Elastic Load Balancing

사용 설명서

Elastic Load Balancing: 사용 설명서

Table of Contents

Elastic Load Balancing란 무엇인가요? 1
로드 밸런서 이점 1
Elastic Load Balancing의 기능 1
Elastic Load Balancing에 액세스 1
관련 서비스 2
요금 2
Elastic Load Balancing 작동 방식 3
가용 영역 및 로드 밸런서 노드 3
교차 영역 로드 밸런싱 3
라우팅 요청 4
라우팅 알고리즘 5
HTTP 연결 5
HTTP 헤더 6
HTTP 헤더 제한 6
로드 밸런서 체계 6
시작하기 8
Application Load Balancer 생성 8
Network Load Balancer 만들기 8
Gateway Load Balancer 생성 8
Classic Load Balancer 만들기 8
보안 9
데이터 보호 9
저장된 암호화 10
전송 중 데이터 암호화 10
ID 및 액세스 관리 10
IAM 정책을 사용하여 권한 부여 11
Elastic Load Balancing에 대한 API 작업 11
Elastic Load Balancing 리소스 12
Elastic Load Balancing의 리소스 수준 권한 13
Elastic Load Balancing의 조건 키 15
사전 정의된 AWS 관리형 정책 16
API 권한 16
서비스 연결 역할 18
Compliance validation 20
Resilience 20
인프라 보안 21
네트워크 격리 21
네트워크 트래픽 제어 21
인터페이스 VPC 엔드포인트 22
Elastic Load Balancing용 인터페이스 엔드포인트 생성 22
Elastic Load Balancing에 대한 VPC 엔드포인트 정책 생성 22
Classic Load Balancer 마이그레이션 24
1단계: 새 로드 밸런서 생성 24
옵션 1: 콘솔에서 마이그레이션 마법사 사용 24
옵션 2: github의 로드 밸런서 복사 유틸리티를 사용 25
옵션 3: 수동으로 Application Load Balancer 또는 Network Load Balancer로 마이그레이션 25
옵션 4: 수동으로 VPC의 Classic Load Balancer로 마이그레이션 26
2단계: 새 로드 밸런서로 점진적으로 트래픽 리디렉션 26
3단계: 정책, 스크립트 및 코드 업데이트 27
4단계: 이전 로드 밸런서 삭제 27
..... xxviii

Elastic Load Balancing란 무엇인가요?

Elastic Load Balancing은(는) 둘 이상의 가용 영역에서 EC2 인스턴스, 컨테이너, IP 주소 등 여러 대상에 걸쳐 수신되는 트래픽을 자동으로 분산합니다. 이는 등록된 대상의 상태를 모니터링하여 정상 상태인 대상으로만 트래픽을 라우팅합니다. 시간이 흐르면서 수신되는 트래픽이 변화함에 따라 Elastic Load Balancing이(가) 로드 밸런서를 확장합니다. 대다수의 워크로드에 맞게 자동으로 조정할 수 있습니다.

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 기능

Elastic Load Balancing는 Application Load Balancer, Network Load Balancer, Gateway Load Balancers 및 Classic Load Balancer 로드 밸런서를 지원합니다. 각자 필요에 따라 가장 적합한 로드 밸런서 유형을 선택할 수 있습니다. 자세한 내용은 제품 비교를 참조하세요.

각 로드 밸런서 사용에 대한 자세한 내용은 Application Load Balancer 사용 설명서, Network Load Balancer 사용 설명서, Gateway Load Balancer 사용 설명서 및 Classic Load Balancer 사용 설명서을(를) 참조하세요.

Elastic Load Balancing에 액세스

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

- AWS Management 콘솔— Elastic Load Balancing에 액세스할 때 사용할 수 있는 웹 인터페이스를 제공합니다.
- AWS 명령줄 인터페이스(AWS CLI) — Elastic Load Balancing를 포함한 다양한 AWS 서비스 명령을 제공합니다. AWS CLI는 Windows, macOS, Linux에서 지원됩니다. 자세한 내용은 AWS Command Line Interface 단원을 참조하십시오.
- AWS SDK — 언어별 API를 제공하고, 서명 계산, 요청 재시도 처리 및 오류 처리와 같은 많은 연결 세부 정보를 관리합니다. 자세한 정보는 AWS SDK를 참조하십시오.
- 쿼리 API— HTTPS 요청을 사용하여 호출하는 하위 수준의 API 작업을 제공합니다. 쿼리 API 사용은 Elastic Load Balancing에 액세스할 수 있는 가장 직접적인 방법입니다. 하지만 쿼리 API를 사용하려면 애플리케이션에서 요청에 서명할 해시 생성 및 오류 처리와 같은 하위 수준의 세부 정보를 처리해야 합니다. 자세한 내용은 다음을 참조하십시오.
  - Application Load Balancer 및 Network 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 실행하는 가상 서버입니다. 로드 밸런서를 구성하여 EC2 인스턴스에 트래픽을 라우팅할 수 있습니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서 또는 Windows 인스턴스용 Amazon EC2 사용 설명서 섹션을 참조하십시오.
- Amazon EC2 Auto Scaling — 인스턴스에 장애가 발생하더라도 원하는 수의 인스턴스를 실행하고 인스턴스의 수요가 변경되면 Amazon EC2 Auto Scaling에서 자동으로 인스턴스 수를 늘리거나 줄일 수 있게 해 줍니다. Elastic Load Balancing에서 Auto Scaling을 활성화한 경우 Auto Scaling에서 시작되는 인스턴스가 로드 밸런서에 자동으로 등록됩니다. 마찬가지로 Auto Scaling에 의해 종료된 인스턴스는 로드 밸런서에서 자동으로 등록 취소됩니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서 단원을 참조하십시오.
- 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 해 줍니다. 자세한 내용은 Amazon CloudWatch 사용 설명서 단원을 참조하십시오.
- Amazon ECS — EC2 인스턴스 클러스터에서 Docker 컨테이너를 실행, 중단 및 관리 가능합니다. 로드 밸런서를 구성하여 컨테이너에 트래픽을 라우팅할 수 있습니다. 자세한 내용은 Amazon Elastic Container Service Developer Guide 단원을 참조하십시오.
- AWS Global Accelerator — 애플리케이션의 가용성과 성능을 향상시킵니다. 액셀러레이터를 사용하여 하나 이상의 AWS 리전에 있는 여러 로드 밸런서에 트래픽을 분산합니다. 자세한 내용은 AWS Global Accelerator 개발자 안내서 단원을 참조하십시오.
- Route 53 — 도메인 이름을 컴퓨터를 사용하여 서로 연결해주는 숫자로 된 IP 주소로 변환하여 방문자를 안정적이며 비용 효율적으로 웹 사이트로 라우팅하도록 합니다. 예를 들어 `www.example.com`을 숫자 IP 주소 `192.0.2.1`로 변환합니다. AWS에서는 리소스에 URL을 지정합니다(예: 로드 밸런서). 그러나 기억하기 쉬운 URL이 필요한 경우도 있습니다. 예를 들어 도메인 이름을 로드 밸런서로 매핑할 수 있습니다. 자세한 내용은 Amazon Route 53 개발자 안내서 단원을 참조하십시오.
- AWS WAF — Application Load Balancer와 함께 AWS WAF를 사용하여 웹 ACL(웹 액세스 제어 목록)의 규칙에 따라 요청을 허용하거나 차단할 수 있습니다. 자세한 내용은 AWS WAF 개발자 안내서 단원을 참조하십시오.

