
AWS Identity and Access Management

사용 설명서



AWS Identity and Access Management: 사용 설명서

Copyright © 2019 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

IAM이란?	1
IAM 소개 동영상	1
IAM 기능	1
IAM에 액세스	2
IAM 작동 방식 이해	3
약관	4
보안 주체	4
요청	5
인증	5
승인	5
작업 또는 연산	6
리소스	6
개요: 사용자	6
첫 액세스에만 해당: 루트 사용자 자격 증명	6
IAM 사용자	7
기존 사용자 연동	7
개요: 권한 및 정책	8
정책 및 계정	8
정책 및 사용자	8
정책 및 그룹	9
연동 사용자 및 역할	10
자격 증명 기반 정책 및 리소스 기반 정책	10
IAM 외부의 보안 기능	11
공통 작업의 빠른 링크	12
설정	14
IAM을 사용하여 AWS 리소스에 대한 사용자 액세스를 허용하는 방법	14
IAM을 사용하려면 가입해야 합니까?	15
추가 리소스	15
시작	16
IAM 관리자 및 그룹 만들기	17
관리자 IAM 사용자 및 그룹 생성(콘솔)	17
IAM 사용자 및 그룹 생성(AWS CLI)	18
관련 리소스	20
위임 사용자 생성	20
위임 IAM 사용자 및 그룹 생성(콘솔)	17
그룹 권한 줄이기	21
사용자의 계정 로그인 방법	22
콘솔 활동에 필요한 권한	23
CloudTrail에 로그인 세부 정보 기록	23
자습서	25
자습서: Billing 콘솔에 대한 액세스 권한 위임	25
사전 조건	25
1단계: AWS 테스트 계정에서 결제 데이터에 대한 액세스 권한 활성화	26
2단계: 결제 데이터에 대한 권한을 부여하는 IAM 정책 생성	26
3단계: 그룹에 결제 정책 연결	27
4단계: Billing 콘솔에 대한 사용자 액세스 권한 테스트	27
관련 리소스	28
요약	29
자습서: 역할을 사용한 AWS 계정 간 액세스 권한 위임	29
사전 조건	30
1단계 - 역할 만들기	30
2단계 - 역할에 액세스 권한 부여	32
3단계 - 역할 전환을 통해 액세스 권한 테스트	34
관련 리소스	37

요약	37
자습서: 고객 관리형 정책 만들기	37
사전 조건	37
1단계: 정책 만들기	38
2단계: 정책 연결	38
3단계: 사용자 액세스 테스트	39
관련 리소스	39
요약	39
자습서: 사용자들이 자신의 자격 증명 및 MFA 설정을 구성할 수 있도록 하기	39
사전 조건	40
1단계: MFA 로그인을 강제할 정책 생성	40
2단계: 테스트 그룹에 정책 연결하기	41
3단계: 사용자 액세스 테스트	41
관련 리소스	42
모범 사례 및 사용 사례	43
모범 사례	43
AWS 계정 루트 사용자 액세스 키 잠금	43
개별 IAM 사용자 만들기	44
그룹을 사용하여 IAM 사용자에게 권한을 할당합니다.	44
최소 권한 부여	44
AWS 관리형 정책으로 권한 사용 시작	45
인라인 정책 대신 고객 관리형 정책 사용	45
액세스 레벨을 이용한 IAM 권한 검토	46
사용자에 대한 강력한 암호 정책 구성	47
권한 있는 사용자에 대해 MFA 활성화	47
Amazon EC2 인스턴스에서 실행되는 애플리케이션에 역할 사용	47
역할을 사용하여 권한 위임	47
액세스 키를 공유하지 마십시오	48
자격 증명을 정기적으로 교체	48
불필요한 자격 증명 삭제	48
보안 강화를 위해 정책 조건 사용	48
AWS 계정의 활동 모니터링	49
IAM 모범 사례에 대한 동영상 프레젠테이션	49
기업 사용 사례	50
Example Corp의 초기 설정	50
Amazon EC2의 IAM 사용 사례	50
Amazon S3의 IAM 사용 사례	51
IAM 콘솔 및 로그인 페이지	53
IAM 사용자 로그인 페이지	53
AWS 계정 루트 사용자 로그인 페이지	54
AWS Management 콘솔에 대한 사용자 액세스 제어	54
AWS 계정 ID 및 별칭	55
AWS 계정 ID 찾기	55
계정 별칭 정보	56
AWS 계정 별칭 만들기, 삭제 및 나열	56
IAM 로그인 페이지에 MFA 디바이스 사용	57
가상 MFA 디바이스로 로그인	57
U2F 보안 키로 로그인	58
하드웨어 MFA 디바이스로 로그인	58
IAM 콘솔 검색	58
IAM 콘솔 검색 사용	59
IAM 콘솔 검색 결과 내 아이콘	59
샘플 검색 문구	59
ID	61
AWS 계정 루트 사용자	61
IAM 사용자	61
IAM 그룹	61

IAM 역할	61
임시 자격 증명	62
IAM 사용자를 만들어야 하는 경우(역할이 아님)	62
IAM 역할을 만들어야 하는 경우(사용자가 아님)	62
사용자	63
AWS가 IAM 사용자를 식별하는 방법	63
사용자 및 자격 증명	63
사용자 및 권한	64
사용자 및 계정	65
서비스 계정인 사용자	65
사용자 추가	65
IAM 사용자가 AWS에 로그인하는 방법	69
사용자 관리	70
사용자의 권한 변경	73
암호	78
액세스 키	88
분실한 암호나 액세스 키 복구	95
멀티 팩터 인증(MFA)	96
미사용 자격 증명 찾기	133
자격 증명 보고서 가져오기	135
IAM과 CodeCommit를 함께 사용: Git 자격 증명, SSH 키 및 AWS 액세스 키	140
서버 인증서 작업	141
그룹	145
그룹 생성	147
그룹 관리	148
역할	153
용어 및 개념	153
일반적인 시나리오	156
자격 증명 공급자 및 연동	161
서비스 연결 역할	195
역할 생성	202
역할 사용	227
역할 관리	246
역할 VS 리소스 기반 정책	257
엔터티 태그 지정	259
AWS 태그 이름 지정 규칙 선택	259
IAM 엔터티 태그 지정 규칙	259
IAM 엔터티 태그 지정에 필요한 권한	260
IAM 엔터티에 대한 태그 관리(콘솔)	262
IAM 엔터티에 대한 태그 관리(AWS CLI 또는 AWS API)	262
임시 보안 자격 증명	263
AWS STS 및 AWS 리전	263
임시 자격 증명과 관련된 일반적인 시나리오	264
임시 보안 자격 증명 요청하기	265
임시 보안 자격 증명을 사용해 AWS 리소스에 대한 액세스 요청하기	274
사용자 임시 보안 자격 증명에 대한 권한 제어	277
AWS 리전에서 AWS STS 활성화 및 비활성화	287
AWS STS 인터페이스 VPC 엔드포인트 사용	289
임시 자격 증명을 사용하는 샘플 애플리케이션	290
임시 자격 증명에 관한 추가 리소스	290
루트 사용자	291
AWS 계정 루트 사용자의 MFA 활성화	291
루트 사용자를 위한 액세스 키 생성	292
루트 사용자로부터 액세스 키 삭제하기	292
루트 사용자의 암호 변경	293
CloudTrail을 사용하여 이벤트 로깅	293
CloudTrail의 IAM 및 AWS STS 정보	294

CloudTrail 파일에 로깅된 이벤트 예제	296
CloudTrail의 중복 로그 항목 방지	302
액세스 관리	304
액세스 관리 리소스	305
정책 및 권한	305
정책 유형	305
정책 및 루트 사용자	309
JSON 정책 개요	309
관리형 정책과 인라인 정책	312
권한 경계	317
자격 증명과 리소스 비교	326
정책을 사용하여 액세스 제어	327
IAM 태그를 사용한 액세스 제어	336
태그를 사용한 액세스 제어	338
정책 예제	341
IAM 정책 관리	377
IAM 정책 만들기	377
JSON 정책 검증	382
IAM 정책 테스트	383
자격 증명 권한 추가 또는 제거	391
IAM 정책 버전 관리	399
IAM 정책 편집	402
IAM 정책 삭제	406
액세스 데이터를 사용하여 권한 줄이기	409
정책 이해	419
정책 요약(서비스 목록)	419
서비스 요약(작업 목록)	429
작업 요약(리소스 목록)	433
정책 요약 예제	435
필요한 권한	443
IAM 자격 증명을 관리하기 위한 권한	443
AWS Management 콘솔에서의 작업 권한	445
전 AWS 계정에 권한 부여	445
한 서비스에서 다른 서비스에 액세스할 권한	445
필수 작업	446
IAM 정책의 예	446
IAM 문제 해결	451
일반적인 문제 해결	451
액세스 키를 분실했습니다	451
예전 계정에 액세스해야 합니다	451
내 계정에 로그인할 수 없음	452
AWS 서비스에 요청하면 "액세스 거부"가 발생합니다	452
임시 보안 자격 증명으로 요청하면 "액세스 거부"가 발생합니다	453
정책 변수가 작동하지 않습니다	453
변경 사항이 매번 즉시 표시되는 것은 아닙니다	454
문제 해결 정책	454
시각적 편집기를 사용하여 문제 해결	455
정책 요약을 사용하여 문제 해결	458
정책 관리 문제 해결	464
JSON 정책 문서 문제 해결	464
U2F 보안 키 문제 해결	468
U2F 보안 키를 활성화할 수 없습니다	468
U2F 보안 키를 사용해 로그인할 수 없습니다	469
U2F 키를 분실했거나 고장 났습니다	469
기타 문제	469
IAM 역할 문제 해결	469
역할을 위임할 수 없음	469

내 AWS 계정에 표시되는 새 역할	470
AWS 계정에서 역할을 편집하거나 삭제할 수 없음	471
iam:PassRole를 수행하도록 인증되지 않음	471
12시간 길이 세션을 선택한 경우 역할을 위임할 수 없는 이유(AWS CLI, AWS API)	471
Amazon EC2 및 IAM 문제 해결	472
인스턴스를 시작하려고 할 때 Amazon EC2 콘솔 IAM 역할 목록에서 보여야 할 역할이 보이지 않습니다.	472
제 인스턴스에 있는 자격 증명의 역할이 잘못되었습니다.	472
AddRoleToInstanceProfile을 호출하려고 하면 AccessDenied 오류가 발생합니다.	472
Amazon EC2: 역할로 인스턴스를 시작하려고 하면 AccessDenied 오류가 발생합니다.	473
제 EC2 인스턴스의 임시 보안 자격 증명에 액세스할 수 없습니다.	473
IAM 하위 트리에서 info 문서의 오류란 무엇인가요?	474
Amazon S3 및 IAM 문제 해결	474
Amazon S3 버킷에 대한 익명 액세스 권한을 부여하는 방법은 무엇입니까?	474
AWS 계정 루트 사용자로 로그인했는데 내 계정으로 Amazon S3 버킷에 액세스할 수 없는 이유가 무엇입니까?	475
AWS로 SAML 2.0 연동 문제 해결	475
잘못된 SAML 응답	475
RoleSessionName은 필수입니다.	476
AssumeRoleWithSAML에 대한 권한이 없음	476
잘못된 RoleSessionName 문자	476
잘못된 응답 서명	476
역할을 위임하지 못함	477
메타데이터를 구문 분석할 수 없음	477
DurationSeconds가 MaxSessionDuration 초과	477
문제 해결을 위해 브라우저에서 SAML 응답을 보는 방법	477
참조	480
IAM 식별자	480
표시 이름 및 경로	480
IAM ARN	480
고유 ID	483
제한 사항	485
IAM 엔터티 이름 제한	485
IAM 엔터티 객체 제한	485
IAM 엔터티 문자 제한	487
IAM으로 작업하는 서비스	488
컴퓨팅	489
스토리지	490
데이터베이스	490
개발자 도구	491
보안, 자격 증명 및 규정 준수	491
기계 학습	492
관리 도구	492
마이그레이션 및 전송	493
모바일	493
네트워킹 및 콘텐츠 전송	494
미디어	494
데스크톱 및 앱 스트리밍	495
분석	495
애플리케이션 통합	495
비즈니스 애플리케이션	496
사물 인터넷	496
로봇 공학	496
게임 개발	497
증강현실 및 가상현실	497
고객 참여	497
최종 사용자 컴퓨팅	497

추가 리소스	497
정책 참조	498
JSON 요소 참조	498
정책 평가 로직	531
정책 문법	538
직무 기능에 대한 AWS 관리형 정책	543
전역 조건 키	551
IAM 조건 키	558
리소스	566
사용자 및 그룹	566
자격 증명(암호, 액세스 키 및 MFA 디바이스)	566
권한 및 정책	566
연동 및 위임	567
IAM 및 기타 AWS 제품	567
Using IAM with Amazon EC2	567
Using IAM with Amazon S3	567
Using IAM with Amazon RDS	568
Using IAM with Amazon DynamoDB	568
일반 보안 사례	568
일반 리소스	568
쿼리 요청 실행	570
엔드포인트	570
HTTPS 필요	570
IAM API 요청에 서명	571
문서 기록	572
AWS Glossary	574

IAM이란?

 Follow us on Twitter

AWS Identity and Access Management(IAM)는 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있는 웹 서비스입니다. IAM을 사용하여 리소스를 사용하도록 인증(로그인) 및 권한 부여(권한 있음)된 대상을 제어합니다.

AWS 계정을 처음 생성할 때는 해당 계정의 모든 AWS 서비스와 리소스에 대해 완전한 액세스 권한이 있는 SSO(single sign-in) 자격 증명으로 시작합니다. 이 자격 증명은 AWS 계정 루트 사용자라고 하며, 계정을 생성할 때 사용한 이메일 주소와 암호로 로그인하여 액세스합니다. 관리 작업이라 할지라도 일상적인 작업에 루트 사용자를 사용하지 마십시오. 대신, [IAM 사용자를 처음 생성할 때만 루트 사용자를 사용하는 모범 사례](#)를 준수하십시오. 그런 다음 루트 사용자를 안전하게 보관해 두고 몇 가지 계정 및 서비스 관리 작업을 수행할 때만 자격 증명을 사용합니다.

주제

- [IAM 소개 동영상 \(p. 1\)](#)
- [IAM 기능 \(p. 1\)](#)
- [IAM에 액세스 \(p. 2\)](#)
- [IAM 작동 방식 이해 \(p. 3\)](#)
- [자격 증명 관리 개요: 사용자 \(p. 6\)](#)
- [액세스 관리 개요: 권한 및 정책 \(p. 8\)](#)
- [IAM 외부의 보안 기능 \(p. 11\)](#)
- [공통 작업의 빠른 링크 \(p. 12\)](#)

IAM 소개 동영상

AWS 교육 및 자격증 팀에서 IAM에 대한 10분 소개 동영상을 제공합니다.

[AWS Identity and Access Management 소개](#)

IAM 기능

IAM에서는 다음 기능을 제공합니다.

AWS 계정에 대한 공유 액세스

암호나 액세스 키를 공유하지 않고도 AWS 계정의 리소스를 관리하고 사용할 수 있는 권한을 다른 사람에게 부여할 수 있습니다.

세분화된 권한

리소스에 따라 여러 사람에게 다양한 권한을 부여할 수 있습니다. 예를 들어 일부 사용자에게는 Amazon Elastic Compute Cloud(Amazon EC2), Amazon Simple Storage Service(Amazon S3), Amazon DynamoDB, Amazon Redshift 및 기타 AWS 서비스에 대한 전체 액세스 권한을 허용하고 다른 사용자에게는 일부 S3 버킷에 대한 읽기 전용 권한, 일부 EC2 인스턴스를 관리할 수 있는 권한 또는 결제 정보에만 액세스할 수 있는 권한을 허용할 수 있습니다.

Amazon EC2에서 실행되는 애플리케이션을 위한 보안 AWS 리소스 액세스

EC2 인스턴스에서 실행되는 애플리케이션의 경우 IAM 기능을 사용하여 자격 증명을 안전하게 제공할 수 있습니다. 이러한 자격 증명은 애플리케이션에 다른 AWS 리소스에 액세스할 수 있는 권한을 제공합니다. 예를 들면 이러한 리소스에는 S3 버킷 및 DynamoDB 테이블이 있습니다.

멀티 팩터 인증(MFA)

보안 강화를 위해 계정과 개별 사용자에게 2팩터 인증을 추가할 수 있습니다. MFA를 사용할 경우 계정 소유자나 사용자가 계정 작업을 위해 암호나 액세스 키뿐 아니라 특별히 구성된 디바이스의 코드도 제공해야 합니다.

자격 증명 연동

기업 네트워크나 인터넷 자격 증명 공급자와 같은 다른 곳에 이미 암호가 있는 사용자에게 AWS 계정에 대한 임시 액세스 권한을 부여할 수 있습니다.

보장을 위한 자격 증명 정보

[AWS CloudTrail](#)을 사용하는 경우 계정의 리소스를 요청한 사람에 대한 정보가 포함된 로그 레코드를 받게 됩니다. 이 정보는 IAM 자격 증명을 기반으로 합니다.

PCI DSS 준수

IAM에서는 전자 상거래 웹사이트 운영자 또는 서비스 공급자에 의한 신용 카드 데이터의 처리, 저장 및 전송을 지원하며, Payment Card Industry(PCI) Data Security Standard(DSS) 준수를 검증 받았습니다. AWS PCI 규정 준수 패키지의 사본을 요청하는 방법 등 PCI DSS에 대해 자세히 알아보려면 [PCI DSS 레벨 1](#)을 참조하십시오.

많은 AWS 서비스와의 통합

IAM과 함께 사용할 수 있는 AWS 서비스의 목록은 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#)를 참조하십시오.

최종 일관성

IAM은 다른 많은 AWS 서비스처럼 [eventually consistent](#)에 해당됩니다. IAM에서는 전 세계의 Amazon 데이터 센터 내의 여러 서버로 데이터를 복제함으로써 고가용성을 구현합니다. 일부 데이터를 변경하겠다는 요청이 성공하면 변경이 실행되고 그 결과는 안전하게 저장됩니다. 그러나 변경 사항은 IAM에 두루 복제되어야 하고, 여기에는 일정한 시간이 걸립니다. 그러한 변경 사항에는 사용자, 그룹, 역할 또는 정책을 만들거나 업데이트한 것이 포함됩니다. 그러한 IAM 변경 사항을 애플리케이션의 중요한 고가용성 코드 경로에 포함시키지 않는 것이 좋습니다. 대신 자주 실행하지 않는 별도의 초기화 루틴이나 설정 루틴에서 IAM을 변경하십시오. 또한 프로덕션 워크플로우에서 변경 사항을 적용하기 전에 변경 사항이 전파되었는지 확인하십시오. 자세한 내용은 [변경 사항이 매번 즉시 표시되는 것은 아닙니다 \(p. 454\)](#) 단원을 참조하십시오.

무료 사용

AWS Identity and Access Management(IAM) 및 AWS Security Token Service(AWS STS)은 추가 비용 없이 AWS 계정에 제공되는 기능입니다. IAM 사용자 또는 AWS STS 임시 보안 자격 증명을 사용하여 다른 AWS 서비스에 액세스하는 경우에만 요금이 부과됩니다. 다른 AWS 제품 요금에 대한 자세한 내용은 [Amazon Web Services 요금 페이지](#)를 참조하십시오.

IAM에 액세스

다음 방법 중 하나를 사용하여 AWS Identity and Access Management(으)로 작업할 수 있습니다.

AWS Management 콘솔

콘솔은 IAM 및 AWS 리소스를 관리하기 위한 브라우저 기반 인터페이스입니다. 콘솔을 통한 IAM 액세스에 대한 자세한 내용은 [IAM 콘솔 및 로그인 페이지 \(p. 53\)](#) 단원을 참조하십시오. 콘솔 사용법을 안내하는 자습서는 [첫 번째 IAM 관리자 및 그룹 생성 \(p. 17\)](#)을 참조하십시오.

AWS 명령줄 도구

AWS 명령줄 도구를 통해 시스템 명령줄에서 명령을 실행하여 IAM 및 AWS 작업을 수행할 수 있습니다. 명령줄을 사용하는 것이 콘솔을 사용하는 것보다 더 빠르고 편리할 수 있습니다. AWS 작업을 수행하는 스크립트를 작성할 때도 명령줄 도구가 유용합니다.

AWS에서는 [AWS Command Line Interface\(AWS CLI\)](#) 및 [Windows PowerShell용 AWS 도구](#)라는 두 가지 명령줄 도구 세트를 제공합니다. AWS CLI 설치 및 사용에 대한 자세한 내용은 [AWS Command Line Interface 사용 설명서](#) 단원을 참조하십시오. Windows PowerShell용 도구 설치 및 사용에 대한 자세한 내용은 [Windows PowerShell용 AWS 도구 사용 설명서](#) 단원을 참조하십시오.

AWS SDK

AWS에서는 다양한 프로그래밍 언어 및 플랫폼(Java, Python, Ruby, .NET, iOS, Android 등)을 위한 라이브러리와 샘플 코드로 구성된 소프트웨어 개발 키트(SDK)를 제공합니다. SDK를 사용하면 편리하게 IAM 및 AWS에 프로그래밍 방식으로 액세스할 수 있습니다. 예를 들어 SDK는 요청에 암호화 방식으로 서명, 오류 관리 및 자동으로 요청 재시도와 같은 작업을 처리합니다. 다운로드 및 설치 방법을 비롯하여 AWS SDK에 대한 자세한 내용은 [Amazon Web Services용 도구](#) 페이지를 참조하십시오.

IAM HTTPS API

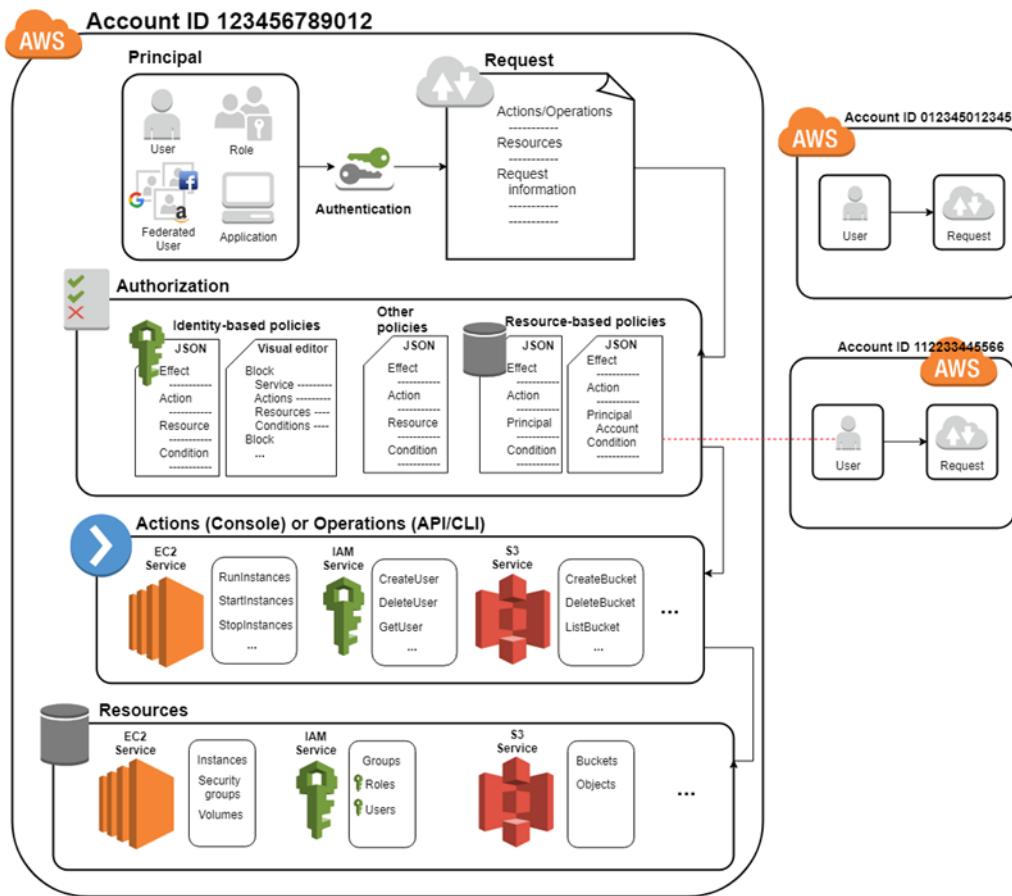
서비스로 직접 HTTPS 요청을 실행할 수 있는 IAM HTTPS API를 사용하여 프로그래밍 방식으로 IAM 및 AWS에 액세스할 수 있습니다. HTTPS API를 사용할 때는 자격 증명을 사용하여 요청에 디지털 방식으로 서명하는 코드를 포함해야 합니다. 자세한 내용은 [HTTP 쿼리 요청을 통한 API 호출 \(p. 570\)](#) 및 [IAM API Reference](#) 단원을 참조하십시오.

IAM 작동 방식 이해

사용자를 생성하기 전에 IAM 작동 방식을 이해해야 합니다. IAM은 계정에 대한 인증 및 권한 부여를 제어하는 데 필요한 인프라를 제공합니다. IAM 인프라에는 다음 요소가 포함되어 있습니다.

주제

- [약관 \(p. 4\)](#)
- [보안 주체 \(p. 4\)](#)
- [요청 \(p. 5\)](#)
- [인증 \(p. 5\)](#)
- [승인 \(p. 5\)](#)
- [작업 또는 연산 \(p. 6\)](#)
- [리소스 \(p. 6\)](#)



약관

IAM 용어에 대해 자세히 알아봅니다.

리소스

IAM에 저장된 사용자, 역할, 그룹 및 정책 객체입니다. 다른 AWS 서비스와 마찬가지로 IAM에서 리소스를 추가, 편집 및 제거할 수 있습니다.

ID

식별 및 그룹화에 사용되는 IAM 리소스 객체입니다. 여기에는 사용자, 그룹 및 역할이 포함됩니다.

객체

AWS가 인증에 사용하는 IAM 리소스 객체입니다. 여기에는 사용자 및 역할이 포함됩니다. 역할은 웹 자격 증명 또는 SAML을 통해 페더레이션된 사용자는 물론 사용자 또는 다른 계정의 IAM 사용자로 가정할 수 있습니다.

Principal

엔터티를 사용하여 AWS에 로그인하고 요청하는 사람 또는 애플리케이션입니다.

보안 주체

보안 주체란 AWS 리소스에 대한 작업을 요청할 수 있는 사람 또는 애플리케이션입니다. 보안 주체는 먼저 AWS 계정 루트 사용자로 로그인해야 합니다. 일별 작업에 대한 루트 사용자는 사용하지 않는 것이 가장 좋습니다.

습니다. 대신에, IAM 엔터티(사용자 및 역할)를 생성합니다. 애플리케이션이 AWS 계정에 액세스할 수 있도록 연동 사용자 또는 프로그래밍 방식의 액세스를 지원할 수 있습니다.

요청

보안 주체가 AWS Management 콘솔, AWS API 또는 AWS CLI를 사용하려고 시도하면 해당 보안 주체가 요청을 AWS에 전송합니다. 이 요청에는 다음 정보가 포함되어 있습니다.

- 작업 또는 작동 – 보안 주체가 수행하고자 하는 작업 또는 작동입니다. AWS CLI 또는 AWS API를 사용하여 AWS Management 콘솔 또는 작동의 작업을 수행할 수 있습니다.
- 리소스 – 수행된 작업 또는 작동에 따른 AWS 리소스 객체입니다.
- 보안 주체 – 엔터티(사용자 또는 역할)를 사용하여 요청을 보내는 사람 또는 애플리케이션입니다. 보안 주체에 대한 정보에는 보안 주체가 로그인하는 데 사용된 엔터티와 관련된 정책이 포함됩니다.
- 환경 데이터 – IP 주소, 사용자 에이전트, SSL 사용 상태 또는 시간대와 같은 정보입니다.
- 리소스 데이터 – 요청되는 리소스와 관련된 데이터. 여기에는 DynamoDB 테이블 이름 또는 Amazon EC2 인스턴스 태그와 같은 정보가 포함될 수 있습니다.

AWS에서 요청을 평가하고 승인하는 데 사용되는 요청 컨텍스트로 이 요청 정보를 수집합니다.

인증

보안 주체가 AWS에게 요청을 보내려면 IAM 엔터티를 사용하여 인증을 받아야 합니다(AWS에 로그인). Amazon S3 및 AWS STS와 같은 몇몇 서비스는 익명 사용자의 몇 가지 요청을 허용하지만 이것들은 규칙에 외입니다.

사용자로서 콘솔에서 인증하려면 사용자 이름 및 암호로 로그인해야 합니다. API 또는 AWS CLI에서 인증하려면 액세스 키 및 보안 키를 제공해야 합니다. 추가 보안 정보도 제공해야 할 수 있습니다. 예를 들어, AWS는 멀티 팩터 인증(MFA)을 사용하여 계정의 보안을 강화하는 것을 권장합니다. AWS가 인증할 수 있는 IAM 엔터티에 대한 자세한 정보는 [IAM 사용자 \(p. 63\)](#) 및 [IAM 역할 \(p. 153\)](#) 단원을 참조하십시오.

승인

또한 요청을 완료할 수 있는 권한이 있어야 합니다. AWS는 권한 부여 동안 요청 컨텍스트의 값을 사용하여 요청을 허용할지 거부할지 여부에 적용되는 정책을 점검합니다. 그런 다음 이것은 정책을 사용하여 요청을 허용하거나 거부할지 여부를 결정합니다. 대부분의 정책은 AWS에 [JSON 문서 \(p. 309\)](#)로 저장되며 보안 주체 엔터티에 대한 권한을 지정합니다. 요청이 권한 부여될지 여부에 영향을 미치는 [정책의 몇 가지 유형 \(p. 305\)](#)가 있습니다. 계정에서 AWS 리소스로의 액세스 권한을 사용자에게 제공하려면 자격 기반 정책만 필요합니다. 리소스 기반 정책은 [교차 계정 액세스 \(p. 445\)](#)를 허용하는 데 좋습니다. 다른 정책 유형은 고급 기능이며 조심스럽게 사용해야 합니다.

AWS는 요청 컨텍스트에 적용되는 각 정책을 확인합니다. 단일 권한 정책에 거부된 작업이 포함된 경우 AWS는 전체 요청을 거부하고 평가를 중지합니다. 이를 명시적 거부라고 합니다. 요청은 기본적으로 거부되므로 AWS는 적용 가능한 권한 정책이 요청의 모든 부분을 허용하는 경우에만 요청에 권한을 부여합니다. 단일 계정 내 요청 평가 로직은 다음 일반 규칙을 따릅니다.

- 기본적으로 모든 요청을 거부합니다. (일반적으로, AWS 계정 루트 사용자 증명을 사용하여 해당 계정의 리소스를 요청하는 경우는 항상 허용됩니다.)
- 권한 정책(자격 증명 기반 또는 리소스 기반)에 포함된 명시적 허용은 이 기본 작동을 재정의합니다.
- 조직 SCP, IAM 권한 경계 또는 세션 정책이 있는 경우 허용이 재정의됩니다. 하나 이상의 이러한 정책 유형이 존재하는 경우 이들 정책 유형 모두가 해당 요청을 허용해야 합니다. 그렇지 않은 경우 이 값은 둑시적으로 거부됩니다.
- 어떠한 정책의 명시적 거부도 허용을 무시합니다.

모든 유형의 정책 평가 방법에 대한 자세한 내용은 [정책 평가 로직 \(p. 531\)](#) 단원을 참조하십시오. 사용자가 다른 계정에서 요청해야 하는 경우 다른 계정의 정책에서 해당 사용자가 해당 리소스를 액세스하도록 허용해야 하며, 또한 요청하는 데 사용하는 IAM 엔터티에도 해당 요청을 허용하는 자격 증명 기반 정책이 있어야 합니다.

작업 또는 연산

요청이 인증 및 권한 부여된 후 AWS가 요청의 작업 또는 작동을 승인합니다. 작업은 서비스로 정의되며 리소스 보기, 생성, 편집 및 삭제와 같이 리소스에 대해 수행할 수 있는 사항입니다. 예를 들어, IAM은 사용자 리소스에 대해 다음 작업을 비롯하여 약 40개의 작업을 수행할 수 있도록 지원합니다.

- `CreateUser`
- `DeleteUser`
- `GetUser`
- `UpdateUser`

보안 주체가 작업을 수행할 수 있도록 허용하려면 보안 주체 또는 영향을 받은 리소스에 적용되는 필요한 작업을 정책에 포함해야 합니다. 각 서비스가 지원하는 작업, 리소스 유형, 조건 키 목록은 [??? 단원](#)을 참조하십시오.

리소스

AWS가 요청의 작업을 승인하면 계정 내의 관련 리소스에서 해당 작업을 수행할 수 있습니다. 리소스는 서비스 내에 존재하는 객체입니다. 예를 들어 Amazon EC2 인스턴스, IAM 사용자 및 Amazon S3 버킷이 있습니다. 서비스는 각 리소스에서 수행할 수 있는 일련의 작업을 정의합니다. 리소스에서 관련되지 않은 작업을 수행하도록 요청을 생성하면 해당 요청이 거부됩니다. 예를 들어 IAM 역할을 삭제하도록 요청하지만 IAM 그룹 리소스를 제공하지 않는 경우 요청이 실패합니다. 작업에 의해 영향을 받는 리소스를 식별하는 AWS 서비스 테이블을 보려면 [??? 단원](#)을 참조하십시오.

자격 증명 관리 개요: 사용자

보안 및 조직을 강화하기 위해 사용자 지정 권한으로 생성한 특정 사용자 자격 증명에 AWS 계정에 대한 액세스 권한을 부여할 수 있습니다. 기존 자격 증명을 AWS에 연동하여 그러한 사용자를 위해 액세스를 더욱 간소화할 수 있습니다.

주제

- [첫 액세스에만 해당: 루트 사용자 자격 증명 \(p. 6\)](#)
- [IAM 사용자 \(p. 7\)](#)
- [기존 사용자 연동 \(p. 7\)](#)

첫 액세스에만 해당: 루트 사용자 자격 증명

AWS 계정을 생성할 때 AWS로 로그인하는 데 사용하는 AWS 계정 루트 사용자 자격 증명을 생성합니다. 이 루트 사용자 자격 증명, 즉 계정을 생성할 때 입력한 이메일 주소와 암호를 사용하여 AWS Management 콘솔에 로그인할 수 있습니다. 이러한 이메일 주소 및 암호의 조합을 루트 사용자 자격 증명이라고도 합니다.

루트 사용자 자격 증명을 사용하면 AWS 계정의 모든 리소스에 완전히 무제한으로 액세스할 수 있습니다. 여기에는 결제 정보에 대한 액세스 및 암호 변경 권한이 포함됩니다. 이러한 수준의 액세스는 계정을 처음 설정했을 때 필요합니다. 그러나 일상적인 액세스에는 루트 사용자 자격 증명을 사용하지 않는 것이 좋습니다. 특

히 루트 사용자 자격 증명을 타인과 공유하면 타인이 내 계정에 무제한으로 액세스할 수 있으므로 공유하지 않는 것이 좋습니다. 루트 사용자에 부여된 권한을 제한할 수는 없습니다.

다음 단원에서는 본인 및 AWS 리소스를 사용하는 사람들에게 AWS 리소스에 대한 안전하고 제한된 액세스를 제공하기 위해 IAM을 사용하여 사용자 자격 증명 및 권한을 생성하고 관리하는 방법을 설명합니다.

IAM 사용자

AWS Identity and Access Management(IAM)의 '자격 증명'을 통해 '사용자의 정체'를 확인할 수 있습니다. 이를 흔히 인증이라고 합니다. 루트 사용자 자격 증명을 타인과 공유하는 대신, 조직의 사용자에 해당되는 계정 내에 개별 IAM 사용자를 생성할 수 있습니다. IAM 사용자는 별개의 계정이 아니라 해당 계정 내의 사용자입니다. 각 사용자는 고유의 AWS Management 콘솔 액세스 암호를 가질 수 있습니다. 또한 사용자가 계정의 리소스를 사용하기 위한 프로그래밍 방식의 요청을 할 수 있도록 각 사용자에 대한 개별 액세스 키를 생성할 수 있습니다. 다음 그림에서는 AWS 계정 하나에 Li, Mateo, DevApp1, DevApp2, TestApp1 및 TestApp2라는 사용자가 추가되었습니다. 각 사용자는 고유의 자격 증명을 가집니다.

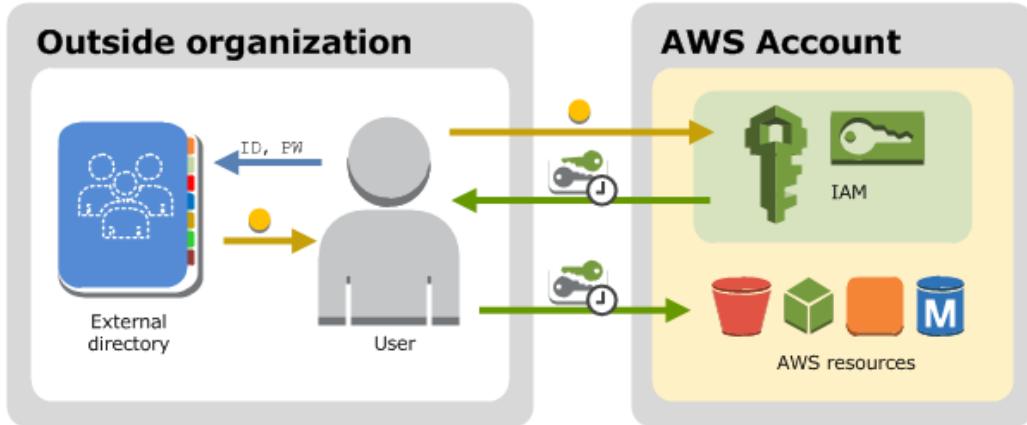
일부 사용자는 사실 애플리케이션입니다(예: DevApp1). IAM 사용자가 실제 사람일 필요는 없습니다. 회사 네트워크에서 실행하며 AWS 액세스를 필요로 하는 애플리케이션에 대한 액세스 키를 생성하기 위해 IAM 사용자를 생성할 수 있습니다.

본인을 위한 IAM 사용자를 생성한 다음, 본인에게 계정에 대한 관리 권한을 할당하는 것이 좋습니다. 그런 다음 해당 사용자로 로그인하여 필요에 따라 사용자를 추가할 수 있습니다.

기존 사용자 연동

조직 내 사용자에게 이미 인증 방법이 있는 경우(예: 회사 네트워크에 로그인) 해당 사용자를 위해 별도의 IAM 사용자를 생성할 필요가 없습니다. 대신 이러한 사용자 자격 증명을 AWS에 연동할 수 있습니다.

다음 다이어그램은 사용자가 IAM을 사용하여 AWS 계정의 리소스에 액세스하기 위한 임시 AWS 보안 자격 증명을 얻는 방법을 보여 줍니다.



연동은 다음과 같은 경우에 특히 유용합니다.

- 사용자가 이미 기업 디렉토리에 자격 증명을 보유한 경우

기업 디렉토리가 SAML 2.0(Security Assertion Markup Language 2.0)과 호환되는 경우, 기업 디렉토리를 구성하여 사용자에게 AWS Management 콘솔에 대한 Single-Sign On(SSO) 액세스를 제공할 수 있습니다. 자세한 내용은 [임시 자격 증명과 관련된 일반적인 시나리오 \(p. 264\)](#) 단원을 참조하십시오.

기업 디렉토리가 SAML 2.0과 호환되지 않는 경우, 자격 증명 브로커 애플리케이션을 생성하여 사용자에게 AWS Management 콘솔에 대한 Single-Sign On(SSO) 액세스를 제공할 수 있습니다. 자세한 내용

은 연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하는 URL 생성(사용자 지정 연동 브로커) (p. 188) 단원을 참조하십시오.

기업 디렉토리가 Microsoft Active Directory인 경우, AWS Directory Service를 사용하여 기업 디렉토리와 AWS 계정 간의 신뢰를 설정할 수 있습니다.

- 사용자가 이미 인터넷 자격 증명을 보유한 경우

사용자가 Login with Amazon, Facebook, Google 또는 OpenID Connect(OIDC) 호환 자격 증명 공급자 등의 인터넷 자격 증명 공급자를 통해 자신을 식별할 수 있도록 모바일 앱 또는 웹 기반 앱을 만들면, 해당 앱에서 연동을 통해 AWS에 액세스할 수 있습니다. 자세한 내용은 웹 자격 증명 연동에 대하여 (p. 162) 단원을 참조하십시오.

도움말

인터넷 자격 증명 공급자를 통해 자격 증명 연동을 사용하려면 Amazon Cognito(를) 사용하는 것이 좋습니다.

액세스 관리 개요: 권한 및 정책

AWS Identity and Access Management(IAM)의 액세스 관리를 통해 계정에서 보안 주체 엔터티에 허용된 권한을 정의할 수 있습니다. 보안 주체 엔터티란 IAM 엔터티(사용자 또는 역할)을 사용하여 인증된 사람 또는 애플리케이션입니다. 액세스 관리를 흔히 권한 부여라고 합니다. 정책을 생성하고 IAM 자격 증명(사용자, 사용자 그룹 또는 역할) 또는 AWS 리소스에 연결하여 AWS에서 액세스를 관리합니다. 정책은 자격 증명이나 리소스와 연결될 때 해당 권한을 정의하는 AWS의 객체입니다. AWS는 IAM 엔터티(사용자 또는 역할)가 요청을 보낼 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는지 여부를 결정합니다. 대부분의 정책은 AWS에 JSON 문서로서 저장됩니다. 정책 유형 및 활용에 대한 자세한 정보는 정책 및 권한 (p. 305) 단원을 참조하십시오.

정책 및 계정

AWS에서 하나의 계정을 관리하려면 정책을 사용하여 해당 계정 내 권한을 정의합니다. 여러 계정 전반의 권한을 관리하고자 한다면 사용자에 대한 권한을 관리하기가 더 어렵습니다. 교차 계정 권한에 대해 IAM 역할, 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 사용할 수 있습니다. 하지만 여러 계정을 소유하는 경우에는 이러한 권한을 쉽게 관리할 수 있도록 ACL 대신에 AWS Organizations 서비스를 사용하는 것이 좋습니다. 자세한 정보는 조직 사용 설명서의 AWS Organizations(이)란 무엇입니까? 단원을 참조하십시오.

정책 및 사용자

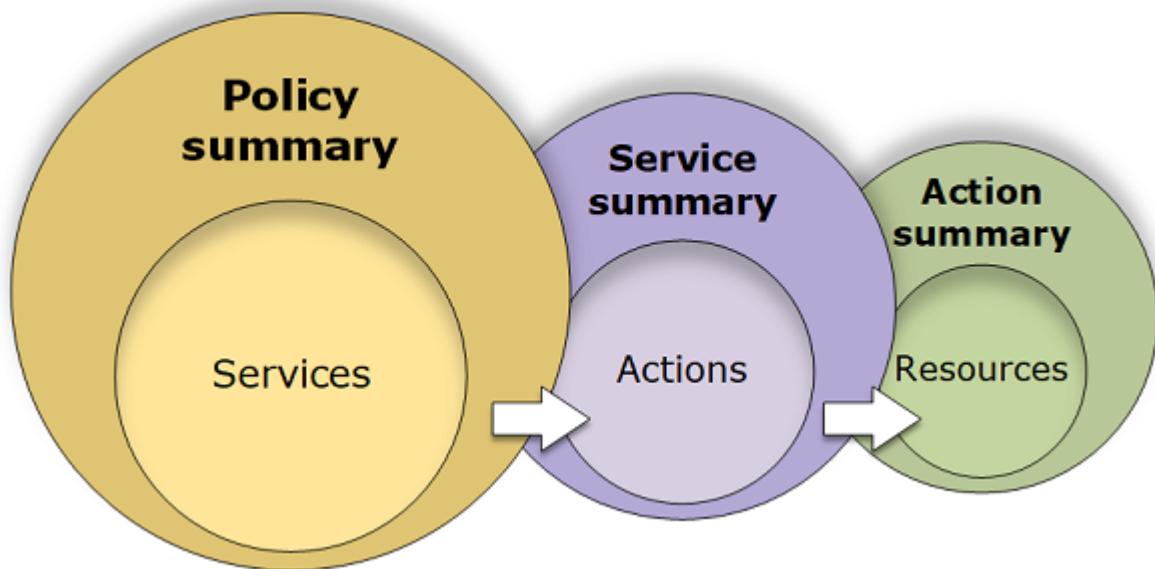
IAM 사용자는 서비스의 자격 증명입니다. IAM 사용자를 생성할 경우, 권한을 부여하지 않는 한 사용자는 계정 내에서 어떠한 것으로도 액세스할 수 없습니다. 사용자 또는 사용자가 속한 그룹에 연결된 정책인 자격 증명 기반 정책을 생성하여 사용자에게 권한을 부여합니다. 다음 예는 사용자가 us-east-2 리전 내의 123456789012 계정에서 Books 테이블의 모든 Amazon DynamoDB 작업(dynamodb:*)을 수행할 수 있도록 허용하는 JSON 정책을 보여 줍니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": "dynamodb:*",  
         "Resource": "arn:aws:dynamodb:us-east-2:123456789012:table/Books"}  
    ]  
}
```

이 정책을 IAM 사용자에게 연결한 후에는 해당 사용자가 이러한 DynamoDB 권한만 부여받습니다. 대부분의 사용자는 해당 사용자의 권한을 함께 나타내는 여러 정책을 부여받습니다.

명시적으로 허용되지 않은 작업 또는 리소스는 기본적으로 모두 거부됩니다. 예를 들어 앞서 다른 정책이 사용자에게 연결된 유일한 정책이라면 이 사용자는 Books 테이블에 대한 DynamoDB 작업만 수행할 수 있습니다. 다른 모든 테이블에 대한 작업은 금지됩니다. 마찬가지로 사용자는 Amazon EC2, Amazon S3 또는 기타 다른 AWS 서비스의 어떠한 작업도 수행할 수 없습니다. 그 이유는 권한 정책이 함께 작업할 이러한 서비스는 정책에 포함되지 않기 때문입니다.

IAM 콘솔에는 정책에서 각 서비스에 대해 허용되거나 거부되는 액세스 레벨, 리소스, 조건을 설명하는 정책 요약 테이블이 포함되어 있습니다. 정책은 3가지 테이블, 즉 [정책 요약 \(p. 419\)](#), [서비스 요약 \(p. 429\)](#), [작업 요약 \(p. 433\)](#)으로 요약됩니다. 정책 요약 테이블에는 서비스 목록이 포함되어 있습니다. 서비스 요약을 보려면 여기서 서비스를 선택합니다. 이 요약 테이블에는 작업 목록과 선택한 서비스에 대해 연결된 권한이 포함되어 있습니다. 해당 테이블에서 작업을 선택하여 작업 요약을 볼 수 있습니다. 이 테이블에는 리소스 목록과 선택한 작업에 대한 조건이 포함되어 있습니다.



사용자 페이지에서 해당 사용자에게 연결된 모든 정책(관리형 및 인라인)에 대한 정책 요약을 볼 수 있습니다. 정책 페이지에서 모든 관리형 정책에 대한 요약을 봅니다.

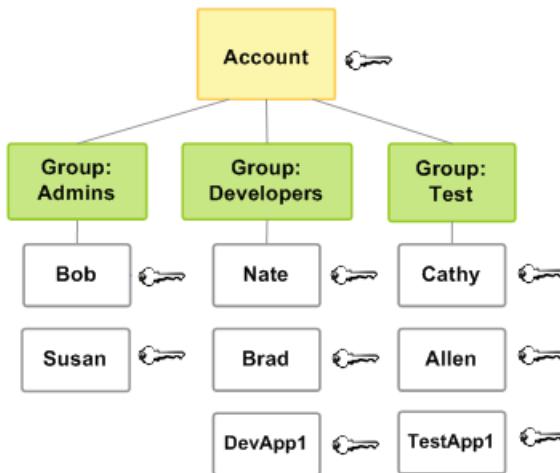
예를 들어 위 정책은 AWS Management 콘솔에 다음과 같이 요약됩니다.

Service	Access level	Resource	Request condition
Allow (1 of 102 services) Show remaining 101			
DynamoDB	Full access	TableName = Books	None

정책의 JSON 문서도 볼 수 있습니다. 요약 또는 JSON 문서를 보는 방법에 대한 자세한 정보는 [정책에 의해 부여된 권한 이해 \(p. 419\)](#) 단원을 참조하십시오.

정책 및 그룹

IAM 사용자를 IAM 그룹으로 구성하고 그룹에 정책을 연결할 수 있습니다. 이 경우 각 사용자는 별도의 자격 증명을 갖고 있지만 그룹에 연결된 정책에 명시된 권한이 그룹 내 모든 사용자에게 부여됩니다. 그룹을 사용하여 간편하게 권한을 관리하고 [IAM 모범 사례 \(p. 43\)](#)에 따를 수 있습니다.



사용자 또는 그룹에 여러 정책을 연결하여 다양한 권한을 부여할 수 있습니다. 이 경우 사용자의 권한은 각 정책의 조합으로 계산됩니다. 그러나 개별 작업 및 리소스에 대한 권한을 명시적으로 부여해야 사용자가 해당 권한을 가지게 되는 기본 원칙은 여전히 적용됩니다.

연동 사용자 및 역할

연동 사용자에게는 IAM 사용자처럼 AWS 계정에서 영구적인 ID가 부여되지 않습니다. 연동 사용자에게 권한을 부여하려면 역할이라고 하는 개체를 만들어 해당 역할의 권한을 정의할 수 있습니다. 연동 사용자가 AWS에 로그인하면 사용자가 역할과 연결되고 역할에 정의된 권한이 부여됩니다. 자세한 정보는 [타사 자격증명 공급자의 역할 만들기\(연동\)](#) (p. 215) 단원을 참조하십시오.

자격 증명 기반 정책 및 리소스 기반 정책

자격 증명 기반 정책은 IAM 사용자, 그룹 또는 역할과 같은 IAM 자격 증명에 연결할 수 있는 권한 정책입니다. 리소스 기반 정책은 Amazon S3 버킷 또는 IAM 역할 신뢰 정책과 같은 리소스에 연결하는 권한 정책입니다.

자격 증명 기반 정책은 자격 증명이 수행할 수 있는 작업, 대상 리소스 및 이에 관한 조건을 제어합니다. 자격 증명 기반 정책을 추가로 분류할 수 있습니다.

- 관리형 정책 – AWS 계정에 속한 다수의 사용자, 그룹 및 역할에게 독립적으로 연결할 수 있는 자격 증명 기반 정책입니다. 사용할 수 있는 관리형 정책은 두 가지가 있습니다.
 - AWS 관리형 정책 – AWS에서 생성 및 관리하는 관리형 정책입니다. 정책 사용이 처음이라면 AWS 관리형 정책 사용을 먼저 권장합니다.
 - 고객 관리형 정책 – 사용자가 자신의 AWS 계정에서 생성 및 관리하는 관리형 정책입니다. 고객 관리형 정책은 AWS 관리형 정책보다 정책에 대해 더욱 정밀하게 제어할 수 있습니다. 시작적 편집기에서 또는 JSON 정책 문서를 직접 생성하여 IAM 정책을 생성 및 편집할 수 있습니다. 자세한 정보는 [IAM 정책 만들기](#) (p. 377) 및 [IAM 정책 편집](#) (p. 402)을(를) 참조하십시오.
- 인라인 정책 – 자신이 생성 및 관리하며, 단일 사용자, 그룹 또는 역할에 직접 포함되는 정책입니다. 대부분의 경우 인라인 정책을 사용하지 않는 것이 좋습니다.

리소스 기반 정책은 지정된 보안 주체가 해당 리소스에 대해 수행할 수 있는 작업 및 이에 관한 조건을 제어합니다. 리소스 기반 정책은 인라인 정책이며, 관리형 리소스 기반 정책은 없습니다. 교차 계정 액세스를 활성화하려는 경우 전체 계정이나 다른 계정의 IAM 엔터티를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다.

IAM 서비스는 역할 신뢰 정책이라고 하는 리소스 기반 정책 유형 하나만 지원하며, 이 유형은 IAM 역할에 연결됩니다. IAM 역할은 리소스 기반 정책을 지원하는 자격 증명이자 리소스이므로 신뢰 정책과 자격 증명 기

반 정책 모두 IAM 역할에 연결해야 합니다. 신뢰 정책은 역할을 수입할 수 있는 보안 주체 엔터티(계정, 사용자, 역할 및 연합된 사용자)를 정의합니다. IAM 역할과 다른 리소스 기반 정책 간의 차이에 대해 알아보려면 [IAM 역할과 리소스 기반 정책의 차이 \(p. 257\)](#) 단원을 참조하십시오.

리소스 기반 정책을 지원하는 서비스를 보려면 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하십시오. 리소스 기반 정책에 대해 자세히 알아보려면 [자격 증명 기반 정책 및 리소스 기반 정책 \(p. 326\)](#) 단원을 참조하십시오.

IAM 외부의 보안 기능

IAM을 사용하여 AWS Management 콘솔 작업, [AWS 명령줄 도구](#) 작업 또는 [AWS SDK](#)를 통한 서비스 API 작업에 대한 액세스를 제어할 수 있습니다. 일부 AWS 제품은 리소스 보안을 위한 다른 방법도 지원합니다. 다음 목록은 전체는 아니지만 몇 가지 예에 해당합니다.

Amazon EC2

Amazon Elastic Compute Cloud에서는 키 페어를 사용하거나(Linux 인스턴스의 경우) 사용자 이름 및 암호를 사용해(Microsoft Windows 인스턴스의 경우) 인스턴스에 로그인합니다.

자세한 내용은 다음 문서를 참조하십시오.

- Linux 인스턴스용 Amazon EC2 사용 설명서의 [Amazon EC2 Linux 인스턴스 시작하기](#)
- Windows 인스턴스용 Amazon EC2 사용 설명서의 [Amazon EC2 Windows 인스턴스 시작하기](#)

Amazon RDS

Amazon Relational Database Service에서는 데이터베이스와 연결되어 있는 사용자 이름 및 암호를 사용해 데이터베이스 엔진에 로그인합니다.

자세한 내용은 Amazon RDS 사용 설명서의 [Amazon RDS 시작하기](#)를 참조하십시오.

Amazon EC2 및 Amazon RDS

Amazon EC2 및 Amazon RDS에서는 보안 그룹을 사용하여 인스턴스 또는 데이터베이스에 대한 트래픽을 제어합니다.

자세한 내용은 다음 문서를 참조하십시오.

- Linux 인스턴스용 Amazon EC2 사용 설명서의 [Linux 인스턴스에 대한 Amazon EC2 보안 그룹](#)
- Windows 인스턴스용 Amazon EC2 사용 설명서의 [Windows 인스턴스에 대한 Amazon EC2 보안 그룹](#)
- Amazon RDS 사용 설명서의 [보안 그룹 Amazon RDS](#).

Amazon WorkSpaces

Amazon WorkSpaces에서는 사용자 이름과 암호를 사용해 데스크톱에 로그인합니다.

자세한 내용은 Amazon WorkSpaces Administration Guide의 [Amazon WorkSpaces 시작하기](#)를 참조하십시오.

Amazon WorkDocs

Amazon WorkDocs에서는 사용자 이름과 암호를 사용해 로그인하여 공유 문서에 액세스합니다.

자세한 내용은 Amazon WorkDocs 관리 안내서의 [Amazon WorkDocs 시작하기](#)를 참조하십시오.

위와 같은 액세스 제어 방법들은 IAM과 다릅니다. IAM에서는 Amazon EC2 인스턴스를 생성 또는 종료하거나 새로운 Amazon WorkSpaces 데스크톱을 설정하는 등 AWS 제품의 관리 방식을 제어할 수 있습니다. 다시 말해서, IAM은 Amazon Web Services 요청을 통한 작업을 제어하는 데 효과적일 뿐만 아니라 AWS Management 콘솔에 대한 액세스를 제어하는 데도 이상적입니다. 단, IAM은 운영 체제(Amazon EC2), 데이

터베이스(Amazon RDS), 데스크톱(Amazon WorkSpaces) 또는 협업 사이트(Amazon WorkDocs)에 로그인하는 등의 작업에 대해서는 보안 관리를 지원하지 않습니다.

특정 AWS 제품을 이용해 작업할 때는 반드시 설명서를 읽고 해당 제품에 속한 모든 리소스의 보안 옵션을 살펴보시기 바랍니다.

공통 작업의 빠른 링크

다음은 IAM과 연결되어 있는 공통 작업에 대한 도움말을 살펴볼 수 있는 링크입니다.

IAM 사용자로 로그인

[IAM 사용자가 AWS에 로그인하는 방법 \(p. 69\)](#) 단원을 참조하십시오.

IAM 사용자 암호 관리

결제 정보에 대한 액세스를 포함하여 AWS Management 콘솔에 액세스하려면 암호가 필요합니다.

AWS 계정 루트 사용자에 대한 내용은 [AWS 계정 루트 사용자 암호 변경 \(p. 78\)](#) 단원을 참조하십시오.

IAM 사용자에 대한 내용은 [IAM 사용자의 암호 관리 \(p. 82\)](#) 단원을 참조하십시오.

IAM 사용자 권한 관리

AWS 계정에 속한 IAM 사용자들에게 권한을 부여할 때는 정책을 사용합니다. 생성 시점에서는 IAM 사용자들에게 AWS 리소스를 사용할 수 있는 권한이 없기 때문에 권한을 추가해야 합니다.

자세한 내용은 [IAM 정책 관리 \(p. 377\)](#) 단원을 참조하십시오.

AWS 계정에 속한 사용자 표시 및 자격 증명 정보 가져오기

[AWS 계정의 자격 증명 보고서 가져오기 \(p. 135\)](#) 단원을 참조하십시오.

멀티 팩터 인증(MFA) 추가

가상 MFA 디바이스를 추가하려면 다음 중 하나 단원을 참조하십시오.

- [AWS 계정 루트 사용자용 가상 MFA 디바이스 활성화\(콘솔\) \(p. 100\)](#)
- [IAM 사용자에 대한 가상 MFA 디바이스 활성화\(콘솔\) \(p. 99\)](#)

U2F 보안 키를 추가하려면 다음 중 하나 단원을 참조하십시오.

- [AWS 계정 루트 사용자에 대한 U2F 보안 키 활성화\(콘솔\) \(p. 105\)](#)
- [다른 IAM 사용자에 대한 U2F 보안 키 활성화\(콘솔\) \(p. 104\)](#)

하드웨어 MFA 디바이스를 추가하려면 다음 중 하나 단원을 참조하십시오.

- [AWS 계정 루트 사용자용 하드웨어 MFA 디바이스 활성화\(콘솔\) \(p. 110\)](#)
- [다른 IAM 사용자에 대해 하드웨어 MFA 디바이스 활성화\(콘솔\) \(p. 109\)](#)

액세스 키 가져오기

[AWS SDK, AWS 명령줄 도구](#) 또는 API 작업을 사용하여 AWS 요청을 하려면 액세스 키가 필요합니다.

Important

보안 액세스 키는 액세스 키를 생성하는 시점에만 보고 다운로드할 수 있습니다. 이후로는 보안 액세스 키를 보거나 복구할 수 없습니다. 하지만 보안 액세스 키를 분실한 경우에는 새로운 액세스 키를 생성할 수 있습니다.

AWS 계정의 경우 [AWS 계정을 위한 액세스 키 관리](#) 단원을 참조하십시오.

IAM 사용자에 대한 내용은 [IAM 사용자의 액세스 키 관리 \(p. 88\)](#) 단원을 참조하십시오.

사용자 또는 역할 태그 지정

AWS SDK 중 하나를 통해 IAM 콘솔, AWS CLI 또는 API를 사용하여 IAM 사용자 또는 역할에 태그를 지정할 수 있습니다.

IAM 태그 지정에 대한 자세한 내용은 [IAM 엔터티 태그 지정 \(p. 259\)](#) 단원을 참조하십시오.

IAM에서의 태그 관리 방법에 대한 자세한 내용은 [IAM 엔터티에 대한 태그 관리\(콘솔\) \(p. 262\)](#) 단원을 참조하십시오.

IAM 태그를 사용하여 AWS에 대한 액세스를 제어하는 방법에 대한 자세한 내용은 [IAM 태그를 사용한 액세스 제어 \(p. 336\)](#) 단원을 참조하십시오.

AWS 시작하기

이 설명서에서는 IAM 서비스에 대해 주로 다릅니다. AWS를 시작하는 방법과 여러 서비스를 사용하여 프로젝트 구축 및 개시 등과 같은 문제를 해결하는 방법을 알아보려면 [리소스 센터 시작하기](#) 단원을 참조하십시오.

설정

AWS Identity and Access Management(IAM)은 Amazon Web Services(AWS) 및 사용자 계정 리소스에 대한 액세스를 안전하게 제어합니다. IAM은 또한 계정 자격 증명을 비밀로 유지합니다. 그 밖에 IAM은 AWS 계정에 다수의 IAM 사용자를 생성하거나, 기업 디렉터리와 자격 증명을 연동하여 임시 액세스를 허용할 수도 있습니다. 여러 AWS 계정의 리소스에 액세스할 수 있는 경우도 있습니다.

하지만 IAM을 사용하지 않을 경우에는 다수의 AWS 계정을 생성하거나(각 계정마다 AWS 제품 결제 및 구독을 따로 해야 합니다), 혹은 직원들이 단일 AWS 계정의 보안 자격 증명을 공유해야 합니다. 그 뿐만 아니라 IAM이 없으면 특정 사용자 또는 시스템의 작업이나 사용하는 AWS 리소스를 제어할 수 없습니다.

이번 안내서에서는 IAM의 개념에 대해 간략히 살펴보고, 비즈니스 사용 사례를 비롯한 AWS 권한 및 정책에 대해 설명하겠습니다.

주제

- [IAM을 사용하여 AWS 리소스에 대한 사용자 액세스를 허용하는 방법 \(p. 14\)](#)
- [IAM을 사용하려면 가입해야 합니까? \(p. 15\)](#)
- [추가 리소스 \(p. 15\)](#)

IAM을 사용하여 AWS 리소스에 대한 사용자 액세스를 허용하는 방법

IAM을 사용하여 AWS 리소스에 대한 액세스를 제어할 수 있는 몇 가지 방법이 있습니다.

액세스 유형	왜 사용해야 합니까?	자세한 정보는 어디에서 얻을 수 있습니까?
AWS 계정에 속한 사용자 액세스	AWS 계정에 사용자를 추가하고 싶거나, IAM을 사용하여 사용자를 생성한 후 그 권한을 관리하려고 합니다.	AWS Management 콘솔을 사용하여 사용자를 생성한 후 AWS 계정에서 사용자 권한을 관리하는 방법에 대한 자세한 내용은 시작 (p. 16) 단원을 참조하십시오. IAM API 또는 AWS Command Line Interface를 사용하여 AWS 계정에 사용자를 생성하는 방법에 대한 자세한 내용은 첫 번째 IAM 관리자 및 그룹 생성 (p. 17) 단원을 참조하십시오. IAM 사용자 작업에 대한 자세한 내용은 자격 증명 (사용자, 그룹, 및 역할) (p. 61) 단원을 참조하십시오.
권한 부여 시스템과 AWS 사이의 자격 증명 연동을 통한 AWS 사용자가 아닌 사용자의 액세스	자격 증명 및 권한 부여 시스템에 AWS 사용자가 아닌 사용자가 있으며, 이 사용자들이 AWS 리소스에 액세스해야 합니다.	보안 토큰을 사용하여 회사 디렉터리와 자격 증명을 연동함으로써 AWS 계정 리소스에 대한 사용자 액세스를 허용하는 방법에 대한 자세한 내용은 임시 보안 자격 증명 (p. 263) 을 참조하십시오. AWS Security Token Service API에 대한 자세한 내용은 AWS Security Token Service API Reference 를 참조하십시오.
AWS 계정 간 교차 계정 액세스	일부 AWS 리소스에 대한 액세스를 AWS 계정에 속한 사용자와 공유하려고 합니다.	IAM을 사용해 다른 AWS 계정에게 권한을 부여하는 방법에 대한 자세한 내용은 역할 용어 및 개념 (p. 153) 단원을 참조하십시오.

IAM을 사용하려면 가입해야 합니까?

아직 AWS 계정이 없다면 IAM을 사용할 계정을 하나 생성해야 합니다. 하지만 IAM을 사용하려고 따로 가입 할 필요는 없습니다. IAM 사용은 무료입니다.

Note

IAM은 IAM과 통합된 AWS 제품에서만 사용할 수 있습니다. IAM 지원 서비스 목록은 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하십시오.

AWS에 가입하려면

1. <https://aws.amazon.com/>을 열고 Create an AWS Account(AWS 계정 생성)를 선택합니다.

Note

전에 AWS 계정 루트 사용자 자격 증명을 사용하여 AWS Management 콘솔에 로그인한 적이 있는 경우 Sign in to a different account(다른 계정으로 로그인)를 선택합니다. 전에 IAM 자격 증명을 사용하여 콘솔에 로그인한 적이 있는 경우 Sign-in using root account credentials(루트 계정 자격 증명으로 로그인)를 선택합니다. 그런 다음 Create a new AWS account(새 AWS 계정 생성)를 선택합니다.

2. 온라인 지시 사항을 따릅니다.

등록 절차 중 전화를 받고 전화 키패드를 사용하여 확인 코드를 입력하는 과정이 있습니다.

추가 리소스

아래는 IAM 사용에 도움이 될 수 있는 몇 가지 리소스들입니다.

- AWS 계정 자격 증명 관리: AWS General Reference의 [AWS 보안 자격 증명](#)
- 시작하기 및 [IAM이란? \(p. 1\)](#)에 대해 자세히 알아보기
- IAM에 사용할 명령줄 인터페이스 설정. 교차 플랫폼 AWS CLI의 경우에는 [AWS 명령줄 인터페이스 설명서](#) 또는 [IAM CLI 참조](#)를 참조하십시오. 그 밖에 Windows PowerShell을 사용해 IAM을 관리할 수도 있습니다. 자세한 내용은 [Windows PowerShell을 위한 AWS 도구 설명서](#) 또는 [IAM Windows PowerShell 참조](#)를 참조하십시오.
- 편리한 프로그래밍 방식의 IAM 액세스를 위한 AWS SDK 다운로드: [Amazon Web Services 도구](#)
- FAQ 보기: [AWS Identity and Access Management FAQ](#)
- 기술 지원: [AWS Support 센터](#)
- 프리미엄 기술 지원: [AWS Premium Support 센터](#)
- AWS 용어 정의 찾기: [Amazon Web Services 글로서리](#)
- 커뮤니티 지원: [IAM 토론 포럼](#)
- AWS 문의: [문의처](#)

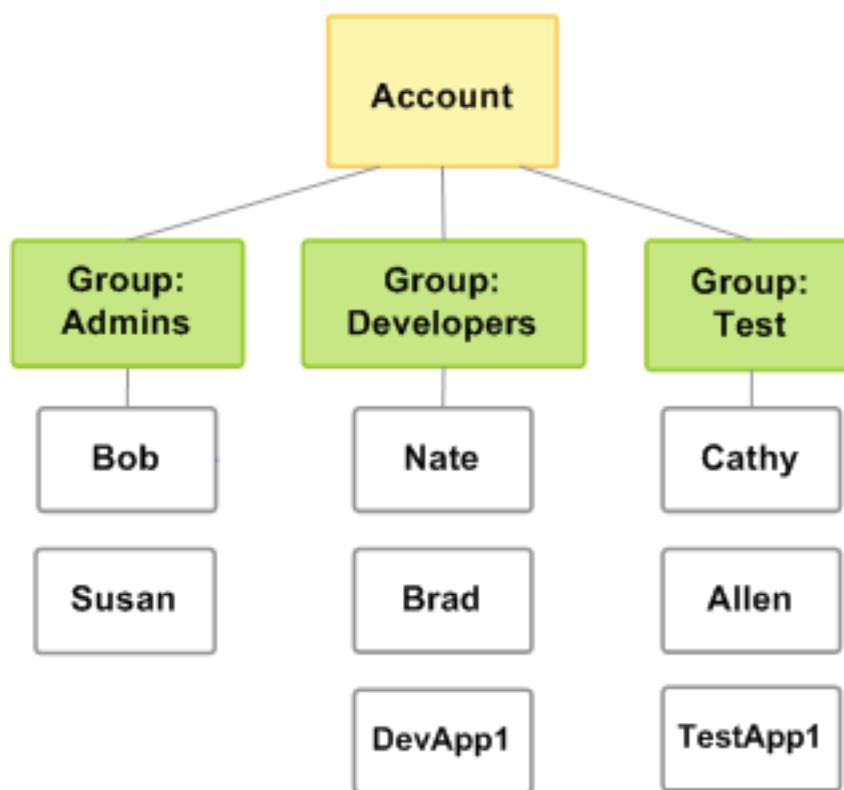
시작

이번 주제에서는 AWS 계정에 AWS Identity and Access Management(IAM) 사용자를 생성하여 AWS 리소스에 대한 액세스 권한을 부여하는 방법에 대해 살펴보겠습니다. 먼저 그룹이나 사용자를 생성하기 전에 알고 있어야 할 IAM 개념 몇 가지에 대해 학습한 후 AWS Management 콘솔을 사용해 필요한 작업을 실행하는 방법에 대해 알아볼 것입니다. 첫 번째 작업은 AWS 계정에 관리자 그룹을 구성하는 방법입니다. AWS 계정의 관리자 그룹은 필수는 아니지만 강력히 권장하는 구성입니다.

Note

이 설명서에서는 IAM 서비스에 대해 주로 다릅니다. AWS를 시작하는 방법과 여러 서비스를 사용하여 프로젝트 구축 및 개시 등과 같은 문제를 해결하는 방법을 알아보려면 [리소스 센터 시작하기](#)를 참조하십시오.

아래 그림은 AWS 계정을 3개 그룹으로 구성한 간단한 예입니다. 하나는 책임이 비슷한 사용자들을 모아놓은 그룹입니다. 이 예에서는 관리자 그룹(Admins라고 표시된 그룹)에 해당합니다. 그 밖에도 Developers 그룹과 Test 그룹이 있습니다. 각 그룹은 사용자가 다수입니다. 그리고 그림과 다르지만 각 사용자는 하나 이상의 그룹에 속할 수 있습니다. 하지만 그룹이 다른 그룹에 포함될 수는 없습니다. 이제 정책을 사용하여 권한을 그룹에게 부여합니다.



다음 절차에서는 아래 작업을 실행하게 됩니다.

- Administrators 그룹을 생성한 후 AWS 계정의 모든 리소스에 액세스할 수 있는 권한을 그룹에게 부여합니다.
- 직접 사용자를 생성하여 Administrators 그룹에 추가합니다.

- AWS Management 콘솔에 로그인할 수 있도록 사용자 암호를 생성합니다.

유효한 모든 AWS 계정 리소스에 액세스할 수 있는 권한을 Administrators 그룹에게 부여합니다. 여기에서 유효한 리소스란 사용 중이거나 등록한 모든 AWS 제품을 의미합니다. Administrators 그룹 사용자는 AWS 계정의 보안 자격 증명만 제외하고 AWS 계정 정보에 액세스할 수 있습니다.

주제

- [첫 번째 IAM 관리자 및 그룹 생성 \(p. 17\)](#)
- [첫 번째 IAM 위임 사용자 및 그룹 생성 \(p. 20\)](#)
- [사용자의 계정 로그인 방법 \(p. 22\)](#)

첫 번째 IAM 관리자 및 그룹 생성

Important

애플리케이션이나 웹 사이트에 Amazon 광고를 설정하려다 이 페이지로 오게 된 경우, [Product Advertising API 개발자로서 시작하기](#) 단원을 참조하십시오.

AWS 계정 루트 사용자가 필요하지 않은 작업에는 루트 사용자를 사용하지 않는 것이 바람직한[모범 사례 \(p. 43\)](#)입니다. 그 대신에 관리자 액세스 권한이 필요한 사람마다 새 IAM 사용자를 생성하십시오. 그 다음에는 AdministratorAccess 관리형 정책을 연결하는 "관리자" 그룹에 사용자를 배치하여 그 사용자들을 관리자로 만듭니다.

그 후에 관리자 그룹에 속한 사용자들은 AWS 계정에 대한 그룹, 사용자 등을 설정해야 합니다. 향후 모든 상호작용은 루트 사용자 대신에 AWS 계정 사용자와 그들의 고유 키를 통해 이루어져야 합니다. 하지만 일부 계정 및 서비스 관리 작업을 수행하려면 루트 사용자 계정 자격 증명을 사용하여 로그인해야 합니다. 루트 사용자로 로그인해야 하는 작업을 보려면 [계정 루트 사용자가 필요한 AWS 작업](#) 단원을 참조하십시오.

관리자 IAM 사용자 및 그룹 생성(콘솔)

AWS Management 콘솔 이번 섹션에서는 IAM 사용자를 직접 생성하고 그 사용자를 연결된 관리형 정책에 따라 관리자 권한을 보유한 그룹에 추가하는 방법에 대해 살펴보겠습니다.

관리자 사용자를 직접 생성하여 관리자 그룹에 추가하려면(콘솔)

1. <https://console.aws.amazon.com/iam/>에서 AWS 계정 루트 사용자 이메일 주소 및 암호를 사용하여 IAM 콘솔에 [AWS 계정 루트 사용자](#) 로그인합니다.

Note

Administrator IAM 사용자를 사용하는 아래 모범 사례를 준수하고, 루트 사용자 자격 증명을 안전하게 보관해 두는 것이 좋습니다. 몇 가지 [계정 및 서비스 관리 작업](#)을 수행하려면 반드시 루트 사용자로 로그인해야 합니다.

2. 탐색 창에서 사용자와 Add user(사용자 추가)를 차례로 선택합니다.
3. [User name]에 **Administrator**를 입력합니다.
4. AWS Management 콘솔 access(콘솔 액세스) 옆의 확인란을 선택하고 Custom password(사용자 지정 암호)를 선택한 다음, 텍스트 상자에 새 암호를 입력합니다. 초기 상태는 AWS가 새로운 사용자가 로그인할 때 새 암호를 만들도록 요구합니다. 선택적으로 User must create a new password at next sign-in(사용자는 다음번 로그인 시 새 암호를 생성해야 합니다) 옆 확인란의 선택을 최소하여 새로운 사용자가 로그인한 후 암호를 재설정할 수 있습니다.
5. 다음: 권한을 선택합니다.
6. 권한 설정 페이지에서 그룹에 사용자 추가(Add user to group)를 선택합니다.
7. Create group을 선택합니다.
8. 그룹 생성 대화 상자의 그룹 이름에 **Administrators**를 입력합니다.

9. 정책 유형을 선택한 후 Job function(직무)을 선택하여 표 내용을 필터링합니다.
10. 정책 목록에서 AdministratorAccess 옆의 확인란을 선택합니다. 그런 다음 Create group을 선택합니다.

Note

AdministratorAccess 권한을 사용하여 AWS Billing and Cost Management 콘솔에 액세스 하려면 먼저 결제에 대한 IAM 사용자 및 역할의 액세스 권한을 활성화해야 합니다. 이를 위해 [결제 콘솔에 액세스를 위임하기 위한 자습서 1단계 \(p. 25\)](#)의 지침을 따르십시오.

11. 그룹 목록으로 돌아가 새로 만든 그룹 옆의 확인란을 선택합니다. 목록에서 그룹을 확인하기 위해 필요 한 경우 Refresh를 선택합니다.
12. 다음: 태그 지정을 선택합니다.
13. (선택 사항) 태그를 키-값 페어로 연결하여 메타데이터를 사용자에게 추가합니다. IAM에서의 태그 사용에 대한 자세한 정보는 [IAM 엔터티 태그 지정 \(p. 259\)](#) 단원을 참조하십시오.
14. Next: Review를 선택하여 새 사용자에 추가될 그룹 멤버십의 목록을 확인합니다. 계속 진행할 준비가 되었으면 Create user를 선택합니다.

이와 동일한 절차에 따라 그룹이나 사용자를 추가 생성하여 사용자에게 AWS 계정 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 사용자 권한을 특정 AWS 리소스로 제한하는 정책을 사용하는 방법은 [액세스 관리 \(p. 304\)](#) 및 [IAM 자격 증명 기반 정책 예제 \(p. 341\)](#) 단원을 참조하십시오. 그룹을 생성한 후 추가로 사용자를 추가하려면 [IAM 그룹에서 사용자 추가 및 제거 \(p. 149\)](#) 단원을 참조하십시오.

IAM 사용자 및 그룹 생성(AWS CLI)

앞 단원의 절차를 따랐다면, AWS Management 콘솔을 사용하여 AWS 계정에 IAM 사용자를 생성하는 한편, 관리자 그룹을 설정했을 것입니다. 이 단원에서는 그룹을 생성하는 다른 방법을 소개합니다.

개요: 관리자 그룹 설정

1. 그룹을 생성하고 이름을 지정합니다(예: Admins). 자세한 정보는 [그룹 생성\(AWS CLI\) \(p. 18\)](#) 단원을 참조하십시오.
2. 그룹 관리 권한(모든 AWS 작업 및 리소스에 대한 액세스 권한)을 부여하는 정책을 연결합니다. 자세한 정보는 [그룹에 정책 연결\(AWS CLI\) \(p. 19\)](#) 단원을 참조하십시오.
3. 그룹에 한 명 이상의 사용자를 추가합니다. 자세한 정보는 [AWS 계정의 IAM 사용자 생성 \(p. 65\)](#) 단원을 참조하십시오.

그룹 생성(AWS CLI)

이 단원에서는 IAM 시스템에 그룹을 생성하는 방법을 소개합니다.

관리자 그룹을 생성하려면(AWS CLI)

1. `aws iam create-group` 명령과 선택한 그룹 이름을 입력합니다. 그룹 이름에 경로를 포함시킬 수도 있습니다. 경로에 대한 자세한 정보는 [표시 이름 및 경로 \(p. 480\)](#) 단원을 참조하십시오. 이름은 문자, 숫자, 그리고 다음과 같은 기호로 구성될 수 있습니다. 더하기(+), 등호(=), 쉼표(,), 마침표(.) 및 밑줄(_), 하이픈(-). 이름은 대소문자를 구분하지 않으며 최대 128자입니다.

이 예제에서는 Admins라는 그룹을 생성합니다.

```
aws iam create-group --group-name Admins
{
    "Group": {
        "Path": "/",
        "CreateDate": "2014-06-05T20:29:53.622Z",
        "GroupId": "ABCDEFGHABCDEFGHABCDE",
```

```
        "Arn": "arn:aws:iam::123456789012:group/Admins",
        "GroupName": "Admins"
    }
}
```

2. `aws iam list-groups` 명령을 입력하여 AWS 계정의 그룹을 나열하고 해당 그룹이 생성되었는지 확인합니다.

```
aws iam list-groups
{
    "Groups": [
        {
            "Path": "/",
            "CreateDate": "2014-06-05T20:29:53.622Z",
            "GroupId": "ABCDEFGHABCDEFGHABCDE",
            "Arn": "arn:aws:iam::123456789012:group/Admins",
            "GroupName": "Admins"
        }
    ]
}
```

응답에는 새 그룹에 대한 Amazon 리소스 이름(ARN)이 포함되어 있습니다. ARN은 AWS에서 리소스를 식별하는데 사용하는 표준 형식입니다. ARN의 12자리 숫자는 AWS 계정 ID입니다. 그룹에 할당한 표시 이름(Admins)은 그룹 ARN의 끝에 나타납니다.

그룹에 정책 연결(AWS CLI)

이 단원에서는 그룹의 사용자가 AWS 계정의 리소스에 대해 작업을 수행할 수 있도록 허용하는 정책의 연결 방법을 보여 줍니다. 방법은 Admins 그룹에 AdministratorAccess라는 [AWS 관리형 정책](#) (p. 312)을 연결하는 것입니다. 정책에 대한 자세한 정보는 [액세스 관리](#) (p. 304) 단원을 참조하십시오.

모든 관리자 권한을 부여하는 정책을 추가하려면(AWS CLI)

1. `aws iam attach-group-policy` 명령을 입력하여 Admins 그룹에 AdministratorAccess라는 정책을 연결합니다. 이 명령은 AdministratorAccess라는 AWS 관리형 정책의 ARN을 사용합니다.

```
aws iam attach-group-policy --group-name Admins --policy-arn arn:aws:iam::aws:policy/AdministratorAccess
```

명령이 성공하면 응답이 없습니다.

2. `aws iam list-attached-group-policies` 명령을 입력하여 Admins 그룹에 정책이 연결되었는지 확인합니다.

```
aws iam list-attached-group-policies --group-name Admins
```

응답에는 Admins 그룹에 연결된 정책 이름이 나열됩니다. 다음과 같은 응답은 Admins 그룹에 AdministratorAccess라는 정책이 연결되었다는 것을 알려 줍니다.

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        }
    ],
    "IsTruncated": false
}
```

`aws iam get-policy` 명령을 통해 특정 정책의 콘텐츠를 확인할 수 있습니다.

Important

Administrators 그룹을 설정한 후, 한 명 이상의 사용자를 추가해야 합니다. 그룹에 사용자를 추가하는 방법에 대한 자세한 정보는 [AWS 계정의 IAM 사용자 생성 \(p. 65\)](#) 단원을 참조하십시오.

관련 리소스

Amazon Web Services 일반 참조에서 수록된 관련 내용은 다음 리소스 단원을 참조하십시오.

- [계정 루트 사용자가 필요한 AWS 작업](#)

IAM 사용 설명서에 수록된 관련 내용은 다음 리소스 단원을 참조하십시오.

- [서비스에서 마지막으로 액세스한 데이터를 사용하여 권한 줄이기 \(p. 409\)](#)
- [자습서: Billing 콘솔에 대한 액세스 권한 위임 \(p. 25\)](#)

첫 번째 IAM 위임 사용자 및 그룹 생성

AWS 계정에서 여러 사용자를 지원하려면 다른 사용자가 허용된 작업만 수행할 수 있도록 권한을 위임해야 합니다. 이렇게 하려면 해당 사용자에게 필요한 권한이 있는 IAM 그룹을 생성한 다음 필요에 따라 IAM 사용자를 필요한 그룹에 추가합니다. 이 프로세스를 사용하여 전체 AWS 계정에 대한 그룹, 사용자 및 권한을 설정할 수 있습니다.

이 솔루션은 AWS 관리자가 수동으로 사용자 및 그룹을 관리할 수 있는 중소 규모 조직에서 가장 적합합니다. 대규모 조직에서는 [사용자 지정 IAM 역할 \(p. 188\)](#), [페더레이션 \(p. 161\)](#) 또는 [Single Sign-On](#)을 사용할 수 있습니다.

위임 IAM 사용자 및 그룹 생성(콘솔)

AWS Management 콘솔을 사용하여 위임된 권한이 있는 IAM 그룹을 생성한 다음 다른 사용자에 대해 IAM 사용자를 만들어 그룹에 추가할 수 있습니다.

다른 사용자에 대해 위임 그룹 및 사용자를 생성하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 왼쪽의 탐색 창에서 정책을 선택합니다.

정책을 처음으로 선택하는 경우 Welcome to Managed Policies 페이지가 나타납니다. [Get Started]를 선택합니다.

3. [Create policy]를 선택합니다.
4. JSON 탭을 선택한 다음 창의 오른쪽에서 관리형 정책 가져오기를 선택합니다.
5. 관리형 정책 가져오기 창에 **power**를 입력하여 정책 목록을 줄입니다. 그런 다음 PowerUserAccess AWS 관리형 정책 옆에 있는 버튼을 선택합니다.
6. [Import]를 선택합니다.

가져온 정책이 JSON 정책에 추가됩니다.

7. [Review policy]를 선택합니다.
8. 검토 페이지의 이름에 **PowerUserExampleCorp**를 입력합니다. 설명에 **Allows full access to all services except those for user management**을 입력합니다. 그런 다음 [Create policy]를 선택하여 작업을 저장합니다.

9. 탐색 창에서 그룹을 선택한 다음, 새 그룹 생성을 선택합니다.
10. 그룹 이름 상자에 **PowerUsers**를 입력합니다.
11. 정책 목록에서 PowerUserExampleCorp 옆의 확인란을 선택합니다. 그런 다음 [Next Step]을 선택합니다.
12. Create Group을 선택합니다.
13. 탐색 창에서 사용자와 Add user(사용자 추가)를 차례로 선택합니다.
14. [User name]에 **mary.major@examplecorp.com**를 입력합니다.
15. 다른 사용자 추가를 선택하고 두 번째 사용자에 대해 **diego.ramirez@examplecorp.com**을 입력합니다.
16. AWS Management 콘솔 액세스 옆의 확인란을 선택하고 자동 생성된 암호를 선택합니다. 초기 상태는 AWS가 새로운 사용자가 로그인할 때 새 암호를 만들도록 요구합니다. 새로운 사용자가 로그인한 후 암호를 재설정할 수 있도록 사용자가 다음에 로그인할 때 새 암호를 생성해야 합니다 확인란의 선택을 취소합니다.
17. 다음: 권한을 선택합니다.
18. 권한 설정 페이지에서 그룹에 사용자 추가를 선택한 다음 PowerUsers 옆의 확인란을 선택합니다.
19. 다음: 태그 지정을 선택합니다.
20. (선택 사항) 태그를 키-값 페어로 연결하여 메타데이터를 사용자에게 추가합니다. IAM에서의 태그 사용에 대한 자세한 정보는 [IAM 엔터티 태그 지정 \(p. 259\)](#) 단원을 참조하십시오.
21. Next: Review를 선택하여 새 사용자에 추가될 그룹 멤버십의 목록을 확인합니다. 계속 진행할 준비가 되었으면 사용자 생성을 선택합니다.
22. 새 사용자의 암호를 다운로드하거나 복사하여 사용자에게 안전하게 전달합니다. 별도로 사용자에게 [IAM 사용자 콘솔 페이지에 대한 링크 \(p. 53\)](#)와 방금 생성한 사용자 이름을 제공합니다.

그룹 권한 줄이기

PowerUser 그룹의 멤버는 사용자 관리 작업(예: IAM 및 조직)을 제공하는 일부 서비스를 제외한 모든 서비스에 대해 전체 권한을 갖습니다. 사전 정의된 비활성 기간(예: 90일)이 지난 후에는 그룹 멤버가 액세스한 서비스를 검토할 수 있습니다. 그런 다음 팀에 필요한 서비스만 포함하도록 PowerUserExampleCorp 정책의 권한을 줄일 수 있습니다.

서비스에서 마지막으로 액세스한 데이터에 대한 자세한 정보는 [서비스에서 마지막으로 액세스한 데이터를 사용하여 권한 줄이기 \(p. 409\)](#) 단원을 참조하십시오.

서비스에서 마지막으로 액세스한 데이터 검토

사전 정의된 비활성 기간(예: 90일)이 끝날 때까지 기다립니다. 그런 다음 사용자 또는 그룹에 대해 서비스에서 마지막으로 액세스한 데이터를 검토하여 사용자가 PowerUserExampleCorp 정책에서 허용하는 서비스에 마지막으로 액세스를 시도한 시기를 확인할 수 있습니다.

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 그룹을 선택한 다음 PowerUser 그룹 이름을 선택합니다.
3. 그룹 요약 페이지에서 액세스 관리자 탭을 선택합니다.

서비스에서 마지막으로 액세스한 데이터 테이블에는 그룹 멤버가 마지막으로 각 서비스에 액세스하려고 시도한 시간이 시간 순으로 표시됩니다. 이 테이블에는 정책에서 허용하는 서비스만 포함됩니다. 이 경우 PowerUserExampleCorp 정책은 모든 AWS 서비스에 대한 액세스를 허용합니다.

4. 테이블을 검토하고 그룹 멤버가 최근에 액세스한 서비스에 대한 목록을 만듭니다.

예를 들어, 지난 달 내에 팀이 Amazon EC2 및 Amazon S3 서비스에만 액세스했다고 가정합니다. 그러나 6개월 전, 팀에서 Amazon EC2 Auto Scaling 및 IAM에 액세스했습니다. EC2 Auto Scaling을 조사했지만 필요하지 않다고 결정했습니다. 또한 IAM을 사용하여 Amazon EC2가 S3 버킷의 데이터에 액세스

할 수 있는 역할을 생성했습니다. 따라서 Amazon EC2 및 Amazon S3 서비스만 액세스할 수 있도록 사용자의 권한을 다시 축소하기로 결정했습니다.

정책을 편집하여 권한 줄이기

서비스에서 마지막으로 액세스한 데이터를 검토한 후 사용자가 필요로 하는 서비스에만 액세스할 수 있도록 정책을 편집할 수 있습니다.

필요한 서비스에만 액세스할 수 있도록 데이터를 사용하려면

1. 왼쪽 탐색 창에서 정책을 선택한 후 정책 생성을 선택합니다.
2. 정책 편집을 선택한 후 JSON 탭을 선택합니다.
3. 원하는 서비스만 허용하도록 JSON 정책 문서를 편집합니다.

예를 들어, Allow 효과와 NotAction 요소를 포함하는 첫 번째 명령문을 편집하여 Amazon EC2 및 Amazon S3 작업만 허용하도록 합니다. 이 작업을 수행하려면 FullAccessToSomeServices ID가 있는 명령문으로 바꿉니다. 새 정책은 다음 예제 정책과 같습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "FullAccessToSomeServices",  
            "Effect": "Allow",  
            "Action": [  
                "ec2:*",  
                "s3:*"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iam:CreateServiceLinkedRole",  
                "iam>DeleteServiceLinkedRole",  
                "iam>ListRoles",  
                "organizations:DescribeOrganization"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

4. 특정 작업 및 리소스에 대한 정책 권한을 추가로 줄이려면 CloudTrail 이벤트 기록에서 이벤트를 확인합니다. 여기에서 사용자가 액세스한 특정 작업 및 리소스에 대한 자세한 정보를 볼 수 있습니다. 자세한 정보는 AWS CloudTrail 사용 설명서에서 [CloudTrail 콘솔에서 CloudTrail 이벤트 보기](#) 단원을 참조하십시오.

사용자의 계정 로그인 방법

IAM 사용자와 각 사용자의 암호를 생성한 후에는 해당 사용자가 여러분의 계정 ID 또는 별칭을 사용하거나, 여러분의 계정 ID가 포함된 사용자 지정 URL에서 AWS Management 콘솔에 로그인할 수 있습니다.

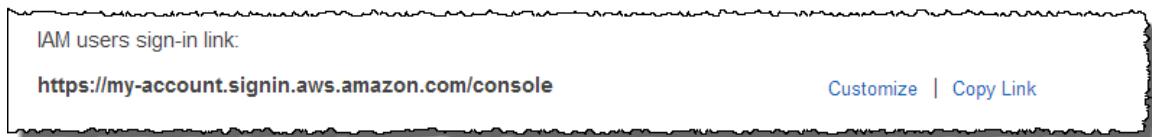
Note

회사에 기존의 자격 증명 시스템이 있는 경우, Single Sign-On(SSO) 옵션을 만드는 것이 좋습니다. SSO는 IAM 사용자 자격 증명이 없어도 AWS Management 콘솔에 액세스할 수 있는 권한을 사용자

에게 제공합니다. 또한 SSO를 사용하면 사용자가 조직의 사이트와 AWS에 따로 로그인하지 않아도 됩니다. 자세한 내용은 [연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하는 URL 생성\(사용자 지정 연동 브로커\) \(p. 188\)](#) 단원을 참조하십시오.

계정의 로그인 URL을 생성하기 전에 계정 별칭을 생성합니다. 그러면 URL에 계정 ID 대신 계정 이름이 포함됩니다. 자세한 내용은 [AWS 계정 ID 및 별칭 \(p. 55\)](#) 단원을 참조하십시오.

IAM 콘솔 대시보드에서 계정의 로그인 URL을 확인할 수 있습니다.



IAM 사용자에 대한 로그인 URL을 생성하려면 다음 패턴을 따르십시오.

`https://account-ID-or-alias.signin.aws.amazon.com/console`

IAM 사용자는 다음 엔드포인트에서 로그인한 다음 사용자 지정 URL을 사용하는 대신 계정 ID 또는 별칭을 수동으로 입력할 수도 있습니다.

`https://signin.aws.amazon.com/console`

콘솔 활동에 필요한 권한

계정 내 IAM 사용자는 사용자 또는 사용자가 속한 IAM 그룹에 첨부되는 정책에서 지정한 AWS 리소스에만 액세스할 수 있습니다. 콘솔에서 작업하려면 AWS 리소스 표시 및 생성과 같은 콘솔이 실행하는 작업을 실행할 허가를 받아야 합니다. 자세한 내용은 [액세스 관리 \(p. 304\)](#) 및 [IAM 자격 증명 기반 정책 예제 \(p. 341\)](#)을(를) 참조하십시오.

CloudTrail에 로그인 세부 정보 기록

CloudTrail에서 로그인 이벤트를 기록하도록 설정할 경우 CloudTrail에서 이벤트를 기록하는 방법을 이해해야 합니다.

- 사용자가 콘솔에 직접 로그인할 경우 글로벌 또는 리전 로그인 엔드포인트로 리디렉션됩니다. 이 리디렉션은 선택한 서비스 콘솔이 리전을 지원하는지 여부를 기준으로 합니다. 예를 들어, 메인 콘솔 홈 페이지가 리전을 지원하여 다음 URL에 로그인할 경우 "기본" 리전 로그인 엔드포인트 `https://us-east-1.signin.aws.amazon.com`으로 리디렉션됩니다.

`https://alias.signin.aws.amazon.com/console`

이렇게 되면 해당 리전의 로그에 리전 CloudTrail 로그 항목이 기록됩니다.

Amazon S3과 같은 일부 서비스를 위한 콘솔은 리전을 지원하지 않습니다. 다음 URL을 사용하여 해당 서비스에 로그인하는 경우 AWS가 사용자를 `https://signin.aws.amazon.com`의 글로벌 로그인 엔드포인트으로 리디렉션합니다.

`https://alias.signin.aws.amazon.com/console/s3`

그러면 글로벌 CloudTrail 로그 항목이 기록됩니다.

- 리전이 활성화된 메인 콘솔 홈 페이지에 로그인하면 특정 리전 로그인 엔드포인트를 수동으로 요청할 수 있습니다. 이렇게 하려면 다음 예제와 같은 URL을 사용합니다.

```
https://alias.signin.aws.amazon.com/console?region=ap-southeast-1
```

그리고 나면 AWS가 사용자를 ap-southeast-1 리전 로그인 엔드포인트로 리디렉션합니다. 그러면 해당 리전의 로그에 리전 CloudTrail 로그 항목이 기록됩니다.

CloudTrail 및 IAM에 대한 자세한 내용은 [AWS CloudTrail로 IAM 이벤트 로깅](#) 단원을 참조하십시오.

계정의 사용자에게 프로그래밍 방식의 액세스가 필요한 경우 각 사용자의 액세스 키 페어(액세스 키 ID 및 보안 액세스 키)를 생성할 수 있습니다. 자세한 내용은 [액세스 키 관리\(콘솔\) \(p. 90\)](#) 단원을 참조하십시오.

IAM 자습서

이 섹션은 IAM에서 수행하는 일반적인 작업에 대한 처음부터 끝까지의 완전한 절차를 일별하는 내용을 담고 있습니다. 그 절차들은 랩 유형의 환경에 맞게 고안되었고 예로 사용할 회사 이름, 사용자 이름 등이 있습니다. 그 목적은 일반적인 지침을 제공하는 것입니다. 조직의 환경이 지닌 고유한 측면에 대한 주의 깊은 검토 및 적용 없이 생산 환경에 바로 사용할 수 있도록 고안된 것이 아닙니다.

주제

- [자습서: Billing 콘솔에 대한 액세스 권한 위임 \(p. 25\)](#)
- [자습서: IAM 역할을 사용한 AWS 계정 간 액세스 권한 위임 \(p. 29\)](#)
- [자습서: 첫 번째 고객 관리형 정책 만들기 및 연결 \(p. 37\)](#)
- [자습서: 사용자들이 자신의 자격 증명 및 MFA 설정을 구성할 수 있도록 하기 \(p. 39\)](#)

자습서: Billing 콘솔에 대한 액세스 권한 위임

AWS 계정 소유자는 IAM 계정의 AWS Billing and Cost Management 데이터를 보거나 관리해야 하는 특정 AWS 사용자에게 액세스 권한을 위임할 수 있습니다. 다음 지침은 주 AWS 프로덕션 계정에 영향을 줄 염려 없이 결제 권한 구성을 실제 경험할 수 있는 사전 테스트된 시나리오를 설정할 수 있게 돕도록 고안되었습니다.

이 워크플로우는 네 가지 기본 단계로 이루어집니다.

1단계: AWS 테스트 계정에서 결제 데이터에 대한 액세스 권한 활성화 (p. 26)

단일 AWS 계정을 생성하는 경우 AWS 계정 소유자([AWS 계정 루트 사용자 \(p. 291\)](#))만 결제 정보를 보고 관리할 수 있습니다. IAM 사용자는 계정 소유자가 IAM 액세스를 활성화하고 사용자 또는 역할에 결제 작업을 제공하는 정책을 연결해야 결제 데이터에 액세스할 수 있습니다. 루트 사용자로 로그인해야 하는 다른 작업을 보려면 [계정 루트 사용자가 필요한 AWS 작업](#) 단원을 참조하십시오.

AWS Organizations를 사용하여 [멤버 계정을 생성하는 경우](#) 이 기능이 기본적으로 활성화됩니다.

2단계: 결제 데이터에 대한 권한을 부여하는 IAM 정책 생성 (p. 26)

계정에서 결제 액세스 권한을 활성화한 후에도 특정 IAM 사용자 또는 그룹에게 명시적으로 결제 데이터 액세스 권한을 부여해야 합니다. 이 액세스 권한은 고객 관리형 정책을 사용하여 부여합니다.

3단계: 그룹에 결제 정책 연결 (p. 27)

정책을 그룹에 연결할 경우 해당 그룹의 모든 멤버가 해당 정책과 관련된 액세스 권한의 전체 집합을 부여받습니다. 이 시나리오에서는 새 결제 정책을 결제 액세스 권한이 필요한 사용자만 포함하는 그룹에 연결합니다.

4단계: Billing 콘솔에 대한 사용자 액세스 권한 테스트 (p. 27)

핵심 과정을 완료했으므로 정책을 테스트할 수 있습니다. 테스트는 정책이 의도된 대로 동작하는지 확인합니다.

사전 조건

이 자습서를 사용해 테스트 AWS 계정을 생성합니다. 다음 테이블에 요약된 대로 이 계정에서 테스트 사용자 2명과 테스트 그룹 2개를 생성합니다. 나중에 4단계에서 로그인할 수 있도록 각 사용자에게 암호를 배정하십시오.

사용자 계정 생성	그룹 계정 생성 및 구성	
FinanceManager	BillingFullAccessGroup	FinanceManager
FinanceUser	BillingViewAccessGroup	FinanceUser

1단계: AWS 테스트 계정에서 결제 데이터에 대한 액세스 권한 활성화

테스트 계정에 로그인하여 결제 액세스를 활성화합니다. 프로덕션 환경에서 이 프로세스를 수행하는 방법에 대한 자세한 정보는 AWS Billing and Cost Management 사용 설명서의 [AWS 웹사이트에 대한 액세스 활성화](#) 단원을 참조하십시오.

Note

AWS Organizations를 사용하여 [멤버 계정을 생성하는 경우](#) 이 기능이 기본적으로 활성화됩니다.

AWS 테스트 계정에서 결제 데이터에 대한 액세스 권한을 활성화하려면

1. AWS 계정 이메일 주소와 암호를 사용하여 [AWS Management 콘솔](#)에 AWS 계정 루트 사용자 (p. 291)로 로그인합니다.
2. 탐색 표시줄에서 계정 이름을 선택한 다음 내 계정을 선택합니다.
3. 결제 정보에 대한 IAM 사용자 및 역할 액세스 옆에 있는 편집을 선택합니다.
4. IAM 액세스 활성화 확인란을 선택하고 업데이트를 선택합니다.
5. 콘솔에서 로그아웃하고 다음([2단계: 결제 데이터에 대한 권한을 부여하는 IAM 정책 생성 \(p. 26\)](#))을 진행합니다.

2단계: 결제 데이터에 대한 권한을 부여하는 IAM 정책 생성

다음 단계에서는 Billing and Cost Management 콘솔 내 페이지에 대한 보기 및 전체 액세스 권한을 모두 부여하는 사용자 지정 정책을 생성합니다. IAM 권한 정책에 대한 일반 정보는 [관리형 정책과 인라인 정책 \(p. 312\)](#) 단원을 참조하십시오.

결제 데이터에 대한 권한을 부여하는 IAM 정책을 생성하려면

1. 관리자 자격 증명을 가진 사용자로 AWS Management 콘솔에 로그인합니다. IAM 모범 사례를 준수하려면 루트 사용자 자격 증명을 사용하여 로그인하지 마십시오. 자세한 정보는 [개별 IAM 사용자 만들기 \(p. 44\)](#) 단원을 참조하십시오.
2. <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
3. 탐색 창에서 정책을 선택한 후 정책 생성을 선택합니다.
4. Visual editor(시각적 편집기) 탭에서 Choose a service(서비스 선택)을 선택하여 시작합니다. 그런 다음 결제를 선택합니다.
5. 다음 두 단계를 사용하여 두 가지 정책을 생성합니다.

모든 액세스

- a. 작업 선택>Select actions)을 선택한 다음 모든 작업(*) (All Actions (*)) 옆에 있는 확인란을 선택합니다. 이 정책에 대한 리소스 또는 조건을 선택할 필요가 없습니다.

- b. [Review policy]를 선택합니다.
- c. 검토 페이지에서 이름 옆에 **BillingFullAccess**를 입력한 다음 정책 생성을 선택하여 저장합니다.

읽기 전용 액세스

- a. [3 및 4 \(p. 26\)](#) 단계를 반복합니다.
- b. Select actions(작업 선택)을 선택한 다음 읽기 옆에 있는 확인란을 선택합니다. 이 정책에 대한 리소스 또는 조건을 선택할 필요가 없습니다.
- c. [Review policy]를 선택합니다.
- d. 검토 페이지의 이름에 **BillingViewAccess**를 입력합니다. 그런 다음 정책 생성을 선택하여 저장합니다.

Billing and Cost Management 콘솔에 대한 사용자 액세스 권한을 부여하는 IAM 정책에서 사용 가능한 각 권한에 대한 설명을 보려면 [결제 권한 설명](#) 단원을 참조하십시오.

3단계: 그룹에 결제 정책 연결

이제 사용자 지정 결제 정책이 준비되었으므로 앞서 만든 그룹에 정책을 연결할 수 있습니다. 정책을 사용자 또는 역할에 직접 연결할 수 있지만, (IAM 모범 사례에 따라) 그룹을 대신 사용하는 것이 좋습니다. 자세한 정보는 [그룹을 사용하여 IAM 사용자에게 권한 할당 \(p. 44\)](#) 단원을 참조하십시오.

그룹에 결제 정책을 연결하려면

1. 탐색 창에서 정책을 선택하여 AWS 계정에 사용 가능한 정책의 전체 목록을 표시합니다. 각 정책을 적절한 그룹에 연결하려면 다음 단계를 수행합니다.

모든 액세스

- a. 정책 검색 상자에 **BillingFullAccess**를 입력한 다음, 정책 이름 옆의 확인란을 선택합니다.
- b. [Policy actions]를 선택한 후 [Attach]를 선택합니다.
- c. 자격 증명(사용자, 그룹 및 역할) 검색 상자에 **BillingFullAccessGroup**를 입력하고 그룹 이름 옆의 확인란을 선택한 다음, 정책 연결을 선택합니다.

읽기 전용 액세스

- a. 정책 검색 상자에 **BillingViewAccess**를 입력한 다음, 정책 이름 옆의 확인란을 선택합니다.
- b. [Policy actions]를 선택한 후 [Attach]를 선택합니다.
- c. 자격 증명(사용자, 그룹 및 역할) 검색 상자에 **BillingViewAccessGroup**를 입력하고 그룹 이름 옆의 확인란을 선택한 다음, 정책 연결을 선택합니다.
2. 콘솔에서 로그아웃하고 다음([4단계: Billing 콘솔에 대한 사용자 액세스 권한 테스트 \(p. 27\)](#))을 진행합니다.

4단계: Billing 콘솔에 대한 사용자 액세스 권한 테스트

사용자 액세스 권한은 두 가지 방법으로 테스트할 수 있습니다. 이 자습서에서는 사용자가 어떤 경험을하게 되는지 볼 수 있도록 각 테스트 사용자로 로그인하여 액세스 권한을 테스트할 것을 권장합니다. 사용자 액세스 권한을 테스트하는 또 하나의 방법(선택 사항)은 [IAM 정책 시뮬레이터](#)를 사용하는 것입니다. 이러한 작업의 유효한 결과를 확인하는 다른 방법을 보려면 다음 단계를 사용하십시오.

선행하는 테스트 방법에 따라 다음 절차 중 하나를 선택하십시오. 첫 번째 절차에서는 두 테스트 계정으로 로그인하여 액세스 권한 차이를 확인합니다.

두 테스트 사용자 계정으로 로그인하여 결제 액세스 권한을 테스트하려면

1. AWS 계정 ID 또는 계정 별칭, IAM 사용자 이름, 암호를 사용하여 [IAM 콘솔](#)에 로그인합니다.

Note

사용자 편의를 위해 AWS 로그인 페이지는 브라우저 쿠키를 사용하여 IAM 사용자 이름 및 계정 정보를 기억합니다. 이전에 다른 사용자로 로그인한 경우, 페이지 하단 근처의 [Sign in to a different account]를 선택하여 기본 로그인 페이지로 돌아갑니다. 여기서 AWS 계정 ID 또는 계정 별칭을 입력하면 계정의 IAM 사용자 로그인 페이지로 리디렉션됩니다.

2. 각 사용자 환경을 비교할 수 있도록 아래 제공된 단계를 사용하여 각 계정으로 로그인합니다.

모든 액세스

- a. 사용자 FinanceManager로 AWS 계정에 로그인합니다.
- b. 탐색 모음에서 FinanceManager@<account alias or ID number>를 선택하고 대금 및 비용 관리를 선택합니다.
- c. 각 페이지를 탐색하면서 다양한 버튼을 선택하여 모든 수정 권한이 있는지 확인합니다.

읽기 전용 액세스

- a. 사용자 FinanceUser로 AWS 계정에 로그인합니다.
- b. 탐색 모음에서 FinanceUser@<account alias or ID number>를 선택하고 대금 및 비용 관리를 선택합니다.
- c. 각 페이지를 탐색해 봅니다. 비용, 보고서 및 결제 데이터는 아무 문제 없이 표시될 것입니다. 하지만 값을 수정하는 옵션을 선택할 경우 액세스 거부됨 메시지가 표시됩니다. 예를 들어 기본 설정 페이지에서 아무 확인란이나 선택하고 기본 설정 저장을 선택합니다. 콘솔이 해당 페이지를 수정하려면 ModifyBilling 권한이 필요하다는 메시지를 표시합니다.

다음 절차(선택 사항)는 IAM 정책 시뮬레이터를 사용하여 위임된 사용자의 결제 페이지에 대한 유효한 권한을 테스트하는 대체 방법을 보여줍니다.

IAM 정책 시뮬레이터에서 유효한 권한을 확인하여 결제 액세스 권한을 테스트하려면

1. <https://pollicysim.aws.amazon.com/>에서 IAM 정책 시뮬레이터를 엽니다. (AWS 로그인 전이라면 로그인 하라는 메시지가 먼저 표시됩니다.)
2. Users, Groups, and Roles(사용자, 그룹 및 역할) 아래에서 최근에 정책을 연결한 그룹의 멤버인 사용자를 하나 선택합니다.
3. Policy Simulator(정책 시뮬레이터) 아래에서 Select service(서비스 선택)을 선택하고 결제를 선택합니다.
4. Select actions(작업 선택) 옆에서 모두 선택을 선택합니다.
5. Run Simulation(시뮬레이션 실행)을 선택하고 해당 사용자의 권한을 모든 가능한 결제 관련 권한 옵션과 비교하여 올바른 권한이 적용되었는지 확인합니다.

관련 리소스

AWS Billing and Cost Management 사용 설명서에서 수록된 관련 내용은 다음 리소스 단원을 참조하십시오.

- [AWS 웹 사이트에 대한 액세스 활성화](#)
- [예제 4: AWS 서비스에 대한 모든 액세스 권한을 허용하되 대금 및 비용 관리 콘솔에 대한 IAM 사용자 액세스는 거부](#).
- [결제 권한 설명](#)

IAM 사용 설명서에 수록된 관련 내용은 다음 리소스 단원을 참조하십시오.

- 관리형 정책과 인라인 정책 (p. 312)
- AWS Management 콘솔에 대한 사용자 액세스 제어 (p. 54)
- IAM 그룹에 정책 연결 (p. 150)

요약

이제 Billing and Cost Management 콘솔에 대한 사용자 액세스 권한을 위임하는 데 필요한 단계를 모두 성공적으로 완료했습니다. 그 결과, 사용자가 Billing 콘솔에서 어떤 경험을 하게 될지 직접 확인했으며 이제 편의에 따라 프로덕션 환경에서 이 로직을 구현할 수 있습니다.

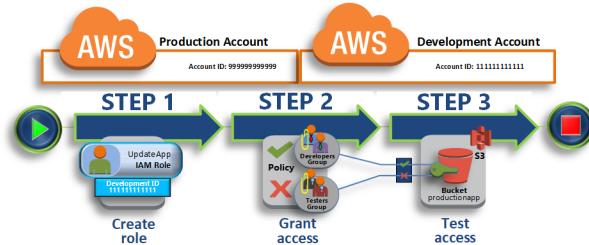
자습서: IAM 역할을 사용한 AWS 계정 간 액세스 권한 위임

이 자습서는 소유하고 있는 다른 AWS 계정(Production 및 Development)의 리소스에 역할을 사용하여 액세스 권한을 위임하는 방법에 대해 설명합니다. 한 계정의 리소스는 다른 계정의 사용자와 공유합니다. 이러한 방식으로 교차 계정 액세스를 설정하면 각 계정에 개별 IAM 사용자를 생성할 필요가 없습니다. 또한 사용자는 다른 AWS 계정의 리소스에 액세스하기 위해 한 계정에서 로그아웃하고 다른 계정에 로그인할 필요가 없습니다. 역할을 구성한 후에는 AWS Management 콘솔, AWS CLI 및 API에서 역할을 사용하는 방법에 대해서도 알아봅니다.

이 자습서에서는 Production 계정은 라이브 애플리케이션을 관리하는 곳이고, Development 계정은 개발자와 테스터들이 자유롭게 애플리케이션을 테스트할 수 있는 샌드박스라고 가정합니다. 각 계정의 애플리케이션 정보는 Amazon S3 버킷에 저장됩니다. Developers와 Testers, 두 IAM 그룹으로 구성된 Development 계정에서는 IAM 사용자를 관리합니다. 두 그룹의 사용자는 Development 계정에서 작업하면서 리소스에 액세스할 수 있는 권한을 갖습니다. 개발자는 종종 Production 계정의 라이브 애플리케이션을 업데이트해야 합니다. 이들 애플리케이션은 productionapp이라고 하는 Amazon S3 버킷에 저장되어 있습니다.

이번 자습서를 마치면 Development 계정(신뢰할 수 있는 계정)의 사용자가 Production 계정(신뢰하는 계정)의 productionapp 버킷에 액세스할 수 있는 역할 하나가 Production 계정에 생기게 됩니다. 그러면 개발자들이 AWS Management 콘솔에서 이 역할을 사용하여 Production 계정의 productionapp 버킷에 액세스할 수 있습니다. 또한 역할을 통해 제공되는 임시 자격 증명으로 API 호출을 인증함으로써 버킷에 액세스하는 것도 가능합니다. 하지만 테스터는 이 역할을 사용하지 못합니다.

이 워크플로우는 세 가지 기본 단계로 이루어집니다.



1단계 - 역할 만들기 (p. 30)

먼저 AWS Management 콘솔을 사용하여 Production 계정(ID 번호 999999999999)과 Development 계정(ID 번호 111111111111) 사이에 신뢰를 구성합니다. UpdateApp이라는 IAM 역할을 생성하여 시작합니다. 역할을 생성하였으면 Development 계정을 신뢰할 수 있는 엔터티로 정의한 다음 신뢰할 수 있는 계정의 사용자가 productionapp 버킷을 업데이트할 수 있는 권한 정책을 지정합니다.

2단계 - 역할에 액세스 권한 부여 (p. 32)

이 자습서 단계에서는 테스터들이 UpdateApp 역할에 액세스하지 못하도록 IAM 그룹 정책을 변경합니다. Testers 그룹은 이 시나리오에서 PowerUser 액세스 권한을 갖기 때문에 역할을 사용하지 못하도록 명시적으로 거부해야 합니다.

3단계 - 역할 전환을 통해 액세스 권한 테스트 (p. 34)

마지막으로, 개발자로서 UpdateApp 역할을 사용하여 Production 계정의 productionapp 버킷을 업데이트합니다. 이제 AWS 콘솔, AWS CLI 및 API를 통해 역할에 액세스할 수 있게 되었습니다.

사전 조건

이 자습서에서는 다음을 이미 완료했다고 가정합니다.

- Development 계정 및 Production 계정으로 사용할 수 있는 2개의 AWS 계정.
- Development 계정에서 다음과 같이 생성 및 구성된 사용자 및 그룹:

사용자	그룹	권한
David	개발자	두 사용자 모두 Development 계정에서 AWS Management 콘솔에 로그인하고 사용할 수 있습니다.
Theresa	테스터	

- Production 계정에서는 사용자 또는 그룹을 생성할 필요가 없습니다.
- Production 계정에서 생성된 Amazon S3 버킷. 이 자습서에서는 이 버킷을 ProductionApp이라고 부릅니다. 하지만 S3 버킷 이름은 전역에서 고유해야 하므로 다른 이름의 버킷을 사용해야 합니다.

1단계 - 역할 만들기

한 AWS 계정의 사용자가 다른 AWS 계정의 리소스에 액세스할 수 있도록 허용하려면 역할을 생성하여 액세스가 가능한 사용자와 전환 사용자에게 부여하는 권한을 정의해야 합니다.

자습서 1단계에서는 Production 계정에서 역할을 생성하고 Development 계정을 신뢰할 수 있는 엔터티로 지정합니다. 또한 역할 권한을 productionapp 버킷에 대한 읽기 및 쓰기 액세스 권한으로 제한합니다. 따라서 역할을 사용할 수 있는 권한을 받은 사용자는 누구나 productionapp 버킷에 대해 읽기나 쓰기가 가능합니다.

역할을 생성하려면 먼저 Development AWS 계정의 ID가 필요합니다. 계정 ID는 AWS 계정에 할당된 고유 식별자입니다.

Development AWS 계정 ID를 가져오는 방법

- Development 계정 관리자로 AWS Management 콘솔에 로그인한 후 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
- 탐색 모음에서 지원을 선택한 후 지원 센터를 선택합니다. 계정 번호는 지원 메뉴 바로 아래의 오른쪽 상단 모서리 부분에 있습니다. 12자리 숫자로 이루어진 것이 계정 ID입니다. 이 시나리오에서는 Development 계정 ID로 111111111111을 가정하여 사용하지만 실제로 테스트 환경에서 시나리오를 재현하는 경우에는 유효한 계정 ID를 사용해야 합니다.

Development 계정에서 사용할 수 있도록 Production 계정의 역할을 생성하는 방법

- Production 계정 관리자로 AWS Management 콘솔에 로그인한 후 IAM 콘솔을 엽니다.
- 역할을 만들기 전에 먼저 해당 역할에 필요한 권한을 정의하는 관리형 정책을 준비합니다. 차후 단계에서 이 정책을 해당 역할에 연결합니다.

productionapp 버킷에 대한 읽기 및 쓰기 액세스 권한을 설정하려고 합니다. AWS에 이미 몇 가지 Amazon S3 관리형 정책이 있지만 단일 Amazon S3 버킷에 대한 읽기 및 쓰기 액세스가 가능한 정책은 없습니다. 대신에 사용자가 직접 정책을 생성할 수 있습니다.

왼쪽 탐색 창에서 정책을 선택한 후 정책 생성을 선택합니다.

- JSON 탭을 선택하고 다음 JSON 정책 문서에서 텍스트를 복사합니다. 이 텍스트를 JSON 텍스트 상자에 붙여 넣어 리소스 ARN(`arn:aws:s3:::productionapp`)을 S3 버킷에 적합한 것으로 바꿉니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "s3>ListAllMyBuckets",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3>ListBucket",  
                "s3:GetBucketLocation"  
            ],  
            "Resource": "arn:aws:s3:::productionapp"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3.GetObject",  
                "s3:PutObject",  
                "s3>DeleteObject"  
            ],  
            "Resource": "arn:aws:s3:::productionapp/*"  
        }  
    ]  
}
```

`ListBucket`은 사용자가 productionapp 버킷에 저장되어 있는 객체를 볼 수 있는 권한입니다.
`GetObject`, `PutObject` 및 `DeleteObject`는 사용자가 productionapp 버킷에 저장되어 있는 콘텐츠를 각각 보거나, 업데이트하거나, 삭제할 수 있는 권한입니다.

- 작업이 완료되면 [Review policy]를 선택합니다. 정책 검토(p. 382)가 모든 구문 오류를 보고합니다.

Note

언제든지 Visual editor(시각적 편집기) 및 JSON 탭을 전환할 수 있습니다. 그러나 변경을 수행하거나 Visual editor(시각적 편집기) 탭에서 정책 검토를 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 내용은 정책 재구성(p. 455) 단원을 참조 하십시오.

- 검토 페이지에서 정책 이름에 `read-write-app-bucket`을 입력합니다. 정책 요약을 검토하여 정책이 부여한 권한을 확인한 다음 정책 생성을 선택하여 작업을 저장합니다.

새로운 정책이 관리형 정책 목록에 나타납니다.

- 왼쪽 탐색 창에서 역할을 선택한 후 역할 만들기를 선택합니다.
- 다른 AWS 계정 역할 유형을 선택합니다.
- 계정 ID에 Development 계정 ID를 입력합니다.

이 자습서에서는 Development 계정 ID의 예로 `111111111111`을 사용합니다. 하지만 실제로는 유효한 계정 ID를 사용해야 합니다. `111111111111` 같이 잘못된 계정 ID를 사용할 경우, IAM에서 새로운 역할을 생성할 수 없습니다.

이 연습에서는 외부 ID를 요구하거나, 사용자가 역할을 위임하기 위해 멀티 팩터 인증(MFA)을 요구할 필요가 없습니다. 따라서 이러한 옵션을 선택하지 않은 상태로 두십시오. 자세한 정보는 [AWS에서 멀티 팩터 인증\(MFA\) 사용하기 \(p. 96\)](#) 단원을 참조하십시오.

9. 다음: 권한을 클릭하여 역할과 연결될 권한을 설정합니다.
10. 앞에서 생성한 정책 옆의 상자를 선택합니다.

도움말

필터에서 Customer managed(고객 관리형)을 선택하여 생성한 정책만 포함하도록 목록을 필터링합니다. 이렇게 하면 AWS에서 생성한 정책이 표시되지 않아서 찾고자 하는 정책을 쉽게 찾을 수 있습니다.

그러고 나서 다음: 태그 지정을 선택합니다.

11. (선택 사항) 태그를 키-값 페어로 연결하여 메타데이터를 사용자에게 추가합니다. IAM에서의 태그 사용에 대한 자세한 정보는 [IAM 엔터티 태그 지정 \(p. 259\)](#) 단원을 참조하십시오.
12. 다음: 검토를 선택하고 역할 이름으로 **UpdateApp**을 입력합니다.
13. (선택 사항) [Role description]에 새 역할에 대한 설명을 입력합니다.
14. 역할을 검토한 후 역할 만들기를 선택합니다.

역할 목록에 UpdateApp 역할이 표시됩니다.

이제 역할의 고유 식별자인 Amazon 리소스 이름(ARN)을 가져와야 합니다. Developers 및 Testers 그룹의 정책을 변경하면서 권한을 부여하거나 거부하려면 역할의 ARN을 지정해야 합니다.

UpdateApp 역할의 ARN을 가져오려면

1. IAM 콘솔의 탐색 창에서 [Roles]를 선택합니다.
2. 역할 목록에서 UpdateApp 역할을 선택합니다.
3. 세부 정보 창의 요약 섹션에서 역할 ARN 값을 복사합니다.

Production 계정 ID는 999999999999입니다. 따라서 역할 ARN은 `arn:aws:iam::999999999999:role/UpdateApp`이 됩니다. 하지만 사용자 환경에서는 'Production' 계정에 실제 AWS 계정 ID를 입력해야 합니다.

이번 연습에서는 Production 계정에서 Development 계정을 신뢰할 수 있는 보안 주체로 식별하는 역할을 생성하여 Production 계정과 Development 계정 사이에 신뢰를 구성했습니다. 그 밖에도 UpdateApp 역할 전환 사용자의 권한까지 정의했습니다.

다음에는 그룹 권한을 변경하는 방법에 대해 알아보겠습니다.

2단계 - 역할에 액세스 권한 부여

여기에서는 Testers 그룹 멤버나 Developers 그룹 멤버 모두 Development 계정의 애플리케이션을 자유롭게 테스트할 수 있는 권한을 갖습니다. 하지만 역할 전환에 필요한 권한을 추가하려면 몇 단계를 거쳐야 합니다.

UpdateApp 역할로 전환할 수 있도록 Developers 그룹을 변경하는 방법

1. Development 계정에 관리자로 로그인한 다음 IAM 콘솔을 엽니다.
2. 그룹을 선택한 후 Developers(개발자)를 선택합니다.
3. 권한 탭을 선택하고 Inline Policies(인라인 정책) 섹션을 확장한 후 Create Group Policy(그룹 정책 생성)을 선택합니다. 인라인 정책이 아직 없으면 이 버튼이 표시되지 않습니다. 이 경우 "To create one, click here" 끝의 링크를 선택합니다.

4. 사용자 지정 정책을 선택한 후 선택 버튼을 선택합니다.
5. **allow-assume-S3-role-in-production**과 같이 정책 이름을 입력합니다.
6. 아래 정책 문을 추가하여 Production 계정에서 AssumeRole 역할의 UpdateApp 작업을 허용합니다. 이 때 Resource 요소에서 **PRODUCTION-ACCOUNT-ID**는 Production 계정의 실제 AWS 계정 ID로 변경해야 합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "sts:AssumeRole",  
            "Resource": "arn:aws:iam::PRODUCTION-ACCOUNT-ID:role/UpdateApp"  
        }  
    ]  
}
```

Allow는 Production 계정에서 UpdateApp 역할에 대한 Developers 그룹의 액세스를 명시적으로 허용하는 값입니다. 따라서 개발자라면 누구나 이 역할에 액세스할 수 있습니다.

7. Apply Policy(정책 적용)을 선택하여 정책을 Developer 그룹에 추가합니다.

대부분 환경에서 다음과 같은 절차는 필요하지 않습니다. 하지만 Power User 권한을 사용하는 경우에는 역할 전환을 할 수 있는 그룹이 있을 수도 있습니다. 아래는 Testers 그룹에게 역할을 위임할 수 없도록 "Deny" 권한을 추가하는 방법을 나타낸 절차입니다. 이 절차가 필요 없는 환경인 경우 추가하지 않는 것이 좋습니다. '거부' 권한은 전체 권한 구조를 관리하고 이해하기가 더 복잡하게 만듭니다. "Deny" 권한은 더 좋은 방법이 없을 때만 사용하십시오.

UpdateApp 역할 위임 권한을 거부하도록 Testers 그룹을 변경하는 방법

1. 그룹을 선택한 후 Testers(테스터)를 선택합니다.
2. 권한 탭을 선택하고 Inline Policies(인라인 정책) 섹션을 확장한 후 Create Group Policy(그룹 정책 생성)을 선택합니다.
3. 사용자 지정 정책을 선택한 후 선택 버튼을 선택합니다.
4. **deny-assume-S3-role-in-production**과 같이 정책 이름을 입력합니다.
5. 다음 정책을 추가하여 AssumeRole 역할의 UpdateApp 작업을 거부합니다. 이때 Resource 요소에서 **PRODUCTION-ACCOUNT-ID**는 Production 계정의 실제 AWS 계정 ID로 변경해야 합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Action": "sts:AssumeRole",  
            "Resource": "arn:aws:iam::PRODUCTION-ACCOUNT-ID:role/UpdateApp"  
        }  
    ]  
}
```

Deny는 Production 계정에서 UpdateApp 역할에 대한 Testers 그룹의 액세스를 명시적으로 거부하는 값입니다. 따라서 이 역할에 액세스하려는 테스터에게는 모두 액세스 거부 메시지가 표시됩니다.

6. Apply Policy(정책 적용)을 선택한 후 정책을 Tester 그룹에 추가합니다.

Developers 그룹은 이제 Production 계정에서 UpdateApp 역할을 사용할 수 있는 권한이 생겼습니다. 그리고 Testers 그룹은 UpdateApp 역할을 사용하지 못합니다.

이제 다음에는 개발자인 David가 AWS Management 콘솔, AWS CLI 명령 및 AssumeRole API 호출을 사용하여 Production 계정의 productionapp 버킷에 액세스할 수 있는 방법에 대해 살펴보겠습니다.

3단계 - 역할 전환을 통해 액세스 권한 테스트

이 자습서의 첫 두 단계를 완료하면 Production 계정에 리소스에 대한 액세스 권한을 부여하는 역할이 생기게 됩니다. 또한 해당 역할을 사용할 수 있는 사용자가 속한 그룹이 하나 Development 계정에 생기게 됩니다. 이제 역할을 사용할 준비가 되었습니다. 이 단계에서는 AWS Management 콘솔, AWS CLI 및 AWS API에서 해당 역할로 전환을 테스트하는 방법을 설명합니다.

Important

IAM 사용자 또는 연합된 사용자로 로그인할 때만 역할을 전환할 수 있습니다. 또한 Amazon EC2 인스턴스를 시작하여 애플리케이션을 실행하는 경우 애플리케이션은 인스턴스 프로파일을 통해 역할을 부여할 수 있습니다. AWS 계정 루트 사용자로 로그인되어 있을 때는 역할을 바꿀 수 없습니다.

역할 전환(콘솔)

David가 AWS Management 콘솔의 Production 환경에서 작업해야 할 경우 Switch Role(역할 전환)을 사용하면 됩니다. David가 계정 ID나 별칭 및 역할 이름을 지정하면 David의 권한이 해당 역할에 허용되는 권한으로 즉시 전환됩니다. 그런 다음 David는 콘솔을 사용하여 `productionapp` 버킷으로 작업할 수 있지만 Production의 다른 리소스로는 작업할 수 없습니다. David가 이 역할을 사용하는 동안에는 Development 계정에서 파워 유저 권한도 사용할 수 없습니다. 한 번에 하나의 권한 집합만 적용할 수 있기 때문입니다.

Important

AWS Management 콘솔을 사용하여 역할을 전환하려면 계정에서 `ExternalId`를 요구하지 않아야 합니다. 계정에 대한 액세스 권한을 제3자에게 부여할 경우 권한 정책상 Condition 요소에 `ExternalId`가 필요하면, 제3자가 AWS API나 명령줄 도구를 사용해서만 계정에 액세스할 수 있습니다. 콘솔은 `ExternalId`에 대한 값을 제공할 수 없기 때문에 사용할 수 없습니다. 이 시나리오에 대한 자세한 정보는 [AWS 리소스에 대한 액세스를 타사에 부여할 때 외부 ID를 사용하는 방법](#)(p. 206)와 [AWS Management 콘솔 보안 블로그](#)의 AWS에 대한 교차 계정 액세스를 가능하게 하는 방법을 참조하십시오.

David는 다음과 같은 두 가지 방법으로 Switch Role(역할 전환) 페이지를 시작할 수 있습니다.

- David가 관리자로부터 미리 정의된 [Switch Role] 구성을 가리키는 링크를 받습니다. 이 링크는 역할 생성 마법사의 마지막 페이지 또는 교차 계정 역할의 Role Summary(역할 요약) 페이지에서 관리자에게 제공됩니다. 이 링크를 선택하면 계정 ID 및 역할 이름 필드에 이미 정보가 채워진 Switch Role(역할 전환) 페이지가 David에게 표시됩니다. David는 Switch Role(역할 전환) 버튼을 선택하기만 하면 됩니다.
- 관리자가 이메일로 링크를 보내는 대신 계정 ID 번호 및 역할 이름 값을 보냅니다. David는 역할을 전환하기 위해 수동으로 이 정보를 입력해야 합니다. 다음 절차에 이 내용이 잘 설명되어 있습니다.

역할을 위임하려면

1. David가 Development 그룹에 있는 일반 사용자로 AWS 콘솔에 로그인합니다.
2. 그는 관리자가 이메일로 보내준 링크를 선택합니다. 계정 ID나 별칭 및 역할 이름 정보가 이미 채워진 Switch Role(역할 전환) 페이지가 David에게 표시됩니다.

—또는—

그는 탐색 모음에서 자신의 이름(자격 증명 메뉴)을 선택한 후 Switch Role(역할 전환)을 선택합니다.

David가 이 방법으로 처음 Switch Role(역할 전환) 페이지 액세스를 시도하는 것이라면 첫 실행 Switch Role(역할 전환) 페이지가 표시됩니다. 이 페이지에는 역할 전환을 통해 사용자가 여러 AWS 계정의 리소스를 관리할 수 있는 방법에 대한 추가 정보가 제공됩니다. 이 절차의 나머지 부분을 완료하려면 David가 이 페이지에서 Switch Role(역할 전환) 버튼을 선택해야 합니다.

3. 다음으로, 해당 역할에 액세스하기 위해 David는 수동으로 Production 계정 ID 번호(999999999999)와 역할 이름(UpdateApp)을 입력해야 합니다.

또한 현재 활성 상태인 역할 및 연결된 권한을 잘 알 수 있도록 David는 Display Name(표시 이름) 텍스트 상자에 PRODUCTION을 입력하고 빨간색 옵션을 선택한 다음 Switch Role(역할 전환)을 선택합니다.

4. 이제 David는 Amazon S3 콘솔을 사용하여 Amazon S3 버킷으로 작업하거나 updateApp 역할에 권한이 있는 다른 모든 리소스로 작업할 수 있습니다.
5. David는 해야 할 일을 마친 뒤 원래의 권한으로 돌아갈 수 있습니다. 이를 위해 그는 탐색 모음에서 프로덕션이라고 표시된 역할 이름을 선택하고 Back to David @ 111111111111(David @ 111111111111로 돌아가기)을 선택합니다.
6. 다음에 David가 역할을 전환하려고 탐색 모음에서 [Identity] 메뉴를 선택하면 지난 번에 사용한 PRODUCTION 항목이 표시되는 것을 볼 수 있습니다. 계정 ID와 역할 이름을 다시 입력할 필요 없이 해당 항목을 선택하기만 하면 즉시 역할이 전환됩니다.

역할 전환(AWS CLI)

David가 명령줄에서 프로덕션 환경으로 작업해야 할 경우 [AWS CLI](#)를 사용하면 됩니다. David는 aws sts assume-role 명령을 실행하고 역할 ARN을 전달하여 해당 역할에 대한 임시 보안 자격 증명을 얻습니다. 그런 다음 후속 AWS CLI 명령이 해당 역할의 권한을 사용하여 작동하도록 환경 변수에서 해당 자격 증명을 구성합니다. 한 번에 한 가지 권한 세트만 적용될 수 있기 때문에 David가 해당 역할을 사용하는 동안에는 Development 계정의 파워 유저 권한을 사용할 수 없습니다.

모든 액세스 키와 토큰은 예제일 뿐이며 표시된 대로 사용할 수 없습니다. 라이브 환경의 적절한 값으로 바꾸십시오.

역할을 위임하려면

1. David가 명령 프롬프트 창을 열고 다음 명령을 실행하여 AWS CLI 클라이언트가 작동하는지 확인합니다.

```
aws help
```

Note

David의 기본 환경에는 David 명령으로 만든 기본 프로필의 aws configure 사용자 자격 증명이 사용됩니다. 자세한 내용은 AWS Command Line Interface 사용 설명서의 [AWS Command Line Interface 구성](#)을 참조하십시오.

2. David가 Production 계정의 UpdateApp 역할로 전환하기 위해 다음 명령을 실행하여 역할 전환 프로세스를 시작합니다. David는 해당 역할을 만든 관리자에게서 역할 ARN을 받았습니다. 이 명령을 실행하면서 세션 이름도 제공해야 합니다. 원하는 아무 텍스트나 선택할 수 있습니다.

```
aws sts assume-role --role-arn "arn:aws:iam::999999999999:role/UpdateApp" --role-session-name "David-ProdUpdate"
```

다음 내용이 출력됩니다.

```
{  
    "Credentials": {  
        "SecretAccessKey": "wJalrXutnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",  
        "SessionToken": "AQoDYXdzEGcaEXAMPLE2gsYULo  
+Im5ZEXAMLeEYjs1M2FUIgIJx9tQqNMBEXAMPLE  
CvSRyhOFW7jEXAMPLEw+vE/7s1HRpXviG7b+qYf4nD00EXAMPLEmj4wxS04L/  
uZEXAMPLEcihzFB5lTYlto9dyBgSDy  
EXAMPLE9/  
g7QRUhZp4bqbEXAMPLENwGPyOj59pFA41NKC1kVgkREXAMPLEj1zxQ7y52gekeVEXAMPLEDiB9ST3Uuysg  
sKdEXAMPLE1TVastU1A0SKFEXAMPLEiywCC/Cs8EXAMPLEpZgOs+6hz4AP4KEXAMPLERbASP  
+4eZScEXAMPLEsnf87e  
NhYDHq6ikBQ==",  
    }  
}
```

```
        "Expiration": "2014-12-11T23:08:07Z",
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"
    }
```

3. 출력의 [Credentials] 섹션에 David에게 필요한 세 가지 항목이 표시됩니다.

- AccessKeyId
- SecretAccessKey
- SessionToken

David는 후속 호출 시 이러한 파라미터를 사용하도록 AWS CLI 환경을 구성해야 합니다. 자격 증명을 구성하는 다양한 방법에 대한 자세한 정보는 [AWS Command Line Interface 구성](#)을 참조하십시오. aws configure 명령은 세션 토큰 캡처를 지원하지 않기 때문에 사용할 수 없습니다. 하지만 구성 파일에 정보를 수동으로 입력할 수 있습니다. 이는 비교적 만료 시간이 짧은 임시 자격 증명이기 때문에 현재 명령 줄 세션의 환경에 추가하는 것이 가장 쉽습니다.

4. 세 값을 환경에 추가하기 위해 David는 이전 단계의 출력을 잘라내어 다음 명령에 붙여 넣습니다. 세션 토큰 출력의 줄 바꿈 문제를 해결하기 위해 출력을 잘라내어 간단한 텍스트 편집기에 붙여 넣는 경우가 있습니다. 여기서는 명확성을 위해 줄을 바꾸어 표시했지만 긴 문자열 하나로 추가해야 합니다.

Note

다음 예제는 Windows 환경에 표시된 명령을 나타내며, 여기서 "set"은 환경 변수를 생성하라는 명령입니다. Linux 또는 Mac 컴퓨터에서는 "export" 명령으로 대신할 수 있습니다. 예제의 나머지 부분은 세 환경에서 모두 유효합니다.

```
set AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
set AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRficyEXAMPLEKEY
set AWS_SESSION_TOKEN=AQoDYXdzEGcaEXAMPLE2gsYULo
+Im5ZEXAMLeYjs1M2FUIgIJx9tQqNMBeAMPLECvS
Ryh0FW7jEXAMLeW+vE/7s1HRpXviG7b+qYf4nD00EXAMPLEmj4wxS04L/
uZEXAMLeCihzFB51TYLto9dyBgSDyEXA
MPLEKEY9/
g7QRUhZp4bqbEXAMLeNwGPyoj59pFA41NKC1kVgkREXAMPLEj1zxQ7y52gekeVEXAMLeDiB9ST3UusKd
EXAMLe1TVastU1A0SKFEXAMLeiywCC/Cs8EXAMLePzgOs+6hz4AP4KEXAMLeRbASP
+4eZSceXAMLeNhYkxiHen
DHq6ikBQ==
```

이 시점에서 모든 후속 명령은 해당 자격 증명으로 식별되는 역할의 권한에 따라 실행됩니다. David의 경우 UpdateApp 역할입니다.

5. 명령을 실행하여 Production 계정의 리소스에 액세스합니다. 이 예제에서 David는 단순히 다음 명령을 사용하여 S3 버킷의 콘텐츠를 나열합니다.

```
aws s3 ls s3://productionapp
```

Amazon S3 버킷 이름은 범용 고유 이름이기 때문에 해당 버킷을 소유하는 계정 ID를 지정할 필요가 없습니다. 다른 AWS 서비스의 리소스에 액세스하려면 해당 서비스의 AWS CLI 설명서에서 해당 리소스를 참조하는 데 필요한 명령과 구문을 참조하십시오.

AssumeRole(AWS API) 사용

David가 코드에서 Production 계정을 업데이트해야 하는 경우 AssumeRole을 호출하여 UpdateApp 역할을 맡습니다. 이 호출로 인해 Production 계정에서 David가 productionapp 버킷에 액세스하기 위해 사용할 수 있는 임시 자격 증명이 반환됩니다. David는 해당 자격 증명을 사용하여 productionapp 버킷을 업데이트하는 API 호출을 실행할 수 있습니다. 그러나 David는 Development 계정의 파워 유저 권한이 있더라도 Production 계정의 다른 리소스에 액세스하는 API 호출을 실행할 수 없습니다.

역할을 위임하려면

1. David가 애플리케이션의 일부로 AssumeRole을 호출합니다. David는 UpdateApp ARN: arn:aws:iam::999999999999:role/UpdateApp를 지정해야 합니다.
AssumeRole 호출의 응답에는 AccessKeyId, SecretAccessKey 및 자격 증명이 만료되어 새 자격 증명을 요청해야 하는 시간을 나타내는 Expiration 시간과 함께 임시 자격 증명이 포함됩니다.
2. David가 임시 자격 증명을 사용하여 s3:PutObject 버킷을 업데이트하는 productionapp 호출을 실행합니다. 이때 자격 증명을 AuthParams 파라미터로 API 호출에 전달합니다. 임시 역할 자격 증명에는 productionapp 버킷에 대한 읽기 및 쓰기 권한만 있기 때문에 Production 계정의 다른 모든 작업은 거부됩니다.

코드 샘플(Python 사용)은 [IAM 역할\(AWS API\)로 전환하기 \(p. 238\)](#) 단원을 참조하십시오.

관련 리소스

- IAM 사용자 및 그룹에 대한 자세한 정보는 [자격 증명\(사용자, 그룹, 및 역할\) \(p. 61\)](#) 단원을 참조하십시오.
- Amazon S3 버킷 생성에 대한 자세한 정보는 Amazon Simple Storage Service 시작 안내서에서 [버킷 생성](#) 단원을 참조하십시오.

요약

교차 계정 API 액세스 자습서를 완료했습니다. 다른 계정과 신뢰 관계를 설정하기 위한 역할을 만들고 신뢰할 수 있는 주체가 수행할 수 있는 작업을 정의했습니다. 그런 다음 해당 역할에 액세스할 수 있는 IAM 사용자를 제어하는 그룹 정책을 수정했습니다. 그 결과 Development 계정의 개발자가 임시 자격 증명을 사용하여 Production 계정에서 productionapp 버킷을 업데이트할 수 있습니다.

자습서: 첫 번째 고객 관리형 정책 만들기 및 연결

이 자습서에서는 AWS Management 콘솔을 사용하여 [고객 관리형 정책 \(p. 313\)](#)을 만든 다음 이 정책을 IAM 계정의 AWS 사용자에게 연결합니다. 여기서 생성하는 정책은 IAM 테스트 사용자가 읽기 전용 권한으로 AWS Management 콘솔에 직업 로그인할 수 있도록 허용합니다.

이 워크플로우는 세 가지 기본 단계로 이루어집니다.

1단계: 정책 만들기 (p. 38)

기본적으로 IAM 사용자에게는 아무런 권한이 없습니다. 관리자가 허용하지 않는 한, 이들 사용자가 AWS Management Console에 액세스하거나 그 안에서 데이터를 관리할 수 없습니다. 이 단계에서는 연결된 사용자가 콘솔에 로그인할 수 있도록 허용하는 고객 관리형 정책을 생성합니다.

2단계: 정책 연결 (p. 38)

정책을 사용자에 연결할 경우 이 사용자는 해당 정책과 관련된 액세스 권한을 모두 상속받습니다. 이 단계에서는 새 정책을 테스트 사용자 계정에 연결합니다.

3단계: 사용자 액세스 테스트 (p. 39)

정책을 연결하고 나면 해당 사용자로 로그인하여 정책을 테스트할 수 있습니다.

사전 조건

이 자습서의 단계를 수행하려면 다음이 준비되어 있어야 합니다.

- 관리 권한을 가진 IAM 사용자로 로그인할 수 있는 AWS 계정.
- 다음과 같이 할당된 권한 또는 그룹 멤버십이 없는 테스트 IAM 사용자.

사용자 이름	그룹	권한
PolicyUser	<없음>	<없음>

1단계: 정책 만들기

이 단계에서는 연결된 사용자가 IAM 데이터에 대한 읽기 전용 권한으로 AWS Management 콘솔에 로그인할 수 있도록 허용하는 고객 관리형 정책을 생성합니다.

테스트 사용자에 대한 정책을 생성하려면

- <https://console.aws.amazon.com/iam/>에서 관리자 권한을 가진 사용자로 IAM 콘솔에 로그인합니다.
- 탐색 창에서 정책을 선택합니다.
- 콘텐츠 창에서 정책 생성을 선택합니다.
- JSON 탭을 선택하고 다음 JSON 정책 문서에서 텍스트를 복사합니다. 이 텍스트를 JSON 텍스트 상자에 붙여 넣습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [ {  
        "Effect": "Allow",  
        "Action": [  
            "iam:GenerateCredentialReport",  
            "iam:Get*",  
            "iam>List*"  
        ],  
        "Resource": "*"  
    } ]  
}
```

- 작업이 완료되면 [Review policy]를 선택합니다. [정책 검사기 \(p. 382\)](#)가 모든 구문 오류를 보고합니다.

Note

언제든지 Visual editor(시각적 편집기) 및 JSON 탭을 전환할 수 있습니다. 그러나 변경을 수행하거나 Visual editor(시각적 편집기) 탭에서 정책 검토를 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 내용은 [정책 재구성 \(p. 455\)](#) 단원을 참조 하십시오.

- 검토 페이지에서 정책 이름에 **UsersReadOnlyAccessToIAMConsole**을 입력합니다. 정책 요약을 검토하여 정책이 부여한 권한을 확인한 다음 정책 생성을 선택하여 작업을 저장합니다.

새로운 정책이 관리형 정책 목록에 나타나며 연결 준비가 완료됩니다.

2단계: 정책 연결

다음에는 방금 생성한 테스트 IAM 사용자에 정책을 연결합니다.

테스트 사용자에 정책을 연결하려면

- IAM 콘솔의 탐색 창에서 정책을 선택합니다.
- 정책 목록의 맨 위의 검색 상자에 **UsersReadOnlyAccessToIAMConsole**을 입력하기 시작하여 목록에서 이 정책이 보이면 UsersReadOnlyAccessToIAMConsole 옆의 확인란을 선택합니다.

3. Policy actions(정책 작업) 버튼을 선택한 후 연결을 선택합니다.
4. 필터에서 사용자를 선택합니다.
5. 검색 상자에 **PolicyUser**를 입력하기 시작하여 목록에 이 사용자가 보이면 사용자 옆의 확인란을 선택합니다.
6. Attach Policy(정책 연결)를 선택합니다.

이제 IAM 테스트 사용자에 정책을 연결했습니다. 즉, 이 사용자가 IAM 콘솔에 읽기 전용으로 액세스할 수 있습니다.

3단계: 사용자 액세스 테스트

이 자습서에서는 결과를 관찰하고 사용자가 어떤 경험을 하게 되는지 볼 수 있도록 테스트 사용자로 로그인하여 액세스 권한을 테스트할 것을 권장합니다.

테스트 사용자 계정으로 로그인하여 액세스 권한을 테스트하려면

1. PolicyUser 테스트 사용자로 <https://console.aws.amazon.com/>에 있는 IAM 콘솔에 로그인합니다.
2. 콘솔에서 각 페이지를 탐색하고 새 사용자 또는 그룹을 생성해 봅니다. PolicyUser는 데이터를 표시할 수는 있지만 IAM 데이터를 생성하거나 수정할 수는 없습니다.

관련 리소스

IAM 사용 설명서에 수록된 관련 내용은 다음 리소스를 참조하십시오.

- 관리형 정책과 인라인 정책 (p. 312)
- AWS Management 콘솔에 대한 사용자 액세스 제어 (p. 54)
- 개별 IAM 사용자 만들기 (p. 44)

요약

이제 고객 관리형 정책을 생성 및 연결하는 데 필요한 모든 단계를 성공적으로 완료했습니다. 그 결과, 테스트 계정을 사용하여 IAM 콘솔에 로그인하고 사용자가 어떤 경험을 하게 될지 직접 확인했습니다.

자습서: 사용자들이 자신의 자격 증명 및 MFA 설정을 구성할 수 있도록 하기

사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 멀티 팩터 인증(MFA) 디바이스와 자격 증명을 스스로 관리하도록 할 수 있습니다. 사용자 수가 적은 경우 AWS Management 콘솔을 사용해 이들에 대한 자격 증명(액세스 키, 암호, 서명 인증서 및 SSH 퍼블릭 키)과 MFA 디바이스를 구성할 수 있지만, 하지만 사용자 수가 증가함에 따라 이 작업은 곧 많은 시간이 소모되는 작업이 될 수 있습니다. 보안 모범 사례에 따라 사용자는 정기적으로 암호를 변경하고 액세스 키를 교체해야 합니다. 또한 사용자는 필요 없는 자격 증명을 삭제하거나 비활성화해야 합니다. 중요한 작업에는 MFA를 사용하는 것이 좋습니다. 이 자습서에서는 관리자에게 부담을 주지 않으면서 이러한 모범 사례를 활성화하는 방법을 보여 줍니다.

이 자습서에서는 사용자가 MFA를 사용하여 로그인하는 경우에만 AWS 서비스에 액세스할 수 있도록 허용하는 방법을 보여 줍니다. MFA 디바이스에 로그인하지 않으면 사용자가 다른 서비스에 액세스할 수 없습니다.

이 워크플로우는 세 가지 기본 단계로 이루어집니다.

1단계: MFA 로그인을 강제할 정책 생성 (p. 40)

사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 자격 증명을 변경하고 MFA 디바이스를 관리할 수 있도록 허용하는 몇 가지 IAM 작업을 제외하고 모든 작업을 금지하는 고객 관리형 정책을 생성합니다. 해당 페이지에 액세스하는 방법에 대한 자세한 내용

2단계: 테스트 그룹에 정책 연결하기 (p. 41)

멤버가 MFA로 로그인한 경우 해당 멤버에게 모든 Amazon EC2 작업의 전체 액세스 권한을 부여한 그룹을 생성합니다. 이러한 그룹을 생성하려면 `AmazonEC2FullAccess`라는 AWS 관리형 정책과 1단계에서 생성한 고객 관리형 정책을 모두 연결합니다.

3단계: 사용자 액세스 테스트 (p. 41)

테스트 사용자로 로그인하여 사용자가 MFA 디바이스를 생성한 후 그 디바이스를 사용해 로그인할 때까지 Amazon EC2에 대한 액세스가 차단되는지 확인합니다.

사전 조건

이 자습서의 단계를 수행하려면 다음이 준비되어 있어야 합니다.

- 관리 권한을 가진 IAM 사용자로 로그인할 수 있는 AWS 계정.
- 1단계에서 정책에 입력한 계정 ID 번호.

계정 ID 번호를 찾으려면 페이지 상단의 탐색 표시줄에서 지원을 선택한 후 지원 센터를 선택합니다. 이 페이지의 지원 메뉴에서 계정 ID를 찾을 수 있습니다.

- [가상\(소프트웨어 기반\) MFA 디바이스 \(p. 99\)](#), [U2F 보안 키 \(p. 102\)](#), 또는 [하드웨어 기반 MFA 디바이스 \(p. 107\)](#).
- 다음과 같은 그룹의 구성원인 테스트 IAM 사용자:

사용자 계정 생성		그룹 계정 생성 및 구성		
MFAUser	AWS Management 콘솔 액세스에 대한 옵션만 선택하고 암호를 지정합니다.	EC2MFA	MFAUser	정책을 연결하거나 이 그룹에 권한을 부여하지 마십시오.

1단계: MFA 로그인을 강제할 정책 생성

IAM 사용자가 자신의 자격 증명과 MFA 디바이스를 관리하는 데 필요한 권한을 제외한 모든 권한을 거부하는 IAM 고객 관리형 정책을 만드는 것부터 시작합니다.

- 관리자 자격 증명을 지닌 사용자로 AWS Management Console에 로그인합니다. IAM 모범 사례를 준수하려면 AWS 계정 루트 사용자 자격 증명을 사용하여 로그인하지 마십시오. 자세한 내용은 [개별 IAM 사용자 만들기](#) 단원을 참조하십시오.
- <https://console.aws.amazon.com/iam>에서 IAM 콘솔을 엽니다.
- 탐색 창에서 정책을 선택한 후 정책 생성을 선택합니다.
- JSON 탭을 선택하고 다음 JSON 정책 문서에서 텍스트를 복사합니다. [AWS: MFA 인증 IAM 사용자가 My Security Credentials\(내 보안 자격 증명\) 페이지에서 자신의 자격 증명을 관리할 수 있도록 허용합니다. \(p. 343\)](#)
- 정책 텍스트를 JSON 텍스트 상자에 붙여 넣은 다음 Review policy(정책 검토)를 선택합니다. [정책 검사 기 \(p. 382\)](#)가 모든 구문 오류를 보고합니다.

Note

언제든지 Visual editor(시각적 편집기) 및 JSON 탭을 전환할 수 있습니다. 그러나 위의 정책에는 시각적 편집기에서 사용할 수 없는 NotAction 요소가 포함되어 있습니다. Visual editor(시각적 편집기) 탭에 이 정책에 대한 알림 메시지가 표시됩니다. 이 정책으로 작업을 계속하려면 JSON 탭으로 돌아가십시오.

6. 검토 페이지에서 정책 이름에 **Force_MFA**를 입력합니다. 정책 설명에 다음을 입력합니다. **This policy allows users to manage their own passwords and MFA devices but nothing else unless they authenticate with MFA.** 정책 요약을 검토하여 정책이 부여한 권한을 확인한 다음 정책 생성을 선택하여 작업을 저장합니다.

새로운 정책이 관리형 정책 목록에 나타나며 연결 준비가 완료됩니다.

2단계: 테스트 그룹에 정책 연결하기

그 다음에는 MFA 보호 권한을 부여하는 데 사용할 테스트 IAM 그룹에 두 개의 정책을 연결합니다.

1. 탐색 창에서 [Groups]를 선택합니다.
2. 검색 상자에 **EC2MFA**를 입력한 다음 목록에서 그룹 이름(확인란 아님)을 선택합니다.
3. 권한 탭에서 정책 연결을 클릭합니다.
4. 정책 연결 페이지의 검색 상자에 **EC2Full**을 입력한 다음, 목록에서 AmazonEC2FullAccess 옆에 있는 확인란을 선택합니다. 변경 내용을 저장하지 마십시오.
5. 검색 상자에 **Force**를 입력한 다음, 목록에서 Force_MFA 옆에 있는 확인란을 선택합니다.
6. Attach Policy(정책 연결)를 선택합니다.

3단계: 사용자 액세스 테스트

자습서의 이 부분에서는 테스트 사용자로 로그인하여 정책이 의도한 대로 작동하는지 검증합니다.

1. 이전 섹션에서 할당한 암호를 사용해 **MFAUser**로 AWS 계정에 로그인합니다. URL은 <https://<alias or account ID number>.signin.aws.amazon.com/console>을 사용합니다.
2. EC2를 선택해 Amazon EC2 콘솔을 열고 사용자에게 어떤 권한도 없는지 확인합니다.
3. 탐색 표시줄에서 서비스를 선택한 다음, IAM을 선택해 IAM 콘솔을 엽니다.
4. 탐색 창에서 사용자를 선택한 다음, 사용자 MFAUser(확인란이 아님)를 선택합니다. 그룹 탭이 기본적으로 표시되는 경우 그룹 멤버십을 볼 수 있는 권한이 없음을 나타냅니다.
5. 이제 MFA 디바이스를 추가합니다. Security credentials(보안 자격 증명) 탭을 선택합니다. Assigned MFA device(할당된 MFA 디바이스) 옆의 관리를 선택합니다.
6. 이 자습서의 경우 휴대폰의 Google Authenticator 앱과 같은 가상(소프트웨어 기반) MFA 디바이스를 사용합니다. Virtual MFA device(가상 MFA 디바이스)를 선택한 다음 다음 단계를 클릭합니다.

IAM은 QR 코드 그래픽을 포함하여 가상 MFA 디바이스의 구성 정보를 생성 및 표시합니다. 그래픽은 QR 코드를 지원하지 않는 디바이스 상에서 수동 입력할 수 있는 보안 구성 키를 표시한 것입니다.

7. 가상 MFA 앱을 엽니다. (가상 MFA 디바이스의 호스팅에 사용되는 앱 목록은 **가상 MFA 애플리케이션**을 참조하십시오) 가상 MFA 앱이 다수의 계정(다수의 가상 MFA 디바이스)을 지원하는 경우 옵션을 선택하여 새로운 계정(새로운 가상 MFA 디바이스)을 생성합니다.
8. MFA 앱의 QR 코드 지원 여부를 결정한 후 다음 중 한 가지를 실행합니다.
 - 마법사에서 Show QR code(QR 코드 표시)를 선택합니다. 그런 다음 앱을 사용하여 QR 코드를 스캔합니다. 예를 들어 카메라 모양의 아이콘을 선택하거나 코드 스캔(Scan code)과 비슷한 옵션을 선택한 다음, 디바이스의 카메라를 사용하여 코드를 스캔합니다.

- Manage MFA Device(MFA 디바이스 관리) 마법사에서 Show secret key(보안 키 표시)을 선택한 다음 MFA 앱에 보안 키를 입력합니다.

모든 작업을 마치면 가상 MFA 디바이스가 일회용 암호 생성을 시작합니다.

- MFA 디바이스 관리 마법사의 MFA Code 1(MFA 코드 1) 상자에 현재 가상 MFA 디바이스에 표시된 일회용 암호를 입력합니다. 디바이스가 새로운 일회용 암호를 생성할 때까지 최대 30초 기다립니다. 그런 다음 두 번째 일회용 암호를 MFA Code 2(MFA 코드 2) 상자에 입력합니다. Assign MFA(MFA 할당)을 선택합니다.

Important

코드를 생성한 후 즉시 요청을 제출하십시오. 코드를 생성하고 너무 오래 시간이 지난 후 요청을 제출하면 MFA 디바이스가 사용자와 연결은 되지만 MFA 디바이스가 동기화되지 않습니다. 이는 시간 기반 일회용 암호(TOTP)가 잠시 후에 만료되기 때문입니다. 이 경우, [디바이스를 재동기화 \(p. 115\)](#)할 수 있습니다.

이제 AWS에서 가상 MFA 디바이스를 사용할 준비를 마쳤습니다.

- 콘솔에서 로그아웃한 다음, **MFAUser**로 다시 로그인합니다. 이번에는 AWS가 휴대전화로 받은 MFA 코드를 입력하도록 요청합니다. 코드를 받아 상자에 입력한 후 전송을 선택합니다.
- EC2를 선택하여 Amazon EC2 콘솔을 다시 엽니다. 이번에는 모든 정보를 볼 수 있으며 원하는 작업은 모두 수행할 수 있습니다. 이 사용자로서 다른 콘솔로 이동하면 이 자습서의 정책이 Amazon EC2에게만 액세스 권한을 부여하기 때문에 액세스가 거부된다는 메시지가 표시됩니다.

관련 리소스

IAM 사용 설명서에서 수록된 관련 내용은 다음 리소스 단원을 참조하십시오.

- [AWS에서 멀티 팩터 인증\(MFA\) 사용하기 \(p. 96\)](#)
- [MFA 디바이스 활성화 \(p. 97\)](#)
- [IAM 로그인 페이지에 MFA 디바이스 사용 \(p. 57\)](#)

IAM 모범 사례 및 사용 사례

IAM의 이점을 극대화하기 위해 권장 모범 사례에 대해 알아보시기 바랍니다. 이를 위한 한 가지 방법은 실제 시나리오에서 다른 AWS 서비스와 함께 IAM을 어떻게 사용하는지 알아보는 것입니다.

주제

- [IAM 모범 사례 \(p. 43\)](#)
- [기업 사용 사례 \(p. 50\)](#)

IAM 모범 사례

 [Follow us on Twitter](#)

AWS 리소스를 안전하게 보호하기 위해 AWS Identity and Access Management(IAM) 서비스에 대한 다음 권장 사항을 따르십시오.

주제

- [AWS 계정 루트 사용자 액세스 키 잠금 \(p. 43\)](#)
- [개별 IAM 사용자 만들기 \(p. 44\)](#)
- [그룹을 사용하여 IAM 사용자에게 권한을 할당합니다. \(p. 44\)](#)
- [최소 권한 부여 \(p. 44\)](#)
- [AWS 관리형 정책으로 권한 사용 시작 \(p. 45\)](#)
- [인라인 정책 대신 고객 관리형 정책 사용 \(p. 45\)](#)
- [액세스 레벨을 이용한 IAM 권한 검토 \(p. 46\)](#)
- [사용자에 대한 강력한 암호 정책 구성 \(p. 47\)](#)
- [권한 있는 사용자에 대해 MFA 활성화 \(p. 47\)](#)
- [Amazon EC2 인스턴스에서 실행되는 애플리케이션에 역할 사용 \(p. 47\)](#)
- [역할을 사용하여 권한 위임 \(p. 47\)](#)
- [액세스 키를 공유하지 마십시오 \(p. 48\)](#)
- [자격 증명을 정기적으로 교체 \(p. 48\)](#)
- [불필요한 자격 증명 삭제 \(p. 48\)](#)
- [보안 강화를 위해 정책 조건 사용 \(p. 48\)](#)
- [AWS 계정의 활동 모니터링 \(p. 49\)](#)
- [IAM 모범 사례에 대한 동영상 프레젠테이션 \(p. 49\)](#)

AWS 계정 루트 사용자 액세스 키 잠금

액세스 키(액세스 키 ID 및 보안 액세스 키)를 사용하여 프로그래밍 방식으로 AWS에 요청을 할 수 있습니다. 그러나 AWS 계정 루트 사용자 액세스 키는 사용하지 마십시오. AWS 계정 루트 사용자에 대한 액세스 키는 결제 정보를 포함하여 모든 AWS 서비스의 전체 리소스에 대해 전체 액세스 권한을 부여합니다. AWS 계정 루트 사용자 액세스 키에 연결된 권한은 줄일 수 없습니다.

따라서 신용카드 번호 또는 다른 중요한 기밀 정보와 같이 루트 사용자 액세스 키를 보호해야 합니다. 이를 위한 몇 가지 방법은 다음과 같습니다.

- AWS 계정 루트 사용자에 대한 액세스 키가 아직 없다면 필요할 때까지 만들지 마십시오. 대신 계정 이메일 주소와 암호를 사용하여 AWS Management 콘솔에 로그인한 후 [IAM 사용자를 만들어 \(p. 17\)](#) 관리 권한을 부여합니다.
- AWS 계정 루트 사용자에 대한 액세스 키가 있다면 삭제하고, 계속 유지해야 할 경우 주기적으로 액세스 키를 교체(변경)하십시오. 루트 사용자 액세스 키를 삭제 또는 교체하려면 AWS Management 콘솔의 [내 보안 자격 증명 페이지](#)에서 계정의 이메일 주소와 암호를 사용하여 로그인합니다. 액세스 키 섹션에서 액세스 키를 관리할 수 있습니다. 액세스 키 교체에 대한 자세한 내용은 [액세스 키 교체 \(p. 92\)](#) 단원을 참조하십시오.
- 다른 사람과 AWS 계정 루트 사용자 암호 또는 액세스 키를 공유하지 마십시오. 이 설명서의 나머지 섹션에서 AWS 계정 루트 사용자 자격 증명을 다른 사용자와 공유하거나 애플리케이션에 포함하는 것을 피할 수 있는 여러 가지 방법을 참조할 수 있습니다.
- 강력한 암호를 사용하여 AWS Management 콘솔에 대한 계정 수준의 액세스를 보호하십시오. AWS 계정 루트 사용자 암호 관리에 대한 자세한 정보는 [AWS 계정 루트 사용자 암호 변경 \(p. 78\)](#) 단원을 참조하십시오.
- AWS 계정 루트 사용자 계정에서 AWS 멀티 팩터 인증(MFA)을 활성화합니다. 자세한 정보는 [AWS에서 멀티 팩터 인증\(MFA\) 사용하기 \(p. 96\)](#) 단원을 참조하십시오.

개별 IAM 사용자 만들기

AWS 계정 루트 사용자 자격 증명을 사용하여 AWS에 액세스하거나 다른 사용자와 공유하지 마십시오. 대신 AWS 계정에 액세스해야 하는 사용자에 대해 별도의 사용자 계정을 만들어주십시오. 관리자에 대해서도 IAM 사용자를 만들어 관리 권한을 부여한 후 모든 관리 작업에 대해 이 IAM 사용자를 사용하십시오. 이를 위한 자세한 방법은 [첫 번째 IAM 관리자 및 그룹 생성 \(p. 17\)](#) 단원을 참조하십시오.

계정에 액세스하는 사용자에 대해 개별 IAM 사용자를 만들면 각 IAM 사용자에 따라 서로 다른 보안 자격 증명 조합을 부여할 수 있습니다. 또한 각 IAM 사용자에게 다양한 권한을 부여하고, 필요할 경우 언제든지 IAM 사용자의 권한을 변경 또는 취소할 수 있습니다. (루트 사용자 자격 증명을 제공한 후에는 다시 취소하기가 쉽지 않으며 권한을 제한할 수 없습니다.)

Note

그러나 개별 IAM 사용자에게 권한을 설정하기 전에 다음과 같은 그룹 관련 참고 사항을 고려해 보십시오.

그룹을 사용하여 IAM 사용자에게 권한을 할당합니다.

개별 IAM 사용자에 대해 권한을 정의하는 대신, 업무(관리자, 개발자, 회계 등)에 관련된 그룹을 만드는 것이 더 편리할 수 있습니다. 그런 다음 각 그룹별로 관련 권한을 정의합니다. 끝으로 해당 그룹에 IAM 사용자를 할당합니다. 그룹에 할당된 권한은 IAM 그룹에 속한 모든 사용자에게 상속됩니다. 따라서 한번에 그룹 내 모든 사용자에 대해 변경 사항을 적용할 수 있습니다. 사내에서 직원의 부서가 변경되면 해당 IAM 사용자가 속한 IAM 그룹만 변경하면 됩니다.

자세한 정보는 다음을 참조하십시오.

- [첫 번째 IAM 관리자 및 그룹 생성 \(p. 17\)](#)
- [IAM 그룹 관리 \(p. 148\)](#)

최소 권한 부여

IAM 정책을 만들 때는 최소 권한 부여의 표준 보안 조언을 따르거나, 작업 수행에 필요한 최소한의 권한만 부여합니다. 사용자들이 수행해야 하는 작업을 파악한 후 사용자들이 해당 작업만 수행하도록 사용자에 대한 정책을 작성합니다.

최소한의 권한 조합으로 시작하여 필요에 따라 추가 권한을 부여합니다. 처음부터 권한을 많이 부여한 후 나중에 줄이는 방법보다 이 방법이 안전합니다.

액세스 레벨 그룹화를 사용하면 정책이 부여하는 액세스 레벨을 이해할 수 있습니다. 정책 작업 (p. 506)은 List, Read, Write, Permissions management 또는 Tagging으로 분류됩니다. 예를 들어 List 및 Read 액세스 레벨에서 작업을 선택하여 사용자에게 읽기 전용 액세스 권한을 부여할 수 있습니다. 정책 요약을 사용하여 액세스 레벨 권한을 이해하는 방법에 대해 알아보려면 [액세스 레벨을 이용한 IAM 권한 검토 \(p. 46\)](#) 단원을 참조하십시오.

이 경우 서비스에서 마지막으로 액세스한 데이터가 유용할 수 있습니다. 사용자, 그룹, 역할 또는 정책에 대한 IAM 콘솔 세부 정보 페이지의 액세스 관리자 탭에서 이 데이터를 확인합니다. AWS CLI 또는 AWS API를 사용하여 마지막으로 액세스한 데이터를 검색할 수도 있습니다. 이 데이터에는 사용자, 그룹, 역할 또는 정책을 사용하는 모든 사용자가 액세스를 시도한 서비스 및 그 시기에 대한 정보가 포함됩니다. 이 정보를 사용하여 불필요한 권한을 확인할 수 있으므로 IAM 정책을 미세 조정함으로써 최소 권한의 원칙을 보다 잘 준수할 수 있습니다. 자세한 정보는 [서비스에서 마지막으로 액세스한 데이터를 사용하여 권한 줄이기 \(p. 409\)](#) 단원을 참조하십시오.

권한을 더 줄이려면 CloudTrail 이벤트 기록에서 계정의 이벤트를 볼 수 있습니다. CloudTrail 이벤트 로그에는 정책 권한을 줄이기 위해 사용할 수 있는 자세한 이벤트 정보가 포함되어 있으며 IAM 엔터티에 필요한 작업과 리소스만 포함됩니다. 자세한 정보는 AWS CloudTrail 사용 설명서에서 [CloudTrail 콘솔에서 CloudTrail 이벤트 보기](#) 단원을 참조하십시오.

자세한 정보는 다음을 참조하십시오.

- [액세스 관리 \(p. 304\)](#)
- 각 서비스의 정책 주제에서는 서비스별 리소스에 대해 정책을 작성하는 방법의 예제를 제공합니다. 예제:
 - Amazon DynamoDB 개발자 안내서의 [Amazon DynamoDB 인증 및 액세스 제어](#)
 - Amazon Simple Storage Service 개발자 가이드의 [버킷 정책 및 사용자 정책 사용](#)
 - Amazon Simple Storage Service 개발자 가이드의 [ACL\(액세스 제어 목록\) 개요](#)

AWS 관리형 정책으로 권한 사용 시작

직원에게 필요한 권한만 제공하려면 IAM 정책에 대한 시간과 자세한 지식이 필요합니다. 직원들은 필요하거나 사용해야 하는 AWS 서비스를 악힐 시간이 필요합니다. 관리자는 IAM에 대해 배우고 테스트할 시간이 필요합니다.

신속하게 시작하려면 AWS 관리형 정책을 사용하여 직원에게 시작해야 하는 권한을 부여합니다. 이 정책은 이미 계정에서 사용할 수 있으며 AWS에 의해 유지 관리 및 업데이트됩니다. AWS 관리형 정책에 대한 자세한 정보는 [AWS 관리형 정책 \(p. 312\)](#) 단원을 참조하십시오.

AWS 관리형 정책은 여러 가지 일반 사용 사례에서 권한을 제공할 목적으로 설계되었습니다. [AmazonDynamoDBFullAccess](#) 및 [IAMFullAccess](#)와 같은 전체 액세스 AWS 관리형 정책은 서비스에 대한 전체 액세스 권한을 부여하여 서비스 관리자에 대한 권한을 정의합니다. [AWSCodeCommitPowerUser](#) 및 [AWSKeyManagementServicePowerUser](#)와 같은 파워 사용자 AWS 관리형 정책은 권한 관리 권한을 허용하지 않고 AWS 서비스에 대한 여러 수준의 액세스를 제공합니다. [AmazonMobileAnalyticsWriteOnlyAccess](#) 및 [AmazonEC2ReadOnlyAccess](#)와 같은 부분 액세스 AWS 관리형 정책은 AWS 서비스에 대한 특정 액세스 수준을 제공합니다. AWS 관리형 정책을 사용하면 정책을 직접 작성하는 것보다 쉽게 사용자, 그룹 및 역할에 적절한 권한을 할당할 수 있습니다.

직무 기능에 관한 AWS 관리형 정책은 다양한 서비스에 적용할 수 있으며 IT 업계의 일반적인 직무 기능과 연계됩니다. 직무 정책의 목록과 설명은 [직무 기능에 대한 AWS 관리형 정책 \(p. 543\)](#) 단원을 참조하십시오.

인라인 정책 대신 고객 관리형 정책 사용

사용자 지정 정책의 경우 인라인 정책보다는 관리형 정책의 사용을 권장합니다. 이 정책을 사용하면 콘솔의 한 위치에서 모든 관리형 정책을 볼 수 있다는 이점이 있습니다. 또한 단일 AWS CLI 또는 AWS API 작업으

로 이 정보를 볼 수도 있습니다. 인라인 정책은 IAM ID(사용자, 그룹 또는 역할)에만 존재하는 정책입니다. 관리형 정책은 여러 자격 증명에 연결할 수 있는 별도의 IAM 리소스입니다. 자세한 정보는 [관리형 정책과 인라인 정책 \(p. 312\)](#) 단원을 참조하십시오.

계정에 인라인 정책이 있는 경우 이를 관리형 정책으로 변환할 수 있습니다. 이렇게 하려면 정책을 새로운 관리형 정책에 복사하고 새 정책을 인라인 정책이 있는 자격 증명에 연결한 다음 인라인 정책을 삭제하십시오. 아래 지침을 사용하여 이 작업을 수행할 수 있습니다.

인라인 정책을 관리형 정책으로 변환하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
 2. 탐색 창에서 그룹, 사용자 또는 역할을 선택합니다.
 3. 목록에서 제거할 정책이 있는 그룹, 사용자 또는 역할 이름을 선택합니다.
 4. Permissions 탭을 선택합니다. 그룹을 선택한 경우 필요에 따라 Inline Policies(인라인 정책) 섹션을 확장합니다.
 5. 그룹의 경우 제거할 인라인 정책 옆의 정책 표시를 선택합니다. 사용자 및 역할에 대해 필요한 경우 Show **n** more(n개 더 표시)를 선택한 다음 제거할 인라인 정책 옆에 있는 화살표를 선택합니다.
 6. 정책에 대한 JSON 정책 문서를 복사합니다.
 7. 탐색 창에서 정책을 선택합니다.
 8. 정책 생성을 선택한 후 JSON 탭을 선택합니다.
 9. 기존 텍스트를 JSON 정책 텍스트로 바꾸고 정책 검토를 선택합니다.
 10. 정책 이름을 입력하고 정책 생성을 선택합니다.
 11. 탐색 창에서 그룹, 사용자 또는 역할을 선택한 다음 제거하려는 정책이 있는 그룹, 사용자 또는 역할의 이름을 다시 선택합니다.
 12. 그룹의 경우 정책 연결을 선택합니다. 사용자 및 역할의 경우 권한 추가를 선택합니다.
 13. 그룹에 대해 새 정책 이름 옆의 확인란을 선택한 다음 정책 연결을 선택합니다. 사용자 또는 역할의 경우 권한 추가를 선택합니다. 다음 페이지에서 기존 정책 직접 연결을 선택하고 새 정책 이름 옆의 확인란을 선택한 다음 다음: 검토를 선택하고 권한 추가를 선택합니다.
- 그룹, 사용자 또는 역할에 대한 요약 페이지로 돌아갑니다.
14. 그룹의 경우 제거할 인라인 정책 옆의 정책 제거를 선택합니다. 사용자 또는 역할의 경우 제거할 인라인 정책 옆의 X를 선택합니다.

경우에 따라 관리형 정책에 대한 인라인 정책을 선택하는 것이 좋습니다. 자세한 정보는 [관리형 정책과 인라인 정책의 선택 \(p. 315\)](#) 단원을 참조하십시오.

액세스 레벨을 이용한 IAM 권한 검토

AWS 계정의 보안을 개선하려면 모든 IAM 정책을 정기적으로 검토하고 모니터링해야 합니다. 정책은 필요한 작업을 수행하는 데 필요한 [최소 권한 \(p. 44\)](#)만 부여해야 합니다.

정책을 보면 그 정책 안에서 각 서비스에 대한 액세스 레벨의 요약이 들어 있는 [정책 요약 \(p. 419\)](#)을 확인할 수 있습니다. AWS는 작업 내용에 따라 각 서비스 작업을 네 액세스 레벨, 즉 List, Read, Write, Permissions management 중 하나로 분류합니다. 이러한 액세스 레벨을 사용하여 어떤 작업을 정책에 포함할지 결정할 수 있습니다.

예를 들어 Amazon S3 서비스의 경우, 다수의 사용자가 List 및 Read 작업에 액세스하도록 허용할 수 있습니다. 이러한 작업은 사용자가 Amazon S3에 버킷을 나열하고 객체를 가져오도록 허용합니다. 하지만 소수의 사용자만 Amazon S3 Write 작업에 액세스하여 버킷을 삭제하거나 S3 버킷에 객체를 넣도록 허용해야 합니다. 또한 관리자만 Amazon S3 Permissions management 작업에 액세스할 수 있도록 권한을 줄여야 합니다. 그래야 제한된 수의 사람만 Amazon S3에서 버킷 정책을 관리할 수 있습니다. IAM 및 AWS Organizations 서비스의 Permissions management 작업에서는 이 점이 특히 중요합니다.

서비스의 각 작업에 할당된 액세스 레벨 분류를 보려면 [??? 단원](#)을 참조하십시오.

정책의 액세스 레벨을 보려면 먼저 정책 요약을 찾아야 합니다. 정책 요약에서 관리형 정책에 관한 부분은 정책 페이지에, 사용자에게 연결되는 정책 부분은 사용자 페이지에 수록됩니다. 자세한 정보는 [정책 요약\(서비스 목록\) \(p. 419\)](#) 단원을 참조하십시오.

정책 요약의 액세스 레벨 열에는 정책이 서비스의 네 가지 AWS 액세스 레벨 중 하나 이상에 대해 전체 또는 제한 액세스 권한을 제공한다고 표시되어 있습니다. 또는 정책이 서비스 내 모든 작업에 모든 액세스를 제공한다고 표시되어 있을 수도 있습니다. 이 액세스 레벨 열에 수록된 정보를 통해 정책이 제공하는 액세스 레벨을 알 수 있습니다. 그런 다음 AWS 계정을 더 안전하게 사용하기 위한 조치를 취할 수 있습니다. 액세스 레벨 요약에 대한 세부 정보와 예제는 [정책 요약에서 액세스 레벨 요약 이해하기 \(p. 427\)](#) 단원을 참조하십시오.

사용자에 대한 강력한 암호 정책 구성

사용자가 직접 암호를 변경하도록 허용할 경우 강력한 암호를 만들고 주기적으로 암호를 변경하도록 해야 합니다. IAM 콘솔의 [계정 설정](#) 페이지에서 계정 암호 정책을 만들 수 있습니다. 암호 정책을 사용하여 최소 길이, 비 알파벳 문자 포함 여부, 교체 주기 등의 암호 요구 사항을 정의할 수 있습니다.

자세한 정보는 [IAM 사용자의 계정 암호 정책 설정 \(p. 79\)](#) 단원을 참조하십시오.

권한 있는 사용자에 대해 MFA 활성화

보안을 강화하기 위해 중요한 리소스 또는 API 작업에 대해 액세스 권한이 부여된 IAM 사용자에 대해 멀티 팩터 인증(MFA)을 적용합니다. MFA에는 인증 문제에 응답을 생성하는 디바이스가 있습니다. 로그인 과정을 완료하려면 사용자의 자격 증명과 디바이스에서 생성한 응답 두 가지가 모두 필요합니다. 응답은 다음 중 한 가지 방법으로 생성합니다.

- 가상 및 하드웨어 MFA 디바이스가 코드를 생성하여 앱 또는 디바이스에 보여준 후 로그인 화면에 입력합니다.
- 사용자가 디바이스를 터치하면 U2F 보안 키가 응답을 생성합니다. 로그인 화면에 사용자가 직접 코드를 입력할 필요가 없습니다.

자세한 정보는 [AWS에서 멀티 팩터 인증\(MFA\) 사용하기 \(p. 96\)](#) 단원을 참조하십시오.

Amazon EC2 인스턴스에서 실행되는 애플리케이션에 역할 사용

Amazon EC2 인스턴스에서 실행되는 애플리케이션이 다른 AWS 서비스에 액세스하려면 자격 증명이 필요하며, 이 애플리케이션에 안전하게 자격 증명을 제공하려면 IAM 역할을 사용합니다. 역할에는 특정 사용자나 그룹이 아닌 권한의 조합이 설정됩니다. 또한 역할에는 IAM 사용자와 달리 영구적인 자격 증명 조합이 부여되지 않습니다. Amazon EC2의 경우 IAM은 EC2 인스턴스에 동적으로 생성되는 임시 자격 증명을 제공하며 이 자격 증명은 자동 교체됩니다.

EC2 인스턴스 실행 시 실행 파라미터로 인스턴스에 대한 역할을 지정할 수 있습니다. EC2 인스턴스에서 실행되는 애플리케이션은 AWS 리소스에 액세스할 때 역할의 자격 증명을 사용할 수 있습니다. 역할의 권한에 따라 애플리케이션에서 수행할 수 있는 작업이 결정됩니다.

자세한 정보는 [IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여하기 \(p. 240\)](#) 단원을 참조하십시오.

역할을 사용하여 권한 위임

다른 AWS 계정의 사용자가 내 AWS 계정의 리소스에 액세스하도록 허용하려면 계정 간에 보안 자격 증명을 공유하지 마십시오. 대신 IAM 역할을 사용하십시오. 다른 계정의 IAM 사용자에게 어떤 권한이 허용되는지

지정하는 역할을 정의할 수 있습니다. 어떤 AWS 계정에 해당 역할을 수임하도록 허용된 IAM 사용자가 있는지도 지정할 수 있습니다.

자세한 정보는 [역할 용어 및 개념 \(p. 153\)](#) 단원을 참조하십시오.

액세스 키를 공유하지 마십시오

액세스 키는 AWS으로의 프로그래밍 방식 액세스를 제공합니다. 액세스 키를 암호화되지 않은 코드에 삽입하거나 AWS 계정 사용자 간에 이들 보안 자격 증명을 공유하지 마십시오. AWS로 액세스가 필요한 애플리케이션은 IAM 역할을 사용하여 임시 보안 자격 증명 검색하도록 프로그램을 구성합니다. 사용자별 프로그래밍 액세스를 허용하고자 한다면 개인 액세스 키가 있는 IAM 사용자를 만들니다.

자세한 정보는 [IAM 역할\(AWS API\)로 전환하기 \(p. 238\)](#) 및 [IAM 사용자의 액세스 키 관리 \(p. 88\)](#)을(를) 참조하십시오.

자격 증명을 정기적으로 교체

암호와 액세스 키를 정기적으로 교체하고, 계정의 모든 IAM 사용자도 이와 같이 하도록 해야 합니다. 그러면 자신도 모르게 암호 또는 액세스 키가 손상되어도 이 손상된 자격 증명이 리소스 액세스에 사용되는 기간을 줄일 수 있습니다. 계정에 모든 IAM 사용자가 주기적으로 암호를 교체하도록 요구하는 암호 정책을 적용하고, 그 빈도를 선택할 수 있습니다.

계정의 암호 정책 설정에 대한 자세한 정보는 [IAM 사용자의 계정 암호 정책 설정 \(p. 79\)](#) 단원을 참조하십시오.

IAM 사용자의 액세스 키 교체에 대한 자세한 정보는 [액세스 키 교체 \(p. 92\)](#) 단원을 참조하십시오.

불필요한 자격 증명 삭제

필요 없는 IAM 사용자 자격 증명(암호 및 액세스 키)은 삭제합니다. 예를 들어 콘솔을 사용하지 않는 애플리케이션에 대해 IAM 사용자를 생성한 경우 IAM 사용자는 암호가 필요하지 않습니다. 마찬가지로 사용자가 콘솔만 사용하는 경우 액세스 키를 제거하십시오. 최근에 사용된 적이 없는 암호와 액세스 키는 삭제해야 할 자격 증명을 식별하기 위한 좋은 기준이 될 수 있습니다. 콘솔, CLI, API를 사용하여 또는 자격 증명 보고서를 다운로드하여 미사용 암호나 액세스 키를 확인할 수 있습니다.

최근 사용되지 않은 IAM 사용자 자격 증명 확인에 대한 자세한 정보는 [미사용 자격 증명 찾기 \(p. 133\)](#) 단원을 참조하십시오.

IAM 사용자의 암호 삭제에 대한 자세한 정보는 [IAM 사용자의 암호 관리 \(p. 82\)](#) 단원을 참조하십시오.

IAM 사용자의 액세스 키 비활성화 또는 삭제에 대한 자세한 정보는 [IAM 사용자의 액세스 키 관리 \(p. 88\)](#) 단원을 참조하십시오.

IAM 자격 증명 보고서에 대한 자세한 정보는 [AWS 계정의 자격 증명 보고서 가져오기 \(p. 135\)](#) 단원을 참조하십시오.

보안 강화를 위해 정책 조건 사용

필요할 경우 IAM 정책에서 리소스에 대한 액세스 허용 조건을 정의할 수 있습니다. 예를 들어 요청을 할 수 있는 IP 주소의 범위를 지정하도록 조건을 작성할 수 있습니다. 특정 기간이나 시간 범위 내에서만 요청이 가능하도록 조건을 작성할 수도 있습니다. 또한 SSL 또는 MFA(멀티 팩터 인증)를 사용하도록 조건을 설정할 수 있습니다. 예를 들어 MFA 디바이스를 사용하여 인증된 사용자만 Amazon EC2 인스턴스를 종료할 수 있도록 조건을 지정할 수 있습니다.

자세한 정보는 IAM 정책 요소 참조에서 [IAM JSON 정책 요소: Condition \(p. 510\)](#) 단원을 참조하십시오.

AWS 계정의 활동 모니터링

AWS의 로깅 기능을 사용하여 사용자가 계정에서 수행한 작업과 사용한 리소스를 확인할 수 있습니다. 로그 파일에는 작업 시간 및 날짜, 작업의 소스 IP, 부족한 권한으로 인해 실패한 작업 등이 나와 있습니다.

로깅 기능은 다음과 같은 AWS 서비스에서 제공됩니다.

- [Amazon CloudFront](#) – CloudFront에서 받는 사용자 요청을 기록합니다. 자세한 정보는 Amazon CloudFront 개발자 안내서의 [액세스 로그](#) 단원을 참조하십시오.
- [AWS CloudTrail](#) – AWS 계정에서 또는 이를 대신하여 수행된 AWS API 호출 및 관련 이벤트를 기록합니다. 자세한 정보는 [AWS CloudTrail User Guide](#) 단원을 참조하십시오.
- [Amazon CloudWatch](#) – AWS 클라우드 리소스 및 AWS에서 실행되는 애플리케이션을 모니터링합니다. 정의한 지표에 기반하여 CloudWatch에서 경보를 설정할 수 있습니다. 자세한 정보는 [Amazon CloudWatch 사용 설명서](#) 단원을 참조하십시오.
- [AWS Config](#) – IAM 사용자, 그룹, 역할 및 정책 등 AWS 리소스의 구성에 대한 세부적인 기록 정보를 제공합니다. 예를 들어, AWS Config를 사용하여 특정 시점에 사용자 또는 그룹에 속한 권한을 확인할 수 있습니다. 자세한 정보는 [AWS Config Developer Guide](#) 단원을 참조하십시오.
- [Amazon Simple Storage Service\(Amazon S3\)](#) – Amazon S3 버킷에 대한 액세스 요청을 기록합니다. 자세한 정보는 Amazon Simple Storage Service 개발자 가이드에서 [서버 액세스 로깅](#) 단원을 참조하십시오.

IAM 모범 사례에 대한 동영상 프레젠테이션

다음 동영상에는 이러한 모범 사례 및 여기서 논의한 기능을 사용하여 작업을 수행하는 방법에 대한 상세 정보를 보여주는 컨퍼런스 프레젠테이션이 포함되어 있습니다.

[AWS re:Invent 2015 - IAM 모범 사례](#)

기업 사용 사례

IAM의 간단한 기업 사용 사례를 통해 사용자의 AWS 액세스 권한을 제어하기 위한 서비스 구현의 기본적인 방법을 이해할 수 있습니다. 사용 사례는 일반적인 용어로 서술되며 원하는 결과를 달성하기 위해 IAM API를 사용하는 방법에 대한 기술적인 내용을 다루지 않습니다.

이 사용 사례에서는 Example Corp라는 가상의 회사가 IAM을 사용하는 2가지 일반적인 방법에 대해 살펴보겠습니다. 첫 번째 시나리오는 Amazon Elastic Compute Cloud(Amazon EC2)를 가정합니다. 두 번째는 Amazon Simple Storage Service(Amazon S3)를 가정합니다.

다른 AWS 서비스와 함께 IAM을 사용하는 방법에 대한 자세한 정보는 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 섹션 단원을 참조하십시오.

주제

- [Example Corp의 초기 설정 \(p. 50\)](#)
- [Amazon EC2의 IAM 사용 사례 \(p. 50\)](#)
- [Amazon S3의 IAM 사용 사례 \(p. 51\)](#)

Example Corp의 초기 설정

Example Corp의 창립자인 John은 회사 초창기에는 자신이 직접 AWS 계정을 만들어 AWS 제품을 관리했으며, 이후 개발자와 관리자, 테스트 담당자, 관리자 및 시스템 관리자로 일할 직원들을 고용했습니다.

John은 AWS Management 콘솔을 사용하여 AWS 계정 루트 사용자 자격 증명으로 자신이 사용할 John이라는 계정과 Admins라는 그룹을 생성했습니다. 그는 AWS 관리형 정책 [AdministratorAccess](#)를 사용하여 Admins 그룹에 AWS 계정의 리소스에서 모든 작업을 수행할 수 있는 권한을 부여합니다. 그런 다음 John 사용자를 Admins 그룹에 추가했습니다. 이처럼 관리자 그룹과 IAM 사용자를 만든 후 이 사용자를 관리자 그룹에 추가하기 위한 단계별 지침은 [첫 번째 IAM 관리자 및 그룹 생성 \(p. 17\)](#) 단원을 참조하십시오.

이제 John은 AWS와 상호 작용하는 데 루트 사용자의 자격 증명을 사용하는 대신 개인 사용자 계정의 자격 증명만 사용합니다.

John은 또한 AWS 계정 내 모든 사용자에게 계정 수준의 권한을 간편하게 적용할 수 있도록 AllUsers라는 그룹을 생성합니다. 그리고 자신도 이 그룹에 추가했습니다. 그런 다음 Developers, Testers, Managers, SysAdmins라고 하는 그룹을 각각 만들었습니다. 그리고 각 직원들에 대해 사용자 계정을 만들어 해당 그룹에 추가했습니다. 또한 모든 사용자를 AllUsers 그룹에도 추가했습니다. 그룹 생성에 대한 자세한 정보는 [IAM 그룹 생성 \(p. 147\)](#) 섹션을, 사용자 생성에 대한 자세한 정보는 [AWS 계정의 IAM 사용자 생성 \(p. 65\)](#) 단원을 참조하고, 사용자를 그룹에 추가하는 방법은 [IAM 그룹 관리 \(p. 148\)](#) 단원을 참조하십시오.

Amazon EC2의 IAM 사용 사례

Example Corp와 같은 회사는 일반적으로 IAM을 사용하여 Amazon EC2와 같은 서비스와 상호 작용합니다. 이 부분의 사용 사례를 이해하기 위해서는 Amazon EC2에 대한 기본적인 지식이 필요합니다. Amazon EC2에 대한 자세한 정보는 [Linux 인스턴스용 Amazon EC2 사용 설명서](#) 단원을 참조하십시오.

그룹에 대한 Amazon EC2 권한

"경계" 제어를 제공하기 위해 John은 정책을 AllUsers 그룹에 연결합니다. 이 정책은 Example Corp 회사 네트워크의 IP 주소가 아닌 주소에서 시작된 모든 사용자의 AWS 요청을 거부합니다.

Example Corp는 다음과 같이 그룹에 따라 서로 다른 권한을 부여했습니다.

- 시스템 관리자 – AMI, 인스턴스, 스냅샷, 볼륨, 보안 그룹 등을 생성하고 관리하기 위한 권한이 필요합니다. John은 그룹 구성원에게 모든 Amazon EC2 작업을 사용할 수 있는 권한을 부여하는 정책을 SysAdmins 그룹에 연결합니다.

- 개발자 – 인스턴스를 사용한 작업 권한만 필요합니다. 따라서 John은 개발자가 `DescribeInstances`, `RunInstances`, `StopInstances`, `StartInstances`, `TerminateInstances`를 호출할 수 있는 권한을 부여하는 정책을 Developers 그룹에 연결합니다.

Note

Amazon EC2는 SSH 키, Windows 암호 및 보안 그룹을 사용하여 특정 Amazon EC2 인스턴스의 운영 체제에 액세스할 사용자를 제어합니다. IAM 시스템에서는 특정 인스턴스의 운영 체제 액세스를 허용 또는 거부할 방법을 제공하지 않습니다.

- 관리자 – 현재 제공되고 있는 Amazon EC2 리소스를 나열하는 것 외, 어떤 Amazon EC2 작업도 수행할 필요가 없습니다. 따라서 John은 Amazon EC2 "Describe" API 작업만 호출할 수 있는 권한을 부여하는 정책을 Managers 그룹에 연결합니다.

이러한 각 정책의 예를 보려면 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [IAM 자격 증명 기반 정책 예제 \(p. 341\)](#) 및 [AWS Identity and Access Management](#) 단원을 참조하십시오.

사용자의 역할 변경

그러다가 개발자 중 한 명인 Paulo가 역할을 바꾸어 관리자가 되었습니다. John은 Paulo를 Developers 그룹에서 Managers 그룹으로 옮겼습니다. 이제 Paulo는 Managers 그룹에 속하며 더 이상 Amazon EC2 인스턴스와 상호 작용할 수 없습니다. 즉, 인스턴스를 실행하거나 시작할 수 없으며, 이전에 자신이 시작한 인스턴스일지라도 더 이상 기존 인스턴스를 중지하거나 종료할 수 없습니다. Example Corp 사용자가 시작한 인스턴스를 나열할 수만 있습니다.

Amazon S3의 IAM 사용 사례

Example Corp와 같은 회사는 또한 기본적으로 Amazon S3와 함께 IAM을 사용합니다. John은 `example_bucket`이라는 회사용 Amazon S3 버킷을 생성했습니다.

추가 사용자와 그룹 생성

직원인 Zhang과 Mary는 모두 회사의 버킷에 데이터를 생성할 수 있어야 합니다. 또한 개발자들이 작업 중인 공유 데이터를 읽고 쓸 수 있어야 합니다. 이를 위해 John은 다음 그림과 같은 Amazon S3 키 접두사 체계에 따라 `example_bucket`의 데이터에 대한 논리적 구조를 정했습니다.

```
/example_bucket
  /home
    /zhang
    /mary
  /share
    /developers
    /managers
```

John은 마스터 `/example_bucket`을 각 직원별 홈 디렉터리, 그리고 개발자와 관리자의 그룹에서 함께 공유하는 영역으로 나누었습니다.

그런 다음 John은 다음과 같이 사용자와 그룹에 대해 권한을 부여하는 정책의 조합을 생성했습니다.

- Zhang의 홈 디렉터리 액세스 – John은 Zhang에게 Amazon S3 키 접두사 `/example_bucket/home/zhang/`로 객체에 대해 읽기, 쓰기 및 나열 권한을 부여하는 정책을 연결했습니다.
- Mary의 홈 디렉터리 액세스 – John은 Mary에게 Amazon S3 키 접두사 `/example_bucket/home/mary/`로 객체에 대해 읽기, 쓰기 및 나열 권한을 부여하는 정책을 연결했습니다.
- Developers 그룹에 대한 공유 디렉터리 액세스 – John은 개발자에게 `/example_bucket/share/developers/` 키 접두사로 객체에 대해 읽기, 쓰기 및 나열 권한을 부여하는 정책을 그룹에 연결했습니다.

- Managers 그룹에 대한 공유 디렉터리 액세스 – John은 관리자에게 /example_bucket/share/managers/ 키 접두사로 객체에 대해 읽기, 쓰기 및 나열 권한을 부여하는 정책을 연결했습니다.

Note

Amazon S3는 버킷 또는 객체를 만든 사용자에게 해당 버킷 또는 객체에 대해 자동으로 다른 작업을 수행할 권한을 부여하지 않습니다. 따라서 IAM 정책에서 명시적으로 사용자에게 사용자가 생성한 Amazon S3 리소스를 사용할 권한을 부여해야 합니다.

이러한 각 정책의 예를 보려면 Amazon Simple Storage Service 개발자 가이드에서 [액세스 제어 단원](#)을 참조하십시오. 런타임 시 정책이 어떻게 평가되는지 알아보려면 [정책 평가 로직 \(p. 531\)](#) 단원을 참조하십시오.

사용자의 역할 변경

그러다가 개발자 중 한 명인 Zhang이 역할을 바꾸어 관리자가 되었습니다. 따라서 더 이상 share/developers 디렉터리의 문서에 액세스할 필요가 없으므로 관리자인 John은 Zhang을 Managers 그룹에서 Developers 그룹으로 옮겼습니다. 이처럼 간단한 재할당만으로 Managers 그룹에 허가된 모든 권한이 자동으로 Zhang에게 부여되고, 더 이상 share/developers 디렉터리의 데이터에서는 액세스하지 못하게 됩니다.

타사 통합

기업은 종종 파트너업체와 컨설턴트, 계약자들과 작업합니다. Example Corp는 Widget Company라고 하는 파트너가 있으며, 이 Widget Company의 직원인 Shirley에게 Example Corp에서 사용하는 버킷에 데이터를 추가할 권한을 부여해야 합니다. John은 WidgetCo라는 그룹과 shirley라는 사용자를 생성하고 Shirley를 WidgetCo 그룹에 추가했습니다. John은 또한 example_partner_bucket이라는 Shirley 전용 버킷을 생성했습니다.

John은 기존 정책을 업데이트하거나 새 정책을 추가하여 Widget Company 파트너에게 적절한 권한을 부여할 수 있습니다. 예를 들어 John은 WidgetCo 그룹의 구성원에게는 쓰기 이외의 모든 작업을 사용할 권한을 거부하는 새 정책을 생성할 수 있습니다. 모든 사용자에게 광범위한 Amazon S3 작업에 대해 액세스 권한을 부여하는 정책이 있을 경우에만 이 정책이 필요합니다.

IAM 콘솔 및 로그인 페이지

AWS Management 콘솔에서는 웹을 기반으로 AWS 서비스를 관리할 수 있습니다. 콘솔에 로그인하여 내 계정의 AWS 서비스를 생성하고, 조회하며, 작업을 수행할 수 있습니다. 이러한 작업에는 Amazon EC2 인스턴스 및 Amazon RDS 데이터베이스 시작/중지, Amazon DynamoDB 테이블 생성, IAM 사용자 생성 등이 포함됩니다.

Amazon Web Services(AWS) 계정을 처음 생성하면 전체 AWS 서비스 및 계정 리소스에 대한 완전한 액세스 권한을 지닌 단일 로그인 자격 증명으로 시작합니다. 이 자격 증명은 AWS 계정 루트 사용자라고 하며, 계정을 생성할 때 사용한 이메일 주소와 암호로 로그인하여 액세스합니다.

Important

일상적인 작업, 심지어 관리 작업의 경우에도 루트 사용자를 사용하지 마실 것을 강력히 권장합니다. 대신, [IAM 사용자를 처음 생성할 때만 루트 사용자를 사용하는 모범 사례 \(p. 44\)](#)를 준수하십시오. 그런 다음 루트 사용자 자격 증명은 안전하게 보관하다가 몇몇 계정 및 서비스 관리 작업을 수행할 때만 사용하십시오. 루트 사용자로 로그인해야 하는 작업을 보려면 [루트 사용자가 필요한 AWS 작업을 참조하십시오](#). 일상적 사용을 위해 관리자를 설정하는 방법에 대한 자습서는 [첫 번째 IAM 관리자 및 그룹 생성 \(p. 17\)](#) 단원을 참조하십시오.

이 단원에서는 AWS Management 콘솔 로그인 페이지에 대한 정보를 제공합니다. 내 계정에 IAM 사용자를 위한 고유의 로그인 URL을 생성하는 방법과 루트 사용자로 로그인하는 방법을 설명합니다.

Note

조직에 기존의 자격 증명 시스템이 있는 경우, Single Sign-On(SSO) 옵션을 만드는 것이 좋습니다. SSO는 IAM 사용자 자격 증명이 없어도 계정의 AWS Management 콘솔에 액세스할 수 있는 권한을 사용자에게 제공합니다. 또한 SSO를 사용하면 사용자가 조직의 사이트와 AWS에 따로 로그인하지 않아도 됩니다. 자세한 내용은 [연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하는 URL 생성\(사용자 지정 연동 브로커\) \(p. 188\)](#) 단원을 참조하십시오.

IAM 사용자 로그인 페이지

AWS Management 콘솔을 사용하려면 IAM 사용자는 사용자 이름과 암호 이외에 계정 ID 또는 계정 별칭을 제공해야 합니다. 관리자로서 [콘솔에서 IAM 사용자를 생성 \(p. 66\)](#)하는 경우, 사용자 이름과 계정 로그인 페이지 URL을 포함한 로그인 자격 증명을 해당 사용자에게 전송해야 합니다.

Important

IAM 사용자를 설정하는 방법에 따라 모든 사용자에게 첫 로그인용 임시 암호와 적절한 경우, MFA 디바이스를 제공합니다. 암호 및 MFA 디바이스에 대한 자세한 내용은 [암호 관리 \(p. 78\)](#) 및 [AWS에서 멀티 팩터 인증\(MFA\) 사용하기 \(p. 96\)](#) 단원을 참조하십시오.

IAM 사용을 시작하면 고유한 계정 로그인 페이지 URL이 자동으로 생성됩니다. 이 로그인 페이지를 사용하기 위해 해야 할 작업은 전혀 없습니다.

```
https://My_AWS_Account_ID.signin.aws.amazon.com/console/
```

AWS 계정 ID 번호 대신 회사 이름(또는 다른 친숙한 식별자)을 URL에 포함하려는 경우 계정 로그인 URL을 사용자 지정할 수도 있습니다. 계정 별칭 만들기에 대한 자세한 내용은 [AWS 계정 ID 및 별칭 \(p. 55\)](#) 단원을 참조하십시오.

도움말

웹 브라우저에서 계정 로그인 페이지를 위한 북마크를 만들려면 북마크 입력란에 계정의 로그인 URL을 직접 입력해야 합니다. 리디렉션은 로그인 URL을 가릴 수 있으므로 웹 브라우저 북마크 기능을 사용하지 마십시오.

언제든지 IAM 콘솔의 대시보드에서 계정 로그인 페이지의 URL을 찾을 수 있습니다.

IAM users sign-in link:

<https://my-account.signin.aws.amazon.com/console>

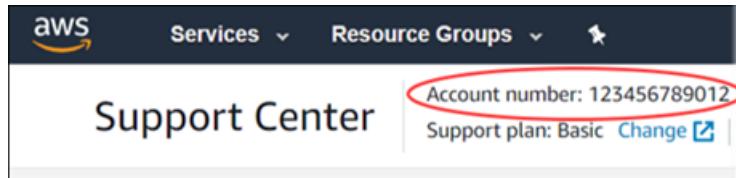
Customize | Copy Link

IAM 사용자는 다음의 일반 로그인 엔드포인트에서 로그인하고 계정 ID 또는 계정 별칭을 직접 입력할 수도 있습니다.

<https://console.aws.amazon.com/>

Note

AWS Management 콘솔에서 AWS 계정 ID 번호를 검색하려면 오른쪽 상단에 있는 탐색 모음에서 지원을 선택한 후 지원 센터를 선택하십시오. 현재 로그인한 계정 번호(ID)는 지원 센터 제목 표시줄에 나타납니다.



사용자 편의를 위해 AWS 로그인 페이지는 브라우저 쿠키를 사용하여 IAM 사용자 이름 및 계정 정보를 기억합니다. 다음에 사용자가 AWS Management 콘솔의 아무 페이지로든 이동하면 콘솔이 쿠키를 사용하여 사용자를 사용자 로그인 페이지로 리디렉션합니다.

AWS 계정 루트 사용자 로그인 페이지

AWS 계정 이메일 주소와 암호를 사용하여 AWS Management 콘솔에 루트 사용자로 로그인합니다.

Note

이전에 IAM 사용자 (p. 63) 자격 증명으로 콘솔에 로그인한 경우, 브라우저가 이 설정을 기억하여 계정별 로그인 페이지를 열 수도 있습니다. IAM 사용자 로그인 페이지에서는 AWS 계정 루트 사용자 자격 증명으로 로그인할 수 없습니다. IAM 사용자 로그인 페이지가 나타날 경우에는 페이지 하단에 있는 [Sign in using root account credentials]를 선택하여 기본 로그인 페이지로 돌아갑니다. 기본 로그인 페이지에서 AWS 계정의 이메일 주소와 암호를 입력합니다.

AWS Management 콘솔에 대한 사용자 액세스 제어

로그인 페이지를 통해 AWS 계정에 로그인하는 사용자는 권한이 허용하는 범위까지 AWS Management 콘솔을 통해 AWS 리소스에 액세스할 수 있습니다. 다음 목록에서는 AWS Management 콘솔을 통해 AWS 계정 리소스에 대한 액세스 권한을 사용자에게 부여할 수 있는 방법을 보여 줍니다. 또한 사용자가 AWS 웹 사이트를 통해 다른 AWS 계정 기능에 액세스할 수 있는 방법도 보여 줍니다.

Note

IAM 사용은 무료입니다.

AWS Management 콘솔

AWS Management 콘솔에 액세스해야 하는 각 사용자에 대해 암호를 만듭니다. 사용자는 IAM 지원 AWS 계정 로그인 페이지를 통해 콘솔에 액세스합니다. 로그인 페이지 액세스에 대한 자세한 내용은

[IAM 콘솔 및 로그인 페이지 \(p. 53\)](#)를 참조하십시오. 암호 만들기에 대한 자세한 내용은 [암호 관리 \(p. 78\)](#)을 참조하십시오.

Amazon EC2 인스턴스, Amazon S3 버킷 등의 AWS 리소스

사용자에게 암호가 있더라도 AWS 리소스에 액세스하려면 권한이 필요합니다. 사용자를 만들 때 이 사용자에게는 기본적으로 권한이 없습니다. 사용자에게 필요한 권한을 부여하려면 해당 사용자에게 정책을 연결합니다. 같은 리소스로 같은 작업을 수행할 사용자가 많은 경우 해당 사용자를 그룹에 할당한 다음 이 그룹에 권한을 할당할 수 있습니다. 사용자 및 그룹 만들기에 대한 자세한 내용은 [자격 증명\(사용자, 그룹, 및 역할\) \(p. 61\)](#)을 참조하십시오. 권한 설정을 위한 정책 사용에 대한 자세한 내용은 [액세스 관리 \(p. 304\)](#)을 참조하십시오.

AWS 토론 포럼

누구나 [AWS 토론 포럼](#)에서 게시물을 읽을 수 있습니다. AWS 토론 포럼에 질문이나 의견을 게시하고자 하는 사용자는 자신의 사용자 이름을 사용하여 그렇게 할 수 있습니다. 사용자가 처음으로 AWS 토론 포럼에 게시하면 AWS 토론 포럼에서 해당 사용자만 사용할 별칭과 이메일 주소를 입력하라는 메시지가 표시됩니다.

AWS 계정 결제 및 사용 정보

AWS 계정 결제 및 사용 정보에 대한 액세스 권한을 사용자에게 부여할 수 있습니다. 자세한 내용은 [AWS Billing and Cost Management 사용 설명서의 결제 정보에 대한 액세스 제어](#)를 참조하십시오.

AWS 계정 프로필 정보

사용자는 계정 소유자의 AWS 계정 프로필 정보에 액세스할 수 없습니다.

AWS 계정 보안 자격 증명

사용자는 계정 소유자의 AWS 계정 보안 자격 증명에 액세스할 수 없습니다.

Note

IAM 정책은 인터페이스와 관계없이 액세스를 제어합니다. 예를 들어 AWS Management 콘솔에 액세스하기 위한 암호를 사용자에게 제공할 수 있습니다. 이렇게 하면 해당 사용자 또는 사용자가 속한 임의의 그룹에 대한 정책을 통해 해당 사용자가 AWS Management 콘솔에서 수행할 수 있는 작업이 제어됩니다. 또는 AWS에 대해 API 호출을 실행하기 위한 AWS 액세스 키를 사용자에게 제공할 수 있습니다. 이렇게 하면 인증을 위해 해당 액세스 키를 사용하는 라이브러리 또는 클라이언트를 통해 사용자가 호출할 수 있는 작업이 정책을 통해 제어됩니다.

AWS 계정 ID 및 별칭

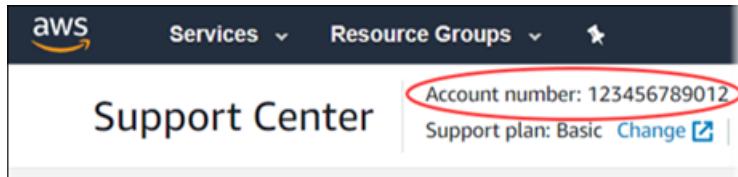
계정 별칭은 계정의 웹 주소에서 계정 ID를 대신합니다. AWS Management 콘솔, AWS CLI 또는 AWS API에서 계정 별칭을 만들고 관리할 수 있습니다.

주제

- [AWS 계정 ID 찾기 \(p. 55\)](#)
- [계정 별칭 정보 \(p. 56\)](#)
- [AWS 계정 별칭 만들기, 삭제 및 나열 \(p. 56\)](#)

AWS 계정 ID 찾기

AWS Management 콘솔에서 AWS 계정 ID 번호를 검색하려면 오른쪽 상단에 있는 탐색 모음에서 지원을 선택한 후 지원 센터를 선택하십시오. 현재 로그인한 계정 번호(ID)는 지원 센터 제목 표시줄에 나타납니다.



계정 별칭 정보

AWS 계정 ID 대신 회사 이름이나 기타 친숙한 식별자를 로그인 페이지의 URL에 포함하려는 경우 계정 별칭을 만들 수 있습니다. 이 섹션에서는 AWS 계정 별칭에 대한 정보를 제공하고 별칭을 만드는 데 사용하는 API 작업을 나열합니다.

로그인 페이지 URL의 형식은 기본적으로 다음과 같습니다.

```
https://Your_AWS_Account_ID.signin.aws.amazon.com/console/
```

AWS 계정 ID의 AWS 계정 별칭을 만드는 경우 로그인 페이지 URL이 다음 예제와 같습니다.

```
https://Your_Alias.signin.aws.amazon.com/console/
```

Note

AWS 계정 별칭을 만든 후에도 AWS 계정 ID를 포함하는 원래 URL은 활성 상태로 유지되며 사용할 수 있습니다.

도움말

웹 브라우저에서 계정의 로그인 페이지를 위한 북마크를 만들려면 북마크 입력란에 로그인 URL을 직접 입력해야 합니다. 웹 브라우저의 "페이지 즐겨찾기" 기능을 사용하지 마십시오.

AWS 계정 별칭 만들기, 삭제 및 나열

AWS Management Console, IAM API 또는 명령줄 인터페이스를 사용하여 AWS 계정 별칭을 만들거나 삭제할 수 있습니다.

Important

- AWS 계정은 별칭을 하나만 가질 수 있습니다. AWS 계정의 새 별칭을 만들면 새 별칭이 이전 별칭을 덮어쓰며 이전 별칭을 포함하는 URL이 작동하지 않습니다.
- 계정 별칭은 모든 Amazon Web Services 제품에서 고유해야 하며, 숫자, 소문자 및 하이픈만 포함해야 합니다. AWS 계정 주체 제한에 대한 자세한 내용은 [IAM 개체 및 객체에 대한 제한](#) (p. 485)을 참조하십시오.

별칭 생성 및 삭제(콘솔)

AWS Management 콘솔에서 계정 별칭을 만들고 삭제할 수 있습니다.

계정 별칭을 만들거나 제거하려면(콘솔)

- AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
- 탐색 창에서 대시보드를 선택합니다.
- IAM users sign-in link(IAM 사용자 로그인 링크)를 찾아 링크 오른쪽에 있는 사용자 지정을 선택합니다.
- 별칭에 사용할 이름을 입력한 후 예, 생성을 선택합니다.

5. 별칭을 제거하려면 사용자 지정을 선택한 다음 예, 삭제합니다.를 선택합니다. 로그인 URL에 다시 AWS 계정 ID가 사용됩니다.

별칭 만들기, 삭제 및 나열(AWS CLI)

AWS Management 콘솔 로그인 페이지 URL의 별칭을 만들려면 다음 명령을 실행합니다.

- `aws iam create-account-alias`

AWS 계정 ID 별칭을 삭제하려면 다음 명령을 실행합니다.

- `aws iam delete-account-alias`

AWS 계정 ID 별칭을 표시하려면 다음 명령을 실행합니다.

- `aws iam list-account-aliases`

별칭 만들기, 삭제 및 나열(AWS API)

AWS Management 콘솔 로그인 페이지 URL의 별칭을 만들려면 다음 연산을 호출합니다.

- `CreateAccountAlias`

AWS 계정 ID 별칭을 삭제하려면 다음 연산을 호출합니다.

- `DeleteAccountAlias`

AWS 계정 ID 별칭을 표시하려면 다음 연산을 호출합니다.

- `ListAccountAliases`

IAM 로그인 페이지에 MFA 디바이스 사용

멀티 팩터 인증(MFA) (p. 96) 디바이스로 구성된 IAM 사용자는 자신의 MFA 디바이스를 사용하여 AWS Management 콘솔에 로그인해야 합니다. 사용자가 사용자 이름과 암호를 입력하면 AWS는 해당 사용자의 계정에서 해당 사용자에게 MFA가 필요한지 여부를 확인합니다. 다음 단원은 MFA가 필요할 때 사용자가 로그인을 완료하는 방법이 나와 있습니다.

주제

- [가상 MFA 디바이스로 로그인 \(p. 57\)](#)
- [U2F 보안 키로 로그인 \(p. 58\)](#)
- [하드웨어 MFA 디바이스로 로그인 \(p. 58\)](#)

가상 MFA 디바이스로 로그인

MFA가 필요한 사용자에게는 두 번째 로그인 페이지가 나타납니다. MFA code(MFA 코드) 상자에 MFA 애플리케이션에서 제공한 숫자 코드를 입력해야 합니다.

MFA 코드가 올바르면 사용자는 AWS Management 콘솔에 액세스할 수 있습니다. 코드가 올바르지 않으면 다른 코드로 다시 시도할 수 있습니다.

가상 MFA 디바이스는 동기화되지 않을 수 있습니다. 여러 번 시도한 후에도 사용자가 AWS Management 콘솔에 로그인할 수 없으면 가상 MFA 디바이스를 동기화하라는 메시지가 표시됩니다. 사용자는 화면에 표시되는 메시지에 따라 가상 MFA 디바이스를 동기화할 수 있습니다. AWS 계정에 속한 사용자 대신 디바이스를 동기화할 수 있는 방법에 대한 자세한 내용은 [가상 및 하드웨어 MFA 디바이스 재동기화 \(p. 115\)](#)를 참조하십시오.

U2F 보안 키로 로그인

MFA가 필요한 사용자에게는 두 번째 로그인 페이지가 나타납니다. 사용자가 U2F 보안 키를 터치해야 합니다.

다른 MFA 디바이스와 달리 U2F 보안 키는 항상 동기화되어 있습니다. U2F 보안 키를 분실했거나 도난당한 경우 관리자가 비활성화할 수 있습니다. 자세한 내용은 [MFA 디바이스 비활성화\(콘솔\) \(p. 120\)](#) 단원을 참조하십시오.

U2F를 지원하는 브라우저 및 AWS를 지원하는 U2F 디바이스 정보는 [U2F 보안 키 사용에 지원되는 구성을 \(p. 106\)](#)을 확인하십시오.

하드웨어 MFA 디바이스로 로그인

MFA가 필요한 사용자에게는 두 번째 로그인 페이지가 나타납니다. MFA code(MFA 코드) 상자에 하드웨어 MFA 디바이스에서 제공한 숫자 코드를 입력해야 합니다.

MFA 코드가 올바르면 사용자는 AWS Management 콘솔에 액세스할 수 있습니다. 코드가 올바르지 않으면 다른 코드로 다시 시도할 수 있습니다.

하드웨어 MFA 디바이스는 동기화되지 않을 수 있습니다. 여러 번 시도하여 실패한 후에도 사용자가 AWS Management 콘솔에 로그인할 수 없으면 MFA 토큰 디바이스를 동기화하라는 메시지가 표시됩니다. 사용자는 화면에 표시되는 메시지에 따라 MFA 토큰 디바이스를 동기화할 수 있습니다. AWS 계정에 속한 사용자 대신 디바이스를 동기화할 수 있는 방법에 대한 자세한 내용은 [가상 및 하드웨어 MFA 디바이스 재동기화 \(p. 115\)](#)를 참조하십시오.

IAM 콘솔 검색

IAM Management Console을 탐색하여 다양한 IAM 리소스를 관리할 때 작업 대상 항목을 찾기 위해 액세스 키를 찾거나 깊숙이 중첩된 IAM 리소스로 이동해야 할 경우가 종종 있습니다. 보다 빠른 옵션은 IAM 콘솔 검색 페이지를 사용하여 계정, IAM 자격 증명(예: 사용자, 그룹, 역할, 자격 증명 공급자), 이름별 정책 등을 찾는 것입니다.

IAM 콘솔 검색 기능으로 찾을 수 있는 항목은 다음과 같습니다.

- 검색 키워드(예: 사용자, 그룹, 역할, 자격 증명 공급자, 정책)와 일치하는 IAM 엔터티 이름
- 검색 키워드와 일치하는 AWS 문서 주제 이름
- 검색 키워드와 일치하는 작업

검색 결과의 각 행은 활성 링크입니다. 예를 들어 검색 결과에서 사용자 이름을 선택할 수 있습니다. 그러면 사용자 세부 정보 페이지로 이동합니다. 또는 예를 들어 사용자 만들기 활성 링크를 선택하여 사용자 생성 페이지로 이동할 수 있습니다.

Note

액세스 키 검색에서는 검색 상자에 전체 액세스 키 ID를 입력해야 합니다. 검색 결과는 해당 키와 연결된 사용자를 보여줍니다. 여기에서 해당 사용자의 액세스 키를 관리할 수 있는 사용자 페이지로 이동할 수 있습니다.

IAM 콘솔 검색 사용

IAM의 검색 페이지를 사용하여 해당 계정과 관련된 항목을 찾습니다.

IAM 콘솔에서 항목을 검색하는 방법

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 검색을 선택합니다.
3. 검색 상자에 검색 키워드를 입력합니다.
4. 검색 결과 목록에서 링크를 선택하여 콘솔 또는 문서의 해당 부분으로 이동합니다.

IAM 콘솔 검색 결과 내 아이콘

다음 아이콘은 검색으로 찾을 수 있는 항목의 유형을 식별합니다.

아이콘	설명
	IAM 사용자
	IAM 그룹
	IAM 역할
	IAM 정책
	"사용자 만들기" 또는 "정책 연결"과 같은 작업
	키워드 delete 의 결과
	IAM 설명서

샘플 검색 문구

IAM 검색 시 다음과 같은 문구를 사용할 수 있습니다. 기울임꼴로 표시된 용어를 찾으려는 실제 IAM 사용자, 그룹, 역할, 액세스 키, 정책 또는 자격 증명 공급자의 이름으로 각각 대체합니다.

- `user_name` 또는 `group_name` 또는 `role_name` 또는 `policy_name` 또는 `identity_provider_name`
- `### #`
- `add user user_name to groups` 또는 `add users to group group_name`
- `remove user user_name from groups`

- delete `user_name` or delete `group_name` or delete `role_name`, or delete `policy_name`, or delete `identity_provider_name`
- manage access keys `user_name`
- manage signing certificates `user_name`
- users
- manage MFA for `user_name`
- manage password for `user_name`
- create role
- password policy
- edit trust policy for role `role_name`
- show policy document for role `role_name`
- attach policy to `role_name`
- create managed policy
- create user
- create group
- attach policy to `group_name`
- attach entities to `policy_name`
- detach entities to `policy_name`
- what is IAM
- how do I create an IAM user
- how do I use IAM console
- what is a user 또는 what is a group, 또는 what is a policy, 또는 what is a role, 또는 what is an identity provider

자격 증명(사용자, 그룹, 및 역할)

이 섹션에서는 AWS 계정의 사용자와 프로세스에 대한 인증을 제공하기 위해 생성하는 IAM 자격 인증을 설명합니다. 이 섹션은 또한 한 단위로 관리할 수 있는 IAM 사용자 집합인 IAM 그룹에 대해서도 설명합니다. 자격 증명은 사용자를 대표하며, 인증된 후 AWS에서 작업을 수행할 수 있는 권한을 부여받습니다. 각 자격 증명은 1개 이상의 정책 (p. 304)과 연결되어 사용자, 역할 또는 그룹 구성원이 어떤 AWS 리소스로 어떤 조건에서 어떤 작업을 할지 결정할 수 있습니다.

AWS 계정 루트 사용자 (p. 291)

Amazon Web Services(AWS) 계정을 처음 생성하는 경우에는 전체 AWS 서비스 및 계정 리소스에 대해 완전한 액세스 권한을 지닌 단일 로그인 자격 증명으로 시작합니다. 이 자격 증명은 AWS 계정 루트 사용자라고 하며, 계정을 생성할 때 사용한 이메일 주소와 암호로 로그인하여 액세스합니다.

Important

일상적인 작업, 심지어 관리 작업의 경우에도 루트 사용자를 사용하지 않는 것이 좋습니다. 대신, [IAM 사용자를 처음 생성할 때만 루트 사용자를 사용하는 모범 사례](#)를 준수하십시오. 그런 다음 루트 사용자를 안전하게 보관해 두고 몇 가지 계정 및 서비스 관리 작업을 수행할 때만 자격 증명을 사용합니다. 루트 사용자로 로그인해야 하는 작업을 보려면 [루트 사용자가 필요한 AWS 작업](#)을 참조하십시오.

IAM 사용자 (p. 63)

[IAM 사용자 \(p. 63\)](#)는 AWS에서 만드는 엔터티입니다. IAM 사용자는 IAM 사용자를 사용하여 AWS와 상호 작용하는 사람 또는 서비스를 나타냅니다. IAM 사용자의 주된 용도는 대화형 작업을 위해 AWS Management 콘솔에 로그인하고 API 또는 CLI를 사용해 AWS 서비스로 프로그래밍 방식의 요청을 보내는데 사용할 수 있는 능력을 사람들에게 제공하는 것입니다. AWS에서 사용자는 이름, AWS Management 콘솔에 로그인할 암호, 그리고 API 또는 CLI와 함께 사용할 수 있는 2개의 액세스 키로 이루어져 있습니다. IAM 사용자를 생성하는 경우, 그 사용자를 적절한 권한 정책이 연결된 그룹의 구성원으로 만들거나(이 방식을 추천함) 그 사용자에게 정책을 직접 연결하여 권한을 부여합니다. 기존 IAM 사용자의 권한을 복제하여 신규 사용자를 자동으로 같은 그룹의 구성원으로 만들고 동일한 정책을 모두 연결할 수도 있습니다.

IAM 그룹 (p. 145)

[IAM 그룹 \(p. 145\)](#)은 IAM 사용자들의 집합입니다. 그룹을 활용하면 사용자 모음에 대한 권한을 지정하여 해당 사용자에 대한 권한을 더 쉽게 관리할 수 있습니다. 예를 들어 Admins라는 그룹을 만들어 일반적으로 관리자에게 필요한 유형의 권한을 부여할 수 있습니다. 이 그룹에 할당된 권한이 이 그룹에 속하는 모든 사용자에게 자동으로 부여됩니다. 관리자 권한을 필요로 하는 새로운 사용자가 조직에 들어올 경우 해당 사용자를 이 그룹에 추가하여 적절한 권한을 할당할 수 있습니다. 마찬가지로 조직에서 직원의 업무가 바뀌면 해당 사용자의 권한을 편집하는 대신 이전 그룹에서 해당 사용자를 제거한 후 적절한 새 그룹에 추가하면 됩니다. 그룹은 [리소스 기반 정책 또는 신뢰 정책 \(p. 326\)](#)에서 Principal로 식별될 수 없기 때문에 진정한 자격 증명이 아니라는 점에 유의하십시오. 그것은 다수의 사용자에게 한 번에 정책을 연결하는 방법일 뿐입니다.

IAM역할 (p. 153)

AWS에서 자격 증명이 할 수 있는 것과 없는 것을 결정하는 권한 정책을 갖춘 자격 증명이라는 점에서 IAM 역할 (p. 153)은 사용자와 아주 유사합니다. 그러나 역할은 그와 연관된 어떤 자격 증명(암호 또는 액세스

키)도 없습니다. 역할은 한 사람과만 연관되는 것이 아니라 그 역할이 필요한 사람으면 누구든지 맡을 수 있도록 고안되었습니다. IAM 사용자는 한 가지 역할을 맡음으로써 특정 작업을 위해 다른 권한을 임시로 얻을 수 있습니다. 역할은 IAM 대신에 외부 자격 증명 공급자를 사용해 로그인하는 [연동 사용자 \(p. 161\)](#)에게 할당될 수 있습니다. AWS는 자격 증명 공급자가 전달하는 세부 정보를 사용해 연동 사용자에게 어떤 역할을 매플할지 결정합니다.

임시 자격 증명 (p. 263)

임시 자격 증명은 기본적으로 IAM 역할에 사용되지만 다른 용도로도 사용됩니다. 일반 IAM 사용자보다 제한된 권한을 갖는 임시 자격 증명을 요청할 수 있습니다. 이렇게 하면 제한된 자격 증명으로는 허용되지 않는 작업을 뜻하지 않게 수행하는 것을 방지할 수 있습니다. 임시 자격 증명의 장점은 설정한 기간이 지나면 자동으로 만료된다는 것입니다. 자격 증명의 유효 기간을 통제할 수 있습니다.

IAM 사용자를 만들어야 하는 경우(역할이 아님)

IAM 사용자는 계정에서 특정 권한을 갖는 자격 증명일 뿐이므로 자격 증명이 필요한 모든 경우를 위해 IAM 사용자를 만들 필요는 없습니다. 많은 경우 IAM 사용자와 연결된 장기 자격 증명 대신 IAM 역할과 그 역할들의 임시 보안 자격 증명을 활용할 수 있습니다.

- AWS 계정을 만들었는데 계정 내에 다른 사람이 없는 경우

AWS 계정의 루트 사용자 자격 증명을 사용하여 AWS로 작업할 수 있지만 이 방법은 권장하지 않습니다. 그 대신 자신을 위한 IAM 사용자를 만들고 AWS로 작업할 때 해당 사용자의 자격 증명을 사용하실 것을 권합니다. 자세한 내용은 [IAM 모범 사례 \(p. 43\)](#) 단원을 참조하십시오.

- 그룹에 속한 다른 사람들이 AWS 계정에서 작업해야 하며 이 그룹이 다른 자격 증명 메커니즘을 사용하고 있지 않는 경우

AWS 리소스에 액세스해야 하는 사람 각자에 대해 IAM 사용자를 만들어 각 사용자에게 적절한 권한을 할당하고 고유한 자격 증명을 부여합니다. 다수의 사용자들이 자격 증명을 공유하는 일이 절대 없도록 해주세요.

- [명령줄 인터페이스\(CLI\)](#)를 사용하여 AWS로 작업하려는 경우

CLI는 AWS를 호출하는 데 사용할 수 있는 자격 증명이 필요합니다. IAM 사용자를 만들고 필요한 CLI 명령을 실행할 권한을 해당 사용자에게 부여합니다. 그런 다음 해당 IAM 사용자에 연결된 액세스 키 자격 증명을 사용할 수 있도록 컴퓨터에서 CLI를 구성합니다.

IAM 역할을 만들어야 하는 경우(사용자가 아님)

다음 상황에서 IAM 역할을 만듭니다.

Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 실행되는 애플리케이션을 만들고 그 애플리케이션이 AWS로 요청을 보내는 경우.

IAM 사용자를 만들어 해당 사용자의 자격 증명을 애플리케이션에 전달하거나 자격 증명을 애플리케이션에 포함하지 않습니다. 그 대신에 EC2 인스턴스에 연결하는 IAM 역할을 생성해 인스턴스에서 실행되는 애플리케이션에 임시 보안 자격 증명을 부여하십시오. 자격 증명은 역할에 연결된 정책에 지정된 권한을 갖습니다. 세부 정보는 [IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여하기 \(p. 240\)](#) 단원을 참조하십시오.

휴대폰에서 실행되는 앱을 만들고 그 앱이 AWS로 요청을 보내는 경우

IAM 사용자를 만들어 앱을 통해 해당 사용자의 액세스 키를 배포하지 않습니다. 대신 [Login with Amazon](#), [Amazon Cognito](#), [Facebook](#) 또는 [Google](#)과 같은 자격 증명 공급자를 사용하여 사용자를 인증

한 다음 사용자를 IAM 역할에 매핑하십시오. 앱은 역할을 사용함으로써 역할에 연결된 정책에 의해 지정된 권한을 갖는 임시 보안 자격 증명을 얻을 수 있습니다. 자세한 내용은 다음 자료를 참조하십시오.

- [Android용 AWS Mobile SDK Developer Guide의 Amazon Cognito 개요](#)
- [AWS Mobile SDK for iOS Developer Guide의 Amazon Cognito 개요](#)
- [웹 자격 증명 연동에 대하여 \(p. 162\)](#)

회사의 사용자들이 기업 네트워크에서 인증을 받았는데 다시 로그인하지 않고도 AWS를 사용할 수 있기를 원합니다. 즉, 사용자들이 AWS로 연동되도록 허용하고 싶습니다.

IAM 사용자는 만들지 마십시오. 엔터프라이즈 자격 증명 시스템과 AWS 사이의 연동 관계를 구성하십시오. 두 가지 방법으로 수행할 수 있습니다.

- 회사의 자격 증명 시스템이 SAML 2.0과 호환된다면 회사의 자격 증명 시스템과 AWS 간에 신뢰를 구축할 수 있습니다. 자세한 내용은 [SAML 2.0 기반 연동에 대하여 \(p. 167\)](#) 단원을 참조하십시오.
- 사용자의 엔터프라이즈 자격 증명을 임시 AWS 보안 자격 증명을 제공하고 IAM 역할로 변환하는 사용자 지정 프록시 서버를 만들고 사용하십시오. 자세한 내용은 [연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하는 URL 생성\(사용자 지정 연동 브로커\) \(p. 188\)](#)를 참조하십시오.

IAM 사용자

AWS Identity and Access Management(IAM) 사용자는 AWS에서 생성하는 엔터티로서 AWS와 상호 작용하기 위해 그 엔터티를 사용하는 사람 또는 애플리케이션을 나타냅니다. AWS에서 사용자는 이름과 자격 증명으로 구성됩니다.

관리자 권한을 가진 IAM 사용자는 AWS 계정 루트 사용자와 같은 것이 아닙니다. 루트 사용자에 대한 자세한 정보는 [AWS 계정 루트 사용자 \(p. 291\)](#) 단원을 참조하십시오.

Important

애플리케이션이나 웹 사이트에 Amazon Advertising을 활성화하려는 중에 이 페이지로 오게 된 경우, [Product Advertising API 구독](#) 단원을 참조하십시오.

AWS가 IAM 사용자를 식별하는 방법

사용자를 생성하면 IAM이 그 사용자를 식별하기 위한 방법을 다음과 같이 생성합니다.

- 사용자 생성시 지정한 이름으로서 Richard 또는 Anaya와 같은 사용자가 "쉽게 알 수 있는 이름"입니다. 이 이름들은 AWS Management 콘솔에서 볼 수 있습니다.
- 사용자의 Amazon 리소스 이름(ARN)입니다. 모든 AWS 전반에 사용자를 특별하게 식별할 필요가 있는 경우 ARN을 사용합니다. 예를 들어, ARN을 사용하여 사용자를 Amazon S3 버킷에 대한 IAM 정책에서 Principal로서 지정할 수 있습니다. IAM 사용자의 ARN은 다음과 같은 모습입니다.

`arn:aws:iam::account-ID-without-hyphens:user/Richard`

- 사용자의 고유 식별자입니다. 이 ID는 사용자를 생성하기 위해 API, Windows PowerShell용 도구 또는 AWS CLI를 사용할 때만 반환됩니다. 콘솔에서는 이 ID를 볼 수 없습니다.

이 식별자에 대한 자세한 정보는 [IAM 식별자 \(p. 480\)](#) 단원을 참조하십시오.

사용자 및 자격 증명

AWS는 사용자 자격 증명에 따라 다양한 방법으로 액세스할 수 있습니다.

- [콘솔 암호 \(p. 78\)](#): 사용자가 입력해 AWS Management 콘솔과 같은 상호 작용 세션으로 로그인할 수 있는 암호.

- **액세스 키 (p. 88)**: 액세스 키 ID와 보안 액세스 키의 조합입니다. 한 사용자에게 한 번에 두 개를 지정 할 수 있습니다. 이것들은 AWS를 프로그래밍 방식으로 호출하는 데 사용될 수 있습니다. 예를 들어, AWS CLI 또는 AWS PowerShell 도구를 사용할 때 코드 또는 명령 프롬프트에 대한 API를 사용할 경우 액세스 키를 사용할 수 있습니다.
- **CodeCommit용 SSH 키 (p. 140)**: CodeCommit을 사용한 인증에 사용할 수 있는 OpenSSH 형식의 SSH 퍼블릭 키.
- **서버 인증서 (p. 141)**: 일부 AWS 서비스를 사용한 인증에 사용할 수 있는 SSL/TLS 인증서. 서버 인증서를 프로비저닝 및 관리하고 배포할 때 AWS Certificate Manager(ACM)을 사용하는 것이 좋습니다. ACM에서 지원되지 않는 리전에서 HTTPS 연결을 지원해야 하는 경우에만 IAM을 사용합니다. ACM을 지원하는 리전을 알아보려면 AWS General Reference의 [AWS Certificate Manager 리전 및 엔드포인트](#)를 참조하십시오.

IAM 사용자에게 적절한 자격 증명을 선택할 수 있습니다. AWS Management 콘솔을 사용하여 사용자를 생성할 때 최소한 콘솔 암호 또는 액세스 키를 포함하도록 선택해야 합니다. 기본적으로 AWS CLI 또는 AWS API를 사용하여 새로 생성된 IAM 사용자는 어떤 종류의 자격 증명도 보유하지 않습니다. 사용자의 요구 사항을 기반으로 IAM 사용자에 대한 자격 증명의 유형을 생성해야 합니다.

다음 옵션을 이용해 암호, 액세스 키 및 MFA 디바이스를 관리하십시오.

- **IAM 사용자 암호 관리 (p. 78)**: AWS Management 콘솔에 대한 액세스를 허용하는 암호를 생성 및 변경 합니다. 암호 정책을 최소 암호 복잡성을 적용하도록 설정 사용자에게 자신의 암호를 변경할 수 있도록 허용
- **IAM 사용자의 액세스 키 관리 (p. 88)**: 계정의 리소스에 대한 프로그래밍 방식의 액세스를 위해 액세스 키를 생성하고 업데이트합니다.
- 사용자에 대해 **멀티 팩터 인증(MFA) (p. 96)**을 활성화하여 해당 사용자 자격 증명의 보안을 강화할 수 있습니다. MFA를 사용할 경우 사용자는 두 가지 형식의 식별이 가능합니다. 먼저, 사용자는 자격 증명(암호 또는 액세스 키)의 일부분인 자격 증명을 제공합니다. 또한, 하드웨어 디바이스에서나 스마트폰 또는 태블릿의 애플리케이션을 통해서 생성된 또는 SMS 호환성 모바일 디바이스로 AWS가 보낸 임시 숫자 코드를 제공해야 합니다.
- **미사용 암호 및 액세스 키 찾기 (p. 133)**: 계정 또는 계정 내 IAM 사용자에 대한 암호 또는 액세스 키를 보유하는 사람은 누구든지 AWS 리소스에 대한 액세스 권한이 있습니다. 보안 [모범 사례](#)는 사용자에게 암호와 액세스 키가 필요하지 않을 때 그것들을 제거하는 것입니다.
- **계정의 자격 증명 보고서 다운로드 (p. 135)**: 계정의 모든 IAM 사용자와 암호, 액세스 키, MFA 디바이스를 포함하여 이들의 자격 증명 상태를 나열하는 자격 증명 보고서를 생성하고 다운로드할 수 있습니다. 암호와 액세스 키의 경우 자격 증명 보고서를 통해 암호 또는 액세스 키가 언제 마지막으로 사용되었는지 알 수 있습니다.

사용자 및 권한

기본적으로 신규 IAM 사용자는 어떤 작업도 할 수 있는 [권한 \(p. 304\)](#)이 없습니다. 사용자는 AWS 작업을 수행하거나 AWS 리소스에 액세스할 수 있는 권한이 없습니다. 개별 IAM 사용자를 두면 각 사용자에게 개별적으로 권한을 할당할 수 있다는 장점이 있습니다. 사용자 몇 명에게 관리 권한을 할당하면 이들이 AWS 리소스를 관리하고 다른 IAM 사용자까지 생성하고 관리할 수 있습니다. 그러나 대부분의 경우 사용자의 업무에 필요한 작업(AWS 작업)과 리소스로 사용자의 권한을 제한합니다.

Diego라는 사용자가 있다고 가정해 보겠습니다. IAM 사용자 Diego를 생성하면 그 사용자의 암호를 생성할 수 있습니다. 특정 Amazon EC2 인스턴스를 시작하고(GET) 정보를 Amazon RDS 데이터베이스의 테이블에서 읽을 수 있도록 IAM 사용자에게 권한을 부여합니다. 사용자를 생성하여 초기 자격 증명과 권한을 부여하는 절차는 [AWS 계정의 IAM 사용자 생성 \(p. 65\)](#) 단원을 참조하십시오. 기존 사용자에 대한 권한을 변경하는 절차는 [IAM 사용자의 권한 변경 \(p. 73\)](#) 단원을 참조하십시오. 사용자의 암호나 액세스 키를 변경하는 절차는 [암호 관리 \(p. 78\)](#) 및 [IAM 사용자의 액세스 키 관리 \(p. 88\)](#) 단원을 참조하십시오.

사용자에게 권한 경계를 추가할 수 있습니다. 권한 경계는 AWS 관리형 정책을 사용하여 자격 증명 기반 정책이 사용자 또는 역할에 부여할 수 있는 최대 권한을 제한할 수 있는 고급 기능입니다. 정책 유형 및 활용에 대한 자세한 정보는 [정책 및 권한 \(p. 305\)](#) 단원을 참조하십시오.

사용자 및 계정

각 IAM 사용자는 오직 한 개의 AWS 계정과만 연결됩니다. 사용자는 AWS 계정 내에서 정의되기 때문에 AWS에서 파일에 결제 방법을 저장해 두지 않아도 됩니다. 계정에 속한 사용자가 수행하는 모든 AWS 활동은 해당 계정으로 청구됩니다.

AWS 계정에서 보유할 수 있는 IAM 사용자 수는 제한되어 있습니다. 자세한 정보는 [IAM 개체 및 객체에 대한 제한 \(p. 485\)](#) 단원을 참조하십시오.

서비스 계정인 사용자

IAM 사용자는 연결된 자격 증명 및 권한을 지닌 IAM의 리소스입니다. IAM 사용자는 자격 증명을 사용하여 AWS 요청을 생성하는 사용자 또는 애플리케이션을 나타낼 수 있습니다. 이를 일반적으로 서비스 계정이라 합니다. 애플리케이션에 있는 IAM 사용자의 장기 자격 증명을 사용하기로 선택한 경우 액세스 키를 애플리케이션 코드에 직접 포함시키지 마십시오. AWS SDK 및 AWS Command Line Interface를 사용하면 코드에서 유지할 필요가 없도록 알려진 위치에 액세스 키를 추가할 수 있습니다. 자세한 정보는 AWS General Reference의 [적절하게 IAM 사용자 액세스 키 관리](#) 단원을 참조하십시오. 또는 모범 사례로서 [장기 액세스 키 대신 임시 보안 자격 증명\(IAM 역할\)](#)을 사용할 수 있습니다.

AWS 계정의 IAM 사용자 생성



AWS 계정에서 하나 이상의 IAM 사용자를 만들 수 있습니다. 팀에 새로 합류하는 사람이 있거나 AWS에 대한 API 호출이 필요한 새 애플리케이션을 생성할 때 IAM 사용자를 생성할 수 있습니다.

Important

애플리케이션이나 웹 사이트에 Amazon 광고를 설정하는 동안 이 페이지로 오게 된 경우, [Product Advertising API 개발자 되기](#) 단원을 참조하십시오.

IAM 콘솔에서 이 페이지로 이동했을 경우 로그인을 했더라도 계정에 IAM 사용자가 포함되지 않을 수 있습니다. 역할을 사용하거나, 임시 자격 증명으로 로그인하여 AWS 계정 루트 사용자로 로그인할 수 있습니다. 이러한 IAM 자격 증명에 대한 자세한 내용은 [자격 증명\(사용자, 그룹, 및 역할\) \(p. 61\)](#) 단원을 참조하십시오.

주제

- [IAM 사용자 생성\(콘솔\) \(p. 66\)](#)
- [IAM 사용자 만들기\(AWS CLI\) \(p. 68\)](#)
- [IAM 사용자 만들기\(AWS API\) \(p. 68\)](#)

다음 단계에 따라 사용자를 생성하고 사용자가 작업을 수행할 수 있습니다.

1. AWS Management 콘솔, AWS CLI, Windows PowerShell용 도구 또는 AWS API 작업을 사용하여 사용자를 생성합니다. AWS Management 콘솔에서 사용자를 생성하면, 자신의 선택에 따라 1~4단계는 자동으로 처리됩니다. 프로그래밍 방식으로 사용자를 생성하는 경우, 각 단계를 개별적으로 수행해야 합니다.
2. 사용자에게 필요한 액세스 유형에 따라 사용자의 자격 증명을 생성합니다.
 - 프로그래밍 방식으로 액세스: IAM 사용자가 API를 호출해야 하거나, AWS CLI 또는 Windows PowerShell용 도구를 사용해야 할 수 있습니다. 이 경우 해당 사용자의 액세스 키를 만드십시오(액세스 키 ID 및 보안 액세스 키).
 - AWS Management 콘솔 액세스: 사용자가 AWS Management 콘솔에서 액세스해야 할 경우, [에서 해당 사용자의 암호를 생성합니다 \(p. 82\)](#).

사용자가 필요한 자격 증명만 생성하는 것이 가장 좋습니다. 예를 들어, AWS Management 콘솔을 통해서만 액세스해야 하는 사용자에게는 액세스 키를 생성해서는 안 됩니다.

3. 해당 사용자를 하나 이상의 그룹에 추가하여 필요한 작업을 수행할 수 있는 권한을 부여합니다. 권한 정책을 사용자에게 직접 연결하여 권한을 부여할 수 있습니다. 하지만, 사용자를 그룹에 추가한 후 그 그룹에

연결된 정책을 통해 정책과 권한을 관리하는 것이 좋습니다. [권한 경계 \(p. 317\)](#)를 사용하여 일반적이지 않지만 사용자에게 있는 권한을 제한할 수 있습니다.

4. (선택 사항) 태그를 연결하여 메타데이터를 사용자에게 추가합니다. IAM에서의 태그 사용에 대한 자세한 내용은 [IAM 엔터티 태그 지정 \(p. 259\)](#) 단원을 참조하십시오.
5. 사용자에게 필요한 로그인 정보를 제공합니다. 여기에는 암호를 비롯해 사용자가 자격 증명을 제공하는 계정 로그인 웹 페이지의 콘솔 URL이 포함됩니다. 자세한 내용은 [IAM 사용자가 AWS에 로그인하는 방법 \(p. 69\)](#) 단원을 참조하십시오.
6. (선택 사항) 사용자에 대한 [멀티 팩터 인증\(MFA\) \(p. 96\)](#)을 구성합니다. MFA의 경우, 사용자가 AWS Management 콘솔에 로그인할 때마다 일회용 코드를 입력해야 합니다.
7. (선택 사항) 사용자에게 자신의 보안 자격 증명을 관리할 권한을 부여합니다. (기본적으로 사용자는 자신의 자격 증명을 관리할 권리가 없습니다.) 자세한 내용은 [IAM 사용자에게 자신의 암호 변경 허용하기 \(p. 85\)](#) 단원을 참조하십시오.

사용자를 생성하기 위해 필요한 권한에 대한 자세한 내용은 [IAM 리소스에 액세스하는 데 필요한 권한 \(p. 443\)](#) 단원을 참조하십시오.

IAM 사용자 생성(콘솔)

AWS Management 콘솔을 사용하여 IAM 사용자를 생성할 수 있습니다.

한 명 이상의 IAM 사용자를 생성하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자와 Add user(사용자 추가)를 차례로 선택합니다.
3. 신규 사용자의 사용자 이름을 입력합니다. 이것은 AWS에 로그인할 때 사용하는 이름입니다. 하나 이상의 사용자를 동시에 추가하려면, 추가하는 각 사용자에 대해 Add another user(다른 사용자 추가)를 선택한 후 사용자 이름을 입력합니다. 한 번에 최대 10명까지 사용자를 추가할 수 있습니다.

Note

사용자 이름에는 최대 64개의 알파벳, 숫자 및 더하기(+), 등호(=), 쉼표(,), 마침표(.) 및 앤(@) 그리고 하이픈(-) 조합을 사용할 수 있습니다. 이름은 계정 내에서 고유해야 합니다. 대소문자는 구별하지 않습니다. 예를 들어 "TESTUSER"와 "testuser"라는 두 사용자를 만들 수는 없습니다. IAM 엔터티 관련 제한에 대한 자세한 내용은 [IAM 개체 및 객체에 대한 제한 \(p. 485\)](#) 단원을 참조하십시오.

