사용자 가이드

Elastic Load Balancing

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Elastic Load Balancing: 사용자 가이드

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

Elastic Load Balancing이란 무엇인가요? 1
- 로드 밸런서 이점 1
- Elastic Load Balancing의 특징 1
- Elastic Load Balancing 액세스 2
- 관련 서비스 2
- 요금 3

Elastic Load Balancing의 작동 방식 4
- 가용 영역 및 로드 밸런서 노드 4
  - 교차 영역 로드 밸런싱 5
  - 영역 이동 7
- 라우팅 요청 7
  - 라우팅 알고리즘 8
  - HTTP 연결 9
  - HTTP 헤더 10
  - HTTP 헤더 제한 10
- 로드 밸런서 체계 10
- IP 주소 유형 11
- 네트워크 MTU 13

시작하기 14
- Application Load Balancer 생성 14
- Network Load Balancer 생성 14
- Gateway Load Balancer 생성 14

보안 16
- 데이터 보호 17
  - 저장된 데이터 암호화 17
  - 전송 중 암호화 18
- 자격 증명 및 액세스 관리 18
  - 대상 18
  - ID를 통한 인증 19
  - 정책을 사용하여 액세스 관리 22
  - Elastic Load Balancing이 IAM과 작동하는 방식 24
  - 리소스 태그 지정 API 권한 36
  - 서비스 연결 역할 38
  - AWS 관리형 정책 39

규정 준수 확인 41
복원성 42
인프라 보안 43
네트워크 격리 43
네트워크 트래픽 제어 44
AWS PrivateLink 45
Elastic Load Balancing을 위한 인터페이스 엔드포인트 생성 45
Elastic Load Balancing을 위한 VPC 엔드포인트 정책 생성 45
API 요청 제한 47
제한 적용 방법 47
요청 속도 제한 47
요청 토큰 버킷 크기 및 리필 속도 48
API 요청 모니터링 51
결제 및 사용 보고서 52
Application Load Balancers 52
Network Load Balancers 53
Gateway Load Balancer 53
Classic Load Balancer 53
API 직접 호출 로깅 54
CloudTrail의 Elastic Load Balancing 관리 이벤트 55
Elastic Load Balancing 이벤트 예제 55
Classic Load Balancer 마이그레이션 60
마이그레이션의 이점 60
마이그레이션 마법사 61
복사 유틸리티 마이그레이션 62
수동 마이그레이션 63
사용자가 Classic Load Balancer를 생성하지 못하도록 방지 65
..... lxvii

Elastic Load Balancing이란 무엇인가요?

Elastic Load Balancing은 둘 이상의 가용 영역에서 EC2 인스턴스, 컨테이너, IP 주소 등 여러 대상에 걸쳐 수신되는 트래픽을 자동으로 분산합니다. 등록된 대상의 상태를 모니터링하면서 상태가 양호한 대상으로만 트래픽을 라우팅합니다. Elastic Load Balancing은 수신 트래픽의 변화에 따라 로드 밸런서 용량을 자동으로 조정합니다.

로드 밸런서 이점

로드 밸런서는 워크로드를 가상 서버와 같은 다수의 컴퓨팅 리소스로 분산합니다. 로드 밸런서를 사용하면 애플리케이션의 가용성과 내결함성이 높아집니다.

애플리케이션에 대한 요청의 전체적인 흐름을 방해하지 않고 필요에 따라 로드 밸런서에서 컴퓨팅 리소스를 추가 및 제거할 수 있습니다.

로드 밸런서가 정상적인 대상에만 요청을 보내도록 컴퓨팅 리소스의 상태를 모니터링하는 상태 확인을 구성할 수 있습니다. 또한 컴퓨팅 리소스가 주요 작업에 집중할 수 있도록 암호화 및 복호화 작업을 로드 밸런서로 오프로드할 수 있습니다.

Elastic Load Balancing의 특징

Elastic Load Balancing은 여러 로드 밸런서 유형을 지원합니다. 각자 필요에 따라 가장 적합한 로드 밸런서 유형을 선택할 수 있습니다. 자세한 내용은 Elastic Load Balancing 기능Elastic Load Balancing 제품의 비교를 참조하세요.Elastic Load Balancing

현재 세대 로드 밸런서에 대한 자세한 내용은 다음 설명서를 참조하세요.

- Application Load Balancer 사용 설명서
- Network Load Balancer 사용 설명서
- Gateway Load Balancer 사용 설명서

Classic Load Balancer는 Elastic Load Balancing의 이전 세대 로드 밸런서입니다. 현재 세대 로드 밸런서로 마이그레이션하는 것이 좋습니다. 자세한 내용은 Classic Load Balancer 마이그레이션을 참조하세요.

Elastic Load Balancing 액세스

다음 인터페이스 중 하나를 사용하여 로드 밸런서를 생성하고, 액세스하고, 관리할 수 있습니다.

- AWS Management Console —Elastic Load Balancing에 액세스할 때 사용할 수 있는 웹 인터페이스를 제공합니다.
- AWS 명령줄 인터페이스(AWS CLI) - Elastic Load Balancing을 비롯한 광범위한 AWS 서비스에 대한 명령을 제공합니다. AWS CLI 는 Windows, macOS 및 Linux에서 지원됩니다. 자세한 내용은 AWS Command Line Interface 단원을 참조하십시오.
- AWS SDKs- 언어별 APIs 제공하고 서명 계산, 요청 재시도 처리, 오류 처리와 같은 많은 연결 세부 정보를 처리합니다. 자세한 정보는 AWS SDK를 참조하세요.
- 쿼리 API— HTTPS 요청을 사용하여 호출하는 하위 수준의 API 작업을 제공합니다. 쿼리 API 사용은 Elastic Load Balancing에 액세스하는 가장 직접적인 방법입니다. 하지만 쿼리 API를 사용하려면 애플리케이션에서 요청에 서명할 해시 생성 및 오류 처리와 같은 하위 수준의 세부 정보를 처리해야 합니다. 자세한 내용은 다음 자료를 참조하세요.
  - Application Load Balancer, Network Load Balancer 및 Gateway Load Balancer - API 버전 2015-12-01
  - Classic Load Balancer - API 버전 2012-06-01

관련 서비스

Elastic Load Balancing은 다음 서비스를 통해 애플리케이션의 가용성 및 확장성을 개선합니다.

- Amazon EC2 — 클라우드에서 애플리케이션을 실행할 수 있는 가상 서버입니다. 로드 밸런서를 구성하여 EC2 인스턴스에 트래픽을 라우팅할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서를 참조하세요.
- Amazon EC2 Auto Scaling — 인스턴스에 장애가 발생하더라도 원하는 수의 인스턴스가 실행되도록 보장합니다. 또한 Amazon EC2 Auto Scaling을 사용하면 인스턴스 수요가 변경될 때 인스턴스 수를 자동으로 늘리거나 줄일 수 있습니다. Elastic Load Balancing에서 Auto Scaling을 활성화하면 Auto Scaling에 의해 시작된 인스턴스가 로드 밸런서에 자동으로 등록됩니다. 마찬가지로 Auto Scaling에 의해 종료된 인스턴스는 로드 밸런서에서 자동으로 등록 취소됩니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서를 참조하세요.
- AWS Certificate Manager — HTTPS 리스너를 생성할 때 ACM에서 제공한 인증서를 지정할 수 있습니다. 로드 밸런서는 인증서를 사용하여 연결을 종료하고 클라이언트의 요청을 암호화 해제합니다.

- Amazon CloudWatch — 로드 밸런서를 모니터링하고 필요에 따라 조치를 취할 수 있게 해줍니다. 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하십시오.
- Amazon ECS — EC2 인스턴스 클러스터에서 Docker 컨테이너를 실행, 중단 및 관리할 수 있게 해줍니다. 로드 밸런서를 구성하여 컨테이너에 트래픽을 라우팅할 수 있습니다. 자세한 내용은 Amazon Elastic Container Service 개발자 안내서를 참조하세요.
- AWS Global Accelerator - 애플리케이션의 가용성과 성능을 향상시킵니다. 액셀러레이터를 사용하여 하나 이상의 AWS 리전에서 여러 로드 밸런서에 트래픽을 분산합니다. 자세한 내용은 AWS Global Accelerator 개발자 안내서를 참조하십시오.
- Route 53 — 도메인 이름을 컴퓨터를 사용하여 서로 연결해주는 숫자로 된 IP 주소로 변환하여 방문자를 안정적이며 비용 효율적으로 웹 사이트로 라우팅하도록 합니다. 예를 들어 숫자 `www.example.com` IP 주소로 변환됩니다`192.0.2.1`.는 로드 밸런서와 같은 리소스에 URLs AWS 할당합니다. 그러나 기억하기 쉬운 URL이 필요한 경우도 있습니다. 예를 들어 도메인 이름을 로드 밸런서로 매핑할 수 있습니다. 자세한 내용은 Amazon Route 53 개발자 안내서를 참조하세요.
- AWS WAF - Application Load Balancer와 AWS WAF 함께를 사용하여 웹 액세스 제어 목록(웹 ACL)의 규칙에 따라 요청을 허용하거나 차단할 수 있습니다. 자세한 내용은 개발자 안내서AWS WAF를 참조하세요.

요금

로드 밸런서에서는 사용한 만큼만 지불하면 됩니다. 자세한 내용은 Elastic Load Balancing 요금을 참조하세요.

Elastic Load Balancing의 작동 방식

로드 밸런서는 클라이언트에서 오는 트래픽을 허용하고, 하나 이상의 가용 영역에서 등록된 대상(예: EC2 인스턴스)으로 요청을 라우팅합니다. 또한, 로드 밸런서는 등록된 대상의 상태를 모니터링하고 정상 대상으로만 트래픽이 라우팅되도록 합니다. 로드 밸런서가 비정상 대상을 감지하면, 해당 대상으로 트래픽 라우팅을 중단합니다. 그런 다음 대상이 다시 정상으로 감지되면 트래픽을 해당 대상으로 다시 라우팅합니다.

하나 이상의 리스너를 지정하여 들어오는 트래픽을 허용하도록 로드 밸런서를 구성합니다. 리스너는 연결 요청을 확인하는 프로세스입니다. 클라이언트와 로드 밸런서 간의 연결을 위한 프로토콜 및 포트 번호로 구성됩니다. 마찬가지로 로드 밸런서와 대상 간의 연결을 위한 프로토콜 및 포트 번호로 구성됩니다.

내용

- 가용 영역 및 로드 밸런서 노드
- 라우팅 요청
- 로드 밸런서 체계
- IP 주소 유형
- 로드 밸런서에 대한 네트워크 MTU

가용 영역 및 로드 밸런서 노드

로드 밸런서에서 가용 영역을 활성화하면 Elastic Load Balancing이 해당 가용 영역에서 로드 밸런서 노드를 생성합니다. 가용 영역에 대상을 등록하지만 가용 영역은 활성화하지 않는 경우 이러한 등록된 대상은 트래픽을 수신하지 않습니다. 활성화된 각 가용 영역에 등록된 대상이 하나 이상 있는지 확인하는 경우에 로드 밸런서가 가장 효과적입니다.

모든 로드 밸런서에 대해 여러 가용 영역을 활성화하는 것이 좋습니다. 그러나 Application Load Balancer를 사용하여 최소한 둘 이상의 가용 영역을 활성화해야 합니다. 이 구성을 사용하면 로드 밸런서가 트래픽을 계속 라우팅할 수 있습니다. 가용 영역 하나가 사용할 수 없는 상태가 되거나 정상 상태 대상을 가지고 있지 않은 경우, 로드 밸런서가 다른 가용 영역에 있는 정상 상태의 대상으로 트래픽을 라우팅할 수 있습니다.

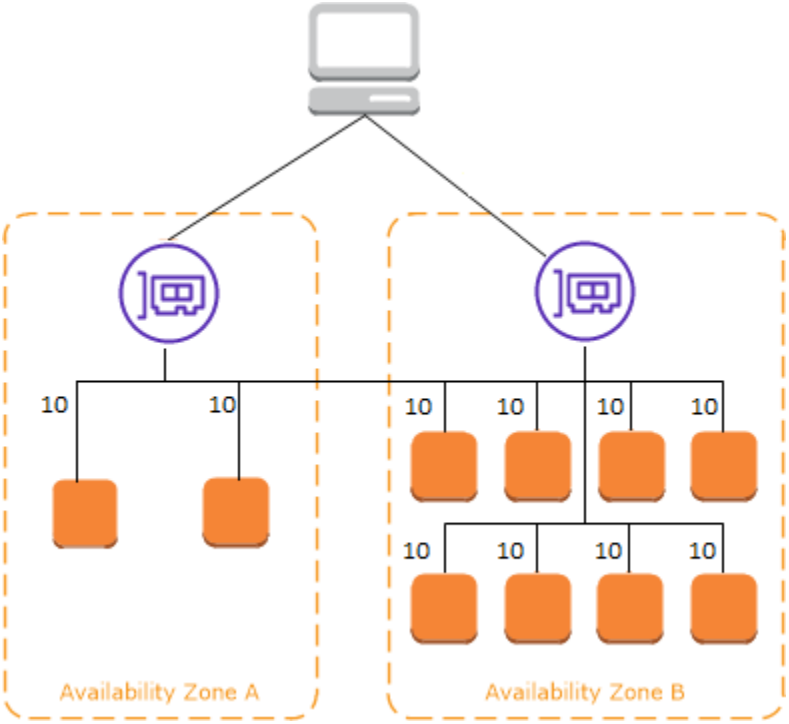
가용 영역을 비활성화해도 해당 가용 영역의 대상은 로드 밸런서에 등록된 상태로 남아 있습니다. 대상은 등록된 상태로 유지되지만 로드 밸런서는 트래픽을 해당 대상으로 라우팅하지 않습니다.

교차 영역 로드 밸런싱

로드 밸런서의 노드는 클라이언트로부터 요청을 가져와서 등록된 대상에 분산합니다. 교차 영역 로드 밸런싱을 활성화하면 각 로드 밸런서 노드가 활성화된 모든 가용 영역에 있는 등록된 대상 간에 트래픽을 분산합니다. 교차 영역 로드 밸런싱을 비활성화하면 각 로드 밸런서 노드가 해당 가용 영역에 있는 등록된 대상 간에만 트래픽을 분산합니다.

다음 다이어그램은 기본 라우팅 알고리즘으로서 라운드 로빈이 포함된 교차 영역 로드 밸런싱의 효과를 보여 줍니다. 활성화된 2개의 가용 영역이 있는데 가용 영역 A에는 2개의 대상이 있고 가용 영역 B에는 8개의 대상이 있습니다. 클라이언트는 요청을 보내며, Amazon Route 53은 로드 밸런서 노드 중 하나의 IP 주소를 통해 각 요청에 응답합니다. 라운드 로빈 라우팅 알고리즘에 따라 트래픽이 분산되므로 각 로드 밸런서 노드가 클라이언트가 보낸 트래픽의 50%를 수신하도록 트래픽이 분산됩니다. 각 로드 밸런서 노드는 공유 트래픽을 해당 범위의 등록된 대상에만 분산합니다.

