AWS 백서

AWS 리소스 태그 지정 모범 사례

AWS 리소스 태그 지정 모범 사례: AWS 백서

Table of Contents

AWS 리소스 태그 지정 모범 사례

발행일: 2023년 3월 30일(문서 수정)

Amazon Web Services(AWS)를 사용하면 많은 AWS 리소스에 태그 형태로 메타데이터를 할당할 수 있습니다. 각 태그는 리소스에 대한 정보 또는 해당 리소스에 보관된 데이터를 저장하기 위한 키와 선택적 값으로 구성된 간단한 레이블입니다. 이 백서는 목적, 팀, 환경 또는 비즈니스와 관련된 기타 기준에 따라 리소스를 분류하는 데 도움이 되는 태그 지정 사용 사례, 전략, 기법 및 도구에 중점을 둡니다. 일관된 태그 지정 전략을 구현하면 리소스를 필터링 및 검색하고 비용 및 사용량을 모니터링하고 AWS 환경을 관리하는 것이 더 쉬워질 수 있습니다.

이 백서는 다중 계정을 사용하여 AWS 환경 구성 백서에 제공된 사례 및 지침을 기반으로 합니다. 이 문서를 읽기 전에 먼저 해당 백서를 읽어 보시기 바랍니다. AWS에서는 전체적인 방식으로 클라우드 기반을 구축할 것을 권장합니다. 자세한 내용은 AWS에서 클라우드 기반 구축을 참조하세요.

귀사는 Well-Architected입니까?

AWS Well-Architected 프레임워크는 클라우드에서 시스템을 구축할 때 내리는 결정의 장단점을 이해하는 데 도움이 됩니다. 이 프레임워크를 사용하여 클라우드에서 안정적이고 안전하며 효율적이고 비용 효율적인 시스템을 설계하고 운영하기 위한 아키텍처 모범 사례를 살펴볼 수 있습니다. AWS Management Console에서 무료로 제공되는 AWS Well-Architected Tool를 사용하면 각 요소에 대한 일련의 질문에 답하여, 이러한 모범 사례와 비교하여 워크로드를 검토할 수 있습니다.

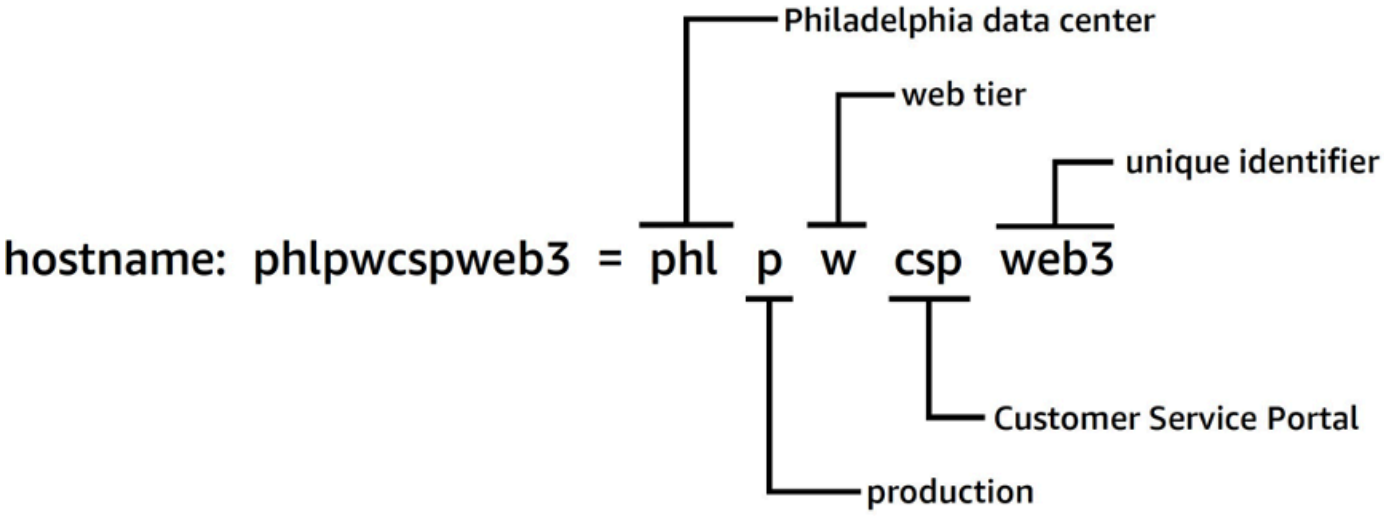
참조 아키텍처 배포, 다이어그램, 백서 등 클라우드 아키텍처에 대한 더 많은 전문가 지침과 모범 사례를 보려면 AWS 아키텍처 센터를 참조하세요.

소개

AWS를 사용하면 Amazon EC2 인스턴스, Amazon EBS 볼륨, 보안 그룹 및 AWS Lambda 함수와 같은 리소스를 생성하여 AWS에서 워크로드를 쉽게 배포할 수 있습니다. 또한 시간이 지남에 따라 애플리케이션을 호스팅하고 데이터를 저장하고 AWS 인프라를 확장하는 AWS 리소스의 규모를 조정하고 및 확장할 수 있습니다. AWS 사용량이 여러 애플리케이션에 걸쳐 다양한 리소스 유형으로 증가함에 따라 어떤 리소스가 어떤 애플리케이션에 할당되었는지 추적하는 메커니즘이 필요합니다. 이 메커니즘을 사용하여 비용 모니터링, 인시던트 관리, 패치, 백업 및 액세스 제어와 같은 운영 활동을 지원합니다.

온-프레미스 환경에서 이러한 지식은 지식 관리 시스템, 문서 관리 시스템 및 내부 wiki 페이지에 캡처되는 경우가 많습니다. 구성 관리 데이터베이스(CMDB)를 사용하면 표준 변경 관리 프로세스를 사용

하여 관련된 세부 메타데이터를 저장하고 관리할 수 있습니다. 이 접근 방식은 거버넌스를 제공하지만 개발 및 유지 관리를 위해서는 추가적인 노력이 필요합니다. 리소스 이름 지정에 구조화된 접근 방식을 사용할 수 있지만 리소스 이름에는 제한된 양의 정보만 포함할 수 있습니다.



리소스 이름 지정에 대한 구조화된 접근 방식

예를 들어 EC2 인스턴스에는 유사한 기능을 제공하는 Name이라는 사전 정의된 태그가 있으며 이를 통해 워크로드가 AWS로 이동할 때 해당 워크로드의 이름을 지정할 수 있습니다.

2010년에는 AWS에서 리소스에 메타데이터를 첨부하기 위한 유연하고 확장 가능한 메커니즘을 제공하는 리소스 태그를 출시했습니다. 이 백서는 AWS 환경 전반에 걸쳐 강력한 태깅 전략을 개발하고 구현하는 프로세스를 안내합니다. 이 지침은 의사 결정 및 운영 활동을 지원하는 태그 지정 일관성과 적용 범위를 보장하는 데 도움이 됩니다.

태그란 무엇입니까?

태그는 리소스에 대한 메타데이터를 보관하기 위해 리소스에 적용되는 키-값 쌍입니다. 각 태그는 키와 값(선택 사항)으로 구성됩니다. 현재 모든 서비스와 리소스 유형이 태그를 지원하는 것은 아닙니다(리소스 그룹 태그 지정 API를 지원하는 서비스 참조). 다른 서비스는 자체 API를 통해 태그를 지원할 수 있습니다. 태그는 암호화되지 않으므로 개인 식별 정보(PII)와 같은 민감한 데이터를 저장하는 데 사용해서는 안 됩니다.

사용자가 AWS CLI, API 또는 AWS Management Console을 사용하여 생성하여 AWS 리소스에 적용하는 태그를 사용자 정의 태그라고 합니다. AWS CloudFormation, Elastic Beanstalk 및 Auto Scaling과 같은 일부 AWS 서비스는 생성 및 관리하는 리소스에 자동으로 태그를 할당합니다. 이러한 키를 AWS 생성된 태그라고 하며 일반적으로 `aws` 접두사가 붙습니다. 사용자 정의 태그 키에는 이 접두사를 사용할 수 없습니다.

AWS 리소스에 추가할 수 있는 사용자 정의 태그 수에는 사용 요구 사항 및 제한이 있습니다. 자세한 내용은 AWS 일반 참조 가이드의 태그 이름 지정 제한 및 요구 사항을 참조하세요. AWS 생성된 태그는 이러한 사용자 정의 태그 제한에 포함되지 않습니다.

표 1 — 사용자 정의 태그 키 및 값의 예

인스턴스 ID	태그 키	태그 값
`i-01234567abcdef89a`	`CostCenter`	`98765`
	`Stack`	`Test`
i-12345678abcdef90b	`CostCenter`	`98765`
	`Stack`	`Production`

표 2 — AWS 생성된 태그의 예

AWS 생성된 태그 키	이론적 근거
`aws:ec2spot:fleet-request-id`	인스턴스를 시작한 Amazon EC2 스팟 인스턴스 요청을 식별합니다.

AWS 생성된 태그 키	이론적 근거
aws:cloudformation:stack-name	리소스를 생성한 AWS CloudFormation 스택을 식별합니다.
lambda-console:blueprint	AWS Lambda 함수의 템플릿으로 사용되는 청사진을 식별합니다.
elasticbeanstalk:environment-name	리소스를 생성한 애플리케이션을 식별합니다.
aws:servicecatalog: provisionedProductArn	프로비저닝된 제품 Amazon 리소스 이름(ARN)
aws:servicecatalog:productArn	프로비저닝된 제품이 시작된 제품의 ARN

AWS 생성된 태그는 네임스페이스를 형성합니다. 예를 들어 AWS CloudFormation 템플릿에서 함께 배포할 리소스 세트를 stack에 정의합니다. 여기서 stack-name은 이를 식별하기 위해 할당하는 설명이 포함된 이름입니다. aws:cloudformation:stack-name과 같은 키를 살펴보면 파라미터의 범위를 지정하는 데 사용되는 네임스페이스가 aws(조직), cloudformation(서비스), stack-name(파라미터) 등 세 가지 요소를 사용하는 것을 볼 수 있습니다.

사용자 정의 태그는 네임스페이스를 사용할 수도 있으므로 조직 식별자를 접두사로 사용하는 것이 좋습니다. 이를 통해 태그가 관리형 스키마의 태그인지 환경에서 사용 중인 서비스 또는 도구에서 정의한 태그인지 빠르게 식별할 수 있습니다.

AWS에서 클라우드 기반 구축 백서에서는 구현해야 하는 태그 세트를 권장합니다. 비즈니스마다 허용 패턴이 다르고 특정 태그에 대한 목록도 다를 가능성이 높습니다. 표 3의 예를 살펴보면 다음과 같습니다.

표 3 – 동일한 태그 키, 다른 값 검증 규칙

조직	태그 키	태그 값 검증	태그 값 예제
A 회사	CostCenter	5432, 5422, 5499	5432
B 회사	CostCenter	ABC*	ABC123

이 두 스키마가 서로 다른 조직에 있는 경우에는 태그 충돌 문제가 없습니다. 그러나 이 두 환경이 병합되면 네임스페이스가 충돌할 수 있고 유효성 검사가 더 복잡해질 수 있습니다. 이 시나리오는 거의 불가능해 보일 수 있지만 기업이 인수되거나 합병되고 다른 시나리오도 있습니다(예: 관리 서비스 제공업체, 게임 게시자 또는 벤처 캐피털 기업과 협력하는 클라이언트). 여기서 서로 다른 조직의 계정이 공유 AWS 조직에 속해 있는 경우도 있습니다. 표 4와 같이 비즈니스 이름을 접두사로 사용하여 고유한 네임스페이스를 정의하면 이러한 문제를 피할 수 있습니다.

