



사용 설명서

AWS 리소스 및 태그 편집기에 태그 지정



버전 1.0

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS 리소스 및 태그 편집기에 태그 지정: 사용 설명서

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 브랜드 디자인은 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon 계열사, 관련 업체 또는 Amazon의 지원 업체 여부에 상관없이 해당 소유자의 자산입니다.

Table of Contents

Tag Editor란?	1
태그 지정 방법	1
자세히 알아보기	2
모범 사례 및 전략	3
모범 사례	3
태그 이름 지정 모범 사례	3
일반적인 태깅 전략	5
태깅 범주	7
시작하기	9
사전 조건	10
에 가입 AWS 계정	10
관리자 액세스 권한이 있는 사용자 생성	10
리소스 만들기	12
권한 설정	12
개별 서비스에 대한 권한	12
Tag Editor 콘솔을 사용하는 데 필요한 권한	12
Tag Editor 사용에 대한 권한 부여	15
태그 기반 권한 부여 및 액세스 제어	16
태그를 지정할 리소스 찾기	18
선택한 리소스의 태그 확인 및 편집	19
결과를 .csv 파일로 내보내기	21
태그 관리	22
선택된 리소스에 태그 추가	22
선택된 리소스의 태그 편집	24
선택된 리소스에서 태그 제거	25
IAM 정책에서 태그 사용	27
태그 및 속성 기반 액세스 제어	27
태그 관련 조건 키	27
태그를 사용하는 IAM 정책 예	28
AWS Organizations 태그 정책	31
사전 조건 및 권한	31
태그 정책 준수 여부를 평가하기 위한 사전 조건	31
계정에 대한 규정 준수 평가 권한	32
조직 전체에 대한 규정 준수 평가 권한	33

보고서 저장에 대한 Amazon S3 버킷 정책	35
계정의 규정 준수 평가	36
조직 전체의 정책 준수 평가	38
태그 변경 사항 모니터링	41
태그 변경으로 EventBridge 이벤트 생성	41
Lambda 및 서버리스	43
모니터링 자습서	43
1단계. Lambda 함수 생성	44
2단계. 필요한 IAM 권한 설정	48
3단계. Lambda 함수의 예비 테스트 수행	49
4단계. 함수를 실행하는 EventBridge 규칙을 만듭니다.	52
5단계. 전체 솔루션을 테스트합니다.	53
자습서 요약	54
태그 변경 문제 해결	56
실패한 태그 변경 재시도	56
보안	57
데이터 보호	57
데이터 암호화	58
인터넷워크 트래픽 개인 정보	59
ID 및 액세스 관리	59
대상	60
ID를 통한 인증	60
정책을 사용하여 액세스 관리	61
Tag Editor가 IAM과 작동하는 방식	62
ID 기반 정책 예시	65
문제 해결	69
로깅 및 모니터링	70
CloudTrail 통합	71
규정 준수 확인	73
복원성	74
인프라 보안	74
Tag Editor Service Quotas	76
문서 기록	78
.....	lxxx

Tag Editor란?

Tag Editor를 사용하면 태그를 효과적으로 관리할 수 있습니다. 태그는 AWS 리소스를 구성하기 위한 메타데이터 역할을 하는 키 및 값 페어입니다. 대부분의 AWS 리소스에서는 리소스를 생성할 때 태그를 추가할 수 있습니다. 리소스의 예에는 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스, Amazon Simple Storage Service(S3) 버킷, AWS Secrets Manager의 시크릿 등이 있습니다.

Important

개인 식별 정보(PII)나 기타 기밀 정보 또는 민감한 정보를 태그에 저장하지 마세요. 당사는 태그를 사용하여 청구 및 관리 서비스를 제공합니다. 태그는 개인 데이터나 민감한 데이터에 사용하기 위한 것이 아닙니다.

태그를 사용하면 리소스를 손쉽게 관리, 식별, 정리, 검색 및 필터링할 수 있습니다. 태그를 생성하여 용도, 소유자, 환경 또는 기타 기준에 따라 리소스를 분류할 수 있습니다.

각 태그는 두 부분으로 구성됩니다.

- 태그 키(예: CostCenter, Environment 또는 Project) 태그 키는 대/소문자를 구별합니다.
- 태그 값(예: 111122223333 또는 Production). 태그 키처럼 태그 값은 대/소문자를 구별합니다.

Note

태그 키는 대소문자를 구분하지만, 대소문자만 다른 태그 키들의 적용을 방지하기 위해 IAM 리소스에 대한 추가 검증 기능이 IAM에 있습니다. 대소문자만 다른 키들은 사용하지 않는 것이 좋습니다. 자세한 내용은 [IAM 리소스에 대한 태그를 참조하세요](#).

리소스 태그 지정 방법

AWS 리소스에 태그를 추가하는 세 가지 방법이 있습니다.

- AWS 서비스 API 작업 -가 직접 지원하는 태깅 API 작업입니다 AWS 서비스. 각 AWS 서비스 에서 제공하는 태그 지정 기능을 알아보려면 설명서 [AWS 인덱스](#)의 서비스 설명서를 참조하세요.
- Tag Editor 콘솔 - 일부 서비스는 Tag Editor 콘솔을 통한 태그 지정 기능을 지원합니다.

- 리소스 그룹 태그 지정 API - 대부분의 서비스는 [AWS Resource Groups Tagging API](#)를 사용하는 태그 지정 기능을 지원합니다.

Note

또한 [AWS Service Catalog TagOptions Library](#)를 사용하여 프로비저닝된 제품의 태그를 쉽게 관리할 수 있습니다. TagOption은 Service Catalog에서 관리되는 키 값 페어입니다. AWS 태그는 아니지만 TagOption을 기반으로 AWS 태그를 생성하기 위한 템플릿 역할을 합니다.

AWS에서 비용이 발생하는 모든 서비스의 리소스에 태깅할 수 있습니다. 다음 서비스의 경우는 고객 사용 사례를 더 잘 충족하기 위해 태그 지정 AWS 서비스 을 지원하는 최신 대안을 AWS 권장합니다.

Amazon Cloud Directory	Amazon CloudSearch	Amazon Cognito Sync
AWS Data Pipeline	Amazon Elastic Transcoder	Amazon Machine Learning
AWS OpsWorks Stacks	Amazon Glacier 다이렉트	Amazon SimpleDB
Amazon WorkSpaces Application Manager	AWS DeepLens	

자세히 알아보기

이 페이지에서는 AWS 리소스 태그 지정에 대한 일반적인 정보를 제공합니다. 특정 AWS 서비스의 리소스 태그 지정에 대한 자세한 내용은 해당 설명서를 참조하세요. 다음은 태깅에 대한 정보의 유용한 출처이기도 합니다.

- 에 대한 자세한 AWS Resource Groups Tagging API내용은 [Resource Groups Tagging API 참조 안내서](#)를 참조하세요.
- 각에서 AWS 서비스 제공하는 태그 지정 기능에 대한 자세한 내용은 설명서 [AWS 인덱스](#)의 서비스 설명서를 참조하세요.
- IAM 정책에서 태그를 사용하여 AWS 리소스를 보고 상호 작용할 수 있는 사용자를 제어하는 방법에 대한 자세한 내용은 [IAM 사용 설명서의 태그를 사용하여 IAM 사용자 및 역할에 대한 액세스 제어를](#) 참조하세요.

모범 사례 및 전략

이러한 섹션에서는 AWS 리소스에 태그를 지정하고 Tag Editor를 사용할 때의 모범 사례와 전략에 대한 정보를 제공합니다.

태그 지정 모범 사례

AWS 리소스에 대한 태그 지정 전략을 생성할 때 모범 사례를 따르세요.

- 개인 식별 정보(PII)나 기타 기밀 정보 또는 민감한 정보를 태그에 추가하지 않습니다. 태그는 결제를 포함한 많은 AWS 서비스에서 액세스할 수 있습니다. 태그는 프라이빗 또는 민감한 데이터에 사용하기 위한 것이 아닙니다.
- 대/소문자를 구분하는 표준화된 태그 형식을 사용하고 모든 리소스 유형에 일관되게 적용합니다.
- 리소스 액세스 제어, 비용 추적, 자동화 및 조직 관리와 같은 다양한 용도를 지원하는 태그 지침을 고려합니다.
- 리소스 태그를 관리하는 데 도움이 되는 자동화 도구를 사용합니다. Tag Editor 및 [리소스 그룹 API 태그 지정](#)을 사용하면 태그를 프로그래밍 방식으로 제어하여 태그와 리소스를 더 쉽게 자동으로 관리, 검색 및 필터링할 수 있습니다.
- 태그를 너무 적게 사용하는 것보다는 너무 많이 사용하는 편이 낫습니다.
- 변화하는 비즈니스 요구 사항에 맞춰 태그를 변경하는 것은 쉽지만 향후 변경에 따른 결과를 고려해야 합니다. 예를 들어 액세스 제어 태그를 변경하는 경우 해당 태그를 참조하며 리소스에 대한 액세스를 제어하는 정책도 업데이트해야 합니다.
- AWS Organizations를 사용하여 태그 정책을 생성하고 배포하여 조직에서 도입하기로 선택한 태깅 표준을 자동으로 적용할 수 있습니다. 태그 정책을 사용하면 각 키에 유효한 키 이름과 값을 정의하는 태깅 규칙을 지정할 수 있습니다. 기존 태그를 평가하고 정리할 기회를 얻기 위해 모니터링만 하도록 선택할 수 있습니다. 태그가 선택한 표준에 부합하면 태그 정책에서 적용을 활성화하여 표준에 부합하지 않는 태그가 생성되지 않도록 할 수 있습니다. 자세한 내용은 AWS Organizations 사용 설명서의 [태그 정책](#)을 참조하세요.

태그 이름 지정 모범 사례

다음은 태그와 함께 사용하는 것이 권장되는 몇 가지 모범 사례 및 이름 지정 규칙입니다. 자세한 내용은 IAM 사용 설명서의 [태그 이름 지정](#)을 참조하세요.

여러 태그가에서 사전 정의 AWS 되거나 다양한에서 자동으로 생성됩니다 AWS 서비스. 많은 AWS 생성된 태그는 이름에서 단어를 구분하는 하이픈과 함께 모두 소문자를 사용하는 키 이름과 태그의 소스

서비스를 식별하기 위해 접두사를 사용하고 그 뒤에 콜론이 따르는 키 이름을 사용합니다. 예를 들어, 다음을 참조하세요.

- `aws:ec2spot:fleet-request-id`는 인스턴스를 시작한 Amazon EC2 스팟 인스턴스 요청을 식별하는 태그입니다.
- `aws:cloudformation:stack-name`은 리소스를 생성한 CloudFormation 스택을 식별하는 태그입니다.
- `elasticbeanstalk:environment-name`은 리소스를 생성한 애플리케이션을 식별하는 태그입니다.

다음 규칙을 사용하여 태그 이름을 지정하는 것이 좋습니다.

- 단어는 모두 소문자를 사용합니다.
- 하이픈을 사용하여 단어를 구분합니다.
- 접두사 뒤에 콜론을 사용하여 조직 이름이나 약어를 구분합니다.

예를 들어, 이름이 AnyCompany인 가상의 회사에 대해 다음과 같은 태그를 정의할 수 있습니다.

- 내부 비용 센터 코드를 식별하는 `anycompany:cost-center`.
- 환경이 개발, 테스트 또는 프로덕션인지 식별하는 `anycompany:environment-type`.
- 리소스가 생성된 애플리케이션을 식별하는 `anycompany:application-id`.

접두사는 또는 사용 중인 AWS 타사 도구가 아닌 조직에서 정의한 대로 태그를 명확하게 인식할 수 있도록 합니다. 구분 기호에 하이픈과 함께 모두 소문자를 사용하면 태그 이름을 대문자로 표시하는 방법에 대한 혼동을 피할 수 있습니다. 예를 들어, `anycompany:project-id`는 `ANYCOMPANY:ProjectID`, `anycompany:projectID` 또는 `Anycompany:ProjectId`보다 간단하게 기억할 수 있습니다.

태그 이름 지정 제한 및 요구 사항

태그에 적용되는 기본 이름 지정 및 사용 요구 사항은 다음과 같습니다.

- 각 리소스에는 최대 50개 사용자 생성 태그가 포함될 수 있습니다.
- `aws:`로 시작하는 시스템 생성 태그는 AWS 용으로 예약되어 있으며 이 제한이 적용되지 않습니다. `aws:` 접두사로 시작하는 태그를 편집하거나 삭제할 수 없습니다.
- 각 리소스에 대해 각 태그 키는 고유해야 하며 각 태그 키는 하나의 값만 가질 수 있습니다.

- 태그 키는 UTF-8 형식의 최소 1자에서 최대 128자여야 합니다.
- 태그 값은 최소 0자, 최대 256자의 UTF-8 형식 유니코드 문자로 지정해야 합니다.
- 허용되는 문자는 AWS 서비스에 따라 다를 수 있습니다. 특정 AWS 서비스의 리소스에 태그를 지정하는 데 사용할 수 있는 문자에 대한 자세한 내용은 설명서를 참조하세요. 일반적으로 허용되는 문자는 UTF-8로 표현할 수 있는 문자, 숫자, 공백 및 `_ . : / = + - @` 문자도 있습니다.
- 태그 키와 값은 대소문자를 구분합니다. 태그를 대문자로 사용하는 전략을 세우고 이러한 전략을 모든 리소스 타입에 대해 일관되게 구현하는 것이 가장 좋습니다. 예컨대, `Costcenter`, `costcenter` 또는 `CostCenter`를 사용할지 결정하고 모든 태그에 대해 동일한 규칙을 사용합니다. 대/소문자가 일치하지 않는 유사한 태그를 사용하지 마십시오.

일반적인 태깅 전략

다음 태깅 전략을 사용하면 AWS 리소스를 식별하고 관리하는 데 도움이 됩니다.

내용

- [리소스 정리용 태그](#)
- [비용 할당용 태그](#)
- [자동화용 태그](#)
- [액세스 제어용 태그](#)
- [태깅 거버넌스](#)

리소스 정리용 태그

태그는에서 AWS 리소스를 구성하는 좋은 방법입니다 AWS Management Console. 리소스와 함께 표시되도록 태그를 구성하고 태그로 리소스를 검색 및 필터링할 수 있습니다. AWS Resource Groups 서비스를 사용하면 하나 이상의 태그 또는 태그 일부를 기반으로 AWS 리소스 그룹을 생성할 수 있습니다. AWS CloudFormation 스택에서의 발생에 따라 그룹을 생성할 수도 있습니다. 리소스 그룹 및 Tag Editor를 사용하면 여러 서비스, 리소스 및 리전으로 구성된 애플리케이션의 데이터를 한 곳에 통합하여 볼 수 있습니다.

비용 할당용 태그

AWS Cost Explorer 및 세부 결제 보고서를 사용하면 태그를 기준으로 AWS 비용을 분류할 수 있습니다. 일반적으로 비용 센터/비즈니스 단위, 고객 또는 프로젝트와 같은 비즈니스 태그를 사용하여 AWS 비용을 기존 비용 할당 차원과 연결합니다. 하지만 비용 할당 보고서는 어떤 태그든 포함할 수 있습니다.

다. 따라서 비용을 특정 애플리케이션, 환경 또는 규정 준수 프로그램과 같은 기술이나 보안 차원에 연결할 수 있습니다.

일부 서비스의 경우 비용 할당 목적으로 AWS생성된 `createdBy` 태그를 사용하여 분류되지 않을 수 있는 리소스를 설명할 수 있습니다. `createdBy` 태그는 지원되는 AWS 서비스 및 리소스에만 사용할 수 있습니다. 이 태그 값에는 특정 API 또는 콘솔 이벤트와 관련된 데이터가 포함되어 있습니다. 자세한 내용은 AWS 결제 및 비용 관리 사용 설명서의 [AWS에서 생성되는 비용 할당 태그](#)를 참조하세요.

자동화용 태그

리소스 또는 서비스별 태그는 자동화 작업 중에 리소스를 필터링하는 데 종종 사용됩니다. 자동화 태그는 자동화된 작업을 옵트인 또는 옵트아웃하거나 아카이브, 업데이트 또는 삭제할 리소스의 특정 버전을 식별하는 데 사용됩니다. 예를 들어, 비용을 절감하기 위해 업무 외 시간에 개발 환경의 가동을 중단하는 자동화된 `start` 또는 `stop` 스크립트를 실행할 수 있습니다. 이 경우 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 태그를 사용하면 이 작업에서 옵트아웃할 인스턴스를 간단하게 식별할 수 있습니다. 오래된 스냅샷, 기한 경과된 스냅샷 또는 롤링 Amazon EBS 스냅샷을 찾아서 삭제하는 스크립트의 경우 스냅샷 태그로 검색 조건의 차원을 추가할 수 있습니다.

액세스 제어용 태그

IAM 정책은 태그 기반 조건을 지원하므로 특정 태그 또는 태그 값에 따라 IAM 권한을 제한할 수 있습니다. 예를 들어 태그에 기반하여 EC2 API 호출을 특정 환경(예: 개발, 테스트 또는 프로덕션)으로 제한하는 조건을 IAM 사용자 또는 역할 권한에 포함할 수 있습니다. 동일한 전략을 사용하여 API 호출을 특정 Amazon Virtual Private Cloud(VPC) 네트워크로 제한할 수 있습니다. 태그 기반의 리소스 수준 IAM 권한에 대한 지원은 서비스별로 다릅니다. 액세스 제어에 태그 기반 조건을 사용하는 경우 태그를 수정할 수 있는 사용자를 정의하고 제한해야 합니다. AWS 리소스에 대한 API 액세스 제어를 위한 태그 사용에 관한 자세한 내용은 IAM 사용 설명서의 [AWS IAM으로 작업하는 서비스](#)를 참조하세요.

태깅 거버넌스

효과적인 태그 지정 전략은 표준화된 태그를 사용하고 AWS 리소스 전체에 일관되고 프로그래밍 방식으로 적용합니다. AWS 환경에서 태그를 관리하기 위해 사후 대응 및 사전 예방 접근 방식을 모두 사용할 수 있습니다.

- 대응형 거버넌스는 Resource Groups 태그 지정 API AWS Config 규칙 및 사용자 지정 스크립트와 같은 도구를 사용하여 제대로 태그 지정되지 않은 리소스를 찾는 것입니다. 리소스를 수동으로 찾으려면 Tag Editor와 세부 결제 보고서를 사용할 수 있습니다.

- 사전 예방적 거버넌스는 CloudFormation, Service Catalog,의 태그 정책 AWS Organizations 또는 IAM 리소스 수준 권한과 같은 도구를 사용하여 리소스 생성 시 표준화된 태그가 일관되게 적용되도록 합니다.

