[AWS] 클라우드 콘솔 보안 진단

AWS 콘솔진단 한번 해봅시다.

* 본 가이드는 2024 SK 쉴더스 AWS 클라우드 보안 가이드를 토대로 합니다.

https://www.skshieldus.com/notice/boangaideu-2024-keulraudeu-boangaideu-balgan-aws-azure-gcp

일단 짧고 간결하게 진행합니다.

따로 공부하느라 무료 플랜 계정을 전부 소비해서 Ui만 참고하시면 됩니다.

1.1. 사용자 계정 관리 [시작]

취약점 설명

AWS에서는 계정마다 권한을 부여하거나, 권한들의 집합인 **Role** 을 부여할 수 있습니다.

* **Role**을 통해 사용자 권한을 관리하는 방식을 **RBAC (Role Based Access Control)** 라고 합니다.

AWS에는 수 많은 서비스 **(200**개 이상**)**가 존재하기 때문에 권한 관리는 굉장히 중요합니다.

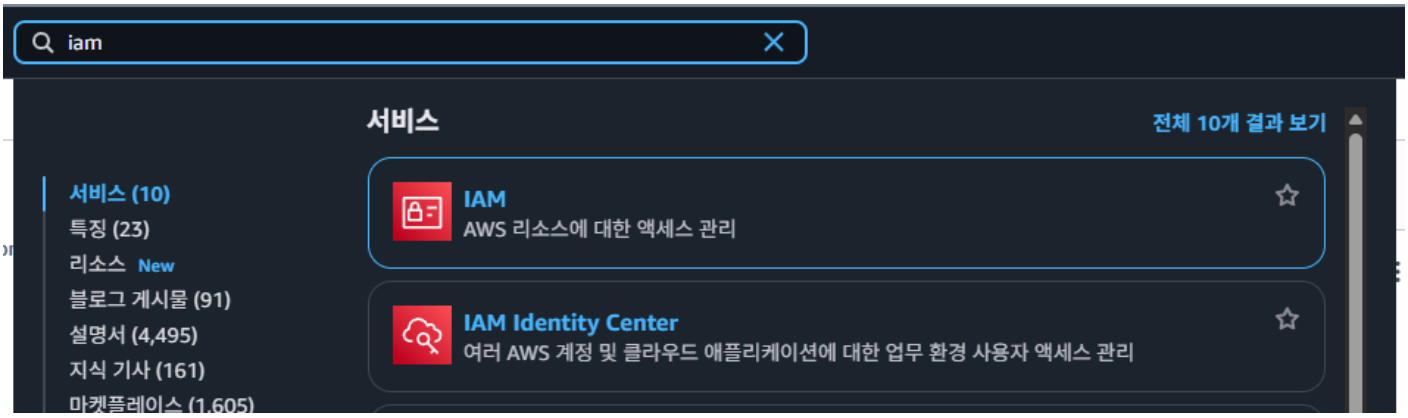
AWS에서는 한 파트에 대한 모든 권한을 관리하는 **Full Access** 권한이 존재함으로 필요에 따라 권한이 부여되어 있는지 확인합니다.

딴소리

진단을 인터뷰로 하는 경우도 있지만 그냥 **Read-Only** 계정 하나 파서 직접 진단하라 그러는 곳도 있습니다.

진단 방법

01. IAM 서비스 진입



IAM
AWS 리소스에 대한 액세스 관리

IAM Identity Center
여러 AWS 계정 및 클라우드 애플리케이션에 대한 업무 환경 사용자 액세스 관리

02. 사용자 정보 확인

- 원래같으면 여러 계정들이 있겠죠?

- 담당자분께 각 사용자들의 대한 권한 관리를 하고 계신지 여쭤보면 됩니다.

- 모든 보안 진단이 그렇듯 계정 별로 필요한 최소한의 계정을 부여하는 것이 중요합니다.

IAM > 사용자

Identity and Access Management(IAM)

IAM 검색

대시보드

액세스 관리

사용자 그룹

사용자

역할

정책

ID 제공업체

계정 설정

루트 액세스 관리 신규

사용자 (0) 정보

IAM 사용자는 계정에서 AWS와 상호 작용하는 데 사용되는 장기 자격 증명을 가진 자격 증명입니다.

검색

☐	사용자 이름	경로	그룹	마지막 활동	MFA	암호 수명	콘솔 마지막 로그인
				표시할 리소스 없음			

양호 취약 기준

진단 기준	**양호기준** : 관리자 권한을 보유한 다수 계정이 존재하지 않고 불필요한 계정이 존재하지 않을 경우 **취약기준** : 관리자 권한을 보유한 다수 계정이 존재하거나 불필요한 계정이 존재할 경우

질문 : 사용자의 역할에 따라 권한 관리를 진행하고 계시나요?

답변

- 각 부서에 맞게 최소 권한을 부여하여 관리하고 있습니다. -> 양호

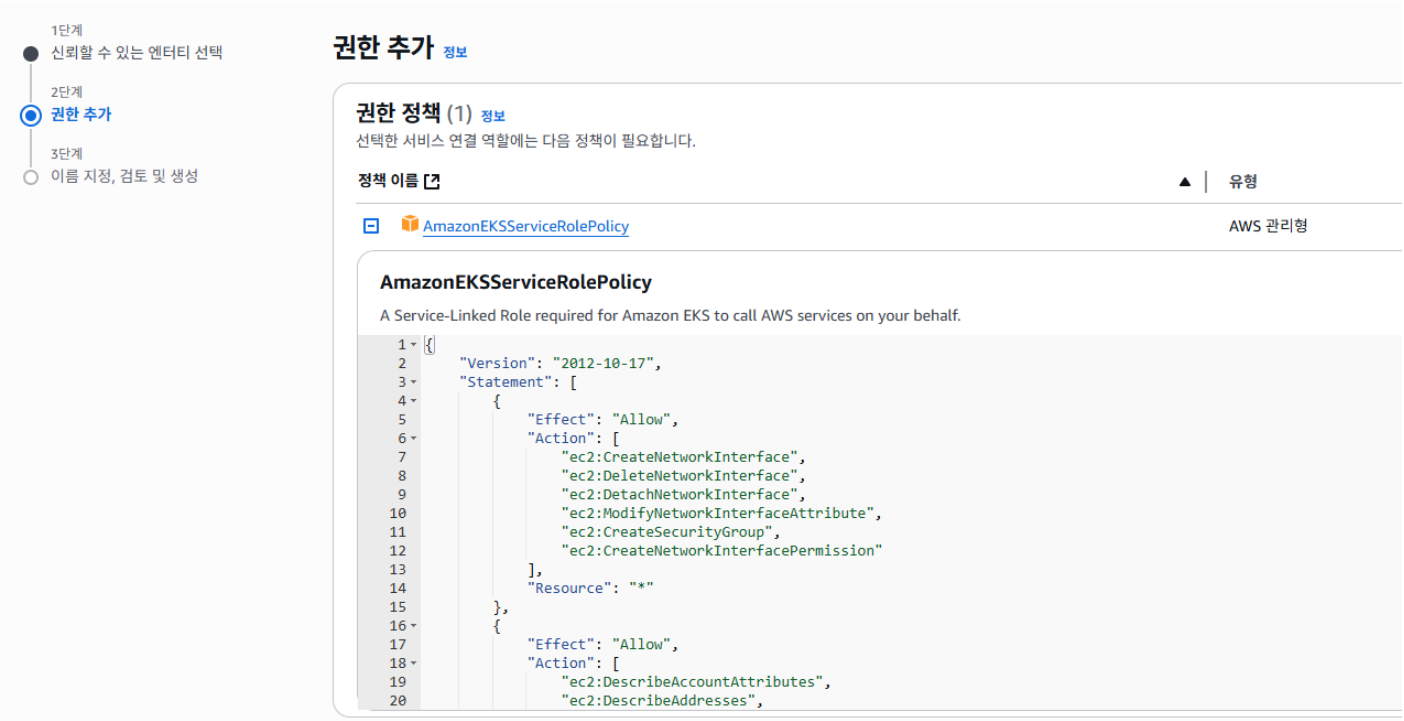
- 관리하긴 하는데 몇몇 불필요한 **FullAccess** 권한이 보이네요 -> 취약

- 따로 권한 관리는 안하고있습니다. -> 취약

참고

1. **(Role)** .

, Json .



2. SSO .

1.2. IAM 사용자 계정 단일화 관리

취약점 설명

사용자 계정은 **1**인 **1**계정 원칙을 우선으로 합니다.

한 계정에 대해 여러명이 같이 쓸 경우 로그를 분석할때 문제가 생기므로 **1**인 **1**계정을 사용하여야 합니다.

딴소리

1. 보안 진단 인터뷰 할때 중요한 점으로는 담당자를 납득시키는 점이라고 생각합니다.

이거 설정하시고 저것도 설정하셔야 되요. 라고 말했을때 담당자분이 **"**왜 그걸 설정해야되요**?"** 라고 했을때

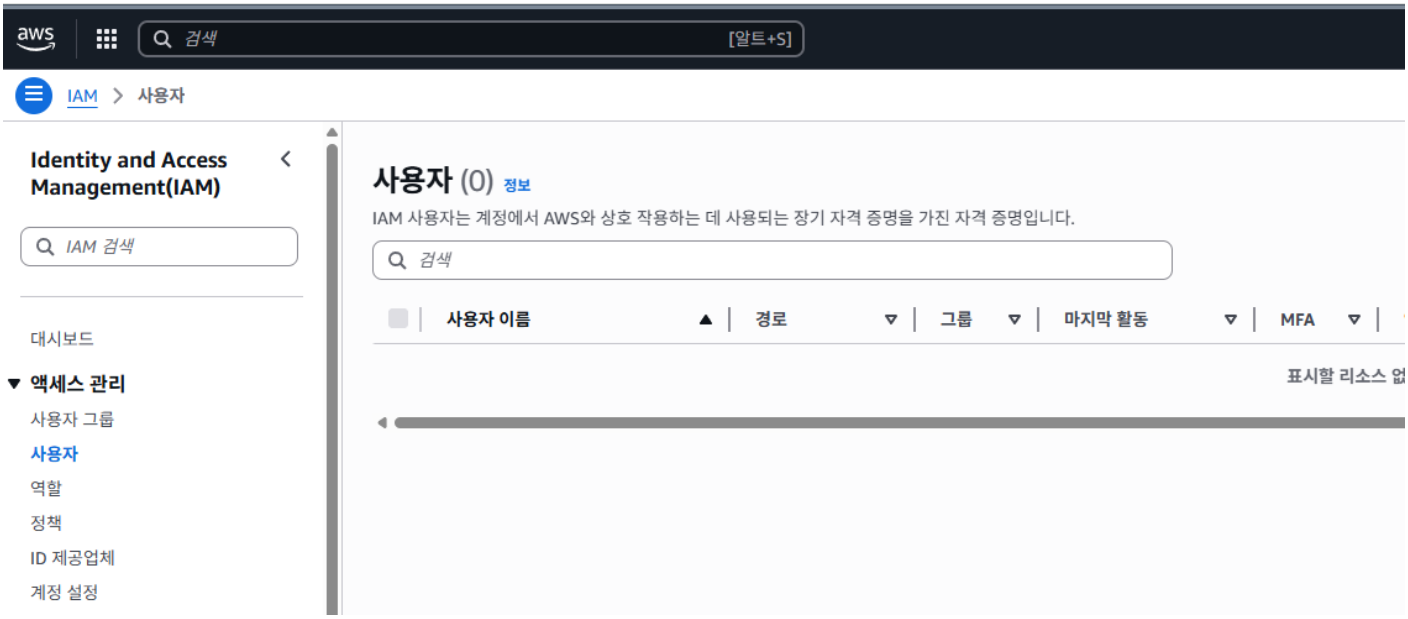
답변 할 줄 알아야합니다.