4. 이 사용자 세트에게 부여할 액세스 권한의 유형을 선택합니다. 프로그래밍 방식 액세스나 AWS Management 콘솔에 대한 액세스 또는 둘 다를 선택할 수 있습니다.
 - 사용자가 API, AWS CLI, 또는 Windows PowerShell용 도구에 대한 액세스 권한이 필요한 경우, Programmatic access(프로그래밍 방식 액세스)를 선택합니다. 이렇게 하면 각 사용자에 대한 액세스 키가 생성됩니다. 최종(Final) 페이지에 이르면 액세스 키를 보거나 다운로드할 수 있습니다.
 - 사용자에게 AWS Management 콘솔에 대한 액세스 권한이 필요한 경우, AWS Management 콘솔 access(콘솔 액세스)를 선택합니다. 이렇게 하면 각 신규 사용자에 대한 암호가 생성됩니다.
- a. 콘솔 암호(Console password)의 경우 다음 중 하나를 선택합니다.
 - Autogenerated password(자동 생성된 비밀 번호). 각 사용자는 유효한 계정 암호 정책(있는 경우)에 따라 임의로 생성되는 암호를 받습니다. Final(최종) 페이지에 이르면 암호를 보거나 다운로드 할 수 있습니다.
 - Custom password(사용자 지정 비밀 번호). 입력란에 입력하는 암호가 각 사용자에게 할당됩니다.
- b. (선택 사항) 암호 재설정 필요를 선택하여 사용자가 처음 로그인할 때 의무적으로 암호를 변경하도록 설정하는 것이 바람직합니다.

Note

Allow users to change their own password(사용자 자신의 암호 변경 허용)으로 설정된 계정 수준 암호 정책을 활성화하지 않은 경우, Require password reset(암호 재설정 필요)를 선택하면 자신의 암호를 변경할 수 있는 권한을 부여하는 [IAMUserChangePassword](#)라는 AWS 관리형 정책이 신규 사용자에게 자동 연결됩니다.

5. Next: Permissions(다음: 권한)을 선택합니다.
6. 권한 설정 페이지에서 이 신규 사용자 세트에 권한을 할당하는 방식을 지정합니다. 다음 세 가지 옵션 중 하나를 선택합니다.
 - Add user to group(그룹에 사용자 추가). 이미 권한 정책을 보유한 하나 이상의 그룹에 사용자를 할당하고자 하는 경우, 이 옵션을 선택합니다. IAM에 계정 그룹의 목록이 연결된 정책과 함께 표시됩니다. 기존의 보안 그룹을 한 개 이상 선택하거나 그룹 생성을 선택하여 새 그룹을 만들 수 있습니다. 자세한 내용은 [IAM 사용자의 권한 변경 \(p. 73\)](#) 단원을 참조하십시오.
 - Copy permissions from existing user(기존 사용자에서 권한 복사). 이 옵션을 선택하여 그룹 멤버십, 연결된 관리형 정책, 포함된 인라인 정책 및 기존 [권한 경계 \(p. 317\)](#)를 기존 사용자에게 신규 사용자로 모두 복사합니다. IAM는 계정에 속한 사용자 목록을 보여줍니다. 보유한 권한이 새로운 사용자의 요구 사항과 가장 근접하는 사용자를 선택합니다.
 - Attach existing policies to user directly(기존 정책을 사용자에게 직접 연결). 이 옵션을 선택하여 계정의 AWS 관리형 또는 사용자 관리형 정책 목록을 봅니다. 신규 사용자에게 연결하려는 정책을 선택하거나 정책 생성을 선택하여 새 브라우저 탭을 열고 완전히 새로운 정책을 생성합니다. 자세한 내용은 [IAM 정책 만들기\(콘솔\) \(p. 378\)](#) 절차의 4단계 단원을 참조하십시오. 정책을 생성하면 탭을 닫고 원래 탭으로 돌아와 신규 사용자에게 정책을 추가합니다. 그 대신에 그룹에 정책을 연결한 다음, 사용자들을 적절한 그룹의 구성원으로 만드는 것이 바람직한 [도법 사례 \(p. 44\)](#)입니다.
7. (선택 사항) [권한 경계 \(p. 317\)](#)로서 설정됨. 이는 고급 기능입니다.

Set permissions boundary(권한 경계 설정) 섹션을 열고 Use a permissions boundary to control the maximum user permissions(최대 사용자 권한을 관리하기 위한 권한 경계 사용)을 선택합니다. IAM은 계정의 AWS 관리형 또는 사용자 관리형 정책 목록을 보여줍니다. 권한 경계를 사용하기 위한 정책을 선택하거나 정책 생성을 선택하여 새 브라우저 탭을 열고 완전히 새로운 정책을 생성합니다. 자세한 내용은 [IAM 정책 만들기\(콘솔\) \(p. 378\)](#) 절차의 4단계 단원을 참조하십시오. 정책을 생성하면 탭을 닫고 원래 탭으로 돌아와 권한 경계를 사용하기 위한 정책을 선택합니다.

8. 다음: 태그 지정을 선택합니다.
9. (선택 사항) 태그를 키-값 페어로 연결하여 메타데이터를 사용자에게 추가합니다. IAM에서의 태그 사용에 대한 자세한 내용은 [IAM 엔터티 태그 지정 \(p. 259\)](#) 단원을 참조하십시오.
10. Next: Review(다음: 검토)를 선택하여 이 시점까지 한 선택을 모두 확인합니다. 계속 진행할 준비가 되었으면 Create user를 선택합니다.
11. 사용자의 액세스 키(액세스 키 ID와 보안 액세스 키)를 보려면 보고 싶은 각 암호와 액세스 키 옆에 있는 표시를 선택합니다. 액세스 키를 저장하려면 Download .csv(csv 다운로드)를 선택한 후 안전한 위치에 파일을 저장합니다.

Important

보안 액세스 키는 이 때만 확인 및 다운로드가 가능하기 때문에 사용자에게 AWS API를 사용하도록 하려면 이 정보를 제공해야 합니다. 사용자의 새 액세스 키 ID와 보안 액세스 키를 안전한 장소에 보관하십시오. 이 단계가 지난 후에는 보안 키에 다시 액세스할 수 없습니다.

12. 각 사용자에게 해당 자격 증명을 제공합니다. 최종 페이지에서 각 사용자 옆에 있는 Send email(이메일 전송)을 선택합니다. 로컬 메일 클라이언트는 사용자 지정을 거쳐 발송할 수 있는 초안 형태로 열립니다. 이메일 템플릿에는 각 사용자에 대한 세부 정보가 다음과 같이 포함되어 있습니다.
 - 사용자 이름
 - 계정 로그인 페이지의 URL. 다음 예를 사용하여 정확한 계정 ID 번호 또는 계정 별칭으로 대체합니다.

`https://AWS-account-ID or alias.signin.aws.amazon.com/console`

자세한 내용은 [IAM 사용자가 AWS에 로그인하는 방법 \(p. 69\)](#) 단원을 참조하십시오.

Important

생성된 이메일에는 사용자 암호가 포함되어 있지 않습니다. 고객에게 보내는 이메일은 소속된 조직의 보안 지침을 준수하는 방식으로 제공되어야 합니다.

IAM 사용자 만들기(AWS CLI)

AWS CLI를 사용하여 IAM 사용자를 생성할 수 있습니다.

IAM 사용자를 생성하려면(AWS CLI)

1. 사용자를 생성합니다.
 - [aws iam create-user](#)
2. (선택 사항) 사용자에게 AWS Management 콘솔에 대한 액세스 권한 부여. 이를 위해서는 암호가 필요합니다. 또한 사용자에게 [계정 로그인 페이지의 URL \(p. 69\)](#)도 제공해야 합니다.
 - [aws iam create-login-profile](#)
3. (선택 사항) 사용자에게 프로그래밍 방식 액세스 권한 부여. 이를 위해서는 액세스 키가 필요합니다.
 - [aws iam create-access-key](#)
 - Windows PowerShell용 도구: [New-IAMAccessKey](#)
 - IAM API: [CreateAccessKey](#)

Important

보안 액세스 키는 이 때만 확인 및 다운로드가 가능하기 때문에 사용자에게 AWS API를 사용하도록 하려면 이 정보를 제공해야 합니다. 사용자의 새 액세스 키 ID와 보안 액세스 키를 안전한 장소에 보관하십시오. 이 단계가 지난 후에는 보안 키에 다시 액세스할 수 없습니다.

4. 사용자를 하나 이상의 그룹에 추가합니다. 지정하는 그룹에는 사용자에게 적절한 권한을 부여하는 연결된 정책이 있어야 합니다.
 - [aws iam add-user-to-group](#)
5. (선택 사항) 사용자 권한을 정의한 정책을 사용자에게 추가합니다. 주의: 사용자에게 직접 정책을 추가하는 대신 그룹에 사용자를 추가하고 그 그룹에 정책을 추가하여 사용자 권한을 관리하시는 것이 좋습니다.
 - [aws iam attach-user-policy](#)
6. (선택 사항) 태그를 연결하여 사용자 지정 속성을 사용자에게 추가합니다. 자세한 내용은 [IAM 엔터티에 대한 태그 관리\(AWS CLI 또는 AWS API\) \(p. 262\)](#) 단원을 참조하십시오.
7. (선택 사항) 사용자에게 자신의 보안 자격 증명을 관리할 수 있는 권한을 부여합니다. 자세한 내용은 [AWS: MFA 인증 IAM 사용자가 My Security Credentials\(내 보안 자격 증명\) 페이지에서 자신의 자격 증명을 관리할 수 있도록 허용합니다. \(p. 343\)](#) 단원을 참조하십시오.

IAM 사용자 만들기(AWS API)

AWS API를 사용하여 IAM 사용자를 생성할 수 있습니다.

(AWS API)에서 IAM 사용자를 생성하려면

1. 사용자를 생성합니다.

- [CreateUser](#)
 - 2. (선택 사항) 사용자에게 AWS Management 콘솔에 대한 액세스 권한 부여. 이를 위해서는 암호가 필요합니다. 또한 사용자에게 [계정 로그인 페이지의 URL \(p. 69\)](#)도 제공해야 합니다.
 - [CreateLoginProfile](#)
 - 3. (선택 사항) 사용자에게 프로그래밍 방식 액세스 권한 부여. 이를 위해서는 액세스 키가 필요합니다.
 - [CreateAccessKey](#)
- Important**
- 보안 액세스 키는 이 때만 확인 및 다운로드가 가능하기 때문에 사용자에게 AWS API를 사용하도록 하려면 이 정보를 제공해야 합니다. 사용자의 새 액세스 키 ID와 보안 액세스 키를 안전한 장소에 보관하십시오. 이 단계가 지난 후에는 보안 키에 다시 액세스할 수 없습니다.
4. 사용자를 하나 이상의 그룹에 추가합니다. 지정하는 그룹에는 사용자에게 적절한 권한을 부여하는 연결된 정책이 있어야 합니다.
 - [AddUserToGroup](#)
 5. (선택 사항) 사용자 권한을 정의한 정책을 사용자에게 추가합니다. 주의: 사용자에게 직접 정책을 추가하는 대신 그룹에 사용자를 추가하고 그 그룹에 정책을 추가하여 사용자 권한을 관리하시는 것이 좋습니다.
 - [AttachUserPolicy](#)
 6. (선택 사항) 태그를 연결하여 사용자 지정 속성을 사용자에게 추가합니다. 자세한 내용은 [IAM 엔터티에 대한 태그 관리\(AWS CLI 또는 AWS API\) \(p. 262\)](#) 단원을 참조하십시오.
 7. (선택 사양) 사용자에게 자신의 보안 자격 증명을 관리할 수 있는 권한을 부여합니다. 자세한 내용은 [AWS: MFA 인증 IAM 사용자가 My Security Credentials\(내 보안 자격 증명\) 페이지에서 자신의 자격 증명을 관리할 수 있도록 허용합니다. \(p. 343\)](#) 단원을 참조하십시오.

IAM 사용자가 AWS에 로그인하는 방법

IAM 사용자로 AWS Management 콘솔에 로그인하려면 사용자 이름과 암호 이외에 계정 ID 또는 계정 별칭을 제공해야 합니다. 관리자가 [콘솔에서 IAM 사용자를 만든 경우 \(p. 66\)](#), 계정 ID 또는 계정 별칭이 포함된 계정 로그인 페이지 URL과 사용자 이름 등 로그인 자격 증명이 전송되었어야 합니다.

`https://My_AWS_Account_ID.signin.aws.amazon.com/console/`

도움말

웹 브라우저에서 계정 로그인 페이지를 위한 북마크를 만들려면 북마크 입력란에 계정의 로그인 URL을 직접 입력해야 합니다. 리디렉션은 로그인 URL을 가릴 수 있으므로 웹 브라우저 북마크 기능을 사용하지 마십시오.

다음의 일반 로그인 엔드포인트에서 로그인하고 계정 ID 또는 계정 별칭을 직접 입력할 수도 있습니다.

`https://console.aws.amazon.com/`

사용자 편의를 위해 AWS 로그인 페이지는 브라우저 쿠키를 사용하여 IAM 사용자 이름 및 계정 정보를 기억합니다. 다음에 사용자가 AWS Management 콘솔의 아무 페이지로든 이동하면 콘솔이 쿠키를 사용하여 사용자를 사용자 로그인 페이지로 리디렉션합니다.

IAM 사용자 ID에 첨부되는 정책에서 관리자가 지정하는 AWS 리소스에만 액세스할 수 있습니다. 콘솔에서 작업하려면 AWS 리소스 나열 및 생성 등 콘솔이 수행하는 작업을 수행할 권한이 있어야 합니다. 자세한 내용은 [액세스 관리 \(p. 304\)](#) 및 [IAM 자격 증명 기반 정책 예제 \(p. 341\)](#) 단원을 참조하십시오.

Note

조직에 기존의 자격 증명 시스템이 있는 경우, Single Sign-On(SSO) 옵션을 만드는 것이 좋습니다. SSO는 IAM 사용자 자격 증명이 없어도 계정의 AWS Management 콘솔에 액세스할 수 있는 권한을 사용자에게 제공합니다. 또한 SSO를 사용하면 사용자가 조직의 사이트와 AWS에 따로 로그인하지 않아도 됩니다. 자세한 내용은 [연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하는 URL 생성\(사용자 지정 연동 브로커\)](#) (p. 188) 단원을 참조하십시오.

CloudTrail의 로그인 세부 정보 기록

CloudTrail에서 로그인 이벤트를 사용자 로그에 기록하도록 설정할 경우 CloudTrail에서 이벤트를 기록할 위치를 어떻게 선택하는지 잘 이해할 필요가 있습니다.

- 사용자가 콘솔에 직접 로그인할 경우 선택한 서비스 콘솔이 리전을 지원하는지 여부를 기준으로 글로벌 또는 리전 로그인 종단점으로 리디렉션됩니다. 예를 들어 메인 콘솔 홈 페이지는 리전을 지원합니다. 따라서 다음 URL에 로그인할 경우

```
https://alias.signin.aws.amazon.com/console
```

<https://us-east-2.signin.aws.amazon.com>과 같은 리전 로그인 종단점으로 리디렉션되어 사용자의 리전 로그에 리전 CloudTrail 로그 항목이 기록됩니다.

반면 Amazon S3 콘솔은 리전을 지원하지 않습니다. 따라서 다음 URL에 로그인할 경우

```
https://alias.signin.aws.amazon.com/console/s3
```

AWS가 사용자의 <https://signin.aws.amazon.com>의 글로벌 로그인 종단점으로 리디렉션하여 글로벌 CloudTrail 로그 항목이 기록됩니다.

- 다음과 같은 URL 구문을 사용하여 리전이 활성화된 메인 콘솔 홈 페이지에 로그인하면 특정 리전 로그인 종단점을 수동으로 요청할 수 있습니다.

```
https://alias.signin.aws.amazon.com/console?region=ap-southeast-1
```

AWS가 사용자를 ap-southeast-1 리전 로그인 종단점으로 리디렉션하고 리전 CloudTrail 로그 이벤트가 발생합니다.

CloudTrail 및 IAM에 대한 자세한 내용은 [AWS CloudTrail로 IAM 이벤트 로깅](#) 단원을 참조하십시오.

계정을 통해 사용자가 작업하기 위해 프로그래밍 방식의 액세스가 필요할 경우에는 [액세스 키 관리\(콘솔\)](#) (p. 90)에 기술된 대로 각 사용자의 액세스 키 페어(액세스 키 ID와 보안 액세스 키)를 생성할 수 있습니다.

IAM 사용자 관리

Amazon Web Services는 AWS 계정에 속한 IAM 사용자들을 관리할 수 있는 다양한 도구를 제공합니다. 계정 또는 그룹에 속한 IAM 사용자를 나열하거나 한 사용자가 속한 모든 그룹을 나열할 수 있습니다. IAM 사용자의 이름을 변경하거나 경로를 변경할 수 있습니다. AWS 계정에서 IAM 사용자를 삭제할 수도 있습니다.

IAM 사용자에 대한 관리형 정책의 추가, 변경, 제거에 대한 자세한 내용은 [IAM 사용자의 권한 변경](#) (p. 73) 단원을 참조하십시오. IAM 사용자에 대한 인라인 정책 관리에 대한 자세한 내용은 [IAM 자격 증명 권한 추가 및 제거](#) (p. 391), [IAM 정책 편집](#) (p. 402), [IAM 정책 삭제](#) (p. 406) 단원을 참조하십시오. 인라인 정책보다는 관리형 정책을 사용하는 것이 좋습니다.

주제

- 사용자 액세스 보기 (p. 71)
- IAM 사용자 표시 (p. 71)
- IAM 사용자 이름 바꾸기 (p. 71)
- IAM 사용자 삭제 (p. 72)

사용자 액세스 보기

사용자를 삭제하기 전에 최근 서비스 수준 활동을 검토해야 합니다. 이 기능은 사용 중인 보안 주체(사람 또는 애플리케이션)의 액세스 권한을 제거하지 않으려는 경우 중요합니다. 서비스에서 마지막으로 액세스한 데이터 보기에 대한 자세한 내용은 [서비스에서 마지막으로 액세스한 데이터를 사용하여 권한 줄이기 \(p. 409\)](#) 단원을 참조하십시오.

IAM 사용자 표시

AWS 계정 또는 특정 IAM 그룹에 속한 IAM 사용자, 그리고 한 사용자가 속한 모든 그룹을 표시할 수 있습니다. 사용자 표시에 필요한 권한에 대한 자세한 내용은 [IAM 리소스에 액세스하는 데 필요한 권한 \(p. 443\)](#) 단원을 참조하십시오.

계정에 속한 모든 사용자를 표시하려면

- AWS Management 콘솔: 탐색 창에서 사용자를 선택합니다. 콘솔에 AWS 계정에 속한 사용자가 표시됩니다.
- AWS CLI: `aws iam list-users`
- AWS API: `ListUsers`

특정 그룹에 속한 사용자를 표시하려면

- AWS Management 콘솔: 탐색 창에서 그룹을 선택하고, 그룹 이름을 선택한 후 사용자 탭을 선택합니다.
- AWS CLI: `aws iam get-group`
- AWS API: `GetGroup`

사용자가 속한 모든 그룹을 표시하려면

- AWS Management 콘솔: 탐색 창에서 사용자를 선택하고, 사용자 이름을 선택한 후 그룹 탭을 선택합니다.
- AWS CLI: `aws iam list-groups-for-user`
- AWS API: `ListGroupsForUser`

IAM 사용자 이름 바꾸기

사용자의 이름 또는 경로를 변경하려면 AWS CLI, Windows PowerShell용 도구 또는 AWS API를 사용해야 합니다. 콘솔에서는 사용자의 이름을 변경할 수 있는 옵션이 없습니다. 사용자의 이름을 변경하기 위해 필요한 권한에 대한 자세한 내용은 [IAM 리소스에 액세스하는 데 필요한 권한 \(p. 443\)](#) 단원을 참조하십시오.

사용자의 이름 또는 경로를 변경하면 다음과 같이 진행됩니다.

- 사용자에 연결된 정책은 이름이 변경되어도 계속 유지됩니다.
- 사용자는 전과 동일한 그룹에 새 이름으로 표시됩니다.
- 사용자의 고유 ID는 전과 같습니다. 고유 ID에 대한 자세한 내용은 [고유 ID \(p. 483\)](#) 단원을 참조하십시오.

- 사용자를 보안 주체로 참조(해당 사용자에게 액세스가 부여됨)하는 리소스 또는 역할 정책은 새 이름 또는 경로를 사용하도록 자동 업데이트됩니다. 예를 들어 Amazon SQS의 대기열 기반 정책 또는 Amazon S3의 리소스 기반 정책은 자동 업데이트되어 새 이름과 경로를 사용합니다.

사용자를 리소스로 참조하는 정책은 새 이름 또는 경로를 사용하도록 IAM에서 자동 업데이트하지 않으므로 수동으로 업데이트해야 합니다. 예를 들어 사용자 Richard에게 자신의 보안 자격 증명을 관리하는 정책이 연결되어 있을 경우 관리자가 Richard에서 Rich로 이름을 변경하면 다음과 같이 리소스가 변경되도록 관리자가 정책도 업데이트해야 합니다.

```
arn:aws:iam::111122223333:user/division_abc/subdivision_xyz/Richard
```

다음으로 업데이트:

```
arn:aws:iam::111122223333:user/division_abc/subdivision_xyz/Rich
```

마찬가지로 경로를 변경할 경우에도 관리자가 사용자에 대한 새 경로를 반영하도록 정책을 업데이트해야 합니다.

사용자의 이름을 바꾸려면

- AWS CLI: [aws iam update-user](#)
- AWS API: [UpdateUser](#)

IAM 사용자 삭제

퇴사자가 생길 경우 계정에서 IAM 사용자를 삭제할 수 있습니다. 사용자가 잠시 회사에 나오지 않는 경우에는 AWS 계정에서 해당 사용자를 완전히 삭제하는 대신 사용자의 자격 증명을 비활성화할 수 있습니다. 이렇게 하면 부재 중 해당 사용자가 AWS 계정의 리소스에 액세스하는 것을 막고 나중에 해당 사용자를 다시 활성화할 수 있습니다.

자격 증명 비활성화에 대한 자세한 내용은 [IAM 사용자의 액세스 키 관리 \(p. 88\)](#) 단원을 참조하십시오. 사용자를 삭제하기 위해 필요한 권한에 대한 자세한 내용은 [IAM 리소스에 액세스하는 데 필요한 권한 \(p. 443\)](#) 단원을 참조하십시오.

주제

- [IAM 사용자 삭제\(콘솔\) \(p. 72\)](#)
- [IAM 사용자 삭제\(AWS CLI\) \(p. 73\)](#)

IAM 사용자 삭제(콘솔)

AWS Management 콘솔을 사용하여 IAM 사용자를 삭제하면 IAM에서 자동으로 다음 정보를 삭제합니다.

- 해당 사용자
- 모든 그룹 멤버십, 즉 속해 있던 모든 IAM 그룹에서 사용자가 제거됨
- 사용자와 연결된 모든 암호
- 사용자에게 속한 모든 액세스 키
- 사용자에게 포함된 모든 인라인 정책(그룹 권한을 통해 사용자에게 적용되는 정책은 영향을 받지 않음)

Note

사용자를 삭제하면 해당 사용자에게 연결된 관리형 정책이 해당 사용자에게서 분리됩니다. 사용자를 삭제해도 관리형 정책은 삭제되지 않습니다.

- 연결된 모든 MFA 디바이스

IAM 사용자를 삭제하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택한 후 삭제하려는 역할 옆에 있는 확인란(사용자 이름이나 행 아님)을 선택합니다.
3. 페이지 상단에서 사용자 삭제를 선택합니다.
4. 확인 대화 상자에 서비스에서 마지막으로 액세스한 데이터가 로드될 때까지 기다렸다 데이터를 검토합니다. 이 대화 상자는 선택한 각 사용자가 언제 마지막으로 AWS 서비스에 액세스했는지 보여줍니다. 직전 30일 이내에 활성화된 적이 있는 사용자를 삭제하려는 경우 활성 사용자를 삭제한다는 추가 확인란을 선택해야 합니다. 계속하려면 예, 삭제를 선택합니다.

IAM 사용자 삭제(AWS CLI)

AWS Management 콘솔과 달리 AWS CLI로 사용자를 삭제할 때는 사용자에게 연결된 항목들을 수동으로 삭제해야 합니다. 다음 절차는 그 과정을 보여줍니다.

계정에서 사용자를 삭제하려면(AWS CLI)

1. 사용자의 키와 인증서를 삭제합니다. 이렇게 하면 사용자가 더 이상 AWS 계정의 리소스에 액세스할 수 없습니다. 보안 자격 증명을 삭제하면 영원히 지워져 검색할 수 없습니다.

`aws iam delete-access-key` 및 `aws iam delete-signing-certificate`

2. 해당 사용자의 암호가 있으면 삭제합니다.

`aws iam delete-login-profile`

3. 해당 사용자의 MFA 디바이스가 있으면 비활성화합니다.

`aws iam deactivate-mfa-device`

4. 사용자에게 연결된 모든 정책을 분리합니다.

`aws iam list-attached-user-policies`(사용자에게 연결된 정책 나열) 및 `aws iam detach-user-policy`(정책 분리)

5. 사용자가 속해 있던 모든 그룹의 목록을 가져오고 해당 그룹에서 사용자를 제거합니다.

`aws iam list-groups-for-user` 및 `aws iam remove-user-from-group`

6. 사용자를 삭제합니다.

`aws iam delete-user`

IAM 사용자의 권한 변경

그룹 멤버십을 변경하거나 기존 사용자에서 권한을 복사, 사용자에게 바로 정책을 연결 또는 [권한 경계\(p. 317\)](#)를 설정하여 AWS 계정의 IAM 사용자에 대한 권한을 변경할 수 있습니다. 이 권한 경계는 사용자가 가질 수 있는 최대 권한을 관리합니다. 권한 경계는 고급 AWS 기능입니다.

사용자의 권한을 수정하기 위해 필요한 권한에 대한 자세한 내용은 [IAM 리소스에 액세스하는 데 필요한 권한\(p. 443\)](#) 단원을 참조하십시오.

주제

- [사용자 액세스 보기 \(p. 74\)](#)
- [사용자\(콘솔\)에게 권한 추가 \(p. 74\)](#)
- [사용자\(콘솔\)의 권한 변경 \(p. 76\)](#)
- [사용자\(콘솔\)에서 권한 정책 제거 \(p. 77\)](#)

- 사용자(콘솔)에게서 권한 경계 제거 (p. 77)
- 사용자 권한(AWS CLI 또는 AWS API) 추가 및 제거 (p. 77)

사용자 액세스 보기

사용자에 대한 권한을 변경하기 전에 최근 서비스 수준 활동을 검토해야 합니다. 이 기능은 사용 중인 보안 주체(사람 또는 애플리케이션)의 액세스 권한을 제거하지 않으려는 경우 중요합니다. 서비스에서 마지막으로 액세스한 데이터 보기에 대한 자세한 내용은 [서비스에서 마지막으로 액세스한 데이터를 사용하여 권한 줄이기 \(p. 409\)](#) 단원을 참조하십시오.

사용자(콘솔)에게 권한 추가

IAM은 사용자에게 권한 정책을 추가하는 세 가지 방법을 제안합니다.

- 그룹에게 사용자 추가 – 사용자를 그룹의 구성원으로 만듭니다. 그룹의 정책은 사용자로 연결됩니다.
- 기존 사용자에서 권한 복사 – 소스 사용자에서 그룹 멤버십, 연결된 관리형 정책, 인라인 정책 및 기존 권한 경계를 모두 복사합니다.
- 정책을 사용자에 직접 연결 – 관리형 정책을 사용자에 직접 연결합니다. 그 대신에 그룹에 정책을 연결한 다음, 사용자들을 적절한 그룹의 구성원으로 만드는 것이 바람직한 [모범 사례 \(p. 44\)](#)입니다.

Important

사용자에게 권한 경계가 있다면 권한 경계가 허용한 권한보다 더 많은 권한을 추가할 수 없습니다.

사용자를 그룹에 추가하여 권한을 추가

사용자를 그룹에 추가하여 사용자에게 바로 영향을 줍니다.

사용자를 그룹에 추가하여 사용자에게 권한을 추가하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 콘솔의 그룹 열에서 사용자에 대한 현재 그룹 멤버십을 검토합니다. 필요할 경우 다음 단계를 통해 사용자 테이블에 열을 추가합니다.

1. 테이블 위 맨 오른쪽에서 설정 기호()를 선택합니다.

2. Manage Columns(열 관리) 대화 상자에서 그룹 열을 선택합니다. 필요할 경우 사용자 테이블에 표시하지 않으려는 열이 있으면 해당 열의 확인란 선택을 취소하면 됩니다.

3. 닫기를 선택하여 사용자 목록으로 돌아갑니다.

그룹 열에는 사용자가 속한 그룹이 표시됩니다. 열에는 최대 2개 그룹에 대한 그룹 이름이 표시됩니다. 사용자가 3개 이상 그룹의 구성원인 경우 처음 두 개 그룹만 알파벳 순서대로 표시되고 나머지 그룹 멤버십 수가 표시됩니다. 예를 들어 사용자가 그룹 A, 그룹 B, 그룹 C, 그룹 D에 속한 경우 필드에 Group A, Group B + 2 more(그룹 A, 그룹 B 외 2개)라고 표시됩니다. 사용자가 속한 총 그룹 수를 보려면 사용자 테이블에 Group count(그룹 수) 열을 추가합니다.

4. 권한을 수정하려는 사용자의 이름을 선택합니다.
5. 권한 탭을 선택한 다음 Add permissions(권한 추가)를 선택합니다. [Add user to group]을 선택합니다.
6. 사용자를 귀속시키려는 각 그룹의 확인란을 선택합니다. 그 목록에는 각 그룹의 이름과 사용자가 그 그룹의 구성원이 되면 받는 정책이 표시됩니다.
7. (선택 사항) 기존 그룹에서 선택할 수 있을 뿐 아니라 다음과 같이 그룹 생성을 선택하여 새 그룹을 정의할 수 있습니다.

- a. 새로운 탭에서 그룹 이름으로 새로운 그룹의 이름을 입력합니다.

Note

그룹 이름에는 최대 128개의 알파벳, 숫자 및 더하기(+) 등호(=), 쉼표(.) 마침표(.) 앤(@), 그리고 하이픈(-) 조합을 사용할 수 있습니다. 이름은 계정 내에서 고유해야 합니다. 대소문자는 구별하지 않습니다. 예를 들어 "TESTGROUP"과 "testgroup"이라는 두 그룹을 만들 수는 없습니다. IAM 엔터티 관련 제한에 대한 자세한 내용은 [IAM 개체 및 객체에 대한 제한 \(p. 485\)](#) 단원을 참조하십시오.

- b. 그룹에 연결하고자 하는 관리형 정책에 대해 한 개 이상의 확인란을 선택합니다. 정책 생성을 선택하여 새로운 관리형 정책을 만들 수도 있습니다. 이렇게 하는 경우, 새 정책이 완료되면 이 브라우저 탭 또는 창으로 돌아가 새로 고침을 선택한 다음, 그룹에 연결할 새로운 정책을 선택합니다. 자세한 내용은 [IAM 정책 만들기 \(p. 377\)](#) 단원을 참조하십시오.
 - c. Create group을 선택합니다.
 - d. 기존 탭으로 반환하고 그룹 목록을 새로 고침합니다. 새로운 그룹에 대한 확인란을 선택합니다.
8. Next: Review(다음: 검토)를 선택하여 사용자에 추가될 그룹 멤버십의 목록을 확인합니다. 그런 다음 Add permissions(권한 추가)를 선택합니다.

다른 사용자에게서 복사하여 권한을 추가

권한 복사는 사용자에게 바로 적용됩니다.

다른 사용자에게서 권한을 복사하여 사용자에게 권한을 추가하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택한 다음, 권한을 수정할 사용자의 이름을 선택하고 권한 탭을 선택합니다.
3. Add permissions(권한 추가)를 선택한 다음, Copy permissions from existing user(기존 사용자에서 권한 복사)를 선택합니다. 목록에는 사용 가능한 사용자들이 그들의 그룹 멤버십 및 연결된 정책과 함께 표시됩니다. 그룹 또는 정책의 전체 목록이 한 줄에 다 표시되지 않는 경우, and *n* more(외 *n*개) 링크를 선택할 수 있습니다. 그러면 새 브라우저 탭이 열리고 정책(권한 탭) 및 그룹(그룹 탭)의 전체 목록을 볼 수 있습니다.
4. 복사하고자 하는 권한을 보유한 사용자 옆에 있는 라디오 버튼을 선택합니다.
5. Next: Review(다음: 검토)를 선택하여 사용자에 대한 변경 사항의 목록을 확인합니다. 그런 다음 Add permissions(권한 추가)를 선택합니다.

사용자에게 직접 정책을 연결하여 권한을 추가

정책 연결은 사용자에게 바로 적용됩니다.

관리형 정책을 직접 연결하여 사용자에게 권한을 추가하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택한 다음, 권한을 수정할 사용자의 이름을 선택하고 권한 탭을 선택합니다.
3. Add permissions(권한 추가)를 선택한 다음, Attach existing policies directly to user(기존 정책을 사용자에게 직접 연결)를 선택합니다.
4. 사용자에 연결하고자 하는 관리형 정책에 대해 한 개 이상의 확인란을 선택합니다. 정책 생성을 선택하여 새로운 관리형 정책을 만들 수도 있습니다. 이렇게 하는 경우, 새 정책이 완료되면 이 브라우저 탭 또는 창으로 돌아가 새로 고침을 선택한 다음, 사용자에게 연결할 새로운 정책 확인란을 선택합니다. 자세한 내용은 [IAM 정책 만들기 \(p. 377\)](#) 단원을 참조하십시오.
5. Next: Review(다음: 검토)를 선택하여 사용자에 연결될 정책의 목록을 확인합니다. 그런 다음 Add permissions(권한 추가)를 선택합니다.

사용자에 대한 권한 경계 설정

권한 경계 설정은 사용자에게 바로 적용됩니다.

사용자에 대한 권한 경계를 설정하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 권한 경계를 변경하려는 사용자의 이름을 선택합니다.
4. Permissions 탭을 선택합니다. 필요하다면 Permissions boundary(권한 경계) 섹션을 열고 Set boundary(경계 설정)를 선택합니다.
5. 정책을 선택하여 원하는 권한 경계를 사용하십시오.
6. Set boundary(경계 설정)을 선택합니다.

사용자(콘솔)의 권한 변경

IAM은 사용자와 관련된 권한을 변경하는 세 가지 방법을 제안합니다.

- 권한 정책 편집 – 사용자 인라인 정책, 사용자 그룹의 인라인 정책을 편집하거나 바로 사용자에게 또는 그룹에서 연결된 관리형 정책을 편집합니다. 사용자에게 권한 경계가 있다면 권한 경계로 사용된 정책이 허용한 권한보다 더 많은 권한을 제공할 수 없습니다.
- 권한 경계 변경 – 사용자에 대한 권한 경계로 사용된 정책을 변경합니다. 이로써 사용자가 가질 수 있는 최대 권한을 확장 또는 제한할 수 있습니다.

사용자에게 연결된 권한 정책을 편집합니다

권한 변경은 사용자에게 바로 적용됩니다.

사용자의 연결된 관리형 정책을 편집하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 권한 정책을 변경하려는 사용자의 이름을 선택합니다.
4. Permissions 탭을 선택합니다. 필요하다면 Permissions policies(권한 정책) 부분을 엽니다.
5. 정책에 대한 세부 정보를 보기 위해서 편집하고자 하는 정책 이름을 선택합니다. Used as(다음과 같이 사용됨) 탭을 선택하여 정책을 편집함으로써 영향을 받을 다른 개체를 봅니다.
6. 그런 다음 Permissions tab(권한 탭)을 정책이 허용한 권한을 검토합니다. 그런 다음 정책 편집을 선택합니다.
7. Visual editor(시각적 편집기) 탭 또는 JSON 탭을 사용하여 정책을 편집합니다. 자세한 내용은 [IAM 정책 편집 \(p. 402\)](#) 단원을 참조하십시오.
8. 정책 검토를 선택한 다음 정책 요약을 검토한 후 변경 사항 저장을 선택합니다.

사용자에 대한 권한 경계를 변경하십시오.

권한 경계 변경은 사용자에게 바로 적용됩니다.

사용자의 권한 경계 설정에 사용된 정책을 변경하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.

2. 탐색 창에서 사용자를 선택합니다.
3. 권한 경계를 변경하려는 사용자의 이름을 선택합니다.
4. Permissions 탭을 선택합니다. 필요하다면 Permissions boundary(권한 경계) 섹션을 열고 Change boundary(경계 변경)을 선택합니다.
5. 정책을 선택하여 원하는 권한 경계를 사용하십시오.
6. Change boundary(경계 변경)을 선택합니다.

사용자(콘솔)에게서 권한 정책 제거

정책 제거는 사용자에게 바로 적용됩니다.

IAM 사용자의 권한을 취소하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 권한 경계를 제거하려는 사용자의 이름을 선택합니다.
4. Permissions 탭을 선택합니다.
5. 기존 정책을 제거하여 권한을 취소하려면 정책을 제거하기 전에 X를 선택하여 정책 유형에서 사용자가 어떻게 정책을 받는지 확인합니다.
 - 그 정책이 그룹 멤버십 때문에 적용되는 경우, X를 선택하면 사용자가 그룹에서 제거됩니다. 한 그룹에 여러 정책이 연결될 수 있습니다. 따라서 그룹에서 사용자를 제거할 경우 사용자는 그 그룹의 멤버십을 통해 받은 모든 정책에 대한 액세스 권한을 잃게 됩니다.
 - 정책이 사용자에 직접 연결된 관리형 정책인 경우 X를 선택하면 정책이 사용자와 분리됩니다. 이렇게 해도 정책 자체 또는 그 정책이 연결되어 있을 수 있는 다른 개체에는 영향을 미치지 않습니다.
 - 정책이 인라인 포함 정책인 경우, X를 선택하면 정책이 IAM에서 제거됩니다. 사용자에 직접 연결된 인라인 정책은 해당 사용자에만 존재합니다.

사용자(콘솔)에게서 권한 경계 제거

권한 경계 제거는 사용자에게 바로 적용됩니다.

사용자(콘솔)에게서 권한 경계를 제거하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 권한 경계를 제거하려는 사용자의 이름을 선택합니다.
4. Permissions 탭을 선택합니다. 필요하다면 Permissions boundary(권한 경계) 섹션을 열고 Remove boundary(경계 제거)를 선택합니다.
5. 제거를 선택하여 권한 경계를 제거합니다.

사용자 권한(AWS CLI 또는 AWS API) 추가 및 제거

프로그래밍 방식으로 권한을 추가 또는 제거하려면 그룹 멤버십을 추가 또는 제거하거나 관리형 정책을 연결 또는 분리하거나 인라인 정책을 추가 또는 삭제해야 합니다. 자세한 내용은 다음 주제 단원을 참조하십시오.

- IAM 그룹에서 사용자 추가 및 제거 (p. 149)
- IAM 자격 증명 권한 추가 및 제거 (p. 391)

암호 관리

AWS 계정 루트 사용자 및 계정의 IAM 사용자의 암호를 관리할 수 있습니다. IAM 사용자가 AWS Management 콘솔에 액세스하려면 암호가 필요합니다. 사용자가 AWS CLI, Windows PowerShell용 도구, AWS SDK 또는 API를 사용하여 프로그래밍 방식으로 AWS 리소스에 액세스하는 경우 암호가 필요 없습니다. 대신 그러한 환경에서는 사용자에게 [액세스 키 \(p. 88\)](#)가 필요합니다.

주제

- [AWS 계정 루트 사용자 암호 변경 \(p. 78\)](#)
- [IAM 사용자의 계정 암호 정책 설정 \(p. 79\)](#)
- [IAM 사용자의 암호 관리 \(p. 82\)](#)
- [IAM 사용자에게 자신의 암호 변경 허용하기 \(p. 85\)](#)
- [IAM 사용자가 자신의 암호를 변경하는 방법 \(p. 87\)](#)

AWS 계정 루트 사용자 암호 변경

암호를 변경하려면 AWS 계정 루트 사용자로 로그인해야 합니다. 잊어버린 루트 사용자 암호를 재설정하는 방법에 대한 자세한 내용은 [분실하거나 잊어버린 암호 또는 액세스 키 재설정 \(p. 95\)](#) 단원을 참조하십시오.

루트 사용자의 암호를 변경하려면

1. AWS 계정 이메일 주소와 암호를 사용하여 [AWS Management 콘솔](#)에 루트 사용자로 로그인합니다.

Note

이전에 [IAM 사용자 \(p. 63\)](#) 자격 증명으로 콘솔에 로그인한 경우, 브라우저가 이 설정을 기억하여 계정별 로그인 페이지를 열 수도 있습니다. IAM 사용자 로그인 페이지에서는 AWS 계정 루트 사용자 자격 증명으로 로그인할 수 없습니다. IAM 사용자 로그인 페이지가 나타날 경우에는 페이지 하단에 있는 [Sign in using root account credentials]를 선택하여 기본 로그인 페이지로 돌아갑니다. 기본 로그인 페이지에서 AWS 계정의 이메일 주소와 암호를 입력합니다.

2. 콘솔의 오른쪽 상단 모서리 부분에서 계정 이름이나 번호를 선택한 후 내 계정을 선택합니다.
3. 페이지 오른쪽의 계정 설정 섹션 옆에서 편집을 선택합니다.
4. 암호 줄에서 편집을 선택하여 암호를 변경합니다.
5. 강력한 암호를 선택하십시오. [IAM 사용자에 대한 계정 암호 정책을 설정 할 수는 있지만 \(p. 79\)](#), AWS 계정 루트 사용자에게는 이 정책이 적용되지 않습니다.

AWS는 암호가 다음 조건을 충족하도록 요구합니다.

- 최소 8자 이상이고 최대 128자 이하여야 함
- 대문자, 소문자, 숫자, 비-영문자 기호(예: ! @ # \$ % ^ & * () <> [] {} | _+=-) 중에서 세 가지 이상의 혼합 문자 유형 포함
- AWS 계정 이름 또는 이메일 주소와 동일하지 않아야 함

Note

AWS가 로그인 프로세스의 개선 사항을 공개합니다. 그중 하나는 사용자 계정에 더 안전한 암호 정책을 강화하는 것입니다. 계정이 업그레이드된 경우 위 암호 정책에 부합해야 합니다. 아직 계정이 업그레이드되지 않았다면 AWS에서는 이 정책이 집행되지 않으나, 더 안전한 암호를 위하여 지침을 따를 것을 강력히 권장합니다.

암호를 보호하려면 다음과 같은 모범 사례를 활용하는 것이 중요합니다.

- 주기적으로 암호를 변경하고 암호는 비공개로 유지하십시오. 암호를 아는 사람이 귀하의 계정에 액세스할 수 있습니다.
- AWS의 암호를 다른 사이트에서 사용하는 것과 다르게 지정하십시오.
- 짐작하기 쉬운 암호를 사용하지 마십시오. 여기에는 secret, password, amazon 또는 123456 같은 암호가 포함됩니다. 또한 사전에 나오는 단어, 사용자 이름, 이메일 주소 또는 알아내기 쉬운 그 밖의 개인 정보도 포함됩니다.

IAM 사용자의 계정 암호 정책 설정

AWS 계정에서 암호 정책을 설정하여 IAM 사용자 암호의 복잡성 요건과 의무적인 교체 주기를 지정할 수 있습니다.

이러한 작업을 실행할 때 암호 정책을 사용할 수 있습니다.

- 최소 암호 길이를 설정합니다.
- 대문자, 소문자, 숫자, 비-영숫자를 포함하는 특정 문자 유형이 필요합니다. 사용자에게 암호의 대소문자가 구분된다는 점을 알려야 합니다.
- 모든 IAM 사용자에게 자신의 암호 변경을 허용합니다.

Note

IAM 사용자에게 자신의 암호 변경을 허용하면 IAM이 자동으로 사용자에게 암호 정책을 보여줍니다. IAM 사용자가 정책에 따르는 암호를 생성하려면 계정의 암호 정책을 볼 수 있어야 합니다.

- IAM 사용자에게 지정 시간(암호 만료 설정)이 지나면 암호를 변경하라고 요구합니다.
- IAM 사용자가 이전 암호를 재사용하는 것을 금지합니다.
- IAM 사용자의 암호가 만료된 경우에는 계정 관리자에게 사용자가 연락하게 합니다.

Important

여기에서 설명한 암호 설정은 IAM 사용자에게 할당된 암호에만 적용되고 사용자들이 갖고 있을 수 있는 액세스 키에는 영향을 미치지 않습니다. 암호가 만료된 경우, 사용자는 AWS Management 콘솔에 로그인할 수 없습니다. 하지만 사용자에게 유효한 액세스 키가 있으면 여전히 AWS CLI 또는 Windows PowerShell용 도구 명령을 실행할 수 있습니다. 또한 애플리케이션을 통해 사용자의 권한이 허용하는 API 작업을 호출할 수도 있습니다.

암호 정책을 생성 또는 변경하더라도 대부분의 암호 정책 설정은 사용자가 다음에 자신의 암호를 변경할 때 적용됩니다. 하지만 일부 설정은 바로 적용됩니다. 예:

- 최소 길이 및 문자 유형 요건을 설정하면 그 설정 사항은 다음 번에 사용자가 자신의 암호를 변경할 때 적용됩니다. 기존 암호가 업데이트된 암호 정책을 따르지 않는 경우에도 사용자들은 기존 암호를 변경할 필요는 없습니다.
- 암호 만료 기간을 설정하면 만료 기간이 바로 적용됩니다. 예를 들어 암호 만료 기간을 90일로 설정한 경우, 현재 90일 이상 지난 기존 암호를 지닌 IAM 사용자들은 모두 다음 로그인 시 자신의 암호를 변경해야 합니다.

암호 정책을 설정하기 위해 필요한 권한에 대한 자세한 내용은 [IAM 사용자에게 자신의 암호 변경 허용하기 \(p. 85\)](#) 단원을 참조하십시오.

IAM 암호 정책은 AWS 계정 루트 사용자 암호에는 적용되지 않습니다.

현재 사용 가능한 이 옵션으로는 "잠금 정책"이라고 하는 것을 만들 수 없습니다. 이러한 정책은 로그인 시도 실패 횟수가 지정한 횟수에 도달하면 사용자 계정을 잠금니다. 보안을 강화하려면 암호 정책과 멀티 팩터 인증(MFA)을 함께 사용하는 것이 좋습니다. MFA에 대한 자세한 내용은 [AWS에서 멀티 팩터 인증\(MFA\) 사용하기 \(p. 96\)](#) 단원을 참조하십시오.

주제

- [암호 정책 옵션 \(p. 80\)](#)
- [암호 정책 설정\(콘솔\) \(p. 81\)](#)
- [암호 정책 설정\(AWS CLI\) \(p. 81\)](#)
- [암호 정책 설정\(AWS API\) \(p. 82\)](#)

암호 정책 옵션

아래는 계정의 암호 정책 구성 시 사용할 수 있는 옵션들입니다.

최소 암호 길이

IAM 사용자 암호에서 허용되는 최소 문자 수를 지정할 수 있습니다. 6~128 범위에서 입력할 수 있습니다.

1개 이상의 대문자 필수

IAM 사용자 암호에 ISO 기본 라틴 알파벳(A~Z) 중 1개 이상의 대문자를 사용하도록 요구할 수 있습니다.

1개 이상의 소문자 필수

IAM 사용자 암호에 ISO 기본 라틴 알파벳(a~z) 중 1개 이상의 소문자를 사용하도록 요구할 수 있습니다.

1개 이상의 숫자 필수

IAM 사용자 암호에 숫자(0~9) 중 1개 이상의 숫자를 사용하도록 요구할 수 있습니다.

알파벳이나 숫자가 아닌 1개 이상의 문자 필수

IAM 사용자 암호에 다음과 같이 알파벳이나 숫자가 아닌 최소 1개의 문자를 사용하도록 요구할 수 있습니다.

! @ # \$ % ^ & * () _ + - = [] { } | '

사용자 자신의 암호 변경 허용

계정의 IAM 사용자 모두 IAM 콘솔을 사용하여 자신의 암호를 변경할 수 있습니다. 자세한 설명은 [IAM 사용자에게 자신의 암호 변경 허용하기 \(p. 85\)](#) 단원을 참조하십시오.

그 밖에 자신이나 다른 사용자의 암호를 관리하는 사용자를 일부로 제한할 수도 있습니다. 이렇게 하려면 사용자 자신의 암호 변경 허용(Allow users to change their own password) 확인란 선택을 해제하면 됩니다. 암호 관리 제한 정책의 사용에 대한 자세한 내용은 [IAM 사용자에게 자신의 암호 변경 허용하기 \(p. 85\)](#) 단원을 참조하십시오.

Note

IAM 사용자에게 자신의 암호 변경을 허용하면 IAM이 자동으로 사용자에게 암호 정책을 보여줍니다. IAM 사용자가 정책에 따르는 암호를 생성하려면 계정의 암호 정책을 볼 수 있어야 합니다.

암호 만료 활성화

IAM 사용자 암호는 지정 일수 동안만 유효하도록 설정할 수 있습니다. 방법은 암호 설정 후 유효 일수를 지정하면 됩니다. 예를 들어 암호 만료를 활성화하여 암호 만료 기간을 90일로 설정하면 IAM 사용자는 최대 90일까지 암호를 사용할 수 있습니다. 90일이 지나면 암호가 만료되어 IAM 사용자가 AWS Management 콘솔에 액세스하려면 암호를 새로 설정해야 합니다. 암호 만료 기간은 1~1,095(1,095 포함)일 중에서 선택할 수 있습니다.

Note

암호 만료까지 15일이 남으면 AWS Management 콘솔이 IAM 사용자에게 경고를 보냅니다. IAM 사용자는 언제든지 자신의 암호를 변경할 수 있습니다(변경 권한이 있는 경우에 한함). 새

암호를 설정하면 암호 변경 기간이 다시 시작됩니다. IAM 사용자는 한 번에 유효 암호 하나만 사용할 수 있습니다.

암호 재사용 제한

IAM 사용자가 이전 암호를 지정한 수만큼 재사용하지 못하도록 제한할 수 있습니다. 설정할 수 있는 암호 수는 1~24(24 포함)개입니다.

암호 만료 시 관리자 재설정

현재 암호가 만료된 후 IAM 사용자가 새 암호를 선택하지 못하도록 제한할 수 있습니다. 예를 들어 암호 정책은 암호 만료 기간을 지정할 수 있습니다. IAM 사용자가 암호 만료 전에 새 암호를 선택하지 않으면 IAM 사용자는 새 암호를 설정할 수 없습니다. 이 경우 IAM 사용자가 AWS Management 콘솔에 대한 액세스 권한을 다시 얻으려면 계정 관리자의 암호 재설정을 요구해야 합니다. 이 확인란을 그냥 비워 놓아도 됩니다. IAM 사용자가 자신의 암호를 만료되게 되 경우 사용자는 AWS Management 콘솔에 액세스하기 전에 새 암호를 설정해야 합니다.

Warning

이 옵션을 활성화하기 전에 AWS 계정에 관리자 권한(IAM 사용자 암호의 재설정 권한)을 가진 사용자가 2명 이상인지 확인해야 합니다. 또는 관리자에게도 AWS CLI 또는 Windows PowerShell용 도구를 AWS Management 콘솔과 별도로 사용할 수 있는 액세스 키가 있는지 확인할 수 있습니다. 이 옵션이 활성화된 상태에서 한 관리자의 암호가 만료된 경우, 첫 번째 관리자의 만료된 암호를 재설정하려면 두 번째 관리자가 콘솔에 로그인해야 합니다. 그러나 암호가 만료된 관리자에게 유효한 액세스 키가 있는 경우에는 AWS CLI 또는 Windows PowerShell 용 도구 명령을 실행할 수 있습니다. 이러한 명령은 관리자의 암호를 재설정할 수 있습니다. 두 번째 관리자에 대한 요건은 암호가 만료되고 첫 번째 관리자에게 액세스 키가 없는 경우에만 적용됩니다.

암호 정책 설정(콘솔)

AWS Management 콘솔에서 암호 정책을 생성, 변경 또는 삭제할 수 있습니다. 암호 정책 관리의 일환으로 모든 사용자가 자신의 암호를 관리하도록 허용할 수 있습니다.

암호 정책을 생성하거나 변경하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 계정 설정을 클릭합니다.
3. 암호 정책 섹션에서 암호 정책에 적용하려는 옵션을 선택합니다.
4. 암호 정책 적용(Apply Password Policy)를 클릭합니다.

암호 정책을 삭제하는 방법

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 계정 설정을 클릭한 다음 암호 정책 섹션에서 암호 정책 삭제(Delete Password Policy)를 클릭합니다.

암호 정책 설정(AWS CLI)

AWS CLI에서 계정 암호 정책을 관리하려면 다음 명령을 실행하십시오.

- 암호 정책을 생성 또는 변경하는 방법: `aws iam update-account-password-policy`
- 암호 정책 가져오기: `aws iam get-account-password-policy`
- 암호 정책을 삭제하는 방법: `aws iam delete-account-password-policy`

암호 정책 설정(AWS API)

AWS API에서 계정 암호 정책을 관리하려면 다음 작업을 호출하십시오.

- 암호 정책을 생성 또는 변경하는 방법: [UpdateAccountPasswordPolicy](#)
- 암호 정책 가져오기: [GetAccountPasswordPolicy](#)
- 암호 정책을 삭제하는 방법: [DeleteAccountPasswordPolicy](#)

IAM 사용자의 암호 관리

AWS Management 콘솔을 사용하여 AWS 리소스를 작업하는 IAM 사용자가 로그인하려면 암호가 필요합니다. AWS 계정에 속한 IAM 사용자의 암호를 생성, 변경 또는 삭제할 수 있습니다.

사용자에게 암호를 할당한 후 사용자는 다음과 같은 계정의 로그인 URL을 사용하여 AWS Management 콘솔에 로그인할 수 있습니다.

`https://12-digit-AWS-account-ID or alias.signin.aws.amazon.com/console`

IAM 사용자가 AWS Management 콘솔에 로그인하는 방법에 대한 자세한 내용은 [IAM 콘솔 및 로그인 페이지 \(p. 53\)](#) 단원을 참조하십시오.

IAM 사용자의 개별 암호를 수동으로 만들 수 있지만 AWS 계정에 속한 모든 IAM 사용자의 암호에 적용되는 암호 정책을 만들 수도 있습니다.

이러한 작업을 실행할 때 암호 정책을 사용할 수 있습니다.

- 최소 암호 길이를 설정합니다.
- 대문자, 소문자, 숫자, 비-영숫자를 포함하는 특정 문자 유형이 필요합니다. 사용자에게 암호의 대소문자가 구분된다는 점을 알려야 합니다.
- 모든 IAM 사용자에게 자신의 암호 변경을 허용합니다.

Note

IAM 사용자에게 자신의 암호 변경을 허용하면 IAM이 자동으로 사용자에게 암호 정책을 보여줍니다. IAM 사용자가 정책에 따르는 암호를 생성하려면 계정의 암호 정책을 볼 수 있어야 합니다.

- IAM 사용자에게 지정 시간(암호 만료 설정)이 지나면 암호를 변경하라고 요구합니다.
- IAM 사용자가 이전 암호를 재사용하는 것을 금지합니다.
- IAM 사용자의 암호가 만료된 경우에는 계정 관리자에게 사용자가 연락하게 합니다.

계정의 암호 정책 관리에 대한 자세한 내용은 [IAM 사용자의 계정 암호 정책 설정 \(p. 79\)](#)을 참조하십시오.

사용자에게 암호가 있더라도 AWS 리소스에 액세스하려면 권한이 필요합니다. 기본적으로 사용자에게는 권한이 없습니다. 사용자에게 필요한 권한을 부여하려면 해당 사용자 또는 사용자가 속한 그룹에 정책을 할당합니다. 사용자 및 그룹 만들기에 대한 자세한 내용은 [자격 증명\(사용자, 그룹, 및 역할\) \(p. 61\)](#)을 참조하십시오. 권한 설정을 위한 정책 사용에 대한 자세한 내용은 [IAM 사용자의 권한 변경 \(p. 73\)](#)을 참조하십시오.

자신의 암호를 변경할 권한을 사용자에게 부여할 수 있습니다. 자세한 내용은 [IAM 사용자에게 자신의 암호 변경 허용하기 \(p. 85\)](#) 단원을 참조하십시오. 사용자가 계정 로그인 페이지에 액세스하는 방법에 대한 자세한 내용은 [IAM 콘솔 및 로그인 페이지 \(p. 53\)](#)을 참조하십시오.

주제

- [IAM 사용자 암호 생성, 변경 또는 삭제\(콘솔\) \(p. 83\)](#)

- IAM 사용자 암호 생성, 변경 또는 삭제(AWS CLI) (p. 84)
- IAM 사용자 암호 생성, 변경 또는 삭제(AWS API) (p. 85)

IAM 사용자 암호 생성, 변경 또는 삭제(콘솔)

AWS Management 콘솔을 사용하여 IAM 사용자의 암호를 관리할 수 있습니다.

사용자가 조직을 떠나거나 AWS 액세스가 더 이상 필요하지 않은 경우 사용 중인 자격 증명을 찾아서 더 이상 작동하지 않도록 해야 합니다. 더 이상 필요 없는 자격 증명을 삭제하는 것이 가장 좋습니다. 나중에 필요 한 경우가 생기면 언제든지 다시 생성할 수 있습니다. 적어도 그 자격 증명을 변경하여 이전 사용자가 더 이상 액세스할 수 없게 해야 합니다.

IAM 사용자의 암호를 추가하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 암호를 생성하려는 사용자의 이름을 선택합니다.
4. 보안 자격 증명(Security credentials) 탭을 선택한 다음, 로그인 자격 증명(Sign-in credentials)에서 콘솔 암호(Console password) 옆에 있는 암호 관리(Manage password)를 선택합니다.
5. 콘솔 액세스 관리(Manage console access)의 콘솔 액세스(Console access)에서 활성화를 선택합니다 (선택되어 있지 않은 경우). 콘솔 액세스가 비활성화되어 있는 경우에는 암호가 필요 없습니다.
6. Set password(암호 설정)에 대해서는 IAM에서 암호를 자동으로 생성할지, 아니면 사용자 지정 암호를 만들지를 선택합니다.
 - IAM에서 암호를 자동으로 생성하려면 Autogenerated password(자동 생성 암호)를 선택합니다.
 - 사용자 지정 암호를 만들려면 사용자 지정 암호(Custom password)를 선택하고 암호를 입력합니다.

Note

만드는 암호는 계정의 암호 정책 (p. 79)(정책을 설정한 경우)에 부합해야 합니다.

7. 사용자가 로그인할 때 새 암호를 만들도록 요구하려면 암호 재설정 요청(Require password reset)을 선택합니다. 그 다음 적용을 선택합니다.

Important

암호 재설정 요청(Require password reset) 옵션을 선택할 경우, 사용자에게 자신의 암호를 변경할 권한이 있는지 확인하십시오. 자세한 내용은 [IAM 사용자에게 자신의 암호 변경 허용하기 \(p. 85\)](#) 단원을 참조하십시오.

8. 암호를 생성하는 옵션을 선택한 경우, 새 비밀번호 대화 상자에서 표시를 선택합니다. 이렇게 하면 암호를 볼 수 있으므로 암호를 사용자와 공유할 수 있습니다.

Important

이 단계를 완료한 후에는 보안상의 이유로 암호에 액세스할 수 없지만, 언제든지 새 암호를 만들 수 있습니다.

IAM 사용자의 암호를 변경하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 암호를 변경할 사용자의 이름을 선택합니다.
4. 보안 자격 증명(Security credentials) 탭을 선택한 다음, 로그인 자격 증명(Sign-in credentials)에서 콘솔 암호(Console password) 옆에 있는 암호 관리(Manage password)를 선택합니다.

5. 콘솔 액세스 관리(Manage console access)의 콘솔 액세스(Console access)에서 활성화를 선택합니다.(선택되어 있지 않은 경우). 콘솔 액세스가 비활성화되어 있는 경우에는 암호가 필요 없습니다.
6. Set password(암호 설정)에 대해서는 IAM에서 암호를 자동으로 생성할지, 아니면 사용자 지정 암호를 만들지를 선택합니다.
 - IAM에서 암호를 자동으로 생성하려면 Autogenerated password(자동 생성 암호)를 선택합니다.
 - 사용자 지정 암호를 만들려면 사용자 지정 암호(Custom password)를 선택하고 암호를 입력합니다.

Note

만드는 암호는 계정의 [암호 정책 \(p. 79\)](#)(정책을 설정한 경우)에 부합해야 합니다.

7. 사용자가 로그인할 때 새 암호를 만들도록 요구하려면 암호 재설정 요청(Require password reset)을 선택합니다. 그 다음 적용을 선택합니다.

Important

암호 재설정 요청(Require password reset) 옵션을 선택할 경우, 사용자에게 자신의 암호를 변경할 권한이 있는지 확인하십시오. 자세한 내용은 [IAM 사용자에게 자신의 암호 변경 허용하기 \(p. 85\)](#) 단원을 참조하십시오.

8. 암호를 생성하는 옵션을 선택한 경우, 새 비밀번호 대화 상자에서 표시를 선택합니다. 이렇게 하면 암호를 볼 수 있으므로 암호를 사용자와 공유할 수 있습니다.

Important

이 단계를 완료한 후에는 보안상의 이유로 암호에 액세스할 수 없지만, 언제든지 새 암호를 만들 수 있습니다.

IAM 사용자의 암호를 삭제(비활성화)하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 암호를 삭제할 사용자의 이름을 선택합니다.
4. 보안 자격 증명(Security credentials) 탭을 선택한 다음, 로그인 자격 증명(Sign-in credentials)에서 콘솔 암호(Console password) 옆에 있는 암호 관리(Manage password)를 선택합니다.
5. 콘솔 액세스(Console access)에 대해서는 비활성화에 이어 적용을 선택합니다.

Important

사용자의 암호를 삭제하면 해당 사용자가 더 이상 AWS Management 콘솔에 로그인할 수 없습니다. 사용자에게 액세스 키가 있다면 액세스 키는 계속 제 기능을 수행하고 AWS CLI, Windows PowerShell용 도구 또는 AWS API 함수 호출을 통해 액세스를 허용합니다.

IAM 사용자 암호 생성, 변경 또는 삭제(AWS CLI)

AWS CLI API를 이용해 IAM 사용자의 암호를 관리할 수 있습니다.

암호를 생성하려면(AWS CLI)

1. (선택 사항) 사용자에게 암호가 있는지 확인하려면 `aws iam get-login-profile` 명령을 실행합니다.
2. 암호를 생성하려면 `aws iam create-login-profile` 명령을 실행합니다.

사용자의 암호를 변경하려면(AWS CLI)

1. (선택 사항) 사용자에게 암호가 있는지 확인하려면 `aws iam get-login-profile` 명령을 실행합니다.

2. 암호를 변경하려면 [aws iam update-login-profile](#) 명령을 실행합니다.

사용자의 암호를 삭제(비활성화)하려면(AWS CLI)

1. (선택 사항) 사용자에게 암호가 있는지 확인하려면 [aws iam get-login-profile](#) 명령을 실행합니다.
2. (선택 사항) 사용자의 암호가 마지막으로 사용된 시간을 확인하려면 [aws iam get-user](#) 명령을 실행합니다.
3. 암호를 삭제하려면 [aws iam delete-login-profile](#) 명령을 실행합니다.

Important

사용자의 암호를 삭제하면 해당 사용자가 더 이상 AWS Management 콘솔에 로그인할 수 없습니다. 사용자에게 액세스 키가 있다면 액세스 키는 계속 제 기능을 수행하고 AWS CLI, Windows PowerShell용 도구 또는 AWS API 함수 호출을 통해 액세스를 허용합니다. AWS CLI, Windows PowerShell용 도구 또는 AWS API를 사용하여 AWS 계정에서 사용자를 삭제하는 경우 먼저 이 작업을 사용하여 암호를 삭제해야 합니다. 자세한 내용은 [IAM 사용자 삭제\(AWS CLI\) \(p. 73\)](#) 단원을 참조하십시오.

IAM 사용자 암호 생성, 변경 또는 삭제(AWS API)

AWS API를 이용해 IAM 사용자의 암호를 관리할 수 있습니다.

암호를 생성하려면(AWS API)

1. (선택 사항) 사용자에게 암호가 있는지 확인하려면 [GetLoginProfile](#) 연산을 호출합니다.
2. 암호를 생성하려면 [CreateLoginProfile](#) 연산을 호출합니다.

사용자의 암호를 변경하려면(AWS API)

1. (선택 사항) 사용자에게 암호가 있는지 확인하려면 [GetLoginProfile](#) 연산을 호출합니다.
2. 암호를 변경하려면 [UpdateLoginProfile](#) 연산을 호출합니다.

사용자의 암호를 삭제(비활성화)하려면(AWS API)

1. (선택 사항) 사용자에게 암호가 있는지 확인하려면 [GetLoginProfile](#) 명령을 실행합니다.
2. (선택 사항) 사용자의 암호가 마지막으로 사용된 시간을 확인하려면 [GetUser](#) 명령을 실행합니다.
3. 암호를 삭제하려면 [DeleteLoginProfile](#) 명령을 실행합니다.

Important

사용자의 암호를 삭제하면 해당 사용자가 더 이상 AWS Management 콘솔에 로그인할 수 없습니다. 사용자에게 액세스 키가 있다면 액세스 키는 계속 제 기능을 수행하고 AWS CLI, Windows PowerShell용 도구 또는 AWS API 함수 호출을 통해 액세스를 허용합니다. AWS CLI, Windows PowerShell용 도구 또는 AWS API를 사용하여 AWS 계정에서 사용자를 삭제하는 경우 먼저 이 작업을 사용하여 암호를 삭제해야 합니다. 자세한 내용은 [IAM 사용자 삭제\(AWS CLI\) \(p. 73\)](#) 단원을 참조하십시오.

IAM 사용자에게 자신의 암호 변경 허용하기

IAM 사용자에게 AWS Management 콘솔에 로그인하기 위해 자신의 암호를 변경할 권한을 부여할 수 있습니다. 이 작업을 두 가지 방법으로 수행할 수 있습니다.

- 계정의 모든 IAM 사용자에게 자신의 암호 변경을 허용합니다 (p. 86).

- 선택된 IAM 사용자에게만 자신의 암호 변경을 허용합니다 (p. 86). 이 시나리오에서는 모든 사용자의 암호 변경 옵션을 비활성화한 후 IAM 정책을 사용하여 일부 사용자에게만 암호, 그리고 선택 사항으로 액세스 키와 같은 기타 자격 증명을 변경할 수 있는 권한을 부여합니다.

Important

사용자가 강력한 암호를 만들도록 암호 정책을 설정 (p. 79)하는 것이 좋습니다.

모든 IAM 사용자에게 자신의 암호 변경을 허용하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 계정 설정을 클릭합니다.
3. 암호 정책 섹션에서 사용자 자신의 암호 변경 허용(Allow users to change their own password)을 선택한 후 암호 정책 적용(Apply Password Policy)을 클릭합니다.
4. 사용자가 암호 변경 방법이 나와 있는 [IAM 사용자가 자신의 암호를 변경하는 방법 \(p. 87\)](#) 지침을 따르도록 해야 합니다.

계정의 암호 정책(모든 사용자가 직접 암호를 변경하게 하는 정책 포함) 변경에 사용할 수 있는 AWS CLI, Windows PowerShell용 도구 및 API 명령에 대한 자세한 내용은 [암호 정책 설정\(AWS CLI\) \(p. 81\)](#) 단원을 참조하십시오.

선택된 IAM 사용자에게 자신의 암호 변경을 허용하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 계정 설정을 클릭합니다.
3. 계정 설정 섹션에서 사용자의 본인 암호 변경 허용(Allow users to change their own password) 확인란이 해제되어 있는지 확인합니다. 이 확인란을 선택하면 모든 사용자가 직접 암호를 변경할 수 있게 됩니다. (위 절차 참조).
4. 암호를 변경하도록 허용할 사용자가 아직 없다면 사용자를 만듭니다. 세부 정보는 [AWS 계정의 IAM 사용자 생성 \(p. 65\)](#) 단원을 참조하십시오.
5. 직접 암호를 변경하게 할 IAM 사용자 그룹을 만든 다음 앞 단계에서 만든 사용자를 그룹에 추가합니다. 자세한 내용은 [첫 번째 IAM 관리자 및 그룹 생성 \(p. 17\)](#) 및 [IAM 그룹 관리 \(p. 148\)](#) 단원을 참조하십시오.

이 단계는 선택 사항이지만, 그룹을 사용하여 권한을 관리하면 사용자를 그룹에 추가 및 삭제하고 전체 그룹에 대해 일괄적으로 권한을 변경할 수 있어 더욱 편리합니다.

6. 그룹에 다음 정책을 할당합니다. 세부 정보는 [IAM 정책 관리 \(p. 377\)](#) 단원을 참조하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iam:GetAccountPasswordPolicy",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "iam:ChangePassword",  
            "Resource": "arn:aws:iam::account-id-without-hyphens:user/${aws:username}"  
        }  
    ]  
}
```

이 정책은 [암호 변경](#) 작업에 대한 액세스 권한을 부여하여 사용자가 콘솔, AWS CLI, Windows PowerShell용 도구 또는 API로부터 본인의 암호만을 변경할 수 있게 합니다. 또한, 사용자가 현재 암호 정책을 볼 수 있도록 [GetAccountPasswordPolicy](#) 작업에 대한 액세스 권한도 부여합니다. 이 권한은 사용자가 콘솔에서 비밀번호 변경 페이지를 표시하는 데 필요합니다. 사용자는 반드시 현재 암호 정책을 읽고 변경된 암호가 정책의 요건을 충족하는지 확인해야 합니다.

7. 사용자가 암호 변경 방법이 나와 있는 [IAM 사용자가 자신의 암호를 변경하는 방법 \(p. 87\)](#) 지침을 따르도록 해야 합니다.

자세한 정보

자격 증명 관리에 대한 자세한 내용은 다음 주제를 참조하십시오.

- [IAM 사용자에게 자신의 암호 변경 허용하기 \(p. 85\)](#)
- [암호 관리 \(p. 78\)](#)
- [IAM 사용자의 계정 암호 정책 설정 \(p. 79\)](#)
- [IAM 정책 관리 \(p. 377\)](#)
- [IAM 사용자가 자신의 암호를 변경하는 방법 \(p. 87\)](#)

IAM 사용자가 자신의 암호를 변경하는 방법

자신의 IAM 사용자 암호를 변경할 수 있는 권한이 부여된 경우 AWS Management 콘솔의 특별 페이지를 사용하여 이 작업을 수행할 수 있습니다. AWS CLI 또는 AWS API도 사용할 수 있습니다.

주제

- [필요한 권한 \(p. 87\)](#)
- [IAM 사용자가 자신의 암호를 변경하는 방법\(콘솔\) \(p. 87\)](#)
- [IAM 사용자가 자신의 암호를 변경하는 방법\(AWS CLI 또는 AWS API\) \(p. 88\)](#)

필요한 권한

자신의 IAM 사용자에 대한 암호를 변경하려면 다음 정책에 따른 권한이 있어야 합니다. [AWS: IAM 사용자가 My Security Credentials\(내 보안 자격 증명\) 페이지에서 자신의 콘솔 암호를 변경할 수 있도록 허용합니다. \(p. 350\)](#)

IAM 사용자가 자신의 암호를 변경하는 방법(콘솔)

다음 절차는 IAM 사용자가 AWS Management 콘솔을 사용하여 자신의 암호를 변경하는 방법을 설명합니다.

자신의 IAM 사용자 암호를 변경하려면(콘솔)

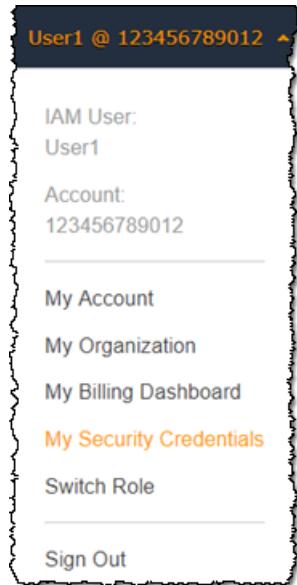
1. AWS 계정 ID 또는 계정 별칭, IAM 사용자 이름, 암호를 사용하여 [IAM 콘솔](#)에 로그인합니다.

Note

사용자 편의를 위해 AWS 로그인 페이지는 브라우저 쿠키를 사용하여 IAM 사용자 이름 및 계정 정보를 기억합니다. 이전에 다른 사용자로 로그인한 경우, 페이지 하단 근처의 [Sign in to a different account]를 선택하여 기본 로그인 페이지로 돌아갑니다. 여기서 AWS 계정 ID 또는 계정 별칭을 입력하면 계정의 IAM 사용자 로그인 페이지로 리디렉션됩니다.

AWS 계정 ID를 받으려면 관리자에게 문의하십시오.

2. 오른쪽 상단의 탐색 모음에서 사용자 이름을 선택한 다음 [My Security Credentials\(내 보안 자격 증명\)](#)를 선택합니다.



3. AWS IAM Credentials(AWS IAM 자격 증명) 탭에서 비밀번호 변경을 선택합니다.
4. Current password(현재 암호)에 현재 암호를 입력합니다. New password(새 암호) 및 Confirm new password(새 암호 확인)에 새 암호를 입력합니다. 그런 다음 Change password(암호 변경)를 클릭합니다.

Note

계정에 암호 정책이 있는 경우에는 새 암호가 해당 정책의 요건을 따라야 합니다. 자세한 내용은 [IAM 사용자의 계정 암호 정책 설정 \(p. 79\)](#) 단원을 참조하십시오.

IAM 사용자가 자신의 암호를 변경하는 방법(AWS CLI 또는 AWS API)

다음 절차는 IAM 사용자가 AWS CLI 또는 AWS API를 사용하여 자신의 암호를 변경하는 방법을 설명합니다.

자신의 IAM 암호를 변경하려면 다음을 사용하십시오.

- AWS CLI: `aws iam change-password`
- AWS API: `ChangePassword`

IAM 사용자의 액세스 키 관리

Follow us on Twitter

Note

웹 사이트에서 Amazon 제품을 팔기 위해 Product Advertising API를 구성하기 위해 이 주제를 찾았다면 다음 주제들 단원을 참조하십시오.

- [Product Advertising API로 시작하기](#)
- [Product Advertising API 개발자로서 시작하기](#)

액세스 키는 IAM 사용자 또는 AWS 계정 루트 사용자에 대한 장기 자격 증명입니다. 액세스 키를 사용하여 AWS CLI 또는 AWS API에 대한 프로그래밍 요청에 서명할 수 있습니다(직접 또는 AWS SDK를 사용하여). 자세한 내용은 Amazon Web Services 일반 참조의 [AWS API 요청 서명](#)을 참조하십시오.

액세스 키는 액세스 키 ID(예: AKIAIOSFODNN7EXAMPLE)와 보안 액세스 키(예: wJalrXUtnFEMI/K7MDENG/bPxRfCYEXAMPLEKEY)의 2가지 부분으로 구성됩니다. 사용자 이름 및 암호와 같이 액세스 키 ID와 보안 액세스 키를 함께 사용하여 요청을 인증해야 합니다. 사용자 이름과 암호를 관리하는 것처럼 안전하게 액세스 키를 관리합니다.

Important

[정식 사용자 ID를 찾는 데](#) 도움이 되더라도 액세스 키를 제3자에게 제공하지 마십시오. 이로 인해 다른 사람에게 계정에 대한 영구 액세스를 제공하게 될 수 있습니다.

가장 좋은 방법은 액세스 키 대신 임시 보안 자격 증명(IAM 역할)을 사용하고 모든 AWS 계정 루트 사용자 액세스 키는 비활성화하는 것입니다. 자세한 내용은 Amazon Web Services 일반 참조의 [AWS 액세스 키 관리 모범 사례](#) 단원을 참조하십시오.

장기 액세스 키를 사용해야 하는 경우 액세스 키(액세스 키 ID 및 보안 액세스 키)를 생성, 수정, 보기 또는 교체할 수 있습니다. 최대 두 개의 액세스 키를 가질 수 있습니다. 이렇게 하면 모범 사례에 따라 활성 키를 교체할 수 있습니다.

액세스 키 페어를 생성할 때는 액세스 키 ID와 보안 액세스 키를 안전한 위치에 저장합니다. 보안 액세스 키는 생성할 때만 사용할 수 있습니다. 보안 액세스 키를 분실한 경우 액세스 키를 삭제하고 새 키를 생성해야 합니다. 자세한 내용은 [분실하거나 잊어버린 암호 또는 액세스 키 재설정](#) (p. 95) 단원을 참조하십시오.

주제

- [필요한 권한](#) (p. 89)
- [액세스 키 관리\(콘솔\)](#) (p. 90)
- [액세스 키 관리\(AWS CLI\)](#) (p. 92)
- [액세스 키 관리\(AWS API\)](#) (p. 92)
- [액세스 키 교체](#) (p. 92)

필요한 권한

자신의 IAM 사용자에 대한 액세스 키를 생성하려면 다음 정책에 따른 권한이 있어야 합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "CreateOwnAccessKeys",  
            "Effect": "Allow",  
            "Action": [  
                "iam:CreateAccessKey",  
                "iam:GetUser",  
                "iam>ListAccessKeys"  
            ],  
            "Resource": "arn:aws:iam::*:user/${aws:username}"  
        }  
    ]  
}
```

자신의 IAM 사용자에 대한 액세스 키를 교체하려면 다음 정책에 따른 권한이 있어야 합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "ManageOwnAccessKeys",  
            "Effect": "Allow",  
            "Action": [  
                "iam:DeleteAccessKey",  
                "iam:UpdateAccessKey"  
            ]  
        }  
    ]  
}
```

```
        "iam:CreateAccessKey",
        "iam>DeleteAccessKey",
        "iam:GetAccessKeyLastUsed",
        "iam:GetUser",
        "iam>ListAccessKeys",
        "iam:UpdateAccessKey"
    ],
    "Resource": "arn:aws:iam::*:user/${aws:username}"
}
]
```

액세스 키 관리(콘솔)

AWS Management 콘솔을 사용하여 IAM 사용자의 액세스 키를 관리할 수 있습니다.

자신의 IAM 사용자 액세스 키를 생성, 수정 또는 삭제하려면(콘솔)

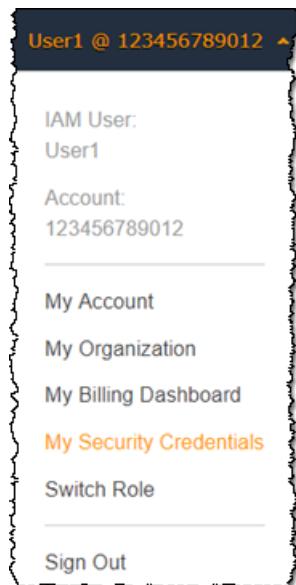
1. AWS 계정 ID 또는 계정 별칭, IAM 사용자 이름, 암호를 사용하여 [IAM 콘솔](#)에 로그인합니다.

Note

사용자 편의를 위해 AWS 로그인 페이지는 브라우저 쿠키를 사용하여 IAM 사용자 이름 및 계정 정보를 기억합니다. 이전에 다른 사용자로 로그인한 경우, 페이지 하단 근처의 [Sign in to a different account]를 선택하여 기본 로그인 페이지로 돌아갑니다. 여기서 AWS 계정 ID 또는 계정 별칭을 입력하면 계정의 IAM 사용자 로그인 페이지로 리디렉션됩니다.

AWS 계정 ID를 받으려면 관리자에게 문의하십시오.

2. 오른쪽 상단의 탐색 모음에서 사용자 이름을 선택한 다음 My Security Credentials(내 보안 자격 증명)를 선택합니다.



3. AWS IAM Credentials(AWS IAM 자격 증명) 탭의 Access keys for CLI, SDK, and API access(CLI, SDK 및 API 액세스를 위한 액세스 키) 섹션에서 다음 작업을 수행합니다.

- 액세스 키를 생성하려면 Create access key(액세스 키 생성)을 선택합니다. 그런 다음 Download .csv file(.csv 파일 다운로드)를 선택하여 액세스 키 ID 및 보안 액세스 키를 컴퓨터에 .csv 파일로 저장합니다. 안전한 위치에 파일을 저장합니다. 이 대화 상자를 닫은 후에는 보안 액세스 키에 다시 액세스할 수 없습니다. .csv 파일을 다운로드한 후 닫기를 클릭합니다. 액세스 키를 생성하면 키 페어가 기본적으로 활성화되므로 해당 페어를 즉시 사용할 수 있습니다.

- 활성 상태의 액세스 키를 비활성화하려면 비활성화를 선택합니다.
- 비활성 상태의 액세스 키를 다시 활성화하려면 활성화를 선택합니다.
- 액세스 키를 삭제하려면 행의 맨 왼쪽에 있는 X 버튼을 선택합니다. 삭제를 선택하여 확인합니다. 액세스 키를 삭제하면 영구 삭제되어 되돌릴 수 없습니다. 그러나 언제든지 새 키를 만들 수 있습니다.

다른 IAM 사용자의 액세스 키를 생성, 수정 또는 삭제하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 Users(사용자)를 선택합니다.
3. 액세스 키를 관리하려는 사용자 이름을 선택한 다음 보안 자격 증명 탭을 선택합니다.
4. 액세스 키 섹션에서 다음 작업을 수행합니다.
 - 액세스 키를 생성하려면 Create access key(액세스 키 생성)을 선택합니다. 그런 다음 .csv 파일 다운로드를 선택하여 액세스 키 ID 및 보안 액세스 키를 컴퓨터에 CSV 파일로 저장합니다. 안전한 위치에 파일을 저장합니다. 이 대화 상자를 닫은 후에는 보안 액세스 키에 다시 액세스할 수 없습니다. CSV 파일을 다운로드한 후 닫기를 선택합니다. 액세스 키를 생성하면 키 페어가 기본적으로 활성화되므로 해당 페어를 즉시 사용할 수 있습니다.
 - 활성 상태의 액세스 키를 비활성화하려면 비활성화를 선택합니다.
 - 비활성 상태의 액세스 키를 다시 활성화하려면 활성화를 선택합니다.
 - 액세스 키를 삭제하려면 행의 맨 왼쪽에 있는 X 버튼을 선택합니다. 삭제를 선택하여 확인합니다. 액세스 키를 삭제하면 영구 삭제되어 되돌릴 수 없습니다. 그러나 언제든지 새 키를 만들 수 있습니다.

IAM 사용자에 대한 액세스 키를 나열하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 해당 사용자의 이름을 선택한 후 보안 자격 증명 탭을 선택합니다. 사용자의 액세스 키와 각 키의 상태가 표시됩니다.

Note

사용자의 액세스 키 ID만 표시됩니다. 보안 액세스 키는 키를 만들 때만 가져올 수 있습니다.

여러 IAM 사용자의 액세스 키 ID를 나열하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 필요할 경우 다음 단계를 통해 사용자 테이블에 액세스 키 ID 열을 추가합니다.
 - a. 테이블 위 맨 오른쪽에서 설정 아이콘()을 선택합니다.
 - b. Manage columns(열 관리)에서 액세스 키 ID를 선택합니다.
 - c. 닫기를 선택하여 사용자 목록으로 돌아갑니다.
4. 액세스 키 ID 열에는 각 액세스 키 ID가 표시되고 그 다음에 키의 상태가 표시됩니다. 예: 23478207027842073230762374023 (Active) 또는 22093740239670237024843420327 (Inactive).

이 정보를 사용하여 한 개 또는 두 개의 액세스 키를 가진 사용자의 액세스 키를 보고 복사할 수 있습니다. 액세스 키가 없는 사용자는 이 열에 없음이라고 표시됩니다.

Note

사용자의 액세스 키 ID와 상태만 표시됩니다. 보안 액세스 키는 키를 만들 때만 가져올 수 있습니다.

어떤 IAM 사용자가 특정 액세스 키를 소유하고 있는지 확인하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 검색 상자에 해당 사용자의 액세스 키 ID를 입력하거나 붙여 넣습니다.
4. 필요할 경우 다음 단계를 통해 사용자 테이블에 액세스 키 ID 열을 추가합니다.
 - a. 테이블 위 맨 오른쪽에서 설정 아이콘()을 선택합니다.
 - b. Manage columns(열 관리)에서 액세스 키 ID를 선택합니다.
 - c. 닫기를 선택하여 사용자 목록으로 돌아간 후 지정된 액세스 키를 소유하는 사용자로 필터링되었는지 확인합니다.

액세스 키 관리(AWS CLI)

AWS CLI에서 IAM 사용자의 액세스 키를 관리하려면 다음 명령을 실행합니다.

- 액세스 키 생성: `aws iam create-access-key`
- 액세스 키 비활성화 또는 다시 활성화: `aws iam update-access-key`
- 사용자의 액세스 키를 나열하려면: `aws iam list-access-keys`
- 가장 최근에 액세스 키를 사용한 시기 확인: `aws iam get-access-key-last-used`
- 액세스 키 삭제: `aws iam delete-access-key`

액세스 키 관리(AWS API)

AWS API에서 IAM 사용자의 액세스 키를 관리하려면 다음 작업을 호출합니다.

- 액세스 키 생성: `CreateAccessKey`
- 액세스 키 비활성화 또는 다시 활성화: `UpdateAccessKey`
- 사용자의 액세스 키를 나열하려면: `ListAccessKeys`
- 가장 최근에 액세스 키를 사용한 시기 확인: `GetAccessKeyLastUsed`
- 액세스 키 삭제: `DeleteAccessKey`

액세스 키 교체

최상의 보안을 위해 IAM 사용자 액세스 키를 정기적으로 교체(변경)하는 것이 좋습니다. 관리자가 필요한 권한을 부여한 경우 사용자 고유의 액세스 키를 교체할 수 있습니다.

관리자가 사용자에게 액세스 키를 직접 교체할 수 있는 권한을 부여하는 방법에 대한 자세한 내용은 [AWS: IAM 사용자가 My Security Credentials\(내 보안 자격 증명\) 페이지에서 자신의 암호, 액세스 키 및 SSH 퍼블릭 키를 관리할 수 있도록 허용합니다. \(p. 351\)](#) 단원을 참조하십시오. 또한, 모든 IAM 사용자가 주기적으로 암호를 교체하도록 요구하는 암호 정책을 계정에 적용할 수 있습니다. 얼마나 자주 교체하도록 할지 선택할 수 있습니다. 자세한 내용은 [IAM 사용자의 계정 암호 정책 설정 \(p. 79\)](#) 단원을 참조하십시오.

Important

가장 좋은 방법은 AWS 계정 루트 사용자를 사용하지 않는 것입니다. AWS 계정 루트 사용자 자격 증명을 사용할 경우 그 자격 증명도 정기적으로 교체할 것을 권장합니다. 계정 암호 정책은 루트 사용자 자격 증명에는 적용되지 않습니다. IAM 사용자는 AWS 계정 루트 사용자의 자격 증명을 관리할 수 없으므로 루트 사용자의 자격 증명(사용자의 자격 증명이 아님)을 사용하여 루트 사용자 자격 증명을 변경해야 합니다. AWS의 일상적인 작업에서는 루트 사용자를 사용하지 않는 것이 좋습니다.

주제

- [IAM 사용자 액세스 키 교체\(콘솔\)](#) (p. 93)
- [액세스 키 교체\(AWS CLI\)](#) (p. 94)
- [액세스 키 교체\(AWS API\)](#) (p. 95)

IAM 사용자 액세스 키 교체(콘솔)

AWS Management 콘솔에서 액세스 키를 교체할 수 있습니다.

애플리케이션을 종단하지 않고 IAM 사용자의 액세스 키를 교체하려면(콘솔)

1. 최초 액세스 키가 활성 상태일 때 두 번째 액세스 키를 만듭니다.

- a. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
- b. 탐색 창에서 Users(사용자)를 선택합니다.
- c. 해당 사용자의 이름을 선택한 후 보안 자격 증명 탭을 선택합니다.
- d. Create access key(액세스 키 생성)을 선택하고 Download .csv file(.csv 파일 다운로드)를 선택하여 액세스 키 ID와 보안 액세스 키를 컴퓨터의 .csv 파일에 저장합니다. 안전한 위치에 파일을 저장합니다. 이 단계를 끝낸 후에는 보안 액세스 키에 다시 액세스할 수 없습니다. .csv 파일을 다운로드 한 후 닫기를 클릭합니다.

새 액세스 키는 기본적으로 활성화됩니다. 따라서 사용자에게 두 개의 활성 액세스 키가 생깁니다.

2. 새 액세스 키를 사용하도록 모든 애플리케이션과 도구를 업데이트합니다.
3. 가장 오래된 액세스 키의 Last used(마지막 사용) 열을 검토하여 최초 액세스 키가 아직 사용 중인지 확인합니다. 한 가지 접근 방식은 며칠을 기다린 다음 계속하기 전에 사용된 적이 있는 기존 액세스 키가 있는지 확인하는 것입니다.
4. Last used(마지막 사용) 열에 오래된 키가 사용된 적이 없다고 표시되더라도 최초 액세스 키를 바로 삭제하지 않는 것이 좋습니다. 대신 Make inactive(비활성화)를 선택하여 최초 액세스 키를 비활성화합니다.
5. 새 액세스 키만 사용하여 애플리케이션이 작동 중인지 확인합니다. 원래 액세스 키를 계속 사용하는 어떤 애플리케이션도 AWS 리소스에 더 이상 액세스할 수 없기 때문에 이 시점에 작업을 종단합니다. 그러한 애플리케이션 또는 도구를 찾는다면 활성화(Make active)를 선택하여 최초 액세스 키를 다시 활성화 할 수 있습니다. 그런 다음 Step 3 (p. 93) 단원으로 돌아가 이 애플리케이션을 업데이트하여 새 키를 사용하십시오.
6. 일정 기간 기다린 후 모든 애플리케이션과 도구가 업데이트되었는지 확인한 뒤에 최초 액세스 키를 삭제할 수 있습니다:
 - a. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
 - b. 탐색 창에서 사용자를 선택합니다.
 - c. 해당 사용자의 이름을 선택한 후 보안 자격 증명 탭을 선택합니다.
 - d. 삭제할 액세스 키를 찾아 해당 행 맨 오른쪽에 있는 X 버튼을 선택합니다. 삭제를 선택하여 확인합니다.

액세스 키 교체 시점을 결정하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 필요할 경우 다음 단계를 통해 사용자 테이블에 Access key age(액세스 키 수명) 열을 추가합니다.
 - a. 테이블 위 맨 오른쪽에서 설정 아이콘()을 선택합니다.
 - b. Manage Columns(열 관리)에서 Access key age(액세스 키 수명)을 선택합니다.
 - c. 닫기를 선택하여 사용자 목록으로 돌아갑니다.
4. Access key age(액세스 키 수명) 열에는 가장 오래된 활성 액세스 키가 생성된 이후로 경과한 일수가 표시됩니다. 이 정보를 사용하여 교체가 필요한 액세스 키를 소유한 사용자를 확인할 수 있습니다. 액세스 키가 없는 사용자는 이 열에 없음이라고 표시됩니다.

액세스 키 교체(AWS CLI)

AWS Command Line Interface에서 액세스 키를 교체할 수 있습니다.

애플리케이션을 중단하지 않고 액세스 키를 교체하려면(AWS CLI)

1. 최초 액세스 키가 활성 상태일 때 두 번째 액세스 키를 만들면 이 키도 기본적으로 활성 상태가 됩니다. 다음 명령을 실행합니다.
 - `aws iam create-access-key`따라서 사용자에게 두 개의 활성 액세스 키가 생깁니다.
2. 새 액세스 키를 사용하도록 모든 애플리케이션과 도구를 업데이트합니다.
3. 다음 명령을 사용하여 최초 액세스 키가 아직 사용 중인지 확인합니다.
 - `aws iam get-access-key-last-used`

한 가지 접근 방식은 며칠을 기다린 다음 계속하기 전에 사용된 적이 있는 기존 액세스 키가 있는지 확인하는 것입니다.

4. Step 3 단계를 통해 기존 키를 사용한 적이 없다는 것이 밝혀진 경우 최초의 액세스 키를 즉시 삭제하지 말 것을 권장합니다. 그 대신 다음 명령을 사용하여 최초 액세스 키의 상태를 `Inactive`로 변경하십시오.
 - `aws iam update-access-key`
5. 새 액세스 키만 사용하여 애플리케이션이 작동 중인지 확인합니다. 원래 액세스 키를 계속 사용하는 어떤 애플리케이션도 AWS 리소스에 더 이상 액세스할 수 없기 때문에 이 시점에 작업을 중단합니다. 그러한 애플리케이션 또는 도구를 찾는다면 그 상태를 `Active`로 되돌려 최초 액세스 키를 다시 활성화할 수 있습니다. 그런 다음 Step 2 단계로 돌아가 이 애플리케이션을 업데이트해 새 키를 사용하십시오.
6. 일정 기간 기다린 후 모든 애플리케이션과 도구가 업데이트되었는지 확인한 뒤에 다음 명령을 사용하여 최초 액세스 키를 삭제할 수 있습니다.
 - `aws iam delete-access-key`

자세한 내용은 다음 단원을 참조하십시오.

- **IAM 사용자의 액세스 키 교체 방법.** AWS 보안 블로그의 이 게시물에서는 키 교체에 대한 자세한 내용을 설명합니다.
- **IAM 모범 사례 (p. 43).** 이 페이지에서는 AWS 리소스를 보호하기 위한 일반적인 권장 사항을 설명합니다.

액세스 키 교체(AWS API)

AWS API를 사용하여 액세스 키를 교체할 수 있습니다.

애플리케이션을 중단하지 않고 액세스 키를 교체하려면(AWS API)

1. 최초 액세스 키가 활성 상태일 때 두 번째 액세스 키를 만들면 이 키도 기본적으로 활성 상태가 됩니다. 다음 작업을 호출합니다.

- [CreateAccessKey](#)

따라서 사용자에게 두 개의 활성 액세스 키가 생깁니다.

2. 새 액세스 키를 사용하도록 모든 애플리케이션과 도구를 업데이트합니다.
3. 다음 연산을 호출하여 최초 액세스 키가 아직 사용 중인지 확인합니다.

- [GetAccessKeyLastUsed](#)

한 가지 접근 방식은 며칠을 기다린 다음 계속하기 전에 사용된 적이 있는 기존 액세스 키가 있는지 확인하는 것입니다.

4. [Step 3](#) 단계를 통해 기존 키를 사용한 적이 없다는 것이 밝혀진 경우 최초의 액세스 키를 즉시 삭제하지 말 것을 권장합니다. 그 대신 다음 연산을 호출하여 최초 액세스 키의 상태를 `Inactive`로 변경하십시오.

- [UpdateAccessKey](#)

5. 새 액세스 키만 사용하여 애플리케이션이 작동 중인지 확인합니다. 원래 액세스 키를 계속 사용하는 어던 애플리케이션도 AWS 리소스에 더 이상 액세스할 수 없기 때문에 이 시점에 작업을 중단합니다. 그러한 애플리케이션 또는 도구를 찾는다면 그 상태를 `Active`로 되돌려 최초 액세스 키를 다시 활성화할 수 있습니다. 그런 다음 [Step 2](#) 단계로 돌아가 이 애플리케이션을 업데이트해 새 키를 사용하십시오.

6. 일정 기간 기다린 후 모든 애플리케이션과 도구가 업데이트되었는지 확인한 뒤에 다음 연산을 호출하여 최초 액세스 키를 삭제할 수 있습니다.

- [DeleteAccessKey](#)

자세한 내용은 다음 단원을 참조하십시오.

- [IAM 사용자의 액세스 키 교체 방법](#). AWS 보안 블로그의 이 게시물에서는 키 교체에 대한 자세한 내용을 설명합니다.
- [IAM 모범 사례 \(p. 43\)](#). 이 페이지에서는 AWS 리소스를 보호하기 위한 일반적인 권장 사항을 설명합니다.

분실하거나 잊어버린 암호 또는 액세스 키 재설정

암호 또는 액세스 키를 분실하거나 잊어버린 경우, IAM에서 검색할 수 없습니다. 그 대신 다음과 같은 방법으로 재설정할 수는 있습니다.

- AWS 계정 루트 사용자 암호 – 사용자 암호를 잊어버린 경우, AWS Management 콘솔에서 암호를 재설정 할 수 있습니다. 자세한 내용은 이 주제의 후반부에 나오는 [the section called “잊거나 분실한 루트 사용자 암호 재설정” \(p. 96\)](#) 단원을 참조하십시오.
- AWS 계정 액세스 키 – 계정 액세스 키를 잊었다면 기존 액세스 키를 비활성화하지 않고 액세스 키를 새로 만들어도 됩니다. 기존 키를 사용하고 있지 않았다면 삭제하면 됩니다. 자세한 내용은 [루트 사용자를 위한 액세스 키 생성 \(p. 292\)](#) 및 [루트 사용자로부터 액세스 키 삭제하기 \(p. 292\)](#) 단원을 참조하십시오.
- IAM 사용자 암호 – IAM 사용자인데 암호를 잊었다면 관리자에게 암호를 재설정해 달라고 부탁해야 합니다. 관리자가 암호를 관리하는 방법에 대한 자세한 내용은 [IAM 사용자의 암호 관리 \(p. 82\)](#) 단원을 참조하십시오.

- IAM 사용자 액세스 키 – IAM 사용자인데 액세스 키를 잊은 경우에는 새 액세스 키가 필요합니다. 고유한 액세스 키를 생성할 권한이 있다면 [액세스 키 관리\(콘솔\) \(p. 90\)](#) 단원에서 새 액세스 키 생성에 관한 지침을 찾아보십시오. 필요한 권한이 없으면 관리자에게 액세스 키를 새로 생성해 달라고 부탁해야 합니다. 예전 키를 아직 사용하고 있다면 관리자에게 예전 키를 삭제하지 말라고 요청하십시오. 관리자가 액세스 키를 관리하는 방법에 대한 자세한 내용은 [IAM 사용자의 액세스 키 관리 \(p. 88\)](#) 단원을 참조하십시오.

AWS 모범 사례 (p. 48)를 따라 주기적으로 암호와 AWS 액세스 키를 변경해야 합니다. AWS에서는 교체를 통해 액세스 키를 변경합니다. 이는 키를 새로 생성하고, 새로 만든 키를 사용하도록 애플리케이션을 구성한 다음, 이전 키를 삭제한다는 의미입니다. 이런 이유만으로도 동시에 2개의 액세스 키 페어를 활성화하도록 허용합니다. 자세한 내용은 [액세스 키 교체 \(p. 92\)](#) 단원을 참조하십시오.

잊거나 분실한 루트 사용자 암호 재설정

AWS 계정을 처음 생성할 때 이메일 주소와 암호를 입력했습니다. 그 주소와 암호가 바로 AWS 계정 루트 사용자 자격 증명입니다. 루트 사용자 암호를 잊어버린 경우, AWS Management 콘솔에서 암호를 재설정할 수 있습니다.

루트 사용자 암호를 재설정하려면:

1. AWS 계정 이메일 주소를 사용하여 [AWS Management 콘솔](#)에 루트 사용자로 로그인합니다.

Note

IAM 사용자 자격 증명으로 [AWS Management 콘솔](#)에 로그인되어 있다면 먼저 로그아웃해야 루트 사용자 암호를 재설정할 수 있습니다. 해당 계정의 IAM 사용자 로그인 페이지가 표시되면, 페이지 하단에 있는 루트 계정 자격 증명을 이용한 로그인을 선택합니다. 필요하면 계정 이메일 주소를 사용하여 루트 사용자 로그인(Root user sign in) 페이지에 액세스합니다.

2. 비밀번호가 생각나지 않는 경우를 선택합니다.
3. 계정을 생성할 때 사용한 이메일 주소를 입력합니다. 그런 다음 CAPTCHA 텍스트를 입력하고 계속을 선택합니다.
4. AWS 계정과 연결된 이메일로 Amazon Web Services의 메시지가 왔는지 점검합니다. @amazon.com 또는 @aws.amazon.com으로 끝나는 주소에서 보낸 이메일입니다. 이메일 지침을 따릅니다. 계정으로 이메일이 오지 않았으면 스팸 폴더를 점검합니다. 그 이메일에 더 이상 액세스할 수 없는 경우에는 [예전 계정에 액세스해야 합니다 \(p. 451\)](#) 단원을 참조하십시오.

AWS에서 멀티 팩터 인증(MFA) 사용하기

 Follow us on Twitter

보안 강화를 위해 멀티 팩터 인증(MFA)을 구성하여 AWS 리소스를 보호하는 것이 좋습니다.

주제

- [MFA란 무엇입니까? \(p. 97\)](#)
- [MFA 디바이스 활성화 \(p. 97\)](#)
- [MFA 상태 확인 \(p. 114\)](#)
- [가상 및 하드웨어 MFA 디바이스 재동기화 \(p. 115\)](#)
- [MFA 디바이스 비활성화 \(p. 119\)](#)
- [MFA 디바이스 분실 또는 작동 중단 시 문제 해결 \(p. 121\)](#)
- [MFA 보호 API 액세스 구성 \(p. 122\)](#)
- [MFA 조건이 포함된 샘플 정책 \(p. 128\)](#)
- [샘플 코드: 멀티 팩터 인증이 포함된 자격 증명 요청하기 \(p. 130\)](#)

MFA란 무엇입니까?

MFA는 사용자가 AWS 웹 사이트 또는 서비스에 액세스할 때 사용자의 정규 로그인 자격 증명 외에도 AWS 가 지원되는 MFA 메커니즘의 고유 인증을 제출하라고 요청함으로써 보안을 더욱 강화합니다.

- 가상 MFA 디바이스 스마트폰 또는 기타 모바일 디바이스에서 실행되며 물리적 디바이스를 에뮬레이션하는 소프트웨어 애플리케이션입니다. 디바이스가 동기화된 1회 암호 알고리즘에 따라 여섯 자리 숫자 코드를 생성합니다. 사용자는 로그인할 때 두 번째 웹페이지에서 디바이스의 유효 코드를 입력해야 합니다. 사용자에게 할당된 각 가상 MFA 디바이스는 고유해야 합니다. 사용자는 다른 사용자의 가상 MFA 디바이스의 코드를 입력하여 인증할 수 없습니다. 가상 MFA 디바이스로 사용할 수 있도록 지원되는 몇 가지 앱의 목록은 [멀티 팩터 인증 단원](#)을 참조하십시오. AWS를 사용하여 가상 MFA 디바이스를 설정하기 위한 지침은 [가상 멀티 팩터 인증\(MFA\) 디바이스 활성화\(콘솔\)](#) (p. 99) 단원을 참조하십시오.
- U2F 보안 키. 컴퓨터의 USB 포트에 연결하는 디바이스입니다. U2F는 [FIDO Alliance](#)에서 호스팅하는 공개 인증 표준입니다. U2F 보안 키를 활성화하려면, 코드를 수동으로 입력하는 대신, 본인의 자격 증명을 입력한 다음 디바이스를 터치하여 로그인합니다. 지원되는 AWS U2F 보안 키에 대한 자세한 내용은 [멀티 팩터 인증 단원](#)을 참조하십시오. AWS를 사용하여 가상 U2F 보안 키를 설정하기 위한 지침은 [U2F 보안 키 활성화\(콘솔\)](#) (p. 102) 단원을 참조하십시오.
- 하드웨어 MFA 디바이스 동기화된 1회 암호 알고리즘에 따라 여섯 자리 숫자 코드를 생성하는 하드웨어 디바이스입니다. 사용자는 로그인할 때 두 번째 웹페이지에서 디바이스의 유효 코드를 입력해야 합니다. 사용자에게 할당된 각 MFA 디바이스는 고유해야 합니다. 사용자는 다른 사용자의 디바이스의 코드를 입력하여 인증받을 수 없습니다. 지원되는 하드웨어 MFA 디바이스에 대한 자세한 내용은 [멀티 팩터 인증 단원](#)을 참조하십시오. AWS를 사용하여 하드웨어 MFA 디바이스를 설정하기 위한 지침은 [하드웨어 MFA 디바이스 활성화\(콘솔\)](#) (p. 107) 단원을 참조하십시오.
- SMS 문자 메시지 기반 MFA. IAM 사용자 설정이 해당 사용자의 SMS 호환 모바일 디바이스의 전화번호를 포함하는 MFA 유형입니다. 사용자가 로그인하면 AWS가 SMS 문자 메시지로 여섯 자리 숫자 코드를 사용자의 모바일 디바이스로 전송합니다. 사용자는 로그인 시 두 번째 웹 페이지에서 이 코드를 입력해야 합니다. SMS 기반 MFA는 IAM 사용자만 사용할 수 있습니다. AWS 계정 루트 사용자에서는 이러한 유형의 MFA를 사용할 수 없습니다. SMS 문자 메시지 기반 MFA 활성화에 대한 자세한 내용은 [미리 보기 – SMS 문자 메시지 MFA 디바이스 활성화](#) (p. 112) 단원을 참조하십시오.

Note

AWS는 곧 SMS 멀티 팩터 인증(MFA) 지원을 종료할 예정입니다. 신규 고객은 이 기능을 미리 볼 수 없습니다. 기존 고객은 [가상\(소프트웨어 기반\) MFA 디바이스](#) (p. 99), [U2F 보안 키](#) (p. 102) 또는 [하드웨어 MFA 디바이스](#) (p. 107) 등 MFA의 대체 방법 중 하나로 전환하는 것 이 좋습니다.

Tip

계정의 사용자 중에서 SMS MFA 디바이스가 할당된 사용자를 볼 수 있습니다. 이렇게 하려면 IAM 콘솔로 이동하여 탐색 창에서 사용자를 선택하고 표의 MFA 열에서 SMS가 표시된 사용자를 찾습니다.

Note

AWS 계정 루트 사용자에 대해 MFA를 활성화하면 해당 루트 사용자 자격 증명에만 적용됩니다. 이 계정의 IAM 사용자들은 자신의 자격 증명에 더하여 별도로 자격 증명을 갖게 되며, 이 별도의 자격 증명에 고유의 MFA가 구성됩니다.

AWS MFA에 대한 공통 질문 답변은 [AWS Multi-Factor Authentication FAQ](#)에서 확인할 수 있습니다.

MFA 디바이스 활성화

MFA 구성 단계는 사용하고 있는 MFA 디바이스의 유형에 따라 다릅니다.

주제

- [MFA 디바이스 활성화의 일반적 단계](#) (p. 98)

- [가상 멀티 팩터 인증\(MFA\) 디바이스 활성화\(콘솔\) \(p. 99\)](#)
- [U2F 보안 키 활성화\(콘솔\) \(p. 102\)](#)
- [하드웨어 MFA 디바이스 활성화\(콘솔\) \(p. 107\)](#)
- [미리 보기 – SMS 문자 메시지 MFA 디바이스 활성화 \(p. 112\)](#)
- [가상 MFA 디바이스 활성화 및 관리\(AWS CLI 또는 AWS API\) \(p. 113\)](#)

MFA 디바이스 활성화의 일반적 단계

다음 개요 절차에는 MFA를 설정하고 사용하는 방법이 설명되어 있으며, 관련된 정보에 대한 링크가 나와 있습니다.

1. 다음과 같은 MFA 디바이스를 가져옵니다. MFA 디바이스는 AWS 계정 루트 사용자 1개 또는 IAM 사용자 1명 당 단 1개를 활성화할 수 있습니다.
 - 가상 MFA 디바이스로, [표준 기반 TOTP\(시간 기반 일회용 암호\) 알고리즘인 RFC 6238과 호환되는 소프트웨어 애플리케이션](#). 태블릿이나 스마트폰과 같은 모바일 디바이스에 이 애플리케이션을 설치할 수 있습니다. 가상 MFA 디바이스로 사용할 수 있도록 지원되는 몇 가지 앱의 목록은 [멀티 팩터 인증](#) 단원을 참조하십시오.
 - [AWS 지원 구성](#) (p. 106)을 갖춘 U2F 보안 키(예: [멀티 팩터 인증](#) 페이지에서 논의한 U2F 디바이스)
 - 하드웨어 기반 MFA 디바이스(예: [멀티 팩터 인증](#) 페이지에서 논의한 AWS 지원 하드웨어 토큰 디바이스).
 - 표준 SMS 문자 메시지를 받을 수 있는 휴대폰.

메모

- SMS 기반 MFA를 사용하는 경우 해당 모바일 디바이스 이동 통신 사업자가 부과하는 문자 메시지 요금이 적용될 수 있습니다.
- SMS 기반 MFA는 IAM 사용자만 사용할 수 있으며 루트 사용자는 사용할 수 없습니다.

2. MFA 디바이스를 활성화합니다.

- 가상 또는 하드웨어 MFA 디바이스를 보유한 IAM 사용자: AWS Management 콘솔, AWS CLI 또는 IAM API에서 활성화합니다.
- SMS 문자 메시지를 수신할 수 있는 U2F 보안 키 또는 휴대폰을 보유한 IAM 사용자: AWS Management 콘솔에서만 활성화할 수 있습니다.
- (루트 사용자에게 지원되지 않는 SMS MFA를 제외한) 모든 유형의 MFA 디바이스를 보유한 AWS 계정 루트 사용자: AWS Management 콘솔에서만 활성화할 수 있습니다.

각 MFA 디바이스 유형 활성화에 대한 자세한 내용은 다음 페이지를 참조하십시오.

3. AWS 리소스에 로그인하거나 액세스할 때 MFA 디바이스를 사용합니다. 다음 사항에 유의하십시오.
 - U2F 보안 키: [U2F 보안 키 활성화\(콘솔\) \(p. 102\)](#)
 - 하드웨어 MFA 디바이스: [하드웨어 MFA 디바이스 활성화\(콘솔\) \(p. 107\)](#)
 - SMS MFA 디바이스: [미리 보기 – SMS 문자 메시지 MFA 디바이스 활성화 \(p. 112\)](#)
- 가상 MFA 디바이스: [가상 멀티 팩터 인증\(MFA\) 디바이스 활성화\(콘솔\) \(p. 99\)](#)

MFA 보호 API 작업에 액세스하려면 다음이 필요합니다.

- MFA 코드
- MFA 디바이스의 식별자(물리적 디바이스의 일련 번호나 가상 또는 AWS에 정의된 SMS 디바이스의 ARN)

- 일반 액세스 키 ID 및 보안 액세스 키.

메모

- U2F 보안 키 또는 SMS MFA 디바이스의 MFA 정보를 AWS STS API 작업으로 전달하여 임시 자격 증명을 요청할 수 없습니다.
- AWS CLI 명령 또는 AWS API 작업을 사용하여 [U2F 보안 키 \(p. 102\)](#)를 활성화할 수 없습니다.

자세한 내용은 [IAM 로그인 페이지에 MFA 디바이스 사용 \(p. 57\)](#) 단원을 참조하십시오.

가상 멀티 팩터 인증(MFA) 디바이스 활성화(콘솔)

스마트폰 또는 태블릿과 같은 모바일 디바이스를 가상 멀티 팩터 인증(MFA) 디바이스로 사용할 수 있습니다. 이 작업을 수행하려면 6자리 인증 코드를 생성하는 AWS 지원 모바일 앱을 설치합니다. 이러한 애플리케이션은 보안되지 않은 모바일 디바이스에서 실행될 수 있으므로 가상 MFA는 U2F 디바이스 또는 하드웨어 MFA 디바이스와 동일한 수준의 보안을 제공하지 않을 수 있습니다. 하드웨어 구매 승인을 기다리는 동안 또는 하드웨어 도착을 기다리는 동안 가상 MFA 디바이스를 사용하는 것이 좋습니다.

부분의 가상 MFA 모바일 앱은 여러 개의 가상 디바이스 생성을 지원하므로 여러 개의 AWS 계정이나 사용자에게 동일한 앱을 사용할 수 있습니다. 그러나 MFA 디바이스는 사용자 1명당 단 1개만 활성화할 수 있습니다.

스마트폰 또는 태블릿에서 사용할 수 있는 가상 MFA 앱 목록은 [멀티 팩터 인증](#) 단원을 참조하십시오. 단, AWS에서 사용하려면 가상 MFA 앱이 6자리 OTP를 생성해야 합니다.

Important

AWS에서 사용할 수 있도록 가상 MFA 디바이스를 구성할 때는 QR 코드 또는 보안 키를 안전한 곳에 저장하는 것이 좋습니다. 그렇게 하면 휴대폰을 잃어버리거나 어떤 이유로 MFA 소프트웨어 앱을 재설치해야 하는 경우 앱을 재구성해 동일한 가상 MFA를 사용할 수 있습니다. 그러므로 사용자 또는 루트 사용자를 위해 AWS에서 새로운 가상 MFA를 생성할 필요가 없습니다.

주제

- [필요한 권한 \(p. 99\)](#)
- [IAM 사용자에 대한 가상 MFA 디바이스 활성화\(콘솔\) \(p. 99\)](#)
- [AWS 계정 루트 사용자용 가상 MFA 디바이스 활성화\(콘솔\) \(p. 100\)](#)
- [가상 MFA 디바이스 교체 또는 "로테이션" \(p. 102\)](#)

필요한 권한

IAM 사용자의 가상 MFA 디바이스를 관리하려면 다음 정책에 따른 권한이 있어야 합니다. [AWS: MFA 인증 IAM 사용자가 My Security Credentials\(내 보안 자격 증명\) 페이지에서 자신의 MFA 디바이스를 관리할 수 있도록 허용합니다. \(p. 348\)](#)

IAM 사용자에 대한 가상 MFA 디바이스 활성화(콘솔)

AWS Management 콘솔에서 IAM을 사용하여 계정의 IAM 사용자를 위한 가상 MFA 디바이스를 활성화 및 관리할 수 있습니다. AWS CLI 또는 AWS API를 사용하여 MFA 장치를 활성화하고 관리하려면 [가상 MFA 디바이스 활성화 및 관리\(AWS CLI 또는 AWS API\) \(p. 113\)](#) 단원을 참조하십시오.

Note

MFA를 구성하려면 사용자의 가상 MFA 디바이스가 호스팅되는 하드웨어에 대한 물리적 액세스가 필요합니다. 예를 들어, 스마트폰에서 가상 MFA 디바이스를 실행하는 사용자에게 MFA를 구성할 수 있습니다. 이 경우 마법사를 완료하기 위해 스마트폰을 사용할 수 있어야 합니다. 이러한 이유로 사용자가 자신의 가상 MFA 디바이스를 직접 구성 및 관리할 수 있도록 허용하는 것이 좋습니다. 이 경우에는 사용자에게 필요한 IAM 작업 권한을 부여해야 합니다. 이러한 작업 권한을 부여하는 IAM 정책에 대한 자세한 내용과 예는 [AWS: MFA 인증 IAM 사용자가 My Security Credentials\(내 보안 자](#)

[격증명](#) 페이지에서 자신의 MFA 디바이스를 관리할 수 있도록 허용합니다. (p. 348) 단원을 참조하십시오.

IAM 사용자에 대한 가상 MFA 디바이스 활성화(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 Users(사용자)를 선택합니다.
3. 사용자 이름 목록에서 원하는 MFA 사용자 이름을 선택합니다.
4. Security credentials(보안 자격 증명) 탭을 선택합니다. Assigned MFA device(할당된 MFA 디바이스) 옆의 관리를 선택합니다.
5. Manage MFA Device(할당된 MFA 디바이스) 마법사에서 Virtual MFA device(가상 MFA 디바이스 비활성화)를 선택한 후 계속을 선택합니다.

IAM은 QR 코드 그래픽을 포함하여 가상 MFA 디바이스의 구성 정보를 생성 및 표시합니다. 그래픽은 QR 코드를 지원하지 않는 디바이스 상에서 수동 입력할 수 있는 '보안 구성 키'를 표시한 것입니다.

6. 가상 MFA 앱을 엽니다. (가상 MFA 디바이스의 호스팅에 사용되는 앱 목록은 [멀티 팩터 인증](#)을 참조하십시오) 가상 MFA 앱이 다수의 계정(다수의 가상 MFA 디바이스)을 지원하는 경우 옵션을 선택하여 새로운 계정(새로운 가상 MFA 디바이스)을 생성합니다.
7. MFA 앱의 QR 코드 지원 여부를 결정한 후 다음 중 한 가지를 실행합니다.
 - 마법사에서 Show QR code(QT 코드 표시)를 선택한 다음 해당 앱을 사용하여 QR 코드를 스캔합니다. 예를 들어 카메라 모양의 아이콘을 선택하거나 코드 스캔(Scan code)과 비슷한 옵션을 선택한 다음, 디바이스의 카메라를 사용하여 코드를 스캔합니다.
 - Manage MFA Device(MFA 디바이스 관리) 마법사에서 Show secret key(보안 키 표시)를 선택한 다음 MFA 앱에 보안 키를 입력합니다.

모든 작업을 마치면 가상 MFA 디바이스가 일회용 암호 생성을 시작합니다.

8. Manage MFA Device(MFA 디바이스 관리) 마법사의 MFA code 1(MFA 코드 1) 상자에 현재 가상 MFA 디바이스에 표시된 일회용 암호를 입력합니다. 디바이스가 새로운 일회용 암호를 생성할 때까지 최대 30초 기다립니다. 그런 다음 두 번째 일회용 암호를 MFA code 2(MFA 코드 2) 상자에 입력합니다. Assign MFA(MFA 할당)을 선택합니다.

Important

코드를 생성한 후 즉시 요청을 제출하십시오. 코드를 생성한 후 너무 오래 기다렸다 요청을 제출할 경우 MFA 디바이스가 사용자와 연결은 되지만 MFA 디바이스가 동기화되지 않습니다. 이는 시간 기반 일회용 암호(TOTP)가 잠시 후에 만료되기 때문입니다. 이 경우, [디바이스를 재동기화](#) (p. 115)할 수 있습니다.

이제 AWS에서 가상 MFA 디바이스를 사용할 준비가 끝났습니다. AWS Management 콘솔의 MFA 사용 방법에 대한 자세한 내용은 [IAM 로그인](#) 페이지에 MFA 디바이스 사용 (p. 57) 단원을 참조하십시오.

AWS 계정 루트 사용자용 가상 MFA 디바이스 활성화(콘솔)

AWS Management 콘솔을 사용하여 루트 사용자의 가상 MFA 디바이스를 구성 및 활성화할 수 있습니다. AWS 계정에 대해 MFA 디바이스를 활성화하려면 루트 사용자 자격 증명으로 AWS에 로그인해야 합니다. IAM 콘솔에서 또는 AWS CLI, AWS API나 Windows PowerShell용 도구 또는 기타 다른 명령줄 도구를 사용하면 AWS 계정 루트 사용자에 대해 MFA 디바이스를 활성화할 수 없습니다.

MFA 디바이스를 분실하거나, 도난당했거나, 디바이스가 작동하지 않을 경우에도 다른 인증 요소를 사용하여 로그인할 수 있습니다. MFA 디바이스로 로그인할 수 없는 경우에 사용자 계정으로 등록된 이메일 및 전화로 사용자 ID를 확인하여 로그인할 수 있습니다. 루트 사용자용 MFA를 활성화하기 전에 계정 설정과 연락처 정보를 검토하여 이메일 및 전화번호에 대한 액세스 권한이 있는지 확인하십시오. 다른 인증 요소를 사용하여 로그인하는 방법은 [MFA 디바이스 분실 또는 작동 중단 시 문제 해결](#) (p. 121) 단원을 참조하십시오. 이 기능을 비활성화하려면 [AWS Support](#)에 문의하십시오.

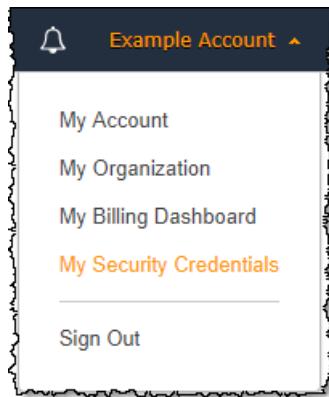
Note

MFA를 사용하여 로그인 및 인증 디바이스 문제 해결과 같은 다른 텍스트가 나타날 수 있습니다. 그러나 동일한 기능이 제공됩니다. 어느 경우든 대체 인증 팩터를 사용하여 계정 이메일 주소 및 전화 번호를 확인할 수 없는 경우 [AWS Support](#)에 문의하여 MFA 설정을 비활성화하십시오.

루트 사용자에서 사용할 목적으로 가상 MFA 디바이스를 구성 및 활성화하려면(콘솔)

1. AWS Management 콘솔에 로그인합니다.
2. 다음 중 하나를 수행하십시오.

- 옵션 1: 대시보드를 선택한 다음 Security Status(보안 상태)에서 Activate MFA on your 루트 사용자(루트에서 MFA 활성화)를 펼칩니다.
- 옵션 2: 탐색 표시줄 오른쪽에서 계정 이름을 선택하고 내 보안 자격 증명(My Security Credentials)을 선택합니다. 필요한 경우 보안 자격 증명으로 계속(Continue to Security Credentials)을 선택합니다. 그런 다음 해당 페이지의 멀티 팩터 인증(MFA) 섹션을 펼칩니다.



3. 이전 단계에서 선택한 옵션에 따라 MFA 관리 또는 MFA 활성화(Activate MFA)를 선택합니다.
4. 마법사에서 Virtual MFA device(가상 MFA 디바이스)를 선택한 후 계속을 선택합니다.
5. 가상 MFA 앱이 디바이스에 설치되었는지 확인한 후에 계속을 선택합니다. IAM은 QR 코드 그래픽을 포함하여 가상 MFA 디바이스의 구성 정보를 생성 및 표시합니다. 그래픽은 QR 코드를 지원하지 않는 디바이스 상에서 수동 입력할 수 있는 보안 구성 키를 표시한 것입니다.
6. MFA 디바이스 관리(Manage MFA Device) 마법사가 열려 있는 상태에서 디바이스에 가상 MFA 앱을 엽니다.
7. 가상 MFA 소프트웨어가 다수의 계정(다수의 가상 MFA 디바이스)을 지원하는 경우 옵션을 선택하여 새로운 계정(새로운 가상 디바이스)을 생성합니다.
8. 앱을 구성하는 가장 쉬운 방법은 앱을 사용하여 QR 코드를 스캔하는 것입니다. 코드를 스캔하지 못하는 경우 구성 정보를 직접 입력할 수 있습니다.
 - QR 코드를 사용하여 가상 MFA 디바이스를 구성하려면, 마법사에서 Show QR code(QT 코드 표시)를 선택합니다. 그리고 코드 스캔에 대한 앱 지침을 따릅니다. 예를 들어 카메라 모양의 아이콘을 터치하거나 계정 바코드 스캔(Scan account barcode)과 같은 명령을 터치한 다음, 디바이스의 카메라를 사용하여 QR 코드를 스캔할 수 있습니다.
 - Manage MFA Device(MFA 디바이스 관리) 마법사에서 Show secret key(보안 키 표시)을 선택한 다음 MFA 앱에 보안 키를 입력합니다.

Important

QR 코드 또는 보안 구성 키를 안전하게 백업하거나, 혹은 계정의 여러 가상 MFA 디바이스를 활성화하십시오. 예를 들어 가상 MFA 디바이스가 호스팅되어 있는 스마트폰을 분실하는 경우 가상 MFA 디바이스를 사용할 수 없습니다). 이 경우, 계정에 로그인할 수 없으므로 고객 서비스 센터에 연락하여 계정의 MFA 보호 기능을 제거해야 합니다.

Note

IAM에서 생성된 QR 코드와 보안 구성 키는 AWS 계정과 연동되기 때문에 다른 계정에서는 사용할 수 없습니다. 하지만 사용하던 MFA 디바이스에 대한 액세스 권한을 잃은 경우 재사용을 통해 계정에 대한 새로운 MFA 디바이스를 구성할 수 있습니다.

그 디바이스는 6자리 번호를 생성합니다.

9. Manage MFA Device(MFA 디바이스 관리) 마법사의 Authentication Code 1(인증 코드 1) 상자에 MFA 디바이스에 현재 표시된 6자리 번호를 입력합니다. 디바이스가 새 번호를 생성할 때까지 최대 30초를 기다린 후 새로 생성된 6자리 번호를 MFA code 2(MFA 코드 2) 상자에 입력합니다.

Important

코드를 생성한 후 즉시 요청을 제출하십시오. 코드를 생성한 후 너무 오래 기다렸다 요청을 제출할 경우 MFA 디바이스가 사용자와 연결은 되지만 MFA 디바이스가 동기화되지 않습니다. 이는 시간 기반 일회용 암호(TOTP)가 잠시 후에 만료되기 때문입니다. 이 경우, [디바이스를 재동기화 \(p. 115\)](#)할 수 있습니다.

10. Assign MFA(MFA 할당)을 선택한 다음 완료를 선택합니다.

이제 AWS에서 디바이스를 사용할 준비가 끝났습니다. AWS Management 콘솔의 MFA 사용 방법에 대한 자세한 내용은 [IAM 로그인 페이지에 MFA 디바이스 사용 \(p. 57\)](#) 단원을 참조하십시오.

가상 MFA 디바이스 교체 또는 "로테이션"

한 사용자에게는 한 번에 하나의 MFA 디바이스만 할당할 수 있습니다. 사용자가 디바이스를 분실하거나 이유를 불문하고 교체할 필요가 있을 경우, 먼저 기존 디바이스를 비활성화해야 합니다. 그런 다음, 해당 사용자를 위한 새 디바이스를 추가할 수 있습니다.

- 현재 다른 IAM 사용자와 연결되어 있는 디바이스를 비활성화하는 방법은 [MFA 디바이스 비활성화 \(p. 119\)](#) 단원을 참조하십시오.
- 다른 IAM 사용자를 위한 교체용 가상 MFA 디바이스를 추가하려면 위의 [IAM 사용자에 대한 가상 MFA 디바이스 활성화\(콘솔\) \(p. 99\)](#) 절차에 나와 있는 단계를 따르십시오.
- AWS 계정 루트 사용자용 교체 가상 MFA 디바이스를 추가하려면 이 주제 앞부분의 [AWS 계정 루트 사용자용 가상 MFA 디바이스 활성화\(콘솔\) \(p. 100\)](#) 절차에 나오는 단계를 따르십시오.

U2F 보안 키 활성화(콘솔)

U2F(Universal 2nd Factor) 보안 키는 AWS 리소스 보호에 사용할 수 있는 [MFA 디바이스 \(p. 96\)](#)의 한 유형입니다. U2F 보안 키를 컴퓨터의 USB 포트에 연결하여 다음의 지침에 따라 활성화할 수 있습니다. 활성화한 후 로그인 절차를 안전하게 완료하라는 메시지가 나타나면 터치합니다. 이미 다른 서비스에 U2F 보안 키를 사용 중이고 [AWS가 지원되는 구성 \(p. 106\)](#)(예: Yubico의 Yubikey 4 또는 5)을 보유한 경우 AWS에도 사용할 수 있습니다. 그렇지 않은 경우, AWS의 MFA에 U2F를 사용하려면 U2F 보안 키를 구입해야 합니다. 사양 및 구입 관련 정보는 [멀티 팩터 인증](#) 단원을 참조하십시오.

U2F는 [FIDO Alliance](#)에서 호스팅하는 공개 인증 표준입니다. AWS에서 U2F 키를 활성화하는 경우 U2F 보안 키가 AWS 전용의 새로운 키 페어를 생성합니다. 먼저 자격 증명을 입력합니다. 메시지가 나타나면 U2F 보안 키를 터치하여 AWS에서 야기된 인증 문제에 대응합니다. U2F 표준에 대해 자세히 알아보려면 [Universal 2nd Factor](#) 단원을 참조하십시오.

루트 사용자 또는 IAM 사용자별로 (어떤 종류든) 한 개의 MFA 디바이스를 활성화할 수 있습니다.

주제

- [필요한 권한 \(p. 103\)](#)
- [자신의 IAM 사용자에 대한 U2F 보안 키 활성화\(콘솔\) \(p. 103\)](#)
- [다른 IAM 사용자에 대한 U2F 보안 키 활성화\(콘솔\) \(p. 104\)](#)

- AWS 계정 루트 사용자에 대한 U2F 보안 키 활성화(콘솔) (p. 105)
- U2F 보안 키 교체 (p. 106)
- U2F 보안 키 사용에 지원되는 구성 (p. 106)

필요한 권한

중요한 MFA 관련 작업을 보호하면서 자신의 IAM 사용자에 대한 U2F 보안 키를 관리하려면 다음 정책에 따른 권한이 있어야 합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowManageOwnUserMFA",  
            "Effect": "Allow",  
            "Action": [  
                "iam:DeactivateMFADevice",  
                "iam:EnableMFADevice",  
                "iam:GetUser",  
                "iam>ListMFADevices",  
                "iam:ResyncMFADevice"  
            ],  
            "Resource": "arn:aws:iam::*:user/${aws:username}"  
        },  
        {  
            "Sid": "DenyAllExceptListedIfNoMFA",  
            "Effect": "Deny",  
            "NotAction": [  
                "iam:EnableMFADevice",  
                "iam:GetUser",  
                "iam>ListMFADevices",  
                "iam:ResyncMFADevice"  
            ],  
            "Resource": "arn:aws:iam::*:user/${aws:username}",  
            "Condition": {  
                "BoolIfExists": {  
                    "aws:MultiFactorAuthPresent": "false"  
                }  
            }  
        }  
    ]  
}
```

자신의 IAM 사용자에 대한 U2F 보안 키 활성화(콘솔)

AWS Management 콘솔에서만 자신의 IAM 사용자에 대한 U2F 보안 키를 활성화할 수 있으며, AWS CLI 또는 AWS API에서는 활성화할 수 없습니다.

Note

U2F 보안 키를 활성화하려면 디바이스에 물리적으로 액세스할 수 있어야 합니다.

자신의 IAM 사용자에 대한 U2F 보안 키를 활성화하려면(콘솔)

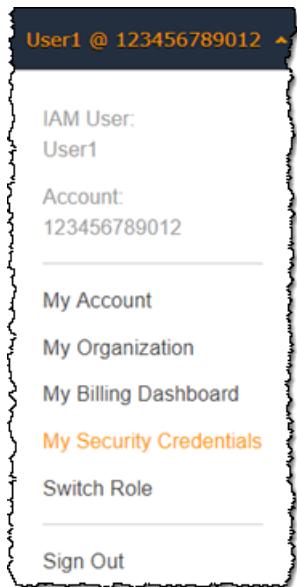
1. AWS 계정 ID 또는 계정 별칭, IAM 사용자 이름, 암호를 사용하여 [IAM 콘솔](#)에 로그인합니다.

Note

사용자 편의를 위해 AWS 로그인 페이지는 브라우저 쿠키를 사용하여 IAM 사용자 이름 및 계정 정보를 기억합니다. 이전에 다른 사용자로 로그인한 경우, 페이지 하단 근처의 [Sign in to a different account]를 선택하여 기본 로그인 페이지로 돌아갑니다. 여기서 AWS 계정 ID 또는 계정 별칭을 입력하면 계정의 IAM 사용자 로그인 페이지로 리디렉션됩니다.

AWS 계정 ID를 받으려면 관리자에게 문의하십시오.

2. 오른쪽 상단의 탐색 모음에서 사용자 이름을 선택한 다음 My Security Credentials(내 보안 자격 증명)를 선택합니다.



3. AWS IAM credentials(AWS IAM 자격 증명) 탭의 Multi-factor authentication(멀티 팩터 인증) 섹션에서 Manage MFA device(내 MFA 디바이스 관리)를 선택합니다.
4. Manage MFA device(MFA 디바이스 관리) 마법사에서 U2F security key(U2F 보안 키)를 선택한 다음 Continue(계속)를 선택합니다.
5. 컴퓨터의 USB 포트에 U2F 보안 키를 삽입합니다.



6. U2F 보안 키를 터치한 다음, U2F 설정이 완료되었을 때 닫기를 선택합니다.

AWS에서 U2F 보안 키를 사용할 준비가 끝났습니다. AWS Management 콘솔의 MFA 사용 방법에 대한 자세한 내용은 [IAM 로그인 페이지](#)에 MFA 디바이스 사용 (p. 57) 단원을 참조하십시오.

다른 IAM 사용자에 대한 U2F 보안 키 활성화(콘솔)

AWS Management 콘솔에서만 다른 IAM 사용자에 대한 U2F 보안 키를 활성화할 수 있으며, AWS CLI 또는 AWS API에서는 활성화할 수 없습니다.

다른 IAM 사용자에 대한 U2F 보안 키를 활성화하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 Users(사용자)를 선택합니다.
3. MFA를 활성화하려는 사용자의 이름을 선택한 다음 Security credentials(보안 자격 증명) 탭을 선택합니다.

4. Assigned MFA device(할당된 MFA 디바이스) 옆의 관리를 선택합니다.
5. Manage MFA device(MFA 디바이스 관리) 마법사에서 U2F security key(U2F 보안 키)를 선택한 다음 Continue(계속)를 선택합니다.
6. 컴퓨터의 USB 포트에 U2F 보안 키를 삽입합니다.



7. U2F 보안 키를 터치한 다음, U2F 설정이 완료되었을 때 닫기를 선택합니다.

AWS에서 U2F 보안 키를 사용할 준비가 끝났습니다. AWS Management 콘솔의 MFA 사용 방법에 대한 자세한 내용은 [IAM 로그인 페이지에 MFA 디바이스 사용 \(p. 57\)](#) 단원을 참조하십시오.

AWS 계정 루트 사용자에 대한 U2F 보안 키 활성화(콘솔)

AWS Management 콘솔에서만 루트 사용자에 대한 가상 MFA 디바이스를 구성하고 활성화할 수 있으며, AWS CLI 또는 AWS API에서는 이 작업을 수행할 수 없습니다.

U2F 보안 키를 분실했거나, 도난당했거나, 작동하지 않을 경우에도 다른 인증 요소를 사용하여 로그인할 수 있습니다. 다른 인증 요소를 사용하여 로그인하는 방법은 [MFA 디바이스 분실 또는 작동 중단 시 문제 해결 \(p. 121\)](#) 단원을 참조하십시오. 이 기능을 비활성화하려면 [AWS Support](#)에 문의하십시오.

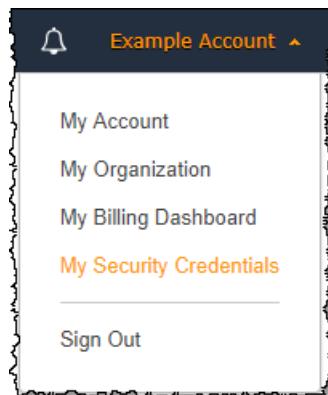
루트 사용자용 U2F 키를 활성화하려면(콘솔)

1. AWS 계정 이메일 주소와 암호를 사용하여 [AWS Management 콘솔](#)에 AWS 계정 루트 사용자로 로그인합니다.

Note

이전에 [IAM 사용자](#) 자격 증명으로 콘솔에 로그인한 경우, 브라우저가 이 설정을 기억하여 계정별로그인 페이지를 열 수도 있습니다. IAM 사용자 로그인 페이지에서는 AWS 계정 루트 사용자 자격 증명으로 로그인할 수 없습니다. IAM 사용자 로그인 페이지가 나타날 경우에는 페이지 하단에 있는 [Sign in using 루트 사용자 credentials]를 선택하여 기본 로그인 페이지로 돌아갑니다. 기본 로그인 페이지에서 AWS 계정의 이메일 주소와 암호를 입력합니다.

2. 탐색 모음의 오른쪽에서 계정 이름을 선택한 다음 My Security Credentials(내 보안 자격 증명)을 선택합니다. 필요한 경우 보안 자격 증명으로 계속(Continue to Security Credentials)을 선택합니다.



3. Multi-factor authentication (MFA)(멀티 팩터 인증(MFA)) 섹션을 확장합니다.
4. 이전 단계에서 선택한 옵션에 따라 MFA 관리 또는 MFA 활성화(Activate MFA)를 선택합니다.

5. 마법사에서 U2F security key(U2F 보안 키)를 선택한 후 계속을 선택합니다.
6. 컴퓨터의 USB 포트에 U2F 보안 키를 삽입합니다.



7. U2F 보안 키를 터치한 다음, U2F 설정이 완료되었을 때 닫기를 선택합니다.

AWS에서 U2F 보안 키를 사용할 준비가 끝났습니다. 다음에 루트 사용자 자격 증명을 사용하여 로그인할 때도 U2F 보안 키를 터치해 로그인 절차를 완료해야 합니다.

U2F 보안 키 교체

한 사용자에게는 한 번에 하나의 MFA 디바이스(가상, U2F 보안 키 또는 하드웨어)만 할당할 수 있습니다. 사용자가 U2F 보안 키를 분실하거나 이유를 불문하고 교체할 필요가 있을 경우, 먼저 기존 U2F 키를 비활성화해야 합니다. 그런 다음, 해당 사용자를 위한 새 MFA 디바이스를 추가할 수 있습니다.

- 현재 어떤 사용자와 연결되어 있는 디바이스를 비활성화하는 방법은 [MFA 디바이스 비활성화 \(p. 119\)](#) 단원을 참조하십시오.
- IAM 사용자에 대한 새 U2F 보안 키를 추가하려면 [U2F 보안 키 활성화\(콘솔\) \(p. 102\)](#) 단원을 참조하십시오.

새로운 U2F 보안 키에 대한 액세스 권한이 없는 경우 새로운 가상 MFA 디바이스 또는 하드웨어 MFA 디바이스를 활성화할 수 있습니다. 관련 지침을 보려면 다음 중 하나를 참조하십시오.

- [가상 멀티 팩터 인증\(MFA\) 디바이스 활성화\(콘솔\) \(p. 99\)](#)
- [하드웨어 MFA 디바이스 활성화\(콘솔\) \(p. 107\)](#)

U2F 보안 키 사용에 지원되는 구성

AWS에서 현재 지원되는 구성을 사용하여 U2F를 멀티 팩터 인증(MFA) 방법으로 사용할 수 있습니다. 이에는 AWS 및 U2F를 지원하는 브라우저가 지원되는 U2F 디바이스가 포함됩니다.

AWS 지원 U2F 디바이스

AWS는 현재 컴퓨터의 USB 포트에 연결되는 U2F 준수 보안 디바이스를 지원합니다.

지원되는 디바이스의 구입에 대한 자세한 내용은 [멀티 팩터 인증](#) 단원을 참조하십시오.

U2F 지원 브라우저

다음 브라우저들은 현재 U2F 보안 키의 사용을 지원합니다.

- Google Chrome 버전 38 이상.
- Opera 버전 40 이상.
- Mozilla Firefox 버전 57 이상.

Note

현재 U2F를 지원하는 대부분의 Firefox 버전은 기본적으로 지원을 활성화하지 않습니다. Firefox에서 U2F 지원을 활성하기 위한 지침은 [U2F 보안 키 문제 해결 \(p. 468\)](#) 단원을 참조하십시오.

브라우저 플러그인

현재 AWS는 U2F 표준을 기본적으로 지원하는 브라우저만을 지원합니다. AWS는 U2F 브라우저 지원을 추가하기 위한 플러그인 사용을 지원하지 않습니다. 또한 일부 브라우저 플러그인은 U2F 표준과 호환되지 않으며 U2F 보안 키와 연결할 때 예기치 않은 결과를 초래할 수 있습니다.

브라우저 플러그인 비활성화 및 기타 문제 해결을 위한 자세한 내용은 [U2F 보안 키를 활성화할 수 없습니다. \(p. 468\)](#) 단원을 참조하십시오.

모바일 환경

AWS에서는 현재 모바일 브라우저 또는 USB 방식이 아닌 U2F 디바이스에 대해서는 U2F 보안 키의 사용을 지원하지 않습니다.

AWS 콘솔 모바일 앱은 현재 MFA에 대한 U2F 보안 키의 사용을 지원하지 않습니다.

AWS CLI 및 AWS API

AWS는 현재 AWS Management 콘솔에서만 U2F 보안 키의 사용을 지원합니다. MFA에 대한 U2F 보안 키의 사용은 현재 [AWS CLI 및 AWS API](#) 또는 [MFA 보호 API 작업 \(p. 122\)](#)에 대한 액세스에는 지원되지 않습니다.

추가 리소스

- AWS에서 U2F 보안 키 사용에 대한 자세한 내용은 [U2F 보안 키 활성화\(콘솔\) \(p. 102\)](#) 단원을 참조하십시오.
- AWS에서 U2F 문제 해결에 대한 도움말은 [U2F 보안 키 문제 해결 \(p. 468\)](#) 단원을 참조하십시오.
- U2F 지원에 대한 전반적인 업계 정보는 [Universal 2nd Factor](#) 단원을 참조하십시오.

하드웨어 MFA 디바이스 활성화(콘솔)

동기화된 일회용 암호 알고리즘에 따라 6자리 숫자 코드를 생성하는 하드웨어 MFA 디바이스입니다. 사용자는 로그인 과정 중 디바이스의 유효 코드를 입력해야 합니다. 사용자에게 할당된 각 MFA 디바이스는 고유해야 합니다. 사용자는 다른 사용자의 디바이스 코드를 입력하여 인증받을 수 없습니다.

하드웨어 MFA 디바이스 및 [U2F 보안 키 \(p. 102\)](#)는 모두 본인이 구입한 물리적 디바이스이어야 합니다. 차이점이 있다면 하드웨어 MFA 디바이스가 코드를 생성하여 보여준 후 AWS에 로그인할 때 메시지가 나타나면 해당 란에 입력한다는 점입니다. U2F 보안 키로는 인증 코드를 확인하거나 입력할 수 없습니다. 대신 U2F 보안 키가 응답을 생성하되 사용자에게 보여주지는 않으며 서비스에서 이를 확인합니다. 두 디바이스 유형의 사양 및 구입 관련 정보는 [멀티 팩터 인증](#) 단원을 참조하십시오.

AWS Management 콘솔, 명령줄 또는 IAM API에서 IAM 사용자의 하드웨어 MFA 디바이스를 활성화할 수 있습니다. AWS 계정 루트 사용자에 따른 MFA 디바이스 활성화 방법은 [AWS 계정 루트 사용자용 하드웨어 MFA 디바이스 활성화\(콘솔\) \(p. 110\)](#) 단원을 참조하십시오.

루트 사용자 또는 IAM 사용자별로 (어떤 종류든) 한 개의 MFA 디바이스를 활성화할 수 있습니다.

Note

명령줄에서 디바이스를 활성화하려는 경우 `iam-userenablemfadevice aws iam enable-mfa-device`를 사용합니다. IAM API를 사용하여 MFA 디바이스를 활성화하려면 `EnableMFADevice` 작업을 사용합니다.

주제

- [필요한 권한 \(p. 108\)](#)
- [자신의 IAM 사용자에 대해 하드웨어 MFA 디바이스 활성화\(콘솔\) \(p. 108\)](#)
- [다른 IAM 사용자에 대해 하드웨어 MFA 디바이스 활성\(콘솔\) \(p. 109\)](#)

- AWS 계정 루트 사용자용 하드웨어 MFA 디바이스 활성화(콘솔) (p. 110)
- 물리적 MFA 디바이스 교체 또는 "회전" (p. 111)

필요한 권한

중요한 MFA 관련 작업을 보호하면서 자신의 IAM 사용자에 대한 하드웨어 MFA 디바이스를 관리하려면 다음 정책에 따른 권한이 있어야 합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowManageOwnUserMFA",  
            "Effect": "Allow",  
            "Action": [  
                "iam:DeactivateMFADevice",  
                "iam:EnableMFADevice",  
                "iam:GetUser",  
                "iam>ListMFADevices",  
                "iam:ResyncMFADevice"  
            ],  
            "Resource": "arn:aws:iam::*:user/${aws:username}"  
        },  
        {  
            "Sid": "DenyAllExceptListedIfNoMFA",  
            "Effect": "Deny",  
            "NotAction": [  
                "iam:EnableMFADevice",  
                "iam:GetUser",  
                "iam>ListMFADevices",  
                "iam:ResyncMFADevice"  
            ],  
            "Resource": "arn:aws:iam::*:user/${aws:username}",  
            "Condition": {  
                "BoolIfExists": {  
                    "aws:MultiFactorAuthPresent": "false"  
                }  
            }  
        }  
    ]  
}
```

자신의 IAM 사용자에 대해 하드웨어 MFA 디바이스 활성화(콘솔)

AWS Management 콘솔에서 자신의 하드웨어 MFA 디바이스를 활성화할 수 있습니다.

Note

하드웨어 MFA 디바이스를 활성화하려면 디바이스에 물리적으로 액세스할 수 있어야 합니다.

자신의 IAM 사용자에 대한 하드웨어 MFA 디바이스를 활성화하려면(콘솔)

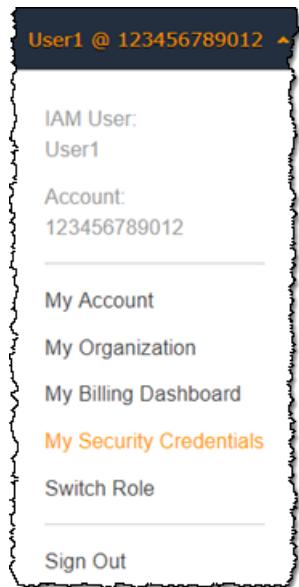
1. AWS 계정 ID 또는 계정 별칭, IAM 사용자 이름, 암호를 사용하여 [IAM 콘솔](#)에 로그인합니다.

Note

사용자 편의를 위해 AWS 로그인 페이지는 브라우저 쿠키를 사용하여 IAM 사용자 이름 및 계정 정보를 기억합니다. 이전에 다른 사용자로 로그인한 경우, 페이지 하단 근처의 [Sign in to a different account]를 선택하여 기본 로그인 페이지로 돌아갑니다. 여기서 AWS 계정 ID 또는 계정 별칭을 입력하면 계정의 IAM 사용자 로그인 페이지로 리디렉션됩니다.

AWS 계정 ID를 받으려면 관리자에게 문의하십시오.

- 오른쪽 상단의 탐색 모음에서 사용자 이름을 선택한 다음 My Security Credentials(내 보안 자격 증명)를 선택합니다.



- AWS IAM credentials(AWS IAM 자격 증명) 탭의 Multi-factor authentication(멀티 팩터 인증) 섹션에서 Manage MFA device(내 MFA 디바이스 관리)를 선택합니다.
- Manage MFA device(MFA 디바이스 관리) 마법사에서 Hardware MFA device(하드웨어 MFA 디바이스)를 선택한 다음 Continue(계속)를 선택합니다.
- 디바이스 일련 번호를 입력합니다. 일련 번호는 보통 디바이스 후면에 있습니다.
- MFA code 1(MFA 코드 1) 상자에 MFA 디바이스에 표시된 6자리 번호를 입력합니다. 디바이스 전면의 버튼을 눌러야 번호가 표시되는 경우도 있습니다.



- 디바이스가 코드를 새로 고칠 때까지 30초 동안 기다린 다음 MFA code 2(MFA 코드 2) 상자에 다음 6자리 번호를 입력합니다. 다시 디바이스 전면의 버튼을 눌러야 두 번째 번호가 표시되는 경우도 있습니다.
- Assign MFA(MFA 할당)을 선택합니다.

Important

인증 코드를 생성한 후 바로 요청을 제출하십시오. 코드를 생성한 후 너무 오래 기다렸다 요청을 제출할 경우 MFA 디바이스가 사용자와 연결은 되지만 MFA 디바이스가 동기화되지 않습니다. 이는 시간 기반 일회용 암호(TOTP)가 잠시 후에 만료되기 때문입니다. 이 경우, [디바이스를 재동기화\(p. 115\)](#)할 수 있습니다.

이제 AWS에서 디바이스를 사용할 준비가 끝났습니다. AWS Management 콘솔의 MFA 사용 방법에 대한 자세한 내용은 [IAM 로그인 페이지에 MFA 디바이스 사용\(p. 57\)](#) 단원을 참조하십시오.

다른 IAM 사용자에 대해 하드웨어 MFA 디바이스 활성화(콘솔)

AWS Management 콘솔에서 다른 IAM 사용자에 대해 하드웨어 MFA 디바이스를 활성화할 수 있습니다.

다른 IAM 사용자에 대해 하드웨어 MFA 디바이스를 활성화하려면(콘솔)

- AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam>에서 IAM 콘솔을 엽니다.

2. 탐색 창에서 Users(사용자)를 선택합니다.
3. MFA를 활성화하려는 사용자의 이름을 선택한 다음 Security credentials(보안 자격 증명) 탭을 선택합니다.
4. Assigned MFA device(할당된 MFA 디바이스) 옆의 관리를 선택합니다.
5. Manage MFA device(MFA 디바이스 관리) 마법사에서 Hardware MFA device(하드웨어 MFA 디바이스)를 선택한 다음 Continue(계속)를 선택합니다.
6. 디바이스 일련 번호를 입력합니다. 일련 번호는 보통 디바이스 후면에 있습니다.
7. MFA code 1(MFA 코드 1) 상자에 MFA 디바이스에 표시된 6자리 번호를 입력합니다. 디바이스 전면의 버튼을 눌러야 번호가 표시되는 경우도 있습니다.



8. 디바이스가 코드를 새로 고칠 때까지 30초 동안 기다린 다음 MFA code 2(MFA 코드 2) 상자에 다음 6자리 번호를 입력합니다. 다시 디바이스 전면의 버튼을 눌러야 두 번째 번호가 표시되는 경우도 있습니다.
9. Assign MFA(MFA 할당)을 선택합니다.

Important

인증 코드를 생성한 후 바로 요청을 제출하십시오. 코드를 생성한 후 너무 오래 기다렸다 요청을 제출할 경우 MFA 디바이스가 사용자와 연결은 되지만 MFA 디바이스가 동기화되지 않습니다. 이는 시간 기반 일회용 암호(TOTP)가 잠시 후에 만료되기 때문입니다. 이 경우, [디바이스를 재동기화 \(p. 115\)](#)할 수 있습니다.

이제 AWS에서 디바이스를 사용할 준비가 끝났습니다. AWS Management 콘솔의 MFA 사용 방법에 대한 자세한 내용은 [IAM 로그인 페이지에 MFA 디바이스 사용 \(p. 57\)](#) 단원을 참조하십시오.

AWS 계정 루트 사용자용 하드웨어 MFA 디바이스 활성화(콘솔)

AWS Management 콘솔에서만 루트 사용자에 대한 가상 MFA 디바이스를 구성하고 활성화할 수 있으며, AWS CLI 또는 AWS API에서는 이 작업을 수행할 수 없습니다.

MFA 디바이스를 분실하거나, 도난당했거나, 디바이스가 작동하지 않을 경우에도 다른 인증 요소를 사용하여 로그인할 수 있습니다. MFA 디바이스로 로그인할 수 없는 경우에 사용자 계정으로 등록된 이메일 및 전화로 사용자 ID를 확인하여 로그인할 수 있습니다. 루트 사용자용 MFA를 활성화하기 전에 계정 설정과 연락처 정보를 검토하여 이메일 및 전화번호에 대한 액세스 권한이 있는지 확인하십시오. 다른 인증 요소를 사용하여 로그인하는 방법은 [MFA 디바이스 분실 또는 작동 중단 시 문제 해결 \(p. 121\)](#) 단원을 참조하십시오. 이 기능을 비활성화하려면 [AWS Support](#)에 문의하십시오.

Note

MFA를 사용하여 로그인 및 인증 디바이스 문제 해결과 같은 다른 텍스트가 나타날 수 있습니다. 그러나 동일한 기능이 제공됩니다. 어느 경우든 대체 인증 팩터를 사용하여 계정 이메일 주소 및 전화 번호를 확인할 수 없는 경우 [AWS Support](#)에 문의하여 MFA 설정을 비활성화하십시오.

루트 사용자용 MFA 디바이스를 활성화하려면(콘솔)

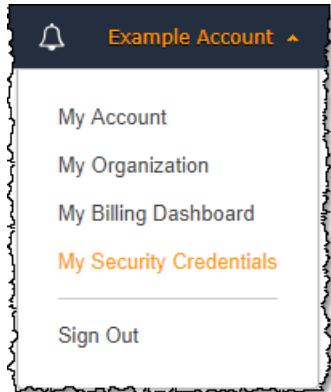
1. AWS 계정 이메일 주소와 암호를 사용하여 [AWS Management 콘솔](#)에 AWS 계정 루트 사용자로 로그인합니다.

Note

이전에 [IAM 사용자](#) 자격 증명으로 콘솔에 로그인한 경우, 브라우저가 이 설정을 기억하여 계정 별 로그인 페이지를 열 수도 있습니다. IAM 사용자 로그인 페이지에서는 AWS 계정 루트 사용자 자격 증명으로 로그인할 수 없습니다. IAM 사용자 로그인 페이지가 나타날 경우에는 페이지

하단에 있는 [Sign in using 루트 사용자 credentials]를 선택하여 기본 로그인 페이지로 돌아갑니다. 기본 로그인 페이지에서 AWS 계정의 이메일 주소와 암호를 입력합니다.

2. 탐색 모음의 오른쪽에서 계정 이름을 선택한 다음 My Security Credentials(내 보안 자격 증명)을 선택합니다. 필요한 경우 보안 자격 증명으로 계속(Continue to Security Credentials)을 선택합니다.



3. Multi-factor authentication (MFA)(멀티 팩터 인증(MFA)) 섹션을 확장합니다.
4. 이전 단계에서 선택한 옵션에 따라 MFA 관리 또는 MFA 활성화(Activate MFA)를 선택합니다.
5. 마법사에서 Hardware MFA device(하드웨어 MFA 디바이스)를 선택한 후 계속을 선택합니다.
6. Serial number(일련 번호) 상자에 MFA 디바이스 뒷면에 있는 일련 번호를 입력합니다.
7. MFA code 1(MFA 코드 1) 상자에 MFA 디바이스에 표시된 6자리 번호를 입력합니다. 디바이스 전면의 버튼을 눌러야 번호가 표시되는 경우도 있습니다.



8. 디바이스가 코드를 새로 고칠 때까지 30초 동안 기다린 다음 MFA code 2(MFA 코드 2) 상자에 다음 6자리 번호를 입력합니다. 다시 디바이스 전면의 버튼을 눌러야 두 번째 번호가 표시되는 경우도 있습니다.
9. Assign MFA(MFA 할당)을 선택합니다. 이제 MFA 디바이스가 AWS 계정과 연결되었습니다.

Important

인증 코드를 생성한 후 바로 요청을 제출하십시오. 코드를 생성한 후 너무 오래 기다렸다 요청을 제출할 경우 MFA 디바이스가 사용자와 연결은 되지만 MFA 디바이스가 동기화되지 않습니다. 이는 시간 기반 일회용 암호(TOTP)가 잠시 후에 만료되기 때문입니다. 이 경우, [디바이스를 재동기화 \(p. 115\)](#)할 수 있습니다.

다음에 루트 사용자 자격 증명을 사용하여 로그인할 때도 MFA 디바이스의 코드를 입력해야 합니다.

물리적 MFA 디바이스 교체 또는 "회전"

한 사용자에게는 한 번에 하나의 MFA 디바이스만 할당할 수 있습니다. 사용자가 디바이스를 분실하거나 이유를 불문하고 교체할 필요가 있을 경우, 먼저 기존 디바이스를 비활성화해야 합니다. 그런 다음, 해당 사용자를 위한 새 디바이스를 추가할 수 있습니다.

- 현재 어떤 사용자와 연결되어 있는 디바이스를 비활성화하는 방법은 [MFA 디바이스 비활성화 \(p. 119\)](#) 단원을 참조하십시오.
- IAM 사용자용 교체 하드웨어 MFA 디바이스를 추가하려면, 이 주제 앞부분의 [다른 IAM 사용자에 대해 하드웨어 MFA 디바이스 활성화\(콘솔\) \(p. 109\)](#) 절차에 나오는 단계를 따르십시오.
- AWS 계정 루트 사용자용 교체 가상 MFA 디바이스를 추가하려면 이 주제 앞부분의 [AWS 계정 루트 사용자용 하드웨어 MFA 디바이스 활성화\(콘솔\) \(p. 110\)](#) 절차에 나오는 단계를 따르십시오.

미리 보기 – SMS 문자 메시지 MFA 디바이스 활성화

AWS는 곧 SMS 멀티 팩터 인증(MFA) 지원을 종료할 예정입니다. 신규 고객은 이 기능을 미리 볼 수 없습니다. 기존 고객은 다음의 MFA 대체 방법 중 하나로 전환하는 것이 좋습니다.

- [가상\(소프트웨어 기반\) \(p. 99\)](#) MFA 디바이스
- [U2F 보안 키 \(p. 102\)](#)
- [하드웨어 기반 \(p. 107\)](#) MFA 디바이스

도움말

계정의 사용자 중에서 SMS MFA 디바이스가 할당된 사용자를 볼 수 있습니다. IAM 콘솔의 탐색 창에서 사용자를 선택하고 표의 MFA 열에서 SMS가 표시된 사용자를 찾습니다.

SMS(문자 서비스) MFA 디바이스는 표준 [SMS 문자 메시지](#)를 받을 수 있는 전화번호를 사용하는 모든 모바일 디바이스일 수 있습니다. MFA 코드가 필요한 경우 AWS가 IAM 사용자에 대해 구성된 전화번호로 해당 코드를 보냅니다.

Note

SMS MFA는 IAM 사용자만 사용할 수 있습니다. AWS 계정 루트 사용자에서는 사용할 수 없습니다. MFA로 루트 사용자를 보호하려면 가상 MFA 디바이스, U2F 보안 키 또는 하드웨어 MFA 디바이스를 사용해야 합니다.

IAM 사용자의 SMS MFA 디바이스 활성화(콘솔)

AWS Management 콘솔에서 IAM을 사용하여 IAM 사용자를 전화번호로 구성함으로써 SMS MFA를 활성화할 수 있습니다.

Note

현재, AWS Management 콘솔에서만 SMS MFA를 관리할 수 있습니다.

IAM 사용자의 SMS MFA를 활성화하려면(콘솔)

1. AWS 계정 ID 또는 계정 별칭, IAM 사용자 이름, 암호를 사용하여 [IAM 콘솔](#)에 로그인합니다.

Note

사용자 편의를 위해 AWS 로그인 페이지는 브라우저 쿠키를 사용하여 IAM 사용자 이름 및 계정 정보를 기억합니다. 이전에 다른 사용자로 로그인한 경우, 페이지 하단 근처의 [Sign in to a different account]를 선택하여 기본 로그인 페이지로 돌아갑니다. 여기서 AWS 계정 ID 또는 계정 별칭을 입력하면 계정의 IAM 사용자 로그인 페이지로 리디렉션됩니다.

2. 탐색 창에서 사용자를 선택합니다.
3. 사용자 이름 목록에서 원하는 MFA 사용자의 이름(확인란 아님)을 선택합니다.
4. Security credentials(보안 자격 증명) 탭을 선택합니다. Assigned MFA device(할당된 MFA 디바이스) 옆의 관리를 선택합니다.
5. Manage MFA Device(MFA 디바이스 관리) 마법사에서 An SMS MFA device(SMS MFA 디바이스)를 선택한 다음 계속을 선택합니다.
6. 이 IAM 사용자에게 MFA 코드를 보낼 전화번호를 입력한 다음 계속을 선택합니다.
7. 확인을 위해 이 지정된 전화번호로 6자리 인증 코드가 즉시 전송됩니다. 6자리 코드를 입력한 후 계속을 선택합니다. 적절한 시간 안에 코드를 받지 못한 경우 코드 재전송(Resend Code)을 선택합니다. SMS 서비스는 전송 시간을 보장하지 않습니다.

8. AWS에서 코드를 확인하면 마법사가 종료됩니다. 종료되지 않으면 마침을 선택하여 마법사를 닫습니다.

IAM 사용자의 SMS MFA 전화번호 변경

IAM 사용자에게 할당된 SMS MFA 디바이스의 전화번호를 변경하려면 현재 MFA 디바이스를 삭제해야 합니다. 그런 다음 새 전화 번호로 새 디바이스를 만들어야 합니다. 디바이스를 삭제하는 방법은 [MFA 디바이스 비활성화 \(p. 119\)](#) 단원을 참조하십시오.

가상 MFA 디바이스 활성화 및 관리(AWS CLI 또는 AWS API)

AWS CLI 명령 또는 AWS API 작업을 사용하여 IAM 사용자를 위한 가상 MFA 디바이스를 활성화할 수 있습니다. AWS CLI, AWS API, Windows PowerShell용 도구 또는 기타 다른 명령줄 도구를 사용하면 AWS 계정 루트 사용자에 대해 MFA 디바이스를 활성화할 수 없습니다. 하지만 AWS Management 콘솔을 사용하여 루트 사용자에 대해 MFA 디바이스를 활성화할 수 있습니다.

AWS Management 콘솔에서 MFA 디바이스를 활성화할 때 콘솔이 사용자를 대신해 여러 단계를 수행합니다. 대신 AWS CLI, Windows PowerShell용 도구 또는 AWS API를 사용해 가상 디바이스를 생성한다면 수동으로 올바른 순서에 따라 단계들을 수행해야 합니다. 예를 들어 가상 MFA 디바이스를 생성하려면 IAM 객체를 생성하고, 코드를 문자열이나 QR 코드 그래픽으로 추출합니다. 그런 다음 디바이스를 동기화하여 IAM 사용자와 연결합니다. 자세한 정보는 [New-IAMVirtualMFADevice](#)의 Examples 단원을 참조하십시오. 물리적 디바이스를 위해서는 생성 단계를 건너뛰고 디바이스를 동기화하고 사용자에게 직접 연결합니다.

IAM에서 가상 디바이스 객체를 생성하여 가상 MFA 디바이스를 나타내려면

이러한 명령은 다음 명령의 많은 일련 번호 대신 사용되는 디바이스에 ARN을 제공합니다.

- AWS CLI: `aws iam create-virtual-mfa-device`
- AWS API: `CreateVirtualMFADevice`

AWS에서 사용할 목적으로 MFA 디바이스를 활성화하려면

다음 명령은 디바이스와 AWS를 동기화하여 사용자 또는 루트 사용자에 연결합니다. 디바이스가 가상이라면 가상 디바이스의 ARN을 일련 번호로 사용합니다.

Important

인증 코드를 생성한 후 바로 요청을 제출하십시오. 코드를 생성한 후 너무 오래 기다렸다 요청을 제출할 경우 MFA 디바이스가 사용자와 연결은 되지만 MFA 디바이스가 동기화되지 않습니다. 이는 시간 기반 일회용 암호(TOTP)가 잠시 후에 만료되기 때문입니다. 이 경우, 아래에서 설명하는 명령을 사용하여 디바이스를 재동기화할 수 있습니다.

- AWS CLI: `aws iam enable-mfa-device`
- AWS API: `EnableMFADevice`

디바이스를 비활성화하려면

다음 명령을 사용하여 디바이스를 사용자에게서 분리하고 비활성화합니다. 디바이스가 가상이라면 가상 디바이스의 ARN을 일련 번호로 사용합니다. 별도로 가상 디바이스 객체를 삭제해야 합니다.

- AWS CLI: `aws iam deactivate-mfa-device`
- AWS API: `DeactivateMFADevice`

가상 MFA 디바이스 객체를 표시하려면

다음 명령을 사용하여 가상 MFA 디바이스 객체의 목록을 봅니다.

- AWS CLI: `aws iam list-virtual-mfa-devices`

- AWS API: [ListVirtualMFADevices](#)

MFA 디바이스를 다시 동기화하려면

디바이스가 AWS에서 허용하지 않는 코드를 생성하는 경우 이러한 명령을 사용하십시오. 디바이스가 가상이라면 가상 디바이스의 ARN을 일련 번호로 사용합니다.

- AWS CLI: [aws iam resync-mfa-device](#)
- AWS API: [ResyncMFADevice](#)

IAM에서 가상 MFA 디바이스 엔터티를 삭제하려면

디바이스가 사용자로부터 분리된 후에 디바이스 개체를 삭제할 수 있습니다.

- AWS CLI: [aws iam delete-virtual-mfa-device](#)
- AWS API: [DeleteVirtualMFADevice](#)

분실되었거나 작동하지 않는 가상 MFA 디바이스를 복구하는 방법

이따금 가상 MFA 앱이 호스팅된 IAM 사용자의 모바일 디바이스가 분실 또는 교체되었거나 작동하지 않는 경우가 있을 수 있습니다. 이러한 경우가 발생하면 사용자는 스스로 디바이스를 복구할 수 없습니다. IAM 사용자는 관리자에게 연락하여 해당 디바이스를 비활성화해야 합니다. 자세한 정보는 [MFA 디바이스 분실 또는 작동 중단 시 문제 해결 \(p. 121\)](#) 단원을 참조하십시오.

MFA 상태 확인

IAM 콘솔을 사용하여 AWS 계정 루트 사용자 또는 IAM 사용자가 유효한 MFA 디바이스를 활성화했는지를 확인할 수 있습니다.

루트 사용자의 MFA 상태를 확인하려면

1. 루트 사용자 자격 증명으로 AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 보안 상태(Security Status) 아래에서 MFA의 활성화 여부를 확인합니다. MFA가 활성화되지 않은 경우, 알림 기호()가 Activate MFA on your 루트 사용자(루트 사용자에서 MFA 활성화) 옆에 표시됩니다.

계정에 대해 MFA를 활성화하고 싶다면 다음 중 하나를 참조하십시오.

- [AWS 계정 루트 사용자용 가상 MFA 디바이스 활성화\(콘솔\) \(p. 100\)](#)
- [AWS 계정 루트 사용자에 대한 U2F 보안 키 활성화\(콘솔\) \(p. 105\)](#)
- [AWS 계정 루트 사용자용 하드웨어 MFA 디바이스 활성화\(콘솔\) \(p. 110\)](#)

IAM 사용자의 MFA 상태를 확인하려면

1. <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 필요할 경우 다음 단계를 통해 사용자 테이블에 MFA 열을 추가합니다.
 - a. 테이블 위 맨 오른쪽에서 설정 아이콘()을 선택합니다.
 - b. 열 관리(Manage Columns)에서 MFA를 선택합니다.
 - c. (선택 사항) 필요할 경우 사용자 테이블에 표시하지 않으려는 열이 있으면 해당 열의 확인란 선택을 취소하면 됩니다.

- d. 담기를 선택하여 사용자 목록으로 돌아갑니다.
4. MFA 열에는 활성화된 MFA 디바이스가 표시됩니다. 사용자에게 활성화되어 있는 MFA 디바이스가 없으면 콘솔에서 활성화되지 않음이라고 표시합니다. 사용자에게 활성화된 MFA 디바이스가 있으면 MFA 열에 활성화된 디바이스의 유형이 가상, U2F Security Key(U2F 보안 키), Hardware(하드웨어) 또는 SMS 값으로 표시됩니다.
 5. 사용자의 MFA 디바이스에 대한 추가 정보를 보려면 MFA 상태를 확인하려는 사용자의 이름을 선택합니다. 그런 다음 보안 자격 증명(Security credentials) 탭을 선택합니다.
 6. 사용자에게 활성화되어 있는 MFA 디바이스가 없으면 콘솔에서 할당된 MFA 디바이스(Assigned MFA device) 옆에 아니요가 표시됩니다. 반대로 활성화되어 있는 MFA 디바이스가 있으면 할당된 MFA 디바이스(Assigned MFA device) 항목에 디바이스 값이 표시됩니다.
 - 하드웨어 디바이스의 디바이스 일련 번호(일반적으로 디바이스 후면의 숫자)(예: GAHT12345678)
 - SMS 디바이스의 AWS의 ARN(예: arn:aws:iam::123456789012:sms-mfa/*username*)
 - 가상 디바이스의 AWS의 ARN(예: arn:aws:iam::123456789012:mfa/*username*)

현재 설정을 변경하려면 Assigned MFA device(할당된 MFA 디바이스) 옆의 관리를 선택합니다.

MFA 활성화에 대한 자세한 내용은 다음을 참조하십시오.

- 가상 멀티 팩터 인증(MFA) 디바이스 활성화(콘솔) (p. 99)
- U2F 보안 키 활성화(콘솔) (p. 102)
- 하드웨어 MFA 디바이스 활성화(콘솔) (p. 107)
- 미리 보기 – SMS 문자 메시지 MFA 디바이스 활성화 (p. 112)

가상 및 하드웨어 MFA 디바이스 재동기화

AWS를 사용하여 가상 및 하드웨어 멀티 팩터 인증(MFA) 디바이스를 다시 동기화할 수 있습니다. 디바이스를 사용하려고 할 때 디바이스가 동기화되지 않으면 로그인 시도가 실패하고 디바이스를 다시 동기화하라는 메시지가 IAM에 표시됩니다.

Note

U2F 보안 키는 항상 동기화됩니다. U2F 보안 키를 분실했거나 도난당한 경우 비활성화할 수 있습니다. 모든 MFA 디바이스 유형의 비활성화에 대한 지침은 [다른 IAM 사용자에 대해 MFA 디바이스를 비활성화하려면\(콘솔\) \(p. 120\)](#) 단원을 참조하십시오.

AWS 관리자로서 IAM 사용자의 가상 및 하드웨어 MFA 디바이스가 동기화 상태를 벗어난 경우 이를 재동기화할 수 있습니다.

AWS 계정 루트 사용자 MFA 디바이스가 작동하지 않는 경우 로그인 프로세스 완료 여부와 관계없이 IAM 콘솔을 사용하여 디바이스를 재동기화할 수 있습니다.

주제

- [필요한 권한 \(p. 115\)](#)
- [가상 및 하드웨어 MFA 디바이스 재동기화\(IAM 콘솔\) \(p. 116\)](#)
- [가상 및 하드웨어 MFA 디바이스 재동기화\(AWS CLI\) \(p. 119\)](#)
- [가상 및 하드웨어 MFA 디바이스 재동기화\(AWS API\) \(p. 119\)](#)

필요한 권한

자신의 IAM 사용자에 대한 가상 또는 하드웨어 MFA 디바이스를 다시 동기화하려면 다음 정책에 따른 권한이 있어야 합니다. 이 정책은 디바이스 생성 또는 비활성화를 허용하지 않습니다

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowListActions",  
            "Effect": "Allow",  
            "Action": [  
                "iam>ListVirtualMFADevices"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Sid": "AllowUserToViewAndManageTheirOwnUserMFA",  
            "Effect": "Allow",  
            "Action": [  
                "iam>ListMFADevices",  
                "iam:ResyncMFADevice"  
            ],  
            "Resource": "arn:aws:iam::*:user/${aws:username}"  
        },  
        {  
            "Sid": "BlockAllExceptListedIfNoMFA",  
            "Effect": "Deny",  
            "NotAction": [  
                "iam>ListMFADevices",  
                "iam>ListVirtualMFADevices",  
                "iam:ResyncMFADevice"  
            ],  
            "Resource": "*",  
            "Condition": {  
                "BoolIfExists": {  
                    "aws:MultiFactorAuthPresent": "false"  
                }  
            }  
        }  
    ]  
}
```

가상 및 하드웨어 MFA 디바이스 재동기화(IAM 콘솔)

IAM 콘솔을 사용하여 가상 및 하드웨어 MFA 디바이스를 재동기화할 수 있습니다.

자신의 IAM 사용자에 대한 가상 또는 하드웨어 MFA 디바이스를 다시 동기화하려면(콘솔)

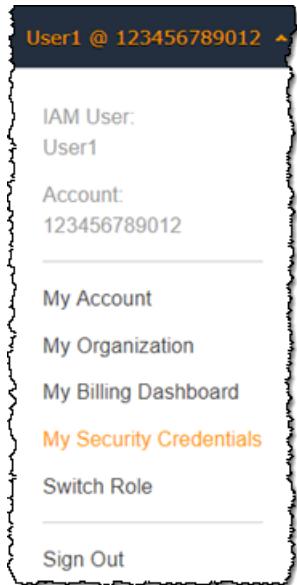
1. AWS 계정 ID 또는 계정 별칭, IAM 사용자 이름, 암호를 사용하여 [IAM 콘솔](#)에 로그인합니다.

Note

사용자 편의를 위해 AWS 로그인 페이지는 브라우저 쿠키를 사용하여 IAM 사용자 이름 및 계정 정보를 기억합니다. 이전에 다른 사용자로 로그인한 경우, 페이지 하단 근처의 [Sign in to a different account]를 선택하여 기본 로그인 페이지로 돌아갑니다. 여기서 AWS 계정 ID 또는 계정 별칭을 입력하면 계정의 IAM 사용자 로그인 페이지로 리디렉션됩니다.

AWS 계정 ID를 받으려면 관리자에게 문의하십시오.

2. 오른쪽 상단의 탐색 모음에서 사용자 이름을 선택한 다음 My Security Credentials(내 보안 자격 증명)를 선택합니다.



3. AWS IAM credentials(AWS IAM 자격 증명) 탭의 Multi-factor authentication(멀티 팩터 인증) 섹션에서 Manage MFA device(내 MFA 디바이스 관리)를 선택합니다.
4. Manage MFA device(MFA 디바이스 관리) 마법사에서 Resync(재동기화)를 선택한 다음 Continue(계속)를 선택합니다.
5. 디바이스에서 순차적으로 생성된 다음 2개의 코드를 MFA code 1(MFA 코드 1) 및 MFA code 2(MFA 코드 2)에 입력합니다. 그런 다음 [Continue]를 선택합니다.

Important

코드를 생성한 후 즉시 요청을 제출하십시오. 코드를 생성한 후 너무 오래 기다렸다 요청을 제출할 경우 요청이 처리되는 것으로 보이지만 디바이스가 동기화되지 않습니다. 이는 시간 기반 일회용 암호(TOTP)가 잠시 후에 만료되기 때문입니다.

다른 IAM 사용자에 대한 가상 및 하드웨어 MFA 디바이스를 다시 동기화하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택한 다음 MFA 디바이스를 재동기화해야 할 사용자의 이름을 선택합니다.
3. Security credentials(보안 자격 증명) 탭을 선택합니다. Assigned MFA device(할당된 MFA 디바이스) 옆의 관리를 선택합니다.
4. Manage MFA device(MFA 디바이스 관리) 마법사에서 Resync(재동기화)를 선택한 다음 Continue(계속)를 선택합니다.
5. 디바이스에서 순차적으로 생성된 다음 2개의 코드를 MFA code 1(MFA 코드 1) 및 MFA code 2(MFA 코드 2)에 입력합니다. 그런 다음 [Continue]를 선택합니다.

Important

코드를 생성한 후 즉시 요청을 제출하십시오. 코드를 생성한 후 너무 오래 기다렸다 요청을 제출할 경우 요청이 처리되는 것으로 보이지만 디바이스가 동기화되지 않습니다. 이는 시간 기반 일회용 암호(TOTP)가 잠시 후에 만료되기 때문입니다.

로그인 전에 루트 사용자 MFA를 재동기화하려면(콘솔)

- Amazon Web Services Sign In With Authentication Device(인증 디바이스로 Amazon Web Services 로그인) 페이지에서 다음을 선택합니다. Having problems with your authentication device?(인증 디바이스에 문제가 있습니까?) Click here.]를 선택합니다.

Note

MFA를 사용하여 로그인 및 인증 디바이스 문제 해결과 같은 다른 텍스트가 나타날 수 있습니다. 그러나 동일한 기능이 제공됩니다.

- Re-Sync With Our Servers(서버를 통한 재동기화) 섹션에서 디바이스에서 순차적으로 생성된 다음 2개의 코드를 MFA code 1(MFA 코드 1) 및 MFA code 2(MFA 코드 2)에 입력합니다. 그런 다음 인증 디바이스 재동기화(Re-sync authentication device)를 선택합니다.
- 필요할 경우 암호를 다시 입력하고 로그인을 선택합니다. 그런 다음 MFA 디바이스를 사용하여 로그인을 완료합니다.

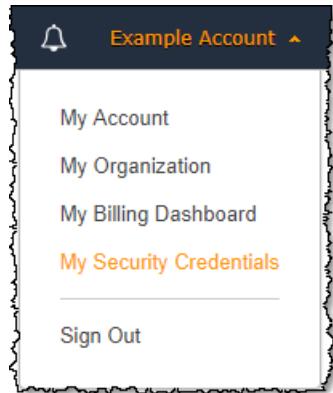
로그인 이후 루트 사용자 MFA 디바이스를 재동기화하려면(콘솔)

- AWS 계정 이메일 주소와 암호를 사용하여 [AWS Management 콘솔](#)에 AWS 계정 루트 사용자로 로그인합니다.

Note

이전에 [IAM 사용자](#) 자격 증명으로 콘솔에 로그인한 경우, 브라우저가 이 설정을 기억하여 계정별 로그인 페이지를 열 수도 있습니다. IAM 사용자 로그인 페이지에서는 AWS 계정 루트 사용자 자격 증명으로 로그인할 수 없습니다. IAM 사용자 로그인 페이지가 나타날 경우에는 페이지 하단에 있는 [Sign in using 루트 사용자 credentials]를 선택하여 기본 로그인 페이지로 돌아갑니다. 기본 로그인 페이지에서 AWS 계정의 이메일 주소와 암호를 입력합니다.

- 탐색 모음의 오른쪽에서 계정 이름을 선택한 다음 My Security Credentials(내 보안 자격 증명)을 선택합니다. 필요한 경우 보안 자격 증명으로 계속(Continue to Security Credentials)을 선택합니다.



- 페이지의 Multi-factor authentication (MFA)(멀티 팩터 인증(MFA)) 섹션을 확장합니다.
- 활성 MFA 디바이스 옆에 있는 Resync(재동기화)를 선택합니다.
- Manage MFA Device(MFA 디바이스 관리) 대화 상자에서 디바이스에서 순차적으로 생성된 다음 2개의 코드를 MFA code 1(MFA 코드 1) 및 MFA code 2(MFA 코드 2)에 입력합니다. 그런 다음 [Continue]를 선택합니다.

Important

코드를 생성한 후 즉시 요청을 제출하십시오. 코드를 생성한 후 너무 오래 기다렸다 요청을 제출할 경우 MFA 디바이스가 사용자와 연결은 되지만 MFA 디바이스가 동기화되지 않습니다. 이는 시간 기반 일회용 암호(TOTP)가 잠시 후에 만료되기 때문입니다.

가상 및 하드웨어 MFA 디바이스 재동기화(AWS CLI)

AWS CLI에서 가상 및 하드웨어 MFA 디바이스를 재동기화할 수 있습니다.

IAM 사용자에 대한 가상 및 하드웨어 MFA 디바이스를 재동기화하려면(AWS CLI)

명령 프롬프트에서 `aws iam resync-mfa-device` 명령을 실행합니다.

- 가상 MFA 디바이스: 디바이스의 Amazon 리소스 이름(ARN)을 `SerialNumber`로 지정합니다.

```
$ aws iam resync-mfa-device --user-name Richard --serial-number  
arn:aws:iam::123456789012:mfa/RichardsMFA --authentication-code-1 123456 --  
authentication-code-2 987654
```

- 하드웨어 MFA 디바이스: 하드웨어 디바이스의 일련 번호를 `SerialNumber`로 지정합니다. 형식은 공급 업체에 따라 다릅니다.

```
PS C:\>Sync-IAMMFADevice -SerialNumber ABCD12345678 -AuthenticationCode1 123456 -  
AuthenticationCode2 987654 -UserName Richard
```

Important

코드를 생성한 후 즉시 요청을 제출하십시오. 코드를 생성한 후 너무 오래 기다렸다 요청을 제출할 경우 잠시 후 코드가 만료되기 때문에 요청이 실패합니다.

가상 및 하드웨어 MFA 디바이스 재동기화(AWS API)

IAM에는 동기화를 실행하는 API 호출이 있습니다. 이러한 경우 가상 및 하드웨어 MFA 사용자에게 API 호출에 액세스할 수 있는 권한을 부여하는 것이 좋습니다. 이때 사용자가 필요할 때마다 디바이스를 재동기화할 수 있도록 API 호출 기반 도구를 구축해야 합니다.

IAM 사용자에 대한 가상 및 하드웨어 MFA 디바이스를 재동기화하려면(AWS API)

- `ResyncMFADevice` 요청을 보냅니다.

MFA 디바이스 비활성화

멀티 팩터 인증(MFA) 디바이스를 사용하여 IAM 사용자로 로그인하는 데 문제가 있는 경우 관리자에게 문의하여 도움을 받으십시오.

관리자는 다른 IAM 사용자에 대해 디바이스를 비활성화할 수 있습니다. 이 방법을 사용하면 MFA를 사용하지 않고 로그인할 수 있습니다. MFA 디바이스가 교체되는 중이거나 디바이스가 일시적으로 사용 불가능할 때 이 방법을 임시 해결 방법으로 사용할 수 있습니다. 그러나 최대한 빨리 사용자를 위한 새 디바이스를 활성화하는 것이 좋습니다. 새 MFA 디바이스를 활성화 하는 방법에 대한 자세한 내용은 [the section called "MFA 디바이스 활성화" \(p. 97\)](#)를 참조하십시오.

Note

API 또는 AWS CLI를 사용하여 AWS 계정에서 사용자를 삭제하는 경우 사용자의 MFA 디바이스를 비활성화 또는 삭제해야 합니다. 이 변경 사항을 사용자 제거 과정의 일부로 활용합니다. 사용자 삭제에 대한 자세한 내용은 [IAM 사용자 관리 \(p. 70\)](#) 단원을 참조하십시오.

주제

- [MFA 디바이스 비활성화\(콘솔\) \(p. 120\)](#)
- [MFA 디바이스 비활성화\(AWS CLI\) \(p. 120\)](#)
- [MFA 디바이스 비활성화\(AWS API\) \(p. 121\)](#)

MFA 디바이스 비활성화(콘솔)

다른 IAM 사용자에 대해 MFA 디바이스를 비활성화하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 Users(사용자)를 선택합니다.
3. 사용자의 MFA 디바이스를 비활성화 하려면 MFA를 제거하려는 사용자의 이름을 선택합니다.
4. Security credentials(보안 자격 증명) 탭을 선택합니다. Assigned MFA device(할당된 MFA 디바이스) 옆의 관리를 선택합니다.
5. Manage MFA device(MFA 디바이스 관리) 마법사에서 Deactivate MFA device(MFA 디바이스 비활성화)를 선택한 다음 Continue(계속)을 선택합니다.

디바이스가 AWS에서 제거됩니다. 디바이스는 다시 활성화되어 AWS 사용자 또는 AWS 계정 루트 사용자에 연결될 때까지는 로그인 또는 요청 인증에 사용할 수 없습니다.

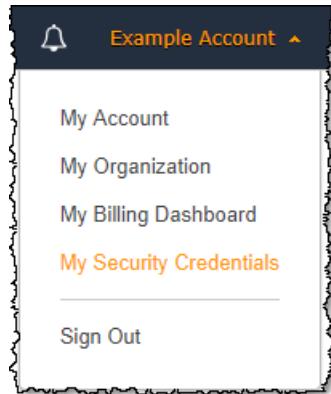
AWS 계정 루트 사용자의 MFA 디바이스를 비활성화하려면(콘솔)

1. AWS 계정 이메일 주소와 암호를 사용하여 [AWS Management 콘솔](#)에 AWS 계정 루트 사용자로 로그인 합니다.

Note

이전에 [IAM 사용자](#) 자격 증명으로 콘솔에 로그인한 경우, 브라우저가 이 설정을 기억하여 계정 별 로그인 페이지를 열 수도 있습니다. IAM 사용자 로그인 페이지에서는 AWS 계정 루트 사용자 자격 증명으로 로그인할 수 없습니다. IAM 사용자 로그인 페이지가 나타날 경우에는 페이지 하단에 있는 [Sign in using 루트 사용자 credentials]를 선택하여 기본 로그인 페이지로 돌아갑니다. 기본 로그인 페이지에서 AWS 계정의 이메일 주소와 암호를 입력합니다.

2. 탐색 모음의 오른쪽에서 계정 이름을 선택한 다음 My Security Credentials(내 보안 자격 증명)을 선택합니다. 필요한 경우 보안 자격 증명으로 계속(Continue to Security Credentials)을 선택합니다.



3. Multi-factor authentication (MFA)(멀티 팩터 인증(MFA)) 섹션을 확장합니다.
4. 비활성화하려는 MFA 디바이스의 행에서 비활성화를 선택합니다.

AWS 계정의 MFA 디바이스가 비활성화됩니다.

MFA 디바이스 비활성화(AWS CLI)

IAM 사용자에 대해 MFA 디바이스를 비활성화하려면(AWS CLI)

- 다음 명령을 실행합니다. `aws iam deactivate-mfa-device`

MFA 디바이스 비활성화(AWS API)

IAM 사용자에 대해 MFA 디바이스를 비활성화하려면(AWS API)

- 다음 연산을 호출합니다. [DeactivateMFADevice](#)

MFA 디바이스 분실 또는 작동 중단 시 문제 해결

AWS 계정 루트 사용자 [멀티 팩터 인증\(MFA\) 디바이스 \(p. 96\)](#) 분실, 손상 또는 고장 시에는 다른 인증 방법을 사용하여 로그인할 수 있습니다. 다시 말해, MFA 디바이스로 로그인할 수 없는 경우에 사용자 계정으로 등록된 이메일 및 전화로 사용자 ID를 확인하여 로그인할 수 있습니다.

가상 MFA 디바이스 (p. 99) 또는 하드웨어 MFA 디바이스 (p. 107)가 정상적으로 작동하는 것처럼 같지만, 이것을 사용하여 AWS 리소스에 액세스하지 못하는 경우에는 AWS와 동기화되지 않는 것이 원인일 수 있습니다. 가상 MFA 디바이스 또는 하드웨어 MFA 디바이스의 동기화에 대한 자세한 내용은 [가상 및 하드웨어 MFA 디바이스 재동기화 \(p. 115\)](#) 단원을 참조하십시오. [U2F 보안 키 \(p. 102\)](#)는 항상 동기화되어 있습니다.

IAM 사용자와 연결된 MFA 디바이스가 손실되었거나 작동하지 않는 경우 사용자가 이를 복구할 수 없습니다. IAM 사용자는 관리자에게 문의하여 디바이스를 비활성화해야 합니다.

다른 인증 요소를 사용하여 루트 사용자 로그인하기 전에 계정과 연결된 이메일과 전화번호에 액세스할 수 있는지 확인하십시오.

AWS 계정 루트 사용자로 다른 인증 요소를 사용하여 로그인하려면

- AWS 계정 이메일 주소와 암호를 사용하여 [AWS Management 콘솔](#)에 AWS 계정 루트 사용자으로 로그인합니다.
- Amazon Web Services Sign In Using MFA(MFA를 사용한 Amazon Web Services 로그인) 페이지에서 Having problems with your authentication device?(인증 디바이스에 문제가 있습니까?)를 선택합니다. Click here.]를 선택합니다.

Note

MFA를 사용하여 로그인 및 인증 디바이스 문제 해결과 같은 다른 텍스트가 나타날 수 있습니다. 그러나 동일한 기능이 제공됩니다. 어느 경우든 대체 인증 팩터를 사용하여 계정 이메일 주소 및 전화 번호를 확인할 수 없는 경우 [AWS Support](#)에 문의하여 MFA 설정을 비활성화하십시오.

- 필요할 경우 암호를 다시 입력하고 로그인을 선택합니다.
- 대체 인증 팩터를 사용하여 로그인(Sign In Using Alternative Factors of Authentication) 섹션에서 대체 팩터를 사용하여 로그인(Sign in using alternative factors)을 선택합니다.
- 이메일 주소를 확인하여 계정을 인증하려면 확인 이메일 전송(Send verification email)을 선택합니다.
- AWS 계정과 연결된 이메일에서 Amazon Web Services의 메시지를 확인합니다(no-reply-aws@amazon.com). 이메일 지침을 따릅니다.

계정에 이메일이 없는 경우에는 스팸 폴더를 확인하거나 브라우저로 돌아가 이메일 재전송(Resend the email)을 선택합니다.

- 이메일 주소를 확인한 후에 계정 인증을 계속 진행할 수 있습니다. 전화 번호를 확인하려면 지금 전화하기(Call me now)를 선택합니다.
- AWS 전화를 받고, 요구에 따라 AWS 웹사이트의 6자리 숫자를 전화 키패드에 입력합니다.

AWS에서 전화가 오지 않을 경우에는 로그인을 선택하여 콘솔에 다시 로그인하고 처음부터 다시 시작합니다. 또는 AWS Support를 선택하여 지원 부서로 문의합니다.

- 전화 번호를 확인한 후에는 콘솔에 로그인(Sign in to the console)을 선택하여 계정에 로그인할 수 있습니다.
- 다음 단계는 사용 중인 MFA의 유형에 따라 다릅니다.

- 가상 MFA 디바이스의 경우, 디바이스에서 계정을 제거합니다. 그런 다음 [AWS 보안 자격 증명\(AWS Security Credentials\)](#) 페이지로 이동하여 기존 MFA 가상 디바이스 개체를 삭제한 다음 새 개체를 생성하십시오.
- U2F 보안 키의 경우, [AWS 보안 자격 증명\(AWS Security Credentials\)](#) 페이지로 이동하여 기존 U2F 키를 비활성화한 다음 새 키를 활성화하십시오.
- 하드웨어 MFA 디바이스의 경우, 타사 공급업체에 연락해 디바이스 수리 또는 교체를 위한 도움을 받습니다. 새 디바이스를 받기 전까지 다른 인증 요소를 사용하여 계속 로그인할 수 있습니다. 하드웨어 MFA 디바이스를 새로 받은 후에는 [AWS 보안 자격 증명\(AWS Security Credentials\)](#) 페이지로 이동하여 기존 MFA 하드웨어 디바이스 개체를 삭제한 다음 새 개체를 생성하십시오.

Note

잃어버렸거나 도난당한 MFA 디바이스를 동일한 유형의 디바이스로 대체해야 하는 것은 아닙니다. 예를 들어, U2F 보안 키가 망가져 새로 주문한 경우, 새로운 U2F 보안 키를 받을 때까지는 가상 MFA 또는 하드웨어 MFA 디바이스를 사용할 수 있습니다.

11. MFA 디바이스가 없거나 도난당한 경우에는 인증 디바이스를 훔친 공격자가 현재 암호를 알 수 있으므로 [AWS 비밀번호도 변경하십시오 \(p. 78\)](#).

IAM 사용자로 MFA 디바이스 도움을 받으려면

1. AWS 관리자나 그 밖에 IAM 사용자의 사용자 이름 및 암호를 제공한 담당자에게 문의합니다. [MFA 디바이스 비활성화 \(p. 119\)](#) 설명대로 관리자가 MFA 디바이스를 비활성화해야 로그인할 수 있습니다.
2. 다음 단계는 사용 중인 MFA의 유형에 따라 다릅니다.
 - 가상 MFA 디바이스의 경우, 디바이스에서 계정을 제거합니다. 그런 다음 [가상 멀티 팩터 인증\(MFA\) 디바이스 활성화\(콘솔\) \(p. 99\)](#) 설명대로 가상 디바이스를 활성화합니다.
 - U2F 보안 키의 경우, 타사 공급업체에 연락해 디바이스 교체를 위한 도움을 받습니다. 새로운 U2F 보안 키를 받은 경우, [U2F 보안 키 활성화\(콘솔\) \(p. 102\)](#)에 설명된 대로 활성화합니다.
 - 하드웨어 MFA 디바이스의 경우, 타사 공급업체에 연락해 디바이스 수리 또는 교체를 위한 도움을 받습니다. 물리적 MFA 디바이스를 새로 받은 후에는 [하드웨어 MFA 디바이스 활성화\(콘솔\) \(p. 107\)](#) 설명대로 디바이스를 활성화합니다.

Note

잃어버렸거나 도난당한 MFA 디바이스를 동일한 유형의 디바이스로 대체해야 하는 것은 아닙니다. 예를 들어, U2F 보안 키가 망가져 새로 주문한 경우, 새로운 U2F 보안 키를 받을 때까지는 가상 MFA 또는 하드웨어 MFA 디바이스를 사용할 수 있습니다.

3. MFA 디바이스가 없거나 도난당한 경우에는 인증 디바이스를 훔친 공격자가 현재 암호를 알 수 있으므로 [비밀번호도 변경하십시오 \(p. 87\)](#).

MFA 보호 API 액세스 구성

사용자가 호출할 수 있는 API 작업을 IAM 정책을 사용해 지정할 수 있습니다. 어떤 경우에는 사용자가 특히 중요한 작업을 수행할 수 있게 허용하기 전에 AWS 멀티 팩터 인증(MFA)으로 인증을 받도록 요구하는 추가 보안이 필요할 수 있습니다.

예를 들어 사용자가 Amazon EC2 RunInstances, DescribeInstances 및 StopInstances 작업을 수행하도록 허용하는 정책이 있을 수 있습니다. 하지만 TerminateInstances처럼 안전하지 않은 작업의 경우 이를 제한해 사용자가 AWS MFA 디바이스에서 인증할 때만 작업을 수행하도록 해야 할 필요가 있을 수 있습니다.

주제

- [개요 \(p. 123\)](#)
- [시나리오: 교차 계정 위임에 대한 MFA 보호 \(p. 125\)](#)
- [시나리오: 현재 계정의 API 작업에 대한 액세스의 MFA 보호 \(p. 126\)](#)
- [시나리오: 리소스 기반 정책이 있는 리소스에 대한 MFA 보호 \(p. 127\)](#)

개요

API 작업에 MFA 보호를 추가하려면 다음과 같은 작업이 필요합니다.

1. 관리자는 MFA 인증이 필요한 API 요청을 해야 하는 각 사용자에 대해 AWS MFA 디바이스를 구성합니다. 이 프로세스는 [MFA 디바이스 활성화 \(p. 97\)](#)에 설명되어 있습니다.
2. 관리자는 사용자가 AWS MFA 디바이스로 인증했는지 여부를 확인하는 Condition 요소가 포함된 사용자 정책을 생성합니다.
3. 나중에 설명할 MFA 보호에 관한 시나리오에 따라 사용자는 MFA 파라미터를 지원하는 AWS STS API 작업인 [AssumeRole](#) 또는 [GetSessionToken](#) 중 하나를 호출합니다. 사용자는 사용자와 연결된 디바이스의 디바이스 식별자를 호출에 포함시킵니다. 또한 사용자는 디바이스에 생성하는 시간 기반 일회용 암호(TOTP)도 포함시킵니다. 각각의 경우, 사용자는 AWS에 추가 요청하는 데 사용하기 위해 임시 보안 자격 증명을 다시 가져옵니다.

Note

서비스의 API 작업에 대한 MFA 보호 기능은 해당 서비스에서 임시 보안 자격 증명을 지원하는 경우에만 사용 가능합니다. 이러한 서비스 목록은 [Using Temporary Security Credentials to Access AWS](#) 단원을 참조하십시오.

권한 부여에 실패한 경우 AWS는 "액세스가 거부되었습니다."라는 오류 메시지를 반환합니다(무단 액세스의 경우와 동일). MFA 보호 API 정책이 적용되는 경우, 사용자가 유효한 MFA 인증 없이 API 작업을 호출하려 하면 AWS에서는 정책에 지정된 API 작업에 대한 액세스를 거부합니다. API 작업 요청의 타임스탬프가 정책에 지정된 허용 범위를 벗어난 경우에도 작업이 거부됩니다. 사용자는 MFA 코드와 디바이스 일련 번호로 새 임시 보안 자격 증명을 요청하여 MFA 인증을 다시 해야 합니다.

MFA 조건이 포함된 IAM 정책

MFA 조건이 포함된 정책은 다음에 연결할 수 있습니다.

- IAM 사용자 또는 그룹
- Amazon S3 버킷, Amazon SQS 대기열 또는 Amazon SNS 주제 등의 리소스
- 사용자가 수입할 수 있는 IAM 역할의 신뢰 정책

정책의 MFA 조건을 사용해 다음과 같은 속성을 확인할 수 있습니다.

- 존재 - 사용자가 MFA로 인증했는지 간단히 확인하려면 `aws:MultiFactorAuthPresent` 키가 Bool 조건에서 `True`인지 확인합니다. 사용자가 단기 자격 증명으로 인증하는 경우에만 키가 있습니다. 액세스 키와 같은 장기 자격 증명에는 이 키가 포함되어 있지 않습니다.
- 기간 - MFA 인증 이후 지정된 시간 내에서만 액세스 권한을 부여하고 싶은 경우, 숫자 조건 유형을 사용하여 `aws:MultiFactorAuthAge` 키의 나이와 값(예: 3,600초)을 비교합니다. MFA가 사용되지 않는 경우 `aws:MultiFactorAuthAge` 키가 없습니다.

다음 예는 MFA 조건을 포함해 MFA 인증이 있는지 테스트하는 IAM 역할의 신뢰 정책을 보여 줍니다. 이 정책을 통해 Principal 요소(ACCOUNT-B-ID를 유효한 AWS 계정 ID로 대체)에 지정된 AWS 계정의 사용자는 이 정책이 연결된 역할을 수입할 수 있습니다. 그러나 이러한 사용자는 MFA를 사용하여 인증을 받은 경우에만 역할을 수입할 수 있습니다.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    "Effect": "Allow",
    "Principal": {"AWS": "ACCOUNT-B-ID"},
    "Action": "sts:AssumeRole",
    "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}
]
```

MFA의 조건 유형에 대한 자세한 내용은 [AWS 전역 조건 컨텍스트 키 \(p. 551\)](#), [숫자 조건 연산자 \(p. 515\)](#) 및 [조건 키의 존재를 확인하는 조건 연산자 \(p. 519\)](#) 단원을 참조하십시오.

GetSessionToken과 AssumeRole 중에서 선택하기

AWS STS에서는 사용자가 MFA 정보를 전달할 수 있도록 GetSessionToken과 AssumeRole이라는 두 가지 API 작업을 제공합니다. 사용자가 임시 보안 자격 증명을 가져오기 위해 호출하는 API 작업은 다음 시나리오 중 어떤 것이 적용되느냐에 따라 달라집니다.

다음 시나리오에는 **GetSessionToken**을 사용합니다.

- 요청을 수행하는 IAM 사용자와 동일한 AWS 계정의 리소스에 액세스하는 API 작업을 호출합니다. GetSessionToken 요청에서 얻는 임시 자격 증명은, 자격 증명 요청에 MFA 정보를 포함하는 경우에 한해, IAM 및 AWS STS API 작업에 액세스할 수 있다는 점에 유의하십시오. GetSessionToken에서 반환하는 임시 자격 증명에 MFA 정보가 포함되어 있으므로 자격 증명에서 수행하는 개별 API 작업에서 MFA를 확인할 수 있습니다.
- MFA 조건이 포함된 리소스 기반 정책으로 보호되는 리소스에 액세스.

GetSessionToken 작업의 목적은 MFA를 사용하는 사용자를 인증하는 것입니다. 정책을 사용하여 인증 작업을 제어할 수는 없습니다.

다음 시나리오에는 **AssumeRole**을 사용합니다.

- 같은 또는 다른 AWS 계정의 리소스에 액세스하는 API 작업을 호출합니다. API 호출은 모든 IAM 또는 AWS STS API를 포함할 수 있습니다. 액세스를 보호하기 위해 사용자가 역할을 수임하는 시각에 MFA를 적용한다는 것에 유의하십시오. AssumeRole에서 반환하는 임시 자격 증명은 컨텍스트에 MFA 정보를 포함하고 있지 않으므로 MFA에 대한 개별 API 작업을 확인할 수 없습니다. 이것이 바로 GetSessionToken을 사용해 리소스 기반 정책에 의해 보호되는 리소스에 대한 액세스를 제한해야 하는 이유입니다.

이러한 시나리오가 구현되는 방식에 대한 세부 정보는 이 문서의 후반부에 나와 있습니다.

MFA 보호 API 액세스에 대한 중요 사항

API 작업에 대한 MFA 보호가 지닌 다음과 같은 측면을 이해하는 것이 중요합니다.

- MFA 보호는 임시 보안 자격 증명을 사용하는 경우에만 제공되며, 임시 보안 자격 증명은 AssumeRole 또는 GetSessionToken을 사용해 얻어야 합니다.
- AWS 계정 루트 사용자 자격 증명으로는 MFA 보호 API 액세스를 사용할 수 없습니다.
- U2F 보안 키로는 MFA 보호 API 액세스를 사용할 수 없습니다.
- 연동 사용자는 AWS 서비스에 사용할 MFA 디바이스를 할당받을 수 없으므로, MFA에서 제어하는 AWS 리소스에 액세스할 수 없습니다. (다음 참조.)
- 임시 자격 증명을 반환하는 다른 AWS STS API 작업에서는 MFA를 지원하지 않습니다. AssumeRoleWithWebIdentity 및 AssumeRoleWithSAML의 경우 사용자는 외부 공급자에 의해 인증되며 AWS에서는 그 공급자가 MFA를 요구했는지 여부를 확인할 수 없습니다. GetFederationToken의 경우 MFA가 특정 사용자와 반드시 연결되는 것은 아닙니다.

- 이와 마찬가지로 장기 자격 증명(IAM 사용자 액세스 키 및 루트 사용자 액세스 키)은 만료되지 않기 때문에 MFA 보호 API 액세스를 통해 사용할 수 없습니다.
- 또한, `AssumeRole` 및 `GetSessionToken`은 MFA 정보 없이도 호출할 수 있습니다. 이 경우 호출자는 임시 보안 자격 증명을 다시 가져오지만, 그러한 임시 자격 증명의 세션 정보에는 사용자가 MFA로 인증했는지 나타나지 않습니다.
- API 작업에 대해 MFA 보호를 설정하려면 정책에 MFA 조건을 추가하면 됩니다. MFA 사용을 적용하기 위해 정책에는 `aws:MultiFactorAuthPresent` 조건 키가 포함되어 있어야 합니다. 교차 계정 위임을 위해 역할의 신뢰 정책에는 조건 키가 포함되어 있어야 합니다.
- 다른 AWS 계정이 내 계정의 리소스에 액세스하도록 허용하는 경우, 리소스의 보안은 신뢰할 수 있는 계정, 즉 다른 계정(내 계정이 아님)의 구성에 따라 달라집니다. 이것은 멀티 팩터 인증이 필요할 때도 마찬가지입니다. 가상 MFA 디바이스를 생성할 권한이 있는 신뢰할 수 있는 계정 내의 어떤 자격 증명도 MFA 클레임을 생성하여 역할의 신뢰 정책의 해당 부분을 충족할 수 있습니다. 멀티 팩터 인증을 요구하는 AWS 리소스에 대한 액세스를 다른 계정의 멤버에게 허용하기 전에 신뢰할 수 있는 계정의 소유자가 보안 모범 사례를 따르도록 해야 합니다. 예를 들어 신뢰할 수 있는 계정에서는 MFA 디바이스 관리 API 작업과 같은 중요 API 작업에 대한 액세스를 신뢰할 수 있는 특정 자격 증명으로 제한해야 합니다.
- 정책에 MFA 조건이 포함된 경우, 사용자가 MFA에 인증되지 않거나 잘못된 MFA 디바이스 식별자 또는 잘못된 TOTP를 제공하는 경우 요청이 거부됩니다.

시나리오: 교차 계정 위임에 대한 MFA 보호

이 시나리오에서는 다른 계정의 IAM 사용자에게 액세스 권한을 위임하려고 합니다. 단 해당 사용자가 AWS MFA 디바이스로 인증된 경우에 한합니다. (교차 계정 위임에 대한 자세한 내용은 [역할 용어 및 개념 \(p. 153\)](#) 단원을 참조하십시오.)

계정 A(액세스할 리소스를 소유한 신뢰하는 계정)에 관리자 권한이 있는 IAM 사용자 Anaya가 있다고 가정해 봅시다. 그녀는 계정 B(신뢰할 수 있는 계정)의 사용자 Richard에게 액세스 권한을 부여하고 싶지만, Richard가 MFA에 인증되었는지 확인한 후에 그가 역할을 수임하기를 원합니다.

1. 신뢰하는 계정 A에서 Anaya는 `CrossAccountRole`이라는 IAM 역할을 생성하고 해당 역할의 신뢰 정책에서 보안 주체를 계정 B의 계정 ID로 설정합니다. 이 신뢰 정책은 AWS STS `AssumeRole` 작업에 권한을 부여합니다. 또한 Anaya는 다음 예와 같이 신뢰 정책에 MFA 조건을 추가합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {"AWS": "ACCOUNT-B-ID"},  
            "Action": "sts:AssumeRole",  
            "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}  
        }  
    ]  
}
```

2. Anaya는 역할이 수행할 수 있는 작업을 지정하는 역할에 권한 정책을 추가합니다. MFA 보호 기능이 포함된 역할의 권한 정책은 다른 역할 권한 정책과 다르지 않습니다. 다음 예는 Anaya가 역할에 추가하는 정책을 보여 줍니다. 역할 위임 사용자는 이 정책을 통해 계정 B의 Books 테이블에서 Amazon DynamoDB 작업을 수행할 수 있고, 아울러 콘솔에서 작업을 수행할 때 필요한 `dynamodb>ListTables` 작업을 할 수 있습니다.

Note

권한 정책은 MFA 조건을 포함하지 않습니다. MFA 인증은 사용자가 역할을 수임할 수 있는지 여부를 결정하는 데에만 사용된다는 점을 알아두십시오. 사용자가 역할을 수임하면 MFA 검사가 추가로 수행되지 않습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": "ACCOUNT-B-ID",  
            "Action": "dynamodb>ListTables",  
            "Resource": "arn:aws:dynamodb:REGION:ACCOUNT-B-ID:table/Books"  
        }  
    ]  
}
```

```
{
    "Sid": "TableActions",
    "Effect": "Allow",
    "Action": "dynamodb:/*",
    "Resource": "arn:aws:dynamodb:*:ACCOUNT-B-ID:table/Books"
},
{
    "Sid": "ListTable",
    "Effect": "Allow",
    "Action": "dynamodb:ListTable",
    "Resource": "*"
}
]
```

3. 신뢰할 수 있는 계정 B에서 관리자는 IAM 사용자 Richard가 AWS MFA 디바이스로 구성되었는지, 그리고 그가 이 디바이스의 ID를 알고 있는지 확인합니다. 디바이스 ID란 하드웨어 MFA 디바이스의 경우 일련 번호이며, 가상 MFA 디바이스의 경우 해당 디바이스의 ARN입니다.
4. 계정 B에서 관리자는 사용자 Richard(또는 그가 소속된 그룹)에게 AssumeRole 작업을 호출할 수 있도록 허용하는 다음과 같은 정책을 연결합니다. 리소스는 Anaya가 1단계에서 생성한 역할의 ARN으로 설정됩니다. 이 정책에는 MFA 조건이 포함되어 있지 않습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["sts:AssumeRole"],
            "Resource": ["arn:aws:iam:ACCOUNT-A-ID:role/CrossAccountRole"]
        }
    ]
}
```

5. 계정 B에서 Richard(또는 Richard가 실행하는 애플리케이션)은 AssumeRole을 호출합니다. API 호출에는 위임할 역할의 ARN(arn:aws:iam:ACCOUNT-A-ID:role/CrossAccountRole), MFA 디바이스의 ID 및 Richard가 자신의 디바이스에서 가져오는 현재 TOTP가 포함되어 있습니다.

Richard가 AssumeRole을 호출하면, AWS에서 그가 MFA에 대한 요건을 포함해 유효한 자격 증명을 갖고 있는지 여부를 확인합니다. 만일 Richard가 유효한 자격 증명을 갖고 있다면 성공적으로 역할을 수임해 역할의 임시 자격 증명을 사용함과 동시에 계정 A에서 Books라는 테이블에 대해 어떤 DynamoDB 작업도 수행할 수 있습니다.

AssumeRole을 호출하는 프로그램의 예는 [MFA 인증이 포함된 AssumeRole 호출하기\(Python\)](#) (p. 132) 단원을 참조하십시오.

시나리오: 현재 계정의 API 작업에 대한 액세스의 MFA 보호

이 시나리오에서는 AWS 계정의 사용자가 AWS MFA 디바이스를 사용해 인증받은 경우에만 중요한 API 작업에 액세스할 수 있는지 확인해야 합니다.

계정 A에 EC2 인스턴스로 작업해야 하는 개발자 그룹이 있다고 가정해 봅시다. 일반적인 개발자들은 이 인스턴스를 사용할 수 있지만, ec2:StopInstances 또는 ec2:TerminateInstances 작업에 대한 권한은 없습니다. 그와 같은 "안전하지 않은" 권한이 있는 작업을 몇몇 신뢰할 수 있는 사용자만 액세스할 수 있게 제한하고자 하여, 이러한 민감한 Amazon EC2 작업을 허용하는 정책에 MFA 보호를 추가합니다.

이 시나리오에서 신뢰할 수 있는 사용자 중 한 명은 사용자 Sofia입니다. 사용자 Anaya는 계정 A의 관리자입니다.

1. Anaya는 Sofia가 AWS MFA 디바이스로 구성되었는지, 그리고 Sofia가 이 디바이스의 ID를 알고 있는지 확인합니다. 디바이스 ID란 하드웨어 MFA 디바이스의 경우 일련 번호이며, 가상 MFA 디바이스의 경우 해당 디바이스의 ARN입니다.
2. Anaya는 EC2-Admins라는 그룹을 생성하고 이 그룹에 사용자 Sofia를 추가합니다.

3. Anaya는 EC2-Admins 그룹에 다음과 같은 정책을 연결합니다. 이 정책은 사용자에게 Amazon EC2 StopInstances 및 TerminateInstances 작업을 호출할 권한을 부여하는데, 단 이 사용자가 MFA를 사용하여 인증되었을 경우에 한합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": [  
            "ec2:StopInstances",  
            "ec2:TerminateInstances"  
        ],  
        "Resource": ["*"],  
        "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}  
    }]  
}
```

4. Note

이 정책의 효력이 발생하려면 사용자는 먼저 로그아웃한 한 후 다시 로그인해야 합니다.

사용자 Sofia가 Amazon EC2 인스턴스를 중지하거나 종료해야 하는 경우, Sofia(또는 Sofia가 실행하는 애플리케이션)는 GetSessionToken을 호출합니다. 이 API 작업에서는 MFA 디바이스의 ID와 Sofia가 자신의 디바이스에서 가져오는 현재 TOTP를 전달합니다.

5. 사용자 Sofia(또는 Sofia가 사용하는 애플리케이션)은 GetSessionToken에서 제공하는 임시 자격 증명을 사용하여 Amazon EC2 StopInstances 또는 TerminateInstances 작업을 호출합니다.

GetSessionToken을 호출하는 프로그램의 예는 이 문서의 후반부에 있는 [MFA 인증이 포함된 GetSessionToken 호출하기\(Python 및 C#\) \(p. 131\)](#) 단원을 참조하십시오.

시나리오: 리소스 기반 정책이 있는 리소스에 대한 MFA 보호

이 시나리오에서는 S3 버킷, SQS 대기열 또는 SNS 주제의 소유자입니다. 리소스에 액세스하는 모든 AWS 계정 사용자가 AWS MFA 디바이스로 인증되었는지 확인하려고 합니다.

이 시나리오는 사용자가 역할을 먼저 수입하지 않고도 교차 계정 MFA 보호를 제공하는 방법을 설명합니다. 이 경우 사용자는 세 가지 조건이 충족되면 리소스에 액세스할 수 있습니다. 즉 사용자는 MFA로 인증을 받아야 하고, GetSessionToken에서 임시 보안 자격 증명을 가져올 수 있어야 하며, 리소스의 정책에서 신뢰하는 계정에 로그인해 있어야 합니다.

계정 A에 속해 있고 S3 버킷을 생성한다고 가정해 봅시다. 여러 AWS 계정에 속한 사용자에게 이 버킷에 대한 액세스를 부여하되, 사용자가 MFA로 인증한 경우에 한하고자 합니다.

이 시나리오에서 사용자 Anaya는 계정 A의 관리자입니다. 사용자 Nikhil은 계정 C의 IAM 사용자입니다.

- 계정 A에서 Anaya는 Account-A-bucket이라는 버킷을 생성합니다.
- Anaya는 이 버킷에 버킷 정책을 추가합니다. 이 정책은 계정 A, 계정 B 또는 계정 C의 모든 사용자가 이 버킷에서 Amazon S3 PutObject 및 DeleteObject 작업을 수행하도록 허용합니다. 이 정책에는 MFA 조건이 포함되어 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Principal": {"AWS": [  
            "ACCOUNT-A-ID",  
            "ACCOUNT-B-ID",  
            "ACCOUNT-C-ID"  
        ]},  
        "Action": [  
            "s3:PutObject",  
            "s3:DeleteObject"  
        ]  
    }]  
}
```

```
    "s3:PutObject",
    "s3:DeleteObject"
],
"Resource": ["arn:aws:s3:::ACCOUNT-A-BUCKET-NAME/*"],
"Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}
}
}
```

Note

Amazon S3는 루트 계정 액세스에 대해(서만) MFA Delete 기능을 제공합니다. 버킷의 버전 관리 상태를 설정할 때 Amazon S3 MFA Delete를 활성화할 수 있습니다. Amazon S3 MFA Delete는 IAM 사용자에게 적용되지 않으며, MFA 보호 API 액세스에서 독립적으로 관리됩니다. 버킷을 삭제할 권한이 있는 IAM 사용자도 Amazon S3 MFA 삭제 기능이 활성화된 버킷은 삭제할 수 없습니다. Amazon S3 MFA Delete에 대한 자세한 내용은 [MFA Delete](#) 단원을 참조하십시오.

3. 계정 C에서 관리자는 사용자 Nikhil이 AWS MFA 디바이스로 구성되었는지, 그리고 그가 이 디바이스의 ID를 알고 있는지 확인합니다. 디바이스 ID란 하드웨어 MFA 디바이스의 경우 일련 번호이며, 가상 MFA 디바이스의 경우 해당 디바이스의 ARN입니다.
4. 계정 C에서 Nikhil(또는 그가 실행하는 애플리케이션)은 `GetSessionToken`을 호출합니다. 이 호출에는 MFA 디바이스의 ID 또는 ARN과 Nikhil이 자신의 디바이스에서 가져오는 현재 TOTP가 포함되어 있습니다.
5. Nikhil(또는 그가 사용하는 애플리케이션)은 `GetSessionToken`에서 반환하는 임시 자격 증명을 사용하여 `PutObject`으로 파일을 업로드하는 Amazon S3 Account-A-bucket 작업을 호출합니다.

`GetSessionToken`을 호출하는 프로그램의 예는 이 문서의 후반부에 있는 [MFA 인증이 포함된 GetSessionToken 호출하기\(Python 및 C#\) \(p. 131\)](#) 단원을 참조하십시오.

Note

`AssumeRole`이 반환하는 임시 자격 증명은 이 경우에는 유효하지 않습니다. 사용자는 역할 수임을 위해 MFA 정보를 제공할 수 있지만 `AssumeRole`에서 반환하는 임시 자격 증명에는 MFA 정보가 포함되어 있지 않습니다. 이 정보는 정책의 MFA 조건을 충족하기 위해 필요합니다.

MFA 조건이 포함된 샘플 정책

다음 예에서는 MFA 조건을 정책에 추가할 수 있는 방법을 더 보여줍니다. 이러한 예제 JSON 정책 문서를 사용하여 IAM 정책을 생성하는 방법에 대해 자세히 알아보려면 [the section called “JSON 탭에서 정책 만들기” \(p. 381\)](#) 단원을 참조하십시오.

Note

다음 예는 AWS 계정의 IAM 사용자 또는 그룹에 직접 연결된 정책을 보여줍니다. 예를 여러 계정 걸쳐 있는 MFA 보호 API 작업에 맞게 수정하려면, 대신 IAM 역할을 사용하고 역할 액세스 정책이 아닌 역할 신뢰 정책에 MFA 조건 검사를 추가합니다. 자세한 내용은 [시나리오: 교차 계정 위임에 대한 MFA 보호 \(p. 125\)](#) 단원을 참조하십시오.

주제

- 예 1: 최근 MFA 인증(`GetSessionToken`) 이후 액세스 부여 (p. 128)
- 예 2: 유효한 MFA 인증(`GetSessionToken`)이 없는 경우 특정 API 작업에 대한 액세스 거부 (p. 129)
- 예 3: 유효한 최신 MFA 인증(`GetSessionToken`)이 없는 경우 특정 API 작업에 대한 액세스 거부 (p. 129)

예 1: 최근 MFA 인증(`GetSessionToken`) 이후 액세스 부여

다음 예는 사용자가 지난 1시간(3600초) 내에 MFA로 인증된 경우에만 Amazon EC2 액세스를 부여하는 사용자 또는 그룹에 연결된 정책을 보여 줍니다. 장기 자격 증명 및 이 정책이 있는 사용자가 Amazon EC2

API를 호출하면, MultiFactorAuthAge 키가 장기 자격 증명에 대한 요청 컨텍스트에 없기 때문에 호출에 실패합니다. 연산자를 NumericLessThanIfExists로 변경하여 장기 자격 증명을 허용하거나 먼저 sts:GetSessionToken API를 사용하여 사용자가 MFA로 유효성이 검사된 단기 자격 증명을 가져오도록 요청할 수 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": ["ec2:*"],  
        "Resource": ["*"],  
        "Condition": {"NumericLessThan": {"aws:MultiFactorAuthAge": "3600"}}  
    }]  
}
```

예 2: 유효한 MFA 인증(GetSessionToken)이 없는 경우 특정 API 작업에 대한 액세스 거부

다음 예는 전체 Amazon EC2 API에 대한 액세스를 부여하지만 사용자가 MFA로 인증되지 않은 경우 StopInstances 및 TerminateInstances에 대한 액세스를 거부하는 사용자 또는 그룹에 연결된 정책을 보여줍니다. 이 정책은 원하는 결과를 얻기 위해 두 개의 설명문을 필요로 합니다. 첫 번째 문("Sid": "AllowAllActionsForEC2" 포함)은 모든 Amazon EC2 작업을 허용합니다. 두 번째 설명문("Sid": "DenyStopAndTerminateWhenMFAIsNotPresent" 포함)은 MFA 인증 컨텍스트가 누락된 경우(MFA가 사용되지 않았음을 의미) StopInstances 및 TerminateInstances 작업을 거부합니다.

Note

MFA를 사용하지 않을 때는 키가 없어 키를 평가할 수 없기 때문에 Deny 문의 MultiFactorAuthPresent에 대한 조건 확인이 {"Bool": {"aws:MultiFactorAuthPresent": false}}이면 안 됩니다. 따라서 값을 확인하기 전에 BoolIfExists를 사용하여 키가 있는지 확인해야 합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowAllActionsForEC2",  
            "Effect": "Allow",  
            "Action": "ec2:*",  
            "Resource": "*"  
        },  
        {  
            "Sid": "DenyStopAndTerminateWhenMFAIsNotPresent",  
            "Effect": "Deny",  
            "Action": [  
                "ec2:StopInstances",  
                "ec2:TerminateInstances"  
            ],  
            "Resource": "*",  
            "Condition": {"BoolIfExists": {"aws:MultiFactorAuthPresent": false}}  
        }  
    ]  
}
```

예 3: 유효한 최신 MFA 인증(GetSessionToken)이 없는 경우 특정 API 작업에 대한 액세스 거부

다음 예는 전체 Amazon EC2 API에 대한 액세스를 부여하지만 사용자가 지난 1시간 내에 MFA를 사용하여 인증하지 않은 경우 StopInstances 및 TerminateInstances에 대한 액세스를 거부하는 사용자 또

는 그룹에 연결된 정책을 보여 줍니다. 이 예는 이전 예를 확장한 것이며, 원하는 결과를 얻기 위해 세 개의 문이 필요합니다. 첫 두 개의 문은 이전의 예제와 동일합니다. MFA가 전혀 사용되지 않는 경우(MFA 컨텍스트가 없음) 두 번째 문에는 여전히 StopInstances와 TerminateInstances를 거부하는 조건이 포함됩니다. 다음 예에서 세 번째 문("Sid": "DenyStopAndTerminateWhenMFAIsOlderThanOneHour" 포함)은 MFA 인증이 있지만 요청이 있기 한 시간 이상 전에 발생한 경우 StopInstances 및 TerminateInstances 작업을 거부하는 추가적인 조건을 포함합니다. 예를 들어 IAM 사용자는 MFA를 사용하여 AWS Management 콘솔에 로그인한 후 2시간이 지나야 EC2 인스턴스에 대한 중지 또는 종료 시도를 할 수 있습니다. 다음 정책은 이를 방지합니다. 이 시나리오에서 EC2 인스턴스를 중지 또는 종료하려면 사용자는 로그아웃한 후 MFA를 사용하여 다시 로그인하고 나서, 로그인 시작으로부터 한 시간 이내에 인스턴스를 중지 또는 종료해야 합니다.

Note

MFA를 사용하지 않을 때는 키가 없어 키를 평가할 수 없기 때문에 첫 번째 Deny 문의 MultiFactorAuthPresent에 대한 조건 확인은 "BoolIfExists"를 사용합니다. MFA를 사용하지 않아 값이 없는 경우, true를 반환하고 문이 일치하며 액세스를 거부합니다.
aws:MultiFactorAuthAge 조건은 MFA 컨텍스트가 요청에 있을 때만 존재합니다. 따라서 문 2는 MFA가 전혀 존재하지 않는 경우에 해당하고, 문 3은 MFA가 존재하는 경우에 해당하며 알맞은 시간 범위 내에 MFA가 발생했는지 평가합니다. 다시 말해 키가 없는 경우 ...IfExists는 테스트에서 true를 반환하도록 하고 문이 일치하며 그러한 API 작업에 대한 사용자의 액세스가 거부됩니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowAllActionsForEC2",  
            "Effect": "Allow",  
            "Action": "ec2:*",  
            "Resource": "*"  
        },  
        {  
            "Sid": "DenyStopAndTerminateWhenMFAIsNotPresent",  
            "Effect": "Deny",  
            "Action": [  
                "ec2:StopInstances",  
                "ec2:TerminateInstances"  
            ],  
            "Resource": "*",  
            "Condition": {"BoolIfExists": {"aws:MultiFactorAuthPresent": false}}  
        },  
        {  
            "Sid": "DenyStopAndTerminateWhenMFAIsOlderThanOneHour",  
            "Effect": "Deny",  
            "Action": [  
                "ec2:StopInstances",  
                "ec2:TerminateInstances"  
            ],  
            "Resource": "*",  
            "Condition": {"NumericGreaterThanIfExists": {"aws:MultiFactorAuthAge": "3600"}}  
        }  
    ]  
}
```

샘플 코드: 멀티 팩터 인증이 포함된 자격 증명 요청하기

다음 예에서는 GetSessionToken 및 AssumeRole 작업을 호출하고 MFA 인증 파라미터를 전달하는 방법을 보여줍니다. 권한이 없어도 GetSessionToken을 호출할 수 있지만, AssumeRole을 호출할 수 있게 허용하는 정책이 있어야 합니다. 반환된 자격 증명은 계정 내 모든 S3 버킷의 목록을 나열하는 데 사용됩니다.

MFA 인증이 포함된 GetSessionToken 호출하기(Python 및 C#)

AWS SDK for Python (Boto) 및 .NET용 AWS SDK를 토대로 작성된 다음에는 GetSessionToken을 호출하고 MFA 인증 정보를 전달하는 방법을 보여 줍니다. GetSessionToken 작업에서 반환하는 임시 자격 증명은 이어서 계정 내 모든 S3 버킷의 목록을 나열하는데 사용됩니다.

이 코드를 실행하는 사용자(또는 사용자가 속한 그룹)에게 연결된 정책에서는 반환된 임시 자격 증명에 대한 권한을 제공합니다. 이 예의 경우 정책에서 사용자에게 Amazon S3 ListBuckets 작업을 요청할 수 있는 권한을 부여해야 합니다.

Python 사용하기

```
import boto
from boto.s3.connection import S3Connection
from boto.sts import STSConnection

# Prompt for MFA time-based one-time password (TOTP)
mfa_TOTP = raw_input("Enter the MFA code: ")

# The calls to AWS STS GetSessionToken must be signed with the access key ID and secret
# access key of an IAM user. The credentials can be in environment variables or in
# a configuration file and will be discovered automatically
# by the STSConnection() function. For more information, see the Python SDK
# documentation: http://boto.readthedocs.org/en/latest/boto_config_tut.html

sts_connection = STSConnection()

# Use the appropriate device ID (serial number for hardware device or ARN for virtual
# device).
# Replace ACCOUNT-NUMBER-WITHOUT-HYPHENS and MFA-DEVICE-ID with appropriate values.

tempCredentials = sts_connection.get_session_token(
    duration=3600,
    mfa_serial_number="&region-arn;iam::ACCOUNT-NUMBER-WITHOUT-HYPHENS:mfa/MFA-DEVICE-ID",
    mfa_token=mfa_TOTP
)

# Use the temporary credentials to list the contents of an S3 bucket
s3_connection = S3Connection(
    aws_access_key_id=tempCredentials.access_key,
    aws_secret_access_key=tempCredentials.secret_key,
    security_token=tempCredentials.session_token
)

# Replace BUCKET-NAME with an appropriate value.
bucket = s3_connection.get_bucket(bucket_name="BUCKET-NAME")
objectlist = bucket.list()
for obj in objectlist:
    print obj.name
```

C# 사용하기

```
Console.Write("Enter MFA code: ");
string mfaTOTP = Console.ReadLine(); // Get string from user

/* The calls to AWS STS GetSessionToken must be signed using the access key ID and secret
access key of an IAM user. The credentials can be in environment variables or in
a configuration file and will be discovered automatically
by the AmazonSecurityTokenServiceClient constructor. For more information, see
http://docs.aws.amazon.com/AWSSdkDocsNET/latest/DeveloperGuide/net-dg-config-creds.html
*/
AmazonSecurityTokenServiceClient stsClient =
```

```
new AmazonSecurityTokenServiceClient();
GetSessionTokenRequest getSessionTokenRequest = new GetSessionTokenRequest();
getSessionTokenRequest.DurationSeconds = 3600;

// Replace ACCOUNT-NUMBER-WITHOUT-HYPHENS and MFA-DEVICE-ID with appropriate values
getSessionTokenRequest.SerialNumber = "arn:aws:iam::ACCOUNT-NUMBER-WITHOUT-HYPHENS:mfa/MFA-
DEVICE-ID";
getSessionTokenRequest.TokenCode = mfaTOTP;

GetSessionTokenResponse getSessionTokenResponse =
    stsClient.GetSessionToken(getSessionTokenRequest);

// Extract temporary credentials from result of GetSessionToken call
GetSessionTokenResult getSessionTokenResult =
    getSessionTokenResponse.GetSessionTokenResult;
string tempAccessKeyId = getSessionTokenResult.Credentials.AccessKeyId;
string tempSessionToken = getSessionTokenResult.Credentials.SessionToken;
string tempSecretAccessKey = getSessionTokenResult.Credentials.SecretAccessKey;
SessionAWSCredentials tempCredentials = new SessionAWSCredentials(tempAccessKeyId,
    tempSecretAccessKey, tempSessionToken);

// Use the temporary credentials to list the contents of an S3 bucket
// Replace BUCKET-NAME with an appropriate value
ListObjectsRequest S3ListObjectsRequest = new ListObjectsRequest();
S3ListObjectsRequest.BucketName = "BUCKET-NAME";
S3Client = AWSClientFactory.CreateAmazonS3Client(tempCredentials);
ListObjectsResponse S3ListObjectsResponse =
    S3Client.ListObjects(S3ListObjectsRequest);
foreach (S3Object s3Object in S3ListObjectsResponse.S3Objects)
{
    Console.WriteLine(s3Object.Key);
}
```

MFA 인증이 포함된 AssumeRole 호출하기(Python)

AWS SDK for Python (Boto)을 토대로 작성된 다음 예는 AssumeRole을 호출하고 MFA 인증 정보를 전달하는 방법을 보여 줍니다. AssumeRole에서 반환한 임시 보안 자격 증명은 계정의 모든 Amazon S3 버킷을 나열하는 데 사용됩니다.

이 시나리오에 대한 자세한 내용은 [시나리오: 교차 계정 위임에 대한 MFA 보호 \(p. 125\)](#)를 참조하십시오.

