사용자 가이드

태깅 리소스 AWS

버전 1.0

태깅 리소스 AWS: 사용자 가이드

Table of Contents

AWS 리소스에 태그 지정

태그는 AWS 리소스 구성을 위한 메타데이터 역할을 하는 한 쌍의 키와 값입니다. 대부분의 AWS 리소스에는 리소스를 생성할 때 태그를 추가할 수 있는 옵션이 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 기준으로 리소스를 분류할 수 있습니다.

각 태그에는 다음 두 가지 부분이 있습니다.

- 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.

태그를 사용하여 용도, 소유자, 환경 또는 기타 기준으로 리소스를 분류할 수 있습니다.

AWS 리소스에 태그를 추가하는 방법

AWS 리소스에 태그를 추가하는 세 가지 방법이 있습니다.

- AWS 서비스API 작업 - API 태그 지정 작업이 AWS 서비스를 직접 지원합니다. 각 AWS 서비스가 어떤 태그 기능을 제공하는지 알아보려면 AWS 설명서 색인의 서비스 설명서를 참조하십시오.
- Tag Editor 콘솔 - 일부 서비스는 AWS Tag Editor 콘솔을 통한 태그 지정 기능을 지원합니다.
- 리소스 그룹 태그 지정 API - 대부분의 서비스는 AWS Resource Groups Tagging API를 사용하는 태그 지정 기능을 지원합니다.

AWS에서 비용이 발생하는 모든 서비스의 리소스에 태깅할 수 있습니다. 다음 서비스의 경우 AWS는 고객 사용 사례에 더 적합한 태깅을 지원하는 새로운 대체 AWS 서비스를 사용할 것을 권장합니다.

Amazon Cloud Directory	아마존 CloudSearch	Amazon Cognito Sync
AWS Data Pipeline	Amazon Elastic Transcoder	Amazon Machine Learning
AWS OpsWorks Stacks	Amazon S3 Glacier Direct	Amazon SimpleDB
Amazon WorkSpaces 애플리케이션 관리자	AWS DeepLens	

모범 사례

AWS 리소스에 대한 태깅 전략을 만들 때는 다음 모범 사례를 따르십시오.

- 개인 식별 정보(PII)나 기타 기밀 정보 또는 민감한 정보를 태그에 추가하지 않습니다. 청구를 비롯한 여러 AWS 서비스에서 태그에 액세스할 수 있습니다. 태그는 개인 데이터나 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 지정를 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 정책을 참조하세요.

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
• 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 • 환경 – 개발, 테스트, 프로덕션 리소스 간 구별 • 버전 – 리소스 또는 애플리케이션의 버전 구별	• 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	• 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	• 기밀성 – 리소스가 지원하는 특정 데이터 기밀성 수준에 대한 식별자 • 규정 준수 – 특정 규정 준수 요구 사항을 준수해야 하는 워크로드에 대한 식별자

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 AWS용으로 예약되어 있으며 이 제한이 적용되지 않습니다. `aws:` 접두사로 시작하는 태그를 편집하거나 삭제할 수 없습니다.
- 각 리소스에 대해 각 태그 키는 고유하며 하나의 값만 가질 수 있습니다.
- 태그 키는 최소 1자, 최대 128자의 UTF-8 형식 유니코드 문자로 지정해야 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 리소스 태깅에 사용할 수 있는 문자에 대한 정보는 해당 문서를 참조하십시오. 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

태그를 사용하면 AWS Management Console에서 AWS 리소스를 손쉽게 정리할 수 있습니다. 리소스와 함께 표시되도록 태그를 구성하고 태그로 리소스를 검색 및 필터링할 수 있습니다. AWS Resource Groups 서비스를 통해 하나 이상의 태그 또는 태그의 일부를 기반으로 AWS 리소스 그룹을 만들 수 있습니다. AWS CloudFormation 스택에 리소스가 있는지 여부에 따라 그룹을 만들 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 비용 탐색기 및 세부 결제 보고서를 사용하여 태그로 AWS 비용을 분류할 수 있습니다. 일반적으로 비용 센터/비즈니스 단위, 고객 또는 프로젝트와 같은 비즈니스 태그를 사용하여 AWS 비용을 기존 비용 할당 범위에 연결합니다. 하지만 비용 할당 보고서는 어떤 태그든 포함할 수 있습니다. 따라서 비용을 특정 애플리케이션, 환경 또는 규정 준수 프로그램과 같은 기술이나 보안 차원에 연결할 수 있습니다. 다음은 부분적 비용 할당 보고서의 예입니다.

Total Cost	user:Owner	user:Stack	user:Cost Center	user:Application
0.95	DbAdmin	Test	80432	Widget2
0.01	DbAdmin	Test	80432	Widget2
3.84	DbAdmin	Prod	80432	Widget2
6.00	DbAdmin	Test	78925	Widget1
234.63	SysEng	Prod	78925	Widget1
0.73	DbAdmin	Test	78925	Widget1
0.00	DbAdmin	Prod	80432	Portal
2.47	DbAdmin	Prod	78925	Portal

일부 서비스의 경우 비용 할당을 위해 AWS에서 생성되는 `createdBy` 태그를 사용하여 다른 방식으로 분류되지 않는 리소스를 나타낼 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 Billing and Cost Management 사용 설명서의 AWS에서 생성되는 비용 할당 태그를 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 Amazon EBS 스냅샷이나 롤링 중인 Amazon EBS 스냅샷을 찾아 삭제하는 스크립트의 경우 스냅샷 태그는 검색 기준을 한 차원 더 추가할 수 있습니다. out-of-date

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 제한 할 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 서비스를 참조하세요.

태깅 거버넌스

효과적인 태깅 전략은 표준화된 태그를 사용하여 프로그래밍 방식으로 일관되게 AWS 리소스 전체에 적용합니다. AWS 환경에서 태그를 관리하는 데는 사후 대응적 방식과 사전 예방적 방식을 모두 사용할 수 있습니다.

- 사후 대응적 거버넌스는 Resource Groups Tagging API, AWS Config 규칙, 사용자 지정 스크립트 등의 도구를 사용하여 제대로 태그가 지정되지 않은 리소스를 찾습니다. 리소스를 수동으로 찾으려면 Tag Editor와 세부 결제 보고서를 사용할 수 있습니다.
- 사전 예방적 거버넌스는 AWS CloudFormation, 서비스 카탈로그, AWS Organizations의 태그 정책 또는 IAM 리소스 수준 권한과 같은 도구를 사용하여 리소스 생성 시 표준화된 태그가 일관되게 적용되도록 합니다.

  예를 들어 AWS CloudFormation `Resource Tags` 속성을 사용하여 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.

자세히 알아보기

이 페이지에서는 AWS 리소스 태깅에 대한 일반적인 정보를 제공합니다. 특정 AWS 서비스의 리소스 태깅에 대한 자세한 내용은 해당 문서를 참조하십시오. 다음은 태깅에 대한 정보의 유용한 출처이기도 합니다.

- AWS Resource Groups Tagging API에 대한 자세한 내용은 리소스 그룹 태깅 API 참조 가이드를 참조하세요.
- Tag Editor에 대한 자세한 내용은 이 설명서의 Tag Editor를 참조하세요.
- 각 AWS 서비스 제공하는 태그 지정 기능에 대한 자세한 내용은 AWS 설명서 색인의 서비스 설명서를 참조하십시오.
- IAM 정책에서 태그를 사용하여 AWS 리소스를 보고 상호 작용할 수 있는 사용자를 제어하는 방법에 대한 자세한 내용은 IAM 사용 설명서에 있는 태그를 사용하여 IAM 사용자 및 역할에 대한 액세스 제어를 참조하십시오.

태그 편집기 사용

태그는 AWS 리소스 구성을 위한 메타데이터 역할을 하는 한 쌍의 키와 값입니다. 대부분의 AWS 리소스에는 리소스를 생성할 때 태그를 추가할 수 있는 옵션이 있습니다. 리소스의 예에는 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스, Amazon Simple Storage Service(S3) 버킷, AWS Secrets Manager의 시크릿 등이 있습니다. 하지만 Tag Editor를 사용하여 지원되는 여러 리소스에 태그를 한 번에 추가할 수도 있습니다. 다양한 유형의 리소스에 대해 쿼리를 작성한 후 검색 결과의 리소스에 태그를 추가, 제거하거나 바꿀 수 있습니다. 태그 기반 쿼리는 태그에 AND 연산자를 추가하여 지정한 리소스 유형 및 지정된 모든 태그와 일치하는 리소스가 쿼리를 통해 반환됩니다.

> **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 Console에 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 유형에 있는 Tag Editor 태그 지정을 참조하십시오. 태그를 지정하려는 리소스 유형이 지원되지 않는 경우, 콘솔 창의 왼쪽 하단에서 피드백을 선택하여 AWS에 알려주십시오.

리소스 태그를 지정하는 데 필요한 권한과 역할은 권한 설정 단원을 참조하십시오.

주제

태그 및 속성 기반 액세스 제어

태그는 AWS 액세스 제어 전략의 중요한 부분이 될 수 있습니다. ABAC(속성 기반 액세스 제어) 전략에서 태그를 속성으로 사용하는 방법에 대한 자세한 내용은 IAM 사용자 가이드에서 태그를 사용하여 AWS 리소스에 대한 액세스 제어하기 및 태그를 사용하여 IAM 사용자 및 역할에 대한 액세스 제어하기를 참조하십시오.

태그를 사용하여 다양한 프로젝트 및 그룹에 액세스 권한을 부여하는 방법을 보여주는 종합 자습서는 AWS Identity and Access Management사용 설명서의 IAM 자습서: 태그를 기반으로 AWS 리소스에 액세스할 수 있는 권한 정의하기에 있습니다.

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 사용을 참조하세요.

태그 이름 모범 사례

다음은 태그에 사용을 권장하는 몇 가지 모범 사례 및 명명 규칙입니다.

AWS 태그의 키 이름은 대/소문자를 구분하므로 일관되게 사용해야 합니다. 예를 들어, 태그 키 `CostCenter`와 `costcenter`는 서로 다릅니다. 태그 키 하나는 재무 분석 및 보고를 위한 비용 할당 태그로 구성될 수 있으며, 다른 태그 키는 동일한 용도로 구성되지 않을 수 있습니다.

여러 태그가 AWS에서 미리 정의되거나 다양한 AWS 서비스에서 자동으로 생성됩니다. 많은 AWS생성된 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 AWS 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 지정된 가상 회사의 AnyCompany경우 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`.

접두사를 사용하면 태그가 AWS 또는 사용할 수 있는 타사 도구가 아닌 조직에서 정의한 것으로 명확하게 식별됩니다. 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.

Tag Editor 시작하기

Tag Editor는 리소스에 태그를 지정하는 한 가지 방법입니다. 아래 섹션을 참조하여 이 기능을 사용하기 위해 충족해야 하는 사전 조건을 알아보십시오.

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

- 가입하기 AWS 계정
- 관리자 사용자 생성
- 리소스 만들기
- 권한 설정

가입하기 AWS 계정

계정이 없는 경우 다음 단계를 완료하여 계정을 만드세요. AWS 계정

가입하려면 AWS 계정

1. https://portal.aws.amazon.com/billing/signup을 엽니다.
2. 온라인 지시 사항을 따릅니다.

   가입 절차 중 전화를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

   에 AWS 계정가입하면 AWS 계정 루트 사용자a가 생성됩니다. 루트 사용자는 계정의 모든 AWS 서비스 및 리소스에 액세스하는 권한이 있습니다. 보안 모범 사례는 관리 사용자에게 관리자 액세스 권한을 할당하고, 루트 사용자만 루트 사용자 액세스 권한이 필요한 태스크를 수행하는 것입니다.

AWS 가입 절차가 완료된 후 확인 이메일을 보냅니다. 언제든지 https://aws.amazon.com/으로 이동하고 내 계정을 선택하여 현재 계정 활동을 보고 계정을 관리할 수 있습니다.

관리자 사용자 생성

등록한 AWS 계정후에는 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 보호하고 AWS IAM Identity Center활성화하고 생성하십시오 AWS 계정 루트 사용자.

보안을 유지하세요. AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 AWS Management Console소유자로 로그인합니다. 다음 페이지에서 암호를 입력합니다.

   루트 사용자를 사용하여 로그인하는 데 도움이 필요하면 AWS 로그인 사용 설명서의 루트 사용자로 로그인을 참조하세요.
2. 루트 사용자에 대해 다중 인증(MFA)을 활성화합니다.

   지침은 IAM 사용 설명서의 AWS 계정 루트 사용자 (콘솔) 에 대한 가상 MFA 디바이스 활성화를 참조하십시오.

관리 사용자 생성

1. IAM Identity Center를 활성화합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 AWS IAM Identity Center설정을 참조하세요.
2. IAM Identity Center에서 관리 사용자에게 관리 액세스 권한을 부여합니다.

   를 ID 소스로 사용하는 방법에 대한 자습서는 사용 설명서의 기본값으로 IAM Identity Center 디렉터리사용자 액세스 구성을 참조하십시오. IAM Identity Center 디렉터리 AWS IAM Identity Center

관리 사용자로 로그인

- IAM 자격 증명 센터 사용자로 로그인하려면 IAM 자격 증명 센터 사용자를 생성할 때 이메일 주소로 전송된 로그인 URL을 사용합니다.

  IAM Identity Center 사용자를 사용하여 로그인하는 데 도움이 필요하면 사용 설명서의 AWS 액세스 포털 로그인을 참조하십시오.AWS 로그인

리소스 만들기

AWS 계정 to 태그에 리소스가 있어야 합니다. 지원되는 리소스 유형에 대한 자세한 내용은 AWS Resource Groups 사용 설명서의 지원되는 리소스 유형에 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 작업을 참조하십시오.