예를 들어 속성을 사용하여 CloudFormation Resource Tags 리소스 유형에 태그를 적용할 수 있습니다. 서비스 카탈로그에서는 제품을 시작할 때 자동으로 결합되고 적용되는 포트폴리오 및 제품 태그를 추가할 수 있습니다. 보다 엄격한 형태의 사전 예방적 거버넌스 방식에는 자동화된 작업이 포함됩니다. 예를 들어 Resource Groups Tagging API를 사용하여 AWS 환경의 태그를 검색하거나 스크립트를 실행하여 태그가 잘못 지정된 리소스를 격리 또는 삭제할 수 있습니다.

태깅 범주

일반적으로 태그를 효과적으로 사용하는 기업은 기술, 비즈니스 및 보안 차원에서 리소스를 정리할 수 있도록 비즈니스 관련 태그 그룹을 생성합니다. 또한 자동화된 프로세스를 사용하여 인프라를 관리하는 기업에서는 자동화 관련 태그를 추가로 사용합니다.

기술 태그	자동화용 태그	비즈니스 태그	보안 태그
<ul style="list-style-type: none"> • 이름 - 개별 리소스 식별 • 애플리케이션 ID - 특정 애플리케이션과 관련된 리소스 식별 • 애플리케이션 역할 - 특정 리소스(예: 웹 서버, 메시지 브로커, 데이터베이스)의 기능 설명 • 클러스터 - 공통 구성을 공유하고 애플리케이션에 대해 특정 기능을 수행하는 리소스 팜 식별 	<ul style="list-style-type: none"> • 날짜/시간 - 리소스를 시작, 중지, 삭제 또는 교체해야 하는 날짜 또는 시간 식별 • 옵트인/옵트아웃 - 인스턴스 시작, 중지 또는 크기 조정과 같은 자동화된 작업에 리소스를 포함할지 여부 지정 • 보안 - 암호화 또는 Amazon VPC 흐름 로그 활성화와 같은 요구 사항 결정, 추가 조사가 필요한 라우팅 테이블 또는 보안 그룹 식별 	<ul style="list-style-type: none"> • 프로젝트 - 리소스가 지원하는 프로젝트 식별 • 소유자 - 리소스에 대한 책임을 지는 사용자 식별 • 비용 센터/비즈니스 단위 - 대개 비용 할당 및 추적을 위해 리소스와 연관된 비용 센터 또는 비즈니스 단위 식별 • 고객 - 특정 리소스 그룹이 제공되는 특정 고객 식별 	<ul style="list-style-type: none"> • 기밀성 - 리소스가 지원하는 특정 데이터 기밀성 수준에 대한 식별자 • 규정 준수 - 특정 규정 준수 요구 사항을 준수해야 하는 워크로드에 대한 식별자

기술 태그	자동화용 태그	비즈니스 태그	보안 태그
<ul style="list-style-type: none">• 환경 - 개발, 테스트, 프로덕션 리소스 간 구별• 버전 - 리소스 또는 애플리케이션의 버전 구별			

Tag Editor 시작하기

Important

개인 식별 정보(PII)나 기타 기밀 정보 또는 민감한 정보를 태그에 저장하지 마세요. 당사는 태그를 사용하여 청구 및 관리 서비스를 제공합니다. 태그는 개인 데이터나 민감한 데이터에 사용하기 위한 것이 아닙니다.

여러 리소스에 태그를 동시에 추가하거나 여러 리소스의 태그를 동시에 편집 또는 삭제하려면 Tag Editor를 사용합니다. Tag Editor에서는 태그를 관리할 리소스를 검색한 후 검색 결과에 나온 리소스에서 바로 태그를 관리합니다.

Tag Editor를 시작하려면

1. [AWS Management Console](#)에 로그인합니다.
2. 다음 단계 중 하나를 수행하세요.
 - 서비스를 선택합니다. 그런 다음 관리 및 거버넌스에서 리소스 그룹 및 Tag Editor를 선택합니다. 왼쪽 탐색 창에서 Tag Editor를 선택합니다.
 - 바로가기 링크: [AWS Tag Editor 콘솔](#).

일부 리소스에는 태그를 적용할 수 없습니다. Tag Editor에서 지원하는 리소스에 대한 자세한 내용은 AWS Resource Groups 사용 설명서의 [지원되는 리소스 유형](#)에 있는 Tag Editor 태그 지정을 참조하십시오. 태그를 지정할 리소스 유형이 지원되지 않는 경우 콘솔 창의 왼쪽 하단 모서리에서 피드백을 선택하여 AWS 알려줍니다.

리소스 태그를 지정하는 데 필요한 권한과 역할은 [권한 설정](#) 단원을 참조하십시오.

주제

- [Tag Editor를 사용하기 위한 사전 조건](#)
- [권한 설정](#)

Tag Editor를 사용하기 위한 사전 조건

리소스에 대한 태그 지정 작업을 시작하려면 먼저, 기존 리소스에 대해 유효한 AWS 계정 이 있어야 하고, 리소스에 태그를 지정하고 그룹을 만들 수 있는 적절한 권한이 있어야 합니다.

주제

- [에 가입 AWS 계정](#)
- [관리자 액세스 권한이 있는 사용자 생성](#)
- [리소스 만들기](#)

에 가입 AWS 계정

이 없는 경우 다음 단계를 AWS 계정완료하여 생성합니다.

에 가입하려면 AWS 계정

1. <https://portal.aws.amazon.com/billing/signup>을 엽니다.
2. 온라인 지시 사항을 따르세요.

등록 절차 중 전화 또는 텍스트 메시지를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

에 가입하면 AWS 계정AWS 계정 루트 사용자인 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스에 액세스할 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 [루트 사용자 액세스 권한이 필요한 작업](#)을 수행하는 것입니다.

AWS 는 가입 프로세스가 완료된 후 확인 이메일을 보냅니다. 언제든지 <https://aws.amazon.com/>으로 이동하고 내 계정을 선택하여 현재 계정 활동을 확인하고 계정을 관리할 수 있습니다.

관리자 액세스 권한이 있는 사용자 생성

에 가입한 후 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 AWS 계정보호 AWS IAM Identity Center, AWS 계정 루트 사용자활성화 및 생성합니다.

보안 AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 소유자 [AWS Management Console](#)로 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

루트 사용자를 사용하여 로그인하는 데 도움이 필요하다면 AWS Sign-In 사용 설명서의 [루트 사용자 로 로그인](#)을 참조하세요.

2. 루트 사용자의 다중 인증(MFA)을 활성화합니다.

지침은 IAM 사용 설명서의 [AWS 계정 루트 사용자\(콘솔\)에 대한 가상 MFA 디바이스 활성화를 참조하세요](#).

관리자 액세스 권한이 있는 사용자 생성

1. IAM Identity Center를 활성화합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [AWS IAM Identity Center 설정](#)을 참조하세요.

2. IAM Identity Center에서 사용자에게 관리 액세스 권한을 부여합니다.

를 자격 증명 소스 IAM Identity Center 디렉터리로 사용하는 방법에 대한 자습서는 사용 AWS IAM Identity Center 설명서의 [기본값으로 사용자 액세스 구성을 IAM Identity Center 디렉터리 참조하세요](#).

관리 액세스 권한이 있는 사용자로 로그인

- IAM Identity Center 사용자로 로그인하려면 IAM Identity Center 사용자를 생성할 때 이메일 주소로 전송된 로그인 URL을 사용합니다.

IAM Identity Center 사용자를 사용하여 로그인하는 데 도움이 필요하다면 [사용 설명서의 AWS 액세스 포털에 로그인](#)을 참조하세요. AWS Sign-In

추가 사용자에게 액세스 권한 할당

1. IAM Identity Center에서 최소 권한 적용 모범 사례를 따르는 권한 세트를 생성합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [Create a permission set](#)를 참조하세요.

2. 사용자를 그룹에 할당하고, 그룹에 Single Sign-On 액세스 권한을 할당합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [그룹 추가](#)를 참조하세요.

리소스 만들기

태그를 지정 AWS 계정 하려면 리소스가 있어야 합니다. 지원되는 리소스 유형에 대한 자세한 내용은 AWS Resource Groups 사용 설명서의 [지원되는 리소스 유형](#)에 있는 Tag Editor 태그 지정 열을 참조하십시오.

권한 설정

Tag Editor를 완전히 활용하려면 리소스에 태그를 지정하거나 혹은 리소스의 태그 키와 값을 볼 수 있는 추가 권한이 필요할 수 있습니다. 이러한 권한은 다음 범주로 분류됩니다.

- 개별 서비스의 리소스에 태그를 지정하고 리소스 그룹에 해당 리소스를 추가할 수 있도록 하기 위한 개별 리소스에 대한 권한.
- Tag Editor 콘솔을 사용하는 데 필요한 권한.

관리자인 경우 AWS Identity and Access Management (IAM) 서비스를 통해 정책을 생성하여 사용자에게 권한을 제공할 수 있습니다. 가장 먼저 IAM 역할, 사용자 또는 그룹을 만든 다음, 필요한 권한을 사용하여 정책을 적용합니다. IAM 정책을 만들고 연결하는 방법은 [정책 작업](#)을 참조하십시오.

개별 서비스에 대한 권한

Important

이 섹션에서는 다른 AWS 서비스 콘솔 및 APIs의 리소스에 태그를 지정하려는 경우 필요한 권한을 설명합니다.

리소스에 태그를 추가하려면 리소스가 속한 서비스에 필요한 권한이 있어야 합니다. 예를 들어, Amazon EC2 인스턴스에 태그를 지정하려면 [Amazon EC2CreateTags](#) 작업과 같은 해당 서비스 API에서 태그를 지정할 수 있는 권한이 있어야 합니다.

Tag Editor 콘솔을 사용하는 데 필요한 권한

Tag Editor 콘솔을 사용하여 리소스를 나열하고 태그를 지정하려면 IAM의 사용자 정책 문에 다음 권한을 추가해야 합니다. 에서 유지 관리 및 최신 상태로 유지하는 AWS 관리형 정책을 추가 AWS하거나 자체 사용자 지정 정책을 생성 및 유지 관리할 수 있습니다.

Tag Editor 권한에 AWS 관리형 정책 사용

Tag Editor는 사용자에게 사전 정의된 권한 세트를 제공하는 데 사용할 수 있는 다음과 같은 AWS 관리형 정책을 지원합니다. 생성한 다른 정책과 마찬가지로 이러한 관리형 정책을 모든 역할, 사용자 또는 그룹에 연결할 수 있습니다.

[ResourceGroupsandTagEditorReadOnlyAccess](#)

이 정책은 연결된 IAM 역할 또는 사용자에게 AWS Resource Groups 및 Tag Editor 모두에 대한 읽기 전용 작업을 호출할 수 있는 권한을 부여합니다. 리소스의 태그를 읽으려면 별도의 정책을 통해 해당 리소스에 대한 권한이 있어야 합니다. 다음 중요 참고 사항에서 자세히 알아보세요.

[ResourceGroupsandTagEditorFullAccess](#)

이 정책은 연결된 IAM 역할 또는 사용자에게 리소스 그룹 작업과 Tag Editor의 읽기 및 쓰기 태그 작업을 호출할 수 있는 권한을 부여합니다. 리소스의 태그를 읽거나 쓰려면 별도의 정책을 통해 해당 리소스에 대한 권한이 있어야 합니다. 다음 중요 참고 사항에서 자세히 알아보세요.

Important

이전 두 정책은 Tag Editor 작업을 호출하고 Tag Editor 콘솔을 사용할 수 있는 권한을 부여합니다. 하지만 작업을 호출할 수 있는 권한뿐만 아니라 액세스하려는 태그가 있는 특정 리소스에 대한 적절한 권한도 있어야 합니다. 태그에 해당 액세스 권한을 부여하려면 다음 정책 중 하나를 첨부해야 합니다.

- AWS 관리형 정책은 모든 서비스의 리소스에 대한 읽기 전용 작업에 권한을 [ReadOnlyAccess](#) 부여합니다. 새 정책이 제공 AWS 서비스 되면 AWS 자동으로 이 정책을 최신 상태로 유지합니다.
- 많은 서비스는 해당 서비스에서 제공하는 리소스로만 액세스를 제한하는 데 사용할 수 있는 서비스별 읽기 전용 AWS 관리형 정책을 제공합니다. 예를 들어, Amazon EC2는 [AmazonEC2ReadOnlyAccess](#)을 제공합니다.
- 사용자가 액세스하도록 허용하려는 몇 가지 서비스 및 리소스에 대한 읽기 전용 작업에만 액세스 권한을 부여하는 고유한 정책을 만들 수 있습니다. 이 정책은 허용 목록 전략 또는 거부 목록 전략을 사용합니다.

허용 목록 전략은 정책에서 명시적으로 허용할 때까지 기본적으로 액세스를 거부하는 전략을 활용합니다. 이에 따라, 다음 예시와 같은 정책을 사용할 수 있습니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "tag:*" ],
      "Resource": [
        "arn:aws:ec2:us-east-1:444455556666:*",
        "arn:aws:s3:::amzn-s3-demo-bucket2"
      ]
    }
  ]
}
```

또는 명시적으로 차단한 리소스를 제외한 모든 리소스에 대한 액세스를 허용하는 거부 목록 전략을 사용할 수도 있습니다. 이를 위해서는 액세스를 허용하는 관련 사용자에게 적용되는 별도의 정책이 필요합니다. 그러면 다음 예시 정책에서는 Amazon 리소스 이름 (ARN)에 있는 특정 리소스에 대한 액세스를 거부합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [ "tag:*" ],
      "Resource": [
        "arn:aws:ec2:us-east-1:123456789012:instance:*",
        "arn:aws:s3:::amzn-s3-demo-bucket3"
      ]
    }
  ]
}
```

Tag Editor 권한을 수동으로 추가하기

- tag:*(이 권한은 모든 Tag Editor 작업을 허용합니다. 대신 사용자가 수행할 수 있는 작업을 제한하려면 별표를 [특정 작업](#)이나 심표로 구분된 작업 목록으로 바꿀 수 있습니다.)
- tag:GetResources
- tag:TagResources
- tag:UntagResources
- tag:getTagKeys
- tag:getTagValues
- resource-explorer:*
- resource-groups:SearchResources
- resource-groups:ListResourceTypes

Note

resource-groups:SearchResources 권한을 사용하면 태그 키 또는 값을 사용하여 검색을 필터링할 때 Tag Editor가 리소스를 나열할 수 있습니다.

resource-explorer:ListResources 권한을 사용하면 검색 태그를 정의하지 않고 리소스를 검색할 때 Tag Editor가 리소스를 나열할 수 있습니다.


Tag Editor 사용에 대한 권한 부여

AWS Resource Groups 및 Tag Editor 사용에 대한 정책을 역할에 추가하려면 다음을 수행합니다.