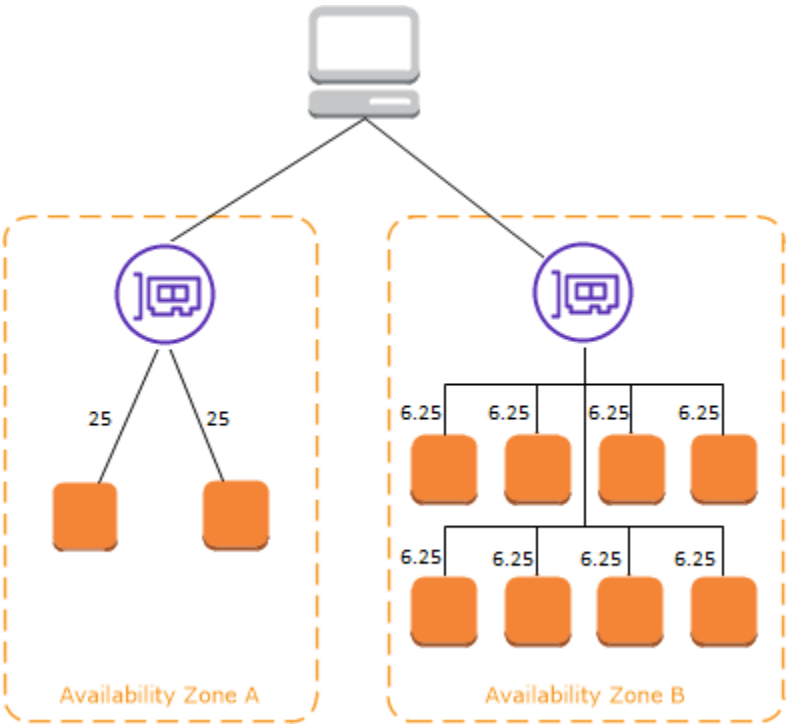
교차 영역 로드 밸런싱이 활성화된 경우 각 10개의 대상이 트래픽의 10%를 수신합니다. 이는 각 로드 밸런서 노드가 클라이언트 트래픽의 50%를 10개의 대상 모두에게 라우팅할 수 있기 때문입니다.



교차 영역 로드 밸런싱이 비활성화되어 있는 경우:

- 가용 영역 A의 두 대상이 각각 트래픽의 25%를 받습니다.
- 가용 영역 B에 있는 8개의 대상은 각각 트래픽의 6.25%를 받습니다.

이는 각 로드 밸런서 노드가 클라이언트 트래픽의 50%를 가용 영역에 있는 대상에만 라우팅할 수 있기 때문입니다.



Application Load Balancer를 사용하면 로드 밸런서 수준에서 교차 영역 로드 밸런싱이 항상 사용됩니다. 대상 그룹 수준에서 교차 영역 로드 밸런싱을 비활성화할 수 있습니다. 자세한 내용은 Application Load Balancer 사용 설명서의 교차 영역 로드 밸런싱 해제를 참조하세요.

Network Load Balancer 및 Gateway Load Balancer를 사용하면 기본적으로 교차 영역 로드 밸런싱이 비활성화됩니다. 로드 밸런서를 생성한 후 언제든지 교차 영역 로드 밸런싱을 활성화하거나 비활성화할 수 있습니다. 자세한 내용은 Network Load Balancer 사용 설명서의 교차 영역 로드 밸런싱을 참조하세요.

Classic Load Balancer를 생성하면 교차 영역 로드 밸런싱에 대한 기본값은 로드 밸런서 생성 방법에 따라 달라집니다. API 또는 CLI에서는 교차 영역 로드 밸런싱이 기본적으로 비활성화되어 있습니다. 에서는 교차 영역 로드 밸런싱 AWS Management Console을 활성화하는 옵션이 기본적으로 선택됩니다. Classic Load Balancer를 생성한 후 언제든지 교차 영역 로드 밸런싱을 활성화하거나 비활성화할 수 있습니다. 자세한 정보는 Classic Load Balancer 사용 설명서의 교차 영역 로드 밸런싱 활성화를 참조하세요.

영역 이동

영역 전환은 Amazon Application Recovery Controller(ARC) 내의 기능입니다. 영역 전환을 사용하면 한 번의 작업으로 손상된 가용 영역에서 로드 밸런서 리소스를 다른 곳으로 이동할 수 있습니다. 이러한 방법을 통해 AWS 리전의 다른 정상 가용 영역에서 계속 운영할 수 있습니다.

영역 전환을 시작하면 로드 밸런서가 해당 리소스에 대한 트래픽을 영향을 받는 가용 영역으로 보내는 것을 중단합니다. ARC는 영역 전환을 즉시 생성합니다. 그러나 영향을 받는 가용 영역에서 진행 중인 기존 연결을 완료하는 데는 보통 몇 분 정도 소요될 수 있습니다. 자세한 내용은 Amazon Application Recovery Controller(ARC) 개발자 안내서의 영역 전환 작동 방식: 상태 확인 및 영역 IP 주소를 참조하세요.

영역 전환을 사용하기 전에 다음을 검토하세요.

- 교차 영역 로드 밸런싱이 켜져 있거나 꺼져 있는 Network Load Balancer를 사용하는 경우 영역 전환이 지원됩니다.
- 특정 로드 밸런서에 대한 영역 전환은 단일 가용 영역에 대해서만 시작할 수 있습니다. 여러 가용 영역에 대한 영역 전환은 시작할 수 없습니다.
- AWS 는 여러 인프라 문제가 서비스에 영향을 미칠 때 DNS에서 영역 로드 밸런서 IP 주소를 사전에 제거합니다. 영역 전환을 시작하기 전에 항상 현재 가용 영역 용량을 확인하세요. 로드 밸런서의 교차 영역 로드 밸런싱이 꺼져 있으며 영역 전환을 사용하여 영역 로드 밸런서 IP 주소를 제거하는 경우, 영역 전환의 영향을 받는 가용 영역도 대상 용량을 잃게 됩니다.

자세한 지침과 정보는 Amazon Application Recovery Controller(ARC) 개발자 안내서의 ARC의 영역 전환 모범 사례를 참조하세요.

라우팅 요청

클라이언트는 로드 밸런서에 요청을 보내기 전에 로드 밸런서는 도메인 이름 시스템(DNS) 서버를 사용하여 로드 밸런서의 도메인 이름을 해석합니다. 로드 밸런서가 `amazonaws.com` 도메인에 있기 때문에 DNS 항목은 Amazon에서 제어합니다. Amazon DNS 서버는 클라이언트에 하나 이상의 IP 주소를 반환합니다. 이 주소는 로드 밸런서에 대한 로드 밸런서 노드의 IP 주소입니다. Network Load Balancer에서 Elastic Load Balancing은 활성화된 각 가용 영역에 대해 네트워크 인터페이스를 생성하고 이를 사용하여 정적 IP 주소를 가져옵니다. Network Load Balancer를 생성할 때 필요에 따라 탄력적 IP 주소 하나를 각 네트워크 인터페이스에 연결할 수 있습니다.

애플리케이션에 대한 트래픽이 시간에 따라 변화하므로 Elastic Load Balancing은 로드 밸런서를 확장하고 DNS 항목을 업데이트합니다. 또한 DNS 항목은 TTL을 60초로 지정합니다. 이렇게 하면 트래픽 변화에 따라 IP 주소를 신속하게 다시 매핑할 수 있습니다.

클라이언트는 로드 밸런서로 요청을 전송하는 데 사용할 IP 주소를 결정합니다. 요청을 수신한 로드 밸런서 노드는 등록된 대상 중 상태가 양호한 대상을 선택하고 프라이빗 IP 주소를 사용하여 해당 대상으로 요청을 전송합니다.

자세한 내용은 Amazon Route 53 개발자 안내서의 ELB 로드 밸런서로 트래픽 라우팅을 참조하세요.

라우팅 알고리즘

Application Load Balancer에서 요청을 수신하는 로드 밸런서 노드는 다음 프로세스를 사용합니다.

1. 적용할 규칙을 결정하기 위해 우선 순위에 따라 리스너 규칙을 평가합니다.
2. 대상 그룹에 대해 구성된 라우팅 알고리즘을 사용하여 규칙 조치에 대한 대상 그룹에서 대상을 선택합니다. 기본 라우팅 알고리즘은 라운드 로빈입니다. 대상이 여러 개의 대상 그룹에 등록이 된 경우에도 각 대상 그룹에 대해 독립적으로 라우팅이 수행됩니다.

Network Load Balancer에서 연결을 수신하는 로드 밸런서 노드는 다음 프로세스를 사용합니다.

1. 흐름 해시 알고리즘을 사용하여 기본 규칙에 대한 대상 그룹에서 대상을 선택합니다. 알고리즘은 다음을 기반으로 합니다.
   - 프로토콜
   - 소스 IP 주소 및 소스 포트
   - 대상 IP 주소 및 대상 포트
   - TCP 시퀀스 번호
2. 각 개별 TCP 연결은 연결 수명 동안 하나의 대상에 라우팅됩니다. 클라이언트로부터의 TCP 연결은 소스 포트와 시퀀스 번호가 서로 다르므로 다른 대상에 라우팅될 수 있습니다.

Gateway Load Balancer를 사용하면 연결을 수신하는 로드 밸런서 노드가 5튜플 흐름 해시 알고리즘을 사용하여 대상 어플라이언스를 선택합니다. 흐름이 설정되면 동일한 흐름에 대한 모든 패킷이 동일한 대상 어플라이언스로 일관되게 라우팅됩니다. 로드 밸런서와 대상 어플라이언스는 포트 6081에서 GENEVE 프로토콜을 사용하여 트래픽을 교환합니다.

Classic Load Balancer에서 요청을 수신하는 로드 밸런서 노드는 다음과 같이 등록된 인스턴스를 선택합니다.

- TCP 리스너에 대한 라운드 로빈 라우팅 알고리즘 사용
- HTTP 및 HTTPS 리스너에 대한 최소 미해결 요청 라우팅 알고리즘 사용

HTTP 연결

Classic Load Balancer는 사전 개방 연결을 사용하지만 Application Load Balancer는 그렇지 않습니다. Classic Load Balancer와 Application Load Balancer는 모두 연결 멀티플렉싱을 사용합니다. 즉, 여러 프런트 엔드 연결에 있는 여러 클라이언트의 요청을 단일 백엔드 연결을 통해 지정된 대상으로 라우팅할 수 있습니다. 연결 멀티플렉싱은 지연 시간을 최소화하고 애플리케이션의 로드를 줄입니다. 연결 멀티플렉싱을 하지 않으려면 HTTP 응답에 `keep-alive` 헤더를 설정하여 HTTP `Connection: close` 헤더를 비활성화합니다.

Application Load Balancer와 Classic Load Balancer는 프런트 엔드 연결에서 파이프라인 HTTP를 지원합니다. 백 엔드 연결에서는 파이프라인 HTTP를 지원하지 않습니다.

Application Load Balancer는 GET, HEAD, POST, PUT, DELETE, OPTIONS 및 PATCH HTTP 요청 메서드를 지원합니다.

Application Load Balancer는 프런트 엔드 연결에서 HTTP/0.9, HTTP/1.0, HTTP/1.1, HTTP/2 등의 프로토콜을 지원합니다. HTTPS 리스너에서만 HTTP/2를 사용할 수 있고, 하나의 HTTP/2 연결을 이용해 최대 128개의 요청을 동시에 전송할 수 있습니다. 또한 Application Load Balancer는 HTTP에서 WebSockets으로 연결을 업그레이드할 수 있도록 지원합니다. 그러나 연결 업그레이드가 있는 경우 Application Load Balancer 리스너 라우팅 규칙 및 AWS WAF 통합이 더 이상 적용되지 않습니다.

Application Load Balancer는 기본적으로 백엔드 연결(로드 밸런서를 등록된 대상에)에서 HTTP/1.1을 사용합니다. 그러나 프로토콜 버전을 사용하면 HTTP/2 또는 gRPC를 사용하여 대상에 요청을 보낼 수 있습니다. 자세한 내용은 프로토콜 버전을 참조하세요. `keep-alive` 헤더는 기본적으로 백엔드 연결에서 지원됩니다. 호스트 헤더가 없는 클라이언트로부터 오는 HTTP/1.0 요청의 경우, 로드 밸런서가 백 엔드 연결로 전송된 HTTP/1.1 요청에 대한 호스트 헤더를 생성합니다. 호스트 헤더에는 로드 밸런서의 DNS 이름이 포함되어 있습니다.

Classic Load Balancer는 프런트 엔드 연결(클라이언트에서 로드 밸런서)에서 HTTP/0.9, HTTP/1.0, HTTP/1.1 등의 프로토콜을 지원합니다. 백엔드 연결(등록된 대상에 로드 밸런서 연결)에서 HTTP/1.1을 사용합니다. `keep-alive` 헤더는 기본적으로 백엔드 연결에서 지원됩니다. 호스트 헤더가 없는 클라이언트로부터 오는 HTTP/1.0 요청의 경우, 로드 밸런서가 백 엔드 연결로 전송된 HTTP/1.1 요청에 대한 호스트 헤더를 생성합니다. 호스트 헤더에는 로드 밸런서 노드의 IP 주소가 포함되어 있습니다.

HTTP 헤더

Application Load Balancer와 Classic Load Balancer는 X-Forwarded-For, X-Forwarded-Proto, X-Forwarded-Port 헤더를 요청에 자동으로 추가합니다.

Application Load Balancer는 HTTP 호스트 헤더의 호스트 이름을 대상으로 보내기 전에 소문자로 변환합니다.

HTTP/2를 사용하는 프런트 엔드 연결의 경우, 헤더 이름이 소문자입니다. HTTP/1.1을 사용하여 대상에 요청이 전송되기 전에 X-Forwarded-For, X-Forwarded-Proto, X-Forwarded-Port, Host, X-Amzn-Trace-Id, Upgrade, Connection과 같이 헤더 이름이 대/소문자 혼용으로 변환됩니다. 기타 모든 헤더 이름은 소문자입니다.

Application Load Balancer와 Classic Load Balancer는 클라이언트로 다시 응답을 프록시한 후에 들어오는 클라이언트 요청에서 연결 헤더를 인식합니다.

HTTP/1.1을 사용하는 Application Load Balancers 및 Classic Load Balancers가 Expect: 100-Continue 헤더를 수신할 시, 내용 길이 헤더를 확인하지 않고 HTTP/1.1 100 Continue로 즉시 응답합니다. Expect: 100-Continue 요청 헤더는 대상으로 전달되지 않습니다.

HTTP/2를 사용하는 경우 Application Load Balancer는 클라이언트 요청 Expect: 100-Continue 헤더를 지원하지 않습니다. Application Load Balancer는 HTTP/2 100 Continue로 응답하거나 이 헤더를 대상으로 전달하지 않습니다.

HTTP 헤더 제한

Application Load Balancer에 대한 다음 크기 제한은 변경할 수 없는 하드 제한입니다.

- 요청 라인: 16K
- 단일 헤더: 16K
- 전체 응답 헤더: 32K
- 전체 요청 헤더: 64K

로드 밸런서 체계

로드 밸런서를 생성할 때 로드 밸런서를 내부 로드 밸런서 또는 인터넷 경계 로드 밸런서로 생성할지 여부를 선택해야 합니다.