표 4 – 태그 키의 네임스페이스 사용

조직	태그 키	태그 값 검증	태그 값 예제
A 회사	`company-a:CostCenter`	5432, 5422, 5499	5432
B 회사	`company-b:CostCenter`	ABC*	ABC123

정기적으로 기업을 인수하고 매각하는 크고 복잡한 조직에서는 이러한 상황이 더 자주 발생합니다. 신규 인수의 프로세스와 관행이 그룹 전반에 걸쳐 조화를 이루면 상황이 해결됩니다. 네임스페이스를 구분하면 이전 태그의 사용을 보고하고 관련 팀에 연락하여 새 스키마를 채택할 수 있기 때문에 도움이 됩니다. 네임스페이스를 사용하여 범위를 나타내거나 조직 소유자에 맞게 조정된 사용 사례 또는 책임 영역을 나타낼 수도 있습니다.

표 5 – 태그 키 내 범위 또는 사용 사례 범위의 예

사용 사례	태그 키	이론적 근거	허용된 값
데이터 분류	`example-inc:info-sec:data-classification`	정보 보안 정의 데이터 분류 세트	`sensitive`, `company-confidential`, `customer-identifiable`
작업	`example-inc:dev-ops:environment`	테스트 및 개발 환경 일정 구현	`development`, `staging`, `quality-assurance`, `production`

사용 사례	태그 키	이론적 근거	허용된 값
재해 복구	`example-inc:disaster-recovery:rpo`	리소스에 대한 Recovery Point Objective(RPO) 정의	6h, 24h
비용 할당	`example-inc:cost-allocation:business-unit`	재무팀은 각 팀의 사용량과 지출에 대한 비용 보고가 필요합니다.	`corporate`, `recruitment`, `support`, `engineering`

태그는 간단하고 유연합니다. 태그의 키와 값은 모두 가변 길이 문자열이며 폭넓은 문자 집합을 지원할 수 있습니다. 길이 및 문자 집합에 대한 자세한 내용은 AWS 일반 참조에서 AWS 리소스 태그 지정을 참조하세요. 태그는 대소문자를 구분합니다. 즉, `costCenter` 및 `costcenter`는 서로 다른 태그 키입니다. 국가마다 단어의 철자가 다를 수 있으며 이로 인해 키에 영향을 미칠 수 있습니다. 예를 들어 미국에서는 키를 `costcenter`로 정의할 수 있지만 영국에서는 `costcentre`를 더 선호할 수 있습니다. 이러한 키는 리소스 태그 지정 관점에서 보면 서로 다릅니다. 태그 지정 전략의 일환으로 철자, 대소문자, 문장 부호를 정의합니다. 리소스를 만들거나 관리하는 모든 사람을 위한 참고 자료로 이 정의를 사용합니다. 이 항목에 대해서는 다음에 이어지는 태그 지정 전략 구축 단원에서 자세히 설명합니다.

태그 지정 전략 구축

많은 운영 관행과 마찬가지로 태그 지정 전략을 구현하는 것은 반복과 개선의 과정입니다. 우선 순위에 따라 작게 시작해서 필요에 따라 태그 지정 스키마를 확장합니다.

태그 지정 전략 반복 및 개선 주기

이 프로세스 전반에 걸쳐 소유권은 책임과 진행의 핵심입니다. 태그는 다양한 용도로 사용될 수 있으므로 전체 태깅 전략을 조직 내 책임 영역으로 나눌 수 있습니다. 태그 지정을 사용하면 리소스의 특성에 따라 달라지는 활동에 프로그래밍 방식으로 접근할 수 있습니다. 태그 지정으로 혜택을 받을 수 있는 이해 관계자의 범위는 조직의 규모와 운영 관행에 따라 달라집니다. 대규모 조직은 태그 지정 전략을 구축하고 구현하는 데 관련된 팀의 책임을 명확하게 정의하면 도움이 될 수 있습니다. 일부 이해 관계자는 태그 지정의 요구 사항 파악(사용 사례 정의)을 담당하고, 다른 이해 관계자는 태그 지정 전략의 유지, 구현 및 개선을 담당할 수 있습니다.

소유권을 할당하면 전략의 개별 측면을 구현할 수 있는 좋은 위치에 있게 됩니다. 적절한 경우 이러한 소유권을 정책으로 공식화하고 책임 매트릭스(예: RACI: Responsible, Accountable, Consulted 및

Informed) 또는 공동 책임 모델에 문서화할 수 있습니다. 소규모 조직에서는 요구 사항 정의부터 구현 및 시행에 이르기까지 팀이 태그 지정 전략에서 다양한 역할을 수행할 수 있습니다.

요구 사항 및 사용 사례 정의

기본적으로 메타데이터를 사용해야 하는 필요성이 있는 이해 관계자와 협력하여 전략 구축을 시작합니다. 이러한 팀은 보고, 자동화 및 데이터 분류와 같이 활동을 지원하기 위해 리소스에 태그를 지정해야 하는 메타데이터를 정의합니다. 리소스를 구성하는 방법과 리소스를 매핑해야 하는 정책을 간략하게 설명합니다. 이러한 이해 관계자가 조직에서 가질 수 있는 역할과 기능의 예는 다음과 같습니다.

- 재무와 LOB(Line of Business)는 투자 가치를 비용과 연계하여 이해함으로써 불평등을 해결할 때 취해야 할 조치의 우선 순위를 정해야 합니다. 창출된 비용 대비 가치를 이해하면 실패한 LOB(Line of Business) 또는 제품 서비스를 식별하는 데 도움이 됩니다. 이는 지속적인 지원, 대안 채택(예: SaaS 서비스 또는 관리 서비스 사용) 또는 수익성이 낮은 비즈니스 서비스의 폐지에 대한 정보에 입각한 결정으로 이어집니다.
- 거버넌스 및 규정 준수는 권한, 정책 및 모니터링과 같은 적절한 제어 및 감독을 적용하기 위해 데이터의 분류(예: 공개, 민감 또는 기밀), 특정 워크로드가 특정 표준 또는 규정에 대한 감사 범위에 속하는지 여부, 서비스의 중요도(서비스 또는 애플리케이션이 비즈니스에 중요한지 여부)를 이해해야 합니다.
- 운영 및 개발 부서는 워크로드 수명 주기, 지원 제품의 구현 단계, 릴리스 단계 관리(예: 개발, 테스트, 프로덕션 분할), 관련 지원 우선 순위 및 이해 관계자 관리 요구 사항을 이해해야 합니다. 백업, 패치, 관찰성, 사용 중단과 같은 업무도 정의하고 이해해야 합니다.
- 정보 보안(InfoSec) 및 보안 운영(SecOps)은 적용해야 하는 제어와 권장되는 제어를 설명합니다. 일반적으로 InfoSec은 제어의 구현을 정의하며 SecOps는 일반적으로 이러한 제어의 관리를 담당합니다.

사용 사례, 우선 순위, 조직 규모, 운영 관행에 따라 재무(조달 포함), 정보 보안, 클라우드 지원, 클라우드 운영 등 조직 내 다양한 팀을 대표해야 할 수도 있습니다. 또한 패치, 백업 및 복원, 모니터링, 작업 예약, 재해 복구와 같은 기능에 대한 애플리케이션 및 프로세스 소유자의 대리인도 필요합니다. 이들 담당자는 태그 지정 전략의 정의, 구현 및 효율성 측정을 주도하는 데 도움을 줍니다. 담당자는 이해관계자와 사용 사례를 거꾸로 검토하여 부서 간 워크숍을 진행해야 합니다. 워크숍에서는 각자의 관점과 요구 사항을 공유하고 전반적인 전략을 수립하는 데 도움을 줄 기회를 얻습니다. 참가자의 예와 다양한 사용 사례에 대한 참여는 이 백서 뒷부분에 설명되어 있습니다.

또한 이해 관계자는 필수 태그의 키를 정의 및 검증하고 선택적 태그의 범위를 추천할 수 있습니다. 예를 들어 재무팀은 리소스를 내부 비용 센터, 사업부 또는 둘 다에 연결해야 할 수 있습니다. 따라서 특

정 태그 키(예: `CostCenter` 및 `BusinessUnit`)를 필수로 설정해야 할 수도 있습니다. 개별 개발 팀은 자동화를 위해 `EnvironmentName`, `OptIn` 또는 `OptOut`와 같은 추가 태그를 사용하기로 결정할 수 있습니다.

주요 이해 관계자는 태그 지정 전략 접근 방식에 동의하고 다음과 같은 규정 준수 및 거버넌스 관련 질문에 대한 답변을 문서화해야 합니다.

- 해결해야 할 사용 사례는 무엇입니까?
- 리소스 태그 지정(구현)은 누가 담당하나요?
- 태그는 어떻게 적용되며 어떤 방법과 자동화가 사용됩니까(사전 또는 사후)?
- 태그 지정의 효율성과 목표는 어떻게 측정되나요?
- 태그 지정 전략을 얼마나 자주 검토해야 하나요?
- 개선을 주도하는 사람은 누구인가요? 어떻게 이루어지나요?

그러면 Cloud Enablement, Cloud Business Office, Cloud Platform Engineering과 같은 비즈니스 부서가 진행 상황을 측정하고, 장애물을 제거하고, 중복되는 노력을 줄임으로써 태그 지정 전략을 구축하는 프로세스를 촉진하고, 채택을 촉진하고, 애플리케이션의 일관성을 보장하는 역할을 할 수 있습니다.

태그 지정 스키마 정의 및 게시

AWS 리소스에 필수 태그와 선택적 태그 모두에 대해 일관된 접근 방식을 사용하여 태그를 지정합니다. 포괄적인 태그 지정 스키마를 사용하면 이러한 일관성을 유지할 수 있습니다. 다음 예제는 시작하는 데 도움이 될 수 있습니다.

- 필수 태그 키에 대한 동의
- 허용 가능한 값 및 태그 이름 지정 규칙(대/소문자, 대시 또는 밑줄, 계층 구조 등) 정의
- 확인 값은 개인 식별 정보(PII)를 구성하지 않음
- 누가 새 태그 키를 정의하고 생성할 수 있는지 결정
- 새 필수 태그 값을 추가하는 방법과 선택적 태그를 관리하는 방법에 대한 동의

다음 태그 지정 범주 표를 검토합니다. 이 표는 태그 지정 스키마에 포함할 수 있는 항목의 기준으로 사용할 수 있습니다. 여전히 태그 키에 사용할 규칙과 각각에 허용되는 값을 결정해야 합니다. 태그 지정 스키마는 환경에 맞게 이를 정의하는 문서입니다.

표 6 — 최종 태그 지정 스키마의 예(1부)

사용 사례	태그 키	이론적 근거	허용된 값(나열된 또는 값 접두사/접미사)	비용 할당에 사용됨	리소스 유형	범위	필수
비용 할당	example-inc:cost-allocation: ApplicationId	각 사업부에서 창출된 비용 대비 가치 추적	DataLakeX , RetailSiteX	Y	모두	모두	필수
비용 할당	example-inc:cost-allocation: BusinessUnitId	사업부별 비용 모니터링	Architecture , DevOps, Finance	Y	모두	모두	필수
비용 할당	example-inc:cost-allocation:CostCenter	비용 센터별 비용 모니터링	123-*	Y	모두	모두	필수
비용 할당	example-inc:cost-allocation:Owner	이 워크로드를 책임지는 예산 보유자는 누구입니까?	Marketing , RetailSupport	Y	모두	모두	필수
액세스 제어	example-inc:access-control:LayerId	역할에 따라 리소스에 대한 액세스 권한을 부여할 하위 구성 요소/계층 식별	DB_Layer, Web_Layer , App_Layer	N	모두	모두	선택

표 6 — 최종 태그 지정 스키마의 예(2부)

사용 사례	태그 키	이론적 근거	허용된 값(나열된 또는 값 접두사/접미사)	비용 할당에 사용됨	리소스 유형	범위	필수
DevOps	`example-inc:operations: Owner`	리소스 생성 및 유지 관리를 담당하는 팀/스쿼드는 어디입니까?	`Squad01`	N	모두	모두	필수
재해 복구	`example-inc:disaster-recovery:rpo`	리소스에 대한 Recovery Point Objective(RPO) 정의	6h, 24h	N	S3, EBS	Prod	필수
데이터 분류	`example-inc:data: classification`	규정 준수 및 거버넌스를 위한 데이터 분류	`Public, Private, Confidential , Restricted`	N	S3, EBS	모두	필수
규정 준수	`example-inc:compliance: framework`	워크로드에 적용되는 규정 준수 프레임워크 식별	`PCI-DSS, HIPAA`	N	모두	Prod	필수

태그 지정 스키마를 정의한 후에는 모든 관련 이해 관계자가 쉽게 참조하고 추적 가능한 업데이트를 위해 액세스할 수 있는 버전 제어 리포지토리에서 스키마를 관리합니다. 이 접근 방식을 사용하면 효율성이 향상되고 민첩성을 허용합니다.