요금

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

Elastic Load Balancing 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.

Elastic Load Balancing은(는) 다음 유형의 로드 밸런서를 지원합니다.

- Application Load Balancer
- Network Load Balancer
- Gateway Load Balancer
- Classic Load Balancer

로드 밸런서 유형이 구성되는 방법에는 주요 차이점이 있습니다. Application Load Balancer, Network Load Balancer 및 Gateway Load Balancer를 사용하여 대상을 대상 그룹에 등록하고 트래픽을 대상 그룹에 라우팅합니다. Classic Load Balancer에서는 로드 밸런서에 인스턴스를 등록합니다.

가용 영역 및 로드 밸런서 노드

로드 밸런서의 가용 영역을 활성화하면 Elastic Load Balancing가 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 Balancer을(를) 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.

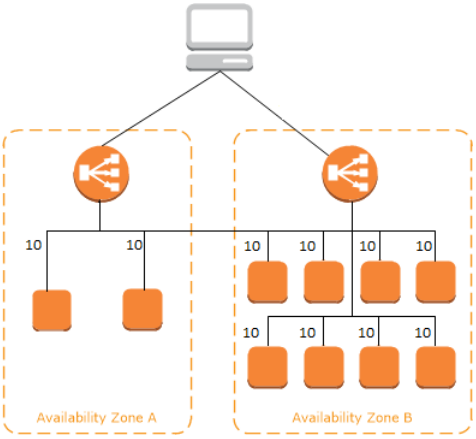
가용 영역을 비활성화해도 해당 가용 영역의 대상은 로드 밸런서에 등록된 상태로 남아 있습니다. 대상은 등록된 상태로 유지되지만 로드 밸런서는 트래픽을 해당 대상으로 라우팅하지 않습니다.

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 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. 이렇게 하면 각 로드 밸런서 노드가 클라이언트가 보내는 트래픽의 50%를 수신하도록 트래픽이 분산됩니다. 각 로드 밸런서 노드는 공유 트래픽을 해당 범위의 등록된 대상에만 분산합니다.

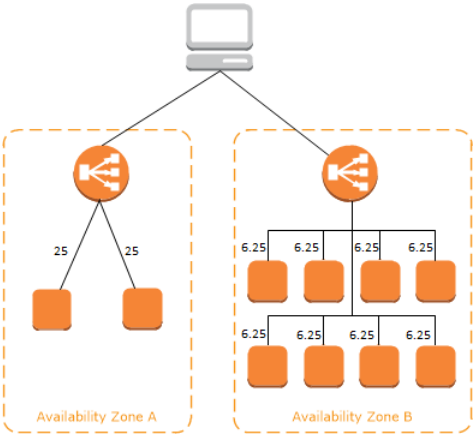
교차 영역 로드 밸런싱이 활성화된 경우 각 10개의 대상이 트래픽의 10%를 수신합니다. 이는 각 로드 밸런서 노드가 클라이언트 트래픽의 50%를 10개의 대상 모두에게 라우팅할 수 있기 때문입니다.



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 Balancer에서는 교차 영역 로드 밸런싱이 항상 활성화되어 있습니다.

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.

Classic Load Balancer를 생성할 경우 교차 영역 로드 밸런싱에 대한 기본값은 로드 밸런서 생성 방법에 따라 달라집니다. API 또는 CLI에서는 교차 영역 로드 밸런싱이 기본적으로 비활성화되어 있습니다. AWS Management 콘솔에서는 교차 영역 로드 밸런싱을 활성화하는 옵션이 기본적으로 선택됩니다. Classic Load Balancer를 생성하면 언제든지 교차 영역 로드 밸런싱을 활성화 또는 비활성화할 수 있습니다. 자세한 내용은 Classic Load Balancer 사용 설명서에서 교차 영역 로드 밸런싱 활성화를 참조하십시오.

라우팅 요청

클라이언트는 로드 밸런서에 요청을 보내기 전에 로드 밸런서는 DNS(도메인 이름 시스템) 서버를 사용하여 로드 밸런서의 도메인 이름을 해석합니다. 로드 밸런서가 `amazonaws.com` 도메인에 있기 때문에 DNS 항목

은 Amazon에서 제어합니다. Amazon DNS 서버는 클라이언트에 하나 이상의 IP 주소를 반환합니다. 이 주소는 로드 밸런서에 대한 로드 밸런서 노드의 IP 주소입니다. Elastic Load Balancing은 Network Load Balancer를 사용하여 사용자가 활성화하는 각 가용 영역에 대해 네트워크 인터페이스를 생성합니다. 가용 영역의 각 로드 밸런서 노드는 이 네트워크 인터페이스를 사용하여 고정 IP 주소를 가져옵니다. 로드 밸런서를 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.

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.

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 HTTP응답에 `Connection: close` 헤더를 설정하여 HTTP `keep-alives`를 비활성화합니다.

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

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. Application Load Balancer도 HTTP에서 웹 소켓까지 연결 업그레이

드를 지원합니다. 그러나 연결 업그레이드가 있는 경우, Application Load Balancer 리스너 라우팅 규칙 및 AWS WAF 통합은 더 이상 적용되지 않습니다.

Application Load Balancer은(는) 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`은(는) 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. 이는 백엔드 연결(로드 밸런서를 등록된 대상에)에서 HTTP/1.1을 사용합니다. `Keep-alive`은(는) 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 Balancer 및 Classic Load Balancer는 X-Forwarded-For, X-Forwarded-Proto, X-Forwarded-Port 헤더를 요청에 자동으로 추가합니다.

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.

Application Load Balancer 및 Classic Load Balancer에 Expect 헤더가 수신되면 콘텐츠 길이 헤더를 테스트하지 않고 HTTP 100 Continue를 사용하여 클라이언트에 즉시 응답하고 Expect 헤더를 제거한 다음 요청을 라우팅합니다.

HTTP 헤더 제한

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

HTTP/1.x 헤더

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

HTTP/2 헤더

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

로드 밸런서 체계

로드 밸런서를 생성할 때 로드 밸런서를 내부 로드 밸런서 또는 인터넷 경계 로드 밸런서로 생성할지 여부를 선택해야 합니다. EC2-Classic에서 Classic Load Balancer를 생성할 때는 반드시 인터넷 경계 로드 밸런서여야 한다는 점을 참고하십시오.

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.

Elastic Load Balancing 시작하기

Elastic Load Balancing은(는) Application Load Balancer, Network Load Balancer, Gateway Load Balancers 및 Classic Load Balancer 로드 밸런서를 지원합니다. 각자 필요에 따라 가장 적합한 로드 밸런서 유형을 선택할 수 있습니다. 자세한 내용은 제품 비교를 참조하세요.

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

기존 Classic Load Balancer가 있을 경우 Application Load Balancer 또는 Network Load Balancer로 마이그레이션할 수 있습니다. 자세한 내용은 Classic Load Balancer 마이그레이션 (p. 24) 단원을 참조하십시오.

