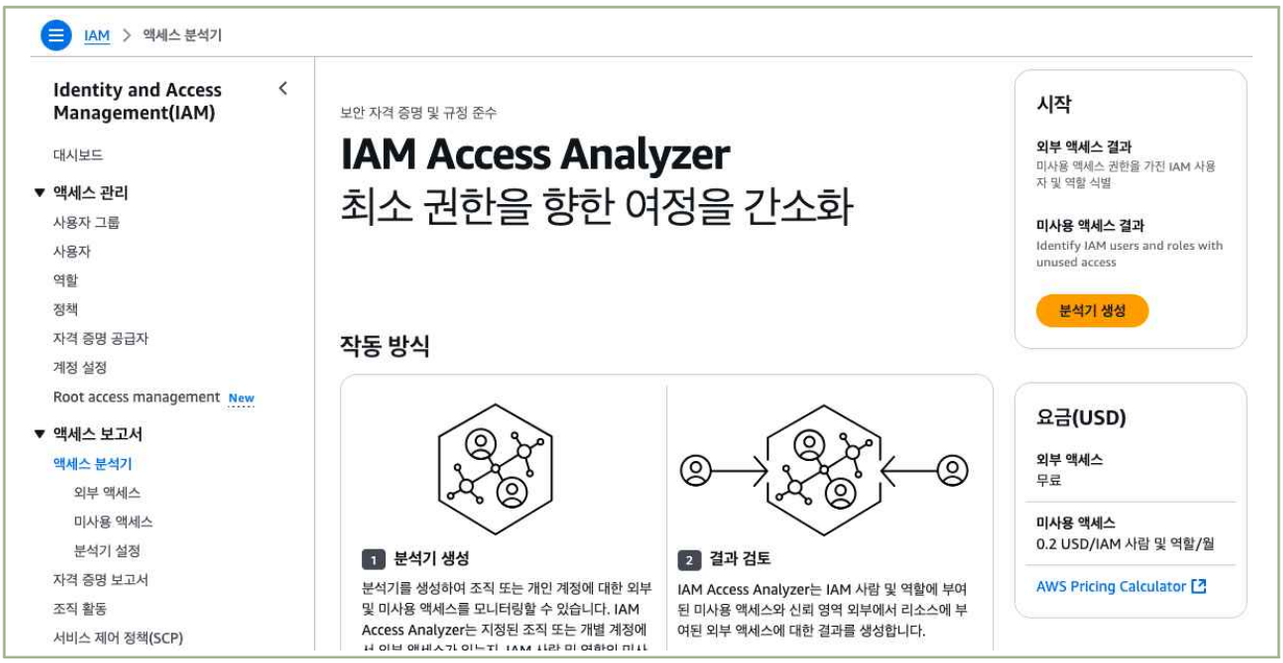
- **(동작 설정 확인)** ‘Amazon S3’ → ‘IAM Access Analyzer for S3’ 에서 확인할 수 있다.

[S3용 IAM Access Analyzer]



- **(분석기 생성)** 동작이 필요한 경우 ‘Identity and Access Management(IAM)’ → ‘액세스 보고서’ → ‘액세스 분석기’ → ‘분석기 생성’ 을 클릭하여 생성해 준다.

[IAM Access Analyzer 분석기 생성]



[CLI 를 통해 IAM Access Analyzer 실행]

```
$ aws accessanalyzer start-policy-generation
```

- **(보고서 확인)** 보고서를 다운로드 받아 버킷별 상태를 확인할 수 있다. 'S3용 IAM Access Analyzer' → '보고서 다운로드' 를 클릭하여 내려 받을 수 있다.

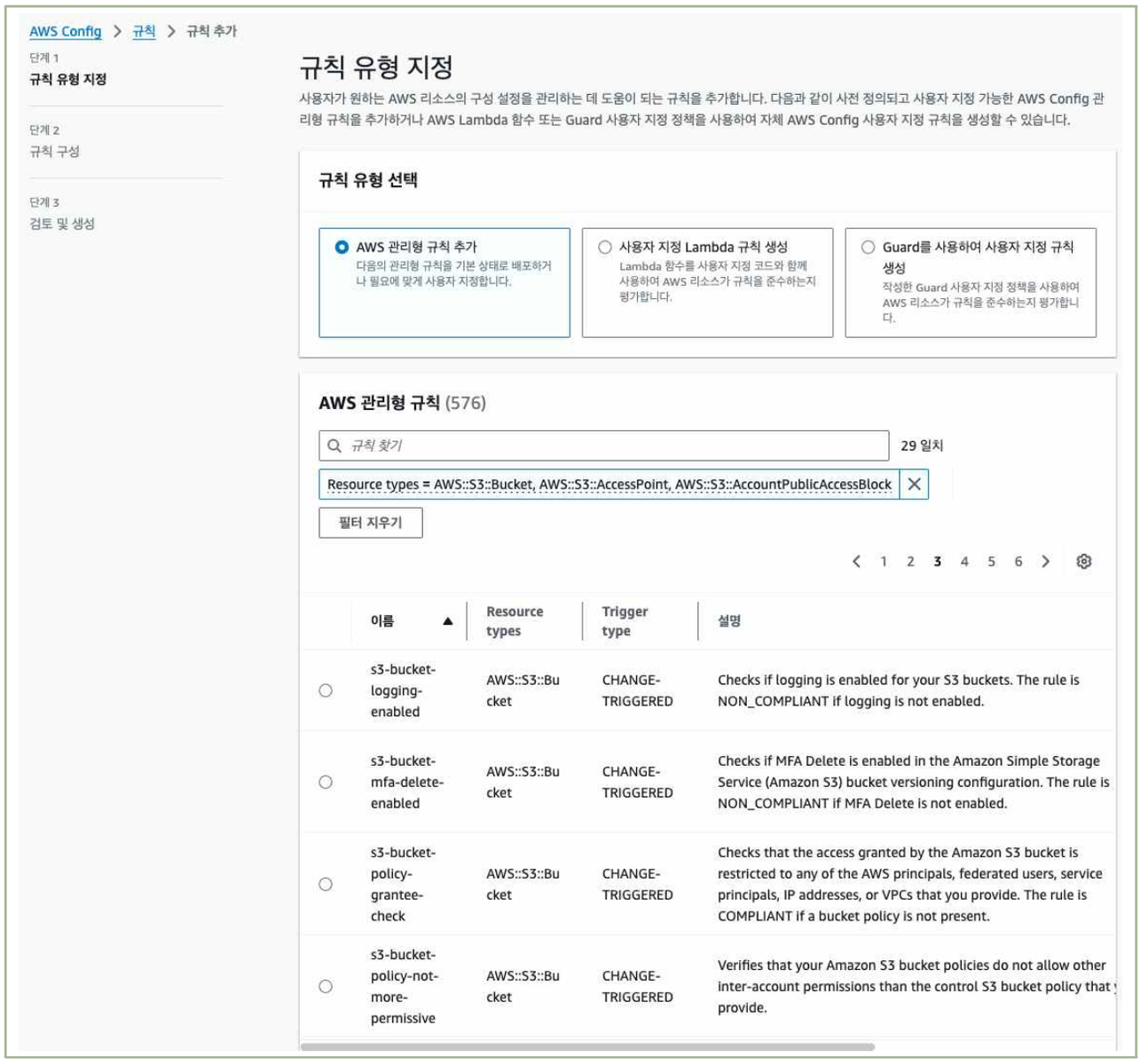
[CSV 로 제공되는 보고서]

access-analyzer-report(ap-northeast-2)2025-04-29

Arn	Sharing	Analyzed	Discovered	SharedThrough	Actions
arn:aws:s3:::access	Private	2025-04-29 07:28:50 +09:00			
arn:aws:s3:::amplify	Private	2025-04-29 07:28:50 +09:00			
arn:aws:s3:::aws-at	Private	2025-04-29 07:28:50 +09:00			
arn:aws:s3:::aws-at	Private	2025-04-29 07:28:50 +09:00			
arn:aws:s3:::aws-gl	Private	2025-04-29 07:28:50 +09:00			
arn:aws:s3:::cf-temp	Private	2025-04-29 07:28:50 +09:00			
arn:aws:s3:::codepi	Private	2025-04-29 07:28:50 +09:00			
arn:aws:s3:::config-	Private	2025-04-29 07:28:50 +09:00			
arn:aws:s3:::config-	Private	2025-04-29 07:28:50 +09:00			
arn:aws:s3:::detecti	Private	2025-04-29 07:28:50 +09:00			
arn:aws:s3:::detecti	Private	2025-04-29 07:28:50 +09:00			
arn:aws:s3:::detecti	Private	2025-04-29 07:28:50 +09:00			
arn:aws:s3:::fsec-bucket	Private	2025-04-29 07:28:50 +09:00			

(2) (AWS Config) 실시간 또는 주기적으로 규정 준수 유무를 체크할 수 있다. 'AWS Config' → 'Rules' → 'Add rule' 에서 필요한 규칙을 선택하여 적용할 수 있다.

[AWS Config 규칙 추가]



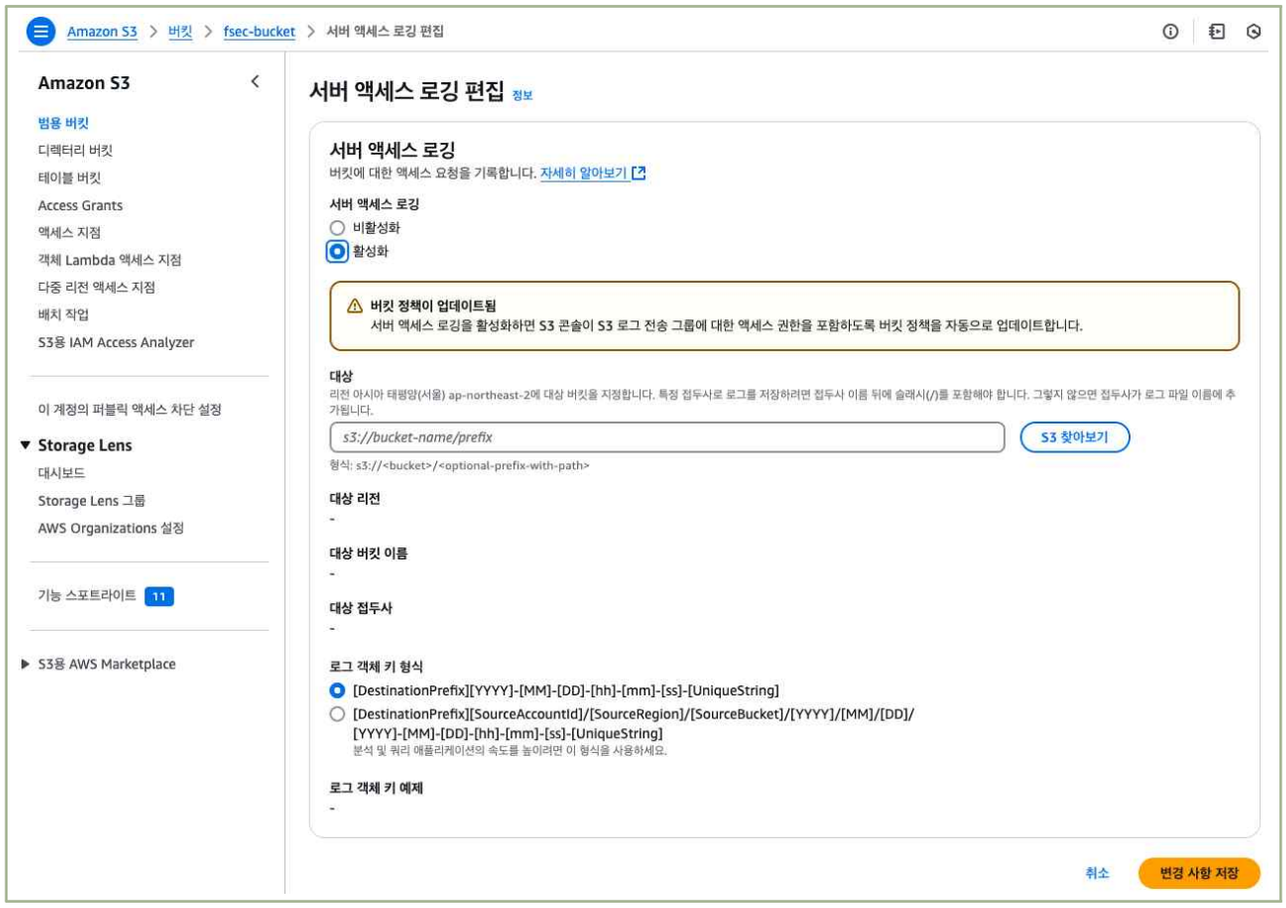
[S3 관련 주요 권한 관련 규칙]

Rule 이름	설명
s3-account-level-public-access-blocks	어카운트 레벨에서 공개접근차단 설정이 되어 있는지 확인
s3-bucket-level-public-access-prohibited	S3 버킷이 퍼블릭하게 접근 가능한지 체크
s3-bucket-public-read-prohibited	S3 버킷이 퍼블릭하게 읽기 접근이 허용되어 있지 않은지 체크
s3-bucket-public-write-prohibited	S3 버킷이 퍼블릭하게 쓰기 접근이 허용되어 있지 않은지 체크

- 위에서 언급된 룰 외에 더욱 많은 규칙을 ‘규칙 추가’ 화면에서 확인해 볼 수 있다.

(3) (S3 로깅 기록) 버킷의 액세스 요청을 기록하여 해당 정보를 기반으로 접근 권한 현황을 검토할 수 있다. ‘Amazon S3’ → 로깅 기록이 필요한 ‘버킷’ → ‘속성’ → ‘서버 액세스 로깅’을 통해 활성화를 할 수 있다.

[서버 액세스 로깅 편집]



- 기록된 서버 액세스 로깅 정보를 기반으로 권한 거부된 요청, 특정 사용자/역할의 접근 패턴, 비정상적인 접근 패턴 탐지를 수행할 수 있다.

4 참고 사항

- Amazon Athena 를 이용하면 S3 에 기록된 로그 데이터에 대해 SQL 쿼리 형태로 빠르게 조회해 볼 수 있다.

1 기준

식별번호	기준	내용
7.3.	**스토리지 업로드 파일 제한**	**스토리지 목적에 맞는 안전한 파일만 업로드 될 수 있도록 보호대책을 마련하여야 한다.**

2 설명

- 스토리지 목적에 맞는 파일만 업로드 될 수 있도록 업로드 가능한 파일을 제한하여야 한다.

  - 예시

    1) 스토리지 버킷 정책 설정을 통한 업로드 파일 확장자 제한 등

    2) 금융회사에서 스토리지 내 파일 업로드 시 확장자 등을 검증할 수 있는 절차 마련

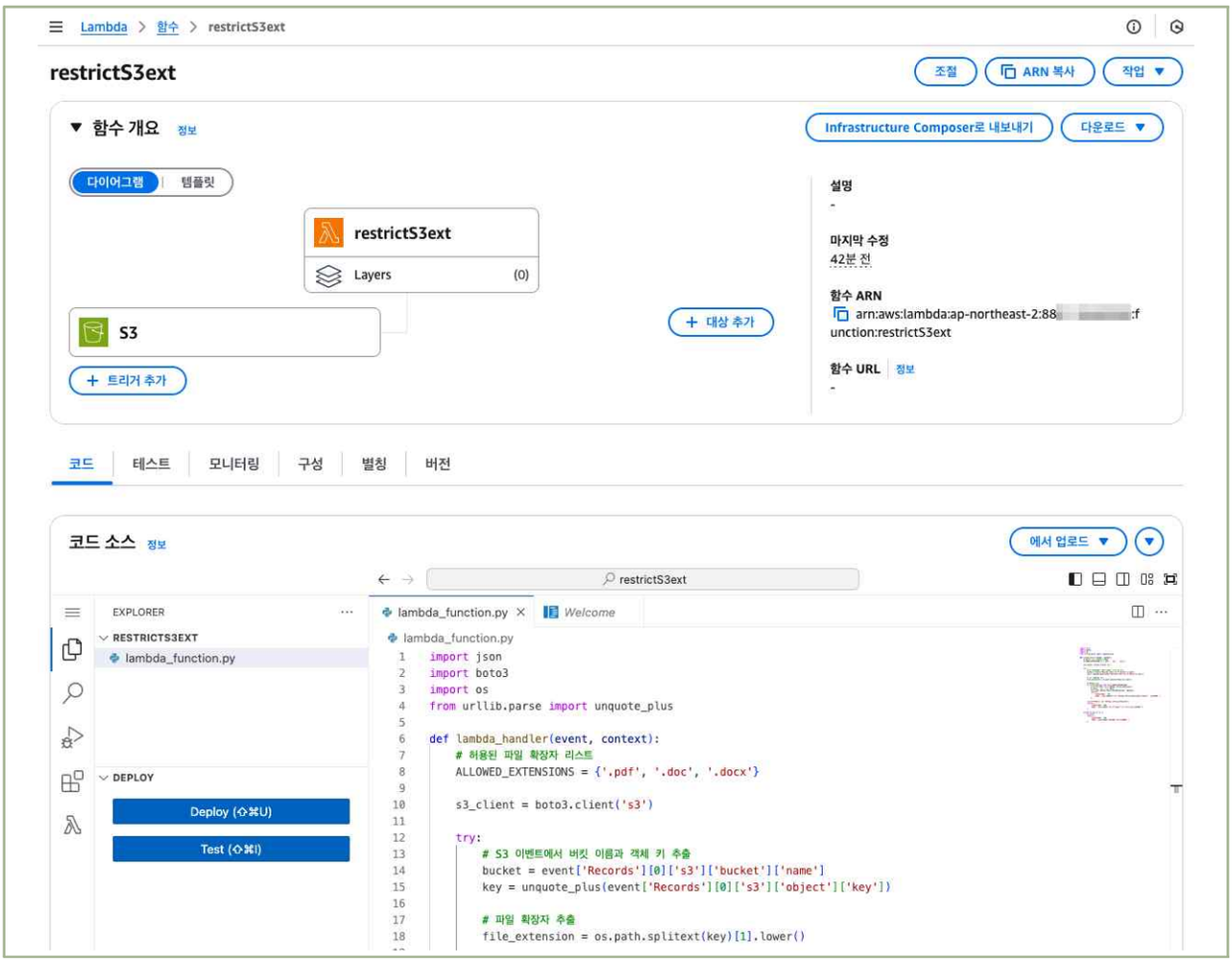
3 우수 사례

1) 스토리지 버킷 정책 설정을 통합 업로드 파일 확장자 제한 등

- Amazon S3 에서는 기본적으로 버킷에 업로드되는 파일의 확장자를 직접적으로 제어할 수 있는 설정은 제공하지 않는다.
- **(Lambda 코드를 활용한 제한)** 특정 확장자만 허용되도록 버킷에서 이벤트를 트리거 하고 확장자를 판단할 수 있는 람다 코드를 호출하여 제한하는 방법을 검토해 볼 수 있다.

(1) (Lambda 활용) ‘Lambda’ → ‘함수’ → ‘함수 생성’ 을 통해 Lambda 함수를 생성하며 다음의 코드를 사용한다.

[Lambda 함수 생성]



- **(확장자 제어)** 예제 샘플은 pdf, doc, docx 확장자만 허용하는 코드이며, ALLOWED_EXTENSIONS에 확장자를 추가하여 사용할 수 있다. 다만 확장자 기반이므로, 필요에 따라 코드를 더 정교하게 수정하여 확장자를 제어할 것을 권장한다.