1. [IAM 콘솔에서 역할 페이지](#)를 엽니다.
2. Tag Editor 권한을 부여하려는 역할을 찾습니다. 새 역할을 선택하여 역할의 요약 페이지를 엽니다.
3. 권한 탭에서 권한 추가를 선택합니다.
4. 기존 정책 직접 연결을 선택합니다.
5. 정책 생성을 선택합니다.
6. JSON 탭에 다음 정책 문을 붙여 넣습니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "tag:GetResources",
        "tag:TagResources",
        "tag:UntagResources",
        "tag:getTagKeys",
        "tag:getTagValues",
        "resource-explorer:*",
        "resource-groups:SearchResources",
        "resource-groups:ListResourceTypes"
      ],
      "Resource": "*"
    }
  ]
}
```

 Note

이 정책 설명은 Tag Editor 작업 수행에 대한 권한만 부여합니다.

7. 다음: 태그를 선택한 후 다음: 검토를 선택합니다.
8. 새 정책 이름 및 설명을 입력합니다. 예를 들어 **AWSTaggingAccess**입니다.
9. 정책 생성을 선택합니다.

이제 정책이 IAM에 저장되고 이 정책을 역할, 그룹 또는 사용자 등 다른 주체에게 연결할 수 있습니다. 주체에게 정책 추가에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 자격 증명 권한 추가 및 제거](#)를 참조하십시오.

태그 기반 권한 부여 및 액세스 제어

AWS 서비스 는 다음을 지원합니다.

- 작업 기반 정책 - 예를 들어, 사용자가 GetTagKeys 또는 GetTagValues 작업을 수행하도록 허용하지만 다른 작업은 허용하지 않는 정책을 생성할 수 있습니다.
- 정책의 리소스 수준 권한 - 많은 서비스에서 정책에서 개별 리소스를 지정하기 위한 [ARN](#) 사용을 지원합니다.
- 태그 기반 권한 부여 - 많은 서비스에서 정책의 조건에서 리소스 태그 사용을 지원합니다. 예를 들어, 사용자에게 사용자와 동일한 태그가 지정된 그룹에 대한 전체 액세스를 허용하는 정책을 생성할 수 있습니다. 자세한 내용은 AWS Identity and Access Management 사용 설명서의 [ABAC란 무엇입니까 AWS?](#)를 참조하세요.
- 임시 보안 인증 정보 - 사용자는 Tag Editor 작업을 허용하는 정책이 있는 역할을 수임할 수 있습니다.

Tag Editor는 서비스 연결 역할을 사용하지 않습니다.

Tag Editor가 AWS Identity and Access Management (IAM)와 통합되는 방법에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 다음 주제를 참조하세요.

- [AWS IAM으로 작업하는 서비스](#)
- [Tag Editor에 사용되는 작업, 리소스 및 조건 키](#)
- [정책을 사용한 AWS 리소스 액세스 제어](#)

태그를 지정할 리소스 찾기

Tag Editor를 사용하면 태깅에 사용할 수 있는 AWS 리전 있는 하나 이상의 리소스를 찾는 쿼리를 빌드할 수 있습니다. 최대 20개 개별 리소스 유형을 선택하거나 모든 리소스 유형에서 쿼리를 작성할 수 있습니다. 쿼리에는 이미 태그가 있는 리소스나 태그가 없는 리소스가 포함될 수 있습니다. 자세한 내용은 AWS Resource Groups 사용 설명서의 [지원되는 리소스 유형](#)에서 Tag Editor 태그 지정 열을 참조하십시오.

태그를 지정할 리소스를 찾은 후 Tag Editor를 사용하여 태그를 추가, 보기, 편집 또는 삭제할 수 있습니다.

태그를 지정할 리소스를 찾으려면

1. [Tag Editor 콘솔](#)을 엽니다.
2. (선택 사항) 태그를 지정할 리소스를 검색할 AWS 리전을 선택합니다. 기본적으로 현재 리전이 사용됩니다. 이 절차에서는 us-east-1 및 us-west-2를 선택합니다.
3. 리소스 유형 드롭다운 목록에서 하나 이상의 리소스 유형을 선택합니다. 한 번에 최대 20개 개별 리소스 유형에 대해 태그를 추가 또는 편집하거나 모든 리소스 유형을 선택할 수 있습니다. 이 절차에서는 AWS::EC2::Instance 및 AWS::S3::Bucket을 선택합니다.
4. (선택 사항) 태그 필드에 태그 키 또는 태그 키 및 값 페어를 입력하여 현재 AWS 리전의 리소스를 지정된 값으로 태그 지정된 리소스로만 제한합니다. 태그 키를 입력하면 현재 리전에서 일치하는 태그 키가 목록에 표시됩니다. 이 목록에서 태그 키를 선택할 수 있습니다. 기존 키와 일치하는 데 충분한 문자를 입력하면 Tag Editor가 태그 키를 자동 완성합니다. 태그 지정을 마쳤으면 추가를 선택하거나 Enter를 누릅니다. 이 예에서는 Stage라는 태그 키를 가진 리소스를 필터링합니다. 태그 값은 선택 사항이지만 쿼리 결과를 더욱 좁힐 수 있습니다. 태그를 추가하려면 추가를 선택합니다. 쿼리는 태그에 AND 연산자를 추가하여 지정한 리소스 유형 및 지정된 모든 태그와 일치하는 리소스만 쿼리를 통해 반환됩니다.

Note

Tag Editor 콘솔은 현재 와일드카드를 지원하지 않습니다.

태그 키에 대해 여러 값을 가진 리소스를 찾으려면 동일한 키를 가진 다른 태그를 쿼리에 추가하고 다른 값을 지정합니다. 결과에는 동일한 태그 키로 태그 지정되고 선택된 값을 가진 모든 리소스가 포함됩니다. 검색은 대/소문자를 구분합니다.

선택된 AWS 리전에 지정된 유형의 모든 리소스를 찾으려면 태그 상자를 비워 둡니다. 이 쿼리는 태그가 있는 리소스를 반환하고 태그가 없는 리소스를 포함합니다. 쿼리에서 태그를 제거하려면 태그의 레이블에서 X를 선택합니다.

태그가 있지만 값이 비어 있는 리소스를 찾으려면 (빈 값)을 선택합니다.

Note

지정된 태그로 리소스를 찾을 수 있기 전에 현재 AWS 리전에서 하나 이상의 지정된 유형의 리소스에 적용해야 합니다.

5. 쿼리가 준비되면 리소스 검색을 선택합니다. 결과는 리소스 검색 결과 영역에 테이블로 표시됩니다.

많은 수의 리소스를 필터링하려면 리소스의 이름의 일부와 같은 필터 텍스트를 Filter resources(리소스 필터링)에 입력합니다.

Note

하위 문자열을 사용하여 결과를 필터링할 수 있습니다.

6. (선택 사항) Tag Editor가 표시하는 열을 리소스 검색 결과에 구성하려면 리소스 검색 결과에서 기본 설정 기어 아이콘을 선택합니다.

기본 설정 페이지에서 검색 결과에 표시하려는 행의 수를 선택합니다. 테이블의 모든 텍스트를 보려면 줄 바꿈 체크박스를 선택합니다.

결과에 표시하려는 Tag Editor 열을 복사합니다. 검색 결과에 발생하는 각 태그 또는 검색 결과의 선택된 하위 집합의 열을 표시할 수 있습니다. 태그를 지정할 리소스를 찾은 후 언제든지 이와 같이 할 수 있습니다. 열을 활성화하려면 태그 옆에 있는 스위치 아이콘을 선택하고 끄기에서 켜기로 변경합니다.

표시되는 열과 표시된 행의 수의 구성이 완료되면 확인을 선택합니다.

선택한 리소스의 태그 확인 및 편집

Tag Editor는 태그를 지정할 리소스 찾기 쿼리의 결과에 있는 선택된 리소스에서 기존 태그를 보여줍니다.

이전 섹션에서 설명한 대로 태그 열을 활성화한 경우, 검색 결과에서 각 리소스에 대한 해당 태그의 현재 값을 볼 수 있습니다.

Note

이 항목에서는 개별 리소스의 태그를 편집하는 방법을 설명합니다. 또한 동시에 여러 리소스를 선택하여 태그를 일괄 편집할 수 있습니다. 자세한 내용은 [Tag Editor를 사용하여 태그 관리](#) 단원을 참조하십시오.

검색 결과 테이블에서 인라인으로 태그를 편집하는 방법

1. 편집하고자 하는 리소스에 대한 태그의 값을 선택합니다.

Note

- 선택한 리소스에 현재 선택한 키가 포함된 태그가 없는 경우, 값은 (태그 없음)으로 표시됩니다.
- 선택한 리소스에 선택한 키는 있지만 값이 없는 태그가 있는 경우, 값은 '—'로 표시됩니다.

2. 새 값을 입력하거나 이 태그가 있는 다른 리소스에 이미 있는 값 중에서 선택할 수 있습니다. 태그 제거를 선택하여 이 하나의 리소스에서 태그를 삭제할 수도 있습니다.

개별 리소스에 대한 모든 태그를 보려면

1. 태그를 지정할 리소스 찾기 쿼리의 결과에서 기존 태그를 보려는 리소스의 태그 수 열에서 숫자를 선택합니다. Tags(태그 수) 열에 대시가 포함된 리소스는 기존 태그를 가지고 있지 않습니다.
2. Resource tags(리소스 태그)에서 기존 태그를 봅니다. 태그 관리 페이지에서 태그를 변경하거나 제거할 때 선택한 리소스의 태그 관리를 선택하여 이 창을 열 수도 있습니다.

Note

최근에 리소스에 적용한 태그가 표시되지 않을 경우 브라우저 창을 새로 고쳐 보십시오.

결과를 .csv 파일로 내보내기

태그를 지정할 리소스 찾기 쿼리의 결과를 쉼표로 분리된 값(.csv) 파일로 내보낼 수 있습니다. .csv 파일에는 리소스 이름, 서비스, 리전, 리소스 ID, 총 태그 수, 모음에서 고유한 각 태그 키의 열이 포함됩니다. .csv 파일을 사용하면 조직 내 리소스의 태그 지정 전략을 개발하거나 리소스에 대해 태그 지정에 중첩이나 불일치가 있는지 확인할 수 있습니다.

1. Find resources to tag(태그를 지정할 리소스 찾기) 쿼리의 결과에서 Export resources to CSV(리소스를 CSV로 내보내기)를 선택합니다.
2. 브라우저에서 메시지가 표시되면 .csv 파일을 열기로 선택하거나 파일을 편리한 위치에 저장하기로 선택합니다.

Tag Editor를 사용하여 태그 관리

태그를 지정하려는 [리소스를 찾은](#) 후, 검색 결과의 일부 또는 전체에 대한 태그를 추가, 제거, 편집할 수 있습니다. Tag Editor에는 리소스에 연결된 모든 태그가 표시됩니다. 또한 해당 태그가 Tag Editor에서 추가되었는지, 리소스의 서비스 콘솔에서 추가되었는지, API를 사용하여 추가되었는지도 표시됩니다.

Important

개인 식별 정보(PII)나 기타 기밀 정보 또는 민감한 정보를 태그에 저장하지 마세요. 당사는 태그를 사용하여 청구 및 관리 서비스를 제공합니다. 태그는 개인 데이터나 민감한 데이터에 사용하기 위한 것이 아닙니다.

태그를 관리하는 다른 방법

이 주제에서는에서 Tag Editor를 사용하여 리소스에 태그를 지정하는 방법에 대해 설명합니다 AWS Management Console. 그러나 다음 도구를 사용하여 AWS 리소스의 태그를 관리할 수도 있습니다.

- AWS Command Line Interface (AWS CLI)의 [resourcegroupstaggingapi 명령](#)을 사용하여 셸 프롬프트에 명령을 입력하거나 스크립트로 작성할 수 있습니다.
- AWS Tools for PowerShell Core에서 [AWS Resource Groups API 태그 지정](#)을 사용하여 PowerShell 스크립트를 만들고 실행할 수 있습니다.
- [Python용 API 태그 지정](#) 또는 [Java용 API 태그 지정](#)과 같은 [리소스 그룹 API 태그 지정](#)을 사용하여 사용 가능한 모든 [AWS SDK](#)로 프로그램을 만들고 실행할 수 있습니다.

기존 태그의 추가, 제거, 편집 시 태그를 지정할 리소스 찾기 쿼리의 결과에 선택하는 리소스에 대해서만 태그가 변경됩니다. 태그를 관리할 리소스는 500개까지 선택할 수 있습니다.

선택된 리소스에 태그 추가

Tag Editor를 사용하여 태그를 지정할 리소스 찾기 쿼리의 결과에 있는 선택된 리소스에 태그를 추가할 수 있습니다.

Note

이 주제에서는 여러 리소스의 태그를 일괄 편집하는 방법을 설명합니다. 또한 개별 리소스에 대한 태그 값을 편집할 수 있습니다. 자세한 내용은 [선택한 리소스의 태그 확인 및 편집](#) 단원을 참조하십시오.

1. [Tag Editor 콘솔](#)을 열고 태그를 지정할 여러 리소스를 반환하는 쿼리를 제출합니다.
2. 태그를 지정할 리소스 찾기 쿼리의 결과 테이블에서 태그를 추가하려는 리소스 옆의 확인란을 선택합니다. 테이블 상단의 리소스 필터링에 텍스트 문자열을 입력하여 리소스의 이름, ID, 태그 키 또는 태그 값의 일부를 필터링합니다. 태그 열에서 결과의 리소스에 태그는 이미 적용되어 있습니다.
3. 하나 이상의 리소스에 대한 확인란을 선택한 다음 선택한 리소스 태그 관리를 선택합니다.
4. 태그 관리 페이지에서 선택한 리소스에 대한 태그를 봅니다. 원본 쿼리가 더 많은 리소스를 반환했지만 1단계에서 선택한 리소스에만 태그가 추가되고 있습니다. 태그 추가를 선택합니다.
5. 태그 키와 태그 값(선택 사항)을 입력합니다. 이 절차에서 태그 키 **Team**와 태그 값 **Development**을 추가합니다.

Note

리소스에는 최대 50개 사용자 적용 태그가 포함될 수 있습니다. 50개의 사용자 적용 태그에 접근하는 경우 리소스에 새 태그를 추가하지 못할 수 있습니다. AWS 생성된 태그는 50개 태그 제한에 적용되지 않습니다. 또한 태그 키는 선택한 리소스 내에서 고유해야 합니다. 선택한 리소스에 이미 존재하는 태그 키와 일치하는 키를 가진 새 태그를 추가할 수 없습니다.

6. 태그 추가를 마치면 변경 사항 검토 및 적용을 선택합니다.
7. 변경 사항을 수락할 경우 선택된 모든 것에 변경 사항 적용을 선택합니다.
8. 선택하는 리소스의 수에 따라 새 태그 적용에는 몇 분이 걸릴 수 있습니다. 동일한 브라우저 탭에서 페이지를 나가거나 다른 페이지를 열지 마십시오. 성공적으로 변경된 경우 페이지의 상단에 녹색 성공 배너가 표시됩니다. 계속하기 전에 성공 또는 실패 배너가 페이지에 표시될 때까지 기다립니다.

일부 또는 모든 리소스의 태그 변경이 실패한 경우 [태그 변경 문제 해결](#) 단원을 참조하십시오. 태그 변경 실패(예: 권한 부족)를 해결한 후 태그 변경이 실패한 리소스에 대해 태그 변경을 다시 시

도할 수 있습니다. 자세한 내용은 [the section called “실패한 태그 변경 재시도” 단원을 참조하십시오.](#)

선택된 리소스의 태그 편집

Tag Editor를 사용하여 [Find resources to tag\(태그를 지정할 리소스 찾기\)](#) 쿼리의 결과에 있는 선택된 리소스에서 기존 태그 값을 변경할 수 있습니다. 태그를 편집하면 태그 키가 동일한 선택된 모든 리소스에 대한 태그의 값이 변경됩니다. 태그 키 이름을 바꿀 수 없지만 태그를 삭제하고 새 태그 키를 만들어 태그 키를 대체할 수 있습니다. 이렇게 하면 선택한 리소스에 대한 해당 키가 있는 모든 태그가 삭제됩니다.

Important

개인 식별 정보(PII)나 기타 기밀 정보 또는 민감한 정보를 태그에 저장하지 마세요. 당사는 태그를 사용하여 청구 및 관리 서비스를 제공합니다. 태그는 개인 데이터나 민감한 데이터에 사용하기 위한 것이 아닙니다.

1. Find resources to tag(태그를 지정할 리소스 찾기) 쿼리의 결과에서 기존 태그를 변경하려는 리소스 옆의 확인란을 선택합니다. 리소스 필터링에 텍스트 문자열을 입력하여 리소스의 이름 또는 ID의 일부를 필터링합니다. 태그 열에서 결과의 리소스에 태그는 이미 적용되어 있습니다.
2. Manage tags of the selected resources(선택된 리소스의 태그 관리)를 선택합니다.
3. 태그 관리 페이지의 Edit tags of selected resources(선택된 리소스의 태그 편집)에서 선택한 리소스에 대한 태그를 봅니다. 원본 쿼리가 더 많은 리소스를 반환했을 수 있지만, 1단계에서 선택한 리소스에만 태그를 변경하는 것입니다.
4. 태그 값을 변경, 추가 또는 삭제합니다. 기존 태그에는 태그 키가 있어야 하지만 태그 값은 선택 사항입니다.

이 절차에서는 **Team** 태그의 값을 **QA**로 변경합니다.

선택한 리소스가 동일한 키에 대해 다른 값을 가지고 있는 경우, 선택된 리소스는 다른 태그 값을 가지고 있음이 태그 값 필드에 표시됩니다. 이 경우 상자에 커서를 두면 선택된 리소스에서 이 태그 키에 대한 사용 가능한 모든 값의 드롭다운 목록이 열립니다.

선택한 리소스에 원하는 태그 값이 있는 경우 원하는 태그 값을 입력하면 태그 값이 강조 표시됩니다. 예를 들어 선택한 리소스에 이미 태그 값 **QA**가 있는 경우 **Q**를 입력하면 해당 값이 강조 표시됩니다. 드롭다운 목록의 값은 리소스에 대해 일관된 태그 값을 유지합니다. 태그 값은 선택된 모든

리소스에서 변경됩니다. 이 예제에서 태그 값은 **Team** 태그 키가 있는 선택된 모든 리소스에 대해 **QA**로 변경됩니다. **Team** 태그가 없는 선택된 리소스에 대해 **QA** 값이 있는 **Team** 태그가 추가됩니다.

5. 태그 변경을 마치면 변경 사항 검토 및 적용을 선택합니다.
6. 변경 사항을 수락할 경우 선택된 모든 것에 변경 사항 적용을 선택합니다.
7. 선택한 리소스의 수에 따라 태그 편집에는 몇 분이 걸릴 수 있습니다. 동일한 브라우저 탭에서 페이지를 나가거나 다른 페이지를 열지 마십시오. 성공적으로 변경된 경우 페이지의 상단에 녹색 성공 배너가 표시됩니다. 계속하기 전에 성공 또는 실패 배너가 페이지에 표시될 때까지 기다립니다.

일부 또는 모든 리소스의 태그 변경이 실패한 경우 [태그 변경 문제 해결](#) 단원을 참조하십시오. 태그 변경 실패의 근본 원인(예: 권한 부족)을 해결한 후 태그 변경이 실패한 리소스에 대해 태그 변경을 다시 시도할 수 있습니다. 자세한 내용은 [the section called “실패한 태그 변경 재시도”](#) 단원을 참조하십시오.

선택된 리소스에서 태그 제거

Tag Editor를 사용하여 [Find resources to tag\(태그를 지정할 리소스 찾기\)](#) 쿼리의 결과에 있는 선택된 리소스에서 태그를 제거할 수 있습니다. 태그 제거의 경우 해당 태그가 있는 선택된 모든 리소스에서 해당 태그가 삭제됩니다. 태그 키를 편집할 수 없기 때문에 태그 키를 편집해야 할 경우 태그를 제거하고 새 태그로 대체할 수 있습니다. 이렇게 하면 선택한 리소스에 대한 해당 키가 있는 모든 태그가 삭제됩니다.

1. Find resources to tag(태그를 지정할 리소스 찾기) 쿼리의 결과에서 태그를 제거하려는 리소스 옆의 확인란을 선택합니다. 리소스 필터링에 텍스트 문자열을 입력하여 리소스의 이름 또는 ID의 일부를 필터링합니다.
2. Manage tags of the selected resources(선택된 리소스의 태그 관리)를 선택합니다.
3. 태그 관리 페이지의 Edit tags of selected resources(선택된 리소스의 태그 편집)에서 선택한 리소스에 대한 태그를 봅니다. 원본 쿼리가 더 많은 리소스를 반환했을 수 있지만, 1단계에서 선택한 리소스에만 태그를 변경하는 것입니다.
4. 삭제할 태그 옆의 태그 제거를 선택합니다. 이 절차에서는 **Team** 태그를 제거합니다.

Note

태그 제거를 선택하면 해당 태그가 있는 선택된 모든 리소스에서 태그가 제거됩니다.

5. Review and apply changes(변경 사항 검토 및 적용)를 선택합니다.

6. 확인 페이지에서 Apply changes to all selected(선택된 모든 것에 변경 사항 적용)를 선택합니다.
7. 선택한 리소스의 수에 따라 태그 제거에는 몇 분이 걸릴 수 있습니다. 동일한 브라우저 탭에서 페이지를 나가거나 다른 페이지를 열지 마십시오. 성공적으로 변경된 경우 페이지의 상단에 녹색 성공 배너가 표시됩니다. 계속하기 전에 성공 또는 실패 배너가 페이지에 표시될 때까지 기다립니다.

일부 또는 모든 리소스의 태그 변경이 실패한 경우 [태그 변경 문제 해결](#) 단원을 참조하십시오. 태그 변경 실패의 근본 원인(예: 권한 부족)을 해결한 후 태그 변경이 실패한 리소스에 대해 태그 변경을 다시 시도할 수 있습니다. 자세한 내용은 [the section called “실패한 태그 변경 재시도”](#) 단원을 참조하십시오.

IAM 권한 정책에서 태그 사용

[AWS Identity and Access Management \(IAM\)](#)은 AWS 리소스에 액세스할 수 있는 AWS 서비스 있는 사용자를 결정하는 권한 정책을 생성하고 관리하는 데 사용하는입니다. AWS 서비스에 액세스하거나 AWS 리소스를 읽거나 쓰려는 모든 시도는 IAM 정책에 의해 액세스가 제어됩니다.

이러한 정책을 통해 리소스에 대한 세분화된 액세스 권한을 부여할 수 있습니다. 이러한 액세스 권한을 세부적으로 조정하는 데 사용할 수 있는 기능 중 하나는 정책의 [Condition](#) 요소입니다. 이 요소를 사용하면 요청 진행 여부를 판단할 수 있는 요청과 일치하는 조건을 지정할 수 있습니다. Condition 요소로 확인할 수 있는 항목은 다음과 같습니다.

- 요청을 수행하는 사용자 또는 역할에 연결된 태그.
- 요청 대상인 리소스에 연결된 태그.

태그 및 속성 기반 액세스 제어

태그는 AWS 액세스 제어 전략의 중요한 부분일 수 있습니다. 속성 기반 액세스 제어(ABAC) 전략에서 태그를 속성으로 사용하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [태그를 사용하여 AWS 리소스에 대한 액세스 제어](#) 및 [태그를 사용하여 IAM 사용자 및 역할에 대한 액세스 제어를](#) 참조하세요.

[IAM 자습서: 사용 설명서의 태그를 기반으로 AWS 리소스에 액세스할 수 있는 권한 정의에서 태그를 사용하여 다양한 프로젝트 및 그룹에 액세스 권한을 부여하는 방법을 보여주는 포괄적인 자습서](#)가 있습니다. AWS Identity and Access Management

단일 로그인에 대해 SAML 기반 ID 제공업체(idP)를 사용하는 경우, 사용자에게 액세스를 제공하는 위임된 역할에 태그를 지정할 수 있습니다. 자세한 내용은 AWS Identity and Access Management 사용 설명서에서 [IAM 자습서: ABAC에 SAML 세션 태그 사용](#)을 참조하세요.

태그 관련 조건 키

다음 표에는 태그를 기반으로 액세스를 제어하기 위해 IAM 권한 정책에서 사용할 수 있는 조건 키가 나와 있습니다. 이 조건 키를 사용하여 다음을 수행할 수 있습니다.

- 작업을 호출하는 주체의 태그를 비교합니다.
- 작업에 제공된 태그를 파라미터로 비교합니다.
- 작업에서 액세스할 리소스에 첨부된 태그를 비교합니다.

조건 키 및 사용 방법에 대한 자세한 내용은 조건 키 이름 옆에 링크된 페이지를 참조하십시오.

조건 키 이름	설명
aws:PrincipalTag	요청한 보안 주체(IAM 역할 또는 사용자)에 연결된 태그를 정책에서 지정한 태그와 비교합니다.
aws:RequestTag	요청에서 파라미터로 전달된 태그 키 값 페어를 정책에서 지정한 태그 키 값 페어와 비교합니다.
aws:ResourceTag	정책에서 지정한 태그 키 값 페어를 리소스에 연결된 키 값 페어와 비교합니다.
aws:TagKeys	요청의 태그 키를 정책에서 지정한 키와 비교합니다.

태그를 사용하는 IAM 정책 예

Example에 1: 사용자가 리소스를 생성할 때 특정 태그를 첨부하도록 강제합니다.

다음 예에 있는 IAM 권한 정책에서는 IAM 정책의 태그를 생성하거나 수정하는 사용자에게 키 Owner와 함께 태그를 포함하도록 강제하는 방법을 설명합니다. 또한 정책에 태그의 값을 현재 호출 주체에 연결된 Owner 태그와 동일한 값으로 설정하도록 명시합니다. 이 전략이 효과를 발휘하려면 모든 주체에 Owner 태그가 지정되어 있어야 하며, 사용자가 해당 태그를 수정할 수 없어야 합니다. Owner 태그를 포함하지 않고 정책을 만들거나 수정하려고 하면 정책이 일치하지 않아 작업이 허용되지 않습니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "TagCustomerManagedPolicies",
      "Effect": "Allow",
      "Action": [
        "iam:CreatePolicy",
        "iam:TagPolicy"
      ],
      "Resource": "arn:aws:iam::123456789012:policy/*",
    }
  ]
}
```

```

        "Condition": {
            "StringEquals": {"aws:RequestTag/Owner": "${aws:PrincipalTag/
Owner}"}
        }
    }
]
}