인터넷 경계 로드 밸런서의 노드는 퍼블릭 IP 주소를 가집니다. 인터넷 경계 로드 밸런서의 DNS 이름은 노드의 퍼블릭 IP 주소로 공개적으로 확인이 가능합니다. 따라서 인터넷 경계 로드 밸런서는 인터넷을 통해 클라이언트의 요청을 라우팅할 수 있습니다.

내부 로드 밸런서의 노드는 오직 프라이빗 IP 주소만 가집니다. 내부 로드 밸런서의 DNS 이름은 노드의 프라이빗 IP 주소로 공개적으로 확인이 가능합니다. 따라서 내부 로드 밸런서는 로드 밸런서를 위한 VPC에 액세스하여 클라이언트의 요청만 라우팅할 수 있습니다.

인터넷 경계 및 내부 로드 밸런서는 모두 프라이빗 IP 주소를 사용하여 대상으로 요청을 라우팅합니다. 따라서 대상이 퍼블릭 IP 주소 없이도 내부 또는 인터넷 경계 로드 밸런서에서 요청을 수신할 수 있습니다.

애플리케이션에 여러 계층이 있는 경우 내부 및 인터넷 경계 로드 밸런서를 모두 사용하는 아키텍처를 설계할 수 있습니다. 예를 들어, 애플리케이션이 인터넷에 연결되어야 하는 웹 서버와 웹 서버에만 연결되는 애플리케이션 서버를 사용하는 경우에 해당됩니다. 인터넷 경계 로드 밸런서를 생성하고 여기에 웹 서버를 등록합니다. 내부 로드 밸런서를 생성하고 여기에 애플리케이션 서버를 등록합니다. 웹 서버는 인터넷 경계 로드 밸런서에서 요청을 수신하고 애플리케이션 서버에서 내부 로드 밸런서로 요청을 전송합니다. 애플리케이션 서버는 내부 로드 밸런서에서 요청을 수신합니다.

IP 주소 유형

로드 밸런서에 지정하는 IP 주소 유형은 클라이언트가 로드 밸런서와 통신하는 방법을 결정합니다.

- IPv4 전용 - 클라이언트는 퍼블릭 및 프라이빗 IPv4 주소를 사용하여 통신합니다. 로드 밸런서에 대해 선택한 서브넷에는 IPv4 주소 범위가 있어야 합니다.
- 듀얼 스택 - 클라이언트는 퍼블릭 및 프라이빗 IPv4 및 IPv6 주소를 사용하여 통신합니다. 로드 밸런서에 대해 선택한 서브넷에는 IPv4 및 IPv6 주소 범위가 있어야 합니다.
- 퍼블릭 IPv4가 없는 듀얼 스택 - 클라이언트는 퍼블릭 및 프라이빗 IPv6 주소와 프라이빗 IPv4 주소를 사용하여 통신합니다. 로드 밸런서에 대해 선택한 서브넷에는 IPv4 및 IPv6 주소 범위가 있어야 합니다. 이 옵션은 `internal` 로드 밸런서 체계에서는 지원되지 않습니다.

다음 표에서는 각 로드 밸런서 유형에 지원되는 IP 주소 유형을 설명합니다.

로드 밸런서 유형	IPv4 전용	듀얼 스택	퍼블릭 IPv4가 없는 듀얼 스택
Application Load Balancer	예	예	예
Network Load Balancer	예	예	아니요
Gateway Load Balancer	예	예	아니요
Classic Load Balancer	예	아니요	아니요

대상 그룹에 지정하는 IP 주소 유형은 로드 밸런서가 대상과 통신하는 방법을 결정합니다.

- IPv4 전용 - 로드 밸런서는 프라이빗 IPv4 주소를 사용하여 통신합니다. IPv4 대상 그룹에 IPv4 주소를 사용하는 대상을 등록해야 합니다.
- IPv6 전용 - 로드 밸런서는 IPv6 주소를 사용하여 통신합니다. IPv6 대상 그룹에 IPv6 주소를 사용하는 대상을 등록해야 합니다. 대상 그룹은 듀얼 스택 로드 밸런서와 함께 사용해야 합니다.

다음 표에서는 각 대상 그룹 프로토콜에 지원되는 IP 주소 유형을 설명합니다.

대상 그룹 프로토콜	IPv4 전용	IPv6 전용	
HTTP 및 HTTPS	예	예	
TCP	예	예	
TLS	예	예	
UDP 및 TCP_UDP	예	예	

대상 그룹 프로토콜	IPv4 전용	IPv6 전용	
GENEVE	-	-	

로드 밸런서에 대한 네트워크 MTU

최대 전송 단위(MTU) 는 네트워크를 통해 전송할 수 있는 최대 패킷의 크기(바이트)를 결정합니다. 연결의 MTU가 클수록 하나의 패킷으로 전달할 수 있는 데이터의 양이 늘어납니다. 이더넷 프레임은 패킷 또는 전송 중인 실제 데이터와 이를 둘러싼 네트워크 오버헤드 정보로 구성됩니다. 인터넷 게이트웨이를 통해 전송되는 트래픽은 1500 MTU입니다. 즉, 패킷이 1500바이트를 초과하면 여러 프레임을 사용하여 전송되도록 프래그먼트화되거나 `Don't Fragment`이(가) IP 헤더에 설정되면 패킷을 삭제합니다.

로드 밸런서 노드의 MTU 크기는 구성할 수 없습니다. 점보 프레임(9001 MTU)은 Application Load Balancer, Network Load Balancer 및 Classic Load Balancer용 로드 밸런서 노드 전체에서 표준입니다. 게이트웨이 로드 밸런서는 8500 MTU를 지원합니다. 자세한 내용은 User Guide for Gateway Load Balancers(Gateway Load Balancer 사용 설명서)의 Maximum transmission unit (MTU)(네트워크 MTU (최대 전송 단위))을 참조하세요.

경로 MTU는 발신 호스트와 수신 호스트 간의 경로에서 지원되는 최대 패킷 사이즈입니다. 경로 MTU 검색(PMTUD)을 사용하여 두 디바이스 간의 경로 MTU를 확인할 수 있습니다. 경로 MTU 검색은 클라이언트 또는 대상이 점보 프레임을 지원하지 않는 경우 특히 중요합니다.

호스트가 수신 호스트의 MTU 또는 경로를 따르는 디바이스의 MTU보다 큰 패킷을 전송하는 경우 수신 호스트 또는 디바이스가 패킷을 삭제한 다음 `Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (Type 3, Code 4)`과(와) 같은 ICMP 메시지를 반환합니다. 이렇게 하면 전송 호스트에 페이로드를 여러 개의 작은 패킷으로 분할한 다음 다시 전송하도록 지시합니다.

클라이언트 또는 대상 인터페이스의 MTU 크기보다 큰 패킷이 계속 삭제되면 경로 MTU 검색(PMTUD)이 작동하지 않는 것일 수 있습니다. 이를 방지하려면 경로 MTU 검색이 종단간에 작동하고, 클라이언트 및 대상에서 점보 프레임을 사용하도록 설정했는지 확인하세요. 경로 MTU 검색 및 점보 프레임 사용에 대한 자세한 내용은 Amazon EC2 사용 설명서에서 경로 MTU 검색을 참조하세요.

Elastic Load Balancing 시작하기

Elastic Load Balancing은 여러 로드 밸런서 유형을 지원합니다. 각자 필요에 따라 가장 적합한 로드 밸런서 유형을 선택할 수 있습니다. 자세한 내용은 Elastic Load Balancing 기능Elastic Load Balancing 제품의 비교를 참조하세요.Elastic Load Balancing

일반적인 로드 밸런서 구성에 대한 데모는 Elastic Load Balancing 데모를 참조하세요.

기존 Classic Load Balancer가 있는 경우 Application Load Balancer 또는 Network Load Balancer로 마이그레이션할 수 있습니다. 자세한 내용은 Classic Load Balancer 마이그레이션 단원을 참조하십시오.

목차

- Application Load Balancer 생성
- Network Load Balancer 생성
- Gateway Load Balancer 생성

Application Load Balancer 생성

를 사용하여 Application Load Balancer를 생성하려면 Application Load Balancer 사용 설명서의 Application Load Balancer 시작하기를 AWS Management Console참조하세요.

를 사용하여 Application Load Balancer를 생성하려면 Application Load Balancer 사용 설명서의를 사용하여 AWS CLI Application Load Balancer 생성을 AWS CLI참조하세요.

Network Load Balancer 생성

를 사용하여 Network Load Balancer를 생성하려면 Network Load Balancer 사용 설명서의 Network Load Balancer 시작하기를 AWS Management Console참조하세요.

를 사용하여 Network Load Balancer를 생성하려면 Network Load Balancer 사용 설명서의를 사용하여 AWS CLI Network Load Balancer 생성을 AWS CLI참조하세요.

Gateway Load Balancer 생성

를 사용하여 Gateway Load Balancer를 생성하려면 Gateway Load Balancer 사용 설명서의 Gateway Load Balancer 시작하기를 AWS Management Console참조하세요.

를 사용하여 Gateway Load Balancer를 생성하려면 Gateway Load Balancer 사용 설명서의를 사용하여 AWS CLI Gateway Load Balancer 시작하기를 AWS CLI참조하세요.

Elastic Load Balancing의 보안

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

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 규정 AWS 준수 프로그램 일환으로 보안의 효과를 정기적으로 테스트하고 확인합니다. Elastic Load Balancing에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 AWS 프로그램 제공 범위 내 서비스규정 준수 .
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Elastic Load Balancing을 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 보안 및 규정 준수 목표에 맞게 Elastic Load Balancing을 구성하는 방법을 보여줍니다. 또한 Elastic Load Balancing 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

Gateway Load Balancer를 사용하면 어플라이언스 공급업체의 소프트웨어를 선택하고 검증할 책임이 귀하에게 있습니다. 로드 밸런서의 트래픽을 검사하거나 수정하려면 어플라이언스 소프트웨어를 신뢰해야 합니다. 이 로드 밸런서는 OSI(Open Systems Interconnection) 모델의 계층 3인 네트워크 계층에서 작동합니다. Elastic Load Balancing 파트너로 등록된 어플라이언스 공급업체는 어플라이언스 소프트웨어를와 통합하고 자격을 부여했습니다 AWS. 이 목록에 있는 공급업체의 어플라이언스 소프트웨어에 대한 신뢰도를 높일 수 있습니다. 그러나는 이러한 공급업체의 소프트웨어의 보안 또는 신뢰성을 보장하지 AWS 않습니다.

내용

- Elastic Load Balancing의 데이터 보호
- Elastic Load Balancing의 ID 및 액세스 관리
- Elastic Load Balancing 규정 준수 검증
- Elastic Load Balancing의 탄력성
- Elastic Load Balancing의 인프라 보안
- 인터페이스 엔드포인트를 사용하여 Elastic Load Balancing에 액세스 (AWS PrivateLink)

Elastic Load Balancing의 데이터 보호

AWS 공동 책임 모델 Elastic Load Balancing의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS 는 모든를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스 의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 FAQ를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

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

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

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 콘솔, API 또는 AWS SDKs를 사용하여 Elastic Load Balancing AWS CLI또는 기타 AWS 서비스 작업을 수행하는 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 보안 인증 정보를 URL에 포함시켜서는 안 됩니다.

저장된 데이터 암호화

Elastic Load Balancing 액세스 로그를 위해 S3 버킷에 대한 Amazon S3 관리형 암호화 키(SSE-S3)를 사용하여 서버 측 암호화를 활성화하는 경우 Elastic Load Balancing은 각 액세스 로그 파일이 S3 버킷

에 저장되기 전에 자동으로 암호화합니다. 또한 Elastic Load Balancing은 액세스 로그 파일에 액세스할 때 해당 파일을 복호화합니다. 각 로그 파일은 고유 키로 암호화되며, 주기적으로 바뀌는 KMS 키를 사용하여 키 자체가 암호화됩니다.

전송 중 암호화

Elastic Load Balancing은 로드 밸런서에서 클라이언트의 HTTPS 및 TLS 트래픽을 종료하여 안전한 웹 애플리케이션 구축 프로세스를 단순화합니다. 로드 밸런서는 각 EC2 인스턴스가 TLS 종료 작업을 처리하도록 요구하지 않고 트래픽을 암호화하고 해독하는 작업을 수행합니다. 보안 리스너를 구성할 때 애플리케이션에서 지원하는 암호 모음 및 프로토콜 버전과 로드 밸런서에 설치할 서버 인증서를 지정합니다. AWS Certificate Manager (ACM) 또는 AWS Identity and Access Management (IAM)을 사용하여 서버 인증서를 관리할 수 있습니다. Application Load Balancer는 HTTPS 리스너를 지원합니다. Network Load Balancer는 TLS 리스너를 지원합니다. Classic Load Balancer는 HTTPS 및 TLS 리스너를 모두 지원합니다.

Elastic Load Balancing의 ID 및 액세스 관리

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는 입니다. IAM 관리자는 어떤 사용자가 Elastic Load Balancing 리소스를 사용할 수 있는 인증(로그인) 및 권한(사용 권한)을 받을 수 있는지 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

내용

- 대상
- ID를 통한 인증
- 정책을 사용하여 액세스 관리
- Elastic Load Balancing이 IAM과 작동하는 방식
- 생성 시 리소스 태그 지정에 대한 Elastic Load Balancing API 권한
- Elastic Load Balancing 서비스 연결 역할
- AWS Elastic Load Balancing에 대한 관리형 정책

대상

AWS Identity and Access Management (IAM)를 사용하는 방법은 Elastic Load Balancing에서 수행하는 작업에 따라 다릅니다.

서비스 사용자 - Elastic Load Balancing 서비스를 사용하여 작업을 수행하는 경우 필요한 보안 인증과 권한을 관리자가 제공합니다. 더 많은 Elastic Load Balancing 기능을 사용하여 작업을 수행하게 되면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방법을 이해하면 관리자에게 올바른 권한을 요청하는 데 도움이 됩니다.

서비스 관리자 - 회사에서 Elastic Load Balancing 리소스를 책임지고 있는 사용자는 Elastic Load Balancing에 대한 전체 액세스 권한을 가지고 있을 것입니다. 서비스 관리자는 서비스 사용자가 액세스해야 하는 Elastic Load Balancing 기능과 리소스를 결정합니다. 그런 다음 IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하세요.

IAM 관리자 - IAM 관리자라면 Elastic Load Balancing에 대한 액세스 권한 관리 정책 작성 방법을 자세히 알고 싶을 것입니다.

ID를 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여에 로그인하는 방법입니다. , AWS 계정 루트 사용자 IAM 사용자 또는 IAM 역할을 수임하여 인증(로그인 AWS)되어야 합니다.

자격 증명 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 자격 증명 AWS 으로에 로그인할 수 있습니다. AWS IAM Identity Center (IAM Identity Center) 사용자, 회사의 Single Sign-On 인증 및 Google 또는 Facebook 자격 증명은 페더레이션 자격 증명의 예입니다. 페더레이션형 ID로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 페더레이션을 사용하여 AWS 에 액세스하면 간접적으로 역할을 수임하게 됩니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 로그인에 대한 자세한 내용은 AWS 로그인 사용 설명서의에 로그인하는 방법을 AWS참조하세요. AWS 계정

AWS 프로그래밍 방식으로에 액세스하는 경우는 자격 증명을 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트(SDK)와 명령줄 인터페이스(CLI)를 AWS 제공합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 요청에 직접 서명하는 자세한 방법은 IAM 사용 설명서에서 API 요청용AWS Signature Version 4를 참조하세요.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어는 멀티 팩터 인증(MFA)을 사용하여 계정의 보안을 강화할 것을 AWS 권장합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 다중 인증 및 IAM 사용 설명서에서 IAM의AWS 다중 인증을 참조하세요.