[확장자 제어하는 Lambda 샘플 코드]

```
import json
import boto3
import os
from urllib.parse import unquote_plus

def lambda_handler(event, context):
    # 허용된 파일 확장자 리스트
    ALLOWED_EXTENSIONS = {'.pdf', '.doc', '.docx'}

    s3_client = boto3.client('s3')

    try:
```

```
        # S3 이벤트에서 버킷 이름과 객체 키 추출
        bucket = event['Records'][0]['s3']['bucket']['name']
        key = unquote_plus(event['Records'][0]['s3']['object']['key'])

        # 파일 확장자 추출
        file_extension = os.path.splitext(key)[1].lower()

        # 확장자 검사
        if file_extension not in ALLOWED_EXTENSIONS:
            print(f"불허된 파일 확장자: {file_extension}")
            # 허용되지 않은 파일 삭제
            s3_client.delete_object(Bucket=bucket, Key=key)
            return {
                'statusCode': 403,
                'body': json.dumps(f'파일 확장자 {file_extension}는 허용되지 않습니다.')
            }

        print(f"허용된 파일 확장자: {file_extension}")
        return {
            'statusCode': 200,
            'body': json.dumps('파일이 성공적으로 업로드되었습니다.')
        }

    except Exception as e:
        print(e)
        return {
            'statusCode': 500,
            'body': json.dumps('오류가 발생했습니다.')
        }
```

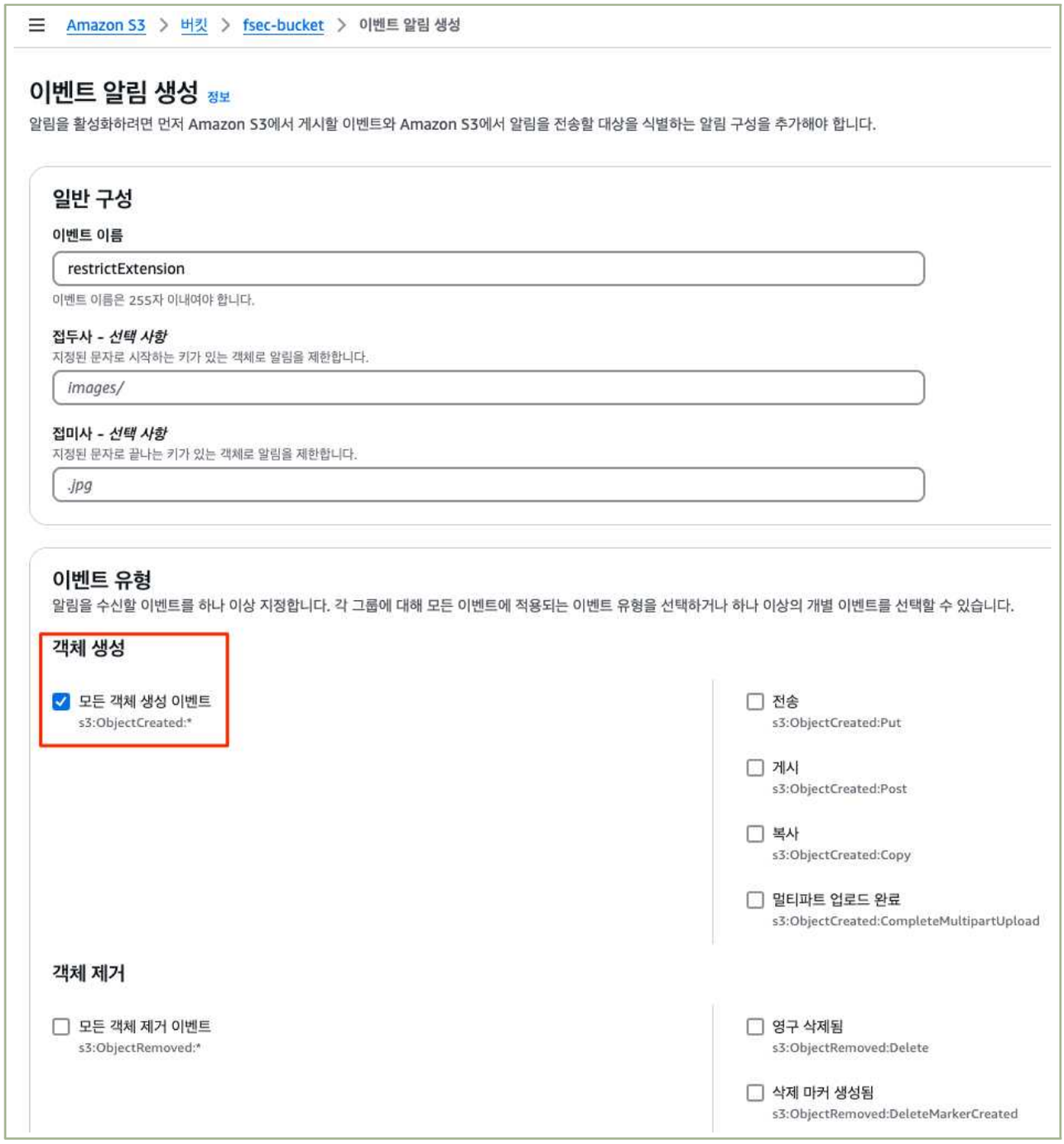
- **(Lambda 권한 부여)** Lambda 가 실행할 때 필요한 권한을 정의해 준다.

[Lambda 에서 사용할 실행 역할]

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:DeleteObject"
            ],
            "Resource": "arn:aws:s3:::your-bucket-name/*"
        }
    ]
}
```

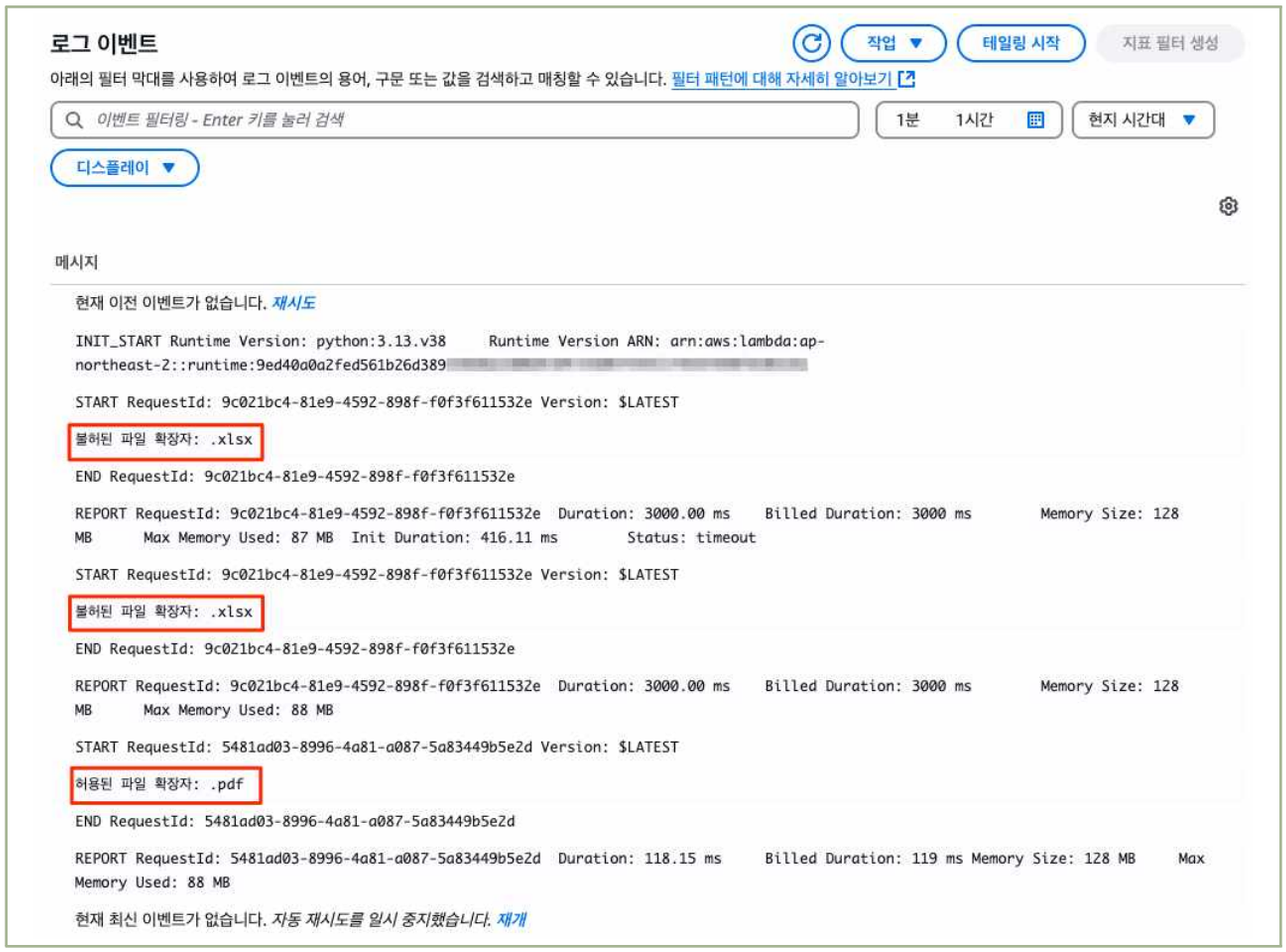
- **(S3 이벤트 트리거)** 버킷에서 '이벤트 알림'을 설정하여 오브젝트가 생성될 때 Lambda 함수가 호출되도록 설정해 준다. 'Amazon S3' → '버킷' 선택→ '속성' → '이벤트 알림' 에서 생성해 준다. 이벤트 유형을 선택해 주고, 대상에서 Lambda 함수를 선택해 앞서 만든 Lambda 함수를 선택한다.

[객체 생성 이벤트에 대한 알림 생성]



- **(Lambda 실행 로그)** Lambda 에서 허용한 확장자만 해당 버킷에 파일이 올라가게 되고, 허용되지 않은 확장자는 파일이 삭제되게 된다.

[CloudWatch 를 통해 본 Lambda 실행 로그]

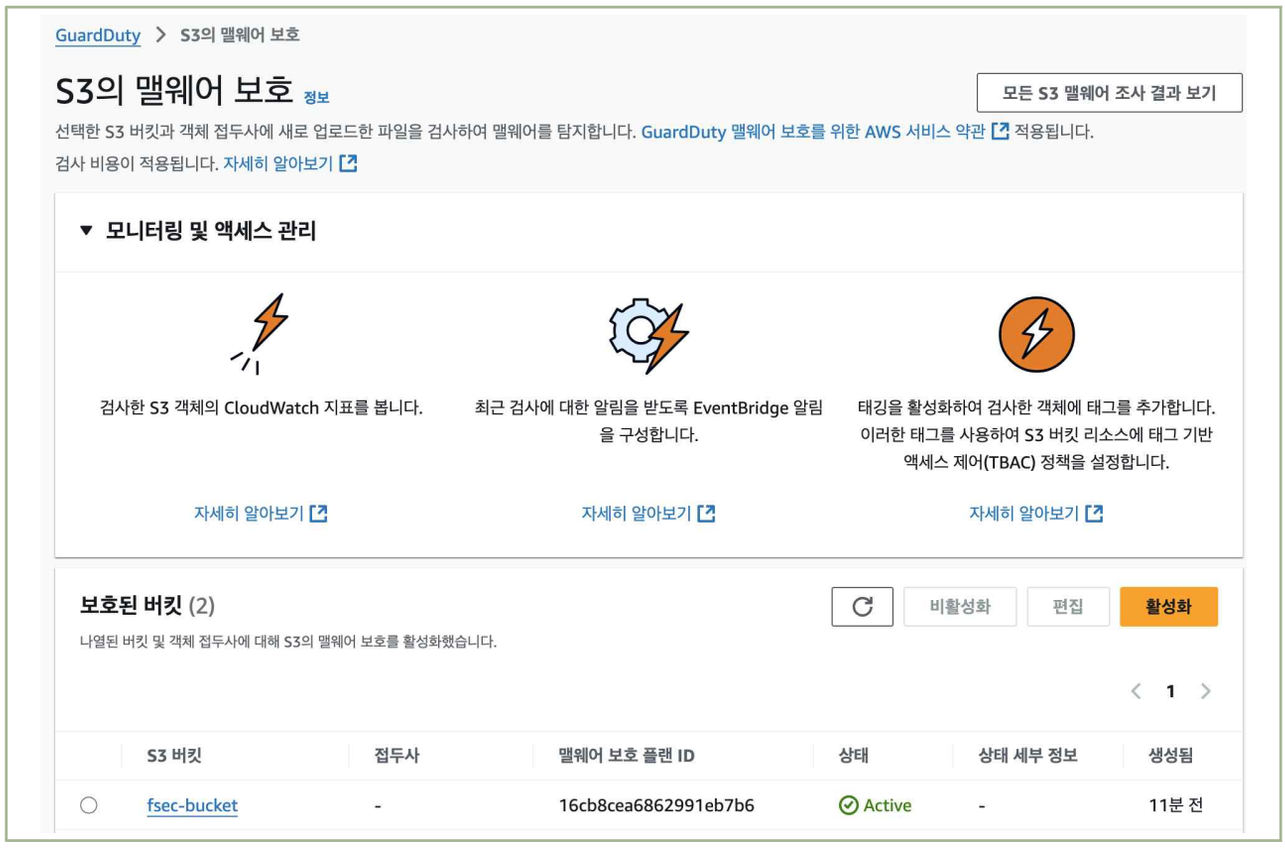


2) 금융회사에서 스토리지 내 파일 업로드 시 확장자 등을 검증할 수 있는 절차 마련

- **(파일 업로드 검증 절차)** 해킹, 악성코드 감염, 개인정보 유출 등 보안사고 예방을 위해 파일 업로드시 보안 절차를 마련하여 안정성을 확보할 수 있으며, 다음과 같은 것을 고려해 볼 수 있다.

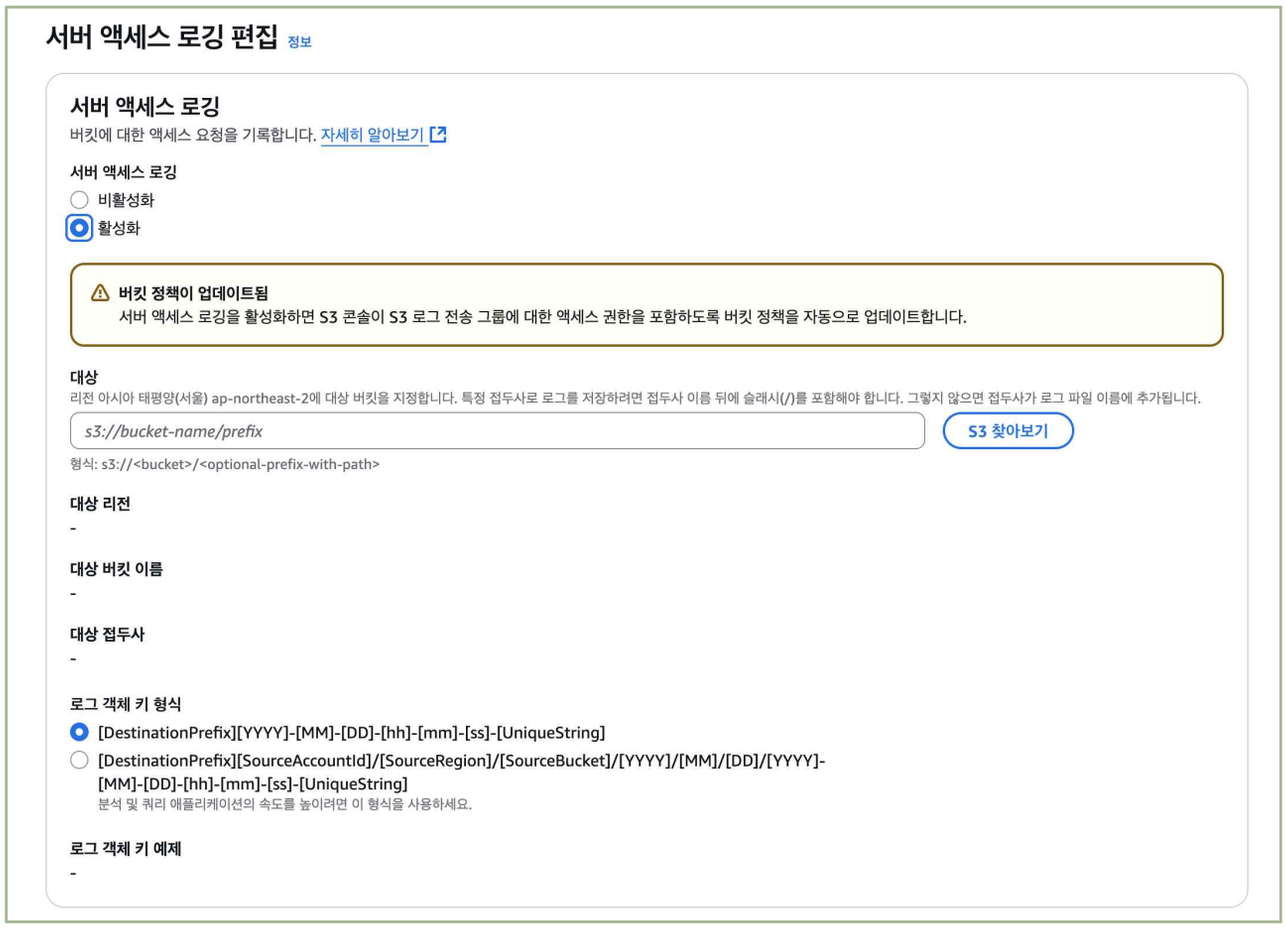
1. **(파일 확장자 및 MIME 타입 이중 검증)** 단순 확장자 필터링이 아닌, 파일의 실제 내용(Mime Type)과 확장자를 모두 확인하는 것이 필요하다. EXE 를 JPG 로 위장하는 것과 같은 확장자 우회 공격의 방지
2. **(악성코드 및 취약점 검사)** 파일 업로드시 악성코드 탐지 솔루션으로 파일 검사를 수행하며, S3 에서는 Amazon GuardDuty Malware Protection for S3 를 통해서 악성코드 검사 수행 가능

[Amazon GuardDuty 의 S3 악성코드 보호]



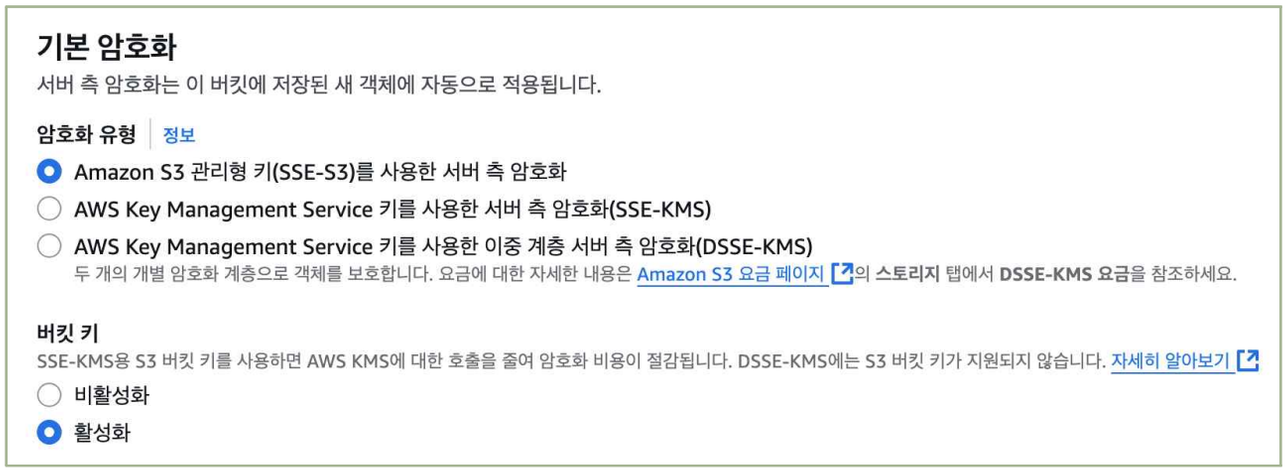
3. **(파일명 및 경로 검증)** 파일 명에 특수 문자, 경로 삽입등 위험 요소가 포함되지 않도록 필터링
4. **(업로드 로그 및 모니터링)** 업로드 이력, 차단 내역등을 기록 보관하며 이상 징후 발생시 관리자 알림 및 추가 조치가 필요하다. S3 의 경우는 서버 엑세스 요청을 기록하여 접근 이력을 확인해 볼 수 있다.

[버킷에 대한 엑세스 요청 기록]



5. (파일 보관 암호화) 파일 보관시 암호화 저장을 사용하며, SSE-S3, SSE-KMS 또는 더 안전하게 보관하기 위해 이중 계층 DSSE-KMS 를 선택할 수 있다.

[Amazon S3 기본 암호화]



4 참고 사항

- **(S3 악성코드 탐지)** Amazon GuardDuty Malware Protection for S3 를 활용하여 S3 에 업로드 되는 파일에 대해서 악성코드 검사가 가능하다. 또한 S3 뿐만 아니라 EC2 에 대해서도 사용이 가능하다.

8. 백업 및 이중화 관리

8.1. 클라우드 이용에 관한 행위추적성 증적(로그 등) 백업
8.2. 행위추적성 증적(로그 등) 백업 파일 무결성 검증
8.3. 금융회사 전산자료 백업
8.4. 금융회사 전산자료 백업 파일 무결성 검증
8.5. 행위추적성 증적, 전산자료 등 백업에 관한 기록 및 관리
8.6. 백업파일 원격 안전지역 보관
8.7. 주요 전산장비 이중화

8 백업 및 이중화 관리

1 기준

식별번호	기준	내용
8.1.	**클라우드 이용에 관한 행위추적성 증적(로그 등) 백업**	**금융회사가 클라우드 이용 시 발생하는 다양한 행위추적성 증적(가상자원, API, 네트워크서비스, 스토리지 관리, 계정 및 권한관리 등)의 보관기관 확보 등을 위해 백업을 수행(1년이상 보관)하여야 한다.**

2 설명

- 금융회사가 클라우드 이용 시 발생하는 다양한 행위추적성 증적에 대해 백업을 수행(1년이상 보관)하여야 한다.

  - 예시

    1) 스토리지서비스를 별도로 생성 및 연동하여 행위 감사로그 백업

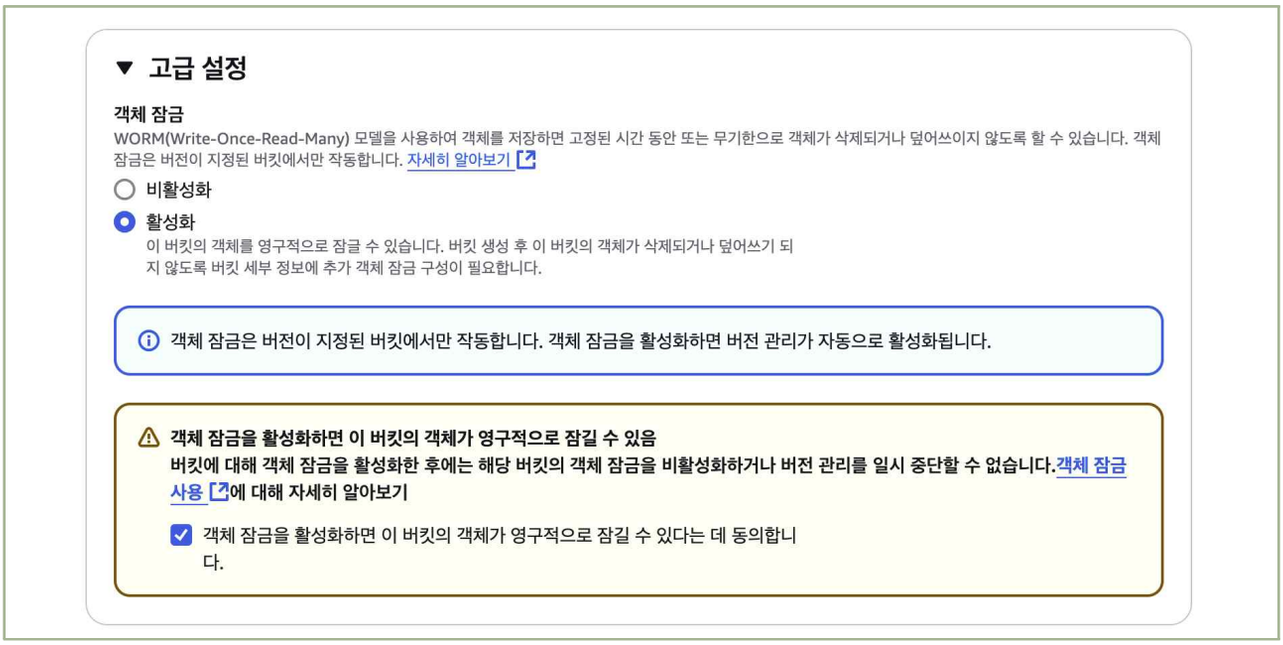
    2) 클라우드 웹 콘솔 내 행위 감사로그를 별도의 파일 형태로 다운로드 하여 별도로 보관 등

3 우수 사례

1) 행위 감사로그 저장소 생성 및 보존 기간 설정

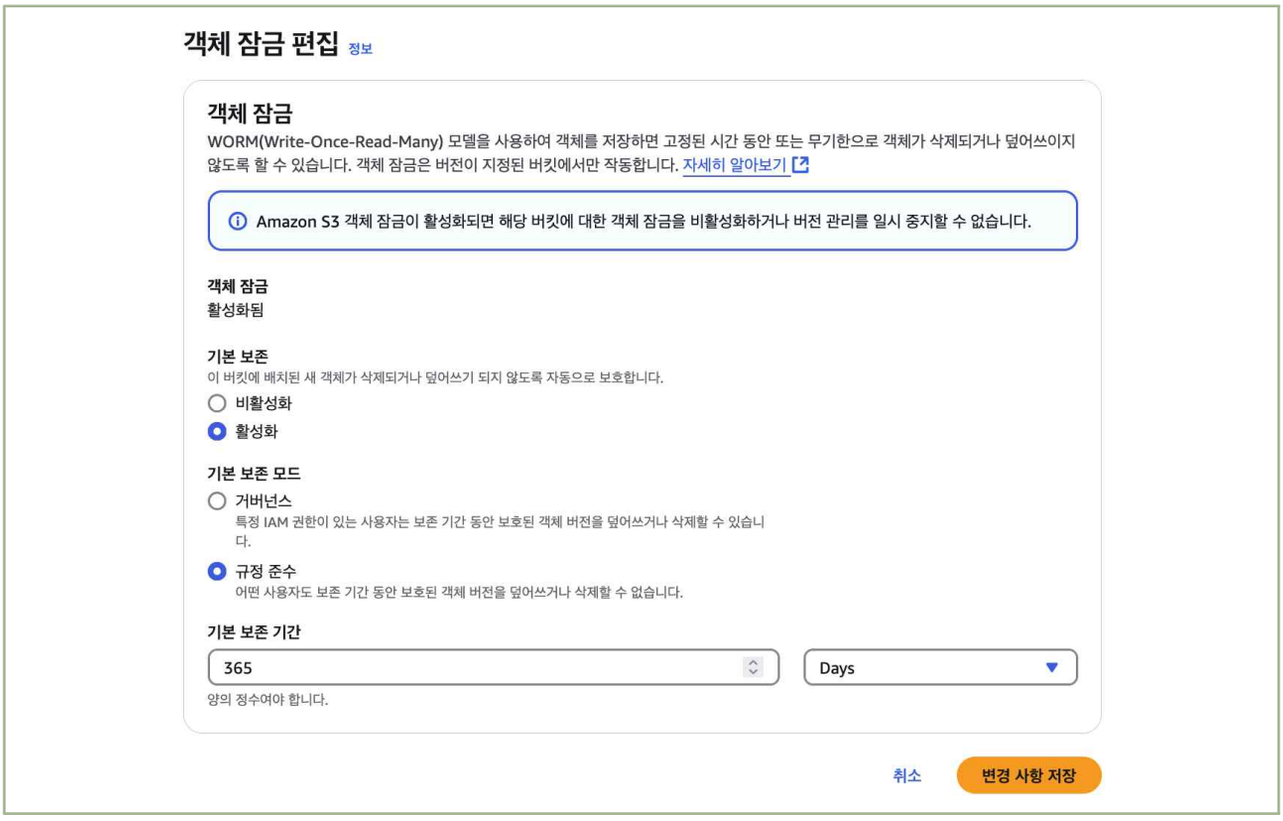
AWS에서 행위 감사로그는 CloudTrail을 통해서 관리합니다. 별도 추적 생성을 하지 않더라도 기본적으로 90일 동안 콘솔에서 볼 수 있지만, 90일 이상의 감사로그 저장을 위해서 별도의 추적을 생성하고 전용 S3 버킷을 만들어서 행위 감사로그를 보관하고, AWS Backup을 활용하여 백업하도록 합니다.

- **(AWS 관리콘솔)** ‘Amazon S3’ → ‘범용 버킷’ → ‘버킷 만들기’ → 버킷 이름 설정 → 기본설정인 ‘객체 소유권 : 버킷 소유자 적용’ 및 ‘모든 퍼블릭 액세스 차단’을 그대로 두고 → ‘버킷 버전 관리’ 와 고급 설정 아래에 있는 ‘객체 잠금’ 을 활성화 한 뒤 S3 버킷을 생성 합니다.



| 그림 8-1-1 | CloudTrail 로그 저장 S3 버킷의 객체 잠금 설정]

이후, 생성된 버킷의 '속성' 탭을 클릭한 뒤 '객체 잠금' 을 편집합니다. 이 때, '기본 보존'을 활성화하고, 기본 보존 모드를 '규정 준수' 모드로 선택 합니다. 이후 '기본 보존 기간'에 보존될 기간을 명시(예, 365일) 하고 '변경 사항 저장'을 클릭한 뒤, 확인 창이 뜨면 '확인'을 입력하여 설정을 완료합니다.



| 그림 8-1-2 | CloudTrail 로그 저장 S3 버킷의 보존 기간 설정

- **(AWS CLI)** 버킷 생성 (기본적으로 '객체 소유권 : 버킷 소유자 적용' 및 '모든 퍼블릭 액세스 차단'이 적용된 상태로 생성됩니다) 및 버킷 버전 관리 설정