2. 사용자 계정이 관리자분 한명밖에 없을 경우 양호처리 하면 됩니다.

(1인당 **1**개의 계정을 사용하는게 맞으므로**)**

진단 방법

1. IAM 서비스로 진입하여 사용자 목록을 확인



사용자 (0)

IAM 사용자는 계정에서 AWS와 상호 작용하는 데 사용되는 장기 자격 증명을 가진 자격 증명입니다.

☐	사용자 이름	경로	그룹	마지막 활동	MFA

표시할 리소스 없

2. 담당자분께 **1**명당 **1**개의 계정을 사용하고 있는지 인터뷰

양호 취약 기준

진단 기준	양호기준 : IAM 사용자 계정을 1인 1계정으로 사용하고 있는 경우 취약기준 : IAM 사용자 계정을 1인 1계정으로 사용하고 있지 않은 경우

사용자마다 개별적으로 계정을 생성하여 사용하고있습니다 **. ->** 양호

관리자 계정 하나를 나눠서 쓰고있습니다 **. ->** 취약

참고

SSO를 사용하는 경우 계정 관리에 대한 부분들이 해당없음으로 처리되는 경우가 있습니다 **.**

1.3. IAM 사용자 계정 식별 관리

취약점 설명

계정별로 만들어진 목적을 네임태그에 적어놓고 관리하는지 확인합니다.

계정이 여러개 일때 관리를 용이하게 하기 위함입니다.

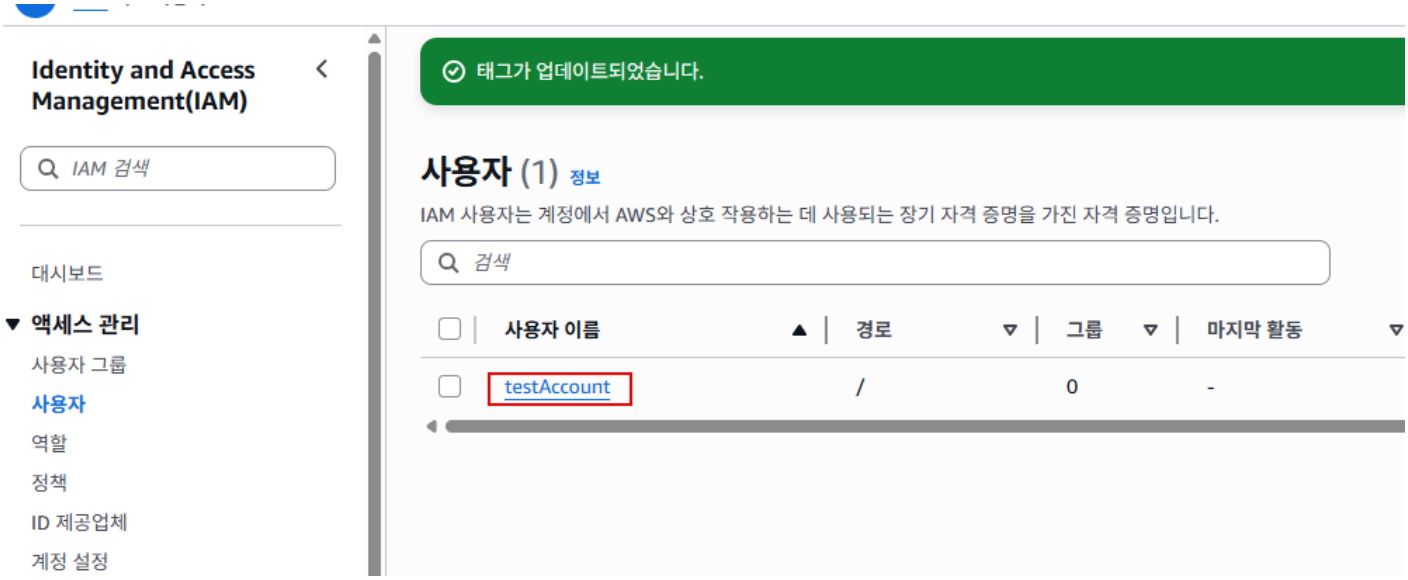
딴소리

SK 가이드에만 포함되어 있는 가이드입니다.

안해두면 위험하다! 보다는 가용성, 관리용이에 대한 내용 같습니다.

진단 방법

01. 임의의 사용자로 진입합니다.



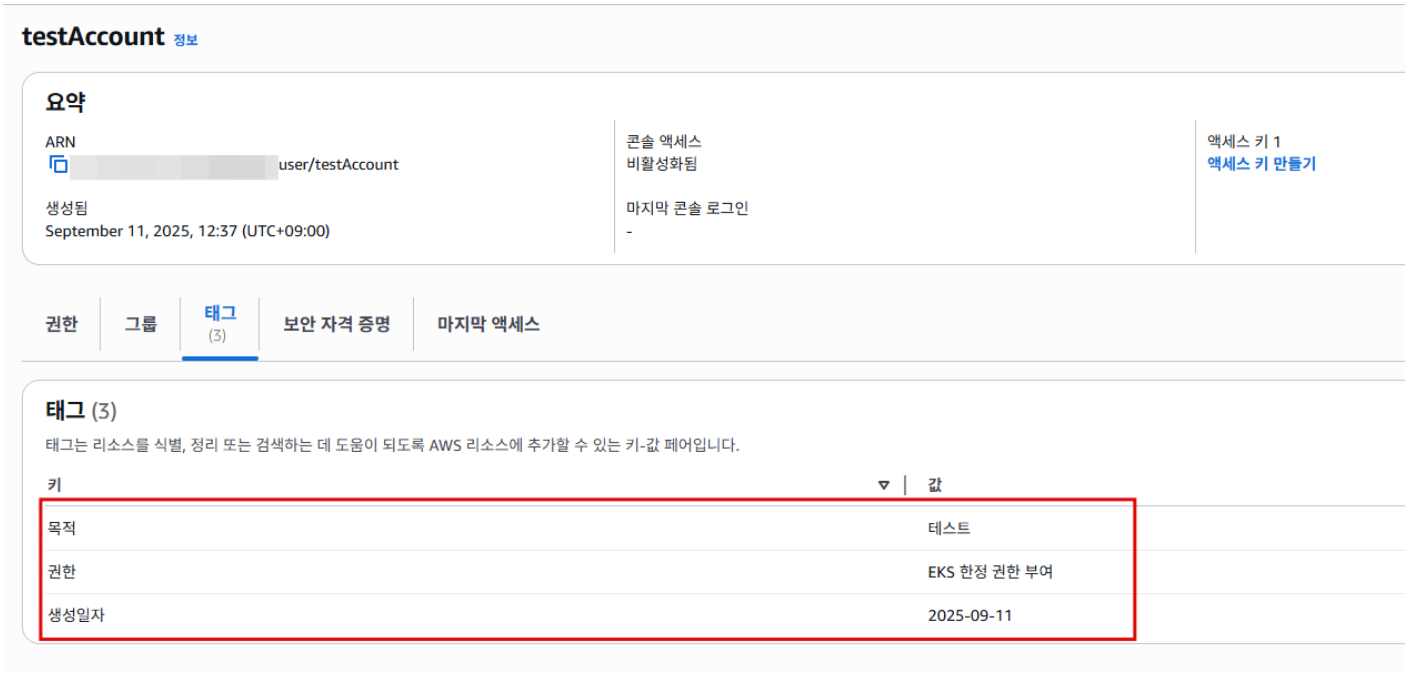
태그가 업데이트되었습니다.

사용자 (1) 정보

IAM 사용자는 계정에서 AWS와 상호 작용하는 데 사용되는 장기 자격 증명을 가진 자격 증명입니다.

☐	사용자 이름 ▲	경로 ▽	그룹 ▽	마지막 활동 ▽
☐	testAccount	/	0	-

02. '태그' 탭으로 진입하여 사용자 설명이 존재하는지 확인합니다.



testAccount 정보

요약

ARN
[redacted]user/testAccount

생성됨
September 11, 2025, 12:37 (UTC+09:00)

콘솔 액세스
비활성화됨

마지막 콘솔 로그인
-

액세스 키 1
액세스 키 만들기

태그 (3)

태그는 리소스를 식별, 정리 또는 검색하는 데 도움이 되도록 AWS 리소스에 추가할 수 있는 키-값 페어입니다.

키	값
목적	테스트
권한	EKS 한정 권한 부여
생성일자	2025-09-11

임의로 태그를 넣어놓았으나, 방식은 담당자마다 상이합니다.

양호 취약 기준

진단 기준	양호기준 : 사용자 정보(이름, 이메일, 부서 등)가 IAM 사용자 태그에 설정되어 있을 경우 취약기준 : 사용자 정보(이름, 이메일, 부서 등)가 IAM 사용자 태그에 설정되어 있지 않을 경우
비고	Organizations 서비스 사용을 통해 계정을 관리하는 경우 AD 계정을 연동하여 사용하기 때문에 계정 정보를 태그 하지 않아도 양호로 처리될 수 있음

질문

계정 별로 태그를 지정하여 관리하고 계신가요

답변

계정 별로 태그를 지정해서 어떤 목적의 계정인지 식별하고 있습니다. -> 양호

따로 태그 지정 안했습니다. -> 취약

참고

SSO를 사용하는 경우 계정 관리에 대한 부분들이 해당없음으로 처리되는 경우가 있습니다.