AWS 계정 루트 사용자

를 생성할 때 계정의 모든 AWS 서비스 및 리소스에 대한 전체 액세스 권한이 있는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 이 자격 증명을 AWS 계정 루트 사용자라고 하며 계정을 생성하는 데 사용한 이메일 주소와 암호로 로그인하여 액세스합니다. 일상적인 작업에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 자격 증명을 보호하고 루트 사용자만 수행할 수 있는 작업을 수행하는 데 사용합니다. 루트 사용자로 로그인해야 하는 전체 작업 목록은 IAM 사용 설명서의 루트 사용자 보안 인증이 필요한 작업을 참조하세요.

페더레이션 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 포함한 인간 사용자에게 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 AWS 서비스 에 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 엔터프라이즈 사용자 디렉터리, 웹 자격 증명 공급자, AWS Directory Service, Identity Center 디렉터리 또는 자격 증명 소스를 통해 제공된 자격 증명을 사용하여 AWS 서비스 에 액세스하는 모든 사용자의 사용자입니다. 페더레이션 자격 증명에 액세스할 때 역할을 AWS 계정수임하고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center을(를) 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 모든 및 애플리케이션에서 사용할 수 있도록 자체 ID 소스의 사용자 AWS 계정 및 그룹 집합에 연결하고 동기화할 수 있습니다. IAM Identity Center에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 IAM Identity Center란 무엇인가요?를 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 사용자 또는 애플리케이션에 대한 특정 권한이 AWS 계정 있는 내의 자격 증명입니다. 가능하면 암호 및 액세스 키와 같은 장기 자격 증명이 있는 IAM 사용자를 생성하는 대신 임시 자격 증명을 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 참조하세요.

IAM 그룹은 IAM 사용자 컬렉션을 지정하는 자격 증명입니다. 사용자는 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, IAMAdmins라는 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 장기 자격 증명을 가지고 있

지만, 역할은 임시 보안 인증만 제공합니다. 자세한 내용은 IAM 사용 설명서에서 IAM 사용자 사용 사례를 참조하세요.

IAM 역할

IAM 역할은 특정 권한이 AWS 계정 있는 내의 자격 증명입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. 에서 IAM 역할을 일시적으로 수임하려면 사용자에서 IAM 역할(콘솔)로 전환할 AWS Management Console수 있습니다. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-console.html 또는 AWS API 작업을 호출하거나 사용자 지정 URL을 AWS CLI 사용하여 역할을 수임할 수 있습니다. 역할 사용 방법에 대한 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

임시 보안 인증이 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 ID가 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 관련 역할에 대한 자세한 내용은 IAM 사용 설명서의 Create a role for a third-party identity provider (federation)를 참조하세요. IAM Identity Center를 사용하는 경우, 권한 집합을 구성합니다. 인증 후 ID가 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 집합을 IAM의 역할과 연관짓습니다. 권한 집합에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 권한 집합을 참조하세요.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수임하여 특정 작업에 대한 다양한 권한을 임시로 받을 수 있습니다.
- 교차 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 에서는 정책을 리소스에 직접 연결할 AWS 서비스수 있습니다(역할을 프록시로 사용하는 대신). 교차 계정 액세스에 대한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 교차 계정 리소스 액세스를 참조하세요.
- 교차 서비스 액세스 - 일부는 다른에서 기능을 AWS 서비스 사용합니다 AWS 서비스. 예를 들어, 서비스에서 호출하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거나 Amazon S3에 객체를 저장합니다. 서비스는 직접적으로 호출하는 위탁자의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 작업을 수행할 수 있습니다.
  - 전달 액세스 세션(FAS) - IAM 사용자 또는 역할을 사용하여에서 작업을 수행하는 경우 AWS보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우, 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 사용합니다. FAS 요청은 서비스가 완료하기 위해 다른 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 수신할 때만 이루어집니다. 이 경우, 두 작업을 모두

수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.

- 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 맡는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 Create a role to delegate permissions to an AWS 서비스를 참조하세요.
- 서비스 연결 역할 - 서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 표시 AWS 계정 되며 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

- Amazon EC2에서 실행되는 애플리케이션 - IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS CLI 또는 AWS API 요청을 수행하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로필에는 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 참조하세요.

정책을 사용하여 액세스 관리

정책을 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결 AWS 될 때 권한을 정의하는의 객체입니다.는 보안 주체(사용자, 루트 사용자 또는 역할 세션)가 요청할 때 이러한 정책을 AWS 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는 지를 결정합니다. 대부분의 정책은에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 정보는 IAM 사용 설명서의 JSON 정책 개요를 참조하세요.

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.

기본적으로, 사용자 및 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` 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책이 있는 사용자는 AWS Management Console AWS CLI, 또는 API에서 역할 정보를 가져올 수 있습니다 AWS .

ID 기반 정책

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

ID 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은의 여러 사용자, 그룹 및 역할에 연결할 수 있는 독립 실행형 정책입니다 AWS 계정. 관리형 정책에는 AWS 관리형 정책 및 고객 관리형 정책이 포함됩니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책 및 인라인 정책 중에서 선택을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예제는 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 위탁자가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 위탁자를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는가 포함될 수 있습니다 AWS 서비스.

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

액세스 제어 목록(ACL)

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

Amazon S3 AWS WAF및 Amazon VPC는 ACLs. ACL에 관한 자세한 내용은 Amazon Simple Storage Service 개발자 가이드의 액세스 제어 목록(ACL) 개요를 참조하세요.

기타 정책 타입

AWS 는 덜 일반적인 추가 정책 유형을 지원합니다. 이러한 정책 타입은 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 얻

는 권한은 객체의 ID 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 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 엔티티에 대한 권한 경계를 참조하세요.

- 서비스 제어 정책(SCPs) - SCPs는의 조직 또는 조직 단위(OU)에 대한 최대 권한을 지정하는 JSON 정책입니다 AWS Organizations. AWS Organizations 는 기업이 소유한 여러 AWS 계정 을 그룹화하고 중앙에서 관리하기 위한 서비스입니다. 조직에서 모든 기능을 활성화할 경우, 서비스 제어 정책(SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 각각을 포함하여 멤버 계정의 엔터티에 대한 권한을 제한합니다 AWS 계정 루트 사용자. 조직 및 SCP에 대한 자세한 내용은 AWS Organizations 사용 설명서에서 Service control policies을 참조하세요.
- 리소스 제어 정책(RCP) - RCP는 소유한 각 리소스에 연결된 IAM 정책을 업데이트하지 않고 계정의 리소스에 대해 사용 가능한 최대 권한을 설정하는 데 사용할 수 있는 JSON 정책입니다. RCP는 멤버 계정의 리소스에 대한 권한을 제한하며 조직에 속하는지 여부에 AWS 계정 루트 사용자관계없이를 포함한 자격 증명에 대한 유효 권한에 영향을 미칠 수 있습니다. RCP를 AWS 서비스 지원하는 목록을 포함하여 조직 및 RCPs에 대한 자세한 내용은 AWS Organizations 사용 설명서의 리소스 제어 정책(RCPs)을 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 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 정보는 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 에서 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

Elastic Load Balancing이 IAM과 작동하는 방식

IAM을 사용하여 Elastic Load Balancing에 대한 액세스를 관리하기 전에 Elastic Load Balancing에서 사용할 수 있는 IAM 기능을 알아봅니다.

Elastic Load Balancing과 함께 사용할 수 있는 IAM 기능

IAM 기능	Elastic Load Balancing의 지원
ID 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예
정책 조건 키(서비스별)	예
ACLs	아니요
ABAC(정책의 태그)	예
임시 보안 인증	예
보안 주체 권한	예
서비스 역할	아니요
서비스 링크 역할	예

Elastic Load Balancing의 자격 증명 기반 정책

ID 기반 정책 지원: 예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. ID 기반 정책에서는 위탁자가 연결된 사용자 또는 역할에 적용되므로 위탁자를 지정할 수 없습니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

Elastic Load Balancing 내 리소스 기반 정책

리소스 기반 정책 지원: 아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예제는 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 위탁자가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 위탁자를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는가 포함될 수 있습니다 AWS 서비스.

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 위탁자로 지정할 수 있습니다. 리소스 기반 정책에 크로스 계정 보안 주체를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념하세요. 보안 주체와 리소스가 다른 경우 신뢰할 수 AWS 계정있는 계정의 IAM 관리자는 보안 주체 엔터티(사용자 또는 역할)에게 리소스에 액세스할 수 있는 권한도 부여해야 합니다. 엔터티에 ID 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 위탁자에 액세스를 부여하는 경우, 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 내용은 IAM 사용 설명서의 교차 계정 리소스 액세스를 참조하세요.

Elastic Load Balancing의 정책 작업

정책 작업 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 위탁자가 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 연결된 AWS API 작업과 이름이 동일합니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

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

Elastic Load Balancing 작업 목록을 보려면 서비스 승인 참조의 Elastic Load Balancing V2에서 정의한 작업 및 Elastic Load Balancing V1에서 정의한 작업을 참조하세요.

Elastic Load Balancing의 정책 작업은 작업 앞에 다음 접두사를 사용합니다:

```
elasticloadbalancing
```

단일 문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
    "elasticloadbalancing:action1",
    "elasticloadbalancing:action2"
]
```

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

```
"Action": "elasticloadbalancing:Describe*"
```

Elastic Load Balancing을 위한 API 작업의 전체 목록은 다음 설명서를 참조하세요.

- Application Load Balancer, Network Load Balancer 및 Gateway Load Balancer - API 참조 버전 2015-12-01
- Classic Load Balancer - API 참조 버전 2012-06-01

Elastic Load Balancing의 정책 리소스

정책 리소스 지원: 예

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. 문에는 `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 이를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

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

일부 Elastic Load Balancing API 작업은 여러 리소스를 지원합니다. 단일 문에서 여러 리소스를 지정하려면 ARN을 쉼표로 구분합니다.

```
"Resource": [
    "resource1",
    "resource2"
```

```
]
```

Elastic Load Balancing 리소스 유형 및 해당 ARNs 목록을 보려면 서비스 승인 참조의 Elastic Load Balancing V2에서 정의한 리소스와 Elastic Load Balancing V1에서 정의한 리소스를 참조하세요. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 Elastic Load Balancing V2에서 정의한 작업과 Elastic Load Balancing V1에서 정의한 작업을 참조하세요.

Elastic Load Balancing에 사용되는 정책 조건 키

서비스별 정책 조건 키 지원: 예

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 작음과 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 는 논리적 `AND` 작업을 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우는 논리적 `OR` 작업을 사용하여 조건을 AWS 평가합니다. 문의 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 내용은 IAM 사용 설명서의 IAM 정책 요소: 변수 및 태그를 참조하세요.

AWS 는 전역 조건 키와 서비스별 조건 키를 지원합니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

Elastic Load Balancing 조건 키 목록을 보려면 서비스 승인 참조의 Elastic Load Balancing V2에 사용되는 조건 키와 Elastic Load Balancing V1에 사용되는 조건 키를 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 Elastic Load Balancing V2에서 정의한 작업과 Elastic Load Balancing V1에서 정의한 작업을 참조하세요.

조건 키

- elasticloadbalancing:ListenerProtocol 조건 키
- elasticloadbalancing:SecurityPolicy 조건 키
- elasticloadbalancing:Scheme 조건 키

- elasticloadbalancing:SecurityGroup 조건 키
- elasticloadbalancing:Subnet 조건 키
- elasticloadbalancing:ResourceTag 조건 키

elasticloadbalancing:ListenerProtocol 조건 키

`elasticloadbalancing:ListenerProtocol` 조건 키는 생성 및 사용할 수 있는 리스너 유형을 정의하는 조건에 사용할 수 있습니다. 이 정책은 Application Load Balancer, Network Load Balancer 및 Classic Load Balancer에 사용할 수 있습니다. 다음 작업은 이 조건 키를 지원합니다.

API 버전 2015-12-01

- `CreateListener`
- `ModifyListener`

API 버전 2012-06-01

- `CreateLoadBalancer`
- `CreateLoadBalancerListeners`

다음 예제 정책은 사용자가 Application Load Balancer의 리스너에 대한 HTTPS 프로토콜과 Network Load Balancer의 리스너에 대한 TLS 프로토콜을 선택하도록 요구합니다.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "elasticloadbalancing:CreateListener",
            "elasticloadbalancing:ModifyListener"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals":{
                "elasticloadbalancing:ListenerProtocol": [
                    "HTTPS",
                    "TLS"
                ]
            },
```

```
        }
    }
}
```

Classic Load Balancer를 사용하면 단일 호출로 여러 리스너를 지정할 수 있습니다. 따라서 다음 예제와 같이 정책은 다중 값 컨텍스트 키를 사용해야 합니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticloadbalancing:CreateLoadBalancer",
                "elasticloadbalancing:CreateLoadBalancerListeners"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "elasticloadbalancing:ListenerProtocol": [
                        "TCP",
                        "HTTP",
                        "HTTPS"
                    ]
                }
            }
        }
    ]
}
```

elasticloadbalancing:SecurityPolicy 조건 키

`elasticloadbalancing:SecurityPolicy` 조건 키는 로드 밸런서에 특정 보안 정책을 정의 및 적용하는 조건에 사용할 수 있습니다. 이 정책은 Application Load Balancer, Network Load Balancer 및 Classic Load Balancer에 사용할 수 있습니다. 다음 작업은 이 조건 키를 지원합니다.

API 버전 2015-12-01

- `CreateListener`

- `ModifyListener`

API 버전 2012-06-01

- `CreateLoadBalancerPolicy`
- `SetLoadBalancerPoliciesOfListener`

다음 예제 정책은 사용자가 Application Load Balancer 및 Network Load Balancer에 대해 지정된 보안 정책 중 하나를 선택하도록 요구합니다.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "elasticloadbalancing:CreateListener",
            "elasticloadbalancing:ModifyListener"
        ],
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEquals":{
                "elasticloadbalancing:SecurityPolicy": [
                    "ELBSecurityPolicy-TLS13-1-2-2021-06",
                    "ELBSecurityPolicy-TLS13-1-2-Res-2021-06",
                    "ELBSEcurityPolicy-TLS13-1-1-2021-06"
                ]
            },
        }
    }
}
```

elasticloadbalancing:Scheme 조건 키

`elasticloadbalancing:Scheme` 조건 키는 로드 밸런서 생성 중에 선택할 수 있는 체계를 정의하는 조건에 사용할 수 있습니다. 이 정책은 Application Load Balancer, Network Load Balancer 및 Classic Load Balancer에 사용할 수 있습니다. 다음 작업은 이 조건 키를 지원합니다.