AWS Organizations – 태그 정책

AWS Organizations의 정책을 사용하면 조직의 AWS 계정에 추가 유형의 거버넌스를 적용할 수 있습니다. 태그 정책은 플랫폼이 AWS 환경 내에서 스키마를 보고하고 선택적으로 적용할 수 있도록 태그 지정 스키마를 JSON 형식으로 표현하는 방법입니다. 태그 정책은 특정 리소스 유형의 태그 키에 허용되는 값을 정의합니다. 이 정책은 값 목록 형식이거나 접두사 뒤에 와일드카드 문자(*)를 붙이는 형식일 수 있습니다. 단순 접두사 접근 방식은 개별 값 목록보다 덜 엄격하지만 유지 관리가 덜 필요합니다.

다음 예제는 태그 지정 정책을 정의하여 지정된 키에 적합한 값을 검증하는 방법을 보여 줍니다. 사람에게 친숙한 표 형식의 스키마 정의를 사용하여 이 정보를 하나 이상의 태그 정책에 기록할 수 있습니다. 소유권 위임을 지원하기 위해 별도의 정책을 사용하거나 특정 시나리오에서만 일부 정책을 적용할 수 있습니다.

ExampleInc-CostAllocation.json

다음은 비용 할당 태그를 보고하는 태그 정책의 예입니다.

```
{
  "tags": {
    "example-inc:cost-allocation:ApplicationId": {
      "tag_key": {
        "@@assign": "example-inc:cost-allocation:ApplicationId"
      },
      "tag_value": {
        "@@assign": [
          "DataLakeX",
          "RetailSiteX"
        ]
      }
    },
    "example-inc:cost-allocation:BusinessUnitId": {
      "tag_key": {
        "@@assign": "example-inc:cost-allocation:BusinessUnitId"
      },
      "tag_value": {
        "@@assign": [
          "Architecture",
```

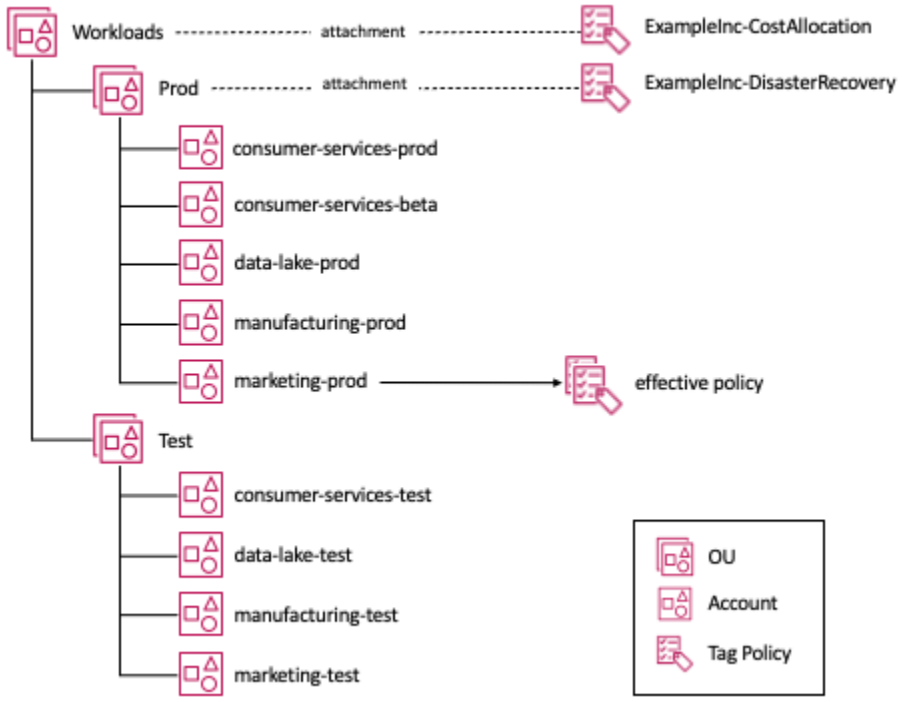
```
          "DevOps",
          "FinanceDataLakeX"
        ]
      }
    },
    "example-inc:cost-allocation:CostCenter": {
      "tag_key": {
        "@@assign": "example-inc:cost-allocation:CostCenter"
      },
      "tag_value": {
        "@@assign": [
          "123-*"
        ]
      }
    }
  }
}
```

ExampleInc-DisasterRecovery.json

다음은 재해 복구 태그를 보고하는 태그 정책의 예입니다.

```
{
    "tags": {
        "example-inc:disaster-recovery:rpo": {
            "tag_key": {
                "@@assign": "example-inc:disaster-recovery:rpo"
            },
            "tag_value": {
                "@@assign": [
                    "6h",
                    "24h"
                ]
            }
        }
    }
}
```

이 예에서는 `ExampleInc-CostAllocation` 태그 정책이 `Workloads` OU에 연결되므로 `Prod` 및 `Test` 하위 OU의 모든 계정에 적용됩니다. 마찬가지로 `ExampleInc-DisasterRecovery` 태그 정책은 `Prod` OU에 연결되므로 이 OU 아래의 계정에만 적용됩니다. 다중 계정을 사용한 환경 구성 백서에서는 권장 OU 구조를 보다 자세히 살펴봅니다.



OU 구조에 태그 정책 첨부

다이어그램에서 `marketing-prod` 계정을 보면 두 태그 정책이 모두 이 계정에 적용되므로 효과적인 정책이라는 개념을 갖게 됩니다. 즉, 계정에 적용되는 특정 유형의 정책 컨볼루션입니다. 주로 리소스를 수동으로 관리하는 경우 콘솔의 Resource Groups & Tag Editor:Tag 정책을 방문하여 효과적인 정책을 검토할 수 있습니다. 코드형 인프라(IaC) 또는 스크립팅을 사용하여 리소스를 관리하는 경우 `AWS::Organizations::DescribeEffectivePolicy` API 호출을 사용할 수 있습니다.

태그 지정 구현 및 적용

이 섹션에서는 수동, 코드형 인프라(IaC), 지속적 통합/지속적 전달(CI/CD)과 같은 리소스 관리 전략에 사용할 수 있는 도구를 소개합니다. 이러한 접근 방식의 주요 차원은 배포 빈도가 점점 더 높아진다는 것입니다.

수동으로 관리되는 리소스

이는 일반적으로 채택의 기초 또는 마이그레이션 단계에 속하는 워크로드입니다. 일반적으로 이러한 워크로드는 기존의 작성된 절차를 사용하여 구축된 단순하고 대부분 정적인 워크로드이거나 온프레미스 환경에서 CloudEndure와 같은 도구를 사용하여 그대로 마이그레이션된 워크로드입니다. Migration Hub 및 CloudEndure와 같은 마이그레이션 도구는 마이그레이션 프로세스의 일부로 태그를 적용할 수

있습니다. 그러나 원래 마이그레이션 중에 태그가 적용되지 않았거나 그 이후에 태그 지정 스키마가 변경된 경우 Tag Editor(AWS Management Console의 기능)를 사용하여 다양한 검색 기준을 사용하여 리소스를 검색하고 태그를 대량으로 추가, 수정 또는 삭제할 수 있습니다. 검색 기준에는 특정 태그나 값이 있거나 없는 리소스가 포함될 수 있습니다. AWS 리소스 태그 지정 API를 사용하면 이러한 기능을 프로그래밍 방식으로 수행할 수 있습니다.

이러한 워크로드가 현대화됨에 따라 Auto Scaling 그룹과 같은 리소스 유형이 도입되었습니다. 이러한 리소스 유형을 사용하면 탄력성이 향상되고 복원력이 향상됩니다. Auto Scaling 그룹이 사용자를 대신하여 Amazon EC2 인스턴스를 관리하지만 수동으로 생성한 리소스로 일관되게 EC2 인스턴스에 태그를 지정해야 할 수도 있습니다. Amazon EC2 시작 템플릿은 Auto Scaling에서 생성하는 인스턴스에 적용할 태그를 지정하는 방법을 제공합니다.

워크로드의 리소스를 수동으로 관리하는 경우 리소스 태그 지정을 자동화하는 것이 유용할 수 있습니다. 다양한 솔루션을 사용할 수 있습니다. 한 가지 접근 방식은 AWS Config 규칙를 사용하는 것인데, 이를 통해 `required_tags`를 확인한 다음 이를 적용하기 위한 Lambda 함수를 시작할 수 있습니다. AWS Config 규칙에 대해서는 이 백서 뒷부분에서 자세히 설명합니다.

코드형 인프라(IaC) 관리 리소스

AWS CloudFormation에서 사용자 AWS 환경의 모든 인프라 리소스를 프로비저닝하기 위한 공통 언어를 제공합니다. CloudFormation 템플릿은 자동화된 방식으로 AWS 리소스를 생성하는 JSON 또는 YAML 파일입니다. CloudFormation 템플릿을 사용하여 AWS 리소스를 생성하는 경우 CloudFormation Resource Tags 속성을 사용하여 생성 시 지원되는 리소스 유형에 태그를 적용할 수 있습니다. IaC로 태그와 리소스를 관리하면 일관성을 유지하는 데 도움이 됩니다.

AWS CloudFormation에서 리소스를 생성하면 서비스가 AWS CloudFormation 템플릿으로 생성된 리소스에 AWS 정의된 태그 세트를 자동으로 적용합니다. 예를 들면 다음과 같습니다.

```
aws:cloudformation:stack-name
aws:cloudformation:stack-id
aws:cloudformation:logical-id
```

AWS CloudFormation 스택을 기반으로 리소스 그룹을 쉽게 정의할 수 있습니다. 이러한 AWS 정의된 태그는 스택에서 생성된 리소스에 상속됩니다. 하지만 Auto Scaling 그룹 내 Amazon EC2 인스턴스의 경우 AWS CloudFormation 템플릿의 Auto Scaling 그룹 정의에서 `AWS::AutoScaling::AutoScalingGroup TagProperty`를 설정해야 합니다. 또는 Auto Scaling 그룹과 함께 EC2 시작 템플릿을 사용하는 경우 해당 정의에 태그를 정의할 수 있습니다. Auto Scaling 그룹 또는 AWS 컨테이너 서비스와 함께 EC2 시작 템플릿을 사용하는 것이 좋습니다. 이러한 서비스는

Amazon EC2 인스턴스의 일관된 태그 지정을 보장하고 여러 인스턴스 유형 및 구매 옵션에 걸친 Auto Scaling을 지원하여 복원력을 개선하고 컴퓨팅 비용을 최적화하는 데 도움이 됩니다.

AWS CloudFormation 후크는 개발자에게 애플리케이션의 주요 측면을 조직의 표준과 일관되게 유지할 수 있는 수단을 제공합니다. 경고를 제공하거나 배포를 방지하도록 후크를 구성할 수 있습니다. 이 기능은 Auto Scaling 그룹이 시작할 모든 Amazon EC2 인스턴스에 고객 정의 태그를 적용하도록 구성되었는지 또는 모든 Amazon S3 버킷이 필수 암호화 설정으로 생성되었는지 등 템플릿의 주요 구성 요소를 확인하는 데 가장 적합합니다. 두 경우 모두 이 규정 준수에 대한 평가는 배포 전 AWS CloudFormation 후크를 사용하여 배포 프로세스의 초기 단계로 진행됩니다.