1.4. IAM [illegible] [illegible] [illegible] [illegible]

[illegible] [illegible]

IAM [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible].

[illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible], [illegible] [illegible] [illegible] [illegible] [illegible] [illegible].

[illegible]

[illegible] [illegible] [illegible] [illegible], **RBAC[illegible] [illegible] [illegible] [illegible] [illegible].**

[illegible] [illegible] [illegible] [illegible].

[illegible] [illegible] EKS [illegible] [illegible]

EKS Manager [[illegible]] -> EKS [illegible], [illegible] [illegible]

manager A[[illegible]],

manager B[[illegible]],

manager C[[illegible]]

***EKS [illegible] [illegible] [illegible] EKS Manager[illegible] [illegible] [illegible] [illegible] [illegible] [illegible].**

[illegible]

RBAC[illegible] [illegible] EKS [illegible] [illegible]

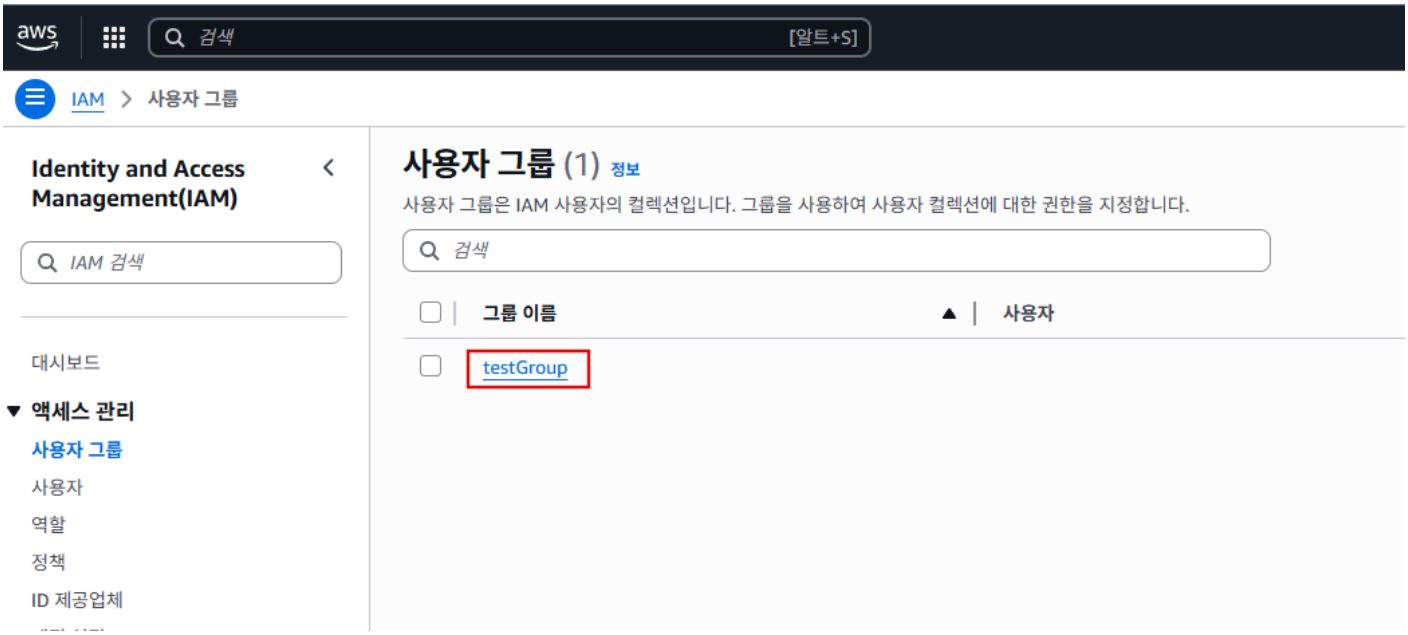
manager A[[illegible]] - EKS Viewer[[illegible]] -> EKS [illegible] [illegible] [illegible]

manager B[[illegible]] - EKS Viewer[[illegible]], EKS Writer[[illegible]] -> EKS [illegible] [illegible], [illegible] [illegible]

manager C[사용자**] - EKS Admin[**역할**] -> EKS** 관련 읽기, 쓰기, 삭제, 공유 등 모든 기능에 대해 가능

> 진단 방법

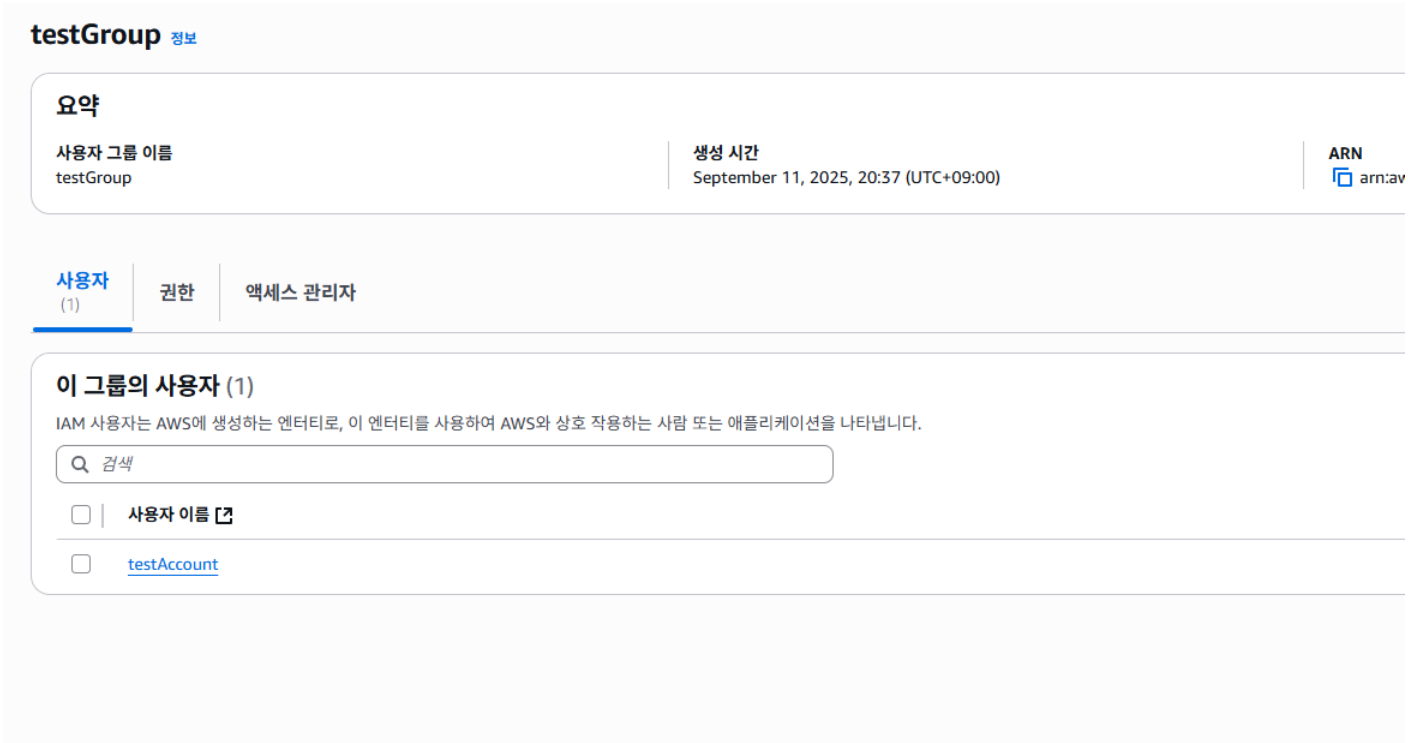
01. [IAM] -> [액세스 관리] -> [사용자 그룹] -> [임의의 그룹] 진입



사용자 그룹은 IAM 사용자의 컬렉션입니다. 그룹을 사용하여 사용자 컬렉션에 대한 권한을 지정합니다.

☐	그룹 이름	사용자
☐	testGroup	

02. 사용자 탭에서 불필요한 계정이 존재하는지 식별 (테스트, 퇴사자, 장기 미사용 계정)



사용자 그룹 이름: testGroup

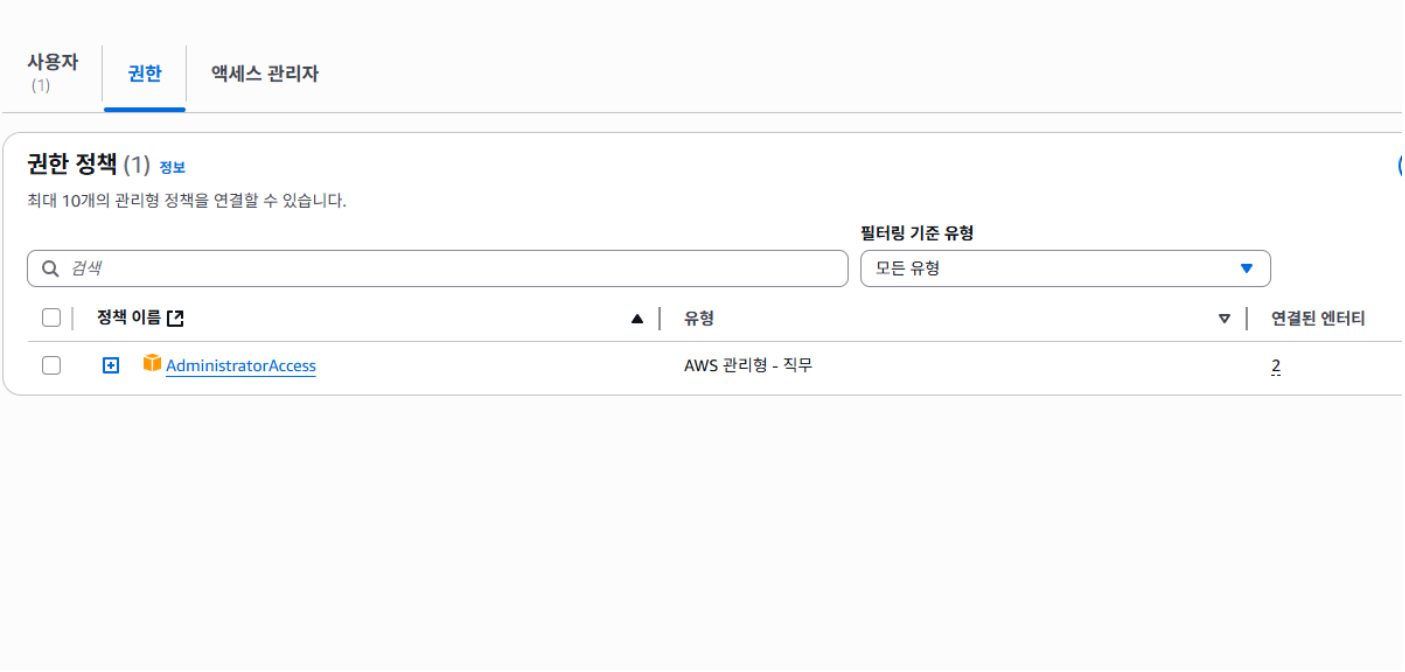
생성 시간: September 11, 2025, 20:37 (UTC+09:00)

ARN: arn:av

IAM 사용자는 AWS에 생성하는 엔터티로, 이 엔터티를 사용하여 AWS와 상호 작용하는 사람 또는 애플리케이션을 나타냅니다.