```

Example에 2: 태그를 사용하여 리소스에 대한 액세스 권한을 해당 '소유자'로 제한합니다.

다음 예에 있는 IAM 권한 정책에서는 호출하는 주체가 인스턴스와 동일한 project 태그 값으로 태그가 지정된 경우에만 Amazon EC2 인스턴스 실행을 중지할 수 있습니다.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": [
        "ec2:StopInstances"
      ],
      "Resource": [
        "arn:aws:iam::123456789012:instance/*"
      ],
      "Condition": {
        "StringEquals": {"aws:ResourceTag/project": "${aws:PrincipalTag/
project}"}
      }
    }
  ]
}

```

이 예는 [ABAC\(속성 기반 액세스 제어\)](#)의 예입니다. IAM 정책을 사용하여 태그 기반 액세스 제어 전략을 구현하는 방법에 대한 자세한 내용 및 다른 예는 AWS Identity and Access Management 사용 설명서의 다음 주제를 참조하십시오.

- [태그를 사용하여 AWS 리소스에 대한 액세스 제어](#)

- [태그를 사용하여 IAM 사용자 및 역할에 대한 액세스 제어](#)
- [IAM 자습서: 태그를 기반으로 AWS 리소스에 액세스할 수 있는 권한 정의](#) - 여러 태그를 사용하여 다양한 프로젝트 및 그룹에 액세스 권한을 부여하는 방법을 보여줍니다.

AWS Organizations 태그 정책

[태그 정책](#)은 사용자가 AWS Organizations에서 생성하는 정책 유형입니다. 태그 정책을 사용하여 조직의 계정에 있는 리소스 전체에서 태그를 표준화할 수 있습니다. 태그 정책을 사용하려면 AWS Organizations 사용 설명서의 [태그 정책 시작하기](#)에 설명된 워크플로우를 따르는 것이 좋습니다. 해당 페이지에 나와 있듯이 권장 워크플로우에는 규정을 준수하지 않는 태그를 찾아 수정하는 작업도 포함됩니다. 이러한 작업은 태그 편집기 콘솔을 사용하여 수행합니다.

사전 조건 및 권한

Tag Editor에서 태그 정책의 규정 준수를 평가하려면 먼저 요구 사항을 충족하고 필요한 권한을 설정해야 합니다.

주제

- [태그 정책 준수 여부를 평가하기 위한 사전 조건](#)
- [계정에 대한 규정 준수 평가 권한](#)
- [조직 전체에 대한 규정 준수 평가 권한](#)
- [보고서 저장에 대한 Amazon S3 버킷 정책](#)

태그 정책 준수 여부를 평가하기 위한 사전 조건

태그 정책 준수 여부를 평가하려면 다음이 필요합니다.

- 먼저에서 기능을 활성화 AWS Organizations하고 태그 정책을 생성 및 연결해야 합니다. 자세한 정보는 AWS Organizations 사용 설명서에서 다음 페이지를 참조하세요.
 - [태그 정책 관리를 위한 사전 조건 및 권한](#)
 - [태그 정책 활성화](#)
 - [태그 정책 시작하기](#)
- [계정 리소스에서 규정을 준수하지 않는 태그를 찾으려면](#) 해당 계정의 로그인 보안 인증 정보와 [계정에 대한 규정 준수 평가 권한](#)에 나와 있는 권한이 필요합니다.
- [조직 전체의 규정 준수 여부를 평가하려면](#) 조직 관리 계정의 로그인 보안 인증 정보와 [조직 전체에 대한 규정 준수 평가 권한](#)에 나와 있는 권한이 필요합니다. AWS 리전 미국 동부(버지니아 북부)에서만 규정 준수 보고서를 요청할 수 있습니다.

계정에 대한 규정 준수 평가 권한

계정 리소스에서 규정을 준수하지 않는 태그를 찾으려면 다음 권한이 필요합니다.

- `organizations:DescribeEffectivePolicy` - 계정에 대한 유효 태그 정책의 내용을 가져옵니다.
- `tag:GetResources` - 첨부된 태그 정책을 준수하지 않는 리소스 목록을 가져옵니다.
- `tag:TagResources` - 태그를 추가하거나 업데이트합니다. 또한 태그를 만들려면 서비스별 권한이 필요합니다. 예를 들어, Amazon Elastic Compute Cloud (Amazon EC2)의 리소스에 태그를 지정하려면 `ec2:CreateTags`에 대한 권한이 필요합니다.
- `tag:UntagResources` - 태그를 제거합니다. 또한 태그를 제거하려면 서비스별 권한이 필요합니다. 예를 들어, Amazon EC2에서 리소스의 태그를 해제하려면 `ec2>DeleteTags`에 대한 권한이 필요합니다.

다음 예제 AWS Identity and Access Management (IAM) 정책은 계정의 태그 규정 준수를 평가할 수 있는 권한을 제공합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EvaluateAccountCompliance",
      "Effect": "Allow",
      "Action": [
        "organizations:DescribeEffectivePolicy",
        "tag:GetResources",
        "tag:TagResources",
        "tag:UntagResources"
      ],
      "Resource": "*"
    }
  ]
}
```

IAM 정책 및 권한에 대한 자세한 내용은 [IAM 사용 설명서](#)를 참조하세요.

조직 전체에 대한 규정 준수 평가 권한

조직 전체에서 태그 정책 준수 여부를 평가하려면 다음 권한이 필요합니다.

- `organizations:DescribeEffectivePolicy` - 조직, OU(조직 단위) 또는 계정에 연결된 태그 정책의 내용을 가져옵니다.
- `tag:GetComplianceSummary` - 조직 내 모든 계정에서 규정을 준수하지 않는 리소스 요약을 가져옵니다.
- `tag:StartReportCreation` - 가장 최근의 규정 준수 평가 결과를 파일로 내보냅니다. 조직 전체의 규정 준수는 48시간마다 평가됩니다.
- `tag:DescribeReportCreation` - 보고서 작성 상태를 확인합니다.
- `s3:ListAllMyBuckets` - 조직 전체의 규정 준수 보고서에 액세스하도록 지원합니다.
- `s3:GetBucketAcl` - 규정 준수 보고서를 수신하는 Amazon S3 버킷의 액세스 제어 목록(ACL)을 검사합니다.
- `s3:GetObject` - 서비스 소유 Amazon S3 버킷에서 규정 준수 보고서를 가져옵니다.
- `s3:PutObject` - 지정된 Amazon S3 버킷에 규정 준수 보고서를 배치합니다.

보고서가 전달되는 Amazon S3 버킷이 SSE-KMS를 통해 암호화된 경우 해당 버킷에 대한 `kms:GenerateDataKey` 권한도 있어야 합니다.

다음 예제 IAM 정책은 조직 전체의 규정 준수 여부를 평가할 수 있는 권한을 제공합니다. 각 `##` `###`를 사용자의 정보로 바꿉니다.

- `bucket_name` - Amazon S3 버킷 이름
- `organization_id` - 조직의 ID

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EvaluateAccountCompliance",
      "Effect": "Allow",
      "Action": [
        "organizations:DescribeEffectivePolicy",
```

```

        "tag:StartReportCreation",
        "tag:DescribeReportCreation",
        "tag:GetComplianceSummary",
        "s3:ListAllMyBuckets"
    ],
    "Resource": "*"
},
{
    "Sid": "GetBucketAclForReportDelivery",
    "Effect": "Allow",
    "Action": "s3:GetBucketAcl",
    "Resource": "arn:aws:s3:::bucket_name",
    "Condition": {
        "StringEquals": {
            "aws:CalledViaLast": "tagpolicies.tag.amazonaws.com"
        }
    }
},
{
    "Sid": "GetObjectForReportDelivery",
    "Effect": "Allow",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::*/tag-policy-compliance-reports/*",
    "Condition": {
        "StringEquals": {
            "aws:CalledViaLast": "tagpolicies.tag.amazonaws.com"
        }
    }
},
{
    "Sid": "PutObjectForReportDelivery",
    "Effect": "Allow",
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::bucket_name/AwsTagPolicies/organization_id/
*",
    "Condition": {
        "StringEquals": {
            "aws:CalledViaLast": "tagpolicies.tag.amazonaws.com"
        },
        "StringLike": {
            "s3:x-amz-copy-source": "*/tag-policy-compliance-reports/*"
        }
    }
}
}

```

```
    ]
  }
```

IAM 정책 및 권한에 대한 자세한 내용은 [IAM 사용 설명서](#)를 참조하세요.

보고서 저장에 대한 Amazon S3 버킷 정책

조직 전체의 규정 준수 보고서를 생성하려면 StartReportCreation API를 직접적으로 호출하는 데 사용하는 ID에서 미국 동부(버지니아 북부) 리전의 Amazon Simple Storage Service(Amazon S3) 버킷에 액세스하여 보고서를 저장해야 합니다. 태그 정책은 직접 호출 ID의 자격 증명을 사용하여 규정 준수 보고서를 지정된 버킷에 전달합니다.

StartReportCreation API를 직접적으로 호출하는 데 사용되는 버킷과 ID가 동일한 계정에 속하는 경우 이 사용 사례에 추가 Amazon S3 버킷 정책이 필요하지 않습니다.

StartReportCreation API를 직접적으로 호출하는 데 사용되는 ID와 연결된 계정이 Amazon S3 버킷을 소유한 계정과 다른 경우 다음 버킷 정책을 버킷에 연결해야 합니다. 각 **## ###**를 사용자의 정보로 바꿉니다.

- **bucket_name** - Amazon S3 버킷 이름
- **organization_id** - 조직의 ID
- **identity_ARN** - StartReportCreation API를 직접적으로 호출하는 데 사용되는 IAM ID의 ARN

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CrossAccountTagPolicyACL",
      "Effect": "Allow",
      "Principal": {
        "AWS": "identity_ARN"
      },
      "Action": "s3:GetBucketAcl",
      "Resource": "arn:aws:s3:::bucket_name"
    },
    {
```

```

        "Sid": "CrossAccountTagPolicyBucketDelivery",
        "Effect": "Allow",
        "Principal": {
            "AWS": "identity_ARN"
        },
        "Action": "s3:PutObject",
        "Resource": "arn:aws:s3::bucket_name/AwsTagPolicies/organization_id/"
    }
}

```

계정의 규정 준수 평가

유효 태그 정책을 통해 조직 내 계정의 규정 준수 여부를 평가할 수 있습니다.

Important

태그가 지정되지 않은 리소스는 결과에 정책 미준수로 나타나지 않습니다. 계정에서 태그가 지정되지 않은 리소스를 찾으려면 사용하는 쿼리와 AWS Resource Explorer 함께를 사용합니다 **tag:none**. 자세한 내용은 AWS Resource Explorer 사용 설명서의 [태그 없는 리소스 검색](#)을 참조하세요.

[유효 태그 정책](#)은 계정에 적용되는 태그 지정 규칙을 지정합니다. 계정이 상속하는 태그 정책과 계정에 직접 연결된 태그 정책을 집계한 것이 유효 태그 정책입니다. 태그 정책을 조직 루트에 연결하면 해당 태그 정책은 조직의 모든 계정에 적용됩니다. 태그 정책을 조직 단위(OU)에 연결하면 해당 정책은 OU에 속한 모든 계정 및 OU에 적용됩니다.

Note

아직 태그 정책을 만들지 않았다면 AWS Organizations 사용 설명서의 [태그 정책 시작하기](#)를 참조하세요.

규정을 준수하지 않는 태그를 찾으려면 다음 권한이 있어야 합니다.

- organizations:DescribeEffectivePolicy
- tag:GetResources

- tag:TagResources
- tag:UntagResources

유효 태그 정책을 사용하여 계정의 규정 준수 여부를 평가(콘솔)

1. 규정 준수를 확인하려는 계정에 로그인한 상태에서 [태그 정책 콘솔](#)을 엽니다.
2. 유효 태그 정책 섹션에는 정책이 마지막으로 업데이트된 시기와 정의된 태그 키가 표시됩니다. 태그 키를 확장하면 해당 값, 대소문자 처리 및 특정 리소스 유형에 대해 값이 적용되는지 여부에 대한 정보를 확인할 수 있습니다.

Note

관리 계정에 로그인한 경우 유효한 정책을 확인하고 규정 준수 정보를 보려면 계정을 선택해야 합니다.

3. 규정 미준수 태그가 있는 리소스 섹션에서 규정 미준수 태그를 검색할 AWS 리전 리소스를 지정합니다. 리소스 유형별로 검색할 수도 있습니다(선택 사항). 그런 다음 리소스 검색을 선택합니다.

실시간 결과는 검색 결과에 표시됩니다. 페이지당 반환되는 결과 수 또는 표시할 열을 변경하려면 설정 아이콘을 선택합니다.

4. 검색 결과에서 규정을 준수하지 않는 태그가 있는 리소스를 선택합니다.
5. 리소스 태그가 나열된 대화 상자에서 하이퍼링크를 선택하여 리소스가 생성된 AWS 서비스를 엽니다. 해당 콘솔에서 규정을 준수하지 않는 태그를 수정합니다.

Tip

어떤 태그가 규정을 준수하지 않는지 잘 모르는 경우, 태그 정책 콘솔에서 해당 계정의 유효 태그 정책 섹션으로 이동합니다. 태그 키를 확장하면 태그 지정 규칙을 볼 수 있습니다.

6. 관리하는 계정 리소스가 각 리전의 규정을 준수할 때까지 태그를 찾고 수정하는 프로세스를 반복합니다.

규정 미준수 태그를 찾으려면(AWS CLI, AWS API)

다음 명령과 작업을 사용하여 규정을 준수하지 않는 태그를 찾을 수 있습니다.

- AWS Command Line Interface (AWS CLI):

- [aws resourcegroupstaggingapi get-resources](#)
- [aws resourcegroupstaggingapi tag-resources](#)
- [aws resourcegroupstaggingapi untag-resources](#)

에서 태그 정책을 사용하는 전체 절차는 AWS Organizations 사용 설명서 [의에서 태그 정책 사용을 AWS CLI](#) AWS CLI 참조하세요.

- AWS Resource Groups Tagging API:
 - [GetResources](#)
 - [TagResources](#)
 - [UntagResources](#)

다음 단계

규정 준수 문제를 찾아 수정하는 프로세스를 반복하는 것이 좋습니다. 관리하는 계정의 리소스가 각 리전의 유효한 태그 정책을 준수할 때까지 계속합니다.