내용

- Application Load Balancer 생성 (p. 8)
- Network Load Balancer 만들기 (p. 8)
- Gateway Load Balancer 생성 (p. 8)
- Classic Load Balancer 만들기 (p. 8)

Application Load Balancer 생성

AWS Management 콘솔을 사용하여 Application Load Balancer를 생성하려면 Application Load Balancer 사용 설명서에서 Application Load Balancer 시작하기를 참조하십시오.

AWS CLI를 사용하여 Application Load Balancer를 생성하려면 Application Load Balancer 사용 설명서에서 AWS CLI를 사용하여 Application Load Balancer 생성을 참조하십시오.

Network Load Balancer 만들기

AWS Management 콘솔을 사용하여 Network Load Balancer를 생성하려면 Network Load Balancer 사용 설명서에서 Network Load Balancer 시작하기를 참조하십시오.

AWS CLI를 사용하여 Network Load Balancer를 생성하려면 Network Load Balancer 사용 설명서에서 AWS CLI를 사용하여 Network Load Balancer 생성을 참조하십시오.

Gateway Load Balancer 생성

AWS Management 콘솔을(를) 사용하여 Gateway Load Balancer를 생성하려면 Gateway Load Balancer 사용 설명서의 Gateway Load Balancer 시작하기를 참조하세요.

AWS CLI을(를) 사용하여 Gateway Load Balancer를 생성하려면 Gateway Load Balancer 사용 설명서의 AWS CLI을(를) 사용하여 Gateway Load Balancer 시작하기를 참조하세요.

Classic Load Balancer 만들기

AWS Management 콘솔을(를) 사용하여 Classic Load Balancer을(를) 생성하려면 Classic Load Balancer 사용 설명서에서 Classic Load Balancer 생성을 참조하십시오.

Elastic Load Balancing의 보안

AWS에서 클라우드 보안을 가장 중요하게 생각합니다. AWS 고객은 보안에 가장 보안에 민감한 조직의 요구 사항에 부합하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 AWS와 귀하의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드의 보안 – AWS는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호합니다. AWS는 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램의 일환으로 보안 효과를 정기적으로 테스트하고 검증합니다. Elastic Load Balancing에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램의 범위에 속하는 AWS 서비스를 참조하십시오.
- 클라우드 내 보안 – 귀하의 책임은 귀하가 사용하는 AWS 서비스에 의해 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 목적에 맞게 Elastic Load Balancing을 구성하는 방법을 보여줍니다. 또한 Elastic Load Balancing 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 배웁니다.

Gateway Load Balancer를 사용할 경우 어플라이언스 공급업체의 소프트웨어를 선택하고 검증할 책임은 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 나열된 어플라이언스 공급업체는 어플라이언스 소프트웨어를 AWS와(과) 통합하고 검증했습니다. 이 목록에 있는 공급업체의 어플라이언스 소프트웨어에 대한 신뢰도를 높일 수 있습니다. 그러나 AWS은(는) 이러한 공급업체에서 제공하는 소프트웨어의 보안 또는 안정성을 보장하지 않습니다.

목차

- Elastic Load Balancing의 데이터 보호 (p. 9)
- Elastic Load Balancing의 ID 및 액세스 관리 (p. 10)
- Compliance validation for Elastic Load Balancing (p. 20)
- Resilience in Elastic Load Balancing (p. 20)
- Elastic Load Balancing의 인프라 보안 (p. 21)
- Elastic Load Balancing 및 인터페이스 VPC 종단점 (p. 22)

Elastic Load Balancing의 데이터 보호

AWS 공동 책임 모델은 Elastic Load Balancing의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS는 모든 AWS 클라우드를 실행하는 글로벌 인프라를 보호해야 합니다. 이 인프라에서 호스팅되는 콘텐츠에 대한 제어를 유지하는 것은 사용자의 책임입니다. 이 콘텐츠에는 사용하는 AWS 서비스에 대한 보안 구성 및 관리 작업이 포함됩니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 FAQ를 참조하십시오. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델과 GDPR 블로그 게시물을 참조하십시오.

데이터를 보호하려면 AWS 계정 자격 증명을 보호하고 AWS Identity and Access Management(IAM)을 사용해 개별 사용자 계정을 설정하는 것이 좋습니다. 이러한 방식에서는 각 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정마다 멀티 팩터 인증(MFA)을 사용합니다.

- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2 이상을 권장합니다.
- AWS CloudTrail로 API 및 사용자 활동 로깅을 설정합니다.
- AWS 암호화 솔루션을 AWS 서비스 내의 모든 기본 보안 컨트롤과 함께 사용합니다.
- Amazon S3에 저장된 개인 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령줄 인터페이스 또는 API를 통해 AWS에 액세스할 때 FIPS 140-2 검증된 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 Federal Information Processing Standard(FIPS) 140-2를 참조하십시오.

이름 필드와 같은 자유 형식 필드에 고객 계정 번호와 같은 중요 식별 정보를 절대 입력하지 마십시오. 여기에는 Elastic Load Balancing 또는 기타 AWS 제품에서 콘솔, API, AWS CLI 또는 AWS SDK를 사용하여 작업하는 경우가 포함됩니다. Elastic Load Balancing 또는 기타 서비스에 입력하는 모든 데이터는 진단 로그에 포함하기 위해 선택될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명 정보를 URL에 포함시키지 마십시오.

저장된 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 버킷에 저장되기 전에 자동으로 암호화합니다. 또한 Elastic Load Balancing은 액세스할 때 액세스 로그 파일을 해독합니다. 각 로그 파일은 고유 키로 암호화되며, 주기적으로 바뀌는 마스터 키를 사용하여 키 자체가 암호화됩니다.

전송 중 데이터 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)를 사용하여 서버 인증서를 관리할 수 있습니다. Application Load Balancer는 HTTPS 리스너를 지원하고, Network Load Balancer는 TLS 리스너를 지원하며, Classic Load Balancer는 HTTPS 및 TLS 리스너를 모두 지원합니다.

Elastic Load Balancing의 ID 및 액세스 관리

AWS는 보안 자격 증명을 사용하여 사용자를 식별하고 AWS 리소스에 대한 액세스 권한을 부여합니다. AWS Identity and Access Management(IAM)의 기능을 사용하여 다른 사용자, 서비스 및 애플리케이션이 AWS 리소스를 완전히 또는 제한된 방식으로 사용하게 할 수 있습니다. 이를 위해 보안 자격 증명을 공유하지 않아도 됩니다.

기본값으로 IAM 사용자는 AWS 리소스를 생성, 확인 또는 수정할 수 있는 권한이 없습니다. IAM 사용자가 로드 밸런서와 같은 리소스에 액세스하여 작업을 수행하도록 허용하려면 다음을 수행하십시오.

1. IAM 사용자에게 특정 리소스와 API 작업을 사용할 권한을 부여하는 IAM 정책을 생성합니다.
2. 정책을 IAM 사용자 또는 IAM 사용자가 속한 그룹에 연결합니다.

사용자 또는 사용자 그룹에 정책을 연결하면 지정된 리소스에 대해 지정된 작업을 수행할 권한이 허용되거나 거부됩니다.