개별 서비스에 대한 권한

> **Important**
>
> 이 섹션에서는 다른 AWS 서비스 콘솔 및 API의 리소스에 태그를 지정하려는 경우 필요한 권한에 대해 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.

태그 편집기 콘솔 사용에 필요한 권한

태그 편집기 콘솔을 사용하여 리소스를 나열하고 태그를 지정하려면 IAM의 사용자 정책 설명에 다음 권한을 추가해야 합니다. 유지 AWS 관리되고 최신 상태로 유지되는 관리형 정책을 추가하거나 자체 사용자 지정 정책을 만들고 유지할 수 있습니다. AWS

태그 편집기 권한에 AWS 관리형 정책 사용

태그 편집기는 사용자에게 미리 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 태그 편집기 AWS Resource Groups 모두에 대해 읽기 전용 작업을 호출할 수 있는 권한을 부여합니다. 리소스의 태그를 읽으려면 별도의 정책을 통해 해당 리소스에 대한 권한이 있어야 합니다. 다음 중요 노트에서 자세히 알아보십시오.

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 노트에서 자세히 알아보십시오.

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 서비스 리소스의 읽기 전용 작업에 권한을 ReadOnlyAccess부여합니다. AWS 새 AWS 서비스 정책이 나오면 자동으로 이 정책을 최신 상태로 유지합니다.
- 많은 서비스는 서비스별 읽기 전용 AWS 관리 정책을 제공하여 해당 서비스에서 제공하는 리소스에만 액세스를 제한하는 데 사용할 수 있습니다. 예를 들어, Amazon EC2는 AmazonEC2ReadOnlyAccess을 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.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [ "tag:*" ],
            "Resource": "<ARNs of resources to allow tagging>"
        }
    ]
```

```
}
```

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.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
             "Action": [ "tag:*" ],
            "Resource": "<ARNs of resources to disallow tagging>"
        }
    ]
}
```

Tag Editor 권한을 수동으로 추가하기

- `tag:*`(이 권한은 모든 Tag Editor 작업을 허용합니다. 대신 사용자가 수행할 수 있는 작업을 제한하려면 별표를 특정 작업이나 쉼표로 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**
>
> 이 `resource-groups:SearchResources` 권한을 사용하면 태그 키나 값을 사용하여 검색을 필터링할 때 Tag Editor가 리소스를 나열할 수 있습니다.

이 `resource-explorer:ListResources` 권한을 사용하면 검색 태그를 정의하지 않고 리소스를 검색할 때 태그 편집기가 리소스를 나열할 수 있습니다.

Tag Editor 사용에 대한 권한 부여

역할에 태그 AWS Resource Groups 편집기를 사용하기 위한 정책을 추가하려면 다음과 같이 하십시오.