AWS CloudFormation은(는) 템플릿에서 프로비전된 리소스(드리프트 감지를 지원하는 리소스 참조)가 수정되어 리소스가 더 이상 예상 템플릿 구성과 일치하지 않는 경우를 감지하는 기능을 제공합니다. 이를 드리프트라고 합니다. 자동화를 사용하여 IaC를 통해 관리되는 리소스에 태그를 적용하면 태그가 수정되어 드리프트가 발생합니다. IaC를 사용할 때는 현재 모든 태그 지정 요구 사항을 IaC 템플릿의 일부로 관리하고 AWS CloudFormation 후크를 구현하고 자동화에 사용할 수 있는 AWS CloudFormation Guard 규칙 세트를 게시하는 것이 좋습니다.

CI/CD 파이프라인 관리 리소스

워크로드의 성숙도가 높아짐에 따라 지속적 통합 및 연속 배포(CI/CD)와 같은 기술이 채택될 가능성이 높습니다. 이러한 기술을 사용하면 테스트 자동화가 향상되어 작은 변경 사항을 더 자주 배포하기 쉬워져 배포 위험을 줄일 수 있습니다. 배포로 인해 발생하는 예상치 못한 동작을 감지하는 관찰성 전략을 사용하면 사용자에게 미치는 영향을 최소화하면서 배포를 자동으로 롤백할 수 있습니다. 하루에 수십 번 배포하는 단계에 이르면 태그를 소급하여 적용하는 것은 더 이상 실용적이지 않습니다. 모든 것을 코드나 구성으로 표현하고 버전을 제어하고 가능한 경우 프로덕션 환경에 배포하기 전에 테스트 및 평가를 거쳐야 합니다. 통합 개발 및 운영(DevOps) 모델에서는 많은 사례가 운영 고려 사항을 코드로 다루고 배포 수명 주기 초기에 이를 검증합니다.

가장 좋은 방법은 이러한 검사를 프로세스 초기에 푸시하여(AWS CloudFormation 후크로 표시된 것처럼) AWS CloudFormation 템플릿이 개발자 컴퓨터를 벗어나기 전에 정책을 준수하는지 확인할 수 있도록 하는 것입니다.

AWS CloudFormation Guard 2.0은 CloudFormation으로 정의할 수 있는 모든 것에 대한 예방적 규정 준수 규칙을 작성할 수 있는 수단을 제공합니다. 템플릿은 개발 환경의 규칙에 따라 검증되었습니다. 분명히 이 기능은 다양한 용도로 사용되지만 이 백서에서는 `AWS::AutoScaling::AutoScalingGroup TagProperty`를 항상 사용할 수 있는 몇 가지 예제를 살펴보겠습니다.

다음은 CloudFormation Guard 규칙에 예입니다.

```
let all_asgs = Resources.*[ Type == 'AWS::AutoScaling::AutoScalingGroup' ]

rule tags_asg_automation_EnvironmentId when %all_asgs !empty {
    let required_tags = %all_asgs.Properties.Tags.*[
        Key == 'example-inc:automation:EnvironmentId' ]
    %required_tags[*] {
        PropagateAtLaunch == 'true'
        Value IN ['Prod', 'Dev', 'Test', 'Sandbox']
        <<Tag must have a permitted value
          Tag must have PropagateAtLaunch set to 'true'>>
    }
}

rule tags_asg_costAllocation_CostCenter when %all_asgs !empty {
    let required_tags = %all_asgs.Properties.Tags.*[
        Key == 'example-inc:cost-allocation:CostCenter' ]
    %required_tags[*] {
        PropagateAtLaunch == 'true'
        Value == /^123-/
        <<Tag must have a permitted value
          Tag must have PropagateAtLaunch set to 'true'>>
    }
}
```

코드 예제에서는 AutoScalingGroup 유형의 모든 리소스에 대해 템플릿을 필터링한 다음 두 가지 규칙을 적용합니다.

- **tags_asg_automation_EnvironmentId** - 이 키를 가진 태그가 존재하고 허용된 값 목록 내에 값이 있고 PropagateAtLaunch가 true로 설정되어 있는지 확인합니다.
- **tags_asg_costAllocation_CostCenter** - 이 키를 가진 태그가 존재하고 정의된 접두사 값으로 시작하는 값이 있고 PropagateAtLaunch가 true로 설정되어 있는지 확인합니다.

적용

앞서 설명한 것처럼 Resource Groups & Tag Editor는 조직의 OU에 적용되는 태그 정책에 정의된 태그 지정 요구 사항을 리소스가 충족하지 못하는 부분을 식별할 수 있는 수단을 제공합니다. 조직 구성원 계정 내에서 Resource Groups & Tag Editor 콘솔 도구에 액세스하면 해당 계정과 태그 정책의 요구 사항을 충족하지 못하는 계정 내 리소스에 적용되는 정책을 확인할 수 있습니다. 관리 계정에서 액세스하는 경우(그리고 AWS Organizations에서 제공하는 서비스에서 태그 정책에 액세스가 활성화된 경우) 조직의 모든 연결 계정에 대한 태그 정책 준수 여부를 확인할 수 있습니다.

태그 정책 자체 내에서 특정 리소스 유형에 대한 적용을 활성화할 수 있습니다. 다음 정책 예시에서는 `ec2:instance` 유형의 모든 리소스 및 `ec2:volume`이 정책을 준수하도록 요구하는 강제 적용을 추가했습니다. 태그 정책으로 리소스를 평가하려면 리소스에 태그가 있어야 하는 등의 몇 가지 알려진 제한 사항이 있습니다. 목록은 태그 정책 시행을 지원하는 리소스를 참조하세요.

ExampleInc-Cost-Allocation.json

다음은 비용 할당 태그를 보고 및/또는 적용하는 태그 정책의 예입니다.

```
{
  "tags": {
    "example-inc:cost-allocation:ApplicationId": {
      "tag_key": {
        "@@assign": "example-inc:cost-allocation:ApplicationId"
      },
      "tag_value": {
        "@@assign": [
          "DataLakeX",
          "RetailSiteX"
        ]
      },
      "enforced_for": {
        "@@assign": [
          "ec2:instance",
          "ec2:volume"
        ]
      }
    },
    "example-inc:cost-allocation:BusinessUnitId": {
      "tag_key": {
        "@@assign": "example-inc:cost-allocation:BusinessUnitId"
      },
      "tag_value": {
        "@@assign": [
          "Architecture",
          "DevOps",
          "FinanceDataLakeX"
        ]
      },
      "enforced_for": {
        "@@assign": [
          "ec2:instance",
          "ec2:volume"
```

```
        ]
      }
    },
    "example-inc:cost-allocation:CostCenter": {
      "tag_key": {
        "@@assign": "example-inc:cost-allocation:CostCenter"
      },
      "tag_value": {
        "@@assign": [
          "123-*"
        ]
      },
      "enforced_for": {
        "@@assign": [
          "ec2:instance",
          "ec2:volume"
        ]
      }
    }
  }
}
```

AWS Config (**`required_tag`**)

AWS Config은(는) AWS 리소스의 구성을 평가, 감사 및 측정할 수 있는 서비스입니다(AWS Config에서 지원되는 리소스 유형 참조). 태그 지정의 경우 `required_tags` 규칙을 사용하여 특정 키의 태그가 없는 리소스를 식별하는 데 사용할 수 있습니다(required_tags에서 지원하는 리소스 유형 참조). 이전 예제에서 모든 Amazon EC2 인스턴스에 키가 있는지 테스트할 수 있습니다. 키가 존재하지 않는 경우 인스턴스는 규정 미준수로 등록됩니다. 이 AWS CloudFormation 템플릿은 Amazon S3 버킷, Amazon EC2 인스턴스 및 Amazon EBS 볼륨에서 표에 설명된 필수 키가 있는지 테스트하는 AWS Config 규칙을 설명합니다.

```
Resources:
 MandatoryTags:
   Type: AWS::Config::ConfigRule
   Properties:
     ConfigRuleName: ExampleIncMandatoryTags
     Description: These tags should be in place
     InputParameters:
       tag1Key: example-inc:cost-allocation:ApplicationId
       tag2Key: example-inc:cost-allocation:BusinessUnitId
       tag3Key: example-inc:cost-allocation:CostCenter
```

```
        tag4Key: example-inc:automation:EnvironmentId
      Scope:
        ComplianceResourceTypes:
        - "AWS::S3::Bucket"
        - "AWS::EC2::Instance"
        - "AWS::EC2::Volume"
      Source:
        Owner: AWS
        SourceIdentifier: REQUIRED_TAGS
```

리소스가 수동으로 관리되는 환경에서는 AWS Lambda 함수를 통한 자동 수정 기능을 사용하여 리소스에 누락된 태그 키를 자동으로 추가하도록 AWS Config 규칙을 개선할 수 있습니다. 정적 워크로드에는 잘 작동하지만 IaC 및 배포 파이프라인을 통해 리소스를 관리하기 시작하면 효율성이 점차 떨어집니다.

AWS Organizations - 서비스 제어 정책(SCP)은 조직의 권한을 관리하는 데 사용할 수 있는 조직 정책 유형입니다. SCP는 조직 또는 조직 단위(OU)의 모든 계정에 사용 가능한 최대 권한을 중앙에서 제어합니다. SCP는 조직에 속한 계정이 관리하는 사용자 및 역할에만 영향을 줍니다. 리소스에 직접적인 영향을 미치지는 않지만 태그 지정 작업에 대한 권한을 포함하여 사용자 및 역할의 권한을 제한합니다. 태그 지정과 관련하여 SCP는 태그 정책이 제공할 수 있는 내용 외에도 태그 적용을 위한 추가 세분성을 제공할 수 있습니다.

다음 예제에서는 정책이 `example-inc:cost-allocation:CostCenter` 태그가 없는 `ec2:RunInstances` 요청은 거부합니다.

다음은 거부 SCP입니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyRunInstanceWithNoCostCenterTag",
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:*:*:instance/"
      ],
      "Condition": {
        "Null": {
          "aws:RequestTag/example-inc:cost-allocation:CostCenter": "true"
        }
      }
```