☐	사용자 이름
☐	testAccount

03. 권한 탭으로 이동, 과도하게 부여된 계정이 없는지, 사용자 역할에 따라 최소한의 권한을 부여하고 있는지 확인한다.



권한 정책 (1)

최대 10개의 관리형 정책을 연결할 수 있습니다.

☐	정책 이름	유형	연결된 엔터티
☐	AdministratorAccess	AWS 관리형 - 직무	2

양호 취약 기준

진단 기준	**양호기준** : IAM 그룹에 포함된 사용자 계정 중 불필요한 계정이 존재하지 않을 경우 **취약기준** : IAM 그룹에 포함된 사용자 계정 중 불필요한 계정이 존재할 경우

질문 : 사용자 그룹에 불필요 계정은 없을까요?, 용자 역할에 맞게 권한을 부여하여 관리하고 계실까요?

답변 :

사용자가 맡은 직무에 맞게 권한을 최소한으로 부여하여 사용 중입니다. - 양호

그룹은 사용하지 않고 **RBAC**으로 권한을 별도 관리하고 있습니다. - (최소한의 권한을 부여하여 사용할 경우)양호

사용자에게 모두 동일한 권한을 부여해서 사용합니다 - 취약

참고

처음 인터뷰 들어갔을땐 **RBAC** 뭔가 했다 . 그냥 역할 기반 접근제어 였던것 .

1.5. Key Pair 접근 관리

취약점 설명

EC2에 접근할때 인증하는 방법으로 **Key Pair**를 사용할 수 있다.

단순히 패스워드를 치고 들어가는 것 보다 보안 성이 뛰어나지만, 해당 파일이 유출 및 탈취될 경우 바로 **EC2**에 진입 할 수 있으므로 보관에 유의하여야 한다.

이번 진단 항목은 **Key Pair**를 사용해 **EC2** 인스턴스에 접근하고있는지에 대한 항목이다.

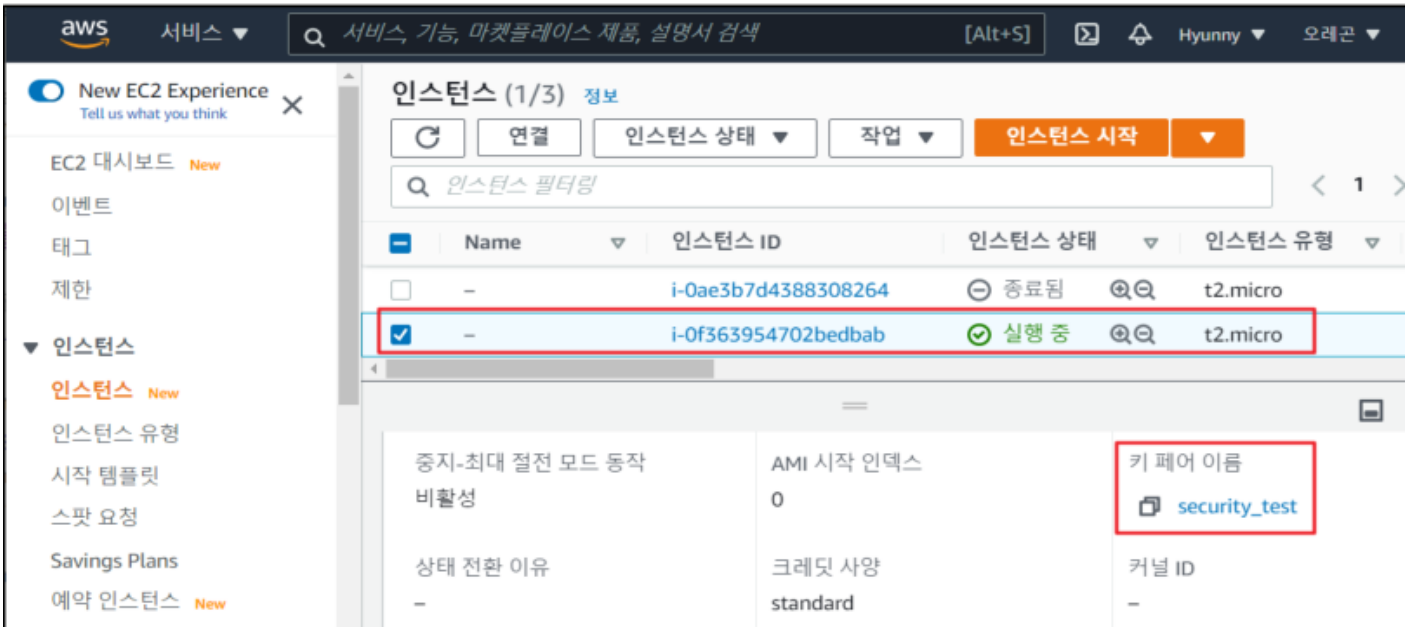
딴소리

KeyPair 보관 및 관리 방법은 바로 다음 항목이다. 검사검사 같이 인터뷰

진단 방법

01. EC2로 진입할 때 **Key Pair**를 사용하고 있는지 여부를 인터뷰하면 된다.

*아래 사진은 **SK** 쉴더스 **AWS** 진단 가이드 에서 가져옴



인스턴스 (1/3)

	Name	인스턴스 ID	인스턴스 상태	인스턴스 유형
☐	–	i-0ae3b7d4388308264	종료됨	t2.micro
☑	–	i-0f363954702bedbab	실행 중	t2.micro

중지-최대 절전 모드 동작
비활성

AMI 시작 인덱스
0

키 페어 이름
security_test

상태 전환 이유
–

크레딧 사양
standard

커널 ID
–

인스턴스 정보에 키 페어가 할당되어 있는지 확인

양호 취약 기준

진단 기준	**양호기준** : Key Pair(PEM)를 통해 EC2 인스턴스에 접근할 경우 **취약기준** : Key Pair(PEM)가 아닌 일반 패스워드로 EC2 인스턴스에 접근할 경우

질문 : **EC2** 인스턴스에 접근할 때 **Key Pair**를 사용하고 계실까요 ?

답변 :

네 **Key Pair** 통해서 접근 중입니다. - 양호

일반 패스워드 써서 진입하고 있습니다. - 취약

참고

1.6. Key Pair 보관 관리

취약점 설명

Key Pair를 통해 **EC2**에 접근할 경우 일반 패스워드 보다 훨씬 보안성은 높지만 키 페어 파일이 유출, 탈취될 경우 공격에 취약할 수 있다.

* 쉽게 말하면 비밀번호를 적어둔 텍스트 파일이라고 생각하면 된다.

파일의 누출을 방지하기 위해 **Keypair** 파일을 예상하기 힘든 경로, 혹은 **S3** 버킷, **USB** 별도 관리 등 접근하기 어려운 장소에 보관하여야 한다.

딴소리

Key Pair가 유출, 탈취된 것을 인지하지 못할 수 있으므로 주기적으로 **Key Pair**를 재 발급하여 사용해야 한다.

진단 방법

01. 인터뷰를 통해 **Key Pair**를 어떤 방식으로 보관하고 있는지, 접근이 어려운 위치에 보관하고 있는지 여쭤본다.

양호 취약 기준

진단 기준	**양호기준** : Key Pair(PEM) File의 보관 위치가 쉽게 유추할 수 없는 공간에 보관되어 있을 경우 **취약기준** : Key Pair(PEM) File의 보관 위치가 다수 접근이 가능한 공용 공간(퍼블릭 S3, EC2 "Admin Console(/)" 디렉터리 등)에 보관되어 있을 경우

질문 : **Key pair**를 예측하기 어려운 위치에 보관하고 계신가요? 어떤 방식으로 **Key Pair**를 보관하고 계신가요?

답변 :

usb에 보관하고 있습니다 . - 양호

예상하기 어려운 복잡한 경로에 보관하고 있습니다 . - 양호

프라이빗 S3 버킷에 보관하고 있습니다 . - 양호

생성자 컴퓨터에 바탕화면에 두고 있습니다 . - 취약

퍼블릭 S3에 보관하고 있습니다 - 취약

> 참고

1.7. Admin Console 관리자 정책 관리

취약점 설명

AWS Console에 가입하게 되면 가장 처음으로 생성하게 되는 계정은 모든 서비스에 접근할 수 있는 최고 관리자 계정이다.

root 계정이라고 보통 부르며, 모든 권한을 가지고 있기 때문에 서비스 운영에는 사용하지 않는 것이 좋다.

진단 방법

인터뷰를 통해 **Admin Console** 계정을 서비스 용도로 사용하고 있지 않은지 확인 한다.

양호 취약 기준

진단 기준	**양호기준** : Admin Console 계정을 서비스 용도로 사용하지 않는 경우 **취약기준** : Admin Console 계정을 서비스 용도로 사용하는 경우

질문 : **AWS Console**의 **root** 계정을 서비스 운영에 사용하고 계신가요?

답변 :

아니요 사용하고 있지 않고 필요에 따라 **IAM** 계정을 생성해 최소한의 권한을 부여하여 사용하고 있습니다. -> 양호

답변 :

네 보통 **root** 계정으로 전체적인 서비스를 관리하고 있습니다 **. ->** 취약

> 참고

1.8. Admin Console 계정 Access Key 활성화 및 사용주기 관리

취약점 설명

Access key는 **AWS**에서 사용하는 인증수단으로 해당 계정의 액세스 키를 가지고 있을 경우 손 쉽게 로그인이 가능하다.

07번 항목에서 설명했다시피 **AWS**의 **root** 계정은 모든 권한을 다 가지고 있기 때문에 최대한 비인가 접근을 막기 위해,

Access Key를 만들거나 사용하지 않는것이 보안적으로 안전하다.

또한 **Access Key**가 탈취당하거나 유출될 수 있으므로 **60**일 간격으로 **Access Key**를 재발급 해주어야 한다.