```
$aws s3api create-bucket --bucket <사용할 버킷 이름> \
--region ap-northeast-2 \
--create-bucket-configuration LocationConstraint=ap-northeast-2 \
--object-lock-enabled-for-bucket
```

```
❯ aws s3api create-bucket --bucket fsi-reference-book2025 --region ap-northeast-
2 --create-bucket-configuration LocationConstraint=ap-northeast-2 --object-lock-
enabled-for-bucket
```

```
{
    "Location": "http://fsi-reference-book2025.s3.amazonaws.com/"
}
(END)
```

```
$aws s3api put-bucket-versioning --bucket <사용할 버킷 이름> \
--versioning-configuration Status=Enabled
```

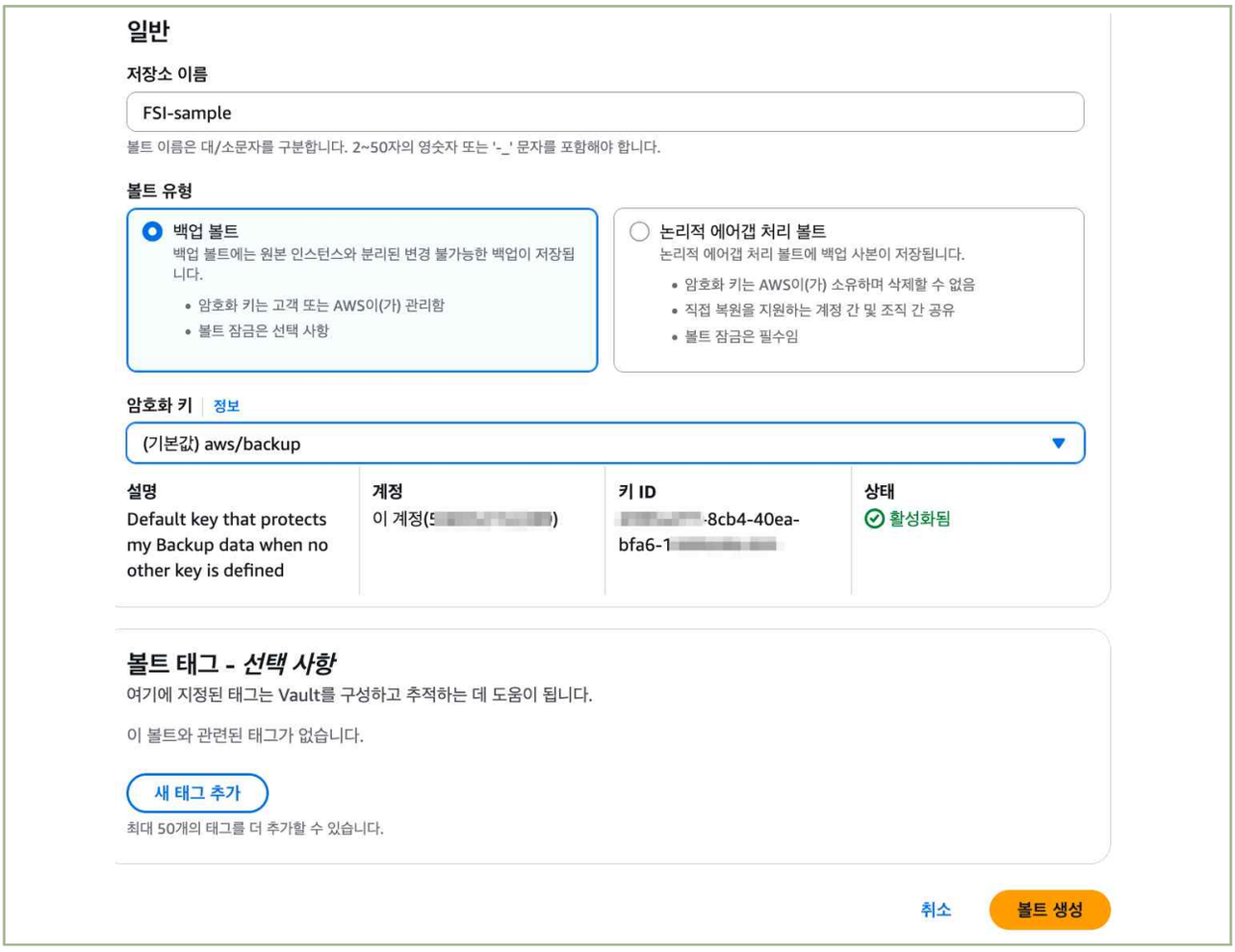
```
❯ aws s3api put-bucket-versioning --bucket fsi-reference-book2025 \
--versioning-configuration Status=Enabled
```

```
$aws s3api put-object-lock-configuration \
    --bucket <사용할 버킷 이름>  \
    --object-lock-configuration '{ "ObjectLockEnabled": "Enabled", "Rule": { "DefaultRetention":
{ "Mode": "GOVERNANCE", "Days": 365 }}}'
```

AWS Backup을 통해서 해당 버킷을 대상으로 하는 볼트를 생성합니다.

AWS Backup 볼트 생성

- (AWS 관리콘솔) 'AWS Backup' → '볼트' → '새 볼트 생성' → 저장소 이름 입력 → AWS KMS 암호화 키 선택(기본 키 또는 사용자 지정 키) → '볼트 생성' 버튼 클릭



| 그림 8-1-3 | 볼트 생성 화면

(AWS CLI)

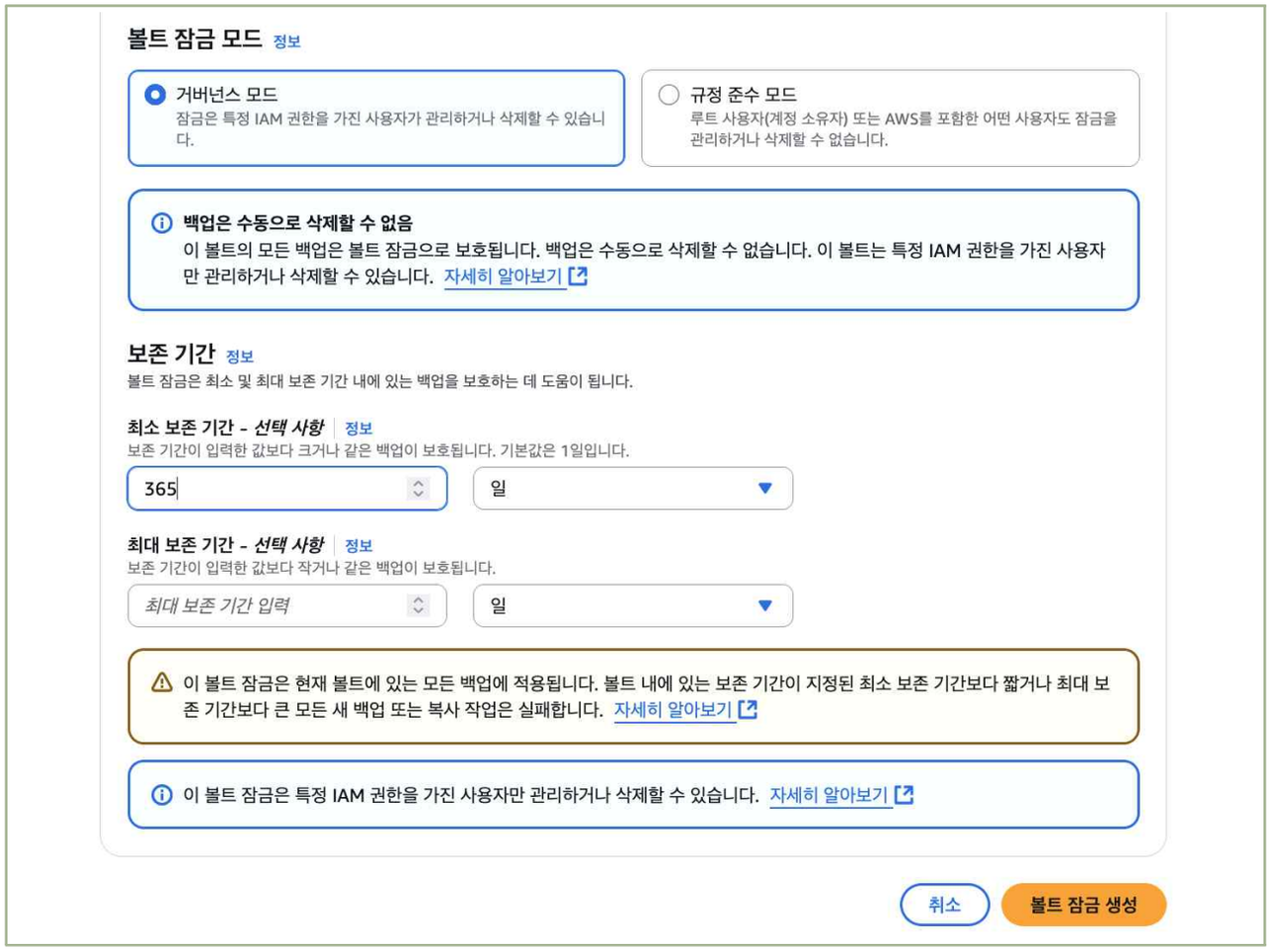
```
aws backup create-backup-vault \
  --backup-vault-name fsi-data-backup \
  --encryption-key-arn
arn:aws:kms:ap-northeast-2:123456789012:key/abcd1234-a123-456a-a12b-a123b4cd56ef
```

```
❯ aws backup create-backup-vault \
  --backup-vault-name fsi-data-backup \
  --encryption-key-arn arn:aws:kms:ap-northeast-2:123456789012:key/abcd1234-a123-456a-a12b-a123b4cd56ef
```

```
{
    "BackupVaultName": "fsi-data-backup",
    "BackupVaultArn": "arn:aws:backup:us-east-1:558352152289:backup-vault:fsi-data-backup",
    "CreationDate": "2025-05-17T11:05:10.686000-07:00"
}
(END)
```

볼트 잠금 생성

- (AWS 관리콘솔) 'AWS Backup' → '백업 볼트 잠금' → '볼트 잠금 생성' 클릭 → 생성한 백업 볼트 선택 → '볼트 잠금 모드' 거버넌스 모드 선태 → '최소 보존 기간'에 '365일' 입력 → '볼트 잠금 생성' 버튼 클릭 → 대화창에 '확인' 을 입력 → '볼트 잠금이 원하는 대로 구성되었으며 볼트가 잠긴 후에는 변경 불가능함을 인정합니다'를 선택 → '생성' 버튼을 눌러서 완료



| 그림 8-1-4 | 볼트 잠금 생성 화면

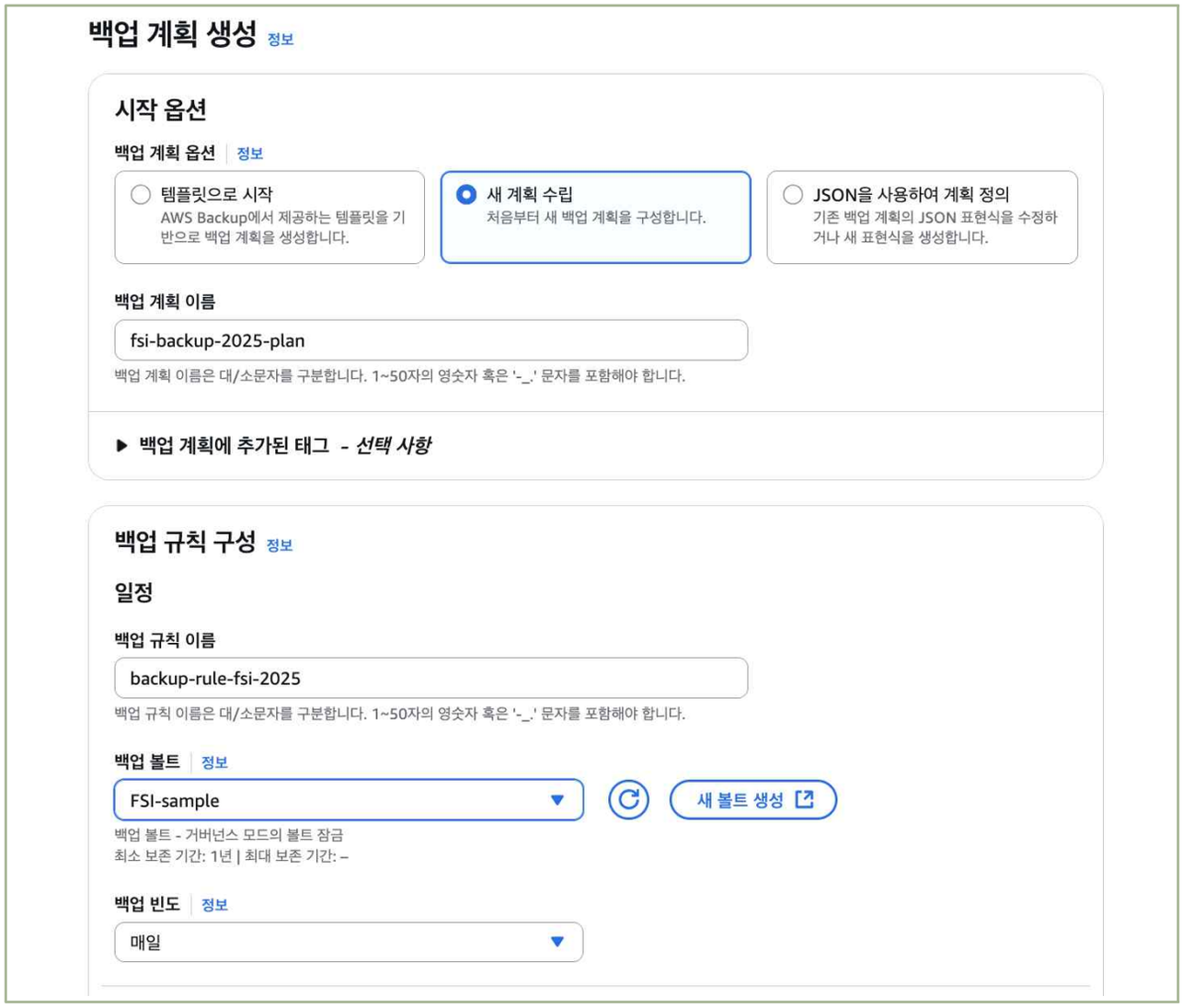
- **(AWS CLI)**