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

```
{
  "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 제거를 참조하십시오.

태그 기반 권한 부여 및 액세스 제어

AWS 서비스 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 ABAC의 AWS용도를 참조하십시오. AWS Identity and Access Management 사용 설명서에서.
- 임시 보안 인증 정보 - 사용자는 Tag Editor 작업을 허용하는 정책이 있는 역할을 수임할 수 있습니다.

Tag Editor는 서비스 연결 역할을 사용하지 않습니다.

태그 편집기와 AWS Identity and Access Management (IAM) 통합 방법에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 다음 항목을 참조하십시오.

- AWS IAM과 함께 작동하는 서비스
- Tag Editor에 사용되는 작업, 리소스 및 조건 키
- 정책을 사용한 AWS 리소스 액세스 제어

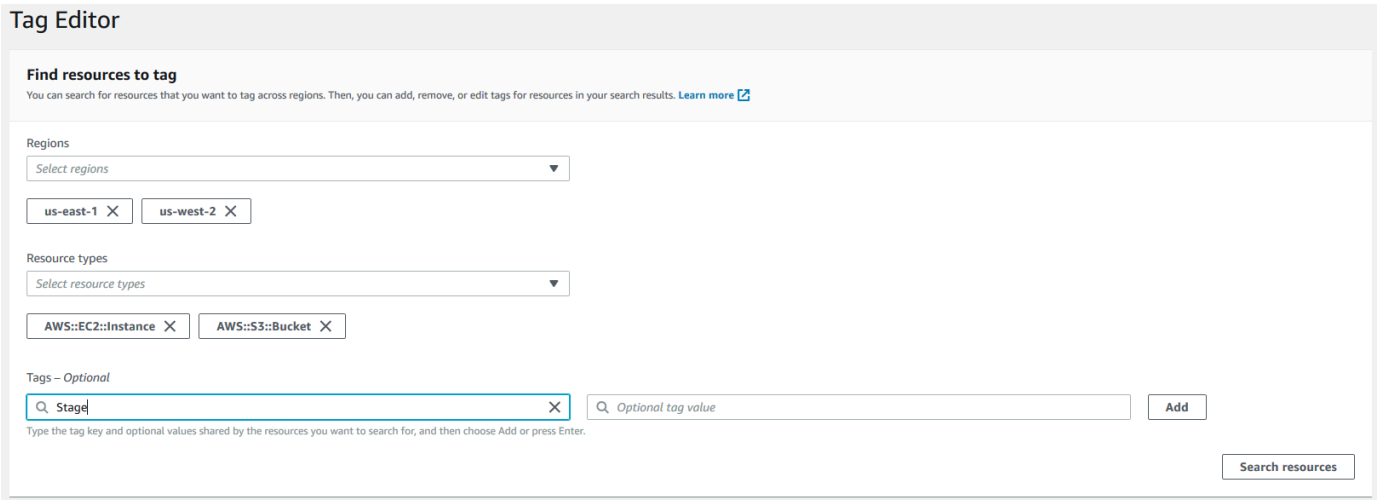
태그를 지정할 리소스 찾기

Tag Editor를 사용하여 하나 이상의 AWS 리전에서 태그 지정에 사용할 수 있는 리소스를 찾도록 쿼리를 작성합니다. 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 Groups사용 설명서의 지원되는 리소스 유형에서 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 콘솔을 엽니다.
2. (선택 사항) 태그를 지정할 리소스를 검색할 AWS 리전를 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 유형를 선택할 수 있습니다. 이 절차에서는 AWS::EC2::Instance및 을 선택합니다 AWS::S3::Bucket.



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 아래의 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

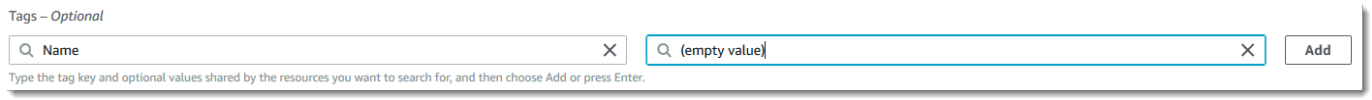
선택합니다. 쿼리는 태그에 AND 연산자를 추가하여 지정한 리소스 유형 및 지정된 모든 태그와 일치하는 리소스만 쿼리를 통해 반환됩니다.

> **Note**
>
> Tag Editor 콘솔은 현재 와일드카드를 지원하지 않습니다.

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

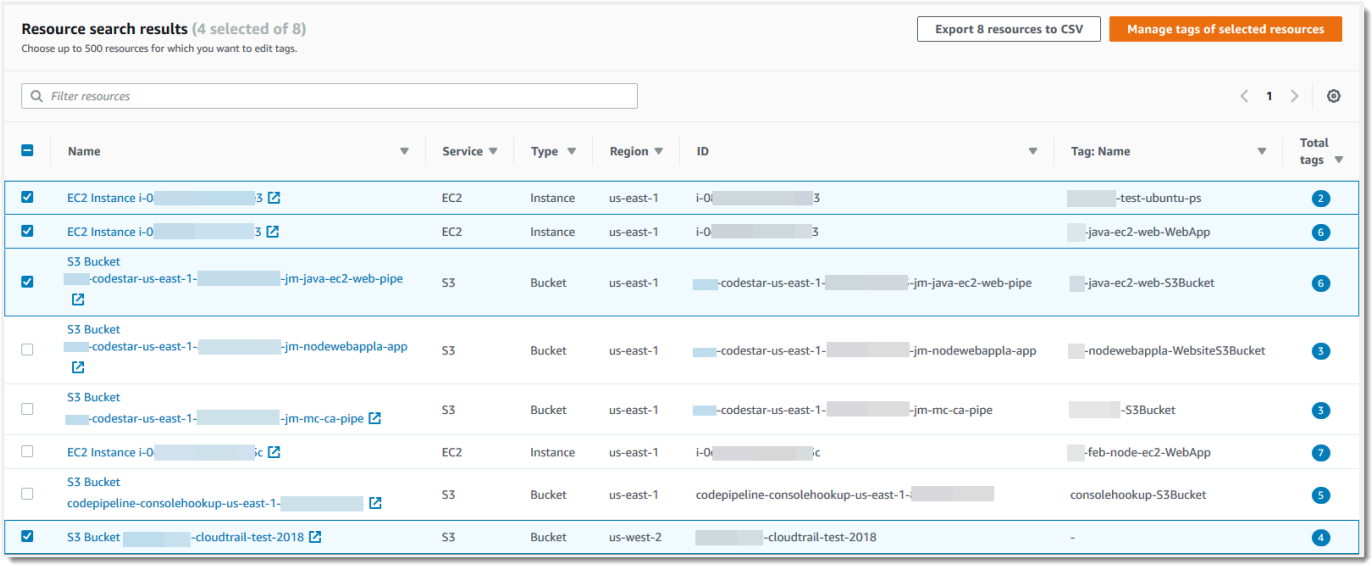
선택된 AWS 리전에 지정된 유형의 모든 리소스를 찾도록 태그 상자를 비워 둡니다. 이 쿼리는 태그가 있는 리소스를 반환하고 태그가 없는 리소스를 포함합니다. 쿼리에서 태그를 제거하려면 태그의 레이블에서 X를 선택합니다.

값이 비어 있지만 태그가 있는 리소스를 찾으려면 커서가 태그 값 상자에 있을 때 (비어 있는 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 검색를 선택합니다. 결과는 리소스 검색 결과 영역에 테이블 로 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 아이콘 을 선택합니다.

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

   결과에 표시하려는 Tag Editor 열을 켭니다. 검색 결과에 발생하는 각 태그 또는 검색 결과의 선택된 하위 집합의 열을 표시할 수 있습니다. 태그를 지정할 리소스를 찾은 후 언제라도 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 태그를 보고 편집합니다.

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.

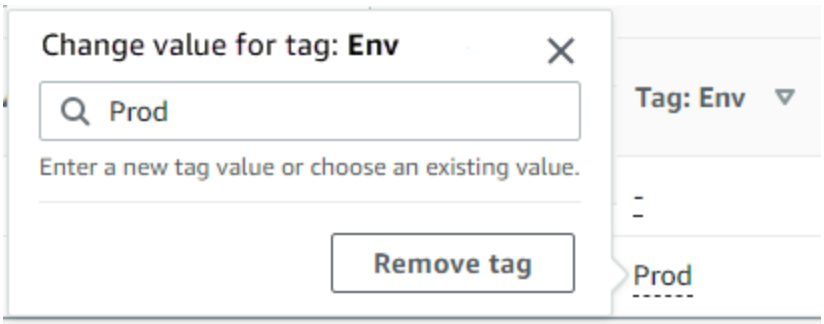
> **Note**
>
> 이 항목에서는 개별 리소스의 태그를 편집하는 방법을 설명합니다. 또한 동시에 여러 리소스를 선택하여 태그를 일괄 편집할 수 있습니다. 자세한 설명은 Tag Editor를 사용하여 태그 관리 섹션을 참조하세요.

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.

   다음 예에서는 Env 태그 열과 현재 값이 Prod인 열을 선택했습니다.



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