딴소리

요즘엔 **AWS** 콘솔 자체적으로 보안적인 조언을 많이 띄워줘서 생각보다 취약한 항목이 많지는 않다.

보안 권장 사항 1

⚠ **루트 사용자에 대해 MFA 추가**

루트 사용자에 대해 MFA 추가 - 루트 사용자에 대해 다중 인증(MFA)을 활성화하여 이 계정의 보안을 강화합니다.

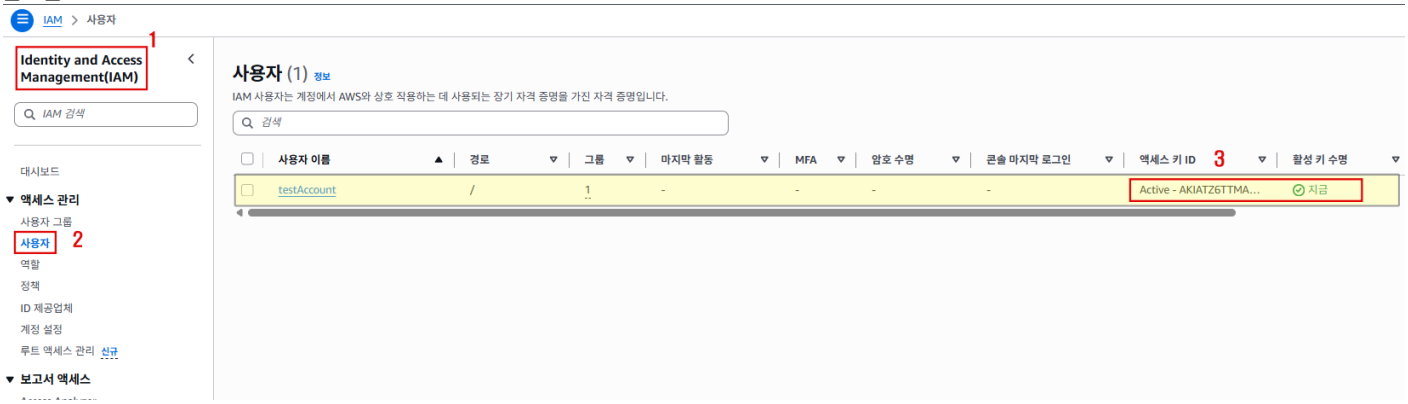
✓ **루트 사용자에게 활성 액세스 키가 없음**

루트 사용자 대신 IAM 사용자에 연결된 액세스 키를 사용하면 보안이 향상됩니다.

진단 방법

01. IAM 에 들어가보면 사용자 별로 활성 키 수명이라는 항목이 존재한다. (활성키들의 수명 확인)

*방금 테스트용으로 생성하여 "지금"이라고 뜨지만 보통 얼마나 사용했는지가 나온다. ex. 3일, 240일 등



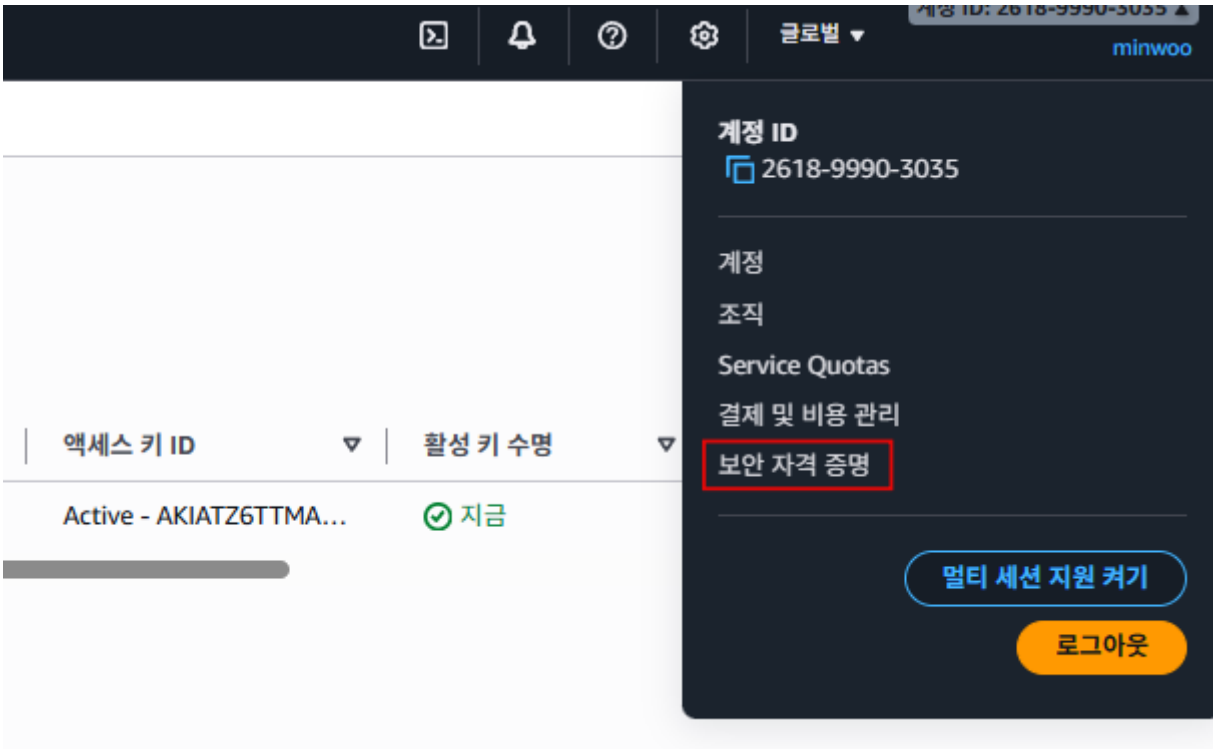
사용자 (1) 정보

IAM 사용자는 계정에서 AWS와 상호 작용하는 데 사용되는 장기 자격 증명을 가진 자격 증명입니다.

☐	사용자 이름	경로	그룹	마지막 활동	MFA	암호 수명	콘솔 마지막 로그인	액세스 키 ID	활성 키 수명
☐	testAccount	/	1	-	-	-	-	Active - AKIATZ6TTMA...	✓ 지금

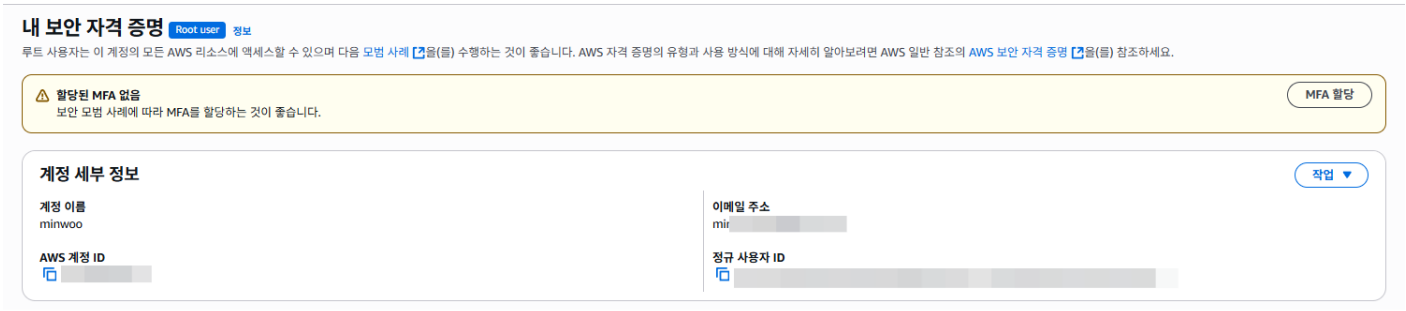
root 계정은 IAM 사용자 목록에 나오지 않는다.

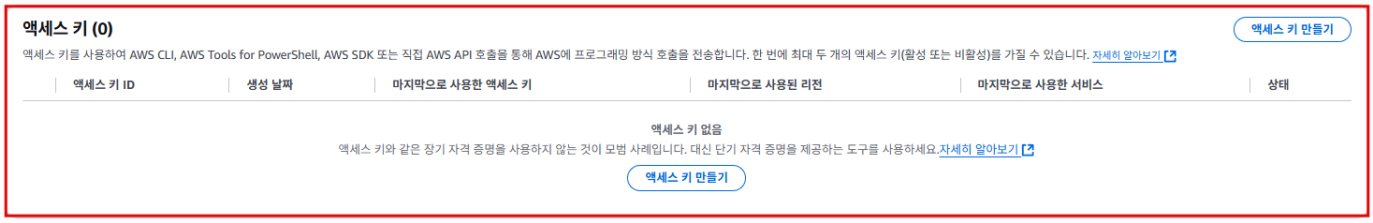
root 계정에 Access Key를 생성하기 위해선 일단 root 계정으로 진입 후 우측 상단의 계정명(minwoo) 클릭 후 "보안 자격 증명"에 진입



진입해보면 액세스 키가 부여되어 있는지 여부를 확인할 수 있다.

* 인터뷰로 root 계정에 액세스 키를 부여하여 사용하는지 확인 후, 담당자분이 모른다고 하면 이 경로로 확인해보면 될 듯





내 보안 자격 증명 Root user 정보

루트 사용자는 이 계정의 모든 AWS 리소스에 액세스할 수 있으며 다음 모범 사례을(를) 수행하는 것이 좋습니다. AWS 자격 증명의 유형과 사용 방식에 대해 자세히 알아보려면 AWS 일반 참조의 AWS 보안 자격 증명을(를) 참조하세요.

⚠ **할당된 MFA 없음**
보안 모범 사례에 따라 MFA를 할당하는 것이 좋습니다.