규정을 준수하지 않는 태그를 찾아 수정하는 작업은 다음과 같은 여러 가지 이유로 인해 반복적으로 이루어집니다.

- 조직의 태그 정책 사용은 시간이 지남에 따라 발전할 수 있습니다.
- 리소스를 만들 때 조직에 변화가 반영되는 데 시간이 걸립니다.
- 규정 준수는 새 리소스가 만들어지거나 리소스에 새 태그가 할당될 때마다 변경될 수 있습니다.
- 계정의 유효 태그 정책은 태그 정책이 계정에 연결되거나 분리될 때마다 업데이트됩니다. 계정이 상속하는 정책에 태그를 지정하기 위해 변경 사항이 발생할 때마다 유효 태그 정책도 업데이트됩니다.

조직에서 관리 계정으로 로그인한 경우 보고서를 작성할 수도 있습니다. 이 보고서에는 조직 계정에 태그가 지정된 모든 리소스에 대한 정보가 표시됩니다. 자세한 내용은 [조직 전체의 정책 준수 평가](#) 단원을 참조하십시오.

조직 전체의 정책 준수 평가

조직이 유효한 태그 정책을 준수하는지 평가할 수 있습니다. 조직의 계정에 있는 태그가 지정된 모든 리소스를 나열하고 각 리소스가 유효한 태그 정책을 준수하는지 여부를 나열하는 보고서를 생성할 수 있습니다.

⚠ Important

태그가 지정되지 않은 리소스는 결과에 정책 미준수로 나타나지 않습니다. 계정에서 태그가 지정되지 않은 리소스를 찾으려면 쿼리와 AWS Resource Explorer 함께를 사용합니다 **tag:none**. 자세한 내용은 AWS Resource Explorer 사용 설명서의 [태그 없는 리소스 검색](#)을 참조하세요.

에서만 조직의 관리 계정에서 보고서를 생성할 수 있습니다 us-east-1 AWS 리전 . 보고서를 생성하는 계정은 미국 동부(버지니아 북부) 리전의 Amazon S3 버킷에 대한 액세스 권한이 있어야 합니다. 버킷에는 [보고서 저장을 위한 Amazon S3 버킷 정책](#)에 표시된 것과 같은 연결된 버킷 정책이 있어야 합니다.

조직 전체에 대한 규정 준수 보고서를 생성하려면 다음 권한이 있어야 합니다.

- organizations:DescribeEffectivePolicy
- tag:GetComplianceSummary
- tag:StartReportCreation
- tag:DescribeReportCreation
- s3:ListAllMyBuckets
- s3:GetBucketAcl
- s3:GetObject
- s3:PutObject

이러한 권한을 표시하는 IAM 정책의 예는 [조직 전체에 대한 규정 준수 평가 권한](#)을 참조하세요.

조직 전체의 정책 준수 보고서를 생성하려면(콘솔)

1. [태그 정책 콘솔](#)을 엽니다.
2. 이 조직 루트 탭을 선택하고 페이지 하단에서 보고서 생성을 선택합니다.
3. 보고서 생성 화면에서 보고서를 저장할 위치를 지정합니다.
4. 내보내기 시작을 선택합니다.

보고서가 완성되면 조직 루트 탭의 미준수 보고서에서 다운로드할 수 있습니다.

i 참고

조직 전체의 규정 준수는 48시간마다 평가됩니다. 이 결과는 다음과 같습니다.

- 태그 정책 또는 리소스에 대한 변경 사항이 조직 전체의 정책 준수 보고서에 반영되려면 최대 48시간까지 걸릴 수 있습니다. 예를 들어, 한 리소스 유형에 대해 표준화된 새 태그를 정의하는 태그 정책이 있다고 가정합니다. 이 태그가 없는 해당 유형의 리소스는 최대 48시간 동안 보고서에서 정책을 준수하는 것으로 표시될 수 있습니다.
- 언제든지 보고서를 생성할 수 있지만 보고서 결과는 다음 평가가 완료될 때까지 업데이트되지 않습니다.
- NonCompliantKeys 열에는 리소스에서 유효한 태그 정책을 준수하지 않는 태그 키가 표시됩니다.
- KeysWithNonCompliantValues 열에는 리소스에서 대소문자 처리가 잘못되었거나 규정을 준수하지 않는 값을 가진 키가 표시됩니다.
- 조직의 멤버 AWS 계정였던를 닫으면 최대 90일 동안 태그 규정 준수 보고서에 계속 표시될 수 있습니다.

조직 전체의 규정 준수 보고서를 생성하려면(AWS CLI, AWS API)

다음 명령과 작업을 사용하여 조직 전체의 규정 준수 보고서를 생성하여 상태를 확인하고 보고서를 볼 수 있습니다.

- AWS Command Line Interface (AWS CLI):
 - [aws resourcegroupstaggingapi start-report-creation](#)
 - [aws resourcegroupstaggingapi describe-report-creation](#)
 - [aws resourcegroupstaggingapi get-compliance-summary](#)

에서 태그 정책을 사용하는 전체 절차는 AWS Organizations 사용 설명서 [의에서 태그 정책 사용을 AWS CLI](#) AWS CLI 참조하세요.

- AWS API:
 - [StartReportCreation](#)
 - [DescribeReportCreation](#)
 - [GetComplianceSummary](#)

서버리스 워크플로우 및 Amazon EventBridge로 태그 변경 사항 모니터링

Amazon EventBridge는 AWS 리소스에 대한 태그 변경을 지원합니다. 이 EventBridge 유형을 사용하면 태그 변경 사항과 일치하고 이벤트를 하나 이상의 대상으로 라우팅하는 EventBridge 규칙을 구축할 수 있습니다. 예를 들어 대상은 자동화된 워크플로를 호출하는 AWS Lambda 함수일 수 있습니다. 이 주제에서는 Lambda를 사용하여 AWS 리소스의 태그 변경을 안전하게 처리하는 비용 효율적인 서버리스 솔루션을 구축하기 위한 자습서를 제공합니다.

태그 변경으로 EventBridge 이벤트 생성

EventBridge는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다. 많은 AWS 리소스가 사용자 정의 사용자 정의 속성인 태그를 지원하여 AWS 리소스를 쉽게 구성하고 분류합니다. 태그의 일반적인 사용 사례로는 비용 할당 분류, 액세스 제어 보안, 자동화 등이 있습니다.

EventBridge를 사용하면 태그 변경 사항을 모니터링하고 AWS 리소스의 태그 상태를 추적할 수 있습니다. 이전에는 유사한 기능을 구현하기 위해 API를 지속적으로 폴링하고 여러 호출을 오케스트레이션했을 수 있습니다. 이제 개별 서비스 API, [Tag Editor](#), [태그 지정 API](#)를 포함하여 태그가 변경되면 리소스 이벤트에서 태그 변경이 시작됩니다. 다음 예는 태그 변경으로 인한 일반적인 EventBridge 이벤트입니다. 새 태그 키, 업데이트 또는 삭제된 태그 키와 관련 값이 나와 있습니다.

```
{
  "version": "0",
  "id": "bddcf1d6-0251-35a1-aab0-adc1fb47c11c",
  "detail-type": "Tag Change on Resource",
  "source": "aws.tag",
  "account": "123456789012",
  "time": "2018-09-18T20:41:38Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-00000000aaaaaaaa"
  ],
  "detail": {
    "changed-tag-keys": [
      "a-new-key",
      "an-updated-key",
      "a-deleted-key"
    ]
  }
}
```

```

    ],
    "tags": {
      "a-new-key": "tag-value-on-new-key-just-added",
      "an-updated-key": "tag-value-was-just-changed",
      "an-unchanged-key": "tag-value-still-the-same"
    },
    "service": "ec2",
    "resource-type": "instance",
    "version": 3,
  }
}

```

모든 EventBridge 이벤트에는 동일한 최상위 필드가 있습니다.

- 버전 - 기본적으로 이 값은 모든 이벤트에서 0(0)으로 설정됩니다.
- id - 모든 이벤트에 대해 고유 값이 생성됩니다. 규칙을 통해 대상으로 이동되어 처리되는 이벤트를 추적하는 데 도움이 될 수 있습니다.
- detail 유형 - source 필드를 함께 사용하여 detail 필드에 나타나는 필드와 값을 식별합니다.
- 소스 - 이벤트 소스인 서비스를 식별합니다. 태그 변경 소스는 `aws.tag`입니다.
- 시간 - 이벤트의 타임스탬프입니다.
- 리전 - 이벤트를 호출한 AWS 리전을 식별합니다.
- 리소스 - JSON 배열에는 이벤트에서 호출된 리소스를 식별하는 Amazon 리소스 이름(ARN)이 포함되어 있습니다. 이는 태그가 변경된 리소스입니다.
- detail - 이벤트 유형에 따라 콘텐츠가 달라지는 JSON 객체입니다. 리소스의 태그 변경 시 다음과 같은 세부 필드가 포함됩니다.
 - 변경된 태그 키 - 이 이벤트로 인해 변경된 태그 키입니다.
 - 서비스 - 리소스가 속한 서비스입니다. 이 예에서 서비스는 `ec2`이며, Amazon EC2입니다.
 - 리소스 유형 - 서비스의 리소스 유형입니다. 이 예제에서는 Amazon EC2 인스턴스입니다.
 - 버전 - 태그 세트의 버전입니다. 버전은 1에서 시작하여 태그가 변경될 때마다 증가합니다. 버전을 사용하여 태그 변경 이벤트의 순서를 확인할 수 있습니다.
 - 태그 - 변경 후 리소스에 지정된 태그입니다.

자세한 내용은 Amazon EventBridge 사용 설명서의 [Amazon EventBridge 이벤트 패턴](#)을 참조하세요.

EventBridge를 사용하면 다양한 필드를 기반으로 특정 이벤트 패턴과 일치하는 규칙을 만들 수 있습니다. 자습서에 이 작업을 수행하는 방법이 설명되어 있습니다. 또한 지정된 태그가 인스턴스에 첨부되지

않은 경우 Amazon EC2 인스턴스를 자동으로 중지하는 방법도 설명합니다. EventBridge 필드를 사용하여 Lambda 함수를 실행하는 인스턴스의 태그 이벤트와 일치하는 패턴을 생성합니다.

Lambda 및 서버리스

AWS Lambda 는 서버리스 패러다임을 따라 클라우드에서 코드를 실행합니다. 서버는 신경 쓰지 않고 필요할 때만 코드를 실행합니다. 사용한 컴퓨팅 시간에 대해서만 비용을 지불하면 됩니다. 서버리스라 하지만 서버가 없다는 의미는 아닙니다. 이 컨텍스트에서 서버리스는 코드를 실행하는 데 사용되는 서버를 프로비저닝, 구성 또는 관리할 필요가 없음을 의미합니다. 이 모든 작업을 자동으로 AWS 수행하므로 코드에 집중할 수 있습니다. Lambda에 대한 자세한 내용은 [AWS Lambda 제품 개요](#)를 참조하십시오.

자습서: 필수 태그가 누락된 Amazon EC2 인스턴스 자동 중지하기

관리하는 AWS 리소스 풀 AWS 계정 이 증가하면 태그를 사용하여 리소스를 더 쉽게 분류할 수 있습니다. 태그는 일반적으로 비용 할당 및 보안과 같은 중요한 사용 사례에 사용됩니다. AWS 리소스를 효과적으로 관리하려면 리소스에 일관되게 태그를 지정해야 합니다. 리소스가 프로비저닝될 때 적절한 태그를 모두 가져오는 경우가 많습니다. 하지만 이후 프로세스에서 태그가 변경되어 회사 태그 정책에서 벗어날 수 있습니다. 태그 변경 사항을 모니터링하면 태그 드리프트를 발견하고 즉시 대응할 수 있습니다. 이를 통해 리소스를 적절하게 분류하는 프로세스가 원하는 결과를 가져올 것이라는 확신을 가질 수 있습니다.

다음 예는 Amazon EC2 인스턴스에서 태그 변경을 모니터링하여 지정된 인스턴스에 필요한 태그가 계속 있는지 확인할 수 있는 방법을 설명합니다. 인스턴스의 태그가 변경되어 인스턴스에 더 이상 필요한 태그가 없는 경우, Lambda 함수를 호출하여 인스턴스가 자동으로 종료됩니다. 왜 이런 작업이 필요할까요? 기업 태그 정책에 따라 모든 리소스에 태그가 지정되면 효과적으로 비용을 할당할 수 있으며, [ABAC\(속성 기반 액세스 제어\)](#)를 기반으로 보안을 신뢰할 수 있습니다.

Important

중요한 인스턴스를 실수로 종료해서는 안 되는 비프로덕션 계정에서 이 자습서를 수행할 것을 강력히 권장합니다.

이 자습서의 예제 코드는 의도적으로 이 시나리오의 영향을 인스턴스 ID 목록에 있는 인스턴스로 제한합니다. 테스트를 위해 종료할 인스턴스 ID로 목록을 업데이트해야 합니다. 이렇게 하면의 리전에 있는 모든 인스턴스를 실수로 종료할 수 없습니다 AWS 계정.

테스트 후에는 회사의 태그 지정 전략에 따라 모든 인스턴스에 태그가 지정되었는지 확인합니다. 그런 다음, 함수를 목록에 있는 인스턴스 ID로 제한하는 코드를 제거하면 됩니다.

이 예에서는 JavaScript 및 16.x 버전의 Node.js를 사용합니다. 이 예제에서는 예제 AWS 계정 ID 123456789012과 AWS 리전 미국 동부(버지니아 북부)()를 사용합니다us-east-1. 이를 계정 ID과 리전으로 바꿉니다.

Note

콘솔이 기본값으로 다른 리전을 사용하는 경우, 콘솔을 변경할 때마다 이 자습서에서 사용 중인 리전을 변경해야 합니다. 이 자습서가 실패하는 일반적인 원인은 인스턴스와 함수를 서로 다른 두 리전에 두고 있기 때문입니다.

us-east-1와 다른 리전을 사용하는 경우, 다음 코드 예시에 있는 모든 참조를 선택한 리전으로 변경해야 합니다.

주제

- [1단계. Lambda 함수 생성](#)
- [2단계. 필요한 IAM 권한 설정](#)
- [3단계. Lambda 함수의 예비 테스트 수행](#)
- [4단계. 함수를 실행하는 EventBridge 규칙을 만듭니다.](#)
- [5단계. 전체 솔루션을 테스트합니다.](#)
- [자습서 요약](#)

1단계. Lambda 함수 생성

Lambda 함수를 만들려면

1. [AWS Lambda 관리 콘솔](#)을 엽니다.
2. 함수 생성을 선택한 다음 새로 작성을 선택합니다.
3. 함수 이름에 **AutoEC2Termination**을 입력합니다.
4. 런타임에는 Node.js 16.x를 선택합니다.
5. 다른 필드를 기본값으로 그대로 두고 함수 생성을 선택합니다.
6. AutoEC2Termination 세부 정보 페이지의 코드 탭에서 index.js 파일을 열어 코드를 확인합니다.
 - index.js 탭이 열려 있는 경우, 해당 탭의 편집 상자를 선택하여 코드를 편집할 수 있습니다.

- index.js 탭이 열려 있지 않은 경우, 탐색 창의 AutoEC2Terminator 폴더에 있는 index.js 파일에서 마우스 오른쪽 버튼을 클릭합니다. 그런 다음 열기를 선택합니다.

7. index.js 탭에서 편집 상자에 다음 코드를 붙여넣고 이미 있는 코드를 대체합니다.

값 RegionToMonitor을 이 함수를 실행할 리전으로 바꿉니다.

```
// Set the following line to specify which Region's instances you want to monitor
// Only instances in this Region are successfully stopped on a match

const RegionToMonitor = "us-east-1"

// Specify the instance ARNs to check.
// This limits the function for safety to avoid the tutorial shutting down all
instances in account
// The first ARN is a "dummy" that matches the test event you create in Step 3.
// Replace the second ARN with one that matches a real instance that you want to
monitor and that you can
// safely stop

const InstanceList = [
  "i-00000000aaaaaaaaaa",
  "i-05db4466d02744f07"
];

// The tag key name and value that marks a "valid" instance. Instances in the
previous list that
// do NOT have the following tag key and value are stopped by this function

const ValidKeyName = "valid-key";
const ValidKeyValue = "valid-value";

// Load and configure the AWS SDK
const AWS = require('aws-sdk');
// Set the AWS Region
AWS.config.update({region: RegionToMonitor});
// Create EC2 service object.
const ec2 = new AWS.EC2({apiVersion: '2016-11-15'});

exports.handler = (event, context, callback) => {

  // Retrieve the details of the reported event.
  var detail = event.detail;
  var tags = detail["tags"];
```

```
var service = detail["service"];
var resourceType = detail["resource-type"];
var resource = event.resources[0];
var resourceSplit = resource.split("/");
var instanceId = resourceSplit[resourceSplit.length - 1];

// If this event is not for an EC2 resource, then do nothing.
if (!(service === "ec2")) {
    console.log("Event not for correct service -- no action (" , service, ")" );
    return;
}

// If this event is not about an instance, then do nothing.
if (!(resourceType === "instance")) {
    console.log("Event not for correct resource type -- no action (" , resourceType,
    ")" );
    return;
}

// CAUTION - Removing the following 'if' statement causes the function to run
against
//          every EC2 instance in the specified Region in the calling AWS ##.
//          If you do this and an instance is not tagged with the approved tag
key
//          and value, this function stops that instance.

// If this event is not for the ARN of an instance in our include list, then do
nothing.
if (InstanceList.indexOf(instanceId)<0) {
    console.log("Event not for one of the monitored instances -- no action (" ,
resource, ")");
    return;
}

console.log("Tags changed on monitored EC2 instance (" ,instanceId,")");

// Check attached tags for expected tag key and value pair
if ( tags.hasOwnProperty(ValidKeyName) && tags[ValidKeyName] == "valid-value"){
    // Required tags ARE present
    console.log("The instance has the required tag key and value -- no action");
    callback(null, "no action");
    return;
}
```

```
// Required tags NOT present
console.log("This instance is missing the required tag key or value -- attempting
to stop the instance");

var params = {
  InstanceIds: [instanceId],
  DryRun: true
};

// call EC2 to stop the selected instances
ec2.stopInstances(params, function(err, data) {
  if (err && err.code === 'DryRunOperation') {
    // dryrun succeeded, so proceed with "real" stop operation
    params.DryRun = false;
    ec2.stopInstances(params, function(err, data) {
      if (err) {
        console.log("Failed to stop instance");
        callback(err, "fail");
      } else if (data) {
        console.log("Successfully stopped instance", data.StoppingInstances);
        callback(null, "Success");
      }
    });
  } else {
    console.log("Dryrun attempt failed");
    callback(err);
  }
});
};
```

8. 배포를 선택하여 변경 사항을 저장하고 새 버전의 함수를 활성화합니다.