```
import boto
from boto.s3.connection import S3Connection
from boto.sts import STSConnection

# Prompt for MFA time-based one-time password (TOTP)
mfa_TOTP = raw_input("Enter the MFA code: ")

# The calls to AWS STS AssumeRole must be signed with the access key ID and secret
# access key of an IAM user. (The AssumeRole API operation can also be called using
# temporary
# credentials, but this example does not show that scenario.)
# The IAM user credentials can be in environment variables or in
# a configuration file and will be discovered automatically
# by the STSConnection() function. For more information, see the Python SDK
# documentation: http://boto.readthedocs.org/en/latest/boto_config_tut.html

sts_connection = STSConnection()

# Use appropriate device ID (serial number for hardware device or ARN for virtual device)
# Replace ACCOUNT-NUMBER-WITHOUT-HYPHENS, ROLE-NAME, and MFA-DEVICE-ID with appropriate
# values
tempCredentials = sts_connection.assume_role(
    role_arn="arn:aws:iam::ACCOUNT-NUMBER-WITHOUT-HYPHENS:role/ROLE-NAME",
```

```
role_session_name="AssumeRoleSession1",
mfa_serial_number="arn:aws:iam::ACCOUNT-NUMBER-WITHOUT-HYPHENS:mfa/MFA-DEVICE-ID",
mfa_token=mfa_TOTP
)

# Use the temporary credentials to list the contents of an S3 bucket
s3_connection = S3Connection(
    aws_access_key_id=tempCredentials.credentials.access_key,
    aws_secret_access_key=tempCredentials.credentials.secret_key,
    security_token=tempCredentials.credentials.session_token
)

# Replace BUCKET-NAME with a real bucket name
bucket = s3_connection.get_bucket(bucket_name="BUCKET-NAME")
objectlist = bucket.list()
for obj in objectlist:
    print obj.name
```

미사용 자격 증명 찾기

AWS 계정의 보안을 강화하려면 필요 없는 IAM 사용자 자격 증명(암호화 액세스 키)을 삭제합니다. 예를 들어, 사용자가 조직을 떠나거나 AWS 액세스가 더 이상 필요하지 않은 경우 해당 자격 증명을 찾아서 더 이상 작동하지 않도록 해야 합니다. 더 이상 필요 없는 자격 증명을 삭제하는 것이 가장 좋습니다. 나중에 필요한 경우가 생기면 언제든지 다시 생성할 수 있습니다. 적어도 암호를 변경하거나 액세스 키를 비활성화하여 이전 사용자가 더 이상 액세스할 수 없게 해야 합니다.

미사용은 이와는 다른 것으로 보통 특정 기간 동안 사용되지 않은 자격 증명을 뜻합니다.

미사용 암호 찾기

AWS Management 콘솔을 사용하여 사용자의 암호 사용 정보를 볼 수 있습니다. 사용자 수가 많을 경우 콘솔을 사용하여 각 사용자가 자신의 콘솔 암호를 사용한 최종 시각에 대한 정보가 담긴 자격 증명 보고서를 다운로드할 수 있습니다. AWS CLI 또는 IAM API에서 그 정보에 액세스할 수도 있습니다.

미사용 암호를 확인하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 필요할 경우 사용자 테이블에 Console last sign-in(콘솔 마지막 로그인) 열을 추가합니다.
 - a. 테이블 위 맨 오른쪽에서 설정 아이콘()을 선택합니다.
 - b. Manage Columns(열 관리)에서 Console last sign-in(콘솔 마지막 로그인)을 선택합니다.
 - c. 닫기를 선택하여 사용자 목록으로 돌아갑니다.
4. Console last sign-in(콘솔 마지막 로그인) 열에는 사용자가 콘솔을 통해 마지막으로 AWS에 로그인한 날짜부터 경과한 일수가 표시됩니다. 이 정보를 통해 지정된 기간 이상 동안 암호를 사용하여 로그인하지 않은 사용자를 확인할 수 있습니다. 암호 사용자 중 로그인한 적이 없는 사용자는 이 열에 없음이라고 표시됩니다. 없음은 암호가 없는 사용자를 나타냅니다. 최근에 사용된 적이 없는 암호는 삭제해야 할 자격 증명을 식별하기 위한 좋은 기준이 될 수 있습니다.

Important

서비스 문제로 인해 암호가 마지막으로 사용된 데이터에 2018년 5월 3일 22:50 PDT ~ 2018년 5월 23일 14:08 PDT 사이의 암호 사용이 포함되어 있지 않습니다. 이는 IAM 콘솔에 표시되는 **마지막 로그인 날짜**, **IAM 자격 증명 보고서**의 암호가 마지막으로 사용된 날짜와 **GetUser API 연산**에 의해 반환되는 암호가 마지막으로 사용된 날짜에 영향을 줍니다. 사용자가 해당 기간에 로그인한 경우 반환되는 암호가 마지막으로 사용된 날짜는 사용자가 2018년 5월 3일 이전에 마

지금으로 로그인한 날짜입니다. 사용자가 2018년 5월 23일 14:08 PDT 이후에 로그인한 경우 반환되는 암호가 마지막으로 사용된 날짜는 정확합니다.

암호가 마지막으로 사용된 정보를 사용하여 삭제할 사용되지 않은 자격 증명을 식별할 경우(예: 지난 90일 동안 AWS에 로그인하지 않은 사용자 삭제) 2018년 5월 23일 이후의 날짜를 포함하도록 평가 기간을 조정하는 것이 좋습니다. 또는 사용자가 액세스 키를 사용하여 AWS에 프로그래밍 방식으로 액세스하는 경우 액세스 키가 마지막으로 사용된 정보가 모든 날짜에 대해 정확하므로 해당 정보를 참조할 수 있습니다.

자격 증명 보고서를 다운로드하여 미사용 암호를 찾으려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 Credential Report(자격 증명 보고서)를 선택합니다.
3. 보고서 다운로드를 선택하여 `status_reports_<date>T<time>.csv`라는 쉼표 구분 값(CSV) 파일을 다운로드합니다. 5번째 열에는 날짜가 있는 `password_last_used` 열 또는 다음 중 하나가 있습니다.
 - 해당 사항 없음 – 할당된 암호가 전혀 없는 사용자
 - no_information – IAM이 2014년 10월 20일 암호 수명을 초적하기 시작한 이후 암호를 사용하지 않은 사용자들

미사용 암호를 찾으려면(AWS CLI)

미사용 암호를 찾으려면 다음 명령을 실행합니다.

- `aws iam list-users`는 각자 `PasswordLastUsed` 값이 있는 사용자 목록을 반환합니다. 값이 비어 있는 경우 사용자가 암호가 없거나 2014년 10월 20일 IAM이 암호 수명을 초적하기 시작한 이후 암호가 사용되지 않은 것입니다.

미사용 암호를 찾으려면(AWS API)

미사용 암호를 찾으려면 다음 연산을 호출합니다.

- `ListUsers`는 각각 `<PasswordLastUsed>` 값이 있는 사용자의 집합을 반환합니다. 값이 비어 있는 경우 사용자가 암호가 없거나 2014년 10월 20일 IAM이 암호 수명을 초적하기 시작한 이후 암호가 사용되지 않은 것입니다.

자격 증명 보고서를 다운로드하기 위한 명령어에 대한 자세한 내용은 [자격 증명 보고서 가져오기\(AWS CLI\) \(p. 139\)](#) 단원을 참조하십시오.

미사용 액세스 키 찾기

AWS Management 콘솔을 사용하여 사용자의 액세스 키 사용 정보를 볼 수 있습니다. 사용자 수가 많을 경우 콘솔을 사용하여 자격 증명 보고서를 다운로드하여 각 사용자가 자신의 액세스 키를 마지막으로 사용한 때를 알 수 있습니다. AWS CLI 또는 IAM API에서 그 정보에 액세스할 수도 있습니다.

미사용 액세스 키를 확인하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 필요할 경우 사용자 테이블에 Access key last used(마지막으로 사용한 액세스 키) 열을 추가합니다.
 - a. 테이블 위 맨 오른쪽에서 설정 아이콘()을 선택합니다.

- b. Manage Columns(열 관리)에서 Access key last used(마지막으로 사용한 액세스 키)를 선택합니다.
 - c. 닫기를 선택하여 사용자 목록으로 돌아갑니다.
4. Access key last used(마지막으로 사용한 액세스 키) 열에는 사용자가 프로그래밍 방식으로 AWS에 마지막으로 액세스한 때부터 경과한 일수가 표시됩니다. 이 정보를 통해 지정된 기간 이상 동안 액세스 키를 사용하지 않은 사용자를 확인할 수 있습니다. 액세스 키가 없는 사용자는 이 열에 없음이라고 표시됩니다. 최근에 사용된 적이 없는 액세스 키는 삭제해야 할 자격 증명을 식별하기 위한 좋은 기준이 될 수 있습니다.

자격 증명 보고서를 다운로드하여 미사용 액세스 키를 찾으려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 Credential Report(자격 증명 보고서)를 선택합니다.
3. 보고서 다운로드를 선택하여 `status_reports_<date>T<time>.csv`라는 쉼표 구분 값(CSV) 파일을 다운로드합니다. 열 11부터 13에는 액세스 키 1의 마지막 사용 날짜, 리전 및 서비스 정보가 표시됩니다. 열 16부터 18에는 액세스 키 2에 대해 동일한 정보가 표시됩니다. 값이 해당 사항 없음으로 되어 있는 것은 사용자에게 액세스 키가 없거나 2015년 4월 22일 IAM이 액세스 키 수명을 추적하기 시작한 이후 사용자가 액세스 키를 사용하지 않았다는 것입니다.

미사용 액세스 키를 확인하려면(AWS CLI)

미사용 액세스 키를 찾으려면 다음 명령을 실행합니다.

- `aws iam list-access-keys`는 AccessKeyID를 포함해 사용자의 액세스 키에 대한 정보를 반환합니다.
- `aws iam get-access-key-last-used`는 액세스 키 ID를 받아들여 LastUsedDate 액세스 키의 마지막 사용 및 Region 마지막으로 요청된 서비스의 ServiceName을 포함하는 출력을 반환합니다. LastUsedDate가 없는 경우 2015년 4월 22일 IAM이 액세스 키 수명을 추적하기 시작한 이후 액세스 키가 사용되지 않은 것입니다.

미사용 액세스 키를 확인하려면(AWS API)

미사용 액세스 키를 찾으려면 다음 연산을 호출합니다.

- `ListAccessKeys`는 지정된 사용자와 연결된 액세스 키에 대한 AccessKeyID 값의 목록을 반환합니다.
- `GetAccessKeyLastUsed`는 액세스 키 ID를 받아들여 값의 집합을 반환합니다. LastUsedDate, 액세스 키가 마지막으로 사용된 Region 및 마지막으로 요청된 서비스의 ServiceName이 포함되어 있습니다. 값이 비어 있는 경우 사용자가 액세스 키가 없거나 2015년 4월 22일 IAM이 액세스 키 수명을 추적하기 시작한 이후 액세스 키가 사용되지 않은 것입니다.

자격 증명 보고서를 다운로드하기 위한 명령어에 대한 자세한 내용은 [자격 증명 보고서 가져오기\(AWS CLI\) \(p. 139\)](#) 단원을 참조하십시오.

AWS 계정의 자격 증명 보고서 가져오기

계정의 모든 사용자와 암호, 액세스 키, MFA 디바이스 등 이들의 자격 증명 상태를 나열하는 자격 증명 보고서를 생성하고 다운로드할 수 있습니다. AWS Management 콘솔, [AWS SDK](#) 및 [명령줄 도구](#) 또는 IAM API에서 자격 증명 보고서를 가져올 수 있습니다.

자격 증명 보고서를 사용하면 감사 및 규정 준수에 도움이 됩니다. 이 보고서를 통해 암호, 액세스 키 교체 등 자격 증명의 수명 주기 요구 사항이 어떤 영향을 주는지 감사할 수 있습니다. 외부 감사자에게 이 보고서를 제공하거나 보고서를 직접 다운로드할 권한을 감사자에게 부여할 수 있습니다.

최소 네 시간에 한 번씩 자격 증명 보고서를 생성할 수 있습니다. 보고서를 요청하면 IAM은 먼저 해당 AWS 계정의 보고서가 4시간 이내에 생성되었는지 여부를 확인합니다. 네 시간 이내에 생성된 경우 최신 보고서를 다운로드하고, 계정의 최신 보고서가 생성된 지 네 시간이 넘었거나 해당 계정에 대한 이전 보고서가 없는 경우 IAM에서 새 보고서를 생성하여 이를 다운로드합니다.

주제

- 필요한 권한 (p. 136)
- 보고서 형식 이해하기 (p. 136)
- 자격 증명 보고서 가져오기(콘솔) (p. 139)
- 자격 증명 보고서 가져오기(AWS CLI) (p. 139)
- 자격 증명 보고서 가져오기(AWS API) (p. 139)

필요한 권한

보고서를 생성하고 다운로드하려면 다음 권한이 필요합니다.

- 자격 증명 보고서를 생성하려면 `GenerateCredentialReport`
- 보고서를 다운로드하려면 `GetCredentialReport`

보고서 형식 이해하기

자격 증명 보고서는 CSV(쉼표로 구분된 값) 파일 형식으로 되어 있습니다. 공통 스프레드시트 소프트웨어로 CSV 파일을 열어 분석을 수행하거나 CSV 파일을 프로그래밍 방식으로 사용하고 사용자 지정 분석을 수행하는 애플리케이션을 구축할 수 있습니다.

CSV 파일에는 다음 열이 포함되어 있습니다:

사용자

사용자의 표시 이름입니다.

arn

사용자의 Amazon 리소스 이름(ARN)입니다. ARN에 대한 자세한 내용은 [IAM ARN \(p. 480\)](#) 단원을 참조하십시오.

user_creation_time

사용자가 생성된 날짜 및 시간([ISO 8601 날짜-시간 형식](#))입니다.

password_enabled

사용자에게 암호가 있는 경우 이 값은 `TRUE`입니다. 그렇지 않으면 `FALSE`입니다. AWS 계정 루트 사용자 값은 항상 `not_supported`입니다.

password_last_used

AWS 웹 사이트에 로그인하는 데 AWS 계정 루트 사용자 또는 IAM 사용자의 암호가 마지막으로 사용된 날짜 및 시간([ISO 8601 날짜-시간 형식](#))입니다. 사용자의 마지막 로그인 시간을 캡처하는 AWS 웹 사이트는 AWS Management 콘솔, AWS 토큰 포럼 및 AWS Marketplace입니다. 암호가 5분 내에 두 번 이상 사용된 경우, 첫 번째 사용만 이 필드에 기록됩니다.

- 다음과 같은 경우 이 필드의 값은 `no_information`입니다.
 - 사용자의 암호가 사용된 적이 없는 경우.
 - 암호와 관련된 로그인 데이터가 없는 경우, 예를 들어 IAM에서 2014년 10월 20일에 이 정보를 추적하기 시작한 이후로 사용자의 암호가 사용되지 않은 경우.

- 사용자에게 암호가 없는 경우, 이 필드의 값은 N/A(해당 사항 없음)입니다.

Important

서비스 문제로 인해 암호가 마지막으로 사용된 데이터에 2018년 5월 3일 22:50 PDT ~ 2018년 5월 23일 14:08 PDT 사이의 암호 사용이 포함되어 있지 않습니다. 이는 IAM 콘솔에 표시되는 [마지막 로그인 날짜](#), [IAM 자격 증명 보고서](#)의 암호가 마지막으로 사용된 날짜와 [GetUser API 연산](#)에 의해 반환되는 암호가 마지막으로 사용된 날짜에 영향을 줍니다. 사용자가 해당 기간에 로그인한 경우 반환되는 암호가 마지막으로 사용된 날짜는 사용자가 2018년 5월 3일 이전에 마지막으로 로그인한 날짜입니다. 사용자가 2018년 5월 23일 14:08 PDT 이후에 로그인한 경우 반환되는 암호가 마지막으로 사용된 날짜는 정확합니다.

암호가 마지막으로 사용된 정보를 사용하여 삭제할 사용되지 않은 자격 증명을 식별할 경우(예: 지난 90일 동안 AWS에 로그인하지 않은 사용자 삭제) 2018년 5월 23일 이후의 날짜를 포함하도록 평가 기간을 조정하는 것이 좋습니다. 또는 사용자가 액세스 키를 사용하여 AWS에 프로그래밍 방식으로 액세스하는 경우 액세스 키가 마지막으로 사용된 정보가 모든 날짜에 대해 정확하므로 해당 정보를 참조할 수 있습니다.

password_last_changed

사용자의 암호가 마지막으로 설정된 날짜 및 시간([ISO 8601 날짜-시간 형식](#))입니다. 사용자에게 암호가 없는 경우, 이 필드의 값은 N/A(해당 사항 없음)입니다. AWS 계정(루트)의 값은 항상 not_supported입니다.

password_next_rotation

계정에 암호 교체를 요구하는 [암호 정책](#)이 있는 경우, 사용자가 새 암호를 설정해야 할 때 이 필드에 날짜 및 시간([ISO 8601 날짜-시간 형식](#))이 포함됩니다. AWS 계정(루트)의 값은 항상 not_supported입니다.

mfa_active

사용자에 대해 [멀티 팩터 인증 \(p. 96\)](#)(MFA) 디바이스를 사용하도록 설정된 경우, 이 값은 TRUE입니다. 그렇지 않은 경우 이 값은 FALSE입니다.

access_key_1_active

사용자에게 액세스 키가 있고 액세스 키의 상태가 Active이면, 이 값은 TRUE입니다. 그렇지 않은 경우 이 값은 FALSE입니다.

access_key_1_last_rotated

사용자의 액세스 키가 생성되었거나 마지막으로 변경된 날짜 및 시간([ISO 8601 날짜-시간 형식](#))입니다. 사용자에게 활성 상태의 액세스 키가 없는 경우, 이 필드의 값은 N/A(해당 사항 없음)입니다.

access_key_1_last_used_date

사용자의 액세스 키를 AWS API 요청 서명에 마지막으로 사용한 날짜 및 시간([ISO 8601 날짜-시간 형식](#))입니다. 액세스 키가 15분 내에 두 번 이상 사용된 경우, 첫 번째 사용만 이 필드에 기록됩니다.

다음과 같은 경우 이 필드의 값은 N/A(해당 사항 없음)입니다.

- 사용자에게 액세스 키가 없는 경우.
- 액세스 키가 사용된 적이 없는 경우.
- IAM에서 2015년 4월 22일에 이 정보를 추적하기 시작한 이후로 액세스 키가 사용되지 않은 경우.

access_key_1_last_used_region

액세스 키가 마지막으로 사용된 [AWS 리전](#)입니다. 액세스 키가 15분 내에 두 번 이상 사용된 경우, 첫 번째 사용만 이 필드에 기록됩니다.

다음과 같은 경우 이 필드의 값은 N/A(해당 사항 없음)입니다.

- 사용자에게 액세스 키가 없는 경우.

- 액세스 키가 사용된 적이 없는 경우.
- IAM에서 2015년 4월 22일에 이 정보의 추적을 시작하기 전에 액세스 키가 마지막으로 사용된 경우.
- 마지막으로 사용한 서비스가 리전 전용이 아닌 경우(예: Amazon S3).

access_key_1_last_used_service

액세스 키로 가장 최근에 액세스한 AWS 제품입니다. 이 필드의 값은 서비스의 [네임스페이스](#)를 사용합니다. 예를 들어 Amazon S3의 경우 s3, Amazon EC2의 경우 ec2입니다. 액세스 키가 15분 내에 두 번 이상 사용된 경우, 첫 번째 사용만 이 필드에 기록됩니다.

다음과 같은 경우 이 필드의 값은 N/A(해당 사항 없음)입니다.

- 사용자에게 액세스 키가 없는 경우.
- 액세스 키가 사용된 적이 없는 경우.
- IAM에서 2015년 4월 22일에 이 정보의 추적을 시작하기 전에 액세스 키가 마지막으로 사용된 경우.

access_key_2_active

사용자에게 두 번째 액세스 키가 있고 두 번째 키의 상태가 Active이면, 이 값은 TRUE입니다. 그렇지 않은 경우 이 값은 FALSE입니다.

Note

사용자는 교체하기 쉽도록 최대 두 개의 액세스 키를 보유할 수 있습니다. 액세스 키 교체에 대한 자세한 내용은 [액세스 키 교체 \(p. 92\)](#) 단원을 참조하십시오.

access_key_2_last_rotated

사용자의 두 번째 액세스 키가 생성되었거나 마지막으로 변경된 날짜 및 시간([ISO 8601 날짜-시간 형식](#))입니다. 사용자에게 활성 상태의 두 번째 액세스 키가 있는 경우, 이 필드의 값은 N/A(해당 사항 없음)입니다.

access_key_2_last_used_date

AWS API 요청에 서명하는 데 사용자의 두 번째 액세스 키가 마지막으로 사용된 날짜 및 시간([ISO 8601 날짜-시간 형식](#))입니다. 액세스 키가 15분 내에 두 번 이상 사용된 경우, 첫 번째 사용만 이 필드에 기록됩니다.

다음과 같은 경우 이 필드의 값은 N/A(해당 사항 없음)입니다.

- 사용자에게 두 번째 액세스 키가 없는 경우.
- 사용자의 두 번째 액세스 키가 사용된 적이 없는 경우.
- IAM에서 2015년 4월 22일에 이 정보의 추적을 시작하기 전에 사용자의 두 번째 액세스 키가 마지막으로 사용된 경우.

access_key_2_last_used_region

사용자의 두 번째 액세스 키가 마지막으로 사용된 AWS 리전입니다. 액세스 키가 15분 내에 두 번 이상 사용된 경우, 첫 번째 사용만 이 필드에 기록됩니다. 다음과 같은 경우 이 필드의 값은 N/A(해당 사항 없음)입니다.

- 사용자에게 두 번째 액세스 키가 없는 경우.
- 사용자의 두 번째 액세스 키가 사용된 적이 없는 경우.
- IAM에서 2015년 4월 22일에 이 정보의 추적을 시작하기 전에 사용자의 두 번째 액세스 키가 마지막으로 사용된 경우.
- 마지막으로 사용한 서비스가 리전 전용이 아닌 경우(예: Amazon S3).

access_key_2_last_used_service

사용자의 두 번째 액세스 키로 가장 최근에 액세스한 AWS 서비스입니다. 이 필드의 값은 서비스의 [네임스페이스](#)를 사용합니다. 예를 들어 Amazon S3의 경우 s3, Amazon EC2의 경우 ec2입니다. 액세스 키

가 15분 내에 두 번 이상 사용된 경우, 첫 번째 사용만 이 필드에 기록됩니다. 다음과 같은 경우 이 필드의 값은 N/A(해당 사항 없음)입니다.

- 사용자에게 두 번째 액세스 키가 없는 경우.
- 사용자의 두 번째 액세스 키가 사용된 적이 없는 경우.
- IAM에서 2015년 4월 22일에 이 정보의 추적을 시작하기 전에 사용자의 두 번째 액세스 키가 마지막으로 사용된 경우.

cert_1_active

사용자에게 X.509 서명 인증서가 있고 해당 인증서의 상태가 Active인 경우, 이 값은 TRUE입니다. 그렇지 않은 경우 이 값은 FALSE입니다.

cert_1_last_rotated

사용자의 서명 인증서가 생성되었거나 마지막으로 변경된 날짜 및 시간([ISO 8601 날짜-시간 형식](#))입니다. 사용자에게 활성 상태의 서명 인증서가 있는 경우, 이 필드의 값은 N/A(해당 사항 없음)입니다.

cert_2_active

사용자에게 두 번째 X.509 서명 인증서가 있고 해당 인증서의 상태가 Active인 경우, 이 값은 TRUE입니다. 그렇지 않은 경우 이 값은 FALSE입니다.

Note

사용자는 인증서 교체가 쉽도록 최대 두 개의 X.509 서명 인증서를 보유할 수 있습니다.

cert_2_last_rotated

사용자의 두 번째 서명 인증서가 생성되었거나 마지막으로 변경된 날짜 및 시간([ISO 8601 날짜-시간 형식](#))입니다. 사용자에게 활성 상태의 두 번째 서명 인증서가 있는 경우, 이 필드의 값은 N/A(해당 사항 없음)입니다.

자격 증명 보고서 가져오기(콘솔)

AWS Management 콘솔을 사용하여 자격 증명 보고서를 CSV(쉼표로 구분된 값) 파일로 다운로드할 수 있습니다.

자격 증명 보고서를 다운로드하려면(콘솔)

- AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam>에서 IAM 콘솔을 엽니다.
- 탐색 창에서 자격 증명 보고서(Credential report)를 클릭합니다.
- 보고서 다운로드를 클릭합니다.

자격 증명 보고서 가져오기(AWS CLI)

다음 명령을 실행합니다:

- 자격 증명 보고서를 생성하려면: `aws iam generate-credential-report`
- 자격 증명 보고서를 가져오려면: `aws iam get-credential-report`

자격 증명 보고서 가져오기(AWS API)

다음 연산을 호출합니다.

- 자격 증명 보고서를 생성하려면: `GenerateCredentialReport`

- 자격 증명 보고서를 가져오려면: [GetCredentialReport](#)

IAM과 CodeCommit을 함께 사용: Git 자격 증명, SSH 키 및 AWS 액세스 키

CodeCommit은 AWS 클라우드에서 프라이빗 Git 리포지토리를 호스팅하는 관리형 버전 관리 서비스입니다. CodeCommit을 사용하려면 CodeCommit 리포지토리와 통신하도록 Git 클라이언트를 구성합니다. 이 구성의 일환으로 CodeCommit에서 사용자 인증에 사용할 수 있는 IAM 자격 증명을 제공합니다. IAM에서는 세 가지 유형의 자격 증명으로 CodeCommit을 지원합니다.

- Git 자격 증명: HTTPS를 통해 CodeCommit 리포지토리와 통신하는 데 사용할 수 있는 IAM; 생성 사용자 이름 및 암호 페어입니다.
- SSH 키: SSH를 통해 CodeCommit 리포지토리와 통신하기 위해 IAM 사용자와 연결할 수 있는 로컬로 생성된 퍼블릭-프라이빗 키 페어입니다.
- [AWS 액세스 키 \(p. 88\)](#): HTTPS를 통해 CodeCommit 리포지토리와 통신하기 위해 AWS CLI에 포함된 자격 증명 헬퍼와 함께 사용할 수 있습니다.

각 옵션에 대한 자세한 내용은 다음 단원을 참조하십시오.

CodeCommit에 Git 자격 증명 및 HTTPS 사용(권장)

Git 자격 증명을 사용하여 IAM 사용자에 대한 정적 사용자 이름 및 암호 페어를 생성한 다음 HTTPS 연결에 이러한 자격 증명을 사용합니다. 정적 Git 자격 증명을 지원하는 타사 도구 또는 IDE(통합 개발 환경)에서도 이러한 자격 증명을 사용할 수 있습니다.

이러한 자격 증명은 모든 지원되는 운영 체제에 공통적이고 대부분의 자격 증명 관리 시스템, 개발 환경 및 기타 소프트웨어 개발 도구와 호환되므로 이는 권장되는 방법입니다. 언제든지 Git 자격 증명에 대한 암호를 재설정할 수 있습니다. 또한 자격 증명이 더 이상 필요하지 않은 경우 자격 증명을 비활성화하거나 삭제할 수 있습니다.

Note

Git 자격 증명에 대해 본인의 사용자 이름 또는 암호를 선택할 수 없습니다. IAM에서는 사용자가 AWS에 대한 보안 표준을 준수하고 CodeCommit에서 리포지토리를 보호하도록 돋기 위해 이러한 자격 증명을 생성합니다. 자격 증명은 생성될 때 한 번만 다운로드할 수 있습니다. 따라서 자격 증명을 안전한 장소에 보관하십시오. 필요한 경우 언제든지 암호를 재설정할 수 있지만, 그러면 이전 암호를 사용하여 구성된 연결은 무효화됩니다. 새 암호를 사용하여 연결하려면 연결을 다시 구성해야 합니다.

자세한 내용은 다음 주제 단원을 참조하십시오.

- IAM 사용자를 만들려면 [AWS 계정의 IAM 사용자 생성 \(p. 65\)](#) 단원을 참조하십시오.
- CodeCommit에서 Git 자격 증명을 생성하여 사용하려면 AWS CodeCommit 사용 설명서의 [Git 자격 증명을 사용하는 HTTPS 사용자의 경우](#) 단원을 참조하십시오.

Note

Git 자격 증명을 생성한 이후에 IAM 사용자의 이름을 변경할 경우 Git 자격 증명의 사용자 이름은 변경되지 않습니다. 사용자 이름과 암호는 동일하게 유지되고 계속 유효합니다.

서비스별 자격 증명을 회전하려면

- 현재 사용 중인 서비스별 자격 증명 세트 이외에 두 번째 세트를 만듭니다.

2. 새 자격 증명 세트를 사용하도록 모든 애플리케이션을 업데이트하고 애플리케이션이 작동하는지 확인합니다.
3. 원래 자격 증명의 상태를 "Inactive"로 변경합니다.
4. 모든 애플리케이션이 계속 작동하는지 확인합니다.
5. 비활성 서버별 자격 증명을 삭제합니다.

CodeCommit에 SSH 키 및 SSH 사용

SSH 연결을 사용하여 Git 및 CodeCommit에서 SSH 인증에 사용하는 퍼블릭 및 프라이빗 키 파일을 로컬 시스템에서 만듭니다. 퍼블릭 키를 IAM 사용자와 연결하고 프라이빗 키를 로컬 시스템에 저장합니다. 자세한 내용은 다음 주제 단원을 참조하십시오.

- IAM 사용자를 만들려면 [AWS 계정의 IAM 사용자 생성 \(p. 65\)](#) 단원을 참조하십시오.
- SSH 퍼블릭 키를 만들어 IAM 사용자와 연결하려면 AWS CodeCommit 사용 설명서의 [Linux, macOS, or Unix에서 SSH 연결](#) 또는 [Windows에서 SSH 연결](#) 단원을 참조하십시오.

Note

퍼블릭 키는 ssh-rsa 형식 또는 PEM 형식으로 인코딩해야 합니다. 퍼블릭 키의 최소 비트 길이는 2048비트이고 최대 길이는 16384비트입니다. 이것은 업로드하는 파일의 크기와는 별개입니다. 예를 들어 2048비트 키를 생성할 수 있으며 결과 PEM 파일의 길이는 1679바이트입니다. 퍼블릭 키를 다른 형식 또는 크기로 제공하면 키 형식이 잘못되었다는 오류 메시지가 표시됩니다.

AWS CLI 자격 증명 헬퍼 및 CodeCommit에 HTTPS 사용

Git 자격 증명을 사용한 HTTPS 연결의 대안으로, Git에서 CodeCommit 리포지토리와 상호 작용하기 위해 AWS에 인증해야 할 때마다 암호화 방식으로 서명된 IAM 사용자 자격 증명 또는 Amazon EC2 인스턴스 역할을 사용하도록 허용할 수 있습니다. 이는 IAM 사용자가 필요하지 않은 CodeCommit 리포지토리에 연결하는 유일한 방법입니다. 또한 연동된 액세스 및 임시 자격 증명으로 작동되는 유일한 방법입니다. 비즈니스에 연동된 액세스 또는 임시 자격 증명을 반드시 사용해야 하는 경우를 제외하고 IAM 사용자를 만들어 액세스에 사용하는 것이 좋습니다. 자세한 내용은 다음 주제 단원을 참조하십시오.

- 연동된 액세스에 대한 자세한 내용은 [자격 증명 공급자 및 연동 \(p. 161\)](#) 및 [외부에서 인증된 사용자에게 액세스 권한 제공\(자격 증명 연동\) \(p. 160\)](#) 단원을 참조하십시오.
- 임시 자격 증명에 대한 자세한 내용은 [임시 보안 자격 증명 \(p. 263\)](#) 및 [CodeCommit 리포지토리에 대한 임시 액세스](#) 단원을 참조하십시오.

AWS CLI Credential Helper는 Keychain Access, Windows Credential Management 등과 같은 다른 자격 증명 헬퍼 시스템과 호환되지 않습니다. HTTPS를 통한 자격 증명 헬퍼 연결을 구성할 때 고려해야 할 추가 사항이 있습니다. 자세한 내용은 AWS CodeCommit 사용 설명서의 [Linux, macOS, or Unix 자격 증명 헬퍼를 사용하여 AWS CLI에서 HTTPS 연결](#) 또는 [AWS CLI 자격 증명 헬퍼를 사용하여 Windows에서 HTTPS 연결](#) 단원을 참조하십시오.

서버 인증서 작업

AWS에서 웹 사이트나 애플리케이션에 대한 HTTPS 연결을 활성화하려면 SSL/TLS 서버 인증서가 필요합니다. AWS Certificate Manager(ACM)에서 지원되는 리전에서 사용되는 인증서의 경우, ACM을 사용하여 서버 인증서를 프로비저닝, 관리 및 배포하는 것이 좋습니다. 지원되지 않는 리전에서는 IAM을 인증서 관리자로 사용해야 합니다. ACM에서 지원하는 리전을 알아보려면 AWS General Reference의 [AWS Certificate Manager Certificate Manager 리전 및 엔드포인트](#)를 참조하십시오.

ACM은 서버 인증서를 프로비저닝 및 관리하고 배포하는 데 선호하는 도구입니다. ACM을 사용하면 인증서를 요청하거나 기존 ACM 또는 외부 인증서를 AWS 리소스에 배포할 수 있습니다. ACM이 제공하는 인증서

는 무료이고 자동으로 갱신됩니다. [지원되는 리전](#)에서는 ACM을 사용하여 콘솔에서 또는 프로그래밍 방식으로 서버 인증서를 관리할 수 있습니다. ACM 사용에 대한 자세한 정보는 [AWS Certificate Manager 사용 설명서](#)를 참조하십시오. ACM 인증서 요청에 대한 자세한 정보는 AWS Certificate Manager 사용 설명서의 [퍼블릭 인증서 요청](#) 또는 [프라이빗 인증서 요청](#) 단원을 참조하십시오. 타사 인증서를 ACM으로 가져오는 작업에 대한 자세한 내용은 AWS Certificate Manager 사용 설명서의 [인증서 가져오기](#) 섹션을 참조하십시오.

[ACM에서 지원되지](#) 않는 리전에서 HTTPS 연결을 지원해야 하는 경우에만 IAM을 인증서 관리자로 사용합니다. IAM은 프라이빗 키를 안전하게 암호화하고 암호화된 버전을 IAM SSL 인증서 스토리지에 저장합니다. IAM은 모든 리전에서 서버 인증서 배포를 지원하지만 외부 공급자로부터 AWS에서 사용할 인증서를 얻어야 합니다. ACM 인증서는 IAM에 업로드할 수 없습니다. 또한 인증서는 IAM 콘솔에서 관리할 수 없습니다.

타사 인증서를 IAM에 업로드하는 방법에 대한 자세한 정보는 다음 주제를 참조하십시오.

주제

- [서버 인증서 업로드\(AWS API\)](#) (p. 142)
- [서버 인증서 가져오기\(AWS API\)](#) (p. 143)
- [서버 인증서 목록 조회\(AWS API\)](#) (p. 143)
- [서비스 인증서 이름 변경 또는 경로 업데이트\(AWS API\)](#) (p. 143)
- [서버 인증서 삭제\(AWS API\)](#) (p. 144)
- [문제 해결](#) (p. 144)

서버 인증서 업로드(AWS API)

IAM에 서버 인증서를 업로드하려면 인증서와 함께 그에 딸린 프라이빗 키를 제공해야 합니다. 인증서에 자체 서명이 되어 있지 않은 경우, 인증서 체인도 제공해야 합니다. (자체 서명된 인증서를 업로드하는 경우에는 인증서 체인이 필요하지 않습니다). 인증서를 업로드하기 전에 이 모든 항목이 있는지, 있다면 각 항목이 다음 기준을 충족하는지 확인하십시오.

- 인증서는 업로드 시점에 유효해야 합니다. 유효 기간이 시작되기 전(인증서의 `NotBefore` 날짜) 또는 만료된 후(인증서의 `NotAfter` 날짜)에는 인증서를 업로드할 수 없습니다.
- 프라이빗 키는 암호화되지 않은 것이어야 합니다. 패스워드나 패스프레이즈로 보호된 프라이빗 키는 업로드할 수 없습니다. 암호화된 프라이빗 키의 해독에 대한 도움말은 [문제 해결](#) (p. 144) 단원을 참조하십시오.
- 인증서, 프라이빗 키 및 인증서 체인은 모두 PEM 인코딩되어야 합니다. 이 항목들을 PEM 형식으로 변환하는 작업에 대한 도움말은 [문제 해결](#) (p. 144) 단원을 참조하십시오.

[IAM API](#)를 사용하여 인증서를 업로드하려면 [UploadServerCertificate](#) 요청을 전송하십시오. 다음 예에서는 [AWS Command Line Interface\(AWS CLI\)](#)에서 이 작업을 수행하는 방법을 보여줍니다. 이 예시에서는 다음과 같이 가정합니다.

- PEM 인코딩된 인증서가 `Certificate.pem`이라는 파일에 저장되어 있다.
- PEM 인코딩된 인증서 체인이 `CertificateChain.pem`이라는 파일에 저장되어 있다.
- PEM 인코딩된 비암호화 프라이빗 키가 `PrivateKey.pem`이라는 파일에 저장되어 있다.

다음 예시 명령을 사용하려면 이 파일들의 이름을 바꾸고 [ExampleCertificate](#)을 업로드된 인증서의 이름으로 대체해야 합니다. 하나의 연속선에 명령을 입력합니다. 다음 예시에는 가독성을 높여주는 줄바꿈과 추가 공백이 포함되어 있습니다.

```
$ aws iam upload-server-certificate --server-certificate-name ExampleCertificate  
--certificate-body file://Certificate.pem  
--certificate-chain file://CertificateChain.pem  
--private-key file://PrivateKey.pem
```

선행 명령은 성공적으로 실행되는 경우 [Amazon Resource Name\(ARN\)](#), 표시 이름, 식별자(ID), 만료 날짜 등 업로드된 인증서에 대한 메타데이터를 반환합니다.

Note

Amazon CloudFront에서 사용할 서버 인증서를 업로드하는 경우, `--path` 옵션을 사용하여 경로를 지정해야 합니다. 경로는 `/cloudfront`로 시작해야 하고 후행 슬래시를 포함해야 합니다(예: `/cloudfront/test/`).

Windows PowerShell용 AWS 도구를 사용하여 인증서를 업로드하려면 [Publish-IAMServerCertificate](#)를 사용하십시오.

서버 인증서 가져오기(AWS API)

IAM API를 사용하여 인증서를 조회하려면 [GetServerCertificate](#) 요청을 전송하십시오. 다음 예에서는 AWS CLI에서 이 작업을 수행하는 방법을 보여줍니다. [ExampleCertificate](#)을 조회할 인증서의 이름으로 대체합니다.

```
$ aws iam get-server-certificate --server-certificate-name ExampleCertificate
```

선행 명령은 성공적으로 실행되는 경우 인증서, 인증서 체인(업로드된 경우) 및 인증서 관련 메타데이터를 반환합니다.

Note

업로드 후에는 IAM에서 프라이빗 키를 다운로드하거나 조회할 수 없습니다.

Windows PowerShell용 AWS 도구를 사용하여 인증서를 조회하려면 [Get-IAMServerCertificate](#)를 사용하십시오.

서버 인증서 목록 조회(AWS API)

IAM API를 사용하여 업로드한 서버 인증서의 목록을 조회하려면 [ListServerCertificates](#) 요청을 전송하십시오. 다음 예에서는 AWS CLI에서 이 작업을 수행하는 방법을 보여줍니다.

```
$ aws iam list-server-certificates
```

선행 명령은 성공적으로 실행되는 경우 각 인증서 관련 메타데이터가 담긴 목록을 반환합니다.

Windows PowerShell용 AWS 도구를 사용하여 업로드한 서버 인증서의 목록을 조회하려면 [Get-IAMServerCertificates](#)를 사용하십시오.

서비스 인증서 이름 변경 또는 경로 업데이트(AWS API)

IAM API를 사용하여 서버 인증서의 이름을 변경하거나 경로를 업데이트하려면 [UpdateServerCertificate](#) 요청을 전송하십시오. 다음 예에서는 AWS CLI에서 이 작업을 수행하는 방법을 보여줍니다.

다음 예시 명령을 사용하려면 기존 및 신규 인증서의 이름과 인증서 경로를 바꾸고 명령을 하나의 연속선에 입력해야 합니다. 다음 예시에는 가독성을 높여주는 줄바꿈과 추가 공백이 포함되어 있습니다.

```
$ aws iam update-server-certificate --server-certificate-name ExampleCertificate  
--new-server-certificate-name CloudFrontCertificate  
--new-path /cloudfront/
```

선행 명령은 성공적으로 실행되는 경우 메타데이터를 반환하지 않습니다.

Windows PowerShell용 AWS 도구(들) 사용하여 서버 인증서의 이름을 변경하거나 경로를 업데이트하려면 [Update-IAMServerCertificate](#)을 사용하십시오.

서버 인증서 삭제(AWS API)

IAM API를 사용하여 서버 인증서를 삭제하려면 [DeleteServerCertificate](#) 요청을 전송하십시오. 다음 예에서 AWS CLI에서 이 작업을 수행하는 방법을 보여줍니다.

다음 예시 명령을 사용하려면 [ExampleCertificate](#)을 삭제할 인증서의 이름으로 대체해야 합니다.

```
$ aws iam delete-server-certificate --server-certificate-name ExampleCertificate
```

선행 명령은 성공적으로 실행되는 경우 메타데이터를 반환하지 않습니다.

Windows PowerShell용 AWS 도구를 사용하여 서버 인증서를 삭제하려면 [Remove-IAMServerCertificate](#)을 사용하십시오.

문제 해결

IAM에 인증서를 업로드하려면 인증서, 프라이빗 키 및 인증서 체인이 모두 PEM 인코딩되어 있어야 합니다. 또한 프라이빗 키가 암호화되지 않은 것이어야 합니다. 다음 예시를 참조하십시오.

Example PEM 인코딩된 인증서

```
-----BEGIN CERTIFICATE-----  
Base64-encoded certificate  
-----END CERTIFICATE-----
```

Example PEM 인코딩된 비암호화 프라이빗 키

```
-----BEGIN RSA PRIVATE KEY-----  
Base64-encoded private key  
-----END RSA PRIVATE KEY-----
```

Example PEM 인코딩된 인증서 체인

인증서 체인에는 한 개 이상의 인증서가 포함되어 있습니다. 다음 예시의 경우에는 세 개의 인증서가 포함되어 있지만, 사용자에 따라 인증서 체인에 포함된 인증서가 그보다 많거나 적을 수 있습니다.

```
-----BEGIN CERTIFICATE-----  
Base64-encoded certificate  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
Base64-encoded certificate  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
Base64-encoded certificate  
-----END CERTIFICATE-----
```

이 항목들이 IAM에 업로드하기에 적합한 형식이 아닌 경우, [OpenSSL](#)을 사용하여 적합한 형식으로 변환할 수 있습니다.

인증서 또는 인증서 체인을 DER에서 PEM으로 변환하려면

다음 예시와 같이 OpenSSL x509 명령을 사용합니다. 다음 예시 명령에서 [Certificate.der](#)을 DER 인코딩된 인증서가 포함된 파일의 이름으로 대체합니다. [Certificate.pem](#)을, PEM 인코딩된 인증서를 포함할 출력 파일에 지정하려는 이름으로 바꿉니다.

```
$ openssl x509 -inform DER -in Certificate.der -outform PEM -out Certificate.pem
```

프라이빗 키를 DER에서 PEM으로 변환하려면

다음 예시와 같이 OpenSSL rsa 명령을 사용합니다. 다음 예시 명령에서 *PrivateKey.der*을 DER 인코딩된 프라이빗 키가 포함된 파일의 이름으로 대체해야 합니다. *PrivateKey.pem*을, PEM 인코딩된 프라이빗 키를 포함할 출력 파일에 지정하려는 이름으로 바꿉니다.

```
$ openssl rsa -inform DER -in PrivateKey.der -outform PEM -out PrivateKey.pem
```

암호화된 프라이빗 키를 해독하려면(패스워드나 패스프레이즈 제거)

다음 예시와 같이 OpenSSL rsa 명령을 사용합니다. 다음 예시 명령을 사용하려면 *EncryptedPrivateKey.pem*을 암호화된 프라이빗 키가 포함된 파일의 이름으로 대체해야 합니다. *PrivateKey.pem*을, PEM 인코딩된 비암호화 프라이빗 키를 포함할 출력 파일에 지정하려는 이름으로 바꿉니다.

```
$ openssl rsa -in EncryptedPrivateKey.pem -out PrivateKey.pem
```

인증서 번들을 PKCS#12(PFX)에서 PEM으로 변환하려면

다음 예시와 같이 OpenSSL pkcs12 명령을 사용합니다. 다음 예시 명령에서 *CertificateBundle.p12*를 PKCS#12 인코딩된 인증서 번들이 포함된 파일의 이름으로 대체합니다. *CertificateBundle.pem*을, PEM 인코딩된 인증서 번들을 포함할 출력 파일에 지정하려는 이름으로 대체합니다.

```
$ openssl pkcs12 -in CertificateBundle.p12 -out CertificateBundle.pem -nodes
```

인증서 번들을 PKCS#7에서 PEM으로 변환하려면

다음 예시와 같이 OpenSSL pkcs7 명령을 사용합니다. 다음 예시 명령에서 *CertificateBundle.p7b*를 PKCS#7 인코딩된 인증서 번들이 포함된 파일의 이름으로 대체합니다. *CertificateBundle.pem*을, PEM 인코딩된 인증서 번들을 포함할 출력 파일에 지정하려는 이름으로 대체합니다.

```
$ openssl pkcs7 -in CertificateBundle.p7b -print_certs -out CertificateBundle.pem
```

IAM 그룹

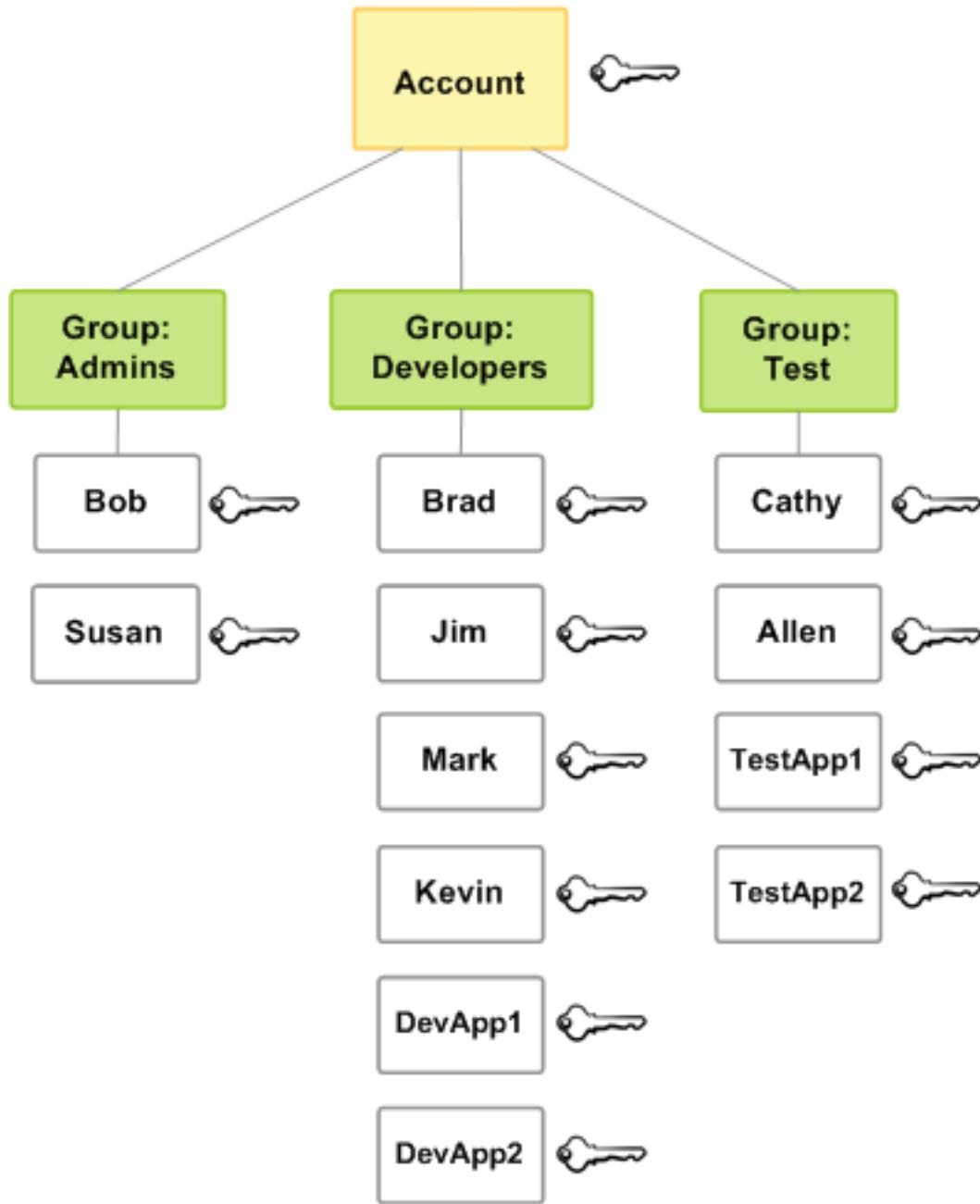
IAM 그룹 (p. 145)은 IAM 사용자들의 집합입니다. 그룹을 활용하면 다수의 사용자들에 대한 권한을 지정함으로써 해당 사용자들에 대한 권한을 더 쉽게 관리할 수 있습니다. 예를 들어 Admins라는 그룹을 만들어 일반적으로 관리자에게 필요한 유형의 권한을 부여할 수 있습니다. 이 그룹에 할당된 권한이 이 그룹에 속하는 모든 사용자에게 자동으로 부여됩니다. 관리자 권한을 필요로 하는 새로운 사용자가 조직에 들어올 경우 해당 사용자를 이 그룹에 추가하여 적절한 권한을 할당할 수 있습니다. 마찬가지로 조직에서 직원의 업무가 바뀌면 해당 사용자의 권한을 편집하는 대신 이전 그룹에서 해당 사용자를 제거한 후 적절한 새 그룹에 추가하면 됩니다.

그룹은 권한 정책에서 Principal로 식별될 수 없기 때문에 IAM에서는 진정한 '자격 증명'이 아니라는 점에 유의하십시오. 그것은 다수의 사용자들에게 한 번에 정책을 연결하는 방법일 뿐입니다.

다음은 그룹이 갖는 몇 가지 중요한 특징입니다.

- 한 그룹에 여러 사용자가 포함될 수 있으며 한 사용자가 다중 그룹에 속할 수 있습니다.
- 그룹은 중첩될 수 없습니다. 즉, 그룹은 사용자만 포함할 수 있으며 다른 그룹은 포함할 수 없습니다.
- AWS 계정의 모든 사용자를 자동으로 포함하는 기본 그룹은 없습니다. 이러한 그룹이 필요한 경우 하나만 들어 새로운 사용자를 각각 해당 그룹에 할당해야 합니다.
- 보유할 수 있는 그룹의 수와 사용자가 속할 수 있는 그룹의 수에는 제한이 있습니다. 자세한 내용은 [IAM 개체 및 객체에 대한 제한 \(p. 485\)](#) 단원을 참조하십시오.

다음 다이어그램에서는 어느 작은 회사의 간단한 예제를 보여줍니다. 회사 소유주는 Admins 그룹을 생성해 회사가 성장함에 따라 사용자들이 다른 사용자들을 생성하고 관리하도록 합니다. Admins 그룹은 Developers 그룹과 Test 그룹을 생성합니다. 이러한 각 그룹은 AWS와 상호 작용하는 사용자(사람 및 애플리케이션: Jim, Brad, DevApp1 등)들로 구성됩니다. 사용자마다 개별적인 보안 자격 증명 세트가 있습니다. 이 예제에서는 각 사용자가 단일 그룹에 속합니다. 하지만 사용자는 다중 그룹에 속할 수 있습니다.



IAM 그룹 생성

그룹을 설정하려면 그룹을 생성해야 합니다. 그런 다음 그룹 내 사용자가 할 수 있는 작업 유형에 따라 권한을 부여해야 합니다. 마지막으로 그룹에 사용자를 추가합니다.

그룹을 만들기 위해 필요한 권한에 대한 자세한 내용은 [IAM 리소스에 액세스하는 데 필요한 권한 \(p. 443\)](#) 단원을 참조하십시오.

IAM 그룹을 만들어 정책을 연결하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 그룹을 클릭한 다음 Create New Group(새 그룹 생성)을 클릭합니다.
3. 그룹 이름 상자에 그룹 이름을 입력한 다음 다음 단계를 클릭합니다.

Important

그룹 이름은 계정 내에서 고유해야 합니다. 대소문자는 구분하지 않습니다. 예를 들어 그룹 **ADMINS**와 그룹 **admins**를 모두 만들 수는 없습니다.

4. 정책 목록에서 그룹 멤버 전체에 적용하고자 하는 정책 이름마다 확인란을 선택합니다. 그런 다음 다음 단계를 클릭합니다.
5. [Create Group]을 클릭합니다.

Administrators 그룹을 설정하는 방법의 예는 [첫 번째 IAM 관리자 및 그룹 생성 \(p. 17\)](#)을 참조하십시오.

IAM 그룹(AWS CLI 또는 AWS API)을 생성하려면

다음 중 하나를 사용하십시오.

- AWS CLI: [aws iam create-group](#)
- AWS API: [CreateGroup](#)

IAM 그룹 관리

Amazon Web Services는 IAM 그룹 관리를 위한 다양한 도구를 제공합니다. 그룹에서 사용자를 추가 및 제거하는 데 필요한 권한에 대한 자세한 내용은 [IAM 리소스에 액세스하는 데 필요한 권한 \(p. 443\)](#)을 참조하십시오.

주제

- [IAM 그룹 표시 \(p. 148\)](#)
- [IAM 그룹에서 사용자 추가 및 제거 \(p. 149\)](#)
- [IAM 그룹에 정책 연결 \(p. 150\)](#)
- [IAM 그룹 이름 바꾸기 \(p. 151\)](#)
- [IAM 그룹 삭제 \(p. 151\)](#)

IAM 그룹 표시

계정의 모든 그룹, 그룹에 속한 사용자 및 한 사용자가 속한 그룹의 목록을 조회할 수 있습니다. AWS CLI 또는 AWS API를 사용하는 경우 특정 경로 접두사를 사용해 전체 그룹의 목록을 조회할 수 있습니다.

계정에 속한 모든 그룹을 표시하는 방법

다음을 수행하십시오.

- [AWS Management 콘솔](#): 탐색 창에서 그룹을 선택합니다.
- AWS CLI: [aws iam list-groups](#)
- AWS API: [ListGroups](#)

특정 그룹에 속한 사용자를 표시하려면

다음을 수행하십시오.

- AWS Management 콘솔: 탐색 창에서 그룹을 선택하고, 그룹 이름을 선택한 후 사용자 탭을 선택합니다.
- AWS CLI: `aws iam get-group`
- AWS API: `GetGroup`

사용자가 속한 모든 그룹을 표시하려면

다음을 수행하십시오.

- AWS Management 콘솔: 탐색 창에서 사용자를 선택하고, 사용자 이름을 선택한 후 그룹 탭을 선택합니다.
- AWS CLI: `aws iam list-groups-for-user`
- AWS API: `ListGroupsForUser`

IAM 그룹에서 사용자 추가 및 제거

그룹을 사용하여 한 번에 여러 사용자에게 동일한 권한 정책을 적용합니다. 그런 다음 IAM 그룹에서 사용자를 추가하거나 제거할 수 있습니다. 이 기능은 사람들이 조직에 들어오거나 조직을 떠날 때 유용합니다.

정책 액세스 보기

정책에 대한 권한을 변경하기 전에 최근 서비스 수준 활동을 검토해야 합니다. 이 기능은 사용 중인 보안 주체(사람 또는 애플리케이션)의 액세스 권한을 제거하지 않으려는 경우 중요합니다. 서비스에서 마지막으로 액세스한 데이터 보기에 대한 자세한 내용은 [서비스에서 마지막으로 액세스한 데이터를 사용하여 권한 줄이기 \(p. 409\)](#) 단원을 참조하십시오.

그룹에서 사용자 추가 또는 제거(콘솔)

AWS Management 콘솔을 사용하여 그룹에서 사용자를 추가 또는 제거할 수 있습니다.

IAM 그룹에 사용자를 추가하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 그룹을 선택한 다음 그룹 이름을 선택합니다.
3. [Users] 탭을 선택한 후 [Add Users to Group]를 선택합니다. 추가할 사용자 옆에 있는 확인란을 선택합니다.
4. 사용자 추가를 선택합니다.

IAM 그룹에서 사용자를 제거하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 그룹을 선택한 다음 그룹 이름을 선택합니다.
3. 사용자 탭을 선택한 후 Remove Users from Group(그룹에서 사용자 제거)를 선택합니다. 제거할 사용자 옆에 있는 확인란을 선택합니다.
4. 사용자 제거를 선택합니다.

그룹에서 사용자 추가 또는 제거(AWS CLI)

AWS CLI를 사용하여 그룹에서 사용자를 추가 또는 제거할 수 있습니다.

IAM 그룹에 사용자를 추가하려면(AWS CLI)

- 다음 명령을 사용합니다.

- [aws iam add-user-to-group](#)

IAM 그룹에서 사용자를 제거하려면(AWS CLI)

- 다음 명령을 사용합니다.
 - [aws iam remove-user-from-group](#)

그룹에서 사용자 추가 또는 제거(AWS API)

AWS API를 사용하여 그룹에서 사용자를 추가 또는 제거할 수 있습니다.

IAM 그룹에 사용자를 추가하려면(AWS API)

- 다음 작업을 완료합니다.
 - [AddUserToGroup](#)

IAM 그룹에서 사용자를 제거하려면(AWS API)

- 다음 작업을 완료합니다.
 - [RemoveUserFromGroup](#)

IAM 그룹에 정책 연결

다음 단계에 설명된 대로 그룹에 AWS 관리형 정책 (p. 312)—즉, AWS가 제공하는 미리 작성된 정책—을 연결할 수 있습니다. 고객 관리형 정책, 즉 생성하는 사용자 지정 권한이 있는 정책을 연결하려면 먼저 정책을 만들어야 합니다. 고객 관리형 정책 만들기에 대한 자세한 내용은 [IAM 정책 만들기 \(p. 377\)](#) 단원을 참조하십시오.

권한 및 정책에 대한 자세한 내용은 [액세스 관리 \(p. 304\)](#)을 참조하십시오.

그룹에 정책을 연결하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.
3. 정책 목록에서 연결할 정책 이름 옆의 확인란을 선택합니다. [Filter] 메뉴와 검색 상자를 사용하여 정책 목록을 필터링할 수 있습니다.
4. Policy actions(정책 작업)와 연결을 차례로 클릭합니다.
5. 필터에서 모든 유형을 선택한 다음 그룹을 클릭합니다.
6. 정책을 연결할 그룹 이름 옆의 확인란을 선택한 다음 정책 연결을 클릭합니다.

그룹(AWS CLI 또는 AWS API)에 정책을 연결하려면

다음 중 하나를 수행하십시오.

- AWS CLI: [aws iam attach-group-policy](#)
- AWS API: [AttachGroupPolicy](#)

IAM 그룹 이름 바꾸기

그룹의 이름 또는 경로를 변경하면 진행됩니다.

- 그룹에 연결된 정책은 이름이 변경되어도 계속 유지됩니다.
- 그룹의 모든 사용자도 이름이 변경되어도 계속 유지됩니다.
- 그룹의 고유 ID는 변동 없이 유지됩니다. 고유 ID에 대한 자세한 내용은 [고유 ID \(p. 483\)](#) 단원을 참조하십시오.

IAM에서는 새 이름을 사용하기 위해 이러한 그룹을 리소스로 참조하는 정책을 자동으로 업데이트하지 않기 때문에 그룹의 이름을 바꿀 때 주의해야 합니다. 그룹의 이름을 바꾸기 전에 모든 정책을 수동으로 확인해 해당 그룹이 이름으로 언급된 모든 정책을 찾아야 합니다. Bob이라는 직원이 회사의 테스트 부서 관리자인 경우를 예로 들어 보겠습니다. Bob에게는 테스트 그룹의 사용자를 추가 및 제거할 수 있는 IAM 사용자에 연결된 정책이 있습니다. 관리자가 그룹의 이름을 변경하거나 그룹 경로를 변경하는 경우, 관리자는 Bob에게 연결된 정책도 업데이트하여 새 이름 또는 새 경로를 사용하도록 해야 합니다. 그렇지 않은 경우 Bob은 그룹에 사용자를 추가할 수도 그룹에서 사용자를 제거할 수도 없습니다.

그룹을 리소스로 참조하는 정책을 찾으려면:

- IAM 콘솔의 탐색 창에서 정책을 선택합니다.
- 정책 유형 드롭다운 목록에서 고객 관리형을 선택하여 사용자 지정 정책만 표시하도록 정책을 필터링합니다.
- 각 정책 이름 옆에 있는 확장 표시기 아이콘을 선택해 정책 요약을 확장합니다.
- 서비스 목록에 IAM이 있으면 선택합니다.
- 리소스 열에서 그룹의 이름을 찾습니다.
- 정책 편집을 선택하여 정책에서 그룹 이름을 변경합니다.

IAM 그룹의 이름을 변경하려면

다음을 수행하십시오.

- [AWS Management 콘솔](#): 탐색 창에서 그룹을 선택한 후 그룹 이름 옆에 있는 확인란을 선택합니다. 페이지 상단의 그룹 작업 목록에서 그룹 이름 편집을 선택합니다. 새 그룹 이름을 입력한 후 예, 편집합니다를 선택합니다.
- AWS CLI: `aws iam update-group`
- AWS API: `UpdateGroup`

IAM 그룹 삭제

AWS Management 콘솔에서 그룹을 삭제하면 콘솔은 모든 그룹 구성원을 자동으로 제거하고 연결된 모든 관리형 정책을 분리하며, 모든 인라인 정책들을 삭제합니다. 그러나 IAM은 이러한 그룹을 리소스로 참조하는 정책을 자동으로 삭제하지 않기 때문에 그룹을 삭제할 때 주의해야 합니다. 그룹을 삭제하기 전에 모든 정책을 수동으로 확인하여 해당 그룹이 이름으로 언급된 모든 정책을 찾아야 합니다. John이라는 직원이 회사의 테스트 부서 관리자인 경우를 예로 들어 보겠습니다. John에게는 테스트 그룹의 사용자를 추가 및 제거할 수 있는 IAM 사용자에 연결된 정책이 있습니다. 관리자는 그룹을 삭제할 경우 John에게 연결된 정책도 삭제해야 합니다.

그룹을 리소스로 참조하는 정책을 찾으려면

- IAM 콘솔의 탐색 창에서 정책을 선택합니다.
- 정책 유형 드롭다운 목록에서 고객 관리형을 선택하여 사용자 지정 정책만 표시하도록 정책을 필터링합니다.

3. 각 정책 이름 옆에 있는 화살표를 선택해 정책 요약을 확장합니다.
4. 서비스 목록에 IAM이 있으면 선택합니다.
5. 리소스 열에서 그룹의 이름을 찾습니다.
6. 정책 삭제를 선택하여 정책을 삭제합니다.

반면에, AWS CLI, Windows PowerShell용 도구 또는 AWS API를 사용하여 그룹을 삭제할 경우에는 먼저 그룹의 사용자를 제거해야 합니다. 그런 다음 이 그룹에 포함된 인라인 정책을 삭제합니다. 그런 다음 그룹에 연결된 관리형 정책을 모두 분리합니다. 이렇게 해야만 그룹을 삭제할 수 있습니다.

IAM 그룹 삭제(콘솔)

AWS Management 콘솔에서 IAM 그룹을 삭제할 수 있습니다.

IAM 그룹을 삭제하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 [Groups]를 선택합니다.
3. 그룹 목록에서 삭제할 그룹 이름 옆의 확인란을 선택합니다. [Filter] 메뉴와 검색 상자를 사용하여 정책 목록을 필터링할 수 있습니다.
4. Group Actions(그룹 작업)을 클릭한 다음 그룹 삭제를 클릭합니다.
5. 확인 상자에서 Yes, Delete(예, 삭제합니다)를 클릭합니다.

IAM 그룹(AWS CLI) 삭제

AWS CLI에서 IAM 그룹을 삭제할 수 있습니다.

IAM 그룹(AWS CLI)을 삭제하려면

1. 그룹에서 모든 사용자를 제거합니다.
 - [aws iam get-group](#)(그룹의 사용자 목록을 가져오는 방법), [aws iam remove-user-from-group](#)(그룹에서 사용자를 제거하는 방법)
2. 그룹에 삽입된 인라인 정책을 모두 삭제합니다.
 - [aws iam list-group-policies](#)(그룹의 인라인 정책 목록을 가져오는 방법), [aws iam delete-group-policy](#)(그룹의 인라인 정책을 삭제하는 방법)
3. 그룹에 추가된 관리형 정책을 모두 분리합니다.
 - [aws iam list-attached-group-policies](#)(그룹에 추가된 관리형 정책 목록을 가져오는 방법), [aws iam detach-group-policy](#)(그룹에서 관리형 정책을 분리하는 방법)
4. 그룹을 삭제합니다.
 - [aws iam delete-group](#)

IAM 그룹(AWS API) 삭제

AWS API를 사용하여 IAM 그룹을 삭제할 수 있습니다.

IAM 그룹(AWS API)을 삭제하려면

1. 그룹에서 모든 사용자를 제거합니다.
 - [GetGroup](#)(그룹의 사용자 목록 확인) 및 [RemoveUserFromGroup](#)(그룹에서 사용자 제거)

2. 그룹에 삽입된 인라인 정책을 모두 삭제합니다.
 - [ListGroupPolicies](#)(그룹의 인라인 정책 목록 확인) 및 [DeleteGroupPolicy](#)(그룹의 인라인 정책 삭제)
3. 그룹에 추가된 관리형 정책을 모두 분리합니다.
 - [ListAttachedGroupPolicies](#)(그룹에 연결된 관리형 정책의 목록 확인) 및 [DetachGroupPolicy](#)(그룹에서 관리형 정책 연결 분리)
4. 그룹을 삭제합니다.
 - [DeleteGroup](#)

IAM역할

IAM 역할은 특정 권한을 가진 계정에 생성할 수 있는 IAM 자격 증명입니다. AWS에서 자격 증명이 할 수 있는 것과 없는 것을 결정하는 권한 정책을 갖춘 AWS 자격 증명이라는 점에서, IAM 역할은 IAM 사용자와 유사합니다. 그러나 역할은 한 사람과만 연관되지 않고 그 역할이 필요한 사람으면 누구든지 맡을 수 있도록 고안되었습니다. 또한 역할에는 그와 연관된 암호 또는 액세스 키와 같은 표준 장기 자격 증명이 없습니다. 그 대신, 역할을 수임하면 역할 세션을 위한 임시 보안 자격 증명을 제공합니다.

역할을 사용하여 일반적으로 AWS 리소스에 액세스할 수 없는 사용자, 애플리케이션 또는 서비스에 액세스 권한을 위임할 수 있습니다. 예를 들어 AWS 계정의 사용자에게 이들이 대개 권한이 없는 리소스에 대한 액세스 권한을 부여하거나 한 AWS 계정의 사용자에게 다른 계정의 리소스에 대한 액세스 권한을 부여해야 할 경우가 있습니다. 또는 모바일 앱에서 AWS 리소스를 사용할 수 있도록 하되 앱에 AWS 키를 내장(교체하기 어렵고 사용자가 추출할 가능성이 있음)하게 원치 않는 경우도 있습니다. 때로는 기업 디렉토리에서처럼 AWS 외부에 정의된 자격 증명을 이미 보유하고 있는 사용자에게 AWS 액세스 권한을 부여해야 하는 경우도 있습니다. 또는 타사에 계정에 대한 액세스 권한을 부여하여 리소스에 대한 감사를 수행할 수 있도록 해야 할 경우도 있을 수 있습니다.

이러한 경우 IAM 역할을 사용하여 AWS 리소스에 대한 액세스 권한을 위임할 수 있습니다. 이 단원에서는 역할 및 역할을 사용할 수 있는 여러 가지 방법, 다양한 접근 방식을 선택하는 경우와 방법, 역할을 생성, 관리, 전환(또는 수임) 및 삭제하는 방법을 소개합니다.

주제

- [역할 용어 및 개념 \(p. 153\)](#)
- [역할에 대한 일반적인 시나리오: 사용자, 애플리케이션 및 서비스 \(p. 156\)](#)
- [자격 증명 공급자 및 연동 \(p. 161\)](#)
- [서비스 연결 역할 사용 \(p. 195\)](#)
- [IAM 역할 생성 \(p. 202\)](#)
- [IAM 역할 사용 \(p. 227\)](#)
- [IAM 역할 관리 \(p. 246\)](#)
- [IAM 역할과 리소스 기반 정책의 차이 \(p. 257\)](#)

역할 용어 및 개념

아래는 역할을 시작하는 데 도움이 되는 몇 가지 기본 용어들입니다.

역할

특정 권한을 가진 계정에 생성할 수 있는 IAM 자격 증명. AWS에서 자격 증명이 할 수 있는 것과 없는 것을 결정하는 권한 정책을 갖춘 AWS 자격 증명이라는 점에서, IAM 역할은 IAM 사용자와 유사합니다. 그러나 역할은 한 사람과만 연관되지 않고 그 역할이 필요한 사람으면 누구든지 맡을 수 있도록 고안되었

습니다. 또한 역할에는 그와 연관된 암호 또는 액세스 키와 같은 표준 장기 자격 증명이 없습니다. 그 대신, 역할을 수임하면 역할 세션을 위한 임시 보안 자격 증명을 제공합니다.

역할은 다음의 주체들이 사용할 수 있습니다.

- 동일한 AWS 계정의 IAM 사용자
- 역할과 다른 AWS 계정의 IAM 사용자
- Amazon Elastic Compute Cloud(Amazon EC2)와 같은 AWS가 제공하는 웹 서비스
- SAML 2.0, OpenID Connect 또는 사용자 지정 구축 자격 증명 브로커와 호환되는 외부 자격 증명 공급자(IdP) 서비스에 의해 인증된 외부 사용자

AWS 서비스 역할

서비스가 사용자를 대신하여 사용자 계정에서 작업을 수행하기 위해 수임한 역할입니다. 일부 AWS 서비스 환경을 설정할 때, 서비스에서 맙을 역할을 정의해야 합니다. 이 서비스 역할에는 서비스가 AWS 리소스에 액세스하는 데 필요한 모든 권한이 포함되어야 합니다. 서비스 역할은 서비스마다 다르지만, 해당 서비스에 대한 문서화된 요구 사항을 충족하는 한 대부분의 경우 권한을 선택할 수 있습니다. 서비스 역할은 해당 계정 내에서만 액세스를 제공하며 다른 계정의 서비스에 대한 액세스를 부여하는 데 사용할 수 없습니다. IAM 내에서 서비스 역할을 만들고, 수정하고, 삭제할 수 있습니다.

EC2 인스턴스의 AWS 서비스 역할

Amazon EC2 인스턴스에서 실행 중인 애플리케이션이 계정에서 작업을 수행하기 위해 맙을 수 있는 특수한 유형의 서비스 역할이 역할은 시작된 EC2 인스턴스에 할당됩니다. 해당 인스턴스에서 실행 중인 애플리케이션은 임시 보안 자격 증명을 검색하고 역할이 허용하는 작업을 수행할 수 있습니다. EC2 인스턴스의 서비스 역할 사용에 대한 세부 정보는 [IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여하기 \(p. 240\)](#)를 참조하십시오.

AWS 서비스 연결 역할

AWS 서비스에 직접 연결된 고유한 유형의 서비스 역할입니다. 서비스 연결 역할은 해당 서비스에서 사전 정의하며 서비스에서 다른 AWS 서비스를 자동으로 호출하기 위해 필요한 모든 권한을 포함합니다. 또한 연결된 서비스는 서비스 연결 역할을 만들고 수정하며 삭제하는 방법을 정의합니다. 서비스는 역할을 자동으로 만들거나 삭제할 수 있습니다. 서비스의 프로세스나 마법사를 사용하여 사용자가 역할을 만들거나 수정하거나 삭제하도록 허용할 수도 있습니다. 또는 사용자가 IAM을 사용하여 역할을 만들거나 삭제하도록 요구할 수도 있습니다. 방법이 어떻든, 서비스 연결 역할은 필요한 권한을 수동으로 추가할 필요가 없으므로 서비스를 더 쉽게 설정할 수 있습니다.

Note

서비스 연결 역할을 시작할 때 이미 서비스를 사용하는 중이라면 계정의 새 역할에 대해 알려주는 이메일을 받게 될 수 있습니다. 이 경우 서비스에서 계정에 서비스 연결 역할을 자동으로 생성합니다. 이 역할을 지원하기 위해 어떤 작업도 수행할 필요가 없으며, 이 역할을 수동으로 삭제할 수 없습니다. 자세한 내용은 [내 AWS 계정에 표시되는 새 역할 \(p. 470\)](#) 단원을 참조하십시오.

서비스 연결 역할의 사용을 지원하는 서비스에 대한 자세한 내용은 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#)를 참조하고 서비스 연결 역할 열에 예가 있는 서비스를 찾으십시오. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 [Yes] 링크를 선택합니다. 서비스에 서비스 연결 역할 만들기, 수정 또는 삭제에 대한 설명서가 포함되어 있지 않으면 IAM 콘솔, AWS CLI 또는 API를 사용하면 됩니다. 자세한 내용은 [서비스 연결 역할 사용 \(p. 195\)](#) 단원을 참조하십시오.

역할 함께 루기

역할 함께 루기는 AWS CLI 또는 API를 통해 역할을 사용하여 두 번째 역할을 수임하는 경우 발생합니다. 예를 들어, User1에게 RoleA 및 RoleB를 수임할 권한이 있다고 가정해 보겠습니다. 또한 RoleA에는 RoleB를 수임할 권한이 있습니다. AssumeRole API 작업에서 User1의 장기 사용자 자격 증명을 사용하여 RoleA를 수임할 수 있습니다. 이 작업은 RoleA의 단기 자격 증명을 반환합니다. 역할 함께 루기에 참여하기 위해 RoleA의 단기 자격 증명을 사용하여 RoleB를 수임할 수 있습니다.

역할 함께 루기를 사용하면 AWS CLI 또는 API 역할 세션은 최대 1시간으로 제한됩니다. AssumeRole API 작업을 사용하여 역할을 수임할 때 DurationSeconds 파라미터를 사용하여 역할 세션 길이를

지정할 수 있습니다. 역할에 대한 [최대 세션 기간 설정 \(p. 228\)](#)에 따라 파라미터 값을 최대 43200 초(12시간)까지 지정할 수 있습니다. 그러나 역할 함께 뮤기를 사용해 역할을 수임하고 1시간보다 큰 DurationSeconds 파라미터 값을 지정하면 작업이 실패합니다.

위임

제어하는 리소스에 대한 액세스를 허용하는 권한을 누군가에게 부여하는 것입니다. 위임에는 리소스를 소유하는 계정(신뢰하는 계정)과 리소스에 액세스해야 하는 사용자를 저장한 계정(신뢰받는 계정) 사이에 신뢰를 설정하는 일이 필요합니다. 신뢰받는 계정과 신뢰하는 계정은 다음 중 하나가 될 수 있습니다.

- 동일 계정
- 조직에서 통제하는 별도의 계정
- 서로 다른 조직이 소유한 2개의 계정

리소스에 대한 액세스 권한을 위임하려면, 2개의 [정책 \(p. 155\)](#)이 연결되어 있는 [IAM 역할을 생성 \(p. 203\)](#)합니다. 권한 정책은 역할 사용자에게 리소스에 대해 의도한 작업을 수행하는 데 필요한 권한을 부여합니다. 신뢰 정책은 역할을 위임하도록 허용된 신뢰할 수 있는 계정 멤버를 지정합니다.

신뢰 정책을 생성할 때 와일드카드(*)를 보안 주체로 지정할 수 없습니다. 신뢰 정책은 신뢰하는 계정의 역할에 연결되어 있고 권한의 절반에 해당합니다. 나머지 절반은 [사용자에게 역할 전환 또는 위임을 허용하는 \(p. 229\)](#) 신뢰받는 계정의 사용자에게 연결된 권한 정책입니다. 임시로 역할을 위임하는 사용자는 자신의 고유 권한을 포기하고 대신 해당 역할의 권한을 위임합니다. 사용자가 역할을 끝내거나 역할 사용을 중지하면 원래 사용자 권한이 자동으로 회복됩니다. [외부 ID \(p. 206\)](#)라 불리는 부가적인 파라미터는 동일한 조직에 의해 제어되지 않는 계정 사이에서 역할을 안전하게 사용하도록 하는 데 도움이 됩니다.

연동

외부 자격 증명 공급자와 AWS 사이에 신뢰 관계를 생성하는 것입니다. 사용자들은 Login with Amazon, Facebook, Google 또는 OpenID Connect(OIDC)와 호환되는 IdP 등의 웹 자격 증명 공급자에 로그인할 수 있습니다. 또한, 사용자는 Microsoft Active Directory 연동 서비스와 같은 Security Assertion Markup Language(SAML) 2.0과 호환되는 엔터프라이즈 자격 증명 시스템에 로그인할 수 있습니다. OIDC 및 SAML 2.0을 사용해 이 외부 자격 증명 공급자와 AWS 사이에 신뢰 관계를 구성할 때, 사용자에게는 IAM 역할이 할당됩니다. 사용자는 임시 보안 자격 증명을 부여받아 AWS 리소스에 대한 액세스가 가능합니다.

연합된 사용자

IAM 사용자를 만드는 대신 AWS Directory Service의 기존 자격 증명, 엔터프라이즈 사용자 디렉터리 또는 웹 자격 증명 공급자를 사용할 수 있습니다. 이 사용자를 연합된 사용자라고 합니다. AWS에서는 [자격 증명 공급자 \(p. 161\)](#)를 통해 액세스가 요청되면 연합된 사용자에게 역할을 할당합니다. 연합된 사용자에 대한 자세한 내용은 IAM 사용 설명서의 [연합된 사용자 및 역할 \(p. 10\)](#)을 참조하십시오.

신뢰 정책

[JSON](#) 형식의 문서로, 역할을 대신하도록 허용할 사용자를 정의합니다. 신뢰할 수 있는 이 개체는 정책에서 문서에 보안 주체로 포함됩니다. 이 문서는 [IAM 정책 언어 \(p. 498\)](#)의 규칙에 따라 작성됩니다.

권한 정책

[JSON](#) 형식의 권한 문서로, 역할이 사용할 수 있는 리소스와 작업을 정의합니다. 이 문서는 [IAM 정책 언어 \(p. 498\)](#)의 규칙에 따라 작성됩니다.

권한 경계

자격 증명 기반 정책이 역할에 부여할 수 있는 최대 권한을 제한하는 정책을 사용하는 고급 기능입니다. 서비스 연결 역할에 권한 경계를 적용할 수 없습니다. 자세한 내용은 [IAM 엔터티에 대한 권한 경계 \(p. 317\)](#) 단원을 참조하십시오.

Principal

작업을 수행하고 리소스에 액세스할 수 있는 AWS의 개체입니다. 보안 주체는 AWS 계정 루트 사용자, IAM 사용자 또는 역할입니다. 리소스에 액세스할 수 있는 권한을 다음 두 가지 중 한 가지 방식으로 부여할 수 있습니다.

- 권한 정책을 사용자에게(직접 또는 그룹을 통해 간접적으로) 또는 역할에게 연결할 수 있습니다.
- [리소스 기반 정책 \(p. 10\)](#)을 지원하는 서비스의 경우 해당 리소스에 연결된 정책의 Principal 요소에서 보안 주체를 식별할 수 있습니다.

AWS 계정을 보안 주체로 참조하는 경우 그 보안 주체는 일반적으로 해당 계정 내에서 정의된 모든 보안 주체를 의미합니다.

Note

역할의 신뢰 정책에서 Principal 요소에 와일드카드(*)를 사용할 수 없습니다.

교차 계정 액세스를 위한 역할

한 계정의 리소스에 대한 액세스 권한을 다른 계정의 신뢰할 수 있는 보안 주체에 부여하는 역할. 역할은 교차 계정 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 AWS 제품을 사용하면 (역할을 프록시로 사용하는 대신) 리소스에 직접 정책을 연결할 수 있습니다. 이를 리소스 기반 정책이라고 하며, 이 정책을 사용하여 다른 AWS 계정의 보안 주체에게 리소스에 대한 액세스 권한을 부여할 수 있습니다. 다음 서비스는 Amazon Simple Storage Service(S3) 버킷, Glacier 볼트, Amazon Simple Notification Service(SNS) 주제 및 Amazon Simple Queue Service(SQS) 대기열 등의 지정된 리소스에 대해 리소스 기반 정책을 지원합니다. 자세한 내용은 [IAM 역할과 리소스 기반 정책의 차이 \(p. 257\)](#) 단원을 참조하십시오.

역할에 대한 일반적인 시나리오: 사용자, 애플리케이션 및 서비스

대부분의 AWS 기능과 마찬가지로 역할 사용에는 일반적으로 2가지 방법이 있습니다. 즉, IAM 콘솔에서 대화식으로 사용하는 것 또는 AWS CLI, Windows PowerShell용 도구 또는 API에서 프로그래밍 방식으로 사용하는 것입니다.

- IAM 콘솔을 사용하는 계정의 IAM 사용자는 역할로 전환하여 콘솔에서 해당 역할의 권한을 임시로 사용할 수 있습니다. 사용자는 자신의 원래 권한을 포기하고 역할에 할당된 권한을 수임합니다. 사용자가 역할을 끝내면 원래 권한이 복원됩니다.
- AWS가 제공하는 애플리케이션 또는 서비스(예: Amazon EC2)에서 AWS에 프로그래밍 방식으로 요청하기 위한 역할에 대한 임시 보안 자격 증명을 요청하여 역할을 수임할 수 있습니다. 역할을 이러한 방식으로 사용함으로써 리소스에 액세스해야 하는 각 엔터티마다 장기 보안 자격 증명을 공유하거나 유지(예를 들면 IAM 사용자를 생성함으로써)할 필요가 없습니다.

Note

이 안내서는 역할로 전환합니다와 역할을 수임합니다라는 표현을 서로 대치할 수 있는 동일한 의미로 사용합니다.

역할을 사용하는 가장 간단한 방법은 IAM 사용자에게 자신 또는 다른 AWS 계정에서 만든 역할로 전환할 권한을 부여하는 것입니다. IAM 사용자는 IAM 콘솔을 통해 역할을 쉽게 전환하여 일반적으로 부여받지 않은 권한을 사용할 수 있습니다. 이후 역할을 끝내 그러한 권한을 포기할 수 있습니다. 이를 통해 중요한 리소스에 잘못 액세스하거나 이를 수정하는 일을 방지할 수 있습니다.

애플리케이션 및 서비스 또는 연동된 외부 사용자에게 액세스 권한을 부여하는 등 역할을 한층 복잡한 방식으로 사용하기 위해 `AssumeRole` API를 호출할 수 있습니다. 이 API 호출은 애플리케이션이 이후의 API 호출에 사용할 수 있는 일련의 임시 자격 증명 세트를 반환합니다. 임시 자격 증명을 사용하여 시도하는 작업에는 연결된 역할에서 부여한 권한만 있습니다. 애플리케이션에서는 콘솔에서 사용자가 하듯이 역할을 "끝낼" 필요가 없습니다. 단지 애플리케이션에서 임시 자격 증명 사용을 중지하고 원래 자격 증명으로 호출을 재개합니다.

연동 사용자는 IdP(자격 증명 공급자)에서 제공하는 자격 증명을 사용하여 로그인합니다. 그 다음 AWS에서 신뢰받는 IdP에 임시 자격 증명을 제공하여 이후의 AWS 리소스 요청에 포함할 수 있도록 사용자에게 전달합니다. 그러한 자격 증명은 할당된 역할에 부여된 권한을 제공합니다.

이 섹션에서는 다음 시나리오의 개요를 제공합니다.

- 소유한 AWS 계정의 IAM 사용자에게 액세스를 제공함으로써 소유한 다른 계정의 리소스에 액세스하도록 하는 경우 (p. 157)
- 타사가 소유한 AWS 계정에 속한 IAM 사용자에게 액세스를 제공하는 경우 (p. 159)
- AWS가 제공하는 서비스를 위해 AWS 리소스에 대한 액세스 권한을 제공하는 경우 (p. 159)
- 외부에서 인증된 사용자에게 액세스 권한 제공(자격 증명 연동) (p. 160)

자신이 소유한 다른 AWS 계정의 IAM 사용자에 대한 액세스 권한 제공

IAM 사용자에게 AWS 계정 내에서 역할을 전환하거나 소유하고 있는 다른 AWS 계정에 정의된 역할로 전환할 수 있는 권한을 부여할 수 있습니다.

Note

소유하지 않은 또는 제어하지 않는 계정에 대한 액세스 권한을 부여하고자 하는 경우, 이 주제 뒷부분의 [타사가 소유한 AWS 계정에 대한 액세스 제공](#) (p. 159) 단원을 참조하십시오.

조직에 중요한 Amazon EC2 인스턴스가 있다고 가정해 봅시다. 사용자에게 인스턴스를 종료할 수 있는 권한을 직접 부여하지 않고, 이러한 권한이 있는 역할을 만들 수 있습니다. 그런 다음 관리자는 인스턴스를 종료해야 하는 경우 해당 역할로 전환할 수 있습니다. 그러면 이러한 인스턴스에 다음과 같은 보호 계층이 추가됩니다.

- 사용자에게 역할을 수임할 권한을 명시적으로 부여해야 합니다.
- 사용자는 AWS Management 콘솔을 사용하여 해당 역할로 능동적으로 전환하거나 AWS CLI 또는 AWS API를 사용하여 역할을 수임해야 합니다.
- 역할에 멀티 팩터 인증(MFA) 보호를 추가하여 MFA 디바이스로 로그인하는 사용자만 역할을 수임할 수 있도록 합니다. 역할을 수임한 사용자가 MFA(멀티 팩터 인증)를 사용하여 처음에 인증을 받도록 역할을 구성하는 방법을 알아보려면 [MFA 보호 API 액세스 구성](#) (p. 122) 단원을 참조하십시오.

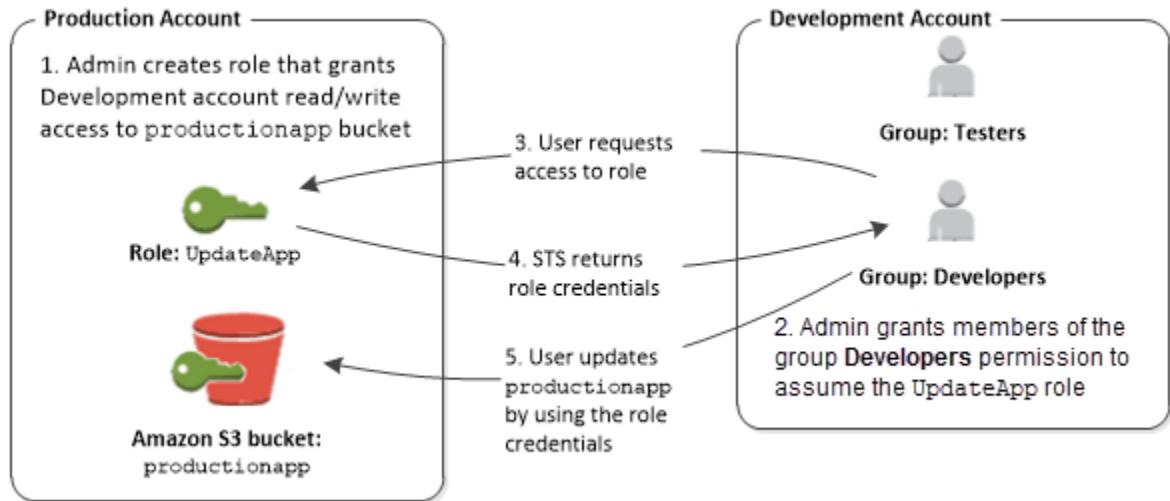
이 방법을 사용하여 최소 액세스 원칙을 적용하는 것이 좋습니다. 다시 말해, 특정 작업이 필요한 경우에만 승격된 권한을 사용하도록 제한하는 것입니다. 역할을 사용하여 중요한 환경을 실수로 변경하는 일을 방지할 수 있습니다. 특히 필요할 때만 역할이 사용되는지 확인하기 위해 중요한 환경을 [감사](#) (p. 293)와 결합하는 경우에 그렇습니다.

이러한 목적을 위해 역할을 만들려면 해당 역할의 신뢰 정책 Principal 요소에서 액세스가 필요한 사용자의 ID로 계정을 지정합니다. 그러면 그러한 다른 계정의 특정 사용자에게 해당 역할로 전환할 수 있는 권한을 부여할 수 있습니다.

한 계정의 사용자는 동일한 또는 다른 계정의 역할로 전환할 수 있습니다. 사용자는 역할을 사용하는 동안 해당 작업만을 수행하고 해당 역할에서 허용한 리소스만 액세스할 수 있지만, 이들의 원래 사용자 권한은 일시 중지된 상태입니다. 사용자가 역할을 끝내면 원래 사용자 권한이 회복됩니다.

분리된 개발 및 프로덕션 계정을 사용한 예제 시나리오

프로덕션 환경에서 개발 환경을 격리하기 위해 조직이 여러 개의 AWS 계정을 갖고 있다고 가정합시다. 개발 계정의 사용자는 프로덕션 계정의 리소스에 액세스해야 하는 경우가 있습니다. 예를 들어, 개발 환경에서 프로덕션 환경으로 업데이트를 승격하려는 경우 교차 계정 액세스 권한이 필요할 수 있습니다. 두 계정을 모두 사용하는 사용자를 위해 별도의 자격 증명(및 암호)을 생성했다 해도 여러 계정에 대한 자격 증명을 관리할 경우 자격 증명 관리가 어려워집니다. 다음 그림을 보면 모든 사용자가 개발 계정에서 관리됩니다. 그러나 일부 개발자에게는 프로덕션 계정에 대한 제한된 액세스 권한이 필요합니다. 개발 계정에는 Testers와 Developers라는 두 개의 그룹이 있으며 각 그룹에는 고유의 정책이 있습니다.



1. 프로덕션 계정에서 관리자는 IAM을 사용하여 그 계정에 UpdateApp 역할을 만듭니다. 관리자는 그 역할에서 개발 계정을 Principal로 지정하는 신뢰 정책을 정의합니다. 이는 개발 계정의 권한이 있는 사용자는 UpdateApp 역할을 사용할 수 있다는 것을 뜻합니다. 또한, 관리자는 이 역할의 사용자가 productionapp이라는 Amazon S3 버킷에 대한 읽기 및 쓰기 권한을 보유하도록 지정하는 역할에 대한 권한 정책을 정의합니다.

그런 다음 관리자는 적절한 정보를 이 역할을 수임해야 하는 대상과 공유합니다. 그러한 정보로는 계정 번호와 역할 이름(AWS 콘솔 사용자들에 대한) 또는 Amazon 리소스 이름(ARN)(AWS CLI 또는 AWS API 액세스용)이 있습니다. 이 역할의 ARN은 `arn:aws:iam::123456789012:role/UpdateApp`과 같은 형태를 띕니다. 여기에서 역할의 이름은 UpdateApp이고, 역할이 생성된 계정 번호는 123456789012입니다.

Note

관리자는 역할을 수임하는 사용자가 먼저 멀티 팩터 인증(MFA)을 사용하여 인증을 받도록 역할을 구성할 수도 있습니다. 자세한 내용은 [MFA 보호 API 액세스 구성 \(p. 122\)](#) 단원을 참조하십시오.

2. 개발 계정에서 관리자는 Developer 그룹의 구성원에게 이 역할로 전환할 수 있는 권한을 부여합니다. 이를 수행하려면 Developers 그룹에 UpdateApp 역할에 대한 AWS Security Token Service(AWS STS) AssumeRole API를 호출할 권한을 부여하면 됩니다. 이제 개발 계정의 Developers 그룹에 속한 모든 IAM 사용자는 프로덕션 계정의 UpdateApp 역할로 전환할 수 있습니다. Developer 그룹에 속하지 않은 다른 사용자는 이 역할로 전환할 수 있는 권한이 없으므로 프로덕션 계정의 S3 버킷에 액세스할 수 없습니다.
3. 사용자가 이 역할로의 전환을 요청:
 - AWS 콘솔: 탐색 표시줄에서 계정 이름을 선택하고 Switch Role(역할 전환)을 선택합니다. 계정 ID(또는 별칭) 및 역할 이름을 지정합니다. 아니면 사용자는 관리자가 이메일로 보낸 링크를 클릭해도 됩니다. 링크를 누르면 세부 정보가 이미 채워져 있는 Switch Role(역할 전환) 페이지로 이동합니다.
 - AWS API/AWS CLI: 개발 계정의 Developers 그룹에 속한 사용자는 AssumeRole 함수를 호출하여 UpdateApp 역할에 대한 자격 증명을 가져옵니다. UpdateApp 역할의 ARN을 이 호출의 일부로 지정합니다. Testers 그룹의 사용자가 동일한 요청을 하는 경우에는 요청이 실패하는데, 이는 Testers가 AssumeRole 역할 ARN을 위해 UpdateApp을 호출할 권한이 없기 때문입니다.
4. AWS STS는 임시 자격 증명을 반환합니다.
 - AWS 콘솔: AWS STS에서 그 요청이 신뢰할 수 있는 대상(개발 계정)에서 온 것인지 확인하기 위해 그 요청에 대해 역할의 신뢰 정책을 확인합니다. 확인 후 AWS STS에서 AWS 콘솔로 [임시 보안 자격 증명](#)을 반환합니다.

- API/CLI: AWS STS에서 신뢰할 수 있는 대상(Development 계정)이 요청을 보낸 것인지 확인하기 위해 역할의 신뢰 정책에 대한 요청을 확인합니다. 확인 후 AWS STS에서 해당 애플리케이션으로 [임시 보안 자격 증명](#)을 반환합니다.
5. 임시 자격 증명은 AWS 리소스에 대한 액세스를 허용합니다.
- AWS 콘솔: AWS 콘솔은 이후의 모든 콘솔 작업에서 사용자를 대신하여 임시 자격 증명을 사용합니다. 이 경우에 그 작업이란 productionapp 버킷에 대한 읽기 및 쓰기입니다. 이 콘솔은 프로덕션 계정의 다른 리소스에는 액세스할 수 없습니다. 사용자가 역할을 끝내면 사용자의 권한은 이 역할로 전환하기 전에 보유한 원래의 권한으로 돌아갑니다.
 - API/CLI: 이 애플리케이션에서는 임시 보안 자격 증명을 사용하여 productionapp 버킷을 업데이트합니다. 이 애플리케이션은 임시 보안 자격 증명을 통해 productionapp 버킷에 대한 읽기 및 쓰기만 할 수 있으며 프로덕션 계정의 다른 리소스에는 액세스할 수 없습니다. 애플리케이션은 역할을 종료하지 않아도 되지만 대신에 임시 자격 증명 사용을 중지하고 이후의 API 호출에서 다시 원래의 자격 증명을 사용합니다.

타사가 소유한 AWS 계정에 대한 액세스 제공

타사가 조직의 AWS 리소스에 액세스해야 하는 경우 역할을 사용하여 해당 사용자에게 그에 대한 액세스 권한을 위임할 수 있습니다. 예를 들어, 타사가 AWS 리소스를 관리하는 서비스를 제공할 경우 IAM 역할을 사용하면 AWS 보안 자격 증명을 공유하지 않고 외부 사용자에게 AWS 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 대신 제3자는 귀하의 AWS 계정에 만든 역할로 가장하여 귀하의 AWS 리소스에 액세스할 수 있습니다.

해당 사용자가 수임할 수 있는 역할을 생성하려면 타사가 다음 정보를 제공해야 합니다.

- 타사의 AWS 계정 ID 역할에 대한 신뢰 정책을 정의할 때 AWS 계정 ID를 보안 주체로 지정합니다.
- 역할을 고유하게 연결하는 데 사용하는 외부 ID. 외부 ID는 여러분과 타사가 알고 있는 임의의 비밀 식별자일 수 있습니다. 예를 들어, 여러분과 타사가 사용하는 인보이스 ID를 사용할 수 있지만 타사의 이름이나 전화번호와 같이 추측 가능한 것은 사용하지 마십시오. 역할에 대한 신뢰 정책을 정의할 때 이 ID를 지정해야 합니다. 타사가 역할을 수임할 때 이 ID를 제공해야 합니다. 외부 ID에 대한 자세한 내용은 [AWS 리소스에 대한 액세스를 타사에 부여할 때 외부 ID를 사용하는 방법](#) (p. 206)을 참조하십시오.
- 귀사의 AWS 리소스를 사용하기 위해 타사에게 필요한 권한. 역할의 권한 정책을 정의할 때 이러한 권한을 지정해야 합니다. 이 정책은 타사에서 수행할 수 있는 작업과 액세스할 수 있는 리소스를 정의합니다.

역할을 정의한 후에는 역할의 Amazon 리소스 이름(ARN)을 타사에 제공해야 합니다. 타사가 역할을 수임하려면 해당 역할의 ARN이 필요합니다.

타사에게 액세스 권한을 위임하는 역할을 생성하는 방법은 [AWS 리소스에 대한 액세스를 타사에 부여할 때 외부 ID를 사용하는 방법](#) (p. 206)을 참조하십시오.

Important

타사에 AWS 리소스에 대한 액세스 권한을 부여하는 경우 타사는 여러분이 정책에서 지정하는 모든 리소스에 액세스할 수 있습니다. 타사의 리소스 사용에 대해서는 여러분에게 과금됩니다. 타사의 리소스 사용을 적절하게 제한해야 합니다.

AWS 서비스에 액세스 권한 제공

많은 AWS 서비스에서는 역할을 사용하여 해당 서비스가 액세스할 수 있는 대상을 제어해야 합니다. 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임한 역할을 [서비스 역할](#) (p. 154)이라고 합니다. 역할이 서비스에 대해 특수한 목적을 수행하는 경우 [EC2 인스턴스의 서비스 역할](#) (p. 154) 또는 [서비스 연결 역할](#) (p. 154)로 분류할 수 있습니다. 서비스에서 역할을 사용하는지 여부와 서비스에서 사용할 역할을 할당하는 방법을 알아보려면 서비스별 [AWS 문서](#)를 참조하십시오.

역할을 생성해 AWS가 제공하는 서비스에 액세스 권한을 위임하는 것에 대한 자세한 내용은 [AWS 서비스에 대한 권한을 위임할 역할 생성](#) (p. 210) 단원을 참조하십시오.

외부에서 인증된 사용자에게 액세스 권한 제공(자격 증명 연동)

사용자는 이미 기업 디렉토리 등 AWS 외부에 자격 증명을 보유할 수 있습니다. 그러한 사용자가 AWS 리소스를 사용해야 하는 경우(또는 그러한 리소스에 액세스하는 애플리케이션을 사용해야 하는 경우), AWS 보안 자격 증명이 필요합니다. IAM 역할을 사용하여 자격 증명이 내 조직 또는 타사 IdP(자격 증명 공급자)로부터 연동되는 사용자에 대한 권한을 지정할 수 있습니다.

모바일 또는 웹 기반 앱 사용자들을 Amazon Cognito와 연동하기

AWS 리소스에 액세스하는 모바일 또는 웹 기반 앱을 만드는 경우, 이 앱에는 AWS에 프로그래밍 방식으로 요청하기 위해 보안 자격 증명이 필요합니다. 대부분의 모바일 애플리케이션 시나리오의 경우 [Amazon Cognito](#) 사용을 권장합니다. 이 서비스와 함께 [iOS용 Mobile SDK](#), [Android 및 Fire OS용 AWS Mobile SDK](#)를 사용하여 사용자 고유 자격 증명을 만들고 AWS 리소스에 대한 보안 액세스를 인증할 수 있습니다. Amazon Cognito는 다음 단원에 나열한 것과 동일한 자격 증명 제공자를 지원하며 [개발자 인증 자격 증명](#) 및 인증되지 않은(게스트) 액세스도 지원합니다. Amazon Cognito는 디바이스를 바꿔 가며 이용해도 데이터를 보존하도록 사용자 데이터 동기화를 위한 API 작업도 제공합니다. 자세한 내용은 [모바일 앱을 위한 Amazon Cognito 사용 \(p. 162\)](#) 단원을 참조하십시오.

사용자를 퍼블릭 자격 증명 서비스 공급자 또는 OpenID Connect와 연동하기

가능한 경우에는 언제든지 모바일 및 웹 기반 애플리케이션 시나리오를 위해 Amazon Cognito를 사용하십시오. Amazon Cognito는 퍼블릭 자격 증명 공급자 서비스를 이용해 대부분의 백그라운드 작업을 수행합니다. 동일한 타사 서비스를 사용하며 익명 로그인을 지원하기도 합니다. 그러나 고급 시나리오의 경우에는 Login with Amazon, Facebook, Google, 또는 OpenID Connect(OIDC)와 호환되는 모든 IdP로 직접 작업할 수 있습니다. 이를 서비스 중 한 가지를 이용한 웹 자격 증명 연동에 대한 자세한 내용은 [웹 자격 증명 연동에 대하여 \(p. 162\)](#)을 참조하십시오.

SAML 2.0으로 사용자 연동하기

조직에서 SAML 2.0(Security Assertion Markup Language 2.0)을 지원하는 자격 증명 공급자 소프트웨어 패키지를 이미 사용하는 경우 IdP(자격 증명 공급자)인 조직과 서비스 공급자인 AWS 간에 신뢰를 형성할 수 있습니다. 그러면 SAML을 사용하여 사용자에게 AWS Management 콘솔에 대한 연동 SSO(Single-Sign On) 또는 AWS API 작업을 호출하기 위한 연동 액세스를 제공할 수 있습니다. 예를 들어 회사가 Microsoft Active Directory와 Active Directory Federation Services를 이용한다면, SAML 2.0을 사용해 연동할 수 있습니다. SAML 2.0를 이용한 사용자 연동에 대한 세부 정보는 [SAML 2.0 기반 연동에 대하여 \(p. 167\)](#)을 참조하십시오.

사용자 지정 자격 증명 브로커 애플리케이션 생성에 의한 사용자 연동

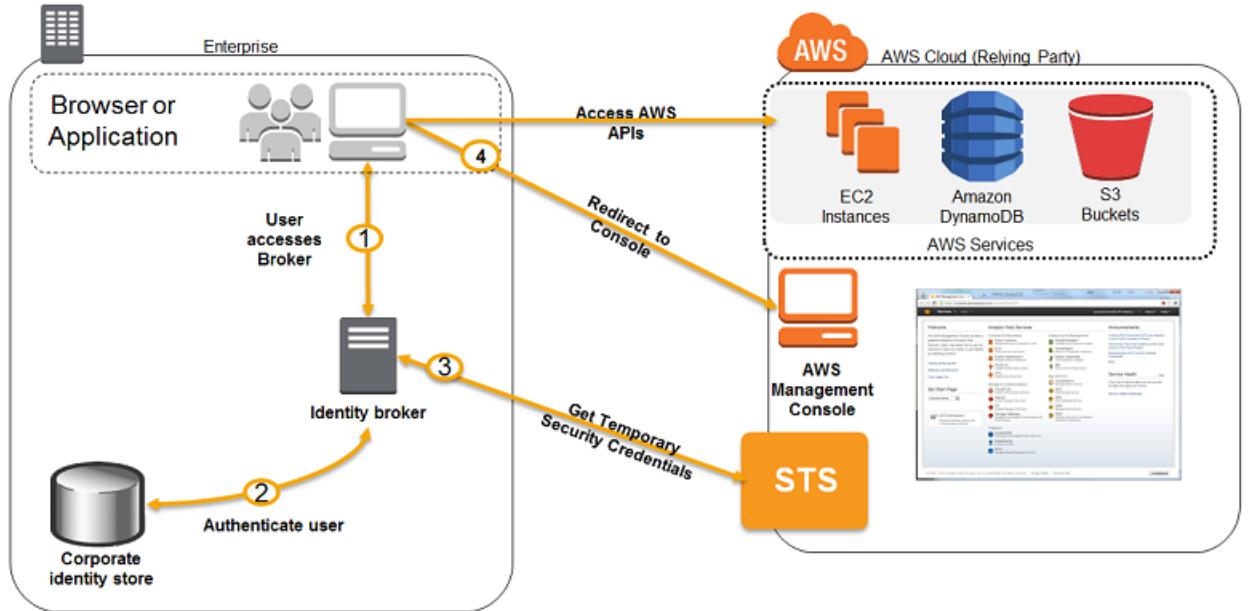
자격 증명 스토어가 SAML 2.0과 호환되지 않는다면, 사용자 지정 자격 증명 브로커 애플리케이션을 구축해 비슷한 기능을 수행할 수 있습니다. 브로커 애플리케이션이 사용자를 인증하고, AWS에게 사용자를 위한 임시 자격 증명을 요청한 다음, 이를 사용자에게 제공해 AWS 리소스에 액세스하도록 합니다.

예를 들어 Example Corp.에 회사의 AWS 리소스에 액세스하는 내부 애플리케이션을 실행해야 하는 직원들이 많다고 합시다. 직원들은 이미 회사 자격 증명 및 인증 시스템에서 자격 증명을 갖고 있어서 Example Corp.은 각 직원들에 대해 별도의 IAM 사용자를 생성하길 원하지 않습니다.

Example Corp의 개발자인 Bob은 내부 애플리케이션이 회사의 AWS 리소스에 액세스하도록 하기 위해 사용자 지정 자격 증명 브로커 애플리케이션을 개발합니다. 그 애플리케이션은 직원들이 기존 Example Corp. 자격 증명 및 인증 시스템에 로그인된 상태인지 확인하는데, 그 시스템은 LDAP, Active Directory, 또는 다른 시스템을 사용할 수 있습니다. 그 다음에 자격 증명 브로커 애플리케이션은 직원들에 대한 임시 보안 자격 증명을 획득합니다. 이 시나리오는 AWS 리소스에 접근할 필요가 있는 애플리케이션들이 모두 회사 네트워크 내에서 실행되고 그 회사는 기존 인증 시스템을 보유하고 있다는 점만 제외하면 이전 것(사용자 지정 인증 시스템을 사용하는 모바일 앱)과 유사합니다.

임시 보안 자격 증명을 얻기 위해 자격 증명 브로커 애플리케이션은 밥(Bob)이 사용자들에 대한 정책을 어떻게 관리하고자 하는지, 그리고 임시 자격 증명이 언제 만료되는지에 따라 `AssumeRole` 또는 `GetFederationToken`을 호출해 임시 보안 자격 증명을 획득합니다. (이러한 API 작업 간의 차이점을 보려

면 [임시 보안 자격 증명 \(p. 263\)](#) 및 [사용자 임시 보안 자격 증명에 대한 권한 제어 \(p. 277\)](#)를 참조하십시오.) 호출은 AWS 액세스 키 ID, 보안 액세스 키, 세션 토큰으로 구성된 임시 보안 자격 증명을 반환합니다. 자격 증명 브로커 애플리케이션은 이 임시 보안 자격 증명을 내부 회사 애플리케이션에서도 사용할 수 있게 해 줍니다. 그 앱은 그 임시 자격 증명을 사용해 AWS를 직접 호출할 수 있습니다. 그 앱은 자격 증명이 만료될 때까지 캐싱한 다음, 새로운 일련의 임시 자격 증명을 요청합니다. 다음은 이 시나리오를 설명한 그림입니다.



이 시나리오에는 다음과 같은 속성이 있습니다.

- 자격 증명 브로커 애플리케이션은 임시 보안 자격 증명을 만들 수 있도록 IAM의 보안 토큰 서비스(STS) API에 액세스할 수 있는 권한이 있습니다.
- 신원 증명 브로커 애플리케이션을 통해 기존 인증 시스템 내에서 직원이 인증되었는지 확인할 수 있습니다.
- 사용자에게 AWS Management Console에 액세스할 수 있는 임시 URL[Single-Sign-On(SSO)]이라고 함]이 제공됩니다.

이 시나리오에 기술된 자격 증명 브로커 애플리케이션과 유사한 샘플 애플리케이션을 보려면, AWS Sample Code & Libraries의 [Identity Federation Sample Application for an Active Directory Use Case](#)를 참조하십시오. 임시 보안 자격 증명 생성에 대한 자세한 내용은 [임시 보안 자격 증명 요청하기 \(p. 265\)](#)를 참조하십시오. AWS Management Console에 액세스하는 연동된 사용자에 대한 자세한 내용은 다음([SAML 2.0 연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하기 \(p. 185\)](#))을 참조하십시오.

자격 증명 공급자 및 연동

AWS 외부의 사용자 자격 증명을 이미 관리하고 있는 경우 AWS 계정에서 IAM 사용자를 생성하는 대신 IAM 자격 증명 공급자를 사용할 수 있습니다. 자격 증명 공급자(IdP)를 사용하면 AWS 외부의 사용자 자격 증명을 관리할 수 있고 이 외부 사용자 자격 증명에 계정의 AWS 리소스에 대한 사용 권한을 부여할 수 있습니다. 기업 사용자 디렉터리처럼 조직 내에 이미 고유의 자격 증명 시스템이 있다면 이 방법이 유용합니다. 그 밖에 AWS 리소스에 액세스해야 하는 모바일 앱이나 웹 애플리케이션을 개발할 때도 효과적입니다.

IAM 자격 증명 공급자를 사용하면 사용자 지정 로그인 코드를 생성할 필요도, 그리고 자신의 사용자 자격 증명을 관리할 필요도 없습니다. IdP에서 이러한 작업을 대신 수행합니다. 외부 사용자는 Login with Amazon, Facebook 또는 Google과 같은 널리 알려진 IdP를 통해 로그인합니다. 사용자에게 계정의 AWS 리소스를 사용할 수 있는 외부 자격 증명 권한을 부여할 수 있습니다. IAM 자격 증명 공급자는 애플리케이션으로 액세스 키 같은 장기 보안 자격 증명을 배포하거나 포함할 필요가 없으므로 AWS 계정의 보안에 도움이 됩니다.

IdP를 사용하기 위해서는, 먼저 IAM 자격 증명 공급자 엔터티를 생성하여 AWS 계정과 IdP 사이에 신뢰 관계를 설정해야 합니다. IAM은 [OpenID Connect\(OIDC\)](#) 또는 [SAML 2.0\(Security Assertion Markup Language 2.0\)](#)과 호환되는 IdP를 지원합니다. AWS에서 해당 IdP 중 하나를 사용하는 것에 대한 자세한 정보는 다음을 참조하십시오.

- 웹 자격 증명 연동에 대하여 (p. 162)
- SAML 2.0 기반 연동에 대하여 (p. 167)

IAM 자격 증명 공급자 엔터티를 생성하여 호환되는 IdP와 AWS 사이에 신뢰 관계를 구축하는 방법에 대한 자세한 정보는 [IAM 자격 증명 공급자 생성 \(p. 171\)](#) 단원을 참조하십시오.

웹 자격 증명 연동에 대하여

모바일 디바이스에서 실행되고 Amazon S3 및 DynamoDB를 사용해 플레이어와 점수 정보를 저장하는 게임과 같은 AWS 리소스에 액세스하는 모바일 앱을 만들고 있다고 상상해 봅시다.

그런 앱을 만들 때 AWS 액세스 키로 서명해야 하는 AWS 서비스에 요청을 할 것입니다. 그러나 암호화된 스토어에서 일지라도 사용자가 디바이스에 다운로드하는 앱으로 장기 AWS 자격 증명을 포함 또는 배포하지 말 것을 강력하게 권고합니다. 대신 앱을 구축해 웹 자격 증명 연동을 사용하여 필요시 동적으로 임시 AWS 보안 자격 증명을 요청할 수 있도록 하십시오. 제공된 임시 자격 증명은 모바일 앱에 필요한 작업을 수행하기 위해 필요한 권한만을 지닌 AWS 역할에 매핑됩니다.

웹 자격 증명 연동을 사용하면 사용자 지정 로그인 코드를 생성하거나 자신의 사용자 자격 증명을 관리할 필요가 없습니다. 대신에, 앱의 사용자는 Login with Amazon, Facebook, Google 또는 다른 [OpenID Connect\(OIDC\)](#) 호환 IdP와 같은 널리 알려진 외부 자격 증명 공급자(IdP)를 사용해 사용자가 로그인할 수 있습니다. 앱의 사용자는 인증 토큰을 받은 다음, AWS에서 이 토큰을 AWS 계정의 리소스를 사용할 수 있는 권한을 가진 IAM 역할에 매핑되는 임시 보안 자격 증명으로 바꿉니다. IdP를 사용하면 AWS 계정을 안전하게 보호할 수 있다는 이점이 있습니다. 애플리케이션으로 장기 보안 자격 증명을 포함하고 배포할 필요가 없기 때문입니다.

대부분의 시나리오에서 [Amazon Cognito](#)를 사용할 것을 권장하는 이유는 Amazon Cognito는 자격 증명 브로커의 역할을 하고 연동 작업의 대부분을 수행하기 때문입니다. 자세한 정보는 [모바일 앱을 위한 Amazon Cognito 사용 \(p. 162\)](#) 단원을 참조하십시오.

Amazon Cognito를 사용하지 않는다면 웹 IdP(예: Facebook, Google 또는 기타 OIDC 호환 IdP)와 상호작용하는 코드를 작성한 다음, `AssumeRoleWithWebIdentity` API를 호출해 그 IdP에서 얻는 인증 토큰을 AWS 임시 보안 자격 증명과 바꾸어야 합니다. 기존 앱에 대해 이러한 접근 방식을 이미 사용해왔다면 그것을 계속 사용할 수 있습니다.

주제

- [모바일 앱을 위한 Amazon Cognito 사용 \(p. 162\)](#)
- [모바일 앱을 위한 웹 자격 증명 연동 API 작업 사용 \(p. 164\)](#)
- [웹 자격 증명 연동을 사용해 사용자 식별하기 \(p. 165\)](#)
- [웹 자격 증명 연동 관련 추가 리소스 \(p. 167\)](#)

모바일 앱을 위한 Amazon Cognito 사용

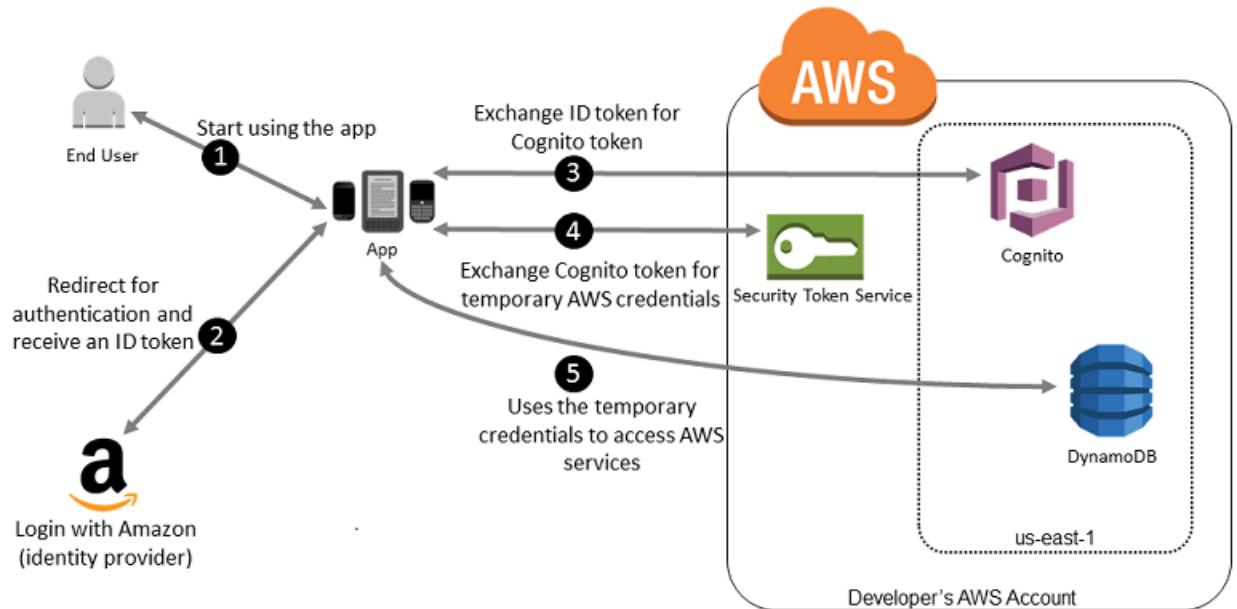
웹 자격 증명 페더레이션 사용에서 선호되는 방식은 [Amazon Cognito](#)를 사용하는 것입니다. 예를 들어 개발자 Adele이 점수와 프로필이 같은 사용자 데이터가 Amazon S3와 Amazon DynamoDB에 저장되는 모바일 디바이스를 위한 게임을 만들고 있다고 합시다. Adele은 그 디바이스에 이 데이터를 로컬 저장하고 Amazon Cognito를 사용해 여러 디바이스에 걸쳐 데이터를 동기화할 수도 있습니다. Adele은 보안 및 유지 보수 상의 이유로 장기 AWS 보안 자격 증명은 게임과 함께 배포되어서는 안 된다는 것을 알고 있습니다. 또한, 게임 사용자가 아주 많을 수도 있다는 것을 알고 있습니다. 이 모든 이유로 인해 Adele은 각 플레이어에 대해 IAM에서 새로운 사용자 자격 증명을 생성하기 원하지 않습니다. 대신에 사용자가 Login with Amazon, Facebook,

Google 또는 OpenID Connect(OIDC) 호환 자격 증명 공급자(IdP)와 같은 널리 알려진 외부 IdP를 통해 이미 설정한 자격 증명을 사용해 로그인할 수 있도록 게임을 구축합니다. Adele의 게임은 이러한 공급자 중 하나의 인증 메커니즘을 이용해 사용자의 자격 증명을 확인할 수 있습니다.

모바일 앱을 활성화해 자신의 AWS 리소스에 액세스하기 위해 Adele은 먼저 자신이 선택한 IdP로 개발자 ID를 등록합니다. Adele은 이들 각 공급자로 애플리케이션을 구성하기도 합니다. Adele은 게임에 대한 Amazon S3 버킷 및 DynamoDB 표가 저장된 AWS 계정에서 Amazon Cognito를 사용해 게임이 필요한 권한을 정확하게 정의하는 IAM 역할을 생성합니다. Adele이 OIDC IdP를 사용하고 있다면, IAM OIDC 자격 증명 공급자 엔터티를 생성하여 자신의 AWS 계정과 IdP 사이에 신뢰를 설정하기도 합니다.

앱의 코드에서 Adele은 자신이 이전에 구성한 IdP에 대한 로그인 인터페이스를 호출합니다. IdP는 사용자가 로그인하도록 허용하는 모든 세부 정보를 처리하고 앱은 공급자에게서 OAuth 액세스 토큰 또는 OIDC ID 토큰을 얻습니다. Adele의 앱은 이 인증 정보를 주고 AWS 액세스 키 ID, 보안 액세스 키 및 세션 토큰으로 구성된 임시 보안 자격 증명 집합을 얻을 수 있습니다. 그러면 앱은 이러한 자격 증명을 사용하여 AWS가 제공하는 웹 서비스에 액세스할 수 있습니다. 앱은 수임하는 역할에 정의된 권한으로 제한됩니다.

다음 그림은 Login with Amazon을 IdP로 사용하는 경우 이것이 어떻게 작동하는지 그 흐름을 단순화해 보여 줍니다. 2단계에서 앱은 Facebook, Google 또는 OIDC 호환 IdP를 사용할 수도 있지만, 여기에서는 생략했습니다.



1. 고객은 모바일 디바이스에서 앱을 시작합니다. 앱은 사용자에게 로그인하도록 요청합니다.
2. 앱은 Login with Amazon 리소스를 사용해 사용자의 자격 증명을 수락합니다.
3. 앱은 Cognito API 작업을 사용해 Login with Amazon ID 토큰을 Cognito 토큰과 교환합니다.
4. 앱은 Cognito 토큰을 전달하면서 AWS STS에서 임시 보안 자격 증명을 요청합니다.
5. 임시 보안 자격 증명은 앱에 의해 사용됨으로써 앱이 작동을 요청하는 어떤 AWS 리소스에도 액세스할 수 있습니다. 임시 보안 자격 증명과 연결된 역할과 그에 할당된 정책은 액세스 가능한 대상을 결정합니다.

다음 절차를 통해 앱이 Amazon Cognito를 사용해 사용자를 인증하도록 구성하고 앱에게 AWS 리소스에 대한 액세스 권한을 부여하십시오. 이 시나리오를 완수하기 위한 특정 단계에 대해서는 Amazon Cognito에 대한 문서 단원을 참조하십시오.

1. (선택 사항) Login with Amazon, Facebook, Google 또는 기타 OpenID Connect(OIDC)-호환 IdP를 통해 개발자로 가입하여 그 공급자를 통해 1개 이상의 앱을 구성합니다. Amazon Cognito는 사용자를 위해 인증되지 않은(게스트) 액세스도 지원하기 때문에 이 단계는 옵션입니다.

2. AWS Management 콘솔의 Amazon Cognito로 이동합니다. Amazon Cognito 마법사를 사용해 자격 증명 풀을 생성합니다. 이 풀은 Amazon Cognito가 앱을 위해 최종 사용자 자격 증명을 정돈된 상태로 유지할 목적으로 사용하는 컨테이너입니다. 앱 간에 자격 증명 풀을 공유할 수 있습니다. 자격 증명 풀을 설정할 때 Amazon Cognito는 Amazon Cognito 사용자에 대한 권한을 정의하는 1개 이상의 IAM 역할(인증된 자격 증명에 대해 1개, 그리고 인증되지 않은 "게스트" 자격 증명을 위해 1개)을 생성합니다.
3. iOS용 AWS SDK 또는 Android용 AWS SDK를 다운로드해 앱과 통합하고 Amazon Cognito를 사용하는데 필요한 파일을 가져옵니다.
4. Amazon Cognito 자격 증명 공급자의 인스턴스를 생성해 자격 증명 풀 ID, AWS 계정 번호 및 자격 증명 풀과 연결한 역할들의 Amazon 리소스 이름(ARN)을 전달합니다. AWS Management 콘솔의 Amazon Cognito 마법사는 샘플 코드를 제공해 시작을 돋습니다.
5. 앱이 AWS 리소스에 액세스할 때 클라이언트 객체에 자격 증명 공급자 인스턴스를 전달합니다. 이렇게 하면 클라이언트에 임시 보안 자격 증명이 전달됩니다. 자격 증명에 대한 권한은 앞서 정의한 역할 또는 역할들에 기반을 두고 있습니다.

자세한 정보는 다음을 참조하십시오.

- [Android용 AWS Mobile SDK Developer Guide의 Amazon Cognito 자격 증명](#)
- [AWS Mobile SDK for iOS Developer Guide의 Amazon Cognito 자격 증명](#)

모바일 앱을 위한 웹 자격 증명 연동 API 작업 사용

최상의 결과를 얻으려면 거의 모든 웹 자격 증명 연동 시나리오에 대해 Amazon Cognito를 자격 증명 브로커로 사용하십시오. Amazon Cognito는 사용하기 쉽고 악명의(인증되지 않은) 액세스, 디바이스 및 공급자 전반에 걸친 사용자 데이터 동기화와 같은 부가적인 기능을 제공합니다. 그러나 AssumeRoleWithWebIdentity API를 수동 호출함으로써 웹 자격 증명 연동을 사용하는 앱을 이미 생성했다면, 그 앱을 계속해서 사용할 수 있고 앱은 여전히 잘 작동될 것입니다.

Note

웹 자격 증명 연동이 어떤 방식으로 작동하는지에 대한 이해를 돋는 [Web Identity Federation Playground](#)를 이용할 수 있습니다. 이 대화형 웹 사이트는 Login with Amazon, Facebook 또는 Google을 통해 인증하고 임시 보안 자격 증명을 얻은 다음, 이러한 자격 증명을 사용하여 AWS에 요청하는 과정을 안내합니다.

Amazon Cognito 없이 웹 자격 증명 연동을 사용하는 과정은 대체로 다음과 같은 개요를 따릅니다.

1. 외부 자격 증명 공급자(IdP)에서 개발자로 로그인하여 앱을 위한 고유 ID를 부여하는 IdP에서 앱을 구성합니다. (서로 다른 공급자는 이 과정에 대해 서로 다른 용어를 사용합니다. 이 개요는 앱을 IdP와 동일시하는 과정에 대해 구성이라는 용어를 사용합니다). 각 IdP는 IdP 고유의 앱 ID를 제공함으로써, 동일한 앱을 다수의 IdP로 구성하는 경우 앱은 여러 개의 앱 ID를 갖게 됩니다. 각 공급자로 여러 개의 앱을 구성할 수 있습니다.

다음 외부 링크는 흔히 사용되는 자격 증명 공급자(IdP) 중 일부를 사용하는 것에 대한 정보를 제공합니다.

- [Login with Amazon 개발자 센터](#)
- [Facebook 개발자 사이트의 앱 또는 웹 사이트에 Facebook 로그인 추가하기](#)
- [Google 개발자 사이트의 OAuth 2.0을 사용한 로그인\(OpenID Connect\)](#)

Note

Amazon Cognito와 Google이 OIDC 기술에 기반을 두고 있다 해도 이러한 공급자를 사용하기 위해 IAM 자격 증명 공급자 엔터티를 생성할 필요는 없습니다. Amazon Cognito와 Google에 대한 지원은 AWS에 내장되어 있습니다.

2. OIDC와 호환되는 IdP를 사용하는 경우 OIDC용 IAM 자격 증명 공급자 엔터티를 생성합니다.
3. IAM에서 [하나 이상의 역할을 생성합니다 \(p. 215\)](#). 각 역할에 대해 그 역할을 위임할 대상(신뢰 정책)과 앱 사용자들이 가져야 할 권한(권한 정책)을 정의할 수 있습니다. 일반적으로 앱이 지원하는 각 IdP마다 하

나의 역할을 생성합니다. 예를 들면 사용자가 Login with Amazon을 통해 로그인할 때 앱이 위임할 수 있는 역할, 사용자가 Facebook을 통해 로그인한 동일 앱에 대한 두 번째 역할 및 사용자가 Google을 통해 로그인하는 앱에 대한 세 번째 역할을 생성할 수 있습니다. 신뢰 관계를 위해서는 IdP(예: Amazon.com)를 Principal(신뢰받는 개체)로 지정하고 앱 ID에 할당된 IdP와 일치하는 Condition을 포함시키십시오. 서로 다른 공급자에 대한 역할의 예는 이 주제의 후반부에 설명되어 있습니다.

- 애플리케이션에서 IdP로 사용자를 인증하십시오. 이렇게 하는 방법에 대한 세부 사항은 사용 중인 IdP(Login with Amazon, Facebook 또는 Google)와 앱이 실행되는 플랫폼에 따라 달라집니다. 예를 들어 Android 앱의 인증 방법은 iOS 앱 또는 JavaScript 기반 웹 앱과 다를 수 있습니다.

일반적으로 사용자가 아직 로그인하지 않은 경우 IdP가 로그인 페이지 표시를 처리합니다. IdP가 사용자를 인증한 후에 IdP는 사용자에 대한 정보가 담긴 인증 토큰을 앱에 반환합니다. 포함된 정보의 내용은 IdP가 노출하는 것과 사용자가 공유하고자 하는 정보가 무엇인지에 달려 있습니다. 앱에서 이 정보를 사용할 수 있습니다.

- 앱에서 AssumeRoleWithWebIdentity 작업을 서명 없이 호출하여 임시 보안 자격 증명을 요청할 수 있습니다. 요청 시 IdP의 인증 토큰을 전달하고 해당 IdP에 대해 생성한 IAM 역할의 Amazon 리소스 이름(ARN)을 지정합니다. AWS는 그 토큰이 신뢰할 수 있고 유효한지 확인하여, 그럴 경우에는 요청 시 이를 지정하는 역할에 대한 권한을 지닌 앱에 임시 보안 자격 증명을 반환합니다. 그 응답에는 IdP가 사용자에게 연결하는 고유 사용자 ID와 같은, IdP에서 오는 사용자에 대한 메타데이터도 포함되어 있습니다.
- AssumeRoleWithWebIdentity 응답의 임시 보안 자격 증명을 사용하여 앱에서 AWS API 작업에 대한 서명된 요청을 생성합니다. IdP에서 받은 사용자 ID 정보는 앱의 사용자를 구별할 수 있습니다. 예를 들어 사용자 ID를 접두사 또는 접미사로 포함하는 Amazon S3 폴더에 객체를 넣을 수 있습니다. 이렇게 함으로써 폴더를 잡그는 액세스 제어 정책을 생성해 그 ID를 지닌 사용자만 그 폴더에 액세스할 수 있게 됩니다. 자세한 정보는 이 주제의 후반부에서 [웹 자격 증명 연동을 사용해 사용자 식별하기 \(p. 165\)](#) 단원을 참조하십시오.
- 앱은 AWS에 요청할 필요가 있을 때마다 새 임시 보안 자격 증명을 받지 않아도 되도록 임시 보안 자격 증명을 캐시해야 합니다. 기본적으로 자격 증명은 1시간 동안 유효합니다. 자격 증명이 만료되면(또는 그 전에) AssumeRoleWithWebIdentity에 또 한 번 호출을 하여 새로운 임시 보안 자격 증명 집합을 얻으십시오. IdP의 토큰 역시 보통 설정된 시간이 지나면 만료되기 때문에, IdP 및 IdP가 토큰을 어떻게 관리하느냐에 따라 AssumeRoleWithWebIdentity에 새로운 호출을 하기 전에 IdP의 토큰을 갱신해야 할 수도 있습니다. iOS를 위한 AWS SDK 또는 Android를 위한 AWS SDK를 사용하는 경우 [AmazonSTSCredentialsProvider](#) 작업을 사용해 IAM 임시 자격 증명을 필요에 따라 갱신하는 등 관리할 수 있습니다.

웹 자격 증명 연동을 사용해 사용자 식별하기

IAM에서 액세스 정책을 생성하는 경우 대체로 외부 자격 증명 공급자(IdP)를 사용하여 인증한 사용자의 ID와 구성된 앱에 기반을 두어 권한을 지정할 수 있는 기능이 유용합니다. 예를 들어 웹 자격 증명 연동을 사용하고 있는 모바일 앱은 다음과 같은 구조를 사용해 Amazon S3에 정보를 저장하고자 할 것입니다.

```
myBucket/app1/user1
myBucket/app1/user2
myBucket/app1/user3
...
myBucket/app2/user1
myBucket/app2/user2
myBucket/app2/user3
...
```

또한, 공급자별로 이 경로를 구별하는 추가 기능을 원할 수도 있습니다. 이 경우에 그 구조는 다음과 같을 것입니다(공간 절약을 위해 2개의 공급자만 나열했습니다).

```
myBucket/Amazon/app1/user1
myBucket/Amazon/app1/user2
myBucket/Amazon/app1/user3
...
myBucket/Amazon/app2/user1
```

```
myBucket/Amazon/app2/user2
myBucket/Amazon/app2/user3

myBucket/Facebook/app1/user1
myBucket/Facebook/app1/user2
myBucket/Facebook/app1/user3
...
myBucket/Facebook/app2/user1
myBucket/Facebook/app2/user2
myBucket/Facebook/app2/user3
...
```

이 구조에서 app1 및 app2는 서로 다른 게임과 같이 서로 다른 앱을 나타내며, 각 앱 사용자는 구분된 폴더를 갖습니다. app1 및 app2에 대한 값은 지정하는 친숙한 이름(예: mynumbersgame)이거나 앱 구성 시 공급자들이 할당하는 앱 ID일 수도 있습니다. 경로에 공급자 이름을 포함하기로 한다면, 그 값은 Cognito, Amazon, Facebook, Google와 같은 친숙한 이름이 될 수도 있습니다.

애플리케이션 이름은 정적 값이므로 일반적으로 AWS Management 콘솔을 통해 app1과 app2에 대한 폴더를 생성할 수 있습니다. 공급자 이름도 정적 값이므로 경로에 공급자 이름을 포함하는 경우에도 그렇게 할 수 있습니다. 이와 대조적으로 사용자 고유 폴더(**user1**, **user2**, **user3** 등)는 AssumeRoleWithWebIdentity에 대한 요청에 의해 반환되는 SubjectFromWebIdentityToken 값에서 얻을 수 있는 사용자 ID를 사용해 앱에서 런타임에 생성되어야 합니다.

개별 사용자에게 리소스에 배타적인 액세스 권한을 허용하는 정책을 작성하려면, 앱 이름과 공급자 이름(사용하는 경우)을 비롯해 완전한 폴더 이름과 일치시킬 수 있습니다. 그런 다음 공급자가 반환하는 사용자 ID를 참조하는 다음 공급자별 콘텍스트 키를 포함할 수 있습니다.

- cognito-identity.amazonaws.com:sub
- www.amazon.com:user_id
- graph.facebook.com:id
- accounts.google.com:sub

OIDC 공급자의 경우 다음 예시와 같이 하위 콘텍스트 키가 있는 OIDC 공급자의 정규화된 URL을 사용합니다.

- **server.example.com**:sub

다음 예는 버킷에 대한 접두사가 문자열과 일치하는 경우에만 Amazon S3 버킷에 액세스 권한을 부여하는 권한 정책을 보여줍니다.

```
myBucket/Amazon/mynumbersgame/user1
```

이 예는 사용자가 Login with Amazon을 사용해 로그인되어 있고 그 사용자는 mynumbersgame이라는 앱을 사용하고 있다고 가정합니다. 사용자의 고유 ID는 user_id라는 속성으로 제시됩니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3>ListBucket"],
      "Resource": ["arn:aws:s3:::myBucket"],
      "Condition": {"StringLike": {"s3:prefix": ["Amazon/mynumbersgame/${www.amazon.com:user_id}/*"]}}
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": ["arn:aws:s3:::myBucket/*"]
    }
  ]
}
```

```
        "s3:PutObject",
        "s3:DeleteObject"
    ],
    "Resource": [
        "arn:aws:s3:::myBucket/amazon/mynumbersgame/${www.amazon.com:user_id}",
        "arn:aws:s3:::myBucket/amazon/mynumbersgame/${www.amazon.com:user_id}/*"
    ]
}
}
```

Amazon Cognito, Facebook, Google 또는 기타 OpenID Connect-호환 IdP를 사용해 로그인하는 사용자를 위해 유사한 정책을 생성할 수도 있습니다. 그 정책은 다른 앱 ID뿐만 아니라 다른 공급자 이름을 경로의 일부로 사용할 것입니다.

정책에서 조건 확인을 위해 사용 가능한 웹 자격 증명 연동 키에 대한 자세한 정보는 [AWS 웹 자격 증명 연동에서 사용할 수 있는 키 \(p. 560\)](#) 단원을 참조하십시오.

웹 자격 증명 연동 관련 추가 리소스

다음 리소스는 웹 자격 증명 연동에 대해 자세히 알아보는 데 도움이 됩니다.

- Android용 AWS Mobile SDK Developer Guide의 [Amazon Cognito 자격 증명](#) 및 AWS Mobile SDK for iOS Developer Guide의 [Amazon Cognito 자격 증명](#)
- [Web Identity Federation Playground](#)는 Login with Amazon, Facebook 또는 Google을 통해 인증하고, 임시 보안 자격 증명을 얻은 다음, 이러한 자격 증명을 사용하여 AWS에 요청하는 과정을 안내하는 대화형 웹 사이트입니다.
- AWS .NET Development 블로그의 [.NET용 AWS SDK를 사용한 웹 자격 증명 연동](#) 항목은 Facebook에서 웹 자격 증명 연동을 사용하는 방법을 안내하며 AssumeRoleWithWebIdentity를 호출하는 방법과 그 API 호출에서 얻은 임시 보안 자격 증명을 사용하여 S3 버킷에 액세스하는 방법을 보여 주는 C# 코드 조각이 포함되어 있습니다.
- [iOS용 AWS SDK](#)와 [Android용 AWS SDK](#)에는 샘플 앱이 포함되어 있습니다. 이러한 앱에는 자격 증명 공급자를 호출하는 방법과 이러한 공급자의 정보를 사용하여 임시 보안 자격 증명을 가져오고 사용하는 방법을 보여주는 코드가 포함되어 있습니다.
- [모바일 애플리케이션을 사용한 웹 자격 증명 연동](#) 항목에서는 웹 자격 증명 연동에 대해 설명하며 웹 자격 증명 연동을 사용하여 Amazon S3 컨텐츠에 액세스하는 방법의 예를 보여 줍니다.

SAML 2.0 기반 연동에 대하여

AWS는 많은 자격 증명 공급자(IdP)가 사용하는 개방형 표준인 [SAML 2.0\(Security Assertion Markup Language 2.0\)](#)이라는 자격 증명 연동을 지원합니다. 이 기능은 연동 SSO(Single Sign-On)를 활성화하여 조직의 모든 이에 대해 IAM 사용자를 생성하지 않고도 사용자가 AWS Management 콘솔에 로그인하거나 AWS API 작업을 호출할 수 있습니다. SAML을 사용함으로써 AWS로 연동을 구성하는 과정을 단순화할 수 있는데, 이는 [사용자 지정 자격 증명 프록시 코드](#)를 작성하는 대신 IdP의 서비스를 사용할 수 있기 때문입니다.

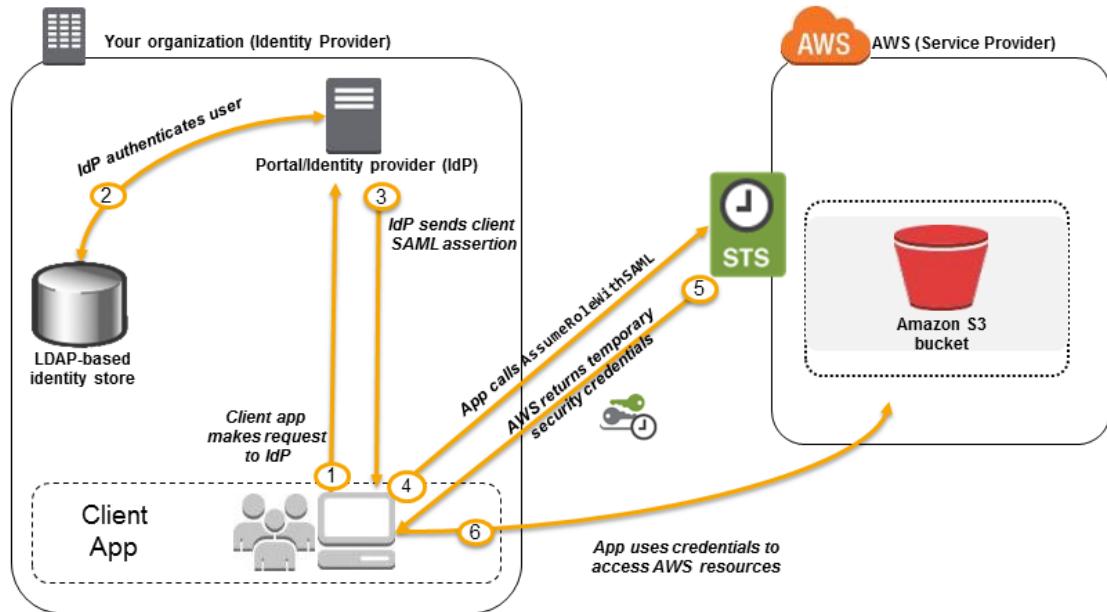
IAM 연동은 다음과 같은 사용 사례를 지원합니다.

- [조직의 사용자 또는 애플리케이션이 AWS API 작업을 호출할 수 있도록 허용하는 연동된 액세스 \(p. 168\)](#). 조직에서 생성되는 SAML 어설션(인증 응답의 일부)을 사용해 임시 보안 자격 증명을 얻습니다. 이 시나리오는 [임시 보안 자격 증명 요청하기 \(p. 265\)](#) 및 [웹 자격 증명 연동에 대하여 \(p. 162\)](#)에 기술된 것과 같이 IAM이 지원하는 다른 연동 시나리오들과 유사합니다. 그러나 조직의 SAML 2.0-기반 IdP는 인증 수행 및 권한 부여 확인을 위한 런타임에 많은 세부 정보를 처리합니다. 이 주제에서는 이러한 시나리오에 대해 설명합니다.
- [조직에서 AWS Management 콘솔로 이루어지는 웹 기반 SSO\(Single Sign-On\) \(p. 185\)](#). SAML 2.0 호환 IdP에서 호스팅하는 조직 내 포털에 사용자가 로그인한 다음 옵션을 선택하여 AWS로 이동하면, 별도의

로그인 정보를 제공하지 않고도 콘솔로 리디렉션됩니다. 타사 SAML IdP를 사용하여 콘솔에 SSO 액세스하거나 사용자 지정 IdP를 만들어 외부 사용자의 콘솔 액세스를 허용할 수 있습니다. 사용자 지정 IdP를 구축하는 방법에 대한 자세한 정보는 [연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하는 URL 생성\(사용자 지정 연동 브로커\) \(p. 188\)](#)를 참조하십시오.

SAML 기반 연동을 이용하여 AWS에 API 액세스

직원들에게 자신의 컴퓨터에서 백업 폴더로 데이터를 복사하는 방법을 제공하려 한다고 가정해 봅시다. 사용자가 컴퓨터에서 실행하는 애플리케이션을 구축합니다. 그 애플리케이션은 백엔드에서 S3 버킷에 있는 객체를 읽고 쓸니다. 사용자는 AWS에 직접 액세스할 수 없습니다. 그 대신 다음 프로세스를 사용합니다.



1. 조직 내 사용자가 클라이언트 앱을 사용해 조직의 IdP로부터 인증을 요청합니다.
2. IdP가 조직의 자격 증명 스토어를 이용하여 사용자를 인증합니다.
3. IdP가 사용자에 대한 정보로 SAML 어설션을 만들어 클라이언트 앱으로 보냅니다.
4. 클라이언트 앱이 AWS STS `AssumeRoleWithSAML` API를 호출하면서 SAML 공급자의 ARN, 수임할 역할의 ARN, IdP로부터 받은 SAML 어설션을 전달합니다.
5. 클라이언트 앱에 대한 API 응답에는 임시 보안 자격 증명이 포함되어 있습니다.
6. 클라이언트 앱은 임시 보안 자격 증명을 사용해 Amazon S3 API 작업을 호출합니다.

SAML 2.0 기반 연동에 대한 개요

앞의 시나리오와 다이어그램을 통해 설명한 대로 SAML 2.0 기반 연동을 사용하기 전에, 서로를 신뢰하도록 조직의 IdP와 AWS 계정을 구성해야 합니다. 이 신뢰를 구성하는 일반적인 프로세스는 다음 단계에서 설명합니다. 조직 내에는 Microsoft Active Directory 연동 서비스(AD FS, Windows Server의 일부), Shibboleth 또는 기타 호환 가능한 SAML 2.0 공급자와 같이 [SAML 2.0을 지원하는 IdP \(p. 180\)](#)가 반드시 있어야 합니다.

조직의 IdP와 AWS가 서로 신뢰하도록 구성하는 방법

1. IdP로 AWS를 등록하는 것으로 시작합니다. 조직의 IdP에서 다음 URL에서 얻는 SAML 메타데이터 문서를 사용함으로써 AWS를 서비스 공급자(SP)로 등록합니다.

<https://signin.aws.amazon.com/static/saml-metadata.xml>

2. 조직의 IdP를 사용해 AWS에서 IdP를 IAM 자격 증명 공급자로 기술하는 동등한 메타데이터 XML 파일을 생성합니다. 그 파일에는 발급자 이름, 생성 일자, 만료 일자 및 AWS가 조직에서 오는 인증 응답의 유효성을 검증하는데 사용할 수 있는 키가 포함되어 있어야 합니다.
3. IAM 콘솔에서 SAML 자격 증명 공급자 엔터티를 생성합니다. 이 과정의 일부로 Step 2에서 조직의 IdP가 생성한 SAML 메타데이터 문서를 업로드합니다. 자세한 정보는 [IAM SAML 자격 증명 공급자 생성 \(p. 177\)](#)을 참조하십시오.
4. IAM에서 하나 이상의 IAM 역할을 생성합니다. 역할의 신뢰 정책에서 SAML 공급자를 보안 주체로 설정함으로써 조직과 AWS 사이에 신뢰 관계를 설정합니다. 역할의 권한 정책은 조직의 사용자가 AWS에서 하도록 허용된 것을 설정합니다. 자세한 정보는 [타사 자격 증명 공급자의 역할 만들기\(연동\) \(p. 215\)](#) 단원을 참조하십시오.
5. 조직의 IdP에서 조직 내 사용자 또는 그룹을 IAM 역할로 매핑하는 어설션을 정의합니다. 조직의 다양한 사용자 및 그룹은 서로 다른 IAM 역할에 매핑될 수 있다는 것에 유의하십시오. 매핑 수행을 위한 정확한 절차는 사용하고 있는 IdP에 따라 다릅니다. 사용자를 위한 Amazon S3 폴더의 [조기 시나리오 \(p. 168\)](#)에서는 모든 사용자들이 Amazon S3 권한을 제공하는 동일한 역할에 매핑되는 것이 가능합니다. 자세한 정보는 [인증 응답을 위한 SAML 어설션 구성 \(p. 181\)](#) 단원을 참조하십시오.

IdP가 AWS 콘솔에 대한 SSO를 지원하는 경우, 콘솔 세션의 최대 지속 기간을 구성할 수 있습니다. 자세한 정보는 [SAML 2.0 연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하기 \(p. 185\)](#) 단원을 참조하십시오.

Note

SAML 2.0 연동의 AWS 구현은 IAM 공급자와 AWS 간에 암호화된 SAML 어설션을 지원하지 않습니다. 하지만 고객의 시스템과 AWS 간의 트래픽은 암호화된(TLS) 채널을 통해 전송됩니다.

6. 생성 중인 애플리케이션에서 AWS Security Token Service `AssumeRoleWithSAML` API를 호출해 그것을 Step 3 단계에서 생성한 SAML 공급자의 ARN, Step 4 단계에서 생성한 수임할 역할의 ARN 및 IdP에서 얻는 현재 사용자에 대한 SAML 어설션으로 전달합니다. AWS는 역할 수임 요청이 SAML 공급자에서 참조된 IdP로부터 오는지 확인합니다.

자세한 정보는 AWS Security Token Service API 참조의 [AssumeRoleWithSAML](#)을 참조하십시오.

7. 요청이 성공하면 API는 일련의 임시 보안 자격 증명을 반환하고 애플리케이션은 이를 사용해 AWS에서 명된 요청을 보냅니다. 애플리케이션은 현재 사용자에 대한 정보를 갖고 있어서 이전 시나리오에 기술된 대로 Amazon S3의 사용자별 폴더에 액세스할 수 있습니다.

AWS 리소스에 대한 SAML 연동 액세스를 허용하는 역할에 대한 개요

IAM에서 생성하는 역할 또는 역할들은 조직의 연동 사용자가 AWS에서 하도록 허용되는 것이 무엇인지 정의합니다. 역할에 대한 신뢰 정책을 생성할 때 앞서 생성한 SAML 공급자를 Principal로 지정합니다. Condition으로 신뢰 정책을 추가로 자세히 살펴봄으로써 특정 SAML 속성과 일치하는 사용자만 그 역할에 액세스하도록 허용할 수 있습니다. 예를 들어 다음 샘플 정책에 설명되어 있듯이 SAML 소속이 staff(<https://openidp.feide.no>에 의해 어설션되듯이)인 사용자만이 그 역할에 액세스할 수 있도록 지정할 수 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Principal": {"Federated": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:saml-provider/ExampleOrgSSOProvider"},  
         "Action": "sts:AssumeRoleWithSAML",  
         "Condition": {  
             "StringEquals": {  
                 "saml:aud": "https://signin.aws.amazon.com/saml",  
                 "saml:iss": "https://openidp.feide.no"  
             },  
             "ForAllValues:StringLike": {"saml:edupersonaffiliation": ["staff"]}  
         }  
    ]  
}
```

```
    } ]  
}
```

정책에서 확인할 수 있는 SAML 키에 대한 자세한 정보는 [SAML 기반 AWS STS 연동에 사용할 수 있는 키 \(p. 561\)](#) 단원을 참조하십시오.

해당 역할의 권한 정책에 대해서는, 역할에 사용하는 방식으로 권한을 지정합니다. 예를 들어 조직의 사용자가 Amazon Elastic Compute Cloud 인스턴스를 관리하도록 허용된다면 AmazonEC2FullAccess 관리형 정책의 작업과 같은 권한 정책의 Amazon EC2 작업을 명시적으로 허용해야 합니다.

SAML 기반 연동에서 사용자를 고유하게 식별하기

IAM에서 액세스 정책을 생성할 때 사용자의 자격 증명에 기반을 두어 권한을 지정할 수 있다는 것은 종종 쓸모가 있습니다. 예를 들어 SAML을 사용해 연동된 사용자들에 대해, 애플리케이션은 다음과 같은 구조를 사용해 Amazon S3에 정보를 저장하고자 할 것입니다.

```
myBucket/app1/user1  
myBucket/app1/user2  
myBucket/app1/user3
```

버킷과 폴더는 정적 값이므로 Amazon S3 콘솔 또는 AWS CLI를 통해 버킷(myBucket)과 폴더(app1)를 생성할 수 있습니다. 그러나 사용자 고유 폴더(**user1**, **user2**, **user3** 등)는 사용자가 연동 프로세스를 통해 최초로 로그인할 때까지 사용자를 식별하는 값이 알려지지 않기 때문에 코드를 사용해 런타임에 생성되어야 합니다.

사용자 고유의 세부 정보를 리소스 이름의 일부로 참조하는 정책을 작성하려면, 정책 조건에서 사용될 수 있는 SAML 키에서 사용자 자격 증명이 사용 가능해야 합니다. 다음 키는 IAM 정책용 SAML 2.0 기반 연동에 대해 사용 가능합니다. 다음 키가 반환하는 값들을 사용해 Amazon S3 폴더와 같은 리소스에 대한 고유의 사용자 식별자를 생성할 수 있습니다.

- saml:namequalifier. Issuer 반응 값(saml:iss)과 AWS 계정 ID 및 IAM의 SAML 공급자 표시 이름(ARN의 마지막 부분)으로 된 문자열의 연속값에 기반을 둔 해시 값 계정 ID, SAML 공급자 표시 이름의 연속값은 IAM 정책에서 키 saml:doc으로 사용 가능합니다. 계정 ID와 공급자 이름은 "123456789012/provider_name"처럼 '/'로 구분되어야 합니다. 자세한 정보는 saml:doc의 [SAML 기반 AWS STS 연동에 사용할 수 있는 키 \(p. 561\)](#) 키를 참조하십시오.

NameQualifier와 Subject의 조합은 연동 사용자를 고유한 이름으로 식별하는데 사용할 수 있습니다. 다음 유사 코드는 이 값이 계산되는 방식을 보여줍니다. 이 유사 코드에서 +는 연결을 나타내고, SHA1은 SHA-1을 사용해 메시지 디자이스트를 생성하는 기능을 나타내며, Base64는 해시 출력의 Base-64 인코딩 버전을 생성하는 기능을 나타냅니다.

```
Base64 ( SHA1 ( "https://example.com/saml" + "123456789012" + "/"  
MySAMLIdP" ) )
```

SAML 기반 연동에 사용 가능한 정책 키에 대한 자세한 정보는 다음([SAML 기반 AWS STS 연동에 사용할 수 있는 키 \(p. 561\)](#))을 참조하십시오.

- saml:sub (문자열). 이것은 클레임의 주체로서 여기에는 조직 내 사용자 개개인을 식별할 수 있는 고유 값이 포함됩니다(예: _cbb88bf52c2510eabe00c1642d4643f41430fe25e3).
- saml:sub_type (문자열). 이 키는 persistent, transient, 또는 SAML 어설션에서 사용되는 Format 및 Subject 요소의 전체 NameID URI일 수 있습니다. persistent라는 값은 saml:sub의 값이 모든 세션에 걸쳐 사용자에게 동일하다는 것을 나타냅니다. 값이 transient인 경우 각 세션마다 사용자의 saml:sub 값이 다릅니다. NameID 요소의 Format 속성에 대한 자세한 정보는 [인증 응답을 위한 SAML 어설션 구성 \(p. 181\)](#) 단원을 참조하십시오.

다음 예는 선행 키를 사용하여 Amazon S3의 사용자 고유 폴더에 대한 권한을 부여하는 권한 정책을 보여줍니다. 그 정책은 saml:namequalifier 및 saml:sub를 둘 다 포함하는 접두사를 사용해 Amazon S3 객체를 식별하는 것으로 가정합니다. Condition 요소에는 saml:sub_type이 persistent로 설정되어 있는

지 확인하는 테스트가 포함되어 있다는 것에 유의하십시오. `transient`로 설정되어 있다면 사용자에 대한 `saml:sub` 값은 각 세션마다 다를 수 있고 값의 조합은 사용자 고유 폴더를 식별하는 데 사용되어서는 안 됩니다.

```
>{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::exampleorgBucket/backup/${saml:namequalifier}/${saml:sub}",
        "arn:aws:s3:::exampleorgBucket/backup/${saml:namequalifier}/${saml:sub}/*"
      ],
      "Condition": {"StringEquals": {"saml:sub_type": "persistent"}}
    }
}
```

IdP의 어설션을 정책 키에 매핑하는 것에 대한 자세한 정보는 [인증 응답을 위한 SAML 어설션 구성 \(p. 181\)](#) 단원을 참조하십시오.

IAM 자격 증명 공급자 생성

외부 자격 증명 공급자(IdP) 서비스와의 연동을 구성하려는 경우 IAM 자격 증명 공급자를 생성하여 IdP 및 구성에 대해 AWS에 알려줍니다. 이렇게 하면 AWS 계정과 IdP 사이에 "신뢰"가 설정됩니다. 다음 주제는 각 IdP 유형별로 IAM 자격 증명 공급자를 생성하는 방법에 대해 자세히 설명합니다.

주제

- [OpenID Connect\(OIDC\) 자격 증명 공급자의 생성 \(p. 171\)](#)
- [IAM SAML 자격 증명 공급자 생성 \(p. 177\)](#)

OpenID Connect(OIDC) 자격 증명 공급자의 생성

IAMOIDC 자격 증명 공급자는 IAM의 엔터티로서 Google이나 Salesforce와 같은 [OpenID Connect\(OIDC\) 표준](#)을 지원하는 자격 증명 공급자(IdP) 서비스를 기술합니다. IAM OIDC 자격 증명 공급자는 OIDC 호환 IdP와 AWS 계정 간에 신뢰를 구축하려 할 때 사용합니다. 예를 들어 AWS 리소스에 액세스하는데 필요한 모바일 앱이나 웹 애플리케이션을 개발하면서 사용자 지정 로그인 코드를 생성하거나 자신의 사용자 자격 증명을 관리하지 않을 때 유용합니다. 이 시나리오에 대한 자세한 정보는 [the section called “웹 자격 증명 연동에 대하여” \(p. 162\)](#)를 참조하십시오.

AWS Management 콘솔, AWS Command Line Interface, Windows PowerShell용 도구 또는 IAM API를 사용하여 IAM OIDC 자격 증명 공급자를 생성 및 관리할 수 있습니다.

주제

- [OIDC 공급자의 생성 및 관리\(콘솔\) \(p. 171\)](#)
- [IAM OIDC 자격 증명 공급자 생성 및 관리\(AWS CLI\) \(p. 172\)](#)
- [OIDC 자격 증명 공급자 만들기 및 관리\(AWS API\) \(p. 173\)](#)
- [OpenID Connect 자격 증명 공급자의 지문 얻기 \(p. 174\)](#)

OIDC 공급자의 생성 및 관리(콘솔)

이 지침에 따라 AWS Management 콘솔에서 IAM OIDC 자격 증명 공급자를 생성 및 관리하십시오.

IAM OIDC 자격 증명 공급자를 생성하는 방법(콘솔)

1. IAM OIDC 자격 증명 공급자를 생성하려면 먼저 애플리케이션을 IdP에 등록하여 클라이언트 ID를 받아야 합니다. 클라이언트 ID(사용자라고도 불림)는 앱을 IdP에 등록할 때 발급되는 고유의 앱 식별자입니다. 클라이언트 ID를 얻는 방법에 대한 자세한 정보는 해당 IdP에 대한 설명서를 참조하십시오.
2. <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
3. 탐색 창에서 자격 증명 공급자를 선택한 다음 공급자 생성을 선택합니다.
4. 공급자 유형에서 공급자 유형 선택을 선택한 다음 OpenID Connect를 선택합니다.
5. Provider URL(공급자 URL)에서 IdP의 URL을 입력합니다. URL은 다음과 같은 제한을 준수해야 합니다.
 - URL은 대/소문자를 구분합니다.
 - URL은 **https://**로 시작해야 합니다.
 - URL은 콜론(:) 문자를 포함할 수 없으므로 포트 번호를 지정할 수 없습니다. 서버가 기본 포트인 443에서 수신 대기해야 함을 의미합니다.
 - IAM OIDC 자격 증명 공급자는 AWS 계정 내에서 고유한 URL을 사용해야 합니다.
6. Audience(대상) 필드에 IdP를 등록하고 Step 1에서 받은 애플리케이션의 클라이언트 ID를 입력하면 AWS에게도 요청됩니다. IdP에 등록한 클라이언트 ID(사용자들이라고도 불림)가 더 있는 경우 나중에 공급자 세부 정보 페이지에서 추가할 수 있습니다. [Next Step]을 선택합니다.
7. Thumbprint(지문)을 사용하여 IdP의 서버 인증서를 확인합니다. 자세한 방법은 [OpenID Connect 자격 증명 공급자의 지문 얻기 \(p. 174\)](#) 단원을 참조하십시오. Create를 선택합니다.
8. 화면 상단에 확인 메시지가 나오면 지금 수행합니다를 클릭하여 역할 탭으로 이동한 후 이 자격 증명 공급자에 사용할 역할을 생성합니다. OIDC 자격 증명 공급자에 사용할 역할 생성에 대한 자세한 정보는 [타사 자격 증명 공급자의 역할 만들기\(연동\) \(p. 215\)](#) 단원을 참조하십시오. OIDC 자격 증명 공급자가 AWS 계정에 액세스하려면 역할이 필요합니다. 이 단계를 건너뛰고 나중에 역할을 생성하려면 닫기를 선택합니다.

IAM OIDC 자격 증명 공급자에 사용할 지문이나 클라이언트 ID(사용자라고도 함)를 추가 또는 삭제하는 방법(콘솔)

1. <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 자격 증명 공급자를 선택하고 나서 업데이트 할 IAM 자격 증명 공급자의 이름을 선택합니다.
3. 지문이나 사용자를 추가하려면 지문 추가 또는 Add an Audience(사용자 추가)를 선택합니다. 지문이나 사용자를 제거하려면 삭제할 항목 옆에 있는 제거를 선택합니다.

Note

IAM OIDC 자격 증명 공급자마다 한 개 이상의 지문이 있어야 하며 최대 5개까지 가능합니다. OIDC 자격 증명 공급자마다 한 명 이상의 사용자가 있어야 하며 최대 100명까지 가능합니다.

작업을 마쳤으면 변경 사항 저장을 선택합니다.

IAM OIDC 자격 증명 공급자를 삭제하는 방법(콘솔)

1. <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 자격 증명 공급자를 선택합니다.
3. 삭제할 IAM 자격 증명 공급자 옆의 확인란을 선택합니다.
4. 공급자 삭제를 선택합니다.

IAM OIDC 자격 증명 공급자 생성 및 관리(AWS CLI)

다음 AWS CLI 명령을 사용하여 IAM OIDC 자격 증명 공급자를 생성하고 관리할 수 있습니다.

IAM OIDC 자격 증명 공급자를 생성하는 방법(AWS CLI)

1. (선택 사항) AWS 계정의 전체 IAM OIDC 자격 증명 공급자 목록을 가져오려면 다음 명령을 실행합니다.
 - `aws iam list-open-id-connect-providers`
2. 새 IAM OIDC 자격 증명 공급자를 만들려면 다음 명령을 실행합니다.
 - `aws iam create-open-id-connect-provider`

기존 IAMOIDC 자격 증명 공급자의 서버 인증서 지문 목록을 업데이트하는 방법(AWS CLI)

- IAM OIDC 자격 증명 공급자의 서버 인증서 지문 목록을 업데이트하려면 다음 명령을 실행합니다.
 - `aws iam update-open-id-connect-provider-thumbprint`

기존 IAM OIDC 자격 증명 공급자에서 클라이언트 ID를 추가하거나 제거하는 방법(AWS CLI)

1. (선택 사항) AWS 계정의 전체 IAM OIDC 자격 증명 공급자 목록을 가져오려면 다음 명령을 실행합니다.
 - `aws iam list-open-id-connect-providers`
2. (선택 사항) IAM OIDC 자격 증명 공급자에 대한 자세한 정보를 보려면 다음 명령을 실행합니다.
 - `aws iam get-open-id-connect-provider`
3. 기존 IAM OIDC 자격 증명 공급자에 새로운 클라이언트 ID를 추가하려면 다음 명령을 실행합니다.
 - `aws iam add-client-id-to-open-id-connect-provider`
4. 기존 IAM OIDC 자격 증명 공급자에서 클라이언트를 제거하려면 다음 명령을 실행합니다.
 - `aws iam remove-client-id-from-open-id-connect-provider`

IAM OIDC 자격 증명 공급자를 삭제하는 방법(AWS CLI)

1. (선택 사항) AWS 계정의 전체 IAM OIDC 자격 증명 공급자 목록을 가져오려면 다음 명령을 실행합니다.
 - `aws iam list-open-id-connect-providers`
2. (선택 사항) IAM OIDC 자격 증명 공급자에 대한 자세한 정보를 보려면 다음 명령을 실행합니다.
 - `aws iam get-open-id-connect-provider`
3. IAM OIDC 자격 증명 공급자를 삭제하려면 다음 명령을 실행합니다.
 - `aws iam delete-open-id-connect-provider`

OIDC 자격 증명 공급자 만들기 및 관리(AWS API)

다음 IAM API 명령을 사용하여 OIDC 공급자를 만들고 관리할 수 있습니다.

IAM OIDC 자격 증명 공급자를 생성하는 방법(AWS API)

1. (선택 사항) AWS 계정의 전체 IAM OIDC 자격 증명 공급자 목록을 가져오려면 다음 작업을 호출합니다.
 - `ListOpenIDConnectProviders`
2. 새로운 IAM OIDC 자격 증명 공급자를 생성하려면 다음 작업을 호출합니다.
 - `CreateOpenIDConnectProvider`

기존 IAMOIDC 자격 증명 공급자의 서버 인증서 지문 목록을 업데이트하는 방법(AWS API)

- IAM OIDC 자격 증명 공급자의 서버 인증서 지문 목록을 업데이트하려면 다음 작업을 호출합니다.
 - [UpdateOpenIDConnectProviderThumbprint](#)

기존 IAM OIDC 자격 증명 공급자에서 클라이언트 ID를 추가하거나 제거하는 방법(AWS API)

1. (선택 사항) AWS 계정의 전체 IAM OIDC 자격 증명 공급자 목록을 가져오려면 다음 작업을 호출합니다.
 - [ListOpenIDConnectProviders](#)
2. (선택 사항) IAM OIDC 자격 증명 공급자에 대한 자세한 정보를 보려면 다음 작업을 호출합니다.
 - [GetOpenIDConnectProvider](#)
3. 기존 IAM OIDC 자격 증명 공급자에 새로운 클라이언트 ID를 추가하려면 다음 작업을 호출합니다.
 - [AddClientIDToOpenIDConnectProvider](#)
4. IAM OIDC 자격 증명 공급자에서 클라이언트 ID를 제거하려면 다음 작업을 호출합니다.
 - [RemoveClientIDFromOpenIDConnectProvider](#)

IAM OIDC 자격 증명 공급자를 삭제하는 방법(AWS API)

1. (선택 사항) AWS 계정의 전체 IAM OIDC 자격 증명 공급자 목록을 가져오려면 다음 작업을 호출합니다.
 - [ListOpenIDConnectProviders](#)
2. (선택 사항) IAM OIDC 자격 증명 공급자에 대한 자세한 정보를 보려면 다음 작업을 호출합니다.
 - [GetOpenIDConnectProvider](#)
3. IAM OIDC 자격 증명 공급자를 삭제하려면 다음 작업을 호출합니다.
 - [DeleteOpenIDConnectProvider](#)

OpenID Connect 자격 증명 공급자의 지문 얻기

IAM에서 [OpenID Connect\(OIDC\) 자격 증명 공급자 생성 \(p. 171\)](#) 시 외부 자격 증명 공급자(IdP)의 지문을 제공해야 합니다. 지문은 OIDC 호환 IdP가 사용하는 고유한 서버 인증서에 대한 서명입니다. IAM OIDC 자격 증명 공급자를 만들 때는 해당 IdP에 의해 인증된 자격 증명에 본인의 AWS 계정에 대한 액세스 권한을 맡깁니다. OIDC IdP의 지문을 제공하면 특정 OIDC IdP에게 이 액세스 권한을 맡기려는 의도가 AWS에 전달됩니다.

[AWS Command Line Interface](#), [Windows PowerShell용 도구](#) 또는 [IAM API \(p. 172\)](#)를 사용하여 IAM OIDC 자격 증명 공급자를 생성할 수 있습니다. 이러한 방법을 사용하는 경우 수동으로 지문을 얻어서 AWS에 제공해야 합니다. [IAM 콘솔 \(p. 171\)](#)을 사용해 OIDC 자격 증명 공급자를 만들 때 콘솔은 지문을 자동으로 가져오려고 합니다. 또한, 수동으로 OIDC IdP의 지문을 얻어 콘솔에서 올바른 지문을 가져왔는지 확인하는 것이 좋습니다.

웹 브라우저와 OpenSSL 명령줄 도구를 사용하여 OIDC 공급자의 지문을 얻습니다. 자세한 정보는 다음을 참조하십시오.

OIDC IdP의 지문을 얻으려면

1. OIDC IdP의 지문을 얻으려면, 먼저 OpenSSL 명령줄 도구를 얻어야 합니다. 이 도구를 사용하여 OIDC IdP의 인증서 체인을 다운로드하고 인증서 체인에 있는 마지막 인증서의 지문을 생성합니다. OpenSSL을 설치 및 구성해야 하는 경우 [OpenSSL 설치 \(p. 176\)](#) 및 [OpenSSL 구성 \(p. 176\)](#)의 지침을 따르십시오.

- OIDC IdP의 URL(예: <https://server.example.com>)로 시작한 다음 /.well-known/openid-configuration을 추가하여 다음과 같이 OIDC IdP의 구성 문서에 대한 URL을 만듭니다.

<https://server.example.com/.well-known/openid-configuration>

웹 브라우저에서 이 URL을 열 때 server.example.com을 OIDC IdP 서버 이름으로 바꾸어 엽니다.

- 웹 브라우저에 표시되는 문서에서 "jwks_uri"를 찾습니다. 웹 브라우저의 찾기 기능을 사용하여 페이지에서 이 텍스트를 찾을 수 있습니다. "jwks_uri"라는 텍스트 바로 뒤에 콜론(:)과 URL이 보일 것입니다. 그 URL의 정규화된 도메인 이름을 복사합니다. https:// 또는 최상위 도메인 다음에 오는 경로는 포함하지 마십시오.
- OpenSSL 명령줄 도구를 사용하여 다음 명령을 실행합니다. 이때 keys.example.com을 Step 3에서 얻은 도메인 이름으로 바꿉니다.

```
openssl s_client -servername keys.example.com -showcerts -connect keys.example.com:443
```

- 명령 창에서 다음 예제와 비슷한 인증서가 보일 때까지 위로 스크롤합니다. 인증서가 2개 이상 있을 경우 명령 출력의 하단에서 표시된 마지막 인증서를 찾습니다.

```
-----BEGIN CERTIFICATE-----  
MIICItCCAfICCQD6m70Rw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC  
VVmxCzAJBgNVBAgTA1dBMRAwDgYDVQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6  
b24xFDASBgNVBAsTC01BTSDb25zb2x1MRIwEAYDVQQDEwlUZXN0Q21sYWMxHzAd  
BggkhkiG9w0BCQEWEg5vb251QGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN  
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTA1dBMRAwDgYD  
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC01BTSDb25z  
b2x1MRIwEAYDVQQDEwlUZXN0Q21sYWMxHzAdBggkhkiG9w0BCQEWEg5vb251QGFt  
YXpvbi5jb20wgZ28wDQYJKoZIhvcNAQEBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ  
21uUSfwfEvYSwTc2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T  
rDHuDZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE  
Ibb3OhjZnzcvQAArHhd1QWIIm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4  
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb  
FFBjvSFpJ1lJ00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp378OD8uTs7fLvjx79LjSTb  
NYiytBzPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE  
-----END CERTIFICATE-----
```

인증서를 복사해(-----BEGIN CERTIFICATE----- 및 -----END CERTIFICATE----- 줄 포함) 텍스트 파일에 붙여 넣습니다. 그 다음에 그 파일을 **certificate.crt**라는 이름으로 저장합니다.

- OpenSSL 명령줄 도구를 사용하여 다음 명령을 실행합니다.

```
openssl x509 -in certificate.crt -fingerprint -noout
```

다음 예제와 비슷한 인증서 지문이 명령 창에 표시됩니다.

```
SHA1 Fingerprint=99:0F:41:93:97:2F:2B:EC:F1:2D:DE:DA:52:37:F9:C9:52:F2:0D:9E
```

이 문자열에서 콜론(:)를 제거하여 다음과 같은 최종 지문을 생성합니다.

```
990F4193972F2BECF12DDEDA5237F9C952F20D9E
```

- AWS CLI, Windows PowerShell용 도구 또는 IAM API를 사용하여 IAM OIDC 자격 증명 공급자를 만드는 경우 공급자를 만들 때 이 지문을 제공합니다.

IAM 콘솔에서 IAM OIDC 자격 증명 공급자를 만드는 경우에는 OIDC 공급자를 만들 때 콘솔에서 공급자 정보 확인 페이지에 표시되는 지문과 이 지문을 비교합니다.

Important

얻은 지문이 콘솔에 표시되는 지문과 일치하지 않을 경우 콘솔에서 OIDC 공급자를 만들어서는 안 됩니다. 대신 잠시 기다렸다가 공급자를 만들기 전에 지문이 일치하는지 확인하며 다시 OIDC 공급자를 만들어 보십시오. 두 번째 시도 후에도 지문이 여전히 일치하지 않을 경우에는 [IAM 포럼](#)을 통해 AWS에 문의하십시오.

OpenSSL 설치

아직 OpenSSL을 설치하지 않았다면 이 단원에 나오는 지침을 따르십시오.

Linux 또는 Unix에서 OpenSSL을 설치하려면

1. [OpenSSL: Source, Tarballs](https://openssl.org/source/)(<https://openssl.org/source/>)로 이동합니다.
2. 최신 소스를 다운로드하여 패키지를 생성합니다.

Windows에서 OpenSSL을 설치하려면

1. Windows 버전을 설치할 수 있는 사이트 목록을 보려면 [OpenSSL: Binary Distributions](https://wiki.openssl.org/index.php/Binaries)(<https://wiki.openssl.org/index.php/Binaries>)로 이동합니다.
2. 선택한 사이트의 지침을 따라 설치를 시작합니다.
3. [Microsoft Visual C++ 2008 재배포 가능 패키지] 설치를 묻는 메시지가 표시되고 아직 시스템에 설치되지 않았다면 환경에 적합한 다운로드 링크를 선택합니다. [Microsoft Visual C++ 2008 Redistributable Setup Wizard]의 지시를 따릅니다.

Note

시스템에 Microsoft Visual C++ 2008 Redistributables가 설치되어 있는지 알 수 없는 경우 OpenSSL을 먼저 설치합니다. Microsoft Visual C++ 2008 Redistributables가 설치되지 않은 경우에는 OpenSSL 설치 관리자에 알림이 표시됩니다. 설치할 OpenSSL 버전에 해당하는 아키텍처(32비트 또는 64비트)를 설치해야 합니다.

4. Microsoft Visual C++ 2008 Redistributables를 설치한 후에는 환경에 맞는 OpenSSL 바이너리를 선택하고 파일을 로컬 위치에 저장합니다. [OpenSSL Setup Wizard]를 시작합니다.
5. [OpenSSL Setup Wizard] 지시에 따릅니다.

OpenSSL 구성

OpenSSL 명령을 사용하려면 OpenSSL이 설치된 위치 정보가 담기도록 운영 체제를 구성해야 합니다.

Linux 또는 Unix에서 OpenSSL을 구성하려면

1. 명령줄에서 `OPENSSL_HOME` 변수를 OpenSSL 설치 위치로 설정합니다.

```
$ export OPENSSL_HOME=path_to_your_OpenSSL_installation
```

2. OpenSSL 설치가 포함되도록 경로를 설정합니다.

```
$ export PATH=$PATH:$OPENSSL_HOME/bin
```

Note

`export` 명령을 사용하여 변경한 환경 변수는 현재 세션에만 유효합니다. 쉘 구성 파일에서 설정하면 환경 변수의 영구 변경이 가능합니다. 자세한 내용은 운영 체제 설명서를 참조하십시오.

Windows에서 OpenSSL을 구성하려면

1. [Command Prompt] 창을 엽니다.
2. OpenSSL_HOME 변수를 OpenSSL 설치 위치로 설정합니다.

```
C:\> set OpenSSL_HOME=path_to_your_OpenSSL_installation
```

3. OpenSSL_CONF 변수를 OpenSSL 설치에 있는 구성 파일 위치로 설정합니다.

```
C:\> set OpenSSL_CONF=path_to_your_OpenSSL_installation\bin\openssl.cfg
```

4. OpenSSL 설치가 포함되도록 경로를 설정합니다.

```
C:\> set Path=%Path%;%OpenSSL_HOME%\bin
```

Note

[Command Prompt] 창에서 변경한 Windows 환경 변수는 현재 명령줄 세션에만 유효합니다. 환경 변수를 시스템 속성으로 설정하면 환경 변수의 영구 변경이 가능합니다. 정확한 절차는 사용 중인 Windows 버전에 따라 달라집니다. (예를 들어, Windows 7에서는 [Control Panel], [System and Security], [System]을 엽니다. 그 다음 [Advanced system settings], [Advanced] 탭, [Environment Variables]를 선택합니다.) 자세한 내용은 Windows 설명서를 참조하십시오.

IAM SAML 자격 증명 공급자 생성

IAM SAML 2.0 자격 증명 공급자는 [SAML 2.0\(Security Assertion Markup Language 2.0\)](#) 표준을 지원하는 외부 자격 증명 공급자(IdP) 서비스를 기술하는 IAM의 엔터티입니다. 조직의 사용자가 AWS 리소스에 액세스 할 수 있도록 Shibboleth 또는 Active Directory 연동 서비스와 같은 SAML 호환 IdP와 AWS 간에 신뢰를 구축하고자 할 때 IAM 자격 증명 공급자를 사용합니다. IAM SAML 자격 증명 공급자는 IAM 신뢰 정책에서 보안 주체로 사용됩니다.

이 시나리오에 대한 자세한 정보는 [SAML 2.0 기반 연동에 대하여 \(p. 167\)](#)를 참조하십시오.

AWS Management 콘솔에서 또는 AWS CLI, Windows PowerShell용 도구 또는 AWS API 호출을 사용해 IAM 자격 증명 공급자를 생성하고 관리할 수 있습니다.

SAML 공급자를 생성한 후에는 1개 이상의 IAM 역할을 생성해야 합니다. 역할은 AWS의 자격 증명으로서 자신만의 고유한 자격 증명이 없지만(사용자가 그러하듯이), 이 콘텐츠에서는 조직의 IdP에 의해 인증된 연동 사용자에게 동적으로 할당됩니다. 그 역할은 조직의 IdP가 AWS에 액세스하기 위해 임시 보안 자격 증명을 요청할 수 있도록 허용합니다. 역할에 할당된 정책은 연동 사용자가 AWS에서 하도록 허용된 것이 무엇인지 결정합니다. SAML 연동을 위한 역할을 생성하려면 [타사 자격 증명 공급자의 역할 만들기\(연동\) \(p. 215\)](#) 단원을 참조하십시오.

마지막으로 역할을 만든 후에는 AWS에 대한 정보와 연동 사용자가 사용하도록 하고 싶은 역할(들)로 IdP를 구성하여 SAML 신뢰를 완료합니다. 이를 가리켜 IdP와 AWS 간 신뢰 당사자 신뢰 구성이라고 합니다. 신뢰 당사자 신뢰를 구성하려면 [신뢰 당사자 신뢰로 SAML 2.0 IdP를 구성하고 클레임 추가하기 \(p. 179\)](#) 단원을 참조하십시오.

주제

- [IAM 자격 증명 공급자 생성 및 관리\(콘솔\) \(p. 178\)](#)
- [IAM SAML 자격 증명 공급자 생성 및 관리\(AWS CLI\) \(p. 178\)](#)
- [IAM SAML 자격 증명 공급자 만들기 및 관리\(AWS API\) \(p. 179\)](#)
- [신뢰 당사자 신뢰로 SAML 2.0 IdP를 구성하고 클레임 추가하기 \(p. 179\)](#)
- [타사 SAML 솔루션 공급자를 AWS와 통합 \(p. 180\)](#)
- [인증 응답을 위한 SAML 어설션 구성 \(p. 181\)](#)

IAM 자격 증명 공급자 생성 및 관리(콘솔)

AWS Management 콘솔 콘솔을 사용하여 IAM SAML 자격 증명 공급자를 만들고 삭제할 수 있습니다.

IAM 자격 증명 공급자를 생성하는 방법(콘솔)

1. IAM 자격 증명 공급자를 생성하기 전에 IdP에게서 얻는 SAML 메타데이터 문서가 필요합니다. 이 문서에는 발급자 이름, 만료 정보 및 IdP에서 받은 SAML 인증 응답(어설션)의 유효성을 검증하는데 사용할 수 있는 키가 포함되어 있습니다. 메타데이터 문서를 생성하려면 조직이 IdP로 사용하는 자격 증명 관리 소프트웨어를 사용하십시오. 필요한 SAML 메타데이터 문서를 생성하는 방법을 비롯해, 사용 가능한 다수의 IdP를 구성하여 AWS에서 작동되도록 하는 방법에 대한 지침은 [타사 SAML 솔루션 공급자를 AWS와 통합 \(p. 180\)](#) 단원을 참조하십시오.

Important

메타데이터 파일은 바이트 순서 표시(BOM)가 없는 UTF-8 형식으로 인코딩되어야 합니다. 또한 SAML 메타데이터 문서의 일부로 포함된 x.509 인증서는 1,024비트 이상의 키를 사용해야 합니다. 키 크기가 이보다 작으면 "메타데이터를 구문 분석할 수 없음" 오류로 인해 IdP 생성에 실패합니다. BOM을 제거하려면 Notepad++와 같은 텍스트 편집 도구를 사용해 파일을 UTF-8로 인코딩합니다.

2. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
3. 탐색 창에서 자격 증명 공급자와 공급자 생성을 차례로 클릭합니다.
4. Provider Type(공급자 유형)에서 공급자 유형 선택(Choose a provider type)을 클릭한 다음 SAML을 클릭합니다.
5. 자격 증명 공급자의 이름을 입력합니다.
6. 메타데이터 문서에서 파일 선택을 클릭하고 Step 1에서 다운로드한 SAML 메타데이터 문서를 지정한 다음, 열기를 클릭합니다. [Next Step]을 클릭합니다.
7. 자신이 제공한 정보를 확인하고 생성을 클릭합니다.

SAML 공급자를 삭제하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 자격 증명 공급자를 클릭합니다.
3. 삭제할 자격 증명 공급자 옆의 확인란을 선택합니다.
4. Delete Providers(공급자 삭제)를 클릭합니다.

IAM SAML 자격 증명 공급자 생성 및 관리(AWS CLI)

AWS CLI를 사용하여 SAML 공급자를 만들고 삭제할 수 있습니다.

IAM 자격 증명 공급자를 생성하고 메타데이터 문서를 업로드하는 방법(AWS CLI)

- 다음 명령을 실행합니다. `aws iam create-saml-provider`

IAM 자격 증명 공급자의 새 메타데이터 문서를 업로드하는 방법(AWS CLI)

- 다음 명령을 실행합니다. `aws iam update-saml-provider`

IAM SAML 자격 증명 공급자를 삭제하는 방법(AWS CLI)

1. (선택 사항) ARN, 생성 날짜, 만료 등 모든 공급자에 대한 정보를 나열하려면 다음 명령을 실행합니다.

- `aws iam list-saml-providers`
2. (선택 사항) ARN, 생성 날짜, 만료 등 특정 공급자에 대한 정보를 얻으려면 다음 명령을 실행합니다.
- `aws iam get-saml-provider`
3. IAM 자격 증명 공급자를 삭제하려면 다음 명령을 실행합니다.
- `aws iam delete-saml-provider`

IAM SAML 자격 증명 공급자 만들기 및 관리(AWS API)

AWS API를 사용하여 SAML 공급자를 만들고 삭제할 수 있습니다.

IAM 자격 증명 공급자를 생성하고 메타데이터 문서를 업로드하려면(AWS API)

- 다음 연산을 호출합니다. [CreateSAMLProvider](#)

IAM 자격 증명 공급자의 새 메타데이터 문서를 업로드하는 방법(AWS API)

- 다음 연산을 호출합니다. [UpdateSAMLProvider](#)

IAM 자격 증명 공급자를 삭제하는 방법(AWS API)

1. (선택 사항) ARN, 생성 날짜, 만료 등 모든 IdP에 대한 정보를 나열하려면 다음 연산을 호출합니다.
 - [ListSAMLProviders](#)
2. (선택 사항) ARN, 생성 날짜, 만료 등 특정 공급자에 대한 정보를 얻으려면 다음 연산을 호출합니다.
 - [GetSAMLProvider](#)
3. IdP를 삭제하려면 다음 연산을 호출합니다.
 - [DeleteSAMLProvider](#)

신뢰 당사자 신뢰로 SAML 2.0 IdP를 구성하고 클레임 추가하기

SAML 액세스를 위한 IAM 자격 증명 공급자 및 역할을 생성한다는 것은 외부 자격 증명 공급자(IdP)와 관련 사용자에게 허용된 작업을 AWS에 알려주는 것입니다. 그 다음 단계는 IdP에게 서비스 공급자인 AWS에 대해 알려주는 것입니다. 이를 가리켜 IdP와 AWS 간 신뢰 당사자 신뢰 추가라고 합니다. 신뢰 당사자 신뢰를 추가하기 위한 정확한 프로세스는 사용 중인 IdP에 따라 달라집니다. 자세한 정보는 자격 증명 관리 소프트웨어의 설명서를 참조하십시오.

오늘날 IdP는 신뢰 당사자 정보와 인증서가 저장된 XML 문서를 IdP가 읽을 수 있도록 URL 지정을 허용하는 곳이 많습니다. AWS의 경우 <https://signin.aws.amazon.com/static/saml-metadata.xml>을 사용할 수 있습니다.

URL을 직접 지정할 수 없는 경우 위 URL에서 XML 문서를 다운로드하여 IdP 소프트웨어로 가져오면 됩니다.

또한, AWS를 신뢰 당사자로 지정하는 IdP에서는 적절한 클레임 규칙을 생성해야 합니다. IdP는 AWS 앤드 포인트로 SAML 반응을 전송할 때 1개 이상의 클레임을 포함하는 SAML 어설션을 포함합니다. 클레임은 사용자 및 사용자 소속 그룹에 대한 정보입니다. 클레임 규칙은 그 정보를 SAML 속성에 매핑합니다. 이는 IAM 정책에서 AWS가 연동 사용자의 권한을 검사하는 데 필요한 속성이 IdP의 SAML 인증 응답에 저장되어 있는지 확인하도록 해줍니다. 자세한 정보는 다음 주제 단원을 참조하십시오.

- **AWS 리소스에 대한 SAML 연동 액세스를 허용하는 역할에 대한 개요 (p. 169)**. 이 주제에서는 IAM 정책의 SAML별 키 사용을 비롯해 이 키를 사용하여 SAML 연동 사용자의 권한을 제한하는 방법에 대해 살펴봅니다.
- **인증 응답을 위한 SAML 어설션 구성 (p. 181)**를 선택하십시오. 이 주제에서는 사용자에 대한 정보가 포함된 SAML 클레임을 구성하는 방법에 대해 살펴봅니다. 그 클레임은 SAML 어설션에 번들링되어 있으며 AWS로 전송되는 SAML 응답에 포함되어 있습니다. AWS 정책에 필요한 그 정보가 AWS가 인식하고 사용할 수 있는 형식으로 SAML 어설션에 반드시 포함되도록 해야 합니다.
- **타사 SAML 솔루션 공급자를 AWS와 통합 (p. 180)**. 이 주제에서는 자격 증명 솔루션과 AWS의 통합 방법에 대한 타사의 설명서 링크를 제공합니다.

타사 SAML 솔루션 공급자를 AWS와 통합

다음 링크는 AWS 연동을 처리할 타사 SAML 2.0 자격 증명 공급자(IdP) 솔루션을 구성하는 데 도움이 됩니다.

Note

AWS 지원 엔지니어는 타사 소프트웨어를 포함하는 몇 가지 통합 작업을 수행하여 비즈니스 및 엔터프라이즈 지원 플랜이 있는 고객을 지원할 수 있습니다. 지원되는 플랫폼 및 애플리케이션의 최신 목록은 AWS 지원 FAQ에서 [지원되는 타사 소프트웨어는 무엇입니까?](#)를 참조하십시오.

솔루션	추가 정보
Auth0	AWS Integration in Auth0 – Auth0 설명서 웹 사이트의 이 페이지에서는 AWS Management 콘솔에서 SSO(Single Sign-On)를 설정하는 방법을 설명하고 JavaScript 예제를 소개합니다.
Bitium	Configuring SAML for Amazon Web Services(AWS) – Bitium 지원 사이트의 이 문서는 Bitium을 사용하여 AWS를 SAML SSO로 설정하는 방법을 설명합니다.
Centrify	Configure Centrify and Use SAML for SSO to AWS – Centrify 웹 사이트의 이 페이지는 Centrify를 구성해 SSO를 위한 SAML을 AWS에 사용하는 방법에 대해 설명합니다.
CertiVox	M-Pin SSO를 AWS 내에서 자격 증명 공급자로 설정 – CertiVox 웹 사이트의 이 페이지는 M-Pin SSO 시스템을 통해 SSO 인증을 하도록 AWS서비스 공급자를 구성하는 방법을 설명합니다.
Clearlogin	Amazon Web Services Setup – Clearlogin 도움말 센터의 이 문서에서는 Clearlogin과 AWS 간에 SSO 기능을 설정하는 방법을 설명합니다.
Google G Suite	Amazon Web Services cloud application – Google G Suite Administrator Help 사이트의 이 문서에서는 G Suite를 SAML 2.0 IdP로 구성하고 AWS를 서비스 공급자로 구성하는 방법을 설명합니다.
Identacor	Configuring SSO (SAML) for AWS – Identacor 웹 사이트의 이 문서에서는 AWS용 SSO를 설정하고 활성화하는 방법을 설명합니다.
Matrix42	MyWorkspace 시작 안내서 - 이 안내서에서는 AWS Identity 서비스를 Matrix42 MyWorkspace와 통합하는 방법에 대해 설명합니다.
Microsoft AD FS(Active Directory Federation Services)	Enabling Federation to AWS Using Windows Active Directory, AD FS, and SAML 2.0 – AWS 보안 블로그의 이 게시물은 EC2

솔루션	추加 정보
	<p>인스턴스에서 AD FS를 설정하고 AWS로 SAML 연동을 활성화하는 방법을 보여줍니다.</p> <p>PowerShell Automation to Give AWS Console Access – Sivaprasad Padisetty의 블로그에 실린 이 게시물은 Active Directory 및 AD FS를 설정하고 AWS를 통해 SAML 연동을 활성화하는 프로세스를 Windows PowerShell을 사용하여 자동화하는 방법을 설명합니다.</p>
miniOrange	<p>SSO for AWS – miniOrange 웹 사이트의 이 페이지에서는 AWS에 대한 대기업 보안 액세스 및 AWS 애플리케이션에 대한 완전한 액세스 제어를 설정하는 방법을 설명합니다.</p>
Okta	<p>Okta를 이용한 Amazon Web Services 명령줄 인터페이스 통합 – Okta 지원 사이트의 이 페이지에서는 Okta를 AWS와 함께 사용하도록 구성하는 방법을 알아볼 수 있습니다.</p>
OneLogin	<p>OneLogin 지식 베이스에서 SAML AWS라는 검색어를 입력하여 단일 역할 및 다중 역할 시나리오에 대비해 OneLogin과 AWS 사이에 AWS SSO 기능을 설정하는 방법이 설명된 일련의 문서를 찾으십시오.</p>
Ping Identity	<p>PingFederate AWS Connector – Ping Identity 웹 사이트의 이 페이지에서는 AWS에서 사용자 계정의 SSO를 사용하도록 PingFederate 서버를 구성하는 방법을 담은 PDF 파일을 다운로드할 수 있습니다.</p> <p>Ping 자격 증명 지원 페이지에서 SSO AWS라는 검색어를 입력하여 PingOne과 AWS 사이에 AWS SSO 기능을 설정하는 방법이 설명된 일련의 문서를 찾으십시오.</p>
RadiantLogic	<p>Radiant Logic Technology Partners – Radiant Logic의 RadiantOne 연동 자격 증명 서비스를 AWS와 통합하여 SAML 기반의 SSO를 위한 자격 증명 허브를 구축할 수 있습니다.</p>
Salesforce.com	<p>How to configure SSO from Salesforce to AWS – Salesforce.com 개발자 사이트에 실린 이 사용 방법 설명서에서는 Salesforce에 IdP(자격 증명 공급자)를 설정하고 AWS를 서비스 공급자로 구성하는 방법을 설명합니다.</p>
SecureAuth	<p>AWS - SecureAuth SAML SSO – SecureAuth 웹 사이트의 이 문서는 SecureAuth 어플라이언스를 위해 SAML과 AWS의 통합을 설정하는 방법을 설명합니다.</p>
Shibboleth	<p>How to Use Shibboleth for SSO to the AWS Management 콘솔 – AWS 보안 블로그의 이 항목에서는 Shibboleth를 설정하고 이를 AWS에 대한 자격 증명 공급자로 구성하는 방법을 단계별로 안내합니다.</p>

자세한 정보는 AWS 웹 사이트의 [IAM Partners](#) 페이지 단원을 참조하십시오.

인증 응답을 위한 SAML 어설션 구성

조직에서 한 사용자의 자격 증명이 확인된 후에 외부 자격 증명 공급자(IdP)는 인증 응답을 <https://signin.aws.amazon.com/saml>의 AWS SAML 엔드포인트로 보냅니다. 이 응답은 [SAML 2.0을 위한 HTTP POST 바인딩](#) 표준을 준수하고 다음 요소 또는 클레임이 저장된 SAML 토큰을 포함하는 POST 요청

입니다. SAML 호환 IdP에서 이 클레임들을 구성합니다. 이 클레임들을 입력하는 방법에 대한 자침에 대해서는 귀하의 IdP를 위한 문서를 참고하십시오.

IdP가 AWS에 클레임이 포함된 리소스를 전송하는 경우 수신 클레임 중 다수가 AWS 콘텍스트 키에 매핑됩니다. 이러한 콘텍스트 키는 Condition 요소를 사용하여 IAM 정책에서 확인할 수 있습니다. 사용 가능한 매핑 목록은 [SAML 속성을 AWS 신뢰 정책 콘텍스트 키에 매핑 \(p. 184\)](#) 섹션에 나와 있습니다.

Subject 및 NameID

다음 발췌문은 한 가지 예를 보여줍니다. 자신의 값을 표시된 것으로 대체합니다.

SubjectConfirmation 속성과 SubjectConfirmationData 속성을 둘 다 포함하는 NotOnOrAfter 요소와 함께 정확하게 Recipient 요소가 하나 있어야 합니다. 이러한 속성에는 다음 예제에서처럼 AWS 엔드포인트(<https://signin.aws.amazon.com/saml>)와 일치해야 하는 값이 포함되어 있습니다. Single Sign-On 상호 작용에 지원되는 이름 식별자 형식에 대한 자세한 정보는 [Oracle Sun OpenSSO Enterprise Administration Reference](#) 단원을 참조하십시오.

```
<Subject>
  <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">cbb88bf52c2510eabe00c1642d4643f41430fe25e3</NameID>
  <SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
    <SubjectConfirmationData NotOnOrAfter="2013-11-05T02:06:42.876Z">
      Recipient="https://signin.&awsdomain;/saml/"
    </SubjectConfirmationData>
  </SubjectConfirmation>
</Subject>
```

AudienceRestriction 및 Audience

보안상의 이유로 AWS는 IdP가 AWS로 보낸다는 SAML 어설션에서 대상으로 포함되어야 합니다. Audience 요소의 값에 대해 <https://signin.aws.amazon.com/saml> 또는 <urn:amazon:webservices>를 지정합니다. SAML 어설션의 다음 샘플 XML 조각은 이 키가 어떻게 IdP에 의해 지정되는지 보여줍니다. 사용 사례에 적용되는 샘플을 포함합니다.

```
<Conditions>
  <AudienceRestriction>
    <Audience>https://signin.&awsdomain;/saml/</Audience>
  </AudienceRestriction>
</Conditions>
```

```
<Conditions>
  <AudienceRestriction>
    <Audience>urn:amazon:webservices</Audience>
  </AudienceRestriction>
</Conditions>
```

Important

IdP의 SAML 어설션에 있는 SAML AudienceRestriction 값은 IAM 정책에서 테스트할 수 있는 [saml:aud](#) 콘텍스트 키에 매핑되지 않습니다. 그 대신에 [saml:aud](#) 콘텍스트 키는 SAML 수신자 속성에서 온 것으로, 그 이유는 이 속성이 [accounts.google.com:aud](#)와 같은 OIDC 대상 필드와 동일한 SAML이기 때문입니다.

Attribute 속성이 Name로 설정된 <https://aws.amazon.com/SAML/Attributes/Role> 요소

이 요소는 IdP에 의해 사용자가 매핑되는 IAM 자격 증명 공급자 및 역할을 나열하는 AttributeValue 요소를 한 개 이상 포함합니다. IAM 역할과 IAM 자격 증명 공급자는 [AssumeRoleWithSAML](#)로 전달되는 RoleArn 및 PrincipalArn 파라미터와 동일한 형식의 쉼표로 구분된 ARN 페어로 지정됩니다. 이 요소는 하나 이상의 역할 공급자 쌍, 즉 하나 이상의 AttributeValue 요소를 포함해야 하며 여러 개의 쌍을 포함할 수 있습니다. 요소가 다수의 페어를 포함하는 경우, 사용자는 AWS Management 콘솔에 로그인하기 위해 WebSSO를 사용할 때 어떤 역할을 수임할지 선택하라는 요구를 받습니다.

Important

Name 태그의 Attribute 속성 값은 대/소문자를 구분합니다. 정확하게 <https://aws.amazon.com/SAML/Attributes/Role>로 설정해야 합니다.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/Role">
  <AttributeValue>arn:aws:iam::account-number:role/role-name1,arn:aws:iam::account-
  number:saml-provider/provider-name</AttributeValue>
  <AttributeValue>arn:aws:iam::account-number:role/role-name2,arn:aws:iam::account-
  number:saml-provider/provider-name</AttributeValue>
  <AttributeValue>arn:aws:iam::account-number:role/role-name3,arn:aws:iam::account-
  number:saml-provider/provider-name</AttributeValue>
</Attribute>
```

Attribute 속성이 Name로 설정된 <https://aws.amazon.com/SAML/Attributes/RoleSessionName> 요소

이 요소에는 SSO를 위해 발급된 AWS 임시 자격 증명에 식별자를 제공하는 AttributeValue 요소가 하나 포함되어 있습니다. 이 요소는 AWS Management 콘솔 콘솔에서 사용자 정보를 표시하는데 사용됩니다. AttributeValue 요소 값은 길이가 2~64자여야 하며 영숫자, 밑줄 및 다음 문자만 포함할 수 있습니다. +(더하기 기호), =(등호), ,(쉼표), ,(마침표), @(@ 기호), -(하이픈). 공백은 포함할 수 없습니다. 값은 일반적으로 사용자 ID(bobsmith) 또는 이메일 주소(bobsmith@example.com)입니다. 사용자의 표시 이름(Bob Smith)과 같이 값이 공백을 포함하면 안 됩니다.

Important

Name 태그의 Attribute 속성 값은 대/소문자를 구분합니다. 정확하게 <https://aws.amazon.com/SAML/Attributes/RoleSessionName>로 설정해야 합니다.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/RoleSessionName">
  <AttributeValue>user-id-name</AttributeValue>
</Attribute>
```

Attribute 속성이 SessionDuration으로 설정된 선택적 <https://aws.amazon.com/SAML/Attributes/SessionDuration> 요소

이 요소에는 사용자가 AWS Management 콘솔에 액세스할 수 있는 기간을 지정하는 AttributeValue 요소가 한 개 들어 있습니다. 이 시간이 지나면 새로운 임시 자격 증명을 요청해야 합니다. 이 값은 세션에 대한 기간(초)을 나타내는 정수입니다. 이 값은 900초(15분)~43200초(12시간)일 수 있습니다. 이 속성이 없으면 자격 증명은 한 시간 동안 지속됩니다(DurationSeconds API의 AssumeRoleWithSAML 파라미터 기본값).

이 속성을 사용하려면 <https://signin.aws.amazon.com/saml>에서 콘솔 로그인 웹 엔드포인트를 통해 AWS Management 콘솔에 대한 SSO(Single Sign-On) 액세스를 제공하도록 SAML 공급자를 구성해야 합니다. 이 속성은 AWS Management 콘솔에 대해서만 세션을 연장할 수 있습니다. 다른 자격 증명의 수명을 늘릴 수는 없습니다. 그러나 AssumeRoleWithSAML API 호출에 이 속성이 있다면 호출에 의해 반환되는 자격 증명의 수명을 기본값인 60분 미만으로 줄일 수 있습니다.

이와 함께 SessionNotOnOrAfter 속성도 정의되어 있다면 SessionDuration 또는 SessionNotOnOrAfter 속성 중 더 작은 값으로 콘솔 세션의 최대 지속 시간을 정합니다.

지속 기간을 더 늘려 콘솔 세션을 활성화하면 자격 증명이 손상될 위험이 높아집니다. 이러한 위험을 줄이려면 IAM 콘솔의 Role Summary(역할 요약) 페이지에서 Revoke Sessions(세션 취소)를 선택하여 원하는 역할의 활성 콘솔 세션을 즉시 비활성화하면 됩니다. 자세한 정보는 [IAM 역할의 임시 보안 자격 증명 취소 \(p. 245\)](#) 단원을 참조하십시오.

Important

Name 태그의 Attribute 속성 값은 대/소문자를 구분합니다. 정확하게 <https://aws.amazon.com/SAML/Attributes/SessionDuration>로 설정해야 합니다.

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/SessionDuration">
  <AttributeValue>1800</AttributeValue>
</Attribute>
```

SAML 속성을 AWS 신뢰 정책 콘텍스트 키에 매핑

이 섹션의 표들은 흔히 사용되는 SAML 속성들을 나열하고, 그 속성들이 AWS에서 신뢰 정책 조건 콘텍스트 키에 어떻게 매핑되는지 보여줍니다. 역할 액세스를 위한 SAML 요청에 달린 어설션에 포함된 값과 이러한 키를 비교하여 평가함으로써 역할의 액세스를 제어할 수 있습니다.

Important

이런 키는 IAM 신뢰 정책(역할을 수임할 수 있는 사용자를 결정하는 정책)에서만 사용할 수 있고, 권한 정책에는 적용할 수 없습니다.

eduPerson 및 eduOrg 속성 표에서 값은 문자열 또는 문자열 목록의 형태로 입력됩니다. 문자열 값의 경우, StringEquals 또는 StringLike 조건을 이용해 IAM 신뢰 정책에서 이러한 값을 테스트할 수 있습니다. 문자열 목록이 포함된 값의 경우에는 ForAnyValue 및 ForAllValues 정책 설정 연산자 ([p. 520](#))를 사용해 신뢰 정책에서 값을 테스트합니다.

Note

AWS 콘텍스트 키당 하나의 클레임만을 포함해야 합니다. 하나 이상의 클레임을 포함하는 경우, 하나의 클레임만 매핑됩니다.

eduPerson 및 eduOrg 속성

eduPerson 또는 eduOrg 속성(Name 키)	이 AWS 콘텍스트 키 (FriendlyName 키)에 매핑	유형
urn:oid:1.3.6.1.4.1.5923.1.1.1.1	eduPersonAffiliation	문자열 목록
urn:oid:1.3.6.1.4.1.5923.1.1.1.2	eduPersonNickname	문자열 목록
urn:oid:1.3.6.1.4.1.5923.1.1.1.3	eduPersonOrgDN	문자열
urn:oid:1.3.6.1.4.1.5923.1.1.1.4	eduPersonOrgUnitDN	문자열 목록
urn:oid:1.3.6.1.4.1.5923.1.1.1.5	eduPersonPrimaryAffiliation	문자열
urn:oid:1.3.6.1.4.1.5923.1.1.1.6	eduPersonPrincipalName	문자열
urn:oid:1.3.6.1.4.1.5923.1.1.1.7	eduPersonEntitlement	문자열 목록
urn:oid:1.3.6.1.4.1.5923.1.1.1.8	eduPersonPrimaryOrgUnitDN	문자열
urn:oid:1.3.6.1.4.1.5923.1.1.1.9	eduPersonScopedAffiliation	문자열 목록
urn:oid:1.3.6.1.4.1.5923.1.1.1.10	eduPersonTargetedID	문자열 목록
urn:oid:1.3.6.1.4.1.5923.1.1.1.11	eduPersonAssurance	문자열 목록
urn:oid:1.3.6.1.4.1.5923.1.2.1.2	eduOrgHomePageURI	문자열 목록
urn:oid:1.3.6.1.4.1.5923.1.2.1.3	eduOrgIdentityAuthNPolicy	문자열 목록
urn:oid:1.3.6.1.4.1.5923.1.2.1.4	eduOrgLegalName	문자열 목록
urn:oid:1.3.6.1.4.1.5923.1.2.1.5	eduOrgSuperiorURI	문자열 목록
urn:oid:1.3.6.1.4.1.5923.1.2.1.6	eduOrgWhitePagesURI	문자열 목록

eduPerson 또는 eduOrg 속성(Name 키)	이 AWS 콘텍스트 키 (FriendlyName 키)에 매핑	유형
urn:oid:2.5.4.3	cn	문자열 목록

Active Directory 속성

AD 속성	이 AWS 콘텍스트 키에 대한 매핑	유형
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name	name	문자열
http://schemas.xmlsoap.org/claims/CommonName	commonName	문자열
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	givenName	문자열
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	surname	문자열
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress	mail	문자열
http://schemas.microsoft.com/ws/2008/06/identity/claims/primarygroupsid	uid	문자열

X.500 속성

X.500 속성	이 AWS 콘텍스트 키에 대한 매핑	유형
2.5.4.3	commonName	문자열
2.5.4.4	surname	문자열
2.4.5.42	givenName	문자열
2.5.4.45	x500UniqueIdentifier	문자열
0.9.2342.19200300100.1.1	uid	문자열
0.9.2342.19200300100.1.3	mail	문자열
0.9.2342.19200300.100.1.45	organizationStatus	문자열

SAML 2.0 연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하기

역할을 사용해 SAML 2.0 호환 자격 증명 공급자(IdP) 및 AWS를 구성하여 연동 사용자가 AWS Management 콘솔에 액세스하도록 허용할 수 있습니다. 역할은 콘솔에서 작업을 수행할 수 있는 권한을 사용자에게 부여합니다. 그 대신에 SAML 연동 사용자가 다른 방법으로 AWS에 액세스할 수 있게 하려면, 다음 주제들 중 하나를 참조하십시오.

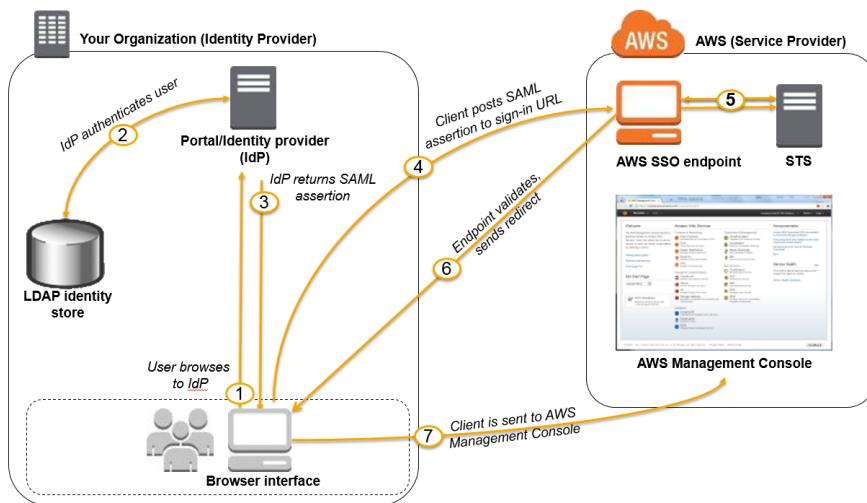
- AWS CLI: IAM 역할로 전환하기(AWS CLI) (p. 235)
- Windows PowerShell용 도구: IAM 역할로 전환하기(Windows PowerShell용 도구) (p. 236)
- AWS API: IAM 역할(AWS API)로 전환하기 (p. 238)

개요

다음 다이어그램은 SAML 지원 Single Sign-On의 흐름을 보여줍니다.

Note

이와 같은 SAML의 특수한 사용이 [SAML 2.0 기반 연동에 대하여 \(p. 167\)](#)에 설명된 더 일반적인 사용과 차이가 나는 이유는, 이 워크플로우가 사용자를 대신해 AWS Management 콘솔을 열기 때문입니다. 이를 위해서는 AssumeRoleWithSAML API를 직접 호출하는 대신 AWS SSO 엔드포인트를 사용해야 합니다. 엔드포인트는 사용자를 위해 API를 호출하고 사용자의 브라우저를 AWS Management 콘솔로 자동 리디렉션하는 URL을 반환합니다.



다이어그램은 다음 단계들을 보여줍니다.

1. 사용자는 검색을 통해 조직의 포털에 이르러 옵션을 선택해 AWS Management 콘솔로 갑니다. 조직에서 포털은 일반적으로, 조직과 AWS 간의 신뢰 교환을 다루는 IdP의 기능을 담당합니다. 예를 들어 Active Directory Federation Services에서 포털 URL은 <https://ADFSServiceName/adfs/ls/IdpInitiatedSignOn.aspx>입니다.
2. 포털은 사용자의 조직 내 자격 증명을 확인합니다.
3. 포털은 사용자를 식별하고 사용자에 대한 속성을 포함하는 어설션이 포함된 SAML 인증 응답을 생성합니다. 콘솔 세션의 유효 기간을 지정하는 SessionDuration이라는 SAML 어설션 속성을 포함하여 IdP를 구성할 수도 있습니다. 포털은 이 응답을 클라이언트 브라우저로 전송합니다.
4. 클라이언트 브라우저는 AWS Single Sign-On 엔드포인트로 리디렉션되고 SAML 어설션을 게시합니다.
5. 엔드포인트는 사용자 대신 임시 보안 자격 증명을 요청하고 그 자격 증명을 사용하는 콘솔 로그인 URL을 생성합니다.
6. AWS는 리디렉션으로 클라이언트에게 로그인 URL을 반송합니다.
7. 클라이언트 브라우저는 AWS Management 콘솔로 리디렉션됩니다. SAML 인증 응답이 여러 개의 IAM 역할에 매핑되는 속성들을 포함하는 경우, 사용자는 콘솔에 액세스하는 데 사용할 역할을 선택하라는 메시지를 먼저 받습니다.

사용자의 시점에서는 그 과정을 투명하게 들여다볼 수 있습니다. 사용자는 조직의 내부 포털에서 시작하여 AWS 자격 증명을 제공할 필요 없이 AWS Management 콘솔에서 마칩니다.

세부 단계들에 대한 링크를 따라 이 행동을 구성하는 방법을 개관하시려면 다음 섹션들을 참조하십시오.

AWS에 대한 SAML 공급자로 네트워크 구성하기

귀하의 조직 네트워크의 내부에서 자격 증명 스토어(Windows Active Directory 등)를 구성해 Windows Active Directory Federation Services, Shibboleth와 같은 SAML 기반 IdP로 작업합니다. IdP를 사용하여 귀하의 조직을 IdP로 기술하고 인증 키를 포함하는 메타데이터 문서를 생성합니다. 또한 조직의 포털을 구성해, AWS Management 콘솔에 대한 사용자 요청을 SAML 어설션을 이용한 인증을 위해 AWS SAML 앤드포인트로 라우팅합니다. metadata.xml 파일을 생성하기 위해 IdP를 어떻게 구성하는가는 IdP에 따라 다릅니다. 자침을 보시려면 IdP의 문서를 참고하시거나 지원되는 SAML 공급자들 중 다수의 웹 문서 링크가 있는 [타사 SAML 솔루션 공급자를 AWS와 통합 \(p. 180\)](#)를 참조하십시오.

IAM에서 SAML 공급자 생성하기

그 다음에는 AWS Management 콘솔에 로그인하여 IAM 콘솔로 이동합니다. 그곳에서 새로운 SAML 공급자를 생성합니다. 그 공급자는 조직의 IdP에 대한 정보를 담고 있는 IAM의 엔터티입니다. 이 과정의 일부로 이 전 섹션에서 조직의 IdP 소프트웨어가 생성한 메타데이터 문서를 업로드합니다. 자세한 정보는 [IAM SAML 자격 증명 공급자 생성 \(p. 177\)](#) 단원을 참조하십시오.

연동된 사용자들을 위해 AWS에서 권한 구성하기

그 다음 단계는 조직의 IdP와 IAM 간에 신뢰 관계를 수립하고 연동을 목적으로 IdP를 보안 주체(신뢰할 수 있는 대상)로 식별하는 IAM 역할을 생성하는 것입니다. 그 역할은 조직의 IdP에 의해 인증된 사용자들이 AWS에서 할 수 있도록 허용되는 것이 무엇인지 정의하기도 합니다. IAM 콘솔을 사용하여 이 역할을 생성할 수 있습니다. 누가 그 역할을 담당할 수 있는지 알려주는 신뢰 정책을 만들 때, 사용자가 그 역할을 맡도록 허용되기 위해 일치시켜야 하는 1개 이상의 SAML 속성들과 함께 앞서 IAM에서 생성한 SAML 공급자를 지정합니다. 예를 들어 SAML eduPersonOrgDN 값이 ExampleOrg인 사용자에게만 로그인을 허용하도록 구성할 수 있습니다. 그 역할 마법사는 조건을 자동으로 추가해 saml:aud 속성을 테스트함으로써 그 역할이 AWS Management 콘솔에 로그인하는 것을 위해서만 위임되는 것인지 확인합니다. 그 역할을 위한 신뢰 정책은 다음과 같을 수 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Principal": {"Federated": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:saml-provider/ExampleOrgSSOProvider"},  
         "Action": "sts:AssumeRoleWithSAML",  
         "Condition": {"StringEquals": {  
             "saml:edupersonorgdn": "ExampleOrg",  
             "saml:aud": "https://signin.aws.amazon.com/saml"  
         }}  
    ]  
}
```

역할의 [권한 정책 \(p. 305\)](#)에 대해 어떤 역할, 사용자, 또는 그룹에 사용하는 방식으로 권한을 지정합니다. 예를 들어, 조직의 사용자가 Amazon EC2 인스턴스를 관리하도록 허용될 경우 권한 정책에서 명시적으로 Amazon EC2 작업을 허용합니다. 이것은 Amazon EC2 전체 액세스 관리형 정책과 같은 [관리형 정책 \(p. 391\)](#)을 배정함으로써 할 수 있습니다.

SAML IdP를 위한 역할 생성에 관한 자세한 정보는 [SAML 2.0 연동을 위한 역할 생성\(콘솔\) \(p. 221\)](#)을 참조하십시오.

SAML IdP 구성을 완료하고 SAML 인증 응답에 대한 어설션 생성하기

역할을 생성한 후에는 <https://signin.aws.amazon.com/static/saml-metadata.xml>에 있는 saml-metadata.xml 파일을 설치하여 SAML IdP에게 서비스 공급자가 AWS라고 알려주십시오. 그 파일의 설치 방법은 IdP에 따라 다릅니다. 어떤 IdP는 URL을 입력할 수 있는 옵션을 제공하고, 그 결과 IdP가 그 파일을 획득하고 설치해 줍니다. 다른 IdP들의 경우에는 URL에서 파일을 내려받은 다음 로컬 파일로 제공해야 합

니다. 세부 정보를 보시려면 IdP의 문서를 참고하시거나 지원되는 SAML 공급자들 중 다수의 웹 문서 링크가 있는 [타사 SAML 솔루션 공급자를 AWS와 통합 \(p. 180\)](#)를 참조하십시오.

또한, IdP가 인증 응답의 일부로 AWS에 SAML 속성으로 전달하기 원하는 정보를 구성합니다. 이 정보의 대부분은 정책에서 평가할 수 있는 조건 콘텍스트 키로 AWS에 나타남으로써 올바른 콘텍스트에서 인증된 사용자만이 AWS 리소스에 대한 액세스 권한을 부여받도록 보장합니다. 콘솔 사용 시간을 일정한 길이로 지정하여 제한하거나 콘솔에 액세스할 수 있는 최대 시간(최대 12시간)을 지정하고 그 이후에는 자격 증명을 새로 고치도록 할 수 있습니다. 자세한 정보는 [인증 응답을 위한 SAML 어설션 구성 \(p. 181\)](#) 단원을 참조하십시오.

연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하는 URL 생성(사용자 지정 연동 브로커)

코드를 작성하고 실행해 조직 네트워크에 로그인하는 사용자가 AWS Management 콘솔에 안전하게 액세스할 수 있게 하는 URL을 생성할 수 있습니다. 그 URL에는 AWS에서 얻고 AWS에 사용자를 인증하는 로그인 토큰이 포함되어 있습니다.

Note

조직에서 SAML과 호환이 되는 자격 증명 공급자(IdP)를 사용한다면, 코드를 작성하지 않고도 콘솔에 대한 액세스를 설정할 수 있습니다. 이는 Microsoft의 Active Directory Federation Services 또는 오픈 소스 Shibboleth와 같은 공급자와 함께 작동합니다. 자세한 정보는 [SAML 2.0 연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하기 \(p. 185\)](#) 단원을 참조하십시오.

조직의 사용자가 AWS Management 콘솔에 액세스할 수 있도록 하려는 경우 다음 단계를 수행하여 사용자 지정 자격 증명 브로커를 생성할 수 있습니다.

1. 사용자가 로컬 자격 증명 시스템에 의해 인증되는지 확인합니다.
2. AWS Security Token Service(AWS STS) [AssumeRole](#)(권장) 또는 [GetFederationToken](#) API 작업을 호출하여 사용자를 위한 임시 보안 자격 증명을 얻을 수 있습니다. 역할을 수임하는 데 사용할 수 있는 여러 방법을 알아보려면 [IAM 역할 사용 \(p. 227\)](#) 단원을 참조하십시오.
 - 역할에 대한 임시 보안 자격 증명을 얻기 위해 [AssumeRole*](#) API 작업 중 하나를 사용한 경우 이 호출에는 DurationSeconds 파라미터를 포함할 수 있습니다. 이 파라미터는 역할 세션 기간을 900초(15초)에서 해당 역할에 대한 최대 세션 기간 설정까지 지정합니다. 역할에 대한 최댓값을 확인 또는 변경하는 방법을 알아보려면 [역할에 대한 최대 세션 기간 설정 보기 \(p. 228\)](#) 단원을 참조하십시오. 또한 [AssumeRole*](#) API 작업을 사용하는 경우 장기 자격 증명을 사용하여 IAM 사용자로 호출해야 합니다. 그렇지 않으면 3단계의 연동 엔드포인트 호출에 실패합니다.
 - 보안 자격 증명을 얻기 위해 [GetFederationToken](#) API 작업을 사용한 경우 이 호출에는 DurationSeconds 파라미터를 포함할 수 있습니다. 이 파라미터는 역할 세션에 대한 기간을 지정합니다. 이 값은 900초(15분)~129,600초(36시간)일 수 있습니다. IAM 사용자에 대한 장기 AWS 보안 자격 증명을 사용하는 경우에만 이 API를 호출할 수 있습니다. AWS 계정 투트 사용자 자격 증명을 사용하여 호출할 수도 있지만 이는 권장되는 방법은 아닙니다. 루트 사용자로 호출한 경우 기본 세션은 한 시간 동안 지속됩니다. 또는 900초(15분)에서 최대 3,600초(1시간)로 세션을 지정할 수 있습니다.
3. AWS 연동 엔드포인트를 호출하고 임시 보안 자격 증명을 제공하여 로그인 토큰을 요청하십시오.
4. 토큰을 포함하는 콘솔에 대한 URL을 생성합니다:
 - URL에 [AssumeRole*](#) API 작업 중 하나를 사용하는 경우 sessionDuration HTTP 파라미터를 포함할 수 있습니다. 이 파라미터는 콘솔 세션 시간을 900초(15분)~43200초(12시간)로 지정합니다.
 - URL에 [GetFederationToken](#) API 작업을 사용하는 경우 DurationSeconds 파라미터를 포함할 수 있습니다. 이 파라미터는 연동된 콘솔 세션에 대한 기간을 지정합니다. 이 값은 900초(15분)~129,600초(36시간)일 수 있습니다.

Note

SessionDuration을 사용하여 임시 자격 증명을 얻을 경우에는 [GetFederationToken](#) HTTP 파라미터를 사용하지 마십시오. 이 파라미터를 사용하면 작업이 실패합니다.

5. 사용자에게 URL을 부여하거나 사용자 대신 URL을 호출합니다.

연동 엔드포인트가 제공하는 URL은 생성된 후 15분 동안 유효합니다. 이 시간은 URL과 연결된 임시 보안 자격 증명 세션의 기간(초)과 다릅니다. 이러한 자격 증명은 생성된 시각을 시작으로 생성 시 지정한 기간 동안 유효합니다.

Important

URL은 연결된 임시 보안 자격 증명에서 권한을 허용한 경우 AWS Management 콘솔을 통해 AWS 리소스에 대한 액세스 권한을 부여한다는 것에 유의하십시오. 이러한 이유 때문에 URL은 비밀로 취급해야 합니다. 예를 들어 SSL 연결을 통해 302 HTTP 응답 상태 코드를 사용함으로써 안전한 리디렉션을 통해 URL을 반환하는 것이 좋습니다. 302 HTTP 응답 상태 코드에 대한 자세한 정보는 [RFC 2616, 단원 10.3.3](#)을 참조하십시오.

Single Sign-On 솔루션을 실행하는 방법을 보여주는 샘플 애플리케이션을 보려면 AWS 샘플 코드 및 라이브러리의 [AWS Management 콘솔 연동 프록시 샘플 사용 사례](#)를 참조하십시오.

이 작업을 완료하려면 [AWS Identity and Access Management를 위한 HTTPS 쿼리 API\(IAM\)](#) 및 [AWS Security Token Service\(AWS STS\)](#)를 참조하십시오. 아니면 적절한 [AWS SDK](#)와 함께 Java, Ruby 또는 C#과 같은 프로그래밍 언어를 사용할 수도 있습니다. 다음 단원에서는 이들 각 메서드에 대해 설명합니다.

주제

- [IAM 쿼리 API 작업을 사용한 예제 코드 \(p. 189\)](#)
- [Python을 사용한 예제 코드 \(p. 191\)](#)
- [Java를 사용한 예제 코드 \(p. 192\)](#)
- [URL을 생성하는 방법을 보여주는 예\(Ruby\) \(p. 194\)](#)

IAM 쿼리 API 작업을 사용한 예제 코드

연동 사용자에게 AWS Management 콘솔에 대한 직접 액세스를 부여하는 URL을 생성할 수 있습니다. 이 작업은 IAM 및 AWS STS HTTPS Query API를 사용합니다. 쿼리 요청에 대한 자세한 정보는 [쿼리 요청 실행 단원](#)을 참조하십시오.

Note

다음 절차에는 텍스트 문자열에 대한 예시가 있습니다. 가독성을 증진하기 위해 일부 긴 예시에는 줄 바꿈이 추가되었습니다. 자신만이 쓸 용도로 이러한 문자열을 생성할 때는 줄 바꿈을 모두 빼야 합니다.

AWS Management 콘솔에서 연동 사용자에게 리소스 액세스 권한을 부여하려면

1. 자격 증명 및 인증 시스템에서 사용자를 인증합니다.
2. 사용자에 대한 임시 보안 자격 증명을 얻습니다. 임시 자격 증명은 액세스 키 ID, 보안 액세스 키 및 보안 토큰으로 구성됩니다. 임시 자격 증명 생성에 대한 자세한 정보는 다음([임시 보안 자격 증명 \(p. 263\)](#))을 참조하십시오.

임시 자격 증명을 얻으려면 AWS STS [AssumeRole API](#)(권장) 또는 [GetFederationToken API](#)를 호출하면 됩니다. 이러한 API 작업 간의 차이에 대한 자세한 정보는 AWS 보안 블로그에서 [AWS 계정에 대한 액세스 권한을 안전하게 위임하기 위한 API 옵션 이해하기](#)를 참조하십시오.

Important

[GetFederationToken API](#)를 사용하여 임시 보안 자격 증명을 생성한 경우 해당 역할을 수입한 사용자에게 자격 증명을 부여하는 권한을 지정해야 합니다. [AssumeRole*](#)로 시작하는 API 작업 중 어느 것에 대해서도 IAM 역할을 사용해 권한을 할당할 수 있습니다. 다른 API 작업의 경우 그 메커니즘이 API에 따라 달라집니다. 자세한 정보는 [사용자 임시 보안 자격 증명에 대한](#)

[권한 제어 \(p. 277\)](#) 단원을 참조하십시오. 또한 `AssumeRole*` API 작업을 사용하는 경우 장기 자격 증명을 사용하여 IAM 사용자로 호출해야 합니다. 그렇지 않으면 3단계의 연동 엔드포인트 호출에 실패합니다.

3. 임시 보안 자격 증명을 획득한 후에는 자격 증명을 JSON 세션 문자열로 구성해 로그인 토큰과 교환합니다. 다음 예에서는 자격 증명을 인코딩하는 방법을 보여줍니다. 자리 표시자 텍스트를 이전 단계에서 받은 자격 증명의 적절한 값들로 교체합니다.

```
{"sessionId": "*** temporary access key ID ***",
 "sessionKey": "*** temporary secret access key ***",
 "sessionToken": "*** security token ***"}
```

4. [URL encode](#) 이전 단계의 세션 문자열. 인코딩하고 있는 정보가 지닌 중요성으로 인해 이러한 인코딩에는 웹 서비스를 사용하지 않는 것이 좋습니다. 대신 개발 도구 키트에 로컬로 설치된 함수 또는 기능을 사용하여 이 정보를 안전하게 인코딩합니다. Python의 `urllib.quote_plus` 함수, Java의 `URLEncoder.encode` 함수 또는 Ruby의 `CGI.escape` 함수를 사용할 수 있습니다. 이 주제 후반의 예제를 참조하십시오.
5. 다음 주소에서 AWS 연동 엔드포인트로 요청을 전송하십시오.

<https://signin.aws.amazon.com/federation>

요청에는 `Action` 및 `Session` 파라미터가 포함되어야 하며, `AssumeRole*` API 사용했다면 다음 예제의 `SessionDuration` HTTP 파라미터를 선택적으로 포함시킬 수 있습니다.

```
Action = getSigninToken
SessionDuration = time in seconds
Session = *** the URL encoded JSON string created in steps 3 & 4 ***
```

`SessionDuration` HTTP 파라미터는 연동된 콘솔 세션에 대한 기간을 지정합니다. 이 기간은 `DurationSeconds` 파라미터를 사용하여 지정하는 임시 자격 증명의 기간과는 다릅니다. `SessionDuration`의 최댓값은 43200(12시간)까지 지정할 수 있습니다. `SessionDuration` 파라미터가 없을 때는 2단계에서 AWS STS에서 검색한 자격 증명의 지속 시간을 세션의 기본값으로 사용합니다(기본 1시간). `DurationSeconds` 파라미터를 사용하여 기간을 지정하는 방법에 대한 자세한 정보는 [AssumeRole API 관련 문서](#)를 참조하십시오. 연동 엔드포인트의 `getSigninToken` 작업을 이용하면 한 시간보다 긴 콘솔 세션을 만들 수 있습니다.

Note

`SessionDuration`을 사용하여 임시 자격 증명을 얻을 경우에는 `GetFederationToken` HTTP 파라미터를 사용하지 마십시오. 이 파라미터를 사용하면 작업이 실패합니다.

지속 기간을 더 늘려 콘솔 세션을 활성화하면 자격 증명이 손상될 위험이 높아집니다. 이러한 위험을 줄이려면 IAM 콘솔 페이지의 Role Summary(역할 요약)에서 Revoke Sessions(세션 취소)를 선택하여 원하는 역할의 활성 콘솔 세션을 즉시 비활성화하면 됩니다. 자세한 정보는 [IAM 역할의 임시 보안 자격 증명 취소 \(p. 245\)](#)를 참조하십시오.

다음은 요청에 대한 예시입니다. 가독성을 위해 줄바꿈이 되어 있지만 한 줄로 된 문자열로 제출해야 합니다.

```
https://signin.aws.amazon.com/federation
?Action=getSigninToken
&SessionDuration=1800
&Session=%7B%22sessionId%22%3A+%22ASIAJUMHIZPTOKTBMK5A%22%2C+%22sessionKey%22
%3A+%22LSD7LWI%2FL%2FN%2BgYpan5QFz0XUpc8s7HYjRsgcsrsm%22%2C+%22sessionToken%2
2%3A+%22FQoDYXdzEBQaDLbj3VWv2u50NN%2F3yyLSASwYtWhPnGPMNmzzFFzsL0Qd3vtYHw5A5dW
AjOsrkPkgomIe3mJip%2F0djDBbo7SmO%2FENDEiCdpQKodTpleKA8xQq0CwFg6a69xdEBQT8
FipATnLbKoyS4b%2FehbnstUjZZQwpowXqFF7gSm%2FMe2tXe0jzsdP0012obeZ9lijPSdF1k2b5
PfGhiuyAR9ad5%2BubMOpY86fKex1qsytjvyTbZ9nXe6DvxVDcnCoHoGETJ7XFksFdH0v%2FYR25C
UAhJ3nXlkIbG7Ucv9cOEpCf%2Fg23ijRgILIBQ%3D%3D%22%7D
```

연동 엔드포인트의 응답은 signinToken 값이 있는 JSON 문서입니다. 다음의 예와 유사합니다.

```
{"SigninToken": "*** the SigninToken string ***"}
```

6. 마지막으로 연동 사용자가 AWS Management 콘솔에 액세스하는 데 사용할 수 있는 URL을 생성하십시오. 그 URL은 다음 파라미터와 함께 Step 5 (p. 190)에서 사용한 것과 동일한 연동 URL 엔드포인트입니다.

```
?Action = login
&Issuer = *** the form-urlencoded URL for your internal sign-in page ***
&Destination = *** the form-urlencoded URL to the desired AWS console page ***
&SigninToken = *** the value of SigninToken received in the previous step ***
```

다음 예는 최종 URL이 결국 어떤 모양을 갖추게 되는지 보여줍니다. 이 URL은 생성된 시각으로부터 15분 동안 유효합니다. URL에 내장된 콘솔 세션과 임시 보안 자격 증명은 이를 처음 요청할 때 SessionDuration HTTP 파라미터에 지정한 지속 기간만큼 유효합니다.

```
https://signin.aws.amazon.com/federation
?Action=login
&Issuer=https%3A%2F%2Fexample.com
&Destination=https%3A%2F%2Fconsole.aws.amazon.com%2Fs
&SigninToken=VCQgs5qZzt3Q6fn8Tr5EXAMPLEmLnwB7JjUc-SHwnUUWabcRdnWsi4DBn-dvC
CZ85wrD0nmlldUcZEXAMPLE-vXYH4Q_mleuF_W2BE5HYexbe9y4Of-kje53SsjNNecATfjIzpW1
WibbnH6YcYRiBoffZBGExbEXAMPLE5aiKX4THWjQKC6gg6alHu6JFrnOJoK3dtP6I9a6hi6yPgm
iOkPZMmNGmhsvVxetKzr8mx3pxhHbMEXAMPLEtvlpij0rok3IyCR2YVcIjqwfwv32HU2Xl471u
3fu6uOfUComeKiqtGX974xzJOZbdm_t_llrhEXAMPLEDDIissnyHgw2xaZzqudm4mo2uTDk9Pv
915KOZCqIgEXAMPLEcA6tgLPykEWGUyH6BdSC6166n4M4JkXIQgac7_7821YqixsNxZ6rsrpzwf
nQoS1407R0eJCCJ684EXAMPLERdBnnuLbUpz2Iw3vIN0tQgOujwnwydPscm9F7foaEK3jwMkg
Apeb1-6L_OB12MzhxFxx55555EXAMPLEhyETEd4ZulKPdXHkg16T9ZkI1Hz2Uy1RUTUhUxNtS9
nWC5xkbBoEcXqpoSiEKe7yhje9Vzhed61AEXAMPLElbWeouACEMG6-Vd3dAgFYd6i5FYoyFrZLWvm
OLSG7RyYKeYN5VIzUk3YWQpyjPORiT5KUrsUi-NEXAMPLExMOMdoODBEgKQsk-iu2ozh6r8bxwCRNhujg
```

Python을 사용한 예제 코드

다음 예는 Python을 사용해 연동 사용자에게 AWS Management 콘솔에 대한 직접 액세스 권한을 부여하는 URL을 프로그래밍 방식으로 생성하는 방법을 보여줍니다. 이 예제에서는 AWS SDK for Python (Boto)을 사용합니다.

코드는 AssumeRole API를 사용해 임시 보안 자격 증명을 획득합니다.

```
import urllib, json
import requests # 'pip install requests'
from boto.sts import STSConnection # AWS SDK for Python (Boto) 'pip install boto'

# Step 1: Authenticate user in your own identity system.

# Step 2: Using the access keys for an IAM user in your AWS account,
# call "AssumeRole" to get temporary access keys for the federated user

# Note: Calls to AWS STS AssumeRole must be signed using the access key ID
# and secret access key of an IAM user or using existing temporary credentials.
# The credentials can be in EC2 instance metadata, in environment variables,
# or in a configuration file, and will be discovered automatically by the
# STSConnection() function. For more information, see the Python SDK docs:
# http://boto.readthedocs.org/en/latest/boto_config_tut.html
sts_connection = STSConnection()

assumed_role_object = sts_connection.assume_role(
```

```
    role_arn="arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:role/ROLE-NAME",
    role_session_name="AssumeRoleSession"
)

# Step 3: Format resulting temporary credentials into JSON
json_string_with_temp_credentials = '{'
json_string_with_temp_credentials += '"sessionId":' + assumed_role_object.credentials.access_key + '',
json_string_with_temp_credentials += '"sessionKey":' + assumed_role_object.credentials.secret_key + '',
json_string_with_temp_credentials += '"sessionToken":' + assumed_role_object.credentials.session_token + ''
json_string_with_temp_credentials += '}'"

# Step 4. Make request to AWS federation endpoint to get sign-in token. Construct the parameter string with
# the sign-in action request, a 12-hour session duration, and the JSON document with temporary credentials
# as parameters.
request_parameters = "?Action=getSigninToken"
request_parameters += "&SessionDuration=43200"
request_parameters += "&Session=" + urllib.quote_plus(json_string_with_temp_credentials)
request_url = "https://signin.aws.amazon.com/federation" + request_parameters
r = requests.get(request_url)
# Returns a JSON document with a single element named SigninToken.
signin_token = json.loads(r.text)

# Step 5: Create URL where users can use the sign-in token to sign in to
# the console. This URL must be used within 15 minutes after the
# sign-in token was issued.
request_parameters = "?Action=login"
request_parameters += "&Issuer=Example.org"
request_parameters += "&Destination=" + urllib.quote_plus("https://console.aws.amazon.com/")
request_parameters += "&SigninToken=" + signin_token["SigninToken"]
request_url = "https://signin.aws.amazon.com/federation" + request_parameters

# Send final URL to stdout
print request_url
```

Java를 사용한 예제 코드

다음 예는 Java를 사용해 연동 사용자에게 AWS Management 콘솔에 대한 직접 액세스 권한을 부여하는 URL을 프로그래밍 방식으로 생성하는 방법을 보여줍니다. 다음 코드 조각은 [Java용 AWS SDK](#)를 사용합니다.

```
import java.net.URL;
import java.net.URLConnection;
import java.io.BufferedReader;
import java.io.InputStreamReader;
// Available at http://www.json.org/java/index.html
import org.json.JSONObject;
import com.amazonaws.auth.AWS Credentials;
import com.amazonaws.auth.BasicAWS Credentials;
import com.amazonaws.services.securitytoken.AWSSecurityTokenServiceClient;
import com.amazonaws.services.securitytoken.model.Credentials;
import com.amazonaws.services.securitytoken.model.GetFederationTokenRequest;
import com.amazonaws.services.securitytoken.model.GetFederationTokenResult;

/* Calls to AWS STS API operations must be signed using the access key ID
and secret access key of an IAM user or using existing temporary
credentials. The credentials should not be embedded in code. For
```

```
this example, the code looks for the credentials in a
standard configuration file.

/*
AWSCredentials credentials =
new PropertiesCredentials(
    AwsConsoleApp.class.getResourceAsStream("AwsCredentials.properties"));

AWSecurityTokenServiceClient stsClient =
new AWSecurityTokenServiceClient(credentials);

GetFederationTokenRequest getFederationTokenRequest =
    new GetFederationTokenRequest();
getFederationTokenRequest.setDurationSeconds(1800);
getFederationTokenRequest.setName("UserName");

// A sample policy for accessing Amazon Simple Notification Service (Amazon SNS) in the
// console.

String policy = "{\"Version\":\"2012-10-17\", \"Statement\":[{\"Action\":\"sns:*\", \" +
    "\"Effect\":\"Allow\", \"Resource\":\"*\"]}]";

getFederationTokenRequest.setPolicy(policy);

GetFederationTokenResult federationTokenResult =
    stsClient.getFederationToken(getFederationTokenRequest);

Credentials federatedCredentials = federationTokenResult.getCredentials();

// The issuer parameter specifies your internal sign-in
// page, for example https://mysignin.internal.mycompany.com/.
// The console parameter specifies the URL to the destination console of the
// AWS Management Console. This example goes to Amazon SNS.
// The signin parameter is the URL to send the request to.

String issuerURL = "https://mysignin.internal.mycompany.com/";
String consoleURL = "https://console.aws.amazon.com/sns";
String signInURL = "https://signin.aws.amazon.com/federation";

// Create the sign-in token using temporary credentials,
// including the access key ID, secret access key, and security token.
String sessionJson = String.format(
    "{\"%1$s\":\"%2$s\", \"%3$s\":\"%4$s\", \"%5$s\":\"%6$s\"}",
    "sessionId", federatedCredentials.getAccessKeyId(),
    "sessionKey", federatedCredentials.getSecretAccessKey(),
    "sessionToken", federatedCredentials.getSessionToken());

// Construct the sign-in request with the request sign-in token action, a
// 12-hour console session duration, and the JSON document with temporary
// credentials as parameters.

String getSigninTokenURL = signInURL +
    "?Action=getSigninToken" +
    "&DurationSeconds=43200" +
    "&SessionType=json&Session=" +
    URLEncoder.encode(sessionJson, "UTF-8");

URL url = new URL(getSigninTokenURL);

// Send the request to the AWS federation endpoint to get the sign-in token
URLConnection conn = url.openConnection ();

BufferedReader bufferReader = new BufferedReader(new
    InputStreamReader(conn.getInputStream()));
String returnContent = bufferReader.readLine();

String signinToken = new JSONObject(returnContent).getString("SigninToken");
```

```
String signinTokenParameter = "&SigninToken=" + URLEncoder.encode(signinToken, "UTF-8");

// The issuer parameter is optional, but recommended. Use it to direct users
// to your sign-in page when their session expires.

String issuerParameter = "&Issuer=" + URLEncoder.encode(issuerURL, "UTF-8");

// Finally, present the completed URL for the AWS console session to the user

String destinationParameter = "&Destination=" + URLEncoder.encode(consoleURL, "UTF-8");
String loginURL = signInURL + "?Action=login"
    + signinTokenParameter + issuerParameter + destinationParameter;
```

URL을 생성하는 방법을 보여주는 예(Ruby)

다음 예는 Ruby를 사용해 연동 사용자에게 AWS Management 콘솔에 대한 직접 액세스 권한을 부여하는 URL을 프로그래밍 방식으로 생성하는 방법을 보여줍니다. 이 코드 조각은 [Ruby용 AWS SDK](#)를 사용합니다.

```
require 'rubygems'
require 'json'
require 'open-uri'
require 'cgi'
require 'aws-sdk'

# Create a new STS instance
#
# Note: Calls to AWS STS API operations must be signed using an access key ID
# and secret access key. The credentials can be in EC2 instance metadata
# or in environment variables and will be automatically discovered by
# the default credentials provider in the AWS Ruby SDK.
sts = Aws::STS::Client.new()

# The following call creates a temporary session that returns
# temporary security credentials and a session token.
# The policy grants permissions to work
# in the AWS SNS console.

session = sts.get_federation_token({
  duration_seconds: 1800,
  name: "UserName",
  policy: "{\"Version\":\"2012-10-17\", \"Statement\":{\\\"Effect\\\":\\\"Allow\\\", \\\"Action\\\":\\\"sns:*\\\", \\\"Resource\\\":\\\"*\\\"}}",
})

# The issuer value is the URL where users are directed (such as
# to your internal sign-in page) when their session expires.
#
# The console value specifies the URL to the destination console.
# This example goes to the Amazon SNS console.
#
# The sign-in value is the URL of the AWS STS federation endpoint.
issuer_url = "https://mysignin.internal.mycompany.com/"
console_url = "https://console.aws.amazon.com/sns"
signin_url = "https://signin.aws.amazon.com/federation"

# Create a block of JSON that contains the temporary credentials
# (including the access key ID, secret access key, and session token).
session_json = {
  :sessionId => session.credentials[:access_key_id],
  :sessionKey => session.credentials[:secret_access_key],
  :sessionToken => session.credentials[:session_token]
}.to_json
```

```
# Call the federation endpoint, passing the parameters
# created earlier and the session information as a JSON block.
# The request returns a sign-in token that's valid for 15 minutes.
# Signing in to the console with the token creates a session
# that is valid for 12 hours.
get_signin_token_url = signin_url +
    "?Action=getSigninToken" +
    "&SessionType=json&Session=" +
    CGI.escape(session_json)

returned_content = URI.parse(get_signin_token_url).read

# Extract the sign-in token from the information returned
# by the federation endpoint.
signin_token = JSON.parse(returned_content)['SigninToken']
signin_token_param = "&SigninToken=" + CGI.escape(signin_token)

# Create the URL to give to the user, which includes the
# sign-in token and the URL of the console to open.
# The "issuer" parameter is optional but recommended.
issuer_param = "&Issuer=" + CGI.escape(issuer_url)
destination_param = "&Destination=" + CGI.escape(console_url)
login_url = signin_url + "?Action=login" + signin_token_param +
    issuer_param + destination_param
```

서비스 연결 역할 사용

서비스 연결 역할은 AWS 서비스에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 해당 서비스에서 사전 정의하며 서비스에서 다른 AWS 서비스를 자동으로 호출하기 위해 필요한 모든 권한을 포함합니다. 또한 연결된 서비스는 서비스 연결 역할을 만들고 수정하며 삭제하는 방법을 정의합니다. 서비스는 역할을 자동으로 만들거나 삭제할 수 있습니다. 서비스의 프로세스나 마법사를 사용하여 사용자가 역할을 만들거나 수정하거나 삭제하도록 허용할 수도 있습니다. 또는 사용자가 IAM을 사용하여 역할을 만들거나 삭제하도록 요구할 수도 있습니다. 서비스 연결 역할은 그 방법에 상관없이 사용자 대신 작업을 완료하는 데 필요한 권한을 수동으로 추가할 필요가 없기 때문에 설정이 쉬워집니다.

연결된 서비스에서 서비스 연결 역할 권한을 정의하므로 정의되지 않은 경우에만 해당 역할로 서비스를 수행할 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며, 이 권한 정책은 다른 IAM 개체에 연결할 수 없습니다.

먼저 역할의 관련 리소스를 삭제해야만 역할을 삭제할 수 있습니다. 이렇게 하면 리소스에 대한 액세스 권한을 부주의로 삭제할 수 없기 때문에 리소스가 보호됩니다.

Tip

서비스 연결 역할의 사용을 지원하는 서비스에 대한 자세한 정보는 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#)를 참조하고 서비스 연결 역할 열에 예가 있는 서비스를 찾으십시오. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 [Yes] 링크를 선택합니다.

서비스 연결 역할 권한

사용자 또는 역할이 서비스 연결 역할을 작성하거나 편집할 수 있도록 IAM 개체(사용자, 그룹, 역할 등)의 권한을 구성해야 합니다.

Note

서비스 링크된 역할에 대한 ARN은 정책에서 **SERVICE-NAME**.amazonaws.com으로 나타내지는 서비스 보안 주체를 포함합니다. 각 경우마다 다르고 AWS 서비스에 따라 형식이 다양하기 때문에 서비스 보안 주체를 알기 어렵습니다. 서비스의 보안 주체를 보려면 해당 서비스 링크된 역할 설명서 단원을 참조하십시오.

IAM 개체가 특정 서비스 연결 역할을 만들 수 있도록 허용하려면

서비스 연결 역할을 생성해야 하는 IAM 개체에 다음 정책을 추가합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iam:CreateServiceLinkedRole",  
            "Resource": "arn:aws:iam::*:role/aws-service-role/SERVICE-NAME.amazonaws.com/SERVICE-LINKED-ROLE-NAME-PREFIX*",  
            "Condition": {"StringLike": {"iam:AWSPropertyName": "SERVICE-NAME.amazonaws.com"}}  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iam:AttachRolePolicy",  
                "iam:PutRolePolicy"  
            ],  
            "Resource": "arn:aws:iam::*:role/aws-service-role/SERVICE-NAME.amazonaws.com/SERVICE-LINKED-ROLE-NAME-PREFIX*"  
        }  
    ]  
}
```

IAM 개체가 서비스 연결 역할을 만들 수 있도록 허용하려면

서비스 연결 역할 또는 필요한 정책을 포함해야 하는 모든 서비스 역할을 생성해야 하는 IAM 개체의 권한 정책에 다음 명령문을 추가합니다. 이 정책 명령문은 IAM 개체가 역할에 정책을 연결하는 것을 허용하지 않습니다.

```
{  
    "Effect": "Allow",  
    "Action": "iam:CreateServiceLinkedRole",  
    "Resource": "arn:aws:iam::*:role/aws-service-role/*"  
}
```

IAM 개체가 서비스 역할의 설명을 편집할 수 있도록 허용하려면

서비스 연결 역할 또는 서비스 역할의 설명을 편집해야 하는 IAM 개체의 권한 정책에 다음 명령문을 추가합니다.

```
{  
    "Effect": "Allow",  
    "Action": "iam:UpdateRoleDescription",  
    "Resource": "arn:aws:iam::*:role/aws-service-role/*"  
}
```

IAM 개체가 특정 서비스 연결 역할을 삭제하도록 허용하려면

서비스 연결 역할을 삭제해야 하는 IAM 개체의 권한 정책에 다음 문장을 추가합니다.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iam>DeleteServiceLinkedRole",  
        "iam:GetServiceLinkedRoleDeletionStatus"  
    ],  
    "Resource": "arn:aws:iam::*:role/aws-service-role/SERVICE-NAME.amazonaws.com/SERVICE-LINKED-ROLE-NAME-PREFIX*"  
}
```

}

IAM 개체가 서비스 연결 역할을 삭제할 수 있도록 허용하려면

서비스 연결 역할만 삭제하고 서비스 역할은 삭제하지 않는 IAM 개체의 권한 정책에 다음 명령문을 추가합니다.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iam:DeleteServiceLinkedRole",  
        "iam:GetServiceLinkedRoleDeletionStatus"  
    ],  
    "Resource": "arn:aws:iam::*:role/aws-service-role/*"  
}
```

IAM 엔터티가 기존 역할을 서비스에 전달하도록 허용하는 방법

일부 AWS 서비스를 사용하면 새 서비스에 연결된 역할을 생성하지 않고, 그 대신에 서비스에 기존 역할을 전달할 수 있습니다. 이렇게 하려면 사용자에게 서비스에 역할을 전달할 수 있는 권한이 있어야 합니다. 역할을 생성해야 하는 IAM 개체의 권한 정책에 다음 명령문을 추가합니다. 또한 이 정책 설명에서는 엔터티가 전달할 역할을 선택할 수 있는 역할 목록을 볼 수 있도록 허용합니다. 자세한 정보는 [사용자에게 AWS 서비스에 역할을 전달할 권한 부여](#) (p. 231) 단원을 참조하십시오.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iam>ListRoles",  
        "iam:PassRole"  
    ],  
    "Resource": "arn:aws:iam::123456789012:role/my-role-for-XYZ"  
}
```

서비스 연결 역할 만들기

서비스 연결 역할을 만드는 데 사용하는 방법은 서비스에 따라 다릅니다. 경우에 따라 서비스 연결 역할을 수동으로 만들 필요가 없습니다. 예를 들어 사용자가 서비스의 특정 작업(리소스 만들기 등)을 수행할 때 서비스가 서비스 연결 역할을 자동으로 생성할 수 있습니다. 또한, 서비스가 서비스 연결 역할 지원을 시작하기 전에 서비스를 사용한 경우에는 서비스가 자동으로 해당 계정에 역할을 생성했을 수 있습니다. 자세히 알아보려면 [내 AWS 계정에 표시되는 새 역할](#) (p. 470) 단원을 참조하십시오.

다른 경우에는 서비스에서 서비스 콘솔, API 또는 CLI를 사용하여 서비스 연결 역할을 수동으로 만들도록 허용할 수 있습니다. 서비스 연결 역할의 사용을 지원하는 서비스에 대한 자세한 정보는 [IAM로 작업하는 AWS 서비스](#) (p. 488)를 참조하고 서비스 연결 역할 열에 예가 있는 서비스를 찾으십시오. 서비스가 서비스 연결 역할 생성을 지원하는지 여부를 알아보려면 예 링크를 선택하여 해당 서비스의 서비스 연결 역할 설명서 단원을 참조하십시오.

서비스가 역할 만들기를 지원하지 않는 경우에는 IAM을 사용하여 서비스 연결 역할을 만들 수 있습니다.

Important

서비스 연결 역할은 [AWS 계정의 IAM 역할 제한](#)을 계산하는 데 포함되지만, 한도에 도달한 경우에도 계정에 서비스 연결 역할을 만들 수 있습니다. 한도를 초과해도 생성할 수 있는 역할은 서비스 연결 역할뿐입니다.

서비스 연결 역할 만들기(콘솔)

IAM에서 서비스 연결 역할을 만들기 전에, 연결된 서비스가 서비스 역할을 자동으로 생성하는지 확인하십시오. 또한 서비스의 콘솔, API, CLI 등에서 역할을 만들 수 있는지를 알아봅니다.

서비스 연결 역할을 만들려면(콘솔 사용)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 [Roles]를 선택합니다. 그런 다음 [Create role]을 선택합니다.
3. AWS 서비스 역할 유형을 선택한 후 이 역할로 수행하도록 허용하려는 서비스를 선택합니다.
4. 서비스의 사용 사례를 선택합니다. 지정한 서비스에 사용 사례가 하나뿐이면 자동으로 선택됩니다. 사용 사례는 서비스에 필요한 신뢰 정책을 포함하기 위해 서비스에서 정합니다. 그런 다음 [Next: Permissions]를 선택합니다.
5. 하나 이상의 권한 정책을 선택하여 역할에 연결합니다. 선택한 사용 사례에 따라 서비스에서 다음을 수행할 수 있습니다.
 - 역할이 사용하는 권한 정의
 - 제한된 권한 집합에서 선택할 수 있도록 허용
 - 모든 권한 집합에서 선택할 수 있도록 허용
 - 여기서 정책을 선택하지 않고, 나중에 정책을 만들어 역할에 연결할 수 있도록 허용합니다.

역할에 부여하려는 권한을 할당하는 정책 옆의 확인란을 선택한 후 다음: 태그 지정을 선택합니다.

Note

지정하는 권한은 역할을 사용하는 모든 주체가 사용할 수 있습니다. 기본적으로 역할은 권한이 없습니다.

6. [Next: Review]를 선택합니다. 생성하는 동안에는 서비스 연결 역할에 태그를 연결할 수 없습니다. IAM에서의 태그 사용에 대한 자세한 정보는 [IAM 엔터티 태그 지정 \(p. 259\)](#) 단원을 참조하십시오.
7. 역할 이름의 경우 역할 이름 사용자 지정 수준은 서비스에서 정합니다. 서비스에서 역할 이름을 정한 경우 이 옵션을 편집할 수 없습니다. 다른 경우에는 서비스에서 역할 이름의 접두사를 정의하고 사용자가 선택적으로 접미부를 입력하도록 할 수 있습니다.

가능한 경우 기본 이름에 추가할 역할 이름 접미사를 입력합니다. 이 접미사는 이 역할의 목적을 파악하는데 도움이 됩니다. 역할 이름은 AWS 계정 내에서 고유해야 합니다. 대소문자는 구별하지 않습니다. 예를 들어, 이름이 <service-linked-role-name>_SAMPLE과 <service-linked-role-name>_sample, 두 가지로 지정된 역할을 만들 수는 없습니다. 다양한 주체가 역할을 참조할 수 있기 때문에 역할이 생성된 후에는 역할 이름을 편집할 수 없습니다.
8. (선택 사항) [Role description]에서 새로운 서비스 연결 역할에 대한 설명을 편집합니다.
9. 역할을 검토한 다음 [Create role]을 선택합니다.

서비스 연결 역할 만들기(AWS CLI)

IAM에서 서비스 연결 역할을 만들기 전에, 연결된 서비스가 서비스 역할을 자동으로 생성하는지 그리고 서비스의 CLI에서 사용자가 역할을 만들 수 있는지를 확인하십시오. 서비스 CLI가 지원되지 않는 경우 IAM 명령을 사용하여 서비스가 역할을 위임하는 데 필요한 인라인 정책과 신뢰 정책을 포함하는 서비스 연결 역할을 만들 수 있습니다.

서비스 연결 역할(AWS CLI)을 만들려면

다음 명령을 실행합니다.

```
$ aws iam create-service-linked-role --aws-service-name SERVICE-NAME.amazonaws.com
```

서비스 연결 역할 만들기(AWS API)

IAM에서 서비스 연결 역할을 만들기 전에, 연결된 서비스가 서비스 역할을 자동으로 생성하는지 그리고 서비스의 API에서 사용자가 역할을 만들 수 있는지를 확인하십시오. 서비스 API가 지원되지 않는 경우 AWS

API를 사용하여 서비스가 역할을 위임하는 데 필요한 인라인 정책과 신뢰 정책을 포함하는 서비스 연결 역할을 만들 수 있습니다.

서비스 연결 역할(AWS API)을 만들려면

[CreateServiceLinkedRole](#) API 호출을 사용합니다. 요청 시 `SERVICE_NAME_URL.amazonaws.com` 서비스 이름을 지정합니다.

예를 들어 Lex 봇 서비스 연결 역할을 만들려면 `lex.amazonaws.com`을 사용합니다.

서비스 연결 역할 편집

서비스 연결 역할을 편집하는 데 사용하는 방법은 서비스에 따라 다릅니다. 일부 서비스는 사용자가 서비스 콘솔, API 또는 CLI에서 서비스 연결 역할의 권한을 편집할 수 있도록 허용합니다. 하지만 서비스 연결 역할을 만든 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. IAM 콘솔, API, CLI에서 역할의 설명을 편집할 수 있습니다.

서비스 연결 역할의 사용을 지원하는 서비스에 대한 자세한 정보는 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#)를 참조하고 서비스 연결 역할 열에 예가 있는 서비스를 찾으십시오. 서비스가 서비스 연결 역할 편집을 지원하는지 여부를 알아보려면 예 링크를 선택하여 해당 서비스의 서비스 연결 역할 설명서 단원을 참조하십시오.

서비스 연결 역할 설명 편집(콘솔 사용)

IAM 콘솔을 사용하여 서비스 연결 역할의 설명을 편집할 수 있습니다.

서비스 연결 역할의 설명을 편집하려면(콘솔 사용)

1. IAM 콘솔의 탐색 창에서 역할을 선택합니다.
2. 변경할 역할 이름을 선택합니다.
3. Role description(역할 설명)의 맨 오른쪽에서 편집을 선택합니다.
4. 상자에 새 설명을 입력하고 저장을 선택합니다.

서비스 연결 역할 설명 편집(AWS CLI)

AWS CLI에서 IAM 명령을 사용하여 서비스 연결 역할의 설명을 편집할 수 있습니다.

서비스 연결 역할의 설명을 변경하려면(AWS CLI)

1. (옵션) 역할의 현재 설명을 보려면 다음 명령 중 하나를 실행합니다.

```
$ aws iam get-role --role-name ROLE-NAME
```

CLI 명령에서 역할을 참조하려면 ARN이 아니라 역할 이름을 사용해야 합니다. 예를 들어, 어떤 역할의 ARN이 `arn:aws:iam::123456789012:role/myrole`인 경우 참조할 역할은 `myrole`입니다.

2. 서비스 연결 역할의 설명을 업데이트하려면 다음 명령을 실행합니다.

```
$ aws iam update-role --role-name ROLE-NAME --description OPTIONAL-DESCRIPTION
```

서비스 연결 역할 설명 편집(AWS API)

AWS API를 사용하여 서비스 연결 역할의 설명을 편집할 수 있습니다.

서비스 연결 역할의 설명을 변경하려면(AWS API 사용)

1. (옵션) 역할의 현재 설명을 보려면 다음 작업을 호출하고 역할 이름을 지정합니다.

AWS API: [GetRole](#)

- 역할의 설명을 업데이트하려면 다음 작업을 호출하고 역할 이름 및 설명(선택 사항)을 지정합니다.

AWS API: [UpdateRole](#)

서비스 연결 역할 삭제

서비스 연결 역할을 만드는 데 사용하는 방법은 서비스에 따라 다릅니다. 일부 경우에는 서비스 연결 역할을 수동으로 삭제할 필요가 없습니다. 예를 들어, 서비스에서 특정 작업(예: 리소스 제거)을 완료하면 서비스에서 사용자의 서비스 연결 역할을 삭제할 수 있습니다.

서비스에서 서비스 연결 역할을 서비스 콘솔, API 또는 CLI에서 수동으로 삭제하는 것이 지원되지 않는 경우도 있을 수 있습니다.

서비스 연결 역할의 사용을 지원하는 서비스에 대한 자세한 정보는 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#)를 참조하고 서비스 연결 역할 열에 예가 있는 서비스를 찾으십시오. 서비스에서 서비스 연결 역할 삭제를 지원하는지 확인하려면 링크를 선택하여 해당 서비스의 서비스 연결 역할 문서를 확인하십시오.

서비스에서 역할 삭제를 지원하지 않는 경우에는 사용자가 IAM 콘솔, API 또는 CLI에서 서비스 연결 역할을 삭제할 수 있습니다. 서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제할 것을 권합니다. 따라서 적극적으로 모니터링하거나 유지하지 않는 미사용 개체가 없도록 합니다. 단, 삭제 전에 서비스 연결 역할을 정리해야 합니다.

서비스 연결 역할 정리

IAM을 사용하여 서비스 연결 역할을 삭제하기 전에 먼저 역할에 활성 세션이 있는지 확인하고 역할에서 사용되는 리소스를 모두 제거해야 합니다.

IAM 콘솔에서 서비스 연결 역할에 활성 세션이 있는지 확인하려면

- AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
- IAM 콘솔의 탐색 창에서 [Roles]를 선택합니다. 그런 다음 서비스 연결 역할의 이름(확인란 아님)을 선택합니다.
- 선택한 역할의 [Summary] 페이지에서 [Access Advisor] 탭을 선택합니다.
- [Access Advisor] 탭에서 서비스 연결 역할의 최근 활동을 검토합니다.

Note

서비스에서 서비스 연결 역할을 사용하는지 잘 모를 경우에는 역할을 삭제해보십시오. 서비스에서 역할을 사용하는 경우에는 삭제가 안 되어 역할이 사용 중인 리전을 볼 수 있습니다. 역할이 사용 중인 경우에는 세션이 종료될 때까지 기다렸다가 역할을 삭제해야 합니다. 서비스 연결 역할에 대한 세션은 취소할 수 없습니다.

서비스 연결 역할에서 사용하는 리소스를 제거하려면

서비스 연결 역할의 사용을 지원하는 서비스에 대한 자세한 정보는 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#)를 참조하고 서비스 연결 역할 열에 예가 있는 서비스를 찾으십시오. 서비스에서 서비스 연결 역할 삭제를 지원하는지 확인하려면 링크를 선택하여 해당 서비스의 서비스 연결 역할 문서를 확인하십시오. 사용자의 서비스 연결 역할에서 사용되는 리소스를 제거하는 방법은 해당 서비스의 문서 단원을 참조하십시오.

서비스 연결 역할(콘솔) 삭제

IAM 콘솔을 사용하여 서비스 연결 역할을 삭제할 수 있습니다.

서비스 연결 역할을 삭제하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 [Roles]를 선택합니다. 그런 다음 삭제할 역할의 이름이나 행이 아닌 이름 옆에 있는 확인란을 선택합니다.
3. 페이지 상단의 [Role actions]에서 [Delete role]를 선택합니다.
4. 확인 대화 상자가 나타나면 서비스 마지막 액세스 데이터를 검토합니다. 이 데이터는 선택한 각 역할이 AWS 서비스를 마지막으로 액세스한 일시를 보여 줍니다. 이를 통해 역할이 현재 활동 중인지 여부를 확인할 수 있습니다. 계속 진행하려면 [Yes, Delete]를 선택하여 삭제할 서비스 연결 역할을 제출합니다.
5. IAM 알림을 보고 서비스 연결 역할 삭제 진행 상황을 모니터링합니다. IAM 서비스 연결 역할 삭제는 비동기이므로 삭제할 역할을 제출한 후에 삭제 작업이 성공하거나 실패할 수 있습니다.
 - 작업에 성공하면 목록에서 역할이 제거되고 성공 알림이 페이지 상단에 나타납니다.
 - 작업에 실패할 경우 알림의 [View details] 또는 [View Resources]를 선택하면 삭제 실패 이유를 확인할 수 있습니다. 역할에서 서비스 리소스를 사용 중이어서 삭제에 실패한 경우에는 알림에 리소스 목록이 포함됩니다(서비스에서 해당 정보를 반환할 경우). 이후 [리소스를 정리하고 \(p. 200\)](#) 삭제를 다시 제출할 수 있습니다.

Note

서비스에서 반환하는 정보에 따라 이 과정을 여러 번 반복해야 할 수 있습니다. 예를 들어, 서비스 연결 역할에서 6개의 리소스를 사용할 수 있으며, 서비스에서 이중 5개에 관한 정보를 반환할 수 있습니다. 5개 리소스를 정리하고 삭제할 역할을 다시 제출할 경우 삭제에 실패하고 서비스에서 나머지 1개의 리소스를 보고합니다. 서비스에서 리소스 전부를 반환하거나, 일부만 반환하거나, 리소스를 보고하지 않을 수 있습니다.

- 작업에 실패했는데 알림에 리소스 목록이 포함되지 않을 경우에는 서비스에서 해당 정보를 반환하지 않을 수 있습니다. 해당 서비스의 리소스를 정리하는 방법은 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하십시오. 표에서 서비스를 확인하고 예 링크를 선택하여 해당 서비스의 서비스 연결 역할 문서를 확인합니다.

서비스 연결 역할 삭제(AWS CLI)

AWS CLI에서 IAM 명령을 사용하여 서비스 연결 역할을 삭제할 수 있습니다.

서비스 연결 역할을 삭제하려면(AWS CLI)

1. 삭제할 서비스 연결 역할의 이름을 모를 경우에는 다음 명령을 입력하여 계정에 역할과 Amazon 리소스 이름(ARN)을 나열합니다.