```
    }
  ]
}
```

연결 계정에 적용되는 효과적인 서비스 제어 정책은 설계상 검색할 수 없습니다. SCP로 태그를 적용하는 경우 개발자가 리소스가 계정에 적용된 정책을 준수하는지 확인할 수 있도록 개발자가 문서를 사용할 수 있어야 합니다. 계정 내에서 CloudTrail 이벤트에 대한 읽기 전용 액세스 권한을 제공하면 개발자가 리소스가 규정을 준수하지 못할 때 디버깅 작업을 수행할 수 있습니다.

태그 지정 효과 측정 및 개선 추진

태그 지정 전략을 구현한 후에는 대상 사용 사례와 비교하여 그 효과를 측정하는 것이 중요합니다. 효과 측정은 사용 사례에 따라 달라집니다. 예:

- 비용 속성 - AWS Cost Explorer또는 AWS Cost & Usage Report와 같은 도구를 사용하여 지출을 기준으로 리소스의 태그 지정 범위를 측정할 수 있습니다. 예를 들어 특히 특정 태그 키를 모니터링하면 태그가 지정되거나 태그가 지정되지 않은 리소스에서 요금이 발생하는 비율을 추적할 수 있습니다.
- 자동화 - 원하는 결과가 달성되었는지 감사해야 할 수도 있습니다. 예를 들어 비프로덕션 Amazon EC2 인스턴스가 업무 시간 외에 일시 중단되는지 여부, 인스턴스 시작 및 중지 시간을 감사할 수 있습니다.

관리 계정 내의 Resource Groups & Tag Editor는 조직의 모든 연결 계정에 대한 태그 정책 규정 준수를 분석할 수 있는 추가 기능을 제공합니다.

태그 지정 효율성 측정 결과를 기반으로 사용 사례 정의, 태그 지정 스키마 구현 또는 적용과 같은 단계에서 개선이나 변경이 필요한지 식별합니다. 필요에 따라 변경하고 원하는 효과를 얻을 때까지 주기를 반복합니다. 비용 기여도가 포함된 예제에서는 개선률을 확인할 수 있습니다.

리소스의 실제 태그 지정을 수행해야 하는 것은 개발자와 운영자이므로 소유권을 갖도록 하는 것이 중요합니다. AWS 채택 과정에서 팀이 일반적으로 부담하는 추가 책임은 이뿐만이 아닙니다. 보안 및 애플리케이션 개발과 운영에 드는 비용에 대한 책임을 높이는 것도 중요합니다. 조직은 종종 새로운 관행을 채택하도록 동기를 부여하는 수단으로 목표와 대상을 사용하므로 여기에도 적용할 수 있습니다.

태그 지정 사용 사례

주제

비용 할당 및 재무 관리를 위한 태그

조직에서 가장 먼저 다루는 태그 지정 사용 사례 중 하나는 비용 및 사용량에 대한 가시성과 관리입니다. 일반적으로 여기에는 몇 가지 이유가 있습니다.

- 이는 일반적으로 잘 이해되는 시나리오이며 요구 사항도 잘 알려져 있습니다. 예를 들어 재무 팀은 여러 서비스, 기능, 계정 또는 팀에 걸친 워크로드 및 인프라의 총 비용을 확인하려고 합니다. 이러한 비용 가시성을 확보하는 한 가지 방법은 리소스에 일관된 태그를 지정하는 것입니다.
- 태그와 그 값은 명확하게 정의되어 있습니다. 일반적으로 비용 할당 메커니즘은 조직의 재무 시스템에 이미 존재합니다(예: 비용 센터, 사업부, 팀 또는 조직 기능별 추적).
- 빠르고 입증 가능한 투자 수익 리소스에 일관되게 태그가 지정되면 시간 경과에 따른 비용 최적화 추세를 추적할 수 있습니다(예: 적정 규모, 자동 크기 조정 또는 일정에 따른 리소스).

AWS에서 비용이 어떻게 발생하는지 이해하면 정보에 입각한 재무 결정을 내릴 수 있습니다. 리소스, 워크로드, 팀 또는 조직 수준에서 비용이 어디서 발생했는지 알면 달성한 비즈니스 성과와 비교할 때 해당 수준에서 제공되는 가치를 더 잘 이해할 수 있습니다.

엔지니어링 팀은 리소스의 재무 관리 경험이 없을 수도 있습니다. 엔지니어링 및 개발 팀에 AWS 재무 관리의 기본 사항을 교육하고 재무와 엔지니어링 간의 관계를 구축하여 FinOps 문화를 조성할 수 있는 AWS 재무 관리 전문 기술을 갖춘 인력을 배치하면 비즈니스에서 측정 가능한 결과를 달성하고 팀이 비용을 염두에 두고 구축하도록 장려할 수 있습니다. Well-Architected Framework의 비용 최적화 요소는 훌륭한 금융 관행을 수립하는 것에 대해 자세히 다루지만 이 백서에서는 몇 가지 기본 원칙을 다룰 것입니다.

비용 할당 태그

비용 할당이란 정의된 프로세스에 따라 발생한 비용을 해당 비용의 사용자 또는 수익자에게 할당하거나 분배하는 것을 말합니다. 이 백서에서는 비용 할당을 쇼백과 차지백이라는 두 가지 유형으로 구분합니다.

쇼백 도구 및 메커니즘은 비용에 대한 인식을 높이는 데 도움이 됩니다. 차지백은 비용 회수에 도움이 되며 비용 인식을 촉진합니다. 쇼백은 사업부, 애플리케이션, 사용자 또는 비용 센터와 같은 특정 주체가 부담한 요금을 제시, 계산 및 보고하는 것을 말합니다. 예를 들어 “인프라 엔지니어링 팀은 지난 달에 X달러의 AWS 지출을 담당했습니다.” 차지백이란 금융 시스템이나 저널 바우처와 같은 조직의 내부 회계 프로세스를 통해 발생한 비용을 해당 엔터티에 실제로 청구하는 것을 말합니다. 예: “인프라 엔지니어링 팀의 AWS 예산에서 X달러가 차감되었습니다.” 두 경우 모두 리소스에 적절하게 태그를 지정하면 엔터티에 비용을 할당하는 데 도움이 될 수 있습니다. 유일한 차이점은 누군가가 지불할 것으로 예상되는지 여부입니다.

조직의 재무 거버넌스를 위해서는 애플리케이션, 사업부, 비용 센터 및 팀 수준에서 발생하는 비용을 투명하게 계산해야 할 수 있습니다. 비용 할당 태그가 지원하는 비용 분담을 수행하면 적절하게 태그가 지정된 리소스에서 엔터티에서 발생한 비용을 정확하게 귀속시키는 데 필요한 데이터가 제공됩니다.

- 책임 — 리소스 사용을 책임지는 사람에게 비용이 할당되도록 하십시오. 단일 서비스 지점 또는 그룹이 지출 검토 및 보고를 담당할 수 있습니다.
- 재무 투명성 — 리더십을 위한 효과적인 대시보드와 의미 있는 비용 분석을 만들어 IT에 대한 현금 할당을 명확하게 보여 줍니다.
- 정보에 입각한 IT 투자 — 프로젝트, 애플리케이션 또는 비즈니스 라인을 기반으로 ROI를 추적하고 팀이 더 나은 비즈니스 결정을 내릴 수 있도록 지원합니다. 예를 들어 수익 창출 애플리케이션에 더 많은 자금을 할당합니다.

요약하면 비용 할당 태그는 다음과 같은 정보를 제공하는 데 도움이 될 수 있습니다.

- 지출을 담당하고 최적화를 책임지는 사람은 누구입니까?
- 지출이 발생하는 워크로드, 애플리케이션 또는 제품은 무엇입니까? 어떤 환경 또는 스테이지입니까?
- 가장 빠르게 성장하고 있는 지출 영역은 어디입니까?
- 과거 추세를 기준으로 AWS 예산에서 얼마나 많은 지출을 공제할 수 있나요?
- 특정 워크로드, 애플리케이션 또는 제품 내에서 비용 최적화 노력이 미친 영향은 무엇이었습니까?

비용 할당을 위한 리소스 태그를 활성화하면 지출 책임에 대한 투명성을 높이는 AWS 사용량 가시성을 제공하는 데 사용할 수 있는 조직 내 측정 방식을 정의하는 데 도움이 됩니다. 또한 비용 및 사용량 가시성과 관련하여 적절한 수준의 세분성을 확보하고 비용 할당 보고 및 KPI 추적을 통해 클라우드 소비 행동에 영향을 미치는 데 중점을 둡니다.

비용 할당 전략 구축

비용 할당 모델 정의 및 구현

AWS에 배포되는 리소스에 대한 계정 및 비용 구조를 생성합니다. AWS 지출로 인한 비용, 비용이 어떻게 발생했는지, 누가 또는 무엇이 비용을 발생시켰는지 간의 관계를 설정합니다. 일반적인 비용 구조는 사업부 또는 워크로드와 같은 조직 내 AWS Organizations, AWS 계정, 환경 및 엔터티를 기반으로 합니다. 비용 구조는 여러 속성을 기반으로 하여 비용을 다양한 방식이나 다양한 수준으로 세분화하여 검토할 수 있습니다. 예를 들어 개별 워크로드의 비용을 해당 업무별로 합산할 수 있습니다.

원하는 결과에 맞는 비용 구조를 선택할 때는 구현의 용이성과 원하는 정확성을 기준으로 비용 할당 메커니즘을 평가합니다. 여기에는 책임, 도구 가용성, 문화적 변화에 대한 고려 사항이 포함될 수 있습니다. AWS 고객이 일반적으로 시작하는 인기 있는 세 가지 비용 할당 모델은 다음과 같습니다.

- 계정 기반 — 이 모델은 최소한의 노력이 필요하고 쇼백 및 차지백에 대한 높은 정확도를 제공하며 계정 구조가 정의되어 있고 복수 계정을 사용한 AWS 환경 구성 백서의 권장 사항과도 일치하는 조직에 적합합니다. 이를 통해 계정별로 비용을 명확하게 파악할 수 있습니다. 비용 가시성 및 할당을 위해 AWS Cost Explorer, Cost & Usage Report와 비용 모니터링 및 추적을 위한 AWS Budgets를 사용할 수 있습니다. 이러한 도구는 AWS 계정별 필터링 및 그룹화 옵션을 기준으로 제공합니다. 비용 할당 관점에서 보면 이 모델은 개별 리소스의 정확한 태그 지정에 의존할 필요가 없습니다.
- 사업부 또는 팀 기반 — 기업 내 팀, 사업부 또는 조직에 비용을 할당할 수 있습니다. 이 모델은 적당한 노력이 필요하고 쇼백 및 차지백에 대한 높은 정확도를 제공하며 다양한 팀, 애플리케이션 및 워크로드 유형을 구분하여 정의된 계정 구조(일반적으로 AWS Organizations 사용)를 가진 조직에 적합합니다. 이를 통해 팀과 애플리케이션 전반의 비용을 명확하게 파악할 수 있으며 추가 혜택으로 단일 AWS 계정 내의 AWS 서비스 할당량 한도에 도달할 위험을 줄일 수 있습니다. 예를 들어 각 팀은 5개의 계정(`prod`, `staging`, `test`, `dev`, `sandbox`)을 가질 수 있으며 두 팀과 애플리케이션이 동일한 계정을 공유하지 않을 수 있습니다. 이러한 구조를 통해 AWS Cost Categories는 계정 또는 기타 태그("meta-tagging")를 범주로 그룹화하는 기능을 제공하며 이 범주는 이전 예제에서 언급한 도구에서 추적할 수 있습니다. AWS Organizations를 통해 계정과 조직 단위(OU)에 태그를 지정할 수 있다는 점은 중요합니다. 하지만 이러한 태그는 비용 할당 및 청구 보고에는 적용할 수 없습니다. 즉, OU별로 AWS Cost Explorer에서 비용을 그룹화하거나 필터링할 수 없습니다. AWS 이를 위해서는 Cost Categories를 사용해야 합니다.

- 태그 기반 — 이 모델은 이전 두 모델에 비해 더 많은 노력이 필요하며 요구 사항 및 최종 목표에 따라 쇼백 및 차지백에 대한 높은 정확도를 제공합니다. 다중 계정을 사용하여 AWS 환경 구성 백서에 설명된 방법을 채택할 것을 강력히 권장하지만 현실적으로 고객은 혼합되고 복잡한 계정 구조를 가지고 있어 마이그레이션하는 데 시간이 걸리는 경우가 많습니다. 이 시나리오에서는 엄격하고 효과적인 태그 지정 전략을 구현하는 것이 핵심이며 그 다음에는 결제 및 비용 관리 콘솔에서 비용 할당을 위한 관련 태그를 활성화합니다(예: AWS Organizations에서 태그는 Management Payer 계정에서만 비용 할당을 위해 활성화할 수 있음). 비용 할당을 위해 태그를 활성화한 후에는 이전 방법에서 언급한 비용 파악 및 할당 도구를 쇼백 및 차지백에 사용할 수 있습니다. 비용 할당 태그는 소급 적용되지 않으며 비용 할당을 위해 활성화된 후에만 청구 보고 및 비용 추적 도구에 표시됩니다.