이 Lambda 함수는 EventBridge의 태그 변경 이벤트에서 보고한 대로 Amazon EC2 인스턴스의 태그를 확인합니다. 이 예에서 이벤트의 인스턴스에 필수 태그 키 `valid-key`가 없거나 해당 태그에 값 `valid-value`가 없는 경우 함수는 인스턴스를 중지하려고 시도합니다. 특정 사용 사례에 맞게 이 논리적 검사 또는 태그 요구 사항을 변경할 수 있습니다.

브라우저에서 Lambda 콘솔 창을 열어 둡니다.

2단계. 필요한 IAM 권한 설정

함수를 성공적으로 실행하려면 먼저 함수에 EC2 인스턴스를 중지할 수 있는 권한을 부여해야 합니다. AWS 제공된 역할에 해당 권한이 [lambda_basic_execution](#) 없습니다. 이 자습서에서는 `AutoEC2Termination-role-uniqueid`라는 함수의 실행 역할에 첨부된 기본 IAM 권한 정책을 수정합니다. 이 자습서에 필요한 최소 추가 권한은 `ec2:StopInstances`입니다.

Amazon EC2별 IAM 정책을 생성하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [Amazon EC2: 프로그래밍 방식과 콘솔에서 EC2 인스턴스를 시작 또는 중지하고 보안 그룹 수정하기](#)를 참조하십시오.

IAM 권한 정책을 생성하여 Lambda 함수의 실행 역할에 연결하는 방법

1. 다른 브라우저 탭 또는 창에서 IAM 콘솔의 [역할](#) 페이지를 엽니다.
2. 역할 이름 **AutoEC2Termination**을 입력하고, 목록에 역할 이름이 나타나면 해당 역할 이름을 선택합니다.
3. 역할의 요약 페이지에서 권한 탭을 선택하고 이미 연결된 정책의 이름을 선택합니다.
4. 정책의 요약 페이지에서 정책 편집을 선택합니다.
5. 시각적 편집기 탭에서 추가 권한 추가를 선택합니다.
6. 서비스에서 EC2를 선택합니다.
7. 작업에서 인스턴스 중지를 선택합니다. 검색 창에 **Stop**를 입력한 다음 표시되는 `StopInstances`를 선택합니다.
8. 리소스의 경우, 모든 리소스를 선택하고 정책 검토를 선택한 다음 변경 사항 저장을 선택합니다.

그러면 새 버전의 정책이 자동으로 생성되고 이 버전이 기본으로 설정됩니다.

최종 파일은 다음 예제와 비슷할 것입니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "ec2:StopInstances",
      "Resource": "*"
    },
  ],
}
```

```

    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": "logs:CreateLogGroup",
      "Resource": "arn:aws:logs:us-east-1:123456789012:*"
    },
    {
      "Sid": "VisualEditor2",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/lambda/AutoEC2Termination:*"
    }
  ]
}

```

3단계. Lambda 함수의 예비 테스트 수행

이 단계에서는 함수에 테스트 이벤트를 제출합니다. Lambda 테스트 기능은 수동으로 제공된 테스트 이벤트를 제출하는 방식입니다. 함수는 이벤트가 EventBridge에서 발생한 것처럼 테스트 이벤트를 처리합니다. 값이 다른 여러 테스트 이벤트를 정의하여 코드의 다양한 부분을 모두 실행할 수 있습니다. 이 단계에서는 Amazon EC2 인스턴스의 태그가 변경되었으며 새 태그에 필수 태그 키와 값이 포함되어 있지 않음을 나타내는 테스트 이벤트를 제출합니다.

Lambda 함수를 테스트하는 방법

1. Lambda 콘솔이 있는 창 또는 탭으로 돌아가서 AutoEC2Termination 함수의 테스트 탭을 엽니다.
2. 새 역할 생성을 선택합니다.
3. 이벤트 이름에 **SampleBadTagChangeEvent**를 입력합니다.
4. Event JSON에서 텍스트를 다음 예제 텍스트에 표시된 샘플 이벤트로 변경합니다. 계정, 리전 또는 인스턴스 ID를 수정하지 않아도 이 테스트 이벤트는 제대로 작동합니다.

```

{
  "version": "0",
  "id": "bddcf1d6-0251-35a1-aab0-adc1fb47c11c",
  "detail-type": "Tag Change on Resource",
  "source": "aws.tag",

```

```

"account": "123456789012",
"time": "2018-09-18T20:41:38Z",
"region": "us-east-1",
"resources": [
  "arn:aws:ec2:us-east-1:123456789012:instance/i-00000000aaaaaaaa"
],
"detail": {
  "changed-tag-keys": [
    "valid-key"
  ],
  "tags": {
    "valid-key": "NOT-valid-value"
  },
  "service": "ec2",
  "resource-type": "instance",
  "version": 3
}
}

```

5. 저장를 선택한 다음 테스트를 선택합니다.

테스트가 실패한 것처럼 보이지만 괜찮습니다.

대응의 실행 결과 탭에 다음 오류가 표시됩니다.

```

{
  "errorType": "InvalidInstanceID.NotFound",
  "errorMessage": "The instance ID 'i-00000000aaaaaaaa' does not exist",
  ...
}

```

이 오류는 테스트 이벤트에 지정된 인스턴스가 존재하지 않기 때문에 발생합니다.

함수 로그의 실행 결과 탭에 있는 정보를 통해 Lambda 함수에서 EC2 인스턴스 종지를 성공적으로 시도했음을 알 수 있습니다. 하지만 코드에서 처음에 인스턴스를 중지하기 위해 [DryRun](#) 작업을 시도했으나 인스턴스 ID가 유효하지 않은 것으로 확인되어 실패했습니다.

```

START RequestId: 390c1f8d-0d9b-4b44-b087-8de64479ab44 Version: $LATEST
2022-11-30T20:17:30.427Z    390c1f8d-0d9b-4b44-b087-8de64479ab44    INFO    Tags
changed on monitored EC2 instance ( i-00000000aaaaaaaa )

```

```

2022-11-30T20:17:30.427Z    390c1f8d-0d9b-4b44-b087-8de64479ab44    INFO    This
instance is missing the required tag key or value -- attempting to stop the
instance
2022-11-30T20:17:31.206Z    390c1f8d-0d9b-4b44-b087-8de64479ab44    INFO    Dryrun
attempt failed
2022-11-30T20:17:31.207Z    390c1f8d-0d9b-4b44-b087-8de64479ab44    ERROR    Invoke
Error    {"errorType":"InvalidInstanceID.NotFound","errorMessage":"The instance
ID 'i-00000000aaaaaaaaaa' does not
exist","code":"InvalidInstanceID.NotFound","message":"The instance ID
'i-00000000aaaaaaaaaa' does not
exist","time":"2022-11-30T20:17:31.205Z","requestId":"a5192c3b-142d-4cec-
bdbc-685a9b7c7abf","statusCode":400,"retryable":false,"retryDelay":36.87870631147607,"stack
["InvalidInstanceID.NotFound: The instance ID 'i-00000000aaaaaaaaaa' does
not exist","    at Request.extractError (/var/runtime/node_modules/aws-sdk/
lib/services/ec2.js:50:35)","    at Request.callListeners (/var/runtime/
node_modules/aws-sdk/lib/sequential_executor.js:106:20)","    at Request.emit
(/var/runtime/node_modules/aws-sdk/lib/sequential_executor.js:78:10)","    at
Request.emit (/var/runtime/node_modules/aws-sdk/lib/request.js:686:14)","    at
Request.transition (/var/runtime/node_modules/aws-sdk/lib/request.js:22:10)","
    at AcceptorStateMachine.runTo (/var/runtime/node_modules/aws-sdk/lib/
state_machine.js:14:12)","    at /var/runtime/node_modules/aws-sdk/lib/
state_machine.js:26:10","    at Request.<anonymous> (/var/runtime/node_modules/aws-
sdk/lib/request.js:38:9)","    at Request.<anonymous> (/var/runtime/node_modules/
aws-sdk/lib/request.js:688:12)","    at Request.callListeners (/var/runtime/
node_modules/aws-sdk/lib/sequential_executor.js:116:18)"]}
END RequestId: 390c1f8d-0d9b-4b44-b087-8de64479ab44

```

- 올바른 태그를 사용했음에도 코드에서 인스턴스를 중지하려고 시도하지 않았다는 것을 증명하기 위해 다른 테스트 이벤트를 만들어 제출해보겠습니다.

코드 소스 위에 있는 테스트 탭을 선택합니다. 콘솔에는 기존 `SampleBadTagChangeEvent` 테스트 이벤트가 표시됩니다.

- 새 역할 생성을 선택합니다.
- 이벤트 이름에 `SampleGoodTagChangeEvent`를 입력합니다.
- 17행에서 `NOT-`를 삭제하여 값을 `valid-value`로 변경합니다.
- 테스트 이벤트 창 상단에서 저장을 선택한 다음 테스트를 선택합니다.

결과는 다음과 같이 표시되며, 이를 통해 함수가 유효한 태그를 인식하고 인스턴스 종료를 시도하지 않았음을 알 수 있습니다.

```
START RequestId: 53631a49-2b54-42fe-bf61-85b9e91e86c4 Version: $LATEST
```

```

2022-12-01T23:24:12.244Z    53631a49-2b54-42fe-bf61-85b9e91e86c4    INFO    Tags
changed on monitored EC2 instance ( i-00000000aaaaaaaa )
2022-12-01T23:24:12.244Z    53631a49-2b54-42fe-bf61-85b9e91e86c4    INFO    The
instance has the required tag key and value -- no action
END RequestId: 53631a49-2b54-42fe-bf61-85b9e91e86c4

```

브라우저에서 Lambda 콘솔을 열어둡니다.

4단계. 함수를 실행하는 EventBridge 규칙을 만듭니다.

이제 이벤트와 일치하고 Lambda 함수를 가리키는 EventBridge 규칙을 만들 수 있습니다.

EventBridge 규칙을 만드는 방법

1. 다른 브라우저 탭 또는 창에서 [EventBridge 콘솔](#)을 열고 규칙 생성 페이지를 엽니다.
2. 이름에 **ec2-instance-rule**을 입력한 후 다음을 선택합니다.
3. 아래로 스크롤하여 생성 방법으로 이동한 다음 사용자 정의 패턴(JSON 편집기)을 선택합니다.
4. 편집 상자에 다음 패턴 텍스트를 붙여넣고 다음을 선택합니다.

```

{
  "source": [
    "aws.tag"
  ],
  "detail-type": [
    "Tag Change on Resource"
  ],
  "detail": {
    "service": [
      "ec2"
    ],
    "resource-type": [
      "instance"
    ]
  }
}

```

이 규칙은 Amazon EC2 인스턴스의 Tag Change on Resource 이벤트를 매칭하고 다음 단계에서 대상으로 지정한 모든 이벤트를 호출합니다.

5. 다음으로, Lambda 함수를 대상으로 추가합니다. 대상 1 상자의 대상 선택에서 Lambda 함수를 선택합니다.
6. 함수에서 이전에 생성한 AutoEC2Termination 함수를 선택한 후 다음을 선택합니다.
7. 태그 구성 페이지에서 다음을 선택합니다. 검토 및 생성 페이지에서 규칙 생성을 선택합니다. 이렇게 하면 EventBridge가 지정된 Lambda 함수를 호출할 수 있는 권한도 자동으로 부여됩니다.

5단계. 전체 솔루션을 테스트합니다.

EC2 인스턴스를 만들고 해당 태그를 변경할 때 어떤 일이 발생하는지 관찰하는 방식으로 최종 결과를 테스트할 수 있습니다.

실제 인스턴스로 모니터링 솔루션을 테스트하는 방법

1. [Amazon EC2 콘솔](#)의 인스턴스 페이지를 엽니다.
2. Amazon EC2 인스턴스 생성 실행하기 전에 키 `valid-key`와 값 `valid-value`이 포함된 태그를 지정합니다. 인스턴스를 생성하고 시작하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [1단계: 인스턴스 시작](#)을 참조하세요. 인스턴스 시작 절차의 3단계에서 이름 태그를 입력할 때도 추가 태그 추가를 선택하고 태그 추가를 선택한 다음 **valid-key**의 키 및 **valid-value**의 값을 입력합니다. 이 인스턴스가 본 자습서의 목적으로만 사용되고 완료 후 인스턴스를 삭제할 계획이라면 키 쌍 없이 계속 진행할 수 있습니다. 1단계가 끝나면 이 자습서로 돌아갑니다. 2단계: 인스턴스에 연결을 수행할 필요가 없습니다.
3. 콘솔에서 InstanceID를 복사합니다.
4. Amazon EC2 콘솔에서 Lambda 콘솔로 변경합니다. AutoEC2Termination 함수를 선택하고 코드 탭을 선택한 다음 `index.js` 탭을 선택하여 코드를 편집합니다.
5. Amazon EC2 콘솔에서 복사한 값을 붙여넣어 InstanceList의 두 번째 항목을 변경합니다. RegionToMonitor 값이 붙여넣은 인스턴스가 포함된 리전과 일치하는지 확인합니다.
6. 배포를 선택하여 변경 사항을 적용합니다. 이제 지정된 리전의 해당 인스턴스에 대한 태그를 변경했으므로 함수를 활성화할 수 있습니다.
7. Lambda 콘솔에서 Amazon EC2 콘솔로 변경합니다.
8. `valid-key` 태그를 삭제하거나 해당 키의 값을 변경하여 인스턴스에 연결된 태그를 변경합니다.

Note

실행 중인 Amazon EC2 인스턴스에서 태그를 변경하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [개별 리소스에 태그 추가 및 삭제](#)를 참조하세요.

9. 몇 초간 기다린 다음 콘솔을 새로 고칩니다. 인스턴스는 인스턴스 상태를 중지 중으로 변경한 다음 중지 됨으로 변경해야 합니다.
10. 함수를 사용하여 Amazon EC2 콘솔에서 Lambda 콘솔로 변경하고 모니터 탭을 선택합니다.
11. 로그 탭을 선택하고 최근 간접 호출 표의 LogStream 열에 있는 가장 최근 항목을 선택합니다.

Amazon CloudWatch 콘솔이 열리면 Lambda 함수를 마지막으로 간접 호출한 로그 이벤트 페이지가 열립니다. 마지막 항목은 다음 예시와 유사합니다.