예를 들어, IAM을 사용하여 AWS 계정 아래에 사용자 및 그룹을 생성할 수 있습니다. IAM 사용자는 사용자, 시스템 또는 애플리케이션입니다. 그런 다음 정책을 사용하여 지정된 리소스에 대한 특정 작업을 수행할 수 있도록 사용자 및 그룹에 권한을 부여합니다.

IAM 정책을 사용하여 권한 부여

사용자 또는 사용자 그룹에 정책을 연결하면 지정된 리소스에 대해 지정된 작업을 수행할 권한이 허용되거나 거부됩니다.

IAM 정책은 하나 이상의 명령문으로 구성된 JSON 문서입니다. 각 문은 다음 예와 같이 구성됩니다.

```
{
  "Version": "2012-10-17",
  "Statement":[{
    "Effect": "effect",
    "Action": "action",
    "Resource": "resource-arn",
    "Condition": {
      "condition": {
        "key":"value"
      }
    }
  }]
}
```

- 효과— 효과는 `Allow` 또는 `Deny`일 수 있습니다. 기본적으로 IAM 사용자에게는 리소스 및 API 작업을 사용할 권한이 없으므로 모든 요청이 거부됩니다. 명시적 허용은 기본 설정을 무시합니다. 명시적 거부는 모든 허용을 무시합니다.
- 작업— 작업은 권한을 허가하거나 거부하기 위한 특정 API 작업입니다. 작업 지정에 대한 자세한 내용은 Elastic Load Balancing에 대한 API 작업 (p. 11)를 참조하십시오.
- 리소스— 리소스는 작업의 영향을 받습니다. Elastic Load Balancing API 작업이 많을 경우, 특정 로드 밸런서에 대해 부여 또는 거부되는 권한을 제한할 수 있습니다. 이렇게 하려면 이 명령문에서 ARN(Amazon 리소스 이름)을 지정합니다. 그렇지 않으면 * 와일드카드를 사용하여 로드 밸런서 전체를 지정할 수 있습니다. 자세한 내용은 Elastic Load Balancing 리소스 (p. 12)를 참조하십시오.
- 조건— 정책이 시행 중일 때 관리하기 위해 조건을 선택적으로 사용할 수 있습니다. 자세한 내용은 Elastic Load Balancing의 조건 키 (p. 15)를 참조하십시오.

자세한 내용은 IAM 사용 설명서 단원을 참조하십시오.

Elastic Load Balancing에 대한 API 작업

IAM 정책문의 작업 요소에서 사용자는 Elastic Load Balancing가 제공하는 API 작업을 지정할 수 있습니다. 다음 예와 같이 작업 이름 앞에 접두사로 소문자 문자열 `elasticloadbalancing:`을 붙여야 합니다.

```
"Action": "elasticloadbalancing:DescribeLoadBalancers"
```

명령문 하나에 여러 작업을 지정하려면 다음 예제와 같이 대괄호로 묶은 후 각 작업을 쉼표로 구분합니다.

```
"Action": [
    "elasticloadbalancing:DescribeLoadBalancers",
    "elasticloadbalancing:DeleteLoadBalancer"
]
```

* 와일드카드를 사용하여 여러 작업을 지정할 수도 있습니다. 다음 예에서는 `Describe`로 시작되는 Elastic Load Balancing에 대해 모든 API 작업 이름을 지정합니다.

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

Elastic Load Balancing에 대해 모든 API 작업을 지정하려면 다음 예와 같이 * 와일드카드를 사용하십시오.

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

Elastic Load Balancing를 위한 API 작업의 전체 목록은 다음 설명서를 참조하십시오.

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

Elastic Load Balancing 리소스

리소스 수준 권한은 사용자가 작업을 수행할 수 있는 리소스를 지정할 수 있는 기능입니다. Elastic Load Balancing는 리소스 수준 권한을 부분적으로 지원합니다. 리소스 수준 권한을 지원하는 API 작업의 경우, 사용자가 작업에서 사용할 수 있도록 허용된 리소스를 제어할 수 있습니다. 정책 명령문에서 로드 리소스를 지정하려면 Amazon 리소스 이름(ARN)을 사용해야 합니다. ARN을 지정할 때 경로에 * 와일드카드를 사용할 수 있습니다. 예를 들어, 정확한 로드 밸런서 이름을 지정하길 원치 않는 경우에는 * 와일드카드를 사용할 수 있습니다.

Application Load Balancer에 대한 ARN의 형식은 다음 예제와 같습니다.

```
arn:aws:elasticloadbalancing:region-code:account-id:loadbalancer/app/load-balancer-name/load-balancer-id
```

Network Load Balancer에 대한 ARN의 형식은 다음 예제와 같습니다.

```
arn:aws:elasticloadbalancing:region-code:account-id:loadbalancer/net/load-balancer-name/load-balancer-id
```

Classic Load Balancer에 대한 ARN의 형식은 다음 예제와 같습니다.

```
arn:aws:elasticloadbalancing:region-code:account-id:loadbalancer/load-balancer-name
```

Application Load Balancer의 경우 리스너 및 리스너 규칙에 대한 ARN의 형식은 다음 예제와 같습니다.

```
arn:aws:elasticloadbalancing:region-code:account-id:listener/app/load-balancer-name/load-balancer-id/listener-id
arn:aws:elasticloadbalancing:region-code:account-id:listener-rule/app/load-balancer-name/load-balancer-id/listener-id/rule-id
```

Network Load Balancer의 경우 리스터에 대한 ARN의 형식은 다음 예제와 같습니다.

```
arn:aws:elasticloadbalancing:region-code:account-id:listener/net/load-balancer-name/load-balancer-id/listener-id
```

대상 그룹에 대한 ARN의 형식은 다음 예제와 같습니다.

```
arn:aws:elasticloadbalancing:region-code:account-id:targetgroup/target-group-name/target-group-id
```

리소스 수준 권한이 지원되지 않는 API 작업

다음 Elastic Load Balancing 작업은 리소스 수준 권한을 지원하지 않습니다.

- API 버전 2015-12-01:

  - `DescribeAccountLimits`
  - `DescribeListenerCertificates`
  - `DescribeListeners`
  - `DescribeLoadBalancerAttributes`
  - `DescribeLoadBalancers`
  - `DescribeRules`
  - `DescribeSSLPolicies`
  - `DescribeTags`
  - `DescribeTargetGroupAttributes`
  - `DescribeTargetGroups`
  - `DescribeTargetHealth`
- API 버전 2012-06-01:
  - `DescribeInstanceHealth`
  - `DescribeLoadBalancerAttributes`
  - `DescribeLoadBalancerPolicyTypes`
  - `DescribeLoadBalancers`
  - `DescribeLoadBalancerPolicies`
  - `DescribeTags`

리소스 수준 권한을 지원하지 않는 API 작업의 경우 다음 예제와 같이 리소스 명령문을 지정해야 합니다.

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

Elastic Load Balancing의 리소스 수준 권한

다음 표에는 리소스 수준 권한을 지원하는 Elastic Load Balancing 작업과 각 작업에 지원되는 리소스가 나와 있습니다.