```
$aws backup put-backup-vault-lock-configuration \
  --backup-vault-name <대상 버킷 이름> \
--min-retention-days 365
```

```
aws backup put-backup-vault-lock-configuration \
  --backup-vault-name fsi-data-backup \
  --min-retention-days 365
```

백업 계획 생성 및 리소스 할당

- **(AWS 관리콘솔)** 'AWS Backup' → '백업 계획' → '백업 계획 생성' → '새 계획 수립' 선택 → 백업 계획 이름 입력 → '백업 규칙 구성' → 백업 규칙 이름 입력 → 백업 볼트 선택 → 백업 빈도 설정(일별, 주별, 월별 등) → '백업 기간' 및 '시작 시간' 설정 → '보존 기간'을 '365일' 이상 입력 → '계획 생성' 버튼 클릭 → '리소스 할당' 화면에서 리소스 할당 이름 입력 → '리소스 선택 정의'에서 '특정 리소스 유형 포함' 선택 → '리소스 유형'으로 'S3'를 리소스 유형으로 포함 하고, CloudTrail 로그 저장 S3 버킷을 선택 → '리소스 할당' 버튼 클릭하여 완료



| 그림 8-1-5 | 백업 계획 생성 화면

(AWS CLI)

```
$aws backup create-backup-plan \
  --backup-plan '{
    "BackupPlanName": "fsi-data-backup-plan",
    "Rules": [
      {
        "RuleName": "fsi-data-backup-rule",
        "TargetBackupVaultName": "fsi-data-backup",
        "ScheduleExpression": "cron(0 1 * * ? *)",
        "StartWindowMinutes": 60,
        "CompletionWindowMinutes": 180,
        "Lifecycle": {
          "DeleteAfterDays": 365
        }
      }
    ]
  }'
```

```
aws backup create-backup-plan \
  --backup-plan '{
    "BackupPlanName": "fsi-data-backup-plan",
    "Rules": [
      {
        "RuleName": "fsi-data-backup-rule",
        "TargetBackupVaultName": "fsi-data-backup",
        "ScheduleExpression": "cron(0 1 * * ? *)",
        "StartWindowMinutes": 60,
        "CompletionWindowMinutes": 180,
        "Lifecycle": {
          "DeleteAfterDays": 365
        }
      }
    ]
  }'
```

```
{
    "BackupPlanId": "de81edc0-e568-4ae3-8008-42525caddcaf",
    "BackupPlanArn": "arn:aws:backup:us-east-1:558352152289:backup-plan:de81edc0-e568-4ae3-8008-42
525caddcaf",
    "CreationDate": "2025-05-17T11:45:11.174000-07:00",
    "VersionId": "NTVjYmMzMWMtZTcwOC00MDQ5LTk0ZTgtMzM5ZjJjMWFiM2Nj"
}
(END)
```

```
$aws backup create-backup-selection \
  --backup-plan-id $(aws backup list-backup-plans --query
"BackupPlansList[?BackupPlanName=='fsi-data-backup-plan'].BackupPlanId" --output text) \
  --backup-selection '{
    "SelectionName": "fsi-resource-selection",
```

```
    "IamRoleArn":
"arn:aws:iam::123456789012:role/service-role/AWSBackupDefaultServiceRole",
    "Resources": [
"arn:aws:s3:::fsi-reference-book2025"
    ]
  }'
```

```
{
    "SelectionId": "6093a7bb-858d-4a59-a590-a47095365da6",
    "BackupPlanId": "de81edc0-e568-4ae3-8008-42525caddcaf",
    "CreationDate": "2025-05-17T13:04:26.043000-07:00"
}
(END)
```

2) 행위 감사로그(CloudTrail 로그)를 별도의 파일 형태로 다운로드 하여 보관

CloudTrail 콘솔에서는 최근 90일간의 이벤트만 조회 가능하며, S3에 저장된 로그파일은 콘솔에서 객체 단위로만 다운로드가 가능합니다. AWS CLI를 사용하면 S3 버킷에 저장된 CloudTrail 로그를 디렉토리 단위로 효율적으로 다운로드할 수 있습니다.

- **(AWS CLI)** S3 sync를 사용한 특정 경로의 CloudTrail 로그 다운로드

  aws s3 sync s3://[버킷이름]/AWSLogs/[계정ID]/CloudTrail/[리전]/[년도]/[월]/ ./cloudtrail-logs/[년도]/[월]/

  - 〈예시〉

    $aws s3 sync s3://fsi-reference-book2025/AWSLogs/123456789012/CloudTrail/ap-northeast-2/2025/ ./cloudtrail-logs/2025/

4 참고 사항

- CloudTrail 콘솔을 통해서도 직전 90일 동안의 기록을 '이벤트 기록'을 CSV 또는 JSON 형태로 다운로드 할 수 있습니다.
- 무료 3rd party 툴 (예, S3 Browser, Cyberduck 등)을 활용하면 Local로 좀더 간편하게 파일을 다운로드할 수 있습니다.

1 기준

식별번호	기준	내용
8.2.	**행위추적성 증적(로그 등) 백업 파일 무결성 검증**	**백업을 통해 보관되고 있는 행위추적성 파일에 대한 무결성이 보장되어야 한다.**

2 설명

- 이용자의 행위추적성 백업파일은 무결하게 보관하여야 한다.

  - 예시

    1) 클라우드 서비스 제공자가 제공하는 백업 저장소(스토리지 등)로 백업하는 경우 저장소 내 파일이 훼손될 시 알람을 받을 수 있도록 설정

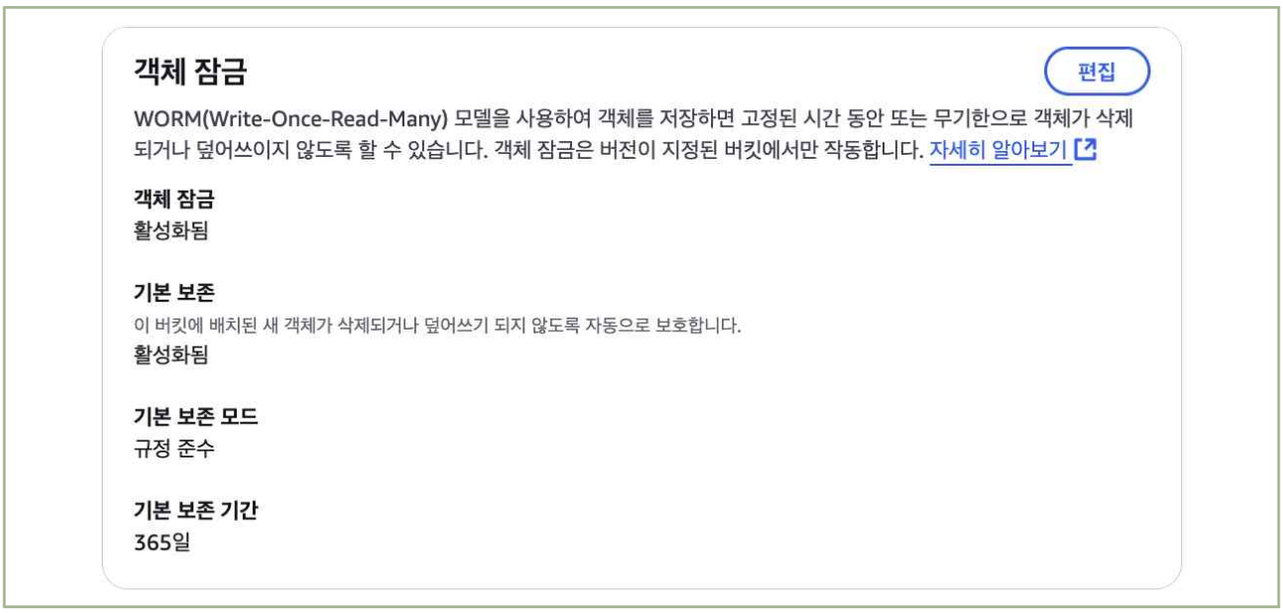
    2) 금융회사가 운영하는 백업 서버로 백업하는 경우 무결하게 보관

3 우수 사례

1) 행위 감사로그 저장소 보존 모드 확인

- 행위 감사로그를 임의로 삭제하거나 변경하지 못하도록 행위 감사로그가 저장되는 S3 버킷에 대한 객체 잠금이 활성화 되었는지 확인합니다.

- **(AWS 관리콘솔)** ‘Amazon S3’ → ‘범용 버킷’ → CloudTrail이 기록되는 버킷을 클릭하고 → ‘속성’ 탭에서 ‘객체 잠금’ 이 활성화 상태인지 확인 합니다. 만약 활성화 되어 있지 않다면, ‘8.1 클라우드 이용에 관한 행위추적성 증적(로그 등) 백업’ 항목을 참조하여 활성화 하고 기본 보존을 설정합니다.

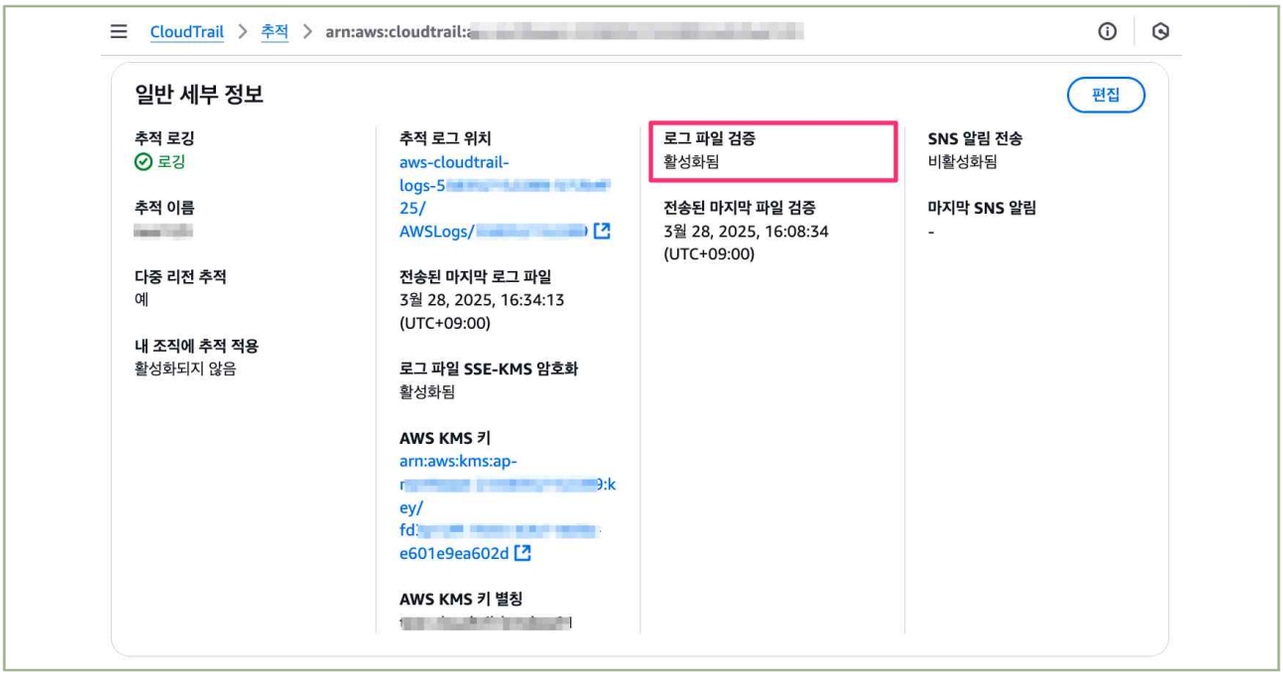


| 그림 8-2-1 | 행위 감사로그 저장 S3 버킷의 객체 잠금 설정 확인

2) 추적에서 로그파일 무결성 검증 사용

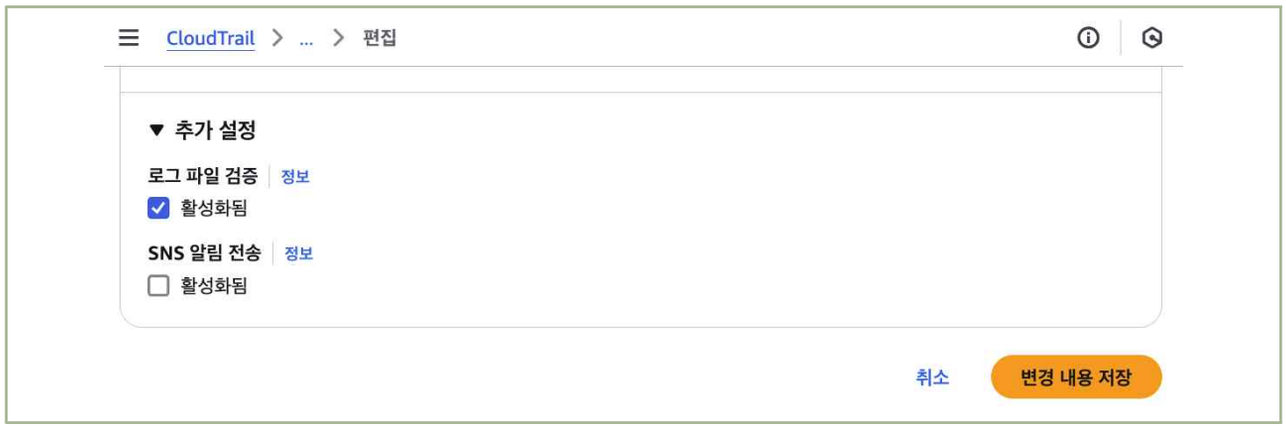
- AWS CloudTrail이 로그 파일을 전송한 후 해당 파일이 수정, 삭제 또는 변경되지 않았는지 확인하도록 CloudTrail 로그 파일 무결성 검증을 사용합니다.

- **(AWS 관리콘솔)** 'CloudTrail → '추적' → 원하는 이름을 클릭하고 → '일반 세부 정보' 란에서 '로그파일 검증' 이 '활성화됨' 상태임을 확인합니다.



| 그림 8-2-2 | CloudTrail 의 로그 파일 검증

만약 활성화 되어 있지 않다면, '일반 세부 정보' 란의 '편집'을 누른 뒤 나타나는 화면에서 아래쪽 '추가 설정'을 확장하고 '로그 파일 검증'의 체크박스를 눌러서 '활성화됨'이 선택되도록 합니다. 이후 '변경 내용 저장'을 눌러서 설정을 완료 합니다.



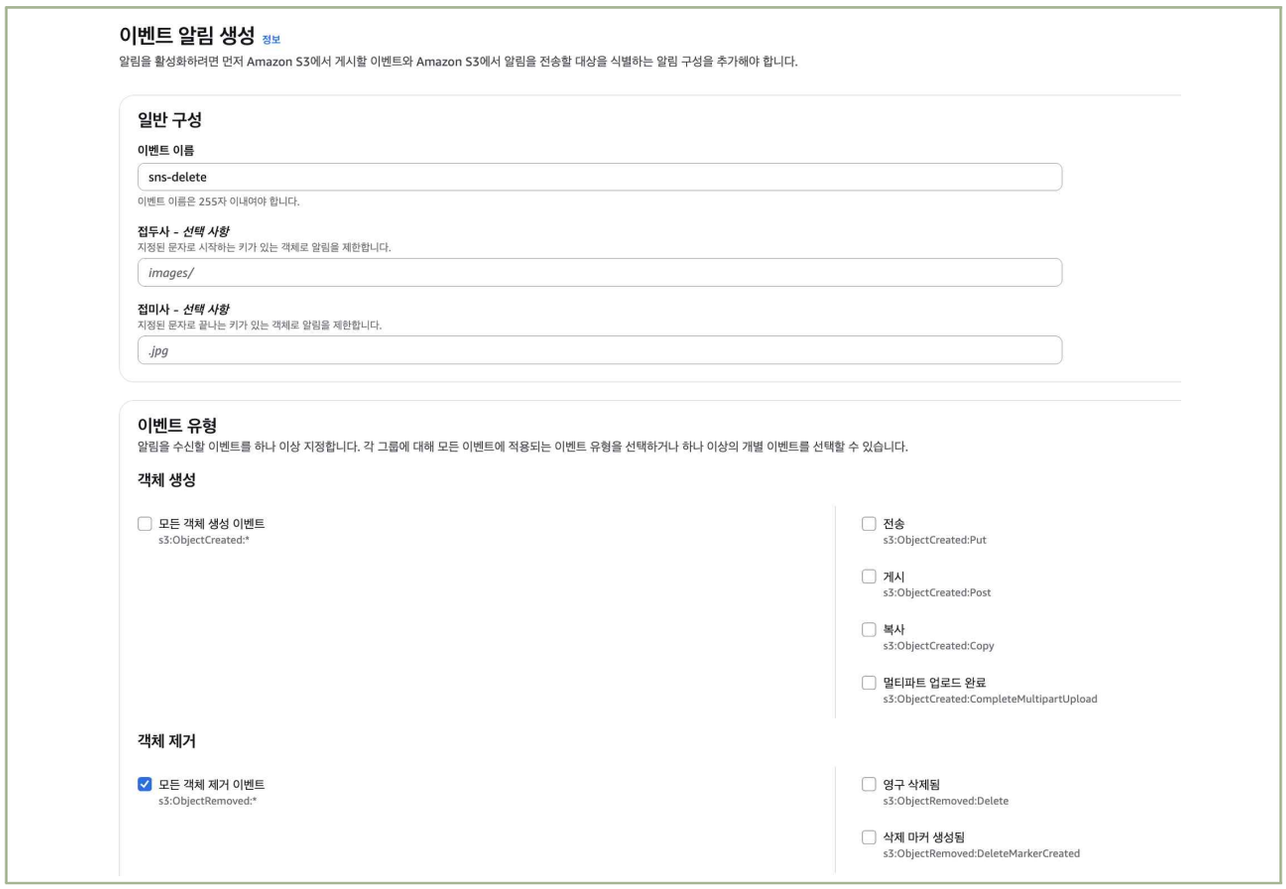
| 그림 8-2-3 | CloudTrail 의 로그 파일 검증 활성화

- **(AWS CLI)**

  $aws cloudtrail update-trail --name 〈대상 CloudTrail 이름〉 --enable-log-file-validation

3) 행위 감사로그 저장소 내 파일을 훼손하려는 시도에 대한 알람 설정

- **(AWS 관리콘솔)** CloudTrail이 기록되는 버킷을 클릭하고 → '속성' 탭으로 이동한 뒤 → 스크롤 다운해서 '이벤트 알림' 섹션의 '이벤트 알림 생성'을 클릭 → '이벤트 이름'을 입력하고 → '이벤트 유형' 에서 '객체 제거 - 모든 객체 제거 이벤트' 를 선택한다 → '대상'으로 'SNS 주제'를 선택하고 적절한 SNS 주제를 선정한 뒤 → '변경 사항 저장'을 눌러서 알람 설정을 완료한다.



| 그림 8-2-4 | CloudTrail 보관 버킷의 이벤트 알림 생성 화면

- **(AWS CLI) 별도 notification.json 형태로 이벤트 알람 규칙을 생성해 두어야 함.**