API 버전 2015-12-01

- `CreateLoadBalancer`

API 버전 2012-06-01

- `CreateLoadBalancer`

다음 예제 정책은 사용자가 로드 밸런서에 대해 지정된 체계를 선택하도록 요구합니다.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action":"elasticloadbalancing:CreateLoadBalancer",
        "Resource": "*",
        "Condition": {
            "StringEquals":{
                "elasticloadbalancing:Scheme": "internal"
            },
        }
    }
}
```

`elasticloadbalancing:SecurityGroup` 조건 키

> **Important**
>
> Elastic Load Balancing은 보안 그룹 IDs의 모든 대문자를 허용합니다. 그러나 `StringEqualsIgnoreCase`와 같이 대/소문자를 구분하지 않는 적절한 조건 연산자를 사용해야 합니다.

`elasticloadbalancing:SecurityGroup` 조건 키는 로드 밸런서에 적용할 수 있는 보안 그룹을 정의하는 조건에 사용할 수 있습니다. 이 정책은 Application Load Balancer, Network Load Balancer 및 Classic Load Balancer에 사용할 수 있습니다. 다음 작업은 이 조건 키를 지원합니다.

API 버전 2015-12-01

- `CreateLoadBalancer`
- `SetSecurityGroups`

API 버전 2012-06-01

- `CreateLoadBalancer`
- `ApplySecurityGroupsToLoadBalancer`

다음 예제 정책은 사용자가 로드 밸런서에 대해 지정된 보안 그룹 중 하나를 선택하도록 요구합니다.

```
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "elasticloadbalancing:CreateLoadBalancer",
            "elasticloadbalancing:SetSecurityGroup"
        ],
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEqualsIgnoreCase":{
                "elasticloadbalancing:SecurityGroup": [
                    "sg-51530134",
                    "sg-51530144",
                    "sg-51530139"
                ]
            },
        }
    }
}
```

elasticloadbalancing:Subnet 조건 키

> **Important**
>
> Elastic Load Balancing은 서브넷 IDs의 모든 대문자를 허용합니다. 그러나 `StringEqualsIgnoreCase`와 같이 대/소문자를 구분하지 않는 적절한 조건 연산자를 사용해야 합니다.

`elasticloadbalancing:Subnet` 조건 키는 생성하여 로드 밸런서에 연결할 수 있는 서브넷을 정의하는 조건에 사용할 수 있습니다. 이 정책은 Application Load Balancer, Network Load Balancer,

Gateway Load Balancer 및 Classic Load Balancer에 사용할 수 있습니다. 다음 작업은 이 조건 키를 지원합니다.

API 버전 2015-12-01

- `CreateLoadBalancer`
- `SetSubnets`

API 버전 2012-06-01

- `CreateLoadBalancer`
- `AttachLoadBalancerToSubnets`

다음 예제 정책은 사용자가 로드 밸런서에 대해 지정된 서브넷 중 하나를 선택하도록 요구합니다.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "elasticloadbalancing:CreateLoadBalancer",
            "elasticloadbalancing:SetSubnets"
        ],
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEqualsIgnoreCase":{
                "elasticloadbalancing:Subnet": [
                    "subnet-01234567890abcdef",
                    "subnet-01234567890abcdeg "
                ]
            },
        }
    }
}
```

elasticloadbalancing:ResourceTag 조건 키

`elasticloadbalancing:ResourceTag/#`는 Elastic Load Balancing에만 해당되는 특정한 조건 키입니다. 모든 변경 작업은이 조건 키를 지원합니다.

Elastic Load Balancing의 ACL

ACL 지원: 아니요

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

Elastic Load Balancing을 사용한 ABAC

ABAC 지원(정책의 태그): 예

속성 기반 액세스 제어(ABAC)는 속성에 근거하여 권한을 정의하는 권한 부여 전략입니다. 에서는 AWS이러한 속성을 태그라고 합니다. IAM 엔터티(사용자 또는 역할) 및 많은 AWS 리소스에 태그를 연결할 수 있습니다. ABAC의 첫 번째 단계로 개체 및 리소스에 태그를 지정합니다. 그런 다음 위탁자의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하도록 ABAC 정책을 설계합니다.

ABAC는 빠르게 성장하는 환경에서 유용하며 정책 관리가 번거로운 상황에 도움이 됩니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 내용은 IAM 사용 설명서의 ABAC 권한 부여를 통한 권한 정의를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하세요.

Elastic Load Balancing으로 임시 보안 인증 사용

임시 자격 증명 지원: 예

일부 AWS 서비스 는 임시 자격 증명을 사용하여 로그인할 때 작동하지 않습니다. 임시 자격 증명으로 AWS 서비스 작업하는를 비롯한 추가 정보는 AWS 서비스 IAM 사용 설명서의 IAM으로 작업하는 섹션을 참조하세요.

사용자 이름 및 암호를 제외한 방법을 AWS Management Console 사용하여에 로그인하는 경우 임시 자격 증명을 사용합니다. 예를 들어 회사의 SSO(Single Sign-On) 링크를 AWS 사용하여에 액세스하면 해당 프로세스가 임시 자격 증명을 자동으로 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 자격 증명을 자동으로 생성합니다. 역할 전환에 대한 자세한 내용은 IAM 사용 설명서의 사용자에서 IAM 역할로 전환(콘솔)을 참조하세요.

AWS CLI 또는 AWS API를 사용하여 임시 자격 증명을 수동으로 생성할 수 있습니다. 그런 다음 이러한 임시 자격 증명을 사용하여 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성하는 access AWS. AWS recommends에 액세스할 수 있습니다. 자세한 정보는 IAM의 임시 보안 자격 증명 섹션을 참조하세요.

Elastic Load Balancing의 서비스 간 주요 권한

전달 액세스 세션(FAS) 지원: 예

IAM 사용자 또는 역할을 사용하여에서 작업을 수행하는 경우 AWS보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우, 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 사용합니다. FAS 요청은 서비스가 완료하기 위해 다른 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 수신할 때만 이루어집니다. 이 경우, 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.

Elastic Load Balancing의 서비스 역할

서비스 역할 지원: 아니요

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 가정하는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 Create a role to delegate permissions to an AWS 서비스를 참조하세요.

Elastic Load Balancing의 서비스 연결 역할

서비스 링크 역할 지원: 예

서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 표시 AWS 계정 되며 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

Elastic Load Balancing 서비스 연결 역할을 생성 또는 관리하는 방법에 대한 자세한 내용은 Elastic Load Balancing 서비스 연결 역할 섹션을 참조하세요.

생성 시 리소스 태그 지정에 대한 Elastic Load Balancing API 권한

사용자가 생성 시 리소스에 태그를 지정할 수 있으려면 리소스를 생성하는 작업을 사용할 권한이 있어야 합니다(예: `elasticloadbalancing:CreateLoadBalancer` 또는 `elasticloadbalancing:CreateTargetGroup`). 리소스 생성 작업에서 태그가 지

정되면 사용자에게 리소스가 생성되는 동안 태그를 적용할 권한이 있는지 확인하기 위해 `elasticloadbalancing:AddTags` 작업에서 추가 권한 부여가 필요합니다. 따라서 사용자는 `elasticloadbalancing:AddTags` 작업을 사용할 명시적 권한도 가지고 있어야 합니다.

`elasticloadbalancing:AddTags` 작업에 대한 IAM 정책 정의에서 `elasticloadbalancing:CreateAction` 조건 키와 함께 `Condition` 요소를 사용하여 리소스를 생성하는 작업에 태그 지정 권한을 부여합니다.

다음 예제의 정책은 사용자가 대상 그룹을 생성하고 생성 도중 대상 그룹에 임의의 태그를 적용하는 것을 허용합니다. 사용자는 기존 리소스에 태그를 지정할 수 없습니다(`elasticloadbalancing:AddTags` 작업을 직접 호출할 수 없습니다).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "elasticloadbalancing:CreateTargetGroup"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "elasticloadbalancing:AddTags"
      ],
      "Resource": "*",
      "Condition": {
         "StringEquals": {
             "elasticloadbalancing:CreateAction" : "CreateTargetGroup"
         }
      }
    }
  ]
}
```

마찬가지로 다음 정책은 사용자가 로드 밸런서를 생성하고 생성 도중 태그를 적용하는 것을 허용합니다. 사용자는 기존 리소스에 태그를 지정할 수 없습니다(`elasticloadbalancing:AddTags` 작업을 직접 호출할 수 없습니다).

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "elasticloadbalancing:CreateLoadBalancer"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "elasticloadbalancing:AddTags"
      ],
      "Resource": "*",
      "Condition": {
         "StringEquals": {
             "elasticloadbalancing:CreateAction" : "CreateLoadBalancer"
         }
      }
    }
  ]
}
```

`elasticloadbalancing:AddTags` 작업은 리소스 생성 작업 도중 태그가 적용되는 경우에만 평가됩니다. 따라서 리소스를 생성할 권한이 있는 사용자(태그 지정 조건은 없다고 가정)는 요청에서 태그가 지정되지 않은 경우, `elasticloadbalancing:AddTags` 작업을 사용할 권한이 필요하지 않습니다. 하지만 사용자가 태그를 사용하여 리소스 생성을 시도하는 경우, 사용자에게 `elasticloadbalancing:AddTags` 작업을 사용할 권한이 없다면 요청은 실패합니다.

Elastic Load Balancing 서비스 연결 역할

Elastic Load Balancing은 다른 AWS 서비스를 자동으로 호출하는 데 필요한 권한에 대해 서비스 연결 역할을 사용합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할을 참조하세요.

서비스 연결 역할에 의해 부여된 권한

Elastic Load Balancing은 라는 서비스 연결 역할을 AWSServiceRoleForElasticLoadBalancing 사용하여 사용자를 대신하여 다른 AWS 서비스를 호출합니다.

AWSServiceRoleForElasticLoadBalancing는 `elasticloadbalancing.amazonaws.com` 서비스를 신뢰하여 역할을 수임합니다.

역할 권한 정책은 입니다AWSElasticLoadBalancingServiceRolePolicy. 이 정책의 권한을 보려면AWS 관리형 정책 참조의 AWSElasticLoadBalancingServiceRolePolicy를 참조하세요.

서비스 연결 역할 생성

AWSServiceRoleForElasticLoadBalancing 역할을 수동으로 생성할 필요가 없습니다. 로드 밸런서 또는 대상 그룹을 생성할 때 Elastic Load Balancing이 이 역할을 생성합니다.

Elastic Load Balancing이 서비스 연결 역할을 자동으로 생성하려면 사용자에게 필수 권한이 있어야 합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요.

서비스 연결 역할 편집

IAM AWSServiceRoleForElasticLoadBalancing 사용에 대한 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 설명 편집을 참조하세요.

서비스 연결 역할 삭제

Elastic Load Balancing을 더 이상 사용할 필요가 없는 경우를 삭제하는 것이 좋습니다 AWSServiceRoleForElasticLoadBalancing.

AWS 계정의 모든 로드 밸런서를 삭제한 후에만이 서비스 연결 역할을 삭제할 수 있습니다. 이렇게 하면 로드 밸런서에 대한 액세스 권한을 실수로 삭제하지 않습니다. 자세한 내용은 Application Load Balancer 삭제, Network Load Balancer 삭제 및 Classic Load Balancer 삭제를 참조하세요.

IAM 콘솔, IAM CLI 또는 IAM API를 사용하여 서비스 연결 역할을 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 삭제를 참조하세요.

를 삭제한 후 로드 밸런서를 생성하면 AWSServiceRoleForElasticLoadBalancingElastic Load Balancing이 역할을 다시 생성합니다.

AWS Elastic Load Balancing에 대한 관리형 정책

AWS 관리형 정책은에서 생성하고 관리하는 독립 실행형 정책입니다 AWS. AWS 관리형 정책은 사용자, 그룹 및 역할에 권한 할당을 시작할 수 있도록 많은 일반적인 사용 사례에 대한 권한을 제공하도록 설계되었습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있으므로 특정 사용 사례에 대해 최소 권한을 부여하지 않을 수 있습니다. 사용 사례에 고유한 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다.

AWS 관리형 정책에 정의된 권한은 변경할 수 없습니다. 가 관리형 정책에 정의된 권한을 AWS 업데이트하는 AWS 경우 업데이트는 정책이 연결된 모든 보안 주체 자격 증명(사용자, 그룹 및 역할)에 영향

을 미칩니다. AWS AWS 서비스 는 새가 시작되거나 기존 서비스에 새 API 작업을 사용할 수 있게 되면 AWS 관리형 정책을 업데이트할 가능성이 높습니다.

자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책을 참조하세요.

AWS 관리형 정책: AWSElasticLoadBalancingClassicServiceRolePolicy

이 정책에는 Elastic Load Balancing(Classic Load Balancer)이 사용자를 대신하여 다른 AWS 서비스를 호출하는 데 필요한 모든 권한이 포함됩니다. 서비스와 연결된 역할은 사전에 정의되어 있습니다. 사전 정의된 역할을 사용하면 Elastic Load Balancing에 필요한 권한을 수동으로 추가할 필요가 없습니다. 이 정책은 연결, 분리, 수정 또는 삭제할 수 없습니다.

이 정책의 권한을 보려면AWS 관리형 정책 참조의 AWSElasticLoadBalancingClassicServiceRolePolicy를 참조하세요.

AWS 관리형 정책: AWSElasticLoadBalancingServiceRolePolicy

이 정책은 Elastic Load Balancing이 다른 AWS 서비스를 자동으로 호출하는 데 필요한 권한을 포함합니다. 서비스와 연결된 역할은 사전에 정의되어 있습니다. 사전 정의된 역할을 사용하면 Elastic Load Balancing에 필요한 권한을 수동으로 추가할 필요가 없습니다. 이 정책은 연결, 분리, 수정 또는 삭제할 수 없습니다.

이 정책의 권한을 보려면AWS 관리형 정책 참조의 AWSElasticLoadBalancingServiceRolePolicy를 참조하세요.

AWS 관리형 정책: ElasticLoadBalancingFullAccess

이 정책은 Elastic Load Balancing 서비스에 대한 전체 액세스 권한을 부여하고 AWS Management Console을 통해 다른 서비스에 대한 제한된 액세스 권한을 부여합니다.

이 정책의 권한을 보려면AWS 관리형 정책 참조의 ElasticLoadBalancingFullAccess를 참조하세요.

AWS 관리형 정책: ElasticLoadBalancingReadOnly

이 정책은 Elastic Load Balancing 및 종속 서비스에 대한 읽기 전용 액세스 권한을 제공합니다.

이 정책의 권한을 보려면AWS 관리형 정책 참조의 ElasticLoadBalancingReadOnly를 참조하세요.

AWS 관리형 정책에 대한 Elastic Load Balancing 업데이트

이 서비스가 이러한 변경 사항을 추적하기 시작한 이후부터 Elastic Load Balancing의 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다.