API 버전 2015-12-01

API 작업	리소스 ARN
`AddListenerCertificates`	리스너
`AddTags`	로드 밸런서, 대상 그룹
`CreateListener`	로드 밸런서
`CreateLoadBalancer`	로드 밸런서
`CreateRule`	리스너
`CreateTargetGroup`	대상 그룹
`DeleteListener`	리스너
`DeleteLoadBalancer`	로드 밸런서
`DeleteRule`	리스너 규칙
`DeleteTargetGroup`	대상 그룹
`DeregisterTargets`	대상 그룹

API 작업	리소스 ARN
ModifyListener	리스너
ModifyLoadBalancerAttributes	로드 밸런서
ModifyRule	리스너 규칙
ModifyTargetGroup	대상 그룹
ModifyTargetGroupAttributes	대상 그룹
RegisterTargets	대상 그룹
RemoveListenerCertificates	리스너
RemoveTags	로드 밸런서, 대상 그룹
SetIpAddressType	로드 밸런서
SetRulePriorities	리스너 규칙
SetSecurityGroups	로드 밸런서
SetSubnets	로드 밸런서

API 버전 2012-06-01

API 작업	리소스 ARN
AddTags	로드 밸런서
ApplySecurityGroupsToLoadBalancer	로드 밸런서
AttachLoadBalancerToSubnets	로드 밸런서
ConfigureHealthCheck	로드 밸런서
CreateAppCookieStickinessPolicy	로드 밸런서
CreateLBCookieStickinessPolicy	로드 밸런서
CreateLoadBalancer	로드 밸런서
CreateLoadBalancerListeners	로드 밸런서
CreateLoadBalancerPolicy	로드 밸런서
DeleteLoadBalancer	로드 밸런서
DeleteLoadBalancerListeners	로드 밸런서
DeleteLoadBalancerPolicy	로드 밸런서
DeregisterInstancesFromLoadBalancer	로드 밸런서
DetachLoadBalancerFromSubnets	로드 밸런서
DisableAvailabilityZonesForLoadBalancer	로드 밸런서
EnableAvailabilityZonesForLoadBalancer	로드 밸런서

API 작업	리소스 ARN
ModifyLoadBalancerAttributes	로드 밸런서
RegisterInstancesWithLoadBalancer	로드 밸런서
RemoveTags	로드 밸런서
SetLoadBalancerListenerSSLCertificate	로드 밸런서
SetLoadBalancerPoliciesForBackendServer	로드 밸런서
SetLoadBalancerPoliciesOfListener	로드 밸런서

Elastic Load Balancing의 조건 키

정책을 만들 때 정책 적용 시기를 제어하는 조건을 지정할 수 있습니다. 각 조건에는 하나 이상의 키-값 쌍이 포함됩니다. 조건 키에는 전역 조건 키와 서비스별 조건 키가 있습니다.

`aws:SourceIp` 조건 키는 Elastic Load Balancing과 함께 사용할 수 없습니다.

`elasticloadbalancing:ResourceTag/#` 조건 키는 Elastic Load Balancing에 특정합니다. 다음 작업은 이 조건 키를 지원합니다.

API 버전 2015-12-01

- AddTags
- CreateListener
- CreateLoadBalancer
- DeleteLoadBalancer
- DeleteTargetGroup
- DeregisterTargets
- ModifyLoadBalancerAttributes
- ModifyTargetGroup
- ModifyTargetGroupAttributes
- RegisterTargets
- RemoveTags
- SetIpAddressType
- SetSecurityGroups
- SetSubnets

API 버전 2012-06-01

- AddTags
- ApplySecurityGroupsToLoadBalancer
- AttachLoadBalancersToSubnets
- ConfigureHealthCheck
- CreateAppCookieStickinessPolicy
- CreateLBCookieStickinessPolicy
- CreateLoadBalancer
- CreateLoadBalancerListeners

- `CreateLoadBalancerPolicy`
- `DeleteLoadBalancer`
- `DeleteLoadBalancerListeners`
- `DeleteLoadBalancerPolicy`
- `DeregisterInstancesFromLoadBalancer`
- `DetachLoadBalancersFromSubnets`
- `DisableAvailabilityZonesForLoadBalancer`
- `EnableAvailabilityZonesForLoadBalancer`
- `ModifyLoadBalancerAttributes`
- `RegisterInstancesWithLoadBalancer`
- `RemoveTags`
- `SetLoadBalancerListenerSSLCertificate`
- `SetLoadBalancerPoliciesForBackendServer`
- `SetLoadBalancerPoliciesOfListener`

전역 조건 키에 대한 자세한 내용은 IAM 사용 설명서에서 AWS 전역 조건 컨텍스트 키를 참조하십시오.

다음 작업은 `aws:RequestTag/#` 및 `aws:TagKeys` 조건 키를 지원합니다.

- `AddTags`
- `CreateLoadBalancer`
- `RemoveTags`

사전 정의된 AWS 관리형 정책

AWS가 생성한 관리형 정책은 일반 사용 사례에서 필요한 권한을 부여합니다. Elastic Load Balancing에 필요한 액세스를 기반으로 IAM 사용자에게 이러한 정책을 연결할 수 있습니다.

- ElasticLoadBalancingFullAccess — Elastic Load Balancing 기능을 사용하는 데 필요한 모든 액세스 권한을 부여합니다.
- ElasticLoadBalancingReadOnly — Elastic Load Balancing 기능에 대한 읽기 전용 액세스 권한을 부여합니다.

각 Elastic Load Balancing 작업별 필수 권한에 대한 자세한 내용은 Elastic Load Balancing API 권한 (p. 16)를 참조하십시오.

Elastic Load Balancing API 권한

Elastic Load Balancing에 대한 API 작업 (p. 11)에 설명된 대로 필요한 Elastic Load Balancing API 작업을 호출할 수 있는 IAM 사용자 권한을 부여해야 합니다. 또한 일부 Elastic Load Balancing 작업을 위해 Amazon EC2 API에서 특정 작업을 호출할 수 있는 IAM 사용자 권한을 부여해야 합니다.

2015-12-01 API의 필수 권한

2015-12-01 API에서 다음 작업을 호출할 때 지정된 작업을 호출할 수 있는 IAM 사용자 권한을 부여해야 합니다.

`CreateLoadBalancer`

- `elasticloadbalancing:CreateLoadBalancer`

- ec2:DescribeAccountAttributes
- ec2:DescribeAddresses
- ec2:DescribeInternetGateways
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:DescribeVpcs
- iam:CreateServiceLinkedRole

CreateTargetGroup

- elasticloadbalancing:CreateTargetGroup
- ec2:DescribeInternetGateways
- ec2:DescribeVpcs

RegisterTargets

- elasticloadbalancing:RegisterTargets
- ec2:DescribeInstances
- ec2:DescribeInternetGateways
- ec2:DescribeSubnets
- ec2:DescribeVpcs

SetIpAddressType

- elasticloadbalancing:SetIpAddressType
- ec2:DescribeSubnets

SetSubnets

- elasticloadbalancing:SetSubnets
- ec2:DescribeSubnets

2012-06-01 API의 필수 권한

2012-06-01 API에서 다음 작업을 호출할 때 지정된 작업을 호출할 수 있는 IAM 사용자 권한을 부여해야 합니다.