```
2022-11-30T12:03:57.544-08:00    START RequestId: b5befd18-2c41-43c8-
a320-3a4b2317cdac Version: $LATEST
2022-11-30T12:03:57.548-08:00    2022-11-30T20:03:57.548Z b5befd18-2c41-43c8-
a320-3a4b2317cdac INFO Tags changed on monitored EC2 instance ( arn:aws:ec2:us-
west-2:123456789012:instance/i-1234567890abcdef0 )
2022-11-30T12:03:57.548-08:00    2022-11-30T20:03:57.548Z b5befd18-2c41-43c8-
a320-3a4b2317cdac INFO This instance is missing the required tag key or value --
attempting to stop the instance
2022-11-30T12:03:58.488-08:00    2022-11-30T20:03:58.488Z b5befd18-2c41-43c8-
a320-3a4b2317cdac INFO Successfully stopped instance [ { CurrentState: { Code: 64,
Name: 'stopping' }, InstanceId: 'i-1234567890abcdef0', PreviousState: { Code: 16,
Name: 'running' } } ]
2022-11-30T12:03:58.546-08:00    END RequestId: b5befd18-2c41-43c8-
a320-3a4b2317cdac
```

자습서 요약

이 자습서에서는 Amazon EC2 인스턴스에 대한 리소스 이벤트의 태그 변경과 일치하는 EventBridge 규칙을 만드는 방법을 설명합니다. 이 규칙은 필요한 태그가 없는 경우 인스턴스가 자동으로 종료되는 Lambda 함수를 가리키고 있습니다.

AWS 리소스의 태그 변경에 대한 Amazon EventBridge 지원을 통해 여러에서 이벤트 기반 자동화를 구축할 수 있습니다 AWS 서비스. 이 기능을와 결합하면 AWS 리소스에 안전하게 액세스하고 온디맨드로 확장하며 비용 효율적인 서버리스 솔루션을 구축할 수 있는 도구가 AWS Lambda 제공됩니다.

리소스에 대한 태그 변경 EventBridge 이벤트의 기타 사용 사례는 다음과 같습니다.

- 누군가 비정상적인 IP 주소에서 리소스에 액세스하는 경우 경고 표시 - 태그를 사용하여 리소스에 액세스하는 각 방문자의 소스 IP 주소를 저장합니다. 태그를 변경하면 CloudWatch 이벤트가 생성됩니다. 이 이벤트를 사용하여 소스 IP 주소를 유효한 IP 주소 목록과 비교하고 소스 IP 주소가 유효하지 않은 경우 경고 이메일이 활성화됩니다.

- 리소스에 대한 태그 기반 액세스 제어에 변경 사항이 있는지 모니터링 - [속성 기반 액세스 제어 \(ABAC\)](#)를 사용하여 리소스에 대한 액세스를 설정한 경우 태그 변경으로 생성된 EventBridge 이벤트를 사용하여 보안팀에 감사를 요청할 수 있습니다.

태그 변경 문제 해결

[Find resources to tag\(태그를 지정할 리소스 찾기\)](#) 쿼리 결과에서 선택된 리소스에 대한 태그 적용 또는 변경 시도 시 오류가 발생한 경우 다음 체크리스트가 도움이 될 수 있습니다.

- 리소스에 이미 최대 태그 수가 있을 수 있습니다. 일반적으로 리소스는 최대 50개의 사용자 정의 태그를 가질 수 있습니다. AWS 생성된 태그는 최대 50개의 태그에 포함되지 않습니다. 또한 다른 사용자들이 동시에 동일한 리소스에 태그를 추가함에 따라 리소스의 태그가 최고 한도로 올라갈 수 있습니다.
- 일부 서비스의 경우 태그 생성에 대해 다른 문자 집합을 허용합니다(또는 허용되는 문자 집합을 제한합니다). 특수 문자를 사용하여 태그를 추가하거나 변경한 경우, 리소스의 서비스 설명서에서 태그 요구 사항을 검토하여 이러한 문자가 해당 서비스에서 허용되는지 확인합니다.
- 리소스의 태그를 수정할 수 있는 권한이 없을 수 있습니다. 리소스에 대한 기존 태그를 볼 권한이 없는 경우, 리소스의 태그를 변경할 수 없습니다.
- 리소스를 변경할 수 있는 권한이 없을 수 있습니다. 리소스의 메타데이터 변경은 다른 관리자에 의해 제한될 수 있습니다.
- 리소스는 다른 사용자 또는 프로세스에 의해 편집 또는 삭제되었을 수 있습니다. 예를 들어, CloudFormation 스택 생성 과정에서 리소스가 시작되었다고 가정해 보겠습니다. 스택이 삭제되었거나 더 이상 활성 상태가 아닌 경우, 리소스를 더 이상 사용하지 못할 수 있습니다.
- 리소스가 오프라인이거나 종료되었거나 또는 리소스에 대해 다른 업데이트(예: 소프트웨어 업그레이드)가 진행 중인 경우 태그 변경이 가능하지 않을 수 있습니다.
- 태그 변경이 완료되기 전에 브라우저 탭을 닫거나 페이지를 변경하면 태그 변경이 실패할 수 있습니다. 태그 변경을 마치고 페이지에서 나가기 전에 성공 또는 실패 배너가 페이지에 표시될 때까지 기다립니다.
- 에 대한 속도 제한이 있지만 AWS Resource Groups Tagging API 태그 지정하려는 서비스는 Resource Groups Tagging API 제한 전에 발생할 수 있는 별도의 제한을 부과할 수 있습니다.

실패한 태그 변경 재시도

선택한 리소스 중 하나 이상에 대해 태그 변경이 실패한 경우 Tag Editor는 페이지의 하단에 빨간색 배너를 표시합니다. 배너는 발생하는 실패의 각 유형에 대한 오류 메시지를 표시합니다. 각 오류에 대해 배너는 Tag Editor가 태그 변경을 할 수 없는 특정 리소스를 식별합니다. 오류에 대한 검토 및 [문제 해결 후](#), 리소스에 대해 실패한 태그 변경 재시도를 선택하여 태그 변경이 실패한 리소스에 대해서만 변경을 다시 시도합니다.

Tag Editor 보안

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS와 사용자 간의 공동 책임입니다. [공동 책임 모델](#)은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

- 클라우드 보안 - AWS 서비스에서 실행되는 인프라를 보호할 AWS 책임이 있습니다 AWS 클라우드. AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사원은 정기적으로 [AWS 규정 준수 프로그램](#)의 일환으로 보안 효과를 테스트하고 검증합니다. Tag Editor에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 [규정 준수 프로그램 제공 범위 내의 AWS 서비스](#)를 참조하십시오.
- 클라우드의 보안 - 사용자의 책임은 AWS 서비스 사용하에 따라 결정됩니다. 또한 귀하는 귀사의 데이터 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Tag Editor 사용 시 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목표를 충족하도록 Tag Editor를 구성하는 방법을 보여줍니다.

주제

- [Tag Editor의 데이터 보호](#)
- [Tag Editor의 ID 및 액세스 관리](#)
- [Tag Editor에서의 로깅 및 모니터링](#)
- [Tag Editor의 규정 준수 검증](#)
- [Tag Editor 복원성](#)
- [Tag Editor의 인프라 보안](#)

Tag Editor의 데이터 보호

AWS [공동 책임 모델](#) Tag Editor의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS는 모든 것을 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 [데이터 프라이버시 FAQ](#)를 참조하십시오. 유럽

의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 [AWS 공동 책임 모델 및 GDPR](#) 블로그 게시물을 참조하세요.

데이터 보호를 위해 자격 증명을 보호하고 AWS 계정 AWS IAM Identity Center 또는 AWS Identity and Access Management (IAM)를 사용하여 개별 사용자를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용하세요.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다 AWS CloudTrail. CloudTrail 추적을 사용하여 AWS 활동을 캡처하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 [CloudTrail 추적 작업을 참조하세요](#).
- AWS 암호화 솔루션과 내부의 모든 기본 보안 제어를 사용합니다 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 API를 AWS 통해 액세스할 때 FIPS 140-3 검증 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 [Federal Information Processing Standard\(FIPS\) 140-3](#)을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 Tag Editor 또는 기타 AWS 서비스에서 콘솔, API AWS CLI 또는 AWS SDKs를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명을 URL에 포함해서는 안 됩니다.

데이터 암호화

태그 지정 정보는 암호화되지 않습니다. 암호화되지는 않지만 태그는 보안 전략의 일부로 사용되는 정보가 포함될 수 있으므로 리소스의 태그에 액세스할 수 있는 사용자를 반드시 관리해야 합니다. 태그를 수정할 수 있는 접근 권한은 특정 사용자의 권한을 높이는 데 사용될 수 있기 때문에 특히 더 관리해야 합니다.

저장 시 암호화

Tag Editor에만 국한된 서비스 또는 네트워크 트래픽을 격리할 수 있는 추가적인 방법은 없습니다. 해당하는 경우 AWS 특정 격리를 사용합니다. Virtual Private Cloud(VPC)에서 Tag Editor API 및 콘솔을 사용하여 개인정보 보호 및 인프라 보안을 극대화할 수 있습니다.

전송 중 암호화

Tag Editor 데이터는 백업을 위해 서비스의 내부 데이터베이스로 전송되는 동안 암호화됩니다. 이는 사용자가 구성할 수 없습니다.

키 관리

Tag Editor는 현재 AWS Key Management Service 와 통합되지 않으므로 지원하지 않습니다 AWS KMS keys.

인터넷워크 트래픽 개인 정보

Tag Editor는 Tag Editor 사용자와 간의 모든 전송에 HTTPS를 사용합니다 AWS. Tag Editor는 전송 계층 보안 (TLS) 1.3을 사용하면서 TLS 1.2도 지원합니다.

Tag Editor의 ID 및 액세스 관리

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는입니다. IAM 관리자는 누가 Tag Editor 리소스를 사용하도록 인증되고(로그인됨) 권한이 부여되는지(권한 있음)를 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는입니다.

주제

- [대상](#)
- [ID를 통한 인증](#)
- [정책을 사용하여 액세스 관리](#)
- [Tag Editor가 IAM과 작동하는 방식](#)
- [Tag Editor 자격 증명 기반 정책 예제](#)
- [Tag Editor 자격 증명 및 액세스 문제 해결](#)

대상

AWS Identity and Access Management (IAM)를 사용하는 방법은 역할에 따라 다릅니다.

- 서비스 사용자 - 기능에 액세스할 수 없는 경우 관리자에게 권한 요청([참조 Tag Editor 자격 증명 및 액세스 문제 해결](#))
- 서비스 관리자 - 사용자 액세스 결정 및 권한 요청 제출([Tag Editor가 IAM과 작동하는 방식](#) 참조)
- IAM 관리자 - 액세스를 관리하기 위한 정책 작성([Tag Editor 자격 증명 기반 정책 예제](#) 참조)

ID를 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여 로그인하는 방법입니다. AWS 계정 루트 사용자, IAM 사용자 또는 IAM 역할을 수임하여 인증해야 합니다.

AWS IAM Identity Center (IAM Identity Center), Single Sign-On 인증 또는 Google/Facebook 자격 증명과 같은 자격 증명 소스의 자격 증명을 사용하여 페더레이션 자격 증명으로 로그인할 수 있습니다. 로그인하는 방법에 대한 자세한 내용은 AWS Sign-In 사용 설명서의 [AWS 계정에 로그인하는 방법](#) 섹션을 참조하세요.

프로그래밍 방식 액세스를 위해서는 요청에 암호화 방식으로 서명할 수 있는 SDK 및 CLI를 AWS 제공합니다. 자세한 내용은 IAM 사용 설명서의 [API 요청용 AWS Signature Version 4](#) 섹션을 참조하세요.

AWS 계정 루트 사용자

를 생성할 때 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 AWS 계정 theroot 사용자라는 하나의 로그인 자격 증명으로 AWS 계정 시작합니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 자격 증명이 필요한 작업은 IAM 사용 설명서의 [루트 사용자 자격 증명](#)이 필요한 작업 섹션을 참조하세요.

사용자 및 그룹

[IAM 사용자](#)는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가진 ID입니다. 장기 자격 증명이 있는 IAM 사용자 대신 임시 자격 증명을 사용하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 [자격 증명 공급자와의 페더레이션을 사용하여 IAM 사용 설명서의 임시 자격 증명을 AWS 사용하여 액세스 하도록 인간 사용자에게 요구](#)를 참조하세요.

[IAM 그룹](#)은 IAM 사용자 모음을 지정하고 대규모 사용자 집합에 대한 관리 권한을 더 쉽게 만듭니다. 자세한 내용은 IAM 사용 설명서의 [IAM 사용자 사용 사례](#) 섹션을 참조하세요.

역할

[IAM 역할](#)은 임시 자격 증명을 제공하는 특정 권한이 있는 자격 증명입니다. [사용자에서 IAM 역할\(콘솔\)로 전환하거나 또는 API 작업을 호출하여 역할을](#) 수임할 수 있습니다. AWS CLI AWS 자세한 내용은 IAM 사용 설명서의 [역할 수임 방법](#)을 참조하세요.

IAM 역할은 페더레이션 사용자 액세스, 임시 IAM 사용자 권한, 교차 계정 액세스, 교차 서비스 액세스 및 Amazon EC2에서 실행되는 애플리케이션에 유용합니다. 자세한 내용은 IAM 사용 설명서의 [교차 계정 리소스 액세스](#)를 참조하세요.

정책을 사용하여 액세스 관리

정책을 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결될 때 권한을 정의합니다.는 보안 주체가 요청할 때 이러한 정책을 AWS 평가합니다. 대부분의 정책은에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서에 대한 자세한 내용은 IAM 사용 설명서의 [JSON 정책 개요](#) 섹션을 참조하세요.

정책을 사용하여 관리자는 어떤 보안 주체가 어떤 리소스에 대해 어떤 조건에서 작업을 수행할 수 있는지 정의하여 누가 무엇을 액세스할 수 있는지 지정합니다.

기본적으로 사용자 및 역할에는 어떠한 권한도 없습니다. IAM 관리자는 IAM 정책을 생성하고 사용자가 수임할 수 있는 역할에 추가합니다. IAM 정책은 작업을 수행하기 위해 사용하는 방법과 관계없이 작업에 대한 권한을 정의합니다.

ID 기반 정책

ID 기반 정책은 ID(사용자, 사용자 그룹 또는 역할)에 연결하는 JSON 권한 정책 문서입니다. 이러한 정책은 자격 증명이 수행할 수 있는 작업, 대상 리소스 및 이에 관한 조건을 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 [고객 관리형 정책으로 사용자 지정 IAM 권한 정의](#)를 참조하세요.

ID 기반 정책은 인라인 정책(단일 ID에 직접 포함) 또는 관리형 정책(여러 ID에 연결된 독립 실행형 정책)일 수 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 [관리형 정책 및 인라인 정책 중에서 선택](#) 섹션을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 예를 들어 IAM 역할 신뢰 정책 및 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 리소스 기반 정책에서 [보안 주체를 지정](#)해야 합니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

액세스 제어 목록(ACL)

액세스 제어 목록(ACL)은 어떤 위탁자(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Amazon S3 AWS WAF 및 Amazon VPC는 ACLs. ACL에 관한 자세한 내용은 Amazon Simple Storage Service 개발자 가이드의 [액세스 제어 목록\(ACL\) 개요](#)를 참조하세요.

기타 정책 타입

AWS 는 보다 일반적인 정책 유형에서 부여한 최대 권한을 설정할 수 있는 추가 정책 유형을 지원합니다.

- 권한 경계 - ID 기반 정책에서 IAM 엔터티에 부여할 수 있는 최대 권한을 설정합니다. 자세한 정보는 IAM 사용 설명서의 [IAM 엔터티의 권한 범위](#)를 참조하세요.
- 서비스 제어 정책(SCP) - AWS Organizations내 조직 또는 조직 단위에 대한 최대 권한을 지정합니다. 자세한 내용은 AWS Organizations 사용 설명서의 [서비스 제어 정책](#)을 참조하세요.
- 리소스 제어 정책(RCP) - 계정의 리소스에 사용할 수 있는 최대 권한을 설정합니다. 자세한 내용은 AWS Organizations 사용 설명서의 [리소스 제어 정책\(RCP\)](#)을 참조하세요.
- 세션 정책 - 역할 또는 페더레이션 사용자에게 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 자세한 내용은 IAM 사용 설명서의 [세션 정책](#)을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 에서 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 [정책 평가 로직](#)을 참조하세요.

Tag Editor가 IAM과 작동하는 방식

IAM을 사용하여 Tag Editor에 대한 액세스를 관리하려면 먼저 어떤 IAM 기능을 Tag Editor에 사용할 수 있는지를 이해해야 합니다. Tag Editor 및 기타에서 IAM을 AWS 서비스 사용하는 방법을 전체적으로 알아보려면 IAM 사용 설명서의 [AWS 서비스 IAM으로 작업하는](#) 섹션을 참조하세요.

주제

- [Tag Editor ID 기반 정책](#)
- [리소스 기반 정책](#)
- [태그 기반 인증](#)
- [Tag Editor IAM 역할](#)

Tag Editor ID 기반 정책

IAM 자격 증명 기반 정책을 사용하면 허용되거나 거부되는 조건 이외에도 허용되거나 거부되는 작업과 리소스를 지정할 수 있습니다. Tag Editor는 특정 작업, 리소스 및 조건 키를 지원합니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용자 설명서의 [IAM JSON 정책 요소 참조](#)를 참조하세요.

작업

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 Action 요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하세요.

Tag Editor의 정책 작업은 작업 앞에 tag: 접두사를 사용합니다. Tag Editor 작업은 전적으로 콘솔에서 수행되지만 로그 항목에 접두사 tag가 붙습니다.

예를 들어, tag:TagResources API 작업을 사용하여 리소스에 태그를 지정할 수 있는 권한을 부여하려면 해당 정책에 tag:TagResources 작업을 포함합니다. 정책 문에는 Action 또는 NotAction 요소가 포함되어야 합니다. Tag Editor는 이 서비스로 수행할 수 있는 작업을 설명하는 고유한 작업 세트를 정의합니다.

명령문 하나에 여러 태그 지정 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
    "tag:action1",
    "tag:action2",
    "tag:action3"
```

와일드카드(*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, Get라는 단어로 시작하는 모든 작업을 지정하려면 다음 작업을 포함합니다.

```
"Action": "tag:Get*"
```

Tag Editor 작업의 목록을 보려면 서비스 승인 참조의 [Tag Editor에 사용되는 작업, 리소스 및 조건 키](#)를 참조하세요.

리소스

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

Resource JSON 정책 요소는 작업이 적용되는 하나 이상의 객체를 지정합니다. 모범 사례에 따라 [Amazon 리소스 이름\(ARN\)](#)을 사용하여 리소스를 지정합니다. 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*" 
```

Tag Editor에는 자체 리소스가 없습니다. 대신 다른 AWS 서비스에서 만든 리소스에 첨부된 메타데이터(태그)를 처리합니다.

조건 키

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

Condition 요소는 정의된 기준에 따라 문이 실행되는 시기를 지정합니다. 같음(equals) 또는 미만(less than)과 같은 [조건 연산자](#)를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 [AWS 전역 조건 컨텍스트 키](#)를 참조하세요.

Tag Editor는 서비스별 조건 키를 정의하지 않습니다.

예제

Tag Editor ID 기반 정책의 예를 보려면 [Tag Editor 자격 증명 기반 정책 예제](#) 단원을 참조하세요.

리소스 기반 정책

Tag Editor는 자체 리소스를 정의하지 않으므로 리소스 기반 정책을 지원하지 않습니다.

태그 기반 인증

태그를 기반으로 한 권한 부여는 ABAC(속성 기반 액세스 제어)라고 하는 보안 전략의 일부입니다.

태그를 기반으로 리소스에 대한 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 [조건 요소](#)에 태그 정보를 제공합니다. 리소스를 만들거나 업데이트할 때 리소스에 태그를 적용할 수 있습니다.

리소스의 태그를 기반으로 리소스에 대한 액세스를 제한하는 자격 증명 기반 정책의 예시는 [태그를 기준으로 그룹 보기](#)에서 확인할 수 있습니다. 속성 기반 액세스 제어(ABAC)에 대한 자세한 내용은 IAM 사용 설명서의 [ABAC란 무엇입니까 AWS?](#)를 참조하세요.

Tag Editor IAM 역할

[IAM 역할](#)은 특정 권한이 AWS 계정 있는 내의 엔터티입니다. Tag Editor에는 서비스 역할이 없거나 이를 사용하지 않습니다.

Tag Editor에서 임시 보안 인증 정보 사용

Tag Editor에서 임시 보안 인증 정보를 사용하여 페더레이션을 통해 로그인하거나, IAM 역할을 맡거나, 교차 계정 역할을 맡을 수 있습니다. [AssumeRole](#) 또는와 같은 AWS STS API 작업을 호출하여 임시 보안 자격 증명을 얻습니다 [GetFederationToken](#).

서비스 연결 역할

[서비스 연결 역할](#)을 사용하면 AWS 서비스 가 다른 서비스의 리소스에 액세스하여 사용자를 대신하여 작업을 완료할 수 있습니다.

Tag Editor에는 서비스 연결 역할이 없거나 이를 사용하지 않습니다.

서비스 역할

이 기능을 사용하면 서비스가 사용자를 대신하여 [서비스 역할](#)을 수입할 수 있습니다.

Tag Editor에는 서비스 역할이 없거나 이를 사용하지 않습니다.

Tag Editor 자격 증명 기반 정책 예제

기본적으로 역할 및 사용자와 같은 IAM 보안 주체는 태그를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface (AWS CLI) 또는 AWS APIs를 사용하여 작업을 수행할 수 없습니다. IAM 관리자는 지정된 리소스에서 특정 API 작업을 수행할 수 있는 권한을 보안 주체에 부여하는 IAM 정책을 생성해야 합니다. 그런 다음 관리자는 해당 권한이 필요한 보안 주체에 이러한 정책을 연결해야 합니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM 자격 증명 기반 정책을 생성하는 방법에 대한 지침은 IAM 사용 설명서의 [JSON 탭에서 정책 생성](#)을 참조하세요.

주제

- [정책 모범 사례](#)
- [Tag Editor 콘솔 및 리소스 그룹 태그 지정 API 사용](#)
- [사용자가 자신의 고유한 권한을 볼 수 있도록 허용](#)
- [태그를 기준으로 그룹 보기](#)

정책 모범 사례

ID 기반 정책에 따라 계정에서 사용자가 Tag Editor 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르세요.

- AWS 관리형 정책을 시작하고 최소 권한으로 전환 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반적인 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 에서 사용할 수 있습니다 AWS 계정. 사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 추가로 줄이는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 [AWS 관리형 정책](#) 또는 [AWS 직무에 대한 관리형 정책](#)을 참조하세요.
- 최소 권한 적용 - IAM 정책을 사용하여 권한을 설정하는 경우, 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 [IAM의 정책 및 권한](#)을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 - 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. AWS 서비스와 같은 특성을 통해 사용되는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다 CloudFormation. 자세한 내용은 IAM 사용 설명서의 [IAM JSON 정책 요소: 조건](#)을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 새로운 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 [IAM Access Analyzer에서 정책 검증](#)을 참조하세요.
- 다중 인증(MFA) 필요 -에서 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 AWS 계정됩니다. API 작업을 직접적으로 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 [MFA를 통한 보안 API 액세스](#)를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 [IAM의 보안 모범 사례](#)를 참조하세요.

Tag Editor 콘솔 및 리소스 그룹 태그 지정 API 사용

Tag Editor 콘솔 및 리소스 그룹 태그 지정 API에 액세스하려면 최소한의 권한 집합이 있어야 합니다. 이러한 권한을 통해 리소스에 연결된 태그에 대한 세부 정보를 나열하고 볼 수 있어야 합니다 AWS 계정. 최소 필수 권한보다 더 제한적인 ID 기반 정책을 만들면 콘솔과 API 명령이 해당 정책이 연결된 IAM 보안 주체에 대해 의도대로 작동하지 않습니다.

이러한 보안 주체가 Tag Editor를 계속 사용할 수 있도록 하려면 다음 정책(또는 다음 정책에 나와 있는 권한이 포함된 정책)을 엔터티에 연결합니다. 자세한 내용은 IAM 사용자 설명서의 [사용자에게 권한 추가](#)를 참조하세요.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "tag:GetResources",
        "tag:TagResources",
        "tag:UntagResources",
        "tag:getTagKeys",
        "tag:getTagValues",
        "resource-explorer:List*"
      ],
      "Resource": "*"
    }
  ]
}
```

Tag Editor 및 리소스 그룹 태그 지정 API에 대한 액세스 권한을 부여하는 방법에 대한 자세한 내용은 [Tag Editor 사용에 대한 권한 부여](#) 섹션을 참조하십시오.