변경 사항	설명	날짜
AWSElasticLoadBalancingServiceRolePolicy - 기존 정책에 대한 업데이트	IPAM 풀에서 CIDR 블록을 할당할 수 있는 권한을 부여하는 `ec2:AllocateIpamPoolCidr` 작업이 추가되었습니다.	2025년 2월 17일
ElasticLoadBalancingFullAccess - 기존 정책에 대한 업데이트	영역 전환에 필요한 권한을 부여하는 `arc-zonal-shift:*` 작업이 추가되었습니다.	2023년 11월 28일
ElasticLoadBalancingReadOnly - 기존 정책에 대한 업데이트	영역 전환에 필요한 권한을 부여하기 위해 `arc-zonal-shift:GetManagedResource` `arc-zonal-shift:ListManagedResources` 및 작업을 추가했습니다`arc-zonal-shift:ListZonalShifts` .	2023년 11월 28일
AWSElasticLoadBalancingServiceRolePolicy - 기존 정책에 대한 업데이트	피어링 연결에 필요한 권한을 부여하는 `ec2:DescribeVpcPeeringConnections` 작업이 추가되었습니다.	2021년 10월 11일
ElasticLoadBalancingFullAccess - 기존 정책에 대한 업데이트	피어링 연결에 필요한 권한을 부여하는 `ec2:DescribeVpcPeeringConnections` 작업이 추가되었습니다.	2021년 10월 11일
ElasticLoadBalancingFullAccess - 새 정책	Elastic Load Balancing 및 종속 서비스에 대한 전체 액세스 권한을 제공합니다.	2018년 9월 20일
ElasticLoadBalancingReadOnly - 새 정책	Elastic Load Balancing 및 종속 서비스에 대한 읽기 전용 액세스 권한을 제공합니다.	2018년 9월 20일
Elastic Load Balancing이 변경 추적을 시작함	Elastic Load Balancing은 AWS 관리형 정책에 대한 변경 사항 추적을 시작했습니다.	2018년 9월 20일

Elastic Load Balancing 규정 준수 검증

AWS 서비스 가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 규정 준수 AWS 서비스 프로그램 범위규정 준수 섹션을 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반 정보는 AWS 규정 준수 프로그램.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 Downloading Reports inDownloading AWS Artifact 참조하세요.

사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 AWS 서비스 결정됩니다.는 규정 준수를 지원하기 위해 다음 리소스를 AWS 제공합니다.

- 보안 규정 준수 및 거버넌스 - 이러한 솔루션 구현 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수 기능을 배포하는 단계를 제공합니다.
- HIPAA 적격 서비스 참조 - HIPAA 적격 서비스가 나열되어 있습니다. 모든가 HIPAA에 적합한 AWS 서비스 것은 아닙니다.
- AWS 규정 준수 리소스 -이 워크북 및 가이드 모음은 업계 및 위치에 적용될 수 있습니다.
- AWS 고객 규정 준수 가이드 - 규정 준수의 관점에서 공동 책임 모델을 이해합니다. 이 가이드에는 여러 프레임워크(미국 국립표준기술연구소(NIST), 결제카드 산업 보안 표준 위원회(PCI), 국제표준화기구(ISO))의 보안 제어에 대한 지침을 보호하고 AWS 서비스 매핑하는 모범 사례가 요약되어 있습니다.
- AWS Config 개발자 안내서의 규칙을 사용하여 리소스 평가 -이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub - 이를 AWS 서비스 통해 내 보안 상태를 포괄적으로 볼 수 있습니다 AWS. Security Hub는 보안 컨트롤을 사용하여 AWS 리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조를 참조하세요.
- Amazon GuardDuty - 의심스러운 악의적인 활동이 있는지 환경을 모니터링하여 사용자, AWS 계정 워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 AWS 서비스 탐지합니다. GuardDuty는 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족하여 PCI DSS와 같은 다양한 규정 준수 요구 사항을 따르는 데 도움을 줄 수 있습니다.
- AWS Audit Manager - 이를 AWS 서비스 통해 AWS 사용량을 지속적으로 감사하여 위험 및 규정 및 업계 표준 준수를 관리하는 방법을 간소화할 수 있습니다.

Elastic Load Balancing의 탄력성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 기반으로 구축됩니다. 리전은 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며, 이러한 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크를 통해 연결되어 있습니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하세요.

AWS 글로벌 인프라 외에도 Elastic Load Balancing은 데이터 복원성을 지원하기 위해 다음과 같은 기능을 제공합니다.

- 단일 가용 영역 또는 여러 가용 영역의 여러 인스턴스 간에 수신 트래픽을 분산합니다.
- Application Load Balancer와 AWS Global Accelerator 함께를 사용하여 하나 이상의 AWS 리전에 있는 여러 로드 밸런서에 수신 트래픽을 분산할 수 있습니다. 자세한 내용은 AWS Global Accelerator 개발자 안내서를 참조하십시오.
- Amazon ECS는 EC2 인스턴스 클러스터에서 Docker 컨테이너를 실행, 중단 및 관리할 수 있게 해줍니다. 로드 밸런서를 사용하여 수신 트래픽을 클러스터의 서비스 간에 분산하도록 Amazon ECS 서비스를 구성할 수 있습니다. 자세한 내용은 Amazon Elastic Container Service 개발자 안내서를 참조하세요.

Elastic Load Balancing의 인프라 보안

관리형 서비스인 Elastic Load Balancing은 AWS 글로벌 네트워크 보안으로 보호됩니다. AWS 보안 서비스 및가 인프라를 AWS 보호하는 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하세요. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 Security Pillar AWS Well-Architected Framework의 인프라 보호를 참조하세요.

AWS 게시된 API 호출을 사용하여 네트워크를 통해 Elastic Load Balancing에 액세스합니다. 고객은 다음을 지원해야 합니다.

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

또한 요청은 액세스 키 ID 및 IAM 위탁자와 관련된 보안 암호 액세스 키를 사용하여 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)을 사용하여 임시 보안 자격 증명을 생성하여 요청에 서명할 수 있습니다.

네트워크 격리

Virtual Private Cloud(VPC)는 AWS 클라우드의 논리적으로 격리된 자체 영역에 있는 가상 네트워크입니다. 서브넷은 VPC의 IP 주소 범위입니다. 로드 밸런서를 생성할 때 로드 밸런서 노드에 대해 하나 이상의 서브넷을 지정할 수 있습니다. VPC의 서브넷에 EC2 인스턴스를 배포하고 로드 밸런서에 등록할 수 있습니다. VPC 및 서브넷에 대한 자세한 내용은 Amazon VPC 사용 설명서를 참조하세요.

VPC에서 로드 밸런서를 생성하면 인터넷에 연결되거나 내부 로드 밸런서가 될 수 있습니다. 내부 로드 밸런서는 로드 밸런서를 위한 VPC에 액세스하여 클라이언트의 요청만 라우팅할 수 있습니다.

로드 밸런서는 프라이빗 IP 주소를 사용하여 등록된 대상으로 요청을 보냅니다. 따라서 대상이 퍼블릭 IP 주소 없이도 로드 밸런서에서 요청을 수신할 수 있습니다.

프라이빗 IP 주소를 사용하여 VPC에서 Elastic Load Balancing API를 호출하려면 AWS PrivateLink을(를) 사용합니다. 자세한 내용은 인터페이스 엔드포인트를 사용하여 Elastic Load Balancing에 액세스 (AWS PrivateLink) 단원을 참조하십시오.

네트워크 트래픽 제어

로드 밸런서를 사용할 때 네트워크 트래픽 보안을 위해 다음 옵션을 고려하십시오.

- 보안 리스너를 사용하여 클라이언트와 로드 밸런서 간의 암호화된 통신을 지원합니다. Application Load Balancer는 HTTPS 리스너를 지원합니다. Network Load Balancer는 TLS 리스너를 지원합니다. Classic Load Balancer는 HTTPS 및 TLS 리스너를 모두 지원합니다. 로드 밸런서에 대해 미리 정의된 보안 정책 중에서 선택하여 애플리케이션에서 지원하는 암호 모음 및 프로토콜 버전을 지정할 수 있습니다. AWS Certificate Manager (ACM) 또는 AWS Identity and Access Management (IAM)을 사용하여 로드 밸런서에 설치된 서버 인증서를 관리할 수 있습니다. SNI(서버 이름 표시) 프로토콜을 사용하여 단일 보안 리스너를 통해 여러 보안 웹 사이트를 제공할 수 있습니다. 둘 이상의 서버 인증서를 보안 리스너와 연결하면 로드 밸런서에 대해 SNI가 자동으로 활성화됩니다.
- 특정 클라이언트의 트래픽만 허용하도록 Application Load Balancer 및 Classic Load Balancer에 대한 보안 그룹을 구성합니다. 이러한 보안 그룹은 리스너 포트에서 클라이언트로부터의 인바운드 트래픽과 클라이언트로의 아웃바운드 트래픽을 허용해야 합니다.
- 로드 밸런서로부터의 트래픽만 허용하도록 Amazon EC2 인스턴스에 대한 보안 그룹을 구성합니다. 이러한 보안 그룹은 리스너 포트와 상태 확인 포트에서 로드 밸런서로부터의 인바운드 트래픽을 허용해야 합니다.
- 자격 증명 공급자를 통해 또는 회사 자격 증명을 사용하여 사용자를 안전하게 인증하도록 Application Load Balancer를 구성합니다. 자세한 내용은 Application Load Balancer를 사용하여 사용자 인증을 참조하세요.
- Application Load Balancer와 함께 AWS WAF를 사용하여 웹 ACL(웹 액세스 제어 목록)의 규칙에 따라 요청을 허용하거나 차단합니다.

인터페이스 엔드포인트를 사용하여 Elastic Load Balancing에 액세스 (AWS PrivateLink)

인터페이스 VPC(Virtual Private Cloud) 엔드포인트를 생성하여 VPC와 Elastic Load Balancing API 간에 프라이빗 연결을 설정할 수 있습니다. 이 연결을 사용하면 VPC에 인터넷 게이트웨이, NAT 인스턴스 또는 VPN 연결을 설정하지 않고 VPC에서 Elastic Load Balancing API를 호출할 수 있습니다. 엔드포인트는 Elastic Load Balancing API 버전 2015-12-01 및 2012-06-01에 안정적이고 확장 가능한 연결을 제공하며 이를 통해 로드 밸런서를 생성하고 관리합니다.

인터페이스 VPC 엔드포인트는 프라이빗 IP 주소를 AWS 서비스 사용하여 애플리케이션과 간의 통신을 가능하게 하는 AWS PrivateLink기능인 로 구동됩니다. 자세한 내용은 AWS PrivateLink 단원을 참조하십시오.

Limit

AWS PrivateLink 는 리스너가 50개 이상인 Network Load Balancer를 지원하지 않습니다.

Elastic Load Balancing을 위한 인터페이스 엔드포인트 생성

다음 서비스 이름을 사용하여 Elastic Load Balancing에 대한 엔드포인트를 생성합니다.

```
com.amazonaws.region.elasticloadbalancing
```

자세한 내용은 AWS PrivateLink 설명서의 인터페이스 엔드포인트 생성을 참조하세요.

Elastic Load Balancing을 위한 VPC 엔드포인트 정책 생성

VPC 엔드포인트에 정책을 연결하여 Elastic Load Balancing API에 대한 액세스를 제어할 수 있습니다. 이 정책은 다음을 지정합니다.