```
$ aws iam get-role --role-name role-name
```

CLI 명령에서 역할을 참조하려면 ARN이 아니라 역할 이름을 사용해야 합니다. 예를 들어, 어떤 역할의 ARN이 arn:aws:iam::123456789012:role/myrole인 경우 참조할 역할은 myrole입니다.

2. 서비스 연결 역할이 사용되지 않거나 연결된 리소스가 없는 경우에는 서비스 연결 역할을 삭제할 수 없으므로 삭제 요청을 제출해야 합니다. 이러한 조건이 충족되지 않으면 요청이 거부될 수 있습니다. 삭제 작업 상태를 확인하려면 응답의 deletion-task-id를 캡처해야 합니다. 다음 명령을 입력하여 서비스 연결 역할 삭제 요청을 제출합니다.

```
$ aws iam delete-service-linked-role --role-name role-name
```

3. 다음 명령을 입력하여 삭제 작업의 상태를 확인합니다.

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

삭제 작업은 NOT_STARTED, IN_PROGRESS, SUCCEEDED 또는 FAILED 상태일 수 있습니다. 삭제에 실패할 경우 문제를 해결할 수 있도록 실패 이유가 호출에 반환됩니다. 역할에서 서비스 리소스를 사용 중이어서 삭제에 실패한 경우에는 알림에 리소스 목록이 포함됩니다(서비스에서 해당 정보를 반환할 경우). 이후 [리소스를 정리하고 \(p. 200\)](#) 삭제를 다시 제출할 수 있습니다.

Note

서비스에서 반환하는 정보에 따라 이 과정을 여러 번 반복해야 할 수 있습니다. 예를 들어, 서비스 연결 역할에서 6개의 리소스를 사용할 수 있으며, 서비스에서 이중 5개에 관한 정보를 반환할 수 있습니다. 5개 리소스를 정리하고 삭제할 역할을 다시 제출할 경우 삭제에 실패하고 서비스에서 나머지 1개의 리소스를 보고합니다. 서비스에서 리소스 전부를 반환하거나, 일부만 반환하거나, 리소스를 보고하지 않을 수 있습니다. 리소스를 보고하지 않는 서비스의 리소스를 정리하는 방법은 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하십시오. 표에서 서비스를 확인하고 링크를 선택하여 해당 서비스의 서비스 연결 역할 문서를 확인합니다.

서비스 연결 역할 삭제(AWS API)

AWS API를 사용하여 서비스 연결 역할을 삭제할 수 있습니다.

서비스 연결 역할을 삭제하려면(AWS API)

1. 서비스 연결 역할 삭제 요청을 제출하려면 [DeleteServiceLinkedRole](#)을 호출합니다. 요청에 역할 이름을 지정합니다.

서비스 연결 역할이 사용되지 않거나 연결된 리소스가 없는 경우에는 서비스 연결 역할을 삭제할 수 없으므로 삭제 요청을 제출해야 합니다. 이러한 조건이 충족되지 않으면 요청이 거부될 수 있습니다. 삭제 작업 상태를 확인하려면 응답의 [DeletionTaskId](#)를 캡처해야 합니다.

2. 삭제 상태를 확인하려면 [GetServiceLinkedRoleDeletionStatus](#)를 호출합니다. 요청에 [DeletionTaskId](#)를 지정합니다.

삭제 작업은 NOT_STARTED, IN_PROGRESS, SUCCEEDED 또는 FAILED 상태일 수 있습니다. 삭제에 실패할 경우 문제를 해결할 수 있도록 실패 이유가 호출에 반환됩니다. 역할에서 서비스 리소스를 사용 중이어서 삭제에 실패한 경우에는 알림에 리소스 목록이 포함됩니다(서비스에서 해당 정보를 반환할 경우). 이후 [리소스를 정리하고 \(p. 200\)](#) 삭제를 다시 제출할 수 있습니다.

Note

서비스에서 반환하는 정보에 따라 이 과정을 여러 번 반복해야 할 수 있습니다. 예를 들어, 서비스 연결 역할에서 6개의 리소스를 사용할 수 있으며, 서비스에서 이중 5개에 관한 정보를 반환할 수 있습니다. 5개 리소스를 정리하고 삭제할 역할을 다시 제출할 경우 삭제에 실패하고 서비스에서 나머지 1개의 리소스를 보고합니다. 서비스에서 리소스 전부를 반환하거나, 일부만 반환하거나, 리소스를 보고하지 않을 수 있습니다. 리소스를 보고하지 않는 서비스의 리소스를 정리하는 방법은 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하십시오. 표에서 서비스를 확인하고 링크를 선택하여 해당 서비스의 서비스 연결 역할 문서를 확인합니다.

IAM 역할 생성

역할을 생성하기 위해서는 AWS Management 콘솔, AWS CLI, Windows PowerShell용 도구 또는 IAM API를 사용할 수 있습니다.

AWS Management 콘솔을 사용하는 경우 마법사가 역할 생성 절차를 단계별로 안내합니다. 마법사의 진행 단계는 생성하는 역할 대상이 AWS 서비스일 때, AWS 계정일 때, 혹은 연동 사용자일 때에 따라 약간 다릅니다.

주제

- [역할을 만들어 IAM 사용자에게 권한 위임 \(p. 203\)](#)

- AWS 서비스에 대한 권한을 위임할 역할 생성 (p. 210)
- 타사 자격 증명 공급자의 역할 만들기(연동) (p. 215)
- 액세스 권한 위임을 위한 정책의 예 (p. 223)

역할을 만들어 IAM 사용자에게 권한 위임

IAM 역할을 사용해 AWS 리소스에 대한 액세스 권한을 위임할 수 있습니다. IAM 역할을 사용해 신뢰하는 계정과 다른 AWS 신뢰 받는 계정 간에 신뢰 관계를 설정할 수 있습니다. 신뢰하는 계정은 액세스되는 리소스를 소유하고 신뢰받는 계정은 리소스에 대한 액세스가 필요한 사용자를 저장합니다. 그러나, 다른 계정이 해당 계정의 리소스를 소유할 수 있는 가능성성이 있습니다. 예를 들어, 신뢰받는 계정은 신뢰 계정이 Amazon S3 버킷의 새로운 객체를 생성하는 것처럼 새로운 리소스를 생성하도록 허용할 수 있습니다. 이러한 경우, 리소스를 생성하는 계정은 리소스를 소유하고 누구에게 리소스에 대한 액세스를 부여할지 제어합니다.

신뢰 관계를 생성한 후 IAM 사용자 또는 신뢰받는 계정의 애플리케이션은 AWS Security Token Service(AWS STS) [AssumeRole](#) API 작업을 사용할 수 있습니다. 이 작업은 계정의 AWS 리소스에 액세스 할 수 있는 임시 보안 자격 증명을 제공합니다.

계정은 둘 다 직접 제어할 수 있거나 사용자가 속한 계정의 경우 타사가 제어할 수 있습니다. 사용자가 있는 다른 계정이 귀하가 제어하지 않는 AWS 계정에 있는 경우 externalId 속성을 사용할 수 있습니다. 외부 ID는 나와 타사 계정의 관리자 간에 합의한 숫자 또는 단어가 될 수 있습니다. 이 옵션은 요청에 올바른 sts:ExternalID가 포함된 경우에만 사용자가 역할을 맡을 수 있도록 허용하는 조건을 신뢰 정책에 자동으로 추가합니다. 자세한 내용은 [AWS 리소스에 대한 액세스를 타사에 부여할 때 외부 ID를 사용하는 방법 \(p. 206\)](#) 단원을 참조하십시오.

역할을 사용해 권한을 위임하는 방법에 대한 자세한 내용은 [역할 용어 및 개념 \(p. 153\)](#) 단원을 참조하십시오. 서비스 연결을 사용하여 서비스가 해당 계정의 리소스에 액세스할 수 있도록 허용하는 방법은 [AWS 서비스에 대한 권한을 위임할 역할 생성 \(p. 210\)](#) 단원을 참조하십시오.

IAM 역할 만들기(콘솔 사용)

AWS Management 콘솔을 사용해 IAM 사용자가 수입할 수 있는 역할을 만들 수 있습니다. 예를 들면 프로덕션 환경에서 개발 환경을 격리하기 위해 조직이 여러 개의 AWS 계정을 갖고 있다고 가정합시다. 개발 계정의 사용자가 프로덕션 계정의 리소스에 액세스하도록 허용하는 역할을 설정하고 사용하는데 필요한 단계에 대한 자세한 설명을 보려면, [분리된 개발 및 프로덕션 계정을 사용한 예제 시나리오 \(p. 157\)](#) 단원을 참조하십시오.

역할을 만들려면(콘솔 사용)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 콘솔의 탐색 창에서 역할을 선택한 후 역할 생성을 선택합니다.
3. 다른 AWS 계정 역할 유형을 선택합니다.
4. 계정 ID에 리소스에 대한 액세스 권한을 부여하려는 AWS 계정 ID를 입력합니다.

지정된 계정의 관리자는 해당 계정의 IAM 사용자에게 이 역할을 맡을 수 있는 권한을 부여할 수 있습니다. 이를 위해 관리자는 sts:AssumeRole 작업에 대한 권한을 부여하는 정책을 사용자나 그룹에 연결합니다. 이 정책은 역할의 ARN을 Resource로 지정해야 합니다.

5. 통제권이 없는 계정의 사용자에게 권한을 부여하려면 사용자는 이 역할을 프로그래밍 방식으로 가정하고 Require external ID(외부 ID 필요)를 선택합니다. 외부 ID는 나와 타사 계정의 관리자 간에 합의한 숫자 또는 단어가 될 수 있습니다. 이 옵션은 요청에 올바른 sts:ExternalID가 포함된 경우에만 사용자가 역할을 맡을 수 있도록 허용하는 조건을 신뢰 정책에 자동으로 추가합니다. 자세한 내용은 [AWS 리소스에 대한 액세스를 타사에 부여할 때 외부 ID를 사용하는 방법 \(p. 206\)](#) 단원을 참조하십시오.

Important

이 옵션을 선택하면 AWS CLI, Windows PowerShell용 도구 또는 AWS API를 통해 이 역할로만 액세스가 제한됩니다. 이는 AWS 콘솔을 사용해 해당 신뢰 정책에 externalId 조건이 있

는 역할로 전환할 수 없기 때문입니다. 하지만 관련 SDK를 통해 스크립트나 애플리케이션을 작성하여 프로그래밍 방식으로 이러한 종류의 액세스를 만들 수 있습니다. 자세한 내용 및 샘플 스크립트는 AWS 보안 블로그의 [AWS Management 콘솔에 대한 교차 계정 액세스를 가능하게 하는 방법](#) 단원을 참조하십시오.

6. 멀티 팩터 인증(MFA)으로 로그인하는 사용자로 역할을 제한하려면, Require MFA(MFA 필요)를 선택합니다. 이렇게 하면 MFA 로그인을 확인하는 역할의 신뢰 정책에 조건이 추가됩니다. 역할을 맡으려는 사용자는 구성된 MFA 디바이스에서 임시 일회용 암호로 로그인해야 합니다. MFA 인증을 사용하지 않는 사용자는 역할을 맡을 수 없습니다. MFA에 대한 자세한 내용은 [AWS에서 멀티 팩터 인증\(MFA\) 사용하기](#) (p. 96) 단원을 참조하십시오.
7. Next: Permissions(다음: 권한)을 선택하십시오.
8. IAM은 계정의 AWS 관리형 또는 사용자 관리형 정책 목록을 포함합니다. 권한 정책을 사용하기 위한 정책을 선택하거나 정책 생성을 선택하여 새 브라우저 탭을 열고 완전히 새로운 정책을 생성합니다. 자세한 내용은 [IAM 정책 만들기\(콘솔\)](#) (p. 378) 절차의 4단계 단원을 참조하십시오. 정책을 생성하면 탭을 닫고 원래 탭으로 돌아갑니다. 누구든지 역할에게 위임하려는 권한 정책 옆의 확인란을 선택합니다. 원할 경우, 여기서 정책을 선택하지 않고 나중에 정책을 만들어서 역할에 연결할 수 있습니다. 기본적으로 역할은 권한이 없습니다.
9. (선택 사항) [권한 경계](#) (p. 317)로서 설정됨. 이는 고급 기능입니다.

Set permissions boundary(권한 경계 설정) 섹션을 열고 Use a permissions boundary to control the maximum role permissions(최대 역할 권한을 관리하기 위한 권한 경계 사용)을 선택합니다. 정책을 선택하여 권한 경계를 사용하십시오.

10. 다음: 태그 지정을 선택합니다.
11. (선택 사항) 태그를 키-값 페어로 연결하여 메타데이터를 역할에 추가합니다. IAM에서의 태그 사용에 대한 자세한 내용은 [IAM 엔터티 태그 지정](#) (p. 259) 단원을 참조하십시오.
12. [Next: Review]를 선택합니다.
13. Role name에 역할의 이름을 입력합니다. 역할 이름은 AWS 계정 내에서 고유해야 합니다. 대소문자는 구별하지 않습니다. 예를 들어, 이름이 **PRODROLE**과 **prodrole**, 두 가지로 지정된 역할을 만들 수는 없습니다. 기타 AWS 리소스가 역할을 참조할 수 있기 때문에 역할이 생성된 후에는 역할 이름을 편집할 수 없습니다.
14. (선택 사항) [Role description]에 새 역할에 대한 설명을 입력합니다.
15. 역할을 검토한 다음 [Create role]을 선택합니다.

Important

필요한 구성의 절반이 끝났습니다. 이제 신뢰할 수 있는 계정의 개별 사용자에게 콘솔의 역할로 전환하거나 역할을 프로그래밍 방식으로 위임할 수 있는 권한을 부여해야 합니다. 이 단계에 대한 자세한 내용은 [사용자에 대한 역할 전환 권한 부여](#) (p. 229) 단원을 참조하십시오.

IAM 역할 생성(AWS CLI)

AWS CLI에서 역할을 만들려면 여러 단계를 거쳐야 합니다. 콘솔을 사용하여 역할을 만들 때는 많은 단계가 자동으로 수행되지만 AWS CLI를 사용하면 각 단계를 직접 명시적으로 수행해야 합니다. 역할을 만든 다음 권한 정책을 역할에 할당해야 합니다. 선택적으로 역할에 대한 [권한 경계](#) (p. 317)을 설정할 수 있습니다.

교차 계정 액세스에 대한 역할을 만들려면(AWS CLI)

1. 역할 생성: [aws iam create-role](#)
2. 역할에 관리형 권한 정책 연결: [aws iam attach-role-policy](#)

또는

역할을 위한 인라인 권한 정책 생성: [aws iam put-role-policy](#)

3. (선택 사항) 태그를 연결하여 사용자 지정 속성을 역할에 추가: [aws iam tag-role](#)

자세한 내용은 [IAM 엔터티에 대한 태그 관리\(AWS CLI 또는 AWS API\)](#) (p. 262) 단원을 참조하십시오.

4. (선택 사항) 역할([aws iam put-role-permissions-boundary](#))에 대한 [권한 경계\(p. 317\)](#)를 설정합니다.

이 권한 경계는 역할이 가질 수 있는 최대 권한을 관리합니다. 권한 경계는 고급 AWS 기능입니다.

다음 예는 단순한 환경에서 교차 계정 역할을 생성하는 가장 일반적인 단계 중 첫 두 단계를 보여줍니다. 이 예제는 123456789012 계정에 있는 모든 사용자가 역할을 가정하고 example_bucket Amazon S3 버킷을 볼 수 있도록 허용합니다. 이 예제에서도 Windows가 구동되는 클라이언트 컴퓨터를 사용 중이며 명령줄 인터페이스를 계정 자격 증명 및 리전으로 이미 구성했다고 가정합니다. 자세한 내용은 [AWS 명령줄 인터페이스 구성](#) 단원을 참조하십시오.

이 예제는 역할을 생성할 경우 첫 번째 명령의 다음 신뢰 정책을 포함합니다. 이 신뢰 정책은 123456789012 계정에서 사용자가 AssumeRole 작업을 사용하여 역할을 가정할 수 있도록 허용합니다. 단, 사용자가 SerialNumber 및 TokenCode 파라미터를 사용하는 MFA 인증을 제공하는 경우에만 허용합니다. MFA에 대한 자세한 내용은 [AWS에서 멀티 팩터 인증\(MFA\) 사용하기\(p. 96\)](#) 단원을 참조하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": { "AWS": "arn:aws:iam::123456789012:root" },  
            "Action": "sts:AssumeRole",  
            "Condition": { "Bool": { "aws:MultiFactorAuthPresent": "true" } }  
        }  
    ]  
}
```

Important

Principal 요소에 특정 IAM 역할 또는 사용자에 대한 ARN이 포함되어 있으면, 정책을 저장할 때 해당 ARN이 고유 보안 주체 ID로 변환됩니다. 그러면 누군가가 해당 역할 또는 사용자를 제거하고 다시 만들어 본인의 권한을 에스컬레이션할 위험을 완화할 수 있습니다. 일반적으로 콘솔에서는 이 ID가 보이지 않습니다. 신뢰 정책이 표시될 때 해당 ARN으로 다시 역변환되기 때문입니다. 그러나 역할 또는 사용자를 삭제할 경우, 보안 주체 ID가 콘솔에 표시됩니다. AWS에서 더 이상 이를 ARN에 다시 매핑할 수 없기 때문입니다. 따라서 신뢰 정책의 Principal 요소에서 참조된 사용자 또는 역할을 삭제하고 다시 생성하는 경우, ARN을 바꾸도록 역할을 편집해야 합니다.

두 번째 명령을 사용할 경우, 기존 관리형 정책을 역할에 연결해야 합니다. 다음 권한 정책에서는 역할을 수립하는 사용자가 example_bucket Amazon S3 버킷에서 ListBucket 작업만 수행하도록 허용합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "s3>ListBucket",  
            "Resource": "arn:aws:s3:::example_bucket"  
        }  
    ]  
}
```

O) Test-UserAccess-Role 역할을 생성하기 위해서는 이전 신뢰 정책을 trustpolicyforacct123456789012.json 이름으로 로컬 policies 드라이브의 C: 폴더에 먼저 저장해야 합니다. 그런 다음 이전 권한 정책을 고객 관리형 정책으로서 PolicyForRole 이름으로 AWS 계정에 저장합니다. 그리고 나면 다음 명령을 사용하여 역할을 만들고 관리형 정책을 연결합니다.

```
# Create the role and attach the trust policy file that allows users in the specified  
account to assume the role.  
$ aws iam create-role --role-name Test-UserAccess-Role --assume-role-policy-document  
file://C:\policies\trustpolicyforacct123456789012.json  
  
# Attach the permissions policy (in this example a managed policy) to the role to specify  
what it is allowed to do.
```

```
$ aws iam attach-role-policy --role-name Test-UserAccess-Role --policy-arn arn:aws:iam::123456789012:role/PolicyForRole
```

Important

필요한 구성의 절반이 끝났습니다. 이제 신뢰할 수 있는 계정의 개별 사용자에게 역할로 전환할 수 있는 권한을 부여해야 합니다. 이 단계에 대한 자세한 내용은 [사용자에 대한 역할 전환 권한 부여](#) (p. 229) 단원을 참조하십시오.

역할을 만든 다음 AWS 작업을 수행하거나 AWS 리소스에 액세스할 수 있는 권한을 부여해야 123456789012 계정의 사용자가 역할을 위임할 수 있습니다. 자세한 내용은 [IAM 역할로 전환하기\(AWS CLI\)](#) (p. 235) 단원을 참조하십시오.

IAM 역할 만들기(AWS API)

AWS API에서 역할을 만들려면 여러 단계를 거쳐야 합니다. 콘솔을 사용하여 역할을 만들 때는 많은 단계가 자동으로 수행되지만 API를 사용하면 각 단계를 직접 명시적으로 수행해야 합니다. 역할을 만든 다음 권한 정책을 역할에 할당해야 합니다. 선택적으로 역할에 대한 [권한 경계](#) (p. 317)를 설정할 수 있습니다.

코드로 역할을 만들려면(AWS API)

1. 역할 만들기: [CreateRole](#)

역할의 신뢰 정책에 대해 파일 위치를 지정할 수 있습니다.

2. 역할에 관리형 권한 정책 연결: [AttachRolePolicy](#)

또는

역할을 위한 인라인 권한 정책 생성: [PutRolePolicy](#)

Important

필요한 구성의 절반이 끝났습니다. 이제 신뢰할 수 있는 계정의 개별 사용자에게 역할로 전환할 수 있는 권한을 부여해야 합니다. 이 단계에 대한 자세한 내용은 [사용자에 대한 역할 전환 권한 부여](#) (p. 229) 단원을 참조하십시오.

3. (선택 사항) 태그를 연결하여 사용자 지정 속성을 사용자에게 추가: [TagRole](#)

자세한 내용은 [IAM 엔터티에 대한 태그 관리\(AWS CLI 또는 AWS API\)](#) (p. 262) 단원을 참조하십시오.

4. (선택 사항) 역할([PutRolePermissionsBoundary](#))에 대한 [권한 경계](#) (p. 317)를 설정합니다.

이 권한 경계는 역할이 가질 수 있는 최대 권한을 관리합니다. 권한 경계는 고급 AWS 기능입니다.

역할을 만든 다음 AWS 작업을 수행하거나 AWS 리소스에 액세스할 수 있는 권한을 부여해야 계정의 사용자에게 권한을 부여하여 역할을 위임할 수 있습니다. 역할 위임하기에 대한 자세한 내용은 [IAM 역할\(AWS API\)로 전환하기](#) (p. 238) 단원을 참조하십시오.

AWS 리소스에 대한 액세스를 타사에 부여할 때 외부 ID를 사용하는 방법

이제 AWS 리소스에 대한 액세스를 타사에 부여해야 할 때가 있습니다(액세스 위임). 이 시나리오의 한 가지 중요한 부분은 IAM 역할 신뢰 정책에서 역할 수임자를 지정하는 데 사용할 수 있는 옵션 정보인 외부 ID입니다.

Important

AWS는 외부 ID를 비밀로 취급하지 않습니다. AWS에서 액세스 키 페어 또는 암호와 같은 비밀 정보를 만든 후에는 다시 볼 수 없습니다. 역할의 외부 ID는 해당 역할을 볼 수 있는 권한을 가진 사람만 볼 수 있습니다.

예를 들어 Example Corp이라는 타사를 고용해 AWS 계정을 모니터링하고 비용을 최적화하기로 했다고 가정해봅시다. 일일 경비를 추적하기 위해 Example Corp은 AWS 리소스에 접근해야 합니다. Example Corp 역시 다른 고객을 위해 다른 많은 AWS 계정을 모니터링합니다.

IAM 사용자 및 AWS 계정의 장기 자격 증명에 대한 액세스 권한을 Example Corp에게 제공하지 마십시오. 대신 IAM 역할과 임시 보안 자격 증명을 사용합니다. IAM 역할은 장기 자격 증명(예: IAM 사용자의 액세스 키)을 공유하지 않고도 AWS 리소스에 액세스할 수 있도록 허용하는 메커니즘을 타사에게 제공합니다.

IAM 역할을 사용하여 AWS 계정과 Example Corp 계정 사이에 신뢰 받는 관계를 설정할 수 있습니다. 이 관계가 설정된 후 Example Corp 계정의 멤버는 AWS STS [AssumeRole API](#)를 호출하여 임시 보안 자격 증명을 얻을 수 있습니다. Example Corp 멤버는 자격 증명을 사용하여 계정의 AWS 리소스에 액세스할 수 있습니다.

Note

임시 보안 자격 증명을 얻기 위해 호출할 수 있는 AssumeRole 및 다른 AWS API 작업에 대한 자세한 내용은 다음([임시 보안 자격 증명 요청하기 \(p. 265\)](#))을 참조하십시오.

이 시나리오에 대한 더 자세한 분석은 다음과 같습니다.

1. Example Corp을 고용해 고유한 사용자 지정 식별자를 생성하도록 합니다. 고유한 사용자 지정 ID 및 AWS 계정 번호를 부여합니다. 이 정보는 다음 단계에서 IAM 역할을 생성하는 데 필요합니다.

Note

이 식별자가 Example Corp의 각 고객에게 고유한 것이라면 Example Corp는 ExternalId에 대해 그들이 원하는 어떤 문자열 값이라도 사용할 수 있습니다. 두 고객이 같은 값을 갖지 않는 한, 고객 계정 번호 또는 임의 문자열이 될 수 있습니다. 이는 '보안 유지'를 위한 것은 아닙니다. Example Corp은 각 고객에게 ExternalId 값을 제공해야 합니다. 가장 중요한 것은 그들의 고객이 아닌 Example Corp이 그것을 생성해야 한다는 것입니다.

2. AWS에 로그인해 Example Corp에 리소스에 대한 액세스 권한을 부여하는 IAM 역할을 생성합니다. IAM 역할과 마찬가지로 해당 역할에도 권한 정책과 신뢰 정책이라는 2가지 정책이 있습니다. 그 역할의 신뢰 정책은 역할을 위임할 사용자를 지정합니다. 이 예시 시나리오에서 정책은 Example Corp의 AWS 계정 번호를 Principal로 지정합니다. 이렇게 하면 계정의 자격 증명이 그 역할을 수임하도록 허용합니다. 또한, Condition 요소를 신뢰 정책에 추가합니다. 이 Condition은 Example Corp의 고유 고객 ID와 일치하는지 확인하기 위해 ExternalId 컨텍스트 키를 테스트합니다. 예를 들면 다음과 같습니다.

```
"Principal": {"AWS": "Example Corp's AWS Account ID"},  
"Condition": {"StringEquals": {"sts:ExternalId": "Unique ID Assigned by Example Corp"}}
```

3. 역할에 대한 권한 정책은 해당 역할이 누군가가 수행하도록 허용할 수 있는 작업을 지정합니다. 예를 들어 그 역할은 누군가에게 IAM 사용자나 그룹이 아닌 Amazon EC2 또는 Amazon RDS 리소스만을 관리할 수 있게 허용하도록 지정할 수 있습니다. 이 예시 시나리오에서는 권한 정책을 사용하여 Example Corp에게 계정의 리소스 전체에 대한 읽기 전용 액세스 권한을 부여합니다.
4. 역할을 정의한 후에는 역할의 Amazon 리소스 이름(ARN)을 Example Corp에 제공합니다.
5. Example Corp이 AWS 리소스에 액세스해야 할 때는 그 회사의 누군가가 AWSsts:AssumeRole API를 호출합니다. 이 호출에는 수임할 역할의 ARN과 사용자 지정 ID에 해당하는 ExternalId 파라미터가 포함되어 있습니다.

Example Corp의 AWS 계정을 사용하는 사람이 요청을 하는 경우와 역할 ARN 및 외부 ID가 올바른 경우에 요청이 성공합니다. 그 경우 요청은 역할이 허용하는 AWS 리소스에 액세스하기 위해 Example Corp이 사용할 수 있는 임시 보안 자격 증명을 제공합니다.

다시 말해서 역할 정책에 외부 ID가 포함된다면 그 역할을 수임하고자 하는 사용자는 누구든지 그 역할에서 보안 주체로 지정되어야 하고 정확한 외부 ID를 포함해야 합니다.

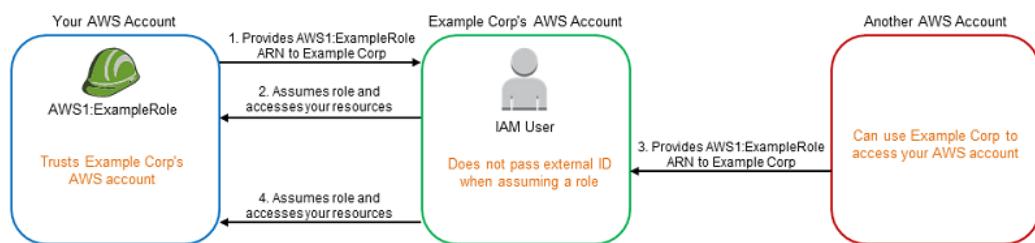
외부 ID를 사용해야 하는 이유는?

주상적인 용어로 말하자면 외부 ID는 그 역할을 위임하고 있는 사용자가 자신이 활동하고 있는 상황을 어설션할 수 있도록 허용합니다. 또한, 계정 소유자가 특정 상황에서만 역할이 위임되도록 허용할 수 있는 방법을 제공합니다. 외부 ID의 주된 기능은 "흔들린 대리자" 문제를 해결하고 방지하는 것입니다.

흔들린 대리자 문제

이전 예시에 이어서 Example Corp은 AWS 계정의 특정 리소스에 대한 액세스 권한이 필요합니다. 그러나 Example Corp에게는 다른 고객도 있으며 각 고객의 AWS 리소스에 액세스할 방법이 필요합니다. 고객들에게 결코 공유되어서는 안 될 비밀인 AWS 계정 액세스 키를 요구하는 대신 Example Corp은 각 사용자에게 역할 ARN을 요청합니다. 하지만 다른 Example Corp 고객은 사용자의 역할 ARN을 추측하거나 얻을 수 있습니다. 해당 고객은 역할 ARN을 사용해 Example Corp을 경유해 AWS 리소스에 대한 액세스 권한을 얻을 수 있습니다. 이러한 형태의 권한 상승은 흔들린 대리자 문제로 알려져 있습니다.

다음 다이어그램은 흔들린 대리자 문제를 보여줍니다.



이 다이어그램은 다음과 같이 가정합니다.

- AWS1은 AWS 계정입니다.
- AWS1:ExampleRole은 계정의 역할입니다. 이 역할의 신뢰 정책은 Example Corp의 AWS 계정의 역할을 위임할 수 있는 것으로 지정함으로써 Example Corp을 신뢰합니다.

다음은 무슨 일이 일어나는지에 대한 것입니다.

1. Example Corp 서비스 사용을 시작할 때 Example Corp에 AWS1:ExampleRole의 ARN을 제공합니다.
2. Example Corp은 그 ARN을 사용해 임시 보안 자격 증명을 얻어 AWS 계정의 리소스에 액세스합니다. 이러한 방식으로 Example Corp을 대신 행위할 수 있는 "대리자"로 신뢰합니다.
3. 또 다른 AWS 고객도 Example Corp의 서비스를 사용하기 시작하고, 이 고객 역시 Example Corp이 사용할 AWS1:ExampleRole의 ARN을 제공합니다. 아마도 그 다른 고객은 비밀이 아닌 AWS1:ExampleRole을 알거나 짐작했을 것입니다.
4. 다른 고객이 Example Corp에게 (자신의 것이라고 주장하는) 계정의 AWS 리소스에 액세스할 수 있는 권한을 요청하면, Example Corp은 AWS1:ExampleRole을 사용해 계정의 리소스에 액세스합니다.

이것이 바로 다른 고객이 리소스에 무단으로 액세스하는 과정입니다. 이 고객은 Example Corp이 자신도 모르게 리소스에 대한 작업을 하도록 속일 수 있었기 때문에 Example Corp은 이제 "흔들린 대리자"가 되었습니다.

외부 ID는 어떻게 흔들린 대리자 문제를 방지할까요?

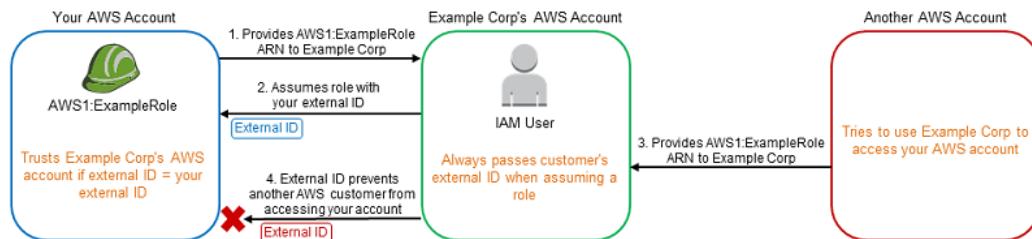
역할의 신뢰 정책에 `ExternalId` 조건 확인을 포함시킴으로써 흔들린 대리자 문제를 해결합니다. "대리자" 회사는 각 고객에 대한 고유 외부 ID 값을 AWS 자격 증명에 대한 요청에 삽입합니다. 외부 ID는 고객 ID 값으로서 Example Corp의 고객 사이에서 고유한 것이어야 하며 Example Corp 고객의 통제를 벗어나 있어야 합니다. 이것이 바로 Example Corp에서 외부 ID를 얻고 그것을 스스로 찾아내지 않는 이유입니다. 이는 다른 고객으로 가장하는 데 성공한 고객을 방지하는 데 도움이 됩니다. Example Corp은 항상 고객의 할당된 외부 ID를 삽입하므로 자신의 것을 제외한 어떤 외부 ID가 포함된 Example Corp의 요청도 결코 눈에 띄어서는 안 됩니다.

이 시나리오에서 Example Corp의 고유 식별자가 "12345"이고, 다른 고객에 대해서는 그 식별자가 "67890"이라고 가정합시다. 이러한 식별자는 이 시나리오를 위해 단순화된 것입니다. 일반적으로 이러한 식별자는 GUID입니다. 이 식별자가 Example Corp의 고객 사이에서 고유한 것이라고 가정할 때, 외부 ID를 위해 사용하기에 합리적인 값들입니다.

Example Corp은 "12345"라는 외부 ID 값을 부여합니다. 그런 다음 Condition 값이 12345가 되어야 한다고 요구하는 역할의 신뢰 정책에 sts:ExternalId 요소를 다음과 같이 추가해야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "sts:AssumeRole",
        "Principal": {"AWS": "Example Corp's AWS Account ID"},
        "Condition": {"StringEquals": {"sts:ExternalId": "12345"}}
    }
}
```

이 정책의 조건 요소는 AssumeRole API 호출에 "12345"라는 외부 ID 값이 포함될 때만 Example Corp이 역할을 수임하도록 허용합니다. Example Corp은 고객을 대신해 역할을 위임할 때마다 항상 AssumeRole 호출에 해당 고객의 외부 ID 값을 포함하도록 보장합니다. 다른 고객이 Example Corp에게 ARN을 공급한다 하더라도 Example Corp이 AWS에 대한 요청 시 포함하는 외부 ID를 제어할 수 없습니다. 이는 다음 디어그램에 나와 있듯이 권한을 부여받지 않은 고객이 리소스에 액세스하지 못하도록 방지하는 데 도움이 됩니다.



- 전과 같이 Example Corp 서비스 사용을 시작할 때 Example Corp에 AWS1:ExampleRole의 ARN을 제공합니다.
- Example Corp이 그 ARN을 사용해 AWS1:ExampleRole 역할을 위임하는 경우 Example Corp은 AssumeRole API 호출에 외부 ID("12345")를 포함시킵니다. 외부 ID는 역할의 신뢰 정책과 일치하므로 AssumeRole API 호출은 성공하고 Example Corp은 임시 보안 자격 증명을 획득해 AWS 계정의 리소스에 액세스합니다.
- 또 다른 AWS 고객도 Example Corp의 서비스를 사용하기 시작하고, 전과 같이 이 고객 역시 Example Corp이 사용할 AWS1:ExampleRole의 ARN을 제공합니다.
- 그러나 이번에는 Example Corp이 AWS1:ExampleRole이라는 역할을 위임하려 할 때 다른 고객과 연결된 외부 ID("67890")를 제공하므로 해당 고객은 이를 바꿀 방법이 없습니다. Example Corp이 이렇게 하는 이유는 역할을 사용하겠다는 요청이 다른 고객에게서 왔으므로, "67890"은 Example Corp이 작용하고 있는 상황을 나타내기 때문입니다. AWS1:ExampleRole의 신뢰 정책에 자신의 외부 ID("12345")가 있는 조건을 추가했기 때문에 AssumeRole API 호출은 실패하고 다른 고객이 계정 리소스에 무단으로 액세스하는 것을 막을 수 있습니다(다이어그램의 빨간색 "X" 참조).

외부 ID는 다른 고객이 Example Corp을 속여 자신도 모르게 리소스에 액세스하지 못하도록 방지함으로써 혼동된 대리자 문제를 완화합니다.

언제 외부 ID를 사용해야 하나요?

다음 상황에서 외부 ID를 사용합니다.

- AWS 계정 소유자이고 다른 AWS 계정에도 액세스하는 타사를 위한 역할을 구성했습니다. 이 경우 타사에 역할을 위임할 때 포함하는 외부 ID를 요청해야 합니다. 그런 다음 역할의 신뢰 정책에서 외부 ID를 확인합니다. 이렇게 하여 외부 사용자를 대신해서 수행하는 경우에만 역할을 맡을 수 있도록 해야 합니다.
- 이전 시나리오의 Example Corp와 같은 다른 고객을 대신하여 역할을 위임할 수 있습니다. 각 고객에게 고유한 외부 ID를 할당하고 외부 ID를 역할의 신뢰 정책에 추가하도록 지시해야 합니다. 그런 다음 역할 위임 요청에 정확한 외부 ID를 항상 포함하도록 해야 합니다.

각 고객에 대한 고유한 식별자를 이미 갖고 있겠지만, 이 고유 ID는 외부 ID로 사용하기에 충분합니다. 외부 ID는 단지 이러한 목적을 위해 명시적으로 생성하거나 별도로 추적할 필요가 있는 특별한 값은 아닙니다.

외부 ID는 항상 `AssumeRole` API 호출에 지정해야 합니다. 이 밖에도 고객이 역할 ARN을 부여할 때 정확한 외부 ID가 있든 없든 그 역할을 위임할 수 있는지 확인하십시오. 정확한 외부 ID 없이 역할을 위임할 수 있는 경우 시스템에 고객의 역할 ARN을 저장하지 마십시오. 고객이 정확한 외부 ID를 요구하도록 역할 신뢰 정책을 업데이트할 때까지 기다립니다. 이러한 방식으로 고객이 올바른 일을 할 수 있도록 돕고, 이는 양자 모두 혼동된 대리자 문제에서 보호 받는 데 도움이 됩니다.

AWS 서비스에 대한 권한을 위임할 역할 생성

AWS 서비스는 역할을 사용하여 서비스가 사용자를 대신하여 다른 서비스의 리소스로 액세스할 수 있어야 합니다. 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임한 역할을 [서비스 역할 \(p. 154\)](#)이라고 합니다. 역할이 서비스에 대해 특수한 목적을 수행하는 경우 [EC2 인스턴스의 서비스 역할 \(p. 154\)](#) 또는 [서비스 연결 역할 \(p. 154\)](#)로 분류됩니다. 서비스 연결 역할을 사용하여 지원되는 서비스 또는 서비스가 임시 자격 증명의 형식을 지원하는지 여부를 확인하려면 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하십시오. 개별 서비스가 역할을 사용하는 방법을 알아보려면 테이블에서 서비스 이름을 선택하여 해당 서비스의 설명서를 확인합니다.

역할을 통해 권한을 위임하는 방법에 대한 자세한 내용은 [역할 용어 및 개념 \(p. 153\)](#) 단원을 참조하십시오.

서비스 역할 권한

IAM 개체(사용자, 그룹, 역할 등)가 서비스 연결 역할을 작성하거나 편집할 수 있도록 권한을 구성해야 합니다.

Note

서비스 링크된 역할에 대한 ARN은 정책에서 `SERVICE-NAME.amazonaws.com`으로 나타내지는 서비스 보안 주체를 포함합니다. 각 경우마다 다르고 AWS 서비스에 따라 형식이 다양하기 때문에 서비스 보안 주체를 알기 어렵습니다. 서비스의 보안 주체를 보려면 해당 서비스 링크된 역할 설명서 단원을 참조하십시오.

IAM 개체가 특정 서비스 연결 역할을 만들 수 있도록 허용하려면

서비스 연결 역할을 생성해야 하는 IAM 개체에 다음 정책을 추가합니다. 이 정책으로 특정 서비스에 대하여 구체적인 이름이 있는 서비스 역할을 만들 수 있습니다. 그런 다음 관리형 또는 인라인 정책을 해당 역할에 연결할 수 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "CreateSpecificRoleForSpecificService",  
            "Effect": "Allow",  
            "Action": "iam:CreateRole",  
            "Resource": "arn:aws:iam::*:role/SERVICE-ROLE-NAME",  
            "Condition": {"StringLike": {"iam:AWSServiceName": "SERVICE-NAME.amazonaws.com"}}  
        },  
        {  
        }  
    ]  
}
```

```
    "Sid": "AddPoliciesToSpecificRole",
    "Effect": "Allow",
    "Action": [
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy"
    ],
    "Resource": "arn:aws:iam::*:role/SERVICE-ROLE-NAME"
}
]
```

IAM 개체가 서비스 연결 역할을 만들 수 있도록 허용하려면

서비스 역할을 생성해야 하는 IAM 개체의 권한 정책에 다음 명령문을 추가합니다. 이 문으로 모든 서비스에 대하여 서비스 역할을 만든 후 관리형 또는 인라인 정책을 해당 역할에 연결할 수 있습니다.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:AttachRolePolicy",
        "iam:CreateRole",
        "iam:PutRolePolicy"
    ],
    "Resource": "*"
}
```

IAM 개체가 서비스 역할을 편집할 수 있도록 허용하려면

서비스 연결 역할을 편집해야 하는 IAM 개체에 다음 정책을 추가합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EditSpecificServiceRole",
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:DeleteRolePolicy",
                "iam:DetachRolePolicy",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam>ListAttachedRolePolicies",
                "iam>ListRolePolicies",
                "iam:PutRolePolicy",
                "iam:UpdateRole",
                "iam:UpdateRoleDescription"
            ],
            "Resource": "arn:aws:iam::*:role/SERVICE-ROLE-NAME"
        },
        {
            "Sid": "ViewRolesAndPolicies",
            "Effect": "Allow",
            "Action": [
                "iam:GetPolicy",
                "iam>ListRoles"
            ],
            "Resource": ""
        }
    ]
}
```

IAM 개체가 특정 서비스 역할을 삭제하도록 허용하려면

특정 서비스 역할을 삭제해야 하는 IAM 개체의 권한 정책에 다음 문장을 추가합니다.

```
{  
    "Effect": "Allow",  
    "Action": "iam:DeleteRole",  
    "Resource": "arn:aws:iam::*:role/SERVICE-ROLE-NAME"  
}
```

IAM 개체가 서비스 역할을 삭제하도록 허용하려면

서비스 역할을 삭제해야 하는 IAM 개체의 권한 정책에 다음 명령문을 추가합니다.

```
{  
    "Effect": "Allow",  
    "Action": "iam:DeleteRole",  
    "Resource": "*"  
}
```

AWS 서비스에 대한 역할 생성(콘솔)

AWS Management 콘솔을 사용하여 서비스의 역할을 만들 수 있습니다. 일부 서비스는 두 개 이상의 서비스 역할을 지원하기 때문에 어떤 사용 사례를 선택할지 확인하려면 해당 서비스의 [AWS 설명서](#) 단원을 참조하십시오. 서비스에서 역할을 위임할 수 있도록 역할에 필요한 신뢰 정책과 권한 정책을 할당하는 방법을 알아볼 수 있습니다. 역할에 대한 권한을 관리할 수 절차는 서비스가 어떻게 사용 사례를 정의하느냐와 서비스 링크된 역할을 생성할 수 있는지 여부에 따라 다양할 수 있습니다.

AWS 서비스에 대한 역할을 만들려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 역할을 선택한 후 역할 생성을 선택합니다.
3. 신뢰할 수 있는 유형의 엔터티 선택에서 AWS 서비스를 선택합니다.
4. 이 역할을 맡을 수 있게 하려는 서비스를 선택합니다.
5. 서비스의 사용 사례를 선택합니다. 지정한 서비스에 사용 사례가 하나뿐이면 자동으로 선택됩니다. 사용 사례는 서비스에 필요한 신뢰 정책을 포함하기 위해 서비스에서 정합니다. 그런 다음 [Next: Permissions]를 선택합니다.
6. 가능하다면, 권한 정책을 사용하기 위한 정책을 선택하거나 정책 생성을 선택하여 새 브라우저 탭을 열고 완전히 새로운 정책을 생성합니다. 자세한 내용은 [IAM 정책 만들기\(콘솔\) \(p. 378\)](#) 절차의 4단계 단원을 참조하십시오. 정책을 생성하면 탭을 닫고 원래 탭으로 돌아갑니다. 서비스에게 부여하려는 권한 정책 옆의 확인란을 선택합니다.

선택한 사용 사례에 따라 서비스에서 다음을 수행할 수 있습니다.

- 서비스에서 역할에 대한 권한을 정의하기 때문에 할 일이 아무 것도 없습니다.
 - 제한된 권한 집합에서 선택할 수 있도록 허용
 - 모든 권한 집합에서 선택할 수 있도록 허용
 - 여기서 정책을 선택하지 않고, 나중에 정책을 만들어 역할에 연결할 수 있도록 허용
7. (선택 사항) [권한 경계 \(p. 317\)](#)로서 설정됨. 이는 서비스 역할에서 가능한 고급 기능이며 서비스 링크된 역할은 아닙니다.

Set permissions boundary(권한 경계 설정) 섹션을 열고 Use a permissions boundary to control the maximum role permissions(최대 사용자 권한을 관리하기 위한 권한 경계 사용)을 선택합니다. IAM에는 계정의 AWS 관리형 또는 사용자 관리형 정책 목록이 있습니다. 권한 경계를 사용하기 위한 정책을 선택하거나 정책 생성을 선택하여 새 브라우저 탭을 열고 완전히 새로운 정책을 생성합니다. 자세한 내용은 [IAM 정책 만들기\(콘솔\) \(p. 378\)](#) 절차의 4단계 단원을 참조하십시오. 정책을 생성하면 탭을 닫고 원래 탭으로 돌아와 권한 경계를 사용하기 위한 정책을 선택합니다.

8. 다음: 태그 지정을 선택합니다.
9. (선택 사항) 태그를 키–값 페어로 연결하여 메타데이터를 역할에 추가합니다. IAM에서의 태그 사용에 대한 자세한 내용은 [IAM 엔터티 태그 지정 \(p. 259\)](#) 단원을 참조하십시오.
10. [Next: Review]를 선택합니다.
11. 역할 이름의 경우 역할 이름 사용자 지정 수준은 서비스에서 정합니다. 서비스에서 역할 이름을 정한 경우 이 옵션을 편집할 수 없습니다. 다른 경우에는 서비스에서 역할 이름의 접두사를 정의하고 사용자가 선택적으로 접미부를 입력하도록 할 수 있습니다. 일부 서비스는 역할의 전체 이름을 지정할 수 있습니다.

가능하다면 역할 이름 또는 역할 이름 접미사를 입력합니다. 역할 이름은 AWS 계정 내에서 고유해야 합니다. 대소문자는 구별하지 않습니다. 예를 들어, 이름이 **PRODROLE**과 **prodrole**, 두 가지로 지정된 역할을 만들 수는 없습니다. 기타 AWS 리소스가 역할을 참조할 수 있기 때문에 역할이 생성된 후에는 역할 이름을 편집할 수 없습니다.
12. (선택 사항) [Role description]에 새 역할에 대한 설명을 입력합니다.
13. 역할을 검토한 다음 [Create role]을 선택합니다.

서비스에 대한 역할 생성(AWS CLI)

AWS CLI에서 역할을 만들려면 여러 단계를 거쳐야 합니다. 콘솔을 사용하여 역할을 만들 때는 많은 단계가 자동으로 수행되지만 AWS CLI를 사용하면 각 단계를 직접 명시적으로 수행해야 합니다. 역할을 만든 다음 권한 정책을 역할에 할당해야 합니다. 작업 중인 서비스가 Amazon EC2인 경우에도 인스턴스 프로파일을 만들어 거기에 역할을 추가해야 합니다. 선택적으로 역할에 대한 [권한 경계 \(p. 317\)](#)를 설정할 수 있습니다.

AWS CLI에서 AWS 서비스에 대한 역할을 만들려면

1. 역할 생성: [aws iam create-role](#)
2. 역할에 관리형 권한 정책 연결: [aws iam attach-role-policy](#)

또는

역할을 위한 인라인 권한 정책 생성: [aws iam put-role-policy](#)

3. (선택 사항) 태그를 연결하여 사용자 지정 속성을 역할에 추가: [aws iam tag-role](#)

자세한 내용은 [IAM 엔터티에 대한 태그 관리\(AWS CLI 또는 AWS API\) \(p. 262\)](#) 단원을 참조하십시오.

4. (선택 사항) 역할([aws iam put-role-permissions-boundary](#))에 대한 [권한 경계 \(p. 317\)](#)를 설정합니다.

이 권한 경계는 역할이 가질 수 있는 최대 권한을 관리합니다. 권한 경계는 고급 AWS 기능입니다.

Amazon EC2 또는 Amazon EC2를 사용하는 다른 AWS 서비스에 대해 역할을 사용할 경우 인스턴스 프로파일에 역할을 저장해야 합니다. 인스턴스 프로파일은 시작할 때 Amazon EC2 인스턴스에 연결할 수 있는 역할을 위한 컨테이너입니다. 하나의 인스턴스 프로파일은 하나의 역할만 포함할 수 있으며 이 제한은 늘릴 수 없습니다. AWS Management 콘솔을 사용하여 역할을 생성한 경우 역할과 동일한 이름을 지닌 인스턴스 프로파일이 자동으로 생성됩니다. 인스턴스 프로파일에 대한 자세한 내용은 [인스턴스 프로파일 사용 \(p. 244\)](#) 단원을 참조하십시오. 역할을 사용하여 EC2 인스턴스를 시작하는 방법에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [Amazon EC2 리소스에 대한 액세스 제어](#) 단원을 참조하십시오.

인스턴스 프로파일을 만들고 여기에 역할을 저장하려면(AWS CLI)

1. 인스턴스 프로파일 생성: [aws iam create-instance-profile](#)
2. 인스턴스 프로파일에 역할 추가: [aws iam add-role-to-instance-profile](#)

아래 AWS CLI 예제 명령 집합은 역할을 생성하고 권한을 연결하는 첫 두 단계를 보여줍니다. 인스턴스 프로파일을 생성하고 프로필에 역할을 추가하는 두 단계를 보여주기도 합니다. 이 예제 신뢰 정책은 Amazon

EC2 서비스가 역할을 맡고 example_bucket Amazon S3 버킷을 볼 수 있도록 허용합니다. 이 예제에서는 Windows를 실행하는 클라이언트 컴퓨터에서 실행 중이며 계정 자격 증명 및 리전으로 이미 명령줄 인터페이스를 구성했다고도 가정합니다. 자세한 정보는 [AWS 명령줄 인터페이스 구성](#)을 참조하십시오.

이 예제는 역할을 생성할 경우 첫 번째 명령의 다음 신뢰 정책을 포함합니다. 이 신뢰 정책은 Amazon EC2 서비스가 역할을 가정하도록 허용합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Principal": {"Service": "ec2.amazonaws.com"},  
         "Action": "sts:AssumeRole"}  
    ]  
}
```

두 번째 명령을 사용할 경우, 권한 정책을 역할에 연결해야 합니다. 다음 예제 권한 정책에서는 역할이 example_bucket Amazon S3 버킷에서 ListBucket 작업만 수행하도록 허용합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": "s3>ListBucket",  
         "Resource": "arn:aws:s3:::example_bucket"}  
    ]  
}
```

이 Test-Role-for-EC2 역할을 생성하기 위해서는 먼저 이전 신뢰 정책을 trustpolicyforec2.json 이름으로, 이전 권한 정책을 permissionspolicyforec2.json 이름으로 로컬 C: 드라이브의 policies 디렉터리에 저장해야 합니다. 그리고 나면 다음 명령을 사용하여 역할을 만들고 인라인 정책을 연결, 인스턴스 프로파일 생성 및 인스턴스 프로파일에 역할을 추가합니다.

```
# Create the role and attach the trust policy that allows EC2 to assume this role.  
$ aws iam create-role --role-name Test-Role-for-EC2 --assume-role-policy-document file://C:/  
policies\trustpolicyforec2.json  
  
# Embed the permissions policy (in this example an inline policy) to the role to specify  
what it is allowed to do.  
$ aws iam put-role-policy --role-name Test-Role-for-EC2 --policy-name Permissions-Policy-  
For-Ec2 --policy-document file://permissionspolicyforec2.json  
  
# Create the instance profile required by EC2 to contain the role  
$ aws iam create-instance-profile --instance-profile-name EC2-ListBucket-S3  
  
# Finally, add the role to the instance profile  
$ aws iam add-role-to-instance-profile --instance-profile-name EC2-ListBucket-S3 --role-  
name Test-Role-for-EC2
```

EC2 인스턴스를 시작할 때 AWS 콘솔을 사용하는 경우 인스턴스 세부 정보 구성 페이지에 인스턴스 프로파일 이름을 지정합니다. aws ec2 run-instances CLI 명령을 사용하는 경우 --iam-instance-profile 파라미터를 지정합니다.

서비스에 대한 역할 생성(AWS API)

AWS API에서 역할을 만들려면 여러 단계를 거쳐야 합니다. 콘솔을 사용하여 역할을 만들 때는 많은 단계가 자동으로 수행되지만 API를 사용하면 각 단계를 직접 명시적으로 수행해야 합니다. 역할을 만든 다음 권한 정책을 역할에 할당해야 합니다. 작업 중인 서비스가 Amazon EC2인 경우에도 인스턴스 프로파일을 만들어 거기에 역할을 추가해야 합니다. 선택적으로 역할에 대한 [권한 경계\(p. 317\)](#)를 설정할 수 있습니다.

AWS 서비스에 대한 역할을 생성하려면(AWS API)

1. 역할 만들기: [CreateRole](#)

역할의 신뢰 정책에 대해 파일 위치를 지정할 수 있습니다.

2. 역할에 관리형 권한 정책 연결: [AttachRolePolicy](#)

또는

역할을 위한 인라인 권한 정책 생성: [PutRolePolicy](#)

3. (선택 사항) 태그를 연결하여 사용자 지정 속성을 사용자에게 추가: [TagRole](#)

자세한 내용은 [IAM 엔터티에 대한 태그 관리\(AWS CLI 또는 AWS API\) \(p. 262\)](#) 단원을 참조하십시오.

4. (선택 사항) 역할([PutRolePermissionsBoundary](#))에 대한 권한 경계 (p. 317)를 설정합니다.

이 권한 경계는 역할이 가질 수 있는 최대 권한을 관리합니다. 권한 경계는 고급 AWS 기능입니다.

Amazon EC2 또는 Amazon EC2를 사용하는 다른 AWS 서비스에 대해 역할을 사용할 경우 인스턴스 프로파일에 역할을 저장해야 합니다. 인스턴스 프로파일은 역할에 대한 컨테이너입니다. 각 인스턴스 프로파일은 하나의 역할만 포함할 수 있으며 이 제한은 늘릴 수 없습니다. AWS Management 콘솔에서 역할을 생성한 경우 역할과 동일한 이름을 지닌 인스턴스 프로파일이 자동으로 생성됩니다. 인스턴스 프로파일에 대한 자세한 내용은 [인스턴스 프로파일 사용 \(p. 244\)](#) 단원을 참조하십시오. 역할을 사용하여 Amazon EC2 인스턴스를 시작하는 방법에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [Amazon EC2 리소스에 대한 액세스 제어](#) 단원을 참조하십시오.

인스턴스 프로파일을 만들고 여기에 역할을 저장하려면(AWS API)

1. 인스턴스 프로파일 생성: [CreateInstanceProfile](#)

2. 인스턴스 프로파일에 역할 추가: [AddRoleToInstanceProfile](#)

타사 자격 증명 공급자의 역할 만들기(연동)

AWS 계정에 속하는 IAM 사용자를 생성하는 대신에 자격 증명 공급자를 사용할 수 있습니다. 자격 증명 공급자(IdP)를 사용하면 AWS 외부의 사용자 자격 증명을 관리할 수 있고 이 외부 사용자 자격 증명에 계정의 AWS 리소스에 대한 사용 권한을 부여할 수 있습니다. 연동 및 자격 증명 공급자에 대한 자세한 내용은 [자격 증명 공급자 및 연동 \(p. 161\)](#) 단원을 참조하십시오.

연동 사용자의 역할 만들기(콘솔)

연동 사용자의 역할을 만드는 절차는 타사 공급자들의 선택에 따라 다릅니다.

- 웹 자격 증명 또는 OpenID Connect 연동(OIDC)을 위한 역할 생성은 [웹 자격 증명 또는 OpenID Connect 연동을 위한 역할 생성\(콘솔\) \(p. 217\)](#) 단원을 참조하십시오.
- SAML 2.0은 [SAML 2.0 연동을 위한 역할 생성\(콘솔\) \(p. 221\)](#) 단원을 참조하십시오.

연동 액세스의 역할 만들기(AWS CLI)

AWS CLI에서 지원되는 자격 증명 공급자(OIDC 또는 SAML)의 역할을 만드는 절차는 동일합니다. 차이는 필수 선행 단계에서 생성하는 신뢰 정책의 내용에 있습니다. 사용하고 있는 공급자의 유형에 대한 필수 선행 조건 섹션에 나와 있는 절차에서부터 시작하십시오.

- OIDC 공급자의 경우 [웹 자격 증명 또는 OIDC의 역할 생성하기 위한 사전 조건 \(p. 217\)](#) 단원을 참조하십시오.
- SAML 공급자의 경우 [SAML 역할 생성하기 위한 사전 조건 \(p. 221\)](#) 단원을 참조하십시오.

AWS CLI에서 역할을 만들려면 여러 단계를 거쳐야 합니다. 콘솔을 사용하여 역할을 만들 때는 많은 단계가 자동으로 수행되지만 AWS CLI를 사용하면 각 단계를 직접 명시적으로 수행해야 합니다. 역할을 만든 다음 권한 정책을 역할에 할당해야 합니다. 선택적으로 역할에 대한 [권한 경계 \(p. 317\)](#)를 설정할 수 있습니다.

자격 증명 연동의 역할을 만들려면(AWS CLI)

1. 역할 생성: [aws iam create-role](#)
2. 역할에 권한 정책 연결: [aws iam attach-role-policy](#)

또는

역할을 위한 인라인 권한 정책 생성: [aws iam put-role-policy](#)

3. (선택 사항) 태그를 연결하여 사용자 지정 속성을 역할에 추가: [aws iam tag-role](#)

자세한 내용은 [IAM 엔터티에 대한 태그 관리\(AWS CLI 또는 AWS API\) \(p. 262\)](#) 단원을 참조하십시오.

4. (선택 사항) 역할([aws iam put-role-permissions-boundary](#))에 대한 [권한 경계 \(p. 317\)](#)를 설정합니다.

이 권한 경계는 역할이 가질 수 있는 최대 권한을 관리합니다. 권한 경계는 고급 AWS 기능입니다.

다음 예는 단순한 환경에서 자격 증명 공급자를 생성하는 가장 일반적인 단계 중 첫 두 단계를 보여줍니다. 이 예제는 123456789012 계정에 있는 모든 사용자가 역할을 가정하고 example_bucket Amazon S3 버킷을 볼 수 있도록 허용합니다. 또한 이 예는 Windows가 구동중인 컴퓨터에서 AWS CLI를 실행하고 있으며 자격 증명으로 AWS CLI를 이미 구성했다고 가정합니다. 자세한 내용은 [AWS Command Line Interface 구성 단원](#)을 참조하십시오.

이 예제는 역할을 생성할 경우 첫 번째 명령의 다음 신뢰 정책을 포함합니다. 이 신뢰 정책은 123456789012 계정에서 사용자가 AssumeRole 작업을 사용하여 역할을 가정할 수 있도록 허용합니다. 단, 사용자가 SerialNumber 및 TokenCode 파라미터를 사용하는 MFA 인증을 제공하는 경우에만 허용합니다. MFA에 대한 자세한 내용은 [AWS에서 멀티 팩터 인증\(MFA\) 사용하기 \(p. 96\)](#) 단원을 참조하십시오.

다음 예는 사용자가 Amazon Cognito를 사용하여 로그인하는 경우 모바일 앱에 대해 설계되는 신뢰 정책을 보여줍니다. 이 예시에서 **us-east:12345678-ffff-ffff-ffff-123456**은 Amazon Cognito에 의해 할당된 자격 증명 풀 ID를 나타냅니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Sid": "RoleForCognito",  
        "Effect": "Allow",  
        "Principal": {"Federated": "cognito-identity.amazonaws.com"},  
        "Action": "sts:AssumeRoleWithWebIdentity",  
        "Condition": {"StringEquals": {"cognito-identity.amazonaws.com:aud": "us-east:12345678-ffff-ffff-ffff-123456"}  
    }  
}
```

다음 권한 정책에서는 역할을 수임하는 사용자가 example_bucket Amazon S3 버킷에서 ListBucket 작업만 수행하도록 허용합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": "s3>ListBucket",  
        "Resource": "arn:aws:s3:::example_bucket"  
    }  
}
```

01 Test-Cognito-Role 역할을 생성하기 위해서는 이전 신뢰 정책을 `trustpolicyforcognitofederation.json` 이름으로 이전 권한 정책을 `permsspolicyforcognitofederation.json` 이름으로 로컬 policies 드라이브의 C: 폴더에 먼저 저장해야 합니다. 그리고 나면 다음 명령을 사용하여 역할을 만들고 인라인 정책을 연결합니다.

```
# Create the role and attach the trust policy that enables users in an account to assume the role.  
$ aws iam create-role --role-name Test-Cognito-Role --assume-role-policy-document file://C:\policies\trustpolicyforcognitofederation.json  
  
# Attach the permissions policy to the role to specify what it is allowed to do.  
aws iam put-role-policy --role-name Test-Cognito-Role --policy-name Perms-Policy-For-CognitoFederation --policy-document file://C:\policies\permsspolicyforcognitofederation.json
```

연동 액세스의 역할 만들기(AWS API)

AWS CLI에서 지원되는 자격 증명 공급자(OIDC 또는 SAML)의 역할을 만드는 절차는 동일합니다. 차이는 필수 선행 단계에서 생성하는 신뢰 정책의 내용에 있습니다. 사용하고 있는 공급자의 유형에 대한 필수 선행 조건 섹션에 나와 있는 절차에서부터 시작하십시오.

- OIDC 공급자의 경우 [웹 자격 증명 또는 OIDC의 역할 생성하기 위한 사전 조건 \(p. 217\)](#) 단원을 참조하십시오.
- SAML 공급자의 경우 [SAML 역할 생성하기 위한 사전 조건 \(p. 221\)](#) 단원을 참조하십시오.

자격 증명 연동의 역할(AWS API)을 만들려면

1. 역할 만들기: [CreateRole](#)
2. 역할에 권한 정책 연결: [AttachRolePolicy](#)

또는

역할을 위한 인라인 권한 정책 생성: [PutRolePolicy](#)

3. (선택 사항) 태그를 연결하여 사용자 지정 속성을 사용자에게 추가: [TagRole](#)

자세한 내용은 [IAM 엔터티에 대한 태그 관리\(AWS CLI 또는 AWS API\) \(p. 262\)](#) 단원을 참조하십시오.

4. (선택 사항) 역할([PutRolePermissionsBoundary](#))에 대한 [권한 경계 \(p. 317\)](#)을 설정합니다.

이 권한 경계는 역할이 가질 수 있는 최대 권한을 관리합니다. 권한 경계는 고급 AWS 기능입니다.

웹 자격 증명 또는 OpenID Connect 연동을 위한 역할 생성(콘솔)

AWS 계정에 IAM 사용자를 생성하는 대신에 웹 자격 증명 연동 또는 OpenID Connect Federation(OIDC) 자격 증명 공급자를 사용할 수 있습니다. 자격 증명 공급자(IdP)를 사용하면 AWS 외부의 사용자 자격 증명을 관리할 수 있고 이 외부 사용자 자격 증명에 계정의 AWS 리소스에 대한 사용 권한을 부여할 수 있습니다. 연동 및 자격 증명 공급자에 대한 자세한 내용은 [자격 증명 공급자 및 연동 \(p. 161\)](#) 단원을 참조하십시오.

웹 자격 증명 또는 OIDC의 역할 생성하기 위한 사전 조건

웹 자격 증명 연동을 위한 역할을 만들기 전에 먼저 다음 필수 선행 단계를 완료해야 합니다.

웹 자격 증명 연동을 위한 역할 만들기를 준비하려면

1. 하나 이상의 IdP를 사용해 개발자로 로그인합니다. AWS 리소스로 액세스가 필요한 앱을 생성하면 공급자 정보로 앱도 구성합니다. 이렇게 하면 공급자는 앱의 고유한 애플리케이션 및 시청자 ID를 제공합니다. 서로 다른 공급자는 이 과정에 대해 서로 다른 용어를 사용합니다. 이 가이드는 앱을 공급자와 동일 시하는 과정에 대해 구성이라는 용어를 사용합니다. 각 공급자로 여러 개의 앱을 구성하거나 단일 앱을 통해 다양한 공급자를 구성할 수 있습니다. 자격 증명 공급자에 대한 정보 보기

- [Login with Amazon 개발자 센터](#)
 - Facebook 개발자 사이트의 앱 또는 웹 사이트에 Facebook 로그인 추가하기
 - Google 개발자 사이트의 OAuth 2.0을 사용한 로그인(OpenID Connect)
2. IAM의 자격 증명 공급자로부터 필요한 정보를 가져온 다음 의 자격 증명 공급자를 만들 수 있습니다. 자세한 내용은 [OpenID Connect\(OIDC\) 자격 증명 공급자의 생성 \(p. 171\)](#) 단원을 참조하십시오.
 3. IdP를 통해 인증된 사용자가 맙을 역할에 대한 정책을 준비합니다. 다른 어떤 역할과 마찬가지로 모바일 앱을 위한 역할에는 2개의 정책이 포함됩니다. 하나는 역할을 위임할 사용자를 지정하는 신뢰 정책입니다. 다른 하나는 모바일 앱의 액세스가 허용 또는 거부되는 AWS 작업 및 리소스를 지정하는 권한 정책입니다.

웹 자격 증명 공급자의 경우, [Amazon Cognito](#)를 사용하여 자격 증명을 관리하는 것이 좋습니다. 이 경우 이 예제와 비슷한 신뢰 정책을 사용합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {"Federated": "cognito-identity.amazonaws.com"},
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {"cognito-identity.amazonaws.com:aud": "us-east-2:12345678-abcd-abcd-abcd-123456"},  

                "ForAnyValue:StringLike": {"cognito-identity.amazonaws.com:amr": "unauthenticated"}
            }
        }
    ]
}
```

us-east-2:12345678-abcd-abcd-abcd-123456을 Amazon Cognito에서 할당한 자격 증명 풀 ID로 대체합니다.

신뢰 정책을 생성할 시 웹 자격 증명 IdP를 수동으로 구성하려면 자체 앱만이 이 역할을 수임한다고 보장하는 세 가지 값을 사용해야 합니다.

- Action 요소에 대해서는 sts:AssumeRoleWithWebIdentity 작업을 사용하십시오.
- Principal 요소에 대해서는 {"Federated":[providerUrl/providerArn](#)} 문자열을 사용하십시오.
- 일부 범용 OpenID Connect(OIDC) IdP의 경우, [providerUrl](#)이 URL입니다. 다음 예제는 일부 범용 IdP에 대해 보안 주체를 지정하는 방법을 포함합니다.

```
"Principal": {"Federated": "cognito-identity.amazonaws.com"}  
  

"Principal": {"Federated": "www.amazon.com"}  
  

"Principal": {"Federated": "graph.facebook.com"}  
  

"Principal": {"Federated": "accounts.google.com"}
```

- 다른 OIDC 공급자의 경우, 다음 예시와 같이 Step 2에서 생성한 OIDC 자격 증명 공급자의 ARN을 사용합니다.

```
"Principal": {"Federated": "arn:aws:iam::123456789012:oidc-provider/  
server.example.com"}
```

- 권한을 제한하려면 Condition 요소에 StringEquals 조건을 사용합니다. 자격 증명 풀 ID(Amazon Cognito용) 또는 앱 ID(다른 공급자용)를 테스트합니다. 이는 IdP를 통해 앱을 구성할 때 얻은 앱 ID와 일치해야 합니다. 이로써 그 요청이 앱으로부터 오는 것임을 확인합니다. 사용하는 IdP에 따라 다음 예제와 비슷한 조건 요소를 생성합니다.

```
"Condition": {"StringEquals": {"cognito-identity.amazonaws.com:aud": "us-east:12345678-ffff-ffff-ffff-123456"}}

"Condition": {"StringEquals": {"www.amazon.com:app_id": "amzn1.application-oa2-123456"}}

"Condition": {"StringEquals": {"graph.facebook.com:app_id": "111222333444555"}}

"Condition": {"StringEquals": {"accounts.google.com:aud": "66677788899900pro0"}}
```

OIDC 공급자의 경우 다음 예시와 같이 aud 컨텍스트 키로 OIDC IdP의 정규화된 URL을 사용합니다.

```
"Condition": {"StringEquals": {"server.example.com:aud": "appid_from_oidc_idp"}}
```

역할의 신뢰 정책에서 보안 주체에 대한 값은 하나의 IdP에 고유한 것이라는 점에 유의하십시오. 하나의 역할은 오직 하나의 보안 주체만을 지정할 수 있습니다. 따라서 모바일 앱이 사용자에게 1개 이상의 IdP에서 로그인할 수 있게 허용한다면 지원하고자 하는 각각의 IdP에 대한 개별 역할을 만들어야 합니다. 따라서, 각 IdP에 대한 개별 신뢰 정책을 생성해야 합니다.

다음 예는 사용자가 Login with Amazon에서 로그인하는 경우 모바일 앱에 대해 설계되는 신뢰 정책을 보여줍니다. 예시에서 [amzn1.application-oa2-123456](#)은 Login with Amazon을 이용해 앱을 구성할 때 Amazon이 할당한 앱 ID를 나타냅니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RoleForLoginWithAmazon",
            "Effect": "Allow",
            "Principal": {"Federated": "www.amazon.com"},
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {"StringEquals": {"www.amazon.com:app_id": "amzn1.application-oa2-123456"}}
        }
    ]
}
```

다음 예는 사용자가 Facebook에서 로그인하는 경우 모바일 앱에 대해 설계되는 신뢰 정책을 보여줍니다. 이 예시에서 [111222333444555](#)는 Facebook에 의해 할당된 앱 ID를 나타냅니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RoleForFacebook",
            "Effect": "Allow",
            "Principal": {"Federated": "graph.facebook.com"},
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {"StringEquals": {"graph.facebook.com:app_id": "111222333444555"}}
        }
    ]
}
```

다음 예는 사용자가 Google에서 로그인하는 경우 모바일 앱에 대해 설계되는 신뢰 정책을 보여줍니다. 이 예시에서 [666777888999000](#)은 Google에 의해 할당된 앱 ID를 나타냅니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Sid": "RoleForGoogle",  
        "Effect": "Allow",  
        "Principal": {"Federated": "accounts.google.com"},  
        "Action": "sts:AssumeRoleWithWebIdentity",  
        "Condition": {"StringEquals": {"accounts.google.com:aud": "666777888999000"}}  
    }]  
}
```

다음 예는 사용자가 Amazon Cognito를 사용하여 로그인하는 경우 모바일 앱에 대해 설계되는 신뢰 정책을 보여줍니다. 이 예시에서 **us-east:12345678-ffff-ffff-ffff-123456**은 Amazon Cognito에 의해 할당된 자격 증명 풀 ID를 나타냅니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Sid": "RoleForCognito",  
        "Effect": "Allow",  
        "Principal": {"Federated": "cognito-identity.amazonaws.com"},  
        "Action": "sts:AssumeRoleWithWebIdentity",  
        "Condition": {"StringEquals": {"cognito-identity.amazonaws.com:aud": "us-  
east:12345678-ffff-ffff-ffff-123456"}}  
    }]  
}
```

웹 자격 증명/OIDC를 위한 역할 생성

사전 요구 사항을 완료한 후에는 IAM에서 역할을 만들 수 있습니다. 다음 절차는 AWS Management 콘솔에서 웹 자격 증명/OIDC에 대한 역할을 만드는 방법을 설명합니다. AWS CLI 또는 AWS API에 역할을 만들려면 [타사 자격 증명 공급자의 역할 만들기\(연동\)](#)(p. 215)의 절차 단원을 참조하십시오.

Important

Amazon Cognito를 사용하고 있는 경우 Amazon Cognito 콘솔을 사용해 역할을 설정해야 합니다. 그렇지 않다면 IAM 콘솔을 사용하여 웹 자격 증명 연동의 역할을 만듭니다.

웹 자격 증명 연동의 IAM 역할을 만들려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 [Roles]를 선택한 후 [Create role]을 선택합니다.
3. 웹 ID 역할 유형을 선택합니다.
4. 자격 증명 공급자에서 역할의 자격 증명 공급자를 선택합니다.
 - 개별 웹 자격 증명 공급자에 대한 역할을 만들 경우, Login with Amazon, Facebook 또는 Google을 선택합니다.

Note

지원할 각 자격 증명 공급자에 대해 별도의 역할을 만들어야 합니다.

- Amazon Cognito의 고급 역할을 만드는 경우 Amazon Cognito를 선택합니다.

Note

고급 시나리오에서 작업할 때는 Amazon Cognito로 사용할 역할을 수동으로 만들기만 하면 됩니다. 그렇지 않은 경우 Amazon Cognito가 역할을 대신 만들 수 있습니다. Amazon Cognito에 대한 자세한 내용은 AWS iOS용 Mobile SDK 개발자 안내서의 [Amazon Cognito 자격 증명](#) 및 AWS Android용 Mobile SDK 개발자 안내서의 [Amazon Cognito 자격 증명](#) 단원을 참조하십시오.

5. 애플리케이션의 ID를 입력합니다. ID의 라벨은 선택한 공급자에 따라 변경됩니다.
 - Login with Amazon에 대한 역할을 만드는 경우 Application ID(애플리케이션 ID) 상자에 애플리케이션 ID를 입력합니다.
 - Facebook에 대한 역할을 만드는 경우 Application ID(애플리케이션 ID) 상자에 애플리케이션 ID를 입력합니다.
 - Google에 대한 역할을 만드는 경우 대상 상자에 대상 사용자 이름을 입력합니다.
 - Amazon Cognito의 역할을 만드는 경우, Amazon Cognito 애플리케이션에 대해 만든 자격 증명 풀의 ID를 자격 증명 풀 ID 상자에 입력합니다.
6. (선택 사항) 애플리케이션 사용자가 역할에서 부여한 권한을 사용하기 위해 총족해야 하는 추가 조건을 만들려면 조건 추가(선택 사항)을 클릭합니다. 예를 들어, 특정 IAM 사용자 ID에만 AWS 리소스에 대한 액세스 권한을 부여하는 조건을 추가할 수 있습니다.
7. 웹 자격 증명 정보를 검토한 후 Next: Permissions(다음: 권한)을 선택합니다.
8. IAM은 계정의 AWS 관리형 또는 사용자 관리형 정책 목록을 포함합니다. 권한 정책을 사용하기 위한 정책을 선택하거나 정책 생성을 선택하여 새 브라우저 탭을 열고 완전히 새로운 정책을 생성합니다. 자세한 내용은 [IAM 정책 만들기\(콘솔\)](#) (p. 378) 절차의 4단계 단원을 참조하십시오. 정책을 생성하면 탭을 닫고 원래 탭으로 돌아갑니다. 웹 ID 사용자에게 부여하려는 권한 정책 옆의 확인란을 선택합니다. 원활 경우, 여기서 정책을 선택하지 않고 나중에 정책을 만들어서 역할에 연결할 수 있습니다. 기본적으로 역할은 권한이 없습니다.
9. (선택 사항) [권한 경계](#) (p. 317)로서 설정됨. 이는 고급 기능입니다.

Set permissions boundary(권한 경계 설정) 섹션을 열고 Use a permissions boundary to control the maximum role permissions(최대 역할 권한을 관리하기 위한 권한 경계 사용)을 선택합니다. 정책을 선택하여 권한 경계를 사용하십시오.

10. 다음: 태그 지정을 선택합니다.
11. (선택 사항) 태그를 키-값 페어로 연결하여 메타데이터를 역할에 추가합니다. IAM에서의 태그 사용에 대한 자세한 내용은 [IAM 엔터티 태그 지정](#) (p. 259) 단원을 참조하십시오.
12. [Next: Review]를 선택합니다.
13. 역할 이름에 역할 이름을 입력합니다. 역할 이름은 AWS 계정 내에서 고유해야 합니다. 대소문자는 구별하지 않습니다. 예를 들어, 이름이 **PRODROLE**과 **prodrole**, 두 가지로 지정된 역할을 만들 수는 없습니다. 기타 AWS 리소스가 역할을 참조할 수 있기 때문에 역할이 생성된 후에는 역할 이름을 편집할 수 없습니다.
14. (선택 사항) [Role description]에 새 역할에 대한 설명을 입력합니다.
15. 역할을 검토한 다음 [Create role]을 선택합니다.

SAML 2.0 연동을 위한 역할 생성(콘솔)

AWS 계정에 속하는 IAM 사용자를 생성하는 대신에 SAML 2.0 연동을 사용할 수 있습니다. 자격 증명 공급자(IdP)를 사용하면 AWS 외부의 사용자 자격 증명을 관리할 수 있고 이 외부 사용자 자격 증명에 계정의 AWS 리소스에 대한 사용 권한을 부여할 수 있습니다. 연동 및 자격 증명 공급자에 대한 자세한 내용은 [자격 증명 공급자 및 연동](#) (p. 161) 단원을 참조하십시오.

SAML 역할 생성하기 위한 사전 조건

SAML 2.0 연동을 위한 역할을 만들기 전에 먼저 다음 필수 단계를 완료해야 합니다.

SAML 2.0 연동을 위한 역할 생성을 준비하려면

1. SAML 기반 연동 역할을 만들기 전에 IAM에서 SAML 공급자를 만들어야 합니다. 자세한 내용은 [IAM SAML 자격 증명 공급자 생성 \(p. 177\)](#) 단원을 참조하십시오.
2. SAML 2.0 인증 사용자들이 맙을 역할에 대한 정책을 준비합니다. 다른 어떤 역할과 마찬가지로 SAML 연동을 위한 역할에는 2개의 정책이 포함됩니다. 하나는 역할을 위임할 사용자를 지정하는 신뢰 정책입니다. 다른 하나는 연동 사용자의 액세스가 허용 또는 거부되는 AWS 작업 및 리소스를 지정하는 권한 정책입니다.

역할에 대한 신뢰 정책을 생성할 시 애플리케이션에만 위임될 수 있는 역할을 보장하는 세 가지 값을 사용해야 합니다.

- Action 요소에 대해서는 `sts:AssumeRoleWithSAML` 작업을 사용하십시오.
- Principal 요소에 대해서는 `{"Federated": "ARNofIdentityProvider"}` 문자열을 사용하십시오. `ARNofIdentityProvider`를 Step 1에서 만든 [SAML 자격 증명 공급자 \(p. 167\)](#)의 ARN으로 바꿉니다.
- Condition 요소에 대해서는 `StringEquals` 조건을 사용하여 SAML 응답의 `saml:aud` 속성이 AWS에 대한 SAML 연동 엔드포인트와 일치하는지 테스트하십시오.

다음 예는 SAML 연동 사용자를 위해 설계된 신뢰 정책입니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": "sts:AssumeRoleWithSAML",  
         "Principal": {"Federated": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:saml-provider/PROVIDER-NAME"},  
         "Condition": {"StringEquals": {"SAML:aud": "https://signin.aws.amazon.com/saml"}}  
    ]  
}
```

보안 주체 ARN을 IAM에서 만든 SAML 공급자의 실제 ARN으로 바꿉니다. ARN에는 고유의 계정 ID와 공급자 이름이 있습니다.

SAML 역할 생성

사전 조건 단계를 완료한 후에는 SAML 기반 연동을 위한 역할을 생성합니다.

SAML 기반 연동을 위한 역할을 생성하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 역할을 선택한 후 역할 생성을 선택합니다.
3. SAML 2.0 federation(SAML 2.0 연동) 역할 유형을 선택합니다.
4. SAML provider(SAML 공급자)에서 역할의 공급자를 선택합니다.
5. SAML 2.0 액세스 수준 방법을 선택합니다.
 - Allow programmatic access only(프로그래밍 방식의 액세스만 허용)을 선택하여 AWS API 또는 AWS CLI에서 프로그래밍 방식으로 위임할 수 있는 역할을 만듭니다.
 - 그런 다음 Allow programmatic and AWS Management 콘솔 access(프로그래밍 방식 및 콘솔 액세스 허용)를 선택하여 콘솔에서 프로그래밍 방식으로 수임할 수 있는 역할을 생성합니다.

이렇게 생성된 두 역할은 비슷하지만 콘솔에서 위임할 수도 있는 역할에는 특정 조건을 포함하는 신뢰 정책을 포함합니다. 이 조건은 SAML 대상(SAML:aud 속성)이 SAML에 대한 AWS 로그인 엔드포인트 (<https://signin.aws.amazon.com/saml>)로 설정되도록 명시적으로 보장합니다.

6. 프로그래밍 방식 액세스를 위한 역할을 만드는 경우, 속성 목록에서 속성을 선택합니다. 그런 다음 값 상자에 역할에 포함시킬 값을 입력합니다. 이렇게 하면 지정한 속성을 포함하는 SAML 인증 응답(어설션)을 소유한 자격 증명 공급자의 사용자로 역할 액세스가 제한됩니다. 하나 이상의 속성을 지정해야 역할이 조직의 일부 사용자 집합으로 제한됩니다.

프로그래밍 방식 액세스 및 콘솔 액세스를 위한 역할을 만드는 경우, SAML:aud 속성이 자동으로 추가되고 AWS SAML 엔드포인트의 URL(<https://signin.aws.amazon.com/saml>)로 설정됩니다.

7. 신뢰 정책에 속성 관련 조건을 더 추가하려면 조건 추가(선택 사항)을 선택하고 추가 조건을 선택한 후 값을 지정합니다.

Note

이 목록에는 가장 많이 사용되는 SAML 속성을 포함합니다. IAM은 조건을 만드는 데 사용할 수 있는 추가 속성을 지원합니다. (지원되는 속성 목록은 [IAM JSON 정책 요소 참조 \(p. 498\)](#) 주제의 [SAML 연동에 사용할 수 있는 키](#) 단원을 참조하십시오.) 목록에는 없지만 지원되는 SAML 속성의 조건이 필요한 경우, 해당 조건을 수동으로 추가할 수 있습니다. 이렇게 하려면 역할을 만든 후 신뢰 정책을 편집합니다.

8. SAML 2.0 신뢰 정보를 검토한 후 Next: Permissions(다음: 권한)을 선택합니다.
9. IAM은 계정의 AWS 관리형 또는 사용자 관리형 정책 목록을 포함합니다. 권한 정책을 사용하기 위한 정책을 선택하거나 정책 생성을 선택하여 새 브라우저 탭을 열고 완전히 새로운 정책을 생성합니다. 자세한 내용은 [IAM 정책 만들기\(콘솔\) \(p. 378\)](#) 절차의 4단계 단원을 참조하십시오. 정책을 생성하면 탭을 닫고 원래 탭으로 돌아갑니다. 웹 ID 사용자에게 부여하려는 권한 정책 옆의 확인란을 선택합니다. 원활 경우, 여기서 정책을 선택하지 않고 나중에 정책을 만들어서 역할에 연결할 수 있습니다. 기본적으로 역할은 권한이 없습니다.
10. (선택 사항) [권한 경계 \(p. 317\)](#)로서 설정됨. 이는 고급 기능입니다.

Set permissions boundary(권한 경계 설정) 섹션을 열고 Use a permissions boundary to control the maximum role permissions(최대 역할 권한을 관리하기 위한 권한 경계 사용)을 선택합니다. 정책을 선택하여 권한 경계를 사용하십시오.

11. 다음: 태그 지정을 선택합니다.
12. (선택 사항) 태그를 키-값 페어로 연결하여 메타데이터를 역할에 추가합니다. IAM에서의 태그 사용에 대한 자세한 내용은 [IAM 엔터티 태그 지정 \(p. 259\)](#) 단원을 참조하십시오.
13. [Next: Review]를 선택합니다.
14. 역할 이름에 역할 이름을 입력합니다. 역할 이름은 AWS 계정 내에서 고유해야 합니다. 대소문자는 구별하지 않습니다. 예를 들어, 이름이 **PRODROLE**과 **prodrole**, 두 가지로 지정된 역할을 만들 수는 없습니다. 기타 AWS 리소스가 역할을 참조할 수 있기 때문에 역할이 생성된 후에는 역할 이름을 편집할 수 없습니다.
15. (선택 사항) [Role description]에 새 역할에 대한 설명을 입력합니다.
16. 역할을 검토한 다음 [Create role]을 선택합니다.

역할을 만든 후, AWS에 대한 정보로 자격 증명 공급자 소프트웨어를 구성하여 SAML 신뢰를 완료합니다. 이 정보는 연합된 사용자가 사용했으면 하는 역할을 포함합니다. 이를 가리켜 IdP와 AWS 간 신뢰 당사자 신뢰 구성이라고 합니다. 자세한 내용은 [신뢰 당사자 신뢰로 SAML 2.0 IdP를 구성하고 클레임 추가하기 \(p. 179\)](#) 단원을 참조하십시오.

액세스 권한 위임을 위한 정책의 예

다음 예제는 AWS 계정의 리소스에 대한 액세스를 AWS 계정에게 허용 또는 부여하는 방법을 보여줍니다. 이러한 예제 JSON 정책 문서를 사용하여 IAM 정책을 생성하는 방법에 대해 자세히 알아보려면 [the section called “JSON 탭에서 정책 만들기” \(p. 381\)](#) 단원을 참조하십시오.

주제

- 역할을 사용하여 다른 AWS 계정의 리소스에 대한 액세스 권한 위임하기 (p. 224)
- 정책을 사용하여 서비스에 대한 액세스 권한 위임 (p. 224)
- 리소스 기반의 정책을 사용하여 다른 계정의 Amazon S3 버킷에 대한 액세스 권한 위임하기 (p. 224)
- 리소스 기반의 정책을 사용하여 다른 계정의 Amazon SQS 대기열에 대한 액세스 권한 위임하기 (p. 225)
- 계정이 액세스 거부될 경우 액세스 권한을 위임할 수 없음 (p. 226)

역할을 사용하여 다른 AWS 계정의 리소스에 대한 액세스 권한 위임하기

IAM 역할을 사용하여 한 계정의 사용자에게 다른 계정의 AWS 리소스에 대한 액세스 권한을 부여하는 방법에 대한 자세한 내용은 [자습서: IAM 역할을 사용한 AWS 계정 간 액세스 권한 위임 \(p. 29\)](#) 단원을 참조하십시오.

Important

역할 신뢰 정책의 `Principal` 요소에 특정 역할이나 사용자에 대한 ARN을 포함할 수 있습니다. 정책을 저장하면 AWS가 ARN을 고유한 보안 주체 ID로 변환합니다. 그러면 누군가가 해당 역할 또는 사용자를 제거하고 다시 만들어 본인의 권한을 에스컬레이션할 위험을 완화할 수 있습니다. 일반적으로 콘솔에서는 이 ID가 보이지 않습니다. 신뢰 정책이 표시될 때 해당 ARN으로 다시 역변환되기 때문입니다. 그러나 해당 역할 또는 사용자를 삭제하면 관계가 깨집니다. 사용자 또는 역할을 다시 만들더라도 해당 정책이 더 이상 적용되지 않습니다. 신뢰 정책에 저장된 보안 주체 ID와 일치하지 않기 때문입니다. 이 경우 보안 주체 ID가 콘솔에 표시됩니다. AWS에서 더 이상 이를 ARN에 다시 매핑할 수 없기 때문입니다. 결과적으로 신뢰 정책의 `Principal` 요소에서 참조된 사용자 또는 역할을 삭제하고 다시 생성하는 경우, ARN을 바꾸도록 역할을 편집해야 합니다. 그러면 정책을 저장할 때 ARN이 새 보안 주체 ID로 변환됩니다.

정책을 사용하여 서비스에 대한 액세스 권한 위임

다음 예제는 역할에 연결할 수 있는 정책을 보여줍니다. 이 정책은 Amazon EMR 서비스와 AWS Data Pipeline 서비스가 역할을 수행할 수 있도록 합니다. 그러면 서비스가 해당 역할에 할당된 권한 정책에서 부여한 모든 작업을 수행할 수 있습니다(표시되지 않음). 여러 서비스 보안 주체를 지정할 때 `Service` 요소를 두 개 지정하면 안 됩니다. 하나만 지정할 수 있습니다. 대신 여러 서비스 보안 주체의 배열을 하나의 `Service` 요소의 값으로 사용합니다.

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

리소스 기반의 정책을 사용하여 다른 계정의 Amazon S3 버킷에 대한 액세스 권한 위임하기

이 예에서 계정 A는 리소스 기반 정책(Amazon S3 버킷 정책)을 사용하여 계정 B에게 계정 A의 S3 버킷에 액세스할 수 있는 완전한 권한을 부여합니다. 그런 다음 계정 B는 IAM 사용자 정책을 생성하여 계정 A의 버킷에 대한 해당 액세스 권한을 계정 B의 사용자 중 하나에게 위임합니다.

계정 A의 S3 버킷 정책은 다음 정책과 같을 수 있습니다. 이 예에서 계정 A의 S3 버킷 이름은 mybucket이고, 계정 B의 계정 번호는 111122223333입니다. 계정 B에서는 개별 사용자 또는 그룹을 지정하지 않고 오직 계정 자체만 지정할 뿐입니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Sid": "AccountBAccess1",  
         "Effect": "Allow",  
         "Principal": {"AWS": "111122223333"},  
         "Action": "s3:*",  
         "Resource": [  
             "arn:aws:s3:::mybucket",  
             "arn:aws:s3:::mybucket/*"  
         ]  
    }  
}
```

또는 계정 A가 Amazon S3 [액세스 제어 목록\(ACL\)](#)을 사용하여 계정 B에 S3 버킷 또는 버킷 내 단일 객체에 대한 액세스 권한을 부여할 수 있습니다. 이 경우 유일한 변경 사항은 계정 A가 계정 B에게 액세스 권한을 부여하는 방식입니다. 이 예의 다음 부분에서 설명한 것처럼 계정 B는 여전히 정책을 사용하여 계정 B의 IAM 그룹에게 액세스 권한을 위임합니다. S3 버킷과 객체에 대한 액세스를 제어하는 자세한 방법을 보려면 Amazon Simple Storage Service 개발자 가이드의 [액세스 제어](#)를 참조하십시오.

계정 B의 관리자는 다음 정책 샘플을 생성할 수 있습니다. 이 정책은 계정 B의 그룹 또는 사용자에게 읽기 액세스를 허용하며, 이전 정책은 B 계정에 대한 액세스 권한을 부여합니다. 하지만 계정 B의 개별 그룹과 사용자는 그룹 또는 사용자 정책이 리소스에 대한 권한을 명시적으로 부여할 때까지는 그 리소스에 액세스할 수 없습니다. 이 정책의 권한은 이전 교차 계정 정책에 있는 권한의 하위 집합에 불과할 수 있습니다. 계정 B는 첫 번째 정책에서 계정 A가 계정 B에게 부여한 권한보다 더 많은 권한을 자신의 그룹 또는 사용자에게 위임할 수 없습니다. 이 정책에서 Action 요소는 List 작업만을 허용하도록 명시적으로 정의되고 이 정책의 Resource 요소는 계정 A에 의해 적용되는 버킷 정책의 Resource와 일치합니다.

이 정책을 적용하기 위해 계정 B는 IAM을 사용하여 이 정책을 계정 B의 해당 사용자(또는 그룹)에게 연결합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": "s3>List*",  
         "Resource": [  
             "arn:aws:s3:::mybucket",  
             "arn:aws:s3:::mybucket/*"  
         ]  
    }  
}
```

리소스 기반의 정책을 사용하여 다른 계정의 Amazon SQS 대기열에 대한 액세스 권한 위임하기

다음 예에서 계정 A에는 계정 B에 대한 액세스 권한을 대기열에 부여하기 위해 대기열에 연결된 리소스 기반 정책을 사용하는 Amazon SQS 대기열이 있습니다. 그러면 계정 B는 IAM 그룹 정책을 사용하여 계정 B의 그룹에게 액세스 권한을 위임합니다.

다음 대기열 정책의 예는 계정 A의 queue1 대기열에서 2014년 11월 30일 정오부터 오후 3시까지만 SendMessage 및 ReceiveMessage 작업을 수행할 수 있는 권한을 계정 B에 부여합니다. 계정 B의 계정 번호는 1111-2222-3333입니다. 계정 A는 Amazon SQS를 사용하여 이 정책을 적용합니다.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    "Effect": "Allow",
    "Principal": {"AWS": "111122223333"},
    "Action": [
        "sns:SendMessage",
        "sns:ReceiveMessage"
    ],
    "Resource": ["arn:aws:sns:*:123456789012:queue1"],
    "Condition": {
        "DateGreaterThan": {"aws:CurrentTime": "2014-11-30T12:00Z"},
        "DateLessThan": {"aws:CurrentTime": "2014-11-30T15:00Z"}
    }
}
```

계정 B의 그룹에게 액세스 권한을 위임하기 위한 계정 B의 정책은 다음 예와 같을 수 있습니다. 계정 B는 IAM을 사용하여 이 정책을 그룹(또는 사용자)에게 연결합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        "Effect": "Allow",
        "Action": "sns:*",
        "Resource": "arn:aws:sns:*:123456789012:queue1"
    ]
}
```

앞의 IAM 사용자 정책에 대한 예시에서 계정 B는 와일드카드를 사용하여 해당 사용자에게 계정 A의 대기열에서 모든 Amazon SQS 작업을 수행할 수 있는 액세스 권한을 부여했습니다. 하지만 계정 B는 액세스 권한이 부여된 범위까지만 액세스 권한을 위임할 수 있습니다. 두 번째 정책이 있는 계정 B 그룹은 2014년 11월 30일 정오부터 오후 3시까지만 대기열에 액세스할 수 있습니다. 사용자는 계정 A 및 Amazon SQS 대기열 정책에 정의된 대로 SendMessage 및 ReceiveMessage 작업만 수행할 수 있습니다.

계정이 액세스 거부될 경우 액세스 권한을 위임할 수 없음

다른 계정에서 사용자의 상위 계정에 대한 액세스를 명시적으로 거부할 경우 AWS 계정은 다른 계정의 리소스에 대한 액세스 권한을 위임할 수 없습니다. 이 거부는 사용자가 액세스 권한을 부여하는 기존 정책을 가지고 있는지 여부에 상관없이 해당 계정의 사용자에게 전파됩니다.

계정 A가 계정 A의 S3 버킷에 계정 A의 버킷에 대한 계정 B의 액세스를 명시적으로 거부하는 버킷 정책을 계정 A의 S3 버킷에 작성하는 경우를 예로 들어 보겠습니다. 계정 B는 계정 B의 사용자에게 계정 A의 버킷에 대한 액세스 권한을 부여하는 IAM 사용자 정책을 작성합니다. 계정 A의 S3 버킷에 적용된 명시적 거부는 계정 B의 사용자에게 전파되고 계정 B의 사용자에게 액세스 권한을 부여하는 IAM 사용자 정책보다 우선합니다. (권한 평가 방식에 대한 자세한 내용은 [정책 평가 로직 \(p. 531\)](#)을 참조하십시오.)

계정 A의 버킷 정책은 다음과 같을 수 있습니다. 이 예에서 계정 A의 S3 버킷 이름은 mybucket이고, 계정 B의 계정 번호는 1111-2222-3333입니다. 계정 A는 Amazon S3를 사용하여 이 정책을 적용합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        "Sid": "AccountBDeny",
        "Effect": "Deny",
        "Principal": {"AWS": "111122223333"},
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::mybucket/*"
    ]
}
```

이 명시적 거부는 계정 A의 S3 버킷에 액세스할 수 있는 권한을 제공하는 계정 B의 모든 정책을 재정의합니다.

IAM 역할 사용

IAM 사용자, 애플리케이션 또는 서비스에서 이전에 생성한 역할을 사용하려면 그 역할로 전환할 수 있는 권한을 부여해야 합니다. IAM 사용자 그룹 중 하나 또는 사용자 자신에게 추가된 어떤 정책도 필요한 권한을 부여하는 데 사용할 수 있습니다. 이 단원에서는 역할 사용 권한을 사용자에게 부여하는 방법과 사용자가 AWS Management 콘솔, Windows PowerShell용 도구, AWS Command Line Interface(AWS CLI) 및 [AssumeRole API](#)를 사용하여 원하는 역할로 전환하는 방법을 살펴보겠습니다.

Important

IAM 콘솔 대신 프로그래밍 방식으로 역할을 생성하는 경우에는 사용자의 선택에 따라 최대 64자인 `RoleName`뿐만 아니라 최대 512자인 `Path`도 추가할 수 있습니다. 그러나 AWS 콘솔에서 `Switch Role(역할 전환)` 기능이 있는 역할을 사용하려면 `Path`와 `RoleName`을 합해 64자를 초과할 수 없습니다.

AWS Management 콘솔에서 역할을 전환할 수 있습니다. AWS CLI 또는 API 작업을 호출하거나 사용자 지정 URL을 사용하여 역할을 위임할 수 있습니다. 사용한 방법에 따라 역할을 수임할 수 있는 사용자와 역할 세션의 지속 가능 기간이 결정됩니다.

역할 사용을 위한 방법 비교

방법	역할을 위임할 수 있는 사용자	자격 증명의 수명을 지정하는 방법	자격 증명의 수명 (최소 최대 기본)
AWS Management 콘솔	IAM 사용자(역할 전환을 통해 (p. 233))	없음	1시간 1시간 1시간
<code>assume-role</code> CLI 또는 AssumeRole API 작업	IAM 사용자 또는 역할 ¹	<code>duration-seconds</code> CLI 또는 <code>DurationSeconds</code> API 파라미터	15분 최대 세션 기간 설정 ² 1시간
<code>assume-role-with-saml</code> CLI 또는 AssumeRoleWithSAML API 작업	SAML을 사용하여 인증된 모든 사용자	<code>duration-seconds</code> CLI 또는 <code>DurationSeconds</code> API 파라미터	15분 최대 세션 기간 설정 ² 1시간
<code>assume-role-with-web-identity</code> CLI 또는 AssumeRoleWithWebIdentity API 작업	웹 자격 증명 공급자를 사용하여 인증된 모든 사용자	<code>duration-seconds</code> CLI 또는 <code>DurationSeconds</code> API 파라미터	15분 최대 세션 기간 설정 ² 1시간
AssumeRole로 구성된 콘솔 URL (p. 188)	IAM 사용자 또는 역할	URL의 <code>SessionDuration</code> HTML 파라미터	15분 12시간 1시간
AssumeRoleWithSAML로 구성된 콘솔 URL (p. 188)	SAML을 사용하여 인증된 모든 사용자	URL의 <code>SessionDuration</code> HTML 파라미터	15분 12시간 1시간
AssumeRoleWithWebIdentity로 구성된 콘솔 URL (p. 188)	웹 자격 증명 공급자를 사용하여 인증된 모든 사용자	URL의 <code>SessionDuration</code> HTML 파라미터	15분 12시간 1시간

¹ 하나의 역할이 자격 증명을 사용하여 다른 역할을 위임하는 것을 [역할 함께 루기 \(p. 154\)](#)라고 합니다. 역할 함께 루기를 사용하는 경우 새 자격 증명의 유효 기간은 최대 1시간으로 제한됩니다.

² 최대 세션 기간은 콘솔 AWS CLI 또는 API에서 역할에 적용할 수 있는 설정입니다. 이 설정은 CLI 또는 API에서 역할을 수임할 때 역할에 대한 최대 세션 기간을 지정합니다. 이 설정에는 1~12시간의 값을 지정할 수 있습니다. 최대 세션 기간 설정에 대한 자세한 내용은 [역할 변경 \(p. 247\)](#) 단원을 참조하십시오. 이 설정은 역할 자격 증명을 얻을 때 요청할 수 있는 최대 세션 기간을 결정합니다. 예를 들어 [AssumeRole*](#) API 작업을 사용하여 역할을 위임할 때 DurationSeconds 파라미터를 사용하여 세션 길이를 지정할 수 있습니다. 이 파라미터를 사용하여 역할 세션 기간을 900초(15초)에서 해당 역할에 대한 최대 세션 기간 설정까지 지정합니다. 역할에 대한 최댓값을 확인하는 방법을 알아보려면 이 페이지 뒷부분에 나오는 [역할에 대한 최대 세션 기간 설정 보기 \(p. 228\)](#) 단원을 참조하십시오.

주제

- [역할에 대한 최대 세션 기간 설정 보기 \(p. 228\)](#)
- [사용자에 대한 역할 전환 권한 부여 \(p. 229\)](#)
- [사용자에게 AWS 서비스에 역할을 전달할 권한 부여 \(p. 231\)](#)
- [역할 전환\(콘솔\) \(p. 233\)](#)
- [IAM 역할로 전환하기\(AWS CLI\) \(p. 235\)](#)
- [IAM 역할로 전환하기\(Windows PowerShell용 도구\) \(p. 236\)](#)
- [IAM 역할\(AWS API\)로 전환하기 \(p. 238\)](#)
- [IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여하기 \(p. 240\)](#)
- [IAM 역할의 임시 보안 자격 증명 취소 \(p. 245\)](#)

역할에 대한 최대 세션 기간 설정 보기

AWS CLI 또는 API 작업을 사용하여 역할을 위임하는 경우 DurationSeconds 파라미터에 대한 값을 지정할 수 있습니다. 이 파라미터를 사용하여 역할 세션 기간을 900초(15초)에서 해당 역할에 대한 Maximum CLI/API session duration(최대 CLI/API 세션 기간) 설정까지 지정할 수 있습니다. 이 파라미터를 지정하기 전에 역할에 대한 이 설정을 확인해야 합니다. DurationSeconds 파라미터의 값을 최대 설정보다 높게 지정하면 작업에 실패합니다.

역할의 최대 세션 기간을 보려면(콘솔)

1. IAM 콘솔의 탐색 창에서 [Roles]를 선택합니다.
2. 보려는 역할의 이름을 선택합니다.
3. Maximum CLI/API session duration(최대 CLI/API 세션 기간) 옆에서 AWS CLI 또는 API 작업에서 지정할 수 있는 최대 세션 길이를 확인합니다.

역할의 최대 세션 기간 설정을 보려면(AWS CLI)

1. 수임할 역할의 이름을 모르는 경우 다음 명령을 실행하여 계정의 역할을 나열합니다.
 - `aws iam list-roles`
2. 역할에 대한 현재 최대 세션 기간 설정을 확인하려면 다음 명령을 실행합니다. 그런 다음 최대 세션 기간 파라미터를 확인합니다.
 - `aws iam get-role`

역할의 최대 세션 기간 설정을 보려면(AWS API)

1. 수임할 역할의 이름을 모르는 경우 다음 연산을 호출하여 계정의 역할을 나열합니다.
 - `ListRoles`

2. 역할에 대한 현재 최대 세션 기간 설정을 확인하려면 다음 연산을 실행합니다. 그런 다음 최대 세션 기간 파라미터를 확인합니다.

- [GetRole](#)

사용자에 대한 역할 전환 권한 부여

[교차 계정 액세스가 가능한 역할을 생성하려면 \(p. 203\)](#) 역할 및 리소스가 저장된 계정(신뢰하는 계정)에서 사용자가 저장된 계정(신뢰 받는 계정)으로 신뢰를 구성합니다. 이 작업을 수행하려면 역할의 신뢰 정책에서 신뢰할 수 있는 계정 번호를 Principal로 지정합니다. 이렇게 하면 신뢰할 수 있는 계정의 잠재적 사용자라면 누구든지 역할을 위임할 수 있습니다. 구성이 완료하려면 신뢰 받는 계정의 관리자는 계정에 속한 특정 그룹 또는 사용자에게 역할 전환 권한을 부여해야 합니다.

사용자에게 역할 전환 권한을 부여하려면 새로운 사용자 정책을 생성하거나 기존 정책을 편집하여 필요한 요소를 추가해야 합니다. 그런 다음 이미 세부 정보가 모두 작성되어 있는 Switch Role(역할 전환) 페이지로 이동할 수 있는 링크를 사용자에게 보낼 수 있습니다. 그 밖에도 계정 ID 번호 또는 역할이 저장된 계정 별칭 및 역할 이름을 사용자에게 제공할 수 있습니다. 이제 사용자는 Switch Role(역할 전환) 페이지로 이동하여 세부 정보를 직접 입력합니다. 사용자의 역할 전환 방법에 대한 세부 정보는 [역할 전환\(콘솔\) \(p. 233\)](#)을 참조하십시오.

IAM 사용자로 로그인할 때만 역할을 바꿀 수 있다는 점에 유의하십시오. AWS 계정 루트 사용자로 로그인할 때는 역할을 바꿀 수 없습니다.

Important

AWS Management 콘솔에서의 역할을 [ExternalId \(p. 206\)](#) 값이 필요한 역할로 전환할 수 없습니다. ExternalId 파라미터를 지원하는 AssumeRole API를 호출해야만 이러한 역할로 변경할 수 있습니다.

참고

- 이 주제는 사용자에 대한 정책들을 다루고 있는데, 이는 AWS가 사용자에게 작업을 완수할 수 있는 권한을 최종적으로 부여하고 있기 때문입니다. 그러나 [개별 사용자에게 직접 권한을 부여하지 않는 것이 최상의 관행 \(p. 44\)](#)입니다. 관리를 더 쉽게 하려면 IAM 그룹에 정책을 배정하고 권한을 부여한 다음 적절한 그룹들의 구성원인 사용자들을 생성하도록 권장합니다.
- AWS Management 콘솔에서 역할을 전환하는 경우, 콘솔은 항상 원래 자격 증명을 사용하여 전환을 승인합니다. 이는 IAM 사용자, SAML 연동 역할 또는 웹 자격 증명 연동 역할 중 어느 것으로 로그인하는지 여부에 관계없이 적용됩니다. 예를 들어, RoleA로 전환하는 경우 원래 사용자 자격 증명 또는 연동 역할 자격 증명을 사용하여 RoleA를 부여할지 여부를 결정합니다. RoleA를 사용하는 중에 RoleB로 전환하려는 경우, RoleA의 자격 증명이 아닌, 원래 사용자 또는 연동 역할 자격 증명이 인증에 사용됩니다.

주제

- [정책 생성 또는 편집 \(p. 229\)](#)
- [사용자에 대한 정보 제공 \(p. 230\)](#)

정책 생성 또는 편집

역할을 맡기 위한 사용자 권한을 부여하는 정책에는 다음에 적용되는 Allow 문이 포함되어야 합니다.

- `sts:AssumeRole` 작업
- Resource 요소에 있는 역할의 ARN(Amazon Resource Name)

다음 예제를 참조하십시오. 그 정책을 가져오는(그룹 멤버십 또는 직접 첨부를 통해) 사용자들은 지정된 역할로 전환하도록 허용됩니다.

Note

Resource가 *로 설정된 경우에는 사용자 계정을 신뢰하는 어떤 계정의 어떤 역할이라도 사용자가 수임할 수 있다는 점에 유의하십시오(역할의 신뢰 정책은 사용자의 계정을 Principal로 지정합니다). [최소 권한의 원칙](#)에 따라 사용자에게 필요한 역할에 대해서만 완전한 ARN을 지정하는 것이 좋습니다.

다음 예제에서는 단 한 개의 계정에서 사용자가 역할을 맡을 수 있는 정책을 보여 줍니다. 또한 이 정책은 와일드카드(*)를 사용하여 역할 이름이 Test 문자로 시작할 경우에만 사용자가 역할을 전환할 수 있도록 지정합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "sts:AssumeRole",  
            "Resource": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:role/Test*"  
        }  
    ]  
}
```

Note

이 역할이 사용자에게 부여하는 권한은 사용자에게 이미 부여된 권한에 추가되지는 않습니다. 사용자가 어떤 역할로 전환할 때는 일시적으로 자신의 원래 권한을 버리고 그 역할이 부여하는 권한으로 갈아탄다. 사용자가 역할을 끝내면 원래 사용자 권한이 자동으로 회복됩니다. 사용자 권한에서 Amazon EC2 인스턴스 작업을 허용하지만, 역할의 권한 정책이 해당 권한을 부여하지 않는 경우를 예로 들어 보겠습니다. 이 경우 역할을 사용할 때 사용자가 콘솔에서 Amazon EC2 인스턴스 작업을 수행할 수 없습니다. 또한 AssumeRole을 통해 받은 임시 자격 증명은 프로그래밍 방식으로 Amazon EC2 인스턴스 작업을 수행할 수 없습니다.

사용자에 대한 정보 제공

역할을 만들어 이 역할로 전환하는 권한을 사용자에게 부여한 후, 사용자에게 다음을 제공해야 합니다.

- 역할 이름
- 해당 역할을 포함하는 계정 ID 번호 또는 계정 별칭

계정 ID와 역할 이름이 미리 구성되어 있는 링크를 사용자에게 보내주는 것이 더 간편합니다. 이 역할 링크는 역할 생성 마법사의 마지막 페이지, 또는 교차 계정 역할의 Role Summary(역할 요약) 페이지에 있습니다.

Note

AWS CLI, Windows PowerShell용 도구, 또는 AWS API로 역할을 생성하는 경우에는, 이름뿐만 아니라 경로도 지닌 역할을 생성할 수 있습니다. 이렇게 하기 위해서는 AWS Management 콘솔의 역할 전환 페이지에 입력할 수 있도록 사용자에게 전체 경로와 역할 이름을 제공해야 합니다. 예: division_abc/subdivision_efg/role_XYZ.

Important

IAM 콘솔 대신 프로그래밍 방식으로 역할을 생성하는 경우에는 RoleName 외에 Path(최대 512자)도 추가할 수 있습니다. RoleName 길이는 최대 64자입니다. 그러나 AWS 콘솔에서 역할 전환 기능이 있는 역할을 사용하려면 Path와 RoleName을 합해 64자를 초과할 수 없습니다.

다음 형식을 사용해 링크를 수동으로 구축할 수도 있습니다. 다음과 같이 계정 ID 또는 별칭과 역할 이름을 요청의 파라미터 2개로 대치하십시오.

```
https://signin.aws.amazon.com/switchrole?  
account=YourAccountIDorAliasHere&roleName=pathIfAny/YourRoleNameHere
```

사용자가 [역할 전환\(콘솔\)](#) (p. 233) 주제에서 프로세스를 살펴볼 수 있도록 기회를 제공하는 것이 좋습니다.

Note

보안상의 목적으로 AWS CloudTrail을 사용해 역할 전환을 감사할 수 있습니다. CloudTrail이 계정에서 활성화되어 있는 경우 IAM이 역할의 임시 보안 자격 증명을 사용해 수행되는 작업을 로깅합니다. 자세한 내용은 AWS CloudTrail User Guide의 [CloudTrail이벤트 참조](#)를 참조하십시오.

사용자에게 AWS 서비스에 역할을 전달할 권한 부여

다수의 AWS 서비스를 구성하려면 IAM 역할을 서비스에 전달해야 합니다. 그러면 서비스가 나중에 역할을 수임하고 사용자 대신 작업을 수행할 수 있습니다. 서비스가 역할을 수임할 때마다 아니라 설정 중에 한 번만 역할을 서비스에 전달해야 합니다. 예를 들어 Amazon EC2 인스턴스에서 실행 중인 애플리케이션이 있다고 가정합니다. 해당 애플리케이션에는 인증을 위한 임시 자격 증명과 AWS에서 작업을 수행할 수 있는 애플리케이션을 승인할 권리가 필요합니다. 애플리케이션을 설정할 때는 역할을 EC2에 전달하여 해당 자격 증명을 제공하는 인스턴스와 함께 사용해야 합니다. IAM 정책을 역할에 연결하여 인스턴스에서 실행 중인 애플리케이션에 대한 권한을 정의합니다. 애플리케이션은 역할이 허용하는 작업을 수행해야 할 때마다 역할을 수임합니다.

AWS 서비스에 역할(및 그 권한)을 전달하려면 사용자에게 서비스에 역할을 전달할 권리가 있어야 합니다. 이를 통해 관리자는 승인된 사용자만 권한이 부여된 역할을 통해 서비스를 구성하도록 할 수 있습니다. 사용자가 AWS 서비스에 역할을 전달하도록 하려면 해당 사용자의 IAM 사용자, 역할 또는 그룹에 PassRole 권한을 부여해야 합니다.

서비스 연결 역할을 생성하는 경우 해당 역할을 서비스에 전달할 권리도 있어야 합니다. 일부 서비스는 서비스에서 작업을 수행할 때 계정에 서비스 연결 역할을 자동으로 생성합니다. 예를 들어 Amazon EC2 Auto Scaling에서는 사용자가 Auto Scaling 그룹을 처음으로 생성할 때 사용자를 대신해 AWS*ServiceRoleForAutoScaling* 서비스 연결 역할을 생성합니다. PassRole 권한 없이 Auto Scaling 그룹을 생성하려고 하면 오류가 발생합니다. 서비스 연결 역할을 지원하는 서비스를 알아보려면 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하십시오. 서비스에서 작업 수행 시 자동으로 서비스 연결 역할을 생성하는 서비스를 알아보려면 예 링크를 선택하고 해당 서비스에 대한 서비스 연결 역할 설명서를 확인합니다.

사용자는 역할을 사용하여 서비스에 권한을 할당하는 API 작업에서 파라미터로 역할 ARN을 전달할 수 있습니다. 그런 다음 서비스는 해당 사용자에게 iam:PassRole 권한이 있는지 확인합니다. 사용자가 승인된 역할만 전달하도록 제한하려면 IAM 정책 문의 Resources 요소로 iam:PassRole 권한을 필터링하면 됩니다.

예 1

인스턴스를 시작한 후 사용자에게 Amazon EC2 서비스에 승인된 역할 집합을 전달할 수 있는 권한을 부여하려 한다고 가정하겠습니다. 다음 세 가지 요소가 필요합니다.

- 역할이 수행할 수 있는 작업을 결정하는, 역할에 연결된 IAM 권한 정책입니다. 역할이 수행해야 하는 작업 및 역할이 그러한 작업을 수행하는 데 필요한 리소스만으로 권한을 한정할 수 있습니다. AWS 관리형 또는 고객이 생성한 IAM 권한 정책을 사용할 수 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": [ "A list of the permissions the role is allowed to use" ],  
        "Resource": [ "A list of the resources the role is allowed to access" ]  
    }  
}
```

- 서비스에서 역할을 위임하도록 허용하는 역할에 대한 신뢰 정책입니다. 예를 들어, `UpdateAssumeRolePolicy` 작업이 있는 역할에 다음과 같은 신뢰 정책을 연결할 수 있습니다. 이 신뢰 정책을 통해 Amazon EC2는 해당 역할 및 해당 역할과 연결된 권한을 사용할 수 있습니다.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Sid": "TrustPolicyStatementThatAllowsEC2ServiceToAssumeTheAttachedRole",
        "Effect": "Allow",
        "Principal": { "Service": "ec2.amazonaws.com" },
        "Action": "sts:AssumeRole"
    }
]
```

- 사용자가 승인된 역할만 전달하도록 허용하는 IAM 사용자에 연결된 IAM 권한 정책입니다. 사용자가 전달 할 역할의 세부 정보를 얻을 수 있도록 iam:PassRole은 일반적으로 iam:GetRole과 함께 제공됩니다. 이 예제에서 사용자는 지정된 계정에 있으며 다음과 같이 이름이 EC2-roles-for-XYZ-로 시작하는 역 할만 전달할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::<account-id>:role/EC2-roles-for-XYZ-*"
        }
    ]
}
```

이제 사용자는 할당된 역할로 Amazon EC2 인스턴스를 시작할 수 있습니다. 이 인스턴스에서 실행되는 애플리케이션은 인스턴스 프로파일 메타데이터를 통해 역할의 임시 자격 증명에 액세스할 수 있습니다. 역할과 연결된 권한 정책은 인스턴스가 수행할 수 있는 작업을 결정합니다.

예 2

Amazon Relational Database Service(Amazon RDS)는 확장 모니터링이라는 기능을 지원합니다. 이 기능을 사용하면 Amazon RDS에서 에이전트를 사용하여 데이터베이스 인스턴스를 모니터링할 수 있습니다. 또한 Amazon RDS에서 Amazon CloudWatch Logs에 측정치를 기록할 수도 있습니다. 이 기능을 사용하려면 서비스 역할을 생성하여 로그에 대한 측정치를 모니터링하고 작성할 수 있는 권한을 Amazon RDS에 부여해야 합니다.

Amazon RDS Enhanced Monitoring에 대한 역할을 만들려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 역할을 선택한 다음 역할 생성을 선택합니다.
3. AWS 서비스 역할 유형을 선택한 후 확장 모니터링을 위한 Amazon RDS 역할(Amazon RDS Role for Enhanced Monitoring) 서비스를 선택합니다. 그런 다음 [Next: Permissions]를 선택합니다.
4. AmazonRDSEnhancedMonitoringRole, 권한 정책을 선택합니다.
5. 다음: 태그 지정을 선택합니다.
6. (선택 사항) 태그를 키-값 페어로 연결하여 메타데이터를 사용자에게 추가합니다. IAM에서의 태그 사용에 대한 자세한 정보는 [IAM 엔터티 태그 지정 \(p. 259\)](#) 단원을 참조하십시오.
7. [Next: Review]를 선택합니다.
8. 역할 이름에서 이 역할의 목적을 나타내는 역할 이름을 입력합니다. 역할 이름은 AWS 계정 내에서 고유해야 합니다. 대소문자는 구별하지 않습니다. 예를 들어, 이름이 **PRODROLE**과 **prodrole**, 두 가지로 지정된 역할을 만들 수는 없습니다. 다양한 주체가 역할을 참조할 수 있기 때문에 역할이 생성된 후에는 역할 이름을 편집할 수 없습니다.
9. (선택 사항) [Role description]에 새 역할에 대한 설명을 입력합니다.
10. 역할을 검토한 다음 [Create role]을 선택합니다.

그러면 역할이 `monitoring.rds.amazonaws.com` 서비스에 해당 역할을 수입할 권한을 부여하는 신뢰 정책을 자동으로 얻습니다. 그러면 Amazon RDS는 `AmazonRDSEnhancedMonitoringRole` 정책에서 허용하는 모든 작업을 수행할 수 있습니다.

사용자가 Enhanced Monitoring을 활성화하려면 이 사용자가 역할을 전달하도록 허용하는 다음과 같은 문이 포함된 정책이 필요합니다. 계정 번호를 사용하여 역할 이름을 3단계에서 입력한 이름으로 바꿉니다.

```
{  
    "Sid": "PolicyStatementToAllowUserToPassOneSpecificRole",  
    "Effect": "Allow",  
    "Action": [ "iam:PassRole" ],  
    "Resource": "arn:aws:iam:::role/RDS-Monitoring-Role"  
}
```

이 문을 다른 정책의 문과 결합하거나 고유 정책에 포함시킬 수 있습니다. 사용자가 RDS-로 시작하는 모든 역할을 전달할 수 있도록 지정하려면 다음과 같이 리소스 ARN의 역할 이름을 와일드카드로 바꿉니다.

```
"Resource": "arn:aws:iam:::role/RDS-*"
```

역할 전환(콘솔)

역할은 필요한 AWS 리소스에 액세스하는 데 사용할 수 있는 일련의 권한을 지정합니다. 이러한 의미에서 [AWS Identity and Access Management](#)(IAM) 사용자와 비슷하다고 할 수 있습니다. 사용자로 로그인할 때는 특정 권한이 부여됩니다. 하지만 역할로 로그인하지는 못하기 때문에 일단 로그인한 후에 역할로 전환할 수 있습니다. 이 경우 초기의 사용자 권한은 잠시 무효화되고 역할에게 할당된 권한이 부여됩니다. 역할은 자신의 계정이나 그 밖에 다른 AWS 계정에도 속할 수 있습니다. 역할, 역할의 이점 및 생성 방법에 대한 자세한 내용은 다음([IAM 역할 \(p. 153\)](#) 및 [IAM 역할 생성 \(p. 202\)](#))을 참조하십시오.

Important

IAM 사용자의 권한과 전환 대상인 역할의 권한은 누적되지 않습니다. 한 번에 오직 하나의 권한 집합만이 활성화됩니다. 어떤 역할로 전환할 때 사용자 권한은 일시적으로 포기하고 역할에 할당된 권한을 가지고 작업합니다. 역할을 끝내면 사용자 권한이 자동으로 회복됩니다.

기본적으로 AWS Management 콘솔 세션은 한 시간 동안 지속됩니다.

AWS Management 콘솔에서 역할을 전환하는 경우, 콘솔은 항상 원래 자격 증명을 사용하여 전환을 승인합니다. 이는 IAM 사용자, SAML 연동 역할 또는 웹 자격 증명 연동 역할 중 어느 것으로 로그인하는지 여부에 관계없이 적용됩니다. 예를 들어, RoleA로 전환하는 경우 IAM에서는 원래 사용자 자격 증명 또는 연동 역할 자격 증명을 사용하여 RoleA를 부여할지 여부를 결정합니다. RoleA를 사용하는 중에 RoleB로 전환하는 경우에도 IAM에서는 RoleA의 자격 증명이 아닌, 원래 사용자 자격 증명 또는 연동된 역할 자격 증명을 사용하여 전환을 승인합니다.

이 단원에서는 IAM 콘솔을 사용한 역할 전환 방법을 설명합니다.

- IAM 사용자로 로그인할 때만 역할을 바꿀 수 있습니다. AWS 계정 루트 사용자로 로그인할 경우 역할을 바꿀 수 없습니다.
- 관리자가 링크를 제공하는 경우 다음 절차에서 링크를 선택하여 [Step 5](#) 단계로 넘어갑니다. 링크를 클릭하면 적절한 웹 페이지로 이동하고 계정 ID(또는 별칭)와 역할 이름이 채워집니다.
- 링크를 수동으로 구성한 후 다음 절차의 [Step 5](#) 단계로 건너뛸 수 있습니다. 링크를 구성하려면 다음 형식을 사용합니다.

```
https://signin.aws.amazon.com/switchrole?  
account=account\_id\_number&roleName=role\_name&displayname=text\_to\_display
```

여기서 다음 텍스트를 바꿉니다.

- **account_id_number**-관리자가 제공한 12자리 계정 식별자. 또는 URL에 계정 ID 대신 계정 이름이 포함되도록 관리자가 계정 별칭을 생성할 수 있습니다. 자세한 내용은 [AWS 계정 ID 및 별칭 \(p. 55\)](#)을 참조하십시오.

- **role_name**—수입하려는 역할의 이름입니다. 이 이름은 역할의 ARN 끝에서 가져올 수 있습니다. 예를 들어 역할 ARN: TestRole에서 수입하려는 역할의 이름은 namearn:aws:iam::403299380220:role/TestRole입니다.
- (선택 사항) **text_to_display**—이 역할이 활성화되었을 때 탐색 표시줄에 사용자 이름 대신 표시되도록 하고 싶은 텍스트를 입력할 수 있습니다.
- 관리자가 제공하는 정보를 사용하여 아래 절차를 통해 역할을 수동으로 전환할 수 있습니다.

역할을 수입할 때 발생할 수 있는 일반적인 문제를 해결하려면 [역할을 위임할 수 없음 \(p. 469\)](#) 단원을 참조하십시오.

역할을 전환하려면(콘솔)

1. IAM 사용자로 AWS Management 콘솔에 로그인하여 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. IAM 콘솔에서 상단 오른쪽 모서리에 있는 탐색 표시줄에서 사용자 이름을 선택합니다. 일반적인 형식은 **username@account_ID_number_or_alias**입니다.
3. [Switch Role]을 선택합니다. 이 옵션을 처음 선택하면 자세한 정보를 제공하는 페이지가 나타납니다. 그 정보를 읽은 후에 역할 전환(Switch Role)을 클릭합니다. 브라우저 쿠키를 청소하면 이 페이지가 다시 나타나게 할 수 있습니다.
4. 역할 전환 페이지에서 계정 ID 번호 또는 계정 별칭 및 관리자가 제공한 역할 이름을 입력합니다.

Note

관리자가 경로를 포함하여 역할을 생성한 경우(예: division_abc/subdivision_efg/roleToDoX)에는 역할 상자에 전체 경로와 이름을 입력해야 합니다. 역할 이름만 입력하는 경우 또는 결합된 Path 및 RoleName이 64자를 초과하는 경우 역할 전환에 실패합니다. 이것은 역할 이름을 저장하는 브라우저 쿠키의 한계입니다. 이러한 경우 관리자에게 문의해 경로 및 역할 이름의 크기를 줄여 달라고 요청하십시오.

5. (선택 사항) 이 역할이 활성화되었을 때 탐색 표시줄에 사용자 이름 대신 표시되도록 하려는 텍스트를 입력할 수 있습니다. 이름은 계정 및 역할 정보에 따라 다르게 제시되지만 특별한 의미를 갖도록 직접 변경하는 것도 가능합니다. 또한, 표시되는 이름이 돋보이도록 색상을 선택할 수도 있습니다. 이름과 색상은 이 역할이 활성화되어 권한이 변경되는 시점을 다시 한 번 알려줍니다. 예를 들어 테스트 환경에 대한 액세스 권한을 부여하는 역할에 대해 표시 이름을 **test**로 지정하고 색상은 녹색으로 선택합니다. 프로덕션에 대한 액세스 권한을 부여하는 역할에 대해서는 표시 이름을 **Production**으로 지정하고 색상은 빨간색으로 선택합니다.
6. [Switch Role]을 선택합니다. 표시 이름과 색상이 탐색 표시줄에 사용자 이름 대신 나타나고, 역할에서 부여하는 권한을 사용하여 시작할 수 있습니다.

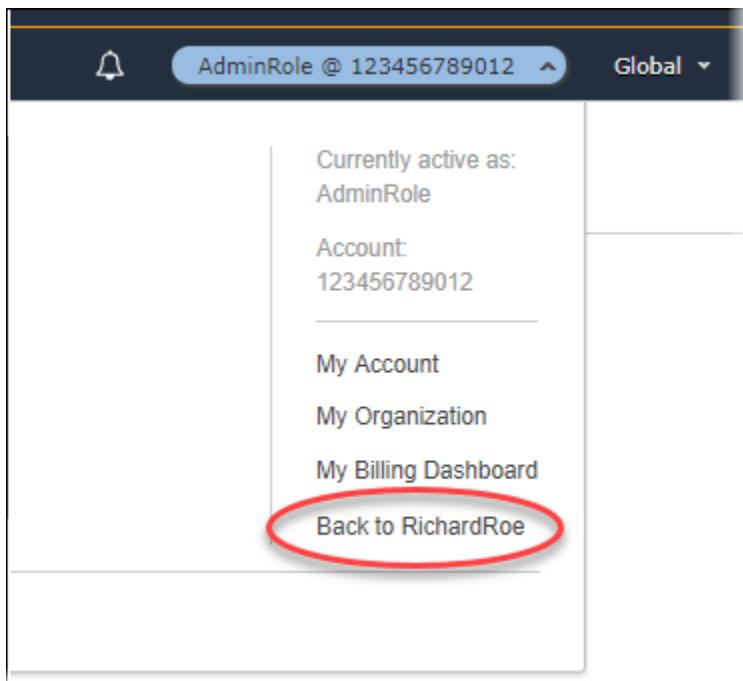
도움말

마지막으로 사용한 몇 가지 역할은 [] 메뉴에 표시됩니다. 다음에 이 중 하나로 역할을 전환해야 할 때는 원하는 역할을 선택하기만 하면 됩니다. 역할이 자격 증명 메뉴에 표시되지 않으면 계정 및 역할 정보를 수동으로 입력하면 됩니다.

역할 사용을 중지하려면(콘솔)

1. IAM 콘솔의 탐색 모음 오른쪽 위쪽에서 역할의 표시 이름을 선택합니다. 일반적인 형식은 **rolename@account_ID_number_or_alias**입니다.
2. Back to **username(username으로 돌아가기)**을 선택합니다. 역할과 그 권한이 비활성화되면서 IAM 사용자 및 그룹에 연결된 권한이 자동으로 복구됩니다.

예를 들어, 사용자 이름 123456789012를 사용하여 계정 번호 RichardRoe로 로그인했다고 가정하십시오. AdminRole 역할을 사용한 후, 사용자가 역할 사용을 중지하고 원래 사용자 권한으로 돌아가고자 합니다. 역할 사용을 중지하려면 AdminRole @ 123456789012을 선택한 후 Back to RichardRoe(RichardRoe로 돌아가기)를 선택합니다.



IAM 역할로 전환하기(AWS CLI)

역할은 필요한 AWS 리소스에 액세스하는 데 사용할 수 있는 일련의 권한을 지정합니다. 이러한 의미에서 [AWS Identity and Access Management\(IAM\)](#) 사용자와 비슷하다고 할 수 있습니다. 사용자로 로그인할 때는 특정 권한이 부여됩니다. 하지만 역할로 로그인하지는 못하기 때문에 일단 사용자로 로그인한 후에 역할로 전환할 수 있습니다. 이 경우 초기의 사용자 권한은 잠시 무효화되고 역할에게 할당된 권한이 부여됩니다. 역할은 자신의 계정 또는 그 밖의 다른 AWS 계정에 속한 것일 수 있습니다. 역할과 역할의 이점, 역할 생성 및 구성 방법에 대한 자세한 내용은 [IAM역할 \(p. 153\)](#) 및 [IAM 역할 생성 \(p. 202\)](#) 단원을 참조하십시오. 역할을 수임하는 데 사용할 수 있는 여러 방법을 알아보려면 [IAM 역할 사용 \(p. 227\)](#) 단원을 참조하십시오.

Important

IAM 사용자의 권한과 수임한 역할의 권한은 누적되지 않습니다. 한 번에 오직 하나의 권한 집합만이 활성화됩니다. 어떤 역할을 수임할 때 이전 사용자 또는 역할 권한은 일시적으로 포기하고 해당 역할에 할당된 권한을 가지고 작업합니다. 역할을 끝내면 사용자 권한이 자동으로 회복됩니다.

IAM 사용자로 로그인한 경우 역할을 사용하여 AWS CLI 명령을 실행할 수 있습니다. 이미 사용 중인 [외부 인증 사용자 \(p. 161\)](#)([SAML \(p. 167\)](#) 또는 [OIDC \(p. 162\)](#))로 로그인하는 경우에도 역할을 사용해 AWS CLI명령을 실행할 수 있습니다. 또한 역할을 사용해 인스턴스 프로파일 전체에서 명령에 연결된 Amazon EC2 인스턴스 내에서 AWS CLI 명령을 실행할 수 있습니다. 또한 역할을 사용해 두 번째 역할을 수임하는 [역할 함께 둑기 \(p. 154\)](#)를 사용할 수도 있습니다. AWS 계정 루트 사용자로 로그인되어 있을 때는 역할을 수임할 수 없습니다.

기본적으로 역할 세션은 한 시간 동안 지속됩니다. `assume-role*` CLI 작업을 사용하여 역할을 수임하는 경우 `duration-seconds` 파라미터에 대한 값을 지정할 수 있습니다. 이 값의 범위는 900초(15분)에서 해당 역할에 대한 최대 세션 기간 설정까지일 수 있습니다. 역할에 대한 최댓값을 확인하는 방법을 알아보려면 [역할에 대한 최대 세션 기간 설정 보기 \(p. 228\)](#) 단원을 참조하십시오.

역할 함께 둑기를 사용하는 경우 세션 기간은 최대 1시간으로 제한됩니다. 그런 다음 `duration-seconds` 파라미터를 사용하여 1시간보다 큰 값을 입력하면 이 작업에 실패합니다.

개발 환경에서 IAM 사용자를 하나 갖고 있는데 이때 AWS CLI 명령줄에서 프로덕션 환경으로 작업해야 할 때가 있다고 가정합시다. 사용할 수 있는 액세스 키 자격 증명 세트가 이미 하나 있습니다. 이 세트는 표준

IAM 사용자에게 할당된 액세스 키 페어일 수도 있고, 연동 사용자로 로그인한 경우에는 초기에 할당된 역할에 대한 액세스 키 페어일 수도 있습니다. 현재 권한에 의해 특정 역할을 수임할 수 있는 능력이 부여되는 경우, AWS CLI 구성 파일의 "profile"에서 해당 역할을 식별할 수 있습니다. 그런 다음 해당 명령은 원래 자격 증명이 아닌 지정된 역할의 권한으로 실행됩니다. AWS CLI 명령에서 해당 프로파일을 지정하는 경우에는 새 역할을 사용하게 됩니다. 이 경우 개발 계정의 원래 권한을 동시에 사용할 수 없습니다. 한 번에 한 가지 권한 세트만 적용될 수 있기 때문입니다.

Note

계정에서 보안을 목적으로 AWS CloudTrail을 사용해 계정의 역할 사용을 감사할 수 있습니다. CloudTrail 로그에서 역할의 작업을 식별하기 위해 역할 세션 이름을 사용할 수 있습니다. 이 항목에 서 설명하는 것처럼 AWS CLI에서 사용자를 대신해 역할을 수임하면 역할 세션 이름이 AWS-CLI-session-*NNNNNNNN*으로 자동으로 생성됩니다. 여기서 *NNNNNNNN*은 Unix epoch time(1970년 1월 1일 자정 UTC 이후 경과된 초 수)으로 시간을 나타낸 정수입니다. 자세한 내용은 AWS CloudTrail User Guide의 [CloudTrail 이벤트 참조](#)를 참조하십시오.

역할을 전환하려면(AWS CLI)

1. AWS CLI를 사용한 적이 없을 경우 먼저 기본 CLI 프로필을 구성해야 합니다. 명령 프롬프트를 열고 IAM 사용자 또는 연동 역할에서 액세스 키를 사용하도록 AWS CLI 설치를 설정하십시오. 자세한 내용은 AWS Command Line Interface 사용 설명서의 [AWS Command Line Interface 구성](#)을 참조하십시오.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-east-2
Default output format [None]: json
```

2. .aws/config 파일에서 역할에 대한 새 프로파일을 만듭니다. 다음 예에서는 123456789012 계정의 **ProductionAccessRole** 역할로 전환하는 "prodaccess"라는 프로필을 만듭니다. 해당 역할을 만든 계정 관리자에게서 역할 ARN을 받습니다. 이 프로필이 호출되면 AWS CLI에서는 source_profile의 자격 증명을 사용하여 해당 역할의 자격 증명을 요청합니다. 이로 인해 source_profile로 참조되는 자격 증명에는 sts:AssumeRole에 지정된 역할에 대한 role_arn 권한이 있어야 합니다.

```
[profile prodaccess]
role_arn = arn:aws:iam::123456789012:role/ProductionAccessRole
source_profile = default
```

3. 새 프로필을 만든 후 --profile prodaccess 파라미터를 지정하는 AWS CLI 명령은 기본 사용자 대신 IAM 역할 ProductionAccessRole에 연결된 권한에 따라 실행됩니다.

```
$ aws iam list-users --profile prodaccess
```

이 명령은 ProductionAccessRole에 할당된 권한이 현재 AWS 계정에 사용자를 나열하는 것을 가능하게 하는 경우에 작동합니다.

4. 원래 자격 증명에 의해 부여된 권한으로 돌아가려면 명령을 --profile 파라미터 없이 실행합니다. AWS CLI에서 다시 기본 프로필의 자격 증명([Step 1](#)에서 구성)이 사용됩니다.

[역할 위임하기](#)에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서 단원을 참조하십시오.

IAM 역할로 전환하기(Windows PowerShell용 도구)

역할은 필요한 AWS 리소스에 액세스하는 데 사용할 수 있는 일련의 권한을 지정합니다. 이러한 의미에서 [AWS Identity and Access Management](#)(IAM) 사용자와 비슷하다고 할 수 있습니다. 사용자로 로그인할 때는 특정 권한이 부여됩니다. 하지만 역할로 로그인하지는 못하기 때문에 일단 로그인한 후에 역할로 전환할 수 있습니다. 이 경우 초기의 사용자 권한은 잠시 무효화되고 역할에게 할당된 권한이 부여됩니다. 역할은 자신

의 계정 또는 그 밖의 다른 AWS 계정에 속한 것일 수 있습니다. 역할과 역할의 이점, 역할 생성 및 구성 방법에 대한 자세한 내용은 [IAM역할 \(p. 153\)](#) 및 [IAM 역할 생성 \(p. 202\)](#) 단원을 참조하십시오.

Important

IAM 사용자의 권한과 전환 대상인 역할의 권한은 누적되지 않습니다. 한 번에 오직 하나의 권한 집합만이 활성화됩니다. 어떤 역할로 전환할 때 사용자 권한은 일시적으로 포기하고 역할에 할당된 권한을 가지고 작업합니다. 역할을 끝내면 사용자 권한이 자동으로 회복됩니다.

이 섹션에서는 Windows PowerShell용 AWS 도구에서 명령줄로 작업할 때 역할을 전환하는 방법에 대해 기술합니다.

개발 환경에서 계정을 하나 갖고 있는데 이따금 [Windows PowerShell용 도구](#)를 사용하는 명령줄에서 프로덕션 환경으로 작업해야 할 때가 있다고 가정합시다. 사용할 수 있는 액세스 키 자격 증명 세트가 이미 하나 있습니다. 이 세트는 표준 IAM 사용자에게 할당된 액세스 키 페어일 수도 있고, 연동 사용자로 로그인한 경우에는 초기에 할당된 역할에 대한 액세스 키 페어일 수도 있습니다. 이 자격 증명을 사용해 새 역할의 ARN을 파라미터로 전달하는 `Use-STSSRole` cmdlet을 실행할 수 있습니다. 해당 명령은 요청된 역할에 대한 임시 보안 자격 증명을 반환합니다. 그런 다음 생산 중인 리소스에 액세스할 수 있는 해당 역할의 권한으로 후속 PowerShell 명령에서 이 자격 증명을 사용할 수 있습니다. 한 번에 한 가지 권한 세트만 적용될 수 있기 때문에 해당 역할을 사용하는 동안에는 개발 계정의 사용자 권한을 사용할 수 없습니다.

Note

계정에서 보안을 목적으로 AWS CloudTrail을 사용해 계정의 역할 사용을 감사할 수 있습니다. cmdlet `Use-STSSRole`에는 반드시 2~64자리의 문자, 숫자 및 `-RoleSessionName` 기호로 된 값과 함께 `=, .@-` 파라미터가 포함되어야 합니다. 역할 세션 이름은 임시 보안 자격 증명으로 수행되는 CloudTrail 로그 작업을 식별합니다. 자세한 내용은 AWS CloudTrail User Guide의 [CloudTrail 이벤트 참조](#)를 참조하십시오.

모든 액세스 키와 토큰은 예제일 뿐이며 표시된 대로 사용할 수 없습니다. 라이브 환경의 적절한 값으로 바꾸십시오.

역할을 전환하려면(Windows PowerShell용 도구)

- PowerShell 명령 프롬프트를 열고 현재 IAM 사용자 또는 연동 역할에서 액세스 키를 사용하도록 기본 프로필을 구성하십시오. 이전에 Windows PowerShell용 도구를 사용했다면 이미 그렇게 한 것이나 다른 없습니다. AWS 계정 루트 사용자가 아닌 IAM 사용자로 로그인한 경우에 한해 역할을 바꿀 수 있다는 것에 유의하십시오.

```
PS C:\> Set-AWSCredentials -AccessKey AKIAIOSFODNN7EXAMPLE -SecretKey wJalrXUtnFEMI/K7MDENG/bPxRfCYEXAMPLEKEY -StoreAs MyMainUserProfile
PS C:\> Initialize-AWSSession -ProfileName MyMainUserProfile -Region us-east-2
```

자세한 내용은 Windows PowerShell용 AWS 도구 사용 설명서의 [AWS 자격 증명 사용](#)을 참조하십시오.

- 새 역할에 대한 자격 증명을 가져오려면, 다음 명령을 실행해 123456789012 계정의 `RoleName` 역할로 전환합니다. 해당 역할을 만든 계정 관리자에게서 역할 ARN을 받습니다. 그 명령은 세션 이름도 제공할 것을 요구합니다. 세션 이름에 대해서는 어떤 텍스트도 선택 가능합니다. 다음 명령은 자격 증명을 요청한 다음, 반환된 결과 객체로부터 `Credentials` 속성 객체를 캡처해 `$creds` 변수에 저장합니다.

```
PS C:\> $creds = (Use-STSSRole -RoleArn "arn:aws:iam::123456789012:role/RoleName" -RoleSessionName "MyRoleSessionName").Credentials
```

`$creds`는 다음 절차에서 필요한 `AccessKeyId`, `SecretAccessKey` 및 `SessionToken` 요소를 포함하는 객체입니다. 다음 샘플 명령은 전형적인 값을 보여줍니다.

```
PS C:\> $creds.AccessKeyId
```

```
AKIAIOSFODNN7EXAMPLE

PS C:\> $creds.SecretAccessKey
wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

PS C:\> $creds.SessionToken
AQoDYXdzEGcaEXAMPLE2gsYULo+Im5ZEXAMPLEeYjs1M2FUIgIJx9tQqNMBEXAMPLECvSRyh0FW7jEXAMPLEW
+vE/7s1HRp
XviG7b+qYf4nD00EXAMPLEmj4wxS04L/uZEXAMPLECiHzFB5lTYLto9dyBgSDyEXAMPLE9/
g7QRUhZp4bqbEXAMPLENwGPY
Oj59pFA41NKCikVgkREXAMPLEjlzxQ7y52gekeVEXAMPLEDiB9ST3UuysgsKdEXAMPLE1TVastU1A0SKFEXAMPLEiywCC/
C
s8EXAMPLEpZgOs+6hz4AP4KEXAMPLERbASP+4eZScEXAMPLEsnf87eNhyDHq6ikBQ==

PS C:\> $creds.Expiration
Thursday, June 18, 2018 2:28:31 PM
```

- 후속 명령에 대해 이 자격 증명을 사용하려면 `-Credentials` 파라미터로 자격 증명을 포함시키십시오. 예를 들어 다음 명령은 그 역할에 `iam>ListRoles` 권한이 부여되고, 따라서 `Get-IAMRoles` cmdlet을 실행할 수 있는 경우에 한해 역할에서 얻은 자격 증명을 사용하고 작동됩니다.

```
PS C:\> get-iamroles -Credential $creds
```

- 원래 자격 증명으로 돌아가려면 `-Credentials $creds` 파라미터 사용을 중지하고 PowerShell이 기본 프로필에 저장된 자격 증명으로 복귀할 수 있도록 허용하기만 하면 됩니다.

IAM 역할(AWS API)로 전환하기

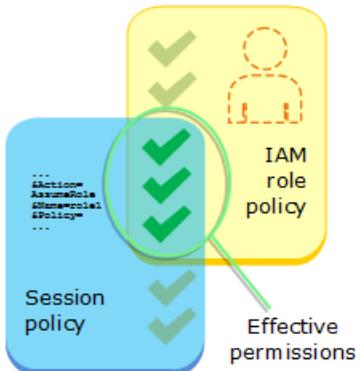
역할은 AWS 리소스에 액세스하는 데 사용할 수 있는 일련의 권한을 지정합니다. 이러한 면에서 [IAM 사용자와 비슷합니다](#). 보안 주체(개인 또는 애플리케이션)은 역할을 수임하여 필요한 작업을 수행하고 AWS 리소스와 상호작용할 수 있는 임시 권한을 부여받습니다. 역할은 자신의 계정 또는 그 밖의 다른 AWS 계정에 속한 것일 수 있습니다. 역할과 그 이점, 역할 생성 및 구성 방법에 대한 자세한 정보는 [IAM 역할 \(p. 153\)](#) 및 [IAM 역할 생성 \(p. 202\)](#) 단원을 참조하십시오. 역할을 수임하는 데 사용할 수 있는 여러 방법을 알아보려면 [IAM 역할 사용 \(p. 227\)](#) 단원을 참조하십시오.

Important

IAM 사용자의 권한과 수임한 역할의 권한은 누적되지 않습니다. 한 번에 오직 하나의 권한 집합만이 활성화됩니다. 어떤 역할을 수임할 때 이전 사용자 또는 역할 권한은 일시적으로 포기하고 해당 역할에 할당된 권한을 가지고 작업합니다. 이 역할을 끝내면 원래 권한이 자동으로 회복됩니다.

이때 역할 위임을 위해 애플리케이션은 AWS STS `AssumeRole` API 작업을 호출하고 사용할 역할의 ARN을 전달합니다. 이 작업은 임시 자격 증명으로 사용하여 새 세션을 생성합니다. 이 세션에는 해당 역할에 대한 자격 증명 기반 정책과 동일한 권한이 지정됩니다.

`AssumeRole`을 호출할 경우, 선택적으로 세션 정책을 전달할 수 있습니다. 세션 정책은 역할 또는 연합된 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 세션에는 역할의 자격 증명 기반 정책과 세션 정책, 이 두 정책 모두에 의해 부여되는 권한만 부여됩니다. 세션 정책은 다른 사람에게 임시 보안 자격 증명을 부여할 필요가 있을 때 유용합니다. 후속 AWS API 호출 시에도 역할의 임시 자격 증명을 사용하여 역할이 속한 계정의 리소스에 액세스할 수 있습니다. 세션 정책을 사용하여 수임된 역할의 자격 증명 기반 정책에서 허용되는 권한을 부여할 수는 없습니다. 이 역할의 효과적인 권한을 AWS가 어떻게 결정하는지 자세히 알아보려면 [정책 평가 로직 \(p. 531\)](#) 단원을 참조하십시오.



IAM 사용자 또는 역할을 이미 사용 중인 [외부 인증 사용자 \(p. 161\)](#)([SAML \(p. 167\)](#) 또는 [OIDC \(p. 162\)](#))로 로그인한 경우 `AssumeRole`을 호출할 수 있습니다. 또한 역할을 사용해 두 번째 역할을 수임하는 [역할 함께 끌기 \(p. 154\)](#)를 사용할 수도 있습니다. AWS 계정 루트 사용자로 로그인되어 있을 때는 역할을 수임할 수 없습니다.

기본적으로 역할 세션은 한 시간 동안 지속됩니다. AWS STS `AssumeRole*` API 작업을 사용하여 역할을 수임하는 경우 `DurationSeconds` 파라미터에 대한 값을 지정할 수 있습니다. 이 값의 범위는 900초(15분)에서 해당 역할에 대한 최대 세션 기간 설정까지일 수 있습니다. 역할에 대한 최댓값을 확인하는 방법을 알아보려면 [역할에 대한 최대 세션 기간 설정 보기 \(p. 228\)](#) 단원을 참조하십시오.

역할 함께 끌기를 사용하는 경우 세션은 최대 1시간으로 제한됩니다. 그런 다음 `DurationSeconds` 파라미터를 사용하여 1시간보다 큰 값을 입력하면 이 작업에 실패합니다.

Note

계정에서 보안을 목적으로 AWS CloudTrail을 사용해 계정의 역할 사용을 감사할 수 있습니다. `AssumeRole` 호출에는 반드시 2~64자의 문자, 숫자, 그리고 =, .@- 기호로 구성된 역할 세션 이름이 포함되어야 합니다. 역할 세션 이름은 임시 보안 자격 증명에 의해 수행되는 작업을 식별하기 위해 CloudTrail 로그에서 사용됩니다. 자세한 내용은 AWS CloudTrail User Guide의 [CloudTrail 이벤트 참조](#)를 참조하십시오.

AWS에 대한 Boto3 인터페이스([AWS SDK for Python \(Boto\) V3](#))를 사용하여 Python으로 작성된 다음 예제에서는 `AssumeRole`을 호출하는 방법을 보여줍니다. 또한 `AssumeRole`에서 반환한 임시 보안 자격 증명을 사용하여 해당 역할을 소유한 계정의 모든 Amazon S3 버킷을 나열하는 방법도 보여줍니다.

```
import boto3

# The calls to AWS STS AssumeRole must be signed with the access key ID
# and secret access key of an existing IAM user or by using existing temporary
# credentials such as those from another role. (You cannot call AssumeRole
# with the access key for the root account.) The credentials can be in
# environment variables or in a configuration file and will be discovered
# automatically by the boto3.client() function. For more information, see the
# Python SDK documentation:
# http://boto3.readthedocs.io/en/latest/reference/services/sts.html#client

# Create an STS client object that represents a live connection to the
# STS service
sts_client = boto3.client('sts')

# Call the assume_role method of the STSConnection object and pass the role
# ARN and a role session name.
assumed_role_object=sts_client.assume_role(
    RoleArn="arn:aws:iam::account-of-role-to-assume:role/name-of-role",
    RoleSessionName="AssumeRoleSession1"
)
```

```
# From the response that contains the assumed role, get the temporary
# credentials that can be used to make subsequent API calls
credentials=assumed_role_object['Credentials']

# Use the temporary credentials that AssumeRole returns to make a
# connection to Amazon S3
s3_resource=boto3.resource(
    's3',
    aws_access_key_id=credentials['AccessKeyId'],
    aws_secret_access_key=credentials['SecretAccessKey'],
    aws_session_token=credentials['SessionToken'],
)

# Use the Amazon S3 resource object that is now configured with the
# credentials to access your S3 buckets.
for bucket in s3_resource.buckets.all():
    print(bucket.name)
```

IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여하기

EC2 인스턴스에서 실행되는 애플리케이션에는 AWS API 요청에 AWS 자격 증명이 포함되어 있어야 합니다. 개발자들로 하여금 AWS 자격 증명을 EC2 인스턴스 내부에 직접 저장하고 그 인스턴스의 애플리케이션이 그 자격 증명의 사용을 허용하도록 했을 수도 있습니다. 그러면 개발자는 자격 증명을 관리하고 각 인스턴스에 자격 증명을 안전하게 전달해야 하며, 자격 증명을 교체할 때가 되면 각 EC2 인스턴스를 업데이트해야 할 것입니다. 이처럼 여기에는 많은 작업이 요구됩니다.

이렇게 하는 대신 IAM 역할을 사용하여 EC2 인스턴스에서 실행되는 애플리케이션의 임시 자격 증명을 관리할 수 있고 또 그렇게 해야 합니다. 역할을 사용할 때 EC2 인스턴스에 장기 자격 증명(예: 사용자 이름, 암호 또는 액세스 키)을 배포하지 않아도 됩니다. 그 대신 역할은 애플리케이션에서 다른 AWS 리소스에 호출할 때 사용할 수 있는 임시 권한을 제공합니다. EC2 인스턴스를 시작할 때 IAM 역할을 지정해 인스턴스에 연결합니다. 그러면 이 인스턴스에서 실행되는 애플리케이션은 역할 제공 임시 자격 증명을 사용하여 API 요청에 서명할 수 있습니다.

역할을 사용하여 EC2 인스턴스에서 실행되는 애플리케이션에 권한을 부여하기 위해서는 약간의 추가적인 구성이 필요합니다. EC2 인스턴스에서 실행되는 애플리케이션은 가상화된 운영 체제에 의해 AWS에서 추상화됩니다. 이러한 추가적인 분리로 인해 EC2 인스턴스에 AWS 역할 및 관련 권한을 할당하고 이를 그 애플리케이션도 사용 가능하게 만들려면 추가 절차가 필요합니다. 여기서 추가 절차란 인스턴스에 연결된 [인스턴스 프로파일](#)을 생성하는 것입니다. 그러면 인스턴스 프로필은 해당 역할을 포함하게 되며 인스턴스에서 실행되는 애플리케이션에 이 역할의 임시 자격 증명을 제공할 수 있습니다. 이 임시 자격 증명은 애플리케이션의 API 호출에 사용되어 리소스에 액세스하고 이 역할이 지정하는 리소스에 대해서만 액세스를 제한할 수 있습니다. 한 번에 하나의 역할만 EC2 인스턴스에 할당할 수 있으며, 인스턴스의 모든 애플리케이션은 동일한 역할과 권한을 공유한다는 것에 유의하십시오.

이러한 방식으로 역할을 사용하면 여러 가지 장점이 있습니다. 역할 자격 증명은 임시적이고 자동으로 교체되므로 자격 증명을 관리하지 않아도 될 뿐만 아니라 장기적인 보안 위험을 염려하지 않아도 됩니다. 또한, 여러 인스턴스에 대해 역할을 하나만 사용하는 경우 그 역할을 변경할 수 있는데, 변경 사항은 모든 인스턴스에 자동으로 전파됩니다.

Note

일반적으로 역할은 EC2 인스턴스를 시작할 때 할당되지만, 이미 실행 중인 EC2 인스턴스에도 연결될 수 있습니다. 실행 중인 인스턴스에 역할을 연결하는 방법을 알아보려면 [Amazon EC2의 IAM 역할 단원](#)을 참조하십시오.

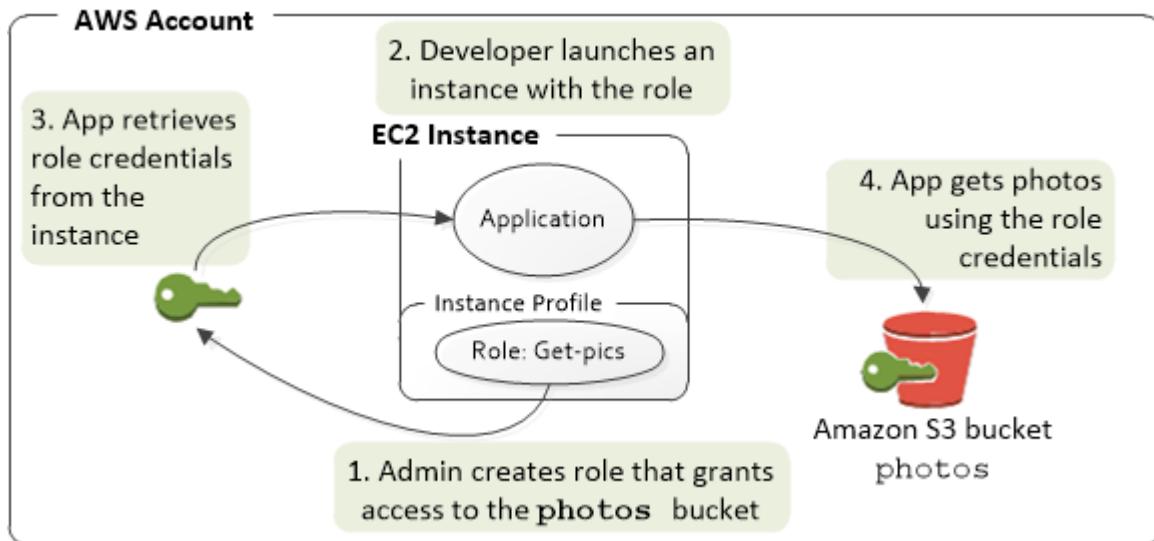
주제

- [EC2 인스턴스의 역할은 어떻게 작동하나요? \(p. 241\)](#)
- [Amazon EC2로 역할을 사용하는 데 필요한 권한 \(p. 242\)](#)

- 어떻게 시작할 수 있습니까? (p. 243)
- 관련 정보 (p. 243)
- 인스턴스 프로파일 사용 (p. 244)

EC2 인스턴스의 역할은 어떻게 작동하나요?

다음 그림에서는 개발자가 photos라는 S3 버킷에 대한 액세스 권한이 필요한 EC2 인스턴스에서 애플리케이션을 실행하고 있습니다. 관리자가 Get-pics 서비스 역할을 생성해 EC2 인스턴스에 연결합니다. 이 역할에는 지정된 S3 버킷에 대한 읽기 전용 액세스 권한을 부여하는 권한 정책이 포함되어 있습니다. 또한 EC2 인스턴스가 해당 역할을 수임하고 임시 자격 증명을 가져오도록 허용하는 신뢰 정책도 포함되어 있습니다. 애플리케이션이 인스턴스에서 실행되면 역할의 임시 자격 증명을 사용하여 photos 버킷에 액세스할 수 있습니다. 관리자는 개발자 권한을 부여하지 않아도 photos 버킷에 액세스할 수 있고 개발자는 자격 증명을 공유하거나 관리할 필요가 전혀 없습니다.



1. 관리자는 IAM을 사용하여 **Get-pics** 역할을 만듭니다. 역할의 신뢰 정책에서 관리자는 EC2 인스턴스만이 역할을 맡을 수 있도록 지정합니다. 역할의 권한 정책에서 관리자는 photos 버킷에 읽기 전용 권한을 지정합니다.
2. 개발자는 EC2 인스턴스를 시작하고 이 인스턴스에 Get-pics 역할을 할당합니다.

Note

IAM 콘솔을 사용하는 경우, 인스턴스 프로파일은 사용자를 위해 관리되고 대개 사용자가 파악하기 쉽습니다. 그러나 AWS CLI 또는 API를 사용하여 역할 및 EC2 인스턴스를 만들고 관리하는 경우 사용자는 인스턴스 프로파일을 만들고 별도 절차에 따라 여기에 역할을 할당해야 합니다. 그런 다음 인스턴스를 시작할 때 역할 이름이 아닌 인스턴스 프로파일 이름을 지정해야 합니다.

3. 애플리케이션이 실행되면 [인스턴스 메타데이터에서 보안 자격 증명 검색](#)에 설명된 대로 Amazon EC2 인스턴스 메타데이터에서 임시 보안 자격 증명을 가져옵니다. 이러한 자격 증명은 제한된 시간 동안에만 유효한 [임시 보안 자격 증명](#) (p. 263)으로 역할을 나타냅니다.

개발자는 일부 [AWS SDK](#)를 통해 임시 보안 자격 증명을 명료하게 관리하는 공급자를 사용할 수 있습니다. (개별 AWS SDK에 대한 설명서에 자격 증명을 관리하기 위해 SDK에서 지원하는 기능이 설명되어 있습니다.)

또는 애플리케이션이 EC2 인스턴스의 인스턴스 메타데이터에서 임시 자격 증명을 얻을 수 있습니다. 자격 증명과 관련 값은 메타데이터의 `iam/security-credentials/role-name` 범주(이 경우 `iam/security-credentials/Get-pics`)에서 구할 수 있습니다. 애플리케이션이 인스턴스 메타데이터에서 자격 증명을 가져오면 자격 증명을 캐시할 수 있습니다.

- 애플리케이션은 가져온 임시 자격 증명을 사용하여 photo 버킷에 액세스합니다. **Get-pics** 역할에 연결된 정책으로 인해 이 애플리케이션에는 읽기 전용 권한만 있습니다.

인스턴스에서 제공되는 임시 보안 자격 증명은 만료되기 전에 자동으로 교체되므로 항상 유효한 설정을 사용할 수 있습니다. 애플리케이션은 현재 자격 증명이 만료되기 전에 인스턴스 메타데이터에서 새 자격 증명을 가져와야 합니다. AWS SDK에서 자격 증명을 관리하는 경우 애플리케이션은 자격 증명을 갱신하기 위해 로직을 추가로 포함하지 않아도 됩니다. 그러나 애플리케이션이 인스턴스 메타데이터에서 임시 보안 자격 증명을 가져와 캐시한 경우, 현재 자격 증명이 만료되기 전에 한 시간 또는 최소 15분마다 갱신한 자격 증명을 가져와야 합니다. 만료 시간은 `iam/security-credentials/role-name` 카테고리에 반환되는 정보에 포함되어 있습니다.

Amazon EC2로 역할을 사용하는 데 필요한 권한

역할을 사용하여 인스턴스를 시작하려면 개발자에게 EC2 인스턴스를 시작할 수 있는 권한과 IAM 역할을 전달할 수 있는 권한이 있어야 합니다.

다음과 같은 샘플 정책은 사용자가 AWS Management 콘솔을 사용하여 역할로 인스턴스를 시작할 수 있도록 허용합니다. 이 정책에는 와일드카드(*)가 포함되어 있어 사용자가 어떤 역할이든 전달하고 어떤 Amazon EC2 작업도 수행할 수 있도록 허용합니다. `ListInstanceProfiles` 작업을 수행하면 사용자는 AWS 계정에서 제공되는 모든 역할을 볼 수 있습니다.

Example 사용자에게 Amazon EC2 콘솔을 사용하여 임의의 역할로 인스턴스를 시작할 권한을 부여하는 정책

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": [  
            "iam:PassRole",  
            "iam>ListInstanceProfiles",  
            "ec2:*"  
        ],  
        "Resource": "*"  
    }]  
}
```

(PassRole을 사용하여) EC2 인스턴스로 전달할 수 있는 역할 제한

`PassRole` 권한을 사용하여 사용자가 EC2 인스턴스를 시작할 때 이 인스턴스에 전달할 수 있는 역할을 제한할 수 있습니다. 이를 통해 사용자가 자신이 받은 권한 보다 더 많은 권한이 있는 애플리케이션을 실행하지 않도록, 즉 높은 권한을 가져오지 않도록 할 수 있습니다. 예를 들어 사용자 Alice는 EC2 인스턴스를 시작하고 Amazon S3 버킷을 사용할 권한만 갖고 있지만, 그녀가 인스턴스에 전달하는 역할에는 IAM 및 Amazon DynamoDB를 사용할 권한이 있다고 가정해 봅시다. 이 경우 Alice는 인스턴스를 시작하고 여기에 로그인하여 임시 보안 자격 증명을 가져온 다음 그녀에게 권한이 없는 IAM 또는 DynamoDB 작업을 수행할 수도 있습니다.

사용자가 EC2 인스턴스에 전달할 수 있는 역할 중 어떤 것을 제한하려면 `PassRole` 작업을 허용하는 정책을 생성합니다. 그런 다음 그 정책을 EC2 인스턴스를 시작할 사용자(또는 사용자가 소속된 IAM 그룹)에게 연결합니다. 이 정책의 `Resource` 요소에서 사용자가 EC2 인스턴스에 전달할 수 있는 역할을 나열합니다. 사용자가 인스턴스를 시작하고 역할을 인스턴스에 연결하면 Amazon EC2에서 사용자가 해당 역할을 전달할 수 있는지 확인합니다. 물론 사용자가 전달할 수 있는 역할에 사용자가 보유하고 있을 것으로 추정되는 권한 보다 더 많은 권한이 포함되어 있지 않은지도 확인해야 합니다.

Note

`PassRole`은 `RunInstances` 또는 `ListInstanceProfiles`와 동일한 방식의 API 작업이 아닙니다. 역할 ARN이 API에 대한 파라미터로 전달될 때마다 AWS에서 검사하는 권한입니다(또는 사

용자 대신 콘솔이 이 기능을 수행). 관리자가 어느 사용자가 어느 역할을 전달할 수 있는지를 제어할 수 있습니다. 이 경우 사용자가 Amazon EC2 인스턴스에 특정 역할을 연결할 수 있습니다.

Example 사용자에게 특정 역할로 EC2 인스턴스를 시작할 권한을 부여하는 정책

다음과 같은 샘플 정책은 사용자가 Amazon EC2 API를 사용하여 역할로 인스턴스를 시작할 수 있도록 허용합니다. Resource 요소는 역할의 Amazon 리소스 이름(ARN)을 지정합니다. ARN을 지정함으로써 정책은 사용자에게 Get-pics 역할만을 전달할 권한을 부여합니다. 사용자가 인스턴스를 시작할 때 다른 역할을 지정하려는 경우 작업이 실패합니다. 사용자는 역할을 전달하는지 여부에 관계없이 모든 인스턴스를 실행할 권리 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "iam:PassRole",  
            "Resource": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:role/Get-pics"  
        }  
    ]  
}
```

어떻게 시작할 수 있습니까?

역할이 EC2 인스턴스를 사용하는 방식을 이해하려면 IAM 콘솔을 사용하여 역할을 만들고 해당 역할을 사용하는 EC2 인스턴스를 시작한 다음, 실행 중인 인스턴스를 검사해야 합니다. 해당 역할의 임시 자격 증명이 인스턴스에서 사용되는 방식을 보기 위해 [인스턴스 메타데이터](#)를 검토할 수 있습니다. 또한, 인스턴스에서 실행되는 애플리케이션이 어떻게 역할을 사용하는지도 알 수 있습니다. 다음 리소스에서 자세히 알아보십시오.

- SDK 설명입니다. AWS SDK 설명서에는 역할에 대한 임시 자격 증명을 사용하여 Amazon S3 버킷을 읽는 EC2 인스턴스에서 실행되는 애플리케이션에 대한 자세한 안내가 있습니다. 다음과 같은 각 설명에서는 여러 프로그래밍 언어를 사용하여 비슷한 절차를 제시합니다.
 - AWS SDK for Java Developer Guide의 [Java용 SDK로 EC2 인스턴스에서 IAM 역할 사용](#)
 - .NET용 AWS SDK Developer Guide의 [.NET용 SDK로 EC2 인스턴스에서 IAM 역할 사용](#)
 - Ruby용 AWS SDK Developer Guide의 [Ruby용 SDK로 EC2 인스턴스에서 IAM 역할 사용](#)

위 설명에서는 예제 프로그램 생성 및 컴파일링, 역할 생성, 인스턴스 시작 및 연결, 예제 프로그램 배포 및 테스트에 대한 완벽한 단계별 지침을 제공합니다.

관련 정보

역할 생성 및 EC2 인스턴스의 역할에 대한 자세한 내용은 다음 정보를 참조하십시오.

- [Amazon EC2 인스턴스로 IAM 역할을 사용](#)하는 방법에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서 단원을 참조하십시오.
- 역할을 만들려면 [IAM 역할 생성 \(p. 202\)](#) 단원을 참조하십시오.
- 임시 보안 자격 증명의 사용에 관한 자세한 내용은 [임시 보안 자격 증명 \(p. 263\)](#)을 확인하십시오.
- IAM API 또는 CLI를 사용하는 경우, IAM 인스턴스 프로필을 생성 및 관리해야 합니다. 인스턴스 프로파일에 대한 자세한 내용은 [인스턴스 프로파일 사용 \(p. 244\)](#) 단원을 참조하십시오.

- 인스턴스 메타데이터의 역할에 대한 임시 보안 자격 증명에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 메타데이터에서 보안 자격 증명 검색](#)을 참조하십시오.

인스턴스 프로파일 사용

인스턴스 프로파일은 IAM 역할을 위한 컨테이너로서 인스턴스 시작 시 EC2 인스턴스에 역할 정보를 전달하는 데 사용됩니다.

인스턴스 프로파일 관리(콘솔)

AWS Management 콘솔을 사용하여 Amazon EC2 역할을 생성하는 경우, 콘솔이 자동으로 인스턴스 프로파일을 생성하여 해당 역할과 동일한 이름을 부여합니다. 그런 다음 Amazon EC2 콘솔을 사용해 IAM 역할과 연동하여 인스턴스를 실행할 때는 인스턴스와 연동할 역할을 선택할 수 있습니다. 콘솔에 표시되는 목록이 실제로 인스턴스 프로파일 이름의 목록입니다. 콘솔은 Amazon EC2와 연결되지 않은 역할에 대한 인스턴스 프로파일은 생성하지 않습니다.

인스턴스 프로파일(AWS CLI 또는 AWS API) 관리

AWS CLI 또는 AWS API에서 역할을 관리할 경우 별도의 작업으로 역할 및 인스턴스 프로파일을 생성합니다. 역할 및 인스턴스 프로파일의 이름이 서로 다를 수 있으므로 인스턴스 프로파일 이름은 물론이고 프로파일이 속하는 역할 이름까지 알고 있어야 합니다. 그러면 EC2 인스턴스를 시작할 때 올바른 인스턴스 프로파일을 선택할 수 있습니다.

Note

하나의 인스턴스 프로파일은 하나의 IAM 역할만 포함할 수 있습니다. 하지만 한 역할이 여러 인스턴스 프로파일에 포함될 수 있습니다. 이 인스턴스 프로파일당 역할 1개 제한은 늘릴 수 없습니다. 기존 역할을 제거하고 나서 인스턴스 프로파일에 다른 역할을 추가할 수 있습니다. [최종 일관성](#)으로 인해 모든 AWS에 변경 사항이 적용될 때까지 기다려야 합니다. 변경을 적용하려면 [인스턴스 프로파일 연결을 해제](#)하고 나서 [인스턴스 프로파일을 연결하거나](#), 인스턴스를 중지했다가 다시 시작합니다.

인스턴스 프로파일 관리(AWS CLI)

AWS 계정의 인스턴스 프로파일 작업을 할 때는 다음 AWS CLI 명령을 사용할 수 있습니다.

- 인스턴스 프로파일을 생성합니다: `aws iam create-instance-profile`
- 인스턴스 프로파일에 역할 추가: `aws iam add-role-to-instance-profile`
- 인스턴스 프로파일 표시: `aws iam list-instance-profiles`, `aws iam list-instance-profiles-for-role`
- 인스턴스 프로파일 정보 가져오기: `aws iam get-instance-profile`
- 인스턴스 프로파일에서 역할 제거: `aws iam remove-role-from-instance-profile`
- 인스턴스 프로파일 삭제: `aws iam delete-instance-profile`

다음 명령을 사용하여 이미 실행 중인 EC2 인스턴스에 역할을 연결할 수도 있습니다. 자세한 내용은 [Amazon EC2의 IAM 역할](#)을 참조하십시오.

- 역할이 있는 인스턴스 프로파일을 중지되었거나 실행 중인 EC2 인스턴스에 연결: `aws ec2 associate-iam-instance-profile`
- EC2 인스턴스에 연결된 인스턴스 프로파일에 대한 정보 가져오기: `aws ec2 describe-iam-instance-profile-associations`
- 역할이 있는 인스턴스 프로파일을 중지되었거나 실행 중인 EC2 인스턴스에서 분리: `aws ec2 disassociate-iam-instance-profile`

인스턴스 프로파일 관리(AWS API)

AWS 계정의 인스턴스 프로파일 작업을 할 때는 다음 AWS API 연산을 호출할 수 있습니다.

- 인스턴스 프로파일을 생성합니다: [CreateInstanceProfile](#)
- 인스턴스 프로파일에 역할 추가: [AddRoleToInstanceProfile](#)
- 인스턴스 프로파일 표시: [ListInstanceProfiles](#), [ListInstanceProfilesForRole](#)
- 인스턴스 프로파일 정보 가져오기: [GetInstanceProfile](#)
- 인스턴스 프로파일에서 역할 제거: [RemoveRoleFromInstanceProfile](#)
- 인스턴스 프로파일 삭제: [DeleteInstanceProfile](#)

다음 연산을 호출하여 이미 실행 중인 EC2 인스턴스에 역할을 연결할 수도 있습니다. 자세한 내용은 [Amazon EC2의 IAM 역할](#)을 참조하십시오.

- 역할이 있는 인스턴스 프로파일을 중지되었거나 실행 중인 EC2 인스턴스에 연결: [AssociateIamInstanceProfile](#)
- EC2 인스턴스에 연결된 인스턴스 프로파일에 대한 정보 가져오기: [DescribeIamInstanceProfileAssociations](#)
- 역할이 있는 인스턴스 프로파일을 중지되었거나 실행 중인 EC2 인스턴스에서 분리: [DisassociateIamInstanceProfile](#)

IAM 역할의 임시 보안 자격 증명 취소

Warning

이 페이지의 단계대로 수행하면, 역할을 수임하여 만들어진 현재 세션의 모든 사용자는 모든 AWS 작업 및 리소스에 액세스할 수 없게 됩니다. 이로 인해 사용자가 저장하지 않은 작업이 사라질 수 있습니다.

세션 지속 시간을 길게 하여(예: 12시간) 사용자가 AWS Management 콘솔에 액세스할 수 있도록 하면 사용자의 임시 자격 증명이 금방 만료되지 않습니다. 사용자가 허가 받지 않은 타사에게 실수로 자격 증명을 노출한 경우, 해당 타사는 세션의 지속 기간 동안 액세스 권한을 가지게 됩니다. 그러나 필요하다면 특정 시점 이전에 발행된 역할의 자격 증명에 대한 모든 권한을 즉시 취소할 수 있습니다. 그러면 지정한 시점 이전에 발행된 해당 역할의 임시 자격 증명은 모두 무효가 됩니다. 이에 따라 모든 사용자는 다시 인증을 받고 새 자격 증명을 요청해야 합니다.

Note

[서비스 연결 역할](#) (p. 154)에 대한 세션은 취소할 수 없습니다.

이 주제의 절차에 따라 역할의 권한을 취소하면 AWS는 모든 작업에 대한 모든 권한을 거부하는 새 인라인 정책을 만들어 해당 역할에 연결합니다. 여기에는 권한을 취소한 시점 이전에 역할을 위임한 사용자에게만 제한을 가하는 조건이 포함됩니다. 권한을 취소한 이후에 역할을 위임한 사용자에게는 거부 정책이 적용되지 않습니다.

Important

이 거부 정책은 콘솔 세션의 지속 기간이 긴 사용자만이 아니라 지정된 역할의 모든 사용자에게 적용됩니다.

역할의 세션 권한을 취소하기 위한 최소 권한

역할의 세션 권한을 취소하려면 해당 역할에 대한 [PutRolePolicy](#) 권한이 있어야 합니다. 이렇게 하면 해당 역할에 [AWSRevokeOlderSessions](#) 인라인 정책을 연결할 수 있게 됩니다.

세션 권한 취소

역할에서 세션 권한을 취소할 수 있습니다.

역할 자격 증명의 현재 사용자에 대해 모든 권한을 즉시 거부하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. IAM Dashboard(IAM 대시보드)의 탐색 창에서 역할을 선택한 다음, 권한을 취소할 역할의 이름(확인란 아님)을 선택합니다.
3. 선택한 역할의 요약 페이지에서 Revoke sessions(세션 취소) 탭을 선택합니다.
4. Revoke sessions(세션 취소) 탭에서 Revoke active sessions(활성 세션 취소)를 선택합니다.
5. AWS에 작업 확인 메시지가 나타납니다. 대화 상자에서 Revoke active sessions(활성 세션 취소)를 선택합니다.

IAM은 AWSRevokeOlderSessions라는 정책을 즉시 해당 역할에 연결합니다. 이 정책은 Revoke active sessions(활성 세션 취소)를 선택한 순간 이전에 해당 역할을 수임한 사용자의 모든 액세스 권한을 거부합니다. Revoke active sessions(활성 세션 취소)를 선택한 이후에 역할을 수임한 사용자에게는 적용되지 않습니다.

사용자나 리소스에 새 정책을 적용할 때 정책 업데이트가 효력이 생기는 데 몇 분이 걸릴 수 있습니다.

Note

정책 삭제에 대해서는 걱정하지 마십시오. 세션을 취소한 이후에 역할을 수임한 사용자에게는 이 정책이 적용되지 않습니다. 나중에 Revoke Sessions(세션 취소)를 다시 선택하는 경우, 정책의 날짜/시간 스템프가 새로 고쳐지면서 새로 지정된 시간 이전에 역할을 수임한 모든 사용자의 모든 권한을 거부하게 됩니다.

이러한 식으로 세션이 취소된 유효한 사용자는 작업을 계속하려면 새 세션을 위한 임시 자격 증명을 가져와야 합니다. AWS CLI는 자격 증명이 만료될 때까지 이를 캐시합니다. CLI가 더 이상 유효하지 않은 캐시된 자격 증명을 강제로 삭제하고 새로 고치게 하려면 다음 명령 중 하나를 실행합니다.

Linux, macOS 또는 Unix

```
$ rm -r ~/.aws/cli/cache
```

Windows

```
C:\> del /s /q %UserProfile%\.aws\cli\cache
```

자세한 내용은 [임시 보안 자격 증명에 대한 권한 비활성화 \(p. 283\)](#) 단원을 참조하십시오.

IAM 역할 관리

때때로 생성한 역할을 수정 또는 삭제해야 할 때가 있습니다. 역할을 변경하려면 다음 중 하나를 수행할 수 있습니다.

- 역할과 연결된 정책을 수정합니다.
- 역할에 액세스할 수 있는 사람을 변경합니다.
- 사용자에게 역할을 부여하는 권한을 편집합니다.
- AWS CLI 또는 API를 사용하여 수임한 역할에 대한 최대 세션 기간 설정을 변경합니다.

또한 더 이상 필요 없는 역할을 삭제할 수 있습니다. AWS Management 콘솔, AWS CLI 및 API에서 역할을 관리할 수 있습니다.

주제

- [역할 변경 \(p. 247\)](#)
- [역할 또는 인스턴스 프로파일 삭제 \(p. 255\)](#)

역할 변경

다음과 같은 방법으로 IAM의 역할을 변경하거나 수정할 수 있습니다.

- 역할을 맡을 수 있는 주체를 바꾸려면 역할의 신뢰 정책을 변경해야 합니다. [서비스 연결 역할 \(p. 154\)](#)에 대한 신뢰 정책을 수정할 수 없습니다.

Note

사용자가 역할 신뢰 정책에 보안 주체로 나열되지만 역할을 수임할 수 없는 경우 사용자의 [권한 경계 \(p. 317\)](#)를 확인하십시오. 사용자에 대한 권한 경계가 설정된 경우 권한 경계에서 `sts:AssumeRole` 작업이 허용되어야 합니다.

- 역할이 허용하는 권한을 변경하려면, 역할의 권한 정책을 수정합니다. IAM의 [서비스 연결 역할 \(p. 154\)](#)에 대한 권한 정책을 수정할 수 없습니다. 역할에 따른 서비스 내 권한 정책을 수정할 수 있는 가능성성이 있습니다. 서비스에서 이 기능을 지원하는지 여부를 알아보려면 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#)를 참조하고 서비스 연결 역할 열에 예가 있는 서비스를 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 [Yes] 링크를 선택합니다.
- 역할의 설명을 변경하려면 설명 텍스트를 수정합니다.
- 역할의 태그 세트를 변경하려면 [IAM 엔터티에 대한 태그 관리\(콘솔\) \(p. 262\)](#) 단원을 참조하십시오.
- AWS CLI 또는 API를 사용하여 수임한 역할에 대한 최대 세션 기간 설정을 지정하려면 최대 세션 기간 설정의 값을 수정합니다. 이 설정에는 1~12시간의 값을 지정할 수 있습니다. 값을 지정하지 않으면 기본 최댓값인 1시간이 적용됩니다.

Note

AWS CLI 또는 API에서 역할을 수임한 사람은 누구나 `duration-seconds` CLI 파라미터 또는 `DurationSeconds` API 파라미터를 사용해 더 긴 세션을 요청할 수 있습니다. `MaxSessionDuration` 설정은 `DurationSeconds` 파라미터를 사용해 요청할 수 있는 역할 세션에 대한 최대 기간을 결정합니다. 사용자가 `DurationSeconds` 파라미터의 값을 지정하지 않으면 보안 자격 증명이 한 시간 동안 유효하게 됩니다.

- 역할이 허용하는 최대 권한을 변경하려면, 역할의 [권한 경계 \(p. 317\)](#)를 수정합니다.

이와 같은 변경은 AWS Management 콘솔, [AWS 명령줄 도구](#), Windows PowerShell용 도구 또는 IAM API를 사용하여 할 수 있습니다.

주제

- [역할 액세스 보기 \(p. 247\)](#)
- [역할 수정\(콘솔\) \(p. 248\)](#)
- [역할 변경\(AWS CLI\) \(p. 250\)](#)
- [역할 변경\(AWS API\) \(p. 252\)](#)

역할 액세스 보기

역할에 대한 권한을 변경하기 전에 최근 서비스 수준 활동을 검토해야 합니다. 이 기능은 사용 중인 보안 주체(사람 또는 애플리케이션)의 액세스 권한을 제거하지 않으려는 경우 중요합니다. 서비스에서 마지막으로

액세스한 데이터 보기에 대한 자세한 정보는 [서비스에서 마지막으로 액세스한 데이터를 사용하여 권한 줄이기 \(p. 409\)](#) 단원을 참조하십시오.

역할 수정(콘솔)

AWS Management 콘솔을 사용하여 역할을 변경할 수 있습니다.

역할(콘솔) 위임자를 변경하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 [Roles]를 선택합니다.
3. 계정의 역할 목록에서 변경할 역할의 이름을 선택합니다.
4. 신뢰 관계 탭을 선택한 후 Edit trust relationship(신뢰 관계 편집)을 선택합니다.
5. 필요에 따라 신뢰 정책을 편집합니다. 역할을 위임할 수 있는 보안 주체를 추가하려면 Principal 요소에 해당 보안 주체를 지정하십시오. 예를 들어 다음 정책 조작은 Principal 요소에서 AWS 계정 2개를 참조하는 방법을 나타냅니다.

```
"Principal": {  
    "AWS": [  
        "arn:aws:iam::111122223333:root",  
        "arn:aws:iam::44445556666:root"  
    ]  
},
```

다른 계정에서 보안 주체를 지정할 경우 역할의 신뢰 정책에 계정을 추가하는 것은 신뢰 관계 설정의 절반밖에 되지 않는다는 것을 유념하십시오. 기본적으로 신뢰 계정의 어떠한 사용자도 역할을 위임할 수 없습니다. 새로이 신뢰받는 계정에 대한 관리자는 사용자가 역할을 수임할 수 있는 권한을 허용해야 합니다. 이를 위해 관리자는 사용자와 연결된 정책을 생성 또는 편집하여 sts:AssumeRole 작업에 대한 사용자 액세스를 허용합니다. 자세한 정보는 다음 절차 또는 [사용자에 대한 역할 전환 권한 부여 \(p. 229\)](#) 단원을 참조하십시오.

다음 정책 조작은 Principal 요소에서 두 가지 AWS 서비스를 참조하는 방법을 나타냅니다.

```
"Principal": {  
    "Service": [  
        "opsworks.amazonaws.com",  
        "ec2.amazonaws.com"  
    ]  
},
```

6. 신뢰 정책 편집을 마쳤으면 Update Trust Policy(신뢰 정책 업데이트)를 선택하여 변경 사항을 저장합니다.

정책 구조 및 구문에 대한 자세한 정보는 [정책 및 권한 \(p. 305\)](#) 단원과 [IAM JSON 정책 요소 참조 \(p. 498\)](#) 단원을 참조하십시오.

신뢰할 수 있는 외부 계정의 사용자가 역할을 사용할 수 있도록 허용하려면(콘솔 사용)

자세한 정보와 이 절차에 대한 세부 정보는 [사용자에 대한 역할 전환 권한 부여 \(p. 229\)](#) 단원을 참조하십시오.

1. 신뢰할 수 있는 외부 AWS 계정에 로그인합니다.
2. 사용자 또는 그룹 중 권한을 어디에 추가할지 결정합니다. 결정에 따라 IAM 콘솔의 탐색 창에서 사용자 또는 그룹을 선택합니다.
3. 액세스 권한을 부여하려는 사용자나 그룹의 이름을 선택한 후 권한 탭을 선택합니다.

4. 다음 중 하나를 수행하십시오.

- 고객 관리형 정책을 편집하려면 정책 이름을 선택하고 정책 편집을 선택한 다음 JSON 탭을 선택합니다. AWS 관리형 정책은 편집할 수 없습니다. AWS 관리형 정책은 AWS 아이콘()으로 나타납니다. AWS 관리형 정책과 고객 관리형 정책의 차이점에 대한 자세한 정보는 [관리형 정책과 인라인 정책 \(p. 312\)](#) 단원을 참조하십시오.
- 인라인 정책을 편집하려면 정책 이름 옆에 있는 화살표를 선택하고 정책 편집을 선택합니다.

5. 정책 편집기에서 새로운 Statement 요소를 추가하여 다음과 같이 지정합니다.

```
{  
    "Effect": "Allow",  
    "Action": "sts:AssumeRole",  
    "Resource": "arn:aws:iam::ACCOUNT-ID:role/ROLE-NAME"  
}
```

설명문의 ARN을 사용자가 수임할 수 있는 역할의 ARN으로 바꿉니다.

6. 화면의 메시지에 따라 정책 편집을 마칩니다.

역할이 허용하는 권한을 변경하려면(콘솔 사용)

1. <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 [Roles]를 선택합니다.
3. 수정하려는 역할의 이름을 선택한 후 권한 탭을 선택합니다.
4. 다음 중 하나를 수행하십시오.
 - 기존의 고객 관리형 정책을 편집하려면 정책 이름을 선택한 후 정책 편집을 선택합니다.

Note

AWS 관리형 정책은 편집할 수 없습니다. AWS 관리형 정책은 AWS 아이콘()으로 나타납니다. AWS 관리형 정책과 고객 관리형 정책의 차이점에 대한 자세한 정보는 [관리형 정책과 인라인 정책 \(p. 312\)](#) 단원을 참조하십시오.

- 기존의 관리형 정책을 역할에 연결하려면 Add permissions(권한 추가)를 선택합니다.
- 기존의 인라인 정책을 편집하려면 정책 이름 옆에 있는 화살표를 선택하고 정책 편집을 선택합니다.
- 새로운 인라인 정책을 포함시키려면 Add inline policy(인라인 정책 추가)를 선택합니다.

역할의 설명을 변경하려면(콘솔 사용)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 [Roles]를 선택합니다.
3. 변경할 역할 이름을 선택합니다.
4. Role description(역할 설명) 옆의 맨 오른쪽에서 편집을 선택합니다.
5. 상자에 새 설명을 입력하고 [Save]를 선택합니다.

AWS CLI 또는 API를 사용하여 수임한 역할에 대한 최대 세션 기간 설정을 변경하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 [Roles]를 선택합니다.

3. 변경할 역할 이름을 선택합니다.
4. Maximum CLI/API session duration(최대 CLI/API 세션 기간) 옆에서 값을 선택합니다. 또는 Custom duration(사용자 지정 기간)을 선택하고 값을(초)를 입력합니다.
5. Save를 선택합니다.

다음에 다른 사람이 이 역할을 수임할 때까지 변경 사항은 적용되지 않습니다. 이 역할에 대한 기존 세션을 취소하는 방법을 알아보려면 [IAM 역할의 임시 보안 자격 증명 취소 \(p. 245\)](#) 단원을 참조하십시오.

역할에 대한 권한 경계 설정에 사용된 정책을 변경하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택합니다.
3. [권한 경계 \(p. 317\)](#)를 변경하려는 역할의 이름을 선택합니다.
4. Permissions 탭을 선택합니다. 필요하다면 Permissions boundary(권한 경계) 섹션을 열고 Change boundary(경계 변경)을 선택합니다.
5. 정책을 선택하여 원하는 권한 경계를 사용하십시오.
6. Change boundary(경계 변경)을 선택합니다.

다음에 다른 사람이 이 역할을 수임할 때까지 변경 사항은 적용되지 않습니다.

역할 변경(AWS CLI)

AWS Command Line Interface를 사용하여 역할을 변경할 수 있습니다.

역할(AWS CLI) 위임자를 변경하려면

1. (선택 사항) 수정할 역할의 이름을 모르는 경우 다음 명령을 실행하여 계정의 역할을 나열합니다.
 - `aws iam list-roles`
2. (옵션) 현재 역할의 신뢰 정책을 확인하려면 다음 명령을 실행합니다.
 - `aws iam get-role`
3. 역할에 액세스할 수 있는 신뢰할 수 있는 보안 주체를 변경하려면 업데이트된 신뢰 정책을 추가하여 텍스트 파일을 생성합니다. 정책 구조를 작성할 때는 어떤 텍스트 편집기든 사용할 수 있습니다.

예를 들어 다음 신뢰 정책 조각은 Principal 요소에서 AWS 계정 2개를 참조하는 방법을 나타냅니다. 사용자가 개별 AWS 계정 2개를 사용하도록 허용하여 이 역할을 수임하도록 합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Principal": {"AWS": [  
             "arn:aws:iam::111122223333:root",  
             "arn:aws:iam::444455556666:root"  
         ]},  
         "Action": "sts:AssumeRole"  
     }  
}
```

다른 계정에서 보안 주체를 지정할 경우 역할의 신뢰 정책에 계정을 추가하는 것은 신뢰 관계 설정의 절반밖에 되지 않는다는 것을 유념하십시오. 기본적으로 신뢰 계정의 어떠한 사용자도 역할을 위임할 수 없습니다. 새로이 신뢰받는 계정에 대한 관리자는 사용자가 역할을 수임할 수 있는 권한을 허용해야 합니다. 이를 위해 관리자는 사용자와 연결된 정책을 생성 또는 편집하여 `sts:AssumeRole` 작업

에 대한 사용자 액세스를 허용합니다. 자세한 정보는 다음 절차 또는 [사용자에 대한 역할 전환 권한 부여 \(p. 229\)](#) 단원을 참조하십시오.

4. 방금 생성한 파일을 사용하여 신뢰 정책을 업데이트하려면 다음 명령을 실행합니다.

- [aws iam update-assume-role-policy](#)

신뢰할 수 있는 외부 계정 사용자에게 역할 사용을 허용하려면(AWS CLI)

자세한 정보와 이 절차에 대한 세부 정보는 [사용자에 대한 역할 전환 권한 부여 \(p. 229\)](#) 단원을 참조하십시오.

1. 역할에 대한 권한 정책을 포함하는 JSON 파일을 생성하여 역할을 수임할 수 있는 권한을 허용합니다. 예를 들어 다음 정책에는 필요한 최소 권한이 포함되어 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "sts:AssumeRole",  
            "Resource": "arn:aws:iam::ACCOUNT-ID-THAT-CONTAINS-ROLE:role/ROLE-NAME"  
        }  
    ]  
}
```

설명문의 ARN을 사용자가 수임할 수 있는 역할의 ARN으로 바꿉니다.

2. 다음 명령을 실행하여 신뢰 정책이 IAM에 포함하는 JSON 파일을 업로드합니다.

- [aws iam create-policy](#)

이 명령의 출력 화면에는 정책의 ARN이 포함됩니다. 이후 단계에서 사용해야 하므로 이 ARN을 기록해 두십시오.

3. 정책을 연결할 사용자 또는 그룹을 결정합니다. 원하는 사용자 또는 그룹의 이름을 모르는 경우에는 다음 명령 중 하나를 사용하여 계정에 속한 사용자 또는 그룹 목록을 조회합니다.

- [aws iam list-users](#)
- [aws iam list-groups](#)

4. 다음 명령 중 한 가지를 사용하여 이전 단계에서 생성한 정책을 사용자 또는 그룹에게 추가합니다.

- [aws iam attach-user-policy](#)
- [aws iam attach-group-policy](#)

역할에서 허용되는 권한을 변경하려면(AWS CLI)

1. (옵션) 현재 역할과 연동되어 있는 권한을 확인하려면 다음 명령을 실행합니다.

1. 인라인 정책을 나열하기 위한 [aws iam list-role-policies](#)
2. 관리형 정책을 나열하기 위한 [aws iam list-attached-role-policies](#)

2. 역할 권한의 업데이트 명령은 관리형 정책을 업데이트할 때와 인라인 정책을 업데이트할 때 서로 다르니다.

관리형 정책을 업데이트하려면 다음 명령을 실행하여 새로운 버전의 관리형 정책을 생성합니다.

- [aws iam create-policy-version](#)

인라인 정책을 업데이트하려면 다음 명령을 실행합니다.

- [aws iam put-role-policy](#)

역할(AWS CLI)에 대한 권한 경계 설정에 사용된 관리형 정책을 변경하려면

1. (선택 사항) 역할의 현재 [권한 경계 \(p. 317\)](#)를 확인하려면 다음 명령을 실행합니다.
 - [aws iam get-role](#)
2. 다른 관리형 정책을 사용하여 역할에 대한 권한 경계를 업데이트하려면 다음 명령 중 하나를 실행합니다.
 - [aws iam put-role-permissions-boundary](#)

역할은 권한 경계로서 하나의 관리형 정책만 가질 수 있습니다. 권한 경계를 변경하면 역할이 허용하는 최대 권한을 변경합니다.

역할의 설명을 변경하려면(AWS CLI)

1. (옵션) 역할의 현재 설명을 보려면 다음 명령을 실행합니다.
 - [aws iam get-role](#)
2. 역할의 설명을 업데이트하려면 설명 파라미터와 함께 다음 명령을 실행합니다.
 - [aws iam update-role](#)

AWS CLI를 사용하여 수입한 역할에 대한 최대 세션 기간 설정을 변경하려면(AWS CLI)

1. (옵션) 역할에 대한 현재 최대 세션 기간 설정을 확인하려면 다음 명령을 실행합니다.
 - [aws iam get-role](#)
2. 역할의 최대 세션 기간 설정을 업데이트하려면 `max-sessionduration` CLI 파라미터 또는 `MaxSessionDuration` API 파라미터와 함께 다음 명령을 실행합니다.
 - [aws iam update-role](#)

다음에 다른 사람이 이 역할을 수임할 때까지 변경 사항은 적용되지 않습니다. 이 역할에 대한 기존 세션을 취소하는 방법을 알아보려면 [IAM 역할의 임시 보안 자격 증명 취소 \(p. 245\)](#) 단원을 참조하십시오.

역할 변경(AWS API)

AWS API를 사용하여 역할을 변경할 수 있습니다.

역할(AWS API) 위임자를 변경하려면

1. (선택 사항) 변경할 역할의 이름을 모르는 경우 다음 연산을 호출하여 계정의 역할을 나열합니다.
 - [ListRoles](#)
2. (옵션) 현재 역할의 신뢰 정책을 확인하려면 다음 연산을 호출합니다.
 - [GetRole](#)
3. 역할에 액세스할 수 있는 신뢰할 수 있는 보안 주체를 변경하려면 업데이트된 신뢰 정책을 추가하여 텍스트 파일을 생성합니다. 정책 구조를 작성할 때는 어떤 텍스트 편집기든 사용할 수 있습니다.

예를 들어 다음 신뢰 정책 조각은 Principal 요소에서 AWS 계정 2개를 참조하는 방법을 나타냅니다. 사용자가 개별 AWS 계정 2개를 사용하도록 허용하여 이 역할을 수임하도록 합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Principal": {"AWS": [  
            "arn:aws:iam::111122223333:root",  
            "arn:aws:iam::444455556666:root"  
        ]},  
         "Action": "sts:AssumeRole"  
    ]  
}
```

다른 계정에서 보안 주체를 지정할 경우 역할의 신뢰 정책에 계정을 추가하는 것은 신뢰 관계 설정의 절반밖에 되지 않는다는 것을 유념하십시오. 기본적으로 신뢰 계정의 어떠한 사용자도 역할을 위임할 수 없습니다. 새로이 신뢰받는 계정에 대한 관리자는 사용자가 역할을 수임할 수 있는 권한을 허용해야 합니다. 이를 위해 관리자는 사용자와 연결된 정책을 생성 또는 편집하여 sts:AssumeRole 작업에 대한 사용자 액세스를 허용합니다. 자세한 정보는 다음 절차 또는 [사용자에 대한 역할 전환 권한 부여 \(p. 229\)](#) 단원을 참조하십시오.

- 방금 생성한 파일을 사용하여 신뢰 정책을 업데이트하려면 다음 작업을 호출합니다.
 - [UpdateAssumeRolePolicy](#)

신뢰할 수 있는 외부 계정 사용자에게 역할 사용을 허용하려면(AWS API)

자세한 정보와 이 절차에 대한 세부 정보는 [사용자에 대한 역할 전환 권한 부여 \(p. 229\)](#) 단원을 참조하십시오.

- 역할에 대한 권한 정책을 포함하는 JSON 파일을 생성하여 역할을 수임할 수 있는 권한을 허용합니다. 예를 들어 다음 정책에는 필요한 최소 권한이 포함되어 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": "sts:AssumeRole",  
         "Resource": "arn:aws:iam::ACCOUNT-ID-THAT-CONTAINS-ROLE:role/ROLE-NAME"  
    ]  
}
```

설명문의 ARN을 사용자가 수임할 수 있는 역할의 ARN으로 바꿉니다.

- 다음 작업을 호출하여 신뢰 정책이 IAM에 포함하는 JSON 파일을 업로드합니다.
 - [CreatePolicy](#)

이 연산의 출력 화면에는 정책의 ARN이 포함됩니다. 이후 단계에서 사용해야 하므로 이 ARN을 기록해 두십시오.

- 정책을 연결할 사용자 또는 그룹을 결정합니다. 원하는 사용자 또는 그룹의 이름을 모르는 경우에는 다음 작업 중 하나를 호출하여 계정에 속한 사용자 또는 그룹 목록을 조회합니다.
 - [ListUsers](#)
 - [ListGroups](#)

- 다음 연산 중 하나를 호출하여 이전 단계에서 생성한 정책을 사용자 또는 그룹에게 추가합니다.

- API: [AttachUserPolicy](#)
- [AttachGroupPolicy](#)

역할에서 허용되는 권한을 변경하려면(AWS API)

1. (옵션) 현재 역할과 연동되어 있는 권한을 확인하려면 다음 연산을 호출합니다.
 1. 인라인 정책을 나열하기 위한 [ListRolePolicies](#)
 2. 관리형 정책을 나열하기 위한 [ListAttachedRolePolicies](#)
2. 역할 권한의 업데이트 작업은 관리형 정책을 업데이트할 때와 인라인 정책을 업데이트할 때 서로 다른 다릅니다.

관리형 정책을 업데이트하려면 다음 연산을 호출하여 새로운 버전의 관리형 정책을 생성합니다.

- [CreatePolicyVersion](#)

인라인 정책을 업데이트하려면 다음 연산을 호출합니다.

- [PutRolePolicy](#)

역할(AWS API)에 대한 권한 경계 설정에 사용된 관리형 정책을 변경하려면

1. (선택 사항) 역할의 현재 [권한 경계 \(p. 317\)](#)를 확인하려면 다음 작업을 호출합니다.
 - [GetRole](#)
2. 다른 관리형 정책을 사용하여 역할에 대한 권한 경계를 업데이트하려면 다음 작업 중 하나를 호출합니다.
 - [PutRolePermissionsBoundary](#)

역할은 권한 경계로서 하나의 관리형 정책만 가질 수 있습니다. 권한 경계를 변경하면 역할이 허용하는 최대 권한을 변경합니다.

역할의 설명을 변경하려면(AWS API)

1. (옵션) 현재 역할의 설명을 확인하려면 다음 연산을 호출합니다.
 - [GetRole](#)
2. 역할의 설명을 업데이트하려면 설명 파라미터와 함께 다음 연산을 호출합니다.
 - [UpdateRole](#)

API를 사용하여 수입한 역할에 대한 최대 세션 기간 설정을 변경하려면(AWS API)

1. (옵션) 역할에 대한 현재 최대 세션 기간 설정을 확인하려면 다음 연산을 호출합니다.
 - [GetRole](#)
2. 역할의 최대 세션 기간 설정을 업데이트하려면 `max-sessionduration` CLI 파라미터 또는 `MaxSessionDuration` API 파라미터와 함께 다음 연산을 호출합니다.
 - [UpdateRole](#)

다음에 다른 사람이 이 역할을 수입할 때까지 변경 사항은 적용되지 않습니다. 이 역할에 대한 기존 세션을 취소하는 방법을 알아보려면 [IAM 역할의 임시 보안 자격 증명 취소 \(p. 245\)](#) 단원을 참조하십시오.

역할 또는 인스턴스 프로파일 삭제

역할이 더 이상 필요하지 않은 경우 역할 및 연결된 권한을 삭제하는 것이 좋습니다. 따라서 적극적으로 모니터링하거나 유지하지 않는 미사용 개체가 없도록 합니다.

역할이 EC2 인스턴스와 연결된 경우 인스턴스 프로파일에서 해당 역할을 제거한 다음 인스턴스 프로파일을 삭제할 수도 있습니다.

Warning

삭제 예정인 역할 또는 인스턴스 프로필로 실행 중인 Amazon EC2 인스턴스가 없어야 합니다. 실행 중인 인스턴스와 관련된 역할 또는 인스턴스 프로필을 삭제하면 인스턴스에서 실행 중인 애플리케이션이 중단됩니다.

주제

- [역할 액세스 보기 \(p. 255\)](#)
- [서비스 연결 역할 삭제 \(p. 255\)](#)
- [IAM 역할 삭제\(콘솔\) \(p. 255\)](#)
- [IAM 역할 삭제\(AWS CLI\) \(p. 256\)](#)
- [IAM 역할 삭제\(AWS API\) \(p. 257\)](#)
- [관련 정보 \(p. 257\)](#)

역할 액세스 보기

역할을 삭제하기 전에 최근 서비스 수준 활동을 검토해야 합니다. 이 기능은 사용 중인 보안 주체(사람 또는 애플리케이션)의 액세스 권한을 제거하지 않으려는 경우 중요합니다. 서비스에서 마지막으로 액세스한 데이터 보기에 대한 자세한 내용은 [서비스에서 마지막으로 액세스한 데이터를 사용하여 권한 줄이기 \(p. 409\)](#) 단원을 참조하십시오.

서비스 연결 역할 삭제

역할이 [서비스 연결 역할 \(p. 154\)](#)인 경우, 연결된 서비스의 설명서를 참조하여 역할을 삭제하는 방법을 알아보십시오. 콘솔의 IAM 역할 페이지에서 계정의 서비스 연결 역할을 볼 수 있습니다. 서비스 연결 역할은 테이블의 Trusted entities(신뢰할 수 있는 개체) 열에 (Service-linked role)((서비스 연결 역할))로 표시됩니다. 역할의 요약 페이지 배너에도 해당 역할이 서비스 역할임이 표시됩니다.

서비스 연결 역할 삭제에 관한 문서가 서비스에 없는 경우에는 IAM 콘솔, AWS CLI 또는 API를 사용하여 역할을 삭제할 수 있습니다. 자세한 내용은 [서비스 연결 역할 삭제 \(p. 200\)](#) 단원을 참조하십시오.

IAM 역할 삭제(콘솔)

AWS Management 콘솔을 사용하여 역할을 삭제하는 경우, IAM 또한 자동으로 해당 역할과 연결된 정책을 삭제합니다. 해당 역할이 포함된 Amazon EC2 인스턴스 프로파일도 삭제됩니다.

Important

경우에 따라 역할이 Amazon EC2 인스턴스 프로파일과 연결될 수 있으며, 역할과 인스턴스 프로파일은 이름이 같을 수 있습니다. 이 경우 AWS 콘솔을 사용하여 역할과 인스턴스 프로파일을 삭제할 수 있습니다. 이 연결은 콘솔에서 생성한 인스턴스 프로파일과 역할에서 자동으로 발생합니다. AWS CLI, Windows PowerShell용 도구 또는 AWS API에서 역할을 생성한 경우 역할과 인스턴스 프로파일의 이름이 서로 다를 수 있습니다. 이 경우 콘솔을 사용하여 역할과 인스턴스 프로파일을 삭제할 수 없습니다. 그 대신 AWS CLI, Windows PowerShell용 도구 또는 AWS API를 사용하여 먼저 인스턴스 프로파일에서 역할을 제거해야 합니다. 그런 다음 별도의 단계로 역할을 삭제해야 합니다.

역할을 삭제하려면(콘솔 사용)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택한 후 삭제하려는 역할 옆에 있는 확인란(역할 이름이나 행 아님)을 선택합니다.
3. 페이지 상단의 [Role actions]에서 [Delete role]를 선택합니다.
4. 확인 대화 상자가 나타나면 서비스 마지막 액세스 데이터를 검토합니다. 이 데이터는 선택한 각 역할이 AWS 서비스를 마지막으로 액세스한 일시를 보여 줍니다. 이를 통해 역할이 현재 활동 중인지 여부를 확인할 수 있습니다. 계속하려면 예, 삭제를 선택합니다. 확신한다면, 서비스 마지막 액세스 데이터가 로드되고 있을 때에도 삭제를 진행할 수 있습니다.

Note

인스턴스 프로파일이 역할과 이름이 동일한 경우를 제외하고는 콘솔을 사용하여 인스턴스 프로파일을 삭제할 수 없습니다. 또한 이전 절차에서 설명한 바와 같이 역할 삭제 과정의 일부로 인스턴스 프로파일을 삭제해야 합니다. 역할까지 삭제하지 않고 인스턴스 프로파일을 삭제하려면 AWS CLI 또는 AWS API를 사용해야 합니다. 자세한 내용은 다음 단원을 참조하십시오.

IAM 역할 삭제(AWS CLI)

AWS CLI를 사용하여 역할을 삭제하는 경우, 먼저 해당 역할과 연결된 정책을 삭제해야 합니다. 또한 해당 역할이 들어 있는 연결된 인스턴스 프로파일은 별도로 삭제해야 합니다.

역할을 삭제하려면(AWS CLI)

1. 삭제할 역할의 이름을 모르는 경우 다음 명령을 입력하여 계정의 역할을 나열합니다.

```
$ aws iam list-roles
```

역할 목록과 Amazon 리소스 이름(ARN)이 표시됩니다. CLI 명령에서 역할을 참조하려면 ARN이 아니라 역할 이름을 사용해야 합니다. 예를 들어, 어떤 역할의 ARN이 arn:aws:iam::123456789012:role/myrole인 경우 참조할 역할은 **myrole**입니다.

2. 역할이 속해 있는 모든 인스턴스 프로파일에서 역할을 제거합니다.

- a. 해당 역할과 연결된 모든 인스턴스 프로파일을 나열하려면 다음 명령을 입력하십시오.

```
$ aws iam list-instance-profiles-for-role --role-name role-name
```

- b. 인스턴스 프로파일에서 역할을 제거하려면 각 인스턴스 프로파일에 대해 다음 명령을 입력하십시오.

```
$ aws iam remove-role-from-instance-profile --instance-profile-name instance-profile-name --role-name role-name
```

3. 역할과 연결된 모든 정책을 삭제합니다.

- a. 해당 역할에 있는 모든 정책을 나열하려면 다음 명령을 입력하십시오.

```
$ aws iam list-role-policies --role-name role-name
```

- b. 역할에서 각 정책을 삭제하려면 각 정책에 대해 다음 명령을 입력하십시오.

```
$ aws iam delete-role-policy --role-name role-name --policy-name policy-name
```

4. 다음 명령을 입력하여 역할을 삭제합니다.

```
$ aws iam delete-role --role-name role-name
```

5. 역할과 연결된 인스턴스 프로파일을 다시 사용할 계획이 없는 경우 다음 명령을 입력하여 삭제할 수 있습니다.

```
$ aws iam delete-instance-profile --instance-profile-name instance-profile-name
```

IAM 역할 삭제(AWS API)

IAM API를 사용하여 역할을 삭제하려면 먼저 해당 역할과 연결된 정책을 삭제해야 합니다. 또한 해당 역할이 들어 있는 연결된 인스턴스 프로파일은 별도로 삭제해야 합니다.

역할을 삭제하려면(AWS API)

1. 역할이 속해 있는 모든 인스턴스 프로파일을 나열하려면 [ListInstanceProfilesForRole](#)을 호출하십시오.
역할이 속해 있는 모든 인스턴스 프로파일에서 해당 역할을 제거하려면 [RemoveRoleFromInstanceProfile](#)을 호출하십시오. 역할 이름과 인스턴스 프로파일 이름을 전달해야 합니다.
역할과 연결된 인스턴스 프로파일을 다시 사용할 계획이 없는 경우 [DeleteInstanceProfile](#)을 호출하여 삭제할 수 있습니다.
2. 역할에 대한 모든 정책을 나열하려면 [ListRolePolicies](#)를 호출하십시오.
역할과 연결된 모든 정책을 삭제하려면 [DeleteRolePolicy](#)를 호출하십시오. 역할 이름과 정책 이름을 전달해야 합니다.
3. [DeleteRole](#)을 호출하여 역할을 삭제하십시오.

관련 정보

인스턴스 프로파일에 대한 일반적인 정보는 [인스턴스 프로파일 사용 \(p. 244\)](#) 단원을 참조하십시오.

서비스 연결 역할에 대한 일반적인 내용은 [서비스 연결 역할 사용 \(p. 195\)](#) 단원을 참조하십시오.

IAM 역할과 리소스 기반 정책의 차이

일부 AWS 서비스에 대해서는 리소스에 대한 교차 계정 액세스 권한을 부여할 수 있습니다. 이렇게 하려면 역할을 프록시로 사용하는 대신 공유하고자 하는 리소스에 정책을 직접 연결하면 됩니다. 공유하려는 리소스는 반드시 [리소스 기반 정책 \(p. 326\)](#)을 지원해야 합니다. 사용자 기반 정책과 달리 리소스 기반 정책은 해당 리소스에 액세스할 수 있는 사용자(AWS 계정 ID 번호 목록의 형태)를 지정합니다.

리소스 기반 정책을 사용한 교차 계정 액세스는 역할에 비해 몇 가지 이점이 있습니다. 리소스 기반 정책을 통해 액세스한 리소스로 인해 사용자는 여전히 신뢰받는 계정에서 작업을 할 수 있고 역할 권한 대신에 자신의 사용자 권한을 포기할 필요가 없습니다. 다시 말해서 사용자는 자신이 신뢰하는 계정의 리소스에 액세스하는 것과 같은 시각에 신뢰받는 계정의 리소스에 계속해서 액세스합니다. 다른 계정의 공유 리소스로 정보를 복사하거나 공유 리소스의 정보를 복사하는 등의 작업에서 이는 특히 유용합니다.

리소스 기반 정책을 지원하는 몇 가지 AWS 서비스가 여기에 나열되어 있습니다.

- Amazon S3 버킷 – 정책은 버킷과 연결되지만, 버킷과 그 안에 포함된 객체에 대한 액세스를 모두 제어합니다. 자세한 내용은 Amazon Simple Storage Service 개발자 가이드에서 [액세스 제어](#) 단원을 참조하십시오.

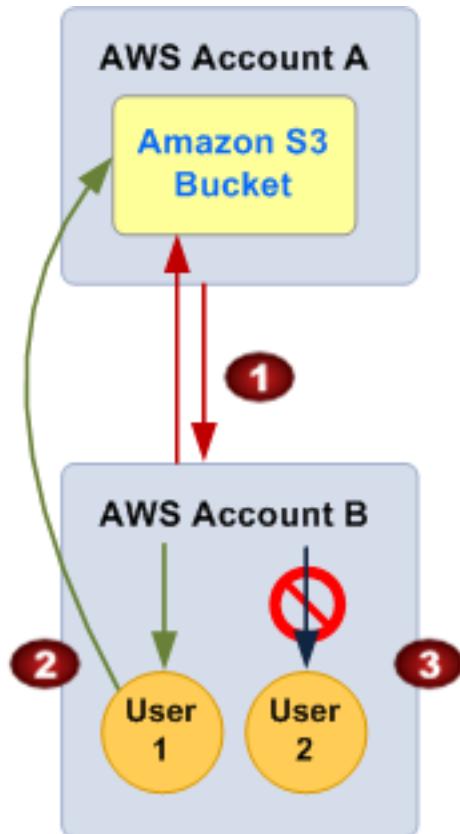
일부의 경우, 교차 계정의 Amazon S3 액세스 권한에 대한 역할을 사용하는 것이 최선일 수 있습니다. 자세한 내용은 Amazon Simple Storage Service 개발자 가이드의 [연습 예제](#)를 참조하십시오.

- Amazon Simple Notification Service(Amazon SNS) 주제 – 자세한 내용은 Amazon Simple Notification Service 개발자 안내서의 [Amazon SNS 주제에 대한 액세스 관리](#) 단원을 참조하십시오.
- Amazon Simple Queue Service(Amazon SQS) 대기열 – 자세한 내용은 Amazon Simple Queue Service 개발자 안내서의 [부록: 액세스 정책 언어](#)를 참조하십시오.

보안 주체 대신 리소스에 권한 정책을 연결할 수 있도록 지원하는 AWS 서비스는 늘어나고 있습니다. 해당 서비스의 전체 목록은 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하고 리소스 기반 정책의 값이 예인 서비스를 찾아보십시오.

리소스 기반 정책에서 AWS 권한 위임에 대하여

리소스 기반 정책에서 리소스가 보안 주체인 AWS 계정에 권한을 부여하면 AWS 계정 내 특정 사용자 또는 그룹에 이 권한을 위임할 수 있습니다. 권한을 위임하려는 사용자 또는 그룹에 정책을 연결합니다. 리소스 소유 계정에서 계정에 부여한 권한과 같거나 더 작은 권한만 위임할 수 있습니다. 예를 들어 계정에 다른 AWS 계정의 리소스에 대한 완전한 액세스 권한이 부여된 경우 AWS 계정 내 사용자에게 완전한 액세스, 조회 액세스 또는 그 밖의 부분 액세스 권한을 위임할 수 있습니다. 반면, 계정에 조회 액세스 권한만 부여된 경우 액세스 조회 액세스 권한만 위임할 수 있습니다. 계정에 부여된 것보다 더 많은 권한을 위임하려고 해도 사용자는 액세스 나열 권한만 갖게 됩니다. 다음 그림에 이 내용이 잘 설명되어 있습니다. 사용자 또는 그룹으로의 정책 연결에 대한 자세한 내용은 [IAM 정책 관리 \(p. 377\)](#) 단원을 참조하십시오.



- 계정 A는 계정 B를 정책의 보안 주체로 지정하여 계정 A의 S3 버킷에 대한 완전한 액세스 권한을 계정 B에게 부여합니다. 그 결과 계정 B는 계정 A의 버킷에서 모든 작업을 수행할 권한이 있으며, 계정 B 관리자는 계정 B에 속한 사용자에게 액세스 권한을 위임할 수 있습니다.
- 계정 B의 관리자가 사용자 1에게 계정 A의 S3 버킷에 대한 읽기 전용 액세스 권한을 부여합니다. 사용자 1은 계정 A의 버킷에 있는 객체를 볼 수 있습니다. 계정 B는 계정에 부여된 액세스 권한과 같거나 그 보다 낮은 수준의 액세스 권한을 위임할 수 있습니다. 이 경우 계정 B에 부여된 완전한 액세스 권한이 사용자 1에게는 읽기 전용으로 필터링됩니다.

3. 계정 B 관리자는 사용자 2에게 액세스 권한을 부여하지 않습니다. 기본적으로 사용자는 명시적으로 부여된 권한을 제외하고는 어떤 권한도 없으므로 사용자 2는 계정 A의 Amazon S3 버킷에 대한 액세스 권한이 없습니다.

Important

위 예에서 계정 B가 와일드카드(*)를 사용하여 사용자 1에게 해당 리소스 전체에 대한 완전한 액세스 권한을 부여한다면, 사용자 1은 다른 계정이 계정 B에게 액세스 권한을 부여한 리소스를 포함하여 계정 B가 액세스하는 어떤 리소스에도 자동적으로 액세스 권한을 갖게 됩니다. 이 경우 사용자 1은 사용자 1에게 명시적으로 권한이 부여된 리소스뿐만 아니라 계정 B에 권한이 부여된 계정 A의 리소스에도 액세스할 수 있습니다.

IAM은 사용자가 요청을 할 때 사용자의 권한을 평가합니다. 따라서 와일드카드(*)를 사용하여 리소스에 대한 완전한 액세스 권한을 부여하면, 사용자 정책을 만든 후에 액세스 권한을 추가 또는 획득한 리소스까지 포함하여 AWS 계정이 액세스하는 모든 리소스에 사용자가 액세스할 수 있습니다.

권한, 정책 및 정책 작성에 사용하는 권한 정책 언어에 대한 자세한 내용은 [액세스 관리 \(p. 304\)](#) 단원을 참조하십시오.

Important

신뢰하는 개체에 한해 필요한 최소 수준의 액세스 권한만 부여하십시오. 신뢰받는 엔터티가 다른 AWS 계정인 경우 언제든지 해당 계정은 IAM 계정에 속한 어떤 사용자에게도 다시 액세스 권한을 위임할 수 있습니다. 신뢰하는 AWS 계정은 권한이 부여된 액세스 범위 내에서만 권한을 위임할 수 있으며, 계정에 부여된 권한보다 더 많은 액세스 권한을 위임할 수 없습니다.

IAM 엔터티 태그 지정

IAM 태그를 사용하여 키-값 페어 태그를 사용하는 IAM 사용자 또는 역할에 사용자 지정 속성을 추가할 수 있습니다. 예를 들어 사용자에게 위치 정보를 추가하려면 태그 키 **location** 및 태그 값 **us_wa_seattle**을 추가할 수 있습니다. 또는 세 개의 개별 위치 태그 키-값 페어 **loc-country = us**, **loc-state = wa** 및 **loc-city = seattle**을 사용할 수도 있습니다. 태그를 사용하여 리소스에 대한 엔터티의 액세스를 제어하거나 엔터티에 연결할 수 있는 태그를 제어할 수 있습니다. 태그를 사용하여 액세스를 제어하는 방법에 대한 자세한 내용은 [IAM 태그를 사용한 액세스 제어 \(p. 336\)](#) 단원을 참조하십시오.

AWS 태그 이름 지정 규칙 선택

IAM 사용자 및 역할에 태그를 첨부하기 시작할 때 태그 이름 지정 규칙을 신중하게 선택하고 모든 AWS 태그에 동일한 규칙을 적용합니다. 정책에서 태그를 사용하여 AWS 리소스에 대한 액세스를 제어하는 경우 특히 중요합니다. AWS에서 태그를 이미 사용하고 있다면 이름 지정 규칙을 검토하고 적절하게 조정하십시오. 이를 지정 전략 만들기에 대한 자세한 내용은 [AWS 태그 지정 전략](#) 단원을 참조하십시오.

IAM 엔터티 태그 지정 규칙

IAM에서 태그의 생성과 적용을 관리하는 규칙은 여러 가지가 있습니다.

태그 이름 지정

IAM 엔터티(사용자 및 역할)에 대한 태그 이름 지정 규칙을 공식화하는 경우 다음 규칙을 준수하십시오.

- 태그 키 및 값은 문자, 숫자, 공백 및 _ : / = + - @ . 기호를 포함할 수 있습니다.
- 태그 키-값 페어는 대소문자를 구분하지 않지만 대소문자는 유지됩니다. 즉, 별도의 **Department** 및 **department** 태그 키를 가질 수 없음을 의미합니다. **Department=foo** 태그로 사용자 태그를 지정하고 **department=bar** 태그를 추가하면 첫 번째 태그가 바뀝니다. 두 번째 태그는 추가되지 않습니다.

- **aws:**로 시작하는 태그 키 또는 값을 생성할 수 없습니다. 이 태그 접두사는 AWS 내부 전용으로 예약되어 있습니다.
- **phoneNumber** = 와 같이 값이 비어 있는 태그를 만들 수 있습니다. 빈 태그 키는 생성할 수 없습니다.
- 단일 태그에 여러 값을 지정할 수 없지만 단일 값으로 사용자 지정 다중 값 구조를 생성할 수 있습니다. 예를 들어 사용자 Zhang이 엔지니어링 팀과 QA 팀에서 근무한다고 가정합니다. **team** = **Engineering** 태그를 연결하고 **team** = **QA** 태그를 연결한 경우 태그 값을 **Engineering**에서 **QA**로 변경합니다. 대신 사용자 지정 구분자를 사용하여 단일 태그에 여러 값을 포함할 수 있습니다. 이 예에서는 Zhang에게 **team** = **Engineering:QA** 태그를 연결할 수 있습니다.

Note

이 예제에서 **team** 태그를 사용하여 엔지니어에 대한 액세스를 제어하려면 **Engineering:QA**를 포함하여 **Engineering**을 포함할 수 있는 모든 구성 허용하는 정책을 만들어야 합니다. 정책에서의 태그 사용에 대한 자세한 내용은 [IAM 태그를 사용한 액세스 제어 \(p. 336\)](#) 단원을 참조하십시오.

태그 적용 및 편집

태그를 IAM 엔터티(사용자 및 역할)에 연결할 때 다음 규칙을 준수하십시오.

- 그룹이나 정책이 아닌 사용자나 역할에 태그를 지정할 수 있습니다.
- Tag Editor를 사용하여 IAM 엔터티에 태그를 지정할 수 없습니다. Tag Editor는 IAM 태그를 지원하지 않습니다. Tag Editor를 다른 서비스와 함께 사용하는 방법에 대한 내용은 AWS Management 콘솔 사용 설명서의 [Tag Editor 작업](#) 단원을 참조하십시오.
- IAM 엔터티에 태그를 지정하려면 특정 권한이 있어야 합니다. 역할과 사용자에 태그를 지정하거나 태그를 해제하려면 태그를 나열할 수 있는 권한이 있어야 합니다. 자세한 내용은 [IAM 엔터티 태그 지정에 필요한 권한 \(p. 260\)](#) 단원을 참조하십시오.
- 라우팅 테이블에 추가할 수 있는 경로의 수에는 제한이 있습니다. 자세한 정보는 [IAM 개체 및 객체에 대한 제한 \(p. 485\)](#) 단원을 참조하십시오.
- 여러 개의 IAM 엔터티에 동일한 태그를 적용할 수 있습니다. 예를 들어 AWS_Development라는 이름의 부서가 12명의 멤버로 구성된 경우 태그 키가 **department**이고 값이 **awsDevelopment(department = awsDevelopment)**인 12개의 사용자와 역할을 가질 수 있습니다. [태그 지정을 지원하는 다른 서비스 \(p. 488\)](#)의 리소스에도 동일한 태그를 사용할 수 있습니다.
- IAM 엔터티는 동일한 태그 키의 여러 인스턴스를 가질 수 없습니다. 예를 들어 태그 키-값 페어 **costCenter = 1234**가 지정된 사용자가 있는 경우 태그 키-값 페어 **costCenter = 5678**을 연결할 수 있습니다. IAM은 **costCenter** 태그의 값을 **5678**로 업데이트합니다.
- IAM 사용자 또는 역할에 연결된 태그를 편집하려면 새 값으로 태그를 연결하여 기존 태그를 덮어씁니다. 예를 들어, 태그 키-값 페어 **department = Engineering**을 가진 사용자가 있다고 가정합니다. 사용자를 QA 부서로 이동해야 하는 경우 **department = QA** 태그 키-값 쌍을 사용자에게 연결할 수 있습니다. 결과적으로 **department** 태그 키의 **Engineering** 값이 **QA** 값으로 대체됩니다.

IAM 엔터티 태그 지정에 필요한 권한

IAM 엔터티(사용자 또는 역할)이 다른 엔터티에 태그를 지정할 수 있도록 권한을 구성해야 합니다. IAM 정책에서 다음 IAM 태그 작업 중 하나 또는 모두를 지정할 수 있습니다.

- **iam>ListRoleTags**
- **iam>ListUserTags**
- **iam:TagRole**
- **iam:TagUser**
- **iam:UntagRole**

- `iam:UntagUser`

특정 사용자의 태그를 추가, 나열 또는 제거하도록 IAM 엔터티를 허용하려면

태그를 관리해야 하는 IAM 엔터티의 권한 정책에 다음 명령문을 추가합니다. 계정 번호를 사용하여 `<username>`을 관리해야 할 사용자 이름으로 바꿉니다. 이 예제 JSON 정책 문서를 사용하여 정책을 생성하는 방법에 대해 자세히 알아보려면 [the section called “JSON 탭에서 정책 만들기” \(p. 381\)](#) 단원을 참조하십시오.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iam>ListUserTags",  
        "iam:TagUser",  
        "iam:UntagUser"  
    ],  
    "Resource": "arn:aws:iam:*<account-number>:user/<username>"  
}
```

IAM 사용자가 태그를 자체 관리할 수 있게 하려면

사용자가 자신의 태그를 관리할 수 있도록 권한 정책에 다음 명령문을 추가합니다. 이 예제 JSON 정책 문서를 사용하여 정책을 생성하는 방법에 대해 자세히 알아보려면 [the section called “JSON 탭에서 정책 만들기” \(p. 381\)](#) 단원을 참조하십시오.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iam>ListUserTags",  
        "iam:TagUser",  
        "iam:UntagUser"  
    ],  
    "Resource": "arn:aws:iam:*<aws:username>"  
}
```

IAM 엔터티를 사용하여 특정 사용자에게 태그를 추가하려면

특정 사용자의 태그를 추가하기만 하고 제거하지는 않으려면 IAM 엔터티의 권한 정책에 다음 명령문을 추가합니다.

Note

`iam:AddRoleTags` 및 `iam:AddUserTags` 작업을 수행하려면 `iam>ListRoleTags` 및 `iam>ListUserTags` 작업도 포함해야 합니다.

이 정책을 사용하려면 `<username>`을 관리해야 할 사용자 이름으로 바꿉니다. 이 예제 JSON 정책 문서를 사용하여 정책을 생성하는 방법에 대해 자세히 알아보려면 [the section called “JSON 탭에서 정책 만들기” \(p. 381\)](#) 단원을 참조하십시오.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iam>ListUserTags",  
        "iam:TagUser"  
    ],  
    "Resource": "arn:aws:iam:*<account-number>:user/<username>"  
}
```

특정 역할의 태그를 추가, 나열 또는 제거하도록 IAM 엔터티를 허용하려면

태그를 관리해야 하는 IAM 엔터티의 권한 정책에 다음 명령문을 추가하여 <rolename>을 관리해야 하는 역할의 이름으로 바꿉니다. 이 예제 JSON 정책 문서를 사용하여 정책을 생성하는 방법에 대해 자세히 알아보려면 the section called “JSON 탭에서 정책 만들기” (p. 381) 단원을 참조하십시오.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iam>ListRoleTags",  
        "iam:TagRole",  
        "iam:UntagRole"  
    ],  
    "Resource": "arn:aws:iam:*:<account-number>:role/<rolename>"  
}
```

또는 [IAMFullAccess](#) 등의 AWS 관리형 정책을 사용하여 IAM에 모든 액세스 권한을 제공할 수 있습니다.

IAM 엔터티에 대한 태그 관리(콘솔)

AWS Management 콘솔에서 IAM 사용자 또는 역할에 대한 태그를 관리할 수 있습니다.

사용자 또는 역할에 대한 태그를 관리하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 콘솔의 탐색 창에서 역할 또는 사용자를 선택한 다음 편집할 엔터티의 이름을 선택합니다.
3. 태그 탭을 선택하고 다음 작업 중 하나를 완료하십시오.
 - 엔터티에 아직 태그가 없는 경우 태그 추가를 선택합니다.
 - 기존 태그 세트를 관리하려면 태그 편집을 선택합니다.
4. 태그를 추가하거나 제거하여 태그 세트를 완성합니다. 변경 사항 저장을 선택합니다.

IAM 엔터티에 대한 태그 관리(AWS CLI 또는 AWS API)

IAM 사용자 및 역할에 대한 태그를 나열, 연결 또는 제거할 수 있습니다. AWS CLI 또는 AWS API를 사용하여 IAM 사용자 및 역할에 대한 태그를 관리할 수 있습니다.

IAM 역할에 현재 연결된 태그를 나열하려면(AWS CLI 또는 AWS API)

- AWS CLI: [aws iam list-role-tags](#)
- AWS API: [ListRoleTags](#)

IAM 사용자에 현재 연결된 태그를 나열하려면(AWS CLI 또는 AWS API)

- AWS CLI: [aws iam list-user-tags](#)
- AWS API: [ListUserTags](#)

IAM 역할에 태그를 연결하려면(AWS CLI 또는 AWS API)

- AWS CLI: [aws iam tag-role](#)
- AWS API: [TagRole](#)

IAM 사용자에 태그를 연결하려면(AWS CLI 또는 AWS API)

- AWS CLI: [aws iam tag-user](#)
- AWS API: [TagUser](#)

IAM 역할에서 태그를 제거하려면(AWS CLI 또는 AWS API)

- AWS CLI: [aws iam untag-role](#)
- AWS API: [UntagRole](#)

IAM 사용자의 태그를 제거하려면(AWS CLI 또는 AWS API)

- AWS CLI: [aws iam untag-user](#)
- AWS API: [UntagUser](#)

다른 AWS 서비스의 리소스에 대한 태그 연결 정보는 해당 서비스의 설명서를 참조하십시오.

태그를 사용하여 IAM 권한 정책으로 보다 세부적인 권한을 설정하는 방법에 대한 자세한 내용은 [IAM 정책 요소: 변수 및 태그 \(p. 524\)](#) 단원을 참조하십시오.

임시 보안 자격 증명

AWS Security Token Service(AWS STS)를 사용하면 AWS 리소스에 대한 액세스를 제어할 수 있는 임시 보안 자격 증명을 생성하여 신뢰받는 사용자에게 제공할 수 있습니다. 임시 보안 자격 증명은 다음과 같은 차이점을 제외하고는 IAM 사용자가 사용할 수 있는 장기 액세스 키 자격 증명과 거의 동일한 효력을 지닙니다.

- 임시 보안 자격 증명은 그 이름이 암시하듯 단기적입니다. 이 자격 증명은 몇 분에서 몇 시간까지 지속되도록 구성할 수 있습니다. 자격 증명이 만료된 후 AWS는 더는 그 자격 증명을 인식하지 못하거나 그 자격 증명을 사용한 API 요청으로부터 이루어지는 어떤 종류의 액세스도 허용하지 않습니다.
- 임시 보안 자격 증명은 사용자와 함께 저장되지 않지만 동적으로 생성되어 요청 시 사용자에게 제공됩니다. 임시 보안 자격 증명이 만료되었을 때(심지어는 만료 전이라도) 사용자는 새 자격 증명을 요청할 수 있습니다. 단, 자격 증명을 요청하는 해당 사용자에게 그렇게 할 수 있는 권한이 있어야 합니다.

이러한 차이점은 다음과 같은 임시 자격 증명 사용의 이점을 발생시킬 수 있습니다.

- 애플리케이션으로 장기 AWS 보안 자격 증명을 배포 또는 포함할 필요가 없습니다.
- 사용자에 대한 AWS 자격 증명을 정의하지 않고도 AWS 리소스에 대한 액세스 권한을 사용자에게 제공할 수 있습니다. 임시 자격 증명은 [역할 및 자격 증명 연동 \(p. 153\)](#)을 위한 기초입니다.
- 임시 보안 자격 증명은 수명이 제한되어 있어서, 더 이상 필요하지 않을 때 교체하거나 명시적으로 취소할 필요가 없습니다. 임시 보안 자격 증명이 만료된 후에는 다시 사용할 수 없습니다. 그 자격 증명에 대해 유효 기간을 최대 한계까지 지정할 수 있습니다.

AWS STS 및 AWS 리전

임시 보안 자격 증명은 AWS STS에 의해 생성됩니다. 기본적으로 AWS STS는 <https://sts.amazonaws.com>에 단일 엔드포인트가 있는 전역적 서비스입니다. 그러나 지원되는 기타 다른 리전에서 엔드포인트에 대한 AWS STS API 호출을 할 수도 있습니다. 이렇게 지리적으로 더 가까운 리전에 있는 서버로 요청을 전송함으로써 지역 시간(서버 랙)을 단축할 수 있습니다. 자격 증명은 어떤 리전에서 오는지 상관없이 전역적으로 유효합니다. 자세한 내용은 [AWS 리전에서 AWS STS 활성화 및 비활성화 \(p. 287\)](#) 단원을 참조하십시오.

임시 자격 증명과 관련된 일반적인 시나리오

임시 자격 증명은 자격 증명 연동, 위임, 교차 계정 액세스, IAM 역할 등의 시나리오에서 유용합니다.

ID 페더레이션

AWS 밖의 외부 시스템에서 사용자 자격 증명을 관리할 수 있고 그 시스템으로부터 로그인하는 사용자에게 액세스 권한을 부여하여 AWS 작업을 수행하고 AWS 리소스에 액세스하도록 할 수 있습니다. IAM은 두 가지 유형의 자격 증명 연동을 지원합니다. 두 경우 모두 자격 증명은 AWS 외부에 저장됩니다. 차이는 외부 시스템이 상주하는 곳이 어디인가 즉, 데이터센터인가 아니면 웹 상의 외부 타사인가 하는 데 있습니다. 외부 자격 증명 공급자에 대한 자세한 내용은 [자격 증명 공급자 및 연동 \(p. 161\)](#) 단원을 참조하십시오.

- 엔터프라이즈 자격 증명 연동 – 조직의 네트워크에서 사용자를 인증한 다음, 해당 사용자에 대한 새로운 AWS 자격 증명을 생성하지 않고 또한, 사용자에게 별도의 사용자 이름 및 암호로 로그인하도록 요구하지 않고도 AWS에 대한 액세스 권한을 사용자에게 제공할 수 있습니다. 이는 임시 액세스 권한에 대한 SSO(Single Sign-On) 접근 방식으로 알려져 있습니다. AWS STS는 SAML 2.0(Security Assertion Markup Language 2.0)과 같은 개방형 표준을 지원합니다. 이를 통해 Microsoft AD FS를 사용해 Microsoft Active Directory를 최대한 활용할 수 있습니다. 또한, SAML 2.0을 사용해 사용자 자격 증명 연동을 위한 자신만의 솔루션을 관리할 수 있습니다. 자세한 내용은 [SAML 2.0 기반 연동에 대하여 \(p. 167\)](#)을 참조하십시오.
- 사용자 지정 연동 브로커 – 조직의 인증 시스템을 사용해 AWS 리소스에 대한 액세스 권한을 부여할 수 있습니다. 시나리오 예시는 [연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하는 URL 생성 \(사용자 지정 연동 브로커\) \(p. 188\)](#) 단원을 참조하십시오.
- SAML 2.0을 사용한 연동 – 조직의 인증 시스템과 SAML을 사용해 AWS 리소스에 대한 액세스를 허용할 수 있습니다. 자세한 내용과 시나리오 예시는 [SAML 2.0 기반 연동에 대하여 \(p. 167\)](#) 단원을 참조하십시오.
- 웹 자격 증명 연동 – Login with Amazon, Facebook, Google 또는 OpenID Connect(OIDC) 2.0 호환 공급자와 같은 유명한 타사 자격 증명 공급자를 사용해 사용자가 로그인할 수 있습니다. 그 공급자로부터 얻은 자격 증명을 AWS 계정 리소스 사용 권한과 교환할 수 있습니다. 이는 임시 액세스 권한에 대한 웹 자격 증명 연동 접근 방식으로 알려져 있습니다. 모바일 또는 웹 애플리케이션을 위해 웹 자격 증명 연동을 사용하면 사용자 지정 로그인 코드를 생성하거나 자신의 사용자 자격 증명을 관리할 필요가 없습니다. 웹 자격 증명 연동을 사용하면 AWS 계정을 안전하게 보호할 수 있다는 이점이 있습니다. 애플리케이션으로 IAM 사용자 액세스 키 같은 장기 보안 자격 증명을 배포할 필요가 없기 때문입니다. 자세한 내용은 [웹 자격 증명 연동에 대하여 \(p. 162\)](#)을 참조하십시오.

AWS STS 웹 자격 증명 연동은 Login with Amazon, Facebook, Google 및 모든 OpenID Connect(OIDC) 호환 자격 증명 공급자를 지원합니다.

Note

모바일 애플리케이션의 경우 Amazon Cognito 사용을 권장합니다. 이 서비스와 함께 [AWS iOS용 Mobile SDK](#), [AWS Android 및 Fire OS용 Mobile SDK](#)를 사용하여 사용자 고유 자격 증명을 만들고 AWS 리소스에 대한 보안 액세스를 인증할 수 있습니다. Amazon Cognito는 AWS STS와 동일한 자격 증명 제공자를 지원하며 인증되지 않은(게스트) 액세스도 지원하고 로그인하면 사용자 데이터를 마이그레이션할 수 있습니다. Amazon Cognito는 디바이스를 바꿔 가며 이용해도 데이터를 보존하도록 사용자 데이터 동기화를 위한 API 작업도 제공합니다. 자세한 내용은 다음 자료를 참조하십시오.

- AWS iOS용 Mobile SDK 개발자 안내서의 [Amazon Cognito 자격 증명](#)
- AWS Android용 Mobile SDK 개발자 안내서의 [Amazon Cognito 자격 증명](#)

교차 계정 액세스를 위한 역할

많은 조직이 1개 이상의 AWS 계정을 유지합니다. 역할 및 교차 계정 액세스를 사용하면 하나의 계정에서 사용자 자격 증명을 정의하고 그 자격 증명을 사용해 조직에 속한 다른 계정의 AWS 리소스에 액세스할 수 있습니다. 이는 임시 액세스 권한에 대한 위임 접근 방식으로 알려져 있습니다. 자세한 내용은 [역할을 만들어 IAM 사용자에게 권한 위임 \(p. 203\)](#)을 참조하십시오.

Amazon EC2의 역할

Amazon EC2 인스턴스에서 애플리케이션을 실행할 때 그 애플리케이션이 AWS 리소스에 대한 액세스 권한이 필요한 경우 인스턴스 시작 시 인스턴스에 대한 임시 보안 자격 증명을 제공할 수 있습니다. 이 임시 보안 자격 증명은 인스턴스에서 실행되는 모든 애플리케이션에서 사용 가능하므로 그 인스턴스에 어떤 장기 자격 증명도 저장할 필요가 없습니다. 자세한 내용은 [IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여하기 \(p. 240\)](#)를 참조하십시오.

기타 AWS 서비스

임시 보안 자격 증명을 사용해 대부분의 AWS 서비스에 액세스할 수 있습니다. 임시 보안 자격 증명을 수락하는 서비스의 목록은 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하십시오.

임시 보안 자격 증명 요청하기

임시 보안 자격 증명을 요청하려는 경우 AWS API의 AWS Security Token Service(AWS STS) 작업을 사용하여 AWS 리소스에 대한 액세스를 제어할 수 있는 임시 보안 자격 증명을 생성한 후 신뢰받는 사용자에게 제공할 수 있습니다. For more information about AWS STS, see [임시 보안 자격 증명 \(p. 263\)](#). 역할을 수임해 임시 보안 자격 증명을 요청하는 데 사용할 수 있는 여러 방법을 알아보려면 [IAM 역할 사용 \(p. 227\)](#) 단원을 참조하십시오.

API 작업을 호출하려면 [AWS SDK](#) 중 하나를 Java, .NET, Python, Ruby, Android 및 iOS 등 다양한 프로그래밍 언어 및 환경에서 사용할 수 있습니다. SDK는 요청에 암호화 방식으로 서명, 필요한 경우 요청 재시도, 오류 응답 처리와 같은 작업들을 다룹니다. [AWS Security Token Service API Reference](#)에 기술된 AWS STS 쿼리 API를 사용할 수도 있습니다. 끝으로 [AWS Command Line Interface](#) 및 [Windows PowerShell용 AWS 도구](#)라는 두 가지 명령줄 도구가 AWS STS 명령을 지원합니다.

AWS STS API 작업은 액세스 키 및 세션 토큰으로 구성된 임시 보안 자격 증명을 사용하여 새 세션을 생성합니다. 액세스 키는 액세스 키 ID와 보안 키로 구성되어 있습니다. 사용자(또는 사용자가 실행하는 애플리케이션)는 이 자격 증명을 사용해 리소스에 액세스할 수 있습니다. AWS STS API 작업을 사용하여 프로그래밍 방식으로 역할 세션을 생성하고 세션 정책을 전달할 수 있습니다. 결과적으로 세션에는 엔터티의 자격 증명 기반 정책과 세션 정책, 이 두 정책 모두에 의해 공통적으로 부여되는 권한만 부여됩니다. 세션 정책에 대한 자세한 정보는 [세션 정책을 참조하십시오.](#)

Note

STS API 작업이 반환하는 보안 토큰의 크기는 고정적이지 않습니다. 따라서 최대 크기를 가정하지 않는 것이 좋습니다. 이 문서의 작성일 현재, 일반적인 크기는 4096바이트 미만이나 경우에 따라 더 클 수 있습니다. 또한 AWS로 추후 업데이트 시 더 큰 크기가 필요할 수 있습니다.

AWS 리전에서 AWS STS 사용하기

전역적 엔드포인트 또는 리전 엔드포인트 중 하나에 AWS STS API 호출을 전송할 수 있습니다. 더 가까이 있는 엔드포인트를 선택하면 지역 시간을 단축해 API 호출의 성능을 향상 시킬 수 있습니다. 또한, 원래 엔드포인트와 더 이상 교신하지 않는 경우 대체 리전 엔드포인트에 호출을 직접 보내는 방법을 선택할 수 있습니다. 다양한 AWS SDK 중 하나를 사용하고 있다면 API 호출 전에 그 SDK의 메서드를 사용해 리전을 선택하십시오. HTTP API 요청을 수동으로 구축하는 경우 그 요청을 정확한 엔드포인트에 직접 전송해야 합니다. 자세한 정보는 [리전 및 엔드포인트의 AWS STS 세션 및 AWS 리전에서 AWS STS 활성화 및 비활성화 \(p. 287\)](#) 단원을 참조하십시오.

다음은 AWS 환경 및 애플리케이션에서 사용할 임시 자격 증명을 획득하는 데 사용할 수 있는 API 작업입니다.

AssumeRole—사용자 지정 자격 증명 브로커를 통한 교차 계정 위임과 연동

`AssumeRole` API 작업은 기존 IAM 사용자에게 액세스 권한이 없는 AWS 리소스(예: 다른 AWS 계정에 있는 리소스)에 액세스할 수 있도록 허용하는 데 유용합니다. 또한, 기존 사용자에게는 임시로 액세스 특권을 얻는

수단으로서 유용합니다. 예를 들면 멀티 팩터 인증(MFA)을 제공할 수 있습니다. 기존 IAM 사용자 자격 증명을 사용해 이 API를 호출해야 합니다. 자세한 정보는 [역할을 만들어 IAM 사용자에게 권한 위임 \(p. 203\)](#) 및 [MFA 보호 API 액세스 구성 \(p. 122\)](#)을(를) 참조하십시오.

이 호출에는 반드시 유효한 AWS 보안 자격 증명을 사용해야 합니다. 이 호출을 할 때 다음과 같은 정보를 전달하게 됩니다.

- 앱이 수임해야 하는 역할의 Amazon 리소스 이름(ARN)
- 임시 보안 자격 증명의 기간을 지정하는 유효 기간. DurationSeconds 파라미터를 사용하여 역할 세션 기간을 900초(15초)에서 해당 역할에 대한 최대 세션 기간 설정까지 지정합니다. 역할에 대한 최댓값을 확인하는 방법을 알아보려면 [역할에 대한 최대 세션 기간 설정 보기 \(p. 228\)](#) 단원을 참조하십시오. 이 파라미터를 전달하지 않으면 임시 자격 증명이 한 시간 내에 만료됩니다. 이 API의 DurationSeconds 파라미터는 콘솔 세션의 기간을 지정하는 데 사용하는 SessionDuration HTTP 파라미터와 다릅니다. 콘솔로 그인 토큰의 연동 앤드포인트에 대한 요청에는 SessionDuration HTTP 파라미터를 사용하십시오. 자세한 정보는 [연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하는 URL 생성\(사용자 지정 연동 브로커\) \(p. 188\)](#) 단원을 참조하십시오.
- 역할 세션 이름은 세션을 식별하는데 사용할 수 있는 문자열 값입니다. 이 값은 CloudTrail가 캡처하고 로깅하여, 감사하는 동안 역할 사용자들을 구분하는데 도움이 될 수 있습니다.
- (선택 사항) 세션 정책(JSON 형식). 이 정책은 역할 세션에 할당된 역할 자격 증명 기반 정책의 권한을 제한합니다. 결과적으로 세션에는 역할의 자격 증명 기반 정책과 세션 정책, 이 두 정책 모두에 의해 부여되는 권한만 부여됩니다. 이 정책은 위임된 역할에 허용된 액세스 권한을 넘어서도록 권한을 승격하는데 사용될 수 없다는 것에 유의하십시오. 역할 세션 권한에 대한 자세한 정보는 [세션 정책 \(p. 307\)](#) 단원을 참조하십시오.
- MFA(멀티 팩터 인증)를 사용하도록 구성한 경우, MFA 디바이스의 식별자와 해당 디바이스에서 제공한 일회용 코드를 포함시켜야 합니다.
- 계정에 대한 액세스 권한을 타사에 위임할 때 사용할 수 있는 ExternalId 값(선택 사항)입니다. 이 값은 지정된 타사만 역할에 액세스할 수 있도록 하는 데 도움이 됩니다. 자세한 정보는 [AWS 리소스에 대한 액세스를 타사에 부여할 때 외부 ID를 사용하는 방법 \(p. 206\)](#) 단원을 참조하십시오.

다음 예제에서는 AssumeRole을 사용한 샘플 요청 및 응답을 보여줍니다. 예를 들어 요청에는 Bob이라는 세션 이름이 포함되어 있습니다. Policy 파라미터에는 결과물로 얻은 자격 증명이 Amazon S3에만 액세스 권한을 갖도록 지정하는 JSON 문서가 포함되어 있습니다.

Example 요청

```
https://sts.amazonaws.com/
?Version=2011-06-15
&Action=AssumeRole
&RoleSessionName=Bob
&RoleArn=arn:aws::iam::123456789012:role/demo
&Policy=%7B%22Version%22%3A%222012-10-17%22%2C%22Statement%22%3A%5B%7B%22Sid%22%3A
%20%22Stmt1%22%2C%22Effect%22%3A%20%22Allow%22%2C%22Action%22%3A%20%22s3%3A*%22%2C
%22Resource%22%3A%20%22*%22%7D%5D%7D
&DurationSeconds=1800
&ExternalId=123ABC
&AUTHPARAMS
```

Note

이전 예제의 정책 값은 다음 정책을 URL로 인코딩한 버전입니다.