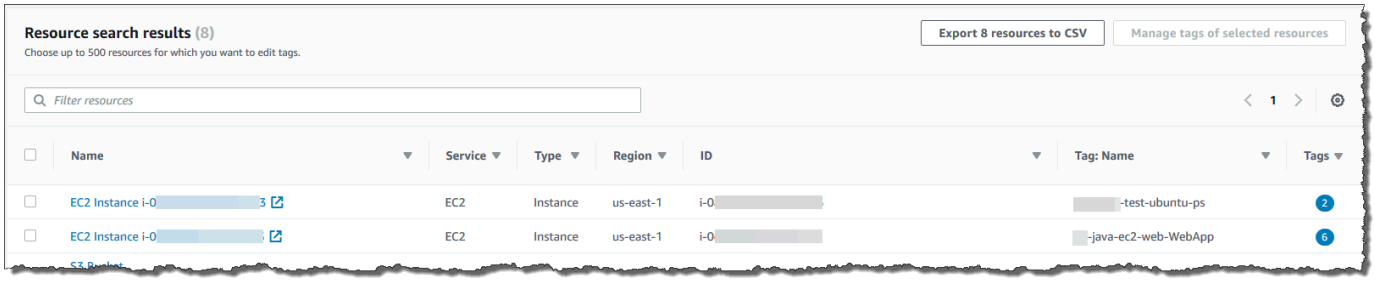
1. 태그를 지정할 리소스 찾기 쿼리의 결과에서 기존 태그를 보려는 리소스의 태그 수 열에서 숫자를 선택합니다. Tags(태그 수) 열에 대시가 포함된 리소스는 기존 태그를 가지고 있지 않습니다.
2. Resource tags(리소스 태그)에서 기존 태그를 봅니다. 태그 관리 페이지에서 태그를 변경하거나 제거할 때 선택한 리소스의 태그 관리를 선택하여 이 창을 열 수도 있습니다.

> **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.

관련 정보

- AWS Billing 사용 설명서의 비용 할당 태그 사용

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 주제에서는 AWS Management Console에서 Tag Editor를 사용하여 리소스에 태그를 지정하는 방법에 대해 설명합니다. 하지만 다음 도구를 사용하여 AWS 리소스의 태그를 관리할 수도 있습니다.

- AWS Command Line Interface(AWS CLI)의 `resourcegroupstaggingapi` 명령을 사용하여 셸 프롬프트에 명령을 입력하거나 스크립트로 작성할 수 있습니다.
- AWS Tools for PowerShell Core에서 AWS Resource GroupsAPI 태그 지정을 사용하여 PowerShell 스크립트를 만들고 실행할 수 있습니다.
- Python용 API 태그 지정 또는 Java용 API 태그 지정과 같은 리소스 그룹 API 태그 지정을 사용하여 사용 가능한 모든 AWSSDK로 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.

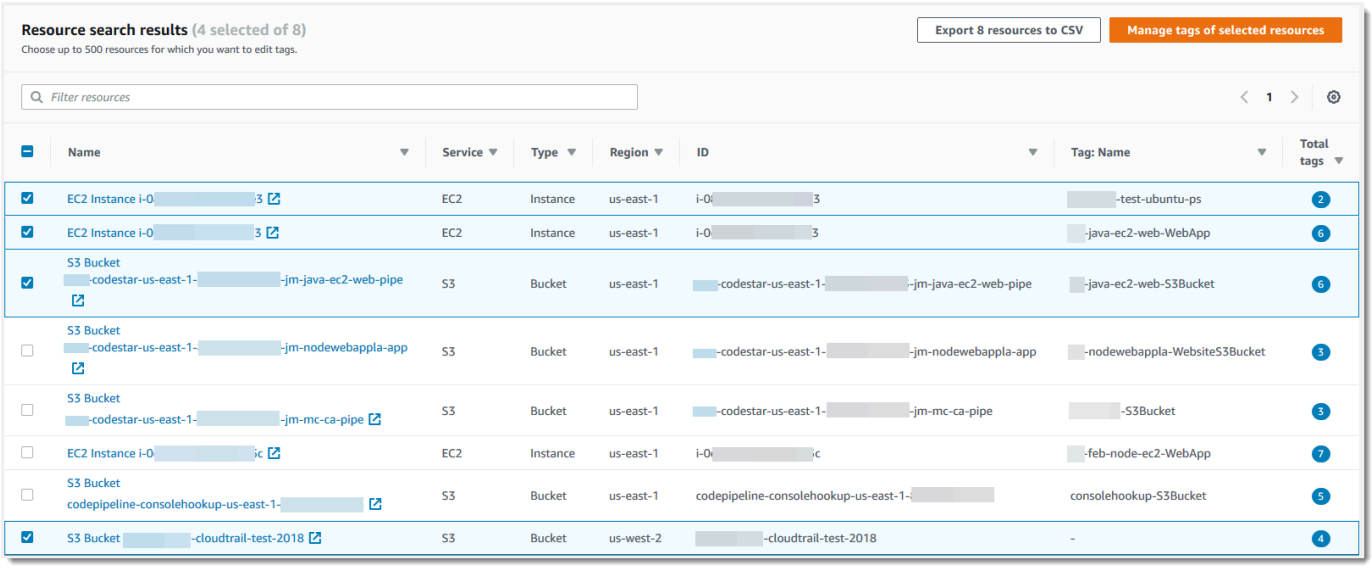
주제

선택된 리소스에 태그 추가

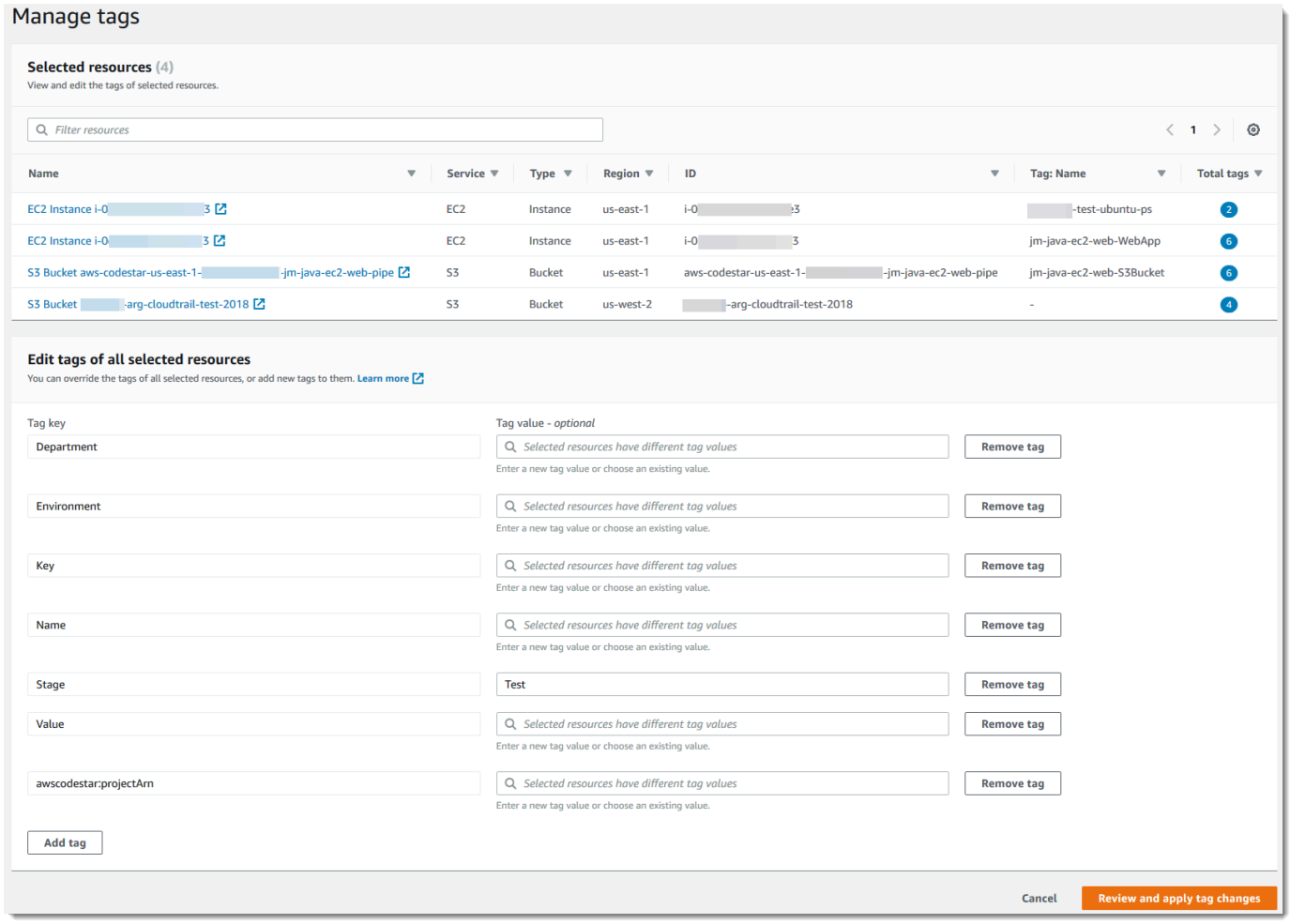
Tag Editor를 사용하여 태그를 지정할 리소스 찾기 쿼리의 결과에 있는 선택된 리소스에 태그를 추가할 수 있습니다.

> **Note**
>
> 이 주제에서는 여러 리소스의 태그를 일괄 편집하는 방법을 설명합니다. 또한 개별 리소스에 대한 태그 값을 편집할 수 있습니다. 자세한 설명은 선택한 리소스의 태그를 보고 편집합니다. 섹션을 참조하세요.

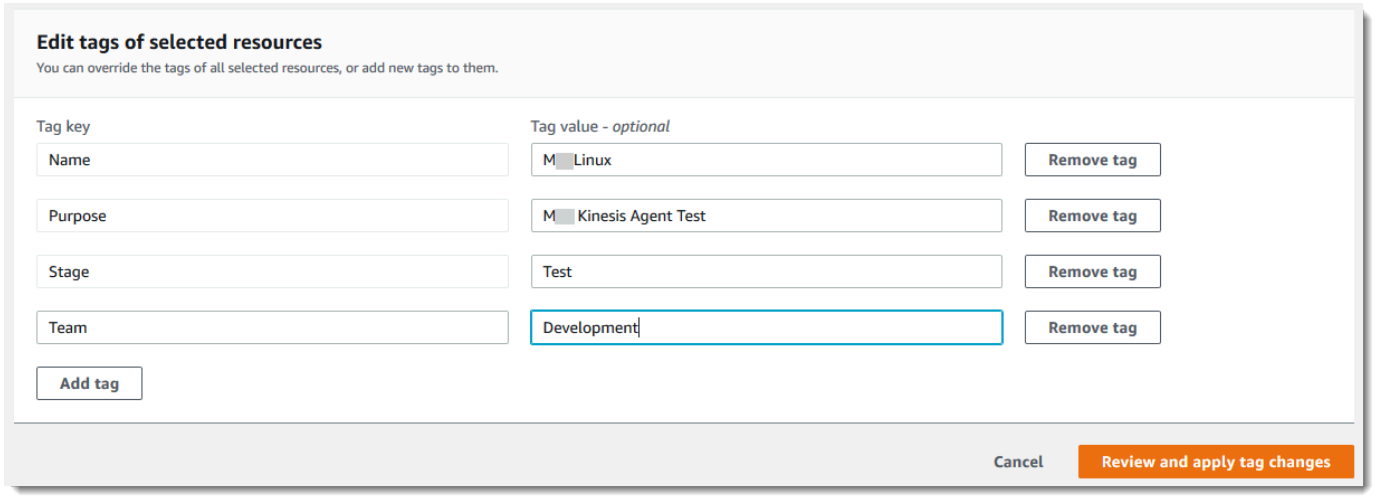
1. Tag Editor 콘솔을 열고 태그를 지정할 여러 리소스를 반환하는 쿼리를 제출합니다.
2. 태그를 지정할 리소스 찾기 쿼리의 결과 테이블에서 태그를 추가하려는 리소스 옆의 확인란을 선택합니다. 테이블 상단의 리소스 필터링에 텍스트 문자열을 입력하여 리소스의 이름, ID, 태그 키 또는 태그 값의 일부를 필터링합니다. 태그 열에서 결과의 리소스에 태그는 이미 적용되어 있습니다. 다음 예제에서 목록에 있는 첫 EC2 인스턴스에는 이미 태그가 두 개 있습니다.



3. 하나 이상의 리소스에 대한 확인란을 선택한 다음 선택한 리소스 태그 관리를 선택합니다.
4. 아래에 표시된 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. 50개 사용자 적용 태그에 근접하는 경우 새 태그를 리소스에 추가하지 못할 수 있습니다. AWS에서 생성된 태그는 50개 태그 한도에 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 적용를 선택합니다.
7. 변경 사항을 수락할 경우 선택된 모든 것에 변경 사항 적용를 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 시도할 수 있습니다. 자세한 설명은 the section called “실패한 태그 변경 재시도” 섹션을 참조하세요.

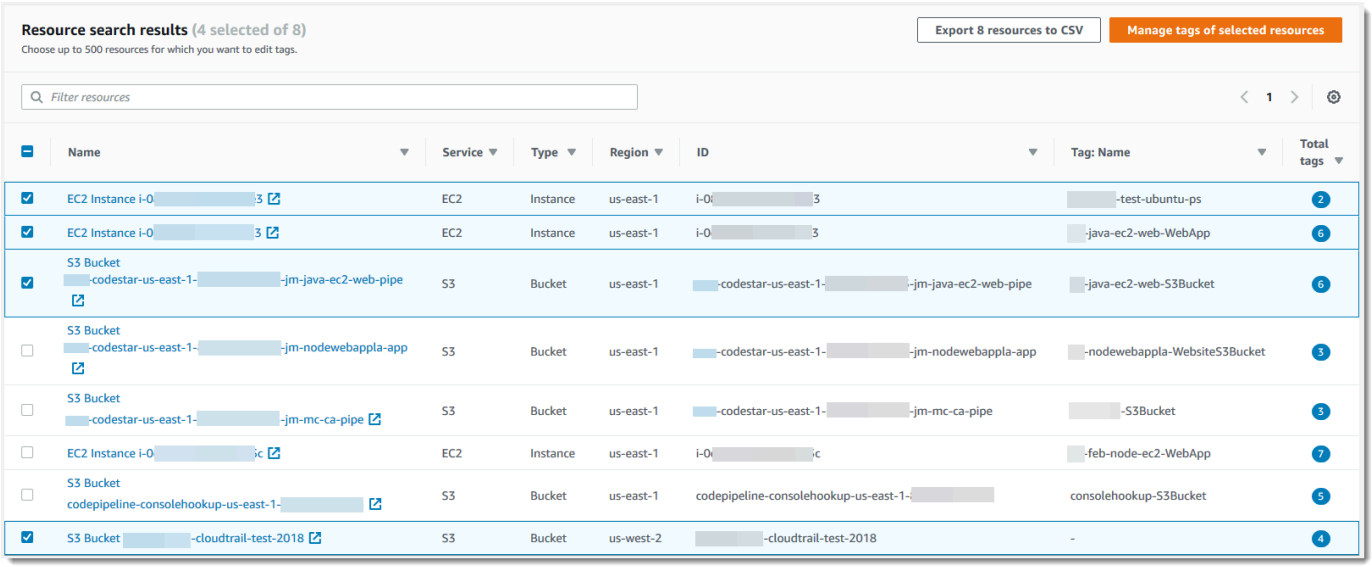
선택된 리소스의 태그 편집

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.

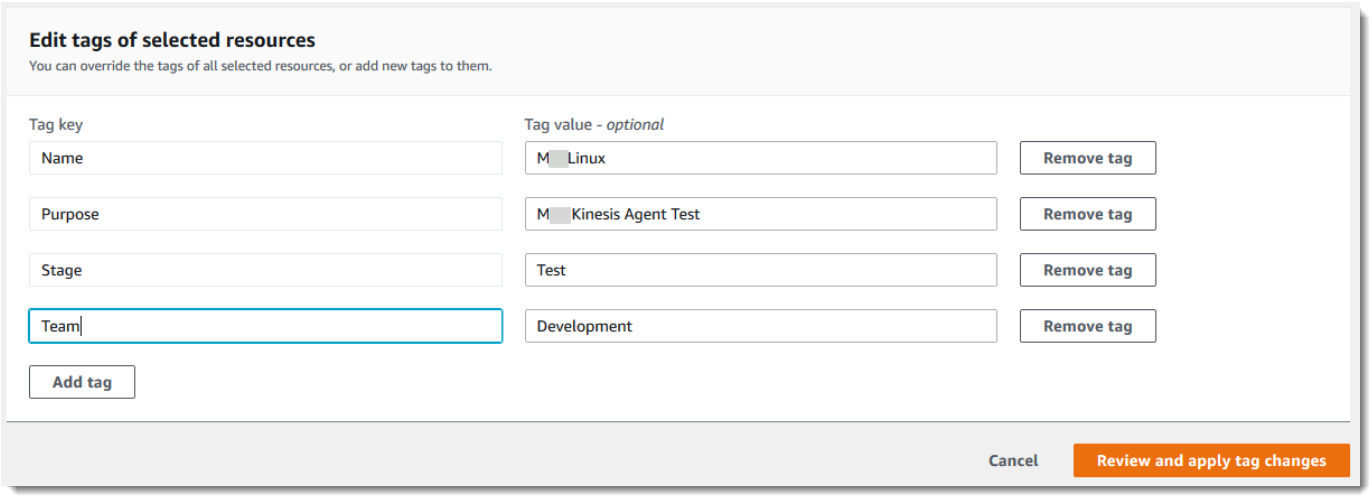
> **Important**
>
> 개인 식별 정보(PII)나 기타 기밀 정보 또는 민감한 정보를 태그에 저장하지 마세요. 당사는 태그를 사용하여 청구 및 관리 서비스를 제공합니다. 태그는 개인 데이터나 민감한 데이터에 사용하기 위한 것이 아닙니다.

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 ID

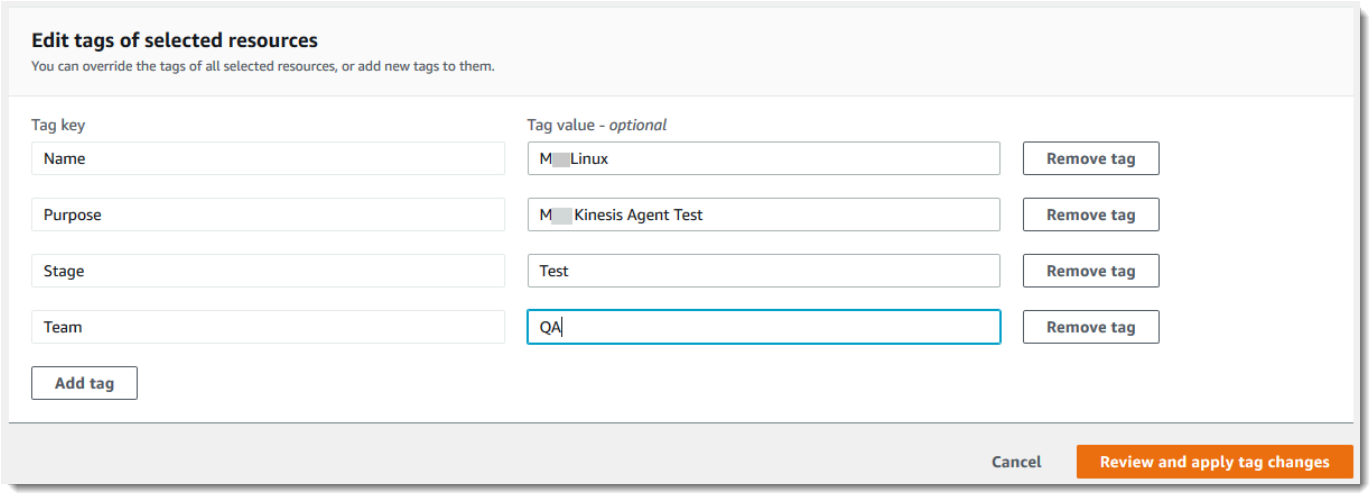