ApplySecurityGroupsToLoadBalancer

- elasticloadbalancing:ApplySecurityGroupsToLoadBalancer
- ec2:DescribeAccountAttributes
- ec2:DescribeSecurityGroups

AttachLoadBalancerToSubnets

- elasticloadbalancing:AttachLoadBalancerToSubnets
- ec2:DescribeSubnets

CreateLoadBalancer

- elasticloadbalancing:CreateLoadBalancer
- ec2:CreateSecurityGroup
- ec2:DescribeAccountAttributes
- ec2:DescribeInternetGateways
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:DescribeVpcs
- iam:CreateServiceLinkedRole

DeregisterInstancesFromLoadBalancer

- elasticloadbalancing:DeregisterInstancesFromLoadBalancer
- ec2:DescribeClassicLinkInstances
- ec2:DescribeInstances

DescribeInstanceHealth

- elasticloadbalancing:DescribeInstanceHealth
- ec2:DescribeClassicLinkInstances
- ec2:DescribeInstances

DescribeLoadBalancers

- elasticloadbalancing:DescribeLoadBalancers
- ec2:DescribeSecurityGroups

DisableAvailabilityZonesForLoadBalancer

- elasticloadbalancing:DisableAvailabilityZonesForLoadBalancer
- ec2:DescribeAccountAttributes
- ec2:DescribeInternetGateways
- ec2:DescribeVpcs

EnableAvailabilityZonesForLoadBalancer

- elasticloadbalancing:EnableAvailabilityZonesForLoadBalancer
- ec2:DescribeAccountAttributes
- ec2:DescribeInternetGateways
- ec2:DescribeSubnets
- ec2:DescribeVpcs

RegisterInstancesWithLoadBalancer

- elasticloadbalancing:RegisterInstancesWithLoadBalancer
- ec2:DescribeAccountAttributes
- ec2:DescribeClassicLinkInstances
- ec2:DescribeInstances
- ec2:DescribeVpcClassicLink

Elastic Load Balancing 서비스 연결 역할

Elastic Load Balancing는 사용자를 대신하여 다른 AWS 서비스를 자동으로 호출하는 데 필요한 권한에 대해 서비스 연결 역할을 사용합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 사용을 참조하십시오.

서비스 연결 역할에 의해 부여된 권한

Elastic Load Balancing은 AWSServiceRoleForElasticLoadBalancing라는 서비스 연결 역할을 사용하여 다음 작업을 자동으로 호출합니다.

- ec2:DescribeAddresses
- ec2:DescribeInstances
- ec2:DescribeNetworkInterfaces
- ec2:DescribeSubnets
- ec2:DescribeSecurityGroups
- ec2:DescribeVpcs

- ec2:DescribeInternetGateways
- ec2:DescribeAccountAttributes
- ec2:DescribeClassicLinkInstances
- ec2:DescribeVpcClassicLink
- ec2:CreateSecurityGroup
- ec2:CreateNetworkInterface
- ec2:DeleteNetworkInterface
- ec2:ModifyNetworkInterfaceAttribute
- ec2:AuthorizeSecurityGroupIngress
- ec2:AssociateAddress
- ec2:DisassociateAddress
- ec2:AttachNetworkInterface
- ec2:DetachNetworkInterface
- ec2:AssignPrivateIpAddresses
- ec2:AssignIpv6Addresses
- ec2:UnassignIpv6Addresses
- logs:CreateLogDelivery
- logs:GetLogDelivery
- logs:UpdateLogDelivery
- logs:DeleteLogDelivery
- logs:ListLogDeliveries

AWSServiceRoleForElasticLoadBalancing은 역할을 맡을 elasticloadbalancing.amazonaws.com 서비스를 신뢰합니다.

서비스 연결 역할 생성

AWSServiceRoleForElasticLoadBalancing 역할을 수동으로 생성할 필요는 없습니다. 사용자가 로드 밸런서를 생성할 때 Elastic Load Balancing가 이 역할을 생성합니다.

Elastic Load Balancing가 서비스 연결 역할을 자동으로 생성하는 경우 사용자에게 필수 권한이 있어야 합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한 단원을 참조하십시오.

사용자가 2018년 1월 11일 전에 로드 밸런서를 생성한 경우 Elastic Load Balancing가 사용자의 AWS 계정에 AWSServiceRoleForElasticLoadBalancing를 생성했습니다. 자세한 내용은 IAM 사용 설명서의 내 AWS 계정에 표시되는 새 역할을 참조하세요.

서비스 연결 역할 편집

사용자는 IAM를 사용하여 AWSServiceRoleForElasticLoadBalancing의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 편집을 참조하십시오.

서비스 연결 역할 삭제

Elastic Load Balancing를 더 이상 사용할 필요가 없다면 AWSServiceRoleForElasticLoadBalancing을 삭제하는 것이 좋습니다.

이 서비스 연결 역할은 AWS 계정에서 로드 밸런서를 모두 삭제한 후에만 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 참조하십시오.

IAM 콘솔, IAM CLI 또는 IAM API를 사용하여 서비스 연결 역할을 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스에 연결 역할 삭제 단원을 참조하십시오.

AWSServiceRoleForElasticLoadBalancing을 삭제한 후 사용자가 로드 밸런서를 생성한 경우에는 Elastic Load Balancing가 역할을 다시 생성합니다.

Compliance validation for Elastic Load Balancing

외부 감사자는 여러 AWS 규정 준수 프로그램의 일환으로 Elastic Load Balancing의 보안 및 규정 준수를 평가합니다. 여기에는 SOC, PCI, FedRAMP, HIPAA 등이 포함됩니다.

특정 규정 준수 프로그램의 범위 내에 있는 AWS 서비스 목록은 규정 준수 프로그램 제공 범위 내 AWS 서비스를 참조하십시오. 일반적인 내용은 AWS 규정 준수 프로그램을 참조하십시오.

AWS Artifact를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 AWS Artifact에서 보고서 다운로드를 참조하십시오.

Elastic Load Balancing 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 결정됩니다. AWS에서는 규정 준수에 도움이 되도록 다음과 같은 리소스를 제공합니다.

- Security and compliance quick start guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- Architecting for HIPAA security and compliance whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS compliance resources – This collection of workbooks and guides might apply to your industry and location.
- Evaluating resources with rules in the AWS Config Developer Guide – AWS Config; assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Resilience in Elastic Load Balancing

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다. 리전은 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며 이러한 가용 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크를 통해 연결되어 있습니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 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 참조하십시오.

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

- Distributes incoming traffic across multiple instances in a single Availability Zone or multiple Availability Zones.
- You can use AWS Global Accelerator with your Application Load Balancer to distribute incoming traffic across multiple load balancers in one or more AWS Regions. For more information, see the AWS Global Accelerator 개발자 안내서.
- Amazon ECS enables you to run, stop, and manage Docker containers on a cluster of EC2 instances. You can configure your Amazon ECS service to use a load balancer to distribute incoming traffic across

the services in a cluster. For more information, see the Amazon Elastic Container Service Developer Guide.

Elastic Load Balancing의 인프라 보안

관리형 서비스인 Elastic Load Balancing는 Amazon Web Services: 보안 프로세스 개요 백서에 설명된 AWS 글로벌 네트워크 보안 절차로 보호됩니다.