```
{"Version": "2012-10-17", "Statement": [
    {"Sid": "Stmt1", "Effect": "Allow", "Action": "s3:*", "Resource": "*"}]}
```

또한 이 예에서 AUTHPARAMS 파라미터는 인증 정보의 자리 표시자, 즉 AWS HTTP API 요청에 포함해야 하는 서명입니다. [AWS SDK](#)를 사용하여 API 요청을 생성하는 것이 좋습니다. 이렇게 하면 SDK가 요청 서명을 대신 처리한다는 장점이 있습니다. API 요청을 직접 생성하여 서명해야 할 경우

Amazon Web Services 일반 참조의 [서명 버전 4를 사용하여 AWS 요청에 서명](#)에서 요청에 서명하는 방법 단원을 참조하십시오.

그 응답에는 임시 보안 자격 증명뿐만 아니라 연동 사용자 및 자격 증명 만료 시간에 대한 Amazon 리소스 아이템(ARN)이 포함되어 있습니다.

Example 응답

```
<AssumeRoleResponse xmlns="https://sts.amazonaws.com/doc/2011-06-15/">
<AssumeRoleResult>
<Credentials>
<SessionToken>
AQoDYXdzEPT//////////wEXAMPLEtc764bNrC9SAPBSM22wD0k4x4HIZ8j4FZTwdQW
LWsKWHGBuFqwAeMicRXmxfpSPfIeoIYRqTf1fKD8YUuwthAx7mSEI/qkPpKPi/kMcGd
QrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburEDy0KPkyQDYwT7WZ0wq5VSXDvp75YU
9HFv1Rd8Tx6q6fE8YQcHNVXAkiY9q6d+xo0rKwT38xVqr7ZD0u0iPPkUL641IZbqBAz
+scqKmlzm8FDrypNC9Yjc8fPOLn9FX9KSvVKTTr4rvx3iSiLTJabIQwj2ICCR/oLxBA==

</SessionToken>
<SecretAccessKey>
wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY
</SecretAccessKey>
<Expiration>2011-07-15T23:28:33.359Z</Expiration>
<AccessKeyId>AKIAIOSFODNN7EXAMPLE</AccessKeyId>
</Credentials>
<AssumedRoleUser>
<Arn>arn:aws:sts::123456789012:assumed-role/demo/Bob</Arn>
<AssumedRoleId>ARO123EXAMPLE123:Bob</AssumedRoleId>
</AssumedRoleUser>
<PackedPolicySize>6</PackedPolicySize>
</AssumeRoleResult>
<ResponseMetadata>
<RequestId>c6104cbe-af31-11e0-8154-cbc7ccf896c7</RequestId>
</ResponseMetadata>
</AssumeRoleResponse>
```

Note

AssumeRole은 정책을 밀집 형식으로 저장합니다. AssumeRole은 허용된 최대 크기의 비율로 크기를 반환하므로 호출 파라미터를 조정할 수 있습니다. 정책 크기 제한에 대한 자세한 정보는 AWS Security Token Service API Reference의 [AssumeRole](#) 단원을 참조하십시오.

AssumeRoleWithWebIdentity—웹 기반 자격 증명 공급자를 통한 연동

AssumeRoleWithWebIdentity API 작업은 퍼블릭 자격 증명 공급자를 통해 인증된 연합된 사용자의 임시 보안 자격 증명 세트를 반환합니다. 퍼블릭 자격 증명 공급자의 예에는 Login with Amazon, Facebook, Google 또는 OpenID Connect(OIDC) 호환 자격 증명 공급자 등이 있습니다. 이 API는 사용자에게 고유한 AWS 또는 IAM 자격 증명이 없는 AWS에 대한 액세스가 필요한 모바일 애플리케이션 또는 클라이언트 기반 애플리케이션을 생성하는데 유용합니다. 자세한 정보는 [웹 자격 증명 연동에 대하여 \(p. 162\)](#) 단원을 참조하십시오.

Note

AssumeRoleWithWebIdentity를 직접 호출하는 대신 모바일 개발을 위한 AWS SDK에서 Amazon Cognito 및 Amazon Cognito 자격 증명 공급자를 사용하실 것을 권장합니다. 자세한 정보는 다음을 참조하십시오.

- Android용 AWS Mobile SDK Developer Guide의 [Amazon Cognito 자격 증명](#)
- AWS Mobile SDK for iOS Developer Guide의 [Amazon Cognito 자격 증명](#)

Amazon Cognito를 사용하고 있지 않다면 AWS STS의 `AssumeRoleWithWebIdentity` 작업을 호출합니다. 이것은 서명되지 않은 호출로서 앱이 이 호출을 하기 위해 어떤 AWS 보안 자격 증명에도 액세스할 필요가 없음을 뜻합니다. 이 호출을 할 때 다음과 같은 정보를 전달하게 됩니다.

- 앱이 수임해야 하는 역할의 Amazon 리소스 이름(ARN) 앱이 사용자가 로그인하는 여러 가지 방식을 지원하는 경우 다양한 역할, 즉 자격 증명 공급자당 하나의 역할을 정의해야 합니다. `AssumeRoleWithWebIdentity`에 대한 호출에는 사용자가 로그인할 때 사용한 공급자에 특정된 역할의 ARN이 포함되어야 합니다.
- 앱이 사용자를 인증한 후에 IdP로부터 얻는 토큰
- 임시 보안 자격 증명의 기간을 지정하는 유효 기간. `DurationSeconds` 파라미터를 사용하여 역할 세션 기간을 900초(15초)에서 해당 역할에 대한 최대 세션 기간 설정까지 지정합니다. 역할에 대한 최댓값을 확인하는 방법을 알아보려면 [역할에 대한 최대 세션 기간 설정 보기 \(p. 228\)](#) 단원을 참조하십시오. 이 파라미터를 전달하지 않으면 임시 자격 증명이 한 시간 내에 만료됩니다. 이 API의 `DurationSeconds` 파라미터는 콘솔 세션의 기간을 지정하는 데 사용하는 `SessionDuration` HTTP 파라미터와 다릅니다. 콘솔로 그인 토큰의 연동 엔드포인트에 대한 요청에는 `SessionDuration` HTTP 파라미터를 사용하십시오. 자세한 정보는 [연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하는 URL 생성\(사용자 지정 연동 브로커\) \(p. 188\)](#) 단원을 참조하십시오.
- 역할 세션 이름은 세션을 식별하는 데 사용할 수 있는 문자열 값입니다. 이 값은 CloudTrail가 캡처하고 로깅하여, 감사하는 동안 역할 사용자들을 구분하는 데 도움이 될 수 있습니다.
- (선택 사항) 세션 정책(JSON 형식). 이 정책은 역할 세션에 할당된 역할 자격 증명 기반 정책의 권한을 제한합니다. 결과적으로 세션에는 역할의 자격 증명 기반 정책과 세션 정책, 이 두 정책 모두에 의해 부여되는 권한만 부여됩니다. 이 정책은 위임된 역할에 허용된 액세스 권한을 넘어서도록 권한을 승격하는 데 사용될 수 없다는 것에 유의하십시오. 역할 세션 권한에 대한 자세한 정보는 [세션 정책 \(p. 307\)](#) 단원을 참조하십시오.

Note

`AssumeRoleWithWebIdentity`에 대한 호출이 서명(암호화)되지 않았습니다. 따라서 요청이 신뢰할 수 있는 중개자를 통해 전송된 경우에만 선택적 세션 정책을 포함해야 합니다. 이러한 경우 누군가가 정책을 변경해 제한을 제거할 수 있습니다.

`AssumeRoleWithWebIdentity`를 호출하면 AWS가 토큰의 신뢰성을 확인합니다. 예를 들어 공급자에 따라 AWS는 해당 공급자를 호출해 앱이 전달한 토큰을 포함할 수 있습니다. 자격 증명 공급자가 토큰을 확인한다고 가정하면, AWS는 다음 정보를 반환합니다.

- 일련의 임시 보안 자격 증명 이러한 임시 보안 자격 증명은 액세스 키 ID, 보안 액세스 키 및 세션 토큰으로 이루어져 있습니다.
- 위임된 역할의 역할 ID 및 ARN
- 고유한 사용자 ID를 포함하는 `SubjectFromWebIdentityToken` 값

임시 보안 자격 증명이 있으면 AWS API 호출에 사용할 수 있습니다. 이는 AWS가 임시 보안 자격 증명이 유효한지 확인하도록 허용하는 세션 토큰을 포함해야 한다는 점만 제외하면 장기 보안 자격 증명으로 AWS API 호출을 하는 프로세스와 동일합니다.

앱은 자격 증명을 캐싱해야 합니다. 언급한 바와 같이, 자격 증명은 한 시간 후에 만료되도록 기본 설정되어 있습니다. AWS SDK의 [AmazonSTSCredentialsProvider](#) 작업을 사용하지 않은 경우 `AssumeRoleWithWebIdentity`를 직접 다시 호출해야 합니다. 이전 자격 증명이 만료되기 전에 이 작업을 호출하여 임시 보안 자격 증명 세트를 새로 받으십시오.

AssumeRoleWithSAML—SAML 2.0과 호환되는 엔터프라이즈 자격 증명 공급자를 통한 연동

`AssumeRoleWithSAML` API 작업은 조직의 기존 자격 증명 시스템을 통해 인증된 연합된 사용자의 임시 보안 자격 증명 세트를 반환합니다. 또한 사용자는 [SAML](#) 2.0(Security Assertion Markup Language)을 사용하

여 AWS에 인증 및 권한 부여 정보를 전달해야 합니다. 이 API 작업은 자격 증명 시스템(예: Windows Active Directory 또는 OpenLDAP)을 SAML 어설션을 생성할 수 있는 소프트웨어와 통합한 조직에 유용합니다. 이러한 통합은 사용자 자격 증명 및 권한에 대한 정보를 제공합니다(예: Active Directory Federation Services 또는 Shibboleth). 자세한 정보는 [SAML 2.0 기반 연동에 대하여 \(p. 167\)](#) 단원을 참조하십시오.

이것은 서명되지 않은 호출로서 앱이 이 호출을 하기 위해 어떤 AWS 보안 자격 증명에도 액세스할 필요가 없음을 뜻합니다. 이 호출을 할 때 다음과 같은 정보를 전달하게 됩니다.

- 앱이 수임해야 하는 역할의 Amazon 리소스 이름(ARN)
- 자격 증명 공급자에 대해 기술하는 IAM에서 만든 SAML 자격 증명 공급자의 ARN
- 앱의 로그인 요청에 대한 인증 응답 시 SAML 자격 증명 공급자가 제공한 base-64 인코딩 SAML 어설션
- 임시 보안 자격 증명의 기간을 지정하는 유효 기간 DurationSeconds 파라미터를 사용하여 역할 세션 기간을 900초(15초)에서 해당 역할에 대한 최대 세션 기간 설정까지 지정합니다. 역할에 대한 최댓값을 확인하는 방법을 알아보려면 [역할에 대한 최대 세션 기간 설정 보기 \(p. 228\)](#) 단원을 참조하십시오. 이 파라미터를 전달하지 않으면 임시 자격 증명이 한 시간 내에 만료됩니다. 이 API의 DurationSeconds 파라미터는 콘솔 세션의 기간을 지정하는 데 사용하는 SessionDuration HTTP 파라미터와 다릅니다. 콘솔로 그인 토큰의 연동 엔드포인트에 대한 요청에는 SessionDuration HTTP 파라미터를 사용하십시오. 자세한 정보는 [연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하는 URL 생성\(사용자 지정 연동 브로커\) \(p. 188\)](#) 단원을 참조하십시오.
- (선택 사항) 세션 정책(JSON 형식). 이 정책은 역할 세션에 할당된 역할 자격 증명 기반 정책의 권한을 제한합니다. 결과적으로 세션에는 역할의 자격 증명 기반 정책과 세션 정책, 이 두 정책 모두에 의해 부여되는 권한만 부여됩니다. 이 정책은 위임된 역할에 허용된 액세스 권한을 넘어서도록 권한을 승격하는 데 사용될 수 없다는 것에 유의하십시오. 역할 세션 권한에 대한 자세한 정보는 [세션 정책 \(p. 307\)](#) 단원을 참조하십시오.

`AssumeRoleWithSAML`을 호출하면 AWS가 SAML 어설션의 신뢰성을 확인합니다. 자격 증명 공급자가 어설션을 확인한다고 가정하면, AWS는 다음 정보를 반환합니다.

- 일련의 임시 보안 자격 증명 이러한 임시 보안 자격 증명은 액세스 키 ID, 보안 액세스 키 및 세션 토큰으로 이루어져 있습니다.
- 위임된 역할의 역할 ID 및 ARN
- SAML 어설션의 Audience 요소의 Recipient 속성 값을 포함하는 SubjectConfirmationData 값
- SAML 어설션의 Issuer 요소 값을 포함하는 Issuer 값
- Issuer 값, AWS 계정 ID, SAML 공급자의 표시 이름으로 구축된 해시 값을 포함하는 NameQualifier 요소 Subject 요소와 결합되면 연동 사용자를 고유한 이름으로 식별할 수 있습니다.
- SAML 어설션의 Subject 요소에 있는 NameID 요소의 값을 포함하는 Subject 요소
- SubjectType 요소의 형식을 나타내는 Subject 요소 그 값은 persistent, transient, 또는 SAML 어설션에서 사용되는 Format 및 Subject 요소의 전체 NameID URI일 수 있습니다. NameID 요소의 Format 속성에 대한 자세한 정보는 [인증 응답을 위한 SAML 어설션 구성 \(p. 181\)](#) 단원을 참조하십시오.

임시 보안 자격 증명이 있으면 AWS API 호출에 사용할 수 있습니다. 이는 AWS가 임시 보안 자격 증명이 유효한지 확인하도록 허용하는 세션 토큰을 포함해야 한다는 점만 제외하면 장기 보안 자격 증명으로 AWS API 호출을 하는 프로세스와 동일합니다.

앱은 자격 증명을 캐싱해야 합니다. 자격 증명은 한 시간 후에 만료되도록 기본 설정되어 있습니다. AWS SDK의 `AmazonSTSCredentialsProvider` 작업을 사용하지 않을 경우, `AssumeRoleWithSAML`을 직접 다시 호출해야 합니다. 이전 자격 증명이 만료되기 전에 이 작업을 호출하여 임시 보안 자격 증명 세트를 새로 받으십시오.

GetFederationToken—사용자 지정 자격 증명 브로커를 통한 연동

`GetFederationToken` API 작업은 연동 사용자에게 일련의 임시 보안 자격 증명을 반환합니다. 이 API는 기본 만료 기간이 상당히 길다는 점이(1시간이 아니라 12시간) `AssumeRole`과 다릅니다. 또한

`DurationSeconds` 파라미터를 사용하여 임시 보안 자격 증명이 유효하게 남아 있을 기간을 지정할 수 있습니다. 결과물로 얻은 자격 증명은 900초(15분)~129,600초(36시간)로 지정된 기간 동안 유효합니다. 만료 기간이 더 길어지면 새 자격 증명을 자주 얻을 필요가 없기 때문에 AWS에 대한 호출 횟수가 줄어들 수 있습니다. 자세한 정보는 [임시 보안 자격 증명 요청하기 \(p. 265\)](#) 단원을 참조하십시오.

연합된 사용자에 대한 임시 보안 자격 증명을 얻기 위한 요청을 생성하는 경우 특정 사용자 자격 증명(IAM 사용자)의 자격 증명을 사용해 요청을 생성해야 합니다. 임시 보안 자격 증명에 대한 권한은 `GetFederationToken`을 호출할 때 전달하는 세션 정책에 의해 결정됩니다. 결과적으로 세션에는 사용자의 자격 증명 기반 정책과 세션 정책, 이 두 정책 모두에 의해 공통적으로 부여되는 권한만 부여됩니다. 역할 세션 권한에 대한 자세한 정보는 [세션 정책 \(p. 307\)](#) 단원을 참조하십시오.

`GetFederationToken` 호출은 보안 토큰, 액세스 키, 보안 키, 만료로 구성된 임시 보안 자격 증명을 반환합니다. 조직 내에서 권한을 관리하고 싶다면 `GetFederationToken`을 사용할 수 있습니다(예: 프록시 애플리케이션을 사용할 권한 할당). `GetFederationToken`을 사용하는 샘플 애플리케이션을 보려면 AWS 샘플 코드 및 라이브러리의 [Active Directory 사용 시 자격 증명 연동 샘플 애플리케이션](#) 단원을 참조하십시오.

다음 예에서는 `GetFederationToken`을 사용한 샘플 요청 및 응답을 보여줍니다. 예를 들어 요청에는 Jean이라는 연동 사용자의 이름이 포함되어 있습니다. `Policy` 파라미터에는 결과물로 얻은 자격 증명이 Amazon S3에만 액세스 권한을 갖도록 지정하는 JSON 문서가 포함되어 있습니다. 그 응답에는 임시 보안 자격 증명뿐만 아니라 연동 사용자 및 자격 증명 만료 시간에 대한 Amazon 리소스 이름(ARN)이 포함되어 있습니다.

Example 요청

```
https://sts.amazonaws.com/
?Version=2011-06-15
&Action=GetFederationToken
&Name=Jean
&Policy=%7B%22Version%22%3A%222012-10-17%22%2C%22Statement%22%3A%5B%7B%22Sid%22%3A
%22Stmt1%22%2C%22Effect%22%3A%22Allow%22%2C%22Action%22%3A%22s3%3A*%22%2C%22Resource%22%3A
%22*%22%7D%5D%7D
&DurationSeconds=1800
&AUTHPARAMS
```

Note

이전 예제의 정책 값은 다음 정책을 URL로 인코딩한 버전입니다.

```
{"Version": "2012-10-17", "Statement": [
    {"Sid": "Stmt1", "Effect": "Allow", "Action": "s3:*", "Resource": "*"}]}
```

또한 이 예에서 `&AUTHPARAMS` 파라미터는 인증 정보의 자리 표시자, 즉 AWS HTTP API 요청에 포함해야 하는 서명입니다. [AWS SDK](#)를 사용하여 API 요청을 생성하는 것이 좋습니다. 이렇게 하면 SDK가 요청 서명을 대신 처리한다는 장점이 있습니다. API 요청을 직접 생성하여 서명해야 할 경우 Amazon Web Services 일반 참조의 [서명 버전 4를 사용하여 AWS 요청에 서명](#)에서 요청에 서명하는 방법 단원을 참조하십시오.

Example 응답

```
<GetFederationTokenResponse xmlns="https://sts.amazonaws.com/doc/2011-06-15/">
<GetFederationTokenResult>
<Credentials>
<SessionToken>
AQoDYXdzEPT//////////wEXAMPLEtc764bNrC9SAPBSM22wDOk4x4HIZ8j4FZTwdQW
LWsKWHGBuFqwAeMicRXmxfpSPfleoIYRqTflfKD8YUuwthAx7mSEI/qkPpKPi/kMcGd
QrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburEDyOKPkyQDYwT7WZ0wq5VSXDvp75YU
9HFv1Rd8Tx6q6fE8YQcHNVXAKiY9q6d+xo0rKwT38xVqr7ZD0u0iPPkUL641IZbqBAz
+scqKmlzm8FDdrypNC9Yjc8fPOLn9FX9KSYvKTr4rvx3ISIlTJabIQwj2ICCEXAMPLE==
</SessionToken>
```

```
<SecretAccessKey>  
wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY  
</SecretAccessKey>  
<Expiration>2011-07-15T23:28:33.359Z</Expiration>  
<AccessKeyId>AKIAIOSFODNN7EXAMPLE;</AccessKeyId>  
</Credentials>  
<FederatedUser>  
  <Arn>arn:aws:sts::123456789012:federated-user/Jean</Arn>  
  <FederatedUserId>123456789012:Jean</FederatedUserId>  
</FederatedUser>  
<PackedPolicySize>6</PackedPolicySize>  
</GetFederationTokenResult>  
<ResponseMetadata>  
  <RequestId>c6104cbe-af31-11e0-8154-cbc7ccf896c7</RequestId>  
</ResponseMetadata>  
</GetFederationTokenResponse>
```

Note

`GetFederationToken`은 세션 정책을 밀집 형식으로 저장합니다. 그 작업은 허용된 최대 크기의 비율로 크기를 반환하므로 호출 파라미터를 조정할 수 있습니다. 정책 크기 제한에 대한 자세한 정보는 AWS Security Token Service API Reference의 [GetFederationToken](#) 단원을 참조하십시오.

리소스 수준에서 권한을 부여하는 것을 선호하면(예: 리소스 기반 정책을 Amazon S3 버킷에 정책 연결), `Policy` 파라미터를 생략할 수 있습니다. 그러나 연동 사용자에 대한 정책을 포함하지 않으면, 임시 보안 자격 증명은 어떤 권한도 부여하지 않을 것입니다. 이 경우 반드시 리소스 정책을 사용해 연동 사용자에게 AWS 리소스에 대한 액세스 권한을 부여해야 합니다.

예를 들어, 나의 AWS 계정 번호가 111122223333이고 Susan이 액세스하도록 허용하려는 Amazon S3 버킷을 내가 가지고 있다고 가정해 보겠습니다. Susan의 임시 보안 자격 증명에는 버킷에 대한 정책은 포함되어 있지 않습니다. 이러한 경우 버킷에 Susan의 ARN과 일치하는 ARN과 관련된 정책이 있는지 확인해야 합니다(예: `arn:aws:sts::111122223333:federated-user/Susan`).

GetSessionToken—신뢰할 수 없는 환경에 있는 사용자를 위한 임시 자격 증명

`GetSessionToken` API 작업은 기존 IAM 사용자에게 일련의 임시 보안 자격 증명을 반환합니다. 예를 들어 MFA가 IAM 사용자에 대해 활성화된 경우에만 AWS 요청을 허용하면 보안을 강화하는 데 유용합니다. 자격 증명은 일시적이므로 모바일 디바이스 또는 웹 브라우저 같은 덜 안전한 환경을 통해 리소스에 액세스하는 IAM 사용자가 있을 때 보안을 강화하는 역할을 합니다. 자세한 정보는 [임시 보안 자격 증명 요청하기](#) (p. 265) 단원 또는 AWS Security Token Service API Reference의 [GetSessionToken](#) 단원을 참조하십시오.

기본적으로 IAM 사용자에 대한 임시 보안 자격 증명은 최대 12시간 동안 유효합니다. 그러나 `DurationSeconds` 파라미터를 사용하여 이 기간을 15분만큼 짧게 또는 36시간만큼 길게 요청할 수 있습니다. 보안상의 이유로 AWS 계정 루트 사용자의 토큰은 1시간의 유효 기간으로 제한됩니다.

`GetSessionToken`은 보안 토큰, 액세스 키 ID 및 보안 액세스 키로 구성된 임시 보안 자격 증명을 반환합니다. 다음 예제에서는 `GetSessionToken`을 사용한 샘플 요청 및 응답을 보여줍니다. 응답에는 임시 보안 자격 증명의 만료 시간도 포함되어 있습니다.

Example 요청

```
https://sts.amazonaws.com/  
?Version=2011-06-15  
&Action=GetSessionToken  
&DurationSeconds=1800
```

&AUTHPARAMS

Note

앞의 예에서 &AUTHPARAMS 파라미터는 인증 정보의 자리 표시자, 즉 AWS HTTP API 요청에 포함해야 하는 서명입니다. [AWS SDK](#)를 사용하여 API 요청을 생성하는 것이 좋습니다. 이렇게 하면 SDK가 요청 서명을 대신 처리한다는 장점이 있습니다. API 요청을 직접 생성하여 서명해야 할 경우 Amazon Web Services 일반 참조의 [서명 버전 4](#)를 사용하여 AWS 요청에 서명하는 방법 단원을 참조하십시오.

Example 응답

```
<GetSessionTokenResponse xmlns="https://sts.amazonaws.com/doc/2011-06-15/">
<GetSessionTokenResult>
<Credentials>
<SessionToken>
AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT+FvwqnKwRcOIfRh3c/L
To6UDdyJwO0vEVPVLXCr0rUtdnniCEXAMPLE/IvU1dYUg2RVAJBanLiHb4IgRmpRV3z
rkuWJ0gQs8IZZaIv2BXIa2R4OlglBN9bkUDNCJiBeb/Ax1zBBko7b15fjrBs2+cTQtp
Z3CYWFVG8C5zqx37wnOE49mRl/+OtkIKGO7fAE
</SessionToken>
<SecretAccessKey>
wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY
</SecretAccessKey>
<Expiration>2011-07-11T19:55:29.611Z</Expiration>
<AccessKeyId>AKIAIOSFODNN7EXAMPLE</AccessKeyId>
</Credentials>
</GetSessionTokenResult>
<ResponseMetadata>
<RequestId>58c5dbae-abef-11e0-8cfe-09039844ac7d</RequestId>
</ResponseMetadata>
</GetSessionTokenResponse>
```

선택 사항으로 GetSessionToken 요청은 AWS 멀티 팩터 인증(MFA) 확인에 대한 `SerialNumber` 및 `TokenCode` 값을 포함할 수 있습니다. 제공한 값이 유효하면 AWS STS에서는 MFA 인증 상태가 포함된 임시 보안 자격 증명을 제공합니다. 그런 다음 임시 보안 자격 증명은 MFA 인증이 유효한 동안 MFA로 보호되는 API 작업 또는 AWS 웹 사이트에 액세스하는 데 사용할 수 있습니다.

다음 예는 MFA 확인 코드 및 디바이스 일련 번호를 포함하는 GetSessionToken 요청을 보여줍니다.

```
https://sts.amazonaws.com/
?Version=2011-06-15
&Action=GetSessionToken
&DurationSeconds=7200
&SerialNumber=YourMFADeviceSerialNumber
&TokenCode=123456
&AUTHPARAMS
```

Note

AWS STS에 대한 호출은 전역적 엔드포인트 또는 AWS가 활성화된 리전 엔드포인트 어느 곳으로도 이루어질 수 있습니다. 자세한 정보는 [리전 및 엔드포인트의 AWS STS 단원을 참조하십시오](#). 또한 앞의 예에서 &AUTHPARAMS 파라미터는 인증 정보의 자리 표시자, 즉 AWS HTTP API 요청에 포함해야 하는 서명입니다. [AWS SDK](#)를 사용하여 API 요청을 생성하는 것이 좋습니다. 이렇게 하면 SDK가 요청 서명을 대신 처리한다는 장점이 있습니다. API 요청을 직접 생성하여 서명해야 할 경우 Amazon Web Services 일반 참조의 [서명 버전 4](#)를 사용하여 AWS 요청에 서명하는 방법 단원을 참조하십시오.

AWS STS API 작업 비교

다음 표는 임시 보안 자격 증명을 반환하는 AWS STS의 API 작업이 수행하는 기능을 비교해 보여줍니다. 역할을 수임해 임시 보안 자격 증명을 요청하는 데 사용할 수 있는 여러 방법을 알아보려면 [IAM 역할 사용 \(p. 227\)](#) 단원을 참조하십시오.

API 옵션 비교

AWS STS API	호출할 수 있는 사용자	자격 증명의 수명 (최소 최대 기본)	MFA 지원 ¹	세션 정책 지원 ²	결과로 얻은 임시 자격 증명에 대한 제한
AssumeRole	IAM 사용자 또는 기존 임시 보안 자격 증명이 있는 사용자	15분 최대 세션 기간 설정 ³ 1시간	예	예	<code>GetFederationToken</code> 또는 <code>GetSessionToken</code> 호출 불가
AssumeRoleWithSAML	이전 사용자나 호출자도 잘 알려진 자격 증명 공급자의 인증을 나타내는 SAML 인증 응답을 반드시 전달해야 합니다.	15분 최대 세션 기간 설정 ³ 1시간	아니요	예	<code>GetFederationToken</code> 또는 <code>GetSessionToken</code> 호출 불가
AssumeRoleWithWebIdentity	이전 사용자나 호출자도 잘 알려진 자격 증명 공급자의 인증을 나타내는 웹 자격 인증 토큰을 반드시 전달해야 합니다.	15분 최대 세션 기간 설정 ³ 1시간	아니요	예	<code>GetFederationToken</code> 또는 <code>GetSessionToken</code> 호출 불가
GetFederationToken	IAM 사용자 또는 AWS 계정 루트 사용자	IAM 사용자: 15분 36시간 12시간 루트 사용자: 15분 1시간 1시간	아니요	예	IAM API 작업 직접 호출 불가 <code>GetCallerIdentity</code> 를 제외한 AWS STS API 작업을 호출할 수 없습니다. 콘솔로의 SSO가 허용됩니다. ⁴
GetSessionToken	IAM 사용자 또는 루트 사용자	IAM 사용자: 15분 36시간 12시간 루트 사용자: 15분 1시간 1시간	예	아니요	요청으로 MFA 정보가 포함되지 않으면 IAM API 작업 호출 불가 <code>AssumeRole</code> 또는 <code>GetCallerIdentity</code> 를 제외한 AWS STS API 작업 호출 불가 콘솔로의 SSO는 허용되지 않습니다. ⁵

¹ MFA 지원. `AssumeRole` 및 `GetSessionToken` API 작업을 호출할 때 멀티 팩터 인증(MFA)에 대한 정보를 포함시킬 수 있습니다. 이는 API 호출의 결과물인 임시 보안 자격 증명을 MFA 디바이스로 인증된 사용자들만 사용할 수 있게 해줍니다. 자세한 정보는 [MFA 보호 API 액세스 구성 \(p. 122\)](#) 단원을 참조하십시오.

² 세션 정책 지원. 세션 정책은 역할 또는 연합된 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 정책입니다. 이 정책은 세션에 할당된 역할/사용자 자격 증명 기반 정책의 권한을 제한

합니다. 결과적으로 세션에는 역할의 자격 증명 기반 정책과 세션 정책, 이 두 정책 모두에 의해 부여되는 권한만 부여됩니다. 이 정책은 위임된 역할에 허용된 액세스 권한을 넘어서도록 권한을 승격하는데 사용될 수 없다는 것에 유의하십시오. 역할 세션 권한에 대한 자세한 정보는 [세션 정책 \(p. 307\)](#) 단원을 참조하십시오.

³ 최대 세션 기간 설정. DurationSeconds 파라미터를 사용하여 역할 세션 기간을 900초(15초)에서 해당 역할에 대한 최대 세션 기간 설정까지 지정합니다. 역할에 대한 최댓값을 확인하는 방법을 알아보려면 [역할에 대한 최대 세션 기간 설정 보기 \(p. 228\)](#) 단원을 참조하십시오.

⁴ 콘솔로 SSO(Single Sign-On)하기 SSO를 지원하기 위해 AWS는 페더레이션 엔드포인트(<https://signin.aws.amazon.com/federation>)를 호출해 임시 보안 자격 증명을 전달할 수 있게 해줍니다. 엔드포인트는 암호 없이도 사용자를 콘솔에 바로 로그인시켜주는 URL을 구성하는데 사용 가능한 토큰을 반환합니다. 자세한 정보는 AWS 보안 블로그의 [SAML 2.0 연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하기 \(p. 185\)](#) 및 [AWS Management Console에 대한 교차 계정 액세스를 가능하게 하는 방법](#) 단원을 참조하십시오.

⁵ 임시 자격 증명을 검색한 이후에 연동 SSO 엔드포인트로 자격 증명을 전달하여 AWS Management 콘솔에 액세스할 수 있습니다. 자세한 정보는 [연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하는 URL 생성\(사용자 지정 연동 브로커\) \(p. 188\)](#) 단원을 참조하십시오.

임시 보안 자격 증명을 사용해 AWS 리소스에 대한 액세스 요청하기

IAM 사용자 자격 증명과 같은 장기 보안 자격 증명을 사용하는 것과 똑같은 방식으로 임시 보안 자격 증명을 사용해 [AWS SDK](#) 또는 API 호출로 AWS 리소스를 프로그래밍 방식으로 요청할 수 있습니다. 그러나 몇 가지 차이점이 있습니다.

- 임시 보안 자격 증명을 사용해 호출할 경우 그 호출에 반드시 세션 토큰이 포함되어야 하는데, 이 세션 토큰은 임시 자격 증명과 함께 반환됩니다. AWS는 세션 토큰을 사용해 임시 보안 자격 증명의 유효성을 검증합니다.
- 임시 자격 증명은 지정된 간격 후에 만료됩니다. 자격 증명이 만료된 후에는 그 자격 증명을 사용한 어떤 요청도 실패할 것이므로 일련의 새로운 자격 증명을 얻어야 합니다.

[AWS SDK](#), [AWS Command Line Interface\(AWS CLI\)](#) 또는 [Windows PowerShell용 도구](#)를 사용하고 있다면 임시 보안 자격 증명을 얻고 사용하는 방식은 컨텍스트에 따라 달라집니다. EC2 인스턴스 내부에서 코드, AWS CLI 또는 Windows PowerShell용 도구 명령을 실행 중이라면 Amazon EC2에 대한 역할을 이용할 수 있습니다. 그렇지 않은 경우 [AWS STS API](#)를 호출해 임시 자격 증명을 얻은 다음, 그 자격 증명을 사용해 AWS 서비스를 명시적으로 호출할 수 있습니다.

Note

AWS Security Token Service(AWS STS)를 사용하면 AWS 리소스에 대한 액세스를 제어할 수 있는 임시 보안 자격 증명을 생성하여 신뢰받는 사용자에게 제공할 수 있습니다. AWS STS에 대한 자세한 정보는 [임시 보안 자격 증명 \(p. 263\)](#) 단원을 참조하십시오. AWS STS는 <https://sts.amazonaws.com>에 기본 엔드포인트가 있는 전역적 서비스입니다. 이 엔드포인트 및 다른 엔드포인트로부터 얻는 자격 증명이 전역적으로 유효하고 어떤 리전의 서비스 및 리소스에서 유효하다 해도, 이 엔드포인트는 미국 동부(오하이오) 리전에 있습니다. 지원되는 리전에서 엔드포인트에 대한 AWS STS API 호출을 할 수도 있습니다. 이렇게 지리적으로 더 가까운 리전에 있는 서버에서 요청함으로써 지연 시간을 단축할 수 있습니다. 자격 증명은 어떤 리전에서 오는지 상관없이 전역적으로 유효합니다. 자세한 내용은 [AWS 리전에서 AWS STS 활성화 및 비활성화 \(p. 287\)](#) 단원을 참조하십시오.

목차

- [Amazon EC2 인스턴스에서 임시 자격 증명 사용하기 \(p. 275\)](#)
- [AWS SDK에서 임시 보안 자격 증명 사용하기 \(p. 275\)](#)
- [AWS CLI에서 임시 보안 자격 증명 사용하기 \(p. 275\)](#)

- API 작업을 통해 임시 보안 자격 증명 사용 (p. 276)
- 추가 정보 (p. 276)

Amazon EC2 인스턴스에서 임시 자격 증명 사용하기

EC2 인스턴스 내에서 AWS CLI 명령 또는 코드를 실행하고자 하는 경우 자격 증명을 얻는 바람직한 방법은 Amazon EC2에 대한 역할을 사용하는 것입니다. EC2 인스턴스 상에서 실행되는 애플리케이션에 부여하고 싶은 권한을 지정하는 IAM 역할을 생성합니다. 인스턴스를 시작할 때 그 역할을 인스턴스에 연결합니다.

인스턴스 상에서 실행되는 애플리케이션, AWS CLI 및 Windows PowerShell용 도구 명령은 인스턴스 메타데이터로부터 자동 임시 보안 자격 증명을 얻을 수 있습니다. 임시 보안 자격 증명을 명시적으로 얻지 않아도 됩니다. AWS SDK, AWS CLI 및 Windows PowerShell용 도구이 EC2 인스턴스 메타데이터 서비스로부터 자격 증명을 자동으로 얻어 그것을 사용하기 때문입니다. 임시 자격 증명은 그 인스턴스에 연결된 역할에 대해 정의한 권한이 있습니다.

자세한 내용 및 예시는 다음을 참조하십시오.

- IAM 역할을 사용하여 Amazon Elastic Compute Cloud의 AWS 리소스에 대한 액세스 권한 부여 — AWS SDK for Java
- IAM 역할을 사용하여 액세스 권한 부여 — .NET용 AWS SDK
- 역할 생성 — Ruby용 AWS SDK

AWS SDK에서 임시 보안 자격 증명 사용하기

코드의 임시 보안 자격 증명을 사용하려면 `AssumeRole`과 같은 AWS STS API를 프로그래밍 방식으로 호출하고 그 결과 얻은 자격 증명 및 세션 토큰을 추출한 다음, 그 값을 AWS에 대한 후속 호출을 위한 자격 증명으로 사용하면 됩니다. 다음 예는 AWS SDK를 사용할 경우 임시 보안 자격 증명을 사용하는 방법에 대한 유사 코드를 보여줍니다.

```
assumeRoleResult = AssumeRole(roleArn);
tempCredentials = new SessionAWSCredentials(
    assumeRoleResult.AccessKeyId,
    assumeRoleResult.SecretAccessKey,
    assumeRoleResult.SessionToken);
s3Request = CreateAmazonS3Client(tempCredentials);
```

`AssumeRole`을 호출하여 임시 보안 자격 증명을 얻은 다음 그 자격 증명을 사용해 AWS SDK for Python (Boto)를 호출하는 방법을 보여주는 Python(Amazon S3 사용)으로 작성된 예를 보려면 [IAM 역할\(AWS API\)로 전환하기 \(p. 238\)](#)을 참조하십시오.

`AssumeRole`, `GetFederationToken` 및 기타 API 작업을 호출하는 방법에 대한 자세한 내용은 [AWS Security Token Service API Reference](#)를 참조하십시오. 이러한 호출의 결과에서 임시 보안 자격 증명 및 세션 토큰을 얻는 방법에 대한 자세한 내용은 사용하고 있는 SDK의 설명서를 참조하십시오. SDK 및 도구 키트 섹션의 [AWS 설명서 메인 페이지](#)에서 모든 AWS SDK의 설명서를 검색할 수 있습니다.

이전 자격 증명이 만료되기 전에 반드시 새로운 일련의 자격 증명을 얻도록 해야 합니다. 일부 SDK에서는 자격 증명 갱신 프로세스를 관리해주는 공급자를 사용할 수 있습니다. 사용하고 있는 SDK의 설명서를 확인하십시오.

AWS CLI에서 임시 보안 자격 증명 사용하기

AWS CLI에서 임시 보안 자격 증명을 사용할 수 있습니다. 이 임시 보안 자격 증명은 정책을 테스트하는 데 유용합니다.

AWS CLI를 사용해 `AssumeRole` 또는 `GetFederationToken`과 같은 AWS STS API를 호출한 다음, 그 결과물로 얻은 출력을 캡처할 수 있습니다. 다음 예는 파일에 출력을 전송하는 `AssumeRole`에 대한 호출을 보

여줍니다. 예시에서 `profile` 파라미터는 AWS CLI 구성 파일에 있는 프로필이라고 가정되고, 아울러 역할 수임 권한을 지닌 IAM 사용자의 자격 증명을 참조한다고 가정됩니다.

```
$ aws sts assume-role --role-arn arn:aws:iam::123456789012:role/role-name --role-session-name "RoleSession1" --profile IAM-user-name > assume-role-output.txt
```

명령이 끝나면 액세스 키 ID, 보안 액세스 키 및 세션 토큰을 해당 명령을 라우팅한 곳이 어디든지 간에 그 곳에서 수동으로 또는 스크립트를 사용해 추출할 수 있습니다. 그런 다음 이 값을 환경 변수에 할당할 수 있습니다.

AWS CLI 명령을 실행한다면 AWS CLI가 환경 변수를 먼저 검색하고 다음 순서로 자격 증명을 검색하는 특정 순서로 구성 파일을 검색합니다. 따라서 임시 자격 증명을 환경 변수에 넣은 후에 AWS CLI는 그 자격 증명을 기본 값으로 사용합니다. (명령에 `profile` 파라미터를 지정한다면 AWS CLI는 환경 변수를 건너 뛰어 구성 파일에서 검색합니다. 이로써 필요한 경우 환경 변수에서 자격 증명을 무시할 수 있게 됩니다.)

다음 예는 임시 보안 자격 증명에 대한 환경 변수를 설정한 다음, AWS CLI 명령을 호출하는 방법을 보여줍니다. `profile` 파라미터는 AWS CLI 명령에 포함되어 있지 않기 때문에 AWS CLI는 먼저 환경 변수에서 자격 증명을 검색하고, 따라서 임시 자격 증명을 사용합니다.

Linux

```
$ export AWS_ACCESS_KEY_ID=AKIAI44QH8DHBEXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
$ export AWS_SESSION_TOKEN=AQoDYXdzEJr...<remainder of security token>
$ aws ec2 describe-instances --region us-west-1
```

Windows

```
C:\> SET AWS_ACCESS_KEY_ID=AKIAI44QH8DHBEXAMPLE
C:\> SET AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
C:\> SET AWS_SESSION_TOKEN=AQoDYXdzEJr...<remainder of token>
C:\> aws ec2 describe-instances --region us-west-1
```

API 작업을 통해 임시 보안 자격 증명 사용

AWS로 직접 HTTPS API 요청을 하는 경우, AWS Security Token Service(AWS STS)에서 가져오는 임시 보안 자격 증명으로 그러한 요청에 서명할 수 있습니다. 이를 위해서는 장기 자격 증명을 사용하는 것과 동일한 방식으로 AWS STS로부터 받는 액세스 키 ID 및 보안 액세스 키를 사용하여 요청에 서명합니다. 또한 AWS STS로부터 받는 세션 토큰을 API 요청에 추가합니다. 그 세션 토큰을 HTTP 헤더 또는 X-Amz-Security-Token이라는 쿼리 문자열 파라미터에 추가합니다. 그 세션 토큰을 HTTP 헤더 또는 쿼리 문자열 파라미터에 추가하되, 하나에만 추가해야 합니다. HTTPS API 요청에 서명하는 방법에 대한 자세한 내용은 AWS General Reference의 [AWS API 요청 서명](#)을 참조하십시오.

추가 정보

다른 AWS 서비스와 함께 AWS STS를 사용하는 방법에 대한 자세한 내용은 다음 링크를 참조하십시오.

- Amazon S3. [Amazon Simple Storage Service 개발자 가이드](#)의 [Making Requests Using IAM User Temporary Credentials](#) 또는 [Making Requests Using Federated User Temporary Credentials](#)를 참조하십시오.
- Amazon SNS. Amazon Simple Notification Service 개발자 안내서의 [Using Temporary Security Credentials](#)를 참조하십시오.
- Amazon SQS. Amazon Simple Queue Service 개발자 안내서의 [Using Temporary Security Credentials](#)를 참조하십시오.
- Amazon SimpleDB. Amazon SimpleDB 개발자 안내서의 [Using Temporary Security Credentials](#)를 참조하십시오.

사용자 임시 보안 자격 증명에 대한 권한 제어

AWS Security Token Service(AWS STS)를 사용하면 AWS 리소스에 대한 액세스를 제어할 수 있는 임시 보안 자격 증명을 생성하여 신뢰받는 사용자에게 제공할 수 있습니다. AWS STS에 대한 자세한 내용은 [임시 보안 자격 증명 \(p. 263\)](#) 단원을 참조하십시오. 임시 보안 자격 증명은 AWS STS가 발급한 후 만료 기간 동안 유효하며 취소될 수 없습니다. 그러나 임시 보안 자격 증명에 할당된 권한은 자격 증명을 사용해 요청이 이루어질 때마다 평가되기 때문에 자격 증명이 발급된 이후에라도 액세스 권한을 변경함으로써 자격 증명 취소 효과를 얻을 수 있습니다.

다음 주제는 독자가 AWS 권한 및 정책에 대한 유효한 지식이 있다고 가정합니다. 이 주제에 대한 자세한 내용은 [액세스 관리 \(p. 304\)](#) 단원을 참조하십시오.

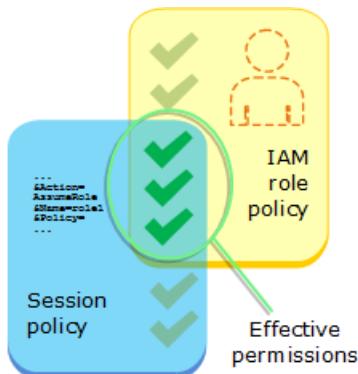
주제

- [AssumeRole, AssumeRoleWithSAML 및 AssumeRoleWithWebIdentity에 대한 권한 \(p. 277\)](#)
- [GetFederationToken에 대한 권한 \(p. 279\)](#)
- [GetSessionToken에 대한 권한 \(p. 282\)](#)
- [임시 보안 자격 증명에 대한 권한 비활성화 \(p. 283\)](#)
- [임시 보안 자격 증명을 생성할 수 있는 권한 부여 \(p. 286\)](#)

AssumeRole, AssumeRoleWithSAML 및 AssumeRoleWithWebIdentity에 대한 권한

수임된 역할에 대한 권한 정책은 [AssumeRole](#), [AssumeRoleWithSAML](#) 및 [AssumeRoleWithWebIdentity](#)에 의해 반환되는 임시 보안 자격 증명에 대한 권한을 결정합니다. 역할을 생성 또는 업데이트할 때 이러한 권한을 정의합니다.

원활 경우 [AssumeRole](#), [AssumeRoleWithSAML](#) 또는 [AssumeRoleWithWebIdentity](#) API 작업의 파라미터로 JSON 정책을 전달할 수 있습니다. 이 세션 정책은 역할의 임시 자격 증명 세션에 대한 권한을 제한합니다. 후속 AWS API 호출 시에도 역할의 임시 자격 증명을 사용하여 역할이 속한 계정의 리소스에 액세스할 수 있습니다. 세션 정책을 사용하여 수임된 역할의 자격 증명 기반 정책에서 허용되는 권한을 부여할 수는 없습니다. 이 역할의 효과적인 권한을 AWS가 어떻게 결정하는지 자세히 알아보려면 [정책 평가 로직 \(p. 531\)](#) 단원을 참조하십시오.



'허용' 또는 '거부' 권한 부여 결정을 내릴 때 [AssumeRole](#)에 대한 원래 호출을 생성한 자격 증명에 연결된 정책은 AWS에서 평가하지 않습니다. 해당 사용자는 맙은 역할에 의해 할당된 권한을 위해 자신의 원래 권한을 일시적으로 포기합니다. [AssumeRoleWithSAML](#) 및 [AssumeRoleWithWebIdentity](#) API 작업의 경우 API 호출자가 AWS 자격 증명이 아니기 때문에 평가할 정책이 없습니다.

예: AssumeRole을 사용한 권한 할당

서로 다른 종류의 정책으로 [AssumeRole](#) API 작업을 사용할 수 있습니다. 여기 몇 가지 예가 있습니다.

역할 권한 정책

이 예에서는 선택 사항인 Policy 파라미터에 세션 정책을 지정하지 않고 AssumeRole API 작업을 호출합니다. 임시 자격 증명에 할당된 권한은 위임된 역할의 권한 정책에 따라 결정됩니다. 다음 예제 권한 정책은 S3 버킷 productionapp에 포함된 객체를 모두 나열하도록 역할 권한을 부여합니다. 또한 해당 역할이 이 버킷 내에서 객체를 가져오고, 배치하고, 삭제하도록 허용합니다.

Example 역할 권한 정책

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "s3>ListBucket",  
            "Resource": "arn:aws:s3:::productionapp"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetObject",  
                "s3:PutObject",  
                "s3>DeleteObject"  
            ],  
            "Resource": "arn:aws:s3:::productionapp/*"  
        }  
    ]  
}
```

파라미터로 전달되는 세션 정책

사용자에게 이전 예제와 동일한 역할을 수임하도록 허용하려 한다고 가정해 보겠습니다. 하지만 이 경우 역할 세션에 대해 productionapp S3 버킷에/에서 객체를 넣거나 가져오는 작업만을 허용하는 권한을 부여하고자 합니다. 객체를 삭제할 수 없도록 하고자 합니다. 이렇게 하기 위한 한 가지 방법은 새 역할을 만들어 그 역할의 권한 정책에 원하는 권한을 지정하는 것입니다. 또 다른 방법은 AssumeRole API를 호출하여 선택 사항인 Policy 파라미터의 권한 세션 정책을 API 작업의 일부로 포함하는 것입니다. 결과적으로 세션에는 역할의 자격 증명 기반 정책과 세션 정책, 이 두 정책 모두에 의해 부여되는 권한만 부여됩니다. 이 정책은 위임된 역할에 허용된 액세스 권한을 넘어서도록 권한을 승격하는데 사용될 수 없다는 것에 유의하십시오. 역할 세션 권한에 대한 자세한 정보는 [세션 정책 \(p. 307\)](#) 단원을 참조하십시오.

예를 들어 다음 정책이 API 호출의 파라미터로 전달된다고 가정합시다. 세션을 사용하는 사람에게는 다음 작업에 대한 수행 권한만 부여됩니다.

- productionapp 버킷에 있는 모든 객체의 목록을 조회합니다.
- 객체를 가져와 productionapp 버킷에 넣습니다.

다음 세션 정책에서는 s3>DeleteObject 권한이 필터링되어 위임된 세션에 s3>DeleteObject 권한이 부여되지 않습니다. 이 정책은 역할 세션에 대한 최대 권한을 설정하여 역할에 대한 기존 권한 정책을 재정의 합니다.

Example AssumeRole API 호출로 전달된 세션 정책

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "s3>ListBucket",  
            "Resource": "arn:aws:s3:::productionapp"  
        }  
    ]  
}
```

```
},
{
  "Effect": "Allow",
  "Action": [
    "s3:GetObject",
    "s3:PutObject"
  ],
  "Resource": "arn:aws:s3:::productionapp/*"
}
]
```

리소스 기반 정책

일부 AWS 리소스는 리소스 기반 정책을 지원하고 이 정책은 임시 보안 자격 증명에 영향을 미치는 권한을 정의하는 또 다른 메커니즘을 제공합니다. Amazon S3 버킷, Amazon SNS 주제, Amazon SQS 대기열 같은 몇몇 리소스만이 리소스 기반 정책을 지원합니다. 다음 예는 앞의 예들을 확장한 것으로서 productionapp이라는 S3 버킷을 사용합니다. 다음 정책은 버킷에 연결되어 있습니다.

다음 리소스 기반 정책을 productionapp 버킷에 연결할 때 모든 사용자들은 버킷에서 객체를 삭제할 권한을 거부당합니다(정책의 Principal 요소에 유의하십시오). 역할 권한 정책이 DeleteObject 권한을 부여한다 해도 여기에는 모든 수임된 역할 사용자들이 포함됩니다. 명시적인 Deny 문은 항상 Allow 문보다 우선 적용됩니다.

Example 버킷 정책

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Principal": {"AWS": "*"},
    "Effect": "Deny",
    "Action": "s3:DeleteObject",
    "Resource": "arn:aws:s3:::productionapp/*"
  }
}
```

다수의 정책 유형이 AWS에 의해 어떻게 결합되고 평가되는지에 대한 자세한 정보는 [정책 평가 로직 \(p. 531\)](#) 단원을 참조하십시오.

GetFederationToken에 대한 권한

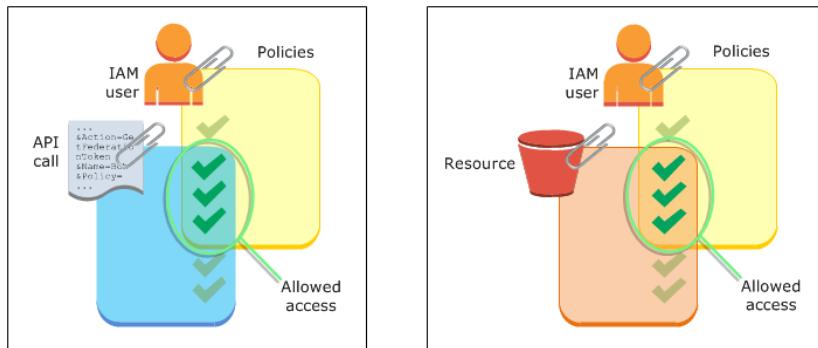
세션 정책은 연합된 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 연합된 사용자 세션을 생성하려면 IAM 사용자의 액세스 키를 사용하여 GetFederationToken API 작업을 프로그래밍 방식으로 호출합니다. 이 작업을 수행하고 세션 정책을 전달할 경우 결과적으로 세션에는 IAM 사용자 자격 증명 기반 정책과 세션 정책, 이 두 정책 모두에 의해 부여되는 권한만 부여됩니다.

연동 사용자에게 할당된 권한은 둘 중 한 곳에 정의되어 있습니다.

- GetFederationToken API 호출의 파라미터로 전달되는 세션 정책. (가장 일반적)
- 정책의 Principal 요소에서 연동 사용자를 명시적으로 호명하는 리소스 기반 정책. (일반적이지 않음)

이는 대부분의 경우 GetFederationToken API 호출로 정책을 전달하지 않으면 그 결과 얻게 되는 임시 보안 자격 증명은 아무 권한이 없다는 것을 뜻합니다. 유일한 예외는 정책의 Principal 요소에서 연합된 사용자 세션을 특별히 참조하는 리소스 기반 정책을 지닌 리소스에 액세스하는 데 자격 증명이 사용될 때입니다.

다음 그림은 GetFederationToken 호출에 의해 반환되는 임시 보안 자격 증명에 대한 권한을 정책들이 어떻게 상호작용해 결정하는지를 시각적으로 재현한 것입니다.



예: GetFederationToken을 사용한 권한 할당

서로 다른 종류의 정책으로 GetFederationToken API 작업을 사용할 수 있습니다. 여기 몇 가지 예가 있습니다.

IAM 사용자에게 연결된 정책

이 예시에는 2개의 백엔드 웹 서비스에 의존하는 브라우저 기반 클라이언트 애플리케이션이 있습니다. 백엔드 서비스 중 하나는 자신만의 인증 서버로서 고유한 자격 증명 시스템을 사용해 클라이언트 애플리케이션을 인증합니다. 다른 백엔드 서비스는 AWS 서비스로, 클라이언트 애플리케이션의 기능 중 일부를 제공합니다. 이 클라이언트 애플리케이션은 서버에 의해 인증되고, 서버는 적절한 권한 정책을 생성하거나 가져옵니다. 서버는 이제 GetFederationToken API를 호출해 임시 보안 자격 증명을 얻은 다음, 그 자격 증명을 클라이언트 애플리케이션에 반환합니다. 이제 클라이언트 애플리케이션은 임시 보안 자격 증명을 사용해 AWS 서비스에 직접 요청할 수 있게 됩니다. 이 아키텍처는 클라이언트 애플리케이션이 장기 AWS 자격 증명을 포함하지 않고도 AWS 요청을 할 수 있도록 허용합니다.

인증 서버는 token-app라는 IAM 사용자의 장기 보안 자격 증명을 사용해 GetFederationToken API를 호출하지만, 장기 IAM 사용자 자격 증명은 서버에 남고 클라이언트에게는 결코 배포되지 않습니다. 다음 예시 정책은 token-app IAM 사용자에게 연결되어 연동 사용자(클라이언트)에게 필요한 가장 폭넓은 권한 집합을 정의합니다. sts:GetFederationToken 권한은 인증 서비스가 연동 사용자에 대한 임시 보안 자격 증명을 얻는 데 필요하다는 점에 유의하십시오.

Note

AWS는 샘플 Java 애플리케이션을 제공함으로써 이 목적에 기여하는데, Java 애플리케이션은 [자격 증명 등록을 위한 토큰 벤딩 머신 - 샘플 Java 웹 애플리케이션](#)에서 다운로드할 수 있습니다.

Example GetFederationToken을 호출하는 IAM 사용자 token-app에 연결된 정책

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "sts:GetFederationToken",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "dynamodb:*",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "sns:*",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "lambda:*",  
            "Resource": "*"  
        }  
    ]  
}
```

```
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": "*"
    },
{
    "Effect": "Allow",
    "Action": "sns:*",
    "Resource": "*"
}
]
```

이전 정책이 몇 가지 권한을 부여한다 해도 그 자체로는 연동 사용자에게 어떤 권한을 부여하기에 충분하지 않습니다. 앞의 정책에 정의된 권한을 지닌 IAM 사용자가 `GetFederationToken`을 호출하고 정책을 API 호출의 파라미터로 전달하지 않는다면, 그 결과로 얻은 연동 사용자에게는 유효한 권한이 없습니다.

파라미터로 전달되는 세션 정책

연동 사용자에게 적절한 권한이 할당되도록 하는 가장 일반적인 방법은 세션 정책을 `GetFederationToken` API 호출의 파라미터로 전달하는 것입니다. 앞의 예시를 확장해 `GetFederationToken`이 IAM 사용자 `token-app`의 자격 증명으로 호출되고 그 다음 세션 정책은 API 호출의 파라미터로 전달된다고 가정해봅시다. 연동 사용자는 다음 작업들만 수행할 수 있는 권한을 갖게 됩니다.

- `productionapp`이라는 Amazon S3 버킷의 컨텐츠 나열
- `productionapp` 버킷의 항목들에 대한 Amazon S3 `GetObject`, `PutObject` 및 `DeleteObject` 작업 수행

권한이 두 사용자에게 부여되었으므로 연합된 사용자에게 이러한 권한이 할당됩니다.

- `GetFederationToken`을 호출한 IAM 사용자(IAM 사용자에게 연결된 정책을 통해)
- 연합된 사용자(세션 정책을 통해)

연합된 사용자는 Amazon SNS, Amazon SQS, Amazon DynamoDB 또는 S3 버킷(`productionapp` 제외)에서 작업을 수행할 수 없습니다. 이러한 작업은 관련 권한이 `GetFederationToken` 호출과 연결된 IAM 사용자에게 부여되었더라도 거부됩니다. 그 이유는 연합 사용자의 유효 권한이 IAM 사용자 정책 및 세션 정책 모두에 부여된 권한으로만 구성되기 때문입니다.

Example `GetFederationToken` API 호출의 파라미터로 전달되는 세션

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["s3>ListBucket"],
            "Resource": ["arn:aws:s3:::productionapp"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": ["arn:aws:s3:::productionapp/*"]
        }
    ]
}
```

리소스 기반 정책

일부 AWS 리소스는 리소스 기반 정책을 지원하고, 이 정책은 연동 사용자에게 직접 권한을 부여하는 또 다른 메커니즘을 제공합니다. 일부 AWS 서비스만이 리소스 기반 정책을 지원합니다. 예를 들어, Amazon S3의 경우 버킷, Amazon SNS의 경우 주제, Amazon SQS의 경우 대기열에 정책을 연결할 수 있습니다. 리소스 기반 정책을 지원하는 모든 서비스 목록은 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 및 표의 "리소스 기반 정책" 열을 참조하십시오. 이 서비스 중 하나를 사용하고 리소스 기반 정책이 시나리오에서 통한다면, 리소스 기반 정책의 Principal 요소에서 연동 사용자의 Amazon 리소스 이름을 지정해 연동 사용자에게 직접 권한을 할당합니다. 다음 예는 이를 보여줍니다. 다음 예는 앞의 예들을 확장한 것으로서 productionapp이라는 S3 버킷을 사용합니다.

다음 리소스 기반 정책은 버킷에 연결되어 있습니다. 이 버킷 정책은 Carol이라는 연합된 사용자가 버킷에 액세스할 수 있도록 허용합니다. 다음 리소스 기반 정책이 적용되고 앞서 기술된 예시 정책이 token-app IAM 사용자에 연결되어 있으면, Carol이라는 연합된 사용자는 productionapp이라는 버킷에 대해 s3:GetObject, s3:PutObject 및 s3:DeleteObject 작업을 수행할 수 있는 권한이 있습니다. 이는 GetFederationToken API 호출의 파라미터로 전달되는 세션 정책이 없을 때에도 해당됩니다. 왜냐하면 이 경우에 Carol이라는 연동 사용자는 다음 리소스 기반 정책에 의해 명시적으로 권한을 부여받았기 때문입니다.

그 권한이 IAM 사용자 및 연동 사용자 둘 다에게 명시적으로 부여될 때만 연동 사용자는 권한을 부여받는다는 것을 명심하십시오. GetFederationToken API 호출의 파라미터로 전달되는 세션 정책을 통해 연합된 사용자에게 권한을 부여할 수 있습니다. 다음 예제에서처럼 정책의 Principal 요소에서 연합된 사용자의 이름을 명시적으로 지정하는 리소스 기반 정책을 통해서도 권한을 부여할 수 있습니다.

Example 연동 사용자에 대한 액세스를 허용하는 버킷 정책

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Principal": {"AWS": "arn:aws:sts::ACCOUNT-ID-WITHOUT-HYPHENS:federated-user/Carol"},  
         "Effect": "Allow",  
         "Action": [  
             "s3:GetObject",  
             "s3:PutObject",  
             "s3:DeleteObject"  
         ],  
         "Resource": ["arn:aws:s3:::productionapp/*"]  
    }  
}
```

GetSessionToken에 대한 권한

GetSessionToken API 작업 또는 get-session-token CLI 명령을 호출해야 하는 기본적인 경우는 사용자가 멀티 팩터 인증(MFA)으로 인증되어야 할 때입니다. MFA로 인증된 사용자가 요청하는 경우에 한해 특정 작업들을 허용하는 정책을 작성하는 것도 가능합니다. MFA 권한 부여 확인을 성공적으로 통과하려면 사용자는 먼저 GetSessionToken을 호출하여 선택 사항인 SerialNumber 및 TokenCode 파라미터를 포함해야 합니다. 사용자가 MFA 디바이스를 통해 인증을 받으면 GetSessionToken API 작업에서 반환하는 자격 증명에는 MFA 컨텍스트가 포함됩니다. 이 컨텍스트에서는 사용자가 MFA 디바이스를 통해 인증을 받았고 MFA 인증이 필요한 API 작업에 대한 권한이 있음을 표시합니다.

GetSessionToken에 필요한 권한

사용자는 권한이 없어도 세션 토큰을 얻을 수 있습니다. GetSessionToken 작업의 목적은 MFA를 사용하는 사용자를 인증하는 것입니다. 정책을 사용하여 인증 작업을 제어할 수는 없습니다.

대부분의 AWS 작업을 수행할 수 있는 권한을 부여하려면 이름이 같은 작업을 정책에 추가합니다. 예를 들어 사용자를 생성하려면 CreateUser API 작업, create-user CLI 명령 또는 AWS Management 콘솔을 사용해야 합니다. 이러한 작업을 수행하려면 CreateUser 작업에 액세스할 수 있게 허용하는 정책이 있어야 합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iam:CreateUser",  
            "Resource": "*"  
        }  
    ]  
}
```

정책에 GetSessionToken 작업을 포함할 수 있지만, 사용자가 GetSessionToken 작업을 수행할 수 있는 권한에는 영향을 미치지 않습니다.

GetSessionToken에서 부여하는 권한

GetSessionToken이 IAM의 자격 증명으로 호출되면, 임시 보안 자격 증명은 IAM 사용자와 동일한 권한을 갖습니다. 마찬가지로 GetSessionToken이 AWS 계정 루트 사용자으로 호출되면, 임시 보안 자격 증명은 루트 사용자 권한을 갖습니다.

Note

루트 사용자 자격 증명으로 GetSessionToken을 호출하지 않는 것이 좋습니다. 대신에 [모범 사례 \(p. 43\)](#)에 따라 필요한 권한을 지닌 IAM 사용자를 생성하십시오. 그런 다음 이러한 IAM 사용자를 AWS와의 일상적인 상호 작용에 사용하십시오.

GetSessionToken을 호출할 때 얻는 임시 자격 증명은 다음과 같은 기능과 한계를 지닙니다.

- <https://signin.aws.amazon.com/federation>에서 페더레이션 Single Sign-On 앤드포인트로 자격 증명을 전달하여 AWS Management 콘솔에 액세스할 수 있습니다. 자세한 내용은 [연동 사용자가 AWS Management 콘솔에 액세스할 수 있게 하는 URL 생성\(사용자 지정 연동 브로커\) \(p. 188\)](#) 단원을 참조하십시오.
- 자격 증명을 사용해 IAM 또는 AWS STS API 작업을 호출할 수 없습니다. 자격 증명을 사용해 다른 AWS 서비스에 대한 API 작업을 호출할 수는 있습니다.

[AWS STS API 작업 비교 \(p. 273\)](#)에서 이 API 작업과 이 작업의 한계 및 기능을 임시 보안 자격 증명을 생성하는 다른 API와 비교해 보십시오.

GetSessionToken을 사용한 MFA 보호 API 액세스에 대한 자세한 내용은 [MFA 보호 API 액세스 구성 \(p. 122\)](#) 단원을 참조하십시오.

임시 보안 자격 증명에 대한 권한 비활성화

임시 보안 자격 증명은 만료될 때까지 유효하며 취소될 수 없습니다. 그러나 권한은 자격 증명을 사용해 AWS 요청이 이루어질 때마다 평가되기 때문에 자격 증명이 발급된 이후에라도 자격 증명에 대한 권한을 변경함으로써 자격 증명 취소 효과를 얻을 수 있습니다. 임시 보안 자격 증명에서 모든 권한을 제거하는 경우 그 자격 증명을 사용하는 후속 AWS 요청은 실패하게 됩니다. 임시 보안 자격 증명에 할당된 권한을 변경 또는 제거하는 메커니즘은 다음 섹션에 설명되어 있습니다.

Note

기존 정책 권한을 업데이트할 때 또는 사용자나 리소스에 새 정책을 적용할 때 정책 업데이트가 효력이 생기는 데 몇 분이 걸릴 수 있습니다.

주제

- [임시 보안 자격 증명 생성자에 대한 액세스 거부 \(p. 284\)](#)
- [이름을 사용한 임시 보안 자격 증명에 대한 액세스 거부 \(p. 284\)](#)
- [특정 시각 이전에 발급된 임시 보안 자격 증명에 대한 액세스 거부 \(p. 285\)](#)

임시 보안 자격 증명 생성자에 대한 액세스 거부

임시 보안 자격 증명에 할당된 권한을 비활성화 또는 제거하려면 자격 증명 생성자와 연결된 권한을 변경 또는 제거하면 됩니다. 자격 증명 생성자는 자격 증명 획득에 사용된 AWS STS API에 의해 결정됩니다. 이 생성자에 연결된 권한을 변경 또는 제거하는 메커니즘은 다음 섹션에 설명되어 있습니다.

AssumeRole, AssumeRoleWithSAML 또는 AssumeRoleWithWebIdentity에 의해 생성된 자격 증명에 대한 액세스 거부

AssumeRole, AssumeRoleWithSAML, 또는 AssumeRoleWithWebIdentity API 작업을 호출함으로써 획득한 임시 보안 자격 증명에 할당된 권한을 변경하거나 제거하려면 위임받은 역할에 대한 권한을 정의하는 역할 권한 정책을 편집 또는 삭제하면 됩니다. 역할을 수임함으로써 획득한 임시 보안 자격 증명은 수임된 역할에 대한 권한 정책에 정의된 것보다 더 많은 권한을 가질 수 없으며, 임시 보안 자격 증명에 할당된 권한은 AWS 호출에 사용될 때마다 평가됩니다. 역할의 권한 정책을 편집하거나 삭제하면 이러한 변경은 역할의 권한 정책을 변경하기 전에 발급된 자격 증명을 비롯해 해당 역할에 연결된 모든 임시 보안 자격 증명의 권한에 영향을 미칩니다. [IAM 역할의 임시 보안 자격 증명 취소 \(p. 245\)](#)의 단계를 따라 특정 세션에 대한 모든 권한을 즉시 최소화할 수 있습니다.

역할 권한 정책 편집에 대한 자세한 정보는 [역할 변경 \(p. 247\)](#) 단원을 참조하십시오.

GetFederationToken 또는 GetSessionToken에 의해 생성된 자격 증명에 대한 액세스 거부

GetFederationToken 또는 GetSessionToken API 작업을 호출함으로써 획득한 임시 보안 자격 증명에 할당된 권한을 변경 또는 제거하려면 GetFederationToken 또는 GetSessionToken을 호출하는 데 사용된 자격 증명의 IAM 사용자에 연결된 정책을 편집 또는 삭제하면 됩니다. GetFederationToken 또는 GetSessionToken을 호출하여 획득한 임시 보안 자격 증명은 권한 획득을 위해 자신의 자격 증명을 사용한 IAM 사용자보다 많은 권한을 가질 수 없습니다. 뿐만 아니라 임시 보안 자격 증명에 할당된 권한은 AWS 요청을 위해 사용될 때마다 평가됩니다. IAM 사용자의 권한을 편집 또는 삭제하면 그 변경 사항이 사용자가 생성한 모든 임시 보안 자격 증명뿐만 아니라 IAM 사용자에게도 영향을 미친다는 것에 유의하십시오.

Important

AWS 계정 루트 사용자에 대한 권한은 변경할 수 없습니다. 따라서 루트 사용자로 로그인할 때 GetFederationToken 또는 GetSessionToken을 호출하여 생성된 임시 보안 자격 증명에 대한 권한도 변경할 수 없습니다. 이런 이유 때문에 루트 사용자로 GetFederationToken 또는 GetSessionToken을 호출하지 않는 것이 좋습니다.

GetFederationToken 또는 GetSessionToken을 호출하는 데 자격 증명이 사용된 IAM 사용자와 연결된 정책을 변경 또는 제거하는 방법에 대한 정보는 [IAM 정책 관리 \(p. 377\)](#) 단원을 참조하십시오.

이름을 사용한 임시 보안 자격 증명에 대한 액세스 거부

자격 증명을 생성한 IAM 사용자 또는 역할의 권한에 영향을 미치지 않고 임시 보안 자격 증명에 대한 액세스를 거부할 수 있습니다. 액세스를 거부하려면 리소스 기반 정책의 Principal 요소에서 임시 보안 자격 증명의 Amazon 리소스 이름(ARN)을 지정하면 됩니다. (일부 AWS 서비스만이 리소스 기반 정책을 지원합니다).

연동 사용자에 대한 액세스 거부

예를 들어 이름이 token-app인 IAM 사용자가 있고 그 사용자의 자격 증명이 GetFederationToken을 호출하는 데 사용된다고 가정합시다. GetFederationToken API 호출로 인해 Bob이라는 연동 사용자(연동 사용자의 이름은 API 호출의 Name 파라미터에서 가져옵니다)와 연결된 임시 보안 자격 증명이 생성되었습니다. 연동 사용자 Bob이 EXAMPLE-BUCKET이라는 S3 버킷에 액세스하는 것을 거부하려면 아래 예시된 버킷 정책을 EXAMPLE-BUCKET에 연결하면 됩니다. 이렇게 하면 연동 사용자의 Amazon S3 권한에만 영향을 미칠 뿐 연동 사용자에게 부여된 다른 권한들은 영향을 받지 않는 것에 유의하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Principal": {"AWS": "arn:aws:sts::ACCOUNT-ID-WITHOUT-HYPHENS:federated-user/Bob"},  
        "Action": "s3:GetObject",  
        "Resource": "arn:aws:s3:::EXAMPLE-BUCKET/*"  
    }  
}
```

```
        "Effect": "Deny",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::EXAMPLE-BUCKET"
    }
}
```

연동 사용자를 지정하는 대신에 버킷 정책의 Principal 요소에 있는 GetFederationToken을 호출하는 데 자격 증명이 사용된 IAM 사용자의 ARN을 지정할 수 있습니다. 이 경우 이전 정책의 Principal 요소는 다음과 같을 것입니다.

```
"Principal": {"AWS": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:user/token-app"}
```

정책에서 IAM 사용자 token-app의 ARN을 지정하면 그 결과 Bob이라는 연동 사용자뿐만 아니라 token-app에 의해 생성된 모든 연동 사용자에 대한 액세스가 거부된다는 점에 유의하십시오.

수임된 역할 사용자에 대한 액세스 거부

역할 수임에 의해 생성된 임시 보안 자격 증명의 ARN을 지정할 수도 있습니다. 차이는 리소스 기반 정책의 Principal 요소에서 사용되는 구문에 있습니다. 예를 들어 사용자가 Accounting-Role이라는 역할을 수임하고 RoleSessionName의 Mary를 지정한다고 가정합시다(RoleSessionName은 AssumeRole API 호출의 파라미터입니다). 이 API 호출로 인해 얻은 임시 보안 자격 증명에 대한 액세스를 거부하려면 리소스 기반 정책의 Principal 요소는 다음과 같아야 합니다.

```
"Principal": {"AWS": "arn:aws:sts::ACCOUNT-ID-WITHOUT-HYPHENS:assumed-role/Accounting-Role/Mary"}
```

리소스 기반 정책의 Principal 요소에 있는 IAM의 ARN을 다음 예시와 같이 지정할 수도 있습니다. 이 경우 그 정책으로 인해 Accounting-Role이라는 역할과 연결된 모든 임시 보안 자격 증명에 대한 액세스는 거부될 것입니다.

```
"Principal": {"AWS": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:role/Accounting-Role"}
```

특정 시각 이전에 발급된 임시 보안 자격 증명에 대한 액세스 거부

특정 시각 또는 날짜 이전에 생성된 임시 보안 자격 증명에 대한 액세스만 거부할 수도 있습니다. 이렇게 하려면 정책의 aws:TokenIssueTime 요소에 있는 condition의 값을 지정해야 합니다. 다음 정책은 한 가지 예를 보여줍니다. 임시 보안 자격 증명을 생성한 IAM 사용자에게 다음 예시와 유사한 정책을 연결합니다. 그 정책은 aws:TokenIssueTime의 값이 지정된 날짜와 시각보다 이른 경우에만 모든 권한을 거부합니다. aws:TokenIssueTime의 값은 임시 보안 자격 증명이 생성된 정확한 시간과 일치합니다. aws:TokenIssueTime 값은 임시 보안 자격 증명으로 로그인된 AWS 요청의 콘텍스트에서만 존재하므로 정책의 Deny 문은 IAM 사용자의 장기 자격 증명으로 로그인한 요청에는 영향을 미치지 않습니다.

다음 정책도 역할에 연결할 수 있습니다. 이 경우 정책은 지정된 시각 및 날짜 이전에 그 역할에 의해 생성된 임시 보안 자격 증명에만 영향을 미칩니다. 자격 증명이 지정된 시각 및 날짜 이후에 그 역할에 의해 생성된 경우 정책의 Condition 요소가 거짓으로 평가되어 Deny 문은 영향을 미치지 않습니다.

Example 발급 시각을 사용해 임시 자격 증명에 대한 모든 권한을 거부하는 정책

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "*",
        "Resource": "*",
        "Condition": {"DateLessThan": {"aws:TokenIssueTime": "2014-05-07T23:47:00Z"}}
    }
}
```

이러한 식으로 세션이 취소된 유효한 사용자는 작업을 계속하려면 새 세션을 위한 임시 자격 증명을 가져와야 합니다. AWS CLI는 자격 증명이 만료될 때까지 이를 캐시합니다. CLI가 더 이상 유효하지 않은 캐시된 자격 증명을 강제로 삭제하고 새로 고치게 하려면 다음 명령 중 하나를 실행합니다.

Linux, MacOS 또는 Unix

```
$ rm -r ~/.aws/cli/cache
```

Windows

```
C:\> del /s /q %UserProfile%\aws\cli\cache
```

임시 보안 자격 증명을 생성할 수 있는 권한 부여

기본적으로 IAM 사용자는 연동 사용자 및 역할을 위한 임시 보안 자격 증명을 생성할 수 있는 권한이 없습니다. 사용자에게 이러한 권한을 제공하려면 정책을 사용해야 합니다. 사용자에게 직접 권한을 부여할 수 있지만, 그룹에게 권한을 부여할 것을 강력히 권고합니다. 그렇게 하면 권한 관리가 훨씬 쉬워집니다. 어떤 사용자가 권한에 연결된 작업을 수행할 필요가 더 이상 없는 경우에는 그룹에서 그 사용자를 삭제하기만 하면 됩니다. 다른 어떤 사용자가 그 작업을 수행해야 한다면 해당 그룹에 추가해 권한을 부여하면 됩니다.

연동 사용자 또는 역할을 위해 임시 보안 자격 증명을 생성할 수 있는 권한을 IAM 그룹에게 부여하려면 다음 권한 중 하나 또는 둘 다를 부여하는 정책을 연결하면 됩니다.

- 연동 사용자들이 IAM 역할에 액세스하도록 하려면 AWS STS AssumeRole에 대한 액세스 권한을 부여하십시오.
- 역할이 필요 없는 연동 사용자에 대해서는 AWS STS GetFederationToken에 대한 액세스 권한을 부여하십시오.

AssumeRole 및 GetFederationToken API 작업 간의 차이점을 보려면 [임시 보안 자격 증명 요청하기 \(p. 265\)](#) 단원을 참조하십시오.

IAM 사용자는 [GetSessionToken](#)을 호출하여 임시 보안 자격 증명을 생성할 수도 있습니다. 사용자는 권한이 없어도 [GetSessionToken](#)을 호출할 수 있습니다. 이 작업의 목적은 MFA를 사용하는 사용자를 인증하는 것입니다. 정책을 사용하여 인증을 제어할 수는 없습니다. 즉 IAM 사용자가 임시 자격 증명을 생성할 목적으로 [GetSessionToken](#)을 호출하는 작업을 하지 못하게 할 수 없습니다.

Example : 역할 수임 권한을 부여하는 정책

다음 정책 예시는 AWS 계정 AssumeRole의 UpdateApp 역할을 위해 123123123123을 호출할 수 있는 권한을 부여합니다. AssumeRole을 사용하는 경우, 연합된 사용자를 대신해 보안 자격 증명을 생성하는 사용자(또는 애플리케이션)는 역할 권한 정책에 아직 지정되지 않은 어떤 권한도 위임할 수 없습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": "sts:AssumeRole",  
         "Resource": "arn:aws:iam::123123123123:role/UpdateAPP"  
     ]  
}
```

Example : 연동 사용자를 위한 임시 보안 자격 증명을 생성할 수 있는 권한을 부여하는 정책

다음과 같은 정책 예시는 GetFederationToken에 액세스할 수 있는 권한을 부여합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": "sts:GetFederationToken",  
         "Resource": "arn:aws:iam::123123123123:role/UpdateAPP"  
     ]  
}
```

```
    "Statement": [{}  
      "Effect": "Allow",  
      "Action": "sts:GetFederationToken",  
      "Resource": "*"  
    ]  
}
```

Important

IAM 사용자에게 GetFederationToken을 사용해 연합된 사용자를 위한 임시 보안 자격 증명을 생성할 수 있는 권한을 부여하면 이로써 해당 사용자가 자신의 권한을 위임할 수 있게 허용하는 것 이므로 주의하시기 바랍니다. 여러 IAM 사용자 및 AWS 계정에 걸쳐 권한을 위임하는 것에 대한 자세한 내용은 [액세스 권한 위임을 위한 정책의 예 \(p. 223\)](#) 단원을 참조하십시오. 임시 보안 자격 증명에서 권한을 제어하는 것에 대한 자세한 정보는 [사용자 임시 보안 자격 증명에 대한 권한 제어 \(p. 277\)](#) 단원을 참조하십시오.

Example : 연동 사용자에 대해 임시 보안 자격 증명을 생성할 수 있는 사용자 제한 권한을 부여하는 정책

IAM 사용자가 GetFederationToken을 호출하도록 허용할 때는 IAM 사용자가 위임할 수 있는 권한을 제한하는 것이 좋습니다. 예를 들어 다음 정책은 IAM 사용자가 이름이 Manager로 시작하는 연동 사용자에 대해서만 임시 보안 자격 증명을 생성하도록 허용하는 방법을 보여줍니다.

```
{  
  "Version": "2012-10-17",  
  "Statement": [{}  
    "Effect": "Allow",  
    "Action": "sts:GetFederationToken",  
    "Resource": ["arn:aws:sts::123456789012:federated-user/Manager*"]  
  ]  
}
```

AWS 리전에서 AWS STS 활성화 및 비활성화

기본적으로 AWS Security Token Service(AWS STS)는 전역적 서비스로 사용 가능하고 모든 AWS STS 요청은 <https://sts.amazonaws.com>의 단일 엔드포인트로 전송됩니다. 아래 표에 나타난 모든 AWS 리전의 엔드포인트로 AWS STS 요청을 보낼 수도 있습니다. 모든 리전이 기본적으로 활성화되어 있지만, 필요하지 않은 리전은 비활성화할 수 있습니다.

다음과 같은 이유로 리전 엔드포인트에 AWS STS 요청을 보낼 수도 있습니다.

- 지연 시간 감소 – 서비스 및 애플리케이션에서 지리적으로 더 가까운 엔드포인트에 AWS STS 호출을 함으로써 지연 시간 및 응답 시간을 단축하며 AWS STS 서비스에 액세스할 수 있습니다.
- 중복된 구축 – AWS STS API 호출을 다른 리전으로 전환하는 코드를 애플리케이션에 추가함으로써 최초 리전이 응답을 멈추더라도 애플리케이션이 계속 작동하도록 합니다. 이러한 중복성은 자동으로 구축되지 않으므로 코드에 해당 기능을 구축해야 합니다.

계정에 대해 어떤 리전을 활성화하면 해당 리전의 엔드포인트로 요청이 전송될 경우 해당 리전의 STS 엔드포인트를 활성화하여 해당 계정의 사용자 및 역할에 대한 임시 자격 증명을 발행할 수 있습니다. 자격 증명은 계속 인식되며 전역적으로 사용 가능합니다. 호출자 계정이 아니라 임시 자격 증명을 요청한 계정으로 리전을 활성화해야 합니다.

예를 들어, 계정 A의 사용자가 STS 리전 엔드포인트 sts:AssumeRole으로 <https://sts.us-west-2.amazonaws.com> API 요청을 보내려 할 수 있습니다. 이는 계정 B의 Developer 역할에 대한 임시 자격 증명을 요청하기 위한 것입니다. 이 요청은 계정 B의 엔터티에 대한 자격 증명을 만들기 위한 것이기 때문에 계정 B는 us-west-2 리전을 활성화해야 합니다. 계정 A(또는 다른 계정)의 사용자는 자신의 계정에서 리전이 활성화되었는지 여부와 상관없이 us-west-2 엔드포인트를 호출하여 계정 B의 자격 증명을 요청할 수 있습니다.

리전에서 AWS STS를 활성화하거나 비활성화하려면

Note

모든 리전은 기본적으로 활성화되어 있습니다. 따라서 이전에 리전을 비활성화한 경우에만 활성화하면 됩니다.

- 새 리전에서 AWS STS를 활성화하고 싶은 계정에 대해 IAM 관리 작업("iam:*)을 수행할 권한을 지닌 IAM 사용자로 로그인합니다.
- IAM 콘솔을 열고 탐색 창에서 [계정 설정](#)을 선택합니다.
- Security Token Service Regions(보안 토큰 서비스 리전) 목록을 확장하여 사용할 리전을 찾은 다음 활성화 또는 비활성화를 선택합니다.

AWS STS 리전 사용을 위한 코드 작성

AWS 계정에 대해 리전을 활성화하면 해당 리전으로 AWS STS API 호출을 보낼 수 있습니다. 다음 Java 코드 조각은 AWSSecurityTokenServiceClient 객체를 구성해 setEndpoint 메서드로 EU(아일랜드) (eu-west-1) 리전에서 요청하는 방법을 보여줍니다.

```
AWSecurityTokenServiceClient stsClient = new AWSsecurityTokenServiceClient();
stsClient.setEndpoint("sts.eu-west-1.amazonaws.com");
```

Important

setRegion 메서드는 이전 버전과의 호환성을 위해 계속해서 원래의 전역적 단일 엔드포인트 인 AWS STS로 확인되기 때문에 이 메서드로 sts에 대한 리전 엔드포인트를 설정해서는 안 됩니다.[amazonaws.com](https://aws.amazon.com)

예제의 첫 번째 줄에서 AWSSecurityTokenServiceClient라는 stsClient 객체를 인스턴스화합니다. 두 번째 줄에서는 stsClient 메서드를 호출하고 엔드포인트의 URL을 유일한 파라미터로 전달하여 setEndpoint 객체를 구성합니다. stsClient 객체를 사용하는 모든 API 호출은 이제 지정된 엔드포인트로 전송됩니다.

다른 모든 언어 및 프로그래밍 환경의 조합에 대해서는 [해당 SDK 문서](#)를 참조하십시오.

리전의 AWS STS 사용과 관련하여 그 밖에 어떤 것도 달라지지 않습니다. 항상 그렇듯이 리전의 AWS STS 엔드포인트에서 가져온 자격 증명은 해당 리전에 국한되지 않고 전역적으로 사용할 수 있습니다.

다음 표에서는 해당 리전과 그 엔드포인트를 나열합니다. 기본적으로 어떤 것들이 활성화되며, 어떤 것을 활성화 또는 비활성화할 수 있는지를 보여줍니다.

리전 이름	엔드포인트	활성화/비활성화될 수 있음
--전 세계--	sts.amazonaws.com	아니요
미국 동부(오하이오)	sts.us-east-2.amazonaws.com	예
미국 동부(버지니아 북부)	sts.us-east-1.amazonaws.com	아니요
미국 서부(캘리포니아 북부 지역)	sts.us-west-1.amazonaws.com	예
미국 서부(오레곤)	sts.us-west-2.amazonaws.com	예
캐나다(중부)	sts.ca-central-1.amazonaws.com	예

리전 이름	엔드포인트	활성화/비활성화될 수 있음
아시아 태평양(도쿄)	sts.ap-northeast-1.amazonaws.com	예
아시아 태평양(서울)	sts.ap-northeast-2.amazonaws.com	예
아시아 태평양(뭄바이)	sts.ap-south-1.amazonaws.com	예
아시아 태평양(싱가포르)	sts.ap-southeast-1.amazonaws.com	예
아시아 태평양(시드니)	sts.ap-southeast-2.amazonaws.com	예
EU(프랑크푸르트)	sts.eu-central-1.amazonaws.com	예
EU(아일랜드)	sts.eu-west-1.amazonaws.com	예
EU(런던)	sts.eu-west-2.amazonaws.com	예
EU(파리)	sts.eu-west-3.amazonaws.com	예
남아메리카(상파울루)	sts.sa-east-1.amazonaws.com	예

Note

`us-east-2.amazonaws.com`과 같은 리전 엔드포인트에 대한 호출은 리전 서비스에 대한 모든 호출과 마찬가지로 AWS CloudTrail에서 로깅됩니다. 전역적 엔드포인트 `sts.amazonaws.com`에 대한 호출은 글로벌 서비스에 대한 호출로 로깅됩니다. 자세한 내용은 [AWS CloudTrail을 사용하여 IAM 및 AWS STS API 호출 로깅 \(p. 293\)](#) 단원을 참조하십시오.

AWS STS 인터페이스 VPC 엔드포인트 사용

Amazon Virtual Private Cloud(Amazon VPC)를 사용하여 AWS 리소스를 호스팅하는 경우, VPC와 AWS STS 간에 프라이빗 연결을 설정할 수 있습니다. 이 연결을 사용하면 AWS STS가 퍼블릭 인터넷을 통하지 않고 VPC의 리소스와 통신하게 할 수 있습니다.

Amazon VPC란 사용자가 정의한 가상 네트워크에서 AWS 리소스를 시작할 때 사용할 수 있는 AWS 서비스입니다. VPC를 사용하여 IP 주소 범위, 서브넷, 라우팅 테이블, 네트워크 게이트웨이 등의 네트워크 설정을 제어할 수 있습니다. VPC를 AWS STS에 연결하려면 AWS STS에 대해 인터페이스 VPC 엔드포인트를 정의하십시오. 이 엔드포인트를 이용하면 인터넷 게이트웨이나 NAT(네트워크 주소 번환) 인스턴스 또는 VPN 연결 없이도 AWS STS에 안정적이고 확장 가능하게 연결됩니다. 자세한 내용은 Amazon VPC 사용 설명서의 [Amazon VPC란 무엇입니까?](#)를 참조하십시오.

인터페이스 VPC 엔드포인트는 프라이빗 IP 주소와 함께 탄력적 네트워크 인터페이스를 사용하여 AWS 서비스 간 프라이빗 통신을 사용할 수 있는 AWS 기술인 AWS PrivateLink에 의해 구동됩니다. 자세한 내용은 [AWS 서비스를 위한 AWS PrivateLink](#)를 참조하십시오.

다음은 Amazon VPC 사용자를 위한 단계들입니다. 자세한 내용은 Amazon VPC 사용 설명서의 [Amazon VPC 시작하기](#)를 참조하십시오.

가용성

현재 AWS STS는 미국 서부(오레곤) 리전에서만 VPC 엔드포인트를 지원합니다.

AWS STS를 위한 VPC 만들기

VPC에서 AWS STS를 사용하기 시작하려면 AWS STS에 대한 인터페이스 VPC 엔드포인트를 생성합니다. 자세한 내용은 Amazon VPC 사용 설명서의 [인터넷포인트 만들기](#)를 참조하십시오.

VPC 엔드포인트를 만든 후에는 해당 리전의 엔드포인트를 사용하여 AWS STS 요청을 보내야 합니다. 자세한 내용은 [AWS 리전에서 AWS STS 활성화 및 비활성화 \(p. 287\)](#) 단원을 참조하십시오. 리전의 엔드포인트를 사용할 경우, AWS STS는 퍼블릭 엔드포인트 또는 프라이빗 인터페이스 VPC 엔드포인트 중 사용 중인 엔드포인트를 사용하여 다른 AWS 서비스를 호출합니다. 예를 들어 AWS STS를 위한 인터페이스 VPC 엔드포인트를 만들어 VPC에 있는 리소스의 AWS STS의 임시 자격 증명을 이미 요청한 경우를 예로 들어 보겠습니다. 이 경우 이러한 자격 증명은 기본적으로 인터페이스 VPC 엔드포인트를 통합니다.

임시 자격 증명을 사용하는 샘플 애플리케이션

AWS Security Token Service(AWS STS)를 사용하면 AWS 리소스에 대한 액세스를 제어할 수 있는 임시 보안 자격 증명을 생성하여 신뢰받는 사용자에게 제공할 수 있습니다. AWS STS에 대한 자세한 내용은 [임시 보안 자격 증명 \(p. 263\)](#) 단원을 참조하십시오. AWS STS를 사용해 임시 보안 자격 증명을 관리하는 방법에 대해 알아보려면, 완전한 샘플 시나리오를 구현하는 다음과 같은 샘플 애플리케이션을 다운로드하십시오.

- [Active Directory 사용 사례를 위한 자격 증명 연동 샘플 애플리케이션](#) Active Directory(.NET/C#)에서 정의된 사용자에 연결된 권한을 사용해 Amazon S3 파일 및 버킷에 액세스하기 위한 임시 보안 자격 증명을 발급하는 방법을 보여줍니다.
- [AWS Management Console 연동 프록시 샘플 사용 사례](#) Single-Sign-On(SSO)을 가능케 하는 사용자 지정 연동 프록시를 생성해 기존 Active Directory 사용자가 AWS Management 콘솔에 로그인할 수 있게 하는 방법을 보여줍니다(.NET/C#).
- [Shibboleth를 AWS Identity and Access Management와 통합하기 Shibboleth 및 SAML \(p. 167\)](#)을 사용해 사용자에게 AWS Management 콘솔에 대한 SSO(Single-Sign-On) 액세스 권한을 제공하는 방법을 보여줍니다.

웹 자격 증명 연동에 대한 예시

다음 샘플 애플리케이션은 Login with Amazon, Amazon Cognito, Facebook 또는 Google 같은 공급자를 통해 웹 자격 증명 연동을 사용하는 방법을 보여 줍니다. 이러한 임시 AWS 보안 자격 증명에 대해 이러한 공급자의 인증을 얻고 AWS 서비스에 액세스할 수 있습니다.

- [Amazon Cognito 자습서](#) – 모바일 개발용 AWS SDK를 통해 Amazon Cognito를 사용하는 것이 좋습니다. Amazon Cognito는 모바일 앱을 위한 자격 증명을 관리하는 가장 간단한 방법으로서, 동기화 및 교차 디바이스 자격 증명과 같은 부가 기능을 제공합니다. Amazon Cognito에 대한 자세한 내용은 Android용 AWS Mobile SDK Developer Guide의 [Amazon Cognito 자격 증명](#) 및 AWS Mobile SDK for iOS Developer Guide의 [Amazon Cognito 자격 증명을 사용한 사용자 인증](#) 단원을 참조하십시오.
- [Web Identity Federation Playground](#). 이 웹 사이트는 [웹 자격 증명 연동 \(p. 162\)](#) 및 `AssumeRoleWithWebIdentity` API를 대화식으로 보여줍니다.
- [AWS Elastic Beanstalk 및 Login with Amazon을 사용한 연동 웹 자격 증명 애플리케이션 구축 및 배포](#) 이 블로그 게시글은 `AssumeRoleWithWebIdentity`를 사용해 웹 자격 증명 연동 및 Login with Amazon을 통해 임시 보안 자격 증명을 얻는 방법을 기술합니다. 또한, Elastic Beanstalk에서 실행되는 Python 웹 애플리케이션에서 그 자격 증명을 사용해 AWS를 호출하는 방법을 설명합니다.

임시 보안 자격 증명에 관한 추가 리소스

다음 시나리오 및 애플리케이션은 임시 보안 자격 증명 사용 방법을 안내합니다.

- [웹 자격 증명 연동에 대하여 \(p. 162\)](#). 이 섹션에서는 웹 자격 증명 연동 및 `AssumeRoleWithWebIdentity` API를 사용할 때 IAM 역할을 구성하는 방법을 설명합니다.
- [MFA 보호 API 액세스 구성 \(p. 122\)](#). 이 주제는 역할을 사용해 멀티 팩터 인증(MFA)이 계정에서 민감한 API 작업을 보호하도록 요구하는 방법을 설명합니다.
- [자격 증명 등록을 위한 토큰 벤딩 머신](#). 이 샘플 Java 웹 애플리케이션은 `GetFederationToken` API를 사용해 원격 클라이언트에게 임시 보안 자격 증명을 제공합니다.

AWS의 정책 및 권한에 대한 자세한 내용은 다음 주제를 참조하십시오.

- [액세스 관리 \(p. 304\)](#)
- [정책 평가 로직 \(p. 531\).](#)
- Amazon Simple Storage Service 개발자 가이드의 [Amazon S3 리소스에 대한 액세스 권한 관리](#).

AWS 계정 루트 사용자

Amazon Web Services(AWS) 계정을 처음 생성하면 전체 AWS 서비스 및 계정 리소스에 대한 완전한 액세스 권한을 지닌 단일 로그인 자격 증명으로 시작합니다. 이 자격 증명은 AWS 계정 루트 사용자라고 하며, 계정을 생성할 때 사용한 이메일 주소와 암호로 로그인하여 액세스합니다.

Important

일상적인 작업, 심지어 관리 작업의 경우에도 루트 사용자를 사용하지 마실 것을 강력히 권장합니다. 대신, [IAM 사용자를 처음 생성할 때만 루트 사용자를 사용하는 모범 사례 \(p. 44\)](#)를 준수하십시오. 그런 다음 루트 사용자 자격 증명은 안전하게 보관하다가 몇몇 계정 및 서비스 관리 작업을 수행할 때만 사용하십시오. 루트 사용자로 로그인해야 하는 작업을 보려면 [루트 사용자가 필요한 AWS 작업을 참조하십시오](#). 일상적 사용을 위해 관리자를 설정하는 방법에 대한 자습서는 [첫 번째 IAM 관리자 및 그룹 생성 \(p. 17\)](#) 단원을 참조하십시오.

AWS 계정 루트 사용자에 대한 액세스 키(액세스 키 ID 및 보안 액세스 키)를 생성, 교체, 비활성화 또는 삭제할 수 있습니다. 루트 사용자 암호를 변경할 수도 있습니다. AWS 계정에 대한 루트 사용자 자격 증명을 보유한 사람은 누구든지 결제 정보를 포함하여 해당 계정의 모든 리소스에 무제한으로 액세스할 수 있습니다.

액세스 키를 만들 때 액세스 키 ID와 보안 액세스 키를 한 세트로 생성합니다. 액세스 키 생성 중에 AWS는 액세스 키의 보안 액세스 키 부분을 확인하고 다운로드할 기회를 한 번 부여합니다. 보안 액세스 키를 다운로드하지 않았거나 분실한 경우 액세스 키를 삭제한 다음 새로 생성할 수 있습니다. [IAM 콘솔](#), AWS CLI 또는 AWS API에서 IAM 사용자 액세스 키를 생성할 수 있습니다. 자세한 내용은 IAM 사용 설명서에서 [IAM 사용자의 액세스 키 관리](#)를 참조하십시오. AWS 계정 루트 사용자에 대한 액세스 키를 생성하려면 AWS Management 콘솔을 사용해야 합니다.

새로 생성한 액세스 키는 활성 상태입니다. 즉, CLI 및 API 호출에 대해 액세스 키를 사용할 수 있습니다. 각 IAM 사용자에 대한 [액세스 키는 두 개로 제한됩니다](#). 이는 [액세스 키를 교체](#)하려는 경우에 유용합니다. 또한 루트 사용자에 최대 두 개의 액세스 키를 할당할 수 있습니다. 액세스 키를 비활성화한 경우 API 호출에 액세스 키를 사용할 수 없으며, 비활성 키는 제한에 포함됩니다. 언제든지 액세스 키를 생성하거나 삭제할 수 있습니다. 그러나 액세스 키를 삭제하면 영구 삭제되어 되돌릴 수 없습니다.

주제

- [AWS 계정 루트 사용자의 MFA 활성화 \(p. 291\)](#)
- [루트 사용자를 위한 액세스 키 생성 \(p. 292\)](#)
- [루트 사용자로부터 액세스 키 삭제하기 \(p. 292\)](#)
- [루트 사용자의 암호 변경 \(p. 293\)](#)

AWS 계정 루트 사용자의 MFA 활성화

루트 사용자 자격 증명을 계속 사용할 경우, 보안 모범 사례에 따라 계정에 대한 멀티 팩터 인증(MFA)을 활성화하는 것이 좋습니다. 루트 사용자는 계정에서 민감한 작업을 수행할 수 있기 때문에 인증 단계를 추가하면 계정의 보안을 강화하는 데 도움이 됩니다. 여러 유형의 MFA가 있습니다. MFA 활성화에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS 계정 루트 사용자용 가상 MFA 디바이스 활성화\(콘솔\) \(p. 100\)](#)

- AWS 계정 루트 사용자용 하드웨어 MFA 디바이스 활성화(콘솔) (p. 110)

루트 사용자를 위한 액세스 키 생성

AWS Management 콘솔 또는 AWS 프로그래밍 도구를 사용하여 루트 사용자에 대한 액세스 키를 생성할 수 있습니다.

AWS 계정 루트 사용자에 대한 액세스 키를 생성하려면(콘솔 사용)

1. AWS 계정 이메일 주소와 암호를 사용하여 [AWS Management 콘솔](#)에 AWS 계정 루트 사용자로 로그인 합니다.

Note

이전에 [IAM 사용자](#) 자격 증명으로 콘솔에 로그인한 경우, 브라우저가 이 설정을 기억하여 계정 별 로그인 페이지를 열 수도 있습니다. IAM 사용자 로그인 페이지에서는 AWS 계정 루트 사용자 자격 증명으로 로그인할 수 없습니다. IAM 사용자 로그인 페이지가 나타날 경우에는 페이지 하단에 있는 [Sign in using 루트 사용자 credentials]를 선택하여 기본 로그인 페이지로 돌아갑니다. 기본 로그인 페이지에서 AWS 계정의 이메일 주소와 암호를 입력합니다.

2. 탐색 표시줄에서 계정 이름을 선택한 다음 내 보안 자격 증명을 선택합니다.
3. AWS 계정의 보안 자격 증명에 대한 액세스와 관련해 경고가 나타나면, Continue to Security Credentials(보안 자격 증명으로 계속)을 선택하십시오.
4. Access keys(access key ID and secret access key)(액세스 키(액세스 키 ID 및 보안 액세스 키)) 섹션을 확장합니다.
5. Create New Access Key(새 액세스 키 생성)을 선택하십시오. 이 기능이 비활성화되면 새 키를 생성할 수 있기 전에 기존 액세스 키 중 하나를 삭제해야 합니다. 자세한 내용은 IAM 사용 설명서의 [IAM 인터티 객체 제한](#)을 참조하십시오.

보안 액세스 키를 보거나 다운로드할 수 있는 기회는 이번 한 번뿐이라는 경고가 표시됩니다. 나중에는 조회할 수 없습니다.

- Show Access Key(액세스 키 표시)를 선택하면 브라우저 창에서 액세스 키 ID 및 보안 키를 복사해 다른 곳에 붙여넣기할 수 있습니다.
 - Download Key File(키 파일 다운로드)을 선택하면 액세스 키 ID 및 보안 키가 저장된 `rootkey.csv`라는 이름의 파일을 받게 됩니다. 파일을 안전한 곳에 저장합니다.
6. 액세스 키를 더 이상 사용하지 않을 때는 오용되지 않도록 [삭제하거나 \(p. 48\)](#) Make Inactive(비활성화)를 선택하여 비활성 상태로 표시할 것을 권합니다.

루트 사용자에 대한 액세스 키(AWS CLI 또는 AWS API)를 생성하려면

다음 중 하나를 사용하십시오.

- AWS CLI: [aws iam create-access-key](#)
- AWS API: [CreateAccessKey](#)

루트 사용자로부터 액세스 키 삭제하기

AWS Management 콘솔 또는 다양한 프로그래밍 도구를 사용하여 루트 사용자에 대한 액세스 키를 삭제할 수 있습니다.

AWS 계정 루트 사용자(콘솔)에서 액세스 키를 삭제하려면

1. AWS 계정 이메일 주소와 암호를 사용하여 [AWS Management 콘솔](#)에 루트 사용자로 로그인합니다.

Note

이전에 [IAM 사용자 \(p. 63\)](#) 자격 증명으로 콘솔에 로그인한 경우, 브라우저가 이 설정을 기억하여 계정별 로그인 페이지를 열 수도 있습니다. IAM 사용자 로그인 페이지에서는 AWS 계정 루트 사용자 자격 증명으로 로그인할 수 없습니다. IAM 사용자 로그인 페이지가 나타날 경우에는 페이지 하단에 있는 [Sign in using root account credentials]를 선택하여 기본 로그인 페이지로 돌아갑니다. 기본 로그인 페이지에서 AWS 계정의 이메일 주소와 암호를 입력합니다.

2. 탐색 표시줄에서 계정 이름을 선택한 다음 내 보안 자격 증명을 선택합니다.
3. AWS 계정의 보안 자격 증명에 대한 액세스와 관련해 경고가 나타나면, Continue to Security Credentials(보안 자격 증명으로 계속)을 선택하십시오.
4. Access keys(access key ID and secret access key)(액세스 키(액세스 키 ID 및 보안 액세스 키)) 섹션을 확장합니다.
5. 삭제하고자 하는 액세스 키를 찾은 다음, 작업 열에서 삭제를 선택합니다.

Note

액세스 키를 삭제하는 대신에 비활성 상태로 표시할 수 있습니다. 이렇게 하면 키 ID나 보안 키를 변경하지 않고도 나중에 액세스 키를 다시 사용할 수 있습니다. 비활성 상태에 있는 동안에는 AWS API에 대한 요청을 통해 액세스 키를 사용하려는 시도는 액세스 거부 상태로 인해 실패합니다.

루트 사용자에 대한 액세스 키(AWS CLI 또는 AWS API)를 삭제하려면 다음 중 하나를 사용하십시오.

- AWS CLI: `aws iam delete-access-key`
- AWS API: `DeleteAccessKey`

루트 사용자의 암호 변경

루트 사용자의 암호 변경에 대한 자세한 내용은 [AWS 계정 루트 사용자 암호 변경 \(p. 78\)](#) 단원을 참조하십시오. 루트 사용자를 변경하려면 루트 사용자 자격 증명을 사용하여 로그인해야 합니다. 루트 사용자로 로그인해야 하는 작업을 보려면 [루트 사용자가 필요한 AWS 작업](#)을 참조하십시오.

AWS CloudTrail을 사용하여 IAM 및 AWS STS API 호출 로깅

IAM 및 AWS STS는 IAM 사용자 또는 역할이 수행한 작업에 대한 레코드를 제공하는 서비스인 AWS CloudTrail과 통합됩니다. CloudTrail은 콘솔과 API 호출에서 오는 호출을 비롯하여 IAM 및 AWS STS에 대한 모든 API 호출을 이벤트로 캡처합니다. 추적을 생성하면 Amazon S3 버킷으로 CloudTrail 이벤트를 지속적으로 배포하도록 할 수 있습니다. 추적을 구성하지 않은 경우 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수도 있습니다. CloudTrail을 사용하여 IAM 또는 AWS STS에 요청한 내용의 정보를 얻을 수 있습니다. 예를 들어 어떤 IP 주소에서 요청했는지, 누가 요청했는지, 언제 생성되었는지 등 추가적인 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 [AWS CloudTrail User Guide](#)을 참조하십시오.

주제

- [CloudTrail의 IAM 및 AWS STS 정보 \(p. 294\)](#)
- [CloudTrail 파일에 로깅된 이벤트 예제 \(p. 296\)](#)
- [CloudTrail의 중복 로그 항목 방지 \(p. 302\)](#)

CloudTrail의 IAM 및 AWS STS 정보

CloudTrail은 계정 생성 시 AWS 계정에서 활성화됩니다. IAM 또는 AWS STS에서 활동이 수행되면 해당 활동은 이벤트 기록에서 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 [CloudTrail 이벤트 기록에서 이벤트 보기](#)를 참조하십시오.

IAM 및 AWS STS 이벤트를 비롯하여 AWS 계정의 이벤트 기록을 보유하려면 추적을 생성하십시오. 추적은 CloudTrail이 Amazon S3 버킷으로 로그 파일을 전송할 수 있도록 합니다. 콘솔에서 추적을 생성하면 기본적으로 모든 리전에 추적이 적용됩니다. 추적은 AWS 파티션에 있는 모든 리전의 이벤트를 로깅하고 지정된 Amazon S3 버킷으로 로그 파일을 전송합니다. 또는 CloudTrail 로그에서 수집된 이벤트 데이터를 추가 분석 및 처리하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음을 참조하십시오.

- [추적 생성 개요](#)
- [CloudTrail 지원 서비스 및 통합](#)
- [CloudTrail에 대한 Amazon SNS 알림 구성](#)
- [여러 리전에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기](#)

모든 IAM 및 AWS STS 작업은 CloudTrail에서 로깅되고 [IAM API Reference](#) 및 [AWS Security Token Service API Reference](#)에 기록됩니다. IAM 정보는 다음과 같은 방법으로 CloudTrail에 로깅됩니다.

- IAM 및 AWS Security Token Service(AWS STS)에 대한 API 요청 – CloudTrail은 인증된 모든 API 요청(자격 증명을 통한)을 IAM 및 AWS STS API 작업으로 기록합니다. CloudTrail은 또한 AWS STS 작업, `AssumeRoleWithSAML` 및 `AssumeRoleWithWebIdentity`에 대한 미인증 요청을 자격 증명 제공자가 제공한 정보와 함께 기록합니다. 이 정보를 사용하여 위임된 역할을 지닌 연동 사용자의 호출을 외부 연동 호출자에 다시 매핑할 수 있습니다. `AssumeRole`의 경우, 호출을 원래 AWS 서비스 또는 원래 사용자의 계정에 다시 매핑할 수 있습니다. CloudTrail 로그 항목에 있는 JSON 데이터의 `userIdentity` 섹션에는 특정 연동 사용자에게 `AssumeRole*` 요청을 매핑하는 데 필요한 정보가 들어 있습니다. 자세한 내용은 AWS CloudTrail User Guide의 [CloudTrail userIdentity 요소](#)를 참조하십시오.

예를 들어 IAM `CreateUser`, `DeleteRole`, `ListGroups` 및 기타 API 작업에 대한 호출은 모두 CloudTrail에 로깅됩니다.

이러한 유형의 로그 항목에 대한 예시는 이번 주제의 뒷 부분에 제시됩니다.

Important

임의 리전에서 기본 글로벌 앤드포인트가 아닌 AWS STS 앤드포인트를 활성화할 경우에는 해당 리전의 CloudTrail 로깅 기능도 함께 활성화합니다. 이것은 해당 리전에서 수행된 AWS STS API 호출을 기록하는 데 필요합니다. 자세한 정보는 AWS CloudTrail User Guide의 [Turning On CloudTrail in Additional Regions](#)([추가 리전에서 CloudTrail 설정](#)) 단원을 참조하십시오.

- 다른 AWS 서비스에 대한 API 요청 – 다른 AWS 서비스 API 작업에 대해 인증된 요청은 CloudTrail에 로깅되며, 이 로그 항목에는 요청자에 대한 정보가 저장됩니다.

예를 들어 Amazon EC2 인스턴스 나열 요청을 했거나 CodeDeploy 배포 그룹을 생성했다고 가정해 보십시오. 요청한 사람이나 서비스에 대한 세부 내용은 그 요청의 로그 항목에 들어 있습니다. 이 정보로 AWS 계정 루트 사용자, IAM 사용자, 역할, 또는 다른 AWS 서비스에 의한 요청인지 판단할 수 있습니다.

CloudTrail 로그 항목의 사용자 자격 증명 정보에 대한 자세한 정보는 AWS CloudTrail User Guide의 [userIdentity Element](#) 단원을 참조하십시오.

- AWS 로그인 이벤트 – AWS Management 콘솔, AWS 토큰 포럼, 그리고 AWS Marketplace에 대한 로그인 이벤트는 CloudTrail에 로깅됩니다.

예를 들어 IAM 및 연동 사용자 로그인 이벤트—로그인 성공 횟수와 실패 횟수—가 CloudTrail에 로깅됩니다. 그 밖에 루트 사용자의 로그인 성공 이벤트 역시 CloudTrail에 로깅됩니다. 단, 루트 사용자의 로그인 실패 이벤트는 CloudTrail에 로깅되지 않습니다.

CloudTrail에서 로그인 이벤트를 사용자 로그에 기록하도록 설정할 경우 CloudTrail에서 이벤트를 기록할 위치를 어떻게 선택하는지 잘 이해할 필요가 있습니다.

- 사용자가 콘솔에 직접 로그인하는 경우, 선택한 서비스 콘솔이 리전을 지원하는지 여부를 기준으로 글로벌 또는 리전 로그인 앤드포인트로 리디렉션됩니다. 예를 들어 메인 콘솔 홈 페이지가 리전을 지원하여 <https://alias.signin.aws.amazon.com/console>에 로그인할 경우, <https://useast-2.signin.aws.amazon.com> 같은 리전 로그인 앤드포인트로 리디렉션됩니다. 이 리디렉션은 사용자의 리전 로그에 리전별 CloudTrail 로그 항목을 생성합니다.

반면 Amazon S3 콘솔은 리전을 지원하지 않으므로 <https://alias.signin.aws.amazon.com/console/s3>에 로그인할 경우 AWS에서 글로벌 로그인 앤드포인트 <https://signin.aws.amazon.com>으로 리디렉션합니다. 이 리디렉션은 글로벌 CloudTrail 로그 항목을 생성합니다.

- <https://alias.signin.aws.amazon.com/console?region=ap-southeast-1> 같은 URL을 사용하여 리전이 활성화된 메인 콘솔 홈 페이지에 로그인하면 특정 리전 로그인 앤드포인트를 수동으로 요청할 수 있습니다. 이 경우 AWS가 사용자를 ap-southeast-1 리전 로그인 앤드포인트로 리디렉션하고 리전 CloudTrail 로그 이벤트가 발생합니다.

Important

보안 모범 사례의 일환으로 AWS는 잘못된 사용자 이름으로 인해 로그인에 실패 하더라도 입력한 사용자 이름 텍스트를 로깅하지 않습니다. 사용자 이름 텍스트는 HIDDEN_DUE_TO_SECURITY_REASONS 값으로 마스킹 처리됩니다. 마스킹 처리의 예는 이번 주 제 후반부의 [잘못된 사용자 이름으로 인한 로그인 실패 이벤트 \(p. 301\)](#) 단원을 참조하십시오. 사용자 이름이 마스킹 처리되는 이유는 다음과 같은 사용자 오류로 인한 로그인 실패를 로깅할 경우 잠재적으로 민감한 정보가 노출될 수 있기 때문입니다.

- 우발적으로 사용자 이름 상자에 암호를 입력한 경우
- AWS 계정의 로그인 페이지 링크를 선택하고서 다른 계정의 계정 번호를 입력하는 경우
- 로그인하려는 계정을 잊고 우발적으로 개인 이메일 계정의 사용자 이름, 금융 서비스 로그인 식별자, 또는 기타 프라이빗 ID를 입력하는 경우

로그인 이벤트가 리전 또는 글로벌 이벤트인지 여부는 사용자가 로그인하는 콘솔과 사용자가 로그인 URL을 구성하는 방식에 따라 다릅니다.

- 서비스 콘솔이 리전화되어 있습니까? 그럴 경우 로그인 요청이 리전 로그인 종단점으로 자동으로 리디렉션되고 이벤트가 해당 리전의 CloudTrail 로그에 기록됩니다. 예를 들어 리전화된 <https://alias.signin.aws.amazon.com/console>에 로그인할 경우, <https://useast-2.signin.aws.amazon.com> 같은 리전 로그인 앤드포인트로 자동으로 리디렉션됩니다. 이벤트는 해당 리전의 로그에 기록됩니다.

하지만 일부 서비스는 아직 리전화되어 있지 않습니다. 예를 들어 Amazon S3 서비스는 현재 리전화되지 않았기 때문에 <https://alias.signin.aws.amazon.com/console/s3>에 로그인하면 글로벌 로그인 앤드포인트 <https://signin.aws.amazon.com>에 리디렉션됩니다. 이 리디렉션은 글로벌 로그에 이벤트를 생성합니다.

- 또한 <https://alias.signin.aws.amazon.com/console?region=ap-southeast-1> 같은 URL을 사용하여 특정 리전의 로그인 앤드포인트로 수동으로 요청할 수 있는데 이 경우 ap-southeast-1 리전 로그인 앤드포인트로 리디렉션됩니다. 이 리디렉션은 리전 로그에 이벤트를 생성합니다.
- 임시 자격 증명 요청이 로그에 기록되는 방법 – 보안 주체가 임시 자격 증명을 요청할 때 보안 주체 유형에 의해 CloudTrail에서 이벤트를 로그에 기록하는 방법이 결정됩니다. 다음 표는 임시 자격 증명을 생성하는 각 API 호출에 대해 CloudTrail에서 다양한 정보를 로그에 기록하는 방법을 보여줍니다.

보안 주체 유형	IAM/STS API	호출 계정에 대한 CloudTrail 로그의 사용자 자격 증명	역할 소유 계정에 대한 CloudTrail 로그의 사용자 자격 증명	이후의 API 호출의 경우 역할 소유자에 대한 CloudTrail 로그의 사용자 자격 증명
AWS 계정 루트 사용자 자격 증명	GetSessionToken	루트 자격 증명	역할 소유자 계정은 호출 계정과 동일	루트 자격 증명
IAM user	GetSessionToken	IAM 사용자 자격 증명	역할 소유자 계정은 호출 계정과 동일	IAM 사용자 자격 증명
IAM user	GetFederationToken	IAM 사용자 자격 증명	역할 소유자 계정은 호출 계정과 동일	IAM 사용자 자격 증명
IAM user	AssumeRole	IAM 사용자 자격 증명	계정 번호 및 보안 주체 ID(사용자인 경우) 또는 AWS 서비스 보안 주체	역할 자격 증명만(사용자 없음)
외부에서 인증된 사용자	AssumeRoleWithSAMLeIdentity	해당 사항 없음	SAML 사용자 자격 증명	역할 자격 증명만(사용자 없음)
외부에서 인증된 사용자	AssumeRoleWithWebIdentity	해당 사항 없음	OIDC/웹 사용자 자격 증명	역할 자격 증명만(사용자 없음)

CloudTrail 파일에 로깅된 이벤트 예제

CloudTrail 로그 파일에는 JSON 형식의 이벤트 정보가 포함되어 있습니다. 여기서 이벤트란 단일 API 요청이나 로그인 이벤트를 의미하여 요청된 작업, 파라미터, 그리고 작업 일시에 대한 정보가 저장됩니다.

CloudTrail 로그 파일의 IAM API 이벤트

다음은 IAM GetUserPolicy 작업 요청에 대한 CloudTrail 로그 항목 예제입니다.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::444455556666:user/Alice",
    "accountId": "444455556666",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "userName": "Alice",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2014-07-15T21:39:40Z"
      }
    },
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2014-07-15T21:40:14Z",
  "eventSource": "iam.amazonaws.com",
  "awsRegion": "us-east-1",
  "eventName": "GetUserPolicy",
  "requestParameters": {
    "policyName": "test"
  },
  "responseElements": {
    "policyDocument": "..."
  },
  "resources": [
    {
      "ARN": "arn:aws:iam::444455556666:policy/test"
    }
  ],
  "resourcesSummary": {
    "count": 1
  }
}
```

```
"eventName": " GetUserPolicy",
"awsRegion": "us-east-2",
"sourceIPAddress": "signin.amazonaws.com",
"userAgent": "signin.amazonaws.com",
"requestParameters": {
    "userName": "Alice",
    "policyName": "ReadOnlyAccess-Alice-201407151307"
},
"responseElements": null,
"requestID": "9EXAMPLE-0c68-11e4-a24e-d5e16EXAMPLE",
"eventID": "cEXAMPLE-127e-4632-980d-505a4EXAMPLE"
}
```

위 이벤트 정보에서 ReadOnlyAccess-Alice-201407151307 요소에 지정한 것처럼 사용자 Alice을 의미하는 requestParameters이라는 이름의 사용자 정책을 가져오는 요청인 것을 알 수 있습니다. 그 밖에도 요청자는 Alice라는 이름의 IAM 사용자이고, 2014년 7월 15일 오후 9시 40분(UTC)에 생성된 것도 확인 가능합니다. 여기서는 userAgent 요소를 통해 요청이 AWS Management 콘솔에서 이루어진 것도 알 수 있습니다.

CloudTrail 로그 파일의 AWS STS API 이벤트

777788889999 계정의 "Bob"이라는 IAM 사용자는 AWS STS AssumeRole 작업을 호출하여 111122223333 계정의 EC2-dev 역할을 맡습니다. 다음 예시 두 개는 가계정 두 개에 대한 CloudTrail 로그 항목입니다. 첫 번째 예시는 AssumeRole을 호출하는 사용자를 소유한 계정인 777788889999 계정에서 보낸 요청에 대한 CloudTrail 로그 항목입니다.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAQRSTUVWXYZEXAMPLE",
    "arn": "arn:aws:iam::777788889999:user/Bob",
    "accountId": "777788889999",
    "accessKeyId": "AKIAQRSTUVWXYZEXAMPLE",
    "userName": "Bob"
  },
  "eventTime": "2014-07-18T15:07:39Z",
  "eventSource": "sts.amazonaws.com",
  "eventName": "AssumeRole",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.101",
  "userAgent": "aws-cli/1.11.10 Python/2.7.8 Linux/3.2.45-0.6.wd.865.49.315.metal1.x86_64 botocore/1.4.67",
  "requestParameters": {
    "roleArn": "arn:aws:iam::111122223333:role/EC2-dev",
    "roleSessionName": "Bob-EC2-dev",
    "serialNumber": "arn:aws:iam::777788889999:mfa"
  },
  "responseElements": {
    "credentials": {
      "sessionToken": "<encoded session token blob>",
      "accessKeyId": "AKIAQRSTUVWXYZEXAMPLE",
      "expiration": "Jul 18, 2014 4:07:39 PM"
    },
    "assumedRoleUser": {
      "assumedRoleId": "AIDAQRSTUVWXYZEXAMPLE:Bob-EC2-dev",
      "arn": "arn:aws:sts::111122223333:assumed-role/EC2-dev/Bob-EC2-dev"
    }
  },
  "resources": [
    {
      "ARN": "arn:aws:iam::111122223333:role/EC2-dev",
      "accountId": "111122223333",
      "region": "us-east-2"
    }
  ]
}
```

```
        "type": "AWS::IAM::Role"
    }
],
"requestID": "4EXAMPLE-0e8d-11e4-96e4-e55c0EXAMPLE",
"sharedEventID": "bEXAMPLE-efea-4a70-b951-19a88EXAMPLE",
"eventID": "dEXAMPLE-ac7f-466c-a608-4ac8dEXAMPLE"
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}
```

두 번째 예시는 동일한 요청에 대한 역할 소유 계정(111122223333)의 CloudTrail 로그 항목입니다.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AWSAccount",
    "principalId": "AIDAQRSTUVWXYZEXAMPLE",
    "accountId": "777788889999"
  },
  "eventTime": "2014-07-18T15:07:39Z",
  "eventSource": "sts.amazonaws.com",
  "eventName": "AssumeRole",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.101",
  "userAgent": "aws-cli/1.11.10 Python/2.7.8 Linux/3.2.45-0.6.wd.865.49.315.metal1.x86_64 botocore/1.4.67",
  "requestParameters": {
    "roleArn": "arn:aws:iam::111122223333:role/EC2-dev",
    "roleSessionName": "Bob-EC2-dev",
    "serialNumber": "arn:aws:iam::777788889999:mfa"
  },
  "responseElements": {
    "credentials": {
      "sessionToken": "<encoded session token blob>",
      "accessKeyId": "AKIAQRSTUVWXYZEXAMPLE",
      "expiration": "Jul 18, 2014 4:07:39 PM"
    },
    "assumedRoleUser": {
      "assumedRoleId": "AIDAQRSTUVWXYZEXAMPLE:Bob-EC2-dev",
      "arn": "arn:aws:sts::111122223333:assumed-role/EC2-dev/Bob-EC2-dev"
    }
  },
  "requestID": "4EXAMPLE-0e8d-11e4-96e4-e55c0EXAMPLE",
  "sharedEventID": "bEXAMPLE-efea-4a70-b951-19a88EXAMPLE",
  "eventID": "dEXAMPLE-ac7f-466c-a608-4ac8dEXAMPLE"
}
```

다음 예시는 IAM 역할에서 권한을 사용하는 API를 호출하는 AWS 서비스의 요청에 대한 CloudTrail 로그 항목입니다.

```
{
  "eventVersion": "1.04",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDAQRSTUVWXYZEXAMPLE:devdsk",
    "arn": "arn:aws:sts::777788889999:assumed-role/AssumeNothing/devdsk",
    "accountId": "777788889999",
    "accessKeyId": "AKIAQRSTUVWXYZEXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2016-11-14T17:25:26Z"
      }
    }
  }
```

```
"sessionIssuer": {
    "type": "Role",
    "principalId": "AIDAQRSTUVWXYZEXAMPLE",
    "arn": "arn:aws:iam::777788889999:role/AssumeNothing",
    "accountId": "777788889999",
    "userName": "AssumeNothing"
}
},
"eventTime": "2016-11-14T17:25:45Z",
"eventSource": "s3.amazonaws.com",
"eventName": "DeleteBucket",
"awsRegion": "us-east-2",
"sourceIPAddress": "192.0.2.1",
"userAgent": "[aws-cli/1.11.10 Python/2.7.8 Linux/3.2.45-0.6.wd.865.49.315.metal1.x86_64 botocore/1.4.67]",
"requestParameters": {
    "bucketName": "my-test-bucket-cross-account"
},
"responseElements": null,
"requestID": "EXAMPLE463D56D4C",
"eventID": "dEXAMPLE-265a-41e0-9352-4401bEXAMPLE",
"eventType": "AwsApiCall",
"recipientAccountId": "777788889999"
}
```

다음은 AWS STS AssumeRoleWithSAML 작업 요청에 대한 CloudTrail 로그 항목 예제입니다.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "SAMLUser",
        "principalId": "<id of identity provider>:<canonical id of user>",
        "userName": "<canonical id of user>",
        "identityProvider": "<id of identity provider>"
    },
    "eventTime": "2016-03-23T01:39:57Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRoleWithSAML",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "192.0.2.101",
    "userAgent": "aws-cli/1.3.23 Python/2.7.6 Linux/2.6.18-164.el5",
    "requestParameters": {
        "sAMLAssertionID": "_c0046cEXAMPLEb9d4b8eEXAMPLE2619aEXAMPLE",
        "roleSessionName": "MyAssignedRoleSessionName",
        "durationSeconds": 3600,
        "roleArn": "arn:aws:iam::444455556666:role/SAMLTestRoleShibboleth",
        "principalArn": "arn:aws:iam::444455556666:saml-provider/Shibboleth"
    },
    "responseElements": {
        "subjectType": "transient",
        "issuer": "https://server.example.com/idp/shibboleth",
        "credentials": {
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "expiration": "Mar 23, 2016 2:39:57 AM",
            "sessionToken": "<encoded session token blob>"
        },
        "nameQualifier": "<id of identity provider>",
        "assumedRoleUser": {
            "assumedRoleId": "AROAD35QRSTUVWEXAMPLE:MyAssignedRoleSessionName",
            "arn": "arn:aws:sts::444455556666:assumed-role/SAMLTestRoleShibboleth/MyAssignedRoleSessionName"
        },
        "subject": "<canonical id of user>",
        "audience": "https://signin.aws.amazon.com/saml"
    }
}
```

```
        },
        "resources": [
            {
                "ARN": "arn:aws:iam::444455556666:role/SAMLTestRoleShibboleth",
                "accountId": "444455556666",
                "type": "AWS::IAM::Role"
            },
            {
                "ARN": "arn:aws:iam::444455556666:saml-provider/test-saml-provider",
                "accountId": "444455556666",
                "type": "AWS::IAM::SAMLProvider"
            }
        ],
        "requestID": "6EXAMPLE-e595-11e5-b2c7-c974fEXAMPLE",
        "eventID": "dEXAMPLE-265a-41e0-9352-4401bEXAMPLE",
        "eventType": "AwsApiCall",
        "recipientAccountId": "444455556666"
    }
}
```

다음은 AWS STS AssumeRoleWithWebIdentity 작업 요청에 대한 CloudTrail 로그 항목 예제입니다.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "WebIdentityUser",
        "principalId": "accounts.google.com:<id-of-application>.apps.googleusercontent.com:<id-of-user>",
        "userName": "<id of user>",
        "identityProvider": "accounts.google.com"
    },
    "eventTime": "2016-03-23T01:39:51Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRoleWithWebIdentity",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "192.0.2.101",
    "userAgent": "aws-cli/1.3.23 Python/2.7.6 Linux/2.6.18-164.el5",
    "requestParameters": {
        "durationSeconds": 3600,
        "roleArn": "arn:aws:iam::444455556666:role/FederatedWebIdentityRole",
        "roleSessionName": "MyAssignedRoleSessionName"
    },
    "responseElements": {
        "provider": "accounts.google.com",
        "subjectFromWebIdentityToken": "<id of user>",
        "audience": "<id of application>.apps.googleusercontent.com",
        "credentials": {
            "accessKeyId": "ASIAACQRSTUVWRRAOEXAMPLE",
            "expiration": "Mar 23, 2016 2:39:51 AM",
            "sessionToken": "<encoded session token blob>"
        },
        "assumedRoleUser": {
            "assumedRoleId": "AROACQRSTUVWRRAOEXAMPLE:MyAssignedRoleSessionName",
            "arn": "arn:aws:sts::444455556666:assumed-role/FederatedWebIdentityRole/MyAssignedRoleSessionName"
        }
    },
    "resources": [
        {
            "ARN": "arn:aws:iam::444455556666:role/FederatedWebIdentityRole",
            "accountId": "444455556666",
            "type": "AWS::IAM::Role"
        }
    ],
    "requestID": "6EXAMPLE-e595-11e5-b2c7-c974fEXAMPLE",
    "eventID": "bEXAMPLE-0b30-4246-b28c-e3da3EXAMPLE",
    "eventType": "AwsApiCall"
}
```

```
    "eventType": "AwsApiCall",
    "recipientAccountId": "444455556666"
}
```

CloudTrail 로그 파일의 로그인 실패 이벤트

다음은 실패한 로그인 이벤트에 대한 CloudTrail 로그 항목을 나타낸 예제입니다.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "userName": "Alice"
  },
  "eventTime": "2014-07-08T17:35:27Z",
  "eventSource": "signin.amazonaws.com",
  "eventName": "ConsoleLogin",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.100",
  "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:24.0) Gecko/20100101 Firefox/24.0",
  "errorMessage": "Failed authentication",
  "requestParameters": null,
  "responseElements": {
    "ConsoleLogin": "Failure"
  },
  "additionalEventData": {
    "MobileVersion": "No",
    "LoginTo": "https://console.aws.amazon.com/sns",
    "MFAUsed": "No"
  },
  "eventID": "11ea990b-4678-4bcd-8fbe-62509088b7cf"
}
```

위 이벤트 정보에서 `userIdentity` 요소에도 나와 있듯이 Alice라는 이름의 IAM 사용자가 로그인을 시도한 것을 알 수 있습니다. 또한 `responseElements` 요소를 보면 로그인 시도가 실패한 것도 확인됩니다. 그리고 Alice가 Amazon SNS 콘솔에 로그인하려고 시도한 일시는 2014년 7월 8일 오후 5시 35분(UTC)입니다.

잘못된 사용자 이름으로 인한 로그인 실패 이벤트

다음은 잘못된 사용자 이름을 입력하여 로그인을 실패한 이벤트의 CloudTrail 로그 항목을 나타낸 예제입니다. 이때 AWS는 `userName` 텍스트를 `HIDDEN_DUE_TO_SECURITY_REASONS`로 마스킹 처리하여 잠재적으로 민감한 정보의 노출을 차단합니다.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "accountId": "123456789012",
    "accessKeyId": "",
    "userName": "HIDDEN_DUE_TO_SECURITY_REASONS"
  },
  "eventTime": "2015-03-31T22:20:42Z",
  "eventSource": "signin.amazonaws.com",
  "eventName": "ConsoleLogin",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.101",
  "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:24.0) Gecko/20100101 Firefox/24.0",
  "errorMessage": "No username found in supplied account",
}
```

```
"requestParameters": null,  
"responseElements": {  
    "ConsoleLogin": "Failure"  
},  
"additionalEventData": {  
    "LoginTo": "https://console.aws.amazon.com/console/home?state=hashArgs%23&isauthcode=true",  
    "MobileVersion": "No",  
    "MFAUsed": "No"  
},  
"eventID": "a7654656-0417-45c6-9386-ea8231385051",  
"eventType": "AwsConsoleSignin",  
"recipientAccountId": "123456789012"  
}
```

CloudTrail 로그 파일의 로그인 성공 이벤트

다음은 성공한 로그인 이벤트에 대한 CloudTrail 로그 항목을 나타낸 예제입니다.

```
{  
    "eventVersion": "1.05",  
    "userIdentity": {  
        "type": "IAMUser",  
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",  
        "arn": "arn:aws:iam::111122223333:user/Bob",  
        "accountId": "111122223333",  
        "userName": "Bob"  
    },  
    "eventTime": "2014-07-16T15:49:27Z",  
    "eventSource": "signin.amazonaws.com",  
    "eventName": "ConsoleLogin",  
    "awsRegion": "us-east-2",  
    "sourceIPAddress": "192.0.2.110",  
    "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:24.0) Gecko/20100101 Firefox/24.0",  
    "requestParameters": null,  
    "responseElements": {  
        "ConsoleLogin": "Success"  
    },  
    "additionalEventData": {  
        "MobileVersion": "No",  
        "LoginTo": "https://console.aws.amazon.com/s3",  
        "MFAUsed": "No"  
    },  
    "eventID": "3fcfb182-98f8-4744-bd45-10a395ab61cb"  
}
```

CloudTrail 로그 파일에 저장된 정보에 대한 자세한 정보는 AWS CloudTrail User Guide의 [CloudTrail 이벤트 참조](#) 단원을 참조하십시오.

CloudTrail의 종복 로그 항목 방지

CloudTrail은 각 리전마다 별도로 추적 파일을 생성합니다. 이 추적 파일에는 각 리전에서 발생하는 이벤트 정보를 비롯해 IAM API 호출, 리전이 아닌 다른 기준의 AWS STS 호출(sts.amazonaws.com 호출 등), 그리고 AWS 로그인 이벤트 같은 글로벌(리전이 아닌 다른 기준의) 이벤트 정보까지 저장됩니다. 예를 들어, 한 리전에 두 개의 추적이 있다고 가정하겠습니다. 그런 다음 IAM 사용자를 새로 생성할 경우 CreateUser 이벤트가 두 리전의 로그 파일에 추가되어 로그 항목이 종복되고 맙니다.

AWS Security Token Service(STS)는 <https://sts.amazonaws.com>에 단일 엔드포인트가 있는 전역적 서비스입니다. 따라서 이 엔드포인트에 대한 호출은 글로벌 서비스에 대한 호출로 로깅됩니다. 하지만 이 엔드포인트가 둘리적으로 미국 동부(버지니아 북부) 리전에 위치하기 때문에 로그가 표시되는 이벤트 리전 역시 us-east-1이 됩니다. 이때는 해당 리전에 글로벌 서비스 로그를 추가하도록 선택해야만 CloudTrail이 이 로그를

미국 동부(오하이오) 리전에 기록합니다. 그러면 CloudTrail이 모든 리전의 엔드포인트 호출을 각 리전으로 기록합니다. 이를테면 sts.us-east-2.amazonaws.com 호출은 미국 동부(오하이오) 리전에 게시되고, sts.eu-central-1.amazonaws.com 호출은 EU(프랑크푸르트) 리전에 게시되는 등의 방식을 따릅니다.

여러 리전 및 AWS STS에 대한 자세한 정보는 [AWS 리전에서 AWS STS 활성화 및 비활성화 \(p. 287\)](#) 단원을 참조하십시오.

아래는 각 리전을 비롯해 리전에 따른 CloudTrail의 AWS STS 요청 로깅 방식을 나타낸 표입니다. "위치" 열은 CloudTrail이 기록하는 로그를 나타냅니다. "글로벌"은 글로벌 서비스 로그를 추가하도록 선택한 모든 리전에 이벤트가 로깅된다는 것을 의미합니다. 그리고, "리전"은 엔드포인트가 위치한 리전에만 이벤트가 로깅된다는 것을 의미합니다. 마지막 열은 로그 항목에서 요청 리전의 식별 방식을 나타냅니다.

리전 이름	CloudTrail 로그의 리 전 자격 증명	엔드포인트	CloudTrail 로그 위치
해당 사항 없음 - 글로벌	us-east-1	sts.amazonaws.com	전 세계
미국 동부(오하이오)	us-east-2	sts.us-east-2.amazonaws.com	리전
미국 동부(버지니아 북부)	us-east-1	sts.us-east-1.amazonaws.com	리전
미국 서부(캘리포니아 북부 지역)	us-west-1	sts.us-west-1.amazonaws.com	리전
미국 서부(오레곤)	us-west-2	sts.us-west-2.amazonaws.com	리전
캐나다(중부)	ca-central-1	sts.ca-central-1.amazonaws.com	리전
EU(프랑크푸르트)	eu-central-1	sts.eu-central-1.amazonaws.com	리전
EU(아일랜드)	eu-west-1	sts.eu-west-1.amazonaws.com	리전
EU(런던)	eu-west-2	sts.eu-west-2.amazonaws.com	리전
아시아 태평양(도쿄)	ap-northeast-1	sts.ap-northeast-1.amazonaws.com	리전
아시아 태평양(서울)	ap-northeast-2	sts.ap-northeast-2.amazonaws.com	리전
아시아 태평양(뭄바이)	ap-south-1	sts.ap-south-1.amazonaws.com	리전
아시아 태평양(싱가포르)	ap-southeast-1	sts.ap-southeast-1.amazonaws.com	리전
아시아 태평양(시드니)	ap-southeast-2	sts.ap-southeast-2.amazonaws.com	리전
남아메리카(상파울루)	sa-east-1	sts.sa-east-1.amazonaws.com	리전

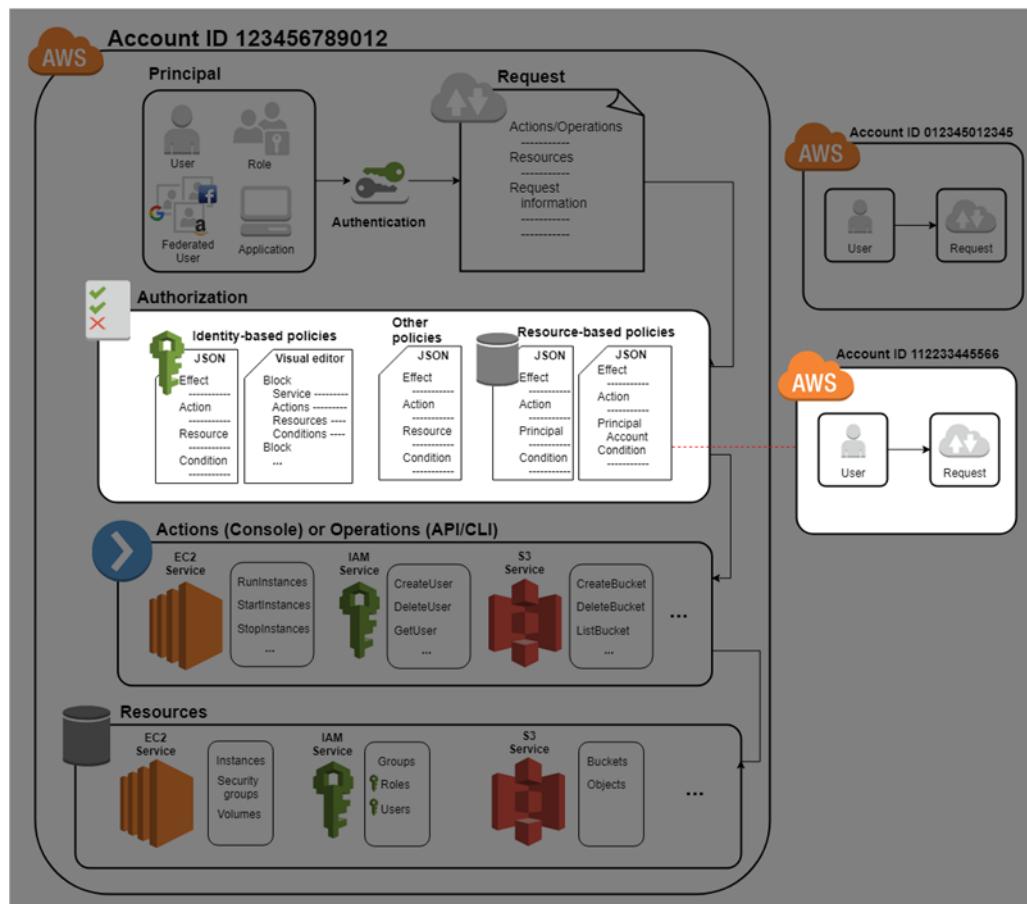
계정 내 다양한 리전의 추적 정보를 단일 Amazon S3 버킷으로 통합하도록 CloudTrail을 구성할 경우, IAM 이벤트가 로그에 중복 저장됩니다. 즉, 각 리전의 추적 파일이 동일한 IAM 이벤트를 통합 로그에 기록합니다. 이러한 중복 문제를 해결하기 위해 글로벌 이벤트를 선택적으로 추가할 수 있습니다. 일반적인 방법은 한 추적 파일에서는 글로벌 이벤트를 활성화하고, 동일한 Amazon S3 버킷에 기록하는 다른 모든 추적 파일에서는 글로벌 이벤트를 비활성화하는 것입니다. 이렇게 하면 글로벌 이벤트는 항상 한 곳에만 기록됩니다.

자세한 정보는 AWS CloudTrail User Guide의 [Aggregating Logs](#) 단원을 참조하십시오.

액세스 관리

AWS Identity and Access Management(IAM)은 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있는 웹 서비스입니다. 보안 주체 (p. 4)가 AWS에 요청하면 AWS 적용 코드는 해당 보안 주체가 인증(로그인) 및 권한 부여(권한 있음)되었는지 확인합니다. 정책을 생성하고 IAM 자격 증명 또는 AWS 리소스에 연결하여 AWS 액세스를 관리합니다. 정책은 자격 증명 또는 리소스에 연결될 때 해당 권한을 정의하는 AWS의 JSON 정책 문서입니다. 정책 유형 및 활용에 대한 자세한 정보는 [정책 및 권한 \(p. 305\)](#) 단원을 참조하십시오.

인증 및 권한 부여 프로세스의 나머지 부분에 대한 자세한 정보는 [IAM 작동 방식 이해 \(p. 3\)](#) 단원을 참조하십시오.



권한 부여 중 AWS 적용 코드는 [요청 콘텍스트 \(p. 5\)](#)의 값을 사용하여 일치하는 정책을 확인하고 요청을 허용할지 거부할지 여부를 결정합니다.

AWS는 요청 콘텍스트에 적용되는 각 정책을 확인합니다. 단일 정책이 요청을 거부한 경우 AWS는 전체 요청을 거부하고 정책 평가를 중지합니다. 이를 명시적 거부라고 합니다. 요청은 기본적으로 거부되므로 IAM은 사용 가능한 정책이 요청의 모든 부분을 허용하는 경우에만 요청에 권한을 부여합니다. 단일 계정 내 요청 평가 로직 (p. 531)은 다음 규칙을 따릅니다.

- 기본적으로 모든 요청이 목시적으로 거부됩니다. 또는 기본적으로 AWS 계정 루트 사용자에 모든 권한이 부여됩니다.
- 자격 증명 기반 또는 리소스 기반 정책에 포함된 명시적 허용은 이 기본 작동을 재정의합니다.
- 권한 경계, 조직 SCP 또는 세션 정책이 있는 경우 이러한 정책 유형이 명시적 거부로 허용을 재정의할 수도 있습니다.

- 어떠한 정책의 명시적 거부도 허용을 무시합니다.

요청이 인증 및 권한 부여된 후 AWS가 요청을 승인합니다. 다른 계정에서 요청해야 하는 경우 다른 계정의 정책에서 요청자에게 해당 리소스에 대한 액세스를 허용해야 합니다. 또한 요청하는 데 사용하는 IAM 엔터티에 해당 요청을 허용하는 자격 증명 기반 정책이 있어야 합니다.

액세스 관리 리소스

권한 및 정책 생성에 대한 자세한 정보는 다음 리소스를 참조하십시오.

- AWS 보안 블로그의 다음 게시물에서는 Amazon S3 버킷과 객체에 액세스하기 위한 정책을 작성하는 일반적인 방법을 소개합니다.
 - [IAM 정책 작성: Amazon S3 버킷에 대한 액세스를 허용하는 방법](#)
 - [IAM 정책 작성: Amazon S3 버킷의 사용자별 폴더에 대한 액세스 허용](#)
 - [IAM 정책 및 버킷 정책과 ACL ACL \(S3 리소스에 대한 액세스 제어\)](#)
 - [RDS 리소스 수준 권한에 대한 소개](#)
 - [EC2 리소스 수준 권한 설명](#)

정책 및 권한

정책을 생성하고 IAM 자격 증명(사용자, 사용자 그룹 또는 역할) 또는 AWS 리소스에 연결하여 AWS에서 액세스를 관리합니다. 정책은 자격 증명이나 리소스와 연결될 때 해당 권한을 정의하는 AWS의 객체입니다. AWS는 보안 주체 엔터티(사용자 또는 역할)가 요청을 보낼 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는지 여부를 결정합니다. 대부분의 정책은 AWS에 JSON 문서로 저장됩니다. AWS에서는 자격 증명 기반 정책, 리소스 기반 정책, 권한 경계, 조직 SCP, ACL 및 세션 정책이라는 6가지 정책 유형을 지원합니다.

IAM 정책은 작업을 실행하기 위한 방법과 상관없이 작업을 정의합니다. 예를 들어, 정책이 [GetUser](#) 작업을 허용한다면 이 정책이 있는 사용자는 AWS Management 콘솔, AWS CLI, 또는 AWS API에서 사용자 정보를 얻을 수 있습니다. IAM 사용자를 생성할 경우 콘솔 또는 프로그래밍 방식 액세스를 허용하도록 선택할 수 있습니다. 콘솔 액세스가 허용되는 경우 IAM 사용자는 사용자 이름 및 암호를 사용하여 콘솔에 로그인할 수 있습니다. 또는 프로그래밍 방식의 액세스가 허용되는 경우 사용자는 액세스 키를 사용하여 CLI 또는 API로 작업할 수 있습니다.

정책 유형

빈도수에 따라 나열된 다음 정책 유형은 AWS에서 사용 가능합니다. 자세한 정보는 각 정책 유형에 따른 섹션을 참조하십시오.

- [자격 증명 기반 정책 \(p. 306\)](#) – 관리형 및 인라인 정책을 IAM 자격 증명(사용자, 사용자가 속한 그룹, 또는 역할)에 연결합니다. 자격 증명 기반 정책은 자격 증명에 권한을 부여합니다.
- [리소스 기반 정책 \(p. 306\)](#) – 인라인 정책을 리소스에 연결합니다. 리소스 기반 정책의 가장 일반적인 예제는 Amazon S3 버킷 정책 및 IAM 역할 신뢰 정책입니다. 리소스 기반 정책은 정책에 지정된 보안 주체 엔터티에 권한을 부여합니다. 보안 주체는 리소스와 동일한 계정 또는 다른 계정에 있을 수 있습니다.
- [권한 경계 \(p. 306\)](#) – 관리형 정책을 IAM 엔터티(사용자 또는 역할)에 대한 권한 경계로 사용합니다. 해당 정책은 자격 증명 기반 정책을 통해 엔터티에 부여할 수 있는 최대 권한을 정의하지만, 권한을 부여하지는 않습니다. 권한 경계는 리소스 기반 정책을 통해 엔터티에 부여할 수 있는 최대 권한을 정의하지 않습니다.
- [조직 SCP \(p. 307\)](#) – AWS Organizations 서비스 제어 정책(SCP)을 사용하여 조직 또는 조직 단위(OU)의 계정 멤버에 대한 최대 권한을 정의합니다. SCP는 자격 증명 기반 정책이나 리소스 기반 정책을 통해 계정 내 엔터티(사용자나 역할)에 부여하는 권한을 제한하지만, 권한을 부여하지는 않습니다.

- **액세스 제어 목록(ACL) (p. 307)** – ACL을 사용하여 ACL이 연결된 리소스에 액세스할 수 있는 다른 계정의 보안 주체를 제어합니다. ACL은 리소스 기반 정책과 비슷합니다. 다만 JSON 정책 문서 구조를 사용하지 않은 유일한 정책 유형입니다. ACL은 지정된 보안 주체 엔터티에 권한을 부여하는 교차 계정 권한 정책입니다. ACL은 동일 계정 내 엔터티에 권한을 부여할 수 없습니다.
- **세션 정책 (p. 307)** – AWS CLI 또는 AWS API를 사용하여 역할이나 연합된 사용자를 수임할 때 고급 세션 정책을 전달합니다. 세션 정책은 역할이나 사용자의 자격 증명 기반 정책을 통해 세션에 부여하는 권한을 제한합니다. 세션 정책은 생성된 세션에 대한 권한을 제한하지 않지만, 권한을 부여하지도 않습니다. 자세한 정보는 [세션 정책](#)을 참조하십시오.

자격 증명 기반 정책

자격 증명 기반 정책은 자격 증명(사용자, 사용자 그룹 또는 역할)에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 엔터티(사용자 또는 역할)가 수행할 수 있는 작업, 작업의 대상 리소스 또는 작업 수행 조건을 제어합니다. 자격 증명 기반 정책을 추가로 분류할 수 있습니다.

- 관리형 정책 – AWS 계정에 속한 다수의 사용자, 그룹 및 역할에게 독립적으로 연결할 수 있는 자격 증명 기반 정책입니다. 사용할 수 있는 관리형 정책은 두 가지가 있습니다.
 - AWS 관리형 정책 – AWS에서 생성 및 관리하는 관리형 정책입니다. 정책 사용이 처음이라면 AWS 관리형 정책 사용을 먼저 권장합니다.
 - 고객 관리형 정책 – 사용자가 자신의 AWS 계정에서 생성 및 관리하는 관리형 정책입니다. 고객 관리형 정책은 AWS 관리형 정책보다 정책에 대해 더욱 정밀하게 제어할 수 있습니다. 시각적 편집기에서 또는 JSON 정책 문서를 직접 생성하여 IAM 정책을 생성 및 편집할 수 있습니다. 자세한 정보는 [IAM 정책 만들기 \(p. 377\)](#) 및 [IAM 정책 편집 \(p. 402\)](#)을(를) 참조하십시오.
- 인라인 정책 – 자신이 생성 및 관리하며, 단일 사용자, 그룹 또는 역할에 직접 포함되는 정책입니다. 대부분의 경우 인라인 정책을 사용하지 않는 것이 좋습니다.

관리형 정책을 사용할지 아니면 인라인 정책을 사용할지를 선택하는 방법은 [관리형 정책과 인라인 정책 \(p. 312\)](#) 단원을 참조하십시오.

리소스 기반 정책

리소스 기반 정책은 Amazon S3 버킷과 같은 리소스에 연결하는 JSON 정책 문서입니다. 이러한 정책은 지정된 보안 주체에 해당 리소스에 대한 특정 작업을 수행할 수 있는 권한을 부여하고 이러한 권한이 적용되는 조건을 정의합니다. 리소스 기반 정책은 인라인 정책입니다. 관리형 리소스 기반 정책은 없습니다.

교차 계정 액세스를 활성화하려는 경우 전체 계정이나 다른 계정의 IAM 엔터티를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 리소스 기반 정책에 교차 계정 보안 주체를 추가하는 것은 신뢰 관계 설정의 절반밖에 되지 않는다는 것을 유념하십시오. 또한 보안 주체와 리소스가 별도의 AWS 계정에 있는 경우 자격 증명 기반 정책을 사용하여 리소스에 보안 주체 엔터티 액세스를 부여해야 합니다. 하지만 리소스 기반 정책이 동일 계정의 보안 주체에 액세스를 부여하는 경우 추가 자격 증명 기반 정책이 필요하지 않습니다.

IAM 서비스는 역할 신뢰 정책이라고 하는 리소스 기반 정책 유형 하나만 지원하며, 이 유형은 IAM 역할에 연결됩니다. IAM 역할은 리소스 기반 정책을 지원하는 자격 증명이자 리소스이므로 신뢰 정책과 자격 증명 기반 정책 모두 IAM 역할에 연결해야 합니다. 신뢰 정책은 역할을 수임할 수 있는 보안 주체 엔터티(계정, 사용자, 역할 및 연합된 사용자)를 정의합니다. IAM 역할과 다른 리소스 기반 정책 간의 차이에 대해 알아보려면 [IAM 역할과 리소스 기반 정책의 차이 \(p. 257\)](#) 단원을 참조하십시오.

리소스 기반 정책을 지원하는 다른 서비스를 확인하려면 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하십시오. 리소스 기반 정책에 대해 자세히 알아보려면 [자격 증명 기반 정책 및 리소스 기반 정책 \(p. 326\)](#) 단원을 참조하십시오.

IAM 권한 경계

권한 경계는 자격 증명 기반 정책을 통해 IAM 엔터티에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 엔터티에 대한 권한 경계를 설정할 경우 해당 엔터티는 자격 증명 기반 정책 및 관련 권한 경계 모두에서

허용되는 작업만 수행할 수 있습니다. 사용자나 역할을 보안 주체로 지정하는 리소스 기반 정책은 권한 경계에 제한을 받지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 정보는 [IAM 엔터티에 대한 권한 경계 \(p. 317\)](#) 단원을 참조하십시오.

서비스 제어 정책(SCP)

AWS Organizations는 기업이 소유하는 AWS 계정을 그룹화하고 중앙에서 관리할 수 있는 서비스입니다. 조직에서 모든 기능을 활성화할 경우 서비스 제어 정책(SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 조직 또는 조직 단위(OU)에 최대 권한을 지정하는 JSON 정책입니다. SCP는 각 AWS 계정 루트 사용자를 비롯하여 멤버 계정의 엔터티에 대한 권한을 제한합니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다.

조직 및 SCP에 대한 자세한 정보는 AWS Organizations 사용 설명서의 [서비스 제어 정책 정보](#)를 참조하십시오.

액세스 제어 정책(ACL)

액세스 제어 정책(ACL)을 통해 리소스에 액세스할 수 있는 다른 계정의 보안 주체를 제어할 수 있습니다. ACL은 동일 계정 내에서 보안 주체에 대한 액세스를 제어하는 데 사용할 수 없습니다. ACL는 리소스 기반 정책과 비슷합니다. 다만 JSON 정책 문서 형식을 사용하지 않은 유일한 정책 유형입니다. Amazon S3, AWS WAF, Amazon VPC는 ACL을 지원하는 서비스의 예입니다. ACL에 대한 자세한 정보는 Amazon Simple Storage Service 개발자 가이드의 [ACL\(액세스 제어 목록\) 개요](#) 단원을 참조하십시오.

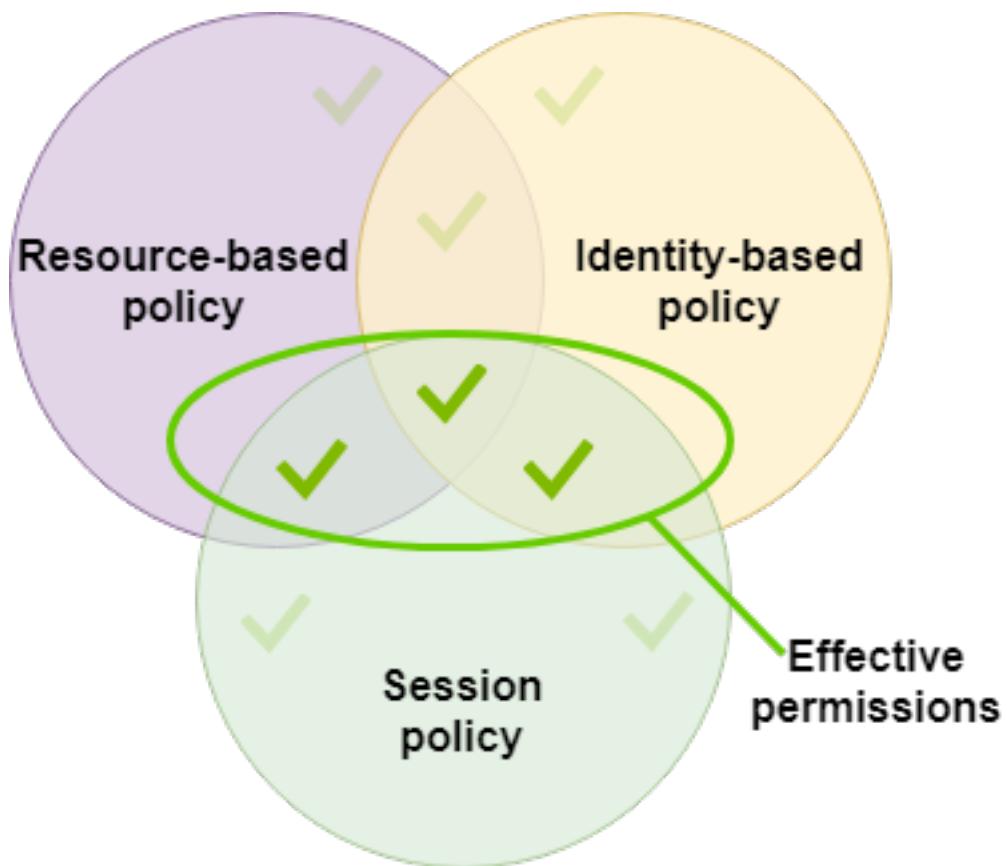
세션 정책

세션 정책은 역할 또는 연합된 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 세션에 대한 권한은 세션을 생성하는 데 사용되는 IAM 엔터티(사용자 또는 역할)에 대한 자격 증명 기반 정책과 세션 정책에서 가져옵니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다.

`AssumeRole`, `AssumeRoleWithSAML` 또는 `AssumeRoleWithWebIdentity` API 작업을 사용하여 프로그래밍 방식으로 역할 세션을 생성하고 세션 정책을 전달할 수 있습니다. 결과적으로 세션에는 역할의 자격 증명 기반 정책과 세션 정책, 이 두 정책 모두에 의해 부여되는 권한만 부여됩니다. 역할 세션 생성에 대한 자세한 정보는 [임시 보안 자격 증명 요청하기 \(p. 265\)](#) 단원을 참조하십시오.

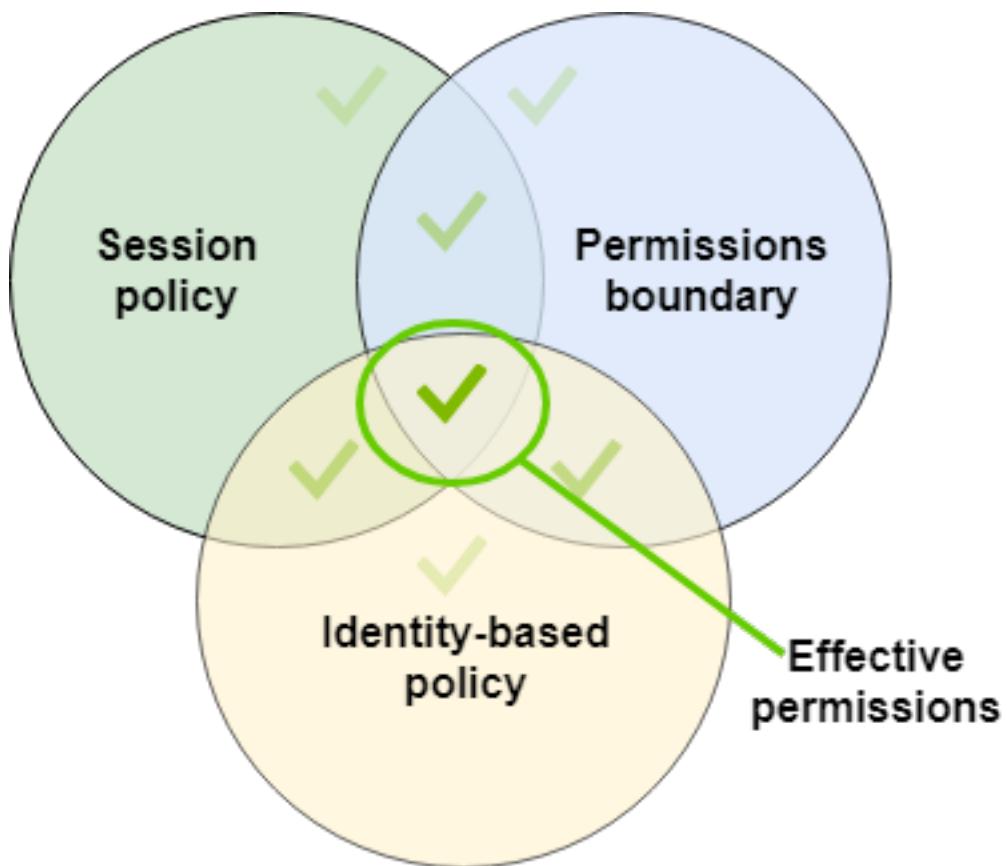
연합된 사용자 세션을 생성할 경우 IAM 사용자의 액세스 키를 사용하여 `GetFederationToken` API 작업을 프로그래밍 방식으로 호출할 수 있습니다. 이 작업을 수행하고 세션 정책을 전달할 경우 결과적으로 세션에는 IAM 사용자 자격 증명 기반 정책과 세션 정책, 이 두 정책 모두에 의해 부여되는 권한만 부여됩니다. 연합된 사용자 생성에 대한 자세한 정보는 [GetFederationToken—사용자 지정 자격 증명 브로커를 통한 연동 \(p. 269\)](#) 단원을 참조하십시오.

리소스 기반 정책이 사용자 또는 역할의 ARN을 보안 주체로 지정할 경우 세션이 생성되기 전에 리소스 기반 정책의 권한이 역할 또는 사용자의 자격 증명 기반 정책에 추가됩니다. 이 세션 정책은 리소스 기반 정책 및 자격 증명 기반 정책을 통해 부여되는 모든 권한을 제한합니다. 결과적으로 세션에는 세션 정책과 리소스 기반 정책이나 자격 증명 기반 정책의 권한이 부여됩니다.



리소스 기반 정책이 세션의 ARN을 보안 주체로 지정하는 경우 세션이 생성된 후 리소스 기반 정책의 권한이 추가됩니다. 리소스 기반 정책 권한은 세션 정책에 제한을 받지 않습니다. 결과적으로 세션에는 리소스 기반 정책의 모든 권한 + 자격 증명 기반 정책과 세션 정책에 의해 부여되는 권한이 부여됩니다.

권한 경계에서 세션 생성에 사용되는 사용자 또는 역할에 대해 최대 권한을 설정할 경우 결과적으로 세션에는 세션 정책, 권한 경계 및 자격 증명 기반 정책의 권한만 부여됩니다.



정책 및 루트 사용자

AWS 계정 루트 사용자는 어떤 정책에는 영향을 받지만 이외의 정책에는 영향을 받지 않습니다. 자격 증명 기반 정책을 루트 사용자로 연결할 수 없고 루트 사용자에 대한 권한 경계를 설정할 수 없습니다. 그러나, 루트 사용자를 리소스 기반 정책 또는 ACL의 보안 주체로 지정할 수 있습니다. 계정의 멤버로서 루트 사용자는 계정의 SCP에 의해 영향을 받습니다.

JSON 정책 개요

대부분의 정책은 AWS에 JSON 문서로서 저장됩니다. 자격 증명 기반 정책, 경계를 설정할 수 있는 정책은 사용자 또는 역할에 연결할 수 있는 JSON 정책 문서입니다. 리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. SCP는 AWS Organizations 조직 단위(OU)에 연결하는 제한된 구문이 있는 JSON 정책 문서입니다. ACL은 리소스에도 연결되지만 다른 구문을 사용해야 합니다. 세션 정책은 역할 세션을 수임할 때 제공하는 JSON 정책입니다.

JSON 구문을 이해할 필요가 없습니다. AWS Management 콘솔의 시각적 편집기를 사용하면 JSON을 사용하지 않고 고객 관리형 정책을 생성하고 편집할 수 있습니다. 그러나 그룹 또는 복잡한 정책에 대해 인라인 정책을 사용하도록 선택한 경우 콘솔을 사용하여 JSON 편집기에서 해당 정책을 생성하고 편집해야 합니다. 시각적 편집기 사용에 대한 자세한 정보는 [IAM 정책 만들기 \(p. 377\)](#) 및 [IAM 정책 편집 \(p. 402\)](#) 단원을 참조하십시오.

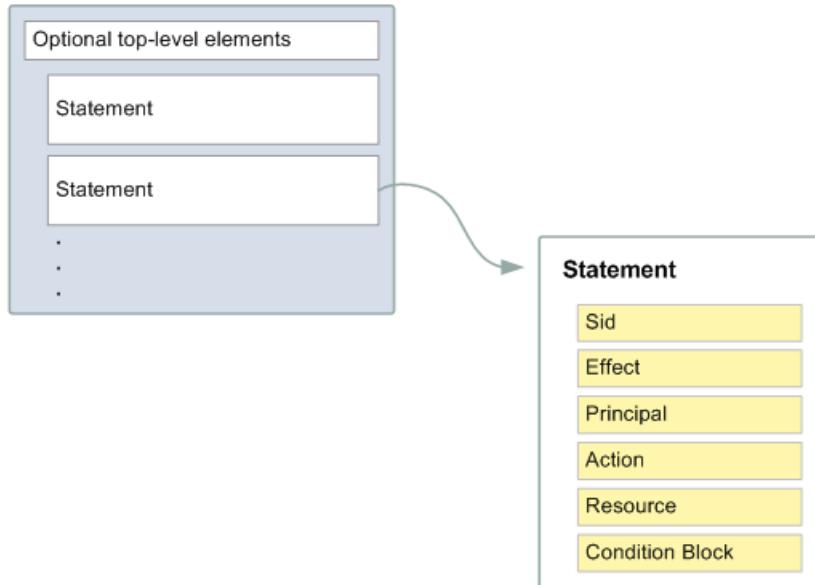
JSON 정책 문서 구조

다음 그림처럼 JSON 정책 문서는 이러한 요소를 포함합니다.

- 문서 상단의 정책 전반의 선택적 정보

- 하나 이상의 개별 문

각 설명문에는 단일 권한에 대한 정보가 포함되어 있습니다. 정책에 설명문이 여러 개 포함되어 있는 경우, AWS는 설명문을 평가하는 동안 전체에 대해 논리 OR을 적용합니다. 요청 하나에 적용되는 정책이 여럿인 경우, AWS는 정책을 평가하는 동안 전체에 걸쳐 논리 OR을 적용합니다.



문의 정보는 일련의 요소 안에 포함되어 있습니다.

- 버전 – 사용하고자 하는 정책 언어의 버전을 지정합니다. 가장 좋은 방법은 최신 2012-10-17 버전을 사용하는 것입니다.
- 설명문 – 이 주요 정책 요소를 다음 요소의 컨테이너로 사용합니다. 정책에 설명문 하나 이상을 포함할 수 있습니다.
- Sid – 선택 설명문 ID를 포함하여 설명문들을 구분합니다.
- 효과 – Allow 또는 Deny를 사용하여 정책에서 액세스를 허용하는지 또는 거부하는지 여부를 설명합니다.
- 보안 주체 – 계정 사용자, 역할, 또는 연합된 사용자로 액세스를 허용할지 거부할지 나타냅니다. 사용자 또는 역할에 연결할 정책을 생성하면 이 요소를 포함할 수 없습니다. 보안 주체는 사용자 또는 역할을 의미합니다.
- 작업 – 정책이 허용하거나 거부하는 작업 목록을 포함합니다.
- 리소스 – 작업이 적용되는 리소스 목록을 지정합니다.
- 조건(선택 사항) – 정책에서 권한을 부여하는 상황을 지정합니다.

이러한 요소와 기타 더 고급 정책 요소에 대한 자세한 정보는 [IAM JSON 정책 요소 참조 \(p. 498\)](#) 단원을 참조하십시오.

복수의 문 및 복수의 정책

개체(사용자, 그룹 또는 역할)에 부여할 권한을 하나 이상 정의하고자 할 경우, 단일 정책에 여러 설명문을 사용하거나 여러 정책을 연결할 수 있습니다. 단일한 설명문에 여러 권한을 정의하고자 할 경우, 정책이 기대하는 액세스를 보장하지 않을 수 있습니다. 가장 좋은 방법은 리소스 유형에 따라 정책을 나누는 것입니다.

[정책의 제한된 크기 \(p. 485\)](#)로 인해 더 복잡한 권한에 대해서는 여러 정책을 사용해야 할 수도 있습니다. 개별 사용자 관리형 정책에 권한의 기능적 그룹화를 만드는 방법이 좋습니다. 예를 들어, IAM 사용자 관리용

정책 하나, 자기 관리용 하나 및 S3 버킷 관리용 기타 정책 하나를 생성합니다. 여러 설명문과 여러 정책의 조합과 상관없이 AWS는 동일한 방식으로 정책을 평가 ([p. 531](#))합니다.

예를 들어, 다음 정책에는 설명문이 세 개 있으며 각 설명문은 단일 계정에 별도의 권한 세트를 부여합니다. 설명문은 다음을 정의합니다.

- Sid(설명문 ID)의 FirstStatement 첫 번째 설명문은 연결된 정책으로 사용자가 자체 암호를 변경하도록 허용합니다. 이 설명문에서 Resource 요소는 "*"("모든 리소스"를 의미)이지만, 실제로 ChangePassword API 작업(또는 그에 상응하는 change-password CLI 명령)은 요청을 하는 사용자의 암호에만 영향을 미칩니다.
- 두 번째 문은 사용자가 자신의 AWS 계정에 있는 모든 Amazon S3 버킷을 나열할 수 있도록 합니다. 이 문서에서 Resource 요소는 "*"("모든 리소스를 의미)이지만 정책에서 다른 계정의 리소스에 대한 액세스 권한을 부여하지 않으므로 사용자는 자신의 AWS 계정에 있는 버킷만 나열할 수 있습니다.
- 세 번째 설명문은 사용자가 confidential-data라는 버킷에 있는 객체를 나열 및 검색할 수 있도록 하지만, 이는 사용자가 멀티 팩터 인증(MFA)에서 인증한 경우에 한합니다. 정책의 Condition 요소는 MFA 인증을 수행합니다.

정책 문에 Condition 요소가 포함된 경우, Condition 요소가 true로 평가된 경우에만 해당 문이 유효합니다. 이때 Condition은 사용자가 MFA 인증된 경우 true로 평가됩니다. 사용자가 MFA 인증되지 않은 경우, 이 Condition은 false로 평가됩니다. 이 경우 이 정책의 세 번째 설명문과 사용자는 confidential-data 버킷을 적용하지 않고 이에 액세스할 수 없습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "FirstStatement",  
            "Effect": "Allow",  
            "Action": ["iam:ChangePassword"],  
            "Resource": "*"  
        },  
        {  
            "Sid": "SecondStatement",  
            "Effect": "Allow",  
            "Action": "s3>ListAllMyBuckets",  
            "Resource": "*"  
        },  
        {  
            "Sid": "ThirdStatement",  
            "Effect": "Allow",  
            "Action": [  
                "s3>List*",  
                "s3:Get*"  
            ],  
            "Resource": [  
                "arn:aws:s3:::confidential-data",  
                "arn:aws:s3:::confidential-data/*"  
            ],  
            "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}  
        }  
    ]  
}
```

JSON 정책 구문 예제

다음 자격 증명 기반 정책은 example_bucket이라는 하나의 Amazon S3 버킷 목록에 암시된 보안 주체를 허용합니다.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    {"Effect": "Allow",
     "Action": "s3>ListBucket",
     "Resource": "arn:aws:s3:::example_bucket"
    }
]
```

다음 리소스 기반 정책은 Amazon S3 버킷에 연결될 수 있습니다. 이 정책에서는 특정 AWS 계정 구성원이 mybucket라는 버킷의 모든 Amazon S3 작업을 수행할 수 있도록 합니다. 작업 내 버킷 또는 객체에 수행될 수 있는 모든 작업을 허용합니다.(이 정책은 계정에만 신뢰를 부여하므로, 해당 계정의 개별 사용자는 지정된 Amazon S3 작업에 대한 권한을 다시 부여받아야 합니다.)

```
{
    "Version": "2012-10-17",
    "Id": "S3-Account-Permissions",
    "Statement": [
        {
            "Sid": "1",
            "Effect": "Allow",
            "Principal": {"AWS": ["arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:root"]},
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::mybucket",
                "arn:aws:s3:::mybucket/*"
            ]
        }
    ]
}
```

공통 시나리오가 포함되는 예제 정책은 [IAM 자격 증명 기반 정책 예제 \(p. 341\)](#) 단원을 참조하십시오.

관리형 정책과 인라인 정책

IAM에서 자격 증명에 대한 권한을 설정해야 할 경우 AWS 관리형 정책, 고객 관리형 정책 또는 인라인 정책 중 어느 것을 사용할지를 결정해야 합니다. 다음 단원에서는 각 자격 증명 기반 정책 유형과 사용 시기에 대해 자세히 살펴보겠습니다.

주제

- [AWS 관리형 정책 \(p. 312\)](#)
- [고객 관리형 정책 \(p. 313\)](#)
- [인라인 정책 \(p. 315\)](#)
- [관리형 정책과 인라인 정책의 선택 \(p. 315\)](#)
- [사용되지 않는 AWS 관리형 정책 \(p. 316\)](#)

AWS 관리형 정책

AWS 관리형 정책은 AWS에서 생성 및 관리하는 독립적인 정책입니다. 여기에서 독립적인 정책이란 정책 스스로 정책 이름이 포함된 Amazon 리소스 이름(ARN)을 갖고 있다는 것을 의미합니다. 예를 들어 `arn:aws:iam::aws:policy/IAMReadOnlyAccess`는 AWS 관리형 정책입니다. ARN에 대한 자세한 내용은 [Amazon 리소스 이름\(ARN\) 및 AWS 서비스 네임스페이스](#) 단원을 참조하십시오.

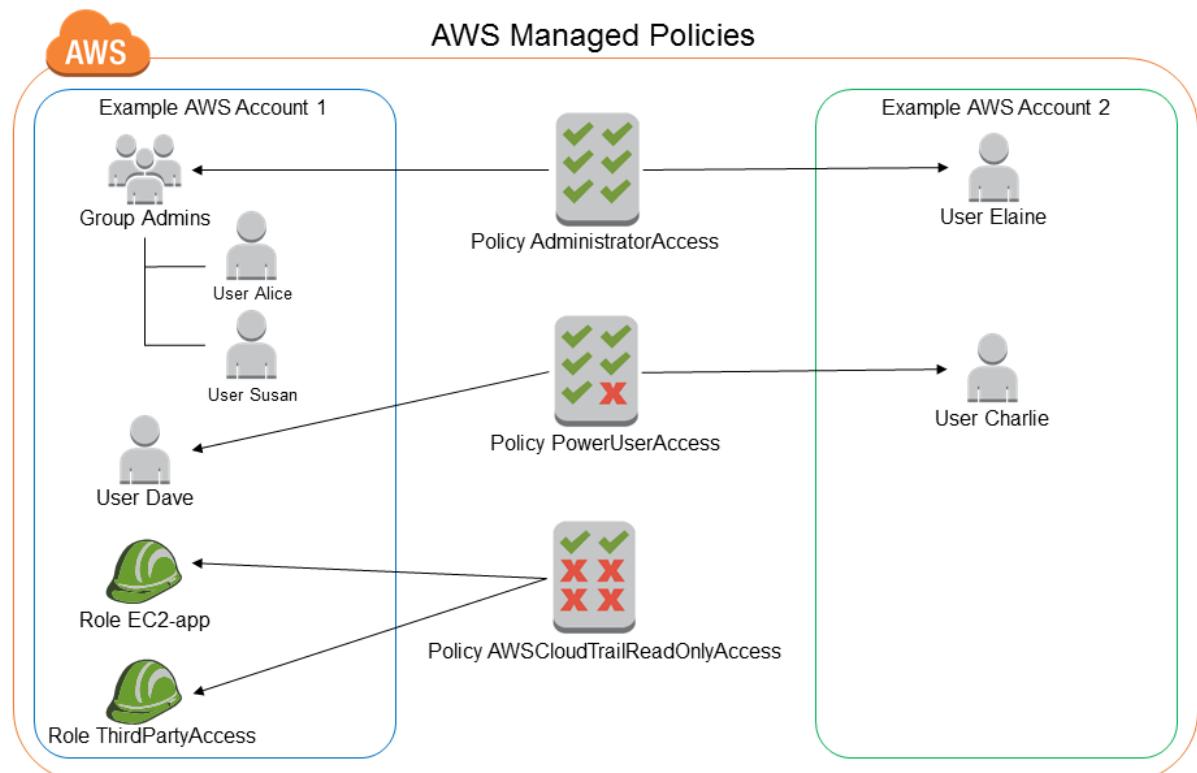
AWS 관리형 정책은 여러 가지 일반 사용 사례에서 권한을 제공할 목적으로 설계되었습니다. `AmazonDynamoDBFullAccess` 및 `IAMFullAccess`와 같은 전체 액세스 AWS 관리형 정책은 서비스에 대한 전체 액세스 권한을 부여하여 서비스 관리자에 대한 권한을 정의합니다. `AWSCodeCommitPowerUser` 및 `AWSKeyManagementServicePowerUser`와 같은 파워 사용자 AWS 관리형 정책은 파워 사용자용으로 설계되었습니다. `AmazonMobileAnalyticsWriteOnlyAccess` 및 `AmazonEC2ReadOnlyAccess`와 같은 부분 액세스 AWS 관리형 정책은 권한 관리 권한을 허용하지 않고 AWS 서비스에 대한 특정 액세스 수준을 제공합니다.

AWS 관리형 정책을 사용하면 정책을 직접 작성하는 것보다 쉽게 사용자, 그룹 및 역할에 적절한 권한을 할당할 수 있습니다.

AWS 관리형 정책에서 특히 유용한 범주 중 하나로, 직무 기능에 대한 범주를 들 수 있습니다. 이러한 정책은 IT 업계에서 일반적으로 사용되는 직무 기능과 긴밀하게 연결됩니다. 이러한 일반적인 직무 기능에 대한 권한 부여를 쉽게 만들기 위해서입니다. 직무 정책을 사용하는 큰 장점 중 하나는 새로운 서비스와 API 작업이 도입될 때마다 AWS가 이를 유지하고 업데이트할 수 있다는 점입니다. 예를 들어 [AdministratorAccess](#) 직무는 AWS의 모든 서비스 및 리소스에 대한 모든 액세스 권한 및 작업 권한을 위임합니다. 이 정책은 계정 관리자에게만 사용하는 것이 좋습니다. IAM 및 조직에 대해서는 제한적인 액세스 권한만 있으면 되지만 그 밖의 모든 서비스에 대해 모든 액세스 권한이 필요한 고급 사용자의 경우, [PowerUserAccess](#) 직무를 사용하십시오. 직무 정책의 목록과 설명은 [직무 기능에 대한 AWS 관리형 정책 \(p. 543\)](#) 단원을 참조하십시오.

AWS 관리형 정책에 정의되어 있는 권한은 변경할 수 없습니다. AWS가 AWS 관리형 정책에서 정의한 권한을 간혹 업데이트합니다. AWS에서 업데이트할 경우 정책이 추가되어 있는 모든 보안 주체 엔터티(사용자, 그룹 및 역할)에게도 업데이트가 적용됩니다. 새로운 AWS 제품을 실행하거나 새로운 API 호출을 기존 서비스에 이용하는 경우 AWS가 AWS 관리형 정책을 업데이트할 가능성이 높습니다. 예를 들어 [ReadOnlyAccess](#)라는 이름의 AWS 관리형 정책은 모든 AWS 서비스 및 리소스에 대한 읽기 전용 액세스 권한을 제공합니다. AWS에서 새로운 서비스가 실행될 때는 AWS가 [ReadOnlyAccess](#) 정책을 업데이트하여 새로운 서비스에 대한 읽기 전용 권한을 추가합니다. 이렇게 업데이트된 권한은 정책이 추가되는 모든 보안 주체 엔터티에게 적용됩니다.

다음은 AWS 관리형 정책을 나타낸 다이어그램입니다. 다이어그램을 보면 [AdministratorAccess](#), [PowerUserAccess](#), 그리고 [AWSCloudTrailReadOnlyAccess](#) 등 3개의 AWS 관리형 정책이 있습니다. 다이어그램에도 나와 있지만 단일 AWS 관리형 정책을 다른 AWS 계정의 보안 주체 엔터티에 추가할 수도 있고, 단일 AWS 계정의 다른 보안 주체 엔터티에 추가할 수도 있습니다.

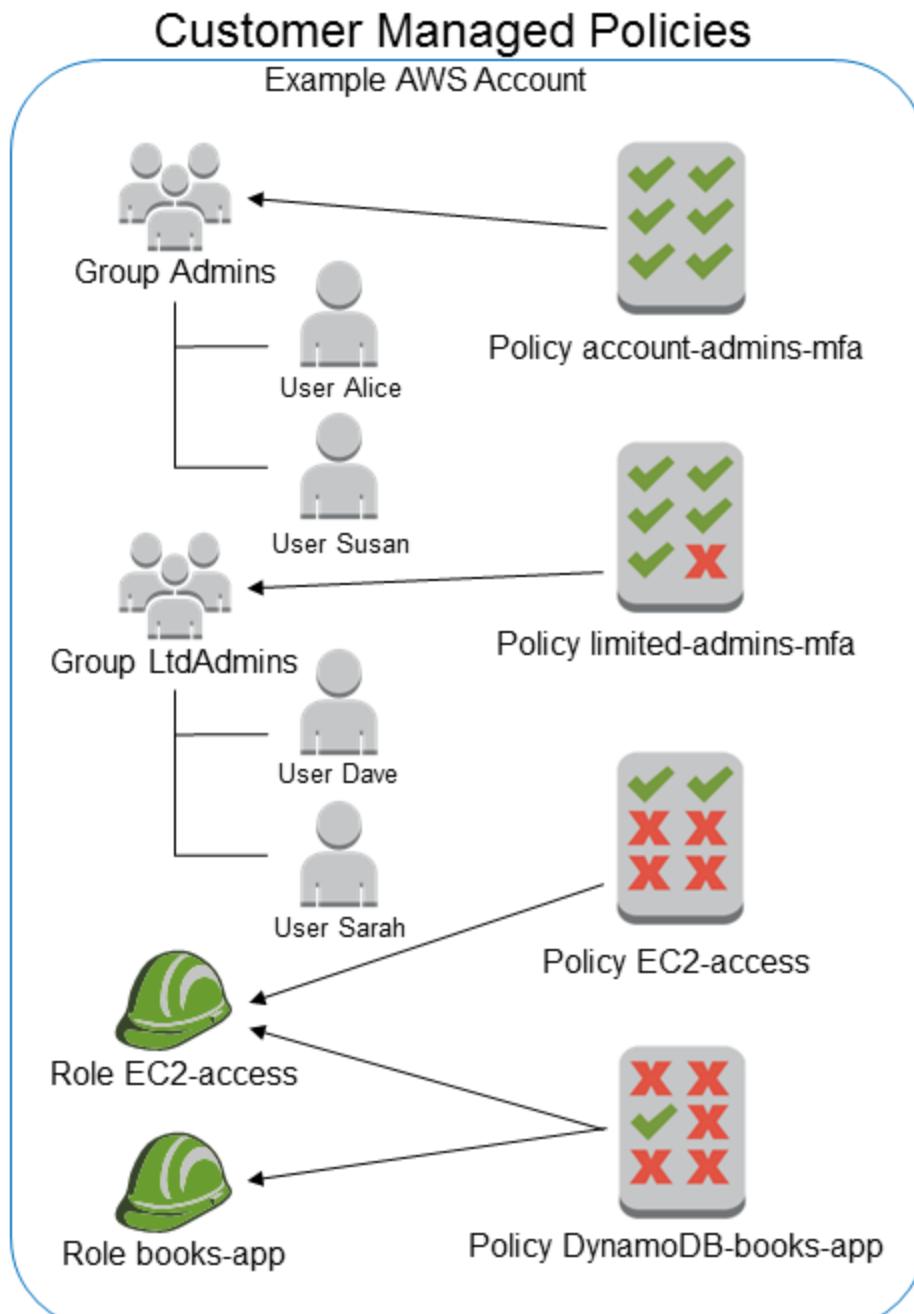


고객 관리형 정책

독립적인 정책은 사용자 자신의 AWS 계정에서 관리하도록 생성할 수도 있습니다. 이러한 정책을 고객 관리형 정책이라고 합니다. 이렇게 생성된 정책은 AWS 계정에 속한 다수의 보안 주체 엔터티에 추가할 수 있습니다. 정책을 보안 주체 엔터티에 추가할 경우 정책에서 정의한 권한까지 엔터티에게 부여하게 됩니다.

고객이 관리하는 정책을 생성하는 좋은 방법은 AWS에서 관리하는 기존의 정책을 복사하여 시작하는 것입니다. 이렇게 하면 시작 시 올바른 정책으로 시작하므로 해당 환경에 맞게 사용자 지정만 하면 됩니다.

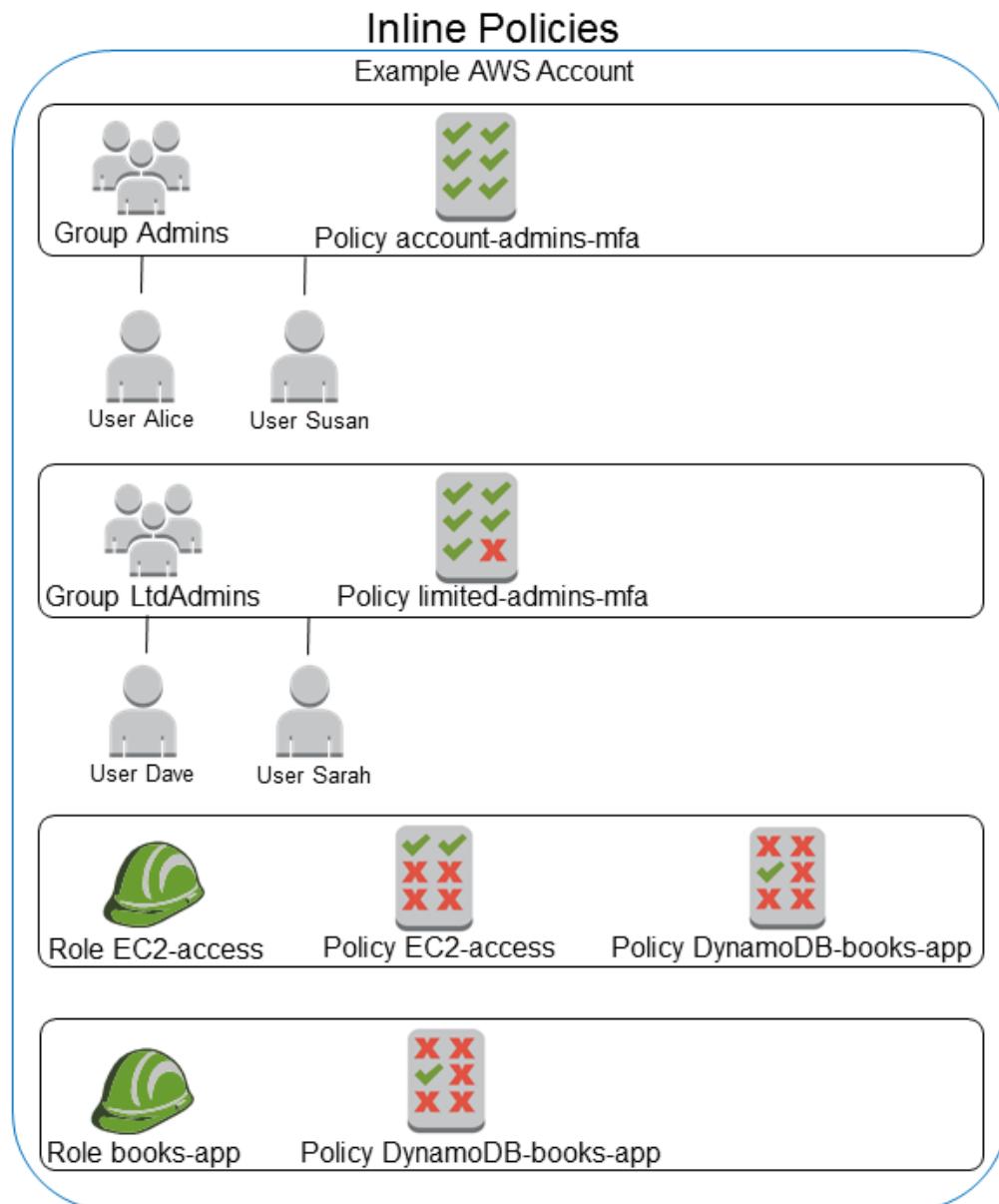
다음은 고객 관리형 정책을 나타낸 다이어그램입니다. 각 정책은 자체적으로 정책 이름이 포함된 Amazon 리소스 이름(ARN)을 갖고 있는 IAM 엔터티입니다. 다이어그램을 보면 동일한 정책을 여러 보안 주체 엔터티에 추가할 수 있습니다.—예를 들어 동일한 DynamoDB-books-app 정책이 2개의 다른 IAM 역할에 추가됩니다.



인라인 정책

인라인 정책이란 보안 주체 개체(사용자, 그룹 또는 역할)에 삽입되는 정책을 가리킵니다. 즉, 이 정책은 보안 주체 개체에서 내재된 부분이라고 할 수 있습니다. 이 정책은 보안 주체 개체 생성 시, 혹은 나중에라도 생성 하여 보안 주체 개체에 삽입할 수 있습니다.

다음은 인라인 정책을 나타낸 다이어그램입니다. 각 정책은 사용자, 그룹 또는 역할에서 내재된 부분입니다. 다이어그램을 보면 2개의 역할에 동일한 정책(the DynamoDB-books-app 정책)이 추가되어 있지만, 단 하나의 정책도 공유하지 않고 역할마다 자체적으로 정책 사본을 갖고 있습니다.



관리형 정책과 인라인 정책의 선택

정책 유형이 다르면 사용 사례도 다릅니다. 대부분 경우 인라인 정책보다는 관리형 정책의 사용을 권장합니다.

관리형 정책은 다음과 같은 기능을 제공합니다.

재사용성

단일 관리형 정책은 다수의 보안 주체 개체(사용자, 그룹 및 역할)에 추가할 수 있습니다. 실제로 정책 라이브러리를 생성하여 AWS 계정에 유용한 권한을 정의한 다음 필요에 따라 생성한 정책을 보안 주체 엔터티에 추가하는 것이 가능합니다.

중앙 변경 관리

관리형 정책 변경 시 정책이 추가되어 있는 모든 보안 주체 엔터티에 변경 사항이 적용됩니다. 예를 들어 AWS API 권한을 추가할 경우 관리형 정책을 업데이트하여 권한을 추가할 수 있습니다. (AWS 관리형 정책을 사용할 때는 AWS가 정책을 업데이트합니다) 정책이 업데이트되면 정책이 추가되어 있는 모든 보안 주체 엔터티에 변경 사항이 적용됩니다. 이와는 대조적으로 인라인 정책을 변경하려면 정책이 추가되어 있는 보안 주체 개체를 일일이 편집해야 합니다. 예를 들어 그룹과 역할에 모두 동일한 인라인 정책이 추가되어 있더라도 정책을 변경하기 위해서는 두 보안 주체 개체를 개별적으로 편집해야만 합니다.

버전 관리 및 룰백

고객 관리 정책을 변경할 경우 변경된 정책은 기존 정책을 덮어쓰지 않습니다. 대신 IAM에서 관리형 정책의 새 버전을 만듭니다. IAM은 고객 관리 정책을 최대 5개 버전까지 저장합니다. 정책 버전은 필요에 따라 정책을 이전 버전으로 되돌리는 데도 사용됩니다.

정책 버전은 **Version** 정책 요소와 다릅니다. **Version** 정책 요소는 정책 내에서 사용되며 정책 언어의 버전을 정의합니다. 정책 버전에 대한 자세한 내용은 [the section called “IAM 정책 버전 관리” \(p. 399\)](#) 단원을 참조하십시오. **Version** 정책 요소에 대한 자세한 내용은 [IAM JSON 정책 요소: Version \(p. 499\)](#) 단원을 참조하십시오.

권한 위임 관리

정책으로 정의한 권한을 지속적으로 제어하면서 AWS 계정에 속한 사용자가 정책을 추가 및 분리하도록 허용할 수 있습니다. 실제로 일부 사용자에게는 전체 관리자 권한을 위임할 수 있습니다. 다시 말해, 전체 관리자란 정책을 생성, 업데이트 및 삭제할 수 있는 것을 말합니다. 제한된 관리자로서 다른 사용자를 지정할 수 있습니다. 다시 말해, 관리자는 다른 보안 주체 개체에게 정책을 추가할 수 있지만 이때 정책은 추가가 허용된 정책으로 제한됩니다.

권한 위임 관리에 대한 자세한 내용은 [정책에 대한 액세스 제어 \(p. 332\)](#) 단원을 참조하십시오.

AWS 관리형 정책의 자동 업데이트

AWS는 AWS 관리형 정책을 유지하면서 필요에 따라 자동으로 업데이트하기 때문에(예를 들어 새로운 AWS 서비스 권한을 추가하기 위해) 직접 변경할 필요가 없습니다. 업데이트는 AWS 관리형 정책을 추가한 보안 주체 엔터티에게 자동으로 적용됩니다.

인라인 정책 사용

인라인 정책은 정책과 정책이 추가된 보안 주체 엔터티를 정확히 1대 1 관계로 유지할 때 유용합니다. 예를 들어 정책 권한을 의도하지 않은 보안 주체 개체에게 실수로 할당하는 일을 배제하려고 합니다. 이때 인라인 정책을 사용하면 정책 권한이 잘못된 보안 주체 개체에게 실수로 추가되는 일이 사라집니다. 그 밖에도 AWS Management 콘솔을 사용하여 보안 주체 개체를 삭제할 경우 보안 주체 개체에 삽입된 정책 역시 삭제됩니다. 정책도 보안 주체 개체의 일부이기 때문입니다.

사용되지 않는 AWS 관리형 정책

권한 할당을 간편하게 하기 위해 AWS는 IAM 사용자, 그룹 및 역할에 연결할 수 있도록 사전에 정의된 [관리형 정책 \(p. 312\)](#)을 제공합니다.

새로운 서비스가 나왔을 때와 같이 AWS는 때때로 기존 정책에 새 권한을 추가해야 합니다. 기존 정책에 새 권한을 추가해도 특성이나 권한이 제거되거나 방해를 받지는 않습니다.

하지만 AWS는 필요한 변경이 기존 정책에 적용될 경우 고객에게 영향을 줄 수 있기 때문에 새로 정책을 만듭니다. 예를 들어 기존 정책에서 권한을 제거하면 이 정책을 사용하는 IAM 주체나 애플리케이션의 권한이 손상되어 중요한 작업에 방해가 될 수 있습니다.

따라서 이러한 변경이 필요할 경우 AWS는 해당 사항을 변경한 정책을 새로 만들어서 고객에게 제공합니다. 기존 정책은 사용되지 않음으로 표시됩니다. 사용되지 않는 관리형 정책은 IAM 콘솔의 정책 목록에서 옆에 경고 아이콘이 표시됩니다.

사용되지 않는 정책은 다음과 같은 특성을 갖습니다.

- 현재 연결된 모든 사용자, 그룹 및 역할에 계속 적용됩니다. 연결이 해제되지 않습니다.
- 새로운 사용자, 그룹 또는 역할에 연결할 수 없습니다. 현재 주체에서 연결을 해제할 경우 다시 연결할 수 없습니다.
- 현재의 모든 주체로부터 연결을 해제하면 더 이상 표시되지 않으며 어떤 경우에도 다시 사용할 수 없습니다.

사용자, 그룹 또는 역할에 정책이 필요할 경우 새로운 정책을 연결해야 합니다. 정책이 사용되지 않음으로 설정되었다고 알림을 받으면 모든 사용자, 그룹 및 역할을 대체 정책에 연결하고 사용되지 않는 정책으로부터 연결을 해제하는 것이 좋습니다. 사용되지 않는 정책을 계속 사용하면 위험이 수반될 수 있으므로 대체 정책으로 전환하는 것이 좋습니다.

IAM 엔터티에 대한 권한 경계

AWS에서는 IAM 엔터티(사용자 또는 역할)에 대한 권한 경계를 지원합니다. 권한 경계는 관리형 정책을 사용하여 자격 증명 기반 정책을 통해 IAM 엔터티에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 엔터티에 대한 권한 경계를 설정할 경우 해당 엔터티는 자격 증명 기반 정책 및 관련 권한 경계 모두에서 허용되는 작업만 수행할 수 있습니다.

정책 유형에 대한 자세한 정보는 [정책 유형 \(p. 305\)](#) 단원을 참조하십시오.

AWS 관리형 정책 또는 고객 관리형 정책을 사용하여 IAM 엔터티(사용자 또는 역할) 경계를 설정할 수 있습니다. 이 정책은 사용자 또는 역할에 대해 최대 권한을 제한합니다.

예를 들어, ShirleyRodriguez라는 IAM 사용자에 대해 Amazon S3, Amazon CloudWatch 및 Amazon EC2만 관리하도록 허용되어야 한다고 가정해 보겠습니다. 이 규칙을 시행하려면 다른 정책을 사용하여 ShirleyRodriguez 사용자의 권한 경계를 설정합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:*",  
                "cloudwatch:*",  
                "ec2:/*"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

정책을 사용하여 사용자 권한 경계를 설정하면 이 정책은 사용자 권한을 제한하지만 자체적으로 권한을 제공하지 않습니다. 이 예제에서 정책은 ShirleyRodriguez의 최대 권한을 Amazon S3, CloudWatch 및 Amazon EC2의 모든 작업으로 설정합니다. Shirley가 작업을 허용하는 권한 정책이 있다고 해도 IAM을 포함한 다른 서비스에서는 이 작업을 절대 수행할 수 없습니다. 예를 들어 다음 정책을 ShirleyRodriguez 사용자에게 추가할 수 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": "iam:CreateUser",  
         "Resource": "*"}  
    ]  
}
```

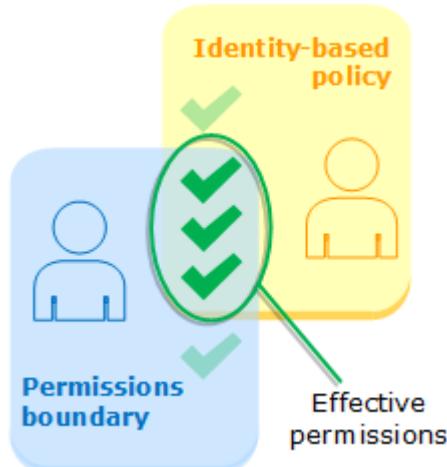
이 정책은 IAM에서 사용자 생성을 허용합니다. 이 정책을 ShirleyRodriguez 사용자에 연결하고 Shirley가 사용자를 생성하고자 할 경우 작업은 실패합니다. 정책 평가 로직은 iam:CreateUser 작업을 허용하지 않는 권한 경계로서 사용된 정책을 확인하기 때문에 이 작업은 실패합니다. Shirley가 AWS에서 작업을 수행하도록 허용하기 위해서는 Amazon S3, Amazon CloudWatch, 또는 Amazon EC2의 작업을 통해 권한 정책을 추가해야 합니다. 또는 권한 경계를 업데이트하여 그녀에게 IAM에서 사용자를 생성하도록 허용할 수도 있습니다.

경계가 있는 효과적인 권한 평가

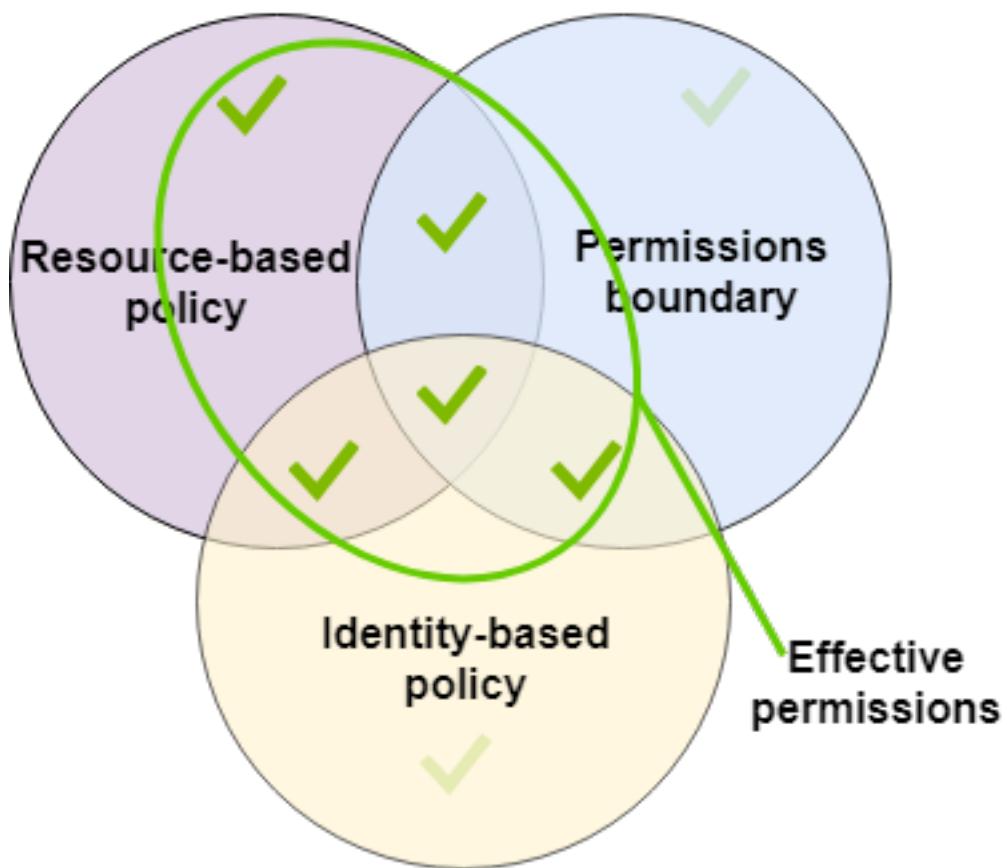
IAM 엔터티(사용자 또는 역할)에 대한 권한 경계는 엔터티에 부여할 수 있는 최대 권한을 설정합니다. 사용자 또는 역할에 대한 효과적 권한을 변경할 수 있습니다. 사용자 또는 역할에 영향을 주는 모든 정책을 통해 부여되는 권한이 개체에 대한 유효 권한입니다. 계정 내에서 엔터티에 대한 권한은 자격 증명 기반 정책, 리소스 기반 정책, 권한 경계, 조직 SCP 또는 세션 정책에 영향을 받을 수 있습니다. 다양한 유형의 정책에 대한 자세한 정보는 [정책 및 권한 \(p. 305\)](#) 단원을 참조하십시오.

이러한 정책 유형 중 하나에서 작업에 대한 액세스가 명시적으로 거부된 경우 해당 요청이 거부됩니다. 여러 권한 유형에 의해 엔터티에 부여된 권한은 훨씬 더 복잡합니다. AWS의 정책 평가에 대한 자세한 정보는 [정책 평가 로직 \(p. 531\)](#) 단원을 참조하십시오.

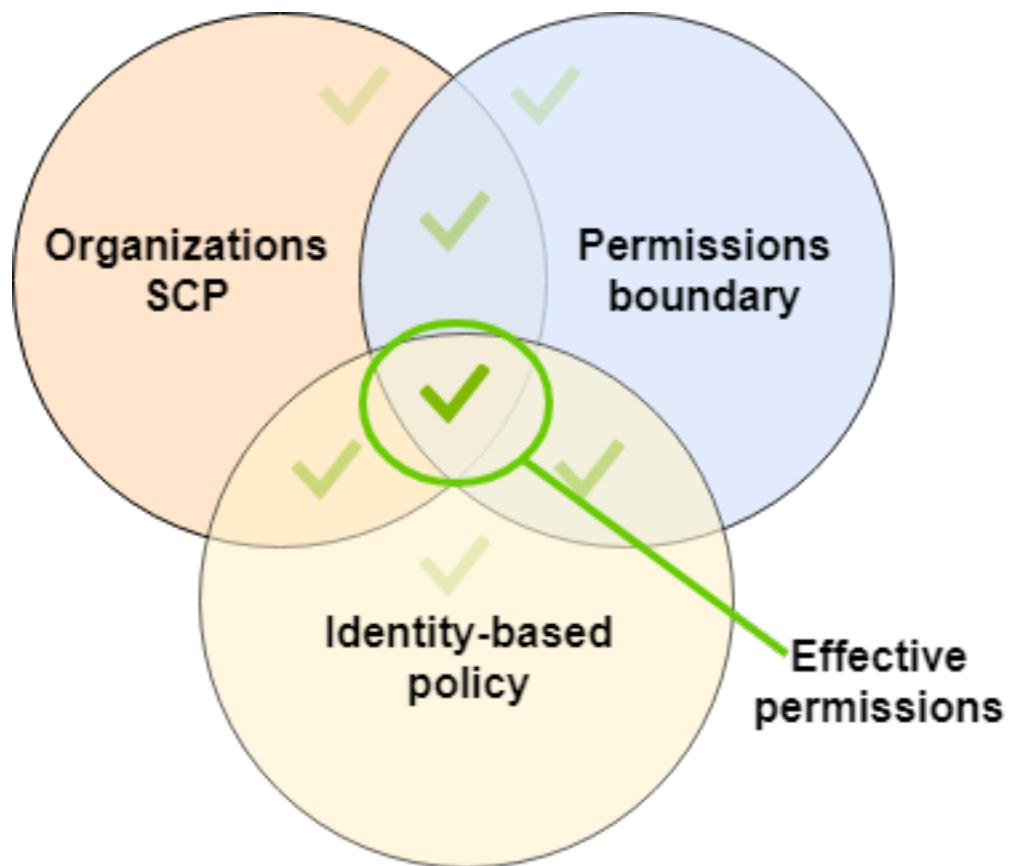
자격 증명 기반 정책과 경계 – 자격 증명 기반 정책은 사용자, 사용자 그룹 또는 역할에 연결된 인라인 또는 관리형 정책입니다. 자격 증명 기반 정책은 엔터티에 권한을 부여하며, 권한 경계는 이러한 권한을 제한합니다. 두 정책 유형 모두에서 허용되는 권한이 유효 권한입니다. 이들 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다.



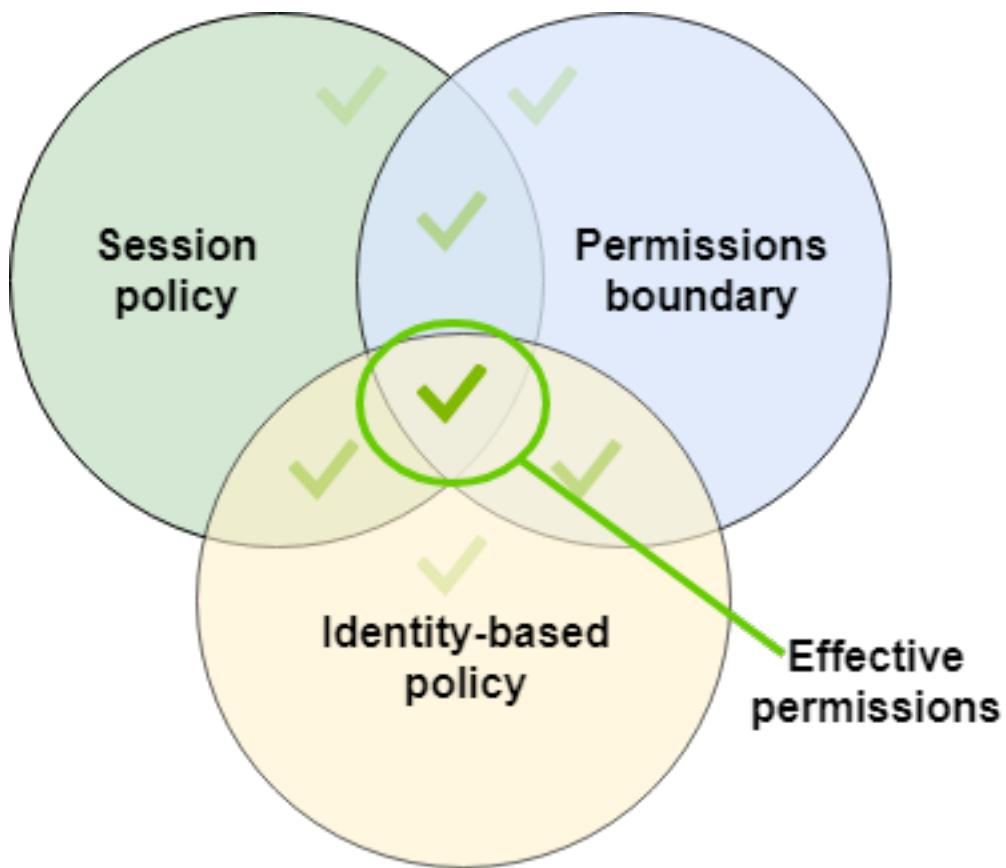
리소스 기반 정책 – 리소스 기반 정책은 지정된 보안 주체가 정책이 연결된 리소스에 액세스하는 방식을 제어합니다. 계정 내에서 권한 경계는 리소스 기반 정책을 통해 부여된 권한을 축소하지 않습니다. 권한 경계가 자격 증명 기반 정책을 통해 엔터티에 부여된 권한을 축소하고 나서, 리소스 기반 정책이 엔터티에 추가 권한을 제공합니다. 리소스 기반 정책을 통해 허용되는 권한과 권한 경계 및 자격 증명 기반 정책 모두에서 허용되는 권한이 유효 권한입니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다.



조직 SCP – SCP는 전체 AWS 계정에 적용됩니다. SCP는 해당 계정 내 보안 주체가 보낸 모든 요청에 대한 권한을 제한합니다. IAM 엔터티(사용자 또는 역할)가 SCP, 권한 경계 및 자격 증명 기반 정책에 영향을 받는 요청을 보내는 경우 해당 요청은 이러한 세 가지 정책 유형 모두에서 허용되는 경우에만 허용됩니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다.



세션 정책 – 세션 정책은 역할 또는 연합된 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 세션에 대한 권한은 세션을 생성하는 데 사용되는 IAM 엔터티(사용자 또는 역할)와 세션 정책에서 가져옵니다. 엔터티의 자격 증명 기반 정책 권한은 세션 정책과 권한 경계에 제한을 받습니다. 세 정책 유형 모두에서 허용되는 권한이 이 정책 유형 세트에 대한 유효 권한입니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 세션 정책에 대한 자세한 정보는 [세션 정책](#)을 참조하십시오.



권한 경계를 사용하여 다른 것에 책임 위임

권한 경계를 사용하여 사용자 생성과 같은 권한 관리 작업을 계정의 IAM 사용자에 위임할 수 있습니다. 이로써 권한의 특정 경계 내에서 다른 사용자가 작업을 대신 수행할 수 있는 권한이 부여됩니다.

예를 들어, María가 X-Company AWS 계정 관리자라고 가정하십시오. María가 Zhang에게 사용자 생성 업무를 위임하고자 합니다. 하지만 Zhang이 다음 회사 규칙에 따라 사용자를 생성하는지 확신해야 합니다.

- 사용자는 IAM을 사용하여 사용자, 그룹, 역할 또는 정책을 생성하고 관리할 수 없습니다.
- 사용자의 Amazon S3 logs 버킷 액세스가 거부되고 사용자가 i-1234567890abcdef0Amazon EC2 인스턴스로 액세스할 수 없습니다.
- 사용자는 사용자 자체 경계 정책을 제거할 수 없습니다.

이런 규칙을 시행하기 위해서는 María는 아래와 같은 세부 정보가 포함된 작업을 완료합니다.

1. María는 `xCompanyBoundaries` 관리형 정책을 생성하여 계정의 모든 새로운 사용자에 대한 권한 경계로서 사용할 수 있습니다.
2. María는 `DelegatedUserBoundary` 관리형 정책을 생성하여 Zhang에 대한 권한 경계로서 할당합니다.
3. María는 `DelegatedUserPermissions` 관리형 정책을 생성하여 Zhang에 대한 권한 정책으로서 연결합니다.
4. María가 Zhang에서 그의 새로운 책임과 제한을 알려줍니다.

작업 1: María는 먼저 관리형 정책을 생성하여 새로운 사용자에 대한 경계를 정의해야 합니다. María는 Zhang이 사용자에게 필요한 권한 정책을 사용자에게 부여할 수 있도록 허용하지만 사용자를 제한하고자 합

니다. 이렇게 하기 위해서는 마리아는 xCompanyBoundaries라는 다음 고객 관리형 정책을 생성합니다. 이 정책을 통해 사용자에게 IAM의 제한된 자체 관리 액세스인 몇 가지 서비스에 대한 완전한 액세스를 허용하고 Amazon S3 로그 버킷 또는 i-1234567890abcdef0 Amazon EC2 인스턴스에 대한 액세스는 거부하게 됩니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "ServiceBoundaries",  
            "Effect": "Allow",  
            "Action": [  
                "s3:*",  
                "cloudwatch:*",  
                "ec2:*",  
                "dynamodb:*"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Sid": "AllowIAMConsoleForCredentials",  
            "Effect": "Allow",  
            "Action": [  
                "iam>ListUsers",  
                "iam:GetAccountPasswordPolicy"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Sid": "AllowManageOwnPasswordAndAccessKeys",  
            "Effect": "Allow",  
            "Action": [  
                "iam:*AccessKey*",  
                "iam:ChangePassword",  
                "iam:GetUser",  
                "iam:*ServiceSpecificCredential*",  
                "iam:*SigningCertificate*"  
            ],  
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]  
        },  
        {  
            "Sid": "DenyS3Logs",  
            "Effect": "Deny",  
            "Action": "s3:",  
            "Resource": [  
                "arn:aws:s3::::logs",  
                "arn:aws:s3::::logs/*"  
            ]  
        },  
        {  
            "Sid": "DenyEC2Production",  
            "Effect": "Deny",  
            "Action": "ec2:",  
            "Resource": "arn:aws:ec2::*:instance/i-1234567890abcdef0"  
        }  
    ]  
}
```

각 설명문은 다른 목적이 있습니다.

1. 이 정책의 ServiceBoundaries 설명문은 지정된 AWS 서비스에 대한 완전한 액세스를 허용합니다. 이런 서비스의 새로운 사용자 작업이 사용자에 연결된 권한 정책에 따라서만 제한된다는 의미입니다.

2. AllowIAMConsoleForCredentials 문에서 모든 IAM 사용자를 나열할 수 있는 액세스를 허용합니다. 이 액세스는 AWS Management 콘솔의 사용자 페이지를 탐색하는 데 필요합니다. 또한 계정의 암호 요구 사항을 확인하도록 허용합니다. 이 액세스는 자신의 고유 암호를 변경할 때 필요합니다.
3. AllowManageOwnPasswordAndAccessKeys 문은 사용자가 자신의 고유 콘솔 암호와 프로그래밍 방식의 액세스 키만 관리하도록 허용합니다. 이런 점은 중요합니다. Zhang 또는 다른 관리자가 새로운 사용자에게 IAM으로 완전한 액세스가 되는 권한 정책을 부여한다면 사용자 자신 또는 다른 사용자 권한을 변경 할 수 있기 때문입니다. 이 설명문은 이런 상황을 방지할 수 있습니다.
4. DenyS3Logs 설명문은 logs 버킷 액세스를 명시적으로 거부합니다.
5. DenyEC2Production 설명문은 i-1234567890abcdef0 인스턴스 액세스를 명시적으로 거부합니다.

작업 2: María는 Zhang이 모든 X-Company 사용자를 생성하도록 허용하지만 XCompanyBoundaries 권한 경계를 통해서만 허용하고자 합니다. 마리아는 DelegatedUserBoundary라는 다음 고객 관리형 정책을 생성합니다. 이런 정책은 Zhang이 가질 수 있는 최대 권한을 정의합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "CreateOrChangeOnlyWithBoundary",  
            "Effect": "Allow",  
            "Action": [  
                "iam:CreateUser",  
                "iam:DeleteUserPolicy",  
                "iam:AttachUserPolicy",  
                "iam:DetachUserPolicy",  
                "iam:PutUserPermissionsBoundary"  
            ],  
            "Resource": "*",  
            "Condition": {"StringEquals":  
                {"iam:PermissionsBoundary": "arn:aws:iam::111122223333:policy/  
XCompanyBoundaries"}}        },  
        {  
            "Sid": "CloudWatchAndOtherIAMTasks",  
            "Effect": "Allow",  
            "Action": [  
                "cloudwatch:*",  
                "iam:GetUser",  
                "iam>ListUsers",  
                "iam>DeleteUser",  
                "iam:UpdateUser",  
                "iam>CreateAccessKey",  
                "iam>CreateLoginProfile",  
                "iam:GetAccountPasswordPolicy",  
                "iam:GetLoginProfile",  
                "iam:*Group*",  
                "iam>CreatePolicy",  
                "iam>DeletePolicy",  
                "iam>DeletePolicyVersion",  
                "iam:GetPolicy",  
                "iam:GetPolicyVersion",  
                "iam GetUserPolicy",  
                "iam GetRolePolicy",  
                "iam>ListPolicies",  
                "iam>ListPolicyVersions",  
                "iam>ListEntitiesForPolicy",  
                "iam>ListUserPolicies",  
                "iam>ListAttachedUserPolicies",  
                "iam>ListRolePolicies",  
                "iam>ListAttachedRolePolicies",  
                "iam:PutUserPolicy",  
                "iam:SetDefaultPolicyVersion",  
            ]  
        }  
    ]  
}
```

```

        "iam:SimulatePrincipalPolicy",
        "iam:SimulateCustomPolicy"
    ],
    "Resource": "*"
},
{
    "Sid": "NoBoundaryPolicyEdit",
    "Effect": "Deny",
    "Action": [
        "iam:CreatePolicyVersion",
        "iam:DeletePolicy",
        "iam:DeletePolicyVersion",
        "iam:SetDefaultPolicyVersion"
    ],
    "Resource": [
        "arn:aws:iam::123456789012:policy/XCompanyBoundaries",
        "arn:aws:iam::123456789012:policy/DelegatedUserBoundary"
    ]
},
{
    "Sid": "NoBoundaryUserDelete",
    "Effect": "Deny",
    "Action": "iam:DeleteUserPermissionsBoundary",
    "Resource": "*"
}
]
}

```

각 설명문은 다른 목적이 있습니다.

1. `CreateOrChangeOnlyWithBoundary` 설명문은 Zhang이 IAM 사용자를 생성하도록 허용하지만 장이 권한 경계를 설정할 때 `XCompanyBoundaries` 정책을 사용할 때만 가능합니다. 이 설명문은 또한 장이 기존 사용자에 대한 권한 경계를 설정하도록 허용하지만 장이 동일한 정책을 사용할 때만 가능합니다. 마지막으로, 이 설명문은 Zhang이 이 권한 경계 설정을 통해 사용자에 대한 권한 정책을 관리하도록 허용합니다.
2. `CloudWatchAndOtherIAMTasks` 설명문은 Zhang이 사용자, 그룹 및 정책 관리 작업을 완료하도록 허용합니다. Zhang이 자신 또는 다른 사용자로부터 권한 경계를 삭제할 수 있는 권리가 없다는 점을 유의하십시오.
3. `NoBoundaryPolicyEdit` 설명문은 Zhang이 `XCompanyBoundaries` 정책을 업데이트할 수 있는 액세스를 거부합니다. 장은 자신 또는 다른 사용자에 대한 권한 경계를 설정하는 데 사용되는 어떤 정책도 변경할 수 없습니다.
4. `NoBoundaryUserDelete` 문에서는 Zhang이 자신 또는 다른 사용자에 대해 권한 경계를 삭제하기 위해 액세스할 때 이를 거부합니다.

그런 다음 María는 Zhang 사용자에 대한 [권한 경계로서 \(p. 76\)](#) `DelegatedUserBoundary` 정책을 할당합니다.

작업 3: 권한 경계가 최대 권한을 제한하지만 자체 액세스를 허용하지 않기 때문에 María는 Zhang에 대한 권한 정책을 생성해야 합니다. 마리아는 `DelegatedUserPermissions`라는 다음 정책을 생성합니다. 이 정책은 정의된 경계 내에서 Zhang이 수행할 수 있는 작업을 정의합니다.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "IAM",
            "Effect": "Allow",
            "Action": "iam:*",
            "Resource": "*"
        },
        ...
    ]
}

```

```
{  
    "Sid": "CloudWatchLimited",  
    "Effect": "Allow",  
    "Action": [  
        "cloudwatch:GetDashboard",  
        "cloudwatch:GetMetricData",  
        "cloudwatch>ListDashboards",  
        "cloudwatch:GetMetricStatistics",  
        "cloudwatch>ListMetrics"  
    ],  
    "Resource": "*"  
},  
{  
    "Sid": "S3BucketContents",  
    "Effect": "Allow",  
    "Action": "s3>ListBucket",  
    "Resource": "arn:aws:s3:::ZhangBucket"  
}  
]  
}
```

각 설명문은 다른 목적이 있습니다.

- 정책의 IAM 설명문은 IAM에 대한 Zhang의 완전한 액세스를 허용합니다. 그러나, Zhang 권한 경계가 몇 가지 IAM 작업만 허용하기 때문에 Zhang의 효과적인 IAM 권한은 그의 권한 경계에 의해서만 제한됩니다.
- CloudWatchLimited 설명문은 Zhang이 CloudWatch에서 5가지 작업을 수행할 수 있도록 허용합니다. Zhang 권한 경계는 CloudWatch의 모든 작업을 허용하기 때문에 그의 효과적인 CloudWatch 권한은 그의 권한 정책에 의해서만 제한됩니다.
- S3BucketContents 설명문은 Zhang이 ZhangBucket Amazon S3 버킷을 나열할 수 있도록 허용합니다. 그러나, Zhang 권한 경계는 어떠한 Amazon S3 작업도 허용하지 않기 때문에 Zhang은 그의 권한 정책과 상관없이 어떠한 S3 작업을 수행할 수 없습니다.

그러면 María는 DelegatedUserPermissions 정책을 Zhang 사용자에 대한 권한 정책으로서 연결합니다.

작업 4: María는 새로운 사용자를 생성하도록 Zhang에게 지침을 내립니다. María는 Zhang에게 새로운 사용자가 원하는 모든 권한을 통해 새로운 사용자를 생성할 수 있지만 xCompanyBoundaries 정책을 권한 경계로서 할당해야 한다고 말합니다.

Zhang은 다음 작업을 완료합니다.

- Zhang은 AWS Management 콘솔로 [사용자를 생성 \(p. 66\)](#)합니다. 그는 사용자 이름 Nikhil를 입력하고 사용자에 대한 콘솔 액세스를 가능하게 합니다.
- 권한 설정 페이지에서 Zhang은 Nikhil가 업무를 할 수 있도록 허용하는 IAMFullAccess 및 AmazonS3ReadOnlyAccess 권한 정책을 선택합니다.
- Zhang은 María의 지침을 읽고 Set permissions boundary(권한 경계 설정) 섹션을 넘깁니다.
- Zhang은 사용자 세부 정보를 검토하고 사용자 생성을 선택합니다.

작업은 실패하고 액세스는 거부됩니다. Zhang의 DelegatedUserBoundary 권한 경계는 그가 생성하는 어떠한 사용자도 xCompanyBoundaries 정책을 권한 경계로서 가지고 있어야 합니다.

- Zhang은 이전 페이지로 돌아갑니다. 그는 Set permissions boundary(권한 경계 설정) 페이지에서 xCompanyBoundaries 정책을 선택합니다.
- Zhang은 사용자 세부 정보를 검토하고 사용자 생성을 선택합니다.

사용자가 생성됩니다.

Nikhil가 로그인할 경우, 그는 권한 경계가 거부한 작업 이외의 IAM 및 Amazon S3로 액세스할 수 있습니다. 예를 들어, 그는 IAM에 자신의 암호를 변경할 수 있지만 다른 사용자를 생성하거나 그의 정책을 편집할 수 없습니다. Nikhil의 경우 Amazon S3에서 소유하고 있는 모든 버킷에 대한 읽기 전용 액세스만이 가능합니다. 하지만 누군가 그에게 logs 버킷에 대한 소유권을 부여하는 경우에도 이를 볼 수는 없습니다. 버킷 소유권에 대한 자세한 정보는 Amazon Simple Storage Service 개발자 가이드의 [Amazon S3 리소스에 대한 액세스 권한 관리](#) 단원을 참조하십시오.

누군가가 Nikhil에게 i-1234567890abcdef0 인스턴스 시작/종지를 허용하는 리소스 기반 정책을 해당 인스턴스에 추가하더라도 Nikhil은 여전히 해당 인스턴스를 관리할 수 없습니다. i-1234567890abcdef0 인스턴스에 대한 작업이 권한 경계에 의해 명시적으로 거부되었기 때문입니다. 정책 유형에 포함된 명시적 거부로 인해 요청이 거부됩니다. 하지만 Secrets Manager 암호에 연결된 리소스 기반 정책이 Nikhil가 secretsmanager:GetSecretsValue 작업을 수행하도록 허용하는 경우 Nikhil은 암호를 불러와서 암호화를 해제할 수 있습니다. 그 이유는 Secrets Manager 작업이 Nikhil의 권한 경계에 의해 명시적으로 거부되지 않았으므로 권한 경계에서 리소스 기반 정책을 제한하지 않기 때문입니다.

자격 증명 기반 정책 및 리소스 기반 정책

정책은 자격 증명 또는 리소스에 연결될 때 해당 권한을 정의하는 AWS의 객체입니다. 리소스에 대한 액세스를 제한하는 권한 정책을 생성할 때 자격 증명 기반 정책 또는 리소스 기반 정책을 선택할 수 있습니다.

자격 증명 기반 정책은 IAM 사용자, 그룹 또는 역할에 연결됩니다. 이러한 정책으로 자격 증명이 수행할 수 있는 작업(권한)을 지정할 수 있습니다. 예를 들어, John이라는 IAM 사용자에게 Amazon EC2 RunInstances 작업을 수행하도록 허용하는 정책을 연결할 수 있습니다. 이 정책은 John이 MyCompany라는 Amazon DynamoDB 테이블에서 항목을 가져오도록 허용되었다는 내용도 명시할 수 있습니다. 또한 John에게 자신의 IAM 보안 자격 증명을 관리하도록 허용할 수도 있습니다. 자격 증명 기반 정책은 [관리형 권한 또는 인라인 권한](#) (p. 312)이 될 수 있습니다.

리소스 기반 정책은 리소스에 연결됩니다. 예를 들어, Amazon S3 버킷, Amazon SQS 대기열 및 AWS Key Management Service 암호화 키에 리소스 기반 정책을 연결할 수 있습니다. 리소스 기반 정책을 지원하는 서비스 목록은 [IAM로 작업하는 AWS 서비스](#) (p. 488) 단원을 참조하십시오. 리소스 기반 정책을 사용하면 이러한 리소스에 액세스할 수 있는 대상 및 해당 대상이 리소스에서 수행할 수 있는 작업을 지정할 수 있습니다. 리소스 기반 정책은 인라인만 있고 관리형은 없습니다.

Note

리소스 기반 정책은 리소스 수준 권한과 다릅니다. 이 주제에서 설명한 바와 같이 리소스 기반 정책은 리소스에 직접 연결할 수 있습니다. 리소스 수준 권한이란 ARN을 사용하여 정책에서 개별 리소스를 지정하는 기능을 말합니다. 리소스 기반 정책은 일부 AWS 서비스에서만 지원됩니다. 리소스 기반 정책 및 리소스 수준 권한을 지원하는 서비스 목록은 [IAM로 작업하는 AWS 서비스](#) (p. 488) 단원을 참조하십시오.

이러한 개념에 대한 이해도를 높이려면 다음 그림 단원을 참조하십시오. 123456789012 계정의 관리자는 JohnSmith, CarlosSalazar 및 MaryMajor 사용자에게 자격 증명 기반 정책을 연결했습니다. 이 정책의 일부 작업은 특정 리소스에서 수행할 수 있습니다. 예를 들어 사용자 JohnSmith는 Resource X에 대해 일부 작업을 수행할 수 있습니다. 이는 자격 증명 기반 정책에서 리소스 수준 권한입니다. 관리자는 또한 리소스 기반 정책을 Resource X, Resource Y 및 Resource Z에 추가했습니다. 리소스 기반 정책을 통해 해당 리소스에 액세스할 수 있는 사용자를 지정할 수 있습니다. 예를 들어 Resource X의 리소스 기반 정책은 JohnSmith 및 MaryMajor 사용자 목록을 표시하고 리소스에 대한 읽기 권한을 허용합니다.

123456789012 계정의 예를 사용하면 다음 사용자가 나열된 작업을 수행할 수 있습니다.

- JohnSmith – John은 Resource X에서 나열 및 읽기 작업을 수행할 수 있습니다. John은 사용자에 대한 자격 증명 기반 정책과 Resource X에 대한 리소스 기반 정책을 통해 이 권한을 부여 받습니다.
- CarlosSalazar – Carlos는 Resource Y에서 나열, 읽기 및 쓰기 작업을 수행할 수 있지만 Resource Z에 대한 액세스는 거부됩니다. Carlos의 자격 증명 기반 정책을 통해 Resource Y에서 나열 및 읽기 작업을 수행할 수 있습니다. Resource Y 리소스 기반 정책을 사용하면 Carlos에게 쓰기 권한도 허용됩니다. 그러나 자격 증명 기반 정책을 통해 Resource Z에 대한 액세스가 허용되더라도 Resource Z 리소스 기

반 정책으로 인해 해당 액세스가 거부됩니다. 명시적 Deny는 Allow를 재정의하므로 Carlos의 Resource z에 대한 액세스가 거부됩니다. 자세한 정보는 [정책 평가 로직 \(p. 531\)](#) 단원을 참조하십시오.

- MaryMajor – Mary는 Resource X, Resource Y 및 Resource z에 대해 나열, 읽기 및 쓰기 작업을 수행 할 수 있습니다. Mary의 자격 증명 기반 정책을 통해 리소스 기반 정책보다 더 많은 리소스에 대해 더 많은 작업을 수행할 수 있지만 액세스를 거부하는 정책은 없습니다.
- ZhangWei – Zhang에게는 Resource z에 대한 모든 액세스 권한이 있습니다. Zhang은 자격 증명 기반 정책이 없지만 Resource z 리소스 기반 정책을 사용하면 리소스에 대한 전체 액세스 권한을 가질 수 있습니다.

자격 증명 기반 정책과 리소스 기반 정책은 모두 권한 정책이며 함께 평가됩니다. 권한 정책만 적용되는 요청의 경우 AWS는 먼저 모든 정책에서 Deny를 확인합니다. 이 정책이 존재하는 경우 요청이 거부됩니다. 그런 다음 AWS는 각 Allow를 확인합니다. 적어도 하나의 정책 설명이 요청의 작업을 허용하는 경우 요청이 허용됩니다. Allow가 자격 증명 기반 정책인지 리소스 기반 정책인지는 중요하지 않습니다.

Important

이 논리는 요청이 하나의 AWS 계정에서 이루어진 경우에만 적용됩니다. 하나의 계정에서 다른 계정으로 요청한 경우 Account A의 요청자는 Account B의 리소스에 대한 요청을 허용하는 자격 증명 기반 정책을 가지고 있어야 합니다. 또한 Account B의 리소스 기반 정책은 Account A의 요청자가 리소스에 액세스할 수 있도록 허용해야 합니다. 두 계정의 정책이 작업을 허용하지 않으면 요청이 실패합니다. 교차 계정 액세스에 대해 리소스 기반 정책을 사용하는 방법에 대한 자세한 정보는 [IAM 역할과 리소스 기반 정책의 차이 \(p. 257\)](#) 단원을 참조하십시오.

특정 권한이 있는 사용자는 해당 권한에 연결된 권한 정책이 있는 리소스를 요청할 수 있습니다. 이 경우 AWS는 해당 리소스에 대한 액세스 권한을 부여할지 여부를 결정할 때 두 권한 세트를 모두 평가합니다. 정책이 평가되는 방식에 대한 자세한 정보는 [정책 평가 로직 \(p. 531\)](#) 단원을 참조하십시오.

Note

Amazon S3는 자격 증명 기반 정책 및 리소스 기반 정책(버킷 정책이라고 함)을 지원합니다. 또한 Amazon S3은 IAM 정책 및 권한과 독립적인 ACL(액세스 제어 목록)이라는 권한 메커니즘을 지원 합니다. IAM 정책을 Amazon S3 ACL과 함께 사용할 수 있습니다. 자세한 정보는 Amazon Simple Storage Service 개발자 가이드의 [액세스 제어](#) 단원을 참조하십시오.

정책을 사용하여 액세스 제어

정책을 사용하여 IAM 또는 모든 AWS 내 리소스에 대한 액세스를 제어할 수 있습니다.

정책 ([p. 305](#))을 사용하여 AWS에서 액세스를 제어하려면 AWS가 액세스를 부여하는 방식을 이해해야 합니다. AWS는 리소스 모음으로 구성되어 있습니다. IAM 사용자는 리소스입니다. Amazon S3 버킷도 리소스입니다. AWS API, AWS CLI 또는 AWS Management 콘솔을 사용하여 작업을 수행할 경우(예: 사용자 생성) 해당 작업에 대한 요청을 전송합니다. 이 요청은 작업, 리소스, 보안 주체 엔터티(사용자 또는 역할), 보안 주체 계정 및 필요한 요청 정보를 지정합니다. 이러한 모든 정보는 콘텍스트를 제공합니다.

그런 다음 AWS는 사용자(보안 주체 엔터티)가 지정된 리소스에 대해 지정된 작업을 수행할 수 있도록 인증(로그인) 및 권한 부여(권한 있음)되었는지 확인합니다. 권한을 부여하는 동안 AWS는 요청 콘텍스트에 적용되는 모든 정책을 확인합니다. 대부분의 정책은 AWS에 [JSON 문서 \(p. 309\)](#)로 저장되며 보안 주체 엔터티에 대한 권한을 지정합니다. 정책 유형 및 활용에 대한 자세한 내용은 [정책 및 권한 \(p. 305\)](#) 단원을 참조하십시오.

AWS는 정책이 요청의 각 부분을 허용한 경우에만 요청에 권한을 부여합니다. 이러한 프로세스의 딜어그램을 보려면 [IAM 작동 방식 이해 \(p. 3\)](#) 단원을 참조하십시오. AWS가 요청이 허용되는지 여부를 결정하는 방법에 대한 자세한 정보는 [정책 평가 로직 \(p. 531\)](#) 단원을 참조하십시오.

IAM 정책을 생성하면 다음에 대한 액세스를 제어할 수 있습니다.

- [보안 주체용 AWS \(p. 328\)](#) – 요청하는 사용자([보안 주체 \(p. 4\)](#))가 수행하도록 허용된 사항을 제어합니다.

- [IAM 자격 증명 \(p. 329\)](#) – 어떤 IAM 자격 증명(그룹, 사용자 및 역할)에 액세스할 수 있는지 및 그 방법을 제어합니다.
- [IAM 정책 \(p. 332\)](#) – 고객 관리형 정책을 생성, 편집 및 삭제할 수 있는 대상과 모든 관리형 정책을 연결하고 분리할 수 있는 대상을 제어합니다.
- [AWS 리소스 \(p. 335\)](#) – 자격 증명 기반 정책 또는 리소스 기반 정책을 사용하여 리소스에 액세스할 수 있는 대상을 제어합니다.
- [AWS 계정 \(p. 335\)](#) – 요청이 특정 계정의 멤버에만 허용되는지 여부를 제어합니다.

이러한 정책을 사용하여 AWS 리소스에 액세스할 수 있는 대상과 액세스한 대상이 리소스에서 수행할 수 있는 작업을 지정할 수 있습니다. 모든 IAM 사용자는 처음에 권한이 없습니다. 다시 말해, 기본적으로 사용자는 아무 작업도 할 수 없으며, 심지어 자신의 액세스 키를 볼 수도 없습니다. 사용자에게 작업을 수행할 권한을 부여하기 위해 사용자에게 권한을 추가(즉 사용자에게 정책 연결)하거나 의도한 권한을 보유한 그룹에 사용자를 추가할 수 있습니다.

예를 들어, 자신의 액세스 키를 나열할 사용자 권한을 부여할 수 있습니다. 해당 권한을 확장하여 각 사용자가 자신의 키를 생성, 업데이트 및 삭제하도록 할 수도 있습니다.

그룹에 권한을 부여하면 그룹에 속한 모든 사용자가 해당 권한을 얻습니다. 예를 들어, Administrators 그룹에 IAM 계정 리소스에서 AWS 작업을 수행할 권한을 부여할 수 있습니다. 또 다른 예로 Managers 그룹에 AWS 계정의 Amazon EC2 인스턴스를 설명할 권한을 부여할 수 있습니다.

사용자, 그룹 및 역할에 기본 권한을 위임하는 방법에 대한 자세한 정보는 [IAM 리소스에 액세스하는 데 필요한 권한 \(p. 443\)](#) 단원을 참조하십시오. 기본 권한을 보여주는 정책의 예를 더 보려면 [IAM 리소스를 관리하기 위한 정책의 예 \(p. 446\)](#) 단원을 참조하십시오.

보안 주체에 대한 액세스 제어

정책을 사용하여 요청하는 사용자(보안 주체)가 수행하도록 허용된 사항을 제어할 수 있습니다. 이렇게 하려면 자격 증명 기반 정책을 해당 사용자의 자격 증명(사용자, 사용자 그룹 또는 역할)에 연결해야 합니다. 또한 [권한 경계 \(p. 317\)](#)를 사용하여 엔터티(사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정할 수 있습니다.

예를 들어 사용자 Zhang Wei의 CloudWatch, Amazon DynamoDB, Amazon EC2 및 Amazon S3에 대한 완전한 액세스를 허용하고자 한다고 가정해보십시오. 다른 사용자에 대해 권한 세트 한 개가 필요한 경우 나중에 분리할 수 있도록 두 가지 다른 정책을 생성할 수 있습니다. 또는 모든 권한을 단일 정책으로 모은 다음 이 정책을 이름이 Zhang Wei인 IAM 사용자에게 연결할 수 있습니다. 정책을 Zhang Wei가 속한 그룹 또는 Zhang Wei가 수임하는 역할에 연결할 수도 있습니다. 그 결과, Zhang이 S3 버킷의 내용을 볼 경우 해당 요청이 허용됩니다. 새 IAM 사용자를 생성하려고 시도할 경우에는 권한이 없으므로 요청이 거부됩니다.

Zhang의 권한 경계를 사용하여 Zhang에게 CompanyConfidential S3 버킷으로의 액세스 권한을 부여해야 합니다. 이렇게 하기 위해서는 Zhang에게 부여하고자 하는 최대 권한을 결정합니다. 이런 경우, Zhang이 그의 권한 정책으로 하는 일을 제어합니다. 여기서는 Zhang이 기밀 버킷으로 액세스하지만 않도록 신경 쓰니다. 따라서 다음 정책을 사용하여 Zhang의 경계를 정의하여 Amazon S3에 대한 모든 AWS 작업 및 몇 가지 기타 서비스를 허용하지만 CompanyConfidential S3 버킷으로의 액세스는 거부합니다. 권한 경계가 모든 IAM 작업을 허용하지 않기 때문에 권한 경계는 Zhang이 그의(또는 어떠한 사람의) 경계를 삭제하지 못하도록 방지합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "SomeServices",  
            "Effect": "Allow",  
            "Action": [  
                "cloudwatch:*",  
                "dynamodb:*",  
                "ec2:*",  
                "s3:ListBucket"  
            ]  
        }  
    ]  
}
```

```
        "s3:*"
    ],
    "Resource": "*"
},
{
    "Sid": "NoConfidentialBucket",
    "Effect": "Deny",
    "Action": "s3:*",
    "Resource": [
        "arn:aws:s3:::CompanyConfidential/*",
        "arn:aws:s3:::CompanyConfidential"
    ]
}
}
```

이 사용자에 대한 권한 경계처럼 정책을 할당할 경우 어떠한 권한도 허용하지 않는다는 점을 유의하십시오. 권한 경계는 자격 증명 기반 정책에서 IAM 엔터티에 부여할 수 있는 최대 권한을 설정합니다. 권한 경계에 대한 자세한 정보는 [IAM 엔터티에 대한 권한 경계 \(p. 317\)](#) 단원을 참조하십시오.

이전 절차에 대한 자세한 정보는 이러한 리소스 단원을 참조하십시오.

- 보안 주체에 연결할 수 있는 IAM 정책의 생성에 대해 자세히 알아보려면 [IAM 정책 만들기 \(p. 377\)](#) 단원을 참조하십시오.
- 보안 주체에 IAM 정책을 연결하는 방법에 대해 자세히 알아보려면 [IAM 자격 증명 권한 추가 및 제거 \(p. 391\)](#) 단원을 참조하십시오.
- EC2에 전체 액세스 권한을 부여하는 예제 정책을 보려면 [Amazon EC2: 특정 리전 내에서의 모든 EC2 액세스를 프로그래밍 방식으로 콘솔에서 허용 \(p. 361\)](#) 단원을 참조하십시오.
- S3 버킷에 읽기 전용 액세스를 허용하려면 [Amazon S3: S3 버킷에 있는 객체에 대한 읽기 및 쓰기 액세스 권한을 프로그래밍 방식으로 콘솔에서 허용 \(p. 376\)](#) 예제 정책의 첫 번째 두 설명문을 사용하십시오.
- 사용자에게 해당 자격 증명을 교체할 수 있도록 허용하는 예제 정책을 보려면 [IAM: 프로그램 방식으로 콘솔에서 IAM 사용자가 자신의 자격 증명을 교체하도록 허용 \(p. 369\)](#) 단원을 참조하십시오.