요약하자면 사업부별로 비용을 추적해야 하는 경우 AWS Cost Categories를 사용하여 AWS 조직 내 연결된 계정을 적절히 그룹화하고 결제 보고서에서 이 그룹화를 확인할 수 있습니다. 프로덕션 및 비프로덕션 환경에 대해 별도의 계정을 만드는 경우 AWS Cost Explorer 등과 같은 도구에서 환경과 관련된 비용을 필터링하거나 AWS Budgets를 사용하여 해당 비용을 추적할 수도 있습니다. 마지막으로 예를 들어 개별 워크로드 또는 애플리케이션별로 보다 세부적인 비용 추적이 필요한 사용 사례의 경우 해당 계정 내의 리소스에 적절하게 태그를 지정하고 관리 계정에서 비용 할당을 위해 해당 태그 키를 활성화한 다음 청구 보고 도구의 태그 키를 기준으로 비용을 필터링할 수 있습니다.

비용 보고 및 모니터링 프로세스 수립

먼저 내부 이해관계자에게 중요한 비용 유형(예: 일일 지출, 계정별 비용, X 기준 비용, 상각 비용)을 파악합니다. 이렇게 하면 최종 AWS 청구서가 나올 때까지 기다리는 것보다 더 빠르게 예상치 못한 지출이나 비정상적인 지출로 인한 예산 위험을 완화할 수 있습니다. 태그는 이러한 보고 시나리오를 가능하게 하는 속성을 제공합니다. 보고를 통해 얻은 인사이트는 변칙적이고 예상치 못한 재정 예산 지출로 인한 영향을 줄이기 위한 조치를 취할 수 있습니다. 비용이 예상치 않게 급증하는 경우 제공되는 가치가 예상치 않게 급증했는지 평가하여 필요한 조치가 필요한지 판단하는 것이 중요합니다.

비용 할당을 지원하기 위한 태그 지정 전략을 개발할 때 다음 사항에 유의하세요.

- AWS Organizations - 여러 계정 내의 비용 할당은 계정, 계정 그룹 또는 해당 계정의 리소스에 대해 생성된 태그 그룹별로 수행할 수 있습니다. AWS Organizations의 개별 계정에 있는 리소스에 대해 생성된 태그는 관리 계정에서만 비용 할당에 사용할 수 있습니다.
- AWS 계정 - 서비스 또는 지역과 같은 추가 차원을 통해 하나의 AWS 계정 내 비용 할당을 수행할 수 있습니다. 계정 내에서 리소스에 추가로 태그를 지정하고 해당 리소스 태그 그룹과 함께 작업할 수 있습니다.

- 비용 할당 태그 - 필요한 경우 사용자가 만든 태그와 AWS 생성 태그를 모두 비용 할당을 위해 활성화할 수 있습니다. (AWS Organizations에서 관리 계정의) 결제 콘솔에서 비용 할당을 위한 태그를 활성화하면 쇼백 및 차지백에 도움이 됩니다.
- Cost Categories - AWS Cost Categories를 사용하면 AWS 조직 내에서 계정을 그룹화하고 태그를 그룹화("meta-tagging")할 수 있으며 이를 통해 AWS Cost Explorer, AWS Budgets 및 AWS Cost & Usage Report와 같은 도구를 통해 이러한 범주와 관련된 비용을 분석할 수 있습니다.

기업 내 사업부, 팀 또는 조직에 대한 쇼백 및 차지백 수행

비용 구조 및 비용 할당 태그가 지원하는 비용 할당 프로세스를 사용하여 비용을 계산합니다. 태그를 사용하여 비용을 직접 지불할 책임이 없지만 해당 비용이 발생했을 책임이 있는 팀에 쇼백을 제공할 수 있습니다. 이 접근 방식을 통해 지출 기여도와 이러한 비용이 어떻게 발생하는지를 파악할 수 있습니다. 비용을 직접 담당하는 팀에 차지백을 수행하여 해당 팀이 소비한 리소스의 비용을 회수하고 해당 비용과 발생 경위를 파악할 수 있도록 합니다.

효율성 또는 가치 KPI 측정 및 순환

일련의 단위 비용 또는 KPI 지표에 합의하여 클라우드 재무 관리 투자의 영향을 측정합니다. 이 실습을 통해 절대적인 총지출에만 초점을 맞춘 이야기가 아니라 기술 및 비즈니스 이해관계자 간에 공통된 언어를 만들고 효율성을 기반으로 한 이야기를 들려줍니다. 자세한 내용은 단위 메트릭이 비즈니스 기능 간의 조정에 어떻게 도움이 될 수 있는지 설명하는 이 블로그를 확인하세요.

할당할 수 없는 지출 할당

조직의 회계 관행에 따라 청구 유형에 따라 다르게 처리해야 할 수도 있습니다. 태그를 지정할 수 없는 리소스 또는 비용 범주를 식별합니다. 사용하는 서비스와 사용할 예정인 서비스에 따라 할당할 수 없는 이러한 지출을 처리하고 측정하는 방법에 대한 메커니즘에 동의합니다. 예를 들어 및 AWS Resource Groups 태그 사용 설명서에서 AWS Resource Groups 및 Tag Editor에서 지원하는 리소스 목록을 확인합니다.

태그를 지정할 수 없는 비용 범주의 일반적인 예로는 예약형 인스턴스(RI) 및 절감형 플랜(SP)과 같은 약정 기반 할인에 대한 일부 수수료가 있습니다. 구독료와 미사용 SP 및 RI 요금을 청구 보고 도구에 표시하기 전에 미리 태그를 지정할 수는 없지만 사후에 AWS Organizations에서 계정, 리소스 및 태그에 RI와 SP 할인이 어떻게 적용되는지 추적할 수 있습니다. 예를 들어 분할 AWS Cost Explorer에서 상환 비용을 살펴보고 관련 태그 키별로 지출을 그룹화하고 사용 사례와 관련된 필터를 적용할 수 있습니다. AWS CUR(Cost & Usage Report)에서 RI 및 SP 할인이 적용되는 사용량에 해당하는 줄을 필터링하고(자세한 내용은 CUR 설명서의 사용 사례 섹션 참조) 해당 항목에만 해당하는 열을 선택할 수 있습니다. 비용 할당을 위해 활성화된 각 태그 키는 월별 비용 할당 보고서와 같은 다른 기존 결제 보고

서에 표시되는 방식과 유사하게 CUR 보고서 끝에 별도의 열에 표시됩니다. 추가 참조는 AWS Well-Architected Labs에서 CUR 데이터에서 비용 및 사용량 인사이트를 얻는 예제를 확인하세요.

보고

쇼백 및 차지백을 지원하는 AWS 도구 외에도 태그가 지정된 리소스의 비용을 모니터링하고 태그 지정 전략의 효과를 측정하는 데 도움이 되는 다양한 AWS 제작 및 타사 솔루션이 있습니다. 조직의 요구 사항과 최종 목표에 따라 시간과 리소스를 투자하여 맞춤형 솔루션을 구축하거나 AWS 클라우드 관리 도구 컴피턴시 파트너 중 한 곳에서 제공하는 도구를 구매할 수 있습니다. 비즈니스와 관련된 제어된 매개 변수를 사용하여 신뢰할 수 있는 단일 소스 비용 할당 도구를 만들기로 결정한 경우 AWS CUR(Cost & Usage Report)에서 가장 상세한 비용 및 사용 데이터를 제공하고 사용자 지정 최적화 대시보드를 만들 수 있으므로 계정, 서비스, 비용 범주, 비용 할당 태그 및 기타 여러 차원별로 필터링하고 그룹화할 수 있습니다. 이러한 도구 중 하나로 사용할 수 있는 AWS에서 개발한 CUR 기반 솔루션 중에서 AWS Well-Architected Labs 웹 사이트에서 클라우드 인텔리전스 대시보드를 확인합니다.

운영 및 지원을 위한 태그

AWS 환경에는 운영 요구 사항이 서로 다른 여러 계정, 리소스 및 워크로드가 있습니다. 태그를 사용하여 운영 팀을 지원하는 데 필요한 컨텍스트와 지침을 제공하여 서비스 관리를 개선할 수 있습니다. 또한 태그를 사용하여 관리 리소스의 운영 거버넌스 투명성을 제공할 수 있습니다.

운영 태그의 일관된 정의를 이끄는 몇 가지 주요 요인은 다음과 같습니다.

- 자동화된 인프라 활동 중에 리소스를 필터링합니다. 리소스를 배포, 업데이트 또는 삭제하는 경우를 예로 들 수 있습니다. 또 다른 하나는 비용 최적화를 위한 리소스 확장과 근무 시간 외 사용량 감소를 들 수 있습니다. 실제 예제는 AWS 인스턴스 스케줄러 솔루션을 참조하세요.
- 격리되거나 더 이상 사용되지 않는 리소스 식별 정의된 수명을 초과했거나 내부 메커니즘에 의해 격리 대상으로 지정된 리소스는 지원 담당자가 조사하는 데 도움이 되도록 적절하게 태그를 지정해야 합니다. 지원 중단된 리소스는 격리, 보관, 삭제 전에 태그를 지정해야 합니다.
- 리소스 그룹에 대한 지원 요구 사항 리소스의 지원 요구 사항이 서로 다른 경우가 많습니다. 예를 들어 이러한 요구 사항은 팀 간에 협상하거나 애플리케이션 중요도의 일부로 설정할 수 있습니다. 운영 모델에 대한 추가 지침은 운영 우수성 요소에서 확인할 수 있습니다.
- 인시던트 관리 프로세스를 개선합니다. 지원 팀과 엔지니어, 주요 인시던트 관리(MIM) 팀은 인시던트 관리 프로세스의 투명성을 높이는 태그로 리소스에 태그를 지정함으로써 이벤트를 보다 효과적으로 관리할 수 있습니다.
- 백업 또한 태그를 사용하여 리소스를 백업해야 하는 빈도, 백업 복사본을 이동해야 하는 위치 또는 백업을 복원할 위치를 식별할 수 있습니다. AWS의 백업 및 복구 접근 방식에 대한 규범적 지침

- 패치 적용 AWS에서 실행 중인 변경 가능한 인스턴스를 패치하는 것은 전반적인 패치 전략과 제로데이 취약성 패치 적용 모두에서 매우 중요합니다. 광범위한 패치 전략에 대한 심층적인 지침은 규범적 지침에서 확인할 수 있습니다. 이 블로그에서는 제로데이 취약점의 패치에 대해 설명합니다.
- 운영 관찰성 운영 KPI 전략을 리소스 태그로 변환하면 운영 팀이 비즈니스 요구 사항을 개선하기 위한 목표 달성 여부를 더 잘 추적할 수 있습니다. KPI 전략을 개발하는 주제는 별개이지만 안정된 상태에서 운영되는 비즈니스 또는 변화의 영향과 결과를 측정할 수 있는 영역에 초점을 맞추는 경향이 있습니다. KPI 대시보드(AWSWell-Architected 랩)와 운영 KPI 워크숍(AWS 엔터프라이즈 지원 사전 서비스)은 모두 안정적인 상태에서 측정 성과를 다룹니다. AWS 기업 전략 블로그 기사 혁신의 성공 여부 측정에서는 IT 현대화 또는 온프레미스에서 AWS로의 마이그레이션과 같은 혁신 프로그램에 대한 KPI 측정치를 살펴봅니다.

자동화된 인프라 활동

인프라를 관리할 때 다양한 자동화 활동에 태그를 사용할 수 있습니다. 예를 들어 AWS Systems Manager를 사용하면 사용자가 생성하는 정의된 키-값 쌍으로 지정된 리소스에 대한 자동화 및 런북을 관리할 수 있습니다. 관리형 노드의 경우 운영 체제 및 환경별로 노드를 추적하거나 대상으로 삼는 태그 세트를 정의할 수 있습니다. 그런 다음 그룹의 모든 노드에 대해 업데이트 스크립트를 실행하거나 해당 노드의 상태를 검토할 수 있습니다. 또한 Systems Manager 리소스에 태그를 지정하여 자동화된 활동을 더욱 세분화하고 추적할 수 있습니다.

환경 리소스의 시작 및 중지 수명 주기를 자동화하면 모든 조직의 비용을 크게 절감할 수 있습니다. AWS의 인스턴스 스케줄러는 필요하지 않을 때 Amazon EC2 및 Amazon RDS 인스턴스를 시작하고 중지할 수 있는 솔루션의 한 예입니다. 예를 들어 주말에 실행할 필요가 없는 Amazon EC2 또는 Amazon RDS 인스턴스를 사용하는 개발자 환경은 해당 인스턴스의 종료가 제공할 수 있는 비용 절감 잠재력을 활용하지 못하고 있습니다. 팀과 환경의 요구 사항을 분석하고 이러한 리소스에 적절하게 태그를 지정하여 관리를 자동화하면 예산을 효과적으로 활용할 수 있습니다.

Amazon EC2 인스턴스의 인스턴스 스케줄러가 사용하는 스케줄 태그의 예:

```
{
    "Tags": [
        {
            "Key": "Schedule",
            "ResourceId": "i-1234567890abcdef8",
            "ResourceType": "instance",
            "Value": "mon-9am-fri-5pm"
        }
    ]
```

```
}
```

워크로드 수명 주기

지원 운영 데이터의 정확성을 검토합니다. 워크로드 수명 주기와 관련된 태그를 정기적으로 검토하고 적절한 이해관계자가 이러한 검토에 참여하는지 확인하십시오.

표 7 – 워크로드 수명 주기의 일부로서 운영 태그 검토

사용 사례	태그 키	이론적 근거	예제 값
계정 소유자	`example-inc:account-owner:owner`	계정 소유자 및 계정에 포함된 리소스	`ops-center` , `dev-ops`, `app-team`
계정 소유자 검토	`example-inc:account-owner:review`	계정 소유권 세부 정보가 최신 상태이고 정확한지 검토합니다.	<태그 지정 라이브러리에 정의된 올바른 형식의 검토 날짜>
데이터 소유자	`example-inc:data-owner:owner`	데이터가 있는 계정의 데이터 소유자	`bi-team`, `logistics` , `security`
데이터 소유자 검토	`example-inc:data-owner:review`	데이터 소유권 세부 정보가 최신 상태이고 정확한지 검토합니다.	<태그 지정 라이브러리에 정의된 올바른 형식의 검토 날짜>

일시 중지된 OU로 마이그레이션하기 전에 일시 중지 계정에 태그 할당

다중 계정을 사용한 AWS 환경 구성 백서에 설명된 대로 계정을 일시 중지하고 일시 중지된 OU로 이동하기 전에 계정 수명 주기에 대한 내부 추적 및 감사를 지원하기 위해 계정에 태그를 추가해야 합니다. 예를 들어 조직의 ITSM 티켓팅 시스템에 있는 상대 URL 또는 티켓 참조는 일시 중지된 애플리케이션의 감사 추적을 보여 줍니다.

표 8 - 워크로드 수명 주기가 새로운 단계로 접어들 때 운영 태그 추가

사용 사례	태그 키	이론적 근거	예제 값
계정 소유자	example-inc:account-owner:owner	계정 소유자 및 계정에 포함된 리소스	ops-center , dev-ops, app-team
데이터 소유자	example-inc:data-owner:owner	데이터가 있는 계정의 데이터 소유자	bi-team, logistics , security
일시 중지된 날짜	example-inc:suspension:date	계정이 일시 중지된 날짜	<태그 지정 라이브러리에 정의된 올바른 형식의 일시 중지된 날짜>
일시 중지 승인	example-inc:suspension:approval	계정 일시 중지 승인 링크	workload/deprecation

인시던트 관리

태그는 인시던트 로깅, 우선 순위 지정, 조사, 커뮤니케이션, 해결에서 종료에 이르기까지 인시던트 관리의 모든 단계에서 중요한 역할을 할 수 있습니다.

태그에는 인시던트를 로깅해야 하는 위치, 인시던트를 알려야 하는 팀, 정의된 에스컬레이션 우선 순위가 상세히 명시될 수 있습니다. 태그는 암호화되지 않는다는 점을 기억해야 합니다. 따라서 태그에 어떤 정보를 저장할지 고려하십시오. 또한 조직, 팀, 보고 라인의 책임도 달라지므로 이 정보를 보다 효과적으로 관리할 수 있는 보안 포털로 연결되는 링크를 저장하는 것도 고려하십시오. 이러한 태그는 배타적일 필요는 없습니다. 예를 들어 애플리케이션 ID를 사용하여 IT 서비스 관리 포털에서 에스컬레이션 경로를 조회할 수 있습니다. 운영 정의에서 이 태그가 다양한 용도로 사용되고 있다는 점을 분명히 하십시오.

운영 요구 사항 태그도 자세히 작성하여 인시던트 관리자와 운영 담당자가 인시던트 또는 이벤트에 대응하여 목표를 더욱 구체화하는 데 도움이 될 수 있습니다.

대응 팀이 해당 프로세스, 절차 및 문서를 식별하는 데 도움이 되도록 런북 및 플레이북의 관련 링크(지식 시스템 기반 URL)를 태그로 포함할 수 있습니다.

표 9 - 운영 태그를 사용하여 인시던트 관리 정보 제공

사용 사례	태그 키	이론적 근거	예제 값
인시던트 관리	example-inc:incident-management:escalationlog	지원 팀이 인시던트를 로깅하는 데 사용하는 시스템	jira, servicenow , zendesk
인시던트 관리	example-inc:incident-management:escalationpath	에스컬레이션 경로	ops-center , dev-ops, app-team
비용 할당 및 인시던트 관리	example-inc:cost-allocation:CostCenter	비용 센터별 비용 모니터링 다음은 비용 센터를 인시던트 로깅을 위한 애플리케이션 코드로 사용하는 이중 용도 태그의 예입니다.	123-*
백업 일정	example-inc:backup:schedule	리소스의 백업 일정	Daily
플레이북/인시던트 관리	example-inc:incident-management:playbook	문서화된 플레이북	webapp/incident/playbook

패치 적용

조직은 AWS Systems Manager Patch Manager 및 AWS Lambda를 사용하여 변경 가능한 컴퓨팅 환경에 대한 패치 전략을 자동화하고 변경 가능한 인스턴스를 해당 애플리케이션 환경의 정의된 패치 기준선에 맞춰 유지할 수 있습니다. 이러한 환경 내의 변경 가능한 인스턴스에 대한 태그 지정 전략은 해당 인스턴스를 패치 그룹 및 유지 관리 창에 할당하여 관리할 수 있습니다. 개발 → 테스트 → 제품 분할

에 대한 다음 예를 참조하세요. 변경 가능한 인스턴스의 패치 관리를 위한 AWS 규범적 지침이 제공됩니다.

표 10 - 운영 태그는 환경에 따라 다를 수 있음

개발	Staging	프로덕션

{
"Tags": [
{
"Key": "Maintenance
Window",
"ResourceId":
"i-012345678ab9ab1
11",
"ResourceType":
"instance",
"Value": "cron(30 23 ?
* TUE#1 *)"
},
{
"Key": "Name",
"ResourceId":
"i-012345678ab9ab2
22",
"ResourceType":
"instance",
"Value": "WEBAPP"
},
{
"Key": "Patch Group",
"ResourceId":
"i-012345678ab9ab3
33",
"ResourceType":
"instance",
"Value": "WEBAPP-DEV-
AL2"
}
]
}
	{
	"Tags": [
	{
	"Key": "Maintenance
	Window",
	"ResourceId":
	"i-012345678ab9ab4
	44",
	"ResourceType":
	"instance",
	"Value": "cron(30 23 ?
	* TUE#2 *)"
	},
	{
	"Key": "Name",
	"ResourceId":
	"i-012345678ab9ab5
	55",
	"ResourceType":
	"instance",
	"Value": "WEBAPP"
	},
	{
	"Key": "Patch Group",
	"ResourceId":
	"i-012345678ab9ab6
	66",
	"ResourceType":
	"instance",
	"Value": "WEBAPP-TEST-
	AL2"
	}
	]
	}
		{
		"Tags": [
		{
		"Key": "Maintenance
		Window",
		"ResourceId":
		"i-012345678ab9ab7
		77",
		"ResourceType":
		"instance",
		"Value": "cron(30 23 ?
		* TUE#3 *)"
		},
		{
		"Key": "Name",
		"ResourceId":
		"i-012345678ab9ab8
		88",
		"ResourceType":
		"instance",
		"Value": "WEBAPP"
		},
		{
		"Key": "Patch Group",
		"ResourceId":
		"i-012345678ab9ab9
		99",
		"ResourceType":
		"instance",
		"Value": "WEBAPP-PROD-
		AL2"
		}
		]
		}

패치 전략을 보완하기 위해 태그를 정의하여 제로데이 취약성을 관리할 수도 있습니다. 자세한 지침은 AWS Systems Manager를 사용한 당일 보안 패치를 통한 제로데이 취약성 방지를 참조하세요.

운영 관찰성

환경 성능에 대한 실행 가능한 인사이트를 얻고 문제를 감지하고 조사하는 데 도움이 되려면 관찰성이 필요합니다. 또한 가동 시간과 같은 핵심 성과 지표(KPI)와 서비스 수준 목표(SLO)를 정의하고 측정할 수 있는 보조 목적도 있습니다. 대부분의 조직에서 중요한 운영 KPI는 인시던트로부터의 평균 감지 시간(MTTD)과 평균 복구 시간(MTTR)입니다.

데이터를 수집한 다음 관련 태그를 수집하기 때문에 관찰성 전반에서 컨텍스트가 중요합니다. 초점을 맞추고 있는 서비스, 애플리케이션 또는 애플리케이션 계층에 관계없이 해당 특정 데이터 세트를 필터링하고 분석할 수 있습니다. 태그를 사용하여 CloudWatch 경보에 대한 온보딩을 자동화하여 특정 지표 임계값 위반 시 적절한 팀에 알림을 보낼 수 있습니다. 예를 들어 태그 키 `example-inc:ops:alarm-tag` 및 그 값은 CloudWatch 경보가 생성되었음을 나타낼 수 있습니다. 이를 보여주는 솔루션은 태그를 사용하여 Amazon EC2 인스턴스에 대한 Amazon CloudWatch 경보 생성 및 유지 관리에 설명되어 있습니다.

경보를 너무 많이 구성하면 알림 폭풍이 쉽게 발생할 수 있습니다. 운영자가 개별 경보를 수동으로 분류하고 우선 순위를 정하는 과정에서 많은 수의 경보나 알림이 운영자에게 빠르게 부담을 주고 전반적인 효율성을 떨어뜨릴 수 있습니다. 경보에 대한 추가 컨텍스트를 태그 형태로 제공할 수 있습니다. 즉, Amazon EventBridge 내에서 규칙을 정의하여 다운스트림 종속성 대신 업스트림 문제에 초점을 맞출 수 있습니다.

DevOps와 함께 운영의 역할을 간과하는 경우가 많지만 많은 조직에서 중앙 운영팀은 정규 업무 시간 외에도 중요한 첫 대응을 제공합니다. (이 모델에 대한 자세한 내용은 운영 우수성 백서에서 확인할 수 있습니다.) 워크로드를 소유하는 DevOps 팀과 달리 일반적으로 깊이 있는 지식을 가지고 있지 않기 때문에 대시보드 및 알림 내에서 태그가 제공하는 컨텍스트를 통해 문제의 올바른 런북을 찾거나 자동화된 런북을 시작할 수 있습니다(AWS Systems Manager를 사용하여 Amazon CloudWatch 경보 자동화 블로그 게시물 참조).

데이터 보안, 위험 관리 및 액세스 제어를 위한 태그

조직은 데이터 스토리지 및 처리의 적절한 처리와 관련하여 충족해야 할 다양한 요구 사항과 의무를 가지고 있습니다. 데이터 분류는 액세스 제어, 데이터 보존, 데이터 분석 및 규정 준수와 같은 여러 사용 사례의 중요한 선행 요소입니다.

데이터 보안 및 위험 관리

AWS 환경 내에서는 규정 준수 및 보안 요구 사항이 서로 다른 계정이 있을 수 있습니다. 예를 들어 개발자 샌드박스가 있고 결제 처리와 같이 규제가 엄격한 워크로드를 위한 프로덕션 환경을 호스팅하는 계정이 있을 수 있습니다. 이들을 서로 다른 계정으로 분리하면 별도의 보안 제어를 적용하고 민감한 데이터에 대한 액세스를 제한하고 규제된 워크로드의 감사 범위를 줄일 수 있습니다.

모든 워크로드에 대해 단일 표준을 채택하면 문제가 발생할 수 있습니다. 많은 제어 항목이 환경 전체에 동일하게 적용되지만 특정 규제 프레임워크를 충족할 필요가 없는 계정 및 개인 식별 데이터가 전혀 없는 계정(예: 개발자 샌드박스 또는 워크로드 개발 계정)에는 일부 제어 항목이 과도하거나 관련이 없습니다. 이로 인해 일반적으로 아무 조치도 취하지 않고 분류하고 종료해야 하는 오탐지 보안 결과가 발생하므로 조사해야 할 결과에 대한 노력이 줄어듭니다.