```
$aws s3api put-bucket-notification-configuration \
    --bucket 〈대상 버킷명〉 \
    --notification-configuration file://notification.json
```

4 참고 사항

- CloudTrail을 생성할 때, '새 S3 버킷 생성'을 선택한 경우에는 '버킷 버전 관리' 와 '객체 잠금'이 활성화 되지 않음. 이 때는 CloudTrail이 저장되는 버킷으로 이동하여 별도로 버전관리와 객체 잠금을 설정해 줄 수 있음.

1 기준

식별번호	기준	내용
8.3.	**금융회사 전산자료 백업**	**관련 법령(전자금융거래법, 전자금융감독규정 등)에 따라 백업이 필요한 금융회사 전산자료에 대해 백업을 수행하여야 한다.**

2 설명

- 금융회사 클라우드 이용 시 관련 법령(전자금융거래법, 전자금융감독규정 등)에 따라 백업이 필요한 전산자료에 대해서는 백업을 수행하여야 하며, 중요업무인 경우 클라우드서비스와 관련한 중요 설정파일 및 가상 시스템 이미지도 백업 대상에 포함하여야 한다.(중요도에 따라 1년이상 보관)

  - 예시

    1) 클라우드 서비스 제공자가 제공하는 백업 서비스 이용

    2) 전산자료를 별도로 다운받아 금융회사가 관리하는 백업 서버내 보관

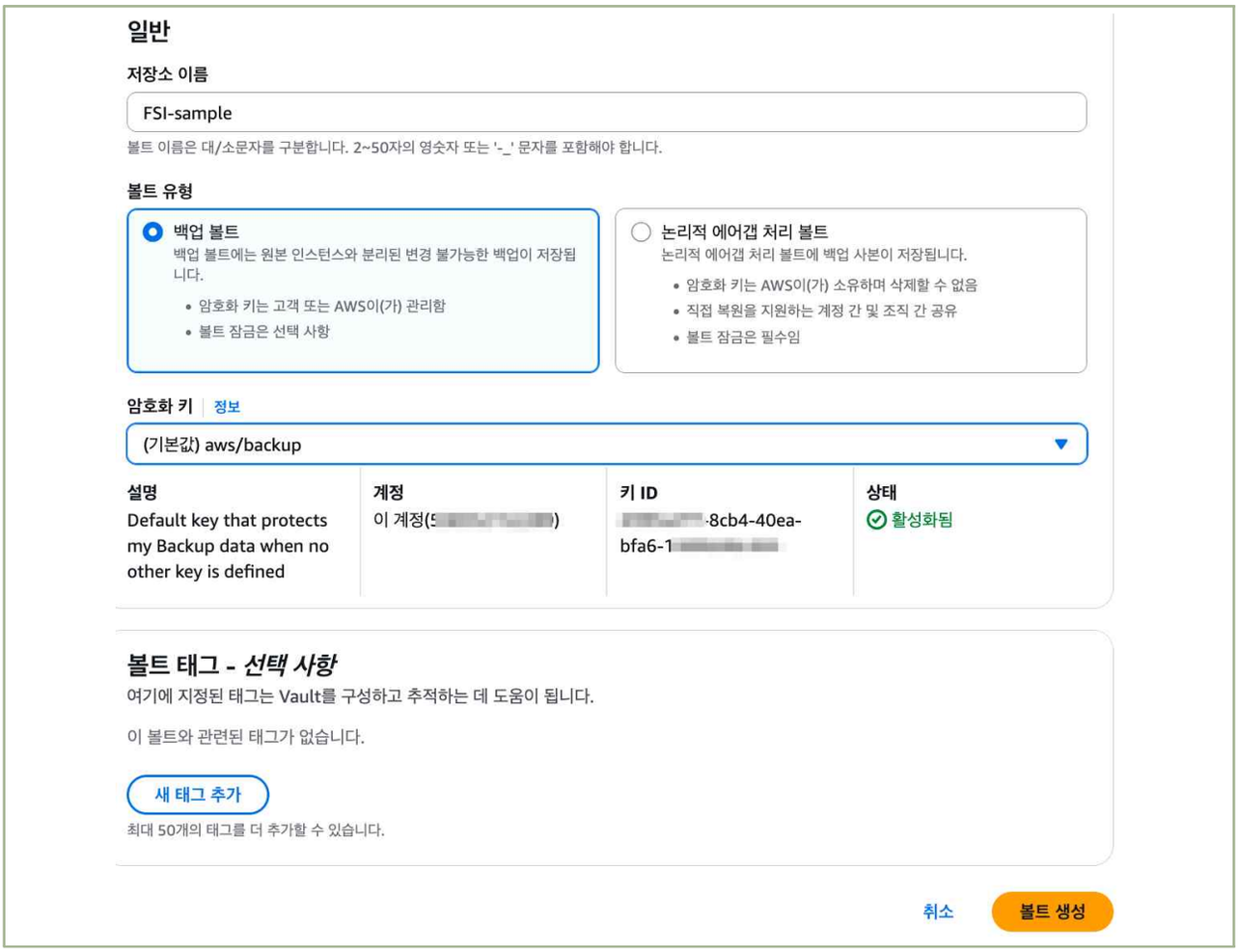
3 우수 사례

1) AWS Backup을 활용한 중앙 집중식 백업 관리

다양한 AWS 서비스(EC2, EBS, RDS, DynamoDB, EFS 등)의 데이터를 중앙에서 관리하고 자동화할 수 있는 완전 관리형 백업 서비스인 AWS Backup을 이용하여 1년 이상 보관이 필요한 전산자료의 백업을 구성할 수 있다.

AWS Backup 볼트 생성

- **(AWS 관리콘솔)** 'AWS Backup' → '볼트' → '새 볼트 생성' → 저장소 이름 입력 → AWS KMS 암호화 키 선택(기본 키 또는 사용자 지정 키) → '볼트 생성' 버튼 클릭



| 그림 8-3-1 | 볼트 생성 화면

(AWS CLI)

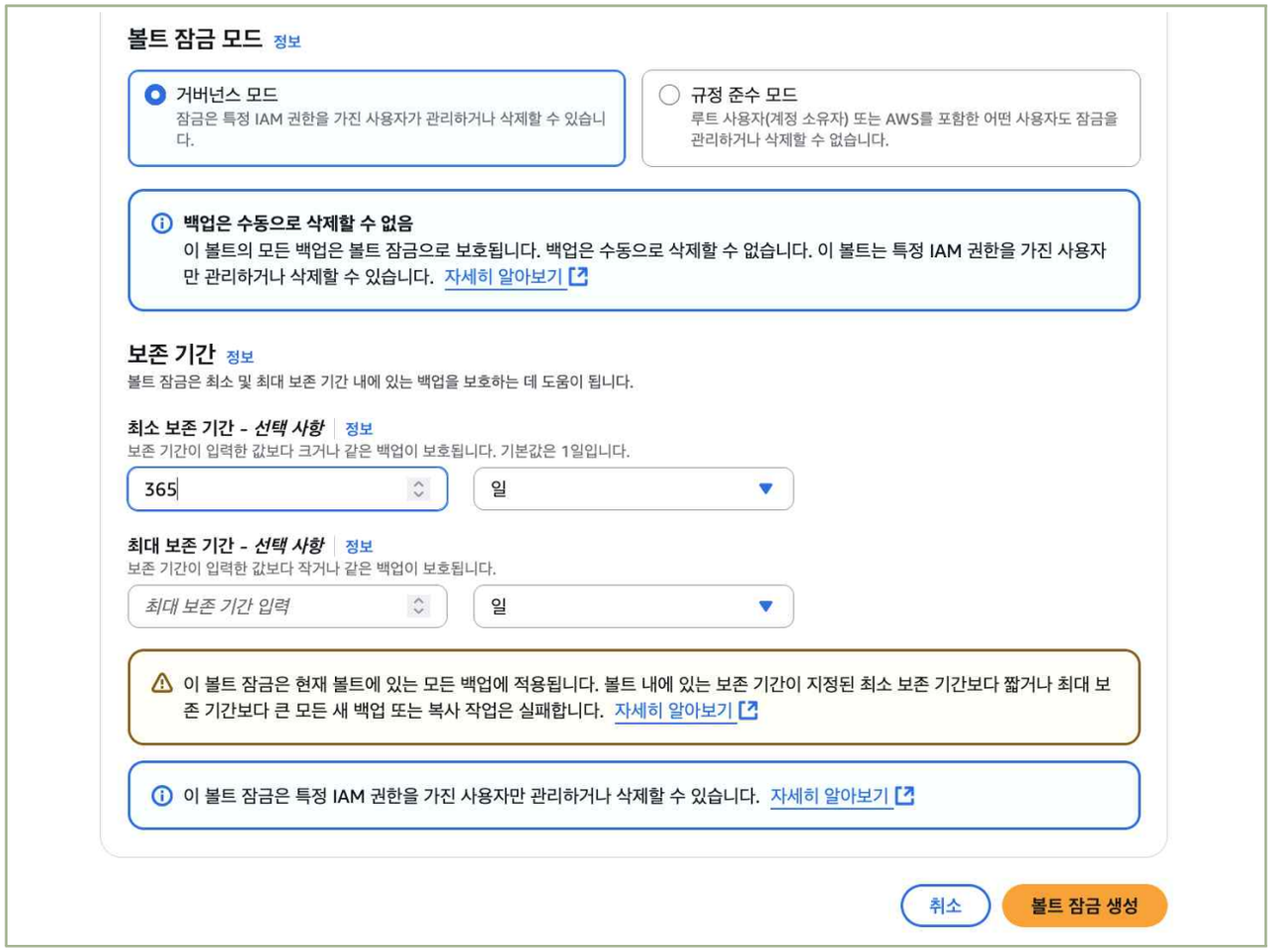
```
$aws backup create-backup-vault \
  --backup-vault-name fsi-data-backup \
  --encryption-key-arn
arn:aws:kms:ap-northeast-2:123456789012:key/abcd1234-a123-456a-a12b-a123b4cd56ef
```

```
❯ aws backup create-backup-vault \
  --backup-vault-name fsi-data-backup \
  --encryption-key-arn arn:aws:kms:ap-northeast-2:123456789012:key/abcd1234-a123-456a-a12b-a123b4cd56ef
```

```
{
    "BackupVaultName": "fsi-data-backup",
    "BackupVaultArn": "arn:aws:backup:us-east-1:558352152289:backup-vault:fsi-data-backup",
    "CreationDate": "2025-05-17T11:05:10.686000-07:00"
}
(END)
```

볼트 잠금 생성

- **(AWS 관리콘솔)** 'AWS Backup' → '백업 볼트 잠금' → '볼트 잠금 생성' 클릭 → 생성한 백업 볼트 선택 → '볼트 잠금 모드' 거버넌스 모드 선택 → '최소 보존 기간'에 '365일' 입력 → '볼트 잠금 생성' 버튼 클릭 → 대화창에 '확인' 을 입력 → '볼트 잠금이 원하는 대로 구성되었으며 볼트가 잠긴 후에는 변경 불가능함을 인정합니다'를 선택 → '생성' 버튼을 눌러서 완료



| 그림 8-3-2 | 볼트 잠금 모드 화면

- **(AWS CLI)**

```
aws backup put-backup-vault-lock-configuration \
  --backup-vault-name fsi-data-backup \
--min-retention-days 365
```

```
aws backup put-backup-vault-lock-configuration \
  --backup-vault-name fsi-data-backup \
  --min-retention-days 365
```

백업 계획 생성 및 리소스 할당

- (AWS 관리콘솔) 'AWS Backup' → '백업 계획' → '백업 계획 생성' → '새 계획 수립' 선택 → 백업 계획 이름 입력 → '백업 규칙 구성' → 백업 규칙 이름 입력 → 백업 볼트 선택 → 백업 빈도 설정(일별, 주별, 월별 등) → '백업 기간' 및 '시작 시간' 설정 → '보존 기간'을 '365일' 이상 입력 → '계획 생성' 버튼 클릭 → ‘리소스 할당’ 화면에서 리소스 할당 이름 입력 → ‘리소스 선택 정의’에서 원하는 리소스 유형 포함 → ‘리소스 할당’ 버튼 클릭하여 완료

```
$aws backup create-backup-plan \
  --backup-plan '{
    "BackupPlanName": "fsi-data-backup-plan",
    "Rules": [
      {
        "RuleName": "fsi-data-backup-rule",
        "TargetBackupVaultName": "fsi-data-backup",
        "ScheduleExpression": "cron(0 1 * * ? *)",
        "StartWindowMinutes": 60,
        "CompletionWindowMinutes": 180,
        "Lifecycle": {
          "DeleteAfterDays": 365
        }
      }
    ]
  }'
```

```
aws backup create-backup-plan \
  --backup-plan '{
    "BackupPlanName": "fsi-data-backup-plan",
    "Rules": [
      {
        "RuleName": "fsi-data-backup-rule",
        "TargetBackupVaultName": "fsi-data-backup",
        "ScheduleExpression": "cron(0 1 * * ? *)",
        "StartWindowMinutes": 60,
        "CompletionWindowMinutes": 180,
        "Lifecycle": {
          "DeleteAfterDays": 365
        }
      }
    ]
  }'
```

```
{
    "BackupPlanId": "de81edc0-e568-4ae3-8008-42525caddcaf",
    "BackupPlanArn": "arn:aws:backup:us-east-1:558352152289:backup-plan:de81edc0-e568-4ae3-8008-42
525caddcaf",
    "CreationDate": "2025-05-17T11:45:11.174000-07:00",
    "VersionId": "NTVjYmMzMWMtZTcwOC00MDQ5LTk0ZTgtMzM5ZjJjMWFiM2Nj"
}
(END)
```

```
$aws backup create-backup-selection \
  --backup-plan-id $(aws backup list-backup-plans --query
"BackupPlansList[?BackupPlanName=='fsi-data-backup-plan'].BackupPlanId" --output text) \
  --backup-selection '{
    "SelectionName": "fsi-resource-selection",
    "IamRoleArn":
"arn:aws:iam::123456789012:role/service-role/AWSBackupDefaultServiceRole",
    "Resources": [
      "arn:aws:ec2:*:*:instance/*",
      "arn:aws:s3:::*"
    ]
  }'
```

```
❯ aws backup create-backup-selection \
  --backup-plan-id $(aws backup list-backup-plans --query "BackupPlansList[?BackupPlanName=='fsi-
ata-backup-plan'].BackupPlanId" --output text) \
  --backup-selection '{
    "SelectionName": "fsi-resource-selection",
    "IamRoleArn": "arn:aws:iam::558352152289:role/service-role/AWSBackupDefaultServiceRole",
    "Resources": [
      "arn:aws:ec2:*:*:instance/*",
      "arn:aws:s3:::*"
    ]
  }'
```

```
{
    "SelectionId": "6093a7bb-858d-4a59-a590-a47095365da6",
    "BackupPlanId": "de81edc0-e568-4ae3-8008-42525caddcaf",
    "CreationDate": "2025-05-17T13:04:26.043000-07:00"
}
(END)
```

4 참고 사항

- AWS 백업에 대한 자세한 내용은 AWS 개발자 안내서를 참조하실 수 있습니다.

  https://docs.aws.amazon.com/ko_kr/aws-backup/latest/devguide/whatisbackup.html

1 기준

식별번호	기준	내용
8.4.	**금융회사 전산자료 백업 파일 무결성 검증**	**백업을 통해 보관되고 있는 전산자료에 대한 무결성이 보장되어야 한다.**

2 설명

- 금융회사의 전산자료 백업파일은 무결하게 보관하여야 한다.

  - 예시

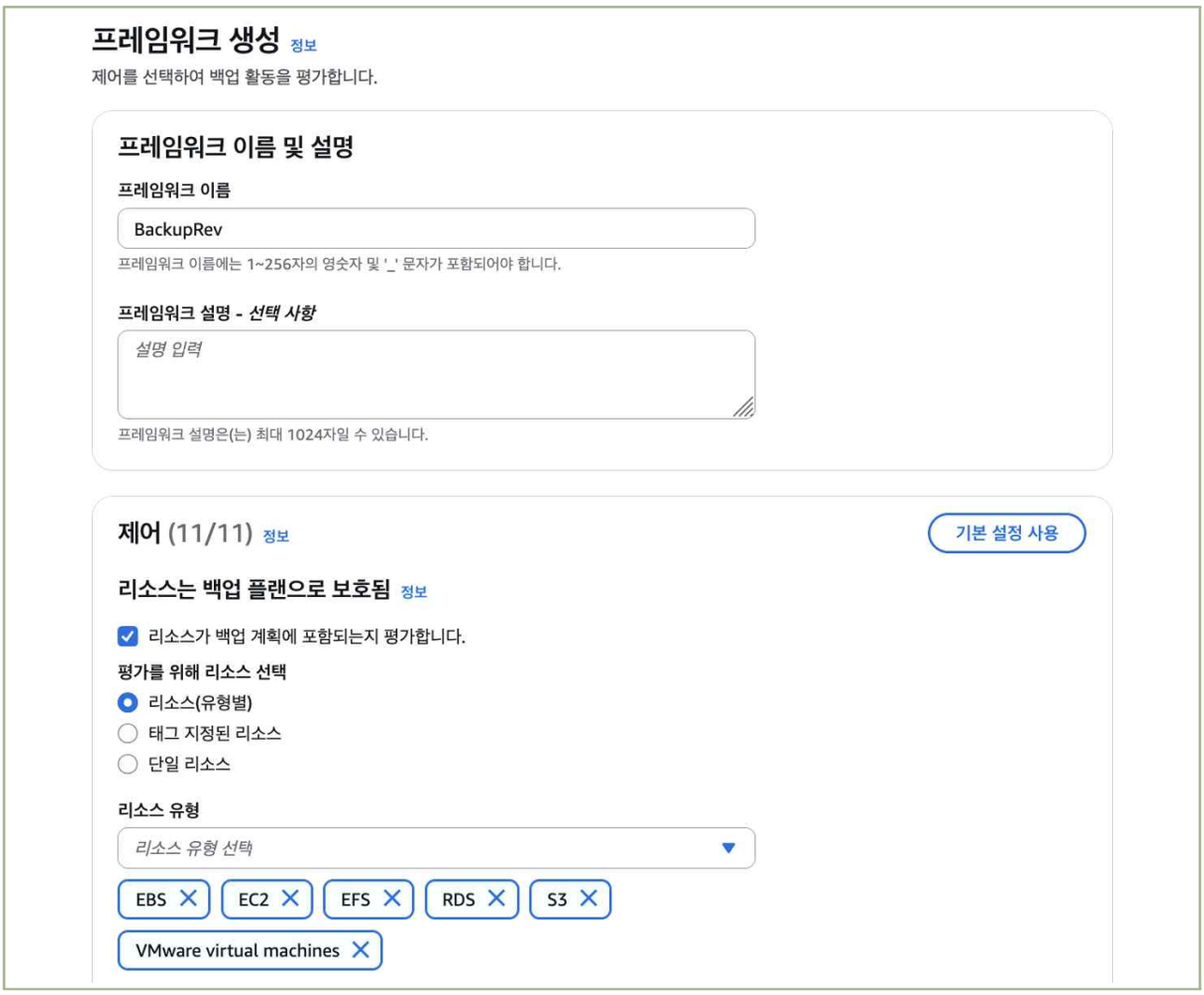
    1) 클라우드 서비스 제공자가 제공하는 백업 저장소(스토리지 등)로 백업하는 경우 저장소 내 파일이 훼손될 시 알람을 받을 수 있도록 설정

    2) 금융회사가 운영하는 백업 서버로 백업하는 경우 무결하게 보관

3 우수 사례

AWS 백업 / 프레임 워크 생성

- **(AWS 관리콘솔)** 'AWS Backup' → 'Backup Audit Manger' 메뉴 아래에 있는 '프레임워크' 선택 → '프레임워크 생성' 클릭 → 프레임워크 이름을 입력한 뒤 → '제어' 항목에 있는 내용에서 '리소스는 백업 플랜으로 보호됨'과 'AWS Backup 저장소 잠금으로 보호되는 백업'을 구성 → '평가를 위한 리소스 선태' 에서 백업 대상 리소스를 선택 → '프레임워크 생성' 버튼을 눌러서 완료



| 그림 8-4-1 | 프레임워크 생성 화면

(AWS CLI)

```
$aws backup create-framework \
  --framework-name <프레임워크 이름> \
  --framework-description "금융 전산자료 무결성 관리 프레임워크" \
  --framework-controls '[
    {
      "ControlName": "BACKUP_RESOURCES_PROTECTED",
      "ControlInputParameters": [
        {
          "ParameterName": "resourceType",
          "ParameterValue": "RDS,EFS,DynamoDB,EC2,S3"
        }
      ]
    },
    {
```