자격 증명에 대한 액세스 제어

IAM 정책에서 그룹 전반의 모든 사용자에게 연결하는 정책을 생성함으로써 사용자가 자격 증명에 대해 수행할 수 있는 사항을 제어할 수 있습니다. 이렇게 하려면 자격 증명에 수행할 수 있는 사항 또는 자격 증명에 액세스할 수 있는 대상을 제어하는 정책을 생성합니다.

예를 들어, 이름이 AllUsers인 그룹을 생성한 다음 해당 그룹을 모든 사용자에 연결할 수 있습니다. 그룹을 생성할 때 이전 섹션에서 설명한 대로 모든 사용자에게 자격 증명을 교체하기 위한 액세스 권한을 부여할 수 있습니다. 그런 다음 정책 조건에 사용자 이름이 포함되지 않은 경우 그룹을 변경하는 액세스를 거부하는 정책을 생성할 수 있습니다. 그러나 정책에서 이 부분은 나열된 사용자를 제외한 모든 사용자의 액세스만 거부합니다. 또한 그룹 사용자 모두에 대한 모든 그룹 관리 작업을 허용하는 권한을 포함해야 합니다. 마지막으로, 모든 사용자에게 적용되도록 이 정책을 그룹에 연결합니다. 그 결과, 정책에 지정되지 않은 사용자가 그룹을 변경하려고 하면 해당 요청이 거부됩니다.

시각적 편집기를 사용하여 이 정책을 생성하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 왼쪽의 탐색 창에서 정책을 선택합니다.

정책을 처음으로 선택하는 경우 Welcome to Managed Policies 페이지가 나타납니다. [Get Started]를 선택합니다.

3. [Create policy]를 선택합니다.

4. Visual editor(시각적 편집기) 탭에서 Choose a service(서비스 선택)을 선택하여 시작합니다. 그런 다음 IAM을 선택합니다.
5. Select actions(작업 선택)을 선택한 다음 검색 상자에 **group**을 입력합니다. 시각적 편집기에 group이라는 단어가 포함된 모든 IAM 작업이 표시됩니다. 모든 확인란을 선택합니다.
6. 리소스를 선택하여 정책에 대한 리소스를 지정합니다. 선택한 작업에 따라 group, group-path 및 user 리소스 유형이 표시됩니다.
 - group – Add ARN(ARN 추가)를 선택합니다. 리소스에서 모두 선택 옆에 있는 확인란을 선택합니다. Group Name With Path(그룹 이름과 경로)에서 그룹 이름 **AllUsers**를 입력합니다. 그런 다음 추가를 선택합니다.
 - group-path – 모두 선택 옆에 있는 확인란을 선택합니다.
 - user – 모두 선택 옆에 있는 확인란을 선택합니다.

선택한 작업 중 하나인 **ListGroups**는 특정 리소스 사용을 지원하지 않습니다. 해당 작업에서 All resources(모든 리소스)를 선택할 필요가 없습니다. 정책을 저장하거나 JSON 탭에서 정책을 보는 경우 IAM이 모든 리소스에 대해 이 작업 권한을 부여하는 새 권한 블록을 자동으로 생성하는 것을 확인할 수 있습니다.

7. 다른 권한 블록을 추가하려면 Add additional permissions(권한 추가)를 선택합니다.
8. Choose a service(서비스 선택)을 선택한 다음 IAM을 선택합니다.
9. Select actions(작업 선택)을 선택한 다음 Switch to deny permissions(권한 거부로 전환)을 선택합니다. 이렇게 하면 권한을 거부할 때 전체 블록이 사용됩니다.
10. 검색 상자에 **group**을 입력합니다. 시각적 편집기에 group이라는 단어가 포함된 모든 IAM 작업이 표시됩니다. 다음 작업 옆에 있는 확인란을 선택합니다.
 - CreateGroup
 - DeleteGroup
 - RemoveUserFromGroup
 - AttachGroupPolicy
 - DeleteGroupPolicy
 - DetachGroupPolicy
 - PutGroupPolicy
 - UpdateGroup
11. 리소스를 선택하여 정책에 대한 리소스를 지정합니다. 선택한 작업에 따라 group 리소스 유형이 표시됩니다. Add ARN(ARN 추가)를 선택합니다. 리소스에서 모두 선택 옆에 있는 확인란을 선택합니다. Group Name With Path(그룹 이름과 경로)에서 그룹 이름 **AllUsers**를 입력합니다. 그런 다음 추가를 선택합니다.
12. Specify request conditions(optional)(요청 조건 지정(선택 사항))을 선택한 다음 조건 추가를 선택합니다. 다음 값을 사용하여 양식 입력을 완료합니다.
 - 키 – aws:username을 선택합니다.
 - 한정어 – 기본값을 선택합니다.
 - 연산자 – StringNotEquals를 선택합니다.
 - 값 – **srodriguez**를 입력한 다음 Add another condition value(다른 조건 값 추가)를 선택합니다. **mjackson**을 입력한 다음 Add another condition value(다른 조건 값 추가)를 선택합니다. **adesai**를 입력한 다음 추가를 선택합니다.

이 조건은 호출한 사용자가 목록에 포함되지 않은 경우 지정된 그룹 관리 작업 액세스가 거부됩니다. 이는 명시적으로 권한을 거부하므로 해당 사용자가 작업을 호출할 수 있도록 허용된 이전 블록을 무시합니다. 목록에 있는 사용자는 액세스가 거부되지 않으며 첫 번째 권한 블록의 권한이 부여되므로 그룹을 전체적으로 관리할 수 있습니다.

13. 작업이 완료되면 [Review policy]를 선택합니다.

Note

언제든지 Visual editor(시각적 편집기) 및 JSON 탭을 전환할 수 있습니다. 그러나 변경을 수행하거나 Visual editor(시각적 편집기) 탭에서 정책 검토를 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 정보는 [정책 재구성 \(p. 455\)](#) 단원을 참조하십시오.

14. 정책 검토 페이지에서 이름에 **LimitAllUserGroupManagement**를 입력합니다. 설명에 다음을 입력합니다. **Allows all users Read-only access to a specific group, and allows only specific users access to make changes to the group** 정책 요약을 검토하여 의도한 권한을 부여했는지 확인합니다. 그런 다음 정책 생성을 선택하여 새 정책을 저장합니다.
15. 그룹에 정책을 연결합니다. 자세한 정보는 [IAM 자격 증명 권한 추가 및 제거 \(p. 391\)](#) 단원을 참조하십시오.

또는 이러한 예제 JSON 정책 문서를 사용하여 동일한 정책을 생성할 수 있습니다. 자세한 정보는 [the section called "JSON 탭에서 정책 만들기" \(p. 381\)](#) 단원을 참조하십시오.

Example 모든 사용자에게 특정 그룹의 읽기 전용 액세스를 허용하고 특정 사용자에만 그룹을 변경할 수 있는 액세스 권한을 허용하는 예제 정책

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAllUsersToListAllGroups",
            "Effect": "Allow",
            "Action": "iam>ListGroups",
            "Resource": "arn:aws:iam::*:*"
        },
        {
            "Sid": "AllowAllUsersToViewAndManageThisGroup",
            "Effect": "Allow",
            "Action": [
                "iam>CreateGroup",
                "iam>DeleteGroup",
                "iam>ListGroupPolicies",
                "iam>UpdateGroup",
                "iam>GetGroup",
                "iam>RemoveUserFromGroup",
                "iam>AddUserToGroup",
                "iam>ListGroupsForUser",
                "iam>AttachGroupPolicy",
                "iam>DetachGroupPolicy",
                "iam>ListAttachedGroupPolicies",
                "iam>GetGroupPolicy",
                "iam>DeleteGroupPolicy",
                "iam>PutGroupPolicy"
            ],
            "Resource": [
                "arn:aws:iam::*:user/*",
                "arn:aws:iam::*:group/AllUsers"
            ]
        },
        {
            "Sid": "LimitGroupManagementAccessToSpecificUsers",
            "Effect": "Deny",
            "Action": [
                "iam>CreateGroup",
                "iam>RemoveUserFromGroup",
                "iam>DeleteGroup",
                "iam>AttachGroupPolicy",
                "iam>UpdateGroup",
                "iam>GetGroupPolicy"
            ],
            "Resource": [
                "arn:aws:iam::*:user/*"
            ]
        }
    ]
}
```

```
        "iam:DetachGroupPolicy",
        "iam>DeleteGroupPolicy",
        "iam:PutGroupPolicy"
    ],
    "Resource": "arn:aws:iam::*:group/AllUsers",
    "Condition": {
        "StringNotEquals": [
            "aws:username": [
                "srodriguez",
                "mjackson",
                "adesai"
            ]
        ]
    }
}
```

정책에 대한 액세스 제어

사용자가 AWS 관리형 정책을 적용하는 방식을 제어할 수 있습니다. 이렇게 하려면 이 정책을 모든 사용자에게 연결합니다. 이 작업에 그룹을 사용하는 것이 좋습니다.

예를 들어, 사용자가 새 IAM 사용자, 그룹 또는 역할에 [IAMUserChangePassword](#) 및 [PowerUserAccess](#) AWS 관리형 정책만 연결하도록 허용하는 정책을 생성할 수 있습니다.

고객 관리형 정책의 경우 이러한 정책을 생성, 업데이트 및 삭제할 수 있는 대상을 제어할 수 있습니다. 정책을 보안 주체 개체(그룹, 사용자 및 역할)에 연결하고 해당 개체에서 분리할 수 있는 대상을 제어할 수 있습니다. 또한 사용자가 어떤 정책을 어떤 주체에 연결하거나 분리할지 제어할 수 있습니다.

예를 들어 계정 관리자에게 정책을 생성, 업데이트 및 삭제할 권한을 부여할 수 있습니다. 그런 다음 팀 리더 또는 기타 제한된 관리자에게 제한된 관리자가 관리하는 보안 주체 개체에 이러한 정책을 연결하고 분리할 권한을 부여합니다.

자세한 정보는 다음 리소스 단원을 참조하십시오.

- 보안 주체에 연결할 수 있는 IAM 정책의 생성에 대해 자세히 알아보려면 [IAM 정책 만들기 \(p. 377\)](#) 단원을 참조하십시오.
- 보안 주체에 IAM 정책을 연결하는 방법에 대해 자세히 알아보려면 [IAM 자격 증명 권한 추가 및 제거 \(p. 391\)](#) 단원을 참조하십시오.
- 관리형 정책 사용을 제한하는 예제 정책을 보려면 [IAM: IAM 사용자, 그룹 또는 역할에 적용 가능한 관리형 정책을 제한 \(p. 369\)](#) 단원을 참조하십시오.

고객 관리형 정책을 생성, 업데이트 및 삭제할 권한 제어

[IAM 정책 \(p. 305\)](#)을 사용하여 AWS 계정에서 고객 관리형 정책을 생성, 업데이트 및 삭제할 수 있는 대상을 제어할 수 있습니다. 다음 목록에는 정책 또는 정책 버전을 생성, 업데이트 및 삭제하는 것과 직접적으로 관련된 API 작업이 포함되어 있습니다.

- [CreatePolicy](#)
- [CreatePolicyVersion](#)
- [DeletePolicy](#)
- [DeletePolicyVersion](#)
- [SetDefaultPolicyVersion](#)

위 목록의 API 작업은 IAM 정책을 사용하여 허용하거나 거부할 수 있는 작업, 다시 말해 부여할 수 있는 권한에 해당합니다.

다음 예제 정책을 고려하십시오. 사용자가 AWS 계정에서 모든 고객 관리형 정책의 기본 버전을 생성, 업데이트(즉, 새 정책 버전 생성), 삭제 및 설정하도록 허용합니다. 또한 이 정책 에에서는 사용자가 정책을 나열하고 가져오도록 허용합니다. 이 예제 JSON 정책 문서를 사용하여 정책을 생성하는 방법에 대해 자세히 알아보려면 [the section called “JSON 탭에서 정책 만들기” \(p. 381\)](#) 단원을 참조하십시오.

Example 모든 정책의 기본 버전을 생성, 업데이트, 삭제, 나열, 가져오기 및 설정하도록 허용하는 정책

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": [  
            "iam:CreatePolicy",  
            "iam:CreatePolicyVersion",  
            "iam>DeletePolicy",  
            "iam>DeletePolicyVersion",  
            "iam:GetPolicy",  
            "iam:GetPolicyVersion",  
            "iam>ListPolicies",  
            "iam>ListPolicyVersions",  
            "iam:SetDefaultPolicyVersion"  
        ],  
        "Resource": "*"  
    }  
}
```

이러한 API 작업의 사용을 제한하는 정책을 생성하여 지정하는 관리형 정책에만 영향을 줄 수 있습니다. 예를 들어, 특정 고객 관리형 정책의 경우에만 사용자가 기본 버전을 설정하고 정책 버전을 삭제하도록 허용해야 할 수 있습니다. 이렇게 하려면 이러한 권한을 부여하는 정책의 Resource 요소에 정책 ARN을 지정합니다.

다음 예제는 사용자가 정책 버전을 삭제하고 기본 버전을 설정할 수 있는 정책을 보여줍니다. 이런 작업은 /TEAM-A/ 경로를 포함하는 고객 관리형 정책에만 허용됩니다. 고객 관리형 정책 ARN은 그 정책의 Resource 요소에 지정되어 있습니다. (이 예에서 ARN에는 경로와 와일드카드가 포함되어 있으므로 경로 /TEAM-A/를 포함하는 모든 고객 관리형 정책과 일치합니다.) 이 예제 JSON 정책 문서를 사용하여 정책을 생성하는 방법에 대해 자세히 알아보려면 [the section called “JSON 탭에서 정책 만들기” \(p. 381\)](#) 단원을 참조하십시오.

고객 관리형 정책 이름에서 경로를 사용하는 방법에 대한 자세한 정보는 [표시 이름 및 경로 \(p. 480\)](#) 단원을 참조하십시오.

Example 특정 정책의 경우에만 정책 버전 삭제와 기본 버전 설정을 허용하는 정책

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": [  
            "iam>DeletePolicyVersion",  
            "iam:SetDefaultPolicyVersion"  
        ],  
        "Resource": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:policy/TEAM-A/*"  
    }  
}
```

관리형 정책을 연결 및 분리하는 권한 제어

IAM 정책을 사용하여 사용자가 특정 관리형 정책만 사용하도록 허용할 수도 있습니다. 사실상 사용자가 다른 보안 주체에 부여할 수 있는 권한을 제어할 수 있습니다.

다음은 보안 주체 개체에 관리형 정책을 연결하고 분리하는 것과 직접적으로 관련된 API 작업 목록입니다.

- [AttachGroupPolicy](#)
- [AttachRolePolicy](#)
- [AttachUserPolicy](#)
- [DetachGroupPolicy](#)
- [DetachRolePolicy](#)
- [DetachUserPolicy](#)

이러한 API 작업의 사용을 제한하는 정책을 생성하여 지정하는 특정 관리형 정책 및/또는 보안 주체 개체에만 영향을 줄 수 있습니다. 예를 들어, 사용자가 지정하는 관리형 정책에만 연결하도록 허용해야 할 수 있습니다. 또는 사용자가 지정하는 보안 주체 엔터티에만 관리형 정책을 연결하도록 허용해야 할 수 있습니다.

다음 정책 예에서는 사용자가 경로 /TEAM-A/를 포함하는 그룹 및 역할에만 관리형 정책을 연결하도록 허용합니다. 그룹 및 역할 ARN은 정책의 Resource 요소에서 지정됩니다. (이 예에서 ARN에는 경로와 와일드 카드 문자가 포함되어 있으므로 경로 /TEAM-A/를 포함하는 모든 그룹 및 역할과 일치합니다.) 이 예제 JSON 정책 문서를 사용하여 정책을 생성하는 방법에 대해 자세히 알아보려면 [the section called “JSON 탭에서 정책 만들기” \(p. 381\)](#) 단원을 참조하십시오.

Example 특정 그룹 또는 역할에만 관리형 정책 연결을 허용하는 정책

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": [  
            "iam:AttachGroupPolicy",  
            "iam:AttachRolePolicy"  
        ],  
        "Resource": [  
            "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:group/TEAM-A/*",  
            "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:role/TEAM-A/*"  
        ]  
    }  
}
```

앞 예에서 지정하는 정책에만 영향을 주도록 작업의 사용을 제한할 수 있습니다. 정책에 조건을 추가함으로써 사실상 사용자가 다른 보안 주체에 연결할 수 있는 권한을 제어할 수 있습니다.

다음 예에서 조건은 연결된 정책이 지정된 정책 가운데 하나와 일치하는 경우에만 AttachGroupPolicy 및 AttachRolePolicy 권한이 허용되도록 합니다. 이 조건은 [iam:PolicyARN 조건 키 \(p. 510\)](#)를 사용하여 연결할 수 있는 정책을 결정합니다. 다음은 위의 예제를 확장한 예제 정책입니다. 사용자가 경로 /TEAM-A/ 경로를 포함하는 그룹 및 역할에만 /TEAM-A/ 경로를 포함하는 관리형 정책만 연결하도록 허용합니다. 이 예제 JSON 정책 문서를 사용하여 정책을 생성하는 방법에 대해 자세히 알아보려면 [the section called “JSON 탭에서 정책 만들기” \(p. 381\)](#) 단원을 참조하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": [  
            "iam:AttachGroupPolicy",  
            "iam:AttachRolePolicy"  
        ],  
        "Resource": [  
            "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:group/TEAM-A/*",  
            "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:role/TEAM-A/*"  
        ],  
        "Condition": {"ArnLike":  
    }
```

```
        {"iam:PolicyARN": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:policy/TEAM-A/*"}  
    }  
}
```

ARN에 와일드카드 문자가 있으므로 이 정책은 ArnLike 조건 연산자를 사용합니다. 특정 ARN의 경우 ArnEquals 조건 연산자를 사용합니다. ArnLike 및 ArnEquals에 대한 자세한 정보는 정책 요소 참조의 조건 유형 단원에서 [Amazon 리소스 이름\(ARN\) 조건 연산자 \(p. 518\)](#) 단원을 참조하십시오.

예를 들어, 지정하는 관리형 정책만 포함하도록 작업 사용을 제한할 수 있습니다. 이렇게 하려면 이러한 권한을 부여하는 정책의 Condition 요소에 정책 ARN을 지정합니다. 예를 들어, 고객 관리형 정책의 ARN을 지정하려면:

```
"Condition": {"ArnEquals":  
    {"iam:PolicyARN": "arn:aws:iam::123456789012:policy/POLICY-NAME"}  
}
```

AWS 관리형 정책의 Condition 요소에서도 정책의 ARN을 지정할 수 있습니다. AWS 관리형 정책의 ARN은 다음 예와 같이 정책 ARN에 계정 ID 대신 aws라는 특별한 별칭을 사용합니다.

```
"Condition": {"ArnEquals":  
    {"iam:PolicyARN": "arn:aws:iam::aws:policy/AmazonEC2FullAccess"}  
}
```

리소스에 대한 액세스 제어

자격 증명 기반 정책 또는 리소스 기반 정책을 사용하여 리소스에 대한 액세스를 제어할 수 있습니다. 자격 증명 기반 정책에서 자격 증명에 정책을 연결하고 자격 증명이 액세스할 수 있는 리소스를 지정합니다. 리소스 기반 정책에서 제어하려는 리소스에 정책을 연결합니다. 정책에서 해당 리소스에 액세스할 수 있는 보안 주체를 지정합니다. 두 정책 유형에 대한 자세한 정보는 [자격 증명 기반 정책 및 리소스 기반 정책 \(p. 326\)](#) 단원을 참조하십시오.

자세한 정보는 다음 리소스 단원을 참조하십시오.

- 보안 주체에 연결할 수 있는 IAM 정책의 생성에 대해 자세히 알아보려면 [IAM 정책 만들기 \(p. 377\)](#) 단원을 참조하십시오.
- 보안 주체에 IAM 정책을 연결하는 방법에 대해 자세히 알아보려면 [IAM 자격 증명 권한 추가 및 제거 \(p. 391\)](#) 단원을 참조하십시오.
- Amazon S3은 해당 버킷의 리소스 기반 정책 사용을 지원합니다. 자세한 정보는 [버킷 정책 예제](#) 단원을 참조하십시오.

리소스 생성자에게 권한이 자동으로 부여되지는 않음

AWS 계정 루트 사용자 자격 증명을 사용하여 로그인한 경우 해당 계정에 속한 리소스에서 모든 작업을 수행할 수 있는 권한이 부여됩니다. 그러나 IAM 사용자의 경우에는 그렇지 않습니다. IAM 사용자는 리소스를 생성할 권한을 받을 수 있지만, 그러한 리소스에 대한 권한은 명시적으로 부여받은 권한으로 제한됩니다. 즉, IAM 역할과 같은 리소스를 생성했다는 이유만으로 해당 역할을 편집 또는 삭제할 권한이 자동으로 부여되지 않습니다. 또한 사용자의 권한은 계정 소유자 또는 해당 권한을 관리할 권한이 있는 다른 사용자가 언제든지 취소할 수 있습니다.

특정 계정에서 보안 주체에 대한 액세스 제어

계정의 IAM 사용자에게 리소스에 대한 액세스 권한을 직접 부여할 수 있습니다. 다른 계정의 사용자가 리소스에 액세스할 필요가 있다면 IAM 역할을 생성합니다. 역할은 권한을 포함한 개체이지만 특정 사용자와 관련이 없습니다. 다른 계정의 사용자는 해당 역할을 수임하여 해당 역할에 할당된 권한에 따라 리소스에 액세스할 수 있습니다. 자세한 정보는 [자신이 소유한 다른 AWS 계정의 IAM 사용자에 대한 액세스 권한 제공 \(p. 157\)](#) 단원을 참조하십시오.

Note

일부 서비스는 [자격 증명 기반 정책 및 리소스 기반 정책 \(p. 326\)](#)에 나온 것처럼 리소스 기반 정책을 지원합니다(Amazon S3, Amazon SNS, Amazon SQS 등). 그런 서비스의 역할 사용 대안은 공유 할 리소스(버킷, 주제 또는 대기열)에 정책을 연결하는 것입니다. 리소스 기반 정책은 리소스에 대한 액세스 허가를 받은 AWS 계정을 지정할 수 있습니다.

IAM 태그를 사용한 액세스 제어

IAM 태그를 사용하면 키-값 페어의 형태로 사용자 또는 역할에 사용자 지정 속성을 추가할 수 있습니다. IAM 태그에 대한 자세한 정보는 [the section called “엔터티 태그 지정” \(p. 259\)](#) 단원을 참조하십시오. 태그를 사용하여 AWS에서 권한을 제어할 수 있습니다. 즉, 사용자 또는 역할이 수행할 수 있는 작업 또는 사용자 또는 역할 리소스에 대해 수행할 수 있는 작업을 제어할 수 있습니다.

태그를 사용하여 액세스를 제어하려면 AWS의 액세스 허용 방식을 이해해야 합니다. AWS는 리소스의 컬렉션으로 구성되어 있습니다. IAM 사용자는 리소스입니다. Amazon S3 버킷도 리소스입니다. AWS API, AWS CLI 또는 AWS Management 콘솔을 사용하여 작업을 수행할 경우(예: 사용자 생성) 해당 작업에 대한 요청을 전송합니다. 이 요청은 작업, 리소스, 보안 주체 엔터티(사용자 또는 역할), 보안 주체 계정 및 필요한 요청 정보를 지정합니다. 이러한 모든 정보는 컨텍스트를 제공합니다.

그런 다음 AWS는 사용자(보안 주체 엔터티)가 지정된 리소스에 대해 지정된 작업을 수행할 수 있도록 인증(로그인) 및 권한 부여(권한 있음)되었는지 확인합니다. 권한을 부여하는 동안 AWS는 요청 컨텍스트에 적용되는 모든 정책을 확인합니다. 대부분의 정책은 AWS에 [JSON 문서 \(p. 309\)](#)로 저장되며 보안 주체 엔터티에 대한 권한을 지정합니다. 정책 유형 및 활용에 대한 자세한 내용은 [정책 및 권한 \(p. 305\)](#) 단원을 참조하십시오.

AWS는 정책이 요청의 각 부분을 허용한 경우에만 요청에 권한을 부여합니다. 다이어그램을 보고 IAM 인프라에 대해 자세히 알아보려면 [IAM 작동 방식 이해 \(p. 3\)](#) 단원을 참조하십시오. IAM가 요청이 허용되는지 여부를 결정하는 방법에 대한 자세한 내용은 [정책 평가 로직 \(p. 531\)](#) 단원을 참조하십시오.

태그는 리소스에 연결되거나 요청에 전달되거나 요청을 하는 보안 주체에 연결될 수 있기 때문에 이 프로세스를 복잡하게 만들 수 있습니다. 태그를 기반으로 액세스를 제어하려면 정책의 [조건 요소 \(p. 510\)](#)에 태그 정보를 제공하십시오.

IAM 정책을 생성할 때 IAM 태그와, 연관된 태그 조건 키를 사용하여 다음 중 하나를 수행하기 위한 액세스를 제어할 수 있습니다.

- 리소스 (p. 336)** – 리소스에 대한 태그를 기반으로 사용자 또는 역할에 대한 액세스를 제어합니다. 이를 수행하려면 `iam:ResourceTag/key-name` 조건 키를 사용하여 리소스에 연결된 태그를 기반으로 IAM 리소스에 대한 액세스를 허용할지 여부를 결정합니다.
- 요청 (p. 337)** – 어떤 태그가 IAM 요청에 전달될 수 있는지 제어합니다. 이를 수행하려면 `aws:RequestTag/key-name` 조건 키를 사용하여 어떤 태그를 IAM 사용자 또는 역할에서 추가, 변경 또는 제거할 수 있는지 지정합니다.
- 보안 주체 (p. 337)** – 요청을 한 사람(보안 주체)이 자신의 자격 증명에 연결된 태그를 기반으로 수행할 수 있는 권한을 제어합니다. 이렇게 하려면 `aws:PrincipalTag/key-name` 조건 키를 사용하여 요청을 허용하려면 어떤 태그가 보안 주체에 연결되어야 하는지 지정합니다.
- 권한 부여 프로세스의 일부 (p. 338)** – `aws:TagKeys` 조건 키를 사용하여 특정 태그 키를 리소스, 요청 또는 보안 주체에서 사용할 수 있는지 여부를 제어합니다. 이 경우 값은 중요하지 않습니다.

JSON을 사용하거나 기존 관리형 정책을 가져와서 시작적으로 IAM 정책을 생성할 수 있습니다. 자세한 내용은 [IAM 정책 만들기 \(p. 377\)](#) 단원을 참조하십시오.

리소스에 대한 액세스 제어

IAM 정책에 태그를 사용하여 IAM 사용자 및 역할에 대한 액세스를 제어할 수 있습니다. 그러나 IAM은 그룹에 대한 태그를 지원하지 않으므로 태그를 사용하여 그룹에 대한 액세스를 제어할 수 없습니다.

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 **status=terminated** 태그가 지정된 사용자의 삭제를 허용합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iam>DeleteUser",  
            "Resource": "*",  
            "Condition": {"StringLike": {"iam:ResourceTag/status": "terminated"}}  
        }  
    ]  
}
```

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 **jobFunction = employee** 태그가 지정된 모든 사용자에 대해 태그 편집을 허용합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iam>ListUserTags",  
                "iam>TagUser",  
                "iam>UntagUser"  
            ],  
            "Resource": "*",  
            "Condition": {"StringLike": {"iam:ResourceTag/jobFunction": "employee"}}  
        }  
    ]  
}
```

요청에 대한 액세스 제어

IAM 정책에서 태그를 사용하여 IAM 사용자 또는 역할에서 추가, 변경 또는 제거할 수 있는 태그를 제어할 수 있습니다.

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 **department = HR** 또는 **department = CS** 태그만을 사용하는 사용자 태그 지정을 허용합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iam>TagUser",  
            "Resource": "*",  
            "Condition": {"StringLike": {"aws:RequestTag/department": [  
                "HR",  
                "CS"  
            ]}}  
        }  
    ]  
}
```

보안 주체에 대한 액세스 제어

IAM 정책에 태그를 사용하여 요청자(보안 주체)가 자신의 자격 증명에 연결된 태그를 기반으로 수행할 수 있는 작업을 제어할 수 있습니다.

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.를 사용하면 **tagManager=true** 태그가 지정된 사용자가 IAM 사용자, 그룹 또는 역할을 관리할 수 있습니다.이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iam:*",  
            "Resource": "*"  
            "Condition": {"StringEquals": {"aws:PrincipalTag/tagManager": "true"}}  
        }  
    ]  
}
```

태그 키를 사용한 액세스 제어

IAM 정책에서 태그를 사용하여 리소스, 요청 또는 보안 주체에 특정 태그 키를 사용할 수 있는지 여부를 제어 할 수 있습니다.

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.를 사용하면 **project** 키가 있는 태그만 사용자로부터 제거할 수 있습니다.이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": "iam:UntagUser",  
        "Resource": "*"  
        "Condition": {"ForAllValues:StringEquals": {"aws:TagKeys": ["project"]}}  
    }]  
}
```

태그를 사용한 액세스 제어

태그를 사용하여 액세스를 제어하기 전에 AWS의 액세스 허용 방식을 이해해야 합니다. 또한 AWS는 리소스의 컬렉션으로 구성되어 있습니다. IAM 사용자는 리소스입니다. Amazon S3 버킷도 리소스입니다. AWS API, AWS CLI 또는 AWS Management 콘솔을 사용하여 작업을 수행할 경우(예: 사용자 생성) 해당 작업에 대한 요청을 전송합니다. 이 요청은 작업, 리소스, 보안 주체 엔터티(사용자 또는 역할), 보안 주체 계정 및 필요한 요청 정보를 지정합니다. 이러한 모든 정보는 컨텍스트를 제공합니다.

그런 다음 AWS는 사용자(보안 주체 엔터티)가 지정된 리소스에 대해 지정된 작업을 수행할 수 있도록 인증(로그인) 및 권한 부여(권한 있음)되었는지 확인합니다. 권한을 부여하는 동안 AWS는 요청 컨텍스트에 적용되는 모든 정책을 확인합니다. 대부분의 정책은 AWS에 [JSON 문서 \(p. 309\)](#)로 저장되며 보안 주체 엔터티에 대한 권한을 지정합니다. 정책 유형 및 활용에 대한 자세한 내용은 [정책 및 권한 \(p. 305\)](#) 단원을 참조하십시오.

AWS는 정책이 요청의 각 부분을 허용한 경우에만 요청에 권한을 부여합니다. 다이어그램을 보고 IAM 인프라에 대해 자세히 알아보려면 [IAM 작동 방식 이해 \(p. 3\)](#) 단원을 참조하십시오. IAM가 요청이 허용되는지 여부를 결정하는 방법에 대한 자세한 내용은 [정책 평가 로직 \(p. 531\)](#) 단원을 참조하십시오.

태그로 인해 이 프로세스가 복잡해질 수 있는데, 리소스에 태그가 연결되거나 태그 지정을 지원하는 서비스에 대한 요청에 전달될 수 있기 때문입니다. 태그를 기반으로 액세스를 제어하려면 정책의 [조건 요소 \(p. 510\)](#)에 태그 정보를 제공하십시오. AWS 서비스에서 태그 지정을 지원하는지 여부를 알아보려면 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하고, 태그 기반 권한 부여 열이 예인 서비스를 찾아보십시오. 서비스의 이름을 선택하여 해당 서비스에 대한 권한 부여 및 액세스 제어 문서를 봅니다.

IAM 정책을 생성할 때 태그 조건 키를 사용하여 다음에 대한 액세스를 제어할 수 있습니다.

- **리소스 (p. 339)** – 리소스에 대한 태그를 기반으로 AWS 서비스 리소스에 대한 액세스를 제어합니다. 이를 수행하려면 ResourceTag/**key-name** 조건 키를 사용하여 리소스에 연결된 태그를 기반으로 리소스에 대한 액세스를 허용하지 여부를 결정합니다.
- **요청 (p. 339)** – 어떤 태그가 IAM 요청에 전달될 수 있는지 제어합니다. 이를 수행하려면 aws:RequestTag/**key-name** 조건 키를 사용하여 어떤 태그 키 – 값 페어를 리소스에 추가, 변경 또는 제거 할 수 있는지 지정합니다.
- **태그 키 (p. 340)** – aws:TagKeys 조건 키를 사용하여 리소스 또는 요청에서 특정 태그 키를 사용할 수 있는지 여부를 제어합니다.

JSON을 사용하거나 기존 관리형 정책을 가져와서 시작적으로 IAM 정책을 생성할 수 있습니다. 자세한 내용은 [IAM 정책 만들기 \(p. 377\)](#) 단원을 참조하십시오.

리소스에 대한 액세스 제어

IAM 정책의 조건을 사용하여 태그를 기반으로 AWS 리소스에 대한 액세스를 제어할 수 있습니다.

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 Amazon EC2 인스턴스의 시작 또는 중지를 허용하지만, 인스턴스 태그 Owner가 사용자의 사용자 이름의 값과 같은 경우로 제한합니다. 이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:StartInstances",  
                "ec2:StopInstances"  
            ],  
            "Resource": "arn:aws:ec2:*:instance/*",  
            "Condition": {  
                "StringEquals": {"ec2:ResourceTag/Owner": "${aws:username}"}  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:DescribeInstances",  
            "Resource": "*"  
        }  
    ]  
}
```

이 정책을 계정의 IAM 사용자에게 연결할 수 있습니다. 이름이 richard-roe인 사용자가 Amazon EC2 인스턴스를 시작하려 하는 경우 인스턴스에 Owner=richard-roe 또는 owner=richard-roe 태그가 지정되어야 합니다. 그렇지 않은 경우 액세스가 거부됩니다. 태그 키 Owner는 Owner 및 owner 모두와 일치하는데, 조건 키가 대/소문자를 구분하지 않기 때문입니다. 자세한 내용은 [IAM JSON 정책 요소: Condition \(p. 510\)](#) 단원을 참조하십시오.

요청에 대한 액세스 제어

IAM 정책의 조건을 사용하여 어떤 태그를 AWS 리소스에 추가, 변경 또는 제거할 수 있는지 제어할 수 있습니다.

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.를 통해 태그에 environment 키와 preprod 또는 production 값이 포함된 경우에만 Amazon EC2 CreateTags 작업을 사용하여 태그를 인

스턴스에 연결할 수 있습니다. 모범 사례로서 `ForAllValues` 변경자를 `aws:TagKeys` 조건 키와 함께 사용하여 요청에서 키 `environment`만 허용됨을 표시합니다(다른 어떤 태그도 허용되지 않습니다). 이를 통해 사용자가 `environment` 대신 `Environment`를 실수로 사용하는 것과 같이 다른 키를 포함시키는 것을 방지합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": "ec2:CreateTags",  
        "Resource": "arn:aws:ec2:*:instance/*",  
        "Condition": {  
            "StringEquals": {  
                "aws:RequestTag/environment": [  
                    "preprod",  
                    "production"  
                ]  
            },  
            "ForAllValues:StringEquals": {"aws:TagKeys": "environment"}  
        }  
    }  
}
```

태그 키 제어

IAM 정책에서 조건을 사용하여 리소스 또는 요청에 특정 태그 키를 사용할 수 있는지 여부를 제어할 수 있습니다.

모범 사례로서 정책을 사용하여 태그를 사용한 액세스를 제어할 때 [aws:TagKeys 조건 키 \(p. 557\)](#)를 사용해야 합니다. 태그를 지원하는 AWS 서비스를 통해 대소문자만 다른 여러 키 이름을 생성할 수 있습니다(예: Amazon EC2 인스턴스에 `foo=bar1` 및 `Foo=bar2` 태그 지정). 정책 조건에서 키 이름은 대/소문자를 구분하지 않습니다. 따라서 정책의 조건 요소에서 `"ec2:ResourceTag:TagKey1": "Value1"` 지정을 완료한 경우 조건은 이름이 `TagKey1` 또는 `tagkey1`인 리소스 태그 키와 일치하지만 두 가지 모두와 일치하지는 않습니다. 대소문자만 다른 키를 포함한 중복 태그를 방지하려면 `aws:TagKeys` 조건을 사용하여 사용자가 적용할 수 있는 태그 키를 정의합니다.

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다. 이를 통해 Secrets Manager 비밀 생성 및 태그 지정이 가능하지만 태그 키 `environment` 또는 `cost-center`를 포함해야만 합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": [  
            "secretsmanager>CreateSecret",  
            "secretsmanager:TagResource"  
        ],  
        "Resource": "*",  
        "Condition": {  
            "ForAllValues:StringEquals": {  
                "aws:TagKeys": [  
                    "environment",  
                    "cost-center"  
                ]  
            }  
        }  
    }  
}
```

IAM 자격 증명 기반 정책 예제

정책 ([p. 305](#))은 자격 증명이나 리소스와 연결될 때 해당 권한을 정의하는 AWS의 객체입니다. AWS는 보안 주체 엔터티(사용자 또는 역할)가 요청을 보낼 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는지 여부를 결정합니다. 대부분의 정책은 IAM 자격 증명(사용자, 사용자 그룹 또는 역할)에 연결되는 JSON 문서로서 AWS에 저장됩니다. 자격 증명 기반 정책에는 AWS 관리형 정책, 고객 관리형 정책 및 인라인 정책이 포함됩니다. 이러한 예제 JSON 정책 문서를 사용하여 IAM 정책을 생성하는 방법에 대해 자세히 알아보려면 [the section called “JSON 템에서 정책 만들기” \(\[p. 381\]\(#\)\)](#) 단원을 참조하십시오.

기본적으로 모든 요청이 거부되기 때문에 ID가 액세스하려는 서비스, 작업 및 리소스에 대한 액세스 권한을 제공해야 합니다. 또한 IAM 콘솔에서 지정된 작업을 완료하기 위해 액세스를 허용하려면 추가 권한을 제공해야 합니다.

다음의 정책 라이브러리가 IAM ID에 대한 권한을 정의하는 데 도움이 될 수 있습니다. 필요로 하는 정책을 찾은 다음에 [View this policy](#)(이 정책 보기)를 선택하여 해당 정책의 JSON을 확인합니다. JSON 정책 문서를 자체 정책의 템플릿으로 활용할 수 있습니다.

Note

이 참조 설명에 포함시킬 정책을 제출하고자 하는 경우 이 페이지의 하단에 있는 의견 버튼을 사용합니다.

정책 예제: AWS

- 특정 날짜 범위 동안 액세스를 허용합니다. ([이 정책 보기 \(p. 343\)](#).)
- MFA 인증 사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 자격 증명을 관리할 수 있도록 허용합니다. ([이 정책 보기 \(p. 343\)](#).)
- 사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 자격 증명을 관리할 수 있도록 허용합니다. ([이 정책 보기 \(p. 346\)](#).)
- 사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 MFA 디바이스를 관리할 수 있도록 허용합니다. ([이 정책 보기 \(p. 348\)](#).)
- 사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 암호를 관리할 수 있도록 허용합니다. ([이 정책 보기 \(p. 350\)](#).)
- 사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 암호, 액세스 키 및 SSH 퍼블릭 키를 관리할 수 있도록 허용합니다. ([이 정책 보기 \(p. 351\)](#).)
- 특정 날짜 범위 동안 MFA를 사용하는 경우 특정 액세스를 허용합니다. ([이 정책 보기 \(p. 352\)](#).)
- 소스 IP 주소를 기반으로 AWS에 대한 액세스를 거부합니다. ([이 정책 보기 \(p. 353\)](#).)

정책 예제: CodeCommit

- 프로그래밍 방식으로 및 콘솔에서 CodeCommit 리포지토리에 대한 Read 액세스를 허용합니다. ([이 정책 보기 \(p. 353\)](#).)

정책 예제: AWS Data Pipeline

- 사용자가 생성하지 않은 파이프라인에 대한 액세스를 거부합니다. ([이 정책 보기 \(p. 354\)](#).)

정책 예제: Amazon DynamoDB

- 특정 Amazon DynamoDB 테이블에 대한 액세스를 허용합니다. ([이 정책 보기 \(p. 354\)](#).)
- 특정 Amazon DynamoDB 열에 대한 액세스를 허용합니다. ([이 정책 보기 \(p. 355\)](#).)

- Amazon Cognito ID를 기준으로 Amazon DynamoDB에 대한 행 수준 액세스를 허용합니다. ([이 정책 보기 \(p. 356\)](#).)

정책 예제: Amazon EC2

- Amazon EC2 인스턴스가 볼륨을 연결하거나 분리할 수 있도록 허용합니다. ([이 정책 보기 \(p. 357\)](#).)
- 태그를 기반으로 Amazon EBS 볼륨을 Amazon EC2 인스턴스에 연결하거나 분리하도록 허용합니다. ([이 정책 보기 \(p. 357\)](#).)
- 프로그래밍 방식으로 및 콘솔에서 특정 서브넷에 있는 Amazon EC2 인스턴스를 시작하도록 허용합니다. ([이 정책 보기 \(p. 358\)](#).)
- 프로그래밍 방식으로 및 콘솔에서 특정 VPC와 연결된 Amazon EC2 보안 그룹을 관리하도록 허용합니다. ([이 정책 보기 \(p. 358\)](#).)
- 프로그래밍 방식으로 및 콘솔에서 사용자가 태그 지정한 Amazon EC2 인스턴스를 시작하거나 중지하도록 허용합니다. ([이 정책 보기 \(p. 359\)](#).)
- 프로그래밍 방식으로 및 콘솔에서 리소스 및 보안 주체 태그를 기반으로 Amazon EC2 인스턴스를 시작하거나 중지하도록 허용합니다. ([이 정책 보기 \(p. 360\)](#).)
- 리소스 및 보안 주체 태그가 일치할 때 Amazon EC2 인스턴스를 시작하거나 중지하도록 허용합니다. ([이 정책 보기 \(p. 360\)](#).)
- 프로그래밍 방식으로 및 콘솔에서 특정 리전 내의 모든 Amazon EC2 액세스를 허용합니다. ([이 정책 보기 \(p. 361\)](#).)
- 프로그래밍 방식으로 및 콘솔에서 특정 Amazon EC2 인스턴스를 시작하거나 중지하고 특정 보안 그룹을 수정하도록 허용합니다. ([이 정책 보기 \(p. 361\)](#).)
- Amazon EC2 인스턴스 종료를 특정 IP 주소 범위로 제한합니다. ([이 정책 보기 \(p. 362\)](#).)

정책 예제: AWS Identity and Access Management(IAM)

- 정책 시뮬레이터 API에 대한 액세스를 허용합니다. ([이 정책 보기 \(p. 362\)](#).)
- 정책 시뮬레이터 콘솔에 대한 액세스를 허용합니다. ([이 정책 보기 \(p. 363\)](#).)
- 프로그래밍 방식으로 및 콘솔에서 특정 태그를 다른 특정 태그가 있는 IAM 사용자에게 추가하도록 허용합니다. ([이 정책 보기 \(p. 363\)](#).)
- 프로그래밍 방식으로 및 콘솔에서 특정 태그를 IAM 사용자 또는 역할에 추가하도록 허용합니다. ([이 정책 보기 \(p. 364\)](#).)
- 특정 태그가 있는 새 사용자만 생성하도록 허용합니다. ([이 정책 보기 \(p. 365\)](#).)
- 특정 태그를 관리하도록 허용합니다. ([이 정책 보기 \(p. 366\)](#).)
- 특정 경로가 있는 사용자에게 정책 시뮬레이터 API를 사용하도록 허용합니다. ([이 정책 보기 \(p. 366\)](#).)
- 특정 경로가 있는 사용자에게 정책 시뮬레이터 콘솔을 사용하도록 허용합니다. ([이 정책 보기 \(p. 367\)](#).)
- IAM 사용자가 MFA 디바이스를 자체적으로 관리할 수 있도록 허용합니다. ([이 정책 보기 \(p. 367\)](#).)
- 프로그래밍 방식으로 및 콘솔에서 IAM 사용자가 자신의 자격 증명을 교체할 수 있도록 허용합니다. ([이 정책 보기 \(p. 369\)](#).)
- IAM 사용자, 그룹 또는 역할에 적용할 수 있는 관리형 정책을 제한합니다([이 정책 보기 \(p. 369\)](#)).

정책 예제: Amazon RDS

- 특정 리전 내에서 모든 Amazon RDS 데이터베이스 액세스를 허용합니다. ([이 정책 보기 \(p. 370\)](#).)
- 프로그래밍 방식으로 및 콘솔에서 Amazon RDS 데이터베이스를 복원하도록 허용합니다. ([이 정책 보기 \(p. 370\)](#).)
- 태그 소유자에게 자신이 태그 지정한 Amazon RDS 리소스에 대한 모든 액세스를 허용합니다. ([이 정책 보기 \(p. 371\)](#).)

정책 예제: Amazon S3

- Amazon Cognito 사용자가 자신의 Amazon S3 버킷에 있는 객체에 액세스할 수 있도록 허용합니다. ([이 정책 보기 \(p. 372\)](#).)
- 연합된 사용자가 프로그램 방식으로 콘솔에서 Amazon S3에 있는 자신의 홈 디렉터리에 액세스하도록 허용([이 정책 보기 \(p. 373\)](#))
- 프로그래밍 방식으로 및 콘솔에서 IAM 사용자가 Amazon S3에 있는 자신의 홈 디렉터리에 액세스할 수 있도록 허용합니다. ([이 정책 보기 \(p. 374\)](#).)
- 사용자가 단일 Amazon S3 버킷을 관리할 수 있도록 허용하고 기타 모든 AWS 작업 및 리소스를 거부합니다. ([이 정책 보기 \(p. 375\)](#).)
- Amazon S3 버킷에 있는 객체에 대한 Write 및 Read 액세스를 허용합니다. ([이 정책 보기 \(p. 376\)](#).)
- Amazon S3 버킷에 있는 객체에 대한 Read 및 Write 액세스를 프로그래밍 방식으로 콘솔에서 허용합니다. ([이 정책 보기 \(p. 376\)](#).)

AWS: 특정 기간 동안 액세스 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.는 SERVICE-NAME라는 서비스의 ACTION-NAME 작업에 대한 액세스를 허용합니다. 액세스는 2017년 7월 1일과 2017년 12월 31일(UTC) 사이에 발생하는 작업으로 제한됩니다. 이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체 하십시오.

IAM 정책의 Condition 블록 내에서 복수 조건을 사용하는 방법에 관한 자세한 내용은 [다수의 조건 값 \(p. 511\)](#) 단원을 참조하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": "service-prefix:action-name",  
        "Resource": "*",  
        "Condition": {  
            "DateGreaterThanOrEqual": {"aws:CurrentTime": "2017-07-01T00:00:00Z"},  
            "DateLessThan": {"aws:CurrentTime": "2017-12-31T23:59:59Z"}  
        }  
    }  
}
```

AWS: MFA 인증 IAM 사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 자격 증명을 관리할 수 있도록 허용합니다.

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.는 멀티 팩터 인증(MFA)을 사용하여 인증된 IAM 사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 자격 증명을 관리할 수 있도록 허용합니다. 이 AWS Management 콘솔 페이지에는 계정 ID 및 정식 사용자 ID와 같은 계정 정보가 표시됩니다. 또한 사용자는 자신의 암호, 액세스 키, MFA 디바이스, X.509 인증서, SSH 키 및 Git 자격 증명을 보고 편집할 수 있습니다. 이 예제 정책에는 페이지에 있는 모든 정보를 보고 편집하는데 필요한 권한이 포함되어 있습니다. 또한 사용자가 AWS에서 다른 작업을 수행하기 전에 MFA 사용을 설정하고 인증해야 합니다. 사용자가 MFA를 사용하지 않고 자신의 자격 증명을 관리하도록 허용하려면 [AWS: IAM 사용자가 My Security Credentials\(내 보안 자격 증명\) 페이지에서 자신의 자격 증명을 관리할 수 있도록 허용합니다. \(p. 346\)](#) 단원을 참조하십시오.

사용자가 My Security Credentials(내 보안 자격 증명) 페이지에 액세스할 수 있는 방법을 알아보려면 [IAM 사용자가 자신의 암호를 변경하는 방법\(콘솔\) \(p. 87\)](#) 단원을 참조하십시오.

Note

이 정책 예제에서는 사용자가 로그인과 암호 재설정을 한 번에 할 수 없습니다. 새 사용자와 암호가 만료된 사용자는 이 작업을 시도할 수 있습니다. `iam:ChangePassword`를 `DenyAllExceptListedIfNoMFA` 문에 추가하여 이 작업을 허용할 수 있습니다. 그러나 IAM에서 이 방식은 권장하지 않습니다. 사용자가 MFA 없이 암호를 변경하도록 허용하면 보안 위험이 발생할 수 있습니다.

이 정책이 하는 일은 무엇입니까?

- `AllowViewAccountInfo` 문은 사용자가 계정 수중 정보를 볼 수 있도록 허용합니다. 이러한 권한은 리소스 ARN을 지원하지 않거나 리소스 ARN을 지정하는 데 필요하지 않기 때문에 자신의 문에 포함되어 있어야 합니다. 권한 대신 "Resource" : "*"를 지정합니다. 이 문에는 사용자가 특정 정보를 볼 수 있도록 허용하는 다음 작업이 포함되어 있습니다.
 - `GetAccountSummary` – 계정 ID 및 계정 정식 사용자 ID를 봅니다.
 - `GetAccountPasswordPolicy` – 자신의 IAM 사용자 암호를 변경하는 동안 계정 암호 요구 사항을 봅니다.
 - `ListVirtualMFADevices` – 사용자에 대해 활성화된 가상 MFA 디바이스에 대한 세부 정보를 봅니다.
- `AllowManageOwnPasswords` 문은 사용자가 자신의 암호를 변경할 수 있도록 허용합니다. 또한 이 문에는 My Security Credentials(내 보안 자격 증명) 페이지에 있는 대부분의 정보를 보는 데 필요한 `GetUser` 작업도 포함되어 있습니다.
- `AllowManageOwnAccessKeys` 문은 사용자가 자신의 액세스 키를 생성, 업데이트 및 삭제할 수 있도록 허용합니다.
- `AllowManageOwnSigningCertificates` 문은 사용자가 자신의 서명 인증서를 업로드, 업데이트 및 삭제할 수 있도록 허용합니다.
- `AllowManageOwnSSHPublicKeys` 문은 사용자가 CodeCommit에 대한 자신의 SSH 퍼블릭 키를 업로드, 업데이트 및 삭제할 수 있도록 허용합니다.
- `AllowManageOwnGitCredentials` 문은 사용자가 CodeCommit에 대한 자신의 Git 자격 증명을 생성, 업데이트 및 삭제할 수 있도록 허용합니다.
- `AllowManageOwnVirtualMFADevice` 문은 사용자가 에 대한 자신의 가상 MFA 디바이스를 생성하고 삭제할 수 있도록 허용합니다. 이 문의 리소스 ARN은 현재 로그인한 사용자와 동일한 이름의 MFA 디바이스에만 액세스를 허용합니다. 사용자는 자신의 것이 아닌 다른 가상 MFA 디바이스를 생성하거나 삭제할 수 없습니다.
- `AllowManageOwnUserMFA` 문은 사용자가 자신의 사용자에 대해 가상, U2F 또는 하드웨어 MFA 디바이스를 보거나 관리할 수 있도록 허용합니다. 이 문의 리소스 ARN은 사용자 자신의 IAM 사용자에 대한 액세스만 허용합니다. 사용자는 다른 사용자의 MFA 디바이스를 보거나 관리할 수 없습니다.
- `DenyAllExceptListedIfNoMFA` 문은 사용자가 MFA를 사용하여 로그인하지 않은 경우에만 몇 가지 나열된 작업을 제외한 모든 AWS의 모든 작업에 대한 액세스를 거부합니다. 이 문은 "Deny" 및 "NotAction"의 조합을 사용하여 나열되지 않은 모든 작업에 대한 액세스를 명시적으로 거부합니다. 나열된 항목은 이 문에 따라 거부되거나 허용되지 않습니다. 하지만 정책의 다른 문에서 작업이 허용됩니다. 이 문의 로직에 대한 자세한 내용은 [NotAction 및 Deny \(p. 507\)](#) 단원을 참조하십시오. 사용자가 MFA를 사용하여 로그인한 경우 Condition 테스트가 실패하며 이 문은 어떠한 작업도 거부하지 않습니다. 이 경우 사용자에 대한 다른 정책 또는 문에 따라 사용자의 권한이 결정됩니다.

이 문을 사용하면 MFA를 사용하여 로그인하지 않은 사용자는 나열된 작업만 수행할 수 있습니다. 또한 사용자는 다른 문 또는 정책이 해당 작업에 대한 액세스를 허용하는 경우에만 나열된 작업을 수행할 수 있습니다. MFA 권한 부여가 없으면 `iam:ChangePassword` 작업이 허용되지 않기 때문에 사용자는 로그인 시 암호를 생성할 수 없습니다.

...`IfExists` 키를 분실했을 경우 `Bool` 연산자의 `aws:MultiFactorAuthPresent` 버전은 조건이 `true`로 반환됩니다. 즉, 액세스 키와 같은 장기 자격 증명으로 API를 액세스하는 사용자는 비 IAM API 작업에 대한 액세스가 거부됩니다.

이 정책은 사용자가 IAM 콘솔에서 Users(사용자) 페이지를 보거나 이 페이지를 사용하여 자신의 사용자 정보에 액세스할 수 있도록 허용합니다. 이 작업을 허용하려면 `iam>ListUsers` 작업을

AllowViewAccountInfo 문과 DenyAllExceptListedIfNoMFA 문에 추가합니다. 또한 이 문은 사용자가 자신의 사용자 페이지에서 암호를 변경하도록 허용하지 않습니다. 이 작업을 허용하려면 iam:CreateLoginProfile, iam>DeleteLoginProfile, iam:GetLoginProfile 및 iam:UpdateLoginProfile 작업을 AllowManageOwnPasswords 문에 추가합니다. 또한 사용자가 MFA를 사용하여 로그인하지 않고 자신의 사용자 페이지에서 자신의 암호를 변경할 수 있도록 허용하려면 iam:CreateLoginProfile 작업을 DenyAllExceptListedIfNoMFA 문에 추가합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowViewAccountInfo",  
            "Effect": "Allow",  
            "Action": [  
                "iam:GetAccountPasswordPolicy",  
                "iam:GetAccountSummary",  
                "iam>ListVirtualMFADevices"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Sid": "AllowManageOwnPasswords",  
            "Effect": "Allow",  
            "Action": [  
                "iam:ChangePassword",  
                "iam:GetUser"  
            ],  
            "Resource": "arn:aws:iam::*:user/${aws:username}"  
        },  
        {  
            "Sid": "AllowManageOwnAccessKeys",  
            "Effect": "Allow",  
            "Action": [  
                "iam>CreateAccessKey",  
                "iam>DeleteAccessKey",  
                "iam>ListAccessKeys",  
                "iam:UpdateAccessKey"  
            ],  
            "Resource": "arn:aws:iam::*:user/${aws:username}"  
        },  
        {  
            "Sid": "AllowManageOwnSigningCertificates",  
            "Effect": "Allow",  
            "Action": [  
                "iam>DeleteSigningCertificate",  
                "iam>ListSigningCertificates",  
                "iam:UpdateSigningCertificate",  
                "iam:UploadSigningCertificate"  
            ],  
            "Resource": "arn:aws:iam::*:user/${aws:username}"  
        },  
        {  
            "Sid": "AllowManageOwnSSHPublicKeys",  
            "Effect": "Allow",  
            "Action": [  
                "iam>DeleteSSHPublicKey",  
                "iam:GetSSHPublicKey",  
                "iam>ListSSHPublicKeys",  
                "iam:UpdateSSHPublicKey",  
                "iam:UploadSSHPublicKey"  
            ],  
            "Resource": "arn:aws:iam::*:user/${aws:username}"  
        },  
    ]  
}
```

```
"Sid": "AllowManageOwnGitCredentials",
"Effect": "Allow",
>Action": [
    "iam>CreateServiceSpecificCredential",
    "iam>DeleteServiceSpecificCredential",
    "iam>ListServiceSpecificCredentials",
    "iam>ResetServiceSpecificCredential",
    "iam>UpdateServiceSpecificCredential"
],
"Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
"Sid": "AllowManageOwnVirtualMFADevice",
"Effect": "Allow",
>Action": [
    "iam>CreateVirtualMFADevice",
    "iam>DeleteVirtualMFADevice"
],
"Resource": "arn:aws:iam::*:mfa/${aws:username}"
},
{
"Sid": "AllowManageOwnUserMFA",
"Effect": "Allow",
>Action": [
    "iam>DeactivateMFADevice",
    "iam>EnableMFADevice",
    "iam>ListMFADevices",
    "iam>ResyncMFADevice"
],
"Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
"Sid": "DenyAllExceptListedIfNoMFA",
"Effect": "Deny",
"NotAction": [
    "iam>CreateVirtualMFADevice",
    "iam>EnableMFADevice",
    "iam GetUser",
    "iam>ListMFADevices",
    "iam>ListVirtualMFADevices",
    "iam>ResyncMFADevice",
    "sts:GetSessionToken"
],
"Resource": "*",
"Condition": {
    "BoolIfExists": {
        "aws:MultiFactorAuthPresent": "false"
    }
}
}
]
```

AWS: IAM 사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 자격 증명을 관리할 수 있도록 허용합니다.

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.는 IAM 사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 모든 자격 증명을 관리할 수 있도록 허용합니다. 이 AWS Management 콘솔 페이지에는 계정 ID 및 정식 사용자 ID와 같은 계정 정보가 표시됩니다. 또한 사용자는 자신의 암호, 액세스 키, X.509 인증서, SSH 키, Git 자격 증명을 보고 편집할 수 있습니다. 이 예제 정책에는 사용자의 MFA 디바이스를 제외하고 페이지에 있는 모든 정보를 보고 편집하는 데 필요한 권한이 포함되어 있습니다. 사용자가 MFA를 사용하여 자신의 자격 증명을 관리하도록 허용하려면 [AWS: MFA 인증 IAM 사용자](#)

가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 자격 증명을 관리할 수 있도록 허용합니다. (p. 343) 단원을 참조하십시오.

사용자가 My Security Credentials(내 보안 자격 증명) 페이지에 액세스할 수 있는 방법을 알아보려면 IAM 사용자가 자신의 암호를 변경하는 방법(콘솔) (p. 87) 단원을 참조하십시오.

이 정책이 하는 일은 무엇입니까?

- AllowViewAccountInfo 문은 사용자가 계정 수중 정보를 볼 수 있도록 허용합니다. 이러한 권한은 리소스 ARN을 지원하지 않거나 리소스 ARN을 지정하는 데 필요하지 않기 때문에 자신의 문에 포함되어 있어야 합니다. 권한 대신 "Resource" : "*"를 지정합니다. 이 문에는 사용자가 특정 정보를 볼 수 있도록 허용하는 다음 작업이 포함되어 있습니다.
 - GetAccountPasswordPolicy – 자신의 IAM 사용자 암호를 변경하는 동안 계정 암호 요구 사항을 봅니다.
 - GetAccountSummary – 계정 ID 및 계정 정식 사용자 ID를 봅니다.
- AllowManageOwnPasswords 문은 사용자가 자신의 암호를 변경할 수 있도록 허용합니다. 또한 이 문에는 My Security Credentials(내 보안 자격 증명) 페이지에 있는 대부분의 정보를 보는 데 필요한 GetUser 작업도 포함되어 있습니다.
- AllowManageOwnAccessKeys 문은 사용자가 자신의 액세스 키를 생성, 업데이트 및 삭제할 수 있도록 허용합니다.
- AllowManageOwnSigningCertificates 문은 사용자가 자신의 서명 인증서를 업로드, 업데이트 및 삭제할 수 있도록 허용합니다.
- AllowManageOwnSSHPublicKeys 문은 사용자가 CodeCommit에 대한 자신의 SSH 퍼블릭 키를 업로드, 업데이트 및 삭제할 수 있도록 허용합니다.
- AllowManageOwnGitCredentials 문을 사용하면 사용자가 CodeCommit에 대한 자신의 Git 자격 증명을 생성, 업데이트 및 삭제할 수 있습니다.

이 정책은 사용자가 자신의 MFA 디바이스를 보거나 관리하도록 허용하지 않습니다. 또한 사용자는 IAM 콘솔에서 Users(사용자) 페이지를 보거나 이 페이지를 사용하여 자신의 사용자 정보에 액세스할 수 없습니다. 이 작업을 허용하려면 iam>ListUsers 작업을 AllowViewAccountInfo 문에 추가합니다. 또한 이 문은 사용자가 자신의 사용자 페이지에서 암호를 변경하도록 허용하지 않습니다. 이 작업을 허용하려면 iam>CreateLoginProfile, iam>DeleteLoginProfile, iam:GetLoginProfile 및 iam:UpdateLoginProfile 작업을 AllowManageOwnPasswords 문에 추가합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowViewAccountInfo",  
            "Effect": "Allow",  
            "Action": [  
                "iam:GetAccountPasswordPolicy",  
                "iam:GetAccountSummary"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Sid": "AllowManageOwnPasswords",  
            "Effect": "Allow",  
            "Action": [  
                "iam:ChangePassword",  
                "iam:GetUser"  
            ],  
            "Resource": "arn:aws:iam::*:user/${aws:username}"  
        },  
        {  
            "Sid": "AllowManageOwnAccessKeys",  
            "Effect": "Allow",  
            "Action": [  
                "iam:CreateAccessKey",  
                "iam:DeleteAccessKey",  
                "iam:ListAccessKeys"  
            ],  
            "Resource": "arn:aws:iam::*:user/${aws:username}"  
        }  
    ]  
}
```

```
"Action": [
    "iam:CreateAccessKey",
    "iam>DeleteAccessKey",
    "iam>ListAccessKeys",
    "iam>UpdateAccessKey"
],
"Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
"Sid": "AllowManageOwnSigningCertificates",
"Effect": "Allow",
>Action": [
    "iam>DeleteSigningCertificate",
    "iam>ListSigningCertificates",
    "iam>UpdateSigningCertificate",
    "iam>UploadSigningCertificate"
],
"Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
"Sid": "AllowManageOwnSSHPublicKeys",
"Effect": "Allow",
>Action": [
    "iam>DeleteSSHPublicKey",
    "iam>GetSSHPublicKey",
    "iam>ListSSHPublicKeys",
    "iam>UpdateSSHPublicKey",
    "iam>UploadSSHPublicKey"
],
"Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
"Sid": "AllowManageOwnGitCredentials",
"Effect": "Allow",
>Action": [
    "iam>CreateServiceSpecificCredential",
    "iam>DeleteServiceSpecificCredential",
    "iam>ListServiceSpecificCredentials",
    "iam>ResetServiceSpecificCredential",
    "iam>UpdateServiceSpecificCredential"
],
"Resource": "arn:aws:iam::*:user/${aws:username}"
}
]
```

AWS: MFA 인증 IAM 사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 MFA 디바이스를 관리할 수 있도록 허용합니다.

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다. 멀티 팩터 인증(MFA)을 통해 인증된 IAM 사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 MFA 디바이스를 관리할 수 있도록 허용합니다. 이 AWS Management 콘솔 페이지에는 계정 및 사용자 정보가 표시되지만, 사용자는 자신의 MFA 디바이스만 보고 편집할 수 있습니다. 사용자가 MFA를 사용하여 자신의 모든 자격 증명을 관리하도록 허용하려면 [AWS: MFA 인증 IAM 사용자가 My Security Credentials\(내 보안 자격 증명\) 페이지에서 자신의 자격 증명을 관리할 수 있도록 허용합니다.](#) (p. 343) 단원을 참조하십시오.

사용자가 My Security Credentials(내 보안 자격 증명) 페이지에 액세스할 수 있는 방법을 알아보려면 [IAM 사용자가 자신의 암호를 변경하는 방법\(콘솔\)](#) (p. 87) 단원을 참조하십시오.

이 정책이 하는 일은 무엇입니까?

- `AllowViewAccountInfo` 문은 사용자가 사용자에 대해 활성화된 가상 MFA 디바이스에 대한 세부 정보를 볼 수 있도록 허용합니다. 이 권한은 리소스 ARN 지정을 지원하지 않으므로 자신의 문에 들어 있어야 합니다. 그 대신 "Resource" : "*"를 지정해야 합니다.
- `AllowManageOwnVirtualMFADevice` 문은 사용자가对自己的 가상 MFA 디바이스를 생성하고 삭제할 수 있도록 허용합니다. 이 문의 리소스 ARN은 현재 로그인한 사용자와 동일한 이름의 MFA 디바이스에만 액세스를 허용합니다. 사용자는 자신의 것이 아닌 다른 가상 MFA 디바이스를 생성하거나 삭제할 수 없습니다.
- `AllowManageOwnUserMFA` 문은 사용자가 자신의 가상, U2F 또는 하드웨어 MFA 디바이스를 보거나 관리할 수 있도록 허용합니다. 이 문의 리소스 ARN은 사용자 자신의 IAM 사용자에 대한 액세스만 허용합니다. 사용자는 다른 사용자의 MFA 디바이스를 보거나 관리할 수 없습니다.
- `DenyAllExceptListedIfNoMFA` 문은 사용자가 MFA를 사용하여 로그인하지 않은 경우에만 몇 가지 나열된 작업을 제외한 모든 AWS의 모든 작업에 대한 액세스를 거부합니다. 이 문은 "Deny" 및 "NotAction"의 조합을 사용하여 나열되지 않은 모든 작업에 대한 액세스를 명시적으로 거부합니다. 나열된 항목은 이 문에 따라 거부되거나 허용되지 않습니다. 하지만 정책의 다른 문에서 작업이 허용됩니다. 이 문의 로직에 대한 자세한 내용은 [NotAction 및 Deny \(p. 507\)](#) 단원을 참조하십시오. 사용자가 MFA를 사용하여 로그인한 경우 Condition 테스트가 실패하며 이 문은 어떠한 작업도 거부하지 않습니다. 이 경우 사용자에 대한 다른 정책 또는 문에 따라 사용자의 권한이 결정됩니다.

이 문을 사용하면 MFA를 사용하여 로그인하지 않은 사용자는 나열된 작업만 수행할 수 있습니다. 또한 사용자는 다른 문 또는 정책이 해당 작업에 대한 액세스를 허용하는 경우에만 나열된 작업을 수행할 수 있습니다.

...`IfExists` 키를 분실했을 경우 `Bool` 연산자의 `aws:MultiFactorAuthPresent` 버전은 조건이 `true`로 반환됩니다. 따라서 액세스 키와 같은 장기 자격 증명을 사용하여 API 작업에 액세스하는 사용자는 비 IAM API 작업에 대한 액세스가 거부됩니다.

이 정책은 사용자가 IAM 콘솔에서 `Users(사용자)` 페이지를 보거나 이 페이지를 사용하여 자신의 사용자 정보에 액세스할 수 있도록 허용합니다. 이 작업을 허용하려면 `iam>ListUsers` 작업을 `AllowViewAccountInfo` 문과 `DenyAllExceptListedIfNoMFA` 문에 추가합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowViewAccountInfo",
            "Effect": "Allow",
            "Action": "iam>ListVirtualMFADevices",
            "Resource": "*"
        },
        {
            "Sid": "AllowManageOwnVirtualMFADevice",
            "Effect": "Allow",
            "Action": [
                "iam>CreateVirtualMFADevice",
                "iam>DeleteVirtualMFADevice"
            ],
            "Resource": "arn:aws:iam::*:mfa/${aws:username}"
        },
        {
            "Sid": "AllowManageOwnUserMFA",
            "Effect": "Allow",
            "Action": [
                "iam>DeactivateMFADevice",
                "iam>EnableMFADevice",
                "iam GetUser",
                "iam>ListMFADevices",
                "iam>ResyncMFADevice"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        }
    ]
}
```

```
{  
    "Sid": "DenyAllExceptListedIfNoMFA",  
    "Effect": "Deny",  
    "NotAction": [  
        "iam:CreateVirtualMFADevice",  
        "iam:EnableMFADevice",  
        "iam:GetUser",  
        "iam>ListMFADevices",  
        "iam>ListVirtualMFADevices",  
        "iam:ResyncMFADevice",  
        "sts:GetSessionToken"  
    ],  
    "Resource": "*",  
    "Condition": {  
        "BoolIfExists": {"aws:MultiFactorAuthPresent": "false"}  
    }  
}  
]  
}
```

AWS: IAM 사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 콘솔 암호를 변경할 수 있도록 허용합니다.

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.는 IAM 사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 AWS Management 콘솔 암호를 변경할 수 있도록 허용합니다. 이 AWS Management 콘솔 페이지에는 계정 및 사용자 정보가 표시되지만, 사용자는 자신의 암호에만 액세스할 수 있습니다. 사용자가 MFA를 사용하여 자신의 모든 자격 증명을 관리하도록 허용하려면 [AWS: MFA 인증 IAM 사용자가 My Security Credentials\(내 보안 자격 증명\) 페이지에서 자신의 자격 증명을 관리할 수 있도록 허용합니다.](#) (p. 343) 단원을 참조하십시오. 사용자가 MFA를 사용하지 않고 자신의 자격 증명을 관리하도록 허용하려면 [AWS: IAM 사용자가 My Security Credentials\(내 보안 자격 증명\) 페이지에서 자신의 자격 증명을 관리할 수 있도록 허용합니다.](#) (p. 346) 단원을 참조하십시오.

사용자가 My Security Credentials(내 보안 자격 증명) 페이지에 액세스할 수 있는 방법을 알아보려면 [IAM 사용자가 자신의 암호를 변경하는 방법\(콘솔\)](#) (p. 87) 단원을 참조하십시오.

이 정책이 하는 일은 무엇입니까?

- `ViewAccountPasswordRequirements` 문은 사용자가 자신의 IAM 사용자 암호를 변경하는 동안 계정 암호 요구 사항을 볼 수 있도록 허용합니다.
- `ChangeOwnPassword` 문은 사용자가 자신의 암호를 변경할 수 있도록 허용합니다. 또한 이 문에는 My Security Credentials(내 보안 자격 증명) 페이지에 있는 대부분의 정보를 보는 데 필요한 `GetUser` 작업도 포함되어 있습니다.

이 정책은 사용자가 IAM 콘솔에서 Users(사용자) 페이지를 보거나 이 페이지를 사용하여 자신의 사용자 정보에 액세스할 수 있도록 허용합니다. 이 작업을 허용하려면 `iam>ListUsers` 작업을 `ViewAccountPasswordRequirements` 문에 추가합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "ViewAccountPasswordRequirements",  
            "Effect": "Allow",  
            "Action": "iam:GetAccountPasswordPolicy",  
            "Resource": "*"  
        },  
        {  
            "Sid": "ChangeOwnPassword",  
            "Effect": "Allow",  
            "Action": "iam:ChangeOwnPassword",  
            "Resource": "*"  
        }  
    ]  
}
```

```
    "Action": [
        "iam:GetUser",
        "iam:ChangePassword"
    ],
    "Resource": "arn:aws:iam::*:user/${aws:username}"
}
}
```

AWS: IAM 사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 암호, 액세스 키 및 SSH 퍼블릭 키를 관리할 수 있도록 허용합니다.

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.는 IAM 사용자가 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 암호, 액세스 키 및 X.509 인증서를 관리할 수 있도록 허용합니다. 이 AWS Management 콘솔 페이지에는 계정 ID 및 정식 사용자 ID와 같은 계정 정보가 표시됩니다. 또한 사용자는 자신의 암호, 액세스 키, MFA 디바이스, X.509 인증서, SSH 키 및 Git 자격 증명을 보고 편집할 수 있습니다. 이 예제 정책에는 자신의 암호, 액세스 키 및 X.509 인증서만 보고 편집하는 데 필요한 권한이 포함되어 있습니다. 사용자가 MFA를 사용하여 자신의 모든 자격 증명을 관리하도록 허용하려면 [AWS: MFA 인증 IAM 사용자가 My Security Credentials\(내 보안 자격 증명\) 페이지에서 자신의 자격 증명을 관리할 수 있도록 허용합니다.](#) (p. 343) 단원을 참조하십시오. 사용자가 MFA를 사용하지 않고 자신의 자격 증명을 관리하도록 허용하려면 [AWS: IAM 사용자가 My Security Credentials\(내 보안 자격 증명\) 페이지에서 자신의 자격 증명을 관리할 수 있도록 허용합니다.](#) (p. 346) 단원을 참조하십시오.

사용자가 My Security Credentials(내 보안 자격 증명) 페이지에 액세스할 수 있는 방법을 알아보려면 [IAM 사용자가 자신의 암호를 변경하는 방법\(콘솔\)](#) (p. 87) 단원을 참조하십시오.

이 정책이 하는 일은 무엇입니까?

- AllowViewAccountInfo 문은 사용자가 계정 수중 정보를 볼 수 있도록 허용합니다. 이러한 권한은 리소스 ARN을 지원하지 않거나 리소스 ARN을 지정하는 데 필요하지 않기 때문에 자신의 문에 포함되어 있어야 합니다. 권한 대신 "Resource" : "*"를 지정합니다. 이 문에는 사용자가 특정 정보를 볼 수 있도록 허용하는 다음 작업이 포함되어 있습니다.
 - GetAccountPasswordPolicy – 자신의 IAM 사용자 암호를 변경하는 동안 계정 암호 요구 사항을 봅니다.
 - GetAccountSummary – 계정 ID 및 계정 정식 사용자 ID를 봅니다.
- AllowManageOwnPasswords 문은 사용자가 자신의 암호를 변경할 수 있도록 허용합니다. 또한 이 문에는 My Security Credentials(내 보안 자격 증명) 페이지에 있는 대부분의 정보를 보는 데 필요한 GetUser 작업도 포함되어 있습니다.
- AllowManageOwnAccessKeys 문은 사용자가 자신의 액세스 키를 생성, 업데이트 및 삭제할 수 있도록 허용합니다.
- AllowManageOwnSSHPublicKeys 문은 사용자가 CodeCommit에 대한 자신의 SSH 퍼블릭 키를 업로드, 업데이트 및 삭제할 수 있도록 허용합니다.

이 정책은 사용자가 자신의 MFA 디바이스를 보거나 관리하도록 허용하지 않습니다. 또한 사용자는 IAM 콘솔에서 Users(사용자) 페이지를 보거나 이 페이지를 사용하여 자신의 사용자 정보에 액세스할 수 없습니다. 이 작업을 허용하려면 iam>ListUsers 작업을 AllowViewAccountInfo 문에 추가합니다. 또한 이 문은 사용자가 자신의 사용자 페이지에서 암호를 변경하도록 허용하지 않습니다. 이 작업을 허용하려면 iam>CreateLoginProfile, iam>DeleteLoginProfile, iam>GetLoginProfile 및 iam>UpdateLoginProfile 작업을 AllowManageOwnPasswords 문에 추가합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "iam:GetUser",
                "iam:ChangePassword"
            ],
            "Resource": "arn:aws:iam::*:user/${aws:username}"
        }
    ]
}
```

```

    "Sid": "AllowViewAccountInfo",
    "Effect": "Allow",
    "Action": [
        "iam:GetAccountPasswordPolicy",
        "iam:GetAccountSummary"
    ],
    "Resource": "*"
},
{
    "Sid": "AllowManageOwnPasswords",
    "Effect": "Allow",
    "Action": [
        "iam:ChangePassword",
        "iam:GetUser"
    ],
    "Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
    "Sid": "AllowManageOwnAccessKeys",
    "Effect": "Allow",
    "Action": [
        "iam>CreateAccessKey",
        "iam>DeleteAccessKey",
        "iam>ListAccessKeys",
        "iam:UpdateAccessKey"
    ],
    "Resource": "arn:aws:iam::*:user/${aws:username}"
},
{
    "Sid": "AllowManageOwnSSHPublicKeys",
    "Effect": "Allow",
    "Action": [
        "iam>DeleteSSHPublicKey",
        "iam:GetSSHPublicKey",
        "iam>ListSSHPublicKeys",
        "iam:UpdateSSHPublicKey",
        "iam:UploadSSHPublicKey"
    ],
    "Resource": "arn:aws:iam::*:user/${aws:username}"
}
]
}

```

AWS: 지정 기간 동안 MFA를 사용한 특정 액세스 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다. 예제에서는 논리 AND를 사용하여 평가되는 여러 조건을 사용합니다. SERVICE-NAME-1으로 명명된 서비스에 대해 모든 액세스를 허용하고 ACTION-NAME-A로 명명된 서비스에서 ACTION-NAME-B 및 SERVICE-NAME-2 작업에 대한 액세스를 허용합니다. 이를 작업은 사용자가 [멀티 팩터 인증\(MFA\)](#)을 통해 인증된 경우에만 허용됩니다. 액세스는 2017년 7월 1일과 2017년 12월 31일(UTC) 사이에 발생하는 작업으로 제한됩니다. 이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

IAM 정책의 Condition 블록 내에서 복수 조건을 사용하는 방법에 관한 자세한 내용은 [다수의 조건 값 \(p. 511\)](#) 단원을 참조하십시오.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "service-prefix-1:*",
            "service-prefix-2:action-name-a",

```

```
        "service-prefix-2:action-name-b"
    ],
    "Resource": "*",
    "Condition": {
        "Bool": {"aws:MultiFactorAuthPresent": true},
        "DateGreaterThanOrEqualTo": {"aws:CurrentTime": "2017-07-01T00:00:00Z"},
        "DateLessThanOrEqualTo": {"aws:CurrentTime": "2017-12-31T23:59:59Z"}
    }
}
```

AWS: 소스 IP를 바탕으로 AWS에 대한 액세스 거부

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 지정된 IP 범위를 벗어나는 IP에서 요청이 오는 경우 계정의 모든 AWS 작업에 대한 액세스를 거부합니다. 이 정책은 회사의 IP 주소가 지정된 범위 내에 있는 경우에 유용합니다. 이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

`aws:SourceIp` 조건 키는 여러분을 대신하여 호출하는 AWS CloudFormation과 같은 AWS 서비스에 대한 액세스를 거부합니다. `aws:SourceIp` 조건 키 사용에 관한 자세한 내용은 [AWS 전역 조건 컨텍스트 키 \(p. 551\)](#) 단원을 참조하십시오.

Important

이 정책은 어떤 작업도 허용하지 않습니다. 이 정책을 특정 작업을 허용하는 다른 정책과 함께 사용합니다.

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

AWS CodeCommit: 프로그램 방식으로 콘솔에서 CodeCommit 리포지토리에 대한 Read 액세스 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 `MyDemoRepo` CodeCommit 리포지토리에 대한 Read 액세스를 허용합니다. 이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AccessSpecificRepo",
            "Effect": "Allow",
            "Action": ["codecommit:GitPull"],
            "Resource": "arn:aws:codecommit:*::*:MyDemoRepo"
        }
    ]
}
```

```
        "Sid": "ViewRepositoriesConsole",
        "Effect": "Allow",
        "Action": [
            "codecommit:Get*",
            "codecommit:BatchGetRepositories",
            "codecommit>List*"
        ],
        "Resource": "*"
    }
]
```

AWS Data Pipeline: 사용자가 생성하지 않은 DataPipeline 파이프라인에 대한 액세스 거부

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.: 사용자가 생성하지 않은 파이프라인에 대한 액세스 거부 PipelineCreator 필드의 값이 IAM 사용자 이름과 일치하는 경우 지정된 작업이 거부되지 않습니다. 이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다.

Important

이 정책은 어떤 작업도 허용하지 않습니다. 이 정책을 특정 작업을 허용하는 다른 정책과 함께 사용합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ExplicitDenyIfNotTheOwner",
            "Effect": "Deny",
            "Action": [
                "datapipeline:ActivatePipeline",
                "datapipeline:AddTags",
                "datapipeline:DeactivatePipeline",
                "datapipeline>DeletePipeline",
                "datapipeline:DescribeObjects",
                "datapipeline:EvaluateExpression",
                "datapipeline:GetPipelineDefinition",
                "datapipeline:PollForTask",
                "datapipeline:PutPipelineDefinition",
                "datapipeline:QueryObjects",
                "datapipeline:RemoveTags",
                "datapipeline:ReportTaskProgress",
                "datapipeline:ReportTaskRunnerHeartbeat",
                "datapipeline:SetStatus",
                "datapipeline:SetTaskStatus",
                "datapipeline:ValidatePipelineDefinition"
            ],
            "Resource": ["*"],
            "Condition": {
                "StringNotEquals": {"datapipeline:PipelineCreator": "${aws:userid}"}
            }
        }
    ]
}
```

Amazon DynamoDB: 특정 테이블에 대한 액세스 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 MyTable DynamoDB 테이블에 대한 모든 액세스를 허용합니다. 이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수

있는 권한을 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

Important

이 정책은 DynamoDB 테이블에서 수행 가능한 모든 작업을 허용합니다. 이를 작업을 검토하려면 Amazon DynamoDB 개발자 안내서의 [DynamoDB API 권한: 작업, 리소스 및 조건 참조](#) 단원을 참조하십시오. 개별 작업 각각을 등록하여 동일한 권한을 제공할 수 있습니다. 그러나 "dynamodb:List*"와 같이 Action에서 와일드카드(*)를 사용하는 경우, DynamoDB에서 새 목록 작업을 추가한다면 정책을 업데이트할 필요가 없습니다.

이 정책은 지정된 이름을 지닌 DynamoDB 테이블에 대해서만 작업을 허용합니다. DynamoDB에 있는 모든 것에 대한 Read 액세스 권한을 사용자에게 허용하려면 [AmazonDynamoDBReadOnlyAccess](#) AWS 관리형 정책을 연결할 수도 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "ListAndDescribe",  
            "Effect": "Allow",  
            "Action": [  
                "dynamodb:List*",  
                "dynamodb:DescribeReservedCapacity*",  
                "dynamodb:DescribeLimits",  
                "dynamodb:DescribeTimeToLive"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Sid": "SpecificTable",  
            "Effect": "Allow",  
            "Action": [  
                "dynamodb:BatchGet*",  
                "dynamodb:DescribeStream",  
                "dynamodb:DescribeTable",  
                "dynamodb:Get*",  
                "dynamodb:Query",  
                "dynamodb:Scan",  
                "dynamodb:BatchWrite*",  
                "dynamodb CreateTable",  
                "dynamodb>Delete*",  
                "dynamodb:Update*",  
                "dynamodb:PutItem"  
            ],  
            "Resource": "arn:aws:dynamodb:*:*:table/MyTable"  
        }  
    ]  
}
```

Amazon DynamoDB: 특정 열에 대한 액세스 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다. 특정 DynamoDB 열에 대한 액세스를 허용합니다. 이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

dynamodb:Select 요건을 설정하면 API 작업이 인덱스 프로젝션 등의 방법으로 허용되지 않는 속성을 반환할 수 없게 됩니다. DynamoDB 조건 키에 대한 자세한 정보는 Amazon DynamoDB 개발자 안내서의 [조건 지정: 조건 키 사용](#) 단원을 참조하십시오. IAM 정책의 Condition 블록 내에서 복수 조건 또는 복수 조건 키를 사용하는 방법에 관한 자세한 내용은 [다수의 조건 값 \(p. 511\)](#) 단원을 참조하십시오.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "dynamodb:GetItem",
            "dynamodb:BatchGetItem",
            "dynamodb:Query",
            "dynamodb:PutItem",
            "dynamodb:UpdateItem",
            "dynamodb:DeleteItem",
            "dynamodb:BatchWriteItem"
        ],
        "Resource": [ "arn:aws:dynamodb:*:*:table/table-name" ],
        "Condition": {
            "ForAllValues:StringEquals": {
                "dynamodb:Attributes": [
                    "column-name-1",
                    "column-name-2",
                    "column-name-3"
                ]
            },
            "StringEqualsIfExists": { "dynamodb:Select": "SPECIFIC_ATTRIBUTES" }
        }
    }
]
```

Amazon DynamoDB: Amazon Cognito ID를 기준으로 DynamoDB에 대한 행 수준 액세스 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 Amazon Cognito ID에 따라 MyTable DynamoDB 테이블에 대한 행 수준 액세스를 허용합니다. 이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

이 정책을 사용하려면 Cognito 사용자 ID가 파티션 키가 되도록 DynamoDB 테이블을 구성해야 합니다. 자세한 정보는 Amazon DynamoDB 개발자 안내서의 [테이블 생성](#) 단원을 참조하십시오.

DynamoDB 조건 키에 대한 자세한 정보는 Amazon DynamoDB 개발자 안내서의 [조건 지정: 조건 키 사용](#) 단원을 참조하십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:DeleteItem",
                "dynamodb:GetItem",
                "dynamodb:PutItem",
                "dynamodb:Query",
                "dynamodb:UpdateItem"
            ],
            "Resource": [ "arn:aws:dynamodb:*:*:table/MyTable" ],
            "Condition": {
                "ForAllValues:StringEquals": {
                    "dynamodb:LeadingKeys": ["${cognito-identity.amazonaws.com:sub}"]
                }
            }
        }
    ]
}
```

}

Amazon EC2: EC2 인스턴스가 볼륨을 연결 또는 분리하도록 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 서비스 역할에 연결할 수 있습니다. 이 정책은 지정된 EC2 인스턴스가 볼륨을 연결 또는 분리하도록 허용합니다. 인스턴스는 Condition 요소에 ARN과 함께 지정됩니다. 이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

Amazon EC2 인스턴스는 인스턴스 프로파일에 연결되어 있는 [EC2 인스턴스의 AWS 서비스 역할 \(p. 154\)](#)에 의해 부여된 권한으로 AWS 명령을 실행할 수 있습니다. 이 정책을 역할에 연결하거나 이 명령문을 기준 정책에 추가할 수 있습니다. [INSTANCE-ID](#)에 의해 식별된 인스턴스만 해당 계정(자신의 계정 포함)의 인스턴스에 볼륨을 연결하거나 분리할 수 있습니다. 더 큰 정책에 존재할 수 있는 다른 명령문 요소는 이 '하나의 명령문' 제한에 의해 영향을 받지 않습니다. IAM 정책을 만들어 Amazon EC2 리소스에 대한 액세스를 제어하는 방법은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Amazon EC2 리소스에 대한 액세스 제어](#) 단원을 참조하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:AttachVolume",  
                "ec2:DetachVolume"  
            ],  
            "Resource": [  
                "arn:aws:ec2:*::volume/*",  
                "arn:aws:ec2:*::instance/*"  
            ],  
            "Condition": {  
                "ArnEquals": {"ec2:SourceInstanceARN": "arn:aws:ec2:*::instance/instance-id"}  
            }  
        }  
    ]  
}
```

Amazon EC2: 태그를 기준으로 Amazon EBS 볼륨을 EC2 인스턴스에 연결 또는 분리

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 EBS 볼륨 소유자가 태그 `volumeUser`를 사용하여 정의한 자신의 EBS 볼륨을 개발 인스턴스(`Department=Dev`)로 태그가 지정된 EC2 인스턴스에 연결하거나 분리할 수 있도록 허용합니다. 이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:AttachVolume",  
                "ec2:DetachVolume"  
            ],  
            "Resource": "arn:aws:ec2:*::instance/*",  
            "Condition": {  
                "StringLike": {"ec2:User": "volumeUser"}  
            }  
        }  
    ]  
}
```

```
        "Condition": {
            "StringEquals": { "ec2:ResourceTag/Department": "Development" }
        }
    },
{
    "Effect": "Allow",
    "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
    ],
    "Resource": "arn:aws:ec2:*::volume/*",
    "Condition": {
        "StringEquals": { "ec2:ResourceTag/VolumeUser": "${aws:username}" }
    }
}
]
```

Amazon EC2: 특정 서브넷에 있는 EC2 인스턴스를 프로그래밍 방식으로 콘솔에서 시작할 수 있도록 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 모든 EC2 객체에 대한 정보의 열거와 특정 서브넷에서의 EC2 인스턴스 시작을 허용합니다.이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다.이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:Describe*",
                "ec2:GetConsole*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*::subnet/subnet-subnet-id",
                "arn:aws:ec2:*::network-interface/*",
                "arn:aws:ec2:*::instance/*",
                "arn:aws:ec2:*::volume/*",
                "arn:aws:ec2::*:image/ami-*",
                "arn:aws:ec2::*:key-pair/*",
                "arn:aws:ec2::*:security-group/*"
            ]
        }
    ]
}
```

Amazon EC2: 특정 VPC와 연결된 EC2 보안 그룹을 콘솔에서 프로그래밍 방식으로 관리할 수 있도록 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 특정 가상 프라이빗 클라우드 (VPC)와 관련된 Amazon EC2 보안 그룹을 관리할 수 있도록 합니다.이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다.이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:AuthorizeSecurityGroupEgress",  
                "ec2:AuthorizeSecurityGroupIngress",  
                "ec2>DeleteSecurityGroup",  
                "ec2:RevokeSecurityGroupEgress",  
                "ec2:RevokeSecurityGroupIngress"  
            ],  
            "Resource": "arn:aws:ec2:*:*:security-group/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:Vpc": "arn:aws:ec2:*:*:vpc/vpc-vpc-id"  
                }  
            }  
        },  
        {  
            "Action": [  
                "ec2:DescribeSecurityGroups",  
                "ec2:DescribeSecurityGroupReferences",  
                "ec2:DescribeStaleSecurityGroups",  
                "ec2:DescribeVpcs"  
            ],  
            "Effect": "Allow",  
            "Resource": "*"  
        }  
    ]  
}
```

Amazon EC2: 프로그래밍 방식으로 콘솔에서 사용자가 태그를 지정한 EC2 인스턴스를 시작 또는 중지할 수 있도록 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 IAM 사용자가 EC2 인스턴스를 시작 또는 중지할 수 있도록 허용하지만, 인스턴스 태그 Owner가 사용자의 사용자 이름의 값과 같은 경우로 제한합니다. 이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:StartInstances",  
                "ec2:StopInstances"  
            ],  
            "Resource": "arn:aws:ec2:*:*:instance/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ResourceTag/Owner": "${aws:username}"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:DescribeInstances",  
            "Resource": "*"  
        }  
    ]  
}
```

}

EC2: 태그를 기반으로 인스턴스 시작 또는 중지

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 태그 키=값 페어 `Project = DataAnalytics`를 통한 인스턴스 시작 또는 중지를 허용하지만, 태그 키=값 페어 `Department = Data`가 있는 보안 주체만 가능합니다.이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다.이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

정책의 조건은 조건의 두 부분이 모두 true인 경우 true를 반환합니다. 인스턴스에 `Project=DataAnalytics` 태그가 있어야 합니다. 또한, 요청을 보내는 IAM 보안 주체 엔터티(사용자나 역할)에 `Department=Data` 태그가 있어야 합니다.

Note

가장 좋은 방법은 `aws:PrincipalTag` 조건 키가 있는 정책을 IAM 그룹에 연결하는 것입니다. 이 경우 일부 사용자는 지정된 태그가 있고 일부 사용자는 그렇지 않을 수 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "StartStopIfTags",  
            "Effect": "Allow",  
            "Action": [  
                "ec2:StartInstances",  
                "ec2:StopInstances",  
                "ec2:DescribeTags"  
            ],  
            "Resource": "arn:aws:ec2:<region>:<account-id>:instance/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ResourceTag/<Project>": "DataAnalytics",  
                    "aws:PrincipalTag/<Department>": "Data"  
                }  
            }  
        }  
    ]  
}
```

EC2: 일치하는 보안 주체 및 리소스 태그를 기반으로 인스턴스 시작 또는 중지

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.를 사용하면 인스턴스의 리소스 태그와 보안 주체 태그가 `CostCenter` 태그 키와 동일한 값을 가질 때 Amazon EC2 인스턴스를 시작하거나 중지할 수 있습니다.이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다.이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

Note

가장 좋은 방법은 `aws:PrincipalTag` 조건 키가 있는 정책을 IAM 그룹에 연결하는 것입니다. 이 경우 일부 사용자는 지정된 태그가 있고 일부 사용자는 그렇지 않을 수 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:startInstances",  
                "ec2:stopInstances"  
            ]  
        }  
    ]  
}
```

```
        ],
        "Resource": "*",
        "Condition": {"StringEquals":
            {"ec2:ResourceTag/CostCenter": "${aws:PrincipalTag/CostCenter}"}}}
    }
}
```

Amazon EC2: 특정 리전 내에서의 모든 EC2 액세스를 프로그래밍 방식으로 콘솔에서 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 특정 리전 내에서 모든 EC2 액세스를 허용합니다.이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다.이 정책을 사용 하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "ec2:*",
            "Resource": "*",
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "ec2:Region": "region"
                }
            }
        }
    ]
}
```

Amazon EC2: 프로그래밍 방식으로 콘솔에서 EC2 인스턴스를 시작 또는 중지하고 보안 그룹을 수정할 수 있도록 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 프로그래밍 방식으로 콘솔에서 특정 EC2 인스턴스를 시작 또는 중지하고 특정 보안 그룹을 수정할 수 있도록 허용합니다.이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다.이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSecurityGroupReferences",
                "ec2:DescribeStaleSecurityGroups"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RevokeSecurityGroupIngress",
                "ec2:StartInstances",
                "ec2:StopInstances"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

```
        ],
        "Resource": [
            "arn:aws:ec2:*.*:instance/i-instance-id",
            "arn:aws:ec2:*.*:security-group/sg-security-group-id"
        ],
        "Effect": "Allow"
    ]
}
```

Amazon EC2: EC2 인스턴스 종료를 IP 주소 범위로 제한

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 작업을 허용하지만 요청이 지정된 IP 범위를 벗어나는 곳에서 오는 경우 액세스를 명시적으로 거부함으로써 EC2 인스턴스를 제한합니다. 이 정책은 회사의 IP 주소가 지정된 범위 내에 있는 경우에 유용합니다.이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다.이 정책을 사용하려면 정책 예제의 빨간색 기울임 텍스트를 본인의 정보로 대체하십시오.

이 정책을 `ec2:TerminateInstances` 작업을 허용하는 다른 정책(예: [AmazonEC2FullAccess](#) AWS 관리형 정책)과 조합하여 사용하는 경우 액세스가 거부됩니다. 이는 명시적 거부문이 허용문보다 우선 적용되기 때문입니다. 자세한 내용은 [the section called “계정 내에서 요청 허용 여부 결정” \(p. 534\)](#)를 참조하십시오.

Important

`aws:SourceIp` 조건 키는 여러분을 대신하여 호출하는 AWS CloudFormation과 같은 AWS 서비스에 대한 액세스를 거부합니다. `aws:SourceIp` 조건 키 사용에 관한 자세한 내용은 [AWS 전역 조건 컨텍스트 키 \(p. 551\)](#) 단원을 참조하십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ec2:TerminateInstances"],
            "Resource": ["*"]
        },
        {
            "Effect": "Deny",
            "Action": ["ec2:TerminateInstances"],
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "203.0.113.0/24"
                    ]
                }
            },
            "Resource": ["*"]
        }
    ]
}
```

IAM: 정책 시뮬레이터 API에 액세스

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 현재 AWS 계정에서 사용자, 그룹 또는 역할에 연결되어 있는 정책에 대해 정책 시뮬레이터 API의 사용을 허용합니다. 또한 이 정책은 API에 문자열로 전달되는 덜 민감한 정책을 시뮬레이션할 수 있도록 액세스를 허용합니다.이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다.

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

```
"Statement": [
    {
        "Action": [
            "iam:GetContextKeysForCustomPolicy",
            "iam:GetContextKeysForPrincipalPolicy",
            "iam:SimulateCustomPolicy",
            "iam:SimulatePrincipalPolicy"
        ],
        "Effect": "Allow",
        "Resource": "*"
    }
]
```

Note

사용자가 정책 시뮬레이터 콘솔에 액세스하여 현재 AWS 계정의 사용자, 그룹 또는 역할에 연결된 정책을 시뮬레이션하도록 허용하는 방법은 [IAM: 정책 시뮬레이터 콘솔 액세스 \(p. 363\)](#) 단원을 참조하십시오.

IAM: 정책 시뮬레이터 콘솔 액세스

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 현재 AWS 계정에서 사용자, 그룹 또는 역할에 연결되어 있는 정책에 대해 정책 시뮬레이터 콘솔의 사용을 허용합니다. 이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다.

다음 위치에서 IAM 정책 시뮬레이터 콘솔에 액세스할 수 있습니다. <https://policysim.aws.amazon.com/>

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "iam:GetGroup",
                "iam:GetGroupPolicy",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:GetUser",
                "iam:GetUserPolicy",
                "iam>ListAttachedGroupPolicies",
                "iam>ListAttachedRolePolicies",
                "iam>ListAttachedUserPolicies",
                "iam>ListGroups",
                "iam>ListGroupPolicies",
                "iam>ListGroupsForUser",
                "iam>ListRolePolicies",
                "iam>ListRoles",
                "iam>ListUserPolicies",
                "iam>ListUsers"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

IAM: 특정 태그가 있는 사용자에게 특정 태그 추가

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 태그 값 Marketing, Development 또는 QualityAssurance가 지정된 태그 키 Department를 IAM 사용자에게 추가하도록

허용합니다. 사용자가 이미 태그 키-값 페어 JobFunction = manager를 포함해야 합니다. 이 정책을 사용하여 관리자가 세 부서 중 하나에 속하도록 요구할 수 있습니다. 이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

ListTagsForAllUsers 문을 사용하면 계정의 모든 사용자에 대한 태그를 볼 수 있습니다.

TagManagerWithSpecificDepartment 문의 첫 번째 조건에는 StringEquals 조건 연산자가 사용됩니다. 이 조건은 조건의 두 부분이 모두 true인 경우 true를 반환합니다. 태그 지정될 사용자에게 이미 JobFunction=Manager 태그가 있어야 합니다. 나열된 태그 값 중 하나가 지정된 Department 태그 키가 요청에 포함되어야 합니다.

두 번째 조건에는 ForAllValues:StringEquals 조건 연산자가 사용됩니다. 이 조건은 요청의 모든 태그 키가 정책의 키와 일치하는 경우 true를 반환합니다. 즉 Department가 요청의 유일한 태그 키어야 합니다. ForAllValues 사용에 관한 자세한 정보는 [여러 키 값을 테스트하는 조건 생성\(설정 작업\) \(p. 520\)](#) 단원을 참조하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "ListTagsForAllUsers",  
            "Effect": "Allow",  
            "Action": [  
                "iam>ListUserTags",  
                "iam>ListUsers"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Sid": "TagManagerWithSpecificDepartment",  
            "Effect": "Allow",  
            "Action": "iam:TagUser",  
            "Resource": "*",  
            "Condition": {"StringEquals": {  
                "iam:ResourceTag/JobFunction": "Manager",  
                "aws:RequestTag/Department": [  
                    "Marketing",  
                    "Development",  
                    "QualityAssurance"  
                ]  
            }},  
            "ForAllValues:StringEquals": {"aws:TagKeys": "Department"}  
        }  
    ]  
}
```

IAM: 특정 값이 있는 특정 태그 추가

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.를 사용하면 모든 IAM 사용자 또는 역할에 태그 키 CostCenter와 태그 값 A-123 또는 태그 값 B-456만 추가할 수 있습니다. 이 정책을 사용하여 특정 태그 키 및 태그 값 세트로 태그 지정을 제한할 수 있습니다. 이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

ConsoleDisplay 문을 사용하면 계정의 모든 사용자 및 역할에 대한 태그를 볼 수 있습니다.

AddTag 문의 첫 번째 조건에는 stringEquals 조건 연산자가 사용됩니다. 이 조건은 나열된 태그 값 중 하나가 지정된 CostCenter 태그 키가 요청에 포함된 경우 true를 반환합니다.

두 번째 조건에는 `ForAllValues:StringEquals` 조건 연산자가 사용됩니다. 이 조건은 요청의 모든 태그 키가 정책의 키와 일치하는 경우 `true`를 반환합니다. 즉 `CostCenter`가 요청의 유일한 태그 키여야 합니다. `ForAllValues` 사용에 관한 자세한 정보는 [여러 키 값을 테스트하는 조건 생성\(설정 작업\) \(p. 520\)](#) 단원을 참조하십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ConsoleDisplay",
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:GetUser",
                "iam>ListRoles",
                "iam>ListRoleTags",
                "iam>ListUsers",
                "iam>ListUserTags"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AddTag",
            "Effect": "Allow",
            "Action": [
                "iam:TagUser",
                "iam:TagRole"
            ],
            "Resource": "*",
            "Condition": [
                {"StringEquals": {"aws:RequestTag/CostCenter": [
                    "A-123",
                    "B-456"
                ]}},
                {"ForAllValues:StringEquals": {"aws:TagKeys": "CostCenter"}}
            ]
        }
    ]
}
```

IAM: 특정 태그가 있는 새 사용자만 생성

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 IAM 사용자 생성을 허용하지만 `Department` 및 `JobFunction` 태그 키 중 하나 또는 두 개 모두를 사용해야 합니다. `Department` 태그 키에는 `Development` 또는 `QualityAssurance` 태그 값이 지정되어야 합니다. `JobFunction` 태그 키에는 `Employee` 태그 값이 지정되어야 합니다. 새 사용자가 특정 업무 기능 및 부서를 갖도록 하려면 이 정책을 사용하십시오. 이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

이 설명문의 첫 번째 조건에는 `StringEqualsIfExists` 조건 연산자가 사용됩니다. 요청에 키가 `Department` 또는 `JobFunction`인 태그가 있는 경우 태그에 지정된 값이 있어야 합니다. 두 키가 모두 없으면 이 조건은 `true`로 평가됩니다. 조건이 `false`로 평가되는 유일한 경우는 지정된 조건 키 중 하나가 요청에 있지만 허용된 값이 아닌 다른 값이 지정된 경우뿐입니다. `IfExists` 사용에 관한 자세한 정보는 [IfExists 조건 연산자 \(p. 518\)](#) 단원을 참조하십시오.

두 번째 조건에는 `ForAllValues:StringEquals` 조건 연산자가 사용됩니다. 이 조건은 요청에 지정된 각각의 태그 키와 정책의 하나 이상의 값이 일치하는 경우 `true`를 반환합니다. 즉 요청의 모든 태그가 이 목록에 있어야 합니다. 하지만 요청은 목록에 있는 태그 중 하나만 포함할 수 있습니다. 예를 들면 `Department=QualityAssurance` 태그만 지정된 IAM 사용자를 생성할 수 있습니다. 하지만 `JobFunction=employee` 태그와 `Project=core` 태그가 지정된 IAM 사용자는 생성할 수 없습니다.

ForAllValues 사용에 관한 자세한 정보는 [여러 키 값을 테스트하는 조건 생성\(설정 작업\) \(p. 520\)](#) 단원을 참조하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "TagUsersWithOnlyTheseTags",  
            "Effect": "Allow",  
            "Action": [  
                "iam:CreateUser",  
                "iam:TagUser"  
            ],  
            "Resource": "*",  
            "Condition": {  
                "StringEqualsIfExists": {  
                    "aws:RequestTag/Department                        "Development",  
                        "QualityAssurance"  
                    ],  
                    "aws:RequestTag/JobFunctionEmployee"  
                },  
                "ForAllValues:StringEquals": {  
                    "aws:TagKeys": [  
                        "Department",  
                        "JobFunction"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

IAM: 특정 태그 관리

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.를 사용하면 태그 키 **Department**가 있는 IAM 태그를 추가 및 제거할 수 있습니다. 이 정책은 **Department** 태그의 가치를 제한하지 않습니다.이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다.이 정책을 사용 하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": [  
            "iam:TagUser",  
            "iam:TagRole",  
            "iam:UntagUser",  
            "iam:UntagRole"  
        ],  
        "Resource": "*",  
        "Condition": {"ForAllValues:StringEquals": {"aws:TagKeys": "Department"}}  
    }  
}
```

IAM: 사용자 경로를 바탕으로 정책 시뮬레이터 API 액세스

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 경로 **Department/Development**을 지닌 사용자에 대해서만 정책 시뮬레이터 API의 사용을 허용합니다.이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다.이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "iam:GetContextKeysForPrincipalPolicy",  
                "iam:SimulatePrincipalPolicy"  
            ],  
            "Effect": "Allow",  
            "Resource": "arn:aws:iam::*:user/Department/Development/*"  
        }  
    ]  
}
```

Note

경로 **Department/Development**을 지닌 사용자에 대해 정책 시뮬레이터 콘솔의 사용을 허용하는 정책을 생성하는 방법은 [IAM: 사용자 경로를 바탕으로 정책 시뮬레이터 콘솔 액세스 \(p. 367\)](#) 단원을 참조하십시오.

IAM: 사용자 경로를 바탕으로 정책 시뮬레이터 콘솔 액세스

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 경로 **Department/Development**을 지닌 사용자에 대해서만 정책 시뮬레이터 콘솔의 사용을 허용합니다. 이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

다음 위치에서 IAM 정책 시뮬레이터에 액세스할 수 있습니다. <https://policysim.aws.amazon.com/>

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "iam:GetPolicy",  
                "iam:GetUserPolicy"  
            ],  
            "Effect": "Allow",  
            "Resource": "*"  
        },  
        {  
            "Action": [  
                "iam:GetUser",  
                "iam>ListAttachedUserPolicies",  
                "iam>ListGroupsForUser",  
                "iam>ListUserPolicies",  
                "iam>ListUsers"  
            ],  
            "Effect": "Allow",  
            "Resource": "arn:aws:iam::*:user/Department/Development/*"  
        }  
    ]  
}
```

IAM: IAM 사용자가 MFA 디바이스를 스스로 관리하도록 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.: IAM 사용자가 MFA 디바이스를 스스로 관리하도록 허용합니다. 이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다.

Note

AWS에 로그인되어 있는 사용자에게 이들 권한을 추가하는 경우 사용자가 로그아웃한 다음 변경을 확인해야 할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowListActions",
            "Effect": "Allow",
            "Action": [
                "iam>ListUsers",
                "iam>ListVirtualMFADevices"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowIndividualUserToListOnlyTheirOwnMFA",
            "Effect": "Allow",
            "Action": [
                "iam>ListMFADevices"
            ],
            "Resource": [
                "arn:aws:iam::*:mfa/*",
                "arn:aws:iam::*:user/${aws:username}"
            ]
        },
        {
            "Sid": "AllowIndividualUserToManageTheirOwnMFA",
            "Effect": "Allow",
            "Action": [
                "iam>CreateVirtualMFADevice",
                "iam>DeleteVirtualMFADevice",
                "iam>EnableMFADevice",
                "iam>ResyncMFADevice"
            ],
            "Resource": [
                "arn:aws:iam::*:mfa/${aws:username}",
                "arn:aws:iam::*:user/${aws:username}"
            ]
        },
        {
            "Sid": "AllowIndividualUserToDeactivateOnlyTheirOwnMFAOnlyWhenUsingMFA",
            "Effect": "Allow",
            "Action": [
                "iam>DeactivateMFADevice"
            ],
            "Resource": [
                "arn:aws:iam::*:mfa/${aws:username}",
                "arn:aws:iam::*:user/${aws:username}"
            ],
            "Condition": {
                "Bool": {
                    "aws:MultiFactorAuthPresent": "true"
                }
            }
        },
        {
            "Sid": "BlockMostAccessUnlessSignedInWithMFA",
            "Effect": "Deny",
            "NotAction": [
                "iam>CreateVirtualMFADevice",
                "iam>EnableMFADevice",
                "iam>ListMFADevices",
                "iam>SetMFADevice"
            ]
        }
    ]
}
```

```
        "iam>ListUsers",
        "iam>ListVirtualMFADevices",
        "iam>ResyncMFADevice"
    ],
    "Resource": "*",
    "Condition": {
        "BoolIfExists": {
            "aws:MultiFactorAuthPresent": "false"
        }
    }
}
]
```

IAM: 프로그램 방식으로 콘솔에서 IAM 사용자가 자신의 자격 증명을 교체하도록 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 IAM 사용자가 자신의 액세스 키, 서명 인증서, 서비스별 자격 증명 및 암호를 교체하도록 허용합니다. 이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam>ListUsers",
                "iam>GetAccountPasswordPolicy"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:*AccessKey*",
                "iam>ChangePassword",
                "iam GetUser",
                "iam:*ServiceSpecificCredential*",
                "iam:*SigningCertificate*"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        }
    ]
}
```

사용자가 콘솔에서 자신의 암호를 변경하는 방법에 대해서는 [the section called “IAM 사용자가 자신의 암호를 변경하는 방법” \(p. 87\)](#) 단원을 참조하십시오.

IAM: IAM 사용자, 그룹 또는 역할에 적용 가능한 관리형 정책을 제한

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 IAM 사용자, 그룹 또는 역할에 적용 가능한 고객 관리형 및 AWS 관리형 정책을 제한합니다.이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다.이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

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

```
"Statement": {
    "Effect": "Allow",
    "Action": [
        "iam:AttachUserPolicy",
        "iam:DetachUserPolicy"
    ],
    "Resource": "*",
    "Condition": {
        "ArnEquals": {
            "iam:PolicyARN": [
                "arn:aws:iam::*:policy/policy-name-1",
                "arn:aws:iam::*:policy/policy-name-2"
            ]
        }
    }
}
```

Amazon RDS: 특정 리전에 있는 RDS 데이터베이스에 대한 완전한 액세스 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 RDS 데이터베이스에 대한 완전한 액세스 허용합니다. 이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다.이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "rds:*",
            "Resource": ["arn:aws:rds:region::*"]
        },
        {
            "Effect": "Allow",
            "Action": ["rds:Describe*"],
            "Resource": ["*"]
        }
    ]
}
```

Amazon RDS: 프로그램 방식으로 콘솔에서 RDS 데이터베이스를 복원하도록 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 RDS 데이터베이스 복원을 허용합니다. 이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:Describe*",
                "rds>CreateDBParameterGroup",
                "rds>CreateDBSnapshot",
                "rds>DeleteDBSnapshot",
                "rds:Describe*",
                "rds:DownloadDBLogFilePortion",
                "rds:RebootDBInstance",
                "rds:RestoreDBFromSnapshot"
            ]
        }
    ]
}
```

```
        "rds>List*",
        "rds:ModifyDBInstance",
        "rds:ModifyDBParameterGroup",
        "rds:ModifyOptionGroup",
        "rds:RebootDBInstance",
        "rds:RestoreDBInstanceFromDBSnapshot",
        "rds:RestoreDBInstanceToPointInTime"
    ],
    "Resource": "*"
}
]
}
```

Amazon RDS: 태그 소유자가 자신이 태그를 지정한 RDS 리소스에 대한 모든 액세스 권한을 가지도록 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 태그 소유자가 자신이 태그를 지정한 RDS 리소스에 대한 모든 액세스 권한을 가지도록 허용합니다.이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "rds:Describe*",
                "rds>List*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "rds>DeleteDBInstance",
                "rds:RebootDBInstance",
                "rds:ModifyDBInstance"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:db-tag/Owner": "${aws:username}"}
            }
        },
        {
            "Action": [
                "rds:ModifyOptionGroup",
                "rds>DeleteOptionGroup"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:og-tag/Owner": "${aws:username}"}
            }
        },
        {
            "Action": [
                "rds:ModifyDBParameterGroup",
                "rds:ResetDBParameterGroup"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:pg-tag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

```

        },
        {
            "Action": [
                "rds:AuthorizeDBSecurityGroupIngress",
                "rds:RevokeDBSecurityGroupIngress",
                "rds:DeleteDBSecurityGroup"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:secgrp-tag/Owner": "${aws:username}"}
            }
        },
        {
            "Action": [
                "rds:DeleteDBSnapshot",
                "rds:RestoreDBInstanceFromDBSnapshot"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:snapshot-tag/Owner": "${aws:username}"}
            }
        },
        {
            "Action": [
                "rds:ModifyDBSubnetGroup",
                "rds:DeleteDBSubnetGroup"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:subgrp-tag/Owner": "${aws:username}"}
            }
        },
        {
            "Action": [
                "rds:ModifyEventSubscription",
                "rds:AddSourceIdentifierToSubscription",
                "rds:RemoveSourceIdentifierFromSubscription",
                "rds:DeleteEventSubscription"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"rds:es-tag/Owner": "${aws:username}"}
            }
        }
    ]
}

```

Amazon S3: Amazon Cognito 사용자가 자신의 버킷에 있는 객체에 액세스할 수 있도록 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 Amazon Cognito 사용자가 특정 S3 버킷에 있는 객체에 액세스하도록 허용합니다. 이 정책은 \${cognito-identity.amazonaws.com:sub} 변수로 표현되는 cognito, 애플리케이션 이름 및 연동 사용자의 ID를 포함하는 이름을 통해 객체에 대한 액세스만을 허용합니다.이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여 합니다.이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{

```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": ["s3>ListBucket"],
        "Resource": ["arn:aws:s3:::bucket-name"],
        "Condition": {
            "StringLike": {
                "s3:prefix": ["cognito/application-name/"]
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:GetObject",
            "s3:PutObject",
            "s3>DeleteObject"
        ],
        "Resource": [
            "arn:aws:s3:::bucket-name/cognito/application-name/${cognito-identity.amazonaws.com:sub}",
            "arn:aws:s3:::bucket-name/cognito/application-name/${cognito-identity.amazonaws.com:sub}/*"
        ]
    }
]
```

Amazon Cognito는 웹 및 모바일 앱에 대한 인증, 권한 부여 및 사용자 관리를 제공합니다. 사용자는 사용자 이름과 암호를 사용하여 직접 로그인하거나 Facebook, Amazon, 또는 Google 같은 타사를 통해 로그인할 수 있습니다.

Amazon Cognito의 두 가지 주요 구성 요소는 사용자 풀과 자격 증명 풀입니다. 사용자 풀은 앱 사용자의 가입 및 로그인 옵션을 제공하는 사용자 디렉터리입니다. 자격 증명 풀을 통해 사용자에게 기타 AWS 서비스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자격 증명 풀과 사용자 풀을 별도로 또는 함께 사용할 수 있습니다.

Amazon Cognito에 대한 자세한 내용은 다음을 참조하십시오.

- Android용 AWS Mobile SDK Developer Guide의 [Amazon Cognito 자격 증명](#)
- AWS Mobile SDK for iOS Developer Guide의 [Amazon Cognito 자격 증명](#)

Amazon S3: 연합된 사용자가 프로그램 방식으로 콘솔에서 자신의 S3 훈 디렉터리에 액세스하도록 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 연합된 사용자가 S3에 있는 자신의 훈 디렉터리 버킷 객체에 액세스하도록 허용합니다. 훈 디렉터리는 개별 연합된 사용자의 home 폴더를 포함하는 버킷입니다. 이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

이 정책의 `#{aws:userid}` 변수가 `role-id:specified-name`로 변환됩니다. 연합된 사용자 ID의 `role-id` 부분은 생성 중에 연합된 사용자의 역할에 할당된 고유한 식별자입니다. 자세한 내용은 [고유 ID \(p. 483\)](#)를 참조하십시오. `specified-name`은 연합된 사용자가 자신의 역할을 맡을 때 `AssumeRoleWithWebIdentity` 요청에 전달된 `RoleSessionName` 파라미터입니다.

AWS CLI 명령 `aws iam get-role --role-name specified-name`을 사용하여 역할 ID를 볼 수 있습니다. 예를 들어, 기억하기 쉬운 이름 `John`을 지정하고 CLI가 역할 ID `AROAXXT2NJT7D3SIQN7Z6`를 반환한다고 가정해 봅시다. 이 경우 연합된 사용자 ID는 `AROAXXT2NJT7D3SIQN7Z6:John`입니다. 그러면 이 정책

에서 연합된 사용자 John이 접두사 AROAXXT2NJT7D3SIQN7Z6:John로 시작하는 Amazon S3 버킷에 액세스할 수 있도록 허용합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3>ListAllMyBuckets",  
                "s3:GetBucketLocation"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "s3>ListBucket",  
            "Resource": "arn:aws:s3:::bucket-name",  
            "Condition": {  
                "StringLike": {  
                    "s3:prefix": [  
                        "",  
                        "home/",  
                        "home/${aws:userid}/*"  
                    ]  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "s3:*",  
            "Resource": [  
                "arn:aws:s3:::bucket-name/home/${aws:userid}",  
                "arn:aws:s3:::bucket-name/home/${aws:userid}/*"  
            ]  
        }  
    ]  
}
```

Amazon S3: IAM 사용자가 프로그램 방식으로 콘솔에서 자신의 S3 홈 디렉터리에 액세스하도록 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 IAM 사용자가 S3에 있는 자신의 홈 디렉터리 버킷 객체에 액세스하도록 허용합니다. 홈 디렉터리는 개별 사용자의 home 폴더를 포함하는 버킷입니다. 이 정책은 콘솔에서 이 작업을 완료하는데 필요한 권한도 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3>ListAllMyBuckets",  
                "s3:GetBucketLocation"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "s3>ListBucket",  
            "Resource": "arn:aws:s3:::bucket-name",  
            "Condition": {  
                "StringLike": {  
                    "s3:prefix": [  
                        "home/<${aws:userid}>"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

```
"Condition": {
    "StringLike": {
        "s3:prefix": [
            "",
            "home/",
            "home/${aws:username}/*"
        ]
    }
},
{
    "Effect": "Allow",
    "Action": "s3:*",
    "Resource": [
        "arn:aws:s3:::bucket-name/home/${aws:username}",
        "arn:aws:s3:::bucket-name/home/${aws:username}/*"
    ]
}
]
```

Amazon S3: 특정 S3 버킷으로 관리를 제한

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 특정 버킷에 대한 모든 S3 작업을 허용하지만 Amazon S3를 제외한 모든 AWS 서비스에 대한 액세스는 명시적으로 거부함으로써 S3 버킷의 관리를 제한합니다. 이 정책에서는 `s3>ListAllMyBuckets` 또는 `s3GetObject`와 같이 S3 버킷에서 수행 불가능한 작업에 대한 액세스도 거부합니다. 이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

이 정책을 이 정책에서 거부하는 작업을 허용하는 다른 정책(예: [AmazonS3FullAccess](#) 또는 [AmazonEC2FullAccess](#) AWS 관리형 정책)과 조합하여 사용하는 경우 액세스가 거부됩니다. 이는 명시적 거부문이 허용문보다 우선 적용되기 때문입니다. 자세한 내용은 [the section called “계정 내에서 요청 허용 여부 결정” \(p. 534\)](#) 단원을 참조하십시오.

Warning

[NotAction \(p. 507\)](#) 및 [NotResource \(p. 509\)](#)는 신중히 사용해야 하는 고급 정책 요소입니다. 이 정책에서는 Amazon S3를 제외한 모든 AWS 서비스에 대한 액세스를 거부합니다. 이 정책을 사용자에게 연결할 경우 다른 서비스에 대한 권한을 부여하는 다른 정책은 무시되거나 액세스가 거부됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ]
        },
        {
            "Effect": "Deny",
            "NotAction": "s3:*",
            "NotResource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ]
        }
    ]
}
```

}

Amazon S3: S3 버킷에 있는 객체에 대한 읽기 및 쓰기 액세스 권한을 허용합니다.

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 Read 및 Write 액세스 권한을 특정 S3 버킷에 있는 객체에 대해 허용합니다.이 정책은 AWS API 또는 AWS CLI를 사용해서만 이 작업을 완료할 수 있는 권한을 부여합니다.이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

s3:`*Object` 작업에서는 와일드카드를 작업 이름의 일부로 사용합니다. `AllObjectActions` 문은 '객체' 단어로 끝나는 `GetObject`, `DeleteObject`, `PutObject` 및 기타 Amazon S3 작업을 허용합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListObjectsInBucket",
            "Effect": "Allow",
            "Action": ["s3>ListBucket"],
            "Resource": ["arn:aws:s3:::<bucket-name>"]
        },
        {
            "Sid": "AllObjectActions",
            "Effect": "Allow",
            "Action": "s3:*Object",
            "Resource": ["arn:aws:s3:::<bucket-name>/*"]
        }
    ]
}
```

Note

Amazon S3 버킷에 있는 객체에 대한 Read 및 Write 액세스 권한을 허용하고 콘솔 액세스에 대한 추가 권한을 포함하려면 [Amazon S3: S3 버킷에 있는 객체에 대한 읽기 및 쓰기 액세스 권한을 프로그래밍 방식으로 콘솔에서 허용](#) (p. 376) 단원을 참조하십시오.

Amazon S3: S3 버킷에 있는 객체에 대한 읽기 및 쓰기 액세스 권한을 프로그래밍 방식으로 콘솔에서 허용

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다.에서는 Read 및 Write 액세스 권한을 특정 S3 버킷에 있는 객체에 대해 허용합니다. 이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

s3:*Object 작업에서는 와일드카드를 작업 이름의 일부로 사용합니다. AllObjectActions 문은 '객체' 단어로 끝나는 GetObject, DeleteObject, PutObject 및 기타 Amazon S3 작업을 허용합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "ConsoleAccess",  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetAccountPublicAccessBlock",  
                "s3:GetBucketAcl",  
                "s3:GetBucketLocation",  
                "s3:GetBucketPolicyStatus".  
            ]  
        }  
    ]  
}
```

```
        "s3:GetBucketPublicAccessBlock",
        "s3>ListAllMyBuckets"
    ],
    "Resource": "*"
},
{
    "Sid": "ListObjectsInBucket",
    "Effect": "Allow",
    "Action": "s3>ListBucket",
    "Resource": ["arn:aws:s3:::bucket-name"]
},
{
    "Sid": "AllObjectActions",
    "Effect": "Allow",
    "Action": "s3:*Object",
    "Resource": ["arn:aws:s3:::bucket-name/*"]
}
]
```

IAM 정책 관리

IAM은 모든 유형의 IAM 정책(관리형 정책 및 인라인 정책)을 생성하고 관리하는 도구를 제공합니다. 권한을 IAM 보안 주체 엔터티, 즉 IAM 사용자, 그룹 또는 역할에게 추가하려면 정책을 먼저 생성한 다음 정책을 보안 주체 엔터티에게 추가하면 됩니다. 다수의 정책을 보안 주체 개체 하나에게 연결하거나, 정책마다 다수의 권한이 포함될 수 있습니다.

자세한 내용은 다음 리소스를 참조하십시오.

- 다른 유형의 IAM 정책에 대한 자세한 내용은 [정책 및 권한 \(p. 305\)](#) 단원을 참조하십시오.
- IAM 내에서 정책 사용에 대한 일반적인 내용은 [액세스 관리 \(p. 304\)](#) 단원을 참조하십시오.
- 다수의 정책이 임의의 IAM 보안 주체 엔터티 하나에게 적용되는 경우 권한 평가 방법에 대한 자세한 내용은 [정책 평가 로직 \(p. 531\)](#) 단원을 참조하십시오.
- 정책 크기 및 이름 지정 제한에 대한 자세한 내용은 [IAM 개체 및 객체에 대한 제한 \(p. 485\)](#) 단원을 참조하십시오.

주제

- [IAM 정책 만들기 \(p. 377\)](#)
- [JSON 정책 검증 \(p. 382\)](#)
- [IAM 정책 시뮬레이터로 IAM 정책 테스트하기 \(p. 383\)](#)
- [IAM 자격 증명 권한 추가 및 제거 \(p. 391\)](#)
- [IAM 정책 버전 관리 \(p. 399\)](#)
- [IAM 정책 편집 \(p. 402\)](#)
- [IAM 정책 삭제 \(p. 406\)](#)
- [서비스에서 마지막으로 액세스한 데이터를 사용하여 권한 줄이기 \(p. 409\)](#)

IAM 정책 만들기

정책 ([p. 305](#))은 자격 증명 또는 리소스에 연결될 때 해당 권한을 정의하는 개체입니다. 정책은 JSON 문서로 AWS에 저장되며 IAM에서 자격 증명 기반 정책으로 보안 주체에 연결됩니다. 자격 증명 기반 정책을 IAM 그룹, 사용자 또는 역할과 같은 보안 주체(또는 자격 증명)에 연결할 수 있습니다. 자격 증명 기반 정책에는 AWS 관리형 정책, 고객 관리형 정책 및 [인라인 정책 \(p. 312\)](#)이 포함됩니다.

다음 방법 중 하나를 사용하여 AWS Management 콘솔에서 새 IAM 정책을 생성할 수 있습니다.

- 가져오기 — 계정으로 관리형 정책을 가져온 다음 정책을 편집하여 특정 요구 사항에 맞게 사용자 지정할 수 있습니다. 관리형 정책은 사용자가 이전에 생성한 고객 관리형 정책이거나 AWS 관리형 정책일 수 있습니다.
- Visual editor(시각적 편집기) — 시각적 편집기에서 정책을 새로 생성할 수 있습니다. 시각적 편집기를 사용할 경우 JSON 구문을 이해할 필요가 없습니다.
- JSON — JSON 탭에서 JSON 구문을 사용하여 정책을 생성할 수 있습니다. 새 JSON 정책 문서를 입력하거나 [예제 정책](#) (p. 341)을 붙여 넣을 수 있습니다.

AWS Management 콘솔에서 인라인 정책을 생성할 수 있습니다. 인라인 정책은 사용자가 생성한 정책으로 IAM 그룹, 사용자 또는 역할에 직접 삽입할 수 있습니다. 자세히 알아보려면 [IAM 자격 증명 권한 추가 및 제거](#) (p. 391) 단원을 참조하십시오. AWS 관리형 정책은 생성할 수 없습니다.

정책 크기 제한 및 기타 할당량에 대한 자세한 정보는 [IAM 개체 및 객체에 대한 제한](#) (p. 485) 단원을 참조하십시오.

주제

- [IAM 정책 만들기\(콘솔\)](#) (p. 378)
- [IAM 정책 생성\(AWS CLI\)](#) (p. 381)
- [IAM 정책 생성\(AWS API\)](#) (p. 382)

IAM 정책 만들기(콘솔)

정책을 생성하기 위해 어떤 방식을 선택하든 모두 같은 방식으로 시작합니다.

새 정책 생성을 시작하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 왼쪽의 탐색 창에서 정책을 선택합니다.

- 정책을 처음으로 선택하는 경우 Welcome to Managed Policies 페이지가 나타납니다. [Get Started]를 선택합니다.
3. [Create policy]를 선택합니다.
 4. 다음 방법 중 하나를 선택하여 정책을 생성합니다. 그런 다음 해당 절차가 제시하는 단계를 따릅니다.
 - [기존 관리형 정책 가져오기](#) (p. 378)
 - [시각적 편집기를 사용하여 정책 만들기](#) (p. 379)
 - [JSON 탭에서 정책 만들기](#) (p. 381)

기존 관리형 정책 가져오기

새 정책을 생성하는 쉬운 방법은 최소한으로 필요한 권한 중 일부가 이미 존재하는 계정으로 기존 관리형 정책을 가져오는 것입니다. 그런 다음, 새로운 요구 사항에 일치하도록 정책을 사용자 지정할 수 있습니다.

인라인 정책은 가져올 수 없습니다. 관리형 정책과 인라인 정책의 차이에 대해 자세히 알아보려면 [관리형 정책과 인라인 정책](#) (p. 312) 단원을 참조하십시오.

시각적 편집기에서 기존 관리형 정책을 가져오려면

1. [IAM 정책 만들기\(콘솔\)](#) (p. 378)의 단계를 따라 정책 생성 마법사를 시작합니다. Visual editor(시각적 편집기) 탭을 선택한 다음 페이지 오른쪽에서 Import managed policy(관리형 정책 가져오기)를 선택합니다.

2. Import managed policies(관리형 정책 가져오기) 창에서 새 정책에 포함할 정책과 가장 근접한 관리형 정책을 선택합니다. 필터 메뉴를 사용하거나 상단의 검색 상자에 입력하여 정책 목록의 결과를 제한할 수 있습니다.
3. [Import]를 선택합니다.
가져온 정책은 정책 하단의 새 권한 블록에 추가됩니다.
4. Visual editor(시각적 편집기)를 사용하거나 JSON을 선택하여 정책을 사용자 지정합니다. 그런 다음 정책 검토를 선택합니다.

Note

언제든지 Visual editor(시각적 편집기) 및 JSON 탭을 전환할 수 있습니다. 그러나 변경을 수행하거나 Visual editor(시각적 편집기) 탭에서 정책 검토를 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 정보는 [정책 재구성 \(p. 455\)](#) 단원을 참조하십시오.

5. 검토 페이지에서 생성하는 정책에 대한 이름과 설명(선택 사항)을 입력합니다. 이러한 설정은 나중에 편집할 수 없습니다. 정책 요약을 검토한 다음 정책 생성을 선택하여 작업을 저장합니다.

JSON 탭에서 기존 관리형 정책을 가져오려면

1. [IAM 정책 만들기\(콘솔\) \(p. 378\)](#)의 단계를 따라 정책 생성 마법사를 시작합니다. JSON 탭을 선택한 다음 페이지 오른쪽에서 Import managed policy(관리형 정책 가져오기)를 선택합니다.
2. Import managed policies(관리형 정책 가져오기) 창에서 새 정책에 포함할 정책과 가장 근접한 관리형 정책을 선택합니다. 필터 메뉴를 사용하거나 상단의 검색 상자에 입력하여 정책 목록의 결과를 제한할 수 있습니다.
3. [Import]를 선택합니다.

가져온 정책의 문은 JSON 정책 하단에 추가됩니다.

4. 정책을 JSON으로 사용자 지정하거나 Visual editor(시각적 편집기)를 선택합니다. 그런 다음 정책 검토를 선택합니다.

Note

언제든지 Visual editor(시각적 편집기) 및 JSON 탭을 전환할 수 있습니다. 그러나 변경을 수행하거나 Visual editor(시각적 편집기) 탭에서 정책 검토를 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 정보는 [정책 재구성 \(p. 455\)](#) 단원을 참조하십시오.

5. 정책 검토 페이지에서 생성하는 정책에 대한 이름과 설명(선택 사항)을 입력합니다. 이러한 필드는 나중에 편집할 수 없습니다. 정책 요약을 검토한 다음 정책 생성을 선택하여 작업을 저장합니다.

정책을 생성한 후 그룹, 사용자 또는 역할에 연결할 수 있습니다. 자세한 정보는 [IAM 자격 증명 권한 추가 및 제거 \(p. 391\)](#) 단원을 참조하십시오.

시각적 편집기를 사용하여 정책 만들기

IAM 콘솔의 시각적 편집기는 JSON 구문을 작성하지 않고 정책을 생성하는 방법을 안내합니다. 시각적 편집기를 사용하여 정책을 생성하는 예제를 보려면 [the section called “자격 증명에 대한 액세스 제어” \(p. 329\)](#) 단원을 참조하십시오.

시각적 편집기를 사용하여 정책을 생성하려면

1. 의 단계를 따라 정책 생성 [IAM 정책 만들기\(콘솔\) \(p. 378\)](#) 마법사를 시작합니다.
2. Visual editor(시각적 편집기) 탭에서 Choose a service(서비스 선택)을 선택합니다. 그런 다음 정책에 추가할 AWS 제품을 선택합니다. 상단의 검색 상자를 사용하여 서비스 목록 결과를 제한할 수 있습니다. 시각적 편집기 권한 블록 내에서 하나의 서비스만 선택할 수 있습니다. 둘 이상의 서비스에 액세스 권한을 부여하려면 Add additional permissions(권한 추가)를 선택하여 여러 개의 권한 블록을 추가합니다.

3. Select actions(작업 선택)을 선택한 다음 정책에 추가할 작업을 선택합니다. 시각적 편집기에는 이전 단계에서 선택한 서비스에서 사용 가능한 작업이 표시됩니다.

작업을 선택하는 방법은 다음과 같습니다.

- 확인란을 사용하여 서비스에 대한 모든 작업을 선택하거나 사전 정의된 액세스 레벨 중 하나에서 모든 작업을 선택합니다.
- 각 액세스 레벨 그룹을 확장하여 개별 작업을 선택합니다.
- add actions(작업 추가)를 선택하여 특정 작업을 입력하거나 와일드카드(*)를 사용하여 여러 개의 작업을 지정합니다.

기본적으로 생성되는 정책은 사용자가 선택하는 작업을 허용합니다. 대신 선택한 작업을 거부하려면 Switch to deny permissions(권한 거부로 전환)을 선택합니다. 기본적으로 IAM은 거부 (p. 531)하기 때문에, 보안 모범 사례로 사용자에게 필요한 작업과 리소스에만 권한을 허용하는 것이 좋습니다. 이것을 "화이트리스트"라고 부르기도 합니다. 다른 문이나 정책에서 허용되는 권한을 별도로 재정의하려는 경우에만 권한을 거부("블랙리스트")하기 위한 JSON 문을 생성해야 합니다. 권한 거부의 수가 늘어나면 권한 문제를 해결하기가 더 어려워질 수 있기 때문에 그 수를 최소한으로 제한하는 것이 좋습니다.

4. 이전 단계에서 선택한 서비스 및 작업이 특정 리소스 (p. 335) 선택을 지원하지 않는 경우 모든 리소스가 선택됩니다. 이러한 경우 이 섹션을 편집할 수 없습니다.

리소스 수준 권한 (p. 335)을 지원하는 작업을 하나 이상 선택하면 시각적 편집기에 해당 리소스가 나열됩니다. 그러면 리소스를 선택하여 정책에 대한 리소스를 지정할 수 있습니다.

리소스를 선택하는 방법은 다음과 같습니다.

- Add ARN(ARN 추가)를 선택하여 리소스에 대한 세부 정보를 제공합니다. 값을 입력하는 대신 모두 선택을 선택하여 지정된 설정을 위한 값에 대한 권한을 제공할 수도 있습니다. 예를 들어, Amazon EC2 읽기 액세스 레벨 그룹을 선택하면 정책의 작업이 instance 리소스 유형을 지원합니다. 리소스에 대해 Region, Account 및 InstanceId 값을 제공해야 합니다. 계정 ID를 제공하지만 리전 및 인스턴스 ID에 대해 모두 선택을 선택한 경우 정책은 계정의 모든 인스턴스에 대해 권한을 부여합니다.
 - Add ARN(ARN 추가)를 선택하여 Amazon 리소스 이름(ARN)별로 리소스를 지정합니다. ARN 필드에 와일드카드(*)를 사용할 수 있습니다(각 콜론 쌍 사이). 자세한 정보는 IAM JSON 정책 요소: Resource (p. 508)를 참조하십시오.
 - 리소스 섹션의 오른쪽 맨 끝에서 모두 선택을 선택하여 특정 유형의 리소스에 대한 권한을 부여합니다.
 - All resources(모든 리소스)를 선택하여 해당 서비스에 대한 모든 리소스를 선택합니다.
5. (선택 사항) Specify request conditions(optional)(요청 조건 지정(선택 사항))를 선택하여 생성하는 정책에 조건을 추가합니다. 조건은 JSON 정책 문의 효과를 제한합니다. 예를 들어 특정 시간 범위 내에 사용자의 요청이 발생하는 경우에만 사용자가 리소스에 대한 작업을 수행할 수 있도록 지정할 수 있습니다. 또한 일반적으로 사용되는 조건을 사용하여 사용자가 멀티 팩터 인증(MFA) 디바이스를 사용하여 인증 받아야 하는지를 제한할 수 있습니다. 또는 요청이 특정 IP 주소 범위에서 발생하도록 요구할 수 있습니다. 정책 조건에서 사용할 수 있는 모든 콘텍스트 키 목록은 ??? 단원을 참조하십시오.

조건을 선택하는 방법은 다음과 같습니다.

- 확인란을 사용하여 일반적으로 사용되는 조건을 선택합니다.
- 조건 추가를 선택하여 다른 조건을 지정합니다. 조건의 조건 키, 한정어, 연산자를 선택한 후 값을 입력합니다. 값을 두 개 이상 추가하려면 Add new value(새 값 추가)를 선택합니다. 해당 값이 논리적 "OR" 연산자로 연결되는 것으로 생각할 수 있습니다. 작업이 완료되면 추가를 선택합니다.

조건을 두 개 이상 추가하려면 다시 조건 추가를 선택합니다. 필요에 따라 반복합니다. 각 조건은 이 시각적 편집기 권한 블록 하나에만 적용됩니다. 권한 블록이 일치하는 것으로 간주되려면 모든 조건이 true여야 합니다. 즉, 이들 조건이 논리적 "AND" 연산자로 연결되는 것으로 간주됩니다.

조건 요소에 대한 자세한 정보는 [IAM JSON 정책 참조 \(p. 498\)](#)에서 [IAM JSON 정책 요소: Condition \(p. 510\)](#) 섹션을 참조하십시오.

6. 더 많은 권한 블록을 추가하려면 Add additional permissions(권한 추가)를 선택합니다. 각 블록에 대해 2~5단계를 반복합니다.
7. 작업이 완료되면 [Review policy]를 선택합니다.

Note

언제든지 Visual editor(시각적 편집기) 및 JSON 탭을 전환할 수 있습니다. 그러나 변경을 수행하거나 Visual editor(시각적 편집기) 탭에서 정책 검토를 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 정보는 [정책 재구성 \(p. 455\)](#) 단원을 참조하십시오.

8. 정책 검토 페이지에서 생성하는 정책에 대한 이름과 설명(선택 사항)을 입력합니다. 정책 요약을 검토하여 의도한 권한이 부여되었는지 확인한 다음 정책 생성을 선택하여 새 정책을 저장합니다.

정책을 생성한 후 그룹, 사용자 또는 역할에 연결할 수 있습니다. 자세한 정보는 [IAM 자격 증명 권한 추가 및 제거 \(p. 391\)](#) 단원을 참조하십시오.

JSON 탭에서 정책 만들기

JSON 탭을 선택하여 JSON에 정책을 입력하거나 붙여 넣을 수 있습니다. 이 방법은 계정에서 사용하기 위해 [예제 정책 \(p. 341\)](#)을 복사할 경우 유용합니다. 또는 JSON 편집기에 고유한 JSON 정책 문서를 입력할 수 있습니다. JSON 탭을 통해 시각적 편집기와 JSON 간에 전환하여 보기를 비교할 수도 있습니다.

JSON 정책 ([p. 305](#)) 문서는 하나 이상의 문으로 구성되어 있습니다. 각 문에는 동일한 효과(Allow 또는 Deny)를 공유하며 동일한 리소스와 조건을 지원하는 모든 작업이 포함되어야 합니다. 한 작업에서 모든 리소스를 지정("*")하도록 요구하고 다른 작업에서 특정 리소스의 [Amazon 리소스 이름\(ARN\)](#)을 지원하는 경우 이들은 두 개의 별개 JSON 문서에 있어야 합니다. IAM 정책에 대한 일반적인 내용은 [정책 및 권한 \(p. 305\)](#) 단원을 참조하십시오. IAM 정책 언어에 대한 자세한 정보는 [IAM JSON 정책 참조 \(p. 498\)](#) 섹션을 참조하십시오.

JSON 정책 편집기를 사용하여 정책을 생성하려면

1. 의 단계를 따라 정책 생성[IAM 정책 만들기\(콘솔\) \(p. 378\)](#) 마법사를 시작합니다.
2. [JSON] 탭을 선택합니다.
3. JSON 정책 문서를 입력하거나 붙여 넣습니다. IAM 정책 언어에 대한 자세한 정보는 [IAM JSON 정책 참조 \(p. 498\)](#) 섹션을 참조하십시오.
4. 작업이 완료되면 [Review policy]를 선택합니다. [정책 검사기 \(p. 382\)](#)가 모든 구문 오류를 보고합니다.

Note

언제든지 Visual editor(시각적 편집기) 및 JSON 탭을 전환할 수 있습니다. 그러나 변경을 수행하거나 Visual editor(시각적 편집기) 탭에서 정책 검토를 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 정보는 [정책 재구성 \(p. 455\)](#) 단원을 참조하십시오.

5. 정책 검토 페이지에서 생성하는 정책에 대한 이름과 설명(선택 사항)을 입력합니다. 정책 요약을 검토하여 정책이 부여한 권한을 확인합니다. 그런 다음 [Create policy]를 선택하여 작업을 저장합니다.

정책을 생성한 후 그룹, 사용자 또는 역할에 연결할 수 있습니다. 자세한 정보는 [IAM 자격 증명 권한 추가 및 제거 \(p. 391\)](#) 단원을 참조하십시오.

IAM 정책 생성(AWS CLI)

AWS Command Line Interface를 사용하여 IAM 정책 또는 인라인 정책을 만들 수 있습니다(AWS CLI).

고객 관리형 정책을 만들려면(AWS CLI)

다음 명령을 사용합니다.

- [create-policy](#)

보안 주체 개체(그룹, 사용자 또는 역할)에 대한 인라인 정책을 만들려면(AWS CLI)

다음 명령 중 하나를 사용합니다.

- [put-group-policy](#)
- [put-role-policy](#)
- [put-user-policy](#)

Note

역할에 따라 달라지는 서비스에만 [서비스 연결 역할 \(p. 154\)](#)에 대한 인라인 정책을 포함할 수 있습니다. 서비스가 이 기능을 지원하는지 여부를 확인하려면 서비스에 대한 [AWS 설명서](#)를 참조하십시오.

IAM 정책 생성(AWS API)

AWS API를 사용하여 IAM 정책 또는 인라인 정책을 만들 수 있습니다.

고객 관리형 정책을 만들려면(AWS API)

다음 작업을 호출합니다.

- [CreatePolicy](#)

보안 주체 개체(그룹, 사용자 또는 역할)에 대한 인라인 정책을 만들려면(AWS API)

다음 작업 중 하나를 호출합니다.

- [PutGroupPolicy](#)
- [PutRolePolicy](#)
- [PutUserPolicy](#)

Note

역할에 따라 달라지는 서비스에만 [서비스 연결 역할 \(p. 154\)](#)에 대한 인라인 정책을 포함할 수 있습니다. 서비스가 이 기능을 지원하는지 여부를 확인하려면 서비스에 대한 [AWS 설명서](#)를 참조하십시오.

JSON 정책 검증

정책 검사기는 신규 및 기존 IAM 액세스 제어 정책을 자동으로 검사하여 IAM 정책 문법의 준수 여부를 확인합니다. 여기에서 정책이란 [IAM 정책 문법](#)에 따라 작성된 JSON 문서를 말합니다. 정책을 연결할 AWS 사용자, 그룹 또는 역할의 액세스 권한을 정의합니다. 정책 검사기에서 정책 문법을 준수하지 않은 정책을 발견하면 해당 정책을 수정하라는 메시지가 표시됩니다. 정책이 문법을 준수하지 않는 경우에는 정책 검사기만 사용할 수 있습니다.

다음 방법을 사용하여 정책 검사기에 액세스할 수도 있습니다.

1. JSON 정책 생성 – 정책 검토를 선택하면 새 JSON 정책을 생성할 때 정책 검사기가 자동으로 실행됩니다. 정책이 유효하지 않으면 알림이 표시되고 계속 진행하기 전에 문제를 해결해야 합니다.

2. JSON 정책 편집 – 정책 검토를 선택하면 기존 JSON 정책을 편집할 때 정책 검사기가 자동으로 실행됩니다. 정책이 유효하지 않으면 알림이 표시되고 계속 진행하기 전에 문제를 해결해야 합니다. 정책 검사기를 도입하기 전에 설정된 기존 정책에 오류가 있어도 그대로 실행됩니다. 그러나 정책 구문 오류를 수정하지 않으면 해당 정책을 편집 및 저장할 수 없습니다.

Note

정책 검사기는 JSON 정책 구문 및 문법만 검사합니다. ARN, 작업 이름 또는 조건 키가 올바른지는 검사하지 않습니다.

IAM 정책 시뮬레이터로 IAM 정책 테스트하기

IAM 정책의 사용 방식과 이유에 대한 자세한 정보는 [정책 및 권한 \(p. 305\)](#) 단원을 참조하십시오.

다음 위치에서 IAM 정책 시뮬레이터 콘솔에 액세스할 수 있습니다. <https://policysim.aws.amazon.com/>

아래는 IAM 정책 시뮬레이터 사용에 필요한 기본 정보를 소개한 동영상입니다.

[IAM 정책 시뮬레이터 시작하기](#)

IAM 정책 시뮬레이터는 다음과 같은 방법으로 IAM 및 리소스 기반 정책을 테스트하여 식별된 문제를 해결하는데 사용됩니다.

- AWS 계정의 IAM 사용자, 그룹 또는 역할에 연결된 정책을 테스트합니다. 사용자, 그룹 또는 역할에 추가된 정책이 다수일 때는 모든 정책을 테스트하거나, 테스트할 정책만 따로 선택할 수 있습니다. 특정 리소스에 대해 선택한 정책에서 어떤 작업을 허용하거나 거부하는지 테스트할 수 있습니다.
- Amazon S3 버킷, Amazon SQS 대기열, Amazon SNS 주제, Amazon S3 Glacier 볼트 등과 같은 AWS 리소스에 연결된 정책을 테스트합니다.
- AWS 계정이 [AWS Organization](#)에 속한 경우, 조직 제어 정책이 IAM 정책 및 리소스 정책에 미치는 영향을 테스트할 수 있습니다.
- 사용자, 그룹 또는 역할에 아직 추가되지 않은 새로운 정책을 시뮬레이터에 입력 또는 복사하여 테스트합니다. 이는 시뮬레이션에서만 사용되며 저장되지 않습니다. 참고: 리소스 기반 정책을 시뮬레이터에 입력하거나 복사하지 마십시오. 시뮬레이터에서 리소스 기반 정책을 사용하려면 시뮬레이션에 리소스를 포함 시킨 후 확인란을 선택하여 해당 리소스의 정책을 시뮬레이션에 포함 시켜야 합니다.
- 선택한 서비스, 작업 및 리소스에 대한 정책을 테스트합니다. 예를 들어 정책이 특정 버킷의 Amazon S3 서비스에서 주체가 `ListAllMyBuckets`, `CreateBucket` 및 `DeleteBucket` 작업을 수행할 수 있도록 허용하는지 확인하기 위해 테스트할 수 있습니다.
- 테스트할 정책에서 키가 지정되어 있는 경우에는 테스트할 정책의 `Condition` 요소에 포함된 IP 주소나 날짜 같은 콘텍스트 키를 제공하여 실제 시나리오를 시뮬레이션합니다.
- 어떤 정책 문이 특정 리소스 또는 작업에 대한 액세스를 허용하거나 거부하는지 식별합니다.

주제

- [IAM 정책 시뮬레이터의 원리 \(p. 383\)](#)
- [IAM 정책 시뮬레이터를 사용하는 데 필요한 권한 \(p. 384\)](#)
- [IAM 정책 시뮬레이터 사용\(콘솔\) \(p. 386\)](#)
- [IAM 정책 시뮬레이터의 사용\(AWS CLI 및 AWS API\) \(p. 391\)](#)

IAM 정책 시뮬레이터의 원리

시뮬레이터는 선택한 정책을 평가한 후 지정 작업 각각에 대해 유효한 권한을 결정합니다. 정책 평가 엔진으로는 실제로 AWS 서비스를 요청할 때와 동일한 엔진을 사용하지만 다음과 같은 방식에서 실시간 AWS 환경과는 차이가 있습니다.

- 시뮬레이터는 실제로 AWS 서비스를 요청하는 것은 아니기 때문에 실행 중인 AWS 환경을 변경하지 않고 요청을 안전하게 테스트할 수 있습니다.
- 시뮬레이터는 선택한 작업 중 실행 중인 작업은 시뮬레이션하지 않기 때문에 시뮬레이션된 요청에 대한 응답을 보고하지 않습니다. 요청된 작업이 허용되는지 아니면 거부되는지 여부만 결과로 반환됩니다.
- 시뮬레이터에서 정책을 편집할 경우 이러한 변경은 시뮬레이터에만 영향을 줍니다. AWS 계정의 해당 정책은 변함없이 그대로 유지됩니다.

IAM 정책 시뮬레이터를 사용하는 데 필요한 권한

정책 시뮬레이터 콘솔 또는 정책 시뮬레이터 API를 사용하여 정책을 테스트할 수 있습니다. 기본적으로 콘솔 사용자는 사용자, 그룹 또는 역할에 아직 연결되지 않은 정책을 시뮬레이터에 입력하거나 복사하여 테스트할 수 있습니다. 이러한 정책은 시뮬레이션에만 사용되며 민감한 정보를 공개하지 않습니다. API 사용자가 연결되지 않은 정책을 테스트하려면 권한이 있어야 합니다. AWS 계정에서 IAM 사용자, 그룹 또는 역할에 연결된 정책을 콘솔 또는 API 사용자가 테스트하도록 허용하려면, 이러한 정책을 검색할 수 있는 권한을 부여해야 합니다. 리소스 기반 정책을 테스트하려면 사용자가 리소스의 정책을 검색할 수 있는 권한을 보유해야 합니다.

사용자가 시뮬레이션할 수 있는 콘솔 및 API 정책의 예시는 [the section called “정책 예제: AWS Identity and Access Management\(IAM\)” \(p. 342\)](#) 단원을 참조하십시오.

정책 시뮬레이터 콘솔을 사용하는 데 필요한 권한

AWS 계정에서 IAM 사용자, 그룹 또는 역할에 연결된 정책을 사용자가 테스트하도록 허용하려면, 이러한 정책을 검색할 수 있는 권한을 사용자에게 부여해야 합니다. 리소스 기반 정책을 테스트하려면 사용자가 리소스의 정책을 검색할 수 있는 권한을 보유해야 합니다.

사용자, 그룹 또는 역할에 연결된 정책에 대해 정책 시뮬레이터 콘솔의 사용을 허용하는 정책의 예시는 [IAM: 정책 시뮬레이터 콘솔 액세스 \(p. 363\)](#) 단원을 참조하십시오.

특정 경로를 지닌 사용자에 대해서만 정책 시뮬레이터 콘솔의 사용을 허용하는 정책의 예시는 [IAM: 사용자 경로를 바탕으로 정책 시뮬레이터 콘솔 액세스 \(p. 367\)](#) 단원을 참조하십시오.

한 가지 유형의 엔터티에 대해서만 정책 시뮬레이터 콘솔의 사용을 허용하는 정책을 만들려면 다음의 절차에 따릅니다.

콘솔 사용자가 사용자를 위한 정책을 시뮬레이션하도록 허용하는 방법

정책에 다음의 작업을 포함시킵니다.

- iam:GetGroupPolicy
- iam:GetPolicy
- iam:GetPolicyVersion
- iam:GetUser
- iam:GetUserPolicy
- iam>ListAttachedUserPolicies
- iam>ListGroupsForUser
- iam>ListGroupPolicies
- iam>ListUserPolicies
- iam>ListUsers

콘솔 사용자가 그룹을 위한 정책을 시뮬레이션하도록 허용하는 방법

정책에 다음의 작업을 포함시킵니다.

- iam:GetGroup
- iam:GetGroupPolicy
- iam:GetPolicy
- iam:GetPolicyVersion
- iam>ListAttachedGroupPolicies
- iam>ListGroupPolicies
- iam>ListGroups

콘솔 사용자가 역할에 대한 정책을 시뮬레이션하도록 허용하는 방법

정책에 다음의 작업을 포함시킵니다.

- iam:GetPolicy
- iam:GetPolicyVersion
- iam:GetRole
- iam:GetRolePolicy
- iam>ListAttachedRolePolicies
- iam>ListRolePolicies
- iam>ListRoles

리소스 기반 정책을 테스트하려면 사용자가 리소스의 정책을 검색할 수 있는 권한을 보유해야 합니다.

콘솔 사용자가 Amazon S3 버킷에서 리소스 기반 정책을 테스트하도록 허용하는 방법

정책에 다음의 작업을 포함시킵니다.

- s3:GetBucketPolicy
- s3:GetObject

예를 들어, 다음의 정책에서 이들 작업을 사용하여 특정 Amazon S3 버킷에서 콘솔 사용자가 리소스 기반 정책을 시뮬레이션하도록 허용합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "s3:GetBucketPolicy",  
                "s3:GetObject"  
            ],  
            "Effect": "Allow",  
            "Resource": "arn:aws:s3:::<BUCKET-NAME>/*"  
        }  
    ]  
}
```

콘솔 사용자가 [AWS Organizations](#)를 위한 정책을 테스트하도록 허용하는 방법

정책에 다음의 작업을 포함시킵니다.

- organizations:DescribePolicy

- organizations>ListPolicies
- organizations>ListPoliciesForTarget
- organizations>ListTargetsForPolicy

API 정책 시뮬레이터를 사용하는 데 필요한 권한

정책 시뮬레이터 API 작업 [GetContextKeyForCustomPolicy](#) 및 [SimulateCustomPolicy](#)는 정책을 문자열로 API 전달함으로써 사용자가 사용자, 그룹 또는 역할에 아직 연결되어 있지 않은 정책을 테스트할 수 있도록 해 줍니다. 이러한 정책은 시뮬레이션에만 사용되며 민감한 정보를 공개하지 않습니다. AWS 계정에서 IAM 사용자, 그룹 또는 역할에 연결된 정책을 콘솔 또는 API 사용자가 테스트하도록 허용하려면, [GetContextKeyForPrincipalPolicy](#) 및 [SimulatePrincipalPolicy](#)를 호출할 수 있는 권한을 사용자에게 부여해야 합니다.

현재 AWS 계정에서 사용자, 그룹 또는 역할에 연결되어 있지 않은 정책과 연결된 정책에 대해 정책 시뮬레이터 API의 사용을 허용하는 정책의 예시는 [IAM: 정책 시뮬레이터 API에 액세스 \(p. 362\)](#) 단원을 참조하십시오.

한 가지 유형의 정책에 대해서만 정책 시뮬레이터 API의 사용을 허용하는 정책을 만들려면 다음의 절차에 따릅니다.

API 사용자가 API에 문자열로 직접 전달되는 정책을 시뮬레이션하도록 허용하는 방법

정책에 다음의 작업을 포함시킵니다.

- iam:GetContextKeysForCustomPolicy
- iam:SimulateCustomPolicy

API 사용자로 하여금 IAM 사용자, 그룹 또는 역할에 연결된 정책을 시뮬레이션하도록 하는 방법

정책에 다음의 작업을 포함시킵니다.

- iam:GetContextKeysForPrincipalPolicy
- iam:SimulatePrincipalPolicy

예를 들어, Alice라는 사용자에게 할당된 정책을 시뮬레이션할 수 있는 권한을 Bob이라는 사용자에게 부여하려면, Bob에게 arn:aws:iam::777788889999:user/alice라는 리소스에 액세스할 수 있는 권한을 부여해야 합니다.

특정 경로를 지닌 사용자에 대해서만 정책 시뮬레이터 API의 사용을 허용하는 정책의 예시는 [IAM: 사용자 경로를 바탕으로 정책 시뮬레이터 API 액세스 \(p. 366\)](#) 단원을 참조하십시오.

IAM 정책 시뮬레이터 사용(콘솔)

기본적으로 사용자는 사용자, 그룹 또는 역할에 아직 연결되지 않은 정책을 정책 시뮬레이터 콘솔에 입력하거나 복사하여 테스트할 수 있습니다. 이러한 정책은 시뮬레이션에만 사용되며 민감한 정보를 공개하지 않습니다.

정책 시뮬레이터 콘솔을 사용하여 사용자, 그룹 또는 역할에 연결되어 있지 않은 정책을 테스트하려면(콘솔)

1. <https://policysim.aws.amazon.com/>에서 IAM 정책 시뮬레이터 콘솔을 엽니다.
2. 페이지의 상단에 있는 Mode:(모드:) 메뉴에서 New Policy(새 정책)을 선택합니다.
3. Policy Sandbox(정책 샌드박스)에서 새 정책 생성을 선택합니다.

4. 시뮬레이터에 입력하거나 복사하여 붙여 넣고, 다음 단계의 설명에 따라 시뮬레이터를 사용합니다.

IAM 정책 시뮬레이터 콘솔을 사용하는 데 필요한 권한을 받은 후에는 시뮬레이터를 사용하여 IAM 사용자, 그룹, 역할 또는 리소스 정책을 테스트할 수 있습니다.

정책 시뮬레이터를 사용하려면(콘솔)

1. <https://policysim.aws.amazon.com/>에서 IAM 정책 시뮬레이터 콘솔을 엽니다.

Note

IAM 사용자로 정책 시뮬레이터에 로그인하려면 고유의 로그인 URL을 사용하여 AWS Management 콘솔에 로그인합니다. 그런 다음 <https://policysim.aws.amazon.com/>으로 이동합니다. IAM 사용자 권한의 로그인에 대한 자세한 정보는 [IAM 사용자가 AWS에 로그인하는 방법 \(p. 69\)](#)를 참조하십시오.

시뮬레이터가 Existing Policies(기존 정책) 모드로 열리며 Users, Groups, and Roles(사용자, 그룹 및 역할) 아래 계정에 속한 IAM 사용자가 표시됩니다.

2. 작업에 적합한 옵션을 선택합니다.

테스트 대상	수행할 작업:
사용자에게 연결된 정책	Users, Groups, and Roles(사용자, 그룹 및 역할) 목록에서 사용자를 선택합니다. 그런 다음 사용자를 선택합니다.
그룹에 연결된 정책	Users, Groups, and Roles(사용자, 그룹 및 역할) 목록에서 그룹을 선택합니다. 그런 다음 그룹을 선택합니다.
역할에 연결된 정책	Users, Groups, and Roles(사용자, 그룹 및 역할) 목록에서 역할을 선택합니다. 그런 다음 역할을 선택합니다.
리소스에 연결된 정책	Step 8 단원을 참조하십시오.
사용자 지정 정책	상단의 모드 목록에서 New Policy(새 정책)을 선택합니다. 그런 다음 좌측의 Policy Sandbox(정책 샌드박스) 창에서 새 정책 생성을 선택하여 정책을 입력하거나 붙여넣은 다음 적용을 선택합니다.

도움말

그룹에 연결된 정책을 테스트하려면 IAM 정책 시뮬레이터를 [IAM 콘솔](#)에서 직접 실행한 후 탐색 창에서 그룹을 선택합니다. 정책을 테스트하려는 그룹 이름을 선택한 후 권한 탭을 선택합니다. Inline Policies(인라인 정책) 또는 Managed Policies(관리형 정책) 섹션에서 테스트하려는 정책을 찾습니다. 해당 정책의 작업 열에서 Simulate Policy(정책 시뮬레이션)을 선택합니다. 사용자에게 연결된 고객 관리형 정책을 테스트하려면 탐색 창에서 사용자를 선택합니다. 정책을 테스트하고자 하는 사용자의 이름을 선택합니다. 그런 다음 권한 탭을 선택하고 테스트할 정책을 확장합니다. 오른쪽 맨 끝에서 Simulate Policy(정책 시뮬레이션)을 선택합니다. IAM 정책 시뮬레이터가 새 창으로 열리면서 선택한 정책을 정책 창에 표시합니다.

3. (선택 사항) 계정이 [AWS Organization](#)에 속한 경우, 시뮬레이션된 사용자의 계정에 영향을 미치는 서비스 제어 정책(SCP)을 IAM 정책 및 리소스 정책과 함께 정책 창에 표시합니다. SCP는 조직 또는 조직 단위(OU)에 최대 권한을 지정하는 JSON 정책입니다. SCP는 멤버 계정의 엔터티에 대한 권한을 제한합니다. OCP가 서비스 또는 작업을 차단하는 경우 해당 계정에 있는 어떤 엔터티도 해당 서비스에 액세스하거나 해당 작업을 수행할 수 없습니다. 이는 관리자가 IAM 또는 리소스 정책을 통해 해당 서비스 또는 작업에 명시적으로 권한을 부여하는 경우에도 해당합니다. 시뮬레이션에서 OCP를 제거하려면 OCP 이름 옆의 확인란에서 선택을 취소하면 됩니다. OCP 콘텐츠를 보려면 해당 OCP 이름을 선택합니다.

계정이 조직에 속하지 않은 경우, 시뮬레이션할 SCP가 없습니다.

4. (선택 사항) 사용자, 그룹 또는 역할에 연결된 정책의 일부만 테스트하려면 정책 창에서 제외하려는 각 정책 옆에 있는 확인란의 선택을 취소합니다.
5. Policy Simulator(정책 시뮬레이터)에서 Select service(서비스 선택)를 선택한 후 테스트할 서비스를 선택합니다. 그런 다음 Select actions(작업 선택)을 선택하고 테스트할 작업을 한 개 이상 선택합니다. 메뉴에는 한 번에 한 서비스에 대해 가능한 선택만 표시되지만 선택한 모든 서비스와 작업이 Action Settings and Results(작업 설정 및 결과)에 나타납니다.
6. (선택 사항) Step 2 및 Step 4에서 선택하는 정책 중 하나라도 AWS 글로벌 조건 키 (p. 551)를 지닌 조건을 포함하는 경우, 해당 키에 대한 값을 제공합니다. 글로벌 설정 섹션을 확장하고 표시된 키 이름의 값을 입력하여 키에 대한 값을 제공할 수 있습니다.

Warning

조건 키의 값을 비워 놓으면 해당 키가 시뮬레이션 중에 무시됩니다. 이로 인해 오류가 발생하고 시뮬레이션이 실행되지 않는 경우가 있습니다. 또한 시뮬레이션은 실행되지만 결과를 신뢰할 수 없습니다. 이 경우 조건 키에 대한 값이나 변수를 포함하는 실제 조건과 시뮬레이션이 일치하지 않습니다.

7. (선택 사항) 선택한 각 작업은 Action Settings and Results(작업 설정 및 결과) 목록에 표시되고 실제로 시뮬레이션을 실행할 때까지는 권한 열에 Not simulated(시뮬레이션되지 않음)이라고 표시됩니다. 시뮬레이션을 실행하기 전에 리소스를 포함하는 각 작업을 구성할 수 있습니다. 특정 시나리오에 맞게 개별 작업을 구성하려면 화살표를 선택하여 작업 행을 확장합니다. 작업이 리소스 수준 권한을 지원할 경우 액세스를 테스트하려는 특정 리소스의 [Amazon 리소스 이름\(ARN\)](#)을 입력할 수 있습니다. 기본적으로 각 리소스는 와일드카드(*)로 설정됩니다. 또한 임의의 조건 콘텍스트 키에 대한 값을 지정할 수 있습니다. 앞에서도 설명했듯이 값이 비어 있는 키는 무시되며, 이로 인해 시뮬레이션이 실패하거나 신뢰할 수 없는 결과가 반환될 수 있습니다.
 - a. 작업 이름 옆에 있는 화살표를 선택하여 각 행을 확장하고 해당 시나리오에 맞게 작업을 정확하게 시뮬레이션하는 데 필요한 추가 정보를 구성합니다. 작업에 리소스 수준 권한이 필요할 경우 액세스를 시뮬레이션하려는 특정 리소스의 [Amazon 리소스 이름\(ARN\)](#)을 입력할 수 있습니다. 기본적으로 각 리소스는 와일드카드(*)로 설정됩니다.
 - b. 작업이 리소스 수준 권한을 지원하지만 그러한 권한이 필요하지 않을 경우 리소스 추가를 선택하여 시뮬레이션에 추가하려는 리소스 유형을 선택합니다.
 - c. 선택한 정책이 해당 작업의 서비스에 대한 콘텍스트 키를 참조하는 Condition 요소를 포함할 경우 해당 키 이름이 작업 아래에 표시됩니다. 지정한 리소스에 대한 해당 작업의 시뮬레이션 중에 사용할 값을 지정할 수 있습니다.

여러 리소스 유형 그룹이 필요한 작업

일부 작업은 서로 다른 환경에서 여러 리소스 유형이 필요합니다. 리소스 유형의 각 그룹은 시나리오와 관련이 있습니다. 이 중 하나가 시뮬레이션에 적용될 경우 리소스를 선택하면 시뮬레이터가 해당 시나리오에 적합한 리소스 유형을 필요로 합니다. 다음 목록에는 지원되는 각 시나리오 옵션과 시뮬레이션을 실행하기 위해 정의해야 하는 리소스가 나와 있습니다.

다음의 Amazon EC2 시나리오 각각에 대해 instance, image, security-group 리소스를 지정해야 합니다. 시나리오에 EBS 볼륨이 포함될 경우에는 해당 volume을 리소스로 지정해야 합니다. Amazon EC2 시나리오에 가상 프라이빗 클라우드(VPC)가 포함될 경우에는 network-interface 리소스를 제공해야 합니다. IP 서브넷이 포함될 경우에는 subnet 리소스를 지정해야 합니다. Amazon EC2 시나리오 옵션에 대한 자세한 정보는 Amazon EC2 사용 설명서의 [지원되는 플랫폼](#)을 참조하십시오.

- EC2-Classic-InstanceStore

인스턴스, 이미지, 보안 그룹

- EC2-Classic-EBS

인스턴스, 이미지, 보안 그룹, 볼륨

- EC2-VPC-InstanceStore

인스턴스, 이미지, 보안 그룹, 네트워크 인터페이스

- EC2-VPC-InstanceStore-Subnet

인스턴스, 이미지, 보안 그룹, 네트워크 인터페이스, 서브넷

- EC2-VPC-EBS

인스턴스, 이미지, 보안 그룹, 네트워크 인터페이스, 볼륨

- EC2-VPC-EBS-Subnet

인스턴스, 이미지, 보안 그룹, 네트워크 인터페이스, 서브넷, 볼륨

8. (선택 사항) 시뮬레이션에 리소스 기반 정책을 포함하려면 먼저 해당 리소스에 대해 시뮬레이션하려는 작업을 Step 5에서 선택해야 합니다. 선택한 작업의 행을 확장하고 시뮬레이션하려는 정책을 포함하는 리소스의 ARN을 입력합니다. 그런 다음 ARN 텍스트 상자 옆의 리소스 정책 포함(Include Resource Policy)을 선택합니다. IAM 정책 시뮬레이터는 현재, Amazon S3(리소스 기반 정책만 해당. ACL은 현재 지원되지 않음), Amazon SQS, Amazon SNS 및 잠겨 있지 않은 Glacier 볼트(잠겨 있는 볼트는 현재 지원되지 않음) 서비스의 리소스 기반 정책만 지원합니다.

9. 상단 오른쪽 모서리 부분에서 Run Simulation(시뮬레이션 실행)을 선택합니다.

Action Settings and Results(작업 설정 및 결과)의 각 행에 있는 권한 열에 지정된 리소스에 대한 해당 작업의 시뮬레이션 결과가 표시됩니다.

10. 정책의 어떤 문이 작업을 허용하거나 거부하는지 확인하려면 권한 열에서 **N matching statement(s)**(일치하는 문 N개) 링크를 선택하여 행을 확장한 후 Show statement(문 표시) 링크를 선택합니다. 정책 창에 해당 정책이 표시되고 시뮬레이션 결과에 영향을 준 문이 강조 표시됩니다.

Note

작업이 암묵적으로 거부된 경우, 즉 명시적으로 허용되지 않아 작업이 거부된 경우에만 목록 및 Show statement(문 표시) 옵션이 표시되지 않습니다.

IAM 정책 시뮬레이터 콘솔 메시지 문제 해결

다음 표에는 IAM 정책 시뮬레이터 사용 시 나타날 수 있는 정보 메시지와 경고 메시지가 나와 있습니다. 그 밖에 문제 해결에 필요한 단계도 나와 있습니다.

Message	문제 해결 단계
This policy has been edited. Changes will not be saved to your account.	작업이 필요하지 않음 이것은 정보 메시지입니다. IAM 정책 시뮬레이터에서 기존 정책을 편집하더라도 AWS 계정에서는 변경 사항이 적용되지 않습니다. 시뮬레이터에서는 테스트 목적으로만 정책을 변경할 수 있습니다.
Cannot get the resource policy. 사유: ## ## ##	요청된 리소스 기반 정책에 시뮬레이터가 액세스할 수 없습니다. 지정된 리소스 ARN이 정확하며, 시뮬레이션을 실행하는 사용자가 리소스의 정책을 읽을 수 있는 권한이 있는지 확인하십시오.
One or more policies require values in the simulation settings. The simulation might fail without these values.	이 메시지는 테스트하려는 정책에 포함되어 있는 조건 키 또는 변수 값을 Simulation Settings(시뮬레이션 설정)에 입력하지 않은 경우 나타납니다. 이 메시지를 닫으려면 Simulation Settings(시뮬레이션 설정)을 선택한 다음 각 조건 키 또는 변수 값을 입력합니다.

Message	문제 해결 단계
You have changed policies. These results are no longer valid.	<p>이 메시지는 결과가 Results 창에 표시되는 중에 선택한 정책을 변경하였을 때 나타납니다. Results 창에 표시되는 결과는 동적으로 업데이트되지 않습니다.</p> <p>이 메시지를 닫으려면 정책 창의 변경에 따라 다시 Run Simulation(시뮬레이션 실행)을 선택하여 새로운 시뮬레이션 결과를 표시합니다.</p>
The resource you typed for this simulation does not match this service.	<p>이 메시지는 현재 시뮬레이션에서 선택한 서비스와 일치하지 않는 Amazon 리소스 이름(ARN)을 Simulation Settings(시뮬레이션 설정) 창에 입력했을 때 나타납니다. 예를 들어, Amazon DynamoDB 리소스의 ARN을 지정하고 시뮬레이션할 서비스로 Amazon Redshift를 선택하면 이 메시지가 나타납니다.</p> <p>이 메시지를 닫으려면 다음 중 한 가지를 실행합니다.</p> <ul style="list-style-type: none"> Simulation Settings(시뮬레이션 설정) 창의 상자에서 ARN을 삭제합니다. Simulation Settings(시뮬레이션 설정)에서 지정한 ARN과 일치하는 서비스를 선택합니다.
이 작업은 Amazon S3 ACL 또는 Glacier 볼트 잠금 정책 같은 리소스 기반 정책 외에 특수 액세스 제어 방식을 지원하는 서비스에 속합니다. The policy simulator does not support these mechanisms, so the results can differ from your production environment.	<p>작업이 필요하지 않음</p> <p>이것은 정보 메시지입니다. 현재 버전에서 시뮬레이터는 사용자와 그룹에 연결된 정책을 평가하며, Amazon S3, Amazon SQS, Amazon SNS, Glacier에 대한 리소스 기반 정책을 평가할 수 있습니다. 정책 시뮬레이터가 다른 AWS 서비스에서 지원하는 액세스 제어 방식을 모두 지원하는 것은 아닙니다.</p>
DynamoDB FGAC is currently not supported.	<p>작업이 필요하지 않음</p> <p>이 정보 메시지는 세분화된 액세스 제어를 가리킵니다. 이는 IAM 정책 조건을 사용하여 DynamoDB 테이블 및 인덱스의 개별 데이터 항목과 속성, 그리고 여기에서 실행 가능한 작업에 액세스할 수 있는 사용자를 결정하는 능력을 말합니다. 현재 버전의 IAM 정책 시뮬레이터는 이 유형의 정책 조건을 지원하지 않습니다. DynamoDB FGAC에 대한 자세한 정보는 DynamoDB에 대한 세분화된 액세스 제어를 참조하십시오.</p>
You have policies that do not comply with the policy syntax. You can use the Policy Validator to review and accept the recommended updates to your policies.	<p>이 메시지는 IAM 정책 문법을 위반하는 정책이 있는 경우 정책 목록 상단에 나타납니다. 이러한 정책은 JSON 정책 검증 (p. 382)의 지침에 따른 시뮬레이션을 통해 식별하여 위반 문제를 해결해야 합니다.</p>
This policy must be updated to comply with the latest policy syntax rules.	<p>이 메시지는 IAM 정책 문법을 위반하는 정책이 있는 경우에 표시됩니다. 이러한 정책은 JSON 정책 검증 (p. 382)의 지침에 따른 시뮬레이션을 통해 식별하여 위반 문제를 해결해야 합니다.</p>

IAM 정책 시뮬레이터의 사용(AWS CLI 및 AWS API)

정책 시뮬레이터 명령어는 다음의 2가지 작업을 수행하는 데 일반적으로 API 작업 호출이 필요합니다.

- 정책을 평가하고 정책이 참조하는 콘텍스트 키 목록을 반환합니다. 어떤 콘텍스트 키가 참조되는지 알아야 다음 단계에서 콘텍스트 키에 값을 제공할 수 있습니다.
- 시뮬레이션 중에 사용되는 작업, 리소스, 콘텍스트 키의 목록을 제공하여 정책을 시뮬레이션합니다.

보안 상의 이유로 API 작업은 2개의 그룹으로 나뉘어 있습니다.

- API에 직접 문자열로 전달되는 정책만을 시뮬레이션하는 API 작업. 이 세트에는 [GetContextKeysForCustomPolicy](#) 및 [SimulateCustomPolicy](#)가 포함됩니다.
- 지정된 IAM 사용자, 그룹, 역할 또는 리소스에 연결된 정책을 시뮬레이션하는 API 작업. 이러한 API 작업은 다른 IAM 주체에 할당된 권한의 세부 정보를 알려주기 때문에 이 API 작업에 대한 액세스 제한을 고려해 보아야 합니다. 이 세트에는 [GetContextKeysForPrincipalPolicy](#) 및 [SimulatePrincipalPolicy](#)가 포함됩니다. API 작업 액세스 제한에 대한 자세한 정보는 [정책 예제: AWS Identity and Access Management\(IAM\) \(p. 342\)](#) 단원을 참조하십시오.

두 경우 모두 API 작업은 1개 이상의 정책들이 작업 및 리소스 목록에 미치는 영향을 시뮬레이션합니다. 각 작업은 각 리소스와 짹을 이루고, 시뮬레이션은 정책이 리소스에 대한 작업을 허용 또는 거부하는지 여부를 결정합니다. 또한, 정책이 참조하는 모든 콘텍스트 키에 대한 값을 제공할 수 있습니다. 정책이 참조하는 콘텍스트 키 목록은 [GetContextKeysForCustomPolicy](#) 또는 [GetContextKeysForPrincipalPolicy](#)를 호출하여 확인할 수 있습니다. 콘텍스트 키에 대한 값을 제공하지 않는다 해도 시뮬레이션은 여전히 실행되고 있지만, 시뮬레이터가 평가 시에 콘텍스트 키를 포함할 수 없기 때문에 그 결과를 신뢰하지 못할 수 있습니다.

조건 키 목록을 확인하려면(AWS CLI, AWS API)

다음을 사용하여 정책 목록을 평가하고, 정책에 사용된 콘텍스트 키 목록을 반환합니다.

- AWS CLI: `aws iam get-context-keys-for-custom-policy` 및 `aws iam get-context-keys-for-principal-policy`
- AWS API: [GetContextKeysForCustomPolicy](#) 및 [GetContextKeysForPrincipalPolicy](#)

IAM 정책을 시뮬레이션하려면(AWS CLI, AWS API)

다음을 통해 IAM 정책을 시뮬레이션하여 사용자의 유효 권한을 확인합니다.

- AWS CLI: `aws iam simulate-custom-policy` 및 `aws iam simulate-principal-policy`
- AWS API: [SimulateCustomPolicy](#) 및 [SimulatePrincipalPolicy](#)

IAM 자격 증명 권한 추가 및 제거

정책을 사용하여 자격 증명(사용자, 그룹 또는 역할)에 대한 권한을 정의합니다. AWS Management 콘솔, AWS Command Line Interface(AWS CLI) 또는 AWS API를 사용하여 자격 증명에 대한 IAM 정책을 첨부 및 분리하여 사용 권한을 추가 및 제거할 수 있습니다. 정책을 사용하여 동일한 방법으로 엔터티(사용자 또는 역할)에 대한 [권한 경계 \(p. 317\)](#)만 설정할 수 있습니다. 권한 경계는 엔터티가 가질 수 있는 최대 권한을 제어하는 고급 AWS 기능입니다.

주제

- [용어 \(p. 392\)](#)
- [자격 증명 작업 보기 \(p. 392\)](#)
- [IAM 자격 증명 권한 추가\(콘솔\) \(p. 393\)](#)

- [IAM 자격 증명 권한 제거\(콘솔\) \(p. 394\)](#)
- [IAM 정책 추가\(AWS CLI\) \(p. 395\)](#)
- [IAM 정책 제거\(AWS CLI\) \(p. 396\)](#)
- [IAM 정책 추가\(AWS API\) \(p. 397\)](#)
- [IAM 정책 제거\(AWS API\) \(p. 398\)](#)

용어

권한 정책을 자격 증명(사용자, 그룹, 역할)과 연결할 때, 관리형 정책을 사용하는지 아니면 인라인 정책을 사용하는지에 따라 용어와 절차가 달라집니다.

- **연결** – 관리형 정책에 사용됩니다. 자격 증명(사용자, 그룹 또는 역할)에 관리형 정책을 연결합니다. 정책을 연결하면 정책의 해당 권한이 자격 증명에 적용됩니다.
- **분리** – 관리형 정책에 사용됩니다. 엔터티(사용자, 그룹, 역할)에서 관리형 정책을 분리합니다. 정책을 분리하면 보안 주체 개체에서 해당 권한이 제거됩니다.
- **포함** – 인라인 정책에 사용됩니다. 자격 증명(사용자, 그룹 또는 역할)에 인라인 정책을 포함시킵니다. 정책을 포함하면 정책의 해당 권한이 자격 증명에 적용됩니다. 인라인 정책은 자격 증명에 저장되므로 결과는 비슷하지만 연결되지 않고 포함됩니다.

Note

역할에 따라 달라지는 서비스에만 [서비스 연결 역할 \(p. 154\)](#)에 대한 인라인 정책을 포함할 수 있습니다. 서비스가 이 기능을 지원하는지 여부를 확인하려면 서비스에 대한 [AWS 설명서](#) 단원을 참조하십시오.

- **삭제** – 인라인 정책에 사용됩니다. 보안 주체 엔터티(사용자, 그룹, 역할)에서 인라인 정책을 삭제합니다. 정책을 삭제하면 보안 주체 개체에서 해당 권한이 제거됩니다.

Note

역할에 따른 서비스에서만 [서비스 연결 역할 \(p. 154\)](#)의 인라인 정책을 삭제할 수 있습니다. 서비스가 이 기능을 지원하는지 여부를 확인하려면 서비스에 대한 [AWS 설명서](#) 단원을 참조하십시오.

콘솔, AWS CLI 또는 AWS API를 사용하여 다음과 같은 작업을 수행할 수 있습니다.

추가 정보

- 관리형 정책과 인라인 정책의 차이점에 대한 자세한 정보는 [관리형 정책과 인라인 정책 \(p. 312\)](#) 단원을 참조하십시오.
- 권한 경계에 대한 자세한 정보는 [IAM 엔터티에 대한 권한 경계 \(p. 317\)](#) 단원을 참조하십시오.
- IAM 정책에 대한 일반적인 내용은 [정책 및 권한 \(p. 305\)](#) 단원을 참조하십시오.
- 정책 크기 제한에 대한 자세한 정보는 [IAM 개체 및 객체에 대한 제한 \(p. 485\)](#) 단원을 참조하십시오.

자격 증명 작업 보기

자격 증명(사용자, 그룹 또는 역할)에 대한 사용 권한을 변경하기 전에 최근 서비스 수준 활동을 검토해야 합니다. 이 기능은 사용 중인 보안 주체(사람 또는 애플리케이션)의 액세스 권한을 제거하지 않으려는 경우 중요합니다. 서비스에서 마지막으로 액세스한 데이터 보기에 대한 자세한 정보는 [서비스에서 마지막으로 액세스한 데이터를 사용하여 권한 줄이기 \(p. 409\)](#) 단원을 참조하십시오.

IAM 자격 증명 권한 추가(콘솔)

AWS Management 콘솔을 사용하여 자격 증명(사용자, 그룹 또는 역할)에 권한을 추가할 수 있습니다. 이렇게 하려면 권한을 제어하는 관리형 정책을 연결하거나 [권한 경계 \(p. 317\)](#) 역할을 하는 정책을 지정하십시오. 인라인 정책을 포함할 수도 있습니다.

자격 증명에 대한 권한 정책으로서 관리형 정책을 사용하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.
3. 정책 목록에서 연결할 정책 이름 옆의 확인란을 선택합니다. [Filter] 메뉴와 검색 상자를 사용하여 정책 목록을 필터링할 수 있습니다.
4. [Policy actions]를 선택한 후 [Attach]를 선택합니다.
5. 정책을 연결할 자격 증명을 하나 이상 선택합니다. [Filter] 메뉴와 검색 상자를 사용하면 보안 주체 개체 목록을 필터링할 수 있습니다. 자격 증명을 선택한 후 정책 연결을 선택합니다.

보안 경계(콘솔)를 설정하기 위해서 관리형 정책을 사용하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.
3. 정책 목록에서 설정할 정책 이름을 선택합니다. [Filter] 메뉴와 검색 상자를 사용하여 정책 목록을 필터링할 수 있습니다.
4. 정책 요약 페이지에서 Policy usage tab(정책 활용 탭)을 선택하고 필요하다면 Permissions boundaries(권한 경계) 섹션에서 Set boundary(경계 설정)을 선택합니다.
5. 권한 경계에 대한 정책이 사용될 하나 이상의 사용자 또는 역할을 선택하십시오. [Filter] 메뉴와 검색 상자를 사용하면 보안 주체 개체 목록을 필터링할 수 있습니다. 보안 주체를 선택한 후 Set boundaries(경계 설정)을 선택합니다.

사용자 또는 역할의 인라인 정책을 포함하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자 또는 역할을 선택합니다.
3. 목록에서 정책을 삽입할 그룹, 사용자 또는 역할 이름을 선택합니다.
4. Permissions 탭을 선택합니다.
5. 페이지의 하단으로 스크롤하고 Add inline policy(인라인 정책 추가)를 선택합니다.

Note

IAM에서 [service-linked role \(p. 154\)](#)에 인라인 정책을 포함시킬 수 없습니다. 링크된 서비스가 역할 권한을 수정할 수 있는지 여부를 결정하기 때문에 서비스 콘솔이나 API 또는 AWS CLI에서 정책을 추가할 수 있습니다. 서비스에 대한 서비스 연결 역할 설명서를 보려면 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하고, 해당 서비스의 Service-Linked Role(서비스 연결 역할) 열에서 예를 선택합니다.

6. 다음 방법 중에서 선택하여 정책을 생성하는 데 필요한 단계를 볼 수 있습니다.
 - [기존 관리형 정책 가져오기 \(p. 378\)](#) – 계정으로 관리형 정책을 가져온 다음 정책을 편집하여 특정 요구 사항에 맞게 사용자 지정할 수 있습니다. 관리형 정책은 사용자가 이전에 생성한 고객 관리형 정책이거나 AWS 관리형 정책일 수 있습니다.

- [시각적 편집기를 사용하여 정책 만들기 \(p. 379\)](#) – 시각적 편집기에서 정책을 새로 생성할 수 있습니다. 시각적 편집기를 사용할 경우 JSON 구문을 이해할 필요가 없습니다.
 - [JSON 탭에서 정책 만들기 \(p. 381\)](#) – JSON 탭에서 JSON 구문을 사용하여 정책을 생성할 수 있습니다. 새 JSON 정책 문서를 입력하거나 [예제 정책 \(p. 341\)](#)을 붙여 넣을 수 있습니다.
7. 인라인 정책을 생성하고 나면 이 정책이 사용자나 역할에 자동으로 포함됩니다.

그룹의 인라인 정책을 포함하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 [Groups]를 선택합니다.
3. 목록에서 정책을 삽입할 그룹 이름을 선택합니다.
4. 권한 탭을 선택하고 필요할 경우 Inline Policies(인라인 정책) 섹션을 확장합니다.
5. Create Group Policy(그룹 정책 생성)을 선택합니다. 그룹에 기존 정책이 없는 경우 여기를 클릭하십시오를 선택하여 첫 번째 인라인 정책을 만듭니다.
6. 정책 생성기 또는 사용자 지정 정책과 선택을 차례대로 선택합니다.
7. 다음 중 하나를 수행하십시오.
 - 사용자 지정 정책을 선택한 경우에는 정책 이름을 지정한 후 정책 문서를 생성합니다. [정책 검사기 \(p. 382\)](#)가 모든 구문 오류를 보고합니다.
 - 정책 생성기를 사용하여 정책을 생성한 경우에는 효과, AWS 서비스 및 작업 옵션을 선택합니다. Amazon 리소스 이름(ARN)(해당하는 경우)을 입력하고 포함하려는 조건을 추가합니다. 그런 다음 설명문 추가를 선택합니다. 문은 원하는 만큼 정책에 추가할 수 있습니다. 문 추가를 마치면 다음 단계를 선택합니다.
8. 정책에 아무런 문제가 없으면 [Apply Policy]를 선택합니다.

하나 이상의 엔터티에 대한 권한 경계 설정을 변경하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.
3. 정책 목록에서 설정할 정책 이름을 선택합니다. [Filter] 메뉴와 검색 상자를 사용하여 정책 목록을 필터링할 수 있습니다.
4. 정책 요약 페이지에서 Policy usage tab(정책 활용 탭)을 선택하고 필요하다면 Permissions boundaries(권한 경계) 섹션을 엽니다. 변경할 경계의 사용자 또는 역할 옆에 있는 확인란을 선택한 후 Change boundary(경계 변경)을 선택합니다.
5. 새로운 정책을 선택하여 권한 경계를 사용하십시오. [Filter] 메뉴와 검색 상자를 사용하여 정책 목록을 필터링할 수 있습니다. 정책을 선택한 후 Change boundary(경계 변경)을 선택합니다.

IAM 자격 증명 권한 제거(콘솔)

AWS Management 콘솔을 사용하여 자격 증명(사용자, 그룹 또는 역할)에서 권한을 제거할 수 있습니다. 이렇게 하려면 권한을 제어하는 관리형 정책을 분리하거나 [권한 경계 \(p. 317\)](#) 역할을 하는 정책을 제거하십시오. 인라인 정책을 삭제할 수도 있습니다.

권한 정책(콘솔)으로서 사용된 관리형 정책을 분리하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.

3. 정책 목록에서 분리할 정책 이름 옆의 확인란을 선택합니다. [Filter] 메뉴와 검색 상자를 사용하여 정책 목록을 필터링할 수 있습니다.
4. Policy actions(정책 작업)을 선택한 후 분리를 선택합니다.
5. 정책을 분리할 자격 증명을 선택합니다. 필터 메뉴와 검색 상자를 사용하여 자격 증명 목록을 필터링할 수 있습니다. 자격 증명을 선택한 후 Detach policy(정책 분리)를 선택합니다.

권한 경계(콘솔)를 제거하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.
3. 정책 목록에서 설정할 정책 이름을 선택합니다. [Filter] 메뉴와 검색 상자를 사용하여 정책 목록을 필터링할 수 있습니다.
4. 정책 요약 페이지에서 Policy usage tab(정책 활용 탭)을 선택하고 필요하다면 Permissions boundaries(권한 경계) 섹션에서 Remove boundary(경계 제거)를 선택합니다.
5. 제거를 선택하여 경계를 제거합니다.

인라인 정책을 삭제하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 그룹, 사용자 또는 역할을 선택합니다.
3. 목록에서 제거할 정책이 있는 그룹, 사용자 또는 역할 이름을 선택합니다.
4. Permissions 탭을 선택합니다. 그룹을 선택한 경우 필요에 따라 Inline Policies(인라인 정책) 섹션을 확장합니다.
5. 그룹에서는 Remove Policy(정책 제거)를 선택합니다. 사용자 또는 역할에서는 X를 선택합니다.

IAM 정책 추가(AWS CLI)

AWS CLI를 사용하여 자격 증명(사용자, 그룹 또는 역할)에 권한을 추가할 수 있습니다. 이렇게 하려면 권한을 제어하는 관리형 정책을 연결하거나 [권한 경계 \(p. 317\)](#) 역할을 하는 정책을 지정하십시오. 인라인 정책을 포함할 수도 있습니다.

엔터티에 대한 권한 정책으로서 관리형 정책을 사용하려면(AWS CLI)

1. (선택 사항) 관리형 정책에 대한 정보를 보려면 다음 명령을 실행합니다.
 - 관리형 정책의 목록 보기: `aws iam list-policies`
 - 관리형 정책에 대한 세부 정보 가져오기: `get-policy`
2. 관리형 정책을 자격 증명(사용자, 그룹 또는 역할)에 연결하려면 다음 명령 중 하나를 사용합니다.
 - `aws iam attach-user-policy`
 - `aws iam attach-group-policy`
 - `aws iam attach-role-policy`

보안 경계(AWS CLI)를 설정하기 위해서 관리형 정책을 사용하려면

1. (선택 사항) 관리형 정책에 대한 정보를 보려면 다음 명령을 실행합니다.
 - 관리형 정책의 목록 보기: `aws iam list-policies`

- 관리형 정책에 대한 세부 정보 가져오기: [aws iam get-policy](#)
- 2. 관리형 정책을 사용하여 엔터티(사용자 또는 역할)에 대한 권한 경계를 설정하려면 다음 명령 중 하나를 사용합니다.
 - [aws iam put-user-permissions-boundary](#)
 - [aws iam put-role-permissions-boundary](#)

인라인 정책을 포함시키려면(AWS CLI)

인라인 정책을 자격 증명(사용자, 그룹 또는 서비스 연결 역할 ([p. 154](#))이 아닌 역할)에 포함시키려면 다음 명령 중 하나를 사용합니다.

- [aws iam put-user-policy](#)
- [aws iam put-group-policy](#)
- [aws iam put-role-policy](#)

IAM 정책 제거(AWS CLI)

AWS CLI를 사용하여 권한을 제어하는 관리형 정책을 분리하거나 [권한 경계 \(p. 317\)](#) 역할을 하는 정책을 제거할 수 있습니다. 인라인 정책을 삭제할 수도 있습니다.

권한 정책(AWS CLI)으로서 사용된 관리형 정책을 분리하려면

1. (선택 사항) 정책에 대한 정보를 보려면 다음 명령을 실행합니다.
 - 관리형 정책의 목록 보기: [aws iam list-policies](#)
 - 관리형 정책에 대한 세부 정보 가져오기: [aws iam get-policy](#)
2. (선택 사항) 정책과 자격 증명 간의 관계에 대해 확인하려면 다음 명령을 실행합니다.
 - 관리형 정책이 연결된 자격 증명(사용자, 그룹 및 역할)의 목록을 보려면 다음과 같이 합니다.
 - [aws iam list-entities-for-policy](#)
 - 자격 증명(사용자, 그룹 또는 역할)에 연결된 관리형 정책의 목록을 보려면 다음 명령 중 하나를 사용합니다.
 - [aws iam list-attached-user-policies](#)
 - [aws iam list-attached-group-policies](#)
 - [aws iam list-attached-role-policies](#)
3. 관리형 정책을 자격 증명(사용자, 그룹 또는 역할)에서 분리하려면 다음 명령 중 하나를 사용합니다.
 - [aws iam detach-user-policy](#)
 - [aws iam detach-group-policy](#)
 - [aws iam detach-role-policy](#)

권한 경계(AWS CLI)를 제거하려면

1. (선택 사항) 현재 어떤 관리형 정책을 사용하여 사용자 또는 역할에 대한 권한 경계를 설정하는지 보려면 다음 명령을 실행하십시오.
 - [aws iam get-user](#)
 - [aws iam get-role](#)
2. (선택 사항) 현재 어떤 관리형 정책의 사용자 또는 역할이 권한 경계로 사용되는지 보려면 다음 명령을 실행하십시오.

- [aws iam list-entities-for-policy](#)
- 3. (선택 사항) 관리형 정책에 대한 정보를 보려면 다음 명령을 실행합니다.
 - 관리형 정책의 목록 보기: [aws iam list-policies](#)
 - 관리형 정책에 대한 세부 정보 가져오기: [aws iam get-policy](#)
- 4. 사용자 또는 역할에서 권한 경계를 제거하려면 다음 명령 중 하나를 사용합니다.
 - [aws iam delete-user-permissions-boundary](#)
 - [aws iam delete-role-permissions-boundary](#)

인라인 정책을 삭제하려면(AWS CLI)

1. (선택 사항) 자격 증명(사용자, 그룹 또는 역할)에 연결된 모든 인라인 정책의 목록을 보려면 다음 명령 중 하나를 사용합니다.
 - [aws iam list-user-policies](#)
 - [aws iam list-group-policies](#)
 - [aws iam list-role-policies](#)
2. (선택 사항) 자격 증명(사용자, 그룹 또는 역할)에 포함된 인라인 정책 문서를 가져오려면 다음 명령 중 하나를 사용합니다.
 - [aws iam get-user-policy](#)
 - [aws iam get-group-policy](#)
 - [aws iam get-role-policy](#)
3. 자격 증명(사용자, 그룹 또는 서비스 연결 역할 (p. 154)이 아닌 역할)에서 인라인 정책을 삭제하려면 다음 명령 중 하나를 사용합니다.
 - [aws iam delete-user-policy](#)
 - [aws iam delete-group-policy](#)
 - [aws iam delete-role-policy](#)

IAM 정책 추가(AWS API)

AWS API를 사용하여 권한을 제어하는 관리형 정책을 연결하거나 [권한 경계 \(p. 317\)](#) 역할을 하는 정책을 지정할 수 있습니다. 인라인 정책을 포함할 수도 있습니다.

엔터티에 대한 권한 정책으로서 관리형 정책을 사용하려면(AWS API)

1. (선택 사항) 정책에 대한 정보를 보려면 다음 작업을 호출합니다.
 - 관리형 정책의 목록 보기: [ListPolicies](#)
 - 관리형 정책에 대한 세부 정보 가져오기: [GetPolicy](#)
2. 관리형 정책을 자격 증명(사용자, 그룹 또는 역할)에 연결하려면 다음 작업 중 하나를 호출합니다.
 - [AttachUserPolicy](#)
 - [AttachGroupPolicy](#)
 - [AttachRolePolicy](#)

보안 경계(AWS API)를 설정하기 위해서 관리형 정책을 사용하려면

1. (선택 사항) 관리형 정책에 대한 정보를 보려면 다음 작업을 호출합니다.

- 관리형 정책의 목록 보기: [ListPolicies](#)
 - 관리형 정책에 대한 세부 정보 가져오기: [GetPolicy](#)
2. 관리형 정책을 사용하여 엔터티(사용자 또는 역할)에 대한 권한 경계를 설정하려면 다음 작업 중 하나를 호출합니다.
- [PutUserPermissionsBoundary](#)
 - [PutRolePermissionsBoundary](#)

인라인 정책을 포함시키려면(AWS API)

인라인 정책을 자격 증명(사용자, 그룹 또는 서비스 연결 역할 ([p. 154](#))이 아닌 역할)에 포함시키려면 다음 작업 중 하나를 호출합니다.

- [PutUserPolicy](#)
- [PutGroupPolicy](#)
- [PutRolePolicy](#)

IAM 정책 제거(AWS API)

AWS API를 사용하여 권한을 제어하는 관리형 정책을 분리하거나 [권한 경계 \(p. 317\)](#) 역할을 하는 정책을 제거할 수 있습니다. 인라인 정책을 삭제할 수도 있습니다.

권한 정책(AWS API)으로서 사용된 관리형 정책을 분리하려면

1. (선택 사항) 정책에 대한 정보를 보려면 다음 작업을 호출합니다.
 - 관리형 정책의 목록 보기: [ListPolicies](#)
 - 관리형 정책에 대한 세부 정보 가져오기: [GetPolicy](#)
2. (선택 사항) 정책과 자격 증명 간의 관계에 대해 확인하려면 다음 작업을 호출합니다.
 - 관리형 정책이 연결된 자격 증명(사용자, 그룹 및 역할)의 목록을 보려면 다음과 같이 합니다.
 - [ListEntitiesForPolicy](#)
 - 자격 증명(사용자, 그룹 또는 역할)에 연결된 관리형 정책의 목록을 보려면 다음 작업 중 하나를 호출합니다.
 - [ListAttachedUserPolicies](#)
 - [ListAttachedGroupPolicies](#)
 - [ListAttachedRolePolicies](#)
3. 관리형 정책을 자격 증명(사용자, 그룹 또는 역할)에서 분리하려면 다음 작업 중 하나를 호출합니다.
 - [DetachUserPolicy](#)
 - [DetachGroupPolicy](#)
 - [DetachRolePolicy](#)

권한 경계(AWS API)를 제거하려면

1. (선택 사항) 현재 어떤 관리형 정책을 사용하여 사용자 또는 역할에 대한 권한 경계를 설정하는지 보려면 다음 작업을 호출하십시오.
 - [GetUser](#)
 - [GetRole](#)

2. (선택 사항) 현재 어떤 관리형 정책의 사용자 또는 역할이 권한 경계로 사용되는지 보려면 다음 작업을 호출하십시오.
 - [ListEntitiesForPolicy](#)
3. (선택 사항) 관리형 정책에 대한 정보를 보려면 다음 작업을 호출합니다.
 - 관리형 정책의 목록 보기: [ListPolicies](#)
 - 관리형 정책에 대한 세부 정보 가져오기: [GetPolicy](#)
4. 사용자 또는 역할에서 권한 경계를 제거하려면 다음 작업 중 하나를 호출합니다.
 - [DeleteUserPermissionsBoundary](#)
 - [DeleteRolePermissionsBoundary](#)

인라인 정책을 삭제하려면(AWS API)

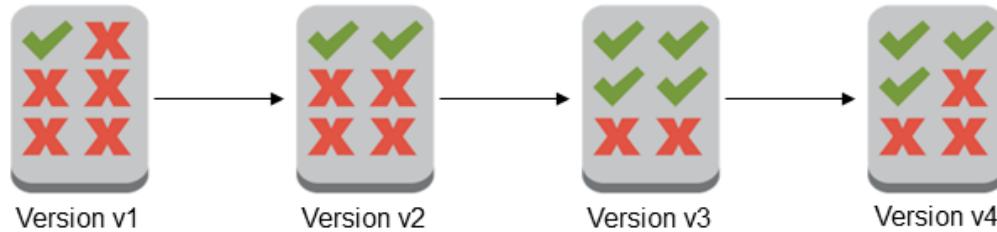
1. (선택 사항) 자격 증명(사용자, 그룹 또는 역할)에 연결된 모든 인라인 정책의 목록을 보려면 다음 작업 중 하나를 호출합니다.
 - [ListUserPolicies](#)
 - [ListGroupPolicies](#)
 - [ListRolePolicies](#)
2. (선택 사항) 자격 증명(사용자, 그룹 또는 역할)에 포함된 인라인 정책 문서를 가져오려면 다음 작업 중 하나를 호출합니다.
 - [GetUserPolicy](#)
 - [GetGroupPolicy](#)
 - [GetRolePolicy](#)
3. 인라인 정책을 자격 증명(사용자, 그룹 또는 [서비스 연결 역할](#) (p. 154)이 아닌 역할)에서 삭제하려면 다음 작업 중 하나를 호출합니다.
 - [DeleteUserPolicy](#)
 - [DeleteGroupPolicy](#)
 - [DeleteRolePolicy](#)

IAM 정책 버전 관리

IAM 고객 관리형 정책을 변경할 때, 그리고 AWS에서 AWS 관리형 정책을 변경할 때 변경된 정책은 기존 정책을 덮어쓰지 않습니다. 대신 IAM은 관리형 정책의 새 버전을 만듭니다. IAM은 고객 관리 정책을 최대 5개 버전까지 저장합니다. IAM은 인라인 정책의 버전 관리를 지원하지 않습니다.

다음은 고객 관리형 정책의 버전 관리를 나타낸 다이어그램입니다.

Multiple versions of a single managed policy



정책 버전은 Version 정책 요소와 다릅니다. Version 정책 요소는 정책 내에서 사용되며 정책 언어의 버전을 정의합니다. Version 정책 요소에 대한 자세한 내용은 [IAM JSON 정책 요소: Version \(p. 499\)](#)을 참조하십시오.

버전을 사용하여 관리형 정책에 대한 변경 사항을 추적할 수 있습니다. 예를 들어 관리형 정책을 변경한 다음 해당 변경 사항으로 인해 의도하지 않은 결과가 발생한 사실을 발견할 수 있습니다. 이 경우 이전 버전을 기본 버전으로 설정하여 관리형 정책의 이전 버전으로 뒤집을 수 있습니다.

다음 섹션에서는 관리형 정책에 버전 관리를 사용할 수 있는 방법에 대해 설명합니다.

주제

- [정책의 기본 버전을 설정할 수 있는 권한 \(p. 400\)](#)
- [고객 관리형 정책의 기본 버전 설정 \(p. 400\)](#)
- [버전을 사용하여 변경 사항 뒤집기 \(p. 402\)](#)
- [버전 제한 \(p. 402\)](#)

정책의 기본 버전을 설정할 수 있는 권한

정책의 기본 버전을 설정하는 데 필요한 권한은 작업에 대한 AWS API 작업에 해당합니다. `CreatePolicyVersion` 또는 `SetDefaultPolicyVersion` API 작업을 사용하여 정책의 기본 버전을 설정할 수 있습니다. 어떤 사람이 기존 정책의 기본 정책 버전을 설정할 수 있게 허용하려면 `iam:CreatePolicyVersion` 작업 또는 `iam:SetDefaultPolicyVersion` 작업에 대한 액세스 권한을 허용하면 됩니다. 그러면 `iam:CreatePolicyVersion` 작업을 이용해 새 버전의 정책을 생성하고 이 버전을 기본으로 설정할 수 있습니다. 또한 `iam:SetDefaultPolicyVersion` 작업을 통해서는 기존 버전의 정책을 기본으로 설정할 수 있습니다.

Important

사용자의 정책에서 `iam:SetDefaultPolicyVersion` 작업을 거부해도 사용자가 새 정책 버전을 생성하고 이 버전을 기본으로 설정하는 작업을 하지 못하게 할 수는 없습니다.

다음 정책을 사용하면 사용자가 기존 고객 관리형 정책을 변경하기 위해 액세스하는 것을 거부할 수 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Action": [  
                "iam:CreatePolicyVersion",  
                "iam:SetDefaultPolicyVersion"  
            ],  
            "Resource": "arn:aws:iam::*:policy/POLICY-NAME"  
        }  
    ]  
}
```

고객 관리형 정책의 기본 버전 설정

관리형 정책의 버전 중 하나가 기본 버전으로 설정됩니다. 정책의 기본 버전은 유효한 버전입니다. 즉, 기본 버전은 관리형 정책이 연결된 모든 보안 주체 엔터티(사용자, 그룹 및 역할)에 적용되는 버전입니다.

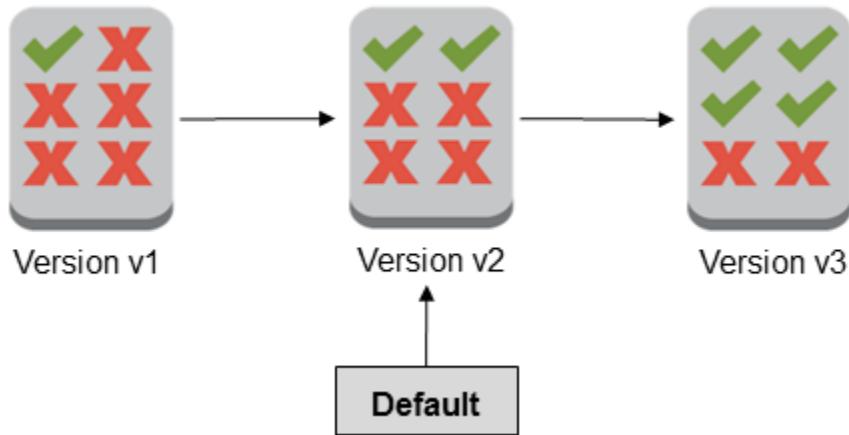
고객 관리형 정책을 만들 때 정책은 v1로 식별되는 단일 버전으로 시작합니다. 버전이 하나뿐인 관리형 정책의 경우 해당 버전이 기본값으로 자동 설정됩니다. 버전이 둘 이상인 고객 관리형 정책의 경우에는 기본값으

로 설정할 버전을 선택해야 합니다. AWS 관리형 정책의 경우 기본 버전은 AWS에서 설정됩니다. 다음 다이어그램에서는 이 개념을 보여 줍니다.

Managed policy with one version



Managed policy with multiple versions



고객 관리형 정책의 기본 버전이 정책이 연결되는 모든 보안 주체 개체(사용자, 그룹 및 역할)에 적용되도록 해당 버전을 설정할 수 있습니다. 단, AWS 관리형 정책 또는 인라인 정책에는 기본 버전을 설정할 수 없습니다.

고객 관리형 정책의 기본 버전을 설정하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.
3. 정책 목록에서 기본 버전을 설정할 정책 이름을 선택합니다. [Filter] 메뉴와 검색 상자를 사용하여 정책 목록을 필터링할 수 있습니다.
4. Policy versions(정책 버전) 탭을 선택합니다. 기본 버전으로 설정할 버전 옆의 확인란을 선택한 후 기본 값으로 설정을 선택합니다.

AWS Command Line Interface 또는 AWS API에서 고객 관리형 정책을 기본 버전으로 설정하는 방법을 알아보려면 [고객 관리형 정책 편집\(AWS CLI\) \(p. 405\)](#) 단원을 참조하십시오.

버전을 사용하여 변경 사항 를 백

변경 사항을 를 백하도록 고객 관리형 정책의 기본 버전을 설정할 수 있습니다. 예를 들어 다음 시나리오를 고려해 보십시오:

사용자가 AWS Management 콘솔을 사용하여 특정 Amazon S3 버킷을 관리할 수 있도록 허용하는 고객 관리형 정책을 만듭니다. 생성 시 고객 관리형 정책의 버전은 v1로 식별되는 한 버전뿐이어서 이 버전이 기본값으로 자동 설정됩니다. 정책이 의도대로 적용됩니다.

나중에 두 번째 Amazon S3 버킷을 관리하기 위한 권한을 추가하기 위해 정책을 업데이트합니다. IAM에서 변경 사항을 포함하고 v2로 식별되는 정책의 새 버전을 만듭니다. v2 버전을 기본값으로 설정하고 얼마 지나지 않아 사용자들이 Amazon S3 콘솔을 사용할 수 있는 권한이 없다고 보고합니다. 이 경우 의도대로 적용되는 정책의 v1 버전으로 를 백할 수 있습니다. 이렇게 하기 위해 v1 버전을 기본 버전으로 설정합니다. 이제 사용자들이 Amazon S3 콘솔을 사용하여 원래 버킷을 관리할 수 있습니다.

나중에 정책의 v2 버전에 있는 오류를 해결한 후 두 번째 Amazon S3 버킷을 관리하기 위한 권한을 추가하기 위해 다시 정책을 업데이트합니다. IAM에서 v3으로 식별되는 정책의 새 버전을 하나 더 만듭니다. v3 버전을 기본값으로 설정합니다. 이 버전이 의도대로 적용됩니다. 이 시점에서 정책의 v2 버전을 삭제합니다.

버전 제한

관리형 정책에는 최대 5개의 버전이 있을 수 있습니다. 5개 버전을 만든 후에도 관리형 정책을 변경해야 할 경우 AWS Command Line Interface 또는 AWS API에서 먼저 기존 버전을 하나 이상 삭제해야 합니다. AWS Management 콘솔을 사용할 경우에는 정책을 편집하기 전에 버전을 삭제할 필요가 없습니다. 6번째 버전을 저장할 경우 정책의 기본 버전이 아닌 버전을 한 개 이상 삭제하라는 메시지가 표시된 대화 상자가 나타납니다. 결정을 위해 각 버전의 JSON 정책 문서를 볼 수 있습니다. 이 대화 상자에 대한 자세한 내용은 [the section called "IAM 정책 편집" \(p. 402\)](#) 단원을 참조하십시오.

기본 버전을 제외하고 원하는 모든 관리형 정책 버전을 삭제할 수 있습니다. 버전을 삭제할 때 나머지 버전의 버전 식별자는 변경되지 않습니다. 따라서 버전 식별자가 순차적이지 않을 수 있습니다. 예를 들어 관리형 정책의 v2 및 v4 버전을 삭제하고 새 버전을 2개 추가하면 나머지 버전 식별자가 v1, v3, v5, v6 및 v7이 될 수 있습니다.

IAM 정책 편집

정책 ([p. 305](#))은 자격 증명 또는 리소스에 연결될 때 해당 권한을 정의하는 개체입니다. 정책은 JSON 문서로 AWS에 저장되며 IAM에서 자격 증명 기반 정책으로 보안 주체에 연결됩니다. 자격 증명 기반 정책을 IAM 그룹, 사용자 또는 역할과 같은 보안 주체(또는 자격 증명)에 연결할 수 있습니다. 자격 증명 기반 정책에는 AWS 관리형 정책, 고객 관리형 정책 및 [인라인 정책 \(p. 312\)](#)이 포함됩니다. IAM에서 고객 관리형 정책 및 인라인 정책을 편집할 수 있습니다. AWS 관리형 정책은 편집할 수 없습니다. 정책 크기 제한 및 기타 할당량에 대한 자세한 내용은 [IAM 개체 및 객체에 대한 제한 \(p. 485\)](#) 단원을 참조하십시오.

주제

- [정책 액세스 보기 \(p. 402\)](#)
- [고객 관리형 정책 편집\(콘솔\) \(p. 403\)](#)
- [인라인 정책 편집\(콘솔\) \(p. 404\)](#)
- [고객 관리형 정책 편집\(AWS CLI\) \(p. 405\)](#)
- [고객 관리형 정책 편집\(AWS API\) \(p. 405\)](#)

정책 액세스 보기

정책에 대한 권한을 변경하기 전에 최근 서비스 수준 활동을 검토해야 합니다. 이 기능은 사용 중인 보안 주체(사람 또는 애플리케이션)의 액세스 권한을 제거하지 않으려는 경우 중요합니다. 서비스에서 마지막으로

액세스한 데이터 보기에 대한 자세한 내용은 [서비스에서 마지막으로 액세스한 데이터를 사용하여 권한 줄이기 \(p. 409\)](#) 단원을 참조하십시오.

고객 관리형 정책 편집(콘솔)

고객 관리형 정책을 편집하여 정책에 정의된 권한을 변경할 수 있습니다. 고객 관리형 정책에 최대 5개의 버전을 사용할 수 있습니다. 이는 관리형 정책을 변경하여 버전이 5개 넘게 생성될 경우 AWS Management 콘솔에서 어느 버전을 삭제할 것이지 결정하라는 메시지가 표시되므로 중요합니다. 메시지가 표시되지 않도록 편집하기 전에 기본 버전을 변경하거나 정책 버전을 삭제할 수도 있습니다. 버전에 대한 자세한 내용은 [IAM 정책 버전 관리 \(p. 399\)](#) 단원을 참조하십시오.

고객 관리형 정책을 편집하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.
3. 정책 목록에서 편집할 정책 이름을 선택합니다. [Filter] 메뉴와 검색 상자를 사용하여 정책 목록을 필터링할 수 있습니다.
4. 권한 탭을 선택한 다음 정책 편집을 선택합니다.
5. 다음 중 하나를 수행하십시오.
 - Visual editor(시각적 편집기) 탭을 선택하면 JSON 구문을 이해하지 않아도 정책을 변경할 수 있습니다. 정책의 각 권한 블록에 대한 서비스, 작업, 리소스 또는 조건(선택 사항)을 변경할 수 있습니다. 정책을 가져와 추가 권한을 정책 하단에 추가할 수도 있습니다. 변경이 완료되면 정책 검토를 선택하여 계속 진행합니다.
 - JSON 탭을 선택하고 JSON 텍스트 상자에 입력하거나 붙여 넣어 정책을 수정합니다. 정책을 가져와 추가 권한을 정책 하단에 추가할 수도 있습니다. 변경이 완료되면 정책 검토를 선택하여 계속 진행합니다. [정책 검사기 \(p. 382\)](#)가 모든 구문 오류를 보고합니다.

Note

언제든지 Visual editor(시각적 편집기) 및 JSON 탭을 전환할 수 있습니다. 그러나 변경을 수행하거나 Visual editor(시각적 편집기) 탭에서 정책 검토를 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 내용은 [정책 재구성 \(p. 455\)](#) 단원을 참조하십시오.

6. 검토 페이지에서 정책 요약을 검토하고 나서 변경 사항 저장을 선택하여 작업을 저장합니다.
7. 관리형 정책 버전이 이미 최댓값인 5개가 있을 경우 저장을 선택하면 대화 상자가 나타납니다. 새 버전을 저장하려면 이전 버전을 한 개 이상 삭제해야 합니다. 기본 버전은 삭제할 수 없습니다. 다음 옵션 중 하나를 선택합니다.
 - Remove oldest non-default policy version(version v# - created # days ago)(기본 정책을 제외하고 가장 오래된 정책 버전 제거(버전 v# - #일 전에 생성됨)) – 어느 버전이 삭제될 것이고 언제 삭제되었는지 보려면 이 옵션을 사용합니다. 두 번째 옵션인 Select versions to remove(제거할 버전 선택)을 선택하면 기본 버전 외 다른 버전 모두에 대한 JSON 정책 문서를 볼 수 있습니다.
 - Select versions to remove(제거할 버전 선택) – JSON 정책 문서를 보고 한 개 이상을 선택하여 삭제하려면 이 옵션을 사용합니다.

삭제할 버전을 선택한 후 Delete version and save(버전 삭제 및 저장)을 선택하여 새 정책 버전을 저장합니다.

고객 관리형 정책의 기본 버전을 설정하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.

3. 정책 목록에서 기본 버전을 설정할 정책 이름을 선택합니다. [Filter] 메뉴와 검색 상자를 사용하여 정책 목록을 필터링할 수 있습니다.
4. Policy versions(정책 버전) 탭을 선택합니다. 기본 버전으로 설정할 버전 옆의 확인란을 선택한 후 기본 값으로 설정을 선택합니다.

고객 관리형 정책의 버전을 삭제하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.
3. 버전을 삭제하려는 고객 관리형 정책의 이름을 선택합니다. [Filter] 메뉴와 검색 상자를 사용하여 정책 목록을 필터링할 수 있습니다.
4. Policy versions(정책 버전) 탭을 선택합니다. 삭제하려는 버전 옆의 확인란을 선택합니다. 그런 다음 [Delete]를 선택합니다.
5. 버전을 정말로 삭제할 것인지 다시 한 번 묻는 메시지가 나오면 확인 후 삭제를 선택합니다.

인라인 정책 편집(콘솔)

AWS Management 콘솔에서 인라인 정책을 편집할 수 있습니다.

그룹, 사용자 또는 역할의 인라인 정책을 편집하려면(콘솔)

1. 탐색 창에서 사용자 또는 역할을 선택합니다.
2. 정책을 변경하려는 사용자 또는 역할 이름을 선택합니다. 그런 다음 권한 탭을 선택하고 정책을 확장합니다.
3. 인라인 정책을 편집하려면 정책 편집을 선택합니다.
4. 다음 중 하나를 수행하십시오.
 - Visual editor(시각적 편집기) 탭을 선택하면 JSON 구문을 이해하지 않아도 정책을 변경할 수 있습니다. 정책의 각 권한 블록에 대한 서비스, 작업, 리소스 또는 조건(선택 사항)을 변경할 수 있습니다. 정책을 가져와 추가 권한을 정책 하단에 추가할 수도 있습니다. 변경이 완료되면 정책 검토를 선택하여 계속 진행합니다.
 - JSON 탭을 선택하고 JSON 텍스트 상자에 입력하거나 붙여 넣어 정책을 수정합니다. 정책을 가져와 추가 권한을 정책 하단에 추가할 수도 있습니다. 변경이 완료되면 정책 검토를 선택하여 계속 진행합니다. [정책 검사기 \(p. 382\)](#)가 모든 구문 오류를 보고합니다. 현재 추가된 주체에 아무런 영향을 주지 않고 변경 사항만 저장하려면 Save as default version(기본 버전으로 저장) 확인란의 선택을 해제합니다.

Note

언제든지 Visual editor(시각적 편집기) 및 JSON 탭을 전환할 수 있습니다. 그러나 변경을 수행하거나 Visual editor(시각적 편집기) 탭에서 정책 검토를 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 내용은 [정책 재구성 \(p. 455\)](#) 단원을 참조하십시오.

5. 검토 페이지에서 정책 요약을 검토하고 나서 변경 사항 저장을 선택하여 작업을 저장합니다.

그룹의 인라인 정책을 편집하려면

1. 탐색 창에서 [Groups]를 선택합니다.
2. 정책을 변경하려는 그룹 이름을 선택합니다. 그런 다음 권한 탭을 선택합니다.
3. 인라인 정책을 편집하려면 정책 편집을 선택합니다.

4. JSON 정책을 수정한 후 저장을 선택하여 변경 사항을 저장합니다.

고객 관리형 정책 편집(AWS CLI)

AWS Command Line Interface에서 고객 관리형 정책을 편집할 수 있습니다(AWS CLI).

Note

관리형 정책에는 최대 5개의 버전이 있을 수 있습니다. 5개 버전을 만든 후에도 고객 관리형 정책을 변경해야 할 경우 먼저 기존 버전을 1개 이상 삭제해야 합니다.

고객 관리형 정책을 편집하려면(AWS CLI)

1. (선택 사항) 정책에 대한 정보를 보려면 다음 명령을 실행합니다.
 - 관리형 정책의 목록 보기: [list-policies](#)
 - 관리형 정책에 대한 세부 정보 가져오기: [get-policy](#)
2. (선택 사항) 정책과 자격 증명 간의 관계에 대해 확인하려면 다음 명령을 실행합니다.
 - 관리형 정책이 연결된 자격 증명(사용자, 그룹 및 역할)의 목록을 보려면 다음과 같이 합니다.
 - [list-entities-for-policy](#)
 - 자격 증명(사용자, 그룹 또는 역할)에 연결된 관리형 정책의 목록을 보려면
 - [list-attached-user-policies](#)
 - [list-attached-group-policies](#)
 - [list-attached-role-policies](#)
3. 고객 관리형 정책을 편집하려면 다음 명령을 실행합니다.
 - [create-policy-version](#)

고객 관리형 정책의 기본 버전을 설정하려면(AWS CLI)

1. (선택 사항) 관리형 정책 목록을 보려면 다음 명령을 실행합니다.
 - [list-policies](#)
2. 고객 관리형 정책의 기본 버전을 설정하려면 다음 명령을 실행합니다.
 - [set-default-policy-version](#)

고객 관리형 정책의 한 버전을 삭제하려면(AWS CLI)

1. (선택 사항) 관리형 정책 목록을 보려면 다음 명령을 실행합니다.
 - [list-policies](#)
2. 고객 관리형 정책을 삭제하려면 다음 명령을 실행합니다.
 - [delete-policy-version](#)

고객 관리형 정책 편집(AWS API)

AWS API를 사용하여 고객 관리형 정책을 편집할 수 있습니다.

Note

관리형 정책에는 최대 5개의 버전이 있을 수 있습니다. 5개 버전을 만든 후에도 고객 관리형 정책을 변경해야 할 경우 먼저 기존 버전을 1개 이상 삭제해야 합니다.

고객 관리형 정책을 편집하려면(AWS API)

1. (선택 사항) 정책에 대한 정보를 보려면 다음 작업을 호출합니다.
 - 관리형 정책의 목록 보기: [ListPolicies](#)
 - 관리형 정책에 대한 세부 정보 가져오기: [GetPolicy](#)
2. (선택 사항) 정책과 자격 증명 간의 관계에 대해 확인하려면 다음 작업을 호출합니다.
 - 관리형 정책이 연결된 자격 증명(사용자, 그룹 및 역할)의 목록을 보려면 다음과 같이 합니다.
 - [ListEntitiesForPolicy](#)
 - 자격 증명(사용자, 그룹 또는 역할)에 연결된 관리형 정책의 목록을 보려면
 - [ListAttachedUserPolicies](#)
 - [ListAttachedGroupPolicies](#)
 - [ListAttachedRolePolicies](#)
3. 고객 관리형 정책을 편집하려면 다음 작업을 호출합니다.
 - [CreatePolicyVersion](#)

고객 관리형 정책의 기본 버전을 설정하려면(AWS API)

1. (선택 사항) 관리형 정책 목록을 보려면 다음 작업을 호출합니다.
 - [ListPolicies](#)
2. 고객 관리형 정책의 기본 버전을 설정하려면 다음 작업을 호출합니다.
 - [SetDefaultPolicyVersion](#)

고객 관리형 정책의 한 버전을 삭제하려면(AWS API)

1. (선택 사항) 관리형 정책 목록을 보려면 다음 작업을 호출합니다.
 - [ListPolicies](#)
2. 고객 관리형 정책을 삭제하려면 다음 작업을 호출합니다.
 - [DeletePolicyVersion](#)

IAM 정책 삭제

AWS Management 콘솔, AWS Command Line Interface(AWS CLI) 또는 IAM API를 사용하여 IAM 정책을 삭제할 수 있습니다.

관리형 정책과 인라인 정책의 차이점에 대한 자세한 내용은 [관리형 정책과 인라인 정책 \(p. 312\)](#) 단원을 참조하십시오.

IAM 정책에 대한 일반적인 내용은 [정책 및 권한 \(p. 305\)](#) 단원을 참조하십시오.

정책 크기 제한 및 기타 할당량에 대한 자세한 내용은 [IAM 개체 및 객체에 대한 제한 \(p. 485\)](#) 단원을 참조하십시오.

주제

- [정책 액세스 보기 \(p. 407\)](#)
- [IAM 정책 삭제\(콘솔\) \(p. 407\)](#)
- [IAM 정책 삭제\(AWS CLI\) \(p. 407\)](#)

- IAM 정책 삭제(AWS API) (p. 408)

정책 액세스 보기

정책을 삭제하기 전에 최근 서비스 수준 활동을 검토해야 합니다. 이 기능은 사용 중인 보안 주체(사람 또는 애플리케이션)의 액세스 권한을 제거하지 않으려는 경우 중요합니다. 서비스에서 마지막으로 액세스한 데이터 보기에 대한 자세한 내용은 [서비스에서 마지막으로 액세스한 데이터를 사용하여 권한 줄이기](#) (p. 409) 단원을 참조하십시오.

IAM 정책 삭제(콘솔)

고객 관리형 정책은 삭제하여 AWS 계정에서 제거할 수 있습니다. 단, AWS 관리형 정책은 삭제할 수 없습니다.

고객 관리형 정책을 삭제하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.
3. 삭제할 고객 관리형 정책 옆의 확인란을 선택합니다. [Filter] 메뉴와 검색 상자를 사용하여 정책 목록을 필터링할 수 있습니다.
4. Policy actions(정책 작업)을 선택한 후 삭제를 선택합니다.
5. 정책을 정말로 삭제할 것인지 다시 한 번 묻는 메시지가 나오면 확인 후 삭제를 선택합니다.

그룹, 사용자 또는 역할의 인라인 정책을 삭제하려면(콘솔)

1. 탐색 창에서 그룹, 사용자 또는 역할을 선택합니다.
2. 정책을 삭제하려는 그룹, 사용자 또는 역할 이름을 선택합니다. 그런 다음 권한 탭을 선택합니다. 사용자 또는 역할을 선택한 경우 정책을 확장합니다.
3. 그룹에서 인라인 정책을 삭제하려면 Remove Policy(정책 제거)를 선택합니다. 사용자 또는 역할에서 인라인 정책을 삭제하려면 X를 선택합니다.

IAM 정책 삭제(AWS CLI)

AWS Command Line Interface에서 고객 관리형 정책을 삭제할 수 있습니다.

고객 관리형 정책을 삭제하려면(AWS CLI)

1. (선택 사항) 정책에 대한 정보를 보려면 다음 명령을 실행합니다.
 - 관리형 정책의 목록 보기: `list-policies`
 - 관리형 정책에 대한 세부 정보 가져오기: `get-policy`
2. (선택 사항) 정책과 자격 증명 간의 관계에 대해 확인하려면 다음 명령을 실행합니다.
 - 관리형 정책이 연결된 자격 증명(사용자, 그룹 및 역할)의 목록을 보려면 다음 명령을 실행합니다.
 - `list-entities-for-policy`
 - 자격 증명(사용자, 그룹 또는 역할)에 연결된 관리형 정책의 목록을 보려면 다음 명령 중 하나를 실행합니다.
 - `list-attached-user-policies`
 - `list-attached-group-policies`
 - `list-attached-role-policies`

3. 고객 관리형 정책을 삭제하려면 다음 명령을 실행합니다.

- [delete-policy](#)

인라인 정책을 삭제하려면(AWS CLI)

1. (선택 사항) 자격 증명(사용자, 그룹 또는 역할)에 연결된 모든 인라인 정책의 목록을 보려면 다음 명령 중 하나를 사용합니다.
 - [aws iam list-user-policies](#)
 - [aws iam list-group-policies](#)
 - [aws iam list-role-policies](#)
2. (선택 사항) 자격 증명(사용자, 그룹 또는 역할)에 포함된 인라인 정책 문서를 가져오려면 다음 명령 중 하나를 사용합니다.
 - [aws iam get-user-policy](#)
 - [aws iam get-group-policy](#)
 - [aws iam get-role-policy](#)
3. 자격 증명(사용자, 그룹 또는 [서비스 연결 역할 \(p. 154\)](#))이 아닌 역할에서 인라인 정책을 삭제하려면 다음 명령 중 하나를 사용합니다.
 - [aws iam delete-user-policy](#)
 - [aws iam delete-group-policy](#)
 - [aws iam delete-role-policy](#)

IAM 정책 삭제(AWS API)

AWS API를 사용하여 고객 관리형 정책을 삭제할 수 있습니다.

고객 관리형 정책을 삭제하려면(AWS API)

1. (선택 사항) 정책에 대한 정보를 보려면 다음 작업을 호출합니다.
 - 관리형 정책의 목록 보기: [ListPolicies](#)
 - 관리형 정책에 대한 세부 정보 가져오기: [GetPolicy](#)
2. (선택 사항) 정책과 자격 증명 간의 관계에 대해 확인하려면 다음 작업을 호출합니다.
 - 관리형 정책이 연결된 자격 증명(사용자, 그룹 및 역할)의 목록을 보려면 다음 작업을 호출합니다.
 - [ListEntitiesForPolicy](#)
 - 자격 증명(사용자, 그룹 또는 역할)에 연결된 관리형 정책의 목록을 보려면 다음 작업 중 하나를 호출합니다.
 - [ListAttachedUserPolicies](#)
 - [ListAttachedGroupPolicies](#)
 - [ListAttachedRolePolicies](#)
3. 고객 관리형 정책을 삭제하려면 다음 작업을 호출합니다.
 - [DeletePolicy](#)

인라인 정책을 삭제하려면(AWS API)

1. (선택 사항) 자격 증명(사용자, 그룹 또는 역할)에 연결된 모든 인라인 정책의 목록을 보려면 다음 작업 중 하나를 호출합니다.

- [ListUserPolicies](#)
 - [ListGroupPolicies](#)
 - [ListRolePolicies](#)
2. (선택 사항) 자격 증명(사용자, 그룹 또는 역할)에 포함된 인라인 정책 문서를 가져오려면 다음 작업 중 하나를 호출합니다.
- [GetUserPolicy](#)
 - [GetGroupPolicy](#)
 - [GetRolePolicy](#)
3. 인라인 정책을 자격 증명(사용자, 그룹 또는 [서비스 연결 역할 \(p. 154\)](#)이 아닌 역할)에서 삭제하려면 다음 작업 중 하나를 호출합니다.
- [DeleteUserPolicy](#)
 - [DeleteGroupPolicy](#)
 - [DeleteRolePolicy](#)

서비스에서 마지막으로 액세스한 데이터를 사용하여 권한 줄이기

IAM 엔터티(사용자 또는 역할)가 서비스에 액세스하려고 시도한 마지막 시간에 대한 보고서를 볼 수 있습니다. 이를 서비스에서 마지막으로 액세스한 데이터라고 합니다. 그런 다음 이 정보를 사용하여 엔터티가 사용하는 서비스에만 액세스할 수 있도록 정책을 구체화할 수 있습니다. IAM에서 각 리소스 유형에 대한 보고서를 생성할 수 있습니다. 각각의 경우 보고서는 지정된 보고 기간 동안 허용된 서비스를 포함합니다.

- 사용자 – 사용자가 서비스에 액세스하려고 시도한 마지막 시간을 표시합니다.
- 그룹 – 그룹 멤버가 서비스에 액세스하려고 시도한 마지막 시간에 대한 정보를 표시합니다. 또한 이 보고서에는 액세스를 시도한 총 멤버 수가 포함됩니다.
- 역할 – 해당 역할이 서비스에 액세스하려고 시도한 마지막 시간을 표시합니다.
- 정책 – 사용자 또는 역할이 서비스에 액세스하려고 시도한 마지막 시간에 대한 정보를 표시합니다. 또한 이 보고서에는 액세스를 시도한 총 엔터티 수가 포함됩니다.

서비스에서 마지막으로 액세스한 데이터를 사용하여 관련 정책에서 한 번도 사용되지 않거나 최근에 사용된 적이 없는 권한을 식별할 수 있습니다. 그러면 사용되지 않은 서비스에 대한 권한을 제거하거나 사용 패턴이 유사한 사용자들을 그룹으로 재구성할 수 있습니다. 이렇게 하면 계정 보안을 향상 시킬 수 있습니다. 엔터티가 권한을 행사했는지 여부와 마지막으로 권한을 행사한 시기를 알면 불필요한 권한을 제거하고 IAM 정책을 좀 더 손쉽게 강화할 수 있습니다.

AWS Management 콘솔, AWS CLI, 또는 AWS API를 사용하여 서비스에서 마지막으로 액세스한 데이터를 보려면 [서비스에서 마지막으로 액세스한 데이터 보기 \(p. 412\)](#) 단원을 참조하십시오.

서비스에서 마지막으로 액세스한 데이터를 사용하여 IAM 엔터티에 부여할 권한을 결정하는 데 대한 예제 시나리오는 [액세스 데이터 사용에 대한 예제 시나리오 \(p. 414\)](#) 단원을 참조하십시오.

알아야 할 것들

서비스에서 마지막으로 액세스한 보고서의 데이터를 사용하여 엔터티의 권한을 변경하려면 먼저 데이터에 대한 다음 세부 정보를 검토하십시오.

- 보고 기간 – 최근 활동은 일반적으로 4시간 이내에 나타납니다. IAM은 지난 365일 동안의 활동을 보고합니다. 단, 해당 지역에서 이 기능을 지원한 지 1년 미만인 경우 더 적을 수 있습니다. 자세한 정보는 [데이터가 추적되는 리전 \(p. 411\)](#) 단원을 참조하십시오.

- 인증된 엔터티 – 보고서에는 계정의 인증된 엔터티(사용자 또는 역할)에 대한 데이터만 포함됩니다. 인증되지 않은 시도에 대한 데이터는 보고서에 포함되지 않습니다. 다른 계정에서 시도한 데이터도 포함되지 않습니다.
- 정책 유형 – 보고서에는 엔터티 정책에서 허용하는 서비스에 대한 데이터만 포함됩니다. 이러한 정책은 역할에 연결되거나 사용자에게 직접 또는 그룹을 통해 연결됩니다. 다른 정책 유형에서 허용하는 액세스는 보고서에 포함되어 있지 않습니다. 제외된 정책 유형에는 리소스 기반 정책, 액세스 제어 목록, AWS Organizations 정책, IAM 권한 경계 및 세션 정책이 있습니다. 다른 정책 유형이 액세스를 허용하거나 거부하는 방법을 알아보려면 [정책 평가 로직 \(p. 531\)](#) 단원을 참조하십시오.

주제

- [필요한 권한 \(p. 410\)](#)
- [엔터티 작업 문제 해결 \(p. 411\)](#)
- [데이터가 추적되는 리전 \(p. 411\)](#)
- [서비스에서 마지막으로 액세스한 데이터 보기 \(p. 412\)](#)
- [액세스 데이터 사용에 대한 예제 시나리오 \(p. 414\)](#)

필요한 권한

AWS Management 콘솔을 사용하여 서비스에서 마지막으로 액세스한 데이터를 확인하려면 다음 작업을 포함하는 정책이 있어야 합니다.

- `iam:GenerateServiceLastAccessedDetails`
- `iam:Get*`
- `iam>List*`

Note

이러한 권한을 사용하면 사용자가 다음을 확인할 수 있습니다.

- [관리형 정책](#)에 연결된 사용자, 그룹 또는 역할
- 사용자 또는 역할이 액세스할 수 있는 서비스
- 서비스에 마지막으로 액세스한 시간

AWS CLI 또는 AWS API를 사용하여 서비스에서 마지막으로 액세스한 데이터를 보려면 사용하려는 작업과 일치하는 권한이 있어야 합니다.

- `iam:GenerateServiceLastAccessedDetails`
- `iam:GetServiceLastAccessedDetails`
- `iam:GetServiceLastAccessedDetailsWithEntities`
- `iam>ListPoliciesGrantingServiceAccess`

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다. 사용하면 서비스에서 마지막으로 액세스한 데이터를 볼 수 있으며 모든 IAM에 대한 읽기 전용 액세스를 허용합니다. 이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": [  
            "iam:GenerateServiceLastAccessedDetails",  
            "iam:GetServiceLastAccessedDetails",  
            "iam:GetServiceLastAccessedDetailsWithEntities"  
        ]  
    }  
}
```

```
    "iam:Get*",
    "iam>List*"
],
"Resource": "*"
}
```

엔터티 작업 문제 해결

AWS Management 콘솔 서비스에서 마지막으로 액세스한 데이터 테이블이 비어 있거나 AWS CLI 또는 AWS API 요청이 빈 데이터 세트 또는 null 필드를 반환하는 경우 다음 예제를 검토하십시오.

- 사용자의 경우 사용자가 직접 또는 그룹 멤버십을 통해 인라인 또는 관리형 정책이 하나 이상 연결되어 있는지 확인합니다.
- 그룹의 경우 그룹에 인라인 또는 관리형 정책이 하나 이상 연결되어 있는지 확인합니다.
- 그룹의 경우 보고서는 그룹 정책을 사용하여 서비스에 액세스한 멤버에 대해서만 서비스에서 마지막으로 액세스한 데이터를 반환합니다. 멤버가 다른 정책을 사용했는지 여부를 확인하려면 해당 사용자에 대해 서비스에서 마지막으로 액세스한 데이터를 검토하십시오.
- 역할의 경우 역할에 인라인 또는 관리형 정책이 하나 이상 연결되어 있는지 확인합니다.
- 엔터티(사용자 또는 역할)의 경우 해당 엔터티의 사용 권한에 영향을 줄 수 있는 다른 정책 유형을 검토합니다. 여기에는 리소스 기반 정책, 액세스 제어 목록, AWS Organizations 정책, IAM 권한 경계 또는 세션 정책이 있습니다. 자세한 내용은 [정책 유형 \(p. 305\)](#) 또는 [단일 계정 내에서 정책 평가 \(p. 532\)](#) 단원을 참조하십시오.
- 정책의 경우 지정된 관리형 정책이 하나 이상의 사용자, 멤버가 있는 그룹 또는 역할에 연결되어 있는지 확인합니다.

변경을 수행할 때 보고서에 작업이 나타날 때까지 최소 4시간을 기다리십시오. AWS CLI 또는 AWS API를 사용하는 경우 업데이트된 데이터를 표시하려면 새 보고서를 생성해야 합니다.

데이터가 추적되는 리전

AWS는 대부분의 지역에서 서비스에서 마지막으로 액세스한 데이터를 수집합니다. 데이터는 최대 365일 동안 저장됩니다. AWS에서 리전을 추가하면 AWS에서 각 리전의 데이터 추적을 시작한 날짜와 함께 해당 리전이 다음 표에 추가됩니다.

리전 이름	리전	추적 시작 날짜
미국 동부(오하이오)	us-east-2	2017년 10월 27일
미국 동부(버지니아 북부)	us-east-1	2015년 10월 1일
미국 서부(캘리포니아 북부 지역)	us-west-1	2015년 10월 1일
미국 서부(오레곤)	us-west-2	2015년 10월 1일
아시아 태평양(도쿄)	ap-northeast-1	2015년 10월 1일
아시아 태평양(서울)	ap-northeast-2	2016년 1월 6일
아시아 태평양(싱가포르)	ap-southeast-1	2015년 10월 1일
아시아 태평양(시드니)	ap-southeast-2	2015년 10월 1일
아시아 태평양(뭄바이)	ap-south-1	2016년 6월 27일
캐나다(중부)	ca-central-1	2017년 10월 28일
EU(프랑크푸르트)	eu-central-1	2015년 10월 1일

리전 이름	리전	추적 시작 날짜
EU(아일랜드)	eu-west-1	2015년 10월 1일
EU(런던)	eu-west-2	2017년 10월 28일
EU(파리)	eu-west-3	2017년 12월 18일
남아메리카(상파울루)	sa-east-1	2015년 12월 11일

앞의 표에 나와 있지 않은 리전은 서비스에서 마지막으로 액세스한 데이터를 아직 제공하지 않습니다.

서비스에서 마지막으로 액세스한 데이터 보기

AWS Management 콘솔, AWS CLI 또는 AWS API를 사용하여 마지막으로 액세스한 데이터를 볼 수 있습니다. 서비스에서 마지막으로 액세스한 데이터에 대한 자세한 내용은 [서비스에서 마지막으로 액세스한 데이터를 사용하여 권한 줄이기 \(p. 409\)](#) 단원을 참조하십시오.

Note

IAM의 리소스에 대한 액세스 데이터를 보려면 먼저 보고 기간, 보고된 엔터티 및 데이터에 대해 평가된 정책 유형을 이해해야 합니다. 자세한 내용은 [the section called “알아야 할 것들” \(p. 409\)](#) 단원을 참조하십시오.

엔터티 작업 보기(콘솔)

IAM 콘솔에서 사용자, 그룹, 역할 또는 정책에 대한 액세스 관리자 탭에서 서비스에서 마지막으로 액세스한 데이터를 볼 수 있습니다.

서비스에서 마지막으로 액세스한 데이터를 보려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 그룹, 사용자, 역할 또는 정책을 선택합니다.
3. 사용자, 그룹, 역할 또는 정책 이름을 선택하여 요약 페이지를 열고 액세스 관리자 탭을 선택합니다. 선택한 리소스를 기반으로 다음 정보를 확인합니다.
 - 그룹 – 그룹 멤버(사용자)가 액세스할 수 있는 서비스 목록, 멤버가 마지막으로 서비스에 액세스한 시간, 사용된 그룹 정책 및 요청한 그룹 멤버를 표시합니다. 정책의 이름을 선택하여 정책이 관리형 정책인지 아니면 인라인 그룹 정책인지 확인합니다. 그룹 멤버의 이름을 선택하여 그룹의 모든 멤버를 확인하고 마지막으로 서비스에 액세스한 시간을 확인합니다.
 - 사용자 – 사용자가 액세스할 수 있는 서비스 목록, 서비스에 마지막으로 액세스한 시간 및 사용된 정책 목록을 표시합니다. 정책이 관리되는지 여부를 확인할 정책 이름, 인라인 사용자 정책 또는 사용자가 속한 그룹의 인라인 정책을 확인합니다.
 - 역할 – 역할이 액세스할 수 있는 서비스 목록, 서비스에 마지막으로 액세스한 역할 및 사용된 정책 목록을 표시합니다. 정책의 이름을 선택하여 정책이 관리형 정책인지 아니면 인라인 역할 정책인지 확인합니다.
 - 정책 – 정책에 허용된 작업, 서비스에 마지막으로 액세스한 정책 및 해당 정책을 사용한 엔터티(사용자 또는 역할)가 포함된 서비스 목록을 표시합니다. 엔터티의 이름을 선택하여 어떤 엔터티에 이 정책이 연결되어 있는지 그리고 마지막으로 서비스에 액세스한 시간을 확인합니다.

엔터티 작업 보기(AWS CLI)

AWS CLI를 사용하여 AWS 서비스에 액세스하기 위해 IAM 리소스(사용자, 그룹, 역할 또는 정책)가 사용된 마지막 시간에 대한 데이터를 검색할 수 있습니다.

서비스에서 마지막으로 액세스한 데이터를 보려면(AWS CLI)

1. 보고서를 생성합니다. 요청에는 보고서가 필요한 IAM 리소스(사용자, 그룹, 역할 또는 정책)의 ARN이 포함되어야 합니다. 작업이 완료될 때까지 `get-service-last-accessed-details` 및 `get-service-last-accessed-details-with-entities` 작업에서 `job-status`를 모니터링하기 위해 사용할 수 있는 `job-id`를 반환합니다.
 - [aws iam generate-service-last-accessed-details](#)
2. 이전 단계의 `job-id` 파라미터를 사용하여 보고서에 대한 세부 정보를 검색합니다.
 - [aws iam get-service-last-accessed-details](#)

이 작업은 `generate-service-last-accessed-details` 작업에서 요청한 리소스 유형에 따라 다음 정보를 반환합니다.

- 사용자 – 지정한 사용자가 액세스할 수 있는 서비스 목록을 반환합니다. 각 서비스에 대해 작업은 사용자의 마지막 시도 날짜 및 시간과 사용자의 ARN을 반환합니다.
 - 그룹 – 그룹에 연결된 정책을 사용하여 지정된 그룹의 멤버가 액세스할 수 있는 서비스 목록을 반환합니다. 각 서비스에 대해 작업은 그룹 멤버(사용자)가 마지막으로 시도한 날짜와 시간을 반환합니다. 또한 해당 사용자의 ARN과 서비스에 액세스하려고 시도한 그룹 멤버의 총 수를 반환합니다. 모든 멤버 목록을 반환하려면 [GetServiceLastAccessedDetailsWithEntities](#) 작업을 사용합니다.
 - 역할 – 지정한 역할이 액세스할 수 있는 서비스 목록을 반환합니다. 각 서비스에 대해 작업은 역할의 마지막 시도 날짜 및 시간과 역할의 ARN을 반환합니다.
 - 정책 – 지정된 정책으로 액세스할 수 있는 서비스 목록을 반환합니다. 각 서비스에 대해 작업은 엔터티(사용자 또는 역할)가 정책을 사용하여 마지막으로 서비스에 액세스하려고 시도한 날짜와 시간을 반환합니다. 또한 엔터티의 ARN과 액세스를 시도한 엔터티의 총 수를 반환합니다.
3. 특정 서비스에 액세스하기 위해 그룹 또는 정책 권한을 사용하는 엔터티에 대해 자세히 알아봅니다. 이 작업은 각 엔터티의 ARN, ID, 이름, 경로, 유형(사용자 또는 역할) 및 마지막으로 서비스에 액세스하려고 시도한 엔터티의 목록을 반환합니다. 사용자와 역할에 대해 이 작업을 사용할 수도 있지만 해당 엔터티에 대한 정보만 반환합니다.
 - [aws iam get-service-last-accessed-details-with-entities](#)
 4. 특정 서비스에 액세스하기 위해 자격 증명(사용자, 그룹 또는 역할)이 사용하는 자격 기반 정책에 대해 자세히 알아봅니다. 자격 증명 및 서비스를 지정한 경우 이 작업은 해당 자격 증명이 지정된 서비스에 액세스하는 데 사용할 수 있는 권한 정책 목록을 반환합니다. 이 작업은 정책의 현재 상태를 제공하며 생성된 보고서에 의존하지 않습니다. 또한 리소스 기반 정책, 액세스 제어 목록, AWS Organizations 정책, IAM 권한 경계 또는 세션 정책 등의 다른 정책 유형을 반환하지 않습니다. 자세한 내용은 [정책 유형](#) (p. 305) 또는 [단일 계정 내에서 정책 평가](#) (p. 532) 단원을 참조하십시오.
 - [aws iam list-policies-granting-service-access](#)

엔터티 작업 보기(AWS API)

AWS API를 사용하여 AWS 서비스에 액세스하기 위해 IAM 리소스(사용자, 그룹, 역할 또는 정책)가 사용된 마지막 시간에 대한 데이터를 검색할 수 있습니다.