의 일부를 필터링합니다. 태그 열에서 결과의 리소스에 태그는 이미 적용되어 있습니다. 다음 예제에서, 처음에 선택된 EC2 인스턴스에는 이미 태그가 두 개 있습니다.



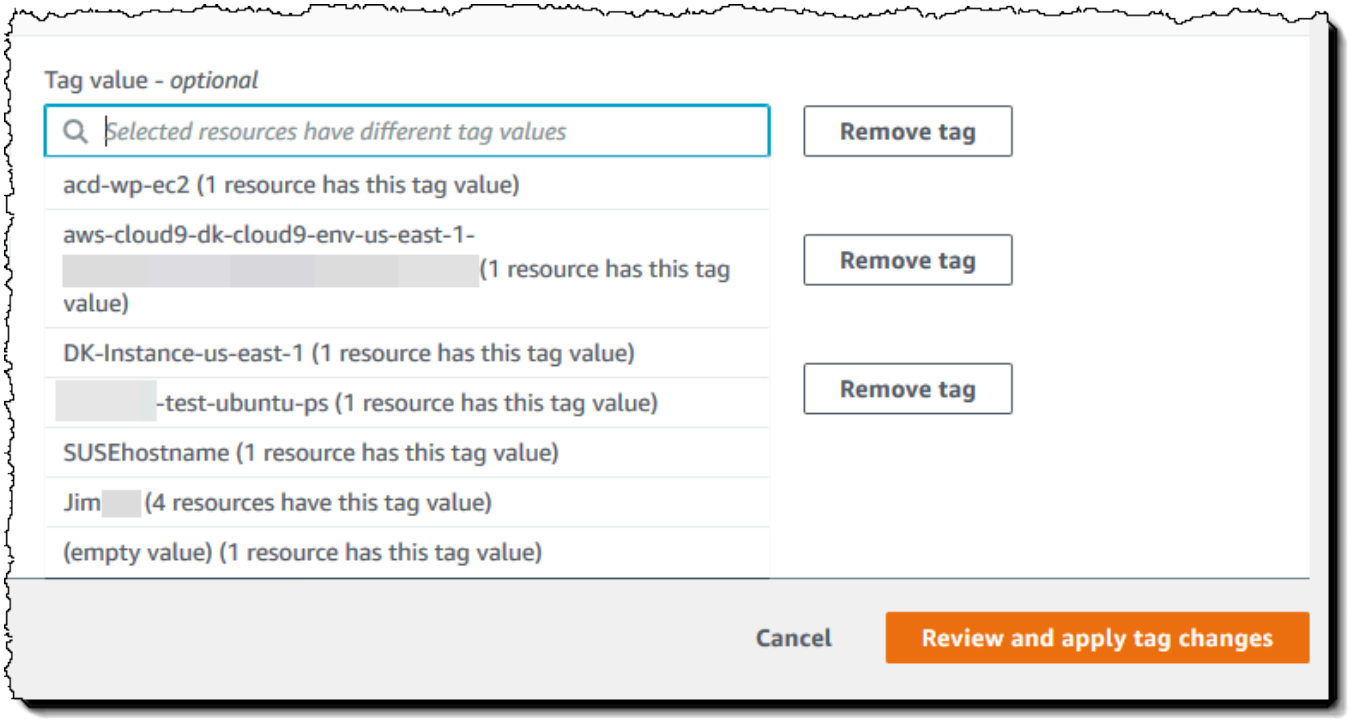
2. Manage tags of the selected resources(선택된 리소스의 태그 관리)를 선택합니다.
3. 태그 관리 페이지의 Edit tags of selected resources(선택된 리소스의 태그 편집)에서 선택한 리소스에 대한 태그를 봅니다. 원본 쿼리가 더 많은 리소스를 반환했을 수 있지만, 1단계에서 선택한 리소스에만 태그를 변경하는 것입니다.



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.



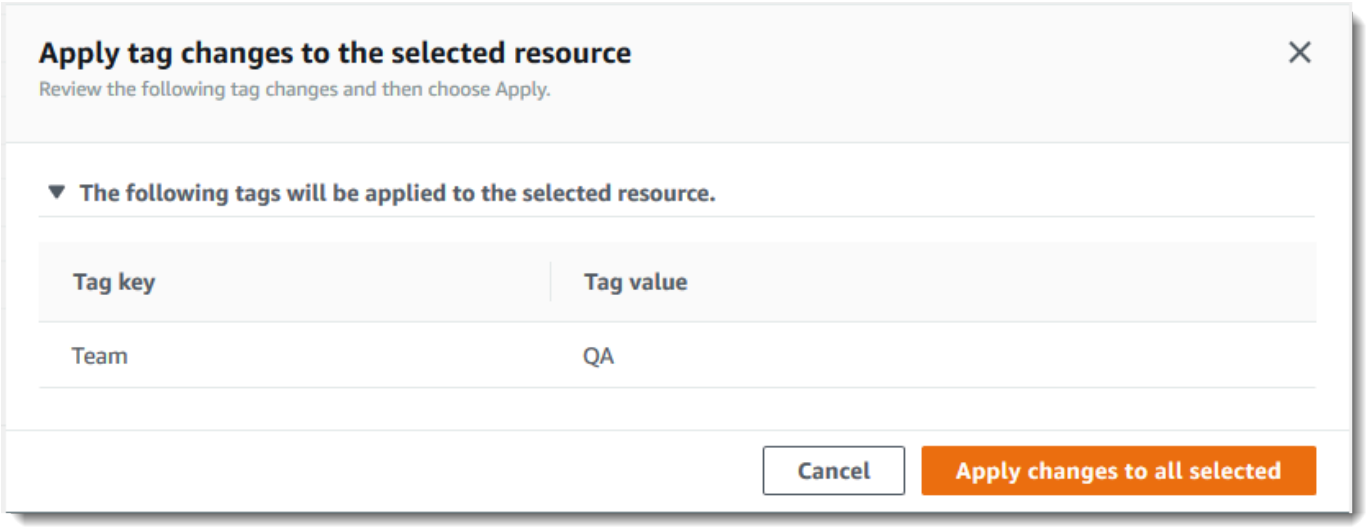
선택한 리소스가 동일한 키에 대해 다른 값을 가지고 있는 경우, 선택된 리소스는 다른 태그 값을 가지고 있음이 태그 값 필드에 표시됩니다. 이 경우 상자에 커서를 두면 선택된 리소스에서 이 태그 키에 대한 사용 가능한 모든 값의 드롭다운 목록이 열립니다.



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

QA로 변경됩니다. **Team** 태그가 없는 선택된 리소스에 대해 **QA** 값이 있는 **Team** 태그가 추가됩니다.

5. 태그 변경을 마치면 변경 사항 검토 및 적용를 선택합니다.
6. 변경 사항을 수락할 경우 선택된 모든 것에 변경 사항 적용를 선택합니다.



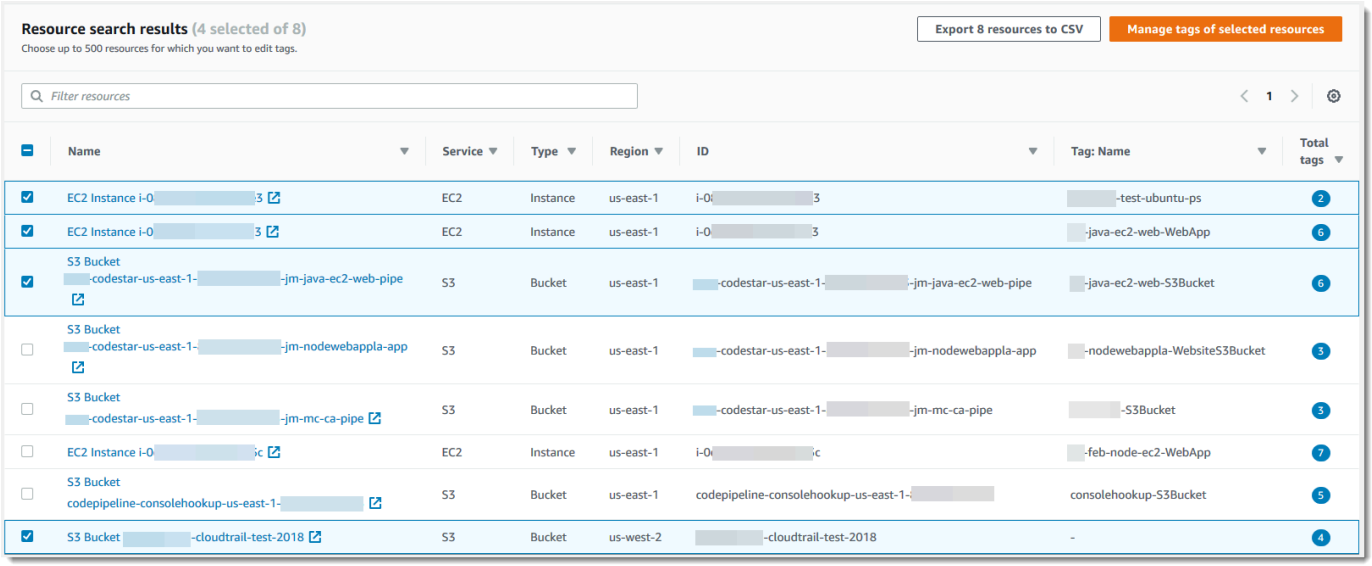
7. 선택한 리소스의 수에 따라 태그 편집에는 몇 분이 걸릴 수 있습니다. 동일한 브라우저 탭에서 페이지를 나가거나 다른 페이지를 열지 마십시오. 성공적으로 변경된 경우 페이지의 상단에 녹색 성공 배너가 표시됩니다. 계속하기 전에 성공 또는 실패 배너가 페이지에 표시될 때까지 기다립니다.

   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 설명은 the section called “실패한 태그 변경 재시도” 섹션을 참조하세요.

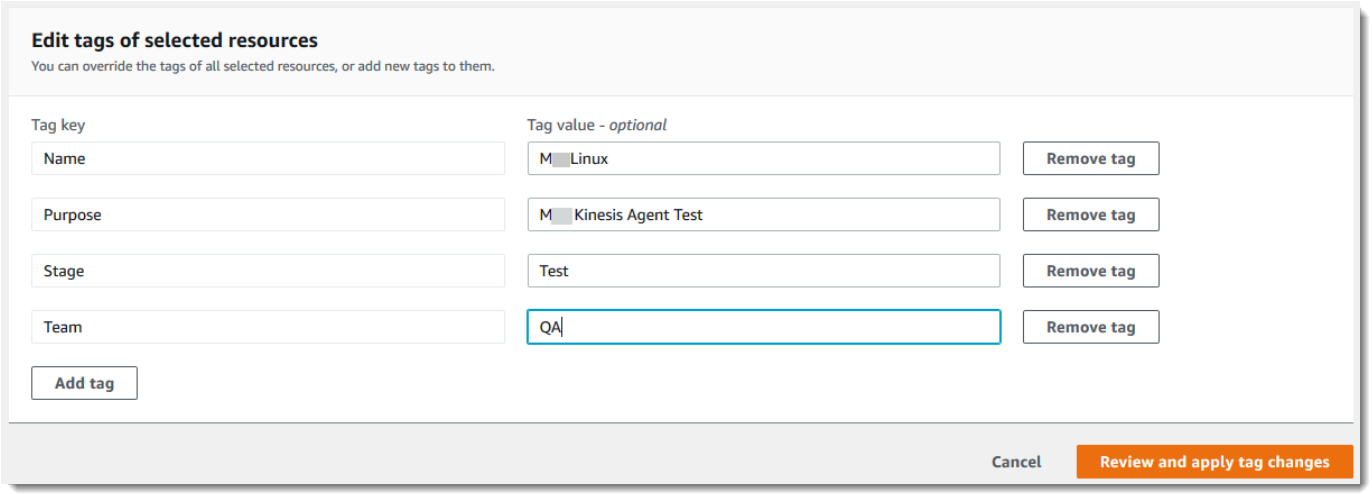
선택된 리소스에서 태그 제거

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.

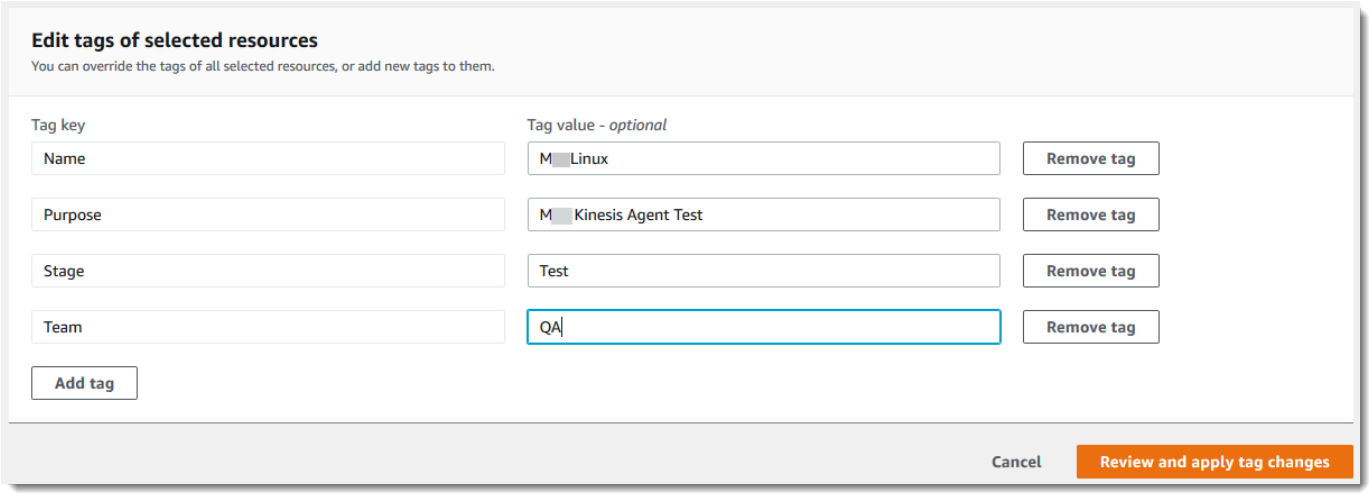


2. Manage tags of the selected resources(선택된 리소스의 태그 관리)를 선택합니다.
3. 태그 관리 페이지의 Edit tags of selected resources(선택된 리소스의 태그 편집)에서 선택한 리소스에 대한 태그를 봅니다. 원본 쿼리가 더 많은 리소스를 반환했을 수 있지만, 1단계에서 선택한 리소스에만 태그를 변경하는 것입니다.



4. 삭제할 태그 옆의 태그 제거를 선택합니다. 이 절차에서는 **Team** 태그를 제거합니다.

> **Note**
>
> 태그 제거를 선택하면 해당 태그가 있는 선택된 모든 리소스에서 태그가 제거됩니다. 표시된 예제에서 이 경우 태그의 값에 상관없이 현재 **Team** 태그가 있는 선택된 모든 리소스에서 **Team** 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 설명은 the section called “실패한 태그 변경 재시도” 섹션을 참조하세요.

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.

관련 정보

- AWS Billing 사용 설명서의 비용 할당 태그 사용

IAM 권한 정책에서 태그 사용

AWS Identity and Access Management (IAM)은 AWS 리소스에 액세스할 수 있는 사용자를 결정하는 권한 정책을 만들고 관리하는 데 사용하는 AWS 서비스입니다. AWS 서비스에 액세스하거나 AWS 리소스를 읽거나 쓰려는 모든 시도는 IAM 정책을 통해 액세스를 제어합니다.

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 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 열에 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

Example 예 1: 사용자가 리소스를 생성할 때 특정 태그를 첨부하도록 강제합니다.

다음 예에 있는 IAM 권한 정책에서는 IAM 정책의 태그를 생성하거나 수정하는 사용자에게 키 `Owner`와 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.