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 권한이 포함됩니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

태그를 기준으로 그룹 보기

자격 증명 기반 정책의 조건을 사용하여 태그를 기반으로 Tag Editor 리소스에 대한 액세스를 제어할 수 있습니다. 이 예제에서는 리소스인 이 경우 리소스 그룹 보기를 허용하는 정책을 생성할 수 있는 방

법을 보여줍니다. 단, 그룹 태그 project의 값이 호출하는 보안 주체에 연결된 project 태그와 동일한 값인 경우에만 권한이 부여됩니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "resource-groups:ListGroup",
      "Resource": "arn:aws:resource-groups:us-east-1:111122223333:group/group_name"
    },
    {
      "Effect": "Allow",
      "Action": "resource-groups:ListGroup",
      "Resource": "arn:aws:resource-groups:us-east-1:111122223333:group/group_name",
      "Condition": {
        "StringEquals": {"aws:ResourceTag/project": "${aws:PrincipalTag/project}"}
      }
    }
  ]
}
```

이 정책을 계정의 사용자에게 연결할 수 있습니다. 태그 키가 project이고 태그 값이 alpha인 사용자가 리소스 그룹을 보려면 해당 그룹에도 project=alpha 태그를 지정해야 합니다. 그렇지 않으면 사용자는 액세스가 거부됩니다. 조건 키 project은(는) 조건 키 이름이 대소문자를 구분하지 않기 때문에 Project 및 project 모두와 일치합니다. 자세한 내용은 IAM 사용자 설명서의 [IAM JSON 정책 요소: 조건](#)을 참조하세요.

Tag Editor 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 Tag Editor 및 IAM에서 발생할 수 있는 공통적인 문제를 진단하고 수정할 수 있습니다.

주제

- [Tag Editor에서 작업을 수행할 권한이 없음](#)

- [iam:PassRole](#)을 수행하도록 인증되지 않음

Tag Editor에서 작업을 수행할 권한이 없음

에서 작업을 수행할 권한이 없다는 AWS Management Console 메시지가 표시되면 관리자에게 문의하여 도움을 받아야 합니다. 관리자는 로그인 보안 인증 정보를 제공한 사람입니다.

다음 예제 오류는 mateojackson 사용자가 콘솔을 사용하여 리소스에 대한 태그를 보려고 하지만 tag:GetTagKeys 권한이 없는 경우에 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
tag:GetTagKeys on resource: arn:aws:resource-groups::us-west-2:123456789012:resource-
type/my-test-resource
```

이 경우, Mateo는 my-test-resource 작업을 사용하여 tag:GetTagKeys 리소스에 액세스하도록 허용하는 정책을 업데이트하라고 관리자에게 요청합니다.

iam:PassRole을 수행하도록 인증되지 않음

iam:PassRole 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 Tag Editor에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 역할을 서비스에 전달할 권한이 있어야 합니다.

다음 예제 오류는 marymajor(이)라는 IAM 사용자가 콘솔을 사용하여 Tag Editor에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 권한이 없습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

이 경우, Mary가 iam:PassRole 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

Tag Editor에서의 로깅 및 모니터링

모든 Tag Editor 작업이 로그인됩니다 AWS CloudTrail.

CloudTrail을 사용하여 Tag Editor API 호출 로깅

Tag Editor는 Tag Editor AWS 서비스에서 사용자 AWS CloudTrail, 역할 또는가 수행한 작업의 레코드를 제공하는 서비스와 통합됩니다. CloudTrail은 Tag Editor 콘솔의 호출 및 리소스 그룹 API 태그 지정에 대한 코드 호출을 포함하여 Tag Editor에 대한 모든 API 호출을 이벤트로 캡처합니다. 추적을 생성하면 Tag Editor 이벤트를 포함한 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. 추적을 구성하지 않은 경우에도 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 Tag Editor에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 [AWS CloudTrail 사용자 안내서](#)를 참조하세요.

CloudTrail의 Tag Editor 정보

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화됩니다. Tag Editor 또는 Tag Editor 콘솔에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 이벤트와 함께 CloudTrail AWS 서비스 이벤트에 기록됩니다. 에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다 AWS 계정. 자세한 내용은 [CloudTrail 이벤트 기록을 사용하여 이벤트 보기](#)를 참조하세요.

Tag Editor에 대한 이벤트를 AWS 계정포함하여에 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 트레일을 생성하면 기본적으로 모든 AWS 리전에 트레일이 적용됩니다. 추적은 AWS 파티션의 모든 리전에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취 AWS 서비스 하도록 다른를 구성할 수 있습니다. 자세한 정보는 다음 자료를 참조하세요.

- [에 대한 추적 생성 AWS 계정](#)
- [CloudTrail 지원 서비스 및 통합](#)
- [CloudTrail에 대한 Amazon SNS 알림 구성](#)
- [여러 리전에서 CloudTrail 로그 파일 수신 및 여러 계정에서 CloudTrail 로그 파일 수신](#)

모든 Tag Editor 작업은 CloudTrail에서 로깅되고 [Tag Editor API 참조](#)에 기록됩니다. 콘솔의 Tag Editor 작업은 CloudTrail에서 기록하며 eventSource가 tagging.amazonaws.com인 이벤트로 표시됩니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에게 관한 정보가 포함됩니다. ID 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 IAM 사용자 자격 증명 정보로 했는지 여부.
- 역할 또는 페더레이션 사용자의 임시 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부

자세한 내용은 [CloudTrail userIdentity 요소](#)를 참조하세요.

Tag Editor 로그 파일 항목 이해

트레일이란 지정한 S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보를 포함하고 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출에 대한 순서 지정된 스택 추적이 아니기 때문에 특정 순서로 표시되지 않습니다.

다음은 TagResources 작업을 보여주는 CloudTrail 로그 항목이 나타낸 예시입니다.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROEXAMPLEEXAMPLE:botocore-session-1661372702",
    "arn": "arn:aws:sts::123456789012:assumed-role/cli-role/botocore-session-1661372702",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROEXAMPLEEXAMPLE",
        "arn": "arn:aws:iam::123456789012:role/cli-role",
        "accountId": "123456789012",
        "userName": "cli-role"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-08-24T20:25:03Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2022-08-24T20:27:14Z",
```

```
"eventSource": "tagging.amazonaws.com",
"eventName": "TagResources",
"awsRegion": "us-east-1",
"sourceIPAddress": "72.21.198.65",
"userAgent": "aws-cli/2.7.14 Python/3.9.11 Windows/10 exe/AMD64 prompt/off command/
resourcegroupstaggingapi.tag-resources",
"requestParameters": {
  "resourceARNList": [
    "arn:aws:events:us-east-1:123456789012:rule/SecretsManagerMonitorRule"
  ],
  "tags": {
    "owner": "alice"
  }
},
"responseElements": {
  "failedResourcesMap": {}
},
"requestID": "8f9ea891-4125-460c-802f-26c11EXAMPLE",
"eventID": "b2c9322a-aad7-424b-8f0b-423daEXAMPLE",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "tagging.us-east-1.amazonaws.com"
}
}
```

Tag Editor의 규정 준수 검증

AWS 서비스 가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 [AWS 서비스 규정 준수 프로그램 범위 내](#)를 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반 정보는 [AWS 규정 준수 프로그램](#).

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 [Downloading Reports in Downloading AWS Artifact](#)을 참조하세요.

사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 AWS 서비스 결정됩니다. 사용 시 규정 준수 책임에 대한 자세한 내용은 [AWS 보안 설명서를](#) AWS 서비스참조하세요.

Tag Editor 복원성

Tag Editor는 내부 서비스 리소스에 대한 자동 백업을 수행합니다. 이러한 백업은 사용자가 구성할 수 없습니다. 백업은 저장 중이거나 전송 중일 때 모두 암호화됩니다. Tag Editor는 Amazon DynamoDB에 고객 데이터를 저장합니다.

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다.는 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹과 연결된 물리적으로 분리되고 격리된 여러 가용 영역을 AWS 리전 제공합니다. 가용 영역을 사용하면 중단 없이 가용 영역 간에 자동으로 장애 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 복수 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

실수로 태그를 삭제한 경우 [AWS Support 센터](#)로 문의하십시오.

AWS 리전 및 가용 영역에 대한 자세한 내용은 [AWS 글로벌 인프라](#)를 참조하세요.

Tag Editor의 인프라 보안

Tag Editor는 서비스 또는 네트워크 트래픽을 격리하는 추가적인 방법을 제공하지 않습니다. 해당하는 경우 AWS 특정 격리를 사용합니다. Virtual Private Cloud(VPC)에서 Tag Editor API 및 콘솔을 사용하여 개인정보 보호 및 인프라 보안을 극대화할 수 있습니다.

AWS 에서 게시한 API 호출을 사용하여 네트워크를 통해 Tag Editor에 액세스합니다. 고객은 다음을 지원해야 합니다.

- Transport Layer Security(TLS) TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군 Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 액세스 키 ID와 AWS Identity and Access Management (IAM) 보안 주체와 연결된 보안 액세스 키를 사용하여 요청에 서명해야 합니다. 또는 [AWS Security Token Service](#)(AWS STS)를 사용하여 임시 보안 자격 증명을 생성하여 요청에 서명할 수 있습니다.

Tag Editor는 리소스 기반 정책을 지원하지 않습니다.


이러한 Tag Editor API 작업은 어떤 네트워크 위치에서든 호출할 수 있지만, Tag Editor는 소스 IP 주소에 따른 제한 사항을 포함할 수 있는 리소스 기반 액세스 정책을 지원합니다. Tag Editor 정책을 사용하여 특정 Amazon Virtual Private Cloud (Amazon VPC) 엔드포인트 또는 특정 VPC에서 액세스를 제어할 수도 있습니다. 이 접근 방식은 네트워크 내의 특정 VPC에서만 지정된 리소스에 대한 AWS 네트워크 액세스를 효과적으로 격리합니다.

Service quotas

다음 표는 Tag Editor의 Service Quotas에 대한 정보입니다.

이러한 Quotas는 현재 [Service Quotas 콘솔](#)을 사용하여 조정할 수 없습니다. [지원](#)에 문의하십시오.

명칭	Default
리소스별로 첨부된 태그	사용자 정의 태그 50개(AWS 생성된 태그는이 제한에 포함되지 않음)
태그 키 이름	<p>UTF-8 형식의 유니코드 문자 최소 1자, 최대 128자.</p> <p>허용되는 문자는 문자, 숫자, 공백 및 다음의 문자입니다.</p> <p>_ . : / = + - @</p> <p>접두사는 AWS 사용을 위해 예약되어 aws: 있으므로 키 이름은 로 시작할 수 없습니다.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>일부 AWS 서비스에는 몇 가지 추가 문자 또는 길이 제한이 있습니다. 세부 정보는 해당 특정 서비스에 대한 설명서를 참조하십시오.</p> </div>
태그 값	<p>UTF-8 형식의 유니코드 문자 최소 0자, 최대 256자.</p> <p>허용되는 문자는 문자, 숫자, 공백 및 다음의 문자입니다.</p>

명칭	Default	
	<p data-bbox="589 212 881 247">_ . : / = + - @</p> <div data-bbox="589 289 1029 699" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p data-bbox="621 327 740 363"> Note</p> <p data-bbox="670 384 995 657">일부 AWS 서비스에는 몇 가지 추가 문자 또는 길이 제한이 있습니다. 세부 정보는 해당 특정 서비스에 대한 설명서를 참조하십시오.</p> </div>	
<p data-bbox="115 737 524 821">GetResources API 작업 호출 비율</p>	<p data-bbox="589 737 878 772">초당 최대 15회 호출</p>	
<p data-bbox="115 863 500 898">다음 API 작업의 호출 비율:</p> <ul data-bbox="115 947 386 1157" style="list-style-type: none"> <li data-bbox="115 947 354 982">• TagResources <li data-bbox="115 1003 386 1039">• UntagResources <li data-bbox="115 1060 326 1096">• GetTagKeys <li data-bbox="115 1117 354 1152">• GetTagValues 	<p data-bbox="589 863 857 898">초당 최대 5회 호출</p>	

Tag Editor 문서 기록

변경 사항	설명	날짜
조직 전체의 규정 준수 평가 권한 업데이트	규정 준수 보고서에 액세스하는 데 도움이 되는 권한을 포함하도록 조직 전체에 대한 규정 준수 평가 권한 을 업데이트했습니다.	2024년 8월 28일
콘텐츠 업데이트	주제 제목을 업데이트하고 콘텐츠를 재구성하여 가독성과 검색 가능성을 개선했습니다.	2024년 7월 25일
에서이 가이드로 AWS 일반 참조 이동한 콘텐츠에 태그 지정	AWS 리소스 태그 지정에 대한 주제가 AWS 일반 참조 에서이 가이드로 이동되었습니다.	2023년 3월 24일
IAM 모범 사례 업데이트	IAM 모범 사례에 따라 설명서가 업데이트되었습니다. 자세한 설명은 IAM의 보안 모범 사례 를 참조하세요.	2023년 1월 3일
Tag Editor 설명서를 별도의 설명서로 옮기기	이제 Tag Editor 설명서가 사용 설명서의 일부가 아닌 자체 AWS Resource Groups 사용 설명서에 제공됩니다.	2022년 12월 13일
태그 정책 준수 여부 확인	를 사용하여 태그 정책을 생성하고 계정에 연결하면 조직 계정의 리소스에서 규정 미준수 태그를 찾을 AWS Organizations 수 있습니다.	2019년 11월 26일
이제 Tag Editor에서 태그가 지정되지 않은 리소스를 찾는 기능을 지원합니다.	이제 Tag Editor에서 특정 태그 키에 태그 값이 적용되지 않은 리소스를 검색할 수 있습니다.	2019년 6월 18일

[Tag Editor 콘솔이 AWS Systems Manager 콘솔 밖으로 이동](#)

이제 Tag Editor 콘솔은 Systems Manager 콘솔에서 독립되었습니다. Systems Manager 왼쪽 탐색 표시줄에서 여전히 Tag Editor 콘솔 포인터가 표시되지만 AWS Management Console 왼쪽 상단의 드롭다운 메뉴에서 직접 Tag Editor 콘솔을 열 수 있습니다.

2019년 6월 5일

[이전의 레거시 Tag Editor 도구는 더 이상 사용할 수 없습니다.](#)

이전의 일반적 또는 레거시 Tag Editor의 멘션은 삭제되었습니다. 이러한 도구는 AWS에서 더 이상 사용할 수 없습니다. 대신 Tag Editor를 사용하십시오.

2019년 5월 14일

[이제 Tag Editor는 여러 리전에 걸쳐 리소스 태그 지정을 지원합니다.](#)

이제 Tag Editor를 사용하면 현재 리전이 기본으로 리소스 쿼리에 추가되고 여러 리전에 걸쳐 리소스의 태그를 검색 및 관리할 수 있습니다.

2019년 5월 2일

[이제 Tag Editor는 쿼리 결과를 CSV로 내보내기를 지원합니다.](#)

태그를 지정할 리소스 찾기 페이지에서 쿼리 결과를 CSV 형식 파일로 내보낼 수 있습니다. 새 리전 열이 Tag Editor 쿼리 결과에 표시됩니다. 이제 Tag Editor를 사용하면 특정 태그 키의 값이 비어 있는 리소스를 검색할 수 있습니다. 기존 키 사이의 고유한 값을 입력하면 태그 키 값이 자동 완성됩니다.

2019년 4월 2일

이제 Tag Editor는 모든 리소스 유형을 쿼리에 추가를 지원합니다.

단일 작업에서 최대 20개 개별 리소스 유형에 태그를 적용하거나 모든 리소스 유형을 선택하여 리전의 모든 리소스 유형을 쿼리할 수 있습니다. 리소스 사이의 일관적인 태그 키를 사용할 수 있도록 자동 완성이 쿼리의 태그 키 필드에 추가되었습니다. 태그 변경이 일부 리소스에서 실패할 경우 태그 변경이 실패한 리소스에 대해서만 태그 변경을 다시 시도할 수 있습니다.

2019년 3월 19일

이제 Tag Editor는 검색에서 여러 리소스 유형을 지원합니다.

단일 작업에서 최대 20개 리소스 유형에 태그를 적용할 수 있습니다. 또한 검색 결과에서 찾은 고유한 각 태그 키의 열 또는 결과에서 선택한 리소스의 열을 포함하여 검색 결과에 표시된 열을 선택할 수 있습니다.

2019년 2월 26일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.