- 작업을 수행할 수 있는 보안 주체.
- 수행할 수 있는 작업.
- 작업을 수행할 수 있는 리소스

다음 예에서는 엔드포인트를 통해 로드 밸런서를 생성할 수 있는 모든 사용자 권한을 거부하는 VPC 엔드포인트 정책을 보여 줍니다. 또한 이 정책 예에서는 모든 사용자에게 다른 모든 작업을 수행할 수 있는 권한을 부여합니다.

```
{
   "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "elasticloadbalancing:CreateLoadBalancer",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

자세한 내용은AWS PrivateLink 가이드의 엔드포인트 정책을 사용하여 서비스에 대한 액세스 제어를 참조하세요.

Elastic Load Balancing API에 대한 요청 제한

Elastic Load Balancing은 리전별로 각 AWS 계정에 대한 API 요청을 제한합니다. 이를 통해 서비스의 성능과 가용성을 높일 수 있습니다. 제한을 사용하면 Elastic Load Balancing API에 대한 요청이 허용되는 최대 API 요청 제한을 초과하지 않습니다. API 요청에는 직접 호출하든 사용자를 대신하여 호출하든(예: AWS Management Console 또는 타사 애플리케이션) 요청 제한이 적용됩니다.

Elastic Load Balancing API 제한 한도를 초과하면 `ThrottlingException` 오류 코드와 `Rate exceeded` 오류 메시지가 표시됩니다.

제한을 정상적으로 처리할 수 있도록 준비하는 것이 좋습니다. 자세한 내용은 시간 제한, 재시도 및 지터를 사용한 백오프를 참조하세요. 제한 수준이 높으면 AWS Support 에 문의하여 API 사용량 및 잠재적 솔루션을 평가할 수 있습니다. 각 사례는 개별적으로 평가됩니다. 지원 는 고가용성과 예측 가능한 성능을 유지하기 위해 시스템의 안전 한도 내에서 한도를 늘릴 수 있습니다.

제한 적용 방법

Elastic Load Balancing은 토큰 버킷 알고리즘을 사용하여 API 제한을 구현합니다. 이 알고리즘을 사용하면 계정에 특정 수의 토큰을 보관하는 버킷이 있습니다. 버킷의 토큰 수는 지정된 초당 제한 한도를 나타냅니다.

Elastic Load Balancing은 두 가지 API 작업 세트를 제공합니다. ELB API 버전 2는 Application Load Balancer, Network Load Balancer 및 Gateway Load Balancer와 같은 유형의 로드 밸런서를 지원합니다. ELB API 버전 1은 Classic Load Balancer를 지원합니다. 각 ELB API 버전에는 자체 버킷과 토큰이 있습니다.

Amazon EC2, Amazon ECS, Amazon EC2 Amazon EC2 Auto Scaling과 같이 사용자를 대신하여 Elastic Load Balancing API를 호출하는 서비스에는 자체 계정 수준 버킷 AWS CloudFormation 이 있습니다. 이러한 서비스는 버킷의 토큰을 사용하지 않습니다.

요청 속도 제한

요청 속도 제한을 사용하면 수행하는 API 요청 수에 제한이 있습니다. 제출한 각 요청은 버킷에서 하나의 토큰을 제거합니다. 예를 들어, 변경되지 않는 API 작업에 대한 토큰 버킷 크기는 토큰 40개입니다. 1초에 최대 40개의 `Describe*` 요청을 수행할 수 있습니다. 1초에 40개의 `Describe*` 요청을 초과하면 제한이 발생하고 해당 초 내의 나머지 요청이 실패합니다.

버킷은 설정된 속도로 자동으로 다시 채워집니다. 버킷이 최대 용량 미만인 경우 버킷이 최대 용량에 도달할 때까지 설정된 수의 토큰이 초당 다시 추가됩니다. 리필 토큰이 도착할 때 버킷이 가득 차면 버킷은 폐기됩니다. 버킷은 최대 토큰 수를 초과할 수 없습니다. 예를 들어, 변경되지 않는 API 작업의 버킷 크기는 토큰 40개이고 리필 속도는 초당 토큰 10개입니다. 1초에 40개의 `DescribeLoadBalancers` 요청을 하면 버킷이 토큰 0(0)개로 줄어듭니다. 최대 용량인 토큰 40개에 도달할 때까지 매초마다 버킷에 리필 토큰 10개를 추가합니다. 즉, 해당 시간 동안 요청이 없는 경우 빈 버킷이 최대 용량에 도달하는 데 4초가 걸립니다.

API 요청을 하기 전에 버킷이 완전히 꽉 찰 때까지 기다릴 필요가 없습니다. 토큰이 버킷에 추가될 때 토큰을 사용할 수 있습니다. 다시 채우기 토큰을 즉시 사용하는 경우 버킷이 최대 용량에 도달하지 않습니다.

모든 Elastic Load Balancing API 작업에서 공유되는 계정 수준 제한 제한이 있습니다. 계정 수준 버킷의 용량은 토큰 40개이고 리필 속도는 초당 요청 토큰 10개입니다.

요청 토큰 버킷 크기 및 리필 속도

요청 속도 제한을 위해 API 작업은 범주로 그룹화됩니다. 각 범주에는 고유한 제한이 있습니다.

Categories

- 변경 작업 - 리소스를 생성, 수정 또는 삭제하는 API 작업입니다. 이 범주에는 일반적으로 변경되지 않는 작업으로 분류되지 않는 모든 API 작업이 포함됩니다. 이러한 작업의 제한 한도는 변경되지 않는 API 작업보다 낮습니다.
- 변경되지 않는 작업 - 리소스에 대한 데이터를 검색하는 API 작업입니다. 이러한 API 작업에는 일반적으로 API 제한 한도가 가장 높습니다.
- 리소스 집약적 작업 - 완료하는 데 가장 많은 시간이 걸리고 가장 많은 리소스를 소비하는 API 작업입니다. 이러한 작업의 제한 한도는 변경 작업보다 훨씬 낮습니다. 이러한 작업은 다른 변경 작업과 별도로 제한됩니다.
- 등록 작업 - 대상을 등록하거나 등록 취소하는 API 작업입니다. 이러한 API 작업은 다른 변경 작업과 별도로 제한됩니다.
- 분류되지 않은 작업 - 이러한 API 작업은 다른 범주 중 하나에 속하더라도 자체 토큰 버킷 크기와 리필 속도를 수신합니다.

다음 표에는 분류된 요청 토큰 버킷의 기본 용량 및 리필 속도가 나와 있습니다.

범주	ELBv2 작업	ELBv1 작업	버킷 용량	리필 속도(초당)
리소스 집약적	CreateLoadBalancer , SetSubnets	CreateLoadBalancer , AttachLoadBalancerToSubnets , DetachLoadBalancerFromSubnets , EnableAvailabilityZonesForLoadBalancer , DisableAvailabilityZonesForLoadBalancer	10	0.2†
등록	RegisterTargets , DeregisterTargets	RegisterInstancesWithLoadBalancer , DeregisterInstancesFromLoadBalancer	20	4
비변환	DescribeAccountLimits , DescribeListenerCertificates , DescribeListeners , DescribeLoadBalancerAttributes , DescribeLoadBalancers , DescribeRules , DescribeSSLPolicies , DescribeTags , DescribeTargetGroupAttributes , DescribeTargetGroups , DescribeTargetHealth	Describe*	40	10

범주	ELBv2 작업	ELBv1 작업	버킷 용량	리필 속도(초당)
변형	AddListenerCertificates , AddTags, CreateListener , CreateRule , CreateTargetGroup , DeleteListener , DeleteLoadBalancer , DeleteRule , DeleteTargetGroup , ModifyListener , ModifyLoadBalancerAttributes , ModifyRule , ModifyTargetGroup , ModifyTargetGroupAttributes , RemoveListenerCertificates , RemoveTags , SetIpAddressType , SetRulePriorities , SetSecurityGroups	AddTags, ApplySecurityGroupsToLoadBalancer , ConfigureHealthCheck , CreateAppCookieStickinessPolicy , CreateLbCookieStickinessPolicy , CreateLoadBalancerListener , CreateLoadBalancerPolicy , Delete*, ModifyLoadBalancerAttributes , RemoveTags , SetLoadBalancer*	20	3

다음 표에는 ELBv2에 대한 분류되지 않은 요청 토큰 버킷의 기본 용량 및 리필 속도가 나와 있습니다.

ELBv2 작업	버킷 용량	리필 속도(초당)
CreateTrustStore	10	0.2†
AddTrustStoreRevocations , DeleteSharedTrustStoreAssociation , DeleteTrustStore , ModifyTrustStore , RemoveTrustStoreRevocations	10	0.2†

ELBv2 작업	버킷 용량	리필 속도(초당)
`GetTrustStoreCaCertificatesBundle` , `GetTrustStoreRevocationContent`	20	4
`DescribeTrustStoreAssociations` , `DescribeTrustStoreRevocations` , `DescribeTrustStores`	40	10

† 전체 토큰 하나를 생성하는 데 몇 초가 걸립니다.

API 요청 모니터링

AWS CloudTrail 를 사용하여 Elastic Load Balancing API 요청을 모니터링할 수 있습니다. 자세한 내용은 를 사용하여 Elastic Load Balancing에 대한 API 호출 로깅 AWS CloudTrail 단원을 참조하십시오.

결제 및 사용 보고서의 Elastic Load Balancing 코드 이해

Elastic Load Balancing을 사용하면 AWS 결제 및 사용 보고서에 관련 코드가 포함됩니다. 이러한 코드를 검토하면 로드 밸런서 비용 및 사용 패턴을 이해하는 데 도움이 됩니다. 비용을 추적하고 관리하는 것은 비용을 최적화하는 데 필수적입니다.

자세한 정보는 Elastic Load Balancing 요금을 참조하세요.

다음 표에서는 결제 및 사용 보고서에 표시되는 Elastic Load Balancing의 코드를 설명합니다. 단위는 시간 또는 로드 밸런서 용량 단위(LCU)입니다. 각 로드 밸런서 유형에는 LCU에 대한 특정 정의가 있습니다. 각 로드 밸런서 유형의 LCUs에 대한 자세한 내용은 Elastic Load Balancing 요금을 참조하세요. 결제 및 사용 보고서에 사용되는 리전 코드 목록은 AWS 리전 결제 코드를 참조하세요.

Application Load Balancers

코드	설명	단위
region-LoadBalancerUsage	실행 시간입니다.	시간
region-LCUUsage	사용된 LCUs.	LCU
region-IdleProvisionedLBCapacity	LCUs는 예약되어 있지만 사용되지 않습니다.	LCU
region-TS-LoadBalancerUsage	상호 TLS에서 트러스트 스토어를 사용하는 시간입니다.	시간
region-Outposts-LoadBalancerUsage	Outposts의 실행 시간입니다.	시간
region-Outposts-LCUUsage	Outposts에 사용되는 LCUs입니다.	LCU
region-ReservedLCUUsage	LCUs.	LCU

Network Load Balancers

코드	설명	단위
region-LoadBalancerUsage	실행 시간입니다.	시간
region-LCUUsage	사용된 LCUs.	LCU

Gateway Load Balancer

코드	설명	단위
region-LoadBalancerUsage	실행 시간입니다.	시간
region-LCUUsage	사용된 LCUs.	LCU

Classic Load Balancer

코드	설명	단위
region-LoadBalancerUsage	실행 시간입니다.	시간
region-DataProcessing-Bytes	처리된 데이터입니다.	GB
region-IdleProvisionedLBCapacity	LCUs는 예약되어 있지만 사용되지 않습니다.	LCU

를 사용하여 Elastic Load Balancing에 대한 API 호출 로깅 AWS CloudTrail

Elastic Load Balancing은 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다. CloudTrail은 Elastic Load Balancing에 대한 모든 API 직접 호출을 이벤트로 캡처합니다. 캡처된 호출에는의 호출 AWS Management Console 과 Elastic Load Balancing API 작업에 대한 코드 호출이 포함됩니다. CloudTrail에서 수집한 정보를 사용하여 Elastic Load Balancing에 수행된 요청, 요청이 수행된 IP 주소, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. 자격 증명을 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트 사용자로 했는지 사용자 보안 인증으로 했는지 여부.
- IAM Identity Center 사용자를 대신하여 요청이 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자에 대한 임시 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화되며 CloudTrail 이벤트 기록에 자동으로 액세스할 수 있습니다. CloudTrail 이벤트 기록은 지난 90일 간 AWS 리전의 관리 이벤트에 대해 보기, 검색 및 다운로드가 가능하고, 수정이 불가능한 레코드를 제공합니다. 자세한 설명은 AWS CloudTrail 사용 설명서의 CloudTrail 이벤트 기록 작업을 참조하세요. Event history(이벤트 기록) 보기는 CloudTrail 요금이 부과되지 않습니다.

AWS 계정 지난 90일 동안의 이벤트를 지속적으로 기록하려면 추적 또는 CloudTrail Lake 이벤트 데이터 스토어를 생성합니다.

CloudTrail 추적

CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 를 사용하여 생성된 모든 추적 AWS Management Console 은 다중 리전입니다. AWS CLI를 사용하여 단일 리전 또는 다중 리전 추적을 생성할 수 있습니다. 계정 AWS 리전 의 모든에서 활동을 캡처하므로 다중 리전 추적을 생성하는 것이 좋습니다. 단일 리전 추적을 생성하는 경우 추적의 AWS 리전에 로깅된 이벤트만 볼 수 있습니다. 추적에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 Creating a trail for your AWS 계정 및 Creating a trail for an organization을 참조하세요.

CloudTrail에서 추적을 생성하여 진행 중인 관리 이벤트의 사본 하나를 Amazon S3 버킷으로 무료로 전송할 수는 있지만, Amazon S3 스토리지 요금이 부과됩니다. CloudTrail 요금에 대한 자세한 내용은 AWS CloudTrail 요금을 참조하세요. Amazon S3 요금에 대한 자세한 내용은 Amazon S3 요금을 참조하세요.

CloudTrail Lake 이벤트 데이터 스토어

CloudTrail Lake를 사용하면 이벤트에 대해 SQL 기반 쿼리를 실행할 수 있습니다. CloudTrail Lake는 행 기반 JSON 형식의 기존 이벤트를 Apache ORC 형식으로 변환합니다. ORC는 빠른 데이터 검색에 최적화된 열 기반 스토리지 형식입니다. 이벤트는 이벤트 데이터 스토어로 집계되며, 이벤트 데이터 스토어는 고급 이벤트 선택기를 적용하여 선택한 기준을 기반으로 하는 변경 불가능한 이벤트 컬렉션입니다. 이벤트 데이터 스토어에 적용하는 선택기는 어떤 이벤트가 지속되고 쿼리할 수 있는지 제어합니다. CloudTrail Lake에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 AWS CloudTrail Lake 작업을 참조하세요.

CloudTrail Lake 이벤트 데이터 스토어 및 쿼리에는 비용이 발생합니다. 이벤트 데이터 스토어를 생성할 때 이벤트 데이터 스토어에 사용할 요금 옵션을 선택합니다. 요금 옵션에 따라 이벤트 모으기 및 저장 비용과 이벤트 데이터 스토어의 기본 및 최대 보존 기간이 결정됩니다. CloudTrail 요금에 대한 자세한 내용은 AWS CloudTrail 요금을 참조하세요.

CloudTrail의 Elastic Load Balancing 관리 이벤트

관리 이벤트는의 리소스에서 수행되는 관리 작업에 대한 정보를 제공합니다 AWS 계정. 이를 컨트롤 플레인 작업이라고도 합니다. 기본적으로 CloudTrail은 관리 이벤트를 로깅합니다.

Elastic Load Balancing은 컨트롤 플레인 작업을 관리 이벤트로 기록합니다. 컨트롤 플레인 작업 목록은 다음을 참조하세요.

- Application Load Balancer - Elastic Load Balancing API 참조 버전 2015-12-01
- Network Load Balancer - Elastic Load Balancing API 참조 버전 2015-12-01
- Gateway Load Balancer - Elastic Load Balancing API 참조 버전 2015-12-01
- Classic Load Balancer - Elastic Load Balancing API 참조 버전 2012-06-01

Elastic Load Balancing 이벤트 예제

이벤트는 모든 소스로부터의 단일 요청을 나타내며 요청된 API 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 추적이 아니므로 이벤트가 특정 순서로 표시되지 않습니다.

다음 예제에서는 로드 밸런서를 생성한 다음 AWS CLI를 사용하여 삭제한 사용자의 CloudTrail 이벤트를 보여줍니다. `userAgent` 요소를 사용해 CLI를 식별할 수 있습니다. `eventName` 요소를 사용해 요청된 API 호출을 식별할 수 있습니다. 그리고 사용자(`Alice`)에 대한 정보는 `userIdentity` 요소를 보면 알 수 있습니다.

Example 예제 1: ELBv2 API의 CreateLoadBalancer

```
{
    "eventVersion": "1.03",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2016-04-01T15:31:48Z",
    "eventSource": "elasticloadbalancing.amazonaws.com",
    "eventName": "CreateLoadBalancer",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
    "requestParameters": {
        "subnets": ["subnet-8360a9e7","subnet-b7d581c0"],
        "securityGroups": ["sg-5943793c"],
        "name": "my-load-balancer",
        "scheme": "internet-facing"
    },
    "responseElements": {
        "loadBalancers":[{
            "type": "application",
            "loadBalancerName": "my-load-balancer",
            "vpcId": "vpc-3ac0fb5f",
            "securityGroups": ["sg-5943793c"],
            "state": {"code":"provisioning"},
            "availabilityZones": [
               {"subnetId":"subnet-8360a9e7","zoneName":"us-west-2a"},
               {"subnetId":"subnet-b7d581c0","zoneName":"us-west-2b"}
            ],
            "dNSName": "my-load-balancer-1836718677.us-west-2.elb.amazonaws.com",
            "canonicalHostedZoneId": "Z2P70J7HTTTPLU",
            "createdTime": "Apr 11, 2016 5:23:50 PM",
```

```
            "loadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/ffcddace1759e1d0",
            "scheme": "internet-facing"
        }]
    },
    "requestID": "b9960276-b9b2-11e3-8a13-f1ef1EXAMPLE",
    "eventID": "6f4ab5bd-2daa-4d00-be14-d92efEXAMPLE",
    "eventType": "AwsApiCall",
    "apiVersion": "2015-12-01",
    "recipientAccountId": "123456789012"
}
```

Example 예제 2: ELBv2 API의 DeleteLoadBalancer

```
{
    "eventVersion": "1.03",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2016-04-01T15:31:48Z",
    "eventSource": "elasticloadbalancing.amazonaws.com",
    "eventName": "DeleteLoadBalancer",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
    "requestParameters": {
        "loadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/ffcddace1759e1d0"
    },
    "responseElements": null,
    "requestID": "349598b3-000e-11e6-a82b-298133eEXAMPLE",
    "eventID": "75e81c95-4012-421f-a0cf-babdaEXAMPLE",
    "eventType": "AwsApiCall",
    "apiVersion": "2015-12-01",
    "recipientAccountId": "123456789012"
}
```

Example 예제 3: ELB API의 CreateLoadBalancer

```
{
    "eventVersion": "1.03",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAJDPLRKLG7UEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2016-04-01T15:31:48Z",
    "eventSource": "elasticloadbalancing.amazonaws.com",
    "eventName": "CreateLoadBalancer",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
    "requestParameters": {
        "subnets": ["subnet-12345678","subnet-76543210"],
        "loadBalancerName": "my-load-balancer",
        "listeners": [{
           "protocol": "HTTP",
           "loadBalancerPort": 80,
           "instanceProtocol": "HTTP",
           "instancePort": 80
        }]
    },
    "responseElements": {
        "dNSName": "my-loadbalancer-1234567890.elb.amazonaws.com"
    },
    "requestID": "b9960276-b9b2-11e3-8a13-f1ef1EXAMPLE",
    "eventID": "6f4ab5bd-2daa-4d00-be14-d92efEXAMPLE",
    "eventType": "AwsApiCall",
    "apiVersion": "2012-06-01",
    "recipientAccountId": "123456789012"
}
```

Example 예제 4: ELB API의 DeleteLoadBalancer