```
{
    "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}"}
            }
        }
    ]
}
```

Example 예 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.

```
{
  "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 Management사용 설명서의 다음 주제를 참조하십시오.

- 태그를 사용한 AWS 리소스 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 설명합니다.

AWS Organizations 태그 정책

태그 정책은 사용자가 AWS Organizations에서 생성하는 정책 유형입니다. 태그 정책을 사용하여 조직의 계정에 있는 리소스 전체에서 태그를 표준화할 수 있습니다. 태그 정책을 사용하려면 AWS Organizations사용 설명서의 태그 정책 시작하기에 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.

주제

- 사전 조건 및 권한
- 계정의 규정 준수 평가
- 조직 전체의 정책 준수 평가

사전 조건 및 권한

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

태그 정책 준수 여부를 평가하기 위한 사전 조건

태그 정책 준수 여부를 평가하려면 다음이 필요합니다.

- 먼저 AWS Organizations에서 기능을 활성화하고 태그 정책을 생성하여 첨부해야 합니다. 자세한 정보는 AWS Organizations사용 설명서에서 다음 페이지를 참조하세요.
  - 태그 정책 관리를 위한 사전 요구 사항 및 권한
  - 태그 정책 활성화
  - 태그 정책 시작하기
- 계정 리소스에서 규정을 준수하지 않는 태그를 찾으려면 해당 계정의 로그인 보안 인증 정보와계정에 대한 규정 준수 평가 권한에 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 준수 여부를 평가할 수 있는 권한을 제공합니다.

```
{
    "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 설명서를 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.

다음 예제 IAM 정책은 조직 전체의 규정 준수 여부를 평가할 수 있는 권한을 제공합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EvaluateOrgCompliance",
            "Effect": "Allow",
            "Action": [
                "organizations:DescribeEffectivePolicy",
                "tag:GetComplianceSummary",
                "tag:StartReportCreation",
                "tag:DescribeReportCreation"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

IAM 정책 및 권한에 대한 자세한 내용은 IAM 사용 설명서를 참조하세요.

보고서 저장에 대한 Amazon S3 버킷 정책

조직 전체의 규정 준수 보고서를 만들려면 보고서 저장을 위해 미국 동부(버지니아 북부) 리전의 Amazon Simple Storage Service(S3) 버킷에 태그 정책 서비스 주체에 대한 액세스 권한을 부여해야 합니다. 버킷에 다음 버킷 정책을 첨부하고 각 *## ###*를 다음과 같은 자체 정보로 대체합니다.

- S3 버킷 이름
- 조직의 ID 번호
- 정책을 적용하려는 조직의 관리 계정의 계정 ID 번호

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TagPolicyACL",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "tagpolicies.tag.amazonaws.com"
                 ]
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::<your-bucket-name>",
            "Condition": {
                "StringLike": {
                    "aws:SourceAccount": "<organization-management-account-id>",
                    "aws:SourceArn": "arn:aws:tag:us-east-1:<organization-management-
account-id>:*"
                }
            }
         },
         {
            "Sid": "TagPolicyBucketDelivery",
            "Effect": "Allow",
```

```
            "Principal": {
                "Service": [
                    "tagpolicies.tag.amazonaws.com"
                 ]
            },
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::<your-bucket-name>/AwsTagPolicies/<your-
organization-id>/*",
            "Condition": {
                "StringLike": {
                    "aws:SourceAccount": "<organization-management-account-id>",
                    "aws:SourceArn": "arn:aws:tag:us-east-1:<organization-management-
account-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 **tag:none**를 사용하는 쿼리와 함께 AWS 리소스 탐색기를 사용합니다. 자세한 내용은 AWS 리소스 탐색기 사용 설명서의 태그 없는 리소스 검색을 참조하세요.

유효 태그 정책은 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 Organizations사용 설명서의 태그 정책 시작하기를 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 콘솔을 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. 규정을 준수하지 않는 태그가 있는 리소스에서 규정 미준수 태그를 위해 검색할 AWS 리전을 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 아이콘
   (
)
   을 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.

규정을 준수하지 않는 태그(AWS CLI, AWS API)를 찾는 방법

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

  AWS CLI에서 태그 정책을 사용하는 절차는 AWS Organizations사용 설명서의 AWS CLI에 있는 태그 사용 정책을 참조하십시오.
- 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 평가을(를) 참조하세요.

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 **tag:none**를 사용하는 쿼리와 함께 AWS 리소스 탐색기를 사용합니다. 자세한 내용은 AWS 리소스 탐색기 사용 설명서의 태그 없는 리소스 검색을 참조하세요.

us-east-1 AWS 리전의 조직 관리 계정에서만 보고서를 생성할 수 있습니다. 보고서를 생성하는 계정은 미국 동부(버지니아 북부) 리전의 Amazon S3 버킷에 대한 액세스 권한이 있어야 합니다. 버킷에는 보고서 저장을 위한 Amazon S3 버킷 정책에 표시된 것과 같은 연결된 버킷 정책이 있어야 합니다.

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

- `organizations:DescribeEffectivePolicy`
- `tag:StartReportCreation`
- `tag:DescribeReportCreation`
- `tag:GetComplianceSummary`

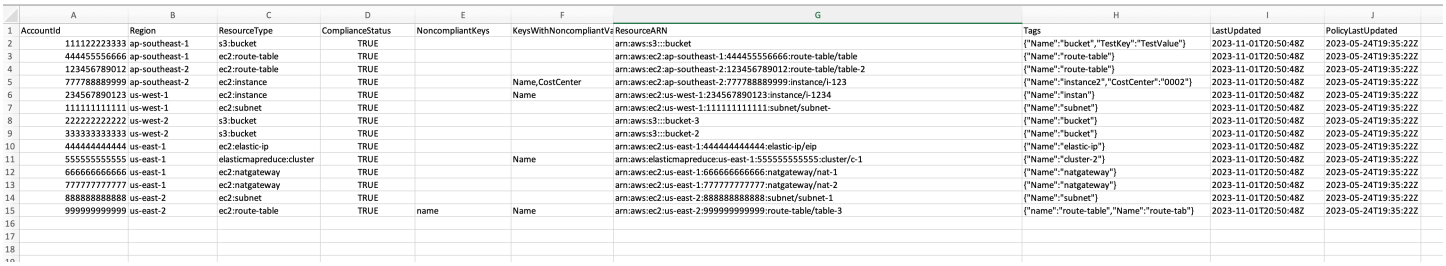
조직 전체의 정책 준수 보고서를 생성하려면(콘솔)

1. 태그 정책 콘솔을 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.

다음 그림에서는 보고서 예제 발췌문을 보여줍니다.



	A	B	C	D	E	F	G	H	I	J
1	AccountId	Region	ResourceType	ComplianceStatus	NoncompliantKeys	KeysWithNoncompliantVa	ResourceARN	Tags	LastUpdated	PolicyLastUpdated
2	111122223333	ap-southeast-1	s3:bucket	TRUE			arn:aws:s3:::bucket	{"Name":"bucket","TestKey":"TestValue"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
3	444455556666	ap-southeast-1	ec2:route-table	TRUE			arn:aws:ec2:ap-southeast-1:444455556666:route-table/table	{"Name":"route-table"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
4	123456789012	ap-southeast-2	ec2:route-table	TRUE			arn:aws:ec2:ap-southeast-2:123456789012:route-table/table-2	{"Name":"route-table"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
5	777788889999	ap-southeast-2	ec2:instance	TRUE		Name,CostCenter	arn:aws:ec2:ap-southeast-2:777788889999:instance/i-123	{"Name":"instance2","CostCenter":"0002"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
6	234567890123	us-west-1	ec2:instance	TRUE		Name	arn:aws:ec2:us-west-1:234567890123:instance/i-1234	{"Name":"instan"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
7	111111111111	us-west-1	ec2:subnet	TRUE			arn:aws:ec2:us-west-1:111111111111:subnet/subnet-	{"Name":"subnet"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
8	222222222222	us-west-2	s3:bucket	TRUE			arn:aws:s3:::bucket-3	{"Name":"bucket"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
9	333333333333	us-west-2	s3:bucket	TRUE			arn:aws:s3:::bucket-2	{"Name":"bucket"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
10	444444444444	us-east-1	ec2:elastic-ip	TRUE			arn:aws:ec2:us-east-1:444444444444:elastic-ip/eip	{"Name":"elastic-ip"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
11	555555555555	us-east-1	elasticmapreduce:cluster	TRUE		Name	arn:aws:elasticmapreduce:us-east-1:555555555555:cluster/c-1	{"Name":"cluster-2"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
12	666666666666	us-east-1	ec2:natgateway	TRUE			arn:aws:ec2:us-east-1:666666666666:natgateway/nat-1	{"Name":"natgateway"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
13	777777777777	us-east-1	ec2:natgateway	TRUE			arn:aws:ec2:us-east-1:777777777777:natgateway/nat-2	{"Name":"natgateway"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
14	888888888888	us-east-2	ec2:subnet	TRUE			arn:aws:ec2:us-east-2:888888888888:subnet/subnet-1	{"Name":"subnet"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
15	999999999999	us-east-2	ec2:route-table	TRUE	name	Name	arn:aws:ec2:us-east-2:999999999999:route-table/table-3	{"name":"route-table","Name":"route-tab"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
16										
17										
18										

> **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.
> - 이 NoncompliantKeys열에는 유효 태그 정책을 준수하지 않는 리소스의 태그 키가 나열됩니다.
> - 이 KeysWithNonCompliantValues열에는 유효 정책에 정의된 키 중 리소스에 있는 키 중 대/소문자 처리가 잘못되었거나 값을 준수하지 않는 키가 나열됩니다.
> - 조직의 구성원이었던 AWS 계정을 닫더라도 최대 90일 동안 태그 규정 준수 보고서에 계속 표시될 수 있습니다.

조직 전체의 정책 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

AWS CLI에서 태그 정책을 사용하는 절차는 AWS Organizations사용 설명서의 AWS CLI에 있는 태그 사용 정책을 참조하십시오.

- AWS API:
  - StartReportCreation
  - DescribeReportCreation
  - GetComplianceSummary

서버리스 워크플로와 Amazon으로 태그 변경을 모니터링합니다. EventBridge

EventBridge Amazon은 AWS 리소스에 대한 태그 변경을 지원합니다. 이 EventBridge 유형을 사용하면 태그 변경과 일치하는 EventBridge 규칙을 구축하고 이벤트를 하나 이상의 대상으로 라우팅할 수 있습니다. 예를 들어, 대상은 자동화된 워크플로우를 간접적으로 호출하는 AWS Lambda 함수일 수 있습니다. 이 주제에서는 Lambda를 사용하여 AWS 리소스의 태그 변경 사항을 안전하게 처리하는 비용 효율적인 서버리스 솔루션을 구축하는 방법에 대한 자습서를 제공합니다.

태그를 변경하면 EventBridge 이벤트가 생성됩니다.

EventBridge AWS리소스 변경을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다. 많은 AWS 리소스가 AWS 리소스를 쉽게 구성하고 분류할 수 있는 사용자 정의 속성인 태그를 지원합니다. 태그의 일반적인 사용 사례로는 비용 할당 분류, 액세스 제어 보안, 자동화 등이 있습니다.

를 사용하면 태그 변경 사항을 모니터링하고 AWS 리소스의 태그 상태를 추적할 수 있습니다. EventBridge 이전에는 유사한 기능을 구현하기 위해 API를 지속적으로 폴링하고 여러 호출을 오케스트레이션했을 수 있습니다. 이제 개별 서비스 API, Tag Editor, 태그 지정 API를 포함하여 태그가 변경되면 리소스 이벤트에서 태그 변경이 시작됩니다. 다음 예제는 태그 변경으로 인해 EventBridge 발생하는 일반적인 이벤트를 보여줍니다. 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-0000000aaaaaaaaaa"
  ],
  "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 이벤트의 최상위 필드는 동일합니다.

- 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.tag입니다.
- 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.
  - changed-tag-keys— 이 이벤트로 변경된 태그 키.
  - 서비스 - 리소스가 속한 서비스입니다. 이 예에서 서비스는 ec2이며, 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 패턴을 참조하십시오.

를 사용하여 EventBridge 다양한 필드를 기반으로 특정 이벤트 패턴과 일치하는 규칙을 생성할 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 시작하는 인스턴스의 태그 이벤트와 일치하는 패턴을 생성합니다.

Lambda 및 서버리스

AWS Lambda는 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. 여기서 서버리스란 코드를 실행하는 데 사용되는 서버를 프로비저닝, 구성 또는 관리할 필요가 없음을 의미합니다. AWS가 이 모든 작업을 대신 수행하므로 코드에만 집중할 수 있습니다. Lambda에 대한 자세한 내용은 AWS Lambda 제품 개요를 참조하십시오.

자습서: 필수 태그가 누락된 Amazon EC2 인스턴스 자동 중지하기

주제

관리하는 AWS 리소스 및 AWS 계정 풀이 커지면 태그를 사용하여 리소스를 더 쉽게 분류할 수 있습니다. 태그는 일반적으로 비용 할당 및 보안과 같은 중요한 사용 사례에 사용됩니다. AWS 리소스를 효과적으로 관리하려면 일관성 있게 리소스에 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 있

습니다. 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 하면 AWS 계정의 리전에 있는 모든 인스턴스를 실수로 종료하지 않도록 할 수 있습니다.
> 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 예에서는 예 AWS 계정 ID 123456789012와 AWS 리전 미국 동부(버지니아 북부)(`us-east-1`)를 사용합니다. 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-1`와 다른 리전을 사용하는 경우, 다음 코드 예시에 있는 모든 참조를 선택한 리전으로 변경해야 합니다.

단계 1. Lambda 함수 생성

Lambda 함수를 생성하려면

1. AWS Lambda 관리 콘솔을 엽니다.

2. 함수 생성을 선택한 다음 새로 작성를 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 열기을 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 succesfully 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-0000000aaaaaaaaaa",
    "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 에서 태그 변경 이벤트가 보고한 대로 Amazon EC2 인스턴스의 태그를 확인합니다. EventBridge 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.

단계 2. 필요한 IAM 권한 설정

함수를 성공적으로 실행하려면 먼저 함수에 EC2 인스턴스를 중지할 수 있는 권한을 부여해야 합니다. AWS에서 제공하는 역할 lambda_basic_execution에는 해당 권한이 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 수정하기를 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 편집를 선택합니다.
5. 시각적 편집기 탭에서 추가 권한 추가를 선택합니다.
6. 서비스에서 EC2를 선택합니다.
7. [Actions] 에서 선택합니다. StopInstances 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.

```
{
    "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:*"
        }
    ]
}
```

단계 3. Lambda 함수의 예비 테스트 수행

이 단계에서는 함수에 테스트 이벤트를 제출합니다. Lambda 테스트 기능은 수동으로 제공된 테스트 이벤트를 제출하는 방식입니다. 함수는 이벤트가 발생한 것처럼 테스트 이벤트를 처리합니다 EventBridge. 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-0000000aaaaaaaaaa"
  ],
  "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-0000000aaaaaaaaaa' 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 중지를 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-0000000aaaaaaaaaa )
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-0000000aaaaaaaaaa' does not
 exist","code":"InvalidInstanceID.NotFound","message":"The instance ID
 'i-0000000aaaaaaaaaa' 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-0000000aaaaaaaaaa' 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
```

6. 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 SampleBadTagChangeEvent테스트 이벤트가 표시됩니다.

7. 새 역할 생성을 선택합니다.

8. 이벤트 이름에 **SampleGoodTagChangeEvent**를 입력합니다.

9. 17행에서 **NOT-**를 삭제하여 값을 **valid-value**로 변경합니다.

10. 테스트 이벤트 창 상단에서 저장를 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-0000000aaaaaaaaaa )
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 시작하는 EventBridge 규칙을 생성합니다.

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

규칙을 생성하려면 EventBridge

1. 다른 브라우저 탭 또는 창에서 EventBridge 콘솔을 열어 규칙 만들기 페이지를 엽니다.
2. 이름에 **ec2-instance-rule**을 입력한 후 다음를 선택합니다.
3. 아래로 스크롤하여 생성 방법으로 이동한 다음 사용자 정의 패턴(JSON 편집기)을 선택합니다.
4. 편집 상자에 다음 패턴 텍스트를 붙여넣고 다음를 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 다음를 선택합니다.
7. 태그 구성 페이지에서 다음를 선택합니다. 검토 및 생성 페이지에서 규칙 생성을 선택합니다. 또한 지정된 Lambda 함수를 EventBridge 호출할 수 있는 권한을 자동으로 부여합니다.

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 콘솔의 인스턴스 페이지를 엽니다.
2. Amazon EC2 인스턴스 생성 실행하기 전에 키 `valid-key`와 값 `valid-value`이 포함된 태그를 지정합니다. 인스턴스를 생성하고 시작하는 다른 방법에 대한 자세한 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 1단계: 인스턴스 시작을 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. InstanceId콘솔에서 복사하십시오.
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 `InstanceList`의 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 Linux 인스턴스용 Amazon EC2 사용 설명서의 개별 리소스에 태그 추가 및 삭제를 참조하십시오.

9. 몇 초간 기다린 다음 콘솔을 새로 고침합니다. 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 호출 테이블에서 LogStream열의 가장 최근 항목을 선택합니다.