AWS에서 게시한 API 호출을 사용하여 네트워크를 통해 Elastic Load Balancing에 액세스합니다. 클라이언트가 TLS(전송 계층 보안) 1.0 이상을 지원해야 합니다. TLS 1.2 이상을 권장합니다. 클라이언트는 Ephemeral Diffie-Hellman(DHE) 또는 Elliptic Curve Ephemeral Diffie-Hellman(ECDHE)과 같은 PFS(전달 완전 보안, Perfect Forward Secrecy)가 포함된 암호 제품군도 지원해야 합니다. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 주체와 연결된 보안 액세스 키를 사용해 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 보안 자격 증명을 생성하여 요청에 서명할 수 있습니다.

네트워크 격리

Virtual Private Cloud(VPC)는 AWS 클라우드에서 논리적으로 격리된 고유한 영역의 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 설명서 단원을 참조하십시오.

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.

퍼블릭 인터넷을 통해 트래픽을 보내지 않고 VPC에서 Elastic Load Balancing API를 호출하려면 AWS PrivateLink를 사용하십시오. 자세한 내용은 Elastic Load Balancing 및 인터페이스 VPC 종단점 (p. 22) 단원을 참조하십시오.

네트워크 트래픽 제어

Elastic Load Balancing은 Application Load Balancer, Network Load Balancer 및 Classic Load Balancer 세 가지 유형의 로드 밸런서를 지원합니다. Application Load Balancer는 OSI(개방형 시스템 간 상호 연결) 모델의 요청 수준(계층 7)에서 작동하고, Network Load Balancer는 OSI 모델의 연결 수준(계층 4)에서 작동하며, Classic Load Balancer는 요청 및 연결 수준 모두에서 작동합니다.

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

- 클라이언트와 로드 밸런서 간의 암호화된 통신을 지원하려면 보안 리스너를 사용하십시오. Application Load Balancer는 HTTPS 리스너를 지원하고, Network Load Balancer는 TLS 리스너를 지원하며, 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)를 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 구성하십시오. 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 참조하십시오.
- AWS WAF와 함께 Application Load Balancer를 사용하여 웹 ACL(웹 액세스 제어 목록)의 규칙에 따라 요청을 허용하거나 차단합니다.

Elastic Load Balancing 및 인터페이스 VPC 종단점

인터페이스 VPC 엔드포인트를 생성하여 Virtual Private Cloud(VPC)와 Elastic Load Balancing API 간에 프라이빗 연결을 설정할 수 있습니다. 인터넷을 통해 트래픽을 보내지 않고 이 연결을 사용하여 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 제공합니다. 이 작업은 인터넷 게이트웨이, NAT 인스턴스 또는 VPN 연결이 필요하지 않습니다.

인터페이스 VPC 엔드포인트는 프라이빗 IP 주소를 사용하여 AWS 서비스 간에 프라이빗 통신을 활성화하는 기능인 AWS PrivateLink에 의해 구동됩니다. 자세한 내용은 AWS PrivateLink 섹션을 참조하세요.

한도

AWS PrivateLink는 리스너가 50개 이상인 Network Load Balancer를 지원하지 않습니다.

Elastic Load Balancing용 인터페이스 엔드포인트 생성

다음 서비스 이름을 사용하여 Elastic Load Balancing에 대한 엔드포인트를 생성합니다.

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

자세한 내용은 Amazon VPC 사용 설명서의 인터페이스 엔드포인트 생성을 참조하십시오.

Elastic Load Balancing에 대한 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 내용은 Amazon VPC 사용 설명서에서 VPC 엔드포인트 정책 사용을 참조하십시오.

Classic Load Balancer 마이그레이션

Elastic Load Balancing는 Classic Load Balancer, Application Load Balancer, Network Load Balancer 등 세 가지 유형의 로드 밸런서를 제공합니다. 각 로드 밸런서 유형의 다양한 기능에 대한 자세한 내용은 Elastic Load Balancing 제품 비교를 참조하십시오.

마이그레이션 시나리오

1. 기존 Classic Load Balancer가 VPC에 있는 경우 Application Load Balancer 또는 Network Load Balancer가 요구 사항을 충족하는지 확인한 다음 Classic Load Balancer를 이러한 로드 밸런서 유형 중 하나로 마이그레이션합니다.
2. 기존 Classic Load Balancer가 EC2-Classic에 있는 경우 Application Load Balancer 또는 Network Load Balancer가 요구 사항을 충족하는지 확인한 다음 Classic Load Balancer를 이러한 로드 밸런서 유형 중 하나로 마이그레이션합니다. 그렇지 않으면 VPC의 Classic Load Balancer로 마이그레이션합니다. 인스턴스를 EC2-Classic에 그대로 두고 ClassicLink를 활성화하여 인스턴스를 로드 밸런서 VPC에 연결하거나, 인스턴스를 EC2-Classic에서 VPC로 마이그레이션할 수 있습니다.

마이그레이션 프로세스

- 1단계: 새 로드 밸런서 생성 (p. 24)
- 2단계: 새 로드 밸런서로 점진적으로 트래픽 리디렉션 (p. 26)
- 3단계: 정책, 스크립트 및 코드 업데이트 (p. 27)
- 4단계: 이전 로드 밸런서 삭제 (p. 27)

1단계: 새 로드 밸런서 생성

마이그레이션할 Classic Load Balancer와 동등한 구성으로 로드 밸런서를 생성합니다. 마이그레이션 프로세스를 완료한 후 새 로드 밸런서의 기능을 활용할 수 있습니다.

Application Load Balancer 또는 Network Load Balancer를 생성하여 VPC의 Classic Load Balancer를 대체하려면 다음 옵션 중 하나를 사용합니다.

- 옵션 1: 콘솔에서 마이그레이션 마법사 사용 (p. 24)
- 옵션 2: github의 로드 밸런서 복사 유틸리티를 사용 (p. 25)
- 옵션 3: 수동으로 Application Load Balancer 또는 Network Load Balancer로 마이그레이션 (p. 25)

VPC에서 Classic Load Balancer를 생성하여 EC2-Classic의 Classic Load Balancer를 대체하려면 다음 옵션을 사용합니다.

- 옵션 4: 수동으로 VPC의 Classic Load Balancer로 마이그레이션 (p. 26)

옵션 1: 콘솔에서 마이그레이션 마법사 사용

마이그레이션 마법사는 VPC의 Classic Load Balancer 구성을 바탕으로 Application Load Balancer 또는 Network Load Balancer를 생성합니다. 생성되는 로드 밸런서 유형은 Classic Load Balancer 구성에 따라 다릅니다.

마이그레이션 마법사 릴리스 정보

- Classic Load Balancer가 VPC에 있어야 합니다.