계정 세부 정보

계정 이름
minwoo

AWS 계정 ID
[redacted]

이메일 주소
mi[redacted]

정규 사용자 ID
[redacted]

액세스 키 (0)

액세스 키를 사용하여 AWS CLI, AWS Tools for PowerShell, AWS SDK 또는 직접 AWS API 호출을 통해 AWS에 프로그래밍 방식 호출을 전송합니다. 한 번에 최대 두 개의 액세스 키(활성 또는 비활성)를 가질 수 있습니다. 자세히 알아보기

액세스 키 ID	생성 날짜	마지막으로 사용한 액세스 키	마지막으로 사용된 리전	마지막으로 사용한 서비스	상태

액세스 키 없음
액세스 키와 같은 장기 자격 증명을 사용하지 않는 것이 모범 사례입니다. 대신 단기 자격 증명을 제공하는 도구를 사용하세요. 자세히 알아보기

양호 취약 기준

진단 기준	**양호기준** : AWS Admin Console 계정에 Access Key가 존재하지 않고 IAM 사용자 계정에 대한 Access Key 사용 주기가 60일 이내일 경우 **취약기준** : AWS Admin Console 계정에 Access Key가 존재하거나 IAM 사용자 계정에 대한 Access Key 사용 주기가 60일 초과일 경우

양호 : **root** 계정 접근 시 **Access Key**를 사용하지 않고 생성된 **Access Key**들의 수명이 **60**일을 넘지 않을 경우

취약 : **root** 계정에 **Access Key**를 통해 접근하거나 **Access Key**들의 수명이 **60**일을 넘을 경우

참고

1.9 MFA (Multi-Factor Authentication) 설정

취약점 설명

MFA, 흔히 말하는 **2**차인증, **OTP,** 다중 인증 과 같다. 사용자의 아이디, 패스워드 뿐만 아니라 다른 방법으로 추가 인증을 진행하는 것

요즘은 다른 곳에서 유출된 계정 정보를 가지고 다른 서비스에도 대입해보는 크리덴셜 스터핑 공격이 성행하기 때문에

사용자의 추가 인증이 거의 필수가 되가는 듯 하다.

딴소리

MFA인증 방식은 아래와 같다.

1단계
● MFA 디바이스 선택
2단계
○ 디바이스 설정

MFA 디바이스 선택 Info

MFA device name

디바이스 이름
이 이름은 이 디바이스의 식별 ARN에 사용됩니다.

디바이스 이름

최대 64자까지 입력할 수 있습니다. 유효한 문자: A~Z, a~z, 0~9, 및 + = , . @ _ -(하이픈)

MFA device

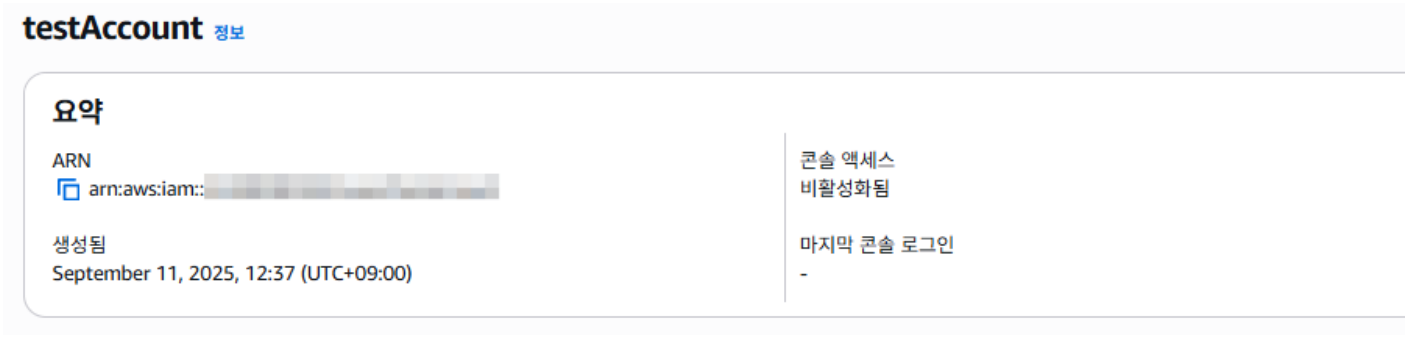
디바이스 옵션
사용자 이름과 암호 외에도 이 디바이스를 사용하여 계정에 인증합니다.

● **패스키 또는 보안 키**
지문, 얼굴 또는 화면 잠금을 사용하여 인증합니다. 이 디바이스에서 패스키를 생성하거나 FIDO2 보안 키와 같은 다른 디바이스를 사용하세요.

○ **인증 관리자 앱**
모바일 디바이스 또는 컴퓨터에 설치된 앱에서 생성된 코드를 사용하여 인증합니다.

○ **하드웨어 TOTP 토큰**
하드웨어 TOTP 토큰 또는 기타 하드웨어 디바이스에서 생성된 코드를 사용하여 인증합니다.

테스트를 해보다 보니 궁금점 콘솔 액세스가 비활성화 된 계정에 **MFA**를 설정하는게 의미가 있을까?



testAccount

요약

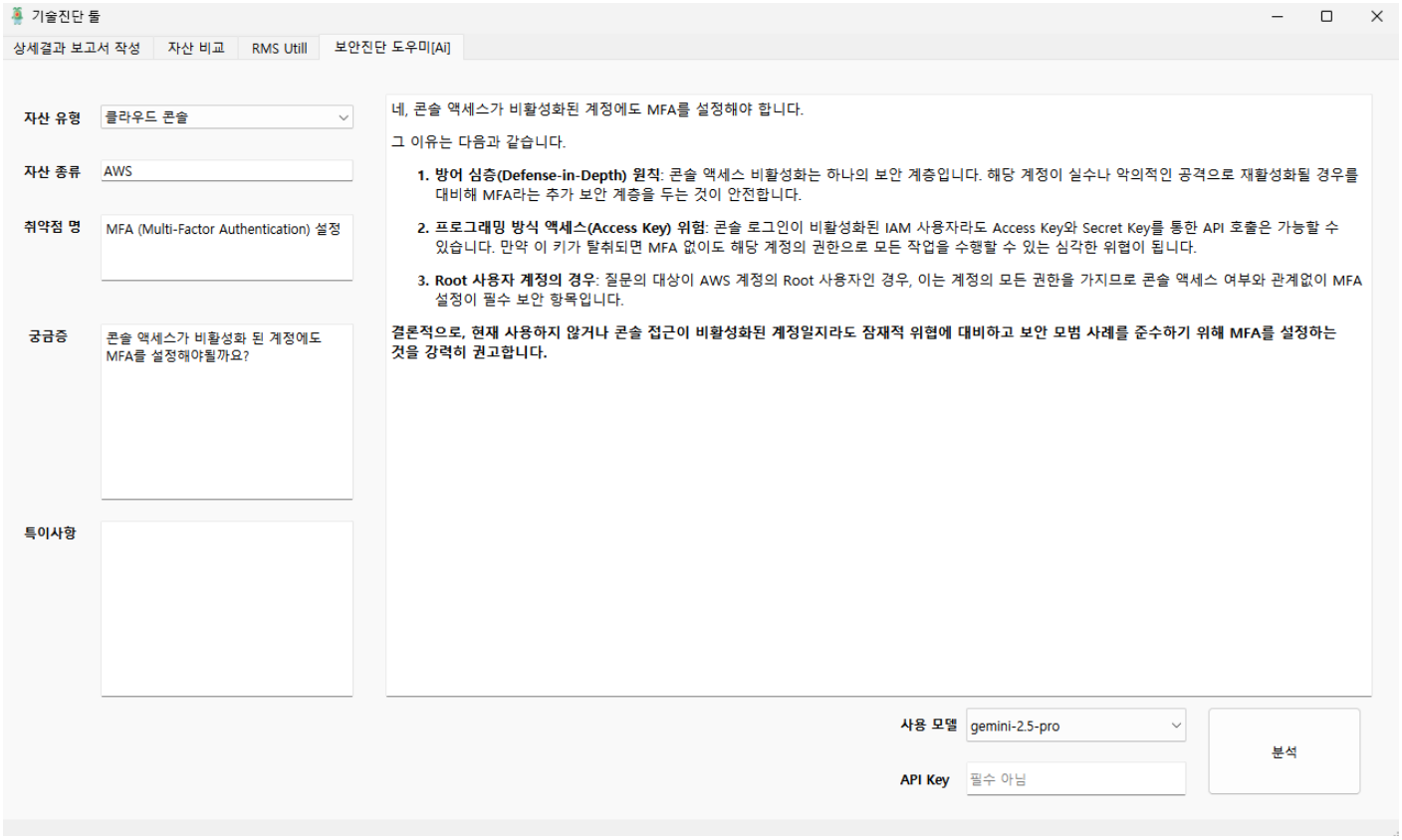
ARN
arn:aws:iam::[redacted]

생성됨
September 11, 2025, 12:37 (UTC+09:00)

콘솔 액세스
비활성화됨

마지막 콘솔 로그인
-

개인적으로 만들어서 쓰고 있는 보안 컨설턴트 페르소나를 가지고 있는 **ai api**에 물어보니 그렇다고 한다 .



자산 유형 클라우드 콘솔

자산 종류 AWS

취약점 명 MFA (Multi-Factor Authentication) 설정

궁금증 콘솔 액세스가 비활성화 된 계정에도 MFA를 설정해야될까요?

특이사항

네, 콘솔 액세스가 비활성화된 계정에도 MFA를 설정해야 합니다.

그 이유는 다음과 같습니다.

1. **방어 심층(Defense-in-Depth) 원칙**: 콘솔 액세스 비활성화는 하나의 보안 계층입니다. 해당 계정이 실수나 악의적인 공격으로 재활성화될 경우를 대비해 MFA라는 추가 보안 계층을 두는 것이 안전합니다.
2. **프로그래밍 방식 액세스(Access Key) 위험**: 콘솔 로그인이 비활성화된 IAM 사용자라도 Access Key와 Secret Key를 통한 API 호출은 가능할 수 있습니다. 만약 이 키가 탈취되면 MFA 없이도 해당 계정의 권한으로 모든 작업을 수행할 수 있는 심각한 위협이 됩니다.
3. **Root 사용자 계정의 경우**: 질문의 대상이 AWS 계정의 Root 사용자인 경우, 이는 계정의 모든 권한을 가지므로 콘솔 액세스 여부와 관계없이 MFA 설정이 필수 보안 항목입니다.