```
{
    "eventVersion": "1.03",
    "userIdentity": {
```

```
        "type": "IAMUser",
        "principalId": "AIDAJDPLRKLG7UEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2016-04-08T12:39:25Z",
    "eventSource": "elasticloadbalancing.amazonaws.com",
    "eventName": "DeleteLoadBalancer",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
    "requestParameters": {
        "loadBalancerName": "my-load-balancer"
    },
    "responseElements": null,
    "requestID": "f0f17bb6-b9ba-11e3-9b20-999fdEXAMPLE",
    "eventID": "4f99f0e8-5cf8-4c30-b6da-3b69fEXAMPLE"
    "eventType": "AwsApiCall",
    "apiVersion": "2012-06-01",
    "recipientAccountId": "123456789012"
}
```


CloudTrail 레코드 콘텐츠에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail record contents를 참조하세요.

Classic Load Balancer 마이그레이션

Elastic Load Balancing은 다음 유형의 로드 밸런서를 지원합니다. Application Load Balancers, Network Load Balancers, 게이트웨이이 로드 밸런서 및 Classic Load Balancer 각 로드 밸런서 유형의 다양한 기능에 대한 자세한 내용은 Elastic Load Balancing 기능을 참조하세요.

VPC의 기존 Classic Load Balancer를 Application Load Balancer 또는 Network Load Balancer로 마이그레이션하도록 선택할 수도 있습니다.

Classic Load Balancer에서 마이그레이션할 때의 이점

각 유형의 로드 밸런서에는 고유한 기능, 함수 및 구성이 있습니다. 가장 적합한 로드 밸런서를 결정하는 데 도움이 되도록 각 로드 밸런서의 이점을 검토합니다.

Application Load Balancer

Classic Load Balancer 대신 Application Load Balancer를 사용하면 다음과 같은 이점이 있습니다.

지원:

- 경로 조건, 호스트 조건 및 HTTP 헤더 조건.
- 요청을 한 URL에서 다른 URL로 리디렉션하고 요청을 단일 EC2 인스턴스의 여러 애플리케이션으로 라우팅.
- 사용자 지정 HTTP 응답을 반환.
- IP 주소별로 대상을 등록하고 Lambda 함수를 대상으로 등록. 로드 밸런서의 VPC 외부 대상을 포함.
- 기업 또는 소셜 자격 증명을 통한 사용자 인증.
- Amazon Elastic Container Service(Amazon ECS) 컨테이너식 애플리케이션.
- 각 서비스의 상태를 독립적으로 모니터링.

액세스 로그는 추가 정보를 포함하며 압축된 형식으로 저장됩니다.

전반적으로 개선된 로드 밸런서 성능.

Network Load Balancer

Classic Load Balancer 대신 Network Load Balancer를 사용하면 다음과 같은 이점이 있습니다.

지원:

- 로드 밸런서에 대해 활성화된 서브넷당 하나의 탄력적 IP 주소를 할당할 수 있는 고정 IP 주소.
- 로드 밸런서의 VPC 외부 대상을 포함하여 IP 주소로 대상을 등록.
- 단일 EC2 인스턴스의 여러 애플리케이션으로 요청을 라우팅.
- Amazon Elastic Container Service(Amazon ECS) 컨테이너식 애플리케이션.
- 각 서비스의 상태를 독립적으로 모니터링.

일시적 워크로드를 처리하고 초당 수백만 개의 요청으로 확장할 수 있습니다.

마이그레이션 마법사를 사용하여 마이그레이션

마이그레이션 마법사는 사용자의 Classic Load Balancer 구성을 사용하여 동등한 Application Load Balancer 또는 Network Load Balancer를 생성합니다. 다른 방법에 비해 Classic Load Balancer를 마이그레이션하는 데 필요한 시간과 노력이 줄어듭니다.

> **Note**
>
> 마법사는 새 로드 밸런서를 생성합니다. 기존 Classic Load Balancer를 Application Load Balancer 또는 Network Load Balancer로 변환하지 않습니다. 트래픽을 새로 생성된 로드 밸런서로 수동으로 리디렉션해야 합니다.

제한 사항

- 새 로드 밸런서의 이름은 동일한 리전에서 동일한 유형의 기존 로드 밸런서와 같을 수 없습니다.
- Classic Load Balancer에 키에 `aws:` 접두사가 포함된 태그가 있는 경우 해당 태그는 마이그레이션되지 않습니다.

Application Load Balancer로 마이그레이션하는 경우

- Classic Load Balancer에 서브넷이 하나만 있는 경우 두 번째 서브넷을 지정해야 합니다.
- Classic Load Balancer에 TCP 상태 확인을 사용하는 HTTP/HTTPS 리스너가 있는 경우 상태 확인 프로토콜이 HTTP로 업데이트되고 경로가 '/'로 설정됩니다.
- Classic Load Balancer에 사용자 지정 또는 지원되지 않는 보안 정책을 사용하는 HTTPS 리스너가 있는 경우 마이그레이션 마법사는 새 로드 밸런서 유형에 기본 보안 정책을 사용합니다.

Network Load Balancer로 마이그레이션하는 경우

- 다음 인스턴스 유형은 새 대상 그룹에 등록되지 않습니다. C1, CC1, CC2, CG1, CG2, CR1, CS1, G1, G2, HI1, HS1, M1, M2, M3, T1
- Classic Load Balancer의 특정 상태 확인 설정은 새 대상 그룹으로 이전할 수 없습니다. 이러한 경우는 마이그레이션 마법사의 요약 섹션에 변경 사항으로 표시됩니다.
- Classic Load Balancer에 SSL 리스너가 있는 경우 마이그레이션 마법사는 SSL 리스너의 인증서 및 보안 정책을 사용하여 TLS 리스너를 생성합니다.

마이그레이션 마법사 프로세스

마이그레이션 마법사를 사용하여 Classic Load Balancer를 마이그레이션하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 Load Balancing 아래에서 로드 밸런서를 선택합니다.
3. 마이그레이션하려는 Classic Load Balancer를 선택합니다.
4. 로드 밸런서 세부 정보 섹션에서 마이그레이션 마법사 시작을 선택합니다.
5. Application Load Balancer로 마이그레이션 또는 Network Load Balancer로 마이그레이션을 선택하여 마이그레이션 마법사를 엽니다.
6. 새 로드 밸런서 이름 지정 아래에서 로드 밸런서 이름에 새 로드 밸런서의 이름을 입력합니다.
7. 새 대상 그룹 이름 지정 및 대상 검토에서 대상 그룹 이름에 새 대상 그룹의 이름을 입력합니다.
8. (선택 사항) 대상에서 새 대상 그룹에 등록할 대상 인스턴스를 검토할 수 있습니다.
9. (선택 사항) 태그 검토에서 새 로드 밸런서에 적용될 태그를 검토할 수 있습니다.
10. Application Load Balancer 요약 또는 Network Load Balancer 요약에서 마이그레이션 마법사가 할당한 구성 옵션을 검토하고 확인합니다.
11. 구성 요약이 만족스러우면 Application Load Balancer 생성 또는 Network Load Balancer 생성을 선택하여 마이그레이션을 시작합니다.

로드 밸런서 복사 유틸리티를 사용하여 마이그레이션

로드 밸런서 복사 유틸리티는 AWS GitHub 페이지의 Elastic Load Balancing Tools 리포지토리 내에서 사용할 수 있습니다.

리소스

- Elastic Load Balancing 도구
- Classic Load Balancer에서 Application Load Balancer로 복사 유틸리티
- Classic Load Balancer에서 Network Load Balancer로 복사 유틸리티

로드 밸런서를 수동으로 마이그레이션

다음 정보는 VPC의 기존 Classic Load Balancer를 기반으로 새 Application Load Balancer 또는 Network Load Balancer를 수동으로 생성하기 위한 일반적인 지침을 제공합니다. AWS Management Console, AWS CLI또는 AWS SDK를 사용하여 마이그레이션할 수 있습니다. 자세한 내용은 Elastic Load Balancing 시작하기 단원을 참조하십시오.

마이그레이션 프로세스를 완료한 후 새 로드 밸런서의 기능을 활용할 수 있습니다.

수동 마이그레이션 프로세스

1단계: 새 로드 밸런서 생성

마이그레이션할 Classic Load Balancer와 동등한 구성으로 로드 밸런서를 생성합니다.

1. Classic Load Balancer와 동일한 체계(인터넷 경계 또는 내부), 서브넷 및 보안 그룹으로 새 로드 밸런서를 생성합니다.
2. Classic Load Balancer와 동일한 상태 확인 설정으로 로드 밸런서에 대한 하나의 대상 그룹을 생성합니다.
3. 다음 중 하나를 수행하십시오.
   - Classic Load Balancer가 Auto Scaling 그룹에 연결된 경우, 대상 그룹을 Auto Scaling 그룹에 연결합니다. 이렇게 하면 Auto Scaling 인스턴스가 대상 그룹에도 등록됩니다.
   - EC2 인스턴스를 대상 그룹에 등록합니다.
4. 요청을 대상 그룹에 전달하는 기본 규칙이 있는 하나 이상의 리스너를 생성합니다. HTTPS 리스너를 생성하는 경우 Classic Load Balancer에 대해 지정한 것과 동일한 인증서를 지정할 수 있습니다. 기본 보안 정책을 사용하는 것이 좋습니다.
5. Classic Load Balancer에 태그가 있는 경우 해당 태그를 검토하고 새 로드 밸런서에 관련 태그를 추가합니다.

2단계: 새 로드 밸런서로 점진적으로 트래픽 리디렉션

인스턴스를 새 로드 밸런서에 등록한 후에는 이전 로드 밸런서에서 새 로드 밸런서로 트래픽 리디렉션 프로세스를 시작할 수 있습니다. 이를 통해 애플리케이션 가용성에 미치는 위험을 최소화하면서 새 로드 밸런서를 테스트할 수 있습니다.

새 로드 밸런서에 트래픽을 점진적으로 리디렉션하려면

1. 새 로드 밸런서의 DNS 이름을 인터넷에 연결된 웹 브라우저의 주소 필드에 붙여 넣습니다. 모든 것이 잘 작동하는 경우 브라우저에 애플리케이션 기본 페이지가 표시됩니다.
2. 도메인 이름을 새 로드 밸런서와 연결하는 새 DNS 레코드를 만듭니다. DNS 서비스가 가중을 지원하는 경우 새 DNS 레코드에서 가중치를 1로 지정하고 이전 로드 밸런서에 대한 기존 DNS 레코드에서 가중치를 9로 지정합니다. 이렇게 하면 새 로드 밸런서에 트래픽의 10%, 이전 로드 밸런서에 트래픽의 90%가 전송됩니다.
3. 새 로드 밸런서를 모니터링하여 인스턴스에 대한 트래픽 및 라우팅 요청을 수신하는지 확인합니다.

   > **Important**
   >
   > DNS 레코드의 TTL(time-to-live)은 60초입니다. 즉, 도메인 이름을 확인하는 DNS 서버는 해당 레코드 정보를 60초 동안 캐시에 보관합니다. 그동안 변경 내용이 전파됩니다. 따라서 이러한 DNS 서버는 이전 단계를 완료한 후 최대 60초 동안 이전 로드 밸런서에 트래픽을 계속 라우팅할 수 있습니다. 전파되는 동안 트래픽은 로드 밸런서에 전송될 수 있습니다.

4. 모든 트래픽이 새 로드 밸런서로 전달될 때까지 계속 DNS 레코드의 가중치를 업데이트합니다. 완료되면 이전 로드 밸런서에 대한 DNS 레코드를 삭제할 수 있습니다.

3단계: 정책, 스크립트 및 코드 업데이트

Classic Load Balancer에서 Application Load Balancer 또는 Network Load Balancer로 마이그레이션하는 경우 다음을 수행해야 합니다.

- API 버전 2012-06-01을 사용하는 IAM 정책을 버전 2015-12-01을 사용하도록 업데이트합니다.
- `AWS/ELB` 네임스페이스의 CloudWatch 지표를 사용하는 프로세스를 `AWS/ApplicationELB` 또는 `AWS/NetworkELB` 네임스페이스의 지표를 사용하도록 업데이트합니다.
- aws elb AWS CLI 명령을 사용하여 aws elbv2 AWS CLI 명령을 사용하는 스크립트를 업데이트합니다.

- `AWS::ElasticLoadBalancing::LoadBalancer` 리소스를 사용하도록 `AWS::ElasticLoadBalancingV2` 리소스를 사용하는 AWS CloudFormation 템플릿을 업데이트합니다.
- Elastic Load Balancing API 버전 2012-06-01을 사용하는 코드를 버전 2015-12-01을 사용하도록 업데이트합니다.

리소스

- AWS CLI 명령 참조의 elbv2
- Elastic Load Balancing API 참조 버전 2015-12-01
- Elastic Load Balancing의 ID 및 액세스 관리
- Application Load Balancer 사용 설명서의 Application Load Balancer 지표
- Network Load Balancer 사용 설명서의 Network Load Balancer 지표
- AWS CloudFormation 사용 설명서의 AWS::ElasticLoadBalancingV2::LoadBalancer

4단계: 이전 로드 밸런서 삭제

다음을 수행하고 나면 이전 Classic Load Balancer를 삭제할 수 있습니다.

- 모든 트래픽을 이전 로드 밸런서에서 새 로드 밸런서로 리디렉션
- 이전 로드 밸런서로 라우팅된 모든 기존 요청을 완료

사용자가 Classic Load Balancer를 생성하지 못하도록 방지

사용자가 계정에서 Classic Load Balancer를 생성하지 못하도록 하는 IAM 정책을 생성할 수 있습니다.

Elastic Load Balancing V2 및 Elastic Load Balancing V1 APIs는 모두 `CreateLoadBalancer` API 작업을 제공합니다. Classic Load Balancer를 생성할 때 로드 밸런서와 리스너를 모두 생성하는 V1 API 작업을 사용합니다. Application Load Balancer, Network Load Balancer 또는 Gateway Load Balancer를 생성할 때 로드 밸런서만 생성하는 V2 API 작업을 사용합니다. V2 API는 로드 밸런서를 생성한 후 리스너를 생성하는 데 사용하는 `CreateListener` 작업을 제공합니다.

다음 정책은 리스너 프로토콜이 지정된 경우 로드 밸런서를 생성할 수 있는 사용자 권한을 거부합니다. Classic Load Balancer를 생성할 때 하나 이상의 리스너를 구성해야 하므로이 정책은 사용자가 Classic Load Balancer를 생성할 수 없도록 합니다. 로드 밸런서와 리스너를 생성하기 위한 별도의 API 작업이 있기 때문에 사용자가 다른 유형의 로드 밸런서를 생성하는 것을 방지하지 않습니다.

```
{
    "Version": "2012-10-17",
    "Effect": "Deny",
    "Action": "elasticloadbalancing:CreateLoadBalancer",
    "Resource": [
        "arn:aws:elasticloadbalancing:*:*:loadbalancer/*"
    ],
    "Condition": {
        "Null": {
            "elasticloadbalancing:ListenerProtocol": false
        }
    }
}
```

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