```
        "ControlName": "BACKUP_RECOVERY_POINT_MANUAL_DELETION_DISABLED",
        "ControlInputParameters": [
          {
            "ParameterName": "resourceType",
            "ParameterValue": "RDS,EFS,DynamoDB,EC2,S3"
          }
        ]
      }
]'
```

4 참고 사항

1 기준

식별번호	기준	내용
8.5.	**행위추적성 증적, 전산자료 등 백업에 관한 기록 및 관리**	**백업 자료의 생성, 변경, 삭제 등 관련 내역을 기록하고 관리하여야 한다.**

2 설명

- 백업 자료의 생성, 변경, 삭제 등 관련 내역을 기록하고 관리하여야 한다.

  - 예시

    1) 백업 파일 훼손에 대한 탐지 및 알람 설정

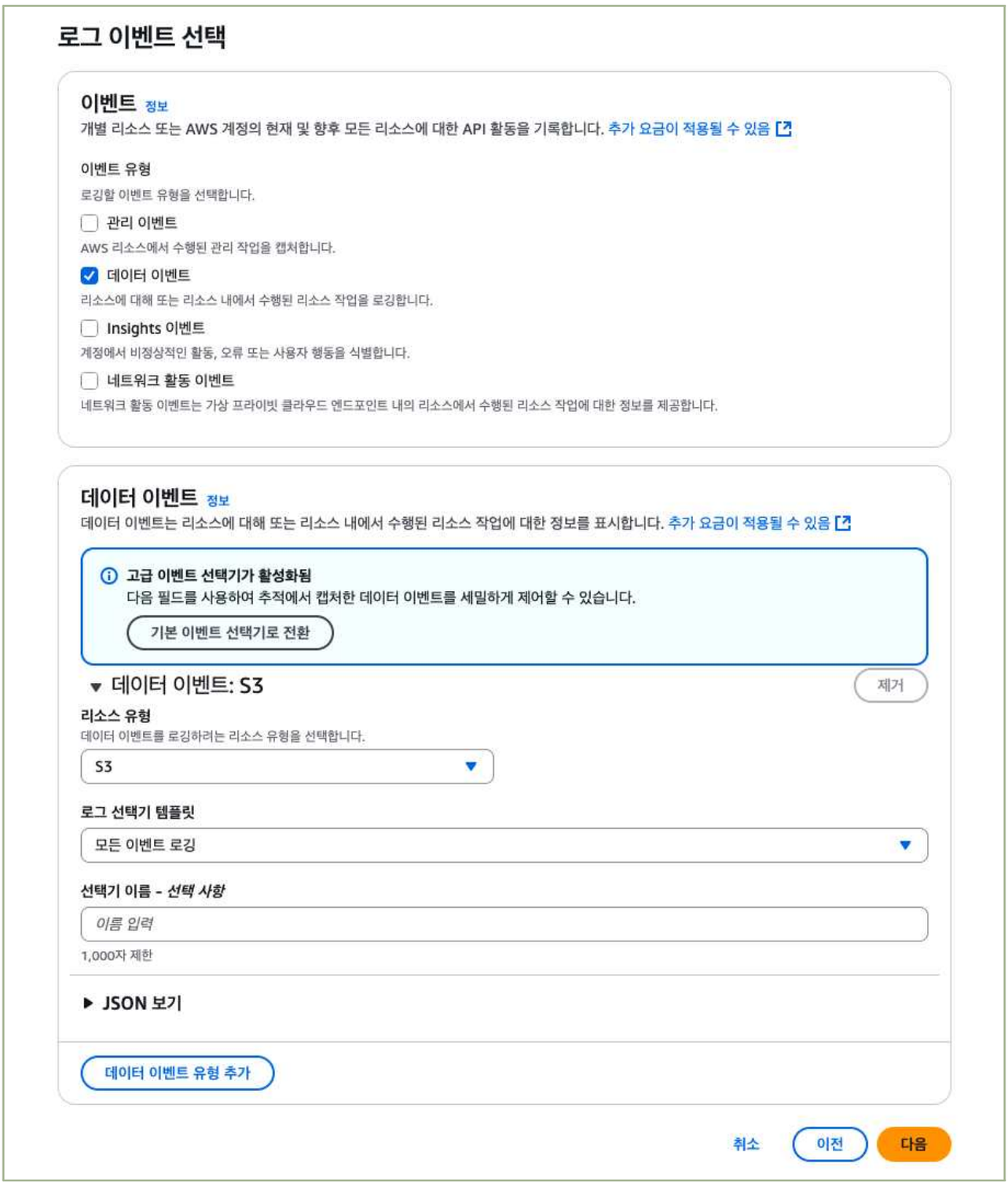
    2) 백업 기능(객체 잠금 등)을 통해 무결성 보장 등

3 우수 사례

1) 백업 파일 훼손에 대한 탐지 및 알람 설정

- AWS에서 백업 데이터 저장은 주로 Amazon S3를 사용합니다. S3를 백업 데이터 저장소로 사용하는 경우 백업 자료의 생성, 변경, 삭제 등에 대한 기록은 기본적으로 AWS CloudTrail에 저장됩니다. CloudTrail는 개별 리소스 또는 AWS 계정의 현재 및 향후 모든 리소스에 대한 API 활동을 기록하는데 이러한 이벤트 유형은 관리, 데이터, Insights, 네트워크 활동 이벤트가 있습니다. 백업 자료는 S3 객체 수준에서 이벤트 로그가 기록되어야 하므로 데이터 이벤트에 대한 추적을 생성하여야 백업 데이터에 대해 기록 및 관리가 가능합니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → 'AWS CloudTrail→ '추적' → '추적 생성' → '단계 2 로그 이벤트 선택'에서 이벤트 유형 중 '데이터 이벤트' 선택하고 데이터 이벤트의 리소스 유형은 'S3', 로그 선택기 템플릿은 '모든 이벤트 로깅'을 선택하여 추적을 생성



| 그림 8-5-1 | CloudTrail 추적 생성 화면

- **(AWS CLI)** AWS CLI를 사용하는 경우에는 create-trail로 추적을 생성하고 put-event-selectors를 사용하여 Trail 이벤트가 이벤트 선택기와 일치하는 경우 기록할 수 있도록 구성할 수 있습니다. Trail에 대해 최대 5개의 이벤트 선택기와 최대 250개의 데이터 리소스를 구성할 수 있습니다. 아래 예제는 먼저 Trail1 이라는 추적을 생성하고 읽기 전용 및 쓰기 전용 관리 이벤트를 포함한 모든 이벤트와 AWS 계정의 모든 Amazon S3 버킷 및 AWS Lambda 함수에 대한 모든 데이터 이벤트를 포함하도록 이벤트 선택기를 생성하는 예제입니다.

```
aws cloudtrail create-trail \
      --name Trail1 \
      --s3-bucket-name amzn-s3-demo-bucket \
      --is-multi-region-trail
```

Output:

```
{
   "IncludeGlobalServiceEvents": true,
   "Name": "Trail1",
   "TrailARN": "arn:aws:cloudtrail:ap-northeast-2:123456789012:trail/Trail1",
   "LogFileValidationEnabled": false,
   "IsMultiRegionTrail": true,
   "S3BucketName": "amzn-s3-demo-bucket"
}
```

| 그림 8-5-2 | CLI 명령 – CloudTrail 추적 생성

```
aws cloudtrail put-event-selectors \
      --trail-name Trail1 \
      --event-selectors '[{"ReadWriteType": "All","IncludeManagementEvents": true,"DataResources":
[{"Type":"AWS::S3::Object", "Values": ["arn:aws:s3:::"]},{"Type": "AWS::Lambda::Function","Values":
["arn:aws:lambda"]}]}]'
```

Output:

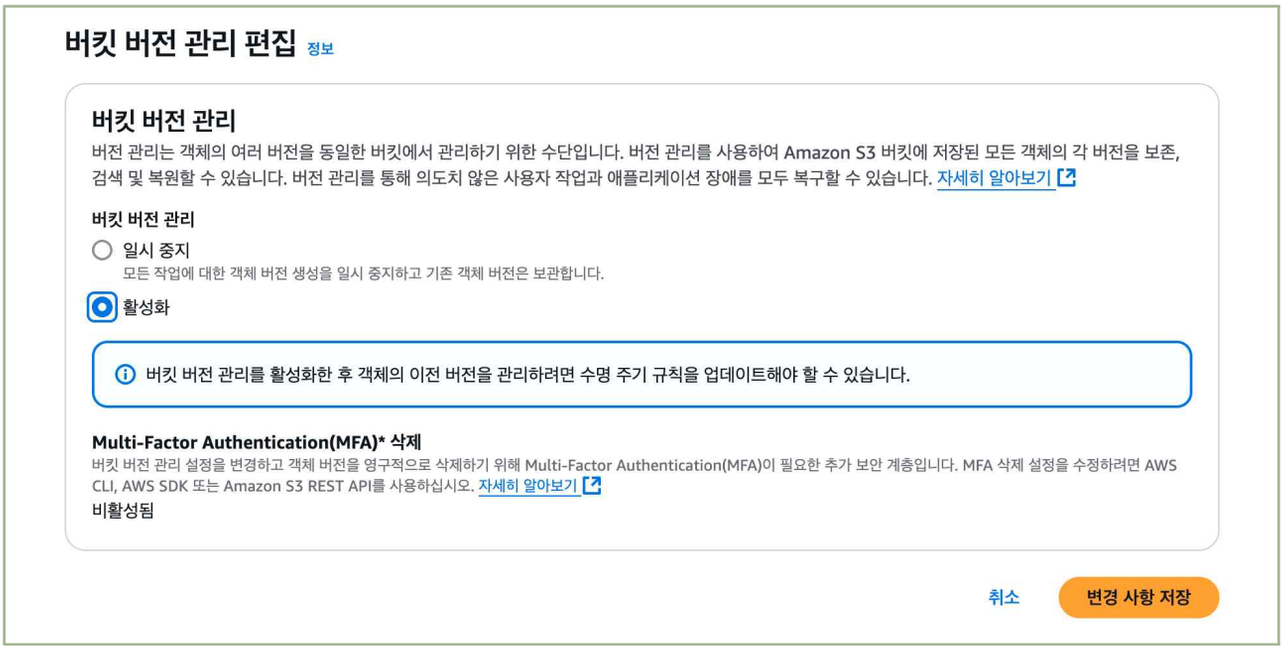
```
{
  "EventSelectors": [
    {
      "IncludeManagementEvents": true,
      "DataResources": [
        {
          "Values": [
            "arn:aws:s3:::"
          ],
          "Type": "AWS::S3::Object"
        },
        {
          "Values": [
            "arn:aws:lambda"
          ],
          "Type": "AWS::Lambda::Function"
        },
      ],
      "ReadWriteType": "All"
    }
  ],
  "TrailARN": "arn:aws:cloudtrail:ap-northeast-2:123456789012:trail/Trail1"
}
```

| 그림 8-5-3 | CLI 명령 – CloudTrail 추적 이벤트 선택기 생성

- 백업 파일 훼손을 방지하기 위해서 버킷 버전 관리 활성화를 권장합니다. 버전 관리를 사용하면 Amazon S3 버킷에 저장된 모든 객체의 각 버전을 보존, 검색 및 복원할 수 있습니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → 'Amazon S3 → 'General purpose buckets' → '해당 버킷' → '속성'에서 버킷 버전 관리 활성화

| 그림 8-5-4 | S3 버킷 버전 관리 화면



| 그림 8-5-5 | S3 버킷 버전 관리 편집 화면

- **(AWS CLI)** put-bucket-versioning을 사용하여 S3 버킷에 버전 관리를 활성화하여 여러 개의 객체 버전을 유지할 수 있습니다.

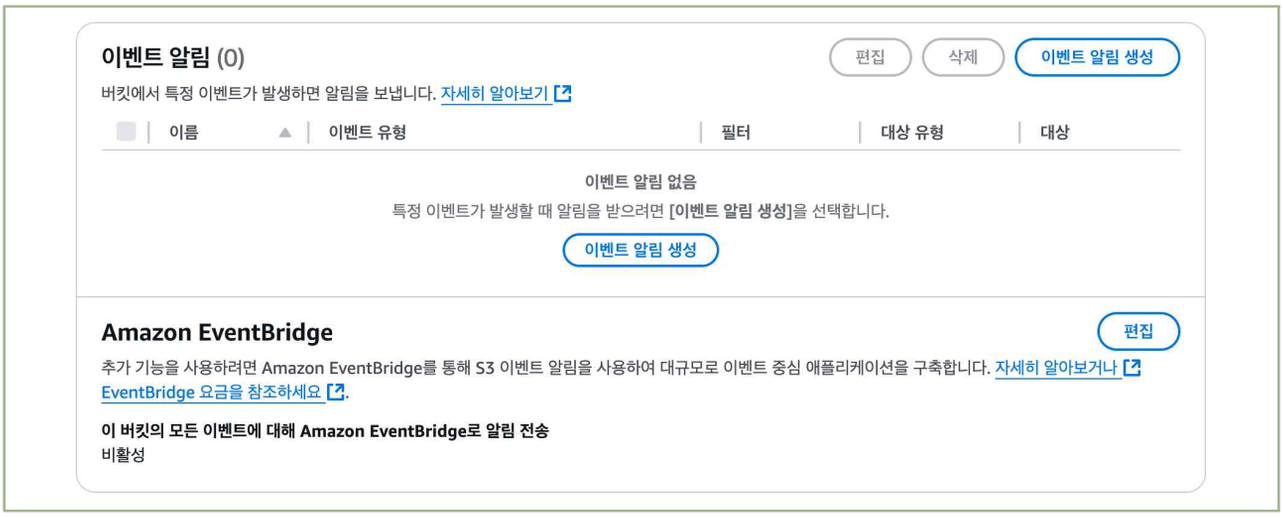
```
aws s3api put-bucket-versioning \
    --bucket amzn-s3-demo-bucket
    --versioning-configuration Status=Enabled
```

Output: 없음

| 그림 8-5-6 | CLI 명령 – S3 버킷 버전 관리 설정

- Amazon S3에 저장된 데이터가 덮어쓰기 및 삭제되어 훼손되는 경우에는 이벤트 알림을 생성하여 탐지할 수 있습니다. 이벤트 유형은 '모든 객체 생성 이벤트', '모든 객체 제거 이벤트'를 선택하고, 대상은 'Lambda 함수' 또는 'SNS 주제'로 지정하여 저장합니다. 이 때 알람 설정을 위해Lambda 함수 및 SNS 주제는 미리 생성되어 있어야 하며 운영자 혹은 관리자에게 '메일', '메시지'등으로 알람이 전송 되도록 코드 구현 및 설정이 필요합니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → 'Amazon S3 → 'General purpose buckets' → '해당 버킷' → '속성' 에서 이벤트 알림 생성



| 그림 8-5-7 | S3 버킷 이벤트 알림 화면

| 그림 8-5-8 | S3 버킷 이벤트 알림 생성 화면

- **(AWS CLI)** put-bucket-notification-configuration을 사용하여 S3 버킷 이벤트에 알림을 생성할 수 있습니다. Notification.json 파일은 현재 폴더에 있는 JSON 문서로 모니터링할 SNS 토픽과 이벤트 유형을 지정해야 합니다.

```
aws s3api put-bucket-notification-configuration \
      --bucket amzn-s3-demo-bucket \
      --notification-configuration file://notification.json
```

Contents of notification.json:

```
{
   "TopicConfigurations": [
      {
         "TopicArn": "arn:aws:sns:ap-northeast-2:123456789012:s3-notification-topic",
         "Events": [
            "s3:ObjectCreated:*"
         ]
      }
   ]
}
```

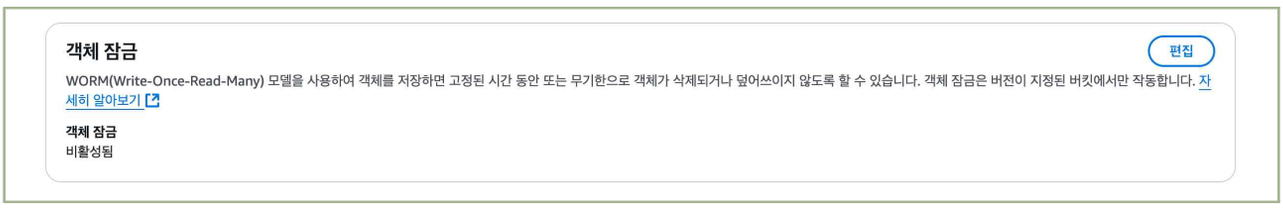
Output: 없음

| 그림 8-5-9 | CLI 명령 – S3 버킷 알림 설정

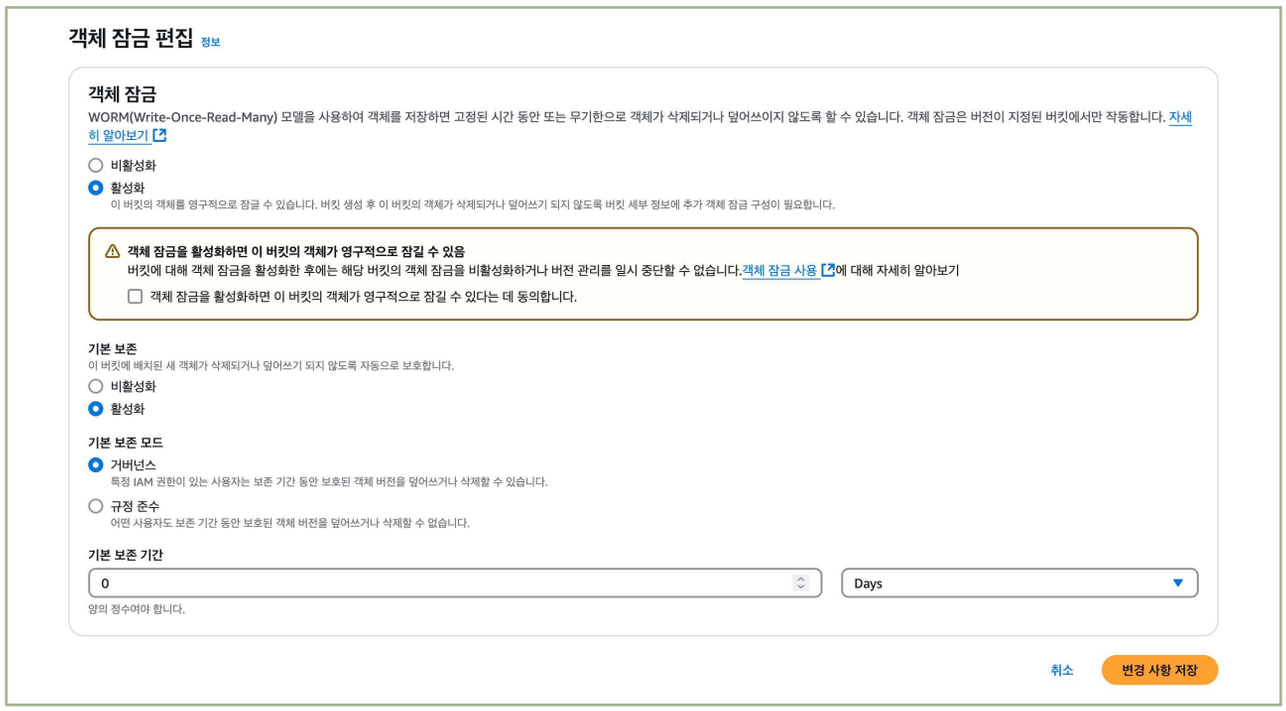
2) 백업 기능(객체 잠금 등)을 통해 무결성 보장 등

- Amazon S3에 저장된 데이터의 무결성 보장이 필요한 경우 객체 잠금 기능을 활성화 하여 무결성을 보장할 수 있습니다. 객체 잠금은 버전이 지정된 버킷에서만 작동하기 때문에 해당 버킷의 버전 관리 활성화가 필요합니다. 기본 보존 모드는 2가지 있으며 '거버넌스' 모드와 '규정 준수' 모드가 있습니다. '거버넌스' 모드의 경우 특정 IAM 권한이 있는 사용자는 보존 기간 동안 보호된 객체 버전을 덮어쓰거나 삭제할 수 있으며 '규정 준수' 모드는 어떠한 사용자도 보존 기간 동안 보호된 객체 버전을 덮어쓰거나 삭제할 수 없습니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → 'Amazon S3 → 'General purpose buckets' → '해당 버킷' → '속성' 에서 객체 잠금 활성화 생성



| 그림 8-5-10 | S3 버킷 객체 잠금 화면



| 그림 8-5-11 | S3 버킷 객체 잠금 편집 화면

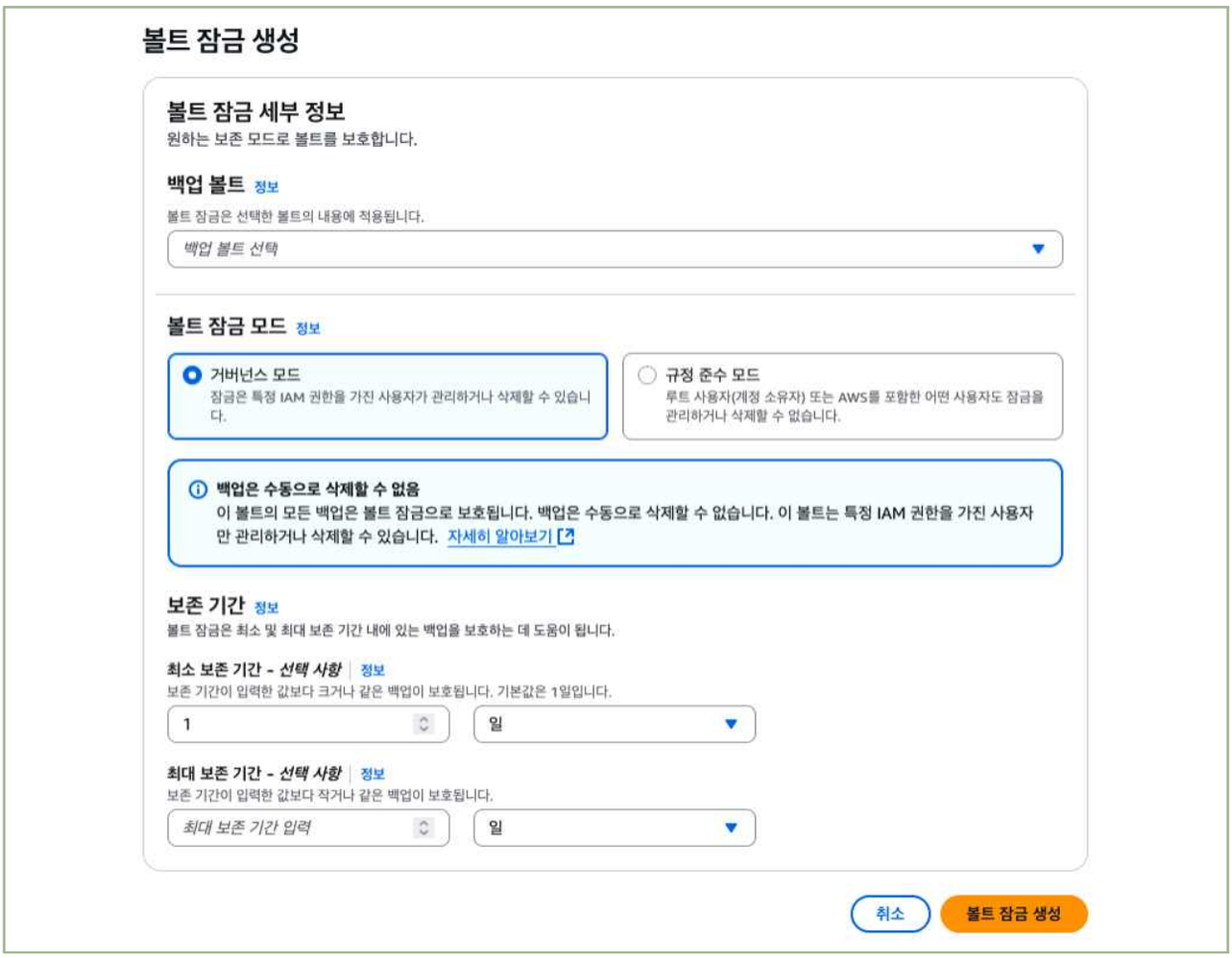
- **(AWS CLI)** put-object-lock-configuration을 사용하여 S3 버킷에 객체 잠금을 설정할 수 있습니다. 버킷 객체 잠금 설정 시에는 해당 버킷의 버전 관리를 먼저 활성화 해야 합니다.