- Classic Load Balancer에 HTTP 또는 HTTPS 리스너가 있으면 마법사가 Application Load Balancer를 생성합니다. Classic Load Balancer에 TCP 리스너가 있으면 마법사가 Network Load Balancer를 생성합니다.
- Classic Load Balancer 이름이 기존 Application Load Balancer 또는 Network Load Balancer 이름과 일치하면 마법사가 마이그레이션 중에 다른 이름을 지정하라고 요구합니다.
- Classic Load Balancer에서 서브넷이 한 개 있으면 마법사가 Application Load Balancer 생성 시 보조 서브넷을 지정하라고 요구합니다.
- Classic Load Balancer에 EC2-Classic에 등록된 인스턴스가 있으면 새 로드 밸런서의 대상 그룹에 등록되지 않습니다.
- Classic Load Balancer에 C1, CC1, CC2, CG1, CG2, CR1, CS1, G1, G2, HI1, HS1, M1, M2, M3, T1 유형의 등록 인스턴스가 있으면 Network Load Balancer의 대상 그룹에 등록되지 않습니다.
- Classic Load Balancer에 HTTP/HTTPS 리스너가 있지만 TCP 상태 확인을 사용하는 경우, 마법사는 리스너를 HTTP 상태 확인으로 변경합니다. 그런 다음 Application Load Balancer를 생성할 때 기본적으로 경로를 "/"로 설정합니다.
- Classic Load Balancer를 Network Load Balancer로 마이그레이션하면 상태 확인 설정이 Network Load Balancer에 대한 요구 사항을 충족하도록 변경됩니다.
- Classic Load Balancer에 HTTPS 리스너가 여러 개 있으면 마법사가 하나를 선택하여 해당 인증서와 정책을 사용합니다. 443 포트에 HTTPS 리스너가 있으면 마법사가 이 리스너를 선택합니다. 선택한 리스너가 사용자 지정 정책을 사용하거나 Application Load Balancer를 지원하지 않는 정책을 사용하는 경우 마법사가 기본 보안 정책으로 변경합니다.
- Classic Load Balancer에 보안 TCP 리스 너가 있는 경우 Network Load Balancer에서는 TCP 리스너를 사용합니다. 그러나 인증서 또는 보안 정책을 사용하지 않습니다.
- Classic Load Balancer에 여러 리스너가 있으면 마법사가 가장 낮은 값의 리스너 포트를 대상 그룹 포트로 사용합니다. 이러한 리스너에 등록된 각 인스턴스는 모든 리스너의 리스너 포트에 대한 대상 그룹에 등록됩니다.
- 태그 이름에 `aws` 접두사가 포함된 태그가 Classic Load Balancer에 있는 경우, 이러한 태그는 새 로드 밸런서에 추가되지 않습니다.

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 [Load Balancers]를 선택합니다.
3. Classic Load Balancer를 선택하십시오.
4. [Migration] 탭에서 [Launch ALB Migration Wizard] 또는 [Launch NLB Migration Wizard]를 선택합니다. 표시되는 버튼은 마법사가 Classic Load Balancer를 검사한 후 선택하는 로드 밸런서 유형에 따라 다릅니다.
5. [Review] 페이지에서 마법사가 선택하는 구성 옵션을 확인합니다. 옵션을 변경하려면 Edit를 선택합니다.
6. 새로운 로드 밸런서의 구성을 모두 마쳤으면 [Create]를 선택합니다.

옵션 2: github의 로드 밸런서 복사 유틸리티를 사용

이 로드 밸런서 복사 유틸리티는 GitHub에서 사용할 수 있습니다. 자세한 내용은 Elastic Load Balancing 도구를 참조하십시오.

옵션 3: 수동으로 Application Load Balancer 또는 Network Load Balancer로 마이그레이션

다음 정보는 VPC의 기존 Classic Load Balancer를 기반으로 새 Application Load Balancer 또는 Network Load Balancer를 수동으로 생성하는 일반적인 지침을 제공합니다. AWS Management 콘솔, AWS CLI 또는

AWS SDK를 사용하여 마이그레이션할 수 있습니다. 자세한 내용은 Elastic Load Balancing 시작하기 (p. 8) 섹션을 참조하세요.

1. Classic Load Balancer와 동일한 체계(인터넷 경계 또는 내부), 서브넷 및 보안 그룹으로 새 로드 밸런서를 생성하십시오.
2. Classic Load Balancer와 동일한 상태 확인 설정으로 로드 밸런서에 대한 하나의 대상 그룹을 생성하십시오.
3. 다음 중 하나를 수행하십시오.
   - Classic Load Balancer가 Auto Scaling 그룹에 연결되어 있는 경우 대상 그룹을 Auto Scaling 그룹에 연결하십시오. 이렇게 하면 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 추가하십시오.

옵션 4: 수동으로 VPC의 Classic Load Balancer로 마이그레이션

다음 정보는 EC2-Classic의 기존 Classic Load Balancer를 기반으로 새 Classic Load Balancer를 수동으로 생성하는 일반적인 지침을 제공합니다. AWS Management 콘솔, AWS CLI 또는 AWS SDK를 사용하여 마이그레이션할 수 있습니다. 자세한 내용은 Classic Load Balancer 사용 설명서에서 자습서: Classic Load Balancer 생성을 참조하십시오.

1. 다음 중 하나를 수행하십시오.
   - VPC에서 ClassicLink를 활성화하고 EC2-Classic 인스턴스를 VPC에 연결합니다. 자세한 내용은 Amazon EC2 사용 설명서에서 증분 마이그레이션에 Classiclink 사용을 참조하십시오.
   - EC2 리소스(예: 인스턴스 및 보안 그룹)를 EC2-Classic에서 VPC로 마이그레이션합니다. 자세한 내용은 Amazon EC2 사용 설명서에서 VPC로 리소스 마이그레이션을 참조하십시오.
2. VPC에서 새 Classic Load Balancer를 생성합니다.
3. 로드 밸런서를 생성할 때 준비한 VPC(ClassicLink가 활성화된 VPC 또는 EC2-Classic에서 인스턴스를 마이그레이션할 때 사용한 VPC)를 선택합니다. 각 가용 영역에서 새 로드 밸런서에 등록할 인스턴스가 포함된 서브넷을 하나 선택합니다.
4. 보안 그룹을 할당하라는 메시지가 표시되면 ClassiclLink로 VPC를 활성화한 경우 ClassiclLink를 활성화할 때 지정한 것과 동일한 보안 그룹을 선택합니다.
5. 메시지가 표시되면 로드 밸런서에 등록할 인스턴스를 선택합니다.
6. 기존 Classic Load Balancer에 태그가 있는 경우 해당 태그를 검토하고 새 Classic Load Balancer에 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 붙여넣습니다. 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 Balancer를 Application Load Balancer 또는 Network Load Balancer로 마이그레이션한 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 사용하는 스크립트를 aws elbv2 AWS CLI 명령을 사용하도록 업데이트합니다.
- `AWS::ElasticLoadBalancing::LoadBalancer` 리소스를 사용하는 AWS CloudFormation 템플릿을 `AWS::ElasticLoadBalancingV2` 리소스를 사용하도록 업데이트합니다.
- Elastic Load Balancing API 버전 2012-06-01을 사용하는 코드를 버전 2015-12-01을 사용하도록 업데이트합니다.

리소스

- AWS CLI Command Reference의 elbv2
- Elastic Load Balancing API 참조 버전 2015-12-01
- Elastic Load Balancing의 ID 및 액세스 관리 (p. 10)
- 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

다음 작업을 수행한 후에는 기존 Classic Load Balancer를 삭제할 수 있습니다.

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

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