서비스에서 마지막으로 액세스한 데이터를 보려면(AWS API)

1. 보고서를 생성합니다. 요청에는 보고서가 필요한 IAM 리소스(사용자, 그룹, 역할 또는 정책)의 ARN이 포함되어야 합니다. 작업이 완료될 때까지 `GetServiceLastAccessedDetails` 및 `GetServiceLastAccessedDetailsWithEntities` 작업에서 `JobStatus`를 모니터링하기 위해 사용할 수 있는 `JobId`를 반환합니다.
 - [GenerateServiceLastAccessedDetails](#)
2. 이전 단계의 `JobId` 파라미터를 사용하여 보고서에 대한 세부 정보를 검색합니다.

- [GetServiceLastAccessedDetails](#)

이 작업은 `GenerateServiceLastAccessedDetails` 작업에서 요청한 리소스 유형에 따라 다음 정보를 반환합니다.

- 사용자 – 지정한 사용자가 액세스할 수 있는 서비스 목록을 반환합니다. 각 서비스에 대해 작업은 사용자의 마지막 시도 날짜 및 시간과 사용자의 ARN을 반환합니다.
 - 그룹 – 그룹에 연결된 정책을 사용하여 지정된 그룹의 멤버가 액세스할 수 있는 서비스 목록을 반환합니다. 각 서비스에 대해 작업은 그룹 멤버(사용자)가 마지막으로 시도한 날짜와 시간을 반환합니다. 또한 해당 사용자의 ARN과 서비스에 액세스하려고 시도한 그룹 멤버의 총 수를 반환합니다. 모든 멤버 목록을 반환하려면 [GetServiceLastAccessedDetailsWithEntities](#) 작업을 사용합니다.
 - 역할 – 지정한 역할이 액세스할 수 있는 서비스 목록을 반환합니다. 각 서비스에 대해 작업은 역할의 마지막 시도 날짜 및 시간과 역할의 ARN을 반환합니다.
 - 정책 – 지정된 정책으로 액세스할 수 있는 서비스 목록을 반환합니다. 각 서비스에 대해 작업은 엔터티(사용자 또는 역할)가 정책을 사용하여 마지막으로 서비스에 액세스하려고 시도한 날짜와 시간을 반환합니다. 또한 엔터티의 ARN과 액세스를 시도한 엔터티의 총 수를 반환합니다.
3. 특정 서비스에 액세스하기 위해 그룹 또는 정책 권한을 사용하는 엔터티에 대해 자세히 알아봅니다. 이 작업은 각 엔터티의 ARN, ID, 이름, 경로, 유형(사용자 또는 역할) 및 마지막으로 서비스에 액세스하려고 시도한 엔터티의 목록을 반환합니다. 사용자와 역할에 대해 이 작업을 사용할 수도 있지만 해당 엔터티에 대한 정보만 반환합니다.
- [GetServiceLastAccessedDetailsWithEntities](#)
4. 특정 서비스에 액세스하기 위해 자격 증명(사용자, 그룹 또는 역할)이 사용하는 자격 기반 정책에 대해 자세히 알아봅니다. 자격 증명 및 서비스를 지정한 경우 이 작업은 해당 자격 증명이 지정된 서비스에 액세스하는 데 사용할 수 있는 권한 정책 목록을 반환합니다. 이 작업은 정책의 현재 상태를 제공하며 생성된 보고서에 의존하지 않습니다. 또한 리소스 기반 정책, 액세스 제어 목록, AWS Organizations 정책, IAM 권한 경계 또는 세션 정책 등의 다른 정책 유형을 반환하지 않습니다. 자세한 내용은 [정책 유형 \(p. 305\)](#) 또는 [단일 계정 내에서 정책 평가 \(p. 532\)](#) 단원을 참조하십시오.
- [ListPoliciesGrantingServiceAccess](#)

액세스 데이터 사용에 대한 예제 시나리오

서비스에서 마지막으로 액세스한 데이터를 사용하여 IAM 엔터티(사용자 또는 역할)에 부여할 권한을 결정할 수 있습니다. 자세한 정보는 [서비스에서 마지막으로 액세스한 데이터를 사용하여 권한 줄이기 \(p. 409\)](#) 단원을 참조하십시오.

Note

IAM의 리소스에 대한 액세스 데이터를 보려면 먼저 보고 기간, 보고된 엔터티 및 데이터에 대해 평가된 정책 유형을 이해해야 합니다. 자세한 정보는 [the section called “알아야 할 것들” \(p. 409\)](#) 단원을 참조하십시오.

조직에 적합한 액세스 가능성과 최소 권한 간에 적절한 균형을 유지하는 것은 IAM 관리자에게 달려 있습니다.

데이터를 사용하여 그룹의 권한 줄이기

서비스에서 마지막으로 액세스한 데이터를 사용하여 사용자에게 필요한 서비스만 포함하도록 그룹 권한을 줄일 수 있습니다. 이 방법은 서비스 수준에서 [최소 권한을 부여 \(p. 44\)](#)하는 데 있어 중요한 단계입니다.

예를 들어, Paulo Santos는 Example Corp.의 AWS 사용자 권한을 정의하는 관리자입니다. 이 회사는 AWS를 사용한 지 얼마 되지 않았기 때문에 소프트웨어 개발 팀에서 아직 사용할 AWS 서비스를 정의하지 않았습니다. Paulo는 팀에게 필요한 서비스에만 액세스할 수 있는 권한을 부여하려고 하지만, 아직 정의되지 않았습니다.

기 때문에 일시적으로 파워 사용자 권한을 부여합니다. 그런 다음 그는 서비스에서 마지막으로 액세스한 데이터를 사용하여 그룹의 권한을 줄입니다.

Paulo는 다음 JSON 텍스트를 사용하여 ExampleDevelopment 관리 정책을 만듭니다. 그런 다음 이 정책을 Development 그룹에 연결하고 모든 개발자를 그룹에 추가합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "FullAccessToAllServicesExceptPeopleManagement",  
            "Effect": "Allow",  
            "NotAction": [  
                "iam:*",  
                "organizations:/*"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Sid": "RequiredIamAndOrgsActions",  
            "Effect": "Allow",  
            "Action": [  
                "iam:CreateServiceLinkedRole",  
                "iam>DeleteServiceLinkedRole",  
                "iam>ListRoles",  
                "organizations:DescribeOrganization"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

Paulo는 90일 후에 AWS Management 콘솔을 사용하여 Development 그룹에 대해 서비스에서 마지막으로 액세스한 데이터 (p. 412)를 보기로 결정합니다. 그는 그룹 멤버가 액세스한 서비스 목록을 확인합니다. 그는 사용자가 지난 주에 5개의 서비스(AWS CloudTrail, Amazon CloudWatch Logs, Amazon EC2, AWS KMS 및 Amazon S3)에 액세스했다는 사실을 알게 되었습니다. 그들은 AWS를 처음 평가할 때 몇 가지 다른 서비스에 액세스했지만 그 이후에는 액세스하지 않았습니다.

Paulo는 5가지 서비스와 필요한 IAM 및 조직 작업만 포함하도록 정책 권한을 줄이기로 결정합니다. 그는 다음 JSON 텍스트를 사용하여 ExampleDevelopment 정책을 편집합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "FullAccessToListedServices",  
            "Effect": "Allow",  
            "Action": [  
                "s3:/*",  
                "kms:/*",  
                "cloudtrail:/*",  
                "logs:/*",  
                "ec2:/*"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Sid": "RequiredIamAndOrgsActions",  
            "Effect": "Allow",  
            "Action": [  
                "iam:CreateServiceLinkedRole",  
                "iam>DeleteServiceLinkedRole",  
                "iam:ListRoles",  
                "organizations:DescribeOrganization"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

```
        "iam>ListRoles",
        "organizations:DescribeOrganization"
    ],
    "Resource": "*"
}
}
```

권한을 더 줄이기 위해 Paulo는 AWS CloudTrail 이벤트 기록에서 계정의 이벤트를 확인할 수 있습니다. 여기서 그는 개발자가 필요로 하는 작업과 리소스만 포함하도록 정책 권한을 줄이기 위해 사용할 수 있는 자세한 이벤트 정보를 볼 수 있습니다. 자세한 정보는 AWS CloudTrail 사용 설명서에서 [CloudTrail 콘솔에서 CloudTrail 이벤트 보기](#) 단원을 참조하십시오.

데이터를 사용하여 IAM 사용자의 권한 축소

서비스에서 마지막으로 액세스한 데이터를 사용하여 개별 IAM 사용자의 권한을 줄일 수 있습니다.

예를 들어 Martha Rivera는 회사 직원들의 AWS 권한이 초과되지 않도록 관리하는 IT 관리자입니다. 정기 보안 검사의 일환으로 Martha는 모든 IAM 사용자의 권한을 검토합니다. 이러한 사용자 가운데는 애플리케이션 개발자로, 이전에 보안 엔지니어의 역할을 담당했던 Nikhil Jayashankar 씨도 있습니다. 작업 요구 사항의 변화로 인해 Nikhil은 app-dev 그룹과 security-team 그룹의 멤버입니다. 그의 새로운 직무와 관련된 app-dev 그룹은 Amazon EC2, Amazon EBS, Auto Scaling, Route 53 및 Elastic Transcoder를 포함하여 여러 서비스에 대한 권한을 부여합니다. 이전 직무와 관련된 security-team 그룹은 IAM 및 CloudTrail에 대한 권한을 부여합니다.

관리자인 Martha는 IAM 콘솔에 로그인하고 사용자를 선택한 다음 nikhilj을 선택하고 액세스 관리자 탭을 선택합니다.

Martha는 마지막 액세스 열을 검토하여 Nikhil이 최근에 IAM, CloudTrail, Route 53, Amazon Elastic Transcoder 및 기타 여러 AWS 서비스에 액세스하지 않았다는 것을 확인합니다. 회사 내에서 Martha는 Nikhil이 더 이상 내부 보안 팀의 멤버가 아니므로 업무적으로 IAM 및 CloudTrail에 액세스할 필요가 없다는 것을 확인합니다.

Martha는 이제 서비스에서 마지막으로 액세스한 데이터에 대한 작업을 수행할 수 있습니다. 그러나 이전 예제의 그룹과 달리 nikhilj과 같은 IAM 사용자는 여러 정책을 준수하고 여러 그룹의 멤버가 될 수 있습니다. Martha는 nikhilj 또는 다른 그룹 멤버의 액세스를 실수로 방해하지 않도록 주의해서 진행해야 합니다. Nikhil에게 부여할 액세스 권한의 종류뿐만 아니라, 이러한 권한을 받는 방법을 결정해야 합니다.

Martha는 권한 탭을 선택합니다. 이 탭에서 nikhilj에 직접 연결된 정책 및 그룹을 통해 연결된 정책을 확인합니다. 그녀는 각 정책을 확장하고 Nikhil이 사용하지 않는 서비스에 대한 액세스를 허용하는 정책을 알아보기 위해 정책 요약을 확인합니다.

- IAM – IAMFullAccess AWS 관리형 정책은 nikhilj에 직접 연결되고 security-team 그룹에 연결됩니다.
- CloudTrail – AWSCloudTrailReadOnlyAccess AWS 관리형 정책은 security-team 그룹에 연결됩니다.
- Route 53 – App-Dev-Route53 고객 관리형 정책은 app-dev 그룹에 연결됩니다.
- Elastic Transcoder – App-Dev-ElasticTranscoder 고객 관리형 정책은 app-dev 그룹에 연결됩니다.

Martha는 nikhilj에 직접 연결된 IAMFullAccess AWS 관리형 정책을 제거하기로 결정했습니다. 또한 security-team 그룹에 대한 Nikhil의 멤버십을 제거합니다. 이 두 작업은 IAM 및 CloudTrail에 대한 불필요한 액세스를 제거합니다.

Route 53 및 Elastic Transcoder에 액세스할 수 있는 Nikhil의 권한은 app-dev 그룹에 의해 부여됩니다. Nikhil은 이러한 서비스를 사용하지 않지만 그룹의 다른 멤버에게는 필요할 수 있습니다. Martha는 app-dev 그룹에 대해 서비스에서 마지막으로 액세스한 데이터를 검토하고 최근에 Route 53에 액세스한 멤버가 여러

명 있지만 작년에는 Elastic Transcoder에 액세스한 그룹 멤버가 없었음을 확인했습니다. 그녀는 그룹에서 App-Dev-ElasticTranscoder 고객 관리형 정책을 제거합니다.

그런 다음 Martha는 고객 관리형 정책인 App-Dev-ElasticTranscoder에 대해 서비스에서 마지막으로 액세스한 데이터를 검토합니다. 그녀는 정책이 다른 IAM 자격 증명에 연결되지 않았다는 것을 알게 됩니다. 그녀는 회사 내에서 앞으로 해당 정책이 필요하지 않다는 것을 조사한 다음 삭제합니다.

리소스 삭제 전 데이터 사용

IAM 리소스를 삭제하기 전에 서비스에서 마지막으로 액세스한 데이터를 사용하여 마지막으로 리소스를 사용한 이후로 일정 시간이 경과했는지 확인할 수 있습니다. 이는 사용자, 그룹, 역할 및 정책에 적용됩니다.

정책 편집 전 데이터 사용

해당 리소스에 영향을 미치는 정책을 편집하기 전에 서비스에서 마지막으로 액세스한 데이터를 IAM 자격 증명(사용자, 그룹 또는 역할) 또는 정책에 대해 검토할 수 있습니다. 이 기능은 사용 중인 사용자의 액세스 권한을 제거하지 않으려는 경우 중요합니다.

예를 들어, Arnav Desai는 개발자이고 Example Corp.의 AWS 관리자입니다. Arnav의 팀이 AWS를 사용하기 시작했을 때 그들은 모든 개발자에게 IAM 및 조직을 제외한 모든 서비스에 대한 전체 액세스를 허용하는 파워 사용자 권한을 부여했습니다. Arnav는 [최소 권한 부여 \(p. 44\)](#)를 위한 첫 걸음으로 AWS CLI를 사용하여 자신의 계정에서 관리형 정책을 검토하려고 합니다.

이렇게 하기 위해 Arnav는 먼저 다음 명령을 사용하여 자격 증명에 연결된 계정에 고객 관리형 권한 정책을 나열합니다.

```
aws iam list-policies --scope Local --only-attached --policy-usage-filter PermissionsPolicy
```

응답에서 그는 각 정책에 대한 ARN을 캡처합니다. 그런 다음 Arnav는 다음 명령을 사용하여 각 정책에 대해 서비스에서 마지막으로 액세스한 데이터 보고서를 생성합니다.

```
aws iam generate-service-last-accessed-details --arn arn:aws:iam::123456789012:policy/ExamplePolicy1
```

이 응답에서 그는 JobId 필드에서 생성된 보고서의 ID를 캡처합니다. 그런 다음 Arnav는 JobStatus 필드가 COMPLETED 또는 FAILED 값을 반환할 때까지 다음 명령을 폴링합니다. 작업이 실패할 경우 오류를 캡처합니다.

```
aws iam get-service-last-accessed-details --job-id 98a765b4-3cde-2101-2345-example678f9
```

작업의 상태가 COMPLETED이면 Arnav는 JSON 형식의 ServicesLastAccessed 배열에 대한 콘텐츠를 구문 분석합니다.

```
"ServicesLastAccessed": [
    {
        "TotalAuthenticatedEntities": 1,
        "LastAuthenticated": "2018-11-01T21:24:33.222Z",
        "ServiceNamespace": "dynamodb",
        "LastAuthenticatedEntity": "arn:aws:iam::123456789012:user/IAMExampleUser",
        "ServiceName": "Amazon DynamoDB"
    },
    {
        "TotalAuthenticatedEntities": 0,
        "ServiceNamespace": "ec2",
        "ServiceName": "Amazon EC2"
    }
]
```

```
},
{
    "TotalAuthenticatedEntities": 3,
    "LastAuthenticated": "2018-08-25T15:29:51.156Z",
    "ServiceNamespace": "s3",
    "LastAuthenticatedEntity": "arn:aws:iam::123456789012:role/IAMExampleRole",
    "ServiceName": "Amazon S3"
}
]
```

이 정보를 통해 Arnav는 ExamplePolicy1 정책이 Amazon DynamoDB, Amazon S3 및 Amazon EC2 세 가지 서비스에 대한 액세스를 허용한다는 것을 알게 됩니다. 11월 1일 IAM 사용자 IAMExampleUser가 DynamoDB에 마지막으로 액세스하려고 시도했으며, 8월 25일에는 어떤 사용자가 Amazon S3에 액세스하기 위해 IAMExampleRole 역할을 사용했습니다. 작년에 Amazon S3에 액세스하려고 시도한 엔터티가 두 개 더 있습니다. 그러나 작년에 Amazon EC2에 액세스하려고 시도한 사용자는 아무도 없었습니다.

이는 Arnav가 정책에서 Amazon EC2 작업을 안전하게 제거할 수 있음을 의미합니다. Arnav는 정책에 대한 현재 JSON 문서를 검토하려고 합니다. 먼저, 다음 명령을 사용하여 정책의 버전 번호를 결정해야 합니다.

```
aws iam list-policy-versions --policy-arn arn:aws:iam::123456789012:policy/ExamplePolicy1
```

응답에서 Arnav는 Versions 배열에서 현재 기본 버전 번호를 수집합니다. 그런 다음, 다음 명령을 통해 해당 버전 번호(v2)를 사용하여 JSON 정책 문서를 요청합니다.

```
aws iam get-policy-version --policy-arn arn:aws:iam::123456789012:policy/ExamplePolicy1 --version-id v2
```

Arnav는 반환된 JSON 정책 문서를 PolicyVersion 배열의 Document 필드에 저장합니다. 정책 문서에서 Arnav는 ec2 네임스페이스에서 작업을 검색합니다. 정책에 남아 있는 다른 네임스페이스의 작업이 없는 경우 영향을 받는 자격 증명(사용자, 그룹 및 역할)에서 정책을 분리한 다음 정책을 삭제합니다. 이 경우 정책에는 Amazon DynamoDB 및 Amazon S3 서비스가 포함되므로 Arnav는 문서에서 Amazon EC2 작업을 제거하고 변경 사항을 저장합니다. 그는 다음 명령을 사용하여 새 버전의 문서를 통해 정책을 업데이트하고 해당 버전을 기본 정책 버전으로 설정합니다.

```
aws iam create-policy-version --policy-arn arn:aws:iam::123456789012:policy/ExamplePolicy1 --policy-document file://UpdatedPolicy.json --set-as-default
```

이제 ExamplePolicy1 정책이 업데이트되어 불필요한 Amazon EC2 서비스에 대한 액세스를 제거합니다.

기타 시나리오

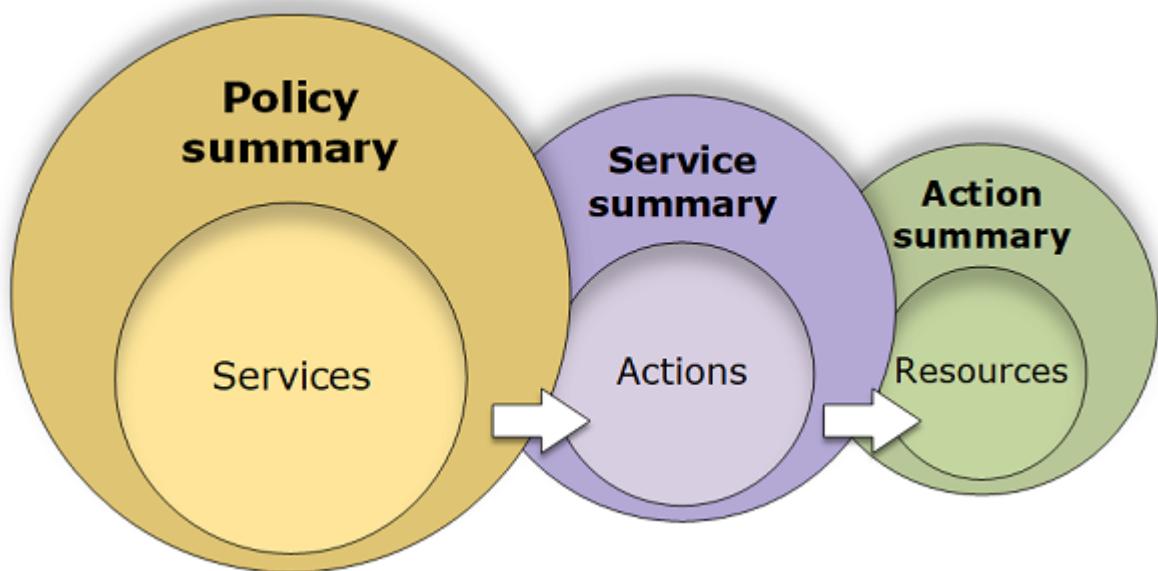
IAM 리소스(사용자, 그룹, 역할 또는 정책)가 서비스에 마지막으로 액세스하려고 시도한 시점에 대한 정보는 다음 작업 중 하나를 완료할 때 도움이 될 수 있습니다.

- 정책 – [기존 고객 관리형 또는 인라인 정책을 편집하여 권한 제거](#) (p. 402)
- 정책 – [인라인 정책을 관리형 정책으로 변환한 다음 삭제](#) (p. 45)
- 정책 – [기존 정책에 명시적 거부 추가](#) (p. 537)
- 정책 – [자격 증명\(사용자, 그룹 또는 역할\)에서 관리형 정책 분리](#) (p. 394)
- 정책 – [관리형 정책 삭제\(이로 인해 자격 증명에서 정책이 분리됨\)](#) (p. 406)
- 엔터티 – [엔터티\(사용자 또는 역할\)가 가질 수 있는 최대 권한을 제어하도록 권한 경계 설정](#) (p. 391)
- 그룹 – [그룹에서 사용자 제거](#) (p. 149)
- 그룹 – [그룹 삭제](#) (p. 151)

- 사용자 – [사용자 삭제 \(p. 72\)](#)
- 역할 – [역할 삭제 \(p. 255\)](#)

정책에 의해 부여된 권한 이해

IAM 콘솔에는 정책에서 각 서비스에 대해 허용되거나 거부되는 액세스 레벨, 리소스, 조건을 설명하는 정책 요약 테이블이 포함되어 있습니다. 정책은 3가지 테이블, 즉 [정책 요약 \(p. 419\)](#), [서비스 요약 \(p. 429\)](#), [작업 요약 \(p. 433\)](#)으로 요약됩니다. 정책 요약 테이블에는 서비스 목록이 포함되어 있습니다. 서비스 요약을 보려면 여기서 서비스를 선택합니다. 이 요약 테이블에는 작업 목록과 선택한 서비스에 대해 연결된 권한이 포함되어 있습니다. 해당 테이블에서 작업을 선택하여 작업 요약을 볼 수 있습니다. 이 테이블에는 리소스 목록과 선택한 작업에 대한 조건이 포함되어 있습니다.



사용자 페이지 또는 역할 페이지에서 해당 사용자에 연결된 모든 정책(관리형 및 인라인)에 대한 정책 요약을 볼 수 있습니다. 정책 페이지에서 모든 관리형 정책에 대한 요약을 봅니다. 관리형 정책에는 AWS 관리형 정책, AWS 관리형 직무 정책, 고객 관리형 정책이 포함되어 있습니다. 정책이 사용자 또는 다른 IAM 자격 증명에 연결되어 있는지 여부와 상관없이 정책 페이지에서 이러한 정책에 대한 요약을 볼 수 있습니다.

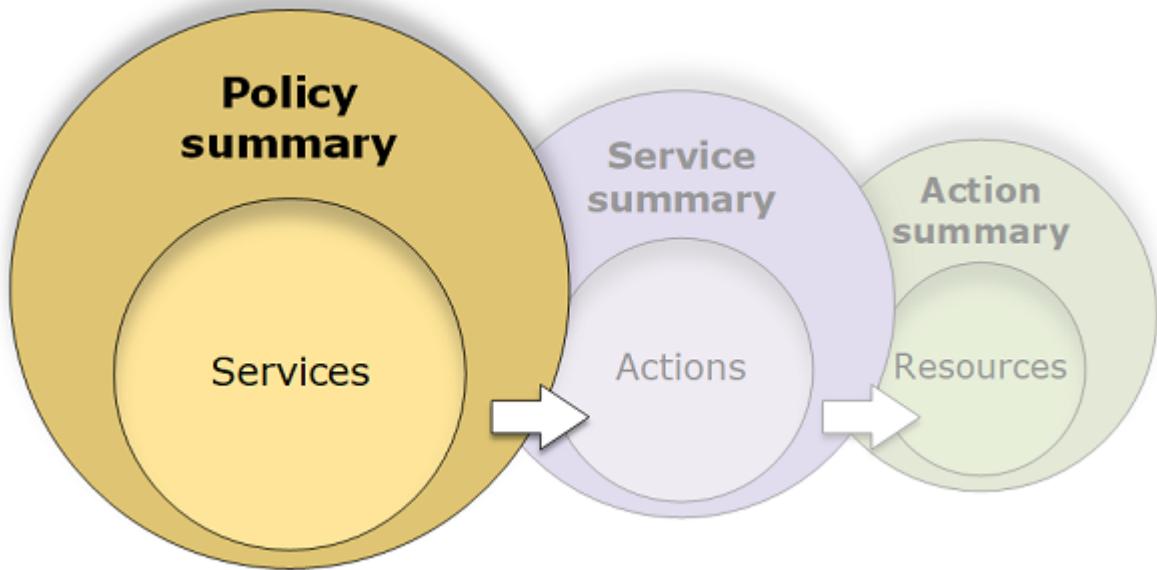
정책 요약의 정보를 사용하여 정책에서 허용되거나 거부된 권한을 확인할 수 있습니다. 정책 요약은 예상한 권한을 제공하지 않는 정책의 [문제를 해결 \(p. 454\)](#)하고 정책을 수정하는 데 도움이 됩니다.

주제

- [정책 요약\(서비스 목록\) \(p. 419\)](#)
- [서비스 요약\(작업 목록\) \(p. 429\)](#)
- [작업 요약\(리소스 목록\) \(p. 433\)](#)
- [정책 요약 예제 \(p. 435\)](#)

정책 요약(서비스 목록)

정책은 3가지 테이블, 즉 정책 요약, [서비스 요약 \(p. 429\)](#), [작업 요약 \(p. 433\)](#)으로 요약됩니다. 정책 요약 테이블에는 서비스 목록과 선택한 정책에 의해 정의된 권한의 요약이 포함되어 있습니다.



정책 요약 테이블은 하나 이상의 Uncategorized services(미분류 서비스), 명시적 거부, 허용 섹션으로 그룹화됩니다. IAM에서 인식하지 못하는 서비스가 정책에 포함되어 있으면 해당 서비스는 테이블의 Uncategorized services(미분류 서비스) 섹션에 포함됩니다. IAM에서 서비스를 인식하면 해당 서비스는 정책(Deny 또는 Allow)의 효과에 따라 테이블의 명시적 거부 또는 허용 섹션에 포함됩니다.

정책 요약 보기

사용자 페이지에서 사용자에게 연결된 정책에 대한 요약을 볼 수 있습니다. 역할 페이지에서 역할에 연결된 정책에 대한 요약을 볼 수 있습니다. 정책 페이지에서 관리형 정책에 대한 정책 요약을 볼 수 있습니다. 정책에 정책 요약이 포함되지 않은 경우 [정책 요약 누락 \(p. 458\)](#)을 참조하여 이유를 알아보십시오.

정책 페이지에서 정책 요약을 확인하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.
3. 정책 목록에서 보려는 정책의 이름을 선택합니다.
4. 정책 요약을 보려면 해당 정책의 요약 페이지에서 권한 탭을 확인합니다.

사용자에 연결된 정책의 요약을 확인하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 사용자 목록에서 정책을 보려는 사용자의 이름을 선택합니다.
4. 사용자에게 직접 연결되거나 그룹에서 연결된 정책의 목록을 보려면 해당 사용자의 요약 페이지에서 권한 탭을 봅니다.
5. 사용자에 대한 정책 테이블에서 보려는 정책의 행을 확장합니다.

역할에 연결된 정책의 요약 정보를 보려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택합니다.
3. 역할 목록에서 정책을 보려는 역할의 이름을 선택합니다.
4. 역할의 요약 페이지에서 권한 탭을 보고 역할에 연결된 정책 목록을 확인합니다.
5. 역할에 대한 정책 테이블에서 보려는 정책의 행을 확장합니다.

정책을 편집하여 경고 수정

정책 요약을 보는 동안 정책에서 예상한 권한을 제공하지 않는다는 알림이나 오타를 찾을 수 있습니다. 정책 요약을 직접 편집할 수 없습니다. 그러나 정책 요약이 보고하는 것과 동일한 여러 개의 오류 및 경고를 파악하는 시각적 정책 편집기를 사용하여 관리형 정책을 편집할 수 있습니다. 그런 다음 정책 요약의 변경 사항을 확인하여 모든 문제가 수정되었는지 확인할 수 있습니다. 인라인 정책을 편집하는 방법에 대해 자세히 알아보려면 [the section called "IAM 정책 편집" \(p. 402\)](#) 단원을 참조하십시오. 단, AWS 관리형 정책은 편집할 수 없습니다.

시각적 편집기 탭을 사용하여 정책 요약에 대한 정책을 편집하려면

1. 이전 절차에서 설명한 대로 정책의 요약을 엽니다.
2. 정책 편집을 선택합니다.

사용자 페이지에서 해당 사용자에게 연결된 고객 관리형 정책을 편집하려는 경우, 정책 페이지로 리디렉션됩니다. 고객 관리형 정책은 정책 페이지에서만 편집할 수 있습니다.

3. 편집 가능한 정책의 시각적 표시를 보려면 시각적 편집기 탭을 선택합니다. IAM은 시각적 편집기에서 모양을 최적화하고 문제를 쉽게 찾아 수정하기 위해 정책을 재구성할 수 있습니다. 페이지의 경고 및 오류 메시지는 정책 문제를 수정하도록 안내할 수 있습니다. IAM이 정책을 재구성하는 방법에 대한 자세한 내용은 [정책 재구성 \(p. 455\)](#) 단원을 참조하십시오.
4. 정책을 편집하고 정책 검토를 선택하여 정책 요약에 반영된 변경 사항을 봅니다. 문제가 계속 표시되면 이전을 선택하여 편집 화면으로 돌아갑니다.
5. [Save]를 선택하여 변경 사항을 저장합니다.

JSON 탭을 사용하여 정책 요약에 대한 정책을 편집하려면

1. 이전 절차에서 설명한 대로 정책의 요약을 엽니다.
2. {} JSON과 정책 요약을 선택하여 정책 요약과 JSON 정책 문서를 비교합니다. 이 정보를 사용하여 정책 문서에서 변경할 행을 결정할 수 있습니다.
3. 정책 편집을 선택한 다음 JSON 탭을 선택하여 JSON 정책 문서를 편집합니다.

Note

언제든지 Visual editor(시각적 편집기) 및 JSON 탭을 전환할 수 있습니다. 그러나 변경을 수행하거나 Visual editor(시각적 편집기) 탭에서 정책 검토를 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 내용은 [정책 재구성 \(p. 455\)](#) 단원을 참조하십시오.

사용자 페이지에서 해당 사용자에게 연결된 고객 관리형 정책을 편집하려는 경우, 정책 페이지로 리디렉션됩니다. 고객 관리형 정책은 정책 페이지에서만 편집할 수 있습니다.

4. 정책을 편집하고 정책 검토를 선택하여 정책 요약에 반영된 변경 사항을 봅니다. 문제가 계속 표시되면 이전을 선택하여 편집 화면으로 돌아갑니다.
5. [Save]를 선택하여 변경 사항을 저장합니다.

정책 요약의 요소 이해하기

다음의 사용자 세부 정보 페이지 예제에서는 PolSumUser 사용자에 8개 정책이 연결되어 있습니다. SummaryAllElements 정책은 사용자에게 직접 연결된 관리형 정책(고객 관리형 정책)입니다. 이 정책이 확장되어 정책 요약을 표시합니다. 이 정책의 JSON 정책 문서를 보려면 [the section called “SummaryAllElements JSON 정책 문서” \(p. 426\)](#) 단원을 참조하십시오.

User ARN: arn:aws:iam::072398337363:user/PolSumUser
Path: /
Creation time: 2017-02-16 12:58 PDT

Permissions Groups (1) Security credentials Access Advisor

Add permissions Attached policies: 8

Policy name	Policy type
SummaryAllElements	Managed policy

This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition. For details, choose Show remaining. [Learn more](#)

Policy summary { } JSON Edit policy Simulate policy

Filter

Service	Access level	Resource	Request condition
Unrecognized services			
codeddeploy	⚠		
Explicit deny (1 of 103 services)			
S3	⚠	Full: Read, Write, Permissions management Limited: List	Multiple
Allow (3 of 103 services) Show remaining 100			
Billing	Full: Read Limited: Write	All resources	Multiple
EC2	⚠	None	All resources
S3	⚠	Limited: Write, Permissions management	BucketName = developer_bucket, ObjectPath = All

EC2_Troubleshoot Managed policy

이전 이미지에서 정책 요약은 사용자 세부 정보 페이지에 표시되어 있습니다.

1. 사용자의 권한 탭에는 PolSumUser 사용자에 연결된 정책이 포함됩니다.
2. SummaryAllElements 정책은 사용자에게 연결된 몇 가지 정책 중 하나입니다. 정책 요약을 보려면 정책을 확장합니다.
3. 정책에서 정책에 정의된 일부 작업, 리소스 및 조건에 권한을 부여하지 않는 경우 페이지 상단에 경고 또는 오류 배너가 나타납니다. 그런 다음 정책 요약에 문제에 대한 세부 정보가 포함됩니다. 정책 요약이 정책에서 부여하는 권한을 이해하고 문제를 해결하는데 얼마나 도움이 되는지 알아보려면 [the section called “정책이 필요한 권한을 부여하지 않음” \(p. 460\)](#) 단원을 참조하십시오.
4. 정책 요약 및 { } JSON 버튼을 사용하여 정책 요약과 JSON 정책 문서 사이를 전환합니다.
5. 정책 시뮬레이션(Simulate policy)을 선택하면 정책을 테스트하기 위한 정책 시뮬레이터가 열립니다.
6. 검색 상자를 사용하여 서비스 목록을 제한하면 용이하게 특정 서비스를 찾을 수 있습니다.

7. 확장된 보기는 SummaryAllElements 정책의 세부 정보를 보여 줍니다.

다음 정책 요약 테이블 이미지는 PoISumUser 사용자 세부 정보 페이지에서 확장된 SummaryAllElements 정책입니다.

A Service	G Access level	H Resource	I Request condition
B Unrecognized services			
codedploy ⚠			
C Explicit deny (1 of 103 services)			
D Allow (3 of 103 services) Show remaining 100			
S3 ⚠	Full: Read, Write, Permissions management Limited: List	Multiple	None
Billing	Full: Read Limited: Write	All resources	Multiple
EC2 ⚠	None	All resources	None
S3 ⚠	Limited: Write, Permissions management	BucketName = developer_bucket, ObjectPath = All	s3:x-amz-acl = public-read

이전 이미지에서 정책 요약은 사용자 세부 정보 페이지에 표시되어 있습니다.

A. 서비스 – 이 열에는 정책에서 정의된 서비스가 나열되고 각 서비스의 세부 정보를 제공합니다. 정책 요약 테이블에서 각 서비스 이름은 서비스 요약 테이블에 대한 링크([서비스 요약\(작업 목록\) \(p. 429\)](#) 단원 참조)입니다. 이 예제에서는 Amazon S3, 결제 및 Amazon EC2 서비스에 대해 권한이 정의되어 있습니다. 정책은 IAM에서 인식하지 못하는 (잘못 입력한) codedploy 서비스에 대한 권한도 정의합니다.

B. Unrecognized services(미분류 서비스) – 이 정책에는 인식할 수 없는 서비스(이 경우 codedploy ⚠)가 포함됩니다. 이 경고를 사용하여 서비스 이름에 오타가 포함되어 있는지 확인할 수 있습니다. 서비스 이름이 정확하면 서비스는 정책 요약을 지원할 수 없거나 프리뷰에 있거나 사용자 지정 서비스일 수 있습니다. 일반적으로 사용할 수 있는(GA) 서비스에 대한 정책 요약 지원을 요청하려면 [서비스가 IAM 정책 요약을 지원하지 않음 \(p. 459\)](#)을 참조하십시오. 이 예제에서는 정책이 codedploy가 누락된 인식할 수 없는 e 서비스를 포함합니다. 이 오타로 인해 정책은 예상되는 AWS CodeDeploy 권한을 제공하지 않습니다. 정확한 codedploy 서비스 이름을 포함하도록 [정책을 편집 \(p. 421\)](#) 할 수 있습니다. 그러면 서비스가 정책 요약에 나타납니다.

C. IAM에서 인식하는 해당 서비스의 경우 정책이 서비스 사용을 허용하거나 명시적으로 거부하는지 여부에 따라 서비스가 정렬됩니다. 이 예제에서는 정책이 Amazon S3 서비스에 대한 Allow 및 Deny 설명문을 포함합니다. 따라서 정책 요약의 명시적 거부 및 허용 섹션 모두에 S3가 포함되어 있습니다.

D. Show remaining 100(나머지 100개 보기) – 이 링크를 선택하여 정책에 의해 정의되지 않은 서비스를 포함하도록 테이블을 확장합니다. 이러한 서비스는 이 정책 내에서 명시적으로 거부(또는 기본적으로 거부)됩니다. 그러나 다른 정책 문으로 서비스를 사용하여 허용하거나 명시적으로 거부할 수 있습니다. 정책 요약에는 단일 정책의 권한이 요약되어 있습니다. AWS 서비스가 지정된 요청을 허용하거나 거부할지 여부를 결정하는 방법에 대해 알아보려면 [정책 평가 로직 \(p. 531\)](#)을 참조하십시오.

E. EC2 ⚠ – 이 서비스에는 미인식 작업이 포함됩니다. IAM은 정책 요약을 지원하는 서비스 이름, 작업 및 리소스 유형을 인식합니다. 서비스는 인식되지만 인식되지 않은 작업을 포함하면 IAM은 해당 서비스 옆에 경고를 포함합니다. 이 예제에서는 IAM이 한 개 이상의 Amazon EC2 작업을 인식하지 못합니다. 인식할 수 없는 작업에 대해 자세히 알아보고 S3 서비스 요약에서 인식할 수 없는 작업을 보려면 [서비스 요약\(작업 목록\) \(p. 429\)](#) 단원을 참조하십시오.

Note

IAM은 정책 요약을 지원하는 서비스의 이름, 작업 및 리소스 유형을 검토합니다. 그러나 존재하지 않는 리소스 값이나 조건이 정책 요약에 포함될 수 있습니다. 항상 [정책 시뮬레이터 \(p. 383\)](#)로 정책을 테스트합니다.

F.

S3  – 이 서비스에는 미인식 리소스가 포함됩니다. IAM은 정책 요약을 지원하는 서비스 이름, 작업 및 리소스 유형을 인식합니다. 서비스가 인식되지만 인식되지 않는 리소스 유형이 있는 경우 IAM은 해당 서비스 옆에 경고를 표시합니다. 이 예제에서는 IAM이 한 개 이상의 Amazon S3 작업을 인식하지 못합니다. 인식할 수 없는 리소스에 대해 자세히 알아보고 S3 서비스 요약에서 인식할 수 없는 리소스 유형을 보려면 [서비스 요약\(작업 목록\) \(p. 429\)](#) 단원을 참조하십시오.

G.

Access level(액세스 레벨) – 이 열은 정책이 각 액세스 레벨(List, Read, Write, 및 Permissions management)의 작업에 대해 Full 또는 Limited 중 어느 권한을 정의했는지 보여줍니다. 액세스 레벨 요약에 대한 자세한 정보 및 예제는 [정책 요약에서 액세스 레벨 요약 이해하기 \(p. 427\)](#) 단원을 참조하십시오.

- Full access(전체 액세스) – 이 항목은 해당 서비스가 서비스에 대해 사용 가능한 4개 액세스 레벨 모두에서 모든 작업에 액세스할 수 있음을 나타냅니다. 이 예제에서는 이 행이 테이블의 명시적 거부 섹션에 포함되어 있으므로 정책에 포함된 리소스에서 모든 Amazon S3 작업이 거부됩니다.
- 항목에 Full access(전체 액세스)가 포함되지 않은 경우 해당 서비스는 서비스를 위한 모든 작업이 아니라 일부 작업에 액세스할 수 있습니다. 그러면 액세스 권한이 4개 액세스 레벨(List, Read, Write 및 Permissions management) 각각에 대한 다음의 설명으로 정의됩니다.

Full(전체): 정책이 나열된 각 액세스 레벨 분류의 모든 작업에 대한 액세스 권한을 제공합니다. 이 예제에서는 정책이 모든 결제 Read 작업에 대한 액세스 권한을 제공합니다.

Limited(제한): 정책이 나열된 각 액세스 레벨 분류에서 하나 이상의 작업(모든 작업은 아님)에 대한 액세스 권한을 제공합니다. 이 예제에서는 정책이 일부 결제 write 작업에 대한 액세스 권한을 제공합니다.

H.

리소스 – 이 열은 정책이 각 서비스에 대해 지정한 리소스를 보여줍니다.

- All – 정책이 서비스 내 둘 이상(모든 리소스는 아님)의 리소스를 포함합니다. 이 예제에서는 둘 이상의 Amazon S3 리소스에 대한 액세스가 명시적으로 거부됩니다.
- All resources(모든 리소스) – 정책이 서비스의 모든 리소스에 대해 정의되어 있습니다. 이 예제에서는 정책이 모든 결제 리소스에 대해 나열된 작업을 수행할 수 있도록 허용합니다.
- Resource text – 정책이 서비스의 리소스 하나를 포함합니다. 이 예제에서는 나열된 작업이 developer_bucket Amazon S3 버킷 리소스에서만 허용됩니다. 서비스가 IAM에 제공하는 정 보에 따라 arn:aws:s3:::developer_bucket/* 등의 ARN이 표시되거나 BucketName = developer_bucket 등의 정의된 리소스 유형이 표시될 수 있습니다.

Note

이 열은 다른 서비스의 리소스를 포함할 수 있습니다. 리소스를 포함하는 정책 설명에 동일한 서비스의 작업과 리소스를 모두 포함하지 않으면 정책에 일치하지 않는 리소스가 포함됩니다. IAM은 정책을 생성하거나 정책 요약에서 정책을 볼 때 일치하지 않는 리소스에 대해 경고하지 않습니다. 이 열에 일치하지 않는 리소스가 포함되어 있으면 정책에 오류가 있는지 검토해야 합니다. 정책을 더 잘 이해하려면 항상 [정책 시뮬레이터 \(p. 383\)](#)로 테스트합니다.

I.

Request condition(요청 조건) – 이 열은 리소스와 연결된 서비스 또는 작업에 조건이 적용되는지 여부를 나타냅니다.

- 없음 – 정책이 서비스에 대한 조건을 포함하지 않습니다. 이 예제에서는 Amazon S3 서비스에서 거부된 작업에 적용된 조건이 없습니다.
- Condition text – 정책이 서비스에 대한 조건 하나를 포함합니다. 이 예제에서는 소스의 IP 주소가 203.0.113.0/24와 일치하는 경우에만 나열된 결제 작업이 허용됩니다.
- All – 정책이 서비스에 대해 둘 이상의 조건을 포함합니다. 이 예제에서는 나열된 Amazon S3 작업에 대한 액세스가 복수의 조건에 따라 허용됩니다. 정책에 대한 여러 조건을 각각 보려면 {} JSON을 선택하여 정책 문서를 봅니다.

정책 또는 정책 내 요소가 권한을 부여하지 않는 경우 IAM은 정책 요약에 추가 경고 및 정보를 제공합니다. 다음 정책 요약 테이블은 PolSumUser 사용자 세부 정보 페이지에 확장된 Show remaining 100(나머지 100 개 보기) 서비스와 가능한 경고를 보여줍니다.

Service	Access level	a Resource	b Request condition
Unrecognized services			
codedploy			
Explicit deny (1 of 103 services)			
S3	Full: Read, Write, Permissions management Limited: List	Multiple ⚠ One or more actions do not have an applicable resource.	None
Allow (3 of 103 services) Hide remaining 100			
d ...	e None		
Billing	Full: Read Limited: Write	All resources	Multiple
CodeBuild	f ⚠ None - No actions are defined.	arn:aws:codebuild:us-east-1:123456789012:project/my-demo-project	None
CodeCommit	None	g ⚠ No resources are defined.	None
CodeDeploy	⚠ None - No actions are defined.	arn:aws:codedeploy:us-west-2:123456789012:deploymentgroup:*	None
EC2	⚠ None	All resources	None
S3	Limited: Write, Permissions management	BucketName = developer_bucket, ObjectPath = All ⚠ One or more resources do not have an applicable action.	s3:x-amz-acl = public-read ⚠ One or more conditions do not have an applicable action.

앞의 그림에는 권한이 없이 정의된 작업, 리소스 또는 조건을 포함하는 모든 서비스가 나와 있습니다.

- a. Resource warnings(리소스 경고) – 포함된 모든 작업이나 리소스에 대해 권한을 제공하지 않는 서비스의 경우 테이블의 리소스 열에 다음 경고 중 하나가 나타납니다.

- ⚠ 정의된 리소스가 없습니다. – 서비스에 작업이 정의되었지만 정책에 지원되는 리소스가 포함되지 않았음을 의미합니다.
- ⚠ 하나 이상의 작업이 적용할 리소스가 없습니다. – 서비스에 정의된 작업이 있지만 일부 작업에 지원되는 리소스가 없음을 의미합니다.
- ⚠ 하나 이상의 리소스가 적용할 작업이 없습니다. – 서비스에 정의된 리소스가 있지만 일부 리소스에 지원 작업이 없음을 의미합니다.

서비스에 적용 가능한 리소스가 없는 작업과 적용 가능한 작업이 없는 리소스가 있는 경우, One or more resources do not have an applicable action(하나 이상의 리소스에 적용할 작업이 없습니다). 경고가 표시됩니다. 그 이유는 서비스에 대한 서비스 요약을 볼 때 어떤 작업에도 적용되지 않는 리소스가 표시되지 않기 때문입니다. `ListAllMyBuckets` 작업의 경우 리소스 수준 권한을 지원하지 않고 `s3:x-amz-acl` 조건 키를 지원하지 않기 때문에 이 정책에 마지막 경고가 포함됩니다. 리소스 문제나 조건 문제를 수정한 경우 나머지 문제가 세부 경고에 나타납니다.

- b. Request condition warnings(요청 조건 경고) – 포함된 모든 조건에 대해 권한을 제공하지 않는 서비스의 경우 테이블의 Request condition(요청 조건)열에 다음 경고 중 하나가 나타납니다.

- ⚠ 하나 이상의 작업이 적용할 조건이 없습니다. – 서비스에 정의된 작업이 있지만 일부 작업에 지원되는 조건이 없음을 의미합니다.
- ⚠ 하나 이상의 조건이 적용할 작업이 없습니다. – 서비스에 정의된 조건이 있지만 일부 조건에 지원 작업이 없음을 의미합니다.

- c. Multiple | ⚠ 하나 이상의 작업이 적용할 리소스가 없습니다. – Amazon S3의 Deny 문에 리소스가 두 개 이상 포함되어 있습니다. 또한 작업이 두 개 이상 포함되어 있으며, 일부 작업은 리소스를 지원하고, 일부 작업은 리소스를 지원하지 않습니다. 정책을 보려면 [the section called "SummaryAllElements JSON 정책](#)

문서” (p. 426)를 참조하십시오. 이 경우 정책에는 모든 Amazon S3 작업과, 정의된 버킷 또는 버킷 객체에서 수행될 수 있는 작업만 포함됩니다.

- d. 줄임표(...)는 모든 서비스가 페이지에 포함되었지만 이 정책과 관련된 정보가 있는 행만 표시되었음을 나타냅니다. AWS Management 콘솔에서 이 페이지를 보면 모든 AWS 서비스를 볼 수 있습니다.
- e. 테이블 행의 배경색은 어떤 권한도 부여하지 않는 서비스를 나타냅니다. 정책 요약에서 이러한 서비스에 대한 추가 정보를 볼 수 없습니다. 흰색 행의 서비스의 경우, 서비스 이름을 선택하여 서비스 요약(작업 목록) 페이지를 볼 수 있습니다. 이 페이지에는 해당 서비스에 대해 부여된 권한에 대한 자세한 정보가 표시됩니다.
- f. 없음 - 정의된 작업이 없습니다. – 서비스가 리소스나 조건으로 정의되었지만, 서비스에 대해 포함된 작업이 없으며 따라서 서비스가 권한을 제공하지 않음을 의미합니다. 이 경우 정책에 CodeBuild 리소스가 포함되지만 CodeBuild 작업은 포함되지 않습니다.
- g. 정의된 리소스가 없습니다. – 서비스에 정의된 작업이 있지만, 지원되는 리소스가 정책에 없으며 따라서 서비스가 권한을 제공하지 않습니다. 이 경우 정책에 CodeCommit 작업이 포함되지만 CodeCommit 리소스는 포함되지 않습니다.
- h. BucketName = developer_bucket, ObjectPath = All | 하나 이상의 리소스가 적용할 작업이 없습니다.
– 서비스에 정의된 버킷 객체 리소스가 한 개 있고, 지원 작업이 없는 리소스가 한 개 이상 있습니다.
- i. s3:x-amz-acl = public-read | 하나 이상의 조건이 적용할 작업이 없습니다. – 서비스에 정의된 s3:x-amz-acl 조건 키가 한 개 있고, 지원 작업이 없는 조건 키가 한 개 이상 있습니다.

SummaryAllElements JSON 정책 문서

SummaryAllElements 정책은 해당 계정의 권한을 정의하는 데 사용하기 위한 것이 아닙니다. 이것은 정책 요약을 보는 중 만날 수 있는 오류와 경고를 보여주기 위한 것입니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "aws-portal:ViewBilling",  
                "aws-portal:ViewPaymentMethods",  
                "aws-portal:ModifyPaymentMethods",  
                "aws-portal:ViewAccount",  
                "aws-portal:ModifyAccount",  
                "aws-portal:ViewUsage"  
            ],  
            "Resource": [  
                "*"  
            ],  
            "Condition": {  
                "IpAddress": {  
                    "aws:SourceIp": "203.0.113.0/24"  
                }  
            }  
        },  
        {  
            "Effect": "Deny",  
            "Action": [  
                "s3:*"  
            ],  
            "Resource": [  
                "arn:aws:s3::::customer",  
                "arn:aws:s3::::customer/*"  
            ]  
        }  
    ]  
}
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:GetConsoleScreenshots"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "codedploy:/*",
                "codecommit:/*"
            ],
            "Resource": [
                "arn:aws:codedeploy:us-west-2:123456789012:deploymentgroup:/*",
                "arn:aws:codebuild:us-east-1:123456789012:project/my-demo-project"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3>ListAllMyBuckets",
                "s3GetObject",
                "s3DeleteObject",
                "s3PutObject",
                "s3PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::developer_bucket",
                "arn:aws:s3:::developer_bucket/*",
                "arn:aws:autoscaling:us-east-2:123456789012:autoscalgrp"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": [
                        "public-read"
                    ],
                    "s3:prefix": [
                        "custom",
                        "other"
                    ]
                }
            }
        }
    ]
}
```

정책 요약에서 액세스 레벨 요약 이해하기

정책 요약에는 해당 정책에서 언급된 각 서비스에 정의된 작업 권한을 설명하는 액세스 레벨 요약이 포함됩니다. 정책 요약에 대한 자세한 내용은 [정책에 의해 부여된 권한 이해 \(p. 419\)](#) 단원을 참조하십시오. 액세스 레벨 요약은 각 액세스 레벨(List, Read, Write, Permissions management)의 작업에 정책에 정의된 Full 또는 Limited 권한이 있는지 여부를 나타냅니다. 서비스의 각 작업에 할당된 액세스 레벨 분류를 보려면 [??? 단원](#)을 참조하십시오.

다음 예제에서는 한 정책이 지정된 서비스에 대해 제공하는 액세스 권한을 설명합니다. 전체 JSON 정책 문서 및 관련 요약의 예는 [정책 요약 예제 \(p. 435\)](#) 단원을 참조하십시오.

서비스	액세스 레벨	이 정책은 다음과 같은 액세스 권한을 제공합니다.
IAM	모든 액세스	IAM 서비스 내의 모든 작업에 대한 액세스
CloudWatch	전체: 목록	List 액세스 레벨의 모든 CloudWatch 작업에 대한 액세스 권한. 하지만 Read, Write 또는 Permissions management 액세스 레벨 분류의 작업에 대한 액세스 권한은 제공하지 않음
데이터 파일 프라인	제한: 목록, 읽기	List 및 Read 액세스 레벨의 하나 이상의 AWS Data Pipeline 작업(모든 작업은 아님)에 대한 액세스 권한. 단, Write 또는 Permissions management 작업에 대한 액세스 권한은 제외됨
EC2	전체: 목록, 읽기 제한: 쓰기	모든 Amazon EC2 List 및 Read 작업에 대한 액세스 권한, 하나 이상의 Amazon EC2 Write 작업(모든 작업은 아님)에 대한 액세스 권한. 단, Permissions management 액세스 레벨 분류의 작업에 대한 액세스 권한은 제외됨
S3	제한: 읽기, 쓰기, 권한 관리	하나 이상의 Amazon S3 Read, Write 및 Permissions management 작업(모든 작업은 아님)에 대한 액세스 권한
codedploy	(비어 있음)	알 수 없는 액세스(IAM에서 이 서비스를 인식하지 않음)
API 게이트 웨이	없음	정책에 정의된 액세스 없음
CodeBuild	 정의된 작업 없음.	서비스에 대해 작업이 정의되지 않아서 액세스할 수 없습니다. 이 문제를 이해하고 문제를 해결하는 방법을 보려면 the section called “정책이 필요한 권한을 부여하지 않음” (p. 460) 단원을 참조하십시오.

앞서 언급한 바와 같이 (p. 424), 모든 액세스는 정책이 서비스 내 모든 작업에 대한 액세스를 제공함을 나타냅니다. 서비스의 모든 작업이 아니라 일부에 대한 액세스 권한을 제공하는 정책은 액세스 레벨 분류에 따라 추가로 그룹화됩니다. 이는 다음 액세스 레벨 그룹 중 하나에 의해 표시됩니다.

- 전체: 정책이 지정된 액세스 레벨 분류의 모든 작업에 대한 액세스 권한을 제공합니다.
- 제한: 정책이 지정된 액세스 레벨 분류 내 하나 이상의 작업(모든 작업은 아님)에 대한 액세스 권한을 제공합니다.
- 없음: 정책에서 액세스를 제공하지 않습니다.
- (비어 있음): IAM에서 이 서비스를 인식하지 않습니다. 서비스 이름에 오타가 포함되어 있으면 정책은 서비스에 대한 액세스를 제공하지 않습니다. 서비스 이름이 정확하면 서비스는 정책 요약을 지원할 수 없거나 프리뷰에 있을 수 있습니다. 이 경우 정책은 액세스를 제공할 수 있지만 해당 액세스를 정책 요약에 표시할 수 없습니다. 일반적으로 사용할 수 있는(GA) 서비스에 대한 정책 요약 지원을 요청하려면 [서비스가 IAM 정책 요약을 지원하지 않음 \(p. 459\)](#)을 참조하십시오.

작업에 대한 부분적 액세스 권한을 포함하는 액세스 레벨 요약은 다음의 액세스 레벨 분류를 사용하여 그룹화됩니다.

- 목록: 서비스의 리소스를 나열하여 객체가 존재하는지 판단할 수 있는 권한입니다. 이 액세스 레벨의 작업은 객체를 나열할 수 있으나 리소스의 내용을 확인할 수 없습니다. 예를 들어 Amazon S3 작업 ListBucket의 액세스 레벨은 목록입니다.

- 읽기: 서비스에서 리소스 내용과 속성을 읽을 수 있으나 편집할 수 없는 권한입니다. 예를 들어 Amazon S3 작업 `GetObject` 및 `GetBucketLocation`의 액세스 레벨은 읽기입니다.
- 쓰기: 서비스에서 리소스를 생성, 삭제하거나 수정할 수 있는 권한입니다. 예를 들어 Amazon S3 작업 `CreateBucket`, `DeleteBucket` 및 `PutObject`의 액세스 레벨은 쓰기입니다.
- 권한 관리: 서비스에서 리소스 권한을 부여하거나 수정할 수 있는 권한입니다. 예를 들어 대부분의 IAM 및 AWS Organizations 작업과 Amazon S3 작업 `PutBucketPolicy` 및 `DeleteBucketPolicy` 등의 액세스 레벨은 권한 관리입니다.

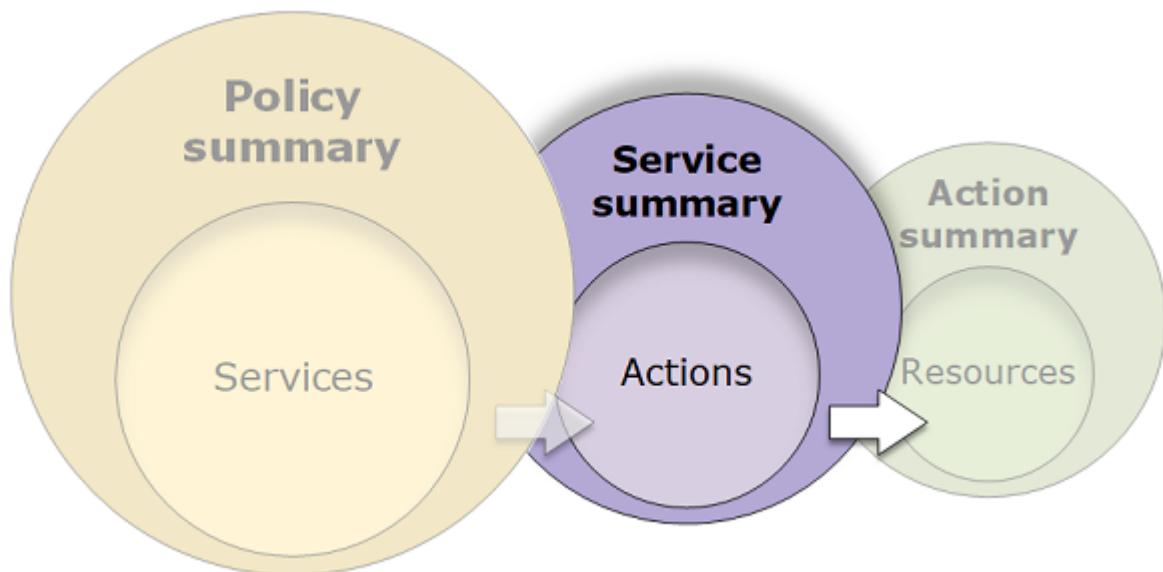
도움말

AWS 계정의 보안을 개선하려면 권한 관리 액세스 레벨 분류를 포함하는 정책을 제한하거나 정기적으로 모니터링합니다.

- 태그 지정: 태그 지정을 지원하는 서비스에서 리소스를 생성, 삭제하거나 수정할 수 있는 권한. 예를 들어, IAM `CreateUser`, `TagUser` 및 `UntagUser` 작업의 액세스 레벨은 태그 지정입니다. 이러한 작업은 태그 추가 또는 제거 등 리소스를 수정할 수 있는 권한을 부여합니다.

서비스 요약(작업 목록)

정책은 3가지 테이블, 즉 정책 요약 (p. 419), 서비스 요약, 작업 요약 (p. 433)으로 요약됩니다. 서비스 요약 테이블에는 작업 목록과 선택한 서비스의 정책에 의해 정의된 권한의 요약이 포함되어 있습니다.



권한을 부여하는 정책 요약에 나열되어 있는 각 서비스에 대해 서비스 요약을 볼 수 있습니다. 이 테이블은 Uncategorized actions(미분류 작업), Uncategorized resource types(미분류 리소스 유형) 및 액세스 수준 섹션으로 분류되어 있습니다. IAM에서 인식하지 못하는 작업이 정책에 포함되어 있으면 해당 작업은 테이블의 Uncategorized actions(미분류 작업) 섹션에 포함됩니다. IAM에서 작업을 인식하면 해당 작업은 테이블의 액세스 레벨(목록, 읽기, 쓰기, 권한 관리) 섹션 중 하나에 포함됩니다. 서비스의 각 작업에 할당된 액세스 레벨 분류를 보려면 ??? 단원을 참조하십시오.

서비스 요약 보기

정책 페이지에서 관리형 정책에 대한 서비스 요약을 보거나, 사용자 페이지 및 역할을 통해 사용자나 역할에 연결된 인라인 및 관리형 정책에 대한 서비스 요약을 볼 수 있습니다. 단, 관리형 정책의 사용자 페이지 또는 역할 페이지에서 서비스 이름을 선택한 경우에는 정책 페이지로 리디렉션됩니다. 관리형 정책에 대한 서비스 요약은 정책 페이지에서 확인해야 합니다.

관리형 정책의 서비스 요약을 확인하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.
3. 정책 목록에서 보려는 정책의 이름을 선택합니다.
4. 정책 요약을 보려면 해당 정책의 요약 페이지에서 권한 탭을 확인합니다.
5. 정책 요약 서비스 목록에서 확인하려는 서비스의 이름을 선택합니다.

사용자에게 연결된 정책의 서비스 요약을 확인하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 사용자 목록에서 정책을 보려는 사용자의 이름을 선택합니다.
4. 사용자에게 직접 연결되거나 그룹에서 연결된 정책의 목록을 보려면 해당 사용자의 요약 페이지에서 권한 탭을 봅니다.
5. 사용자에 대한 정책 테이블에서 보려는 정책의 행을 확장합니다.
6. 정책 요약 서비스 목록에서 확인하려는 서비스의 이름을 선택합니다.

Note

선택한 정책이 사용자에게 직접 연결된 인라인 정책인 경우 서비스 요약 테이블이 표시됩니다. 정책이 그룹에서 연결한 인라인 정책인 경우 해당 그룹의 JSON 정책 문서로 자동으로 이동합니다. 정책이 관리형 정책인 경우 정책 페이지에서 해당 정책의 서비스 요약이 게시된 부분으로 자동으로 이동합니다.

역할에 연결된 정책의 서비스 요약 정보를 보려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택합니다.
3. 역할 목록에서 정책을 보려는 역할의 이름을 선택합니다.
4. 역할의 요약 페이지에서 권한 탭을 보고 역할에 연결된 정책 목록을 확인합니다.
5. 역할에 대한 정책 테이블에서 보려는 정책의 행을 확장합니다.
6. 정책 요약 서비스 목록에서 확인하려는 서비스의 이름을 선택합니다.

서비스 요약의 요소 이해하기

아래 예제는 SummaryAllElements 정책 요약에서 허용한 Amazon S3 작업의 서비스 요약입니다([the section called "SummaryAllElements JSON 정책 문서" \(p. 426\)](#) 참조). 이 서비스에 대한 작업은 Uncategorized actions(미분류 작업), Uncategorized resource types(미분류 리소스 유형) 및 액세스 레벨로 그룹화됩니다. 예를 들어 서비스에서 이용 가능한 총 21개 쓰기 작업 중에서 2개 쓰기 작업이 정의됩니다.

This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition. For details, choose Show remaining. [Learn more](#)

[Policy summary](#) [JSON](#) [Edit policy](#)

Filter

Action (2 of 69) Hide remaining 67

Resource

Request condition

Unrecognized resource types

arn:aws:autoscaling:us-east-2:123456789012:autoscalgrp

Unrecognized actions

DeleteObject

List (0 of 4 actions)

... (No access)

ListAllMyBuckets (No access)

11 This action does not have an applicable resource and condition.

None

Read (0 of 30 actions)

GetObject (No access) BucketName = developer_bucket, ObjectPath = All

16 s3:x-amz-acl = public-read is not a supported condition key for this action.

Write (1 of 29 actions)

PutObject BucketName = developer_bucket, ObjectPath = All s3:x-amz-acl = public-read

Permissions management (1 of 6 actions)

PutObjectAcl BucketName = developer_bucket, ObjectPath = All s3:x-amz-acl = public-read

관리형 정책에 대한 서비스 요약 페이지에 포함되는 정보는 다음과 같습니다.

- 정책에서 정책의 서비스에 대해 정의된 일부 작업, 리소스 및 조건에 권한을 부여하지 않는 경우 페이지 상단에 경고 배너가 나타납니다. 그런 다음 서비스 요약에 문제에 대한 세부 정보가 포함됩니다. 정책 요약이 정책에서 부여하는 권한을 이해하고 문제를 해결하는 데 얼마나 도움이 되는지 알아보려면 [the section called "정책이 필요한 권한을 부여하지 않음" \(p. 460\)](#) 단원을 참조하십시오.
- 뒤로 링크 옆에 서비스 이름(이 경우 S3)이 표시됩니다. 이 서비스의 서비스 요약에는 정책에서 정의한 허용되는 작업의 목록이 수록되어 있습니다. 그 대신, 서비스 이름 옆에(명시적으로 거부됨) 텍스트가 표시된 경우 서비스 요약 테이블에 나열된 작업은 명시적으로 거부됩니다.
- { } JSON을 선택하면 정책에 대한 추가 세부 정보를 볼 수 있습니다. 이를 통해 작업에 적용된 모든 조건을 볼 수 있습니다. (사용자에게 직접 연결된 인라인 정책의 서비스 요약을 보려면 서비스 요약 대화 상자를 닫고 정책 요약으로 돌아가 JSON 정책 문서에 액세스해야 합니다.)
- 특정 작업의 요약을 보려면 검색 상자에 키워드를 입력하여, 사용할 수 있는 작업의 목록을 줄이십시오.
- 작업(69개 중 2개 작업) – 이 열에는 정책 내에 정의된 작업이 나열되고 각 작업에 해당하는 리소스와 조건이 제시됩니다. 정책에서 작업에 권한을 부여한 경우 작업 이름이 [작업 요약 \(p. 433\)](#) 테이블에 링크됩니다. 개수는 권한을 제공하는 인식할 수 있는 작업의 수를 나타냅니다. 총계는 서비스에 대해 알려진 작업의 수입니다. 이 예제에서는 총 69개의 알려진 S3 작업에서 2개의 작업이 권한을 제공합니다.
- Show/Hide remaining 67(나머지 67개 작업 보기/숨기기) – 알려졌지만 이 서비스에 대한 권한을 제공하지 않는 작업을 포함하는 테이블을 확장하거나 숨기려면 이 링크를 선택합니다. 링크를 확장하면 권한을 제공하지 않는 모든 요소에 대해 경고가 표시됩니다.
- Unrecognized resource types(인식되지 않은 리소스 유형) – 이 정책에, 이 서비스에 대한 정책 내에서 인식되지 않은 리소스 유형이 한 개 이상 있습니다. 이 경고를 사용하여 리소스 유형에 오타가 포함되어 있는지 확인할 수 있습니다. 리소스 유형이 정확하면 서비스는 정책 요약을 완전히 지원할 수 없거나 프리뷰에 있거나 사용자 지정 서비스일 수 있습니다. 일반적으로 사용할 수 있는(GA) 서비스에서 특정 리소스 유형에 대한 정책 요약 지원을 요청하려면 [서비스가 IAM 정책 요약을 지원하지 않음 \(p. 459\)](#)을 참조하십시오. 이 예제에서는 autoscaling 서비스 이름에 a가 누락되었습니다.

8. Unrecognized actions(인식되지 않은 작업) – 이 정책에, 이 서비스에 대한 정책 내에서 인식되지 않은 작업이 한 개 이상 있습니다. 이 경고를 사용하여 작업에 오타가 포함되어 있는지 확인할 수 있습니다. 작업 이름이 정확하면 서비스는 정책 요약을 완전히 지원할 수 없거나 프리뷰에 있거나 사용자 지정 서비스일 수 있습니다. 일반적으로 사용할 수 있는(GA) 서비스에서 특정 작업에 대한 정책 요약 지원을 요청하려면

[서비스가 IAM 정책 요약을 지원하지 않음 \(p. 459\)](#)을 참조하십시오. 이 예제에서는 `DeleteObject` 작업에 `e`가 누락되었습니다.

Note

IAM은 정책 요약을 지원하는 서비스의 이름, 작업 및 리소스 유형을 검토합니다. 그러나 존재하지 않는 리소스 값이나 조건이 정책 요약에 포함될 수 있습니다. 항상 [정책 시뮬레이터 \(p. 383\)](#)로 정책을 테스트합니다.

9. IAM에서 인식하는 해당 작업의 경우 테이블은 정책이 허용하거나 거부하는 액세스 레벨에 따라 최소 1개 이상에서 최대 4개의 섹션으로 이러한 작업을 그룹화합니다. 섹션은 목록, 읽기, 쓰기, 권한 관리입니다. 각 액세스 레벨 내에서 사용할 수 있는 총 작업 수로부터 정의된 작업 수도 확인할 수 있습니다. 서비스의 각 작업에 할당된 액세스 레벨 분류를 보려면 [??? 단원](#)을 참조하십시오.

10. 줄임표(...)는 모든 작업이 페이지에 포함되었지만 이 정책과 관련된 정보가 있는 행만 표시되었음을 나타냅니다. AWS Management 콘솔에서 이 페이지를 보면 서비스에 대한 모든 작업을 볼 수 있습니다.

11. (No access)(액세스 권한 없음) – 이 정책에 권한을 제공하지 않는 작업이 한 개 있습니다.

12. 권한을 제공하지 않는 작업에는 작업 요약에 대한 링크가 포함됩니다.

13. 리소스 – 이 열은 정책이 서비스에 대해 정의한 리소스를 보여줍니다. IAM은 리소스가 각 작업에 적용되는지 여부를 확인하지 않습니다. 이 예제에서는 S3 서비스의 작업이 `developer_bucket` Amazon S3 버킷 리소스에서만 허용됩니다. 서비스가 IAM에 제공하는 정보에 따라 `arn:aws:s3:::developer_bucket/*` 등의 ARN이 표시되거나 `BucketName = developer_bucket` 등의 정의된 리소스 유형이 표시될 수 있습니다.

Note

이 열은 다른 서비스의 리소스를 포함할 수 있습니다. 리소스를 포함하는 정책 설명에 동일한 서비스의 작업과 리소스를 모두 포함하지 않으면 정책에 일치하지 않는 리소스가 포함됩니다. IAM은 정책을 생성하거나 정책 요약에서 정책을 볼 때 일치하지 않는 리소스에 대해 경고하지 않습니다. 또한 IAM은 작업이 리소스에 적용되는지 여부는 나타내지 않고 서비스가 일치하는지 여부만 나타냅니다. 이 열에 일치하지 않는 리소스가 포함되어 있으면 정책에 오류가 있는지 검토해야 합니다. 정책을 더 잘 이해하려면 항상 [정책 시뮬레이터 \(p. 383\)](#)로 테스트합니다.

14. 리소스 경고 – 전체 권한을 제공하지 않는 리소스를 포함하는 작업의 경우 다음 경고 중 하나가 나타납니다.

- This action does not support resource-level permissions. 리소스에 대하여 와일드카드(*)가 필요합니다. – 정책에 리소스 수준 권한이 있지만, 이 작업에 대한 권한을 제공하려면 "Resource": ["*"]를 포함해야 함을 의미합니다.
- 이 작업은 적용할 리소스가 없습니다. – 지원되는 리소스 없이 작업이 정책에 포함됨을 의미합니다.
- 이 작업은 적용할 리소스 및 조건이 없습니다. – 지원되는 리소스 및 조건 없이 작업이 정책에 포함됨을 의미합니다. 이 경우 이 서비스의 정책에 포함된 조건도 있지만 이 작업에 적용되는 조건은 없습니다.

`ListAllMyBuckets` 작업의 경우 리소스 수준 권한을 지원하지 않고 `s3:x-amz-acl` 조건 키를 지원하지 않기 때문에 이 정책에 마지막 경고가 포함됩니다. 리소스 문제나 조건 문제를 수정한 경우 나머지 문제가 세부 경고에 나타납니다.

15. Request condition(요청 조건) – 이 열은 리소스와 연결된 작업에 조건이 적용되는지 여부를 나타냅니다. 이러한 조건에 대해 자세히 알아보려면 `{ }` JSON을 선택하여 JSON 정책 문서를 검토합니다.

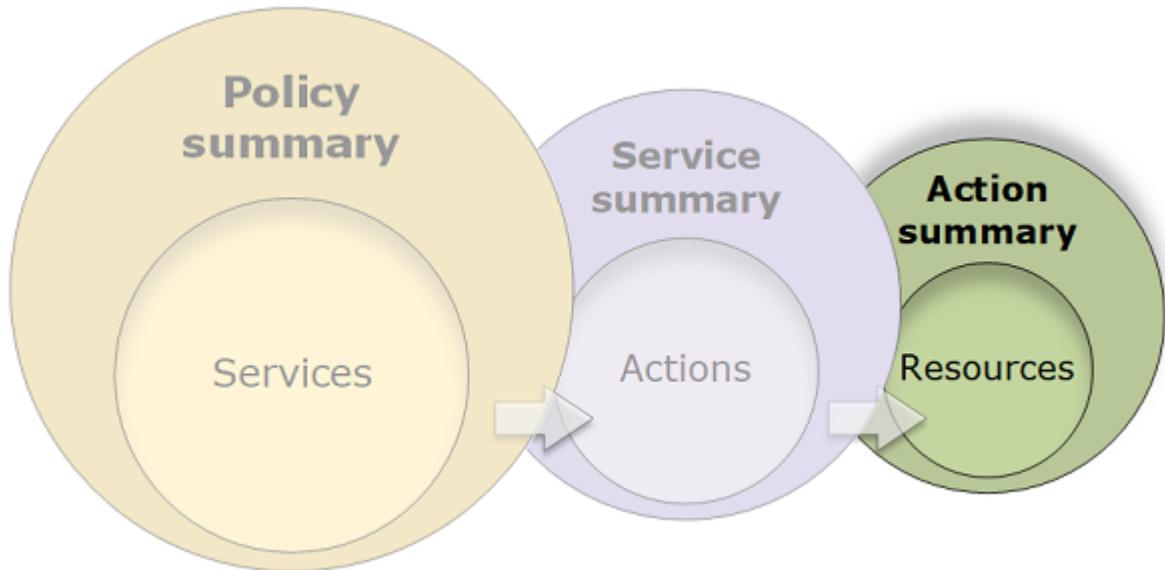
16. Condition warning(조건 경고) – 전체 권한을 제공하지 않는 조건을 포함하는 작업의 경우 다음 경고 중 하나가 나타납니다.

- <CONDITION_KEY>는 이 작업에 대해 지원되는 조건 키가 아닙니다. – 정책에 이 작업에 지원되지 않는 서비스 조건 키가 한 개 있습니다.
- 해당 작업은 여러 개의 조건 키를 지원하지 않습니다. – 정책에 이 작업에 지원되지 않는 서비스 조건 키가 두 개 이상 있습니다.

GetObject의 경우 이 정책에 s3:x-amz-acl 조건 키가 포함되며 이 키는 이 작업에서 작동하지 않습니다. 작업이 리소스를 지원하더라도, 조건이 이 작업에 대해 true가 되지 않기 때문에 정책에서 이 작업을 위한 권한을 부여하지 않습니다.

작업 요약(리소스 목록)

정책은 3가지 테이블, 즉 정책 요약 (p. 419), 서비스 요약 (p. 429), 작업 요약으로 요약됩니다. 작업 요약 테이블에는 리소스 목록과 선택한 작업에 적용되는 연결 조건이 포함되어 있습니다.



권한을 부여하는 각 작업에 대한 작업 요약을 보려면 서비스 요약의 링크를 선택합니다. 작업 요약 테이블에는 리소스의 리전 및 계정을 비롯하여 리소스에 대한 세부 정보가 포함되어 있습니다. 또한 각 리소스에 적용하는 조건을 볼 수 있습니다. 이를 통해 일부 리소스에 적용되고 다른 리소스에는 적용되지 않는 조건을 볼 수 있습니다.

작업 요약 보기

사용자 페이지에서는 사용자에게 연결된 정책에 대한 작업 요약을 볼 수 있습니다. 역할 페이지에서는 역할에 연결된 정책에 대한 작업 요약을 볼 수 있습니다. 정책 페이지에서는 관리형 정책에 대한 작업 요약을 볼 수 있습니다. 그러나 사용자 또는 역할 페이지에서 관리형 정책에 대한 작업 요약을 보려고 하면 정책 페이지로 리디렉션됩니다.

관리형 정책의 작업 요약을 확인하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.
3. 정책 목록에서 보려는 정책의 이름을 선택합니다.
4. 정책 요약을 보려면 해당 정책의 요약 페이지에서 권한 탭을 확인합니다.
5. 정책 요약 서비스 목록에서 확인하려는 서비스의 이름을 선택합니다.
6. 작업의 서비스 요약 목록에서 확인하려는 작업의 이름을 선택합니다.

사용자에게 연결된 정책의 작업 요약을 확인하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자를 선택합니다.
3. 사용자 목록에서 정책을 보려는 사용자의 이름을 선택합니다.
4. 사용자에게 직접 연결되거나 그룹에서 연결된 정책의 목록을 보려면 해당 사용자의 요약 페이지에서 권한 탭을 봅니다.
5. 사용자에 대한 정책 테이블에서 보려는 정책의 행을 확장합니다.
6. 정책 요약 서비스 목록에서 확인하려는 서비스의 이름을 선택합니다.

Note

선택한 정책이 사용자에게 직접 연결된 인라인 정책인 경우 서비스 요약 테이블이 표시됩니다. 정책이 그룹에서 연결된 인라인 정책인 경우 해당 그룹의 JSON 정책 문서로 자동으로 이동합니다. 정책이 관리형 정책인 경우 정책 페이지에서 해당 정책의 서비스 요약이 게시된 부분으로 자동으로 이동합니다.

7. 작업의 서비스 요약 목록에서 확인하려는 작업의 이름을 선택합니다.

역할 연결된 정책의 작업 요약을 보려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택합니다.
3. 역할 목록에서 정책을 보려는 역할의 이름을 선택합니다.
4. 역할의 요약 페이지에서 권한 탭을 보고 역할에 연결된 정책 목록을 확인합니다.
5. 역할에 대한 정책 테이블에서 보려는 정책의 행을 확장합니다.
6. 정책 요약 서비스 목록에서 확인하려는 서비스의 이름을 선택합니다.
7. 작업의 서비스 요약 목록에서 확인하려는 작업의 이름을 선택합니다.

작업 요약의 요소 이해하기

아래 예제는 Amazon S3 서비스 요약의 PutObject(쓰기) 작업에 대한 작업 요약입니다([서비스 요약\(작업 목록\)](#) (p. 429) 참조). 이 작업의 경우 정책이 단일 리소스에 대한 여러 조건을 정의합니다.

The screenshot shows the AWS IAM Policy Summary page for the S3:PutObject action. It displays a single result for the developer_bucket resource. The interface includes a filter input field, dropdown menus for Region and Account, and a Request condition section. The selected request condition is s3:x-amz-acl = public-read.

작업 요약 페이지에 포함되는 정보는 다음과 같습니다.

1. 뒤로 링크 옆에 서비스와 작업의 이름이 형식 `service: action`으로 표시됩니다(이 경우 S3: PutObject). 이 서비스의 작업 요약에는 정책에서 정의된 리소스의 목록이 포함되어 있습니다.
2. `{ } JSON`을 선택하면 작업에 적용되는 여러 가지 조건 등 정책에 관한 추가 세부 정보를 볼 수 있습니다. (사용자에게 직접 연결된 인라인 정책에 대한 작업 요약을 보는 경우에는 단계가 달라집니다. 그러한 경우에 JSON 정책 문서에 액세스하려면 작업 요약 대화 상자를 닫고 정책 요약으로 돌아가야 합니다.)
3. 특정 리소스의 요약을 보려면 검색 상자에 키워드를 입력하여 사용할 수 있는 리소스의 목록을 줄입니다.

4. 리소스 – 이 열에는 정책이 선택한 서비스에 대해 정의한 리소스가 나열됩니다. 이 예제에서는 PutObject 작업이 모든 객체 경로와 developer_bucket Amazon S3 버킷 리소스에서만 허용됩니다. 서비스가 IAM에 제공하는 정보에 따라 arn:aws:s3:::developer_bucket/* 등의 ARN이 표시되거나 BucketName = developer_bucket, ObjectPath = All 등의 정의된 리소스 유형이 표시될 수 있습니다.
5. 리전 – 이 열은 리소스가 정의된 리전을 보여줍니다. 리소스는 모든 리전 또는 단일 리전에 대해 정의할 수 있습니다. 리소스는 둘 이상의 특정 리전에 존재할 수 없습니다.
 - All regions(모든 리전) – 리소스와 연결된 작업은 모든 리전에 적용됩니다. 이 예제에서는 작업이 전역적 서비스 Amazon S3에 속합니다. 전역적 서비스에 속하는 작업은 모든 리전에 적용됩니다.
 - Region text(리전 텍스트) – 리소스와 연결된 작업은 한 리전에 적용됩니다. 예를 들어 정책은 리소스에 대한 us-east-2 리전을 지정할 수 있습니다.
6. 계정 – 이 열은 리소스와 연결된 서비스 또는 작업이 특정 계정에 적용되는지를 나타냅니다. 리소스는 모든 계정 또는 단일 계정에 존재할 수 있습니다. 리소스는 둘 이상의 특정 계정에 존재할 수 없습니다.
 - All accounts(모든 계정) – 리소스와 연결된 작업은 모든 계정에 적용됩니다. 이 예제에서는 작업이 전역적 서비스 Amazon S3에 속합니다. 전역적 서비스에 속하는 작업은 모든 계정에 적용됩니다.
 - This account(현재 계정) – 리소스와 연결된 작업은 현재 로그인된 계정에만 적용됩니다.
 - Account number(계정 번호) – 리소스와 연결된 작업은 하나의 계정(현재 로그인되지 않은 계정)에 적용됩니다. 예를 들어 정책이 리소스에 대한 123456789012 계정을 지정하면 계정 번호가 정책 요약에 나타납니다.
7. Request condition(요청 조건) – 이 열은 리소스와 연결된 작업에 조건이 적용되는지를 보여줍니다. 이 예제에는 s3:x-amz-acl = public-read 조건이 포함됩니다. 이러한 조건에 대해 자세히 알아보려면 {} JSON을 선택하여 JSON 정책 문서를 검토합니다.

정책 요약 예제

다음 예제에는 정책을 통해 부여되는 권한을 이해하는 데 도움이 되는 JSON 정책과 그에 연결된 정책 요약 (p. 419), 서비스 요약 (p. 429), 작업 요약 (p. 433)이 포함되어 있습니다.

정책 1: DenyCustomerBucket

이 정책은 동일한 서비스에 대한 허용과 거부를 보여줍니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "FullAccess",  
            "Effect": "Allow",  
            "Action": ["s3:*"],  
            "Resource": ["*"]  
        },  
        {  
            "Sid": "DenyCustomerBucket",  
            "Action": ["s3:*"],  
            "Effect": "Deny",  
            "Resource": ["arn:aws:s3:::customer", "arn:aws:s3:::customer/*"]  
        }  
    ]  
}
```

DenyCustomerBucket 정책 요약:

This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition. For details, choose Show remaining. [Learn more](#)

Service	Access level	Resource	Request condition
S3	Full: Read, Write, Permissions management Limited: List	Multiple	None
Allow (1 of 103 services) Show remaining 102			
S3	Full access	All resources	None

DenyCustomerBucket S3 (Explicit deny) 서비스 요약:

Action (66 of 69) Hide remaining 3	Resource	Request condition
List (1 of 4 actions)		
HeadBucket (No access)	⚠ This action does not support resource-level permissions. This requires a wildcard (*) for the resource.	None
ListAllMyBuckets(No access)	⚠ This action does not support resource-level permissions. This requires a wildcard (*) for the resource.	None
ListBucket	BucketName = customer	None
ListObjects (No access)	⚠ This action does not support resource-level permissions. This requires a wildcard (*) for the resource.	None
Read (30 of 30 actions)		
GetAccelerateConfiguration	BucketName = customer	None
GetAnalyticsConfiguration	BucketName = customer	None
GetBucketAcl	BucketName = customer	None
GetBucketCORS	BucketName = customer	None
GetBucketLocation	BucketName = customer	None
GetBucketLogging	BucketName = customer	None
GetBucketNotification	BucketName = customer	None
GetBucketPolicy	BucketName = customer	None
GetBucketRequestPayment	BucketName = customer	None
GetBucketTagging	BucketName = customer	None
GetBucketVersioning	BucketName = customer	None
GetBucketWebsite	BucketName = customer	None
GetInventoryConfiguration	BucketName = customer	None
GetIpConfiguration	BucketName = customer	None
GetLifecycleConfiguration	BucketName = customer	None
GetMetricsConfiguration	BucketName = customer	None
GetObject	BucketName = customer, ObjectPath = All	None
GetObjectAcl	BucketName = customer, ObjectPath = All	None
GetObjectTagging	BucketName = customer, ObjectPath = All	None
GetObjectTorrent	BucketName = customer, ObjectPath = All	None
GetObjectVersion	BucketName = customer, ObjectPath = All	None

GetObject(읽기) 작업 요약:

Resource	Region	Account	Request condition
BucketName = customer, ObjectPath = All	All regions	All accounts	None

정책 2: DynamoDbRowCognitoID

이 정책은 사용자의 Amazon Cognito ID를 기반으로 Amazon DynamoDB에 대한 행 수준 액세스 권한을 제공합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb>DeleteItem",
                "dynamodb>GetItem",
                "dynamodb>PutItem",
                "dynamodb>UpdateItem"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-west-1:123456789012:table/myDynamoTable"
            ],
            "Condition": {
                "ForAllValues:StringEquals": {
                    "dynamodb:LeadingKeys": [
                        "${cognito-identity.amazonaws.com:sub}"
                    ]
                }
            }
        }
    ]
}
```

DynamoDbRowCognitoID 정책 요약:

Service	Access level	Resource	Request condition
Allow (1 of 102 services) Show remaining 101			
DynamoDB	Limited: Read, Write	TableName = myDynamoTable	dynamodb:LeadingKeys = \${cognito-identity.amazonaws.com:sub}

DynamoDbRowCognitoID DynamoDB(허용) 서비스 요약:

Action (4 of 25) Show remaining 21	Resource	Request condition
Read (1 of 14 actions)		
GetItem	TableName = myDynamoTable	dynamodb:LeadingKeys = \${cognito-identity.amazonaws.com:sub}
Write (3 of 10 actions)		
DeleteItem	TableName = myDynamoTable	dynamodb:LeadingKeys = \${cognito-identity.amazonaws.com:sub}
PutItem	TableName = myDynamoTable	dynamodb:LeadingKeys = \${cognito-identity.amazonaws.com:sub}
UpdateItem	TableName = myDynamoTable	dynamodb:LeadingKeys = \${cognito-identity.amazonaws.com:sub}

.GetItem(나열) 작업 요약:

Resource	Region	Account	Request
TableName = myDynamoTable	us-west-1	123456789012	dynamodb:LeadingKeys = \${cognito-identity.amazonaws.com:sub}

정책 3: MultipleResourceCondition

이 정책에는 다수의 리소스와 조건이 포함됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": ["arn:aws:s3:::Apple_bucket/*"],
            "Condition": {"StringEquals": {"s3:x-amz-acl": ["public-read"]}}
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": ["arn:aws:s3:::Orange_bucket/*"],
            "Condition": {"StringEquals": {
                "s3:x-amz-acl": ["custom"],
                "s3:x-amz-grant-full-control": ["1234"]
            }}
        }
    ]
}
```

MultipleResourceCondition 정책 요약:

Service	Access level	Resource	Request condition
Allow (1 of 100 services) Show remaining 99			
S3	Limited: Write, Permissions management	Multiple	Multiple

MultipleResourceCondition S3(허용) 서비스 요약:

Action (2 of 52 actions)	Resource	Request condition
Write (1 of 21 actions)		
PutObject	Multiple	Multiple
Permissions management (1 of 5 actions)		
PutObjectAcl	Multiple	Multiple

PutObject(쓰기) 작업 요약:

Resource	Region	Account	Request condition
BucketName = Orange_bucket, ObjectPath = All	All regions	All accounts	Multiple
BucketName = Apple_bucket, ObjectPath = All	All regions	All accounts	s3:x-amz-acl = public-read

정책 4: EC2_Troubleshoot

다음 정책을 통해 사용자는 실행 중인 Amazon EC2 인스턴스의 스크린샷을 만들어 EC2 문제 해결에 필요한 도움을 얻을 수 있습니다. 또한 이 정책은 Amazon S3 개발자 버킷의 항목에 대한 정보를 확인할 수 있도록 허용합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:GetConsoleScreenshot"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3>ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::developer"
            ]
        }
    ]
}
```

EC2_Troubleshoot 정책 요약:

Service	Access level	Resource	Request condition
Allow (2 of 102 services) Show remaining 100			
EC2	Limited: Read	All resources	None
S3	Limited: List	BucketName = developer	None

EC2_Troubleshoot S3(허용) 서비스 요약:

Action (1 of 52) Show remaining 51	Resource	Request condition
<u>List (1 of 4 actions)</u>		
ListBucket	BucketName = developer	None

ListBucket(나열) 작업 요약:

Resource	Region	Account	Request
BucketName = developer	All regions	All accounts	None

정책 5: Unrecognized_Service_Action

다음 정책은 DynamoDB에 대한 모든 액세스 권한을 제공하기 위해 마련되었지만 dynamodb가 dynamobd로 잘못 입력되었기 때문에 해당 액세스는 실패합니다. 이 정책은 us-east-2 리전의 일부 Amazon EC2 작업에 대한 액세스를 허용하지만 ap-northeast-2 리전에 대한 해당 액세스를 거부하기 위

해 마련되었습니다. 그러나 ap-northeast-2 리전에서 인스턴스를 재부팅하기 위한 액세스는 ○ 작업 중에 인식할 수 없는 RebootInstances로 인해 명시적으로 거부되지 않습니다. 이 예제에서는 정책 요약을 사용하여 정책에서 오류를 찾는 방법을 보여줍니다. 정책 요약의 정보를 기반으로 정책을 편집하는 방법을 알아보려면 [정책을 편집하여 경고 수정 \(p. 421\)](#)을 참조하십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Action": [
                "ec2:RunInstances",
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:RebootInstances"
            ],
            "Resource": "*",
            "Effect": "Deny",
            "Condition": {
                "StringEquals": {
                    "ec2:Region": "ap-northeast-2"
                }
            }
        },
        {
            "Action": [
                "ec2:RunInstances",
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:RebootInstances"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "ec2:Region": "us-east-2"
                }
            }
        }
    ]
}
```

Unrecognized_Service_Action 정책 요약:

Service	Access level	Resource	Request condition
<u>Unrecognized services</u>			
dynamodb ⚠			
<u>Explicit deny (1 of 103 services)</u>			
EC2 ⚠	Limited: Write	All resources	ec2:Region = ap-northeast-2
<u>Allow (1 of 103 services)</u> Show remaining 102			
EC2	Limited: Write	All resources	ec2:Region = us-east-2

Unrecognized_Service_Action EC2 (Explicit deny) 서비스 요약:

Action (3 of 229) Show remaining 226	Resource	Request condition
Unrecognized actions		
RebootInstances ⚠		
Write (3 of 157 actions)		
RunInstances	All resources	ec2:Region = ap-northeast-2
StartInstances	All resources	ec2:Region = ap-northeast-2
StopInstances	All resources	ec2:Region = ap-northeast-2

Unrecognized_Service_Action StartInstances (Write) 작업 요약:

Resource	Region	Account	Request condition
All resources	All regions	All accounts	ec2:Region = ap-northeast-2

정책 6: CodeBuild_CodeCommit_CodeDeploy

이 정책은 특정 CodeBuild, CodeCommit, CodeDeploy 리소스에 대한 액세스 권한을 제공합니다. 이러한 리소스는 각 서비스에 고유하므로 일치하는 서비스에서만 나타납니다. Action 요소에 서비스와 일치하지 않는 리소스를 포함하는 경우 리소스가 모든 작업 요약에 나타납니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1487980617000",
            "Effect": "Allow",
            "Action": [
                "codebuild:*",
                "codecommit:*",
                "codedeploy:*
            ],
            "Resource": [
                "arn:aws:codebuild:us-east-2:123456789012:project/my-demo-project",
                "arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo",
                "arn:aws:codedeploy:us-east-2:123456789012:application:WordPress_App",
                "arn:aws:codedeploy:us-east-2:123456789012:instance/AssetTag*"
            ]
        }
    ]
}
```

CodeBuild_CodeCommit_CodeDeploy 정책 요약:

Service ▾	Access level	Resource	Request condition
Allow (3 of 103 services) Show remaining 100			
CodeBuild	Limited: List, Read, Write	arn:aws:codebuild:us-east-2:123456789012:project/my-demo-project	None
CodeCommit	Full: Read, Write Limited: List	arn:aws:codecommit:us-east-2:123456789012:MyDemoRepo	None
CodeDeploy	Limited: List, Read, Write	Multiple	None

CodeBuild_CodeCommit_CodeDeploy CodeBuild(허용) 서비스 요약:

Action (9 of 15) Show remaining 6	Resource	Request condition
List (1 of 3 actions)		
ListBuildsForProject	arn:aws:codebuild:us-east-2:123456789012:project/my-demo-project	None
Read (2 of 5 actions)		
BatchGetBuilds	arn:aws:codebuild:us-east-2:123456789012:project/my-demo-project	None
BatchGetProjects	arn:aws:codebuild:us-east-2:123456789012:project/my-demo-project	None
Write (6 of 7 actions)		
BatchDeleteBuilds	arn:aws:codebuild:us-east-2:123456789012:project/my-demo-project	None
CreateProject	arn:aws:codebuild:us-east-2:123456789012:project/my-demo-project	None
DeleteProject	arn:aws:codebuild:us-east-2:123456789012:project/my-demo-project	None
StartBuild	arn:aws:codebuild:us-east-2:123456789012:project/my-demo-project	None
StopBuild	arn:aws:codebuild:us-east-2:123456789012:project/my-demo-project	None
UpdateProject	arn:aws:codebuild:us-east-2:123456789012:project/my-demo-project	None

CodeBuild_CodeCommit_CodeDeploy StartBuild (Write) 작업 요약:

Resource	Region	Account	Request condition
arn:aws:codebuild:us-east-2:123456789012:project/my-demo-project	us-east-2	123456789012	None

IAM 리소스에 액세스하는 데 필요한 권한

리소스는 서비스 내의 객체입니다. IAM에는 그룹, 사용자, 역할 및 정책이 있습니다. AWS 계정 루트 사용자 자격 증명으로 로그인한 경우 IAM 자격 증명 또는 IAM 리소스를 관리하는 데 아무런 제한이 없습니다. 하지만 IAM 사용자가 자격 증명이나 IAM 리소스를 관리하려면 그러한 권한이 명시적으로 부여되어야 합니다. 자격 증명 기반 정책을 사용자에 연결하여 권한을 부여할 수 있습니다.

Note

AWS 설명서 전체에서 특정 범주를 언급하지 않고 IAM 정책을 칭할 때는 자격 증명 기반 고객 관리형 정책을 의미합니다. 정책 범주에 대한 자세한 내용은 [the section called “정책 및 권한” \(p. 305\)](#) 단원을 참조하십시오.

IAM 자격 증명을 관리하기 위한 권한

IAM 그룹, 사용자, 역할 및 자격 증명을 관리하는 데 필요한 권한은 일반적으로 작업에 대한 API 작업에 해당합니다. 예를 들어 IAM 사용자를 생성하려면 해당하는 API 명령 `CreateUser`가 있는 `iam:CreateUser` 권

한이 있어야 합니다. IAM 사용자가 다른 IAM 사용자를 생성할 수 있도록 다음과 같은 IAM 정책을 해당 사용자에게 연결할 수 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iam:CreateUser",  
            "Resource": "*"  
        }  
    ]  
}
```

정책에서 Resource 요소의 값은 작업 및 그 작업이 적용될 수 있는 리소스에 따라 다릅니다. 앞의 예에서 정책은 사용자가 어떤 사용자도 생성할 수 있도록 허용합니다(*는 모든 문자열을 나타내는 와일드카드). 반면에, 사용자가 자신의 액세스 키(API 작업 [CreateAccessKey](#) 및 [UpdateAccessKey](#))만 변경할 수 있도록 하는 정책에는 일반적으로 Resource 요소가 포함됩니다. 이 경우 ARN에는 다음 예제와 같이 현재 사용자의 이름을 해석하는 변수(\${aws:username})가 포함됩니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "ListUsersForConsole",  
            "Effect": "Allow",  
            "Action": "iam>ListUsers",  
            "Resource": "arn:aws:iam::*:*"  
        },  
        {  
            "Sid": "ViewAndUpdateAccessKeys",  
            "Effect": "Allow",  
            "Action": [  
                "iam:UpdateAccessKey",  
                "iam>CreateAccessKey",  
                "iam>ListAccessKeys"  
            ],  
            "Resource": "arn:aws:iam::*:user/${aws:username}"  
        }  
    ]  
}
```

앞의 예에서 \${aws:username}은 현재 사용자의 사용자 이름으로 변환되는 변수입니다. 정책 변수에 대한 자세한 내용은 [IAM 정책 요소: 변수 및 태그 \(p. 524\)](#) 단원을 참조하십시오.

작업 이름에 와일드카드 문자(*)를 사용하면 특정 작업에 관련된 모든 작업에 대한 권한을 쉽게 부여할 수 있습니다. 예를 들어 사용자가 IAM 작업을 수행할 수 있게 하려면 그 작업에 대해 iam:*를 사용하면 됩니다. 사용자가 액세스 키에 관련된 작업만 수행할 수 있게 하려면 정책 문의 iam:AccessKey* 요소에 Action을 사용하면 됩니다. 이렇게 하면 사용자에게 [CreateAccessKey](#), [DeleteAccessKey](#), [GetAccessKeyLastUsed](#), [ListAccessKeys](#), [UpdateAccessKey](#) 작업을 수행할 수 있는 권한이 부여됩니다. (나중에 이름에 "AccessKey"가 포함되는 작업이 IAM에 추가될 경우에도 Action 요소에 대해 iam:AccessKey*를 사용하며 사용자에게 새 작업에 대한 권한이 부여됩니다.) 다음 예는 사용자가 자신의 액세스 키에 속하는 모든 작업을 수행할 수 있게 허용하는 정책을 보여 줍니다(ACCOUNT-ID-WITHOUT-HYPHENS를 해당 AWS 계정 ID로 변경).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iam:AccessKey*",  
            "Resource": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:user/${aws:username}"  
        }  
    ]  
}
```

}

그룹 삭제와 같은 일부 작업에는 여러 작업이 포함됩니다. 즉, 먼저 그룹에서 사용자를 제거한 후 그룹의 정책을 분리 또는 삭제하고 나서 실제로 그룹을 삭제합니다. 사용자가 그룹을 삭제할 수 있게 하려는 경우 이러한 모든 관련 작업을 수행할 수 있는 권한을 부여해야 합니다.

AWS Management 콘솔에서의 작업 권한

앞의 예는 사용자가 [AWS CLI](#) 또는 [AWS SDK](#)를 사용하여 작업을 수행할 수 있게 허용하는 정책을 보여 줍니다.

사용자가 콘솔에서 작업할 경우 콘솔은 그룹, 사용자, 역할 및 정책을 나열하고 그룹, 사용자 또는 역할과 연결된 정책을 가져오는 요청을 IAM에 보냅니다. 또한 콘솔은 AWS 계정 정보와 보안 주체에 대한 정보를 가져오는 요청도 보냅니다. 보안 주체는 콘솔에서 요청하는 사용자입니다.

일반적으로 작업을 수행하려면 일치하는 작업만 정책에 포함해야 합니다. 사용자를 생성하려면 [CreateUser](#) 작업을 호출하는 권한이 필요합니다. 콘솔을 사용하여 작업을 수행할 때 경우에 따라 콘솔에서 리소스를 표시하고 나열하며 가져오거나 볼 권리가 있어야 합니다. 이는 콘솔을 탐색하여 지정된 작업을 수행하기 위해 필요합니다. 예를 들어 Jorge라는 사용자가 콘솔을 사용하여 자신의 액세스 키를 변경하려면 IAM 콘솔에서 사용자를 선택할 것입니다. 이 작업은 콘솔에서 [ListUsers](#) 요청을 생성하게 합니다. Jorge에게 `iam>ListUsers` 작업에 대한 권한이 없을 경우 사용자를 나열하려고 시도할 때 콘솔이 액세스를 거부합니다. 그 결과, Jorge는 [CreateAccessKey](#) 및 [UpdateAccessKey](#) 작업에 대한 권한이 있는 경우에도 자신의 이름과 액세스 키를 가져올 수 없습니다.

예를 들어 Bob이라는 사용자가 콘솔을 사용하여 자신의 액세스 키를 변경하려면 IAM 콘솔에서 사용자를 선택할 것입니다. 이 작업은 콘솔에서 [ListUsers](#) 요청을 생성하게 합니다. Bob에게 `iam>ListUsers` 작업에 대한 권한이 없을 경우 콘솔은 사용자를 조회하려고 시도할 때 액세스를 거부당합니다. 따라서 Bob은 [CreateAccessKey](#) 및 [UpdateAccessKey](#) 작업에 대한 권한이 있을 경우에도 자신의 이름과 액세스 키를 받지 못합니다.

사용자에게 AWS Management 콘솔에서 그룹, 사용자, 역할, 정책, 자격 증명을 관리할 수 있는 권한을 부여하려면 콘솔에서 수행하는 작업에 대한 권한도 포함시켜야 합니다. 사용자에게 이러한 권한들을 부여하는 데 사용할 수 있는 몇 가지 정책의 예를 보려면 [IAM 리소스를 관리하기 위한 정책의 예 \(p. 446\)](#) 단원을 참조하십시오.

전 AWS 계정에 권한 부여

계정의 IAM 사용자에게 리소스에 대한 액세스 권한을 직접 부여할 수 있습니다. 다른 계정의 사용자에게 리소스에 대한 액세스 권한이 필요한 경우, 권한을 포함하지만 특정 사용자와 연결되지 않는 엔터티인 IAM 역할을 만들 수 있습니다. 다른 계정의 사용자는 해당 역할을 사용하여 해당 역할에 할당된 권한에 따라 리소스에 액세스할 수 있습니다. 자세한 내용은 [자신이 소유한 다른 AWS 계정의 IAM 사용자에 대한 액세스 권한 제공 \(p. 157\)](#)을 참조하십시오.

Note

일부 서비스는 [자격 증명 기반 정책 및 리소스 기반 정책 \(p. 326\)](#)에 나온 것처럼 리소스 기반 정책을 지원합니다(Amazon S3, Amazon SNS, Amazon SQS 등). 그런 서비스의 역할 사용 대안은 공유 할 리소스(버킷, 주제 또는 대기열)에 정책을 연결하는 것입니다. 리소스 기반 정책은 리소스에 대한 액세스 허가를 받은 AWS 계정을 지정할 수 있습니다.

한 서비스에서 다른 서비스에 액세스할 권한

많은 AWS 제품은 다른 AWS 제품에 액세스합니다. 예를 들어, 어떤 AWS 제품(Amazon EMR, Elastic Load Balancing, Amazon EC2 Auto Scaling 등)은 Amazon EC2 인스턴스를 관리하고 다른 AWS 제품은 Amazon S3 버킷, Amazon SNS 주제, Amazon SQS 대기열 등을 사용합니다.

이러한 경우 권한 관리 시나리오가 서비스별로 다릅니다. 다음은 다양한 서비스에 대한 권한을 처리하는 방법의 예입니다.

- Amazon EC2 Auto Scaling에서 사용자는 Auto Scaling을 사용할 권한이 있어야 하지만, 이 사용자에게 Amazon EC2 인스턴스를 관리할 권한을 명시적으로 부여할 필요는 없습니다.
- AWS Data Pipeline에서 IAM 역할은 파이프라인에서 수행할 수 있는 작업을 결정합니다. 또한 사용자에게는 해당 역할을 수임할 권리가 필요합니다. (자세한 내용은 AWS Data Pipeline 개발자 안내서의 [Granting Permissions to Pipelines with IAM](#) 단원을 참조하십시오.)

AWS 제품에서 원하는 작업을 수행할 수 있도록 권한을 적절히 구성하는 방법에 대한 자세한 내용은 요청할 서비스 설명서를 참조하십시오. 서비스에 대한 역할을 생성하는 방법에 대해 알아보려면 [AWS 서비스에 대한 권한을 위임할 역할 생성 \(p. 210\)](#) 단원을 참조하십시오.

사용자를 대신하여 작동하도록 IAM 역할로 서비스 구성

AWS 서비스를 사용자를 대신하여 작동하도록 구성하려면 일반적으로 서비스에서 수행할 수 있는 작업을 정의하는 IAM 역할의 ARN을 입력합니다. AWS는 사용자에게 서비스에 역할을 전달할 권리가 있는지 확인합니다. 자세한 내용은 [사용자에게 AWS 서비스에 역할을 전달할 권한 부여 \(p. 231\)](#)을 참조하십시오.

필수 작업

작업은 리소스 보기, 생성, 편집 및 삭제와 같이 리소스에 대해 수행할 수 있는 사항입니다. 작업은 각 AWS 서비스별로 정의됩니다.

누군가 작업을 수행할 수 있도록 허용하려면 호출 자격 증명 또는 영향을 받은 리소스에 적용되는 정책에 필요한 작업을 포함시켜야 합니다. 일반적으로 작업을 수행하는 데 필요한 권한을 제공하려면 정책에 해당 작업을 포함시켜야 합니다. 예를 들어 사용자를 생성하려면 정책에 CreateUser 작업을 추가해야 합니다.

경우에 따라 정책에 관련된 작업을 추가로 포함해야 할 수도 있습니다. 예를 들어, `ds:CreateDirectory` 작업을 사용하여 AWS Directory Service에서 누군가에게 디렉터리를 생성할 권한을 제공하려면 정책에 다음 작업을 포함시켜야 합니다.

- `ds:CreateDirectory`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcs`
- `ec2:CreateSecurityGroup`
- `ec2:CreateNetworkInterface`
- `ec2:DescribeNetworkInterfaces`
- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:AuthorizeSecurityGroupEgress`

시각적 편집기를 사용하여 정책을 생성하거나 편집할 때 경고 및 정책에 필요한 모든 작업을 선택하라는 메시지가 표시됩니다.

AWS Directory Service에서 디렉터리를 생성하는 데 필요한 권한에 대한 자세한 내용은 [예제 2: 사용자에게 디렉터리 생성 허용](#)을 참조하십시오.

IAM 리소스를 관리하기 위한 정책의 예

다음은 사용자가 IAM 사용자, 그룹 및 자격 증명을 관리하기 위한 작업을 수행할 수 있게 하는 IAM 정책의 예시입니다. 여기에는 사용자가 자신의 암호, 액세스 키 및 멀티 팩터 인증(MFA) 디바이스를 관리할 수 있게 하는 정책이 포함됩니다.

사용자가 다른 AWS 제품(Amazon S3, Amazon EC2, DynamoDB 등)으로 작업을 수행할 수 있도록 허용하는 예제 정책은 [IAM 자격 증명 기반 정책 예제 \(p. 341\)](#) 단원을 참조하십시오.

주제

- 사용자가 보고를 목적으로 계정의 그룹, 사용자, 정책 및 그 이상의 정보를 조회할 수 있도록 허용 (p. 447)
- 사용자가 그룹의 멤버십을 관리할 수 있도록 허용 (p. 447)
- IAM 사용자를 관리할 수 있도록 허용 (p. 448)
- 사용자가 계정 암호 정책을 설정할 수 있도록 허용 (p. 449)
- 사용자가 IAM 자격 증명 보고서를 생성하고 검색할 수 있도록 허용 (p. 449)
- 모든 IAM 작업 허용 (관리자 액세스 권한) (p. 449)

사용자가 보고를 목적으로 계정의 그룹, 사용자, 정책 및 그 이상의 정보를 조회할 수 있도록 허용

다음 정책은 사용자가 Get 또는 List 문자열로 시작하는 모든 IAM 작업을 호출할 수 있게 허용합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": [  
            "iam:Get*",  
            "iam>List*"  
        ],  
        "Resource": "*"  
    }  
}
```

Get* 및 List* 작업을 사용하면 Get 및 List로 시작하는 모든 기존 및 향후 IAM 작업이 허용됩니다. 예를 들어, IAM이 새 위젯 리소스를 추가하는 경우 이 정책은 GetWidget 및 ListWidgets 작업을 허용합니다.

사용자가 그룹의 멤버십을 관리할 수 있도록 허용

다음 정책은 사용자가 MarketingGroup이라는 그룹의 멤버십을 업데이트할 수 있도록 허용합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "ViewGroups",  
            "Effect": "Allow",  
            "Action": [  
                "iam>ListGroups",  
                "iam GetUser",  
                "iam>ListGroupsForUser"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Sid": "ViewEditThisGroup",  
            "Effect": "Allow",  
            "Action": [  
                "iam>AddUserToGroup",  
                "iam RemoveUserFromGroup",  
                "iam GetGroup"  
            ],  
            "Resource": "arn:aws:iam::*:group/MarketingGroup"  
        }  
    ]  
}
```

}

IAM 사용자를 관리할 수 있도록 허용

다음 정책은 사용자가 IAM 사용자 관리와 관련된 모든 작업을 수행할 수 있게 허용하지만 그룹이나 정책의 생성과 같은 다른 엔터티에 대한 작업 수행은 허용하지 않습니다. 허용되는 작업은 다음과 같습니다.

- 사용자 생성([CreateUser](#) 작업).
- 사용자 삭제. 이 작업은 [DeleteSigningCertificate](#), [DeleteLoginProfile](#), [RemoveUserFromGroup](#), [DeleteUser](#) 작업 모두를 수행할 수 있는 권한이 필요합니다.
- 계정 및 그룹의 사용자 조회([GetUser](#), [ListUsers](#), [ListGroupsForUser](#) 작업).
- 사용자의 정책 조회 및 제거([ListUserPolicies](#), [ListAttachedUserPolicies](#), [DetachUserPolicy](#), [DeleteUserPolicy](#) 작업).
- 사용자의 경로 이름 바꾸기 또는 변경([UpdateUser](#) 작업). Resource 요소에는 소스 경로와 대상 경로를 모두 다루는 ARN이 포함되어야 합니다. 경로에 대한 자세한 내용은 [표시 이름 및 경로 \(p. 480\)](#) 단원을 참조하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowUsersToPerformUserActions",  
            "Effect": "Allow",  
            "Action": [  
                "iam>ListPolicies",  
                "iam:GetPolicy",  
                "iam:UpdateUser",  
                "iam:AttachUserPolicy",  
                "iam>ListEntitiesForPolicy",  
                "iam>DeleteUserPolicy",  
                "iam>DeleteUser",  
                "iam>ListUserPolicies",  
                "iam>CreateUser",  
                "iam>RemoveUserFromGroup",  
                "iam>AddUserToGroup",  
                "iam:GetUserPolicy",  
                "iam>ListGroupsForUser",  
                "iam>PutUserPolicy",  
                "iam>ListAttachedUserPolicies",  
                "iam>ListUsers",  
                "iam> GetUser",  
                "iam>DetachUserPolicy"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Sid": "AllowUsersToSeeStatsOnIAMConsoleDashboard",  
            "Effect": "Allow",  
            "Action": [  
                "iam>GetAccount*",  
                "iam>ListAccount*"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

위의 정책에 포함된 많은 권한은 사용자가 AWS Management 콘솔에서 작업을 수행하도록 허용합니다. 사용자가 [AWS CLI](#), [AWS SDK](#) 또는 IAM HTTP 쿼리 API를 사용하여 사용자 관련 작업을 수행할 경우 특정 권

한이 필요하지 않을 수 있습니다. 예를 들어 사용자가 어떤 사용자에게서 연결을 해제할 정책의 ARN을 이미 알고 있다면 `iam>ListAttachedUserPolicies` 권한이 필요하지 않습니다. 사용자에게 필요한 권한의 정확한 목록은 사용자가 다른 사용자를 관리할 때 수행해야 하는 작업에 따라 다릅니다.

정책에 있는 다음 권한들은 AWS Management 콘솔을 통해 사용자 작업에 액세스할 수 있도록 허용합니다.

- `iam:GetAccount*`
- `iam>ListAccount*`

사용자가 계정 암호 정책을 설정할 수 있도록 허용

일부 사용자들에게 AWS 계정의 암호 정책 ([p. 79](#))을 확인하고 업데이트할 수 있는 권한을 부여할 수도 있습니다. 다음 예시와 같은 정책은 이러한 권한들을 부여합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": [  
            "iam:GetAccountPasswordPolicy",  
            "iam:UpdateAccountPasswordPolicy"  
        ],  
        "Resource": "*"  
    }  
}
```

사용자가 IAM 자격 증명 보고서를 생성하고 검색할 수 있도록 허용

AWS 계정에 있는 모든 사용자가 나열된 보고서를 생성하고 다운로드할 수 있는 권한을 사용자에게 부여할 수 있습니다. 이 보고서에는 암호, 액세스 키, MFA 디바이스, 서명 인증서를 포함한 다양한 사용자 자격 증명의 상태도 나열됩니다. 다음 예시와 같은 정책은 이러한 권한들을 부여합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": [  
            "iam:GenerateCredentialReport",  
            "iam:GetCredentialReport"  
        ],  
        "Resource": "*"  
    }  
}
```

자격 증명 보고서에 대한 자세한 내용은 [AWS 계정의 자격 증명 보고서 가져오기 \(p. 135\)](#) 단원을 참조하십시오.

모든 IAM 작업 허용 (관리자 액세스 권한)

일부 사용자들에게 암호, 액세스 키, MFA 디바이스, 사용자 인증서 관리를 비롯하여 IAM에서의 모든 작업을 수행할 수 있는 권한을 부여할 수 있습니다. 다음 예시와 같은 정책은 이러한 권한들을 부여합니다.

Warning

사용자에게 IAM에 대한 모든 액세스 권한을 부여하면 해당 사용자가 자기 자신 또는 다른 사용자에게 부여할 수 있는 권한에 제한이 없습니다. 사용자는 새로운 IAM 엔터티(사용자나 역할)을 생성하

고 그러한 엔터티에 AWS 계정의 모든 리소스에 대한 모든 액세스 권한을 부여할 수 있습니다. 사용자에게 IAM에 대한 모든 액세스 권한을 부여하면 실제로 사용자들에게 AWS 계정의 모든 리소스에 대한 모든 액세스 권한을 부여하는 것입니다. 여기에는 모든 리소스를 삭제하는 권한도 포함됩니다. 이러한 권한은 신뢰할 수 있는 관리자에게만 부여해야 하며, 그러한 관리자에 대해 멀티 팩터 인증(MFA)을 적용해야 합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": "iam:*",  
         "Resource": "*"}  
    ]  
}
```

IAM 문제 해결

AWS Identity and Access Management(IAM) 작업 시 액세스 거부 또는 이와 유사한 문제가 발생하면 이 섹션의 주제를 참조하십시오.

주제

- [일반적인 문제 해결 \(p. 451\)](#)
- [IAM 정책 문제 해결 \(p. 454\)](#)
- [U2F 보안 키 문제 해결 \(p. 468\)](#)
- [IAM 역할 문제 해결 \(p. 469\)](#)
- [Amazon EC2 및 IAM 문제 해결 \(p. 472\)](#)
- [Amazon S3 및 IAM 문제 해결 \(p. 474\)](#)
- [AWS로 SAML 2.0 연동 문제 해결 \(p. 475\)](#)

일반적인 문제 해결

이 문서의 정보를 사용하여 AWS Identity and Access Management(IAM) 작업 시 발생할 수 있는 액세스 거부 또는 기타 일반적인 문제를 진단하고 해결할 수 있습니다.

주제

- [액세스 키를 분실했습니다 \(p. 451\)](#)
- [예전 계정에 액세스해야 합니다 \(p. 451\)](#)
- [내 계정에 로그인할 수 없음 \(p. 452\)](#)
- [AWS 서비스에 요청하면 "액세스 거부"가 발생합니다 \(p. 452\)](#)
- [임시 보안 자격 증명으로 요청하면 "액세스 거부"가 발생합니다 \(p. 453\)](#)
- [정책 변수가 작동하지 않습니다 \(p. 453\)](#)
- [변경 사항이 매번 즉시 표시되는 것은 아닙니다 \(p. 454\)](#)

액세스 키를 분실했습니다

액세스 키는 다음 두 부분으로 구성됩니다.

- 액세스 키 식별자. 이 식별자는 비밀이 아니며 사용자 요약 페이지 등 IAM 콘솔에서 액세스 키가 나열되어 있는 곳 어디서나 확인할 수 있습니다.
- 보안 액세스 키. 액세스 키 페어를 처음 만들 때 제공됩니다. 암호와 마찬가지로 나중에 검색할 수 없습니다. 하지만 보안 액세스 키를 분실한 경우에는 새로운 액세스 키 페어를 생성해야 합니다. 이미 [액세스 키의 최대 수 \(p. 485\)](#)인 경우 기존 페어를 삭제해야만 다른 페어를 생성할 수 있습니다.

자세한 정보는 [분실하거나 잊어버린 암호 또는 액세스 키 재설정 \(p. 95\)](#) 단원을 참조하십시오.

예전 계정에 액세스해야 합니다

AWS 계정을 처음 생성할 때 이메일 주소와 암호를 입력했습니다. 그 주소와 암호가 바로 AWS 계정 루트 사용자 자격 증명입니다. 암호를 잊거나 분실하여 더 이상 액세스할 수 없게 된 AWS 계정이 있다면 암호를 복

구하면 됩니다. 자세한 정보는 [분실하거나 잊어버린 암호 또는 액세스 키 재설정 \(p. 95\)](#) 단원을 참조하십시오.

그 이메일에 더 이상 액세스할 수 없는 경우, 먼저 이메일에 대한 액세스부터 복구해 보아야 합니다. 복구에 실패하면 AWS 고객 서비스에 문의하십시오.

다음 옵션 중 하나를 사용하여 이메일에 대한 액세스 권한을 복구해 볼 수 있습니다.

- 이메일 주소의 호스팅 도메인이 본인 소유인 경우, 그 이메일 주소를 다시 이메일 서버에 추가하면 됩니다. 아니면 이메일 계정에 대한 완전 포착(catch-all) 조건을 설정할 수 있습니다. 도메인에 설정된 완전 포착(catch-all) 조건은 메일 서버에 존재하지 않는 이메일 주소로 전송된 "모든 메시지를 포착(catches all)"입니다. 그리고 그러한 메시지를 특정 이메일 주소로 리디렉션합니다. 예를 들어, AWS 계정 루트 사용자 이메일 주소가 paulo@sample-domain.com인데 도메인 이메일 주소만 paulo.santos@sample-domain.com으로 변경했다면 새 이메일을 catch-all 대상으로 설정할 수 있습니다. 그러면 AWS 같은 곳에서 paulo@sample-domain.com 또는 다른 text@sample-domain.com으로 메시지를 보내면 사용자는 paulo.santos@sample-domain.com 주소로 그 메시지를 받게 됩니다.
- 계정의 이메일 주소가 회사 이메일 시스템에 속한 경우라면 IT 시스템 관리자에게 문의하는 것이 좋습니다. 시스템 관리자가 이메일 주소에 대한 액세스 권한을 다시 받을 수 있도록 도와 줄 것입니다.

그래도 AWS 계정에 액세스할 수 없는 경우, [문의처](#)에서 AWS를 이용하고 있으며, 결제 혹은 계정 지원이 필요합니다(I'm an AWS customer and I'm looking for billing or account support) 메뉴를 확장하여 다른 지원 옵션을 찾아볼 수 있습니다. 고객 서비스에 문의할 때 다음 정보를 알려 주십시오.

- 본인 이름, 전화번호, 주소, 이메일 주소, 신용카드의 마지막 네 자리 번호 등 계정에 나열되어 있는 모든 세부 정보. 고객 서비스 문의를 위해 AWS 계정을 새로 만들어야 할 수도 있지만 이것은 요청 내용을 조사하기 위해 필요한 과정입니다.
- 암호 재설정 지침을 받아야 하는데 이메일 계정에 액세스할 수 없는 이유.
- 지원 팀에 사용하지 않는 계정은 모두 삭제해 달라고 요청하십시오. 요금이 부과될 가능성이 있으므로 본인 이름으로 계정을 열어 두지 않는 것이 좋습니다.

내 계정에 로그인할 수 없음

AWS 계정을 처음 생성할 때 이메일 주소와 암호를 입력했습니다. 그 주소와 암호가 바로 AWS 계정 루트 사용자 자격 증명입니다. 암호를 잊거나 분실하여 더 이상 액세스할 수 없게 된 계정이 있다면 해당 암호를 복구하면 됩니다. 자세한 정보는 [분실하거나 잊어버린 암호 또는 액세스 키 재설정 \(p. 95\)](#) 단원을 참조하십시오.

계정 이메일 주소와 암호를 입력한 경우 AWS는 일회성 확인 코드를 입력해야 하는 경우도 있습니다. 확인 코드를 검색하려면 AWS 계정과 연결된 이메일에서 Amazon Web Services의 메시지를 확인합니다. 이메일 주소는 @amazon.com 또는 @aws.amazon.com으로 끝납니다. 메시지의 지침을 따릅니다. 계정으로 메시지가 오지 않았으면 스팸 폴더를 점검합니다. 그 이메일에 더 이상 액세스할 수 없는 경우에는 [예전 계정에 액세스해야 합니다 \(p. 451\)](#) 단원을 참조하십시오.

AWS 서비스에 요청하면 "액세스 거부"가 발생합니다

- 요청한 작업 및 리소스를 호출할 자격 증명 기반 정책 권한이 있는지 확인합니다. 조건이 설정된 경우 요청을 보낼 때 그러한 조건 또한 충족해야 합니다. IAM 사용자, 그룹 또는 역할에 대한 정책을 보거나 수정하는 방법에 대한 자세한 정보는 [IAM 정책 관리 \(p. 377\)](#) 단원을 참조하십시오.
- Amazon S3, Amazon SNS 또는 Amazon SQS 등 [리소스 기반 정책 \(p. 326\)](#)을 지원하는 서비스에 액세스하려는 경우, 해당 리소스 정책에서 자신을 보안 주체로 지정하고 액세스 권한을 부여했는지 확인합니다. 자신의 계정 내에서 서비스를 요청하는 경우 자격 증명 기반 정책이나 리소스 기반 정책에서 요청자에게 권한을 부여할 수 있습니다. 다른 계정에서 서비스를 요청하는 경우 자격 증명 기반 정책 및 리소스 기반 정책 모두에서 요청자에게 권한을 부여해야 합니다. 리소스 기반 정책을 지원하는 서비스를 보려면 [IAM으로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하십시오.

- 정책에 키-값 페어가 있는 조건이 포함된 경우 이를 주의하여 검토하십시오. [aws:RequestTag/tag-key](#) (p. 551) 전역 조건 키, AWS KMS [kms:EncryptionContext:encryption_context_key](#) 및 여러 서비스에서 지원하는 [ResourceTag>tag-key](#) 조건 키가 그 예입니다. 키 이름이 여러 개의 결과와 일치하지 않도록 하십시오. 조건 키 이름이 대/소문자를 구분하지 않으므로 이름이 foo인 키를 검사하는 조건은 foo, Foo 또는 FOO와 일치합니다. 대/소문자로만 구분되는 키 이름을 가진 여러 키-값 페어가 요청에 포함된 경우 액세스가 예기치 않게 거부될 수 있습니다. 자세한 정보는 [IAM JSON 정책 요소: Condition](#) (p. 510) 단원을 참조하십시오.
- [권한 경계](#) (p. 317)가 있다면, 권한 경계에 사용된 정책이 요청을 허용하는지 확인합니다. 자격 증명 기반 정책에서는 요청이 허용되지만 권한 경계에서는 허용되지 않는 경우 요청이 거부됩니다. 이 권한 경계는 보안 주체 엔터티(사용자나 역할)에 부여할 수 있는 최대 권한을 제어합니다. 리소스 기반 정책은 권한 경계에 제한을 받지 않습니다. 권한 경계는 일반적이지 않습니다. AWS 평가 정책에 대한 자세한 정보는 [정책 평가 로직](#) (p. 531)을 참조하십시오.
- (AWS SDK를 사용하지 않고) 요청에 수동으로 서명할 경우, 요청에 올바르게 서명했는지 확인합니다.

임시 보안 자격 증명으로 요청하면 "액세스 거부"가 발생합니다

- 우선, 임시 자격 증명과 무관한 이유로 액세스가 거부되지는 않았는지 확인합니다. 자세한 정보는 [AWS 서비스에 요청하면 "액세스 거부"가 발생합니다](#) (p. 452) 단원을 참조하십시오.
- [IAM로 작업하는 AWS 서비스](#) (p. 488) 단원을 참조하여 서비스에서 임시 보안 자격 증명을 허용하는지 확인합니다.
- 요청에 올바르게 서명했고 요청이 잘 구성되었는지 확인합니다. 자세한 정보는 [도구 키트](#) 문서 또는 [임시 보안 자격 증명을 사용해 AWS 리소스에 대한 액세스 요청하기](#) (p. 274) 단원을 참조하십시오.
- 임시 보안 자격 증명이 만료되지 않았는지 확인합니다. 자세한 정보는 [임시 보안 자격 증명](#) (p. 263) 단원을 참조하십시오.
- IAM 사용자 또는 역할 권한이 올바른지 확인합니다. 임시 보안 자격 증명에 대한 권한은 IAM 사용자 또는 역할에서 파생됩니다. 결과적으로 위임한 역할(임시 자격 증명이 제공됨)에 부여된 권한으로 제한됩니다. 임시 보안 자격 증명의 권한이 결정되는 방법에 대한 자세한 정보는 [사용자 임시 보안 자격 증명에 대한 권한 제어](#) (p. 277) 단원을 참조하십시오.
- 역할을 사용하여 리소스 기반 정책이 있는 리소스에 액세스할 경우, 해당 정책에서 역할에 권한을 부여하는지 확인합니다. 예를 들어, 다음과 같은 정책에서는 계정 MyRole의 111122223333이 MyBucket에 액세스하도록 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3BucketPolicy",
      "Effect": "Allow",
      "Principal": {"AWS": ["arn:aws:iam::111122223333:role/MyRole"]},
      "Action": ["s3:PutObject"],
      "Resource": ["arn:aws:s3:::MyBucket/*"]
    }
  ]
}
```

정책 변수가 작동하지 않습니다

- 변수가 포함된 모든 정책에 버전 번호 "Version": "2012-10-17"이 있는지 확인하십시오. 올바른 버전 번호가 없으면 평가 도중에 변수가 대체되지 않습니다. 대신 변수는 문자 그대로 평가됩니다. 최신 버전 번호를 포함시키더라도 변수를 포함하지 않은 정책은 계속 작동합니다.

Version 정책 요소는 정책 버전과 다릅니다. Version 정책 요소는 정책 내에서 사용되며 정책 언어의 버전을 정의합니다. 반면에 정책 버전은 IAM에서 고객 관리형 정책을 변경할 때 생성됩니다. 변경된 정책은

기존 정책을 덮어쓰지 않습니다. 대신 IAM에서 관리형 정책의 새 버전을 만듭니다. Version 정책 요소에 대한 자세한 정보는 [IAM JSON 정책 요소: Version \(p. 499\)](#)을 참조하십시오. 정책 버전에 대한 자세한 정보는 [the section called "IAM 정책 버전 관리" \(p. 399\)](#) 단원을 참조하십시오.

- 정책 변수가 올바른지 확인합니다. 자세한 정보는 [IAM 정책 요소: 변수 및 태그 \(p. 524\)](#) 단원을 참조하십시오.

변경 사항이 매번 즉시 표시되는 것은 아닙니다

사용자들이 전세계 데이터 센터의 컴퓨터들을 통해 액세스하는 서비스인 IAM은 **최종 일관성**이라고 하는 분산 컴퓨팅 모델을 사용합니다. IAM(또는 다른 AWS 서비스)에서 변경한 사항을, 있을 수 있는 모든 엔드포인트에서 보게 될 때까지는 시간이 걸립니다. 일부 지역은 서버에서 서버로, 복제 영역에서 복제 영역으로, 전 세계의 리전에서 리전으로 데이터를 보내는 데 걸리는 시간으로 인해 발생합니다. 또한 IAM은 캐싱을 사용하여 성능을 개선하지만 이 경우 종 몇몇 경우는 더 많은 시간이 소요될 수 있습니다. 이전에 캐시된 데이터가 시간 초과될 때까지 변경 사항이 표시되지 않을 수 있기 때문입니다.

이러한 잠재적 지연을 고려하도록 전역 애플리케이션을 설계해야 합니다. 한 위치에서 변경한 내용이 다른 위치에서 즉시 보이지 않을 때조차도 예상대로 작동하는지 확인합니다. 그러한 변경 사항에는 사용자, 그룹, 역할 또는 정책을 만들거나 업데이트한 것이 포함됩니다. 그러한 IAM 변경 사항을 애플리케이션의 중요한 고가용성 코드 경로에 포함시키지 않는 것이 좋습니다. 대신 자주 실행하지 않는 별도의 초기화 루틴이나 설정 루틴에서 IAM을 변경하십시오. 또한 프로덕션 워크플로우에서 변경 사항을 적용하기 전에 변경 사항이 전파되었는지 확인하십시오.

이로 인해 일부 다른 AWS 서비스가 받게 되는 영향에 대한 자세한 정보는 다음 자료를 참고하십시오.

- Amazon DynamoDB: DynamoDB FAQ에서 [Amazon DynamoDB의 일관성 모델이란 무엇입니까?](#) 및 [Amazon DynamoDB 개발자 안내서에서의 읽기 일관성이란 무엇입니까?](#)
- Amazon EC2: Amazon EC2 API Reference에서의 [EC2 최종 일관성](#).
- Amazon EMR: 빅 데이터 블로그에서 [AWSETL 워크플로우에 대해 Amazon S3 및 Amazon Elastic MapReduce 사용 시 일관성 유지](#)
- Amazon Redshift: Amazon Redshift Database Developer Guide에서 [데이터 일관성 관리](#)
- Amazon S3: Amazon Simple Storage Service 개발자 가이드에서의 [Amazon S3 데이터 일관성 모델](#)

IAM 정책 문제 해결

정책 ([p. 305](#))은 자격 증명 또는 리소스에 연결될 때 해당 권한을 정의하는 AWS의 엔터티입니다. AWS는 사용자와 같은 보안 주체가 요청할 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는지 여부를 결정합니다. 정책은 JSON 문서로 AWS에 저장되며 자격 증명 기반 정책으로 보안 주체에 연결되거나 리소스 기반 정책으로 리소스에 연결됩니다. 자격 증명 기반 정책은 IAM 그룹, 사용자 또는 역할과 같은 보안 주체(또는 자격 증명)에 연결할 수 있습니다. 자격 증명 기반 정책에는 AWS 관리형 정책, 고객 관리형 정책 및 인라인 정책이 포함됩니다. AWS Management 콘솔에서 Visual editor(시각적 편집기) 탭 또는 JSON 탭을 통해 고객 관리형 정책을 생성하고 편집할 수 있습니다. AWS Management 콘솔에서 정책을 볼 때 정책에서 부여된 권한의 요약을 볼 수 있습니다. 시각적 편집기 및 정책 요약을 사용하여 IAM 정책을 관리하는 동안 발생한 일반 오류를 진단하고 해결할 수 있습니다.

모든 IAM 정책은 [JavaScript Object Notation\(JSON\)](#) 규칙으로 시작하는 구문을 사용하여 저장됩니다. 정책을 생성 또는 관리하기 위해 이 구문을 이해할 필요가 없습니다. AWS Management 콘솔에서 시각적 편집기를 사용하여 정책을 생성하고 편집할 수 있습니다. IAM 정책의 JSON 구문에 대한 자세한 정보는 [IAM JSON 정책 언어의 문법 \(p. 538\)](#) 단원을 참조하십시오.

IAM 정책 주제 문제 해결

- [시각적 편집기를 사용하여 문제 해결 \(p. 455\)](#)
- [정책 재구성 \(p. 455\)](#)

- 시각적 편집기에서 리소스 ARN 선택 (p. 456)
- 시각적 편집기에서 권한 거부 (p. 456)
- 시각적 편집기에서 여러 서비스 지정 (p. 456)
- 시각적 편집기에서 정책의 크기 줄이기 (p. 457)
- 시각적 편집기에서 인식할 수 없는 서비스, 작업 또는 리소스 유형 수정 (p. 457)
- 정책 요약을 사용하여 문제 해결 (p. 458)
 - 정책 요약 누락 (p. 458)
 - 정책 요약에 인식할 수 없는 서비스, 작업 또는 리소스 유형 포함됨 (p. 459)
- 서비스가 IAM 정책 요약을 지원하지 않음 (p. 459)
- 정책이 필요한 권한을 부여하지 않음 (p. 460)
- 정책 관리 문제 해결 (p. 464)
 - IAM 계정에서 정책 연결 또는 분리 (p. 464)
 - 작업 기반 IAM 자격 증명 관련 정책 변경 (p. 464)
- JSON 정책 문서 문제 해결 (p. 464)
 - JSON 정책 객체가 둘 이상인 경우 (p. 464)
 - JSON Statement 요소가 둘 이상인 경우 (p. 465)
 - JSON Statement 요소의 Effect, Action 또는 Resource 요소가 둘 이상인 경우 (p. 466)
 - JSON 버전 요소 누락 (p. 467)

시각적 편집기를 사용하여 문제 해결

고객 관리형 정책을 생성 또는 편집할 때 Visual editor(시각적 편집기) 탭의 정보를 사용하여 정책의 오류를 해결할 수 있습니다. 시각적 편집기를 사용하여 정책을 생성하는 예제를 보려면 [the section called “자격 증명에 대한 액세스 제어” \(p. 329\)](#) 단원을 참조하십시오.

정책 재구성

정책을 생성할 때 AWS는 정책을 검증, 처리 및 변환한 후 저장합니다. AWS가 사용자 쿼리에 응답하여 정책을 반환하거나 콘솔에 표시할 경우 AWS는 정책에서 부여한 권한을 변경하지 않고 해당 정책을 사람이 읽을 수 있는 형식으로 다시 변환합니다. 이렇게 하면 정책 시각적 편집기 또는 JSON 탭에 표시되는 사항이 달라질 수 있습니다. 시각적 편집기 권한 블록이 추가, 제거 또는 재정렬될 수 있으며 블록 내의 내용이 최적화될 수 있습니다. JSON 탭에서 사소한 공백은 제거되며, JSON 맵 내의 요소는 재정렬될 수 있습니다. 또한 보안 주체 요소 내의 AWS 계정 ID는 AWS 계정 루트 사용자의 ARN으로 교체할 수 있습니다. 이러한 변경 가능성 때문에 JSON 정책 문서를 문자열로 비교하면 안 됩니다.

AWS Management 콘솔에서 고객 관리형 정책을 생성할 때 JSON 탭에서 완전히 작업하도록 선택할 수 있습니다. Visual editor(시각적 편집기) 탭에서 변경을 수행하지 않고 JSON 탭에서 정책 검토를 선택하면 정책을 재구성할 가능성이 적습니다. 그러나 정책을 생성하고 Visual editor(시각적 편집기) 탭을 사용하여 수정한 경우 또는 Visual editor(시각적 편집기) 탭에서 정책 검토를 선택한 경우 IAM은 시각적 편집기에서 모양을 최적화하기 위해 정책을 재구성할 수 있습니다.

이러한 재구성은 편집 세션에만 존재하며 자동으로 저장되지 않습니다.

편집 세션에서 정책이 재구성되면 IAM이 다음 상황에 따라 재구성을 저장할지 여부를 결정합니다.

이 탭에서	정책을 편집한 경우	그런 다음 이 탭에서 정책 검토를 선택한 경우	변경 사항 저장을 선택한 경우
Visual editor(시각적 편집기)	편집됨	Visual editor(시각적 편집기)	정책이 재구성됨

이 탭에서	정책을 편집한 경우	그런 다음 이 탭에서 정책 검토를 선택한 경우	변경 사항 저장을 선택한 경우
Visual editor(시각적 편집기)	편집됨	JSON	정책이 재구성됨
Visual editor(시각적 편집기)	편집되지 않음	Visual editor(시각적 편집기)	정책이 재구성됨
JSON	편집됨	Visual editor(시각적 편집기)	정책이 재구성됨
JSON	편집됨	JSON	정책 구성이 변경되지 않음
JSON	편집되지 않음	JSON	정책 구성이 변경되지 않음

IAM은 여러 서비스, 리소스 유형 또는 조건 키를 허용하는 문이나 권한 블록이 있는 정책 또는 복잡한 정책을 재구성할 수 있습니다.

시각적 편집기에서 리소스 ARN 선택

시각적 편집기를 사용하여 정책을 생성하거나 편집할 때 먼저 서비스를 선택한 다음 해당 서비스에서 작업을 선택해야 합니다. 선택한 서비스 및 작업이 [특정 리소스 \(p. 335\)](#) 선택을 지원하는 경우에는 시각적 편집기에 지원되는 리소스 유형이 나열됩니다. 그런 다음 Add ARN(ARN 추가)를 선택하여 리소스에 대한 세부 정보를 제공합니다. 리소스 유형에 대한 ARN을 추가하기 위해 다음 옵션에서 선택할 수 있습니다.

- ARN 빌더 사용 – 리소스 유형에 따라 ARN을 빌드하는 여러 필드가 표시될 수 있습니다. 모두 선택을 선택하여 지정된 설정의 값에 대한 권한을 제공할 수도 있습니다. 예를 들어, Amazon EC2 읽기 액세스 레벨 그룹을 선택하면 정책의 작업이 instance 리소스 유형을 지원합니다. 리소스에 대해 리전, 계정 및 InstanceId 값을 제공해야 합니다. 계정 ID를 제공하지만 리전 및 인스턴스 ID에 대해 모두 선택을 선택한 경우 정책은 계정의 모든 인스턴스에 대해 권한을 부여합니다.
- ARN 입력 또는 붙여넣기 – [Amazon 리소스 이름\(ARN\)](#) 별로 리소스를 지정할 수 있습니다. ARN의 필드(각 콜론 쌍 사이)에 와일드카드 문자(*)를 포함할 수 있습니다. 자세한 정보는 [IAM JSON 정책 요소: Resource \(p. 508\)](#) 단원을 참조하십시오.

시각적 편집기에서 권한 거부

기본적으로 시각적 편집기를 사용하여 생성하는 정책은 사용자가 선택하는 작업을 허용합니다. 대신 선택한 작업을 거부하려면 Switch to deny permissions(권한 거부로 전환)을 선택합니다. 요청은 기본적으로 거부되므로 사용자에게 필요한 작업과 리소스에만 권한을 허용하는 것이 보안 모범 사례입니다. 이것을 "화이트리스트"라고 부르기도 합니다. 다른 문이나 정책에서 허용되는 권한을 별도로 재정의하려는 경우에만 권한을 거부("블랙리스트")하기 위한 문을 생성해야 합니다. 권한 거부의 수가 늘어나면 권한 문제를 해결하기가 더 어려워질 수 있기 때문에 그 수를 최소한으로 제한하는 것이 좋습니다. IAM이 정책 로직을 평가하는 방법에 대한 자세한 정보는 [정책 평가 로직 \(p. 531\)](#) 단원을 참조하십시오.

Note

기본적으로 AWS 계정 루트 사용자만 해당 계정의 모든 리소스에 액세스할 수 있습니다. 따라서 루트 사용자로 로그인하지 않은 경우 정책이 부여한 권한이 있어야 합니다.

시각적 편집기에서 여러 서비스 지정

시각적 편집기를 사용하여 정책을 생성할 때 한 번에 서비스 하나만 선택할 수 있습니다. 시각적 편집기는 해당 서비스 하나에 대한 작업에서 선택할 수 있도록 허용하므로 이렇게 하는 것이 모범 사례입니다. 그런 다음

해당 서비스 및 선택한 작업에서 지원되는 리소스 중에서 선택합니다. 이렇게 하면 정책을 쉽게 생성하고 문제를 해결할 수 있습니다.

JSON 구문에 대해 잘 알고 있는 경우 와일드카드 문자(*)를 사용하여 여러 서비스를 수동으로 지정할 수도 있습니다. 예를 들어, **code***를 입력하여 CodeBuild 및 CodeCommit과 같이 Code로 시작하는 모든 서비스에 대한 권한을 제공합니다. 그러나 정책을 완료하려면 작업 및 리소스 ARN을 입력해야 합니다. 또한 정책을 저장하면 각 서비스를 별도의 권한 블록에 포함하도록 정책이 [재구성 \(p. 455\)](#)될 수 있습니다.

또는 서비스에 대해 JSON 구문(예: 와일드카드)을 사용하기 위해 JSON 탭을 통해 정책을 생성, 편집 및 저장합니다.

시각적 편집기에서 정책의 크기 줄이기

시각적 편집기를 사용하여 정책을 생성할 때 IAM은 정책을 저장하기 위해 JSON 문서를 생성합니다. JSON 탭으로 전환하여 이 문서를 볼 수 있습니다. 이 JSON 문서가 정책의 크기 제한을 초과할 경우, 시각적 편집기에 오류 메시지가 표시되며 정책을 검토하거나 저장할 수 없습니다. 관리형 정책의 크기에 대한 IAM 제한을 보려면 [IAM 엔터티 문자 제한 \(p. 487\)](#) 단원을 참조하십시오.

시각적 편집기에서 정책의 크기를 줄이려면 정책을 편집하거나 권한 블록을 다른 정책으로 옮깁니다. 오류 메시지에는 정책 문서에 포함된 문자 수가 포함되며, 이 정보를 통해 정책의 크기를 줄일 수 있습니다.

시각적 편집기에서 인식할 수 없는 서비스, 작업 또는 리소스 유형 수정

시각적 편집기에서 정책을 생성하거나 편집할 때 정책에 인식할 수 없는 서비스, 작업 또는 리소스 유형이 포함되어 있다는 경고가 표시될 수 있습니다.

Note

IAM은 정책 요약을 지원하는 서비스의 이름, 작업 및 리소스 유형을 검토합니다. 그러나 존재하지 않는 리소스 값이나 조건이 정책 요약에 포함될 수 있습니다. 항상 [정책 시뮬레이터 \(p. 383\)](#)로 정책을 테스트합니다.

정책에 인식할 수 없는 서비스, 작업 또는 리소스 유형이 포함되는 경우 다음 오류 중 하나가 발생한 것입니다.

- 미리 보기 서비스 – 미리 보기에 있는 서비스는 시각적 편집기를 지원하지 않습니다. 미리 보기에 참여하고 있는 경우 경고를 무시하고 계속 진행할 수 있지만, 정책을 완료하려면 작업 및 리소스 ARN을 수동으로 입력해야 합니다. 또는 JSON 탭을 선택하여 JSON 정책 문서를 입력하거나 붙여 넣을 수 있습니다.
- 사용자 지정 서비스 – 사용자 지정 서비스는 시각적 편집기를 지원하지 않습니다. 사용자 지정 서비스를 사용하고 있는 경우 경고를 무시하고 계속 진행할 수 있지만, 정책을 완료하려면 작업 및 리소스 ARN을 수동으로 입력해야 합니다. 또는 JSON 탭을 선택하여 JSON 정책 문서를 입력하거나 붙여 넣을 수 있습니다.
- 시각적 편집기를 지원하지 않는 서비스 – 정책에 시각적 편집기를 지원하지 않는 정식 버전(GA) 서비스가 포함되어 있는 경우 경고를 무시하고 계속 진행할 수 있지만, 정책을 완료하려면 작업 및 리소스 ARN을 수동으로 입력해야 합니다. 또는 JSON 탭을 선택하여 JSON 정책 문서를 입력하거나 붙여 넣을 수 있습니다.

일반적으로 사용할 수 있는 서비스는 공개적으로 출시된 서비스이며 프리뷰 또는 사용자 지정 서비스가 아닙니다. 인식할 수 없는 서비스를 일반적으로 사용할 수 있고 이름을 올바르게 입력한 경우 서비스는 시각적 편집기를 지원하지 않습니다. GA 서비스에 대한 시각적 편집기 또는 정책 요약 지원을 요청하는 방법을 알아보려면 [서비스가 IAM 정책 요약을 지원하지 않음 \(p. 459\)](#) 단원을 참조하십시오.

- 시각적 편집기를 지원하지 않는 작업 – 지원되지 않는 작업과 함께 지원되는 서비스가 정책에 포함되어 있는 경우 경고를 무시하고 계속 진행할 수 있지만, 정책을 완료하려면 리소스 ARN을 수동으로 입력해야 합니다. 또는 JSON 탭을 선택하여 JSON 정책 문서를 입력하거나 붙여 넣을 수 있습니다.

지원되지 않는 작업과 함께 지원되는 서비스가 정책에 포함되어 있으면 서비스가 시각적 편집기를 완전히 지원하지 않습니다. GA 서비스에 대한 시각적 편집기 또는 정책 요약 지원을 요청하는 방법을 알아보려면 [서비스가 IAM 정책 요약을 지원하지 않음 \(p. 459\)](#) 단원을 참조하십시오.

- 시각적 편집기를 지원하지 않는 리소스 유형 – 지원되지 않는 리소스 유형과 함께 지원되는 작업이 정책에 포함되어 있는 경우 경고를 무시하고 계속 진행할 수 있습니다. 그러나 IAM은 선택한 모든 작업에 대한 리소스를 포함했는지 여부를 확인할 수 없으며, 추가 경고가 표시될 수 있습니다.
- 오타 – 시각적 편집기에 서비스, 작업 또는 리소스를 수동으로 입력할 경우 오타가 포함된 정책이 생성될 수 있습니다. 시각적 편집기를 사용하여 서비스 및 작업 목록에서 선택한 다음 표시되는 메시지에 따라 리소스 섹션을 완료하는 것이 모범 사례입니다. 그러나 서비스가 시각적 편집기를 완전히 지원하지 않는 경우 정책의 부분을 수동으로 입력해야 할 수 있습니다.

정책에 위와 같은 오류가 없다는 것을 확신한다면 오타가 포함된 것일 수 있습니다. 서비스, 작업 및 리소스 유형 이름에 오탈자가 있는지 확인합니다. 예를 들어 s2 대신 s3를 사용하고, `ListMyBuckets` 대신 `ListAllMyBuckets`을 사용할 수 있습니다. 또 다른 일반적인 작업 오타는 ARN에 불필요한 텍스트를 추가(예: arn:aws:s3: : :*)하거나 작업에서 콜론을 누락(예: `AWSAuthRuntimeService.AuthenticatePassword`)하는 것입니다. 정책 검토를 선택하여 정책 요약을 검토하고 정책이 의도한 권한을 제공하는지 여부를 확인하여 정책에 오타가 있는지를 평가할 수 있습니다.

정책 요약을 사용하여 문제 해결

정책 요약과 관련된 문제를 진단하고 해결할 수 있습니다.

정책 요약 누락

IAM 콘솔에는 정책에서 각 서비스에 대해 허용되거나 거부되는 액세스 레벨, 리소스, 조건을 설명하는 정책 요약 테이블이 포함되어 있습니다. 정책은 3가지 테이블, 즉 [정책 요약 \(p. 419\)](#), [서비스 요약 \(p. 429\)](#), [작업 요약 \(p. 433\)](#)으로 요약됩니다. 정책 요약 테이블에는 서비스 목록과 선택한 정책에 의해 정의된 권한의 요약이 포함되어 있습니다. 사용자 페이지에서 사용자에게 연결된 정책에 대한 [정책 요약 \(p. 419\)](#)을 볼 수 있습니다. 정책 페이지에서 관리형 정책에 대한 정책 요약을 볼 수 있습니다. AWS가 정책 요약을 렌더링할 수 없는 경우 요약 대신 JSON 정책 문서가 제공되며 다음 오류가 표시됩니다.

A summary for this policy cannot be generated. You can still view or edit the JSON policy document.

정책이 요약을 포함하지 않을 경우 다음 오류 중 하나가 발생한 것입니다.

- 지원되지 않는 정책 요소 – IAM은 다음 [정책 요소 \(p. 498\)](#) 중 하나를 포함하는 정책에 대해 정책 요약 생성을 지원하지 않습니다.
 - Principal
 - NotPrincipal
 - NotResource
- 정책 권한 없음 – 정책이 유효한 권한을 제공하지 않을 경우 정책 요약을 생성할 수 없습니다. 예를 들어 정책이 요소 "NotAction": "*"과 함께 단일 명령문을 포함하는 경우 이 정책은 "모든 작업"(*)을 제외한 모든 작업에 대한 액세스 권한을 부여합니다. 즉 어떤 작업에 대해서도 Deny 또는 Allow 액세스 권한을 부여하지 않습니다.

Note

`NotPrincipal`, `NotAction`, `NotResource` 등의 이러한 정책 요소를 사용할 때는 주의해야 합니다. 정책 요소 사용에 대한 자세한 정보는 [IAM JSON 정책 요소 참조 \(p. 498\)](#) 단원을 참조하십시오.

일치하지 않는 서비스와 리소스를 제공하는 경우 유효한 권한을 제공하지 않는 정책을 생성할 수 있습니다. 이는 한 서비스의 작업과 다른 서비스의 리소스를 지정하는 경우에 발생할 수 있습니다. 이 경우에는

정책 요약이 나타납니다. 요약의 리소스 열에 다른 서비스의 리소스를 포함할 수 있는 경우에만 문제가 있다는 표시가 나타납니다. 이 열에 일치하지 않는 리소스가 포함되어 있으면 정책에 오류가 있는지 검토해야 합니다. 정책을 더 잘 이해하려면 항상 [정책 시뮬레이터 \(p. 383\)](#)로 테스트합니다.

정책 요약에 인식할 수 없는 서비스, 작업 또는 리소스 유형 포함됨

IAM 콘솔에서 [정책 요약 \(p. 419\)](#)에 경고 기호()가 있으면 정책 요약에 인식할 수 없는 서비스, 작업 또는 리소스 유형이 포함되었을 수 있습니다. 정책 요약 내의 경고에 대해 알아보려면 [정책 요약\(서비스 목록\) \(p. 419\)](#)을 참조하십시오.

Note

IAM은 정책 요약을 지원하는 서비스의 이름, 작업 및 리소스 유형을 검토합니다. 그러나 존재하지 않는 리소스 값이나 조건이 정책 요약에 포함될 수 있습니다. 항상 [정책 시뮬레이터 \(p. 383\)](#)로 정책을 테스트합니다.

정책에 인식할 수 없는 서비스, 작업 또는 리소스 유형이 포함되는 경우 다음 오류 중 하나가 발생한 것입니다.

- 미리 보기 서비스 – 미리 보기와 있는 서비스는 정책 요약을 지원하지 않습니다.
- 사용자 지정 서비스 – 사용자 지정 서비스는 정책 요약을 지원하지 않습니다.
- 서비스가 요약을 지원하지 않음 – 정책 요약을 지원하지 않는 정식 버전(GA) 서비스가 정책에 포함되어 있으면 서비스가 정책 요약 테이블의 Unrecognized services(알 수 없는 서비스) 섹션에 포함됩니다. 일반적으로 사용할 수 있는 서비스는 공개적으로 출시된 서비스이며 프리뷰 또는 사용자 지정 서비스가 아닙니다. 인식할 수 없는 서비스가 정식 버전이고 이름을 올바르게 입력한 경우에는 서비스에서 IAM 정책 요약을 지원하지 않습니다. GA 서비스에 대한 정책 요약 지원을 요청하는 방법을 알아보려면 [서비스가 IAM 정책 요약을 지원하지 않음 \(p. 459\)](#)을 참조하십시오.
- 작업이 요약을 지원하지 않음 – 지원되지 않는 작업과 함께 지원되는 서비스가 정책에 포함되어 있으면 작업이 서비스 요약 테이블의 Unrecognized actions(알 수 없는 작업) 섹션에 포함됩니다. 서비스 요약 내의 경고에 대해 알아보려면 [서비스 요약\(작업 목록\) \(p. 429\)](#)을 참조하십시오.
- 리소스 유형이 요약을 지원하지 않음 – 정책에 지원되지 않는 리소스 유형을 가진 지원되는 작업이 포함된 경우, 서비스 요약 테이블의 Unrecognized resource types(인식되지 않은 리소스 유형) 섹션에 리소스가 포함됩니다. 서비스 요약 내의 경고에 대해 알아보려면 [서비스 요약\(작업 목록\) \(p. 429\)](#)을 참조하십시오.
- 오타 – AWS의 정책 검증기는 JSON의 구문이 정확한지 여부만 검사하므로 생성한 정책에 오타가 포함될 수 있습니다. 정책에 위와 같은 오류가 없다는 것을 확인한다면 오타가 포함된 것일 수 있습니다. 서비스, 작업 및 리소스 유형 이름에 오탈자가 있는지 확인합니다. 예를 들어 s2 대신 s3를 사용하고, `ListMyBuckets` 대신 `ListAllMyBuckets`을 사용할 수 있습니다. 또 다른 일반적인 작업 오타는 ARN에 불필요한 텍스트를 추가(예: `arn:aws:s3::: : :*`)하거나 작업에서 콜론을 누락(예: `AWSAuthRuntimeService.AuthenticatePassword`)하는 것입니다. 정책 검증기 (p. 383)를 사용하여 정책이 의도된 권한을 제공하는지 여부를 확인하여 정책에 오타가 있는지 검사할 수 있습니다.

서비스가 IAM 정책 요약을 지원하지 않음

정식 버전(GA) 서비스 또는 작업이 IAM 정책 요약 또는 시각적 편집기에서 인식되지 않으면 서비스가 이러한 기능을 지원하지 않을 수 있습니다. 일반적으로 사용할 수 있는 서비스는 공개적으로 출시된 서비스이며 프리뷰 또는 사용자 지정 서비스가 아닙니다. 인식할 수 없는 서비스를 일반적으로 사용할 수 있고 이름을 올바르게 입력한 경우 서비스는 이러한 기능을 지원하지 않습니다. 지원되지 않는 작업과 함께 지원되는 서비스가 정책에 포함되어 있으면 서비스가 IAM 정책 요약을 완전히 지원하지 않습니다.

서비스에서 IAM 정책 요약 또는 시각적 편집기 지원을 추가하도록 요청하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.

2. 지원되지 않는 서비스가 포함된 정책을 찾습니다.
 - 그 정책이 관리형 정책인 경우, 탐색 창에서 정책을 선택합니다. 정책 목록에서 보려는 정책의 이름을 선택합니다.
 - 사용자에게 연결된 인라인 정책인 경우, 탐색 창에서 사용자를 선택합니다. 사용자 목록에서 정책을 보려는 사용자의 이름을 선택합니다. 사용자에 대한 정책 테이블에서 보려는 정책 요약의 헤더를 확장 합니다.
3. 왼쪽의 AWS Management 콘솔 바닥글에서 의견을 선택합니다. Tell us about your experience(귀하의 작업 환경에 대해 말씀해 주십시오) 상자에 **I request that the <ServiceName> service add support for IAM policy summaries and the visual editor**를 입력하십시오. 요약 지원을 바라는 서비스가 두 개 이상인 경우 **I request that the <ServiceName1>, <ServiceName2>, and <ServiceName3> services add support for IAM policy summaries and the visual editor**라고 입력합니다

서비스에서 누락된 작업에 대한 IAM 정책 요약 지원을 추가하도록 요청하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 지원되지 않는 서비스가 포함된 정책을 찾습니다.
 - 그 정책이 관리형 정책인 경우, 탐색 창에서 정책을 선택합니다. 정책 목록에서 보려는 정책의 이름을 선택합니다.
 - 사용자에게 연결된 인라인 정책인 경우, 탐색 창에서 사용자를 선택합니다. 사용자 목록에서 정책을 보려는 사용자의 이름을 선택합니다. 사용자에 대한 정책 표에서 보려는 정책의 이름을 선택하여 정책 요약을 펼칩니다.
3. 정책 요약에서 지원되지 않는 작업을 포함하는 서비스의 이름을 선택합니다.
4. 왼쪽의 AWS Management 콘솔 바닥글에서 의견을 선택합니다. Tell us about your experience(귀하의 작업 환경에 대해 말씀해 주십시오) 상자에 **I request that the <ServiceName> service add IAM policy summary and the visual editor support for the <ActionName> action**를 입력하십시오. 지원되지 않는 작업을 두 개 이상 보고하는 경우 **I request that the <ServiceName> service add IAM policy summary and the visual editor support for the <ActionName1>, <ActionName2>, and <ActionName3> actions**라고 입력합니다

다른 서비스에 누락된 작업을 포함하도록 요청하려면 마지막 세 단계를 반복합니다.

정책이 필요한 권한을 부여하지 않음

사용자, 그룹, 역할 또는 리소스에 권한을 할당하려면 권한을 정의하는 문서인 정책을 생성해야 합니다. 정책 문서에는 다음 요소가 포함됩니다.

- Effect – 정책에서 액세스를 허용하는지 또는 거부하는지 여부
- Action – 정책에서 허용하거나 거부하는 작업 목록
- Resource – 작업이 발생할 수 있는 리소스 목록
- 조건(선택 사항) – 정책에서 권한을 부여하는 상황

이러한 요소와 기타 정책 요소에 대한 자세한 정보는 [IAM JSON 정책 요소 참조 \(p. 498\)](#) 단원을 참조하십시오.

액세스 권한을 부여하려면 정책이 지원되는 리소스를 가진 작업을 정의해야 합니다. 정책에 조건도 있는 경우 이 조건은 [전역 조건 키 \(p. 551\)](#)를 포함해야 하거나, 작업에 적용해야 합니다. 작업에서 어떤 리소스를 지원하는지 확인하려면 해당 서비스의 [AWS 설명서](#)를 참조하십시오. 작업에서 어떤 조건을 지원하는지 확인하려면 [???](#) 단원을 참조하십시오.

정책에서 권한을 부여하지 않는 작업, 리소스 또는 조건을 정의하는지 확인하려면 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 사용하여 해당 정책의 정책 요약 (p. 419)을 보십시오. 정책 요약을 사용하여 정책의 문제를 식별하고 수정할 수 있습니다.

IAM 정책에 정의되었는데도 요소가 권한을 부여하지 않는 몇 가지 이유는 다음과 같습니다.

- 적용 가능한 리소스 없이 작업이 정의된 경우 (p. 461)
- 적용 가능한 작업 없이 리소스가 정의된 경우 (p. 461)
- 적용 가능한 작업 없이 조건이 정의된 경우 (p. 462)

경고를 포함하는 정책 요약의 예를 보려면 the section called “정책 요약(서비스 목록)” (p. 419) 단원을 참조 하십시오.

적용 가능한 리소스 없이 작업이 정의된 경우

아래 정책은 모든 ec2:Describe* 작업과 특정 리소스를 정의합니다. 이러한 작업 중에서 리소스 수준 권한을 지원하는 작업이 없기 때문에 어떤 ec2:Describe 작업도 부여되지 않습니다. 리소스 수준 권한이란 작업이 정책의 Resource (p. 508) 요소에 있는 ARN을 사용하여 리소스를 지원함을 의미합니다. 작업이 리소스 수준 권한을 지원하지 않는 경우에는 정책의 이 명령문에서 * 요소에 와일드카드(Resource)를 사용해야 합니다. 리소스 수준 권한을 서비스에 대해 알아보려면 IAM로 작업하는 AWS 서비스 (p. 488) 단원을 참조하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": "ec2:Describe*",  
        "Resource": "arn:aws:ec2:us-east-2:ACCOUNT-ID:instance/*"  
    }]  
}
```

이 정책은 어떤 권한도 제공하지 않으며, 정책 요약에는 다음 오류가 포함됩니다.

This policy does not grant any permissions. To grant access, policies must have an action that has an applicable resource or condition.

정책을 수정하려면 * 요소에 Resource를 사용해야 합니다.

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

적용 가능한 작업 없이 리소스가 정의된 경우

아래 정책은 Amazon S3 버킷 리소스를 정의하지만, 해당 리소스에서 수행할 수 있는 S3 작업을 포함하지 않습니다. 이 정책은 또한 모든 Amazon CloudFront 작업에 대한 전체 액세스 권한을 부여합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": "cloudfront:*    }]
```

```
    "Resource": [
        "arn:aws:cloudfront:*",
        "arn:aws:s3:::examplebucket"
    ],
}
```

이 정책은 모든 CloudFront 작업에 대한 권한을 제공합니다. 하지만 정책에서 S3 작업을 정의하지 않고 S3 **examplebucket** 리소스를 정의하기 때문에 정책 요약에 다음 경고가 표시됩니다.

This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition.

이 정책을 수정하여 S3 버킷 권한을 제공하려면 버킷 리소스에서 수행할 수 있는 S3 작업을 정의해야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudfront:*",
                "s3:CreateBucket",
                "s3>ListBucket*",
                "s3:PutBucket*",
                "s3:GetBucket*"
            ],
            "Resource": [
                "arn:aws:cloudfront:*",
                "arn:aws:s3:::examplebucket"
            ]
        }
    ]
}
```

또는 이 정책을 수정하여 CloudFront 권한만 제공하려면 S3 리소스를 삭제하십시오.

적용 가능한 작업 없이 조건이 정의된 경우

아래 정책은 S3 접두사가 `custom`이고 버전 ID가 1234일 경우 모든 S3 리소스에 대해 2개의 Amazon S3 작업을 정의합니다. 하지만 `s3:VersionId` 조건 키가 객체 버전 태그 지정에 사용되었으며, 정의된 버킷 작업이 이 조건 키를 지원하지 않습니다. 작업에서 어떤 조건을 지원하는지 알아보려면 [???](#) 단원을 참조하고, 링크를 클릭하여 조건 키에 대한 서비스 설명서를 보십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3>ListBucketVersions",
                "s3>ListBucket"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:prefix": [
                        "custom"
                    ],
                    "s3:VersionId": [
                        "1234"
                    ]
                }
            }
        }
    ]
}
```

```
        ]
    }
}
]
```

이 정책은 버킷 이름에 `s3>ListBucketVersions` 접두사가 있는 경우 `s3ListBucket` 작업 및 `custom` 작업에 대한 권한을 제공합니다. 하지만 정의된 작업 중 `s3VersionId` 조건을 지원하는 작업이 없기 때문에 정책 요약에 다음 오류가 표시됩니다.

This policy does not grant any permissions. To grant access, policies must have an action that has an applicable resource or condition.

이 정책을 수정하여 S3 객체 버전 태그 지정을 사용하려면, `s3VersionId` 조건 키를 지원하는 S3 작업을 정의해야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3>ListBucketVersions",
                "s3>ListBucket",
                "s3GetObjectVersion"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:prefix": [
                        "custom"
                    ],
                    "s3:VersionId": [
                        "1234"
                    ]
                }
            }
        }
    ]
}
```

이 정책은 정책의 모든 작업과 조건에 대한 권한을 제공합니다. 하지만 하나의 작업이 모든 조건을 충족하는 경우가 없기 때문에 어떤 권한도 제공하지 않습니다. 이렇게 하는 대신, 적용할 조건을 갖는 작업만 각각 포함하도록 두 개의 구문을 별도로 작성해야 합니다.

이 정책을 수정하려면 두 개의 구문을 작성합니다. 첫째 구문에는 `s3:prefix` 조건을 지원하는 작업이 포함되고, 둘째 구문에는 `s3:VersionId` 조건을 지원하는 작업이 포함됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3>ListBucketVersions",
                "s3>ListBucket"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:prefix": "custom"
                }
            }
        }
    ]
}
```

```
        }
    },
{
    "Effect": "Allow",
    "Action": "s3:GetObjectVersion",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "s3:VersionId": "1234"
        }
    }
}
]
```

정책 관리 문제 해결

정책 관리와 관련된 문제를 진단하고 해결할 수 있습니다.

IAM 계정에서 정책 연결 또는 분리

일부 AWS 관리형 정책은 서비스에 연결되어 있습니다. 이러한 정책은 해당 서비스에 대한 [서비스 연결 역할](#) (p. 154)에서만 사용됩니다. IAM 콘솔에서 정책의 요약 페이지를 보면 페이지에 정책이 서비스에 연결되어 있음을 나타내는 배너가 포함되어 있습니다. 이 정책을 IAM 내의 사용자, 그룹 또는 역할에 연결할 수 없습니다. 서비스에 대한 서비스 연결 역할을 생성하면 이 정책이 새 역할에 자동으로 연결됩니다. 정책이 필요 하므로 서비스 연결 역할에서 정책을 분리할 수 없습니다.

작업 기반 IAM 자격 증명 관련 정책 변경

작업에 따라 IAM 자격 증명(사용자, 그룹 및 역할)에 대한 정책을 업데이트할 수 있습니다. 이 작업을 수행 하려면 CloudTrail Event history(이벤트 이력)에서 계정의 이벤트를 확인합니다. CloudTrail 이벤트 로그에는 정책의 권한을 변경하는 데 사용할 수 있는 자세한 이벤트 정보가 포함되어 있습니다. 사용자 또는 역할이 AWS에서 작업을 수행하려 하지만 요청이 거부된 경우 이들에게 해당 작업을 수행할 수 있는 권한을 부여해야 할지 여부를 고려할 수 있습니다. 권한을 부여해야 하는 경우 해당 작업과 이들이 액세스하려고 했던 리소스의 ARN도 정책에 추가할 수 있습니다. 또는, 사용자나 역할에 사용하지 않는 권한이 있는 경우 정책에서 그러한 권한을 제거하는 것을 고려할 수도 있습니다. 정책은 필요한 작업을 수행하는 데 필요한 [최소 권한](#) (p. 44)만 부여해야 합니다. CloudTrail 사용에 대한 자세한 정보는 AWS CloudTrail 사용 설명서의 [CloudTrail 콘솔에서 CloudTrail 이벤트 보기](#) 단원을 참조하십시오.

JSON 정책 문서 문제 해결

JSON 정책 문서와 관련된 문제를 진단하고 해결할 수 있습니다.

JSON 정책 객체가 둘 이상인 경우

IAM 정책은 단 하나의 JSON 객체로 구성되어야 합니다. 객체는 중괄호 {}로 묶어 표시합니다. 대괄호 [] 안에 중괄호 {}를 추가로 삽입하여 JSON 객체 내에 다른 객체를 종합시킬 수도 있지만 정책에 따라 중괄호 {}를 묶는 대괄호 []는 하나로 제한됩니다. 다음 예제는 최상위 레벨에 객체 2개가 추가되었기 때문에 올바르지 않습니다(###으로 표시).

```
{
    "Version": "2012-10-17",
    "Statement":
    {
        "Effect": "Allow",
        "Action": "ec2:Describe*",
```

```
        "Resource": "*"
    }
}

{
    "Statement": [
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::my-bucket/*"
    ]
}
```

하지만 올바른 정책 문법을 사용하여 위 예제의 의도를 만족하는 방법도 있습니다. 2개의 정책 객체에 Statement 요소를 각각 추가하지 않고 두 블록을 단일 Statement 요소로 결합하면 됩니다. 그러면 다음 예제와 같이 Statement 요소가 두 객체의 배열을 값으로 인식합니다(굵은체로 표시).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::my-bucket/*"
        }
    ]
}
```

JSON Statement 요소가 둘 이상인 경우

이 오류는 처음에는 위 오류의 변형으로 보일 수도 있습니다. 하지만 구문으로 보면 다른 유형의 오류입니다. 다음 예제에는 중괄호 {} 한 쌍이 최상위 레벨로 정책 객체 하나만 표시하고 있습니다. 하지만 객체에 포함된 Statement 요소는 2개입니다.

IAM 정책에서는 콜론 왼쪽의 이름(Statement)과 오른쪽의 값으로 구성된 Statement 요소 1개만 추가할 수 있습니다. 그리고, statement 요소의 값은 Effect 요소 1개와 Action 요소 1개, 그리고 Resource 요소 1개가 중괄호 {}로 묶여 구성된 객체가 되어야 합니다. 다음 예제는 정책 객체에 Statement 요소가 2개 포함되었기 때문에 올바르지 않습니다(###으로 표시).

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "ec2:Describe*",
        "Resource": "*"
    },
    "Statement": {
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::my-bucket/*"
    }
}
```

값 객체는 여러 값 객체의 배열일 수 있습니다. 이 문제를 해결하려면, 다음 예제와 같이 객체 배열을 사용하여 2개의 Statement 요소를 하나로 결합합니다(굵은체로 표시).

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": "ec2:Describe*",
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::my-bucket/*"
    }
]
```

Statement 요소 값이 객체 배열이 되었습니다. 이제 위 예제의 배열은 두 객체로 구성되며, 각 객체 자체가 정확한 Statement 요소 값으로 인식됩니다. 배열을 구성하는 각 객체는 쉼표로 구분합니다.

JSON Statement 요소의 Effect, Action 또는 Resource 요소가 둘 이상인 경우

Statement 이름/값 쌍에서 값 부분을 보면 객체가 Effect 요소 1개, Action 요소 1개, 그리고 Resource 요소 1개로 구성되어야 합니다. 다음은 Effect의 값 객체에 Statement 요소가 2개이기 때문에 잘못된 정책입니다.

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

Note

정책 엔진은 새로운 정책이나 편집된 정책에서 이러한 오류를 허용하지 않습니다. 하지만 정책 엔진은 엔진 업데이트 이전에 저장된 정책은 계속 허용합니다. 오류와 관련한 기존 정책 특성은 아래와 같습니다.

- Effect 요소가 다수일 때: 마지막 Effect 요소만 따릅니다. 나머지 요소는 모두 무시됩니다.
- Action 요소가 다수일 때: Action 요소가 모두 내부적으로 결합되어 마치 단일 목록인 것처럼 처리됩니다.
- Resource 요소가 다수일 때: Resource 요소가 모두 내부적으로 결합되어 마치 단일 목록인 것처럼 처리됩니다.

정책 엔진은 구문 오류 정책을 저장하도록 허용하지 않습니다. 따라서 저장하기 전에 정책 오류를 정정해야 합니다. 정책 검사기 ([p. 382](#))는 이전 정책 오류를 찾는데 효과적일 뿐만 아니라 정정 방법까지 알려주는 도구입니다.

모든 경우 해결책은 잘못 추가된 요소를 삭제하는 것입니다. Effect 요소일 때는 삭제 방법이 간단합니다. 앞의 예제에서 Amazon EC2 인스턴스에 대한 권한을 거부하고 싶다면 다음과 같이 정책에서 "Effect": "Allow", 라인을 삭제하면 됩니다.

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

```
        "Effect": "Deny",
        "Action": "ec2:*",
        "Resource": "*"
    }
}
```

하지만 중복 요소가 Action 또는 Resource인 경우에는 해결 방법이 더욱 복잡합니다. 권한을 허용(또는 거부)하려는 작업이 다수일 수도 있고, 여러 리소스에 대한 액세스를 제어할 수도 있기 때문입니다. 예를 들어 다음 예제는 Resource 요소가 여러 개이기 때문에 옮바르지 않습니다(###으로 표시).

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::my-bucket",
        "Resource": "arn:aws:s3:::my-bucket/*"
    }
}
```

Statement 요소의 값 객체에서 필요한 요소는 각각 한 번만 표시할 수 있습니다. 해결책은 객체 배열에 값을 하나씩만 지정하는 것입니다. 다음 예제는 배열을 값 객체로 사용하여 2개의 리소스 요소를 1개의 Resource 요소로 결합함으로써 이를 설명합니다(굵은체로 표시).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::my-bucket",
                "arn:aws:s3:::my-bucket/*"
            ]
        }
    ]
}
```

JSON 버전 요소 누락

Version 정책 요소는 정책 버전과 다릅니다. Version 정책 요소는 정책 내에서 사용되며 정책 언어의 버전을 정의합니다. 반면에 정책 버전은 IAM에서 고객 관리형 정책을 변경할 때 생성됩니다. 변경된 정책은 기존 정책을 덮어쓰지 않습니다. 대신 IAM에서 관리형 정책의 새 버전을 만듭니다. Version 정책 요소에 대한 자세한 정보는 [IAM JSON 정책 요소: Version \(p. 499\)](#)을 참조하십시오. 정책 버전에 대한 자세한 정보는 [the section called "IAM 정책 버전 관리" \(p. 399\)](#) 단원을 참조하십시오.

AWS 기능이 점차 진화하면서 IAM 정책에도 이를 지원할 수 있도록 새로운 기능이 추가되었습니다. 간혹 정책 구문이 업데이트될 때마다 새로운 버전 번호가 추가됩니다. 정책 문법에서 최신 기능을 사용하는 경우에는 정책 구문 분석 엔진에게 사용 버전을 알려주어야 합니다. 기본 정책 버전은 "2008-10-17"입니다. 이때 이후 추가된 정책 기능을 사용하려면 원하는 기능을 지원하는 버전 번호를 지정해야 합니다. 따라서 항상 최신 정책 구문 버전 번호("Version": "2012-10-17")를 추가할 것을 권장합니다. 예를 들어 다음은 정책 변수를 지원하는 정책 구문 버전을 지정하지 않고 리소스 ARN에 정책 변수 \${...}를 사용했기 때문에 옮바르지 않습니다(###으로 표시).

```
{
    "Statement": [
        {
            "Action": "iam:*AccessKey*",
            "Effect": "Allow",
            "Resource": "arn:aws:iam::123456789012:user/${aws:username}"
        }
    ]
}
```

}

다음과 같이 정책 상단에 정책 변수를 지원하는 첫 번째 IAM API 버전인 2012-10-17 값과 함께 Version 요소를 추가하면 이 문제가 해결됩니다(굵은체로 표시).

```
{  
    "Version": "2012-10-17",  
    "Statement":  
    {  
        "Action": "iam:*AccessKey*",  
        "Effect": "Allow",  
        "Resource": "arn:aws:iam::123456789012:user/${aws:username}"  
    }  
}
```

U2F 보안 키 문제 해결

여기 정보를 사용하여 U2F 보안 키 작업 시 공통적으로 발생할 수 있는 문제를 진단하십시오.

주제

- [U2F 보안 키를 활성화할 수 없습니다. \(p. 468\)](#)
- [U2F 보안 키를 사용해 로그인할 수 없습니다. \(p. 469\)](#)
- [U2F 키를 분실했거나 고장 났습니다. \(p. 469\)](#)
- [기타 문제 \(p. 469\)](#)

U2F 보안 키를 활성화할 수 없습니다.

IAM 사용자인지 시스템 관리자인지 자신의 상태에 따라 다음 해결 방법을 문의하십시오.

IAM 사용자

U2F 보안 키가 활성화되지 않으면 다음 사항을 확인하십시오.

- 지원되는 구성을 사용 중입니까?

U2F 및 AWS에 사용할 수 있는 디바이스 및 브라우저 정보는 [U2F 보안 키 사용에 지원되는 구성 \(p. 106\)](#)을 확인하십시오.

- Mozilla Firefox를 사용 중입니까?

U2F를 지원하는 대부분의 Firefox 버전은 기본적으로 지원을 활성화하지 않습니다. Firefox에서 U2F 지원을 활성화하려면 다음과 같이 하십시오.

- Firefox 주소 표시줄에 `about:config`를 입력합니다.

- 열리는 화면의 검색줄에 기본 `u2f`를 입력합니다.

- `security.webauth.u2f`를 선택하고 값을 `true`로 변경합니다.

- 브라우저 플러그인을 사용 중입니까?

AWS는 플러그인 사용을 통한 U2F 브라우저 지원 추가를 지원하지 않습니다. 대신 U2F 표준을 기본적으로 지원하는 브라우저를 사용하십시오.

지원되는 브라우저를 사용하더라도 U2F와 호환되지 않는 플러그인이 있을 수 있습니다. 호환되지 않는 플러그인이 있으면 U2F 보안 키를 활성화하고 사용하지 못할 수 있습니다. 호환되지 않는 플러그인을 모두 비활성화하고 브라우저를 새로 시작해야 합니다. 그런 다음 U2F 보안 키를 다시 활성화해 보십시오.

- 적절한 권한이 있습니까?

위 호환성 문제가 없는 경우 적절한 권한이 없는 경우일 수 있습니다. 시스템 관리자에게 문의하십시오.

시스템 관리자

본인이 관리자이고 IAM 사용자가 지원되는 구성 사용 중인데도 U2F 보안 키를 활성화할 수 없다면 그 사용자에게 적절한 권한이 있는지 확인하십시오. 자세한 예제는 [자습서: 사용자들이 자신의 자격 증명 및 MFA 설정을 구성할 수 있도록 하기 \(p. 39\)](#) 단원을 참조하십시오.

U2F 보안 키를 사용해 로그인할 수 없습니다.

IAM 사용자인데 U2F를 사용해서 AWS Management 콘솔에 로그인할 수 없는 경우 먼저 [U2F 보안 키 사용에 지원되는 구성 \(p. 106\)](#)을 확인하십시오. 지원되는 구성 사용하는데 로그인이 안 되면 시스템 관리자에게 연락해 도움을 받으십시오.

U2F 키를 분실했거나 고장 났습니다.

한 사용자에게는 한 번에 하나의 MFA 디바이스(가상, U2F 보안 키 또는 하드웨어)만 할당됩니다. U2F 보안 키의 교체는 하드웨어 MFA 디바이스의 교체와 비슷합니다. 어떤 유형의 MFA 디바이스를 분실했거나 고장 난 경우 해결 방법은 [MFA 디바이스 분실 또는 작동 중단 시 문제 해결 \(p. 121\)](#)을 참조하십시오.

기타 문제

여기 나오지 않은 U2F 보안 키의 문제는 다음 중 하나를 수행해 보십시오.

- IAM 사용자: 시스템 관리자에게 문의하십시오.
- AWS 계정 루트 사용자: [AWS 지원](#)에 문의하십시오.

IAM 역할 문제 해결

여기 정보를 사용하여 IAM 역할 작업 시 공통적으로 발생할 수 있는 문제를 진단 및 수정하십시오.

주제

- [역할을 위임할 수 없음 \(p. 469\)](#)
- [내 AWS 계정에 표시되는 새 역할 \(p. 470\)](#)
- [AWS 계정에서 역할을 편집하거나 삭제할 수 없음 \(p. 471\)](#)
- [iam:PassRole를 수행하도록 인증되지 않음 \(p. 471\)](#)
- [12시간 길이 세션을 선택한 경우 역할을 위임할 수 없는 이유\(AWS CLI, AWS API\) \(p. 471\)](#)

역할을 위임할 수 없음

- 역할 이름은 대소문자를 구분하므로 역할 이름을 정확하게 사용하십시오.
- 해당 IAM 정책에서 사용자가 위임하려는 역할에 대해 sts:AssumeRole을 호출할 수 있는 권한을 부여하는지 확인하십시오. IAM 정책의 Action 요소는 AssumeRole 액션을 호출할 수 있어야 합니다. 또한 IAM 정책의 Resource 요소는 위임하려는 역할을 지정해야 합니다. 예를 들어 Resource 요소는 ARN(Amazon Resource Name) 또는 와일드카드(*)를 통해 역할을 지정할 수 있습니다. 예를 들어 사용자에게 적용되는 하나 이상의 정책은 다음과 유사한 권한을 부여해야 합니다.

```
"Effect": "Allow",
```

```
"Action": "sts:AssumeRole",
"Resource": "arn:aws:iam::account_id_number:role/role-name-you-want-to-assume"
```

- IAM 자격 증명에 IAM 정책에 필요한 태그가 있는지 확인하십시오. 예를 들어 다음 정책 권한에서 Condition 요소는 역할을 위임하도록 요청할 보안 주체에게 특정 태그가 있어야 한다는 것을 요구합니다. department = HR 또는 department = CS 태그가 지정되어 있어야 합니다. 그렇지 않으면 역할을 위임할 수 없습니다. IAM 사용자 및 역할 태그 지정에 대한 자세한 내용은 [the section called “엔터티 태그 지정” \(p. 259\)](#) 단원을 참조하십시오.

```
"Effect": "Allow",
"Action": "sts:AssumeRole",
"Resource": "*",
"Condition": {"StringEquals": {"aws:PrincipalTag/department": [
    "HR",
    "CS"
]}}}
```

- 역할의 신뢰 정책에 지정된 모든 조건을 만족하고 있는지 확인하십시오. Condition 요소는 만료 날짜와 외부 ID를 지정하거나, 반드시 특정 IP 주소를 이용해야만 요청이 가능하도록 지정할 수 있습니다. 다음 예에서 현재 날짜가 지정된 날짜 이후의 시간이면 이 정책은 일치하지 않으며 해당 역할을 수임할 권한을 사용자에게 부여할 수 없습니다.

```
"Effect": "Allow",
"Action": "sts:AssumeRole",
"Resource": "arn:aws:iam::account_id_number:role/role-name-you-want-to-assume"
"Condition": {
    "DateLessThan" : {
        "aws:CurrentTime" : "2016-05-01T12:00:00Z"
    }
}
```

- AssumeRole을 호출하는 AWS 계정이 위임하려는 역할에 대해 신뢰할 수 있는 엔터티인지 확인하십시오. 신뢰할 수 있는 대상이라면 역할의 신뢰 정책에 Principal로 정의되어 있습니다. 다음은 수임할 역할에 연결된 신뢰 정책의 예입니다. 이 예에서 IAM 사용자가 로그인한 계정 ID는 123456789012여야 합니다. 계정 번호가 역할의 신뢰 정책의 Principal 요소에 명시되어 있지 않은 경우 해당 역할을 수임할 수 없습니다. 액세스 정책에서 어떤 권한이 부여되었는지는 중요하지 않습니다. 예제 정책은 2017년 7월 1일부터 2017년 12월 31일(UTC)까지 발생한 작업에 대한 권한을 제한합니다. 이 날짜 전이나 후에 로그인 경우에는 정책이 일치하지 않기 때문에 해당 역할을 수행할 수 없습니다.

```
"Effect": "Allow",
"Principal": { "AWS": "arn:aws:iam::123456789012:root" },
"Action": "sts:AssumeRole",
"Condition": {
    "DateGreaterThan": {"aws:CurrentTime": "2017-07-01T00:00:00Z"},
    "DateLessThan": {"aws:CurrentTime": "2017-12-31T23:59:59Z"}
}
```

내 AWS 계정에 표시되는 새 역할

일부 AWS 서비스에서는 해당 서비스에 직접 연결된 고유한 유형의 서비스 역할을 사용해야 합니다. 이 [서비스 연결 역할 \(p. 154\)](#)은 해당 서비스에서 사전 정의하여 해당 서비스에 필요한 모든 권한을 포함합니다. 필요 한 권한을 수동으로 추가할 필요가 없으므로 서비스를 더 쉽게 설정할 수 있습니다. 서비스 연결 역할에 대한 일반적인 내용은 [서비스 연결 역할 사용 \(p. 195\)](#) 단원을 참조하십시오.

서비스 연결 역할을 지원하려 할 때 이미 서비스를 사용 중일 수 있습니다. 그런 경우 계정의 새 역할에 대해 알리는 이메일을 받을 수 있습니다. 이 역할에는 서비스에서 사용자를 대신하여 작업을 수행하는데 필요한 모든 권한이 포함되어 있습니다. 따라서 이 역할을 지원하기 위해 별도의 조치를 취할 필요가 없습니다. 그러나 계정에서 역할을 삭제하면 안 됩니다. 그렇게 하면 서비스가 AWS 리소스에 액세스하는 데 필요한 권한을

제거할 수 있습니다. IAM 콘솔의 IAM 역할 페이지에서 계정의 서비스 연결 역할을 볼 수 있습니다. 서비스 연결 역할은 테이블의 Trusted entities(신뢰할 수 있는 개체) 열에 (Service-linked role)((서비스 연결 역할))로 표시됩니다.

서비스 연결 역할을 지원하는 서비스에 대한 자세한 내용은 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#)를 참조하고 서비스 연결 역할 열에 예가 있는 서비스를 찾습니다. 서비스의 서비스 연결 역할 사용에 대한 정보를 보려면 예 링크를 선택합니다.

AWS 계정에서 역할을 편집하거나 삭제할 수 없음

IAM에서 [서비스 연결 역할 \(p. 154\)](#)에 대한 권한을 삭제하거나 편집할 수 없습니다. 이러한 역할에는 사용자 대신 작업을 수행하기 위해 서비스에 필요한 신뢰 및 권한이 미리 지정되어 포함됩니다. IAM 콘솔, AWS CLI, API 등을 사용하여 서비스 연결 역할의 설명만 편집할 수 있습니다. 콘솔의 IAM 역할 페이지에서 계정의 서비스 연결 역할을 볼 수 있습니다. 서비스 연결 역할은 테이블의 신뢰할 수 있는 개체(Trusted entities) 열에 (서비스 연결 역할)((Service-linked role))로 표시됩니다. 역할의 요약 페이지 배너에도 해당 역할이 서비스 역할임이 표시됩니다. 이러한 역할은 해당 서비스가 관리 및 삭제 작업을 지원할 경우 연결 서비스를 통해서만 관리하고 삭제할 수 있습니다. 서비스 연결 역할을 수정하거나 삭제하면 서비스에서 AWS 리소스에 액세스 하는 데 필요한 권한이 제거될 수 있으므로 주의하십시오.

서비스 연결 역할을 지원하는 서비스에 대한 자세한 내용은 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#)를 참조하고 서비스 연결 역할 열에 예가 있는 서비스를 찾습니다.

iam:PassRole를 수행하도록 인증되지 않음

서비스 연결 역할을 생성하는 경우 해당 역할을 서비스에 전달할 권한이 있어야 합니다. 일부 서비스는 서비스에서 작업을 수행할 때 계정에 서비스 연결 역할을 자동으로 생성합니다. 예를 들어 Amazon EC2 Auto Scaling에서는 사용자가 Auto Scaling 그룹을 처음으로 생성할 때 사용자를 대신해 AWSServiceRoleForAutoScaling 서비스 연결 역할을 생성합니다. PassRole 권한 없이 Auto Scaling 그룹을 생성하려고 하면 다음 오류가 발생합니다.

```
ClientError: An error occurred (AccessDenied) when calling the PutLifecycleHook operation: User: arn:aws:sts::111122223333:assumed-role/Testrole/Diego is not authorized to perform: iam:PassRole on resource: arn:aws:iam::111122223333:role/aws-service-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling
```

이 오류를 해결하려면 관리자에게 iam:PassRole 권한을 추가해 달라고 요청합니다.

서비스 연결 역할을 지원하는 서비스를 알아보려면 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하십시오. 서비스에서 작업 수행 시 자동으로 서비스 연결 역할을 생성하는 서비스를 알아보려면 예 링크를 선택하고 해당 서비스에 대한 서비스 연결 역할 설명서를 확인합니다.

12시간 길이 세션을 선택한 경우 역할을 위임할 수 없는 이유(AWS CLI, AWS API)

AWS STS AssumeRole* API 또는 assume-role* CLI 작업을 사용하여 역할을 위임하는 경우 DurationSeconds 파라미터에 대한 값을 지정할 수 있습니다. 이 값의 범위는 900초(15분)에서 해당 역할에 대한 최대 CLI/API 세션 기간 설정까지 지정할 수 있습니다. 이 설정보다 높게 값을 지정하면 작업에 실패합니다. 이 설정의 최댓값은 12시간입니다. 예를 들어 세션 기간으로 12시간을 지정했는데 관리자가 최대 세션 기간으로 6시간을 설정하면 작업에 실패합니다. 역할에 대한 최댓값을 확인하는 방법을 알아보려면 [역할에 대한 최대 세션 기간 설정 보기 \(p. 228\)](#) 단원을 참조하십시오.

[역할 함께 둑기 \(p. 154\)](#)(역할을 사용하여 두 번째 역할 위임)를 사용하는 경우 세션은 최대 1시간으로 제한됩니다. 그런 다음 DurationSeconds 파라미터를 사용하여 1시간보다 큰 값을 입력하면 이 작업에 실패합니다.

Amazon EC2 및 IAM 문제 해결

이 문서의 정보를 사용하여 Amazon EC2 및 IAM 작업 시 발생할 수 있는 액세스 거부 또는 기타 문제를 해결할 수 있습니다.

주제

- [인스턴스를 시작하려고 할 때 Amazon EC2 콘솔 IAM 역할 목록에서 보여야 할 역할이 보이지 않습니다. \(p. 472\)](#)
- [제 인스턴스에 있는 자격 증명의 역할이 잘못되었습니다. \(p. 472\)](#)
- [AddRoleToInstanceProfile을 호출하려고 하면 AccessDenied 오류가 발생합니다. \(p. 472\)](#)
- [Amazon EC2: 역할로 인스턴스를 시작하려고 하면 AccessDenied 오류가 발생합니다. \(p. 473\)](#)
- [제 EC2 인스턴스의 임시 보안 자격 증명에 액세스할 수 없습니다. \(p. 473\)](#)
- [IAM 하위 트리에서 info 문서의 오류란 무엇인가요? \(p. 474\)](#)

인스턴스를 시작하려고 할 때 Amazon EC2 콘솔 IAM 역할 목록에서 보여야 할 역할이 보이지 않습니다.

다음을 확인하십시오.

- IAM 사용자로 로그인한 경우, `ListInstanceProfiles`를 호출할 권한이 있는지 확인하십시오. 역할 사용 시 필요한 권한에 대한 자세한 내용은 [IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여하기 \(p. 240\)](#)의 "Amazon EC2로 역할을 사용하는 데 필요한 권한"을 참조하십시오. 사용자에게 권한을 추가하는 방법에 대한 자세한 내용은 [IAM 정책 관리 \(p. 377\)](#)을 참조하십시오.
권한을 수정할 수 없는 경우, IAM을 사용할 수 있는 관리자에게 문의하여 권한을 업데이트해야 합니다.
- IAM CLI 또는 API를 사용하여 역할을 만든 경우, 인스턴스 프로파일을 만들고 이 인스턴스 프로파일에 해당 역할을 추가했는지 확인하십시오. 또한 역할과 인스턴스 프로필의 이름을 다르게 설정한 경우, Amazon EC2 콘솔의 IAM 역할 목록에서 올바른 역할 이름을 볼 수 없습니다. Amazon EC2 콘솔의 IAM 역할 목록에는 역할 이름이 아니라 인스턴스 프로필 이름이 나열되어 있습니다. 원하는 역할을 포함한 인스턴스 프로필 이름을 선택해야 합니다. 인스턴스 프로파일에 대한 자세한 내용은 [인스턴스 프로파일 사용 \(p. 244\)](#)을 참조하십시오.

Note

IAM 콘솔을 사용하여 역할을 만드는 경우, 인스턴스 프로파일을 사용하지 않아도 됩니다. 인스턴스 프로파일은 IAM 콘솔에서 만드는 각 역할과 동일한 이름으로 생성되며, 역할은 해당 인스턴스 프로파일에 자동으로 추가됩니다. 하나의 인스턴스 프로파일은 하나의 IAM 역할만 포함할 수 있으며 이 제한은 늘릴 수 없습니다.

제 인스턴스에 있는 자격 증명의 역할이 잘못되었습니다.

인스턴스 프로파일의 역할을 최근에 바꾼 경우, 다음에 예정된 자동 자격 증명 교체 이후에 역할의 자격 증명을 사용할 수 있습니다.

AddRoleToInstanceProfile을 호출하려고 하면 AccessDenied 오류가 발생합니다.

IAM 사용자로 요청을 하는 경우, 다음과 같은 권한이 있는지 확인하십시오.

- 인스턴스 프로파일 ARN과 일치하는 리소스가 포함된 iam:AddRoleToInstanceProfile(예: arn:aws:iam::999999999999:instance-profile/ExampleInstanceProfile).

역할 사용에 필요한 권한에 대한 자세한 내용은 [IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여하기](#) (p. 240)의 "어떻게 시작할 수 있습니까?"를 참조하십시오. 사용자에게 권한을 추가하는 방법에 대한 자세한 내용은 [IAM 정책 관리](#) (p. 377)을 참조하십시오.

Amazon EC2: 역할로 인스턴스를 시작하려고 하면 AccessDenied 오류가 발생합니다.

다음을 확인하십시오.

- 인스턴스 프로필 없이 인스턴스를 시작합니다. 이를 통해 문제가 Amazon EC2 인스턴스의 IAM 역할로 제한되어 있는지 확인할 수 있습니다.
- IAM 사용자로 요청을 하는 경우, 다음과 같은 권한이 있는지 확인하십시오.
 - 와일드 카드 리소스("*)가 포함된 ec2:RunInstances
 - 역할 ARN과 일치하는 리소스가 포함된 iam:PassRole(예: arn:aws:iam::999999999999:role/ExampleRoleName)
- IAM GetInstanceProfile 작업을 호출하여 올바른 인스턴스 프로필 이름 또는 올바른 인스턴스 프로필 ARN을 사용 중인지 확인하십시오. 자세한 내용은 [Amazon EC2 인스턴스로 IAM 역할 사용 단원](#)을 참조하십시오.
- IAM GetInstanceProfile 작업을 호출하여 인스턴스 프로필에 역할이 있는지 확인하십시오. 인스턴스 프로파일이 비어 있으면 AccessDenied 오류가 발생합니다. 역할 만들기에 대한 자세한 내용은 [IAM 역할 생성](#) (p. 202) 단원을 참조하십시오.

역할 사용에 필요한 권한에 대한 자세한 내용은 [IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여하기](#) (p. 240)의 "어떻게 시작할 수 있습니까?"를 참조하십시오. 사용자에게 권한을 추가하는 방법에 대한 자세한 내용은 [IAM 정책 관리](#) (p. 377)을 참조하십시오.

제 EC2 인스턴스의 임시 보안 자격 증명에 액세스할 수 없습니다.

다음을 확인하십시오.

- Instance Metadata Service(IMDS)의 다른 부분에는 액세스할 수 있습니까? 액세스할 수 없는 경우 IMDS로의 요청에 대한 액세스를 차단하는 방화벽 규칙이 있는지 확인하십시오.

```
[ec2-user@domU-12-31-39-0A-8D-DE ~]$ GET http://169.254.169.254/latest/meta-data/hostname; echo
```

- IMDS의 iam 하위 트리가 있습니까? 하위 트리가 없는 경우 ec2:DescribeInstances를 호출하여 인스턴스에 연결된 IAM 인스턴스 프로필이 있는지 확인하십시오.

```
[ec2-user@domU-12-31-39-0A-8D-DE ~]$ GET http://169.254.169.254/latest/meta-data/iam; echo
```

- 오류가 있는지 IAM 하위 트리의 info 문서를 확인하십시오. 오류가 있는 경우 자세한 내용은 [IAM 하위 트리에서 info 문서의 오류란 무엇인가요?](#) (p. 474)를 참조하십시오.

```
[ec2-user@domU-12-31-39-0A-8D-DE ~]$ GET http://169.254.169.254/latest/meta-data/iam/info; echo
```

IAM 하위 트리에서 `info` 문서의 오류란 무엇인가요?

`iam/info` 문서는 "Code": "InstanceProfileNotFound". 를 나타냅니다.

IAM 인스턴스 프로필이 삭제되었으므로 Amazon EC2에서 더 이상 인스턴스에 자격 증명을 제공할 수 없습니다. Amazon EC2 인스턴스에 올바른 인스턴스 프로파일을 연결해야 합니다.

해당 이름의 인스턴스 프로파일이 있는 경우, 원래 인스턴스 프로파일이 삭제되고 동일한 이름의 다른 인스턴스가 생성된 것이 아닌지 확인하십시오.

1. IAM `GetInstanceProfile` 작업을 호출하여 `InstanceProfileId`를 가져옵니다.
2. Amazon EC2 `DescribeInstances` 작업을 호출하여 인스턴스의 `IamInstanceProfileId`를 가져옵니다.
3. IAM 작업의 `InstanceProfileId`와 Amazon EC2 작업의 `IamInstanceProfileId`가 일치하는지 확인합니다.

ID가 다르면 인스턴스에 연결된 인스턴스 프로파일이 더 이상 유효하지 않습니다. 인스턴스에 올바른 인스턴스 프로파일을 연결해야 합니다.

`iam/info` 문서는 성공을 나타내지만 "Message": "Instance Profile does not contain a role..."을 나타냅니다.

역할이 IAM `RemoveRoleFromInstanceProfile` 작업에 의해 인스턴스 프로필에서 제거되었습니다. IAM `AddRoleToInstanceProfile` 작업을 사용하여 인스턴스 프로필에 역할을 연결할 수 있습니다. 역할의 자격 증명에 액세스하려면 다음에 예정된 새로 고침까지 기다려야 합니다.

`iam/security-credentials/[role-name]` 문서는 "Code": "AssumeRoleUnauthorizedAccess"를 나타냅니다.

Amazon EC2에는 역할을 위임할 권한이 없습니다. 다음 예와 같이 역할을 수임할 권한은 해당 역할에 연결된 신뢰 정책에서 관리합니다. IAM `UpdateAssumeRolePolicy` API를 사용하여 신뢰 정책을 업데이트합니다.

```
{"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Principal": {"Service": ["ec2.amazonaws.com"]}, "Action": ["sts:AssumeRole"]}]}
```

역할의 자격 증명에 액세스하려면 다음에 예정된 자동 새로 고침까지 기다려야 합니다.

Amazon S3 및 IAM 문제 해결

이 문서의 정보를 사용하여 Amazon S3 및 IAM 작업 시 발생할 수 있는 문제를 해결하십시오.

Amazon S3 버킷에 대한 익명 액세스 권한을 부여하는 방법은 무엇입니까?

`principal` 요소에 와일드카드(*)를 지정하는 Amazon S3 버킷 정책을 사용합니다. 이는 누구나 버킷에 액세스할 수 있다는 의미입니다. 익명 액세스를 통하여 누구나(AWS 계정이 없는 사용자 포함) 버킷에 액세스

할 수 있게 됩니다. 샘플 정책은 Amazon Simple Storage Service 개발자 가이드의 [Amazon S3 버킷 정책 사례](#)를 참조하십시오.

AWS 계정 루트 사용자로 로그인했는데 내 계정으로 Amazon S3 버킷에 액세스할 수 없는 이유가 무엇입니까?

IAM 및 Amazon S3에 대해 모든 권한을 가진 IAM 사용자가 있기도 합니다. IAM 사용자가 Amazon S3 버킷에 버킷 정책을 할당하고 AWS 계정 루트 사용자를 보안 주체로 지정하지 않으면 루트 사용자의 버킷 액세스가 거부됩니다. 하지만 루트 사용자는 Amazon S3 콘솔 또는 AWS CLI를 사용해 루트 사용자 액세스를 허용하도록 버킷 정책을 수정하여 계속 버킷에 액세스할 수 있습니다.

AWS로 SAML 2.0 연동 문제 해결

이 문서의 정보를 사용하여 IAM 연동 및 SAML 2.0 작업 시 발생할 수 있는 문제를 진단하고 해결할 수 있습니다.

주제

- [오류: 요청에 잘못된 SAML 응답이 포함되어 있습니다. 로그아웃하려면 여기를 클릭하십시오.](#) (p. 475)
- [오류: AuthnResponse에 RoleSessionName 필요\(서비스: AWSSecurityTokenService, 상태 코드: 400, 오류 코드: InvalidIdentityToken\)](#) (p. 476)
- [오류: sts:AssumeRoleWithSAML을 수행할 권한 없음\(서비스: AWSSecurityTokenService, 상태 코드: 403, 오류 코드: AccessDenied\)](#) (p. 476)
- [오류: AuthnResponse에 RoleSessionName은 \[a-zA-Z_0-9+=,.@-\]{2,64}와 일치해야 함\(서비스: AWSSecurityTokenService, 상태 코드: 400, 오류 코드: InvalidIdentityToken\)](#) (p. 476)
- [오류: 유효하지 않은 응답 서명\(서비스: AWSSecurityTokenService, 상태 코드: 400, 오류 코드: InvalidIdentityToken\)](#) (p. 476)
- [오류: 역할을 수임하지 못함: 지정한 공급자에 발행자가 없음\(서비스: AWSOpenIdDiscoveryService, 상태 코드: 400, 오류 코드: AuthSamlInvalidSamlResponseException\)](#) (p. 477)
- [오류: 메타데이터를 구문 분석할 수 없습니다.](#) (p. 477)
- [오류: 요청된 DurationSeconds가 이 역할에 대해 설정된 MaxSessionDuration을 초과합니다.](#) (p. 477)
- [문제 해결을 위해 브라우저에서 SAML 응답을 보는 방법](#) (p. 477)

오류: 요청에 잘못된 SAML 응답이 포함되어 있습니다. 로그아웃하려면 여기를 클릭하십시오.

이 오류는 자격 증명 공급자의 SAML 응답에 NameID <https://aws.amazon.com/SAML/Attributes/Role>로 설정된 속성이 포함되지 않은 경우 발생할 수 있습니다. 이 속성은 하나 이상의 AttributeValue 요소를 포함해야 하며, 각 요소에 다음과 같은 문자열 쌍이 쉼표로 구분되어 있어야 합니다.

- 사용자를 매핑할 수 있는 역할의 ARN
- SAML 공급자의 ARN

자세한 내용은 [인증 응답을 위한 SAML 어설션 구성](#) (p. 181) 단원을 참조하십시오. 브라우저에서 SAML 응답을 보려면 [문제 해결을 위해 브라우저에서 SAML 응답을 보는 방법](#) (p. 477)의 단계를 따르십시오.

오류: AuthnResponse에 RoleSessionName 필요(서비스: AWSSecurityTokenService, 상태 코드: 400, 오류 코드: InvalidIdentityToken)

이 오류는 자격 증명 공급자의 SAML 응답에 `Name`이 <https://aws.amazon.com/SAML/Attributes/> `RoleSessionName`로 설정된 속성이 포함되지 않은 경우 발생할 수 있습니다. 속성 값은 사용자의 식별자이며, 일반적으로 사용자 ID 또는 이메일 주소입니다.

자세한 내용은 [인증 응답을 위한 SAML 어설션 구성 \(p. 181\)](#) 단원을 참조하십시오. 브라우저에서 SAML 응답을 보려면 문제 해결을 위해 브라우저에서 SAML 응답을 보는 방법 ([p. 477](#))의 단계를 따르십시오.

오류: sts:AssumeRoleWithSAML을 수행할 권한 없음 (서비스: AWSSecurityTokenService, 상태 코드: 403, 오류 코드: AccessDenied)

이 오류는 SAML 응답에 지정된 IAM 역할이 잘못 기재되었거나 존재하지 않는 경우 발생할 수 있습니다. 역할 이름은 대소문자를 구분하므로 역할 이름을 정확하게 사용하십시오. SAML 서비스 공급자 구성의 역할 이름을 올바르게 수정하십시오.

이 오류는 연동 사용자가 역할을 수임할 권한이 없는 경우에도 발생할 수 있습니다. 역할에는 IAM SAML 자격 증명 공급자의 ARN을 `Principal`로 지정하는 신뢰 정책이 있어야 합니다. 또한 역할에는 어떤 사용자가 해당 역할을 수임할 수 있는지 제어하는 조건이 포함됩니다. 사용자는 조건의 요구 사항을 준수해야 합니다.

이 오류는 SAML 응답에 `Subject`가 포함된 `NameID`가 없는 경우에도 발생할 수 있습니다.

자세한 내용은 [Establish Permissions in AWS for Federated Users](#) 및 [인증 응답을 위한 SAML 어설션 구성 \(p. 181\)](#) 단원을 참조하십시오. 브라우저에서 SAML 응답을 보려면 문제 해결을 위해 브라우저에서 SAML 응답을 보는 방법 ([p. 477](#))의 단계를 따르십시오.

오류: AuthnResponse에 RoleSessionName은 [a-zA-Z_0-9+=,.@-]{2,64}와 일치해야 함(서비스: AWSSecurityTokenService, 상태 코드: 400, 오류 코드: InvalidIdentityToken)

이 오류는 `RoleSessionName` 속성 값이 너무 길거나 유효하지 않은 문자가 포함된 경우 발생할 수 있습니다. 유효한 최대 길이는 64자입니다.

자세한 내용은 [인증 응답을 위한 SAML 어설션 구성 \(p. 181\)](#) 단원을 참조하십시오. 브라우저에서 SAML 응답을 보려면 문제 해결을 위해 브라우저에서 SAML 응답을 보는 방법 ([p. 477](#))의 단계를 따르십시오.

오류: 유효하지 않은 응답 서명(서비스: AWSSecurityTokenService, 상태 코드: 400, 오류 코드: InvalidIdentityToken)

이 오류는 자격 증명 공급자의 연동 메타데이터가 IAM 자격 증명 공급자의 메타데이터와 일치하지 않는 경우 발생할 수 있습니다. 예를 들어, 만료된 인증서를 업데이트하기 위해 자격 증명 서비스 공급자의 메타데이터 파일이 변경되었을 수 있습니다. 이 경우, 자격 증명 서비스 공급자의 업데이트된 SAML 메타데이터 파일

을 다운로드합니다. 그런 다음 `aws iam update-saml-provider` 크로스플랫폼 CLI 명령 또는 `Update-IAMSSMProvider` PowerShell cmdlet을 통해 IAM에서 정의한 AWS 자격 증명 공급자 엔터티에 이를 업데이트합니다.

오류: 역할을 수임하지 못함: 지정한 공급자에 발행자가 없음(서비스: AWSOpenIdDiscoveryService, 상태 코드: 400, 오류 코드: AuthSamlInvalidSamlResponseException)

이 오류는 IAM에서 자격 증명 공급자를 생성할 때 AWS에 업로드한 연동 메타데이터 파일에 선언되어 있는 발행자와 SAML 응답의 발행자가 일치하지 않는 경우 발생할 수 있습니다.

오류: 메타데이터를 구문 분석할 수 없습니다.

이 오류는 메타데이터 파일이 적절한 형식이 아닌 경우에 발생할 수 있습니다.

AWS Management 콘솔에서 [SAML 자격 증명 공급자를 생성하거나 관리할 때](#) (p. 178), 사용자의 자격 증명 공급자에서 SAML 메타데이터 문서를 가져와야 합니다. 이 메타데이터 파일에는 발급자 이름, 만료 정보 및 IdP에서 가져온 SAML 인증 응답(어설션)을 확인하는 데 사용할 수 있는 키가 포함되어 있습니다. 메타데이터 파일은 바이트 순서 표시(BOM)가 없는 UTF-8 형식으로 인코딩되어야 합니다. 또한 SAML 메타데이터 문서의 일부로 포함된 x.509 인증서는 1,024비트 이상의 키를 사용해야 합니다. 키 크기가 이보다 작으면 "메타데이터를 구문 분석할 수 없음" 오류로 인해 IdP 생성에 실패합니다. BOM을 제거하려면 Notepad++와 같은 텍스트 편집 도구를 사용해 파일을 UTF-8로 인코딩합니다.

오류: 요청된 DurationSeconds가 이 역할에 대해 설정된 MaxSessionDuration을 초과합니다.

이 오류는 AWS CLI 또는 API에서 역할을 위임한 경우 발생할 수 있습니다.

`assume-role-with-saml` CLI 또는 `AssumeRoleWithSAML` API 작업을 사용하여 역할을 위임하는 경우 `DurationSeconds` 파라미터의 값을 지정할 수 있습니다. 이 값의 범위는 900초(15분)에서 해당 역할에 대한 최대 세션 기간 설정까지 지정할 수 있습니다. 이 설정보다 높게 값을 지정하면 작업에 실패합니다. 예를 들어 세션 기간으로 12시간을 지정했는데 관리자가 최대 세션 기간으로 6시간을 설정하면 작업에 실패합니다. 역할에 대한 최댓값을 확인하는 방법을 알아보려면 [역할에 대한 최대 세션 기간 설정 보기](#) (p. 228) 단원을 참조하십시오.

문제 해결을 위해 브라우저에서 SAML 응답을 보는 방법

다음 절차는 SAML 2.0 관련 문제를 해결할 경우 브라우저에서 서비스 공급자로부터의 SAML 응답을 보는 방법을 설명합니다.

브라우저에서 문제를 재현할 수 있는 페이지로 이동합니다. 그런 다음 해당 브라우저의 단계를 따릅니다.

주제

- [Google Chrome](#) (p. 478)
- [Mozilla Firefox](#) (p. 478)
- [Apple Safari](#) (p. 478)
- [Microsoft Internet Explorer](#) (p. 478)
- [Base64 인코딩 SAML 응답에 대해 해야 할 작업](#) (p. 479)

Google Chrome

Chrome에서 SAML 응답을 보려면

이 단계는 54.0.2840.87m 버전을 사용하여 테스트했습니다. 다른 버전을 사용할 경우 그에 맞게 단계를 적용해야 할 수 있습니다.

1. F12를 눌러 개발자 콘솔을 시작합니다.
2. 네트워크 탭을 선택한 후 로그 보관(Preserve log)을 선택합니다.
3. 문제를 재현합니다.
4. 개발자 콘솔 창에서 SAML 게시물(SAML Post)을 확인합니다. 해당 행을 선택하고 하단에서 헤더(Headers) 탭을 봅니다. 인코딩된 요청을 포함하는 SAMLResponse 속성을 확인합니다.

Mozilla Firefox

Firefox에서 SAML 응답을 보려면

이 절차는 37.0.2 of Mozilla Firefox에서 테스트했습니다. 다른 버전을 사용할 경우 그에 맞게 단계를 적용해야 할 수 있습니다.

1. F12를 눌러 개발자 콘솔을 시작합니다.
2. 개발자 콘솔 창 상단 오른쪽에서 옵션(작은 기어 모양 아이콘)을 클릭합니다. 공통 기본 설정(Common Preferences)에서 지속적 로그 활성화(Enable persistent logs)를 선택합니다.
3. 네트워크 탭을 선택합니다.
4. 문제를 재현합니다.
5. 테이블에서 POST SAML을 확인합니다. 해당 행을 선택합니다. 오른쪽의 양식 데이터(Form Data) 창에서 Params 탭을 선택하고 SAMLResponse 요소를 확인합니다.

Apple Safari

Safari에서 SAML 응답을 보려면

이 단계는 8.0.6(10600.6.3) 버전을 사용하여 테스트했습니다. 다른 버전을 사용할 경우 그에 맞게 단계를 적용해야 할 수 있습니다.

1. Safari에서 Web Inspector을 사용하도록 설정합니다. 기본 설정 창을 열고 고급 탭을 선택한 후 메뉴 표시줄에 Develop 메뉴 표시>Show Develop menu in the menu bar)를 선택합니다.
2. 이제 Web Inspector을 열 수 있습니다. Develop를 클릭한 후 웹 검사기 표시>Show Web Inspector)를 선택합니다.
3. 리소스 탭을 선택합니다.
4. 문제를 재현합니다.
5. saml-signin.aws.amazon.com 요청을 확인합니다.
6. 아래로 스크롤하여 Request Data라는 SAMLResponse를 확인합니다. 연결된 값은 Base64 인코딩 응답입니다.

Microsoft Internet Explorer

Internet Explorer에서 SAML 응답을 보려면

Internet Explorer의 네트워크 트래픽을 분석하는 가장 좋은 방법은 타사 도구를 사용하는 것입니다.

- <http://social.technet.microsoft.com/wiki/contents/articles/3286.ad-fs-2-0-how-to-use-fiddler-web-debugger-to-analyze-a-ws-federation-passive-sign-in.aspx>의 단계에 따라 Fiddler를 다운로드하여 설치한 후 데이터를 수집하십시오.

Base64 인코딩 SAML 응답에 대해 해야 할 작업

브라우저에서 Base64 인코딩 SAML 응답 요소를 확인했으면 복사한 후 Base-64 디코딩 도구에서 사용하여 XML 태그 응답을 추출합니다.

보안 팁

표시되는 SAML 응답 데이터에는 중요한 보안 데이터가 포함되어 있을 수 있으므로 온라인 base64 디코더를 사용하지 않을 것을 권장합니다. 대신 로컬 컴퓨터에 설치된 도구를 사용하십시오. 로컬 컴퓨터에 설치된 도구는 네트워크를 통해 SAML 데이터를 전송하지 않습니다.

Windows 시스템용 내장 옵션(PowerShell):

```
PS C:\> [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String("base64encodedtext"))
```

MacOS 및 Linux 시스템용 내장 옵션:

```
$ echo "base64encodedtext" | base64 --decode
```

AWS Identity and Access Management에 대한 참조 정보

이 단원의 주제를 통해 IAM 및 AWS STS의 다양한 측면에 대한 자세한 참조 자료를 찾아보십시오.

주제

- [IAM 식별자 \(p. 480\)](#)
- [IAM 개체 및 객체에 대한 제한 \(p. 485\)](#)
- [IAM로 작업하는 AWS 서비스 \(p. 488\)](#)
- [IAM JSON 정책 참조 \(p. 498\)](#)

IAM 식별자

IAM은(는) 사용자, 그룹, 역할, 정책 및 서버 인증서에 대해 몇 가지 다른 식별자를 사용합니다. 이 단원에서는 그러한 식별자와 각 식별자를 사용하는 경우를 설명합니다.

주제

- [표시 이름 및 경로 \(p. 480\)](#)
- [IAM ARN \(p. 480\)](#)
- [고유 ID \(p. 483\)](#)

표시 이름 및 경로

사용자, 역할, 그룹 또는 정책을 생성하거나 서버 인증서를 업로드할 때, Bob, TestApp1, Developers, ManageCredentialsPermissions 또는 ProdServerCert 등의 표시 이름을 부여합니다.

IAM API 또는 AWS Command Line Interface(AWS CLI)를 사용하여 IAM 엔터티를 생성하는 경우, 해당 엔터티에 선택적 경로를 부여할 수도 있습니다. 하나의 경로를 사용하거나, 하나의 폴더 구조인 것처럼 여러 경로를 중첩할 수 있습니다. 예를 들면 중첩된 경로 /division_abc/subdivision_xyz/product_1234/engineering/를 사용하여 귀하 회사의 조직 구조를 일치시킬 수 있습니다. 그런 다음 정책을 생성하여 그 경로의 모든 사용자가 정책 시뮬레이터 API에 액세스할 수 있도록 허용할 수 있습니다. 정책을 보려면 [IAM: 사용자 경로를 바탕으로 정책 시뮬레이터 API 액세스 \(p. 366\)](#) 단원을 참조하십시오. 경로를 사용하는 방법의 추가 예제는 [IAM ARN \(p. 480\)](#) 단원을 참조하십시오.

어떤 사용자 및 그룹에 동일한 경로를 부여했다고 해서 해당 사용자가 그룹에 자동으로 추가되지는 않습니다. 예를 들어, Developers 그룹을 생성하고 이 그룹의 경로를 /division_abc/subdivision_xyz/product_1234/engineering/으로 지정할 수 있습니다. Bob이라는 사용자를 만들고 그에게 동일한 경로를 부여했다고 해서 Bob이 Developers 그룹에 자동으로 추가되지는 않습니다. IAM는 경로를 기반으로 사용자 또는 그룹 간에 경계를 적용하지 않습니다. 경로가 다른 사용자가 (해당 리소스에 대한 권한이 있다는 가정하에) 동일한 리소스를 사용할 수 있습니다. 이름의 제한 사항에 대한 자세한 내용은 [IAM 개체 및 객체에 대한 제한 \(p. 485\)](#) 단원을 참조하십시오.

IAM ARN

대부분의 리소스에는 표시 이름이 있습니다(예를 들어, Bob이라는 사용자 또는 Developers라는 그룹). 그러나 권한 정책 언어에는 다음과 같은 ARN(Amazon 리소스 이름) 형식을 사용하여 하나 이상의 리소스를 지정해야 합니다.

```
arn:partition:service:region:account:resource
```

여기서 각 항목은 다음과 같습니다.

- partition은 리소스가 위치하는 파티션을 식별합니다. 표준 AWS 리전에서 파티션은 aws입니다. 리소스가 다른 파티션에 있는 경우 파티션은 aws-*partitionname*입니다. 예를 들어 중국(베이징) 리전에 있는 리소스의 파티션은 aws-cn입니다.
- service에서는 AWS 제품을 식별합니다. IAM 리소스의 경우 항상 iam입니다.
- region은 리소스가 상주하는 리전입니다. IAM 리소스의 경우 항상 공백입니다.
- account는 AWS 계정 ID이며 하이픈은 제외합니다(예: 123456789012).
- resource는 특정 리소스를 이름으로 식별하는 부분입니다.

사용자(IAM 및 연동), 그룹, 역할, 정책, 인스턴스 프로파일, 가상 MFA 디바이스 및 [서버 인증서 \(p. 141\)](#)에 대해 IAM에서 ARN을 사용할 수 있습니다. 다음 표는 각각에 대한 ARN 형식과 예시를 보여 줍니다. IAM 리소스가 글로벌이기 때문에 ARN의 리전 부분은 공백입니다.

Note

다음과 같은 많은 예에는 ARN의 리소스 부분에 경로가 포함됩니다. AWS Management 콘솔에서 경로를 생성하거나 조작할 수 없습니다. 경로를 사용하려면 AWS API나 AWS CLI 또는 Windows PowerShell용 도구을(를) 사용하여 리소스 작업을 해야 합니다.

다음 예는 다른 유형의 IAM 리소스에 대한 ARN을 보여 줍니다.

- AWS 계정 - 계정 자체:

```
arn:aws:iam::123456789012:root
```

- 계정의 IAM 사용자:

```
arn:aws:iam::123456789012:user/Bob
```

- 조직 차트를 반영하는 경로를 갖는 다른 사용자:

```
arn:aws:iam::123456789012:user/division_abc/subdivision_xyz/Bob
```

- IAM 그룹:

```
arn:aws:iam::123456789012:group/Developers
```

- 경로를 포함하는 IAM 그룹:

```
arn:aws:iam::123456789012:group/division_abc/subdivision_xyz/product_A/Developers
```

- IAM 역할:

```
arn:aws:iam::123456789012:role/S3Access
```

- 관리형 정책:

```
arn:aws:iam::123456789012:policy/ManageCredentialsPermissions
```

- EC2 인스턴스와 연결될 수 있는 인스턴스 프로파일:

```
arn:aws:iam::123456789012:instance-profile/Webserver
```

- IAM에서 'Bob'으로 식별되는 연동 사용자:

```
arn:aws:sts::123456789012:federated-user/Bob
```

- 역할 세션 이름 'Mary'로 역할 'Accounting-Role'을 수임하는 누군가의 활성 세션:

```
arn:aws:sts::123456789012:assumed-role/Accounting-Role/Mary
```

- 사용자 이름 Bob에 할당된 멀티 팩터 인증 디바이스:

```
arn:aws:iam::123456789012:mfa/Bob
```

- 서버 인증서

```
arn:aws:iam::123456789012:server-certificate/ProdServerCert
```

- 조직 차트를 반영하는 경로를 갖는 서버 인증서:

```
arn:aws:iam::123456789012:server-certificate/division_abc/subdivision_xyz/  
ProdServerCert
```

- 자격 증명 공급자(SAML 및 OIDC):

```
arn:aws:iam::123456789012:saml-provider/ADFSProvider
```

```
arn:aws:iam::123456789012:oidc-provider/GoogleProvider
```

다음 예에서는 Richard가 자신의 액세스 키를 관리할 수 있도록 그에게 할당할 수 있는 정책을 보여줍니다. 리소스는 IAM 사용자 Richard입니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "ManageRichardAccessKeys",  
            "Effect": "Allow",  
            "Action": [  
                "iam:*AccessKey*",  
                "iam:GetUser"  
            ],  
            "Resource": "arn:aws:iam::*:user/division_abc/subdivision_xyz/Richard"  
        },  
        {  
            "Sid": "ListForConsole",  
            "Effect": "Allow",  
            "Action": "iam>ListUsers",  
            "Resource": "*"  
        }  
    ]  
}
```

Note

ARN을 사용하여 IAM 정책의 리소스를 식별할 때, ARN의 일부로 정책 변수를 포함시켜 런타임 정 보(예: 사용자 이름)에 대한 자리 표시자를 넣을 수 있습니다. 자세한 내용은 [IAM 정책 요소: 변수 및 태그 \(p. 524\)](#) 단원을 참조하십시오.

ARN의 **###** 부분에 와일드카드를 사용하여 여러 사용자, 그룹 또는 정책을 지정할 수 있습니다. 예를 들어, product_1234를 작업하는 모든 사용자를 지정하려면 다음을 사용합니다.

```
arn:aws:iam::123456789012:user/division_abc/subdivision_xyz/product_1234/*
```

이름이 app_ 문자열로 시작하는 사용자가 여럿 있다고 가정해 봅시다. 다음과 같은 ARN을 사용하여 이를 모두를 언급할 수 있습니다.

```
arn:aws:iam::123456789012:user/division_abc/subdivision_xyz/product_1234/app_*
```

AWS 계정의 모든 사용자, 그룹 또는 정책을 지정하려면, ARN의 user/, group/ 또는 policy 부분 다음에 각각 와일드카드를 사용합니다.

```
arn:aws:iam::123456789012:user/*
arn:aws:iam::123456789012:group/*
arn:aws:iam::123456789012:policy/*
```

ARN의 user/, group/ 또는 policy 부분에 와일드카드를 사용하지 마십시오. 즉 다음과 같이 사용할 수 없습니다.

```
arn:aws:iam::123456789012:u*
```

Example 프로젝트 기반 그룹의 경로 및 ARN 사용

AWS Management 콘솔에서는 경로를 생성하거나 조작할 수 없습니다. 경로를 사용하려면 AWS API나 AWS CLI 또는 Windows PowerShell용 도구을(를) 사용하여 리소스 작업을 해야 합니다.

이 예에서 Marketing_Admin 그룹의 Jules는 /marketing/ 경로 내에 프로젝트 기반 그룹을 생성한 다음, 회사의 여러 부서에 있는 사용자를 이 그룹에 할당합니다. 이 예는 사용자의 경로가 그가 속한 그룹과 관련되지 않는다는 점을 보여줍니다.

마케팅 그룹에는 이들이 출시할 신제품이 있으므로 Jules는 /marketing/ 경로에 Widget_Launch라는 새 그룹을 생성합니다. 그런 다음 Jules는 이 그룹에 다음과 같은 정책을 할당합니다. 이 정책은 그룹에 이 특정 출시에 지정된 example_bucket 부분의 객체에 대한 액세스 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::example_bucket/marketing/newproductlaunch/widget/*"
    },
    {
      "Effect": "Allow",
      "Action": "s3>ListBucket*",
      "Resource": "arn:aws:s3:::example_bucket",
      "Condition": {"StringLike": {"s3:prefix": "marketing/newproductlaunch/widget/*"}}
    }
  ]
}
```

그런 다음 Jules는 해당 그룹에 이 출시와 관련된 사용자를 할당합니다. 여기에는 /marketing/ 경로의 Patricia와 Eli가 포함됩니다. 또한 /sales/ 경로의 Chris와 Chloe, /legal/ 경로의 Aline과 Jim이 포함됩니다.

고유 ID

IAM에서 사용자, 그룹, 역할, 정책, 인스턴스 프로파일 또는 서버 인증서를 생성할 때, 각 엔터티에 다음 예와 같은 고유 ID를 할당합니다.

AIDAJQABLZS4A3QDU576Q

대개 IAM 엔터티를 사용하는 경우 표시 이름 및 ARN을 사용하므로, 특정 엔터티에 대한 고유 ID를 알지 않아도 됩니다. 그러나 표시 이름을 사용하는 것이 적절치 않은 경우에는 고유 ID를 사용하는 것이 유용합니다.

한 예로써 AWS 계정에서 표시 이름을 재사용하는 경우를 살펴보겠습니다. 계정 내에서 사용자, 그룹 또는 정책의 표시 이름은 고유해야 합니다. 예를 들어, David라는 IAM 사용자를 생성할 수 있습니다. 회사에서 Amazon S3를 사용하고 있고 각 직원에 대한 폴더가 포함된 버킷이 있다고 합시다. 이 버킷에는 사용자가 버킷에서 자신의 폴더에만 액세스할 수 있도록 하는 리소스 기반 정책(버킷 정책)이 있습니다. David라는 직원이 퇴사하여 해당하는 IAM 사용자를 삭제한다고 가정해 봅시다. 그러나 이후 David라는 또 다른 직원이 입사

하여 David라는 새 IAM 사용자를 생성합니다. 버킷 정책에서 David라는 IAM 사용자를 지정하면, 해당 정책은 결국 새로운 David에게 이전 David가 남긴 Amazon S3 버킷의 정보에 대한 액세스 권한을 부여하게 됩니다.

그러나 이전에 삭제한 표시 이름을 재사용하는 새 IAM 사용자를 생성한다 하더라도 모든 IAM 사용자는 고유 ID를 갖습니다. 이 예에서 이전 IAM 사용자 David와 새 IAM 사용자 David는 서로 다른 고유 ID를 갖습니다. 사용자 이름이 아닌 고유 ID로 액세스 권한을 부여하는 Amazon S3 버킷 리소스 정책을 생성하면, 액세스해서는 안 되는 정보에 대한 권한을 직원에게 실수로 부여할 가능성을 줄일 수 있습니다.

사용자 ID가 유용한 또 다른 예는 IAM 사용자 정보의 데이터베이스(또는 다른 저장소)를 유지하는 경우입니다. 앞의 예와 마찬가지로 차후 어떤 이름을 재사용하는 IAM 사용자가 생길지라도 고유 ID는 생성하는 각 IAM 사용자에 대한 고유 식별자를 제공할 수 있습니다.

고유 ID 접두사에 대한 이해

IAM에서는 다음과 같은 접두사를 사용해 각 고유 ID가 적용되는 엔터티의 유형을 표시합니다.

접두사	엔터티 유형
AAGA	작업 그룹
ACCA	Context Specific Credential
AGPA	그룹
AIDA	IAM user
AIPA	Amazon EC2 인스턴스 프로파일
AKIA	액세스 키
ANPA	관리형 정책
ANVA	관리 정책 내 버전
APKA	퍼블릭 키
AROA	역할
ASCA	Certificate
ASIA	임시(AWS STS) 키

고유 ID 가져오기

IAM 엔터티의 고유 ID는 IAM 콘솔에서 제공되지 않습니다. 고유 ID를 가져오기 위해 다음과 같은 AWS CLI 명령 또는 IAM API 호출을 사용할 수 있습니다.

AWS CLI:

- [get-group](#)
- [get-role](#)
- [get-user](#)
- [get-policy](#)
- [get-instance-profile](#)
- [get-server-certificate](#)

IAM API:

- [GetGroup](#)
- [GetRole](#)
- [GetUser](#)
- [GetPolicy](#)
- [GetInstanceProfile](#)
- [GetServerCertificate](#)

IAM 개체 및 객체에 대한 제한

IAM 엔터티 및 객체에는 크기 제한이 있습니다. IAM은 엔터티 이름 지정 방법, 생성할 수 있는 엔터티의 수, 엔터티에서 사용할 수 있는 문자 수를 제한합니다.

Note

실체 사용 및 할당량에 관한 계정 수준 정보를 가져오려면 [GetAccountSummary](#) API 작업 또는 `get-account-summary` AWS CLI 명령을 사용합니다.

IAM 엔터티 이름 제한

다음은 IAM 이름에 대한 제한 사항입니다.

- 정책 설명서에는 수평 탭(U+0009), 라인 피드(U+000A), 캐리지 리턴(U+000D), 그리고 U+0020 ~ U+00FF 범위의 문자 등 유니코드 문자만 넣을 수 있습니다.
- 사용자, 그룹, 역할, 정책, 인스턴스 프로파일 및 서버 인증서 이름은 더하기(+), 등호(=), 콤마(,), 마침표(.), at(@), 밑줄(_), 하이픈(-)을 포함하는 영숫자여야 합니다.
- 사용자, 그룹 및 역할의 이름은 계정 내에서 고유해야 합니다. 대소문자는 구분하지 않습니다. 예를 들어 그룹 **ADMINs**와 그룹 **admins**를 모두 만들 수는 없습니다.
- 타사에서 역할을 맡기 위해 사용하는 외부 ID 값은 최소 2자 이상, 최대 1,224자 이상이어야 합니다. 이 값은 공백 없이 영숫자여야 합니다. 이 값은 더하기(+), 등호(=), 쉼표(.) 마침표(.), 기호(@), 콜론(:), 슬래시(/) 및 하이픈(-)과 같은 기호도 포함할 수 있습니다. 외부 ID에 대한 자세한 내용은 [AWS 리소스에 대한 액세스를 타사에 부여할 때 외부 ID를 사용하는 방법](#) (p. 206)을 참조하십시오.
- 경로 이름은 슬래시(/)로 시작하고 끝나야 합니다.
- [인라인 정책](#) (p. 312)의 정책 이름은 이러한 정책이 포함된 사용자, 그룹 또는 역할에 대해 고유해야 합니다. 이름에는 라틴어 기본(ASCII) 문자가 포함될 수 있는데 예약된 문자인 역슬래시(\), 슬래시(/), 별표(*), 물음표(?), 공백이 없어야 합니다. 이러한 문자는 [RFC 3986](#)에 따라 설정됩니다.
- 사용자 암호(로그인 프로필)에는 라틴어 기본(ASCII) 문자가 포함될 수 있습니다.
- AWS 계정 ID 별칭은 AWS 제품 전체에서 고유해야 하며, 다음 DNS 명명 규칙을 따르는 영숫자여야 합니다. 별칭은 소문자여야 하며, 하이픈으로 시작하거나 끝나면 안 되고, 2개의 하이픈이 연속으로 있으면 안 되고, 12자리 숫자는 안 됩니다.

로마자 기본(ASCII) 문자 목록을 보려면 [의회 도서관 로마자 기본\(ASCII\) 코드 표](#)를 확인하십시오.

IAM 엔터티 객체 제한

AWS를 이용하면 기본 IAM 엔터티 제한에 증가를 요청할 수 있습니다. 이러한 기본 제한에 대하여 한도 상승을 요청하는 방법에 대한 자세한 내용은 Amazon Web Services 일반 참조 설명서의 [AWS 서비스 제한](#) 단원을 참조하십시오.

IAM 엔터티에 대한 기본 제한:

리소스	기본 한도
AWS 계정 내 고객 관리형 정책	1500
AWS 계정 내 그룹	300
AWS 계정 내 역할	1000
IAM 역할에 연결된 관리형 정책	10
IAM 사용자에 연결된 관리형 정책	10
AWS 계정 내 가상 MFA 장치(할당된 또는 할당되지 않은 상태)	계정에 대한 사용자 할당량과 동일
AWS 계정 내 인스턴스 프로파일	1000
하나의 AWS 계정에 저장되는 서버 인증서	20

다음 제한에 대한 한도 상승을 요청할 수 없습니다.

IAM 엔터티에 대한 제한:

리소스	제한
한 명의 IAM 사용자에게 할당되는 액세스 키	2
AWS 계정 루트 사용자에 할당되는 액세스 키	2
AWS 계정당 별칭	1
IAM 사용자 한 명이 소속될 수 있는 그룹 수	10
그룹의 IAM 사용자 수	계정에 대한 사용자 할당량과 동일
AWS 계정 내 사용자	5000(다수의 사용자를 추가해야 한다면 임시 보안 자격 증명 (p. 263)의 사용을 고려할 수 있습니다.)
IAM SAML 공급자 객체와 연결된 자격 증명 공급자 (IdP)	10
SAML 제공자당 키	10
IAM 사용자당 로그인 프로필	1
IAM 그룹에 연결된 관리형 정책	10
IAM 사용자에 대한 권한 경계	1
IAM 역할에 대한 권한 경계	1
한 명의 IAM 사용자가 사용하는 MFA 디바이스	1
AWS 계정 루트 사용자당 사용하는 MFA 디바이스:	1
인스턴스 프로파일 내 역할	1
AWS 계정당 SAML 제공자	100
한 명의 IAM 사용자에게 할당되는 서명 인증서	2

리소스	제한
한 명의 IAM 사용자에게 할당되는 SSH 퍼블릭 키	5
IAM 역할에 연결할 수 있는 태그	50
IAM 사용자에 연결할 수 있는 태그	50
저장할 수 있는 관리형 정책의 버전 수	5

IAM 엔터티 문자 제한

다음은 엔터티에 대한 최대 길이입니다.

설명	한도
경로	512자
사용자 이름	64자
그룹 이름	128자
역할 이름	64자
	<p style="text-align: center;">Important</p> <p>AWS 콘솔에서 역할 전환 기능이 있는 역할을 사용하려면 Path와 RoleName을 합해 64자를 초과할 수 없습니다.</p>
태그 키	128자
태그 값	256자
인스턴스 프로파일 이름	128자
IAM에서 생성된 고유 ID, 예:	128자
<ul style="list-style-type: none"> • AIDA로 시작되는 사용자 ID • AGPA로 시작되는 그룹 ID • AROA로 시작되는 역할 ID • ANPA로 시작되는 관리형 정책 ID • ASCA로 시작되는 서버 인증서 ID <p>Note</p> <p>이는 포괄적인 목록을 제공하기 위한 것이 아니며 특정 유형의 ID가 지정된 문자 조합으로만 시작됨을 보장하지도 않습니다.</p>	
정책 이름	128자
로그인 프로필의 암호	1~128자

설명	한도
AWS 계정 ID의 별칭	3~63자
역할 신뢰 정책 JSON 텍스트(역할을 맡을 수 있는 사람을 결정하는 정책)	2,048자
역할 세션 이름	64자
역할 세션 기간	12시간 AWS CLI 또는 API에서 역할을 위임할 때 duration-seconds CLI 파라미터 또는 DurationSeconds API 파라미터를 사용해 더 긴 역할 세션을 요청할 수 있습니다. 이 값은 900 초(15분)에서 역할에 대한 최대 세션 기간 설정 까지 지정할 수 있습니다. 이 설정은 1시간~12 시간일 수 있습니다. 역할에 대한 최댓값을 확인하는 방법을 알아보려면 역할에 대한 최대 세션 기간 설정 보기 (p. 228) 단원을 참조하십시오. DurationSeconds 파라미터의 값을 지정하지 않으면 보안 자격 증명이 한 시간 동안 유효하게 됩니다.
인라인 정책 (p. 312):	원하는 만큼의 인라인 정책을 IAM 사용자, 역할 또는 그룹에게 추가할 수 있습니다. 단, 엔터티당 총 누적 정책 크기(모든 인라인 정책의 합)은 다음 한계를 초과할 수 없습니다. <ul style="list-style-type: none">• 사용자 정책 크기는 2,048자를 초과할 수 없음• 역할 정책 크기는 10,240자를 초과할 수 없음• 그룹 정책 크기는 5,120자를 초과할 수 없음
관리형 정책 (p. 312):	<p>Note</p> <p>IAM은 이러한 한계를 기준으로 정책의 크기를 계산할 때 공백을 계수하지 않습니다.</p> <ul style="list-style-type: none">• IAM 사용자, 역할 또는 그룹당 최대 10개의 관리형 정책을 추가할 수 있습니다.• 각 관리 정책의 크기는 6,144자를 초과할 수 없습니다. <p>Note</p> <p>IAM은 이 한계를 기준으로 정책의 크기를 계산할 때 공백을 계수하지 않습니다.</p>

IAM로 작업하는 AWS 서비스

아래 나열된 AWS 서비스는 [AWS 제품 범주](#)에 의해 그룹화되며, 지원되는 IAM 기능에 대한 정보를 포함합니다.

- 서비스 – 서비스의 이름을 선택하여 해당 서비스의 IAM 권한 부여 및 액세스에 대한 AWS 문서를 볼 수 있습니다.
- 작업 – 정책에서 개별 작업을 지정할 수 있습니다. 서비스에서 이 기능을 지원하지 않는 경우 [시각적 편집기\(visual editor\) \(p. 379\)](#)의 모든 작업(All actions)이 선택됩니다. JSON 정책 문서의 * 요소에서 Action를 사용해야 합니다. 각 서비스의 작업 목록은 ??? 단원을 참조하십시오.
- 리소스 수준 권한 – ARN을 사용하여 정책에서 개별 리소스를 지정할 수 있습니다. 서비스에서 이 기능을 지원하지 않는 경우 [정책 시각적 편집기 \(p. 379\)](#)에 모든 리소스(All resources)가 선택됩니다. JSON 정책 문서의 * 요소에서 Resource를 사용해야 합니다. List* 작업과 같은 일부 작업은 ARN 지정을 지원하지 않습니다. 여러 리소스를 반환하기로 설계되었기 때문입니다. 서비스에서 일부 리소스에 대해서만 이 기능을 지원하지 않는 경우 표의 노란색 셀에 표시됩니다. 자세한 정보는 서비스에 대한 문서 단원을 참조하십시오.
- 리소스 기반 정책 – 리소스 기반 정책을 서비스 내 리소스에 연결할 수 있습니다. 리소스 기반 정책은 해당 리소스에 액세스할 수 있는 IAM 자격 증명을 지정하는 Principal 요소를 포함합니다. 자세한 정보는 [자격 증명 기반 정책 및 리소스 기반 정책 \(p. 326\)](#) 단원을 참조하십시오.
- 태그 기반 권한 부여 – 정책의 조건에서 [리소스 태그](#)를 사용할 수 있습니다. 예를 들면, 태그 지정된 [Amazon RDS 리소스에 대한 모든 액세스를 태그 소유자에게 허용하는 정책 \(p. 371\)](#)을 생성해야 합니다. rds:db-tag/Owner 같은 조건 키를 사용하여 이 작업을 수행할 수 있습니다.
- 임시 자격 증명 – 연동, 교차 계정 역할 또는 [서비스 역할 \(p. 154\)](#)을 사용하여 로그인한 사용자는 이 서비스에 액세스할 수 있습니다. AssumeRole 또는 GetFederationToken 같은 AWS STS API 작업을 호출하여 임시 보안 자격 증명을 가져옵니다. 자세한 정보는 [임시 보안 자격 증명 \(p. 263\)](#) 단원을 참조하십시오.
- 서비스 연결 역할 – [서비스 연결 역할 \(p. 154\)](#)은 사용자를 대신하여 작업을 완료하기 위해 다른 서비스의 리소스에 액세스할 수 있는 서비스 권한을 부여합니다. 이러한 역할을 지원하는 서비스에 대한 설명서를 보려면 [yes] 링크를 선택합니다. 자세한 정보는 [서비스 연결 역할 사용 \(p. 195\)](#) 단원을 참조하십시오.
- 추가 정보 – 서비스가 기능을 완전히 지원하지 않는 경우 항목에 대한 각주를 검토하여 제한 사항과 관련 정보의 링크를 볼 수 있습니다.

컴퓨팅 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
Application Auto Scaling	예	아니요	아니요	아니요	예	예
AWS Auto Scaling	예	아니요	아니요	아니요	예	예
AWS Batch	예	예	아니요	아니요	예	아니요
Amazon Elastic Compute Cloud (Amazon EC2)	예	예	아니요	예	예	예 ¹
Amazon EC2 Auto Scaling	예	예	아니요	아니요	예	예
AWS Elastic Beanstalk	예	예	아니요	예	예	예
Amazon Elastic Container Registry (Amazon ECR)	예	예	예	아니요	예	아니요
Amazon Elastic Container Service (Amazon ECS)	예	예	아니요	아니요	예	예
Amazon Elastic Container Service for Kubernetes (Amazon EKS)	예	아니요	아니요	아니요	예	아니요
Amazon Elastic Inference	예	예	예	아니요	아니요	아니요

Elastic Load Balancing	예	예	아니요	아니요	예	예
AWS Lambda	예	예	예	아니요	예	예
Amazon Lightsail	예	아니요	아니요	아니요	예	아니요

¹ Amazon EC2 서비스 연결 역할은 AWS Management 콘솔을 사용하여 생성할 수 없으며 [예약된 인스턴스](#), [스팟 인스턴스 요청](#), [스팟 집합 요청](#) 기능에 대해서만 사용할 수 있습니다.

스토리지 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
AWS Backup	예	예	예	아니요	예	아니요
Amazon Elastic Block Store (Amazon EBS)	예	예	아니요	예	예	아니요
Amazon Elastic File System (Amazon EFS)	예	예	아니요	아니요	예	아니요
Amazon S3 Glacier	예	예	예	예	예	아니요
AWS Import/Export	예	아니요	아니요	아니요	예	아니요
AWS Migration Hub	예	예	아니요	아니요	예	아니요
Amazon Simple Storage Service (Amazon S3)	예	예	예	예	예	아니요
AWS Snowball	예	아니요	아니요	아니요	예	아니요
AWS Snowball 엣지	예	아니요	아니요	아니요	아니요	아니요
AWS Storage Gateway	예	예	아니요	아니요	예	아니요

데이터베이스 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
Amazon DynamoDB	예	예	아니요	아니요	예	예
Amazon ElastiCache	예	아니요 ¹	아니요	아니요	예	예
Amazon Redshift	예	예	아니요	예	예	예
Amazon Relational Database Service (Amazon RDS)	예	예	아니요	예	예	예
Amazon SimpleDB	예	예	아니요	아니요	예	아니요

¹ 클러스터/복제 그룹을 시드할 때 두 API 작업이 Amazon S3 ARN 리소스 한 개를 지정합니다.

개발자 도구 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
AWS Cloud9	예	예	예	예	예	예
CodeBuild	예	예	아니요	아니요	예	아니요
CodeCommit	예	예	아니요	아니요	예	아니요
AWS CodeDeploy	예	예	아니요	아니요	예	아니요
CodePipeline	예	예	아니요	아니요	예	아니요
AWS CodeStar	예	예 ¹	아니요	아니요	예	아니요
AWS X-Ray	예	아니요	아니요	아니요	예	아니요

보안, 자격 증명 및 규정 준수 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
AWS Artifact	예	예	아니요	아니요	예	아니요
AWS Certificate Manager (ACM)	예	예	아니요	아니요	예	아니요
AWS CloudHSM	예	아니요	아니요	아니요	예	예
AWS CloudHSM 클래식	예	아니요	아니요	아니요	아니요	아니요
Amazon Cognito	예	예	아니요	아니요	예	아니요
AWS Directory Service	예	아니요	아니요	아니요	예	아니요
Amazon GuardDuty	예	예	아니요	아니요	예	예
AWS Identity and Access Management (IAM)	예	예	예 ¹	아니요	예 ²	아니요
Amazon Inspector	예	아니요	아니요	아니요	예	예
AWS Key Management Service (AWS KMS)	예	예	예	아니요	예	예
Amazon Macie	예	아니요	아니요	아니요	예	예
AWS Organizations	예	예	아니요	아니요	예	예
AWS Secrets Manager	예	예	예	예	예	아니요
AWS Security Hub	예	예	아니요	아니요	예	예

AWS Single Sign-On (AWS SSO)	예	아니요	아니요	아니요	예	예
AWS Security Token Service (AWS STS)	예	예 ³	아니요	아니요	예 ⁴	아니요
AWS Shield Advanced	예	아니요	아니요	아니요	예	아니요
AWS WAF	예	예	아니요	아니요	예	예

¹ IAM은 역할 신뢰 정책이라고 하는 리소스 기반 정책 유형 하나만 지원하며, 이 유형은 IAM 역할에 연결됩니다. 자세한 정보는 [사용자에 대한 역할 전환 권한 부여 \(p. 229\)](#) 단원을 참조하십시오.

² IAM에 대한 일부 API 작업만 임시 자격 증명으로 호출할 수 있습니다. 자세한 정보는 [API 옵션 비교](#) 단원을 참조하십시오.

³ AWS STS는 '리소스'가 없지만 사용자에게 유사한 방식으로 액세스를 제한하는 것을 허용합니다. 자세한 정보는 [이름을 사용한 임시 보안 자격 증명 액세스 거부](#) 단원을 참조하십시오.

⁴ AWS STS에 대한 일부 API만 임시 자격 증명을 이용한 호출을 지원합니다. 자세한 정보는 [API 옵션 비교](#) 단원을 참조하십시오.

Machine Learning 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
Amazon Comprehend	예	아니요	아니요	아니요	예	아니요
AWS DeepRacer	예	아니요	아니요	아니요	예	예
Amazon Lex	예	예	아니요	아니요	예	예
Amazon Machine Learning	예	예	아니요	예	예	아니요
Amazon Polly	예	예	아니요	아니요	예	아니요
Amazon Rekognition	예	예	아니요	아니요	아니요	아니요
Amazon SageMaker	예	예	아니요	예 ¹	예	아니요
Amazon Transcribe	예	아니요	아니요	아니요	예	아니요
Amazon Translate	예	아니요	아니요	아니요	예	아니요

¹ Amazon SageMaker에서는 `InvokeEndpoint` 호출에는 태그 기반 권한 부여 기능을 지원하지 않습니다.

관리 및 거버넌스 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
AWS CloudFormation	예	예	아니요	아니요	예	아니요

AWS CloudTrail	예	예	아니요	아니요	예	아니요
Amazon CloudWatch	예	아니요	아니요	아니요	예	예 ¹
Amazon CloudWatch Events	예	예	아니요	아니요	예	아니요
Amazon CloudWatch Logs	예	예	아니요	아니요	예	아니요
AWS Config	예	예 ²	아니요	아니요	예	예
AWS Health	예	아니요	아니요	아니요	예	아니요
AWS OpsWorks	예	예	아니요	아니요	예	아니요
AWS OpsWorks for Chef Automate	예	예	아니요	아니요	예	아니요
AWS Service Catalog	예	아니요	아니요	아니요	예	아니요
AWS 시스템 관리자	예	예	아니요	예	예	예
AWS Trusted Advisor	예 ³	예	아니요	아니요	예 ⁴	예

¹ Amazon CloudWatch 서비스 연결 역할은 AWS Management 콘솔을 사용하여 생성할 수 없으며 [경보 작업](#) 기능만 지원합니다.

² AWS Config은(는) 다중 계정 다중 리전 데이터 집계 및 AWS Config 역할에 대한 리소스 수준 권한을 지원합니다. 지원되는 리소스 목록을 보려면 [AWS Config API 가이드](#)의 다중 계정 다중 리전 데이터 섹션 및 AWS Config 역할 섹션을 참조하십시오.

³ Trusted Advisor에 대한 API 액세스는 AWS Support API를 통해 이루어지며 AWS Support IAM 정책에 의해 제어됩니다.

⁴ Trusted Advisor는 키/값 페어 및 ssm:Overwrite에 대해 ssm:resourceTag 태그 지정 조건을 지원합니다.

マイグレーション 및 전송 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
AWS Application Discovery Service	예	아니요	아니요	아니요	아니요	예
AWS Database Migration Service	예	예	아니요	예	예	아니요
AWS Migration Hub	예	예	아니요	아니요	예	아니요

모바일 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
AWS 증폭	예	예	아니요	아니요	아니요	아니요

AWS Device Farm	예	아니요	아니요	아니요	예	아니요
Amazon Pinpoint	예	예	아니요	아니요	예	아니요

네트워킹 및 콘텐츠 전송 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
Amazon API Gateway	예	예	예	아니요	예	아니요
AWS App Mesh	예	아니요	아니요	아니요	예	아니요
Amazon CloudFront	예 ¹	아니요	아니요	아니요	예	아니요
AWS Cloud Map	예	아니요	아니요	아니요	예	아니요
AWS Direct Connect	예	아니요	아니요	아니요	예	아니요
AWS Global Accelerator	예	예	아니요	아니요	예	아니요
Amazon Route 53	예	예	아니요	아니요	예	아니요
Amazon Virtual Private Cloud (Amazon VPC)	예	예 ²	예 ³	예	예	아니요

¹ CloudFront는 CloudFront 키 페어 생성을 위한 작업 수준 권한을 지원하지 않습니다. AWS 계정 루트 사용자(를) 사용하여 CloudFront 키 페어를 생성해야 합니다. 자세한 정보는 Amazon CloudFront 개발자 안내서의 [신뢰할 수 있는 서명자에 대해 CloudFront 키 페어 생성](#) 단원을 참조하십시오.

² IAM 사용자 정책에서는 특정 Amazon VPC 엔드포인트에 대해 권한을 제한할 수 없습니다. Action 또는 ec2:*VpcEndpoint* API 작업을 포함하는 모든 ec2:DescribePrefixLists 요소는 ""Resource": "*""를 포함해야 합니다. 자세한 정보는 Amazon VPC 사용 설명서의 [엔드포인트 사용 제어](#) 단원을 참조하십시오.

³ Amazon VPC에서는 단일 리소스 정책을 VPC 엔드포인트에 연결하여 해당 엔드포인트를 통해 액세스 가능한 대상을 제한할 수 있도록 지원합니다. 특정 Amazon VPC 엔드포인트의 리소스에 대한 액세스를 제어하기 위해 리소스 기반 정책을 사용하는 방법에 대한 자세한 정보는 Amazon VPC 사용 설명서의 [엔드포인트 정책 사용](#) 단원을 참조하십시오.

미디어 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
Amazon Elastic Transcoder	예	예	아니요	아니요	예	아니요
AWS Elemental MediaConnect	예	예	아니요	아니요	예	아니요
AWS Elemental MediaConvert	예	예	아니요	아니요	예	아니요
AWS Elemental MediaStore	예	예	예	아니요	예	아니요

AWS Elemental MediaTailor	예	예	아니요	아니요	예	아니요
Kinesis 비디오 스트림	예	예	아니요	아니요	예	아니요

데스크톱 및 앱 스트리밍 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
Amazon AppStream	예	아니요	아니요	아니요	예	아니요
Amazon AppStream 2.0	예	아니요	아니요	아니요	예	아니요
Amazon WorkSpaces	예	예	아니요	아니요	예	아니요
Amazon WAM	예	아니요	아니요	아니요	예	아니요

분석 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
Amazon Athena	예	아니요	아니요	아니요	예	아니요
Amazon CloudSearch	예	예	아니요	아니요	예	아니요
AWS Data Pipeline	예	아니요	아니요	예	예	아니요
Amazon Elasticsearch Service	예	예	예	아니요	예	예
Amazon EMR	예	아니요	아니요	예	예	예
AWS Glue	예	예	예	아니요	예	아니요
Amazon Kinesis Data Analytics	예	예	아니요	아니요	예	아니요
Amazon Kinesis Data Firehose	예	예	아니요	아니요	예	아니요
Amazon Kinesis Data Streams	예	예	아니요	아니요	예	아니요
Amazon QuickSight	예	아니요	아니요	아니요	아니요	아니요

애플리케이션 통합 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
Amazon MQ	예	아니요	아니요	아니요	예	아니요

Amazon Simple Email Service (Amazon SES)	예	예 ¹	아니요	아니요	예 ²	아니요
Amazon Simple Notification Service (Amazon SNS)	예	예	예	아니요	예	아니요
Amazon Simple Queue Service (Amazon SQS)	예	예	예	아니요	예	아니요
Amazon Simple Workflow Service (Amazon SWF)	예	예	아니요	예	예	아니요

¹ Amazon SES는 특정 SES ID에 액세스하도록 발신자에게 위임할 수 있는 권한을 부여하는 정책의 리소스 수준 권한을 지원합니다.

² Amazon SES API만이 임시 보안 자격 증명을 지원합니다. Amazon SES SMTP 인터페이스는 임시 보안 자격 증명에서 파생된 SMTP 자격 증명을 지원하지 않습니다.

비즈니스 애플리케이션 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
Alexa for Business	예	예	아니요	예	예	아니요
Amazon WorkDocs	예	아니요	아니요	아니요	예	아니요
Amazon WorkMail	예	아니요	아니요	아니요	예	아니요

사물인터넷 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
AWS IoT Greengrass	예	예	아니요	아니요	예	아니요
AWS IoT	예	예 ¹	예 ¹	예	예	아니요
AWS IoT Things Graph	예	아니요	아니요	아니요	예	아니요

¹ AWS IoT에 연결된 디바이스는 X.509 인증서 또는 Amazon Cognito 자격 증명을 통해 인증됩니다. AWS IoT 정책을 X.509 인증서 또는 Amazon Cognito 자격 증명에 연결하여 디바이스가 어떤 작업을 수행하도록 권한 부여할 것인지 제어할 수 있습니다. 자세한 정보는 AWS IoT 개발자 안내서의 [AWS IoT의 보안 및 자격 증명 단원](#)을 참조하십시오.

로봇 공학 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할

RoboMaker	예	예	아니요	아니요	아니요	예
-----------	---	---	-----	-----	-----	---

게임 개발 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
Amazon GameLift	예	아니요	아니요	아니요	예	아니요

AR 및 VR 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
Amazon Sumerian	예	예	아니요	아니요	예	아니요

고객 참여 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
Amazon Connect	예	예	아니요	아니요	예	예

최종 사용자 컴퓨팅 서비스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
Amazon WorkLink	예	예	예	아니요	예	아니요

추가 리소스

서비스	작업	리소스 수준 권한	리소스 기반 정책	태그 기반 권한 부여	2013년 5월 22일	서비스 연결 역할
AWS Billing and Cost Management	예	아니요	아니요	아니요	예	아니요
AWS Marketplace	예	예	아니요	아니요	예	아니요
AWS Support	아니요	아니요	아니요	아니요	예	예

IAM JSON 정책 참조

이 단원에서는 IAM에서 JSON 정책의 자세한 구문과 설명, 요소의 예, 변수, 평가 로직을 설명합니다. 더 일반적인 내용은 [JSON 정책 개요 \(p. 309\)](#) 단원을 참조하십시오.

본 참조는 다음 섹션을 포함합니다:

- [IAM JSON 정책 요소 참조 \(p. 498\)](#) — 정책을 생성할 때 사용할 수 있는 요소에 대해 자세히 알아봅니다. 더 많은 정책 예제를 보면서 조건, 지원되는 데이터 유형, 다양한 서비스에서 사용되는 방법을 살펴봅니다.
- [정책 평가 로직 \(p. 531\)](#) — 이 단원에서는 AWS 요청, 그 요청이 인증되는 방식 및 AWS가 정책을 사용하여 리소스에 대한 액세스를 결정하는 방식을 기술합니다.
- [IAM JSON 정책 언어의 문법 \(p. 538\)](#) — 이 단원에서는 IAM에서 정책 생성 시 사용되는 언어의 정규 문법에 대해 살펴보겠습니다.
- [직무 기능에 대한 AWS 관리형 정책 \(p. 543\)](#) — 이 단원에서는 IT 업계의 일반적인 직무와 직접 매핑되는 모든 AWS 관리형 정책이 나열됩니다. 이러한 정책을 사용하여 특정 직무 담당자에게 기대되는 작업 수행에 필요한 권한을 부여할 수 있습니다. 이러한 정책은 다수의 서비스에 대한 권한을 단일 정책으로 통합합니다.
- [AWS 전역 조건 컨텍스트 키 \(p. 551\)](#) — 이 단원에는 IAM 정책에서 권한을 제한하는데 사용할 수 있는 모든 AWS 전역 조건 키 목록이 포함되어 있습니다.
- [IAM 및 AWS STS 조건 컨텍스트 키 \(p. 558\)](#) — 이 단원에는 IAM 정책에서 권한을 제한하는데 사용할 수 있는 모든 IAM 및 AWS STS 조건 키 목록이 포함되어 있습니다.
- [???](#) — 이 단원에는 IAM 정책에서 권한으로 사용할 수 있는 모든 AWS API 작업 목록이 나와 있습니다. 또한 요청을 추가로 미세 조정하는데 사용할 수 있는 서비스별 조건 키도 포함되어 있습니다.

IAM JSON 정책 요소 참조

JSON 정책 문서는 여러 요소로 구성됩니다. 여기에 나열되는 요소들은 정책에서 사용되는 일반적인 순서를 따릅니다. 요소 순서는 중요하지 않습니다.—예를 들어 `Resource` 요소는 `Action` 요소 앞에 올 수 있습니다. 또한 정책에서 `Condition` 요소는 지정하지 않아도 됩니다. JSON 정책 문서의 일반적인 구조와 목적에 대해 자세히 알아보려면 [JSON 정책 개요 \(p. 309\)](#)를 참조하십시오.

일부 JSON 정책 요소는 함께 사용할 수 없습니다. 즉, 둘 다 사용하는 정책을 생성할 수 없습니다. 예를 들어 동일한 정책 문서에서 `Action`과 `NotAction` 둘 다 사용할 수 없습니다. 함께 사용할 수 없는 다른 쌍에는 `Principal/NotPrincipal` 및 `Resource/NotResource`가 있습니다.

정책 세부 정보는 서비스에서 유효한 작업이나 추가되는 리소스 유형 등에 따라 각 서비스마다 차이가 있습니다. 따라서 특정 서비스에 맞는 정책을 작성할 때는 해당 서비스의 정책 예제를 살펴보는 것이 좋습니다. IAM을 지원하는 모든 서비스 목록을 비롯해 각 서비스의 IAM 및 정책 설명서 링크는 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하십시오.

주제

- [IAM JSON 정책 요소: Version \(p. 499\)](#)
- [IAM JSON 정책 요소: Id \(p. 499\)](#)
- [IAM JSON 정책 요소: Statement \(p. 499\)](#)
- [IAM JSON 정책 요소: Sid \(p. 500\)](#)
- [IAM JSON 정책 요소: Effect \(p. 500\)](#)
- [AWS JSON 정책 요소: Principal \(p. 501\)](#)
- [AWS JSON 정책 요소: NotPrincipal \(p. 504\)](#)
- [IAM JSON 정책 요소: Action \(p. 506\)](#)
- [IAM JSON 정책 요소: NotAction \(p. 507\)](#)
- [IAM JSON 정책 요소: Resource \(p. 508\)](#)
- [IAM JSON 정책 요소: NotResource \(p. 509\)](#)

- IAM JSON 정책 요소: Condition (p. 510)
- IAM 정책 요소: 변수 및 태그 (p. 524)
- IAM JSON 정책 요소: 지원되는 데이터 형식 (p. 531)

IAM JSON 정책 요소: Version

동음이의어 참고

Version JSON 정책 요소는 정책 버전과 다릅니다. Version 정책 요소는 정책 내에서 사용되며 정책 언어의 버전을 정의합니다. 반면에 정책 버전은 IAM에서 고객 관리형 정책을 변경할 때 생성됩니다. 변경된 정책은 기존 정책을 덮어쓰지 않습니다. 대신 IAM에서 관리형 정책의 새 버전을 만듭니다. 관리형 정책에서 사용할 수 있는 여러 버전 지원에 대한 정보를 찾고 있다면 [the section called "IAM 정책 버전 관리" \(p. 399\)](#) 단원을 참조하십시오.

Version 정책 요소는 정책의 처리에 사용할 언어 구문 규칙을 지정합니다. 사용 가능한 모든 정책 기능을 사용하려면 모든 정책에서 Statement 요소 앞에 다음의 version 요소를 포함시킵니다.