```
aws s3api put-object-lock-configuration \
      --bucket amzn-s3-demo-bucket-with-object-lock \
      --object-lock-configuration '{ "ObjectLockEnabled": "Enabled", "Rule": { "DefaultRetention": {
"Mode": "COMPLIANCE", "Days": 50 }}}'

Output: 없음
```

| 그림 8-5-12 | CLI 명령 – S3 버킷 객체 잠금

- AWS Backup을 사용하는 경우 백업 볼트의 선택적 기능인 백업 볼트 잠금을 사용할 수 있습니다. 볼트 잠금을 생성할 경우, 거버넌스 모드 또는 규정 준수 모드 중 하나를 선택하여 사용할 수 있습니다. 거버넌스 모드는 충분한 IAM 권한을 가진 사용자만 저장소 잠금을 제거할 수 있으며 규정 준수 모드에서는 데이터 보존 기간이 완료될 때까지 저장소 및 저장소의 컨텐츠가 삭제되거나 변경되지 않도록 할 수 있습니다. 또한 보존 기간 만료 이후에도 해당 저장소 복구 시점이 있는 경우 삭제할 수 없으며 저장소가 잠기면 변경이 불가능하여 잠금을 제거할 수 없습니다.(저장소 자체가 비어 있고 복구 시점이 없는 경우에는 삭제 가능) 최대 보존 기간은 36,500일(약 100년) 입니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → 'AWS Backup' → '백업 볼트 잠금' → '볼트 잠금 생성'



| 그림 8-5-13 | 백업 볼트 잠금 생성 화면

- **(AWS CLI)** put-backup-vault-lock-configuration을 사용하여 백업 볼트 잠금을 구성할 수 있습니다. 포함되는 파라미터는 사용하려는 볼트 잠금 모드에 따라 달라집니다. 거버넌스 모드에서 볼트 잠금을 생성하는 경우에는 changeable-for-days를 포함하면 안됩니다. 이 파라미터를 포함하면 규정 준수 모드에서 볼트 잠금이 생성됩니다. 아래 예제는 규정 준수 모드 백업 볼트 잠금 생성의 CLI 예제입니다.

```
aws backup put-backup-vault-lock-configuration \
      --backup-vault-name my_vault_to_lock \
      --changeable-for-days 3 \
      --min-retention-days 7 \
      --max-retention-days 30
```

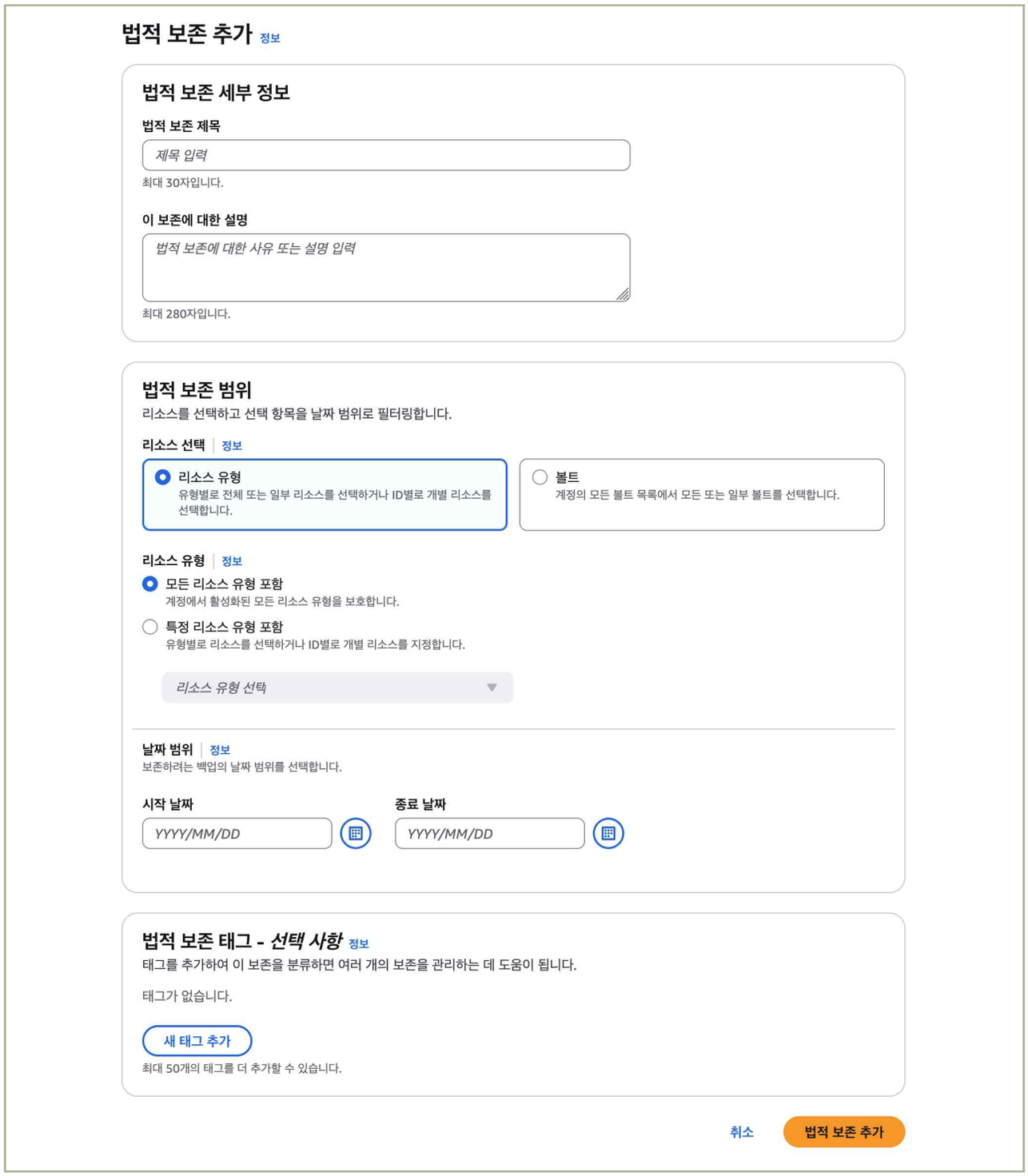
Output: 없음

| 그림 8-5-14 | CLI 명령 - 백업 볼트 잠금

- AWS Backup은 보존 기간 중에 백업이 삭제되는 것을 방지하는데 도움이 되는 법적 보존이라는 관리 도구가 있습니다. 보존이 적용되는 동안에는 보존 상태의 백업을 삭제할 수 없으며, 백업 상태를 변경하는 수명 주기 정책은 법적 보존이 제거될 때까지 연기됩니다. 백업 볼트 잠금은 저장소에 추가

보호 및 불변성을 제공하지만, 법적 보존은 개별 백업(복구 시점)의 삭제를 방지하는 추가 보호를 제공합니다. 법적 보존은 기존 백업(복구 시점)에 추가할 수 있으며 상태가 EXPIRED 또는 DELETING인 백업은 법적 보존에 포함되지 않습니다.다. 상태가 CREATING인 백업은 완료 시점에 따라 법적 보존에 포함되지 않을 수 있습니다.

- **(AWS 관리 콘솔)** '콘솔 홈' → 'AWS Backup' → '법적 보존' → '법적 보존 추가'



| 그림 8-5-15 | 법적 보존 추가 화면

- **(AWS CLI)** create-legal-hold 명령을 사용하여 법적 보존을 생성할 수 있습니다.

```
aws backup create-legal-hold --title "my legal hold" \
      --description "my description" \
      --recovery-point-selection
      "VaultNames=string,DateRange={FromDate=timestamp,ToDate=timestamp}"
```

Output:

```
{
   "Title": "my legal hold",
   "Status": "CREATING",
   "Description": "my description",
   "LegalHoldId": "22978f64-96e1-4131-815b-4e582da7a6b0",
   "LegalHoldArn": "arn:aws:backup:ap-northeast-2:123456789012:legal-hold:22978f64-96e1-4131-
815b-4e582da7a6b0",
   "CreationDate": "2025-05-29T09:10:14.087000+00:00",
   "RecoveryPointSelection": {
      "VaultNames": [
         "string"
      ],
      "DateRange": {
         "FromDate": "2025-01-01T00:00:00+00:00",
         "ToDate": "2025-12-31T23:59:59+00:00"
      }
   }
}
```

| 그림 8-5-16 | CLI 명령 – 법적 보존 생성

4 참고 사항

1 기준

식별번호	기준	내용
8.6.	**백업파일 원격 안전지역 보관**	**중요한 금융회사 전산자료는 보안이 강화된 원격 저장소에 보관하여야 한다.**

2 설명

- 중요한 금융회사 전산자료는 보안이 강화된 원격 저장소에 보관하여야 한다.

  - 예시

    1) 클라우드 서비스 제공자(CSP)의 DR 서비스 이용

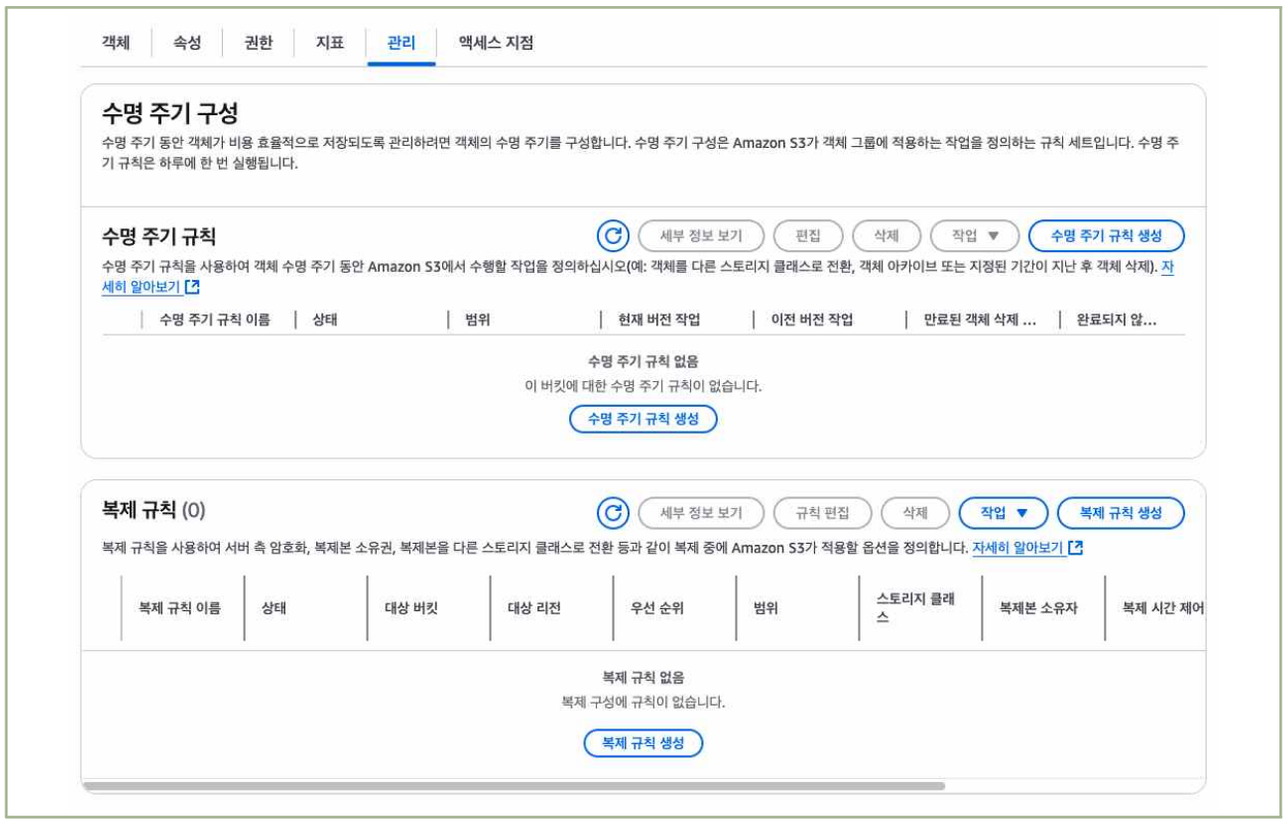
    2) 금융회사 자체 데이터센터로 소산하여 보관 등

3 우수 사례

1) S3를 사용하여 S3 동일 리전 복제(SRR) 혹은 S3 크로스 리전 복제(CRR)로 원격 저장소 보관

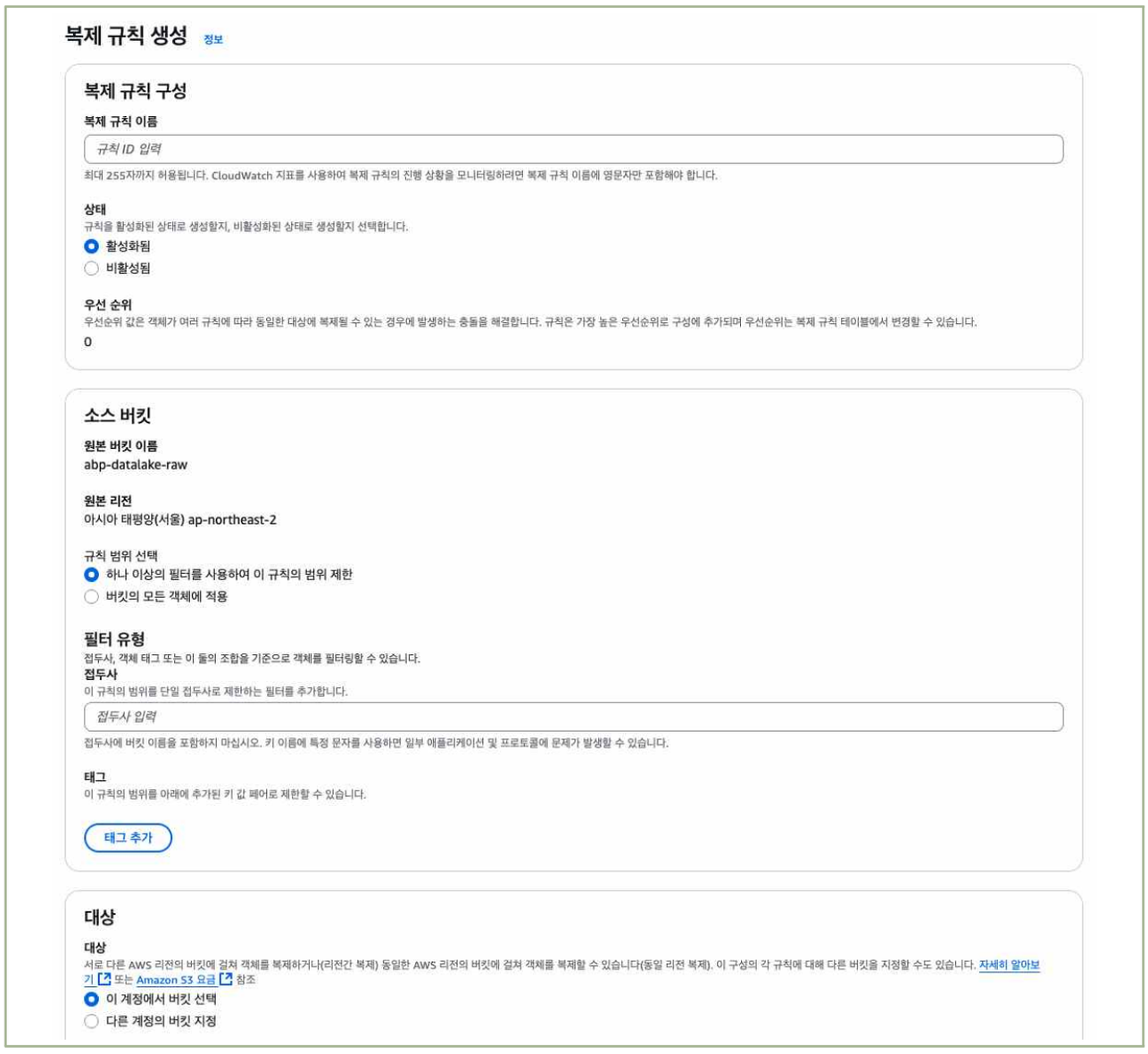
- Amazon S3에 백업 파일을 저장하는 경우 복제를 사용하여 Amazon S3 버킷 전체에 걸쳐 비동기식으로 자동 복사를 할 수 있습니다. 이 때 복제 대상 S3 버킷은 동일한 리전 및 다른 AWS 리전도 가능합니다. 복제를 사용하려면 원본 버킷에 버전 관리가 활성화되어 있어야 하며 복제에는 라이브 복제와 온디맨드 복제의 두 가지 유형이 있습니다. 라이브 복제의 경우 새 객체 및 업데이트된 객체가 원본 객체에 기록될 때 자동으로 복제되는 경우 사용하며 이경우 기존 버킷에 있던 객체는 자동 복제되지 않습니다. 온디맨드 복제는 원본 버킷의 기존 객체를 하나 이상의 대상 버킷으로 복제하며 S3 Batch Replication을 사용하여 복제 합니다.

- **(AWS 관리 콘솔)** ‘콘솔 홈’ → ‘Amazon S3 → ‘General purpose buckets’ → ‘해당 버킷’ → ‘관리 탭’ → ‘복제 규칙 생성’



| 그림 8-6-1 | S3 버킷 관리 화면

- S3 버킷 복제 시 전체 버킷을 저장하는 경우에는 소스 버킷의 규칙 범위 선택에서 '버킷의 모든 객체에 적용'을 선택하고 접두사가 같은 모든 객체를 복제하는 경우 '하나 이상의 필터를 사용하여 이 규칙의 범위 제한'을 선택합니다.



| 그림 8-6-2 | S3 버킷 복제 규칙 생성 화면

- **(AWS CLI)** AWS CLI에서는 복제 구성을 JSON으로 지정해야 합니다. 아래 샘플을 참고하여 replication.json 파일로 저장 후 put-bucket-replication 명령을 실행하여 소스 버킷에 복제 구성을 추가할 수 있습니다. 이 때 반드시 소스 버킷 이름을 제공해야 합니다.