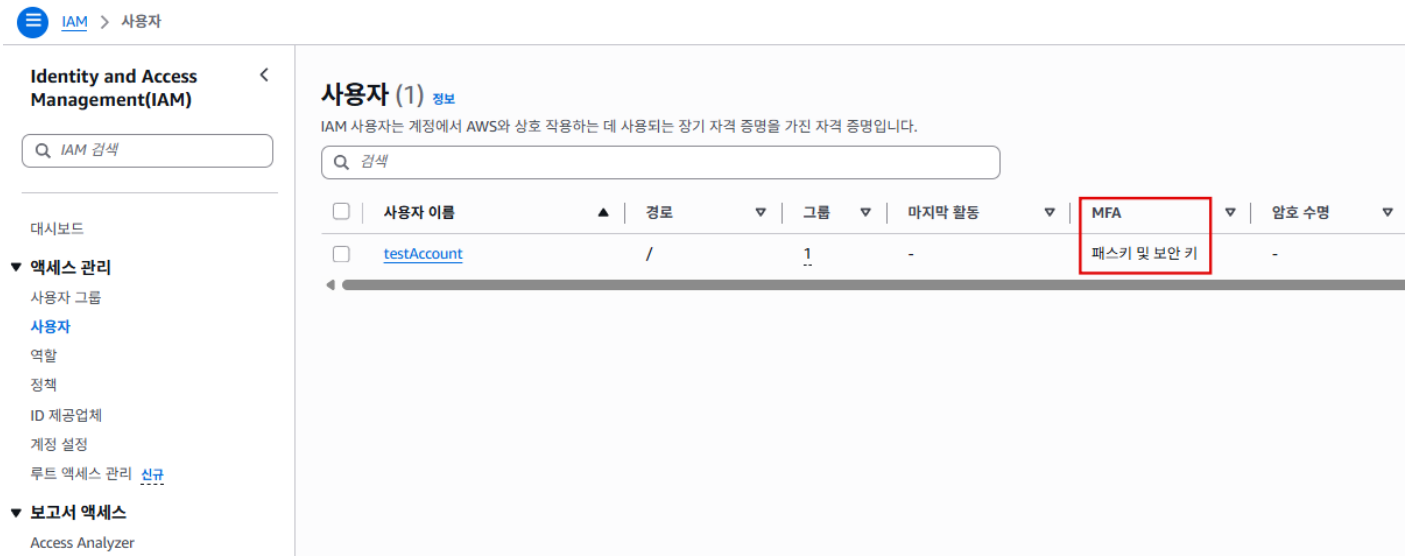
결론적으로, 현재 사용하지 않거나 콘솔 접근이 비활성화된 계정일지라도 잠재적 위협에 대비하고 보안 모범 사례를 준수하기 위해 MFA를 설정하는 것을 강력히 권고합니다.

사용 모델 gemini-2.5-pro

API Key 필수 아님

진단 방법

01. [IAM] -> [사용자 **] ->** 사용자 목록에 **MFA**가 활성화 되어 있는지 확인



사용자 (1)

IAM 사용자는 계정에서 AWS와 상호 작용하는 데 사용되는 장기 자격 증명을 가진 자격 증명입니다.

☐	사용자 이름	경로	그룹	마지막 활동	MFA	암호 수명
☐	testAccount	/	1	-	패스키 및 보안 키	-

양호 취약 기준

진단 기준	**양호기준** : AWS 계정 및 IAM 사용자 계정 로그인 시 MFA가 활성화 되어 있을 경우 **취약기준** : AWS 계정 및 IAM 사용자 계정 로그인 시 MFA 가 비활성화 되어 있을 경우
비고	MFA 인증을 사용하지 않고 SSO 인증을 통해서 로그인할 경우 양호로 처리될 수 있음

양호 : 사용자 계정 로그인 시 **MFA**를 사용하고 있을 시

취약 : **MFA**가 활성화 되어 있지 않은 계정이 존재할 경우

참고

1.10 AWS 계정 패스워드 정책 관리

취약점 설명

패스워드를 생성할 때는 복잡도를 충족하도록 설정해야 한다.

일반적으로 **3**종류 이상의 조합 **(**대 소문자 **,** 숫자 특수문자 **)** 및 **8**자리 이상의 패스워드를 기준으로 한다 **.**

또는 **2**종류 이상의 조합 **+ 10**자리 이상의 패스워드도 가능 **.**

AWS에서는 계정의 패스워드를 설정할때 복잡도를 충족하도록 강제하는 정책이 기본적으로 적용되는데 **,**

해당 설정을 개인 설정으로 변경하여 취약하게 사용할 경우 보안적으로 위험할 수 있다 **.**

딴소리

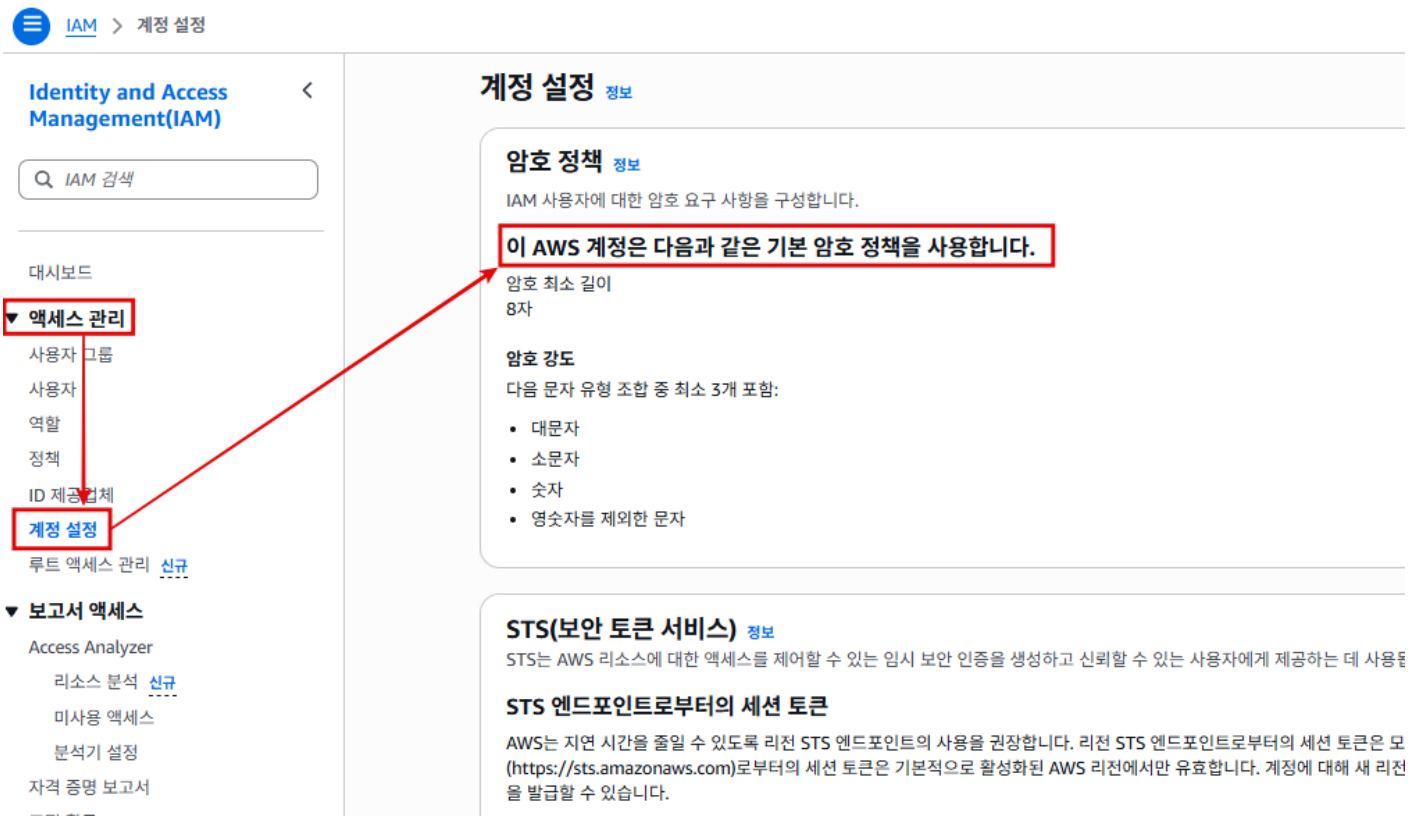
SK AWS 진단 가이드에선 패스워드 만료 및 이전 패스워드 사용 제한을 설정하도록 권고하고 있는데 **,**

해당 사항은 **AWS** 기본 정책에 적용되어 있지 않기 때문에 계정 설정에서 개인 설정을 통해 해당 정책을 설정해주어야 한다 **.**

진단 방법

01. [IAM] -> [액세스 관리 **] -> [**계정 설정 **] ->** 암호 정책을 올바르게 설정하여 사용하고 있는지 확인

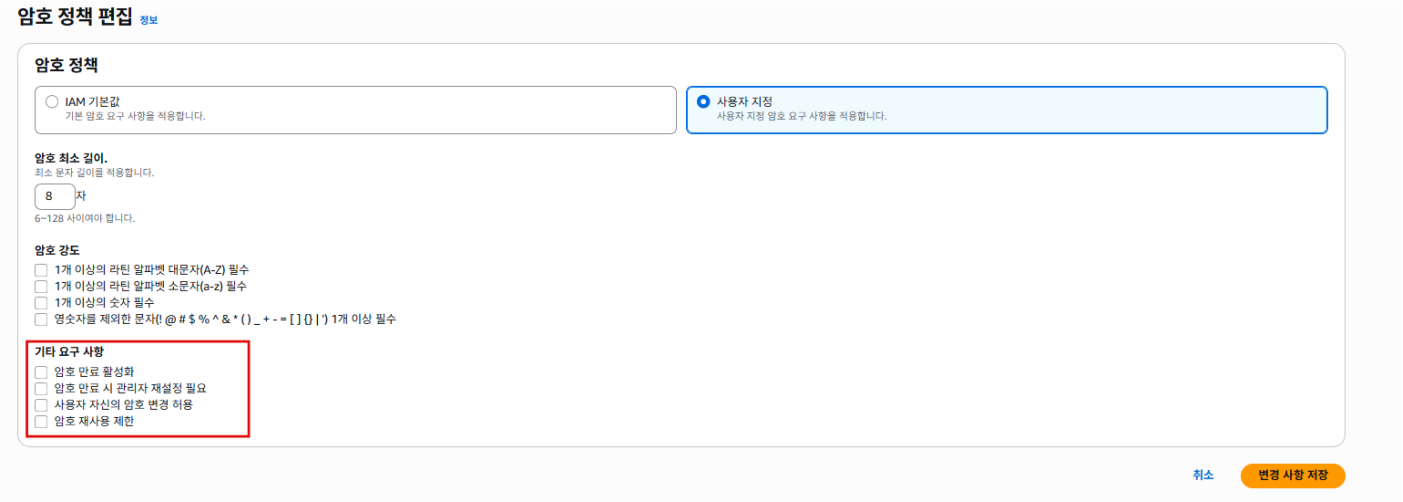
*** AWS** 기본암호 정책은 패스워드 만료 **,** 이전패스워드 사용 금지 정책이 존재하지 않으므로 취약


IAM > 계정 설정
Identity and Access Management(IAM)
IAM 검색
대시보드
액세스 관리
사용자 그룹
사용자
역할
정책
ID 제공업체
계정 설정
루트 액세스 관리 신규
보고서 액세스
Access Analyzer
리소스 분석 신규
미사용 액세스
분석기 설정
자격 증명 보고서
계정 설정 정보
암호 정책 정보
IAM 사용자에 대한 암호 요구 사항을 구성합니다.
이 AWS 계정은 다음과 같은 기본 암호 정책을 사용합니다.
암호 최소 길이
8자
암호 강도
다음 문자 유형 조합 중 최소 3개 포함:
대문자
소문자
숫자
영숫자를 제외한 문자
STS(보안 토큰 서비스) 정보
STS는 AWS 리소스에 대한 액세스를 제어할 수 있는 임시 보안 인증을 생성하고 신뢰할 수 있는 사용자에게 제공하는 데 사용
STS 엔드포인트로부터의 세션 토큰
AWS는 지연 시간을 줄일 수 있도록 리전 STS 엔드포인트의 사용을 권장합니다. 리전 STS 엔드포인트로부터의 세션 토큰은 모
(https://sts.amazonaws.com)로부터의 세션 토큰은 기본적으로 활성화된 AWS 리전에서만 유효합니다. 계정에 대해 새 리전
을 발급할 수 있습니다.