```
"Version": "2012-10-17"
```

IAM은 다음과 같은 Version 요소 값을 지원합니다.

- 2012-10-17. 이 값은 정책 언어의 현재 버전이며, 항상 Version 요소를 포함하여 2012-10-17로 설정해야 합니다. 그렇지 않으면 이 버전에 채택되지 않은 [정책 변수 \(p. 524\)](#) 등의 기능을 사용할 수 없습니다.
- 2008-10-17. 이 값은 이전 정책 언어 버전입니다. 따라서 오래된 기존 정책에서는 이 버전이 표시될 수도 있습니다. 새로운 정책에서는 또는 기존 정책을 업데이트하는 경우에는 이 버전을 사용하지 마십시오.

Version 요소를 포함하지 않을 경우의 기본값은 2008-10-17이지만 정책 기능 등 최신 기능을 정책에서 사용할 수 없습니다. 예를 들어 \${aws:username} 같은 변수가 정책에서 변수로 인식되지 않고 리터럴 문자열로 취급됩니다.

IAM JSON 정책 요소: Id

Id 요소는 정책 식별자(옵션)를 지정합니다. 다른 서비스의 ID 사용 방법과는 다릅니다.

ID 요소를 설정할 수 있는 서비스의 경우에는 UUID(GUID)를 값으로 사용하거나, UUID를 ID 일부로 사용하여 고유성을 확보하는 것이 좋습니다.

```
"Id": "cd3ad3d9-2776-4ef1-a904-4c229d1642ee"
```

Note

AWS 서비스(예: Amazon SQS 또는 Amazon SNS 등) 중에는 이 요소가 필요하거나 고유성 요건을 따로 요구하는 경우도 있습니다. 정책 작성에 대한 서비스별 정보는 이용하려는 서비스의 설명서를 참조하십시오.

IAM JSON 정책 요소: Statement

Statement 요소는 정책의 주요 요소로서 필수입니다. 또한 다수 추가할 수도 있습니다(이 페이지의 후속 섹션 참조). Statement 요소는 개별 요소가 모여 하나의 배열을 이룹니다. 개별 문은 중괄호 {}로 묶인 JSON 블록입니다.

```
"Statement": [{"...},{...},{...}]
```

다음은 단일 Statement 요소에서 3개의 문이 하나의 배열을 이루는 정책을 나타낸 예제입니다 (이 정책에서는 Amazon S3 콘솔에서 자신의 'home 폴더'에 액세스할 수 있습니다). 정책에는 aws:username 변

수가 추가되었습니다. 이 변수는 정책 평가 중 요청이 있으면 사용자 이름으로 바뀝니다. 자세한 내용은 [스개 \(p. 524\)](#)를 참조하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3>ListAllMyBuckets",  
                "s3:GetBucketLocation"  
            ],  
            "Resource": "arn:aws:s3:::*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "s3>ListBucket",  
            "Resource": "arn:aws:s3:::BUCKET-NAME",  
            "Condition": {"StringLike": {"s3:prefix": [  
                "",  
                "home/",  
                "home/${aws:username}/"  
            ]}}  
        },  
        {  
            "Effect": "Allow",  
            "Action": "s3:*",  
            "Resource": [  
                "arn:aws:s3:::BUCKET-NAME/home/${aws:username}",  
                "arn:aws:s3:::BUCKET-NAME/home/${aws:username}/*"  
            ]  
        }  
    ]  
}
```

IAM JSON 정책 요소: Sid

Sid(문 ID)는 정책 문에 입력되는 식별자(옵션)입니다. Sid 같은 문 배열에서 각 문에 할당할 수 있습니다. SQS나 SNS처럼 ID 요소를 지정할 수 있는 서비스에서는 Sid 값이 정책 문서 ID의 하위 ID나 마찬가지입니다. IAM에서 Sid 같은 JSON 정책 내 고유성이 보장되어야 합니다.

```
"Sid": "1"
```

IAM에서 Sid는 IAM API에 노출되지 않습니다. 따라서 이 ID를 근거로 특정 문을 가져올 수는 없습니다.

Note

AWS 서비스(예: Amazon SQS 또는 Amazon SNS 등) 중에는 이 요소가 필요하거나 고유성 요건을 따로 요구하는 경우도 있습니다. 정책 작성에 대한 서비스별 정보는 이용하려는 서비스의 설명서를 참조하십시오.

IAM JSON 정책 요소: Effect

Effect 요소는 필수로서, 문의 허용(allow) 또는 명시적 거부(explicit deny) 중 하나를 지정합니다. Effect 유효값은 Allow 및 Deny입니다.

```
"Effect": "Allow"
```

기본적으로 리소스 액세스는 거부됩니다. 리소스 액세스를 허용하려면 Effect 요소를 Allow로 설정해야 합니다. 허용을 재정의하려면(예: 그 밖에 다른 방법으로 실행 중인 허용을 재정의하려면) Effect 요소를 Deny로 설정합니다. 자세한 내용은 [정책 평가 로직 \(p. 531\)](#) 단원을 참조하십시오.

AWS JSON 정책 요소: Principal

Principal 요소를 사용하여 IAM 사용자, 연합된 사용자, IAM 역할, AWS 계정, AWS 서비스 또는 그 밖에 리소스에 대한 액세스가 허용되거나 거부된 보안 주체 엔터티를 지정할 수 있습니다. IAM 자격 증명 기반 정책에서는 Principal 요소를 사용할 수 없습니다. IAM 역할을 위한 신뢰 정책 및 리소스 기반 정책에서는 사용할 수 있습니다. 리소스 기반 정책은 IAM 리소스에 직접 삽입할 수 있는 정책입니다. 예를 들어, Amazon S3 버킷 또는 AWS KMS 고객 마스터 키(CMK)에 정책을 삽입할 수 있습니다.

Principal 요소의 사용 방법은 아래와 같습니다.

- IAM 역할의 신뢰 정책에서 Principal 요소를 사용하여 역할을 위임할 사용자를 지정합니다. 교차 계정 액세스인 경우에는 신뢰할 수 있는 계정의 12자리 식별자를 지정합니다.

Note

역할을 생성한 이후 계정을 "*"로 변경하여 모두가 이 역할을 수임하도록 할 수 있습니다. 이렇게 하는 경우 다른 방법(예: 특정 IP 주소로만 액세스를 제한하는 Condition 요소)을 통해 역할에 액세스할 수 있는 사용자를 제한하는 것이 좋습니다. 역할을 모두 액세스할 수 있는 상태로 두지 마십시오.

- 리소스 기반 정책에서는 Principal 요소를 사용해 리소스 액세스가 허용된 계정 또는 사용자를 지정합니다.

IAM 사용자 및 그룹에 연결한 정책에서는 Principal 요소를 사용하지 마십시오. 마찬가지로 IAM 역할의 권한 정책에서도 보안 주체를 지정해서는 안 됩니다. 이 경우 보안 주체는 복시적으로 정책이 연결되어 있는 사용자(IAM 사용자) 또는 역할을 위임하는 사용자(역할 액세스 정책)가 됩니다. 그리고, 정책이 IAM 그룹에 연결되면 해당 그룹 내에서 요청하는 IAM 사용자가 보안 주체가 됩니다.

보안 주체 지정

보안 주체는 AWS 계정, IAM 사용자, IAM 역할, 연동 사용자 또는 위임된 역할 사용자의 [Amazon 리소스 이름\(ARN\)](#) ([p. 480](#))을 사용해 지정합니다. IAM 그룹 및 인스턴스 프로파일을 보안 주체로 지정할 수 없습니다.

다음은 보안 주체를 지정할 수 있는 다양한 방법을 나타낸 예제들입니다.

특정 AWS 계정

AWS 계정 자격 증명을 정책에서 보안 주체로 사용할 경우, 해당 계정에 속하는 모든 자격 증명에 정책 문의 권한을 부여할 수 있습니다. 여기에는 해당 계정의 IAM 사용자 및 역할이 포함됩니다. AWS 계정을 지정하는 경우 계정 ARN(`arn:aws:iam::AWS-account-ID:root`)을 사용하거나 접두사 `AWS:`와 그 뒤에 계정 ID가 따라오는 약식 형태를 사용할 수 있습니다.

예를 들어, 계정 ID 123456789012의 경우 다음 방법 중 하나를 사용하여 Principal 요소에서 해당 계정을 지정할 수 있습니다.

```
"Principal": { "AWS": "arn:aws:iam::123456789012:root" }
```

```
"Principal": { "AWS": "123456789012" }
```

또한 앞서 언급한 방법을 자유롭게 조합하여 배열을 사용해 두 개 이상의 AWS 계정을 보안 주체로 지정할 수 있습니다.

```
"Principal": {
    "AWS": [
        "arn:aws:iam::123456789012:root",
        "999999999999"
    ]
}
```

개별 IAM 사용자

다음 예제와 같이 개별 IAM 사용자를 보안 주체로 지정할 수 있습니다.

Note

Principal 요소에서 사용자 이름은 대/소문자를 구분합니다.

```
"Principal": { "AWS": "arn:aws:iam::AWS-account-ID:user/user-name" }
```

```
"Principal": {  
    "AWS": [  
        "arn:aws:iam::AWS-account-ID:user/user-name-1",  
        "arn:aws:iam::AWS-account-ID:user/UserName2"  
    ]  
}
```

Principal 요소로 사용자를 지정할 때는 "모든 사용자"의 의미로 와일드카드(*)를 사용할 수 없습니다. 보안 주체는 항상 특정 사용자가 되어야 하기 때문입니다.

Important

역할 신뢰 정책의 Principal 요소에 특정 IAM 사용자를 가리키는 ARN이 포함되어 있으면, 정책을 저장할 때 해당 ARN이 해당 사용자의 고유 보안 주체 ID로 변환됩니다. 그러면 누군가가 해당 사용자를 제거하고 다시 만들어 본인의 권한을 에스컬레이션할 위험을 완화할 수 있습니다. 일반적으로 콘솔에서는 이 ID가 보이지 않습니다. 신뢰 정책이 표시될 때 해당 사용자의 ARN으로 다시 역변환되기 때문입니다. 그러나 해당 사용자를 삭제하면 관계가 깨집니다. 사용자를 다시 생성해도 정책은 더 이상 적용되지 않습니다. 새 사용자의 새 보안 주체 ID가 신뢰 정책에 저장된 ID와 일치하지 않기 때문입니다. 이 경우 보안 주체 ID가 콘솔에 표시됩니다. AWS에서 더 이상 이를 유효한 ARN에 다시 매핑할 수 없기 때문입니다. 결과적으로 신뢰 정책의 Principal 요소에서 참조된 사용자를 삭제하고 다시 만드는 경우, 역할을 편집하여 현재의 잘못된 보안 주체 ID를 올바른 ARN으로 바꿔야 합니다. 정책을 저장하면 ARN이 다시 해당 사용자의 새로운 보안 주체 ID로 변환됩니다.

연동 사용자(웹 자격 증명 연동 사용)

```
"Principal": { "Federated": "cognito-identity.amazonaws.com" }
```

```
"Principal": { "Federated": "www.amazon.com" }
```

```
"Principal": { "Federated": "graph.facebook.com" }
```

```
"Principal": { "Federated": "accounts.google.com" }
```

연동 사용자(SAML 자격 증명 공급자 사용)

```
"Principal": { "Federated": "arn:aws:iam::AWS-account-ID:saml-provider/provider-name" }
```

IAM 역할

```
"Principal": { "AWS": "arn:aws:iam::AWS-account-ID:role/role-name" }
```

Important

역할 신뢰 정책의 Principal 요소에 특정 IAM 역할을 가리키는 ARN이 포함되어 있으면, 정책을 저장할 때 해당 ARN이 해당 역할의 고유 보안 주체 ID로 변환됩니다. 그러면 누군가가 해당 역할을 제거하고 다시 만들어 본인의 권한을 에스컬레이션할 위험을 완화할 수 있습니다. 일반적으로 콘솔

에서는 이 ID가 보이지 않습니다. 신뢰 정책이 표시될 때 해당 역할의 ARN으로 다시 역변환되기 때문입니다. 그러나 해당 역할을 삭제하면 관계가 깨집니다. 역할을 다시 만들더라도 해당 정책이 더 이상 적용되지 않습니다. 새 역할의 새 보안 주체 ID가 신뢰 정책에 저장된 ID와 일치하지 않기 때문입니다. 이 경우 보안 주체 ID가 콘솔에 표시됩니다. AWS에서 더 이상 이를 유효한 ARN에 다시 맵핑할 수 없기 때문입니다. 결과적으로 신뢰 정책의 Principal 요소에서 참조된 역할을 삭제하고 다시 만드는 경우, 현재 잘못된 보안 주체 ID를 올바른 ARN으로 바꾸도록 역할을 편집해야 합니다. 정책을 저장하면 ARN이 다시 해당 역할의 새로운 보안 주체 ID로 변환됩니다.

특정 위임된 역할 사용자

```
"Principal": { "AWS": "arn:aws:sts::AWS-account-ID:assumed-role/role-name/role-session-name" }
```

AWS 서비스

AWS 서비스에서 위임할 수 있는 IAM 역할을 [서비스 역할 \(p. 154\)](#)이라고 합니다. 서비스 역할에는 신뢰 정책이 포함되어 있어야 합니다. 신뢰 정책은 역할을 위임할 수 있는 보안 주체를 정의하는 역할에 연결된 리소스 기반 정책입니다. 일부 서비스 역할에는 신뢰 정책이 미리 정의되어 있습니다. 그러나 신뢰 정책에 서비스 보안 주체를 지정해야 하는 경우도 있습니다. 서비스 보안 주체는 서비스에 권한을 부여하는 데 사용되는 식별자입니다. 식별자에는 긴 버전의 서비스 이름이 포함되어 있고 보통은 다음과 같은 형식을 갖습니다.

`long-service-name.amazonaws.com`

서비스 보안 주체는 서비스가 정의합니다. 서비스의 보안 주체를 확인하려면 해당 서비스의 설명서를 참조하십시오. 일부 서비스에 대해서는 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#)을 참조하여 서비스 연결 역할 열에 예라고 표시된 서비스를 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 [Yes] 링크를 선택합니다. 서비스의 보안 주체를 보려면 Service-Linked Role Permissions(서비스 연결 역할 권한) 단원을 참조하십시오.

다음 예제는 서비스 역할에 연결할 수 있는 정책을 보여줍니다. 이 정책은 Amazon EMR 서비스와 AWS Data Pipeline 서비스가 역할을 수행할 수 있도록 합니다. 그러면 서비스가 해당 역할에 할당된 권한 정책에서 부여한 모든 작업을 수행할 수 있습니다(표시되지 않음). 여러 서비스 보안 주체를 지정할 때 Service 요소를 두 개 지정하면 안 됩니다. 하나만 지정할 수 있습니다. 대신 여러 서비스 보안 주체의 배열을 하나의 Service 요소의 값으로 사용합니다.

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

일부 AWS 서비스는 보안 주체를 지정하는 데 몇 가지 옵션을 추가로 지원합니다. 예를 들어 Amazon S3에서 다음 형식을 사용하여 [정식 사용자 ID](#)를 지정할 수 있습니다.

```
"Principal": { "CanonicalUser":  
    "79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be" }
```

모든 사용자(익명 사용자)

다음은 동일합니다.

```
"Principal": "*"
```

```
"Principal" : { "AWS" : "*" }
```

Note

이 예제에서 Everyone/Anonymous에 대한 자리 표시자로 별표를 사용할 수 있습니다. 이름이나 ARN의 일부를 나타내기 위한 와일드카드로 사용할 수는 없습니다. 또한 정책에서 Principal 요소를 통해 액세스를 달리 제한하지 않을 경우 역할의 신뢰 정책에서 condition 요소에 와일드카드를 사용하지 않는 것이 좋습니다. 그렇지 않으면 계정의 모든 IAM 사용자가 역할에 액세스할 수 있습니다.

추가 정보

자세한 내용은 다음을 참조하십시오.

- Amazon Simple Storage Service 개발자 가이드의 [버킷 정책 예제](#)
- Amazon Simple Notification Service 개발자 안내서의 [Amazon SNS 정책의 예](#)
- Amazon Simple Queue Service 개발자 안내서의 [Amazon SQS 정책 예제](#)
- AWS Key Management Service Developer Guide의 키 정책
- AWS General Reference의 [계정 식별자](#)
- 웹 자격 증명 연동에 대하여 (p. 162)

AWS JSON 정책 요소: NotPrincipal

NotPrincipal 요소를 사용하여 IAM 사용자, 연합된 사용자, IAM 역할, AWS 계정, AWS 서비스 또는 그 밖에 리소스에 대한 액세스가 허용되거나 거부되지 않은 보안 주체 엔터티를 지정할 수 있습니다. NotPrincipal 요소를 사용하면 보안 주체 목록에 예외를 지정할 수 있습니다. 이 요소를 사용하여 NotPrincipal 요소로 지정된 보안 주체를 제외하고 모든 보안 주체에 대한 액세스를 거부할 수 있습니다. NotPrincipal 지정 구문은 [AWS JSON 정책 요소: Principal \(p. 501\)](#) 지정할 때와 동일합니다.

IAM 자격 증명 기반 정책에서는 NotPrincipal 요소를 사용할 수 없습니다. IAM 역할을 위한 신뢰 정책 및 리소스 기반 정책에서는 사용할 수 있습니다. 리소스 기반 정책은 IAM 리소스에 직접 삽입할 수 있는 정책입니다.

Important

NotPrincipal을 사용해야 하는 시나리오는 극히 드뭅니다. 따라서 NotPrincipal 사용을 결정하기 전에 다른 권한 부여 옵션을 살펴보는 것이 바람직합니다.

NotPrincipal 다음으로 바꿉니다.Allow

따라서 "Effect": "Allow"와 동일한 정책 문에는 NotPrincipal을 사용하지 않는 것이 좋습니다. 사용하면 NotPrincipal 요소에 지정된 보안 주체를 제외하고 모든 보안 주체가 허용됩니다. 그러면 지정된 보안 주체를 제외한 모든 보안 주체에 정책 문에 지정된 권한이 부여되기 때문에 이렇게 하지 않는 것이 좋습니다. 익명(비인증) 사용자에게 액세스를 부여하게 될 수도 있기 때문입니다.

NotPrincipal 다음으로 바꿉니다.Deny

"Effect": "Deny"와 동일한 정책 문에 NotPrincipal을 사용할 경우, 정책 문에 지정된 작업은 지정된 보안 주체를 제외한 모든 보안 주체에 대해 명시적으로 거부됩니다. 이 방법을 사용하여 화이트리스트를 구현할 수 있습니다. NotPrincipal과 Deny를 함께 사용할 경우, 거부되지 않은 보안 주체의 계정 ARN도 지정해야 합니다. 지정하지 않으면 정책에서 해당 보안 주체를 포함하는 전체 계정에 대한 액세스가 거부될 수 있습니다. 정책에 포함하는 서비스에 따라 AWS에서 먼저 계정을 검증한 후 사용자를 검증할 수 있습니다. 위임된 역할 사용자(역할을 사용하는 사람)를 평가할 때 AWS는 먼저 계정을 검증한 후 위임된 역할 사용자

를 평가합니다. 위임된 역할 사용자는 그 역할을 위임 받을 때 지정된 역할 세션 이름으로 식별할 수 있습니다. 따라서 사용자 계정의 ARN을 명시적으로 포함시키거나, 역할의 ARN과 해당 역할을 포함하는 계정의 ARN을 모두 포함시킬 것을 권장합니다.

Note

최선의 결과를 위해 정책에 계정의 ARN을 포함시켜야 합니다. 모든 경우에 해당하는 것은 아니지만 일부 서비스에서는 계정 ARN이 필요합니다. 기존 정책은 필요한 ARN 없이 계속 적용되지만 이러한 서비스를 포함하는 새 정책은 계정 ARN을 포함시켜야 합니다. IAM은 이러한 서비스를 추적하지 않으므로 계정 ARN을 항상 포함시키는 것이 좋습니다.

다음 예제들은 같은 정책 설명에 있는 `NotPrincipal`과 "Effect": "Deny"를 효과적으로 사용하는 방법을 보여줍니다.

Example 1: 같은 또는 다른 계정의 IAM 사용자

다음 예제에서는 AWS 계정 444455556666에서 Bob이라는 이름의 사용자만 제외하고 모든 보안 주체의 리소스 액세스가 명시적으로 거부되었습니다. 모범 사례로서 `NotPrincipal` 요소는 사용자 Bob과 Bob이 속한 AWS 계정(`arn:aws:iam::444455556666:root`)의 ARN을 모두 포함한다는 점을 유념하십시오. `NotPrincipal` 요소에 Bob의 ARN만 추가될 경우, 정책의 효과에 따라 사용자 Bob을 포함하는 AWS 계정에 대한 액세스가 명시적으로 거부될 수 있습니다. 경우에 따라, 사용자는 자신의 상위 계정보다 많은 권한을 가질 수 없습니다. 따라서 Bob의 계정에 대한 액세스가 명시적으로 거부되면 Bob은 리소스에도 액세스하지 못할 수도 있습니다.

이 예제는 444455556666가 아닌 같은 또는 다른 AWS 계정의 리소스에 연결된 리소스 기반 정책의 정책 설명에 포함될 때 의도한 대로 작동합니다. 이 예제 자체로는 Bob에게 액세스 권한을 부여하지 않지만 명시적으로 거부된 보안 주체 목록에서 Bob만 제외됩니다. Bob에게 리소스 액세스 권한을 허용하려면 다른 정책 문으로 "Effect": "Allow"를 작성함으로써 액세스를 명시적으로 허용해야 합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "NotPrincipal": {"AWS": [  
                "arn:aws:iam::444455556666:user/Bob",  
                "arn:aws:iam::444455556666:root"  
            ]},  
            "Action": "s3:*",  
            "Resource": [  
                "arn:aws:s3::::BUCKETNAME",  
                "arn:aws:s3::::BUCKETNAME/*"  
            ]  
        }]  
    }]
```

Example 2: 같은 또는 다른 계정의 IAM 역할

다음 예제에서는 AWS 계정 444455556666에서 cross-account-audit-app이라는 이름의 위임된 역할 사용자만 제외하고 모든 보안 주체의 리소스 액세스가 명시적으로 거부되었습니다. 모범 사례로서, `NotPrincipal` 요소는 수입된 역할 사용자(cross-account-audit-app), 역할(cross-account-read-only-role), 및 역할이 속한 AWS 계정(444455556666)의 ARN을 포함합니다. `NotPrincipal` 요소에 역할의 ARN이 누락될 경우 정책 효과에 따라 역할에 대한 액세스가 명시적으로 거부될 수 있습니다. 마찬가지로, `NotPrincipal` 요소에 역할이 속한 AWS 계정의 ARN이 누락될 경우에는 정책의 효과에 따라 AWS 계정과 그 계정의 모든 엔터티에 대한 액세스가 명시적으로 거부될 수 있습니다. 경우에 따라, 위임된 역할 사용자는 자신의 상위 역할보다 많은 권한을 가질 수 없고, 역할은 자신의 상위 AWS 계정보다 많은 권한을 가질 수 없습니다. 따라서 역할 또는 계정에 대한 액세스가 명시적으로 거부되면 위임된 역할 사용자는 리소스에 액세스하지 못할 수 있습니다.

이 예제는 444455556666가 아닌 다른 AWS 계정의 리소스에 연결된 정책 문이 리소스 기반 정책의 정책 문에 포함되기 때문에 의도한 대로 효과가 나타납니다. 이 예제 자체에서는 위임된 역할 사용자인 cross-account-audit-app에게 액세스를 허용하지 않지만 명시적으로 거부된 보안 주체 목록에서 cross-account-

audit-app만 제외됩니다. cross-account-audit-app에게 리소스 액세스 권한을 부여하려면 다른 정책 문으로 "Effect": "Allow"를 작성함으로써 액세스를 명시적으로 허용해야 합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "NotPrincipal": {"AWS": [  
                "arn:aws:sts::444455556666:assumed-role/cross-account-read-only-role/cross-  
                account-audit-app",  
                "arn:aws:iam::444455556666:role/cross-account-read-only-role",  
                "arn:aws:iam::444455556666:root"  
            ]},  
            "Action": "s3:*",  
            "Resource": [  
                "arn:aws:s3:::Bucket_AccountAudit",  
                "arn:aws:s3:::Bucket_AccountAudit/*"  
            ]  
        }  
    ]  
}
```

IAM JSON 정책 요소: Action

Action 요소는 특정 작업의 허용 또는 거부 여부를 지정합니다. 문에는 Action 또는 NotAction 요소가 반드시 추가되어야 합니다. AWS 서비스마다 실행할 수 있는 작업을 설명한 목록이 있습니다. 예를 들어 Amazon S3 작업 목록은 Amazon Simple Storage Service 개발자 가이드의 [정책에서 권한 지정](#)에서, Amazon EC2 작업 목록은 [Amazon EC2 API Reference](#)에서, 그리고 AWS Identity and Access Management 작업 목록은 [IAM API Reference](#)에서 확인할 수 있습니다. 그 밖에 다른 서비스의 작업 목록은 해당 서비스의 API 참조 [설명서](#)를 참조하십시오.

값은 서비스(iam, ec2, sqs, sns, s3 등)를 식별할 수 있는 네임스페이스와 허용 또는 거부할 작업 이름을 사용해 지정합니다. 이름은 서비스에서 지원되는 작업과 일치해야 합니다. 접두사와 작업 이름은 대/소문자를 구분하지 않습니다. 예를 들어 iam>ListAccessKeys는 IAM:listaccesskeys와 동일합니다. 다음은 각 서비스의 Action 요소를 나타낸 예제입니다.

Amazon SQS 작업

```
"Action": "sqs:SendMessage"
```

Amazon EC2 작업

```
"Action": "ec2:StartInstances"
```

IAM 작업

```
"Action": "iam:ChangePassword"
```

Amazon S3 작업

```
"Action": "s3:GetObject"
```

Action 요소는 다수의 값을 지정할 수도 있습니다.

```
"Action": [ "sqs:SendMessage", "sqs:ReceiveMessage", "ec2:StartInstances",  
           "iam:ChangePassword", "s3:GetObject" ]
```

특정 AWS 제품이 제공하는 모든 작업에 대해 액세스 권한을 부여하려면 와일드카드(*)를 사용하면 됩니다. 예를 들어, 다음 Action 요소는 모든 S3 작업에 적용됩니다.

```
"Action": "s3:*
```

와일드카드(*)는 작업 이름에도 사용할 수 있습니다. 예를 들어 다음 Action 요소는 CreateAccessKey, DeleteAccessKey, ListAccessKeys, UpdateAccessKey 등 문자열 AccessKey를 포함하는 IAM 작업 모두에게 적용됩니다.

```
"Action": "iam:*AccessKey*
```

일부 서비스에서는 사용 가능한 작업을 제한할 수도 있습니다. 예를 들어 Amazon SQS에서는 가능한 모든 Amazon SQS 작업의 하위 집합만 사용할 수 있습니다. 이 경우 와일드카드(*)는 대기열 전체를 제어하지 못하고, 공유한 작업의 하위 집합만 제어가 가능합니다. 자세한 내용은 Amazon Simple Queue Service 개발자 안내서의 [Understanding Permissions](#) 단원을 참조하십시오.

IAM JSON 정책 요소: NotAction

NotAction은 지정된 작업의 목록을 제외한 모든 작업과 명시적으로 일치하는 고급 정책 요소입니다. NotAction을 사용하면 일치하는 작업의 긴 목록을 포함하는 대신 일치하지 않는 몇몇 작업만 나열함으로써 정책을 줄일 수 있습니다. NotAction을 사용할 때는 이 요소에 지정된 작업들이 제한되는 유일한 작업이라는 점을 유의해야 합니다. 따라서 나열되지 않은 모든 해당 작업 또는 서비스가 Allow 효과를 사용할 경우 허용되고, Deny를 사용하려는 경우에는 나열되지 않은 작업 또는 서비스가 거부됩니다. NotAction을 Resource 요소와 함께 사용할 경우 정책 범위를 제공해야 합니다. 이에 따라 AWS는 어떤 작업이나 서비스를 적용할 수 있는지 결정합니다. 자세한 내용은 다음 예제 정책을 참조하십시오.

NotAction 및 Allow

설명문에서 NotAction 요소를 "Effect": "Allow"와 함께 사용하여 AWS 서비스에서 NotAction에 지정된 작업을 제외한 모든 작업에 대한 액세스 권한을 제공할 수 있습니다. 이 요소와 Resource 요소를 함께 사용하여 정책에 대한 범위를 제공하고 지정한 리소스에서 수행할 수 있는 작업으로만 작업을 제한할 수 있습니다.

다음 예제는 사용자에게 버킷 삭제를 제외하고 S3 리소스에서 수행할 수 있는 모든 Amazon S3 작업에 대한 액세스를 제공합니다. ListAllMyBuckets S3 API 작업은 "*" 리소스가 필요하기 때문에 이 작업은 사용자가 사용할 수 없습니다. 이 정책은 또한 다른 서비스에서의 작업을 허용하지 않습니다. 다른 서비스 작업은 S3 리소스에 적용되지 않기 때문입니다.

```
"Effect": "Allow",
"NotAction": "s3>DeleteBucket",
"Resource": "arn:aws:s3:::*
```

때로는 다수의 작업에 액세스하도록 허용해야 할 수 있습니다. NotAction 요소를 사용하여 효과적으로 설명문을 반전시켜 작업 목록을 단축시킬 수 있습니다. 예를 들어 AWS 서비스는 종류가 다양하므로 사용자에게 IAM 작업에 대한 액세스를 제외한 모든 것을 허용하는 정책을 만들기를 원할 수 있습니다.

다음 예제는 사용자가 IAM을 제외한 모든 AWS 서비스에서 모든 작업에 액세스하도록 허용합니다.

```
"Effect": "Allow",
"NotAction": "iam:*",
"Resource": "*"
```

동일한 설명문에서 또는 동일한 정책의 다른 설명문에서 NotAction 요소와 "Effect": "Allow"를 사용할 경우 주의하십시오. NotAction은 명시적으로 나열되지 않거나 특정 리소스에 적용되지 않는 모든 서비스 및 작업과 일치하므로 사용자에게 의도한 것보다 많은 권한을 부여하는 결과를 가져올 수 있습니다.

NotAction 및 Deny

설명문에서 NotAction 요소를 "Effect": "Deny"와 함께 사용하여 NotAction 요소에 지정된 작업을 제외하고 모든 나열된 리소스에 대한 액세스를 거부할 수 있습니다. 이 조합은 나열된 항목을 허용하는 것이 아니라 나열되지 않은 작업을 명시적으로 거부합니다. 그러므로 허용하려는 작업은 별도로 허용해야 합니다.

다음의 조건부 예제는 사용자가 MFA를 사용하여 로그인하지 않은 경우 비 IAM 작업에 대한 액세스를 거부합니다. 사용자가 MFA를 사용하여 로그인한 경우에는 "Condition" 테스트에 실패하며 최종 "Deny" 문은 효과가 없습니다. 단, 이 정책은 사용자에게 작업에 대한 액세스 권한을 부여하는 것이 아니라 IAM 작업을 제외한 다른 모든 작업을 명시적으로 거부할 뿐입니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Sid": "DenyAllUsersNotUsingMFA",  
        "Effect": "Deny",  
        "NotAction": "iam:*",  
        "Resource": "*",  
        "Condition": {"BoolIfExists": {"aws:MultiFactorAuthPresent": "false"}}  
    }]  
}
```

다음 예제 정책은 나열된 서비스의 작업을 제외하고 eu-central-1 및 eu-west-1 리전 외부의 작업에 대한 액세스를 거부합니다. NotActions 요소에 나열된 서비스는 us-east-1 리전에 실제로 단일 엔드포인트가 있는 AWS 글로벌 서비스의 일부입니다. 그렇지 않다면 이러한 서비스의 작업은 실패할 것입니다. 이 정책은 액세스를 거부하고 다른 정책이 액세스 권한을 부여하도록 요구합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Sid": "DenyAllOutsideEU",  
        "Effect": "Deny",  
        "NotAction": [  
            "aws-portal:*",  
            "iam:*",  
            "organizations:*",  
            "support:*",  
            "sts:*"  
        ],  
        "Resource": "*",  
        "Condition": {"StringNotEquals": {"aws:RequestedRegion": [  
            "eu-central-1",  
            "eu-west-1"  
        ]}}  
    }]  
}
```

IAM JSON 정책 요소: Resource

Resource 요소는 문에서 다루는 객체를 지정합니다. 문에는 Resource 또는 NotResource 요소가 반드시 추가되어야 합니다. 리소스는 ARN을 사용하여 지정할 수 있습니다. ARN 형식에 대한 자세한 내용은 [Amazon 리소스 이름\(ARN\)](#) 및 [AWS 서비스 네임스페이스](#) 단원을 참조하십시오.

각 서비스마다 고유의 리소스가 있습니다. 리소스를 지정하려면 항상 ARN을 사용해야 하지만 리소스의 ARN 세부 정보는 서비스와 리소스에 따라 달라집니다. 리소스 지정 방법에 대한 자세한 내용은 문을 작성하려는 리소스의 서비스 설명서를 참조하십시오.

Note

서비스 중에는 개별적인 리소스로 작업을 지정하지 못하는 서비스도 있습니다. 대신 Action 또는 NotAction 요소로 나열하는 작업이 모두 해당 서비스의 모든 리소스에 적용됩니다. 이 경우에는 * 요소에 와일드카드(Resource)를 사용합니다.

다음은 특정 Amazon SQS 대기열을 나타낸 예제입니다.

```
"Resource": "arn:aws:sqs:us-east-2:account-ID-without-hyphens:queue1"
```

다음은 AWS 계정에서 Bob이라는 이름의 IAM 사용자를 나타내는 예제입니다.

```
"Resource": "arn:aws:iam::account-ID-without-hyphens:user/Bob"
```

와일드카드는 리소스 ARN에도 사용할 수 있습니다. ARN 세그먼트(콜론으로 구분된 부분) 내에서 와일드카드 문자(*) 및 (?)를 사용할 수 있습니다. 별표(*)는 0개 이상의 문자 조합을 나타내고 물음표(?)는 단일 문자를 나타냅니다. * 또는 ? 문자를 각 세그먼트에서 여러 번 사용할 수 있지만, 와일드카드 한 개를 여러 세그먼트에 걸쳐서 적용할 수는 없습니다. 다음은 경로가 /accounting인 IAM 사용자를 모두 나타낸 예제입니다.

```
"Resource": "arn:aws:iam::account-ID-without-hyphens:user/accounting/*"
```

다음은 특정 Amazon S3 버킷 내에 포함된 모든 항목을 나타낸 예제입니다.

```
"Resource": "arn:aws:s3::::my_corporate_bucket/*"
```

다수의 리소스를 지정할 수도 있습니다. 다음은 DynamoDB 테이블을 2개 나타낸 예제입니다.

```
"Resource": [
    "arn:aws:dynamodb:us-east-2:account-ID-without-hyphens:table/books_table",
    "arn:aws:dynamodb:us-east-2:account-ID-without-hyphens:table/magazines_table"
]
```

Resource 요소에서 ARN의 부분에 JSON 정책 변수 ([p. 524](#))를 사용하여 특정 리소스를 식별할 수 있습니다(ARN의 끝 부분에 사용). 예를 들어 {aws:username} 키를 리소스 ARN에 사용하여 현재 사용자의 이름을 리소스 이름에 추가해야 한다는 것을 나타낼 수 있습니다. 다음은 {aws:username} 요소에서 Resource 키를 사용하는 방법을 나타낸 예제입니다. 이 정책에서는 현재 사용자 이름과 일치하는 Amazon DynamoDB 테이블에 대한 액세스가 허용됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "dynamodb:*",
            "Resource": "arn:aws:dynamodb:us-east-2:ACCOUNT-ID-WITHOUT-HYPHENS:table/
${aws:username}"
        }
    ]
}
```

JSON 정책 변수에 대한 자세한 내용은 [IAM 정책 요소: 변수 및 태그 \(p. 524\)](#) 단원을 참조하십시오.

IAM JSON 정책 요소: NotResource

NotResource는 지정된 리소스의 목록을 제외한 모든 리소스와 명시적으로 일치하는 고급 정책 요소입니다. NotResource를 사용하면 일치하는 리소스의 긴 목록을 포함하는 대신 일치하지 않는 몇몇 리소스만 나열함으로써 정책을 줄일 수 있습니다. NotResource를 사용할 때는 이 요소에 지정된 리소스들이 제한되는 유일한 리소스라는 점을 유의해야 합니다. 따라서 다른 모든 서비스의 리소스를 포함하여 나열되지 않은 모든 리소스가 Allow 효과를 사용할 경우 허용되고, Deny 효과를 사용할 경우 거부됩니다. 설명문은 ARN을 사용하여 리소스를 지정하는 Resource 또는 NotResource 요소를 반드시 포함해야 합니다. ARN 형식에 대한 자세한 내용은 [Amazon 리소스 이름\(ARN\) 및 AWS 서비스 네임스페이스](#) 단원을 참조하십시오.

동일한 설명문에서 또는 동일한 정책의 다른 설명문에서 NotResource 요소와 "Effect": "Allow"를 사용할 경우 주의하십시오. NotResource는 명시적으로 나열되지 않은 모든 서비스 및 작업을 허용하므로 사용자에게 의도한 것보다 많은 권한을 부여하는 결과를 가져올 수 있습니다. 동일한 설명문에서 NotResource 요소와 "Effect": "Deny"를 사용하면 명시적으로 나열되지 않은 서비스 및 리소스를 거부합니다.

예를 들어 HRPayroll라는 이름의 그룹이 있다고 가정하겠습니다. 그리고 HRPayroll 멤버는 HRBucket 버킷의 Payroll 폴더를 제외하고 모든 Amazon S3 리소스에 액세스할 수 없습니다. 다음 정책은 나열된 리

소스 이외의 모든 Amazon S3 리소스에 대한 액세스를 거부합니다. 단, 이 정책은 사용자에게 리소스에 대한 액세스 권한을 부여하는 것이 아닙니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Deny",  
        "Action": "s3:*",  
        "NotResource": [  
            "arn:aws:s3:::HRBucket/Payroll",  
            "arn:aws:s3:::HRBucket/Payroll/*"  
        ]  
    }  
}
```

일반적으로 리소스에 대한 액세스를 명시적으로 거부하려면 "Effect": "Deny"를 사용하고 각 폴더를 개별적으로 나열하는 Resource 요소를 포함하는 정책을 작성합니다. 하지만 이때 사용자가 HRBucket에 폴더를 추가하거나 액세스하면 안 되는 Amazon S3에 리소스를 추가할 때마다 그 이름 역시 Resource 목록에 추가해야 합니다. 그렇지 않고 NotResource 요소를 사용할 때는 폴더 이름을 NotResource 요소에 추가하지 않더라도 사용자가 새 폴더에 대한 액세스 권한이 자동으로 거부됩니다.

IAM JSON 정책 요소: Condition

Condition 요소(또는 Condition 블록)를 사용하여 정책의 효력이 발생하는 시점에 대한 조건을 지정할 수 있습니다. Condition 요소는 선택 사항입니다. Condition 요소에서 조건 연산자 (p. 513)(equal, less than 등)를 사용하여 정책의 조건을 요청의 값에 일치시키는 표현식을 작성합니다. 예를 들어 조건 값에 날짜 또는 요청자의 IP 주소를 사용할 수 있습니다. 일부 서비스에서는 조건 값을 추가로 지정할 수도 있습니다. 예를 들어 Amazon EC2는 고유 기능으로서 ec2:InstanceType 키를 사용하여 조건을 작성하는 것이 가능합니다.

조건 키 이름은 대/소문자를 구분하지 않습니다. 예를 들어 aws:SourceIP 조건 키를 포함시키는 것은 AWS:SourceIp에 대한 테스트와 동일합니다. 조건 키 값의 대/소문자 구분은 사용하는 조건 연산자 (p. 513)에 따라 다릅니다. 예를 들어 다음 조건에는 StringEquals 연산자가 포함되어 johndoe에서 생성하는 요청만 일치하도록 합니다. 이름이 JohnDoe인 사용자는 액세스가 거부됩니다.

```
"Condition" : { "StringEquals" : { "aws:username" : "johndoe" }}
```

다음 조건은 StringEqualsIgnoreCase (p. 513) 연산자를 사용하여 이름이 johndoe 또는 JohnDoe인 사용자와 일치합니다.

```
"Condition" : { "StringEqualsIgnoreCase" : { "aws:username" : "johndoe" }}
```

일부 조건 키는 키 이름의 특정 부분을 지정하도록 허용하는 키-값 페어를 지원합니다. aws:RequestTag/[tag-key \(p. 551\)](#) 전역 조건 키, AWS KMS kms:EncryptionContext:[encryption_context_key](#) 및 여러 서비스에서 지원하는 ResourceTag/[tag-key](#) 조건 키가 그 예입니다. Amazon EC2와 같은 서비스에 대해 ResourceTag/[tag-key](#) 조건 키를 사용하는 경우 tag-key에 대한 키 이름을 지정해야 합니다. 키 이름은 대/소문자를 구분하지 않습니다. 따라서 정책의 조건 요소에서 "ec2:ResourceTag:TagKey1": "Value1" 지정을 수행한 경우 조건은 이름이 TagKey1 또는 tagkey1인 리소스 태그 키와 일치하지만, 두 가지 모두와 일치하지는 않습니다. 이러한 속성을 지원하는 AWS 서비스를 통해 대소문자만 다른 여러 키 이름을 생성할 수 있습니다(예: Amazon EC2 인스턴스에 foo=bar1 및 Foo=bar2 태그 지정). "ec2:ResourceTag:Foo": "bar1" 같은 조건을 사용하여 리소스에 대한 액세스를 허용하는 경우 키 이름은 두 태그 모두와 일치하지만, 하나의 값만 일치합니다. 이로 인해 예기치 않은 조건 실패가 발생할 수 있습니다.

Important

모범 사례로서 태그 또는 AWS KMS 암호화 컨텍스트와 같은 키-값 페어 속성 이름을 지정할 때 계정의 멤버가 일관적인 명명 규칙을 따르도록 해야 합니다. 태그 지정에 대

해 [aws:TagKeys \(p. 557\)](#) 조건 키를 사용하거나 AWS KMS 암호화 컨텍스트에 대해 [kms:EncryptionContextKeys](#) 사용을 통해 이를 적용할 수 있습니다.

- 모든 조건 연산자의 목록과 각 연산자의 작동 방식에 대한 설명을 보려면 [조건 연산자 \(p. 513\)](#)를 참조하십시오.
- 복수 값을 가진 조건 키를 취급하는 방법에 대한 설명은 [여러 키 값을 테스트하는 조건 생성\(설정 작업\) \(p. 520\)](#) 부분을 참조하십시오.
- 전역에서 사용 가능한 모든 조건 키의 목록은 [AWS 전역 조건 컨텍스트 키 \(p. 551\)](#) 단원을 참조하십시오.
- 각 서비스에서 정의된 조건 키는 [??? 단원](#)을 참조하십시오.

조건 블록

다음은 Condition 요소의 기본 형식을 나타낸 예제입니다.

```
"Condition": {  
    "DateGreaterThan" : {  
        "aws:CurrentTime" : "2013-12-15T12:00:00Z"  
    }  
}
```

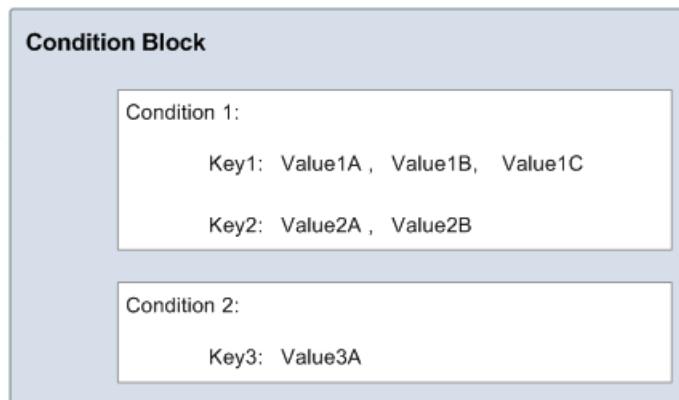
요청 값은 키로 나타내며, 여기에서는 `aws:CurrentTime`이 요청 값에 해당합니다. 키 값은 나중에 설명하겠지만 리터럴 값(2013-08-16T12:00:00Z) 또는 정책 변수로 지정하는 값과 비교됩니다. 비교 유형은 [조건 연산자 \(p. 513\)](#)에서 지정합니다(여기서는 `DateGreaterThan`). `equals`, `greater than` 및 `less than`과 같은 일반적인 부울 비교를 사용하여 문자열, 날짜, 숫자 등을 비교하는 조건을 만들 수 있습니다.

키에 다수의 값을 추가할 수 있는 경우도 있습니다. 예를 들어 Amazon DynamoDB에 대한 요청에서는 다수의 테이블 속성 반환이나 업데이트를 요청할 수 있습니다. DynamoDB 테이블에 대한 액세스 정책에 따르면 `dynamodb:Attributes` 키를 추가하여 요청 시 나열되는 모든 속성 저장이 가능합니다. Condition 요소의 설정 연산자를 사용하여 정책에 허용된 속성 목록과 요청에 포함된 속성 여러 가지를 비교함으로써 테스트할 수 있습니다. 자세한 내용은 [여러 키 값을 테스트하는 조건 생성\(설정 작업\) \(p. 520\)](#) 단원을 참조하십시오.

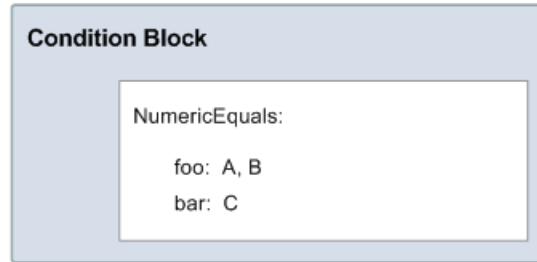
요청 단계에서 정책을 평가할 때는 AWS가 키를 해당하는 요청 값으로 변환합니다. (이 예제에서는 AWS가 요청 날짜와 시간을 사용합니다). 조건 평가에 따라 `true` 또는 `false`가 반환되고, 이후 이 조건 평가 결과를 고려하여 정책 전반적인 요청 허용 또는 거부 여부를 결정합니다.

다수의 조건 값

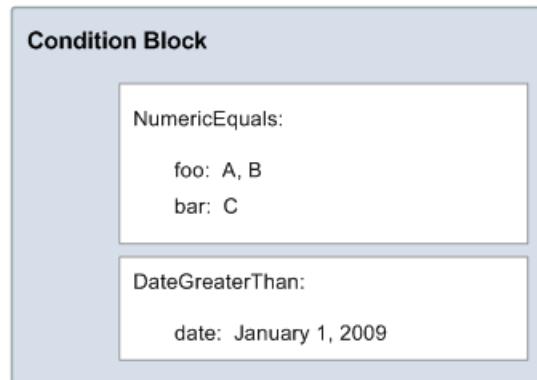
Condition 요소에는 여러 조건을 추가할 수 있으며, 다시 한 번 각 조건마다 다수의 키-값 페어가 포함됩니다. 다음은 이것을 설명한 그림입니다. 달리 지정하지 않는 경우 모든 키는 다수의 값을 가질 수 있습니다.



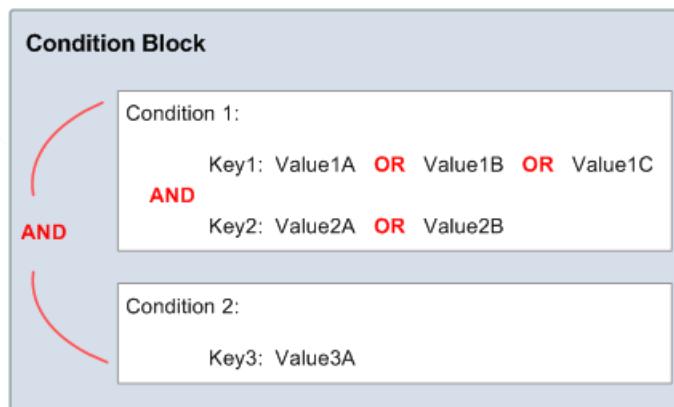
예를 들어, 숫자 값 foo가 A 또는 B와 일치하고, 다른 숫자 값 bar가 C와 일치하는 경우에 한해 John에게 리소스 사용을 허용한다고 가정하겠습니다. 이 경우 다음 그림과 같이 조건 블록을 생성하게 됩니다.



이번에는 2009년 1월 1일 이후에 대한 John의 액세스 권한을 제한한다고 가정하겠습니다. 그렇다면 다른 조건으로 2009년 1월 1일에 해당하는 날짜와 함께 DateGreaterThanOrEqualTo 추가해야 합니다. 조건 블록의 모습은 다음 그림과 같습니다.



조건 연산자가 다수이거나 단일 조건 연산자에 여러 키가 추가된 경우에는 논리 연산자인 AND를 사용하여 조건을 평가합니다. 단일 조건 연산자에 키 하나마다 여러 값이 포함된 경우에는 논리 연산자 OR를 사용하여 해당 조건 연산자를 평가합니다. 허용 또는 명시적 거부를 위해서는 모든 조건 연산자가 충족되어야 합니다. 조건 연산자 중 하나라도 충족되지 않을 경우 결과는 거부입니다.



언급한 바와 같이 AWS에는 사전 정의된 조건 연산자 및 키가 있습니다(aws:currentTime 등). 마찬가지로 AWS 서비스 역시 각각 정의되어 있는 키가 따로 있습니다.

예를 들어 다음과 같은 조건에서 사용자 John에게 Amazon SQS 대기열에 대한 액세스를 허용한다고 가정하겠습니다.

- 시간은 2013년 8월 16일 정오 12:00 이후입니다.
- 시간은 2013년 8월 16일 오후 3:00 이전입니다.
- 요청(IAM 또는 Amazon SQS) 또는 메시지(Amazon SNS)가 전송되는 IP 주소의 범위는 192.0.2.0~192.0.2.255 또는 203.0.113.0~203.0.113.255입니다.

조건 블록에는 서로 다른 세 개의 조건 연산자가 있으며, John이 대기열, 주제 또는 리소스에 액세스하려면 이 세 가지 조건 연산자가 모두 충족되어야 합니다.

다음은 정책의 조건 블록을 나타낸 예제입니다. `aws:SourceIp`의 값 2개는 OR로 평가합니다. 서로 다른 세 개의 조건 연산자는 AND를 사용하여 평가합니다.

```
"Condition" : {
    "DateGreaterThan" : {
        "aws:CurrentTime" : "2013-08-16T12:00:00Z"
    },
    "DateLessThan": {
        "aws:CurrentTime" : "2013-08-16T15:00:00Z"
    },
    "IpAddress" : {
        "aws:SourceIp" : ["192.0.2.0/24", "203.0.113.0/24"]
    }
}
```

마지막으로 정책의 개별 키에 다수의 값을 추가할 수 있는 경우도 있습니다. 조건 설정 연산자를 사용하여 정책에 나열된 하나 이상의 값에 대해 다중 값 키를 테스트할 수 있습니다. 자세한 내용은 [여러 키 값을 테스트하는 조건 생성\(설정 작업\) \(p. 520\)](#) 단원을 참조하십시오.

IAM JSON; 정책 요소: 조건 연산자

조건 연산자는 조건의 "동사"이며, IAM에서 수행할 비교 유형을 지정합니다. 조건 연산자는 다음 범주로 그룹화할 수 있습니다.

- [문자열 \(p. 513\)](#)
- [숫자 \(p. 515\)](#)
- [날짜 및 시간 \(p. 515\)](#)
- [부울 \(p. 516\)](#)
- [Binary \(p. 516\)](#)
- [IP 주소 \(p. 517\)](#)
- [Amazon 리소스 이름\(ARN\) \(p. 518\)](#)(일부 서비스에서만 사용 가능.)
- [...IfExists \(p. 518\)](#)(키 값이 다른 확인을 위해 존재하는지 여부를 확인)
- [Null 확인 \(p. 519\)](#)(키 값이 단독 확인을 위해 존재하는지 여부를 확인)

문자열 조건 연산자

문자열 조건 연산자를 사용하여 키와 문자열 값을 비교한 결과에 따라 액세스를 제한하는 `Condition` 요소를 생성할 수 있습니다.

조건 연산자	설명
<code>StringEquals</code>	정확한 일치, 대소문자 구분
<code>StringNotEquals</code>	불일치
<code>StringEqualsIgnoreCase</code>	정확한 일치, 대소문자 무시

조건 연산자	설명
<code>StringNotEqualsIgnoreCase</code>	불일치, 대소문자 무시
<code>StringLike</code>	대소문자 구분 일치. 문자열 어디에서나 다중 문자 매칭 와일드카드(*) 또는 단일 문자 매칭 와일드카드(?)를 값에 포함할 수 있습니다. Note 키에 다수의 값이 저장되는 경우에는 설정 연산자 <code>ForAllValues:StringLike</code> 및 <code>ForAnyValue:StringLike</code> 를 사용해 <code>StringLike</code> 를 한정할 수 있습니다. 자세한 정보는 여러 키 값을 테스트하는 조건 생성(설정 작업) (p. 520) 단원을 참조하십시오.
<code>StringNotLike</code>	대소문자 구분 불일치. 문자열 어디에서나 다중 문자 매칭 와일드카드(*) 또는 단일 문자 매칭 와일드카드(?)를 값에 포함할 수 있습니다.

예를 들어, 다음 문에는 Condition 조건 연산자에 `StringEquals` 키를 사용하여 요청에 사용자 에이전트 헤더의 특정 값을 추가해야 한다고 지정하는 `aws:UserAgent` 요소가 포함되어 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iam:*AccessKey*",
    "Resource": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:user/*",
    "Condition": {"StringEquals": {"aws:UserAgent": "Example Corp Java Client"}}
  }
}
```

다음은 정책 변수 (p. 524)와의 문자열 일치를 수행하는 `StringLike` 조건 연산자를 사용하여 정책을 만드는 예제입니다. 이 정책에서는 IAM 사용자가 Amazon S3 콘솔을 사용하여 Amazon S3 버킷에 있는 자신의 '홈 디렉터리'를 관리할 수 있습니다. 이 정책은 `s3:prefix`가 지정된 패턴 중 하나와 일치하는 경우 S3 버킷에서 지정된 작업을 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3>ListAllMyBuckets",
        "s3:GetBucketLocation"
      ],
      "Resource": "arn:aws:s3::::*"
    },
    {
      "Effect": "Allow",
      "Action": "s3>ListBucket",
      "Resource": "arn:aws:s3:::BUCKET-NAME",
      "Condition": {"StringLike": {"s3:prefix": [
        "",
        "home/",
        "home/${aws:username}/"
      ]}}
    },
    {
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": [
        ...
      ]
    }
  ]
}
```

```

        "arn:aws:s3:::BUCKET-NAME/home/${aws:username}",
        "arn:aws:s3:::BUCKET-NAME/home/${aws:username}/*"
    ]
}
]
```

웹 자격 증명 연동 시 `Condition` 요소를 사용하여 애플리케이션 ID와 사용자 ID에 따라 리소스 액세스를 제한하는 방법을 나타낸 정책 예는 [Amazon S3: Amazon Cognito 사용자가 자신의 버킷에 있는 객체에 액세스 할 수 있도록 허용 \(p. 372\)](#) 단원을 참조하십시오.

숫자 조건 연산자

숫자 조건 연산자를 사용하여 키와 정수 또는 십진수 값을 비교한 결과에 따라 액세스를 제한하는 `Condition` 요소를 생성할 수 있습니다.

조건 연산자	설명
<code>NumericEquals</code>	일치
<code>NumericNotEquals</code>	불일치
<code>NumericLessThan</code>	"미만" 일치
<code>NumericLessThanEquals</code>	"이하" 일치
<code>NumericGreaterThan</code>	"초과" 일치
<code>NumericGreaterThanOrEqual</code>	"이상" 일치

예를 들어, 다음 문에는 `NumericLessThanEquals` 조건 연산자에 `s3:max-keys` 키를 사용하여 요청자가 `example_bucket`에서 한 번에 최대 10개까지 객체를 나열할 수 있다고 지정하는 `Condition` 요소가 포함되어 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "s3>ListBucket",
    "Resource": "arn:aws:s3:::example_bucket",
    "Condition": {"NumericLessThanEquals": {"s3:max-keys": "10"}}
  }
}
```

날짜 조건 연산자

날짜 조건 연산자를 사용하여 키와 날짜/시간 값을 비교한 결과에 따라 액세스를 제한하는 `Condition` 요소를 생성할 수 있습니다. 이러한 조건 연산자는 `aws:currentTime` 키 또는 `aws:EpochTime` 키와 함께 사용합니다. 날짜/시간 값은 [ISO 8601 날짜 형식의 W3C 구현](#) 값 하나로 혹은 epoch(UNIX) 시간으로 지정해야 합니다.

Note

날짜 조건 연산자에는 와일드카드를 사용할 수 없습니다.

조건 연산자	설명
<code>DateEquals</code>	특정 날짜 일치

조건 연산자	설명
DateNotEquals	불일치
DateLessThan	특정 날짜/시간 이전에 일치
DateLessThanEquals	특정 날짜/시간 또는 이전에 일치
DateGreaterThan	특정 날짜/시간 이후에 일치
DateGreaterThanOrEqual	특정 날짜/시간 또는 이후에 일치

예를 들어, 다음 문에는 Condition 조건 연산자에 DateLessThan 키를 사용하여 2013년 6월 30일 이전에 요청이 수신되어야 한다고 지정하는 aws:currentTime 요소가 포함되어 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iam:*AccessKey*",
    "Resource": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:user/*",
    "Condition": {"DateLessThan": {"aws:currentTime": "2013-06-30T00:00:00Z"}}
  }
}
```

부울 조건 연산자

부울 조건을 사용하여 키를 "true" 또는 "false"와 비교하고 그에 따라 액세스를 제한하는 Condition 요소를 생성할 수 있습니다.

조건 연산자	설명
Bool	부울 일치

예를 들어, 다음 문은 Bool 조건 연산자에 aws:SecureTransport 키를 사용하여 요청에 SSL을 사용해야 한다고 지정하고 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iam:*AccessKey*",
    "Resource": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:user/*",
    "Condition": {"Bool": {"aws:SecureTransport": "true"}}
  }
}
```

이진 조건 연산자

BinaryEquals 조건 연산자를 사용하면 이진 형식의 키 값을 테스트하는 Condition 요소를 생성할 수 있습니다. 지정한 키 값을 정책 내 이진 값의 [base-64](#) 인코딩 표시와 바이트 단위(byte for byte)로 비교합니다.

```
"Condition" : {
  "BinaryEquals": {
    "key" : "QmluYXJ5VmFsdWVJbkJhc2U2NA=="
  }
}
```

IP 주소 조건 연산자

IP 주소 조건 연산자를 사용하여 IPv4/IPv6 주소 또는 IP 주소 범위와 키를 비교한 결과에 따라 액세스를 제한하는 Condition 요소를 생성할 수 있습니다. 이 조건에는 `aws:SourceIp` 키가 사용됩니다. 값은 표준 CIDR 형식(예: 203.0.113.0/24 또는 2001:DB8:1234:5678::/64)을 따라야 합니다. 연결된 라우팅 접두사 없이 IP 주소를 지정하면 IAM은 기본 접두사 값 /32를 사용합니다.

일부 AWS 서비스는 0의 범위를 나타내기 위해 ::을 사용해 IPv6를 지원합니다. 서비스가 IPv6를 지원하는지 여부를 확인하려면 서비스 설명서를 참조하십시오.

조건 연산자	설명
<code>IpAddress</code>	지정된 IP 주소 또는 범위
<code>NotIpAddress</code>	지정된 IP 주소 또는 범위를 제외한 모든 IP 주소

예를 들어, 다음 문은 `IpAddress` 조건 연산자에 `aws:SourceIp` 키를 사용하여 IP 범위 203.0.113.0 - 203.0.113.255에서 요청이 전송되어야 한다고 지정하고 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "iam:*AccessKey*",
    "Resource": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:user/*",
    "Condition": {"IpAddress": {"aws:SourceIp": "203.0.113.0/24"}}
  }
}
```

`aws:SourceIp` 조건 키는 요청이 전송되는 IP 주소를 확인합니다. 요청이 Amazon EC2 인스턴스에서 전송된 경우에는 `aws:SourceIp`가 인스턴스의 퍼블릭 IP 주소로 계산되어야 합니다.

다음 예제에서는 IPv4와 IPv6 주소를 혼합하여 조직의 유효 IP 주소를 모두 표현하는 방법을 보여줍니다. IPv6으로 전환하는 동안 조직의 정책이 계속 적용되도록 하려면 기존의 IPv4 주소 범위에 IPv6 범위를 더하여 정책을 보완하는 것이 좋습니다.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "someservice: *",
    "Resource": "*",
    "Condition": {
      "IpAddress": {
        "aws:SourceIp": [
          "203.0.113.0/24",
          "2001:DB8:1234:5678::/64"
        ]
      }
    }
  }
}
```

사용자 자격으로 직접 테스트한 API를 호출하는 경우 `aws:SourceIp` 조건 키는 JSON 정책에서만 작동합니다. 서비스를 사용하여 사용자를 대신해 대상 서비스를 호출하는 경우, 대상 서비스는 원래 사용자의 IP 주소 대신 호출 서비스의 IP 주소를 봅니다. 이러한 상황은 예를 들어 AWS CloudFormation을 사용하여 인스턴스를 생성하는 Amazon EC2를 호출하는 경우에 발생할 수 있습니다. 현재로서는 JSON 정책에 따라 평가하기 위해 원본 IP 주소를 호출 서비스를 통해 대상 서비스로 보낼 방법이 없습니다. 이러한 서비스 API 호출 유형의 경우 `aws:SourceIp` 조건 키를 사용하지 마십시오.

Amazon 리소스 이름(ARN) 조건 연산자

Amazon 리소스 이름(ARN) 조건 연산자를 사용하면 키와 ARN을 비교한 결과에 따라 액세스를 제한하는 Condition 요소를 생성할 수 있습니다. ARN은 문자열로 알려져 있습니다. 이 값을 일부 서비스에서만 사용이 가능하기 때문에 ARN으로 비교할 수 있는 요청 값을 지원하는 서비스도 제한적입니다.

조건 연산자	설명
ArnEquals, ArnLike	ARN 대소문자 구분 일치. ARN에서 콜론으로 구분된 구성요소 6개는 각각 별도로 확인하며, 다중 문자 매칭 와일드카드(*) 또는 단일 문자 매칭 와일드카드(?)가 추가될 수 있습니다. 이들은 동일하게 동작합니다.
ArnNotEquals, ArnNotLike	ARN 불일치. 이들은 동일하게 동작합니다.

다음 예제에서는 SNS 메시지를 전송할 모든 Amazon SQS 대기열에 첨부해야 하는 정책을 보여 줍니다. 이 예제에서는 선택한 대기열로 메시지를 전송할 수 있는 권한을 Amazon SNS에 부여하고 있습니다. 단, 서비스에서 특정 Amazon SNS 주제와 관련하여 메시지를 전송하는 경우로 제한됩니다. 대기열을 Resource 필드에, 그리고 Amazon SNS 주제는 SourceArn 키 값으로 지정합니다.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {"AWS": "123456789012"},
    "Action": "SQS:SendMessage",
    "Resource": "arn:aws:sqs:REGION:123456789012:QUEUE-ID",
    "Condition": {"ArnEquals": {"aws:SourceArn": "arn:aws:sns:REGION:123456789012:TOPIC-ID"}}
  }
}
```

IfExists 조건 연산자

Null 조건 - 예를 들어 StringLikeIfExists를 제외한 모든 조건 연산자 이름 끝에 IfExists를 추가할 수 있습니다. 이렇게 하면 "요청 컨텍스트에 정책 키가 있으면 정책에 지정된 대로 키를 처리하고, 키가 없으면 조건 요소를 true로 평가합니다." 문의 다른 조건 요소는 여전히 불일치한 결과를 발생시킬 수 있지만 ...IfExists로 확인하면 누락되는 키는 없습니다.

IfExists 사용 예제

대부분 조건 키는 특정 형식의 리소스 정보를 의미하기 때문에 해당 형식의 리소스에 액세스할 때만 존재합니다. 이러한 조건 키는 다른 형식의 리소스에는 표시되지 않습니다. 그렇다고 정책 문이 한 가지 형식의 리소스에만 적용된다고 해서 문제가 되지는 않습니다. 하지만 정책 문이 여러 서비스의 작업을 참조하는 경우나, 혹은 한 가지 서비스 내에서 임의의 작업이 동일한 서비스에서 여러 가지 다른 리소스 형식에 액세스하는 경우처럼 단일 문이 여러 유형의 리소스에 적용될 수 있는 경우도 있습니다. 이런 경우 오직 한 가지 리소스에만 적용되는 조건 키를 정책 문에 추가하면 정책 문의 Condition 요소를 충족하지 못하고 결국 "Effect"가 적용되지 않습니다.

예를 들어 다음과 같은 정책 예제를 살펴보십시오.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "THISPOLICYDOESNOTWORK",
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": "*",
  }
}
```

```

    "Condition": {"StringLike": {"ec2:InstanceType": [
        "t1.*",
        "t2.*",
        "m3.*"
    ]}}
}
}

```

위 정책은 사용자가 t1, t2 또는 m3 형식의 인스턴스를 모두 실행할 수 있도록 하는 것이 목적입니다. 하지만 실제로 인스턴스를 실행하려면 인스턴스 외에도 이미지, 키 페어, 보안 그룹 등 다양한 리소스에 액세스해야 합니다. 전체 문은 인스턴스를 실행하는 데 필요한 모든 리소스와 비교하여 평가됩니다. 하지만 이러한 추가 리소스에는 ec2:InstanceType 조건 키가 없기 때문에 StringLike 검사는 fail로 끝나고 사용자에게 권한이 부여되지 않아 어떤 인스턴스 유형도 실행하지 못합니다. 이 문제를 해결하려면 그 대신 StringLikeIfExists 조건 연산자를 사용해야 합니다. 이렇게 하면 조건 키가 존재하는 경우에만 테스트가 실행됩니다. 그 결과 다음 예제는 이렇게 해석할 수 있습니다. '검사 대상 리소스에 'ec2:InstanceType' 조건 키가 있으면 키 값이 "t1.*", "t2.*" 또는 "m3.*"로 시작할 때에만 작업을 허용한다.' 검사 대상 리소스에 조건 키가 없으면 그냥 듭니다." DescribeActions 문에는 콘솔에서 해당 인스턴스를 보는 데 필요한 작업이 포함되어 있습니다.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RunInstances",
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "*",
            "Condition": {
                "StringLikeIfExists": {
                    "ec2:InstanceType": [
                        "t1.*",
                        "t2.*",
                        "m3.*"
                    ]}
            },
            {
                "Sid": "DescribeActions",
                "Effect": "Allow",
                "Action": [
                    "ec2:DescribeImages",
                    "ec2:DescribeInstances",
                    "ec2:DescribeVpcs",
                    "ec2:DescribeKeyPairs",
                    "ec2:DescribeSubnets",
                    "ec2:DescribeSecurityGroups"
                ],
                "Resource": "*"
            }
        ]
    }
}

```

조건 키의 존재를 확인하는 조건 연산자

Null 조건 연산자를 사용하여 권한을 부여하는 시점에 조건 키의 유무를 검사할 수 있습니다. 정책 문에서는 true(키가 부재하며 — 값이 null임) 또는 false(키가 존재하며 값이 null이 아님)를 사용합니다.

예를 들어, 이 조건 연산자를 사용하여 작업 시 사용자가 자신의 자격 증명을 사용하는지, 혹은 임시 자격 증명을 사용하는지 알 수 있습니다. 사용자가 임시 자격 증명을 사용하는 경우에는 aws:TokenIssueTime 키가 존재하며, 값을 갖고 있습니다. 다음은 Amazon EC2 API 사용자의 경우 임시 자격 증명의 사용이 제한된다는 것(키가 존재해서는 안 됨)을 명시하는 조건을 나타낸 예제입니다.

```
{
}
```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Action": "ec2:*",
        "Effect": "Allow",
        "Resource": "*",
        "Condition": {"Null": {"aws:TokenIssueTime": "true"}}
    }
]
```

여러 키 값을 테스트하는 조건 생성(설정 작업)

정책의 Condition 요소를 사용하여 요청의 단일 키에 대한 여러 값을 테스트할 수 있습니다. AWS에 요청 할 경우, 프로그래밍 방식으로 또는 AWS Management 콘솔을 통하여 요청 정보를 포함합니다. 조건 키를 사용하여 요청의 일치하는 키의 값을 테스트할 수 있습니다. 예를 들어, 조건 키를 사용하여 DynamoDB 테이블의 특정 속성 또는 태그에 근거한 Amazon EC2 인스턴스에 대한 액세스를 제어할 수 있습니다.

단일 키에 대한 여러 값을 포함하는 요청의 경우, `ForAllValues` 한정자를 사용하여 요청 값의 전체 집합이 조건 키의 전체 집합과 일치하는지 테스트합니다. 요청 값의 집합의 1개 이상의 멤버가 조건 키 값의 집합의 1개 이상의 멤버와 일치하는지 테스트하려면 `ForAnyValue` 한정자를 사용합니다. 이러한 한정자는 조건 연산자에 설정 작업 기능을 추가하므로 여러 요청 값을 여러 조건 값에 대해 테스트할 수 있습니다.

요청을 보내는 경우, AWS가 해당 정책을 평가하여 조건이 `true`인지 `false`인지 확인합니다. 다수의 키 값이 포함된 요청의 경우, AWS가 다음의 한정자에 근거하여 조건을 평가합니다.

- `ForAllValues` – 요청의 지정된 키 값과 최소 한 개의 정책 값이 일치하면 조건이 `true`를 반환합니다. 또한 요청에 일치하는 키가 없거나 키 값이 빈 문자열 등 빈 데이터 세트로 확인되는 경우 `true`를 반환합니다.
- `ForAnyValue` – 요청의 키 값 중 하나가 정책의 조건 값 중 하나와 일치하면 조건이 `true`를 반환합니다. 일치하는 키가 없거나 빈 데이터 세트인 경우 조건에서 `false`를 반환합니다.

목차

- [조건 설정 연산자 예제 \(p. 520\)](#)
- [조건 설정 연산자에 대한 평가 로직 \(p. 522\)](#)

조건 설정 연산자 예제

정책을 만들어 그 정책에 지정된 하나 이상의 값에 대해 요청의 여러 값을 테스트할 수 있습니다. 기술 지원 포럼의 스레드에 대한 정보를 저장하는 데 사용되는 `Thread`라는 Amazon DynamoDB 테이블이 있다고 가정해 보십시오. 이 테이블에는 `ID`, `UserName`, `PostDateTime`, `Message`, `Tags`라는 이름의 속성이 있을 수 있습니다.

```
{
    ID=101
    UserName=Bob
    PostDateTime=20130930T231548Z
    Message="A good resource for this question is http://aws.amazon.com/documentation/"
    Tags=[ "AWS", "Database", "Security" ]
}
```

DynamoDB에서 설정된 연산자를 사용하여 개별 데이터 항목 및 속성에 대한 세부적인 액세스를 구현하는 방법은 Amazon DynamoDB 개발자 안내서 가이드의 [DynamoDB에 대한 세분화된 액세스 제어 단원](#)을 참조 하십시오.

`PostDateTime`, `Message`, `Tags` 속성만 볼 수 있도록 허용하는 정책을 만들 수 있습니다. 사용자의 요청에 이러한 속성이 포함된 경우 요청이 허용됩니다. 그러나 요청에 다른 속성(예: `ID`)이 포함된 경우 요청은 거부 됩니다. 논리적으로 말하면, 허용되는 속성(`PostDateTime`, `Message`, `Tags`) 목록을 생성하고, 사용자가 요청한 모든 속성이 허용되는 속성 목록에 포함되어야 한다고 정책에 명시하는 것입니다.

다음 정책 예제는 `ForAllValues` 한정자를 `StringEquals` 조건 연산자와 함께 사용하는 방법을 보여줍니다. 이 조건에서는 사용자가 `Thread`라는 DynamoDB 테이블에서 `ID`, `Message` 또는 `Tags` 속성만 요청하도록 허용합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "dynamodb:GetItem",  
            "Resource": "arn:aws:dynamodb:*::table/Thread",  
            "Condition": {  
                "ForAllValues:StringEquals": {  
                    "dynamodb:Attributes": [  
                        "ID",  
                        "Message",  
                        "Tags"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

사용자가 `Thread` 테이블에서 `Message` 및 `Tags` 속성을 가져오기 위해 DynamoDB에 요청하는 경우 요청은 허용됩니다. 사용자가 요청한 속성이 정책에 지정된 값과 모두 일치하기 때문입니다. `GetItem` 작업을 하려면 사용자는 `ID` 속성을 정책에서도 허용되는 데이터베이스 테이블 키로 전달해야 합니다. 그러나 사용자의 요청에 `UserName` 속성이 포함되어 있으면 요청은 실패합니다. 왜냐하면 `UserName`은 허용되는 속성 목록에 들어 있지 않고, `ForAllValues` 한정자는 요청된 모든 값이 정책에 나열되어 있을 것을 요구하기 때문입니다.

Important

`dynamodb:Attributes`를 사용하는 경우 해당 정책에 나열된 테이블 및 보조 인덱스에 대한 모든 기본 키 및 인덱스 속성의 이름을 지정해야 합니다. 그렇지 않으면, DynamoDB에서 이러한 키 속성을 사용하여 요청한 작업을 수행할 수 없습니다.

또는 사용자가 `ID` 및 `UserName` 등 일부 속성을 요청에 포함시키는 것을 명시적으로 금지할 수 있습니다. 예를 들어, 업데이트(PUT 작업)로 인해 특정 속성이 변경되지 않도록 사용자가 DynamoDB 테이블을 업데이트 할 때 일부 속성을 제외할 수 있습니다. 이 경우에는 금지된 속성(`ID`, `UserName`) 목록을 생성합니다. 그러면 사용자가 요청한 속성 중 금지된 속성이 있는 경우, 요청이 거부됩니다.

다음 예제에서는 사용자가 `ForAnyValue` 작업을 수행하려는 경우, `ID` 한정자를 사용해 `PostDateTime` 및 `PutItem` 속성에 대한 액세스를 거부하는 방법을 보여줍니다. 즉 사용자가 `Thread` 테이블에 있는 이 두 가지 속성 중 어느 하나를 업데이트하려 하는 경우를 말합니다. `Effect` 요소는 `Deny`로 설정됩니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Deny",  
        "Action": "dynamodb:PutItem",  
        "Resource": "arn:aws:dynamodb:*::table/Thread",  
        "Condition": {  
            "ForAnyValue:StringEquals": {  
                "dynamodb:Attributes": [  
                    "ID",  
                    "PostDateTime"  
                ]  
            }  
        }  
    }  
}
```

}

사용자가 PostDateTime 테이블의 Message 및 Thread 속성의 업데이트를 요청한다고 가정할 경우, ForAnyValue 한정자는 요청된 속성 중 정책 목록에 표시되는 것이 있는지 여부를 결정합니다. 이 경우 하나가 일치하므로(PostDateTime) 조건은 true입니다. 요청의 다른 값(예: 리소스)도 일치한다고 가정하면, 전체 정책 평가는 true를 반환합니다. 정책의 효력이 Deny이므로 요청은 거부됩니다.

한편 사용자가 PutItem 속성만으로 UserName 수행을 요청한다고 가정해 보겠습니다. 요청의 속성 (UserName이 유일) 중 어떤 것도 정책에 나열된 속성(ID, PostDateTime)과 일치하지 않습니다. 조건이 false를 반환하므로 정책의 효력(Deny) 또한 false이고, 따라서 이 정책은 요청을 거부하지 않습니다. (요청이 성공하려면 다른 정책에서 이를 명시적으로 허용해야 합니다. 이 요청은 이 정책에 의해 명시적으로 거부되지 않지만, 모든 요청은 둑시적으로 거부됩니다.)

조건 설정 연산자에 대한 평가 로직

이 단원에서는 ForAllValues 및 ForAnyValue 한정자와 함께 사용되는 평가 로직의 세부 사항을 다릅니다. 요청에 포함될 수 있는 키(PostDateTime 및 UserName)와 PostDateTime, Message 및 Tags 값을 포함하는 정책 조건이 아래 표에 나와 있습니다.

키(요청)	조건 값(정책)
PostDateTime	PostDateTime
UserName	Message
	Tags

조합에 대한 평가는 다음과 같습니다.

PostDateTime matches PostDateTime?

PostDateTime matches Message?

PostDateTime matches Tags?

UserName matches PostDateTime?

UserName matches Message?

UserName matches Tags?

조건 연산자의 결과는 정책 조건에 사용된 변경자에 따라 달라집니다.

- ForAllValues를 선택하십시오. 요청의 모든 키(PostDateTime 또는 UserName)가 최소 한 개의 정책 조건 값(PostDateTime, Message, Tags)과 일치하면, 조건 연산자는 true를 반환합니다. 다시 말해, 조건이 true가 되려면 (PostDateTime은 PostDateTime, Message 또는 Tags와 일치) 그리고 (UserName은 PostDateTime, Message 또는 Tags와 일치) 둘 다 만족해야 합니다.
- ForAnyValue. 요청 값과 정책 값의 여섯 가지 조합 중 하나에서 true를 반환하면 조건 연산자는 true를 반환합니다.

다음 정책은 ForAllValues 한정자를 포함합니다.

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

```

    "Statement": {
        "Effect": "Allow",
        "Action": "dynamodb:GetItem",
        "Resource": "arn:aws:dynamodb:*:*:table/Thread",
        "Condition": {
            "ForAllValues:StringEquals": {
                "dynamodb:Attributes": [
                    "PostDateTime",
                    "Message",
                    "Tags"
                ]
            }
        }
    }
}

```

사용자가 PostDateTime 및 UserName 속성을 가져오기 위해 DynamoDB에 요청을 한다고 가정하겠습니다. 조합에 대한 평가는 다음과 같습니다.

PostDateTime matches PostDateTime?	True
PostDateTime matches Message?	False
PostDateTime matches Tags?	False
UserName matches PostDateTime?	False
UserName matches Message?	False
UserName matches Tags?	False

이 정책에는 ForAllValues 조건 연산 변수가 포함되어 있는데, 이는 PostDateTime 일치 항목과 UserName 일치 항목이 최소한 하나는 있어야 한다는 것을 뜻합니다. UserName과 일치하는 항목이 없으므로 조건 연산자는 false를 반환하며, 정책은 요청을 허용하지 않습니다.

다음 정책은 ForAnyValue 한정자를 포함합니다.

```

{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "dynamodb:PutItem",
        "Resource": "arn:aws:dynamodb:*:*:table/Thread",
        "Condition": {
            "ForAnyValue:StringEquals": {
                "dynamodb:Attributes": [
                    "ID",
                    "PostDateTime"
                ]
            }
        }
    }
}

```

이 정책은 "Effect": "Deny"를 포함하며, 작업은 PutItem입니다. 사용자가 PutItem, UserName 및 Message 속성을 포함하는 PostDateTime 요청을 한다고 가정하겠습니다. 평가는 다음과 같습니다.

UserName matches ID?	False
----------------------	-------

UserName matches PostDateTime?	False
Messages matches ID?	False
Message matches PostDateTime?	False
PostDateTime matches ID?	False
PostDateTime matches PostDateTime?	True

`ForAnyValue` 변경자에 따라, 이러한 테스트 중 하나가 `true`를 반환하면 조건은 `true`를 반환합니다. 마지막 테스트가 `true`를 반환하므로 조건은 `true`입니다. `Effect` 요소가 `Deny`로 설정되어 있으므로 요청은 거부됩니다.

Note

요청의 키 값이 빈 데이터 세트(예: 빈 문자열)로 확인되면 `ForAllValues`에서 수정한 조건 연산자는 `true`를 반환하고, `ForAnyValue`에서 수정한 조건 연산자는 `false`를 반환합니다.

IAM 정책 요소: 변수 및 태그

주제

- [소개 \(p. 524\)](#)
- [정책 변수로서의 태그 \(p. 526\)](#)
- [정책 변수를 사용할 수 있는 경우 \(p. 526\)](#)
- [정책 변수로 사용할 수 있는 요청 정보 \(p. 528\)](#)
- [자세한 정보 \(p. 530\)](#)

소개

IAM 정책에서는 다양한 작업을 통해 액세스를 제어하려는 특정 리소스에 이름을 지정할 수 있습니다. 예를 들어 다음은 사용자가 Amazon S3 버킷 `mybucket`에서 접두사 `David`가 사용된 객체를 표시하거나, 읽거나, 쓸 수 있는 정책입니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": ["s3>ListBucket"],
            "Effect": "Allow",
            "Resource": ["arn:aws:s3:::mybucket"],
            "Condition": {"StringLike": {"s3:prefix": ["David/*"]}}
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Effect": "Allow",
            "Resource": ["arn:aws:s3:::mybucket/David/*"]
        }
    ]
}
```

정책을 작성하다 보면 정확한 리소스 이름을 모를 때도 있습니다. 사용자마다 고유한 정책 사본을 만들 필요 없이 여러 사용자에게 작용하도록 정책을 일반화해야 할 수 있습니다. 예를 들어 앞의 예와 마찬가지로 사용

자마다 Amazon S3 버킷에 자신의 객체를 액세스하도록 허용하는 정책을 쓸 수도 있습니다. 그러나 리소스의 일부로 사용자의 이름을 명시적으로 지정하는 각 사용자에 대해 별도의 정책을 만들지 마십시오. 대신 해당 그룹의 모든 사용자에 대해 작동하는 단일 그룹 정책을 만듭니다.

이때는 정책에 자리 표시자를 지정할 수 있는 정책 변수 기능을 사용하면 가능합니다. 정책을 평가할 때는 이 정책 변수가 요청 자체의 맥락에서 온 값으로 바뀝니다.

다음은 Amazon S3 버킷에서 정책 변수를 사용하는 정책을 나타낸 예제입니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": ["s3>ListBucket"],  
            "Effect": "Allow",  
            "Resource": ["arn:aws:s3:::mybucket"],  
            "Condition": {"StringLike": {"s3:prefix": ["${aws:username}/*"]}}  
        },  
        {  
            "Action": [  
                "s3:GetObject",  
                "s3:PutObject"  
            ],  
            "Effect": "Allow",  
            "Resource": ["arn:aws:s3:::mybucket/${aws:username}/*"]  
        }  
    ]  
}
```

이 정책을 평가할 때는 IAM이 \${aws:username} 변수를 실제 현재 사용자의 [알기 쉬운 이름 \(p. 480\)](#)으로 대체합니다. 이 말은 사용자 그룹에 단일 정책을 적용하여 사용자 이름을 리소스 이름 일부로 사용함으로써 버킷에 대한 액세스 제어가 가능함을 의미합니다.

변수는 \$ 접두사 뒤에 중괄호({ })를 사용하여 표시합니다. \${ } 문자 안에는 정책에서 사용할 요청 값의 이름을 추가할 수 있습니다. 사용할 수 있는 값은 이 페이지 후반에서 다루겠습니다.

Note

정책 변수를 사용하려면 Version 요소를 문에 추가해야 하며, 이때 버전은 정책 변수를 지원하는 버전으로 설정해야 합니다. 변수는 버전 2012-10-17에서 도입되었습니다. 정책 언어의 조기 버전은 정책 변수를 지원하지 않기 때문입니다. Version 요소를 추가하지 않고 해당 버전 날짜로 설정하면 \${aws:username} 같은 변수가 정책에서 리터럴 문자열로 처리됩니다.
Version 정책 요소는 정책 버전과 다릅니다. Version 정책 요소는 정책 내에서 사용되며 정책 언어의 버전을 정의합니다. 반면에 정책 버전은 IAM에서 고객 관리형 정책을 변경할 때 생성됩니다. 변경된 정책은 기존 정책을 덮어쓰지 않습니다. 대신 IAM에서 관리형 정책의 새 버전을 만들습니다. Version 정책 요소에 대한 자세한 내용은 [the section called “Version” \(p. 499\)](#) 단원을 참조하십시오. 정책 버전에 대한 자세한 내용은 [the section called “IAM 정책 버전 관리” \(p. 399\)](#) 단원을 참조하십시오.

비슷한 방식으로 정책 변수를 사용하여 각 사용자가 자신의 액세스 키를 관리할 수 있도록 할 수 있습니다. 사용자가 프로그래밍 방식으로 David 사용자의 액세스 키를 변경할 수 있는 정책은 아래와 유사합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Action": ["iam:*AccessKey*"],  
        "Effect": "Allow",  
        "Resource": ["arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:user/David"]  
    }]  
}
```

이 정책이 david 사용자에게 추가되면 해당 사용자는 자신의 액세스 키를 변경할 수 있습니다. 사용자별 Amazon S3 객체에 대한 정책과 마찬가지로 사용자 이름을 포함하는 각 사용자에 대해 별도의 정책을 생성합니다. 그런 다음 각 정책을 개별 사용자에 연결합니다.

정책 변수를 사용하여 생성할 수 있는 정책은 다음과 같습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Action": ["iam:*AccessKey*"],  
        "Effect": "Allow",  
        "Resource": ["arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:user/${aws:username}"]  
    }]  
}
```

이처럼 사용자 이름에 정책 변수를 사용할 때는 개별 사용자마다 별도의 정책을 생성할 필요가 없습니다. 대신에 이 새로운 정책을 자신의 액세스 키를 직접 관리해야 하는 사용자가 모두 포함된 IAM 그룹에 추가하면 됩니다. 이후 사용자가 자신의 액세스 키 변경을 요청하면 IAM이 현재 요청의 사용자 이름을 \${aws:username} 변수에 치환한 후 정책을 평가합니다.

정책 변수로서의 태그

일부 AWS 서비스에서는 사용자 지정 속성을 해당 서비스가 생성한 리소스에 연결할 수 있습니다. 예를 들어, Amazon S3 버킷 또는 IAM 사용자 및 역할에 태그를 적용할 수 있습니다. 이러한 태그는 키-값 페어입니다. 태그 키 이름과 해당 키 이름과 연관된 값을 정의합니다. 예를 들어 **department** 키와 **Human Resources** 값으로 태그를 만들 수 있습니다. IAM 엔터티 태그 지정에 대한 자세한 내용은 [IAM 엔터티 태그 지정 \(p. 259\)](#) 단원을 참조하십시오. 다른 AWS 서비스에서 생성한 리소스에 대한 태그 지정 정보는 해당 서비스의 문서 단원을 참조하십시오. Tag Editor에 대한 자세한 내용은 AWS Management 콘솔 사용 설명서의 [Tag Editor 작업](#) 단원을 참조하십시오.

IAM 자격 증명에 태그를 추가하면 IAM 리소스를 쉽게 찾고, 구성하고, 추적할 수 있습니다. 또한 IAM 자격 증명에 태그를 지정하여 리소스에 대한 액세스를 제어하거나 자체 태그를 지정할 수 있습니다. 태그를 사용하여 액세스를 제어하는 방법에 대한 자세한 내용은 [IAM 태그를 사용한 액세스 제어 \(p. 336\)](#) 단원을 참조하십시오.

정책 변수를 사용할 수 있는 경우

정책 변수는 Resource 요소를 비롯해 Condition 요소의 문자열 비교에 사용할 수 있습니다.

리소스 요소

정책 변수는 리소스 식별자인 ARN의 마지막 부분에 표시됩니다. 다음은 그룹에 추가할 수 있는 정책입니다. 이 정책은 그룹 내 각 사용자에게 Amazon S3의 사용자별 객체(자신의 "홈 디렉터리")에 프로그래밍 방식으로 완전히 액세스할 수 있는 권한을 부여하고 있습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": ["s3>ListBucket"],  
            "Effect": "Allow",  
            "Resource": ["arn:aws:s3:::mybucket"],  
            "Condition": {"StringLike": {"s3:prefix": ["${aws:username}/*"]}}  
        },  
        {  
            "Action": [  
                "s3:GetObject",  
                "s3:PutObject"  
            ],  
            "Effect": "Allow",  
            "Resource": ["arn:aws:s3:::mybucket/*"]  
        }  
    ]  
}
```

```
        "Effect": "Allow",
        "Resource": ["arn:aws:s3:::mybucket/${aws:username}/*"]
    }
}
```

Note

위 예제는 `aws:username` 키를 사용하여 알기 쉬운 사용자 이름("Adele" 또는 "David" 등)을 반환합니다. 하지만 글로벌 고유 값인 `aws:userid` 키를 사용해야 하는 경우도 있습니다. 자세한 내용은 [고유 ID \(p. 483\)](#) 단원을 참조하십시오.

다음은 IAM 그룹에 사용할 수 있는 정책입니다. 이 정책에 따라 해당 그룹의 사용자들은 자신의 이름이 포함된 대기열과 us-east-2 리전에 속한 대기열을 생성, 사용 및 삭제할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListForConsole",
            "Effect": "Allow",
            "Action": "sns>ListQueues",
            "Resource": "*"
        },
        {
            "Sid": "AllQueueActions",
            "Effect": "Allow",
            "Action": "sns:*",
            "Resource": "arn:aws:sqs:us-east-2:*$aws:username}-queue"
        }
    ]
}
```

ARN의 일부를 태그 값으로 바꾸려면 접두사와 키 이름을 \${}로 둑습니다. 예를 들어 다음 Resource 요소는 요청한 사용자의 department 태그 값과 동일한 이름의 버킷만 참조합니다.

```
"Resource": ["arn:aws:s3:::bucket/${aws:PrincipalTag/department}"]
```

조건 요소

정책 변수는 문자열 연산자(Condition, StringEquals, StringLike 등) 또는 ARN 연산자(StringNotLike, ArnEquals 등)이 추가된 모든 조건에서 ArnLike 값으로도 사용할 수 있습니다. 다음 Amazon SNS 주제 정책은 AWS 계정 999999999999의 사용자들에게 URL이 AWS 사용자 이름과 일치하는 경우에만 이러한 주제를 관리할 수 있도록 권한(모든 작업 실행)을 부여합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Principal": {"AWS": "999999999999"},
            "Effect": "Allow",
            "Action": "sns:*",
            "Condition": {"StringLike": {"sns:endpoint": "https://example.com/${aws:username}/*"}}
        }
    ]
}
```

Condition 요소 표현식에서 태그를 참조할 때는 관련 접두사와 키 이름을 조건 키로 사용하십시오. 그런 다음 조건 값에서 테스트 할 값을 사용합니다. 예를 들어 다음 정책 예제에서는 costCenter 태그가 리소스에 연결된 경우에만 리소스에 액세스할 수 있습니다. 태그의 값은 12345 또는 67890이어야 합니다. 태그에 값이 없거나 다른 값이 있으면 요청이 실패합니다.

```
{  
    "Version": "2015-01-01",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "*",  
            "Resource": "*",  
            "Condition": {  
                "StringLike": {  
                    "iam:ResourceTag/costCenter": [ "12345", "67890" ]  
                }  
            }  
        }  
    ]  
}
```

정책 변수로 사용할 수 있는 요청 정보

정책 변수의 치환 값은 현재 [요청 컨텍스트 \(p. 532\)](#)에서 나와야 합니다.

모든 요청에 사용할 수 있는 정보

정책에는 정책 변수로 사용할 수 있는 값의 키가 포함됩니다. (일부 값이 포함되지 않는 키도 있습니다. 자세한 내용은 이번 목록 다음의 정보 단원을 참조하십시오).

- **aws:CurrentTime** 날짜와 시간을 확인하는 조건에 사용할 수 있습니다.
- **aws:EpochTime** epoch의 날짜 또는 Unix 시간으로, 날짜/시간 조건에 사용합니다.
- **aws:TokenIssueTime** 임시 보안 자격 증명이 발급된 날짜와 시간으로, 날짜/시간 조건에 사용할 수 있습니다. 참고: 이 키는 임시 보안 자격 증명을 사용해 서명된 요청에만 사용할 수 있습니다. 임시 보안 자격 증명에 대한 자세한 내용은 [임시 보안 자격 증명 \(p. 263\)](#) 단원을 참조하십시오.
- **aws:principaltype** 이 값은 계정, 사용자, 연동된 역할 또는 위임된 역할 등 보안 주체가 무엇인지를 나타냅니다. 뒤에 나오는 설명 단원을 참조하십시오.
- **aws:SecureTransport** 요청이 SSL을 사용하여 전송되었는지 여부를 나타내는 부울 값입니다.
- **aws:SourceIp** 요청자의 IP 주소로, IP 주소 조건에 사용합니다. 언제 [IP 주소 조건 연산자 \(p. 517\)](#)가 유효한지와 언제 VPC 전용 키를 대신 사용해야 하는지에 대한 정보는 SourceIp 단원을 참조하십시오.
- **aws:UserAgent** 이 값은 요청자의 클라이언트 애플리케이션에 대한 정보를 포함하는 문자열입니다. 이 문자열은 클라이언트에 의해 생성되며 신뢰성이 떨어질 수 있습니다. AWS CLI에서는 이 컨텍스트 키를 사용만 할 수 있습니다.
- **aws:userid** 이 값은 현재 사용자의 고유 ID입니다. 다음에 나오는 차트 단원을 참조하십시오.
- **aws:username** 현재 사용자의 [알기 쉬운 이름 \(p. 480\)](#)을 포함하는 문자열입니다. 다음에 나오는 차트 단원을 참조하십시오.
- **ec2:SourceInstanceARN** 요청이 이루어진 Amazon EC2 인스턴스의 Amazon 리소스 이름(ARN)입니다. 이 키는 EC2 인스턴스 프로필과 연결된 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 해당 요청이 들어오는 경우에만 존재합니다.

Important

키 이름은 대/소문자를 구분합니다. 예를 들어, `aws:CurrentTime`은 `AWS:currenttime`과 같습니다.

`aws:username`, `aws:userid` 및 `aws:principaltype` 값은 요청을 시작한 보안 주체 유형에 따라 달립니다. 예를 들어 요청은 AWS 계정, IAM 사용자, IAM 역할 등의 자격 증명을 사용하여 수행할 수 있습니다. 다음은 다른 유형의 보안 주체에 사용되는 키 값을 나타낸 목록입니다.

- AWS 계정

- aws:username: (없음)
- aws:userid: AWS 계정 ID
- aws:principaltype: Account
- IAM 사용자
 - aws:username: *IAM-user-name*
 - aws:userid: 고유 ID (p. 483)
 - aws:principaltype: User
- 연동 사용자
 - aws:username: (없음)
 - aws:userid: *account:caller-specified-name*
 - aws:principaltype: FederatedUser
- 웹 연동 사용자 및 SAML 연동 사용자

Note

웹 자격 증명 연동을 사용할 때 사용 가능한 정책 키에 대한 자세한 내용은 [웹 자격 증명 연동을 사용해 사용자 식별하기 \(p. 165\)](#) 단원을 참조하십시오.

- aws:username: (없음)
- aws:userid: (없음)
- aws:principaltype: AssumedRole
- 위임된 역할
 - aws:username: (없음)
 - aws:userid: *role-id:caller-specified-role-name*
 - aws:principaltype: Assumed role
- Amazon EC2 인스턴스에 할당된 역할
 - aws:username: (없음)
 - aws:userid: *role-id:ec2-instance-id*
 - aws:principaltype: Assumed role
- 익명 호출자(Amazon SQS Amazon SNS 및 Amazon S3)
 - aws:username: (없음)
 - aws:userid: (없음)
 - aws:principaltype: Anonymous

이 목록에 있는 항목의 경우 다음을 참고하십시오.

- 없음이란 현재 요청 정보에 값이 없다는 의미이며, 이때 일치시키려고 하면 실패하고 요청이 거부됩니다.
- *role-id*는 각 역할 생성 시 할당되는 고유 식별자입니다. 역할 ID는 AWS CLI 명령 `aws iam get-role --role-name rolename`으로 표시할 수 있습니다.
- *caller-specified-name* 및 *caller-specified-role-name*은 임시 자격 증명을 가져오기 위해 호출할 때 호출 프로세스(예: 애플리케이션 또는 서비스 등)에서 전달되는 이름입니다.
- *ec2-instance-id*는 실행 시 인스턴스에 할당되는 값으로서 Amazon EC2 콘솔의 인스턴스 페이지에 표시됩니다. 그 밖에 AWS CLI 명령 `aws ec2 describe-instances`를 실행해도 인스턴스 ID를 표시할 수 있습니다.

연동 사용자 요청에 사용할 수 있는 정보

연동 사용자란 IAM 외에 다른 시스템을 사용하여 인증된 사용자를 말합니다. 예를 들어 AWS 호출 시 자체적으로 애플리케이션을 사용하는 회사가 있다고 가정하겠습니다. 이때는 회사의 애플리케이션 사용자 모두에게 IAM 자격 증명을 제공하는 것이 현실적으로 어렵습니다. 대신에 단일 IAM 자격 증명을 갖춘 프록시(미

들 티어) 애플리케이션을 사용하거나, SAML 자격 증명 공급자(IdP)를 사용할 수 있습니다. 프록시 애플리케이션이나 SAML IdP는 회사 네트워크를 사용해 각 사용자를 인증합니다. 그런 다음 프록시 애플리케이션이 IAM 자격 증명을 사용하여 개별 사용자에 대한 임시 보안 자격 증명을 얻을 수 있습니다. SAML IdP는 AWS 임시 보안 자격 증명에 대한 ID 정보를 사실상 교환할 수 있습니다. 이후 임시 자격 증명을 사용하면 AWS 리소스에 액세스할 수 있습니다.

이와 유사한 방식으로 앱을 통해 AWS 리소스에 액세스해야 하는 모바일 디바이스용 앱을 개발하는 것도 가능합니다. 이런 경우에는 웹 자격 증명 연동을 사용할 수 있습니다. 웹 자격 증명 연동에서는 앱이 Login with Amazon, Amazon Cognito, Facebook 또는 Google처럼 잘 알려진 자격 증명 공급자를 통해 사용자를 인증합니다. 인증이 완료되면 앱이 공급자의 사용자 인증 정보를 사용하여 임시 보안 자격 증명을 가져온 후 AWS 리소스에 액세스합니다.

웹 자격 증명 연동을 위해 가장 바람직한 방법은 Amazon Cognito와 AWS 모바일 SDK를 이용하는 것입니다. 자세한 내용은 다음 단원을 참조하십시오.

- Android용 AWS Mobile SDK Developer Guide의 [Amazon Cognito 개요](#)
- AWS Mobile SDK for iOS Developer Guide의 [Amazon Cognito 개요](#)
- [임시 자격 증명과 관련된 일반적인 시나리오 \(p. 264\)](#).

서비스별 정보

요청에는 서비스에 따른 키와 값이 요청 컨텍스트에 추가될 수 있습니다. 예는 다음과 같습니다.

- `s3:prefix`
- `s3:max-keys`
- `s3:x-amz-acl`
- `sns:Endpoint`
- `sns:Protocol`

정책 변수 값을 가져오는 데 사용할 수 있는 서비스별 키에 대한 자세한 내용은 각 서비스 설명서 단원을 참조하십시오. 예를 들어 다음 주제를 참조하시면 됩니다.

- Amazon Simple Storage Service 개발자 가이드의 [버킷 정책 및 사용자 정책 사용](#).
- Amazon Simple Notification Service 개발자 안내서의 [Amazon SNS 키](#).

특수 문자

정책 변수 중에는 다른 특별한 의미를 갖는 문자를 나타낼 수 있도록 사전에 정의되어 있는 고정 값의 변수들도 몇 가지 있습니다. 이 특수 문자들은 일치시키려는 문자열의 일부이지만 리터럴로 삽입하였다며 오해할 가능성이 있습니다. 예를 들어 문자열에 별표(*)를 삽입하면 리터럴(*)이 아닌 모든 문자와 일치하는 와일드 카드로 해석될 수 있습니다. 이 경우에는 다음과 같이 사전에 정의된 정책 변수를 사용할 수 있습니다.

- `{*}` - 별표(*) 문자가 필요한 경우에 사용
- `{?}` - 물음표(?)가 필요한 경우에 사용
- `${}` - 달러 문자(\$)가 필요한 경우에 사용

위처럼 사전 정의된 정책 변수들은 정규 정책 변수를 사용할 수 있는 문자열이라면 어디든지 사용 가능합니다.

자세한 정보

정책에 대한 자세한 정보는 다음 단원을 참조하십시오.

- 정책 및 권한 (p. 305)
- IAM 자격 증명 기반 정책 예제 (p. 341)
- IAM JSON 정책 요소 참조 (p. 498)
- 정책 평가 로직 (p. 531)
- 웹 자격 증명 연동에 대하여 (p. 162)

IAM JSON 정책 요소: 지원되는 데이터 형식

이 단원에서는 JSON 정책에서 값을 지정할 때 지원되는 데이터 형식을 설명합니다. 정책 언어는 각 정책 요소마다 모든 형식을 지원하지 않기 때문에 각 요소에 대한 자세한 내용은 이전 단원을 참조하십시오.

- 문자열
- 숫자(정수 및 부동 소수점)
- 부울
- Null
- 목록
- 맵
- 구조(중첩 맵)

다음은 각 데이터 형식을 직렬화로 매핑한 표입니다. 모든 정책은 UTF-8 형식을 따라야 합니다. JSON 데이터 형식에 대한 자세한 내용은 [RFC 4627](#)에서 확인할 수 있습니다.

유형	JSON
문자열	문자열
정수	번호
부동 소수점	번호
부울	true false
Null	null
날짜	ISO 8601의 W3C 프로파일을 준수하는 문자열
IpAddress	RFC 4632를 준수하는 문자열
List	배열
Object	Object

정책 평가 로직

보안 주체가 AWS Management 콘솔, AWS API 또는 AWS CLI를 사용하려고 시도하면 해당 보안 주체가 요청을 AWS에 전송합니다. AWS 서비스가 요청을 받으면 AWS는 여러 단계를 완료하여 요청을 허용할지 거부할지 여부를 결정합니다.

1. 인증 – AWS는 먼저 필요하다면 요청을 생성하는 보안 주체를 인증합니다. 이 단계는 익명 사용자의 요청을 허용하는 Amazon S3와 같은 몇몇 서비스에서는 필요하지 않습니다.
2. 요청 콘텍스트 처리 (p. 532) – AWS는 요청에 담긴 내용을 처리하여 어떤 정책을 요청에 적용할지 결정합니다.

3. 단일 계정 내에서 정책 평가 (p. 532) – AWS는 정책의 평가 순서에 영향을 받는 모든 정책 유형을 평가 합니다.
4. 계정 내에서 요청 허용 여부 결정 (p. 534) – 이때 AWS는 요청에 따른 정책을 처리하여 요청을 허용할지 거부할지 여부를 결정합니다.

요청 콘텍스트 처리

AWS는 요청을 처리하여 다음 정보를 요청 콘텍스트에 모읍니다.

- 작업(또는 작동) – 보안 주체가 수행하고자 하는 작업 또는 작동입니다.
- 리소스 – 수행된 작업 또는 작동에 따른 AWS 리소스 객체입니다.
- 보안 주체 – 요청을 보내는 사용자, 역할, 연합된 사용자 또는 애플리케이션입니다. 보안 주체에 대한 정보는 보안 주체와 관련된 정책을 포함합니다.
- 환경 데이터 – IP 주소, 사용자 에이전트, SSL 사용 상태 또는 시간대와 같은 정보입니다.
- 리소스 데이터 – 요청되는 리소스와 관련된 데이터. 여기에는 DynamoDB 테이블 이름 또는 Amazon EC2 인스턴스 태그와 같은 정보가 포함될 수 있습니다.

AWS는 이러한 정보를 사용하여 요청 콘텍스트에 적용되는 정책을 찾습니다.

단일 계정 내에서 정책 평가

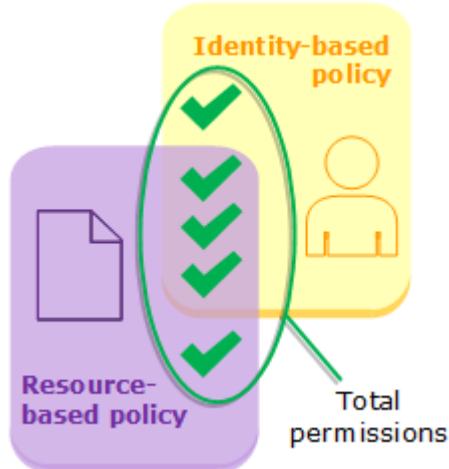
AWS는 요청 콘텍스트에 적용되는 정책 유형에 따라 정책을 평가합니다. 빈도 순으로 나열된 다음 정책 유형을 단일 AWS 계정 내에서 사용할 수 있습니다. 이러한 정책 유형에 대한 자세한 정보는 [정책 및 권한 \(p. 305\)](#)를 참조하십시오. 교차 계정 권한 부여에 대한 자세한 정보는 [IAM 역할과 리소스 기반 정책의 차이 \(p. 257\)](#) 단원을 참조하십시오.

1. 자격 증명 기반 정책 – 자격 증명 기반 정책은 IAM 자격 증명(사용자, 사용자 그룹 또는 역할)에 연결되어 IAM 엔터티(사용자 및 역할)에 권한을 부여합니다. 자격 증명 기반 정책만 요청에 적용되는 경우 AWS에서는 하나 이상의 Allow에 대해 이러한 정책을 모두 확인합니다.
2. 리소스 기반 정책 – 리소스 기반 정책을 통해 보안 주체로서 지정된 보안 주체 엔터티(계정, 사용자, 역할 또는 연합된 사용자)에 권한을 부여합니다. 권한은 보안 주체가 정책이 연결된 리소스를 사용하여 수행할 수 있는 작업을 정의합니다. 리소스 기반 정책 및 자격 증명 기반 정책 둘 다 요청에 적용되는 경우 AWS에서는 하나 이상의 Allow에 대해 이러한 정책을 모두 확인합니다.
3. IAM 권한 경계 – 권한 경계는 자격 증명 기반 정책을 통해 IAM 엔터티(사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 엔터티에 대한 권한 경계를 설정할 경우 해당 엔터티는 자격 증명 기반 정책 및 관련 권한 경계 모두에서 허용되는 작업만 수행할 수 있습니다. 권한 경계는 리소스 기반 정책을 통해 부여된 권한에 영향을 미치지 않습니다.
4. AWS Organizations 서비스 제어 정책(SCP) – 조직 SCP는 조직 또는 조직 단위(OU)에 대한 최대 권한을 지정합니다. SCP는 각 AWS 계정 루트 사용자를 비롯하여 멤버 계정의 엔터티에 최대 권한을 적용합니다. SCP가 있는 경우 자격 증명 기반 및 리소스 기반 정책은 이러한 정책과 SCP에서 해당 작업을 허용하는 경우에 한해서만 엔터티에 권한을 부여합니다. 권한 경계와 SCP가 둘 다 있는 경우 권한 경계, SCP 및 자격 증명 기반 정책 모두에서 해당 작업을 허용해야 합니다.
5. 세션 정책 – 세션 정책은 역할 또는 연합된 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 역할 세션을 프로그래밍 방식으로 생성하려면 `AssumeRole*` API 작업 중 하나를 사용합니다. 이 작업을 수행하고 세션 정책을 전달할 경우 결과적으로 세션에는 해당 역할의 사용자 자격 증명 기반 정책과 세션 정책, 이 두 정책 모두에 의해 부여되는 권한만 지정됩니다. 연합된 사용자 세션을 생성하려면 IAM 사용자의 액세스 키를 사용하여 `GetFederationToken` API 작업을 프로그래밍 방식으로 호출합니다. 이 작업을 수행하고 세션 정책을 전달할 경우 결과적으로 세션에는 IAM 사용자 자격 증명 기반 정책과 세션 정책, 이 두 정책 모두에 의해 부여되는 권한만 부여됩니다. 리소스 기반 정책은 리소스 기반 정책에서 보안 주체로 나열되는 것이 사용자/역할의 ARN인지 세션의 ARN인지에 따라 세션 정책 권한 평가에 서로 다른 영향을 미칩니다. 자세한 정보는 [세션 정책 \(p. 307\)](#) 단원을 참조하십시오.

이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의함을 명심하십시오.

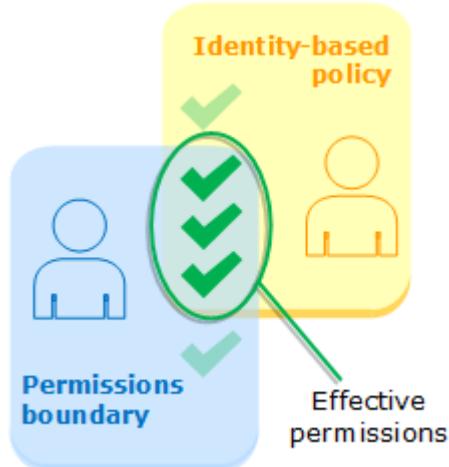
리소스 기반 정책과 함께 자격 증명 기반 정책 평가

자격 증명 기반 정책 및 리소스 기반 정책은 연결된 자격 증명이나 리소스에 권한을 부여합니다. IAM 엔터티(사용자 또는 역할)가 동일 계정 내에서 리소스에 대한 액세스를 요청할 경우 AWS는 자격 증명 기반 정책 및 리소스 기반 정책을 통해 부여된 모든 권한을 평가합니다. 결과적으로 두 정책 유형의 모든 권한이 권한으로 부여됩니다. 자격 증명 기반 정책, 리소스 기반 정책 또는 두 정책 모두에 의해 작업이 허용되는 경우 AWS에서는 해당 작업을 허용합니다. 이들 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다.



권한 경계와 함께 자격 증명 기반 정책 평가

AWS에서 사용자의 자격 증명 기반 정책 및 권한 경계를 평가하는 경우 결과적으로 두 범주의 공통된 권한만 권한으로 부여됩니다. 기존 자격 증명 기반 정책으로 사용자에 권한 경계를 추가하면 사용자가 수행할 수 있는 작업을 축소할 수 있습니다. 또는 사용자에게서 권한 경계를 제거하면 사용자가 수행할 수 있는 작업이 늘어날 수 있습니다. 이들 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 다른 정책 유형을 권한 경계와 함께 평가하는 방식에 대해 자세히 알아보려면 경계가 있는 효과적인 권한 평가 (p. 318) 단원을 참조하십시오.



조직 SCP와 함께 자격 증명 기반 정책 평가

AWS에서 사용자의 자격 증명 기반 정책과 사용자 계정이 속한 조직의 SCP를 평가하는 경우 결과적으로 두 범주의 공통된 권한만 권한으로 부여됩니다. 즉, 자격 증명 기반 정책 및 SCP 모두에서 작업이 허용되어야 합니다. 이들 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다.

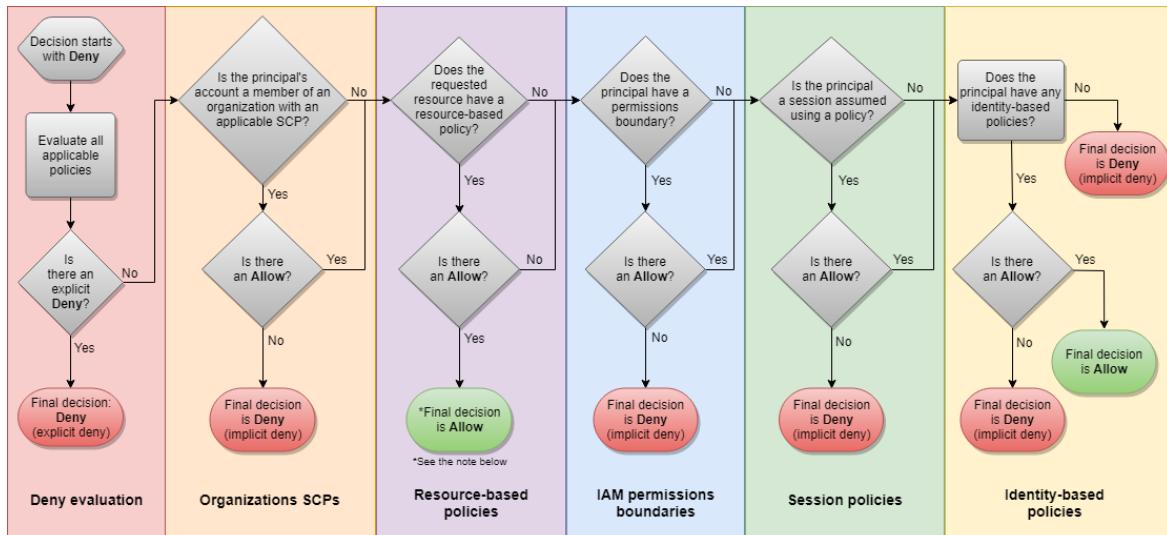


계정 내에서 요청 허용 여부 결정

보안 주체가 보안 주체의 엔터티와 동일한 계정 내 리소스에 대한 액세스 요청을 AWS에 보내는 경우 AWS 적용 코드는 해당 요청의 허용/거부 여부를 결정합니다. AWS는 요청 컨텍스트에 적용되는 모든 정책을 수집합니다. 다음은 단일 계정에 적용되는 이러한 정책에 대한 AWS 평가 로직을 간략하게 요약한 것입니다.

- 기본적으로 모든 요청이 농시적으로 거부됩니다. 또는 기본적으로 AWS 계정 루트 사용자에 모든 권한이 부여됩니다.
- 자격 증명 기반 또는 리소스 기반 정책에 포함된 명시적 허용은 이 기본 작동을 재정의합니다.
- 권한 경계, 조직 SCP 또는 세션 정책이 있는 경우 이러한 정책 유형이 명시적 거부로 허용을 재정의할 수도 있습니다.
- 어떠한 정책의 명시적 거부도 허용을 무시합니다.

다음 순서도에 결정 방법에 대한 세부 정보가 나와 있습니다.



- 거부 평가 – 기본적으로 모든 요청이 거부됩니다. 이를 [농시적 거부](#) (p. 537)라고 합니다. AWS 적용 코드는 해당 요청에 적용될 수 있는 계정 내의 모든 정책을 평가합니다. 여기에는 AWS Organizations SCP, 리소스 기반 정책, IAM 권한 경계, 역할 세션 정책 및 자격 증명 기반 정책이 포함됩니다. 이런 모든 정책에 적용 코드는 해당 요청에 적용되는 Deny 설명문을 찾습니다. 이를 [명시적 거부](#) (p. 537)라고 합니다.

적용되는 명시적 거부가 하나라도 발견되면 이 코드는 최종 거부 결정을 반환합니다. 명시적 거부가 없으면 코드 실행이 계속됩니다.

2. 조직 SCP – 그 다음에는 요청에 적용되는 AWS Organizations 서비스 제어 정책(SCP)을 평가합니다. SCP가 연결된 계정에서 요청된 경우 해당 SCP가 적용됩니다. 적용 가능한 Allow 문이 SCP에 없는 경우 요청이 무시적으로 거부됩니다. 적용 코드가 최종 거부 결정을 반환합니다. SCP가 없거나 요청한 작업이 SCP에서 허용된 경우 코드 실행이 계속됩니다.
3. 리소스 기반 정책 – 보안 주체 엔터티에 대해 요청한 작업의 수행을 허용하는 리소스 기반 정책이 요청한 리소스에 지정된 경우 적용 코드는 최종 허용 결정을 반환합니다. 리소스 기반 정책이 없거나 이 정책에 Allow 문이 포함되지 않은 경우 코드 실행이 계속됩니다.

Note

IAM 역할이나 사용자의 ARN을 리소스 기반 정책의 보안 주체로 지정한 이후에 다른 사람이 세션 정책을 사용하여 해당 역할이나 연합된 사용자에 대해 임시 세션을 생성하는 경우에는 이 로직이 다르게 작동합니다. 자세한 정보는 [세션 정책](#)을 참조하십시오.

4. IAM 권한 경계 – 다음에는 적용 코드가 보안 주체에 사용되는 IAM 엔터티에 권한 경계가 지정되어 있는지 여부를 확인합니다. 권한 경계를 설정하는데 사용되는 정책에서 요청한 작업을 허용하지 않는 경우 요청이 무시적으로 거부됩니다. 적용 코드가 최종 거부 결정을 반환합니다. 권한 경계가 없거나 요청한 작업이 권한 경계에서 허용된 경우 코드 실행이 계속됩니다.
5. 세션 정책 – 그 다음, 적용 코드는 세션 정책을 전달하여 보안 주체에서 위임된 세션을 사용 중인지 확인합니다. AWS CLI 또는 AWS API를 사용하는 동안 세션 정책을 전달하여 역할이나 연합된 사용자에 대한 임시 자격 증명을 가져올 수 있습니다. 세션 정책이 있지만 요청한 작업이 세션 정책에서 허용되지 않는 경우 해당 요청이 무시적으로 거부됩니다. 적용 코드가 최종 거부 결정을 반환합니다. 세션 정책이 없거나 요청한 작업이 세션 정책에서 허용된 경우 코드 실행이 계속됩니다.
6. 자격 증명 기반 정책 – 그 다음, 적용 코드는 보안 주체 엔터티에 대한 자격 증명 기반 정책을 확인합니다. IAM 사용자의 경우 이러한 정책에는 사용자 정책과 사용자가 속한 그룹의 정책이 포함됩니다. 적용 가능한 자격 증명 기반 정책에 요청한 작업을 허용하는 설명문이 있는 경우 적용 코드는 최종 허용 결정을 반환합니다. 요청한 작업을 허용하는 설명문이 없는 경우 해당 요청이 무시적으로 거부되고, 적용 코드는 최종 거부 결정을 반환합니다.
7. 오류 – AWS 적용 코드를 평가하는 도중 오류가 발생할 경우 코드는 예외를 생성한 후 닫힙니다.

자격 증명 기반 정책 및 리소스 기반 정책 평가 예제

가장 일반적인 정책 유형은 자격 증명 정책 및 리소스 기반 정책입니다.

Carlos가 carlossalazar라는 사용자 이름을 쓰고 있고 carlossalazar-logs Amazon S3 버킷에 파일을 저장하고자 한다고 가정하십시오.

또한 다음 정책이 carlossalazar IAM 사용자와 연결되었다고 가정하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowS3ListRead",  
            "Effect": "Allow",  
            "Action": [  
                "s3>ListAllMyBuckets",  
                "s3:HeadBucket"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Sid": "AllowS3Self",  
            "Effect": "Allow",  
            "Action": "s3:*",  
            "Resource": "arn:aws:s3:::carlossalazar-logs"  
        }  
    ]  
}
```

```
    "Resource": [
        "arn:aws:s3::::carlossalazar/*",
        "arn:aws:s3::::carlossalazar"
    ],
},
{
    "Sid": "DenyS3Logs",
    "Effect": "Deny",
    "Action": "s3:*",
    "Resource": [
        "arn:aws:s3::::*log*",
        "arn:aws:s3::::*log*/"
    ]
}
]
```

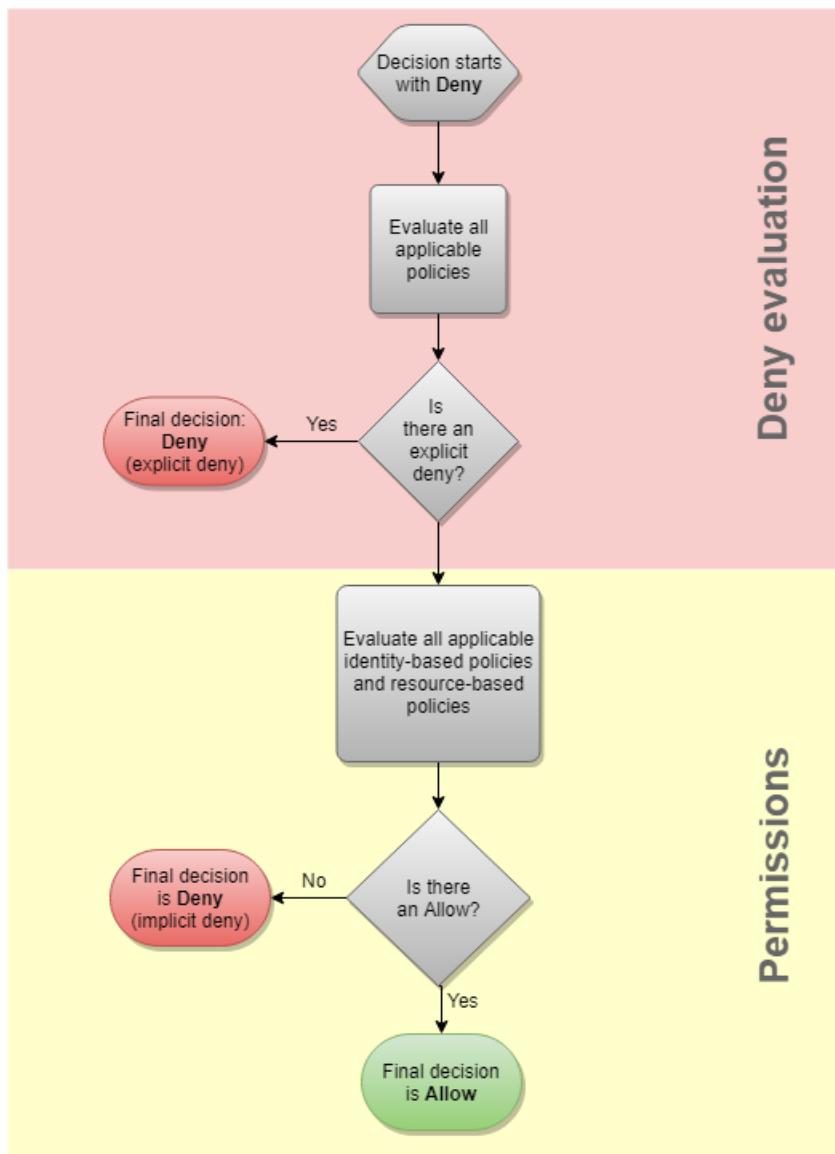
이 정책의 AllowS3ListRead 설명문은 카를로스가 계정에 있는 모든 버킷 목록을 보도록 허용합니다. AllowS3Self 설명문은 카를로스가 그의 사용자 이름과 동일한 버킷에 모두 액세스할 수 있도록 허용합니다. DenyS3Logs 설명문은 카를로스가 그의 이름 아래에 있는 log를 통해 모든 S3 버킷의 액세스를 거부합니다.

또한, 다음 리소스 기반 정책(버킷 정책이라고 함)은 carlossalazar 버킷에 연결됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Principal": { "AWS": "arn:aws:iam::111122223333:user/carlossalazar" },
            "Resource": "*"
        }
    ]
}
```

이 정책은 carlossalazar 사용자만 carlossalazar 버킷에 액세스할 수 있도록 지정합니다.

Carlos가 carlossalazar-logs 버킷에 파일을 저장하도록 요청하면 AWS는 해당 요청에 어떤 정책을 적용할지 결정합니다. 이 경우, 자격 증명 기반 정책과 리소스 기반 정책만 적용합니다. 이들은 모두 권한 정책입니다. 어떠한 권한 경계도 적용되지 않기 때문에 평가 로직은 다음 로직으로 줄어듭니다.



AWS는 먼저 요청 콘텍스트에 적용되는 Deny 설명문을 확인합니다. 자격 증명 기반 정책은 카를로스의 로깅을 통한 모든 S3 버킷의 액세스를 명시적으로 거부하기 때문에 이를 찾습니다. 카를로스의 액세스가 거부됩니다.

Carlos가 실수를 알아차리고 carlossalazar 버킷에 파일을 저장하고자 한다고 가정하십시오. AWS는 Deny 설명문을 확인하지만 찾지 못합니다. 그러면 권한 정책을 확인합니다. 자격 증명 기반 정책은 요청을 허용합니다. 따라서 AWS는 요청을 허용합니다. 이를 중 하나라도 설명문을 명시적으로 거부한다면 요청은 거부됩니다.

명시적 거부와 묵시적 거부 차이

적용 가능한 정책이 Deny 설명문을 포함한다면 요청은 명시적으로 거부됩니다. 정책이 Allow 설명문과 Deny 설명문을 포함한 요청에 적용된다면 Deny 설명문은 Allow 설명문에 우선합니다. 이 요청은 명시적으로 거부됩니다.

적용 가능한 Deny 설명문이 없고 적용 가능한 Allow 설명문도 없다면 뮤시적 거부가 발생합니다. IAM 사용자, 역할 또는 연합된 사용자가 기본적으로 액세스를 거부하기 때문에 명시적으로 작업을 허용해야 합니다. 그렇지 않으면 액세스는 뮤시적으로 거부됩니다.

권한 부여 전략을 설계한다면 Allow 설명문으로 정책을 생성하여 보안 주체가 성공적으로 요청하도록 허용합니다. 그러나 명시적 또는 뮤시적 거부 조합을 선택할 수 있습니다. 예를 들어, 다음 정책을 생성하여 AWS의 모든 리소스에 관리자가 완전히 액세스하도록 허용하지만 결제 액세스를 명시적으로 거부할 수 있습니다. 다른 사람이 관리자에게 다른 정책을 추가하여 결제를 허용하려고 해도 이 명시적 거부 때문에 결제는 여전히 거부됩니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "*",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Deny",  
            "Action": "aws-portal:*",  
            "Resource": "*"  
        }  
    ]  
}
```

또한, 다음 정책을 생성하여 그룹 또는 IAM의 기타 리소스가 아닌 사용자가 사용자를 관리할 수 있도록 허용합니다. 이러한 작업은 기타 서비스의 작업처럼 뮤시적으로 거부됩니다. 그러나 다른 사람이 정책을 이런 다른 작업을 수행하도록 허용하는 사용자에게 추가한다면 이런 작업이 허용됩니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": [  
            "iam:AttachUserPolicy",  
            "iam>CreateUser",  
            "iam>DeleteUser",  
            "iam>DeleteUserPolicy",  
            "iam:DetachUserPolicy",  
            "iam:GetUser",  
            "iam: GetUserPolicy",  
            "iam>ListAttachedUserPolicies",  
            "iam>ListUserPolicies",  
            "iam>ListUsers",  
            "iam:PutUserPolicy",  
            "iam:UpdateUser"  
        ],  
        "Resource": "*"  
    }  
}
```

IAM JSON 정책 언어의 문법

이 페이지에서는 IAM에서 JSON 정책 생성 시 사용되는 언어의 정규 문법에 대해 살펴보겠습니다. 이 문법에 대해 살펴본 후 정책의 체계적 작성 및 검증 방법에 대해 이해할 수 있게 될 것입니다.

정책 예는 다음 주제를 참조하십시오.

- 정책 및 권한 (p. 305)

- IAM 자격 증명 기반 정책 예제 (p. 341)
- Linux 인스턴스용 Amazon EC2 사용 설명서의 [Amazon EC2 콘솔 작업을 위한 예제 정책 및 AWS CLI, Amazon EC2 CLI 또는 AWS SDK 작업을 위한 예제 정책](#)
- Amazon Simple Storage Service 개발자 가이드의 [버킷 정책 예제 및 사용자 정책 예제](#)

다른 AWS 서비스의 정책 예는 해당 서비스 설명서를 참조하십시오.

주제

- 정책 언어 및 JSON (p. 539)
- JSON 문법에 사용되는 규칙 (p. 539)
- 문법 (p. 540)
- 정책 문법 참고 사항 (p. 541)

정책 언어 및 JSON

정책은 JSON으로 작성됩니다. 정책을 IAM에 제출하면 먼저 검증을 통해 JSON 구문의 정확성을 확인합니다. 여기에서는 유효한 JSON 구성에 대해 자세히 설명하지는 않지만 다음과 같이 몇 가지 기본 JSON 규칙을 소개합니다.

- 각 개체 간에 공백을 넣을 수 있습니다.
- 값은 인용 부호로 묶입니다. 숫자나 부울(Boolean) 값에서 인용 부호는 옵션입니다.
- 대부분 요소(예: `action_string_list`, `resource_string_list`)는 JSON 배열을 값으로 사용할 수 있습니다. 배열은 하나 이상의 값을 갖습니다. 값이 2개 이상 추가되면 배열은 다음 예제와 같이 대괄호([및])로 묶여 쉼표로 구분됩니다.

`"Action" : ["ec2:Describe*", "ec2>List*"]`
- 기본 JSON 데이터 형식(부울, 숫자, 문자열)은 [RFC 7159](#)에 정의되어 있습니다.

정책 구문은 JSON 검증기를 사용해 검사합니다. 검증기는 온라인에서 찾아볼 수 있으며, 그 밖에 JSON 검증 기능을 지원하는 코드 편집기나 XML 편집 도구도 많습니다.

JSON 문법에 사용되는 규칙

JSON 문법에는 다음과 같은 규칙이 사용됩니다.

- 다음 문자는 JSON 토큰으로서 정책에 추가됩니다.

{ } [] " , :

- 다음은 문법에 사용되는 특수 문자로서 정책에는 추가되지 않습니다.

= < > () |

- 한 요소에 여러 값을 추가할 수 있는 경우에는 반복되는 값, 쉼표 구분자, 그리고 줄임표(...)를 사용하여 나타냅니다. 예:

[<action_string>, <action_string>, ...]

<principal_map> = { <principal_map_entry>, <principal_map_entry>, ... }

여러 값이 허용되면 단일 값을 추가하는 것도 유효합니다. 값이 단 하나인 경우에는 마지막 쉼표를 반드시 생략해야 합니다. 요소가 배열([])로 표시)로 이루어지더라도 추가된 값이 단 하나일 때는 괄호가 선택 사항입니다. 예:

`"Action": [<action_string>]`

"Action": <action_string>

- 요소 뒤에 나오는 물음표(?)는 요소가 선택 사항인 것을 나타냅니다. 예:

<version_block?>

하지만 선택 요소에 대한 자세한 내용은 문법 목록 이후에 나오는 참고 사항을 반드시 확인하시기 바랍니다.

- 요소 사이의 수직선(|)은 다자간 텍일을 나타냅니다. 이 문법에서는 팔호로 다자간 텍일의 범위를 정의합니다. 예:

("Principal" | "NotPrincipal")

- 리터럴 문자열 요소는 큰따옴표(")로 묶습니다. 예:

<version_block> = "Version" : ("2008-10-17" | "2012-10-17")

기타 참고 사항은 문법 목록 다음 정책 문법 참고 사항 ([p. 541](#)) 단원을 참조하십시오.

문법

다음 목록은 정책 언어 문법에 대한 설명입니다. 문법 목록에 사용된 규칙에 대해서는 앞의 단원을 참조하십시오. 그리고, 추가 정보는 이후 참고 사항을 참조하십시오.

Note

이 문법은 버전이 2008-10-17 및 2012-10-17이라고 표시된 정책에 대한 설명입니다. Version 정책 요소는 정책 버전과 다릅니다. Version 정책 요소는 정책 내에서 사용되며 정책 언어의 버전을 정의합니다. 반면에 정책 버전은 IAM에서 고객 관리형 정책을 변경할 때 생성됩니다. 변경된 정책은 기존 정책을 덮어쓰지 않습니다. 대신 IAM에서 관리형 정책의 새 버전을 만듭니다. Version 정책 요소에 대한 자세한 내용은 [IAM JSON 정책 요소: Version \(p. 499\)](#)을 참조하십시오. 정책 버전에 대한 자세한 내용은 [the section called “IAM 정책 버전 관리” \(p. 399\)](#) 단원을 참조하십시오.

```
policy  = {
    <version_block?>
    <id_block?>
    <statement_block>
}

<version_block> = "Version" : ("2008-10-17" | "2012-10-17")

<id_block> = "Id" : <policy_id_string>

<statement_block> = "Statement" : [ <statement>, <statement>, ... ]

<statement> = {
    <sid_block?>,
    <principal_block?>,
    <effect_block>,
    <action_block>,
    <resource_block>,
    <condition_block?>
}

<sid_block> = "Sid" : <sid_string>

<effect_block> = "Effect" : ("Allow" | "Deny")

<principal_block> = ("Principal" | "NotPrincipal") : ("*" | <principal_map>)

<principal_map> = { <principal_map_entry>, <principal_map_entry>, ... }
```

```
<principal_map_entry> = ("AWS" | "Federated" | "Service") :  
    [<principal_id_string>, <principal_id_string>, ...]  
  
<action_block> = ("Action" | "NotAction") :  
    ("*" | [<action_string>, <action_string>, ...])  
  
<resource_block> = ("Resource" | "NotResource") :  
    ("*" | [<resource_string>, <resource_string>, ...])  
  
<condition_block> = "Condition" : { <condition_map> }  
<condition_map> {  
    <condition_type_string> : { <condition_key_string> : <condition_value_list> },  
    <condition_type_string> : { <condition_key_string> : <condition_value_list> }, ...  
}  
<condition_value_list> = [<condition_value>, <condition_value>, ...]  
<condition_value> = ("string" | "number" | "Boolean")
```

정책 문법 참고 사항

- 단일 정책에는 다수의 문이 배열로 추가될 수 있습니다.
- 정책은 추가되는 개체에 따라 2,048~10,240 사이에서 최대 문자 수를 갖습니다. 자세한 내용은 [IAM 개체 및 객체에 대한 제한 \(p. 485\)](#) 단원을 참조하십시오. 정책 크기 계산에 공백 문자는 포함되지 않습니다.
- 개별 요소에는 동일한 키 인스턴스를 여러 개 추가할 수 없습니다. 예를 들어 동일한 문에 Effect 블록을 2개 추가할 수는 없습니다.
- 블록은 순서에 상관없이 표시됩니다. 예를 들어 정책에서 version_block은 id_block 뒤에 올 수 있습니다. 마찬가지로 effect_block, principal_block 및 action_block 역시 동일 문에서 순서에 상관없이 표시됩니다.
- 리소스 기반 정책에서는 id_block이 선택 사항입니다. ID 기반 정책에는 포함 시킬 수 없습니다.
- principal_block 요소는 리소스 기반 정책(예: Amazon S3 버킷 정책)과 IAM 역할의 신뢰 정책에 필요합니다. ID 기반 정책에는 포함 시킬 수 없습니다.
- 각 문자열 값(policy_id_string, sid_string, principal_id_string, action_string, resource_string, condition_type_string, condition_key_string, 그리고 condition_value의 문자열 버전)은 자체적인 최소/최대 길이 제한, 특정 허용 값 또는 필수 내부 포맷을 가질 수 있습니다.

문자열 값에 대한 참고 사항

이 섹션에서는 정책에서 각각 다른 요소에 사용되는 문자열 값에 대한 추가 정보에 대해 살펴보겠습니다.

action_string

서비스 네임스페이스, 콜론 및 작업 이름으로 구성됩니다. 작업 이름에는 와일드카드를 추가할 수 있습니다. 예:

```
"Action":"ec2:StartInstances"  
  
"Action": [  
    "ec2:StartInstances",  
    "ec2:StopInstances"  
]  
  
"Action":"cloudformation:*"  
  
"Action":"*"  
  
"Action": [
```

```
"s3:Get*",
"s3>List*"
]
```

policy_id_string

정책 관련 정보를 전체적으로 추가하는 방법을 제공합니다. Amazon SQS나 Amazon SNS 같은 일부 서비스는 Id 요소를 예약 방식으로 사용합니다. 개별 서비스에서 달리 제한하지 않는다면 policy_id_string에 공백을 추가할 수 있습니다. AWS 계정 내에서 이 값의 고유성을 요구하는 서비스도 있습니다.

Note

id_block은 리소스 기반 정책에서는 허용되지만 ID 기반 정책에서는 사용할 수 없습니다.

이 문자열이 제한된 전체 정책 길이에 영향을 끼치기는 하지만 문자열 길이에 제한은 없습니다.

```
"Id":"Admin_Policy"
"Id":"cd3ad3d9-2776-4ef1-a904-4c229d1642ee"
```

sid_string

개별 문에 대한 정보를 추가하는 방법을 제공합니다. IAM 정책의 경우 기본 영숫자 문자(A-Z,a-z,0-9)만 Sid 값의 문자로 허용됩니다. 리소스 정책을 지원하는 다른 AWS 서비스는 sid 값 요구 사항이 다를 수 있습니다. 예를 들어 일부 서비스는 이 값이 특정 AWS 계정에서 고유할 것을 요구하며, 일부 서비스는 Sid 값으로 공백과 같은 문자를 추가로 허용합니다.

```
"Sid":"1"
"Sid": "ThisStatementProvidesPermissionsForConsoleAccess"
```

principal_id_string

AWS 계정, IAM 사용자, IAM 역할, 연동 사용자 또는 위임된 역할 사용자의 [Amazon 리소스 이름\(ARN\)](#) (p. 480)을 사용해 보안 주체를 지정하는 방법을 제공합니다. AWS 계정의 경우, 전체 ARN 대신 짧은 형식인 [AWS:accountnumber](#)를 사용할 수도 있습니다. AWS 서비스, 위임된 역할 등을 포함한 모든 옵션에 대해서는 [보안 주체 지정](#) (p. 501) 단원을 참조하십시오.

"모든 사용자/익명 사용자"를 지정할 때만 *를 사용할 수 있습니다. 이름이나 ARN의 일부를 지정하기 위해 사용할 수는 없습니다.

resource_string

대부분의 경우 [Amazon 리소스 이름\(ARN\)](#)으로 구성됩니다.

```
"Resource": "arn:aws:iam::123456789012:user/Bob"
"Resource": "arn:aws:s3:::examplebucket/*"
```

condition_type_string

StringEquals, StringLike, NumericLessThan, DateGreaterThanOrEqual, Bool, BinaryEquals, IpAddress, ArnEquals 등 테스트 할 조건 형식을 식별합니다. 조건 형식에 대한 전체 목록은 [IAM JSON: 정책 요소: 조건 연산자](#) (p. 513) 단원을 참조하십시오.

```
"Condition": {
    "NumericLessThan": {
        "s3:max-keys": "10"
    }
}
```

```
        }
    }

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

    "Condition": {
        "StringEquals": {
            "s3:x-amz-server-side-encryption": "AES256"
        }
    }
}
```

condition_key_string

값을 테스트하여 조건 총족 여부를 판단할 수 있는 조건 키를 식별합니다. AWS는 `aws:principaltype`, `aws:SecureTransport` 및 `aws:userid`를 포함하여 모든 AWS 서비스에서 사용할 수 있는 조건 키 집합을 정의합니다.

AWS 조건 키 목록에 대한 자세한 내용은 [AWS 전역 조건 컨텍스트 키 \(p. 551\)](#) 단원을 참조하십시오. 서비스별 조건 키에 대한 자세한 내용은 다음과 같은 서비스 설명서를 참조하십시오.

- Amazon Simple Storage Service 개발자 가이드의 [정책에서 조건 지정](#)
- Linux 인스턴스용 Amazon EC2 사용 설명서의 [Amazon EC2에 대한 IAM 정책](#).

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

"Condition": {
    "StringNotEquals": {
        "s3:x-amz-server-side-encryption": "AES256"
    }
}

"Condition": {
    "StringEquals": {
        "ec2:ResourceTag/purpose": "test"
    }
}
```

직무 기능에 대한 AWS 관리형 정책

직무에 관한 AWS 관리형 정책은 IT 업계의 일반적인 직무 기능과 긴밀하게 연결되도록 구성됩니다. 이 정책을 적용하면 특정 직무 담당자에게 기대되는 작업 수행에 필요한 권한을 쉽게 부여할 수 있습니다. 이 정책은 여러 서비스에 대한 권한을 정책 하나에 통합하기 때문에, 여러 정책에 권한이 분산되어 있는 경우보다 업무 절차가 간소합니다.

직무 기능에 관한 이 정책은 모든 그룹, 사용자 또는 역할에 연결할 수 있습니다.

역할을 이용한 서비스 결합

일부 정책은 IAM 서비스 역할을 이용하여 다른 AWS 서비스에 포함된 기능을 활용할 수 있도록 지원합니다. 이 정책은 `iam:passrole`에 대한 액세스를 허용하여, 정책에 정의된 사용자가 역할을 AWS 서비스에 전달할 수 있도록 합니다. 이 역할은 AWS 서비스에서 사용자를 대행할 수 있도록 IAM 권한을 위임합니다.

필요에 따라 역할을 만들어야 합니다. 예를 들어 네트워크 관리자 정책의 경우, 정책에 정의된 사용자가 "flow-logs-vpc"라는 역할을 Amazon CloudWatch 서비스로 전달하도록 허용합니다. CloudWatch는 이 역할을 이용해 사용자가 생성한 VPC의 IP 트래픽을 기록하고 캡처합니다.

보안 모범 사례를 따르기 위해 직무 기능에 관한 정책에는 전달할 수 있는 유효한 역할의 이름을 제한하는 필터가 포함되어 있습니다. 따라서 불필요한 권한을 부여할 가능성이 없습니다. 사용자가 선택적 서비스 역할을 필요로 할 경우, 정책에 정의된 명명 규칙에 따라 역할을 만들어야 합니다. 그런 다음 해당 역할에 권한을 부여합니다. 그러면 사용자는 역할이 제공하는 모든 권한을 부여해 서비스에서 이 역할을 사용하도록 구성할 수 있습니다.

최신 정보 유지

이러한 정책은 모두 AWS가 유지하며, AWS에서 정책을 추가할 때 새로운 서비스와 새로운 기능에 대한 지원을 포함시켜 모든 것을 최신 상태로 유지합니다. 이러한 정책은 고객이 수정할 수 없습니다. 정책 사본을 만든 후 이를 수정할 수 있으나 AWS가 새로운 서비스와 API 작업을 도입할 때 이 사본이 자동으로 업데이트되지는 않습니다.

직무 기능

정책 이름

- [Administrator \(p. 544\)](#)
- [결제 \(p. 544\)](#)
- [데이터베이스 관리자 \(p. 545\)](#)
- [데이터 과학자 \(p. 546\)](#)
- [개발자 파워 유저 \(p. 546\)](#)
- [네트워크 관리자 \(p. 547\)](#)
- [보안 감사 \(p. 547\)](#)
- [지원 사용자 \(p. 547\)](#)
- [시스템 관리자 \(p. 547\)](#)
- [보기 전용 사용자 \(p. 548\)](#)

다음 섹션에서 각 정책의 이름에는 AWS Management 콘솔의 정책 세부 정보 페이지로 이동하는 링크가 연결되어 있습니다. 해당 페이지에서 정책 문서를 확인하고 부여된 권한을 검토할 수 있습니다.

Administrator

AWS 관리형 정책 이름: [AdministratorAccess](#)

사용 사례: 이 사용자는 모든 액세스를 가지며 AWS 내 모든 서비스와 리소스에 권한을 위임할 수 있습니다.

정책 설명: 이 정책은 모든 AWS 서비스와 계정 내 모든 리소스에 대한 모든 작업을 허용합니다.

Note

IAM 사용자 또는 역할이 이 정책의 권한을 통해 AWS Billing and Cost Management 콘솔에 액세스할 수 있으려면, 먼저 IAM 사용자 및 역할 액세스를 활성화해야 합니다. 이를 위해 [결제 콘솔에 액세스를 위임하기 위한 자습서 1단계 \(p. 25\)](#)의 지침을 따르십시오.

결제

AWS 관리형 정책 이름: [Billing](#)

사용 사례: 이 사용자는 결제 정보를 확인하고 지불을 설정 및 승인해야 합니다. 사용자가 전체 AWS 서비스에 누적된 비용을 모니터링할 수 있습니다.

정책 설명: 이 정책은 결제 관리, 비용, 결제 방법, 예산, 보고서 등에 필요한 모든 권한을 부여합니다.

Note

IAM 사용자 또는 역할이 이 정책의 권한을 통해 AWS Billing and Cost Management 콘솔에 액세스 할 수 있으려면, 먼저 IAM 사용자 및 역할 액세스를 활성화해야 합니다. 이를 위해 [결제 콘솔에 액세스를 위임하기 위한 자습서 1단계 \(p. 25\)](#)의 지침을 따르십시오.

데이터베이스 관리자

AWS 관리형 정책 이름: [DatabaseAdministrator](#)

사용 사례: 이 사용자는 AWS 클라우드에서 데이터베이스를 설정, 구성, 유지합니다.

정책 설명: 이 정책은 데이터베이스를 생성, 구성, 유지할 수 있는 권한을 부여합니다. 여기에는 Amazon DynamoDB, Amazon Relational Database Service(RDS) 및 Amazon Redshift 등 AWS 데이터베이스 서비스에 대한 액세스가 포함됩니다. 이 정책이 지원하는 데이터베이스 서비스의 전체 목록에 대한 정책을 봅니다.

이 직무 정책은 AWS 서비스로 역할을 전달할 수 있는 기능을 지원합니다. 이 정책은 다음 표에 명시된 역할에만 `iam:PassRole` 작업을 허용합니다. 자세한 정보는 이 주제의 후반부에서 [역할 생성 및 정책 연결\(콘솔\) \(p. 548\)](#) 단원을 참조하십시오.

데이터베이스 관리자 직무에 대한 선택적 IAM 서비스 역할

사용 사례	역할 이름(*는 와일드카드)	선택할 서비스 역할 유형	이 AWS 관리형 정책 선택
사용자가 RDS 데이터베이스를 모니터링하도록 허용	rds-monitoring-role	Enhanced Monitoring을 위한 Amazon RDS 역할	AmazonRDSEnhancedMonitoringRole
AWS Lambda의 데이터베이스 모니터링과 외부 데이터베이스 액세스를 허용	rdbms-lambda-access	Amazon EC2	AWSLambdaFullAccess
Lambda가 DynamoDB를 이용해 Amazon S3와 Amazon Redshift 클러스터에 파일을 업로드하도록 허용	lambda_exec_role	AWS Lambda	AWS 빅 데이터 블로그에 정의된 대로 새로운 관리형 정책 구성
Lambda 기능이 DynamoDB 테이블의 트리거 역할을 하도록 허용	lambda-dynamodb-*	AWS Lambda	AWSLambdaDynamoDBExecutionRole
Lambda 기능이 VPC에서 Amazon RDS에 액세스하도록 허용	lambda-vpc-execution-role	AWS Lambda Developer Guide에 정의된 대로 신뢰 정책으로 역할 생성	AWSLambdaVPCAccessExecutionRole
AWS Data Pipeline가 AWS 리소스에 액세스하도록 허용	DataPipelineDefaultRole	AWS Data Pipeline 개발자 안내서에 정의된 대로 신뢰 정책으로 역할 생성	AWSDataPipelineRole
Amazon EC2 인스턴스에서 실행 중인 애플리케이션이 AWS 리소스에 액세스하도록 허용	DataPipelineDefaultResourceRole	AWS Data Pipeline 개발자 안내서에 정의된 대로 신뢰 정책으로 역할 생성	AmazonEC2RoleforDataPipelineRole

데이터 과학자

AWS 관리형 정책 이름: [DataScientist](#)

사용 사례: 이 사용자는 하둡 작업과 쿼리를 실행합니다. 또한 데이터 분석 및 비즈니스 인텔리전스에 관한 정보에 액세스하고 이를 분석합니다.

정책 설명: 이 정책은 Amazon EMR 클러스터에서 쿼리를 생성, 관리, 실행하고 Amazon QuickSight 같은 도구로 데이터 분석을 수행할 수 있는 권한을 부여합니다. 정책에는 AWS Data Pipeline, Amazon EC2, Amazon Kinesis, Amazon Machine Learning 및 Amazon SageMaker 등 추가 데이터 과학자 서비스에 대한 액세스가 포함됩니다. 이 정책이 지원하는 데이터 과학자 서비스의 전체 목록에 대한 정책을 봅니다.

이 직무 정책은 AWS 서비스로 역할을 전달할 수 있는 기능을 지원합니다. 한 개의 문이 역할을 Amazon SageMaker에 전달하도록 허용합니다. 또 다른 문은 다음 표에 명시된 역할에만 `iam:PassRole` 작업을 허용합니다. 자세한 정보는 이 주제의 후반부에서 [역할 생성 및 정책 연결\(콘솔\) \(p. 548\)](#) 단원을 참조하십시오.

데이터 과학자 직무에 대한 선택적 IAM 서비스 역할

사용 사례	역할 이름(*는 와일드 카드)	선택할 서비스 역할 유형	선택할 AWS 관리형 정책
클러스터에 적합한 서비스와 리소스에 대한 Amazon EC2 인스턴스 액세스를 허용	EMR-EC2_DefaultRole	EC2의 경우 Amazon EMR	AmazonElasticMapReduceforEC2Role
클러스터의 Amazon EC2 서비스와 리소스에 액세스 할 수 있는 Amazon EMR 액세스를 허용	EMR_DefaultRole	Amazon EMR	AmazonElasticMapReduceRole
Kinesis Kinesis Data Analytics가 스트리밍 데이터 소스에 액세스하도록 허용	kinesis-*	AWS 빅 데이터 블로그에 정의된 대로 신뢰 정책으로 역할을 생성합니다.	사용 사례에 따라 네 가지 가능한 옵션을 소개한 AWS 빅 데이터 블로그 참조
AWS Data Pipeline가 AWS 리소스에 액세스하도록 허용	DataPipelineDefaultRole	AWS Data Pipeline 개발자 안내서에 정의된 대로 신뢰 정책으로 역할 생성	AWSDataPipelineRole
Amazon EC2 인스턴스에서 실행 중인 애플리케이션이 AWS 리소스에 액세스하도록 허용	DataPipelineDefaultRole	AWS Data Pipeline 개발자 안내서에 정의된 대로 신뢰 정책으로 역할 생성	AmazonEC2RoleforDataPipelineRole

개발자 파워 유저

AWS 관리형 정책 이름: [PowerUserAccess](#)

사용 사례: 이 사용자는 애플리케이션 개발 작업을 수행하며, AWS 인식 애플리케이션 개발을 지원하는 리소스와 서비스를 생성하고 구성할 수 있습니다.

정책 설명: 이 정책의 첫 번째 설명문은 [NotAction \(p. 507\)](#) 요소를 사용하여 모든 AWS 서비스와 모든 리소스(AWS Identity and Access Management 및 AWS Organizations 제외)에 대해 모든 작업을 허용합니다. 두 번째 설명문은 서비스에 연결된 역할을 생성할 수 있는 IAM 권한을 부여합니다. 이것은 Amazon S3 버킷처럼 다른 서비스에서 리소스에 액세스해야 하는 서비스에 필요합니다. 또한 조직에게 마스터 계정 이메일과 조직 한도 등 사용자 조직에 대한 정보를 볼 권한을 부여합니다.

네트워크 관리자

AWS 관리형 정책 이름: [NetworkAdministrator](#)

사용 사례: 이 사용자는 AWS 네트워크 리소스를 설정하고 유지하는 작업을 담당합니다.

정책 설명: 이 정책은 Auto Scaling, Amazon EC2, AWS Direct Connect, Route 53, Amazon CloudFront, Elastic Load Balancing, AWS Elastic Beanstalk, Amazon SNS, CloudWatch, CloudWatch Logs, Amazon S3, IAM, Amazon Virtual Private Cloud에서 네트워크 리소스를 생성하고 유지할 수 있는 권한을 부여합니다.

이 직무는 AWS 서비스로 역할을 전달할 수 있는 기능을 필요로 합니다. 이 정책은 다음 표에 명시된 역할에만 `iam:GetRole` 및 `iam:PassRole`을 허용합니다. 자세한 정보는 이 주제의 후반부에서 [역할 생성 및 정책 연결\(콘솔\) \(p. 548\)](#) 단원을 참조하십시오.

네트워크 관리자 직무에 대한 선택적 IAM 서비스 역할

사용 사례	역할 이름(*는 와일드 카드)	선택할 서비스 역할 유형	선택할 AWS 관리형 정책
Amazon VPC가 사용자를 대신해 CloudWatch Logs의 로그를 생성하고 관리하여 VPC로 들어오고 나가는 IP 트래픽을 모니터링하도록 허용	flow-logs-*	Amazon VPC 사용 설명서 에 정의된 대로 신뢰 정책으로 역할 생성	이 사용 사례에는 기존 AWS 관리형 정책이 없지만 설명서에는 필요한 권한이 나열되어 있습니다. Amazon VPC 사용 설명서 단원을 참조하십시오.

보안 감사

AWS 관리형 정책 이름: [SecurityAudit](#)

사용 사례: 이 사용자는 보안 요구 사항을 준수하기 위해 계정을 모니터링합니다. 이 사용자는 로그와 이벤트에 액세스하여 잠재적인 보안 위반이나 악의적인 활동을 조사할 수 있습니다.

정책 설명: 이 정책은 많은 AWS 서비스에 대한 구성 데이터를 확인하고, 해당 로그를 검토할 수 있는 권한을 부여합니다.

지원 사용자

AWS 관리형 정책 이름: [SupportUser](#)

사용 사례: 이 사용자는 AWS 지원을 통해 지원 사례를 생성하고 기존 사례의 상태를 확인합니다.

정책 설명: 이 정책은 AWS 지원 사례를 생성하고 업데이트할 수 있는 권한을 부여합니다.

시스템 관리자

AWS 관리형 정책 이름: [SystemAdministrator](#)

사용 사례: 이 사용자는 개발 작업에 필요한 리소스를 설정하고 유지합니다.

정책 설명: 이 정책은 AWS CloudTrail, Amazon CloudWatch, AWS CodeCommit, AWS CodeDeploy, AWS Config, AWS Directory Service, Amazon EC2, AWS Identity and Access Management, AWS Key Management Service, AWS Lambda, Amazon RDS, Route 53, Amazon S3, Amazon SES, Amazon SQS, AWS Trusted Advisor, Amazon VPC 등 다양한 AWS 서비스에서 리소스를 생성하고 유지할 수 있는 권한을 부여합니다.

이 직무는 AWS 서비스로 역할을 전달할 수 있는 기능을 필요로 합니다. 이 정책은 다음 표에 명시된 역할에만 `iam:GetRole` 및 `iam:PassRole`을 허용합니다. 자세한 정보는 이 주제의 후반부에서 [역할 생성 및 정책 연결\(콘솔\) \(p. 548\)](#) 단원을 참조하십시오.

시스템 관리자 직무에 대한 선택적 IAM 서비스 역할

사용 사례	역할 이름(*는 와일드 카드)	선택할 서비스 역할 유형	선택할 AWS 관리형 정책
Amazon ECS 클러스터 내 EC2 인스턴스에서 실행 중인 앱이 Amazon ECS에 액세스하도록 허용	<code>ecr-sysadmin-*</code>	EC2 Container Service에 대한 Amazon EC2 역할	AmazonEC2ContainerServiceforEC2Role
사용자가 데이터베이스를 모니터링하도록 허용	<code>rds-monitoring-role</code>	Enhanced Monitoring을 위한 Amazon RDS 역할	AmazonRDSEnhancedMonitoringRole
EC2 인스턴스에서 실행 중인 앱이 AWS 리소스에 액세스하도록 허용합니다.	<code>ec2-sysadmin-*</code>	Amazon EC2	Linux 인스턴스용 Amazon EC2 사용 설명서 에서와 같이 S3 버킷에 대한 액세스를 부여하는 역할의 정책 표본. 필요에 따라 사용자 지정
Lambda와 DynamoDB 스트림을 읽고 CloudWatch 로그에 기록하도록 허용	<code>lambda-sysadmin-*</code>	AWS Lambda	AWSLambdaDynamoDBExecutionRole

보기 전용 사용자

AWS 관리형 정책 이름: [ViewOnlyAccess](#)

사용 사례: 이 사용자는 모든 서비스에 걸쳐 계정 내 AWS 리소스와 기본 메타데이터 목록을 확인할 수 있습니다. 하지만 할당량을 초과하는 리소스 콘텐츠나 메타데이터, 리소스의 목록 정보를 읽을 수 없습니다.

정책 설명: 이 정책은 대부분의 AWS 서비스 리소스에 대한 `List*`, `Describe*`, `Get*`, `View*`, `Lookup*` 액세스를 부여합니다. 각 서비스에 대해 이 정책에 포함된 작업을 보려면 [ViewOnlyAccess](#) 단원을 참조하십시오.

역할 생성 및 정책 연결(콘솔)

앞서 나열한 여러 가지 정책은 AWS 서비스에서 사용자 대신 작업을 수행할 수 있도록 해주는 역할을 이용해 해당 서비스를 구성할 수 있는 권한을 부여합니다. 직무 정책은 반드시 사용해야 하는 정확한 역할 이름을 정의하거나, 사용할 수 있는 이름의 앞부분을 지정하는 접두사만이라도 포함합니다. 이러한 역할 중 하나를 생성하려면 다음 절차의 단계를 따릅니다.

AWS 서비스에 대한 역할을 만들려면(IAM 콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 [Roles]를 선택한 다음, [Create Role]를 선택합니다.
3. [AWS Service] 역할 유형을 선택한 후 이 역할로 수행하도록 허용하려는 서비스를 선택합니다.
4. 서비스의 사용 사례를 선택합니다. 지정한 서비스에 사용 사례가 하나뿐이면 자동으로 선택됩니다. 사용 사례는 서비스에 필요한 신뢰 정책을 포함하기 위해 서비스에서 정합니다. 그런 다음 [Next: Permissions]를 선택합니다.

5. 하나 이상의 권한 정책을 선택하여 역할에 연결합니다. 선택한 사용 사례에 따라 서비스에서 다음을 수행할 수 있습니다.
 - 역할이 사용하는 권한 정의
 - 제한된 권한 집합에서 선택할 수 있도록 허용
 - 모든 권한 집합에서 선택할 수 있도록 허용
 - 여기서 정책을 선택하지 않고, 나중에 정책을 만들어 역할에 연결할 수 있도록 허용

사용자에게 부여하려는 권한을 할당하는 정책 옆의 확인란을 선택한 후 [Next: Review]를 선택합니다.

Note

지정하는 권한은 역할을 사용하는 모든 주체가 사용할 수 있습니다. 기본적으로 역할은 권한이 없습니다.

6. [Role name]의 경우 역할 이름 사용자 지정 수준은 서비스에서 정합니다. 서비스에서 역할 이름을 정한 경우 이 옵션을 편집할 수 없습니다. 다른 경우에는 서비스에서 역할 이름의 접두사를 정의하고 사용자가 선택적으로 접미부를 입력하도록 할 수 있습니다. 일부 서비스는 역할의 전체 이름을 지정할 수 있습니다.

가능한 경우, 이 역할의 목적을 식별하는 데 도움이 되는 역할 이름이나 역할 이름 접두사를 입력합니다. 역할 이름은 AWS 계정 내에서 고유해야 합니다. 대소문자는 구별하지 않습니다. 예를 들어, 이름이 **PRODROLE**과 **prodrole**, 두 가지로 지정된 역할을 만들 수는 없습니다. 다양한 주체가 역할을 참조할 수 있기 때문에 역할이 생성된 후에는 역할 이름을 편집할 수 없습니다.

7. (선택 사항) [Role description]에 새 역할에 대한 설명을 입력합니다.
8. 역할을 검토한 다음 [Create role]을 선택합니다.

예제 1: 사용자를 데이터베이스 관리자로 구성(콘솔)

이 예제는 IAM 사용자 Alice를 [데이터베이스 관리자 \(p. 545\)](#)로 구성하는 데 필요한 단계를 보여줍니다. 이 섹션에서 테이블 첫 번째 행의 정보를 사용하여 사용자가 Amazon RDS 모니터링을 지원하도록 허용합니다. Alice가 Amazon 데이터베이스 서비스를 관리할 수 있도록 **DatabaseAdministrator** 정책을 Alice의 IAM 사용자에게 연결합니다. 이 정책을 통해 Alice는 **rds-monitoring-role**라는 역할을 Amazon RDS 서비스로 전달할 수도 있습니다. 그러면 Alice를 대신해 RDS 데이터베이스를 모니터링합니다.

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 정책을 선택한 후 검색 상자에 **database**를 입력합니다.
3. DatabaseAdministrator 정책 확인란과 Policy actions(정책 작업), 연결을 차례로 선택합니다.
4. 사용자 목록에서 Alice를 선택한 후 정책 연결을 선택합니다. 이제 Alice가 AWS 데이터베이스를 관리할 수 있습니다. 하지만 Alice가 이 데이터베이스를 모니터링하도록 허용하려면 서비스 역할을 구성해야 합니다.
5. IAM 콘솔의 탐색 창에서 역할을 선택한 후 역할 생성을 선택합니다.
6. AWS 서비스 역할 유형을 선택한 후 Amazon RDS를 선택합니다.
7. Amazon RDS Role for Enhanced Monitoring(확장 모니터링을 위한 RDS 역할) 사용 사례를 선택합니다.
8. Amazon RDS는 역할에 대한 권한을 정의합니다. 계속하려면 Next: Review(다음: 검토)를 선택합니다.
9. 역할 이름은 현재 Alice가 적용하는 DatabaseAdministrator 정책에 지정된 것 중 하나여야 합니다. 그중 하나는 **rds-monitoring-role**입니다. 이 이름을 역할 이름에 입력합니다.
10. (선택 사항) [Role description]에 새 역할에 대한 설명을 입력합니다.
11. 세부 정보를 검토한 후 역할 생성을 선택합니다.
12. 이제 Alice는 Amazon RDS 콘솔의 모니터링 섹션에서 RDS Enhanced Monitoring(RDS 확장 모니터링)을 활성화할 수 있습니다. 예를 들어, DB 인스턴스 또는 읽기 전용 복제본을 생성하거나 DB 인스턴스

를 수정할 때 이렇게 합니다. Alice는 확장 모니터링 활성화를 예로 설정하면서 역할 모니터링 텍스트 상자에 본인이 생성한 역할 이름(rds-monitoring-role)을 입력해야 합니다.

예제 2: 사용자를 네트워크 관리자로 구성(콘솔)

이 예제는 IAM 사용자 Juan을 [네트워크 관리자 \(p. 547\)](#)로 구성하는 데 필요한 단계를 보여줍니다. 해당 섹션에서 테이블의 정보를 사용하여 Juan이 VPC로 들어오고 나가는 IP 트래픽을 모니터링하도록 허용합니다. 또한 Juan이 CloudWatch Logs의 로그에서 해당 정보를 캡처하도록 허용합니다. Juan이 AWS 네트워크 리소스를 구성할 수 있도록 Juan의 IAM 사용자에게 [NetworkAdministrator](#) 정책을 연결합니다. 또한 이 정책 덕분에 흐름 로그를 작성할 때 Juan이 `f1ow-logs*`로 시작하는 역할을 Amazon EC2로 전달하도록 설정할 수 있습니다. 예제 1과 달리 이 시나리오에서는 사전 정의된 서비스 역할 유형이 없기 때문에 몇 가지 단계를 다르게 수행해야 합니다.

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택한 후 검색 상자에 **network**를 입력합니다.
3. NetworkAdministrator 정책 옆의 확인란에 이어 Policy actions(정책 작업), 연결을 차례로 선택합니다.
4. 사용자 목록에서 Juan 옆에 있는 확인란을 선택한 후 정책 연결을 선택합니다. 이제 Juan이 AWS 네트워크 리소스를 관리할 수 있습니다. 하지만 VPC 내 트래픽을 모니터링하도록 하려면 서비스 역할을 구성해야 합니다.
5. 생성해야 하는 서비스 역할에 사전 정의된 관리형 정책이 없기 때문에 먼저 이 정책부터 생성해야 합니다. 탐색 창에서 정책을 선택한 다음 정책 생성을 선택합니다.
6. JSON 탭을 선택하고 다음 JSON 정책 문서에서 텍스트를 복사합니다. 이 텍스트를 JSON 텍스트 상자에 붙여 넣습니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "logs>CreateLogGroup",  
                "logs>CreateLogStream",  
                "logs>PutLogEvents",  
                "logs>DescribeLogGroups",  
                "logs>DescribeLogStreams"  
            ],  
            "Effect": "Allow",  
            "Resource": "*"  
        }  
    ]  
}
```

7. 작업이 완료되면 [Review policy]를 선택합니다. [정책 검사기 \(p. 382\)](#)가 모든 구문 오류를 보고합니다.

Note

언제든지 Visual editor(시각적 편집기) 및 JSON 탭을 전환할 수 있습니다. 그러나 변경을 수행하거나 Visual editor(시각적 편집기) 탭에서 정책 검토를 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 내용은 [정책 재구성 \(p. 455\)](#) 단원을 참조 하십시오.

8. 검토 페이지에서 정책 이름에 **vpc-flow-logs-policy-for-service-role**을 입력합니다. 정책 요약을 검토하여 정책이 부여한 권한을 확인한 다음 정책 생성을 선택하여 작업을 저장합니다.
새로운 정책이 관리형 정책 목록에 나타나며 연결 준비가 완료됩니다.
9. IAM 콘솔의 탐색 창에서 역할을 선택한 후 역할 생성을 선택합니다.
10. AWS 서비스 역할 유형을 선택한 후 Amazon EC2를 선택합니다.

11. Amazon EC2 사용 사례를 선택합니다.
12. 권한 정책 연결(Attach permissions policies) 페이지에서 앞서 생성한 정책을 선택하고 vpc-flow-logs-policy-for-service-role과 Next: Review(다음: 검토)를 차례로 선택합니다.
13. 역할 이름은 현재 Juan이 적용하는 NetworkAdministrator 정책에서 허용한 것이어야 합니다. flow-logs-로 시작하는 이름은 무엇이든 허용됩니다. 이 예에서는 역할 이름에 **flow-logs-for-juan**을 입력합니다.
14. (선택 사항) [Role description]에 새 역할에 대한 설명을 입력합니다.
15. 세부 정보를 검토한 후 역할 생성을 선택합니다.
16. 이제 이 시나리오에 필요한 신뢰 정책을 구성할 수 있습니다. 역할 페이지에서 flow-logs-for-juan 역할(확 인란이 아닌 이름)을 선택합니다. 새 역할의 세부 정보 페이지에서 신뢰 관계 탭을 선택한 다음 Edit trust relationship(신뢰 관계 편집)을 선택합니다.
17. "Service" 라인을 다음과 같이 변경해 ec2.amazonaws.com의 항목을 교체합니다.

```
"Service": "vpc-flow-logs.amazonaws.com"
```

18. 이제 Juan이 Amazon EC2 콘솔에서 VPC나 서브넷의 흐름 로그를 생성할 수 있습니다. 흐름 로그를 생성할 때 flow-logs-for-juan 역할을 지정합니다. 이 역할에는 로그를 생성하고 데이터를 쓸 수 있는 권한이 있습니다.

AWS 전역 조건 컨텍스트 키

IAM에서 JSON 정책의 Condition 요소를 사용하여 모든 AWS API 요청의 평가 컨텍스트에 포함된 키의 값을 테스트할 수 있습니다. 이러한 키는 요청 자체 또는 해당 요청이 참조하는 리소스에 대한 정보를 제공합니다. 사용자가 요청한 작업을 허용하기 전에 키에 지정된 값이 있는지 확인할 수 있습니다. 이렇게 하면 JSON 정책 문이 수신되는 API 요청과 일치 또는 불일치할 경우 보다 세분화된 제어가 가능합니다. JSON 정책의 Condition 요소 사용에 대한 자세한 방법은 [IAM JSON 정책 요소: Condition \(p. 510\)](#) 단원을 참조하십시오.

이 주제에서는 전역으로 사용할 수 있는 키(접두사 `aws:`가 붙음)에 대해 설명합니다. [IAM과 같은 \(p. 558\)](#) AWS 서비스는 해당 서비스에서 정의하는 작업 및 리소스와 관련된 서비스별 키를 제공합니다. 자세한 내용은 [??? 단원](#)을 참조하십시오. 대개의 경우 조건 키를 지원하는 서비스의 설명서에 추가 정보를 확인할 수 있습니다. 예를 들어 Amazon S3 리소스 정책에서 사용할 수 있는 키에 대한 자세한 내용은 [Amazon Simple Storage Service 개발자 가이드의 Amazon S3 정책 키](#) 단원을 참조하십시오.

모든 AWS 서비스에 사용할 수 있는 전역 조건 키도 있지만 어떤 전역 조건 키는 일부 서비스에만 지원됩니다.

주제

- [모든 서비스에 사용 가능한 키 \(p. 551\)](#)
- [일부 서비스에 사용 가능한 키 \(p. 553\)](#)

모든 서비스에 사용 가능한 키

AWS는 모든 AWS 서비스에 다음과 같이 사전 정의된 조건 키를 제공하여 IAM의 액세스 제어를 지원합니다. 이 서비스에 대한 쓰기 정책에 대한 자세한 내용은 [??? 단원](#)을 참조하십시오.

`aws:CurrentTime`

[날짜 연산자 \(p. 515\)](#)를 사용합니다.

날짜/시간 조건을 확인하기 위한 현재 날짜 및 시간입니다.

`aws:EpochTime`

[날짜 연산자 \(p. 515\)](#)를 사용합니다.

날짜/시간 조건을 확인하기 위한 현재 날짜 및 시간(epoch 또는 Unix 시간)입니다.

aws:MultiFactorAuthAge

숫자 연산자 ([p. 515](#))를 사용합니다.

멀티 팩터 인증(MFA)을 사용하여 요청을 생성한 MFA 확인 보안 자격 증명이 얼마나 오래 전에 발행되었는지를 확인합니다. 초 단위로 표시됩니다. MFA를 사용하지 않으면 이 키는 표시되지 않습니다. [AWS에서 멀티 팩터 인증\(MFA\) 사용하기 \(p. 96\)](#) 단원을 참조하십시오. 따라서 비교 연산자의 [*IfExists \(p. 518\)](#) 버전을 사용하여 이 키도 고려해야 합니다. 이를 통해 키가 요청 컨텍스트에 없는 경우에도 비교 결과가 기대한 바와 일치하도록 할 수 있습니다.

aws:MultiFactorAuthPresent

부울 연산자 ([p. 516](#))를 사용합니다.

멀티 팩터 인증(MFA)을 사용하여 현재 요청을 한 임시 보안 자격 증명의 유효성을 검사할지 여부를 확인합니다. 이 키는 사용자가 임시 자격 증명을 사용하여 API를 호출하는 경우에만 요청 컨텍스트에 있습니다. 그러한 자격 증명은 IAM 역할, 연동 사용자, sts:GetSessionToken의 자격 증명이 있는 IAM 사용자, AWS Management 콘솔 사용자와 함께 사용합니다. 콘솔은 백그라운드에서 사용자를 대신하여 생성된 임시 자격 증명을 사용합니다. aws:MultiFactorAuthPresent 키는 표준 액세스 키 페어와 같은 장기 자격 증명으로 API 또는 CLI 명령을 호출하는 경우 존재하지 않습니다. 따라서 이 키를 확인할 때 조건 연산자의 [...IfExists \(p. 518\)](#) 버전을 사용하는 것이 좋습니다.

다음 조건 요소는 MFA를 사용하여 요청을 인증했는지를 확인할 수 있는 신뢰성 있는 방법이 아니라는 점을 이해해야 합니다.

```
##### WARNING: USE WITH CAUTION #####
"Effect" : "Deny",
"Condition" : { "Bool" : { "aws:MultiFactorAuthPresent" : false } }
```

Deny 효과, Bool 요소 및 false 값을 이렇게 조합할 경우, MFA를 사용하여 인증 가능하나 인증받지 않은 요청을 거부합니다. 이러한 조합은 MFA의 사용을 지원하는 임시 자격 증명에만 지원됩니다. 이 명령문은 장기 자격 증명을 사용해 이루어진 요청 또는 MFA를 사용해 인증받은 요청에 대한 액세스를 거부하지 않습니다. 이 예의 로직이 복잡하며 MFA 인증이 실제로 사용되었는지 테스트되지 않으므로 이 예를 사용할 때는 주의해야 합니다.

또한 Deny 효과, Null 요소 및 true의 조합은 동일한 방식으로 작동하며 그 로직이 훨씬 더 복잡하기 때문에 이 조합을 사용하지 말아야 합니다.

대신에 [BoolIfExists \(p. 518\)](#) 연산자를 사용하여, 요청이 MFA를 사용하여 인증되는지 여부를 확인하는 것이 좋습니다.

```
"Effect" : "Deny",
"Condition" : { "BoolIfExists" : { "aws:MultiFactorAuthPresent" : false } }
```

Deny, BoolIfExists 및 false를 조합할 경우, MFA를 사용해 인증되지 않은 요청을 거부합니다. 특히 MFA를 포함하지 않는 임시 자격 증명의 요청을 거부합니다. 또한 IAM 사용자 액세스 키와 같은 장기 자격 증명을 사용해 이루어진 요청을 거부합니다. *IfExists 연산자는 aws:MultiFactorAuthPresent 키의 존재 여부 및 존재 가능성 여부를, 그 실제 여부에 근거하여 확인합니다. MFA를 사용해 인증되지 않은 요청을 거부하려면 이 연산자를 사용하십시오.

또한 [BoolIfExists \(p. 518\)](#) 연산자를 사용하여, MFA를 사용해 인증받은 요청 및 장기 자격 증명을 사용해 이루어진 요청을 허용할 수 있습니다.

```
"Effect" : "Allow",
"Condition" : { "BoolIfExists" : { "aws:MultiFactorAuthPresent" : true } }
```

이 조건은 키가 존재하든 존재하지 않든 마찬가지로 일치합니다. `Allow`, `BoolIfExists` 및 `true`를 조합할 경우, MFA를 사용해 인증된 요청 또는 MFA를 사용해 인증받지 않은 요청을 허용합니다. MFA를 포함하지 않는 임시 자격 증명의 요청을 허용하지 않습니다. 이 조합을 사용하면 MFA 요청 및 장기 자격 증명을 사용해 이루어진 요청을 허용할 수 있습니다.

또는 MFA를 사용해 인증된 요청만을 허용할 수 있습니다.

```
"Effect" : "Allow",
"Condition" : { "Bool" : { "aws:MultiFactorAuthPresent" : true } }
```

`Allow`, `Bool` 및 `true`를 조합할 경우, MFA를 사용해 인증된 요청만을 허용합니다. 이러한 조합은 MFA의 사용을 지원하는 임시 자격 증명에만 지원됩니다. 이 명령문은 장기 자격 증명을 사용해 이루어진 요청 또는 MFA 없이 임시 자격 증명을 사용해 이루어진 요청에 대한 액세스를 허용하지 않습니다.

MFA 키가 있는지 여부를 확인하는 데 다음과 유사한 정책 구문을 사용하지 마십시오.

```
##### WARNING: USE WITH CAUTION #####
"Effect" : "Allow",
"Condition" : { "Null" : { "aws:MultiFactorAuthPresent" : false } }
```

`Allow` 효과, `Null` 요소 및 `false` 값을 조합할 경우, 그 요청의 실제 인증 여부와 상관없이, MFA를 사용해 인증받을 수 있는 요청만을 허용합니다. 이렇게 하여 임시 자격 증명을 사용해 이루어진 모든 요청을 허용하고 장기 자격 증명에 대한 액세스를 거부합니다. 이 예에서는 MFA 인증이 실제로 사용되었는지 여부를 테스트하지 않으므로 이 예를 사용할 때는 주의해야 합니다.

`aws:SecureTransport`

[부울 연산자 \(p. 516\)](#)를 사용합니다.

요청이 SSL을 사용하여 전송되었는지를 확인합니다.

`aws:UserAgent`

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

요청자의 클라이언트 애플리케이션을 확인합니다.

Warning

이 키를 사용할 때는 주의해야 합니다. `aws:UserAgent` 값은 HTTP 헤더의 호출자가 제공하기 때문에, 권한이 없는 사용자가 수정된 브라우저나 사용자 지정 브라우저를 사용하여 원하는 `aws:UserAgent` 값을 제공할 수 있습니다. 따라서 무단 사용자의 직접 AWS 요청을 차단할 목적으로 `aws:UserAgent`를 사용해서는 안 됩니다. 특정 클라이언트 애플리케이션을 허용하는 데 사용할 수 있으며 정책을 테스트한 후에만 사용할 수 있습니다.

일부 서비스에 사용 가능한 키

AWS는 이러한 기능을 지원하는 일부 AWS 서비스에만 다음과 같이 사전 정의된 조건 키를 제공합니다. 서비스가 이러한 조건 키를 지원하는지 여부를 확인하려면 해당 서비스에 대한 설명서를 참조해야 합니다.

Note

일부 시나리오에서만 사용할 수 있는 조건 키를 사용하는 경우 조건 연산자의 [IfExists \(p. 518\)](#) 버전을 사용할 수 있습니다. 요청 컨텍스트에 조건 키가 누락된 경우 정책 엔진이 평가에 실패할 수 있습니다. 예를 들어 특정 IP 범위 또는 특정 VPC로부터 요청이 오는 경우 ...`IfExists` 연산자와 함께 다음 정책을 사용하여 일치시켜야 합니다. 키가 하나 또는 둘 모두 없는 경우에도 조건은 일치됩니다. 요청에 지정된 키가 있는 경우에만 값을 확인합니다.

```
"Condition": { "IpAddressIfExists": { "aws:SourceIp" : [ "xxx" ] },
```

```
"StringEqualsIfExists" : {"aws:SourceVpc" : ["yyy"]} }
```

aws:PrincipalOrgID

문자열 연산자 (p. 513)를 사용합니다.

AWS Organizations를 사용하여 생성한 조직의 식별자입니다. 이 전역 키는 조직 내 모든 AWS 계정의 계정 ID를 전부 나열하는 대안을 제공합니다. 이 조건 키를 사용하여 리소스 기반 정책 (p. 326)에서 Principal 요소를 간단하게 지정할 수 있습니다. 조직의 멤버인 모든 계정을 나열하는 대신 조건 요소에 조직 ID를 지정할 수 있습니다. 계정을 추가 및 제거할 때 aws:PrincipalOrgID가 포함된 정책에는 자동으로 올바른 계정이 포함되므로 수동 업데이트가 필요하지 않습니다.

예를 들어, 다음 Amazon S3 버킷 정책을 통해 o-xxxxxxxxxxxxx 조직 내 모든 계정의 멤버는 policy-ninja-dev 버킷에 객체를 추가할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:putobject",
            "Resource": "arn:aws:s3:::policy-ninja-dev/*",
            "Condition": {"StringEquals":
                {"aws:PrincipalOrgID": ["o-xxxxxxxxxxxxx"]}
            }
        }
    ]
}
```

Note

이 전역 조건은 AWS 조직의 마스터 계정에 적용됩니다.

AWS Organizations에 대한 자세한 내용은 AWS Organizations 사용 설명서의 [AWS Organizations\(이란 무엇인가?\) 단원](#)을 참조하십시오.

이 조건 키는 일부 서비스에서만 사용할 수 있습니다.

aws:PrincipalTag/**tag-key**

문자열 연산자 (p. 513)를 사용합니다.

요청을 수행하는 보안 주체에 연결된 태그가 지정된 키 이름 및 값과 일치하는지 확인합니다.

키-값 페어의 형태로 사용자 또는 역할에 사용자 지정 속성을 추가할 수 있습니다. IAM 태그에 대한 자세한 내용은 [the section called “엔터티 태그 지정” \(p. 259\)](#) 단원을 참조하십시오. aws:PrincipalTag를 사용하여 AWS 보안 주체에 대한 액세스를 제어 (p. 336)할 수 있습니다.

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다. 정책을 사용하면 **tagManager=true** 태그가 지정된 사용자가 IAM 사용자, 그룹 또는 역할을 관리할 수 있습니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:*",
            "Resource": "*",
            "Condition": {"StringEquals": {"aws:PrincipalTag/tagManager": "true"}}
        }
    ]
}
```

}

aws:PrincipalType

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

현재 요청에 대한 보안 주체 유형(사용자, 계정, 연동 사용자 등)을 확인합니다.

이 조건 키는 일부 서비스에서만 사용할 수 있습니다.

aws:Referer

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

요청이 전송되는 주소로 클라이언트 브라우저를 참조한 사람을 확인합니다. 이 키는 [웹 브라우저에서 직접 주소를 지정할 수 있는 서비스인 Amazon S3](#) 같은 일부 서비스에서만 지원됩니다. 이 키 값은 AWS에 대한 HTTPS 요청의 referer 헤더에서 가져옵니다.

Warning

이 키를 사용할 때는 각별한 주의가 필요합니다. Amazon S3 버킷 소유자는 aws:referer를 사용하여, 권한이 없는 타사 사이트에서 해당 콘텐츠를 표준 웹 브라우저로 공급하지 못하도록 차단할 수 있습니다. 자세한 내용은 위의 링크 단원을 참조하십시오. aws:referer 값은 HTTP 헤더의 호출자가 제공하기 때문에, 권한이 없는 사용자가 수정된 브라우저나 사용자 지정 브라우저를 사용하여 원하는 aws:referer 값을 제공할 수 있습니다. 따라서 무단 사용자의 직접 AWS 요청을 차단할 목적으로 aws:referer를 사용해서는 안 됩니다. Amazon S3에 저장된 디지털 콘텐츠를 권한이 없는 타사 사이트에서 참조하지 못하도록 차단하기 위해서만 사용하십시오.

이 조건 키는 일부 서비스에서만 사용할 수 있습니다.

aws:RequestedRegion

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

특정 리전에 대한 AWS 요청이 생성되었는지 확인합니다. 이 전역 조건 키를 사용하여 사용자가 호출할 수 있는 리전을 제어할 수 있습니다. 각 서비스에 대한 AWS 리전을 보려면 [Amazon Web Services 일반 참조의 AWS 리전 및 엔드포인트](#) 단원을 참조하십시오.

IAM 등과 같은 전역 서비스에는 단일 엔드포인트가 있습니다. 그러나 이 엔드포인트는 미국 동부(버지니아 북부) 리전에 실제로 위치하기 때문에 IAM 호출은 항상 us-east-1 리전에 대해 생성됩니다. 예를 들어 요청된 리전이 us-west-2이(가) 아닌 경우 모든 서비스에 대한 액세스를 거부하는 정책을 생성하면 IAM 호출에 항상 실패합니다. 이 문제에 대한 해결 방법을 보여주는 예는 [NotAction 및 Deny \(p. 507\)](#) 단원을 참조하십시오.

Note

aws:RequestedRegion 조건 키를 사용하면 서비스의 어떤 엔드포인트를 호출할지 제어할 수 있지만 작업의 영향은 제어할 수 없습니다. 일부 서비스의 경우 교차 리전 영향이 있을 수 있습니다. 예를 들어 Amazon S3에 교차 리전 복제를 제어하는 API 작업이 있습니다. (s3:PutBucketReplication 조건 키의 영향을 받는) 한 리전에서 aws:RequestedRegion을 호출할 수 있는데 다른 리전은 복제 구성 설정에 따른 영향을 받습니다.

이 컨텍스트 키를 사용하여 지정된 리전 세트 내에서 AWS 서비스에 대한 액세스를 제한할 수 있습니다. 예를 들어, 다음 정책은 사용자가 AWS Management 콘솔에서 모든 Amazon EC2 인스턴스를 조회하도록 허용합니다. 그러나 이 정책은 아일랜드(eu-west-1), 런던(eu-west-2) 또는 파리(eu-west-3)의 인스턴스만 변경하도록 허용합니다.

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

```
{  
    "Sid": "InstanceConsoleReadOnly",  
    "Effect": "Allow",  
    "Action": [  
        "ec2:DescribeInstances",  
        "ec2:DescribeIamInstanceProfileAssociations",  
        "ec2:DescribeInstanceAttribute",  
        "ec2:DescribeReservedInstancesOfferings",  
        "ec2:DescribeClassicLinkInstances",  
        "ec2:DescribeSpotInstanceRequests",  
        "ec2:GetReservedInstancesExchangeQuote",  
        "ec2:DescribeInstanceCreditSpecifications",  
        "ec2:DescribeSpotFleetInstances",  
        "ec2:DescribeScheduledInstances",  
        "ec2:DescribeScheduledInstanceAvailability",  
        "ec2:DescribeReservedInstancesModifications",  
        "ec2:DescribeReservedInstances",  
        "ec2:DescribeReservedInstancesListings",  
        "ec2:DescribeInstanceState"  
    ],  
    "Resource": "*"  
},  
{  
    "Sid": "InstanceWriteRegionRestricted",  
    "Effect": "Allow",  
    "Action": [  
        "ec2:ModifyInstancePlacement",  
        "ec2:TerminateInstances",  
        "ec2:ImportInstance",  
        "ec2:StartInstances",  
        "ec2:MonitorInstances",  
        "ec2:RunScheduledInstances",  
        "ec2:ResetInstanceAttribute",  
        "ec2:RunInstances",  
        "ec2:ModifyInstanceAttribute",  
        "ec2:StopInstances",  
        "ec2:AssociateIamInstanceProfile",  
        "ec2:ModifyReservedInstances"  
    ],  
    "Resource": "*",  
    "Condition": {"StringEquals": {"aws:RequestedRegion": [  
        "eu-west-1",  
        "eu-west-2",  
        "eu-west-3"  
    ]}}}  
}]  
}
```

이 조건 키는 일부 서비스에서만 사용할 수 있습니다.

aws:RequestTag/tag-key

문자열 연산자 (p. 513)를 사용합니다.

이 컨텍스트 키는 "aws:RequestTag/**tag-key**": "**tag-value**" 형식으로, 여기서 **tag-key** 및 **tag-value**는 한 쌍의 태그 키와 값입니다.

AWS 요청에서 태그와 해당 값을 확인합니다. 예를 들어, 요청에 태그 키 "Dept"가 포함되어 있으며 그 값이 "Accounting"인지 확인할 수 있습니다.

이 조건 키는 일부 서비스에서만 사용 가능하며 [Amazon EC2를 위해 출시되었습니다](#).

aws:SourceAccount

문자열 연산자 (p. 513)를 사용합니다.

요청의 출처가 특정 계정인지를 확인합니다. 예를 들어, SNS 주제에 객체 생성 이벤트를 전달하도록 구성된 S3 버킷이 계정에 있다고 가정합니다. 이 경우 이 조건 키를 사용하여 Amazon S3가 혼동된 대리자로 사용되지 않는지 확인할 수 있습니다. Amazon S3는 Amazon SNS에 해당 버킷이 속한 계정을 알려 줍니다.

이 조건 키는 일부 서비스에서만 사용할 수 있습니다.

`aws:SourceArn`

[ARN 연산자 \(p. 518\)](#)를 사용합니다.

소스의 ARN(Amazon 리소스 이름) ([p. 480](#))을 사용하여 요청의 소스를 확인합니다.

이 조건 키는 일부 서비스에서만 사용할 수 있습니다.

`aws:SourceIp`

[IP 주소 연산자 \(p. 517\)](#)를 사용합니다.

요청자의 IP 주소를 확인하려면 [IP 주소 조건 연산자 \(p. 517\)](#) 단원을 참조하십시오.

Note

지정된 IP 범위 내에서 API를 호출하는 IAM 사용자, 그룹, 역할 또는 연동 사용자에 대해서만 `aws:SourceIp` 조건 키를 JSON 정책에 사용해야 합니다. 이 정책은 사용자를 대신하여 호출하는 AWS 제품에 대한 액세스를 거부합니다. 예를 들어 AWS CloudFormation에서 인스턴스를 중지하도록 Amazon EC2 호출을 허용하는 [서비스 역할 \(p. 154\)](#)이 있다고 가정하겠습니다. 이 경우, 대상 서비스(Amazon EC2)는 원래 사용자의 IP 주소가 아니라 호출 서비스(AWS CloudFormation)의 IP 주소를 인식하기 때문에 요청이 거부됩니다. JSON 정책에 따라 평가하기 위해 원본 IP 주소를 호출 서비스를 통해 대상 서비스로 보낼 방법이 없습니다.

요청이 Amazon VPC 엔드포인트를 사용하는 호스트로부터 오는 경우, `aws:SourceIp` 키를 사용할 수 없습니다. 대신에 VPC 전용 키를 사용해야 합니다. 자세한 내용은 Amazon VPC 사용 설명서의 [VPC 엔드포인트 - 엔드포인트 사용 제어](#) 단원을 참조하십시오.

`aws:SourceVpc`

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

특정 VPC에 대한 액세스를 제한합니다. 자세한 내용은 Amazon Simple Storage Service 개발자 가이드의 [특정 VPC에 대한 액세스 제한](#) 단원을 참조하십시오.

이 조건 키는 VPC 엔드포인트를 통해 트래픽을 지원하는 서비스에 사용 가능합니다.

`aws:SourceVpce`

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

특정 VPC 엔드포인트에 대한 액세스를 제한합니다. 자세한 내용은 Amazon Simple Storage Service 개발자 가이드의 [특정 VPC 엔드포인트에 대한 액세스 제한](#) 단원을 참조하십시오.

이 조건 키는 VPC 엔드포인트를 통해 트래픽을 지원하는 서비스에 사용 가능합니다.

`aws:TagKeys`

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

이 컨텍스트 키는 "`aws:TagKeys": "tag-key"`" 형식이며, 여기서 `tag-key`는 값이 없는 태그 키 목록입니다(예: ["Dept", "Cost-Center"]).

AWS 요청에 있는 태그 키를 확인합니다.

모범 사례로서 정책을 사용하여 태그를 사용한 액세스를 제어할 때 `aws:TagKeys` 조건 키를 사용하여 어떤 태그 키가 허용되는지 정의해야 합니다. 예제 정책과 자세한 내용은 [the section called “태그 키 제어” \(p. 340\)](#) 단원을 참조하십시오.

Note

일부 서비스는 리소스 생성, 수정 또는 삭제와 같은 리소스 작업을 포함한 태그 지정을 지원합니다. 태그 지정 및 단일 호출과 같은 작업을 허용하려면 태그 지정 작업 및 리소스 수정 작업을 모두 포함하는 정책을 생성해야 합니다. 그런 다음 aws:TagKeys 조건 키를 사용하여 요청 내 특정 태그 키 사용을 적용할 수 있습니다. 예를 들어 누군가 Amazon EC2 스냅샷을 생성할 때 태그를 제한하려면 ec2:CreateSnapshot 생성 작업 및 ec2:CreateTags 태그 지정 작업을 정책에 포함시켜야 합니다. aws:TagKeys를 사용하는 이 시나리오에 대한 정책을 보려면 Linux 인스턴스용 Amazon EC2 사용 설명서의 [태그를 사용하여 스냅샷 생성 단원](#)을 참조하십시오.

이 조건 키는 일부 서비스에서만 사용 가능하며 [Amazon EC2를 위해 출시되었습니다](#).

aws:TokenIssueTime

[날짜 연산자 \(p. 515\)](#)를 사용합니다.

임시 보안 자격 증명이 발급된 날짜/시간을 확인합니다.

이 조건 키는 임시 보안 자격 증명의 사용을 지원하는 일부 서비스에 대해서만 사용 가능합니다. 임시 자격 증명의 사용을 지원하는 서비스에 대해 알아보려면 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하십시오.

aws:userid

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

요청자의 사용자 ID를 확인합니다.

이 조건 키는 일부 서비스에서만 사용할 수 있습니다.

aws:username

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

요청자의 사용자 이름을 확인합니다.

이 조건 키는 일부 서비스에서만 사용할 수 있습니다.

IAM 및 AWS STS 조건 컨텍스트 키

IAM에서 JSON 정책의 Condition 요소를 사용하여 모든 AWS API 요청의 평가 컨텍스트에 포함된 키의 값을 테스트할 수 있습니다. 이러한 키는 요청 자체 또는 해당 요청이 참조하는 리소스에 대한 정보를 제공합니다. 사용자가 요청한 작업을 허용하기 전에 키에 값이 지정되었는지 확인할 수 있습니다. 이렇게 하면 JSON 정책 문이 수신되는 API 요청과 일치 또는 불일치할 경우 보다 세분화된 제어가 가능합니다. JSON 정책의 Condition 요소 사용에 대한 자세한 방법은 [IAM JSON 정책 요소: Condition \(p. 510\)](#) 단원을 참조하십시오.

이 주제에서는 IAM 서비스(iam: 접두사 포함) 및 AWS Security Token Service(AWS STS) 서비스(sts: 접두사 포함)에서 정의 및 제공하는 키에 대해 설명합니다. 다른 여러 AWS 서비스에서도 해당 서비스가 정의 한 작업 및 리소스와 관련된 서비스 고유 키를 제공합니다. 자세한 내용은 [??? 단원](#)을 참조하십시오. 대개의 경우 조건 키를 지원하는 서비스의 설명서에 추가 정보를 확인할 수 있습니다. 예를 들어 Amazon S3 리소스 정책에서 사용할 수 있는 키에 대한 자세한 내용은 Amazon Simple Storage Service 개발자 가이드의 [Amazon S3 정책 키](#) 단원을 참조하십시오.

주제

- [IAM에서 사용할 수 있는 키 \(p. 559\)](#)
- [AWS 웹 자격 증명 연동에서 사용할 수 있는 키 \(p. 560\)](#)
- [SAML 기반 AWS STS 연동에 사용할 수 있는 키 \(p. 561\)](#)
- [AWS STS에서 사용할 수 있는 키 \(p. 564\)](#)

IAM에서 사용할 수 있는 키

IAM 리소스에 대한 액세스 제어 정책에서는 다음과 같은 조건 키를 사용할 수 있습니다.

iam:AWSServiceName

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

이 역할이 연결되는 AWS 서비스를 지정합니다.

iam:PassedToService

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

역할을 전달할 수 있는 서비스의 서비스 보안 주체를 지정합니다. 서비스 보안 주체는 정책의 Principal 요소에 지정할 수 있는 서비스 이름입니다. SERVICE_NAME_URL.amazonaws.com이 일 반적인 형식입니다. iam:PassedToService 조건 키에서 역할을 수임한 서비스의 서비스 보안 주체를 제공합니다.

iam:PassedToService를 사용하여 특정 서비스에만 역할을 전달할 수 있도록 사용자를 제한할 수 있습니다. 예를 들어, 사용자는 Amazon S3 버킷에 로그 데이터를 대신 쓸 수 있도록 CloudWatch를 신뢰하는 [서비스 역할 \(p. 154\)](#)을 생성할 수 있습니다. 그런 다음 사용자는 새 서비스 역할에 권한 정책 및 신뢰 정책을 연결해야 합니다. 이 경우, 신뢰 정책은 cloudwatch.amazonaws.com 요소에 Principal을 지정해야 합니다. 사용자가 역할을 CloudWatch에 전달하도록 허용하는 다음 정책을 연결합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iam:PassRole",  
            "Resource": "*",  
            "Condition": {"StringEquals": {"iam:PassedToService":  
                "cloudwatch.amazonaws.com"} }  
        }  
    ]  
}
```

이 조건 키를 사용하면 사용자가 여러분이 지정한 서비스에 대해서만 서비스 역할을 생성하도록 할 수 있습니다. 예를 들어 앞서 언급한 정책이 적용되는 사용자가 Amazon EC2에 대한 서비스 역할을 생성하려고 하면 해당 사용자에게는 Amazon EC2로 역할을 전달할 권한이 없기 때문에 작업에 실패합니다.

iam:PermissionsBoundary

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

지정한 정책이 IAM 보안 주체 리소스에 권한 경계로서 연결되어 있는지 확인합니다. 자세한 내용은 [IAM 엔터티에 대한 권한 경계 \(p. 317\)](#) 단원을 참조하십시오.

iam:PolicyARN

[ARN 연산자 \(p. 518\)](#)를 사용합니다.

관리형 정책이 포함된 요청에서 관리형 정책의 Amazon 리소스 이름(ARN)을 확인합니다. 자세한 내용은 [정책에 대한 액세스 제어 \(p. 332\)](#) 단원을 참조하십시오.

iam:ResourceTag/**key-name**

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

자격 검증 리소스(사용자 또는 역할)에 연결된 태그가 지정된 키 이름 및 값과 일치하는지 확인합니다.

키-값 페어의 형태로 사용자 또는 역할에 사용자 지정 속성을 추가할 수 있습니다. IAM 태그에 대한 자세한 내용은 [the section called “엔터티 태그 지정” \(p. 259\)](#) 단원을 참조하십시오. `iam:ResourceTag`를 사용하여 IAM 사용자 및 역할에 대한 액세스를 제어 ([p. 336](#)) 할 수 있습니다. 그러나 IAM은 그룹에 대한 태그를 지원하지 않으므로 태그를 사용하여 그룹에 대한 액세스를 제어할 수 없습니다.

이 예제에서는 다음과 같은 정책을 생성할 수 있는 방법을 보여 줍니다. `status=terminated` 태그를 통해 사용자를 삭제할 수 있습니다. 이 정책을 사용하려면 정책 예제의 빨간색 기울임꼴 텍스트를 본인의 정보로 대체하십시오.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iam>DeleteUser",  
            "Resource": "*",  
            "Condition": {"StringLike": {"iam:ResourceTag/status": "terminated"}}  
        }  
    ]  
}
```

AWS 웹 자격 증명 연동에서 사용할 수 있는 키

웹 자격 증명 연동을 사용하여 ID 공급자(IdP)를 통해 인증된 사용자에게 임시 보안 자격 증명을 제공할 수 있습니다. 이러한 공급자의 예로는 Login with Amazon, Amazon Cognito, Google 또는 Facebook 등이 있습니다. 이 경우, 임시 보안 자격 증명을 사용해 요청하는 경우 추가 조건 키를 사용할 수 있습니다. 이러한 키를 사용해 연동 사용자가 특정 공급자, 앱 또는 사용자와 연동된 리소스에만 액세스할 수 있도록 정책 작성이 가능합니다.

`aws:FederatedProvider`

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

`FederatedProvider` 키는 사용자 인증에 사용된 IdP를 식별합니다. 예를 들어 Amazon Cognito를 통해 사용자가 인증된 경우 키에 `cognito-identity.amazonaws.com`이 포함됩니다. 마찬가지로 Login with Amazon을 통해 사용자가 인증된 경우에는 키에 `www.amazon.com` 값이 포함됩니다. 이러한 리소스 키는 다음과 같이 `aws:FederatedProvider` 키를 리소스 ARN의 정책 변수로 사용하는 리소스 정책에서 사용할 수 있습니다. 이 정책은 IdP를 사용하여 인증된 모든 사용자가 Amazon S3 버킷의 폴더에서 객체를 가져올 수 있도록 허용합니다. 그러나 버킷은 사용자를 인증한 공급자마다 달라야 합니다.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": "s3:GetObject",  
            "Resource": "arn:aws:s3:::BUCKET-NAME/${aws:FederatedProvider}/*"  
        }  
    ]  
}
```

애플리케이션 ID 및 사용자 ID

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

그 밖에도 2가지 키를 통해 사용자 고유 식별자와 사용자 인증에 사용되는 애플리케이션 또는 사이트 식별자를 제공할 수 있습니다. 이 키들은 IdP에 따라 다음과 같은 이름을 같습니다.

- Amazon Cognito 사용자의 경우 키의 이름은 `cognito-identity.amazonaws.com:aud`(자격 증명 풀 ID)와 `cognito-identity.amazonaws.com:sub`(사용자 ID)입니다.

- Login With Amazon 사용자의 경우 키의 이름은 `www.amazon.com:app_id`와 `www.amazon.com:user_id`입니다.
- Facebook 사용자의 경우 키의 이름은 `graph.facebook.com:app_id`와 `graph.facebook.com:id`입니다.
- Google 사용자의 경우 키의 이름은 `accounts.google.com:aud`(앱 ID)와 `accounts.google.com:sub`(사용자 ID)입니다.

Amazon Cognito의 amr 키

문자열 연산자 (p. 513)를 사용합니다.

웹 자격 증명 연동으로 Amazon Cognito를 사용하는 경우에는 신뢰 정책의 `cognito-identity.amazonaws.com:amr` 키(Authentication Methods Reference)에 사용자 로그인 정보가 추가됩니다. 이 키는 다수의 값을 갖습니다. 이 말은 정책 내에서 조건 설정 연산자 (p. 520)를 사용하여 테스트한다는 것을 의미합니다. 키에 추가되는 값은 다음과 같습니다.

- 사용자 인증 전에는 키에 `unauthenticated` 값만 추가됩니다.
- 사용자 인증 후에는 키에 `authenticated` 값과 호출 시 사용된 로그인 공급자 이름 (`graph.facebook.com`, `accounts.google.com` 또는 `www.amazon.com`)이 추가됩니다.

한 예로, Amazon Cognito 역할의 신뢰 정책에서는 다음 조건에 따라 사용자의 인증 여부를 테스트합니다.

```
"Condition": {  
    "StringEquals":  
        { "cognito-identity.amazonaws.com:aud": "us-east-2:identity-pool-id" },  
    "ForAnyValue:StringLike":  
        { "cognito-identity.amazonaws.com:amr": "unauthenticated" }  
}
```

웹 자격 증명 연동에 대한 자세한 내용

웹 자격 증명 연동에 대한 자세한 내용은 다음 주제 단원을 참조하십시오.

- Android용 AWS Mobile SDK Developer Guide 안내서의 [Amazon Cognito 개요](#)
- AWS Mobile SDK for iOS Developer Guide 안내서의 [Amazon Cognito 개요](#)
- 웹 자격 증명 연동에 대하여 (p. 162)

SAML 기반 AWS STS 연동에 사용할 수 있는 키

AWS Security Token Service(AWS STS)를 사용하여 SAML 기반 연동으로 작업하는 경우 정책에 조건 키를 추가할 수 있습니다.

SAML 역할 신뢰 정책

역할 신뢰 정책에서는 다음과 같은 키를 추가하여 호출자의 역할 위임 가능 여부를 구성할 수 있습니다. `saml:doc`를 제외한 모든 값은 SAML 어설션에서 가져옵니다. 목록에서 별표(*)가 표시된 항목은 조건 생성을 위한 콘솔 UI로 사용할 수 있습니다. []가 표시된 항목은 지정된 유형의 목록을 값으로 가질 수 있습니다.

`saml:aud`

문자열 연산자 (p. 513)를 사용합니다.

SAML 어설션이 전송되는 엔드포인트 URL입니다. 이 키에 대한 값은 Audience 필드가 아닌 어설션의 SAML Recipient 필드에서 얻습니다.

`saml:cn[]`

문자열 연산자 (p. 513)를 사용합니다.

이 키는 eduOrg 속성입니다.

saml:doc*

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

이 키는 역할 위임 시 사용한 보안 주체를 나타냅니다. 형식은 *account-ID/provider-friendly-name*(예: 123456789012/SAMLProviderName)을 따릅니다. account-ID 값은 SAML 공급자 ([p. 177](#))가 속한 계정을 참조합니다.

saml:edupersonaffiliation[]

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

이 키는 eduPerson 속성입니다.

saml:edupersonassurance[]

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

이 키는 eduPerson 속성입니다.

saml:edupersonentitlement[]*

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

이 키는 eduPerson 속성입니다.

saml:edupersonnickname[]

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

이 키는 eduPerson 속성입니다.

saml:edupersonorgdn*

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

이 키는 eduPerson 속성입니다.

saml:edupersonorgunitdn[]

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

이 키는 eduPerson 속성입니다.

saml:edupersonprimaryaffiliation

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

이 키는 eduPerson 속성입니다.

saml:edupersonprimaryorgunitdn

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

이 키는 eduPerson 속성입니다.

saml:edupersonprincipalname

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

이 키는 eduPerson 속성입니다.

saml:edupersonscopedaffiliation[]

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

이 키는 eduPerson 속성입니다.

saml:edupersontargetedid[]

문자열 연산자 ([p. 513](#))를 사용합니다.

이 키는 eduPerson 속성입니다.

saml:eduorghomepageuri[]

문자열 연산자 ([p. 513](#))를 사용합니다.

이 키는 eduOrg 속성입니다.

saml:eduorgidentityauthnpolicyuri[]

문자열 연산자 ([p. 513](#))를 사용합니다.

이 키는 eduOrg 속성입니다.

saml:eduorglegalname[]

문자열 연산자 ([p. 513](#))를 사용합니다.

이 키는 eduOrg 속성입니다.

saml:eduorgsuperioruri[]

문자열 연산자 ([p. 513](#))를 사용합니다.

이 키는 eduOrg 속성입니다.

saml:eduorgwhitepagesuri[]

문자열 연산자 ([p. 513](#))를 사용합니다.

이 키는 eduOrg 속성입니다.

saml:namequalifier*

문자열 연산자 ([p. 513](#))를 사용합니다.

' 문자로 구분되는 Issuer 반응 값(saml:iss), AWS 계정 ID, IAM의 SAML 공급자 표시 이름(ARN의 마지막 부분)의 연속값에 기반을 둔 해시 값. 계정 ID, SAML 공급자 표시 이름의 연속값은 IAM 정책에서 키 saml:doc으로 사용 가능합니다. 자세한 내용은 [SAML 기반 연동에서 사용자를 고유하게 식별하기](#) ([p. 170](#)) 단원을 참조하십시오.

saml:iss*

문자열 연산자 ([p. 513](#))를 사용합니다.

발급자로서 URN으로 표시됩니다.

saml:sub*

문자열 연산자 ([p. 513](#))를 사용합니다.

이것은 클레임의 주체로서 여기에는 조직 내 사용자 개개인을 식별할 수 있는 고유 값이 포함됩니다(예: _cbb88bf52c2510eabe00c1642d4643f41430fe25e3).

saml:sub_type*

문자열 연산자 ([p. 513](#))를 사용합니다.

이 키는 persistent, transient 값을 갖거나 SAML 어설션에서 사용되는 Format 및 Subject 요소의 전체 NameID URI로 구성될 수 있습니다. persistent의 값은 saml:sub의 값이 세션 간 사용자에서도 동일하다는 것을 나타냅니다. 값이 transient인 경우 각 세션마다 사용자의 saml:sub 값이 다릅니다. NameID 요소의 Format 속성에 대한 자세한 내용은 [인증 응답을 위한 SAML 어설션 구성](#) ([p. 181](#)) 단원을 참조하십시오.

eduPerson 및 eduOrg 속성에 대한 일반적인 정보는 [Internet2 웹사이트](#) 단원을 참조하십시오.
eduPerson 속성 목록은 [eduPerson Object Class Specification\(201203\)](#) 단원을 참조하십시오.

형식이 목록인 조건 키에는 다수의 값이 추가될 수 있습니다. 목록 값 정책에서 조건을 생성하려면 [설정 연산자 \(p. 520\)](#)(ForAllValues, ForAnyValue)를 사용하면 됩니다. 예를 들어 소속이 "faculty", "staff", 또는 "employee"(student", alum" 또는 기타 가능한 소속 제외)인 사용자는 모두 허용하려면 다음과 같이 조건을 사용할 수 있습니다.

```
"Condition": {  
    "ForAllValues:StringLike": {  
        "saml:edupersonaffiliation": [ "faculty", "staff", "employee" ]  
    }  
}
```

SAML 역할 권한 정책

역할 권한 정책에서 SAML 연동으로 액세스 가능한 AWS 서비스를 정의할 때는 다음과 같은 키를 추가할 수 있습니다.

saml:namequalifier

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

이 키에는 saml:doc와 saml:iss 값의 조합을 나타내는 해시 값이 저장됩니다. 이 해시 값은 네임스페이스 한정자로 사용되어 saml:namequalifier와 saml:sub의 조합으로 사용자를 식별합니다.

saml:sub

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

이것은 클레임의 주체로서 여기에는 조직 내 사용자 개개인을 식별할 수 있는 고유 값이 포함됩니다(예: _cbb88bf52c2510eabe00c1642d4643f41430fe25e3).

saml:sub_type

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

이 키는 persistent, transient 값을 갖거나 SAML 어설션에서 사용되는 Format 및 Subject 요소의 전체 NameID URI로 구성될 수 있습니다. persistent의 값은 saml:sub의 값이 세션 간 사용자에서도 동일하다는 것을 나타냅니다. 값이 transient인 경우 각 세션마다 사용자의 saml:sub 값이 다릅니다. NameID 요소의 Format 속성에 대한 자세한 내용은 [인증 응답을 위한 SAML 어설션 구성 \(p. 181\)](#) 단원을 참조하십시오.

이러한 키를 사용하는 방법은 [SAML 2.0 기반 연동에 대하여 \(p. 167\)](#) 단원을 참조하십시오.

AWS STS에서 사용할 수 있는 키

AWS Security Token Service(AWS STS) 작업을 사용하여 맙는 역할에 대한 IAM 역할 신뢰 정책에서는 다음 조건 키를 사용할 수 있습니다.

sts:ExternalId

[문자열 연산자 \(p. 513\)](#)를 사용합니다.

다른 계정에서 역할을 맡을 때 필요할 수도 있는 고유한 식별자. 역할이 속한 계정의 관리자가 외부 ID를 제공한 경우에는 해당 값을 ExternalId 파라미터에 제공하십시오. 이 값은 암호 또는 계정 번호와 같은 어떤 문자열도 가능합니다. 교차 계정 역할은 일반적으로 계정의 모든 사용자를 신뢰하도록 설정되므로 신뢰하는 계정의 관리자는 신뢰할 수 있는 계정의 관리자에게 외부 ID를 보낼 수 있습니다. 이와 같이 계정의 모든 사용자가 아닌, ID를 가진 사람만이 역할을 맡을 수 있습니다. 외부 ID에 대한 자세한 내용은 [AWS 리소스에 대한 액세스를 타사에 부여할 때 외부 ID를 사용하는 방법 \(p. 206\)](#)을 참조하십시오.

`ExternalId` 값은 최소 2자, 최대 1,224자여야 합니다. 이 값은 공백 없이 영숫자여야 합니다. 이 값은 더하기(+), 등호(=), 쉼표(,), 마침표(.), 기호(@), 콜론(:), 슬래시(/) 및 하이픈(-)과 같은 기호도 포함할 수 있습니다.

리소스

IAM은 풍부한 기능을 갖춘 제품이며, IAM으로 AWS 계정 및 리소스를 보호하는 방법이 자세히 설명된 리소스가 많이 있습니다.

주제

- [사용자 및 그룹 \(p. 566\)](#)
- [자격 증명\(암호, 액세스 키 및 MFA 디바이스\) \(p. 566\)](#)
- [권한 및 정책 \(p. 566\)](#)
- [연동 및 위임 \(p. 567\)](#)
- [IAM 및 기타 AWS 제품 \(p. 567\)](#)
- [일반 보안 사례 \(p. 568\)](#)
- [일반 리소스 \(p. 568\)](#)

사용자 및 그룹

사용자 및 그룹을 생성하고, 관리하고, 사용하는 방법은 다음 리소스를 참조하십시오.

- [첫 번째 IAM 관리자 및 그룹 생성 \(p. 17\)](#) – IAM 사용자를 생성하고 권한을 할당하는 방법을 보여 주는 단계별 절차입니다.
- [자격 증명\(사용자, 그룹, 및 역할\) \(p. 61\)](#) – IAM 사용자 및 그룹 관리 방법을 심층적으로 다룹니다.
- [계정, 사용자 및 그룹을 사용하는 경우에 대한 지침](#) – AWS 보안 블로그 게시물로, IAM 사용자 및 그룹이 단일 계정을 사용하거나 별도의 AWS 계정을 사용하도록 사용자 액세스를 구성하는 방법을 다룹니다.

자격 증명(암호, 액세스 키 및 MFA 디바이스)

AWS 계정 및 IAM 사용자의 암호를 관리하는 방법은 다음 가이드를 검토하십시오. AWS에 대한 프로그래밍 호출을 하는 데 사용되는 보안 키인 액세스 키에 대한 정보도 찾을 수 있습니다.

- [AWS 보안 자격 증명](#) – Amazon Web Services에 액세스하는 데 사용되는 자격 증명의 유형을 설명하고, 자격 증명을 생성 및 관리하는 방법을 살펴보고, 액세스 키를 안전하게 관리하기 위한 권장 사항을 소개합니다.
- [암호 관리 \(p. 78\)](#) 및 [IAM 사용자의 액세스 키 관리 \(p. 88\)](#) – 계정의 IAM 사용자에 대한 자격 증명 관리 옵션을 설명합니다.
- [AWS에서 멀티 팩터 인증\(MFA\) 사용하기 \(p. 96\)](#) – 디바이스에서 암호와 일회용 코드를 둘 다 입력해야 로그인이 허용되도록 계정 및 IAM 사용자를 구성하는 방법을 설명합니다. (이를 이중 인증이라고도 부릅니다.)

권한 및 정책

IAM 정책 내부의 작동 방식과 함께 가장 효과적으로 권한을 부여하는 방법도 알아보십시오.

- 정책 및 권한 (p. 305) – 사용자나 그룹에 또는 일부 AWS 제품의 경우 리소스 자체에 권한을 연결하는 방법을 설명합니다.
- 정책 및 권한 (p. 305) – 권한을 정의하는 데 사용되는 정책 언어를 소개합니다.
- IAM JSON 정책 요소 참조 (p. 498) – 각 정책 언어 요소에 대한 설명과 예제를 소개합니다.
- IAM 자격 증명 기반 정책 예제 (p. 341) – 다양한 AWS 제품의 일반적인 작업에 대한 몇 가지 정책 예제를 보여 줍니다.
- AWS 정책 생성기 – 목록에서 제품과 작업을 선택하여 사용자 지정 정책을 생성합니다.
- IAM 정책 시뮬레이터 – 정책에서 특정 AWS 작업을 허용할지 아니면 거부할지 여부를 테스트합니다. 다음 동영상(6:28)에서는 정책 시뮬레이터를 소개하고 작동하는 모습도 보여 줍니다.

[IAM 정책 시뮬레이터 시작하기](#)

연동 및 위임

다른 곳에서 인증된(로그인한) 사용자에게 AWS 계정의 리소스에 대한 액세스 권한을 부여할 수 있습니다. 여기에는 다른 AWS 계정의 IAM 사용자(위임), 해당 조직의 로그인 프로세스를 통해 인증된 사용자, Login with Amazon, Facebook, Google 또는 기타 OpenID Connect(OIDC) 호환 자격 증명 공급자 등 인터넷 자격 증명 공급자의 사용자가 포함될 수 있습니다. 이 경우 사용자는 AWS 리소스에 액세스할 수 있는 임시 보안 자격 증명을 받게 됩니다.

- [자습서: IAM 역할을 사용한 AWS 계정 간 액세스 권한 위임 \(p. 29\)](#) – 다른 AWS 계정의 IAM 사용자에게 교차 계정 액세스 권한을 부여하는 방법을 설명합니다.
- [임시 자격 증명과 관련된 일반적인 시나리오 \(p. 264\)](#) – AWS 외부에서 인증된 사용자를 AWS로 연동하는 방법을 설명합니다.
- [Web Identity Federation Playground](#) – Login with Amazon, Google 또는 Facebook을 사용하여 Amazon S3에 인증한 다음 호출해 봅니다.

IAM 및 기타 AWS 제품

대부분의 AWS 제품은 IAM과 통합되므로 IAM 기능을 사용하여 그러한 제품의 리소스에 대한 액세스를 보호할 수 있습니다. 다음 리소스에서는 IAM 및 가장 인기 있는 일부 AWS 제품의 보안을 다룹니다. IAM을 사용하는 전체 제품 목록과 각각의 추가 정보 링크는 [IAM로 작업하는 AWS 서비스 \(p. 488\)](#) 단원을 참조하십시오.

Using IAM with Amazon EC2

- [Amazon EC2 리소스에 대한 액세스 제어](#) – 사용자가 Amazon EC2 인스턴스, 볼륨 등을 관리할 수 있도록 IAM 기능으로 허용하는 방법을 설명합니다.
- [인스턴스 프로파일 사용 \(p. 244\)](#) – IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되면서 다른 AWS 제품에 액세스해야 하는 애플리케이션에 안전하게 자격 증명을 제공하는 방법을 설명합니다.

Using IAM with Amazon S3

- [Amazon S3 리소스에 대한 액세스 권한 관리](#) – Amazon S3 정책을 포함하여 버킷 및 객체에 대한 IAM 보안 모델을 다룹니다.
- [IAM 정책 작성: Amazon S3 버킷의 사용자별 폴더에 대한 액세스 권한 부여](#) – 사용자가 Amazon S3의 자체 폴더를 직접 보호하는 방법을 다룹니다. (Amazon S3 및 IAM에 대한 게시물을 더 보려면 블로그 게시물 제목 아래에서 S3 태그를 선택하십시오.)

Using IAM with Amazon RDS

- [AWS Identity and Access Management\(IAM\)를 사용하여 Amazon RDS 리소스에 대한 액세스 관리](#) – IAM을 사용하여 데이터베이스 인스턴스, 데이터베이스 스냅샷 등에 대한 액세스 권한을 제어하는 방법을 설명합니다.
- [RDS 리소스 수준 권한에 대한 소개](#) – IAM을 사용하여 특정 Amazon RDS 인스턴스에 대한 액세스를 제어하는 방법을 다룹니다.

Using IAM with Amazon DynamoDB

- [IAM을 사용하여 DynamoDB 리소스에 대한 액세스 제어](#) – 사용자가 DynamoDB 테이블과 인덱스를 관리할 수 있도록 IAM으로 허용하는 방법을 설명합니다.
- 다음 동영상(8:55)에서는 개별 DynamoDB 데이터베이스 항목이나 속성(또는 둘 다)에 대한 액세스 제어를 제공하는 방법을 설명합니다.

[Getting Started with Fine-Grained Access Control for DynamoDB](#)

일반 보안 사례

AWS 계정과 리소스를 보호하는 가장 좋은 방법에 대한 전문적인 팁과 지침을 찾아보십시오.

- [AWS 보안 모범 사례\(PDF\)](#) – AWS 계정과 제품 전반에서 보안을 관리하는 방법을 비롯하여 보안 아키텍처, IAM 사용, 암호화 및 데이터 보안 등에 대한 권장 사항을 자세하게 살펴 봅니다.
- [IAM 모범 사례 \(p. 43\)](#) – IAM을 사용하여 AWS 계정과 리소스를 보호하는 방법에 대한 권장 사항을 제시합니다.
- [AWS CloudTrail User Guide](#) – AWS CloudTrail을 사용하여 AWS에 대한 API 호출 기록을 추적하고 로그 파일에 해당 정보를 저장합니다. 이를 통해 계정의 리소스에 액세스한 사용자와 계정, 호출이 발생한 시기, 요청된 작업 등을 확인할 수 있습니다.

일반 리소스

다음 리소스에서 IAM 및 AWS에 대해 자세히 알아보십시오.

- [IAM에 대한 제품 정보](#) – AWS Identity and Access Management 제품에 대한 일반 정보입니다.
- [AWS Identity and Access Management 토론 포럼](#) – IAM과 관련된 기술적 질문에 대해 토론할 수 있는 고객을 위한 커뮤니티 포럼입니다.
- [교육 및 워크숍](#) – 역할 기반의 과정 및 전문 과정은 물론 자습형 실습에 대한 링크를 통해 AWS 기술을 연마하고 실용적인 경험을 쌓을 수 있습니다.
- [AWS 개발자 도구](#) – AWS 애플리케이션을 개발 및 관리하기 위한 개발자 도구, SDK, IDE 도구 키트 및 명령줄 도구 링크.
- [AWS 백서](#) – AWS 솔루션 아키텍트 또는 기타 기술 전문가가 아키텍처, 보안 및 경제 등의 주제에 대해 작성한 포괄적 AWS 기술 백서 목록의 링크.
- [AWS Support 센터](#) – AWS 지원 사례를 생성 및 관리하는 허브. 또한 포럼, 기술 FAQ, 서비스 상태 및 AWS Trusted Advisor 등의 기타 유용한 자료에 대한 링크가 있습니다.
- [AWS Support](#) – 클라우드에서 1대 1로 애플리케이션을 구축 및 실행하도록 지원하는 빠른 응답 지원 채널인 AWS Support에 대한 정보가 포함된 기본 웹 페이지.
- [Contact Us](#) – AWS 결제, 계정, 이벤트, 침해 및 기타 문제에 대해 문의할 수 있는 중앙 연락 창구.

- [AWS 사이트 약관](#) – 저작권 및 상표, 사용자 계정, 라이선스 및 사이트 액세스와 기타 주제에 대한 세부 정보.

HTTP 쿼리 요청을 통한 API 호출

주제

- [엔드포인트 \(p. 570\)](#)
- [HTTPS 필요 \(p. 570\)](#)
- [IAM API 요청에 서명 \(p. 571\)](#)

이 단원에는 Query API for AWS Identity and Access Management(IAM) 및 AWS Security Token Service(AWS STS) 사용에 대한 일반적인 정보가 포함되어 있습니다. API 작업 및 오류에 대한 자세한 내용은 [IAM API Reference](#) 또는 [AWS Security Token Service API Reference](#)를 참조하십시오.

Note

IAM 또는 AWS STS API 작업을 직접 호출하는 대신 AWS SDK 중 하나를 사용할 수 있습니다. AWS SDK는 다양한 프로그래밍 언어 및 플랫폼(Java, Ruby, .NET, iOS, Android 등)을 위한 라이브러리와 샘플 코드로 구성되어 있습니다. SDK를 사용하면 편리하게 IAM 및 AWS에 프로그래밍 방식으로 액세스할 수 있습니다. 예를 들어 SDK는 요청에 암호화 방식으로 서명(아래 참조), 오류 관리 및 자동으로 요청 재시도와 같은 작업을 처리합니다. 다운로드 및 설치 방법을 비롯하여 AWS SDK에 대한 자세한 내용은 [Amazon Web Services용 도구](#) 페이지를 참조하십시오.

Query API for IAM 및 AWS STS를 사용하면 서비스 작업을 호출할 수 있습니다. 쿼리 API 요청은 수행할 작업을 나타내기 위해 Action 파라미터를 포함해야 하는 HTTPS 요청입니다. IAM 및 AWS STS에서는 모든 작업에 대해 GET 및 POST 요청을 지원합니다. 즉, API 사용 시 어떤 작업에는 GET을 사용하고 또 어떤 작업에는 POST를 사용할 필요가 없습니다. 하지만 GET 요청에는 URL 크기 제한이 적용됩니다. 이 제한은 브라우저에 따라 다르지만 일반적으로 2,048바이트입니다. 따라서 더 큰 크기가 필요한 쿼리 API 요청의 경우 POST 요청을 사용해야 합니다.

응답은 XML 문서입니다. 응답에 대한 자세한 내용은 [IAM API Reference](#) 또는 [AWS Security Token Service API Reference](#)의 개별 작업 페이지를 참조하십시오.

엔드포인트

IAM 및 AWS STS에는 전역적 엔드포인트가 하나씩 있습니다.

- (IAM) <https://iam.amazonaws.com>
- (AWS STS) <https://sts.amazonaws.com>

Note

AWS STS에서는 전역적 엔드포인트 외에 리전 엔드포인트로 요청을 보내는 작업도 지원됩니다. 한 리전에서 AWS STS를 사용하려면 먼저 해당 리전에서 본인의 AWS 계정에 대해 STS를 활성화해야 합니다. AWS STS에 대해 추가 리전을 활성화하는 방법은 [AWS 리전에서 AWS STS 활성화 및 비활성화 \(p. 287\)](#) 단원을 참조하십시오.

모든 서비스용 AWS 엔드포인트 및 리전에 대한 자세한 내용은 [AWS General Reference](#)의 리전 및 엔드포인트를 참조하십시오.

HTTPS 필요

쿼리 API는 보안 자격 증명과 같이 민감한 정보를 반환하므로 모든 API 요청에 HTTPS를 사용해야 합니다.

IAM API 요청에 서명

액세스 키 ID와 보안 액세스 키를 사용하여 요청에 서명해야 합니다. IAM에서의 일상적인 작업에는 AWS 계정 루트 사용자 자격 증명을 사용하지 않는 것이 좋습니다. IAM 사용자용 자격 증명을 사용하거나 AWS STS 를 사용하여 임시 보안 자격 증명을 생성할 수 있습니다.

API 요청에 서명하려면 AWS 서명 버전 4를 사용하는 것이 좋습니다. 서명 버전 4 사용에 대한 자세한 내용은 AWS 일반 참조의 [서명 버전 4 서명 프로세스](#)를 참조하십시오.

서명 버전 2를 사용해야 할 경우 서명 버전 2 사용에 대한 자세한 내용은 [AWS 일반 참조](#)를 참조하십시오.

자세한 내용은 다음 자료를 참조하십시오.

- [AWS 보안 자격 증명](#). AWS 액세스에 사용되는 자격 증명 유형에 대한 일반적인 정보를 제공합니다.
- [IAM 모범 사례 \(p. 43\)](#)를 선택하십시오. IAM 서비스를 사용하여 AWS 리소스를 보호하기 위한 제안 사항의 목록을 제공합니다.
- [임시 보안 자격 증명 \(p. 263\)](#)를 선택하십시오. 임시 보안 자격 증명을 만들고 사용하는 방법에 대해 설명합니다.

IAM 문서 기록

다음 표에서는 본 IAM 관련 주요 설명서 업데이트를 설명합니다.

update-history-change	update-history-description	update-history-date
IAM 사용자 내 보안 자격 증명 페이지	이제 IAM 사용자는 My Security Credentials(내 보안 자격 증명) 페이지에서 자신의 모든 자격 증명을 관리할 수 있습니다. 이 AWS Management 콘솔 페이지에는 계정 ID 및 정식 사용자 ID와 같은 계정 정보가 표시됩니다. 또한 사용자는 자신의 암호, 액세스 키, X.509 인증서, SSH 키, Git 자격 증명을 보고 편집할 수 있습니다.	January 24, 2019
액세스 관리자 API	이제 AWS CLI 및 AWS API를 사용하여 서비스에서 마지막으로 액세스한 데이터를 볼 수 있습니다.	December 7, 2018
IAM 사용자 및 역할 태그 지정	이제 IAM 태그를 사용하여 태그 키-값 페어를 통해 사용자 지정 속성을 자격 증명(IAM 사용자 또는 역할)에 추가할 수 있습니다. 태그를 사용하여 리소스에 대한 자격 증명의 액세스를 제어하거나 자격 증명에 연결할 수 있는 태그를 제어할 수도 있습니다.	November 14, 2018
U2F 보안 키	이제 AWS Management 콘솔에 로그인할 때 멀티 팩터 인증(MFA) 옵션으로 U2F 보안 키를 사용할 수 있습니다.	September 25, 2018
Amazon VPC 엔드포인트에 대한 지원	이제 미국 서부(오레곤) 리전에서 VPC와 AWS STS 간에 프라이빗 연결을 설정할 수 있습니다.	July 31, 2018
권한 경계	새 기능을 사용하면 IAM 권한을 관리할 수 있는 권한을 신뢰할 수 있는 직원에게 부여하는 작업이 더 간편해질 뿐 아니라 IAM 관리자 액세스 권한 전체를 부여하지 않아도 됩니다.	July 12, 2018
aws:PrincipalOrgID	새로운 조건 키는 IAM 보안 주체의 AWS 조직을 지정하여 AWS 리소스로의 액세스를 제어하는 쉬운 방법을 제공합니다.	May 17, 2018
aws:RequestedRegion	새로운 조건 키는 IAM 정책을 사용하여 AWS 리전으로의 액세스를 제어하는 쉬운 방법을 제공합니다.	April 25, 2018

IAM 역할에 대한 세션 기간 증가	이제 IAM 역할은 12시간의 세션 기간을 가질 수 있습니다.	March 28, 2018
역할 생성된 워크플로우 업데이트	새로운 워크플로우는 신뢰 관계를 생성하고 권한을 역할에 연결하는 과정을 개선합니다.	September 8, 2017
AWS 계정 로그인 절차	AWS 로그인 경험을 업데이트 하여 루트 사용자와 IAM 사용자 모두가 AWS Management 콘솔 콘솔 홈페이지의 Sign In to the Console(콘솔로 로그인) 링크를 사용할 수 있도록 허용합니다.	August 25, 2017
IAM 정책 예제	30 가지의 예제 정책이 넘는 설명 서 업데이트 기능.	August 2, 2017
IAM 모범 사례	IAM 콘솔의 사용자 섹션에 추가된 정보는 IAM 모범 사례를 쉽게 따라할 수 있게 만듭니다.	July 5, 2017
Auto Scaling 리소스	리소스 수준 권한은 Auto Scaling 리소스로의 액세스 및 권한을 제어할 수 있습니다.	May 16, 2017
MySQL 및 Amazon Aurora 데이터베이스를 위한 Amazon RDS	데이터베이스 관리자는 데이터베이스 사용자를 IAM 사용자 및 역할과 연관시킬 수 있어 사용자가 단일 위치에서 모든 AWS 리소스로의 사용자 액세스를 관리할 수 있습니다.	April 24, 2017
서비스 연결 역할	서비스 링크된 역할은 AWS 서비스로 권한을 위임하는 더욱 쉽고 보안한 방법을 제공합니다.	April 19, 2017
정책 요약	새로운 정책 요약을 통해 IAM 정책의 권한을 더욱 손쉽게 이해할 수 있습니다.	March 23, 2017

AWS Glossary

For the latest AWS terminology, see the [AWS Glossary](#) in the AWS General Reference.