```
aws s3api put-bucket-replication \
      --bucket amzn-s3-demo-bucket1 \
      --replication-configuration file://replication.json
```

Contents of replication.json:

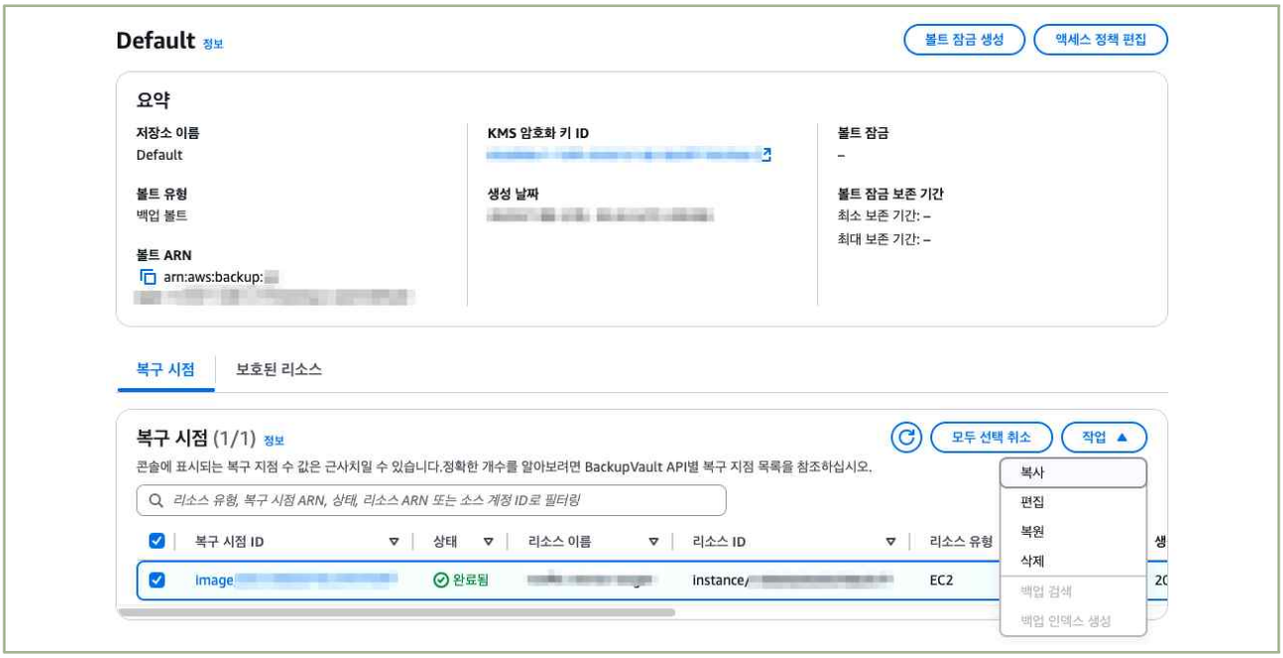
```
{
   "Role": "arn:aws:iam::123456789012:role/s3-replication-role",
   "Rules": [
      {
         "Status": "Enabled",
         "Priority": 1,
         "DeleteMarkerReplication": { "Status": "Disabled" },
         "Filter" : { "Prefix": ""},
         "Destination": {
            "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket2"
         }
      }
   ]
}
```

Output: 없음

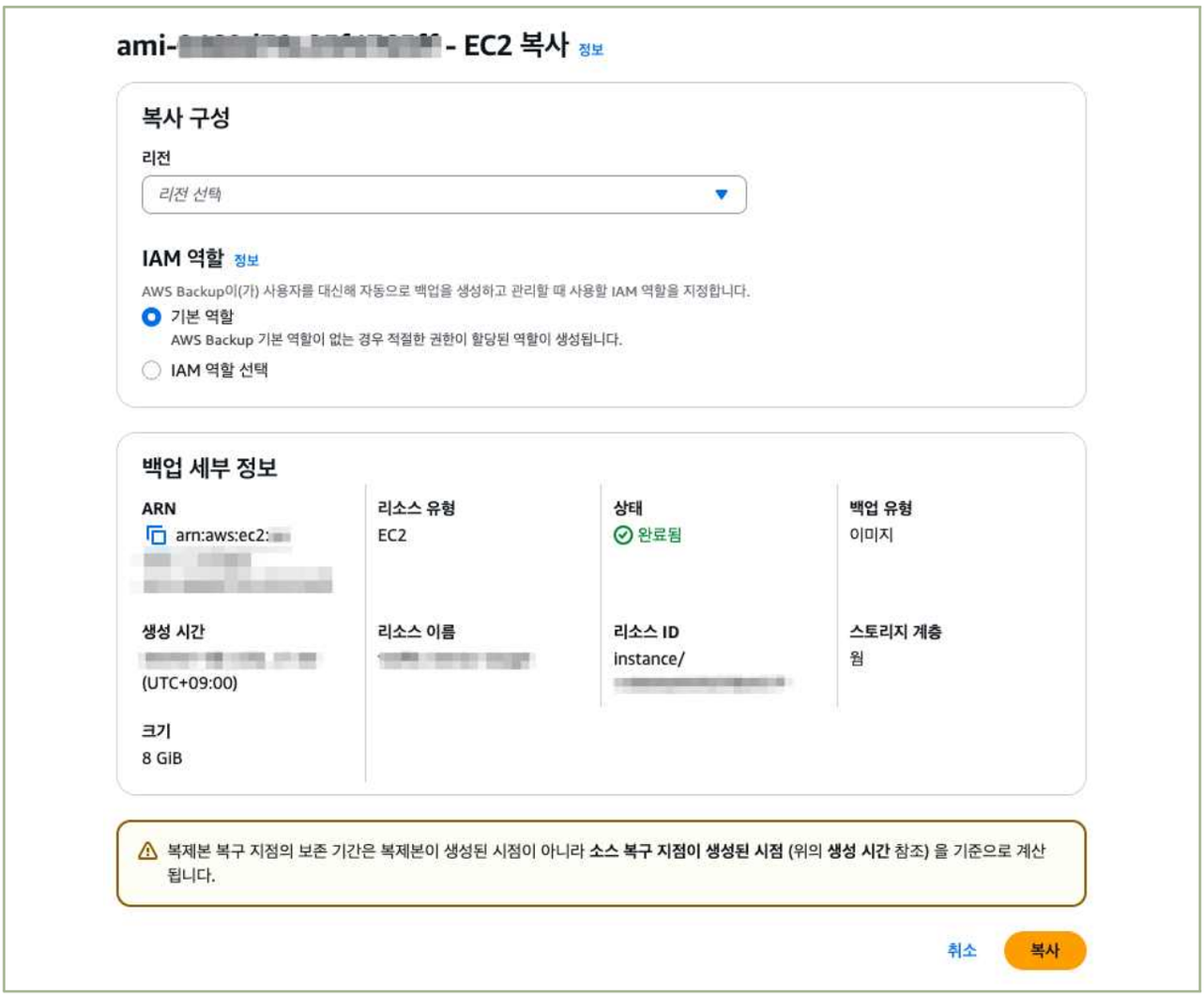
| 그림 8-6-3 | CLI 명령 – 동일 리전에서 S3 버킷 객체 라이브 복제

2) AWS Backup을 사용하여 백업 파일 원격 저장소 보관

- AWS Backup은 백업이 완료된 복구 시점을 AWS 다른 리전에 추가로 복사할 수 있습니다. 복구 시점을 선택하고 작업 버튼을 누르면 추가 메뉴가 나오며 '복사'를 클릭하면 복사 화면으로 이동합니다. 복사 구성에서 리전을 선택하여 다른 AWS 리전에 추가로 복사할 수 있습니다.
- **(AWS 관리 콘솔)** '콘솔 홈' → 'AWS Backup → '볼트' → '이 계정이 소유한 볼트 탭에서 해당 백업 볼트 선택' → '복구 시점 탭' → '해당 복구 시점 선택하고 작업 드롭다운 버튼에서 복사'



| 그림 8-6-4 | AWS Backup 볼트 복구 시점 화면



| 그림 8-6-5 | AWS Backup 볼트 복구 시점 복사 화면

4 참고 사항

- AWS Backup은 크로스 리전 간 백업 외에도 AWS Organizations에서 동일한 조직에 속하는 두 개의 계정이 있는 경우 다른 AWS 계정에 백업할 수 있는 계정 간 백업을 사용할 수 있습니다.
- AWS Elastic Disaster Recovery (AWS DRS)는 기업의 중요한 IT 시스템을 AWS 클라우드로 신속하고 안정적으로 복구할 수 있게 해주는 서비스입니다. 이 서비스는 온프레미스 및 클라우드 기반 애플리케이션의 복구를 간소화하고 비용을 절감하며 규정 준수를 개선하는 데 도움을 줍니다. AWS DRS는 소스 서버의 데이터, 애플리케이션 및 시스템 구성을 AWS로 지속적으로 복제하고 이를 통해 최신 상태의 백업을 유지할 수 있으며 사전 정의된 복구 계획에 따라 자동으로 시스템을 복구할 수 있어 인적 오류를 줄이고 복구 시간을 단축할 수 있습니다.

1 기준

식별번호	기준	내용
8.7.	주요 전산장비 이중화	**금융회사는 클라우드 환경을 통한 인프라 구성 시 주요 전산장비를 이중화 하여야 한다.**

2 설명

- 금융회사는 클라우드 환경을 통한 인프라 구성 시 주요 전산장비를 이중화 하여야 한다.

  - 예시

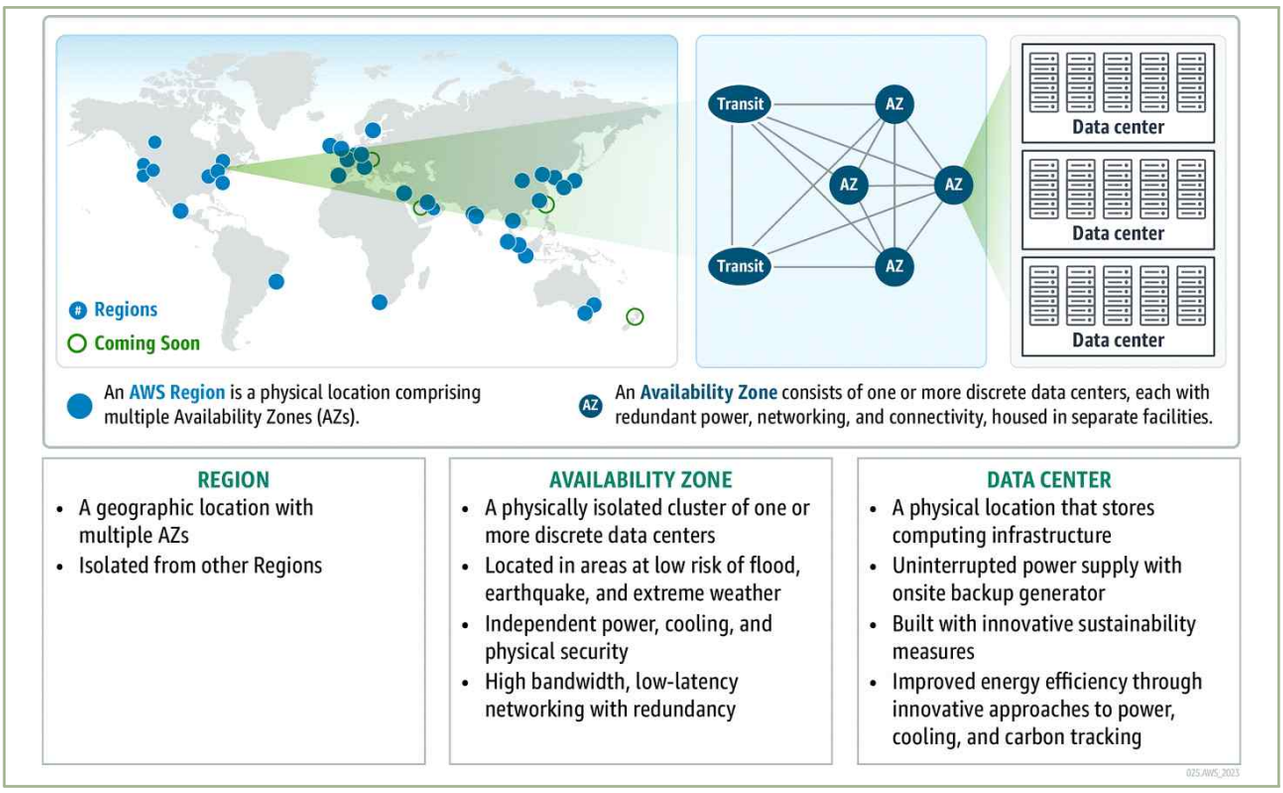
    1) 클라우드 가상화 기능을 이용하여 주요 전산장비(서버, 데이터베이스 등) 이중화 구성

    2) 이중화 구성 시 원격 안전지역 고려 등

3 우수 사례

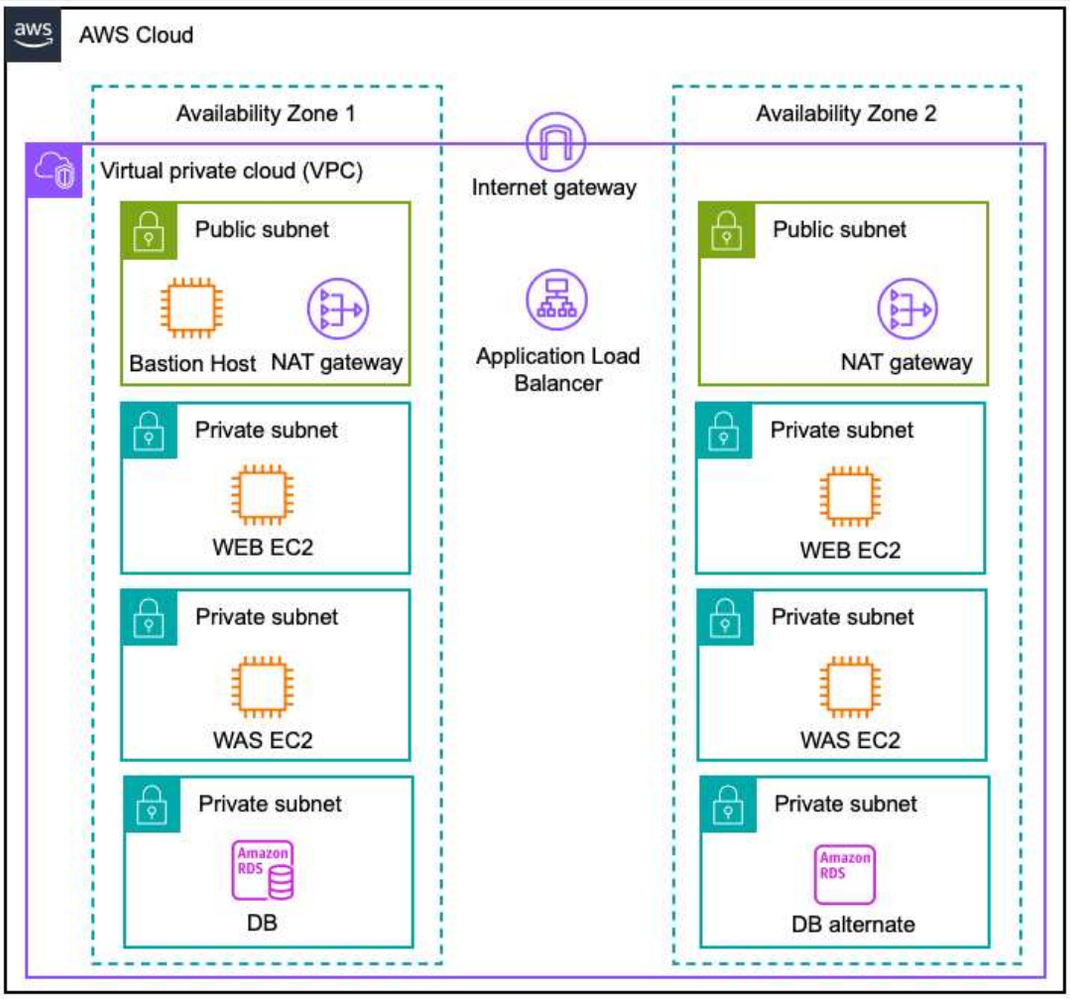
1) AWS AZ(가용 영역)로 전산장비 이중화

- AWS에는 리전이라는 개념이 있고 AWS가 전 세계에서 데이터 센터를 클러스터링하는 물리적인 위치를 의미합니다. AZ(가용 영역)은 AWS 리전의 중복 전력, 네트워킹 및 연결이 제공되는 하나 이상의 개별 데이터 센터로 구성됩니다. 각 AWS 리전은 지리적 영역 내에서 격리되고 물리적으로 분리된 최소 3개의 AZ로 구성되며 AZ는 고가용성을 위한 애플리케이션 분할을 용이하게 합니다. 애플리케이션 하나를 여러 AZ에 걸쳐 분할하면 기업의 격리가 더 원활하게 이루어지며 정전, 낙뢰, 토네이도, 지진 등과 같은 문제로부터 안전하게 보호됩니다. AZ는 다른 모든 AZ와 수 킬로미터에 상당하는 유의미한 거리를 두고 물리적으로 분리되어 있습니다. 다만 모든 AZ는 서로 100km(60마일) 이내의 거리에 위치합니다.



| 그림 8-7-1 | AWS 리전 및 가용 영역 아키텍처

- 아래 그림은 AWS에서 일반적으로 구성하는 3티어 아키텍처의 예시를 보여줍니다. 멀티 AZ를 기반으로 웹/애플리케이션/데이터베이스 계층을 이중화하여 하나의 가용 영역에 장애가 발생해도 서비스가 지속적으로 가능합니다. AWS 클라우드에서 웹 티어 계층은 Elastic Load Balancing(ELB)를 사용하여 하나 이상의 가용 영역에 있는 여러 대상 및 가상 어플라이언스에서 들어오는 애플리케이션 트래픽을 분산할 수 있어 기존의 웹서버를 대체할 수 있습니다. 이러한 멀티 AZ 기반의 아키텍처는 현대적인 애플리케이션에 필수적인 고가용성, 확장성, 안정성을 제공하며 특히 금융, 의료, 전자상거래와 같은 미션 크리티컬한 시스템에서 더욱 의미를 가집니다. AWS는 리전과 가용 영역을 기반으로 비즈니스 연속성을 보장하며 안정적인 서비스를 제공하는 글로벌 인프라를 제공하고 있습니다.



| 그림 8-7-2 | AWS 3 Tire 이중화 아키텍처 예제

4 참고 사항

- 글로벌 서비스 구축을 위한 AWS 멀티 리전 아키텍처 구성 가이드

금융분야 상용 클라우드컴퓨팅서비스 보안 관리 참고서 (Amazon Web Services)

발 행 일	2025년 10월
발 행 인	금융보안원(원장 박상원)
공 동 발 행 인	Amazon Web Services

금 융 보 안 원		AWS	
클라우드대응부	부장 김제광	**보안전문 솔루션즈아키텍트**	수석 신은수
클라우드기획팀	팀장 장지현		선임 황재훈
	차장 정희선		선임 정관진
	과장 김용규	**솔루션즈아키텍트**	선임 성문규
	과장 안성현		
	과장 마승영		
	대리 최주섭		
	대리 송창석		
	주임 전동현		
	주임 전하은		

발 행 처	금융보안원
	02-3495-9000
	경기도 용인시 수지구 대지로 132

〈비 매 품〉

금융분야
상용 클라우드컴퓨팅서비스 보안 관리 참고서