    Amazon CloudWatch 콘솔에서 Lambda 함수를 마지막으로 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
```

요약

이 자습서에서는 Amazon EC2 인스턴스의 리소스 이벤트에서 태그 변경에 대응하도록 EventBridge 규칙을 생성하는 방법을 보여 주었습니다. 이 규칙은 필요한 태그가 없는 경우 인스턴스가 자동으로 종료되는 Lambda 함수를 가리키고 있습니다.

리소스의 태그 변경을 Amazon에서 EventBridge 지원하므로 많은 AWS 리소스에서 이벤트 기반 자동화를 구축할 수 있습니다. AWS 서비스 이 기능을 AWS Lambda와 결합하면 AWS 리소스에 안전하게

액세스하고 필요에 따라 확장하며 비용 효율적인 서버리스 솔루션을 구축할 수 있는 도구가 제공됩니다.

tag-change-on-resource EventBridge 이벤트의 기타 가능한 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 (태그) 기반 액세스 제어 (ABAC) 를 사용하여 리소스에 대한 액세스를 설정한 경우 태그 변경으로 생성된 EventBridge 이벤트를 사용하여 보안 팀의 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 보유할 수 있습니다. AWS에서 생성된 태그는 최대 50개 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, AWS 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.
- AWS Resource Groups Tagging API에는 속도 제한이 있지만 태그를 지정하는 서비스에 별도의 한도가 부과될 수 있으며, 이 한도는 리소스 그룹 API 태그 지정 한도보다 먼저 도달할 수 있습니다.

관련 정보

- AWS Billing 사용 설명서의 비용 할당 태그 사용

Tag Editor 보안

AWS에서는 클라우드 보안을 가장 중요하게 생각합니다. 여러분은 AWS 고객으로서 보안에 민감한 기관의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 AWS와(과) 귀하의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드 내 클라우드 및 보안의 보안으로 설명합니다.

- 클라우드의 보안 - AWS는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호합니다. AWS는 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사원은 정기적으로 AWS 규정 준수 프로그램의 일환으로 보안 효과를 테스트하고 확인합니다. Tag Editor에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램 제공 범위 내의 AWS 서비스를 참조하세요.
- 클라우드 내 보안 – 사용자의 책임은 사용자가 사용하는 AWS 서비스에 의해 결정됩니다. 또한 귀하는 귀사의 데이터의 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

AWS 공동 책임 모델은 Tag Editor의 데이터 보호에 적용됩니다. 이 모델이 설명하는 것처럼 AWS는 모든 AWS 클라우드를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 FAQ를 참조하십시오. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공유 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터를 보호하려면 AWS 계정 보안 인증 정보를 보호하고 AWS IAM Identity Center또는 AWS Identity and Access Management(IAM)를 통해 개별 사용자 계정을 설정하는 것이 좋습니다. 이 방식을 사용하면 각 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS리소스와 통신합니다. TLS 1.2가 필수이며 TLS 1.3을 권장합니다.
- AWS CloudTrail(으)로 API 및 사용자 활동 로깅을 설정합니다.
- AWS 암호화 솔루션을 AWS 서비스내의 모든 기본 보안 컨트롤과 함께 사용합니다.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 API를 통해 AWS에 액세스할 때 FIPS 140-2 검증된 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 Federal Information Processing Standard(FIPS) 140-2 섹션을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 Tag Editor 또는 기타 AWS 서비스에서 콘솔, API, AWS CLI 또는 AWS SDK를 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 보안 인증를 URL에 포함시켜서는 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 전용 격리 방법을 사용하십시오. 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 Service와 통합되어 있지 않으며 AWS KMS keys을(를) 지원하지 않습니다.

인터네트워크 트래픽 개인 정보

Tag Editor는 Tag Editor 사용자와 AWS 간의 모든 전송에서 HTTPS를 사용합니다. 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)은 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있도록 지원하는 AWS 서비스입니다. IAM 관리자는 누가 Tag Editor 리소스를 사용하도록 인증되고(로그인됨) 권한이 부여되는지(권한 있음)를 제어합니다. IAM은 추가 비용 없이 사용할 수 있는 AWS 서비스입니다.

주제

고객

AWS Identity and Access Management (IAM)을 사용하는 방법은 Tag Editor에서 수행하는 작업에 따라 달라집니다.

서비스 사용자 - Tag Editor 서비스를 사용하여 작업을 수행하는 경우, 필요한 보안 인증 정보와 권한을 관리자가 제공합니다. 더 많은 Tag Editor 기능을 사용하여 작업을 수행하게 되면 추가 권한이 필요할

수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. Tag Editor의 기능에 액세스할 수 없는 경우, Tag Editor 자격 증명 및 액세스 문제 해결 단원을 참조하세요.

서비스 관리자 - 회사에서 Tag Editor 리소스를 책임지고 있는 경우 Tag Editor에 대한 전체 액세스 권한을 가지고 있을 것입니다. 서비스 관리자는 서비스 사용자가 액세스해야 하는 Tag Editor 기능과 리소스를 결정합니다. 그런 다음, IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하십시오. 회사가 Tag Editor에서 IAM을 사용하는 방법에 대해 자세히 알아보려면 Tag Editor가 IAM과 작동하는 방식을(를) 참조하세요.

IAM 관리자 - IAM 관리자라면 Tag Editor에 대한 액세스 권한 관리 정책 작성 방법을 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 Tag Editor 자격 증명 기반 정책 예제를 보려면 Tag Editor 자격 증명 기반 정책 예제 단원을 참조하세요.

보안 인증 정보를 통한 인증

인증은 ID 보안 인증 정보를 사용하여 AWS에 로그인하는 방식입니다. AWS 계정 루트 사용자이나 IAM 사용자로 또는 IAM 역할을 수임하여 인증(AWS에 로그인)되어야 합니다.

보안 인증 정보 소스를 통해 제공된 보안 인증 정보를 사용하여 페더레이션형 ID로 AWS에 로그인할 수 있습니다. AWS IAM Identity Center (IAM Identity Center) 사용자, 회사의 Single Sign-On 인증, Google 또는 Facebook 보안 인증 정보가 페더레이션형 ID의 예입니다. 페더레이션형 ID로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 페더레이션을 사용하여 AWS에 액세스하면 간접적으로 역할을 수임합니다.

사용자 유형에 따라 AWS Management Console 또는AWS 액세스 포털에 로그인할 수 있습니다. AWS에 로그인하는 방법에 대한 자세한 내용은 AWS 로그인사용 설명서의 AWS 계정에 로그인하는 방법을 참조하세요.

AWS에 프로그래밍 방식으로 액세스하는 경우, AWS에서는 보안 인증 정보를 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트(SDK) 및 명령줄 인터페이스(CLI)를 제공합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 요청에 직접 서명하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS API 요청에 서명을 참조하십시오.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, AWS는 다중 인증(MFA)을 사용하여 계정의 보안을 강화하는 것을 권장합니다. 자세한 내용은 AWS IAM Identity Center사용 설명서의 다중 인증 및 IAM 사용 설명서의 AWS에서 다중 인증(MFA) 사용을 참조하세요.

AWS 계정 루트 사용자

AWS 계정을 생성할 때는 해당 계정의 모든 AWS 서비스및 리소스에 대한 완전한 액세스 권한이 있는 단일 로그인 ID로 시작합니다. 이 자격 증명은 AWS 계정루트 사용자라고 하며, 계정을 생성할 때 사용

한 이메일 주소와 암호로 로그인하여 액세스합니다. 일상적인 작업에는 루트 사용자를 가급적 사용하지 않는 것이 좋습니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 작업을 수행하는 데 사용합니다. 루트 사용자로 로그인해야 하는 전체 작업 목록은 IAM 사용 설명서의 Tasks that require root user credentials를 참조하세요.

사용자 및 그룹

IAM 사용자는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가지고 있는 AWS 계정내 자격 증명입니다. 가능하면 암호 및 액세스 키와 같은 장기 보안 인증이 있는 IAM 사용자를 생성하는 대신 임시 보안 인증을 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 자격 증명이 필요한 특정 사용 사례가 있는 경우 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 장기 보안 인증이 필요한 사용 사례의 경우 정기적으로 액세스 키 교체를 참조하십시오.

IAM 그룹은 IAM 사용자 컬렉션을 지정하는 자격 증명입니다. 귀하는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어, IAMAdmins(이)라는 그룹이 있고 이 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 보안 인증을 가지고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세한 정보는 IAM 사용 설명서의 IAM 사용자를 만들어야 하는 경우(역할이 아님)를 참조하세요.

역할

IAM 역할은 특정 권한을 가지고 있는 AWS 계정계정 내 ID입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. 역할 전환하여 AWS Management Console에서 IAM 역할을 임시로 수임할 수 있습니다. AWS CLI 또는 AWSAPI 태스크를 호출하거나 사용자 지정 URL을 사용하여 역할을 수임할 수 있습니다. 역할 사용 방법에 대한 자세한 정보는 IAM 사용 설명서의 IAM 역할 사용을 참조하십시오.

임시 보안 인증 정보가 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 자격 증명에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 자격 증명이 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 Creating a role for a third-party Identity Provider(서드 파티 자격 증명 공급자의 역할 만들기) 부분을 참조하세요. IAM Identity Center를 사용하는 경우 권한 세트를 구성합니다. 인증 후 보안 인증 정보에서 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 집합을 IAM의 역할과 연결합니다. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center사용 설명서의 권한 세트를 참조하세요.

- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수임하여 특정 작업에 대한 다양한 권한을 임시로 받을 수 있습니다.
- 크로스 계정 액세스: IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 AWS 서비스를 사용하면 역할을(프록시로 사용하는 대신) 리소스에 정책을 직접 연결할 수 있습니다. 크로스 계정 액세스를 위한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 IAM 역할과 리소스 기반 정책의 차이를 참조하십시오.
- 교차 서비스 액세스 - 일부 AWS 서비스는 다른 AWS 서비스의 기능을 사용합니다. 예를 들어 서비스에서 직접적으로 호출하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거나 Amazon S3에 객체를 저장합니다. 서비스는 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 작업을 수행할 수 있습니다.
  - 전달 액세스 세션(FAS) - IAM 사용자 또는 역할을 사용하여 AWS에서 작업을 수행하는 사람은 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 AWS 서비스를 직접 호출하는 보안 주체의 권한과 요청하는 AWS 서비스를 함께 사용하여 다운스트림 서비스에 대한 요청을 수행합니다. FAS 요청은 서비스에서 완료를 위해 다른 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 받은 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하십시오.
  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 태스크를 수행하기 위해 맡는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하십시오.
  - 서비스 연결 역할 - 서비스 연결 역할은 AWS 서비스에 연결된 서비스 역할의 한 유형입니다. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 AWS 계정에 나타나고, 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집할 수는 없습니다.
- Amazon EC2에서 실행 중인 애플리케이션 – IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS CLI또는 AWSAPI 요청을 수행하는 애플리케이션의 임시 보안 인증 정보를 관리할 수 있습니다. 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS역할을 할당하고 해당 역할을 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인증을 얻을 수 있습니다. 자세한 정보는 IAM 사용 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 참조하십시오.

IAM 역할을 사용할지 또는 IAM 사용자를 사용할지를 알아보려면 IAM 사용 설명서의 IAM 역할(사용자 대신)을 생성하는 경우를 참조하십시오.

정책을 사용한 액세스 관리

정책을 생성하고 AWSID 또는 리소스에 연결하여 AWS내 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결될 때 해당 권한을 정의하는 AWS의 객체입니다. AWS는 보안 주체(사용자, 루트 사용자 또는 역할 세션)가 요청을 보낼 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되는지 또는 거부되는지를 결정합니다. 대부분의 정책은 AWS에 JSON 설명서로서 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 정보는 IAM 사용 설명서의 JSON 정책 개요를 참조하십시오.

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

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수임할 수 있습니다.

IAM 정책은 작업을 수행하기 위해 사용하는 방법과 상관없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole`태스크를 허용하는 정책이 있다고 가정합니다. 해당 정책이 있는 사용자는 AWS Management Console, AWS CLI또는 AWSAPI에서 역할 정보를 가져올 수 있습니다.

ID 기반 정책

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 자격 증명에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하십시오.

자격 증명 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 AWS 계정에 속한 다수의 사용자, 그룹 및 역할에 독립적으로 추가할 수 있는 정책입니다. 관리형 정책에는 AWS관리형 정책과 고객 관리형 정책이 포함되어 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책과 인라인 정책의 선택을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는 AWS 서비스가 포함될 수 있습니다.

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

액세스 제어 목록(ACL)

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

Amazon S3, AWS WAF및 Amazon VPC는 ACL을 지원하는 대표적인 서비스입니다. ACL에 대해 자세히 알아보려면 Amazon Simple Storage Service 개발자 안내서의 액세스 제어 목록(ACL) 개요를 참조하십시오.

기타 정책 유형

AWS은(는) 비교적 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 유형은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 – 권한 경계는 ID 기반 정책에 따라 IAM 엔터티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 엔터티의 ID 기반 정책 및 해당 권한 경계의 교집합입니다. `Principal`필드에서 사용자나 역할을 보안 주체로 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 정보는 IAM 사용 설명서의 IAM 엔터티에 대한 권한 경계를 참조하십시오.
- 서비스 제어 정책(SCP) – SCP는 AWS Organizations에서 조직 또는 조직 단위(OU)에 최대 권한을 지정하는 JSON 정책입니다. AWS Organizations은 기업이 소유하는 여러 개의 AWS 계정를 그룹화하고 중앙에서 관리하기 위한 서비스입니다. 조직에서 모든 기능을 활성화할 경우 서비스 제어 정책(SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 각 AWS 계정 루트 사용자을 비롯하여 멤버 계정의 엔터티에 대한 권한을 제한합니다. 조직 및 SCP에 대한 자세한 정보는 AWS Organizations사용 설명서의 SCP 작동 방식을 참조하세요.
- 세션 정책 – 세션 정책은 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 ID 기반 정책 및 세션 정책의 교집합입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 정보는 IAM 사용 설명서의 세션 정책을 참조하십시오.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련될 때 AWS가 요청을 허용할지를 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 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 기타 제품이 IAM과 어떻게 AWS 서비스 연동되는지 자세히 알아보려면 IAM 사용 설명서의 IAM과AWS 서비스 연동하는 방법을 참조하십시오.

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 사용 설명서의 IAM JSON 정책 요소 참조를 참조하십시오.

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. 정책 작업은 일반적으로 관련 AWS API 작업과 이름이 같습니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하십시오.

Tag Editor의 정책 작업은 작업 앞에 `tag:` 접두사를 사용합니다. Tag Editor 작업은 전적으로 콘솔에서 수행되지만 로그 항목에 접두사 `tag`가 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, `Get`라는 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 키를 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. 보고서에는 `Resource` 또는 `NotResource` 요소가 반드시 추가되어야 합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이 작업을 수행할 수 있습니다.

작업 나열과 같이 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` 요소(또는 `Condition` 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. `Condition` 요소는 옵션입니다. 같거나 적음 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 `Condition` 요소를 지정하거나 단일 `Condition` 요소에서 여러 키를 지정하는 경우 AWS (은)는 논리적 `AND` 작업을 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우 는 논리적 `OR` 연산을 사용하여 조건을 AWS 평가합니다. 명령문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예를 들어, IAM 사용자에게 IAM 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 정보는 IAM 사용자 설명서의 IAM 정책 요소: 변수 및 태그를 참조합니다.

AWS 글로벌 조건 키 및 서비스별 조건 키를 지원합니다. 모든 AWS 글로벌 조건 키를 보려면 IAM 사용 AWS 설명서의 글로벌 조건 컨텍스트 키를 참조하십시오.

Tag Editor는 서비스별 조건 키를 정의하지 않습니다.

예

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 요소에 태그 정보를 제공합니다. 리소스를 만들거나 업데이트할 때 리소스에 태그를 적용할 수 있습니다.

리소스의 태그를 기반으로 리소스에 대한 액세스를 제한하는 자격 증명 기반 정책의 예제는 태그를 기준으로 그룹 보기 섹션에서 확인할 수 있습니다. 속성 기반 액세스 제어 (ABAC) 에 대한 자세한 내용은 ABAC의 용도를 참조하십시오. AWS IAM 사용 설명서에서

Tag Editor IAM 역할

IAM 역할은 특정 권한을 AWS 계정 가진 사용자 내의 엔티티입니다. 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. AssumeRole또는 GetFederationToken와 같은 AWS STS API 작업을 호출하여 임시 보안 자격 증명을 얻습니다.

서비스 링크 역할

서비스 연결 역할을 사용하면 다른 서비스의 AWS 서비스 리소스에 액세스하여 사용자를 대신하여 작업을 완료할 수 있습니다.

Tag Editor에는 서비스 연결 역할이 없거나 이를 사용하지 않습니다.

Service roles(서비스 역할)

이 기능을 사용하면 서비스가 사용자를 대신하여 서비스 역할을 수임할 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 API를 사용해 태스크를 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 생성을 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 따릅니다.

- AWS 관리형 정책으로 시작하고 최소 권한을 향해 나아가기 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반 사용 사례에 대한 권한을 부여하는 AWS관리형 정책을 사용합니다. 관리형 정책은 AWS 계정에서 사용할 수 있습니다. 사용 사례에 고유한 AWS고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다. 자세한 정보는 IAM 사용 설명서의 AWSmanaged policies(관리형 정책) 또는 AWSmanaged policies for job functions(직무에 대한 관리형 정책)를 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 Policies and permissions in IAM(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. 특정 AWS 서비스(예: AWS CloudFormation)을(를) 통해 사용되는 경우에만 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다. 자세한 정보는 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 검증하여 안전하고 기능적인 권한 보장 – IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 IAM 모범 사례가 정책에서 준수되도록 신규 및 기존 정책을 검증합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 권장 사항을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 정보는 IAM 사용 설명서의 IAM Access Analyzer 정책 검증을 참조하십시오.
- 다중 인증(MFA) 필요 – AWS 계정계정에 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 설정합니다. API 작업을 직접적으로 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 정보는 IAM 사용 설명서의 Configuring MFA-protected API access(MFA 보호 API 액세스 구성)를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례 섹션을 참조하십시오.

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 권한은 AWS 계정에서 리소스에 연결된 태그에 대한 세부 정보를 나열하고 볼 수 있도록 허용

해야 합니다. 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 사용 설명서의 사용자에게 권한 추가를 참조하십시오.