표 11 – 데이터 보안 및 위험 관리 태그의 예

사용 사례	태그 키	이론적 근거	예제 값
인시던트 관리	`example-inc:incident-management:escalationlog`	지원 팀이 인시던트를 로깅하는 데 사용하는 시스템	`jira`, `servicenow` , `zendesk`
인시던트 관리	`example-inc:incident-management:escalationpath`	에스컬레이션 경로	`ops-center` , `dev-ops`, `app-team`
데이터 분류	`example-inc:data:classification`	규정 준수 및 거버넌스를 위한 데이터 분류	`Public`, `Private`, `Confidential` , `Restricted`
규정 준수	`example-inc:compliance:framework`	워크로드에 적용되는 규정 준수 프레임워크 식별	`PCI-DSS`, `HIPAA`

AWS 환경 전반에 걸쳐 다양한 제어를 수동으로 관리하는 것은 시간이 많이 걸리고 오류가 발생하기 쉽습니다. 다음 단계는 적절한 보안 제어 항목의 배포를 자동화하고 해당 계정의 분류에 따라 리소스

검사를 구성하는 것입니다. 계정과 계정 내 리소스에 태그를 적용하면 제어 항목 배포를 자동화하고 워크로드에 맞게 구성할 수 있습니다.

예:

워크로드에는 `Private` 값이 있는 `example-inc:data:classification` 태그가 있는 Amazon S3 버킷이 포함됩니다. 보안 도구 자동화는 AWS Config 규칙 `s3-bucket-public-read-prohibited`를 배포합니다. Amazon S3 버킷의 퍼블릭 액세스 차단 설정, 버킷 정책, 버킷 액세스 제어 목록(ACL)을 확인하여 버킷 구성이 해당 데이터 분류에 적합한지 확인합니다. 버킷 콘텐츠가 분류와 일치하도록 Amazon Macie를 구성하여 개인 식별 정보(PII)를 확인할 수 있습니다. Amazon Macie를 사용하여 S3 버킷 데이터 분류 검증 블로그에서는 이 패턴을 더 자세히 살펴봅니다.

보험 및 의료와 같은 특정 규제 환경에는 필수 데이터 보존 정책이 적용될 수 있습니다. Amazon S3 수명 주기 정책과 함께 태그를 사용한 데이터 보존은 객체 전환 범위를 다른 스토리지 계층으로 지정하는 효과적이고 간단한 방법이 될 수 있습니다. Amazon S3 수명 주기 규칙을 사용하여 필수 보류 기간이 만료된 후 데이터 삭제를 위해 객체를 만료시킬 수도 있습니다. 이 프로세스에 대한 심층 가이드는 Amazon S3 수명 주기와 함께 객체 태그를 사용하여 데이터 수명 주기 단순화를 참조하세요.

또한 보안 탐지 결과를 분류하거나 처리할 때 태그를 사용하면 조사자에게 위험을 판단하는 데 도움이 되는 중요한 컨텍스트를 제공하고 해당 팀이 결과를 조사하거나 완화하도록 유도하는 데 도움이 될 수 있습니다.

자격 증명 관리 및 액세스 제어에 대한 태그

AWS IAM Identity Center을 사용하여 AWS 환경 전반의 액세스 제어를 관리할 때 태그를 사용하면 여러 패턴을 통해 규모를 조정할 수 있습니다. 적용할 수 있는 위임 패턴은 여러 가지가 있으며 일부는 태그 지정을 기반으로 합니다. 이를 개별적으로 다루고 각각에 대해 자세히 알아볼 수 있는 링크를 제공할 것입니다.

개별 리소스에 대한 ABAC

IAM Identity Center 사용자 및 IAM 역할은 태그를 기반으로 작업 및 리소스에 대한 액세스를 정의할 수 있는 속성 기반 액세스 제어(ABAC)를 지원합니다. ABAC를 사용하면 권한 정책을 업데이트해야 할 필요성을 줄이고 회사 디렉터리의 직원 속성을 기반으로 액세스할 수 있습니다. 이미 다중 계정 전략을 사용하고 있는 경우 역할 기반 액세스 제어(RBAC)와 함께 ABAC를 사용하여 동일한 계정을 운영하는 여러 팀이 서로 다른 리소스에 세밀하게 액세스할 수 있도록 할 수 있습니다. 예를 들어 IAM Identity Center 사용자 또는 IAM 역할에는 특정 Amazon EC2 인스턴스에 대한 액세스를 제한하는 조건이 포함될 수 있습니다. 그렇지 않으면 액세스하려면 각 정책에 명시적으로 나열되어야 하는 조건이 있습니다.

ABAC 인증 모델은 태그에 따라 운영 및 리소스에 액세스하므로 의도하지 않은 액세스를 방지하는 가드레일을 제공하는 것이 중요합니다. SCP는 특정 조건에서만 태그 수정을 허용하여 조직 전체에서 태그를 보호하는 데 사용할 수 있습니다. AWS Organizations에서 서비스 제어 정책을 사용하여 권한 부여에 사용되는 리소스 태그 보호 및 IAM 엔터티의 권한 범위 블로그에서는 이를 구현하는 방법에 대한 정보를 제공합니다.

수명이 긴 Amazon EC2 인스턴스를 사용하여 보다 전통적인 운영 방식을 지원하는 경우 이 접근 방식을 활용할 수 있습니다. Amazon EC2 인스턴스용 IAM Identity Center ABAC 구성 및 Systems Manager Session Manager 블로그에서는 이러한 형태의 속성 기반 액세스 제어에 대해 자세히 설명합니다. 앞서 언급했듯이 모든 리소스 유형이 태그 지정을 지원하는 것은 아니며 지원하는 리소스 유형도 모두 태그 정책을 사용한 적용을 지원하는 것은 아니므로 이 전략을 AWS 계정에 구현하기 전에 먼저 이를 평가하는 것이 좋습니다.

ABAC를 지원하는 서비스에 대해 알아보려면 IAM을 사용하는 AWS 서비스를 참조하세요.

결론

비용 할당 전략 구현부터 자동화 지원 또는 AWS 리소스 액세스 권한 부여에 이르기까지 다양한 목적으로 AWS 리소스에 태그를 지정할 수 있습니다. 관련된 이해 관계자 그룹의 수와 데이터 소싱 및 태그 거버넌스와 같은 고려 사항으로 인해 일부 조직에서는 태그 지정 전략을 구현하는 것이 어려울 수 있습니다.

이 백서에서는 운영 관행, 정의된 사용 사례, 프로세스에 관련된 이해 관계자, AWS에서 제공하는 도구 및 서비스를 기반으로 조직의 태그 지정 전략을 설계하고 구현하는 것과 관련된 권장 사항을 개략적으로 설명했습니다. 태그 지정 전략은 반복과 개선의 과정으로, 즉각적인 우선 순위에서 작게 시작하여 조직 전반의 관련 사용 사례를 식별한 다음 필요에 따라 태그 지정 스키마를 구현하고 확장하는 동시에 효과를 지속적으로 측정하고 개선하는 프로세스입니다. 우리는 조직 내에서 잘 정의된 태그 세트를 사용하면 조직의 전략 및 가치에 맞게 AWS 사용 및 소비를 해당 태그가 존재하는 리소스 및 비즈니스 목적을 담당하는 팀과 연관시킬 수 있다는 점을 지적했습니다.

기여자

다음은 이 문서의 기여자입니다.

- Chris Pates, Sr Specialist Technical Account Manager, Amazon Web Services
- Vijay Shekhar Rao, Enterprise Support Lead, Amazon Web Services
- Nataliya Godunok, Sr Specialist Technical Account Manager, Amazon Web Services
- Yogish Kutkunje Pai, Sr Solutions Architect, Amazon Internet Services Private Limited
- Jamie Ibbs, Sr Specialist Technical Account Manager, Amazon Web Services

참조 자료

자세한 정보는 다음 단원을 참조하세요.

- AWS re:Invent 2020: 거꾸로 작업: 혁신에 대한 Amazon의 접근 방식
- AWS 규범적 지침: AWS Systems Manager를 사용하여 하이브리드 클라우드의 변경 가능한 인스턴스에 대한 자동 패치 적용
- AWS 아키텍처 센터

AWS Well-Architected

- AWS Well-Architected Framework
- 운영 우수성 요소 - AWS Well-Architected Framework
- 재해 복구(DR) 계획 - AWS Well-Architected 신뢰성 요소
- 비용 최적화 요소 - AWS Well-Architected Framework
- AWS Well-Architected Labs: AWS에서 생성된 비용 할당 태그 활성화
- AWS Well-Architected Labs: 태그 정책
- AWS Well-Architected Labs: AWS CUR 쿼리 라이브러리

AWS 블로그

- AWS Health Aware — 조직 및 개인 AWS 계정에 대한 사용자 지정 AWS Health 알림
- API 이벤트에 대한 응답으로 Amazon EC2 리소스에 자동으로 태그를 지정하는 방법
- AWS에서 생성된 비용 할당 태그 및 사용자 정의 비용 할당 태그
- AWS Organizations을 통한 비용 태그 지정 및 보고
- AWS Systems Manager 패치 관리자를 사용하여 Windows EC2 인스턴스에 패치 적용
- AWS Systems Manager을 사용하여 당일 보안 패치를 적용하여 제로데이 취약성 방지

AWS 설명서

- 비용 할당 태그 사용 - AWS Billing and Cost Management 및 비용 관리 및 비용 관리
- AWS Cost & Usage Report란?
- AWS Resource Groups API 참조

- IAM 정책 태그를 사용하여 EC2 인스턴스 또는 EBS 볼륨 생성 방법을 제한하려면 어떻게 해야 합니까?
- 변경 가능 및 변경 불가 업데이트 모델

기타

- Bryar, C. 및 Carr, B.(2021) Working Backwards: Insights, Stories, and Secrets from Inside Amazon(거꾸로 작업: Amazon 내부의 통찰력, 이야기, 비밀) London Macmillan
- AWS CloudFormation 보호(GitHub)

문서 수정

이 백서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하면 됩니다.

변경 사항	설명	날짜
마이너 업데이트	자격 증명 관리 업데이트	2023년 3월 30일
마이너 개정	개별 리소스에 대한 ABAC의 참조가 업데이트되었습니다.	2023년 2월 24일
마이너 개정	IAM 모범 실무에 따라 가이드가 업데이트되었습니다. 자세한 내용은 IAM의 보안 모범 사례를 참조하세요.	2023년 2월 6일
주요 내용 개정	AWS Config 규칙 `required_tags` 에서 지원하는 리소스 유형에 대한 보다 구체적인 참조가 추가되었습니다.	2023년 1월 18일
주요 내용 개정	특히 자격 증명 영역의 최신 사례 및 서비스 기능을 포함하도록 업데이트되었습니다.	2022년 9월 29일
마이너 업데이트	PDF 버전의 테이블 서식이 수정되었습니다.	2022년 4월 25일
주요 내용 개정	문서 구조가 업데이트되고 태그 지정 전략 및 사용 사례 섹션이 확장되었습니다. 최신 도구, 기법, 사용 가능한 리소스를 기반으로 보다 규범적인 지침이 추가되었습니다.	2022년 4월 22일
최초 게시	백서가 처음 게시되었습니다.	2018년 12월 1일

> **Note**
>
> RSS 업데이트를 구독하려면 사용 중인 브라우저에서 RSS 플러그인을 활성화해야 합니다.

고지 사항

고객은 본 문서의 정보를 독립적으로 평가할 책임이 있습니다. 본 문서는 (a) 정보 제공의 목적으로만 제공되고, (b) 사전 통지 없이 변경될 수 있는 현재 AWS 제품 및 관행을 나타내고, (c) AWS 및 그 계열사, 공급업체 또는 라이선스 제공자로부터 어떠한 약속이나 보증도 하지 않습니다. AWS 제품 또는 서비스는 명시적이든 묵시적이든 어떠한 종류의 보증, 진술 또는 조건 없이 '있는 그대로' 제공됩니다. 고객에 대한 AWS의 책임 및 채무는 AWS 계약에 준거합니다. 본 문서는 AWS와 고객 간의 어떠한 계약도 구성하지 않으며 이를 변경하지도 않습니다.

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조서의 AWS 용어집을 참조하세요.