02. □□ □□ □□□ □□ □□ □□□ □□


계정 설정 정보
암호 정책 정보
편집
IAM 사용자에 대한 암호 요구 사항을 구성합니다.
이 AWS 계정은 다음과 같은 기본 암호 정책을 사용합니다.
암호 최소 길이
8자
암호 강도
다음 문자 유형 조합 중 최소 3개 포함:
대문자
소문자
숫자
영숫자를 제외한 문자
기타 요구 사항
비밀번호가 만료되지 않음
AWS 계정 이름 또는 이메일 주소와 동일할 수 없음
STS(보안 토큰 서비스) 정보
STS는 AWS 리소스에 대한 액세스를 제어할 수 있는 임시 보안 인증을 생성하고 신뢰할
STS 엔드포인트로부터의 세션 토큰
AWS는 지연 시간을 줄일 수 있도록 리전 STS 엔드포인트의 사용을 권장합니다. 리전 STS 엔드포인트를 사용할 경우 봊포인트
(https://sts.amazonaws.com)로부터의 세션 토큰은 기본적으로 활성화된 AWS 리전에서 활성화하려는 경우 리전 STS 엔드포인트로부터의 세션 토큰을 사용하AWS에서 유효한 세션 토큰
을 발급할 수 있습니다.

03. □□□ □□□□ □□□□ □□ □□□□□ □□□□ □□ □□ □□□ □□□□ □□□ □ □□ .



□□ □□ □□

진단 기준	**양호기준** : Admin Console 및 IAM 계정의 패스워드 복잡성 기준 준수 및 암호 만료/재사용 제한을 설정하고 있을 경우 **취약기준** : Admin Console 및 IAM 계정의 패스워드 복잡성 기준 준수 및 암호 만료/재사용 제한을 설정하고 있지 않을 경우
비고	

□□

1.11 EKS 사용자 관리

취약점 설명

EKS는 **AWS**에서 제공하는 쿠버네티스 서비스이다.

쿠버네티스는 컨테이너를 종합적으로 관리하는 서비스라고 생각하면 되며,

EKS에서 마스터노드를 자체적으로 관리, 업데이트한다.

현 취약점은 역할 기반 접근 제어 정책 **(RBAC)**을 통해 사용자들의 역할이 적절히 지정 되어있는지를 확인한다.

※ aws-auth ConfigMap 변경 시에는 잘못된 형식의 aws-auth 컨피그맵으로 인해 Cluster에 대한 접근 권한을 잃을 수 있어 ConfigMap을 변경해야 하는 경우 도구(eksctl, keikoproj의 aws-auth, AWS IAM Authenticator CLI)를 사용해야합니다.

딴소리

요즘 **AWS**가 굉장히 상용화되어 있다보니 점점 진단 빈도수가 늘고있는데, **EKS**부분은 보안조치가 취해지지 않은 경우가 좀 있다.

그런 설정이 있는지도 모르는 분들도 있고, 보통 **EKS Shell**에 접근하는 사용자는 한명으로 두기 때문에 그럴경우 **RBAC** 정책이 의미가 없어 양호로

진단한다.

진단 방법

현 취약점 항목에서는 **EKS** 상에 **RBAC** 정책을 적용하였는지를 확인하며,

AWS Console 상에서 설정이 불가능한 설정이므로 **kubectl** 명령어를 통하여 확인한다.

- ConfigMap 확인 명령어 (사용자 목록 및 사용자에 부여된 클러스터 롤 확인)

```
kubectl get configmap aws-auth -n kube-system -o yaml
```

- ClusterRole 확인 명령어 (생성되어 있는 클러스터롤 확인)

```
kubectl get clusterrole
```

```
rasecureJDH:~/environment $ kubectl get clusterrole
NAME                                                                    CREATED AT
admin                                                                   2024-02-07T03:23:41Z
aws-node                                                                2024-02-07T03:25:03Z
cluster-admin                                                           2024-02-07T03:23:41Z
cluster-pod-reader                                                      2024-02-14T05:09:53Z
edit                                                                    2024-02-07T03:23:41Z
eks:addon-manager                                                       2024-02-07T03:23:50Z
eks:az-poller                                                           2024-02-07T03:23:47Z
eks:certificate-controller-approver                                     2024-02-07T03:23:47Z
```

양호 취약 기준

EKS의 리소스에 접근 가능한 계정이 하나뿐이거나, 사용자들의 권한이 역할에 맞게 부여되어 있을 경우 - 양호

EKS 리소스에 접근 가능한 계정들에 대한 역할 및 권한이 관리되지 않고 과도한 권한을 부여하였을 경우 - 취약

참고

EKS는 **AWS**의 유료계정을 사용해야 해서 관련 캡쳐는 **SK** 쉴더스의 가이드를 가져왔다.

1.12 EKS 서비스 어카운트 관리

취약점 설명

서비스 어카운트 **=** 파드에 **K8S RBAC** 역할을 할당할 수 있는 특수한 유형의 개체

Cluster 내의 각 네임스페이스에 기본 서비스 어카운트가 자동으로 생성됨

쉽게 설명하면

01. "서비스 어카운트 **(SA)"** 를 **Pod** 에 부여하여 **"RBAC(**역할 기반 접근 제어 **)"** 를 수행한다 **.**

02. 서비스어카운트에는 **RBAC** 정책과 **,** 인증에 사용할 **Token**이 존재한다 **.**

03. 별도의 서비스어카운트를 지정하지 않을경우 **default** 서비스 어카운트가 할당되며 **, Token**은 특정 경로에 자동으로 마운트된다 **.**

04. 토큰은 **K8S API**를 호출할때 인증에 쓰이며 **,** 해당 어플리케이션이 **K8S API**를 호출할 일이 없을경우 **Token**을 쓸 일이 없다 **.**

05. "automountServiceAccountToken" 속성을 **false**로 설정할 경우 불필요하게 **SA** 토큰을 자동으로 마운트하는 것을 방지할 수 있다 **.**

딴소리

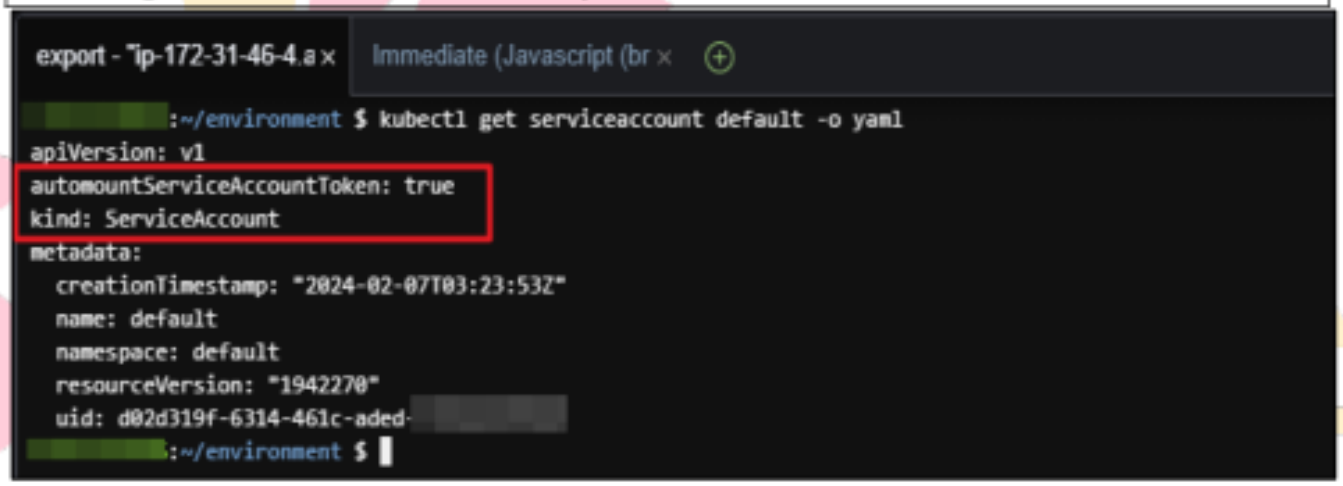
K8S 구조를 파악하는게 중요하다 **.**

서비스 **,** 파드 **,** 클러스터 **,** 시크릿 **,** 인그레스 등 **k8s**에서 쓰이는 단어들의 쓰임을 확실히 알고 어떤 식으로 흘러가는지 공부해보면 감이 온다 **.**

진단 방법

01. 사용자가 **SA**를 지정하지 않고 **default SA**를 쓸 경우에 **automount**를 방지하는 것이기 때문에 **default SA**의 설정 값을 확인한다 **.**

```
kubectl get serviceaccount default -o yaml
```



가. 서비스 어카운트 토큰 자동 마운트 비활성화

1) 서비스 어카운트 토큰 자동 마운트 비활성화 여부 확인

```
kubectl get serviceaccount default -o yaml
```

```
[redacted]:~/environment $ kubectl get serviceaccount default -o yaml
apiVersion: v1
automountServiceAccountToken: true
kind: ServiceAccount
metadata:
  creationTimestamp: "2024-02-07T03:23:53Z"
  name: default
  namespace: default
  resourceVersion: "1942270"
  uid: d02d319f-6314-461c-aded-[redacted]
[redacted]:~/environment $ ▌
```

* **"automountServiceAccountToken"** 옵션이 **true**로 활성화되어있는것을 확인 **(**취약**)**

양호 취약 기준

automountServiceAccountToken 옵션이 **false**로 비활성화 되어 있을 경우 **->** 양호

automountServiceAccountToken 옵션이 **true**로 활성화되어 있을 경우 **->** 취약

automountServiceAccountToken 옵션이 존재하지 않을 경우 **(**기본 값 **:** 활성화 **) ->** 취약

참고