```
{
  "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 예시는 IAM 사용자가 자신의 사용자 자격 증명에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 AWS CLI나 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:ListGroupsForUser",
                "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 방법을 보여줍니다. 단, 그룹 태그 `project`의 값이 호출하는 보안 주체에 연결된 `project` 태그와 동일한 값인 경우에만 권한이 부여됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "resource-groups:ListGroups",
            "Resource": "arn:aws:resource-groups::region:account_ID:group/group_name"
        },
        {
            "Effect": "Allow",
            "Action": "resource-groups:ListGroups",
```

```
            "Resource": "arn:aws:resource-groups::region:account_ID:group/group_name",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/project": "${aws:PrincipalTag/
project}"}
            }
        }
    ]
}
```

이 정책을 계정의 사용자에게 연결할 수 있습니다. 태그 키가 `project`이고 태그 값이 `alpha`인 사용자가 리소스 그룹을 보려면 해당 그룹에도 `project=alpha` 태그를 지정해야 합니다. 그렇지 않으면 사용자는 액세스가 거부됩니다. 조건 키 `project`은(는) 조건 키 이름이 대소문자를 구분하지 않기 때문에 `Project` 및 `project` 모두와 일치합니다. 자세한 정보는 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하십시오.

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
- IAM을 수행할 권한이 없습니다. PassRole

Tag Editor에서 작업을 수행할 권한이 없음

AWS Management Console에서 작업을 수행할 권한이 없다는 메시지가 나타나는 경우 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.

IAM을 수행할 권한이 없습니다. PassRole

`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 해당 서비스에 기존 역할을 전달할 수 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 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 수 있는 권한을 가지고 있지 않습니다.

```
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 AWS관리자에게 문의하세요. 관리자는 로그인 보안 인증을 제공한 사용자입니다.

Tag Editor에서의 로깅 및 모니터링

모든 Tag Editor 작업은 AWS CloudTrail에 기록됩니다.

를 사용하여 태그 편집기 API 호출 로깅 CloudTrail

태그 편집기는 사용자AWS CloudTrail, 역할 또는 태그 편집기에서 수행한 작업의 기록을 제공하는 서비스와 통합됩니다. AWS 서비스 CloudTrail 태그 편집기 콘솔에서의 호출 및 Resource Groups Tagging API에 대한 코드 호출을 포함하여 태그 편집기에 대한 모든 API 호출을 이벤트로 캡처합니다. 트레일을 생성하면 Tag Editor용 CloudTrail 이벤트를 포함하여 Amazon S3 버킷에 이벤트를 지속적으로 전송할 수 있습니다. 트레일을 구성하지 않아도 CloudTrail 콘솔의 이벤트 기록에서 가장 최근 이벤트를 계속 볼 수 있습니다. 에서 수집한 CloudTrail 정보를 사용하여 Tag Editor에 이루어진 요청, 요청이 이루어진 IP 주소, 요청한 사람, 요청 시기 및 추가 세부 정보를 확인할 수 있습니다.

에 대한 CloudTrail 자세한 내용은 AWS CloudTrail사용 설명서를 참조하십시오.

의 태그 편집기 정보 CloudTrail

CloudTrail 계정을 만들 AWS 계정 때 활성화됩니다. 태그 편집기나 태그 편집기 콘솔에서 활동이 발생하면 해당 활동이 CloudTrail 이벤트 기록의 다른 이벤트와 함께 AWS 서비스 이벤트에 기록됩니다.

AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 이벤트 기록으로 CloudTrail 이벤트 보기를 참조하십시오.

Tag Editor에 대한 이벤트를 포함하여 AWS 계정에 이벤트를 지속적으로 기록하려면 추적을 생성합니다. 트레일을 사용하면 CloudTrail Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 적용됩니다. 추적은 AWS 파티션에 있는 모든 영역의 이벤트를 로깅하고 지정된 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 이에 따라 조치를 AWS 서비스 취하도록 기타를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- AWS 계정에 대한 추적 생성
- CloudTrail 지원 서비스 및 통합
- 에 대한 Amazon SNS 알림 구성 CloudTrail
- 여러 지역에서 CloudTrail 로그 파일 수신 및 여러 계정으로부터 CloudTrail 로그 파일 수신

모든 태그 편집기 작업은 태그 편집기 API 참조에 의해 CloudTrail 기록되고 문서화됩니다. 콘솔에서의 태그 편집기 작업은 에 의해 CloudTrail 기록되며 다음과 같은 이벤트로 `tagging.amazonaws.com` 표시됩니다. `eventSource`

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. ID 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 IAM 사용자 보안 인증 정보로 했는지 여부.
- 역할 또는 페더레이션 사용자에 대한 임시 보안 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 CloudTrail`userIdentity`요소를 참조하십시오.

Tag Editor 로그 파일 항목 이해

트레일은 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 전송할 수 있는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함되어 있습니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보를 포함합니다. CloudTrail 로그 파일은 공개 API 호출의 정렬된 스택 추적이 아니므로 특정 순서로 표시되지 않습니다.

다음 예제는 작업을 `TagResources` 보여주는 CloudTrail 로그 항목을 보여줍니다.

```
{
```

```
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAEXAMPLEEXAMPLE:botocore-session-1661372702",
        "arn": "arn:aws:sts::123456789012:assumed-role/cli-role/botocore-
session-1661372702",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAEXAMPLEEXAMPLE",
                "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 서비스가 특정 규정 준수 프로그램의 범위에 포함되는지 알아보려면 규정 준수 프로그램 제공 범위 내 AWS 서비스를 참조하고 관심 있는 규정 준수 프로그램을 선택하십시오. 일반적인 정보는 AWS 규정 준수 프로그램 섹션을 참조하세요.

AWS Artifact을(를) 사용하여 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 AWS Artifact에서 보고서 다운로드 섹션을 참조하세요.

AWS 서비스 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 결정됩니다.AWS에서는 규정 준수를 지원할 다음과 같은 리소스를 제공합니다.

- 보안 및 규정 준수 빠른 시작 안내서 – 이 배포 안내서에서는 아키텍처 고려 사항에 대해 설명하고 보안 및 규정 준수에 중점을 둔 기본 AWS환경을 배포하기 위한 단계를 제공합니다.
- Amazon Web Services에서 HIPAA 보안 및 규정 준수 기술 백서 설계 - 이 백서는 기업에서 AWS를 사용하여 HIPAA를 준수하는 애플리케이션을 만드는 방법을 설명합니다.

  > **Note**
  >
  > 모든 AWS 서비스에 HIPAA 자격이 있는 것은 아닙니다. 자세한 내용은 HIPAA 적격 서비스 참조 섹션을 참조하세요.

- AWS 규정 준수 리소스 - 고객 조직이 속한 산업 및 위치에 적용될 수 있는 워크북 및 가이드 컬렉션입니다.
- AWS 고객 규정 준수 가이드 - 규정 준수의 관점에서 공동 책임 모델을 이해합니다. 이 가이드에서는 AWS 서비스를 보호하기 위한 모범 사례를 요약하고 여러 프레임워크(미국 표준 기술 연구소(NIST), 결제 카드 산업 보안 표준 위원회(PCI), 국제 표준화기구(ISO) 등)에서 보안 제어에 대한 지침을 매핑합니다.

- AWS Config 개발자 가이드의 규칙을 사용하여 리소스 평가 – AWS Config 서비스는 내부 사례, 산업 지침 및 규제에 대한 리소스 구성의 준수 상태를 평가합니다.
- AWS Security Hub – 이 AWS 서비스는 AWS내의 보안 상태에 대한 포괄적인 보기를 제공합니다. Security Hub는 보안 제어를 사용하여 AWS리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조 섹션을 참조하세요.
- AWS Audit Manager - 이 AWS 서비스는 AWS 사용을 지속적으로 감사하여 리스크를 관리하고 규정 및 업계 표준을 준수하는 방법을 간소화할 수 있도록 지원합니다.

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. AWS 리전에서는 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며 이러한 가용 영역은 짧은 대기 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크에 연결되어 있습니다. 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 센터로 문의하십시오.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라 섹션을 참조하세요.

Tag Editor의 인프라 보안

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

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

- 전송 계층 보안(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 요청은 액세스 키 ID와 AWS Identity and Access Management(IAM) 주체에 연결된 비밀 액세스 키를 사용하여 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. 그러면 이러한 방식은 AWS 네트워크의 특정 VPC에서만 특정 리소스에 대한 네트워크 액세스를 효과적으로 격리시킵니다.

Tag Editor 참조 정보

Tag Editor 참조 정보에는 해당 Service Quotas가 나와 있습니다.

Tag Editor의 Service Quotas

다음 표는 Tag Editor의 Service Quotas에 대한 정보입니다.

이러한 Quotas는 현재 Service Quotas 콘솔을 사용하여 조정할 수 없습니다. AWS Support에 문의하십시오.

명칭	기본값
리소스별로 첨부된 태그	사용자 정의 태그 50개 (AWS 생성된 태그는 이 한도에 포함되지 않음)
태그 키 이름	UTF-8 형식의 유니코드 문자 최소 1자, 최대 128자. 허용되는 문자는 문자, 숫자, 공백 및 다음의 문자입니다. `_ . : / = + - @` 키 이름은 로 시작할 수 없습니다. 해당 접두사는 사용하도록 AWS 예약되어 `aws:` 있기 때문입니다. **Note** AWS 서비스 일부에는 몇 가지 추가 문자 또는 길이 제한이 있습니다. 세부 정보는 해당 특정

명칭	기본값
	서비스에 대한 설명서를 참조하십시오.
태그 값	UTF-8 형식의 유니코드 문자 최소 0자, 최대 256자. 허용되는 문자는 문자, 숫자, 공백 및 다음의 문자입니다. `_ . : / = + - @` **Note** AWS 서비스 일부에는 몇 가지 추가 문자 또는 길이 제한이 있습니다. 세부 정보는 해당 특정 서비스에 대한 설명서를 참조하십시오.
GetResources API 작업 호출 비율	초당 최대 15회 호출
다음 API 작업의 호출 비율: • TagResources • UntagResources • GetTagKeys • GetTagValues	초당 최대 5회 호출

Tag Editor 문서 기록

변경 사항	설명	날짜
콘텐츠 태그 지정하기가 AWS 일반 참조에서 이 설명서로 이동함	AWS 리소스 태그 지정에 대한 주제가 AWS 일반 참조에서 이 설명서로 옮겨졌습니다.	2023년 3월 24일
IAM 모범 사례 업데이트	IAM 모범 사례에 따라 설명서가 업데이트되었습니다. 자세한 내용은 IAM의 보안 모범 사례 섹션을 참조하세요.	2023년 1월 3일
Tag Editor 설명서를 별도의 설명서로 옮기기	이제 Tag Editor 설명서가 AWS Resource Groups 사용 설명서의 일부가 아닌 별도의 사용 설명서로 제공됩니다.	2022년 12월 13일
태그 정책 준수 여부 확인	AWS Organizations를 사용하여 태그 정책을 만들고 계정에 첨부한 후에는 조직 계정의 리소스에서 규정을 준수하지 않는 태그를 찾을 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 콘솔에서 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	2019년 6월 5일

	Tag Editor 콘솔을 열 수 있습니다.	
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

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조서의 AWS 용어집을 참조하십시오.

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